**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ**

**DIRETORIA DE PESQUISA E PÓS-GRADUAÇÃO**

**PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO**

**RAFAEL FERNANDES SIQUEIRA**

# REDUÇÃO DE DIMENSIONALIDADE EM BASES DE DADOS DE CLASSIFICAÇÃO HIERÁRQUICA MULTIRRÓTULO USANDO AUTOENCODERS

**DISSERTAÇÃO**

**PONTA GROSSA**

**2019**

**RAFAEL FERNANDES SIQUEIRA**

# REDUÇÃO DE DIMENSIONALIDADE EM BASES DE DADOS DE CLASSIFICAÇÃO HIERÁRQUICA MULTIRRÓTULO USANDO AUTOENCODERS

Dissertação apresentada como requisito parcial à obtenção do título de Mestre em Ciência da Computação do Programa de Pós-Graduação em Ciência da Computação, da Universidade Tecnológica Federal do Paraná - Campus Ponta Grossa.

Área de Concentração: Inteligência Computacional

Orientadora: Profª. Dra. Helyane B. Borges

**PONTA GROSSA**

**2019**

Ficha catalográfica elaborada pelo Departamento de Biblioteca
da Universidade Tecnológica Federal do Paraná, Campus Ponta Grossa
n.58/19

R845 Siqueira, Rafael Fernandes

Redução de dimensionalidade em bases de dados de classificação hierárquica multirrótulo usando autoencoders. / Rafael Fernandes Siqueira, 2019.
118 f. : il. ; 30 cm.

Orientadora: Profa. Dra. Helyane Bronoski Borges

Dissertação (Mestrado em Ciência da Computação) - Programa de Pós-Graduação em Ciência da Computação. Universidade Tecnológica Federal do Paraná, Ponta Grossa, 2019.

1. Bioinformática. 2. Análise dimensional. 3. Controle preditivo. 4. Banco de dados. I. Borges, Helyane Bronoski. II. Universidade Tecnológica Federal do Paraná. III. Título.

CDD 004

Elson Heraldo Ribeiro Junior. CRB-9/1413. 29/07/2019.

| | | |
|---|---|---|
| | **Ministério da Educação**<br>**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ**<br>Diretoria de Pesquisa e Pós-Graduação<br>Programa de Pós-Graduação em Ciência da Computação | |

# FOLHA DE APROVAÇÃO

Título de Dissertação Nº 14/2019

**REDUÇÃO DE DIMENSIONALIDADE EM BASES DE DADOS DE CLASSIFICAÇÃO HIERÁRQUICA MULTIRRÓTULO USANDO AUTOENCODERS**

Por

**Rafael Fernandes Siqueira**

Esta dissertação foi apresentada às **14 horas** de **02 de julho de2019**, na **COTED**, como requisito parcial para a obtenção do título de MESTRE EM CIÊNCIA DA COMPUTAÇÃO, Programa de Pós-Graduação em Ciência da Computação. O candidato foi arguido pela Banca Examinadora, composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho APROVADO.

**Prof. Dr. Júlio César Nievola**
**(PUCPR)**

**Profª. Drª. Simone Nasser Matos**
**(UTFPR)**

**Profª. Drª. Helyane Bronoski Borges**
**(UTFPR)**
*Orientador(a) e presidente da banca*

Universidade Tecnológica Federal do Paraná
Campus Ponta Grossa

Visto da Coordenadora:

**Profª. Drª. Helyane Bronoski Borges**
Coordenadora do PPGCC
UTFPR - Câmpus Ponta Grossa

- A FOLHA DE APROVAÇÃO ASSINADA ENCONTRA-SE ARQUIVADA NA SECRETARIA ACADÊMICA -

*"Portanto, quer comais, quer bebais*
*ou façais outra coisa qualquer,*
*fazei tudo para a glória de Deus."*
*1 Coríntios 10:31 - Bíblia Sagrada*

## AGRADECIMENTOS

Quero agradecer, em primeiro lugar, a DEUS, pelo dom da vida. Agradeço imensamente, por ter tranquilizado o meu espírito e me concedido saúde, força e disposição para vencer essa etapa e superar as dificuldades ao longo de toda minha caminhada.

Aos meus tão amados pais, Vania e Valdecyr, meus incentivadores, meus exemplos. Agradeço por todo o carinho e educação que me deram durante toda a minha criação. Mesmo com a distância e correria do dia a dia, sempre se fazem presentes. Saber que tenho vocês ao meu lado, me fez ter esperança para sempre seguir em frente.

Obrigado a minha noiva, Dilmara, que me estimulou durante todo o tempo e compreendeu minha ausência pelo tempo dedicado aos estudos, que me deu apoio e incentivo nas horas difíceis, não me deixando ser vencido pelo cansaço. Pelo companheirismo, pelo auxílio, por ouvir minhas lamentações e por sempre acreditar que eu seria capaz de vencer mais esta etapa, meu sincero muito obrigado, meu amor. Hoje sabemos que valeu a pena todo sofrimento e todas as renúncias desse período.

Um agradecimento especial a minha orientadora, Prof[a]. Helyane, que nunca perdeu a fé na minha pesquisa, esclareceu tantas dúvidas, sempre tão atenciosa e paciente. Obrigado por ter acreditado no meu potencial e exigir de mim muito mais do que eu imaginava ser capaz de fazer. Minha gratidão pela oportunidade concedida e por compartilhar comigo sua sabedoria, seu tempo e sua experiência.

À UTFPR, que ao longo da minha formação ofereceu um ambiente de estudo agradável, motivador e repleto de oportunidades.

A todos os meus amigos, especialmente Albino e Edinho. Vocês foram fundamentais para minha formação profissional, pelo constante incentivo e apoio, e principalmente pelo exemplo de profissionais que são, merecem o meu eterno agradecimento.

A todos meus colegas de trabalho, principalmente meu grande amigo Emílio, que sempre perguntava como estava o andamento da pesquisa, sempre disposto a escutar. Nos momentos difíceis e de desânimo, compartilhou valiosos conselhos.

E, por fim, sou grato a todos os professores que contribuíram com a minha trajetória acadêmica e a todos que de alguma forma contribuíram para a realização desse trabalho.

# RESUMO

SIQUEIRA, Rafael Fernandes. **Redução de dimensionalidade em bases de dados de classificação hierárquica multirrótulo usando Autoencoders**. 2019. 118 f. Dissertação (Mestrado em Ciência da Computação) - Universidade Tecnológica Federal do Paraná, Ponta Grossa, 2019.

A predição de proteínas em dados de bioinformática é um exemplo de problema de Classificação Hierárquica Multirrótulo no qual cada instância pode estar associada a múltiplas classes, e estas por sua vez, estão organizadas em uma hierarquia. A alta dimensionalidade dos atributos e das classes influencia no desempenho dos classificadores, tanto no custo computacional quanto na capacidade preditiva, pois prejudica a busca por padrões e descoberta de conhecimento útil. A extração de atributos é uma das técnicas utilizadas para alcançar a redução de dimensionalidade em base de dados, e assim eliminar atributos irrelevantes e/ou redundantes que tendem a confundir um algoritmo de aprendizagem. Nessa técnica, por meio de combinações e/ou transformações dos atributos originais, geram-se novos atributos, mais significativos e que melhor representam a base de dados, em um espaço de menor dimensão. Desse modo, neste trabalho propõe-se um novo método de extração de atributos, FEAE-HMC, para classificação hierárquica multirrótulo, baseado em conceitos e técnicas de *Deep Learning*, por meio de adaptações em uma rede *Autoencoder* clássica. O método FEAE-HMC é dividido em duas etapas principais: a extração de atributos e a avaliação do conjunto de dados reduzido por meio de um classificador hierárquico multirrótulo (Clus-HMC e MHC-CNN) e sua medida de desempenho (AUPRC). Para a realização dos experimentos são utilizados dados biológicos de 10 bases de dados da Ontologia Gênica, sendo que as classes das mesmas estão estruturadas em uma hierarquia no formato de um Grafo Acíclico Dirigido (DAG). Conforme os resultados experimentais, o método FEAE-HMC se mostrou capaz de extrair representações de menor dimensão, que podem agregar correlações entre os atributos e rótulos. Essas representações, quando submetidas a um Classificador Hierárquico Multirrótulo, geram modelos nos quais se obtêm o desempenho preditivo equivalente e até mesmo superior ao desempenho da base original. A diferença obtida entre a medida AUPRC da base completa e uma base reduzida, com uma redução de até 90% da dimensionalidade original, é inferior a 0,047 em ambos classificadores. Testes estatísticos demonstram que as bases reduzidas extraídas pelo FEAE-HMC, são no mínimo estatisticamente equivalentes as bases originais.

**Palavras-chaves:** Classificação hierárquica multirrótulo. Redução de dimensionalidade. Extração de atributos. Autoencoders.

## ABSTRACT

SIQUEIRA, Rafael Fernandes**. Dimensionality reduction in hierarchical multi-label databases using autoencoders.** 2019. 118 p. Thesis (Master's Degree in Computer Science) - Federal University of Technology - Paraná. Ponta Grossa, 2019.

Protein prediction in bioinformatics data is an example of a Hierarchical Multilabel Classification problem in which each instance can be associated with multiple classes, which in turn are organized in a hierarchy. The high dimensionality of attributes and classes influences the performance of the classifiers, both in computational cost and in predictive capacity, as it impairs the search for patterns and the discovery of useful knowledge. Feature Extraction is one of the techniques used to achieve dimensionality reduction in databases, and thus eliminate irrelevant and/or redundant attributes that tend to confuse a learning algorithm. In this technique, by means of combinations and/or transformations of the original attributes, new attributes, which are more significant and represent the database, are generated in a smaller space. Thus, this work proposes a new method of feature extraction, FEAE-HMC, for the hierarchical multi-label classification, based on concepts and techniques of Deep Learning, through adaptations in a classic Autoencoder network. The FEAE-HMC method is divided into two main steps: the feature extraction and the evaluation of the reduced data set using a hierarchical multi-label classifier (Clus-HMC and MHC-CNN) and its performance measure (AUPRC). To perform the experiments, biological data from 10 Genetic Ontology databases are used, and their classes are structured in a hierarchy in the form of a Directed Acyclic Graph (DAG). According to the experimental results, the FEAE-HMC method was able to extract representations of smaller dimension that can add correlations between the attributes and labels. These representations, when submitted to a Hierarchical Multi-label Classifier, generate models with predictive performance equivalent or even superior to the performance of the original base. The difference between the full-base AUPRC measurement and a reduced base with a reduction of up to 90% of the original dimensionality is less than 0.047 in both classifiers. Statistical tests show that the reduced bases extracted by the FEAE-HMC are at least statistically equivalent to the original bases.



**Key-Words:** Hierarchical multi-label classification. Dimensionality reduction. Feature extraction. Autoencoders.

## LISTA DE FIGURAS

Figura 1 - Hierarquia de Aprendizado ..... 20
Figura 2 - Problemas de Classificação ..... 21
Figura 3 - Técnicas para Classificação Multirrótulo ..... 22
Figura 4 - Classificação Plana x Classificação Hierárquica ..... 23
Figura 5 - Tipos de Hierarquia de Classe ..... 24
Figura 6 - Passos básicos para a seleção de atributos ..... 33
Figura 7 - Abordagem wrapper para seleção de atributos ..... 34
Figura 8 - Esquema da extração de atributos ..... 36
Figura 9 - O Neurônio Artificial ..... 39
Figura 10 - Funções de ativação - Linear, Limiar e Sigmoide ..... 40
Figura 11 - Funções de ativação - Tanh e ReLU ..... 41
Figura 12 - Exemplo de Rede Neural com duas camadas ocultas ..... 42
Figura 13 - Evolução da Inteligência Artificial ..... 43
Figura 14 - Estrutura de um Autoencoder de 3 camadas ..... 46
Figura 15 - Estrutura de um Autoencoder com múltiplas camadas ocultas ..... 47
Figura 16 - Funcionamento do Algoritmo Backpropagation ..... 53
Figura 17 - Etapas da Revisão ..... 61
Figura 18 - Funcionamento Geral do Método ..... 76
Figura 19 - Hierarquia das classes ..... 77
Figura 20 - Etapas do Método Proposto ..... 79
Figura 21 - Estrutura da rede DSC-AE ..... 80
Figura 22 - Redução de Dimensionalidade usando a rede FEAE-HMC ..... 84
Figura 23 - Metodologia para realização dos experimentos ..... 88
Figura 24 - Rede DSC-AE - Redução de 90% - conjunto CellCycle ..... 90
Figura 25 - Comparativo de desempenho médio dos classificadores nas bases reduzidas ..... 100
Figura 26 - Comparativo de desempenho máximo dos classificadores nas bases reduzidas ..... 100
Figura 27 - Comparativo de desempenho - Base com 90% de redução x Base Completa ..... 101

## LISTA DE QUADROS

Quadro 1 - Categorização quanto à estrutura e técnica empregada ..... 45
Quadro 2 - Algoritmo de treinamento de um Autoencoder ..... 51
Quadro 3 - Resumo dos passos realizados pelo algoritmo de retropropagação ..... 54
Quadro 4 - Questões de Pesquisa ..... 62
Quadro 5 - Definição das bases de pesquisa ..... 63
Quadro 6 - Definição das palavras-chave ..... 63
Quadro 7 - Definição das strings de busca ..... 63
Quadro 8 - Descrição dos Artigos de Periódicos selecionados ..... 66
Quadro 9 - Descrição dos Artigos em Conferência selecionados ..... 66
Quadro 10 - Métodos e Técnicas de Redução de dimensionalidade ..... 67
Quadro 11 - Métodos e Técnicas de Redução x Quantidade ..... 68
Quadro 12 - Método e Técnica x Quantidade de Trabalhos ..... 69
Quadro 13 - Área de aplicação x Tipo de Hierarquia ..... 69
Quadro 14 - Artigo x Contribuição científica na HMC ..... 72
Quadro 15 - Equações para definição do número de neurônios ..... 81
Quadro 16 - Equações e resultados para definição do número de neurônios ..... 81
Quadro 17 - Estrutura de Rede em relação ao número de neurônios ..... 82

## LISTA DE TABELAS

Tabela 1 - Total de Resultados das buscas ..... 64
Tabela 2 - Aplicação das etapas 1 e 2 no processo de filtragem ..... 64
Tabela 3 - Aplicação da etapa 3 no processo de filtragem ..... 65
Tabela 4 - Aplicação da etapa 4 no processo de filtragem ..... 65
Tabela 5 - Ranking de ordenação dos trabalhos - Periódicos ..... 65
Tabela 6 - Base de dados fictícia ..... 77
Tabela 7 - Bases fictícias de treinamento e teste ..... 78
Tabela 8 - Parâmetros das redes DSC-AE e FEAE-HMC ..... 88
Tabela 9 - Características das Bases de Dados GO ..... 87
Tabela 10 - Número de neurônios por camada na rede DSC-AE para base Cellcycle ..... 89
Tabela 11 - Medida AUPRC - CLUS-HMC - sem a aplicação do método FEAE-HMC ..... 92
Tabela 12 - Comparativo entre atributos das bases originais e bases reduzidas obtidas ..... 92
Tabela 13 - Medidas AUPRC para os conjuntos de dados reduzidos - Cellcycle ..... 93
Tabela 14 - Comparativo da medida AURPC nos diferentes percentuais - Cellcycle ..... 93
Tabela 15 - Medidas AUPRC para as bases reduzidas do conjunto de dados Derisi ..... 94
Tabela 16 - Comparativo da medida AURPC nos diferentes percentuais - Derisi ..... 95
Tabela 17 - Medidas AUPRC para as bases reduzidas do conjunto de dados Gasch1 ..... 95
Tabela 18 - Comparativo da medida AURPC nos diferentes percentuais - Gasch1 ..... 96
Tabela 19 - Comparativo da medida AURPC - CLUS-HMC ..... 97
Tabela 20 - Medida AUPRC - MHC-CNN - sem a aplicação do método FEAE-HMC ..... 98
Tabela 21 - Comparativo da medida AURPC - MHC-CNN ..... 98
Tabela 22 - Medidas AUPRC para os conjuntos de dados reduzidos - Church ..... 111
Tabela 23 - Comparativo da medida AURPC nos diferentes percentuais - Church ..... 111
Tabela 24 - Medidas AUPRC para as bases reduzidas do conjunto de dados Eisen ..... 112
Tabela 25 - Comparativo da medida AURPC nos diferentes percentuais - Eisen ..... 112
Tabela 26 - Medidas AUPRC para as bases reduzidas do conjunto de dados Expr ..... 113
Tabela 27 - Comparativo da medida AURPC nos diferentes percentuais - Expr ..... 114
Tabela 28 - Medidas AUPRC para as bases reduzidas do conjunto de dados Gasch2 ..... 114
Tabela 29 - Comparativo da medida AURPC nos diferentes percentuais - Gasch2 ..... 115
Tabela 30 - Medidas AUPRC para as bases reduzidas do conjunto de dados Pheno ..... 116
Tabela 31 - Comparativo da medida AURPC nos diferentes percentuais - Pheno ..... 116
Tabela 32 - Medidas AUPRC para os conjuntos de dados reduzidos - Seq ..... 117
Tabela 33 - Comparativo da medida AURPC nos diferentes percentuais - Seq ..... 117
Tabela 34 - Medidas AUPRC para os conjuntos de dados reduzidos - Spo ..... 118
Tabela 35 - Comparativo da medida AURPC nos diferentes percentuais - Spo ..... 118

## LISTA DE SIGLAS

| | |
|---|---|
| API | *Application Programming Interface* |
| AUPRC | *Area Under the Precision Recall Curve* |
| CNN | *Convolutional Neural Network* |
| DAG | *Directed Acyclic Graph* |
| DBM | *Deep Boltzmann Machines* |
| DNN | *Deep Neural Network* |
| EHD | *Edge Histogram Descriptors* |
| GO | *Gene Ontology* |
| HLSE | *Hierarchical Label Set Expansion* |
| HMC | *Hierarchical Multi-label Classification* |
| JCR | *Journal Citation Reports* |
| SJR | *Scientific Journal Rankings* |
| LBP | *Local Binary Patterns* |
| LCPN | *Local Classifier per Node* |
| LDA | *Linear Discriminant Analysis* |
| MDS | *Multidimensional Scaling* |
| MHC-CNN | *Multi-label Hierarchical Classification - Competitive Neural Network* |
| MLP | *Multilayer Perceptron* |
| MSE | *Mean Squared Error* |
| NMF | *Nonnegative Matrix Factorization Framework* |
| PCA | *Principal Component Analysis* |
| PCT | *Predictive Clustering Trees* |
| ReLU | *Rectified Linear Function* |
| RNA | *Redes Neurais Artificiais* |
| RNN | *Recurrent Neural Network* |
| RPR | *Raw Pixel Representation* |
| SDA | *Stacked Denoising Autoencoder* |
| SGD | *Stochastic Gradient Descent* |
| SIFT | *Scale-Invariant Feature Transform* |
| JCR | *Journal Citation Reports* |
| SOM | *Self-Organizing Map* |
| SE | *Spectral Entropy* |

**SUMÁRIO**

**1 INTRODUÇÃO ..... 13**
1.1 DEFINIÇÃO DO PROBLEMA E MOTIVAÇÃO ..... 15
1.2 OBJETIVOS ..... 17
1.2.1 Objetivo Geral ..... 17
1.2.2 Objetivos Específicos ..... 17
1.3 ESTRUTURA DO TRABALHO ..... 17
**2 CLASSIFICAÇÃO DE DADOS ..... 19**
2.1 CONCEITOS DE CLASSIFICAÇÃO: UMA VISÃO GERAL ..... 19
2.2 CLASSIFICAÇÃO MULTIRRÓTULO ..... 21
2.3 CLASSIFICAÇÃO HIERÁRQUICA ..... 23
2.4 CLASSIFICAÇÃO HIERÁRQUICA MULTIRRÓTULO ..... 24
2.4.1 Abordagens de Classificação Hierárquica ..... 25
2.4.2 Algoritmos de Classificação Hierárquica Multirrótulo ..... 27
2.4.3 Medidas de Avaliação ..... 29
2.5 CONSIDERAÇÕES FINAIS DO CAPÍTULO ..... 30
**3 REDUÇÃO DE DIMENSIONALIDADE ..... 31**
3.1 CONCEITOS DE REDUÇÃO DE DIMENSIONALIDADE ..... 31
3.2 TÉCNICAS DE REDUÇÃO DE DIMENSIONALIDADE ..... 32
3.2.1 Seleção de Atributos ..... 32
3.2.2 Extração de Atributos ..... 35
3.3 CONSIDERAÇÕES FINAIS DO CAPÍTULO ..... 37
**4 REDES NEURAIS ARTIFICIAIS ..... 39**
4.1 CONCEITOS FUNDAMENTAIS SOBRE REDES NEURAIS ARTIFICIAIS ..... 39
4.2 APRENDIZAGEM PROFUNDA ..... 43
4.3 AUTOENCODERS ..... 46
4.3.1 Processo de Treinamento de um Autoencoder ..... 51
4.4 CONSIDERAÇÕES FINAIS DO CAPÍTULO ..... 58
**5 REVISÃO SISTEMÁTICA DA LITERATURA ..... 60**
5.1 DESCRIÇÃO DO MÉTODO DE MAPEAMENTO SISTEMÁTICO ..... 60
5.2 QUESTÃO DE PESQUISA ..... 62
5.3 SELEÇÃO DAS BASES E DEFINIÇÃO DOS TERMOS DE BUSCA ..... 62
5.4 REALIZAÇÃO DAS BUSCAS ..... 64
5.5 PROCEDIMENTOS DE FILTRAGEM ..... 64
5.6 ORDENAÇÃO DOS ARTIGOS ..... 65
5.7 RESULTADOS ..... 67
5.8 CONSIDERAÇÕES DO CAPÍTULO ..... 74
**6 EXTRAÇÃO DE ATRIBUTOS BASEADA EM AUTOENCODERS PARA A CLASSIFICAÇÃO HIERÁRQUICA MULTIRRÓTULO (FEAE-HMC) ..... 75**
6.1 VISÃO GERAL ..... 75
6.2 FUNCIONAMENTO ..... 77

6.2.1 Estratégia para Criação da Estrutura da rede FEAE-HMC ..... 78
6.2.2 Extração de Características - FEAE-HMC ..... 83
6.3 CONSIDERAÇÕES FINAIS DO CAPÍTULO ..... 85
**7 EXPERIMENTOS E RESULTADOS ..... 86**
7.1 FERRAMENTAS UTILIZADAS ..... 86
7.2 BASE DE DADOS ..... 86
7.3 METODOLOGIA UTILIZADA NOS EXPERIMENTOS ..... 87
7.4 EXTRAÇÃO DE ATRIBUTOS ..... 89
7.5 CLASSIFICAÇÃO HIERÁRQUICA ..... 91
7.6 RESULTADOS EXPERIMENTAIS ..... 91
7.6.1 Resultados - Clus HMC ..... 92
7.6.2 Resultados - MHC-CNN ..... 98
7.6.3 Comparação Geral dos Resultados Obtidos ..... 99
7.7 CONSIDERAÇÕES FINAIS DO CAPÍTULO ..... 102
**8 CONCLUSÃO ..... 103**
8.1 TRABALHOS FUTUROS ..... 104
**REFERÊNCIAS ..... 105**
**APÊNDICE A - RESULTADOS DOS EXPERIMENTOS NO CLUS-HMC ..... 111**

# 1 INTRODUÇÃO

A Aprendizagem de Máquina é um campo de pesquisa da Inteligência Artificial que estuda o desenvolvimento de métodos computacionais capazes de aprender (CARVALHO et al., 2011). O aprendizado acontece por meio do treinamento utilizando grandes quantidades de dados que possibilitam ao algoritmo o ajuste e, consequentemente, melhora de desempenho. Diversos algoritmos de aprendizagem de máquina, por exemplo, são utilizados para gerar classificadores para conjunto de dados.

Em problemas de classificação, cada amostra possui um conjunto de atributos chamados preditores e um atributo supervisor (classe). Um classificador deve ser capaz de prever a classe de quaisquer novos dados do domínio em que ele foi treinado. O modelo construído para a classificação baseia-se na análise prévia de um conjunto de dados de amostragem ou treinamento, em que cada amostra está corretamente classificada (CAMARGO, 2012).

No que se refere a predição da quantidade de classes pode-se citar dois tipos de classificação: monorrótulo ou multirrótulo (CARVALHO et al., 2011). Na classificação monorrótulo, os classificadores associam uma única classe ou rótulo a cada exemplo. Já na chamada classificação multirrótulo, cada exemplo pode pertencer simultaneamente a mais de uma classe, assim o objetivo de um classificador multirrótulo é classificar instâncias automaticamente a um conjunto de classes de um domínio específico.

As soluções propostas na classificação monorrótulo e multirrótulo são voltadas para problemas de classificação plana (tradicional), na qual as classes disponíveis estão todas em um mesmo nível, ou seja, não existe dependência entre as classes envolvidas. Porém, há muitos problemas mais complexos, denominados problemas de classificação hierárquica, nos quais as classes a serem preditas estão dispostas em uma estrutura hierárquica, em que uma ou mais classes podem ser divididas em subclasses e superclasses.

Observa-se ainda a existência problemas de classificação hierárquica em que duas ou mais classes disponíveis podem ser atribuídas simultaneamente à mesma instância, denominados como problemas de classificação hierárquica multirrótulo.

Na Classificação Hierárquica Multirrótulo, (*Hierarchical Multi-label Classification* - HMC), os exemplos são associados a múltiplas classes, mas nesse caso, as classes do domínio descrevem uma estrutura hierárquica com diversos níveis de especificidade e regras bem definidas que governam as relações entre as diferentes classes presentes na hierarquia (METZ, 2011).

Uma característica da classificação hierárquica multirrótulo é que ao associar uma classe da hierarquia para uma instância, automaticamente associará todas as suas classes ancestrais. Entre alguns exemplos de utilização da classificação hierárquica multirrótulo pode-se citar a predição de: proteínas em dados de bioinformática, de textos, de gêneros musicais entre outros (BORGES, 2012).

Os classificadores hierárquicos podem fazer uso de duas abordagens principais: Classificador Local que avalia cada nó do conjunto de dados hierárquico independentemente e o Classificador Global que leva em consideração toda a hierarquia de classes do conjunto ao predizer uma instância (BORGES, 2012). A classificação hierárquica para estruturas do tipo Grafo Acíclico Dirigido, do inglês *Directed Acyclic Graph* (DAG), é complexa, pois toda a estrutura hierárquica é respeitada.

Em problemas de classificação hierárquica multirrótulo a quantidade de exemplos, atributos e classes influência no desempenho dos métodos de classificação, pois os dados, geralmente, possuem poucas amostras para treinamento, alta dimensionalidade de atributos e uma variação no número de classes a qual cada amostra pertence, fazendo com que os dados não sejam igualmente multirrótulo (o número de classes que cada instância pode estar associada não é uniforme).

Um dos grandes desafios do aprendizado de máquina é o desenvolvimento de algoritmos que consigam identificar mudanças na evolução de dados, quando estes são representados em espaços de alta dimensão. De acordo com Kira e Rendell (1992), a quantidade excessiva de atributos prejudica a busca por padrões e extração de conhecimento útil, pois afeta o desempenho dos algoritmos de aprendizagem tanto no custo computacional quanto na performance do classificador.

Nessa perspectiva, torna-se necessário utilizar métodos de redução de dimensionalidade, por meio da representação dos dados originais em um espaço de menor dimensão, para identificar quais são os atributos mais relevantes em uma base de dados, sendo que este processo tem considerável importância no cenário computacional (ROMDHANE; FADHEL; AYEB, 2010). Entretanto, os métodos e técnicas que buscam a representação reduzida da dimensionalidade intrínseca de um determinado conjunto de dados, são de grande variedade, sendo assim a escolha do método apropriado não é trivial.

Como alternativa, a aprendizagem profunda (*Deep Learning*), é uma subcategoria do aprendizado de máquina, que surgiu inspirada em estudos biológicos do funcionamento do cérebro humano, com o objetivo de tratar a dificuldade que as Redes Neurais Artificiais (RNA) possuem com dados de alta dimensionalidade (HAYKIN, 2009).

Um *Autoencoder* é uma rede neural que pode ser utilizada dentro do contexto da aprendizagem profunda (*Deep Learing*) para reduzir a dimensão dos dados. Tem como objetivo aprender uma função para aproximar os dados de entrada em si, de modo que $f(x) = x$. A estrutura de um *Autoencoder* é composta necessariamente por uma estrutura codificadora e uma decodificadora. Enquanto a codificadora pretende comprimir os dados de entrada originais em uma representação de baixa dimensão, a decodificadora tenta reconstruir os dados de entrada originais com base na representação de baixa dimensão gerada pela estrutura codificadora (BENGIO, 2009).

Dessa maneira, verifica-se nesse trabalho a possibilidade de aplicar a redução de dimensionalidade em dados de classificação hierárquica multirrótulo, que possuem a hierarquia de classes estruturadas no formato de uma DAG, utilizando *Autoencoders*. Para tanto, apresenta-se um método de extração de atributos utilizando adaptações em uma rede *Autoencoder* clássica, para que a nova estrutura possa ser capaz de capturar correlações entre os atributos e os rótulos de cada amostra.

Nos experimentos foram utilizados dados biológicos de 10 bases de dados da Ontologia Gênica, que tem a hierarquia das classes estruturada como um DAG. As representações de menor dimensão fornecidas para o espaço original dos atributos, agregam correlações entre os atributos e rótulos, e quando submetidas a um Classificador Hierárquico Multirrótulo, geram um modelo, onde-se obtêm o desempenho preditivo para a tarefa de HMC equivalente e até mesmo superior ao desempenho da base original.

## 1.1 DEFINIÇÃO DO PROBLEMA E MOTIVAÇÃO

Um dos problemas enfrentados pelas tarefas de classificação é quando os dados apresentam grande volume de atributos, sendo que muitos destes são redundantes e prejudicam a performance do algoritmo de aprendizagem tanto na velocidade de execução, quanto na taxa de acerto (WITTEN et al., 2016).

A redução de dimensionalidade em dados para tarefa de classificação, normalmente, busca minimizar a taxa de erro do classificador, eliminando atributos irrelevantes por meio da seleção e/ou composição de atributos mais adequados, além de simplificar o modelo gerado e facilitar a interpretação e a visualização dos dados.

Dentre as técnicas que tem o intuito de reduzir a dimensionalidade das bases de dados, pode-se citar a extração de atributos, que consiste em extrair, a partir dos dados brutos,

características de alto nível com grande riqueza de informação relevante sobre os dados originais, ou seja, criam-se novas características que sejam mais expressivas e que melhor representem a variabilidade dos dados (CAMPOS, 2001).

Em um *Autoencoder*, as saídas são treinadas para igualar as entradas, sendo que as camadas de entrada e saída tem o mesmo tamanho $n$, que equivale ao número de atributos/dimensão de entrada. Entre essas camadas existem $k$ camadas ocultas, sendo que a camada oculta central (ou gargalo) fornece os dados em forma comprimida com dimensão de saída igual ao número $m$ dos neurônios dessa camada, em que $m < n$ (BENGIO, 2009).

Para Medeiros e Costa (2008), é justamente a possibilidade de extrair os sinais de saída da camada gargalo que permite a utilização da rede *Autoencoder* como um método de redução de dimensionalidade baseado em extração de características para um conjunto de dados com alta dimensão.

Técnicas de aprendizagem profunda relataram resultados encorajadores em uma variedade de aplicações, incluindo reconhecimento de fala (YU et al., 2013; HINTON et al., 2012), reconhecimento facial (LI; GUO, 2014 ; PANG; WANG; YUAN, 2014); reconhecimento de objetos (KRIZHEVSKY; SUTSKEVER; HINTON, 2012; ZEILER; FERGUS, 2014) processamento de linguagem natural (GIRSHICK et al., 2014; MIKOLOV; YIH; ZWEIG, 2013), entre outros (WAN et al., 2014).

Inspirado nos sucessos da aplicação e nas possibilidades de utilização da aprendizagem profunda no meio computacional e que, especificamente, os *Autoencoders* têm potencial para alcançar a redução de dimensionalidade de dados, busca-se aplicá-los na redução da dimensionalidade em dados hierárquicos multirrótulos, por meio da extração de um conjunto reduzido de características da base de dados, que seja representativo e significativo a ponto de ser utilizado posteriormente em um Classificador Hierárquico Multirrótulo Global.

No que se refere a redução de dimensionalidade em problemas de classificação hierárquica multirrótulo, com base na revisão de literatura apresentada no Capítulo 5, métodos de extração de atributos são utilizados apenas no pré-processamento de alguns classificadores, principalmente na área de processamento de imagens. A apresentação de novos métodos de redução de dimensionalidade para a área de HMC, proposta deste trabalho, ocorre apenas em 3 trabalhos da revisão sistemática apresentada, e esses utilizam o método de seleção de atributos.

Outro desafio observado, é levar em consideração a estrutura hierárquica do tipo DAG, sendo que não foram observadas técnicas de extração de atributos que levem em consideração esse tipo de estrutura hierárquica na HMC, sendo esse o requisito fundamental para o tipo de

base de dados da Ontologia Gênica (Gene Ontology - GO), as quais tem a dimensionalidade dos atributos reduzidas nesse trabalho e posteriormente são avaliadas na HMC.

Dessa maneira, motivado pela possibilidade de aplicar a extração de atributos, utilizando *Autoencoders*, apresenta-se um método de extração que se mostra capaz de extrair representações de menor dimensão, que agregam correlações entre os atributos e rótulos, nas quais se obtem desempenho preditivo na tarefa de HMC equivalente e até mesmo superior ao desempenho preditivo da base original.

## 1.2 OBJETIVOS

Esta Seção apresenta o objetivo geral e os específicos deste trabalho.

### 1.2.1 Objetivo Geral

Desenvolver um método de redução de dimensionalidade, utilizando a extração de atributos e *Autoencoders* para a classificação hierárquica multirrótulo.

### 1.2.2 Objetivos Específicos

Como objetivos específicos deste trabalho definiu-se:

- Realizar uma revisão sistemática da literatura para identificar técnicas de redução de dimensionalidade para classificação hierárquica multirrótulo;
- Utilizar a técnica de extração de atributos por meio de *Autoencoders* para classificação hierárquica multirrótulo;
- Analisar o desempenho dos conjuntos de dados com todos os atributos e dos conjuntos com dimensionalidade reduzida gerados pelo método proposto em um classificador hierárquico multirrótulo;
- Avaliar os resultados obtidos por meio de métricas e testes estatísticos.

## 1.3 ESTRUTURA DO TRABALHO

Este trabalho está organizado em oito Capítulos, do seguinte modo:

- **Capítulo 1:** O Capítulo introdutório contextualiza o trabalho, apresentando o problema estudado, os objetivos e a motivação para sua realização.
- **Capítulo 2:** Este Capítulo aborda os principais conceitos e técnicas referentes à classificação de dados, que proporcionam o embasamento teórico, bem como a contextualização em que este trabalho se insere.
- **Capítulo 3:** O Capítulo apresenta os conceitos, abordagens e principais técnicas de redução de dimensionalidade em bases de dados.
- **Capítulo 4:** Neste Capítulo são apresentados os conceitos fundamentais sobre redes neurais, aprendizagem profunda bem como a contextualização dos *Autoencoders* que serão utilizadas nesse trabalho.
- **Capítulo 5:** O Capítulo aborda os trabalhos correlatos, bem como suas aplicações, que fundamentam o tema proposto.
- **Capítulo 6:** O Capítulo apresenta o método proposto, FEAE-HMC, para o problema de extração de atributos em HMC, por meio do seu detalhamento em uma base fictícia para melhor entendimento da estratégia e do seu funcionamento.
- **Capítulo 7:** Este Capítulo contempla a descrição das bases de dados utilizadas, a metodologia aplicada no desenvolvimento dos experimentos, assim como a análise dos resultados obtidos.
- **Capítulo 8:** Esse Capítulo relata as conclusões obtidas com os experimentos e resultados do método proposto FEAE-HMC. Além disso, destacam-se as limitações do método e os trabalhos futuros.

# 2 CLASSIFICAÇÃO DE DADOS

Este Capítulo apresenta conceitos básicos sobre a classificação de dados. A Seção 2.1 descreve os principais conceitos de classificação. A Seção 2.2 aborda os conceitos relacionados à classificação multirrótulo. A Seção 2.3 descreve conceitos fundamentais e as abordagens de problemas de classificação hierárquica. Por fim, a Seção 2.4 apresenta as considerações finais do Capítulo.

## 2.1 CONCEITOS DE CLASSIFICAÇÃO: UMA VISÃO GERAL

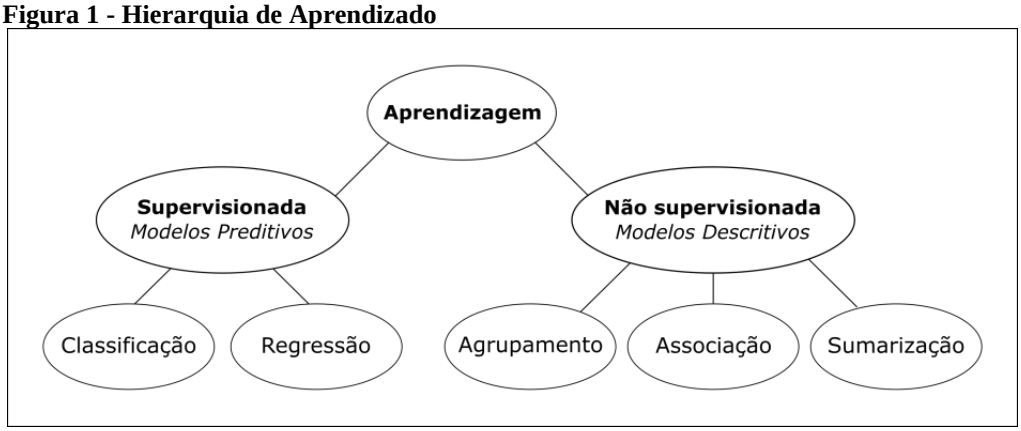
Tarefas de aprendizado de máquina podem ser divididas em preditivas e descritivas. Nas tarefas preditivas, o objetivo é obter uma função ou modelo por meio de dados de treinamento, sendo que esta função ou modelo deverá ser capaz de prever ou caracterizar um novo exemplo com base nos valores de seus atributos. Algoritmos utilizados nessa tarefa utilizam o paradigma de aprendizado supervisionado, no qual se conhece o rótulo desejado para cada exemplo (CARVALHO et al., 2011).

Algoritmos de aprendizado supervisionado têm seus modelos treinados usando um conjunto de valores dos atributos de diversos exemplos rotulados. Métodos como a classificação, regressão e previsão usam padrões para prever os valores do rótulo em dados adicionais não rotulados com base nos conjuntos de valores dos atributos de entrada.

Nas tarefas descritivas, a meta é explorar ou descrever um conjunto de dados, já que não existe ou se utiliza o atributo de saída (rótulo). Algoritmos de tarefas descritivas utilizam o paradigma do aprendizado não supervisionado, em que não se sabe a “resposta certa”, busca-se apenas explorar os dados e encontrar alguma estrutura relevante, ou seja, encontrar grupos de objetos semelhantes em um conjunto de dados (CARVALHO et al., 2011). A Figura 1 exibe a hierarquia de aprendizado, processo pelo qual são realizadas generalizações dos dados.

Tarefas supervisionadas se diferenciam pelo tipo do rótulo, na classificação trata-se de um atributo discreto e na regressão um atributo contínuo. A tarefa não supervisionada de agrupamento consiste em agrupar os dados de acordo com a similaridade; a sumarização busca encontrar uma descrição simples do conjunto de dados e a associação tem como objetivo encontrar padrões frequentes de associações entre atributos de um conjunto de dados (CARVALHO et al., 2011).

**Figura 1 - Hierarquia de Aprendizado**




**Fonte: Adaptado de Carvalho et al. (2011)**

Na classificação cada registro pertence a uma classe pré-definida e possui um conjunto de atributos preditores e um atributo classe. Na classificação o modelo analisa o conjunto de registros fornecidos e a indicação à qual classe pertence, a fim de 'aprender' como classificar um novo registro (BORGES; NIEVOLA, 2006). O objetivo dessa tarefa é descobrir um relacionamento entre os atributos previsores e o de classe, criar um modelo ou função para que posteriormente sejam aplicados aos atributos previsores para prever a classe ainda desconhecida de um registro.

Os exemplos disponíveis para geração de um modelo em um classificador são divididos em dois conjuntos mutuamente exclusivos: um de treinamento e um de teste. O conjunto de treinamento fica disponível para o classificador analisar as relações entre os atributos previsores e a classe. Os relacionamentos descobertos, a partir desses exemplos, são utilizados para prever a classe dos registros presentes no conjunto de teste (BORGES; NIEVOLA, 2006).

Para o classificador, o atributo meta do conjunto de teste fica indisponível. Após prever a classe dos exemplos do conjunto de teste, as classes previstas são comparadas com as classes reais dos exemplos. Se a classe prevista for igual a real, a previsão foi correta, caso contrário, incorreta. Um dos principais objetivos na tarefa de classificação é maximizar a taxa de classificações corretas nos dados de teste, que corresponde à razão entre o número de exemplos corretamente classificados e o número total de exemplos disponíveis no conjunto de testes.

Quando um exemplo pode pertencer simultaneamente a mais de uma classe, existe um problema de Classificação Multirrótulo, assunto abordado na próxima subseção.

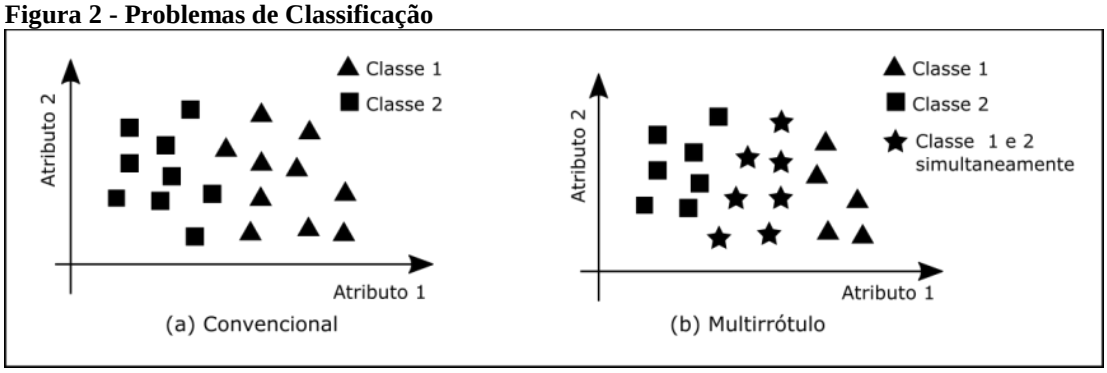
## 2.2 CLASSIFICAÇÃO MULTIRRÓTULO

Os classificadores convencionais utilizados na Aprendizagem de Máquina associam uma única classe ou rótulo a cada exemplo. Já na chamada classificação multirrótulo cada exemplo pode pertencer a mais de uma classe (CARVALHO et al., 2011).

Problemas de classificação multirrótulo são encontrados por exemplo na classificação de documentos (associar um documento a mais de uma categoria), classificação semântica de imagens (uma mesma imagem pode possuir vários contextos semânticos), categorização musical (uma música pode pertencer a mais de um gênero musical), classificação da função de proteínas (uma proteína pode ter uma ou mais funções simultaneamente associadas a ela), entre outros (CARVALHO et al., 2011).

Em um problema de classificação convencional, cada exemplo pertence somente a uma das classes possíveis. Já em um problema de classificação multirrótulo, alguns dos exemplos pertencem simultaneamente a mais de uma classe (CARVALHO et al., 2011). A Figura 2 demonstra os diferentes problemas de classificação.

**Figura 2 - Problemas de Classificação**




**Fonte: Carvalho et al. (2011)**

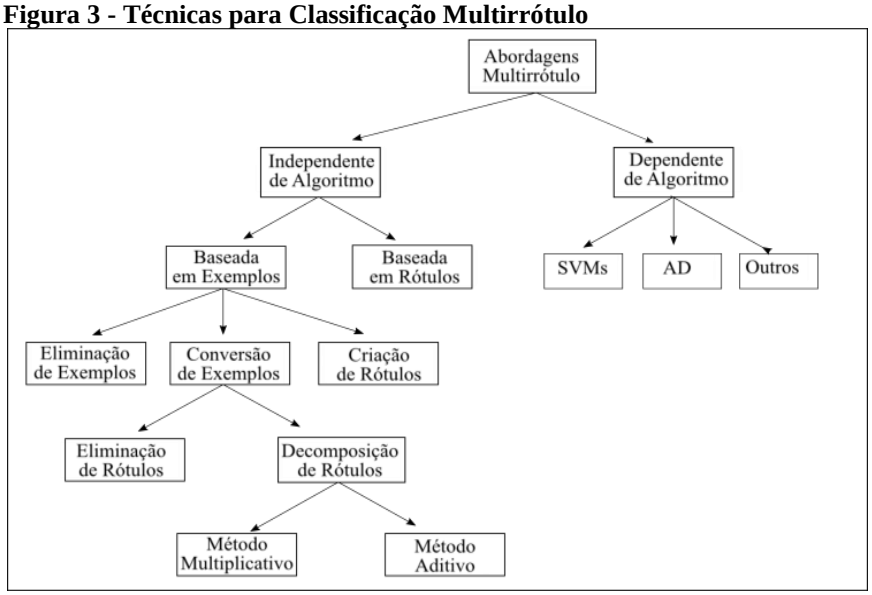
A Figura 2(a) mostra, por exemplo, um problema de Classificação de documentos, no qual cada documento pode ser classificado somente com uma das classes disponíveis (ex: Esporte ou Política). Já na Figura 2(b), relata um cenário em que alguns documentos podem ser classificados com ambas as classes simultaneamente (Esporte e Política).

Na literatura são encontrados três métodos para resolver problemas de Classificação Multirrótulo. O primeiro baseia-se na combinação de classificadores convencionais, no segundo, os classificadores convencionais são modificados e adaptados para serem utilizados em problemas multirrótulo, e o último, no qual algoritmos são desenvolvidos especificamente para este fim (CERRI, 2010).

Os métodos para classificação multirrótulo podem ser divididos em duas abordagens principais: a dependente e a independente do algoritmo. A abordagem independente de algoritmo transforma um problema multirrótulo em um problema simples-rótulo (convencional). Essa transformação pode ser baseada nos rótulos das classes ou nos exemplos (instâncias) disponíveis (CARVALHO et al., 2011).

A abordagem dependente de algoritmo se baseia na criação de novos algoritmos específicos para o problema multirrótulo a ser investigado, sendo que assim, este poderá obter resultados melhores que a abordagem independente de algoritmo. A Figura 3 ilustra uma visão geral das diferentes técnicas propostas na literatura para tratar problemas de classificação multirrótulo.

**Figura 3 - Técnicas para Classificação Multirrótulo**




**Fonte: Freitas e Carvalho (2007)**

Outra característica de problemas multirrótulo é que a quantidade de exemplos e classes influência no desempenho dos métodos de classificação. Cada exemplo pode estar vinculado a diferentes quantidades de classes simultaneamente, e a quantidade de vezes que essa condição vai se repetir em outros exemplos, pode ser insuficiente para garantir o correto aprendizado do classificador. Assim, são utilizados os conceitos de cardinalidade e densidade e algumas medidas de desempenho para a avaliação de classificadores multirrótulo (CARVALHO et al., 2011).

A avaliação de classificadores multirrótulo requer medidas diferentes das utilizadas em problemas de classificação monorrótulo, no qual um exemplo é classificado de maneira errada ou correta. Na classificação multirrótulo, um exemplo pode ser classificado de maneira parcialmente errada ou parcialmente correta. Assim, um classificador pode atribuir

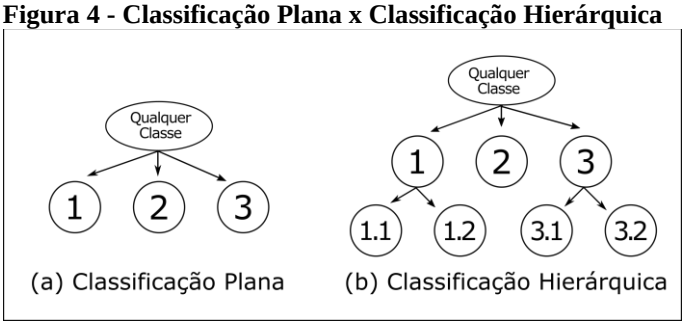
corretamente a um exemplo pelo menos uma das classes a que ele pertence, deixando de atribuir uma ou mais classes que ele pertence. Ao mesmo passo que o classificador também pode atribuir a um exemplo uma ou mais classes as quais ele não pertence (CERRI, 2010).

Quando as classes a serem preditas (monorrótulo ou multirrótulo), podem ser divididas em subclasses ou agrupadas em superclasses, não se observando total independência entre as classes, e estas estão dispostas em uma estrutura hierárquica, existe um problema de Classificação Hierárquica.

## 2.3 CLASSIFICAÇÃO HIERÁRQUICA

Na classificação tradicional, nenhuma ordem é imposta para as classes, e cada exemplo é associado a uma única classe pertencente a um conjunto finito de classes. Porém, os esforços de investigação são voltados para a tarefa de Classificação Hierárquica, em que os exemplos devem ser associados a classes organizadas em uma hierarquia de classes pré-definidas, como pode ser observado na Figura 4.

**Figura 4 - Classificação Plana x Classificação Hierárquica**



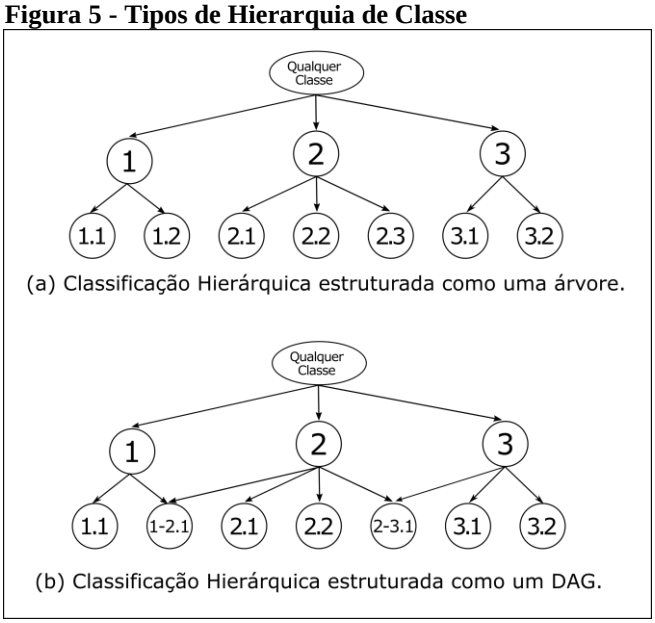

**Fonte: Freitas e Carvalho (2007)**

A classificação hierárquica caracteriza-se pela existência de uma relação de taxonomia e dependências entre as classes. Observa-se que na classificação plana, as classes estão organizadas horizontalmente, indicando que não há dependências entre elas. Já na Classificação Hierárquica, as classes 1.1 e 1.2 estão diretamente ligadas a classe 1, indicando uma relação de dependência. Assim, ao se considerar a existência dessa hierarquia, é possível realizar a indução de classificadores com maior acurácia preditiva (CARVALHO et al., 2011).

Na classificação hierárquica, quanto mais profundo é o nível, mais difícil é a predição. Isso ocorre devido ao fato que a quantidade de exemplos em níveis mais profundos tende a ser menor, o que leva a modelos mais específicos, que por sua vez, são modelos com confiabilidade menor. Por outro lado, quanto mais profunda a classe predita na hierarquia, mais específico e útil será o conhecimento descoberto (FREITAS; CARVALHO, 2007).

Problemas de classificação hierárquica podem ser diferenciados observando a organização entre as classes representadas na hierarquia e como ocorrem os relacionamentos entre elas, podendo ser uma hierarquia do tipo de árvore ou de um *Directed Acyclic Graph* (DAG), conforme observado na Figura 5.

**Figura 5 - Tipos de Hierarquia de Classe**




**Fonte: Adaptado de Freitas e Carvalho (2007)**

Os nós representam as classes envolvidas no problema. A Classificação Hierárquica tem por objetivo a classificação de cada novo padrão de entrada em qualquer nó ou em um dos nós folha. A principal diferença entre a estrutura em árvore e a estrutura DAG é que na estrutura em árvore (Figura 5-a) cada nó tem somente um nó pai, enquanto na DAG (Figura 5-b) cada nó pode ter um ou mais nós pai (FREITAS; CARVALHO, 2007).

## 2.4 CLASSIFICAÇÃO HIERÁRQUICA MULTIRRÓTULO

Problemas de classificação que possuem características tanto dos problemas de classificação multirrótulo, quanto de problemas de classificação hierárquica, são problemas de Classificação Hierárquica Multirrótulo, na qual uma instância pode pertencer a múltiplas classes ao mesmo tempo e essas classes estão organizadas de maneira hierárquica (CERRI, 2010).

Como exemplo, pode-se citar que na bioinformática as funções exercidas por uma proteína podem ser categorizadas hierarquicamente, na qual uma ou mais classes podem ser

divididas em subclasses ou agrupadas em superclasses. Nesse caso, as classes estão dispostas em uma estrutura hierárquica, tal como uma árvore ou um como um DAG.

Assim, esse tipo de problema é comum, principalmente em categorização de textos, predição de proteínas em dados de bioinformática, classificação de gêneros musicais e imagens, entre outros. Então, a classificação hierárquica multirrótulo aborda problemas que são mais complexos em relação aos demais encontrados na tarefa de classificação, uma vez que as classes envolvidas no problema, estão estruturadas em uma hierarquia e os exemplos podem pertencer a mais de uma classe ao mesmo tempo (CERRI, 2010).

A análise de um classificador, em uma hierarquia estruturada do tipo DAG, é um pouco mais complexa, tendo em vista que um nó-filho pode ter mais de um nó pai, e classes em níveis profundos podem ter mais exemplos do que classes de níveis anteriores. Essa situação faz com que a taxa de acerto tenda a diminuir com o aumento da profundidade. Assim, se o classificador não apresentar uma confiabilidade desejada na classificação em uma das classes do nível mais profundo, pode ser mais eficiente e vantajoso realizar a classificação em níveis mais elevados (FREITAS; CARVALHO, 2007).

Nessa perspectiva, em alguns problemas, todos os exemplos devem ter um nó-folha que o representem, são os chamados “problemas de predição obrigatória em nós folha”. Em contraste, problemas onde não se aplica esta obrigatoriedade são chamados de “problemas de predição opcional em nós folha”, nos quais um exemplo pode ser representado por um nó interno da hierarquia (COSTA, 2008).

Estas condições influenciam diretamente no nível hierárquico em que ocorrerá a predição de classes, o que é uma característica útil a ser analisada conforme o grau de confiabilidade desejado para o modelo criado.

### 2.4.1 Abordagens de Classificação Hierárquica

Para Freitas e Carvalho (2007), as soluções propostas para a indução de modelos de classificação para problemas hierárquicos multirrótulo podem ser agrupadas em quatro tipos de abordagens: transformação do problema hierárquico em um problema de classificação plana, predição hierárquica utilizando algoritmos de classificação plana, classificação local e classificação global. A seguir os principais conceitos relacionados a estas abordagens serão apresentados.

- Transformação em um problema de Classificação Plana

Esta abordagem opta por transformar o problema de Classificação Hierárquica em Classificação Plana, considerando este como um caso específico de Classificação Hierárquica em que não são consideradas as subclasses e superclasses. Dessa maneira, o problema pode ser tratado com as abordagens tradicionais de classificação plana.

Uma possível transformação é considerar apenas o primeiro nível da hierarquia para realizar a predição em Classificação Plana, mas existe uma desvantagem de não se explorar os níveis mais profundos, em que as classes representam informações mais específicas. Assim, uma alternativa consiste em realizar a predição em um nível mais específico, o que naturalmente refletirá nos níveis superiores. Por exemplo, se um novo dado de entrada é classificado como pertencente à classe 1.1 de um modelo, ele também é automaticamente predito como pertencente à classe 1 (COSTA, 2008).

Entretanto, ao considerar todas as classes de um nível mais específico, sem explorar as informações contidas nos níveis mais gerais, a classificação tende a ser mais difícil.

- Predição Hierárquica utilizando Algoritmos de Classificação Plana

Esta abordagem divide o problema de Classificação Hierárquica em um conjunto de problemas de Classificação Plana. Uma possível divisão é considerar cada nível da hierarquia como um problema de classificação plana independente dos outros. Dessa maneira em cada nível um classificador é induzido por meio de um algoritmo de Classificação Plana (COSTA, 2008).

Uma desvantagem está na possibilidade de ocorrerem inconsistências entre os classificadores, ou seja, a saída de dois classificadores ou mais não são coerentes entre si.

- Classificação Hierárquica Local

Essa abordagem consiste em treinar um ou mais classificadores para cada nó da hierarquia, obtendo uma árvore de classificadores, nos quais os exemplos de teste são classificados de maneira iterativa e *Top-Down* (do menos específico para o mais específico) ao longo dos níveis da hierarquia (METZ, 2011).

Quando um exemplo é classificado como pertencente a uma classe, ele é aplicado a um classificador para predizer a qual das subclasses da classe predita no nível anterior ele

pertence. Esse procedimento é repetido até se encontrar uma classe representada por um nó-folha, em caso de predição obrigatória em nós-folha, ou até que se associe o exemplo a um nó-interno da hierarquia, em caso de predição opcional em nós-folha (COSTA, 2008).

Um classificador só é gerado para os nós em que haja a necessidade de um processo de decisão, ou seja, para classes representadas por nós internos que possuem apenas uma subclasse e por nós-folha onde não existe a necessidade de criação de classificadores. No entanto, possui a desvantagem que erros cometidos nos níveis superiores sejam propagados para os mais específicos.

- Classificação Hierárquica Global

Esta abordagem apresenta uma maior complexidade, pois o modelo de classificação é criado considerando a hierarquia de classes como um todo. Desse modo, após a fase de treinamento do modelo de classificação, a predição da classe de um exemplo é feita em um único passo utilizando informações de toda a hierarquia.

Ao contrário das demais abordagens, na abordagem Global (*Big-Bang)*, técnicas de classificação plana não podem ser aplicadas sem que modificações sejam efetuadas no modo como a indução do classificador é realizada, para que o modelo gerado considere toda a hierarquia. Dessa forma, a construção de um algoritmo global é mais complexa quando comparada a um algoritmo local, que realiza a classificação por meio de um processo de modularização (COSTA, 2008). Se a predição for obrigatória para nós folha, a predição é feita para um desses nós e as predições nos níveis superiores são derivadas dessa predição.

### 2.4.2 Algoritmos de Classificação Hierárquica Multirrótulo

Na literatura, há vários trabalhos propondo métodos para tarefas de classificação hierárquica multirrótulo em diferentes domínios como bioinformática (MELO; PAULHEIM, 2019), classificação de texto (GARGIULO et al., 2019), processamento de imagem (DIMITROVSKI et al., 2011), processamento de vídeo e áudio (WANG et al., 2017), entre outros. Contudo, não há um consenso sobre que abordagem utilizar para o tratamento de problemas hierárquicos multirrótulo.

Nesse trabalho, optou-se Clus-HMC e MHC-CNN, para tanto uma breve descrição é apresentada sobre esses classificadores.

#### 2.4.2.1 Clus-HMC

Proposto em Blockeel et al. (2002) e posteriormente explorado em Vens et al. (2008), o Clus-HMC é um método baseado em *Predictive Cluster Trees* (PCT) capaz de gerar uma única Árvore de Decisão (*Decision Tree*) considerando toda a hierarquia de classes de um problema.

No Clus-HMC as árvores de decisão são vistas como uma hierarquia de grupos, que predizem um conjunto de classes. O nó raiz contém todas as instâncias de treinamento, que são recursivamente divididas em grupos menores conforme a hierarquia é percorrida em direção as folhas. A classificação de uma instância ocorre segundo uma métrica baseada em distância, que calcula o quão similar a instância é a alguma árvore. Ao final, os grupos formados são combinados a fim de que seja induzida uma única árvore de decisão. A saída do Clus-HMC é um vetor binário correspondente as possíveis classes de uma instância (BLOCKEEL et al., 2002).

Assim o algoritmo Clus-HMC, pode ser definido como um algoritmo que constrói classificadores baseados em árvores de decisão, pode ser aplicado tanto para classificação quanto para agrupamento, e vem sendo investigado e melhorado ao longo dos anos. A versão mais utilizada, gera classificadores globais para classificação hierárquica multirrótulo, e possui a capacidade de lidar com estruturas hierárquicas na forma de árvores e DAGs. Deste modo, cada nó folha prevê uma ou mais classes (VENS *et al.*,2008).

#### 2.4.2.2 MHC-CNN

Proposto por Borges (2012), o MHC-CNN (*Multi-label Hierarchical Classification using a Competitive Neural Network*) é um classificador hierárquico multirrótulo que utiliza a abordagem de classificação global baseada em uma Rede Neural Artificial Competitiva.

A rede competitiva é composta por uma camada de entrada e uma de saída. A topologia da camada de saída tem a estrutura de um grafo acíclico dirigido, em que cada neurônio está conectado com seus neurônios ancestrais (pais) e descendentes (filhos) e a todos os neurônios da camada de entrada (BORGES, 2012).

O treinamento é realizado via cálculo das distâncias entre os nós da hierarquia e as instâncias de treino. Os neurônios da camada de saída competem entre si para serem ativados. Dessa forma, os neurônios que apresentam as menores distâncias são considerados vencedores, e terão seus pesos atualizados influenciando suas vizinhanças (BORGES, 2012).

### 2.4.3 Medidas de Avaliação

Os trabalhos apresentados na literatura de classificação hierárquica multirrótulo demonstram o desempenho da classificação, por meio de algumas medidas de avaliação definidas para essa área. Porém, não há um consenso de qual medida deve ser adotada como regra para essa avaliação (CERRI, 2010).

Entre as medidas utilizadas na literatura especificamente para a avaliação de classificadores hierárquicos multirrótulo, pode-se se citar as medidas: baseadas nas relações de ancestralidade e descendência; baseadas em distância; baseadas em similaridade; baseadas em semântica; baseada em matriz de custos; baseadas na curva de precisão e revocação (CERRI, 2010; BORGES, 2012).

Nesse trabalho, optou-se pela medida baseada na curva de precisão e revocação (AUPRC), disponíveis nos algoritmos Clus-HMC e MHC-CNN. Para tanto, apresenta-se uma breve descrição sobre essa medida.

- Medida baseada na curva de precisão e revocação

Vens et. al. (2008) propôs uma medida baseada na análise de curvas de precisão e revocação (curvas PR). Nessa medida é escolhido um conjunto de limiares entre 0 e 1, sendo que cada limiar corresponde a um ponto no espaço da curva PR e variando esses limiares obtêm-se a curva PR. Esses limiares podem ser interpretados como a probabilidade de associação de dada classe a uma instância.

Por meio desses vetores é possível aplicar diversos limiares que resultarão em diferentes valores de precisão e revocação (um para cada limiar), em outras palavras, pontos dentro do espaço precisão-revocação (PR). Com a união desses pontos se obtém a curva PR, o que possibilita o cálculo da área sob a curva (AUC) (DAVIS; GOADRICH, 2006)

Para um cálculo mais aproximado da área da curva, é necessário a utilização de algum método de interpolação, como por exemplo, o método de interpolação não-linear (DAVIS; GOADRICH, 2006). Para um determinado limiar, um ponto de precisão e revocação no espaço da curva PR por meio das Equações (1) e da Equação (2), respectivamente.

$$\overline{Prec} = \frac{\sum_i VP_i}{\sum_i VP_i + \sum_i FP_i} \quad (1)$$

$$\overline{Rev} = \frac{\sum_i VP_i}{\sum_i VP_i + \sum_i FN_i} \quad (2)$$

Nessas equações, *i* varia de 1 até o número de classes do problema, enquanto o número de Verdadeiros Positivos, Falsos Positivos e Falsos Negativos são representados por *VP*, *FP e FN*, respectivamente. Os valores de AUPRC variam entre [0, 1] e quanto mais próximo de 1, melhor (VENS et al., 2008).

Essas medidas funcionam apenas para classificadores que apresentam um valor contínuo em sua saída e este é comparado com um determinado limiar. Assim, se uma saída é maior que o limiar então se atribui a instância à classe, caso contrário, a instância não é atribuída a classe. Quanto menor for o valor do limiar mais exemplos são atribuídos a uma classe aumentando a medida de revocação (BORGES, 2012).

## 2.5 CONSIDERAÇÕES FINAIS DO CAPÍTULO

Neste Capítulo apresentou-se uma breve introdução dos conceitos fundamentais de classificação de dados: plana, multirrótulo, hierárquica e hierárquica multirrótulo.

Na aprendizagem de máquina supervisionada, a classificação é uma tarefa preditiva, que tem como entrada um conjunto de exemplos rotulados e gera um modelo, que pode ser usado em novos dados ainda não rotulados.

Diferente da classificação plana, em que uma única classe ou rótulo está associada a um exemplo, este trabalho aborda a classificação hierárquica multirrótulo na qual cada exemplo pode pertencer simultaneamente a mais de uma classe, ou seja, duas ou mais classes, e estas podem ser agrupadas em subclasses ou superclasses e estão dispostas em uma estrutura hierárquica.

Foram apresentadas as técnicas utilizadas para a resolução desse tipo de problema de classificação, sendo a abordagem global capaz de realizar a predição considerando toda a informação da hierarquia. Em problemas de classificação hierárquica multirrótulo global, o número de exemplos, atributos e classes influenciam no desempenho dos métodos de classificação, pois os dados não são igualmente homogêneos, possuindo muitas vezes uma alta dimensionalidade de atributos, poucas amostras para treinamento e uma variação no número de classes a qual cada amostra pertence. Para enfrentar esse desafio, pode-se utilizar métodos de redução de dimensionalidade, detalhada no próximo Capítulo.

# 3 REDUÇÃO DE DIMENSIONALIDADE

Neste Capítulo serão apresentados conceitos básicos referentes à redução de dimensionalidade em base de dados. A Seção 3.1 apresenta os conceitos fundamentais sobre redução. Na Seção 3.2 são descritas as principais técnicas redução de dimensionalidade. Por fim, a Seção 3.3 relata as considerações finais do Capítulo.

## 3.1 CONCEITOS DE REDUÇÃO DE DIMENSIONALIDADE

Analisar dados de alta dimensão, com grande quantidade de variáveis, implicam em problemas importantes de capacidade computacional. A aquisição de informações fácil e barata, faz com que dados sejam coletados em diversas formas, por exemplo texto, áudio, imagens e vídeo, e estes por suas vez possuem uma alta dimensionalidade de atributos**.** Muitas vezes, para analisar esses dados de maneira adequada, a dimensionalidade precisa ser reduzida (ZHU; JIN; JI, 2017).

Dados de alta dimensão podem ser interpretados considerando dois aspectos: a quantidade de instâncias e número de atributos. Para superar este problema, em que existe uma grande quantidade de atributos, a redução da dimensionalidade dos dados é utilizada para minimizar o volume de dados a serem tratados em relação ao número de atributos e aumentar a capacidade de generalização dos métodos de aprendizagem com a eliminação de dados irrelevantes e/ou redundantes (BORGES; NIEVOLA, 2006).

A redução da dimensionalidade consiste em encontrar para dados de alta dimensão, uma representação significativa de dimensionalidade reduzida. Idealmente, a representação reduzida deve corresponder à dimensionalidade intrínseca dos dados de alta dimensão. A dimensionalidade intrínseca dos dados é o número mínimo de parâmetros necessários para explicar as propriedades observadas dos dados. A redução da dimensão é importante em muitos domínios, pois facilita, entre outros, a classificação, a visualização e a compressão dos dados (VAN DER MAATEN; POSTMA; VAN DEN HERIK, 2009).

A aplicação de técnicas de redução de dimensionalidade em bases de dados traz benefícios para a condução de estudos em diferentes áreas, pois pode remover o ruído e a redundância existente (não perceptível de outra maneira) entre os atributos, aumentando o potencial para a aprendizagem (categorizar, filtrar e procurar informações relevantes).

## 3.2 TÉCNICAS DE REDUÇÃO DE DIMENSIONALIDADE

Técnicas de redução de dimensionalidade em bases de dados surgiram como uma forma de melhorar o desempenho de algoritmos de Mineração de Dados e Inteligência Computacional, pois estes por muitas vezes, precisam processar um grande volume de dados com alta dimensionalidade. Quanto maior for à complexidade do processamento, maior será o custo temporal e computacional envolvido. Se a dimensionalidade for suficientemente grande, pode vir a tornar o processo de análise dos dados impraticável (FAYYAD et al., 1996).

A redução da dimensionalidade, segundo Ghodsi (2006), também pode ser vista como o processo de derivar um conjunto de graus de liberdade, o qual pode ser usado para reproduzir a maior parte da variabilidade de um conjunto de dados.

De acordo com Borges e Nievola (2006), a redução de dimensionalidade pode ser realizada por meio da Seleção de Atributos que escolhe um subconjunto ótimo de atributos a partir dos originais ou pela Extração de Atributos (Extração de Características) que cria novos atributos por meio de transformações, lineares ou não, dos atributos originais.

Portanto, a escolha entre seleção e a extração de atributos, torna-se dependente do domínio da aplicação e dos dados de treinamento disponíveis.

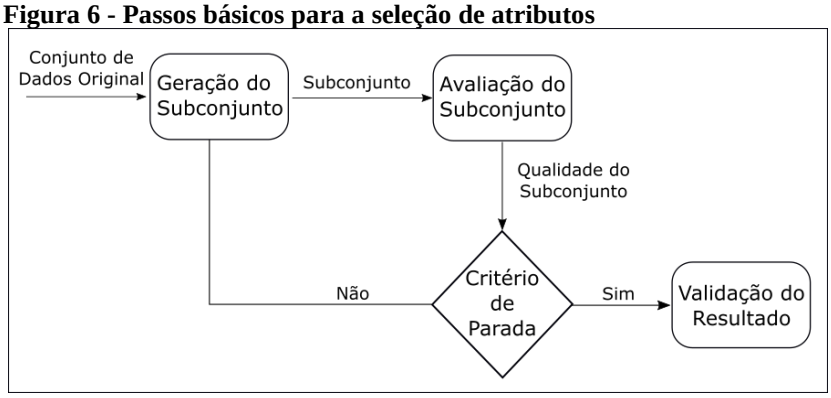
### 3.2.1 Seleção de Atributos

A tarefa de selecionar o subconjunto de atributos "subótimos" é conhecida como seleção de atributos, que é uma das técnicas utilizadas para análise de padrões, mineração de dados e aprendizado de máquina. De maneira geral, a seleção de atributos é o processo no qual se escolhe um subconjunto ideal de $m$ atributos em um conjunto original de alta dimensão de $n$ atributos ($m \leq n$), reduzindo a quantidade de atributos dado um determinado critério (LIN; GUNOPULOS, 2003).

O procedimento geral para seleção de recursos tem quatro etapas principais, apresentadas na Figura 6. A geração do subconjunto pode ser dita como uma busca heurística para se produzir subconjuntos de atributos para posterior avaliação no espaço de busca. Primeiramente, decide-se por um subconjunto de partida, através da organização e seleção de atributos com uma estratégia específica. O subconjunto gerado é avaliado por um determinado critério de avaliação, envolvendo ou não algoritmos pré-determinados, buscando garantir a qualidade dos atributos selecionados. Quando os critérios de parada são satisfeitos, a seleção

de atributos termina no procedimento de validação, caso contrário, o processo de geração e avaliação do subconjunto é repetido novamente (DASH; LIU, 1997).

**Figura 6 - Passos básicos para a seleção de atributos**




**Fonte: Adaptado de Dash e Liu (1997)**

A seleção de atributos procura estudar o relacionamento existente entre os atributos de um conjunto de dados e escolher os mais significativos para formar um subconjunto de dados ideal simplificado, que produz o mesmo resultado analítico (ou o mais próximo possível) aos obtidos quando analisado o conjunto completo dos dados para uma determinada tarefa alvo.

O subconjunto de recursos ideal é um subconjunto de todos os recursos relevantes. Na literatura, as características são classificadas por sua relevância com três qualificadores: irrelevantes, fracamente e altamente relevantes. O processo de seleção de atributos busca encontrar os atributos que possuem relacionamentos relevantes, para formar um novo subconjunto, e nesse processo é desejável que atributos irrelevantes sejam desconsiderados. (KUMAR; MINZ, 2014, P. 213).

Algoritmos de seleção de atributos foram desenvolvidos com diferentes critérios de avaliação, sendo as principais a abordagem filtro, *wrapper* e embutido.

- Abordagem Filtro

A abordagem filtro tenta escolher um subconjunto de atributos independente do algoritmo de aprendizagem, estimando a qualidade dos atributos apenas em relação aos dados. Essa abordagem é computacionalmente eficiente, porém os métodos de filtragem podem perder recursos que não são úteis por si mesmos, mas podem ser úteis quando combinados com outros (KUMAR; MINZ, 2014).

O processo de seleção de um subconjunto de atributos que ocorre antes da aplicação do algoritmo de indução tem como objetivo filtrar atributos irrelevantes, usando características gerais do conjunto de treinamento para selecionar alguns recursos e excluir outros. Assim,

métodos de filtragem são independentes do algoritmo de indução que usará sua saída e podem ser combinados com qualquer método (JOHN et al., 1994).

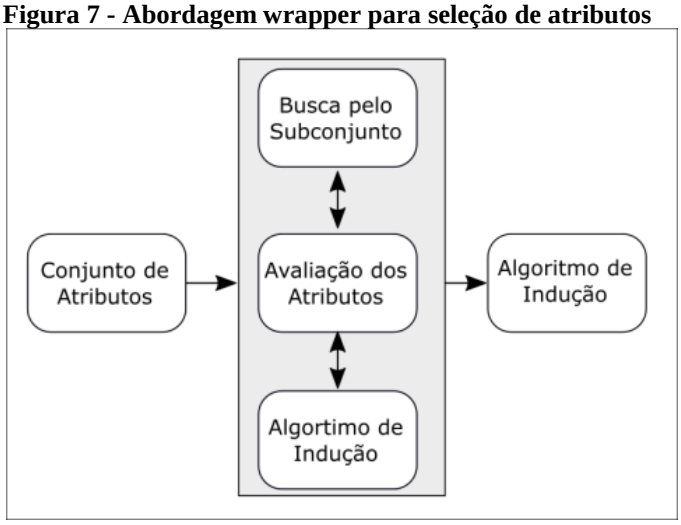
- Abordagem *Wrapper*

A abordagem *wrapper* necessita pré-determinar um algoritmo de mineração utilizando seu desempenho como critério de avaliação. Para Kumar e Minz (2014), a ideia geral da abordagem *wrapper* é a avaliação do conjunto de atributos usando um algoritmo que funciona como uma caixa preta para encontrar os melhores subconjuntos de atributos.

O propósito é encontrar o conjunto de atributos que melhor se adeque ao algoritmo de aprendizado, por isso essa abordagem é dependente do algoritmo de aprendizado. Tem desempenho superior, mas tende a ser mais custosa computacionalmente por estimar a precisão de cada subconjunto de característica (JOHN et al., 1994).

Em contraste a abordagem filtro, a abordagem *wrapper* (Figura 7) gera um subconjunto de atributos para avaliação, executa o algoritmo de indução com apenas esses atributos no conjunto de treinamento e utiliza a precisão do classificador para avaliar o subconjunto de atributos em questão. O processo é repetido para todos os subconjuntos gerados para a avaliação ou até que o critério de parada seja satisfeito.

**Figura 7 - Abordagem wrapper para seleção de atributos**




**Fonte: John et al. (1994)**

- Abordagem *Embedding*

A abordagem Embutida (do inglês *Embedding*) considera que o processo de seleção de variáveis está implícito no processo de síntese do modelo classificador ou regressor. Alguns

indutores são capazes de realizar sua própria seleção de atributos de forma dinâmica enquanto procuram por uma solução, por exemplo, as árvores de decisão que utilizam uma função de avaliação para selecionar o atributo que tem a melhor capacidade de discriminar entre as classes. O conjunto de treinamento é particionado baseado nesse atributo e o processo é repetido para cada subconjunto, estendendo a árvore até que nenhuma discriminação adicional seja possível.

Dessa maneira, em ambas abordagens, durante a busca heurística para se produzir os subconjuntos de atributos ideal, dados redundantes e/ou irrelevantes, por não terem informação útil, são descartados

### 3.2.2 Extração de Atributos

Na abordagem denominada extração de atributos (ou extração de características), ao invés de se escolher um subconjunto de atributos, todos os atributos originais são utilizados para projetar um espaço de menor dimensão, no qual o máximo de informações dos dados originais é preservada.

Para Jain et al. (2000), dado um espaço de características de dimensão *n*, os métodos de extração de características determinam um subespaço apropriado de dimensionalidade *m* tal que $m < n$.

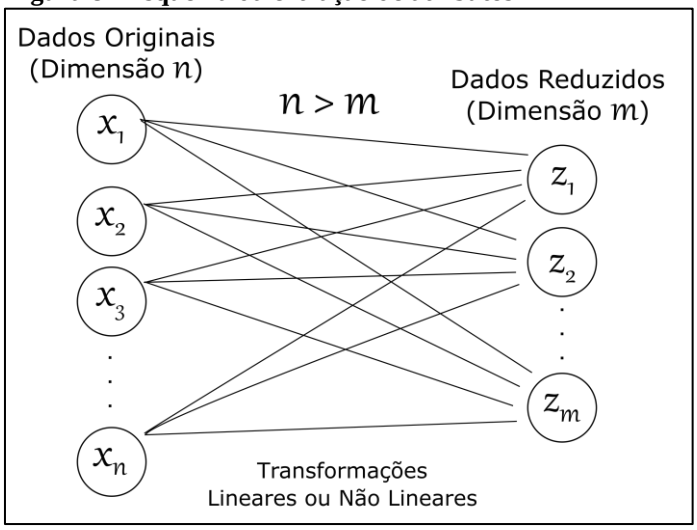
Isso faz com que essa abordagem seja utilizada principalmente em aplicações em que a dimensionalidade dos dados é um fator limitante, como por exemplo, no reconhecimento de padrões em imagens, sendo capaz de extrair as características mais significativas dos dados (THEODORIDIS et al., 2010).

De acordo com Campos (2001), a extração de atributos é o processo que cria novas características a partir de transformações ou combinações do conjunto de características inicial, ou seja, criam-se novas características que sejam mais expressivas e que melhor representam a variabilidade dos dados.

Os atributos dos dados originais são representados na Figura 8 por $x_n$, em que $n$ representa o número de atributos (dimensionalidade). Os dados reduzidos são representados $z_m$, em que $m$ representa a dimensionalidade dos dados reduzidos. O funcionamento básico da extração de atributos baseia-se na escolha de transformações (lineares ou não) nos dados $(x_1, x_2, \dots, x_n)$ de maneira que ocorra uma alta condensação de informações em poucas características $(z_1, z_2, \dots, z_m)$ que representam mais significativamente o conjunto original em

um espaço de menor dimensão, proporcionando que a redundância e o ruído dos dados sejam removidos.

**Figura 8 - Esquema da extração de atributos**




**Fonte: Autoria própria**

Formalmente, a extração de atributos pode ser definida da como um conjunto $X$ de $n$ características originais, tal que $X = \{x_1, x_2, \dots, x_n\}$, que após ser submetido a um processo de extração de atributos, gera um novo conjunto de características $Z$, com $m$ características, tal que $Z = \{z_1, z_2, \dots, z_m\}$, sendo que $m < n$. Assim, uma nova característica extraída $z_m$, é obtida por $z_m = f_m(X)$, onde $f$ é uma função de mapeamento que realiza transformações, lineares ou não lineares, sobre as características conjunto de atributos originais $X$.

Transformações lineares não modificam a estrutura espacial dos dados, conservando assim as relações entre as observações. A Análise de Componentes Principais (*Principal Component Analysis* - PCA) segundo Jolliffe (2011), é considerada a transformação linear ótima para a redução de dimensionalidade, pois gera novos atributos que são funções lineares dos atributos originais, que maximizam a proporção da variância do conjunto de dados expressa por sucessivas componentes principais que não são correlacionadas entre si. No entanto, o PCA tem sua eficácia limitada a linearidade dos dados (GHODSI, 2006).

Em problemas de natureza não linear, a extração de características frequentemente envolve a aplicação de transformações não lineares. Estes métodos de transformação são eficientes na aproximação de funções e robustos no tratamento de problemas reais não lineares (WANG, 2009).

As transformações não lineares transformam (aumentam ou diminuem) as relações lineares entre as variáveis e, assim, alteram a correlação entre as variáveis. Por exemplo, o

método de projeção aleatória *(Random Projection)*, segundo Bingham e Mannila (2001), é um método não linear, no qual os dados originais de alta dimensão são projetados em um subespaço de dimensão inferior usando uma matriz aleatória.

Existem na literatura diversas técnicas para redução de dimensionalidade, para diversos domínios, que utilizam a abordagem de extração. São exemplos de técnicas de extração de atributos já consolidadas, além das citadas anteriormente: *Multidimensional Scaling - MDS* (KRUSKAL, 1964); *Projection Pursuit* (FRIEDMAN; TUKEY, 1974); *FastMap* (FALOUTSOS; LIN, 1995); *Linear Discriminant Analysis - LDA* (BALAKRISHNAMA; GANAPATHIRAJU, 1998)*; Self Organizing Map - SOM* (KOHONEN, 1990)*;* entre outros.

O objetivo principal da extração de características é encontrar um conjunto mínimo de novos atributos que obedeça a uma medida de desempenho. Vale ressaltar, que para atingir este objetivo é necessária uma busca intensiva, geralmente demorada e possivelmente com alto custo computacional envolvido. A definição de uma medida de desempenho que tem como finalidade avaliar se uma característica construída é boa ou não, também é uma atividade complexa (CAMARGO, 2010).

Assim a aplicação de técnicas não lineares de extração deve considerar que esse processo consome tempo pela necessidade de pesquisar novas características que satisfaçam o critério de desempenho. O modelo de classificação ou regressão gerado a partir de características extraídas pode ser de difícil compreensão, dada a redução do espaço e a transformação sofrida pelos dados originais. Desta forma, deve ser analisada a relação custo x benefício entre o tempo gasto e a otimização obtida no processo de redução de dimensionalidade.

## 3.3 CONSIDERAÇÕES FINAIS DO CAPÍTULO

Neste Capítulo foram apresentados os conceitos fundamentais da redução de dimensionalidade em base de dados. A redução tem como objetivo aumentar a capacidade de generalização dos métodos de aprendizagem de máquina, quando estes utilizam dados que possuem uma grande quantidade de atributos.

Para que os algoritmos tratem esses dados de maneira adequada, a dimensionalidade dos dados precisa ser reduzida. Na literatura a seleção e a extração de atributos são apresentadas como as duas principais técnicas de redução de dimensionalidade. A técnica de seleção de

atributos identifica e seleciona um subconjunto de atributos relevantes, por meio da avaliação do conjunto original de atributos, utilizando a abordagem filtro, *wrapper* ou embutida.

Na extração de atributos, novos atributos são projetados em um espaço de menor dimensão, a partir de transformações ou combinações do conjunto de atributos originais. Espera-se que características e informações relevantes do conjunto de dados de alta dimensionalidade sejam preservadas nesse processo. A abordagem de extração de atributos é utilizada neste trabalho, para reduzir a dimensionalidade dos dados utilizados na classificação hierárquica multirrótulo global.

# 4 REDES NEURAIS ARTIFICIAIS

Neste Capítulo serão apresentados conceitos gerais sobre Redes Neurais Artificiais, Aprendizagem Profunda e Autoencoders. A Seção 4.1 descreve os conceitos fundamentais sobre redes neurais artificiais. A Seção 4.2 apresenta a aprendizagem profunda. A Seção 4.3 apresenta os Autoencoders. Por fim, a Seção 4.4 relata as considerações finais do Capítulo.

## 4.1 CONCEITOS FUNDAMENTAIS SOBRE REDES NEURAIS ARTIFICIAIS

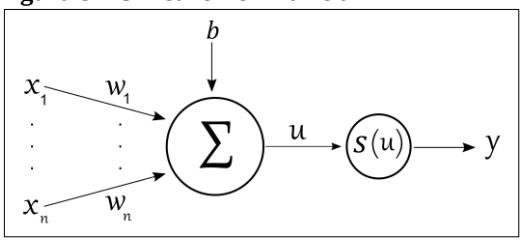
As redes neurais também chamadas de Redes Neurais Artificiais (RNA) são um tipo de aprendizado de máquina baseada no modo como os neurônios funcionam no cérebro humano. São métodos computacionais que, na maioria das vezes, usam camadas de nós (ou "neurônios") que funcionam em paralelo para aprender conceitos, identificar padrões e tomar decisões de maneira parecida com a humana (HAYKIN, 2009).

Para Carvalho (2011) uma RNA pode ser caracterizada por dois aspectos básicos: a arquitetura, que está relacionada ao tipo, ao número e forma de conexão das unidades de processamento (neurônios); e o aprendizado, que são as regras e as informações utilizadas para ajustes dos pesos da rede.

De acordo com Haikyn (2009) as RNA têm a capacidade de "aprender" por meio de exemplos e de generalizar este aprendizado de forma a reconhecer elementos similares. O aprendizado por meio de exemplos possui bom desempenho em tarefas pouco ou mal definidas, nas quais falta o conhecimento explícito de como resolvê-las.

Assim como o neurônio biológico, o neurônio artificial (Figura 9) possui um ou mais sinais de entrada e apenas um sinal de saída. Cada sinal de entrada recebe um valor ($x_1, \dots, x_n$), e estes são ponderados pelos pesos ($w_1, \dots, w_n$), combinados por uma função que equivale ao processamento realizado pela soma ($\sum$), acrescido do valor do bias, ou viés, ($b$). A saída de um neurônio ($y$) é a resposta para as entradas.

**Figura 9 - O Neurônio Artificial**




**Fonte: Adaptado de Haykin (2009)**

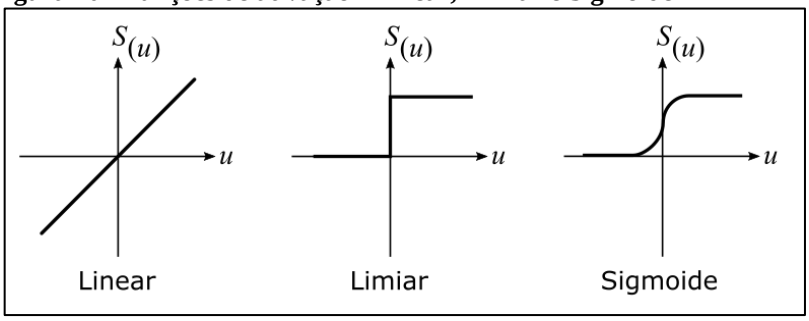
Assim, uma instância $x$ com $n$ atributos representados na forma de um vetor $x = [x_1, x_2, \dots, x_n]$ e um neurônio $j$ com $n$ terminais de entrada cujos os pesos são representados por um vetor $W = [w_{1j}, w_{2j}, \dots, w_{nj}]$. A entrada total recebida por um neurônio $j$, chamada de potencial de ativação ($u$), para uma instância $x$ pode ser definida pela Equação (3) (HAYKIN, 2009).

$$u = \sum_{i=1}^{n} x_i w_{ij} + b_j \tag{3}$$

As conexões de entrada podem ser negativas ($w_i < 0$), positivas ($w_i > 0$) e iguais à zero. A saída $y$ de um neurônio é definida por meio da aplicação de uma função de ativação ($S$) a entrada total (CARVALHO et al., 2011).

As funções de ativação (Figura 10) geralmente utilizadas na literatura são: linear, limiar e sigmoidal. A função linear retorna como saída o valor de $u$. Na função limiar é definido o resultado da função como sendo 1 ou 0 de forma alternada e na sigmoidal é uma representação aproximada, contínua e diferençável da função limiar (CARVALHO et al., 2011).

**Figura 10 - Funções de ativação - Linear, Limiar e Sigmoide**




**Fonte: Autoria própria**

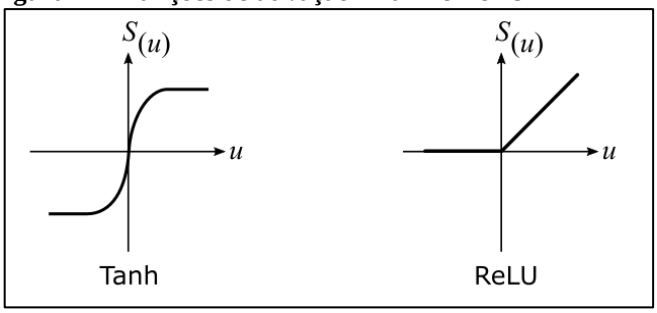
A função de ativação sigmóide, é utilizada para dados contínuos. É definida como uma função crescente, que recebe como entrada o potencial de ativação ($u$) do neurônio e então apresenta um balanço entre o comportamento linear e não-linear, e está definida pela Equação (4) (CARVALHO et al., 2011).

$$s(u) = \frac{1}{1+e^{u}} \tag{4}$$

Pode-se se citar a tangente hiperbólica *Tanh* (Figura 11) que funciona de forma semelhante à função sigmoide, varia de -1 a 1, em vez de 0 a 1 como na sigmoide. Em redes neurais é comum o uso de funções sigmoides (como a logística e a tangente hiperbólica).

Porém na *Deep Learning*, a função retificadora linear (*Rectified Linear Function* - ReLU) tem sido mais utilizada por facilitar o processo de treinamento. Isso porque as funções sigmoides saturam a partir de um determinado ponto, enquanto a ReLU (Figura 11), é simplesmente a função identidade para valores positivos (GOODFELLOW; BENGIO; COURVILLE, 2016).

**Figura 11 - Funções de ativação - Tanh e ReLU**




**Fonte: Autoria própria**

Uma comparação entre essas funções, é que funções sigmoidais comprimem a saída para um intervalo curto, enquanto ReLU cancela todos os valores negativos, sendo linear para os positivos. A função ReLU possui relações com a restrição de não-negatividade presente em regularização, como em restauração de imagens utilizando projeções em subespaços (NAIR; HINTON, 2010).

A principal vantagem da função de ativação ReLU sobre outras funções de ativação, é que ela não ativa todos os neurônios ao mesmo tempo. É definida como uma função não linear que produz zero em metade do seu domínio. É definida pela Equação (5) (GOODFELLOW; BENGIO; COURVILLE, 2016).

$$f(x) = max\,(0, x) \tag{5}$$

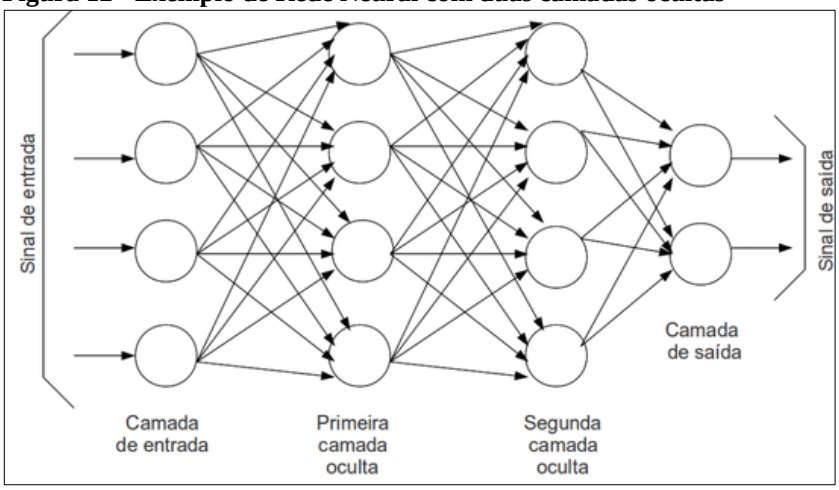
De acordo com Haykin (2009), um componente importante do neurônio artificial são os pesos ($w$), que representam o grau de importância que determinada entrada possui em relação àquele neurônio. O valor do peso é alterado em função da intensidade do sinal de entrada, e dessa forma, o peso muda o seu valor representativo para a rede (processo de aprendizagem).

Se o valor resultante da soma, entre o produto dos sinais de entrada pelos respectivos pesos atingiu o limiar, ele é repassado posteriormente na saída. Caso contrário, se o valor não atingiu o limiar, o sinal não será transferido. Esse processo de verificação é chamado de função de transferência, que também é conhecido como limiar lógico (HAYKIN, 2009).

Nas redes neurais biológicas, o conjunto de vários neurônios artificiais interconectados formam as redes neurais artificiais, que podem ter uma ou várias camadas. As redes que possuem uma única camada são as que possuem um nó entre uma entrada e uma saída e são indicadas para a solução de problemas linearmente separáveis.

As redes neurais artificiais multicamadas (Figura 12), possuem mais de uma camada (a quantidade de camadas pode ser indeterminada), que ficam escondidas (ocultas) entre as camadas de entrada e saída. Redes *Perceptron* de Múltiplas Camadas (MLP do inglês *Multilayer Perceptron*) são consideradas redes aproximadoras universais, podem realizar qualquer tipo de mapeamento linear ou não-linear entre as entradas e saídas, pois apresentam uma ou mais camadas intermediárias de neurônios que intervém entre a camada de entrada e a de saída da rede de forma útil (HAYKIN, 2009).

**Figura 12 - Exemplo de Rede Neural com duas camadas ocultas**




**Fonte: Adaptado de Haykin (2009)**

Em uma rede multicamadas, os padrões de conectividade, segundo Carvalho et. al. (2011), podem ser classificados como:

- Completamente conectada: quando os neurônios da rede estão conectados a todos os neurônios da camada anterior e/ou seguinte.
- Parcialmente conectada: quando os neurônios da rede estão conectados a apenas alguns os neurônios da camada anterior e/ou seguinte.
- Localmente conectada: são redes parcialmente conectadas, em que os neurônios que se conectam a outro neurônio são de uma região bem definida.

A informação em uma rede neural geralmente flui da camada de entrada para a camada de saída (*feedforward*). Em redes multicamadas o fluxo ocorre em camada por camada, mas uma RNA pode ou não apresentar conexões de retroalimentação (*feedback*).

Conexões de retroalimentação permitem que um neurônio receba em seus terminais de entrada, a saída de um neurônio da mesma camada, de uma camada posterior ou até mesmo a sua própria saída. Redes que utilizam conexões de retroalimentação são chamadas de recorrentes. Uma rede não recorrente, isto é, que não utiliza conexões de retroalimentação, é chamada de *feedforward* (CARVALHO et al., 2011)..

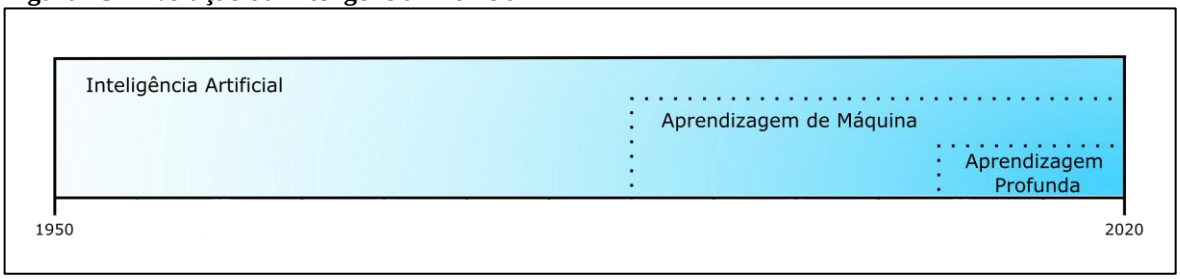
Dessa maneira, pode-se dizer que a topologia de uma RNA é definida pelo número de camadas, quantidade de neurônios por camada, a conectividade entre camadas e a presença ou não de conexões de retroalimentação entre neurônios.

## 4.2 APRENDIZAGEM PROFUNDA

Na Inteligência Artificial, como uma subcategoria da aprendizagem de máquina, a aprendizagem profunda (*Deep Learning*), também conhecida como: aprendizagem estruturada profunda, aprendizagem hierárquica ou aprendizagem profunda de máquina surgiu inspirada em estudos biológicos do funcionamento da região do córtex-visual dos mamíferos. É principalmente utilizada com o objetivo de tratar a dificuldade com que as arquiteturas tradicionais utilizadas em RNA possuem com dados de alta dimensionalidade (GOODFELLOW; BENGIO; COURVILLE, 2016).

A Figura 13 apresenta uma linha do tempo e a relação entre esses conceitos.

**Figura 13 - Evolução da Inteligência Artificial**




**Fonte: Autoria própria**

Essa subcategoria da aprendizagem de máquina pode ser compreendida como uma interseção entre as áreas de pesquisa de redes neurais, inteligência artificial, modelagem gráfica, otimização, reconhecimento de padrões e processamento de sinais. Permite identificar abstrações dos dados, de níveis mais baixos para os mais altos, obtendo novas representações,

na qual as características de alto nível são formadas pela composição das de baixo nível, sendo sua principal atribuição a extração de características dos dados (BENGIO, 2012).

Os modelos gerados, para Bengio (2012), podem ser caracterizados pela utilização de múltiplas camadas ou estágios de processamento de informação. Essas redes são empregadas para a aprendizagem supervisionada ou não supervisionada, buscando a representação de características em camadas sucessivamente mais altas e mais abstratas.

Deng e Yu (2014) caracterizam a aprendizagem profunda como RNA que utilizam várias camadas de informação não lineares de natureza hierárquica para a extração de características tanto de maneira supervisionada quanto não-supervisionada e pelo uso de algoritmos de aprendizado de múltiplos níveis de representação para identificar relações entre os dados.

De maneira semelhante, Bengio, Courville e Vincent (2013), explicam que a aprendizagem profunda é parte de uma família de métodos de aprendizagem de máquina baseada em representações de aprendizagem de dados. Comentam que uma imagem pode ser representada, por exemplo, simplesmente como valores de intensidade de pixel ou de maneira mais abstrata, como conjunto de arestas, contornos, formas etc. Concluem que algumas dessas representações, são melhores do que outras, por exemplo, para a área de reconhecimento facial. Dessa maneira, pode ser interessante e útil explorar quais são as melhores representações para objetos de um determinado domínio.

O aprendizado baseado em representações tem como objetivo melhorar e criar modelos para aprender representações de maneira eficiente, principalmente para a aprendizagem de recursos não-supervisionados ou semi-supervisionados e extração de recursos hierárquicos, dentro de um domínio específico (BENGIO; COURVILLE; VINCENT, 2013).

Outra consideração interessante, é que as representações geradas pelas interações de características presentes nos dados, estão organizadas nas camadas, que correspondem a níveis de abstração ou composição. A variação no tamanho e número de camadas pode fornecer diferentes quantidades de abstração, em que conceitos de nível superior mais abstrato são aprendidos com os de nível inferior (BENGIO; COURVILLE; VINCENT, 2013).

Segundo Neto et al. (2017), dependendo de como as arquiteturas e técnicas de aprendizagem profunda são empregadas, por exemplo, na síntese/geração ou reconhecimento/classificação, pode-se categorizar (Quadro 1) a maior parte do trabalho nessa área em três grupos principais: discriminativos, generativos e híbridos.

**Quadro 1 - Categorização quanto à estrutura e técnica empregada**

| | |
|---|---|
| **Aprendizagem supervisionada** (Modelos Discriminativos) | Se destinam a fornecer diretamente poder discriminativo para fins de classificação de padrões. Também chamadas de redes profundas discriminativas. Os dados do rótulo-alvo estão sempre disponíveis de forma direta ou indireta. |
| **Aprendizado não-supervisionado** (Modelos Generativos) | Busca capturar a correlação de alta ordem dos dados observados ou visíveis para fins de análise ou síntese de padrões quando não há informações sobre os rótulos das classes disponíveis na base. |
| **Híbridos** | Se destacam pela discriminação que é assistida, muitas vezes de forma significativa, com os resultados de redes profundas generativas ou não supervisionadas. Combinam a abordagem não-supervisionada e supervisionada. |

**Fonte: Adaptado de Neto et. al. (2017)**

Dentre os discriminativos, podem ser elencadas: *Deep Neural Network - DNN*, *Recurrent Neural Network - RNN*, *Convolutional Neural Network - CNN*, dentre outras. Quanto aos modelos generativos, observam-se as *Deep Boltzmann Machines - DBM*, *Deep Autoencoders* (NETO et al., 2017).

Essas arquiteturas e técnicas, por meio de suas séries de muitas camadas, buscam dividir um problema complexo em questões simples e específicas sobre as entradas, em suas camadas iniciais e, posteriormente, nas camadas mais profundas constrói-se uma hierarquia de conceitos cada vez mais complexos e abstratos (BENGIO, 2012).

O aspecto chave da *Deep Learning* são as múltiplas camadas ou estágios de processamento de informação não linear que constituem sua estrutura e que pode ser aplicada na aprendizagem supervisionada, não supervisionada ou na combinação de ambas. A representação de características ocorre em camadas sucessivamente mais profundas e abstratas sendo que a aprendizagem profunda ajuda a encontrar essas abstrações e escolher quais recursos são úteis para a aprendizagem (NETO et al., 2017).

Dentro dessa perspectiva, Wang, Yao e Zhao (2016) exploram o modelo generativo fornecido por um *Autoencoder*, e demostram que esta rede de aprendizagem profunda, tem em sua estrutura de camadas, uma interessante propriedade: a capacidade de redução de dimensionalidade dos dados de entrada.

Um *Autoencoder* ao reconstruir na sua camada de saída os dados fornecidos na entrada, e restringir o número de neurônios existentes nas camadas ocultas, entre as camadas de entrada e saída, pode obter uma representação reduzida, em suas camdas ocultas, dos dados fornecidos na entrada. A próxima seção, descreve brevemente o funcionamento desse tipo de rede.

## 4.3 AUTOENCODERS

Segundo Haykin (2009), neurônios ocultos de um *perceptron* multicamadas desempenham um papel crítico como extratores de características e esta propriedade pode ser explorada como um replicador ou mapa de identidade dos dados.

*Autoencoders* (Redes Autocodificadoras ou Redes Autoassociativas), não são treinados para predizer alguma classe, mas para reproduzir na sua saída à própria entrada. Porém, essa estrutura fornece uma representação de menor dimensão do conjunto de dados. Os seguintes requisitos estruturais são descritos, para uma *Replicator Neural Networks* (Redes Neurais Replicadoras) em Haykin (2009), e estes satisfazem o *layout* de uma rede autocodificadora clássica de 3 camadas, em que:

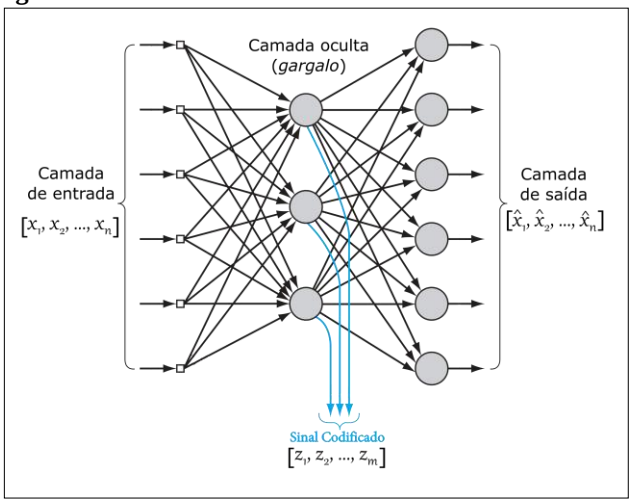
- As camadas de entrada e saída têm a mesma dimensão $n$.
- A dimensão da camada oculta, tem valor $m$, e $m$ é menor que $n$.
- A rede é simétrica e está totalmente conectada.

Com esses requisitos estruturais, para Haykin (2009), pode-se afirmar que a rede terá:

- Uma Função Codificadora.
- Uma Função Decodificadora.

A menor estrutura de um *Autoencoder* é composta por no mínimo 3 camadas, sendo: uma camada de entrada, uma de saída e uma camada oculta entre essas. A Figura 14 demonstra visualmente o *layout* da estrutura descrita.

**Figura 14 - Estrutura de um Autoencoder de 3 camadas**




**Fonte: Adaptado de Haykin (2009)**

Nessa estrutura, a camada de entrada recebe um conjunto de pontos de dados $X$, no qual cada ponto $x$ (que representa um exemplo de treinamento) com $n$ dimensões, tal que $x =$

$[x_1, x_2, \dots, x_n]$, pode ser mapeado para outro conjunto de pontos de dados $Z$, no qual cada ponto $z$ representa uma codificação do exemplo de treinamento com dimensionalidade $m$, tal que $z = [z_1, z_2, \dots, z_m]$ (HAYKIN, 2009).

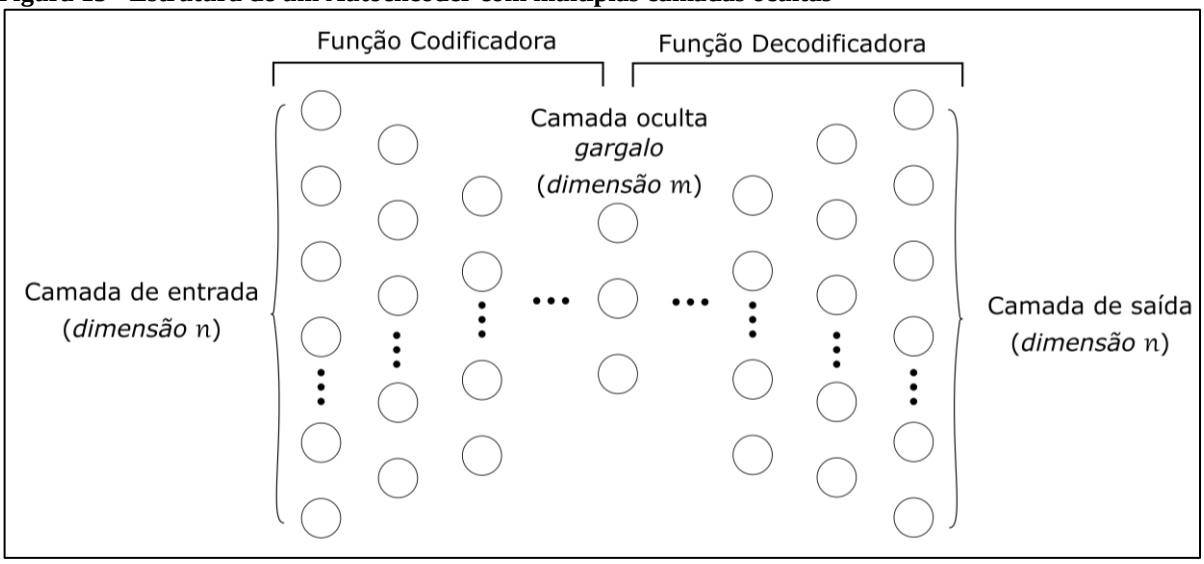
Como esta estrutura tenta reconstruir as entradas em sua camada de saída; tem a mesma quantidade *n* de neurônios nas camadas de entrada e saída; e necessariamente a camada oculta tem uma dimensão menor que *n*, pode-se afirmar que:

- O conjunto de pontos de dados $Z$ têm menor dimensionalidade do que o conjunto de pontos de dados $X$;
- O conjunto de pontos $Z$ pode reconstruir $X$.

Vale ressaltar que um *Autoencoder* pode ter múltiplos pares de camadas ocultas, como descrito em Bengio et al. (2007), porém como representado na Figura 14 e nos requisitos estruturais descritos por Haykin (2009), deve ter somente uma camada oculta com dimensão $m$. Esta camada pode ser chamada de *bottleneck* (gargalo).

Outro requisito importante a ser observado em um *Autoencoder* com múltiplos pares de camadas ocultas (Figura 15), diz repeito a simetria estrutural, que deve garantir que: as camadas ocultas possuam dimensionalidade inferior a *n* (camadas de entrada e saída); superior a *m* (*bottleneck*) e ainda sejam simétricas entre si e em relação a camada *bottleneck.* As camadas ocultas contêm menos neurônios que a camada de entrada e saída para que a rede seja capaz de codificar as informações das entradas por meio da função codificadora e reconstruir as entradas nas saídas utilizando a função decodificadora (BENGIO et al., 2007; HAYKIN, 2009).

**Figura 15 - Estrutura de um Autoencoder com múltiplas camadas ocultas**




**Fonte: Autoria própria**

- Função Codificadora

A função codificadora (estrutura codificadora), denotada por $f$, deve mapear um exemplo de entrada $x \in \mathbb{R}^n$ para uma representação compacta $z \in \mathbb{R}^m$ através das ativações dos $m$ neurônios da camada oculta, tal que $m < n$ (BENGIO et al., 2007; HAYKIN, 2009).

A função codificadora $f$, para um exemplo de entrada $x$ com $n$ atributos $x = [x_1, x_2, \dots, x_n]$, em uma rede que contém somente uma camada oculta, na qual se obtém uma representação reduzida $z$ com $m$ atributos, tal que $z = [z_1, z_2, \dots, z_m]$, pode ser escrita conforme a Equação (6).

$$z = f(x) = s(Wx + B) \tag{6}$$

na qual:

- $W$ é uma matriz dos pesos, tal que $W \in \mathbb{R}^{n \times m}$;
- $B \in \mathbb{R}^m$ representa um vetor com os valores viés (bias) dos neurônios.
- $s$ a função de ativação;

O conjunto de exemplos de entrada $X$, também pode ser representado pela notação matricial. O resultado da combinação linear computada pelos neurônios é demonstrado pela Equação (7).

$$Z = s(X.W + B) \tag{7}$$

Sendo $X.W + B$:

$$\begin{bmatrix} x_{1,1} & x_{1,2} & x_{1,3} & \dots & x_{1,n} \\ x_{2,1} & x_{2,2} & x_{2,3} & \dots & x_{2,n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_{i,1} & x_{i,2} & x_{i,2} & \dots & x_{i,n} \end{bmatrix} . \begin{bmatrix} w_{1,1} & w_{1,2} & w_{1,3} & \dots & w_{1,m} \\ w_{2,1} & w_{2,2} & w_{2,3} & \dots & w_{2,m} \\ w_{3,1} & w_{3,2} & w_{3,3} & \dots & w_{3,m} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ w_{n,1} & w_{n,2} & w_{n,3} & \dots & w_{n,m} \end{bmatrix} + [b_1 \quad b_2 \quad b_3 \quad \dots \quad b_m]$$

Tal que:

- $X$ é uma matriz em que cada linha é um exemplo $i$ (ou seja, cada linha possui $n$ valores, referentes aos $n$ atributos).
- $W$ é uma matriz que contém os pesos com os quais as entradas de cada exemplo serão transformadas. Na função de codificação, $W$ possuí $m$ colunas (quantidade de neurônios) e $n$ linhas (quantidade de atributos de entrada). Cada coluna contém os $n$ pesos de um neurônio.

- $B$ é um vetor de termos bias, sendo que o mesmo contém $m$ colunas que armazena o bias de cada neurônio.

Então, deve-se aplicar a função de ativação $s$ no resultado da multiplicação de linha da matriz X pelos elementos da coluna da matriz W e somá-los, acrescentando a essa soma o bias correspondente.

A estrutura da rede é utilizada para realizar o mapeamento de identidade (extração de características) por meio das camadas ocultas. Uma versão codificada do padrão de entrada, em Z, que pode ser denotada por $z_{ij}$, é produzida na saída da camada *bottleneck*.

$$Z = \begin{bmatrix} z_{1,1} & z_{1,2} & z_{1,3} & \cdots & z_{1,m} \\ z_{2,1} & z_{2,2} & z_{2,3} & \cdots & z_{2,m} \\ z_{3,1} & z_{3,2} & z_{3,3} & \cdots & z_{3,m} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ z_{i,1} & z_{i,2} & z_{i,3} & \cdots & z_{i,m} \end{bmatrix}$$

Ao final da função codificadora $f$, $Z$ é uma matriz em que cada linha é uma amostra com $m$ novos atributos que são transformações/combinações dos $n$ atributos originais.

- Função Decodificadora

A camada de saída é composta de $n$ neurônios e estes recebem como entradas os sinais de saída da camada oculta das $m$ representações reduzidas obtidas $z = [z_1, z_2, \dots, z_m]$. A função decodificadora, representada por $g$, tenta reconstruir um exemplo de entrada $x$ com $n$ atributos, $x = [x_1, x_2, \dots, x_n]$, a partir de uma representação reduzida $z$ com $m$ atributos, em que $z = [z_1, z_2, \dots, z_m]$, obtida pela função codificadora na camada oculta (BENGIO et al., 2007; HAYKIN, 2009).

Para uma representação reduzida $z$, obtida para uma entrada $x$, em uma rede que contém somente uma camada oculta, a função decodificadora $g$, pela qual se obtém a reconstrução $\hat{x}$ é mostrada pela Equação (8).

$$g(z) = s(\widehat{W}z + \hat{B}) \tag{8}$$

Com $\widehat{W} \in \mathbb{R}^{n \times m}$ contendo os pesos para a reconstrução e $\hat{B} \in \mathbb{R}^n$ que é o bias dos neurônios da camada de saída. Logo, tem-se que:

- $z$ é uma representação reduzida com $m$ atributos de uma entrada $x$;
- $\widehat{W}$ é uma matriz dos pesos da estrutura decodificadora, tal que $W \in \mathbb{R}^{n \times m}$;

- $s$ é função de ativação da camada de saída;
- $\hat{x}$ é um exemplo reconstruído.

Utilizando a notação matricial, o resultado da combinação linear computada pelos neurônios é descrita por meio da Equação (9):

$$\hat{X} = s(Z.\widehat{W} + \hat{B}) \tag{9}$$

Em que:

$$\begin{bmatrix} z_{1,1} & z_{1,2} & z_{1,3} & \dots & z_{1,m} \\ z_{2,1} & z_{2,2} & z_{2,3} & \dots & z_{2,m} \\ z_{3,1} & z_{3,2} & z_{3,3} & \dots & z_{3,m} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ z_{i,1} & z_{i,2} & z_{i,3} & \dots & z_{i,m} \end{bmatrix} . \begin{bmatrix} \widehat{w}_{1,1} & \widehat{w}_{1,2} & \widehat{w}_{1,3} & \dots & \widehat{w}_{1,m} \\ \widehat{w}_{2,1} & \widehat{w}_{2,2} & \widehat{w}_{2,3} & \dots & \widehat{w}_{2,n} \\ \widehat{w}_{3,1} & \widehat{w}_{3,2} & \widehat{w}_{3,3} & \dots & \widehat{w}_{3,n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \widehat{w}_{m,1} & \widehat{w}_{m,2} & \widehat{w}_{m,3} & \dots & \widehat{w}_{m,n} \end{bmatrix} + \begin{bmatrix} \hat{b}_1 & \hat{b}_2 & \hat{b}_3 & \dots & \hat{b}_n \end{bmatrix}$$

Tal que:

- $\hat{Z}$ é uma matriz em que cada linha é uma representação de um exemplo $x$ (ou seja, cada linha possui $m$ valores, referentes aos $n$ atributos originais).
- $\widehat{W}$ é uma matriz que contém os pesos com os quais os dados reduzidos de cada exemplo foram reconstruídas. Na função de decodificação, $\widehat{W}$ possuí $n$ colunas (quantidade de neurônios) e $m$ linhas (quantidade de atributos de entrada). Cada coluna contém os $m$ pesos de um neurônio.
- $\hat{B}$ é um vetor de termos bias, sendo que o mesmo contém $n$ colunas que armazena o bias de cada neurônio da camada de saída.

Deve-se aplicar a função de ativação $s$, no resultado da multiplicação de linha da matriz $Z$ pelos elementos da coluna da matriz $\widehat{W}$ e somá-los, acrescentando a essa soma o bias correspondente. Ao final da função decodificadora $g$, $\hat{X}$ é uma matriz em que cada linha é uma amostra com $n$ atributos reconstruídos.

$$\hat{X} = \begin{bmatrix} \hat{x}_{1,1} & \hat{x}_{1,2} & \hat{x}_{1,3} & \dots & \hat{x}_{1,n} \\ \hat{x}_{2,1} & \hat{x}_{2,2} & \hat{x}_{2,3} & \dots & \hat{x}_{2,n} \\ \hat{x}_{3,1} & \hat{x}_{3,2} & \hat{x}_{3,3} & \dots & \hat{x}_{3,n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \hat{x}_{i,1} & \hat{x}_{i,2} & \hat{x}_{i,3} & \dots & \hat{x}_{i,n} \end{bmatrix}$$

Se o treinamento da rede resultar em uma solução aceitável, de baixo erro, então as camadas de entrada e ocultas funcionam como uma função codificadora que permite que a camada (gargalo), represente somente o conhecimento relacionado às principais características do conjunto de dados de entrada (sinal codificado $Z$).

Para reconstruir uma estimativa do padrão de entrada original $\hat{x}_{ij}$ (ou seja, para executar a decodificação), aplica-se o sinal codificado à camada de saída, que desempenha o papel de um "decodificador". Quanto menor for o tamanho $m$ da camada oculta em comparação com o tamanho $n$ da camada entrada-saída, mais efetiva tende a ser a compressão dos dados (HAYKIN, 2009).

Portanto, um *Autoencoder* é dividido em um *codificador* que é uma função $f(x)$ que transforma um exemplo de entrada para uma representação $z$ de menor dimensão e um *decodificador* que é uma função $g(z)$ que transforma a representação $z$ em sua reconstrução $\hat{x}$.

### 4.3.1 Processo de Treinamento de um Autoencoder

Miranda, Castro e Lima (2012) explicam *Autoencoders* podem ser treinados de forma não supervisionada para aprender características principais de um conjunto de dados de entrada, e essas podem ser úteis para uso posterior em tarefas de aprendizado supervisionado. Um resumo do algoritmo de treinamento para um *Autoencoder* é apresentado no Quadro 2.

**Quadro 2 - Algoritmo de treinamento de um Autoencoder**

| |
|---|
| **ENTRADA DO ALGORITMO** |
| - Conjunto de pontos de dados $X$, no qual cada ponto $x$ (que representa um exemplo de treinamento) com $n$ dimensões, tal que $x = [x_1, x_2, \dots, x_n]$. |
| **PASSO 1: PROPAGAÇÃO DO SINAL** (*FOWARD*) |
| - Para cada entrada *x*:<br>o Propagar o sinal para computar ativações em todas as camadas ocultas e na camada de saída para obter uma saída $\hat{x}$. |
| **PASSO 2: CÁLCULO DO ERRO DE RECONTRUÇÃO** |
| - Calcular o erro de reconstrução de cada reconstrução $\hat{x}$ em relação a entrada $x$ utilizando uma função de perda de informação. |
| **PASSO 3: RETROPROPAGAÇÃO (***BACKWARD***)** |
| - Retropropagar o erro através das camadas da rede e realizar atualizações nos pesos. |
| **PASSO 5: ITERAÇÃO** |
| - Repete-se os passos 1, 2 e 3, até que um determinado critério de parada seja satisfeito. |
| **SAÍDA DO ALGORITMO** |
| - Valores dos componentes da rede (pesos e bias) ajustados. |

**Fonte: Autoria própria**

Para realizar o treinamento de um Autoencoder, é necessário utilizar um algoritmo de otimização, o qual é combinado com o algoritmo de retropropagação (Backpropagation), para realizar o cálculo e a atualização do conjunto de pesos e bias da rede.

#### 4.3.1.1 Otimização

Algoritmos de otimização, também chamados de otimizadores (*optimizers*), tem como objetivo atualizar o conjunto de parâmetros (pesos e bias) de forma iterativa para minimizar uma função de erro (*loss function*), a qual também chamada de função de custo ou de perda de informação (SZALKAI; GROLMUSZ, 2018).

Para a tarefa de otimização, pode-se citar a Descida de Gradiente (*Gradient Descent*) e a Descida de Gradiente Estocástica (*Stochastic Gradient Descent - SGD*), sendo esses os algoritmos mais usados para problemas de otimização (LE, 2015).

A descida de gradiente é utilizada para encontrar um mínimo (local) de uma função. O método utiliza um esquema iterativo, onde para cada passo se toma a direção negativa do gradiente calculado. Pode-se fazer uma analogia desse método como soltar um objeto do topo de uma montanha, de modo em que esse irá descer até determinado ponto e parar. O ponto em que o objeto parou é análogo ao mínimo local (GOODFELLOW; BENGIO; COURVILLE, 2016).

A descida de gradiente estocástica converge mais rápido que a descida de gradiente, pois a otimização da função de custo inicia-se imediatamente a partir da primeira amostra, e não após o conjunto todo como na descida gradiente, porém não é minimizada de maneira tão eficaz como essa. Contudo, a aproximação obtida na descida de gradiente estocástica é suficiente, sendo que os resultados finais são oscilações dos valores ótimos que poderiam ser encontrados na Descida Gradiente (SZALKAI; GROLMUSZ, 2018).

No entanto, pode ser muito difícil otimizar *Autoencoders* usando *backpropagation*, devido aos problemas de dissipação, explosão ou a instabilidade desses algoritmos em relação ao gradiente. Além do que na descida gradiente alguns *datasets* podem ser intratáveis, devido a necessidade do carregamento de todos os dados em memória para cada iteração. Já na descida gradiente estocástica as frequentes atualizações podem tornar o algoritmo sensível aos dados, fazendo com que possíveis ruídos tenham maior impacto no treinamento (GOODFELLOW; BENGIO; COURVILLE, 2016).

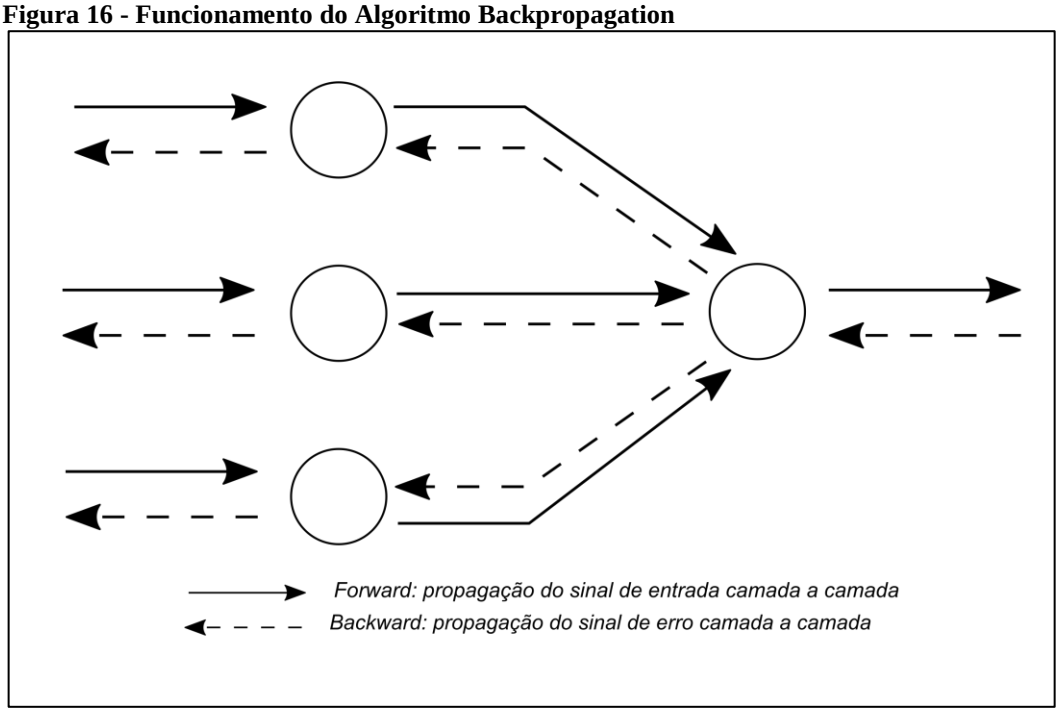
Contudo, existem outras abordagens para otimizar a função de custo que oferecem desempenho superior a descida de gradiente ou de gradiente estocástica. Entre os algoritmos de

otimização disponíveis para o problema de otimização, pode-se citar: *Batch Gradient Descent*; *Mini-Batch Gradient Descent, Momentum, Adagrad, Adadelta, RMSprop, Adam* (GOODFELLOW; BENGIO; COURVILLE, 2016).

#### 4.3.1.2 Retropropagação

O algoritmo de retropropagação (do inglês *backpropagation*) utiliza a entrada e a saída desejada para ajustar os pesos sinápticos e os bias da rede neural por meio de um mecanismo de correção fornecido pelo algoritmo otimizador. O treinamento realizado pelo algoritmo de *backpropagation* é apresentado na Figura 16, ocorre em duas fases distintas conhecidas como *forward* (para frente) e *backward (para trás).*

**Figura 16 - Funcionamento do Algoritmo Backpropagation**




**Fonte: Adaptado de Haykin (2001)**

Na fase *forward*, é definida a saída da rede para um dado padrão de entrada e caracteriza-se pela propagação do sinal de entrada, camada a camada, até a camada de saída. Na fase *backward*, a diferença (erro) entre a saída desejada e a atual é utilizada pela rede para atualizar os pesos de suas conexões, propagando o sinal de erro da camada de saída até a primeira camada oculta (BRAGA; CARVALHO; LUDERMIR, 2000).

Um resumo do algoritmo de retropropagação tradicional é apresentado no Quadro 3, na sequência, utilizando o passo a passo apresentado por Haykin (2001), descreve-se os

principais conceitos e etapas relacionadas ao treinamento de um *Autoencoder* com o algoritmo de retropropagação.

**Quadro 3 - Resumo dos passos realizados pelo algoritmo de retropropagação**

| |
|---|
| **PASSO 1: INICIALIZAÇÃO** |
| ✓ Assumindo que nenhuma informação prévia está disponível, os pesos sinápticos e limiares (bias) são iniciados com valores aleatórios que seguem uma distribuição uniforme. |
| **PASSO 2: APRESENTAÇÃO** |
| ✓ Apresenta-se uma época de exemplos de treinamento à rede. Para cada exemplo do conjunto de treinamento executa-se os passos 3 e 4. |
| **PASSO 3: PROPAGAÇÃO** |
| ✓ Calcula-se os potenciais de ativação e sinal funcional de saída por meio da função de ativação para cada neurônio, camada por camada.<br>✓ Na camada de saída calcula-se o sinal de erro (função de custo ou perda). |
| **PASSO 4: RETROPROPAGAÇÃO** |
| ✓ Calcula-se os gradientes locais para a camada saída e para as demais camadas.<br>✓ Calcula-se o ajuste de pesos sinápticos e limiares da rede.<br>✓ Atualiza-se os pesos sinápticos e limiares das redes camada a camada. |
| **PASSO 5: ITERAÇÃO** |
| ✓ Repete-se os passos 3 e 4, apresentando a rede novos exemplos de treinamento até que um determinado critério de parada seja satisfeito. |
| ✓ Saída: valores dos componentes da rede (pesos e bias) ajustados. |

**Fonte: Adaptado de Haykin (2001).**

Um detalhamento do passo a passo apresentado por Haykin (2001) é descrito abaixo.

- **Passo 1 - Inicialização**

Nesta etapa, os pesos sinápticos e limiares (bias) de cada neurônio da rede, podem ser inicializados com valores aleatórios que seguem uma distribuição uniforme, por exemplo, no intervalo entre [0,1].

- **Passo 2 - Apresentação**

Os exemplos de treinamento são apresentados à rede. Em uma *Autoencoder*, denomina-se época uma apresentação de todos os $i$ exemplos de treinamento no processo de aprendizado da rede. Os exemplos podem ser apresentados de três maneiras segundo Goodfellow, Bengio e Courville (2016):

1) *batch mode*: o tamanho do lote é igual ao conjunto de dados total, tornando os valores de iteração e épocas equivalentes.

2) *mini-batch mode*: o tamanho do lote é maior que um, mas menor que o tamanho total do conjunto de dados. Geralmente, um número que pode ser dividido no tamanho total do conjunto de dados.
3) *stochastic mode*: o tamanho do lote é igual a um. Portanto, o otimizador e os parâmetros da rede neural são atualizados após cada amostra.

O processo de treinamento, que está diretamente ligado com a maneira que os exemplos são apresentados a rede, compreende a propagação do sinal de entrada apresentado até a camada de saída e a retropropagação da correção e atualização dos pesos para o sinal apresentado a rede.

- **Passo 3 - Propagação**

Após a apresentação do sinal de entrada, deve-se calcular os potenciais de ativação (Equação 3) e aplicar uma função de ativação para cada neurônio, camada por camada até a camada de saída.

Na camada de saída de um *Autoencoder*, ao invés de realizar a predição de uma classe, é desejada a reconstrução dos dados de entrada, ou seja, a reconstrução do sinal apresentado à rede. Com as reconstruções obtidas, calcula-se o sinal de erro (função de custo ou perda) a ser retropropagado. Dessa maneira, o treinamento pode ser visto como um exercício de otimização da função de custo.

A função de custo clássica adotada em *Autoencoders* é o erro quadrático médio (*Mean Squared Error - MSE*) , obtido pela Equação (10) (GOODFELLOW; BENGIO; COURVILLE, 2016).

$$MSE = \frac{1}{r}\sum_{i=1}^{r}\sum_{j=1}^{n}(x_{ij} - \hat{x}_{ij})^2 \tag{10}$$

Na qual:

- $r$ é a quantidade de exemplos de treinamento;
- $n$ é a quantidade de atributos (dimensões) das amostras de treinamento;
- $x_i$ é uma amostra de entrada;
- $x_{ij}$ é um dos atributos da entrada de $x_i$;
- $\hat{x}_i$ é uma reconstrução da amostra de entrada $x_i$;

- $\hat{x}_{ij}$ é um dos atributos da reconstrução de $\hat{x}_i$.

O MSE calcula a média dos quadrados dos erros, ou seja, a diferença entre a entrada e a saída para todos os exemplos de treinamento. Os valores de resultado são sempre não negativos e valores próximos a zero tendem a ser melhores.

- **Passo 4 - Retropropagação**

Nesta etapa, primeiramente deve-se calcular o erro (Gradiente Local) de cada neurônio da rede, mas isso depende da camada na qual o neurônio se encontra como mostra Equação 11.

$$\delta_i = \begin{cases} f_s^{'} e_i & se\ neurônio \in c_{saída} \\ f_s^{'} \sum w_{ij}\delta_k & se\ neurônio \in c_{oculta} \end{cases} \quad (11)$$

Para um neurônio da camada de saída, deve se utilizar a Equação (12).

$$\delta i_{c_{saída}} = s(u) * [1 - s(u)] * \ e_i \quad (12)$$

Tal que:

- $\delta i_{c_{saída}}$ é gradiente do $i$-ésimo neurônio da camada de saída;
- $s(u) * [1 - s(u)]$ é a derivada da função de ativação ($f_s'$), em que $u$ é o potencial de ativação (Equação 3) do neurônio e $s$ é a função de ativação do neurônio;
- $e_i$ é o erro de treinamento obtido no $i$-ésimo neurônio da camada de saída**.**

O erro de treinamento $e_i$ de cada neurônio da camada de saída é dado pela Equação (13).

$$e_i = x_i - \ \hat{x}_i \quad (13)$$

Tal que:

- $x_i$ é o $i$-ésimo atributo da entrada apresentada a rede;
- $\hat{x}_i$ é a $i$-ésima reconstrução obtida para o respectivo atributo de entrada $x_i$.

Para calcular o gradiente local dos neurônios das demais camadas, a Equação (14) deve ser utilizada.

$$\delta i_{c_{oculta}} = s(u) * [1 - s(u)] * \sum_{j=1}^{r} \delta j_{c_{ult}} * w_{ij} \quad (14)$$

Em que:

- $\delta i_{c_{oculta}}$ é o gradiente local do $i$-ésimo neurônio da camada;
- $s(u) * [1 - s(u)]$ é a derivada da função de ativação ($f_s'$), em que $u$ é o potencial de ativação e $s$ é a função de ativação do neurônio;
- $\sum_{j=1}^{r} \delta j_{c_{ult}} * w_{ij}$ é somatório do o erro do $j$-ésimo neurônio da camada anteriormente calculada, multiplicado pelo seu respectivo peso;
- $r$ a quantidade de neurônios da camada.

Após encontrar os valores dos gradientes locais para a camada saída e as demais camadas, deve-se calcular o ajuste dos pesos sinápticos e limiares da rede. O valor da correção do erro, para os pesos e bias da rede, que prossegue da camada de saída até a primeira camada oculta é dado pela Equação (15).

$$\Delta w_{ij} = \alpha * x_i * \delta j \quad (15)$$

Em que:

- $\Delta w_{ij}$ será o valor de correção do peso (ou bias) em relação ao erro;
- $\alpha$ é o valor da taxa de aprendizado;
- $x_i$ é o valor de entrada do neurônio;
- $\delta j$ é o valor do gradiente local do $j$-ésimo neurônio da camada.

O valor da taxa de aprendizado $\alpha$ tem forte influência no tempo de convergência da rede. Quando pequena muitos ciclos são necessários para encontrar um bom modelo. Uma taxa elevada, pode provocar oscilações que dificultem a convergência (CARVALHO et al., 2011).

Ao final da etapa de retropropagação, depois de calculado o valor de correção $\Delta w_{ij}$ dos pesos sinápticos e limiares das redes, os mesmos devem ser atualizados camada a camada utilizando a Equação (16).

$$w_{ij} = w_{ij} + \Delta w_{ij} \quad (16)$$

Tal que:

- $w_{ij}$ é o valor do peso;
- $\Delta w_{ij}$ é o valor de correção do peso.

- **Passo 5 - Iteração**

Os passos 3 e 4 devem ser repetidos, apresentando a rede novos exemplos de treinamento até que um determinado critério de parada seja satisfeito. Ao término do algoritmo de retropropagação, o *Autoencoder* treinado tem os valores dos componentes da rede (pesos e bias) ajustados, pelos quais se obtém uma representação reduzida dos dados dos exemplos de entrada, assim como a possibilidade de reconstrução dos exemplos de entrada utilizando as representações reduzidas.

Vale ressaltar que, existem variações da arquitetura e do processo de treinamento aqui apresentados, tais como: *Sparse Autoencoder* (HINTON et al., 2006), *Stacked Autoencoder* (BENGIO et al., 2007), *Convolutional Autoencoder, Denoising Autoencoder* (VINCENT et al., 2008), *Variational Autoencoder* (KINGMA; BA, 2014), *Contractive Autoencoder* (RIFAI et al., 2011).

## 4.4 CONSIDERAÇÕES FINAIS DO CAPÍTULO

Neste Capítulo foram apresentados conceitos fundamentais de RNA, aprendizagem profunda e *Autoencoders*. As RNA são um tipo de aprendizagem de máquina baseada no funcionamento do cérebro humano, em que os neurônios artificiais são as estruturas fundamentais e estes formam redes que são organizadas em camadas.

Uma RNA tradicional apresenta somente a entrada conectada a uma camada de saída. Alguns dos métodos mais bem-sucedidos de aprendizagem profunda envolvem RNA que podem possuir várias camadas ocultas conectadas entre as camadas de entrada e saída, e geralmente, utilizam o algoritmo de retropropagação para realizar o treinamento.

*Autoencoders*, utilizados neste trabalho, são redes que não são treinadas para predizer alguma classe na sua saída, mas para reproduzir na saída à própria entrada. As camadas ocultas de uma rede *Autoencoder* são simétricas e estão organizadas em duas estruturas distintas, uma codificadora e uma decodificadora.

As camadas da estrutura codificadora são utilizadas como extratoras de características, e por meio destas características, uma representação reduzida da entrada é obtida na camada *bottleneck* (gargalo). A estrutura decodificadora é responsável por reconstruir os dados de entrada, utilizando a representação reduzida, e então apresentá-los na camada de saída.

Portanto, um *Autoencoder* possui uma função de codificação e uma função de decodificação. Para realizar o treinamento, é necessário utilizar um algoritmo para otimizar uma função custo, ou seja, uma função de distância entre a diferença da representação reconstruída e a sua respectiva entrada, utilizando o algoritmo de retropropagação para calcular e atualizar os pesos e bias da rede. Dessa maneira, o treinamento nesse tipo de rede pode ser entendido como um exercício de otimização da função de custo.

# 5 REVISÃO SISTEMÁTICA DA LITERATURA

Este Capítulo apresenta o levantamento bibliográfico visando estabelecer o estado da arte atual a respeito das técnicas de redução de dimensionalidade em classificação hierárquica multirrótulos, dando enfoque na extração de atributos. A Seção 5.1 aborda o método utilizado para realização da revisão sistemática. A Seção 5.2 define as questões de pesquisa e apresenta as bases de pesquisa utilizadas nas buscas. A Seção 5.3 define as palavras-chaves utilizadas na pesquisa. A Seção 5.4 descreve o resultado das buscas e a Seção 5.5 apresenta a filtragem realizada sobre eles. A Seção 5.6 explica quais foram os critérios de ordenação aplicado nos resultados encontrado e por fim, a Seção 5.7 apresenta os resultados obtidos com o mapeamento sistemático. A Seção 5.8 contempla as considerações do Capítulo.

## 5.1 DESCRIÇÃO DO MÉTODO DE MAPEAMENTO SISTEMÁTICO

Para a realização do levantamento bibliográfico foi utilizado o método de revisão sistemática *Methodi Ordinatio* proposto por Pagani, Kovaleski e Resende (2015). Esse método foi adotado por apresentar uma etapa em que os trabalhos são classificados em um ranking de importância, ao equacionar os três fatores mais importantes a serem considerados na busca de um artigo científico: o ano de publicação do artigo; o número de citações e o fator de impacto (JCR - *Journal Citation Reports* ou SJR - *Scientific Journal Rankings*).

A ordem de importância baseia-se em um cálculo em que são considerados o fator de impacto, o ano de publicação e o número de citações, permitindo a diminuição e refinamento dos artigos a serem lidos na íntegra (PAGANI; KOVALESKI; RESENDE, 2015). A Figura 17 apresenta as etapas do método de revisão sistemática utilizado nesse estudo.

Na primeira etapa deve-se definir a intenção de pesquisa, delimitar o tema a ser trabalhado e criar as questões de pesquisa a serem respondidas pela revisão sistemática. Também são definidas as características (ano de publicação, tipo de trabalho, idioma da publicação etc.) de um trabalho para que este seja considerado na revisão sistemática.

Com o estabelecimento do tema da pesquisa, deve-se realizar uma busca preliminar exploratória com palavras-chave relacionadas ao tema, com o intuito de validar o resultado de busca obtido e verificar a necessidade, assim como as possibilidades, de alteração e/ou combinação delas.

**Figura 17 - Etapas da Revisão**

**Fonte: Adaptado de Pagani, Kovaleski e Resende (2015)**

Esse processo é realizado na segunda etapa, juntamente com a definição das bases de dados, nas quais as buscas serão realizadas, sendo que estas devem ser compatíveis com a intenção e o tema da pesquisa.

Na terceira etapa, pode-se utilizar um gerenciador de bibliografia, para facilitar o controle dos resultados das buscas realizadas nas bases de dados com as palavras-chave definidas.

Com os resultados obtidos, na quarta etapa, realizadam-se as primeiras filtragens. São descartados os resultados duplicados, os trabalhos em que o título, resumo ou palavras-chave não tem relação com o tema especificado na etapa 1 e outros filtros definidos pelo pesquisador, também devem ser aplicados nessa etapa.

Após a filtragem, deve-se realizar a identificação do ano, número de citações e fator de impacto, por meio dos índices JCR (*Journal Citation Reports*) ou SJR (*Scientific Journal Rankings*), de todos os trabalhos. A ordenação dos resultados é realizada na quinta etapa e ocorre por meio da equação *InOrdinatio*, definida por Pagani, Kovaleski e Resende (2015), apresentada na Equação (17).

$$InOrdinatio = (Fi/1000) + \alpha * [10 - (Aa - Ap)] + Ci \quad (17)$$

Em que:

- $Fi$ representa o Fator de impacto (índice JCR ou SJR);
- $Aa$ o ano atual em que a revisão sistemática está sendo realizada;
- $Ap$ o ano da publicação do artigo e $Ci$ o número total de citações do artigo;

- $\alpha$ é um número no intervalo entre 1 e 10 que representa quão importante um estudo novo é em relação a um mais velho, sendo 1 pouca importância e 10 mais importância.

A última etapa é a busca dos artigos selecionados no formato completo. Se o artigo for localizado, realiza-se a leitura e análise sistêmica, caso contrário deve-se descartá-lo. O objetivo da leitura dos artigos na íntegra, é verificar como o trabalho ranqueado responde às perguntas definidas na etapa 1.

## 5.2 QUESTÃO DE PESQUISA

O mapeamento sistemático realizado nesse trabalho, tem por objetivo identificar trabalhos que apresentem métodos que reduzem a dimensionalidade dos atributos em classificação hierárquica multirrótulo. Tem como foco identificar características nos métodos que possibilitem propor um novo método de extração de atributos ou melhorias em métodos já existentes utilizados na classificação hierárquica multirrótulo, utilizando conceitos e técnicas de *Deep Learning*. Foram definidas perguntas que cada trabalho selecionado deve responder, sendo as mesmas elencadas no Quadro 4.

**Quadro 4 - Questões de Pesquisa**

| | |
|---|---|
| **P1** | Quais são as abordagens e/ou técnicas de Redução de Dimensionalidade (seleção ou extração) utilizadas na Classificação Hierárquica Multirrótulo? É uma abordagem/técnica de Aprendizagem Profunda? |
| **P2** | Quais foram as áreas em que a Redução de Dimensionalidade foi realizada? Qual o formato da estrutura hierárquica das classes? (*Tree* ou DAG) |
| **P3** | Qual foi a contribuição científica dos autores no trabalho? (criação de novo método ou aprimoramento de método existente) |
| **P4** | O resultado obtido pelo conjunto de dados reduzido, foi superior ao conjunto de dados formado por todos os atributos? |

**Fonte: Autoria própria**

O mapeamento sistemático foi realizado pelo autor do trabalho e sua orientadora. Foi limitado para trabalhos que tenham sido publicados entre os anos de 1990 e 2019.

## 5.3 SELEÇÃO DAS BASES E DEFINIÇÃO DOS TERMOS DE BUSCA

O primeiro passo para a realização das buscas é a definição de quais são as bases de dados utilizadas no processo. Com isso, foram escolhidas as bases que atendessem os seguintes

requisitos de filtragem: seleção de publicações na área de Ciência da Computação, artigos na língua inglesa e a possibilidade de realizar consultas usando combinações de palavras-chave.

As bases utilizadas nesse processo de mapeamento sistemático foram selecionadas por meio de uma busca no site da CAPES. O Quadro 5 apresenta a relação das bases de pesquisa e seus respectivos sites.

**Quadro 5 - Definição das bases de pesquisa**

| Base de Pesquisa | Site |
|---|---|
| *ArXiv.org* | <https://arxiv.org/> |
| *Emerald Insight* | <https://www.emeraldinsight.com/> |
| *IEEE* | <https://ieeexplore.ieee.org> |
| *Science Direct* | <https://www.sciencedirect.com> |
| *Scopus* | <https://www.scopus.com> |
| *Springer* | <https://link.springer.com/> |

**Fonte: Autoria própria**

Nessa etapa foram definidas as palavras-chave que estão relacionadas às questões de pesquisa definidas no Quadro 4. Estas foram utilizadas na execução das buscas nas bases de pesquisa, com o objetivo que os resultados retornados sejam os mais precisos dentro do tema pesquisado. As palavras-chave escolhidas são apresentadas no Quadro 6.

**Quadro 6 - Definição das palavras-chave**

| ID | Termo na Língua Inglesa | Termo na Língua Portuguesa |
|---|---|---|
| 1 | *Dimensionality reduction* | Redução de Dimensionalidade |
| 2 | *Deep Learning* | Aprendizagem Profunda |
| 3 | *Feature Extraction* | Extração de Atributos |
| 4 | *Hierarchical Multi-label Classification* | Classificação Hierárquica Multirrótulo |

**Fonte: Autoria própria**

Com a definição das palavras-chave, formou-se algumas combinações entre esses termos, utilizando o operador lógico AND, com o objetivo de refinar os resultados. Definindo essas combinações, obteve-se as *strings* de busca, as quais foram utilizadas nos mecanismos de pesquisa das bases. As strings obtidas estão listadas no Quadro 7.

**Quadro 7 - Definição das strings de busca**

| ID | String de Busca |
|---|---|
| **S1** | *"Hierarchical Multi-label Classification" AND "Dimensionality reduction"* |
| **S2** | *"Hierarchical Multi-label Classification" AND "Feature extraction"* |
| **S3** | *"Hierarchical Multi-label Classification" AND "Deep Learning"* |
| **S4** | *"Hierarchical Multi-label Classification" AND "Autoencoder"* |

**Fonte: Autoria própria**

## 5.4 REALIZAÇÃO DAS BUSCAS

O resultado das buscas nas bases de pesquisa, utilizando as strings de busca definidas no Quadro 7 são apresentadas na Tabela 1.

**Tabela 1 - Total de Resultados das buscas**

| Base de Pesquisa | S1 | S2 | S3 | S4 | Busca Inicial |
|---|---|---|---|---|---|
| ArXiv.org | 00 | 00 | 00 | 00 | 0 |
| Emerald | 00 | 00 | 00 | 00 | 0 |
| IEEE | 00 | 02 | 01 | 01 | 4 |
| Science Direct | 01 | 08 | 05 | 00 | 14 |
| Scopus | 00 | 08 | 01 | 01 | 10 |
| Springer | 04 | 09 | 08 | 03 | 24 |
| Total | 05 | 27 | 15 | 05 | 52 |

**Fonte: Autoria própria**

Ao final dessa etapa foram encontrados 52 trabalhos e estes foram importados para o gerenciador de bibliografia *Zotero*. Como alguns trabalhos podem ser encontrados em buscas de diferentes bases e podem não estar relacionados diretamente com tema pesquisado, a aplicação da etapa de filtragem é um processo imprescindível.

## 5.5 PROCEDIMENTOS DE FILTRAGEM

O processo de filtragem dos resultados obtidos por meio da busca nas bases de dados foi realizado em quatro etapas:

1) Exclusão de artigos duplicados pela junção dos resultados.
2) Eliminação de todas as publicações de livros e capítulos de livros.
3) Separação dos resultados nas seguintes categorias: artigos e conferências.
4) Exclusão de artigos que não possuem relação com o tema do trabalho, levando em consideração os seus títulos, resumos e palavras-chave.

Para a realização das etapas 1 e 2 foi criado um único repositório na ferramenta *Zotero* e feita a junção dos resultados obtidos em todas bases de pesquisa com as strings. Com isso, aplicou-se as etapas de filtragem 1 e 2, retornando os resultados observados na Tabela 2.

**Tabela 2 - Aplicação das etapas 1 e 2 no processo de filtragem**

| | Busca Inicial | Filtragem 1 e 2 |
|---|---|---|
| Total | 52 | 33 |

**Fonte: Autoria própria**

Na etapa 3, os resultados são separados por categorias, devido ao fato de que o método de revisão sistemática utilizado, realiza a avaliação dos trabalhos de uma forma diferenciada para os artigos e publicações em congressos, por exemplo. A Tabela 3 apresenta a divisão das publicações por tipo.

**Tabela 3 - Aplicação da etapa 3 no processo de filtragem**

| Tipo de publicação | Filtragem 3 |
|---|---|
| Artigos | 29 |
| Conferências | 4 |
| Total | 33 |

**Fonte: Autoria própria**

Na etapa 4, a leitura do título, palavras-chave e resumos dos trabalhos possibilitou identificar e remover aqueles que não tinham relação com o tema desta revisão sistemática. O resultado obtido após a aplicação dessa etapa é exibido na Tabela 4.

**Tabela 4 - Aplicação da etapa 4 no processo de filtragem**

| Tipo de publicação | Filtragem 4 |
|---|---|
| Artigos | 11 |
| Conferências | 1 |
| Total | 12 |

**Fonte: Autoria própria**

## 5.6 ORDENAÇÃO DOS ARTIGOS

Todos os trabalhos resultantes da etapa de filtragem foram ordenados, conforme proposta do *Methodi Ordinatio*, na Tabela 5. Para o cálculo, utilizou-se como Fator de impacto ($Fi$) os valores dos índices JCR, o número de citações $Ci$ obtido por meio do *Google Scholar* (GOOGLE SCHOLAR, 2019) e o valor de $\alpha$ = 10.

**Tabela 5 - Ranking de ordenação dos trabalhos - Periódicos**

| ID | Fi | Ap | Ci | InOrdinatio |
|---|---|---|---|---|
| 1 | 3,965 | 2011 | 105 | 125,0040 |
| 2 | 3,814 | 2019 | 9 | 109,0038 |
| 3 | 4,151 | 2019 | 2 | 102,0042 |
| 4 | 3,907 | 2019 | 0 | 100,0039 |
| 5 | 1,855 | 2018 | 2 | 92,0019 |
| 6 | 3,034 | 2017 | 4 | 84,0030 |
| 7 | 2,882 | 2017 | 0 | 80,0029 |
| 8 | 1,820 | 2012 | 43 | 73,0018 |
| 9 | 1,600 | 2014 | 12 | 62,0016 |
| 10 | 2,158 | 2014 | 7 | 57,0022 |
| 11 | 1,600 | 2010 | 12 | 22,0016 |

**Fonte: Autoria própria**

Para simplificar a citação dos artigos encontrados nesta busca, na Tabela 5 cada trabalho recebeu um número de identificação (ID), o qual foi utilizado como referência no restante do trabalho. Os autores, periódico e respectivo Qualis, dos artigos citados na Tabela 5 estão são apresentados no Quadro 8.

**Quadro 8 - Descrição dos Artigos de Periódicos selecionados**

| ID | Autores | Periódico | Qualis | Título |
|---|---|---|---|---|
| 1 | Dimitrovski; I.; Kocev; D.; Loskovska; S.; Džeroski; S. | *Pattern Recognition* | A1 | *Hierarchical annotation of medical images* |
| 2 | Melo; André; Paulheim; Heiko | *Artificial Intelligence Review* | A2 | *Local and global feature selection for multilabel classification with binary relevance* |
| 3 | Zou; Z.; Tian; S.; Gao; X.; Li; Y. | *Frontiers in Genetics* | B5 | *mlDEEPre: Multi-functional enzyme function prediction with hierarchical multi-label deep learning* |
| 4 | Gargiulo; F.; Silvestri; S; Ciampi; M.; Pietro; G. De. | *Applied Soft Computing* | A1 | *Deep neural network for hierarchical extreme multi-label text classification* |
| 5 | Colonna; J. G.; Gama; J.; Nakamura; E. F. | *Machine Learning* | A1 | *A comparison of hierarchical multi-output recognition approaches for anuran classification* |
| 6 | Wang; Jingya; Zhu; Xiatian; Gong; Shaogang | *Artificial Intelligence* | A1 | *Discovering visual concept structure with sparse and incomplete tags* |
| 7 | Yan; Shankai; Wong; Ka-Chun | *Journal of Biomedical Informatics* | A2 | *Elucidating high-dimensional cancer hallmark annotation via enriched ontology* |
| 8 | Dimitrovski; I.; Kocev; D.; Loskovska; S.; Džeroski; S. | *Ecological Informatics* | B1 | *Hierarchical classification of diatom images using ensembles of predictive clustering trees* |
| 9 | Slavkov; I.; Karcheska; J.; Kocev; D.; Kalajdziski; S.; Džeroski; S. | *Lecture Notes in Computer Science* | C | *ReliefF for hierarchical multi-label classification* |
| 10 | Chen; J.; Tang; Y. Y.; Chen; C. L. P.; Fang; B.; Lin; Y.; Shang; Z. | *IEEE Transactions on NanoBioscience* | A2 | *Multi-Label Learning With Fuzzy Hypergraph Regularization for Protein Subcellular Location Prediction* |
| 11 | Dimitrovski; I.; Kocev; D.; Loskovska; S.; Džeroski; S. | *Lecture Notes in Computer Science* | C | *Detection of visual concepts and annotation of images using ensembles of trees for hierarchical multi-label classification* |

**Fonte: Autoria própria**

O *Methodi Ordinatio* foi criado inicialmente apenas para a ordenação de artigos, então uma adaptação no método para ranquear trabalhos publicados em conferências ou seminários seria necessária. Porém, dos 5 artigos iniciais publicados em conferências, somente um artigo restou após a aplicação da etapa de filtragem, que por estar relacionado com o tema do trabalho, foi mantido e apresentado no Quadro 9.

**Quadro 9 - Descrição dos Artigos em Conferência selecionados**

| ID | Autores | Conferência | Qualis | Título |
|---|---|---|---|---|
| 12 | Cerri; R.; Mantovani; R. G.; Basgalupp; M. P.; Carvalho; A. C. P. L. F. de. | *IJCNN-2018 (International Joint Conference on Neural Networks)* | A1 | *Multi-label Feature Selection Techniques for Hierarchical Multi-label Protein Function Prediction* |

**Fonte: Autoria própria**

Os artigos apresentados nos Quadros 8 e 9 foram localizados e então realizou-se a leitura e análise sistêmica com o objetivo de responder às questões de pesquisa definidas para esta revisão.

## 5.7 RESULTADOS

Nesta etapa do método *Methodi Ordinatio* são apresentadas as respostas obtidas para as perguntas definidas no Quadro 4. Para isso, as perguntas são representadas pelos seus identificadores P1 a P4.

***P1:*** *Quais são as técnicas de Redução de Dimensionalidade (seleção ou extração) utilizadas na Classificação Hierárquica Multirrótulo? É uma abordagem/técnica de Aprendizagem Profunda?*

Com a leitura e análise dos estudos foi possível identificar as técnicas de redução de dimensionalidade utilizadas nos trabalhos selecionados. Na Quadro 10 são apresentados os métodos e as técnicas encontradas.

**Quadro 10 - Métodos e Técnicas de Redução de dimensionalidade**

| ***ID*** | ***Método*** | ***Técnicas*** |
|---|---|---|
| *1* | Extração | *Raw Pixel Representation* (RPR)<br>*Local Binary Patterns* (LBP)<br>*Edge Histogram Descriptors* (EHD)<br>*Scale-Invariant Feature Transform* (SIFT) |
| *2* | Seleção | *Filter - Information Gain* |
| *3* | Extração | *Deep Learning - Convolutional Networks* |
| *4* | Extração | *Deep Learning - Convolutional Networks* |
| *5* | Extração | *Spectral Entropy* |
| *6* | Seleção | *Wrapper - Random Forest* |
| *7* | Seleção | *Filter* - (UDT)<br>*Filter* - UGSS *Coefficient*<br>*Filter* - UNGL *Coefficient* |
| *8* | Extração | *Fourier Descriptors* (FD)<br>*Scale-Invariant Feature Transform* (SIFT) |
| *9* | Seleção | Filter - RelieF |
| *10* | Extração | *Nonnegative Matrix Factorization Framework* (NMF) |
| *11* | Extração | *Scale-Invariant Feature Transform* (SIFT)<br>*RGBHist*<br>*Gist* |
| *12* | Seleção | *Wrapper* - Clus-HMC |

**Fonte: Autoria própria**

No Quadro 11, pode-se observar os métodos e técnicas dos trabalhos que fazem parte desta revisão. São 5 trabalhos que utilizam técnicas de seleção de atributos e 7 que usam a

extração de atributos, foco deste trabalho. No quadro 11 visualiza-se melhor o cenário levantado, em que os trabalhos são apresentados pelo método e a quantidade de vezes que foram citadas nos trabalhos.

**Quadro 11 - Métodos e Técnicas de Redução x Quantidade**

| IDs | Método | Técnica | Qnt. | IDs |
|---|---|---|---|---|
| 2, 6, 7, 9, e 12 (5 trabalhos) | Seleção | *Filter - UGSS Coefficient* | 1 | 7 |
| | | *Filter - Information Gain* | 1 | 2 |
| | | *Filter - UDT* | 1 | 7 |
| | | *Filter - UNGL Coefficient* | 1 | 7 |
| | | *Filter - ReliefF* | 1 | 9 |
| | | *Wrapper - Clus-HMC* | 1 | 12 |
| | | *Wrapper - Random Forest* | 1 | 6 |
| 1, 3, 4, 5, 8, 10 e 11 (7 trabalhos) | Extração | *Deep Learning - Convolutional Networks* | 2 | 3 e 4 |
| | | *Edge Histogram Descriptors (EHD)* | 1 | 1 |
| | | *Fourier Descriptors (FD)* | 1 | 8 |
| | | *Local Binary Patterns (LBP)* | 1 | 1 |
| | | *Nonnegative Matrix Factorization Framework (NMF)* | 1 | 10 |
| | | *Raw Pixel Representation (RPR)* | 1 | 1 |
| | | *RGBHist* | 1 | 11 |
| | | *Gist* | 1 | 11 |
| | | *Scale-Invariant Feature Transform (SIFT)* | 3 | 11 |
| | | *Spectral Entropy (SE)* | 1 | 1, 8 e 11 |

**Fonte: Autoria própria**

Pode-se verificar, por meio dos dados apresentados, 17 técnicas de redução de dimensionalidade em HMC, sendo 7 técnicas de seleção de atributos e 10 de extração de atributos.

Na seleção de atributos observa-se que a abordagem filtro é mais frequente que a abordagem *Wrapper*. Na extração de atributos, são 10 técnicas sendo a mais frequente a SIFT, utilizada como extratora de características em imagens em 3 trabalhos. Observa-se que apenas 2 trabalhos utilizam o Aprendizado Profundo, especificamente redes de estrutura convolucional para extração de novos atributos.

***P2 -*** *Quais foram as áreas em que a Redução de Dimensionalidade foi realizada? Qual o formato da estrutura hierárquica das classes? (Tree ou DAG)*

O Quadro 12 apresenta a distribuição dos trabalhos nas seguintes áreas: Bioinformática, Classificação de Texto, Processamento de imagem, Processamento de vídeo e Reconhecimento Bioacústico.

**Quadro 12 - Método e Técnica x Quantidade de Trabalhos**

| Área | Método | Quantidade | IDs |
|---|---|---|---|
| Bioinformática | Seleção | 03 | 2, 9, 12 |
| | Extração | 02 | 3 e 10 |
| Classificação de Texto | Seleção | 01 | 2 |
| | Extração | 01 | 4 |
| Processamento de Imagem | Seleção | 03 | 6, 7 e 9 |
| | Extração | 03 | 1, 8 e 11 |
| Processamento de Vídeo | Seleção | 01 | 6 |
| Reconhecimento Bioacústico | Extração | 01 | 5 |

**Fonte: Autoria própria**

Pode-se verificar, 5 áreas onde foram aplicadas técnicas de redução de dimensionalidade em HMC, sendo a área de processamento de imagem a mais abordada, em 6 trabalhos, seguida pela bioinformática em 5. Já o processamento de vídeo e reconhecimento bioacústico são as áreas abordadas apenas por 1 trabalho.

No Quadro 13 é apresentada a área de aplicação de cada trabalho e a estrutura hierárquica em que os dados estavam organizados.

**Quadro 13 - Área de aplicação x Tipo de Hierarquia**

| ID | Método | Área | Hierarquia |
|---|---|---|---|
| 1 | Extração | Processamento de Imagem | *Tree* |
| 2 | Seleção | Bioinformática / Classificação de Texto | *(DAG)*Tree* / *Tree* |
| 3 | Extração | Bioinformática | *Tree* |
| 4 | Extração | Classificação de Texto | *Tree* |
| 5 | Extração | Reconhecimento Bioacústico | *Tree* |
| 6 | Seleção | Processamento de Imagem e Vídeo | *Tree* |
| 7 | Seleção | Processamento de Imagem | *Tree* |
| 8 | Extração | Processamento de Imagem | *Tree* |
| 9 | Seleção | Bioinformática /Processamento de Imagem | DAG / Tree |
| 10 | Extração | Bioinformática | *Tree* |
| 11 | Extração | Processamento de Imagem | *Tree* |
| 12 | Seleção | Bioinformática | DAG |

**Fonte: Autoria própria**

Observa-se que a hierarquia mais abordada nos trabalhos é do tipo árvore (Tree), presente em 11 dos 12 trabalhos. É possível observar que a hierarquia estruturada como um

DAG, é abordada somente na área da Bioinformática e em apenas 2 trabalhos. Vale ressaltar que no trabalho de ID 2, a base de dados de bioinformática tem a hierarquia organizada em forma de um DAG, porém, no escopo do trabalho, teve sua estrutura convertida/simplificada para *Tree*, para ser submetida a ferramenta que realiza a HMC.

***P3** - Qual foi a contribuição científica dos autores no trabalho? (criação de novo método ou aprimoramento de método existente)*

Em Dimitrovski et al. (2010), uma abordagem que cria um classificador para detecção de conceitos visuais e rotulação de imagens. A abordagem tem duas etapas: extração de recursos e classificação. A extração de recurso, utiliza diferentes técnicas de extração existentes, que fornecem descrições globais e locais das imagens para prever simultaneamente todos os rótulos da amostra.

Dimitrovski et al. (2011), propõem um novo Classificador HMC global, porém não apresentam uma nova abordagem de extração de atributos, apenas utilizam várias abordagens e combinações de técnicas de extração de atributos, em imagens de raio-x. A extração, no trabalho chamada de pré-processamento, é uma etapa do método proposto.

Dimitrovski et al. (2012), propõe um novo Classificador HMC global, usando duas abordagens e combinações de técnicas de extração de atributos em imagens de diatomáceas. O classificador consiste em duas partes, a etapa de processamento, onde ocorre a extração de atributos, e a etapa de classificação. Além disso, os autores investigam se a combinação dessas técnicas aumenta o desempenho preditivo.

Em Slavkov et al. (2014), foi proposto um novo método de seleção de atributos para HMC, o HMC-ReliefF, que é capaz de identificar recursos mais expressivos nos dados e lidar com a hierarquia de classes como um todo, ou seja, sem a decomposição do problema hierárquico em diversos problemas de classificação plana. Este trabalho realizou uma adaptação do algoritmo ReliefF para o contexto hierárquico multirrótulo.

Chen et al. (2014), geram um modelo de aprendizado multirrótulo hierárquico com regularização dupla do hipergrafo fuzzy, explorando as relações intrínsecas tanto no espaço dimensional dos atributos quanto no espaço das classes (rótulos), utilizando a técnica de extração NMF, para viabilizar a predição.

No trabalho de Yan e Wong (2017), é apresentada uma nova abordagem para a HMC em dados de texto, composta por 3 etapas. A primeira chamada de "NLP pipeline" responsável pela entrada, representação dos recursos e classificação. Para a etapa representação, apresentam

uma nova técnica de seleção de atributos, que aprimoram três técnicas existentes: ganho de informação, GSS *Coefficient* e NGL *Coefficient*. Essas novas técnicas são nomeadas como UDT, UGSS e UNGL (onde U significa "*United*"), respectivamente, que buscam selecionar os atributos mais discriminativos em relação a cada classe (rótulo).

Wang et al. (2017), apresentam uma nova abordagem de seleção, Hierarchical-Multi-Label Random Forest (HML-RF), que se beneficia do mecanismo de seleção de atributos, para manipular dados ruidosos em imagens e vídeo, para rotulação de imagens.

Colonna, Gama e Nakamura (2018) transformam um problema originalmente multiclasse em um problema de classificação hierárquica multirrótulo. A Extração é parte do método proposto e utiliza uma técnica de extração em dados de áudio, a fim de representar a amostra em um espaço de menor dimensão e com características que facilitem a HMC, utilizando a abordagem de Classificador Local por Nó, LCPN (*Local Classifier per Node*).

Cerri et al. (2018), baseados na abordagem *wrapper*, propõe usar o Clus-HMC para a seleção de atributos aplicada em HMC. Os experimentos foram validados em dois classificadores HMC não lineares baseados em redes neurais e algoritmos genéticos: HMC-LMLP e HMC-GA respectivamente.

Em Melo e Paulheim (2019), é realizada uma comparação experimental sistemática entre abordagens de seleção, locais e globais, para a classificação hierárquica plana e multirrótulo baseada na abordagem de transformação.

Zou et al. (2019) apresentam um novo método de HMC, chamado de mlDEEPre, para predição de funções em enzimas. Baseado em aprendizado profundo, para reduzir a dimensionalidade das anotações e das sequências, em um mesmo modelo, que beneficia os resultados na etapa de classificação.

Gargiulo et al. (2019) apresentam uma metodologia denominada HLSE (*Hierarchical Label Set Expansion*), que utiliza uma arquitetura de aprendizado profundo, dedicada para a classificação de texto. O método proposto possui 3 módulos: "*embedding*" responsável por encontrar uma representação adequada do texto de entrada, o modelo de extração de características onde ocorre a redução de dimensionalidade e o módulo responsável pela classificação.

O resumo sobre o cenário apresentado sobre a principal contribuição dos trabalhos analisados é apresentado no Quadro 14.

**Quadro 14 - Artigo x Contribuição científica na HMC**

| ID | Método | Técnicas |
|---|---|---|
| 1 | Extração | Novo Classificador |
| 2 | Seleção | Comparação de técnicas de seleção |
| 3 | Extração | Novo Classificador |
| 4 | Extração | Novo Classificador |
| 5 | Extração | Nova abordagem de Transformação para HMC |
| 6 | Seleção | Novo Classificador |
| 7 | Seleção | Aprimoramento / Novas técnicas de seleção |
| 8 | Extração | Novo Classificador |
| 9 | Seleção | Aprimoramento / Nova técnica de seleção |
| 10 | Extração | Novo Classificador |
| 11 | Extração | Novo Classificador |
| 12 | Seleção | Aprimoramento / Nova técnica de seleção |

**Fonte: Autoria própria**

No Quadro 14 é possível observar que 7 trabalhos apresentam como principal contribuição um novo Classificador Hierárquico Multirrótulo, sendo que 6 destes utilizam o método de extração e apenas 1 a seleção. Vale ressaltar também que novas abordagens/técnicas de redução de dimensionalidade, foco deste trabalho, são apresentadas apenas em 3 trabalhos, e todas estas novas abordagens são técnicas/abordagens de seleção de atributos.

***P4** - O resultado obtido pelo conjunto de dados reduzido, foi superior ao conjunto de dados formado por todos os atributos?*

A abordagem de Dimitrovski et al. (2010) supera alguns dos métodos comparados, porém como alguns resultados foram piores, os autores sugerem a necessidade de reengenharia em alguma etapa da abordagem, principalmente durante a etapa de extração, na qual a redução foi para 10% do número de atributos originais, que podem ter causado perca significativa de informações intrínsecas dos atributos originais.

Em Dimitrovski et al. (2011), os resultados experimentais mostram que o sistema supera a abordagem de SVMs, tanto em termos de erro quanto de eficiência, aos se aplicar vários extratores e combinações entre eles. Os resultados obtidos mostram que essa abordagem é generalizável e facilmente aplicável em diferentes domínios, oferecendo bom desempenho.

Os resultados em Dimitrovski et al. (2012) demonstram que o método proposto tem melhor desempenho preditivo que as abordagens comparadas na classificação de imagens, especificamente, no banco de dados de imagens diatomáceas ADIAC. A classificação é realizada usando as características extraídas e considerando a estrutura hierárquica da taxonomia.

Na avaliação experimental de Slavkov et al. (2014), foram considerados conjuntos de dados de dois domínios proeminentes para HMC - genômica funcional e anotação de imagem. Os resultados mostram que o HMC-ReliefF consegue identificar os recursos relevantes presentes nos dados nos dois domínios, sendo que obteve melhor desempenho na classificação de imagens.

Em Chen et al. (2014), os resultados experimentais em seis conjuntos de dados de referência de proteína, demonstram a superioridade do método proposto em comparação com outros métodos e ilustram o benefício da exploração de correlações entre atributos e rótulos.

Os resultados em Wang et al. (2017) demonstram que o HML-RF se beneficia do mecanismo de seleção de características que é inerente à floresta aleatória (*Random Forest*) para lidar com dados ruidosos, selecionando as informações mais discriminativas, que evitam a "maldição" de alta dimensionalidade, pois recursos visuais são frequentemente representados em um espaço de alta dimensionalidade, assim, é possível selecionar características mais discriminativas para a predição com precisão semântica.

Em Yan e Wong (2017), a estratégia de seleção apresentada, especialmente projetada para o problema multirrótulo, foi capaz de reduzir a dimensionalidade dos dados, preservando os recursos mais informativos, filtrando ruídos e consequentemente melhorando a eficácia da predição. Comparada com outros métodos e avaliada por uma multiplicidade de métricas de desempenho, demonstram que a abordagem proposta pode revelar novos *insights* sobre cânceres.

A abordagem proposta de transformação de Colonna, Gama e Nakamura (2018), faz uso da extração em áudio, para representar os dados originais em um espaço de menor dimensão. Os resultados demonstram que a abordagem obtém de 70 a 80% de eficiência na HMC.

Em Cerri et al. (2018), a seleção de recursos utilizando o algoritmo Clus-HMC melhorou os resultados em alguns conjuntos de dados, superando métodos populares de seleção. Os experimentos mostraram que o algoritmo baseado em redes neurais, HMC-LMLP, quando alimentado com todos os atributos, forneceu melhores resultados do que quando a redução de atributos foi aplicada. Considerando o classificador que utiliza a abordagem de algoritmos genéticos, HMC-GA, a seleção de recursos com o Clus-HMC melhorou os resultados em alguns conjuntos de dados.

Em Melo e Paulheim (2019), os resultados comparativos mostram que a abordagem de seleção de atributos local supera a seleção de recursos globais em termos de precisão de classificação, sem inconvenientes no desempenho e no tempo de execução.

Os experimentos de Zou et al. (2019) demonstraram que o mlDEEPre, como técnica de aprendizagem profunda, valendo-se da extração de características obtida na sua estrutura convolucional, supera outros métodos na predição da função enzimática.

A metodologia HLSE de Gargiulo et al. (2019), é capaz de fornecer um aumento de desempenho com o conjunto de dados reduzido, ao criar um espaço vetorial mais simples, tornando o classificador capaz de extrair melhor os recursos importantes do texto relacionado aos rótulos de saída.

Vale ressaltar que motivado pelos recentes avanços nos modelos de aprendizagem profunda, Cerri et al. (2018), pretendem investigar se, técnicas como máquinas Restritas (RBMs) e *Autoencoders* (foco deste trabalho), podem ser usadas para extrair com sucesso representações de características que diminuam a dimensionalidade dos dados e melhorem o desempenho da tarefa de HMC.

## 5.8 CONSIDERAÇÕES DO CAPÍTULO

Neste Capítulo foram apresentados os trabalhos correlatos encontrados por meio do mapeamento sistemático de estudos no tema desta pesquisa. Essa revisão permitiu com que fossem identificados, avaliados e interpretados os artigos de periódicos disponíveis e relevantes para a questão de pesquisa definida.

Após o levantamento bibliográfico foi verificado que os trabalhos revisados propõem novos Classificadores Hierárquico Multirrótulo, e que a redução de dimensionalidade é apenas utilizada no pré-processamento destes Classificadores, sendo que a apresentação de novos métodos de redução de dimensionalidade para a área de HMC, ocorre apenas em 3 trabalhos desta revisão e utilizam o método de seleção de atributos.

Em relação a utilização da extração de atributos por meio de técnicas de aprendizagem profunda na HMC, apenas dois trabalhos exploram a potencialidade dessa abordagem para a extração de características e consequente redução de dimensionalidade de dados apresentando, porém, novos classificadores.

Dessa maneira pretende-se investigar a utilização de modelos de aprendizagem profunda para extrair características, representadas em um espaço de menor dimensão, que melhorem o desempenho da HMC.

# 6 EXTRAÇÃO DE ATRIBUTOS BASEADA EM AUTOENCODERS PARA A CLASSIFICAÇÃO HIERÁRQUICA MULTIRRÓTULO (FEAE-HMC)

Este Capítulo apresenta o método proposto para realizar a redução da dimensionalidade dos atributos em dados hierárquicos multirrótulo utilizando *Autoencoders*. Para tanto, a Seção 6.1 trata do funcionamento geral da proposta com a descrição das etapas. A Seção 6.2 aborda as etapas do método FEAE-HMC (*Feature Extraction based on Autoencoders for the Hierarchical Multi-label Classification*). Já a Seção 6.3 relata as considerações finais do Capítulo.

## 6.1 VISÃO GERAL

O método proposto para a redução de dimensionalidade, utiliza a abordagem de extração de atributos, que a partir de transformações/combinações, lineares ou não-lineares, dos atributos originais, busca criar atributos mais expressivos e que melhor representem a variabilidade dos dados originais.

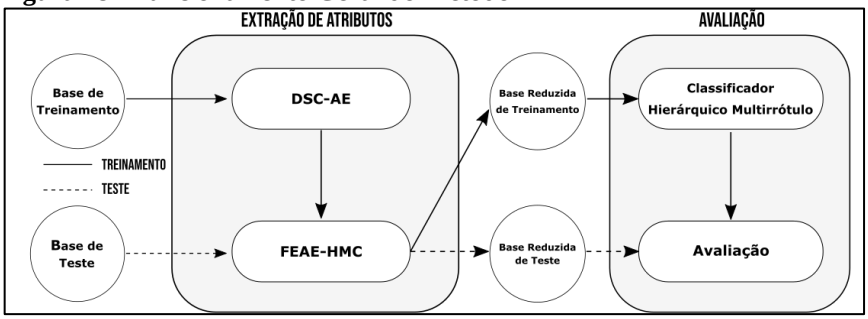
Nesse tipo de abordagem, a redução de dimensionalidade acontece ao se utilizar todos os atributos originais, de dimensão $n$, para projetar um espaço de menor dimensão $m$, em que o máximo de informações dos dados originais é preservada. A redução ocorre ao se projetar $m$, em que $m < n$.

Como alternativa para a tarefa de redução, utiliza-se a aprendizagem profunda (*Deep Learning*) por meio da criação de uma rede auto codificadora (*Autoencoder*), que com sua estrutura pode realizar o mapeamento de identidade (extração de características) em suas camadas ocultas.

Assim, busca-se obter uma representação $m$ da dimensionalidade intrínseca de um conjunto de dados hierárquicos multirrótulo. A Figura 18 exibe de maneira simplificada, as etapas e o funcionamento geral do método.

O método FEAE-HMC (*Feature Extraction based on Autoencoders for the Hierarchical Multi-label Classification*) ilustrado na Figura 18 tem como requisito de entrada uma base de dados hierárquica multirrótulo, sendo que esta deve estar dividida em dois arquivos: treinamento e teste.

**Figura 18 - Funcionamento Geral do Método**




**Fonte: Autoria própria**

A base de treinamento, composta pelos atributos e respectivos rótulos, é pré-processada para fornecer informações na definição da estrutura de uma rede baseada em *Autoencoders*, porém com algumas adaptações e estratégias propostas, para o método FEAE-HMC.

O *Autoencoder* com adaptações, denominado DSC-AE (*Double Shared Cross - Autoencoder*), tem duas entradas, uma para os atributos e outra para os rótulos da base de treinamento. Para essas entradas, a rede compartilha 5 das suas camadas ocultas durante o treinamento.

Na rede DSC-AE, as saídas esperadas na última camada são cruzadas, diferentemente da abordagem tradicional em que se espera a reconstrução dos dados de entrada. Pode-se dizer então que, para a entrada que recebeu os atributos, espera-se na saída a reconstrução dos rótulos. Ao mesmo tempo que para a entrada que recebeu os rótulos espera-se na saída a reconstrução dos atributos.

Por meio do compartilhamento dos pesos e bias das camadas ocultas e da inversão das saídas esperadas, espera-se que essa estrutura de rede possa ser capaz de capturar correlações entre os atributos e os rótulos de cada amostra, durante o treinamento e que possa refletir positivamente durante a etapa de redução de atributos.

Ao final do treinamento da rede DSC-AE com os dados da base de treinamento, cria-se uma nova rede, denominada FEAE-HMC, composta pela:

- Estrutura Codificadora da rede DSC-AE, que recebeu como entrada os atributos, e os respectivos pesos e bias.
- Estrutura Decodificadora da rede DSC-AE, que reconstruiu os atributos na camada de saída, e seus respectivos pesos e bias.

As bases de treinamento e teste tem a dimensionalidade dos atributos reduzida, utilizando a estrutura codificadora da rede FEAE-HMC, que deve fornecer representações que agregam correlações entre os atributos e respectivos rótulos.

Nas bases reduzidas são acrescidos os rótulos originais de cada amostra, e então são submetidas a um Classificador Hierárquico Multirrótulo, gerando um modelo, em que se obtêm o desempenho preditivo da tarefa de HMC. Espera-se que este desempenho seja equivalente ou superior ao desempenho preditivo da base original.

## 6.2 FUNCIONAMENTO

Para melhor entendimento do funcionamento do método FEAE-HMC, a descrição do processo será feita utilizando uma base de dados de classificação hierárquica multirrótulo fictícia, sendo está representada na Tabela 6.

**Tabela 6 - Base de dados fictícia**

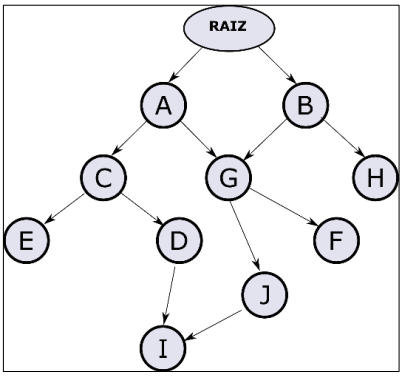
| | $x_1$ | $x_2$ | $x_3$ | $x_4$ | $x_5$ | $x_6$ | $x_7$ | $x_8$ | $x_9$ | CLASSE |
|---|---|---|---|---|---|---|---|---|---|---|
| **E1** | 0,45 | 0,42 | 0,56 | 0,92 | 0,95 | 0,34 | 0,21 | 0,88 | 0,16 | C@E@I |
| **E2** | 0,84 | 0,37 | 0,10 | 0,35 | 0,47 | 0,78 | 0,92 | 0,81 | 0,01 | B@H@I |
| **E3** | 0,27 | 0,49 | 0,04 | 0,41 | 0,83 | 0,19 | 0,36 | 0,67 | 0,91 | G |
| **E4** | 0,85 | 0,11 | 0,95 | 0,89 | 0,69 | 0,47 | 0,16 | 0,35 | 0,67 | B@I |
| **E5** | 0,26 | 0,82 | 0,53 | 0,95 | 0,74 | 0,89 | 0,33 | 0,41 | 0,49 | D |
| **E6** | 0,36 | 0,76 | 0,88 | 0,04 | 0,11 | 0,61 | 0,77 | 0,58 | 0,24 | H@J |
| **E7** | 0,67 | 0,49 | 0,06 | 0,54 | 0,35 | 0,01 | 0,63 | 0,19 | 0,16 | A |
| **E8** | 0,08 | 0,93 | 0,20 | 0,36 | 0,76 | 0,05 | 0,58 | 0,88 | 0,90 | B@F |
| **E9** | 0,79 | 0,89 | 0,26 | 0,22 | 0,44 | 0,57 | 0,21 | 0,75 | 0,46 | A@D |
| **E10** | 0,75 | 0,32 | 0,35 | 0,74 | 0,07 | 0,28 | 0,42 | 0,24 | 0,18 | A@D@C@F |

**Fonte: Autoria própria**

Cada linha da Tabela 6 representa uma instância (exemplo), representadas pela letra E seguido de um número identificador sequencial. As colunas representam os atributos (características) sendo representadas pela letra $x$ seguida de um identificador sequencial.

Por fim, a última coluna armazena o atributo classe (rótulo) de cada instância. Como se trata de base de dados multirrótulo, cada classe é separada pelo caractere '@'. A hierarquia das classes base fictícia está representada na Figura 19.

**Figura 19 - Hierarquia das classes**




**Fonte: Autoria própria**

Observado o requisito de entrada, a base fictícia apresentada na Tabela 7, é dividida em dois arquivos, uma base de treinamento e uma de teste.

**Tabela 7 - Bases fictícias de treinamento e teste**

| | | $x_1$ | $x_2$ | $x_3$ | $x_4$ | $x_5$ | $x_6$ | $x_7$ | $x_8$ | $x_9$ | **CLASSE** |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | **E1** | 0,45 | 0,42 | 0,56 | 0,92 | 0,95 | 0,34 | 0,21 | 0,88 | 0,16 | C@E@I |
| | **E2** | 0,84 | 0,37 | 0,10 | 0,35 | 0,47 | 0,78 | 0,92 | 0,81 | 0,01 | B@H@I |
| **Base** | **E3** | 0,27 | 0,49 | 0,04 | 0,41 | 0,83 | 0,19 | 0,36 | 0,67 | 0,91 | G |
| **Treinamento** | **E4** | 0,85 | 0,11 | 0,95 | 0,89 | 0,69 | 0,47 | 0,16 | 0,35 | 0,67 | B@I |
| | **E5** | 0,26 | 0,82 | 0,53 | 0,95 | 0,74 | 0,89 | 0,33 | 0,41 | 0,49 | D |
| | **E6** | 0,36 | 0,76 | 0,88 | 0,04 | 0,11 | 0,61 | 0,77 | 0,58 | 0,24 | H@J |
| | **E7** | 0,67 | 0,49 | 0,06 | 0,54 | 0,35 | 0,01 | 0,63 | 0,19 | 0,16 | A |
| **Base** | **E8** | 0,08 | 0,93 | 0,20 | 0,36 | 0,76 | 0,05 | 0,58 | 0,88 | 0,90 | B@F |
| **Teste** | **E9** | 0,79 | 0,89 | 0,26 | 0,22 | 0,44 | 0,57 | 0,21 | 0,75 | 0,46 | A@D |
| | **E10** | 0,75 | 0,32 | 0,35 | 0,74 | 0,07 | 0,28 | 0,42 | 0,24 | 0,18 | A@D@C@F |

**Fonte: Autoria própria**

Para facilitar a descrição do método, assume-se que a base de treinamento compreende os exemplos de E1 a E6, enquanto a base de teste compreende os exemplos de E7 a E10, como apresentado na Tabela 7.

### 6.2.1 Estratégia para Criação da Estrutura da Rede FEAE-HMC

A etapa detalhada da criação da rede no método FEAE-HMC, utilizando a base fictícia de treinamento, é apresentada na Figura 20. Abaixo segue um detalhamento de cada etapa apresentada. Observa-se que essa é formada por 4 etapas: Pré-Processamento dos Rótulos, Pré-Processamento dos Atributos, Criação da rede DSC-AE e Definição da rede FEAE-HMC.

Ⓐ - Pré-Processamento dos Rótulos

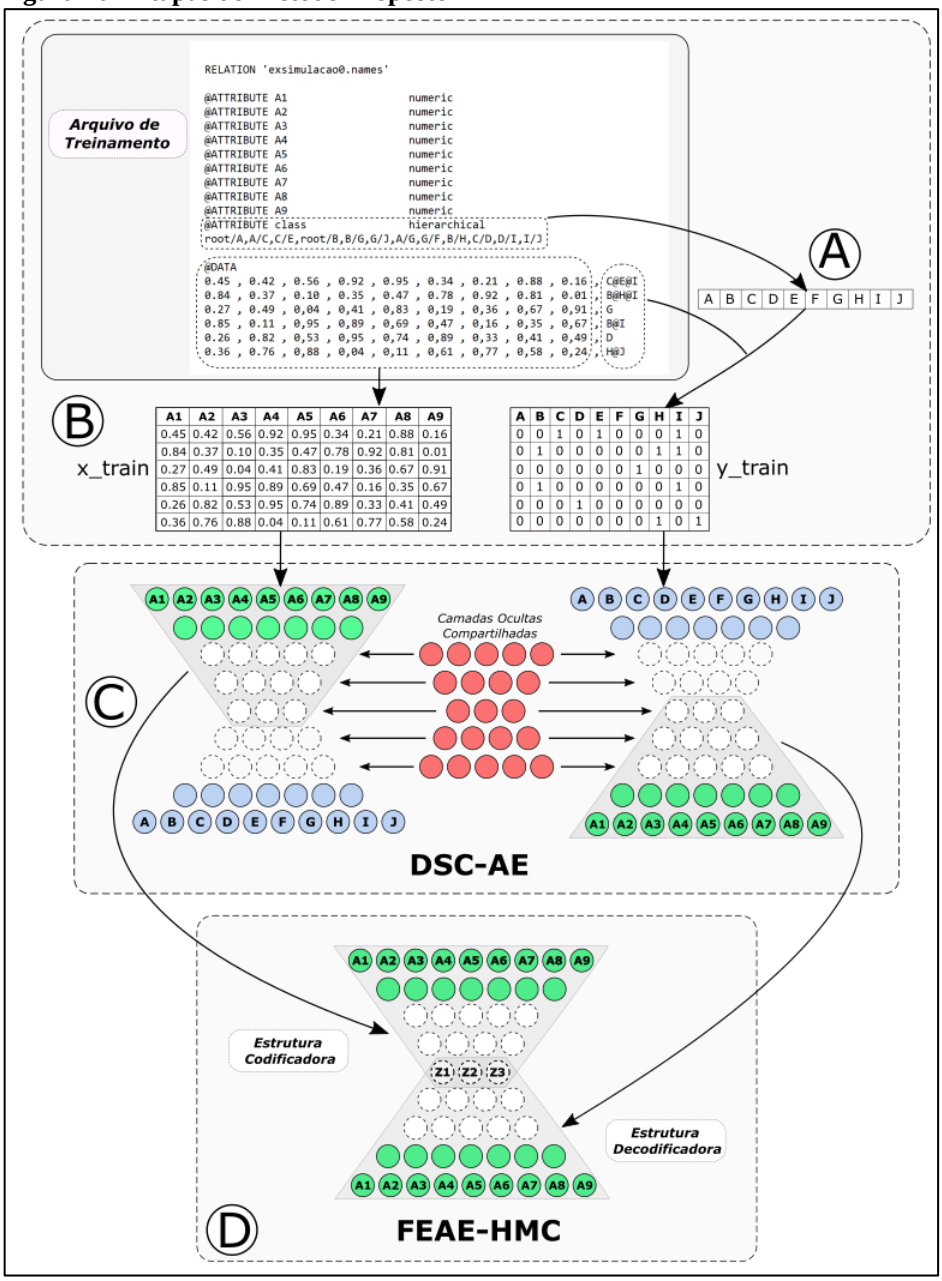
Nesta etapa, observando base de treinamento, ocorre o pré-processamento das informações da base, em um primeiro momento são extraídos todos os rótulos presentes na hierarquia de classes da base. Nesse caso, foram encontradas 10 classes: A, B, C, D, E, F, G, H, I, J. Nesta etapa se define a dimensão de entrada do espaço de rótulos, a qual é identificada pela variável j, nesse caso tem-se j = 10.

Em seguida, observando as classes encontradas, é feita a binarização dos rótulos de todos os exemplos. Uma base denominada *y_train* é gerada, contendo os rótulos no formato binarizado, em relação a hierarquia de classes.

Ⓑ - Pré-Processamento dos Atributos

Nesta etapa, todos os dados referentes aos atributos de cada exemplo são organizados em uma base denominada *x_train*. Define-se a dimensão de entrada do espaço dos atributos a qual é identificada pela variável n, nesse caso temos $n = 9$.

**Figura 20 - Etapas do Método Proposto**

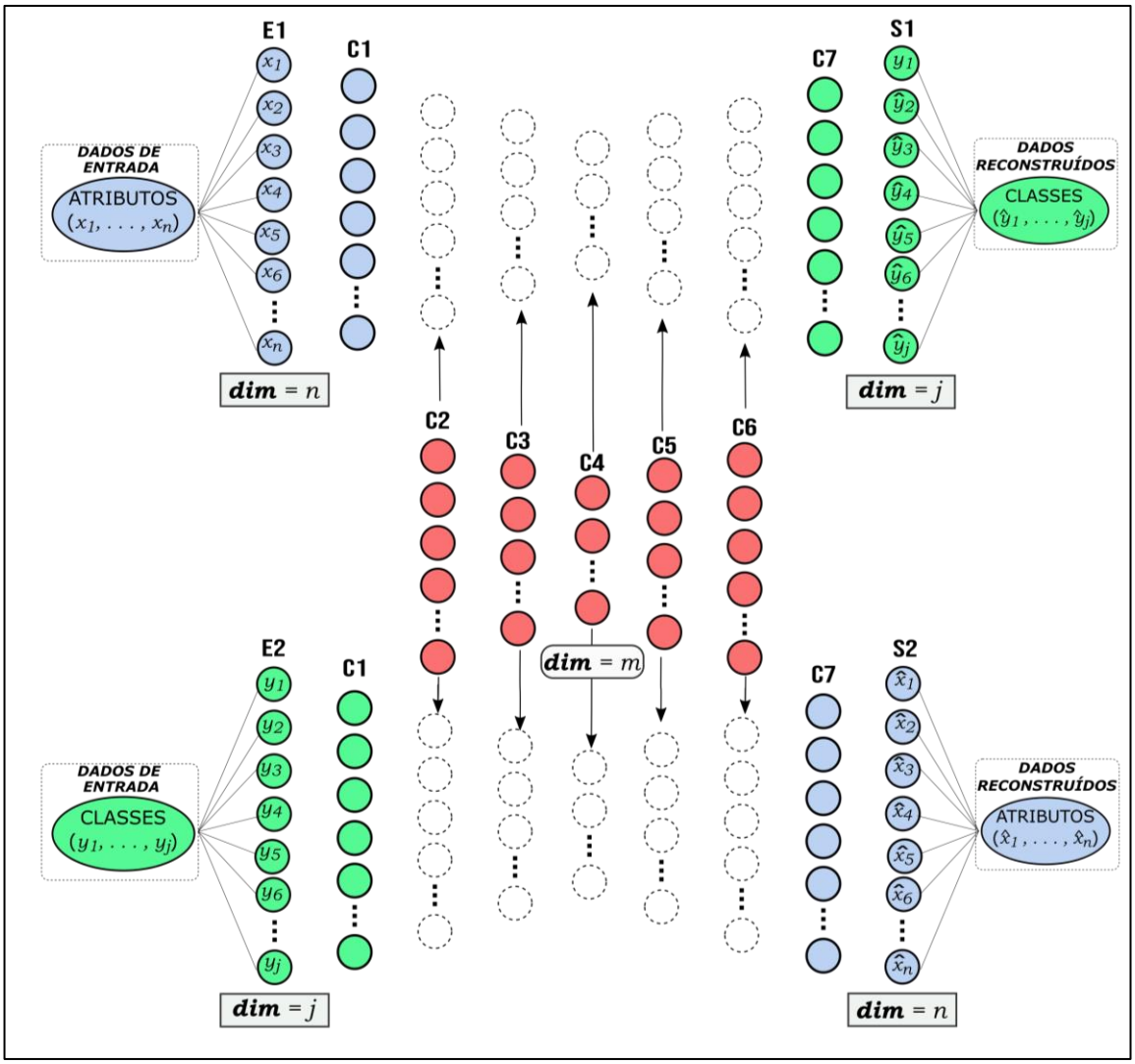



**Fonte: Autoria própria**

Ⓒ - Criação da rede DSC-AE

Nesta etapa, os dados referentes a base de treinamento, *x_train* e *y_train*, são submetidos a uma rede *Autoencoder* com algumas adaptações, denominada DSC-AE, visualizada na Figura 21.

**Figura 21 - Estrutura da rede DSC-AE**




**Fonte: Autoria própria**

A estrutura da rede DSC-AE proposta, possuí 2 camadas de entrada, 7 camadas ocultas para cada uma das entradas e 2 camadas de saída.

Por exemplo, na primeira entrada (**E1**) espera-se os dados referentes aos atributos, *x_train* (dimensão *n*), então temos 7 camadas ocultas (**C1 a C7**) e uma camada de saída (**S1**) na qual espera-se a reconstrução dos rótulos, *y_train* (dimensão *j*).

A segunda entrada (**E2**), recebe os dados referentes aos rótulos, *y_train* (dimensão j), é seguida por 7 camadas ocultas (**C1 a C7**) e na camada de saída (**S2**) espera-se a reconstrução dos atributos, *x_train* (dimensão *n*).

As 7 camadas ocultas nessas redes para cada entrada, tem a mesma quantidade de neurônios. Para isso, define-se o Percentual de Redução desejado para a base apresentada, representada pela variável *Pr*, que é equivalente ao valor de *m*, dimensão da base reduzida obtido na redução.

Com o valor de *Pr*, por meio da Equação (18), encontra-se o Fator de Redução, denominado *Fr*, que é utilizado para obter a razão entre a dimensão de cada camada da rede.

$$Fr = Pr/4 \quad (18)$$

Neste caso, considera-se que a redução desejada para a base fictícia é de 70% da dimensão original dos atributos, ou seja, para *Pr* = 0,7 tem-se:

$$Fr = Pr/4$$
$$Fr = 0{,}7/4$$
$$Fr = 0{,}175$$

Com o valor de *Fr*, é possível encontrar a dimensão adequada de cada uma das 7 camadas ocultas da rede, representadas de **C1** a **C7**, em relação *n*, que é a dimensão dos atributos de entrada. As equações que foram propostas para definir o tamanho de cada camada são observadas no Quadro 15.

**Quadro 15 - Equações para definição do número de neurônios**

| Camada | Equação Utilizada |
|---|---|
| **C1** | *dim = n * (1 - Fr)* |
| **C2** | *dim = n * [1 - (Fr * 2)]* |
| **C3** | *dim = n * [1 - (Fr * 3)]* |
| **C4 = *m* = *Pr*** | *dim = n * (1 - Pr)* |
| **C5** | *dim = n * [1 - (Fr * 3)]* |
| **C6** | *dim = n * [1 - (Fr * 2)]* |
| **C7** | *dim = n * (1 - Fr)* |

**Fonte: Autoria própria**

Nesse caso, para o valor de *Fr* = 0,175 e *n* = 9, a quantidade de neurônios das camadas obtidas com as respectivas equações são as apresentadas no Quadro 16.

**Quadro 16 - Equações e resultados para definição do número de neurônios**

| Camada | Equação Utilizada | Solução | Neurônios |
|---|---|---|---|
| **C1** | *dim = n * (1 - Fr)* | *9 * (1 - 0,175) = 8,25* | 8 |
| **C2** | *dim = n * [1 - (Fr * 2)]* | *9 * [1- (0,175 * 2)] = 5,85* | 6 |
| **C3** | *dim = n * [1 - (Fr * 3)]* | *9 * [1- (0,175 * 3)] = 4,27* | 4 |
| **C4 = *m*** | *dim = n * (1 - Pr)* | *9 * (1 - 0,7) =2,7* | 3 |
| **C5** | *dim = n * [1 - (Fr * 3)]* | *9 * [1- (0,175 * 3)] = 6,3* | 4 |
| **C6** | *dim = n * [1 - (Fr * 2)]* | *9 * [1- (0,175 * 2)] = 7,2* | 6 |
| **C7** | *dim = n * (1 - Fr)* | *9 * (1 - 0,175) =8,1* | 8 |

**Fonte: Autoria própria**

Para facilitar a visualização da estrutura criada nesse caso, o Quadro 17 apresenta de maneira resumida, a estrutura da rede em relação ao número de neurônios obtidos para cada uma das 9 camadas.

**Quadro 17 - Estrutura de Rede em relação ao número de neurônios**

| **DADOS ENTRADA** | **CAMADAS DA REDE** | | | | | | | | | | **DADOS SAÍDA** |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | **ENTRADA** | | **C1** | ***C2*** | ***C3*** | ***C4*** | ***C5*** | ***C6*** | **C7** | **SAÍDA** | |
| *x_train* | **E1** | 9 | 8 | *6* | *4* | *3* | *4* | *6* | 8 | 10 | **S1** | *y_train* |
| *y_train* | **E2** | 10 | 8 | *6* | *4* | *3* | *4* | *6* | 8 | 9 | **S2** | *x_train* |

**Fonte: Autoria própria**

Outro ponto importante do método proposto é o uso de 5 camadas ocultas (C2, C3, C4, C5, C6), compartilham seus pesos durante o treinamento, para ambas entradas, como observado na Figura 21. Esse compartilhamento busca permitir que a rede, ao cruzar as entradas e as saídas, consiga capturar correlações existentes entre os atributos e os rótulos.

Então as bases *x_train* e *y_train* são submetidas para realizar o treinamento da rede DSC-AE, sendo que resumidamente, as configurações e estruturas utilizadas durante o treinamento, nesse trabalho, são as seguintes:

- Topologia: parcialmente conectada, sem presença de retroalimentação.
- Quantidade de camadas: 02 camadas de entrada, 07 camadas ocultas para cada entrada e 02 camadas de saída.
- Quantidade de neurônios: conforme descrito no Quadro 17.
- Neurônios: cada neurônio da rede está associado a pesos e termos bias. Os pesos e bias são iniciados aleatoriamente com valores entre [-1,1].
- Função de ativação: as camadas **C1, C4 e C7** possuem funções de ativação Sigmoide (Equação 4). As demais camadas a função ReLU (Equação 5).
- Otimizador: é utilizado o otimizador *Adam*.
- Função de custo: é utilizado o MSE (Erro Quadrático Médio) (Equação 10), em ambas as saídas.

Dessa maneira, o treinamento ocorre até que uma condição de parada seja satisfeita. Nesse trabalho, condição de parada estabelecida é quantidade de épocas.

Ⓓ - Definição da rede FEAE-HMC.

Após o treinamento da rede DSC-AE, utilizando parte da estrutura e os respectivos pesos e bias, cria-se uma rede para aplicar a redução de dimensionalidade dos atributos, denominada FEAE-HMC, composta pela:

- Estrutura Codificadora da rede DSC-AE, que recebeu como entrada os atributos;
- Estrutura Decodificadora da rede DSC-AE, que reconstruiu os atributos na camada de saída.

Vale ressaltar que na rede FEAE-HMC, é possível fazer o ajuste da reconstrução dos atributos, realizando o processo tradicional de treinamento de um *Autoencoder*, ou seja, que tem como objetivo reconstruir na camada de saída os dados que foram apresentados na camada de entrada.

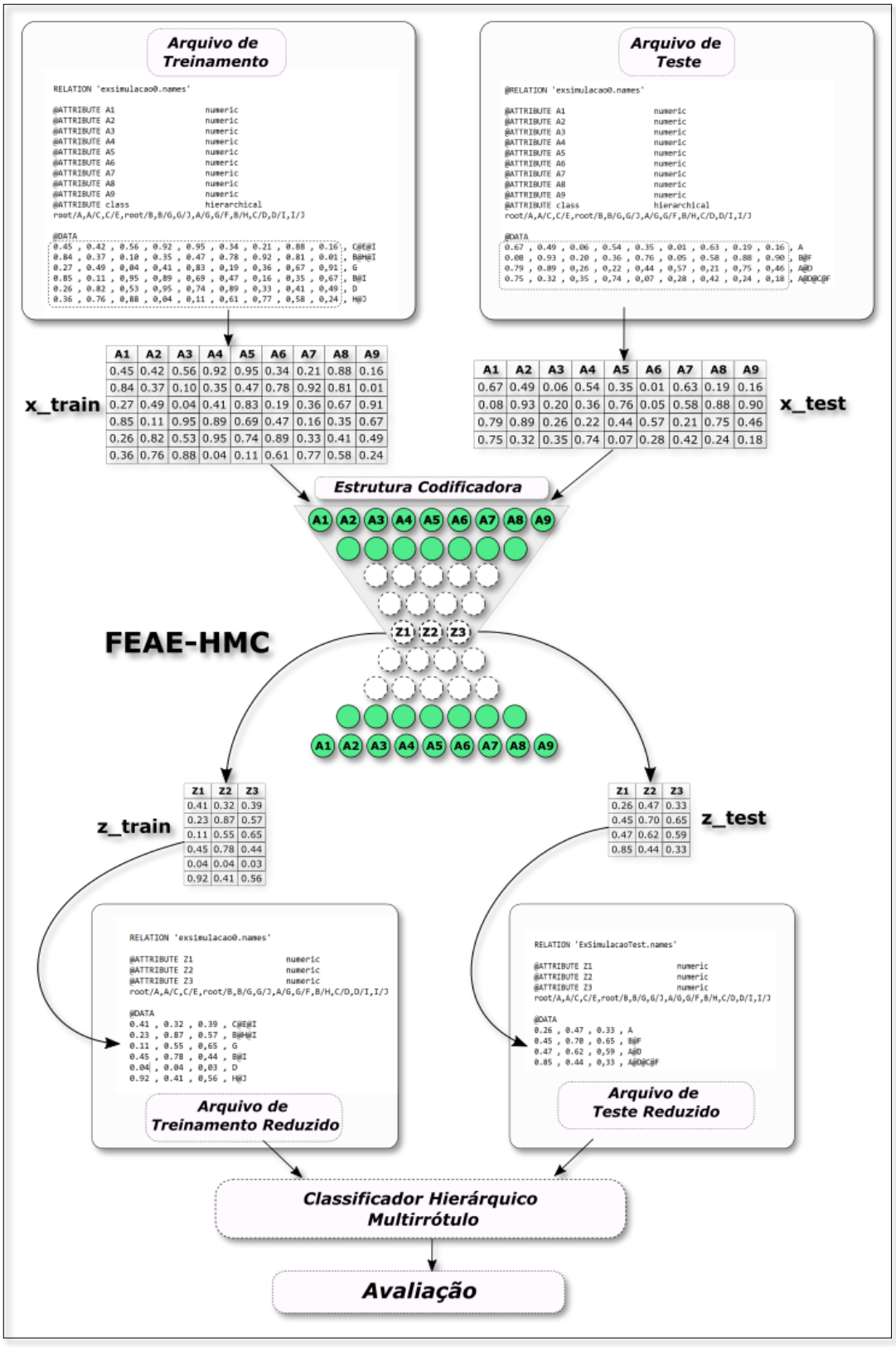
Porém, durante esse processo não é recomendável realizar a atualização dos pesos e bias da estrutura codificadora, pois estes capturam a correlação entre os rótulos e os atributos da base. Optando-se pelo ajuste da reconstrução dos atributos, o mesmo deve ser feito somente nos pesos e bias da estrutura decodificadora, para não interferir na codificação reduzida dos dados.

### 6.2.2 Extração de Características - FEAE-HMC

A rede FEAE-HMC obtida com o processo descrito anteriormente, possui na camada **C4,** a dimensionalidade $m$, que é a dimensionalidade esperada para a base de dados reduzida. A estrutura codificadora dessa rede é utilizada para reduzir a dimensionalidade dos atributos da base hierárquica multirrótulo. O processo de redução é apresentado na Figura 22.

Na Figura 22, observa-se que das bases de treinamento e de teste originais, somente os atributos são utilizados para compor as bases de treinamento (*x_train*) e de teste (*x_test*) que são submetidas a rede FEAE-HMC. A camada **C4,** que tem dimensão igual $m$, é a camada de saída da estrutura codificadora da rede FEAE-HMC. Dessa maneira, os atributos originais com dimensão $n$, são codificados por esta estrutura, disponibilizando uma representação do espaço dos atributos de dimensionalidade $m$ e os dados reduzidos são organizados nas bases *z_train e z_test*, respectivamente.

**Figura 22 - Redução de Dimensionalidade usando a rede FEAE-HMC**




**Fonte: Autoria própria**

As bases *z_train e z_test* tem para cada amostra os valores dos novos atributos obtidos pelo processo de extração de atributos FEAE-HMC, nesse caso, 3 novos atributos (Z1, Z2, Z3),

são a representação de menor dimensão equivalente aos 9 atributos originais (A1, A2, A3, A4, A5, A6, A7, A8, A9). Então para finalizar a composição das novas bases, os rótulos de treinamento e de teste de cada da base são acrescentados a sua respectiva amostra.

Espera-se que as bases reduzidas, *z_train e z_test,* representadas em um espaço de menor dimensão $m$, consigam ter desempenho preditivo equivalente ou superior, quando submetidas a um Classificador Hierárquico Multirrótulo, se comparado com o desempenho obtido pela base original, de dimensão $n$, nesse mesmo classificador

## 6.3 CONSIDERAÇÕES FINAIS DO CAPÍTULO

Neste Capítulo foi apresentado o método desenvolvido, FEAE-HMC. Esse método utiliza como estratégia uma combinação de redes auto codificadoras, para reduzir a dimensionalidade dos atributos em dados hierárquicos multirrótulos. Diferente de outros métodos de redução, o FEAE-HMC leva em consideração os rótulos das amostras para encontrar um espaço de menor dimensão para os atributos.

O método FEAE-HMC pode reduzir a dimensão dos atributos de uma base de dados de HMC, para uma nova representação reduzida qualquer, sendo que os novos atributos podem ser avaliados por qualquer classificador hierárquico multirrótulo, e por consequência, qualquer medida de avaliação de desempenho preditivo. Neste trabalho optou-se pelo classificador Clus-HMC e a medida AUPRC.

# 7 EXPERIMENTOS E RESULTADOS

Neste Capítulo são apresentados os experimentos realizados e os resultados obtidos com o método FEAE-HMC. A Seção 7.1 descreve brevemente as ferramentas utilizadas na implementação do método. A Seção 7.2 apresenta a metodologia usada para o desenvolvimento dos experimentos, descreve os conjuntos de dados e os classificadores utilizados. Na Seção 7.3 são apresentados os resultados obtidos nos experimentos realizados nos classificadores Clus-HMC e MHC-CNN, assim como uma breve comparação dos resultados. Por Fim a seção 7.4 relata as considerações finais do Capítulo.

## 7.1 FERRAMENTAS UTILIZADAS

Para implementação das redes de *Deep Learining* propostas nesse trabalho, optou-se pela utilização do *framework TensorFlow* (1.8) e da API (*Application Programming Interface*) *Keras* (2.2.4), que possibilitam a implementação de modelos simplificados em *Pyhton* (3.6.8).

*TensorFlow* é uma das bibliotecas de código aberto para aprendizado de máquina e pesquisa em redes neurais profundas. O maior benefício é a abstração que ela fornece, permitindo que o desenvolvedor não se preocupe com detalhes básicos de implementação dessas redes, e sim concentre-se na lógica geral da aplicação. Outro ponto positivo é sua flexibilidade, que permite utilizar APIs que facilitam ainda mais a utilização dessa tecnologia.

*Keras* é uma API que possui uma interface simples e otimizada, com funções e comandos que abstraem grande parte do funcionamento do *Tensorflow*. Oferece a construção sequencial de modelos, a qual consiste no empilhamento de camadas sequencialmente conectadas, permitindo uma construção simples e rápida de modelos, com uma pequena quantidade de código. Essa API facilita a criação de modelos de *Deep Learning*, de fácil de implementação e entendimento, quando comparada a utilização somente do *TensorFlow*.

## 7.2 BASE DE DADOS

Os experimentos foram realizados em dados biológicos da área genômica funcional, provenientes do projeto GO (*Gene Ontology*), os quais têm a estrutura hierárquica de classes organizadas na forma de um DAG. Experimentos utilizando tais dados também foram realizados em Vens et. al (2008), Borges (2012), Cerri et al. (2018) e Melo e Paulheim (2019).

Os conjuntos de dados utilizados nos experimentos do método FEAE-HMC, são parte do trabalho de Borges (2012), no qual essas e outras bases foram transformadas, normalizadas e tiveram a imputação de valores faltantes. As principais características dessas bases de dados são apresentadas na Tabela 9.

**Tabela 8 - Características das Bases de Dados GO**

| ID | Nome | Amostras | Atributos | Classes | Classes/Amostra | Amostras/Classe |
|---|---|---|---|---|---|---|
| 1 | Cellcycle | 3751 | 77 | 4126 | 03/28 | 0/785 |
| 2 | Church | 3749 | 27 | 4126 | 03/28 | 0/786 |
| 3 | Derisi | 3719 | 63 | 4120 | 03/28 | 0/781 |
| 4 | Eisen | 2418 | 79 | 3574 | 03/28 | 0/492 |
| 5 | Expr | 3773 | 551 | 4132 | 03/28 | 0/789 |
| 6 | Gasch1 | 3758 | 173 | 4126 | 03/28 | 0/786 |
| 7 | Gasch2 | 3773 | 52 | 4132 | 03/28 | 0/789 |
| 8 | Pheno | 1586 | 69 | 3128 | 03/28 | 0/388 |
| 9 | Seq | 3900 | 478 | 4134 | 03/28 | 0/791 |
| 10 | Spo | 3697 | 80 | 4120 | 03/28 | 0/775 |

**Fonte: Autoria própria**

Com o intuito de facilitar a citação das bases de dados utilizadas, cada base recebeu um código de identificação (ID), o qual foi utilizado como referência na apresentação dos experimentos e resultados.

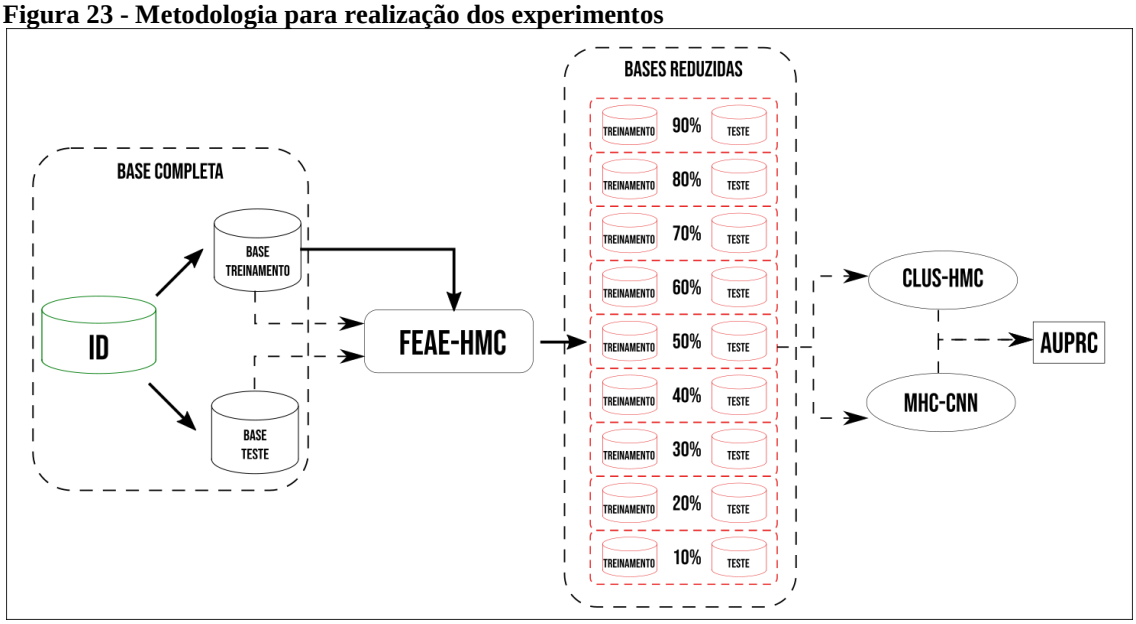
As bases de ID 1, 2, 3, 4, 6, 7 e 10 são dados biológicos de expressão gênica obtidos pela técnica de microarranjo. Já as bases de ID 5, 8 e 9 correspondem respetivamente a: concatenação das bases de dados de expressão gênica; genes relativos ao fenótipo de leveduras mutantes; dados estatísticos da sequência de aminoácidos como: taxa de aminoácidos, tamanho da sequência, peso molecular etc. (BORGES, 2012).

É importante destacar que esses conjuntos de dados, em Borges (2012), foram divididos em conjuntos de treinamento e teste, sendo 2/3 das amostras para o conjunto de treinamento e 1/3 para o teste.

## 7.3 METODOLOGIA UTILIZADA NOS EXPERIMENTOS

A metodologia utilizada para a realização dos experimentos é formada por três etapas principais: obtenção das bases de dados, extração de atributos e classificação hierárquica. A Figura 23 ilustra a metodologia dos experimentos.

**Figura 23 - Metodologia para realização dos experimentos**




**Fonte: Autoria própria**

A base de treinamento do conjunto de dados é utilizada para definir a rede FEAE-HMC. O conjunto de dados, dividido nas bases de treinamento é teste, é submetido ao FEAE-HMC em 9 percentuais de redução nos quais se obtêm as bases reduzidas, de treinamento e teste, equivalentes ao respectivos percentuais. As bases reduzidas são avaliadas em dois classificadores hierárquicos multirrótulos observando-se a medida AUPRC obtida.

O pré-processamento das bases de dados, como mencionado, foi realizado em Borges (2012). A extração de atributos, é realizada pelo método proposto FEAE-HMC. Todos os conjuntos de dados utilizados nos experimentos foram submetidos a esse método.

Para a etapa de treinamento nas redes DSC-AE e FEAE-HMC, alguns parâmetros foram definidos e são descritos na Tabela 8.

**Tabela 9 - Parâmetros das redes DSC-AE e FEAE-HMC**

| **Parâmetro** | **Valor utilizado** |
|---|---|
| Épocas | *1000* |
| Função de Ativação camadas *(E1, E2, C4, S1, S2)* | *Sigmoid* |
| Função de Ativação camadas *(C1, C2, C3, C5, C6, C7)* | *ReLU* |
| Otimizador | *Adam* |
| Função de custo | *MSE* |

**Fonte: Autoria própria**

Os parâmetros relacionados ao número de neurônios na camada de entrada e saída, estão diretamente relacionados a quantidade de atributos e rótulos disponíveis nos conjuntos de dados. Já as quantidades de neurônios nas camadas ocultas, estão diretamente relacionadas a quantidade de atributos da base e a quantidade de atributos da base reduzida a ser alcançada na etapa de extração de atributos.

## 7.4 EXTRAÇÃO DE ATRIBUTOS

Como o objetivo da extração é representar os atributos de uma base de dados de dimensão *n*, para uma nova representação *m*, em que $m < n$ (qualquer valor inteiro menor que *n*), para verificar o desempenho do FEAE-HMC, foram definidos 9 Percentuais de Redução (*Pr*), com o qual se obtém o valor de *m*, para que assim as representações reduzidas obtidas sejam equivalentes ao percentual informado da base original, como observado na Figura 24.

A estrutura da rede DSC-AE, no que diz respeito a quantidade de neurônios em cada uma de suas 7 camadas, é obtida baseada no *Pr* desejado (Equação 16) e conforme equações definidas para cada camada do Quadro 15.

Para exemplificar tal situação, a Tabela 10 demonstra a quantidade de neurônios utilizada nos experimentos na rede DSC-AE, na base CellCycle (ID 1), para cada entrada (E1 e E2), para cada uma das 7 camadas de cada entrada (C1 a C7) e para cada saída (S1 e S2), para os respectivos valores *Pr* estabelecidos.

**Tabela 10 - Número de neurônios por camada na rede DSC-AE para base Cellcycle**

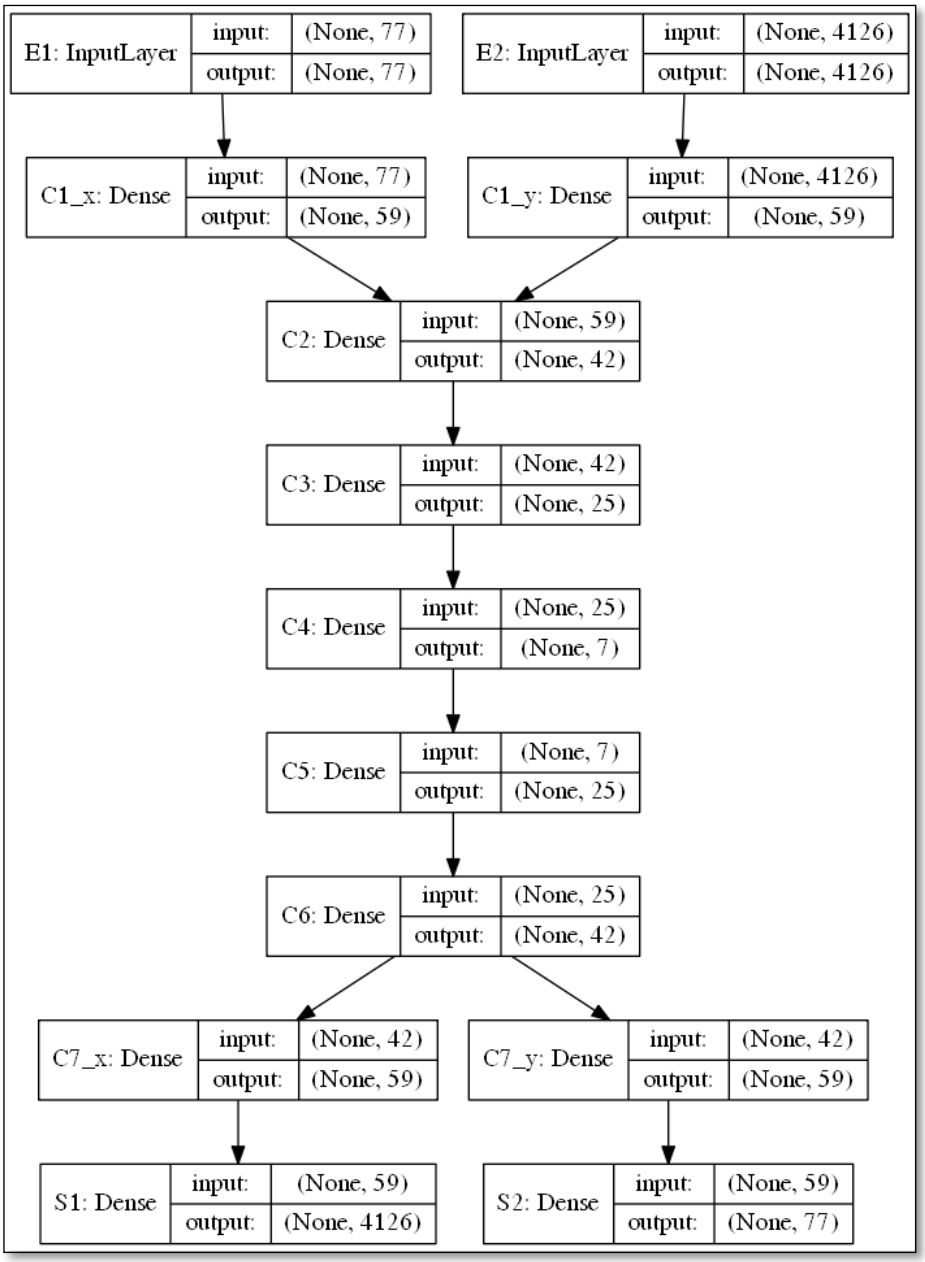
| Pr | Dados Entrada | ENTRADA | | C1 | C2 | C3 | C4 | C5 | C6 | C7 | SAÍDA | | DADOS SAÍDA |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | CAMADAS DA REDE | | | | | | | | | |
| 10% | *x_train* | E1 | 77 | 75 | 73 | 71 | **69** | 71 | 73 | 75 | 4126 | S1 | *y_train* |
| | *y_train* | E2 | 4126 | 75 | 73 | 71 | **69** | 71 | 73 | 75 | 77 | S2 | *x_train* |
| 20% | *x_train* | E1 | 77 | 73 | 69 | 65 | **61** | 65 | 69 | 73 | 4126 | S1 | *y_train* |
| | *y_train* | E2 | 4126 | 73 | 69 | 65 | **61** | 65 | 69 | 73 | 77 | S2 | *x_train* |
| 30% | *x_train* | E1 | 77 | 71 | 65 | 59 | **53** | 59 | 65 | 71 | 4126 | S1 | *y_train* |
| | *y_train* | E2 | 4126 | 71 | 65 | 59 | **53** | 59 | 65 | 71 | 77 | S2 | *x_train* |
| 40% | *x_train* | E1 | 77 | 69 | 61 | 53 | **46** | 53 | 61 | 69 | 4126 | S1 | *y_train* |
| | *y_train* | E2 | 4126 | 69 | 61 | 53 | **46** | 53 | 61 | 69 | 77 | S2 | *x_train* |
| 50% | *x_train* | E1 | 77 | 67 | 57 | 48 | **38** | 48 | 57 | 67 | 4126 | S1 | *y_train* |
| | *y_train* | E2 | 4126 | 67 | 57 | 48 | **38** | 48 | 57 | 67 | 77 | S2 | *x_train* |
| 60% | *x_train* | E1 | 77 | 65 | 53 | 42 | **30** | 42 | 53 | 65 | 4126 | S1 | *y_train* |
| | *y_train* | E2 | 4126 | 65 | 53 | 42 | **30** | 42 | 53 | 65 | 77 | S2 | *x_train* |
| 70% | *x_train* | E1 | 77 | 63 | 50 | 36 | **23** | 36 | 50 | 63 | 4126 | S1 | *y_train* |
| | *y_train* | E2 | 4126 | 63 | 50 | 36 | **23** | 36 | 50 | 63 | 77 | S2 | *x_train* |
| 80% | *x_train* | E1 | 77 | 61 | 46 | 30 | **15** | 30 | 46 | 61 | 4126 | S1 | *y_train* |
| | *y_train* | E2 | 4126 | 61 | 46 | 30 | **15** | 30 | 46 | 61 | 77 | S2 | *x_train* |
| 90% | *x_train* | E1 | 77 | 59 | 42 | 25 | **7** | 25 | 42 | 59 | 4126 | S1 | *y_train* |
| | *y_train* | E2 | 4126 | 59 | 42 | 25 | **7** | 25 | 42 | 59 | 77 | S2 | *x_train* |

**Fonte: Autoria própria**

Na Tabela 10 observa-se a estrutura utilizada na rede DSC-AE para cada um dos 9 percentuais de redução. A quantidade de atributos da base reduzida obtida com o respectivo valor *Pr*, em destaque, é o valor *m* obtido em relação a dimensão original *n*. Já os neurônios das camadas compartilhadas, estão sublinhadas.

A Figura 24, para exemplificar, demonstra como a rede DSC-AE fica estruturada para atingir a redução de 90% dos atributos, para a base de ID 1. No caso, a base será reduzida de 77 atributos para uma nova representação com 7 atributos, na camada oculta C4. Ressalta-se que as camadas C2, C3, C4, C5, C6 são compartilhadas para as entradas E1 e E2.

**Figura 24 - Rede DSC-AE - Redução de 90% - conjunto CellCycle**




**Fonte: Autoria própria**

Todos os experimentos realizados na rede FEAE-HMC, tiveram como condição de parada o número de épocas, nesse caso foi estabelecido o valor 1000. No entanto, optou-se em um primeiro momento, obter conjuntos reduzidos com 100, 200, 300, 400, 500, 600, 700, 800, 900 e 1000 épocas, para todos os percentuais de redução de cada base.

## 7.5 CLASSIFICAÇÃO HIERÁRQUICA

Na etapa de avaliação das bases de dados reduzidas, foram utilizados para a realização dos experimentos os classificadores hierárquicos multirrótulo Clus-HMC e MHC-CNN. Todos os conjuntos de dados foram submetidos a esses classificadores.

Para o classificador Clus-HMC foram utilizados os parâmetros padrões desse algoritmo. Já os parâmetros utilizados para o MHC-CNN foram os descritos em Borges (2012), sendo que a condição de parada foi estabelecida em 50 épocas.

O Clus-HMC foi escolhido devido ao fato de ser frequentemente utilizado em classificação hierárquica multirrótulo, possuindo resultados consolidados na literatura para que posteriormente possa ser realizada uma comparação do desempenho do método de extração de atributos FEAE-HMC. O MHC-CNN foi abordado nos experimentos, tendo em vista que as bases de dados utilizadas nesse trabalho, além de pré-processadas em Borges (2012), também foram avaliadas nesse classificador e no Clus-HMC.

Após a extração de atributos, é realizada a classificação hierárquica nas bases reduzidas de treinamento e teste (somente com o conjunto de novos atributos FEAE-HMC), nos algoritmos Clus-HMC e MHC-CNN. A medida de avaliação utilizada em ambos os classificadores foi a AUPRC.

Os resultados obtidos na etapa de classificação hierárquica foram avaliados utilizando o teste de hipótese de Friedman (FRIEDMAN,1940). Tal teste é indicado para comparar o desempenho de vários algoritmos em diferentes bases de dados. O teste é sugerido por não ser paramétrico, haja vista a dificuldade de se conhecer a distribuição dos dados.

Segundo os autores Iman e Davenport (1980) o Teste de Friedman é considerado muito conservador, e por tal motivo, sugerem a utilização da estatística FF distribuída de acordo com a distribuição F de Snedecor. Para a análise de quais algoritmos têm diferença significativa de desempenho, Desmar (2006) sugere o Teste de Nemenyi, que calcula a distância crítica entre os desempenhos, o qual foi usado nestas situações dentro deste trabalho.

## 7.6 RESULTADOS EXPERIMENTAIS

A avaliação do método FEAE-HMC foi realizada por meio da medida AUPRC obtida no Classificador Hierárquico Multirrótulo. Primeiramente, realizou-se o processo de classificação nas bases hierárquicas multirrótulo sem a aplicação do processo de extração. A

seguir, aplica-se o processo de extração de atributos (FEAE-HMC) nesses conjuntos, para cada um dos valores percentuais de redução estabelecidos e submete-se os conjuntos reduzidos obtidos ao longo do treinamento, ao processo de classificação. Os resultados são apresentados por conjunto de dados, sendo primeiramente os resultados obtidos e avaliados no classificador Clus-HMC, e posteriormente no MHC-CNN.

### 7.6.1 Resultados - CLUS HMC

Inicialmente, realizou-se o processo de classificação nas bases hierárquicas multirrótulo sem a aplicação do processo de extração, ou seja, com todos os atributos originais. Dessa maneira, a Tabela 11 apresenta os resultados obtidos, em relação a medida AUPRC, para os conjuntos de dados no classificador Clus-HMC, sendo que esses resultados foram relatados por Borges (2012).

**Tabela 11 - Medida AUPRC - CLUS-HMC - sem a aplicação do método FEAE-HMC**

| ID | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| AUPRC | 0,439 | 0,452 | 0,438 | 0,456 | 0,473 | 0,456 | 0,438 | 0,421 | 0,470 | 0,468 |

**Fonte: Autoria própria**

A metodologia para realização dos experimentos é aplicada para cada um dos conjuntos de dados utilizados. Cada conjunto de dados ao passar pelo método FEAE-HMC, até atingir a condição de parada, gera 10 bases de dados reduzidas, para cada um dos 9 percentuais de redução (*Pr*) estabelecidos, totalizando 90 bases reduzidas para cada conjunto de dados.

A Tabela 12 apresenta um resumo da quantidade de atributos originais (*n*) e a quantidade de novos atributos gerados (*m*) para cada um dos percentuais de redução (*Pr*), para fins de comparação, em cada conjunto de dados utilizados nos experimentos do método FEAE-HMC.

**Tabela 12 - Comparativo entre atributos das bases originais e bases reduzidas obtidas**

| ID | Atributos | Novos atributos (*m*) conforme *Pr* | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | (*n*) | 10% | 20% | 30% | 40% | 50% | 60% | 70% | 80% | 90% |
| 1 | 77 | 69 | 61 | 53 | 46 | 38 | 30 | 23 | 15 | 7 |
| 2 | 27 | 24 | 21 | 18 | 16 | 13 | 10 | 8 | 5 | 2 |
| 3 | 63 | 56 | 50 | 44 | 37 | 31 | 25 | 18 | 12 | 6 |
| 4 | 79 | 71 | 63 | 55 | 47 | 39 | 31 | 23 | 15 | 7 |
| 5 | 551 | 495 | 440 | 385 | 330 | 275 | 220 | 165 | 110 | 55 |
| 6 | 173 | 155 | 138 | 121 | 103 | 86 | 69 | 51 | 34 | 17 |
| 7 | 52 | 46 | 41 | 36 | 31 | 26 | 20 | 15 | 10 | 5 |
| 8 | 69 | 62 | 55 | 48 | 41 | 34 | 27 | 20 | 13 | 6 |
| 9 | 478 | 430 | 382 | 334 | 286 | 239 | 191 | 143 | 95 | 47 |
| 10 | 80 | 72 | 64 | 56 | 48 | 40 | 32 | 24 | 16 | 8 |

**Fonte: Autoria própria**

Para fins de demonstração, selecionou-se três conjuntos de dados para apresentar os resultados obtidos com os 9 percentuais durante as épocas selecionadas. As bases de dados selecionadas foram as com ID 1, 3, e 4. Os resultados das demais bases, estão descritas no Apêndice A.

A Tabela 13 apresenta os resultados das bases reduzidas pelo método FEAE-HMC para a base de dados de ID 1 quando avaliadas pelo classificador Clus-HMC. Os dados estão destacados em relação ao valor obtido com todos os atributos originais (Tabela 11). Para isso, tem-se destacado em vermelho quando o valor obtido é inferior; em preto quando o valor é superior. Já o destaque em azul é o valor máximo obtido pelo conjunto reduzido.

**Tabela 13 - Medidas AUPRC para os conjuntos de dados reduzidos - Cellcycle**

| *Pr* | Atributos | 100 | 200 | 300 | 400 | 500 | 600 | 700 | 800 | 900 | 1000 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 10% | 69 | 0,434 | 0,451 | 0,455 | **0,456** | 0,451 | 0,453 | 0,452 | 0,454 | 0,451 | 0,450 |
| 20% | 61 | 0,440 | **0,457** | 0,453 | 0,454 | 0,453 | 0,447 | 0,445 | 0,443 | 0,451 | 0,447 |
| 30% | 53 | 0,444 | 0,453 | 0,450 | 0,452 | 0,451 | 0,442 | 0,451 | **0,453** | 0,448 | 0,452 |
| 40% | 46 | 0,437 | 0,454 | 0,450 | 0,447 | 0,452 | 0,454 | **0,455** | 0,452 | 0,449 | 0,452 |
| 50% | 38 | 0,445 | **0,455** | 0,444 | 0,454 | 0,452 | 0,451 | 0,442 | 0,446 | 0,446 | 0,444 |
| 60% | 30 | 0,433 | 0,452 | 0,453 | 0,450 | **0,456** | 0,454 | 0,451 | 0,445 | 0,445 | 0,449 |
| 70% | 23 | 0,437 | 0,451 | **0,453** | 0,447 | 0,452 | 0,452 | 0,452 | 0,449 | 0,450 | 0,448 |
| 80% | 15 | 0,435 | 0,448 | 0,449 | **0,455** | 0,450 | 0,450 | 0,446 | 0,449 | 0,447 | 0,445 |
| 90% | 7 | 0,428 | 0,448 | **0,454** | 0,453 | 0,451 | 0,451 | 0,453 | 0,446 | 0,452 | 0,447 |

**Fonte: Autoria própria**

Observa-se que entre as 90 bases reduzidas geradas para o conjunto de dados de ID 1, 93% obtiveram valor superior a medida AUPRC de 0,439 da base original, sendo que nos conjuntos com 61, 53 e 38 atributos, todas as épocas selecionadas demonstram resultados superiores. Uma análise com algumas informações estatísticas observadas na Tabela 13, é apresentada na Tabela 14.

**Tabela 14 - Comparativo da medida AURPC nos diferentes percentuais - Cellcycle**

| Pr | Atributos | Média | Max | Desvio Padrão | % Média | % Max |
|---|---|---|---|---|---|---|
| 10% | 69 | **0,451** | 0,456 | 0,006 | 2,7% | 3,9% |
| 20% | 61 | 0,449 | **0,457** | 0,006 | 2,3% | 4,1% |
| 30% | 53 | 0,449 | 0,453 | 0,004 | 2,4% | 3,2% |
| 40% | 46 | 0,450 | 0,455 | 0,005 | 2,5% | 3,7% |
| 50% | 38 | 0,448 | 0,455 | 0,005 | 2,1% | 3,6% |
| 60% | 30 | 0,449 | 0,456 | 0,007 | 2,2% | 3,8% |
| 70% | 23 | 0,449 | 0,453 | 0,005 | 2,3% | 3,2% |
| 80% | 15 | 0,447 | 0,455 | 0,005 | 1,9% | 3,6% |
| 90% | 7 | 0,448 | 0,454 | 0,008 | 2,1% | 3,4% |
| | *Média* | *0,449* | *0,455* | *0,006* | *2,3%* | *3,6%* |

**Fonte: Autoria própria**

Nota-se que em média os valores obtidos de AUPRC para todos os percentuais de redução foi de 0,449, sendo que este valor é 2,3% superior ao valor obtido sem a redução de dimensionalidade. Já o valor máximo observado, em média é de 0,455, valor este 3,6% superior

ao valor obtido sem o processo de redução proposto. O desvio padrão demonstra que o método FEAE-HMC, nestes resultados, é uniforme independentemente do tamanho da redução aplicada, indicando que os valores obtidos estão condensados próximos da média.

Ao se analisar estatisticamente os resultados da Tabela 13, pelo Teste de Friedman, têm-se $x_F^2$= 26,15 e $F_F$= 3,686. De acordo com o valor tabelado na distribuição Snedecor, com α = 0,05, encontra-se o valor de F (9;81) = 1,998. Desse modo a hipótese nula pode ser rejeitada, pois $F_F$ > 1,998.

Rejeitada a hipótese nula, identifica-se que existem diferenças significativas de desempenho por meio do valor obtido para a distância crítica pelo Teste de Nemenyi, que nesse caso foi de 4,28. Considerando o nível de significância de 95%, os resultados com as bases reduzidas pelo método FEAE-HMC (nas épocas selecionadas) para a base de ID 1, são estatisticamente superiores a medida AUPRC da base completa nos conjuntos com 69, 61, 53, 46 e 23 atributos. Considerando o nível de significância de 90%, observa-se que os conjuntos com 30 e 7 atributos são estatisticamente superiores ao resultado obtido com os 77 atributos.

A Tabela 15, apresenta os resultados obtidos para o conjunto de dados de ID 3, sendo que os valores estão destacados, da mesma maneira descrita anteriormente para a Tabela 13, porém em relação ao valor 0,438 de AUPRC obtido por essa base com os 63 atributos originais.

**Tabela 15 - Medidas AUPRC para as bases reduzidas do conjunto de dados Derisi**

| ***Pr*** | **Atributos** | **100** | **200** | **300** | **400** | **500** | **600** | **700** | **800** | **900** | **1000** |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 10% | 56 | 0,436 | 0,441 | 0,441 | 0,436 | 0,436 | 0,440 | 0,439 | 0,438 | 0,442 | **0,442** |
| 20% | 50 | 0,440 | **0,442** | 0,441 | 0,439 | 0,438 | 0,436 | 0,434 | 0,435 | 0,433 | 0,435 |
| 30% | 44 | 0,437 | 0,439 | **0,440** | 0,435 | 0,437 | 0,437 | 0,439 | 0,436 | 0,434 | 0,435 |
| 40% | 37 | 0,437 | 0,440 | 0,438 | 0,436 | 0,438 | **0,440** | 0,440 | 0,439 | 0,436 | 0,437 |
| 50% | 31 | 0,436 | 0,438 | **0,440** | 0,439 | 0,440 | 0,435 | 0,436 | 0,435 | 0,437 | 0,434 |
| 60% | 25 | 0,439 | 0,439 | **0,440** | 0,436 | 0,437 | 0,437 | 0,435 | 0,434 | 0,434 | 0,434 |
| 70% | 18 | 0,434 | 0,439 | 0,440 | 0,441 | **0,442** | 0,438 | 0,439 | 0,437 | 0,438 | 0,438 |
| 80% | 12 | 0,437 | 0,439 | 0,439 | **0,442** | 0,438 | 0,438 | 0,437 | 0,438 | 0,436 | 0,438 |
| 90% | 6 | 0,431 | 0,440 | **0,440** | 0,439 | 0,439 | 0,435 | 0,434 | 0,434 | 0,432 | 0,434 |

**Fonte: Autoria própria**

Entre as 90 bases reduzidas geradas para o conjunto de dados de ID 3, 47,8% obtiveram valor superior a medida AUPRC de 0,438 da base original, sendo esses localizados, principalmente, nas épocas 200 e 300, para todos os percentuais de redução. Uma análise com algumas informações estatísticas observadas na Tabela 15, é apresentada na Tabela 16.

Observa-se que em média os valores obtidos de AUPRC para todos os percentuais de redução foi de 0,437, sendo este valor é 0,1% inferior ao valor obtido com os 63 atributos originais. Já o valor máximo foi em média de 0,441, valor 0,7% superior ao valor obtido sem o processo de redução de dimensionalidade.

**Tabela 16 - Comparativo da medida AURPC nos diferentes percentuais - Derisi**

| Pr | Atributos | Média | Max | Desvio Padrão | % Média | % Max |
|---|---|---|---|---|---|---|
| 10% | 56 | **0,439** | 0,442 | 0,002 | 0,2% | 0,8% |
| 20% | 50 | 0,437 | 0,442 | 0,003 | -0,1% | 0,9% |
| 30% | 44 | 0,437 | 0,440 | 0,002 | -0,3% | 0,5% |
| 40% | 37 | 0,438 | 0,440 | 0,002 | 0,0% | 0,5% |
| 50% | 31 | 0,437 | 0,440 | 0,002 | -0,2% | 0,5% |
| 60% | 25 | 0,436 | 0,440 | 0,002 | -0,4% | 0,4% |
| 70% | 18 | 0,439 | 0,442 | 0,002 | 0,1% | 0,9% |
| 80% | 12 | 0,438 | **0,442** | 0,002 | 0,1% | 0,9% |
| 90% | 6 | 0,436 | 0,440 | 0,003 | -0,5% | 0,5% |
| | *Média* | *0,437* | *0,441* | *0,002* | *-0,1%* | *0,7%* |

**Fonte: Autoria própria**

O desvio padrão demonstra que o método FEAE-HMC, nestes resultados, também é uniforme independentemente do tamanho da redução aplicada, indicando que os valores obtidos estão condensados próximos da média.

Ao se analisar estatisticamente os resultados da Tabela 15, pelo Teste de Friedman, têm-se $x_F^2$= 15,14 e $F_F$= 1,819. De acordo com o valor tabelado na distribuição Snedecor, com α = 0,05, encontra-se o valor de F (9;81) = 1,998. Desse modo, a hipótese nula não pode ser rejeitada, pois $F_F$ < 1,998. Isso significa que não há diferença estatística significativa entre o desempenho obtido entre a base completa e as bases reduzidas.

A Tabela 17, apresenta os resultados obtidos para o conjunto de dados de ID 6, sendo que os valores destacados, estão em conformidade com a maneira descrita para as tabelas anteriores. Essa base tem 173 atributos e obteve com esses atributos o valor 0,456 de AUPRC.

**Tabela 17 - Medidas AUPRC para as bases reduzidas do conjunto de dados Gasch1**

| *Pr* | Atributos | 100 | 200 | 300 | 400 | 500 | 600 | 700 | 800 | 900 | 1000 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 10% | 155 | 0,451 | 0,467 | 0,466 | **0,474** | 0,467 | 0,465 | 0,465 | 0,467 | 0,466 | 0,465 |
| 20% | 138 | 0,463 | **0,468** | 0,468 | 0,463 | 0,459 | 0,461 | 0,466 | 0,467 | 0,467 | 0,468 |
| 30% | 121 | 0,461 | 0,471 | 0,471 | 0,464 | 0,459 | **0,472** | 0,462 | 0,464 | 0,461 | 0,463 |
| 40% | 103 | 0,466 | **0,471** | 0,467 | 0,468 | 0,464 | 0,466 | 0,459 | 0,464 | 0,464 | 0,463 |
| 50% | 86 | 0,467 | 0,467 | **0,469** | 0,453 | 0,465 | 0,464 | 0,458 | 0,459 | 0,458 | 0,462 |
| 60% | 69 | 0,465 | **0,472** | 0,466 | 0,470 | 0,467 | 0,463 | 0,462 | 0,461 | 0,466 | 0,467 |
| 70% | 51 | 0,464 | 0,460 | 0,462 | **0,468** | 0,463 | 0,464 | 0,461 | 0,463 | 0,463 | 0,464 |
| 80% | 34 | 0,465 | 0,468 | 0,469 | 0,459 | 0,471 | **0,472** | 0,470 | 0,469 | 0,469 | 0,469 |
| 90% | 17 | 0,469 | 0,467 | 0,458 | **0,472** | 0,464 | 0,469 | 0,465 | 0,464 | 0,463 | 0,466 |

**Fonte: Autoria própria**

Nas 90 bases reduzidas geradas para o conjunto de dados de ID 6, 97,8% obtiveram valor superior a medida AUPRC de 0,456 da base original. Informações estatísticas são apresentadas na Tabela 18. Observa-se que em média os valores obtidos de AUPRC para todos os percentuais de redução foi de 0,465, sendo esse valor em média 2,0% superior ao valor obtido com os 155 atributos originais. O valor máximo foi em média de 0,471, valor 3,2 % superior ao valor obtido sem o processo de redução de dimensionalidade.

**Tabela 18 - Comparativo da medida AURPC nos diferentes percentuais - Gasch1**

| Pr | Atributos | Média | Max | Desvio Padrão | % Média | % Max |
|---|---|---|---|---|---|---|
| 10% | 155 | 0,465 | **0,474** | 0,006 | 2,0% | 4,0% |
| 20% | 138 | 0,465 | 0,468 | 0,003 | 2,0% | 2,6% |
| 30% | 121 | 0,465 | 0,472 | 0,005 | 1,9% | 3,5% |
| 40% | 103 | 0,465 | 0,471 | 0,003 | 2,0% | 3,2% |
| 50% | 86 | 0,462 | 0,469 | 0,005 | 1,3% | 2,9% |
| 60% | 69 | 0,466 | 0,472 | 0,003 | 2,2% | 3,4% |
| 70% | 51 | 0,463 | 0,468 | 0,002 | 1,5% | 2,6% |
| 80% | 34 | **0,468** | 0,472 | 0,004 | 2,6% | 3,5% |
| 90% | 17 | 0,466 | 0,472 | 0,004 | 2,1% | 3,4% |
| | *Média* | *0,465* | *0,471* | *0,004* | *2,0%* | *3,2%* |

**Fonte: Autoria própria**

O valor baixo para o desvio padrão demonstra que os dados reduzidos, são uniformes, independentemente do tamanho da redução aplicada e estão condensados próximos da média.

Ao se analisar estatisticamente os resultados da Tabela 17, pelo Teste de Friedman, têm-se $x_F^2$= 36,21 e $F_F$= 6,059. De acordo com o valor tabelado na distribuição Snedecor, com α α = 0,05, encontra-se o valor de F (9;81) = 1,998. Desse modo a hipótese nula pode ser rejeitada, pois $F_F$ > 1,998.

Existe diferença significativa de desempenho por meio do valor 4,28 obtido para a distância crítica pelo Teste de Nemenyi. Considerando o nível de significância de 95%, os resultados com as bases reduzidas pelo método FEAE-HMC (nas épocas selecionadas) para a base de ID 6, são estatisticamente superiores a medida AUPRC da base completa nos conjuntos com 155, 138, 121, 103, 69, 34 e 17 atributos. Para as bases com 38 e 15 conjuntos, observa-se que não existe diferença estatística significativa quando analisados os desempenhos entre a base original e as bases reduzidas.

Os resultados das bases reduzidas pelo FEAE-HMC foram obtidos, para todas as bases, através da execução da metodologia descrita. Desse modo, foram realizados diversos experimentos para um único conjunto de dados, pois cada conjunto ao ser submetido ao método FEAE-HMC, conforme a metodologia utilizada, gerou 10 bases de dados reduzidas, para cada um dos 9 Percentuais de Redução (Pr) estabelecidos, o que totalizou 90 bases reduzidas e avaliadas no Clus-HMC, para cada um dos 10 conjuntos de dados da GO.

Devido ao fato da dificuldade de apresentar e comparar nesta seção cada uma das 90 bases reduzidas geradas por conjunto de dados inicial, esses resultados foram apresentados e descritos no apêndice A. Nesse sentido, considera-se como resultado obtido ao longo do treinamento, para um valor de *Pr*, somente a iteração selecionada com o maior valor de AUPRC obtido. Dessa maneira, para cada conjunto de dados, compara-se o valor de AUPRC da base completa (Tabela 11) e o desempenho encontrado para cada uma das 10 bases reduzidas geradas para o *Pr* estabelecido.

Para fins de comparação de desempenho, a Tabela 19 apresenta para cada conjunto de dados o valor AUPRC obtido para a base: com todos os atributos (TA); com cada percentual de redução; o valor médio AUPRC (Méd); o valor máximo AUPRC (Max); o percentual de desempenho médio (%Méd) e máximo (%Max) do processo de redução em relação a medida AUPRC obtida nas bases com todos os atributos.

**Tabela 19 - Comparativo da medida AURPC - CLUS-HMC**

| ID | TA | Percentual de Redução Aplicado | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 10% | 20% | 30% | 40% | 50% | 60% | 70% | 80% | 90% | Med | Max | %Méd | %Max |
| **1** | 0,439 | 0,456 | 0,457 | 0,453 | 0,455 | 0,455 | 0,456 | 0,453 | 0,455 | 0,454 | 0,455 | 0,457 | 3,6% | 4,1% |
| **2** | 0,452 | 0,441 | 0,441 | 0,446 | 0,444 | 0,443 | 0,445 | 0,442 | 0,439 | 0,440 | 0,442 | 0,446 | -2,1% | -1,3% |
| **3** | 0,438 | 0,442 | 0,442 | 0,440 | 0,440 | 0,440 | 0,440 | 0,442 | 0,442 | 0,440 | 0,441 | 0,442 | 0,7% | 0,9% |
| **4** | 0,456 | 0,470 | 0,475 | 0,473 | 0,469 | 0,472 | 0,476 | 0,475 | 0,474 | 0,476 | 0,473 | 0,476 | 3,8% | 4,4% |
| **5** | 0,473 | 0,468 | 0,473 | 0,472 | 0,467 | 0,469 | 0,469 | 0,468 | 0,473 | 0,471 | 0,470 | 0,473 | -0,6% | 0,0% |
| **6** | 0,456 | 0,474 | 0,468 | 0,472 | 0,471 | 0,469 | 0,472 | 0,468 | 0,472 | 0,472 | 0,471 | 0,474 | 3,3% | 3,9% |
| **7** | 0,438 | 0,450 | 0,447 | 0,447 | 0,445 | 0,448 | 0,443 | 0,448 | 0,447 | 0,446 | 0,447 | 0,450 | 2,0% | 2,7% |
| **8** | 0,421 | 0,421 | 0,421 | 0,421 | 0,422 | 0,422 | 0,423 | 0,422 | 0,422 | 0,422 | 0,422 | 0,423 | 0,2% | 0,5% |
| **9** | 0,470 | 0,459 | 0,460 | 0,459 | 0,461 | 0,462 | 0,455 | 0,457 | 0,458 | 0,452 | 0,458 | 0,462 | -2,5% | -1,7% |
| **10** | 0,468 | 0,454 | 0,455 | 0,455 | 0,457 | 0,457 | 0,452 | 0,457 | 0,451 | 0,458 | 0,455 | 0,458 | -2,8% | -2,1% |

**Fonte: Autoria própria**

Observa-se que em 6 conjuntos de dados, o desempenho médio das bases reduzidas gerados pelo FEAE-HMC no Clus-HMC, foi superior em relação a medida AUPRC obtida com o conjunto com todos os atributos (TA), chegando ao percentual de 3,8% (ID 4). Por outro lado, 4 conjuntos de dados tiveram em média um desempenho inferior em relação a medida AUPRC obtida com todos os atributos.

Ao se analisar estatisticamente os resultados da Tabela 19, pelo Teste de Friedman, têm-se $x_F^2$= 1,92 e $F_F$= 0,02. De acordo com o valor tabelado na distribuição Snedecor, com α α = 0,05, encontra-se o valor de F (9;81) = 1,998. Desse modo a hipótese nula não pode ser rejeitada, pois $F_F$ < 1,998. Então, observa-se que não há diferença significativa entre os resultados dos algoritmos.

Dessa maneira, é possível observar que o método FEAE-HMC pode reduzir a dimensionalidade em bases de dados de classificação hierárquica, por meio de novas representações dos dados originais em espaços de menor dimensão, sendo os resultados obtidos com os conjuntos reduzidos, para a tarefa de HMC, equivalentes estatisticamente com os resultados obtidos com todos os atributos.

A diferença obtida entre a medida AUPRC da base completa em relação a todas as bases reduzidas é inferior 0,017 (ID 10). Com relação ao ganho na redução da quantidade de atributos, pode-se verificar que em todas as bases analisadas, as bases reduzidas, são equivalentes e muito próximo à média de maneira geral, podendo a redução atingir, conforme os percentuais utilizados nos experimentos, 90% da dimensionalidade original.

### 7.6.2 Resultados - MHC-CNN

Para o processo de classificação nas bases hierárquicas multirrótulo, inicialmente foram avaliadas no MHC-CNN as bases com todos os atributos originais, ou seja, sem aplicação do processo de extração de atributos. Dessa maneira, a Tabela 20 apresenta os resultados obtidos em relação a medida AUPRC, para os conjuntos de dados no classificador MHC-CNN, sendo que esses resultados foram relatados por Borges (2012).

**Tabela 20 - Medida AUPRC - MHC-CNN - sem a aplicação do método FEAE-HMC**

| ID | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| AUPRC | 0,09 | 0,13 | 0,12 | 0,09 | 0,15 | 0,11 | 0,10 | 0,12 | 0,09 | 0,09 |

**Fonte: Autoria própria**

Nesse experimento, para propiciar a correta comparação do desempenho, utilizam-se somente as bases reduzidas geradas pelo FEAE-HMC apresentadas na Tabela 19, que representam o maior resultado AUPRC obtido ao longo do treinamento para cada percentual de redução (*Pr*). Portanto, para cada conjunto de dados, 9 bases reduzidas foram avaliadas pelo classificador MHC-CNN.

Para fins de comparação de desempenho, a Tabela 21 apresenta para cada conjunto de dados (ID) o valor AUPRC obtido para a base: com todos os atributos (TA); com cada percentual de redução; o valor médio AUPRC (Méd); o valor máximo AUPRC (Max); o percentual de desempenho médio (%Méd) e máximo (%Max) do processo de redução em relação a medida AUPRC obtida nas bases com todos os atributos.

**Tabela 21 - Comparativo da medida AURPC - MHC-CNN**

| | | Percentual de Redução Aplicado | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **ID** | **TA** | **10%** | **20%** | **30%** | **40%** | **50%** | **60%** | **70%** | **80%** | **90%** | **Méd** | **Max** | **%Méd** | **%Max** |
| **1** | 0,09 | 0,176 | 0,195 | 0,161 | 0,102 | 0,191 | 0,133 | 0,171 | 0,157 | 0,180 | 0,163 | 0,195 | 81% | 116% |
| **2** | 0,13 | 0,127 | 0,122 | 0,155 | 0,141 | 0,083 | 0,136 | 0,164 | 0,142 | 0,063 | 0,126 | 0,164 | -3% | 26% |
| **3** | 0,12 | 0,140 | 0,162 | 0,160 | 0,106 | 0,145 | 0,164 | 0,151 | 0,164 | 0,149 | 0,149 | 0,164 | 24% | 37% |
| **4** | 0,09 | 0,166 | 0,170 | 0,128 | 0,149 | 0,178 | 0,183 | 0,153 | 0,118 | 0,182 | 0,159 | 0,183 | 76% | 103% |
| **5** | 0,15 | 0,143 | 0,147 | 0,150 | 0,140 | 0,128 | 0,156 | 0,174 | 0,167 | 0,165 | 0,152 | 0,174 | 1% | 16% |
| **6** | 0,11 | 0,141 | 0,171 | 0,146 | 0,175 | 0,186 | 0,171 | 0,150 | 0,124 | 0,139 | 0,156 | 0,186 | 42% | 69% |
| **7** | 0,1 | 0,188 | 0,115 | 0,137 | 0,173 | 0,160 | 0,116 | 0,147 | 0,148 | 0,139 | 0,147 | 0,188 | 47% | 88% |
| **8** | 0,12 | 0,110 | 0,116 | 0,125 | 0,089 | 0,096 | 0,106 | 0,130 | 0,095 | 0,108 | 0,108 | 0,130 | -10% | 9% |
| **9** | 0,09 | 0,137 | 0,120 | 0,130 | 0,108 | 0,118 | 0,130 | 0,139 | 0,117 | 0,092 | 0,121 | 0,139 | 35% | 54% |
| **10** | 0,09 | 0,087 | 0,115 | 0,110 | 0,102 | 0,124 | 0,113 | 0,084 | 0,154 | 0,108 | 0,111 | 0,154 | 23% | 72% |

**Fonte: Autoria própria**

Observa-se que em 8 conjunto de dados, o desempenho médio dos conjuntos reduzidos pelo FEAE-HMC, é superior em relação ao valor da medida AUPRC obtida com o conjunto com todos os atributos (TA), chegando a 81% (ID 1). Por outro lado, 2 conjuntos de dados tiveram em média um desempenho inferior em relação a medida AUPRC obtida com TA.

No entanto, ao se ter como referência o valor máximo, é possível verificar que existe ganho no desempenho do algoritmo MHC-CNN em relação a medida de avaliação AUPRC utilizando as bases reduzidas pelo método FEAE-HMC, sendo que essa melhora é de no mínimo 9% para todos as bases, podendo chegar até o percentual de 116% (base ID 1).

Ao se analisar estatisticamente os resultados da Tabela 21, pelo Teste de Friedman, têm-se $x_F^2$= 14,38 e $F_F$= 1,711. De acordo com o valor tabelado na distribuição Snedecor, com α = 0,10, encontra-se o valor de F (9;81) = 1,710. Desse modo a hipótese nula pode ser rejeitada, pois $F_F$ > 1,710. Então, observa-se que há diferença estatística significativa entre os resultados obtidos com as bases reduzidas em relação a base completa.

Rejeitada a hipótese nula, identifica-se a existência de diferenças significativas de desempenho por meio do valor obtido para a distância crítica de 4,28 pelo Teste de Nemenyi. Considerando o nível de significância de 90%, os resultados com as bases reduzidas pelo método FEAE-HMC para o percentual de redução 70%, são estatisticamente superiores a medida AUPRC, obtida no MHC-CNN, em todas as bases com os atributos originais.

É possível observar que o método FEAE-HMC obteve redução de dimensionalidade nas bases de dados, de até 90% da dimensionalidade original, por meio de novas representações dos dados originais, em espaços de menor dimensão. A diferença obtida da medida AUPRC da base completa em relação a todas as bases reduzidas é inferior a 0,047 (ID 2). Esses resultados são no mínimo equivalentes estatisticamente com os resultados obtidos com todos os atributos, mas podem atingir diferença estatística de 90%.

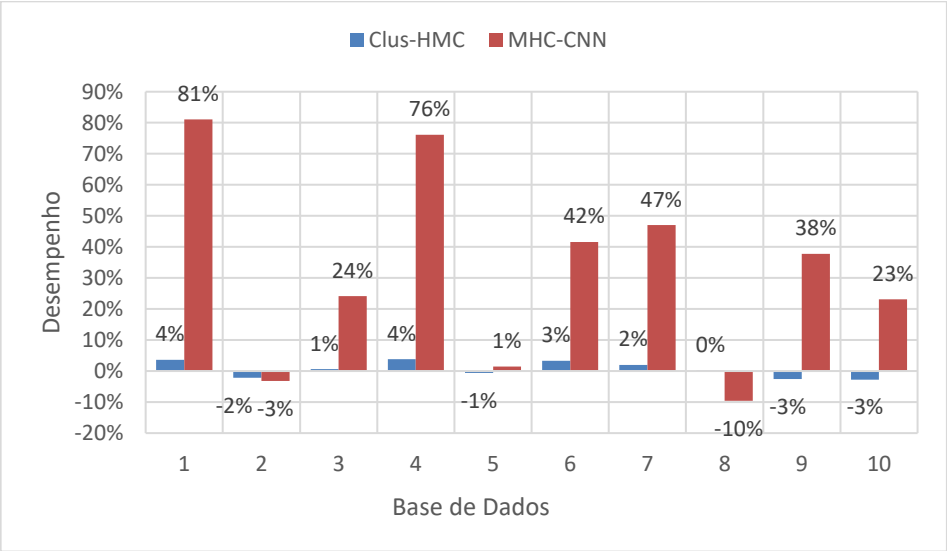
### 7.6.3 Comparação Geral dos Resultados Obtidos

Para fins de comparação do desempenho dos classificadores nos experimentos, conforme os resultados apresentados na Tabela 19, para o algoritmo Clus-HMC, e na Tabela 21, para o algoritmo MHC-CNN utilizando as bases reduzidas pelo método FEAE-HMC. A Figura 25 apresenta para cada conjunto de dados (ID) a porcentagem de ganho médio em relação a medida AUPRC obtida nas bases com todos os atributos.

Observa-se na Figura 25 que de modo geral, o classificador MHC-CNN teve em média seu desempenho preditivo aumentado significativamente ao utilizar as bases reduzidas pelo método FEAE-HMC, sendo que apenas em 2 bases seu desempenho preditivo foi inferior. De maneira semelhante, o Clus-HMC para a maioria das bases teve seu desempenho incrementado

ao utilizar as bases reduzidas, porém não foi tão significativo. O desempenho médio, de modo geral, foi inferior nas bases de ID 2, 9 e 10.

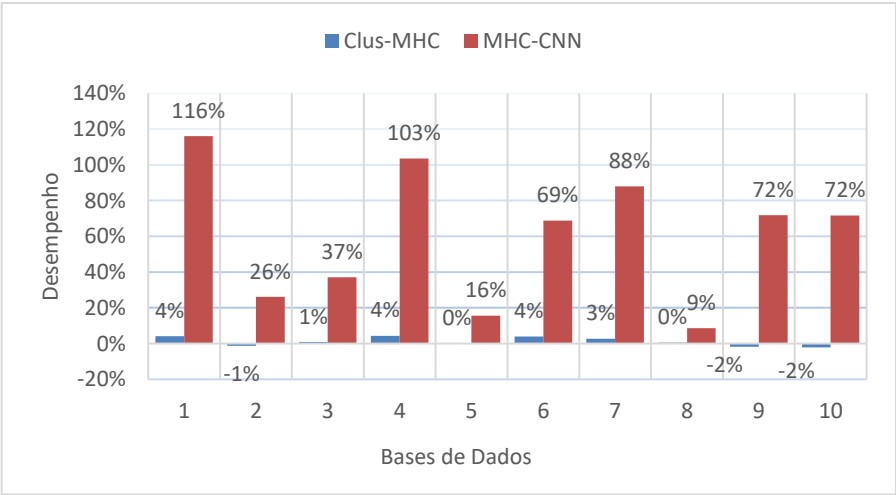
**Figura 25 - Comparativo de desempenho médio dos classificadores nas bases reduzidas**




**Fonte: Autoria própria**

Um comparativo sobre o desempenho preditivo que considera o valor AUPRC máximo observado nos classificadores com as bases reduzidas pelo método FEAE-HMC e o valor obtido com a base de dados com todos os atributos, é apresentado na Figura 26.

**Figura 26 - Comparativo de desempenho máximo dos classificadores nas bases reduzidas**



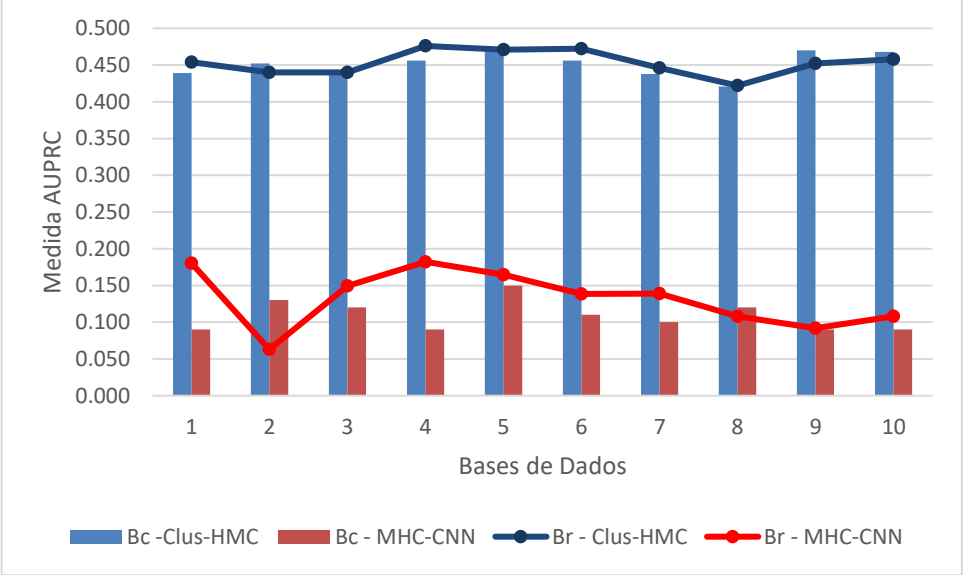

**Fonte: Autoria própria**

Na Figura 26 é possível verificar que novamente o algoritmo MHC-CNN teve seu desempenho preditivo melhorado significativamente ao utilizar as bases reduzidas pelo método FEAE-HMC, sendo que em todos os conjuntos de dados seu desempenho preditivo foi superior.

No Clus-HMC verifica-se que ainda assim, os ganhos de desempenho preditivo são pequenos e em alguns casos inferiores, como observados nas bases de ID 2, 9 e 10.

Para comparar o desempenho e o comportamento dos algoritmos, em relação a redução de dimensionalidade, a Figura 27, apresenta como exemplo, uma comparação do desempenho nos classificadores utilizando as bases reduzidas (Br), que foram geradas com um percentual de 90% de redução, em relação ao desempenho obtido na base completa (Bc), com todos os atributos originais.

**Figura 27 - Comparativo de desempenho - Base com 90% de redução x Base Completa**




**Fonte: Autoria própria**

Pode-se observar que as bases reduzidas pelo método FEAE-HMC, mesmo em um cenário em que cada base reduzida corresponde a 90% da dimensão original dos dados, possuem desempenho equivalente ao da base com todos os atributos, e até superior em alguns casos, pois as linhas que representam o desempenho das bases reduzidas, acompanham o resultado obtido nas colunas que representam as bases originais.

Dessa maneira, no cenário apresentado durante os experimentos, o método FEAE-HMC proporcionou aos algoritmos de classificação hierárquica multirrótulo, ganho computacional no que diz respeito a redução de dimensionalidade dos atributos, mantendo as caraterísticas intrínsecas dos conjuntos de dados originais, sendo que o desempenho do algoritmo com uma base reduzida é muito próximo, equivalente e até mesmo superior ao desempenho das bases originais.

## 7.7 CONSIDERAÇÕES FINAIS DO CAPÍTULO

Neste capítulo apresentou-se a metodologia utilizada nos experimentos e os resultados preliminares do método de redução proposto. Foi realizada uma comparação do desempenho de dois classificadores hierárquicos multirrótulo, Clus-HMC e MHC-CNN, em relação a medida AUPRC obtida nesses classificadores com os atributos originais de cada conjunto de dados e com bases de dados reduzidas pelo método FEAE-HMC.

Para verificar a capacidade de ganho na aplicação da redução de dimensionalidade e a adaptação do método para diferentes espaços de menor dimensão, as bases reduzidas foram extraídas através de 9 percentuais de redução, em relação a respectiva dimensionalidade *n* de cada conjunto de dados utilizado nos experimentos. Assim, diferentes representações de menor dimensão do espaço dos atributos foram avaliadas pelos classificadores.

No Clus-HMC, o método proposto apresentou resultados similares, e até mesmo superiores em alguns casos, aos obtidos sem a extração de atributos. A avaliação estatística demonstra que os resultados obtidos com as bases reduzidas no Clus-HMC são no mínimo equivalentes as medidas AUPRC obtidas com os atributos originais.

Já no classificador MHC-CNN, o desempenho em relação a medida AUPRC foi superior em todos os casos testados. Isso demonstrou que o método FEAE-HMC foi capaz de gerar características mais relevantes para a tarefa de classificação, sendo que os espaços de menor dimensão por ele extraídos, podem agregar correlações existentes entre os atributos e as classes. Com a realização do teste estatístico, verificou-se que os resultados obtidos pelas bases reduzidas, podem ser estatisticamente superiores aos resultados obtidos nas bases com todos os atributos originais.

Por fim, como os resultados demonstrados para os diversos experimentos realizados, foram validados estatisticamente por meio do Teste de Friedman e pelo teste Nemenyi, é possível verificar que uma base reduzida extraída pela aplicação do método FEAE-HMC é, no mínimo, estatisticamente equivalente ao resultado da base original, mas pode obter significância estatística superior a 90% em relação ao desempenho obtido com todos os atributos originais.

# 8 CONCLUSÃO

Um dos aspectos em que ainda há um caminho longo para evolução na aprendizagem de máquina é em relação ao tratamento de dados com alta dimensionalidade. Na tarefa de classificação, a performance de um algoritmo de aprendizagem em uma base de dados com alta dimensão, pode ser prejudicada tanto na velocidade de execução, quanto na taxa de acerto.

A extração de atributos é uma das técnicas utilizadas para alcançar a redução de dimensionalidade em base de dados, e assim eliminar atributos irrelevantes e/ou redundantes que tendem a confundir o algoritmo de aprendizagem. O processo de extração, por meio de combinações e ou transformações dos atributos originais, gera novos atributos, mais significativos e que melhor representam a base de dados, em um espaço de menor dimensão.

Nesse sentido, esse trabalho apresentou o método de extração de atributos FEAE-HMC, utilizando conceitos e técnicas de *Deep Learning*, por meio de adaptações em uma rede *Autoencoder* clássica.

Nos experimentos foram utilizadas 10 bases de dados da Ontologia Gênica em que as classes estão dispostas hierarquicamente em uma estrutura do tipo DAG. Em relação a avaliação do método de redução, optou-se por dois classificadores hierárquicos multirrótulo, Clus-HMC e MHC-CNN, e pela medida de avaliação de desempenho AUPRC.

Para verificar a capacidade preditiva do método em relação a redução alcançada, definiu-se uma metodologia para a execução dos experimentos, na qual o método FEAE-HMC foi testado em 9 percentuais de redução, para cada conjunto de dados submetido ao método, gerou. Assim, foi possível verificar o desempenho de diversas bases de dados reduzidas pelo método FEAE-HMC, em relação ao desempenho obtido com os atributos originais.

Ao realizar-se a análise dos melhores resultados das bases reduzidas extraídas pelo FEAE-HMC, verifica-se que a diferença obtida da medida AUPRC da base completa em relação a todas as bases reduzidas é inferior a 0,047 em ambos classificadores. Os resultados obtidos foram validados estatisticamente por meio do Teste de Friedman e pelo Teste de Nemenyi, no qual foi possível verificar que uma base reduzida extraída pela aplicação do método FEAE-HMC é, no mínimo, estatisticamente equivalente ao resultado da base original, mas pode obter significância estatística superior a 90% em relação ao resultado obtido com a base com todos os atributos originais.

Ressalta-se que durante a realização dos experimentos, foi possível constatar que a performance de ambos classificadores, obteve ganho tanto na velocidade de execução, quanto na taxa de acerto, quando receberam as bases reduzidas pelo método FEAE-HMC.

Uma característica do método FEAE-HMC é que por tratar-se de um método de redução genérico, pode ser utilizado tanto para bases hierárquicas monorrótulo como multirrótulo. Com relação a estrutura hierárquica, pode ser utilizado tanto para o tipo DAG quanto para árvore. O método permite com que sejam utilizados outros classificadores e/ou medidas de avaliação de capacidade preditiva.

No que se refere a redução de dimensionalidade em problemas de classificação hierárquica multirrótulo, conforme a revisão de literatura apresentada no Capítulo 5, métodos de extração de atributos são utilizados apenas no pré-processamento de alguns classificadores. Portanto, a apresentação de novos métodos de redução de dimensionalidade para a área de HMC, é uma contribuição deste trabalho.

Outra contribuição é que o método FEAE-HMC se mostrou capaz de extrair representações de menor dimensão, agregando correlações entre os atributos e rótulos, que quando submetidas a um Classificador Hierárquico Multirrótulo, geram modelos, onde-se obtêm o desempenho preditivo equivalente e até mesmo superior ao desempenho da base original.

## 8.1 TRABALHOS FUTUROS

Várias são as extensões que podem ser exploradas visando a ampliação e aprimoramento desse trabalho. Uma das propostas é aplicar o método FEAE-HMC em outras bases de dados de diferentes domínios, para comparação de desempenho. Outra proposta está relacionada em utilizar outros classificadores hierárquicos multirrótulo, assim como outras medidas de avaliação, a fim de validar os resultados obtidos.

Por fim, um desafio é desenvolver uma métrica específica para avaliar o desempenho do método FEAE-HMC, para que seja possível definir um critério de parada que não esteja vinculado somente a quantidade de épocas de treinamento da rede. Desse modo, será possível obter a melhor estrutura codificadora, e consequentemente, a melhor representação reduzida para a base na qual o método FEAE-HMC esteja sendo aplicado.

# REFERÊNCIAS


BALAKRISHNAMA, S.; GANAPATHIRAJU, A. Linear discriminant analysis - A brief tutorial. **Institute for Signal and information Processing**, v. 18, p. 1-8, 1998.

BENGIO, Y. *et al.* Greedy layer-wise training of deep networks. *In:* **Advances in neural information processing systems**. 2007. p. 153-160.

BENGIO, Y. Learning Deep Architectures for AI. **Foundations and trends® in Machine Learning**, v. 2, n. 1, p. 1-127, 2009.

___. Deep learning of representations for unsupervised and transfer learning. In: **Proceedings of ICML Workshop on Unsupervised and Transfer Learning**. 2012. p. 17-36.

BENGIO, Y.; COURVILLE, A.; VINCENT, P. Representation Learning: A Review and New Perspectives. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 35, n. 8, p. 1798-1828, 2013.

BLOCKEEL, H. *et al*. Hierarchical multi-classification. *In:* **Proceedings of the ACM SIGKDD 2002 Workshop on Multi-relational Data Mining (MRDM 2002)**. Edmonton, 2002.

BORGES, H. B. **Redução de Dimensionalidade em Bases de Dados de Expressão Gênica**. Dissertação (Mestrado) - Pontifícia Universidade Católica do Paraná, Curitiba, Brasil, 2006.

BORGES, H. B. **Classificador hierárquico multirrótulo usando uma rede neural competitiva.** Tese (Doutorado) - Pontifícia Universidade Católica do Paraná, Curitiba, Brasil, 2012.

BRAGA, A. DE P. **Redes neurais artificiais: teoria e aplicações**. Rio de Janeiro: LTC Editora, 2007.

CAMARGO, S. DA S. **Um modelo neural de aprimoramento progressivo para redução de dimensionalidade**. (Tese de Doutorado) - Universidade Federal do Rio Grande do Sul, Porto Alegre, Brasil, 2010.

CAMPOS, T. E. **Técnicas de seleção de características com aplicações em reconhecimento de faces.** (Tese de Doutorado) - Universidade de São Paulo, São Paulo, Brasil, 2001.

CARVALHO, A. *et al.* **Inteligência Artificial:** uma abordagem de aprendizado de máquina. Rio de Janeiro: LTC Editora, 2011.

CERRI, R. **Técnicas de classificação hierárquica multirrótulo**. (Tese de Doutorado) - Universidade de São Paulo, São Paulo, Brasil, 2010.

CERRI, R. *et al.* Multi-label Feature Selection Techniques for Hierarchical Multi-label Protein Function Prediction. *In*: IEEE. **2018 International Joint Conference on Neural Networks (IJCNN)**. [S. l.], 2018. p. 1-7.

CHEN, J. *et al*. Multi-Label Learning With Fuzzy Hypergraph Regularization for Protein Subcellular Location Prediction. **IEEE Transactions on NanoBioscience**, IEEE, v. 13, n. 4, p. 438-447, 2014.

COLONNA, J. G.; GAMA, J.; NAKAMURA, E. F. A comparison of hierarchical multi-output recognition approaches for anuran classification. **Machine Learning**, Springer, v. 107, n. 11, p. 1651-1671, 2018.

COSTA, E. P. **Investigação de técnicas de classificação hierárquica para problemas de bioinformática**. (Tese de Doutorado) - Universidade de São Paulo, São Paulo, Brasil, 2008.

DASH, M.; LIU, H. Feature selection for classification. **Intelligent Data Analysis**, Elsevier, v. 1, n. 1, p. 131-156, 1 jan. 1997.

DAVIS, J.; GOADRICH, M. The Relationship Between Precision-Recall and ROC Curves. *In:* ACM. **Proceedings of the 23rd International Conference on Machine learning**. [S. l.], 2006. p. 233-240.

DESMAR, J. Statistical comparisons of classifiers over multiple data sets. **Journal of Machine Learning Research**. v.7, p. 1-30, 2006.

DIMITROVSKI, I. *et al.* Detection of visual concepts and annotation of images using ensembles of trees for hierarchical multi-label classification. *In:* SPRINGER. **International Conference on Pattern Recognition**. [S. l.], 2010. p. 152-161.

___. *et al*. Hierarchical annotation of medical images. **Pattern Recognition**, Elsevier, v. 44, n. 10-11, p. 2436-2449, 2011.

___. *et al*. Hierarchical classification of diatom images using ensembles of predictive clustering trees. **Ecological Informatics**, Elsevier, v. 7, n. 1, p. 19-29, 2012.

FAYYAD, U. *et al*. From data mining to knowledge discovery in databases. **AI Magazine**, v. 17, n. 3, p. 37-37, 1996.

FALOUTSOS, C.; LIN, K.-I. **FastMap: A Fast Algorithm for Indexing, Data-mining and Visualization of Traditional and Multimedia Datasets**. v. 24, [s. l.]: ACM, 1995.

FREITAS, A.; CARVALHO, A. A tutorial on hierarchical classification with applications in bioinformatics. *In:* **Research and trends in data mining technologies and applications**. [S. l.]: IGI Global, 2007. p. 175-208.

FRIEDMAN, J. H.; TUKEY, J. W. A Projection Pursuit Algorithm for Exploratory Data Analysis. **IEEE Transactions on Computers**, IEEE, v. 100, n. 9, p. 881-890, 1974.

GARGIULO, F. *et al.* Deep neural network for hierarchical extreme multi-label text classification. **Applied Soft Computing**, Elsevier, v. 79, p. 125-138, 2019.

GHODSI, A. Dimensionality reduction a short tutorial. **Department of Statistics and Actuarial Science,** Univ. of Waterloo, Ontario, Canada, v. 37, p. 38, 2006.

GIRSHICK, R. *et al*. Rich feature hierarchies for accurate object detection and semantic segmentation. *In:* In: **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition**. [S. l.: s. n.], 2014. p. 580-587.
GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep learning**. [s.l.] MIT Press, 2016.

**GOOGLE SCHOLAR**. Disponível em <http://scholar.google.com.br>. Acessado em: 02 Fev.de 2019.

HAYKIN, S. S. **Neural networks and learning machines**. Nova York: Prentice Hall, 2009.

HINTON, G. *et al*. Deep Neural Networks for Acoustic Modeling in Speech Recognition. **IEEE Signal Processing Magazine**, IEEE, v. 29, 1 nov. 2012.

HINTON, G. E.; OSINDERO, S.; TEH, Y.-W. A fast learning algorithm for deep belief nets. **Neural Computation**, MIT Press, v. 18, n. 7, p. 1527-1554, 2006.

IMAN, R. L.; DAVENPORT, J. M. Approximations of the critical region of the friedman statistic. **Communications in Statistics**. p. 571-595, 1980.

JAIN, A. K.; DUIN, R. P. W.; MAO, J. Statistical pattern recognition: A review. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, IEEE, v. 22, n. 1, p. 4-37, 2000.

JOHN, G. H.; KOHAVI, R.; PFLEGER, K. Irrelevant Features and the Subset Selection Problem. *In:* **Machine Learning Proceedings 1994**. San Francisco: Elsevier, 1994. p. 121-129.

JOLLIFFE, I. **Principal component analysis**. [S. l.]: Springer, 2011.

KINGMA, D. P.; BA, J. Adam: A method for stochastic optimization. **arXiv preprint arXiv:1412.6980**, 2014.

KIRA, K.; RENDELL, L. A. The Feature Selection Problem: Traditional Methods and a New Algorithm. *In:* **Proceedings of the Tenth National Conference on Artificial Intelligence.** [S. l.: s. n.], 1992. v. 2, p. 129-134.

KOHONEN, T. The self-organizing map. **Proceedings of the IEEE**, IEEE, v. 78, n. 9, p. 1464-1480, set. 1990.

KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. Imagenet classification with deep convolutional neural networks. *In:* **Advances in neural information processing systems.** [S. l.: s. n.], 2012. p. 1097-1105.

KRUSKAL, J. B. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. **Psychometrika**, Springer, v. 29, n. 1, p. 1-27, 1964.

KUMAR, V.; MINZ, S. Feature selection. **SmartCR**, v. 4, n. 3, p. 211-229, 2014.

LE, Q. V. A tutorial on deep learning part 2: Autoencoders, convolutional neural networks and recurrent neural networks. **Google Brain**, Citeseer, p. 1-20, 2015.

LI, X.; GUO, Y. Bi-directional representation learning for multi-label classification. *In:* SPRINGER. **Joint European conference on machine learning and knowledge discovery in databases**. [S. l.], 2014. p. 209-224.
LIN, J.; GUNOPULOS, D. Dimensionality reduction by random projection and latent semantic indexing. In: CITESEER. **Proceedings of the Text Mining Workshop, at the 3rd SIAM International Conference on Data Mining**. [S. l.], 2003.

MELO, A.; PAULHEIM, H. Local and global feature selection for multilabel classification with binary relevance. **Artificial Intelligence Review**, Springer, v. 51, n. 1, p. 33-60. 2019.

METZ, J. **Abordagens para aprendizado semissupervisionado multirrótulo e hierárquico**. (Tese de Doutorado) - Universidade de São Paulo, São Paulo, Brasil, 2008.

MIKOLOV, T.; YIH, W.; ZWEIG, G. Linguistic regularities in continuous space word representations. *In:* **Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies**. [S. l.: s. n.], 2013. p. 746-751.

MIRANDA, V.; CASTRO, A. R. G.; LIMA, S. Diagnosing faults in power transformers with autoassociative neural networks and mean shift. **IEEE Transactions on Power Delivery**, IEEE, v. 27, n. 3, p. 1350-1357, 2012.

NAIR, V.; HINTON, G. E. Rectified linear units improve restricted boltzmann machines. In: **Proceedings of the 27th International Conference on Machine Learning**. [S. l.: s. n.], 2010. p. 807-814.

NETO, F. G. M. et al. Aprendizado profundo: conceitos, técnicas e estudo de caso de análise de imagens com Java. **III Escola Regional de Informática do Piauí**, ERIPI, Picos, Piauí, v. 1, n. 1, p. 465-486, 2017.

PAGANI, R. N.; KOVALESKI, J. L.; RESENDE, L. M. Methodi Ordinatio: a proposed methodology to select and rank relevant scientific papers encompassing the impact factor, number of citation, and year of publication. **Scientometrics**, Springer, v. 105, n. 3, p. 2109-2135, 1 dez. 2015.

PANG, Y.; WANG, S.; YUAN, Y. Learning Regularized LDA by Clustering. **IEEE Transactions on Neural Networks and Learning Systems,** IEEE, v. 25, n. 12, p. 2191-2201, 2014.

RIFAI, S. *et al*. Contractive auto-encoders: Explicit invariance during feature extraction. *In:* OMNIPRESS. **Proceedings of the 28th International Conference on International Conference on Machine Learning.** [S. l.], 2011. p. 833-840.

ROMDHANE, L. B.; FADHEL, N.; AYEB, B. An efficient approach for building customer profiles from business data. **Expert Systems with Applications**, Elsevier, v. 37, n. 2, p. 1573-1585, 2010.

SLAVKOV, I. *et al.* ReliefF for hierarchical multi-label classification. **Lecture Notes in Computer Science**, LNAI, v. 8399, p. 148-161, 2014.

SZALKAI, B.; GROLMUSZ, V. Near perfect protein multi-label classification with deep neural networks. **Methods**, Elsevier, v. 132, p. 50-56, 2018.

THEODORIDIS, S. *et al*. **Introduction to pattern recognition: a matlab approach**. [S. l.] Academic Press, 2010.

VAN DER MAATEN, L.; POSTMA, E.; VAN DEN HERIK, J. Dimensionality reduction: a comparative. **Journal of Machine Learning Research**, v. 10, n. 66-71, p. 13, 2009.

VENS, C. *et al*. Decision trees for hierarchical multi-label classification. **Machine learning**, Springer, v. 73, n. 2, p. 185, 2008.

VINCENT, P. *et al*. Extracting and composing robust features with denoising autoencoders. *In:* ACM. **Proceedings of the 25th international conference on Machine learning.** [S. l.], 2008. p. 1096-1103.

WAN, J. *et al*. Deep learning for content-based image retrieval: A comprehensive study. *In:* ACM. **Proceedings of the 22nd ACM international conference on Multimedia.** Nova York, 2014. p. 157-166.

WANG, J.; ZHU, X.; GONG, S. Discovering visual concept structure with sparse and incomplete tags. **Artificial Intelligence**, Elsevier v. 250, p. 16-36, 2017.

WANG, L. **Data Mining with Computational Intelligence**. Berlin, Heidelberg: Springer-Verlag, 2009.

WANG, Y.; YAO, H.; ZHAO, S. Auto-encoder based dimensionality reduction. **Neurocomputing**, Elsevier, v. 184, p. 232-242, 2016.

WITTEN, I. H. *et al*. **Data Mining: Practical machine learning tools and techniques**. [S. l.]: Morgan Kaufmann, 2016.

YAN, S.; WONG, K.-C. Elucidating high-dimensional cancer hallmark annotation via enriched ontology. **Journal of Biomedical Informatics**, Elsevier, v.73, p. 84-94, 2017.

YU, D. *et al*. Feature learning in deep neural networks-studies on speech recognition tasks. **arXiv preprint arXiv:1301.3605**, 2013.

ZEILER, M. D.; FERGUS, R. Visualizing and understanding convolutional networks. *In:* SPRINGER. **European Conference on Computer Vision**. [S. l.], 2014. p. 818-833.

ZHU, X.; JIN, Z.; JI, R. Learning high-dimensional multimedia data. **Multimedia Systems**, Springer, v. 23, n. 3, p. 281-283, 2017.

ZOU, Z. *et al*. mlDEEPre: Multi-functional enzyme function prediction with hierarchical multi-label deep learning. **Frontiers in Genetics**, v. 9, p. 714, 2019.

# APÊNDICE A - Resultados dos Experimentos no CLUS-HMC

## RESULTADOS DOS EXPERIMENTOS NO CLUS-HMC

A Tabela 22 apresenta os resultados das bases reduzidas pelo método FEAE-HMC para a base de dados de ID 2 quando avaliadas pelo classificador Clus-HMC. Os dados estão destacados em relação ao valor AUPRC de 0,452 obtido com todos os atributos originais (Tabela 11). O destaque é: em vermelho quando o valor obtido é inferior; em preto quando o valor é superior. Já o destaque em azul é o valor máximo obtido pelo conjunto reduzido.

**Tabela 22 - Medidas AUPRC para os conjuntos de dados reduzidos - Church**

| *Pr* | Atributos | 100 | 200 | 300 | 400 | 500 | 600 | 700 | 800 | 900 | 1000 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 10% | 24 | 0,441 | **0,441** | 0,440 | 0,440 | 0,440 | 0,437 | 0,437 | 0,435 | 0,437 | 0,438 |
| 20% | 21 | 0,436 | 0,435 | 0,440 | 0,439 | 0,440 | 0,441 | 0,439 | 0,440 | **0,441** | 0,440 |
| 30% | 18 | 0,440 | 0,445 | **0,446** | 0,443 | 0,443 | 0,442 | 0,443 | 0,441 | 0,439 | 0,442 |
| 40% | 16 | 0,443 | **0,444** | 0,442 | 0,438 | 0,438 | 0,438 | 0,439 | 0,440 | 0,436 | 0,433 |
| 50% | 13 | 0,430 | 0,437 | 0,439 | 0,443 | 0,443 | 0,439 | **0,443** | 0,442 | 0,440 | 0,441 |
| 60% | 10 | 0,435 | 0,441 | 0,440 | 0,439 | 0,443 | **0,445** | 0,443 | 0,438 | 0,441 | 0,442 |
| 70% | 8 | 0,438 | **0,442** | 0,442 | 0,441 | 0,441 | 0,438 | 0,437 | 0,436 | 0,436 | 0,439 |
| 80% | 5 | 0,434 | **0,439** | 0,435 | 0,439 | 0,429 | 0,430 | 0,433 | 0,437 | 0,433 | 0,431 |
| 90% | 2 | 0,432 | 0,439 | 0,439 | 0,439 | **0,440** | 0,437 | 0,437 | 0,437 | 0,437 | 0,435 |

**Fonte: Autoria própria**

Observa-se que entre as 90 bases reduzidas geradas para o conjunto de dados de ID 2, nenhuma foi superior a medida AUPRC de 0,452 da base original. Uma análise com algumas informações estatísticas observadas na Tabela 22, é apresentada na Tabela 23.

**Tabela 23 - Comparativo da medida AURPC nos diferentes percentuais - Church**

| Pr | Atributos | Média | Max | Desvio Padrão | % Média | % Max |
|---|---|---|---|---|---|---|
| 10% | 24 | 0,439 | 0,441 | *0,002* | -3,0% | -2,5% |
| 20% | 21 | 0,439 | 0,441 | *0,002* | -2,9% | -2,4% |
| 30% | 18 | **0,442** | **0,446** | *0,002* | -2,2% | -1,5% |
| 40% | 16 | 0,439 | 0,444 | *0,003* | -2,9% | -1,9% |
| 50% | 13 | 0,440 | 0,443 | *0,004* | -2,8% | -2,1% |
| 60% | 10 | 0,441 | 0,445 | *0,003* | -2,5% | -1,7% |
| 70% | 8 | 0,439 | 0,442 | *0,002* | -3,0% | -2,4% |
| 80% | 5 | 0,434 | 0,439 | *0,003* | -4,1% | -2,9% |
| 90% | 2 | 0,437 | 0,440 | *0,002* | -3,4% | -2,8% |
| | *Média* | *0,439* | *0,442* | *0,003* | *-3,0%* | *-2,2%* |

**Fonte: Autoria própria**

Nota-se que em média os valores obtidos de AUPRC para todos os percentuais de redução foi de 0,439, sendo que este valor é 3% inferior ao valor obtido sem a redução de dimensionalidade. Já o valor máximo observado, em média é de 0,442, valor este 2,2% inferior ao valor obtido sem o processo de redução proposto. O desvio padrão demonstra que o método FEAE-HMC, nestes resultados, é uniforme independentemente do tamanho da redução aplicada, indicando que os valores obtidos estão condensados próximos da média.

Ao se analisar estatisticamente os resultados da Tabela 22, pelo Teste de Friedman, têm-se $x_F^2$= 51,92 e $F_F$= 12,269. De acordo com o valor tabelado na distribuição Snedecor, com α = 0,05, encontra-se o valor de F (9;81) = 1,998. Desse modo a hipótese nula pode ser rejeitada, pois $F_F$ > 1,998.

Rejeitada a hipótese nula, identifica-se que existem diferenças significativas de desempenho por meio do valor obtido para a distância crítica pelo Teste de Nemenyi, que nesse caso foi de 4,28. Considerando o nível de significância de 95%, o resultado obtido com a base com todos os atributos, é estatisticamente superior aos resultados obtidos em 7 bases reduzidas. Nas bases com 18 e 10 atributos não há diferença estatística significativa.

A Tabela 24, apresenta os resultados obtidos para o conjunto de dados de ID 4, sendo que os valores estão destacados, da mesma maneira descrita anteriormente, porém em relação ao valor 0,456 de AUPRC obtido por essa base com os 79 atributos originais.

**Tabela 24 - Medidas AUPRC para as bases reduzidas do conjunto de dados Eisen**

| ***Pr*** | **Atributos** | **100** | **200** | **300** | **400** | **500** | **600** | **700** | **800** | **900** | **1000** |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 10% | 71 | 0,457 | 0,468 | **0,470** | 0,464 | 0,469 | 0,470 | 0,467 | 0,459 | 0,465 | 0,458 |
| 20% | 63 | 0,456 | 0,467 | 0,471 | **0,475** | 0,475 | 0,472 | 0,465 | 0,449 | 0,464 | 0,458 |
| 30% | 55 | 0,457 | 0,467 | **0,473** | 0,454 | 0,468 | 0,467 | 0,466 | 0,470 | 0,458 | 0,467 |
| 40% | 47 | 0,458 | 0,469 | 0,465 | **0,469** | 0,463 | 0,468 | 0,467 | 0,458 | 0,458 | 0,463 |
| 50% | 39 | 0,461 | 0,465 | **0,472** | 0,468 | 0,467 | 0,463 | 0,462 | 0,466 | 0,467 | 0,460 |
| 60% | 31 | 0,463 | 0,469 | 0,468 | **0,476** | 0,475 | 0,465 | 0,467 | 0,460 | 0,467 | 0,474 |
| 70% | 23 | 0,464 | 0,471 | 0,474 | 0,472 | **0,475** | 0,470 | 0,465 | 0,470 | 0,461 | 0,467 |
| 80% | 15 | 0,455 | 0,466 | 0,470 | 0,470 | 0,472 | 0,464 | **0,474** | 0,465 | 0,473 | 0,470 |
| 90% | 7 | 0,451 | 0,463 | 0,472 | 0,471 | 0,468 | 0,476 | **0,476** | 0,473 | 0,468 | 0,467 |

**Fonte: Autoria própria**

Entre as 90 bases reduzidas geradas para o conjunto de dados de ID 4, 94,44% obtiveram valor superior a medida AUPRC de 0,456 da base original, sendo que esses valores estão localizados, principalmente, nas épocas 300 e 400. Uma análise com algumas informações estatísticas é apresentada na Tabela 25.

**Tabela 25 - Comparativo da medida AURPC nos diferentes percentuais - Eisen**

| **Pr** | **Atributos** | **Média** | **Max** | **Desvio Padrão** | **% Média** | **% Max** |
|---|---|---|---|---|---|---|
| 10% | 71 | 0,465 | 0,470 | 0,005 | 1,9% | 2,9% |
| 20% | 63 | 0,465 | 0,475 | 0,008 | 2,0% | 4,2% |
| 30% | 55 | 0,465 | 0,473 | 0,006 | 1,9% | 3,8% |
| 40% | 47 | 0,464 | 0,469 | 0,004 | 1,7% | 2,9% |
| 50% | 39 | 0,465 | 0,472 | 0,003 | 2,0% | 3,4% |
| 60% | 31 | 0,468 | 0,476 | 0,005 | 2,7% | 4,3% |
| 70% | 23 | **0,469** | 0,475 | 0,004 | 2,8% | 4,1% |
| 80% | 15 | 0,468 | 0,474 | 0,005 | 2,6% | 3,8% |
| 90% | 7 | 0,469 | **0,476** | 0,007 | 2,7% | 4,4% |
| | *Média* | *0,466* | *0,473* | *0,005* | ***2,2%*** | ***3,8%*** |

**Fonte: Autoria própria**

Observa-se que em média os valores obtidos de AUPRC para todos os percentuais de redução foi de 0,466, sendo este valor é 2,2% superior ao valor obtido com os atributos originais. Já o valor máximo foi em média de 0,473, valor 3,8% superior ao valor obtido sem o processo de redução de dimensionalidade. O desvio padrão demonstra que o método FEAE-HMC, nestes resultados, também é uniforme independentemente do tamanho da redução aplicada, indicando que os valores obtidos estão condensados próximos da média.

Ao se analisar estatisticamente os resultados, têm-se $x_F^2$= 30,272 e $F_F$= 4,56. De acordo com o valor tabelado na distribuição Snedecor, com α = 0,05, encontra-se o valor de F (9;81) = 1,998. Desse modo a hipótese nula pode ser rejeitada, pois $F_F$ >1,998. Isso significa que há diferença estatística significativa entre o desempenho obtido entre as bases reduzidas e a base completa.

Existe diferença significativa de desempenho por meio do valor 4,28 obtido para a distância crítica pelo Teste de Nemenyi. Considerando o nível de significância de 95%, os resultados com as bases reduzidas pelo método FEAE-HMC (nas épocas selecionadas) para a base de ID 4, são estatisticamente superiores a medida AUPRC da base completa nos conjuntos com 31, 23, 15, e 7 atributos. Para as demais bases, observa-se que não há diferença estatística significativa quando analisados os desempenhos entre a base original e as bases reduzidas.

A Tabela 26, apresenta os resultados obtidos para o conjunto de dados de ID 5, sendo que os valores destacados, estão em conformidade com a maneira descrita para as tabelas anteriores. Essa base com seus 551 atributos obteve o valor 0,473 de medida AUPRC.

**Tabela 26 - Medidas AUPRC para as bases reduzidas do conjunto de dados Expr**

| ***Pr*** | **Atributos** | **100** | **200** | **300** | **400** | **500** | **600** | **700** | **800** | **900** | **1000** |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 10% | 495 | 0,449 | 0,465 | 0,458 | **0,468** | 0,463 | 0,462 | 0,463 | 0,463 | 0,465 | 0,467 |
| 20% | 440 | 0,450 | 0,456 | 0,470 | **0,473** | 0,469 | 0,466 | 0,466 | 0,468 | 0,467 | 0,468 |
| 30% | 385 | 0,447 | 0,472 | 0,453 | **0,472** | 0,469 | 0,460 | 0,467 | 0,467 | 0,467 | 0,462 |
| 40% | 330 | 0,450 | 0,457 | 0,454 | 0,453 | **0,467** | 0,459 | 0,459 | 0,460 | 0,465 | 0,455 |
| 50% | 275 | 0,461 | **0,469** | 0,466 | 0,463 | 0,467 | 0,460 | 0,466 | 0,465 | 0,462 | 0,448 |
| 60% | 220 | 0,457 | 0,466 | 0,467 | 0,467 | **0,469** | 0,463 | 0,468 | 0,466 | 0,465 | 0,464 |
| 70% | 165 | 0,467 | **0,468** | 0,467 | 0,462 | 0,468 | 0,464 | 0,464 | 0,466 | 0,464 | 0,461 |
| 80% | 110 | 0,462 | **0,473** | 0,472 | 0,456 | 0,462 | 0,465 | 0,461 | 0,453 | 0,463 | 0,463 |
| 90% | 55 | 0,454 | **0,471** | 0,467 | 0,458 | 0,459 | 0,465 | 0,463 | 0,466 | 0,467 | 0,459 |

**Fonte: Autoria própria**

Observa-se que entre as 90 bases reduzidas geradas para o conjunto de dados de ID 5, nenhuma foi superior a medida AUPRC de 0,473 da base original. Uma análise com algumas informações estatísticas observadas na Tabela 26, é apresentada na Tabela 27.

**Tabela 27 - Comparativo da medida AURPC nos diferentes percentuais - Expr**

| Pr | Atributos | Média | Max | Desvio Padrão | % Média | % Max |
|---|---|---|---|---|---|---|
| 10% | 495 | 0,462 | 0,468 | 0,005 | -2,2% | -1,1% |
| 20% | 440 | **0,465** | **0,473** | 0,007 | -1,6% | 0,1% |
| 30% | 385 | 0,464 | 0,472 | 0,008 | -1,9% | -0,1% |
| 40% | 330 | 0,458 | 0,467 | 0,005 | -3,1% | -1,2% |
| 50% | 275 | 0,463 | 0,469 | 0,005 | -2,1% | -0,8% |
| 60% | 220 | 0,465 | 0,469 | 0,003 | -1,6% | -0,8% |
| 70% | 165 | 0,465 | 0,468 | 0,002 | -1,6% | -1,0% |
| 80% | 110 | 0,463 | 0,473 | 0,006 | -2,1% | 0,0% |
| 90% | 55 | 0,463 | 0,471 | 0,005 | -2,1% | -0,4% |
| | *Média* | *0,463* | *0,470* | *0,003* | ***-2,0%*** | ***-0,6%*** |

**Fonte: Autoria própria**

Observa-se que em média os valores obtidos de AUPRC para todos os percentuais de redução foi de 0,463, sendo esse valor em média 2,0% inferior ao valor obtido com os 551 atributos originais. O valor máximo foi em média de 0,470, valor 0,2 % inferior ao valor obtido sem o processo de redução de dimensionalidade. Vale destacar que com 110 atributos, que equivale uma redução de 80%, foi obtido o mesmo valor AUPRC da base com 551 atributos.

Ao se analisar estatisticamente os resultados da Tabela 26, têm-se $x_F^2$= 39,83 e $F_F$= 7,145. De acordo com o valor tabelado na distribuição Snedecor, com α = 0,05, encontra-se o valor de F (9;81) = 1,998. Desse modo a hipótese nula pode ser rejeitada, pois $F_F$ > 1,998.

Há diferença significativa de desempenho por meio do valor 4,28 obtido para a distância crítica pelo Teste de Nemenyi, considerando o nível de significância de 95%, os resultados obtidos com a base com todos os atributos, é estatisticamente superior aos resultados obtidos em 7 bases reduzidas pelo método FEAE-HMC. Nas bases com 440 e 220 atributos não há diferença estatística significativa.

A Tabela 28, apresenta os resultados obtidos para o conjunto de dados de ID 7, sendo que os valores estão destacados, da mesma maneira descrita anteriormente, porém em relação ao valor 0,438 de AUPRC obtido por essa base com os 52 atributos originais.

**Tabela 28 - Medidas AUPRC para as bases reduzidas do conjunto de dados Gasch2**

| ***Pr*** | **Atributos** | **100** | **200** | **300** | **400** | **500** | **600** | **700** | **800** | **900** | **1000** |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 10% | 46 | 0,438 | 0,443 | 0,446 | **0,450** | 0,444 | 0,439 | 0,442 | 0,441 | 0,445 | 0,445 |
| 20% | 41 | 0,426 | 0,441 | 0,443 | 0,446 | 0,443 | 0,444 | 0,445 | **0,447** | 0,444 | 0,445 |
| 30% | 36 | 0,428 | 0,445 | 0,445 | 0,444 | **0,447** | 0,433 | 0,441 | 0,443 | 0,444 | 0,447 |
| 40% | 31 | 0,435 | 0,441 | 0,445 | 0,432 | 0,442 | 0,428 | 0,439 | 0,434 | 0,443 | **0,445** |
| 50% | 26 | 0,431 | 0,441 | 0,446 | 0,443 | 0,437 | 0,445 | 0,446 | **0,448** | 0,443 | 0,447 |
| 60% | 20 | 0,427 | 0,442 | 0,441 | 0,443 | 0,437 | 0,432 | 0,443 | **0,443** | 0,441 | 0,443 |
| 70% | 15 | 0,426 | 0,434 | 0,440 | 0,443 | 0,443 | **0,448** | 0,447 | 0,448 | 0,443 | 0,448 |
| 80% | 10 | 0,427 | 0,439 | 0,443 | **0,447** | 0,444 | 0,446 | 0,446 | 0,446 | 0,446 | 0,441 |
| 90% | 5 | 0,423 | 0,435 | 0,437 | 0,444 | 0,440 | **0,446** | 0,445 | 0,442 | 0,443 | 0,442 |

**Fonte: Autoria própria**

Entre as 90 bases reduzidas geradas para o conjunto de dados de ID 7, 78,88% obtiveram valor superior a medida AUPRC de 0,438 da base original. Uma análise com algumas informações estatísticas é apresentada na Tabela 29.

**Tabela 29 - Comparativo da medida AURPC nos diferentes percentuais - Gasch2**

| Pr | Atributos | Média | Max | Desvio Padrão | % Média | % Max |
|---|---|---|---|---|---|---|
| 10% | 46 | **0,443** | **0,450** | 0,003 | 1,2% | 2,6% |
| 20% | 41 | 0,442 | 0,447 | 0,006 | 0,9% | 1,9% |
| 30% | 36 | 0,442 | 0,447 | 0,006 | 0,8% | 2,0% |
| 40% | 31 | 0,438 | 0,445 | 0,006 | 0,0% | 1,5% |
| 50% | 26 | 0,443 | 0,448 | 0,005 | 1,0% | 2,3% |
| 60% | 20 | 0,439 | 0,443 | 0,005 | 0,2% | 1,2% |
| 70% | 15 | 0,442 | 0,448 | 0,007 | 0,9% | 2,3% |
| 80% | 10 | 0,442 | 0,447 | 0,006 | 0,9% | 2,0% |
| 90% | 5 | 0,440 | 0,446 | 0,006 | 0,3% | 1,7% |
| | *Média* | *0,441* | *0,447* | *0,005* | ***0,7%*** | ***1,9%*** |

**Fonte: Autoria própria**

Observa-se que em média os valores obtidos de AUPRC para todos os percentuais de redução foi de 0,441, sendo este valor é 0,7% superior ao valor obtido com os atributos originais. Já o valor máximo foi em média de 0,447, valor 1,9% superior ao valor obtido sem o processo de redução de dimensionalidade. O desvio padrão demonstra que nestes resultados, o método também é uniforme independentemente do tamanho da redução aplicada, indicando que os valores obtidos estão condensados próximos da média.

Ao se analisar estatisticamente os resultados da Tabela 24, pelo Teste de Friedman, têm-se $x_F^2$= 23,40 e $F_F$= 3,163. De acordo com o valor tabelado na distribuição Snedecor, com $\alpha = 0{,}05$, encontra-se o valor de F (9;81) = 1,998. Desse modo a hipótese nula pode ser rejeitada, pois $F_F$ >1,998. Isso significa que há diferença estatística significativa entre o desempenho obtido entre a base completa e as bases reduzidas, por meio do valor 4,28 obtido para a distância crítica pelo Teste de Nemenyi.

Considerando o nível de significância de 95%, o resultado com a base reduzida é estatisticamente superior a medida AUPRC da base completa no conjunto com 46 atributos. Para as demais bases, observa-se que não há diferença estatística significativa quando analisados os desempenhos entre a base original e as bases reduzidas.

A Tabela 30, apresenta os resultados obtidos para o conjunto de dados de ID 8, sendo que os valores estão destacados, da mesma maneira descrita anteriormente, porém em relação ao valor 0,421 de AUPRC obtido por essa base com os 69 atributos originais.

**Tabela 30 - Medidas AUPRC para as bases reduzidas do conjunto de dados Pheno**

| *Pr* | Atributos | 100 | 200 | 300 | 400 | 500 | 600 | 700 | 800 | 900 | 1000 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 10% | 62 | 0,421 | 0,418 | 0,419 | **0,421** | 0,417 | 0,418 | **0,421** | 0,418 | **0,421** | 0,420 |
| 20% | 55 | 0,421 | 0,415 | 0,418 | **0,421** | **0,421** | 0,420 | 0,421 | 0,420 | 0,418 | **0,421** |
| 30% | 48 | **0,421** | **0,421** | 0,420 | **0,421** | 0,419 | 0,419 | 0,420 | 0,420 | 0,420 | 0,421 |
| 40% | 41 | 0,418 | 0,420 | 0,420 | 0,421 | 0,421 | 0,421 | **0,422** | 0,419 | 0,420 | 0,420 |
| 50% | 34 | **0,422** | 0,420 | 0,422 | 0,420 | 0,421 | 0,421 | 0,421 | 0,421 | 0,421 | 0,418 |
| 60% | 27 | 0,421 | 0,417 | 0,421 | 0,420 | **0,423** | 0,422 | 0,421 | 0,421 | 0,421 | 0,420 |
| 70% | 20 | **0,422** | 0,419 | 0,419 | 0,421 | 0,420 | 0,420 | 0,421 | 0,421 | 0,420 | 0,419 |
| 80% | 13 | 0,421 | 0,419 | 0,421 | 0,421 | 0,421 | 0,418 | 0,414 | 0,420 | **0,422** | 0,421 |
| 90% | 6 | 0,421 | 0,421 | 0,416 | 0,415 | **0,422** | 0,414 | 0,421 | 0,417 | 0,413 | 0,418 |

**Fonte: Autoria própria**

Entre as 90 bases reduzidas geradas para o conjunto de dados de ID 8, 32,2% obtiveram valor superior a medida AUPRC de 0,421 da base original. Uma análise com algumas informações estatísticas é apresentada na Tabela 31.

**Tabela 31 - Comparativo da medida AURPC nos diferentes percentuais - Pheno**

| Pr | Atributos | Média | Max | Desvio Padrão | % Média | % Max |
|---|---|---|---|---|---|---|
| 10% | 62 | 0,420 | 0,421 | 0,002 | -0,4% | 0,0% |
| 20% | 55 | 0,420 | 0,421 | 0,002 | -0,4% | 0,0% |
| 30% | 48 | 0,420 | 0,421 | 0,001 | -0,2% | 0,0% |
| 40% | 41 | 0,420 | 0,422 | 0,001 | -0,3% | 0,1% |
| 50% | 34 | **0,421** | 0,422 | 0,001 | -0,1% | 0,2% |
| 60% | 27 | 0,421 | **0,423** | 0,002 | -0,2% | 0,4% |
| 70% | 20 | 0,420 | 0,422 | 0,001 | -0,3% | 0,0% |
| 80% | 13 | 0,420 | 0,422 | 0,002 | -0,3% | 0,1% |
| 90% | 6 | 0,418 | 0,422 | 0,003 | -0,9% | 0,0% |
| | *Média* | *0,420* | *0,422* | *0,002* | ***-0,4%*** | ***0,1%*** |

**Fonte: Autoria própria**

Observa-se que em média os valores obtidos de AUPRC para todos os percentuais de redução foi de 0,420, sendo este valor é 0,4% inferior ao valor obtido com os 69 atributos originais. Já o valor máximo foi em média de 0,422, valor 0,1% superior ao valor obtido sem o processo de redução de dimensionalidade. O desvio padrão demonstra que o método FEAE-HMC, nestes resultados, também é uniforme independentemente do tamanho da redução aplicada, indicando que os valores obtidos estão condensados próximos da média.

Ao se analisar estatisticamente os resultados da Tabela 31, têm-se $x_F^2$= 12,60 e $F_F$= 1,465. De acordo com o valor tabelado na distribuição Snedecor, com α = 0,05, encontra-se o valor de F (9;81) = 1,998. Desse modo a hipótese nula não pode ser rejeitada, pois $F_F$ < 1,998. Isso significa que não há diferença estatística significativa entre o desempenho obtido entre a base completa e as bases reduzidas.

A Tabela 32 apresenta os resultados das bases reduzidas pelo método FEAE-HMC para a base de dados de ID 9 quando avaliadas pelo classificador Clus-HMC. Os dados estão destacados em relação ao valor AUPRC de 0,470 obtido com os 478 atributos originais

**Tabela 32 - Medidas AUPRC para os conjuntos de dados reduzidos - Seq**

| *Pr* | Atributos | 100 | 200 | 300 | 400 | 500 | 600 | 700 | 800 | 900 | 1000 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 10% | 430 | 0,438 | 0,448 | 0,445 | 0,454 | 0,443 | 0,438 | 0,443 | 0,431 | 0,436 | **0,459** |
| 20% | 382 | **0,460** | 0,454 | 0,438 | 0,435 | 0,445 | 0,451 | 0,446 | 0,447 | 0,438 | 0,454 |
| 30% | 334 | **0,459** | 0,453 | 0,443 | 0,443 | 0,435 | 0,443 | 0,447 | 0,448 | 0,446 | 0,443 |
| 40% | 286 | **0,461** | 0,460 | 0,445 | 0,429 | 0,459 | 0,454 | 0,445 | 0,454 | 0,446 | 0,457 |
| 50% | 239 | **0,462** | 0,455 | 0,447 | 0,444 | 0,444 | 0,443 | 0,446 | 0,445 | 0,445 | 0,456 |
| 60% | 191 | **0,455** | 0,447 | 0,444 | 0,442 | 0,441 | 0,452 | 0,440 | 0,454 | 0,446 | 0,453 |
| 70% | 143 | **0,457** | 0,444 | 0,451 | 0,443 | 0,437 | 0,447 | 0,437 | 0,442 | 0,438 | 0,439 |
| 80% | 95 | **0,458** | 0,447 | 0,428 | 0,439 | 0,444 | 0,442 | 0,447 | 0,450 | 0,447 | 0,444 |
| 90% | 47 | **0,452** | 0,451 | 0,442 | 0,441 | 0,442 | 0,429 | 0,436 | 0,442 | 0,437 | 0,435 |

**Fonte: Autoria própria**

Observa-se que entre as 90 bases reduzidas geradas para o conjunto de dados de ID 9, nenhuma foi superior a medida AUPRC de 0,470 da base original. Uma análise com algumas informações estatísticas é apresentada na Tabela 33.

**Tabela 33 - Comparativo da medida AURPC nos diferentes percentuais - Seq**

| Pr | Atributos | Média | Max | Desvio Padrão | % Média | % Max |
|---|---|---|---|---|---|---|
| 10% | 430 | 0,443 | 0,459 | 0,008 | -5,6% | -2,4% |
| 20% | 382 | 0,447 | 0,460 | 0,008 | -4,9% | -2,0% |
| 30% | 334 | 0,446 | 0,459 | 0,006 | -5,1% | -2,4% |
| 40% | 286 | **0,451** | 0,461 | 0,010 | -4,0% | -2,0% |
| 50% | 239 | 0,449 | **0,462** | 0,007 | -4,5% | -1,6% |
| 60% | 191 | 0,448 | 0,455 | 0,006 | -4,7% | -3,1% |
| 70% | 143 | 0,443 | 0,457 | 0,007 | -5,6% | -2,7% |
| 80% | 95 | 0,445 | 0,458 | 0,008 | -5,3% | -2,5% |
| 90% | 47 | 0,441 | 0,452 | 0,007 | -6,2% | -3,7% |
| | *Média* | *0,446* | *0,458* | *0,007* | ***-5,1%*** | ***-2,5%*** |

**Fonte: Autoria própria**

Nota-se que em média os valores obtidos de AUPRC para todos os percentuais de redução foi de 0,446, sendo que este valor é 5,1% inferior ao valor obtido sem a redução de dimensionalidade. Já o valor máximo observado, em média é de 0,458, valor este 2,5% inferior ao valor obtido sem o processo de redução.

Ao se analisar estatisticamente os resultados da Tabela 33, pelo Teste de Friedman, têm-se $x_F^2$= 41,18 e $F_F$= 7,59. De acordo com o valor tabelado na distribuição Snedecor, com $\alpha = 0,05$, encontra-se o valor de F (9;81) = 1,998. Desse modo a hipótese nula pode ser rejeitada, pois $F_F > 1,998$.

Rejeitada a hipótese nula, identifica-se que há diferença significativa de desempenho por meio do valor obtido para a distância crítica pelo Teste de Nemenyi, que nesse caso foi de 4,28. Considerando o nível de significância de 95%, o resultado obtido com a base com todos os atributos, é estatisticamente superior aos resultados obtidos em 7 bases reduzidas pelo método FEAE-HMC. Nas bases com 286 e 239 atributos não há diferença estatística significativa.

A Tabela 34 apresenta os resultados das bases reduzidas para a base de dados de ID 10. O destaque está em relação ao valor AUPRC de 0,468 obtido com os 80 atributos originais.

**Tabela 34 - Medidas AUPRC para os conjuntos de dados reduzidos - Spo**

| *Pr* | Atributos | 100 | 200 | 300 | 400 | 500 | 600 | 700 | 800 | 900 | 1000 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 10% | 72 | 0,449 | **0,454** | 0,453 | 0,448 | 0,439 | 0,446 | 0,450 | 0,449 | 0,449 | 0,443 |
| 20% | 64 | 0,449 | 0,448 | 0,451 | 0,451 | 0,446 | **0,455** | 0,439 | 0,450 | 0,446 | 0,446 |
| 30% | 56 | 0,450 | 0,452 | 0,447 | 0,451 | 0,442 | 0,447 | 0,454 | **0,455** | 0,450 | 0,451 |
| 40% | 48 | 0,450 | 0,453 | **0,457** | 0,451 | 0,454 | 0,442 | 0,451 | 0,450 | 0,449 | 0,445 |
| 50% | 40 | 0,450 | **0,457** | 0,446 | 0,454 | 0,454 | 0,448 | 0,449 | 0,448 | 0,449 | 0,451 |
| 60% | 32 | 0,451 | 0,452 | 0,451 | 0,448 | **0,452** | 0,445 | 0,436 | 0,438 | 0,447 | 0,445 |
| 70% | 24 | 0,451 | 0,446 | **0,457** | 0,454 | 0,449 | 0,444 | 0,447 | 0,457 | 0,449 | 0,448 |
| 80% | 16 | 0,448 | 0,448 | 0,445 | 0,441 | 0,447 | **0,451** | 0,442 | 0,443 | 0,447 | 0,446 |
| 90% | 8 | 0,440 | 0,448 | 0,457 | **0,458** | 0,446 | 0,455 | 0,447 | 0,441 | 0,449 | 0,440 |

**Fonte: Autoria própria**

Observa-se que entre as 90 bases reduzidas geradas para o conjunto de dados de ID 10, nenhuma foi superior a medida AUPRC de 0,468 da base original. Uma análise com algumas informações estatísticas é apresentada na Tabela 35.

**Tabela 35 - Comparativo da medida AURPC nos diferentes percentuais - Spo**

| Pr | Atributos | Média | Max | Desvio Padrão | % Média | % Max |
|---|---|---|---|---|---|---|
| 10% | 72 | 0,454 | 0,448 | 0,004 | -4,2% | -2,9% |
| 20% | 64 | 0,455 | 0,448 | 0,005 | -4,3% | -2,7% |
| 30% | 56 | 0,455 | 0,450 | 0,004 | -3,9% | -2,7% |
| 40% | 48 | 0,457 | 0,450 | 0,004 | -3,8% | -2,3% |
| 50% | 40 | 0,457 | **0,451** | 0,003 | -3,7% | -2,4% |
| 60% | 32 | 0,452 | 0,446 | 0,006 | -4,6% | -3,4% |
| 70% | 24 | 0,457 | 0,450 | 0,005 | -3,8% | -2,3% |
| 80% | 16 | 0,451 | 0,446 | 0,003 | -4,7% | -3,6% |
| 90% | 8 | **0,458** | 0,448 | 0,007 | -4,2% | -2,1% |
| | *Média* | 0,455 | 0,449 | 0,004 | -4,1% | -2,7% |

**Fonte: Autoria própria**

Nota-se que em média os valores obtidos de AUPRC para todos os percentuais de redução foi de 0,455, sendo que este valor é 4,1% inferior ao valor obtido sem a aplicação do método FEAE-HMC. Já o valor máximo observado, em média é de 0,449, valor este 2,7% inferior ao valor obtido sem o processo de redução.

Ao se analisar estatisticamente os resultados da Tabela 33, têm-se $x_F^2$= 35,24 e $F_F$= 5,79. De acordo com o valor tabelado na distribuição Snedecor, com α = 0,05, encontra-se o valor de F (9;81) = 1,998. Desse modo a hipótese nula pode ser rejeitada, pois $F_F$ > 1,998.

Rejeitada a hipótese nula, há diferença significativa de desempenho por meio do valor obtido para a distância crítica de 4,28 pelo Teste de Nemenyi. Considerando o nível de significância de 95%, o resultado obtido com a base com todos os atributos, é estatisticamente superior aos resultados obtidos em 7 bases reduzidas pelo método FEAE-HMC. Já nas bases com 56 e 40 atributos não há diferença estatística significativa.