Sistema de Reconhecimento das Configurações de Mãos da Língua Brasileira de Sinais por Redes Neurais Artificiais

Bianca Rodrigues de Castro
Instituto Federal de Educação, Ciência
e Tecnologia do Rio Grande do Sul - IFRS
Av. São Vicente, 785,
Farroupilha, RS, Brasil
Email: biancardc24@gmail.com

Matheus Antônio Corrêa Ribeiro
Instituto Federal de Educação, Ciência
e Tecnologia do Rio Grande do Sul - IFRS
Av. São Vicente, 785,
Farroupilha, RS, Brasil
Email: matheus.ribeiro@farroupilha.ifrs.edu.br

***Resumo*—A dificuldade na comunicação é um dos maiores empecilhos do cotidiano de algumas pessoas com deficiência auditiva. Pensando nisto, o presente trabalho propõe o desenvolvimento de um sistema de reconhecimento das configurações de mão da Língua Brasileira de Sinais, dentro da disciplina de Trabalho de Conclusão de Curso do curso de Engenharia de Controle e Automação do IFRS - Campus Farroupilha. Para tanto, faz-se a utilização de uma luva sensorial especializada na captação da configuração da mão e de algoritmos de Redes Neurais Artificiais para o reconhecimento das mesmas. Foi proposta a identificação das setenta e nove configurações de mão que são a base da língua, com uma abordagem inicial da identificação dos dezenove sinais estáticos que compõem o alfabeto. Através deste sistema foi possível obter um resultado de aproximadamente 76% de acurácia com o tempo de treinamento de 1,82s.**

***Palavras-chave*—Libras, Identificação de movimentos das mãos, Redes Neurais Artificiais, Luva sensorial especializada.**

I. Introdução

A comunicação é uma das necessidades humanas mais básicas e ela pode ocorrer através da fala, escrita ou de imagens. A língua de sinais é a principal forma de comunicação para algumas Pessoas com Deficiência - PcD. Ela é formada pela Configuração das Mãos - CM, Ponto de Articulação, Movimento e Orientação dos sinais e Expressões Faciais. Apesar de registros datarem o Período Imperial Brasileiro, quando em 1857 Dom Pedro II fundou o Colégio Nacional para Surdos, apenas no ano de 2002, com a criação de Lei 10.436, a Língua Brasileira de Sinais (Libras) foi reconhecida como língua oficial [1]. A partir deste momento, são oficializadas as regras e estruturas independentes que a Libras possui em relação à língua portuguesa. Por isso, este foi um passo importante para o desenvolvimento de comunicação por meio de Libras e sua disseminação no meio acadêmico.

Um sistema computacional capaz de reconhecer a língua de sinais e traduzi-la em texto ou fala pode facilitar a comunicação de pessoas surdas com aqueles que não tem conhecimento de Libras, aumentando a qualidade de vida destes indivíduos. Mas a aplicação do reconhecimento dos movimentos das mãos vai além do âmbito social.

Com a interação entre o homem e as máquinas cada vez mais presente no nosso dia a dia, surge uma necessidade cada vez maior pelo controle de dispositivos eletrônicos a partir da movimentação do corpo humano [2]. A utilização de movimento das mãos pode ser aplicada na transmissão de comandos, facilitando a comunicação homem-máquina. Também podem ser aplicados no controle de sistemas robóticos, na realidade virtual e em jogos virtuais [3].

O presente trabalho tem como objetivo apresentar um sistema de reconhecimento de configuração de mãos em Libras utilizando uma luva especializada para a detecção da posição dos dedos com a aplicação de Redes Neurais Artificiais (RNAs) para o processamento dos sinais captados. Este está organizado da seguinte maneira: na seção II é realizada a fundamentação teórica, abordando tópicos como o desenvolvimento da língua de sinais no Brasil, diferentes formas de captação do movimento das mãos e do papel das RNAs no reconhecimento e classificação das configurações de mão; na seção III são apresentados os materiais e métodos utilizados, definindo as partes integrantes de cada etapa do projeto; na seção IV são demonstrados os resultados obtidos e por fim, na seção V são apresentadas as considerações finais.

II. Revisão Bibliográfica

Para o melhor entendimento do assunto a ser abordado, essa seção está dividida de forma a contemplar separadamente todos os assuntos necessários para o entendimento do sistema proposto. Primeiramente aborda-se a Língua Brasileira de Sinais e todos os aspectos pertinentes à realização da proposta. Posteriormente são apresentadas as diferentes formas de captação das configurações de mão. Por último, apresenta-se o método de processamentos dos sinais captados para fins de reconhecimento dos mesmos.

A. LIBRAS

De acordo com a Organização Mundial da Saúde, 466 milhões de pessoas são surdos ou mudos em todo o mundo, isto equivale a 5% da população mundial (sendo 34 milhões

de crianças e 432 milhões de adultos). Estima-se que em 2050, uma a cada dez pessoas sofrerão de deficiência auditiva [4]. Um dos principais empecilhos enfrentados por essas pessoas está na dificuldade de comunicação. A língua de sinais é o seu principal meio de troca de informações, porém, ainda é pouco utilizada por pessoas que não possuem nenhum tipo de deficiência.

No Brasil, o Instituto Nacional de Educação de Surdos (INES) é reconhecido como centro de referência nacional na área da surdez e ocupa um importante papel na educação de surdos, tanto na formação de profissionais capacitados quanto na propagação de conhecimento desenvolvido neste âmbito. Ele foi criado em 1857 pelo francês E. Huet, que já havia atuado anteriormente no colégio francês Instituto dos Surdos-Mudos de Bourges [1]. Com o apoio do Governo Imperial, foi fundado o então Colégio Nacional para Surdos no Rio de Janeiro, que tinha como objetivo o ensino desde nível básico até o profissionalizante para a comunidade surda brasileira [1]. A língua praticada no Colégio tinhas fortes influências francesas, devido a nacionalidade de Huet. Ela foi propagada pelo Brasil por estudantes que retornavam a seus estados de origem após o fim de seus estudos [1].

O movimento de oficialização da Língua Brasileira de Sinais teve início na década de oitenta, seguindo o exemplo norte-americano. Em 1933, foi criado um projeto de Lei para a legalização e regulamentação da comunicação gestual. Esse processo só foi finalizado em 2002 com a criação da Lei 10.436 e do Decreto de número 5.626 em dezembro de 2005. Com isso, Libras passa a ser tratada como uma das línguas oficiais brasileiras, ganhando o caráter de componente curricular. O decreto também dispõe sobre o ensino desta como primeira língua para alunos surdos e fica regulamentado o ensino profissionalizante de professores bilíngues e tradutores.

A língua de sinais é formada por três parâmetros principais e dois secundários, são eles respectivamente:

- **Configuração das mãos:** trata-se da posição adotada pelos dedos, sendo que oficialmente são reconhecidos setenta e nove configurações possíveis pelo INES;
- **Ponto de articulação:** consiste no local de realização do sinal, podendo tocar parte do corpo ou ser realizado no espaço neutro, meio do corpo até a cabeça;
- **Movimento:** é o deslocamento das mãos durante a realização dos sinais, podendo ser retilíneo, circular, semicircular, helicoidal, sinuoso ou angular, conforme apresentado na Figura 1;
- **Orientação:** é a direcionalidade dos movimentos, e já que a mesma configuração e movimento com diferentes sentidos podem ter significados diferentes, ela é determinada pelo sentido apontado pela palma da mão durante a execução do sinal;
- **Expressão facial e corporal:** responsável por dar sentido aos sinais e ao tipo de frase (interrogação, afirmação, negação e exclamação).

Há três níveis de comunicação através de Libras: soletrar as palavras com as letras do alfabeto, palavras isoladas com significados completos e sinalização contínua para frases [6].

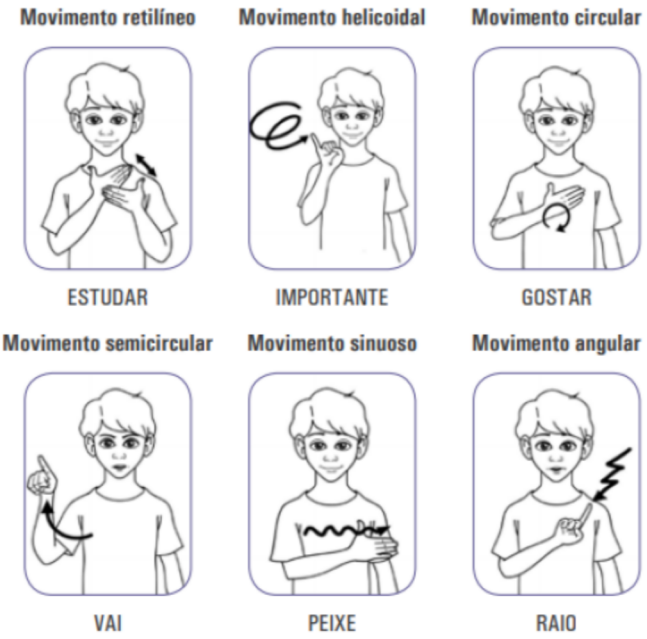


Figura 1. As possibilidades do parâmetro movimento [5].

Independente do nível de comunicação utilizado, Libras sempre seguirá os cinco parâmetros apresentados anteriormente. Assim como a língua portuguesa, ela apresenta variações regionais, mas qualquer sinal tem como base as possíveis configurações das mãos apresentados na Figura 2.

B. Captação de configuração de mão

Os sistemas de reconhecimento de movimentos das mãos são utilizados nas mais diversas aplicações. Por exemplo, na medicina é possível que estudantes simulem a realização de cirurgias mesmo não estando em um centro cirúrgico e sem colocar nenhuma pessoa em risco [2]. Outro exemplo está na operação de robôs sem necessariamente estar no mesmo ambiente que eles, podendo ser aplicado para tornar mais seguras as atividades de operadores em zonas de risco [7]. Ainda há a possibilidade de utilizar esses sistemas na captação, processamento e reconhecimentos nas mais diferentes línguas de sinais.

Grande parte dos sistemas de reconhecimento utilizam processamento de imagem baseado em sistemas de visão computacional [8]. Eles fazem uso de câmeras como principal fonte de aquisição de dados e apresentam uma taxa de assertividade superior a 90%. As principais desvantagens deste método estão nas influências causadas pelas condições do ambiente, como a luminosidade e o plano fundo onde as imagens são captadas e o alto custo computacional [2].Também existe a necessidade de materiais específicos onde a qualidade está ligada ao alto valor agregado, o que dificulta acesso para a maior parte da população, e na baixa confiabilidade para aplicações muito delicadas. Outra dificuldade está na impossibilidade de um sistema de aquisição em tempo real, pois seriam necessários

Figura 2. Tabela de Configuração das mãos do INES [1].

muitos equipamentos que não são comumente utilizados no dia a dia.

Outra forma de captação destes movimentos consiste em luvas especializadas equipadas com diversos tipos de sensores capazes de captar diretamente a variação da posição dos dedos [8]. Estes normalmente não sofrem influências de condições externas e podem ser facilmente transportados no cotidiano, porém a utilização de fios e sensores pode restringir a liberdade dos movimentos. Uma terceira opção consiste em um sistema híbrido na união das duas opções anteriores.

A proposta de [3] seria criar um dispositivo *wearable*, portátil e de fácil utilização no dia a dia, seja em ambientes fechados ou abertos, capaz de identificar os movimentos e dar uma resposta táctil quando aplicado à realidade virtual. Para isso foi utilizada uma luva especializada composta de acelerômetros, giroscópios de três eixos, magnetômetros e dispositivos táteis hápticos.

Já [8] propõe um sistema de identificação de gestos comumente utilizados na linguagem corporal não oficial. Para tanto utilizou uma luva composta de sensores flexíveis para a medição da flexão de cada um dos dedos, e de três acelerômetros, sendo um para cada eixo, e dois giroscópios fixados no pulso.

Fazendo uso da terceira opção apresentada como forma de captação de sinais, [2] cria um sistema híbrido para o reconhecimento da Língua Americana de Sinais. Foi desenvolvida uma luva composta por sete acelerômetros de três eixos para ser utilizada em sincronia com uma câmera RGB. O posicionamento dos sensores foi escolhido como sendo um posicionado em cada um dos dedos, outro no pulso e o último na parte superior do braço.

Todos os dispositivos apresentados possuem um condicionamento do sinal captado através de uma placa de aquisição e tratamento de sinais para depois serem enviados ao computador para o processamento.

C. *Processamento de sinais captados*

A necessidade por soluções onde a programação de resolução passo a passo não é suficiente vem crescendo cada vez mais a medida que os problemas abordados ficam mais complexos. A partir disto foram sendo desenvolvidos métodos de programação mais sofisticados e com uma menor dependência da intervenção humana. Para tanto, é necessário que esses métodos sejam capazes de criar sozinhos soluções para cada problema a ser resolvido, a partir de um conjunto de eventos similares já solucionados. Este processo é denominado Aprendizagem de Máquina (AM) [9].

A Rede Neural Artificial é uma das técnicas de AM utilizada para a criação de uma hipótese para a resolução de problemas a partir da minimização de uma função objetivo. Ela é baseada no funcionamento do sistema nervoso, e assim como ele, tem na base de seu funcionamento o neurônio [9]. É considerada uma técnica de aprendizagem supervisionada, tendo em vista que para a determinação de uma hipótese é alimentada com uma base de dados já previamente classificados [10]. Entretanto existem outros modelos de aprendizagem por reforço que não são superisionados.

O neurônio artificial é unidade fundamental de processamento. Os valores de entrada são recebidos e para cada um deles é atribuído um peso w. Posteriormente a combinação linear destes é processada através da função de ativação (f), então o resultado é enviado para a saída, podendo ir para uma entrada de outro neurônio ou não. Este funcionamento pode ser verificado na Figura 3.

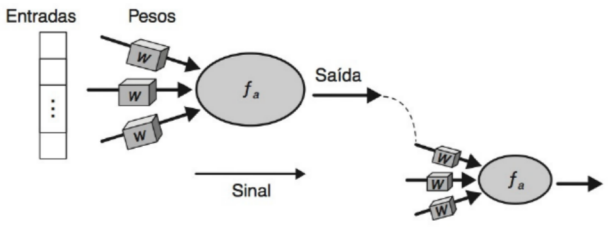


Figura 3. Representação do Neurônio Artificial [9].

Na Equação 1, o valor c é referente a combinação linear dos sinais de entrada de um neurônio, sendo x a representação dos sinais de entrada, w equivalente ao peso atribuído a cada sinal e b uma constante característica de cada neurônio.

$$c = b + \sum_{i=1}^{n} x_i w_i \tag{1}$$

A função de ativação recebe o valor de c e determina o comportamento do neurônio, conforme verificado nas Equações 3 e 4. O valor de saída do neurônio pode variar de acordo com as entradas recebidas e com as funções de ativação escolhidas. Apesar de existirem inúmeras funções de ativação na literatura [11], as mais comuns são as funções identidade, de limiar, sigmoidal, hiperbólica e ReLU, estas estão respectivamente descritas nas equações 2, 3, 4, 5 e 6. Isto se dá devido a facilidade de implementação, baixa demanda por capacidade computacional quando comparadas a outras funções e a variação conhecida dos valores de saída. Ambas as funções apresentam valores entre 0 e 1.

$$f(c) = c \tag{2}$$

$$f_a(c) = \begin{cases} 1, & \text{se } c < a \\ 0, & \text{se } c >= a \end{cases} \tag{3}$$

$$f_a(c) = \frac{1}{1 + e^{-c}} \tag{4}$$

$$f(c) = tanh(c) \tag{5}$$

$$f(c) = max(0, c) \tag{6}$$

A função limiar apresenta dois valores possíveis, 0 ou 1, sendo muitas vezes o 0 substituído pelo -1. Um valor mínimo é estipulado e sempre que a combinação dos valores de entrada forem maior que este limitante o neurônio é ativado. Na Equação 3, sempre que c for maior que o valor limitante a, o neurônio será ativado, ou seja, sua saída assumirá o valor de 1. Já a saída da função sigmoidal na Equação 4 apresenta diversos valores dentro da faixa especificada, de acordo com o valor c da combinação das entradas ponderadas.

A arquitetura de uma RNA é composta por vários neurônios, esses são agrupados em camadas. A camada de neurônios que recebe os valores dos atributos de entrada do sistema, ou seja, aqueles para serem classificados, é denominada camada de entrada. A camada que determina qual é a resposta final da classificação é denominada camada de saída. Já os neurônios que não estão nestas duas camadas, formam as camadas intermediárias ou ocultas [9]. Quanto maior é o grau de dificuldade do problema abordado, maior é a necessidade por neurônios. Porém a recomendação é que o valor mínimo para os neurônios destas camadas sigam a Equação 7, calculados a partir da quantidade de neurônios da camada de entrada e da camada de saída [10].

$$N_{oculta} = \frac{N_{entrada} + N_{saida}}{2} \tag{7}$$

As conexões entre os neurônios podem ser classificadas em completamente, parcialmente ou localmente conectada. No primeiro caso, os neurônios estão conectados a todos os que compõem a camada anterior e a próxima. Este geralmente é o mais utilizado. Já nos dois últimos casos, nem todos os neurônios trocam informações entre si [9].

Outra informação importante, além do grau de conectividade da rede, trata-se da possibilidade de utilização da retroalimentação, ou *feedback*. Nas RNAs o fluxo de informação geralmente ocorre no sentido unidirecional da camada de entrada para a de saída. Porém, quando a retroalimentação é inserida, um neurônio pode receber informações de outro da mesma camada ou de camadas posteriores como sinal de entrada [9].

O grau de conectividade, o número de neurônios e a presença de retroalimentação em uma RNA define sua topologia. Um exemplo pode ser verificado na Figura 4 , onde é utilizado um total de doze neurônios, duas camadas ocultas, completamente conectada e sem a presença de *feedback*, também chamada de rede *feedforward*.

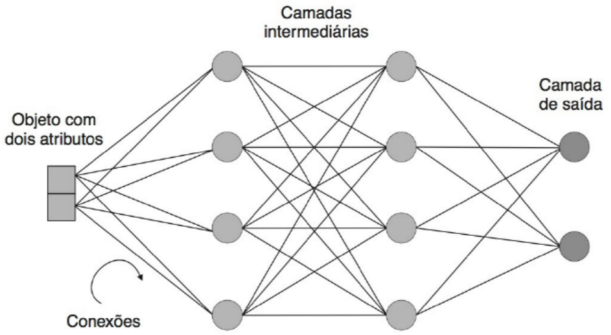


Figura 4. Exemplo de Rede Neural Artificial multicamadas *feedforward* [9].

Os algoritmos de RNAs tem como foco a correção de erros através da minimização da função objetivo, sendo esta uma métrica entre o erro das respostas geradas pelos neurônios da rede e a real classificação de cada caso [9]. Isto é feito por meio do ajuste dos pesos utilizados nas conexões da rede, de forma a reduzir a quantidade de erros cometidos. Este processo é chamado de treinamento de RNA.

Existem muitos algoritmos de determinação dos pesos ideais para uma RNA. O principais deles podem ser verificados na Tabela I, nela estão descritas suas principais vantagens e desvantagens.

Tabela I
VANTEGENS E DESVANTAGENS DOS DIFERENTES ALGORITMOS DE TREINAMENTOS [10].

Algoritmo	**Vantagem**	**Desvantagem**
Backpropagation	Simples e Eficiente	Necessária grande base de treinamento
Counterpropagation	Rapidez no treinamento	Topologia muito complexa
Hopfield	Implementação em larga escala	Sem aprendizagem, pesos estabelecidos
BAM	Estável	Pouco eficiente
Kohonen	Auto-organizável	Pouco eficiente

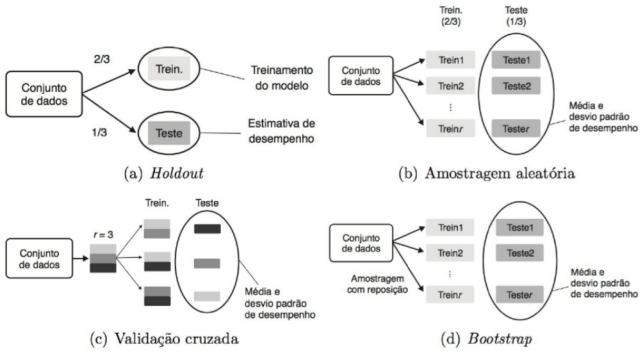
O algoritmo mais utilizado para o treinamento de uma Rede Neural Artificial é o *backpropagation*. Entre os principais motivos para a sua disseminação estão a sua tolerância à falhas, devido ao uso de computação local, e a possibilidade de ser utilizada em arquiteturas paralelas de modo eficiente [11].

O algoritmo *backpropagation* é baseado na regra do delta e no gradiente descendente, onde os pesos de cada um dos neurônios é atualizado de forma a obter o melhor desempenho para o problema apresentado. Este método é dividido em duas etapas, inicialmente cada atributo é enviado para a camada de entrada da rede, cada neurônio alimenta a entrada da próxima camada até a camada de saída classificar sua saída. A diferença entre o resultado esperado e obtido é tido como o erro cometido pela rede [9]. O ajuste destes pesos é feito a partir da derivada parcial da função objetivo, para que seja propagada a correção dos pesos até a camada de entrada.

O objetivo do algoritmo é localizar o mínimo global da função de objetivo em relação ao conjunto de dados apresentados aos neurônios. Para que esse algoritmo seja utilizado é necessário que a função de ativação seja contínua, diferenciável e preferencialmente não-decrescente [9].

Na aplicação de AM em problemas reais, o conhecimento que se tem domínio é proveniente de um único conjunto de dados o qual é utilizado para a determinação do modelo classificatório. Por isso é necessária a execução de testes para garantir a validade e a reprodutibilidade da classificação para dados que não participaram da base de treinamento [9].

Para os casos onde é existente uma única fonte de dados, tanto para o treinamento quanto para os testes, é necessário utilizar métodos de amostragem diferentes para obter estimativas de desempenho mais confiáveis [9]. Torna-se então necessária a definição de subconjuntos dentro da base de dados, um para treinamento e outro para testes. Os principais métodos de amostragem estão apresentados na Figura 5.



(a) *Holdout* (b) Amostragem aleatória

(c) Validação cruzada (d) *Bootstrap*

Figura 5. Métodos de amostragem da base de dados [9].

As principais características a serem avaliadas em um teste de capacidade de generalização de um modelo são a taxa de assertividade e o desvio padrão. Enquanto a taxa de assertividade diz o quão capaz de classificar dados fora da base de treinamento uma RNA é, o desvio padrão diz o quão sensível ela é a poucas alterações nos dados utilizados em seu treinamento [9].

Os métodos *Holdout* e Amostragem aleatória são indicados para uma grande base de dados. Já que a taxa de acerto para o total de objetos tende a ser maior do que a taxa para uma parte desses objetos [9]. A Validação cruzada ou *cross validation* é mais utilizada para a comparação entre diferentes algoritmos, já que determinam ambos os critérios de avaliação de desempenho com precisão e não está sujeita a variabilidade do tipo de dados separados. O método *Bootstrap* faz a separação aleatória da amostras para subconjuntos, logo um mesmo exemplo pode estar em mais de um subconjunto.

III. Materiais e Métodos

O desenvolvimento do sistema de reconhecimento de configurações de mão de Libras está dividido em duas etapas: a captação dos movimentos e o treinamento de uma Rede Neural Artificial para o reconhecimento dos mesmos.

As configurações de mão a serem executadas, para a formação da base e posterior identificação, fazem parte daquelas apresentadas na Figura 2. Porém, como é necessário uma grande quantidade de exemplos de cada movimento, foi optado em uma primeira abordagem apenas com os 19 sinais estáticos que compõem o alfabeto, estes estão destacados em vermelho na Figura 6.

Figura 6. Letras do Alfabeto em Libras [12] .

A. Captação de sinais e a formação da base de dados

Para a captação dos sinais provenientes das configurações de mão, que serão posteriormente utilizados como base para o treinamento da RNA, foi optado por uma luva sensorial especializada, desenvolvida para melhor atender as necessidades do projeto. Ela pode ser observada na Figura 7.

Ela é composta por cinco sensores do tipo flexível, apresentado na Figura 8, que são responsáveis por captar a variação na configuração dos dedos. Este tipo sensor varia a sua resistência de acordo com a distorção a que é submetida. O sensor utilizado tem sua faixa de variação de 10k à 50k Ohms.

Para a melhor identificação de alguns movimentos não captados pelos sensores flexíveis foram inseridos dois sensores de efeito hall. Um dos sensores foi inserido no polegar para

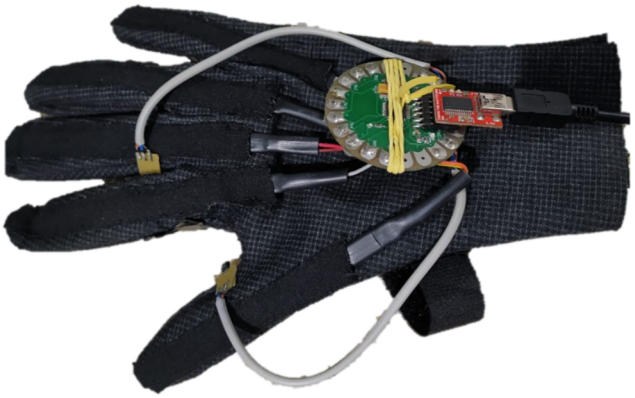


Figura 7. Luva de captação dos sinais dos movimentos de Libras.

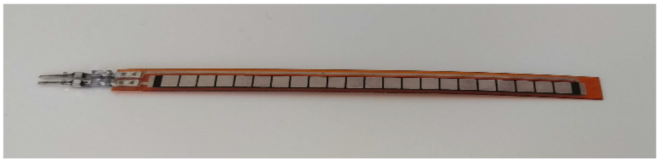


Figura 8. Sensor flexível de variação de resistência.

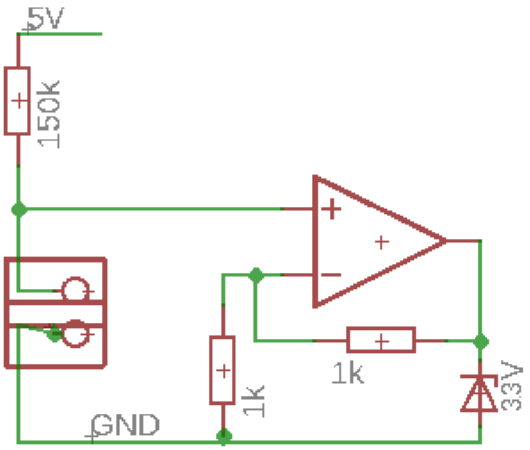


Figura 9. Circuito de condicionamento de sinais.

captar a abertura deste dedo em relação a mão. O outro sensor foi inserido no dedo médio para a captação do contato deste com o dedo indicador. Além disso, pequenos imãs foram distribuídos na região onde o sensor hall teria algum contato.

Os sinais provenientes dos sensores flexíveis passam por um circuito de condicionamento de sinais, que está demonstrado na Figura 9. Ele é composto por amplificadores operacionais na configuração não inversora de ganho igual à 2 e um diodo do tipo Zener. Com isso é possível aumentar a faixa de aquisição do conversor AD e limitar a tensão máxima de saída da placa em 3.3V, valor máximo de entrada para muitos microcontroladores. Porém, isso também gera um aumento na quantidade de ruído, apesar da alta impedância de entrada causada pelo amplificador. Posteriormente esse sinais são enviados para um microcontrolador Arduino lilypad, que possui um conversor AD de 10 bits e também a capacidade de realizar comunicação serial. A variação da tensão gerada pelos sensores flexíveis no circuito de condicionamento é de 0.8 a 2.8V.

Para a aquisição da base de dados foi optado por uma taxa de 1ms. Este tempo foi escolhido por possibilitar a realização de diversas leituras durante um único movimento. Esses valores posteriormente passam por um tratamento para a inserção na RNA. Além disso, foi estabelecida a utilização de uma comunicação do tipo serial entre o microcontrolador e o computador. A velocidade de transmissão a ser utilizada é de 38400bps, considerando que os valores dos sensores flexíveis serão transmitidos em 2 bytes, totalizando 13 bytes de dados a serem transmitidos a cada leitura, e que a taxa de transferência para o Arduino é de 34ms.

As leituras feitas pelos sensores só eram enviadas a partir do momento em que elas eram solicitadas por uma aplicação desenvolvida através do Matlab App Designer. O objetivo desta aplicação é ter uma interface para facilitar a visualização dos sinais captados pelos sensores, além de uma indicação visual do processo de aquisição de sinais das configurações de mão.

A interface desenvolvida pode ser observada na Figura 10. Na parte central dela é indicada qual a configuração de mão a ser feita e na parte inferior são retratados os sinais adquiridos atráves da luva. Além disso, devido aos sensores apresentarem pequenas variações de resistência dependendo das condições de temperaturas às quais estão sendo submetidos, foi optado pelo desenvolvimento de um sistema de calibragem.

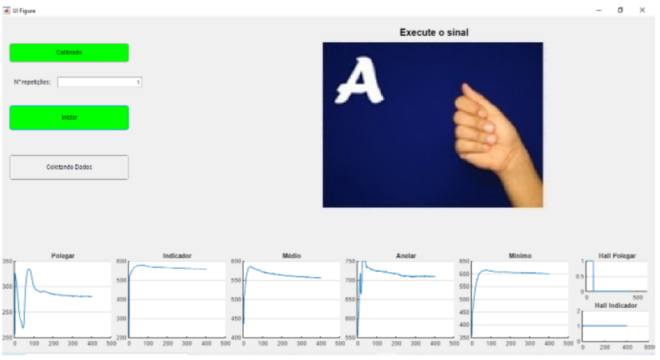


Figura 10. Aplicação desenvolvida para a captação de sinais.

Essa calibragem é feita no início de cada sequência de aquisição de sinais. Para isso, o portador da luva precisa ficar com a mão esticada sobre uma superfície plana por alguns segundos enquanto o sistema referência a variação no sinal de cada dedo como sendo aquela que deve ser encontrada durante a captura de sinais após a estabilização da mão, ou seja, após todo o movimento ser feito e esta se encontrar na configuração de mão pretendida.

Com base no resultado da calibragem é determinado um desvio padrão aceitável para cada sensor para leituras feitas na mão estática. Com base nestes números, é possível fazer com que o sistema entenda quando o portador da luva finalizou a movimenação de sua mão.

Devido a rede neural permitir apenas a entrada de um único dado em seus neurônios da camada de entrada, foi optado por alimentar a camada de entrada da RNA com a média dos últimos valores captados para cada configuração de mão executada. Essa escolha foi baseada no fato que nestas amostras já temos o movimento estabilizado e seriam as que melhor representariam aquela configuração.

Para a aquisição da base foram realizadas várias captações para cada uma das configurações de mão em dias e condições climáticas diferentes. Foram escolhidas dentre as amostras as captações mais consistentes, onde nenhuma configuração de mão foi executada da forma errada ou quando não houve nenhuma resposta inconsistente por parte da interface aos sinais enviados pela luva. Dentro destas aquisições, foram selecionadas 25 amostras de cada um dos sinais para a compor uma base.

Além disso, para melhor verificar a efetividade do algoritmo desenvolvido, esta base foi dividida entre base treinamento e base de teste. Com isto foi possível avaliar a capacidade de generalização da RNA para amostras que não foram utilizadas durante o treinamento. Na base de treinamento foi empregado o método de Validação Cruzada, onde o conjunto de dados é dividido em *R* subconjuntos de tamanho aproximadamente idênticos. As amostras dos subconjuntos *R-1* serão utilizadas no treinamento da RNA enquanto os demais serão utilizadas na validação de desempenho. Este procedimento de teste é repetido para cada um dos subconjuntos *R* e o resultado do treinamento é calculado pela média dos resultados obtidos em cada teste. Com isso, é possível avaliar o desempenho do modelo desenvolvido de forma mais eficaz, especialmente para conjuntos pequenos de dados, que é o caso das amostras utilizadas neste projeto. Foi optado por utilizar um *R* igual a 5. Sendo assim, das 25 amostras obtidas para cada uma das 19 configurações de mão 60% foram utilizadas para o treinamento da RNA, 20% para validação do modelo e 20% para os testes finais. Portanto, para as bases de treinamento e validação temos 380 amostras e para a base de teste 95.

B. Rede Neural Artificial para a determinação do modelo

A primeira etapa para o treinamento de uma rede neural é a preparação dos dados. Este pré-processamento é feito para eliminar a interferência de inconsistências presentes nos dados, como por exemplo valores faltantes, inconsistentes ou duplicados. Também pode ser feito o condicionamento dos dados para serem utilizados como entradas no treinamento da RNA. Como as amostras de configurações de mão de cada movimento se tornaram muito exaustivas, foi optado pela criação de bases sintéticas para serem utilizadas no treinamento da RNA.

Para a criação das bases sintéticas foi desenvolvido um algoritmo capaz de ler o valor das amostras da base original e para cada configuração de mão calcular os valores mínimos e máximos obtidos nas leituras de cada um dos sensores flexores. Com base nestes valores e adicionando uma tolerância de mais ou menos 10% são gerados valores aleatórios de amostras. Foram criadas bases sintéticas de 30, 40, 60, 100, 500 e 1000 amostras para cada configuração de mão e isto resulta nas respectivas bases 570, 760, 1140, 1900, 9500 e 19000 amostras.

Para o treinamento da RNA foi optado pela utilização do algoritmo de *backpropagation*. Essa escolha foi feita devido a capacidade de tolerância a falhas deste algoritmo e a facilidade de implementação. Também possui pequena sensibilidade a variação nos dados de treinamento. O algoritmo de *back-propagation* é composto de duas etapas: a propagação e a retropropagação. A propagação é sobre a classificação dos dados de entrada. Inicialmente os pesos iniciais da RNA são atribuídos de forma aleatória. Em seguida, os valores que representam cada sensor são inseridos nos neurônios da camada de entrada. Então o neurônio faz a ponderação com os pesos e o cálculo da saída através da função de ativação e transmite a saída para os neurônios da próxima camada. Esse processo continua até que todos os neurônios da camada de saída tenham uma resposta e que o movimento seja classificado como uma das letras estáticas do alfabeto. Posteriormente é calculado o erro cometido pela rede, através da diferença entre a resposta desejada e a resposta obtida para cada neurônio da camada de saída . O erro de um neurônio da camada de saída está demonstrado na Equação 8 o qual é definido pela função quadrática da diferença entre resposta desejada d e a resposta obtida y.

$$erro = \frac{1}{2}\sum_{q=1}^{k}(y_q - d_q)^2 \tag{8}$$

Após o cálculo do erro, é dado início a segunda etapa do algoritmo. A retropropagação consiste na correção dos pesos da RNA de acordo com o erro obtido. O ajuste dos pesos é iniciado na camada de saída e prossegue até a camada de entrada.

A Equação 9 demonstra como são ajustados os pesos no algoritmo. O novo peso de uma entrada *w(t+1)* é calculado através da soma do peso atual desta entrada *w(t)* com o coeficiente de ajuste. O coeficiente de ajuste é dado pelo coeficiente n, pelo erro associado ao neurônio d e pela a entrada recebida por este neurônio x [9].

$$w_{ij}(t+1) = w_{ij}(t) + nx_i d_j \tag{9}$$

Como o erro é calculado apenas para os neurônios da camada de saída, é necessário estimar o erro para as camadas intermediarias utilizando os erros da camada posterior. O erro da camada intermediaria é calculado pela soma do erro das camadas seguintes ponderados pelos pesos associados a cada neurônio [9].

O coeficiente de ajuste do erro do neurônio d pode ser calculado pela Equação 10. Para cálculos pertinentes à camada

de saída é utilizado a derivada da função de ativação e o erro de cada neurônio. Já para as camadas intermediarias é utilizada a derivada da função de ativação e o erro estimado das camadas posteriores ponderadas pelos respectivos pesos.

$$d_j(c) = \begin{cases} f'_a erro, & \text{se C} \in \text{Csaida} \\ f'_a \sum_{i=1}^{n} w_{ik} d_k, & \text{se C} \in \text{Cint} \end{cases} \quad (10)$$

Foi optado por uma topologia para a RNA completamente conectada, *feedforward* e de múltiplas camadas. Esse modelo foi determinado devido à complexidade do problema abordado e ao algoritmo de treinamento utilizado.

A camada inicial é formada pela mesma quantidade de dados de entrada, ou seja, para cada um dos sensores utilizados será adicionado um neurônio, totalizando sete neurônios, um para cada um dos sensores flexíveis e os outros dois para os sensores de efeito Hall. Já a camada de saída foi composta por um neurônio para cada uma das classificações possíveis, totalizando 19 neurônios para cada uma das configurações de mão, para a primeira abordagem.

Foi optado pelo desenvolvimento do algoritmo na linguagem de programação Python e execução do treinamento em um computador. A linguagem de programação Python possui inúmeras bibliotecas que facilitam a implementação de algoritmos de RNAs. Foi optado por construir a solução utilizando da biblioteca [13] que possui modelos de Redes Neurais Artificiais supervisionadas através do algoritmo de classificação *Multi-layer Perceptron (MLP)*. Este algoritmo permite a implementação de uma solução robusta a partir de poucas configurações.

Dentro das possíveis configurações oferecidas pela biblioteca [13], os seguintes parâmetros foram escolhidos para serem analisados: número de camadas ocultas, quantidade de neurônios nestas camadas, função de ativação das camadas ocultas, função de otimização de pesos da rede e o parâmetro de regularização. Na Tabela II é possível verificar os parâmetros e as variações empregadas neles durantes os testes.

Tabela II
PARÂMETROS ANALISADOS EM TESTES COM RNA.

Parâmetros	**Testes**	**Padrão da RNA**
Camadas ocultas	1 a 5	1
Quantidade de neurônios	13 a 130	100
Funções de ativação	Identidade eq.2, Sigmoidal eq.4 Hiperbólica eq.5 e ReLU eq.6	ReLU
Funções de otimização de pesos (solver)	LBFGS, sgd e adam	adam
Parâmetro de regularização	0,00001 a 0,01	0,0001

As funções de ativação apresentadas na tabela II são demonstradas nas equações de 2 até a 6. No MLP, além destas funções de ativação é utilizada na camada de saída a função Softmax, para incentivar os modelos de classificação multiclasse [13]. Para a otimização dos pesos foram utilizadas duas funções baseadas no cálculo da descida gradiente estocástico, SDG e ADAM, além de um otimizador na família de métodos quase-Newton, LBFGS.

Os testes iniciais foram feitos a partir das configurações padrões do algoritmo de classificação apresentados na Tabela II utilizando a base original com diferentes quantidades de amostras e também as bases sintéticas. Com isso, foi optado por dar continuidade aos testes com apenas uma das bases, aquela que apresentou a maior acurácia, já que o tempo para treinamento não teve um aumento considerável.

Para cada parâmetro a ser alterado foram feitos testes com a base que mostrou o melhor desempenho, partindo das configurações padrões da rede e alterando um único parâmetro a cada teste. Com isso foram escolhidos as configurações de cada parâmetro que obtiveram os melhores resultados e a partir disso foi optado por variar a combinação desses para encontrar a RNA que seja mais adequada a este sistema.

Após o treinamento da rede para todos os movimentos da base de treinamento, é iniciada a validação do modelo determinado. Para isso foram utilizadas as amostras da base de testes que foram submetidas a classificações pelos modelos resultantes do treinamento. A partir da comparação entre as respostas encontradas pela RNA e classe original das amostras é calculada a eficiência da rede.

IV. RESULTADOS

Os resultados obtidos neste trabalho podem ser divididos em três tópicos principais: o impacto das diferentes bases de dados utilizadas no treinamento da RNA, o ajuste de parâmetros para encontrar o melhor modelo e o modelo final obtido para a classificação das configurações de mão.

A. Impacto das bases de treinamento

As análises de resposta do modelo às diferentes bases de treinamento foram obtidas através dos parâmetros padrões conforme indicado na Tabela II. A base de amostras originais foi utilizada em diferentes quantidades de tamanhos 95, 190, 285 ou 380 amostras, com o propósito de avaliar o impacto do aumento de amostras por configuração de mão. Já as bases de dados sintéticas foram utilizadas apenas na sua integralidade. Com o modelo treinado e validado, foram executados testes de classificação em outra base. A partir destes procedimentos foram obtidas as acurácias de treinamento e de teste, as quais podem ser observado na Figura 11. As barras em verde (esquerda) são referentes às acurácias de treinamento enquanto as barras em azul (direita) são referentes às acurácias com a base de teste.

Nos treinamentos feitos com as bases de amostras originais podemos observar um excelente resultado durante os treinamentos, chegando próximo a 80%, conforme demostrado na Figura 11. Porém, este mesmo comportamento não se repete ao empregar os modelos na base de teste. Isto se deve à baixa capacidade de generalização dos modelos obtidos. Com uma amostragem muito pequena não é possível ter uma resposta satisfatória para amostras fora da base de treinamento, já que o modelo foi obtido partindo de pouca diversidade e tornou-se muito específico.

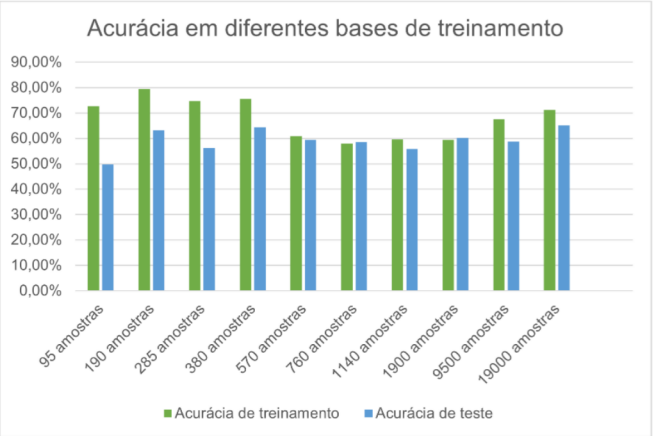


Figura 11. Acurácias de treinamento e de teste com diferentes tamanhos de base de dados .

Para as bases sintéticas de 570 a 1900 amostras foram obtidas acurácias de treinamento e de teste muito próximas, mas os resultados obtidos foram relativamente inferiores quando comparados as bases de amostras originais. Já para as bases com 9500 e 19000 amostras é possível observar novamente um pequeno aumento nas acurárias de teste, mas também um distanciamento quando comparado estes aos resultados de treinamento. Considerando os melhores resultados de acurácia de treinamento, de teste e a capacidade de generalização do modelo foram escolhidas três bases: 380 amostras originais e as bases sintéticas com 1900 e 19000 amostras.

Outro fator importante a ser avalido é o tempo gasto para o treinamento da RNA. Na Figura 12 é possível avaliar a perda em performance com o aumento da quantidade de amostras. Este aumento torna-se especialmente significativo quando é comparado as bases de 380 e 19000 amostras, que obtiveram respectivamente 64,4% e 65% de acurácia de teste, onde o tempo aumenta em quase 20 vezes para a base de maior tamanho. Com base nestas análises foi optado por dar seguimento aos testes com duas bases: 380 e 1900 amostras.

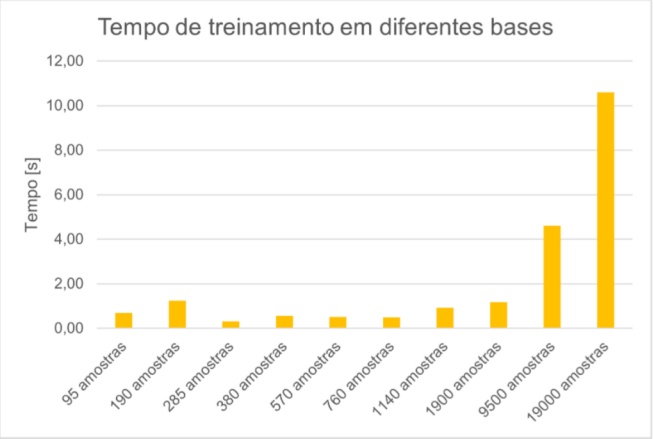


Figura 12. Tempo de treinamento com diferentes tamanhos de base de dados.

Tabela III
MELHORES RESULTADOS OBTIDOS COM AS ALTERAÇÕES DE PARÂMENTROS

Base	Parâmetros alterados	Acurácia de treino	Acurácia de teste	Tempo de treino
380	Função de ativação Sigmoidal e alfa=0.0007	82,37%	70,32%	2,75s
380	Função de ativação Sigmoidal	82,11%	70,95%	2,64s
380	Função de ativação Sigmoidal e alfa=0.0005	82,89%	70,32%	2,95s
380	alfa = 0.003 e solver = LBFGS	78,68%	64,42%	7,53s
380	alfa = 0.005 e 4 camadas ocultas com 100 neurônios	76,84%	63,16%	13,34s
380	alfa = 0.01 , solver = LBFGS e 3 camadas ocultas com 100 neurônios	74,74%	66,53%	11,20s
1900	Função de ativação Sigmoidal e alfa=0.0007	74,05%	66,95%	5,58s
1900	Função de ativação Sigmoidal	73,11%	68,42%	5,70s
1900	Função de ativação Sigmoidal e alfa=0.005	72,89%	67,37%	5,82s
1900	alfa = 0.003 e solver = LBFGS	72,89%	66,74%	28,66s
1900	alfa = 0.005 e 4 camadas ocultas com 100 neurônios	58,26%	57,89%	135,30s
1900	alfa = 0.01 , solver = LBFGS e 3 camadas ocultas com 100 neurônios	59,21%	55,58%	95,58s

B. Ajuste de parâmentros do MLP

A busca pelos melhores paramêtros foi conduzida com o treinamento e o teste da RNA com as bases de treinamento de 380 e 1900 amostras para os diferentes parâmetros e valores apresentados na Tabela III. Os resultados das acurácias de treinamento variaram de 35% a 85% e as acurácias de teste variaram de 40% a 70%, para parâmetros que convergiram, ou seja, onde foi possível determinar um modelo de classificação com base nos treinamentos realizados. As alterações de parâmetros padrões que obtiveram os melhores resultados podem ser observadas na Tabela III.

Um dos fatores muito importantes observados durante os testes com diferentes parâmetros é que a quantidade de camadas ocultas não necessariamente garantem uma melhor acurácia. Na Tabela III é possível observar que para este sistema as acurácias mais altas foram obtidas para as cofiguraçõe padrões da rede: uma única camada oculta.

Outra observação pertinente é que os melhores resultados tanto de treinamento quanto de teste foram obtidos com a base original. Logo é possível concluir que trabalhar com uma grande quantidade de dados sintéticos não garante uma melhoria na eficácia do sistema,ou seja, é preciso escolher um conjunto de dados que realmente represente bem os dados a serem classificados. Além disso, outro impacto observado com o aumento da base de treinamento foi o tempo de treinamento que aumentou em aproximadamente 10 vezes para a base com

1900 amostras.

Na função de ativação apenas duas foram consideradas relevantes para terem seus resultados apresentados: função Sigmoidal da equação 4 e função ReLU da equação 6. As outras funções tiveram resultados inferiores e em muitos casos chegaram a não convergir. Entre a função Sigmoidal e ReLU aquela que obteve o melhor desempenho e o menor tempo de treinamento foi a função sigmoide com os resultados superiores a 70%. Já na função de otimização de pesos a função adam e LBFGS tiveram um destaque maior, porém com resultados bem diferentes em relação ao tempo e por isso foi optado por seguir apenas com a função adam. Como os resultados obtidos para os valores de alfa igual a 0.0001, 0.0005 e 0.0007 eram muito próximos, os três foram utilizados para o teste a ser apresentado a seguir.

Com isso, o último parâmetro a ser determinado seria a quantidade de neurônios na camada oculta. Para isso, foram executados testes variando o seu valor, partindo do mínimo recomendado conforme a Equação 7 até dez vezes este valor. Os valores das acurácias de teste obtidas para as diferentes quantidades de neurônios presentes na camadas oculta podem ser observados na Figura 13.

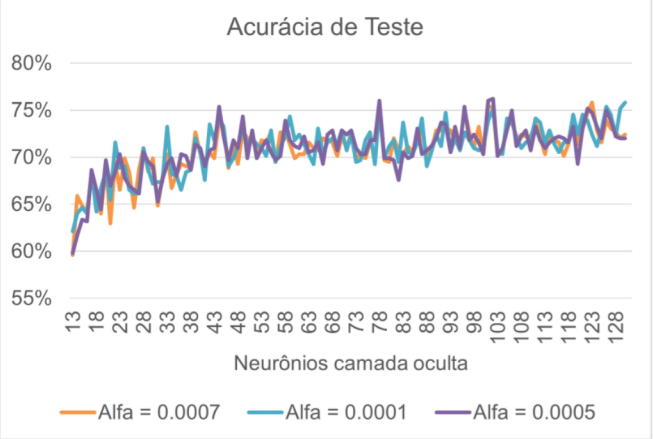


Figura 13. Acurácia de teste variando com a quantidade de neurônios da camada oculta.

Um fator relevante a ser observado na Figura 13 é que a acurácia não necessariamente aumenta com a quantidade de neurônios. Este aumento direto só pode ser observado até o 48º neurônio, após este valor, a RNA estabiliza com acurácias variando de 70% a 75%. Para os três valores de alfa, o máximo de acurácia foi alcançado ao executar o treinamento com 102 neurônios na camda oculta, chegando a 76%. Dentre as opções de alfa, o que obteve o melhor desempenho nos testes com o menor tempo de treinamento foi o de 0.0005, com um tempo de 1,82 segundos.

C. O modelo final de classificação

O modelo de RNA foi obtido conforme as configurações abaixo:

1) **Camada de entrada:**são 7 neurônios, equivalente a quantidade de sensores que alimentam a rede.
2) **Base de treinamento:** 380 amostras originais
3) **Camadas ocultas:** uma única camada oculta com 102 neurônios
4) **Função de ativação:** Sigmoidal
5) **Função de otimização dos pesos:** ADAM, um otimizador baseado em gradiente estocástico
6) **Parâmetro de regularização:** 0.0005

Com estas configurações foram executadas uma série de 50 treinamentos diferentes, agrupando as 380 amostras pelo modelo de validação cruzada de forma aleatória. Com base no melhor modelo de classificação obtido foi verificada a média da acurácia do sistema e sua variação. Estas informações podem ser observados na Figura 14. A acurácia média do sitema é de 75,3%, este valor pode variar 4,4% para mais ou para menos, dependendo da forma como as amostras são divididas nos R subconjuntos.

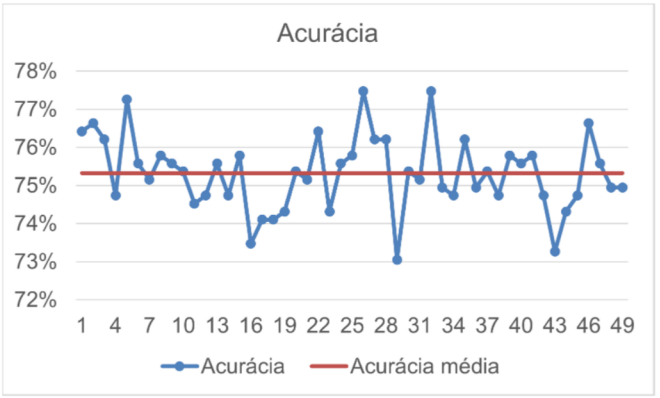


Figura 14. Acurácia média do sitema e sua variação.

Além da acurácia foram avalidas duas métricas de suma importância para avaliar o desempenho do modelo: a precisão e o *recall* com base nos resultados positivos, que tiveram a classificação correta e negativos, que tiveram a classificação incorreta. Precisão é a capacidade do classificador de não rotular como positiva uma amostra negativa e o *recall* é a capacidade do classificador de encontrar todas as amostras positivas.

Na Figura 15 é possivel identificar que o modelo teve dificuldades de identificar as configurações de mão referente às Letras G, L, N e V, devido ao *Recall* destes serem inferiores a 60%. Além disso, para as Letras G, Q e U o modelo teve dificuldades em separar os padrões destas letras em relações às demais abordadas, já que os valores de precisão são inferiores a 60%.

Uma possível causa para essa dificuldade é a similaridade entre as configurações de mão das letras G, L e Q, conforme pode ser observado na Figura 6. A principal diferença entre elas está na posição do polegar, onde foi observada uma baixa confiabilidade na resposta do sensor hall do polegar e a posição do punho, que nem chega a ser abordada pelo sistema de captação proposto. Outra possível causa desta confusão durante as classificações é a similariedade entre as configurações de mão das letras N, U e V, por motivos

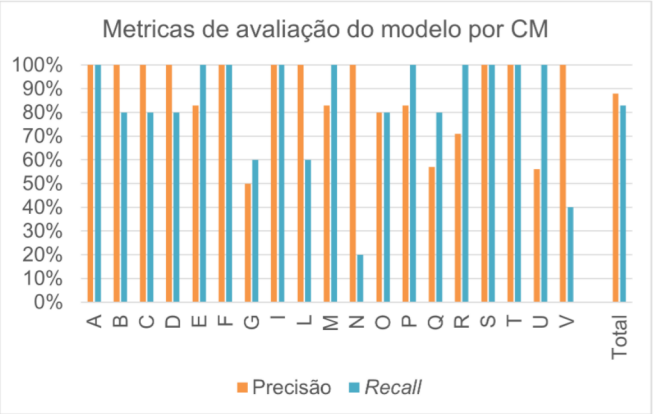


Figura 15. Métricas de avaliação da RNA por configuração de mão.

similares ao das letras anteriores, com exceção que desta vez o sensor seria o do dedo indicador. Essas constatações foram feitas através da observação da matriz confusão gerada com base na classificação das amostras de teste.

V. Considerações Finais

Este trabalho abordou o desenvolvimento de um sistema de reconhecimento de configuração de mãos em Libras a partir de uma luva sensorial especializada com a a aplicação de Redes Neurais Artificiais. Os fatores principais que influenciaram na escolha da luva como fonte de amostras foi o fato de ser mais facilmente aplicada a situações cotidianas e desta não sofrer grandes influências do meio externo. A partir de reconhecimento de configurações de mão é possível desenvolver sistemas mais complexos de interpretação de Libras e auxiliar na comunicação de PcDs.

As configurações de mão abordadas para o reconhecimento são as setenta e nove configuração de mãos que compõem a base de Libras. Porém, foi optado por uma abordagem reduzida com apenas as 19 configurações de mão, equivalente às letras estáticas do alfabeto em Libras. Isto devido a dificuldade de aquisição de amostras além do prótótipo confeccionado ter apresentado algumas falhas e problemas ergonômicos durante as aquisições.

Na construção de um algoritmo classificador baseado em RNA foi optado pela utilização de uma biblioteca da linguagem *Python* [13] devido a facilidade de implementação com poucas configurações. Foi utilizada uma rede *feedforward*, multicamadas e completamente conectada. Para o treinamento da RNA foi escolhida a utilização do algoritmo de *backpropagation* devido a baixa sensibilidade em relação às mudanças no banco de dados, a capacidade de tolerância a falhas e facilidade na implementação.

Para a validação da capacidade de generalização do modelo, optou-se pela utilização do método de Validação Cruzada, onde as amostras foram divididas em base de treinamento, validação e teste com as respectivas porcentagens: 60%, 20% e 20%. O melhor modelo de classificação foi obtido atráves de uma base de 380 amostras. Os resultados obtidos com este modelo tiveram uma acurácia final de 75,3% com um tempo de treinamento de aproximadamente 1,82s.

O modelo final teve dificuldade em classificar algumas configurações de mão corretamente devido a problemas nos sensores de efeito hall utilizados no protótipo para trazer informações extras sobre a posição do polegar e dedo médio. Além disso, foram encontradas algumas variações de posições nas configurações de mão que não foram abordadas no protótipo da luva, como por exemplo o movimento do pulso.

Como trabalhos furutos, seria necessário avaliar outros tipos de sensores para verificar as informações de movimentos laterais do dedo médio e do polegar, assim como o movimento do pulso. Além disso, para garantir uma melhor usabilidade da luva, seria necessário substituir uma forma de comunicação para uma sem fio. Com estas alterações é entendido que o sistema poderia se tornar mais robusto e com um desempenho melhor na classificação das configurações de mão, abrindo caminho para abordar as 79 configurações reconhecidas na Libras.

Referências

[1] INES, "Conheça o INES," accessed 2019-04-06. [Online]. Available: http://www.ines.gov.br/conheca-o-ines

[2] J. Gałka, M. Masior, M. Zaborski, and K. Barczewska, "Inertial motion sensing glove for sign language gesture acquisition and recognition," 2016.

[3] T. L. Baldi, S. Scheggi, L. Meli, M. Mohammadi, and D. Prattichizzo, "Gesto: A glove for enhanced sensing and touching based on inertial and magnetic sensors for hand tracking and cutaneous feedback," 2017.

[4] OMS, "Deafness and hearing loss," accessed 2019-05-01. [Online]. Available: https://www.who.int/en/news-room/fact-sheets/detail/deafness-and-hearing-loss

[5] I. de Moura Frajano Rosa, M. Krieger, R. M. E. de Araujo, and S. L. Porta, "Mapeamento estruturado da libras para utilização em sistemas de comunicação," 2016.

[6] N. Tubaiz, T. Shanableh, and K. Assaleh, "Glove-based continuous arabic sign language recognition in user-dependent mode," 2015.

[7] M. A. Ahmed, B. B. Zaidan, A. A. Zaidan, M. M. Salih, and M. M. bin Lakulu, "A review on systems-based sensory gloves for sign language recognition state of the art between 2007 and 2017," 2018.

[8] P. Plawiak, T. Sosnicki, M. Niedzwiecki, Z. Tabor, and K. Rzecki, "Hand body language gesture recognition based on signals from specialized glove and machine learning algorithms," 2016.

[9] A. Carvalho and K. Faceli, *Inteligência artificial: uma abordagem de aprendizado de máquina*. Grupo Gen - LTC, 2011.

[10] G. Bittencourt., *Inteligência artificial. Ferramentas e Teorias*. Editora UFSC, 2001.

[11] S. Haykin, *Redes Neurais: princípios e práticas*. Editora Bookman, 2001.

[12] Símbolos, "Alfabeto em Libras," accessed 2019-05-25. [Online]. Available: https://www.simbolos.net.br/alfabeto-em-libras/

[13] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.