



UNIVERSIDADE FEDERAL DO AMAZONAS

FACULDADE DE TECNOLOGIA

ENGENHARIA DA COMPUTAÇÃO

Avaliação de arquiteturas de Redes Neurais Recorrentes  
para Reconhecimento de Atividades Humanas  
utilizando acelerômetros de dispositivos móveis

Guilherme Souza da Silva

Manaus - AM

2022

Guilherme Souza da Silva

Avaliação de arquiteturas de Redes Neurais Recorrentes  
para Reconhecimento de Atividades Humanas  
utilizando acelerômetros de dispositivos móveis

Monografia apresentada à Coordenação do Curso de Engenharia da Computação da Universidade Federal do Amazonas, como parte dos requisitos necessários à obtenção de título de Engenheiro de Computação.

Orientador(a)

Cícero Ferreira F Costa Filho, Dr.

Universidade Federal do Amazonas  
Faculdade de Tecnologia

Manaus - AM

2022

## Ficha Catalográfica

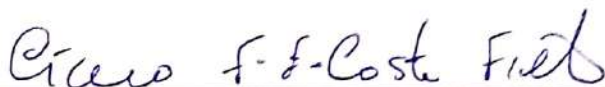
Ficha catalográfica elaborada automaticamente de acordo com os dados fornecidos pelo(a) autor(a).

S586a Silva, Guilherme Souza da  
Avaliação de arquiteturas de redes neurais recorrentes para reconhecimento de atividades humanas utilizando acelerômetros de dispositivos móveis / Guilherme Souza da Silva . 2022  
45 f.: 31 cm.

Orientador: Cícero Ferreira Fernandes Costa Filho  
TCC de Graduação (Engenharia da Computação) - Universidade Federal do Amazonas.

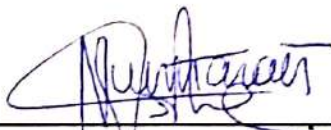
1. Reconhecimento de Atividades Humanas. 2. Redes Neurais Artificiais. 3. Redes Neurais Recorrentes. 4. UniMib-SHAR. I. Costa Filho, Cícero Ferreira Fernandes. II. Universidade Federal do Amazonas III. Título

Monografia de Graduação sob o título *Avaliação de arquiteturas de Redes Neurais Recorrentes para Reconhecimento de Atividades Humanas utilizando acelerômetros de dispositivos móveis* apresentada por Guilherme Souza da Silva e aceita pela Coordenação de Engenharia da Computação da Universidade Federal do Amazonas, sendo aprovada por todos os membros da banca examinadora abaixo especificada:



---

Dr. Cícero Ferreira Fernandes Costa Filho  
Departamento de Eletrônica e Computação  
Universidade Federal do Amazonas



---

Dra. Marly Guimarães Fernandes Costa  
Departamento de Eletrônica e Computação  
Universidade Federal do Amazonas



---

Dr. Celso Barbosa Carvalho  
Departamento de Eletrônica e Computação  
Universidade Federal do Amazonas

Manaus - AM, 02 de setembro de 2022

Dedico este trabalho à minha família, meus amigos, colegas e professores que me acompanharam e me auxiliaram ao longo dessa jornada.

# Avaliação de arquiteturas de Redes Neurais Recorrentes para Reconhecimento de Atividades Humanas utilizando acelerômetros de dispositivos móveis

Autor: Guilherme Souza da Silva

Orientador: Cícero Ferreira F Costa Filho, Dr.

## Resumo

O Reconhecimento de Atividades Humanas (RAH) utilizando dispositivos móveis possibilita classificar atividades físicas de um indivíduo com dados de sensores não intrusivos e de baixo custo. Dessa forma, o RAH pode colaborar com o tratamento fisioterapêutico ao permitir o acompanhamento de atividades como andar, correr, e subir ou descer escadas. A base de dados *UniMib SHAR* disponibiliza dados recolhidos de acelerômetros de dispositivos móveis a fim de serem uma referência para avaliação de algoritmos de aprendizado de máquina para RAH. Este trabalho visa comparar diferentes arquiteturas para classificação da base de dados *UniMib SHAR*, combinando os modelos *LSTM*, *GRU*, e *BLSTM*, com algoritmos de otimização, e técnicas de melhoria de generalização, resultando em quase 40 modelos de redes neurais diferentes. Os resultados mostram que as redes *BLSTM* e *GRU*, combinadas com os algoritmos de otimização *RMSProp* e *Adam*, e método de regularização *Dropout*, obtiveram desempenho superior comparado às demais combinações.

*Palavras-chave:* Reconhecimento de Atividades Humanas, Redes Neurais Artificiais, Redes Neurais Recorrentes, UniMib-SHAR.

# Avaliação de arquiteturas de Redes Neurais Recorrentes para Reconhecimento de Atividades Humanas utilizando acelerômetros de dispositivos móveis

Autor: Guilherme Souza da Silva

Orientador: Cícero Ferreira F Costa Filho, Dr.

## Abstract

Human Activity Recognition (HAR) using mobile devices makes it possible to classify an individual's physical activities with non-intrusive and low-cost sensor data. Thus, HAR can collaborate with physiotherapeutic treatment by allowing the monitoring of activities such as walking, running, and going up or down stairs. *UniMib SHAR* database provides data collected from mobile device accelerometers to be a reference for evaluating machine learning algorithms for HAR. This work aims to compare different architectures for classifying the *UniMib SHAR* database, combining *LSTM*, *GRU*, and *BLSTM* models, with optimization algorithms, and regularization techniques, resulting in almost 40 different neural network models. The results show that the *BLSTM* and *GRU* networks, combined with the *RMSProp* and *Adam* optimization algorithms, and *Dropout* regularization method, obtained superior performance compared to the other combinations.

*Keywords:* Human Activity Recognition, Artificial Neural Networks, Recurrent Neural Networks, UniMib-SHAR.

---

## LISTA DE ILUSTRAÇÕES

Figura 1 – Rede neural biológica (ACADEMY, 2022) . . . . .	16
Figura 2 – Rede neural artificial (COSTA et al., 2016) . . . . .	16
Figura 3 – Rede neural Perceptron (DESHPANDE, 2020): da esquerda para a direita: entradas ( <i>inputs</i> ), pesos ( <i>weights</i> ), somatório ( <i>sum</i> ) e função de ativação ( <i>activation function</i> ) . . . . .	17
Figura 4 – Exemplo de dados lineares e não linearmente separáveis (AGGARWAL et al., 2018) . . . . .	17
Figura 5 – Rede neural Perceptron Multicamadas (KANG; CHOI, 2021): da esquerda para a direita: nós de entrada ( <i>input node</i> ); nós escondidos ( <i>hidden node</i> ); nós de saída ( <i>output node</i> ) . . . . .	19
Figura 6 – Rede neural recorrente e sua representação temporal. Baseado em Goodfellow, Bengio e Courville (2016) . . . . .	19
Figura 7 – Arquitetura de uma rede recorrente <i>LSTM</i> (GRAVES, 2012) . . . . .	21
Figura 8 – Exemplo de três redes com performance de generalização diferentes (GOODFELLOW; BENGIO; COURVILLE, 2016): figura da esquerda: sub-treinamento ( <i>underfitting</i> ); figura do meio: rede com boa generalização; figura da direita: sobre treinamento ( <i>overfitting</i> ) . . . . .	23
Figura 9 – Comparação entre uma rede neural completa com a aplicação da técnica de regularização <i>Dropout</i> (SRIVASTAVA et al., 2014) . . . . .	24
Figura 10 – Gráfico da acurácia e da perda em função das épocas para a arquitetura de rede <i>BRS</i> . . . . .	37



Figura 11 – Gráfico da acurácia e da perda em função das épocas para a arquitetura de rede <i>GSDL</i> . . . . .	37
--	----

---

## LISTA DE TABELAS

Tabela 1 – Atividades humanas disponíveis na base de dados <i>UniMiB SHAR</i> . .	30
Tabela 2 – Atividades diárias utilizadas para classificação da base de dados <i>UniMiB SHAR</i> . . . . .	33
Tabela 3 – Perda e acurácia da rede <i>LSTM</i> . . . . .	35
Tabela 4 – Perda e acurácia da rede <i>GRU</i> . . . . .	36
Tabela 5 – Perda e acurácia da rede <i>BLSTM</i> . . . . .	36
Tabela 6 – Perda e acurácia das arquiteturas de rede com maior performance . .	38
Tabela 7 – Perda e acurácia das arquiteturas de rede com menor performance .	38

---

# SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b> . . . . .	<b>11</b>
<b>1.1</b>	<b>Objetivos</b> . . . . .	<b>12</b>
1.1.1	Objetivo geral . . . . .	12
1.1.2	Objetivos específicos . . . . .	12
<b>1.2</b>	<b>Organização do Trabalho</b> . . . . .	<b>12</b>
<b>2</b>	<b>REDES NEURAIS ARTIFICIAIS</b> . . . . .	<b>14</b>
<b>2.1</b>	<b>Aprendizagem de Máquina</b> . . . . .	<b>14</b>
<b>2.2</b>	<b>Redes Neurais Rasas</b> . . . . .	<b>15</b>
2.2.1	Perceptron . . . . .	16
2.2.2	Perceptron Multicamadas . . . . .	17
<b>2.3</b>	<b>Redes Neurais Recorrentes</b> . . . . .	<b>18</b>
2.3.1	<i>Long Short-Term Memory – LSTM</i> . . . . .	20
2.3.2	<i>Bidirectional Long Short-Term Memory – BLSTM</i> . . . . .	20
2.3.3	<i>Gated Recurrent Unit – GRU</i> . . . . .	21
<b>2.4</b>	<b>Regularização</b> . . . . .	<b>22</b>
2.4.1	Parada antecipada . . . . .	23
2.4.2	<i>Dropout</i> . . . . .	24
2.4.3	Decaimento de peso . . . . .	24
<b>2.5</b>	<b>Otimização</b> . . . . .	<b>25</b>
<b>2.6</b>	<b>Medidas de Desempenho</b> . . . . .	<b>26</b>
<b>3</b>	<b>RECONHECIMENTO DE ATIVIDADES HUMANAS</b> . . . . .	<b>27</b>
<b>3.1</b>	<b>Smartphones e sensores</b> . . . . .	<b>28</b>
<b>3.2</b>	<b>Etapas</b> . . . . .	<b>29</b>

<b>3.3</b>	<b>Base de dados <i>UniMiB SHAR</i></b> . . . . .	<b>30</b>
<b>3.4</b>	<b>Trabalhos relacionados</b> . . . . .	<b>31</b>
<b>4</b>	<b>METODOLOGIA</b> . . . . .	<b>32</b>
<b>4.1</b>	<b>Arquiteturas de Redes Neurais Recorrentes</b> . . . . .	<b>32</b>
<b>4.2</b>	<b>Base de Dados</b> . . . . .	<b>32</b>
<b>4.3</b>	<b>Recursos utilizados</b> . . . . .	<b>33</b>
<b>5</b>	<b>RESULTADOS E DISCUSSÃO</b> . . . . .	<b>35</b>
<b>6</b>	<b>CONCLUSÕES</b> . . . . .	<b>40</b>
	<b>Referências</b> . . . . .	<b>41</b>

# 1

---

## INTRODUÇÃO

Reconhecimento de Atividades Humanas (RAH) utilizando sensores de dispositivos móveis interliga a emergente área de *wearables* a técnicas de aprendizado de máquina com o intuito de reconhecer diversos tipos de atividades humanas em tempo real. RAH possui uma vasta gama de aplicações, incluindo detecção de quedas para idosos com dificuldade de locomoção (Ngu et al., 2020), detecção de atividades anormais em pacientes com deficiência cognitiva (YIN; YANG; PAN, 2008) e monitoramento de atividades diárias para incentivo ao exercício físico (SUBASI et al., 2018).

Segundo Varshney (2014), tais aplicações inteligentes e personalizadas ao paciente podem levar a uma melhora na qualidade de saúde a um menor custo. Aparelhos vestíveis são adequados para este cenário devido a seu baixo custo e alto poder computacional, possibilitando a extração de sinais adquiridos por sensores de forma não intrusiva (Perez; Labrador; Barbeau, 2010).

A base de dados *UniMiB SHAR* (Reconhecimento de Atividades Humanas baseadas em Smartphone, disponibilizada por pesquisadores da Universidade de Milano Bicocca - *University of Milano Bicocca Smartphone-based Human Activity Recognition*) tem como objetivo tornar acessível um vasto conjunto de dados recolhidos de sensores de acelerômetro de *smartphones* para ser usado como uma referência comum para avaliação de reconhecimento de atividades diárias (MICUCCI; MOBILIO; NAPOLETANO, 2017).

Segundo Wang et al. (2019), embora técnicas convencionais de aprendizado de máquina tenham tido êxito no reconhecimento de atividades humanas, como a técnica de K Vizinhos Mais Próximos (FALCO; PIETRO; SANNINO, 2019) e Floresta

Aleatória (AL-TAEI, 2017), as técnicas de aprendizado profundo (*deep learning*) reduzem a dependência do fator humano na etapa de extração de características e atingem melhores desempenhos ao aprenderem sozinhas as representações de alto nível a partir dos dados de sensores.

Nesse contexto, este trabalho visa comparar o desempenho de arquiteturas de Redes Neurais Recorrentes (*RNR*), um tipo de rede profunda, utilizando diferentes métodos de otimização e generalização no reconhecimento de atividades humanas, utilizando dados da base *UniMiB SHAR*.

## 1.1 Objetivos

### 1.1.1 Objetivo geral

O presente trabalho tem como objetivo geral avaliar o desempenho de redes recorrentes no reconhecimento de atividades humanas, utilizando dados provenientes de sensores de acelerômetros de dispositivos móveis disponíveis na base *UniMiB SHAR*.

### 1.1.2 Objetivos específicos

1. Treinar e avaliar o desempenho das redes recorrentes *LSTM*, *GRU* e *LSTM* bidirecional no reconhecimento de nove atividades diárias de 7 indivíduos da base *UniMiB SHAR*;
2. Utilizar e avaliar o desempenho de diferentes técnicas de melhoria de generalização e otimização em conjunto com as redes citadas no item anterior.

## 1.2 Organização do Trabalho

Este trabalho está organizado da seguinte forma. No capítulo 2, apresenta-se uma fundamentação que abrange aprendizado de máquina, redes neurais artificiais e, mais especificamente, redes neurais recorrentes. No capítulo 3, são apresentados conceitos

---

relacionados ao Reconhecimento de Atividades Humanas, à base de dados utilizada, bem como uma revisão da literatura sobre aplicações de Redes Neurais Recorrentes em reconhecimento de atividades humanas. No capítulo 4, descreve-se a metodologia empregada no treinamento das redes neurais e os resultados dos experimentos. No capítulo 5, estão expostos os resultados e a discussão a respeito dos mesmos. Por fim, no capítulo 6, serão apresentadas as conclusões acerca dos resultados obtidos.

## 2

---

# REDES NEURAIS ARTIFICIAIS

Neste capítulo serão apresentados os principais conceitos e fundamentos relacionados à aprendizagem de máquina e redes neurais profundas utilizados no decorrer deste trabalho.

## 2.1 Aprendizagem de Máquina

Segundo Mohri, Rostamizadeh e Talwalkar (2018), aprendizado de máquina pode ser definido como métodos computacionais que utilizam a experiência para realizar previsões acuradas. Nesse contexto, experiência se refere a informações já conhecidas e recuperadas para serem utilizadas como objeto de aprendizado, de modo que a qualidade e quantidade de dados disponíveis para análise, ou o conjunto de treinamento, são essenciais para o sucesso durante a aprendizagem.

Existem três paradigmas principais quanto aos métodos de aprendizado:

- Aprendizado supervisionado: o algoritmo é treinado inicialmente com um conjunto de dados contendo suas características e classificações. Ou seja, os dados utilizados para o treinamento já contém os resultados corretos das previsões;
- Aprendizado não-supervisionado: aprendizado com um conjunto de dados que não possuem rótulos. Dessa forma, o algoritmo precisa identificar padrões e classificar os dados utilizando apenas as características dos mesmos;



- **Aprendizado por reforço:** algoritmo treinado para realizar tomadas de decisões dependendo das circunstâncias aos quais o sistema está inserido. A ideia é treinar a máquina com recompensas ou punições de acordo com a ação tomada; com um número considerável de repetições, o algoritmo irá aprender qual ação mais recompensadora e menos punitiva.

O método de aprendizado utilizado neste trabalho será supervisionado, o qual pode ser dividido nas seguintes etapas:

1. Aquisição dos dados;
2. Processamento dos dados;
3. Divisão dos dados entre conjunto de treinamento e conjunto de teste;
4. Treinamento do algoritmo para classificar o conjunto de treinamento nas respectivas classes já definidas;
5. Avaliação da eficácia do algoritmo, comparando as classificações preditas no conjunto de teste com as classificações verdadeiras;
6. Caso o algoritmo apresente eficácia insuficiente ou inadequada, altera-se parâmetros ou até mesmo o modelo utilizado e treina-se novamente;
7. Repetição dos passos anteriores até que o algoritmo obtenha resultados satisfatórios com base em métricas especificadas.

Segundo Mohri, Rostamizadeh e Talwalkar (2018), o objetivo principal do aprendizado de máquina consiste em gerar previsões acuradas e desenvolver algoritmos robustos e eficientes para produzir tais previsões, mesmo para problemas em larga escala.

## 2.2 Redes Neurais Rasas

Rede Neural Artificial (RNA) é uma técnica de aprendizado de máquina que tem como principal aspecto simular o comportamento de redes neurais do sistema nervoso

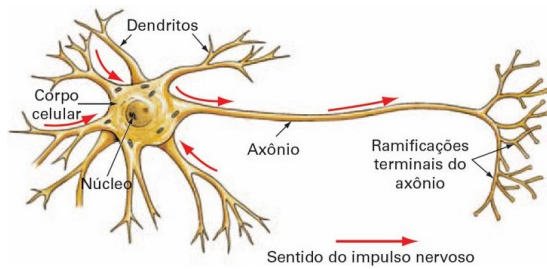


Figura 1 – Rede neural biológica (ACADEMY, 2022)

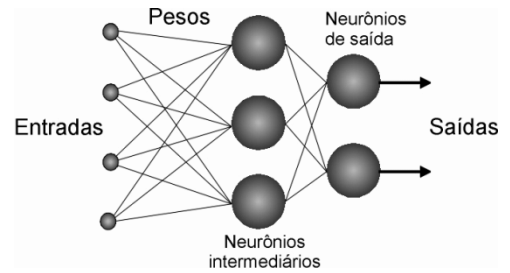


Figura 2 – Rede neural artificial (COSTA et al., 2016)

humano. Esse sistema contém células, conhecidas como neurônios, que são conectadas umas as outras por meio de axônios e dendritos, conforme mostrado na Figura 1. A região de conexão entre axônios e dendritos é chamada de sinápsis. A aprendizagem está diretamente relacionada com o nível de força das conexões sinápticas (AGGARWAL et al., 2018).

De forma similar, as RNAs possuem unidades computacionais, chamadas de neurônios, as quais são conectadas umas as outras por meio de pesos, conforme ilustrado na Figura 2. Cada entrada na rede neural é associada a um peso e calcula-se a função das entradas multiplicando-as pelos pesos intermediários até o neurônios de saída. A aprendizagem se dá ajustando os valores dos pesos intermediários até que se alcance a resposta esperada.

### 2.2.1 Perceptron

Perceptron (ROSENBLATT, 1958) é o primeiro e mais simples modelo de rede neural conhecido, como ilustrado na Figura 3. Contendo apenas uma única camada, foi concebido para a classificação de dados linearmente separáveis. O modelo calcula a saída multiplicando as entradas por seus respectivos pesos e por uma função de ativação. O algoritmo repete essa operação percorrendo os dados de treinamento e ajustando os pesos até que haja uma convergência, ou seja, até que a rede classifique corretamente todos os dados de entrada (AGGARWAL et al., 2018).

A função de ativação é responsável por rotular os dados de entrada em uma das classes. Com um único neurônio, classifica os dados em duas classes. Com  $n$  neurônios, classifica em  $2^n$  classes.

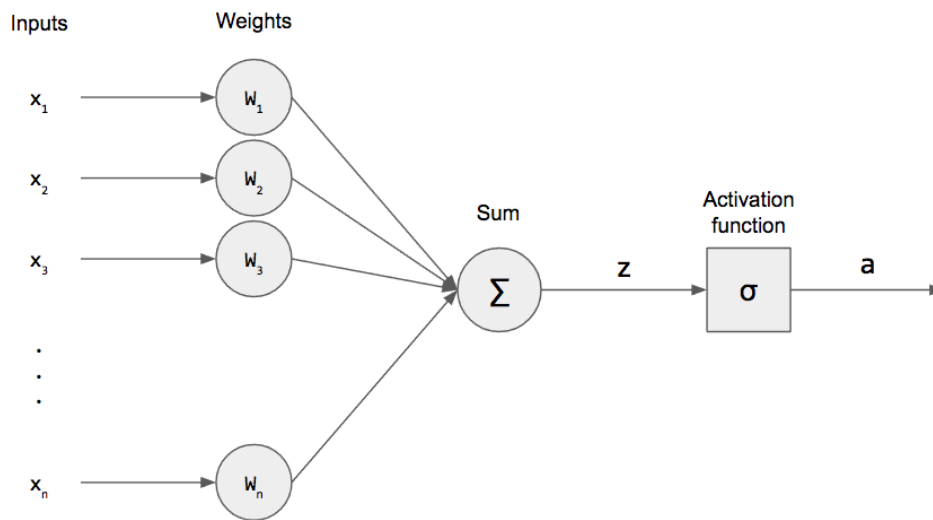


Figura 3 – Rede neural Perceptron (DESHPANDE, 2020): da esquerda para a direita: entradas (*inputs*), pesos (*weights*), somatório (*sum*) e função de ativação (*activation function*)

### 2.2.2 Perceptron Multicamadas

A rede neural Perceptron se limita a problemas de classificação linear, nos quais é possível dividir as classes de dados, como mostrado no lado esquerdo da Figura 4. O Perceptron Multicamadas abrange também casos onde os dados não são linearmente separáveis, como o lado direito da Figura 4.

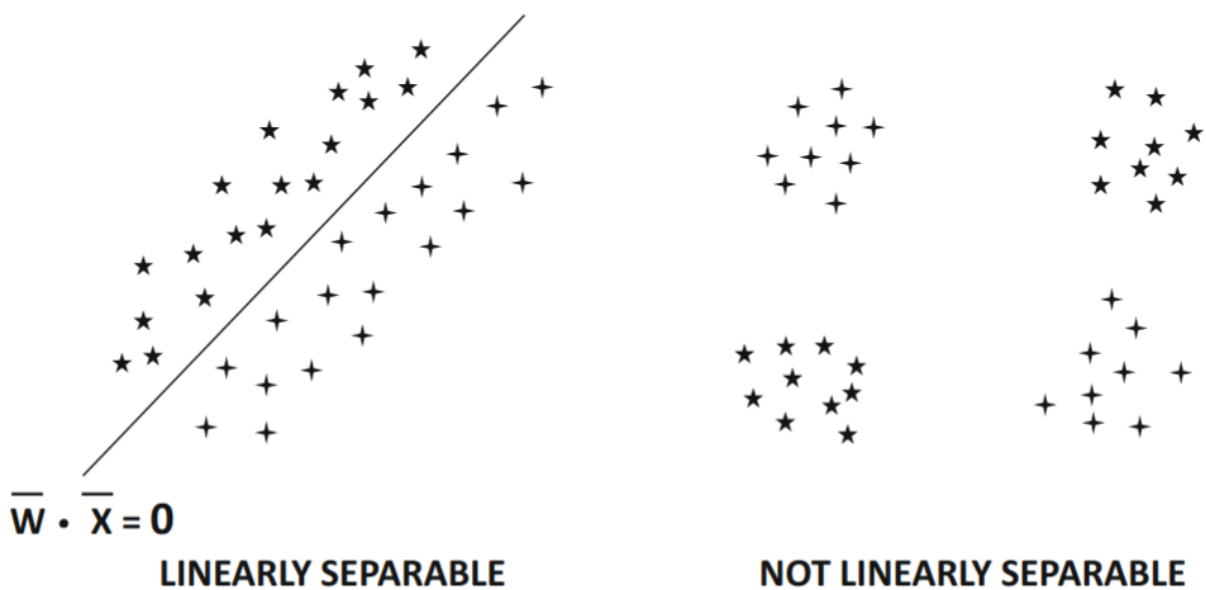


Figura 4 – Exemplo de dados lineares e não linearmente separáveis (AGGARWAL et al., 2018)

A arquitetura de uma rede neural multicamadas é também conhecida como

*feed-forward*. Isso significa que as camadas sucessivas alimentam umas as outras na direção da entrada para a saída, assumindo que todas as camadas estão conectadas à camada seguinte (AGGARWAL et al., 2018).

Perceptron Multicamadas possui camadas intermediárias, chamadas de camadas escondidas. A saída de cada camada é o cálculo da combinação linear das entradas com os respectivos pesos, e é usada como entrada para as camadas seguintes. Ao final, a função de ativação é aplicada ao resultado de todas as combinações lineares.

O algoritmo de retropropagação (GOODFELLOW; BENGIO; COURVILLE, 2016), ou *backpropagation*, é um método que aplica o gradiente da função de perda de cada peso, em cada camada, propagando da última camada até a primeira. Os pesos de cada camada são então ajustados para que se obtenha um erro mínimo em relação aos resultados esperados. O algoritmo de retropropagação torna viável utilizar métodos de gradiente, como o Gradiente Descendente Estocástico, para treinar redes neurais multicamadas, atualizando os pesos para minimizar a perda.

## 2.3 Redes Neurais Recorrentes

Redes Neurais Recorrentes (RNNs) são um conjunto de redes neurais utilizadas para o processamento sequencial de dados (RUMELHART; HINTON; WILLIAMS, 1986). Segundo Graves (2012), uma rede neural multicamadas pode mapear apenas entradas com saídas, enquanto que RNNs guardam um histórico das entradas anteriores para cada saída. A representação de uma RNN com apenas uma camada escondida é mostrada na Figura 6.

Em uma RNN, a camada da rede corresponde a um elemento da sequência de dados. A rede contém um número variável de camadas, e cada camada possui uma única entrada correspondendo a um instante de tempo. Cada camada também possui um conjunto fixo de parâmetros utilizados repetidamente no decorrer do tempo (AGGARWAL et al., 2018).

Dessa forma, uma entrada como *"the cat chased the mouse"* tem uma representação geral mostrada na Figura 6a e uma representação temporal mostrada na Figura 6b. O

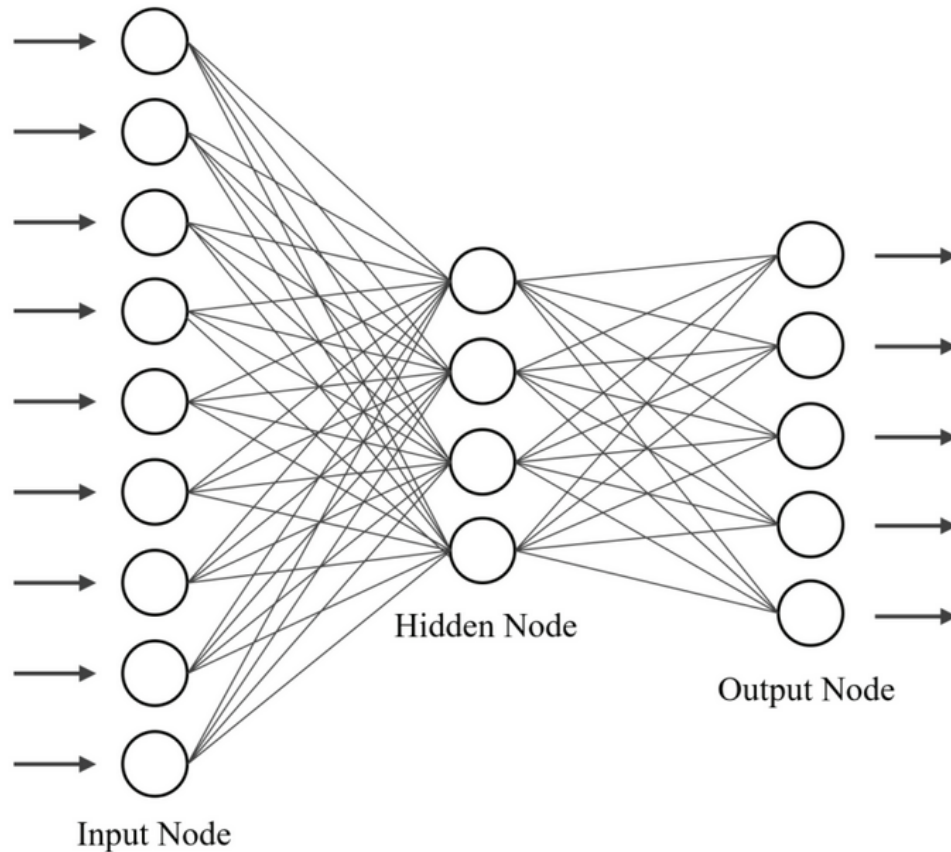


Figura 5 – Rede neural Perceptron Multicamadas (KANG; CHOI, 2021): da esquerda para a direita: nós de entrada (*input node*); nós escondidos (*hidden node*); nós de saída (*output node*)

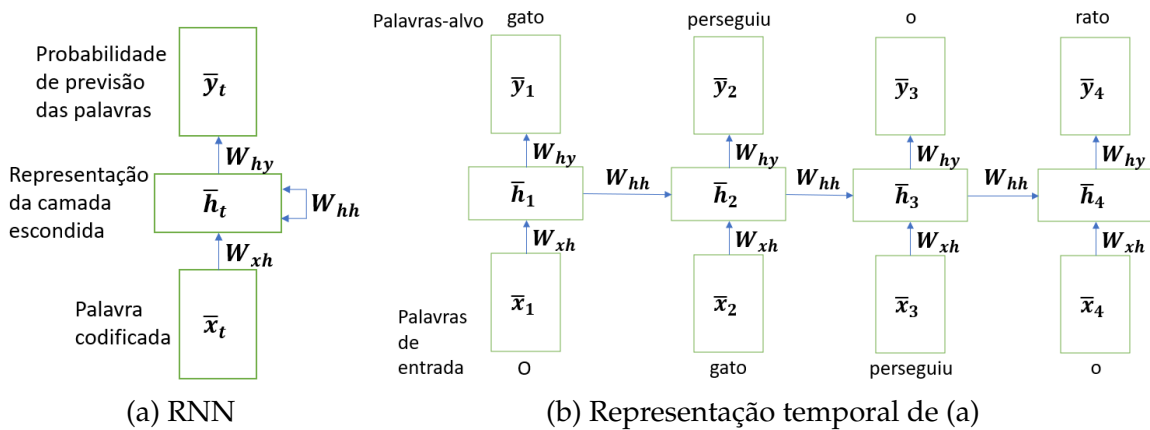


Figura 6 – Rede neural recorrente e sua representação temporal. Baseado em Goodfellow, Bengio e Courville (2016)

primeiro elemento da sequência – *the* – é processado pela camada. O próximo elemento – *cat* – é então processado na mesma camada, porém com informações agregadas do elemento anterior. Esse processo ocorre de forma recorrente até que todos os elementos da entrada sejam processados.

Essa abordagem se torna especialmente importante quando alguma informação puder ocorrer em diferentes posições. Enquanto que uma rede neural *feedforward* precisaria aprender todas as combinação de entrada, uma rede neural recorrente compartilha o peso de cada camada nos instantes de tempo (GOODFELLOW; BENGIO; COURVILLE, 2016).

### 2.3.1 *Long Short-Term Memory – LSTM*

Redes recorrentes são utilizadas para treinar dados onde o histórico das entradas é relevante para a saída. Contudo, a influência da entrada na camada de saída é bastante elevada nos primeiros instantes, mas decai consideravelmente com o passar do tempo. A rede neural recorrente *LSTM* (HOCHREITER; SCHMIDHUBER, 1997) traz como solução um algoritmo baseado em gradiente que busca manter constante o erro nas camadas internas.

A arquitetura consiste em um conjunto de blocos de memória, com uma ou mais células autoconectadas, e três portas – *input gate*, *output gate* e *forget gate*, como mostrado na Figura 7. Enquanto que na *RNN* todo o conteúdo é sobrescrito em cada instante de tempo, uma rede *LSTM* consegue manter algumas informações que considere importantes. As portas, portanto, têm a função de armazenar características dos dados por um longo período de tempo (GRAVES, 2012).

### 2.3.2 *Bidirectional Long Short-Term Memory – BLSTM*

A *RNN* pura, assim como a *LSTM*, analisa uma sequência de dados em uma única direção. Contudo, em alguns casos se faz necessário analisar tanto elementos antecedentes quanto sucessores. Por exemplo, a tradução de um texto envolve a análise do contexto ao qual estão inseridas as palavras, ou seja, palavras anteriores e posteriores influenciam na tradução de determinada palavra ou sentença.

Uma rede *RNNs* bidirecional (SCHUSTER; PALIWAL, 1997) aborda esse problema treinando a entrada em ambas as direções usando duas camadas recorrentes

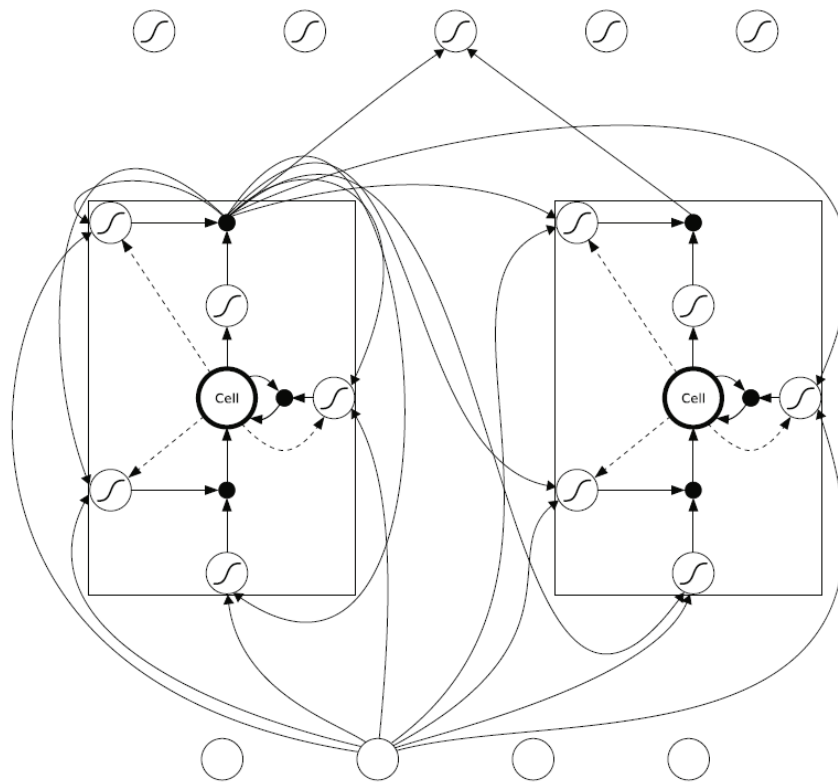


Figura 7 – Arquitetura de uma rede recorrente *LSTM* (GRAVES, 2012)

escondidas conectadas com a mesma camada de saída. *LSTM* bidirecional (GRAVES; SCHMIDHUBER, 2005) combina a arquitetura de uma rede *LSTM* com as características do processamento bidirecional.

### 2.3.3 Gated Recurrent Unit – GRU

Uma Unidade Recorrente Fechada (*Gated Recurrent Unit – GRU*) é uma *RNN* que se assemelha bastante a *LSTM*, pois ambas guardam características dos dados durante um período de tempo, mas diferem ao usar apenas duas portas para controlar as informações armazenadas: de atualização e de reinicialização.

A rede *GRU* pode ser vista como uma simplificação da rede *LSTM* (GOODFELLOW; BENGIO; COURVILLE, 2016) – enquanto que na *LSTM* a porta de saída controla quais informações serão usadas para o processamento em cada instante de tempo, a rede *GRU* expõe sempre todo o conteúdo armazenado. Segundo (CHUNG et al., 2014), muito embora as redes *GRU* e *LSTM* possuem uma performance semelhante, ambas

possuem resultados mais promissores que uma rede *RNN* tradicional.

## 2.4 Regularização

As entradas das redes neurais são divididas em dois grupos: conjunto de treinamento e conjunto de teste. O conjunto de treinamento é utilizado para que a rede ajuste os pesos de modo a aprender quais são as características que definem a base de dados. O conjunto de teste, por sua vez, é usado como parâmetro para definir, ao final do treinamento, se a rede neural conseguiu efetivamente extrair as características corretas.

O objetivo das redes neurais é construir uma arquitetura de rede que consiga identificar elementos comuns em uma base de dados de modo a caracterizá-la. Hagan, Demuth e Beale (2014) afirmam que a rede com uma boa generalização possui uma boa performance tanto no conjunto de treino quanto no conjunto de teste. Um dos maiores desafios das redes neurais é encontrar o tamanho e complexidade da rede ideais para generalizar de forma satisfatória.

Tamanho e complexidade exagerados, por exemplo, podem ter um efeito adverso. Nesse caso, a rede tem uma performance excelente no conjunto de treinamento mas apresenta dificuldade em identificar os padrões em novos conjuntos de dados. Esse cenário é conhecido como *sobreajuste*, ou *overfitting*, e significa que a rede não aprendeu de fato a reconhecer os padrões de informação, apenas memorizou o comportamento da base de dados.

Por outro lado, *subajuste* ou *underfitting*, ocorre quando a rede tem performance baixa tanto no conjunto de treinamento quanto no conjunto de teste. Nesse caso, pode ser que tenha sido usada uma arquitetura inadequada ou com tamanho inferior ao necessário para a generalização. Na Figura 8 podemos ver a diferença entre uma rede com *overfitting*, *underfitting* e uma com boa generalização. A seguir serão vistas técnicas de regularização, úteis para reduzir erros de generalização.



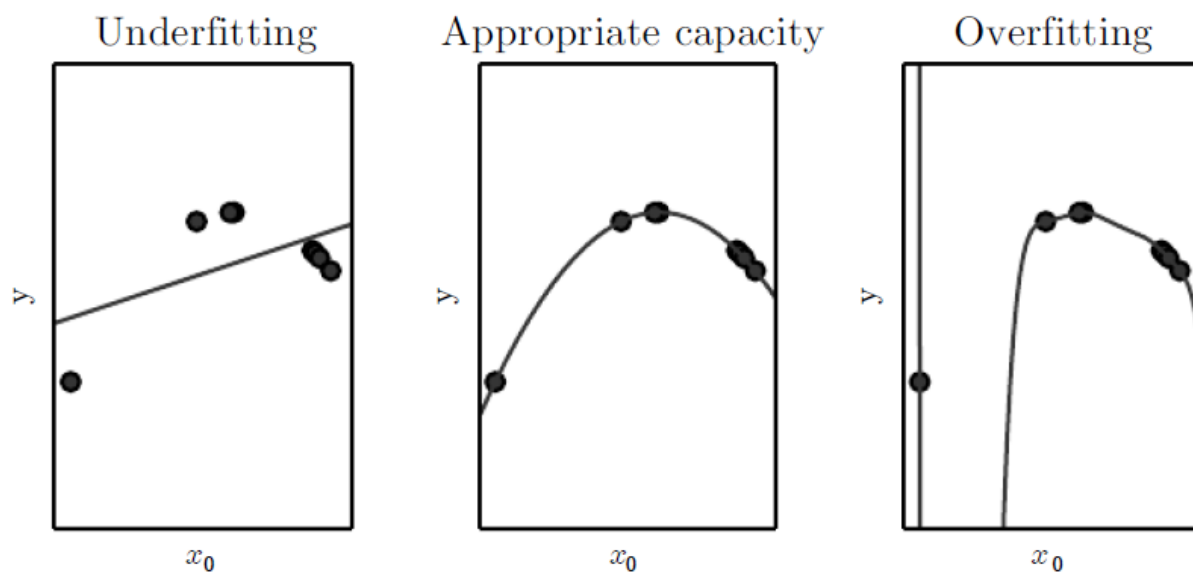


Figura 8 – Exemplo de três redes com performance de generalização diferentes (GOOD-FELLOW; BENGIO; COURVILLE, 2016): figura da esquerda: sub-treinamento (*underfitting*); figura do meio: rede com boa generalização; figura da direita: sobre treinamento (*overfitting*)

### 2.4.1 Parada antecipada

Sabe-se que existe uma região crítica de número de épocas no qual a rede generaliza da melhor forma possível, e após essa região o erro de generalização aumenta (WANG; VENKATESH; JUDD, 1993). O sobre-treinamento, ou *overtraining*, ocorre quando se treina a rede por muito mais épocas que o necessário, e, por consequência, aumenta sua complexidade. Uma alternativa simples para melhoria de generalização é parar antecipadamente o treinamento quando for identificado que a rede já atingiu a região crítica (HAGAN; DEMUTH; BEALE, 2014).

Um conjunto de validação é usado junto com o conjunto de treinamento com o intuito de identificar o momento ideal para interromper o treinamento antecipadamente. A ideia do conjunto de validação é monitorar a performance da rede em cada iteração; se diminuir por várias épocas, a rede atingiu a região crítica e o treinamento pode ser parado.

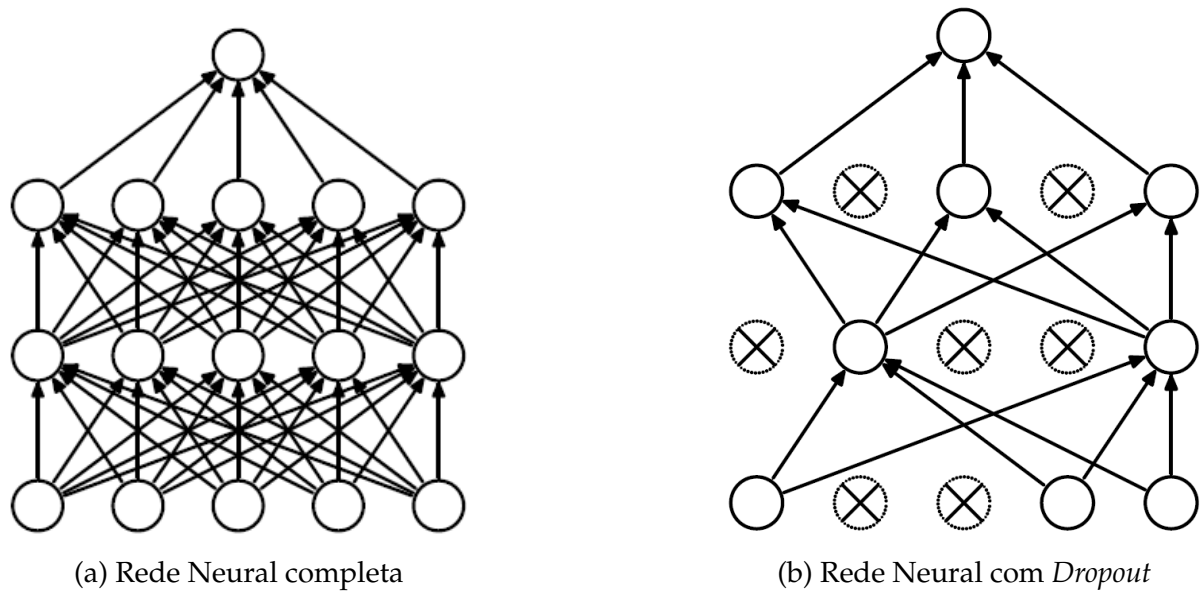


Figura 9 – Comparação entre uma rede neural completa com a aplicação da técnica de regularização *Dropout* (SRIVASTAVA et al., 2014)

### 2.4.2 *Dropout*

Uma rede neural muito complexa, com o número de neurônios acima do necessário, também pode sofrer de *overfitting*. *Dropout* é um método de regularização que remove aleatoriamente neurônios da rede neural durante o treinamento para evitar co-adaptação entre neurônios vizinhos (SRIVASTAVA et al., 2014). Um exemplo de remoção de alguns neurônios durante o treinamento comparado com a rede neural completa está ilustrado na Figura 9.

Alguns neurônios escolhidos de forma aleatória, bem como suas conexões de entrada e saída, são removidos da rede, tornando-a bem menos complexa durante o treinamento. A rede neural é então treinada com as várias sub-redes formadas removendo neurônios intermediários. A rede permanece completa durante a fase de teste, porém os pesos dos neurônios são escalonados conforme a taxa de *Dropout*, um parâmetro indicando a probabilidade de treinar de fato um nó em uma camada.

### 2.4.3 Decaimento de peso

Além da quantidade de camadas e neurônios da rede e a quantidade de épocas de treinamento, outra característica que pode influenciar na generalização precária de uma

rede neural é o valor dos pesos dos neurônios. Uma forma de reduzir a complexidade da rede é por meio do decaimento de peso, ou *weight decay* – a ideia é prevenir que o valor dos pesos dos neurônios aumente desnecessariamente por meio de uma função de regularização que escalona valores muito altos (KROGH; HERTZ, 1991).

Existem dois métodos principais de decaimento de peso: Regularização  $L1$  e  $L2$ . Enquanto que a Regularização  $L1$  reduz a complexidade da rede anulando os menores pesos de forma a reduzir o número de características relevantes para o aprendizado, a Regularização  $L2$  escalona os pesos de modo a impedir que alguma característica se torne irrelevante (GOLDBERG; HIRST, 2017).

## 2.5 Otimização

No contexto de aprendizagem de máquina, a função de custo é utilizada para avaliar o quão bom um modelo é em prever o conjunto de dados. A função de custo ( $J$ ) compara as previsões da rede com a classificação de fato do conjunto de treinamento. A otimização no contexto de redes neurais tem como objetivo ajustar os parâmetros para que se reduza ao máximo a função de custo, aumentando a probabilidade de melhorar a performance da rede.

O Gradiente Descendente Estocástico é um método iterativo que ajusta os pesos  $w$  de modo a reduzir a diferença entre as previsões da rede  $\hat{y}$  e das classificações corretas  $y$ . Ou seja, o método procura um mínimo global ou local da função de custo  $J(y, \hat{y})$  em função dos pesos da rede neural. Para isso, escolhe uma coordenada inicial da função de custo e itera na direção do mínimo local ajustando os pesos relativos àquele ponto, tendo como base a primeira derivada (BUDUMA; LOCASCIO, 2017). O algoritmo se repete até que o erro se torne consideravelmente pequeno.

Variações do algoritmo de Gradiente Descendente Estocástico podem considerar também as atualizações de peso das iterações anteriores para determinar as atualizações futuras, como os métodos *Momentum* (SUTSKEVER et al., 2013) e *Adam* (KINGMA; BA, 2014). Outros métodos buscam adaptar a taxa de aprendizado durante o treinamento de acordo com alguns parâmetros da rede, como *AdaGrad* (DUCHI; HAZAN; SINGER,

2011) e *RMSPprop* (BUDUMA; LOCASCIO, 2017).

## 2.6 Medidas de Desempenho

As medidas de desempenho em algoritmos de aprendizagem de máquina são essenciais para que as diversas arquiteturas, métodos de otimização e regularização sejam comparadas entre si a fim de alguma combinação de técnicas ser escolhida. Além disso, é importante para acompanhar o progresso ou não em ajustes na arquitetura, como número de camadas, taxas de regularização ou algoritmo de otimização. Goodfellow, Bengio e Courville (2016) define algumas das medidas de desempenho utilizadas em algoritmos de aprendizagem de máquina:

- **Acurácia:** proporção de exemplos que o modelo classificou corretamente;
- **Perda:** proporção de exemplos que o modelo classificou incorretamente;
- **Precisão:** fração de detecções corretas reportadas pelo modelo;
- **Sensibilidade:** fração de eventos verdadeiros que foram detectados. Avalia a capacidade de detectar resultados positivos;
- **Especificidade:** fração de eventos negativos que foram detectados como negativo. Avalia a capacidade de detectar resultados negativos.

### 3

---

## RECONHECIMENTO DE ATIVIDADES HUMANAS

A microeletrônica e sistemas computacionais têm se desenvolvido de forma bastante promissora durante os últimos anos, tornando dispositivos como sensores e *smartphones*, cada vez menores, mais poderosos e baratos. Esse avanço tornou possível o desenvolvimento de Ambientes Inteligentes, um paradigma com o propósito de tornar ambientes adaptáveis e inteligentes a fim de melhorar a qualidade de vida das pessoas inseridas no ambiente, seja resolvendo tarefas de forma mais eficiente, ou oferecendo assistência mais rápida com maior conforto (ORTIZ, 2015).

Dentro do contexto de Ambientes Inteligentes, há um grande interesse em coletar e interpretar informações a partir de dados coletados de sensores de forma não intrusiva, área conhecida como Sensoriamento Onipresente, ou *Ubiquitous Sensing* (PAULOVICH; OLIVEIRA; JR, 2018). De acordo com Ortiz (2015), tem se tornado vantajoso Ambientes Inteligentes para atender ao crescimento populacional e recursos humanos e tecnológicos limitados.

Também com o intuito de promover o bem-estar, o RAH surgiu como o processo de observar e analisar o comportamento de um indivíduo por meio de sensores a fim de classificar quais atividades ele está realizando (CHEN et al., 2011). Reconhecer atividades como andar, correr, e subir ou descer escadas são interessantes para acompanhamento de pacientes com diabetes, obesidade ou problemas cardíacos, visto que tais atividades são uma parte essencial do tratamento (PRAET; LOON, 2009; FOCK; KHOO,

2013).

O RAH pode ser feito por meio de sensores externos, como câmeras em locais estratégicos, ou sensores vestíveis, caracterizados por seu tamanho menor e acoplados ao corpo ou vestimenta do usuário. Segundo Sazonov (2014), sensores vestíveis podem estar associados às seguintes funcionalidades: responder a um determinado estímulo físico ou químico, processar, armazenar, transmitir e tomar alguma ação. Dependendo da aplicação, a execução desses funcionalidades podem ocorrer tanto no dispositivo sensorial quanto de forma remota.

### 3.1 *Smartphones* e sensores

Os *Smartphones* são dispositivos vestíveis bastante utilizados, especialmente durante a execução de atividades diárias (BULBUL; CETIN; DOGRU, 2018; BAYAT; POMPLUN; TRAN, 2014; RONAO; CHO, 2016). Os *Smartphones* surgiram como uma nova geração de Telefones Celulares devido ao aumento da necessidade de acesso instantâneo a qualquer tipo de informação, com mais capacidade computacional, hardware integrado, e serviços de comunicação sem fio (ORTIZ, 2015). Além disso, possuem diversos sensores embarcados, como microfone, câmeras, acelerômetros e giroscópios, de modo a facilitar o seu uso para fins de acompanhamento de atividades diárias.

Dentre os principais sensores integrados nos *Smartphones* úteis para RAH, estão os sensores inerciais acelerômetro e o giroscópio. O acelerômetro é um instrumento que mede a aceleração física de um objeto. Além de estar embarcado em quase todos os dispositivos móveis atuais, possui um baixo consumo de energia (PENDÃO; MOREIRA; RODRIGUES, 2014) e é bastante usado para detecção de atividades humanas, como caminhar, correr, sentar, dentre outros (CASALE; PUJOL; RADEVA, 2011; LEE; YOON; CHO, 2017; BAYAT; POMPLUN; TRAN, 2014). Em dispositivos eletrônicos, como os *Smartphones*, normalmente há um único circuito integrado com três acelerômetros dispostos ortogonalmente, um para cada dimensão espacial (ORTIZ, 2015).

O giroscópio mede a orientação e velocidade angular de determinado objeto. O giroscópio e acelerômetro se tornaram populares miniaturizados por meio da tec-

nologia *MEMS* (Microelectromechanical systems – Sistemas microeletromecânicos). O giroscópio utiliza uma estrutura de vibração para inferir a velocidade e sentido de rotação nos eixos ortogonais (KWON; SEOK; LIM, 2017). O acelerômetro pode utilizar a piezoeletricidade ou a detecção de mudanças na capacitância (ANDREJAŠIĆ, 2008).

Acelerômetro e giroscópio podem ser utilizados em conjunto para RAH. Em Reyes-Ortiz et al. (2012), foi publicada uma base de dados de ambos os sensores oriundos de um *smartphone*, composto por atividades diárias realizadas por 30 participantes. A base de dados já foi utilizada para comparar modelos modificados de SVN (ANGUITA et al., 2013), detecção de distúrbios de movimento (REYES-ORTIZ et al., 2013) e reconhecimento de mudanças de postura (REYES-ORTIZ et al., 2014). Este trabalho tem como foco o RAH por meio de sensores vestíveis oriundos de um *smartphone*.

## 3.2 Etapas

Segundo Bulling, Blanke e Schiele (2014), os estágios de processamento para RAH são: aquisição de dados, segmentação e pré-processamento de sinais, seleção e extração de características, treinamento e classificação. Outros autores definem as etapas de forma semelhante. Para Ortiz (2015), por exemplo, as etapas principais são sensoriamento, pré-processamento de dados, discriminação de características, treinamento e classificação.

No primeiro estágio, aquisição de dados ou sensoriamento, os dados são coletados a partir de diferentes sensores localizados no corpo do indivíduo. Os dados são então pré-processados a fim de identificar os segmentos que contém de fato informações relacionadas às atividades. Nessa etapa também são realizados os filtros para remoção de ruídos que não fazem parte da informação. Após a filtragem dos dados, é realizada a extração de características dos sinais consideradas relevantes para a classificação das atividades. Finalmente, o conjunto de dados coletados são treinados utilizando algoritmos de aprendizagem de máquina. Diferentes arquiteturas são comparadas até que se encontre a que mais se adéqua às medidas de avaliação de desempenho.

### 3.3 Base de dados *UniMiB SHAR*

A base de dados *UniMiB SHAR* (MICUCCI; MOBILIO; NAPOLETANO, 2017) teve como motivação o escasso conjunto de dados disponíveis publicamente relacionados a detecção de atividades humanas e diferentes tipos de queda a partir de dados de acelerômetros, a fim de ser usada como uma base comum para avaliação de diferentes técnicas de aprendizado de máquina. A base reúne um conjunto de amostras de dados de acelerômetros adquiridos por um *smartphone Android*, divididas entre 17 tipos de atividades humanas e tipos de quedas realizadas por 30 indivíduos entre 18 a 60 anos, descritas na Tabela 1.

Tabela 1 – Atividades humanas disponíveis na base de dados *UniMiB SHAR*

Grupo	Nome	Descrição
Atividades	Levantar estando deitado	Ficar em pé estando previamente deitado
	Deitar estando levantado	Deitar estando previamente em pé
	Levantar estando sentado	De estar sentado em uma cadeira para levantar
	Correr	Corrida moderada
	Sentar	Sentar estando previamente em pé
	Subir escadas	Subir escadas moderadamente
	Descer escadas	Descer escadas moderadamente
	Andar	Andar normalmente
	Pular	Pular continuamente
Quedas	Cair para trás	Cair para trás enquanto sentado em uma cadeira Cair para trás estando em pé
	Cair para frente	Cair prevenindo o impacto por compensação Cair para a frente estando em pé, apoiando com as mãos
	Cair para a lateral	Cair para a lateral direita estando em pé Cair para a lateral esquerda estando em pé
	Cair tropeçando Desmaio	Cair tropeçando em um objeto antes de atingir o solo Perda súbita de consciência

Os dados foram coletados de 24 indivíduos do sexo feminino e 6 do sexo masculino. Cada amostra da base de dados é um conjunto de 151 valores do acelerômetro de cada eixo,  $x$ ,  $y$  e  $z$ , correspondendo a aproximadamente três segundos de coleta. O desempenho de quatro classificadores foram comparados: K-Vizinhos Mais Próximos, Máquina de Vetores de Suporte, e Floresta Aleatória. Com base na características dos va-



lores de acelerômetro dos diferentes tipos de atividades e quedas, dois comportamentos foram identificados: a classificação de diferentes atividades é mais simples do que a de diferentes tipos de quedas e é fácil distinguir entre atividades de queda e de não queda.

### 3.4 Trabalhos relacionados

Os trabalhos relacionados abaixo propuseram diferentes arquiteturas para Reconhecimento de Atividades Humanas com a base de dados *UniMiB SHAR* utilizando Redes Neurais Recorrentes, foco deste trabalho.

Alsarhan et al. (2019) obteve resultados promissores com uma arquitetura de rede recorrente *GRU* bidirecional. O modelo *BiGRU* proposto é semelhante ao *BLSTM*, consistindo em duas camadas de *GRU* distintas, cada uma delas responsável por uma direção no tempo. Logo depois, em Alawneh et al. (2020), os autores propuseram uma comparação entre a arquitetura *LSTM* e *BLSTM* em duas bases de dados: *UniMiB SHAR* e *WISDM* (CHEN et al., 2017). Em ambas, utilizou-se o algoritmo de otimização *Adam* e o método *Dropout* de regularização. Verificou-se que a arquitetura *BLSTM* obteve resultados melhores para a diferenciar entre atividades diárias e quedas, mas ambos tiveram performance baixa para comparar entre diferentes tipos de queda. De forma geral, concluiu-se que a arquitetura *BLSTM* obteve acurácia sutilmente melhor para os dois conjuntos.

Já em Serrão et al. (2021), foram comparadas três arquiteturas: rede neural *CNN*, e as redes recorrentes *LSTM* e *GRU*. Para a rede neural *CNN*, foram propostos métodos para transformar os dados de sensores em imagens, visto que possui aplicabilidade maior em tarefas relacionadas a visão computacional. Nesse trabalho, as arquiteturas foram avaliadas a partir dos métodos de *leave-one-subject out* e *cross validação 5-Fold*.

## 4

---

# METODOLOGIA

### 4.1 Arquiteturas de Redes Neurais Recorrentes

Este trabalho avaliou três arquiteturas de Redes Neurais Recorrentes, *LSTM*, *BLSTM* e *GRU*, para Reconhecimento de Atividades Humanas. Cada arquitetura de rede foi avaliada sob quatro condições: (1) sem técnica de regularização, (2) com o método de *Dropout*, (3) com o método de Regularização L2, e (4) com os métodos *Dropout* e Regularização L2 em conjunto. Cada cenário foi avaliado com os métodos de otimização Adam, RMSProp e SGDM, totalizando 36 arquiteturas de rede diferentes. As redes neurais foram comparadas entre si partir dos resultados obtidos de acurácia e perda.

Todas as arquiteturas de rede possuem quatro camadas, com 512, 256, 128 e 32 neurônios, da entrada para a saída, nessa ordem, e utilizam a função de ativação *softmax*, função de perda *Categorical Cross-Entropy* e o método de parada antecipada. Nas arquiteturas com Regularização L2 e *Dropout*, foram utilizadas as taxas de  $10^{-5}$  e 0,2, respectivamente. Para o método de otimização *RMSProp*, utilizou-se a taxa de momento 0,1, e para *SGDM*, 0,02. Além disso, as redes foram treinadas com 100 épocas no total.

### 4.2 Base de Dados

A base de dados *UniMiB SHAR* (MICUCCI; MOBILIO; NAPOLETANO, 2017) foi utilizada da seguinte forma: selecionou-se nove atividades diárias dentre as 15 e sete dos 30 indivíduos. No início de cada treinamento selecionou-se, de forma aleatória, 85%

das amostras para o conjunto de treinamento, e as 15% restantes para o conjunto de teste e validação.

Os dados foram dispostos em dois arquivos *CSV*, um contendo os valores de cada amostra e outro informando as atividades correspondentes de cada amostra. Os valores das amostras estão dispostos em 1724 colunas e 453 linhas. Cada coluna corresponde aos dados de uma atividade e as linhas possuem os dados dos três acelerômetros – 151 valores para cada eixo. Os rótulos de cada amostra, por sua vez, possuem 1724 colunas e uma linha, onde cada coluna informa a atividade correspondente àquela amostra. Na Tabela 2 estão dispostas as nove atividades diárias utilizadas da base de dados *UniMiB SHAR*.

Tabela 2 – Atividades diárias utilizadas para classificação da base de dados *UniMiB SHAR*

<b>Grupo</b>	<b>Nome</b>	<b>Descrição</b>
<b>Atividades</b>	Levantar estando deitado	Ficar em pé estando previamente deitado
	Deitar estando levantado	Deitar estando previamente em pé
	Levantar estando sentado	De estar sentado em uma cadeira para levantar
	Correr	Corrida moderada
	Sentar	Sentar estando previamente em pé
	Subir escadas	Subir escadas moderadamente
	Descer escadas	Descer escadas moderadamente
	Andar	Andar normalmente
	Pular	Pular continuamente

### 4.3 Recursos utilizados

Este trabalho teve como recursos utilizados os arquivos *CSV* relacionados à base de dados e a plataforma *Google Colaboratory* para hospedagem e treinamento das redes neurais, bem como para a elaboração dos gráficos correspondentes aos treinamentos. *Google Colaboratory* disponibiliza um ambiente de execução com as seguintes características: GPU Tesla T4, Intel(R) Xeon(R) CPU @ 2.20GHz, 56320 KB de memória cache L3, 13 GB de memória RAM e 40 GB em espaço de armazenamento.

O treinamento das redes neurais na plataforma *Google Colaboratory* se deu por

meio da biblioteca de código aberto *TensorFlow*, criada para aprendizado de máquina com atuação em aprendizado profundo, aplicação de algoritmos de pré-processamento e métodos de avaliação. Os recursos do *TensorFlow* foram acessados por meio da biblioteca de código aberto *Keras*, escrita na linguagem de programação Python.

## 5

---

## RESULTADOS E DISCUSSÃO

Os resultados obtidos para as arquiteturas *LSTM*, *GRU* e *BLSTM* estão dispostos nas tabelas 3, 4 e 5, respectivamente. As redes foram citadas a partir de uma sigla com a inicial de suas características, nessa ordem: arquitetura da rede (L, G, e B), método de otimização (A, R, S), e métodos de melhoria de generalização (S, L, D, DL). Por exemplo, a arquitetura de rede *LSTM*, com método de otimização *Adam*, sem melhoria de otimização, será citada como *LAS*.

Tabela 3 – Perda e acurácia da rede *LSTM*

	Otimização	Regularização	Perda	Acurácia
<i>LSTM</i>	<b>Adam</b>	<i>Sem Melhoria</i>	0,1074	0,9653
		<i>L2</i>	0,1644	0,9653
		<i>Dropout</i>	0,2570	0,9498
		<i>Dropout e L2</i>	0,2617	0,9344
	<b>RMSProp</b>	<i>Sem Melhoria</i>	0,1002	0,9653
		<i>L2</i>	0,2698	0,9421
		<i>Dropout</i>	0,2415	0,9305
		<i>Dropout e L2</i>	0,2654	0,9344
	<b>SGDM</b>	<i>Sem Melhoria</i>	0,3861	0,8764
		<i>L2</i>	0,3472	0,9266
		<i>Dropout</i>	0,1372	0,9614
		<i>Dropout e L2</i>	0,2966	0,9382

Tabela 4 – Perda e acurácia da rede GRU

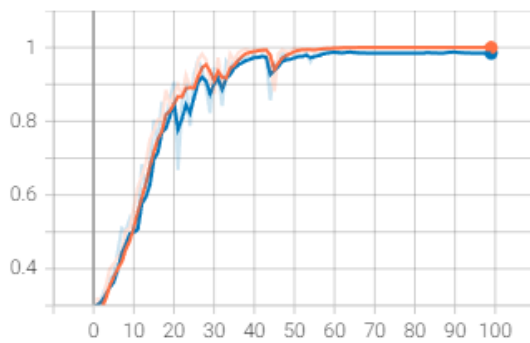
	Otimização	Regularização	Perda	Acurácia
GRU	Adam	<i>Sem Melhoria</i>	0,1435	0,9730
		<i>L2</i>	0,1232	0,9768
		<i>Dropout</i>	0,0871	0,9807
		<i>Dropout e L2</i>	0,1096	0,9768
	RMSProp	<i>Sem Melhoria</i>	0,0997	0,9846
		<i>L2</i>	0,1446	0,9653
		<i>Dropout</i>	0,0853	0,9807
		<i>Dropout e L2</i>	0,2313	0,9421
	SGDM	<i>Sem Melhoria</i>	0,1260	0,9575
		<i>L2</i>	0,5783	0,8649
		<i>Dropout</i>	0,1423	0,9498
		<i>Dropout e L2</i>	0,8161	0,7954

Tabela 5 – Perda e acurácia da rede BLSTM

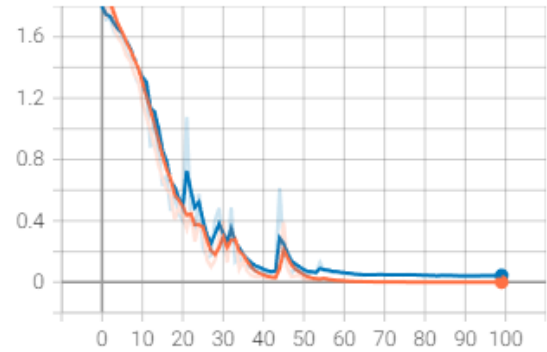
	Otimização	Regularização	Perda	Acurácia
BLSTM	Adam	<i>Sem Melhoria</i>	0,7370	0,9846
		<i>L2</i>	0,2085	0,9653
		<i>Dropout</i>	0,0869	0,9807
		<i>Dropout e L2</i>	0,1225	0,9807
	RMSProp	<i>Sem Melhoria</i>	0,0439	0,9846
		<i>L2</i>	0,1711	0,9691
		<i>Dropout</i>	0,0469	0,9846
		<i>Dropout e L2</i>	0,1564	0,9768
	SGDM	<i>Sem Melhoria</i>	0,1956	0,9614
		<i>L2</i>	0,3689	0,9421
		<i>Dropout</i>	0,1073	0,9653
		<i>Dropout e L2</i>	0,4806	0,9035

De acordo com os resultados, a rede BRS obteve a maior acurácia, 0,9846%, e menor taxa de perda, 0,0439%. Por outro lado, a arquitetura de rede GSDL obteve a acurácia mais baixa, 0,7954%, e taxa de perda mais alta, 0,8161%. Os gráficos de perda e acurácia de ambas as redes estão dispostos nas figuras 10 e 11, respectivamente.

Observa-se também que os gráficos da arquitetura *BRS* são bem mais estáveis que a arquitetura *GSDL*.

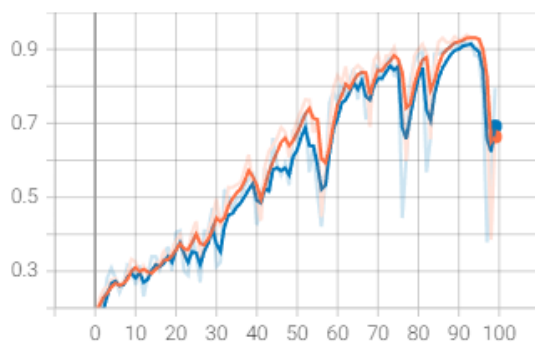


(a) Gráfico da acurácia

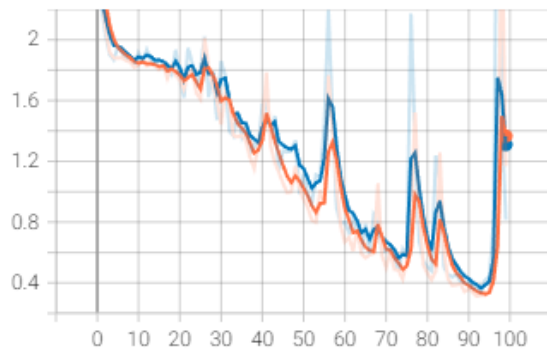


(b) Gráfico da perda

Figura 10 – Gráfico da acurácia e da perda em função das épocas para a arquitetura de rede *BRS*



(a) Gráfico da acurácia



(b) Gráfico da perda

Figura 11 – Gráfico da acurácia e da perda em função das épocas para a arquitetura de rede *GSDL*

Na tabela 6 estão dispostas as 10 arquiteturas de rede que obtiveram melhor desempenho dentre as demais, considerando as métricas de perda e acurácia. Na tabela 7, por outro lado, estão dispostos os resultados das redes com pior desempenho. A arquitetura *LSTM*, de forma geral, mostrou resultados inferiores na maioria das combinações, exceto quando combinada com o algoritmo *RMSProp* e sem técnica de regularização.

Tabela 6 – Perda e acurácia das arquiteturas de rede com maior performance

<b>Arquitetura</b>	<b>Perda</b>	<b>Acurácia</b>
<i>BRS</i>	0,0439	0,9846
<i>BRD</i>	0,0469	0,9846
<i>GRD</i>	0,0853	0,9807
<i>BAD</i>	0,0869	0,9807
<i>GAD</i>	0,0871	0,9807
<i>GRS</i>	0,0997	0,9846
<i>GADL</i>	0,1096	0,9768
<i>LRS</i>	0,1002	0,9653
<i>BADL</i>	0,1225	0,9807
<i>BSD</i>	0,1073	0,9653

Tabela 7 – Perda e acurácia das arquiteturas de rede com menor performance

<b>Arquitetura</b>	<b>Perda</b>	<b>Acurácia</b>
<i>GSDL</i>	0,8161	0,7954
<i>BAS</i>	0,7370	0,9846
<i>GSL</i>	0,5783	0,8649
<i>BSDL</i>	0,4806	0,9035
<i>LSS</i>	0,3861	0,8764
<i>BSL</i>	0,3689	0,9421
<i>LSL</i>	0,3472	0,9266
<i>LSDL</i>	0,2966	0,9382
<i>LRDL</i>	0,2654	0,9344
<i>LRL</i>	0,2698	0,9421

Para comparação, a arquitetura *BiGRU* treinada em Alsarhan et al. (2019), obteve acurácia de 93,79% para o conjunto de atividades diárias da base de dados *UniMiB SHAR*. Em Alawneh et al. (2020), obteve-se 90,25% de acurácia para a arquitetura *LSTM* e 91,2% para a arquitetura *BLSTM*. O modelo com um conjunto de redes *CNN* proposto em Mukherjee et al. (2020) obteve 98,7% de acurácia.



Observa-se que, de forma geral, as redes *BLSTM* e *GRU* obtiveram desempenho superior às arquiteturas *LSTM*, os algoritmos de otimização *RMSProp* e *Adam* se destacam em relação à *SGDM*, e o método de melhoria de generalização *Dropout* isolado se mostra mais eficiente do que sem nenhum método de generalização ou combinado com a regularização *L2*.

## 6

---

# CONCLUSÕES

Neste trabalho foi avaliado o desempenho de arquiteturas de redes neurais recorrentes a fim de reconhecer atividades humanas, com base em dados de acelerômetros de um dispositivo móvel. Com isso, foi possível comparar diferentes combinações de modelos de arquiteturas de rede com algoritmos de otimização e técnicas de regularização.

A partir dos experimentos, foi possível concluir que as arquiteturas *BLSTM* e *GRU*, combinadas com os algoritmos de otimização *RMSProp* e *Adam*, e a técnica de regularização *Dropout*, se mostram mais adequadas para o RAH dos dados presentes na base *UniMiB SHAR*. Ainda, observou-se que a arquitetura *BLSTM* obteve melhor desempenho que sua versão unidirecional, *LSTM*.

Este trabalho demonstrou que, embora não haja um modelo ideal ou perfeito para uma base de dados, especialmente com um conjunto extenso de dados sujeitos a diversos tipos de ruídos, é possível investigar modelos mais apropriados à base de dados, sendo assim mais propensos a serem escolhidos para o treinamento.

Como proposta para trabalhos futuros, propõe-se analisar outros indicadores de desempenho, além da perda e acurácia, comparar combinações de modelos envolvendo a arquitetura *GRU* com sua versão bidirecional, *BiGRU*, bem como validar o modelo mais promissor em um dispositivo móvel.

---

## REFERÊNCIAS

- ACADEMY, D. S. *Deep Learning Book*. 2022. <<https://www.deeplearningbook.com.br>>. Último acesso: 2022-05-31. 6, 16
- AGGARWAL, C. C. et al. Neural networks and deep learning. *Springer*, Springer, v. 10, p. 978–3, 2018. 6, 16, 17, 18
- AL-TAEI, A. A smartphone-based model for human activity recognition. *Ibn AL-Haitham Journal For Pure and Applied Science*, v. 30, n. 3, p. 243–250, 2017. 12
- ALAWNEH, L. et al. A comparison of unidirectional and bidirectional lstm networks for human activity recognition. In: *2020 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*. [S.l.: s.n.], 2020. p. 1–6. 31, 38
- ALSARHAN, T. et al. Bidirectional gated recurrent units for human activity recognition using accelerometer data. In: *IEEE. 2019 IEEE SENSORS*. [S.l.], 2019. p. 1–4. 31, 38
- ANDREJAŠIĆ, M. Mems accelerometers. In: *University of Ljubljana. Faculty for mathematics and physics, Department of physics, Seminar*. [S.l.: s.n.], 2008. 29
- ANGUITA, D. et al. A novel procedure for training l1-l2 support vector machine classifiers. In: *SPRINGER. International Conference on Artificial Neural Networks*. [S.l.], 2013. p. 434–441. 29
- BAYAT, A.; POMPLUN, M.; TRAN, D. A. A study on human activity recognition using accelerometer data from smartphones. *Procedia Computer Science*, Elsevier, v. 34, p. 450–457, 2014. 28
- BUDUMA, N.; LOCASCIO, N. *Fundamentals of Deep Learning: Designing Next-Generation Machine Intelligence Algorithms*. O'Reilly Media, 2017. ISBN 9781491925560. Disponível em: <<https://books.google.com.br/books?id=n0IIDwAAQBAJ>>. 25, 26
- BULBUL, E.; CETIN, A.; DOGRU, I. A. Human activity recognition using smartphones. In: *IEEE. 2018 2nd international symposium on multidisciplinary studies and innovative technologies (ismsit)*. [S.l.], 2018. p. 1–6. 28
- BULLING, A.; BLANKE, U.; SCHIELE, B. A tutorial on human activity recognition using body-worn inertial sensors. *ACM Computing Surveys (CSUR)*, ACM New York, NY, USA, v. 46, n. 3, p. 1–33, 2014. 29

- CASALE, P.; PUJOL, O.; RADEVA, P. Human activity recognition from accelerometer data using a wearable device. In: SPRINGER. *Iberian conference on pattern recognition and image analysis*. [S.l.], 2011. p. 289–296. 28
- CHEN, L. et al. *Activity recognition in pervasive intelligent environments*. [S.l.]: Springer Science & Business Media, 2011. v. 4. 27
- CHEN, Z. et al. Robust human activity recognition using smartphone sensors via ct-pca and online svm. *IEEE Transactions on Industrial Informatics*, IEEE, v. 13, n. 6, p. 3070–3080, 2017. 31
- CHUNG, J. et al. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014. 21
- COSTA, B. et al. Desenvolvimento de software educacional para representação e reconhecimento de som aplicado à ausculta cardiovascular. *Revista Eletrônica TECCEN*, v. 2, p. 17, 10 2016. 6, 16
- DESHPANDE, M. *Perceptrons: The First Neural Networks*. 2020. <<https://pythonmachinelearning.pro/perceptrons-the-first-neural-networks/>>. Accessed: 2022-01-23. 6, 17
- DUCHI, J.; HAZAN, E.; SINGER, Y. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, v. 12, n. 7, 2011. 26
- FALCO, I. D.; PIETRO, G. D.; SANNINO, G. Evaluation of artificial intelligence techniques for the classification of different activities of daily living and falls. *Neural Computing and Applications*, Springer, v. 32, n. 3, p. 747–758, 2019. 11
- FOCK, K. M.; KHOO, J. Diet and exercise in management of obesity and overweight. *Journal of gastroenterology and hepatology*, Wiley Online Library, v. 28, p. 59–63, 2013. 27, 28
- GOLDBERG, Y.; HIRST, G. *Neural Network Methods in Natural Language Processing*. Morgan & Claypool Publishers, 2017. (Synthesis Lectures on Human Language Technologies). ISBN 9781681731551. Disponível em: <<https://books.google.com.br/books?id=8wklDwAAQBAJ>>. 25
- GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. *Deep Learning*. [S.l.]: MIT Press, 2016. <<http://www.deeplearningbook.org>>. 6, 18, 19, 20, 21, 23, 26
- GRAVES, A. *Supervised Sequence Labelling with Recurrent Neural Networks*. Springer Berlin Heidelberg, 2012. (Studies in Computational Intelligence). ISBN 9783642247965. Disponível em: <<https://books.google.com.br/books?id=4UauNDGQWN4C>>. 6, 18, 20, 21
- GRAVES, A.; SCHMIDHUBER, J. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural networks*, Elsevier, v. 18, n. 5-6, p. 602–610, 2005. 21
- HAGAN, M. T.; DEMUTH, H. B.; BEALE, M. *Neural Network Design*. USA: PWS Publishing Co., 2014. 22, 23

- HOCHREITER, S.; SCHMIDHUBER, J. Long short-term memory. *Neural computation*, MIT Press, v. 9, n. 8, p. 1735–1780, 1997. 20
- KANG, M.; CHOI, E. *Machine Learning: Concepts, Tools And Data Visualization*. World Scientific Publishing Company, 2021. ISBN 9789811228162. Disponível em: <<https://books.google.com.br/books?id=IjsoEAAAQBAJ>>. 6, 19
- KINGMA, D. P.; BA, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 25
- KROGH, A.; HERTZ, J. A simple weight decay can improve generalization. *Advances in neural information processing systems*, v. 4, 1991. 25
- KWON, H. J.; SEOK, S.; LIM, G. System modeling of a mems vibratory gyroscope and integration to circuit simulation. *Sensors*, Multidisciplinary Digital Publishing Institute, v. 17, n. 11, p. 2663, 2017. 29
- LEE, S.-M.; YOON, S. M.; CHO, H. Human activity recognition from accelerometer data using convolutional neural network. In: IEEE. *2017 IEEE International Conference on Big Data and Smart Computing (BigComp)*. [S.l.], 2017. p. 131–134. 28
- MICUCCI, D.; MOBILIO, M.; NAPOLETANO, P. Unimib shar: A dataset for human activity recognition using acceleration data from smartphones. *Applied Sciences*, v. 7, n. 10, 2017. ISSN 2076-3417. Disponível em: <<http://www.mdpi.com/2076-3417/7/10/1101>>. 11, 30, 32
- MOHRI, M.; ROSTAMIZADEH, A.; TALWALKAR, A. *Foundations of machine learning*. [S.l.]: MIT press, 2018. 14, 15
- MUKHERJEE, D. et al. Ensemconvnet: a deep learning approach for human activity recognition using smartphone sensors for healthcare applications. *Multimedia Tools and Applications*, Springer, v. 79, n. 41, p. 31663–31690, 2020. 38
- Ngu, A. H. et al. Personalized fall detection system. p. 1–7, 2020. 11
- ORTIZ, J. *Smartphone-Based Human Activity Recognition*. Springer International Publishing, 2015. (Springer Theses). ISBN 9783319142746. Disponível em: <<https://books.google.com.br/books?id=Uq0qBgAAQBAJ>>. 27, 28, 29
- PAULOVICH, F. V.; OLIVEIRA, M. C. F. D.; JR, O. N. O. A future with ubiquitous sensing and intelligent systems. *ACS sensors*, ACS Publications, v. 3, n. 8, p. 1433–1438, 2018. 27
- PENDÃO, C. G.; MOREIRA, A. C.; RODRIGUES, H. Energy consumption in personal mobile devices sensing applications. In: IEEE. *2014 7th IFIP wireless and mobile networking conference (WMNC)*. [S.l.], 2014. p. 1–8. 28
- Perez, A. J.; Labrador, M. A.; Barbeau, S. J. G-sense: a scalable architecture for global sensing and monitoring. *IEEE Network*, v. 24, n. 4, p. 57–64, 2010. 11
- PRAET, S. F.; LOON, L. J. van. Exercise therapy in type 2 diabetes. *Acta diabetologica*, Springer, v. 46, n. 4, p. 263–278, 2009. 27, 28

- REYES-ORTIZ, J.-L. et al. Human activity recognition using smartphones data set. *UCI Machine Learning Repository; University of California, Irvine, School of Information and Computer Sciences: Irvine, CA, USA*, 2012. 29
- REYES-ORTIZ, J. L. et al. Human activity and motion disorder recognition: towards smarter interactive cognitive environments. In: CITESEER. *ESANN*. [S.l.], 2013. 29
- REYES-ORTIZ, J.-L. et al. Human activity recognition on smartphones with awareness of basic activities and postural transitions. In: SPRINGER. *International conference on artificial neural networks*. [S.l.], 2014. p. 177–184. 29
- RONAO, C. A.; CHO, S.-B. Human activity recognition with smartphone sensors using deep learning neural networks. *Expert systems with applications*, Elsevier, v. 59, p. 235–244, 2016. 28
- ROSENBLATT, F. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, v. 65 6, p. 386–408, 1958. 16
- RUMELHART, D. E.; HINTON, G. E.; WILLIAMS, R. J. Learning representations by back-propagating errors. *nature*, Nature Publishing Group, v. 323, n. 6088, p. 533–536, 1986. 18
- SAZONOV, E. *Wearable Sensors: Fundamentals, Implementation and Applications*. Elsevier Science, 2014. ISBN 9780124186668. Disponível em: <<https://books.google.com.br/books?id=wGJzAAwAAQBAJ>>. 28
- SCHUSTER, M.; PALIWAL, K. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, v. 45, n. 11, p. 2673–2681, 1997. 20
- SERRÃO, M. et al. Human activity recognition from accelerometer with convolutional and recurrent neural networks. *Polytechnica*, Springer, v. 4, n. 1, p. 15–25, 2021. 31
- SRIVASTAVA, N. et al. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, JMLR. org, v. 15, n. 1, p. 1929–1958, 2014. 6, 24
- SUBASI, A. et al. Iot based mobile healthcare system for human activity recognition. In: IEEE. *2018 15th Learning and Technology Conference (L&T)*. [S.l.], 2018. p. 29–34. 11
- SUTSKEVER, I. et al. On the importance of initialization and momentum in deep learning. In: PMLR. *International conference on machine learning*. [S.l.], 2013. p. 1139–1147. 25
- VARSHNEY, U. Mobile health: Four emerging themes of research. *Decision Support Systems*, Elsevier, v. 66, p. 20–35, 2014. 11
- WANG, C.; VENKATESH, S.; JUDD, J. Optimal stopping and effective machine complexity in learning. In: COWAN, J.; TESAURO, G.; ALSPECTOR, J. (Ed.). *Advances in Neural Information Processing Systems*. Morgan-Kaufmann, 1993. v. 6. Disponível em: <<https://proceedings.neurips.cc/paper/1993/file/43fa7f58b7eac7ac872209342e62e8f1-Paper.pdf>>. 23
- WANG, J. et al. Deep learning for sensor-based activity recognition: A survey. *Pattern Recognition Letters*, Elsevier, v. 119, p. 3–11, 2019. 11

---

YIN, J.; YANG, Q.; PAN, J. J. Sensor-based abnormal human-activity detection. *IEEE Transactions on Knowledge and Data Engineering*, IEEE, v. 20, n. 8, p. 1082–1090, 2008. 11