

**INSTITUTO FEDERAL DE SANTA CATARINA**  
**CAMPUS CAÇADOR**  
**BACHARELADO EM ENGENHARIA DE PRODUÇÃO**

Loureno de Lara

**OTIMIZAÇÃO DA PROGRAMAÇÃO DA PRODUÇÃO EM UMA INDÚSTRIA DE**  
**EMBALAGENS UTILIZANDO REDES NEURAIIS ARTIFICIAIS**

Caçador  
Dezembro 2020

**INSTITUTO FEDERAL DE SANTA CATARINA**  
**CAMPUS CAÇADOR**  
**BACHARELADO EM ENGENHARIA DE PRODUÇÃO**

Loureno de Lara

**OTIMIZAÇÃO DA PROGRAMAÇÃO DA PRODUÇÃO EM UMA INDÚSTRIA DE  
EMBALAGENS UTILIZANDO REDES NEURAIS ARTIFICIAIS**

Trabalho de Conclusão de Curso  
submetido ao curso de Bacharelado em  
Engenharia de Produção Plena pelo  
Instituto Federal de Santa Catarina para a  
obtenção do título de Bacharel em  
Engenharia de Produção.

Orientador: Dr. Gabriel Hermann Negri

Caçador  
Dezembro 2020

Lara, Loureno de  
L318o Otimização da programação da produção em uma indústria de  
embalagens utilizando redes neurais artificiais. / Loureno de Lara ;  
orientador: Gabriel Hermann Negri. -- Caçador, SC, 2020.  
78 f.

Trabalho de Conclusão de Curso (Graduação)-Instituto Federal  
de Educação, Ciência e Tecnologia de Santa Catarina, Curso de  
Engenharia de Produção.  
Inclui bibliografias

1. Programação da produção. 2. Redes neurais artificiais. 3. Multi-  
Layer Perceptron. I. Negri, Gabriel Hermann. II. Instituto Federal de  
Educação, Ciência e Tecnologia de Santa Catarina. Curso de Engenharia  
de Produção. III. Título.

CDD 658.5



Loureno de Lara

**OTIMIZAÇÃO DA PROGRAMAÇÃO DA PRODUÇÃO EM UMA INDÚSTRIA DE  
EMBALAGENS UTILIZANDO REDES NEURAIAS ARTIFICIAIS**

Este trabalho de conclusão de curso foi julgado adequado para obtenção do título de Bacharel em Engenharia de Produção, e aprovado em sua forma final pelo curso superior Bacharelado em Engenharia de Produção Plena do Instituto Federal de Santa Catarina.

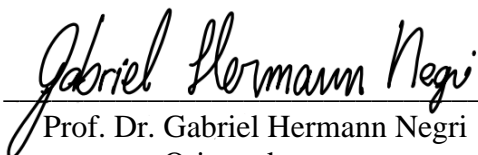
Caçador, 15 de dezembro de 2020.



---

Prof. Me. Lúcio Galvão Mendes  
Coordenador do Curso

Banca Examinadora:



---

Prof. Dr. Gabriel Hermann Negri  
Orientador  
Instituto Federal de Santa Catarina



---

Prof. Me. Lúcio Galvão Mendes  
Instituto Federal de Santa Catarina



---

Prof. Dr. Samuel da Silva Feitosa  
Instituto Federal de Santa Catarina



## AGRADECIMENTOS

Gostaria de agradecer e dedicar este trabalho,

Primeiramente à minha família, que compreendeu a minha ausência nas reuniões de família e me ofereceu o suporte necessário durante o difícil caminho percorrido para que conseguisse concluir mais essa etapa de minha vida.

À Milena Cardoso Mariano, que me incentivou, suportou e esteve ao meu lado por todos esses anos de estudo.

Ao professor e orientador Dr. Gabriel Hermann Negri, pelo apoio e direcionamento no desafio que foi escrever este Trabalho de Conclusão de Curso.

Aos professores membros da banca Me. Lúcio Galvão Mendes e Dr. Samuel da Silva Feitosa que foram grandes fontes de conhecimento e que inspiraram a realização deste trabalho.

Á todos os professores que compartilharam um pouco do seu conhecimento durante essa jornada:

Me. Bruno Santos Vieira  
Me. Eric Costa  
Dr. Eduardo do Nascimento  
Me. Eduardo Nascimento Pires  
Me. Francisco Sartori  
Me. Ismael Nogueira Rabelo de Melo  
Prof. Iuri Rafael Destro  
Me. Marcio Antônio Paulo  
Me. Paulo Roberto Cordova  
Me. Ricardo Guz  
Me. Robson Piacente Alves  
Dr. Rodrigo Acácio Paggi  
Me. Thaisa Rodrigues  
Dr. Vitor Sales Dias da Rosa

Aos colegas de sala que compartilharam dos momentos de alegria e sofrimento durante as provas, os trabalhos individuais e em grupo (principalmente), aos momentos de descontração nos corredores e na cantina, as viagens e visitas técnicas, aos finais de semana de estudo, tanto aqueles que estiveram juntos até o final como aqueles que passaram por um breve período de tempo.





Aos servidores do Instituto Federal de Santa Catarina, Campus Caçador, pela atenção e dedicação que sempre tiveram comigo e com todos os estudantes.

E principalmente, aos professores:

Dr. Pierry Teza

Dra. Danielle Regina Ullrich

Me. Marisa Santos Sanson

Maiores incentivadores de toda a minha carreira acadêmica e sem os quais provavelmente eu não estaria me formando e nem me tornado a pessoa que sou hoje.

E por fim, o meu muito obrigado à todos os que por ventura eu não mencionei, mas também tiveram a sua contribuição nessa odisséia conhecida como Engenharia.



*“Uma mente necessita de livros da mesma forma  
que uma espada precisa de uma pedra de amolar  
para se manter afiada”*

(Tyrion Lannister, As Crônicas de Gelo e Fogo)



## RESUMO

A maioria dos produtos que são comercializados possuem algum tipo de embalagem que tem por funcionalidade proteger fisicamente o produto, facilitar o transporte, a armazenagem e melhorar a estética. A empresa participante deste trabalho atua no ramo de fornecimento de embalagens sob encomenda, sendo o seu processo produtivo organizado para a fabricação em lotes de produção. As quantidades dos lotes de produção são variadas e dependem da demanda de cada cliente, cabendo ao setor de Programação e Controle da Produção (PCP), a atividade de programar a quantidade inicial de material necessário a entrar em processo de transformação para atender ao pedido. O PCP ao liberar uma ordem de produção precisa decidir qual a quantidade ótima a ser programada, levando em consideração as perdas do processo, para que não haja a falta ou o excesso de produtos bons ao final, ocasionando perda de faturamento quando não consegue atender integralmente o pedido do cliente, e aumento do custo com estocagem, deterioração e obsolescência quando a quantidade é fabricada em excesso. Pensando no potencial de utilização de uma Rede Neural Artificial (RNA) como ferramenta de auxílio ao PCP, estabeleceu-se como questão de pesquisa o desenvolvimento de uma RNA para otimizar a atividade de programação da quantidade ideal de produção. Dentre os modelos encontrados na literatura, optou-se pelo Multi-Layer Perceptron (MLP) e seus resultados obtidos foram comparados com os resultados do método utilizado atualmente de 5, 10 e 15% de produção à mais sobre a quantidade. Como conclusão o desempenho do modelo mostrou ser mais eficaz para a tarefa.

**Palavras-chave:** Programação da produção. Redes neurais artificiais. RNA. Multi-Layer Perceptron. MLP.



## ABSTRACT

Most products that are marketed have some type of packaging that has the functionality to physically protect the product, facilitate transportation, storage and improve aesthetics. The company participating in this work operates in the field of supply of custom packages, and its production process is organized for manufacturing in production batches. The quantities of the production batches are varied and depend on the demand of each customer, the Programming and Production Control (PCP) sector being responsible for programming the initial quantity of material necessary to enter the transformation process to fulfill the order. The PCP, when releasing a production order, needs to decide the optimum quantity to be programmed, taking into account the losses of the process, so that there is no shortage or excess of good products at the end, causing loss of revenue when it is unable to fully meet customer order, and increased cost of storage, deterioration and obsolescence when the quantity is manufactured in excess. Thinking about the potential of using an Artificial Neural Network (ANN) as a tool to help the PCP, the development of an ANN was established as a research question to optimize the programming activity of the ideal amount of production. Among the models found in the literature, the Multi-Layer Perceptron (MLP) was chosen and its results were compared with the results of the method currently used of 5, 10 and 15% of production over quantity. In conclusion, the performance of the model proved to be more effective for the task.

**Keywords:** Production schedule. Artificial Neural Network. ANN. Multi-Layer Perceptron. MLP.





## RESUMEM

La mayoría de los productos que se comercializan cuentan con algún tipo de empaque que tiene la funcionalidad de proteger físicamente el producto, facilitar el transporte, almacenamiento y mejorar la estética. La empresa que participa en este trabajo opera en el campo del suministro de envases personalizados, y su proceso de producción está organizado para la fabricación en lotes de producción. Las cantidades de los lotes de producción son variadas y dependen de la demanda de cada cliente, siendo el sector de Programación y Control de Producción (PCP) el responsable de programar la cantidad inicial de material necesaria para ingresar al proceso de transformación para cumplir con el pedido. El PCP, al liberar una orden de producción, necesita decidir la cantidad óptima a programar, teniendo en cuenta las pérdidas del proceso, para que no haya escasez o exceso de buenos productos al final, provocando pérdida de ingresos cuando se incapaz de satisfacer completamente el pedido del cliente, y aumento del costo de almacenamiento, deterioro y obsolescencia cuando la cantidad se fabrica en exceso. Pensando en el potencial de utilizar una Red Neural Artificial (RNA) como herramienta de ayuda al PCP, se estableció el desarrollo de una RNA como cuestión de investigación para optimizar la actividad de programación de la cantidad ideal de producción. Entre los modelos encontrados en la literatura, se eligió el Perceptrón Multicapa (MLP) y sus resultados se compararon con los resultados del método actualmente utilizado de 5, 10 y 15% de producción sobre cantidad. En conclusión, el desempeño del modelo demostró ser más efectivo para la tarea.

**Palabras clave:** Programación de la producción. Redes Neuronales Artificiales. RNA. Multi-Layer Perceptron. MLP.



## LISTA DE FIGURAS

Figura 1 - Modelo do processo de produção.....	32
Figura 2 – Relação entre Volume e Variedade dos tipos de arranjos de produção.....	33
Figura 3 - Prazos, atividades e objetivos desempenhados pelo PCP.....	34
Figura 4 - Neurônio biológico.....	41
Figura 5 - Modelo de neurônio artificial.....	43
Figura 6 - Sentido do fluxo de produção e do algoritmo da RNA.....	56
Figura 7 – Faixa de tolerância dos $\pm 10\%$ e ponto ótimo de atuação da RNA.....	73
Gráfico 1 - Dispersão do erro percentual dos resultados da RNA.....	65
Gráfico 2 - Dispersão do erro percentual do método atual.....	65
Gráfico 3 - Boxplot da distribuição do erro percentual da RNA.....	66
Gráfico 4 - Boxplot da distribuição do erro percentual do método atual (5, 10 e 15%)....	67



## LISTA DE TABELAS

Tabela 1 - Similaridade do funcionamento entre o neurônio biológico e o artificial.....	52
Tabela 2 - Características da RNA do trabalho.....	53
Tabela 3 - Etapas e atividades de desenvolvimento do modelo de RNA.....	58
Tabela 4 - Hipóteses formuladas para o modelo.....	59
Tabela 5 - Divisão do conjunto de dados conforme a quantidade.....	60
Tabela 6 – Resultados da simulação da RNA.....	63
Tabela 7 - Resultados do conjunto geral de dados.....	64
Tabela 8 - Resultados dos pedidos até 10.000.....	67
Tabela 9 - Resultados dos pedidos entre 10.000 e 20.000 unidades.....	68
Tabela 10 - Resultados dos pedidos entre 20.000 e 30.000 unidades.....	69
Tabela 11 - Resultados dos pedidos entre 30.000 e 40.000 unidades.....	69
Tabela 12 - Resultados dos pedidos entre 40.000 e 50.000 unidades.....	70
Tabela 13 - Resultados dos pedidos acima de 50.000 unidades.....	71
Tabela 14 - Síntese dos resultados.....	72



## LISTA DE ABREVIATURAS E SIGLAS

ABEPRO	Associação Brasileira de Engenharia de Produção
ERP	<i>Enterprise Resource Planning</i> (Planejamento de Recursos Naturais)
IA	Inteligência Artificial
KNN	<i>K Nearest Neighbor</i> (K Vizinho mais Próximo)
MAE	<i>Mean Absolute Error</i> (Erro Médio Absoluto)
MLP	<i>Multi-Layer Perceptron</i> ( <i>Perceptron</i> Multi-Camadas)
PCP	Programação e Controle da Produção
PET	<i>Polyethylene Terephthalate</i> (Poli Tereftalato de Etila)
PMP	Plano Mestre da Produção
PO	Pesquisa Operacional
RBF	<i>Radial Basis Function</i> (Função de Base Radial)
RMSE	<i>Radial Basis Function</i> (Função de Base Radial)
RNA	<i>Radial Basis Function</i> (Função de Base Radial)





## SUMÁRIO

<b>1 INTRODUÇÃO</b> .....	27
1.1 CONTEXTO E QUESTÃO DE PESQUISA .....	27
1.2 OBJETIVOS .....	28
<b>1.2.1 Objetivo geral</b> .....	28
<b>1.2.2 Objetivos específicos</b> .....	29
1.3 JUSTIFICATIVA .....	29
1.4 ESTRUTURA DO TRABALHO .....	30
<b>2 REVISÃO DE LITERATURA</b> .....	31
2.1 SISTEMA DE PRODUÇÃO .....	31
<b>2.1.1 Função produção</b> .....	31
<b>2.1.2 Tipos de arranjos físicos de produção</b> .....	32
<b>2.1.3 Planejamento e Controle da Produção</b> .....	34
<b>2.1.4 Pesquisa Operacional</b> .....	36
<b>2.1.5 Conclusão da revisão da literatura sobre sistema de produção</b> .....	38
2.2 REDES NEURAIS ARTIFICIAIS .....	39
<b>2.2.1 Histórico das RNA</b> .....	39
<b>2.2.2 Neurônio biológico</b> .....	40
<b>2.2.3 Neurônio artificial</b> .....	41
<b>2.2.4 Aprendizagem de máquina</b> .....	45
<b>2.2.5 Tipos de aprendizagem</b> .....	50
<b>2.2.6 Modelos de arquitetura de RNA</b> .....	51
<b>2.2.7 Conclusão da revisão de literatura sobre RNA</b> .....	52
<b>3 PROCEDIMENTOS METODOLÓGICOS</b> .....	54
3.1 DEFINIÇÃO DO TEMA E DOS OBJETIVOS DO TRABALHO .....	54
3.2 LEVANTAMENTO SISTEMÁTICO DA LITERATURA .....	55
3.3 DESENVOLVIMENTO DO MODELO .....	55
3.4 FUNCIONAMENTO ALGORITMO .....	55
3.5 VERIFICAÇÃO DO MODELO .....	57
<b>4 DESENVOLVIMENTO DO MODELO</b> .....	58
4.1 PLANEJAMENTO .....	58
4.2 PREPARAÇÃO DOS DADOS .....	60
4.3 MODELAGEM DO ALGORITMO DA RNA .....	61
4.4 CÁLCULO DO ERRO DO MÉTODO ATUAL DE PROGRAMAÇÃO .....	61

<b>5 RESULTADOS</b> .....	63
5.1 SENSIBILIDADE DA RNA .....	63
5.2 RESULTADOS PARA O CONJUNTO GERAL .....	64
5.2 RESULTADOS PARA QUANTIDADES ATÉ 10.000 UNIDADES .....	67
5.3 RESULTADOS PARA QUANTIDADES ENTRE 10.000 E 20.000 UNIDADES .....	68
5.4 RESULTADOS PARA QUANTIDADES ENTRE 20.000 E 30.000 UNIDADES .....	68
5.5 RESULTADOS PARA QUANTIDADES ENTRE 30.000 E 40.000 UNIDADES .....	69
5.6 RESULTADOS PARA QUANTIDADES ENTRE 40.000 E 50.000 UNIDADES .....	70
5.7 RESULTADOS PARA QUANTIDADES SUPERIORES À 50.000 UNIDADES .....	70
<b>6 CONCLUSÕES</b> .....	72
6.1 SÍNTESE DOS RESULTADOS .....	72
6.2 ATENDIMENTO AOS OBJETIVOS DA PESQUISA .....	74
6.3 LIMITAÇÕES DA PESQUISA .....	74
6.4 CONTINUIDADE DA PESQUISA E TRABALHOS FUTUROS .....	74
<b>REFERÊNCIAS</b> .....	76
<b>APÊNDICE A.</b> <i>Algoritmo do modelo de RNA aplicado</i> .....	77
<b>APÊNDICE B.</b> <i>Algoritmo usado para calcular o erro do método atual de programação de 5, 10 e 15%</i> .....	79

## 1 INTRODUÇÃO

O presente trabalho tem como tema central a otimização da programação da produção em uma indústria de embalagens fazendo o uso de uma rede neural artificial (RNA). Dentre os pontos que serão abordados neste capítulo estão a contextualização do tema, a questão de pesquisa, o objetivo geral, os objetivos específicos, a justificativa, o método de pesquisa, a estrutura do trabalho e a concordância do tema com o curso de Graduação em Engenharia de Produção.

### 1.1 CONTEXTO E QUESTÃO DE PESQUISA

Diariamente uma grande variedade de produtos e serviços são produzidos e comercializados por empresas dos mais diversos segmentos de mercado com o objetivo de obterem lucro, aumentarem o seu capital e prosperarem. Para conseguirem isso, elas procuram satisfazer as necessidades básicas, gerais e de conforto dos clientes através de ofertas de valor pela qual estes estejam dispostos a pagar.

Tomando mais especificamente como exemplo produtos tangíveis como alimentos, eletrodomésticos, equipamentos eletrônicos ou brinquedos, todos possuem alguma espécie de embalagem, seja para proteção, transporte, armazenamento ou estética, mas não somente isso, a embalagem hoje é parte importante na estratégia de marketing das empresas, pois torna a experiência de compra mais agradável ao consumidor e ajuda a alavancar as vendas. Por ser uma característica comum presente na maioria dos produtos, a indústria que atua no ramo de fornecimento de embalagens (flexíveis, papelão, alumínio, vidro, PET ou tetrapak) possui uma demanda potencial de consumo advinda dos outros segmentos de mercado.

A empresa foco de estudo da problemática deste trabalho científico tem atuação no setor de fornecimento de embalagens sob encomenda, sendo o seu processo produtivo organizado para a fabricação de lotes de produção. As quantidades dos lotes de produção são variadas e dependem da demanda de cada cliente, cabendo ao setor de Programação e Controle da Produção (PCP), a atividade de programar a quantidade inicial de material necessário a entrar em processo de transformação para atender ao pedido do cliente. O PCP ao liberar uma ordem de produção precisa decidir qual a quantidade ótima a ser programada para que não haja a falta ou o excesso de produtos bons ao final do processo, ocasionando perda de faturamento por não atender integralmente ao pedido do cliente e às vezes a perda do mesmo,

e o aumento do custo com estocagem, deterioração e obsolescência das quantidades fabricadas em excesso.

Na hora de programar o lote inicial de produção, a quantidade programada deve ser maior que a solicitada, pois ao longo do processo a quantidade de itens bons em conformidade com os requisitos do cliente vai diminuindo em decorrência de *setup* de máquina, defeito no material, falhas no processo, erro operacional, medição errada ou ambiente inadequado, que são caracterizadas como perdas do processo. Por isso, para garantir que ao final da última etapa do roteiro de produção a empresa consiga entregar o combinado, o método atual utilizado pelo PCP consiste em mandar produzir uma porcentagem à mais sobre a quantidade do pedido, que hoje é de 5% para a entrega de quantidades exatas, 10% para pedidos normais e 15% para pedidos com histórico de grandes perdas ou características difíceis de serem produzidas. Existe ainda uma tolerância sobre a quantidade total do pedido que é  $\pm 10\%$  a qual o cliente aceita receber, firmada quando o pedido de venda é confirmado.

Uma Rede Neural Artificial (RNA) é um algoritmo baseado no funcionamento do cérebro humano e que assim como ele, pode aprender a executar tarefas específicas e ser aplicado para resolver problemas. Alguns pontos importantes no desenvolvimento de uma RNA são a escolha dos atributos de entrada corretos que possuam correlação com as saídas esperadas, uma significativa quantidade de dados de amostra, o treinamento e a validação do modelo.

Acredita-se que o desenvolvimento de um modelo possa ser usado como ferramenta de auxílio à tomada de decisão pelo departamento de PCP respondendo a seguinte questão de pesquisa “como otimizar o processo de programação da produção em uma indústria de embalagens utilizando uma RNA”.

## 1.2 OBJETIVOS

A partir da questão de pesquisa são definidos os objetivos geral e específicos do presente trabalho.

### 1.2.1 Objetivo geral

O objetivo geral é criar um modelo de RNA que auxilie o departamento de Programação e Controle da Produção (PCP) na otimização da atividade de Programação da Produção através da predição da quantidade necessária de material a ser transformado, para

que ao final de todas as etapas do processo produtivo a variação entre a quantidade de itens bons do lote entregue e a quantidade solicitada pelo cliente seja a menor possível.

### 1.2.2 Objetivos específicos

Com base no objetivo geral definiram-se os seguintes objetivos específicos:

- a) Compreender os principais conceitos e o funcionamento das RNA;
- b) Conhecer os modelos de RNA e algoritmos de aprendizagem mais comumente utilizados;
- c) Escrever o algoritmo de uma RNA para executar a tarefa de predição do *input* inicial de produção;
- d) Comparar os resultados obtidos pela RNA com a metodologia atualmente utilizada pelo PCP para determinar a quantidade de *input* inicial de produção;
- e) Realizar discussões acerca das implicações dos resultados obtidos nas receitas e custos da empresa.

### 1.3 JUSTIFICATIVA

O uso de redes neurais artificiais (RNA) apresenta-se como uma promissora área da tecnologia da informação para a solução de problemas e aproveitamento de oportunidades nos mais diversos ramos de negócios, envolvendo por exemplo avaliação de imagens via satélite, controle de trens em alta velocidade, previsão de ações no mercado financeiro, identificação de fraudes de cartão de crédito, controle de aparelhos eletrônicos e diagnóstico de doenças (SILVA, SPATTI, FLAUZINO, 2016).

O presente trabalho também vem a contribuir, através da indissociabilidade, interdisciplinaridade e interprofissionalidade entre *Ensino – Pesquisa – Extensão*. A indissociabilidade é relativa à algo que não existe sem a presença do outro, a falta de uma das partes, ensino, pesquisa ou extensão não permite a constituição do todo, sendo este todo a relação entre a instituição de ensino, o graduando e a sociedade. A interdisciplinaridade refere-se a interligação e interdependência entre as disciplinas, sendo que uma é dependente da outra, não sendo possível a sua completa existência isoladamente. A interprofissionalidade assemelha-se à interdisciplinaridade, com a diferença de a interligação e interdependência são relativas ao trabalho ou profissão.

Desse modo, o objetivo geral “criar um modelo de RNA que auxilie o departamento de PCP na otimização da atividade de Programação da Produção através da predição da quantidade necessária de material a ser transformado, para que ao final de todas as etapas do processo produtivo a variação entre a quantidade de itens bons do lote entregue e a quantidade solicitada pelo cliente seja a menor possível”, enquadra-se na proposta de manter uma relação entre a instituição de ensino, o graduando e a sociedade, envolvendo nesse processo de pesquisa científica as diversas áreas da engenharia de produção (computação, sistemas de informação, processos industriais, cálculo numérico, planejamento e controle da produção, pesquisa operacional, modelagem de problemas de engenharia) e as profissões presentes no ambiente industrial.

#### 1.4 ESTRUTURA DO TRABALHO

O presente trabalho está estruturado em 6 capítulos, conforme descritos à seguir:

- a) O primeiro capítulo diz respeito à introdução, onde estão descritos o tema, a proposta de trabalho, o objetivo geral e os específicos, a justificativa e a estrutura do trabalho;
- b) O segundo capítulo refere-se à revisão da literatura, onde são desenvolvidos os conceitos que fundamentam a proposta do trabalho. Esse capítulo está dividido em duas partes sendo a primeira referente aos processos de produção e a segunda referente às redes neurais artificiais (RNA);
- c) No terceiro capítulo, estão apresentados os procedimentos metodológicos utilizados para o desenvolvimento do trabalho, como a definição do tema e dos objetivos, o método de levantamento sistemático da revisão de literatura, o desenvolvimento e a verificação do modelo de RNA;
- d) No quarto capítulo serão apresentadas as etapas de desenvolvimento do modelo de RNA;
- e) No quinto capítulo, estão apresentados os resultados deste trabalho;
- g) No sexto capítulo, estão expostas as conclusões, assim como a síntese dos resultados, os objetivos alcançados, as limitações e proposta para trabalhos futuros.

## 2 REVISÃO DE LITERATURA

O presente capítulo está dividido em duas partes e tem por objetivo revisar a literatura sobre: (1) os principais conceitos relacionados ao processo de produção, Planejamento e Controle de Produção (PCP) e Pesquisa Operacional (PO) e, (2) os principais conceitos referentes às redes neurais artificiais (RNA) e seus subtemas relacionados.

### 2.1 SISTEMA DE PRODUÇÃO

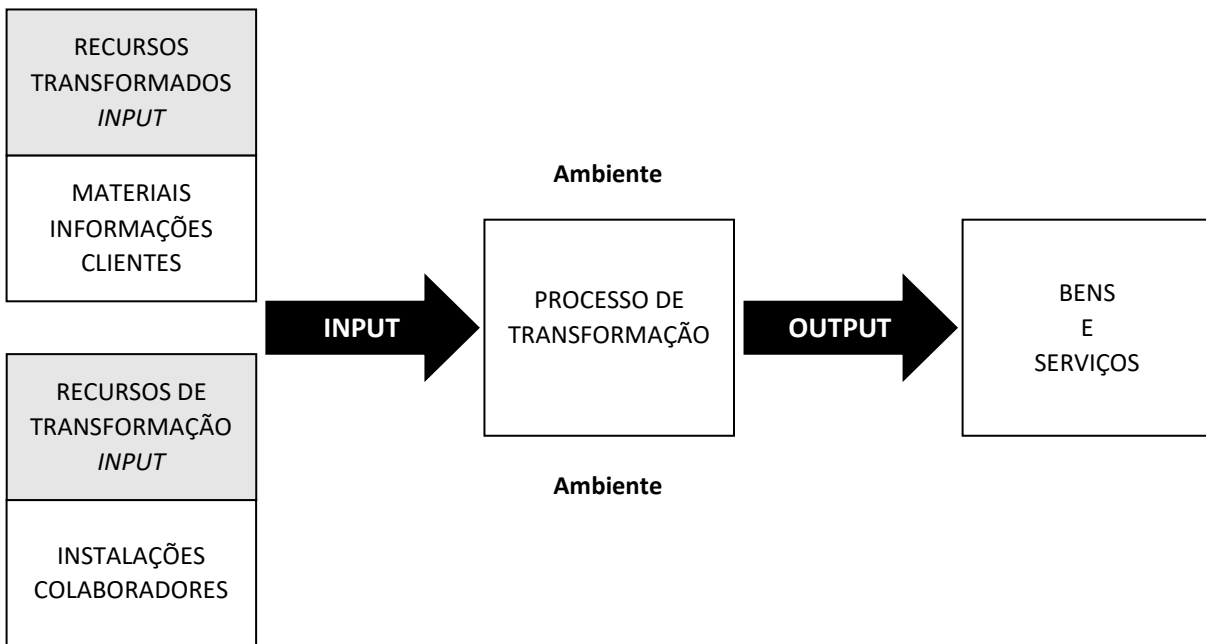
A razão da existência de uma organização é a obtenção de lucro por meio de uma oferta de valor que atenda às necessidades e desejos de uma demanda. Neste contexto define-se valor como a razão entre o que o cliente recebe (bens, serviços, informações, experiências ou ideias) e o que ele paga (KOTLER, KELLER, 2012). Todo o processo de criação de valor envolve inúmeras variáveis como capital, instalações, equipamentos, colaboradores, relacionamento com clientes e fornecedores, que são agrupadas em atividades e funções cada qual com seu papel dentro da organização. Mais especificamente, a reunião de recursos destinados à produção de bens e serviços representa a chamada função produção (SLACK et al, 2006).

#### 2.1.1 Função produção

A função produção segue um modelo de operação, representado na figura 1, no qual um conjunto de entradas (*inputs*) são transformados ou usados para transformar algo em saídas (*outputs*) de bens e serviços, por meio de um processo de transformação (SLACK et al, 2006). Fatores ambientais estão presentes externamente influenciando direta ou indiretamente nas decisões e ações da produção (SILVA, 2008).

Os *inputs* são divididos em recursos transformados (materiais, clientes, informações) e recursos de transformação (equipamentos, colaboradores). Os recursos transformados tem suas características modificadas para alguma outra forma ou recebem algum tipo de serviço vindo a se tornar os *outputs*, enquanto os recursos de transformação trabalham sobre os recursos transformados, modificando suas propriedades e entregando os *outputs* (SLACK et al, 2006).

Figura 1: Modelo do processo de produção



Fonte: Adaptado de SLACK et al, 2006

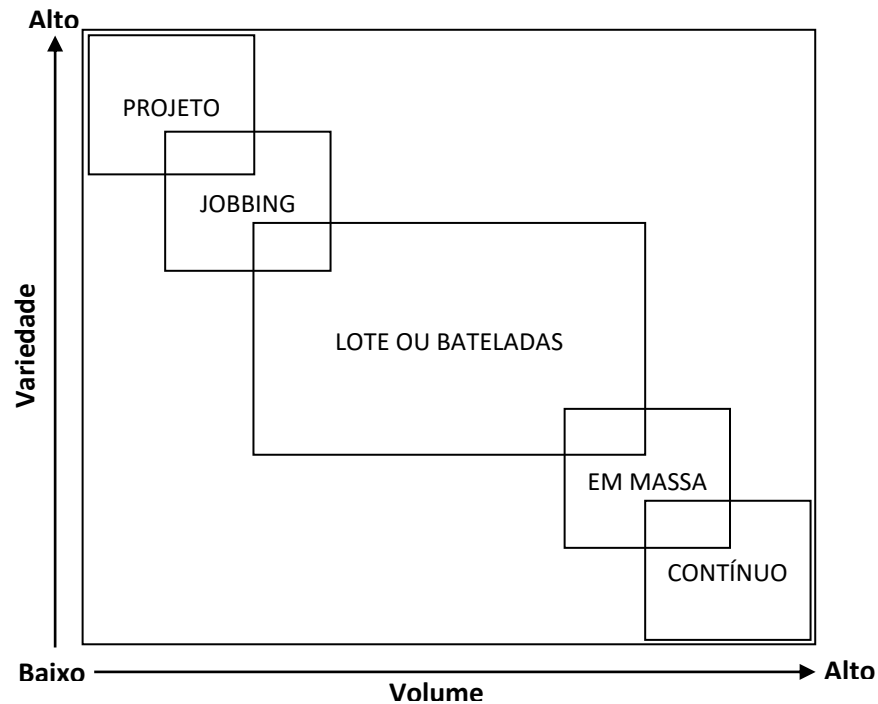
Geralmente nas indústrias, mesmo a fabricação dos *outputs* mais simples demandam várias etapas no processo de transformação. Uma etapa pode então entregar *outputs* para alimentar a etapa seguinte, que por sua vez transforma o material e alimenta a seguinte e assim sucessivamente até o produto final ser obtido, a essa relação de recebimento e entrega entre as etapas de produção dentro de uma empresa é o que chamamos de fornecedor interno e cliente interno (SLACK et al, 2006).

### 2.1.2 Tipos de arranjos físicos de produção

Existem vários tipos de arranjos físicos de produção, pois dependendo do produto e do serviço oferecido torna-se mais vantajoso para a empresa adotar um modelo onde a variedade e o volume alinhados com o uso eficiente dos recursos resultem no menor custo possível. Os processos de produção são organizados segundo um dos seguintes modelos de arranjos físicos das atividades: por produto, por *jobbing*, em lotes ou bateladas, produção em massa ou contínuo (SLACK et al, 2006). A relação entre o volume e a variedade de produção entre os arranjos citados é representado na figura 2.



Figura 2: Relação entre Volume e Variedade dos tipos de arranjos de produção



Fonte: Adaptado de SLACK et al, 2006

O arranjo por projeto é o mais indicado quando o produto ou serviço oferecido demanda um alto nível de customização, alta variedade, baixo volume e períodos de produção longos (SLACK et al, 2006), como é o caso da construção de prédios, navios e aeronaves.

Os processos de *jobbing* são muito parecidos com o processo por projeto, pois também lidam com alta customização, alta variedade, baixo volume e períodos grandes de produção, no entanto o seu diferencial é a possibilidade de compartilhamento dos recursos com outros produtos (SLACK et al, 2006), podendo o processo de *jobbing* ser empregado, por exemplo na fabricação de máquinas industriais que compartilham as mesmas matérias-primas (chapas de aço, ferro fundido, alumínio) e recursos (centros de usinagem, solda e pintura).

Processos que demandam maior equilíbrio entre volume e variedade podem adotar processos baseados em lotes ou bateladas. As atividades e recursos neste tipo de processo podem ser organizados de forma a obter alto volume de produção e maior flexibilidade na variedade de produtos. Um lote de produção significa que durante um certo período de tempo um produto será produzido inúmeras vezes até que se obtenha uma certa quantidade, sendo a familiaridade a principal característica que permite a entrada e saída rápida de diferentes produtos em um processo (SLACK et al, 2006). Empresas que trabalham com embalagens, alimentos, roupas e calçados normalmente tem a sua produção baseada em lotes.

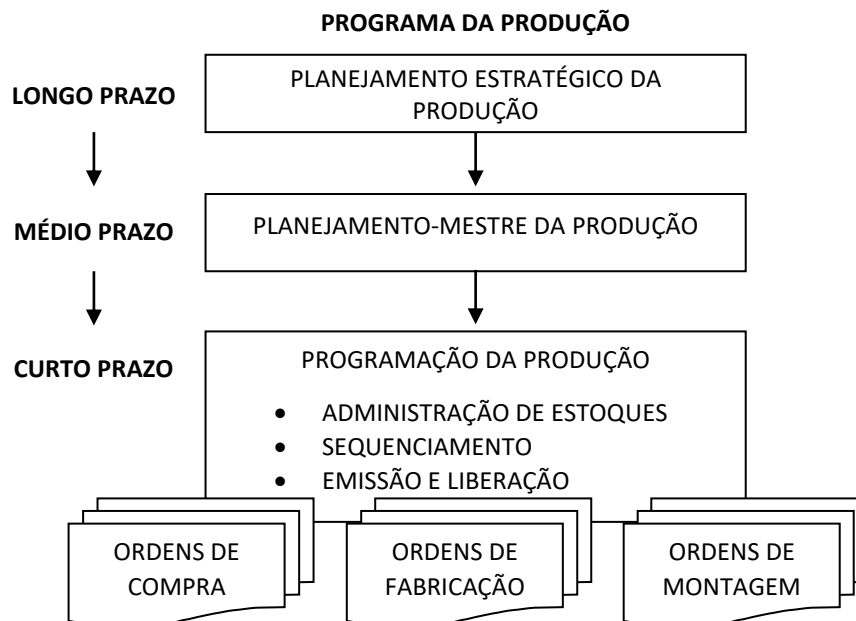
Os processos de produção em massa por sua vez conseguem produzir um alto volume de itens em um espaço de tempo relativamente curto graças à padronização e repetitividade das atividades. Essa grande capacidade de produção, no entanto é acompanhada de uma menor variedade de produtos (SLACK et al, 2006). A produção em massa é utilizada na fabricação de eletrodomésticos, automóveis, eletrônicos e remédios.

Por fim, o processo com o maior volume de produção é realizado por meio do fluxo contínuo, ou seja, ininterruptamente (SLACK et AL, 2006). A combinação de complexas instalações, investimento em equipamentos, elevada padronização dos processos, alto grau de repetição das tarefas, abastecimento constante da linha e paradas menos frequentes, permitem que o processo de produção contínuo tenha o menor custo operacional dentre todos os processos.

### 2.1.3 Planejamento e Controle da Produção

O planejamento e controle da produção (PCP) é um departamento de apoio responsável pela coordenação e aplicação dos recursos produtivos de forma a atender da melhor maneira possível aos planos estabelecidos nos níveis estratégicos, táticos e operacionais (TUBINO, 2007). Normalmente fazendo isso por meio de um conjunto de sistemas, métodos e procedimentos (SLACK et al, 2006). O PCP desempenha suas atividades dentro de um Programa de Produção nos períodos de longo, médio e curto prazo apresentadas na figura 3.

Figura 3: Prazos, atividades e objetivos desempenhados pelo PCP



Fonte: Adaptado de TUBINO, 2007

Para atender aos objetivos estratégicos da organização no longo prazo, durante o planejamento estratégico da produção é elaborado o Plano de Produção contendo as metas de direcionamento da produção. O Plano de Produção relaciona as informações de vendas e produção para definir as decisões relativas às questões produtivas como capacidade de produção necessária, alocação de recursos físicos, financeiros e de pessoal. O horizonte de perspectiva do Plano de Produção geralmente envolve o prazo de um à dois anos e é medido em meses ou trimestres (TUBINO, 2007).

O planejamento-mestre da produção é realizado por sua vez em períodos de médio prazo, sendo responsável por:

O planejamento-mestre da produção está encarregado de desmembrar os planos produtivos estratégicos de longo prazo em planos específicos de produtos acabados (bens ou serviços) para o médio prazo, no sentido de direcionar as etapas de programação e execução das atividades operacionais da empresa (montagem, fabricação e compras) (TUBINO, 2007, P. 51).

As decisões tomadas no planejamento-mestre da produção serão oficializadas em um documento chamado Plano Mestre de Produção (PMP). O PMP contém as táticas e informações necessárias para conduzir eficientemente o sistema de produção de forma a atender as expectativas estratégicas, validar o planejamento de produção e atender as previsões de vendas futuras de médio prazo e de pedidos já em carteira. O período de abrangência do PMP é de meses ou semanas (TUBINO, 2007).

No curto prazo encontra-se a Programação da Produção, responsável principalmente pelas atividades de (1) administração de estoques, que envolve o planejamento e controle dos estoques de matérias-primas, produtos semi-acabados e acabados, (2) o sequenciamento da produção através da definição de uma sequência de fabricação que utilize de forma inteligente os recursos disponíveis e, (3) a emissão e liberação de ordens de compra e fabricação. O período de alcance da Programação da Produção é de alguns dias (TUBINO, 2007).

A atividade de Programação da Produção, mais especificamente o sequenciamento (programação) é uma das atividades mais complexas do PCP, pois o número de programações possíveis cresce à medida que se aumenta o número de tarefas e processos de fabricação (SLACK et al, 2006). Como consequências, uma programação feita de forma inadequada pode ocasionar perdas de receita pela empresa quando a demanda não é atendida, ou aumento dos custos quando a quantidade produzida excede a demanda e a empresa tem que arcar com o estoque até que seja vendido. Os estoques de produtos acabados gerados pelo desequilíbrio entre fornecimento e demanda, são custosos e arriscados de se manterem por muito tempo,

pois além de ocuparem espaço e representarem “capital parado” também podem se perder por deterioração ou obsolescência (SLACK et al, 2006). Por isso a adoção de métodos de programação adequados, podem ajudar o departamento de PCP à alocar os recursos exatos para produzir a quantidade adequada de produtos.

#### **2.1.4 Pesquisa Operacional**

No cotidiano das organizações a execução das atividades de produção encontra diversos percalços pelo caminho. Alguns problemas possuem tempo de resposta rápido, enquanto outros demandam de maior planejamento para serem resolvidos. A tomada de decisão é fundamental para uma organização conseguir entregar uma demanda adequada, em um tempo apropriado, com o nível de qualidade desejado e de uma maneira compatível com as metas (SILVA, 2008). Independente do problema, a tomada de decisão deve encontrar a melhor solução possível que esteja dentro da realidade da empresa, ou seja, deve ser uma solução viável física, temporal e financeiramente. Um método muito utilizado para ajudar os gestores na tomada de decisão é a Pesquisa Operacional (PO).

Embora muito antes já existisse uma abordagem científica da gestão das organizações, o surgimento da PO é atribuído às ações militares durante as guerras mundiais do século XX. Durante a Primeira Guerra Mundial (1914 - 1918) surgiram os trabalhos pioneiros desenvolvidos na Inglaterra e EUA para a análise de operações militares e na guerra antissubmarinos, ficando o estudo inicialmente conhecido como análise de operações (HILLIER, LIEBERMAN, 2013; SILVA, 2008). Anos mais tarde no período da Segunda Guerra Mundial (1939 - 1945), Inglaterra e EUA convocaram uma grande quantidade de cientistas com o objetivo de realizarem pesquisas militares que auxiliassem as decisões militares estratégicas e táticas do campo de batalha, como por exemplo a utilização eficiente dos limitados recursos (HILLIER, LIEBERMAN, 2013). Foi a partir de então que a nova ciência passou a ser conhecida como pesquisa operacional. Após o final da guerra, o sucesso das técnicas de análise e otimização desenvolvidas na PO despertaram o interesse de outras áreas, destacando-se no setor industrial onde muitas pessoas que haviam participado das equipes de pesquisa na guerra tornaram-se consultores (HILLIER, LIEBERMAN, 2013; SILVA, 2008). Nas próximas décadas o interesse crescente foi motivado principalmente pelo (1) desenvolvimento de estudos importantes na área que levaram a criação de métodos como simplex, programação linear, programação dinâmica, teoria das filas e teoria do inventário, e (2) a revolução computacional que trouxe computadores mais rápidos e potentes, ideais para

processar grandes quantidades de dados e resolverem problemas mais complexos (HILLIER, LIEBERMAN, 2013).

De acordo com Silva (2008), PO “é um método quantitativo que ajuda no planejamento, na solução de problemas e no processo de tomada de decisão”. Hillier e Lieberman (2013), complementam dizendo que “a pesquisa operacional usa uma abordagem que relembra a maneira pela qual são conduzidas pesquisas em campos científicos usuais. Em grau considerável, o método científico é utilizado para investigar o problema empresarial”.

O principal benefício da utilização da pesquisa operacional é a simulação de diversos estados possíveis de resolução sem haver a necessidade de colocar todos em prática, evitando riscos, gastos com tempo e capital (SILVA, 2008).

A aplicação de sua metodologia envolve primeiramente uma análise do problema de pesquisa, observando, formulando e coletando dados para em um segundo momento construir um modelo matemático que simule uma representação suficientemente próxima da realidade, formulando uma hipótese que contenha objetivo(s), variáveis e restrições. O modelo então experimenta diversas configurações possíveis durante a fase de testes até encontrar um resultado que atinja o objetivo e o desempenho que seja considerado ótimo, para na sequência realizar a etapa de validação do modelo e, por fim, usá-lo como ferramenta para auxiliar na tomada de decisão (HILLIER, LIEBERMAN, 2013; SILVA, 2008).

Uma das principais características da PO é a busca por encontrar uma melhor solução (solução ótima) de muitas possíveis para um modelo que representa um problema (HILLIER, LIEBERMAN, 2013). Segundo Silva (2008), outras características fundamentais da PO são:

- O foco na solução do problema;
- A orientação por critérios lógicos e econômicos;
- O uso de modelos e técnicas matemáticas;
- A ênfase no uso de computadores nos sistemas de suporte à decisão.

As principais técnicas utilizadas pela pesquisa operacional para auxiliar na tomada de decisão são a previsão, a matriz de resultados e a árvore de decisão. A previsão utiliza-se das variáveis e condições do ambiente para fazer previsões do futuro. A matriz de resultados é uma ferramenta na forma de tabela para análise de um problema a partir de quatro variáveis principais: estratégias, estados da natureza, probabilidade e resultados. Por último, a árvore de decisão é uma técnica de representação gráfica em forma de ramificações para visualização

das consequências de uma ação. As variáveis utilizadas para a ramificação da árvore envolvem probabilidade estimada, resultados e valor estimado (SILVA, 2008).

Existem três condições ou estados básicos da natureza, que exercem influência sobre a tomada de decisão e determinam suas consequências sobre um problema. O primeiro deles diz respeito às (1) condições de certeza que ocorrem quando se sabe exatamente qual será a consequência de uma ação, sendo o processo de tomada de decisão baseado em modelos determinísticos. O segundo estado básico diz respeito à (2) condição de risco, ou seja, é possível relacionar cada decisão a um conjunto de resultados possíveis e suas respectivas probabilidades de ocorrerem. Nessa condição a decisão é tomada baseada em modelos estocásticos. Por fim, existem ainda as (3) condições de incerteza, nas quais encontram-se as mais difíceis situações para a tomada de decisão, pois neste estado é até possível identificar os resultados para uma decisão, no entanto não há nenhuma probabilidade desses resultados acontecerem (SILVA, 2018).

### **2.1.5 Conclusão da revisão da literatura sobre sistema de produção**

Uma organização tem por objetivo obter lucro através da venda de produtos e serviços que atendam a uma demanda específica. Seus produtos ou serviços são obtidos por meio de um processo de produção que consiste em submeter *inputs* (materiais, pessoas, informações) a um processo de transformação por meio de recursos transformadores (equipamentos, colaboradores) em *outputs* (produtos acabados, finalização de um serviço).

Dependendo do tipo de produto ou serviço que a empresa comercializa, o seu processo de produção pode ser organizado e estruturado para obter o melhor custo x benefício entre as variáveis volume e variedade. Os tipos mais comuns de organização e estruturação de processos são por projeto, *jobbing*, em lote ou batelada, produção em massa e produção contínua.

Para gerenciar o processo de produção, existe dentro das empresas o departamento de Planejamento e Controle da Produção, responsável por elaborar um Programa de Produção composto pelo Planejamento Estratégico da Produção (longo prazo), o Planejamento Mestre da Produção (médio prazo) e a atividade de Programação e Controle da Produção (curto prazo).

Para auxiliar no planejamento e na solução de problemas relacionados à produção podem ser aplicados métodos de PO que utilizam modelos de simulação. Esses modelos

permitem experimentar várias configurações possíveis para alocação dos recursos com o objetivo de encontrar aquela que represente *uma* das soluções ótimas possíveis.

Uma das técnicas utilizadas em PO é a previsão, na qual utilizando as variáveis e condições do ambiente, um modelo matemático e computacional simula a realidade e faz previsões do futuro.

Pensando-se na criação de um modelo de simulação e previsão de soluções ótimas para a PO, uma RNA apresenta-se como uma técnica *diferente* de construção de um modelo que pode ser usado pelo departamento de PCP de uma organização para gerenciar de forma mais eficiente seus recursos.

## 2.2 REDES NEURAIS ARTIFICIAIS

A ideia de construir sistemas racionais capazes de reproduzir habilidades humanas como criatividade, aprendizado e raciocínio lógico vem sendo estudados desde meados da década de 1950 pelo campo da Inteligência Artificial (IA). A IA é uma ciência que se desenvolveu a partir da combinação de conhecimentos advindos do estudo da matemática, neurociência, engenharia e computação. Atualmente seu campo de estudo pode ser dividido em áreas com propósitos gerais (aprendizagem e percepção) e com propósitos específicos (xadrez, demonstração de teoremas matemáticos, diagnóstico de doenças) (RUSSEL, NORVIG, 2004).

As redes neurais artificiais (RNA) são um campo de estudo que encontra-se dentro da IA e são responsáveis pelo desenvolvimento de modelos de algoritmos de aprendizagem de máquina (*machine learning*) usados em aplicações de regressão, classificação e *clusterização*.

### 2.2.1 Histórico das RNA

Em 1943, McCulloch e Pitts publicaram em seu trabalho “*A Logical Calculus of the Ideas Immanent in Nervous Activity*”, a apresentação de um modelo matemático que descrevia o comportamento de um neurônio. Nesse modelo, os sinais de entrada são multiplicados por pesos sinápticos e o somatório resultante produz um valor de saída binário (FLORENCIO, VIEIRA, 2016; RUSSEL, NORVIG, 2004; TERRA, 2000; TURKIEWICZ, FRACAROLLI, 2019). Anos mais tarde, em 1949, Donald Hebb publicou o primeiro trabalho relacionado à aprendizagem de RNA, no qual o primeiro método de treinamento proposto

ficou conhecido como regra de aprendizado de Hebb (RUSSEL, NORVIG, 2004; SILVA, SPATTI, FLAUZINO, 2016).

O primeiro modelo de RNA denominado *Perceptron* foi proposto por Rosenblatt em 1958, sendo este um algoritmo de aprendizagem supervisionada que depois de treinado definia a classe de uma entrada. Logo depois em 1960, Widrow e Hoff desenvolveram o *Adaline (Adaptive Linear Element)*, modelo similar de aprendizagem supervisionada que teve como principal contribuição a aplicação da regra delta (método dos mínimos quadrados) em seu algoritmo de aprendizado (SILVA, SPATTI, FLAUZINO, 2016). O modelo *Perceptron* e o *Adaline* estimularam os estudos na área das RNA até o ano de 1969, quando a publicação do trabalho de Minsky e Papert “*Perceptrons: an introduction to computational geometry*” desacelerou o interesse pelo estudo das RNA, por demonstrar as limitações dos modelos propostos constituídos por apenas uma camada, como o *Perceptron* e o *Adaline*, em não conseguirem classificar padrões para classes não linearmente separáveis (SILVA, SPATTI, FLAUZINO, 2016; RUSSEK, NORVIG, 2004; TERRA, 2000; TURKIEWICZ, FRACAROLLI, 2019). Esse período durou até a década de 1980, quando o interesse pelas RNA ganhou novo impulso após o desenvolvimento do modelo recorrente de Hopfield (1982), a proposição do algoritmo *back propagation* desenvolvido por Rumelhart, McClelland e Williams (1986) e o desenvolvimento de computadores com maior capacidade de processamento de dados (SILVA, SPATTI, FLAUZINO, 2016; TERRA, 2000).

### **2.2.2 Neurônio biológico**

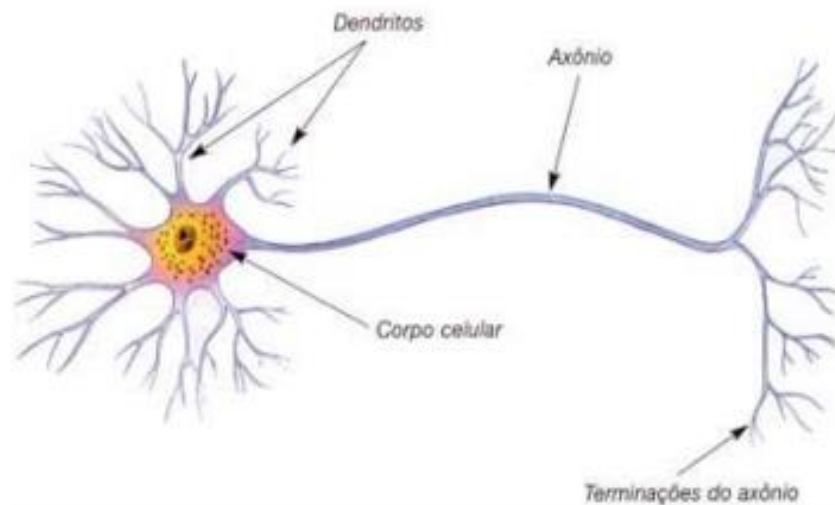
As RNA conhecidas hoje desenvolveram-se a partir dos avanços no estudo do cérebro humano. Hoje é de conhecimento que os neurônios são as unidades básicas de processamento do cérebro, atuando como elementos processadores que operam em paralelo com o objetivo de desempenhar tarefas específicas como memorizar, pensar e agir. Seu funcionamento se dá pela condução de impulsos elétricos gerados por reações físico-químicas por toda a rede neural (SILVA, SPATTI, FLAUZINO, 2016).

Cada neurônio pode ser dividido em dendritos, corpo celular e axônio como mostrado na figura 4. Os dendritos são um conjunto de terminais de entrada que captam as sinapses vindas de outros neurônios e levam até o corpo celular onde esses sinais são processados e geram um potencial de ativação na forma de impulso elétrico que se propaga ao longo do axônio. O axônio é um prolongamento fino com alta resistência elétrica e capacitância que possui terminações sinápticas por onde o impulso elétrico gerado pelo neurônio é transmitido



para os dendritos de outro neurônio, que recebe e transmite para o próximo, formando assim uma rede neural (SILVA, SPATTI, FLAUZINO, 2016; TERRA, 2000; TURKIEWICZ, FRACAROLLI, 2019). Estima-se que uma rede neural seja formada por aproximadamente 100 bilhões de neurônios, cada qual ligado em média a outros 6.000 neurônios, totalizando 600 trilhões de sinapses em uma rede neural (SHEPHERD apud SILVA, SPATTI, FLAUZINO, 2016).

Figura 4: Neurônio biológico



Fonte: BARCA, SILVEIRA, MAGINI, 2005

O cérebro possui uma inerente capacidade de aprendizado durante toda a vida. No entanto, são nos dois primeiros anos que o seu desenvolvimento é mais rápido, assimilando e respondendo aos estímulos do ambiente através da apreensão de regras, geralmente traduzido como “experiência” (HAYKIN, 2001). Hoje pode-se dizer que o funcionamento do cérebro humano possui um alto nível de complexidade, possui uma característica não linear e um sistema paralelo de processamento, sendo ele capaz de organizar sua estrutura de neurônios de tal forma que a assimilação de tarefas aconteça em apenas 1 à 2 milissegundos (*ms*) (HAYKIN, 2001).

### 2.2.3 Neurônio artificial

Segundo NG (2020), sobre a utilização de modelos de RNA baseadas no cérebro humano, destaca que “as redes neurais foram originadas como algoritmos que tentavam imitar o cérebro e da noção de que se nós quiséssemos construir sistemas que aprendem, porque não

tentar imitar o que talvez seja a máquina de aprendizagem mais incrível que conhecemos, que é o cérebro”.

Os neurônios artificiais são modelos computacionais simplificados dos neurônios biológicos (SILVA, SPATTI, FLAUZINO, 2016). Tem-se o neurônio como a unidade básica de processamento que recebe sinais de entrada, multiplica pelos pesos sinápticos, realiza a soma dos resultados e entrega um sinal de saída. As RNA são então formadas a partir da conexão de vários neurônios. De acordo com Haykin (2001), “uma rede neural é uma máquina que é projetada para modelar a maneira pela qual o cérebro executa uma tarefa ou função específica de interesse; geralmente é implementada utilizando componentes eletrônicos ou é simulada por software em computador”. Ainda segundo Haykin (2001), uma RNA possui dois aspectos semelhantes ao cérebro humano:

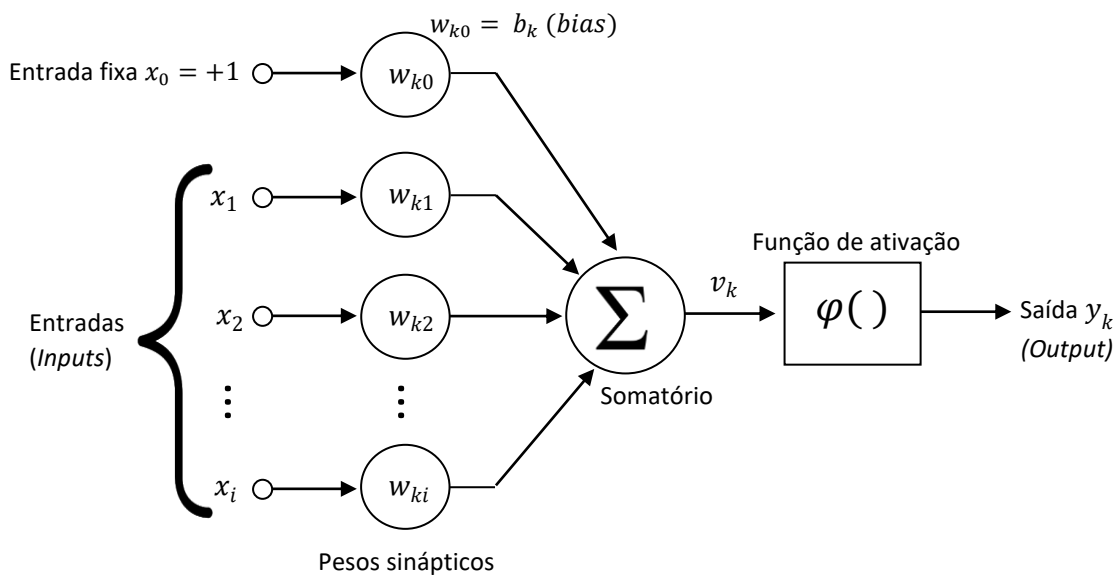
1. O conhecimento é adquirido por meio do ambiente em um processo de aprendizado;
2. As forças de conexão entre os neurônios, conhecidas como pesos sinápticos, são usadas para armazenar o conhecimento adquirido.

Os elementos básicos que compõem uma rede neural são (HAYKIN, 2001; SILVA, SPATTI, FLAUZINO, 2016):

1. Sinais de entrada ( $x_i$ ) - dados coletados do ambiente;
2. Pesos ou forças sinápticas ( $w_{ki}$ ) - pelos quais um sinal  $x_i$  de uma entrada  $i$  conectada ao neurônio  $k$  é ponderado pelo peso sináptico  $w_{ki}$ ;
3. Combinador linear ( $\Sigma$ ) - uma função somatória que agrega os sinais de entrada ponderados pelo peso sináptico  $w_{ki}$ ;
4. Limiar de ativação ou *bias* ( $b_k$ ) - uma variável que servirá como parâmetro do disparo ou não de um valor para a saída do neurônio;
5. Potencial de ativação ( $v_k$ ) - valor da diferença do combinador linear e do potencial de ativação;
6. Função de ativação ( $\varphi$ ) – utilizada para atribuir características não lineares e limitar a amplitude da saída do neurônio;
7. Sinal de saída ( $y_k$ ) - valor final produzido pelo neurônio.

A figura 5 mostra a representação do modelo de neurônio artificial com os seus elementos devidamente apresentados: os sinais de entrada  $x_i$  e saída  $y_k$ , os pesos sinápticos  $w_{ki}$ , o combinador linear  $\Sigma$ , o potencial de ativação  $v_k$ , a função de ativação  $\varphi$  e o *bias*  $b_k$ . A inserção do *bias* é acompanhada da adição de uma nova entrada  $x_0$  de valor 1 e adicionado um novo peso sináptico  $w_{k0}$  de valor igual à  $b_k$  (HAYKIN, 2001).

Figura 5: Modelo de neurônio artificial



Fonte: Adaptado de Haykin, 2001

Matematicamente a figura 5 é representada pelas equações (HAYKIN, 2000):

$$v_k = \sum_{i=1}^n w_{ki} * x_i + b_k \quad \text{Eq. 1}$$

e,

$$y_k = \varphi(v_k) \quad \text{Eq. 2}$$

Nas Eq. 1 e Eq. 2 estão presentes as seguintes variáveis:

$x_i \rightarrow$  sinais de entrada;

$w_{ki} \rightarrow$  pesos sinápticos do neurônio;

$b_k \rightarrow$  *bias*;

$v_k \rightarrow$  somatório das entradas multiplicadas pelos pesos mais o *bias*;

$\varphi \rightarrow$  função de ativação;

$y_k \rightarrow$  sinal de saída do neurônio.

O funcionamento de um neurônio artificial ocorre da seguinte forma (SILVA, SPATTI, FLAUZINO, 2016):

- Apresentação de um conjunto de valores de entrada;
- Multiplicação de cada entrada pelos pesos sinápticos;
- Obtenção do potencial de ativação pela diferença entre o combinador linear e o *bias*;
- Aplicação da função de ativação como limitador para a saída do neurônio;
- Compilação da saída do neurônio.

Normalmente, após a normalização da amplitude do sinal de saída pela função de ativação, o valor da saída encontra-se em um intervalo fechado  $[a, b]$ , como  $[0, 1]$  ou  $[-1, 1]$  (HAYKIN, 2001). Os tipos mais comuns de função de ativação são a função limite (ou degrau), a linear por partes e a sigmoideal (ou logística).

A função de limite (Eq. 3) corresponde com valor de saída 1 se  $v$  for  $\geq 0$  e, valor de saída 0 se  $v$  for  $< 0$ . A função linear por partes (Eq. 4) retorna 1 se  $v$  for  $> \frac{1}{2}$ , 0 se for  $< -\frac{1}{2}$  e o próprio valor de  $v$  caso fique entre  $\frac{1}{2}$  e  $-\frac{1}{2}$ . Por fim, a função de ativação sigmoideal (Eq. 5), a mais comumente utilizada por ser diferenciável, assumindo valores contínuos entre 0 e 1, onde  $\alpha$  é um parâmetro de declínio (HAYKIN, 2001):

$$\varphi(v) = \begin{cases} 1, & \text{se } v \geq 0 \\ 0, & \text{se } v < 0 \end{cases} \quad \text{Eq. 3}$$

$$\varphi(v) = \begin{cases} 1, & \text{se } v \geq \frac{1}{2} \\ v, & \text{se } \frac{1}{2} > v > -\frac{1}{2} \\ 0, & \text{se } v \leq -\frac{1}{2} \end{cases} \quad \text{Eq. 4}$$

$$\varphi(v) = \frac{1}{1 + e^{-\alpha v}} \quad \text{Eq. 5}$$

As Eq. 3, Eq. 4 assumem valores discretos, enquanto a Eq. 5 assume valores contínuos entre  $[0, 1]$ . Caso seja necessário que a função de ativação assumira valores entre -1 e 1, pode-

se utilizar a Eq. 6 para a função de limite e a Eq.07 (tangente hiperbólica) para a função sigmoidal (HAYKIN, 2001; SILVA, SPATTI, FLAUZINO, 2016; TERRA, 2000).

$$\varphi(v) = \begin{cases} 1, & \text{se } v > 0 \\ 0, & \text{se } v = 0 \\ -1, & \text{se } v < 0 \end{cases} \quad \text{Eq. 6}$$

$$\varphi(v) = \tanh(v) = \frac{1 - e^{-av}}{1 + e^{-av}} \quad \text{Eq. 7}$$

#### 2.2.4 Aprendizagem de máquina

Para que uma RNA desempenhe uma tarefa com eficiência e eficácia, um dos seus objetivos deve ser aprender o “modelo do mundo” e emular um modelo que seja capaz de reproduzir os resultados do mundo real. Para isso, a modelagem do algoritmo da rede deve ser capaz de armazenar conhecimento, aplicar este conhecimento para resolver problemas e adquirir novos conhecimentos a partir da experiência (HAYKIN, 2001).

O nível de aprendizado de uma RNA é dependente da qualidade dos dados de treinamento e do algoritmo de aprendizagem que ajusta os pesos das sinapses. Outra característica importante é a sua capacidade de generalização, ou seja, refere-se à produção de saídas para as entradas que não foram aplicadas durante o treinamento (HAYKIN, 2001).

Pode-se então dividir o processo de aprendizagem de máquina em três fases: aquisição de conhecimento, aprendizado e generalização.

O processo de construção do conhecimento pela rede deve (HAYKIN, 2001):

1. Conhecer o estado do mundo (*prior information*), representado pelos fatos sobre o que é e o que se sabe;
2. Observar o mundo, mensurar o ambiente por meio de sensores.

O conhecimento adquirido pela rede é dependente da análise crítica de seus dados de entrada e saída  $(x_i, y_i)$ , quanto melhor for a qualidade dos dados de entrada, maior será a precisão dos dados de saída do modelo. Para aumentar a qualidade dos dados de entrada, antes de serem utilizados para o treinamento da rede eles precisam passar por um pré-processamento (*data wrangling*) no qual será realizada a filtragem, limpeza, correção, conversão, transformação, preenchimento de dados faltantes e, caso necessário, a derivação

em novos dados. A essa atividade de realizar o pré-processamento dos dados brutos para extrair informações mais relevantes é conhecida como engenharia de atributos (SILVA, 2019).

Primeiramente, deve-se definir quais são os dados de interesse para o modelo que precisam ser coletados. Para isso, fazer o levantamento das características do objeto de estudo para o qual deseja-se desenvolver o algoritmo de aprendizagem pode ajudar a definir o rumo inicial. Em seguida, definir a fonte onde os dados serão coletados. Muitas vezes será necessário realizar a combinação de dados de diferentes fontes para compor ou complementar um conjunto.

Após a fase de coleta, é realizada uma filtragem nos dados coletados com o objetivo de selecionar apenas aqueles registros que são de interesse, pois geralmente a coleta resulta em dados brutos e podem incluir valores repetidos ou redundantes para o modelo. A utilização de um número excessivo de atributos tornam os modelos muito complexos e podem resultar em sobreajuste (SILVA, 2019). O sobreajuste diminui a capacidade de generalização do modelo.

A conversão também pode ser necessária em casos onde a ferramenta utilizada para desenvolver o algoritmo não reconheça certos formatos, como na utilização de datas na sua forma normal de escrita (p. ex. 01/12/20) pode não ser reconhecida pelo programa (SILVA, 2019).

Na atividade de transformação há dados que se encontram em formato de texto, unidades de medida ou escalas muito diferentes. Algumas técnicas aplicadas à transformação de atributos são a binarização que define o valor do atributo como sendo 0 ou 1, a quantização que discretiza valores contínuos, a transformação logarítmica que serve para suavizar os extremos e aumentar a correlação dos dados (Eq.08), a normalização min-max capaz de colocar a escala do atributo dentro de um intervalo  $[a, b]$  (Eq.09) e a padronização que utiliza a média e o desvio padrão (Eq.10) (SILVA, 2019).

$$y = \log_b(x) \quad \text{Eq. 08}$$

$$x = \frac{x_i - \min(x)}{\max(x) - \min(x)}(b - a) + a \quad \text{Eq. 09}$$

Onde:

$x_i$  → valor do atributo a ser normalizado;  
 $\min(x)$  → valor mínimo de todo o conjunto do atributo;  
 $\max(x)$  → valor máximo de todo o conjunto do atributo;  
 $a$  → limite mínimo do intervalo de normalização;  
 $b$  → limite máximo do intervalo de normalização;  
 $x$  → novo valor do atributo normalizado.

$$x_i = \frac{x_i - \bar{x}}{\sigma_x} \quad \text{Eq. 10}$$

Onde:

$\bar{x}$  → média de  $x$ ;  
 $\sigma_x$  → desvio padrão de  $x$ .

Na derivação, novos atributos podem ser criados da combinação dos atributos existentes, como por exemplo, combinar um atributo de largura e outro de comprimento e criar um novo atributo relativo à área. Nem todos os atributos de alguns dados coletados podem estar devidamente preenchidos, sendo necessário fazer o seu preenchimento ou a remoção destes (SILVA, 2019).

Dentre os principais benefícios da aplicação da engenharia de atributos é possível destacar a melhora na representação dos dados no conjunto de treinamento, a utilização de qualquer tipo de dado, maior compreensão da tarefa, o aumento do desempenho do modelo, a elaboração de modelos mais simples que atinjam o objetivo ou mais complexos para um aprendizado mais profundo do problema, dentre outros (SILVA, 2019).

Após a fase de coleta, tratamento e seleção, os dados são inseridos no algoritmo de aprendizado para a fase de treinamento.

No treinamento, cada par de vetores  $(x_i, y_i)$  é apresentado ao modelo para que este aprenda a associação entre os vetores e realize o ajuste dos pesos sinápticos para que produza as saídas desejadas (SILVA, 2019). Após receber o treinamento, o modelo será usado para prever novas saídas a partir de dados de entradas não utilizados no treinamento, ou seja, nessa fase é verificado se com o que o modelo aprendeu durante a fase anterior ele é capaz de generalizar para qualquer outros dados novos fornecidos (SILVA, 2019).

Para o desenvolvimento de uma RNA são usados métodos de aprendizagem. Um deles é o método dos mínimos quadrados (Regra Delta) elaborado por Widrow e Hoff (1960) que tem a função de ajustar os pesos sinápticos após cada passo do treinamento (TURKIEWICZ, FRACAROLLI, 2019). Outro método de aprendizagem baseada na lei de Hebb (1949), dois neurônios simultaneamente ativos tem a força sináptica aumentada (TURKIEWICZ, FRACAROLLI, 2019). O Método KNN (*K Nearest Neighbor*) ou método do vizinho mais próximo, pelo qual o processo de treinamento do algoritmo é baseado no k-ésimo vizinho mais próximo de acordo com uma medida de similaridade que pode ser calculada, por exemplo, por meio de distância euclidiana (SILVA, 2019). Pelo método da aprendizagem competitiva, onde os neurônios competem entre si e o neurônio “vencedor” é o único a ter os seus pesos ajustados (SILVA, SPATTI, FLAUZINO, 2016). Ou ainda por meio da aprendizagem de Boltzmann ou máquina de Boltzmann, que possui um sistema de camadas binárias visíveis e outro de camadas binárias escondidas e é aplicado em problemas de otimização (VALE, 2016).

O aprendizado pode ser realizado em lote (*offline*) ou continuado (*online*). Por meio do aprendizado em lote de dados (*offline*), o algoritmo somente ajusta seus pesos e limiares após todo o conjunto de treinamento ter passado por ele. É denominada “época” de treinamento a quantidade de vezes que os dados passam pela rede e conseqüentemente ajusta os pesos e limiares de ativação (SILVA, SPATTI, FLAUZINO, 2016). Sua principal desvantagem é que ao longo do tempo o seu desempenho pode perder a eficiência sendo necessário treiná-lo novamente (SILVA, 2019). No aprendizado realizado de forma contínua (*online*), após receber o treinamento inicial e entrar em funcionamento, as saídas que o algoritmo produz são usadas como novos dados de treinamento, permitindo que o processo de aprendizagem seja realizado continuamente durante todo o período de execução do algoritmo. As principais desvantagens da utilização desse método são que a rede só começará a fornecer respostas precisas quando uma grande quantidade de amostras de dados são apresentadas à ela e a sua exposição à dados ruins que podem induzir o algoritmo ao erro (SILVA, 2019).

Quanto ao sentido do processamento, as RNAs podem ser alimentadas à frente (*feedforward*) onde o fluxo das informações segue em sentido único da entrada para a saída da rede, ou de maneira recorrente (*feedback*) quando pelo menos uma saída é reintroduzida na rede (FLORÊNCIO, VIEIRA, 2016).

O algoritmo *back propagation* utilizado para treinamento de redes neurais é dividido em duas fases. O fluxo de informação advindo da camada de entrada, passando para as camadas escondidas e depois para a camada de saída é chamada de fase de propagação para a



frente (*forward*). A informação processada nesse percurso produz a predição de uma saída desejada. Na segunda fase o fluxo da informação segue para trás (*backward*), propagando o erro calculado e fazendo o ajuste dos pesos (TERRA, 2000).

O algoritmo *back propagation* utiliza o método do gradiente descendente (Eq.11) para otimizar os pesos e reduzir o erro. Utiliza-se uma constante  $\eta$  de proporcionalidade de valor compreendido no intervalo  $[0, 1]$  (SILVA, SPATTI, FLAUZINO, 2016).

$$w_{ki} \leftarrow w_{ki} + \eta \cdot \delta_k \cdot y_i \quad \text{Eq. 11}$$

Onde:

$w_{ki}$  → pesos sinápticos do neurônio

$\eta$  → constante de proporcionalidade ou taxa de aprendizagem

$\delta_k$  → gradiente local em relação ao  $k$ -ésimo neurônio

$y_i$  → valor da saída real

Após o treinamento da rede, é apresentada ao conjunto de dados de teste e utilizará somente a propagação para a frente (*forward*), pois os pesos já estão ajustados (TERRA, 2000).

A avaliação do desempenho do modelo pode ser realizada pela verificação do erro entre a saída predita e a saída real durante a fase de treinamento. Essa verificação de desempenho pode ser feita utilizando-se duas métricas conhecidas: *Mean Absolute Error* (MAE) e *Root Mean Squared Error* (RMSE). A principal diferença entre as duas métricas é que a MAE (Eq.12) atribui o mesmo peso a todos os erros, enquanto a RMSE (Eq.13), por elevar o erro ao quadrado, penaliza erros grandes cometidos pelo modelo (SILVA, 2019).

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - y_k| \quad \text{Eq. 12}$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (y_i - y_k)^2}{n}} \quad \text{Eq. 13}$$

Onde:

$n$  → quantidade de dados;

$y_i \rightarrow$  valor da saída real;

$y_k \rightarrow$  valor da saída do neurônio.

Na literatura, alguns autores consideram como sendo aceitável para um modelo o erro de 4% para a fase de treinamento e 12% para a fase de teste (CHEN, YIH, 1996 apud TERRA,2000), enquanto outros consideram 7% para o treinamento e 17% para os testes (SAAD, 1996 apud TERRA,2000).

### 2.2.5 Tipos de aprendizagem

Os modelos de RNA podem ser classificados em dois tipos de aprendizagem: supervisionada e não supervisionada.

De maneira simples, a aprendizagem supervisionada possui um conjunto de pares de treinamento composto por vetores de entrada ( $x_i$ ) e saída ( $y_i$ ), no qual para cada entrada existe uma saída correspondente com o valor desejado. Durante o treinamento, o erro entre a saída desejada e a saída do algoritmo é usado como parâmetro de ajuste dos pesos sinápticos para a próxima iteração. Por meio então de uma grande quantidade de exemplos de entrada e saída, o algoritmo busca então encontrar uma função de aproximação que associe os vetores de  $x_i$  com  $y_i$ , que posteriormente será usada para prever uma saída  $y$  desconhecida de uma entrada  $x$  (FLORENCIO, VIEIRA, 2016; SILVA, 2019; TERRA, 2000; TURKIEWICZ, FRACAROLLI, 2019). A rede é considerada treinada quando o erro entre a saída predita e a real estiver dentro de padrões aceitáveis. Normalmente 60 a 90% do total de dados disponíveis são reservados para o treinamento, enquanto que o restante de 10 a 40% são usados para testar a capacidade de generalização do modelo (SILVA, SPATTI, FLAUZINO, 2016).

O aprendizado supervisionado pode ser subdividido em classificação e regressão. O aprendizado supervisionado feito para realizar uma tarefa de classificação busca classificar uma saída discreta, de valores reais ou inteiros, dentro de um conjunto finito de categorias pré-determinado. No modelo de regressão, as variáveis preditoras (entradas), estimam um valor de saída contínuo (SILVA, 2019).

O aprendizado de máquina semi-supervisionado, geralmente tratado na literatura também como um aprendizado supervisionado por conta de seu objetivo final ser o mesmo (a partir dos dados de entrada  $x_i$  prever uma saída  $y_i$  desconhecida) encontra-se em um cenário onde parte dos dados de entrada  $x_i$  não estão completos. Essa situação é encontrada nos casos

em que o custo e o tempo gasto para preencher todos os dados de entrada são muito altos (SILVA, 2019), o algoritmo então é escrito de forma a conseguir contornar esse problema, funcionando normalmente para os exemplos totalmente rotulados e nos parcialmente rotulados faz a inserção dos dados faltantes via algoritmos estimadores de verossimilhança.

No treinamento por aprendizagem não supervisionada existem apenas os vetores de entrada, inexistindo no conjunto de dados a saída desejada. A rede deve então se auto organizar para encontrar similaridades entre os dados de entrada e produzir as saídas para o modelo. Este tipo de aprendizagem pode ser subdividido em *clusters*, no qual o modelo busca agrupar os dados em diferentes classes por meio da identificação de padrões de similaridade existentes, ou aprendizado de redução de dimensionalidade cujo objetivo é reduzir a quantidade de atributos sem que as principais características iniciais sejam alteradas (SILVA, SPATTI, FLAUZINO, 2016; SILVA, 2019).

Existe ainda o aprendizado por reforço, no qual o algoritmo não recebe nenhum conjunto de dados para realizar o treinamento inicial e o seu aprendizado se dá por meio da interação com o ambiente e por um conjunto de regras e estratégias previamente estabelecidas. A cada iteração o algoritmo recebe uma resposta positiva ou negativa até que este aprenda o suficiente para fornecer as respostas desejadas, sendo assim o processo de treinamento baseado por tentativa e erro (SILVA, SPATTI, FLAUZINO, 2016; SILVA, 2019).

### 2.2.6 Modelos de arquitetura de RNA

A modelagem de uma rede neural deve levar em consideração a estrutura dos neurônios e o algoritmo de aprendizado utilizado para treinar a rede.

Uma dessas arquiteturas é a *Single-Layer Feedforward* ou rede *Feedforward* de Camada Simples, na qual a única camada de neurônios é a própria saída, o treinamento é feito utilizando a regra de Hebb, a regra Delta e a alimentação é *feedforward*, ou seja, alimentação para à frente em uma única direção, da entrada para a saída da rede. Esta arquitetura recebe este nome por conta da camada de entrada não ter neurônios (não realiza nenhum cálculo), sendo a camada de saída considerada a “camada simples”. O modelo *Perceptron* e o *Adaline* possuem esta arquitetura (HAYKIN, 2001; SILVA, SPATTI, FLAUZINO, 2016).

Uma das arquiteturas mais usadas em RNAs é a *Multi-Layer Feedforward*, na qual além da camada de saída, a rede contém uma ou mais camadas escondidas (ocultas). Os neurônios das camadas escondidas possuem uma alta capacidade estatística conseguindo

encontrar associações entre os dados de entrada e saída que uma rede de camada simples não conseguiria. Nesta rede todos os sinais de entrada são conectados aos neurônios da primeira camada à frente (*feedforward*) que produz uma saída para cada neurônio, as quais são usadas como entradas da próxima camada à frente, que pode ser já a camada de saída ou outra camada escondida. As principais redes neurais que utilizam essa estrutura são o *Multi-Layer Perceptron* (MLP) e o *Radial Basis Function* (RBF) (HAYKIN, 2001; SILVA, SPATTI, FLAUZINO, 2016; TERRA, 2000).

Outra arquitetura existente se refere às chamadas redes recorrentes, nas quais a principal diferença é que a informação não segue em uma única direção à frente (*feedforward*), a informação gerada por uma saída retroalimenta (*feedback*) a rede pelo menos uma vez. Entre as redes que são compostas nessa arquitetura estão a rede de Hopfield e o MLP com realimentação (HAYKIN, 2001; SILVA, SPATTI, FLAUZINO, 2016).

### 2.2.7 Conclusão da revisão de literatura sobre RNA

Resumindo até o momento, o sistema de funcionamento do neurônio artificial é baseado no funcionamento do neurônio biológico, conforme apresentado na tabela 1.

Tabela 1: Similaridade do funcionamento entre o neurônio biológico e o artificial

Etapa	Neurônio Biológico	Neurônio Artificial
1	Sinapses chegam aos dendritos	Dados inseridos na camada de entrada
2	Corpo celular processa o sinal sináptico	Somatório das entradas ponderadas pelos pesos sinápticos
3	Impulso elétrico é enviado pelo prolongamento do axônio	Calculado o potencial de ativação
4	Sinapse gerada é enviada para o próximo neurônio	Saída (resposta) do neurônio

Fonte: O autor

Uma RNA é formada por um conjunto de neurônios artificiais organizados na forma de camada simples (*Single-Layer*) ou camadas múltiplas (*Multi-Layer*), com o fluxo de informação unidirecional (*feedforward*) ou com realimentação (*feedback*). As saídas produzidas pelos neurônios de uma camada são todas transmitidas aos neurônios da camada seguinte e assim por diante até que se chegue à última camada e se obtenha a resposta final.

Existem dois tipos de aprendizado de máquina, a supervisionada (regressão e classificação) e a não supervisionada (*clusters*, redução de dimensionalidade e por reforço).

Para garantir uma resposta final coerente e confiável, a RNA de aprendizado supervisionado deve ser treinada e testada com um conjunto de pares de entradas e saídas desejadas, as quais sua qualidade pode ser aumentada por meio do processo de engenharia de atributos. A divisão recomendada para o conjunto de dados é de 60 à 90% do total reservado para a fase treinamento e o restante para fase de teste.

Durante o treinamento ocorre o ajuste dos pesos sinápticos por meio de um algoritmo de aprendizagem (Regra Delta, Lei de Hebb, *backpropagation*, KNN, aprendizagem competitiva, máquina de Boltzmann).

Depois de treinado o algoritmo está pronto para ser testado. A fase de teste consiste em fornecer à RNA dados que não foram utilizados na fase de treinamento para avaliar o aprendizado resultante. A avaliação do desempenho da RNA pode ser medida por meio da aplicação de duas métricas: MAE e RMSE.

Neste trabalho o modelo de RNA construído utilizará as características apresentadas na tabela 2.

Tabela 2: Características da RNA do trabalho

<b>Característica:</b>	<b>Modelo:</b>
Aprendizagem	Supervisionada
Tarefa de aprendizagem	Regressão
Arquitetura	MLP
Alimentação	<i>Feedforward</i>
Algoritmo de aprendizagem	<i>Back propagation</i>
Apresentação dos dados	Em lote ( <i>offline</i> )
Medição do erro	MAE e RMSE

Fonte: O autor

Pelo conjunto de dados usados terem definidas as entradas e saídas, o modelo é classificado como tendo uma abordagem supervisionada de regressão. A arquitetura escolhida foi a MLP com o objetivo de se obter uma maior correlação entre os atributos de entrada e saída. O fluxo de alimentação será *feedforward* e a apresentação dos dados em lote por serem menos complexas. Foi definido o *back propagation* como algoritmo de aprendizagem por utilizar o método de descida do gradiente descendente capaz de obter um bom ajuste dos pesos pelo cálculo iterativo do mínimo de uma função. E por fim, a avaliação do desempenho do modelo será avaliado pela medição do erro percentual calculado pelas funções MAE e RMSE.

### 3 PROCEDIMENTOS METODOLÓGICOS

No presente capítulo são descritos os procedimentos metodológicos adotados no trabalho, definidos a partir do tema, dos objetivos gerais e específicos, do levantamento sistemático da literatura, do desenvolvimento do modelo, até chegar aos resultados alcançados.

#### 3.1 DEFINIÇÃO DO TEMA E DOS OBJETIVOS DO TRABALHO

A definição do tema deste trabalho foi norteada pelos requisitos da indissociabilidade, interdisciplinaridade e interprofissionalidade entre *Ensino – Pesquisa – Extensão*. A indissociabilidade é relativa à algo que não existe sem a presença do outro, a falta de uma das partes (ensino, pesquisa ou extensão) não permite a completa constituição do todo, sendo este todo a relação entre a instituição de ensino, o graduando e a sociedade. A interdisciplinaridade refere-se a interligação e interdependência entre as disciplinas, sendo que uma é dependente da outra, não sendo possível a sua completa existência isoladamente. A interprofissionalidade assemelha-se à interdisciplinaridade, com a diferença de a interligação e interdependência estarem relacionadas ao trabalho ou a profissão.

Desse modo, o objetivo geral de criar um modelo de RNA que auxilie o departamento de PCP na otimização da atividade de Programação da Produção através da predição da quantidade necessária de *inputs* a serem transformados, para que ao final de todas as etapas do processo produtivo a variação entre a quantidade de itens bons do lote entregue e a quantidade solicitada pelo cliente seja a menor possível, enquadrando-se na proposta de manter uma relação entre a instituição de ensino, o graduando e a sociedade, envolvendo nesse processo de pesquisa científica as diversas áreas da engenharia de produção (processos industriais, cálculo numérico, planejamento e controle da produção, pesquisa operacional, modelagem de problemas de engenharia, computação, sistemas de informação) e as profissões do ambiente industrial.

A formulação do escopo do estudo foi completada com a definição dos objetivos específicos, derivados do objetivo geral.

### 3.2 LEVANTAMENTO SISTEMÁTICO DA LITERATURA

O levantamento sistemático da literatura do presente trabalho ocorreu inicialmente pela busca de artigos, textos, livros e trabalhos de graduação relacionados ao tema nas bases de dados da ABEPRO, Capes, Google Scholar, Scielo, Science.gov e Science Research. As palavras-chave utilizadas nas buscas foram combinações de “rede neural artificial”, “*machine learnig*”, “engenharia de produção”, “indústria”, “planejamento e controle de produção”, “administração da produção” e “pesquisa operacional”. Os resultados obtidos da pesquisa foram filtrados por meio da análise de seus respectivos resumos e classificados em ordem de relevância em uma escala de 1 a 10, enquanto outros foram descartados.

Em paralelo com a leitura dos textos foi realizado o fichamento das fontes e a tradução de material publicado em língua estrangeira para posterior construção do referencial teórico deste trabalho.

### 3.3 DESENVOLVIMENTO DO MODELO

O desenvolvimento do modelo de RNA começou com a formulação de uma hipótese de modelo que seria trabalhado para na sequência se iniciar a coleta de dados através do histórico de pedidos no sistema ERP (*Enterprise Resource Planning*) da empresa. Os dados obtidos receberam tratamento conforme a engenharia de atributos vista na literatura. Logo na sequência foi possível definir o modelo de RNA mais adequado para com as características do problema e dos dados coletados, sendo escolhido o modelo MLP de aprendizagem supervisionada de regressão com o algoritmo *back propagation* para ajuste dos pesos sinápticos.

O software utilizado foi o Anaconda 3 versão 2020.11 (64-bit) para Windows, escolhido por ser gratuito, conter o programa Spider com a linguagem de programação Python (versão 3.8.5) e um grande número de bibliotecas inclusas.

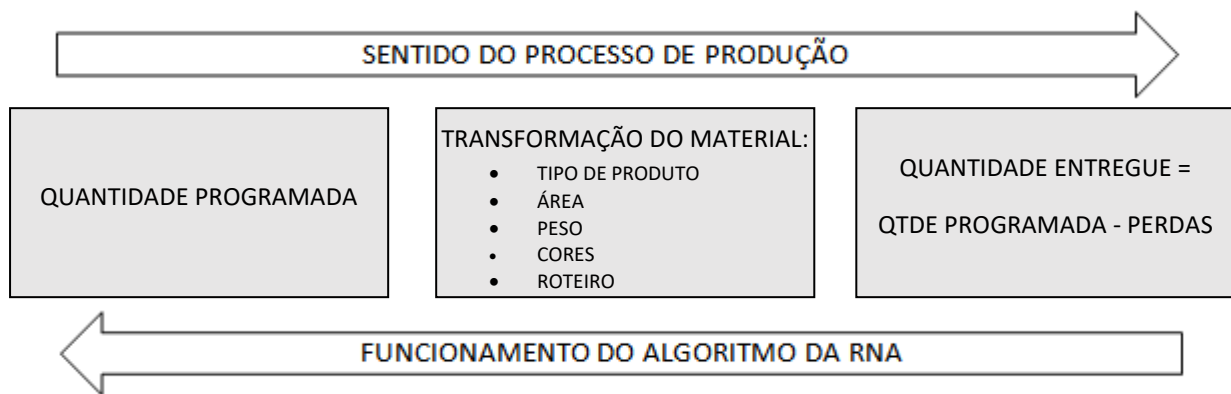
O algoritmo da RNA foi escrito e treinado em Python com 70 % do total de dados disponíveis usado para treinamento e o restante usado para teste.

### 3.4 FUNCIONAMENTO ALGORITMO

Primeiramente, para melhorar a compreensão da resolução do problema pelo uso da RNA proposta neste trabalho, existem dois dados que se encontram em extremos opostos, um

no início e outro no final do processo, que são a quantidade programada e a quantidade entregue. Como mostrado na figura 6, a ideia base do funcionamento do modelo é fazer o sentido inverso do fluxo de produção, ou seja, enquanto o processo de produção segue pela (1) definição da quantidade programada → (2) transformação do material → (3) quantidade entregue, por sua vez a RNA é executada a partir da (1) quantidade entregue → (2) compilação do algoritmo → (3) definição da quantidade programada.

Figura 6: Sentido do fluxo de produção e do algoritmo da RNA



Fonte: O autor

Em segundo lugar, para a resolução do problema de pesquisa existem três fases no processo para validação da RNA como ferramenta: treinamento, simulação e aplicação prática.

Na fase de treinamento, dados de entrada e de saída serão apresentados ao algoritmo para que este aprenda a sua correlação.

Ainda utilizando dados históricos, inicialmente para a fase de simulação apenas os dados de entrada são fornecidos à RNA e o seu objetivo é predizer um valor de saída relativo à quantidade necessária de material que deverá ser programada para que após retirarem-se as perdas do processo (itens com defeito) a empresa consiga entregar a quantidade pedida pelo cliente. Depois de se obterem as saídas, o erro da simulação é calculado comparando-se a saída predita com o valor real. As principais vantagens da etapa de simulação são:

- O ajuste fino da RNA pode ser feito até que o erro seja minimizado à valores aceitáveis;
- O algoritmo pode ser considerado “aplicado” em uma situação real, pois basta não informar a resposta (saída correta) e depois que ele predizer sua própria saída compará-los;



- Os riscos e as desvantagens de se aplicar uma nova ferramenta que não funcione corretamente são minimizados.

Por fim, após treinar e validar o modelo por simulações, a RNA pode ser utilizada como ferramenta de auxílio à tomada de decisão e aplicada na prática em novos casos bastando fornecer os dados de entrada para a obtenção de uma resposta.

### 3.5 VERIFICAÇÃO DO MODELO

A verificação das respostas fornecidas pelo modelo foi feita de forma simples, através da escolha de respostas aleatórias e de medição do erro, foram consultados os valores originais de entradas e saídas no conjunto de dados e calculados manualmente para atestar a coerência do modelo. Nesta etapa também foi avaliada a influência dos atributos de entrada escolhidos no resultado fornecido pelo modelo.

Os métodos MAE e RMSE foram as métricas usadas para calcular a variação do erro percentual dos resultados nas fases de treinamento e de teste. Logo em seguida, as mesmas métricas foram usadas para calcular a variação do erro do método atual utilizado pelo setor de PCP para a programação da produção.

Os resultados obtidos pelos métodos do modelo de RNA e o atual foram confrontados sob a perspectiva de sete cenários diferentes de demanda, sendo para cada um destes analisada a relação entre a quantidade programada e a demanda atendida.

Por fim, com base nos resultados preditos pelo modelo de RNA, concluiu-se o estudo com as considerações finais e propostas para trabalhos futuros.

## 4 DESENVOLVIMENTO DO MODELO

O presente capítulo tem por objetivo descrever as atividades realizadas durante a fase de desenvolvimento do modelo de RNA. Um resumo das etapas, suas atividades e o tempo despendido para a realização das mesmas são apresentados na tabela 3.

Tabela 3: Etapas e atividades de desenvolvimento do modelo de RNA

<b>Etapa</b>	<b>Atividade</b>	<b>Tempo (%)</b>
Planejamento	Formular uma hipótese inicial de modelo	2,0
	Escolher o modelo de RNA	1,0
	Definir os atributos dos dados	2,0
Preparação dos dados	Coletar os dados	40,0
	Realizar a engenharia de atributos nos dados	30,0
	Classificar e exportar os dados	10,0
Modelagem do algoritmo da RNA	Escrever o algoritmo da RNA	5,0
	Aperfeiçoar o algoritmo da RNA	2,0
	Treinar o algoritmo da RNA	1,0
	Testar o Algoritmo da RNA	1,0
	Calcular o erro da RNA	1,0
	Plotar os gráficos	1,0
Cálculo do erro do método atual de programação	Escrever o algoritmo para o método atual de programação	2,0
	Calcular o erro do método atual	1,0
	Plotar os gráficos do método atual	1,0

Fonte: O autor

Pela tabela 3 observa-se que a maior parte do tempo foi gasto com o a preparação dos dados (80,0%), enquanto as outras três etapas despenderam menor tempo de execução (20%). As etapas de desenvolvimento do modelo são detalhadas a seguir.

### 4.1 PLANEJAMENTO

O planejamento do modelo envolveu basicamente três atividades: Formular uma hipótese inicial de modelo, escolher o modelo de RNA e definir os atributos dos dados.

A formulação da hipótese inicial ocorreu após um maior entendimento do problema de pesquisa e do estudo da literatura sobre as RNA. A hipótese inicial foi definida como sendo composta de um conjunto de entradas onde a maior parte dos atributos seriam binários, porém

logo depois a ideia foi deixada de lado por conta de exigir um grande número de dados de entrada para cada amostra, o que representaria um custo computacional mais elevado. A segunda hipótese reduziu o número de variáveis binárias e assumiu maior quantidade de valores discretos e contínuos conforme a tabela 4.

Tabela 4: Hipóteses formuladas para o modelo

Atributos	Hipótese I		Hipótese II	
	Tipo	Nº Variáveis	Tipo	Nº Variáveis
Lote do pedido	Discreto	1	Discreto	1
Lote entregue	Discreto	1	Discreto	1
Tipo de produto	Binário	1	Binário	1
Área	-	-	Contínuo	1
Largura	Binário	16	-	-
Comprimento	Binário	15	-	-
Peso	Binário	24	Contínuo	1
Cores frente	Binário	8	Discreto	1
Cores verso	Binário	8	Discreto	1
Máquina 1	Binário	1	Binário	1
Máquina 2	Binário	1	Binário	1
Máquina 3	Binário	1	Binário	1
Máquina 4	Binário	1	Binário	1
Lote programado	Discreto	1	Discreto	1
<b>Total variáveis</b>		<b>79</b>		<b>12</b>

Fonte: O autor

Na hipótese I, há a necessidade de relacionar cada possibilidade de um recurso a uma entrada binária, por exemplo, existem 8 possibilidades de quantidades de cores frente (1, 2, 3, 4, 5, 6, 7, e 8) e cada uma é relacionada à um binário de verdadeiro ou falso, ou seja, para extrair a informação da quantidade de cores frente usadas na amostra, o algoritmo precisa relacionar 7 entradas como sendo falsas e 1 como verdadeira. Na hipótese II, a quantidade da mesma variável é reduzida para 1, pois a entrada simplesmente recebe a informação “verdadeira”.

A atividade de escolher o modelo de RNA que seria implementada foi definida rapidamente, pois além de todas as entradas e saídas desejadas estarem à disposição do problema, a saída que o modelo deve predizer é uma variável contínua, caracterizando um aprendizado supervisionado que utiliza a regressão. Entre os modelos de regressão apresentados na revisão de literatura foi escolhido o MLP.

A definição dos atributos de entrada do modelo foi feita por meio de uma listagem de características importantes do produto e do processo que teriam maior capacidade de influência nas perdas, ou seja, características com potencial de ocasionarem certa quantidade itens defeituosos. Os atributos de entrada e saída escolhidos são os que compõem a hipótese II na tabela 4.

Na sequência foi realizada a engenharia de atributos, atividade na qual foi realizada a filtragem dos dados buscando analisar e retirar do conjunto as amostras que poderiam influenciar na má qualidade dos dados e reduzir o desempenho do modelo, como por exemplo, dados *outliers*.

#### 4.2 PREPARAÇÃO DOS DADOS

A atividade de preparação dos dados demandaram maior tempo de trabalho que as outras. Primeiramente a coleta dos dados envolveu a combinação de diferentes relatórios de produção para que fosse possível obter os atributos de entrada e saída definidos anteriormente. Também foi realizado a transformação dos dados para a mesma unidade de medida com o propósito de aumentar a homogeneidade do conjunto. Os atributos de entrada “largura” e “comprimento” foram combinados acabando por derivar um novo atributo de entrada denominado “área”. Por fim, houve o preenchimento dos dados que estavam incompletos. O total final de amostras válidas coletadas foi de 2.602, dispostas em uma matriz de (2.602, 11) para as entradas  $x$  e em um vetor (2.602, 1) para as saídas  $y$ .

Com a coleta e engenharia dos dados finalizados, eles foram divididos em sete conjuntos de amostras conforme a tabela 5.

Tabela 5: Divisão do conjunto de dados conforme a quantidade

Conjunto	Quantidade de itens do pedido	Quantidade de amostras
1	Geral (Todas as qunatidades)	2.602
2	Até 10 mil unidades	1.192
3	De 10 à 20 mil unidades	1.015
4	De 20 à 30 mil unidades	208
5	De 30 à 40 mil unidades	75
6	De 40 à 50 mil unidades	51
7	Acima de 50 mil unidades	61

Fonte: O autor

Os conjuntos foram então exportados para o formato de arquivo csv (*comma-separated-values* ou valores separados por vírgula).

#### 4.3 MODELAGEM DO ALGORITMO DA RNA

A modelagem do algoritmo da RNA foi feita por meio do software Spyder utilizando a linguagem de programação python. O algoritmo desenvolvido encontra-se no Apêndice A e os blocos de códigos estão dispostos na seguinte ordem:

- Importação das bibliotecas – pandas, numpy, matplotlib;
- Carregamento do arquivo de dados – arquivo csv;
- Separação dos dados em variáveis de entrada e saída – entradas (quantidade pedida, produto, área, peso, quantidade de cores na frente, quantidade de cores no verso, máquina 1, máquina 2, máquina 3, máquina 4) e saída (quantidade programada);
- Divisão dos conjuntos de treino e teste – 70% para treino e 30% para teste;
- Parametrização do modelo MLP – 2 camadas ocultas, 1ª camada com 50 e 2ª camada com 100 neurônios, função de ativação *relu*, otimizador dos pesos sináticos *lbfgs*;
- Impressão dos resultados – saídas do modelo;
- Cálculo do erro utilizando MAE – erro percentual (%) do treinamento e do teste;
- Cálculo do erro utilizando RMSE – erro percentual (%) do treinamento e do teste;
- Plot do gráfico – gráfico de dispersão e Boxplot;
- Salvamento do gráfico – arquivo png;
- Inserção de novos dados e verificação da sensibilidade do algoritmo – aplicação do algoritmo treinado para predição da quantidade a ser programada;
- Resultado dos novos dados – saídas do modelo.

#### 4.4 CÁLCULO DO ERRO DO MÉTODO ATUAL DE PROGRAMAÇÃO

Após escrever o algoritmo do modelo de RNA para o problema de pesquisa, foi escrito o algoritmo para calcular o erro MAE e RMSE do método atual (5, 10 e 15%) de programação, também escrito em python. O modelo desenvolvido para testar o método atual encontra-se no Apêndice B e está disposto na seguinte sequência:

- Importação das bibliotecas – pandas, numpy, matplotlib;
- Carregamento do arquivo de dados – arquivo csv;
- Separação dos dados em variáveis de entrada e saída – entradas (quantidade pedida) e saída (quantidade programada);
- Cálculo do erro utilizando MAE – erro percentual (%) do método atual;
- Cálculo do erro utilizando RMSE – erro percentual (%) do método atual;
- Plot do gráfico – gráfico de dispersão e Boxplot;
- Salvamento do gráfico – arquivo png.

## 5 RESULTADOS

Nesta seção serão apresentados os resultados obtidos do teste de sensibilidade da RNA frente aos atributos de entrada, o erro percentual calculado utilizando as métricas MAE e RMSE e a comparação dos resultados alcançados pelo modelo com o método atual utilizado pelo PCP da empresa que utiliza valores fixos de 5, 10 e 15% para programar a produção.

### 5.1 SENSIBILIDADE DA RNA

Primeiramente, após o treinamento da RNA foi analisada a sensibilidade do modelo frente aos dados de entrada com o objetivo de verificar se os atributos escolhidos exerciam influência na predição das saídas. Os resultados obtidos são apresentados na tabela 6, onde a primeira linha corresponde aos dez atributos escolhidos, a primeira coluna da esquerda enumera os dez cenários simulados e a última coluna da direita apresenta o resultado da saída da rede.

Tabela 6: Resultados da simulação da RNA

	Pedido	Produto	Área	Peso	C.Frente	C.Verso	Máq.1	Máq.2	Máq.3	Máq.4	Saída
m1	10.000	1	0,749	0,075	1	1	0	0	0	1	10.543
m2	15.000	1	0,749	0,075	1	1	0	0	0	1	15.739
m3	10.000	0	0,749	0,075	1	1	0	0	0	1	10.528
m4	10.000	1	1,749	0,075	1	1	0	0	0	1	10.556
m5	10.000	1	0,749	0,575	1	1	0	0	0	1	10.543
m6	10.000	1	0,749	0,075	4	4	0	0	0	1	10.839
m7	10.000	1	0,749	0,075	1	1	1	0	0	1	10.548
m8	10.000	1	0,749	0,075	1	1	0	1	0	1	10.540
m9	10.000	1	0,749	0,075	1	1	0	0	1	1	10.557
m10	10.000	1	0,749	0,075	1	1	0	0	0	0	10.469

Fonte: O autor

O resultado da primeira amostra (m1) ficou sendo a saída padrão para este conjunto de validação, as demais simulações foram feitas de modo a alterar um atributo de cada vez e o resultado obtido comparado com a saída padrão. Se a quantidade sofrer alteração, significa que o atributo possui potencial para gerar perdas durante o processo, caso contrário ele pode ser descartado do conjunto de dados sem prejudicar a predição do modelo.

De todas as simulações realizadas, o único atributo que mostrou não exercer influência no resultado final foi o atributo peso, o qual foi alterado na simulação do cenário m5 e não resultou em alteração na quantidade da saída quando em comparação com a saída padrão.

## 5.2 RESULTADOS PARA O CONJUNTO GERAL

O conjunto geral dos dados é referente à todos os dados coletados, ou seja, abrange todas as amostras do histórico de pedidos.

Os principais resultados obtidos pela RNA para este conjunto de treinamento e de teste estão apresentados na tabela 7 juntamente com os dados calculados para o método atual de programação (5, 10 e 15%) para comparação.

Tabela 7: Resultados do conjunto geral de dados

<b>Resultado</b>	<b>RNA</b>	<b>Modelo atual</b>
Número total de amostras de treino	1.821	-
Número total de amostras de teste	781	-
Número total de amostras do conjunto	2.602	2.602
Erro MAE do treino (%)	3,8030	7,2838
Erro MAE do teste (%)	3,7281	7,2838
Erro RMSE do treino (%)	5,4510	8,2161
Erro RMSE do teste (%)	5,2803	8,2161

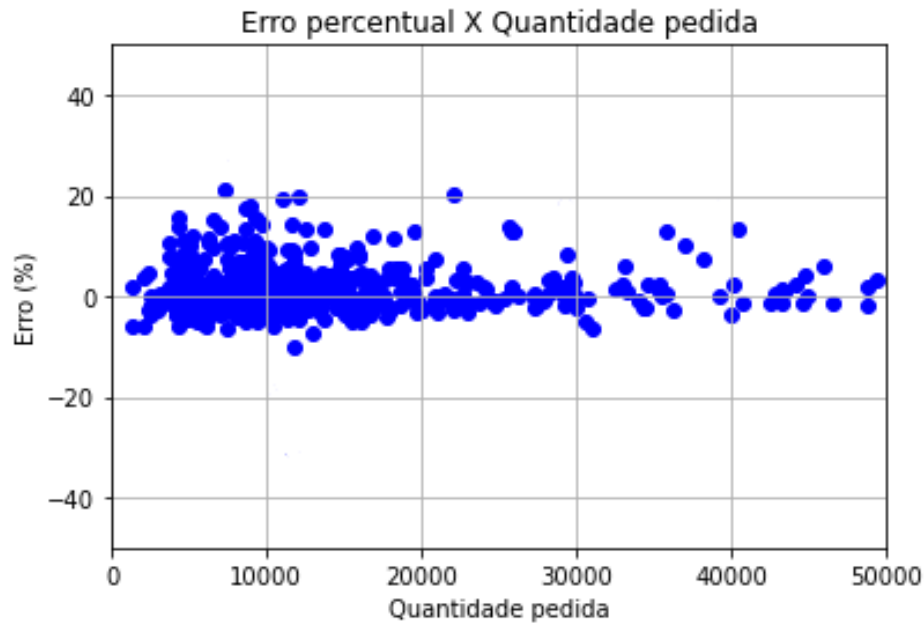
Fonte: O autor

Para este conjunto de dados o modelo de RNA mostrou-se mais eficaz em prever qual a quantidade ideal para a programação da produção em qualquer tamanho de lote solicitado, pois apresentou melhores resultados em ambas as métricas de avaliação da variação do erro percentual, tanto a MAE quanto a RMSE, ficando em torno de 3,80 e 5,45 % para o conjunto de treinamento e 3,73 e 5,28 % para o conjunto de teste, frente ao erro entre 7,28 e 8,22 % do método atual de programação.

O gráfico 1 apresenta a dispersão dos resultados obtidos no teste da RNA em relação à quantidade pedida para todo o conjunto de dados, pelo qual é possível perceber uma diminuição do erro percentual a medida que a quantidade pedida aumenta.



Gráfico 1: Resultados da RNA para o conjunto geral

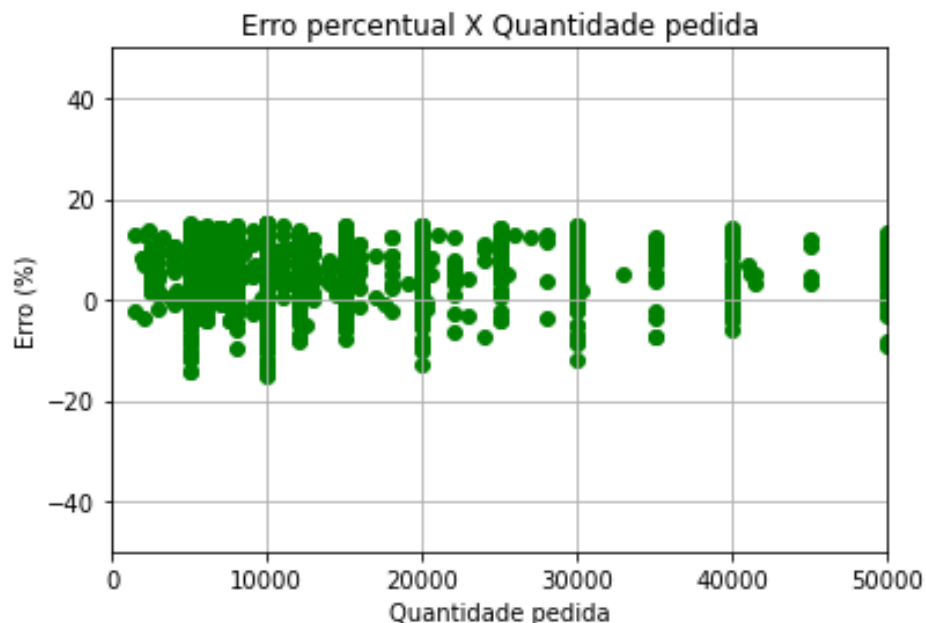


Fonte: O autor

Os pontos em azul representam o erro percentual entre a saída da RNA e a quantidade pedida, representando os 3,73 % de variação do erro calculado utilizando o método MAE e 5,28 % pelo método RMSE. O gráfico 1 inclui apenas as saídas previstas durante a fase de teste, não sendo considerados os erros da fase de treinamento porque o modelo estava em processo de aprendizagem e portanto, o erro pode ser considerado “normal” nesta etapa.

O gráfico 2 apresenta a dispersão do erro do método atual de programação em relação a quantidade entregue e pedida

Gráfico 2: Resultados do método atual para o conjunto geral

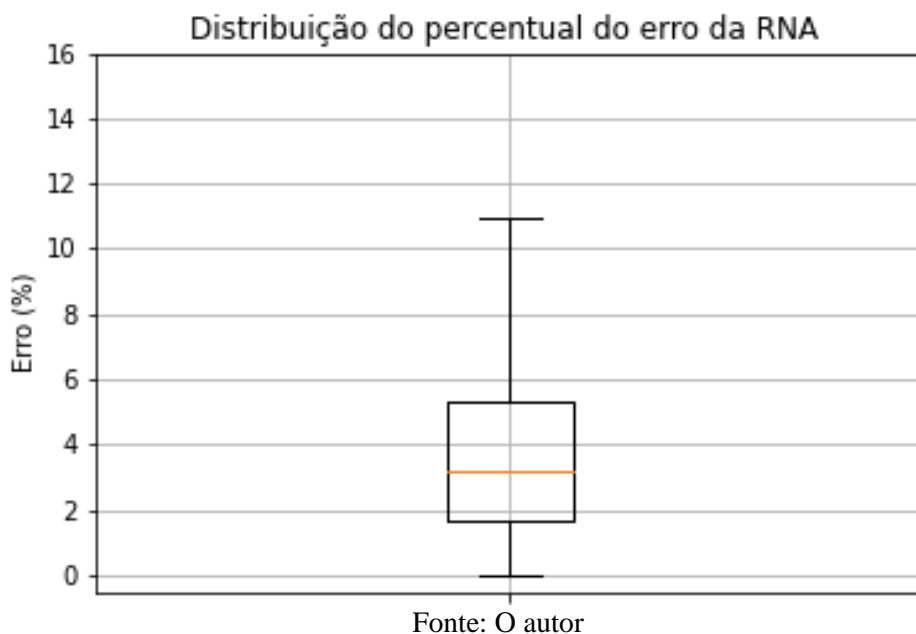


Fonte: O autor

Os pontos em verde do gráfico 2 representam o erro percentual entre o que foi entregue e a quantidade pedida, representando os 7,28 % de variação do erro calculado utilizando o método MAE e 8,22 % pelo método RMSE. O gráfico inclui todas as amostras do conjunto de dados e é possível analisar pela dispersão que a amplitude do erro percentual do método atual é constante independentemente da quantidade pedida.

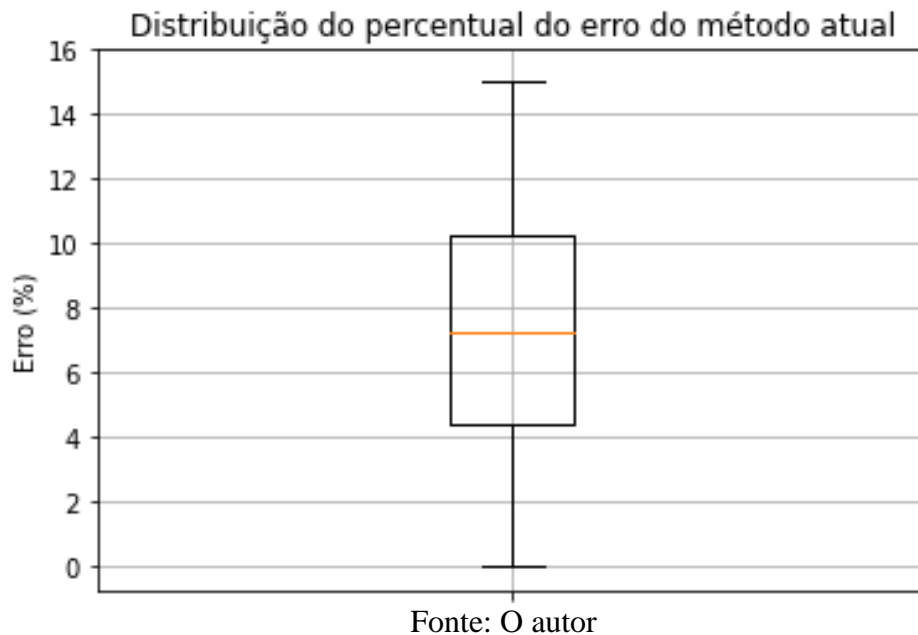
Outra visão sobre o erro percentual da RNA pode ser vista no gráfico 3, o qual apresenta por meio de boxplot uma visão estatística dos resultados obtidos. Nele o erro absoluto dos resultados tem a sua amplitude de 11%, sendo de 0% para o erro mínimo e 11% para o erro máximo. A maior concentração do erro encontra-se entre 2 e 5,5% aproximadamente, sendo que 25% dos resultados obtiveram erro inferior à 2%, 50% obtiveram erro inferior à 3,5% e 75% com erro inferior à 5,5%.

Gráfico 3: Boxplot da distribuição do erro percentual da RNA



O gráfico 4 apresenta também por meio de boxplot o erro percentual do método atual de programação. Nele o erro absoluto dos resultados tem a sua amplitude em 15%, sendo de 0% para o erro mínimo e 15% para o erro máximo. A maior concentração do erro encontra-se entre 4 e 10% aproximadamente, sendo que 25% dos resultados obtiveram erro inferior à 4%, 50% obtiveram erro inferior à 7,5% e 75% com erro inferior à 10%, todos os valores aproximados.

Gráfico 4: Boxplot da distribuição do erro percentual do método atual (5, 10 e 15%)



Para este conjunto de dados gerais a RNA obteve um desempenho melhor em prever a quantidade do lote a ser programado para que a quantidade entregue esteja mais próxima da quantidade pedida. Tanto o erro percentual MAE quanto o RMSE do modelo foi menor em comparação com o erro do método atual, assim como a visualização dos gráficos mostrou uma menor dispersão do erro da RNA.

## 5.2 RESULTADOS PARA QUANTIDADES ATÉ 10.000 UNIDADES

Este conjunto de dados é referente aos pedidos de até 10.000 unidades, ou seja, abrange todas as amostras do histórico de pedidos até 10.000. Os principais resultados obtidos pela RNA para este conjunto de treinamento e de teste estão apresentados na tabela 8 juntamente com os dados calculados para o método atual de programação.

Tabela 8: Resultados dos pedidos até 10.000

<b>Resultado</b>	<b>RNA</b>	<b>Modelo atual</b>
Número total de amostras de treino	834	-
Número total de amostras de teste	358	-
Número total de amostras do conjunto	1.192	1.377
Erro MAE do treino (%)	3,6602	7,4476
Erro MAE do teste (%)	3,4874	7,4476
Erro RMSE do treino (%)	5,2590	8,3796
Erro RMSE do teste (%)	5,0325	8,3796

Fonte: O autor

Para este conjunto de dados o modelo de RNA mostrou-se mais eficaz em prever qual a quantidade ideal da programação da quantidade de material para produção para atender ao tamanho de lote de até 10.000 unidades, pois apresentou melhores resultados em ambas as métricas de avaliação da variação do erro percentual, tanto a MAE quanto a RMSE, ficando em torno de 3,66 e 5,26 % para o conjunto de treinamento e 3,49 e 5,03 % para o conjunto de teste, frente ao erro entre 7,45 e 8,38 % do método atual de programação.

### 5.3 RESULTADOS PARA QUANTIDADES ENTRE 10.000 E 20.000 UNIDADES

O próximo conjunto de dados testado é referente aos pedidos feitos entre 10.000 e 20.000 unidades. Os principais resultados obtidos pela RNA para este conjunto de treinamento e de teste estão apresentados na tabela 9 juntamente com os dados calculados para o método atual de programação.

Tabela 9: Resultados dos pedidos entre 10.000 e 20.000 unidades

<b>Resultado</b>	<b>RNA</b>	<b>Modelo atual</b>
Número total de amostras de treino	710	-
Número total de amostras de teste	305	-
Número total de amostras do conjunto	1.015	856
Erro MAE do treino (%)	3,2154	7,0557
Erro MAE do teste (%)	3,3190	7,0557
Erro RMSE do treino (%)	4,9199	7,9770
Erro RMSE do teste (%)	4,6579	7,9770

Fonte: O autor

Para este conjunto de dados o modelo de RNA também mostrou-se mais eficaz em prever qual a quantidade ideal da programação da produção para o tamanho de lote entre 10.000 e 20.000 unidades, pois apresentou melhores resultados em ambas as métricas de avaliação da variação do erro percentual, tanto a MAE quanto a RMSE, ficando em torno de 3,22 e 4,92 % para o conjunto de treinamento e 3,32 e 4,66 % para o conjunto de teste, frente ao erro entre 7,06 e 7,98 % do método atual de programação.

### 5.4 RESULTADOS PARA QUANTIDADES ENTRE 20.000 E 30.000 UNIDADES

O conjunto de dados seguinte é referente aos pedidos feitos entre 20.000 e 30.000 unidades. Os principais resultados obtidos pela RNA para este conjunto de treinamento e de

teste estão apresentados na tabela 10 juntamente com os dados calculados para o método atual de programação.

Tabela 10: Resultados dos pedidos entre 20.000 e 30.000 unidades

<b>Resultado</b>	<b>RNA</b>	<b>Modelo atual</b>
Número total de amostras de treino	145	-
Número total de amostras de teste	63	-
Número total de amostras do conjunto	208	211
Erro MAE do treino (%)	3,3940	7,4371
Erro MAE do teste (%)	2,9010	7,4371
Erro RMSE do treino (%)	4,8773	8,3484
Erro RMSE do teste (%)	3,7565	8,3484

Fonte: O autor

Neste conjunto de dados o modelo de RNA também mostrou-se mais eficaz em prever qual a quantidade ideal da programação da produção para o tamanho de lote entre 20.000 e 30.000 unidades, pois apresentou melhores resultados em ambas as métricas de avaliação da variação do erro percentual, tanto a MAE quanto a RMSE, ficando em torno de 3,39 e 4,88 % para o conjunto de treinamento e 2,90 e 3,76 % para o conjunto de teste, frente ao erro entre 7,44 e 8,35 % do método atual de programação.

## 5.5 RESULTADOS PARA QUANTIDADES ENTRE 30.000 E 40.000 UNIDADES

Este conjunto de dados é referente aos pedidos feitos entre 30.000 e 40.000 unidades. Os principais resultados obtidos pela RNA para este conjunto de treinamento e de teste estão apresentados na tabela 11 juntamente com os dados calculados para o método atual de programação.

Tabela 11: Resultados dos pedidos entre 30.000 e 40.000 unidades

<b>Resultado</b>	<b>RNA</b>	<b>Modelo atual</b>
Número total de amostras de treino	52	-
Número total de amostras de teste	23	-
Número total de amostras do conjunto	75	62
Erro MAE do treino (%)	3,3099	6,8375
Erro MAE do teste (%)	3,5785	6,8375
Erro RMSE do treino (%)	4,5379	7,7561
Erro RMSE do teste (%)	4,7559	7,7561

Fonte: O autor

Para este conjunto de dados o modelo de RNA mostrou-se mais eficaz em prever qual a quantidade ideal da programação da produção para o tamanho de lote entre 30.000 e 40.000 unidades, pois apresentou melhores resultados em ambas as métricas de avaliação da variação do erro percentual, tanto a MAE quanto a RMSE, ficando em torno de 3,31 e 4,54 % para o conjunto de treinamento e 3,58 e 4,76 % para o conjunto de teste, frente ao erro entre 6,84 e 7,76 % do método atual de programação.

## 5.6 RESULTADOS PARA QUANTIDADES ENTRE 40.000 E 50.000 UNIDADES

O próximo conjunto de dados é referente aos pedidos feitos entre 40.000 e 50.000 unidades. Os principais resultados obtidos pela RNA para este conjunto de treinamento e de teste estão apresentados na tabela 12 juntamente com os dados calculados para o método atual de programação.

Tabela 12: Resultados dos pedidos entre 40.000 e 50.000 unidades

<b>Resultado</b>	<b>RNA</b>	<b>Modelo atual</b>
Número total de amostras de treino	35	-
Número total de amostras de teste	16	-
Número total de amostras do conjunto	51	48
Erro MAE do treino (%)	2,2696	6,4374
Erro MAE do teste (%)	2,2006	6,4374
Erro RMSE do treino (%)	3,0952	7,5954
Erro RMSE do teste (%)	3,4585	7,5954

Fonte: O autor

Para este conjunto de dados o modelo de RNA também mostrou-se mais eficaz em prever qual a quantidade ideal da programação da produção para o tamanho de lote entre 40.000 e 50.000 unidades, pois apresentou melhores resultados em ambas as métricas de avaliação da variação do erro percentual, tanto a MAE quanto a RMSE, ficando em torno de 2,27 e 3,10 % para o conjunto de treinamento e 2,20 e 3,46 % para o conjunto de teste, frente ao erro entre 6,44 e 7,60 % do método atual de programação.

## 5.7 RESULTADOS PARA QUANTIDADES SUPERIORES À 50.000 UNIDADES

Por fim, o último conjunto de dados é referente aos pedidos de feitos acima de 50.000 unidades. Os principais resultados obtidos pela RNA para este conjunto de treinamento e de

teste estão apresentados na tabela 13 juntamente com os dados calculados para o método atual de programação.

Tabela 13: Resultados dos pedidos acima de 50.000 unidades

<b>Resultado</b>	<b>RNA</b>	<b>Modelo atual</b>
Número total de amostras de treino	42	-
Número total de amostras de teste	19	-
Número total de amostras do conjunto	61	48
Erro MAE do treino (%)	3,0164	7,1878
Erro MAE do teste (%)	3,2651	7,1878
Erro RMSE do treino (%)	4,2179	8,0523
Erro RMSE do teste (%)	4,9694	8,0523

Fonte: O autor

Para este último conjunto de dados o modelo de RNA novamente mostrou-se mais eficaz em predizer qual a quantidade ideal da programação da produção para o tamanho de lote superior à 50.000 unidades, pois apresentou melhores resultados em ambas as métricas de avaliação da variação do erro percentual, tanto a MAE quanto a RMSE, ficando em torno de 3,02 e 4,22 % para o conjunto de treinamento e 3,27 e 4,97 % para o conjunto de teste, frente ao erro entre 7,19 e 8,05 % do método atual de programação.

## 6 CONCLUSÕES

A presente seção reserva-se em fazer as considerações finais sobre todo o trabalho apresentado, discutindo a síntese dos resultados, o atendimento aos objetivos de pesquisa, as limitações e as proposta para continuidade do trabalho.

### 6.1 SÍNTESE DOS RESULTADOS

Ao analisar os resultados do teste de sensibilidade dos atributos, conclui-se que através de seu treinamento a RNA conseguiu correlacionar a maioria dos atributos de entrada com a saída esperada, pois dos 10 atributos selecionados, 9 tiveram influência direta no valor de saída do modelo. Apenas o atributo peso ( $\text{g/m}^2$ ) não mostrou-se relevante a ponto de ser correlacionado pela RNA como uma causa que gera perdas no processo.

A partir da análise dos resultados obtidos pelo algoritmo da RNA e pelo método atual de programação, é possível concluir que os resultados alcançados pela RNA nas simulações quando comparados com os dados reais de produção foram os que mais se aproximaram da quantidade final desejada, pois nas simulações apresentaram em média uma variação menor. A síntese dos resultados pode ser conferida na tabela 14.

Tabela 14: Síntese dos resultados

Conjunto testado	RNA		Modelo atual	
	MAE	RSME	MAE	RSME
Geral	3,7281	5,2803	7,2838	8,2161
Até 10.000	3,4874	5,0325	7,4476	8,3796
10.000 – 20.000	3,3190	4,6579	7,0557	7,9770
20.000 – 30.000	2,9010	3,7565	7,4371	8,3484
30.000 – 40.000	3,5785	4,7559	6,8375	7,7561
40.000 – 50.000	2,2006	3,4585	6,4374	7,5954
Acima de 50.000	3,2651	4,9694	7,1878	8,0523

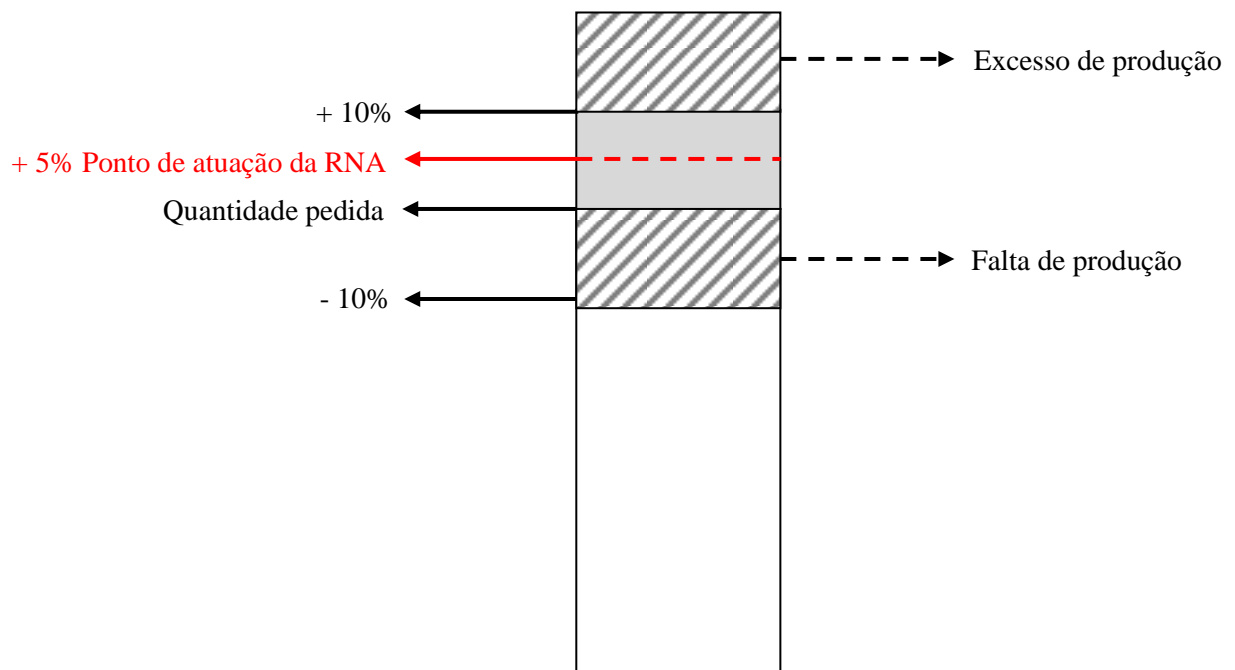
Fonte: O autor

Percebe-se a partir da análise da tabela 13, que o modelo de RNA obteve uma significativa melhora do método de programação da produção em todos os conjuntos de dados testados. Por meio da simulação da realidade o modelo obteve sucesso no objetivo de minimizar a variação na quantidade entregue com relação à pedida, podendo vir a servir como ferramenta para auxiliar o departamento de PCP na atividade de Programação da Produção.



O principal benefício da aplicação da RNA para produzir somente a quantidade necessária de itens pela empresa reflete-se na redução dos custos e no aumento do faturamento. Como representado na figura 7, pensando-se na tolerância dos  $\pm 10\%$  sobre a quantidade pedida pelo cliente, o modelo de RNA desenvolvido pode auxiliar a tomada de decisão para evitar que a empresa produza em excesso e conseqüentemente reduza os custos com estoque, deterioração, obsolescência, espaço físico e alocação de funcionários que poderiam estar realizando outras atividades. Por outro lado, ao evitar que ela produza quantidade insuficiente para atender ao pedido, a RNA estará garantindo o faturamento total do que foi vendido. A ideia é que a empresa mantenha o seu processo de produção padronizado dentro da faixa entre 0 e  $+10\%$ .

Figura 7: Faixa de tolerância dos  $\pm 10\%$  e ponto ótimo de atuação da RNA



Fonte: O autor

O ponto ideal de operação do modelo, que também pode ser visto na figura 7, deve então ser fixado em 5% a mais sobre o pedido do cliente, garantindo dessa maneira que a variação do erro médio de 3,72% calculado pelo MAE e 5,28% calculado pelo RMSE fique entre 0 e  $+10\%$ , ou seja, se a RNA errar para menos a quantidade ainda será suficiente para atender ao pedido e se por acaso ela errar para mais a quantidade ainda estará dentro dos  $+10\%$  que o cliente aceita.

## 6.2 ATENDIMENTO AOS OBJETIVOS DA PESQUISA

Em resgate aos objetivos gerais e específicos definidos ainda no capítulo 1, o trabalho obteve êxito em conseguir alinhar a maior parte do seu desenvolvimento com o caminho traçado.

Inicialmente é possível dizer que o objetivo geral de otimizar a atividade de programação da produção através do desenvolvimento de um modelo de RNA que auxilie na tomada de decisão do departamento de PCP por meio da predição da quantidade necessária de material a ser transformado, de modo que ao final de todas as etapas do processo produtivo a variação entre a quantidade de itens bons do lote entregue e a quantidade solicitada pelo cliente seja a menor possível, foi alcançado. O modelo de RNA criado foi capaz de executar a tarefa proposta pelo objetivo geral e retornar um erro percentual menor que o erro do método atual.

Os objetivos específicos também foram atendidos, o conhecimento adquirido sobre: (a) a compreensão dos conceitos e funcionamento das RNA foi obtida; (b) os modelos de RNA e algoritmos de aprendizagem existentes foram explorados; (c) o algoritmo escrito com o objetivo de executar uma tarefa de predição conseguiu desempenhar a sua função e ofereceu bons resultados; (d) a comparação dos resultados do modelo de RNA e o método atual utilizado para a programação da produção forneceu resultados interessantes e promissores e; os resultados permitiram (e) discutir as implicações do uso da RNA nas receitas e custos da empresa.

## 6.3 LIMITAÇÕES DA PESQUISA

Uma das limitações da pesquisa foi não ter testados outros algoritmos de RNA frente ao MLP e nem ter realizado a comparação dos resultados do modelo com outras ferramentas alternativas que poderiam ser utilizadas para resolver o mesmo problema de pesquisa.

## 6.4 CONTINUIDADE DA PESQUISA E TRABALHOS FUTUROS

Como sugestões para trabalhos futuros, pode-se tentar combinar o modelo com outra ferramenta de análise, testar variações com novos atributos de entrada ou criar uma interface de alto nível que facilite o uso do modelo por pessoas não familiarizadas com a linguagem de programação em Python.

Por último, como sugestão para dar continuidade ao ponto em que este trabalho de pesquisa parou, seja a aplicação prática do modelo na programação real de amostras de pedidos que ainda não foram produzidos, pois este trabalho limitou-se a fazer uma simulação baseada nos dados históricos de produções.

## REFERÊNCIAS

- BARCA, M.C.S., SILVEIRA, T.R., MAGINI, M. **Treinamento de redes neurais artificiais: o algoritmo backpropagation**. IX Encontro Latino Americano de Iniciação Científica. 2005
- FLORENCIO, P.H.B, VIEIRA, S.L. **Aplicação de redes neurais artificiais de Elman na previsão de demanda de peças** . XXXVI Encontro Nacional de Engenharia de Produção. Paraíba, 2016.
- KOTLER, P.; KELLER, K. L. **Administração de marketing**. 14. ed. São Paulo: Pearson Education do Brasil. São Paulo: Prentice Hall, 2012.
- HAYKIN, S. **Neural networks: A comprehensive foundation**. 2ª Ed. New Jersey. Person Prentice Hall, 2001.
- HILLIER, F.S., LIEBERMAN, G.J. **Introdução à pesquisa operacional**. 9ª Ed. Brasil.McGraw Hill, 2013.
- NG, A.Y.T. **Coursera: Machine Learning**. Stanford University. 2020.
- RUSSEL, S.J., NORVIG, P. **Inteligência artificial: un enfoque moderno**, 2ª Ed. Madrid. Person Prentice Hall, 2004.
- SLACK, N. et al. **Administração da produção: edição compacta**. 10ª Reimpressão. São Paulo: Atlas, 2006.
- SILVA, I.N., SPATTI, D.H., FLAUZINO, R.A. **Redes neurais artificiais: para engenharia e ciências aplicadas**. 2ª Ed. São Paulo. Artliber, 2010.
- SILVA, L.C. **Aprendizado de máquina com treinamento continuado aplicado à previsão de demanda de curto prazo: o caso do restaurante universitário da Universidade Federal de Uberlândia**. Tese (Mestrado Ciências) - Universidade Federal de Uberlândia. Uberlândia, 2019.
- SILVA, L.N.C. **Análise e síntese de estratégias de aprendizado para redes neurais artificiais**. Tese (Mestrado Engenharia Elétrica) - UNICAMP. Campinas, 1998.
- SILVA, R.O. **Teorias da administração**. São Paulo. Pearson Prentice Hall, 2008.
- TERRA, A.R.T. **Programação da produção: uma abordagem por redes neurais artificiais**. Tese (Mestrado Engenharia de Produção) - Universidade Federal de São Carlos. São Carlo, 2000.
- TUBINO, D.F. **Planejamento e controle da produção: teoria e prática**. São Paulo: Atlas, 2007.
- TURKIEWICZ, M.S., FRACAROLLI, R.L. **Redes neurais artificiais: importância da aplicação na indústria brasileira**. XXXIX Encontro nacional de engenharia de produção. São Paulo, 2019.
- VALE, G.B. **Reconstrução e reconhecimento de imagens binárias utilizando o algoritmo máquina de Boltzmann**. Tese (Mestrado Ciências) – Universidade Federal de Uberlândia. Uberlândia, 2016.

## APÊNDICE A . Algoritmo do modelo de RNA aplicado

```

#importante as bibliotecas
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

#carregando o arquivo
arquivo = pd.read_csv('C:/Users/Loureno/Downloads/Dados_csv_25_11_20.csv')
arquivo.head()

#separando as variaveis entre predictoras e variavel alvo
y = arquivo['Y_Programado']
x = arquivo.drop('Y_Programado', axis=1)

from sklearn.model_selection import train_test_split

#criando os conjuntos de dados de treino e teste
x_treino, x_teste, y_treino, y_teste = train_test_split(x, y, test_size = 0.3)

#importando algoritmo MLP
from sklearn.neural_network import MLPRegressor

#modelo
modelo = MLPRegressor(hidden_layer_sizes=(50,100),activation='relu',solver='lbfgs')
modelo.fit(x_treino, y_treino)

#tamanho das matrizes e vetores dos dados de treino e teste
print("Tamanho da matriz x de entradas do treino:",x_treino.shape)
print("Tamanho do vetor y de saidas do treino:",y_treino.shape)
print("Tamanho da matriz x de entradas do teste:",x_teste.shape)
print("Tamanho do vetor y de saidas do teste:",y_teste.shape)
print("\nNumero total de amostras de treino:",len(y_treino))
print("Numero total de amostras de teste:",len(y_teste))
Total_de_amostras = len(y_treino)+len(y_teste)
print("Numero total de amostras do conjunto:",Total_de_amostras)

#imprimindo resultado
resultado_treino = modelo.score(x_treino, y_treino)
resultado_teste = modelo.score(x_teste, y_teste)

#erro do treinamento e do teste do modelo utilizando MAE
real_treino = y_treino
x_treino
previsoes_treino = modelo.predict(x_treino)
erro_treino_MAE = sum(abs((real_treino-previsoes_treino)/real_treino))/(len(real_treino))
print("\nErro MAE do treino (%):",erro_treino_MAE*100)

real_teste = y_teste
x_teste
previsoes = modelo.predict(x_teste)
erro_teste_MAE = sum(abs((real_teste-previsoes)/real_teste))/(len(real_teste))
print("Erro MAE do teste (%):",erro_teste_MAE*100)

```

```

#erro do treinamento e do teste do modelo utilizando RMSE
import math
erro_treino_RMSE = math.sqrt(sum(((real_treino-
previsoes_treino)/real_treino)**2)/(len(y_treino)))
print("\nErro RMSE do treino (%):",erro_treino_RMSE*100)

erro_teste_RMSE = math.sqrt(sum(((real_teste-previsoes)/real_teste)**2)/(len(real_teste)))
print("Erro RMSE do teste (%):",erro_teste_RMSE*100)

print("\nReposta real :\n",real_teste)
print("\nPrevisão do modelo :", previsoes)

#PLOTANDO O GRAFICO
#definindo variaveis dos eixos
y = (((real_teste-previsoes)/real_teste)**2)**(1/2))*100
x = x_teste['Lote_entregue']

#plotando grafico
plt.figure()
plt.title("Distribuição do percentual do erro da RNA ")
plt.ylabel("Erro (%)")
plt.boxplot(y, showfliers=False)
plt.yticks([0,2,4,6,8,10,12,14,16])
plt.grid(True)
plt.show

#salvando o grafico como imagem
plt.savefig('Quantidade entregue X quantidade programada.png')

amplitude_erro_teste= abs((real_teste-previsoes)/real_teste)*100

maior_erro_teste= max(amplitude_erro_teste)
maior_acerto_teste= min(amplitude_erro_teste)

print("\nO pior erro do teste foi de ",maior_erro_teste)
print("\nO melhor acerto do teste foi de ",maior_acerto_teste)

#Inserção de novos dados e verificação da sensibilidade dos atributos
x1 = [10000, 1, 0.749, 0.075, 1, 1, 0, 0, 0, 1]
x2 = [15000, 1, 0.749, 0.075, 1, 1, 0, 0, 0, 1]
x3 = [10000, 0, 0.749, 0.075, 1, 1, 0, 0, 0, 1]
x4 = [10000, 1, 1.749, 0.075, 1, 1, 0, 0, 0, 1]
x5 = [10000, 1, 0.749, 0.575, 1, 1, 0, 0, 0, 1]
x6 = [10000, 1, 0.749, 0.075, 4, 4, 0, 0, 0, 1]
x7 = [10000, 1, 0.749, 0.075, 1, 1, 1, 0, 0, 1]
x8 = [10000, 1, 0.749, 0.075, 1, 1, 0, 1, 0, 1]
x9 = [10000, 1, 0.749, 0.075, 1, 1, 0, 0, 1, 1]
x10 = [10000, 1, 0.749, 0.075, 1, 1, 0, 0, 1, 1]

#resultado de predição das novas saídas
pred = modelo.predict([x1, x2, x3, x4, x5, x6, x7, x8,x9])
print("\n ",(pred))

```

**APÊNDICE B.** *Algoritmo usado para calcular o erro do método atual de programação de 5, 10 e 15%*

```

#importando as bibliotecas
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

#carregando o arquivo
arquivo_atual = pd.read_csv('C:/Users/Loureno/Downloads/Metodo_atual.csv')
arquivo_atual.head()

#separando as variaveis de entrada e saída
x_atual = arquivo_atual['Lote_pedido']
y_atual = arquivo_atual['Lote_entregue']

print("Tamanho do lote pedido:\n",x_atual)
print("\nTamanho do lote entregue:\n",y_atual)

#tamanho das matrizes e vetores do conjunto de dados
print("\nTamanho da matriz do conjunto:",arquivo_atual.shape)
print("\nNumero total de amostras do conjunto de dados:",len(arquivo_atual))

#erro do metodo atual utilizando MAE
erro_atual_MAE = sum(abs((y_atual-x_atual)/x_atual))/(len(x_atual))
print("\nErro MAE do metodo atual (%):",erro_atual_MAE*100)

#erro do metodo atual utilizando RMSE
import math
erro_atual_RMSE = math.sqrt(sum(((y_atual-x_atual)/x_atual)**2)/(len(x_atual)))
print("\nErro RMSE do metodo atual (%):",erro_atual_RMSE*100)

#PLOTANDO O GRAFICO
#definindo variaveis dos eixos
y = (((y_atual-x_atual)/x_atual)**2)**(1/2)*100
x = x_atual

#plotando grafico
plt.figure(figsize=(10,10))
plt.title("Distribuição do percentual do erro do método atual")
plt.ylabel("Erro (%)")
plt.boxplot(y, showfliers=False)
plt.yticks([0,2,4,6,8,10,12,14,16])
plt.grid(True)
plt.show

#salvando o grafico como imagem
plt.savefig('Quantidade pedida X quantidade entregue.png')

#amplitude do erro do metodo atual
erro_amplitude= abs((y_atual-x_atual)/x_atual)*100
maior_erro_atual=max(erro_amplitude)
maior_acerto_atual=min(erro_amplitude)

```

```
print("O maior erro do método atual foi de ",maior_erro_atual)  
print("O melhor acerto do metodo atual foi de ",maior_acerto_atual)
```