



UNIVERSIDADE FEDERAL DE GOIÁS (UFG)/  
UNIVERSIDADE FEDERAL DE CATALÃO (UFCAT) em implantação  
UNIDADE ACADÊMICA DE MATEMÁTICA E TECNOLOGIA  
PROGRAMA DE PÓS GRADUAÇÃO EM  
MODELAGEM E OTIMIZAÇÃO

CAMILA PEREIRA DA SILVA

PREDITOR DE DESEMPENHO DE ESTUDANTES DO CURSO DE MATEMÁTICA  
INDUSTRIAL/UFG VIA REDES NEURAIAS MLP E SOM

CATALÃO  
2021



UNIVERSIDADE FEDERAL DE GOIÁS  
UNIDADE ACADÊMICA ESPECIAL DE MATEMÁTICA E TECNOLOGIA

## TERMO DE CIÊNCIA E DE AUTORIZAÇÃO (TECA) PARA DISPONIBILIZAR VERSÕES ELETRÔNICAS DE TESES

### E DISSERTAÇÕES NA BIBLIOTECA DIGITAL DA UFG

Na qualidade de titular dos direitos de autor, autorizo a Universidade Federal de Goiás (UFG) a disponibilizar, gratuitamente, por meio da Biblioteca Digital de Teses e Dissertações (BDTD/UFG), regulamentada pela Resolução CEPEC nº 832/2007, sem ressarcimento dos direitos autorais, de acordo com a [Lei 9.610/98](#), o documento conforme permissões assinaladas abaixo, para fins de leitura, impressão e/ou download, a título de divulgação da produção científica brasileira, a partir desta data.

O conteúdo das Teses e Dissertações disponibilizado na BDTD/UFG é de responsabilidade exclusiva do autor. Ao encaminhar o produto final, o autor(a) e o(a) orientador(a) firmam o compromisso de que o trabalho não contém nenhuma violação de quaisquer direitos autorais ou outro direito de terceiros.

#### 1. Identificação do material bibliográfico

Dissertação      Tese

#### 2. Nome completo do autor

Camila Pereira da Silva

#### 3. Título do trabalho

PREDITOR DE DESEMPENHO DE ESTUDANTES DO CURSO DE MATEMÁTICA INDUSTRIAL/UFG VIA REDES NEURAIS MLP E SOM

#### 4. Informações de acesso ao documento (este campo deve ser preenchido pelo orientador)

Concorda com a liberação total do documento  SIM      NÃO<sup>1</sup>

[1] Neste caso o documento será embargado por até um ano a partir da data de defesa. Após esse período, a possível disponibilização ocorrerá apenas mediante:

**a)** consulta ao(a) autor(a) e ao(a) orientador(a);

**b)** novo Termo de Ciência e de Autorização (TECA) assinado e inserido no arquivo da tese ou dissertação.

O documento não será disponibilizado durante o período de embargo.

Casos de embargo:

- Solicitação de registro de patente;
- Submissão de artigo em revista científica;
- Publicação como capítulo de livro;
- Publicação da dissertação/tese em livro.

**Obs. Este termo deverá ser assinado no SEI pelo orientador e pelo autor.**



Documento assinado eletronicamente por **Jose Dos Reis Vieira De Moura Junior, Orientador**, em 03/02/2022, às 21:02, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do [Decreto nº 10.543, de 13 de novembro de 2020](#).

Documento assinado eletronicamente por **CAMILA PEREIRA DA SILVA, Discente**, em 03/02/2022, às 21:14, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do [Decreto nº 10.543](#),



[de 13 de novembro de 2020.](#)

---



A autenticidade deste documento pode ser conferida no site

[https://sei.ufg.br/sei/controlador\\_externo.php?](https://sei.ufg.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0)

[acao=documento\\_conferir&id\\_orgao\\_acesso\\_externo=0](https://sei.ufg.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0), informando o código verificador **2582010** e

o código CRC **7AC9BBC4**.

---

Referência: Processo nº 23070.067906/2021-19

SEI nº 2582010

CAMILA PEREIRA DA SILVA

PREDITOR DE DESEMPENHO DE  
ESTUDANTES DO CURSO DE MATEMÁTICA  
INDUSTRIAL/UFG VIA REDES NEURAIAS MLP  
E SOM

Dissertação apresentada ao Programa de Pós-Graduação em Modelagem e Otimização, da Unidade Acadêmica Especial de Matemática e Tecnologia, da Universidade Federal de Goiás (UFG) / Universidade Federal de Catalão (UFCAT) em implantação, como requisito para obtenção do título de Mestre em Modelagem e Otimização. Área de concentração: Modelagem e Otimização  
Linha de pesquisa: Modelagem Computacional e Otimização.

Orientador: Professor Doutor José dos Reis Vieira de Moura Júnior

Ficha de identificação da obra elaborada pelo autor, através do Programa de Geração Automática do Sistema de Bibliotecas da UFCAT.

Silva, Camila Pereira da  
PREDITOR DE DESEMPENHO DE ESTUDANTES DO CURSO  
DE MATEMÁTICA INDUSTRIAL/UFG VIA REDES NEURAIAS MLP E  
SOM / Camila Pereira da Silva. - 2021.  
88, f.: il.

Orientador: Prof. Dr. José dos Reis Vieira de Moura Júnior.  
Dissertação (Mestrado) - Universidade Federal de Catalão, Instituto  
de Matemática e Tecnologia, Catalão, Programa de Pós-Graduação  
em Modelagem e Otimização, Catalão, 2021.

Bibliografia. Anexos.

Inclui siglas, abreviaturas, símbolos, gráfico, tabelas, lista de  
figuras.

1. Mineração de Dados Educacionais. 2. Desempenho Acadêmico.  
3. Redes Neurais Artificiais. 4. Modelagem. 5. Ferramenta de auxílio..  
I. Júnior, José dos Reis Vieira de Moura , orient. II. Título.

CDU 517



UNIVERSIDADE FEDERAL DE GOIÁS

UNIDADE ACADÊMICA ESPECIAL DE MATEMÁTICA E TECNOLOGIA

**ATA DE DEFESA DE DISSERTAÇÃO**

Ata nº 11 da sessão de Defesa de Dissertação de **Camila Pereira da Silva**, que confere o título de **Mestre(a) em Modelagem e Otimização**, na área de concentração em **Modelagem e Otimização**.

Aos quatorze dias do mês de dezembro de 2021, a partir das 16 h 30 min, na sala google meeting, realizou-se a sessão pública de Defesa de Dissertação, intitulada "*PREDITOR DE DESEMPENHO DE ESTUDANTES DO CURSO DE MATEMÁTICA INDUSTRIAL VIA REDES NEURAIS MLP E SOM*" nas dependências da Universidade Federal de Catalão, onde os programas de pós-graduação *Stricto Sensu* em funcionamento encontram-se provisoriamente vinculados à Universidade Federal de Goiás, em virtude de procedimentos técnicos relacionados à CAPES, já sendo realizada a transferência da Biblioteca Digital de Dissertações e Teses (BDTD). Assim, justifica-se os nomes das instituições neste documento, uma no cabeçalho (UFG), outra no corpo do texto (UFCAT). Os trabalhos foram instalados pelo(a) **Orientador(a), Professor(a) Doutor(a) José dos Reis Vieira de Moura Júnior (orientador) (PPGMO/IMTec/UFCAT)**, com a participação dos demais membros da Banca Examinadora: **Professor(a) Doutor(a) Donald Mark Santee (IMTec/UFCAT), membro titular externo ao Programa; Professor(a) Doutor(a) Marcelo Henrique Stoppa (PPGMO/IMTec/UFCAT), membro titular interno**. A Banca Examinadora reuniu-se em sessão secreta a fim de concluir o julgamento da Dissertação, tendo sido o(a) candidato(a) (X ) Aprovado(a) ( ) Reprovado(a) pelos seus membros. Proclamados os resultados pelo(a) Professor(a) Doutor(a) **José dos Reis Vieira de Moura Júnior**, Presidente da Banca Examinadora, foram encerrados os trabalhos e, para constar, lavrou-se a presente ata que é assinada pelos Membros da Banca Examinadora, aos quatorze dias do mês de dezembro de dois mil e vinte e um.

Obs: "*Banca Examinadora de Qualificação/Defesa Pública de Dissertação/Tese realizada em conformidade com a Portaria da CAPES n. 36, de 19 de março de 2020, de acordo com seu segundo artigo:*

*Art. 2º A suspensão de que trata esta Portaria não afasta a possibilidade de defesas de tese utilizando tecnologias de comunicação à distância, quando admissíveis pelo programa de pós-graduação stricto sensu, nos termos da regulamentação do Ministério da Educação.*"

TÍTULO SUGERIDO PELA BANCA

PREDITOR DE DESEMPENHO DE ESTUDANTES DO CURSO DE MATEMÁTICA INDUSTRIAL/UFG VIA REDES NEURAIS MLP E SOM



Documento assinado eletronicamente por **Jose Dos Reis Vieira De Moura Junior, Orientador**, em 03/02/2022, às 21:08, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do [Decreto nº 10.543, de 13 de novembro de 2020](#).

Documento assinado eletronicamente por **CAMILA PEREIRA DA SILVA, Discente**, em 03/02/2022, às 21:15, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



Documento assinado eletronicamente por **Donald Mark Santee, Professor do Magistério Superior**, em 03/02/2022, às 21:22, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



Documento assinado eletronicamente por **Marcelo Henrique Stoppa, Professor do Magistério Superior**, em 04/02/2022, às 09:25, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



A autenticidade deste documento pode ser conferida no site [https://sei.ufg.br/sei/controlador\\_externo.php?acao=documento\\_conferir&id\\_orgao\\_acesso\\_externo=0](https://sei.ufg.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0), informando o código verificador **2582006** e o código CRC **DFA3D7C0**.

**Referência:** Processo nº 23070.067906/2021-19

SEI nº 2582006

# RESUMO

Neste trabalho, utiliza-se técnicas de Mineração de Dados Educacionais baseadas na combinação de métodos de aprendizado de máquina supervisionado e não supervisionado pelos métodos de Redes Neurais Artificiais do tipo Multilayer Perceptron (MLP) e Mapas Auto-organizáveis de Kohonen (SOM), respectivamente, para o desenvolvimento de uma ferramenta capaz de classificar o desempenho acadêmico de estudantes a partir de notas em disciplinas da grade curricular do curso presencial de Bacharelado em Matemática Industrial. A ferramenta desenvolvida para classificação dos estudantes, com os dados utilizados neste trabalho, obteve com precisão a previsão do desempenho acadêmico, possibilitando que os gestores do curso possam utilizá-la como método auxiliador de tomada de decisões.

**Palavras-chaves:** Mineração de Dados Educacionais, Desempenho Acadêmico, Redes Neurais Artificiais, Modelagem, Ferramenta de auxílio.

# ABSTRACT

In this work, Educational Data Mining techniques are used based on the combination of supervised and unsupervised machine learning methods by Multilayer Perceptron-type Artificial Neural Networks. (MLP) and Auto-Maps organizers of Kohonen (SOM), respectively, for the development of a tool capable of classifying the academic performance of students based on grades in subjects in the on-site course of Bachelor of Industrial Mathematics. The tool developed for classifying students, with the data used in this work, accurately obtained the prediction of academic performance, enabling course managers to use it as an aiding method for decision-making.

**Keywords:** educational data mining, Academic achievement, Artificial neural networks , Modeling , aid tool.

---

# LISTA DE FIGURAS

---

Figura 2.1 – Áreas relacionadas à Mineração de Dados Educacionais . . . . .	20
Figura 2.2 – Neurônio biológico . . . . .	30
Figura 2.3 – Representação Simplificada do Neurônio Matemático . . . . .	32
Figura 2.4 – Gráficos funções de ativação a) e b) . . . . .	34
Figura 2.5 – Gráficos funções de ativação c) e d) . . . . .	34
Figura 2.6 – Rede alimentada adiante com uma única camada de neurônios . . . . .	36
Figura 2.7 – Rede alimentada adiante totalmente conectada com uma camada oculta e uma camada de saída . . . . .	37
Figura 2.8 – Rede recorrente sem laços de auto-realimentação e sem neurônios ocultos	37
Figura 2.9 – Rede recorrente com neurônios ocultos . . . . .	38
Figura 2.10 – Diagrama de blocos da aprendizagem com professor . . . . .	41
Figura 2.11 – Diagrama de blocos da aprendizagem não supervisionada . . . . .	42
Figura 2.12 – Redução de dimensionalidade usando SOM . . . . .	47
Figura 2.13 – Uma simples rede de Kohonen . . . . .	48
Figura 2.14 – Exemplo de SOM . . . . .	49
Figura 2.15 – Exemplo de SOM . . . . .	50
Figura 2.16 – Exemplo de SOM . . . . .	51
Figura 2.17 – Exemplo de SOM . . . . .	51
Figura 2.18 – Exemplo de SOM . . . . .	52
Figura 2.19 – Diferentes resultados para uma predição de duas classes . . . . .	54
Figura 2.20 – Matriz de confusão com resultados de uma predição com três classes . . .	55
Figura 3.1 – Dados com valores ausentes . . . . .	60
Figura 3.2 – Imputação nos dados usando a maior frequência dos valores . . . . .	60
Figura 4.1 – Topologia da Rede para Experimento . . . . .	68
Figura 4.2 – Matriz de confusão da MLP no experimento . . . . .	69
Figura 4.3 – Erro no treinamento e validação . . . . .	70
Figura 4.4 – Precisão no treinamento e validação . . . . .	70
Figura 4.5 – Configuração a . . . . .	72
Figura 4.6 – Configuração b . . . . .	72

Figura 4.7 – Configuração c . . . . .	72
Figura 4.8 – Configuração d . . . . .	72
Figura 4.9 – Resultados do modelo perante as Configurações do Conjunto de Dados . .	72
Figura 4.10 –Matriz de confusão da MLP no experimento ajustado . . . . .	72
Figura 4.11 –Mapa de distribuição de distância entre os nós da rede . . . . .	74
Figura 4.12 –Matriz de confusão da SOM no experimento . . . . .	75
Figura 4.13 –Configuração a - 75% precisão . . . . .	77
Figura 4.14 –Configuração b - 75% precisão . . . . .	77
Figura 4.15 –Configuração c - 75% precisão . . . . .	77
Figura 4.16 –Configuração d - 88% precisão . . . . .	77
Figura 4.17 –Resultados do modelo perante as Configurações do Conjunto de Dados . .	77
Figura 4.18 –Matriz de confusão da MLP no experimento ajustado . . . . .	79

---

# LISTA DE QUADROS

---

Quadro 3.1 – Colunas que possuem valores ausentes no Conjunto . . . . .	59
Quadro 3.2 – Resumo dos modelos parametrizados . . . . .	64
Quadro 4.1 – Medidas de Precisão, Recall e F-measure da MLP no experimento 02 . . .	69
Quadro 4.2 – Medidas de Precisão, Recall e F-measure da MLP no experimento 02 . . .	73
Quadro 4.3 – Medidas de Precisão, Recall e F-measure da MLP no experimento 02 . . .	76
Quadro 4.4 – Erro topográfico e quantizado na SOM . . . . .	78
Quadro 4.5 – Precisão do modelo diante evolução de iterações . . . . .	78

---

# SUMÁRIO

---

<b>1</b>	<b>INTRODUÇÃO</b>	<b>13</b>
1.1	Problemática	13
1.2	Objetivos	15
1.3	Estrutura da Dissertação	15
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b>	<b>17</b>
2.1	Mineração de Dados Educacionais	17
2.1.1	Definição	17
2.1.2	Objetivos da Mineração de Dados Educacionais	19
2.1.3	Áreas relacionadas à Mineração de Dados Educacionais	20
2.1.4	Predição de Desempenho de Estudantes	20
2.2	Trabalhos Relacionados	22
2.3	Pré Processamento de Dados	23
2.4	Falta de dados em modelos de Machine Learning (ML)	23
2.4.1	Imputação Estatística	25
2.5	Redes Neurais Artificiais	27
2.6	O Neurônio Biológico	30
2.7	O Neurônio Matemático	31
2.8	A arquitetura das Redes Neurais	35
2.8.1	Redes feedforward de Camada Única	35
2.8.2	Redes feedforward de Múltiplas Camadas	36
2.8.3	Redes Recorrentes	37
2.9	Aprendizado	38
2.9.1	Aprendizagem Supervisionada	40
2.9.2	Aprendizagem Não-Supervisionada	42
2.9.3	Redes Neurais Artificiais Multilayer Perceptron	42
2.10	Algoritmo de Aprendizagem Back Propagation	44
2.11	Mapas Auto-Organizáveis de Kohonen	46
2.11.1	Arquitetura de rede de SOM	47
2.11.2	O algoritmo de aprendizagem	48
2.11.3	Etapas do Algoritmo de aprendizagem	49
2.11.3.1	Passo 1: Inicializando os pesos	50

2.11.3.2	Calcular a melhor unidade de correspondência (BMU)	52
2.11.3.3	Ajuste dos pesos	53
2.12	Estratégias de Validação	53
2.12.1	Validação Cruzada	53
2.12.1.1	Matriz de Confusão	54
2.13	Medidas de Avaliação	55
2.13.1	Precisão	55
2.13.2	F-Measure	55
3	METODOLOGIA	57
3.1	Coleta de Dados Educacionais	57
3.2	Imputação de dados	58
3.3	Modelagem da Rede Neural	61
3.3.1	Parâmetros da MLP	61
3.3.2	Função de ativação para a MLP	62
3.3.3	Parâmetros da SOM	62
3.3.4	Resumo dos Modelos	64
3.4	Ferramentas Computacionais	64
4	RESULTADOS E DISCUSSÕES	67
4.1	Experimento 01 - Predição usando a rede neural MLP	67
4.2	Experimento 02 - Kohonen	73
5	CONCLUSÕES	81
	REFERÊNCIAS	83

## Capítulo 1

---

# INTRODUÇÃO

---

## 1.1 Problemática

Atualmente, todos os cursos de graduação no Brasil precisam ser avaliados conforme a Lei nº 10.861, de 14 de abril de 2004, pelo Sistema Nacional de Avaliação da Educação Superior (SINAES), órgão avaliador formado por três indicadores educacionais: a avaliação das instituições, dos cursos e do desempenho dos estudantes (RISTOFF; GIOLO, 2006).

Os Indicadores Educacionais concedem valores estatísticos à qualidade do ensino e também ao contexto econômico e social. São essenciais para o controle dos ambientes educacionais, levando em consideração o acesso, a permanência e a aprendizagem dos alunos. Deste modo, contribui diretamente no desenvolvimento de políticas públicas voltadas para a melhoria da qualidade da educação e dos serviços oferecidos à sociedade pela instituição acadêmica (RISTOFF; GIOLO, 2006).

Alguns indicadores de avaliação de ensino são: o Exame Nacional de Desempenho dos Estudantes (ENADE), a Certificação Profissional ANBIMA Série 10 (CPA- 10), o Conceito Preliminar de Curso (CPC) e são outorgados à cursos de graduação, influenciando o desenvolvimento das Instituições. O ENADE, um dos indicadores e instrumentos criados pelo governo, tem como objetivo avaliar a qualidade da educação superior no Brasil utilizando-se de Indicadores estabelecido pelo MEC (POLIDORI; MARINHO-ARAUJO; BARREYRO, 2006).

Performance ou desempenho é um conjunto de competências de comportamento e rendimento de um indivíduo. Para descrever o desempenho envolve o seu comprometimento e o rendimento é o resultado da sua avaliação, expresso na forma de notas ou resultados obtidos pelo sujeito em determinado trabalho (MUNHOZ, 2004).

O desempenho acadêmico é o critério mais importante para avaliar a qualidade dos estudantes universitários, possibilitando que a universidade monitore a qualidade do ensino e aprendizagem dos alunos (MUNHOZ, 2004).

O desempenho acadêmico de alunos do ensino superior, tem sido um elemento pesquisado como em (SANTOS, 2012b), (CORBUCCI, 2007), (CERQUEIRA *et al.*, 2000), tais estudos indicam a relevância de ampliar possibilidades eficazes e práticas competentes de ensino-aprendizagem em que a aprendizagem dos alunos é o intuito fundamental de uma instituição de ensino superior. (ARAÚJO *et al.*, 2013).

A qualificação do desempenho acadêmico necessita ser vista como uma ferramenta de auxílio na gestão por parte da Instituição de Ensino Superior (IES) na procura por qualidade, excelência e relevância pelos docentes ou pelos respectivos gestores e, por essa razão, a avaliação de desempenho dos discentes é fundamental para o prestígio da instituição de ensino, influenciando diretamente na sua imagem diante da sociedade e comunidades acadêmicas (ARAÚJO *et al.*, 2013).

O ENADE realiza a avaliação do desempenho dos formandos dos cursos de ensino superior público e privado do país. São avaliações obrigatórias compostas de questões objetivas e subjetivas, onde a frequência máxima é trienal para cada área do conhecimento, onde avalia-se o rendimento em relação aos conteúdos programáticos específicos, habilidades e competências adquiridas ao longo de toda a formação (GRIBOSKI, 2012).

Explorar sobre o desempenho discente e as variáveis que o influenciam é de grande importância para contribuir com o aprimoramento de políticas de ensino que resultem em melhor desempenho acadêmico e, conseqüentemente, melhor qualidade do ensino (CORBUCCI, 2007).

Nos últimos anos, técnicas de Mineração de Dados, *do inglês Data Mining - DM*, têm sido utilizadas como forma de predição de desempenho dos estudantes. A DM consiste na aplicação de algoritmos específicos para extrair padrões a partir de dados. É parte de um contexto mais amplo, conhecido como Descoberta de Conhecimento em Banco de Dados *do inglês Knowledge Discovery in Databases- KDD*.

KDD é o processo de descobrir conhecimento útil a partir de dados e, além da etapa de DM, envolve etapas adicionais como preparação, seleção, limpeza dos dados, incorporação de conhecimento prévio apropriado e interpretação adequada dos resultados da mineração de dados, a fim de garantir que resultados úteis sejam derivados dos dados (FAYYAD; PIATETSKY-SHAPIRO; SMYTH, 1996).

O conhecimento obtido a partir de dados pode ser útil para o aprimoramento dos sistemas de ensino, originando uma área de pesquisa denominada Mineração de Dados Educacionais *do inglês Educational Data Mining - EDM*. Esta área está focada com o desenvolvimento de métodos para explorar informações coletadas de ambientes educacionais, permitindo compreender os alunos de forma mais eficaz e adequada, proporcionando melhores benefícios educacionais aos mesmos (BAKER; ISOTANI; CARVALHO, 2011).

A etapa central da mineração de dados é a aplicação de técnicas de mineração aos

dados previamente processados. Alguns exemplos de técnicas de mineração de dados são: visualização, regressão, classificação, agrupamentos, regras de associação, entre outros (BAKER; YACEF *et al.*, 2009).

O aprendizado de máquina, *do inglês Machine Learning - ML* tem sido cada vez mais usado em áreas da mineração de dados educacionais para analisar e prever dados educacionais. É capaz de aprender automaticamente sem ser programado explicitamente. Também é capaz de detectar implicitamente relacionamentos não lineares entre variáveis dependentes e independentes.

Os dados educacionais foram coletados, pré-processados e processados em classificadores supervisionado e não supervisionado, a fim de desenvolver uma ferramenta baseada em machine learning com configurações adequadas para auxiliar na classificação precisa do desempenho dos alunos.

## 1.2 Objetivos

De maneira geral, esta dissertação tem como objetivo a construção de uma ferramenta de apoio no acompanhamento do desempenho dos estudantes do curso de Bacharelado em Matemática Industrial da Universidade Federal de Catalão, permitindo que os gestores educacionais tomem decisões, a partir das classificações do modelo para melhorar a qualidade do ensino, identificando alunos que precisam de uma atenção maior.

## 1.3 Estrutura da Dissertação

Os próximos capítulos da dissertação está organizada da seguinte forma:

- Capítulo 2: são introduzidos conceitos envolvendo Mineração de Dados Educacionais, Predição de Desempenho de Estudantes e Redes Neurais Artificiais, essenciais para um bom entendimento deste trabalho. Além disso, são expostos trabalhos que utilizam técnicas de mineração de dados educacionais para predição do desempenho acadêmico do estudante.
- Capítulo 3: neste capítulo são apresentados os métodos de coleta e pré-processamento dos dados que serão aplicados para prepará-los para o processo de classificação. Também apresenta-se as redes neurais artificiais que serão utilizadas para classificação dos estudantes quanto ao seu desempenho acadêmico, os parâmetros utilizados em cada rede e os métodos que avaliam a precisão dos modelos preditivos obtidos.
- Capítulo 4: neste capítulo é descrito os experimentos realizados e os resultados obtidos para avaliar a capacidade dos métodos propostos em identificar estudantes em diferentes níveis de desempenho.

- O Capítulo 5 faz uma conclusão deste trabalho e algumas considerações finais.

## Capítulo 2

---

# Fundamentação Teórica

---

Este capítulo apresenta conceitos essenciais para o entendimento deste trabalho. A primeira seção aborda o conceito de Mineração de Dados Educacionais, que é a aplicação de técnicas de mineração de dados em dados provenientes de ambientes de ensino. A seção seguinte introduz os conceitos de Machine Learning com ênfase nas Redes Neurais Artificiais. A terceira seção explana a Predição de Desempenho de Estudantes, que é o objetivo da utilização de Redes Neurais Artificiais neste trabalho.

## 2.1 Mineração de Dados Educacionais

### 2.1.1 Definição

A Mineração de Dados Educacionais, do inglês Educational Data Mining (EDM), é a aplicação de técnicas de Mineração de Dados em dados educacionais. É uma área de pesquisa que emergiu nos últimos anos e é utilizada por pesquisadores de diversas áreas, tais como ciências da computação, educação, estatística, psicologia, sistemas de tutoria inteligente, na análise de grandes conjuntos de dados com objetivo de resolver problemas de pesquisas educacionais (BAKER; YACEF *et al.*, 2009).

A preocupação da EDM está relacionada ao desenvolvimento de métodos para explorar informações coletadas de ambientes educacionais, permitindo a compreensão dos alunos de forma mais eficaz e adequada, proporcionando melhores benefícios educacionais aos mesmos (BAKER; ISOTANI; CARVALHO, 2011).

Em EDM, os dados provenientes de sistemas educacionais são convertidos em informações úteis que podem ter grandes impactos em pesquisas e práticas educacionais. O processo realizado não é muito diferente de outras aplicações de Mineração de Dados, tais como, negócios, genética, medicina, porque baseia-se nos mesmos passos do processo de mineração em geral. De acordo com (GARCÍA *et al.*, 2011), as etapas são:

- Etapa I - Pré-processamento: Os dados obtidos de sistemas educacionais têm que primeiro ser pré-processados e transformados em um formato apropriado para a mineração. Algumas das tarefas principais desta etapa são: limpeza, seleção de atributos, imputação de dados ausentes e integração de dados.
- Etapa II - Mineração de Dados: É a etapa central, e nela, técnicas de mineração são aplicados aos dados previamente pré-processados. Alguns exemplos de técnicas de mineração de dados são: visualização, regressão, classificação, agrupamentos, regras de associação, entre outros.
- Etapa III - Pós-processamento: Nesta etapa, os resultados ou modelos obtidos são interpretados e usados em tomadas de decisão sobre o ambiente educacional.

As técnicas utilizadas em EDM passam por adaptações devido as particularidades existentes em ambientes educacionais e seus dados. A seguir, é apresentado uma taxonomia, proposta por (BAKER; ISOTANI; CARVALHO, 2011), que informa a categorização das sub-áreas de pesquisa da EDM.

- Predição
  - Classificação
  - Regressão
  - Estimativa de Densidade
- Agrupamento
- Mineração de Relações
  - Regras de Associação
  - Correlações
  - Mineração de causas e padrões

Na predição são desenvolvidos modelos que farão inferências sobre dados a partir de informações de outros dados previamente analisados. No agrupamento os dados são dispostos em grupos que se formam de acordo com as similaridades de cada item. Na mineração de relações são descobertas relações entre variáveis contidas nos dados. A mineração de regras de associação procura gerar ou identificar regras do tipo  $X \rightarrow Y$ , ou seja, uma regra associa a ocorrência de um valor Y dado que ocorreu o valor X.

Na mineração de correlações obtêm-se correlações lineares, que podem ser positivas ou negativas, entre duas variáveis. Na mineração de causas, a partir da análise dos padrões de covariância, verifica-se se um evento causa outro evento.

### 2.1.2 Objetivos da Mineração de Dados Educacionais

(ROMERO; VENTURA, 2013) sugeriram a classificação dos objetivos da EDM de acordo com o usuário final (estudante, educador, administrador de curso e pesquisador) e o problema que será resolvido:

- **Estudantes:** Apoiar os estudantes no processo de ensino-aprendizagem, fornecer feedback ou recomendações, responder as necessidades do estudante, melhorar o desempenho no aprendizado.
- **Educadores:** Entender o processo de aprendizagem dos alunos e como os métodos aplicados são recebidos pelos alunos, além de melhorar o desempenho no ensino.
- **Administradores de Curso:** Avaliar a melhor maneira de organizar os recursos (humanos e didáticos) disponíveis para a realização dos cursos.
- **Pesquisadores:** Desenvolver, comparar e avaliar técnicas de mineração de dados com o objetivo de recomendar a melhor técnica que traga informações úteis a cada problema ou tarefa específica dos ambientes educacionais. Estas informações apoiaram os trabalhos dos alunos, educadores e administradores de curso.

(ROMERO *et al.*, 2010) ainda destacam algumas aplicações da mineração de dados educacionais:

- **Comunicação entre as partes envolvidas:** O objetivo é ajudar administradores de cursos e educadores na análise das atividades realizadas pelos estudantes e nas informações de uso (acesso a fóruns, materiais didáticos, questionários) em cursos. As técnicas mais utilizadas para este propósito são a análise exploratória de dados através de análise estatísticas e técnicas de visualização de informações.
- **Manutenção e melhorias dos cursos:** Objetiva ajudar administradores de curso e educadores no que se pode melhorar em cursos (conteúdos, atividades, links, e etc.) utilizando, por exemplo, informações de uso do sistema educacional, ou seja, como e quais recursos foram utilizados pelos estudantes. As técnicas de mineração de dados mais comuns para este objetivo são as regras de associação, sequencias frequentes, classificação e agrupamentos.
- **Recomendações de conteúdos:** Recomenda a estudantes conteúdos que sejam mais apropriados para seu aprendizado. As técnicas mais comuns nesta aplicação são as regras de associação, sequencias frequentes, classificação e agrupamentos.
- **Predição de notas de estudantes e desempenho em cursos:** Obtém uma previsão de notas finais em cursos ou chances de um estudante abandonar o curso, baseados nos

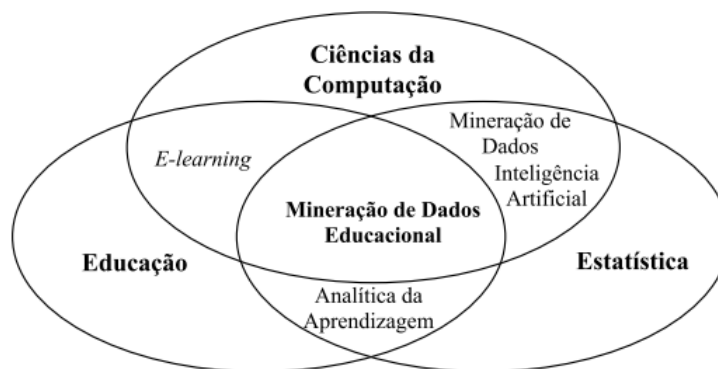
dados gerados por outros estudantes durante a realização de cursos. As técnicas utilizadas são classificação, agrupamentos e regras de associação.

- **Modelagem do perfil do usuário:** No domínio educacional, a modelagem do perfil do usuário pode detectar o estado e características dos estudantes, tais como satisfação, motivação, progresso da aprendizagem, preferências e outros fatores que impactam negativamente o desempenho destes estudantes (motivos que levam a erros em questionários, por exemplo). Além da classificação, agrupamentos, regras de associação, também são utilizadas técnicas de análises estatísticas e redes Bayesianas para este propósito.

### 2.1.3 Áreas relacionadas à Mineração de Dados Educacionais

EDM pode ser vista como uma junção de três áreas, conforme a figura 2.1: Ciência da computação, estatística e educação. É uma área interdisciplinar e utiliza métodos e técnicas da estatística, aprendizado de máquina, mineração de dados, recuperação de informação, sistemas de recomendação, psicologia cognitiva, entre outros.

Figura 2.1 – Áreas relacionadas à Mineração de Dados Educacionais



Fonte: (ROMERO; VENTURA, 2013)

### 2.1.4 Predição de Desempenho de Estudantes

Várias técnicas de mineração de dados podem ser utilizadas na criação de modelos que façam inferências sobre o desempenho de estudantes. Um modelo pode ser entendido como uma abstração que faz previsões sobre o futuro em informações do passado ou presente. Um modelo inclui um algoritmo de aprendizado de máquina que aprende sobre características de objetos em um conjunto de dados de treinamento com o objetivo de realizar previsões sobre outros objetos.

Os modelos preditivos são basicamente divididos em duas sub-áreas: regressão e classificação de padrões. Na regressão analisa-se a relação entre duas variáveis com o objetivo de encontrar tendências entre estas variáveis. Na classificação de padrões, dados são atribuídos

às classes de acordo com as informações dos dados previamente informados ao algoritmo de aprendizagem de máquina.

A partir da entrada de dados, o modelo pode inferir sobre estes dados e produzir uma predição como saída. A predição obtida pelo modelo, entre outras coisas, depende da qualidade e da quantidade de dados disponíveis. Um modelo é construído em função das informações disponíveis na base de dados e da seleção adequada de técnicas de mineração de dados.

O modelo gerado precisa ser validado para verificar se ele atende as condições impostas para solucionar o problema. Diferentes modelos podem solucionar o problema, escolhe-se aquele que apresenta melhores resultados na etapa de validação. No processo de obtenção do modelo, duas fases são executadas: na primeira, um conjunto de dados de treinamento será utilizado pelo algoritmo para construir o modelo descritivo dos dados.

Na segunda fase, novos dados, chamados dados de teste são utilizados com objetivo de testar o modelo gerado na primeira fase. A análise dos resultados é feita pelo responsável pela execução dos processos, afim de verificar se o modelo atende ao propósito. Caso um modelo não atenda as necessidades definidas, podem ser realizadas as seguintes ações:

- Executar o algoritmo novamente utilizando novos dados ou modificando os parâmetros;
- Utilizar outro algoritmo de aprendizado de máquina.

Neste trabalho, o objetivo da predição é estimar valores desconhecidos de uma variável que descreve o estudante. Na educação, este valor pode informar sobre desempenho acadêmico, conhecimento sobre determinado assunto, notas em avaliações, rendimentos em disciplinas, entre outras. Este valor pode ser numérico ou contínuo (regressão), ou, categórico ou discretos (classificação).

A regressão procura a relação entre uma variável dependente e uma ou mais variáveis independentes. A classificação é um procedimento em que itens são individualmente colocados em grupos com base em informações quantitativas sobre uma ou mais características comuns ao itens e com base em um conjunto de treinamento com itens previamente rotulados, ou seja, os itens contêm informações sobre quais grupos pertencem.

A predição de desempenho de estudantes é uma das técnicas mais abordadas em aplicações de mineração de dados na educação, e diferentes técnicas e modelos foram aplicados (redes neurais, classificadores Bayesianos, regressão e análise de correlação) (ROMERO *et al.*, 2010).

Este trabalho utiliza Redes Neurais Artificiais (RNA) para obtenção de modelos que façam inferências sobre o desempenhos de estudantes. Duas arquiteturas de redes neurais

foram utilizadas para predição do desempenho dos alunos de universidades de acordo com as notas obtidas nas disciplinas da grade curricular do curso (predição de desempenho que leva a aprovação ou a reprovação).

## 2.2 Trabalhos Relacionados

Diversos autores abordaram a Mineração de Dados Educacionais, do inglês Educational Data Mining (EDM), visando resolver problemas ou propor melhorias nos processos educacionais. (ROMERO; VENTURA, 2013), realizaram trabalhos que fornecem feedbacks para o apoio a educadores nas tomadas de decisões. (BARCELLOS *et al.*, 2007) propôs uma aplicação que recomenda artigos a estudantes com base nas informações de perfil destes estudantes. Outros autores trabalharam para identificar comportamentos indesejáveis nos estudantes, como falta de motivação (HERSHKOVITZ; NACHMIAS, 2008), predisposição à desistência (DEKKER; PECHENIZKIY; VLEESHOUWERS, 2009), reprovações (BRESFELEAN *et al.*, 2008) ou tentativas de burlar as avaliações (BAKER; YACEF *et al.*, 2009).

O trabalho de (AYERS; NUGENT; DEAN, 2009) utiliza técnicas de agrupamento para proporcionar a estudantes um ambiente de ensino e conteúdos personalizados de acordo com o seu perfil e habilidades. (RIGO; CAZELLA; CAMBRUZZI, 2012) utiliza técnicas de Data Mining (DM) e learning analytics para estudar fatores que levam a evasão escolar.

Este trabalho objetiva a realização de predições de desempenho de estudantes. A técnica de DM utilizada é conhecida como classificação. O objetivo da classificação é fornecer modelos que possam inferir um aspecto particular dos dados através de alguma combinação de outros aspectos destes dados (WITTEN IAN H E FRANK, 2002). A previsão do desempenho, ou situação, acadêmica de alunos é um tema também muito estudado entre pesquisadores.

Em (MANHÃES *et al.*, 2012), os autores utilizaram técnicas de mineração de dados para prever o risco de evasão de estudantes em cursos presenciais da Escola Politécnica da Universidade do Rio de Janeiro. Dez modelos diferentes foram testados, com acurácia média variando entre 75% e 80% , com os algoritmos de classificação Multilayer Perceptron e RandomForest apresentando os melhores desempenhos. O estudo mostrou que a partir das primeiras notas semestrais dos calouros é possível identificar com precisão de 80% a situação final do aluno.

Em (SANTOS, 2012a) utiliza-se dados de estudantes do curso de Engenharia da Computação da Universidade Federal do Pampa visando de contribuir para diminuição de problemas de estudantes que reprovam no início do curso. Com dados de ambientes virtuais de aprendizagem e técnicas de mineração de dados baseadas em agrupamentos com K-means e classificação com árvores de decisão REPTree e J48 os autores criaram modelos capazes de identificar tendências a reprovação com acurácia em torno de 69%. Os estudos mostraram

que a medida que o aluno se dedica as atividades presenciais e semipresenciais, aumenta o seu desempenho nas avaliações somativas, e com isso, suas chances de aprovação.

Um trabalho de doutorado descrito em (MANHÃES, 2015), propõe-se uma arquitetura que engloba todo o processo de descoberta de conhecimentos em dados educacionais. Foram utilizados dados coletados durante 16 anos sobre graduandos da Universidade Federal do Rio de Janeiro (UFRJ) para testar a arquitetura e os modelos propostos.

Na abordagem proposta, os dados utilizados foram obtidos das bases de dados do sistema de gestão acadêmica da UFRJ. São utilizados diversos algoritmos de mineração de dados, com o objetivo de fazer previsões de desempenhos destes estudantes, entre eles J48, MultiLayer Perceptron, Random Forest e Support Vector Machine (SVM).

Os resultados dos experimentos mostram que a arquitetura proposta é capaz de prever o desempenho acadêmico dos graduandos a cada semestre letivo com precisão em torno de 80%. Além da previsão, o estudo também identificou as principais variáveis que distinguem os estudantes que obtêm sucesso ou não na conclusão do curso de graduação.

## 2.3 Pré Processamento de Dados

O pré-processamento dos dados é uma etapa essencial da Mineração de Dados, do inglês Data Mining (DM), pois os dados provenientes de bases de dados do mundo real geralmente têm inconsistências, erros, valores ausentes, ou simplesmente não estão adequados para a utilização nos processos de DM (GARCÍA *et al.*, 2011). No caso deste trabalho, a ausência de dados influencia no desempenho do classificador da Rede Neural Artificial.

A etapa de pré-processamento consiste na preparação dos dados para atenderem os requisitos de entrada de cada algoritmo de DM que serão utilizados nas próximas etapas. São realizadas integração de dados, limpeza, normalização e transformação dos dados, além da redução dos dados. Tais técnicas visam melhorar a acurácia e eficiência de algoritmos de classificação.

Nenhuma técnica supracitada é mutualmente exclusiva, ou seja, elas podem ser abordadas em conjunto ou usadas apenas quando necessárias. Este trabalho utiliza técnicas de imputação de dados ausentes com o objetivo de obter melhores resultados nos processos de classificação.

## 2.4 Falta de dados em modelos de Machine Learning (ML)

A aprendizagem de máquina ou machine learning (ML) abrange o estudo de métodos computacionais que adquirem experiência e aprimora seu desempenho através do aprendizado (MITCHELL *et al.*, 1997). Métodos de ML têm sido amplamente empregados em

diversas aplicações, como por exemplo, controle de acesso (reconhecimento facial), desenvolvimento de jogos, diagnóstico e análise de registros médicos e etc.

Em qualquer área referente à busca e exploração de conhecimento, existe uma grande preocupação envolvendo a qualidade dos dados, pois para um resultado atingir as finalidades ambicionadas ele depende dos dados utilizados (CIOS *et al.*, 2007). Se tratando em qualidade dos dados, os estudos abrangem a classificação e o progresso da aplicação de métodos de tratamento em dados de má qualidade (PIPINO LEO L E LEE, 2002).

Dados ausentes em um conjunto de dados podem gerar problemas para muitos algoritmos de aprendizado de máquina. Os valores ausentes são originados da confusão dos dados do mundo real havendo infinitas razões pelas quais eles acontecem, desde erros humanos durante a entrada de dados, leituras incorretas do sensor a bugs de software no pipeline de processamento de dados (LITTLE, 1988).

Para completar valores ausentes (missing data) são utilizados métodos para fazer a limpeza dos dados (data cleaning), amenizar ruídos, retificar inconsistências e identificar ou remover dados que se diferem completamente de todos os outros (outliers) (HAN JIAWEI E KAMBER, 2000).

Limpeza de dados, uma importante etapa dos métodos destinados a esta finalidade, equivale ao tratamento dos dados com valores ausentes (CIOS *et al.*, 2007). Encontrados frequentemente na prática e podem suceder por diversos motivos, como por exemplo: omissão de dados pela própria fonte de informação e mau-funcionamento do mecanismo de coleta de dados.

Ainda que alguns algoritmos de ML sejam transigentes aos valores ausentes (BREIMAN LEO E FRIEDMAN, 1984), (QUINLAN, 1993), a maior parte deles não apresentam esta habilidade (PYLE, 1999), (JOHANSSON PETER E H "A KINEN, 2006). Desta maneira, vários métodos têm sido apresentados para tratar os valores ausentes (SCHAFER, 1997).

Um recurso simples para trabalhar com dados ausentes compreende em desconsiderar atributos contendo valores ausentes. Todavia, a exclusão de dados pode ser relevante, sendo capaz de influenciar desfavoravelmente nas análises dos dados. No trabalho feito por (KIM KI-YEOL E KIM, 2004) foi analisado que após excluir dados relevantes que continham valores ausentes de um conjunto contendo no total 6361 genes (instâncias) e 156 medidas (atributos), para a execução de seus estudos, verificou-se uma redução de 88 % dos dados.

Uma técnica para tratar os dados envolve a substituição dos atributos ausentes, também chamado de imputação, que tem como objetivo preencher os valores ausentes por valores mais viáveis. Essa técnica consiste na estimação de valores por meio das informações disponíveis no conjunto de dados.

A imputação é adaptável, de maneira que o método escolhido para alteração de valores ausentes pode ser utilizado independente do algoritmo de aprendizagem de máquina

utilizado, possibilitando a escolha do método mais apropriado.

É uma recomendação constatar e substituir os dados com valores ausentes para cada coluna em seus dados de entrada antes de modelar sua tarefa de previsão. A primeira etapa a ser realizada é localizar e quantificar os dados que estão ausentes no conjunto de dados. É importante para que esta etapa funcione corretamente, a inspeção dos dados devem ser feitas manualmente para tentar determinar como estão sendo intitulados. As variações possíveis são: 'NaN', 'NA', 'Nenhum', ", '?' e outros.

O cálculo do valor estatístico para cada coluna é uma forma de imputação de dados, onde uma estatística como uma média é determinada no conjunto de dados de treinamento para substituir todos os valores ausentes dessa coluna, resultando geralmente em bom desempenho.

### 2.4.1 Imputação Estatística

São dados em que um ou alguns valores não estão compreensíveis ou claros. Esses dados podem ser explícitos de várias maneiras, como: estar marcados com um caractere ou valor especial, como NAN, uma string vazia, uma string explícita NULL ou indefinido, o número 0, etc (MCCALLUM, 2012).

Os dados ou valores podem estar em falta no banco de dados por diversas razões (aparelho de medição defeituoso ou mudança na coleta de dados, entre outros) normalmente por questões específicas do problema (WITTEN, 2016).

É comum que os bancos de dados utilizados em estudos nas mais variadas áreas de atividade humana apresentem-se com falhas, denominados de dados “faltantes”, “perdidos” ou “ausentes”. Essa incompletude pode ter vários motivos, pois os procedimentos na constituição dos bancos são dependentes das características das áreas científicas nas quais os dados se aplicam (ciências sociais, ciências da saúde, ciências ambientais, etc.).

Desprezar os valores faltantes na análise estatística é um método que pode servir apenas quando é pequeno o número de dados perdidos e quando ocorrem completamente ao acaso (LITTLE, 1988). Todavia, as indefinições associadas à imputação precisam ser estudadas corretamente pois os dados imputados não são realistas.

Três tipos de indefinições estão associados ao método de imputação:

- aquelas ocasionadas pela modelagem da distribuição conjunta das variáveis respostas e as indicadoras de dados faltantes;
- aquelas relativas à amostragem (seleção aleatória de valores a serem imputados) a partir de um modelo de imputação assumindo que os dados observados e os valores dos parâmetros do modelo são conhecidos

- as indefinições sobre os valores dos parâmetros do modelo (ZHANG, 2003).

Como possibilidade à imputação única e com o propósito de retificar suas desvantagens, Rubin introduziu na década de 80 a imputação múltipla (IM). Desde Então, a imputação de dados viralizou e sua aplicação é um procedimento de trabalho de banco de dados que ganhou uma enorme importância (RUBIN, 1987).

O conceito da imputação múltipla é imputar o valor ausente  $x$  vezes, para produzir bancos de dados completos. Os  $x$  bancos obtidos são analisados separadamente e estatisticamente e por fim os  $x$  resultados obtidos são (combinados de maneira simples) para a observação final (ZHOU; ECKERT; TIERNEY, 2001), (TANG *et al.*, 2005), (SCHAFER, 1997).

Como as técnicas de análises estatísticas computacionais são desenvolvidos, em sua maior parte, para dados completos, mesmo uma pequena parcela de dados faltantes podem causar dificuldades na estimação (viés, ineficiência), por isso a necessidade de ser considerado nas análises o problema de dados ausentes (JR, 2015).

Técnicas conhecidas como métodos de imputação única ou imputação pela média ou pela mediana, também têm sido bastante empregado pela sua facilidade de implementação. No entanto, existem prejuízos na aplicação desses métodos, tais como desvalorizar ou desprezar a variabilidade da variável imputada que gerará intervalos de confiança mais estreitos do se espera e a impossibilidade de levar em consideração a variabilidade que possa existir entre diferentes imputações (ZHOU; ECKERT; TIERNEY, 2001), (HEIJDEN *et al.*, 2006), (SCHAFER, 1997).

A grande maioria dos algoritmos de ML requerem valores de entrada numéricos e um valor para cada linha e coluna em um conjunto de dados. Desta maneira, os dados faltantes podem causar dificuldades para algoritmos de aprendizado de máquina.

Uma abordagem simples e popular para imputação de dados envolve o uso de métodos estatísticos para estimar um valor para uma coluna a partir dos valores que estão presentes e, em seguida, substituir todos os valores ausentes na coluna pela estatística calculada...

As estatísticas comuns calculadas incluem:

- O valor médio da coluna.
- O valor mediano da coluna.
- O valor do modo da coluna.
- Um valor constante.

O método mais promissor de imputação é através de modelos de regressão. Os dados faltantes são trocados por valores preditos através de um modelo de regressão ajustado com

os dados estudados. Essa técnica pode ser combinada com algum pareamento, no qual não é imediatamente imputado o valor predito pelo modelo, mas selecionado dentre os analisados aquele mais próximo do predito. Isto limita o intervalo de dados que podem ser imputados, impedindo discrepâncias entre estes dados imputados e aqueles valores realmente observados nos dados completos.

A imputação é um procedimento de muito importante e indispensável pois a exigência de se trabalhar com banco de dados completos é essencial para a análise estatística adequada. A escolha do método de imputação depende da classificação do banco de dados faltantes, ou seja, aquele que contém tanto os valores presentes quanto os valores faltantes.

## 2.5 Redes Neurais Artificiais

Uma das áreas de pesquisa mais fascinante presentemente é a simulação de capacidades cognitivas de um ser humano. Projetam-se máquinas capazes de exibir um comportamento inteligente, como se fossem reações humanas. (RAUBER, 2005)

A inteligência do ser humano é a mais avançada dentro do universo das criaturas e o local dessa inteligência dentro do corpo humano é o cérebro. As entidades básicas são os neurônios, interconectados em redes o que permite a troca de informação entre eles, criando a inteligência biológica. Uma ambição óbvia que surge desses fatos é a tentativa de copiar a estrutura e o funcionamento do cérebro em um ambiente técnico. (RAUBER, 2005)

A Rede Neural Artificial (RNA) é um mecanismo de modelagem poderosa e profunda utilizada em funções não lineares que frequentemente descreve os sistemas do mundo real. Uma RNA consegue captar as características mais complexas dos dados, o que nem sempre é possível empregando as técnicas tradicionais de estatística. (BRAGA; CARVALHO; LUDERMIR, 2000)

São técnicas computacionais baseadas na estrutura neural de organismos inteligentes e que conseguem habilidades por meio de experiências. Pode ser definida como uma rede grande paralela de componentes interconectados em suas organizações que estão arquitetadas para iterar com objetos do mundo real da mesma maneira que um sistema nervoso biológico o faz (AKKO, 2012).

A RNA busca representar as funções das redes biológicas, implementando em seu comportamento uma semelhança com o sistema biológico básico e sua dinâmica de aprendizado, assim entender melhor como funciona o sistema nervoso (VALENÇA, 2005).

Uma rede neural artificial (RNA) tem duas particularidades: a arquitetura e o algoritmo de aprendizagem. Essa divisão surge naturalmente pelo modo de como a rede é treinada que seria por exemplos de treino. (RAUBER, 2005) São distribuídos em camadas associadas, passando por um processo de treinamento, obtendo a sistemática necessária para

executar devidamente o processo desejado dos dados fornecidos. (ZANETTI *et al.*, 2008)

Aptos na resolução de problemas de grande complexidade, são modelos utilizados em circunstâncias onde não é possível definir claramente uma lista de regras, ou quando não possuem um padrão de constância. Portanto a RNA especifica padrões e previsões. (NUNES, 2018)

Constituídos de unidades de processamento para determinar funções matemáticas onde podem ser determinadas em conexões (sinapses) que são associadas a pesos (valores atribuídos) e que após o processamento de aprendizagem, armazenam regularmente o conhecimento obtido pela própria rede. (NUNES, 2018)

As RNAs possuem basicamente 3 camadas. A camada de entrada reúne dados de informações numéricas com conjuntos de recursos e valores de ativação. Os valores de entrada são propagados através dos neurônios interconectados para a camada oculta.

Nas camadas Intermediárias ou ocultas onde é feita a maior parte do processamento, os neurônios de entrada são somados para calcular a soma ponderada dos neurônios de entrada já a camada de saída os neurônios somados são combinados para produzir resultados na camada de saída usando uma função de ativação (ou transferência), ou seja, onde o resultado final é concluído e apresentado. (VANDAMME JEAN-PHILIPPE E MESKENS, 2007), (RASHID TARIK A E AHMAD, 2016)

O funcionamento de RNA é baseado em uma estrutura biológica: o cérebro humano. É composta por um agrupamento de neurônios artificiais que se assemelham à geometria de conexão dos neurônios no cérebro humano, com o propósito de executar uma tarefa com desempenho aprimorado por meio de aprendizagem, treinamento e melhoria contínua. (KOSE UTKU E ARSLAN, 2017)

Sua vantagem é o de otimização e não requerer informações aprofundas sobre o problema a ser modelado. Uma rede neural é capaz de extrair regras básicas a partir de dados reais ou como são denominados banco de dados, onde é necessário um conjunto de regras rígidas pré-fixadas e algoritmos. (NUNES, 2018)

Outra grande vantagem provém de algumas propriedades e capacidades úteis, como:

- **Aprendizagem:** é a habilidade da RNA aprender automaticamente, o mapeamento desejado, entre as entradas e saída, através de um processo iterativo de ajustes aplicados aos seus pesos sinápticos;
- **Generalização:** capacidade da RNA apresentar uma saída adequada para uma entrada não presente no processo de aprendizagem;
- **Não-linearidade:** uma RNA é não-linear se está for constituída de neurônios artificiais também não-lineares;

- **Adaptabilidade:** capacidade que as RNAs possuem de adaptar seus pesos perante a modificações no meio ambiente, ou seja, uma RNA treinada para operar em um ambiente específico pode facilmente ser re-treinada para absorver pequenas alterações no ambiente;
- **Tolerância e falhas:** o conhecimento é distribuído pela RNA, desta forma, uma parte das conexões pode estar inoperante, sem mudanças significativas no desempenho de toda a RNA;
- **Resposta a evidências:** uma RNA pode fornecer, em sua saída, não somente a informação relativa a qual conjunto a entrada pertence, mas também uma informação sobre a confiança do resultado.

Uma RNA realiza seu aprendizado baseado das variáveis de entrada. Apesar de que uma rede neural tenha condições de modelar problemas difíceis e complexos, é preciso que existam dados e observações confiáveis e suficientemente representativas para que o conhecimento possa ser extraído de forma que o aprendizado da RNA aconteça com sucesso e segurança. (NUNES, 2018)

RNA tem a propensão natural para armazenar conhecimento experimental e torná-lo disponível para o uso (HAYKIN, 2001). Sua modelagem é inspirada no funcionamento nos neurônios biológicos e assemelha-se ao cérebro em dois aspectos:

- o conhecimento é adquirido pela rede a partir de seu ambiente através de um processo de aprendizagem;
- forças de conexão entre neurônios, conhecidas como pesos sinápticos, são utilizadas para armazenar o conhecimento adquirido.

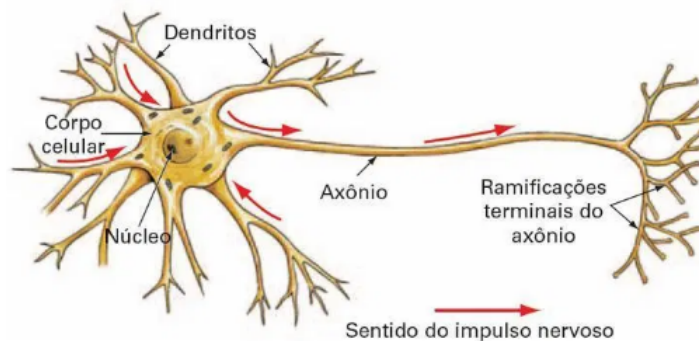
Constantemente as funções de treinamento são utilizadas para atualizar os pesos de conexão no processo de alimentar os valores de entrada e concluir com os valores de saída em RNA é chamado de Epoch (RASHID TARIK A E AHMAD, 2016). Onde as entradas dos neurônios artificiais são multiplicadas por pesos, e o resultado dessa soma é enviado para a camada de saída por meio de uma função de ativação (KOSE UTKU E ARSLAN, 2017).

As funções de ativação freqüentemente usadas incluem funções tangentes lineares, sigmóides e hiperbólicas. O treinamento é finalizado quando o valor máximo de época ou as verificações de validação são atingidas. Os resultados treinados são testados para explorar a performance da RNA. (LAU ET E SUN, 2019)

## 2.6 O Neurônio Biológico

A unidade básica do processamento é de uma RNA é o neurônio cuja figura representativa pode ser vista na Figura 2.2. Sendo uma célula especializada na transmissão de informações, pois nelas estão introduzidas propriedades de excitabilidade e condução de mensagens nervosas. Nesta podem ser vistas as partes constituintes do neurônio.

Figura 2.2 – Neurônio biológico



Fonte:([HAYKIN, 2001](#))

O neurônio é constituído por 3 partes principais: a soma ou corpo celular, do qual advêm algumas ramificações chamadas de dendritos, e por uma outra ramificação, porém mais extensa, chamada de axônio. Nas extremidades dos axônios estão os nervos terminais, pelos quais é realizada a transmissão das informações para outros neurônios. Esta transmissão é conhecida como sinapse.

Os dendritos são os receptores, as entradas do neurônio. O axônio é o que transporta o sinal de saída do neurônio. A sinapse que converte um sinal elétrico pré-sináptico em um sinal químico e então de volta em um sinal elétrico pós-sináptico. O corpo celular que é responsável pelo processamento dos sinais de entrada do neurônio. ([HAYKIN, 2001](#))

Quando os valores das entradas atingem um determinado limiar, o neurônio dispara liberando um impulso elétrico que flui do corpo celular para o axônio, que pode estar conectado à entrada de outro neurônio ([HAYKIN, 2001](#)).

O corpo celular e os dendritos constituem a parte de entrada e o axônio a superfície de saída do fluxo de informação. Os resultados propagados pelos neurônios são impulsos elétricos. O impulso elétrico é a mensagem que os neurônios transmitem uns aos outros, ou seja, é a disseminação de um estímulo ao longo dos neurônios que pode ser qualquer sinal captado pelos receptores nervosos. A passagem desses sinais não é elétrica, mas química (através da substância serotonina) ([HAYKIN, 2001](#)).

Influenciados pelo neurônio biológico, os pesquisadores apresentaram um modelo de neurônio matemático cuja a ideia era reproduzir as redes neurais que formam a inteli-

gência humana em uma Inteligência Artificial. E assim foi desenvolvido o neurônio matemático.

## 2.7 O Neurônio Matemático

Muitos pesquisadores empenharam-se em simular o funcionamento do cérebro, principalmente o processo de aprendizagem por experiência, com o intuito de gerar sistemas inteligentes capazes de realizar tarefas como classificação, reconhecimento de padrões, processamento de imagens, entre outras atividades.

Como consequência destes estudos sobreveio o modelo do neurônio artificial e posteriormente um sistema com vários neurônios interconectados, a chamada Rede Neural ([MCCULLOCH WARREN S E PITTS, 1943](#)).

Warren McCulloch e Walter Pitts desenvolveram um modelo computacional para redes neurais baseadas em matemática e algoritmos denominados lógica de limiar (threshold logic). A partir deste modelo criado o estudo da pesquisa sobre rede neural foi dividida em duas abordagens: uma ligada em processos biológicos no cérebro, enquanto a outra aplicada à inteligência artificial ([MCCULLOCH WARREN S E PITTS, 1943](#)).

Donald Hebb relata em sua obra o fato de que os neurônios são revigorados cada vez que são utilizados, um conceito essencial para entender a maneira como os humanos conseguem aprender. Se dois nervos dispararem ao mesmo tempo ele relata que a conexão entre eles é melhorada ([HEBB, 2005](#)).

O Projeto de Pesquisa de Verão de Dartmouth em 1955 favoreceu um grande avanço à Inteligência Artificial como às Redes Neurais. Um dos resultados deste projeto foi incentivar os estudos em IA na parte de processamento neural ([MCCARTHY JOHN E MINSKY, 2006](#)).

Intrigado com o funcionamento do olho de uma mosca Frank Rosenblatt, um neurobiologista inciou uma pesquisa que resultou grande parte do processamento feito por uma mosca ao decidir fugir, é feito utilizando seus olhos. O resultado desse estudo originou o Perceptron que foi construído em hardware e é a mais antiga rede neural ainda em uso hoje ([ROSENBLATT, 1960](#)).

O modelo de Warren McCulloch e Walter Pitts em 1943, foi bem aceito o qual realiza de maneira simples as partes principais e o funcionamento de um neurônio biológico. Mais especificamente, um neurônio matemático de uma rede neural artificial é um componente que calcula a soma ponderada de vários inputs (entradas), aplica uma função e se obtém o resultado ([MCCULLOCH WARREN S E PITTS, 1943](#)).

Os sinais de entrada ou os dados que alimentam seu modelo de rede neural artificial, representado pela letra  $x$ . Em meio aos estímulos recebidos, alguns excitarão mais do que outros o neurônio receptor e essa medida de quão excitatório é o impulso é representada no

modelo por meio dos pesos sinápticos (MCCULLOCH WARREN S E PITTS, 1943).

Quanto maior o valor do peso, mais excitatório é o estímulo. Os pesos sinápticos são representados por  $w_{kn}$ , onde  $k$  representa o índice do neurônio em questão e  $n$  se refere ao terminal de entrada da sinapse a qual o peso sináptico se refere (MCCULLOCH WARREN S E PITTS, 1943).

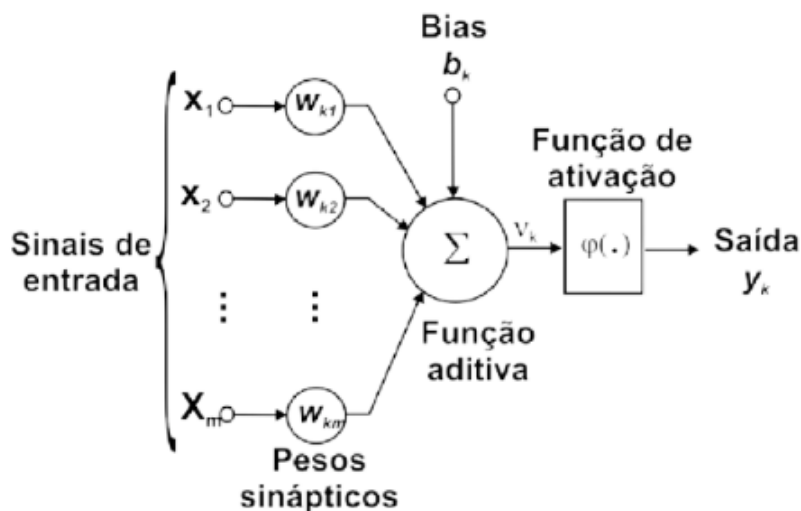
A soma (corpo da célula) é retratada por dois parâmetros, o primeiro é um somatório dos estímulos (sinais de entrada) multiplicado pelo seu fator excitatório (pesos sinápticos), e adiante uma função de ativação, que definirá com base nas entradas e pesos sinápticos, qual será a saída do neurônio e o bias  $b$  aumenta ou diminui a entrada líquida da função de ativação (HAYKIN, 2001).

O bias é incluída ao somatório da função de ativação, com o objetivo de aumentar o grau de liberdade desta função e, conseqüentemente, a capacidade de aproximação da rede (HAYKIN, 2001).

O valor do bias é estabelecido assim como os pesos sinápticos. Garante que um neurônio apresente saída não nula ainda que todas as suas entradas sejam nulas (HAYKIN, 2001).

O axônio é representado pela saída  $y_k$  obtida pela aplicação da função de ativação. Assim como no modelo biológico, o estímulo pode ser excitatório ou inibitório, representado pelo peso sináptico positivo ou negativo respectivamente (HAYKIN, 2001).

Figura 2.3 – Representação Simplificada do Neurônio Matemático



Fonte: Adaptado de (HAYKIN, 2001)

O modelo possui uma natureza binária. Tanto os sinais de entrada quanto a saída, são valores binários (MCCULLOCH WARREN S E PITTS, 1943).

O neurônio matemático utiliza um ou mais sinais de entrada e envia um único sinal

de saída, que pode ser distribuído como sinal de saída da rede, ou então como sinal de entrada para um ou vários outros neurônios da camada posterior (que formam a rede neural artificial) (HAYKIN, 2001).

Os dendritos e axônios são representados matematicamente apenas pelas sinapses, e a intensidade da ligação é representada pelo peso sináptico, simbolizada pela letra  $w$ . Quando as entradas,  $x$  são enviadas ao neurônio, elas são multiplicadas pelos pesos sinápticos correspondentes, gerando as entradas ponderadas que representa uma das bases matemáticas do funcionamento de uma rede neural artificial, a multiplicação de matrizes:

$$Xw = y \quad (2.1)$$

$$\begin{bmatrix} 1 & x_{11} & \cdots & x_{1d} \\ 1 & x_{21} & \cdots & x_{2d} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & x_{n1} & \cdots & x_{nd} \end{bmatrix} \times \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_d \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{bmatrix}$$

O neurônio realiza os produtos gerando um único resultado chamado de função de combinação. Este resultado é apresentado a uma função de ativação (função de transferência) que tem, dentre outras, a função de evitar o acréscimo progressivo dos valores de saída ao longo das camadas da rede, visto que tais funções possuem valores máximos e mínimos contidos em intervalos determinados (HAYKIN, 2001).

A função de ativação  $\theta$  é a que processa o sinal  $u(k)$  para produzir a saída final do neurônio,  $y(k)$ . Segundo (HAYKIN, 2001), existem quatro tipos básicos de função de ativação utilizados em RNAs, conforme são descritas a seguir:

a) Função de limiar: utilizada no neurônio de McCulloch Pitts, limita a saída do neurônio a apenas dois valores (binário: 0 ou 1, ou bipolar: -1 ou 1). Normalmente é utilizada para criar neurônios que tomem decisões binárias, como nos classificadores, possui a seguinte definição:

$$\theta(.) = \begin{cases} 1 & \text{se } (.) \geq 0 \\ 0 & \text{se } (.) < 0 \end{cases} \quad (2.2)$$

b) Função linear por partes, possui a seguinte definição:

$$\theta(.) = \begin{cases} 1 & \text{se } (.) \geq 0 \\ \theta & \text{se } (.) < 0 \end{cases} \quad (2.3)$$

c) Função sigmóide, é a função geralmente adotada em redes neurais em virtude de ser contínua, monotônica, não linear e facilmente diferenciável em qualquer ponto. Possui a seguinte definição, onde  $a$  é o parâmetro de inclinação da função.

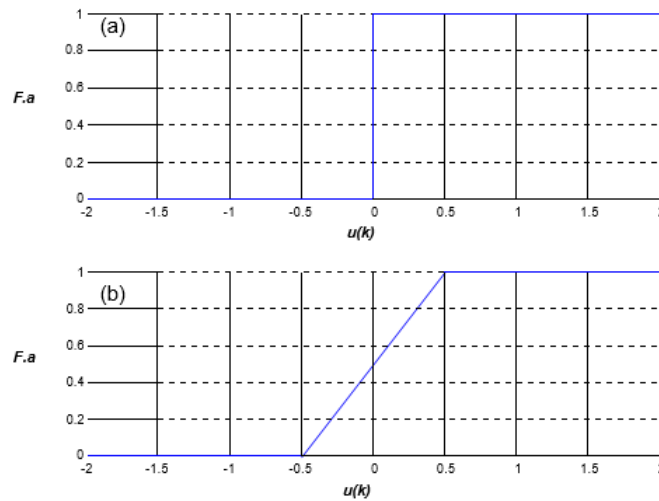
$$\theta(.) = \frac{1}{1 + e^{-a \cdot (.)}} \quad (2.4)$$

d) Função tangente hiperbólica, as funções de ativação definidas se estendem de 0 a +1. Algumas vezes é desejável que a função de ativação se estenda de -1 a +1, assumindo neste caso uma forma anti-simétrica em relação à origem. Possui a seguinte definição: Para a forma correspondente, utiliza-se a função tangente hiperbólica, definida por:

$$\theta(.) = \tanh(.) \quad (2.5)$$

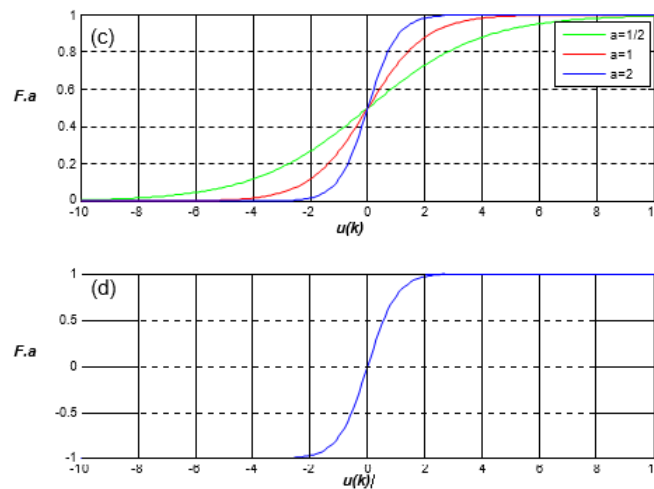
As Figuras apresenta os gráficos das funções de ativação, demonstrando o comportamento das mesmas:

Figura 2.4 – Gráficos funções de ativação a) e b)



Fonte: (HAYKIN, 2001)

Figura 2.5 – Gráficos funções de ativação c) e d)



Fonte: (HAYKIN, 2001)

Desde a criação do modelo do neurônio matemático, muitos outros modelos foram apresentados e facilitaram a criação de arquiteturas avançadas de Deep Learning como Redes Neurais Convolucionais, Redes Neurais Recorrentes, Auto Encoders, Generative Adversarial Networks, Memory Networks, entre outras.

## 2.8 A arquitetura das Redes Neurais

Um conceito importante sobre a competência da RNA, é sua arquitetura, melhor dizendo, a quantidade de neurônios e a forma como são conectados. As redes neurais são ordenadas de forma tridimensional por componentes microscópicos, contendo milhares de neurônios. Já em uma RNA, existe uma restrição do número de neurônios e camadas por causa do grande esforço computacional requerido (HAYKIN, 2001).

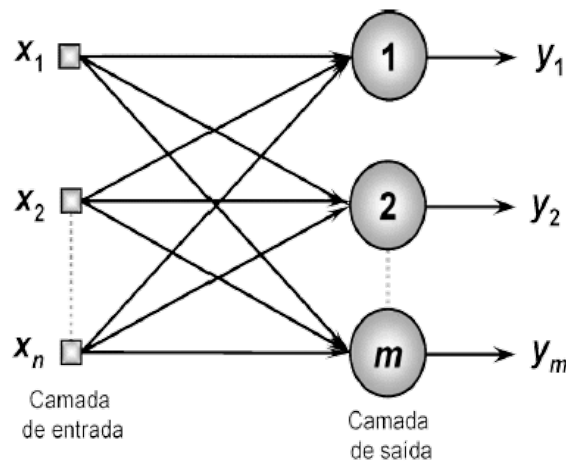
Uma rede neural pode ser estruturada com uma ou várias camadas. As que apresentam uma única camada são as que tem um nó entre uma entrada e uma saída da rede. Essa caracterização de rede é recomendada para a solução de problemas linearmente separáveis. Já as redes multicamadas detêm mais de uma camada entre as já existentes camadas de entrada e saída (HAYKIN, 2001).

O modo como os neurônios de uma rede neural estão organizados está ligado com o algoritmo de aprendizagem que é utilizado para o treinamento da rede (HAYKIN, 2001).

### 2.8.1 Redes feedforward de Camada Única

Sendo a mais simples rede de camadas os neurônios são organizados em uma única camada. A saída de cada um dos neurônios constitui uma saída da rede. Ilustrada na figura 2.6. A rede é sempre alimentada adiante, ou acíclica pois a camada de nós fonte fornece os sinais de entrada para a camada de saída e não vice-versa (não há realimentação). A definição camada única se refere ao fato de possuir uma camada de nós computacionais (a camada de saída) (HAYKIN, 2001).

Figura 2.6 – Rede alimentada adiante com uma única camada de neurônios



Fonte: A Autora

## 2.8.2 Redes feedforward de Múltiplas Camadas

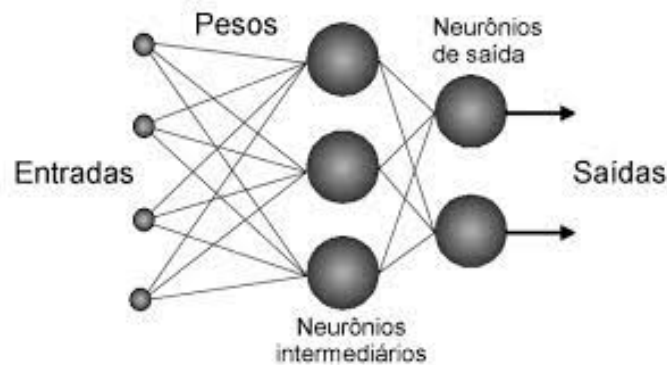
A segunda classe de redes neurais alimentada adiante possui uma ou mais camadas de nós computacionais entre as camadas de entrada e saída, as quais são denominados de camadas ocultas ou intermediárias e constituídas, por sua vez, de neurônios ocultos (SILVA; SPATTI; FLAUZINO, 2010).

O objetivo dos neurônios ocultos é interceder entre a entrada e saída da rede de maneira adequada, tornando a rede capaz de extrair estatísticas de ordem elevadas (JSTOR, ).

A competência dos neurônios ocultos extrair estatísticas de ordem elevada é muito vantajosa quando o tamanho da camada de entrada é avantajado. O processamento computacional não-linear é aumentado juntamente com a armazenagem da rede. O conjunto de saídas dos neurônios de cada camada da rede é utilizada como entrada para a camada seguinte. (HAYKIN, 2001).

Os nós de entrada da rede fornecem os elementos que formam os sinais de entrada aplicados aos neurônios (nós computacionais) na segunda camada (a primeira camada oculta). Os sinais de saída da segunda camada são usados como entradas para a terceira camada, e assim sucessivamente. A Figura 2.8 ilustra uma rede neural com uma única camada oculta, conhecida como uma rede 10-4-2 pois possui 10 neurônios de entrada, 4 ocultos e 2 de saída (HAYKIN, 2001).

Figura 2.7 – Rede alimentada adiante totalmente conectada com uma camada oculta e uma camada de saída



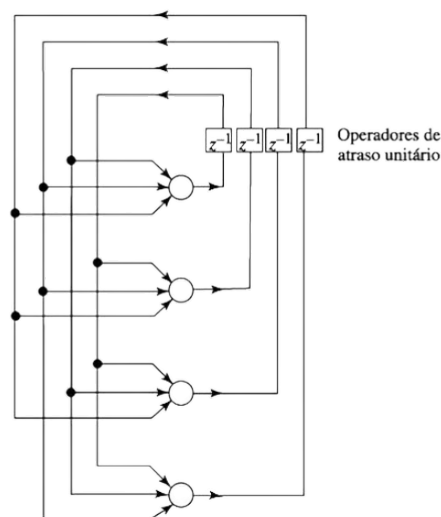
Fonte: (HAYKIN, 2001)

São geralmente treinadas usando o algoritmo de retro-propagação do erro (error back-propagation), embora existam outros algoritmos de treinamento. Este algoritmo requer a propagação direta (feedforward) do sinal de entrada através da rede, e a retro-propagação (propagação reversa, ou backpropagation) do sinal de erro,

### 2.8.3 Redes Recorrentes

As redes neurais recorrentes diferem das redes alimentadas adiante por possuírem pelo menos um laço de realimentação. Uma rede como ilustrado na 2.8 se constitui de uma única camada de neurônios como cada neurônio alimentado seu sinal de saída de volta para as entradas. Na figura em questão não há laços de auto-realimentação (auto-realimentação é uma condição onde a saída de um neurônio é realimentada para sua própria entrada) e também não possui neurônios ocultos (HAYKIN, 2001).

Figura 2.8 – Rede recorrente sem laços de auto-realimentação e sem neurônios ocultos

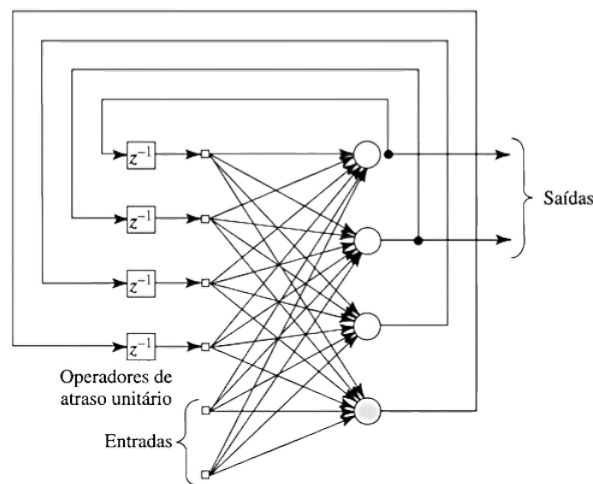


Fonte: (HAYKIN, 2001)

Na Figura 2.9 é representada uma rede recorrente com neurônios ocultos. A realimentação se originam dos neurônios ocultos e os de saída. A realimentação impacta na capacidade de aprendizagem da rede juntamente com o seu desempenho.

Além do que, os laços de realimentação envolvem a utilização de ramos particulares constituídos de elementos de atraso unitário, o que sucedem em um comportamento dinâmico não-linear, demonstrando que a rede neural abranja unidades não-lineares (HAYKIN, 2001).

Figura 2.9 – Rede recorrente com neurônios ocultos



Fonte: (HAYKIN, 2001)

## 2.9 Aprendizado

Uma RNA adquire conhecimento e melhora seu desempenho em relação a seu ambiente por meio de um processo iterativo de ajuste aplicados a seus pesos sinápticos, o treinamento. A rede se assume mais precisa sobre seu ambiente após cada interação do processo de aprendizagem (HAYKIN, 2001).

As RNA apresentam a competência de aprendizado conexionista, ou seja, não se procura obter regras como na abordagem simbólica da Inteligência Artificial (IA), mas sim estabelecer a intensidade de conexões entre neurônios (BINOTI *et al.*, 2012).

Dentre as semelhanças das RNA com o cérebro humano, as RNAs se ressaltam pela habilidade de aprender algo por meio de exemplos práticos. Ela se consiste em várias unidades de processamento simples (neurônios artificiais) conectadas entre si e arranjadas em camadas paralelas para realizar uma determinada função (BINOTI *et al.*, 2012).

Quanto à RNA em multicamadas, o processamento executado por cada neurônio é estabelecido pela combinação dos processamentos exercidos pelos neurônios da camada anterior que estão conectados a ele. Quando se avança da primeira camada intermediária

à camada de saída, as funções utilizadas se tornam cada vez mais complexas. Estas funções definem como é efetuada a divisão do espaço de decisão (VALENÇA, 2010) e (HAYKIN, 2001).

É chamado de algoritmo de aprendizado a associação de métodos bem estabelecidos para adaptar os parâmetros de uma RNA para que dessa forma possa aprender uma determinada finalidade (HAYKIN, 2001). Existem diversos algoritmos de aprendizado, cada qual com suas características suas vantagens e desvantagens. Se divergem basicamente no modo de como o ajuste dos pesos é realizado (GORGENS *et al.*, 2009), (BINOTI *et al.*, 2012).

Uma RNA deriva sua capacidade computacional com sua capacidade e aptidão de aprender e, conseqüentemente, de generalizar. A RNA gera saídas satisfatórias para as entradas que não se encontram presentes durante a aprendizagem. Dessa forma a RNA consegue solucionar problemas complexos, de grande escala, devido a essas duas capacidades de processamentos de informação (HAYKIN, 2001).

O problema a ser trabalhado é destrinchado em um número de tarefas um tanto simples e atribui-se à RNA um subconjunto de tarefas compatíveis às suas capacidades específicas (HAYKIN, 2001).

A competência de aprender é uma das qualidades marcantes das RNAs. Uma rede neural aprende, basicamente, através de um processo iterativo de ajuste de pesos e limiares (bias). Existem processos de aprendizagem ou de treinamento, que são aptos de ajustar não apenas os pesos da rede, mas também sua arquitetura e as funções de ativação dos neurônios (MANDERICK, ).

Um algoritmo de aprendizagem é formado por um conjunto pré estabelecido de regras bem definidas para a solução de um problema de aprendizagem. Não há um único algoritmo de aprendizagem para um projeto de redes neurais, existem muitas ferramentas representados por uma variedade de algoritmos, cada qual oferecendo suas vantagens específicas (HAYKIN, 2001).

Os algoritmo de aprendizagem se diferem entre si pela forma como é elaborado o ajuste de um peso sináptico de um neurônio. Outro fator que deve ser levado em consideração é a maneira pela qual a rede é constituída e como se relaciona com seu meio ambiente (HAYKIN, 2001).

O processo de aprendizagem abrange a adequação dos pesos da rede neural a partir de estímulos fornecidos por um conjunto de dados. A intenção do processo de aprendizagem é encontrar o ajuste do vetor de pesos para que o objetivo do treinamento que tende à convergência seja alcançado (REZENDE, 2003).

Aprendizagem (ou treinamento) é um procedimento pelo qual os parâmetros livres de uma rede neural são adaptados através de estímulos oferecidos pelo ambiente no qual a rede está incluída. O tipo de treinamento é definido pela forma na qual os parâmetros são modificados (HAYKIN, 2001).

Esta definição de aprendizagem implica na seguinte seqüência de eventos:

- Apresentação de estímulos à rede neural;
- Alteração dos parâmetros livres da rede; e
- Novo padrão de resposta ao ambiente.

É executado por meio de processos iterativos de adaptações aplicados aos pesos sinápticos, o chamado treinamento. O aprendizado só ocorre quando a rede neural alcança uma solução generalizada para um determinado problema. Ou seja, treinar uma rede é ajustar a sua matriz de pesos sinápticos de forma que o vetor de saída coincida com um valor desejado para cada vetor de entrada (MIRANDA; FREITAS; FAGGION, 2009).

Na etapa de aprendizagem a RNA extrai informações relevantes de informação apresentados a ela, originando uma representação própria do problema. Os parâmetros da rede são ajustados (pesos das conexões entre as unidades de processamento). Ao final da etapa de treinamento, a rede terá aprendido sobre o ambiente em que está operando, sendo este, “armazenado” em seus parâmetros (EYNG *et al.*, 2008).

Para o aprendizado das redes neurais, foram recomendados vários métodos de treinamento, sendo estes subdivididos em dois padrões principais: o aprendizado supervisionado e o não supervisionado. Um conjunto de metodologias definidas para ajustar os parâmetros de uma RNA, com a finalidade de que possa assimilar uma determinada atribuição, é intitulado de algoritmo de aprendizado (BISHOP, 1995).

A designação de uma RNA, na resolução de um determinado problema, passa inicialmente por um processo de aprendizagem, onde a rede procura extrair informações relevantes de padrões de informação apresentados a ela, modelando uma representação própria (HAYKIN, 2001).

### 2.9.1 Aprendizagem Supervisionada

A maioria das RNAs tem usufruído do treinamento supervisionado. Assim, a saída atual da rede neural é relativa a saída pretendida. Os valores iniciais dos pesos manifestados aleatoriamente, e ajustados, de acordo com algoritmo de aprendizagem, pela rede na próxima iteração (PATTERSON, 1998).

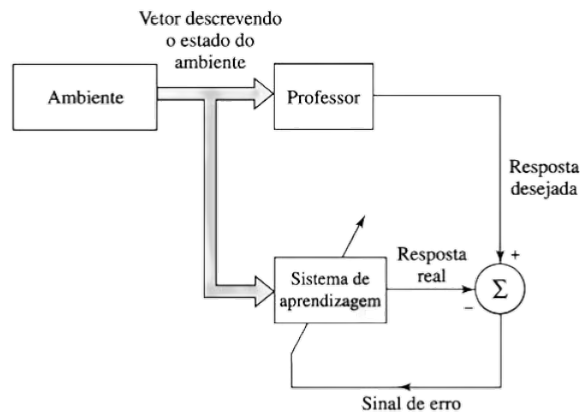
O peso sináptico é ajustado de acordo com o valor esperado e do sinal atual de saída. O método de aprendizado diminui os erros dos dados em processamento. Esta minimização de erros se empenha modificando os pesos até o momento em que a rede alcance uma precisão (PATTERSON, 1998).

O aprendizado supervisionado ou aprendizagem com professor é caracterizado pela presença de um tutor. O estado do ambiente é representado por um vetor que é aplicado

à entrada da RNA e ao professor com conhecimento sobre o ambiente o qual é reproduzido por um conjunto de exemplos de entrada e saída fornecendo assim a resposta desejada (HAYKIN, 2001).

O diagrama a seguir representa a aprendizagem supervisionada:

Figura 2.10 – Diagrama de blocos da aprendizagem com professor



Fonte: (HAYKIN, 2001)

O ambiente é desconhecido pela rede neural. Em razão de seu conhecimento prévio, o professor é apto a oferecer a rede neural uma resposta esperada para aquele vetor de treinamento (ação ótima a ser realizada pela rede neural) (HAYKIN, 2001).

Os critérios de rede são adequados sob a influência do vetor de treinamento e do sinal de erro. O sinal de erro é definido como a diferença entre a resposta desejada e a resposta real da rede. Este ajuste é realizado aos poucos, iterativamente, com o objetivo de fazer a rede neural igualar ao professor (HAYKIN, 2001).

Assim sendo, o conhecimento do ambiente é transferido para a rede neural através de treinamento. Quando esta condição é obtida, podemos então dispensar o professor e deixar a rede neural trabalhar com o ambiente (HAYKIN, 2001).

A maneira com que os sinais de entrada são representados, ou codificados, determina o maior componente constituinte para o sucesso de instrução da rede. Normalmente, as redes neurais artificiais somente manipulam, ou trabalham, com dados numéricos como entrada.

Dentre os algoritmos de treinamento supervisionado, pode-se citar como exemplo o do Erro Médio Quadrático e a generalização do mesmo, o backpropagation. Entre os modelos que utilizam este tipo de treinamento, tem-se o dos Perceptrons Multicamadas e as Redes de Base Radial.

## 2.9.2 Aprendizagem Não-Supervisionada

Nesta aprendizagem não existe um professor ou tutor responsável para oferecer a resposta desejada, ou seja, não há exemplos da função a ser aprendida pela rede. Um exemplo de rede neural que faz uso desta forma de treinamento é o denominado mapa auto-organizável (KOHONEN TEUVO E KASKI, 1997).

Na aprendizagem não-supervisionada ou auto-organizada são dados requisitos para executar uma representação que a rede deve aprender, como indicado na figura abaixo. Visto que a rede tenha se ajustado às regularidades estatísticas dos dados de entrada, ela desenvolve a capacidade de gerar representações internas para compilar as características da entrada, e assim, de criar automaticamente novas classes (BECKER, 1991).

Figura 2.11 – Diagrama de blocos da aprendizagem não supervisionada



Fonte: (HAYKIN, 2001)

Não toleram influências externas para regular os seus pesos sinápticos, pois há um monitoramento de desempenho interno, averiguando as tendências dos sinais de entrada, e dessa forma adaptando-se automaticamente as necessidades da rede (HAYKIN, 2001).

Não utiliza ajuda ou assistênica para definir se o aprendizado converge ou não para o caminho correto (autonivelção). A rede possui conhecimento de como se estruturar. Esta capacidade da rede devem-se as regras de aprendizado utilizados pela rede neural artificial (HAYKIN, 2001).

É necessário ter ressaltado a cooperação entre as camadas de unidades de processamento. A disputa entre estas unidades é a base de aprendizado da rede. Habitualmente, quando a disputa pelo aprendizado ocorre de fato, somente os pesos pertencentes à unidade de processamento ganhadora são ajustados (HAYKIN, 2001).

## 2.9.3 Redes Neurais Artificiais Multilayer Perceptron

Uma Rede Multilayer Perceptron (MLP) é caracterizada por uma camada de entrada, uma ou mais camadas ocultas/intermediárias e uma camada de saída sendo do tipo feed-forward. A RNA é inteiramente conectada, onde a saída de cada neurônio se relaciona com o neurônio da camada seguinte (HAN; PEI; KAMBER, 2011).

Muitos autores estudaram a arquitetura ideal de uma Rede MLP a exemplo de (CYBENKO, 1989), (HORNIK KURT E STINCHCOMBE; WHITE, 1989), (FUNAHASHI, 1989). Os resultados obtidos mostraram que uma camada oculta é eficaz para aproximar qualquer função contínua e duas camadas ocultas são suficientes para aproximar qualquer função matemática, visto que a quantidade de neurônios por camada oculta deve ser definido experimentalmente (RESEARCH; DEVELOPMENT; CYBENKO, 1988), (CYBENKO, 1989), (HERTZ; KROGH; PALMER, 2018).

Apesar de em algumas situações a aplicação de mais de duas camadas intermediárias possa facilitar o treinamento, sua utilização não é indicada, pois o erro propagado através da rede neural se torna menos produtivo e preciso (BRAGA; CARVALHO; LUDERMIR, 2000).

Nessa arquitetura, cada neurônio se relaciona a todos os neurônios da camada posterior. Como existe múltiplas camadas de neurônios, cada neurônio é muito semelhante ao Perceptron sugerido originalmente. Uma significativa diferença é que a saída não é mais binária, nessa configuração é dada pela aplicação de uma função de ativação não linear ao resultado da soma ponderada da entrada dos pesos da RNA (HORNIK KURT E STINCHCOMBE; WHITE, 1989).

A atuação imposta a cada neurônio é resultado da união do processamento executado pelos neurônios da camada anterior. Conforme cada camada intermediária da rede se aproxima da camada de saída, há uma delimitação do espaço de decisão dos dados que está recebendo (CARVALHO; BRAGA; LUDERMIR, 1998).

Para uma RNA com duas camadas intermediárias, a primeira camada oculta, delimita o espaço de treinamento entre as "retas traçadas" pelos neurônios, a segunda camada forma regiões convexas, onde o número de lados que constitui tal região é decidida pela quantidade de unidades conectadas a este neurônio, que combina com as retas da camada anterior. Cada neurônio da camada de saída forma regiões, originadas das combinações das regiões convexas (CARVALHO; BRAGA; LUDERMIR, 1998).

Encontra-se na literatura muitos algoritmos de aprendizado para as redes MLP, e estes geralmente possuem um aprendizado supervisionado e se classificam em dois grupos: estáticos e dinâmicos (HAYKIN, 2001).

Os algoritmos de aprendizado estáticos não modificam a organização da rede, eles mudam apenas o valor dos pesos sinápticos. Já os algoritmos dinâmicos, podem mudar a organização/forma da rede, isto é, podem alterar a quantidade de camadas, o número de neurônios da camada oculta/intermediária e o número de conexões da RNA (HAYKIN, 2001).

A MLP emprega o algoritmo de treinamento mais conhecido e aplicado, o Backpropagation. Este algoritmo dispõe de uma metodologia de adaptação de massa baseado no método do gradiente descendente, de forma que, o erro quadrático da RNA torna-se mí-

nimo (WIDROW BERNARD E LEHR, 1990). O algoritmo possui o padrão de treinamento supervisionado cujo o mesmo ocorre em duas fases: a fase forward e a fase backward.

## 2.10 Algoritmo de Aprendizagem Back Propagation

Na fase forward o conjunto de dados de entrada são apresentados à primeira camada da RNA, que determina seus sinais de saída e envia os valores para a camada seguinte e assim sucessivamente até a camada de saída calcular as saídas da rede, as quais são comparadas com as saídas desejadas.

Já a fase backward faz o caminho inverso. Ao começar da camada de saída até a de entrada e os pesos dos neurônios são acertados de forma que seus erros são diminuídos (os erros das camadas ocultas são calculados utilizando o erro dos neurônios da camada seguinte equilibrado pelo peso da conexão entre eles). Este procedimento é utilizado até atingir o critério de parada.

Tendo em vista o neurônio que compõe uma rede Multilayer Perceptron de índice  $i$  da rede, e usando o método do gradiente descendente (WIDROW BERNARD E LEHR, 1990), o ajuste dos pesos  $V$  é elaborado pela Equação 2.6:

$$V_i = V_{(k+1)} = V_i(k) + 2\gamma\beta_i X_i \quad (2.6)$$

em que:

$\gamma$  é o parâmetro de controle da estabilidade ou a taxa de treinamento,

$X_i$  o vetor contendo as entradas do neurônio  $i$ ,

$V_i$  vetor contendo os pesos do neurônio  $i$ , representados por  $[w_{0,i}, w_{1,i}, \dots, w_{n,i}]$ .

$k$  é o índice de iteração

E o  $\beta_i$  é o gradiente do erro quadrático relacionado aos pesos do neurônio  $i$ . A Equação 2.7 descreve  $\beta_i$  quando o elemento  $i$  encontra-se na última camada, então:

$$\beta_i = sgm'(S_i)\epsilon_i \quad (2.7)$$

no qual  $sgm'(si)$  é a derivada da função sigmóide com relação a  $si$ ,  $\epsilon(i) = di - yi$ ,  $di$  é a saída desejada do neurônio  $i$  e  $S_i$  é a soma dos valores do  $i$ -ésimo elemento da camada de saída da rede. Se o elemento  $i$  encontrar-se em outras colunas da camada,  $\beta_i$  é determinado na Equação 2.8:

$$\beta_i = sgm'(S_i)\epsilon_i \sum_{k \in R(j)} w_{i,k}\beta_k \quad (2.8)$$

no qual  $sgm'(si)$  é a derivada da função sigmóide em relação a  $si$ ,  $R(j)$  é o conjunto dos índices dos elementos que se encontram na fileira seguinte à fileira do elemento  $i$  e que estão interligados ao elemento  $i$ .

Para qualquer problema de aprendizagem supervisionada, seleciona-se os pesos que modela os dados de treinamento, em outras palavras, busca-se achar um conjunto de pesos  $w$  que minimize a saída de  $V(i)$ , onde  $V(i)$  é a função de perda (erro) (HASTIE TREVOR E TIBSHIRANI, 2009).

O propósito do algoritmo backpropagation é aprimorar os pesos dos neurônios para que a RNA possa compreender a mapear corretamente as entradas para as saídas (HASTIE TREVOR E TIBSHIRANI, 2009). Esse algoritmo colaborou para decifrar a contrariedade da definição do erro dos neurônios das camadas intermediárias, tornando possível, com isso, a atualização dos pesos intermediários da rede (DIAMANTOPOULOU, 2006)

Para o treinamento das redes são organizados primeiramente com as escolhas do pesos, depois as polarizações e demais parâmetros. Atribuem-se os padrões de entrada do conjunto de treinamento definindo as entradas e saídas, para depois calcular o erro para os neurônios da camada de saída, subtraindo a saída desejada da saída calculada, depois ajustando os pesos da camada de saída (PANDORFI *et al.*, 2011), (HAYKIN, 2001), (VALENÇA, 2010).

De forma simplificada pode se reunir as etapas do treinamento de uma rede usando Backpropagation da seguinte forma:

- Inicialização dos pesos, definição dos valores da taxa de aprendizado, momento do critério de parada;
- Inserção das amostras (dados) de treinamento à rede. Para cada amostra, é realizada a série de etapas do treinamento;
- Propagação: supondo que cada amostra de treinamento seja representada por  $(x(n), d(n))$ , com vetor  $x(n)$  sendo apresentado na entrada da rede e o vetor da resposta desejada  $d(n)$  apresentada na camada de saída da rede, são calculadas as saídas locais induzidas e os sinais funcionais da rede prosseguindo camada por camada;
- Retropropagação: cálculo dos gradientes locais (em cada neurônio). Ajuste dos pesos sinápticos da rede na(s) camada (s) oculta (s) de acordo com a regra delta generalizada;
- Iteração: iterar as computações de propagação e na retropropagação apresentando novas coletas de amostras de treinamento para rede até que o critério de parada determinado seja alcançado.

O algoritmo backpropagation pode ser empregado a qualquer RNA que desfrute de uma função de ativação diferencial e aprendizado supervisionado. Suas correções são exe-

cutadas nos pesos dos neurônios, durante o treinamento, os padrões de entrada são informados a rede em uma ordem pré definida, cada padrão de treinamento é propagado adiante, camada após camada, até o padrão de saída. O resultado computado pela rede é então comparado com uma saída esperada, gerando um valor que será utilizado para o cálculo do erro (PATTERSON, 1998).

(CARVALHO; BRAGA; LUDERMIR, 1998) e (PATTERSON, 1998) argumentam que o erro gerado pela comparação da saída com a esperada será aplicado como uma realimentação para as os neurônios, que resultará no ajuste dos pesos sinápticos de cada camada num sentido inverso à propagação dos sinais de treinamento, ajustando os pesos para as camadas intermediárias de acordo com as camadas posteriores.

Dessa forma, os erros calculados na camada de saída serão utilizados para ajustar os pesos entre a última camada intermediária com a camada de saída. Com isso, o erro computado de uma camada oculta será usado para ajustar os pesos da camada oculta anterior. Este procedimento será repetido até que a primeira camada intermediária seja adequada.

## 2.11 Mapas Auto-Organizáveis de Kohonen

Os mapas auto-organizáveis como modelo neural foi inspirado por uma característica do córtex cerebral humano, que é a organização das informações em muitas regiões, de tal forma que entradas sensoriais distintas são representadas como mapas bidimensionais sendo considerado relativamente simples e com a capacidade de organizar dimensionalmente dados complexos em grupos (clusters), de acordo com suas relações.

O Mapa Auto-Organizável de Kohonen (Self-Organizing Map- SOM) desenvolvido por Teuvo Kohonen, é um tipo de rede neural artificial fundamentada em aprendizado competitivo e não supervisionado, sendo preparado para mapear um conjunto de dados, estes algoritmos podem analisar dados por agrupamentos com o objetivo de descobrir estruturas e padrões multidimensionais.

Habilitado para reduzir a dimensão de um conjunto de dados, conseguindo manter a representação real com relação as propriedades relevantes dos vetores (dados) de entrada, apresentando como resultado um conjunto das características do espaço de entrada (KOHONEN, 1981).

A rede de Kohonen possui a capacidade de efetuar mapeamentos que preservam a topologia entre as camadas de entrada e de saída. São aplicadas em projetos industriais como ferramentas para resolver problemas práticos de difícil solução. Seu objetivo é encontrar padrões significativos ou característicos dos valores de entrada.

É indicada para resolver problemas não-lineares de alta dimensionalidade, como por exemplo a extração de características e classificação de imagens e padrões acústicos, con-

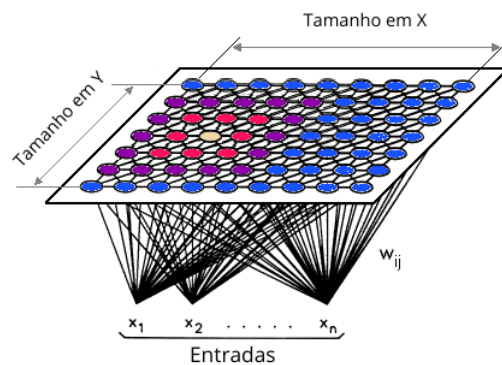
trole adaptativo de robôs, equalização, modulação e transmissão de sinais (CASTRO; CASTRO, 2001).

O algoritmo tradicional de treinamento do SOM utiliza da atualização dos pesos sinápticos dos neurônios do conjunto sempre que um item de dados é inserido à rede, ficando conhecido como incremental ou “on-line”. Em uma outra versão, as atualizações individuais são desprezadas e utilizadas após a apresentação de todos os valores do conjunto de dados recebendo o nome de algoritmo em lote ou “batch” (KOHONEN, 2001).

O modelo mais divulgado e aplicado para fins de mineração de dados são os que sugere a construção de arranjos em 2 dimensões mas são possíveis construções em uma, duas ou mais dimensões, de acordo com a necessidade e o objetivo. Uma vez adaptado, é indispensável algum método que possibilite a interpretação do resultado alcançado.

Com o domínio de criar de modo efetivo as representações internas espacialmente organizadas de vários recursos de sinais de entrada e suas abstrações, os neurônios em uma SOM são estabelecidos nos nós de uma grade (que possui uma topologia particular, que pode ser retangular, hexagonal etc.), que é habitualmente de uma ou duas dimensões. Mapas de dimensões maiores são também possíveis, contudo de mais difícil aplicação e compreensão. Logo, pode-se dizer que o SOM diminui a dimensão dos dados e é exibido de forma semelhante entre os dados (KOHONEN TEUVO E KASKI, 1997).

Figura 2.12 – Redução de dimensionalidade usando SOM



Fonte: Adaptado de (VILLMANN; BAUER, 1998)

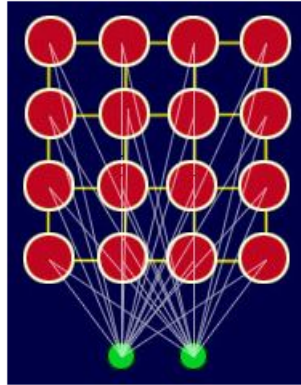
### 2.11.1 Arquitetura de rede de SOM

São indicados os dados de entrada à rede, e a cada valor apresentado possui uma região de atividade na grade. Cada neurônio na grade é conectado a todos os neurônios da camada de entrada. A localização e natureza de uma estabelecida região variam de um padrão de entrada para outro.

Portanto, todos os neurônios da rede devem ser conectados a um número sufici-

ente de diferentes padrões de entrada, assegurando assim que o procedimento de auto-organização ocorra de forma apropriada. Para entendimento, a Figura 2.14 ilustra uma rede Kohonen 2D de dimensões 4x4 que está totalmente conectada a dois nós de entrada (KOHO-NEN, 1981).

Figura 2.13 – Uma simples rede de Kohonen



Fonte:(VILLMANN; BAUER, 1998)

Cada nó tem uma posição topológica específica (uma coordenada  $x, y$  na rede) e contém um vetor de pesos da mesma dimensão que os vetores de entrada. Ou seja, se os dados de treinamento consistem em vetores,  $V$ , de  $n$  dimensões:

$$V_1, V_2, V_3 \dots V_n \quad (2.9)$$

Em seguida, cada nó conterá um vetor de peso correspondente  $W$ , de  $n$  dimensões:

$$W_1, W_2, W_3 \dots W_n \quad (2.10)$$

As linhas que ligam os nós na Figura 2.14 estão apenas lá para representar a adjacência e não significam uma conexão como normalmente indicado ao discutir uma rede neural. Não há conexões laterais entre nós dentro da rede.

### 2.11.2 O algoritmo de aprendizagem

Um SOM não precisa de uma saída de destino para ser especificado ao contrário de muitos outros tipos de rede. Em vez disso, quando os pesos do nó correspondem ao vetor de entrada, essa área da rede é seletivamente otimizada para se assemelhar mais aos dados da classe do qual o vetor de entrada é membro (KOHONEN, 2001).

A partir de uma distribuição inicial de pesos aleatórios, e ao longo de muitas iterações, o SOM eventualmente se instala em um mapa de zonas estáveis. Cada região é efetivamente um classificador de recursos, então pode-se pensar na saída gráfica como um tipo de mapa de recurso do espaço de entrada (KOHONEN, 1981).

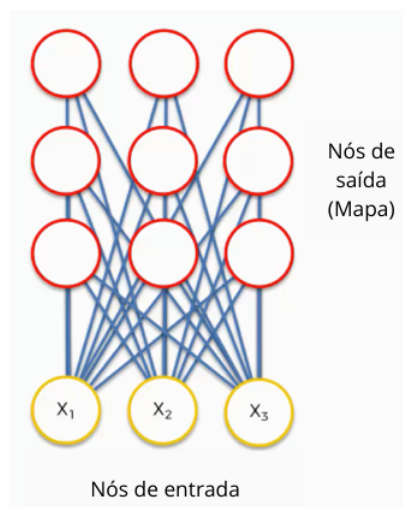
O treinamento ocorre em várias etapas e em muitas iterações:

- 1. Os pesos de cada nó são inicializados.
- 2. Um vetor é escolhido aleatoriamente a partir do conjunto de dados de treinamento e apresentado à rede.
- 3. Cada nó é examinado para calcular quais os pesos são mais parecidos com o vetor de entrada. O nó vencedor é comumente conhecido como a Melhor Unidade de Correspondência (BMU).
- 4. O raio do bairro da BMU é agora calculado. Este é um valor que começa grande, tipicamente definido para o "raio" da rede, mas diminui a cada passo de tempo. Qualquer nódulo encontrado neste raio é considerado dentro do bairro da BMU.
- 5. Os pesos de cada nó vizinho (os nós encontrados na etapa 4) são ajustados para torná-los mais parecidos com o vetor de entrada. Quanto mais perto um nó está para o BMU; quanto mais seus pesos são alterados.
- 6. Repita o passo 2 para as iterações N.

### 2.11.3 Etapas do Algoritmo de aprendizagem

Para entendimento do Algoritmo, considere o seguinte mapa auto-organizável da Figura 2.15:

Figura 2.14 – Exemplo de SOM



Fonte: Adaptado de (VILLMANN; BAUER, 1998)

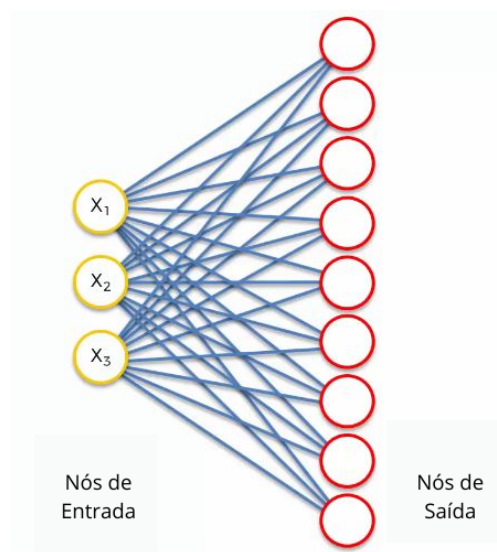
Os vetores de entrada somam três características, e tem-se nove nós de saída. Dito isso, os três nós de entrada representam três colunas (dimensões) no conjunto de dados,

mas cada uma dessas colunas pode conter milhares de linhas. Os nós de saída em um SOM são sempre bidimensionais (KOHONEN, 1981).

Agora, transforma-se este SOM em um conjunto de entrada que seria mais familiar a partir dos métodos supervisionados de aprendizado de máquina (redes neurais artificiais, convolucionais e recorrentes) discutidos em seções anteriores.

Considere a Estrutura de Auto-Organização que possui 3 nós de entrada visíveis e 9 saídas que estão conectadas diretamente à entrada conforme mostrado abaixo da Figura 2.16.

Figura 2.15 – Exemplo de SOM



Fonte: Adaptado de (VILLMANN; BAUER, 1998)

Os valores de entrada são:

$$x_1 = 0,7 \quad x_2 = 0,6 \quad x_3 = 0,9 \quad (2.11)$$

### 2.11.3.1 Passo 1: Inicializando os pesos

Agora, seleciona-se o nó de saída mais alto e foca em suas conexões com os nós de entrada. Como ilustrado na figura 2.16, há um peso atribuído a cada uma dessas conexões.

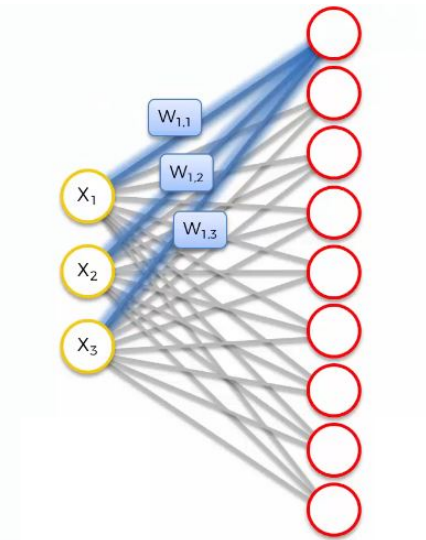
Mais uma vez, a palavra "peso" aqui carrega um significado totalmente diferente do que em redes neurais artificiais e convolucionais. Por exemplo, com redes neurais artificiais multiplica-se o valor do nó de entrada pelo peso e, finalmente, aplica-se uma função de ativação. Com SOMs, por outro lado, não há função de ativação (VILLMANN; BAUER, 1998).

Em um SOM, os pesos pertencem ao nó de saída em si. Em vez de ser o resultado da soma dos pesos, o nó de saída em um SOM contém os pesos como suas coordenadas. Carregando esses pesos, ele tenta furtivamente encontrar seu caminho para o espaço de entrada (KOHONEN, 2001).

Neste exemplo, tem-se um conjunto de dados 3D, e cada um dos nós de entrada representa uma coordenada  $x$ . O SOM comprimiria-os em um único nó de saída que carrega três pesos. Se considerado um conjunto de dados de 20 dimensões, o nó de saída, neste caso, carregaria 20 coordenadas de peso.

Cada um desses nós de saída não se torna exatamente partes do espaço de entrada, mas tenta integrar-se a ele, no entanto, desenvolvendo lugares imaginários para si mesmos (KOHONEN, 1981).

Figura 2.16 – Exemplo de SOM



Fonte: Adaptado de (VILLMANN; BAUER, 1998)

Inicia-se aleatoriamente os valores dos pesos (perto de 0, mas não 0).

Figura 2.17 – Exemplo de SOM

$W_{1,1} = 0.31$	$W_{1,2} = 0.22$	$W_{1,3} = 0.10$
$W_{2,1} = 0.21$	$W_{2,2} = 0.34$	$W_{2,3} = 0.19$
$W_{3,1} = 0.39$	$W_{3,2} = 0.42$	$W_{3,3} = 0.45$
$W_{4,1} = 0.25$	$W_{4,2} = 0.32$	$W_{4,3} = 0.62$
$W_{5,1} = 0.24$	$W_{5,2} = 0.31$	$W_{5,3} = 0.16$
$W_{6,1} = 0.52$	$W_{6,2} = 0.33$	$W_{6,3} = 0.42$
$W_{7,1} = 0.31$	$W_{7,2} = 0.22$	$W_{7,3} = 0.10$
$W_{8,1} = 0.12$	$W_{8,2} = 0.41$	$W_{8,3} = 0.19$
$W_{9,1} = 0.34$	$W_{9,2} = 0.40$	$W_{9,3} = 0.51$

Fonte:(VILLMANN; BAUER, 1998)

### 2.11.3.2 Calcular a melhor unidade de correspondência (BMU)

O próximo passo é analisar o conjunto de dados. Para cada uma das linhas do conjunto de dados, tenta-se encontrar o nó mais próximo a ele. Diga-se que seja escolhido a linha número 1, e seja extraído seu valor para cada uma das três colunas. Então deseja-se descobrir qual dos nós de saída está mais próximo dessa linha.

Para determinar a melhor unidade de correspondência, um método é iterar através de todos os nós e calcular a distância euclidiana entre o vetor de peso de cada nó e o vetor de entrada atual. O nó com um vetor de peso mais próximo do vetor de entrada é marcado como BMU (KOHONEN TEUVO E KASKI, 1997).

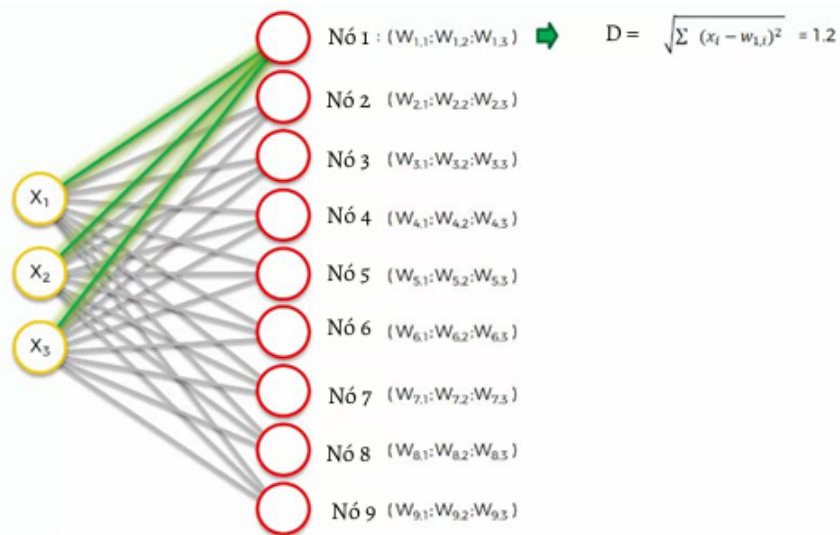
A distância euclidiana é dada como:

$$D = \sqrt{\sum_{i=1}^n (X_i - W_i)^2} \quad (2.12)$$

Onde  $X$  é o vetor de entrada atual e  $W$  é o vetor de peso do nó. Agora calcula-se a Melhor Unidade de Correspondência usando a fórmula Distância para todos os nós.

Para o 1º Nó:

Figura 2.18 – Exemplo de SOM



Fonte: Adaptado de (VILLMANN; BAUER, 1998)

Por exemplo, considere que o nó número 3 é o mais próximo com uma distância de 0,4. Então será chamado de nó de BMU (unidade mais compatível).

Em termos simples, o SOM está se aproximando do ponto de dados, esticando o BMU em direção a ele. O objetivo final é ter o mapa tão alinhado com o conjunto de dados como visto acima.

### 2.11.3.3 Ajuste dos pesos

Cada nó dentro do bairro da BMU (incluindo o BMU) tem seu vetor de peso ajustado de acordo com a seguinte equação: Novos Pesos = Pesos Antigos + Taxa de Aprendizagem (Vetor de Entrada — Pesos Antigos)

$$W(t + 1) = W(t) + L(t)(V(t) - W(t)) \quad (2.13)$$

Onde  $t$  representa o passo do tempo e  $L$  é uma pequena variável chamada taxa de aprendizagem, que diminui com o tempo.

Finalmente, a partir de uma distribuição aleatória de pesos e através de muitas iterações, SOM pode chegar a um mapa de zonas estáveis. No final, a interpretação dos dados deve ser feita por um humano, mas a SOM é uma ótima técnica para apresentar os padrões invisíveis nos dados (KOHONEN, 1981).

## 2.12 Estratégias de Validação

A acurácia preditiva informa a capacidade do modelo de classificação prever corretamente a classe de um objeto desconhecido. Esta estimativa permite a validação do modelo gerado, bem como a comparação de desempenhos entre diferentes classificadores sobre uma mesma base de teste.

### 2.12.1 Validação Cruzada

Neste trabalho, para obter a acurácia preditiva dos modelos de classificação, utilizou-se técnicas de validação cruzada, do inglês cross-validation, que são técnicas estatísticas úteis para determinar a capacidade de generalização do modelo a partir dos dados (DUDA; HART *et al.*, 2006).

Para realizar a validação cruzada, a base de dados é dividida em duas partes, são elas:

- **Treinamento:** Utilizados para treinar o modelo. O treinamento permite que sejam observadas características dos objetos e as relações entre os dados de entrada e saída para criar uma representação deste dados.
- **Teste:** Geralmente é um conjunto menor que o utilizado no treinamento. O conjunto de teste permite a validação do modelo gerado.

A validação cruzada garante que os dados utilizados para treinamento e teste são diferentes. Garantindo assim, uma maior veracidade na capacidade de generalização do modelo preditivo.

A seguir, é descrita o tipo de validação cruzada usada neste trabalho, a Matriz de Confusão.

### 2.12.1.1 Matriz de Confusão

Em um processo de classificação com apenas duas classes, por exemplo, aprovado ou reprovado, cada predição pode gerar resultados conforme a 2.19. Os verdadeiros positivos ( $Vp$ ) e os verdadeiros negativos ( $Vn$ ) são classificações corretas. Um falso positivo ( $Fp$ ) ocorre quando o resultado foi predito incorretamente como aprovado. Um falso negativo ( $Fn$ ) ocorrerá quando um resultado é predito incorretamente como reprovado.

Figura 2.19 – Diferentes resultados para uma predição de duas classes

		Classe Predita	
		Aprovado	Reprovado
Classe Correta	Aprovado	Verdadeiro Positivo	Falso Negativo
	Reprovado	Falso Positivo	Verdadeiro Negativo

Fonte: A Autora

A taxa de verdadeiros positivos é  $Vp$  dividida pelo total de números positivos, que é  $Vp + Vn$ . A taxa de falsos negativos é  $Fn$  dividido pelo total de números dos negativos,  $Fn + Vp$ . A taxa de sucesso global, ou acurácia do classificador  $Ac$ , é o número de classificações corretas dividida pelo número total de classificações:

$$Ac = \frac{Vp + Vn}{Vp + Vn + Fp + Fn} \quad (2.14)$$

Em uma predição com mais de duas classes, o resultado pode ser informado em uma matriz de confusão com duas dimensões, com uma linha e uma coluna para cada classe. Cada célula da matriz mostra o número de acertos da predição em relação ao esperado. As linhas representam os resultados esperados e as colunas os valores preditos. Bons resultados são obtidos quando grandes números ficam na diagonal principal, indicando que uma grande quantidade de predições das classes foram de acordo com as classes esperadas.

A 2.20 mostra um exemplo de matriz de confusão para uma predição de três classes.

Figura 2.20 – Matriz de confusão com resultados de uma predição com três classes

		Classes Previstas		
		A	B	C
Classes Corretas	A	84	15	1
	B	6	78	8
	C	0	10	59

Fonte: A Autora

## 2.13 Medidas de Avaliação

### 2.13.1 Precisão

A precisão é definida pela razão entre os exemplos classificados corretamente, representado pelo parâmetro  $Vp$  (verdadeiros positivos), presente no numerador da equação 2.9, e todos aqueles que foram associados a uma determinada classe pelo modelo avaliado, representado pelo denominador da mesma equação, que calcula a soma entre  $Vp$  e  $Fp$  (falsos positivos).

$$P = \frac{Vp}{Vp + Fp} \quad (2.15)$$

### 2.13.2 F-Measure

F-measure é uma medida que calcula a média harmônica ponderada dos valores de precisão e revocação, representados por  $P$  e  $R$ , respectivamente, na equação 2.14.

$$F = 2 \frac{PR}{P + R} \quad (2.16)$$

## Capítulo 3

---

# Metodologia

---

Este capítulo apresenta a metodologia de aquisição de dados educacionais, pré- processamento dos dados, estudos de caso para os modelos de RNA, modelagem de rede neural artificial para cada técnica de aprendizado e a metodologia de avaliação do desempenho da RNA modelada.

### 3.1 Coleta de Dados Educacionais

Este trabalho utilizou dados reais gerados entre 2009 e 2019 nas bases de dados do SIGAA - Sistema Intregado de Gestão de Atividades Acadêmicas, da instituição de ensino superior UFCat (Universidade Federal de Catalão, anteriormente regional da UFG - Universidade Federal de Goiás), localizada em Catalão, Goiás.

Por ser o objetivo deste trabalho utilizar métodos para predição de desempenho de estudantes ao longo da jornada acadêmica, foram considerados estudantes desistentes, formados e aqueles que não conseguiram terminar o curso no tempo previsto de formação (4 anos).

Os dados da amostra foram coletados sem a identificação do aluno, disponibilizados pelo Chefe do departamento do IMTEC (Unidade Acadêmica Especial de Matemática e Tecnologia), autorizados para estudo pelo Pró reitor de Graduação atual da Universidade Federal de Catalão (UFCAT), dos alunos graduados e desistentes no curso de Matemática Industrial nos anos de 2009 a 2019.

Dentre as diversas disciplinas disponíveis na base de dados, seguiu-se os seguintes critérios para escolha de uma amostra para os experimentos:

- Disciplinas obrigatórias do curso
- Disciplinas cursadas por todos os alunos supracitados

O resultado abrangeu 61 alunos graduados e 89 alunos que não conseguiram concluir a formação (seja por desistência ou por estar a mais de 4 anos no curso), onde foram coletados na amostra os dados contendo notas de todas as disciplinas cursadas por cada aluno.

Por não ser foco deste trabalho não foram considerados os alunos que transferiram para outro curso, pois estes não caracterizariam nem bom nem mau desempenho enquanto esteve no curso.

Após o processo de coleta de dados ser finalizada, foi utilizado as disciplinas cursadas até o 3º Período do curso, pelo fato de que após este período existiram a maior quantidade de evasão.

Com isso, tem-se o conjunto de dados com as notas dos alunos para serem considerados nos modelos de RNA:

- Conjunto - Disciplinas cursadas até o 3º Período - Cálculo 1, Álgebra Linear, Algoritmo e Programação de Computadores, Geometria Analítica, Leitura e Produção Textual I, Química Geral, Seminários sobre a profissão e Matemática Financeira, Inglês I, Química Geral Experimental, Cálculo II, Programação Orientada a Objetos, Cálculos em Química, Instrumentação Mecânica, Cálculo Numérico, Eletromagnetismo aplicado a Indústria, Estrutura de Dados, Inglês Instrumental II, Leitura e Produção Textual II, Programação Linear, Cálculo III e Estatística Descritiva - total de 22 disciplinas.

Em suma, o conjunto possui dados estruturados (mesmos campos e quantidades para todas as ocorrências) possuindo 22 variáveis de entrada e com a 23ª coluna representando as classes de resposta (saída dos modelos).

A classe de saída, no qual os modelos devem se basear para classificação no treinamento e teste, é obtida pela evolução da nota do aluno. Em resumo, as classes são:

- Classe 0 - Bom Desempenho
- Classe 1 - Melhorando
- Classe 2 - Piorando
- Classe 3 - Desempenho ruim

## 3.2 Imputação de dados

Após a coleta dos dados educacionais, foi notado a falta de notas de disciplinas, ocasionadas por falha do sistema e pelo aproveitamento da matéria, o que é apresentado no histórico do aluno apenas um 'AV'.

Treinar um modelo com um conjunto de dados com muitos valores ausentes pode afetar drasticamente a qualidade do modelo de aprendizado de máquina. Alguns algoritmos, como estimadores scikit-learn, pressupõem que todos os valores são numéricos e têm e mantêm um valor significativo.

Fazendo um levantamento da quantidade de colunas dos dados que possui valores ausentes, 19 colunas apresentaram dados faltantes no Conjunto.

O Quadro 3.1 descreve a quantidade de dados ausentes em cada coluna das 19 do conjunto, e a porcentagem de ausência na amostra total.

Coluna	Valores ausentes	% sobre o Conjunto
0	1	0,7%
2	3	2%
3	2	1,3%
4	5	3,3%
5	5	3,3%
6	3	2%
7	4	2,7%
8	8	5,3%
9	4	2,7%
11	2	1,3%
13	3	2%
14	8	5,3%
15	5	3,3%
16	1	0,7%
17	3	2%
18	4	2,7%
19	1	0,7%
20	1	0,7%
21	5	3,3%
Total	68	45,3%

Fonte: A autora.

quadro 3.1 – Colunas que possuem valores ausentes no Conjunto

Após localizar os valores ausentes no conjunto de dados, atribui-ser o valor NaN ou "?" para cada campo ausente e, assim, possibilitar o processo de imputação numérica. A Figura 3.1 ilustra uma parte dos dados onde ocorreu o processo de localização e atribuição.

Figura 3.1 – Dados com valores ausentes

95	6.0	4.3	3.6	6.0	NaN	7.8	10.0	NaN	6.6	3.1	0.0	1.4	0.0	0.0	0.0	0.0	7.1	9.3	0.0	0.0	6.3	
104	0.0	0.0	0.0	0.0	NaN	0.0	NaN	0.0	6.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
112	0.0	1.6	1.4	1.3	6.4	4.3	10.0	7.5	NaN	2.2	0.5	6.1	0.0	3.1	0.0	0.1	0.0	NaN	0.0	0.0	4.3	
115	NaN	0.7	NaN	4.5	NaN	NaN	NaN	NaN	8.0	3.1	0.0	0.0	0.0	0.5	0.0	0.0	0.0	3.2	0.0	0.0	4.9	
116	0.0	0.0	0.0	0.0	0.0	0.0	10.0	0.0	8.5	2.5	0.0	1.0	0.0	4.7	5.5	1.0	0.0	NaN	6.3	0.0	5.0	
134	0.5	1.7	1.6	0.1	5.0	3.5	7.2	5.6	NaN	NaN	0.0	0.0	0.0	NaN	NaN	0.0	0.0	NaN	NaN	NaN	NaN	
135	0.0	0.5	1.0	0.1	5.4	0.2	6.8	4.4	NaN	NaN	0.0	0.0	0.0	NaN	0.0	0.0	0.0	NaN	0.0	0.0	NaN	
141	0.0	0.0	2.3	0.7	6.9	4.0	5.5	5.0	8.8	2.6	6.0	0.0	0.0	6.7	NaN	0.0	0.0	9.7	9.2	6.0	2.8	5.0
144	2.3	0.6	0.5	5.0	3.4	0.6	1.2	3.1	NaN	NaN	0.0	0.0	5.0	0.0	0.0	0.0	5.0	10.0	10.0	8.0	0.0	9.8
145	0.5	0.6	0.7	1.5	1.9	5.2	4.3	0.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	NaN	
148	1.0	0.0	2.3	0.0	4.1	5.0	3.0	0.5	NaN	6.5	0.0	0.0	6.0	0.0	0.0	0.0	0.0	8.0	6.8	0.0	0.0	0.0
149	1.0	0.0	2.3	0.0	4.1	5.0	3.0	0.5	NaN	6.5	0.0	0.0	6.0	0.0	0.0	0.0	0.0	8.0	6.8	0.0	0.0	0.0

Fonte: Autora

Para compensar os valores ausentes nas amostras coletadas, utilizou-se a forma de imputação de dados por maior frequência. Essa técnica tem desempenho considerável em dados numéricos como é o caso deste trabalho.

A imputação Mais frequente (moda/mediana) é uma estratégia estatística para imputar valores ausentes. Ela funciona com recursos categóricos (strings ou representações numéricas), substituindo os dados ausentes pelos valores mais frequentes em cada coluna.

As figuras 3.2 ilustra a imputação por maior frequência feita no conjunto de dados (Figura 3.3).

Figura 3.2 – Imputação nos dados usando a maior frequência dos valores

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
0	5.0	8.5	8.4	7.0	9.1	5.3	8.5	8.3	7.7	5.3	5.2	5.8	8.3	8.5	9.3	7.8	7.0	8.0	9.6	5.7	7.1	9.7
1	9.0	6.1	6.6	5.5	7.0	5.8	7.5	8.8	7.2	5.8	6.1	5.5	5.6	6.0	9.2	5.3	5.4	7.5	7.1	6.6	5.4	7.4
2	5.0	5.6	7.7	5.5	7.5	5.6	5.0	7.4	5.0	5.6	5.5	5.4	5.4	5.5	7.2	5.2	5.4	6.0	6.7	6.0	7.0	9.9
3	10.0	8.9	7.4	6.4	9.4	5.6	8.5	7.4	5.4	5.6	7.3	5.8	6.7	6.2	7.1	5.2	6.9	9.0	7.2	5.0	5.2	9.2
4	5.0	5.0	5.1	7.5	7.8	5.8	8.0	8.3	6.3	5.8	5.1	5.1	5.3	5.2	6.7	5.0	5.1	9.0	8.0	6.0	6.1	8.7
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
145	0.5	0.6	0.7	1.5	1.9	5.2	4.3	0.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
146	1.2	0.2	4.6	6.2	1.0	0.2	4.0	0.3	10.0	6.0	0.0	0.0	5.0	0.0	3.2	0.0	0.0	10.0	9.5	0.0	0.0	10.0
147	0.0	0.0	0.0	1.3	0.5	0.0	7.3	0.0	8.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	8.0	5.6	0.0	0.0	0.0
148	1.0	0.0	2.3	0.0	4.1	5.0	3.0	0.5	0.0	6.5	0.0	0.0	6.0	0.0	0.0	0.0	0.0	8.0	6.8	0.0	0.0	0.0
149	1.0	0.0	2.3	0.0	4.1	5.0	3.0	0.5	0.0	6.5	0.0	0.0	6.0	0.0	0.0	0.0	0.0	8.0	6.8	0.0	0.0	0.0

Fonte: Autora

Agora, com os dados trabalhados e sem valores ausentes, os testes e modelagens podem ocorrer sobre a amostra.

### 3.3 Modelagem da Rede Neural

O primeiro passo para a construção de uma RNA é definir qual o tipo de arquitetura que ela terá. A partir da revisão de literatura, determinou-se que neste trabalho será utilizada as arquiteturas de perceptron de multi camadas (MLP - supervisionado) e Mapas auto-organizáveis de Kohonen (aprendizado não supervisionado).

A MLP permite, além da camada de entrada e saída da rede, camadas ocultas ou intermediárias. Estudos comprovam que os perceptrons de multi camadas são aproximadores universais, e as redes MLP com apenas uma camada oculta são suficientes para aproximar qualquer função contínua, dado o número necessário de neurônios ((CYBENKO, 1989); (HORNIK KURT E STINCHCOMBE; WHITE, 1989)). Portanto, a RNA do tipo MLP aqui proposta apresenta apenas uma camada oculta, além das camadas de entrada e de saída.

Por sua vez, os mapas auto organizáveis de Kohonen é uma RNA caracterizada por um algoritmo de aprendizagem não-supervisionado para o qual são fornecidos apenas os dados de entrada, e neurônios em uma segunda camada discreta e ordenada, dita competitiva, gradualmente adaptados para responder seletivamente a entradas similares.

Essencialmente, para cada vetor de dados na entrada o neurônio na camada competitiva cujo vetor de pesos é o mais próximo do vetor entrante, segundo uma métrica qualquer, toma-se o vencedor e tem seu vetor de pesos ajustado para aproximar do vetor de entrada.

#### 3.3.1 Parâmetros da MLP

Os dados do Conjunto, com exceção da coluna de Desempenho, são os dados de entrada da RNA. Ou seja, como foram consideradas vinte e duas variáveis na rede, ela foi composta por uma camada de entrada com vinte e dois neurônios, cada qual correspondendo a uma das disciplinas já citadas neste trabalho.

Na camada seguinte, denominada camada oculta ou intermediária, foram testadas inicialmente arquiteturas com quantidades de 01 a 20 neurônios, de forma a verificar qual quantidade de neurônios gera os menores erros na camada de saída.

Na camada de saída, os valores gerados são 0, 1, 2 ou 3, resultantes da interpretação prévia da evolução das notas de cada aluno do conjunto de treinamento. Assim, realizou-se um arredondamento entre os valores de saída para as quatro classes de desempenho, onde:

- Classe 0 - Alunos com ótimas notas ao longo dos períodos
- Classe 1 - Alunos que estão melhorando suas notas
- Classe 2 - Alunos em que suas notas estão em declínio
- Classe 3 - Alunos com notas ruins ao longo dos períodos

Como são quatro classes na camada de saída, a RNA contempla quatro neurônios na última camada.

### 3.3.2 Função de ativação para a MLP

Outro parâmetro a ser considerado é a função de ativação. Para ambos modelos acima descritos, a função de ativação escolhida para as camadas de entrada e intermediária foi a ReLU. A função ReLU é a unidade linear rectificadora. É definida como:

$$f(x) = \max(0, x) \quad (3.1)$$

ReLU é a função de ativação amplamente utilizada ao projetar redes neurais atualmente. Primeiramente, a função ReLU é não linear, o que significa que os erros podem ser limitados e ter várias camadas de neurônios ativados pela função ReLU. Sua vantagem está na não ativação de todos os neurônios ao mesmo tempo, isso significa que, ao mesmo tempo, apenas alguns neurônios são ativados, tornando a rede esparsa e eficiente e fácil para a computação.

Para a camada de saída, foi adotado a MLP com classificação multiclasse, que utiliza como função de ativação a SoftMax. A função softmax transforma as saídas para cada classe para valores entre 0 e 1 e também divide pela soma das saídas. Isso essencialmente dá a probabilidade de a entrada estar em uma determinada classe. Pode ser definido como:

$$\sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^K e^{z_k}} \text{ para } j = 1, \dots, K \quad (3.2)$$

Por exemplo, as saídas como [1.2, 0.9, 0.75], quando aplicada a função softmax, obtêm-se [0.42, 0.31, 0.27]. Então, usa-se como probabilidades de que o valor seja de cada classe. A função softmax é idealmente usada na camada de saída do classificador, onde realmente tenta-se gerar as probabilidades para definir a classe de cada entrada.

### 3.3.3 Parâmetros da SOM

A rede SOM, assim como a maioria dos modelos conexionistas, necessita que alguns parâmetros sejam pré-especificados de modo a garantir um correto treinamento. (KOHONEN, 2001) ressalta que a determinação do tamanho da rede SOM e dos seus parâmetros de aprendizagem é um processo empírico, baseado na experiência do usuário e em métodos de tentativa e erro.

Outro item de fundamental importância no projeto da rede SOM é a definição do número de neurônios. Outros parâmetros livres a serem ajustados para o correto treinamento da rede SOM são o número total de épocas de treinamento, o número de repetições da simulação e o raio inicial da função de vizinhança. A seguir, discutir-se-ão as heurísticas para especificação de tais parâmetros livres.

A determinação dos parâmetros da rede SOM que ofereçam os melhores resultados para a visualização dos agrupamentos e que satisfaçam as propriedades inerentes da rede (dimensionalidade do mapa – 3 x 3, 5 x 5, 10 x 10; número de épocas de treinamento, número de repetições do treinamento etc.), pode vir a influenciar o resultado produzido pelo algoritmo, o que torna a aplicabilidade da rede SOM a um problema prático um grande desafio (BOSCARIOLI, 2008).

Para o Conjunto de dados educacionais, a rede apresenta vinte e dois neurônios na camada de entrada, onde cada um representa uma disciplina da grade curricular do 1º período ao 3º período de formação e a rede realizará os agrupamentos de acordo com estes recursos.

Vários mapas foram elaborados para cada classificação, variando a topologia (retangular ou hexagonal), raio inicial e taxa de aprendizagem inicial. Os dados foram fornecidos em forma seqüencial, por espécies e de forma aleatória, até que um mapa fosse classificado como bom, ou seja, que conseguisse separar as classes de uma forma visualmente clara.

Cada época do treinamento corresponde à apresentação de todos os vetores de entrada para a rede. O critério de parada utilizado foi o número máximo de épocas, buscando um mapa que apresentasse menor erro. Para se avaliar a qualidade de um mapa, normalmente são utilizados dois critérios: resolução e preservação da topologia.

Quando o trabalho envolve um conjunto de dados, com uma dimensão maior que a grade do mapa, estes critérios tornam-se contraditórios. A contradição reside no fato de que, ao escolher um mapa que apresente um menor erro quantitativo, ele deverá ser um mapa “grande”, para que existam maiores possibilidades de escolha para o neurônio vencedor.

No entanto, quando isto ocorre, é aumentado substancialmente o erro topográfico, pois quando o número de neurônios aumenta, aumenta a dispersão dos vetores (pertencentes a uma mesma classe) pelos neurônios vencedores. A escolha do “melhor” mapa segue pelo bom senso, considerando estes dois fatores.

Desta forma, busca-se o “melhor” mapa, considerando os dois erros, o quantitativo, que mede a resolução, ou seja, a média da distância entre cada vetor do conjunto de dados e seu neurônio vencedor e o erro topográfico que apresenta a proporção de todos os vetores dados, para os quais, o primeiro e o segundo neurônio vencedor não são unidades adjacentes.

A convergência é garantida pelo processo de decrescimento do raio topológico e da taxa de aprendizagem e ocorre a partir do momento em que não ocorrem mais alterações significativas na matriz dos pesos. Para facilitar a convergência, os pesos são inicializados de forma aleatória, dentro do intervalo [0, 1], tendo em vista que os dados foram normalizados dentro do mesmo intervalo reduzindo todas as variáveis ao mesmo nível de agrupabilidade.

### 3.3.4 Resumo dos Modelos

Para resumir os parâmetros de modelagem das redes neurais artificiais para cada conjunto de dados, o Quadro 3.3 apresenta as principais características de cada modelo deste trabalho.

Modelo	camada de entrada	camada oculta	camada de saída
MLP	22 neurônios	20 neurônios	4 neurônios
SOM	22 neurônios	NÃO POSSUI	neurônio vencedor

Fonte: A autora.

quadro 3.2 – Resumo dos modelos parametrizados

## 3.4 Ferramentas Computacionais

Este trabalho foi realizado no ambiente computacional oogle Colaboratory — também conhecido como Google Colab — que é uma ferramenta em nuvem que permite criar e executar códigos na linguagem Python. Com ele, é possível rodar os programas diretamente do navegador, de forma simples e rápida.

A escolha por esse ambiente é devido a utilização do hardware de qualidade disponibilizado pelos servidores da Google. Mais precisamente a utilização de Unidades de Processamento Gráfico (GPUs) na hora de executar os códigos, permitindo a execução dos códigos de forma mais rápida com prioridade para uso das GPUs de melhor desempenho e com mais memória RAM para armazenar dados.

Como supracitado, a linguagem de programação utilizada foi o Python versão 3.7, uma ferramenta poderosa que possui uma coleção de algoritmos de aprendizagem de máquina para diversas tarefas de mineração de dados, além de ferramentas para processamento dos dados, classificação, regressão, agrupamentos, regras de associação e visualização de informações.

As bibliotecas utilizadas em todos os modelos foram:

- NumPy - que fornece ferramentas para cálculos matemáticos
- Pandas - que fornece ferramentas para análise e manipulação de dados.
- Matplotlib - que fornece ferramentas para geração de gráficos 2D a partir de arrays.

As funções utilizadas inerentes das bibliotecas supracitadas foram: `argmax`, `astype`, `read_csv`, `isnan`, `isnull`, `fillna`, `pyplot` e `searborn`.

Para a implementação das redes neurais artificiais, foi utilizada as bibliotecas TensorFlow, scikit-learn, Keras para TensorFlow e MiniSom.

A biblioteca TensorFlow é um conjunto de ferramentas de código aberto criada para aprendizado de máquina, computação numérica e muitas outras tarefas. Foi desenvolvido pelo Google em 2015 e rapidamente se tornou uma das principais ferramentas para machine learning e deep learning (ABADI *et al.*, 2015).

A biblioteca scikit-learn dispõe de ferramentas simples e eficientes para análise preditiva de dados, é reutilizável em diferentes situações, possui código aberto, sendo acessível a todos e foi construída sobre os pacotes NumPy, SciPy e matplotlib (PEDREGOSA *et al.*, 2011).

Keras é uma API de aprendizado profundo escrita em Python e executada na biblioteca de aprendizado de máquina TensorFlow. Foi desenvolvido com foco em possibilitar experimentação rápida. Ser capaz de ir da ideia ao resultado o mais rápido possível é a chave para fazer uma boa pesquisa (CHOLLET *et al.*, 2015).

MiniSom é uma implementação minimalista e baseada em Numpy do Self Organizing Maps (SOM). SOM é um tipo de Rede Neural Artificial capaz de converter relações estatísticas complexas e não lineares entre itens de dados de alta dimensão em relações geométricas simples em uma tela de baixa dimensão. O Minisom foi projetado para permitir que os pesquisadores construam facilmente em cima dele e tenham a compreensão rapidamente de seus detalhes (VETTIGLI, 2018).

As funções utilizadas neste trabalho das bibliotecas supracitadas foram: `sklearn.model_selection`, `train_test_split`, `sklearn.preprocessing`, `LabelEncoder`, `MinMaxScaler`, `StandardScaler`, `tensorflow.keras.layers`, `Sequential`, `Dense`, `sklearn.metrics`, `classification_report`, `confusion_matrix`, `tensorflow.keras.utils`, `plot_model`.

## Capítulo 4

---

# Resultados e Discussões

---

Para comprovar a validade dos métodos propostos e verificar a capacidade destes na predição do desempenho de estudantes, realizou-se os experimentos descritos a seguir.

### 4.1 Experimento 01 - Predição usando a rede neural MLP

Os dados que descrevem o comportamento dos estudantes referentes as disciplinas obrigatórias do 1º Período ao 3º Período de formação do curso, vetores de notas, foram submetidos ao tratamento de detecção e imputação de notas ausentes como detalhado no capítulo anterior.

O objetivo deste experimento é treinar as redes neurais para predição de desempenho de alunos nos primeiros 3 semestres letivos do curso, para que em caso de mau desempenho ainda seja possível reverter a situação dando uma atenção especial ao aluno, no apoio ao aprendizado. Para validação do treinamento, foram divididos os dados em dois conjuntos independentes: Conjunto de Treinamento, com 90% dos dados que representa 135 alunos e 2970 notas. E o Conjunto de Teste, com 10% dos dados que representa 15 alunos e 330 notas.

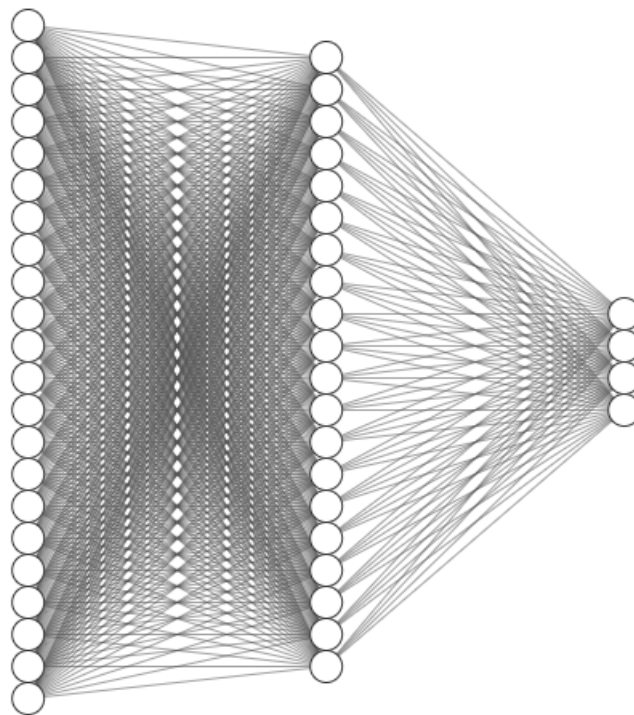
A rede neural MLP foi configurada com três camadas, camada de entrada com 22 neurônios, uma única camada oculta contendo 20 neurônios e uma camada de saída com 4 neurônios. A topologia da rede modelada está ilustrada na Figura 4.7. O algoritmo de treinamento foi o backpropagation e as funções de ativação dos neurônios do tipo ReLU e SoftMax para a camada de saída. A inicialização dos pesos nos neurônios é feita pela função 'he\_normal' que cria uma distribuição uniforme dentro de limites calculados em função do número de entradas.

A função de perda do classificador multiclasse é o cálculo da entropia cruzada que quantifica a diferença entre duas distribuições de probabilidade resultantes da saída da função de ativação SoftMax. Essa função está descrita na biblioteca do tensorflow. O modelo de otimização utilizado é o Adam, um método de gradiente descendente estocástico. De

acordo com (KINGMA; BA, 2014), o método é computacionalmente eficiente, tem pouca necessidade de memória, invariante ao reescalonamento diagonal de gradientes e é adequado para problemas que são grandes em termos de dados / parâmetros.

Cada treinamento da rede neural possui 200 épocas (ciclo de apresentação dos dados), com 32 amostras propagadas pela rede por época e tem o objetivo de atingir o erro quadrático médio de  $10^{-3}$

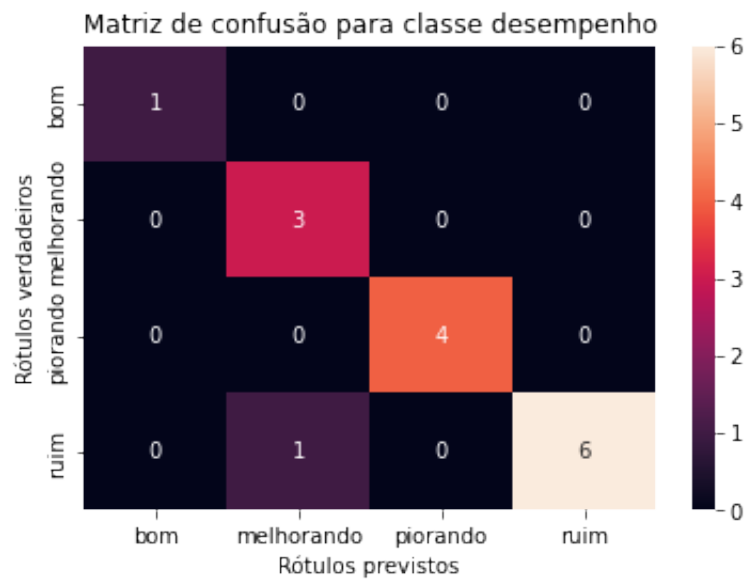
Figura 4.1 – Topologia da Rede para Experimento



Fonte: a autora

A figura 4.2 mostra a matriz de confusão da rede neural MLP na classificação dos estudantes para o experimento. Os valores acima de cada coluna são os desempenhos (classes) esperados pela classificação, e os valores ao lado de cada linha, o desempenho predito pela rede neural, e a diagonal principal da matriz de confusão apresenta as classificações corretas. Nota-se que a rede neural MLP, neste experimento, classificou corretamente 14 alunos do conjunto de dados de teste.

Figura 4.2 – Matriz de confusão da MLP no experimento



Fonte: A autora

O Quadro 4.1 mostra outras medidas do desempenho: precisão, recall e F-measure (F1-score) da MLP neste experimento. O quadro permite uma análise detalhada dos resultados da classificação por classe

	Precisão	Recall	F1 - score	Suporte
Classe 0	100%	100%	100%	1
Classe 1	93%	93%	93%	4
Classe 2	100%	100%	100%	4
Classe 3	100%	100%	100%	6
Precisão			93%	15
Desemp Médio	93%	93%	93%	15
Desemp Máx	93%	93%	93%	15

Fonte: A autora.

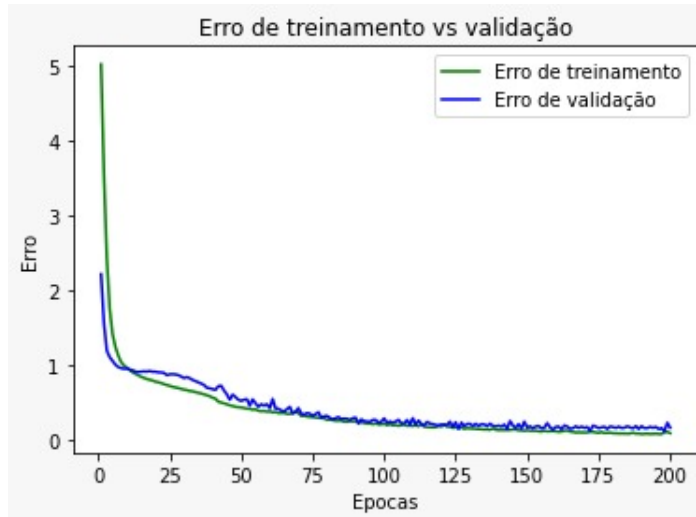
quadro 4.1 – Medidas de Precisão, Recall e F-measure da MLP no experimento 02

No experimento, a MLP classifica corretamente 93% das amostras, desta forma, 14 estudantes no total de 15 da amostra de teste foram classificados corretamente. O erro na classificação ficou em um aluno que estava com um desempenho ruim e o modelo classificou como melhorando.

Durante o treinamento, uma avaliação da rede foi feita a cada época para acompanhar e avaliar se ela está conseguindo aprender ou está apenas decorando os padrões. Para realizar essa avaliação, observa-se a evolução das perdas e a evolução da precisão, tanto no

treinamento quanto na validação, a cada época. A Figura 4.3 ilustra o gráfico da evolução de erros durante o treinamento e validação do modelo em 200 épocas.

Figura 4.3 – Erro no treinamento e validação

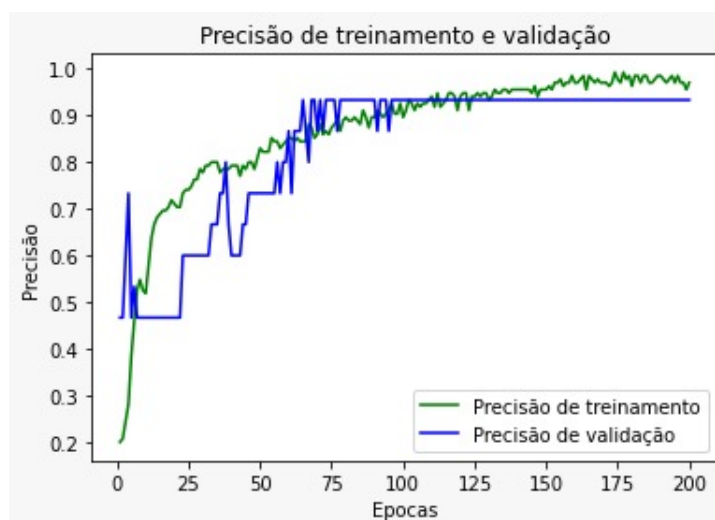


Fonte: A autora

Observa-se pela Figura 4.3 que o modelo obteve um aprendizado com poucas épocas, reduzindo a taxa de erro drasticamente nas primeiras 20 épocas e manteve a redução de erro constante. Observa-se também que o erro na validação seguiu o mesmo padrão do treinamento, o que indica que o modelo teve um ótimo desempenho de aprendizagem, mas que ajustes nas épocas devem ser feitos para garantir que não houve um falso-aprendizado da rede.

A Figura 4.3 ilustra o gráfico da evolução da precisão do modelo durante o treinamento e validação em 200 épocas.

Figura 4.4 – Precisão no treinamento e validação



Fonte: A autora

Analisando os gráficos 4.3 e 4.4 observa-se uma divergência no treinamento do modelo, onde ele alcança um ponto ótimo e começa piorar ao longo das épocas. Um outro ponto observado é a existência de uma discrepância significativa entre os dados de treinamento e validação, evidenciando que está ocorrendo overfitting no modelo.

Um cenário de overfitting ocorre quando, nos dados de treino, o modelo tem um desempenho excelente, porém quando aplicado os dados de teste o resultado não tem a mesma excelência. Entende-se que, neste caso, o modelo aprendeu tão bem as relações existentes no treino, que acabou apenas decorando o que deveria ser feito, e ao receber as informações das variáveis preditoras nos dados de teste, o modelo tenta aplicar as mesmas regras decoradas, porém com dados diferentes esta regra não tem validade, e o desempenho é afetado. Neste caso, o modelo treinado pode não ter capacidade de generalização.

Para tentar solucionar este problema de overfitting, testa-se duas hipóteses: aumentar a quantidade de épocas pode elevar a excelência do modelo quando exposto a novos dados, ou ajustar o tamanho do conjunto de treinamento, seja diminuindo ou aumentando.

Testando-se a primeira hipótese, o aumento das épocas não solucionou o problema nem elevou a precisão do modelo.

Para testar a segunda hipótese, testou-se as seguintes configurações para o tamanho de dados:

- a) 60% Treinamento - 40% Validação
- b) 70% Treinamento - 30% Validação
- c) 80% Treinamento - 20% Validação
- d) 95% Treinamento - 05% Validação

Para realizar as comparações, a Figura 4.9 ilustra os resultados do modelo diante das configurações supracitadas.

Analisando os resultados ilustrados na Figura 4.9, tem-se que ajustando o conjunto de dados para 95% treinamento e 05% para validação o cenário de overfitting é eliminado. Logo, a melhor configuração para o modelo de MLP é utilizando as notas de 142 alunos para treinamento da rede e as notas de 8 alunos para validação, de acordo com o total de apenas 150 alunos disponibilizados para este trabalho.

Com o modelo ajustado, tem-se a Matriz de confusão para análise da eficiência do modelo, ilustrada na Figura 4.10.

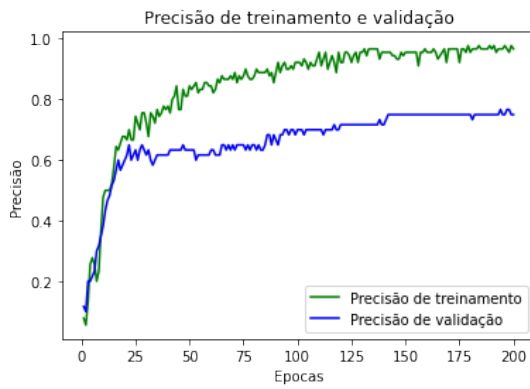


Figura 4.5 – Configuração a

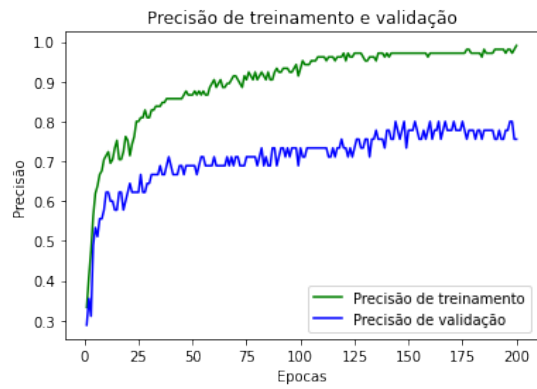


Figura 4.6 – Configuração b

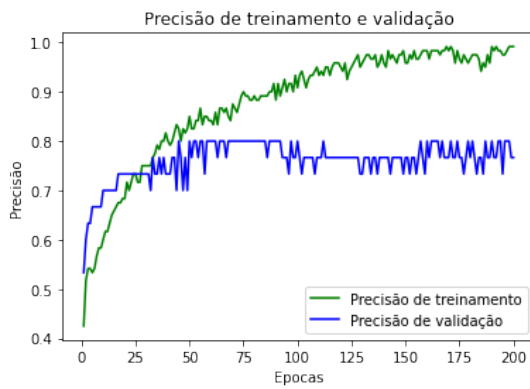


Figura 4.7 – Configuração c

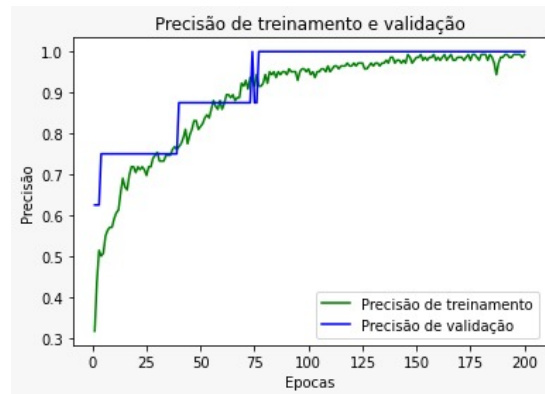
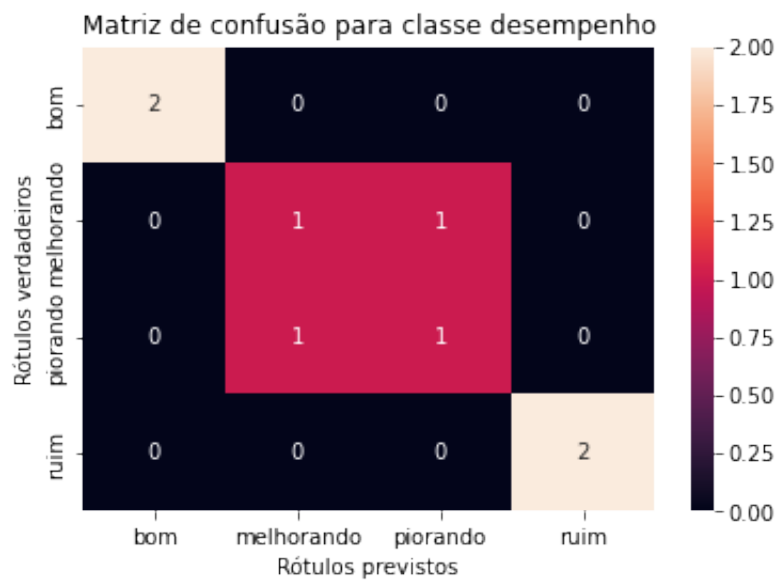


Figura 4.8 – Configuração d

Figura 4.9 – Resultados do modelo perante as Configurações do Conjunto de Dados

Figura 4.10 – Matriz de confusão da MLP no experimento ajustado



Fonte: A autora

O Quadro 4.2 mostra as novas medidas do desempenho: precisão, recall e F-measure

(F1-score) da MLP neste novo modelo.

	Precisão	Recall	F1 - score	Suporte
Classe 0	100%	100%	100%	2
Classe 1	98%	98%	98%	2
Classe 2	98%	98%	98%	2
Classe 3	100%	100%	100%	2
Precisão			98%	8
Desemp Médio	98%	98%	98%	8
Desemp Máx	98%	98%	98%	8

Fonte: A autora.

quadro 4.2 – Medidas de Precisão, Recall e F-measure da MLP no experimento 02

Agora, o erro na classificação ficou em um aluno que estava com um desempenho melhorando e o modelo classificou como piorando e outro aluno que estava piorando e o modelo classificou como melhorando.

Mesmo com estes erros, o modelo ajustado obteve uma melhor precisão do que o modelo anterior, e o problema de overfitting foi solucionado, o que trás maior confiabilidade para a utilização.

## 4.2 Experimento 02 - Kohonen

Os Mapas auto-organizáveis de Kohonen empregam aprendizagem não supervisionada. A rede é alimentada com padrões/registros para que seja feita a combinação daqueles registros que possuem propriedades semelhantes aos nós na grade SOM que permanecerão como clusters (neurônios parecido).

Os registros que possuem características/propriedades incomuns ou diferentes serão combinados com nós na grade que permanecerão como outliers (neurônios diferentes). O objeto da classe minisom é criado e inicializado com pesos aleatórios usando a função `som.random_weights_init()`.

Neste experimento, a rede neural SOM foi configurada com 30x30 de dimensões (não deve ser pequeno para que os outliers possam ser detectados). O número total de classes/nós máximos presentes na grade SOM, nos quais os padrões/registros de entrada do conjunto de dados podem ser classificados, são de 900 nós.

A camada de entrada da rede foi configurada com 22 neurônios, número de disciplinas cursadas do 1º Período ao 3º Período. A taxa de aprendizado da rede foi configurada em 0.05. O raio  $\sigma$  foi configurado com 1 nó com a função de atualização gaussiana, ou seja, a cada época o nó vencedor tem seu pesos atualizados e os nós vizinhos no raio gaussi-

ano de 1 neurônio também sofrem atualizações nos pesos. Os pesos iniciais foram gerados pela função `random_seed`, que gera pesos pseudo-aleatórios para cada neurônio. A função `som.train_random()` treina o SOM para os padrões de entrada fornecidos e isso é feito para 500 iterações.

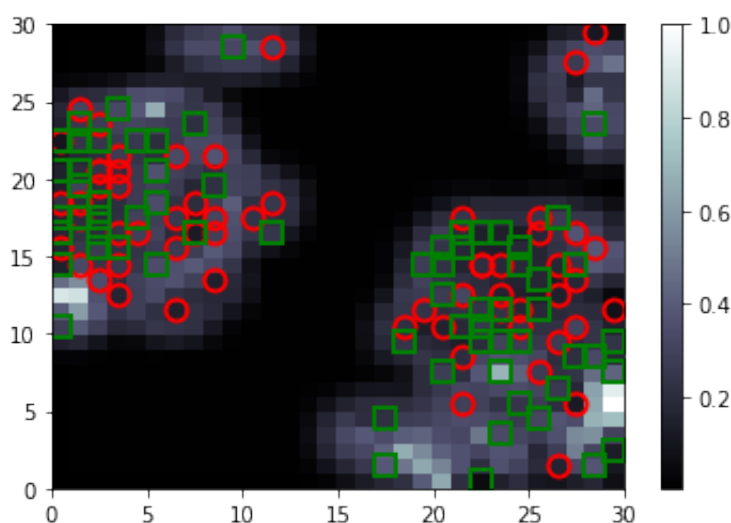
A função `som.train_random()` treina o SOM para os padrões de entrada fornecidos e isso é feito para 500 iterações. Aleatoriamente, um registro é selecionado a partir dos padrões dados e é então alimentado para a rede. O nó vencedor (o nó cuja distância euclidiana para o padrão de entrada fornecido é mínimo) é a classe prevista para o padrão / registro de entrada dado.

Então, a mudança em pesos (ou características) do nó vencedor no SOM é feita de forma que a distância euclidiana entre o nó vencedor e o padrão de entrada diminua. Além disso, os pesos dos vizinhos dos nós vencedores também são atualizados no mesmo sentido para diminuir a distância euclidiana de modo que os pesos dos nós vizinhos que estão mais próximos do nó vencedor sejam atualizados mais do que aqueles que estão longe do nó vencedor.

Após o processo de treinamento, todos os padrões / registros de entrada do conjunto de dados são combinados com os nós da grade SOM. Os nós da grade com propriedades semelhantes ficarão mais próximos e aqueles com propriedades diferentes ficarão a uma distância maior para outros nós (esses nós serão outliers na grade SOM).

Portanto, o modelo poderá determinar o desempenho do aluno descobrindo em qual padrão suas notas mais se assemelharão. A Figura 4.11 mostra o mapa onde estão agrupados os nós de cada classe e a distância dos nós que estão mais distantes dos nós da outra classe.

Figura 4.11 – Mapa de distribuição de distância entre os nós da rede



Fonte: A autora

As caixas brancas no mapa de distância na Figura 4.11 correspondem aos nós para os quais a distância média entre neurônios é grande. Preto representa a distância média entre neurônios é mínima e as manchas brancas representam a distância média máxima entre neurônios. Para cada padrão no conjunto de dados, o neurônio vencedor correspondente foi marcado (verde - ganhador, vermelho - perdedores).

A figura 4.12 mostra a matriz de confusão da rede neural SOM na classificação dos estudantes para o experimento 02. Os valores acima de cada coluna são os desempenhos (classes) esperados pela classificação, e os valores ao lado de cada linha, o desempenho previsto pela rede neural, e a diagonal principal da matriz de confusão apresenta as classificações corretas. Nota-se também que a rede neural SOM, neste experimento, não classificou corretamente apenas 2 alunos, por outro lado, 13 alunos do conjunto de dados de teste foram classificados corretamente.

Figura 4.12 – Matriz de confusão da SOM no experimento



Fonte: A autora

O Quadro 4.3 mostra outras medidas do desempenho: precisão, recall e F-measure (F1-score) da SOM neste experimento. O quadro permite uma análise detalhada do resultados da classificação por classe.

	Precisão	Recall	F1 - score	Suporte
Classe 0	0.00%	0.00%	0.00%	1
Classe 1	33%	80%	50%	5
Classe 2	50%	33%	40%	3
Classe 3	100%	100%	100%	6
Precisão			75%	15
Desemp Médio	75%	75%	85%	15
Desemp Máx	89%	75%	85%	15

Fonte: A autora.

quadro 4.3 – Medidas de Precisão, Recall e F-measure da MLP no experimento 02

No experimento, o SOM classifica corretamente 75% das amostras, desta forma, 6 estudantes no total de 15 da amostra de teste foram classificados corretamente. Os resultados podem variar a cada execução do modelo porque a inicialização dos pesos dos nós da grade SOM é feita selecionando-se aleatoriamente os registros / padrões do espaço de entrada, ou seja, selecionando aleatoriamente os registros do conjunto de dados fornecido.

Analisando a Figura 4.12, percebe-se que o modelo não conseguiu classificar 7 alunos e nenhuma das classes, além de ter uma precisão baixa para o problema em questão, no qual precisa-se de uma assertividade maior.

Para tentar solucionar este cenário, testa-se duas hipóteses: a primeira que é aumentar as iterações do modelo fará com que ele aumente seu aprendizado e a segunda que é ajustar a dimensão do mapa de forma que todas características sejam extraídas e agrupadas corretamente.

Iniciando pela segunda hipótese, realizou-se as seguintes configurações para o mapa de Kohonen, considerando que o mapa não pode ter uma dimensão muito pequena nem muito extrapolada, pois pode haver perda de informação, sendo considerado como limite um mapa com até o dobro da dimensão inicial:

- a) Mapa 10x10
- b) Mapa 15x15
- c) Mapa 20x20
- d) Mapa 60x60

Como resultado, foi gerado a matriz de confusão de cada configuração do mapa. A Figura 4.17 ilustra as matrizes de confusão.

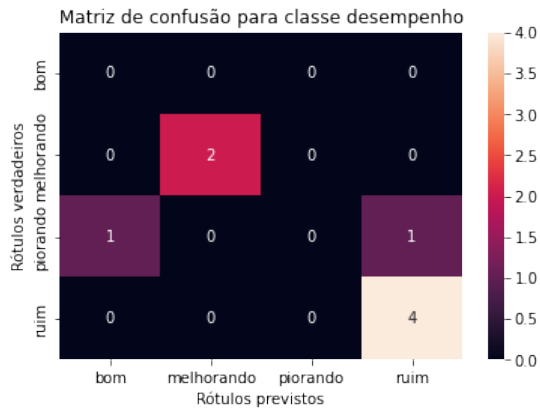


Figura 4.13 – Configuração a - 75% precisão

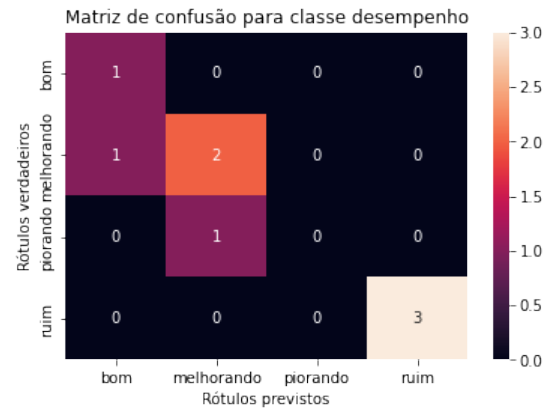


Figura 4.14 – Configuração b - 75% precisão

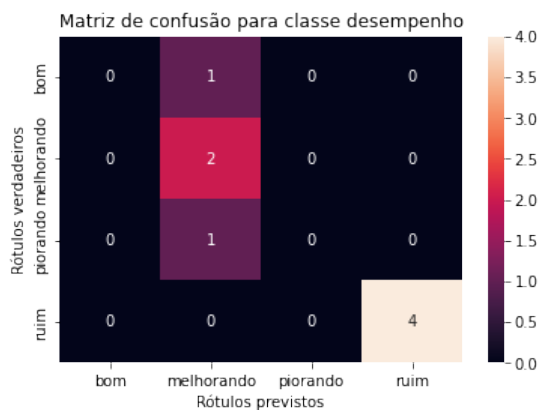


Figura 4.15 – Configuração c - 75% precisão



Figura 4.16 – Configuração d - 88% precisão

Figura 4.17 – Resultados do modelo perante as Configurações do Conjunto de Dados

Analisando os resultados ilustrados na Figura 4.17, tem-se que a melhor configuração para o modelo de SOM é utilizando um mapa de 60x60 de dimensão, o que gera 3.600 nós na rede.

Para validar essa hipótese alguns critérios foram propostos para quantificar estas propriedades. O erro de quantização é simples de ser avaliado, sendo computado como a distância média das amostras do conjunto de dados  $X$  aos neurônios mais próximos. Para quantificar a continuidade vários critérios foram propostos na literatura.

O mais simples, o Erro Topográfico,  $t$ , é obtido considerando a adjacência dos campos receptivos e a proporção dos objetos que indicam a descontinuidade local do mapeamento. O erro topográfico para o mapeamento é obtido somando-se todos os erros topográficos locais para todos os padrões.

Mapa	Erro topográfico	% Erro quantizado
a	0.0491	2.294
b	0.07	2.201
c	0.154	2.239
d	0.091	2.203

Fonte: A autora.

quadro 4.4 – Erro topográfico e quantizado na SOM

Porém, verifica-se que mesmo aumentando a precisão em relação ao modelo anterior, a rede neural não conseguiu classificar uma das classes do conjunto de dados de teste, o que reafirma que os poucos dados para este trabalho limita seu desempenho.

Testando a primeira hipótese e considerando o mapa configurado em 60x60, ajusta-se as iterações do treinamento de 500 iterações para 1000 iterações e 2000 iterações. Por se tratar de um pequeno conjunto de dados, aumentar muito as iterações não surtirá efeito nenhum no desempenho.

O Quadro mostra a precisão da rede neural de acordo com o ajuste das iterações do aprendizado.

Iterações	Precisão
500	75%
1000	80%
2000	100%

Fonte: A autora.

quadro 4.5 – Precisão do modelo diante evolução de iterações

Analisando os resultados ilustrados na Figura 4.16d e o Quadro 4.5, tem-se que a melhor configuração para o modelo de SOM é utilizando um mapa com 60x60 de dimensão e 2000 iterações na etapa de aprendizado, de acordo com o total de apenas 150 alunos disponibilizados para este trabalho.

Com o modelo ajustado, tem-se a Matriz de confusão para análise da eficácia do modelo, ilustrada na Figura 4.18.

Figura 4.18 – Matriz de confusão da MLP no experimento ajustado



Por fim, foi possível chegar no melhor modelo de ambas arquiteturas de redes neurais, MLP e SOM, que atenda a necessidade e disponibilidade do conjunto de dados.

É possível confiar em um dos modelos com 98% e 100% de precisão para realizar o acompanhamento de alunos para que seja possível avaliar seu desempenho.

## Capítulo 5

---

# Conclusões

---

Este trabalho reafirma que as técnicas de Mineração de Dados, quando aplicada corretamente, podem trazer informações úteis para o acompanhamento de estudantes e melhorias dos processos de ensino-aprendizagem.

Uma abordagem baseada na imputação estatística dos dados ausentes, objetivou a substituição de informações ausentes nos vetores de características, de forma que o valor ausente assumia o valor da nota de maior frequência na coluna.

Esta etapa de transformação e preparação dos dados dos estudantes foi fundamental para os processos de classificação com redes neurais de tipo Multilayer Perceptron e Mapas Auto-Organizáveis de Kohonen, que se mostraram métodos eficientes para predição do desempenho acadêmico a partir de informações sobre as notas obtidas pelos alunos nas disciplinas obrigatórias nos primeiros semestres letivos.

No modelo de MLP foi detectada a existência de uma discrepância significativa entre os dados de treinamento e validação no modelo que utiliza 90% e 10% respectivamente, evidenciando a ocorrência de overfitting, que impede que o modelo consiga generalizar as classificações para todos os dados de entrada.

Como solução para este cenário, foi reajustado o tamanho do conjunto de treinamento com 95% do conjunto total e conseqüentemente 5% para validação, este ajuste foi necessário devido ao pequeno volume de dados, mas que são inerentes ao problema.

No modelo de SOM inicial sua precisão foi de 75%, o que não é confiável. Como solução, foi realizado dois ajustes no modelo: na dimensão do mapa e nas iterações de aprendizado. Com isso o melhor modelo, o que obteve precisão máxima de 100% foi com o mapa de 60X60 com 2.000 iterações.

Contudo, ambos os modelos desenvolvidos trazem confiabilidade na identificação dos estudantes em quatro níveis de desempenhos. A precisão da MLP foi de 98% enquanto a SOM obteve 100%.

Ambos os métodos de aprendizagem supervisionado (MLP) e não-supervisionado (SOM) foram capazes de realizar as classificações para o conjunto de dados do curso de Bacharelado em Matemática Industrial, o que pode ser replicado em outros cursos que possuem a mesma quantidade de disciplinas.

Por fim, neste trabalho foi possível desenvolver uma ferramenta de Machine Learning capaz de auxiliar na identificação do rendimento dos estudantes, para que os gestores possam desenvolver tomadas de decisões acerca dos resultados obtidos.

---

## REFERÊNCIAS

---

- ABADI, M. *et al.* *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. 2015. Software available from tensorflow.org. Disponível em: <<https://www.tensorflow.org/>>. Citado na página 65.
- AKKO, S. Uma comparação empírica de técnicas convencionais, redes neurais e o modelo híbrido adaptive neuro fuzzy inference system (anfis) para análise de pontuação de crédito: o caso dos dados de cartão de crédito da turquia. v. 222, 2012. Citado na página 27.
- ARAÚJO, E. A. T. *et al.* Desempenho acadêmico de discentes do curso de ciências contábeis: Uma análise dos seus fatores determinantes em uma ies privada. *Contabilidade Vista & Revista*, v. 24, n. 1, p. 60–83, 2013. Citado na página 14.
- AYERS, E.; NUGENT, R.; DEAN, N. A comparison of student skill knowledge estimates. *International Working Group on Educational Data Mining*, ERIC, 2009. Citado na página 22.
- BAKER, R.; ISOTANI, S.; CARVALHO, A. Mineração de dados educacionais: Oportunidades para o brasil. *Revista Brasileira de Informática na Educação*, v. 19, n. 02, p. 03, 2011. Citado 3 vezes nas páginas 14, 17 e 18.
- BAKER, R. S.; YACEF, K. *et al.* The state of educational data mining in 2009: A review and future visions. *Journal of educational data mining*, v. 1, n. 1, p. 3–17, 2009. Citado 3 vezes nas páginas 15, 17 e 22.
- BARCELLOS, C. D. *et al.* Sistema de recomendação acadêmico para apoio a aprendizagem. *RENOTE*, v. 5, n. 2, 2007. Citado na página 22.
- BECKER, S. Procedimentos de aprendizagem não supervisionados para redes neurais. v. 2, 1991. Citado na página 42.
- BINOTI, M. L. M. d. S. *et al.* Emprego de redes neurais artificiais em mensuração e manejo florestal. Universidade Federal de Viçosa, 2012. Citado 2 vezes nas páginas 38 e 39.
- BISHOP, C. M. e. o. *Redes neurais para reconhecimento de padrões*. [S.l.: s.n.], 1995. Citado na página 40.
- BOSCARIOLI, C. *Análise de agrupamentos baseada na topologia dos dados e em mapas auto-organizáveis*. Tese (Doutorado) — Universidade de São Paulo, 2008. Citado na página 63.
- BRAGA, A. d. P.; CARVALHO, A.; LUDERMIR, T. *Redes neurais artificiais: teoria e prática*. Editora LTC, 2000. Citado 2 vezes nas páginas 27 e 43.

BREIMAN LEO E FRIEDMAN, J. e. S. C. J. e. O. R. A. *Árvores de classificação e regressão*. [S.l.: s.n.], 1984. Citado na página 24.

BRESFELEAN, V. P. *et al.* Determining students' academic failure profile founded on data mining methods. In: IEEE. *ITI 2008-30th International Conference on Information Technology Interfaces*. [S.l.], 2008. p. 317–322. Citado na página 22.

CARVALHO, A.; BRAGA, A.; LUDERMIR, T. Fundamentos de redes neurais. *Publicação da COPPE/UF RJ para a XI Escola de Computação Brasileira, Rio de Janeiro*, 1998. Citado 2 vezes nas páginas 43 e 46.

CASTRO, F. D.; CASTRO, M. D. Redes neurais artificiais. *Porto Alegre, RS: Pontifícia Universidade Católica do Rio Grande do Sul*, 2001. Citado na página 47.

CERQUEIRA, T. C. S. *et al.* Estilos de aprendizagem em universitários. [sn], 2000. Citado na página 14.

CHOLLET, F. *et al.* Keras. GitHub, 2015. Disponível em: <<https://github.com/fchollet/keras>>. Citado na página 65.

CIOS, K. J. *et al.* *Data mining: a knowledge discovery approach*. [S.l.]: Springer Science & Business Media, 2007. Citado na página 24.

CORBUCCI, P. R. Desafios da educação superior e desenvolvimento no brasil. Instituto de Pesquisa Econômica Aplicada (IPEA), 2007. Citado na página 14.

CYBENKO, G. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, Springer, v. 2, n. 4, p. 303–314, 1989. Citado 2 vezes nas páginas 43 e 61.

DEKKER, G. W.; PECHENIZKIY, M.; VLEESHOUWERS, J. M. Predicting students drop out: A case study. *International Working Group on Educational Data Mining*, ERIC, 2009. Citado na página 22.

DIAMANTOPOULOU, M. J. Estimativa de volume de tronco de árvore em pinheiros em pé usando modelos de rede neural artificial de correlação em cascata. 2006. Citado na página 45.

DUDA, R. O.; HART, P. E. *et al.* *Pattern classification*. [S.l.]: John Wiley & Sons, 2006. Citado na página 53.

EYNG, E. *et al.* Controle feedforward-feedback aplicado as colunas de absorção do processo de produção de etanol por fermentação. [sn], 2008. Citado na página 40.

FAYYAD, U.; PIATETSKY-SHAPIRO, G.; SMYTH, P. From data mining to knowledge discovery in databases. *AI magazine*, v. 17, n. 3, p. 37–37, 1996. Citado na página 14.

FUNAHASHI, K.-I. On the approximate realization of continuous mappings by neural networks. *Neural networks*, Elsevier, v. 2, n. 3, p. 183–192, 1989. Citado na página 43.

GARCÍA, E. *et al.* A collaborative educational association rule mining tool. *The Internet and Higher Education*, Elsevier, v. 14, n. 2, p. 77–88, 2011. Citado 2 vezes nas páginas 17 e 23.

GORGENS, E. B. *et al.* Estimacão do volume de árvores utilizando redes neurais artificiais. *Revista Árvore*, SciELO Brasil, v. 33, n. 6, p. 1141–1147, 2009. Citado na página 39.

GRIBOSKI, C. M. O enade como indutor da qualidade da educação superior. *Estudos em avaliação educacional*, v. 23, n. 53, p. 178–195, 2012. Citado na página 14.

HAN, J.; PEI, J.; KAMBER, M. *Data mining: concepts and techniques*. [S.l.]: Elsevier, 2011. Citado na página 42.

HAN JIAWEI E KAMBER, M. *Data mining: Concepts and techniques*, the morgan kaufmann series in data management systems, jim gray, editor da série morgan kaufmann publishers. 2.000. Citado na página 24.

HASTIE TREVOR E TIBSHIRANI, R. e. F. J. *Os elementos de aprendizagem estatística: mineração de dados, inferência e previsão*. [S.l.: s.n.], 2009. Citado na página 45.

HAYKIN, S. *Redes neurais: princípios e prática*, 2ª edição, tradução: Paulo martins engel. Editora: Bookman, Porto Alegre, Cap, v. 1, n. 2, p. 3, 2001. Citado 15 vezes nas páginas 29, 30, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43 e 45.

HEBB, D. O. *The organization of behavior: A neuropsychological theory*. [S.l.]: Psychology Press, 2005. Citado na página 31.

HEIJDEN, G. J. Van der *et al.* Imputation of missing values is superior to complete case analysis and the missing-indicator method in multivariable diagnostic research: a clinical example. *Journal of clinical epidemiology*, Elsevier, v. 59, n. 10, p. 1102–1109, 2006. Citado na página 26.

HERSHKOVITZ, A.; NACHMIAS, R. Developing a log-based motivation measuring tool. In: CITESEER. *EDM*. [S.l.], 2008. p. 226–233. Citado na página 22.

HERTZ, J.; KROGH, A.; PALMER, R. G. *Introduction to the theory of neural computation*. [S.l.]: CRC Press, 2018. Citado na página 43.

HORNIK KURT E STINCHCOMBE, M.; WHITE, H. Multilayer feedforward networks are universal approximators. v. 2, 1989. Citado 2 vezes nas páginas 43 e 61.

JOHANSSON PETER E H "A KINEN, J. Improving missing value imputation of microarray data using spot quality weight. v. 7, 2006. Citado na página 24.

JR, F. E. H. *Regression modeling strategies: with applications to linear models, logistic and ordinal regression, and survival analysis*. [S.l.]: springer, 2015. Citado na página 26.

representação neural e computação neural. v. 4. Citado na página 36.

KIM KI-YEOL E KIM, B.-J. e. Y. G.-S. *Reutilização de dados imputados na análise de microarray aumenta a eficiência de imputação*. 2004. Citado na página 24.

KINGMA, D. P.; BA, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. Citado na página 68.

KOHONEN, T. *Construction of similarity diagrams for phonemes by a self-organizing algorithm*. [S.l.]: Teknillinen korkeakoulu, 1981. Citado 5 vezes nas páginas 46, 48, 50, 51 e 53.

\_\_\_\_\_. Learning vector quantization. In: *Self-organizing maps*. [S.l.]: Springer, 2001. p. 245–261. Citado 4 vezes nas páginas 47, 48, 50 e 62.

KOHONEN TEUVO E KASKI, S. e. L. H. formação auto-organizada de vários filtros de características invariantes no subespaço adaptativo som. v. 9, 1997. Citado 3 vezes nas páginas 42, 47 e 52.

KOSE UTKU E ARSLAN, A. Otimização de autoaprendizagem em cursos de engenharia da computação: um sistema de software inteligente suportado por rede neural artificial e algoritmo de otimização de vórtice. *Computer Applications in Engineering Education*, v. 25, 2017. Citado 2 vezes nas páginas 28 e 29.

LAU ET E SUN, L. e. Y. Q. Modelagem, previsão e classificação do desempenho acadêmico do aluno usando redes neurais artificiais. v. 1, 2019. Citado na página 29.

LITTLE, R. J. *ajustes de dados ausentes em pesquisas grandes*. 1988. Citado 2 vezes nas páginas 24 e 25.

MANDERICK, B. Ais in context with other computational intelligence paradigms. Citado na página 39.

MANHÃES, L. M. B. Predição do desempenho acadêmico de graduandos utilizando mineração de dados educacionais. *Doutorado em Engenharia de Sistemas e Computação Instituição de Ensino: Universidade Federal do Rio de Janeiro, Rio de Janeiro. Biblioteca Depositária: BIBLIOTECA DO CT*, 2015. Citado na página 23.

MANHÃES, L. M. B. *et al.* Previsão de estudantes com risco de evasão utilizando técnicas de mineração de dados. In: *Brazilian symposium on computers in education (simpósio brasileiro de informática na educação-sbie)*. [S.l.: s.n.], 2012. v. 1, n. 1. Citado na página 22.

MCCALLUM, Q. E. *Manual de dados incorreto: limpando os dados para que você possa voltar ao trabalho*. [S.l.: s.n.], 2012. Citado na página 25.

MCCARTHY JOHN E MINSKY, M. L. e. R. N. e. S. C. E. Uma proposta para o projeto de pesquisa de verão de dartmouth em inteligência artificial, 31 de agosto de 1955. v. 27, 2006. Citado na página 31.

MCCULLOCH WARREN S E PITTS, W. Um cálculo lógico das idéias imanentes na atividade nervosa. v. 5, 1943. Citado 2 vezes nas páginas 31 e 32.

MIRANDA, F. A.; FREITAS, S. R. C. D.; FAGGION, P. L. Integração e interpolação de dados de anomalias ar livre utilizando-se a técnica de rna e krigagem. *Boletim de Ciências Geodésicas*, v. 15, n. 3, 2009. Citado na página 40.

MITCHELL, T. M. *et al.* Machine learning. McGraw-hill New York, 1997. Citado na página 23.

MUNHOZ, A. M. H. *Uma análise multidisciplinar da relação entre inteligência e desempenho acadêmico em universitários ingressantes*. Tese (Doutorado) — Tese de Doutorado, Universidade Estadual de Campinas, Campinas, SP, Brasil, 2004. Citado na página 13.

NUNES, R. T. M. Cálculo preditivo de classificação das notas do enade utilizando redes neurais artificiais. *Revista de Tecnologia Aplicada*, v. 7, n. 2, 2018. Citado 2 vezes nas páginas 28 e 29.

PANDORFI, H. *et al.* Uso de redes neurais artificiais para predição de índices zootécnicos nas fases de gestação e maternidade na suinocultura. *Revista Brasileira de Zootecnia, SciELO Brasil*, v. 40, n. 3, p. 676–681, 2011. Citado na página 45.

- PATTERSON, D. W. *Redes neurais artificiais: teoria e aplicações*. [S.l.: s.n.], 1998. Citado 2 vezes nas páginas 40 e 46.
- PEDREGOSA, F. *et al.* Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, v. 12, p. 2825–2830, 2011. Citado na página 65.
- PIPINO LEO L E LEE, Y. W. e. W. R. Y. v. 45, 2002. Citado na página 24.
- POLIDORI, M. M.; MARINHO-ARAUJO, C. M.; BARREYRO, G. B. *SINAES: perspectivas e desafios na avaliação da educação superior brasileira*. [S.l.]: SciELO Brasil, 2006. Citado na página 13.
- PYLE, D. *Data preparation for data mining*. [S.l.]: morgan kaufmann, 1999. Citado na página 24.
- QUINLAN, J. R. Indução de árvores de decisão. In: *Leituras na aquisição e aprendizagem de conhecimento: Automatizando a construção e melhoria de sistemas especialistas*. [S.l.: s.n.], 1993. Citado na página 24.
- RASHID TARIK A E AHMAD, H. A. Sistema de desempenho de palestrante usando rede neural com otimização de enxame de partículas. *Computer Applications in Engineering Education*, v. 24, 2016. Citado 2 vezes nas páginas 28 e 29.
- RAUBER, T. W. *Redes neurais artificiais*. Universidade Federal do Espírito Santo, p. 29, 2005. Citado na página 27.
- RESEARCH, U. of Illinois at Urbana-Champaign. Center for S.; DEVELOPMENT; CYBENKO, G. *Continuous valued neural networks with two hidden layers are sufficient*. [S.l.: s.n.], 1988. Citado na página 43.
- REZENDE, S. O. *Sistemas inteligentes: fundamentos e aplicações*. [S.l.]: Editora Manole Ltda, 2003. Citado na página 39.
- RIGO, S. J.; CAZELLA, S. C.; CAMBRUZZI, W. Minerando dados educacionais com foco na evasão escolar: oportunidades, desafios e necessidades. In: *Anais do Workshop de Desafios da Computação Aplicada à Educação*. [S.l.: s.n.], 2012. p. 168–177. Citado na página 22.
- RISTOFF, D.; GIOLO, J. O sinaes como sistema. *Revista Brasileira de Pós-Graduação*, v. 3, n. 6, 2006. Citado na página 13.
- ROMERO, C.; VENTURA, S. Data mining in education. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, Wiley Online Library, v. 3, n. 1, p. 12–27, 2013. Citado 3 vezes nas páginas 19, 20 e 22.
- ROMERO, C. *et al.* *Handbook of educational data mining*. [S.l.]: CRC press, 2010. Citado 2 vezes nas páginas 19 e 21.
- ROSENBLATT, F. Perceptron simulation experiments. *Proceedings of the IRE, IEEE*, v. 48, n. 3, p. 301–309, 1960. Citado na página 31.
- RUBIN, D. B. *Multiple imputation for survey nonresponse*. [S.l.]: New York: Wiley, 1987. Citado na página 26.
- SANTOS, H. L. d. Minerando dados de ambientes virtuais de aprendizagem para predição de desempenho de estudantes. 2012. Citado na página 22.

- SANTOS, N. d. A. *Determinantes do desempenho acadêmico dos alunos dos cursos de ciências contábeis*. Tese (Doutorado) — Universidade de São Paulo, 2012. Citado na página 14.
- SCHAFFER, J. L. *Analysis of incomplete multivariate data*. [S.l.]: CRC press, 1997. Citado 2 vezes nas páginas 24 e 26.
- SILVA, I. N. D.; SPATTI, D. H.; FLAUZINO, R. A. *Redes neurais artificiais para engenharia e ciências aplicadas curso prático*. São Paulo: Artliber, 2010. Citado na página 36.
- TANG, L. *et al.* A comparison of imputation methods in a longitudinal randomized clinical trial. *Statistics in medicine*, Wiley Online Library, v. 24, n. 14, p. 2111–2128, 2005. Citado na página 26.
- VALENÇA, M. *Fundamentos das redes neurais: exemplos em java*. Olinda, Pernambuco: Editora Livro Rápido, 2010. Citado 2 vezes nas páginas 39 e 45.
- VALENÇA, M. J. *Aplicando redes neurais: um guia completo*. Livro Rápido, Olinda, 2005. Citado na página 27.
- VANDAMME JEAN-PHILIPPE E MESKENS, N. e. S. J.-F. e. o. Previsão do desempenho acadêmico por métodos de mineração de dados. v. 15, 2007. Citado na página 28.
- VETTIGLI, G. *MiniSom: minimalistic and NumPy-based implementation of the Self Organizing Map*. 2018. Disponível em: <<https://github.com/JustGlowing/minisom/>>. Citado na página 65.
- VILLMANN, T.; BAUER, H.-U. Applications of the growing self-organizing map. *Neurocomputing*, Elsevier, v. 21, n. 1-3, p. 91–100, 1998. Citado 6 vezes nas páginas 47, 48, 49, 50, 51 e 52.
- WIDROW BERNARD E LEHR, M. A. 30 anos de redes neurais adaptativas: perceptron, madaline e backpropagation. v. 78, 1990. Citado na página 44.
- WITTEN, I. H. *Data mining*. Springer, 2016. Citado na página 25.
- WITTEN IAN H E FRANK, E. *Data mining: ferramentas e técnicas práticas de aprendizado de máquina com implementações Java*. 2002. Citado na página 22.
- ZANETTI, S. S. *et al.* Estimção da evapotranspiração de referência no estado do rio de janeiro usando redes neurais artificiais. *Revista Brasileira de Engenharia Agrícola e Ambiental*, SciELO Brasil, v. 12, n. 2, p. 174–180, 2008. Citado na página 28.
- ZHANG, P. Multiple imputation: theory and method. *International Statistical Review/Revue Internationale de Statistique*, JSTOR, p. 581–592, 2003. Citado na página 26.
- ZHOU, X.-H.; ECKERT, G. J.; TIERNEY, W. M. Multiple imputation in public health research. *Statistics in medicine*, Wiley Online Library, v. 20, n. 9-10, p. 1541–1549, 2001. Citado na página 26.



UNIVERSIDADE FEDERAL DE GOIÁS  
UNIDADE ACADÊMICA ESPECIAL DE MATEMÁTICA E TECNOLOGIA

## SOLICITAÇÃO

Prezado Pró-Reitor de Graduação Pró-Tempore da UFCAT

Professor Moisés Fernandes Lemos

Catalão-GO

Venho através desta solicitar a V. Sa. a utilização de dados específicos (notas e frequências de todos estudantes) sobre o curso de graduação em Matemática Industrial. Uma estudante de mestrado sob minha orientação no Programa de Pós Graduação em Modelagem e Otimização utilizaria o conjunto de dados sobre notas dos estudantes ao longo da história do curso na tentativa de elaboração de um modelo preditivo baseado em redes neurais artificiais para nos auxiliar na identificação de estudantes com maior possibilidade de desistência. Assim, busca-se através do conjunto de dados disponíveis por nós ao longo da história do curso, utiliza-lo de forma a aumentar a capacidade da coordenação do curso, professores do Núcleo Docente Estruturante e Chefia da Unidade em apoio e tomada de decisões frente a políticas de evasão escolar. Ainda, ressalto que os dados jamais serão tratados de forma individualizada e identificada, mas apenas utilizada como o volume de dados no treinamento, calibração e testes dos modelos. Durante a aplicação, os nomes dos alunos e seus números de matrícula não serão disponibilizados, tendo apenas para manipulação, o volume com dados (notas e frequências) de aluno 1, aluno 2, assim por diante.

Agradeço antecipadamente a compreensão e suporte nesta demanda.

Respeitosamente,



Documento assinado eletronicamente por **Jose Dos Reis Vieira De Moura Junior, Professor do Magistério Superior**, em 19/08/2021, às 14:30, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



A autenticidade deste documento pode ser conferida no site [https://sei.ufg.br/sei/controlador\\_externo.php?acao=documento\\_conferir&id\\_orgao\\_acesso\\_externo=0](https://sei.ufg.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0), informando o código verificador **2290705** e o código CRC **1EA81287**.



UNIVERSIDADE FEDERAL DE GOIÁS  
COORDENAÇÃO DE GRADUAÇÃO - REGIONAL CATALÃO

### DESPACHO

Ao

Prof. Dr. José dos Reis Vieira de Moura Junior

IMTec - UFCAT

Prezado professor,

Autorizamos o acesso ao conjunto de dados sobre notas dos estudantes ao longo da história do curso de Matemática Industrial na tentativa de elaboração de um modelo preditivo baseado em redes neurais artificiais, conforme sua solicitação.

Atenciosamente,



Documento assinado eletronicamente por **Moisés Fernandes Lemos, Pró-Reitor Pro Tempore**, em 19/08/2021, às 14:38, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



A autenticidade deste documento pode ser conferida no site [https://sei.ufg.br/sei/controlador\\_externo.php?acao=documento\\_conferir&id\\_orgao\\_acesso\\_externo=0](https://sei.ufg.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0), informando o código verificador **2290735** e o código CRC **5C716041**.