

Evaluation of recurrent neural network architectures to help motor disabled people through brain computer interface

Fabricao T. Leite¹, Lucas O. Dutra¹, André B. Carneiro¹, Johannes V. Lochter¹

¹Departamento de Engenharia de Computação
Centro Universitário Facens – Sorocaba, SP – Brasil

fabricao.torquatol@gmail.com, lucas_o_dutra@hotmail.com

andre.carneiro@facens.br, johannes.lochter@facens.br

Abstract. *This work evaluates different recurrent neural network architectures to control a virtual object on Robot Operating System (ROS) using electroencephalogram for signal acquisition. For the interface controls, voluntary hand motor actions were used, each hand for a different direction. The LSTM architecture presented better results due to the sequence size and the experimental protocol allowed us to understand that there is an adaptation phase of the individual by improving the result after different attempts without needing retraining of the network.*

Resumo. *Este trabalho avalia diferentes arquiteturas de redes neurais recorrentes para realizar o controle de um objeto virtual construído a partir do Robot Operating System (ROS) utilizando eletroencefalograma para aquisição de sinal. Para as funções de controle da interface foram utilizadas ações motoras voluntárias das mãos, onde cada mão indicava uma direção. A arquitetura LSTM apresentou melhores resultados devido ao tamanho da sequência e o protocolo experimental permitiu compreender que existe uma fase de adaptação do indivíduo ao melhorar o resultado após diferentes tentativas sem precisar do retreinamento da rede.*

1. Introdução

A Pesquisa Nacional de Saúde¹ de 2013 conduzida pelo Instituto Brasileiro de Geografia e Estatística (IBGE) apontou que 6,9% dos brasileiros declararam ter algum tipo de deficiência, a qual considerou deficiências visual, auditiva, motora e mental. No Brasil, a ocorrência de deficiências motoras é considerável, pois 1,3% de toda população têm algum tipo de deficiência física e quase a metade desse total (46,8%) tem grau intenso ou muito intenso de limitações [França et al. 2005].

Esse público têm necessidades especiais devido às suas deficiências e depende de produtos e tecnologias diferenciadas para que suas necessidades sejam atendidas, formando um nicho de mercado e de pesquisa. Existem diversas tecnologias desenvolvidas com o intuito de auxiliar as pessoas com alguma deficiência, como os aplicativos que leem os textos dos computadores, as próteses mecânicas e biônicas, aparelhos auditivos, entre outros.

¹Pesquisa Nacional de Saúde 2013. Disponível em <https://bit.ly/2IVc7Mm>, acessado em 11/07/2019.

As deficiências motoras podem dificultar ou até mesmo impossibilitar o uso de interfaces mais comuns devido às limitações que a deficiência impõe. Nesse contexto, trabalhos na literatura associados ao desenvolvimento de tecnologias para pessoas com deficiências motoras têm utilizado o eletroencefalograma (EEG) como interface cérebro-máquina [Choi and Jo 2013].

O equipamento EEG é um sensor de monitoramento não invasivo capaz de medir a atividade elétrica cerebral espontânea, geralmente utilizado para detectar epilepsia, lesões cerebrais, tumores cerebrais, entre outras doenças. As medições do EEG também tornam possíveis a análise do padrão das ondas e a conversão dos padrões encontrados em sinais de controle de dispositivos [Schalk et al. 2004].

As aplicações da interface cérebro-máquina utilizando EEG são diversas. Os relatos da literatura demonstram algumas tarefas executadas com êxito através desse tipo de interface, como o controle de um jogo feito mentalmente, a substituição de um teclado e um mouse, passando comandos diretamente ao computador com ondas cerebrais e o manuseio de uma cadeira de rodas [Heidrich et al. 2016].

Para que se torne possível a transformação de comandos mentais em sinais de controle de dispositivos, é necessário que exista uma aplicação capaz de fazer essa tradução de forma individual, pois cada pessoa tem um padrão diferente na forma com que as ondas mentais ocorrem [Armstrong et al. 2015]. Nesse sentido, para que a aplicação seja capaz de fazer tal tradução, se faz necessário empregar um componente de aprendizado de máquina, onde cada novo indivíduo terá um modelo para fazer a tradução a partir dos dados fornecidos individualmente.

A área de aprendizado de máquina engloba os algoritmos que habilitam sistemas a aprenderem a partir de dados, os quais geram modelos treinados para realizar tarefas específicas. Nesse trabalho, esses algoritmos têm a tarefa de encontrar o padrão de ondas cerebrais que mapeia cada intenção do usuário na execução de uma ação ou sinal de controle de dispositivo diferente. Trabalhos na literatura usam frequentemente redes neurais recorrentes para lidar com essa tarefa [Alhagry et al. 2017].

Este artigo é derivado de um Trabalho de Conclusão de Curso em andamento a ser defendido no curso de Engenharia da Computação. A proposta deste artigo é propor um sistema de controle utilizando um dispositivo EEG para controlar a direção que um agente pode se movimentar em um ambiente virtual. Desse modo, se for possível traduzir com aprendizado de máquina os comandos mentais em ações para o sistema de controle, o mesmo modelo poderia ser replicado para sistemas de controle de navegação em situações reais.

O restante deste artigo está organizado da seguinte forma: na Seção 2 são abordados os trabalhos relacionados para contextualizar a proposta deste artigo na literatura; na Seção 3 aborda-se a metodologia da aquisição de dados, tratamento e treinamento das redes neurais; a Seção 4 traz os resultados e a discussão deles; finalmente, na Seção 5, este artigo é encerrado com as conclusões e direções de trabalhos futuros.

2. Trabalhos relacionados

Os estudos e pesquisas na área da neurociência têm convergido a avanços promissores. Um exemplo de trabalho nesse campo é a percepção de que cada área do cérebro humano é

responsável por executar diferentes funções do corpo, tais como: motricidade, afetividade e cognição [de Souza et al. 2015].

Ao perder completa ou parcialmente uma dessas funções, como motricidade, seria possível utilizar a cognição para supri-la, conforme foi observado em [Schalk et al. 2004]. Essa transferência de funções auxiliada por máquinas foi possível devido ao desenvolvimento da área de interface cérebro-máquina. Essa interface possibilita monitorar a atividade cerebral e traduzi-la para sinais de controle de dispositivos.

As pesquisas em interface cérebro-máquina estão diretamente relacionadas com acessibilidade. Em [Schuh et al. 2013] os autores avaliam a utilização de EEG para controlar uma cadeira de rodas em um ambiente virtual utilizando o piscar dos olhos. Analogamente, em [Roque et al. 2017] foi avaliado o uso de EEG para controlar um protótipo real de uma cadeira de rodas.

Outro exemplo de trabalho na área de interface cérebro-máquina e acessibilidade é a construção de um sistema tradutor dos sinais límbicos em respostas digitais. Esse tipo de aplicação abre espaço para novas aplicações e aumenta a autonomia de pessoas com limitações motoras [Gonçalves et al. 2017]. Nesse trabalho especificamente, são discutidas técnicas de pré-processamento de sinais necessárias para tratar a leitura das ondas a partir do EEG.

Trabalhos voltados para a inclusão social também se beneficiam dos avanços em interface cérebro-máquina, como [Heidrich et al. 2016] para a inclusão escolar de pessoas com paralisia cerebral. Nesse trabalho foi desenvolvido um sistema que permite a uma pessoa transferir comandos a um computador diretamente sem depender de um dispositivo de entrada como teclado ou *mouse*. As atividades cerebrais do usuário são monitoradas e traduzidas em comandos, os quais o computador responde.

Especificamente ao trabalhar com a tradução de sinais de controle é muito comum encontrar trabalhos que utilizam ambientes virtuais, pois reduz os custos com prototipação e permite realizar testes de maneira mais rápida. Esse trabalho conduz os testes de maneira virtual, como foi feito em [Schuh et al. 2013].

2.1. Redes neurais recorrentes

Alinhado com a literatura também está a proposta de utilizar redes neurais como componente de aprendizado de máquina para traduzir as ondas cerebrais em sinais de controle de máquina. Devido os dados de entrada serem impulsos cerebrais capturados pelos sensores EEG, e estes terem natureza intrinsecamente temporal, uma escolha natural são as redes neurais recorrentes, as quais são apropriadas para capturar padrões em dados com características temporais [Wu et al. 2017].

A escolha das redes neurais recorrentes é apoiada na literatura. Trabalhos semelhantes ao proposto aqui, no qual há a utilização de sensores EEG e um componente de aprendizado de máquina, incluem [Michielli et al. 2019], no qual foi avaliado o uso de redes neurais recorrentes para classificar o estágio de sono a partir dos dados coletados com EEG; e [Alhagry et al. 2017], onde foi avaliada a mesma arquitetura de redes neurais para a tarefa de reconhecimento de emoções.

Um exemplo de rede neural recorrente é dado na Figura 1, onde é possível notar que diferente de uma rede *perceptron* mais tradicional, a camada de saída (“*output layer*”)

retroalimenta a camada intermediária (“*hidden layer*”).

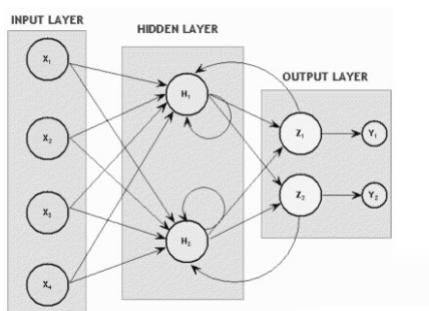
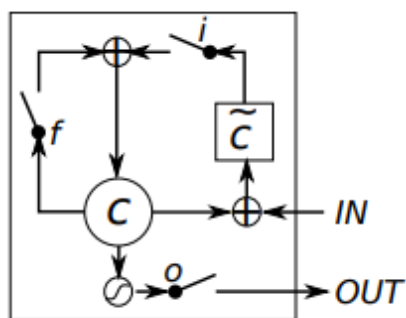


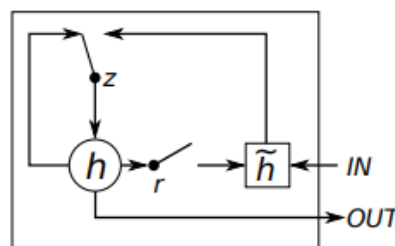
Figura 1. Arquitetura de uma RNN [Bishop 2006]

As principais arquiteturas recorrentes encontradas na literatura são Long Short-Term Memory (LSTM) [Hochreiter and Schmidhuber 1997] na qual sua principal célula está representada na Figura 2a e a Gated Recurrent Units (GRU) [Dey and Salem 2017] que pode ser observada na Figura 2b. Essas arquiteturas surgiram como alternativas a um problema chamado “*catastrophic forgetting*” que afetava diretamente as redes neurais recorrentes mais simples, na qual longas sequências não eram capazes de serem aprendidas [McCloskey and Cohen 1989].

A retroalimentação simples não é capaz de representar padrões sequenciais longos, portanto foram propostos mecanismos que compõem as arquiteturas LSTM e GRU, capazes de aprender quais características devem ser esquecidas e quais devem ser retidas a partir de diferentes elementos de uma sequência [Chung et al. 2014].



(a) Long Short-Term Memory (LSTM)



(b) Gated Recurrent Unit (GRU)

Figura 2. Organização celular de redes neurais recorrentes [Chung et al. 2014]

A proposta de utilização de EEG, o pré-processamento de sinais e as redes neurais recorrentes deste artigo para traduzir comandos mentais em sinais de controle de dispositivo estão alinhados com os trabalhos mais recentes da literatura. Toda a metodologia experimental referente aos elementos anteriores são discutidos a seguir.

3. Materiais e Métodos

Com o intuito de validar a proposta deste trabalho e tornar os experimentos reproduzíveis, nas subseções a seguir são descritas as etapas de coleta de dados com os sensores de EEG,

o pré-processamento dos sinais e todas configurações utilizadas nas arquiteturas de redes neurais recorrentes para realizar a classificação dos comandos. Os comandos, por sua vez, são utilizados para manipular um agente em ambiente virtual na plataforma Robot Operating System (ROS).

3.1. Coleta de dados

Para a realização da captura dos impulsos cerebrais foi utilizado um eletroencefalograma EMOTIV EPOC+ 14 Channel, da empresa EMOTIV (Figura 3). Esse equipamento possui frequência de amostragem de 128 Hz com conversor analógico-digital de 16 bits com 14 canais de eletrodos: AF3, F7, F3, FC5, T7, P7, O1, O2, P8, T8, FC6, F4, F8 e AF4 [Emotiv 2019]. O mesmo equipamento já foi utilizado em outros trabalhos na literatura, tanto pela alta sensibilidade quanto o custo atrativo [Roque et al. 2017].



Figura 3. EMOTIV EPOC+ 14 Channel [Emotiv 2019]

Para transmissão e captura dos dados obtidos pelo equipamento para o computador utilizou-se a biblioteca de código fonte aberto EMOKIT [Brocius and Machulis 2019].

Para o objetivo de controlar o objeto virtual, foi criado um protocolo de coleta de dados com duração de vinte e cinco minutos e diferentes ciclos que abordam situações diferentes, enquanto o usuário tenta controlar o objeto. As situações diferentes foram propostas como parte do protocolo para que o componente de aprendizado de máquina possa generalizar o comportamento das ondas cerebrais com relação aos comandos de controle do objeto independente da situação. As situações consideradas são:

- olhos abertos com o participante sentado;
- olhos abertos com o participante em pé;
- olhos fechados, sentado e ouvindo músicas do seu gosto musical;
- olhos abertos, sentado e ouvindo músicas do seu gosto musical;
- olhos abertos, em pé e respondendo a um questionário não-invasivo.

Durante cada um dos ciclos de aquisição, o participante foi exposto às imagens retratando ações motoras voluntárias, sendo elas: uma seta para a direita que representaria ação motora na direção direita, uma seta para a esquerda que representaria ação motora da direção esquerda e um círculo que representaria nenhuma ação motora.

A única exceção é a situação em que o participante está de olhos fechados, onde um sinal sonoro é dado para o participante abrir o olho somente em momentos específicos para visualizar as imagens.

3.2. Pré-Processamento de Dados

Os dados coletados nas diferentes situações mencionadas anteriormente precisaram ser pré-processados para uso posterior no componente de aprendizado de máquina. Para

compreender melhor o pré-processamento, tanto a fonte dos dados, quanto o formato e as características são discutidos a seguir.

O equipamento EEG na frequência de 128 Hz e 14 canais fornece uma matriz 128x14 com 16 bits a cada leitura. Como as ondas cerebrais de interesse estão na faixa de 0 a 30 Hz (Tabela 1), essas informações coletadas são passadas pelo algoritmo Fast Fourier Transform (FFT) [Brigham and Brigham 1988] na faixa de frequência de 0 a 30 Hz, resultando em uma matriz de 30x14 no domínio da frequência.

Tabela 1. Classificação das Bandas de Frequências [Bonini-Rocha et al. 2008]

Delta (0-4 Hz)	Vigília com estado de sono profundo, meditação, ações involuntárias e estado inconsciente.
Teta (4-8 Hz)	Vigília com estado de atenção forçada (concentração), resolução de problemas lógicos e processamento de memórias.
Alfa (8-12 Hz)	Vigília com relaxamento.
Beta (12-30 Hz)	Vigília com estado de atenção.

Após a transformação dos dados coletados para o domínio da frequência, foi realizada a média ponderada e aritmética de cada onda, de modo que a matriz resultante tem as dimensões 14x4x2. Desse modo, cada instante de dados coletados são representados pela média ponderada e aritmética para cada um dos 14 canais do dispositivo e as 4 classificações de onda.

3.3. Arquiteturas avaliadas

O principal objetivo deste trabalho é avaliar arquiteturas diferentes de redes neurais recorrentes e analisar o desempenho delas na tarefa de traduzir comandos mentais em sinais de controle de dispositivos. Os dados obtidos a partir dos sensores EEG e pré-processados pela transformada de Fourier são transmitidos a uma rede neural recorrente para aprender o padrão entre os comandos mentais e como a tradução deverá ser feita para sinais de controle.

No trabalho [Alhagry et al. 2017] foi avaliada a utilização da variação LSTM de redes neurais recorrentes utilizando 2 camadas empilhadas com *dropout* de 20% entre elas. A função do *dropout* é desativar alguns neurônios para forçar com que os neurônios vizinhos ajustem o modelo, reduzindo o efeito de *overfitting* ao custo de dificultar a convergência do modelo. Nesse trabalho, a primeira camada da LSTM tem ativação ReLU e a segunda camada tem ativação *softmax*.

Utilizando o trabalho [Alhagry et al. 2017] como base, são propostas 4 variações diversificadas. A implementação foi feita com a biblioteca PyTorch [Paszke et al. 2017], na qual os números de parâmetros a serem treinados em cada arquitetura pode ser observado na Tabela 2. A partir da escolha dos autores de utilizar LSTM, esse trabalho inclui a arquitetura GRU também. Além de utilizar 2 camadas empilhadas (LSTM-2 e GRU-2), esse trabalho avalia a utilização de 4 camadas também (LSTM-4 e GRU-4) para aumentar a complexidade de análise das sequências de informações passadas pela rede. Também

são consideradas as variações de utilização com *dropout* e a mudança das ativações entre as camadas.

Tabela 2. Tabela de comparação de número de parâmetros treináveis

Arquitetura	Parâmetros
LSTM-2	4083
LSTM-4	8435
GRU-2	3075
GRU-4	6339

A camada de entrada das arquiteturas recorrentes descritas neste trabalho são sequências de 32 elementos com 128 valores que correspondem a um tempo de coleta de 0,15 segundos de dados pré-processados provenientes do EEG. A última camada da arquitetura recorrente produz um vetor de 16 valores e se conecta a uma camada densa que produz um vetor de 11 valores. A camada densa é conectada à saída de 3 neurônios da rede, um para cada classe. Os valores de 16 neurônios para as células recorrentes e 11 neurônios para camada densa foram escolhidos arbitrariamente.

Finalmente, a função de otimização escolhida foi *Stochastic Adam* por possuir vantagens para o aprendizado de problemas com grandes número de parâmetros e apropriado para problemas com gradientes muito ruidosos ou esparsos [Kingma and Ba 2014].

3.4. Conjunto de dados

Para a criação do conjunto de dados, o protocolo de coleta foi seguido com um indivíduo utilizando o sensor EEG e os dados foram armazenados utilizando o protocolo de pré-processamento. O conjunto de dados utilizado contém 2400 amostras e cada amostra está associada a uma de 3 classes: sem atividade motora, atividade motora esquerda e motora direita.

As redes foram avaliadas utilizando *hold-out* com uma parcela de dados para validação. As três divisões de dados (treino, teste e validação) foram feitas de forma estratificada, de modo a manter a proporção da ocorrência de cada classe de maneira igual entre as divisões.

Para treinamento foram separadas 1536 amostras (64%) e a cada época o desempenho da rede era medido utilizando 480 amostras (20%) diferentes das utilizadas no treino. Ao terminar 25 épocas de treinamento, uma parcela diferente de dados (validação) era utilizada para avaliar o resultado final do treinamento, contendo 384 amostras (16%).

4. Resultados e Discussões

Os resultados para as quatro variações de arquiteturas mencionadas na metodologia são apresentados na Tabela 3. O tempo reportado é referente ao tempo de treinamento com 25 épocas, enquanto a coluna *loss* representa o custo final de otimização após todas as épocas. A coluna de acurácia se refere ao resultado obtido com as amostras da etapa de validação.

Os resultados apontam que as melhores redes foram GRU-2 e LSTM-2, indicando que utilizar quatro camadas empilhadas pode trazer complexidade desnecessária ao modelo. De fato, a quantidade de amostras pode ser pequena para a complexidade alcançada

Tabela 3. Tabela de comparação de resultados

Arquitetura	Tempo	Loss	Acurácia
[Alhagry et al. 2017]	30s	0,1077	60,20%
LSTM-2	25s	0,0967	69,90%
LSTM-4	34s	0,1007	66,45%
GRU-2	27s	0,0919	70,41%
GRU-4	35s	0,0959	67,91%

com quatro camadas. Também é possível perceber que a quantidade de dados não deixa evidente a diferença de tempo de execução entre LSTM e GRU, onde a segunda é notoriamente conhecida na literatura por ser mais rápida e ter menos parâmetros, o que não é possível de se concluir nos resultados obtidos.

Observando os resultados das arquiteturas propostas neste trabalho em comparação com a arquitetura proposta em [Alhagry et al. 2017], é possível perceber que as arquiteturas mencionadas neste trabalho conduzem a resultados melhores para o problema da identificação de ações motoras.

4.1. Avaliação no Robot Operating System

Este trabalho também propõe um protocolo experimental para classificar o grau de abstração do sistema de controle construído no Robot Operating System (ROS). ROS é um sistema de código fonte aberto para o desenvolvimento de protótipos de robótica em ambientes virtuais [Quigley et al. 2009].

Uma das ferramentas que o ROS oferece é *turtlesim*, uma ferramenta didática para simular um agente [ROS 2019]. Para uma melhor representatividade do grau de abstração do sistema de controle, ficou estabelecido como teste o controle de um modelo do *turtlesim* através da interface cérebro-máquina.

Para avaliar a adaptação do usuário à plataforma, um objeto estático é instanciado de forma aleatória no ambiente do *turtlesim* e o objetivo do usuário é alcançar o objeto no menor tempo possível utilizando as ondas cerebrais.

Como os resultados indicaram que LSTM-2 e GRU-2 são as melhores arquiteturas, optou-se por utilizar a LSTM-2 nesse experimento com ROS já que a sequência de informações de entrada é longa. Os resultados conduzidos com um indivíduo em três etapas diferentes são apresentados na Tabela 4 medidos em segundos e nas Figuras 4a, 4b e 4c são mostrados os caminhos feitos.

Tabela 4. Resultados do tempo dos participantes para concluir o teste

	Rodada 1	Rodada 2	Rodada 3
Voluntario 1	19s	13s	10s

Através do caminho percorrido pelo indivíduo e dos tempos de execução anotados para que fosse atingido o objetivo em cada rodada é possível perceber que existe uma curva de aprendizado e de adaptação ao sistema. É importante notar que não houve retreinamento da rede entre as rodadas, mostrando que os ganhos subsequentes são provenientes da adaptação do indivíduo ao uso dos sensores e à rede treinada.

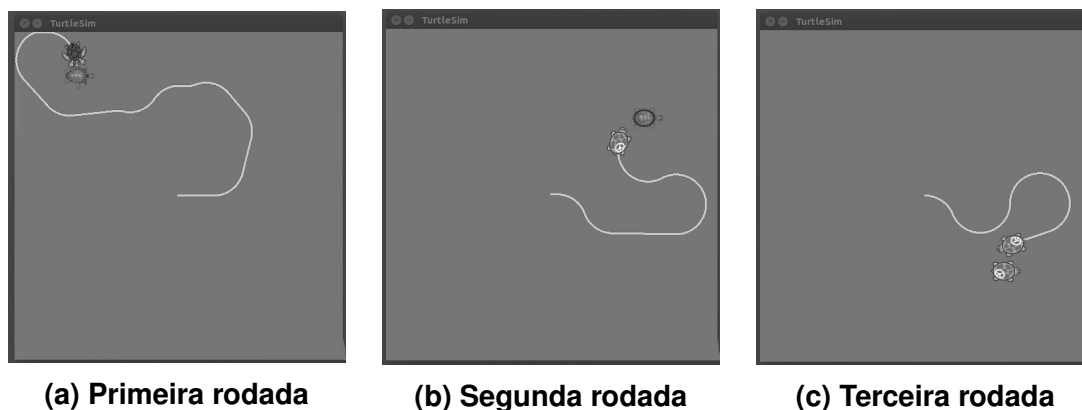


Figura 4. Caminho percorrido pelo indivíduo durante o teste

5. Conclusão

A deficiência motora é uma das deficiências físicas mais comum entre os brasileiros e as necessidades especiais desse público têm aberto oportunidades de mercado e de pesquisa. Uma das opções que pode auxiliar esse público são os sensores EEG e o avanço da tecnologia para traduzir os comandos mentais capturado através do EEG em sinais de controle de dispositivos.

Esse artigo é oriundo de um Trabalho de Conclusão de Curso em Bacharelado de Engenharia de Computação e explorou a aplicação de redes neurais recorrentes para traduzir os comandos mentais do EEG em sinais de controle, avaliando diferentes arquiteturas e comparando com trabalhos semelhantes na literatura.

Os resultados obtidos mostraram que não existe diferenças muito significativas entre as arquiteturas avaliadas LSTM e GRU, com 2 e 4 camadas empilhadas, porém apresentaram resultados melhores que o trabalho da literatura mais próximo ao presente artigo. As principais diferenças foram a utilização de *dropout* e a ativação ReLU no trabalho comparado.

A arquitetura eleita a melhor (LSTM com 2 camadas empilhadas), seguindo o critério de desempenho e considerando que este problema trata de sequências longas, foi utilizada em um experimento com *turtlesim*, mostrando que o sistema também depende de adaptação do usuário à utilização do EEG para controle.

Como trabalhos futuros, mais dados deverão ser coletados e análises mais robustas deverão ser empregadas para analisar o impacto do protocolo de coleta de dados. Esse protocolo tem papel fundamental no desempenho dos sistemas que dependem de interface cérebro-máquina.

Referências

- Alhagry, S., Fahmy, A. A., and El-Khoribi, R. A. (2017). Emotion recognition based on eeg using lstm recurrent neural network. *International Journal of Advanced Computer Science and Applications*, 8(10).
- Armstrong, B. C., Ruiz-Blondet, M. V., Khalifian, N., Kurtz, K. J., Jin, Z., and Laszlo, S. (2015). Brainprint: Assessing the uniqueness, collectability, and permanence of a novel method for erp biometrics. *Neurocomputing*, 166:59 – 67.

- Bishop, C. M. (2006). *Pattern recognition and machine learning*. springer.
- Bonini-Rocha, A. C., Timm, M. I., Chiaramonte, M., Zaro, M., Rasia-Filho, A. A., Wolff, D., Ayres, E. d. P., and Petersen, R. D. d. S. (2008). Metodologia para observação e quantificação de sinais de eeg relativos a evidências cognitivas de aprendizagem motora. *Ciências & Cognição*, 13(2):27–50.
- Brigham, E. O. and Brigham, E. O. (1988). *The fast Fourier transform and its applications*, volume 448. prentice Hall Englewood Cliffs, NJ.
- Brocious, C. and Machulis, K. (2019). Emokit. *OpenYou*, [Online]. Available: <http://www.openyou.org>.
- Choi, B. and Jo, S. (2013). A low-cost eeg system-based hybrid brain-computer interface for humanoid robot navigation and recognition. *PloS one*, 8(9):e74583.
- Chung, J., Gulcehre, C., Cho, K., and Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.
- de Souza, J. P. G., Krizan, J., de Melo Costa, G., and de Oliveira Fermoseli, A. F. (2015). Interface cérebro-máquina (icm): da transdução do estímulo externo em impulso nervoso à tradução em comandos digitais. *Caderno de Graduação - Ciências Biológicas e da Saúde - UNIT-ALAGOAS*, 3(1):139–152.
- Dey, R. and Salem, F. M. (2017). Gate-variants of gated recurrent unit (GRU) neural networks. *CoRR*, abs/1701.05923.
- Emotiv (2019). Epoc+ 14 channel - emotiv. <https://www.emotiv.com/epoc-14-channel-eeg-2/>. (Accessed on 05/12/2019).
- França, C. R., Borges, J. A., and Sampaio, F. F. (2005). Tupi—recursos de acessibilidade para educação especial e inclusiva dos deficientes motores. In *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informação na Educação – SBIE)*, volume 1, pages 591–600.
- Gonçalves, W. d. O. et al. (2017). Utilização dos sinais de eletroencefalograma e eletrodermal no aprendizado por reforço de uma interface cérebro-máquina.
- Heidrich, R. d. O., Branco, M. A., Mossmann, J. B., Schuh, A. R., and Jensen, E. (2016). Jogos digitais para interação com brain computer interface para auxiliar no processo de inclusão escolar de pessoas com paralisia cerebral. *Blucher Design Proceedings*, 2(9):3397–3407.
- Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9:1735–80.
- Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- McCloskey, M. and Cohen, N. J. (1989). Catastrophic interference in connectionist networks: The sequential learning problem. volume 24 of *Psychology of Learning and Motivation*, pages 109 – 165. Academic Press.
- Michielli, N., Acharya, U. R., and Molinari, F. (2019). Cascaded lstm recurrent neural network for automated sleep stage classification using single-channel eeg signals. *Computers in Biology and Medicine*, 106:71 – 81.

- Paszke, A., Gross, S., Chintala, S., and Chanan, G. (2017). Pytorch. *Computer software. Vers. 0.3*, 1.
- Quigley, M., Conley, K., Gerkey, B. P., Faust, J., Foote, T., Leibs, J., Wheeler, R., and Ng, A. Y. (2009). Ros: an open-source robot operating system. In *ICRA Workshop on Open Source Software*.
- Roque, K., Cezar, P., et al. (2017). Controle de um protótipo de cadeira de rodas através de uma interface cérebro-computador.
- ROS (2019). Turtlesim. <http://wiki.ros.org/turtlesim>. (Accessed on 05/12/2019).
- Schalk, G., McFarland, D. J., Hinterberger, T., Birbaumer, N., and Wolpaw, J. R. (2004). Bci2000: a general-purpose brain-computer interface (bci) system. *IEEE Transactions on biomedical engineering*, 51(6):1034–1043.
- Schuh, A. R., Lima, A., Heidrich, R. d. O., Mossmann, J., Flores, C., and Bez, M. R. (2013). Desenvolvimento de um simulador controlado por interface cérebro-computador não-invasiva para treinamento na utilização de cadeira de rodas. *RENOTE*, 11(3).
- Wu, C.-Y., Ahmed, A., Beutel, A., Smola, A. J., and Jing, H. (2017). Recurrent recommender networks. In *Proceedings of the tenth ACM international conference on web search and data mining*, pages 495–503. ACM.