

UNIVERSIDADE DE CAXIAS DO SUL - UCS
CAMPUS UNIVERSITÁRIO DA REGIÃO DOS VINHEDOS - CARVI
ÁREA DO CONHECIMENTO DE CIÊNCIAS EXATAS E ENGENHARIA
CURSO DE ENGENHARIA ELETRÔNICA

ALAN NUNES BONATTO

**DESENVOLVIMENTO DE PROTÓTIPO PARA
RECONHECIMENTO DE GESTOS DE UMA DAS MÃOS**

BENTO GONÇALVES

2017

ALAN NUNES BONATTO

**DESENVOLVIMENTO DE PROTÓTIPO PARA RECONHECIMENTO DE
GESTOS DE UMA DAS MÃOS**

Relatório do Trabalho de Conclusão de Curso II, apresentado a Área do Conhecimento de Ciências Exatas e Engenharias da Universidade de Caxias do Sul, no Campus Universitário da Região dos Vinhedos, como requisito parcial para a conclusão do curso em Engenharia Eletrônica.

Orientador: Prof. Me. Ricardo Becker

BENTO GONÇALVES

2017

ALAN NUNES BONATTO

**DESENVOLVIMENTO DE PROTÓTIPO PARA
RECONHECIMENTO DE GESTOS DE UMA DAS MÃOS**

Relatório do Trabalho de Conclusão de Curso II, apresentado a Área do Conhecimento de Ciências Exatas e Engenharias da Universidade de Caxias do Sul, no Campus Universitário da Região dos Vinhedos, como requisito parcial para a conclusão do curso em Engenharia Eletrônica.

Trabalho aprovado. Bento Gonçalves, dezembro de 2017.

Prof. Me. Ricardo Becker
Orientador

Prof. Me. Angelo Zerbetto Neto
Convidado 1

Prof. Dra. Marilda Machado Spíndola
Convidado 2

BENTO GONÇALVES

2017

Agradecimentos

Agradeço primeiramente aos meus pais Reni José Bonatto e Vera Lúcia Nunes Bonatto por todo apoio e incentivo não apenas a este trabalho, mas em todas as etapas de minha formação. Aos meus irmãos, segue meu reconhecimento pelos conselhos e auxílio durante todos esses anos.

Em especial a minha namorada Milena Bin, por sua compreensão durante todo este período, paciência, amor e apoio em todos os momentos de dificuldade. De mesmo modo, agradeço também a toda sua família.

Ao professor Me. Ricardo Becker, pela sua imprescindível orientação e apoio depositados a mim, os quais tornaram possível o desenvolvimento deste trabalho.

Ao participantes da banca examinadora, professor Me. Ângelo Zerbetto Neto e professora Dra. Marilda Machado Spíndola pelas críticas dirigidas, cujos conselhos acresceram conhecimento e qualidade a todo trabalho desempenhado.

Por fim, agradeço a todos os professores, funcionários e colegas da instituição que, por algum momento, compartilharam conhecimentos e experiências durante a minha vida acadêmica.

A todos o meu muito obrigado.

*“ A menos que modifiquemos a nossa
maneira de pensar, não seremos capazes de
resolver os problemas causados pela forma
como nos acostumamos a ver o mundo.
(Albert Einstein)*

Resumo

O crescimento contínuo de sistemas industriais e domésticos frente a deficiências na comunicação, sejam por questões de acessibilidade, segurança ou entretenimento, demonstram a necessidade atual por melhorias na efetividade de comunicação entre as diferentes partes de um processo. Neste âmbito, o reconhecimento de gestos torna-se uma solução interessante para aplicações que demandam maior interatividade entre o ser humano e um sistema final, seja este um meio tecnológico ou social. O presente trabalho apresenta a realização de um estudo das tecnologias envolvidas na detecção e caracterização de gestos das mãos e, por conseguinte, o desenvolvimento de um protótipo para a classificação destes sinais em ambiente embarcado, de modo a reduzir tamanho físico, recursos computacionais e custo. No protótipo descrito aplicaram-se tecnologias baseadas em sensores flexivos e inerciais, onde os sinais provenientes destes foram devidamente filtrados e codificados. Estes valores ajustaram-se aos dados de entrada para um classificador baseado em uma rede neural artificial, implementada em um microcontrolador, destinada as rotinas de treinamento e execução. Posteriormente, validou-se o sistema com a aplicação dos movimentos que caracterizam o alfabeto da linguagem brasileira de sinais.

Palavras-chave: reconhecimento de gestos. sistema embarcado. rede neural artificial. interação homem-máquina. linguagem brasileira de sinais.

Abstract

The continuous growth of industrial and household systems against communication deficiencies, for accessibility, safety or entertainment reasons, demonstrate the current need for improvements in effectiveness of communication between the different parts of a process. In this context, the gesture recognition becomes an interesting solution for applications requiring greater interactivity between human beings and a final system: a technological or social environment. This work presents a study of the technologies involved in the detection and characterisation of hand gestures, and therefore the proposal to develop a prototype for the classification of these signals in the embedded environment, so as to reduce physical size, computational resources and cost. In this prototype, technologies based on bend and inertial sensors are applied, where this signals was duly filtered and encoded. These values was defined as input data for a classifier based on an artificial neural network, implemented in a microcontroller, for the training and executing process. Subsequently, validation of the system was performed with the application of the movements that characterize the alphabet of the Brazilian sign language.

Keywords: gesture recognition. embedded environment. artificial neural network. human-machine interactions. Brazilian sign language.

Lista de ilustrações

Figura 1 – Protótipo para aquisição de gestos dos dedos com sensores inerciais. . .	17
Figura 2 – Sensores flexivos.	18
Figura 3 – Circuitos básicos de utilização do sensor flexivo.	18
Figura 4 – Protótipo de luva com superfícies condutivas.	19
Figura 5 – Resposta no domínio do tempo do filtro de média móvel.	20
Figura 6 – Resposta no domínio da frequência do filtro de média móvel.	21
Figura 7 – Atributos extraídos do sinal triaxial de um acelerômetro.	22
Figura 8 – Exemplificações do plano dimensional no modelo SVM.	23
Figura 9 – Séries temporais sobre a aplicação do DTW.	24
Figura 10 – Exemplo de padrão linearmente separável, classificável por um modelo perceptron de camada única.	25
Figura 11 – Neurônio artificial.	25
Figura 12 – Exemplo de padrões não linearmente separáveis, classificáveis por um modelo perceptron de múltiplas camadas.	25
Figura 13 – Múltiplos neurônios em uma rede neural artificial de três camadas. . .	26
Figura 14 – Funções de ativação típicas.	26
Figura 15 – Rede Alimentada Diretamente de Camada Única	27
Figura 16 – Rede Alimentada Diretamente de Múltiplas Camadas	28
Figura 17 – Rede Recorrente	29
Figura 18 – Esquema em blocos do projeto proposto.	34
Figura 19 – Fonte linear 6,5V.	36
Figura 20 – Circuito de aquisição e amplificação baseado em ponte de Wheatstone e ganho diferencial, para os sensores flexivos de 2,2". Ganho total de 1,55.	37
Figura 21 – Circuito de aquisição e amplificação baseado em ponte de Wheatstone e ganho diferencial, para os sensores flexivos de 4,5". Ganho total de 1,75.	38
Figura 22 – Alfabeto LIBRAS.	39
Figura 23 – Exemplo de resposta obtida com o método de validação cruzada. . . .	46
Figura 24 – Convergência da rede neural artificial implementada. MSE mínimo atingido de 0,1248% após 173700 épocas.	47
Figura 25 – Fluxograma simplificado do <i>firmware</i> (inicialização).	48
Figura 26 – Fluxograma simplificado do <i>firmware</i> (treinamento).	49
Figura 27 – Fluxograma simplificado do <i>firmware</i> (execução).	50
Figura 28 – Fluxograma da lógica de teste elaborada para realizar a aproximação do tempo de reconhecimento de gestos.	52
Figura 29 – Protótipo desenvolvido.	60
Figura 30 – Linguagem de Sinais Americana	67

Lista de tabelas

Tabela 1 – Comparativo entre as principais classificações de gestos.	15
Tabela 2 – Comparativo entre meios de aquisição usuais para caracterização gestual, com base nos estudos de Lin et al.(2013), Barczewska e Drozd (2013), Sidek e Hadi (2014), Quiapo e Ramos (2016), Lavanya et al. (2014) e Mulay e Yembarwar (2014).	32
Tabela 3 – Comparativo entre diferentes abordagens para caracterização gestual, com base nos estudos referenciados.	33
Tabela 4 – Variações de resistência para diferentes ângulos sobre a superfície dos sensores flexivos.	37
Tabela 5 – Método de diferenciação de símbolos do alfabeto LIBRAS, pela utilização de sensores complementares.	40
Tabela 6 – Normalização realizada sobre os dados dos sensores de entrada, após realização de ensaios aplicando o alfabeto LIBRAS.	42
Tabela 7 – Codificação para a saída do classificador.	43
Tabela 8 – Parâmetros da rede neural artificial aplicada.	47
Tabela 9 – Resultados da classificação do sistema proposto junto aos estudos de Carneiro et al. (2016) e Pathak et al. (2007).	53

Lista de abreviaturas e siglas

VGR	<i>Vision-Based Gesture Recognition</i>
SGR	<i>Sensor-Based Gesture Recognition</i>
HCI	<i>Human-Computer Interaction</i>
HHI	<i>Human-Human Interaction</i>
FPGA	<i>Field Programmable Gate Array</i>
MSE	<i>Mean Square Error</i>
RNA	Redes Neurais Artificiais
FANN	<i>Fast Artificial Neural Network</i>
SVM	<i>Support Vector Machine</i>
ELM	<i>Extreme Learning Machine</i>
DTW	<i>Dynamic Time Warping</i>
LIBRAS	Linguagem Brasileira de Sinais
RISC	<i>Reduced Instruction Set Computer</i>
ARM	<i>Advanced RISC Machine</i>
CMSIS	<i>Cortex Microcontroller Software Interface Standard</i>
DSP	<i>Digital Signal Processing</i>
ADC	<i>Analog to Digital Converter</i>
I2C	<i>Inter Integrated Circuit</i>
SPI	<i>Serial Peripheral Interface</i>
DMA	<i>Direct Memory Access</i>
RTOS	<i>Real Time Operating System</i>
AD	<i>Analog to Digital</i>
FAT	<i>File Allocation Table</i>

USB	<i>Universal Serial Bus</i>
RAM	<i>Random Access Memory</i>
ASL	<i>American Sign Language</i>

Sumário

1	INTRODUÇÃO	13
1.1	Objetivo Geral	13
1.2	Objetivos Específicos	14
1.3	Descrição Geral do Trabalho	14
2	RECONHECIMENTO DE GESTOS	15
2.1	Caracterização dos Gestos	15
2.2	Meios de Aquisição	16
2.2.1	Sistemas de Visão	16
2.2.2	Sistemas Sensoriais	16
2.2.2.1	Sensores Inerciais	17
2.2.2.2	Sensores Flexivos	17
2.2.2.3	Sensores de Contato	18
2.3	Filtragem	19
2.4	Extração de Atributos	20
2.5	Codificação	21
2.6	Classificadores	22
2.6.1	Suport Vector Machines (SVM)	22
2.6.2	Dynamic Time Warping (DTW)	23
2.6.3	Redes Neurais Artificiais (RNA)	24
2.6.3.1	Arquiteturas de Rede	26
2.6.3.1.1	Redes Alimentadas Diretamente (<i>Feedforward</i>) de Camada Única	26
2.6.3.1.2	Redes Alimentadas Diretamente (<i>Feedforward</i>) de Múltipla Camada	27
2.6.3.1.3	Redes Recorrentes (<i>Feedback</i>)	27
2.6.3.2	Processos de Aprendizagem	28
2.6.3.3	Algoritmos de Treino	30
2.6.3.3.1	<i>Batch Backprop</i>	30
2.6.3.3.2	<i>Incremental Backprop</i>	30
2.6.4	Extreme Learning Machines (ELM)	31
3	METODOLOGIA PROPOSTA	32
4	DESENVOLVIMENTO	35
4.1	Hardware	35
4.2	Firmware	38
4.2.1	Aquisição, Filtragem e Extração de Características	39

4.2.2	Classificação	41
4.2.3	Codificação	41
4.2.4	Sistema de Arquivos e Armazenamento USB	44
4.2.5	Treinamento e Operação da RNA	44
4.2.5.1	Metodologia de Treinamento	44
4.2.5.2	Algoritmo de Treinamento e Execução	46
5	RESULTADOS	51
5.1	Detecção de Gestos e Resultados da Classificação	51
5.2	Acuracidade da Técnica Proposta	51
6	CONSIDERAÇÕES FINAIS	54
	Referências	56
	APÊNDICE A – PROTÓTIPO DESENVOLVIDO	60
	APÊNDICE B – DATASET (PARCIAL) DE TREINO SENSOR FLEXIVO	61
	APÊNDICE C – DATASET (PARCIAL) DE TESTE SENSOR FLEXIVO	63
	APÊNDICE D – NETLIST (PARCIAL) DO SENSOR FLEXIVO	65
	APÊNDICE E – LINGUAGEM AMERICANA DE SINAIS	67

1 Introdução

Atualmente, com o avanço de tecnologias aplicadas em ambientes industriais e domésticos, observa-se uma crescente demanda por soluções voltadas a comunicação entre meios distintos. Sistemas situados em diferentes plataformas ou ambientes por vezes apresentam limitações que envolvem tempo e dificuldade de aprendizado, e conseqüentemente são ineficazes na troca de informação entre meios. Portanto, de modo a prover soluções que garantam maior facilidade e fluidez entre interações, torna-se necessário o desenvolvimento de tecnologias amigáveis capazes de traduzir sem esforço as intenções dos utilizadores em comandos, sem exigir que os mesmos aprendam linguagens específicas (HSU et al., 2015). Neste contexto, o reconhecimento de gestos torna-se uma solução interessante, uma vez que constitui uma maneira natural, intuitiva e conveniente de interação que facilita este processo (XIE e CAO, 2016). Caracteriza-se por aferir e classificar movimentos dos dedos, mãos, braços ou expressões faciais, sejam estes aplicados em interações homem-máquina (HCI), ou entre humanos (HHI). Estas tecnologias possuem aplicações variadas nos campos de acesso remoto, realidade virtual, linguagem gestual, autenticação de assinatura, ciência esportiva, cuidados de saúde e reabilitação médica (HSU et al., 2015).

Quando voltados a sistemas HCI, os esforços são baseados em soluções para ambientes agressivos ou de difícil acesso e, portanto, o reconhecimento de gestos pode ser aplicado para transmitir comandos de maneira segura e a distância (TRINDADE, 2010). Ademais, aplicações relacionadas a sistemas de entretenimento também são usuais, garantindo maior interatividade e facilidade de operação pelo usuário. Em relação a aplicações HHI, torna-se conveniente a utilização de sistemas que auxiliem na comunicação entre diferentes usuários, quando estes possuem limitações físicas ou cognitivas. Assim sendo, o reconhecimento de gestos neste meio dirige-se a facilitar e aumentar a qualidade de vida destas pessoas. Nesse sentido, a efetividade na comunicação entre essas pessoas permite uma grande conquista social, de valor significativo para aqueles que carregam tais limitações.

1.1 Objetivo Geral

Este trabalho tem como objetivo geral desenvolver um protótipo para a aquisição e classificação de sinais provenientes da movimentação de uma mão humana, aplicados ao reconhecimento de gestos estáticos e dinâmicos.

1.2 Objetivos Específicos

Em vista do objetivo geral abordado, tem-se como objetivos específicos necessários a execução do projeto:

- a) Avaliar a viabilidade de aplicação de diferentes sensores;
- b) Implementar *hardware* para a aquisição dos sinais provenientes dos sensores viáveis a aplicação;
- c) Desenvolver *firmware* em ambiente embarcado para o tratamento e classificação dos sinais aferidos dos sensores;
- d) Validar sistema desenvolvido pela execução de gestos específicos da linguagem brasileira de sinais (LIBRAS), através do auxílio de voluntários portadores de deficiência relacionada a fala.

1.3 Descrição Geral do Trabalho

O presente trabalho é composto de 5 capítulos. Para uma melhor compreensão, descreve-se brevemente o assunto abordado nos tópicos a seguir.

No segundo capítulo são apresentados métodos utilizados em estudos atuais para a aquisição de sinais provenientes de gestos da mão, bem como formas de tratamento e classificação.

O terceiro capítulo demonstra a metodologia abordada no desenvolvimento do trabalho, com base nas suas especificações e etapas de implementação. Adiante, no quarto capítulo, é exposto o detalhamento técnico acerca do desenvolvimento do trabalho.

Por fim, no quinto capítulo são apresentados os resultados do estudo, demonstrados conforme metodologia abordada.

2 Reconhecimento de Gestos

Neste capítulo serão abordados os principais tópicos relacionados ao processo de reconhecimento de gestos. Após sua contextualização, serão descritos os meios mais comuns de aferição, caracterização e classificação.

2.1 Caracterização dos Gestos

Os gestos são definidos como manifestações corporais, sendo elas voluntárias ou não, que revelam um estado psicológico ou a intenção de se transmitir uma informação. No contexto de comunicação, os esforços dividem-se entre reconhecimento de gestos classificados como estáticos, trajetórios ou dinâmicos, envolvendo movimentos dos dedos e das mãos (CHENG et al., 2016).

Gestos estáticos caracterizam-se pelo posicionamento fixo dos dedos em relação a mão. As posições dos dedos podem variar com o tempo, porém, no momento da aquisição os dados são aferidos sem qualquer relação com informações anteriores. Movimentos trajetórios, geralmente tratados apenas em sistemas de visão, são uma extensão dos movimentos estáticos, que indicam uma sequência de gestos com início e fim previamente definidos. Gestos dinâmicos por sua vez indicam que o posicionamento dos dedos e da mão podem variar com o tempo, e a indicação de começo e término do movimento é dada apenas pelo sistema de reconhecimento. Gestos estáticos podem representar caracteres enquanto informações mais complexas como palavras geralmente são descritas por movimentos dinâmicos (CHENG et al., 2016). Além dos movimentos relacionados a mão, gestos mais complexos que incluem o movimento do braço também são alvos de estudo, uma vez que os dados coletados com a utilização de sensores específicos permitem caracterização de movimentos mais amplos, como demonstram os estudos de Xu et al. (2012), Ji et al. (2015), Xie e Cao (2016) e Gupta et al. (2016). A tabela 1 expõe, de maneira sucinta, as diferenças entre os principais padrões de classificação gestual.

Tabela 1 – Comparativo entre as principais classificações de gestos.

Gesto	Dependência de Amostras Passadas	Característica	Segmentação	Informação
Estático	Não	Posição	Não	Informação Simples (Ex.: Caractere)
Dinâmico	Sim	Velocidade / Direção / Sequenciamento	Sim	Informação Complexa (Ex.: Sequência de Caracteres)

Fonte: O autor (2017).

2.2 Meios de Aquisição

Existem diversos meios de realizar o reconhecimento de gestos, que variam com os recursos alocados e viabilidade da aplicação. Tais abordagens são refinadas conforme o avanço das tecnologias nestes meios, podendo ser divididas em diferentes categorias.

2.2.1 Sistemas de Visão

Abordagens por sistemas de visão para reconhecimento de gestos (VGR) são caracterizadas por não necessitarem de *hardware* adicional ao sistema de visão propriamente dito. Em contrapartida, são altamente sujeitos a variações de luminosidade do ambiente, posicionamento e resolução da câmera para aquisição (ALON et al., 2009).

Os métodos de aquisição variam basicamente com a tecnologia e algoritmo de seleção empregado. Alon et al. (2009) realizou simultaneamente técnicas de segmentação espacial e temporal, com base em estimativas da posição atual e movimento da mão. Os dados foram classificados conforme identificação de bordas relacionadas a cor da pele a partir da subtração do fundo. Com a utilização de uma câmera 2D, foi possível a obtenção de taxas de acuracidade de até 85% para o reconhecimento de gestos dinâmicos. Carneiro et al. (2017) apresenta um estudo realizado sobre o reconhecimento de gestos estáticos aplicados a linguagem brasileira de sinais, baseado em FPGA e câmera com sensor de profundidade (Microsoft Kinect). Neste estudo foram apresentadas taxas de acuracidade superiores a 96%. Ainda utilizando sensoriamento 3D, Varshini et al. (2016) realizou a detecção de movimentos dinâmicos dos dedos com o auxílio de algoritmos de contorno, obtendo resultados similares. Enquanto grande parte dos estudos realizados utilizaram algoritmos de seleção específicos para aplicações visuais, classificadores baseados em redes neurais se mostraram soluções interessantes para gestos contínuos resultando em altas taxas de acuracidade, conforme estudos de Mekala et al. (2013) e Bobic et al. (2016).

2.2.2 Sistemas Sensoriais

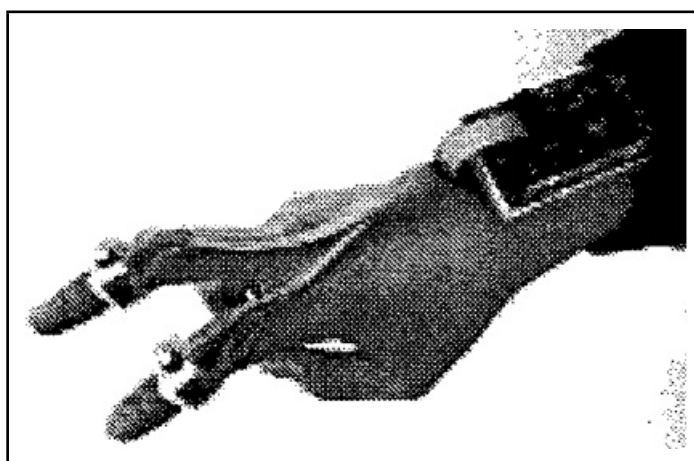
Sistemas sensoriais aplicados ao reconhecimento de gestos (SGR) caracterizam-se pela aquisição simples ou combinada de diferentes tipos de sensores, para a extração de parâmetros relacionados a posição angular dos dedos e a aceleração da mão durante o movimento. Estudos atuais operam com o auxílio de luvas para o posicionamento ótimo de tais sensores e a correta caracterização do gesto, a exemplo de Quiapo et al. (2016), Pathak et al. (2015), Hayek et al. (2014) e Mulay e Yembarwar (2014). Destacam-se a seguir os principais métodos de aquisição sensoriais.

2.2.2.1 Sensores Inerciais

Acelerômetros e giroscópios são descritos como componentes inerciais, uma vez que são sensíveis a variação linear e angular de movimento, respectivamente. Atualmente são o meio mais utilizado para aquisição dos sinais de interesse, pois fornecem todas as informações necessárias para a decomposição do movimento. No entanto, estão limitados a superfície em que são fixados, seja na palma da mão ou nos dedos.

Possuem variadas formas de aplicação. Em muitos casos são utilizados apenas para diferenciar gestos estáticos similares, a partir da orientação da mão. Quiapo et al. (2016) e Pathak et al. (2015) utilizaram o acelerômetro para este propósito em seu estudo, aumentando o nível de acuracidade sobre o reconhecimento do alfabeto proposto. Abordagens relacionadas ao movimento dos dedos também foram realizadas, conforme estudos de Sidek e Hadi (2014), Trindade (2010) e Lam et al. (2002), demonstrando níveis maiores de processamento, quando comparados aos sensores flexíveis. Por sua vez, Xie e Cao, (2016) e Gupta et al. (2016) o utilizaram como meio de caracterização de gestos mais amplos da mão, a partir da extração de características específicas do sinal. Por conter um número maior de informações que os demais sensores, geralmente necessita de algoritmo próprio para pré-processamento, filtragem e extração de dados para o classificador, como será discutido posteriormente. A Figura 1 demonstra o protótipo construído por Lam et al. (2002) em seu estudo.

Figura 1 – Protótipo para aquisição de gestos dos dedos com sensores inerciais.



Fonte: Lam et al. (2002).

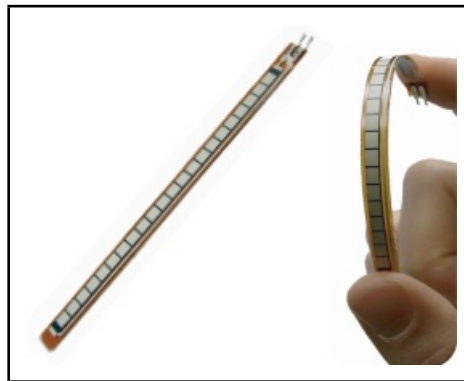
2.2.2.2 Sensores Flexivos

Sensores flexivos são transdutores axiais que sofrem variação de resistência com a flexão de seu material. A Figura 2 demonstra as características mecânicas de um sensor flexivo comercial. Por se tratar de um elemento resistivo, os circuitos para sua utilização

não envolvem soluções complexas ou componentes com maior custo, no entanto, podem variar de acordo com a aplicação. Exemplos de circuitos básicos com o sensor flexivo podem ser observados na Figura 3.

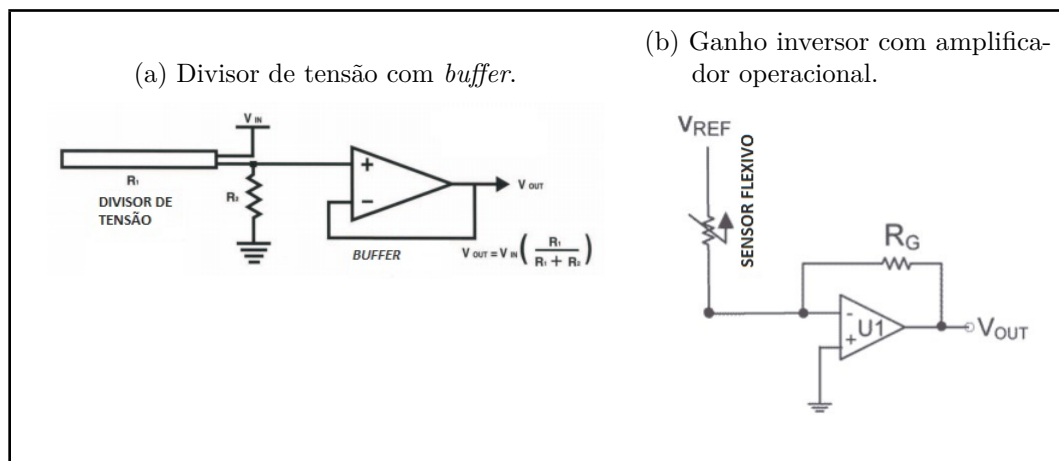
Já empregado em diversas áreas como industrial e médica, suas aplicações mais comuns incluem acesso remoto, acessibilidade e interfaces homem-máquina em geral, conforme reporta Mulay e Yembarwar (2014), em sua análise de estudos anteriores. No contexto de reconhecimento de gestos propriamente dito, são utilizados para a detecção da posição angular dos dedos em gestos estáticos, quando acoplados em luvas para a aquisição.

Figura 2 – Sensores flexivos.



Fonte: Adaptado de www.spectrasymbol.com (2014).

Figura 3 – Circuitos básicos de utilização do sensor flexivo.



Fonte: Adaptado de www.spectrasymbol.com (2014).

2.2.2.3 Sensores de Contato

Sensores de contato são comumente usados em conjunto com os flexivos. Geralmente são constituídos de superfícies condutivas sobre as extremidades dos dedos e a palma

da mão, as quais fornecem sinais digitais quando em contato. Em reconhecimento de gestos para linguagem de sinais mais complexos, auxiliam na distinção de gestos similares entre os dedos, aumentando a taxa de acuracidade nesses sistemas. Apesar de possuírem baixo custo de implementação, quando utilizados como solução isolada, permitem baixa variabilidade na detecção de gestos. Em seu estudo, Sekar et al. (2014) aplicou esta solução em conjunto aos sensores flexivos, estendendo significativamente o número de caracteres passíveis de serem detectados. Por sua vez, Lavanya et al. (2014) utilizou tais sensores como solução única, realizando um comparativo a outras tecnologias da área. Na Figura 4 é possível observar o protótipo construído neste estudo.

Figura 4 – Protótipo de luva com superfícies condutivas. O contato dos dedos com a área central da mão caracteriza diferentes variações de gestos.



Fonte: Lavanya (2014).

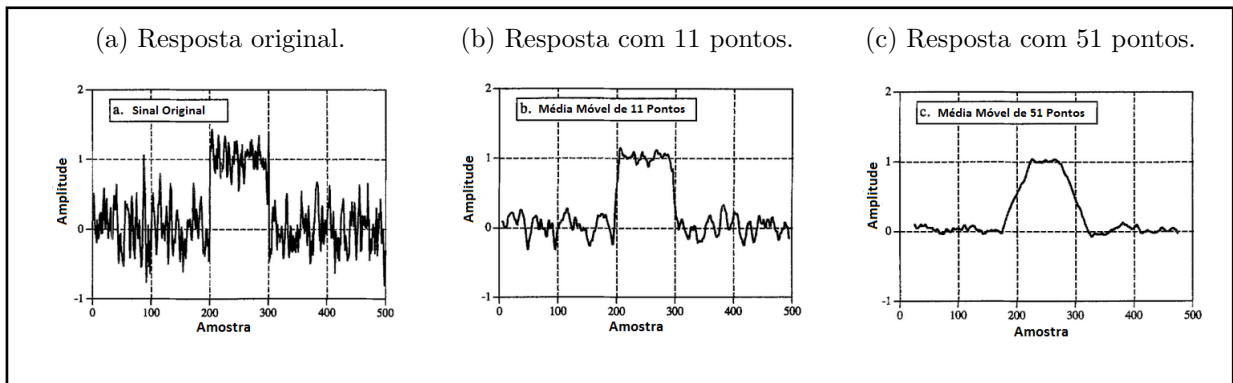
2.3 Filtragem

Devido as características intrínsecas do sinal e seu meio de aquisição, a aplicação de filtros torna-se necessária. Uma vez que sinais adquiridos pelo reconhecimento de gestos não contém informações significativas em frequência, conforme relatam Marqués e Basterretxea (2015) e Zhou et al. (2012), a aplicação de filtros no domínio do tempo geralmente é indicada. Um filtro com atuação ótima sob estas especificações é o filtro de médias móveis que, apesar de possuir comportamento inadequado em frequência, minimiza o erro aleatório e suaviza a resposta temporal (SMITH, 2013). Ao aplicar esta solução, uma janela de valores é definida de modo a delimitar o espaço de trabalho utilizado

no cálculo da média para cada amostra (TRINDADE, 2010). A equação 1 demonstra a representação matemática de um filtro básico baseado no cálculo contínuo de médias, onde o parâmetro W define o tamanho do *buffer* de trabalho, este obtido experimentalmente. A Figura 5 exibe a resposta do filtro quando aplicado um pulso limitado no tempo, para diferentes tamanhos de *buffer*. Por sua vez, a resposta em frequência do mesmo filtro pode ser observada na Figura 6. A técnica em questão apresentou desempenho adequado na maioria das aplicações relacionadas ao reconhecimento de gestos baseados em sistemas sensoriais, conforme disposto pelos estudos de Havalagi e Nivedita (2013), Trindade (2010), e mais recentemente por Hsu et al. (2015) e Xie e Cao (2016).

$$f_i = \frac{1}{N} \sum_{k=-N/2}^{N/2} s_{(i+k)} \quad , \quad N = \begin{cases} W - 1 & , \text{ se } W \text{ é ímpar} \\ W & , \text{ se } W \text{ é par} \end{cases} \quad (1)$$

Figura 5 – Resposta no domínio do tempo do filtro de média móvel. O incremento no número de dados do vetor de trabalho resulta em maiores níveis de suavização do sinal, no entanto, pode ocasionar atrasos de processamento.

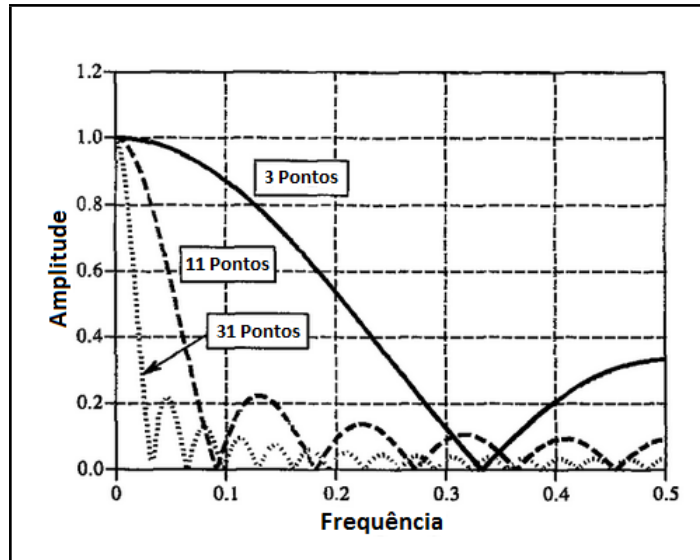


Fonte: Adaptado de Smith (2013).

2.4 Extração de Atributos

A extração de atributos é o processo pelo qual são obtidas as informações do sinal que refletem o comportamento do movimento para diferentes gestos, simplificando a quantidade de recursos que descrevem o movimento com acuracidade. Este é um fator importante para a eficiência de qualquer sistema de reconhecimento, uma vez que a seleção de características efetivas determinam o desempenho, bem como a complexidade computacional do sistema final (MARQUÉS e BASTERRETXEA, 2015). Geralmente a extração específica de dados torna-se necessária em sistemas baseados em acelerômetros, giroscópios e câmeras de visão, visto que os sinais captados por estes meios contém um número maior de parâmetros fornecidos. Xie e Cao (2016) aplicaram em seu estudo a seleção de 25 características extraídas do gesto segmentado, provenientes de um acelerômetro. De

Figura 6 – Resposta no domínio da frequência do filtro de média móvel. O incremento no número de dados do vetor de trabalho resulta em resposta minimamente plana sobre a faixa de passagem e *ripple* considerável sobre banda de rejeição.



Fonte: Adaptado de Smith (2013).

modo a avaliar a qualidade dos parâmetros escolhidos, os mesmos foram inseridos em um algoritmo de classificação de atributos. Com base no erro médio quadrático (MSE) do valor de saída do classificador, os melhores atributos foram escolhidos para a aplicação. Em outro estudo, Marqués e Basterretxea (2015) realizaram a comparação entre quatro grupos de características provenientes dos três eixos de um acelerômetro, divididas entre análises no domínio do tempo, da frequência, com a utilização de derivadas do domínio temporal e a transformada de *Wavelet*. A Figura 7 demonstra os atributos escolhidos neste estudo.

Com a utilização de diferentes classificadores para comparação, os autores concluíram que os grupos resultaram em altos níveis de acuracidade quando aplicados, porém ressaltam a utilização de atributos no domínio do tempo para ambientes embarcados, devido aos menores níveis de complexidade e esforço computacional. Ademais, aplicando-se algoritmos de seleção de atributos, foi verificada a possibilidade de redução do número de dados por grupo (para 7 ou 8), mantendo as acuracidades e restringindo a complexidade do preditor.

2.5 Codificação

A etapa de codificação consiste em preparar os dados extraídos do sinal de interesse, com base em um padrão pré-definido, para que estejam em um formato aceitável pelo método classificador. Diversas abordagens são realizadas, com base na aplicação, que

Figura 7 – Atributos extraídos do sinal triaxial de um acelerômetro.

SEL 1 (3x8) Domínio Temporal	SEL 2 (3x11) Domínio do Tempo e Frequência	SEL 3 (3x16) Domínio Temporal e Derivativo	SEL 4 Coefficientes de Wavelet
Média Mediana STD Abs Integral Correlação Máx Min MAD	Média Mediana STD Abs Integral Correlação Máx Min MAD Energia Entropia Máx (0-1-2-3 Hz) Banda de Potência (0-2,75Hz)	SEL 1 somada as derivadas de: Média Mediana STD Abs Integral Correlação Máx Min MAD	Diferentes "wavelet mãe" Diferentes níveis para os coeficientes de decomposição

Fonte: Adaptado de Marqués e Basterretxea (2015).

variam desde normalizações dos parâmetros de entrada (SOLA e SEVILLA, 1997) e (HSU et al. 2015), a aplicações de funções densidade de probabilidade (GUPTA et al., 2016) e código Jhonson de 4-bits (XIE e CAO, 2016). A escolha de uma codificação adequada permite maior eficiência do sistema e conseqüentemente menor tempo de treino pelo classificador empregado.

2.6 Classificadores

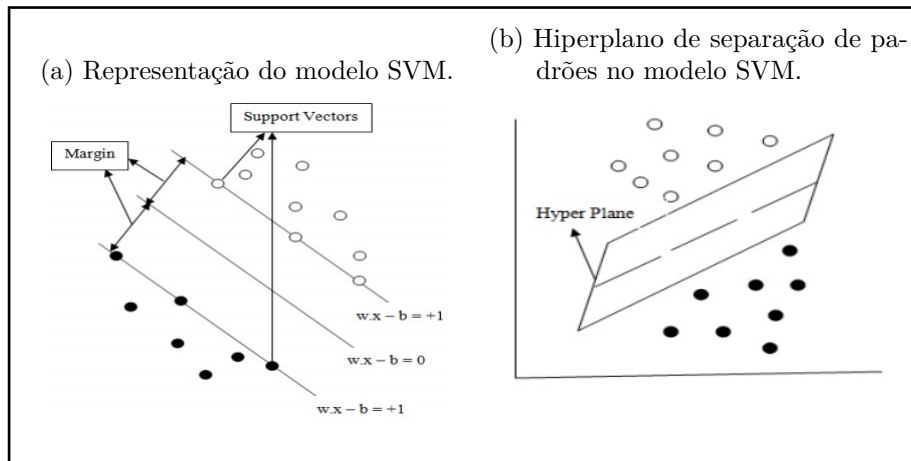
A última etapa do processo de reconhecimento de gestos consiste da interpretação dos atributos escolhidos e codificados de forma a identificar os gestos. A seguir serão descritos os métodos mais comuns utilizados como classificadores nesses sistemas.

2.6.1 Suport Vector Machines (SVM)

Máquinas de vetores de suporte (SVM) são algoritmos de aprendizado de máquina baseados na construção de hiperplanos lineares em um modelo hiper dimensional de representação dos padrões a serem classificados. A Figura 8 exemplifica a metodologia aplicada pelos algoritmos SVM. O exemplo descrito consiste de dois padrões diferentes a serem classificados pelo SVM (separados geograficamente pelo gráfico). A linha $w \cdot x - b = 0$ é denominada margem de divisão e, em conjunto com as demais linhas paralelas, constitui um hiperplano de separação dos padrões. A distância perpendicular entre as linhas que delimitam o hiperplano são chamadas margem e os componentes que se localizam nas

extremidades do hiperplano são denominados vetores de suporte. O objetivo do algoritmo então é maximizar a margem do hiperplano de forma a minimizar a ocorrência de erros e consequentemente obter uma classificação mais efetiva. O modelo é aplicado a sistemas lineares e não lineares. No entanto, nestes últimos, um mapeamento é realizado de modo a realocar os padrões e torná-los linearmente separáveis.

Figura 8 – Exemplificações do plano dimensional no modelo SVM.



Fonte: Pradhan (2012).

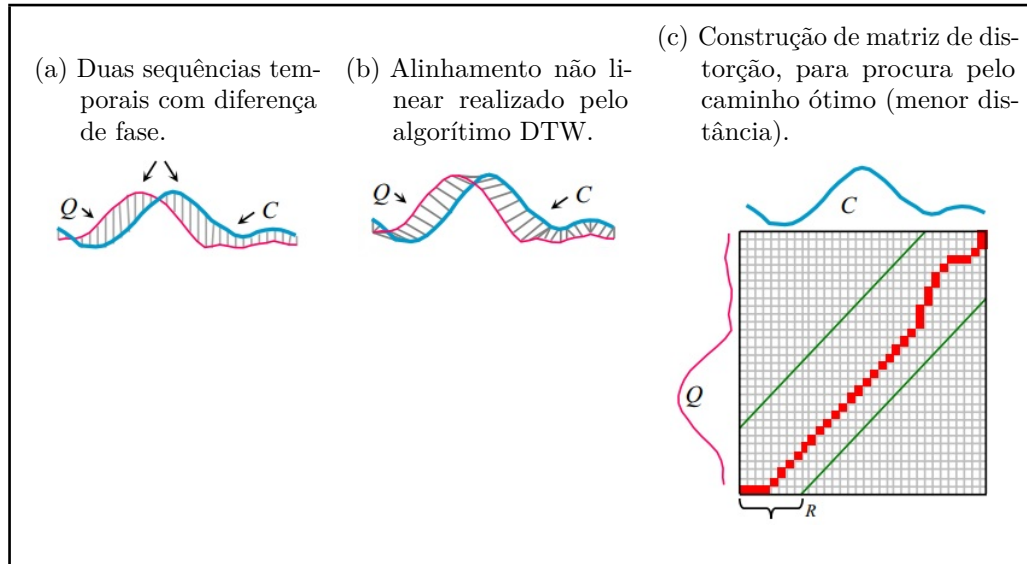
No estudo realizado por Marqués e Basterretxea (2015), o modelo obteve resultado ligeiramente superior aos demais classificadores. No entanto, como concluído pelos autores, sua maior complexidade de implementação não justificou o uso frente ao aumento mínimo de acuracidade observado.

2.6.2 Dynamic Time Warping (DTW)

Algoritmos baseados na distorção dinâmica do tempo (DTW) mensuram a similaridade entre duas sequências de dados temporais, que podem variar em número de amostras, encontrando a mínima distância entre elas. O método é chamado de *time warping* pois os dados são expandidos ou comprimidos de modo a se obter o melhor mapeamento. Desse modo, torna-se possível encontrar padrões entre duas medições de eventos realizados, mesmo que não estejam com uma mesma progressão sobre a série. No contexto de reconhecimento de gestos, é utilizado para comparar uma sequência aferida a um conjunto de dados base. A Figura 9 demonstra duas sequências de séries temporais e exemplifica como o algoritmo relaciona os dados para encontrar o melhor caminho. Durante a execução do algoritmo, este determina os passos através da matriz que minimizam a distância total entre as sequências. A distância nesse modelo é computada como a soma dos passos dados dividida por uma função de normalização. Através da Figura 9, torna-se clara a limitação do algoritmo: o número de possibilidades para a trajetória do caminho ótimo cresce com

o aumento das sequências temporais de entrada e, portanto, a utilização do algoritmo torna-se inviável com o aumento do esforço computacional requerido.

Figura 9 – Séries temporais sobre a aplicação do DTW.



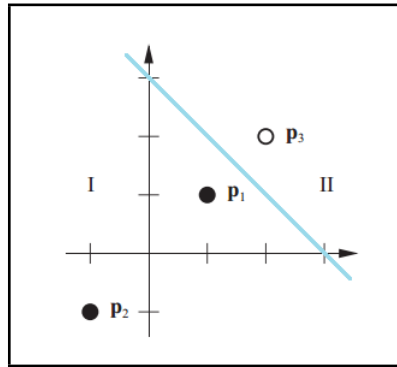
Fonte: Rakthanmanon et al. (2012).

2.6.3 Redes Neurais Artificiais (RNA)

Redes neurais artificiais são modelos computacionais com estrutura interna baseada no sistema nervoso humano. Consistem de elementos denominados neurônios, que contém entradas, saídas, coeficientes e uma função matemática de ativação. Diante disso, RNA's tem a possibilidade de combinar múltiplos neurônios para processar informações complexas. Possuem duas formas construtivas básicas, baseadas no modelo perceptron:

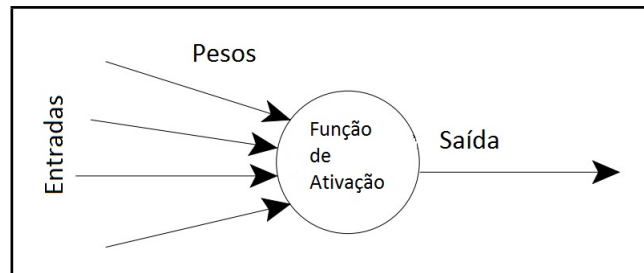
- (i) Perceptron de Camada Única: São a forma mais simples de rede neural utilizada para classificação de padrões ditos linearmente separáveis (Figura 10), ou em outros termos, diferenciáveis por uma reta em um hiperplano. Assim são constituídos de apenas um neurônio em sua estrutura, este com pesos ajustáveis, conforme demonstra o modelo da Figura 11.
- (ii) Perceptron de Múltiplas Camadas: constituem redes de maior complexidade, formadas por conjuntos de perceptrons de camada única, capazes de classificar padrões não linearmente separáveis (Figura 12). Neste modelo multicamadas, podem dispor de n camadas ocultas além das de entrada e saída, com neurônios interligados, conforme necessidade e complexidade da aplicação. A Figura 13 demonstra um exemplo de neurônios interligados em uma rede neural artificial de três camadas.

Figura 10 – Exemplo de padrão linearmente separável, classificável por um modelo perceptron de camada única.



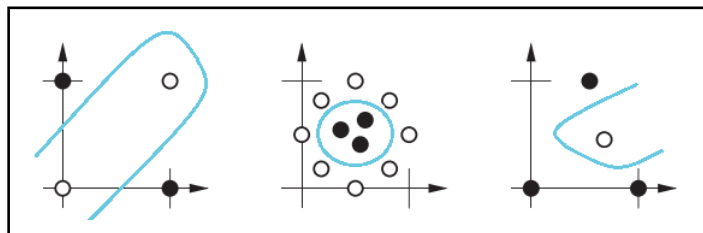
Fonte: Adaptado de Hagan et al. (1996).

Figura 11 – Neurônio artificial.



Fonte: Adptado de Gershenson (2003).

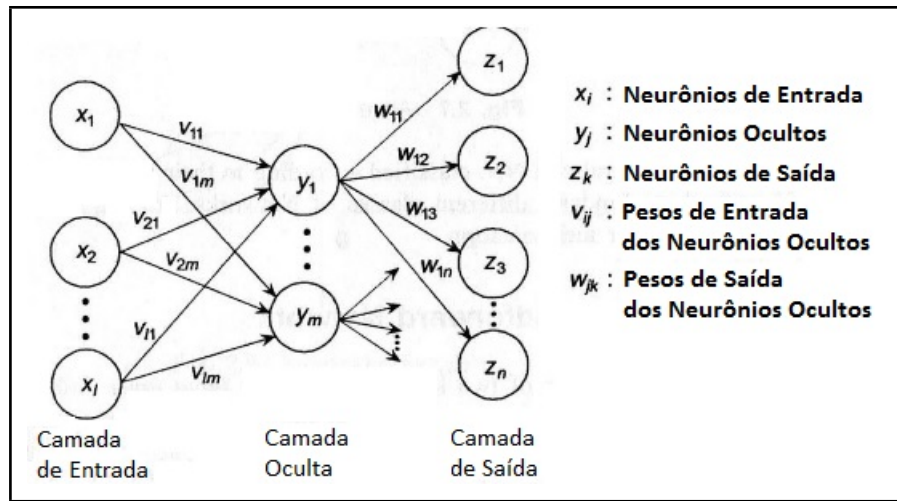
Figura 12 – Exemplo de padrões não linearmente separáveis, classificáveis por um modelo perceptron de múltiplas camadas.



Fonte: Adaptado de Hagan et al. (1996).

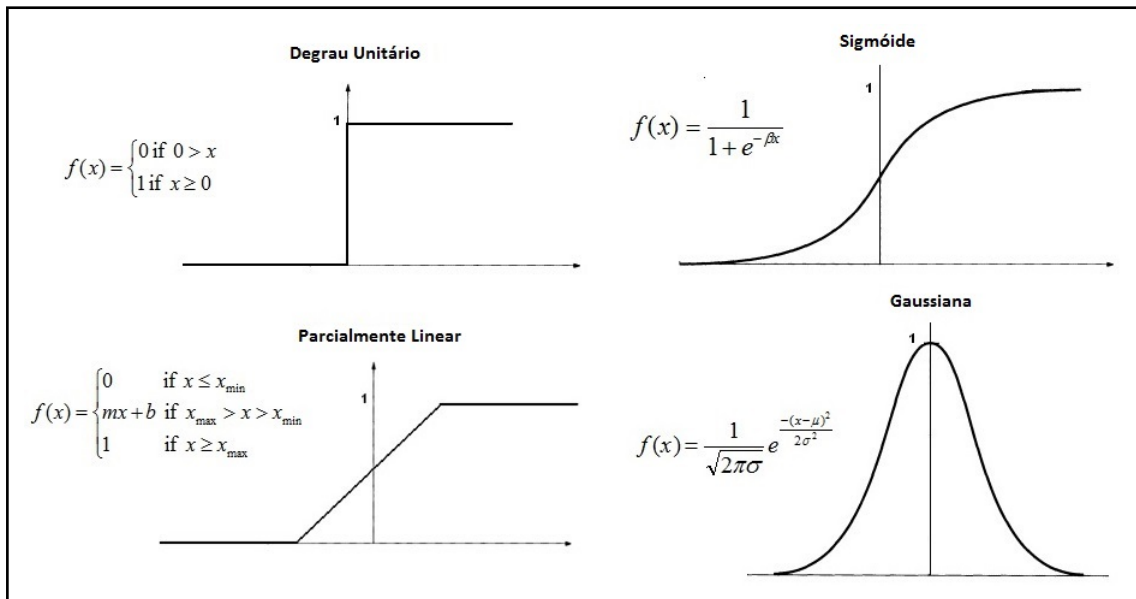
Cada coeficiente do neurônio, relacionado a uma entrada, determina o peso desta no sistema. A partir dos valores dos múltiplos coeficientes, o neurônio pode agir de determinada forma. As funções de ativação (Figura 14) descrevem como será realizado o processamento do valor de entrada no neurônio, similar a uma função de transferência. A escolha destas é um fator importante uma vez que definem como será o comportamento de cada elemento do sistema e, conseqüentemente, da rede neural. A cada iteração da rede, os coeficientes são atualizados de acordo com a aproximação das saídas a um valor esperado, em um processo denominado treinamento. Este varia com a arquitetura da rede, processo de aprendizagem e algoritmos de treino, itens descritos nos tópicos a seguir.

Figura 13 – Múltiplos neurônios em uma rede neural artificial de três camadas.



Fonte: Adaptado de Vora e Yagnik (2014).

Figura 14 – Funções de ativação típicas.



Fonte: Adaptado de Sayad (2017).

2.6.3.1 Arquiteturas de Rede

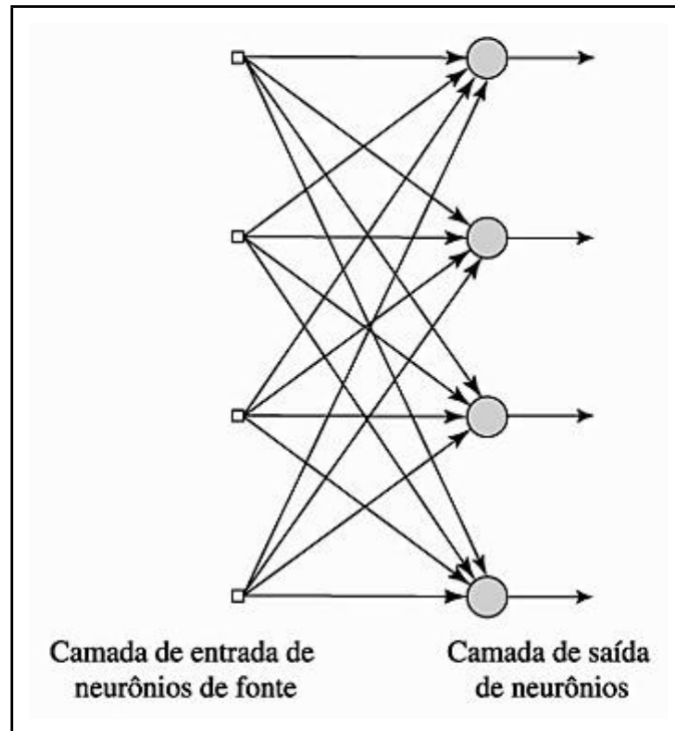
A topologia da RNA varia basicamente com o modo como os neurônios se relacionam para formar a rede. As topologias tradicionais se dividem em três estruturas: as redes recorrentes e as alimentadas diretamente de camada simples ou múltipla.

2.6.3.1.1 Redes Alimentadas Diretamente (*Feedforward*) de Camada Única

As redes alimentadas diretamente, ou *Feedforward* de camada única caracterizam-se por estabelecer conexões entre os neurônios de entrada e saída da rede diretamente, como

demonstra a Figura 15. Nela, o fluxo de informações percorre apenas um sentido, da entrada para a saída da rede. São geralmente aplicadas a problemas e classificação de padrões e filtragem.

Figura 15 – Rede Alimentada Diretamente de Camada Única



Fonte: Adaptado de Haykin (2001).

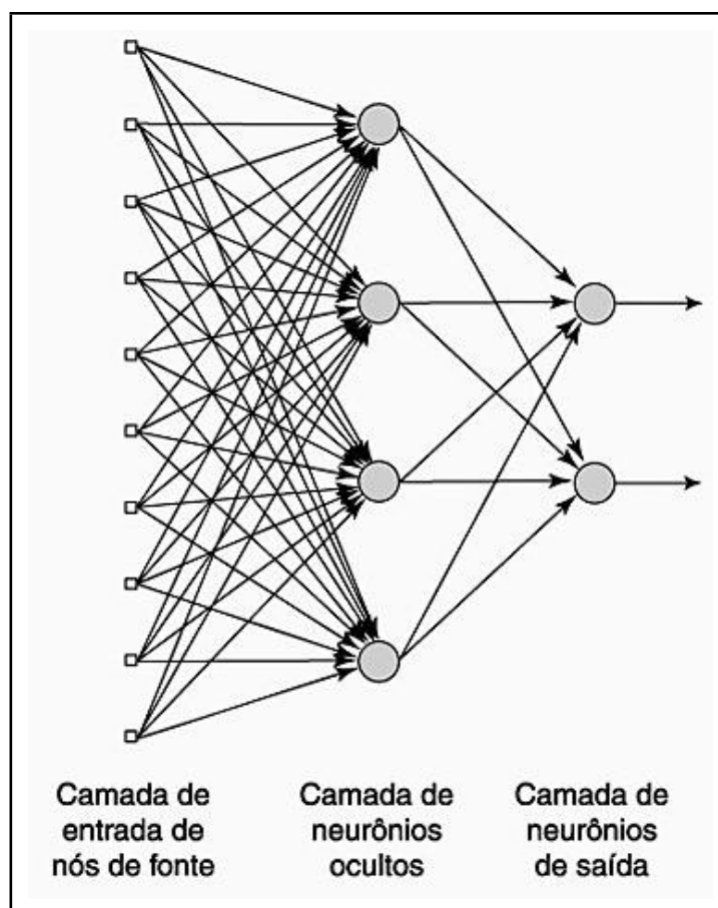
2.6.3.1.2 Redes Alimentadas Diretamente (*Feedforward*) de Múltipla Camada

As redes *Feedforward* de múltipla camada caracterizam-se com o mesmo sentido no fluxo de dados da rede que o modelo anterior, no entanto, possuem uma ou mais camadas ocultas. De modo similar, são aplicadas em problemas de classificação de padrões, aproximação de funções, identificação de sistemas e otimização. A Figura 16 demonstra esta topologia, em uma rede de três camadas.

2.6.3.1.3 Redes Recorrentes (*Feedback*)

As redes recorrentes possuem ao menos um ciclo ou realimentação na sua estruturação, permitindo que a informação percorra das saídas as entradas da rede, como demonstra a Figura 17, sendo aplicáveis assim a problemas variantes com o tempo. São utilizadas na identificação de sistemas, problemas de otimização e previsões de séries temporais.

Figura 16 – Rede Alimentada Diretamente de Múltiplas Camadas



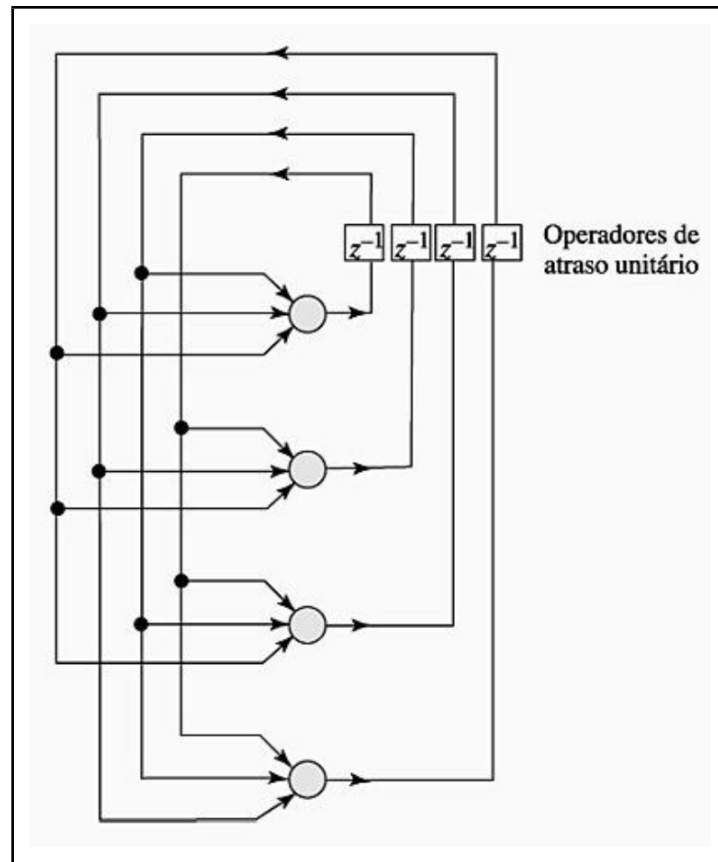
Fonte: Adaptado de Haykin (2001).

2.6.3.2 Processos de Aprendizagem

Uma propriedade de importância primordial para uma rede neural é sua habilidade de aprender a partir de seu ambiente e melhorar seu desempenho através da aprendizagem (Haykin, 2001). A melhoria da acuracidade da rede ocorre de acordo com padrões pré estabelecidos, que constituem modelos de cálculo para o ajuste dos pesos sinápticos da rede. Abaixo são descritas as principais regras básicas de aprendizagem de redes neurais.

- (i) Aprendizagem por Correção de Erro: Este modelo relaciona o valor esperado de saída de um neurônio a seu valor atual, resultando em um sinal de erro. Este por sua vez é aplicado a uma rotina de controle, de modo a corrigir os pesos sinápticos do neurônio em questão, projetados para aproximar a resposta atual a desejada.
- (ii) Aprendizagem Hebbiana: Método baseado na teoria de Hebb que indica a alteração do valor de um sinal sináptico com base em determinadas condições da rede. Quando dois neurônios são acionados simultaneamente a energia sináptica é relativamente aumentada, de outro modo, quando os neurônios são acionados de forma assíncrona, a força da sinapse é reduzida. O aprendizado Hebbiano portanto é realizado localmente,

Figura 17 – Rede Recorrente



Fonte: Adaptado de Haykin (2001).

ou seja, a mudança dos pesos sinápticos tem dependência somente dos neurônios a eles relacionados.

- (iii) **Aprendizagem Competitiva:** Neste modelo, os neurônios da camada de saída competem entre si para determinar qual será ativado, logo, apenas um deles pode estar ativo em um instante. Isto torna-se adequado para aplicações onde se deseja obter dados estatísticos a cerca de determinados valores de entrada. Neste contexto, um neurônio é denominado ativo quando a ação combinada de todas entradas diretas e realimentadas a ele seja superior as demais.
- (iv) **Aprendizagem de Boltzman:** Modelo de treinamento estabelecido para redes neurais recorrentes, onde os neurônios assumem apenas dois estados conhecidos e caracterizados por uma função de energia. A atualização dos neurônios é realizada de forma aleatória, até o momento em que o mínimo da função seja alcançada. Dois conjuntos de neurônios são previstos neste modelo: ocultos e visíveis. Enquanto estes formam uma interface com a rede, os demais operam livremente, sem contato com os meios de entrada ou saída.

As tarefas de aprendizado ainda podem ser classificadas em duas categorias básicas,

quanto a natureza dos dados e as informações disponíveis e requeridas pelo problema. Dessa forma, define-se:

- (i) **Aprendizagem Supervisionada:** caracterizado pela disposição dos padrões de aprendizado a rede, dados pelas relações entre as entradas e saídas desejadas, para que os pesos dos neurônios sejam calculados continuamente pelo algoritmo de treinamento.
- (ii) **Aprendizagem Não Supervisionada:** permite abordar problemas onde, em um primeiro momento, não se conhece o comportamento das saídas ou a relação destas com as entradas. Desta forma, pode-se trabalhar em uma estrutura de dados onde não se conhece o efeito das variáveis.

2.6.3.3 Algoritmos de Treino

Em redes neurais avançadas, que relacionam centenas de neurônios em sua estrutura, o ajuste dos pesos pode ser dificultado, logo, algoritmos específicos de treinamento são usualmente aplicados. A seguir serão descritos os principais modelos de treinamento utilizados.

2.6.3.3.1 *Batch Backprop*

Algoritmos *batch backpropagation* são baseados em modelos RNA multicamadas que operam em duas etapas: propagação e atualização de pesos. Quando uma entrada é apresentada a rede, a mesma é propagada camada a camada para a saída. O valor resultante então é comparado ao valor esperado através de uma função de perda, e o MSE é calculado para cada neurônio na camada de saída. Estes valores então são propagados a entrada, onde cada neurônio obtém um valor de erro associado a sua contribuição na saída. O método calcula o gradiente da função de perda com relação a todos os pesos da rede, que é então inserido ao algoritmo para a atualização dos pesos, de modo a minimizar a função de perda iterativamente. Isso significa que os pesos são atualizados após o cálculo do MSE de todo o conjunto de treino, período esse denominado época. Possuem maior simplicidade de implementação e cálculo comparadas a outras técnicas matematicamente complexas (VORA e YAGNIK, 2014), como *Quickprop* e *Rprop*, porém com períodos de aprendizado maiores e menores índices de erro.

2.6.3.3.2 *Incremental Backprop*

Este algoritmo tem operação similar ao *batch backprop*, porém os pesos são atualizados após cada padrão do conjunto de treino ser iterado. Isso significa que os pesos são atualizados diversas vezes em uma época. Em consequência, tempos menores de aprendizagem são atingidos para a maioria dos problemas bem como um valor maior no MSE da rede neural.

2.6.4 Extreme Learning Machines (ELM)

Sistemas baseados em máquinas de aprendizado extremo (ELM) são modelos simplificados das RNA tradicionais de três camadas, onde os coeficientes que interligam a camada de entrada a oculta são randomicamente gerados e nunca alterados. A partir desta definição, os coeficientes da camada de saída são determinados analiticamente, e não interativamente como no modelo tradicional. Introduzido por Huang et al. (2006), o método apresentou em seu estudo desempenho adequado de generalização na maioria dos casos e tempo de aprendizagem milhares de vezes inferior se comparados aos modelos tradicionais de treinamento de redes neurais artificiais. Aplicado ao reconhecimento de gestos, Marqués e Basterretxea (2015) demonstraram acuracidades similares entre os modelos ELM e RNA, ressaltando porém a velocidade superior de treinamento do modelo simplificado.

3 Metodologia Proposta

A caracterização da metodologia resulta da avaliação de trabalhos anteriores na área de reconhecimento gestual, de modo a identificar os métodos viáveis de implementação que, posteriormente, atinjam um resultado satisfatório perante os objetivos específicos do trabalho.

Um dos principais fatores no projeto são os meios de aquisição, pois estes aliados a uma extração de atributos que represente adequadamente as variáveis de interesse reduzem o erro do sistema. Desse modo, comparativos a cerca do meio aquisitivo foram realizados, dispostos na Tabela 2. Visto que a aplicação de detecção de gestos das mãos voltada a linguagem de sinais não estabelece local ou condições fixas do ambiente para a sua utilização, sensores que não possuem esta dependência são indicados. Em consequência, o processamento requerido para a aplicação diminui, viabilizando a utilização dos métodos sensoriais em ambientes embarcados. A presença de gestos similares no alfabeto, por vezes diferenciados pela posição ou até mesmo movimentação da mão em relação aos dedos, sugere a utilização de sensores complementares para a redução do erro de classificação.

Tabela 2 – Comparativo entre meios de aquisição usuais para caracterização gestual, com base nos estudos de Lin et al.(2013), Barczewska e Drozd (2013), Sidek e Hadi (2014), Quiapo e Ramos (2016), Lavanya et al. (2014) e Mulay e Yembarwar (2014).

Meio	Gesto Aplicáveis	Sensor	Aquisição	Aplicação	Processamento Requerido	Dependência de Condições Ambientais
Visão	Estático / Dinâmico	Câmera 2D/3D	Imagem	Mão e/ou Dedos	Alto	Sim
Inercial	Estático / Dinâmico	Acelerômetro / Giroscópio	Velocidade / Orientação	Mão e/ou Dedos	Médio	Não
Flexão	Estático	Haste Flexiva	Curvatura	Dedos	Médio	Não
Contato	Estático	Superfície Cobre	Contato	Dedos	Baixo	Não

Fonte: O autor (2017).

Os estudos relacionados de modo geral contemplam a utilização de variados meios de aquisição sensoriais, classificadores, plataformas de cálculo e variabilidade de gestos aplicados. A Tabela 3 traz um comparativo entre as características básicas de projeto avaliadas entre estes estudos. Nela, observa-se que sistemas baseados em arquiteturas com maior poder de processamento possuem as maiores taxas de acuracidade entre as referências, mesmo quando comparadas a um mesmo meio de aquisição e classificação de gesto. Dentre elas é observada maior ocorrência dos algoritmos que utilizam RNA e *DTW*, por sua baixa complexidade de implementação frente a outros classificadores além da elevada variabilidade de técnicas e parâmetros ajustáveis, que possibilitam maiores

opções de ajuste do modelo aplicado. Um fator importante a ser considerado nos estudos no entanto, é a limitação imposta pela aplicação, pois restringe ou amplia a variabilidade dos gestos aplicados e, conseqüentemente, a dificuldade de reconhecimento. Desse modo, gestos codificados pelos autores adicionam um grau de liberdade ao projeto, o que justifica em parte o nível de acuracidade observado. Em contrapartida, sistemas baseados em microcontroladores geralmente utilizam tabelas de comparação para a classificação dos dados, devido a maior necessidade de processamento imposto por classificadores mais complexos. Neste casos, os esforços de desenvolvimento geralmente são direcionados não aos classificadores, e sim aos meios de aquisição. Portanto, como observado anteriormente, aplicações relacionadas ao alfabeto de sinais geralmente requerem a combinação de múltiplos sensores para um reconhecimento com níveis superiores de acuracidade.

Tabela 3 – Comparativo entre diferentes abordagens para caracterização gestual, com base nos estudos referenciados.

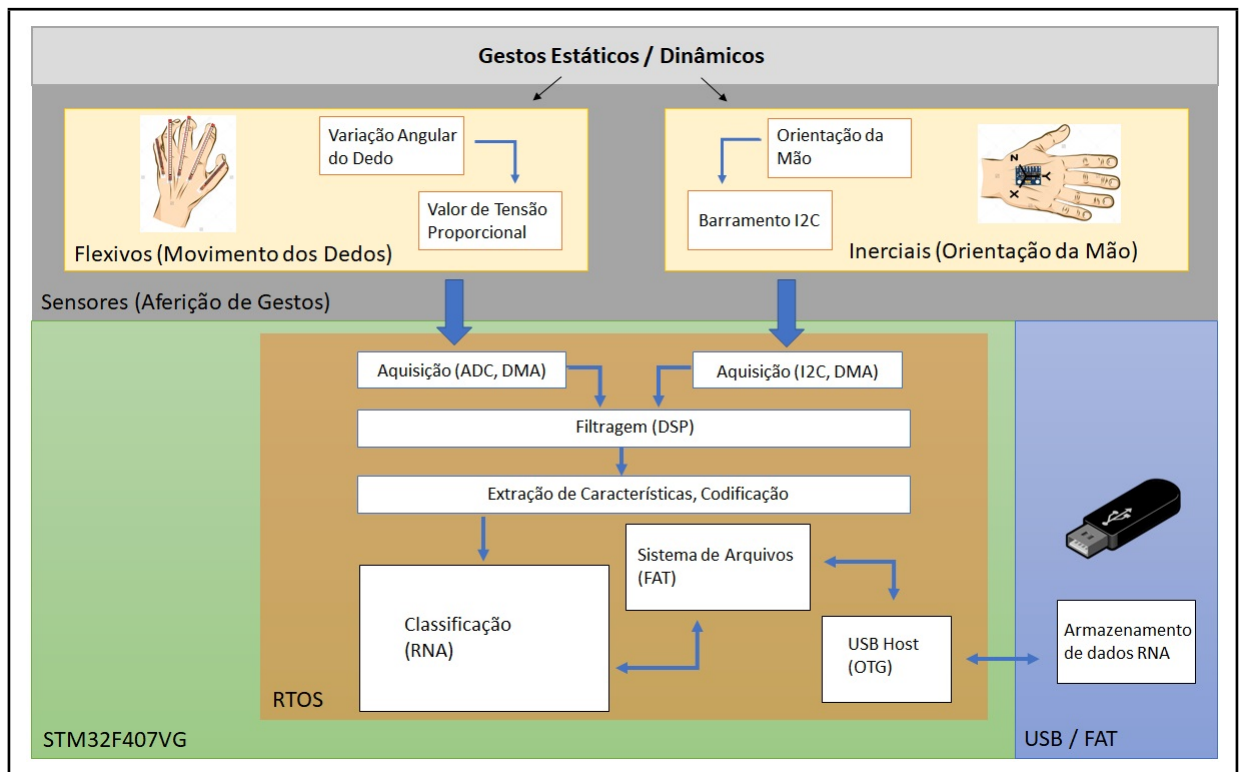
Trabalho Apresentado	Sensor	Classificador	Plataforma Cálculo	Gesto		Aplicação	Acuracidade Média
GUPTA et al. (2016)	Inercial	DTW	Smartphone	Mão	Dinâmico	Gestos Codificados	93,57%
XIE & CAO (2016)	Inercial	RNA	Computador	Mão	Dinâmico	Gestos Codificados	99,88%
MARQUÉS & BASTERRETXEA (2015)	Inercial	RNA	Computador	Mão	Dinâmico	Gestos Codificados	98,98%
		SVM					99,04%
		ELM					98,54%
HSU et al. (2015)	Inercial	DTW	Computador	Mão	Dinâmico	Gestos Codificados	97,90%
BARCZEWSKA & DROZD (2013)	Inercial	DTW	Computador	Dedos	Dinâmico	Gestos Codificados	94,80%
MULAY & YEMBARWAR (2014)	Flexivo	Comparação	Microcontrolador	Dedos	Estático	Sistemas de Segurança	88,70%
HAYEK et al. (2014)	Flexivo	Comparação	Microcontrolador	Dedos	Estático	Alfabeto Codificado	94%
PATHAK et al. (2015)	Inercial, Flexivo e Contato	Comparação	Microcontrolador	Dedos e Mão	Dinâmico	Alfabeto de Sinais Americano	91,53%
QUIAPO & RAMOS (2016)	Flexivo	Comparação	Microcontrolador	Dedos e Mão	Dinâmico	Alfabeto de Sinais Americano (Parcial)	93,75%
SEKAR et al. (2014)	Flexivo e Contato	Comparação	Microcontrolador	Dedos	Estático	Alfabeto de Sinais Americano	80%

Fonte: O autor (2017).

Com base na análise sobre o referencial teórico apresentado e com os objetivos estabelecidos, define-se a proposição do trabalho. A metodologia proposta consiste em aplicar uma rede neural artificial, em ambiente embarcado, para a classificação de gestos

estáticos e dinâmicos com base em valores pré definidos, estes relacionados ao alfabeto LIBRAS. Neste meio, a proposta inclui o desenvolvimento de *hardware* e *firmware* necessários para a implementação, envolvendo a utilização de sensores flexivos e inerciais para a aquisição de sinais em sistemas de tempo real, além de rotinas de filtragem, extração de atributos, codificação, classificação e sistemas de arquivos para armazenamento externo de parâmetros. Apresenta-se na Figura 18 o diagrama de blocos da metodologia proposta, contendo as etapas descritas anteriormente, para a classificação dos sinais de interesse.

Figura 18 – Esquema em blocos do projeto proposto.



Fonte: O autor (2017).

Desse modo, descreve-se no capítulo a seguir as etapas de implementação da metodologia proposta.

4 Desenvolvimento

Os passos do projeto podem ser detalhados em duas etapas específicas de desenvolvimento, relativos a construção dos elementos físicos e lógicos necessários. Os componentes do *hardware* proposto contemplam a utilização de microcontrolador, sensores e circuitos auxiliares de instrumentação. Os itens de *firmware* por sua vez, dispõem de rotinas de aquisição, filtragem, codificação, classificação e armazenamento de arquivos. As etapas de projeto, divididas entre estas implementações, serão especificadas nos tópicos seguintes.

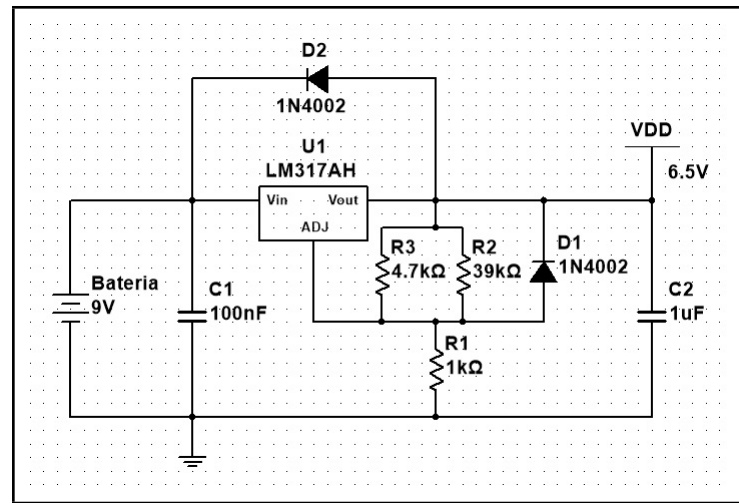
4.1 *Hardware*

A proposta de desenvolvimento de uma aplicação móvel sugeriu a utilização de *hardware* dedicado, composto de uma plataforma de cálculo simplificada, porém robusta. O microcontrolador STM32F407VG, produzido pela *ST Microelectronics*¹, foi então alocado de modo a se obter o desenvolvimento do projeto em ambiente embarcado, com nível de processamento adequado para a aplicação. O mesmo possui núcleo baseado em *Advanced RISC Machine* (ARM) Cortex M4, 1 Mb de memória *flash*, 196 kB de memória de acesso randômico (RAM) e frequência de operação de 168 MHz. Por operar sobre a família 4 do cortex ARM, conta com rotinas internas de processamento digital de sinais (DSP): opção interessante por viabilizar o tratamento interno dos sinais aferidos. Definiu-se a utilização do microcontrolador junto ao *kit* de desenvolvimento *Discovery* com o intuito de facilitar o andamento do projeto, possibilitando a concepção do *firmware* ser realizada de forma independente do *hardware*, uma vez que o circuito base do microcontrolador não teve a necessidade de ser implementado. Em um primeiro momento foi sugerida a implementação do hardware em uma configuração para testes e, portanto, questões de eficiência e consumo de energia não foram avaliadas nesta etapa. Desse modo, uma fonte linear de 6,5V foi prevista, em conjunto com uma bateria de corrente de carga reduzida, para alimentar o microcontrolador e demais circuitos. A fonte projetada pode ser observada na Figura 19.

A implementação do sistema, conforme modelo proposto, foi baseada na caracterização de gestos da mão através de tecnologias sensoriais. Estas demonstram, conforme referências, maior viabilidade de implementação em meio embarcado pois permitem *hardware* com complexidade inferior, bem como reduzida dependência de condições do ambiente e posição do usuário. As aquisições relacionadas ao *hardware* empregado envolvem a aferição de dados relativos a posição dos dedos em conjunto com a orientação da mão, de modo a elevar a acuracidade do sistema de reconhecimento de gestos. A partir desta definição, tornou-se necessária a construção de um protótipo para a aquisição de ambos os

¹ <http://www.st.com/>

Figura 19 – Fonte linear 6,5V.



Fonte: O autor (2017).

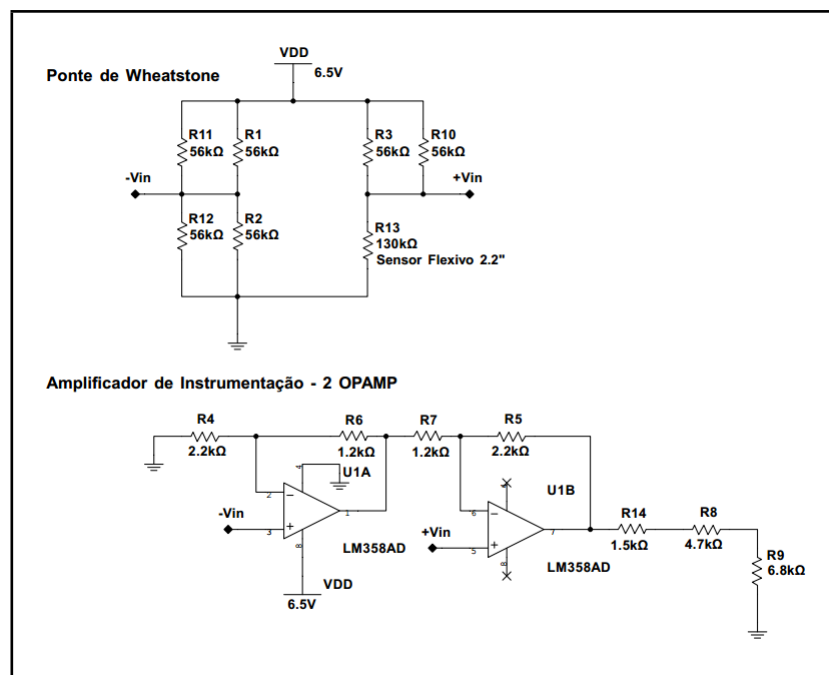
sinais de interesse. Este, constituído de sensores posicionados em locais específicos de uma luva para a correta caracterização do gesto, similar ao aplicado nos trabalhos de Hayek et al. (2014) e Pathak et al. (2015). Para a identificação da posição dos dedos, soluções com sensores inerciais e flexivos foram avaliadas. Enquanto a aquisição destes é realizada com o auxílio de conversores analógico-digital (ADC), os acelerômetros comerciais trabalham com protocolos de comunicação, geralmente *Inter Integrated Circuit* (I2C) ou *Serial Peripheral Interface* (SPI). Conseqüentemente, apresentam maiores atrasos na leitura do sinal. Ademais, a complexidade relacionada as aquisições de múltiplos acelerômetros (necessários para a caracterização da posição de cada dedo) associada as limitações de número de canais I2C do microcontrolador; bem como taxas de acuracidade similares a outras soluções, observadas entre os estudos de Pathak et al. (2015) e Barczewska e Drozd (2013), não viabilizaram a utilização deste sensor para a caracterização das posições dos dedos. Estas tiveram então a aquisição realizada através dos sensores flexivos comerciais de 4,5" para os dedos indicador, médio e anelar, e 2,2" para o polegar e o mínimo. Com base em suas variações máximas de resistência em relação a curvatura de sua superfície e os limites de tensão desejados para a entrada dos canais ADC do microcontrolador, foi desenvolvido um circuito de instrumentação para os sinais aferidos, de forma a dispersar estes ao intervalo de 0V a 3V. A partir dos dados dispostos da Tabela 4, obtidos através de ensaios sobre os sensores adquiridos, foram calculados os ganhos necessários para os sinais. O circuito de instrumentação descrito constitui uma ponte de Wheatstone junto a um estágio de ganho diferencial, disposto nas Figuras 20 e 21 para ambos os modelos de sensores. Posteriormente os sinais foram adquiridos simultaneamente através de 5 canais ADC independentes, com o auxílio de rotinas de acesso direto a memória (DMA). O protótipo desenvolvido, contemplando o *hardware* de aquisição para os sinais dos movimentos relativos aos dedos e a mão, pode ser observado no apêndice A.

Tabela 4 – Variações de resistência para diferentes ângulos sobre a superfície dos sensores flexivos.

Sensor Flexivo	Resistência (0°)	Resistência (45°)	Resistência (90°)	Resistência (aprox. 180°)
2,2"	28 kΩ	45 kΩ	70 kΩ	130 kΩ
4,5"	9,4 kΩ	14 kΩ	25 kΩ	35 kΩ

Fonte: O autor (2017).

Figura 20 – Circuito de aquisição e amplificação baseado em ponte de Wheatstone e ganho diferencial, para os sensores flexivos de 2,2". Ganho total de 1,55.

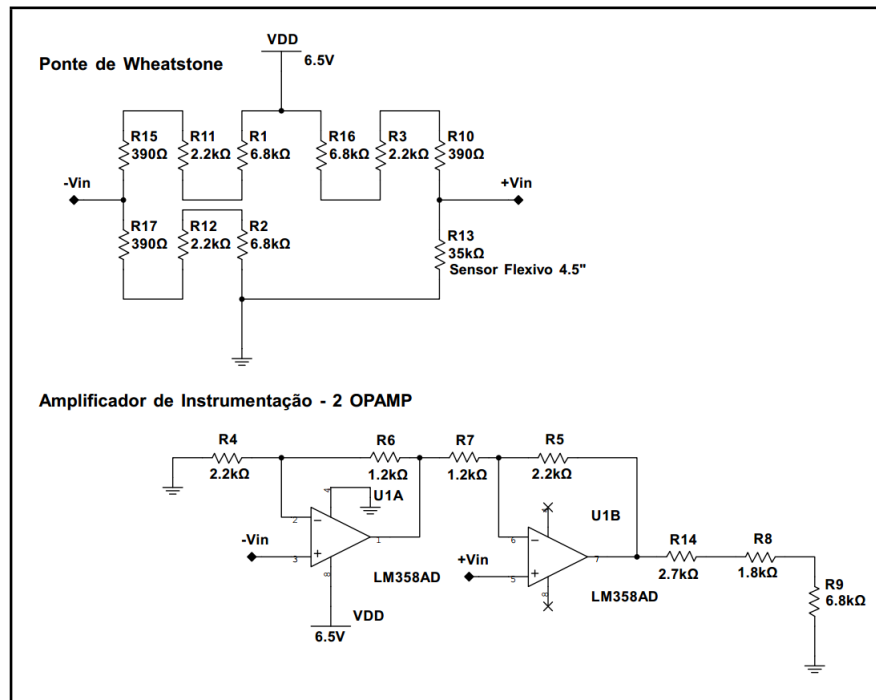


Fonte: O autor (2017).

Com o intuito de detectar a orientação da mão, e deste modo, diferenciar gestos estáticos similares em LIBRAS, foi considerada a utilização de um acelerômetro no sistema. Para tal, foi escolhido o módulo GY-521 com o circuito integrado MPU6050, que contempla a aquisição de dados de temperatura e acelerações linear e angular. Para a aplicação, se mostram interessantes apenas os dados aferidos pelo acelerômetro linear, através de comunicação I2C com o módulo. Nesta, a utilização de DMA também foi implementada, de modo a minimizar os atrasos na leitura do sinal. Além disso, as dimensões reduzidas do módulo bem como sua interface de comunicação simplificada o tornaram uma opção viável para a posterior fixação no protótipo proposto.

Posteriormente, foi prevista a validação do sistema em um modelo que fosse possível obter variadas posições dos dedos e da mão, com um significado lógico, de modo a representar seu comportamento para um caso usual. Assim, a partir do alfabeto

Figura 21 – Circuito de aquisição e amplificação baseado em ponte de Wheatstone e ganho diferencial, para os sensores flexivos de 4,5". Ganho total de 1,75.



Fonte: O autor (2017).

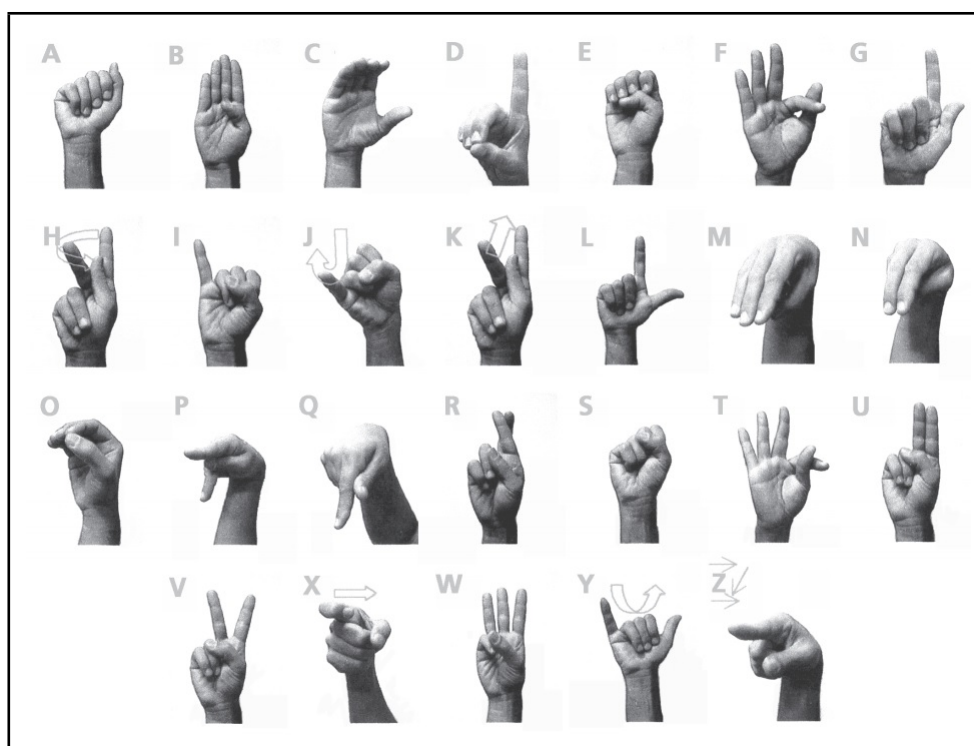
LIBRAS disposto na Figura 22, foi estabelecida a relação de sensores (Tabela 5) necessária para a caracterização de cada gesto sendo que, em um primeiro momento, sugeriu-se o reconhecimento dos gestos estáticos para a aplicação.

4.2 Firmware

As aplicações do modelo proposto necessitam que o *firmware* contemple a utilização de diferentes periféricos, como ADC, I2C, DMA, *Universal Serial Bus* (USB) e *Timers*. Para o correto funcionamento destes, uma escala de prioridades teve que ser definida, de modo a não haver interrupções de código em rotinas críticas. Por vezes, tal controle é dificultado, seja pelo número de periféricos relacionados ou pelo tempo de execução de tarefas. Sendo assim, de modo a se obter o controle preciso de rotinas a serem executadas, com níveis de prioridade de interrupção e execução, foi implementada na etapa inicial um sistema operacional de tempo real (RTOS). Para tal, foram utilizadas as bibliotecas FreeRTOS², de ampla utilização em plataformas onde sistemas mais complexos, como o Linux, não estão disponíveis. O desenvolvimento do *firmware* foi baseado nessa estrutura e, inicialmente, uma *thread* voltada a execução das rotinas de aquisição e tratamento dos dados foi prevista. Outras ainda foram criadas no decorrer do trabalho, de modo a auxiliar as tarefas principais do sistema. Todo o desenvolvimento e experimentação do *firmware*

² <http://www.freertos.org/>

Figura 22 – Alfabeto LIBRAS.



Fonte: Felipe e Monteiro (2006).

foram realizados pelo *software* IAR Embedded Workbench, da proprietária IAR Systems. As seções a seguir descrevem as rotinas nele implementadas.

4.2.1 Aquisição, Filtragem e Extração de Características

A etapa de aquisição, primeira tarefa de pré-processamento realizada, teve esforços aplicados no sentido de otimizar a captura de dados para o classificador. Atrasos nesta etapa podem ocasionar a obtenção de parâmetros incorretos para o gesto aplicado e, conseqüentemente, comportamentos indesejados do preditor. Conforme comentado anteriormente, a aquisição foi dividida em dois meios no modelo proposto. Os dados dos canais analógico-digital (AD) foram convertidos utilizando 12 bits de resolução e referência interna de tensão. Interrupções neste caso não foram necessárias, e os dados puderam ser filtrados diretamente do *buffer* de transmissão DMA. As demais aquisições foram realizadas através do componente MPU6050. Apesar deste circuito integrado dispor de funções de processamento internas, estas não foram utilizadas, permitindo assim a aquisição dos dados a uma taxa máxima suportada de 1 kHz. A sensibilidade do acelerômetro varia entre as referências adotadas e, para a aplicação, teve seu ajuste inicial em $\pm 4g$. Estas opções assim como outras relacionadas a interrupções externas e *clock* de operação tiveram suas configurações estabelecidas em uma primeira etapa, esta anterior ao início das aquisições DMA. Nesta, uma interrupção tornou-se necessária, de modo a viabilizar a obtenção dos

Tabela 5 – Método de diferenciação de símbolos do alfabeto LIBRAS, pela utilização de sensores complementares.

Caractere	Sensor Base	Posição de Dedos	Sensor Complementar	Posição da Mão	Gesto
A	Flexivo	A	-	-	Estático
B	Flexivo	B	-	-	Estático
C	Flexivo	C	-	-	Estático
D	Flexivo	D	-	-	Estático
E	Flexivo	E	-	-	Estático
F	Flexivo	F	-	-	Estático
G	Flexivo	G	Inercial	Y(90°)	Estático
H	Flexivo	P	Inercial	X(rotação)	Dinâmico
I	Flexivo	I	Inercial	Y(90°)	Estático
J	Flexivo	I	Inercial	Y(90°, 45°, -45°, 90°)	Dinâmico
K	Flexivo	P	Inercial	Y(-45°, 90°)	Dinâmico
L	Flexivo	L	-	-	Estático
M	Flexivo	W	Inercial	Y(-45°)	Estático
N	Flexivo	U	Inercial	Y(-45°)	Estático
O	Flexivo	O	-	-	Estático
P	Flexivo	P	Inercial	Y(0°)	Estático
Q	Flexivo	G	Inercial	Y(-90°)	Estático
R	Flexivo	R	-	-	Estático
S	Flexivo	S	-	-	Estático
T	Flexivo	T	-	-	Estático
U	Flexivo	U	Inercial	Y(90°)	Estático
V	Flexivo	V	-	-	Estático
W	Flexivo	W	Inercial	Y(90°)	Estático
X	Flexivo	X	-	-	Estático
Y	Flexivo	Y	-	-	Estático
Z	Flexivo	Z	-	-	Estático
REPOUSO	Flexivo	Dedos Alongados	-	-	Estático

Fonte: O autor (2017).

dados do acelerômetro contidos no pacote de 14 bytes transmitido pelo circuito integrado.

Os dados provenientes da aquisição foram filtrados no domínio temporal, conforme referências avaliadas. Foi sugerida a utilização de filtro de médias móveis assimétrico e recursivo, conforme conceitua Smith (2013), de tamanho a ser definido empiricamente. Assim, foi determinado inicialmente um valor, com base nos estudos de Hsu et al. (2015) e Xie e Cao (2016), e a partir deste avaliado o impacto sobre a estabilidade e atraso do sinal. Os ensaios resultaram em um *buffer* de 5 posições, sem qualquer influência significativa sobre o sinal. Foi prevista também a utilização das bibliotecas *Cortex Microcontroller Software Interface Standard*³ (CMSIS) relacionadas a operações de DSP, desenvolvidas para otimizar as rotinas de processamento em plataformas ARM.

Posteriormente, baseado nos estudos comparativos de Marqués e Basterretxea (2015), realizou-se a seleção de características temporais para a extração. Foi prevista

³ <https://www.arm.com/>

em um primeiro momento a utilização das médias obtidas na etapa de filtragem para caracterizar a aquisição dos gestos estáticos. Em seguida, a partir do comportamento esperado da rede, verificou-se a não necessidade de se extrair informações adicionais que poderiam ser interessantes para a aplicação.

4.2.2 Classificação

Frente as análises realizadas no referencial, foi proposta a utilização da rede neural artificial como método de classificação, pois constitui uma técnica amplamente difundida entre os estudos recentes, a qual possibilita acuracidades elevadas quando aplicada a sistemas similares ao proposto. Ademais, juntamente com as técnicas ELM, possuem as menores complexidades entre os modelos estudados e, portanto, tornam-se viáveis em ambientes embarcados. Desse modo, foi estabelecida uma rede neural baseada na aprendizagem por correção de erros, seguindo a topologia *Feedforward* com três camadas de neurônios. Portanto, além das conexões de entrada e saída tradicionais, houve a necessidade de implementação de uma camada oculta, de tamanho definido empiricamente. Foi prevista a utilização da biblioteca de código aberto FANN⁴, disponível em diversas linguagens de programação, para a implementação da rede. Para tal, a compatibilidade das rotinas nela implementadas com as plataformas ARM teve que ser verificada, logo, alterações de funções tornaram-se necessárias. A biblioteca FANN, dentro de suas funcionalidades, abrange dois conjuntos de dados para funcionamento: o primeiro contendo parâmetros da rede (*netlist*), que armazena informações responsáveis por caracterizar uma determinada rede já treinada; e os arquivos com padrões de aprendizado (*dataset*), contendo as relações entre neurônios de entrada e saída. As informações associadas a estes dados e outros aspectos da rede estão dispostos com maiores detalhes nos tópicos de codificação e, adiante, no treino e operação da rede neural artificial.

4.2.3 Codificação

As amplitudes relacionadas aos sinais filtrados são geralmente elevadas e não uniformes, portanto, não se adequam as funções de ativação dos neurônios da rede. Ademais, a possibilidade das entradas estarem dispostas em diferentes unidades e magnitudes eleva a memória requerida pela rede, ocasiona atrasos no processo de treino e resulta em maiores erros do preditor (SOLLA e SEVILLA, 1997). Assim sendo, de modo a reduzir o esforço computacional, otimizar a memória alocada e adequar os valores de entrada a rede neural, uma codificação foi aplicada aos dados. Neste contexto, transformação e a normalização são dois métodos de pré-processamento amplamente utilizados. A transformação envolve a manipulação de entradas de dados brutos para criar uma única entrada para uma rede, enquanto a normalização é uma transformação realizada em uma única entrada de dados

⁴ <http://leenissen.dk/fann/wp/>

para distribuí-los uniformemente e dimensioná-lo em um intervalo aceitável para a rede (MENDELSON, 1993). Sugeriu-se realizar então a normalização dos dados de entrada dos sensores flexivos, conforme técnica descrita por Solla e Sevilla (1997) em seu estudo. Deste modo, valores provenientes dos sensores flexivos foram divididos pelo valor máximo de sua variação, o que corresponde a redimensionar os dados aplicando a escala linear de 12 bits do canal ADC. A realização de ensaios sobre os gestos propostos resultou nos dados da Tabela 6. Conforme indicam Hagan et al. (1996), constitui uma boa prática associar as entradas e saídas do sistema aos neurônios das respectivas camadas da rede. Assim, foram previstos 5 neurônios de entrada, um para cada canal de aquisição dos sensores flexivos. Os sinais provenientes do acelerômetro, por sua vez, não tiveram influência sobre a rede neural, e dentro do contexto proposto, serviram apenas de base para a distinção de gestos estáticos similares, ao exemplo da relação estipulada pela Tabela 5.

Tabela 6 – Normalização realizada sobre os dados dos sensores de entrada, após realização de ensaios aplicando o alfabeto LIBRAS.

Caractere	Valor AD					Valor Normalizado				
	Polegar	Indicador	Médio	Anelar	Mínimo	Polegar	Indicador	Médio	Anelar	Mínimo
A	652	4008	3991	3488	2313	0,159179	0,978516	0,974365	0,851562	0,564697
B	1358	1256	691	642	676	0,331543	0,306641	0,168701	0,156738	0,165039
C	648	2811	3331	2164	674	0,158203	0,686279	0,813232	0,52832	0,16455
D	652	1048	3251	2302	678	0,15918	0,255859	0,793701	0,562012	0,165527
E	1668	3965	3928	4064	1807	0,407227	0,968018	0,958984	0,992236	0,441162
F	647	2686	621	638	678	0,157959	0,655762	0,151611	0,155762	0,165527
G	644	941	3951	3446	2258	0,157227	0,229736	0,9646	0,841309	0,55127
H	640	1139	1366	3420	2143	0,15625	0,278076	0,333496	0,83496	0,523193
I	714	3979	3915	3012	671	0,174316	0,971436	0,955811	0,735352	0,163818
J	714	3979	3915	3012	671	0,174316	0,971436	0,955811	0,735352	0,163818
K	640	1139	1366	3420	2143	0,15625	0,278076	0,333496	0,83496	0,523193
L	854	861	3976	3445	2169	0,208496	0,210205	0,970703	0,841064	0,529541
M	643	1417	650	637	1582	0,156982	0,345947	0,158691	0,155518	0,38623
N	648	1034	769	3162	1859	0,158203	0,252441	0,187744	0,771973	0,453857
O	642	3445	3469	2332	675	0,156738	0,841064	0,846924	0,569336	0,164795
P	640	1139	1366	3420	2143	0,15625	0,278076	0,333496	0,83496	0,523193
Q	644	941	3951	3446	2258	0,157227	0,229736	0,9646	0,841309	0,55127
R	650	1012	733	2951	1909	0,158691	0,247073	0,178955	0,720459	0,466064
S	1359	4095	4007	3360	1936	0,331787	0,999756	0,978271	0,820313	0,472656
T	648	3138	623	645	674	0,158203	0,766113	0,1521	0,157471	0,164551
U	648	1034	769	3162	1859	0,158203	0,252441	0,187744	0,771973	0,453857
V	644	1099	621	2851	1393	0,157227	0,268311	0,151611	0,696045	0,340088
W	643	1417	650	637	1582	0,156982	0,345947	0,158691	0,155518	0,38623
X	1145	3042	3994	3557	2215	0,279541	0,742676	0,975098	0,868408	0,540771
Y	639	3619	3815	2956	681	0,156006	0,883545	0,931396	0,72168	0,16626
Z	645	1401	3983	3516	2118	0,157471	0,342041	0,972412	0,858398	0,51709
REPOUSO	647	1320	1445	803	673	0,157959	0,322266	0,352783	0,196045	0,164307

Fonte: O autor (2017).

Na etapa posterior, após a operação da rede, foi prevista uma codificação que relacionasse os dados de saída a valores pré-definidos para a caracterização dos gestos. Desse modo, uma codificação em 32 bits foi proposta, abrangendo os 26 caracteres do alfabeto somados a um gesto criado para o repouso do usuário. Neste modelo, diferente-

mente do proposto por Sekar et al. (2014) e Quiapo e Ramos (2016), cada caractere foi representado por uma potência de 2, ou seja, um bit nível alto sobre a sequência (codificação *dummy*). Apesar do elevado número de dados sobre esta, a ocorrência de múltiplos zeros possibilitou significativa redução de erros durante o treinamento e tratamento da resposta de classificação em *firmware*, quando comparados as referências, uma vez que elevou a probabilidade de acerto do caractere ao limitar apenas um valor do vetor de saída a um nível alto, corroborando com a codificação sugerida por McCaffrey (2013) para características categóricas. Relacionados a RNA, cada um dos neurônios de saída foi associado a um dos 32 bits da sequência, de modo a otimizar a rede ao limitar os valores a dois estados possíveis. As 5 camadas extras, não relacionadas a um gesto específico, não representaram qualquer esforço computacional extra pelo classificador, uma vez que possuíam valor nulo. Com a codificação aplicada, a resposta do classificador foi constituída da posição do vetor de neurônios da camada de saída que obtivesse o maior valor sobre os outros, caracterizado nesta condição como nível alto. A Tabela 7 relaciona as letras do alfabeto aos valores definidos para a rede neural. A ocorrência de gestos similares indica um mesmo código e, portanto, maior número de exemplos durante o processo de treinamento.

Tabela 7 – Codificação para a saída do classificador.

Caractere	Posição do bit																																
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	
A	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
E	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
F	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
G	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
H	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
I	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
J	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
K	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
L	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
M	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
N	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
O	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
P	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Q	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
S	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
U	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
V	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
Z	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
Y	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
Z	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
REPOUSO	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Fonte: O autor (2017).

4.2.4 Sistema de Arquivos e Armazenamento USB

Devido a quantidade elevada de parâmetros relacionados a configuração da rede neural, relativos ao número de neurônios, camadas e padrões de aprendizado, uma memória considerável necessita ser alocada. Ademais, limitações de memória da plataforma escolhida indicaram a necessidade de solução para o armazenamento de parâmetros da rede neural externas a memória *flash* do microcontrolador.

Sendo assim, foi adicionado ao projeto um sistema de arquivos baseado em *File Allocation Table* (FAT), em conjunto com *drivers* USB. Foi proposta a utilização da biblioteca de Chan⁵, desenvolvida para sistemas embarcados com recursos restritos. Devido a limitações das funções observadas nesta, quando comparadas as funcionalidades de acesso a arquivos da biblioteca C padrão, ocorreu o desenvolvimento e adequação de funções necessárias para as operações de armazenamento de dados da rede neural. As rotinas USB por sua vez, foram implementadas em modo *host*, para que a interface entre o sistema de arquivos e o dispositivo de armazenamento externo fosse realizada. Com isso, houve a possibilidade de estabelecer redes mais complexas e de alterar parâmetros de forma externa.

4.2.5 Treinamento e Operação da RNA

O processo de treinamento e operação pode ser abordado de duas maneiras, relativas a metodologia aplicada e ao fluxo de rotinas desenvolvido em *firmware*, ambas descritas nos tópicos a seguir.

4.2.5.1 Metodologia de Treinamento

Baseada nas referências apresentadas, utilizou-se o *backpropagation* como algoritmo de treino, devido a sua acuracidade e facilidade de aplicação. Apesar de possuir desvantagens em relação a outras técnicas, como tempos elevados de treino e a recorrência de mínimos locais, estas são consequências apenas de *datasets* numerosos e problemas relativamente complexos (VORA e YAGNIK, 2014). Como afirma Hagan et al. (1996), a arquitetura varia de acordo com a complexidade do problema e não pode ser definida antes do treinamento, constituindo uma procura baseada em tentativas e erros. Portanto, a partir dos resultados obtidos com esta técnica, poderiam ser avaliadas novas soluções caso estas não fossem satisfatórias. Inicialmente foi aplicado o método *batch* do algoritmo, porém testes posteriores com o modelo *incremental* resultaram em erros menores para um mesmo período de treinamento. Para aplicações com reconhecimento de padrões, geralmente são aplicadas funções de ativação sigmóide (HAGAN et al., 1996). Direcionada a biblioteca FANN, sugeriu-se o modo *stepwise* da função, caracterizada por implementar

⁵ http://elm-chan.org/fsw/ff/00index_e.html

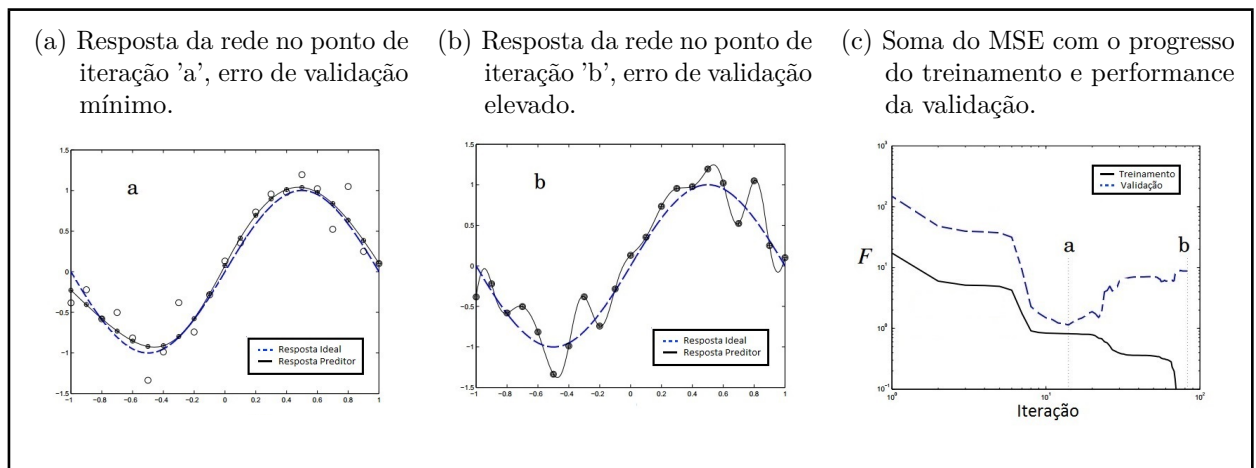
uma aproximação linear da original, tendo em vista a redução do período de treino. Como consequência, um aumento mínimo no MSE foi observado, embora não tenha o inviabilizado para uso na aplicação. Os valores relacionados a taxa e momento de aprendizagem foram definidos empiricamente e posteriormente reajustados conforme análise sobre a resposta do preditor. Algumas arquiteturas realizam o ajuste fino destas taxas ao decorrer do processo de treino, porém estas não foram aplicadas neste trabalho. O número de neurônios da camada oculta, de mesmo modo, foi determinado de forma empírica em um primeiro momento. Como informa Hagan et al. (1996), a determinação deste parâmetro pode ser realizada ao determinar um valor entre o número de neurônios de entrada e saída. Posteriormente, aumenta-se o número de neurônios da camada oculta até a obtenção de um resultado que atinja os objetivos do sistema proposto, sem degradação do MSE da rede.

Durante o treinamento de redes artificiais multicamadas, esta pode por vezes memorizar o *dataset* de treino e falhar ao não reconhecer novos dados. Esta situação é demonstrada por Hagan et al. (1996) na Figura 23. A linha contínua demonstra a resposta da rede ao treinamento, em termos de MSE, enquanto a linha tracejada exhibe o comportamento da rede obtido ao aplicar uma sequência de teste. O treino em excesso do primeiro direciona a rede a uma solução ótima para o *dataset*, no entanto, ocasiona perda de generalização: a apresentação de novos dados (não treinados) a rede resultam em altos índices de erro do preditor. Desta forma, a região de treinamento em que o erro da sequência de treino é mínimo e o de teste é elevado caracteriza o chamado *overfitting*. De modo a evitar a perda de generalização da rede, sugeriu-se realizar o método de validação cruzada indicado por Hagan et al. (1996), ao qual se concentra em aplicar uma parada antecipada ao treino. O treinamento foi realizado sobre um *dataset* específico de estimação e, a cada intervalo pré determinado de épocas, avaliada a soma do MSE da rede através de um *dataset* de teste. Dessa forma, o momento que ocorreu o menor erro de validação entre as iterações, correspondeu as épocas necessárias para aplicar a parada antecipada ao treinamento, resultando no ponto ótimo de generalização da rede. A Tabela 8 demonstra os parâmetros finais aplicados, a partir das considerações trabalhadas, que resultaram em um erro aceitável para a aplicação. A Figura 24 por sua vez demonstra o comportamento da RNA implementada, através da relação do MSE e número de épocas iteradas. A convergência da rede neural frente aos dados de treinamento e validação pode ser observada através da linha tracejada: no limite desta curva o MSE de validação tende a ser elevado com o número de treinos, indicando que o treinamento excessivo, mesmo que aplicado com a análise resultante da validação cruzada, tende a reduzir a performance do preditor. O menor erro sobre esta curva indicou o número de épocas necessárias para o comportamento ótimo da rede. Nesse ponto, os parâmetros da rede foram transferidos da RAM ao arquivo *netlist* (apêndice D). Por fim, foi avaliado se a rede apresentava valores aceitáveis, caso contrário, o treinamento poderia ser realizado novamente com diferentes

parâmetros. Conforme ressaltam Hagan et al. (1996), esta análise possibilita a descoberta de problemas com os dados, arquitetura e algoritmo de treino pois, dessa forma, o processo como um todo é iterado até que a resposta da rede seja satisfatória.

As construção dos *datasets* para o treinamento da rede foi realizada em duas etapas. Na primeira, foram adquiridos os sinais para o treino da rede, através da média relativa a 5 ensaios depurados do *firmware*, baseados nas posições específicas da LIBRAS conforme disposto pela Tabela 6. Em seguida, a partir dos dados já adquiridos, foi empregada a técnica de aumento do *dataset* (do inglês, *data augmentation*), utilizada por Wang e Perez (2017), de larga utilização em redes neurais para classificação de imagens. A partir deste conceito, foi aplicada uma variação aleatória com valor máximo de 5% sobre os dados originais, de modo a simular as variações de leitura observadas nas aferições dos canais AD do microcontrolador. Com isso foi possível obter uma larga quantia de exemplos para o *dataset*, conforme necessidade. Partindo da relação entre os dados de estimação e validação indicados por Haykin (2001), 80% dos dados de treinamento corresponderam ao subconjunto de estimação e os 20% restantes ao dados de validação. Ao fim do processo de treinamento, um total de 50 conjuntos foi utilizado. Os *datasets* de treino e validação constam parcialmente nos apêndices B e C.

Figura 23 – Exemplo de resposta obtida com o método de validação cruzada.



Fonte: Adaptado de Hagan et al. (1996).

4.2.5.2 Algoritmo de Treinamento e Execução

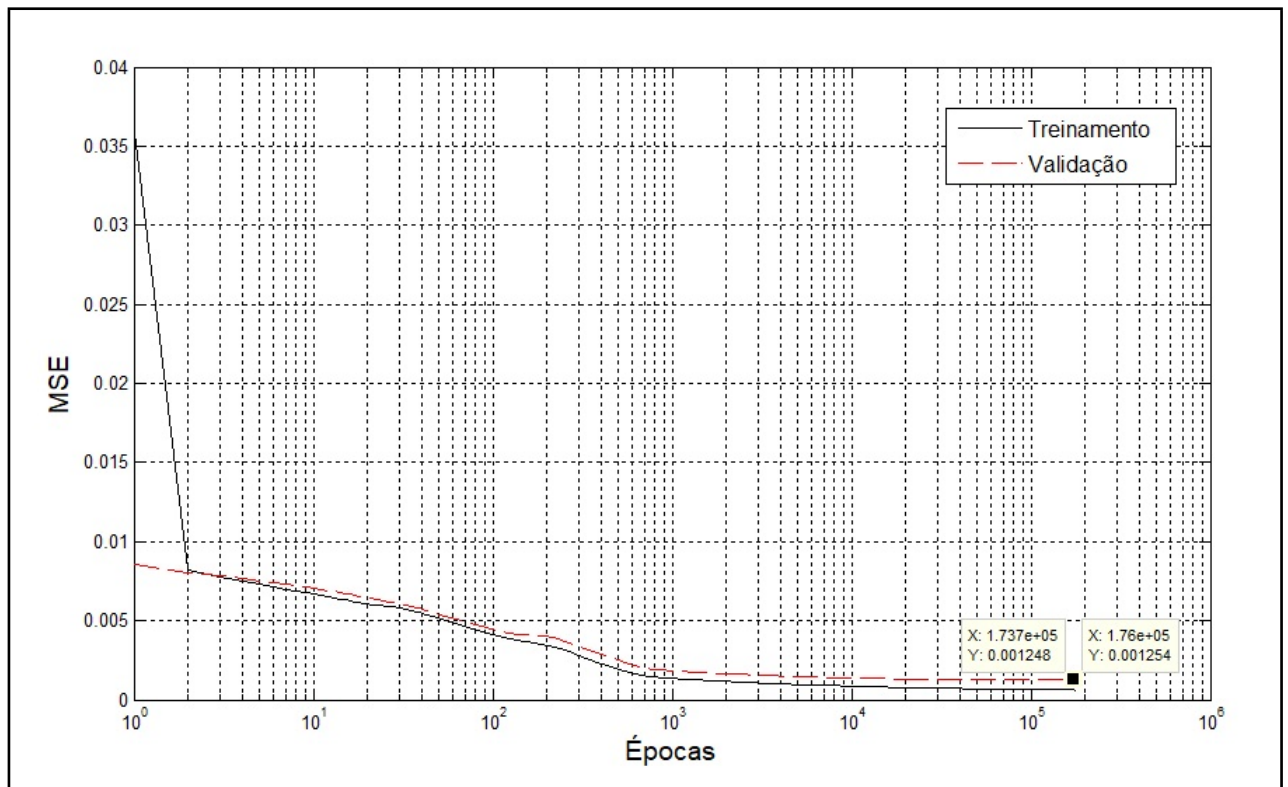
Os processos de treinamento e execução das lógicas da RNA, dentro do escopo do *firmware*, tem sua operação em dois momentos distintos. A rotina de treinamento tem início com a verificação da unidade USB pelo arquivo de parâmetros da rede (*netlist*) e os arquivos com padrões de aprendizado (*dataset*). Caso o arquivo *netlist* seja encontrado, os dados são carregados para a memória RAM do microcontrolador e o treino é iniciado.

Tabela 8 – Parâmetros da rede neural artificial aplicada.

Função de Ativação Neurônios Camada Oculta	Sigmóide <i>Stepwise</i>
Função de Ativação Neurônios Camada Saída	Sigmóide <i>Stepwise</i>
Algoritmo de Treino	<i>Incremental Backpropagation</i>
Taxa de Aprendizado	0.1
Momento de Aprendizado	0.0
Número de <i>Epochs</i> mínimo	10000
Camadas	3
Nuerônios de Entrada	5
Neurônios Ocultos	32
Neurônios de Saída	32

Fonte: O autor (2017).

Figura 24 – Convergência da rede neural artificial implementada. MSE mínimo atingido de 0,1248% após 173700 épocas.

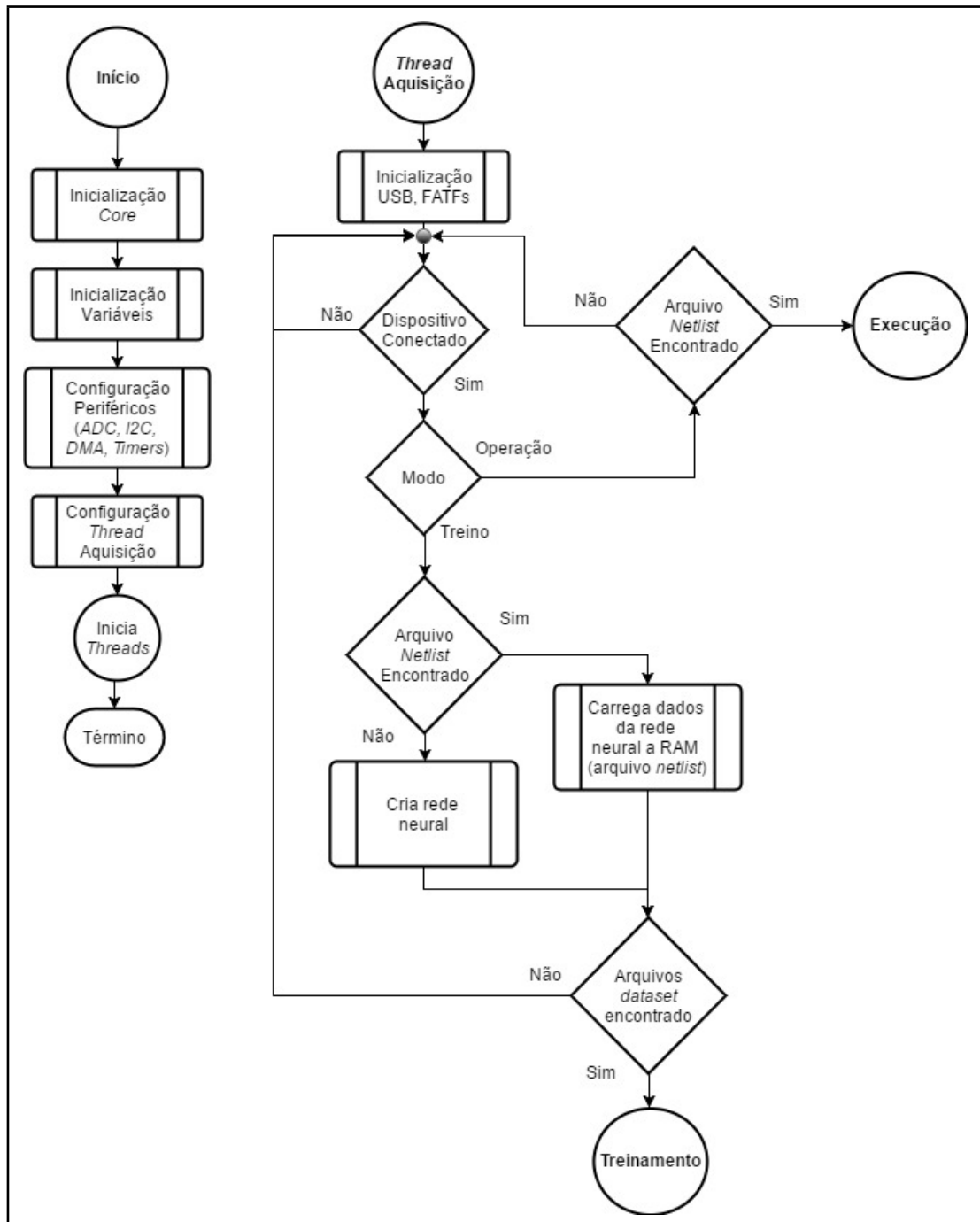


Fonte: O autor (2017).

Caso contrário, a rede é criada com base em parâmetros pré definidos. Isso permite a realização de novos treinamentos sobre uma rede já estabelecida, em casos onde é desejado alterar parâmetros para comparação de resultados. Após o treinamento da rede, os dados são armazenados no arquivo *netlist* para serem carregados a rede no momento da execução. Esta por sua vez inicia ao ocorrer a leitura do arquivo *netlist* do dispositivo externo, quando esta opção é determinada pelo usuário. Após ser bem sucedida, os dados da rede são

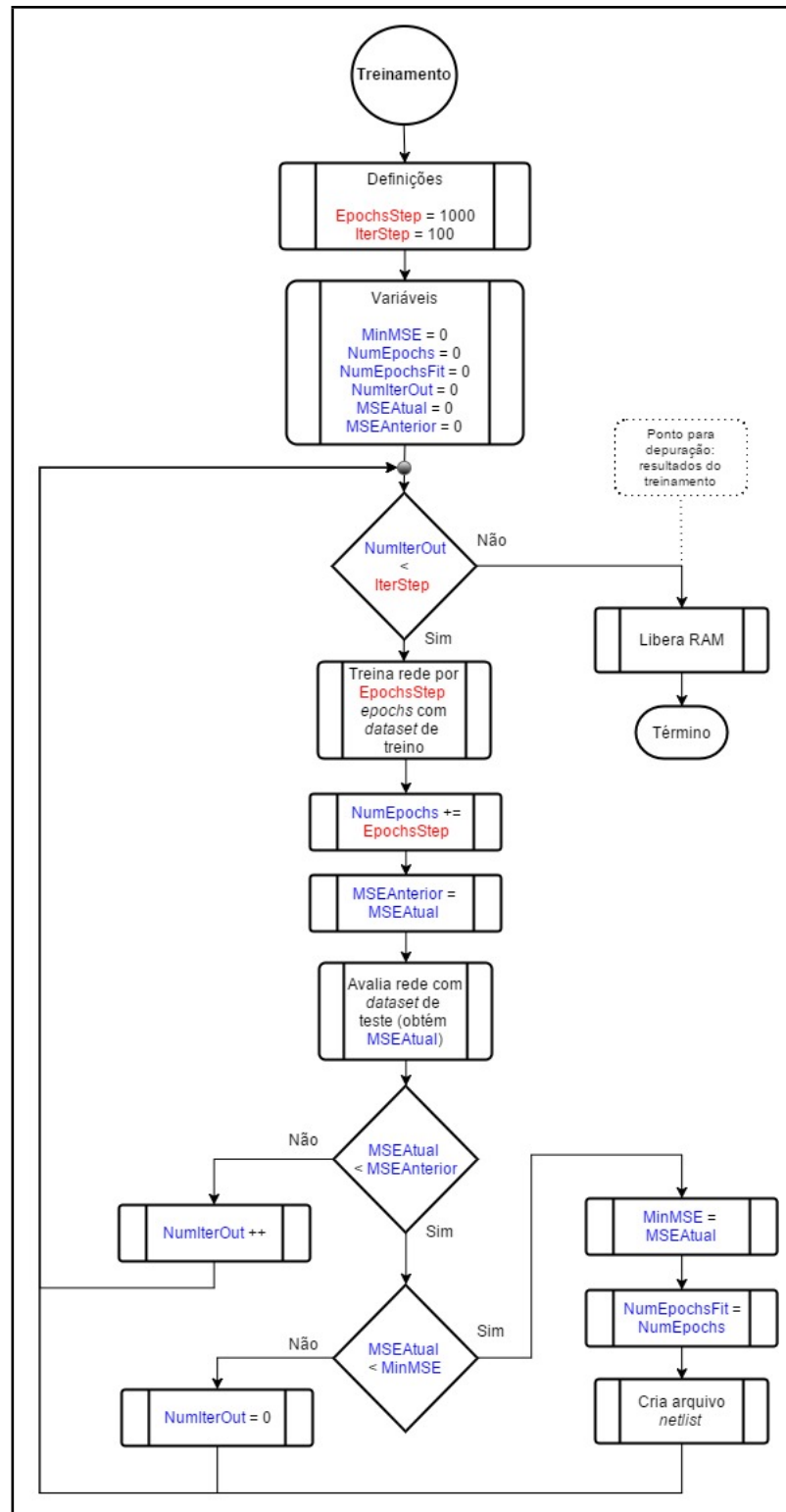
transferidos a RAM, os periféricos inicializados e os sensores calibrados. As aquisições são então realizadas indefinidamente e, a resposta da rede, na condição de protótipo, pode ser observada via depuração do *firmware*. A Figura 25 demonstra o processo de inicialização do *firmware* desenvolvido, enquanto as Figuras 26 e 27 indicam as etapas de treinamento e execução da rede, respectivamente.

Figura 25 – Fluxograma simplificado do *firmware* (inicialização).

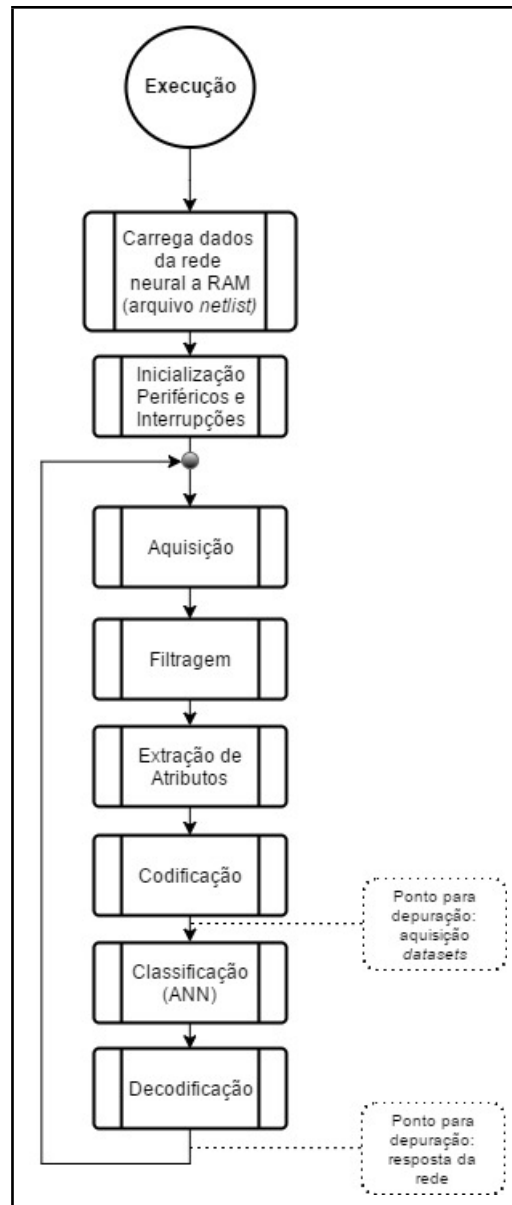


Fonte: O autor (2017).

Figura 26 – Fluxograma simplificado do *firmware* (treinamento).



Fonte: O autor (2017).

Figura 27 – Fluxograma simplificado do *firmware* (execução).

Fonte: O autor (2017).

5 Resultados

Neste capítulo, estão descritos os resultados obtidos com o protótipo desenvolvido, com base nas definições atribuídas na metodologia proposta. As atividades contidas nesta etapa contribuem aos objetivos levantados e servem de base para comparação aos trabalhos relacionados.

5.1 Detecção de Gestos e Resultados da Classificação

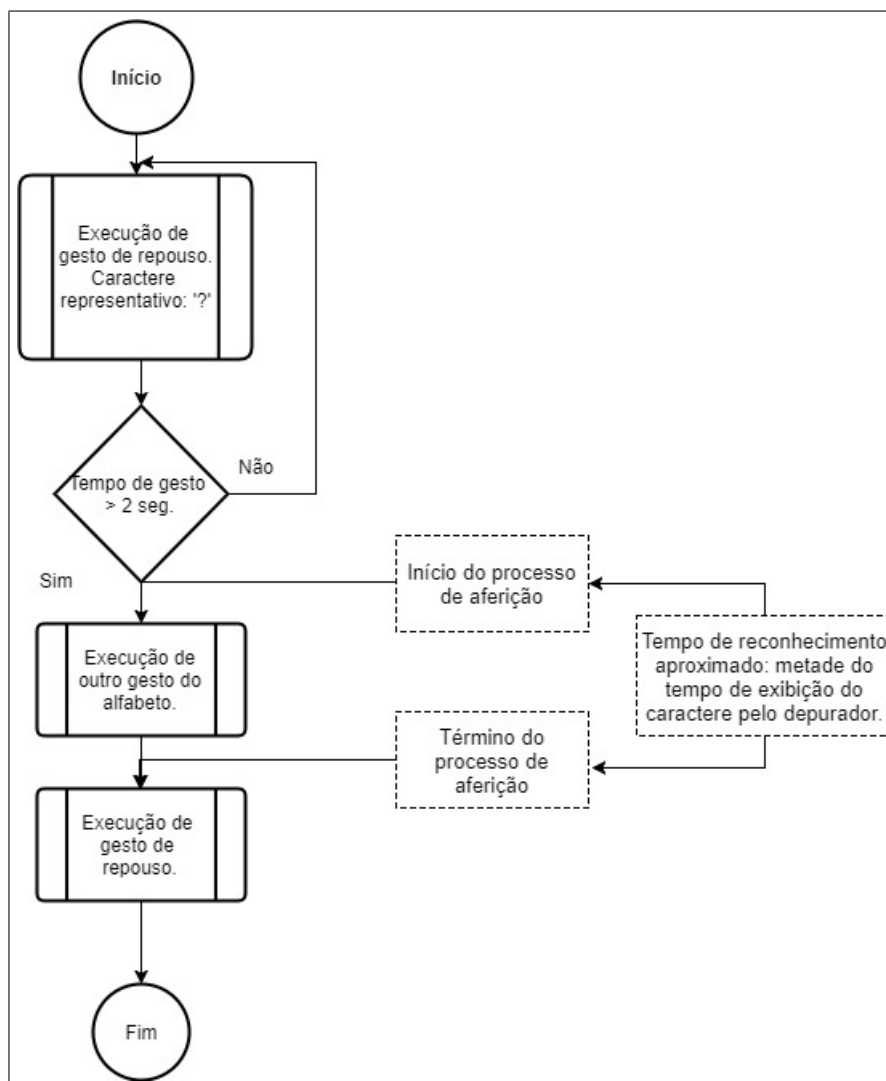
Os primeiros ensaios sobre o protótipo finalizado foram realizados com o objetivo de validar o funcionamento do sistema proposto. Assim, foi realizada a verificação da classificação de gestos pela execução destes em diferentes momentos. Esta constituiu uma importante etapa de desenvolvimento do protótipo, uma vez que permitiu melhorias significativas acerca da taxa de reconhecimento gestual e dos parâmetros de classificação.

Posterior a validação dos símbolos, foi aferido o tempo de reconhecimento de gestos médio: um fator importante quando considerada a aplicação real do protótipo. Esta foi calculada com base no tempo de detecção de um gesto qualquer pelo *firmware*, quando este foi aplicado entre duas posições gestuais de referência. O fluxograma da Figura 28 exemplifica os passos do processo realizado. Foi aplicado inicialmente o gesto de repouso, fixando-o por cerca de 2 segundos. Deste modo, procurou-se garantir que o gesto estava detectado e estacionário, antes de sua transição, garantindo maior confiabilidade na aferição do tempo de reconhecimento do gesto posterior. Em seguida, aplicou-se o gesto sobre ensaio rapidamente, retornando a posição de repouso logo em seguida. Desta forma, com o auxílio do registros de dados (do inglês, *data logs*) do modo de depuração do *firmware*, foi possível obter o tempo de permanência do gesto específico sobre a detecção do classificador. Considerando então que houve duas transições de gesto, a aproximação foi considerada como a metade do tempo aferido. A taxa de reconhecimento média aferida entre dois símbolos foi de 150ms para gestos estáticos e 400ms para os dinâmicos, atingindo em média a capacidade de detecção de 3 símbolos por segundo.

5.2 Acuracidade da Técnica Proposta

De modo a realizar a validação do sistema em condições reais de operação, testes foram conduzidos com dois voluntários portadores de deficiência relacionada a fala e, portanto, usuários da LIBRAS. O teste foi restringido a aplicação de gestos isolados, ou seja, não sequenciais, com o intuito de simplificar a avaliação da acuracidade do sistema de classificação sem a necessidade de segmentação gestual. A partir desta definição, foi

Figura 28 – Fluxograma da lógica de teste elaborada para realizar a aproximação do tempo de reconhecimento de gestos.



Fonte: O autor (2007).

solicitado aos voluntários a execução de cada um dos gestos do alfabeto em momentos distintos, partindo sempre da posição de repouso a outra determinada. Este passo foi repetido 10 vezes pelos voluntários, para todos os símbolos relacionados na Tabela 5. O resultado dos ensaios pode ser observado na Tabela 9, junto aos estudos de Carneiro et al. (2016) e Pathak et al. (2007) para comparação. Foi observado que alguns símbolos apresentaram acuracidade significativamente inferior a média do conjunto, calculada em 85,5%. Este fato foi relacionado a baixa variabilidade dos valores aferidos pelos sensores quando em posições similares, como ocorre nas diferenciações entre os caracteres R, U e V ou até mesmo F e T. Em contrapartida, 78% dos gestos obtiveram níveis de acuracidade acima da média, indicando resultados satisfatórios mesmo quando comparados ao estudo de Carneiro et al. (2016), voltado ao alfabeto LIBRAS. Ao relacioná-lo ao trabalho de Pathak et al. (2007), baseado no *American Sign Language* (ASL) do anexo E, o trabalho

proposto apresentou resultados similares, tendo em consideração a utilização de um número inferior de meios aquisitivos, além da proposta de um alfabeto com gestos mais complexos. O tempo de reconhecimento de símbolos se mostrou inferior aos trabalhos avaliados, ainda que se tenha utilizado um meio de classificação com maior demanda computacional.

Tabela 9 – Resultados da classificação do sistema proposto junto aos estudos de Carneiro et al. (2016) e Pathak et al. (2007).

Estudos	Carneiro et al. (2016)	Pathak et al. (2007)	Sistema Proposto (2017)
Aquisição	Visão (Kinect)	Sensorial (Flexivo, Inercial e Contato)	Sensorial (Flexivo e Inercial)
Classificador	Auto Valores e Distância Euclidiana	Tabela de Comparação	RNA
Tempo de Reconhecimento Médio	Não informado.	500ms	150ms
Alfabeto	LIBRAS	ASL	LIBRAS
Caractere	Acuracidade Média (%)	Acuracidade Média (%)	Acuracidade Média (%)
A	90	100	100
B	100	100	100
C	100	90	100
D	80	90	100
E	90	90	90
F	80	100	100
G	90	100	100
H	-	100	70
I	100	90	100
J	-	90	100
K	-	80	95
L	100	100	60
M	80	80	100
N	80	80	100
O	100	90	100
P	100	90	95
Q	80	100	100
R	100	100	10
S	70	90	90
T	80	90	40
U	90	80	100
V	80	80	0
W	90	100	100
X	-	90	90
Y	-	100	90
Z	-	80	80
REPOUSO	-	-	100
Média	89	91,54	85,56

Fonte: O autor (2017).

6 Considerações Finais

O desenvolvimento do presente trabalho possibilitou a detecção e classificação de gestos do alfabeto LIBRAS, com a implementação de um protótipo baseado em métodos de aquisição sensoriais e classificação por RNA aplicados sobre uma plataforma microcontrolada, com resultados comparáveis as referências adotadas. Além disso, permitiu um amplo entendimento sobre as tecnologias aplicadas neste campo de estudo, as dificuldades de pessoas com estas limitações e as oportunidades de trabalho na área de reconhecimento gestual. Foram abordados diversos aspectos no decorrer do estudo, como a caracterização do gesto, métodos de aquisição, classificação, codificação, rotinas de programação e detalhes de *hardware*, continuamente fundamentados em referências bibliográficas e trabalhos recentes na área. Os resultados obtidos com o protótipo desenvolvido se mostraram alinhados ao objetivo proposto, diferenciando-se das referências ao utilizar um método de classificação com maior confiabilidade sobre uma plataforma com menor poder de processamento. Apesar de alguns símbolos apresentarem baixa acuracidade de reconhecimento, os resultados foram satisfatórios uma vez que se mostraram similares aos trabalhos referenciados. Ademais, a classificação de gestos dinâmicos do alfabeto LIBRAS, não sugeridos em uma primeira avaliação, elevaram a acuracidade geral do sistema, mesmo em vista do maior número de gestos dinâmicos deste quando comparados ao ASL.

Conforme tratado no decorrer do trabalho, a validação do sistema ocorreu ao apresentar o protótipo desenvolvido a pessoas com deficiência da fala. Estas demonstraram interesse na aplicação e reforçaram a importância do desenvolvimento de dispositivos voltados a acessibilidade. Durante sua experiência com o protótipo, consideraram sua utilização simplificada e funcional para diversas atividades diárias de comunicação, em situações relacionadas a pessoas que desconhecem a linguagem LIBRAS. No entanto, salientaram a perda de praticidade ao se deparar com um método baseado na comunicação por caracteres em vez de palavras, este último dado por gestos do braço e expressões faciais. A taxa de reconhecimento atingida se mostrou, em média, adequada para o nível de destreza apresentado pelos voluntários, viabilizando a aplicação do protótipo para situações reais de conversação.

Almeja-se com o desenvolvimento deste trabalho contribuir para o avanço científico no que pauta a acessibilidade de pessoas com recursos limitados relacionados a fala, tendo em vista o aumento da qualidade de vida ao facilitar sua interação com a sociedade através da disposição de um protótipo móvel e eficiente para a detecção de gestos específicos. O protótipo ainda possibilita a utilização por quaisquer outras aplicações que demandem aquisição de sinais e classificação em tempo real de variáveis físicas, onde não há possibilidade ou intenção de se utilizar arquiteturas com maiores níveis de processamento.

A base do protótipo desenvolvido sugere melhorias acerca de seu método funcional, de modo a agregar qualidade e confiabilidade ao trabalho final. Entre as principais melhorias previstas, relacionadas a aplicabilidade do protótipo para o público, observa-se a necessidade do desenvolvimento de lógicas para segmentação de gestos, além da inclusão de um módulo para a transmissão sem fio da classificação a dispositivos externos. A inclusão de sensores adicionais para a remoção de ambiguidades do alfabeto LIBRAS, melhorias na fixação do *hardware* a luva, o estudo dos componentes utilizados e sua eficiência relativos ao consumo quando em uso contínuo ou até mesmo a utilização de diferentes plataformas microprocessadas para a obtenção de menores períodos de classificação, também constam entre as sugestões do trabalho descrito.

Referências

- ALON, Jonathan; ATHITSOS, Vassilis; YUAN, Quan; SCLAROFF, Stan. **A Unified Framework for Gesture Recognition and Spatio Temporal Gesture Segmentation**. IEEE Transactions of Pattern Analysis and Machine Intelligence (PAMI). 2009.
- CHENG, Hong; YANG, Lu; ZICHENG, Liu. **Survey on 3D Hand Gesture Recognition**. IEEE Transactions on Circuits and Systems for Video Technology. Vol.26, NO.9. 2016.
- BOBIC, Vladislava; TADIC, Predrag; KVASCEV, Goran. **Hand Gesture Recognition Using Neural Network Based Techniques**. 13th Symposium on Neural Networks And Applications (NEUREL), SAVA Center. Belgrade, Serbia. 2016.
- CARNEIRO, Sérgio Bessa; SANTOS, Edson D.F. de M.; Barbosa, Talles M. de A.; FERREIRA, José O.; ALCALÁ, Symone G. Soares; ROCHA, Adson F. da. **Static Gestures Recognition for Brazilian Sign Language with Kinect Sensor**. IEEE Sensors. Goiás, Brasil. 2016.
- VARSHINI, Lavanya; C.M., Vidhyapathi. **Dynamic Fingure Gesture Recognition using KINECT**. International Conference on Advanced Communication Control and Computing Technologies (ICACCCT). Vellore, Índia. 2016.
- MEKALA, Priyanka; FAN, Jeffrey; LAI, Wen-Cheng; HSUE, Ching-Wen. **Gesture Recognition Using Neural Networks Based on HW/SW Cosimulation Platform**. Advances in Software Engineering. Taipei, Taiwan. 2013.
- LAVANYA, Vajjarapu; M.S., Akulapraavin; MOHAN, Madhan. **Hand Gesture Recognition And Voice Conversion System Using Sign Language Transcription System**. IJECT, Vol.5. Odalarevu AP, Índia. 2014.
- SEKAR, Harini; C., Rishikanth; RAJESH, Ramesh; VIJAYARAGHAVAN, Vineeth. **Low-Cost Intelligent Gesture Recognition Engine for Audio-Vocally Impaired Individuals**. IEEE Global Humanitarian Technology Conference. Chennai, Índia. 2014.
- MULAY, Gururaj; YEMBARWAR, Akshay. **A novel authentication method for security systems using multidimensional characteristics of cost-effective flex sensors**. Maharashtra Institute of Technology, IEEE. Pune. 2014.

HAYEK, Hanine El; NACOUZI, Jessica; KASSEM, Abdallah; HAMAD, Mustapha; EL MURR, Sami. **Sign to Letter Translator System using a Hand Glove**. Notre Dame University-Louaize, IEEE. Zouk Mikayel, Lebanon. 2014.

PATHAK, Vishal; MONGIA, Sushant; CHITRANSHI, Gaurav. **A Framework for Hand Gesture Recognition based on fusion of Flex, Contact and Accelerometer Sensor**. IEEE Third International Conference on Image Information Processing. Uttar Pradesh, India. 2015.

HAVALAGI, Praveenkumar S.; NIVEDITA, Shruthi Urf. **The Amazing Digital Gloves That Give Voice to The Voiceless**. International Journal of Advances in Engineering & Technology (IJAET). Bijapur, India. 2013.

QUIAPO, Carlos Emmanuel A.; RAMOS, Katrina Nicole M. **Development of a Sign Language Translator Using Simplified Tilt, Flex and Contact Sensor Modules**. IEEE Region 10 Conference (TENCON). Quenzon City, Philippines. 2015.

XU, Ruize; ZHOU, Shengli; LI, Wen J. **MEMS Accelerometer Based Nonspecific User Hand Gesture Recognition**. IEEE Sensors Journal. Vol.12, NO.5. 2012.

TRINDADE, Pedro Emanuel dos Santos Vale Sousa. **Using Distributed Accelerometers for Gesture Recognition and Visualization**. Faculty Of Science And Technology, Department Of Electrical and Computer Engineering. 2010.

LAM, Alan H. F.; LI, Wen J.; LIU, Yunhui; XI, Ning. **MIDS: Micro Input Devices System Using MEMS Sensors**. Dept. of Electrical and Computer Eng., Michigan State University. USA. 2002.

GUPTA, Hari Prabhat; CHUDGAR, Haresh S.; MUKHERJEE, Siddhartha; DUTTA, Tanima; SHARMA, Kulwant. **A Continuous Hand Gestures Recognition Technique for Human-Machine Interaction Using Accelerometer and Gyroscope Sensors**. IEEE Sensors Journal. Vol.16, NO.16. 2016.

XIE, Renqiang; CAO, Juncheng. **Accelerometer-Based Hand Gesture Recognition by Neural Network and Similarity Matching**. IEEE Sensors Journal. Vol.16, NO.11. 2016.

JI, Zhe; LI, Zhi Yi; LI, Peng; AN, MaoBo. **A New Effective Wearable Hand Gesture Recognition Algorithm With 3-axis Accelerometer**. 12th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD). Beijing, China. 2015.

MARQUES, Gorka; BASTERRETXEA, Koldo. **Efficient Algorithms for Accelerometer-based Wearable Hand Gesture Recognition Systems**. IEEE 13th International Conference on Embedded and Ubiquitous Computing. Basque Country, Spain. 2015.

SIDEK, Othman; HADI, Munajat Abdul. **Wireless Gesture Recognition System Using MEMS Accelerometer**. International Symposium on Technology Management and Emerging Technologies (ISTMET). Bandung, Indonesia. 2014.

HSU, Yu Liang; CHU, Cheng Ling; TSAI, Yi Ju; WANG, Jeen Shing. **An Inertial Pen With Dynamic Time Warping Recognizer for Handwriting and Gesture Recognition**. IEEE Sensors Journal. Vol.15, NO.1. 2015.

BARCZEWSKA, Katarzyna; DROZD, Aleksandra. **Comparision of methods for hand gesture recognition based on Dynamic Time Warping algorithm**. IEEE Proceedings of the 2013 Federated Conference on Computer Science and Information Systems. pp. 207-210. 2013.

ZHOU, Yinghui; JING, Lei; WANG, Junbo; CHENG, Zixue. **Analysis and Selection of Features for Gesture Recognition Based on a Micro Wearable Device**. International Journal of Advanced Computer Science and Applications (IJACSA). Vol.3, NO.1. 2012.

PRADHAN, Ashis. **Support Vector Machine - A Survey**. International Journal of Emerging Technology and Advanced Engineering. Vol 2, Issue 8. 2012.

HUANG, Guang Bin; ZHU, Qin Yu; SIEW, Chee Kheong. **Extreme learning machine: Theory and applications**. School of Electrical and Electronic Engineering, Nanyang Technological University. Singapore. 2006.

RAKTHANMANON, Thanawin; CAMPANA, Bilson; MUEEN, Adbullah; BATISTA, Gustavo; WESTOVER, Brandon; ZHU, Qiang; ZAKARIA, Jesin; KEOGH, Eamonn. **Searching and Mining Trillions of Time Series Subsequences under Dynamic Time Warping**. Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM. 2012.

GERSHENSON, Carlos. **Artificial Neural Networks for Begginers**. Neural and Evolutionary Computing. Cornell University. 2003.

VORA, Kuldeep; YAGNIK, Shruti. **A Survey on Backpropagation Algorithms for Feedforward Neural Networks**. International Journal of Engineering Development and Research (IJEDR), ISSN:2321-9939, Vol.1, Issue 3, pp.193 - 197, Gujarat, India 2014.

SAYAD, Saed. **Artificial Neural Network**. Disponível em: <http://www.saedsayad.com/artificial_neural_network.htm>. Acesso em: 29 de maio de 2017.

SMITH, Steven W. **Digital Signal Processing: A Practical Guide for Engineers and Scientists**. Elsevier Science, 2013.

HAGAN, M.T.; DEMUTH, H.B. & BEALE, M. **Neural network design**. PWS Publishing Company. Boston, 1996.

FELIPE, Tanya A.; MONTEIRO, Myrna S. **Libras em Contexto: Curso Básico: Livro do Professor**. 6ª ed. Brasília: Ministério da Educação, Secretaria de Educação Especial, 2006.

MENDELSON, Louis B. **Preprocessing Data For Neural Networks**. Technical Analysis of Stocks & Commodities, 1993. Disponível em: <<https://www.tradertech.com/mendelsohn/library/neural-networks>>. Acesso em: 29 de maio de 2017.

SOLA, J; SEVILLA, J. **Importance of Input Data Normalization for the Application of Neural Networks to Complex Industrial Problems**. IEEE Transactions On Nuclear Science, Vol.44, NO.3, 1997.

MOREIRA, Michel Castro; CECÍLIO, Roberto Avelino; PINTO, Francisco de Assis de Carvalho; PRUSKI, Fernando Falco. **Desenvolvimento e Análise de Uma Rede Neural Artificial Para Estimativa da Erosividade da Chuva Para o Estado de São Paulo**. R. Bras. Ci., 2006.

HAYKIN, Simon. **Redes neurais: Princípios e prática**.. Porto Alegre: Bookman, 2ª Ed. 2001.

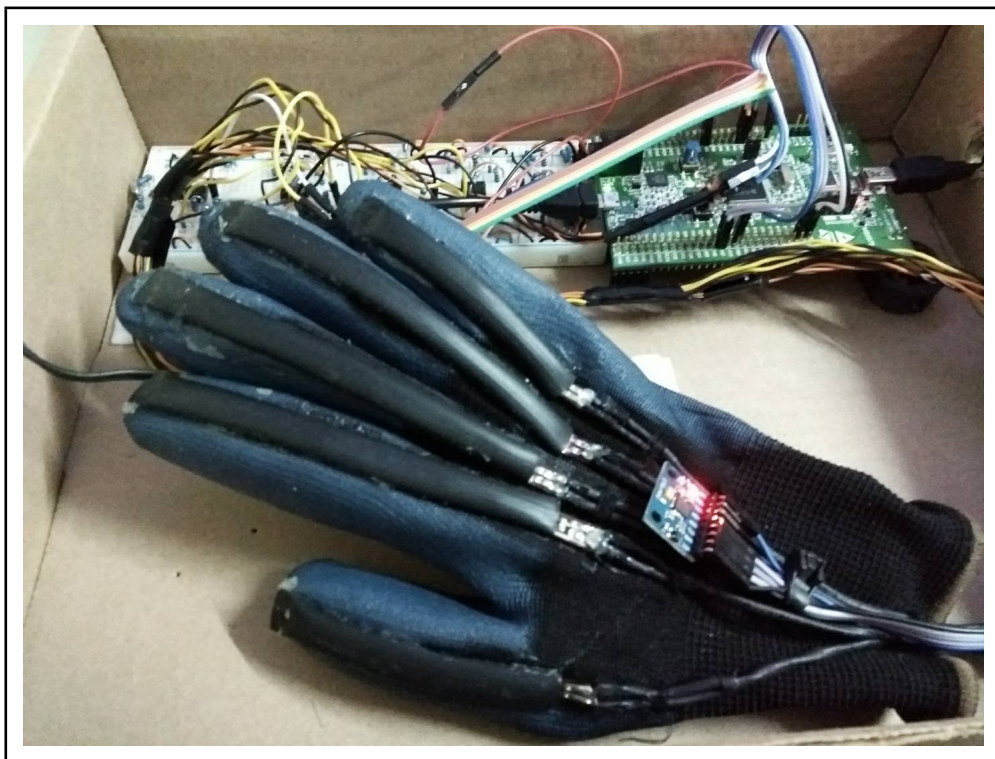
WANG, Jason; PEREZ, Luiz. **The Effectiveness of Data Augmentation in Image Classification using Deep Learning**. Stanford, 2017. Disponível em: <<http://cs231n.stanford.edu/reports/2017/pdfs/300.pdf>>. Acesso em: 01 de novembro de 2017.

MCCRAFFEY, James; **Neural Network Data Normalization and Encoding**. Visual Studio Magazine, 2013. Disponível em: <<https://visualstudiomagazine.com/articles/2013/07/01/neural-network-data-normalization-and-encoding.aspx>>. Acesso em: 10 de dezembro de 2017.

APÊNDICE A – Protótipo Desenvolvido

Imagem do protótipo com sensores flexivos e inerciais fixados, junto ao circuito de instrumentação.

Figura 29 – Protótipo desenvolvido.



Fonte: O autor (2017).

APÊNDICE B – *Dataset* (parcial) de Treino Sensor Flexivo

Arquivo criado pelo autor, a partir dos dados aferidos dos sensores ao aplicar os gestos da LIBRAS, para o treinamento da RNA. Arquivo em forma parcial, para exemplificação, devido ao volume elevado dos dados.

```
1080 5 32
0.160200 0.973600 0.977700 0.872500 0.583100
1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0.328400 0.321600 0.171900 0.155700 0.171700
0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0.157000 0.667600 0.833600 0.539500 0.156400
0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0.166500 0.267500 0.826200 0.558400 0.162000
0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0.407900 0.986800 0.984400 0.971200 0.427600
0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0.156200 0.658900 0.157400 0.154400 0.158000
0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0.157400 0.226900 0.921000 0.857000 0.563000
0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0.155400 0.266200 0.326400 0.803900 0.508200
0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0.182000 0.998700 0.974600 0.699800 0.156500
0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0.177700 0.952600 0.924900 0.757800 0.171500
0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0.152600 0.265100 0.323100 0.803000 0.519900
0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0.216800 0.220500 0.976400 0.821600 0.536300
0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0.162700 0.334700 0.158300 0.148000 0.387800
0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0.161200 0.259800 0.181500 0.809500 0.443100
0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0.156600 0.856800 0.813300 0.588200 0.164100
```

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0.155700 0.277300 0.321500 0.873300 0.512800
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0.153000 0.241000 0.974600 0.815600 0.543600
 0 0 0 0 0 0 1 0
 0.154900 0.246500 0.187800 0.707600 0.478600
 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0
 0.325900 1.000000 0.984100 0.843900 0.478500
 0 1 0 0 0 0 0 0 0 0 0 0 0 0
 0.158300 0.762500 0.153100 0.154600 0.169500
 0 1 0 0 0 0 0 0 0 0 0 0 0
 0.160000 0.263800 0.190500 0.800600 0.441200
 0 1 0 0 0 0 0 0 0 0 0 0
 0.155500 0.275700 0.155100 0.700000 0.339100
 0 1 0 0 0 0 0 0 0 0 0
 0.149500 0.356700 0.155400 0.162200 0.368200
 0 1 0 0 0 0 0 0 0 0
 0.288800 0.746400 0.969300 0.892700 0.530900
 0 1 0 0 0 0 0 0 0
 0.152500 0.907800 0.894600 0.755000 0.169000
 0 1 0 0 0 0 0 0
 0.150700 0.343000 1.000000 0.845600 0.493800
 0 1 0 0 0 0 0
 0.156700 0.337600 0.369800 0.193200 0.161100
 0 1
 0.152200 0.989600 0.949200 0.814000 0.562000
 1 0
 0.326000 0.299100 0.167300 0.161000 0.172200
 0 1 0
 0.157000 0.708200 0.848800 0.552700 0.169000
 0 0 1 0
 0.164900 0.244300 0.828200 0.582200 0.161100
 0 0 0 1 0
 0.398600 0.993700 0.931700 1.000000 0.433400
 0 0 0 0 1 0
 0.150200 0.651800 0.157700 0.152600 0.169900
 0 0 0 0 0 1 0
 0.152900 0.239400 0.924100 0.854200 0.563800 ...

APÊNDICE C – *Dataset* (parcial) de Teste Sensor Flexivo

Arquivo criado pelo autor, a partir dos dados aferidos dos sensores ao aplicar os gestos da LIBRAS, para a validação do treinamento da RNA. Arquivo em forma parcial, para exemplificação, devido ao volume elevado dos dados.

```
270 5 32
0.162200 0.950600 1.000000 0.891000 0.575000
1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0.317900 0.299700 0.161200 0.151400 0.167600
0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0.152800 0.653200 0.809100 0.547400 0.171500
0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0.159000 0.255900 0.786700 0.554700 0.163400
0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0.393200 0.971500 0.977600 1.000000 0.457500
0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0.154800 0.685900 0.152400 0.160700 0.162100
0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0.160400 0.231000 1.000000 0.863600 0.547600
0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0.158100 0.283200 0.336100 0.861200 0.544300
0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0.177600 0.963200 0.951600 0.709500 0.162800
0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0.167400 0.966900 0.980100 0.736200 0.161200
0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0.155900 0.283400 0.333900 0.834100 0.536600
0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0.214800 0.201800 0.967300 0.854000 0.549000
0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0.156300 0.347300 0.151200 0.161600 0.383900
0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0.163400 0.262300 0.185800 0.777400 0.457800
0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0.153700 0.870500 0.868200 0.596800 0.162800
```


0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0.154300 0.284500 0.344300 0.847300 0.520700
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0.151900 0.237200 0.955100 0.849100 0.565200
 0 0 0 0 0 0 1 0
 0.161500 0.251900 0.175900 0.745600 0.476500
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0.330700 0.975200 0.999900 0.859800 0.457600
 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0
 0.160400 0.763300 0.156100 0.158000 0.161100
 0 1 0 0 0 0 0 0 0 0 0 0 0 0
 0.165100 0.247800 0.187000 0.736800 0.441600
 0 1 0 0 0 0 0 0 0 0 0 0 0
 0.153600 0.265700 0.149200 0.674600 0.339100
 0 1 0 0 0 0 0 0 0 0 0 0
 0.162400 0.353000 0.164100 0.159300 0.372600
 0 1 0 0 0 0 0 0 0 0 0
 0.271400 0.776700 1.000000 0.876800 0.516400
 0 1 0 0 0 0 0 0 0 0
 0.150000 0.890000 0.916900 0.687500 0.172800
 0 1 0 0 0 0 0 0 0
 0.163000 0.340700 0.954300 0.864000 0.535700
 0 1 0 0 0 0 0 0
 0.160600 0.317600 0.337800 0.204500 0.169600
 0 1
 0.155800 0.947800 1.000000 0.872800 0.578700
 1 0
 0.323300 0.301800 0.168500 0.161500 0.159800
 0 1 0
 0.161200 0.678000 0.777100 0.503700 0.161000
 0 0 1 0
 0.152500 0.252000 0.831100 0.553200 0.163100
 0 0 0 1 0
 0.410900 1.000000 0.973200 1.000000 0.421900
 0 0 0 0 1 0
 0.156100 0.684000 0.158300 0.156500 0.169100
 0 0 0 0 0 1 0
 0.160600 0.235100 0.987600 0.825900 0.525200 ...

APÊNDICE D – *Netlist* (parcial) do Sensor Flexivo

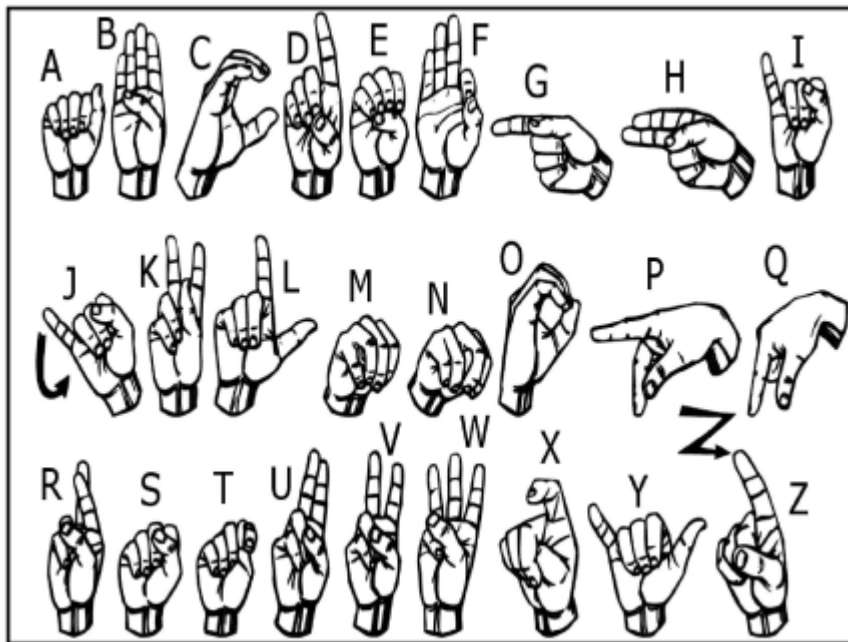
Arquivo gerado pela RNA após o treinamento, com as informações necessárias para a execução da rede. Arquivo em forma parcial, para exemplificação, devido ao volume elevado dos dados.

```
FANN_FLO_2.1
num_layers=3
learning_rate=0.100000
connection_rate=1.000000
network_type=0
learning_momentum=0.100000
training_algorithm=2
train_error_function=0
train_stop_function=0
cascade_output_change_fraction=0.010000
quickprop_decay=-0.000100
quickprop_mu=1.750000
rprop_increase_factor=1.200000
rprop_decrease_factor=0.500000
rprop_delta_min=0.000000
rprop_delta_max=50.000000
rprop_delta_zero=0.100000
cascade_output_stagnation_epochs=12
cascade_candidate_change_fraction=0.010000
cascade_candidate_stagnation_epochs=12
cascade_max_out_epochs=150
cascade_min_out_epochs=50
cascade_max_cand_epochs=150
cascade_min_cand_epochs=50
cascade_num_candidate_groups=2
bit_fail_limit=3.49999994039535520000e-001
cascade_candidate_limit=1.00000000000000000000e+003
cascade_weight_multiplier=4.00000005960464480000e-001
cascade_activation_functions_count=10
cascade_activation_functions=3 5 7 8 10 11 14 15 16 17
```


APÊNDICE E – Linguagem Americana de Sinais

Imagem representativa do alfabeto americano de sinais relacionada no estudo de Pathak et al. (2007).

Figura 30 – Linguagem de Sinais Americana



Fonte: Pathak et al. (2007).