Universidade Federal de Campina Grande - UFCG

Centro de Engenharia Elétrica e Informática - CEEI

Programa de Pós-Graduação em Engenharia Elétrica - PPgEE

Bruna Salles Moreira

# Análise Comparativa entre Redes Neurais Artificiais e Modelos Escondidos de Markov para Aplicações de Reconhecimento de Gestos em Superfícies

Campina Grande - PB

2020

Bruna Salles Moreira

# Análise Comparativa entre Redes Neurais Artificiais e Modelos Escondidos de Markov para Aplicações de Reconhecimento de Gestos em Superfícies

Dissertação de mestrado apresentada à Coordenação do Programa de Pós Graduação em Engenharia Elétrica da Universidade Federal de Campina Grande - Campus de Campina Grande como parte dos requisitos necessários para a obtenção do grau de Mestre em Engenharia Elétrica.

Área de Concentração: Processamento da Informação

Universidade Federal de Campina Grande - UFCG

Centro de Engenharia Elétrica e Informática - CEEI

Programa de Pós-Graduação em Engenharia Elétrica - PPgEE

Orientador:
Angelo Perkusich
Saulo Oliveira Dornellas Luiz

Campina Grande - PB
2020

M838a Moreira, Bruna Salles.

Análise comparativa entre redes neurais artificiais e modelos escondidos de Markov para aplicações de reconhecimento de gestos em superfícies / Bruna Salles Moreira. - Campina Grande, 2020.

63 f. : il. color

Dissertação (Mestrado em Engenharia Elétrica) - Universidade ´Federal de Campina Grande, Centro de Engenharia Elétrica e Informática, 2019.

"Orientação: Prof. Dr. Angelo Perkusich, Prof. Dr. Saulo Oliveira Dornellas Luiz.

Referências.

1. Reconhecimento de Gestos. 2. Entrada Acústica. 3. Rede Neural Artificial. 4. Modelos Escondidos de Markov. I. Perkusich, Angelo. II. Luiz, Saulo Oliveira Dornellas. III. Título.

CDU 621.3:004.8(043)

FICHA CATALOGRÁFICA ELABORADA PELO BIBLIOTECÁRIO GUSTAVO DINIZ DO NASCIMENTO CRB-15/515

‘‘Ninguém é tão grande que não possa aprender,
nem tão pequeno que não possa ensinar.’’

Esopo

# Agradecimentos

À Deus, dedico o meu agradecimento maior, pela dádiva da vida e por iluminar minha caminhada.

Agradeço a minha mãe, Lívia Salles, por sempre ter sido meu porto seguro nas tempestades mais severas, minha eterna companheira e melhor amiga. Ao meu pai, André Moreira, pelo amor e incentivo.

A minha avó Lêda Salles, na qual espelhei-me em sua espiritualidade e sabedoria.

Aos professores Angelo Perkusich e Saulo Oliveira pelos ensinamentos, pelos conselhos e pela orientação, sem a qual este trabalho não seria realizado.

A José Antônio Neto, amigo e companheiro, agradeço por toda a compreensão, incentivo e carinho.

Aos amigos Alequine, Charles, Marcus, Paulo e Clara Liz que fizeram esta jornada muita mais alegre e divertida.

E a todos os meus familiares, mestres e demais amigos que participaram dessa trajetória, colaborando para concretização desta etapa em minha vida.

Por fim, agradeço ao CNPq pelo suporte financeiro durante a execução desse mestrado.

# Resumo

Muitas atividades humanas são táteis. Reconhecer como uma pessoa toca um objeto ou uma superfície que os cerca diariamente é uma área ativa de pesquisa e gera um forte interesse na comunidade de superfícies interativas. Nesta dissertação, compara-se duas técnicas de aprendizado de máquina, a Rede Neural Artificial (RNA) e os Modelos de Markov escondidos (HMM), pois são técnicas comuns e com baixo custo computacional utilizadas para classificar uma entrada acústica, baseando-se em o som único produzido quando uma unha é arrastada sobre uma superfície. Empregou-se um microfone pequeno e de baixo custo que pode ser facilmente incorporado a uma superfície para ser utilizado como entrada passiva de reconhecimento de gestos. Nossa contribuição é analisar as vantagens e limitações dessas técnicas no contexto do reconhecimento de gestos usando um alfabeto simples de três figuras geométricas: círculo, quadrado e triângulo. Para isso, usamos as *toolboxes* do Matlab para implementar os modelos e avaliar o conjunto de dados utilizados para treinar os modelos.

**Palavras-chave:** reconhecimento de gestos, entrada acústica, Rede Neural Artificial, Modelos escondidos de Markov.

# Abstract

Many human activities are tactile. Recognizing how a person touches an object or a surface that surrounds them daily is an active area of research and has generated a strong interest within the interactive surfaces community. In this thesis, we compare two machine learning techniques, namely Artificial Neural Network (ANN) and Hidden Markov Models (HMM), as they are some of the most common techniques with low computational cost used to classify an acoustic-based input that relies on the unique sound produced when a fingernail is dragged over a surface. We employ a small and low cost microphone that could be easily incorporated into a surface on which it rests to be applied as a passive gesture recognition input. Our contribution is to analyze the advantages and limitations of these techniques in the context of gesture recognition using a simple alphabet of three geometrical figures: circle, square and triangle. To do so, we use Matlab's toolboxes to implement the models and evaluate the dataset used to train the ANN and the HMM.

**Keywords:** Gesture recognition, Acoustic-based input, Artificial Neural Network, Hidden Markov Models.

# Lista de ilustrações

# Lista de tabelas

# Lista de abreviaturas e siglas

| | |
|---|---|
| ANN | *Artificial Neural Network* |
| ASD | *Amplitude Spectrum Density* |
| CNN | *Convolutional Neural Network* |
| DFT | *Discrete Fourier Transform* |
| ELM | *Extreme Learning Machine* |
| EMQ | Erro Médio Quadrático |
| FFT | *Fast Fourier Transform* |
| GAD | Grafo Acíclico Direcionado |
| HMM | *Hidden Markov Models* |
| IAT | Interfaces Acústicas Tangíveis |
| MFCC | *Mel Frequency Cepstral Coefcients* |
| MLP | *Multi-Layer-Perceptron* |
| RNA | Redes Neurais Artificiais |
| STFT | *Short-term Fourier Transform* |
| SVM | *Support Vector Machine* |
| TDOA | *Time Difference Of Arrival* |
| UART | *Universal Asynchrounous Receiver/Transmiter* |
| UFCG | Universidade Federal de Campina Grande |
| VANT | Veículo aéreo não tripulado |

# Sumário

# 1 Introdução

Muitas atividades no dia-a-dia são uma combinação de sensações táteis e auditivas. Nós tocamos, ouvimos e interagimos com uma infinidade de objetos todos os dias. Alguns desses objetos estão registrando nossas atividades, como as telas sensíveis ao toque do *smartphone* ou do *tablet*. Dessa forma, adicionar percepção a objetos arbitrários e superfícies que estão a nossa volta é uma área ativa de pesquisa e tem gerado um forte interesse na comunidade de superfícies interativas [43].

Ao conectar um microfone a uma superfície, um sinal é capturado pelo sensor acústico e é possível analisar o sinal adquirido para identificar as informações que podem ser usadas no desenvolvimento de interfaces acústicas tangíveis [11]. Essas interfaces funcionam com base no princípio de que, quando um objeto é tocado e manipulado, suas propriedades são alteradas. Em particular, o modo como ressoa (ou vibra) quando é tocado varia dependendo de como e onde é tocado. As vibrações produzidas podem ser adquiridas e analisadas, de forma a inferir sobre como a interação com o objeto está sendo realizada [61].

Da utilização de sensores acústicos para obter essas informações da interação do usuário com o objeto ou superfície surge o conceito de *acoustic sensing* (em português, detecção acústica).

Assim, a técnica de detecção acústica tem sido explorada como uma ferramenta para recuperar informações da interação de um usuário com uma superfície ou objeto [61]. Algumas dessas informações podem ser usadas pelos desenvolvedores para adicionar novos recursos às interfaces ou melhorar a interação nas interfaces atuais, podendo fornecer mais expressão aos gestos definidos pelo usuário e expandir a linguagem de entrada da interação.

Em contraste com sistemas de reconhecimento de gestos baseados em câmera, todos os elementos sensores da técnica de detecção acústica podem ser integrados à superfície, e este método não sofre com problemas de iluminação e oclusão [56].

Pesquisadores exploraram diversas técnicas de entrada baseadas em acústica. Essas técnicas são classificadas em abordagens passivas e ativas. As abordagens passivas detectam a entrada de um usuário, capturando e analisando sons gerados pelas ações do mesmo, sem a necessidade de um componente ativo, dependendo apenas das propriedades do objeto utilizado (sua estrutura, material ou propriedades exclusivas). Já nas abordagens ativas, um atuador deve ser usado para interagir com uma superfície, como um alto falante. A seguir, alguns trabalhos que utilizam essas técnicas são brevemente apresentados e, no Capítulo 2, são discorridos mais detalhes sobre essas abordagens.

Harrison *et al.* [23] utilizam o método classificador árvores de decisões, uma abordagem mais simples, que se torna insuficiente com entradas mais complexas, como símbolos mais detalhados ou até mesmo para diferenciar entre as letras parecidas como “P” e “B”.

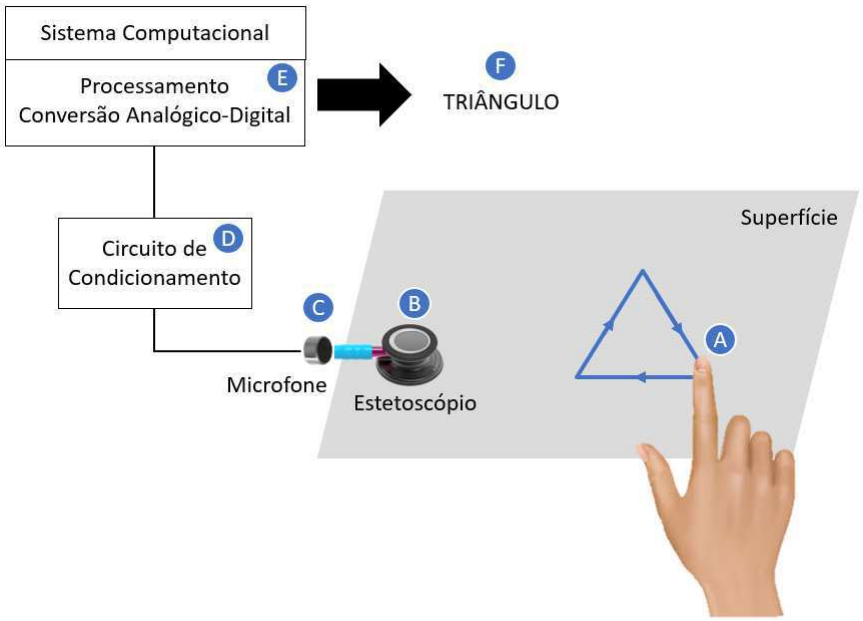
*Acoustic Barcodes* [26] são códigos de barras táteis, que representam um ID binário do som produzido, quando um objeto como uma unha atravessa os entalhes. Para o reconhecimento *Acoustic Barcodes* [26] realiza uma simples detecção de pico.

*Hambone* [24] é um sensor acústico usado no punho que detecta o movimento dos braços por condução óssea e realiza o reconhecimento de gestos por meio do classificador Modelos escondidos de Markov para classificar os vários sons produzidos. Da mesma forma, *Skinput* [25] usa uma braçadeira de detecção acústica para localizar os dedos na pele, por meio do classificador *Support Vector Machine* (SVM). Já *TapSense* [24] usa um microfone conectado a uma superfície interativa (por exemplo, tela sensível ao toque) para diferenciar entre toques de objetos diferentes e partes do dedo, e utiliza, assim como *Skinput* [25] o classificador SVM.

Na Figura 1, apresenta-se a ilustração do funcionamento do sistema proposto. Quando uma unha (A) arranha uma superfície, resulta em uma série de vibrações mecânicas, que se propagam pela superfície e são capturadas pelo microfone (C) que está conectado à cânula do estetoscópio (B). O uso de estetoscópio para aquisição de sons fornece naturalmente um alto nível de supressão de ruído ambiental. Isso permite que os impactos sejam prontamente segmentados de qualquer ruído de fundo com um limiar de amplitude simples [7]. O sinal do microfone é então filtrado e amplificado por um circuito de condicionamento (D) e adquirido por um conversor analógico-digital de 10 bits de um microcontrolador. Em seguida, o microcontrolador envia o sinal digital via comunicação serial a uma taxa de transmissão de 115200 bits/s para o computador (E), que finalmente será capaz de reconhecer o gesto realizado (F) na superfície por um dos aplicativos de aprendizado de máquina desenvolvidos no Matlab.

Afirma-se a necessidade de uma superfície, visto que o som se propaga por meio de materiais sólidos e líquidos com muito mais eficiência do que pelo ar. Assim, enquanto o atrito da unha em uma superfície produz apenas um ruído audível e suave, o sinal é propagado consideravelmente melhor através do material sólido. Esta propagação superior de som significa que um sinal não só é propagado por uma distância maior, como também é melhor preservado, isto é, menos ruidoso. [23].

Figura 1 – Representação do funcionamento do sistema proposto.




Fonte: Elaborado pela autora.

Após análise da literatura referente a detecção acústica, percebe-se que vários trabalhos se concentram no reconhecimento de gestos usando técnicas de aprendizado de máquina, como redes neurais artificiais e os modelos escondidos de Markov que são soluções de baixo custo computacional para essa problemática. Desta forma, propõe-se realizar uma análise comparativa entre esses dois tipos de algoritmos de aprendizagem supervisionada, no contexto de detecção acústica, mais especificamente em abordagens passivas.

Os modelos de Markov escondidos são bastante utilizados em sistemas de reconhecimento de fala ou reconhecimentos por meio de sinais acústicos pois têm um algoritmo eficiente e robusto para o treinamento e reconhecimento. Como o sinal acústico tem estrutura temporal e pode ser codificada como uma sequência de vetores espectrais que abrangem a faixa de frequência de áudio, o modelo escondido de Markov fornece uma estrutura natural para a construção de tais modelos [5].

O treinamento do modelo consiste em modelar o conjunto dos parâmetros acústicos extraídos do sinal por uma sequência de estados (cadeia de Markov de primeira ordem) de acordo com a variação temporal do sinal. Já no reconhecimento, a sequência de observações de teste é aceita como verdadeira se possuir uma medida de similaridade (verossimilhança) acima de um limiar estipulado com os parâmetros do modelo [57].

Já as Redes Neurais Artificias (RNA) são sistemas computacionais de implementação em *software* ou em *hardware*, que imitam as habilidades "computacionais" do sistema nervoso biológico, utilizando um grande número de neurônios artificiais interconectados

[41]. Os principais aspectos estruturais de redes neurais são adaptação e aprendizagem, possibilitando lidar com dados imprecisos e situações não totalmente definidas. Uma rede treinada tem a habilidade de generalizar quando é apresentada a entradas que não estão presentes em dados já conhecidos por ela [33].

## 1.1 Objetivos

O objetivo neste trabalho é realizar uma análise comparativa entre os dois tipos de classificadores: Modelos escondidos de Markov e Redes Neurais Artificiais, no contexto do problema de reconhecimento de padrões referentes às assinaturas acústicas de diferentes gestos realizados em uma superfície. Deseja-se alcançar uma alta taxa de sucesso de reconhecimento com um conjunto de dados pequeno (máximo de 10 desenhos por gesto) e um tempo de treinamento curto (máximo de 2 minutos por gesto), dispondo apenas de um *hardware* simples e baixo custo computacional para o algoritmo de aprendizado de máquina.

### 1.1.1 Objetivos Específicos

Além do objetivo principal, os seguintes objetivos específicos foram definidos:

1. Projetar o *hardware* da plataforma experimental (microfone, microcontrolador, circuito de condicionamento);
2. Pesquisar, avaliar e definir se a análise dos dados será realizada no domínio do tempo ou no domínio da frequência (FFT);
3. Desenvolver a aplicação utilizando o algoritmo de aprendizagem supervisionada Redes Neurais Artificiais;
4. Desenvolver a aplicação utilizando o algoritmo de aprendizagem Modelos escondidos de Markov;
5. Analisar e comparar os resultados dos passos 4 e 5.

## 1.2 Organização do Texto

O presente trabalho está organizado da seguinte forma:

- **Capítulo 2 - Revisão da Literatura**: o capítulo apresenta uma revisão da literatura, focando em trabalhos com aplicações na área de detecção acústica, no *hardware* utilizado, as técnicas de processamento do sinal e os classificadores para realizar o reconhecimento acústico.

- **Capítulo 3 - Fundamentação Teórica**: conceitos das técnicas de aprendizado de máquina selecionadas: Redes Neurais Artificiais e Modelos escondidos de Markov são apresentados.
- **Capítulo 4 - Plataforma e Resultados Experimentais**: a plataforma experimental desenvolvida para execução dos experimentos é apresentada detalhadamente e os resultados experimentais com análise dos conjunto de dados e taxas de sucesso para cada classificador e algumas questões de discussões e limitações do sistema.
- **Capítulo 5 - Considerações Finais**: Conclusões gerais e possíveis trabalhos futuros são discutidos neste Capítulo.

# 2 Revisão da Literatura

Para a realização desta revisão da literatura foi desenvolvido um protocolo que estabelece critérios, estratégias e métodos de busca. Desta forma, como ponto principal de uma revisão, é necessário estabelecer uma problemática para respaldar toda a pesquisa e as possíveis questões de pesquisa.

## 2.1 Problemática

O trabalho proposto visa realizar uma análise comparativa entre os classificadores Modelos escondidos de Markov e Redes Neurais Artificiais, no contexto de reconhecimento de padrões referentes às assinaturas acústicas de diferentes gestos realizados em uma superfície. Assim, essa revisão da literatura foi desenvolvida com a finalidade de validar a inovação do trabalho proposto, visto que, tem como objetivo pesquisar na literatura trabalhos que implementem as técnicas computacionais, como também analisar outras tecnologias de entrada baseadas em detecção acústica.

## 2.2 Trabalhos selecionados na Revisão da Literatura

Nesta seção são analisados artigos científicos que utilizam a técnica de detecção acústica, com foco no *hardware* utilizado, nas características do sinal de entrada e em seu processamento, e nas principais técnicas utilizadas para o reconhecimento dos gestos em superfícies.

A detecção acústica tem sido objeto de intensa pesquisa nas últimas décadas. Vários algoritmos para o processamento de informações acústicas foram explorados. Estes incluem Máquina de Vetores de Suporte, Redes Neurais, Árvores de Decisão, Redes escondidas de Markov e Análise Bayesiana. Esses esforços levaram ao desenvolvimento de sistemas que foram usados em uma variedade de tarefas de reconhecimento acústico, incluindo reconhecimento de gestos, texturas, localização espacial do toque em uma superfície e classificação de objetos.

Dependendo da abordagem usada para implementar a detecção acústica e da finalidade do trabalho, dividiu-se os artigos científicos nos seguintes grupos:

- Abordagens Passivas
  - *Smartwatches*;
  - Corpo humano.

- Abordagens Ativas
  - Com utilização de objetos;
- Identificação de objetos e texturas;
- Localização espacial da entrada em uma superfície.

### 2.2.1 Abordagens Passivas

Pesquisadores têm explorado muitas técnicas de entrada baseadas em acústica. Essas técnicas são classificadas em abordagens passivas e ativas. Nas abordagens passivas detecta-se a entrada de um usuário, adquirindo e analisando sons gerados pelas ações do mesmo. Uma abordagem passiva permite uma interação muito mais conveniente e simples com as superfícies, pois não precisa de um componente ativo para funcionar corretamente. Em vez disso, ela depende das propriedades do objeto usado (sua estrutura, material ou propriedades exclusivas).

Harrison e Hudson [23] desenvolveram *Scratch Input*, uma técnica de entrada acústica baseada no som produzido quando uma unha é arrastada sobre a superfície de um material texturizado. Eles utilizaram um microfone genérico agregado a um estetoscópio modificado, particularmente adequado para amplificar o som e detectar ruídos de alta frequência. O microfone converte o som em sinal elétrico, o qual é amplificado e conectado a um computador através do conector de entrada de áudio. Esse *hardware* pode ser facilmente acoplado a superfícies existentes, como paredes e mesas, transformando-as em superfícies de entrada grandes, sem alimentação e *ad hoc*.

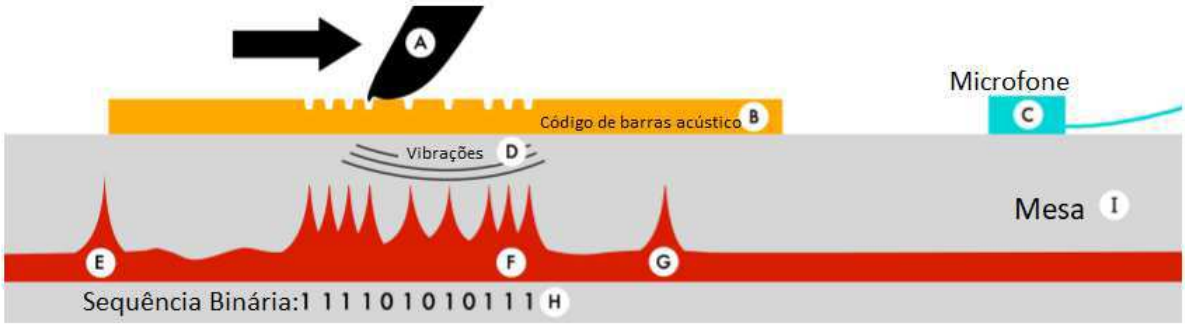
O reconhecedor de padrões acústicos de *Scratch Input* [23] usa uma árvore de decisão superficial baseada principalmente na contagem de pico e variação de amplitude. No entanto, um mecanismo de reconhecimento mais sofisticado poderia incorporar outras dimensões, como frequência e duração, e provavelmente ser capaz de suportar consideravelmente mais gestos e com maior precisão.

Harrison *et al.* [26] criaram *Acoustic Barcodes*, códigos de barras táteis que representam um ID binário do som produzido, quando um objeto como uma unha atravessa os entalhes. O sinal de áudio é amostrado, filtrado, refinado e finalmente alimentado para uma rotina de detecção de pico.

Na Figura 2, tem-se um exemplo de um código de barras acústico (B) situado em uma mesa (I), à qual um microfone (C) está conectado. Quando uma unha (A) passa pelo código de barras, resulta em uma série de vibrações mecânicas (D), que se propagam pela mesa e são capturadas pelo microfone. O primeiro som é o impacto inicial do dedo (E). À medida que a unha passa sobre os entalhes do código de barras acústico, uma série de picos de som são produzidas (F). Finalmente, o dedo se solta ou cai no final do código

de barras (G). A forma de onda resultante é processada, resultando em uma sequência binária decodificada (H).

Figura 2 – Representação do funcionamento de *Acoustic Barcodes* [26].




Fonte:[26] (Modificada).

Harrison *et al.* [24] apresentaram *TapSense*, uma abordagem que permite que superfícies convencionais identifiquem o tipo de objeto sendo usado para entrada. Isso é conseguido segmentando e classificando os sons resultantes do impacto de um objeto. A abordagem é composta por dois processos-chave operando em conjunto. O primeiro é um método para detectar e rastrear a posição da entrada, seja de um ou de vários toques. O segundo componente ouve, segmenta e classifica os impactos na superfície interativa usando recursos acústicos. Ele se baseia no princípio físico de que diferentes materiais produzem diferentes assinaturas acústicas e têm diferentes frequências ressonantes (Figura 3). A configuração do *hardware* é semelhante a *Scratch Input* [23], embora com um objetivo diferente (reconhecimento de gestos em superfícies *versus* reconhecimento de ponta em telas sensíveis ao toque). O reconhecedor de padrões acústicos utilizou uma implementação de SVM fornecida pelo kit de ferramentas *Weka Machine Learning*.

Figura 3 – Espectrograma dos quatro tipos de entrada de *Tapsense* [24].


Fonte:[24] (Modificada).

Lopes *et al.* [43] propõem um reconhecimento de toque sonicamente aprimorado. Assim, toques que possuem assinaturas de som idênticas, podem, no entanto, ser distinguidos por suas características acústicas: intensidade e timbre. Para capturar vibrações causadas pela interação dos usuários, utilizou-se um microfone de contato colocado na

superfície. Assim, os usuários podem expressar diferentes ações tocando a superfície com partes diferentes do corpo, como dedos, unhas, punho, entre outros – nem sempre distinguíveis pelas tecnologias de toque, mas reconhecidas pelo sensor acústico. A contribuição é a integração do toque e do som para expandir a linguagem de entrada da interação superficial.

O módulo de reconhecimento de gestos é implementado no *PureData*. O fluxo de trabalho do processamento de áudio é a captura do sinal, o qual é amostrado, filtrado, analisado matematicamente por meio de uma Transformada Discreta de Fourier (do inglês, *Discrete Fourier transform* - DFT) e finalmente, a assinatura espectral é comparada com um banco de dados de gestos treinados usando SVM. Se uma correspondência for encontrada, será emitido um evento com o tipo de gesto, intensidade (dBs) e carimbo de data e hora [43].

O *Rubbinput* apresentado por Kawakatsu e Hirai [34] é uma interface de usuários para ambientes úmidos que utiliza os sons feitos ao esfregar uma superfície lisa e úmida com os dedos. O *Rubbinput* detecta apenas sons, rastreando estruturas harmônicas contínuas em séries temporais usando técnicas de processamento de sinais, interpretando "duração do tempo", "movimentos alternativos" e "número de dedos friccionados" e convertendo-os em eventos da interface do usuário.

Tomoyuki *et al.* [62] apresentam o *RapTapBath*, um sistema de interface de usuário que utiliza uma banheira como entrada para um controlador que reconhece padrões e tons tocados pela mão na borda de uma banheira. Este sistema utiliza sensores piezoelétricos embutidos na borda da banheira para analisar sinais acústicos de sons tocados e projeta um menu na borda da cuba usando um projetor instalado acima da mesma. Os locais das tomadas são detectados por diferenças de medição dos tempos de propagação do sinal usando os sensores piezoelétricos. Tons de toque são identificados por padrões de espectro usando fatoração de matriz não negativa.

Antonacci *et al.* [2] apresentaram um método que permite avaliar o objeto arranhando a superfície sólida, a partir de um sinal vibracional. A técnica pode ser utilizada de forma eficaz no contexto das Interfaces Acústicas Tangíveis (IATs), baseando-se simplesmente em sinais acústicos. Não foram consideradas soluções mais sofisticadas, devido ao seu custo computacional. Validou-se a técnica apresentada nesse trabalho com resultados experimentais, demonstrando que diferentes classes de objetos podem ser efetivamente distinguidas na maioria das vezes, mesmo em um ambiente ruidoso.

Braun *et al.* [7] apresentaram um método que usa sinais acústicos, captados por um ou mais microfones de contato anexados a uma superfície, e uma abordagem de aprendizado de máquina SVM, permitindo detectar múltiplas formas de interação com essa superfície, sendo capaz de distinguir vários eventos de impacto (como tocar e bater), bem como arrastar do dedo ou da mão. Braun *et al.* [7] analisam os efeitos da inclusão de

vários microfones na configuração de detecção e fornece uma extensão do uso da localização fornecida por um sistema de sensor capacitivo para melhorar a precisão de classificação.

Mingshi et al. [10] exploram a possibilidade de estender a entrada e as interações além da tela do dispositivo móvel para superfícies adjacentes *ad hoc*. O sistema proposto chama-se *Ipanel* e emprega os sinais acústicos gerados pelo deslizamento dos dedos na mesa para reconhecimento de gestos. Recursos exclusivos são extraídos explorando as propriedades espaço-temporais e de domínio de frequência dos sinais acústicos gerados. Os recursos são transformados em imagens e, em seguida, redes neurais convolucionais (CNN) foram empregadas para reconhecer o movimento dos dedos na mesa. O *Ipanel* pode suportar não apenas reconhecimento de gestos comumente usados (clique, virar, rolagem, zoom, etc.), mas também reconhecimento de manuscrito (10 números e 26 alfabetos) com alta precisão.

Além disso, o desempenho da *Ipanel* é robusto contra diferentes níveis de ruído ambiente e diferentes materiais de superfície. Embora quando os dedos estejam a 20 cm do móvel, a precisão do reconhecimento diminua significativamente porque os sinais se tornam fracos demais para serem usados para reconhecimento e a CNN de 5 camadas adotada possui uma alta carga computacional.

#### 2.2.1.1 Corpo Humano

Harrison *et al.* [25] desenvolveram o *Skinput*, uma tecnologia que se apropria do corpo humano para a transmissão acústica, permitindo que a pele seja usada como uma superfície de entrada, em particular, pelos dedos, no braço e na mão. Nesse trabalho, são analisadas as vibrações mecânicas que se propagam pelo corpo. Coleta-se esses sinais usando um novo conjunto de sensores usados como uma braçadeira, apresentada na Figura 4. Essa abordagem fornece um sistema de entrada digital sempre disponível, naturalmente portátil e no corpo. *Skinput* é o projeto de um novo sensor vestível para aquisição de sinais bioacústicos que apresenta um sistema capaz de resolver a localização dos toques dos dedos no corpo.

*Skinput* [25] utilizou um conjunto de sensores de vibração altamente sintonizados, pequenos filmes piezoelétricos em cantilever (*MiniSense100*, *Measurement Specialties, Inc.*). SVM fornecido no kit de ferramentas de aprendizado de máquina *Weka* foi a abordagem escolhida dos autores para realizar a classificação.

Deyle *et al.* [16] implementaram *Hambone*, um sistema leve e discreto que oferece acesso rápido e capacidade de multitarefa para a interação de dispositivos móveis. *Hambone* usa dois pequenos sensores piezoelétricos colocados no pulso ou no tornozelo. Quando um usuário move suas mãos ou pés, os sons gerados pelo movimento viajam para o dispositivo por condução óssea. Este transmite, então, os sinais digitalmente para um dispositivo móvel ou computador. Os sinais são reconhecidos usando HMMs e são mapeados para um

Figura 4 – Fotografia da pulseira protótipo de *Skinput* [25].

Fonte:[25].

conjunto de comandos que controlam uma aplicação.

Amento *et al.* [1] apresentam *The Sound of One Hand*. O usuário realiza movimentos simples com os dedos (por exemplo, tocar, friccionar, sacudir, estalar), e o som interno nas pontas dos dedos é conduzido pelos ossos até a pele, na qual um pequeno microfone piezoelétrico montado em uma pulseira, apresentada na Figura 5, pode captar esses sons de modo confiável.

Amento *et al.* [1] desenvolvem no Matlab, dois classificadores. O primeiro classificador reproduz o áudio do microfone a uma taxa de 8000 amostras por segundo. As tensões são quantificadas em 10 níveis distintos com base no máximo em cada período de amostragem. Esta tensão quantizada é usada como entrada para uma máquina de estados finitos para determinar qual gesto foi executado. O segundo classificador utiliza os HMMs. Esse classificador é treinado com múltiplas repetições de cada sinal até que um modelo de cada gesto seja obtido.

Figura 5 – Fotografia da pulseira desenvolvida em *The Sound of One Hand* [1].

Fonte:[1].

#### 2.2.1.2 Aplicação *Smartwatch*

Chen *et al.* [9] desenvolveram *WritePad*, o qual apresenta um sensor acústico passivo, em que os *smartwatches* testam o som ambiente durante a gravação. Primeiro, emprega-se a transformação *wavelet* para eliminar a interferência do ruído ao redor. Em seguida, projeta-se um modelo de rede neural convolucional híbrida para reconhecimento

de número, em que três camadas de convolução são seguidas por três camadas de *pool* máximo. A precisão do reconhecimento de números pode ser superior a 95% quando se adota dados únicos e é de cerca de 91% quando 10 pessoas são incorporadas.

O trabalho inicial de Zhang [71] apresentou o reconhecimento de gestos baseado em sinal acústico utilizando apenas o ASD (*Amplitude Spectrum Density*) para obter um bom desempenho. Posteriormente, o trabalho *Sound Write II* de Luo *et al.* [44] melhorou o trabalho de Zhang *et al.*[71], utilizando um recurso mais sólido para obter melhor precisão no reconhecimento de gestos: MFCC.

### 2.2.2 Abordagens Ativas

Nas abordagens ativas, um objeto explícito (por exemplo, uma caneta especial ou um alto-falante) deve ser usado para interagir com uma superfície, o que geralmente requer que alguns componentes eletrônicos sejam usados. Alternativas para este método envolvem instrumentar a entrada de fornecimento de objeto (marcadores fiduciários, alto-falantes, sensores de pulso, entre outros).

Murray-Smith *et al.* [47] desenvolveram *Stane*, um pequeno dispositivo (Figura 6) com um microfone interno e uma infinidade de texturas projetadas em sua superfície; o dispositivo classifica os sons produzidos ao esfregar diferentes áreas. As vibrações detectadas são classificadas em tempo real, com sinais de esfregar diferentes áreas do dispositivo atribuídas a classes distintas. *Stane* utiliza um processo de classificação de dois estágios, com classificação instantânea de baixo nível e classificadores de nível mais alto que agregam as evidências do primeiro estágio ao longo do tempo. Para a classificação instantânea, o áudio de entrada é classificado por *Multi-Layer-Perceptron* (MLP). Quatro classes diferentes são treinadas; estes são: arranhar a frente circular no sentido horário, contornar arranhões no lado direito, riscar com ponta e uma classe miscelânea.

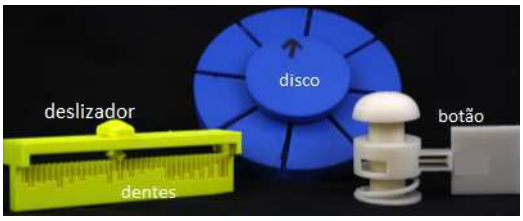
Figura 6 – Fotografia de dedo arranhando uma superfície no objeto com microfone acoplado.

Fonte:[47].

Savage *et al.* [59] apresenta *Lamello*, uma abordagem para criar componentes de entrada tangíveis que reconhecem a interação do usuário por meio de acústica. *Lamello*

[59] emprega estruturas do tipo *fscom*: caminhos de comprimento variável em pontos de interação. Um *pipeline* de processamento de áudio em tempo real analisa os sons resultantes do mover nos dentes dos objetos e emite eventos de interação de alto nível. As principais contribuições estão no *co-design* das estruturas dos objetos, nos esquemas de codificação de informações e na análise de áudio. Demonstram-se botões, *sliders* e mostradores acionados em *Lamello* impressos em 3D, apresendados na Figura 7.

Figura 7 – Representação dos botões, deslizadores e discos acionados em *Lamello* impressos em 3D.




Fonte:[59] (Modificada).

Schrapel *et al.* [60] apresentaram *Pentelligence*, uma caneta para reconhecimento de dígitos manuscritos que opera em papel comum e não requer dispositivo de separação de partituras. *Pentelligence* percebe os movimentos e as emissões sonoras da ponta da caneta ao escrever. A combinação dos dois tipos de dados do sensor melhora substancialmente a taxa de reconhecimento. Os dados dos sons de escrita e dados de movimento são alimentados para redes neurais para votação majoritária.

Ono *et al.* [48], em *Touch and Activate*, apresentam uma técnica de detecção acústica pelo toque que permite a prototipagem de objetos interativos que possuem capacidade de entrada de toque rápida e fácil. Ele reconhece um rico contexto de toques, conectando apenas um alto-falante e um microfone piezoelétrico.

Todos os objetos possuem sua própria propriedade ressonante, representada pelos modos ressonantes, frequência natural e amortecimento modal. A propriedade depende da forma, do material e da fronteira. O estudo foca nas condições de contorno entre eles. Quando um objeto é tocado, a condição de contorno é alterada. Como resultado, sua propriedade ressonante também é alterada. As alterações na propriedade ressonante são observadas como diferentes espectros ressonantes. *Touch and Activate* [48] usa esse fenômeno para estimar como um objeto é tocado pela análise da propriedade ressonante.

### 2.2.3 Reconhecimento de objetos e texturas

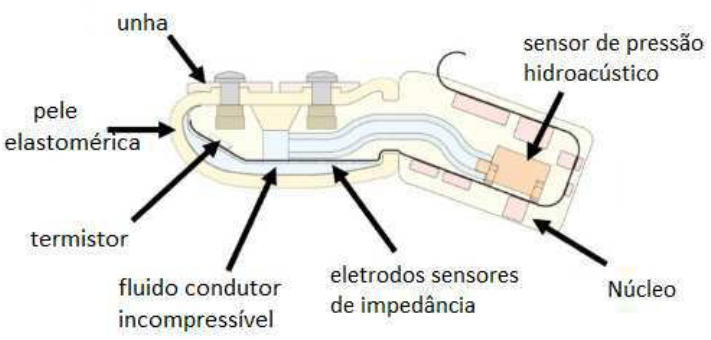
Rasouli *et al.* [55] apresentam uma abordagem biologicamente inspirada para o reconhecimento de objetos. Eles desenvolveram um módulo tátil que pode ser aplicado a grandes áreas por sensores integrados com circuitos de processamento. Para isso, Rasouli *et al.* [55] desenvolveram uma matriz de sensores táteis flexíveis, utilizando material de

tecido piezoresistivo. A saída do arranjo tátil foi representada por um padrão de espícula espaço-temporal para emular sinais neurais de mecanorreceptores na pele. Um *hardware* implementado *Extreme Learning Machine* (ELM) foi usado para processar a informação táctil. Assim, a arquitetura proposta oferece uma alternativa rápida e eficiente em termos energéticos para o processamento de padrões táteis espaço-temporais. O desempenho do sistema foi avaliado durante uma tarefa de classificação de objetos em tempo real, onde obteve 90% de precisão para classificação binária.

Rasouli *et al.* [54] também desenvolveram um sistema neuromórfico para um reconhecimento de texturas, pois é uma das tarefas mais necessárias e desafiadoras para os sistemas sensoriais artificiais. O sistema consiste em um material de tecido piezoresistivo como o sensor para emular a pele, uma interface que produz padrões de pico para imitar os sinais neurais dos mecanorreceptores e um *chip* ELM para analisar a atividade de pico. Beneficiando-se de vantagens intrínsecas de sistemas orientados a eventos biologicamente inspirados e capacidades de processamento maciçamente paralelas e energeticamente eficientes do *chip* ELM, a arquitetura proposta oferece uma alternativa rápida e eficiente para processar informações táteis.

Xu *et al.* [69] integraram sensores multimodal táteis (força, vibração e temperatura) do BioTac®, com uma Mão Sombria Dexterosa, apresentada na Figura 8, e programaram o robô para fazer movimentos exploratórios semelhantes aos que os humanos fazem ao identificar objetos por sua conformidade, textura e propriedades térmicas. Ao identificar um objeto, os movimentos exploratórios são inteligentemente selecionados usando exploração bayesiana, pela qual movimentos exploratórios que fornecem mais desambiguação entre prováveis candidatos a objetos são automaticamente selecionados. O algoritmo foi ampliado com aprendizado por reforço, por meio do qual suas representações internas de objetos evoluíram de acordo com sua experiência cumulativa com eles. O robô identificou corretamente 10 objetos diferentes em 99 de 100 apresentações.

Figura 8 – Esquemático do Biotac®.




Fonte:[69] (Modificada).

Friedl *et al.* [22] se inspiraram na biologia da percepção tátil humana, para imple-

mentar um classificador de texturas neurorobóticas com uma rede neural pontual recorrente, usando uma nova abordagem semi-supervisionada para classificar estímulos dinâmicos. A entrada na rede é fornecida por acelerômetros montados em um braço robótico. Os dados do sensor são codificados por uma população heterogênea de neurônios, modelados para coincidir com a atividade de pico de células mecanorreceptoras. A representação de alta dimensionalidade resultante é então classificada continuamente usando uma SVM. Rasouli *et al.* [55] demonstraram que o sistema desenvolvido classifica 18 texturas de superfícies digitalizadas em duas direções opostas a uma velocidade constante.

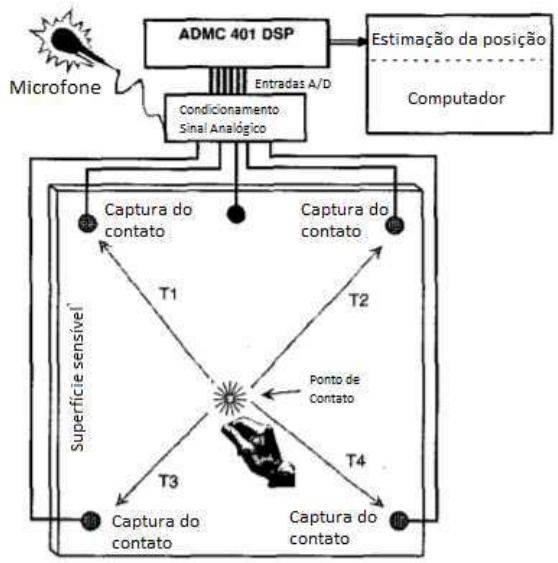
### 2.2.4 Localização espacial da entrada em uma superfície

Para localizar fontes sonoras, diferentes abordagens acústicas têm sido propostas na literatura. Entre eles, a diferença de tempo de chegada (do inglês *Time of arrival* - TDOA) é uma técnica de localização acústica passiva bem conhecida e amplamente utilizada que se baseia na entrada de um conjunto de sensores acústicos, isto é, microfones, estrategicamente conectados ao dispositivo. No TDOA, a diferença de tempo na qual os sinais acústicos são processados permite que o sistema localize a fonte sonora. Em *PingpongPlus* [29], os autores demonstraram essa técnica para rastrear o local onde uma bola de pingue-pongue pousa em uma mesa para projetar *feedback* digital na superfície daquela mesa. Já *Toffee* [68] explorou os princípios do TDOA para detectar eventos de toque de mão em torno de dispositivos móveis na mesa.

Paradiso *et al.* [49] descrevem um sistema que localiza a posição de batidas em uma grande folha de vidro. Utilizando quatro captadores piezoelétricos de contato localizados perto dos cantos da folha para registrar a frente de onda acústica proveniente dos impactos. Um processador de sinal digital extrai características relevantes desses sinais, como amplitudes, componentes de frequência e temporizações diferenciais, que são usadas para estimar a localização da ocorrência e fornecer outros parâmetros, incluindo a resolução aproximada da posição, a natureza de cada ocorrência e a intensidade da batida. Como este sistema requer apenas um *hardware* simples, ele não precisa de nenhuma adaptação especial do painel de vidro, e permite que todos os transdutores sejam montados na superfície interna, portanto, é bastante fácil de implementar como um *retrofit* para as janelas existentes. Isso abre muitos aplicativos, como uma vitrine interativa, com conteúdo controlado por batidas na janela de exibição. A configuração de *hardware* utilizada é apresentada na Figura 9.

*Diamond Touch* [17] é uma tecnologia de toque multiusuário para telas projetadas na frente da mesa. Ele permite que várias pessoas diferentes usem a mesma superfície de toque simultaneamente sem interferir entre si ou serem afetadas por objetos estranhos. *Diamond Touch* [17] funciona transmitindo um sinal elétrico diferente para cada parte da superfície da mesa que se deseja identificar. Quando um usuário toca a mesa, os sinais são acoplados capacitivamente diretamente abaixo do ponto de contato, através do usuário, e

Figura 9 – Configuração do *hardware* de Paradiso *et al.* [49].

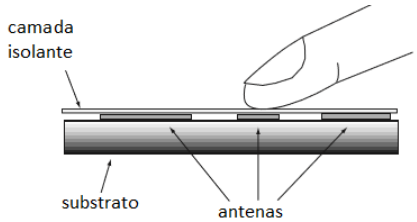



Fonte:[49] (Modificada).

em uma unidade receptora associada a esse usuário. O receptor pode determinar quais partes da superfície da mesa o usuário está tocando.

A superfície da mesa é construída com um conjunto de antenas embutidas que podem ter formato e tamanho arbitrários. Como o acoplamento de sinais para os usuários é feito de forma capacitiva, as antenas também são isoladas dos usuários, e toda a superfície da mesa pode ser coberta por uma camada de material protetor isolante, como apresentado na Figura 10.

Figura 10 – Representação do conjunto de antenas cobertas por um material isolante utilizado em *DiamondTouch* [17].




Fonte:[17].

Matson *et al.* [45] desenvolveram um sistema de detecção de Veículo Aéreo Não Tripulado (VANT) com múltiplos nós acústicos usando modelos de aprendizado de máquina. Características, incluindo MFCC e Transformada de Fourier de Curta Duração (do inglês *Short-time Fourier transform* - STFT) foram usadas para o treinamento. As SVM e as CNN foram treinadas com os dados coletados. Experimentos foram feitos para avaliar a capacidade dos modelos de encontrar o caminho do VANT que estava voando.

## 2.3 Considerações Finais

Neste capítulo foram analisados artigos científicos que utilizam a técnica de detecção acústica, com foco no *hardware* utilizado, nas características do sinal e em seu processamento, e nas técnicas de aprendizado de máquina utilizadas, com a finalidade de validar a inovação do trabalho proposto.

Dependendo da abordagem utilizada para implementar a detecção acústica e da finalidade do trabalho, dividiu-se os artigos científicos em abordagens passivas e ativas. Nas abordagens passivas detecta-se a entrada de um usuário, adquirindo e analisando sons gerados pelas ações do mesmo. Uma abordagem passiva permite uma interação muito mais conveniente e simples com as superfícies, pois não precisa de um componente ativo para funcionar corretamente. Já as abordagens ativas, um objeto explícito (por exemplo, uma caneta especial ou um alto-falante) deve ser usado para interagir com uma superfície.

Este capítulo também apresentou a escolha de cada autor para o reconhecedor de padrões utilizado em cada artigo científico. Dessa forma, vários algoritmos para o processamento de informações acústicas foram explorados. Estes incluem Máquina de Vetores de Suporte, Redes Neurais, Árvores de Decisão, Redes escondidas de Markov e Análise Bayesiana.

Como Rede Neural Artificial e os Modelos escondidos de Markov são técnicas mais comuns no contexto da tarefa de reconhecimento de gestos em superfícies, com baixo custo computacional, no Capítulo 3, é apresentada uma fundamentação teórica sobre essas técnicas de aprendizado de máquina.

# 3 Fundamentação Teórica

Neste capítulo é apresentada uma revisão bibliográfica sobre Redes Neurais Artificiais e Modelos de Markov escondidos, modelos de algoritmos que podem ser aplicados em diversos tipos de tarefas, por exemplo, na estimativa de variáveis para monitoramento de processos, aproximação de funções e predição. Neste trabalho, a tarefa adotada é a de reconhecimento de gestos em superfícies.

## 3.1 Redes Neurais Artificiais

Os computadores podem ser inteligentes? Alan Turing deu muita atenção a esta questão. Ele acreditava que as máquinas poderiam ser criações que imitam os processos do cérebro humano e que não havia nada que o cérebro poderia fazer que um computador bem projetado não pudesse também [18].

A busca por um modelo computacional que simulasse o funcionamento das células do cérebro data da década de 50. Em 1958, Rosenblatt propôs um método inovador de aprendizagem para as redes neurais artificiais denominado *perceptron*.

Em 1960, [67] Widrow e Hoff apresentaram o seu modelo computacional denominado *Adaline Adaptive Linear Neuron* que se diferenciava do *perceptron* por possuir saídas binárias bipolares (-1 ou 1). No final dos anos 60, Minsky e Pappert (1969) publicaram um livro no qual apresentaram importantes limitações do *perceptron*. A retomada das pesquisas aconteceu na década de 80, após Hopfield demonstrar as propriedades associativas das RNAs com problemas físicos, e a descrição do algoritmo de treinamento *back propagation* (1986) que possibilitou as redes de múltiplas camadas solucionarem problemas com alta complexidade [28]. A partir dessas descobertas e graças a avanços metodológicos importantes e ao aumento dos recursos computacionais disponíveis, houve uma nova explosão de interesse científico pelas RNAs.

A capacidade de implementar computacionalmente versões simplificadas de neurônios biológicos deu origem a uma subespecialidade da inteligência artificial, conhecida como Redes Neurais Artificiais (RNAs), que podem ser definidas como sistemas paralelos compostos por unidades de processamento simples, dispostas em camadas e altamente interligadas, inspiradas no cérebro humano [13].

O cérebro é um sistema de processamento de informação altamente complexo, não linear e paralelo. Ele tem a capacidade de organizar seus constituintes estruturais, conhecidos por neurônios, de forma a realizar certos processamentos (reconhecimento de padrões, percepção, e controle motor) [27]. Algumas definições sobre as redes neurais

artificiais são apresentadas a seguir.

Segundo Haykin [27], a definição de RNA é: uma rede neural é um processador paralelamente distribuído constituído de unidades de processamento simples, que têm a propensão natural para armazenar conhecimento experimental e torná-lo disponível para o uso. Ela se assemelha ao cérebro em dois aspectos:

1. o conhecimento é adquirido pela rede a partir de seu ambiente através de um processo de aprendizagem;

2. forças de conexão entre neurônios, conhecidas como pesos sinápticos, são utilizadas para armazenar o conhecimento adquirido.

Segundo Kohonen [36], as RNAs são definidas como redes massivamente paralelas e interconectadas, de simples elementos. Esses elementos devem interagir com dados do mundo real, assim como o sistema nervoso biológico.

Segundo Loesch e Sari [41], as RNAs são sistemas computacionais, de implementação em *software* ou *hardware*, que imitam as habilidades "computacionais" do sistema nervoso biológico, utilizando um grande número de neurônios artificiais interconectados.

A principal força na estrutura de redes neurais reside em suas habilidades de adaptação e aprendizagem. A habilidade de adaptação e aprendizagem pelo ambiente significa que modelos de redes neurais podem lidar com dados imprecisos e situações não totalmente definidas. Uma rede treinada tem a habilidade de generalizar quando é apresentada a entradas que não estão presentes em dados já conhecidos por ela [33].

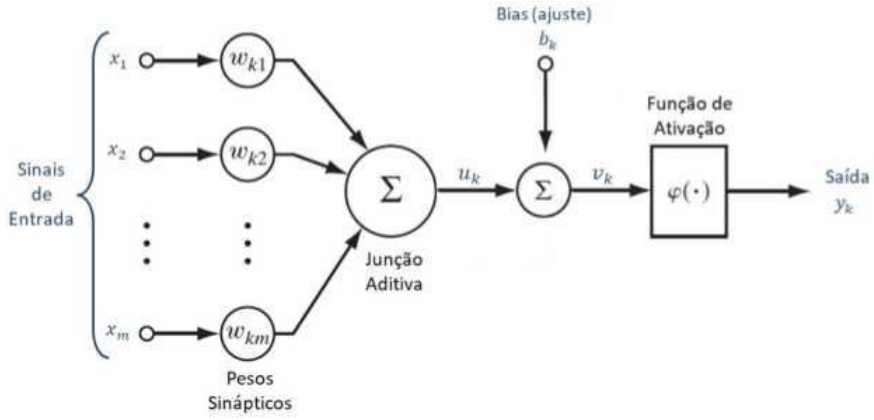
É importante saber que esses níveis estruturais de organização são uma característica única do cérebro. Eles não são encontrados em lugar algum em um computador digital convencional, e não se está próximo de recriá-los com redes neurais artificiais. Os neurônios artificiais utilizados para construir as redes neurais são realmente primitivos em comparação com aqueles encontrados no cérebro. Apesar disso, com a analogia neurobiológica como fonte de inspiração e com a riqueza das ferramentas teóricas e tecnológicas que se tem acumulado, se tem avançado gradualmente na compreensão das redes neurais artificiais e o seu uso tem gerado resultados satisfatórios [27].

### 3.1.1 O modelo do neurônio artificial

No diagrama de blocos da Figura 11, observa-se o modelo de um neurônio, a unidade de processamento de informações fundamental para o projeto de redes neurais artificiais.

É possível identificar três elementos básicos em um modelo neuronal [46]:

Figura 11 – Modelo de neurônio base para projetos de RNA.




Fonte:[27] (modificada).

1. conjunto de sinapses ou elos de conexão, caracterizados por seu peso ou força própria. Um sinal $x_j$ na entrada da sinapse j conectado a um neurônio k é multiplicado pelo peso sináptico $w_{kj}$, onde o índice $k$ refere-se ao neurônio em questão e o índice j refere-se ao terminal de entrada da sinapse. Ao contrário de uma sinapse do cérebro, os valores dos pesos sinápticos de um neurônio artificial podem ser positivos ou negativos, dependendo de as conexões serem inibitórias ou excitatórias;

2. um somador dos sinais de entrada, ponderados pelas respectivas sinapses do neurônio;

3. uma função de ativação para restringir o intervalo permissível de amplitude do sinal de saída de um neurônio. A função de ativação é também referida como função restritiva já que restringe (filtra) o intervalo aceitável de amplitude do sinal de saída a um valor finito. Tipicamente, o intervalo normalizado da amplitude de saída de um neurônio é escrito como o intervalo unitário fechado [0,1] ou [-1,1].

Neste modelo, pode ser notado o acréscimo de um limiar (*bias*) $b_k$, que tem o efeito de aumentar ou diminuir a entrada líquida da função de ativação, promovendo um deslocamento na curva da função de ativação, dependendo se ele é positivo ou negativo [46].

Em termos matemáticos, um neurônio $k$ é descrito pelas equações (3.1), (3.2) e (3.3):

$$u_k = \sum_{j=1}^{m} w_{kj} x_j \tag{3.1}$$

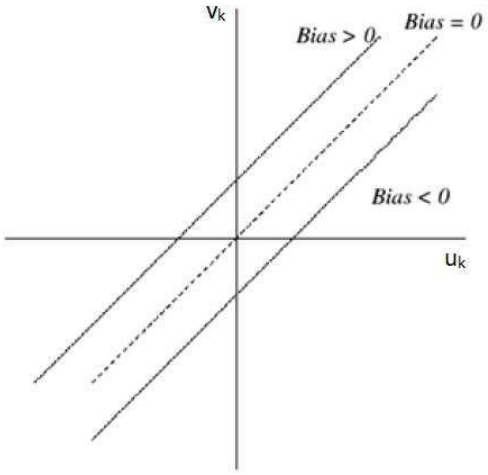
$$v_k = u_k + b_k \tag{3.2}$$

$$y_k = \Phi(v_k) \tag{3.3}$$

em que $x_1$, $x_2$, ..., $x_m$ são sinais de entrada; $w_{k1}$, $w_{k2}$, ..., $w_{km}$, são os pesos sinápticos do neurônio $k$; $u_k$ é a saída do combinador linear devido aos sinais de entrada; $b_k$ é o *bias*, $v_k$ é a soma de $u_k$ com a aplicação do *bias* $b_k$; $\Phi(.)$ é a função de ativação; $y_k$ é o sinal de saída do neurônio.

O uso do *bias* $b_k$ aplica uma transformação na saída $u_k$ do combinador linear, aumentando os graus de liberdade e permitindo uma melhor adaptação, por parte da rede neural, ao conhecimento a ela fornecido. Se o *bias* $b_k$ for positivo ou negativo, a relação entre o potencial de ativação $v_k$ do neurônio $k$ e do integrador de saída $u_k$ é modificado de acordo com a Figura 12.

Figura 12 – Representação da transformação afim produzida pelo bias.




Fonte: [8].

A função de ativação $\phi(.)$ define a saída de um neurônio j em termos de um potencial de ativação $v_k$, limitando o resultado a um intervalo conhecido. Isto adiciona uma não-linearidade ao sistema e evita que informações se propaguem pelas camadas da RNA sem limite numérico de crescimento, o que pode ocasionar a saturação dos neurônios e a perda de eficiência da rede. Existem diversas funções de ativação que são aplicadas nas redes neurais artificiais, as mais comuns são: a função de limiar (degrau) e a função logística, descritas à seguir.

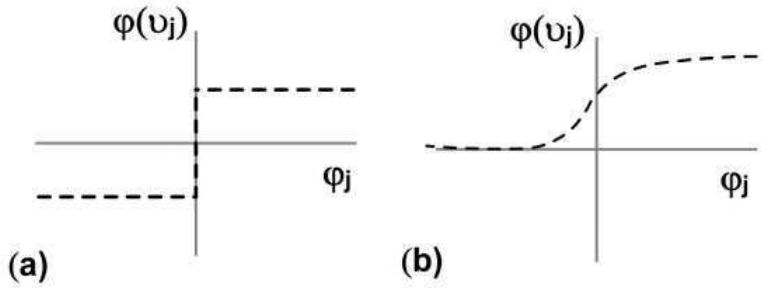
Função de limiar (degrau): A saída do neurônio é igual a zero, quando seu valor for negativo e 1, quando seu valor for positivo. Em termos matemáticos, essa função é descrita pela expressão (3.4). O gráfico desta função é apresentado na Figura 13 (a).

$$\Phi(v_k) = \begin{cases} 0, v_k < 0 \\ 1, v_k \geq 0 \end{cases} \tag{3.4}$$

Função logística: é uma função contínua não linear que permanece dentro de alguns limites superiores e inferiores. O aspecto não linear permite que redes neurais façam mapeamento não linear entre entradas e saídas. A função logística é um dos modelos mais utilizados em projetos de redes neurais [27]. O gráfico desta função forma um "S", como apresentado na Figura 13 (b). As funções sigmoides são as funções mais empregadas nas camadas internas de uma *Multilayer Percepton* (MLP) por serem continuas, crescentes, diferenciáveis e não lineares. Um exemplo de função logística é a função definida pela expressão (3.5), onde $a$ representa a suavidade da função.:

$$\Phi(v_k) = \frac{1}{1 + e^{-av_k}} \tag{3.5}$$

Figura 13 – Representação gráfica de diferentes funções de ativação: (a) função degrau; (b) função logística.




Fonte:[8].

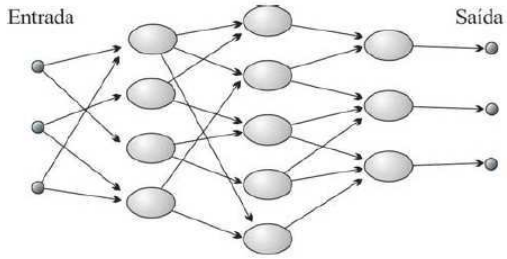
### 3.1.2 Funcionamento das Redes Neurais Artificiais

Combinando diversos neurônios, forma-se uma rede neural artificial. De uma forma simplificada, uma rede neural artificial pode ser vista como um grafo onde os nós são os neurônios e as ligações fazem a função das sinapses, como exemplificado na Figura 14.

Os neurônios são dispostos em camadas, de forma que a camada que recebe os sinais de entrada e a camada da qual se extraem os sinais de saída são denominadas camadas visíveis, de entrada e de saída, respectivamente, e as demais camadas intermediárias, caso existentes, são denominadas de camadas escondidas [27].

Definido o modelo de neurônio, a estrutura de combinação destes e suas conexões na rede devem ser especificadas. As RNAs se distinguem em função da forma como se dá seu treinamento e pela topologia ou arquitetura que apresentam. A arquitetura de uma rede neural restringe o tipo de problema no qual a rede poderá ser utilizada, e é

Figura 14 – Representação simplificada de uma RNA.



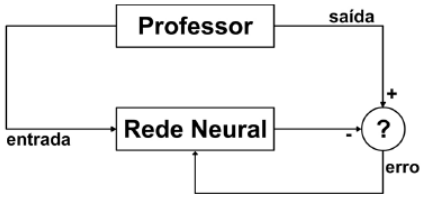

Fonte:[20].

definida pelo número de camadas (camada única ou múltiplas camadas), pelo número de nós em cada camada, pelo tipo de conexão entre os nós (*feedforward* ou *feedback*) e por sua topologia [27].

O treinamento de uma RNA é um método de alterar os pesos associados às sinapses, $w_{kj}$, de maneira que estes convirjam de forma tal que a RNA possa representar um modelo matemático ou realizar a classificação do objeto em estudo. Esses treinamentos podem ser com ou sem supervisão.

O aprendizado supervisionado, apresentado na Figura 15, requer um conjunto de treinamento que consiste em vetores de entrada e um vetor de destino (saída desejada) associado a cada vetor de entrada. Portanto, é necessário ter um conhecimento prévio do comportamento que se deseja ou se espera da rede. Para cada entrada, o conjunto de treinamento indica explicitamente se a resposta calculada é boa ou ruim. O objetivo com o treinamento supervisionado é ajustar os valores de pesos sinápticos de forma que o erro entre a saída real e a saída desejada seja minimizado [18].

Figura 15 – Diagrama de um mecanismo de aprendizado supervisionado.




Fonte: [21].

Existem diferentes tipos de redes neurais que aprendem sob supervisão. Pode-se citar, por exemplo, redes neurais *feedforward* de múltiplas camadas, redes *feedforward* em cascata e recorrentes.

Na aprendizagem não supervisionada, ou aprendizado auto-supervisionado, somente os padrões de entrada estão disponíveis para a rede neural. A rede processa as entradas e, detectando suas regularidades, tenta progressivamente estabelecer representações internas

para codificar características e classificá-las automaticamente. Este tipo de aprendizado só é possível quando existe redundância nos dados de entrada, para que se consiga encontrar padrões em tais dados [20].

### 3.1.3 Tipos de Redes Neurais

Geralmente, identifica-se dois tipos básicos de arquiteturas de RNA: redes com propagação para frente (*feedforward*) ou redes recorrentes. As redes *feedforward*, formadas por uma ou mais camadas, recebem os sinais externos e propagam esses sinais por todas as camadas para obter o resultado (saída) da rede neural. Não há conexões de realimentação para as conexões anteriores. As conexões são feitas somente para neurônios da camada seguinte [18]. Dentre as diversas arquiteturas de RNA com propagação para frente de múltiplas camadas, destaca-se como a mais difundida a *Multilayer Perceptron* (MLP). Elas representam uma generalização do *perceptron* de camada única [27].

As redes neurais recorrentes, por outro lado, têm conexões sinápticas realimentadas (ou laços de realimentação) permitindo o fluxo de sinais de ativação e saída neurais entre neurônios de camadas distintas para modelar as características temporais do problema que está sendo aprendido [18]. Estas redes podem ter suas estruturas não obrigatoriamente organizadas em camadas, e quando são, podem possuir interligações entre neurônios de mesma camada e entre camadas não consecutivas, gerando interconexões bem mais complexas do que as redes não-recorrentes.

Entre as principais classes de RNAs, destacam-se as redes *Adaline*, as *Multilayer Perceptron*, Memórias Matriciais, *Self-Organizing*, Processamento Temporal, entre outras. A autora deste trabalho propõe-se a utilizar a rede com propagação para frente de camadas múltiplas MLP, que será descrita em detalhes no próximo tópico.

#### 3.1.3.1 *Multilayer Perceptron*

O *Perceptron* é a mais simples forma de uma rede neural, usada para classificação de problemas de um tipo especial de padrões ditos linearmente separáveis [27]). Basicamente consiste de um único neurônio com pesos sinápticos ajustáveis e limiar. Contudo, o mundo real possui problemas com maior grau de complexidade, sendo apenas um perceptron insuficiente nesses casos.

Na MLP, os nós de origem na camada de entrada da rede fornecem os respectivos elementos do padrão de ativação (vetor de entrada), que constituem os sinais de entrada aplicados aos neurônios (nós de computação) na segunda camada (ou seja, a primeira camada oculta). Os sinais da segunda camada são usados como entradas para a terceira camada, e assim por diante, para o resto da rede. Tipicamente, os neurônios em cada camada da rede têm como entradas apenas os sinais de saída da camada precedente. Os

sinais de saída dos neurônios na camada de saída (final) da rede constituem a resposta geral da rede ao padrão de ativação fornecido pelos nós de origem na (primeira) camada de entrada [46].

Entre a camada de entrada e a de saída, pode-se ter uma ou mais camadas ocultas. As camadas ocultas proporcionam complexidade e não-linearidade para a rede. Não existe um método que determine o número ideal de camadas ocultas e de neurônios nessa camada. Porém a escolha desses parâmetros é muito importante e influencia diretamente o desempenho do sistema, pois o tempo computacional para o cálculo da resposta e para o treinamento da rede aumenta significativamente com o aumento das conexões e de neurônios nas camadas ocultas [8].

Segundo Haykin [27] Uma rede *perceptron* multicamada tem três características:

1. O modelo de cada neurônio na rede inclui uma não-linearidade na saída final. Esta não-linearidade pode ser garantida por funções de ativação do tipo logística, por exemplo.

2. A rede contém uma ou mais camadas escondidas que não são partes da entrada ou saída.

3. A rede exibe um alto grau de conectividade, determinada pelas sinapses da rede.

Na Figura 16 é apresentada uma rede neural *feedforward* multicamada para o caso de uma única camada oculta. A rede da Figura 16 é referida como uma rede 10-4-2, porque possui 10 nós de origem, 4 neurônios ocultos e 2 neurônios de saída [27].

### 3.1.4 Algoritmos de Treinamento

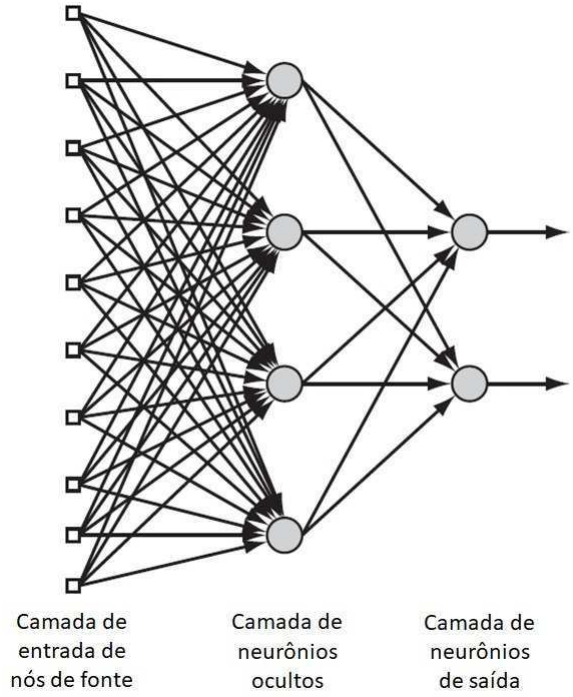
Durante a etapa de treinamento, um conjunto de dados de entrada e saída é apresentado sucessivas vezes à RNA. A cada iteração, os pesos sinápticos são ajustados até o erro sobre todo o conjunto de treinamento convergir para um valor mínimo, ou até que o número de iterações, determinado anteriormente, tenha sido atingido [27].

Algebricamente, o termo $e_j(t)$, descrita em (3.6), relativo ao erro é escrito como a diferença entre o valor $y_j(t)$ calculado pela RNA no instante t e o valor da saída esperado $d_j(t)$.

$$e_j(t) = d_j(t) - y_j(t) \tag{3.6}$$

A cada etapa de treinamento são feitas pequenas modificações nos pesos, provocando uma minimização incremental dos erros, convergindo em direção ao valor esperado. Adota-se a soma dos erros quadráticos de todas as saídas como parâmetro de desempenho da

Figura 16 – Representação das redes alimentadas adiante com uma camada oculta e uma de saída.




Fonte: [27].

rede, e também como função de custo $E(t)$, descrita em (3.7), a ser minimizada pelo algoritmo de treinamento. Este é o princípio da regra delta de Widrow e Hoff utilizada no treinamento de RNAs [21].

$$E(t) = \frac{1}{2}\sum_{j=0}^{n} e_j^2(t) \tag{3.7}$$

A função de custo pode ser visualizada como uma superfície de erro, com os parâmetros livres do sistema (pesos e *bias*) como coordenadas. Ela pode apresentar um único ponto de mínimo, quando possui somente funções de ativação lineares, ou pode apresentar vários mínimos locais além do mínimo global, se houverem nodos não lineares [13]. O objetivo é que, ao longo do treinamento, o erro parta de um ponto arbitrário da superfície e se desloque até o mínimo global.

Existem diversos algoritmos de treinamento que diferem basicamente pelo modo como é realizado o ajuste dos pesos sinápticos, dentre os quais podem-se citar *Resilient Back-Propagation*, gradiente conjugado, Quase-Newton e Levenberg-Marquardt *backpropagation*. O algoritmo de retropropagação é o mais comumente empregado no treinamento supervisionado de redes MLP [27] e descrito em detalhes no próximo tópico.

#### 3.1.4.1 Algoritmos de Retropropagação

Em 1986, Rumelhart *et al* apresentaram o algoritmo de treinamento de retropropagação do erro (*backpropagation*). Esse algoritmo de aprendizagem supervisionado utiliza pares (entrada, saída desejada) para, por meio de um mecanismo de correção de erros, ajustar os pesos sinápticos da rede, minimizando o Erro Médio Quadrático (EMQ) entre o valor desejado e o valor da saída atual da rede. Neste tipo de treinamento, um sinal de erro é propagado de volta através da rede, começando com os neurônios de saída e se movendo em direção aos neurônios de entrada. As sinapses são modificadas com base na atividade neural e no sinal de erro retropropagado [32].

Então, existem duas fases para o treinamento, cada uma em um sentido da rede. A fase de propagação (*forward*), em que é realizado o cálculo das saídas e seus respectivos erros e a fase de retropropagação (*backward*), em que é realizada a atualização dos pesos sinápticos de suas conexões [40].

A implementação computacional do algoritmo de retropropagação apresenta os seguintes passos [27]:

- Passo 1: Inicialização de todos os pesos e parâmetros.
- Passo 2: Fase *forward*, consiste em:
  - Dadas as entradas, calcular as saídas para todas as camadas da rede;
  - Calcular o erro de saída da rede
- Passo 3: Fase *backward*, consiste em efetuar o cálculo das atualizações dos pesos entre as camadas da rede, iniciando a partir da última camada, até chegar à camada de entrada;

O ajuste dos pesos sinápticos pode ser realizado a cada iteração (treinamento sequencial) ou após a apresentação na rede de todos os exemplos do conjunto de dados de treinamento (treinamento por lote). Segundo Braga *et al.* [13], o treinamento sequencial é geralmente mais rápido e requer menor demanda computacional (tempo e memória física), porém é mais instável.

Existem diversos algoritmos para implementar o treinamento por retropropagação. Alguns serão brevemente descritos a seguir com foco nas suas principais características.

O *backpropagation* padrão emprega o gradiente descendente como método de aproximação do mínimo da função erro. O algoritmo de Levenberg-Marquardt (LM) utiliza uma aproximação pelo método de Newton, obtida a partir da modificação do método de Gauss-Newton [35].

O algoritmo de redução do gradiente por lote (*Batch Gradient Descent*) ajusta os pesos e desvios em direção ao gradiente negativo da função custo. O algoritmo *Batch Gradient Descent with Momentum* apresenta frequentemente melhores resultados ao introduzir o conceito de momento, que leva em conta os ajustes anteriores nos pesos, observando a recente tendência de mudanças na superfície de erro, desprezando pequenas sinuosidades na superfície de erro. Tal comportamento evita que o algoritmo fique preso a um pequeno mínimo local da superfície de erro, podendo mais facilmente chegar ao mínimo global [15].

O algoritmo de Levenberg-Marquardt é considerado o algoritmo mais rápido para redes de tamanho moderado (até algumas centenas de pesos). Nele, o gradiente é computado pela transposta da matriz Jacobiana que possui as derivadas primeiras da função custo em função dos erros. Foi incorporado ao algoritmo de retropropagação do erro para resolver problemas de otimização que aparecem no treinamento de redes multicamadas [65].

Como neste projeto utilizar-se-á o *software* MATLAB nas simulações para o projeto das redes neurais, apresentam-se na Tabela 1 as funções do MATLAB referentes aos algoritmos de aprendizado para uma rede do tipo retropropagação.

Tabela 1 – Algoritmos de aprendizado.

| ALGORITMO | DESCRIÇÃO |
|---|---|
| TRAINLM | *Backpropagation* Levenberg-Marquardt |
| TRAINGD | *Backpropagation* de gradiente descrescente |
| TRAINGDM | *Backpropagation* de gradiente descrescente com momentum |
| TRAINGDA | *Backpropagation* de gradiente descrescente com taxa adaptativa |
| TRAINGDX | *Backpropagation* de gradiente descrescente com momentum e taxa adaptativa |

Serão realizadas comparações do comportamento de diferentes algoritmos de treinamento, com o objetivo de contribuir para as pesquisas na área, além de encontrar o algoritmo que mais se adapte ao conjunto de dados para o reconhecimento dos padrões em estudo, seguindo a metodologia proposta.

Na próxima seção, os Modelos Escondidos de Markov serão apresentados como uma alternativa de algoritmo de treinamento de máquina para problemas de reconhecimento de padrões.

## 3.2 Modelos Escondidos de Markov

A teoria relativa aos Modelos Escondidos de Markov já é bem conhecida e extensivamente documentada. Esta Seção tem o propósito de apresentar alguns conceitos básicos relacionados aos Modelos Escondidos de Markov, com o objetivo de fornecer uma base teórica ao entendimento do experimento realizado utilizando HMM para o reconhecimento de gestos apresentado na Seção 4.2. Os textos de Lawrence e Deller [38, 14] contém mais informações sobre HMM.

A teoria de Modelos Escondidos de Markov foi introduzida na literatura na década de 1960 por Baum. Sua utilização na área de reconhecimento automático de fala se deu na década de 1970, introduzida pelos trabalhos independentes de Baker, na Carnegie Mellon University [4], e Jelinek e colegas, na IBM [30] e, desde então, passou a ser largamente utilizada em diversas aplicações, pois modelam bem os aspectos estatísticos e as sequências do sinal de voz.

Este trabalho procura realizar o reconhecimento de gestos realizados em uma superfície por meio da entrada do sinal acústico. Dessa forma, tem-se comportamento parecido ao sinal de voz, pois os sinais possuem a mesma forma de aquisição e a mesma natureza no tempo. Por isso que referências de trabalhos com reconhecimento de fala foram pesquisadas.

Os Modelos Escondidos de Markov são bastante utilizados em sistemas de reconhecimento por meio de entradas acústicas pois têm um algoritmo eficiente e robusto para o treinamento e reconhecimento. O treinamento do modelo consiste em modelar o conjunto dos parâmetros acústicos extraídos do sinal observado por uma sequência de estados (cadeia de Markov de primeira ordem) de acordo com a variação temporal do sinal apresentado. Já no reconhecimento, a sequência de observações de teste é aceita como verdadeira se possuir uma medida de similaridade (verossimilhança) acima de um limiar estipulado com os parâmetros do modelo [57].

Um HMM consiste em um modelo estatístico baseado na teoria dos processos de Markov, diferenciando-se pelo fato dos seus estados não serem conhecidos, mas apenas o sinal emitido em cada um dos estados. Deste modo, é definido como um par de processos estocásticos (X,Y), em que X representa uma cadeia de Markov de primeira ordem e não é diretamente observável, enquanto Y é uma sequência de variáveis aleatórias que assumem valores no espaço de parâmetros acústicos (observações) [57].

Assim, um HMM é caracterizado por um conjunto de N estados conectados por transições. Em cada instante de tempo $t$ existe uma transição do estado atual para outro estado, ou a transição para o mesmo estado atual, e um símbolo é emitido com uma determinada densidade de probabilidade de saída, função do estado atual. A sequência de símbolos emitidos é chamada de sequência de observações, que representa a saída do HMM [57].

Na próxima seção, aborda-se a teoria das cadeia de Markov, para que na seção seguinte o leitor possa ter um melhor entendimento modelo de Markov escondido e suas características.

### 3.2.1 Cadeias de Markov

A sequência $X_1, X_2, ..., X_n$ de variáveis aleatórias é chamada de uma sequência de Markov. Uma cadeia de Markov é uma sequência de Markov de estado discreto com a propriedade de que a distribuição de probabilidade do próximo estado depende apenas do estado atual e não na sequência de eventos que precederam. O espaço de estados de uma cadeia de Markov é de natureza discreta e finita: $q_t \in \{s^j, j = 1, 2, ..., N\}$. Cada um desses valores discretos está associado a um estado na cadeia de Markov [70].

As mudanças de estado do sistema são chamadas de transições. As probabilidades associadas a várias mudanças de estado são chamadas probabilidades de transição.

A cadeia de Markov, $q_1^T = q_1, q_2, ..., q_T$, é completamente caracterizada pela probabilidade de transição, definida por:

$$P(q_t = s^{(j)} | q_{t-1} = s^{(i)}) = a_{ij}(t) \tag{3.8}$$

e pelas probabilidades iniciais de distribuição de estado ou distribuição inicial $P(0)$. Se essas probabilidades de transição são independentes do tempo $t$, tem-se uma cadeia de Markov homogênea [70].

As probabilidades de transição de uma cadeia de Markov (homogênea) geralmente são convenientemente colocadas em forma de matriz:

$$A = [a_{ij}], \text{ onde } a_{ij} \leq 1 \text{ e } \sum_{j=1}^{N} a_{ij} = 1 \tag{3.9}$$

que é chamada de matriz de transição da cadeia de Markov [70].

### 3.2.2 Caracterização de um modelo de Markov escondido

Analisando a cadeia de Markov discutida acima como uma fonte de informação capaz de gerar sequências de saída observacionais, pode-se chamar a cadeia de Markov de uma sequência observável de Markov porque sua saída tem correspondência individual com um estado. Ou seja, cada estado corresponde a uma variável ou evento deterministicamente observável. Não há aleatoriedade na saída em qualquer estado. Essa falta de aleatoriedade torna a cadeia de Markov muito restritiva para descrever de maneira adequada muitas fontes de informação do mundo real, como sequências de recursos de fala de maneira adequada.

Uma extensão do princípio de cadeia de Markov para incorporar aleatoriedade que se sobrepõe entre os estados da cadeia Markov dá origem à sequência escondida de Markov. Essa extensão é alcançada associando uma distribuição de probabilidade de observação com cada estado da cadeia de Markov. A sequência de Markov assim definida

é uma sequência aleatória duplamente embutida cuja cadeia de Markov subjacente não é diretamente observável, daí uma sequência escondida. A cadeia de Markov subjacente na sequência escondida de Markov pode ser observada apenas através de uma função aleatória separada caracterizada pelas distribuições de probabilidade de observação [70].

Se as distribuições de probabilidade de observação não se sobrepuserem entre os estados, a sequência de Markov subjacente não ficará escondida. Isso ocorre porque, apesar da aleatoriedade incorporada nos estados, qualquer valor observacional em um intervalo fixo específico de um estado seria capaz de mapear exclusivamente esse estado. Nesse caso, a sequência escondida de Markov reduz-se essencialmente a uma cadeia de Markov. Alguma exposição mais detalhada sobre a relação entre uma cadeia de Markov e sua função probabilística, ou uma sequência escondida de Markov, pode ser encontrada em [51].

Quando uma sequência escondida de Markov é usada para descrever uma fonte informativa do mundo real, isto é, para aproximar as características estatísticas de uma fonte, geralmente a chamamos de modelo de Markov escondido. Um uso prático muito bem-sucedido do HMM foi em aplicações de processamento de voz, incluindo reconhecimento de voz e sua robustez de ruído, síntese de voz e aprimoramento de fala.

Assim, uma caracterização formal de um modelo de sequência escondida de Markov em termos de seus elementos e parâmetros básicos é:

1. N, o número de estados do modelo;

2. Distribuição de probabilidade de transição entre os estados, $A = [a_{ij}], i, j = 1, 2, ..., N$, de uma cadeia Markov homogênea com um total de N estados;

$$a_{ij} = P(q_t = j | q_{t-1} = i) \tag{3.10}$$

3. Distribuição do estado inicial $\Pi = [\pi_i]$, onde $\pi_i = P(q_1 = i)$, em que $1 \leq i \leq N$.

4. B, representa a distribuição de probabilidade de observação dos símbolos, $B = [b_j(k)]$. Para o HMM discreto, seus elementos são do tipo $b_j(k) = P[o_t = v_k | q_t = j]$, sendo $1 \leq j \leq N$, $1 \leq k \leq M$, em que $b_j(k)$ é a probabilidade da variável aleatória $o_t$ (observação) ser igual ao k-ésimo símbolo observado no alfabeto, $v_k$, dado que o estado atual da cadeia de Markov é o estado $j$. As condições estocásticas seguintes devem ser satisfeitas:

$$0 \leq b_j(k) \leq 1, 1 \leq j \leq N, 1 \leq k \leq M \text{ e } \sum_{k=1}^{M} b_j(k) = 1 \tag{3.11}$$

Uma forma compacta é utilizada para indicar o conjunto completo de parâmetros do modelo

$$\lambda = (A, B, \Pi) \tag{3.12}$$

### 3.2.3 Classificação dos HMMs

Os HMMs podem ser classificados segundo dois critérios: quanto à distribuição de probabilidade associada a cada estado e quanto à topologia [70].

A cada estado do HMM é associada uma distribuição de probabilidade. De acordo com essa distribuição, que pode ser função densidade de probabilidade (caso contínuo), ou função massa de probabilidade (caso discreto), os HMMs podem ser classificados como contínuos ou discretos [70].

O HMM é dito discreto quando o número de possíveis símbolos de saída é finito e a probabilidade de se emitir o símbolo $v_k$, no estado $q_j$, é dada por $b_j(k)$, com as seguintes propriedades:

$$b_j(k) \geq 0, 1 \leq j \leq N, 1 \leq k \leq M \text{ e } \sum_{k=1}^{M} b_j(k) = 1 \tag{3.13}$$

em que, N é o número de estados do HMM; M é o número de símbolos discretos do modelo e $b_j(k)$ é a probabilidade de emitir o símbolo $v_k$ no estado $q_j$.

O HMM é dito contínuo quando sua função densidade de probabilidade for contínua. Usualmente utiliza-se a função densidade de probabilidade modelada por uma mistura de M gaussianas multidimensionais, representadas por:

$$b_j(o_t) = \sum_{k=1}^{M} c_{jk} G(o_{t,j\,k}, U_{jk}) \tag{3.14}$$

em que:

- $o_t$ é o vetor de parâmetros de entrada de dimensão D no instante de tempo t;
- $M$ é o número de gaussianas na mistura para cada estado;
- $c_{jk}$ é o coeficiente de ponderação para a k-ésima mistura no estado $j$;
- $G$ é a função densidade de probabilidade gaussiana multidimensional com vetor média $_{jk}$ e matriz de covariância $U_{jk}$ para o componente da k-ésima mistura no estado $j$.

### 3.2.4 Topologia dos HMMs

As probabilidades das transições definem o processo de Markov e sua ordem. Quando a transição feita para o estado atual não depender da ocorrência de todos os

estados anteriores, mas, somente do estado imediatamente anterior, é caracterizado um Processo de Markov de Primeira Ordem. De acordo com a matriz de transição $A$, a Cadeia de Markov assume uma certa topologia. A topologia, ou estrutura de um HMM, é determinada pelas transições que ocorrem entre estados.

Deste modo, os HMMs são classificados em ergódicos (totalmente conectados) ou *left-right* (esquerda-direita), também conhecido como modelo de Bakis [38].

O modelo ergódico, ilustrado na Figura 17 (a), tem a característica de não restringir nenhuma transição entre os estados, ou seja, a partir de um estado é possível atingir todos os outros estados, fazendo com que sua matriz de transição de estados fique totalmente preenchida [50].

A principal desvantagem deste tipo de modelo está na dificuldade em modelar a sequência temporal dos eventos acústicos em cada estado, além de, no processo de treinamento, aumentar o risco de convergência em um máximo local. Sendo assim, quando este modelo é utilizado para o reconhecimento de voz, as probabilidades de transição de retorno obtidas são próximas a zero [38].

O modelo esquerda-direita é apresentado na Figura 17 (b), na qual a sequência de estados associada ao modelo tem a propriedade de nenhuma transição ser permitida para estados cujo índice seja menor do que o atual. No HMM do tipo esquerda-direita, a primeira observação é obtida quando a cadeia de Markov se encontra em um estado determinado, chamado de estado inicial. A última observação é gerada enquanto a cadeia de Markov está em um outro estado, chamado estado final ou estado de absorção. Outra propriedade deste tipo de modelo é que uma vez que a cadeia de Markov deixa um estado, aquele estado não pode ser visitado em um instante posterior [39].

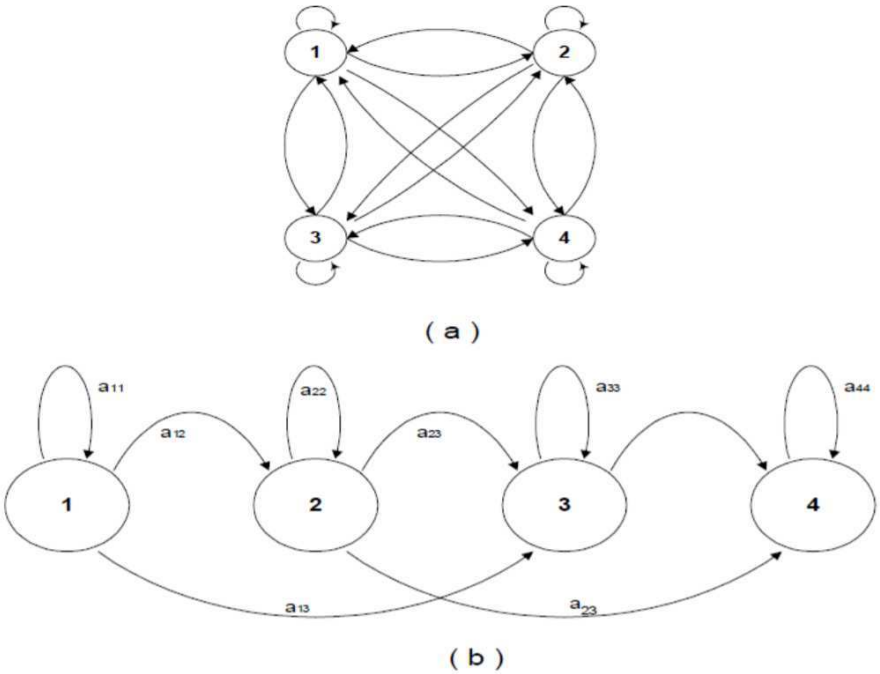
Uma vez que a fala possui uma estrutura inerentemente sequencial, para um sistema de reconhecimento de voz, os modelos esquerda-direta apresentam melhores resultados comparados aos modelos ergódicos. Além disso, a liberdade adicional de transição de estados presente nos modelos ergódicos não reflete as variações dos parâmetros da fala caracterizados por um vetor de padrões [53].

Na tarefa de reconhecimento de fala geralmente são adotadas algumas simplificações da teoria de modelos de Markov, que podem ser formalizadas da seguinte maneira [38]:

1. A suposição de que a cadeia de Markov é de primeira ordem: o próximo estado do HMM depende somente do estado atual, o modelo resultante torna-se, então, um HMM de primeira ordem.

2. A suposição da estacionariedade: assume-se que as probabilidades de transição de um estado para outro não se alteram durante o tempo. Suposição das observações independentes: Assume-se que uma dada observação corrente é estatisticamente

Figura 17 – (a) Modelo ergódico com quatro estados; (b) Modelo esquerda-direita com quatro estados.




Fonte: [50].

independente das observações anteriores e posteriores, ou seja, não há correlação entre observações adjacentes.

### 3.2.5 Modelagem do HMM

A modelagem de um HMM é realizada em três etapas: treinamento, reconhecimento e decodificação.

A etapa de treinamento tem o objetivo de determinar os parâmetros do modelo, como também determinar o melhor conjunto de dados que alimentará o modelo a ser treinado, representando com maior eficiência o sinal que está sendo modelado, de forma que maximize a probabilidade de geração da observação. O método mais conhecido e utilizado para o treinamento dos HMMs é o algoritmo *Forward-Backward*, também conhecido como o algoritmo de re-estimação de Baum-Welch. Este método consiste em um conjunto de equações recursivas, empregando o critério da maximização da verossimilhança, em que o processo de treinamento é repetido enquanto a verossimilhança na interação atual é maior do que a verossimilhança da iteração anterior [52].

O reconhecimento consiste em determinar qual o modelo, dentre os vários obtidos na etapa de treinamento, que provavelmente gerou uma dada sequência de observação. O algoritmo *Forward* é utilizado na solução deste problema.

Na decodificação determina-se a sequência de estados que provavelmente produziu uma determinada sequência de entrada. Este problema é solucionado com o algoritmo de Viterbi.

A solução desses três problemas permite a elaboração de um sistema de reconhecimento automático de gestos utilizando HMM.

#### 3.2.5.1 Treinamento do HMM

O treinamento dos HMMs consiste em ajustar os parâmetros do modelo para satisfazer algum critério de otimização. Ou seja, dado um modelo $\lambda$ e uma sequência de observações $O = O_1, O_2, ..., O_T$, ajustar os parâmetros do modelo $\{A, B, \pi\}$ de modo a representar com maior eficiência o sinal que está sendo modelado maximizando $P\{O|\lambda\}$. Sendo A a distribuição de probabilidade de transição entre os estados, B a distribuição de probabilidade de observação dos símbolos e $\pi$ Distribuição do estado inicial.

O método mais conhecido e utilizado para o treinamento dos HMMs é o algoritmo de Baum-Welch. Este método consiste em um conjunto de equações recursivas, empregando o critério da maximização da verossimilhança, em que o processo de treinamento é repetido enquanto a verossimilhança na interação atual é maior do que a verossimilhança da iteração anterior. Ou seja, o método de Baum-Welch pode ser descrito por meio dos seguintes passos [52]:

1. Para cada $l$-ésima unidade de treinamento, que pode ser palavras ou fonemas, atribuir valores iniciais para os parâmetros do modelo $\lambda_l(A, B, \pi)$ e para a probabilidade $P_l$ que representa os modelos HMM de referência para cada uma das L unidades de treinamento.

2. Com base no algoritmo de re-estimação de Baum-Welch, o segundo passo consiste na re-estimação dos parâmetros do modelo para a obtenção de $\lambda_l^-$.

3. No terceiro passo, deve-se calcular a probabilidade $\overline{P_l}$ associada ao modelo $\lambda_l^-$ re-estimado e fazer a comparação com a probabilidade anteriormente calculada.

4. Se $\overline{P_l} - P_l \leq \delta$(limiar), o processo de re-estimação é finalizado. Caso contrário, retorna-se ao passo 2.

Para definir o conjunto de equações de re-estimação dos parâmetros do modelo por meio do algoritmo de Baum-Welch é necessário definir dois outros algoritmos, *forward* e *backward* [6].

Algoritmo *Forward*

Inicialmente é definida a variável *forward* $\alpha_t(i)$, denominada probabilidade de avanço, como

$$\alpha_t(i) = P\{o_1, o_2, ..., o_t, q_t = i|\lambda\} \tag{3.15}$$

que representa a probabilidade da sequência de observações parciais $o_1, o_2, ..., o_t$ segundo o tempo crescente (iniciando em $t = 1$ até $t = T$), dado o modelo $\lambda$. O algoritmo pode ser resumido em três passos: inicialização, recursão e término.

Inicialização:

$$\alpha_1(i) = \pi_i b_i(o_i), 1 \leq i \leq N \tag{3.16}$$

Recursão:

$$\alpha_{t+1}(j) = [\sum_{i=1}^{N} \alpha_t(i) a_{ij}] b_j(o_{t+1}) \tag{3.17}$$

Término:

$$P\{O|\lambda\} = \sum_{i=1}^{N} \alpha_T(i) \tag{3.18}$$

O valor de $P\{O|\lambda\}$ é uma medida da probabilidade de uma determinado gesto formada pela sequência de observações $O$ ter sido produzida pela sequência de estados $Q = [q_1, q_2, ..., q_t, ..., q_T]$.

Algoritmo *Backward*

De forma similar, inicialmente é definida a variável $\beta_t(i)$, denominada probabilidade de retrocesso, definida como

$$\beta_t(i) = P\{o_{t+1}, o_{t+2}, ..., o_T|q_t = i, \lambda\} \tag{3.19}$$

que representa a probabilidade da sequência de observações parciais do instante $t + 1$ até a última observação no instante $T$, dado que o caminho passa pelo estado $i$ no instante $t$ e dado o modelo $\lambda$. O algoritmo pode ser resumido em dois passos: inicialização e recursão.

Inicialização:

$$\beta_T(i) = 1, 1 \leq i \leq N \tag{3.20}$$

Recursão:

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(o_{t+1}) \beta_{t+1}(j) \tag{3.21}$$

Os valores que são calculados de forma recursiva das variáveis de *forward* e *backward* tendem a se tornar bem menores que um à medida que a sequência de observações é processada, ocasionando *underflow*. Com o objetivo de contornar este problema, utiliza-se um fator de normalização ou o logaritmo dos parâmetros. A normalização consiste em multiplicar os termos $\alpha_t(i)$ e $\beta_t(i)$ por um fator que é independente de $i$, mantendo estes termos dentro da faixa de precisão do computador para $1 \leq t \leq T$.

Para a obtenção de uma boa estimativa dos parâmetros do modelo, uma sequência com uma única observação não é suficiente. Assim, sequências com múltiplas observações devem ser usadas. A sequência de treinamento com múltiplas observações é composta por uma ou mais observações das mesmas palavras.

Os parâmetros são calculados a partir das variáveis *forward* e *backward* normalizadas, do fator de normalização, dos vetores de parâmetros acústicos, dos parâmetros que compõem o modelo e dos valores de verossimilhança normalizada.

#### 3.2.5.2 Reconhecimento do HMM

O problema do reconhecimento consiste em determinar qual modelo HMM que mais provavelmente gerou uma determinada sequência de observações. Matematicamente, dado um modelo $\lambda$ e uma sequência de observações $O = O_1, O_1, ..., O_T$, o reconhecimento busca calcular $P(O|\lambda)$, ou seja a probabilidade de que as observações tenham sido geradas por aquele modelo. O valor de $P\{O|\lambda\}$ é uma medida da probabilidade de uma determinado gesto formada pela sequência de observações $O$ ter sido produzida pela sequência de estados $Q = [q_1, q_2, ..., q_t, ..., q_T]$.

Em sistemas de reconhecimento de fala, há um modelo de HMM treinado para cada unidade acústica. Para cada unidade a ser reconhecida, a sequência de observação é comparada com os modelos treinados por meio do cálculo da probabilidade associada a cada modelo de referência. Para calcular a probabilidade, utilizam-se as equações apresentadas na seção anterior.

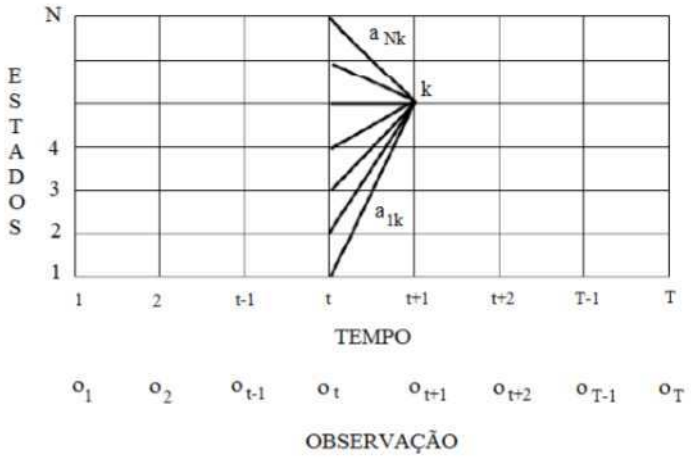
#### 3.2.5.3 Decodificação do HMM

O problema da decodificação consiste em encontrar a sequência de estados ótima, dado um modelo $\lambda$ e uma sequência de observações $O = O_1, O_2, ..., O_T$. Para isto, utiliza-se o algoritmo de Viterbi.

O algoritmo de Viterbi encontra a sequência de estados ótima $q_t$, entre todas as possíveis sequências $q$, utilizando o seguinte critério:

$$q_t* = \arg\max P(q_t = i, O|\lambda) \tag{3.22}$$

Figura 18 – Modelo de Viterbi.




Fonte: [19].

Na Figura 18, ilustra-se uma estrutura de treliça que relaciona a sequência de estados e intervalos de tempo. É possível observar nesta figura que há mais de um caminho parcial chegando a cada nó ou estado, cada um com determinado comprimento ou valor de probabilidade, para vários instantes de tempo diferentes. É chamado de sobrevivente correspondente a cada nó, aquele segmento de caminho mais curto, ou seja, o que apresenta maior valor de probabilidade. Deste modo, para cada instante de tempo, existe um número de sobreviventes igual ao número de nós na treliça [19].

A cadeia de Markov deve terminar em um estado bem determinado, existindo apenas um sobrevivente no último instante de tempo. O caminho total (de $t = 1$ até $t = T$) representa o menor caminho percorrido, ou seja, apresenta o maior valor de probabilidade. Percorrendo de volta a sequência de estados desse caminho, determina-se a sequência de estados associada que fornece o caminho mais provável, ou seja, a sequência de estados ótima [19].

Para a aplicação do algoritmo de Viterbi, é necessário inicialmente definir o maior valor da probabilidade em um caminho, no instante de tempo $t$, ou seja, considerando as $t$ primeiras observações que terminam no estado $q_i$, tem-se por indução que

$$\delta_t(i) = \max_{q_1,q_2,\ldots,q_t} P[q_1, q_2, ..., q_t = i, o_1, o_2..., o_t|\lambda] \tag{3.23}$$

Para se obter a sequência ótima dos estados e maximizar $P[q_1, q_2, ..., q_t = i, o_1, o_2..., o_t|\lambda]$, inicialmente defini-se a variável $\psi_t(j)$. O método para se encontrar a sequência de estados ótima é dado por [52]:

1. Inicialização

$$\delta_1(i) = \pi_i b_i(o_1), \text{para } 1 \leq i \leq N, \tag{3.24}$$

$$\Psi_t(i) = 0 \tag{3.25}$$

2. Recursão

$$\delta_t(j) = \max_{1 \leq i \leq N}[\delta_{t-1}(i)a_{ij}]b_j(o_t), 2 \leq t \leq T, 1 \leq j \leq N \tag{3.26}$$

$$\psi_t(j) = \arg\max_{1 \leq i \leq N}[\delta_{t-1}(i)a_{ij}], 2 \leq t \leq T, 1 \leq j \leq N \tag{3.27}$$

3. Término

$$p* = \max_{1 \leq i \leq N}[\delta_T(i)] \tag{3.28}$$

$$q_T* = \arg\max_{1 \leq i \leq N}[\delta_T(i)] \tag{3.29}$$

4. Sequência de estados ótimos

$$q_t* = \psi_{t+1}(q^*_t{}_{+1}), t = T-1, T-2, ..., 1. \tag{3.30}$$

## 3.3 Considerações Finais

Neste capítulo abordou-se a fundamentação teórica das técnicas de aprendizado de máquina: Rede Neural Artificial e os Modelos Escondidos de Markov, pois são técnicas comuns no contexto da tarefa de reconhecimento de gestos em superfícies e com baixo custo computacional.

As Redes Neurais Artificias são sistemas computacionais de implementação em *software* ou em *hardware*, que imitam as habilidades "computacionais" do sistema nervoso biológico, utilizando um grande número de neurônios artificiais interconectados [41]. Os principais aspectos estruturais de redes neurais são adaptação e aprendizagem, possibilitando lidar com dados imprecisos e situações não totalmente definidas. Uma rede treinada tem a habilidade de generalizar quando é apresentada a entradas que não estão presentes em dados já conhecidos por ela [33].

Este capítulo discorreu conceitos sobre RNA necessários ao entendimento do funcionamento do sistema de reconhecimento de gestos em superfícies. Inicialmente apresentou-se a definição das RNA e o modelo do neurônio artificial. em seguida, descreveu-se o funcionamento da rede e os tipos das arquiteturas da RNA (redes com propagação para frente

ou redes recorrentes). E finalmente, foi explicado com detalhes *Multilayer Perceptron*, classe de RNA que a autora utilizou neste trabalho e o algorítimo de treinamento de retropropagação.

Já os Modelos Escondidos de Markov são bastante utilizados em sistemas de reconhecimento de fala ou reconhecimentos por meio de sinais acústicos, pois têm um algoritmo eficiente e robusto para o treinamento e reconhecimento. O treinamento do modelo consiste em modelar o conjunto dos parâmetros acústicos extraídos do sinal por uma sequência de estados (cadeia de Markov de primeira ordem) de acordo com a variação temporal do sinal. Já no reconhecimento, a sequência de observações de teste é aceita como verdadeira se possuir uma medida de similaridade (verossimilhança) acima de um limiar estipulado com os parâmetros do modelo [57].

Este capítulo abordou conceitos sobre HMM necessários ao entendimento do funcionamento do sistema de reconhecimento de gestos em superfícies. Inicialmente foi apresentado a definição dos Modelos Escondidos de Markov e da teoria da cadeia de Markov. Foram descritas as classificações dos HMMs segundo dois critérios: quanto à distribuição de probabilidade (discreta ou contínua) e quanto à topologia (ergódico e esquerda-direita). Em seguida, explicou-se o funcionamento do HMM, tanto para a geração de padrões de referência (etapa de treinamento) quanto para a identificação dos padrões de teste (etapa do reconhecimento).

No capítulo seguinte, é apresentada a plataforma experimental de baixo custo construída e são discutidos os resultados dos experimentos com ambas as técnicas de aprendizado de máquina: RNA e HMM.

# 4 Plataforma e Resultados Experimentais

Neste capítulo, apresenta-se a plataforma experimental desenvolvida para a realização dos experimentos com Redes Neurais Artificiais e com Modelos Escondidos de Markov no contexto da tarefa de reconhecimento de gestos em superfícies. Neste Capítulo também são detalhados os resultados obtidos desses experimentos, algumas questões de discussões levantadas e as limitações do sistema proposto.

## 4.1 Plataforma Experimental

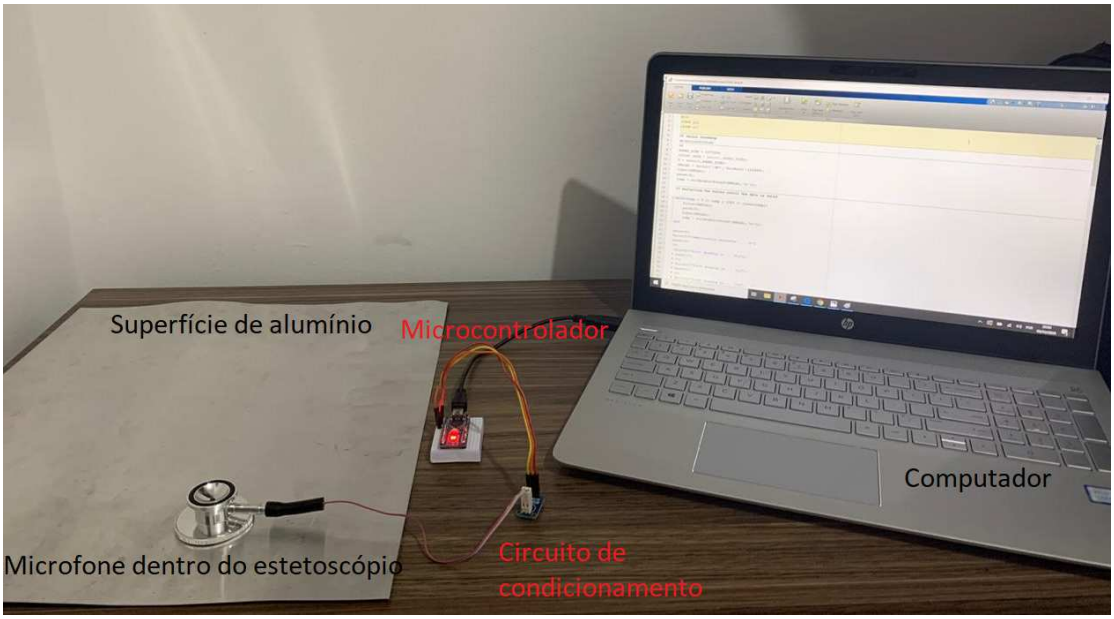
A plataforma experimental, apresentada na Figura 19, foi construída com o intuito de avaliar a utilização do *hardware* e dos algoritmos de treinamento propostos. Quando uma unha arranha uma superfície, resulta em uma série de vibrações mecânicas, que se propagam pela superfície e são capturadas pelo microfone conectado à cânula do estetoscópio. O uso de estetoscópios para aquisição de sons fornece naturalmente um alto nível de supressão de ruído ambiental. Isso permite que os impactos sejam prontamente segmentados de qualquer ruído de fundo com um limiar de amplitude simples [7]. O sinal do microfone é então filtrado e amplificado por um circuito de condicionamento e adquirido por um conversor analógico-digital de 10 bits do Arduino Nano. Em seguida, o microcontrolador envia o sinal digital via comunicação serial (UART) para o processamento da informação pelo computador, que finalmente será capaz de reconhecer o gesto realizado na superfície por um dos aplicativos de aprendizado de máquina desenvolvidos no MATLAB.

O estetoscópio é duplo lúmen e é fabricado por Ever Ready First Aid. As dimensões do microfone de mini-eletreto são de 0,16 x 0,06 polegadas e devem ser suficientemente pequenas para caber dentro da cânula do estetoscópio, como apresentado na Figura 20. O circuito de condicionamento é um amplificador de microfone de baixo custo da Adafruit MAX4466. Este circuito possui um potenciômetro para ajustar o ganho de 25x a 125x. A saída é *rail-to-rail*, tensão de saída é Vcc/2, a tensão de alimentação é de 2,4V a 5,5V e a largura de banda de 600 kHz.

O Arduino Nano V3.0 tem como principal diferencial o seu tamanho reduzido, permitindo uma flexibilidade maior para o uso dessa placa em projetos cujo tamanho seja importante. Ele usa o mesmo microcontrolador que o Arduino Uno, o ATmega328.

O Arduino Nano pode ser alimentado por uma conexão mini-B USB, por uma fonte externa não regulada de 6 a 20 volts (pino 30), ou por uma fonte externa regulada de 5V (pino 27). A fonte de alimentação selecionada automaticamente é a de maior tensão. O ATmega328 possui 32KB de memória flash para armazenamento de código (dos quais

Figura 19 – Fotografia da plataforma experimental.




Fonte: Elaborado pela autora.

Tabela 2 – Custo da Plataforma Experimental.

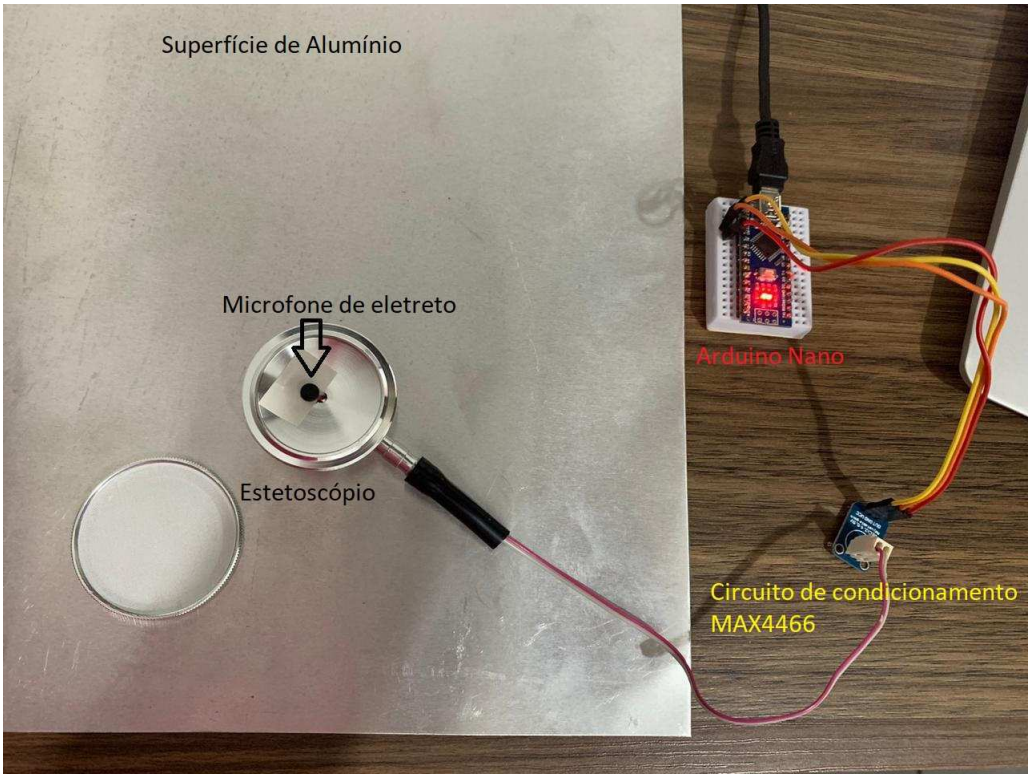
| Material | Preço ($) |
|---|---|
| Mini Microfone Eletreto | 0,71 |
| Adafruit MAX4466 | 8,99 |
| Arduino Nano | 7,65 |
| Estetoscópio Ever Ready First Aid | 5,84 |

2KB são usados pelo *bootloader*) e 2KB de SRAM e 1KB de EEPROM (que podem ser lidos ou escritos com a biblioteca EEPROM). Quatorze pinos digitais no Nano podem ser usados como uma entrada ou uma saída. Eles operam a 5 volts. Cada pino pode fornecer ou receber um máximo de 40 mA e possui um resistor interno (desconectado por padrão) de 20-50K. Em adição alguns pinos possuem funções especializadas: tais como serial, interruptores externos, PWM e SPI. O Nano tem também 8 entradas analógicas, cada uma das quais com 10 bits de resolução. Por padrão elas medem de 0 a 5 volts. Além disso, alguns pinos têm funcionalidades especializadas: I2C e *reset*. Na Tabela 2 é apresentado o custo, em dólar, dos materiais utilizados na plataforma experimental apresentada.

Definiu-se para o alfabeto inicial três figuras geométricas: círculo, quadrado e triângulo equilátero. Portanto, no início da aplicação, o usuário é notificado para começar a desenhar um desses gestos.

O som se propaga através de materiais sólidos e líquidos com muito mais eficiência do que através do ar. Assim, enquanto o atrito da unha em uma superfície produz apenas um ruído audível e suave, o sinal é propagado consideravelmente melhor através do material sólido. Essa propagação de som superior significa que um sinal não é apenas propagado

Figura 20 – Fotografia da plataforma experimental, com ênfase nos dispositivos.




Fonte: Elaborado pela autora.

a uma distância maior, mas também é melhor preservado [23]. Observando a Tabela 3, percebe-se que a velocidade de propagação do som no alumínio é uma das mais altas entre os materiais comuns, em torno de 6300 m/s. Dessa forma, uma superfície de alumínio foi escolhida para realizar os experimentos que foram realizadas em um nível máximo de ruído ambiente de 45 dB.

Tabela 3 – Velocidade de propagação do som em alguns materiais.

| Material | Temperatura | Velocidade (m/s) |
|---|---|---|
| Ar (1 atm) | 35°C | 340 |
| Água | 8°C | 1435 |
| Alumínio | - | 6300 |
| Ferro | - | 5130 |
| Cobre | - | 4600 |
| Vidro | - | 4540 |
| Ouro | - | 3240 |

Na próxima seção, analisa-se o conjunto de dados mais adequados para treinar os modelos, as caixas de ferramentas do MATLAB utilizadas e o índice de sucesso de reconhecimento nos experimentos usando as técnicas de aprendizado de máquina: Redes Neurais Artificiais e Modelos Escondidos de Markov.

## 4.2 Resultados Experimentais

Dispondo apenas de um *hardware* simples e considerando um baixo custo computacional para o algoritmo de aprendizado de máquina, deseja-se alcançar uma alta taxa de sucesso de reconhecimento com um conjunto de dados pequeno para treinamento e modelagem (máximo de 10 desenhos por gesto) e um tempo de treinamento curto (máximo de 2 minutos por gesto), para que o usuário não fique fatigado na etapa de treinamento, um problema comum em outro trabalhos, como em TapSense [24] que coleta mais de 160 desenhos, por gesto e em Touch and Activate [48] que coleta 5 gestos em 12 rodadas, totalizando 60 gestos por participante do treinamento, e também em WritePad [9] que coleta 200 desenhos por digito (0-9), por voluntário de treinamento.

As próximas sessões apresentam os experimentos realizados, utilizando as técnicas de aprendizado de máquina, com Redes Neurais e Modelos Escondidos de Markov, analisando o conjunto de dados utilizados para alimentar os modelos e as suas características.

### 4.2.1 Experimento com Redes Neurais Artificiais

Encontrar uma configuração ideal para redes neurais é um desafio, pois muitos parâmetros precisam ser considerados. Determinar o número ideal de camadas e neurônios escondidos para todos os classificadores é um aspecto crucial. Se a complexidade das redes for muito baixa para modelar as características do conjunto de dados, as taxas de erro serão maiores. Se a complexidade for muito alta, longos períodos de treinamento e tempos de *recall* são a consequência [64].

Assim, para treinar a RNA, foram coletadas 10 desenhos de cada gesto (círculo, quadrado e triângulo), dos quais 7 foram usados para treinamento e 3 para validação. Em seguida, todos os dados foram concatenados para serem usados como entrada na caixa de ferramentas *Neural Net Pattern Recognition Tool*, do MATLAB. A configuração oferecida por esta caixa de ferramentas é uma rede alimentada adiante de duas camadas, com neurônios sigmóides ocultos e *soft-max* de saída. A rede é treinada com retropropagação de gradiente conjugado em escala. Nesta caixa de ferramentas, pode-se definir o número de neurônios na camada oculta. Com 10 neurônios, obteve-se o melhor desempenho, visto que após a adição de um certo número de neurônios ocultos, para nós, mais de 10, o modelo começa a ajustar os dados, viciando a rede e fornecendo estimativas ruins no teste. Portanto, com 10 neurônios, tem-se a configuração que minimiza a entropia cruzada, estando sempre atento com relação ao ajuste excessivo.

#### 4.2.1.1 Conjunto de dados para treinamento da Rede Neural

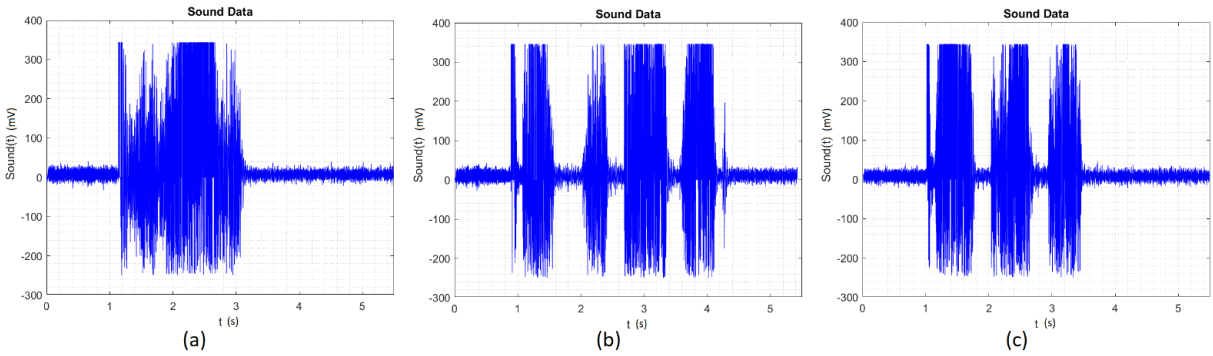
Conseguir uma alta generalização em uma rede neural é desafiador devido às diferenças observadas nos estilos de desenhos no conjunto de dados. A complexidade da

topologia ideal é possivelmente muito alta para ser suficientemente treinada, considerando o conjunto ilimitado de amostras de treinamento. Por esse motivo, processou-se o sinal apresentado para treinar a rede neural.

### Sinal puro no tempo

O primeiro conjunto de dados que foi utilizado para treinar a RNA foi o sinal de tempo puro, conforme apresentado na Figura 21. Pode-se observar claramente a diferença entre o quadrado (Figura 21 (b)) e o triângulo (Figura 21 (c)) devido à diferença no número de picos relacionados às paradas nos vértices do desenho. Para a RNA, no entanto, todos esses picos que representam o gesto na superfície tornam a classificação da RNA mais complexa, porque cada gesto terá um número e posição completamente diferentes de picos e ranhuras, sendo difícil para a RNA fazer uma generalização.

Figura 21 – (a) Sinal puro no tempo para um círculo; (b) Sinal puro no tempo para um quadrado; (c) Sinal puro no tempo para um triângulo.



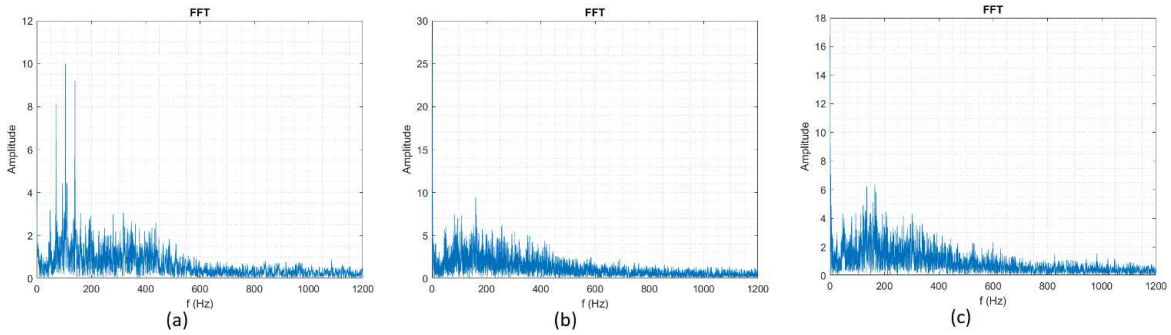

Fonte: Elaborado pela autora.

Outro fator complicador para a rede neural é que cada pessoa pode iniciar o gesto em um momento diferente, dificultando para RNA analisar esse primeiro pico e classificar a figura corretamente, pois cada gesto inicia em um instante de tempo completamente diverso. Portanto, com esse conjunto de dados, obteve-se uma taxa de 33% de sucesso.

### Sinal puro na frequência

O segundo conjunto de dados treinou a RNA usando o sinal na frequência, a FFT apresentada na Figura 22, mas foi ainda mais complexo para a RNA fazer o reconhecimento do gesto e obteve-se uma taxa de sucesso de 15%.

Figura 22 – (a) Sinal na frequência para um círculo; (b) Sinal na frequência para um quadrado; (c) Sinal na frequência para um triângulo.

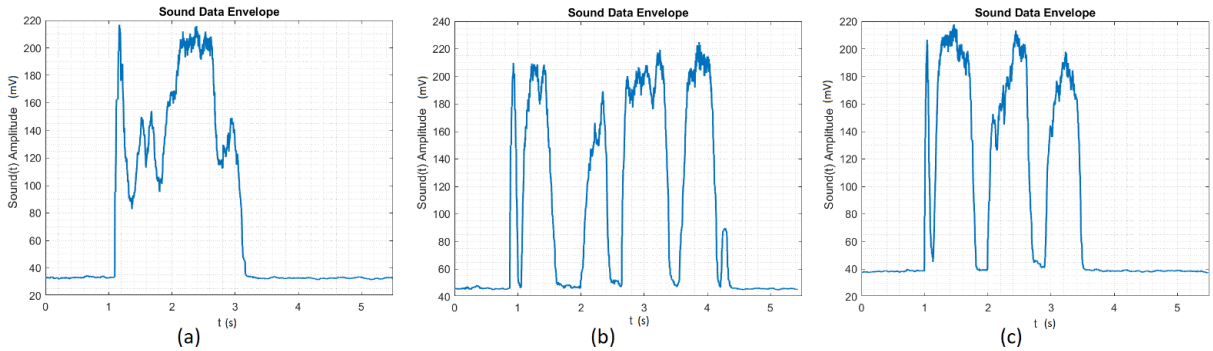



Fonte: Elaborado pela autora.

### Sinal envelopado no tempo

Como o desempenho do sinal de FFT foi pior que o do sinal no tempo puro no conjunto de treinamento da RNA, decidiu-se processar o sinal no tempo puro. Utilizou-se a função do MATLAB de envelope que retorna a parte superior e inferior da raiz quadrada média do sinal no tempo. O envelope é determinado usando uma janela deslizante de 200 amostras, definida pelo autor, porque, coletando mais amostras, não se generalizaria o sinal suficiente para a RNA, e se coletando menos amostras, não se preservaria a forma original do sinal. Como apresentado na Figura 23, a saída da função envelope é maior ou igual a zero, pois retorna o envelope da raiz quadrada média do sinal de tempo. A janela deslizante de 200 amostras suaviza os picos e preserva a forma individual. Portanto, será mais genérico para a RNA reconhecer o gesto, mas ainda tem-se o problema de quando o gesto iniciou. Para este conjunto de dados, obteve-se uma taxa de sucesso de 65 %.

Figura 23 – (a) Sinal no tempo envelopado para um círculo; (b) Sinal no tempo envelopado para um quadrado; (c) Sinal no tempo envelopado para um triângulo.

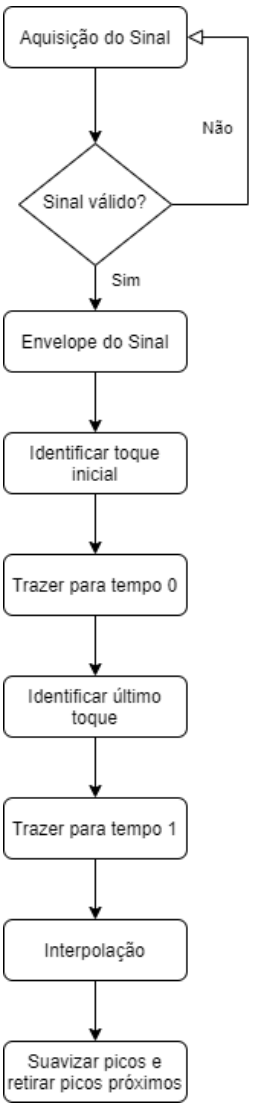



Fonte: Elaborado pela autora.

### Aprimoramento no sinal envelopado no tempo

A tentativa final do conjunto de dados de treinar a RNA foi a melhoria do sinal de tempo envelopado, com a utilização do algoritmo desenvolvido neste trabalho e apresentado na Figura 24. Neste conjunto de dados, primeiro procura-se esse pico inicial, considerando que tem-se um pico toda vez que o dedo toca a superfície, e traz essa amostra para o tempo 0 (zero) e o último pico é levado para o tempo 1, com 100 amostras entre eles. Dessa forma, não é mais relevante o instante que o usuário começa a desenhar. A segunda melhoria foi a velocidade do desenho, já que os usuários têm ritmos diferentes, utilizou-se o recurso de interpolação para manter a proporção dos dados e, em seguida, todos os conjuntos de dados tinham o mesmo comprimento de 100 amostras. É importante relembrar que o sinal apresentado para esse algorítimo já foi adquirido e envelopado, como apresentado na Seção anterior: Sinal envelopado no tempo.

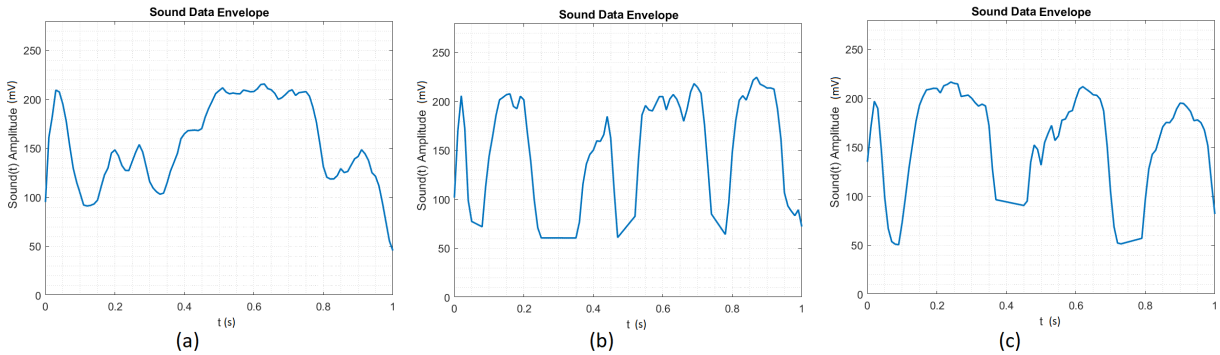
Figura 24 – Algorítimo proposto para problema de início e velocidade do desenho.




Fonte: Elaborado pela autora.

Além disso, com a função *findpeaks* do Matlab e as opções *MinPeakHeight* e *MinPeakDistance*, pode-se encontrar picos com amplitude mínima e ignorar picos muito próximos um do outro. Portanto, suavizou-se os picos e preservou-se a forma inicial do sinal. Todas essas melhorias tornaram mais simples a tarefa de reconhecimento da RNA, com os sinais aprimorados apresentados na Figura 25. Portanto, para esse conjunto de dados, obteve-se a maior precisão de 90 % de sucesso.

Figura 25 – (a) Sinal no tempo envelopado e aprimorado para um círculo; (b) Sinal no tempo envelopado e aprimorado para um quadrado; (c) Sinal no tempo envelopado e aprimorado para um triângulo.




Fonte: Elaborado pela autora.

### 4.2.2 Experimento com Modelos Escondidos de Markov

Desde a introdução dos Modelos Escondidos de Markov no processamento de fala em meados da década de 1970, com os trabalhos como o Dragon System [4] na Universidade Carnegie Mellon e o esforço de longa data da IBM em um sistema de ditado por voz [3], a técnica de reconhecimento de fala contínua por HMM vem avançando.

Levando em consideração que o sinal acústico da fala possui características semelhantes ao sinal acústico produzido quando uma unha é arrastado sobre uma superfície, trabalhos anteriores sobre reconhecimento de fala utilizando HMMs foram utilizados como base e inspiração para este experimento. Vale ressaltar que a estrutura do sinal acústico é semelhante, mas a complexidade do sinal arrastado por uma unha comparado ao sinal da fala é inferior, visto que não faz-se necessário procurar por uma gama de fonemas ou ter uma grande base de dados para treinar o modelo. Como também os sistemas de reconhecimento de fala ainda podem ser classificados como dependente do locutor e do contexto.

Nos últimos anos, o HMM é uma das técnicas de modelagem mais populares e eficazes para séries temporais acústicas [37]. As razões pelas quais esse método se tornou tão popular são a estrutura estatística (matematicamente precisa) inerente; a facilidade e disponibilidade de algoritmos de treinamento para estimar os parâmetros dos modelos a partir de conjuntos finitos de treinamento de dados de fala; a flexibilidade do sistema

de reconhecimento resultante no qual é possível alterar facilmente o tamanho, tipo ou arquitetura dos modelos para se adequar a palavras, sons e assim por diante; e a facilidade de implementação do sistema de reconhecimento geral [31].

Como o sinal acústico tem estrutura temporal e pode ser codificada como uma sequência de vetores espectrais que abrangem a faixa de frequência de áudio, o modelo escondido de Markov fornece uma estrutura natural para a construção de tais modelos [5].

A força teórica básica do HMM é que ele combina a modelagem de processos estocásticos estacionários (para os espectros de curto prazo) e a relação temporal entre os processos (via cadeia de Markov) juntos em um espaço de probabilidade bem definido. Essa combinação permite estudar esses dois aspectos separados da modelagem de um processo dinâmico (como a fala) usando uma estrutura consistente [31].

A escolha da configuração topológica e o número de estados no modelo geralmente refletem o conhecimento a priori da fonte de som específica a ser modelada e não está relacionada à tratabilidade matemática ou considerações de implementação [31]. De acordo com Deller et al. [14], os estados do HMM frequentemente representam fonemas acústicos identificáveis no reconhecimento de fala. O número de estados é geralmente escolhido para corresponder aproximadamente ao número esperado de fonemas nas frases. No entanto, o número ideal de estados é melhor determinado através de experimentos, pois a relação do número de estados com o desempenho do HMM é muito imprecisa.

Luigi Rosa [58] propôs um algoritmo rápido e confiável para reconhecimento de fala, baseado nos modelos escondidos de Markov. A caixa de ferramentas do MATLAB implementada por Luigi Rosa [58] possui uma frequência de amostragem de 22050 bits/s.

A estrutura topológica da HMM implementada é do tipo esquerda-direita, com três estados. Essas configurações foram pré-escolhidas pela caixa de ferramentas [58] por apresentar bons resultados no reconhecimento de gestos em superfícies. A topologia esquerda-direita foi escolhida uma vez que o gesto começa e termina em instantes de tempo bem identificados e o comportamento sequencial do movimento é bem representado por um HMM sequencial.

A modelagem de um HMM é realizada em três etapas: treinamento, reconhecimento e decodificação.

Desenvolveu-se um aplicativo no MATLAB para dar suporte à coleta de dados por gestos e ao treinamento do modelo de Markov. Para cada gesto, o aplicativo solicita que o usuário comece a desenhar e grava o sinal do microfone por um intervalo de três segundos e o salva com extensão *.wav* pela função MATLAB da *audiowrite*. Assim, a forma de onda do áudio de entrada do microfone é convertida em uma sequência de vetores acústicos de tamanho fixo. Quando a gravação é concluída, o sistema pergunta ao usuário se deve reter a amostra específica. Depois que o aplicativo coleta cinco desenhos para um gesto

específico, ele repete o mesmo processo de coleta de dados para o próximo gesto.

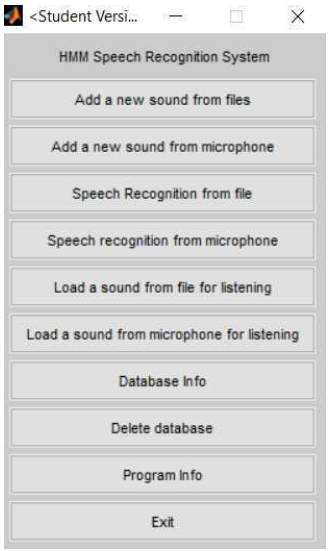
A etapa de treinamento tem o objetivo de determinar os parâmetros do modelo, como também determinar o melhor conjunto de dados que alimentará o modelo a ser treinado, representando com maior eficiência o sinal que está sendo modelado, de forma que maximize a probabilidade de geração da observação. O método utilizado para o treinamento do HMM é o algoritmo *Forward-Backward*, também conhecido como o algoritmo de re-estimação de Baum-Welch.

Depois de coletar dados para todos os três gestos (círculo, quadrado e triângulo), o aplicativo gera modelos escondidos de Markov via uma caixa de ferramentas MATLAB desenvolvida por uma equipe de Luigi Rosa [58]. Em seguida, a etapa de reconhecimento utiliza o algoritmo *Forward*, que consiste em determinar qual o modelo, dentre os vários obtidos na etapa de treinamento, que provavelmente gerou uma dada sequência de observação.

E finalmente a etapa de decodificação que opera pesquisando todas as sequências de sons possíveis usando a remoção para remover hipóteses improváveis, mantendo assim a pesquisa tratável. Assim, determina-se a sequência de estados mais provável de ter produzido uma determinada sequência de entrada. O algoritmo de Viterbi é utilizado nesta etapa para solucionar o problema.

A Figura 26 apresenta a interface da caixa de ferramentas desenvolvida por Luigi Rosa no MATLAB. Para treinar o modelo, utiliza-se a aba *Add a new sound from files* (apenas aceita extensões de arquivo do tipo .wav) e para reconhecer e decodificar um novo gesto, utiliza-se a aba *Speech Recognition from file*.

Figura 26 – Imagem da interface da caixa de ferramentas desenvolvida por Luigi Rosa no MATLAB.




Fonte: [58].

A primeira tentativa de conjunto de dados para treinar o modelo escondido de

Markov foi o sinal puro no tempo, assim como para a RNA. Adicionou-se cinco desenhos para cada gesto coletadas na etapa de treinamento e, em seguida, usando a opção *Speech Recognition from file* da caixa de ferramentas [58], carregou-se um novo desenho a ser reconhecido, obtendo uma taxa de reconhecimento de 90%. O erro ocorreu no reconhecimento do quadrado e do triângulo, mas para o círculo, obtive-se um reconhecimento bem-sucedido de 100%.

## 4.3 Discussão e Limitações do Sistema

Considerando a revisão de literatura e os resultados experimentais apresentados neste trabalho, são abordadas nessa Seção as questões que levantaram discussões e as limitações da plataforma apresentada.

### 4.3.1 Discussão sobre as ferramentas utilizadas e escopo do trabalho

Este trabalho não tem a intenção de desenvolver as caixas de ferramentas que treinarão e implementarão a rede neural ou o modelo escondido de Markov, mas visa implementar o *hardware* que adquire o sinal, e aprimorar e testar os conjuntos de dados que alimentarão os modelos. Por isso, utilizou-se a caixa de ferramentas de reconhecimento de padrões de rede neural do MATLAB para treinar a rede com a retropropagação em gradiente conjugado em escala. Nesta caixa de ferramentas, configura-se os parâmetros de rede, como o número de neurônios na camada oculta, seleciona-se os desenhos para validação e teste, e é possível avaliar o desempenho da RNA utilizando matrizes de entropia cruzada. Se o desempenho da rede não for bom o suficiente, é possível treiná-la novamente, ajustando o tamanho da rede ou até importando um novo conjunto de dados para melhorar o desempenho da rede.

Para o modelo escondido de Markov, pesquisou-se algumas caixas de ferramentas para reconhecimento de fala, como o Georgia Tech Gesture Toolkit (GT2k) [66] e o gpdsHMM [12]. Mas alguns problemas foram encontrados no funcionamento e não encontrou-se suporte necessário. Então, após coletar dados para todos os gestos, o aplicativo gera um Modelo Escondido de Markov via Reconhecimento de Fala usando Modelos de Markov escondidos de Luigi Rosa [58]. A limitação dessa caixa de ferramentas é a necessidade de uma versão do MATLAB anterior à 7.5 (R2007b). Contudo, diferentemente da caixa de ferramentas da RNA que foi utilizada, nesta só é possível fornecer os dados para treinar o modelo e não pode-se escolher nenhum outro parâmetro do modelo. Portanto, a caixa de ferramentas da RNA é mais configurável, oferece ao usuário mais liberdade para escolher os parâmetros e também fornece gráficos que permitem avaliar o desempenho da rede.

### 4.3.2 Discussão sobre as características do sinal acústico ao utilizar uma técnica de aprendizado de máquina para reconhecer padrões acústicos

Uma questão central ao desenvolver sistemas utilizando aprendizado de máquina é a quantidade de dados necessários para o treinamento para um alto nível de precisão. Consequentemente, a escolha do conjunto de dados e o número de amostras para cada técnica de aprendizado de máquina é uma etapa importante do processo. Para a RNA, tentou-se o sinal puro no tempo e, em seguida, o sinal na frequência. Este último teve o pior desempenho. Em seguida, processou-se o sinal puro por meio do envelope e a interpolação, para tornar os picos mais suaves. Após todas essas tentativas, alcançou-se uma taxa de sucesso de 90% por meio de um conjunto de 10 desenhos, dos quais 7 foram usados para treinamento e 3 para validação. Já para o modelo escondido de Markov, treinando o modelo com o sinal puro no tempo e apenas 5 desenhos de cada gesto, alcançou-se uma taxa de reconhecimento de 90%. Para algumas amostras, HMM confundiu o quadrado com o triângulo e vice-versa, mas com o círculo o reconhecimento foi de 100% de sucesso. É importante ressaltar que, com um conjunto de dados menor, menor tempo de treinamento e menor custo computacional, o modelo escondido de Markov alcançou um resultado melhor do que a rede neural.

### 4.3.3 Discussão sobre qual é a técnica de aprendizado de máquina mais adequada para aplicações de reconhecimento de gestos em superfícies

Um artigo anterior que inspirou este trabalho foi o *Ipanel* [10], por possuir a mesma problemática de rastrear os sinais acústicos gerados pelo deslizamento dos dedos em uma superfície. *Ipanel* [10] alcançou um desempenho robusto contra diferentes níveis de ruído ambiente. Mas *Ipanel* [10] transformou os recursos em imagens e depois empregou CNN para reconhecer o movimento dos dedos sobre a mesa. Também tentou-se desenvolver uma CNN neste trabalho, com o *Matlab Deep Learning Toolbox*, mas para treinar uma rede do zero, é necessário que o arquiteto defina o número de camadas e filtros, além de outros parâmetros ajustáveis. O treinamento de um modelo preciso a partir do zero também exige grandes quantidades de dados, da ordem de milhões de amostras [63].

Redes neurais profundas com uma quantidade maior de dados de treinamento e diferentes topologias podem ser observadas para obter altas taxas de reconhecimento, bem como o impacto de diferentes superfícies ou ruídos de fundo [60]. No entanto, neste trabalho, busca-se uma solução simples e de baixo custo computacional. A abordagem da CNN ou qualquer rede neural profunda acaba sendo muito mais trabalhosa devido à necessidade desse grande conjunto de dados e ao tempo mais longo para treinar a rede.

Da mesma forma, artigos científicos como Skinput [25] e TapSense [24] inspiraram este trabalho. Eles utilizam a técnica SVM para a classificação, fornecida no kit de

ferramentas de aprendizado de máquina Weka. No entanto, como pode-se observar nesses trabalhos, para treinar o modelo, eles precisam coletar mais de 160 conjunto de dados, por gesto. SVM também possui alguns pontos fracos quando aplicado a tarefas com baixos requisitos de esforço computacional devido à grande necessidade de memória e longo tempo de computação desse algoritmo de treinamento [42]. Dessa forma, a necessidade de centenas de amostras para treinar o modelo e o alto custo computacional não atenderam ao requisito de pequeno conjunto de dados (máximo de 10 desenhos por gesto) e curto tempo de treinamento (máximo de 2 minutos por gesto) deste trabalho. Em razão dessas limitações, a técnica de SVM não foi implementada e outras mais apropriadas para os requisitos deste trabalho, como HMM, foram aplicadas.

Considerando as taxas de sucesso das duas técnicas de aprendizado de máquina apresentadas neste Capítulo, a maior taxa de sucesso foi alcançada por meio do modelo escondido de Markov, indicando que a solução proposta com este classificador é mais simples de implementar e treinar do que a técnica da RNA e é suficiente para resolução do problema. Portanto, com um conjunto de dados menor, menor tempo para treinamento e menor custo computacional, o HMM obteve melhor resultado do que as redes neurais.

### 4.3.4 Discussão e análise para embarcar no Arduino o modelo escondido de Markov treinado

Depois de treinar o modelo escondido de Markov, avaliou-se embarcar o código gerado a partir do modelo treinado pelas caixas de ferramentas descritas previamente na Seção 4.2.2, no *hardware* de baixo custo já utilizado na plataforma para fazer a conversão analógico-digital: o Arduino Nano, cujo microcontrolador é o ATmega328. Dessa forma, após o treinamento, o sistema ficaria independente do *software* MATLAB, que possui um custo mais elevado.

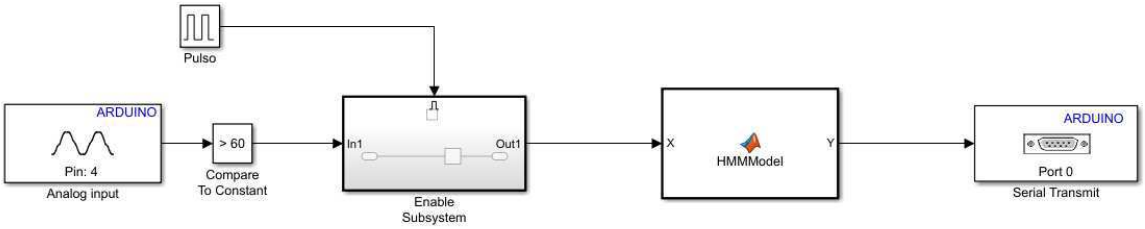
Primeiramente, instalou-se o pacote de suporte no *Simulink* para o *hardware* do Arduino. Em seguida, criou-se o modelo apresentado na Figura 27. Nesse modelo, inicialmente faz-se a leitura da entrada analógica (porta 4) do Arduino Nano, e se o valor dessa leitura for maior que 60, o que significa que alguém está desenhando na superfície, o sistema fica executando por 3 segundos o bloco *enable subsystem* que possui um bloco de memória acoplado, com a finalidade de salvar os dados adquiridos nesse intervalo de tempo (tempo de aquisição do gesto). Esse dado é então utilizado para alimentar o modelo escondido de Markov que foi previamente elaborado e está carregado por meio da função do MATLAB *HMMModel*, para finalmente, na saída serial, ser impresso o resultado do reconhecimento do gesto (círculo, triângulo ou quadrado).

Contudo, para o modelo escondido de Markov, esta implementação não obteve êxito, visto que para realizar o reconhecimento do gesto faz-se necessário carregar, no Arduino

Nano, 25 arquivos do MATLAB, que excedem 50KB, com uma sequência específica e relações de entrada e saída únicas, gerados pela caixa de ferramentas de Luigi Rosa [58].

Levando em consideração as especificações de memória do ATmega328, que possui 32KB de memória flash para armazenamento de código (dos quais 2KB são usados pelo *bootloader*) e 2KB de SRAM e 1KB de EEPROM (que podem ser lidos ou escritos com a biblioteca EEPROM), não há memória suficiente para alocar o modelo escondido de Markov gerado pelo caixa de ferramentas de Luigi Rosa [58], que foi utilizada neste projeto para a elaboração do modelo escondido de Markov e reconhecimento dos gestos.

Figura 27 – Imagem do modelo elaborado no *Simulink* para embarcar no *hardware* do Arduino o HMM desenvolvido.




Fonte: Elaborado pela autora.

### 4.3.5 Limitações

Com o objetivo de um *hardware* simples e baixo custo computacional para o algoritmo de aprendizado de máquina, obtiveram-se algumas limitações na plataforma experimental: o nível máximo de ruído ambiente é de 45 dB, a superfície é de alumínio devido à sua alta velocidade de propagação do som, e o alfabeto inicial é simples contendo figuras geométricas: círculo, quadrado e triângulo.

Pequenas variações no posicionamento do sensor causam grandes variações nas características do sinal. Assim, a posição repetida do sensor é crítica para um treinamento e reconhecimento eficazes entre sessões.

## 4.4 Considerações Finais

Neste capítulo a plataforma experimental construída foi descrita, levando em consideração o objetivo de um projeto de baixo custo. Foram detalhadas as especificações do *hardware* utilizado, como apresentado na Figura 19, com o microfone, o circuito de condicionamento do sinal e o microcontrolador, juntamente com a escolha do material da superfície, considerando a velocidade de propagação do som nos materiais, como apresentado na Tabela 3.

Em seguida, analisou-se o conjunto de dados mais adequados para treinar os modelos, as caixas de ferramentas do MATLAB utilizadas e o índice de sucessos de reconhecimento nos experimentos usando aprendizado de máquina, com Redes Neurais e Modelos escondidos de Markov.

O experimento com RNA foi bem sucedido apenas após o processamento do sinal puro no tempo, como apresentado na Figura 25. Fez-se necessário obter a envoltória do sinal, fazer uma interpolação dos dados em um intervalo de 100 amostras para cada conjunto de dados e utilizar funções que ignoravam picos muito próximos dos outros e com amplitude mínima, de forma que não é mais relevante o instante de tempo que o usuário começa a desenhar, nem a velocidade que o usuário desenha. Com essas melhorias, alcançou-se uma taxa de acerto de 90%.

Já o experimento com HMM não necessitou de processamento do sinal. Apenas com as amostras do sinal puro no tempo, o modelo com HMM alcançou uma taxa de 90% de sucesso, sendo que para o círculo obteve-se 100% de acerto. Para esta técnica utilizou-se a caixa de ferramentas desenvolvida pelo time de Luigi Rosa [58], apresentada na Figura 26.

Em seguida, questões foram discutidas, como as caixas de ferramentas utilizadas no MATLAB e suas limitações de configurações, as características do sinal acústico para treinamento dos modelos, a técnica mais adequada, com taxa de acerto mais elevada e com baixo custo computacional para a tarefa de reconhecimento de gestos em superfícies e uma análise sobre embarcar no microcontrolador o HMM treinado.

Finalmente algumas limitações do sistema proposto foram abordadas, como o nível de ruído ambiente e o posicionamento do microfone na superfície, sendo crucial a posição repetida do sensor para um treinamento e reconhecimento eficazes.

# 5 Considerações Finais

Nesta dissertação, foi realizada uma análise comparativa das técnicas de aprendizado de máquina: Redes Neurais Artificiais e Modelos Escondidos de Markov no contexto da tarefa de reconhecimento de gestos em superfícies, considerando a utilização de um *hardware* simples e uma menor demanda computacional (tempo e memória física) para o algoritmo de aprendizado de máquina poder alcançar uma alta taxa de sucesso com um conjunto de dados pequeno (máximo de 10 desenhos por gesto) e um tempo de treinamento curto (máximo de 2 minutos por gesto).

Considerando as taxas de sucesso das duas técnicas de aprendizado de máquina apresentadas no Capítulo 4, o sistema mais bem-sucedido foi alcançado por meio do modelo escondido de Markov. A solução proposta com este classificador é mais simples de implementar e de treinar do que a técnica da RNA e é suficiente para resolução do problema. Para a etapa de treinamento, com 5 desenhos por gesto e com o sinal puro no tempo, o modelo escondido de Markov alcançou uma taxa de sucesso de 90%. Portanto, o HMM com um conjunto de dados menor e menor tempo de treino obteve o melhor resultado entre os dois.

Os resultados demonstraram também a importância da representação dos dados para o processamento eficiente de informações para a RNA, visto que quanto maior o aperfeiçoamento do conjunto de dados, mais alta a taxa de sucesso alcançado.

## 5.1 Sugestões para trabalhos futuros

Como sugestão para trabalhos futuros, são colocados os seguintes pontos:

- Avaliação da utilização de vários microfones na configuração do sistema para fornecer uma precisão do local do gesto na superfície e melhorar a precisão da classificação.
- Embarcar em um *hardware* de baixo custo o modelo escondido de Markov previamente treinado, para classificação dos gestos sem computador e cálculo realizado em computação em borda.

# Referências


[1] B. Amento, W. Hill, and L. Terveen. The sound of one hand: A wrist-mounted bio-acoustic fingertip gesture interface. In *CHI'02 Extended Abstracts on Human Factors in Computing Systems*, pages 724–725. ACM, 2002. Citado 2 vezes nas páginas 6 e 11.

[2] F. Antonacci, L. Gerosa, A. Sarti, S. Tubaro, and G. Valenzise. Sound-based classification of objects using a robust fingerprinting approach. In *2007 15th European Signal Processing Conference*, pages 2321–2325. IEEE, 2007. Citado na página 9.

[3] A. Averbuch, L. Bahl, R. Bakis, P. Brown, G. Daggett, S. Das, K. Davies, S. De Gennaro, P. De Souza, E. Epstein, et al. Experiments with the tangora 20,000 word speech recognizer. In *ICASSP'87. IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 12, pages 701–704. IEEE, 1987. Citado na página 48.

[4] J. Baker. The dragon system–an overview. *IEEE Transactions on Acoustics, speech, and signal Processing*, 23(1):24–29, 1975. Citado 2 vezes nas páginas 29 e 48.

[5] L. E. Baum and J. A. Eagon. An inequality with applications to statistical estimation for probabilistic functions of markov processes and to a model for ecology. *Bulletin of the American Mathematical Society*, 73(3):360–363, 1967. Citado 2 vezes nas páginas 3 e 49.

[6] L. E. Baum, T. Petrie, G. Soules, and N. Weiss. A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains. *The annals of mathematical statistics*, 41(1):164–171, 1970. Citado na página 35.

[7] A. Braun, S. Krepp, and A. Kuijper. Acoustic tracking of hand activities on surfaces. In *Proceedings of the 2nd international Workshop on Sensor-based Activity Recognition and Interaction*, page 9. ACM, 2015. Citado 3 vezes nas páginas 2, 9 e 41.

[8] E. I. Bueno. *Group Method of Data Handling (GMDH) e Redes Neurais na Monitoraçao e Detecçao de Falhas em sensores de centrais nucleares*. PhD thesis, Universidade de São Paulo, 2011. Citado 3 vezes nas páginas 21, 22 e 25.

[9] M. Chen, P. Yang, S. Cao, M. Zhang, and P. Li. Writepad: Consecutive number writing on your hand with smart acoustic sensing. *IEEE Access*, 6:77240–77249, 2018. Citado 2 vezes nas páginas 11 e 44.

[10] M. Chen, P. Yang, J. Xiong, M. Zhang, Y. Lee, C. Xiang, and C. Tian. Your table can be an input panel: Acoustic-based device-free interaction recognition. *Proceedings*

*of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 3(1):1–21, 2019. Citado 2 vezes nas páginas 10 e 52.

[11] T.-R. Chou and J.-C. Lo. Research on tangible acoustic interface and its applications. In *Proceedings of the 2nd International Conference on Computer Science and Electronics Engineering*. Atlantis Press, 2013. Citado na página 1.

[12] S. David, M. A. Ferrer, C. M. Travieso, J. B. Alonso, and D. D. S. y Comunicaciones. gpdshmm: A hidden markov model toolbox in the matlab environment. *CSIMTA, Complex Systems Intelligence and Modern Technological Applications*, pages 476–479, 2004. Citado na página 51.

[13] A. de Pádua Braga, A. C. P. de Leon Ferreira, and T. B. Ludermir. *Redes neurais artificiais: teoria e aplicações*. LTC Editora Rio de Janeiro, Brazil:, 2007. Citado 3 vezes nas páginas 18, 26 e 27.

[14] J. Deller, J. Proakis, and J. Hansen. Discretetime processing of speech signals,"mcmillan publish, 1993. Citado 2 vezes nas páginas 28 e 49.

[15] H. Demuth, M. Beale, and M. Hagan. Neural network toolbox user's guide–version 4–for use with matlab®. *The Math Works Inc*, 2000. Citado na página 28.

[16] T. Deyle, S. Palinko, E. S. Poole, and T. Starner. Hambone: A bio-acoustic gesture interface. In *2007 11th IEEE International Symposium on Wearable Computers*, pages 3–10. IEEE, 2007. Citado na página 10.

[17] P. Dietz and D. Leigh. Diamondtouch: a multi-user touch technology. In *Proceedings of the 14th annual ACM symposium on User interface software and technology*, pages 219–226. ACM, 2001. Citado 3 vezes nas páginas 6, 15 e 16.

[18] A. P. Engelbrecht. *Computational intelligence: an introduction*. John Wiley & Sons, 2007. Citado 3 vezes nas páginas 18, 23 e 24.

[19] J. M. Fechine et al. Reconhecimento automático de identidade vocal utilizando modelagem híbrida: paramétrica e estatística. 2000. Citado na página 38.

[20] E. Ferneda. Redes neurais e sua aplicação em sistemas de recuperação de informação. *Ciência da Informação*, 35(1), 2006. Citado 2 vezes nas páginas 23 e 24.

[21] D. V. Fiorin, F. R. Martins, N. J. Schuch, and E. Pereira. Aplicações de redes neurais e previsões de disponibilidade de recursos energéticos solares. *Revista Brasileira de Ensino de Física*, 33(1):1309, 2011. Citado 2 vezes nas páginas 23 e 26.

[22] K. E. Friedl, A. R. Voelker, A. Peer, and C. Eliasmith. Human-inspired neurorobotic system for classifying surface textures by touch. *IEEE Robotics and Automation Letters*, 1(1):516–523, 2016. Citado na página 14.

[23] C. Harrison and S. E. Hudson. Scratch input: creating large, inexpensive, unpowered and mobile finger input surfaces. In *Proceedings of the 21st annual ACM symposium on User interface software and technology*, pages 205–208. ACM, 2008. Citado 4 vezes nas páginas 2, 7, 8 e 43.

[24] C. Harrison, J. Schwarz, and S. E. Hudson. Tapsense: enhancing finger interaction on touch surfaces. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, pages 627–636. ACM, 2011. Citado 5 vezes nas páginas 6, 2, 8, 44 e 52.

[25] C. Harrison, D. Tan, and D. Morris. Skinput: appropriating the body as an input surface. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 453–462. ACM, 2010. Citado 5 vezes nas páginas 6, 2, 10, 11 e 52.

[26] C. Harrison, R. Xiao, and S. Hudson. Acoustic barcodes: passive, durable and inexpensive notched identification tags. In *Proceedings of the 25th annual ACM symposium on User interface software and technology*, pages 563–568. ACM, 2012. Citado 4 vezes nas páginas 6, 2, 7 e 8.

[27] S. S. Haykin and K. Elektroingenieur. *Neural networks and learning machines*, volume 3. Pearson education Upper Saddle River, 2009. Citado 9 vezes nas páginas 18, 19, 20, 22, 23, 24, 25, 26 e 27.

[28] J. J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the national academy of sciences*, 79(8):2554–2558, 1982. Citado na página 18.

[29] H. Ishii, C. Wisneski, J. Orbanes, B. Chun, and J. Paradiso. Pingpongplus: design of an athletic-tangible interface for computer-supported cooperative play. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, pages 394–401. ACM, 1999. Citado na página 15.

[30] F. Jelinek, L. Bahl, and R. Mercer. Design of a linguistic statistical decoder for the recognition of continuous speech. *IEEE Transactions on Information Theory*, 21(3):250–256, 1975. Citado na página 29.

[31] B. H. Juang and L. R. Rabiner. Hidden markov models for speech recognition. *Technometrics*, 33(3):251–272, 1991. Citado na página 49.

[32] E. R. Kandel, J. H. Schwartz, T. M. Jessell, D. of Biochemistry, M. B. T. Jessell, S. Siegelbaum, and A. Hudspeth. *Principles of neural science*, volume 4. McGraw-hill New York, 2000. Citado na página 27.

[33] N. K. Kasabov. Foundations of neural networks, fuzzy systems, and knowledge engineering. 1996. Citado 3 vezes nas páginas 4, 19 e 39.

[34] R. Kawakatsu and S. Hirai. Rubbinput: An interaction technique for wet environments utilizing squeak sounds caused by finger-rubbing. In *2018 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, pages 512–517. IEEE, 2018. Citado na página 9.

[35] B. G. Kermani, S. S. Schiffman, and H. T. Nagle. Performance of the levenberg–marquardt neural network training method in electronic nose applications. *Sensors and Actuators B: Chemical*, 110(1):13–22, 2005. Citado na página 27.

[36] T. Kohonen. Correlation matrix memories. *IEEE transactions on computers*, 100(4):353–359, 1972. Citado na página 19.

[37] J. Koolwaaji. ispeak- consultancy in speech technology. In *Fundamentals of HMM Based Speaker Verification, year=2001,*. Citado na página 48.

[38] R. Lawrence. *Fundamentals of speech recognition*. Pearson Education India, 2008. Citado 2 vezes nas páginas 28 e 33.

[39] S. E. Levinson, L. R. Rabiner, and M. M. Sondhi. An introduction to the application of the theory of probabilistic functions of a markov process to automatic speech recognition. *Bell System Technical Journal*, 62(4):1035–1074, 1983. Citado na página 33.

[40] L. Liu, J. Chen, and L. Xu. Realization and application research of bp neural network based on matlab. In *2008 International Seminar on Future Biomedical Information Engineering*, pages 130–133. IEEE, 2008. Citado na página 27.

[41] C. Loesch and S. T. Sari. *Redes neurais artificiais: fundamentos e modelos*. Ed. da FURB, 1996. Citado 3 vezes nas páginas 4, 19 e 39.

[42] C. Lopes and F. Perdigão. Event detection by hmm, svm and ann: a comparative study. In *International Conference on Computational Processing of the Portuguese Language*, pages 1–10. Springer, 2008. Citado na página 53.

[43] P. Lopes, R. Jota, and J. A. Jorge. Augmenting touch interaction through acoustic sensing. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*, pages 53–56. ACM, 2011. Citado 3 vezes nas páginas 1, 8 e 9.

[44] G. Luo, M. Chen, P. Li, M. Zhang, and P. Yang. Soundwrite ii: Ambient acoustic sensing for noise tolerant device-free gesture recognition. In *2017 IEEE 23rd International Conference on Parallel and Distributed Systems (ICPADS)*, pages 121–126. IEEE, 2017. Citado na página 12.

[45] E. Matson, B. Yang, A. Smith, E. Dietz, and J. Gallagher. Uav detection system with multiple acoustic nodes using machine learning models. In *2019 Third IEEE*

*International Conference on Robotic Computing (IRC)*, pages 493–498. IEEE, 2019. Citado na página 16.

[46] N. A. B. Monteiro. *Desenvolvimento de Sensores Virtuais para Monitoramento de Processos Não Lineares Multivariáveis Utilizando Redes Neurais*. PhD thesis, Universidade Federal de Campina Grande, 2018. Citado 3 vezes nas páginas 19, 20 e 25.

[47] R. Murray-Smith, J. Williamson, S. Hughes, and T. Quaade. Stane: synthesized surfaces for tactile input. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1299–1302. ACM, 2008. Citado na página 12.

[48] M. Ono, B. Shizuki, and J. Tanaka. Touch & activate: adding interactivity to existing objects using active acoustic sensing. In *Proceedings of the 26th annual ACM symposium on User interface software and technology*, pages 31–40. ACM, 2013. Citado 2 vezes nas páginas 13 e 44.

[49] J. A. Paradiso, C. K. Leo, N. Checka, and K. Hsiao. Passive acoustic sensing for tracking knocks atop large interactive displays. In *SENSORS, 2002 IEEE*, volume 1, pages 521–527. IEEE, 2002. Citado 3 vezes nas páginas 6, 15 e 16.

[50] E. D. S. Paranaguá. Reconhecimento de locutores utilizando modelos de markov escondidos contínuos. *Mestrado em ciências em engenharia, Instituto Militar de Engenharia, Rio de Janeiro*, 1997. Citado 2 vezes nas páginas 33 e 34.

[51] L. Rabiner and B. Juang. An introduction to hidden markov models. *ieee assp magazine*, 3(1):4–16, 1986. Citado na página 31.

[52] L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989. Citado 3 vezes nas páginas 34, 35 e 38.

[53] L. R. Rabiner, S. E. Levinson, and M. M. Sondhi. On the application of vector quantization and hidden markov models to speaker-independent, isolated word recognition. *Bell System Technical Journal*, 62(4):1075–1105, 1983. Citado na página 33.

[54] M. Rasouli, Y. Chen, A. Basu, S. L. Kukreja, and N. V. Thakor. An extreme learning machine-based neuromorphic tactile sensing system for texture recognition. *IEEE transactions on biomedical circuits and systems*, 12(2):313–325, 2018. Citado na página 14.

[55] M. Rasouli, C. Yi, A. Basu, N. V. Thakor, and S. Kukreja. Spike-based tactile pattern recognition using an extreme learning machine. In *2015 IEEE Biomedical Circuits and Systems Conference (BioCAS)*, pages 1–4. IEEE, 2015. Citado 2 vezes nas páginas 13 e 15.

[56] J. Rekimoto. Smartskin: an infrastructure for freehand manipulation on interactive surfaces. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 113–120. ACM, 2002. Citado na página 1.

[57] R. B. Rocha et al. Desenvolvimento de um codificador de voz pessoal de baixa taxa baseada em modelos de markov escondidos. 2012. Citado 3 vezes nas páginas 3, 29 e 40.

[58] L. Rosa. Speech recognition using hidden markov models. *Available: http://www.advancedsourcecode.com/hmmspeech.asp*, accessed 3 December 2019. Citado 5 vezes nas páginas 49, 50, 51, 54 e 55.

[59] V. Savage, A. Head, B. Hartmann, D. B. Goldman, G. Mysore, and W. Li. Lamello: Passive acoustic sensing for tangible input components. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 1277–1280. ACM, 2015. Citado 2 vezes nas páginas 12 e 13.

[60] M. Schrapel, M.-L. Stadler, and M. Rohs. Pentelligence: Combining pen tip motion and writing sounds for handwritten digit recognition. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, page 131. ACM, 2018. Citado 2 vezes nas páginas 13 e 52.

[61] P. M. G. Soldado. Toolkit For Gesture Classification Through Acoustic Sensing. Master's thesis, Técnico Lisboa, 2015. Citado na página 1.

[62] T. Sumida, S. Hirai, D. Ito, and R. Kawakatsu. Raptapbath: User interface system by tapping on a bathtub edge utilizing embedded acoustic sensors. In *Proceedings of the 2017 ACM International Conference on Interactive Surfaces and Spaces*, pages 181–190. ACM, 2017. Citado na página 9.

[63] I. The MathWorks. Convolutional neural network. *Available: https://www.mathworks.com/solutions/deep-learning/convolutional-neural-network.html*, accessed 28 December 2019. Citado na página 52.

[64] T. Vujicic, T. Matijevic, J. Ljucovic, A. Balota, and Z. Sevarac. Comparative analysis of methods for determining number of hidden neurons in artificial neural network. In *Central european conference on information and intelligent systems*, page 219. Faculty of Organization and Informatics Varazdin, 2016. Citado na página 44.

[65] L. O. Werle et al. Analisadores virtuais baseados em modelo neural para monitoramento e controle de colunas de destilação com aquecimento distribuído. 2012. Citado na página 28.

[66] T. Westeyn, H. Brashear, A. Atrash, and T. Starner. Georgia tech gesture toolkit: supporting experiments in gesture recognition. In *Proceedings of the 5th international conference on Multimodal interfaces*, pages 85–92, 2003. Citado na página 51.

[67] B. Widrow and M. Hoff. Ire wescon convention record. *IRE, New York*, pages 96–104, 1960. Citado na página 18.

[68] R. Xiao, G. Lew, J. Marsanico, D. Hariharan, S. Hudson, and C. Harrison. Toffee: enabling ad hoc, around-device interaction with acoustic time-of-arrival correlation. In *Proceedings of the 16th international conference on Human-computer interaction with mobile devices & services*, pages 67–76. ACM, 2014. Citado na página 15.

[69] D. Xu, G. E. Loeb, and J. A. Fishel. Tactile identification of objects using bayesian exploration. In *2013 IEEE International Conference on Robotics and Automation*, pages 3056–3061. IEEE, 2013. Citado na página 14.

[70] D. Yu and L. Deng. Hidden markov models and the variants. In *Automatic Speech Recognition*, pages 23–54. Springer, 2015. Citado 3 vezes nas páginas 30, 31 e 32.

[71] M. Zhang, P. Yang, C. Tian, L. Shi, S. Tang, and F. Xiao. Soundwrite: Text input on surfaces through mobile acoustic sensing. In *Proceedings of the 1st International Workshop on Experiences with the Design and Implementation of Smart Objects*, pages 13–17. ACM, 2015. Citado na página 12.