

# Trabalho de Conclusão de Curso 1

Marcos Wenneton Araújo  
Orientadora: Elloá B. Guedes

<sup>1</sup>Laboratório de Sistemas Inteligentes  
Grupo de Pesquisas em Sistemas Inteligentes  
Escola Superior de Tecnologia  
Universidade do Estado do Amazonas  
Av. Darcy Vargas, 1200, Manaus, AM

{mwvda.eng, ebgcosta}@uea.edu.br

## 1. Introdução

### 1.1. Objetivos

### 1.2. Justificativa

### 1.3. Metodologia

### 1.4. Cronograma

## 2. Fundamentação Teórica

A fundamentação teórica para a elaboração deste trabalho consiste em conceitos ligados ao *Machine Learning*. Primeiramente, os conceitos gerais desta área serão apresentados na Seção 2.1, seguidos pelas características das Redes Neurais Artificiais na Seção 2.2. As definições elementares da técnica de *Machine Learning* conhecida como *Deep Learning* são apresentadas na Seção 2.3. A Seção 2.3.1 discorre sobre as características das Redes Neurais Convolucionais, prosseguindo até a Seção 2.3.2 onde são apresentadas algumas de suas arquiteturas canônicas. Por fim, a Seção 2.3.3 contém informações sobre a técnica conhecida como *Transfer Learning*, um conceito emergente utilizado em aplicações de *Deep Learning*.

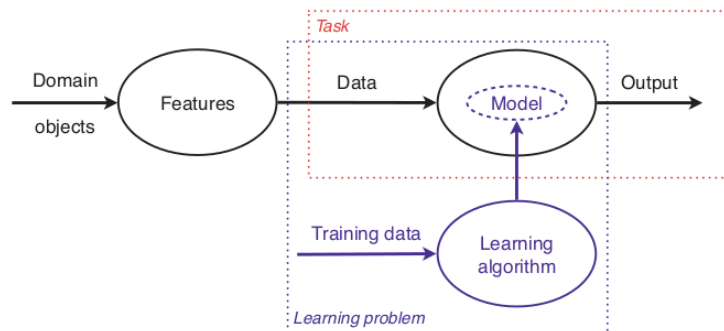
### 2.1. *Machine Learning*

*Machine Learning* (ML), também conhecido como Aprendizado de Máquina, é o estudo sistemático de algoritmos e sistemas que melhoram seu conhecimento ou desempenho com o uso da experiência (FLACH, 2012). Em 1959, o pioneiro em jogos de computador Arthur Samuels definiu *Machine Learning* como um “campo de estudos que dá aos computadores a habilidade de aprender sem serem explicitamente programados” (SIMON, 2013). De acordo com Murphy (MURPHY, 2012), *Machine Learning* pode ser definido como um conjunto de métodos que podem detectar padrões em dados automaticamente e, em seguida, utilizar os padrões detectados para prever dados futuros, ou para realizar outros tipos de decisão sob algum tipo de incerteza.

O *Machine Learning* é comumente dividido em três tipos principais de aprendizado, chamados de supervisionado, não-supervisionado e semi-supervisionado. No caso dos algoritmos de aprendizado supervisionado, o objetivo é aprender um mapeamento de entradas para saídas, dado um conjunto rotulado de pares de entradas e saídas. No aprendizado não supervisionado, o algoritmo é apresentado somente aos dados de entrada, e o

seu propósito é encontrar padrões significativos nos mesmos. Este problema não é bem definido, porque geralmente não é especificado o tipo de padrão que deve ser encontrado nos dados. Além disso, diferentemente do aprendizado supervisionado, não existe nenhuma métrica de erro óbvia para utilizar. O aprendizado semi-supervisionado, por sua vez, normalmente combina uma pequena quantidade de dados rotulados com uma grande quantidade de dados não rotulados para criar um classificador próprio. Em alguns casos, a abordagem de aprendizagem semi-supervisionada pode ser de grande valor prático. (KHAN et al., 2018)

Como mostrado na Figura 1, utilizar os atributos corretos para construir os modelos certos que conquistem determinadas tarefas é a essência de ML. Atributos podem ser definidos como uma “linguagem” que descreve os objetos relevantes no domínio do problema. Tendo a representação característica adequada dos objetos do domínio, normalmente, não é necessário retornar ao objeto em si, e é por isso que os atributos têm um papel importante em ML. A tarefa é uma representação abstrata de um problema que deseja-se resolver levando em consideração os objetos do domínio. Dentre as tarefas de aprendizado supervisionado mais utilizadas estão a tarefa de classificação binária ou multiclasse e a tarefa de regressão. Um modelo, por sua vez, é a saída produzida por dados de treinamento aplicados em determinado algoritmo de ML (FLACH, 2012).



**Figura 1: Uma visão geral de como ML é utilizado para adereçar uma tarefa (FLACH, 2012).**

Em uma tarefa de classificação, um algoritmo é selecionado para especificar quais das  $k$  categorias possíveis uma entrada pertence. Para resolver essa tarefa, o algoritmo de aprendizado normalmente produz uma função  $f : \mathbb{R}^n \rightarrow 1, \dots, k$ . Quando  $y = f(x)$ , o modelo denomina uma entrada descrita com o vetor  $x$  para uma categoria identificado por um valor numérico  $y$  (GOODFELLOW; BENGIO; COURVILLE, 2016). Alguns exemplos de tarefa de classificação são a detecção de faces em imagens, a determinação do gênero do indivíduo nessas imagens, a verificação da espécie de uma planta, entre outros.

Quanto à tarefa de regressão, é solicitado a um programa a predição de um valor numérico a partir de uma entrada. Desta forma, o algoritmo de aprendizado é proposto a retornar uma função  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  (GOODFELLOW; BENGIO; COURVILLE, 2016). Algumas tarefas de regressão podem ser, por exemplo, a determinação do valor de uma corrida de táxi, a identificação da idade de um indivíduo em uma imagem, prever o preço

de uma casa baseado nos dados de casas vendidas anteriormente, etc.

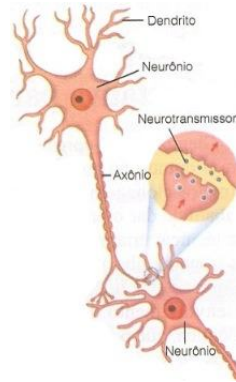
Dentre os tipos de modelos existentes, podemos citar os geométricos, probabilísticos e lógicos. Um modelo geométrico é construído diretamente em função do espaço, utilizando-se de conceitos geométricos como linhas, planos e distâncias. Como exemplos de modelos geométricos temos a regressão linear, o *perceptron* e, em consequência, as redes neurais artificiais. Nos modelos probabilísticos, como o classificador Bayesiano, a questão principal é modelar a relação entre os dados de entrada e saída assumindo que existe algum processo aleatório implícito que produz os valores para essas variáveis, de acordo com uma distribuição de probabilidade bem definida, porém desconhecida. Um modelo lógico, em contrapartida, é o mais naturalmente algorítmico, considerando a capacidade de ser facilmente transformado em regras que podem ser entendidas por seres humanos. Dentre os modelos lógicos estão as árvores de decisão, onde as suas folhas são rotuladas para caracterizar o tipo de tarefa associado ao modelo (FLACH, 2012).

Dentre os modelos geométricos, as redes neurais artificiais têm demonstrado grande desempenho em diversas áreas. Aplicações de *Deep Learning* para detecção de padrões em imagens e reconhecimento de voz, por exemplo, utilizam-se desses modelos para a obtenção de resultados relevantes. Tomando isto e levando em consideração o contexto deste trabalho, as próximas seções apresentadas abordam significativamente estes conceitos.

## 2.2. Redes Neurais Artificiais

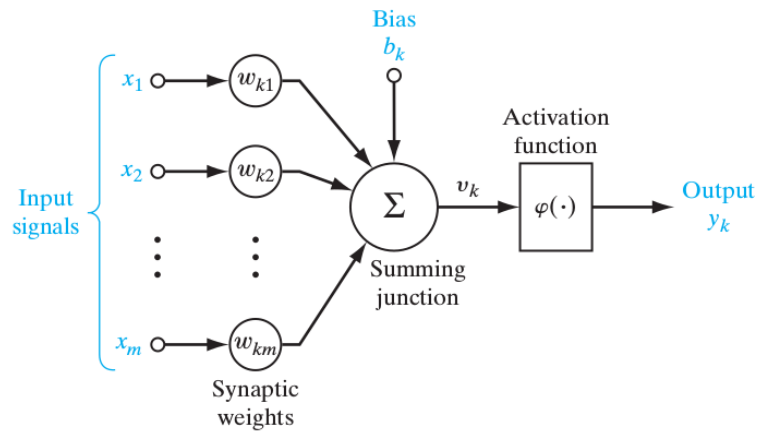
As *Redes Neurais Artificiais* (RNAs) são uma tentativa computacional de modelar a capacidade de processamento de informação do sistema nervoso humano (ROJAS, 1996). Para alcançarem um bom desempenho, as RNAs empregam uma interligação de estruturas bases chamadas de neurônios artificiais que, por sua vez, possuem pesos com valores numéricos positivos ou negativos associados a si. Uma vantagem das RNAs, é a grande capacidade de generalização, ou seja, a habilidade de produzir saídas adequadas para entradas que não estavam presente anteriormente durante sua aprendizagem (HAYKIN, 2009). As RNAs têm sido frequentemente utilizadas nas áreas de processamento de sinais, reconhecimento de padrões, medicina, reconhecimento de voz, negócios, entre outras (FAUSETT, 1993).

A idealização dos neurônios artificiais foi inspirada nos neurônios biológicos encontrados no cérebro humano. Como mostrado na Figura 2, cada neurônio biológico é composto pelo corpo celular, os dendritos e o axônio. Os dendritos têm como papel a recepção das informações, ou impulsos nervosos, de outros neurônios e submetê-las ao corpo celular, onde as informações são processadas e novos impulsos são gerados. Estes impulsos são enviados aos dendritos de outros neurônios através do axônio. O ponto de contato entre os neurônios através do axônio e os dendritos, denominado sinapse, é onde ocorre toda a troca de informação necessária para conceber uma rede neural (BRAGA; CARVALHO; LUDERMIR, 2000).



**Figura 2: Estrutura de um neurônio e zona de conexão funcional entre as células (sinapse) (BARDINE, ).**

Um neurônio biológico dispara quando a soma dos impulsos que ele recebe ultrapassa o seu limiar de excitação, denominado *threshold*. O corpo do neurônio é controlado por um mecanismo simples que faz a soma dos valores  $x_i w_i$  recebidos pelos seus dendritos e decide se o neurônio deve ou não disparar, comparando a soma obtida com o *threshold* do neurônio. No modelo de neurônio artificial proposto por McCulloch e Pitts (MCP) em (MCCULLOCH; PITTS, 1943), a ativação do neurônio é obtida através da aplicação de uma *função de ativação*, que ativa ou não a saída, dependendo do valor da soma ponderada das suas entradas, como é mostrado na Figura ?? (BRAGA; CARVALHO; LUDERMIR, 2000).



**Figura 3: Estrutura de um neurônio artificial. Fonte: (HAYKIN, 2009)**

Uma função de ativação  $f$  é responsável por determinar a saída  $y$  do neurônio, a partir das entradas  $x = x_1, \dots, x_n$  e pesos  $w = w_1, \dots, w_n$ . No caso do neurônio MCP, a função de ativação é do tipo degrau deslocada e é comparada ao seu *threshold*  $\theta$ , como mostrado nas Equações 1 e 2.

$$\phi(x, w) = \sum_{i=1}^n x_i w_i \quad (1)$$

$$f(\phi) = \begin{cases} 1, & \text{se } \phi > \theta \\ 0, & \text{caso contrário} \end{cases} \quad (2)$$

Para a correta funcionalidade de certos algoritmos em RNAs, como o de *back-propagation* em redes *multilayer perceptron*, sabe-se que a escolha da função de ativação deve considerar funções contínuas e diferenciáveis (HAYKIN, 2009). As funções identidade, sigmóide, tangente hiperbólica, retificada linear (ReLU) e *softmax*, são as mais utilizadas no processo de construção de RNAs. Algumas propriedades dessas funções estão apresentadas na Figura ??.

Em 1958, visando a melhoria do neurônio MCP, Frank Rosenblatt desenvolveu o modelo *Perceptron* (ROSENBLATT, 1958). Neste modelo, criou-se o primeiro conceito de aprendizagem através de neurônios artificiais, onde foi projetada uma regra de correção de erros para modificar os pesos associados a um neurônio quando suas respostas aos estímulos apresentados à rede forem erradas (ARBIB, 2003). Durante o processo de adaptação à resposta real, deseja-se identificar um valor  $\Delta w$  a ser aplicado ao vetor de pesos  $w$ , para que seu valor atualizado  $w(t+1)$  esteja mais próximo da solução desejada do que o valor atual  $w(t)$ . Para isso, definiu-se a Equação 3, onde  $\eta$  indica uma *taxa de aprendizado*, ou a velocidade em que o vetor de pesos será atualizado, e  $\hat{y}$  significa o valor previsto pela rede naquela interação. Desta forma, o modelo Perceptron adquiriu a capacidade de resolver problemas linearmente separáveis (BRAGA; CARVALHO; LUDERMIR, 2000).

$$w(t+1) = w(t) + \eta(y - \hat{y})x(t) \quad (3)$$

Neurônios artificiais possuem uma capacidade de generalização limitada, independentemente da função de ativação escolhida, devido a sua habilidade de resolver apenas problemas linearmente separáveis. Entretanto, a combinação desses neurônios para a formação de uma rede é capaz de resolver problemas de elevada complexidade (BRAGA; CARVALHO; LUDERMIR, 2000). Geralmente, identificam-se três classes fundamentais de RNAs, as *feedforward* com uma única camada, as *feedforward* com múltiplas camadas e as recorrentes. Numa rede do tipo *feedforward*, mostrada nas Figuras ?? e ??, existe uma camada de entrada que é projetada diretamente para uma camada de saída constituída de neurônios, e nunca ao contrário. Uma rede recorrente, por sua vez, possui conexões ponderadas dentro de uma camada e diferencia-se pela presença de pelo menos um loop de retorno a camadas anteriores (HAYKIN, 2009).

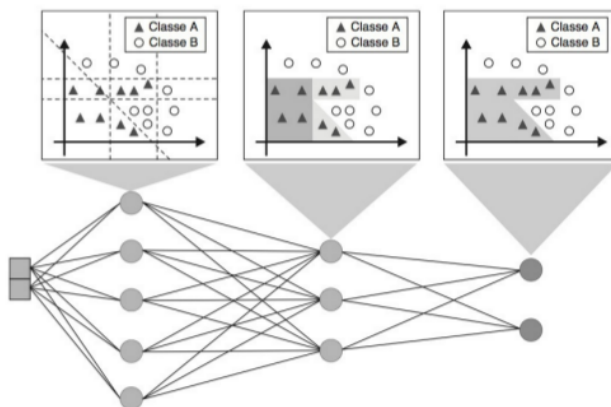
Redes com múltiplas camadas, como na Figura ??, caracterizam-se pela presença de pelo menos uma camada oculta. Segundo Cybenko, uma rede com uma camada oculta pode mapear qualquer função contínua, enquanto uma rede com duas camadas ocultas é suficiente para mapear qualquer função (CYBENKO, 1989).

### 2.2.1. Multilayer Perceptron

As RNAs do tipo *Multilayer Perceptron* (MLP), são redes constituídas do neurônio Perceptron, *feedforward* e com múltiplas camadas, sendo estas uma camada de entrada, uma

ou mais camadas ocultas e uma camada de saída. A arquitetura mais comum para uma rede MLP é a completamente conectada, de forma que os neurônios de uma camada estão conectados com todos os neurônios da próxima camada (FACELI et al., 2011).

Em uma rede MLP, a função implementada por um neurônio de certa camada é uma combinação das funções realizadas pelos neurônios da camada anterior que estão conectados a ele. Na primeira camada, cada neurônio aprende uma função que define um hiperplano. Na camada seguinte, os neurônios combinam um grupo de hiperplanos, formando regiões convexas. Os neurônios da camada seguinte combinam então um sub-conjunto das regiões convexas em regiões de formato arbitrário (FACELI et al., 2011). Na Figura 4, tem-se uma visualização do processo ocorrido.



**Figura 4: Papel exercido pelos neurônios em cada camada de uma rede MLP. Fonte: (FACELI et al., 2011).**

O algoritmo de aprendizado supervisionado mais conhecido e utilizado para treinamento das MLPs é o *backpropagation*. Neste algoritmo utiliza-se as entradas e as saídas desejadas para o ajuste dos erros da rede. O treinamento ocorre em duas fases, a fase *forward* e a fase *backward*, em que cada fase percorre a rede em um sentido. Na fase *forward*, a saída da rede é definida considerando certo padrão de entrada. A fase *backward* utiliza a saída desejada e a saída fornecida pela rede para atualizar os pesos nas suas conexões (BRAGA; CARVALHO; LUDERMIR, 2000). O *backpropagation* é simplesmente um método que utiliza o gradiente descendente para minimizar o erro quadrático total da saída calculada pela rede, na qual a derivada parcial define o ajuste dos pesos. Essa derivada mede a contribuição de cada peso no erro da rede para a classificação de dado objeto (FAUSETT, 1993; FACELI et al., 2011).

No âmbito do cálculo, o gradiente indica o sentido e a direção para os quais devem-se mover os valores dos pesos e dos bias nas camadas de forma a se obter o maior incremento possível de perda. Porém, nas técnica de *backpropagation*, queremos mudanças de peso que trarão a inclinação mais íngreme ao longo da função de erro (GODFELLOW; BENGIO; COURVILLE, 2016; KUBAT, 2015).

### **2.3. Deep Learning**

#### **2.3.1. Redes Neurais Convolucionais**

#### **2.3.2. Arquiteturas canônicas de Redes Neurais Convolucionais**

#### **2.3.3. Transfer Learning**

### **3. Trabalhos Relacionados**

### **4. Solução Proposta**

#### **4.1. Visão Geral da solução proposta**

#### **4.2. -haverá mais-**

### **5. Resultados Parciais**

### **6. Considerações Parciais**

### **Referências**

ARBIB, M. A. (Ed.). *The Handbook of Brain Theory and Neural Networks*. Cambridge, Massachussets: The MIT Press, 2003.

BARDINE, R. *Sinapse do Neurônio*. <https://www.coladaweb.com/biologia/histologia/sinapse-do-neuronio>. Acesso em 24 de fevereiro de 2019.

BRAGA, A. de P.; CARVALHO, A. P. de Leon F. de; LUDERMIR, T. B. *Redes Neurais Artificiais: Teorias e Aplicações*. Rio de Janeiro, RJ: Livros Técnicos e Científicos Editora S.A., 2000.

CYBENKO, G. Approximation by superpositions of a sigmoidal function. *Math. Control Signals Systems*, v. 2, p. 303–314, 1989.

FACELI, K. et al. *Inteligência Artificial: Uma abordagem de Aprendizado de Máquina*. Rio de Janeiro, RJ: Livros Técnicos e Científicos Editora S.A., 2011.

FAUSETT, L. *Fundamentals of Neural Networks: Architectures, algorithms and applications*. [S.l.]: Pearson, 1993.

FLACH, P. *Machine Learning: The Art and Science of Algorithms that Make Sense of Data*. The Edinburgh Building, Cambridge, UK: Cambridge University Press, 2012.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. *Deep Learning*. [S.l.]: MIT press, 2016.

HAYKIN, S. *Neural Networks and Learning Machines*. Hamilton, Ontario, Canada: Pearson, 2009.

KHAN, S. et al. *A Guide to Convolutional Neural Networks for Computer Vision*. Austrália: Morgan & Claypool, 2018.

KUBAT, M. *An Introduction to Machine Learning*. Coral Gables, FL, USA: Springer International Publishing, 2015.

MCCULLOCH, W. S.; PITTS, W. H. A logical calculus of ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, v. 5, p. 115–133, 1943.

MURPHY, K. P. *Machine Learning: A Probabilistic Perspective*. Cambridge, Massachussets: The MIT Press, 2012.

ROJAS, R. *Neural Networks: A Systematic Introduction*. Berlin: Springer, 1996.

ROSENBLATT, F. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, v. 6, n. 6, p. 386, 1958.

SIMON, P. *Too Big to ignore*. Hoboken, New Jersey: John Wiley and Sons, Inc., 2013.