



Otimizando Parâmetros em Redes Neurais para Aprimorar um Ambiente Inteligente

Lucas Rodrigues Frank

JUIZ DE FORA
NOVEMBRO, 2019

Otimizando Parâmetros em Redes Neurais para Aprimorar um Ambiente Inteligente

LUCAS RODRIGUES FRANK

Universidade Federal de Juiz de Fora
Instituto de Ciências Exatas
Departamento de Ciência da Computação
Bacharelado em Ciência da Computação

Orientador: Edelberto Franco Silva

JUIZ DE FORA
NOVEMBRO, 2019

OTIMIZANDO PARÂMETROS EM REDES NEURAIS PARA APRIMORAR UM AMBIENTE INTELIGENTE

Lucas Rodrigues Frank

MONOGRAFIA SUBMETIDA AO CORPO DOCENTE DO INSTITUTO DE CIÊNCIAS EXATAS DA UNIVERSIDADE FEDERAL DE JUIZ DE FORA, COMO PARTE INTEGRANTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE BACHAREL EM CIÊNCIA DA COMPUTAÇÃO.

Aprovada por:

Edelberto Franco Silva
Doutor em Computação

Luciana Brugiolo Gonçalves
Doutora em Computação

Francisco Henrique Cerdeira Ferreira
Mestre em Ciência da Computação

JUIZ DE FORA
28 DE NOVEMBRO, 2019

Aos meus pais e ao meu irmão pelo apoio.

Resumo

Nos dias atuais, a busca por modelos preditivos mais precisos são sempre um foco importante independentemente da sua área de aplicação. Sendo um destes modelos, as Redes Neurais (RN), que são utilizadas já há algumas décadas, possuem um grande poder de computação por apresentar variados parâmetros que influenciam diretamente em seu funcionamento. Este trabalho tem como objetivo otimizar o número de neurônios e a taxa de aprendizagem da RN usando uma meta-heurística, assim também, analisar a Função de Ativação (FA) utilizada. Para tal, propõe-se uma análise de três FAs já encontradas na literatura para que possa ser visto o seu impacto nos resultados de duas RNs.

Neste contexto, outro ponto que se destaca são os ambientes em que a avaliação será conduzida, referente ao tráfego de veículos e carga de conexões em redes sem fio. A predição de tráfego de veículos e do volume de uso da rede sem fio são problemas relevantes na atualidade. Uma vez que o gerenciamento do tráfego pode auxiliar na vida da população no trânsito, o monitoramento de carga da rede pode oferecer uma conexão de melhor qualidade para o cliente. Sendo assim, para a validação deste estudo, o propósito é de realizar testes das FAs selecionadas, submetendo o cenário de tráfego de veículos e de rede sem fio a dois modelos de RNs, com o intuito de prever o número de veículos e carga de usuários.

Palavras-chave: Otimização, Rede Neural, Função de Ativação, Predição, Ambiente Inteligente.

Abstract

Nowadays, looking for more accurate predictive models is an important focus regardless of the field. Being one of these models, Neural Networks (NN), which have been used for a few decades, have a great computing power because it presents various parameters that directly influence its functioning. This work aims to optimize the number of neurons and the learning rate parameter of NN using a metaheuristic, as well as to analyze the Activation Function (AF) used. To do it is proposes an analysis of three AFs already found in the literature and their impact on the results on two different NNs.

In this context, another point that stands out is the environment in which it will be used, in our study will be the vehicle traffic and wireless network load. The prediction of vehicle traffic and the wireless network load are currently on the rise. Traffic management can help the population's life in movement, while network load monitoring can provide a better quality connection to the customer. Therefore, in order to validate this study, we aim to perform tests of the selected FAs, submitting the vehicle traffic and wireless network scenario to two models of NNs in order to predict the number of vehicles and user load.

Keywords: Optimization, Neural Network, Activation Function, Prediction, Smart Environment.

Agradecimentos

A todos os meus parentes, pelo encorajamento e apoio. Em especial, a minha mãezinha que se foi, mas se faz presente todos os dias em meu coração.

Ao professor Edelberto pela orientação, amizade e principalmente, pela paciência, sem a qual este trabalho não se realizaria.

Aos amigos que fiz e também, aos que já fazem parte da minha vida e que de alguma forma me ajudou ao longo dessa jornada.

Aos professores do Departamento de Ciência da Computação pelos seus ensinamentos e aos funcionários do curso, que durante esses anos, contribuíram de algum modo para o nosso enriquecimento pessoal e profissional.

"Hope is what makes us strong. It is why we are here. It is what we fight with when all else is lost."

Pandora (God Of War)

Conteúdo

Lista de Figuras	8
Lista de Tabelas	11
Lista de Abreviações	12
1 Introdução	13
1.1 Justificativa/Motivação	15
1.2 Objetivos	16
2 Fundamentação Teórica	17
2.1 Redes Neurais Artificiais	17
2.1.1 Modelo <i>Feedforward</i>	19
2.1.2 Modelo Recorrente	19
2.2 Função de Ativação	21
2.2.1 Logística	22
2.2.2 ReLU	23
2.2.3 <i>Softplus</i>	24
2.3 Otimização por Enxame de Partículas	26
3 Trabalhos Relacionados	28
3.1 <i>Comparison of Activation Functions in Multilayer Neural Network for Pattern Classification</i>	28
3.2 <i>A Comparison of Two Sigmoidal-Type Activation Functions</i>	29
3.3 <i>A Neural Network Based User Identification for Tor Networks</i>	30
3.4 <i>On Hyperbolic Sine Activation Functions Used in ZNN</i>	31
3.5 <i>Implementation of Activation Functions for ELM based Classifiers</i>	31
3.6 <i>Parameters optimization of deep learning models using Particle swarm optimization</i>	32
3.7 <i>Traffic flow forecasting with particle swarm optimization and support vector regression</i>	33
3.8 <i>Using LSTM and GRU neural network methods for traffic flow prediction</i>	33
3.9 <i>Applying a Multilayer Perceptron for Traffic Flow Prediction</i>	34
3.10 <i>Multilayer Perceptron and Particle Swarm Optimization Applied</i>	35
3.11 Considerações Sobre o Capítulo	35
4 Abordagem Proposta	37
4.1 Rede Neural	37
4.1.1 Entrada de Dados	37
4.1.2 MLP	38
4.1.3 GRU	39
4.2 PSO	40
4.2.1 Função Objetivo	41

5	Avaliação e Resultados	44
5.1	Base de Dados	45
5.1.1	PeMS	45
5.1.2	UFJF	45
5.2	Resultados	46
5.2.1	PeMS	48
5.2.2	UFJF	51
6	Conclusão e Trabalhos Futuros	59
	Referências Bibliográficas	61
A	Apêndice	65

Lista de Figuras

2.1	Composição da estrutura de um neurônio. Fonte (Furtado, 2019).	18
2.2	(a) Arquitetura de uma rede neural <i>Feedforward</i> com múltiplas camadas e duas saídas; (b) Arquitetura de uma rede neural <i>Feedforward</i> sem camada oculta e com 4 saídas. Fonte (Sazli, 2006).	20
2.3	Exemplificando arquitetura de uma célula em rede neural GRU. Fonte (R. Fu et al, 2016)	21
2.4	Gráfico da função de ativação logística com seu parâmetro α sendo variado.	22
2.5	Gráfico da função de ativação logística com seu parâmetro α sendo variado.	23
2.6	Gráfico da função de ativação ReLU.	24
2.7	Gráfico da função de ativação ReLU.	25
2.8	Gráfico da função de ativação softplus.	25
2.9	Gráfico da derivada da função de ativação softplus.	26
4.1	Vetor de característica para entrada na rede neural.	38
4.2	Arquitetura do modelo MLP utilizado.	39
4.3	Arquitetura do modelo GRU utilizado.	39
5.1	Conjunto de dados do PeMS.	46
5.2	Conjunto de dados da UFJF.	47
5.3	Mapa da UFJF com os locais selecionados para o cenário de rede sem fio. .	48
5.4	Comparando valor predito com valor real para modelo MLP com função logística. Linha azul corresponde ao valor real e linha laranja ao valor predito. Cenário: Tráfego de Veículos.	49
5.5	Comparando valor predito com valor real para modelo GRU com função logística. Linha azul corresponde ao valor real e linha laranja ao valor predito. Cenário: Tráfego de Veículos.	50
5.6	Comparando o desempenho dos modelos em relação ao número de épocas médio até sua convergência. Cenário: Tráfego de Veículos.	50
5.7	Comparando valor predito com valor real para modelo MLP com função logística. Linha azul corresponde ao valor real e linha laranja ao valor predito. Cenário: Rede sem fio. Local: Engenharia.	53
5.8	Comparando valor predito com valor real para modelo GRU com função logística. Linha azul corresponde ao valor real e linha laranja ao valor predito. Cenário: Rede sem fio. Local: Engenharia.	53
5.9	Comparando valor predito com valor real para modelo MLP com função logística. Linha azul corresponde ao valor real e linha laranja ao valor predito. Cenário: Rede sem fio. Local: ICE	55
5.10	Comparando valor predito com valor real para modelo GRU com função logística. Linha azul corresponde ao valor real e linha laranja ao valor predito. Cenário: Rede sem fio. Local: ICE	55
5.11	Comparando valor predito com valor real para modelo MLP com função logística. Linha azul corresponde ao valor real e linha laranja ao valor predito. Cenário: Rede sem fio. Local: IAD	56

5.12	Comparando valor predito com valor real para modelo GRU com função ReLU. Linha azul corresponde ao valor real e linha laranja ao valor predito. Cenário: Rede sem fio. Local: IAD	57
A.1	Comparando valor predito com valor real para modelo MLP com função ReLU. Linha azul corresponde ao valor real e linha laranja ao valor predito. Cenário: Tráfego de Veículos.	65
A.2	Comparando valor predito com valor real para modelo MLP com função <i>Softplus</i> . Linha azul corresponde ao valor real e linha laranja ao valor predito. Cenário: Tráfego de Veículos.	65
A.3	Comparando valor predito com valor real para modelo GRU com função ReLU. Linha azul corresponde ao valor real e linha laranja ao valor predito. Cenário: Tráfego de Veículos.	66
A.4	Comparando valor predito com valor real para modelo GRU com função <i>Softplus</i> . Linha azul corresponde ao valor real e linha laranja ao valor predito. Cenário: Tráfego de Veículos.	66
A.5	Comparando valor predito com valor real para modelo MLP com função ReLU. Linha azul corresponde ao valor real e linha laranja ao valor predito. Cenário: Rede sem fio. Local: Engenharia	67
A.6	Comparando valor predito com valor real para modelo GRU com função ReLU. Linha azul corresponde ao valor real e linha laranja ao valor predito. Cenário: Rede sem fio. Local: Engenharia	67
A.7	Comparando valor predito com valor real para modelo MLP com função <i>Softplus</i> . Linha azul corresponde ao valor real e linha laranja ao valor predito. Cenário: Rede sem fio. Local: Engenharia	68
A.8	Comparando valor predito com valor real para modelo GRU com função <i>Softplus</i> . Linha azul corresponde ao valor real e linha laranja ao valor predito. Cenário: Rede sem fio. Local: Engenharia	68
A.9	Comparando valor predito com valor real para modelo MLP com função ReLU. Linha azul corresponde ao valor real e linha laranja ao valor predito. Cenário: Rede sem fio. Local: ICE	69
A.10	Comparando valor predito com valor real para modelo GRU com função ReLU. Linha azul corresponde ao valor real e linha laranja ao valor predito. Cenário: Rede sem fio. Local: ICE	69
A.11	Comparando valor predito com valor real para modelo MLP com função <i>Softplus</i> . Linha azul corresponde ao valor real e linha laranja ao valor predito. Cenário: Rede sem fio. Local: ICE	70
A.12	Comparando valor predito com valor real para modelo GRU com função <i>Softplus</i> . Linha azul corresponde ao valor real e linha laranja ao valor predito. Cenário: Rede sem fio. Local: ICE	70
A.13	Comparando valor predito com valor real para modelo MLP com função ReLU. Linha azul corresponde ao valor real e linha laranja ao valor predito. Cenário: Rede sem fio. Local: IAD	71
A.14	Comparando valor predito com valor real para modelo GRU com função Logística. Linha azul corresponde ao valor real e linha laranja ao valor predito. Cenário: Rede sem fio. Local: IAD	71
A.15	Comparando valor predito com valor real para modelo MLP com função <i>Softplus</i> . Linha azul corresponde ao valor real e linha laranja ao valor predito. Cenário: Rede sem fio. Local: IAD	72

A.16 Comparando valor predito com valor real para modelo GRU com função *Softplus*. Linha azul corresponde ao valor real e linha laranja ao valor predito. Cenário: Rede sem fio. Local: IAD 72

Lista de Tabelas

3.1	Comparação entre os trabalhos relacionados.	36
4.1	Intervalo dos parâmetros que serão otimizados, constituído por limite inferior e superior.	40
5.1	Resumo dos melhores resultados obtidos para cada função de ativação usada em cada modelo para o cenário de tráfego de veículos. Melhor resultado de cada modelo destacado em negrito.	48
5.2	Resumo do tempo de execução médio e total pra cada combinação dos modelos com as FAs. Cenário: Tráfego de Veículos. Melhor resultado de cada modelo destacado em negrito.	51
5.3	Resumo dos melhores resultados obtidos para cada função de ativação usada em cada modelo para o cenário de rede sem fio. Local: Engenharia. Melhor resultado de cada modelo destacado em negrito.	52
5.4	Resumo do tempo de execução médio e total pra cada combinação dos modelos com as FAs. Cenário: Rede sem fio. Local: Engenharia.	52
5.5	Resumo dos melhores resultados obtidos para cada função de ativação usada em cada modelo para o cenário de rede sem fio. Local: ICE. Melhor resultado de cada modelo destacado em negrito.	54
5.6	Resumo do tempo de execução médio e total pra cada combinação dos modelos com as FAs. Cenário: Rede sem fio. Local: ICE.	54
5.7	Resumo dos melhores resultados obtidos para cada função de ativação usada em cada modelo para o cenário de rede sem fio. Local: IAD. Melhor resultado de cada modelo destacado em negrito.	56
5.8	Resumo do tempo de execução médio e total pra cada combinação dos modelos com as FAs. Cenário: Rede sem fio. Local: IAD.	57

Lista de Abreviações

DCC	Departamento de Ciência da Computação
UFJF	Universidade Federal de Juiz de Fora
FA	Função de Ativação
RN	Rede Neural
RNA	Rede Neural Artificial
PSO	Otimização por Enxame de Partícula
ReLU	<i>Rectified Linear Unit</i>
STI	Sistemas de Transporte Inteligente
EP	Elementos de Processamento
LSTM	<i>Long Short Term Memory</i>
GRU	<i>Gated Recurrent Unit</i>
AG	Algoritmos Genéticos
Tor	<i>The Onion Router</i>

1 Introdução

Nos últimos anos a busca por soluções mais eficientes em todos os campos de pesquisa vem crescendo. Essa busca tem diversas motivações, e aquela que se destaca neste trabalho é o intuito de melhorar a qualidade de vida da população, aperfeiçoando o ambiente de modo a deixá-lo inteligente. Com isso, desenvolver e evoluir técnicas computacionais que possam vir a facilitar a vida da população, torna-se essencial para o aprofundamento neste assunto. Diversas técnicas como a análise, o monitoramento de dados e a tomada de decisão, envolvem uma pequena parte deste contexto que auxilia no aprimoramento do cenário no qual está sendo aplicado. Tendo isso em mente, este trabalho tem como propósito otimizar e analisar o desempenho de modelos preditivos quando aplicados em ambientes inteligentes¹.

A busca por métodos inteligentes que auxiliam em áreas da computação como aprendizado de máquina, tornou-se bem comum nas últimas décadas (Jiang et al, 2016). Pode ser destacado na área de Inteligência Artificial, o uso de Redes Neurais (RN) que já é utilizada para diferentes propósitos, caracteriza-se por ser um modelo computacional que tem como objetivo reproduzir o sistema nervoso do ser-humano (Guresen et al, 2011).

Muitas das redes neurais populares utilizam-se dos algoritmos de retro-propagação ou gradiente para treinamento, sua arquitetura é disposta em camadas, sendo definidas por: camada de entrada, escondida e saída, que por sua vez, possuem um determinado número de neurônios que são interligados, onde esses neurônios resultarão na representação da célula nervosa (Dlugosz et al, 2018). Entretanto, nunca foi uma tarefa fácil essa reprodução, assim como também, o seu ajuste de valores dos parâmetros (*i.e.* variáveis que influenciam no funcionamento do modelo).

Para entender o funcionamento de alguns parâmetros que podem ser manipulados em uma rede neural, destacamos entre esses, a Função de Ativação (FA) que tem um papel importantíssimo em sua escolha. O principal objetivo da FA é determinar se um

¹Ambientes inteligentes neste trabalho compreendem: cidades inteligentes, campus inteligentes, entre outros quaisquer, onde há a necessidade de entendimento do histórico do comportamento do ambiente e seus atores para o desenvolvimento de evoluções tecnológicas que facilitem a vida do cidadão.

neurônio é ativado ou não, e isso é feito por limitar o valor que um neurônio repassa para as próximas camadas (Fleck et al, 2016). A escolha desta FA requer um estudo mais aprofundado, já que diversas funções já foram propostas e cada uma possui um comportamento particular. Para auxiliar neste entendimento, são citados exemplos de algumas das funções amplamente utilizadas na atualidade, como: função Sigmoide (Menon et al, 2011), também conhecida por função Logística, e a função Unidade Linear Retificada que também pode ser encontrada por função ReLU (Agarap, 2018). Esta última, ressalta-se que é uma das mais utilizadas na literatura com um alto índice de bons resultados dentre as mais diversas FAs criadas ou modificadas (Bircanoğlu et al, 2018), sendo também, mais utilizada mundialmente por ser aplicada em Aprendizagem Profunda e Redes Neurais Convolucionais (Agostinelli et al, 2014).

No conceito de ambientes inteligentes busca-se a utilização de métodos eficientes para facilitar o cotidiano das práticas urbanas, como as atividades relacionadas ao meio ambiente, culturais e econômicas (Abuarqoub et al, 2017). Considerando isso, o foco será a aplicação de uma abordagem no gerenciamento de tráfego de veículos, que é uma parte do Sistemas de Transporte Inteligente (STI), e no volume de uso da rede sem fio. Analisar os dois casos tem como objetivo, uma melhor administração e orientação de recursos (Dalal et al, 2017; Loce et al, 2017). A partir disso, modelos que têm a finalidade de predição de tráfego e da carga são de extrema importância para criação de um ambiente autônomo e inteligente (Huang et al, 2014; Qolomany et al, 2017; Medeiros et al, 2019). Por consequência, obter uma melhor concepção do tráfego de veículos, e isso agregado com uma validação de um modelo de previsão eficiente, podem auxiliar ao público a ter melhores resultados no dia-a-dia, de acordo com uma decisão tomada. Da mesma forma, entender e prever a carga em pontos de acesso (APs) em redes sem fio conforme a utilização por número de usuários auxilia o administrador de redes a dimensionar sua rede, compreender o comportamento de usuários, realizar o balanceamento de carga e até gerar economia de energia (*e.g.* desligando equipamentos com pouca ou nenhuma utilização), por exemplo.

A busca por ajustes de parâmetros é comum de se ver nesta área de pesquisa. Diante deste cenário, existem alguns trabalhos direcionados para este meio que buscam

aplicar técnicas que otimizam estes parâmetros (Qolomany et al, 2017; Hu et al, 2014). Porém, cada cenário pode requerer técnicas diferentes que se adaptam melhor ao padrão do ambiente. Treinar modelos preditivos precisos nem sempre é uma tarefa fácil, e pensando nos ambientes inteligentes em que podem ser aplicados, ter um modelo que adapte da melhor forma ao perfil do cenário é importante, pois ações podem ser tomadas baseadas nestas informações. Para isto, mostra-se necessário utilizar algum método que guie e auxilie a escolher estes parâmetros de forma eficaz.

1.1 Justificativa/Motivação

O campo de pesquisa em aprendizagem de máquina e aprendizagem profunda está em constante crescimento. A busca por aplicações mais eficientes é uma motivação para aplicar estas técnicas em diversos cenários, principalmente quando trata-se de ambientes inteligentes, quando o objetivo principal é tornar a vida de todos mais simples. Porém, utilizar estas técnicas em um ambiente inteligente requer que seja feito um estudo prévio para determinar, por exemplo, quais parâmetros ou modelos se adaptam melhor.

Dentre os cenários que serão aplicados, a predição de tráfego de veículos contribui para a área de cidades inteligentes. Como exemplos, esta contribuição gera planejamentos que se relacionam a vagas de estacionamentos para eventos, fluxo de veículos em uma rua e horários de pico em que a situação demonstra engarrafamentos. Esses são cenários importantes onde a aplicação de modelos preditivos mostram-se necessária para auxiliar no gerenciamento do tráfego. Já para o cenário de rede sem fio, a previsão de número de usuários influencia para um gerenciamento melhor da rede como escolha de rotas alternativas ou aumento de largura de banda para o usuário, caso haja sobrecarga na rede e, desligar e ligar equipamentos pensando no consumo de energia - *Green Computing* - relacionado também a cidades inteligentes. Outro ponto que pode ser citado que envolve a aplicação de técnicas preditivas em rede sem fio, é a divulgação de anúncios em shopping, onde ter conhecimento de horários e lugares que possuem uma alta quantidade de carga de usuários é o foco para alcançar um maior número de pessoas que visualizam o anúncio.

1.2 Objetivos

Este trabalho tem como propósito utilizar uma meta-heurística para otimizar os parâmetros de dois modelos de redes neurais distintos variando entre três funções de ativação bem conhecidas na literatura. Para avaliar o desempenho destes modelos dois cenários distintos servirão como estudo de caso, sendo estes: fluxo de tráfego de veículos e carga (em número de usuários) em redes sem fio.

2 Fundamentação Teórica

Neste capítulo serão abordados conceitos e informações sobre a estrutura e funcionalidade de duas redes neurais. Também será visto o embasamento sobre as funções de ativação que serão utilizadas neste trabalho. Por fim, a meta-heurística usada para otimizar os parâmetros das redes neurais será apresentada.

2.1 Redes Neurais Artificiais

As Redes Neurais Artificiais (RNA) são estruturas que se baseiam em modelos matemáticos para simular o funcionamento do sistema nervoso de um ser vivo. Segundo (Principe et al, 2000), as RNAs podem ser definidas como sendo geralmente uma máquina de aprendizagem não-linear distribuída e adaptativa que possui diversos Elementos de Processamento (EP). Por sua vez, estes EP podem receber conexões próprias ou de outros EP, e tem como objetivo carregar funções aritméticas por toda sua estrutura. Desta forma, os EPs que também são denominados como neurônios, tem como objetivo entender e emular paralelismo em uma rede densa, assim como é o sistema nervoso.

Uma RNA não somente possui um elevado número de neurônios, como também estabelece uma ligação entre estes neurônios que recebe um valor que define o grau de conexão entre estes, esta conexão é comumente denominada por peso ou sinapse (Furtado, 2019). De acordo com (Furtado, 2019), apesar de cada neurônio realizar um processamento bem simples, o conjunto de diversos neurônios possibilita na resolução de problemas mais complexos, dependendo apenas de como é definido sua arquitetura (*e.g.* número de neurônios, definição dos pesos, número de camadas).

Como pode ser visto na Figura 2.1, um simples neurônio pode ser composto por n entradas X , e para cada uma destas entradas é associado um peso w . O neurônio processa essas entradas somando-as e, em seguida, é aplicada uma função de ativação sobre seu resultado. Tal resultado é enviado para os demais neurônios das camadas seguintes, ou (se localizado na última camada) representará o resultado da rede neural. Assim, para

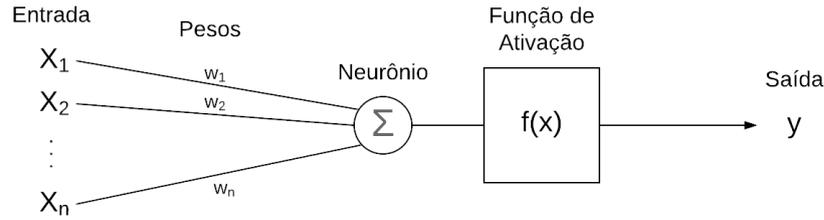


Figura 2.1: Composição da estrutura de um neurônio. Fonte (Furtado, 2019).

o treinamento de uma RNA, é necessário alimentá-la com um conjunto de informações, também denominado como vetor de características, de modo que a RNA aprenda um pouco sobre o que foi apresentado. Para alcançar tal objetivo, é preciso que a rede neural atualize seus pesos minimizando uma função de erro da rede. Existem diversas técnicas que auxiliam na atualização destas sinapses, porém uma das mais utilizadas na atualidade é o algoritmo de retro-propagação. Segundo (Russell et al, 2016): “O método mais popular para aprendizado em múltiplas camadas é chamado retro-propagação”. Este método tem como objetivo utilizar da regra Delta, que é um aprendizado do tipo gradiente descendente (Kingma et al, 2014). Esse tipo de aprendizado atualiza os pesos por meio de cálculos que levam em consideração o erro obtido em um dado conjunto a partir da derivada da função de ativação e da taxa de aprendizagem (Russell, 2012).

$$y_k = f \left(\left(\sum_{i=1}^m w_{ki} * x_i \right) + b_k \right) \quad (2.1)$$

Na Equação 2.1 é possível explicar matematicamente o funcionamento do neurônio descrito na Figura 2.1. Sendo que, para cada neurônio k , é feito um somatório de cada entrada x_i com seu peso w_{ki} e ao final somado a um bias b_k , que é uma constante de excitação de valor igual 1 (Furtado, 2019). Mais a frente será abordado um pouco mais sobre a função f apresentada nessa equação.

Outro parâmetro que influencia no funcionamento das redes neurais é o número de épocas que, por sua vez, determina a quantidade de vezes que a rede neural irá treinar sobre o conjunto de dados. Este parâmetro é importante, pois o aumento consciente deste pode melhorar o desempenho do modelo, entretanto pode ocasionar outros problemas como *overfitting*, que é quando o modelo se adapta demais a um conjunto, e não funciona

bem para prever novos dados.

As redes neurais podem ser classificadas entre várias categorias diferentes, sejam estas: redes neurais *feedforward*, recorrente e convolucionais. As seções seguintes apresentam o embasamento teórico dos dois primeiros modelos citados, uma vez que são os adotados neste trabalho.

2.1.1 Modelo *Feedforward*

Considerando o que já foi visto na seção anterior, o modelo *Feedforward* se utiliza da mesma estrutura do neurônio apresentado na Figura 2.1 e na Equação 2.1 para processar os dados. Porém, neste modelo, como o próprio nome sugere, o dado de entrada segue um único sentido, para frente, uma vez que entra na rede neural, segue o fluxo desta até o neurônio de saída (Sazli, 2006).

As redes neurais tiveram início na década de 50, e logo em seguida, houve o surgimento do primeiro *Perceptron*, no qual seu foco inicial seria com processamento de camada única (Rosenblatt, 1958). Tal técnica teve grande aplicação na área de classificação, principalmente se tratando de classificação binária onde o objetivo é mapear um série de entradas X para saída $f(X)$, onde $f(X)$ é uma função limiar.

Na Figura 2.2(a) e 2.2(b), pode ser vista uma estrutura de rede neural *feedforward*. Porém, o que as diferenciam, além do número de neurônios, é que na primeira tem-se uma camada oculta, classificando esta como múltiplas camadas e com dois neurônios de saídas; já na segunda, não há camada oculta, assim sendo classificada como rede neural de camada única com quatro neurônios na camada de saída.

2.1.2 Modelo Recorrente

Já as redes neurais recorrentes são caracterizadas por executar a mesma tarefa para cada elemento de uma sequência, com a saída sendo dependente dos cálculos anteriores. Este modelo teve início na década de 80 quando John Hopfield relacionou redes recorrentes auto-associativas com sistemas físicos, abrindo espaço para a utilização de teorias para aprofundar em tais modelos. A rede Hopfield, que deu início às redes neurais recorrentes, é utilizada como uma rede de memória associativa, entre suas principais características

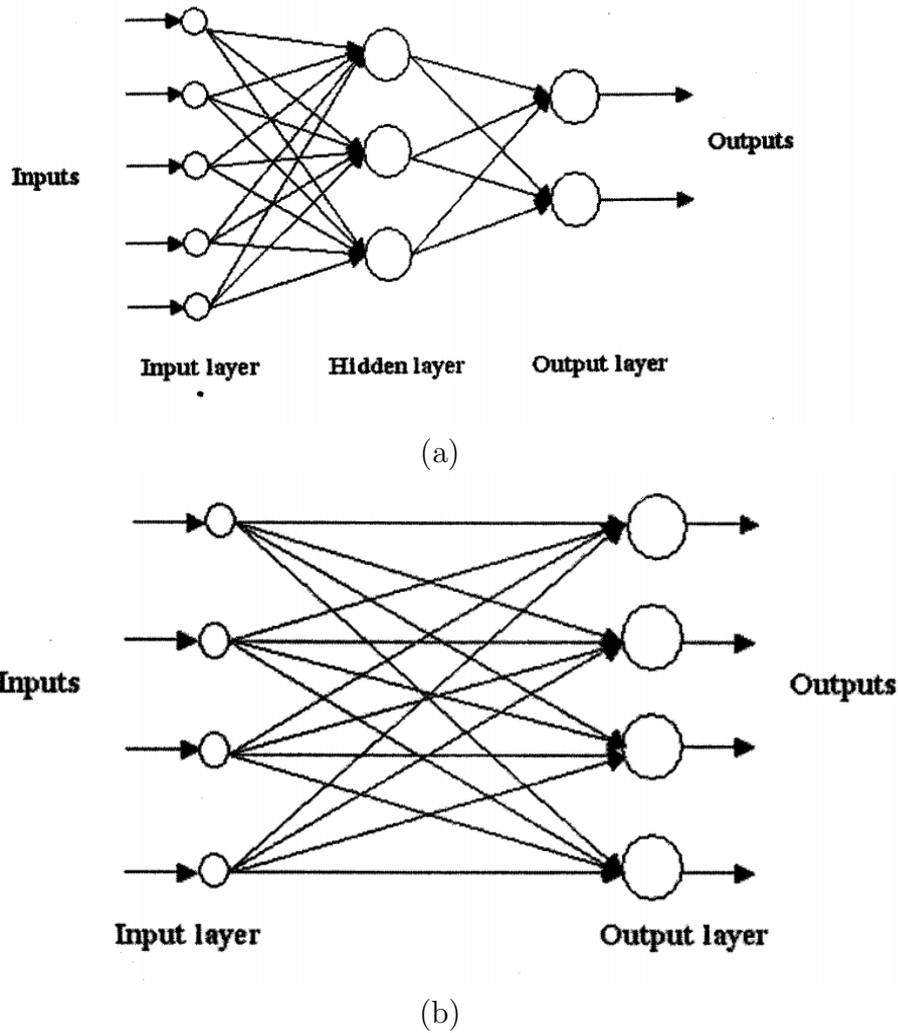


Figura 2.2: (a) Arquitetura de uma rede neural *Feedforward* com múltiplas camadas e duas saídas; (b) Arquitetura de uma rede neural *Feedforward* sem camada oculta e com 4 saídas. Fonte (Sazli, 2006).

tems que cada unidade é conectada a todas as outras sem permitir *self-loops* (Hopfield, 1982).

Desde Hopfield, as redes neurais recorrentes foram sendo estudadas e evoluídas ao longo do tempo, e um dos modelos mais famosos da atualidade é a rede neural *Long Short Term Memory* (LSTM) que tem grande aplicação em modelos de linguagem e que apresenta bons resultados em situações de dependência a longo prazo (Hochreiter et al, 1997). Entretanto, este modelo tem sua desvantagem quando se trata em tempo de processamento devido à sua complexa estrutura, e para isso a rede neural *Gated Recurrent Unit* (GRU) foi proposta para acelerar este tempo de processamento com características similares a LSTM, porém mais simples de computar e implementar (Cho et al, 2014).

Como é exemplificado na Figura 2.3, a célula de processamento de uma GRU é definida por dois portões, z e r . O portão z (que é o portão de atualização), serve para decidir qual dado não é importante, logo será desconsiderado, e qual será mantido e processado pela rede. Já o portão r , que é definido como um portão de redefinição, é usado para decidir o quanto dos dados já passados serão esquecidos (Cho et al, 2014). A rede neural cria um estado oculto que representa os dados processados no instante de tempo t , de tal modo, a rede processa um estado oculto no tempo t utilizando o estado oculto do tempo $t - 1$ com a informação de entrada no tempo t .

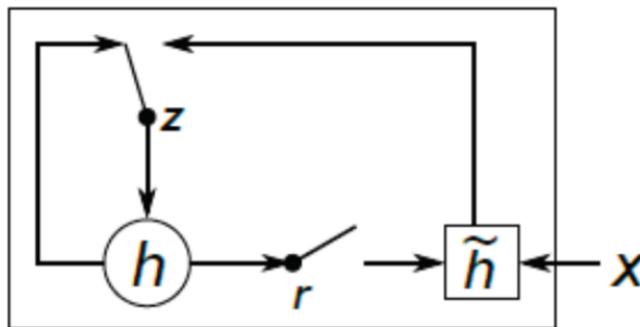


Figura 2.3: Exemplificando arquitetura de uma célula em rede neural GRU. Fonte (R. Fu et al, 2016)

2.2 Função de Ativação

A função de ativação tem um importante papel quando se trata de redes neurais, além de transformar as somas dos pesos com suas respectivas entradas. De acordo com a Equação 2.1 para um novo intervalo, a função serve para indicar quando um neurônio está ativo e assim escolher a função para um modelo ou problema influencia diretamente na convergência da rede neural (Gomes et al, 2011).

Para este trabalho foram selecionadas três funções de ativação com base em sua adoção pela literatura para estudo e comparação (Bircanoğlu et al, 2018). A seguir, é mostrado o comportamento de cada uma destas funções, como também de suas derivadas.

2.2.1 Logística

A função Logística é uma das funções mais antigas utilizadas e tem a característica marcante de não-linearidade – esta função também é conhecida por função sigmóide por conta de seu comportamento em forma de S (Cybenko, 1989). Por um tempo, as redes neurais utilizavam funções lineares que limitavam seu potencial de resolver problemas mais complexos, com o uso de funções não-lineares, o poder de mapeamento das redes neurais cresceram em conjunto com seu processamento.

$$f(x) = \frac{1}{1 + e^{-\alpha x}} \quad (2.2)$$

$$f'(x) = \frac{\alpha e^{-\alpha x}}{(1 + e^{-\alpha x})^2} \quad (2.3)$$

Nas Equações 2.2 e 2.3, pode-se ver a definição da função logística e sua derivada, respectivamente. Dentre suas características, são destacadas que a função limita seus resultados entre o intervalo de $[0,1]$ (veja Figura 2.4) e possui derivada não constante para todo o domínio como pode ser visto na Figura 2.5.

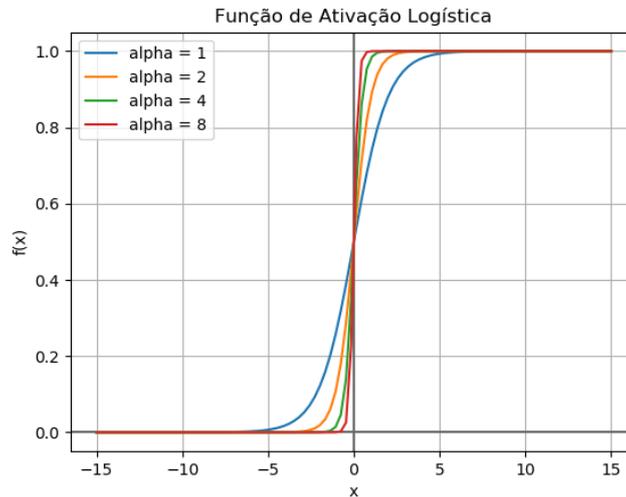


Figura 2.4: Gráfico da função de ativação logística com seu parâmetro α sendo variado.

Na Figura 2.4 pode ser visto que ao variar o parâmetro α da função, sendo $\alpha = 2^x$, onde $x = [0,1,2,3]$, a curva da função fica mais fechada se aproximando da origem de acordo que x aumenta. Seguindo esta mesma ideia, variando o parâmetro α para a

derivada, os valores de $f'(x)$ aumentam de acordo que x aumenta, porém o intervalo onde $f'(x) > 0$ diminui. Por padrão, e através dos testes realizados previamente, é usado o $\alpha = 1$.

Um problema que a função logística encontra é quando utiliza da otimização por retropropagação. O problema do desaparecimento do gradiente é conhecido e acontece pelo fato de a derivada destas funções tenderem a ficar cada vez menores fazendo com que os neurônios aprendam muito devagar (Hochreiter et al, 1997).

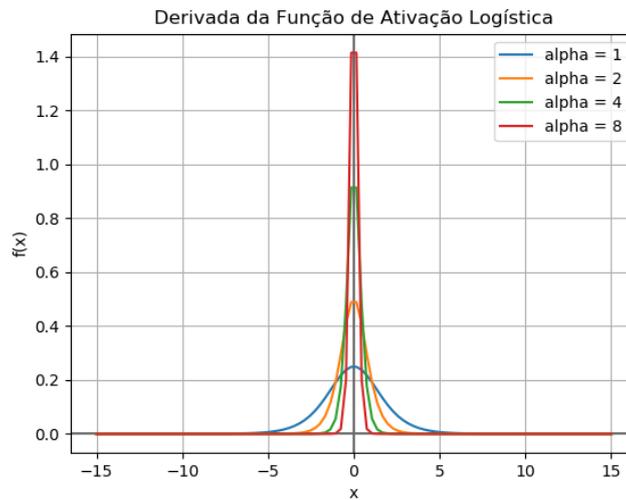


Figura 2.5: Gráfico da função de ativação logística com seu parâmetro α sendo variado.

2.2.2 ReLU

A *Rectified Linear Unit* (ReLU) é uma função que, assim como a logística, também é definida como não-linear, e logo pode ser utilizada como função de ativação entre as redes neurais para aumentar seu poder de processamento (Cybenko, 2010). Outro ponto, como pode ser visto nas Equações 2.4 e 2.5, a ReLU ganha em relação as suas concorrentes por sua função e derivada ser de fácil cálculo, fazendo com que a rede neural processe mais rapidamente. Uma das grandes áreas que a ReLU está sendo mais aplicada atualmente é a de aprendizagem profunda, e isso se deve, principalmente, pelo seu desempenho (Maas et al, 2013).

$$f(x) = \max(0, x) \quad (2.4)$$

$$f'(x) = \begin{cases} 1 & \text{se } x > 0 \\ 0 & \text{se } x \leq 0 \end{cases} \quad (2.5)$$

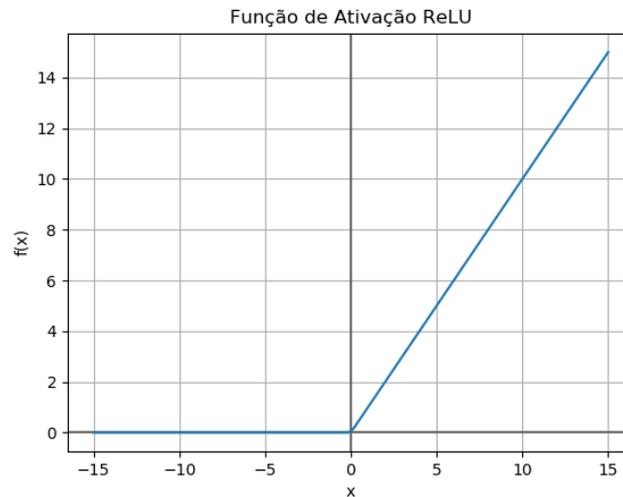


Figura 2.6: Gráfico da função de ativação ReLU.

Uma de suas vantagens é conseguir resolver o problema do desaparecimento do gradiente que as funções sigmóides encontram, porém outro problema que acontece e pode ser visto na Figura 2.6, é que para valores negativos, a saída é sempre 0, fazendo com que neurônios que atinjam este valor, possivelmente não conseguirão se recuperar. Um ponto importante que vale ressaltar, é que, por definição, é adotado que a derivada da ReLU para $x = 0$, é igual a 0. Logo, tem a continuidade que pode ser vista na Figura 2.7.

2.2.3 Softplus

A função *Softplus* foi introduzida no início dos anos 2000 por (Dugas et al, 2001) e é definida pela Equação 2.6, e a sua derivada, que já foi introduzida anteriormente, e que também é conhecida por função Logística, está definida na Equação 2.7.

$$f(x) = \ln(1 + e^x) \quad (2.6)$$

$$f'(x) = \frac{1}{1 + e^{-x}} \quad (2.7)$$

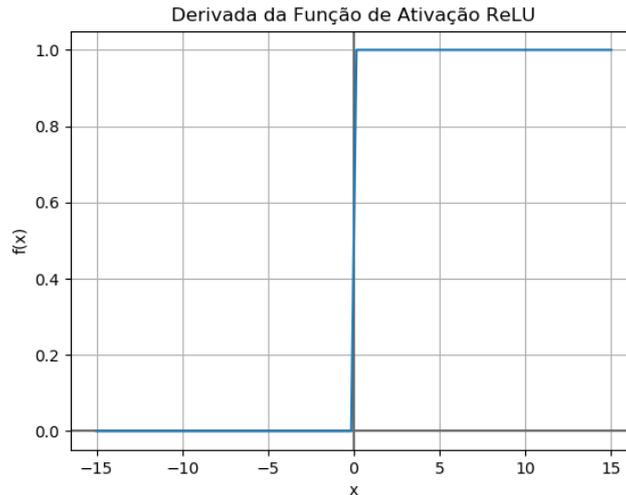


Figura 2.7: Gráfico da função de ativação ReLU.

Com um comportamento de função bem similar com a ReLU, a função *softplus* tende a ter um funcionamento parecido dentro da rede neural (Glorot et al, 2011). Entretanto, em comparação ao processamento, a ReLU se mostra superior à *softplus*. Todavia, uma característica a qual essa função de ativação se destaca é em ser diferenciável em todo o domínio, com derivada correspondente à função logística com valor de $\alpha = 1$. Esta igualdade pode ser observada comparando a Figura 2.9 com a Figura 2.4.

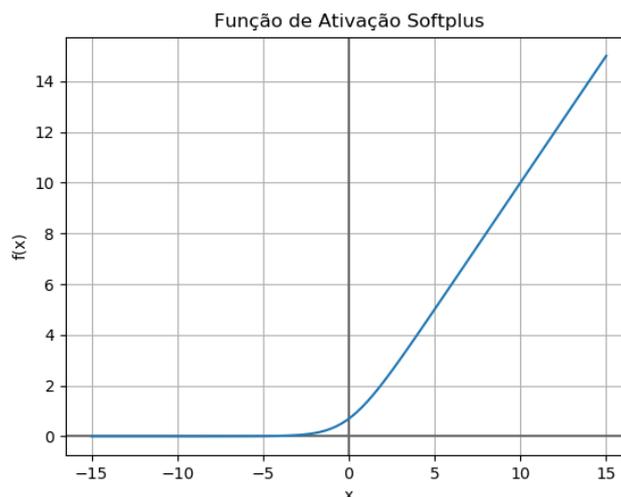


Figura 2.8: Gráfico da função de ativação softplus.

Ao se comparar a curva da função ReLU (Figura 2.6) com a função *softplus* (Figura 2.8), pode ser considerado que o comportamento assemelha-se, exceto com valores

próximo a 0, onde a *softplus* tem por vantagem por ser mais suave. Outra vantagem é que $f(x)$ possui um pequeno intervalo onde seus valores são maiores que 0 para $x < 0$, logo, possivelmente não terá a característica negativa da ReLU em não conseguir recuperar alguns neurônios.

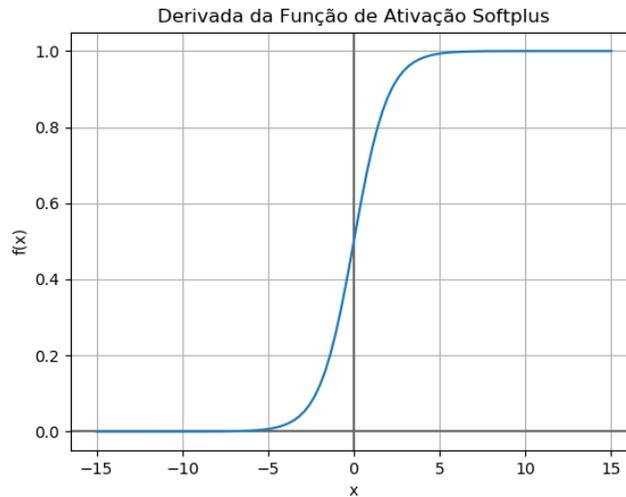


Figura 2.9: Gráfico da derivada da função de ativação softplus.

2.3 Otimização por Enxame de Partículas

A Otimização por Enxame de Partículas (PSO) é uma técnica de otimização baseada em uma população, inspirada pelo comportamento de pássaros e peixes (Eberhart et al, 1995). Seu funcionamento leva em consideração que cada indivíduo possui uma posição no espaço, e dada uma velocidade para este indivíduo, ele se movimenta a fim de melhorar a solução do problema de acordo com uma função objetivo. Entre os métodos evolucionários, o PSO possui algumas similaridades com Algoritmos Genéticos (AG), (*e.g.* por trabalhar com população e geração inicial aleatórias, e se guiar por uma função objetivo), porém o PSO se diferencia de tal técnica por não possuir mutação e recombinação (Eberhart et al, 1998).

A técnica PSO tem o funcionamento que se descreve a seguir. Primeiramente, é necessário definir o número de partículas e o número de iterações que atuará sobre a população. Após isto, é necessário iniciar cada partícula com uma posição e velocidade,

e assim, esta será avaliada em relação a função objetivo. E então, em cada iteração, é escolhido o melhor resultado pra cada partícula, e também a melhor partícula da população com intuito de direcionar as partículas pelo espaço através do melhor caminho já encontrado. Caso o critério de parada não seja atingido, uma nova velocidade e posição será calculada, e novamente avaliada até atingir tal critério.

Com isso, pode ser ilustrado como o movimento das partículas, definido pelas Equações 2.8 e 2.9, segundo (Eberhart et al, 1995). A Equação 2.8 calcula a próxima velocidade a ser utilizada, e para isto é utilizada a velocidade atual da partícula v_i^t em conjunto com um peso w , que corresponde ao fator de inércia da partícula. Já x_i^t corresponde à posição atual da partícula, assim como, $pBest$ equivale à melhor posição encontrada pela partícula i e $gBest$ à melhor posição conhecida pela população até o momento. Já as constantes c_1 e c_2 são coeficientes de aprendizado que servem para auxiliar no cálculo junto com um valor aleatório $rand$ gerado entre 0 e 1. Agora para o cálculo da nova posição da partícula, é usado a Equação 2.9, onde ambos os termos v_i^{t+1} e x_i^t já são conhecidos.

$$v_i^{t+1} = w.v_i^t + c_1.rand.(pBest - x_i^t) + c_2.rand.(gBest - x_i^t). \quad (2.8)$$

$$x_i^{t+1} = x_i^t + v_i^{t+1} \quad (2.9)$$

Outro ponto importante para o início da aplicação desta técnica, é definir o espaço de busca por onde as partículas irão se mover, e para isso, é necessário definir os limites para as posições, assim como os limites de velocidade para que se possa gerar a população inicial com as partículas em posições aleatórias.

Empregar uma técnica de otimização para melhorar a solução de um problema é bem comum quando se trata de modelos de predição com diversos parâmetros (Zhang et al, 2012). Para isto, o PSO tem grande aplicação na área, envolvendo otimização de modelos preditivos distintos (Hu et al, 2014; Qolomany et al, 2017). Logo, o PSO será utilizado para otimizar parâmetros como número de neurônios e taxa de aprendizagem dos modelos utilizados neste trabalho.

3 Trabalhos Relacionados

Neste capítulo são apresentados os trabalhos relacionados à esta pesquisa. Após a introdução do estado da arte, é sumarizada uma tabela de comparação entre os trabalhos.

3.1 *Comparison of Activation Functions in Multi-layer Neural Network for Pattern Classification*

No trabalho de Hara et al (1994), é feito um estudo de funções de ativação com a utilização de redes neurais de várias camadas em um cenário de sinais de múltiplas frequências. O problema abordado tem como foco a classificação de padrões, analisando e comparando as implicações das diferentes funções de ativação e a combinação entre elas. Dentre as funções abordadas no trabalho estão: a Sigmoide, Gaussiana e Senoide com intuito de estudar o seu comportamento.

A princípio, foram feitas análises das taxas de convergência das três funções separadamente, com o uso de uma e três unidades ocultas na rede neural. Os resultados obtidos mostram que a função gaussiana demonstrou ser a que converge mais rápido, porém, a senoidal por ser uma função periódica, atingiu um resultado melhor de acurácia.

Diversas combinações foram feitas entre essas funções de ativação, onde obtiveram o resultado que a combinação das três funções abordadas tiveram um bom resultado em relação a acurácia, porém, a combinação de três funções gaussianas teve o melhor resultado superando o uso das três funções distintas. Entretanto, os resultados dessas combinações ao serem comparados em suas aplicações isoladas de cada função, demonstram que não possuem vantagens em seu uso.

Por fim, é concluído que a função Sigmoide não foi útil com uso de uma única unidade oculta. Já com a função Gaussiana, os autores obtiveram a melhor performance quando se trata de sinais de entrada sem ruído. Mas entre todas, a com maior destaque entre dados com e sem ruídos foi a Senoidal, que mostrou a menor taxa de erro.

3.2 A Comparison of Two Sigmoidal-Type Activation Functions in Video Game Controller Evolution

Em Tan et al (2011) é apresentada a comparação entre funções de ativação aplicada a redes neurais, desta vez com foco no cenário de jogos. O jogo em questão, usado neste trabalho foi o *Ms. Pac-man*, lançado na década de 80 se tornando um dos jogos mais populares, desde então.

A proposta sugerida foi a Rede Subida da Colina (*Hill Climb Network*) que é uma combinação do algoritmo Subida da Colina (HC - *Hill Climb*) com uma Rede Neural *feedforward*. O algoritmo HC tem como objetivo otimizar os vetores de pesos e bias com uma distribuição Gaussiana se guiando por uma função objetivo usando operações como mutação. Por sua vez, foram utilizadas as funções de ativação Logística Sigmoide e a Tangente Sigmoide como um dos parâmetros da Rede Neural para avaliar seu desempenho no resultado.

Como é destacado pelo autor, o objetivo da rede é determinar uma direção pelo labirinto que o agente deve tomar considerando os elementos disponíveis. Esses elementos são a distância do agente até os seguintes itens: pílula, pílula de energia, fantasma normal, fantasma comestível e as frutas. Esse conjunto de elementos são utilizados para treino da rede neural a fim de determinar a direção na qual o agente deve fazer seu movimento (Tan et al, 2011).

Após 500 avaliações, em 10 execuções dessa proposta, os resultados obtidos a partir de suas comparações demonstram que a função logística teve uma performance melhor em 9 das 10 execuções comparado à tangente hiperbólica. Concluindo-se que um dos possíveis motivos para esse resultado deve-se ao fato do intervalo dessa função ser maior, vindo que ela varia entre $[-1,1]$ e a função logística entre $[0,1]$.

3.3 A Neural Network Based User Identification for Tor Networks: Comparison Analysis of Different Activation Functions Using Friedman Test

Para o trabalho de Oda et al (2016), os estudos de funções de ativação foram mais longe, trazendo diversas funções para análise dos resultados de uma rede neural, aplicada na classificação da identificação do usuário em uma rede *The Onion Router*, mais conhecido como Tor. Segundo o autor, Tor é definido por “(...) um dos vários sistemas que foram desenvolvidos para fornecer aos usuários da Internet com um alto nível de privacidade e anonimato, a fim de lidar com as medidas de censura tomadas pelas autoridades e proteger contra as crescentes ameaças (...)” (Oda et al, 2016).

Neste estudo, foram utilizadas 6 funções de ativação para comparação, sendo elas:

- Sigmoides,
- Tangente Hiperbólica,
- *Rectified Linear Unit* (ReLU),
- *Softplus*,
- *Softsign*,
- *Softsign/x*.

Para testes, foi adotada uma rede neural com o algoritmo de retropropagação para as correções necessárias. Já para análise, foi utilizado o teste de *Friedman*, que é uma métrica estatística para medir testes de múltiplos grupos. Após várias simulações, foi concluído que a função que melhor se adaptou ao problema foi a *softsign/x*, por apresentar o menor índice de diferença entre servidor e cliente, sendo indicada para problemas de identificação de usuários maliciosos em redes Tor.

3.4 On Hyperbolic Sine Activation Functions Used in ZNN for Time-Varying Matrix Square Roots Finding

Em busca de resolver o problema de encontrar raízes quadradas de uma matriz variante no tempo, os autores de Zhang et al (2012), propuseram o uso de uma Rede Neural Zhang (RNZ), que é uma variação da Rede Neural Recorrente, porém direcionada para o problema específico.

Para demonstrar a eficácia da técnica proposta, análises matemáticas foram realizadas em relação a duas funções de ativação: função linear e função seno hiperbólica. A partir de uma análise inicial, foi notado que a taxa de convergência de uma rede neural pode convergir mais rapidamente com a função seno hiperbólica do que com a linear.

Após isso, para analisar mais a fundo a performance de convergência, é utilizada métrica de erros residuais para sua comparação, o que demonstra que seus valores decrescem rapidamente para zero. Segundo o autor a taxa de convergência da função seno hiperbólica é aproximadamente cinco vezes mais eficaz que sua rival.

Ao fim do trabalho, são destacados métodos numéricos direcionados para esse mesmo problema, conforme citado pelo autor: “(...) quatro métodos numéricos (*i.e.*, iteração de Newton, iteração DB, iteração CR e iteração IN) são usados com frequência para resolver Raízes quadradas da matriz estática (...)” (Zhang et al, 2012). É ainda descrito que pós alguns testes adicionais foi possível comprovar que a utilização de RNZ se mostrou mais eficaz que os quatro métodos numéricos citados.

3.5 Implementation of Activation Functions for ELM based Classifiers

O impacto que as funções de ativação podem ter sobre um modelo de predição não fica apenas restrito ao nível de *software*, também podendo ser utilizados para o nível de *hardware*, e é demonstrado no trabalho de Baraha et al (2017).

Utilizando *Extreme learning machine* (ELM), que segundo o autor é: “(...) uma ferramenta poderosa que é amplamente usada para treinar redes neurais *feedforward* de camada única ” (Baraha et al, 2017). Os nós ocultos desse modelo possuem uma liberdade maior ao serem escolhidos aleatoriamente e também porquê contam com parâmetros como pesos de entrada em que não são necessários ajustes manuais, por serem calculados analiticamente. (Baraha et al, 2017)

Dentre suas vantagens em comparação com uma Rede Neural tradicional, estão: velocidade de aprendizado superior, erro de treinamento menor e uma melhor performance em generalização. A partir disso, são consideradas três funções de ativação para análise, cuja essas são: *Hard Limit*, Tangente Hiperbólica e Cosseno.

Após uma série de análises entre as três funções citadas, é concluído que a função mais rápida é a *Hard Limit*, entretanto, não é a mais eficiente. Para problemas de classificação binária, a Tangente Hiperbólica e a Cosseno demonstrou ter grande eficiência nessa área por apresentar uma acurácia similar, porém, em problemas de classificação múltipla, a cosseno se mostrou superior.

3.6 Parameters optimization of deep learning models using Particle swarm optimization

Verificando o benefício da adoção de uma meta-heurística, dois trabalhos em ITS (*Intelligent Transportation System*) são apresentados. O primeiro deles é Qolomany et al (2017), sendo nossa aplicação mais conhecida de uma meta-heurística para melhorar um modelo de rede neural. Neste artigo foram propostos dois mecanismos para aumentar o desempenho dos modelos. Em primeiro lugar, o uso da aprendizagem profunda, que é uma técnica recente que implica o uso de uma rede neural com múltiplas camadas. Com base nisso, a aprendizagem profunda ajuda a resolver um problema de previsão do número de ocupantes em um determinado local a 15, 30 e 60 minutos a partir da hora atual, via dados coletados pelo Wi-Fi. Para atingir o objetivo, os autores utilizam um método de inteligente computacional, ou meta-heurística, chamado Otimização por Enxame de Partículas (PSO). PSO, como já introduzido neste trabalho, é considerado um bom método iterativo

baseado em população para criar partículas que representam soluções, a fim de alcançar a melhor solução possível para uma função objetivo. Essas partículas usam formulações matemáticas para se mover no espaço de busca, sendo este o número de camadas e o número de neurônios que eles buscam otimizar. Após os resultados, eles concluíram que o PSO foi bem sucedido na redução do tempo de treinamento do modelo de 77 % para 85%.

3.7 Traffic flow forecasting with particle swarm optimization and support vector regression

Em Hu et al (2014), uma combinação de *Regressão por Vetor Suporte* (SVR) e PSO é proposta para melhorar os resultados de um problema de previsão de tráfego de veículos. Os autores realizaram uma comparação entre seu desempenho com outros métodos, como regressões lineares múltiplas e redes neurais. Neste caso, o PSO procura otimizar os parâmetros de entrada do SVR, que são: penalidade C, raio ϵ e função kernel. A métrica de avaliação utilizada como função de adequação foi a *Root Mean Square Error* (RMSE). Ao final, foi possível concluir que a combinação de PSO e SVR teve um impacto positivo, acrescentando um melhor desempenho em quase 10% em relação aos demais modelos. É necessário que fique clara a limitação de cenário e modelos comparados pelos autores, onde o uso do fluxo de tráfego não foi baseado em um banco de dados aberto, e os modelos comparados não foram os principais no estado da arte atual.

3.8 Using LSTM and GRU neural network methods for traffic flow prediction

Em R. Fu et al (2016), os autores propuseram o uso da rede neural Gated Recurrent Units (GRU), sendo esta uma modificação da rede neural Long Short Term Memory (LSTM) que tem como principal característica o armazenamento de informações, esta memória possibilita considerar entradas anteriores ao processar novas entradas, tornando-a ideal para trabalhar com série temporal. Enquanto o LSTM já vem sendo usado como modelo

de previsão de tráfego de fluxo desde 2015, a GRU até então não tinha sido usada para este meio. Sendo assim, eles avaliam a performance desses dois modelos com o modelo Auto Regressive Integrated Moving Average (ARIMA), onde nesse caso, eles concluíram que ambos os modelos propostos tiveram um desempenho melhor que o ARIMA, e que em 84% dos testes o modelo GRU superou o LSTM.

3.9 Applying a Multilayer Perceptron for Traffic Flow Prediction to Empower a Smart Ecosystem

Este trabalho Ferreira et al (2019) e o da próxima seção Frank et al (2019) foram publicados como resultados intermediários do desenvolvimento da pesquisa para esta monografia.

No trabalho Ferreira et al (2019), os autores aplicaram uma RNA Perceptron de Múltiplas Camadas (MLP) para prever número de veículos a partir de um certo horário. Tal proposta foi baseada em usar um MLP com 3 camadas ocultas, variando entre duas funções de ativação (Logística e ReLU) e treinar com dados históricos e recentes de uma série temporal de 2 cenários de fluxo de tráfego de veículos, sendo um com fluxo livre e o outro com fluxo atípico.

Por fim, o trabalho foi validado comparando os resultados obtidos a partir da técnica usada com o trabalho de Huang et al (2014), que é um dos trabalhos relacionando previsão de tráfego de veículos mais citados na literatura. Em ambos os cenários de testes, os resultados obtidos por Ferreira et al (2019) superou os da literatura, atingindo um MAPE mínimo de 4.68% para o cenário de fluxo livre e 13.46% para o de fluxo atípico.

3.10 *Multilayer Perceptron and Particle Swarm*

Optimization Applied to Traffic Flow Prediction on Smart Cities

Continuando parte do trabalho desenvolvido em Ferreira et al (2019), o artigo Frank et al (2019) propôs aplicar a meta-heurística PSO com intuito de otimizar os parâmetros de uma RNA MLP com 3 camadas ocultas. Tais parâmetros escolhidos para serem otimizados foram o número de neurônios e a taxa de aprendizagem. Os testes foram direcionados apenas para o cenário de tráfego de veículos de fluxo livre.

Após a otimização dos parâmetros do modelo, a avaliação se deu pela métrica MAPE, onde alcançou um valor mínimo de 3.1%. Permanecendo com resultados melhores que os obtidos por Huang et al (2014) e mostrando que a utilização da técnica PSO foi eficaz ao melhorar os resultados do cenário de fluxo livre.

3.11 Considerações Sobre o Capítulo

Dentre todos os trabalhos apresentados, em relação à FA, cada um deles segue um propósito parecido, mas, em geral, avaliando FAs diferentes, assim como cenários e modelos totalmente distintos. Sendo assim, em alguns casos, as funções estudadas ficam bem restritas ao problema aplicado, não permitindo uma comparação direta, mas demonstrando que uma FA pode se comportar melhor para um ou outro problema. Outro ponto a ser ressaltado é o número de funções estudadas por trabalho, exceto em Oda et al (2016) que são utilizadas seis funções de ativação, todos os outros trabalhos focam seu estudo entre duas ou três funções. Um fator interessante e oportunidade com relação aos trabalhos é, de forma geral, avaliá-los com um número maior ou novas FAs.

Algo que destaca-se é que, apesar de existirem diversas meta-heurísticas na atualidade, o PSO foi escolhido baseado nos resultados de Qolomany et al (2017); Hu et al (2014). Sendo assim, o PSO é aplicado para otimizar uma rede neural MLP e uma GRU, de modo que possa ser comparado o desempenho da técnica em dois tipos de redes neurais distintos em conjunto com diferentes funções de ativação.

	TR	Otimização	Modelo	FA's	Cenário
1	(Hara et al, 1994)	-	RNA	Sigmoide Gaussiana Senoide	Sinais de Múltiplas Freqüências
2	(Zhang et al, 2012)	HC	RNA	Logística TanH	Video-Game
3	(Oda et al, 2016)	-	RNA	Sigmoide TanH ReLU Softplus Softsign Softsign/x	Redes Tor
4	(Zhang et al, 2012)	-	RNR	Seno	Raízes Quadradas de Matriz Variante
5	(Baraha et al, 2017)	-	RNA	HardLimit TanH Cosseno	Classificação Binária
6	(Qolomany et al, 2017)	PSO	RNA	-	Rede Sem Fio
7	(Hu et al, 2014)	PSO	SVR	-	Tráfego de Veículos
8	(R. Fu et al, 2016)	-	RNR	Logística	Tráfego de Veículos
9	(Ferreira et al, 2019)	-	RNA	Logística ReLU	Tráfego de Veículos
10	(Frank et al, 2019)	PSO	RNA	Logística ReLU	Tráfego de Veículos

Tabela 3.1: Comparação entre os trabalhos relacionados.

Na Tabela 3.1 é apresentado um resumo dos trabalhos relacionados com suas respectivas propostas contendo: técnicas, modelos, as funções de ativação utilizadas e cenários onde foram aplicados. Os trabalhos relacionados de 1-5 são focados em comparar o desempenho do modelo ao se alterar a função de ativação para variados cenários, por exceção do número 2 que se utiliza de uma técnica de otimização, os outros 4 comparam o desempenho do modelo puro. Já o 6, 7 e 10, tem seus resultados obtidos após a utilização da técnica de otimização PSO, sendo o primeiro aplicado pro cenário de predição de carga de rede sem fio e os outros 2 para o cenário de tráfego de veículos. Ainda sim, o 10 pode ser relacionado ao 7 e 8, pois ambos são aplicados ao mesmo cenário, e principalmente ao 9, por este trabalho ser continuidade deste.

4 Abordagem Proposta

Considerando a técnica PSO como uma boa abordagem para otimizar a solução de um problema, é considerado a proposta do trabalho de Qolomany et al (2017). Porém, em nosso estudo de caso, além de aplicar em um cenário de rede sem fio, também é aplicado para um outro cenário, o de tráfego de veículos. O foco é a otimização de parâmetros que implicam na funcionalidade de uma rede neural, assim como também, analisar essa proposta em cima de dois modelos de redes neurais distintos.

4.1 Rede Neural

A seguir é definido como foi processada e estabelecida a entrada de dados para ambas as redes neurais, assim como a arquitetura de cada modelo adotado (*i.e.* MLP e GRU). As FAs que serão utilizadas para teste são: a Logística, ReLU e *Softplus*, que já foram descritas anteriormente.

4.1.1 Entrada de Dados

Para a definição do vetor de características que serão utilizadas para entrada de dados com intuito de alimentar a rede neural, é seguido o trabalho de Ferreira et al (2019), que também trabalhou com predição de séries temporais em um cenário de tráfego de veículos. Para Ferreira et al (2019), dois parâmetros Q e P foram chaves para conseguir pré-processar e preparar a entrada de treinamento para a rede neural.

1. Primeiramente o parâmetro Q , se refere a dados históricos, ou seja, dado um ponto, são considerados dados anteriores a este ponto, (*e.g.* no instante t , são considerados pontos nos instantes $t - 1$ até $t - j$, sendo $Q = j$).
2. Já para o parâmetro P , este tem como objetivo considerar dados recentes ao valor que se quer prever, de tal forma que, dado um ponto no instante t , são escolhidos os pontos nos instantes $t + 1$ até $t + k$, sendo $P = k$ e o dado no instante $t + k$, o

valor a ser utilizado como alvo da rede neural.

Na Figura 4.1 é representado o vetor de característica x_i para entrada da rede neural, onde temos a união das características que utilizam o parâmetro Q com as características do parâmetro P . Importante destacar que x_t^Q e x_t^P são equivalentes, logo, apenas um permanece neste vetor para alimentar a rede neural.

$$x_i = \left| x_t^Q \quad x_{t-1}^Q \quad x_{t-2}^Q \quad \dots \quad x_{t-m}^Q \quad \cup \quad x_t^P \quad x_{t+1}^P \quad x_{t+2}^P \quad \dots \quad x_{t+k}^P \right|.$$

Figura 4.1: Vetor de característica para entrada na rede neural.

Dentre os testes conduzidos em Ferreira et al (2019), obteve-se como melhor performance de rede neural os valores para P e Q iguais a 11 e 3, respectivamente. Logo, é fixado estes valores para o cenário de tráfego de veículos com intuito do foco ser direcionado apenas aos parâmetros referentes ao modelo. Desta mesma forma, foi realizado diversos testes variando estes parâmetros para o cenário de rede sem fio, e foi registrado que os parâmetros $P = 10$ e $Q = 2$ representam o melhor modelo até o momento. Vale ressaltar que o parâmetro Q para o cenário de tráfego de veículos se refere a pontos em semanas passadas, enquanto para o cenário de rede sem fio – a interpretação que gerou melhores resultados – foi de pontos em dias passados.

4.1.2 MLP

A Figura 4.2 mostra a arquitetura utilizada para o modelo MLP neste trabalho. Como a proposta é conseguir comparar o desempenho de cada FA aplicada ao modelo, e analisar seu ganho em comparação a rede neural recorrente, definimos a arquitetura com apenas uma camada oculta. Um exemplo deste modelo para o cenário de tráfego de veículos, é que a camada de entrada será alimentada com $14(P + Q)$ características, logo, pode ser visto para esta arquitetura com $I = 14$ neurônios nesta camada. Já na camada oculta, o número de neurônios N irá variar entre um intervalo que será definido mais a frente.

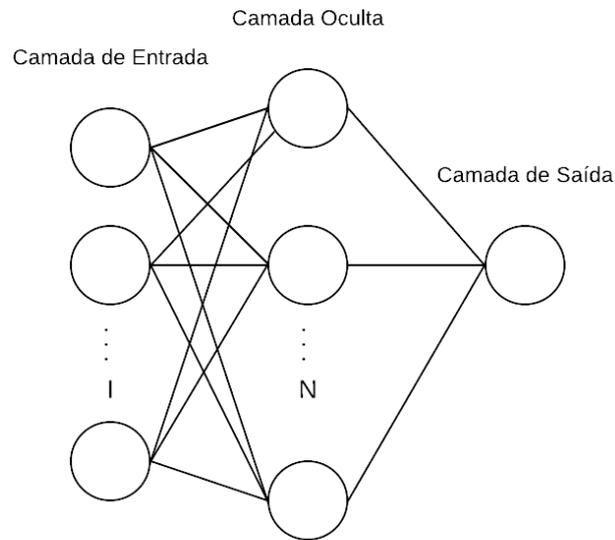


Figura 4.2: Arquitetura do modelo MLP utilizado.

4.1.3 GRU

Na Figura 4.3 é definida a arquitetura usada para o modelo GRU. Considerando o modelo MLP visto na Figura 4.2, é adicionado a camada recorrente antes da camada oculta, desta forma, é analisado o desempenho com a adição da GRU em relação ao MLP. Dentro da camada GRU existe N células que foram descritas na Figura 2.3.

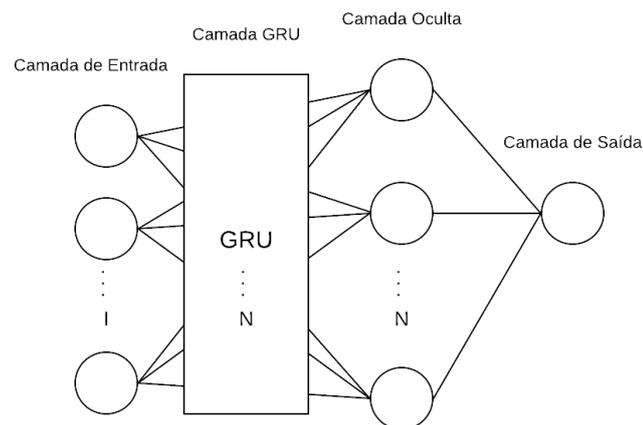


Figura 4.3: Arquitetura do modelo GRU utilizado.

4.2 PSO

Aplicar uma meta-heurística a um problema nem sempre é uma tarefa fácil, é necessário conhecer e definir muito bem o seu domínio de ação e suas funções para que a técnica funcione de forma esperada e encontre um bom resultado. Nesta seção apresentaremos mais detalhes sobre a parametrização escolhida para a meta-heurística PSO.

Como já mencionado, a técnica PSO age de forma populacional, fazendo com que as partículas da população tentem encontrar uma melhor solução a cada iteração. Porém, para encontrar essa melhor solução é necessário definir exatamente o que está tentando otimizar e quais variáveis implicam nessa otimização. Sendo assim, as variáveis a ser focadas, são:

- Número de Neurônios,
- Taxa de Aprendizagem.

Essas variáveis são tratadas diretamente como o posicionamento da partícula dentro da população e todo o processo de otimização é feito tendo essas como entrada para a rede neural. Entretanto, alguns desses parâmetros precisam ser restringidos a um intervalo, simplesmente por questões de tempo de processamento (*e.g.* como o espaço de busca e o próprio convergimento da rede neural, que já foi mencionado na seção anterior). Desta forma, foram realizados diversos testes diretamente com a rede neural para definirmos o domínio de cada parâmetro. Tais domínios podem ser vistos na Tabela 4.1, que é composta pelos limites que os parâmetros podem atingir.

Tabela 4.1: Intervalo dos parâmetros que serão otimizados, constituído por limite inferior e superior.

Posição Parâmetros	Intervalo	
	Posição Mínima	Posição Máxima
Número de Neurônios	60	140
Taxa de Aprendizagem	0.00001	0.00009

Diferentemente do que foi proposto em Qolomany et al (2017), é desconsiderado o parâmetro referente ao número de camadas e seguido com número de neurônios, incluindo taxa de aprendizagem. Outro ponto importante para ser destacado sobre a execução desta técnica é tratar a forma como as partículas se locomovem. E, para isso, é necessário

também definirmos uma velocidade mínima e máxima para cada um dos parâmetros citados na Tabela 4.1. Estes valores limites podem ser calculados através das Equações 4.1 e 4.2, que utilizam os valores limitantes predefinidos da posição para calcular seu limite de velocidade inferior e superior correspondente.

$$v_{min} = -0.1 * (x_{max} - x_{min}) \quad (4.1)$$

$$v_{max} = 0.1 * (x_{max} - x_{min}) \quad (4.2)$$

Deste modo, são definidas a velocidade mínima e máxima da partícula como sendo 10% do valor da diferença entre suas posições limites.

Esta abordagem segue a mesma ideia proposta anteriormente na literatura, porém com algumas adaptações para este problema, com isso pode ser expresso para esta adaptação seguindo o Algoritmo 1. O algoritmo pode ser visto como duas partes: inicialização e iteração. A parte de inicialização corresponde em, pra cada partícula da população, ser gerado uma posição e velocidade inicial. Após isso, treinar a rede neural com essa posição gerada, em seguida, calcular a função objetivo correspondente a essa rede neural e, por fim, guardar as informações sobre esta partícula, além de selecionar o *pBest* desta e o *gBest* da população. Seguindo para a segunda parte, é onde o maior esforço da técnica ocorre, dentro de um certo número de iterações, será passado por toda população calculando uma nova velocidade e posição, e em decorrência disto, será recalculada a função objetivo gerada pelo treino de uma nova rede neural. Além disto, neste segundo passo será avaliado se a função objetivo é melhor que o *pBest* e o *gBest*, caso seja em alguns desses casos, as informações sobre esta partícula são guardadas como solução. Ao fim deste algoritmo, é retornado os valores do *gBest* como melhor solução encontrada por aquela população.

4.2.1 Função Objetivo

Nesta subseção, será abordado as métricas que foram usadas como função objetivo para avaliar o desempenho dos modelos implementados. Dentre as diversas métricas existentes

Algoritmo 1: PSO

```

Entrada: númeroIterações, populaçãoTamanho
Saída : númeroNeurônios, taxaAprendizagem
particulaPop ← lista de partículas;
for i ← 0 até populaçãoTamanho do
    criar uma partícula p;
    inicializar p com parâmetros aleatórios dentro de seus intervalos
    correspondentes;
    p.pBest ← função objetivo calculada após a rede neural ser computada
    pela partícula p;
    if p.pBest < gBest.pBest then
        | gBest ← p;
    end if
    Inserir partícula p na particulaPop;
end for
while numeroIteracoes ≠ 0 do
    for cada partícula p em particulaPop do
        | p ← calcular nova velocidade;
        | p ← calcular nova posição;
        | erro ← função objetivo calculada após rodar os novos parâmetros
        | para a rede neural pela partícula p;
        | if erro < p.pBest then
        | | p.pBest ← erro
        | | p.positionBest ← p.position
        | | if erro < gBest.pBest then
        | | | gBest ← p;
        | | end if
        | end if
    end for
    numeroIteracoes ← numeroIteracoes − 1;
end while

```

para avaliar modelos de predição, escolhemos o *Mean Absolute Percentage Error* (MAPE) para o cenário de tráfego de veículos e Coeficiente de Determinação (R^2) para o cenário de rede sem fio.

MAPE

Provê uma interpretação intuitiva de erro relativo e é normalmente utilizado em problemas de regressão (Myttenaere et al, 2016). De acordo com a equação 4.3, n é o número de amostras para teste, \hat{y}_t é equivalente ao valor predito e y_t referente ao valor real no instante de tempo t . Como é uma métrica relacionada ao erro do modelo, quanto menor este valor,

melhor o modelo.

$$MAPE = \frac{1}{n} \sum_{t=1}^n \left| \frac{y_t - \hat{y}_t}{y_t} \right| \quad (4.3)$$

Coefficiente de Determinação - R^2

Indica o quanto o modelo consegue explicar os valores observados, variando entre 0 e 1, sendo que valores mais próximos de 1 indicam um modelo mais explicativo de acordo com as amostras utilizadas (Cameron et al, 1997). Seguindo a Equação 4.4, onde y_t é referente ao valor real, \hat{y}_t é o valor predito pelo modelo e \bar{y}_t é a média dos valores reais, no instante t .

$$R^2 = 1 - \frac{\sum_{t=1}^n (y_t - \hat{y}_t)^2}{\sum_{t=1}^n (y_t - \bar{y}_t)^2} \quad (4.4)$$

5 Avaliação e Resultados

Neste capítulo são apresentados os dois cenários, além de uma breve descrição das bases de dados utilizadas para os testes dos modelos. Em seguida, tem-se uma seção para apresentar todos os resultados obtidos para os dois modelos de redes neurais. Cada modelo de rede neural utiliza todas as FAs escolhidas (uma por vez) para serem analisadas neste trabalho. A análise das FAs foi composta, principalmente, do desempenho sob os modelos para ambos os cenários. Sendo este desempenho composto por uma análise feita em cima das métricas MAPE e R^2 para validar a acurácia dos modelos, e ao fim de cada cenário uma análise feita em relação ao tempo de execução.

Os testes apresentados neste capítulo foram executados em um computador com a seguinte configuração: Intel Core i5-7400 CPU @ 3.00GHz, 8GB de RAM e no sistema operacional Windows 10. Para a implementação das redes neurais e da técnica empregada neste trabalho, foi usado a linguagem de programação Python 3.6². Ainda sobre os modelos, foi utilizada a biblioteca de redes neurais Keras³ que possui código aberto e de forma simples, pode rodar em cima de bibliotecas como TensorFlow⁴, CNTK⁵ ou Theano⁶ para a criação e manipulação dos modelos - neste caso, foi escolhido a TensorFlow. Algumas outras bibliotecas como scikit-learn⁷ foram usadas para facilitar no pré-processamento e avaliação dos modelos.

Os *scripts* utilizados para implementar os algoritmos deste trabalho podem ser encontrados no *Github*⁸.

²<https://www.python.org/>

³<https://keras.io/>

⁴<https://github.com/tensorflow/tensorflow>

⁵<https://github.com/Microsoft/cntk>

⁶<https://github.com/Theano/Theano>

⁷<https://scikit-learn.org/stable/>

⁸<https://github.com/LucasFrank/TCC-code>

5.1 Base de Dados

Dentre os cenários que foram aplicados, tem o cenário de tráfego de veículos e o de rede sem fio. Sendo que para o primeiro cenário foi usada a base de dados *Caltrans Performance Measurement System* (PeMS) e para o segundo, foi utilizado o conjunto de dados fornecidos pela Universidade Federal de Juiz de Fora (UFJF). Portanto, ambas as base são de dados reais.

5.1.1 PeMS

O PeMS⁹ é um conjunto de dados bem conhecido para problemas relacionados ao tráfego de veículos (Chen et al, 2001). Através de uma rede de sensores, coleta-se os dados de tráfego e os fornece de forma aberta em seu repositório. Esses sensores estão localizados por diversos pontos pelo estado da Califórnia no Estados Unidos, e coletam várias informações referentes ao tráfego, como: quantidade de veículos, velocidade, etc.

A base que é utilizada é composta por 12 meses de dados coletados do ano de 2011, onde tem-se uma coluna para a **data e hora**, e uma outra coluna referente à **quantidade de veículos** que passaram naquele instante. Os dados são agregados em intervalos de 1 em 1 hora para contabilizar o número de veículos que o sensor capturou. Na Figura 5.1 é mostrado uma pequena parte do conjunto de dados PeMS composto pela coluna *date*, contendo **data e hora** e uma outro coluna *count* que fica por conta da quantidade média de veículos durante o intervalo de 1 hora. Essa quantidade está relacionada por média por se tratar de um cruzamento entre rodovias.

5.1.2 UFJF

Por meio da iniciativa de Campus Inteligente da UFJF¹⁰ e sob a responsabilidade de resguardar a privacidade dos usuários, a base de dados para rede sem fio foi fornecida pela UFJF. Esta base foi extraída a partir de registros (*logs*) de controladoras *Extreme*¹¹, que têm domínio sobre as informações geradas pelos pontos de acessos (AP) distribuídos por

⁹<http://pems.dot.ca.gov>

¹⁰<https://www.campusinteligente.ufjf.br/>

¹¹<https://www.extremenetworks.com/product/wireless-controllers/>

date	count
01/01/2011 0:00	647346.8
01/01/2011 1:00	882870.6
01/01/2011 2:00	696550.6
01/01/2011 3:00	483028.9
01/01/2011 4:00	375405.3
01/01/2011 5:00	401899
01/01/2011 6:00	527753
01/01/2011 7:00	591253.4
01/01/2011 8:00	751540.9
01/01/2011 9:00	996939.1
01/01/2011 10:00	1257571.9
01/01/2011 11:00	1553467.2
01/01/2011 12:00	1770240
01/01/2011 13:00	1916434.2
01/01/2011 14:00	1974171.1
01/01/2011 15:00	1917465.2

Figura 5.1: Conjunto de dados do PeMS.

todo o campus. A princípio, foram pré-processados vários *logs*, onde foi feita a extração com os seguintes dados: `data/hora`, `local de acesso`, e `id do usuário`. Pode ser visualizado um exemplo deste conjunto de dados da UFJF na Figura 5.2 com a coluna de `id do usuário` oculta.

Esta base é constituída de 3 meses (Agosto, Setembro e Outubro) do ano de 2018. Mas diferentemente do cenário de tráfego de veículos, são agregados os dados em intervalos de 5 em 5 minutos, já que se quer prever a carga em um intervalo de tempo menor. Sendo assim, é possível filtrar esta base de dados por localidade, e deste modo, criar um modelo de predição com intuito de prever a quantidade de carga por AP para este local.

Na Figura 5.3, tem o mapa da UFJF com os locais selecionados para teste realçados. Em vermelho, o Instituto de Artes e Design (IAD), já em amarelo tem os prédios da Engenharia e, por último, de azul, o Instituto de Ciências Exatas (ICE).

5.2 Resultados

A seguir, são apresentados os resultados alcançados para cada um dos cenários que foram avaliados neste trabalho. Além disso, uma comparação entre os desempenhos das FAs em ambos os modelos após a execução da técnica PSO.

2018 Aug 1 00:00:21	ICE
2018 Aug 1 00:05:32	ICE
2018 Aug 1 00:07:38	2andarSala4258
2018 Aug 1 00:07:43	ENG
2018 Aug 1 00:11:13	DeptConstrucaoCivilSala4107
2018 Aug 1 00:14:06	ICE
2018 Aug 1 00:16:30	ICE
2018 Aug 1 00:20:15	ap4600-67AE5C
2018 Aug 1 00:25:27	ap4600-67AEF4
2018 Aug 1 00:27:38	SegundoAndar_Corredor
2018 Aug 1 00:29:55	RU
2018 Aug 1 00:30:59	MED
2018 Aug 1 00:31:20	RU
2018 Aug 1 00:31:28	RU
2018 Aug 1 00:31:50	RU
2018 Aug 1 00:32:22	RU

Figura 5.2: Conjunto de dados da UFJF.

Para demonstrar o desempenho dos modelos, gráficos que apresentam comparação entre valores preditos e valores reais serão mostrados. Porém, nesta seção, apenas os melhores resultados de cada modelo em ambos os cenários serão exibidos, o restante ficará disponível para ser visualizado no fim da monografia. E para analisar esta performance, é fixado o número de épocas em 100, por razões de tempo de execução, e utilizando uma funcionalidade do *Keras* para finalizar o treinamento, caso a rede neural não melhore os resultados por 2 épocas consecutivas. Deste modo, pode ser acompanhado como o modelo se adaptou à FA, e se poderia continuar o treinamento.

É válido mencionar que para os testes feitos, foram separados 2/3 da base de dados para treino, e 1/3 para teste. Desta forma temos uma relação de 8 meses para treino e 4 para testes no cenário de tráfego de veículos e 2 meses para treino e 1 para teste no cenário de rede sem fio.

Para a técnica PSO, são definidos o número de iterações e tamanho da população como sendo igual a 20, pois foi a quantidade mínima que apresentou resultados mais satisfatórios após uma série de testes. Assim, teremos ao total 400 resultados a serem analisados por modelo com cada FA. Para as variáveis w , c_1 e c_2 encontradas na Equação 2.8, são definidas sendo, 0.7 para a primeira após testes iniciais e o valor 2 para as últimas duas, de acordo com (Alam et al, 2015).

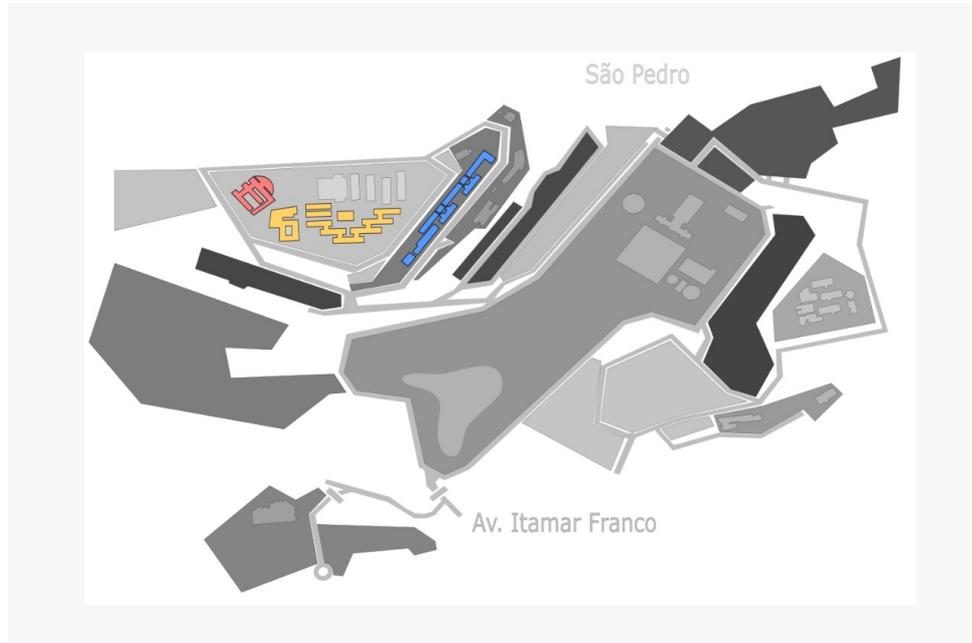


Figura 5.3: Mapa da UFJF com os locais selecionados para o cenário de rede sem fio.

5.2.1 PeMS

Os resultados apresentados na Tabela 5.1 correspondem aos testes feitos no cenário de tráfego de veículos, onde foi aplicado o PSO para cada modelo variando a FA a cada execução. Avaliando este cenário com a métrica MAPE, nota-se que a função logística teve melhores resultados entre as FAs utilizadas, ganhando com melhor desempenho no caso do modelo GRU com MAPE de 0.044 (*i.e.* 4.4% em porcentagem), número de neurônios $N = 113$ e taxa de aprendizagem = 0.000052.

Tabela 5.1: Resumo dos melhores resultados obtidos para cada função de ativação usada em cada modelo para o cenário de tráfego de veículos. Melhor resultado de cada modelo destacado em negrito.

Modelo	Função de Ativação	Resultados		
		N	Taxa de Aprendizagem	MAPE
MLP	Logística	104	0.000085	0.048
MLP	ReLU	124	0.000073	0.068
MLP	<i>Softplus</i>	118	0.000049	0.070
GRU	Logística	113	0.000052	0.044
GRU	ReLU	116	0.000055	0.058
GRU	<i>Softplus</i>	75	0.000061	0.069

Fazendo uma análise do desempenho dos modelos comparando as FAs correspondentes, pode ser visto que o modelo GRU apresentou um MAPE menor para os 3

casos. Mostrando que para este cenário, a adição da camada GRU teve efeito positivo nos resultados.

Nas Figuras 5.4 e 5.5 temos os gráficos que mostram sobre o desempenho do melhor resultado para o modelo MLP e GRU, respectivamente. Os gráficos podem ser visto como sendo uma comparação entre o valor real separado para teste (linha azul) e o valor predito pelo modelo (linha laranja). Para melhor visualização, foram selecionados 100 pontos aleatórios. Vale ressaltar que está sendo predito quantidade de veículos, porém o gráfico varia entre 0 e 1, pois foi feita uma normalização dos valores para facilitar na avaliação. A partir dos valores preditos acima de 50% usados no gráfico, é calculado uma acurácia média de 96.04% para o MLP e 96.69% para o GRU.

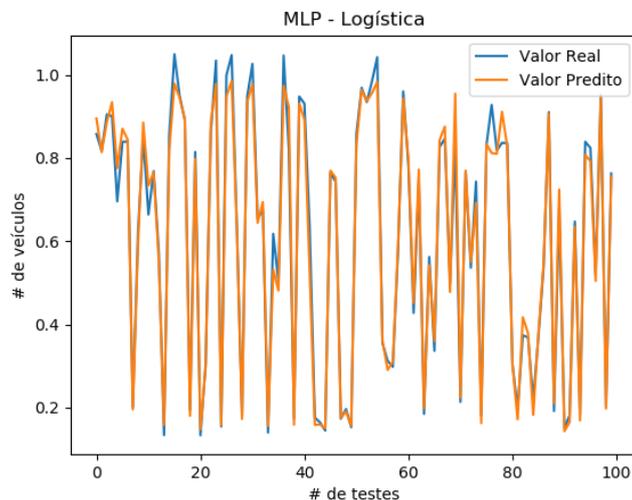


Figura 5.4: Comparando valor predito com valor real para modelo MLP com função logística. Linha azul corresponde ao valor real e linha laranja ao valor predito. Cenário: Tráfego de Veículos.

Neste cenário ainda pode ser feito uma análise e comparação do número de épocas médio que cada modelo precisou até alcançar sua convergência final. Como pode ser visto na Figura 5.6, no geral, os modelos MLPs levaram mais épocas para treinar que os GRUs, entretanto, o intervalo de confiança dos modelos GRUs são bem maiores. Comparando este mesmo gráfico com a Tabela 5.2, o MLP utilizando mais épocas, ainda tem seu processamento bem mais rápido que o GRU. E, considerando que a ReLU é uma FA de rápido processamento, seu impacto dentro do modelo resultou em um treinamento bem lento para ambos os modelos, com destaque para GRU com ReLU, que teve um tempo

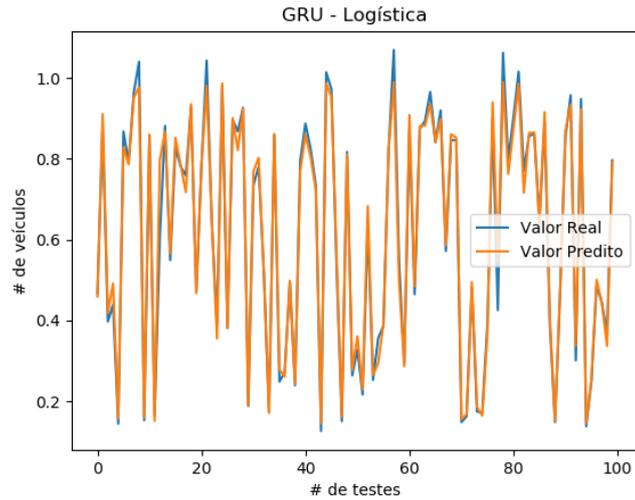


Figura 5.5: Comparando valor predito com valor real para modelo GRU com função logística. Linha azul corresponde ao valor real e linha laranja ao valor predito. Cenário: Tráfego de Veículos.

bem maior devido a rede neural recorrente já ter um processamento mais lento que a *feedforward*, e também por esta combinação ter levado mais tempo para treinar quando verificado o número médio de épocas e seu intervalo de confiança.

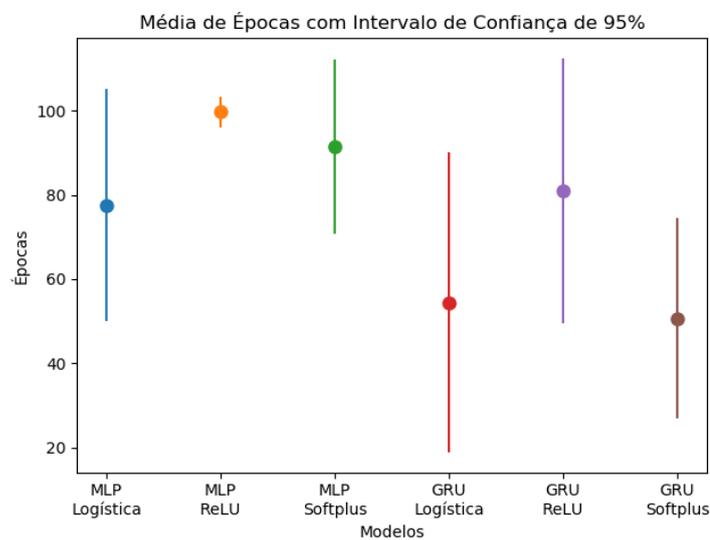


Figura 5.6: Comparando o desempenho dos modelos em relação ao número de épocas médio até sua convergência. Cenário: Tráfego de Veículos.

Tabela 5.2: Resumo do tempo de execução médio e total pra cada combinação dos modelos com as FAs. Cenário: Tráfego de Veículos. Melhor resultado de cada modelo destacado em negrito.

		Resultados - Tempo de Execução (s)	
Modelo	Função de Ativação	Tempo Médio por Partícula	Tempo Total
MLP	Logística	35	13919
MLP	ReLU	44.1	17630
MLP	<i>Softplus</i>	43.0	17188
GRU	Logística	122.6	49046
GRU	ReLU	187.0	75194.5
GRU	<i>Softplus</i>	121.1	48433

Discussão

Para o cenário de tráfego de veículos, os trabalhos em Ferreira et al (2019); Frank et al (2019) podem ser comparados, onde em ambos os casos utilizaram uma arquitetura de rede neural MLP com três camadas ocultas para o cenário do PeMS. Os modelos MLP e GRU aplicados neste trabalho ao serem comparados aos resultados de Ferreira et al (2019), superam os resultados atingidos em aproximadamente 1%. Quando comparado com Frank et al (2019), considerando que a estrutura dos modelos difere de 3 para 1 camada oculta deste trabalho, têm-se um resultado bem próximo, mas não superando.

5.2.2 UFJF

Já para o cenário da UFJF, temos o intuito de gerar modelos por localidade, então para isto, a base de dados foi separada nos 3 lugares descritos anteriormente para os testes, sendo estes: Engenharia, ICE e IAD. Logo, teremos resultados para cada um destes locais.

Engenharia

Levando em consideração os diversos APs espalhados pela engenharia, tivemos em média 36 conexões a cada 5 minutos. Na Tabela 5.3 podem ser vistos os resultados obtidos para cada modelo. Porém, diferente do cenário de tráfego de veículos, iremos avaliar o desempenho dos modelos utilizando a métrica R^2 . Como queremos que o modelo se adapte ao perfil do local, quanto mais próximo o valor de R^2 ficar de 1, melhor será. Destaca-se a utilização da métrica R^2 pelo fato de que a métrica MAPE não se adapta muito bem quando se trata de valores pequenos, principalmente quando o valor real pode

ser 0, (*i.e.* gerando erro de divisão por zero).

Tabela 5.3: Resumo dos melhores resultados obtidos para cada função de ativação usada em cada modelo para o cenário de rede sem fio. Local: Engenharia. Melhor resultado de cada modelo destacado em negrito.

		Resultados		
Modelo	Função de Ativação	N	Taxa de Aprendizagem	R^2
MLP	Logística	96	0.000044	0.904
MLP	ReLU	112	0.000065	0.895
MLP	<i>Softplus</i>	68	0.000089	0.872
GRU	Logística	107	0.000057	0.908
GRU	ReLU	127	0.000080	0.902
GRU	<i>Softplus</i>	117	0.000078	0.885

Deste modo, novamente temos a função logística ganhando em ambos os modelos com melhor desempenho, ficando com R^2 de 0.908, $N = 107$ e taxa de aprendizagem = 0.000057. Porém, desta vez, a função ReLU em conjunto com o modelo GRU ficou bem próximo do melhor resultado com $R^2 = 0.902$.

Na Figura 5.7 é exibido o gráfico comparativo de valor predito e valor real para o modelo MLP usando a função logística, que obteve $R^2 = 0.904$. Já na Figura 5.8, tem este mesmo comparativo, mas para o modelo GRU com a função logística, que para este local, é o melhor resultado obtido. Em ambos os casos, considerando os valores preditos acima de 80%, é alcançado uma acurácia média de 86% para o primeiro modelo, e 95% para o segundo.

A partir da Tabela 5.4 é resumido o tempo de execução de cada teste feito para a Engenharia, e em comparação ao cenário anterior, tem um tempo mais elevado, devido a base de dados ser maior em quantidade de pontos. Novamente a função logística se mostrou ser a mais rápida para os dois modelos, enquanto a ReLU a mais lenta.

Tabela 5.4: Resumo do tempo de execução médio e total pra cada combinação dos modelos com as FAs. Cenário: Rede sem fio. Local: Engenharia.

		Resultados - Tempo de Execução (s)	
Modelo	Função de Ativação	Tempo Médio por Partícula	Tempo Total
MLP	Logística	79	31694.6
MLP	ReLU	114.62	45850.4
MLP	<i>Softplus</i>	133.65	53463.7
GRU	Logística	233.5	93399.1
GRU	ReLU	376.2	150499.1
GRU	<i>Softplus</i>	691.4	276551.5

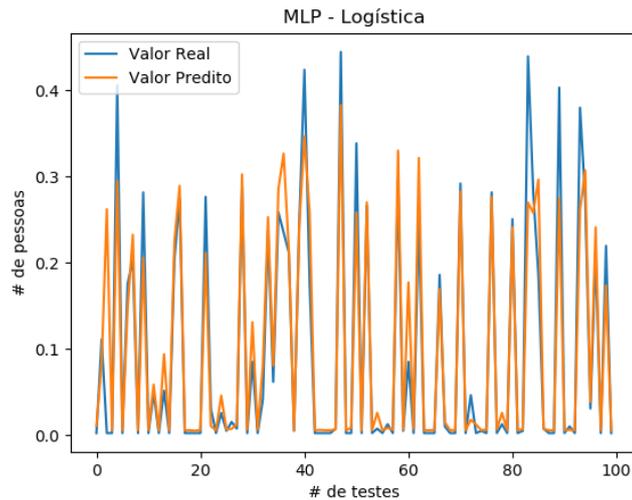


Figura 5.7: Comparando valor predito com valor real para modelo MLP com função logística. Linha azul corresponde ao valor real e linha laranja ao valor predito. Cenário: Rede sem fio. Local: Engenharia.

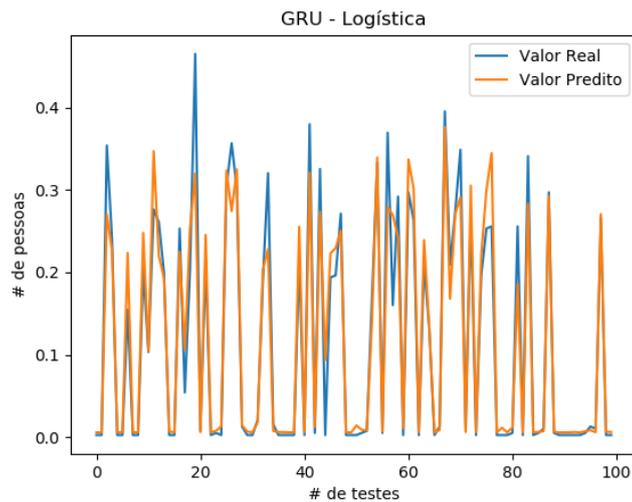


Figura 5.8: Comparando valor predito com valor real para modelo GRU com função logística. Linha azul corresponde ao valor real e linha laranja ao valor predito. Cenário: Rede sem fio. Local: Engenharia.

ICE

Desta vez, para os testes feitos nos APs do ICE, que possui em média 37 conexões a cada 5 minutos, os resultados com R^2 são um pouco maior que o local anterior. A Tabela 5.5 resume os resultados obtidos com os parâmetros escolhidos. O melhor modelo MLP atingiu um R^2 de 0.913 com a função Logística, enquanto o melhor modelo GRU alcançou 0.916, também com a mesma função, sendo este último no geral, o melhor modelo treinado

para este local dentro dos resultados encontrados.

Considerando os resultados apresentados, as combinações de MLP e GRU com as funções Logística e ReLU, tiveram resultados muito próximos, sendo diferenciado apenas na terceira casa decimal. Diferente dos testes anteriores, onde numa classificação geral, a Logística ocupava as 2 primeiras colocações, desta vez, a combinação GRU com ReLU ficando como segundo melhor modelo.

Tabela 5.5: Resumo dos melhores resultados obtidos para cada função de ativação usada em cada modelo para o cenário de rede sem fio. Local: ICE. Melhor resultado de cada modelo destacado em negrito.

		Resultados		
Modelo	Função de Ativação	N	Taxa de Aprendizagem	R^2
MLP	Logística	105	0.00004	0.913
MLP	ReLU	110	0.000075	0.912
MLP	<i>Softplus</i>	92	0.000081	0.880
GRU	Logística	92	0.000069	0.916
GRU	ReLU	112	0.000061	0.915
GRU	<i>Softplus</i>	89	0.000086	0.887

Em ambas as Figuras 5.9 e 5.10, podem ser vistos os desempenhos dos modelos MLP e GRU, respectivamente, utilizando a função Logística, que foi a que produziu melhor resultado pra cada modelo. Considerando o mesmo limiar de 80% dos valores preditos, foi calculado uma acurácia média para valores acima deste, de 92% para o MLP, e 89% para o GRU.

Para o tempo de execução no ICE, a Tabela 5.6 mostra que apesar do tempo ter aumentado mais um pouco, o que foi concluído para o cenário da Engenharia permanece igual para este.

Tabela 5.6: Resumo do tempo de execução médio e total pra cada combinação dos modelos com as FAs. Cenário: Rede sem fio. Local: ICE.

		Resultados - Tempo de Execução (s)	
Modelo	Função de Ativação	Tempo Médio por Partícula	Tempo Total
MLP	Logística	118.6	47426.8
MLP	ReLU	134.1	53652.5
MLP	<i>Softplus</i>	133.2	52281.3
GRU	Logística	233.7	93489.2
GRU	ReLU	524.9	209958.6
GRU	<i>Softplus</i>	479.5	191781.6

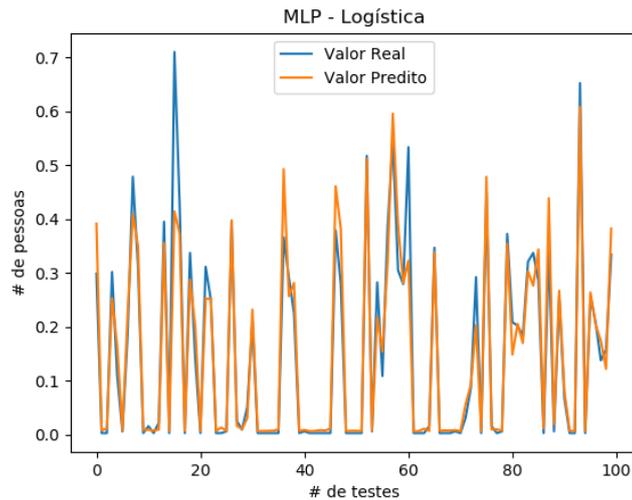


Figura 5.9: Comparando valor predito com valor real para modelo MLP com função logística. Linha azul corresponde ao valor real e linha laranja ao valor predito. Cenário: Rede sem fio. Local: ICE

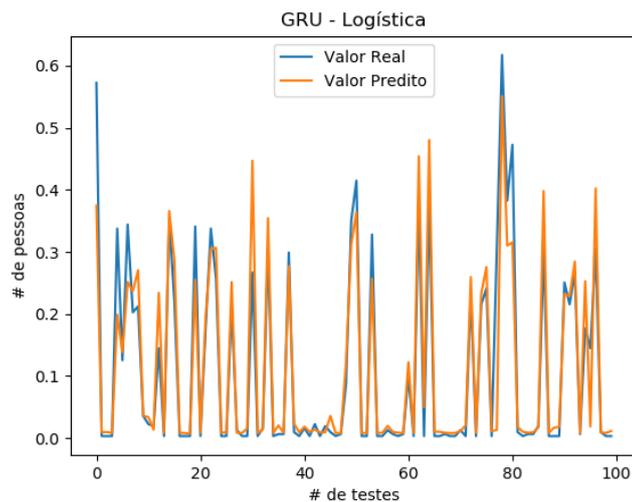


Figura 5.10: Comparando valor predito com valor real para modelo GRU com função logística. Linha azul corresponde ao valor real e linha laranja ao valor predito. Cenário: Rede sem fio. Local: ICE

IAD

Para os testes efetuados nos AP's próximo ao IAD, destaca-se uma diferença, que é a quantidade de carga que ocorre neste local comparado ao ICE e Engenharia, em média, ocorre 11 conexões a cada 5 minutos.

Analisando pela Tabela 5.7, nota-se que o melhor resultado atingido para este local, foi com o modelo MLP utilizando a função Logística atingindo um R^2 de 0.85,

diferente dos outros cenários onde a GRU levou maior destaque. E analisando o melhor resultado do modelo GRU, pode ser visto com $R^2 = 0.84$ alcançado com a função ReLU. Pode ser apontado também, o quão próximo o resultado da função *Softplus* ficou da ReLU com uma diferença de apenas 0.005.

Tabela 5.7: Resumo dos melhores resultados obtidos para cada função de ativação usada em cada modelo para o cenário de rede sem fio. Local: IAD. Melhor resultado de cada modelo destacado em negrito.

Modelo	Função de Ativação	Resultados		
		N	Taxa de Aprendizagem	R^2
MLP	Logística	124	0.000068	0.856
MLP	ReLU	126	0.000069	0.815
MLP	<i>Softplus</i>	60	0.000083	0.740
GRU	Logística	79	0.000086	0.806
GRU	ReLU	89	0.000062	0.841
GRU	<i>Softplus</i>	112	0.000085	0.836

Seguindo o mesmo propósito dos testes feitos nos locais anteriores, um gráfico comparando o desempenho dos valores preditos e reais foi construído sobre os modelos treinados. Nas Figuras 5.11 e 5.12 podem ser visualizados a performance dos dois melhores modelos, MLP e GRU respectivamente. O primeiro modelo atingiu uma acurácia média de 76%, enquanto o segundo atingiu 87% considerando valores preditos acima de 80% para estes gráficos.

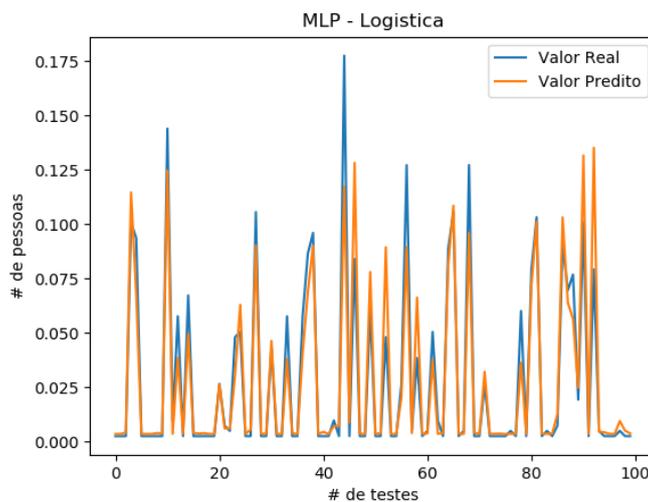


Figura 5.11: Comparando valor predito com valor real para modelo MLP com função logística. Linha azul corresponde ao valor real e linha laranja ao valor predito. Cenário: Rede sem fio. Local: IAD

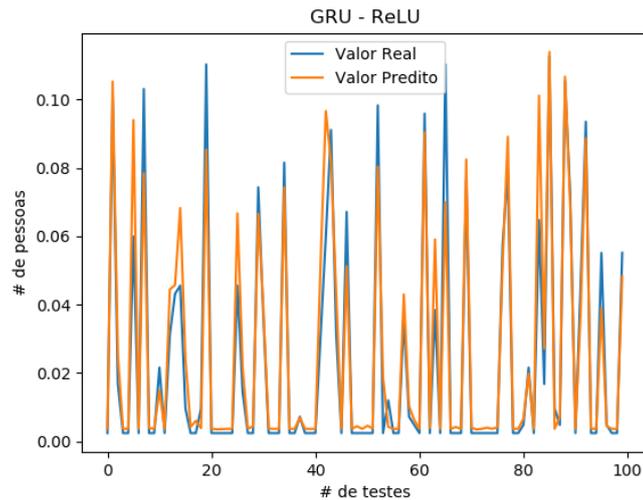


Figura 5.12: Comparando valor predito com valor real para modelo GRU com função ReLU. Linha azul corresponde ao valor real e linha laranja ao valor predito. Cenário: Rede sem fio. Local: IAD

Finalmente, analisando o tempo de execução para este local, temos a Tabela 5.8 para auxiliar. Levando em conta que MLP com Logística foi a que atingiu melhor desempenho de acurácia, e menor tempo de execução entre todos, fica como melhor modelo para este local considerando estas duas formas de avaliação.

Tabela 5.8: Resumo do tempo de execução médio e total pra cada combinação dos modelos com as FAs. Cenário: Rede sem fio. Local: IAD.

Modelo	Função de Ativação	Resultados - Tempo de Execução (s)	
		Tempo Médio por Partícula	Tempo Total
MLP	Logística	152.4	60951.45
MLP	ReLU	219.3	87710.80
MLP	<i>Softplus</i>	201.8	80711.2
GRU	Logística	483.8	193517.4
GRU	ReLU	468.54	187415.1
GRU	<i>Softplus</i>	490.5	196188.2

Discussão

Para o cenário de rede sem fio da UFJF, destaca-se que para os locais da Engenharia e ICE onde tem maior quantidade de usuários (mais de 30 conexões a cada 5 minutos na média), os modelos MLP e GRU se adaptaram melhor com a FA Logística, com ênfase para o GRU que foi o melhor modelo. No local do IAD, tendo um perfil com média de menos de 15 conexões a cada 5 minutos, o modelo que se melhor destaca é o MLP com

a FA Logística, e dos modelos GRU, a função ReLU foi a que melhor se adaptou tendo uma acurácia média maior por conseguir prever valores altos de carga.

6 Conclusão e Trabalhos Futuros

Este trabalho teve como propósito utilizar a técnica PSO para otimizar os parâmetros - número de neurônios e taxa de aprendizagem - sobre dois modelos de redes neurais distintos (MLP e GRU). Os modelos foram avaliados ao comparar o desempenho obtido variando entre as três funções de ativação selecionadas (Logística, ReLU e *Softplus*) para cada modelo em dois cenários diferentes - fluxo de tráfego de veículos e carga de rede sem fio. Tais cenários possuem a idealização de alcançar um ambiente inteligente, podendo aplicar modelos preditivos a fim de auxiliar o usuário com uma melhor saúde do serviço ou com tomadas de decisão, por exemplo.

Os resultados obtidos mostram que os modelos treinados se adaptaram satisfatoriamente bem ao perfil dos cenários, podendo concluir que a utilização da técnica PSO teve efeito positivo neste trabalho, como foi eficaz em outros (Qolomany et al, 2017; Hu et al, 2014; Frank et al, 2019). Sobre os modelos, tivemos na maioria dos testes uma melhora dos resultados com o modelo GRU, entretanto, o tempo de execução chega a ser de 3 a 4 vezes maior que o MLP. Em relação as FAs, tivemos na maior parte, a superioridade da função Logística, sendo para precisão do modelo, e até mesmo para o tempo de execução.

Com os modelos otimizados e treinados, agora podem ser aplicados aos cenários respectivos com intuito de prover uma predição mais precisa. Com isso, predição no cenário de fluxo de tráfego de veículos auxilia o esquema de cidades inteligentes, ajudando por exemplo, identificar horários de pico quando tem engarrafamentos. Assim como também, para o cenário de carga em rede sem fio, podemos citar a aplicação para dimensionar melhor a rede criando rotas alternativas.

Como trabalhos futuros pretende-se incluir outras FAs para avaliação, assim como expandir os parâmetros que estão sendo otimizados pela técnica PSO. Um exemplo disso, seria adicionar o número de camadas, como realizado em (Qolomany et al, 2017), aplicando aprendizagem profunda de forma eficaz para os cenários utilizados. Há também outros parâmetros interessantes a serem analisados, como o número de épocas e o tamanho do lote, que além de agir na precisão e convergência do modelo, influencia no tempo

de treinamento. Outro ponto importante a se explorar seria implementação de outras meta-heurísticas, e aplicação destas técnicas para novos ambientes inteligentes. Por fim, algo a ser considerado é a modificação da forma de pré-processar as bases de dados, seja separando-a em grupos que possuem características similares ou a inclusão das variáveis P e Q , que constituem o processamento principal da entrada de dados dos modelos, como parâmetros a serem otimizados pela meta-heurística.

Bibliografia

- Abuarqoub, A.; Abusaimh, H.; Hammoudeh, M.; Uliyan, D.; Abu-Hashem, M. A.; Murad, S.; Al-Jarrah, M. ; Al-Fayez, F. **A survey on internet of thing enabled smart campus applications**. In: Proceedings of the International Conference on Future Networks and Distributed Systems, p. 38. ACM, 2017.
- Agarap, A. F. Deep learning using rectified linear units (relu). **arXiv preprint arXiv:1803.08375**, 2018.
- Agostinelli, F.; Hoffman, M.; Sadowski, P. ; Baldi, P. Learning activation functions to improve deep neural networks. **arXiv preprint arXiv:1412.6830**, 2014.
- Alam, M. N.; Das, B. ; Pant, V. A comparative study of metaheuristic optimization approaches for directional overcurrent relays coordination. **Electric Power Systems Research**, v.128, p. 39–52, 2015.
- Baraha, S.; Biswal, P. K. **Implementation of activation functions for elm based classifiers**. In: Wireless Communications, Signal Processing and Networking (WiSP-NET), 2017 International Conference on, p. 1038–1042. IEEE, 2017.
- Bircanoğlu, C.; Arıca, N. **A comparison of activation functions in artificial neural networks**. In: 2018 26th Signal Processing and Communications Applications Conference (SIU), p. 1–4. IEEE, 2018.
- Cameron, A. C.; Windmeijer, F. A. An r-squared measure of goodness of fit for some common nonlinear regression models. **Journal of econometrics**, v.77, n.2, p. 329–342, 1997.
- Chen, C.; Petty, K.; Skabardonis, A.; Varaiya, P. ; Jia, Z. Freeway performance measurement system: mining loop detector data. **Transportation Research Record**, v.1748, n.1, p. 96–102, 2001.
- Cho, K.; Van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H. ; Bengio, Y. Learning phrase representations using rnn encoder-decoder for statistical machine translation. **arXiv preprint arXiv:1406.1078**, 2014.
- Cybenko, G. Approximation by superpositions of a sigmoidal function. **Mathematics of control, signals and systems**, v.2, n.4, p. 303–314, 1989.
- Nair, V.; Hinton, G. E. **Rectified linear units improve restricted boltzmann machines**. In: Proceedings of the 27th international conference on machine learning (ICML-10), p. 807–814, 2010.
- Dalal, K.; Dahiya, P. State-of-the-art in vanets: The core of intelligent transportation system. **IUP Journal of Electrical and Electronics Engineering**, v.10, n.1, p. 27, 2017.
- Dlugosz, Z.; Dlugosz, R. **Nonlinear activation functions for artificial neural networks realized in hardware**. In: 2018 25th International Conference "Mixed Design of Integrated Circuits and System" (MIXDES), p. 381–384. IEEE, 2018.

- Dugas, C.; Bengio, Y.; Bélisle, F.; Nadeau, C. ; Garcia, R. **Incorporating second-order functional knowledge for better option pricing**. In: Advances in neural information processing systems, p. 472–478, 2001.
- Eberhart, R.; Kennedy, J. **Particle swarm optimization**. In: Proceedings of the IEEE international conference on neural networks, volume 4, p. 1942–1948. Citeseer, 1995.
- Eberhart, R. C.; Shi, Y. **Comparison between genetic algorithms and particle swarm optimization**. In: International conference on evolutionary programming, p. 611–616. Springer, 1998.
- Ferreira, Y. M.; Frank, L. R.; Julio, E. P.; Ferreira, F. H. C.; Dembogurski, B. J. ; Silva, E. F. **Applying a multilayer perceptron for traffic flow prediction to empower a smart ecosystem**. In: International Conference on Computational Science and Its Applications, p. 633–648. Springer, 2019.
- Fleck, L.; Tavares, M. H. F.; Eyng, E.; Helmann, A. C. ; de Moares Andrade, M. A. Redes neurais artificiais: Princípios básicos. **Revista Eletrônica Científica Inovação e Tecnologia**, v.7, n.13, p. 47–57, 2016.
- Frank, L. R.; Ferreira, Y. M.; Julio, E. P.; Ferreira, F. H. C.; Dembogurski, B. J. ; Silva, E. F. **Multilayer perceptron and particle swarm optimization applied to traffic flow prediction on smart cities**. In: International Conference on Computational Science and Its Applications, p. 35–47. Springer, 2019.
- Furtado, M. I. V. **Redes Neurais Artificiais: Uma Abordagem Para Sala de Aula**. Atena Editora, 2019.
- Glorot, X.; Bordes, A. ; Bengio, Y. **Deep sparse rectifier neural networks**. In: Proceedings of the fourteenth international conference on artificial intelligence and statistics, p. 315–323, 2011.
- Gomes, G. S. d. S.; Ludermir, T. B. ; Lima, L. M. Comparison of new activation functions in neural network for forecasting financial time series. **Neural Computing and Applications**, v.20, n.3, p. 417–439, 2011.
- Guresen, E.; Kayakutlu, G. Definition of artificial neural networks with comparison to other networks. **Procedia Computer Science**, v.3, p. 426–433, 2011.
- Hara, K.; Nakayamma, K. **Comparison of activation functions in multilayer neural network for pattern classification**. In: Neural Networks, 1994. IEEE World Congress on Computational Intelligence., 1994 IEEE International Conference on, volume 5, p. 2997–3002. IEEE, 1994.
- Hochreiter, S.; Schmidhuber, J. Long short-term memory. **Neural computation**, v.9, n.8, p. 1735–1780, 1997.
- Hopfield, J. J. Neural networks and physical systems with emergent collective computational abilities. **Proceedings of the national academy of sciences**, v.79, n.8, p. 2554–2558, 1982.
- Hu, J.; Gao, P.; Yao, Y. ; Xie, X. **Traffic flow forecasting with particle swarm optimization and support vector regression**. In: 17th International IEEE Conference on Intelligent Transportation Systems (ITSC), p. 2267–2268. IEEE, 2014.

- Huang, W.; Song, G.; Hong, H. ; Xie, K. Deep architecture for traffic flow prediction: deep belief networks with multitask learning. **IEEE Transactions on Intelligent Transportation Systems**, v.15, n.5, p. 2191–2201, 2014.
- Jiang, C.; Zhang, H.; Ren, Y.; Han, Z.; Chen, K.-C. ; Hanzo, L. Machine learning paradigms for next-generation wireless networks. **IEEE Wireless Communications**, v.24, n.2, p. 98–105, 2016.
- Kingma, D. P.; Ba, J. Adam: A method for stochastic optimization. **arXiv preprint arXiv:1412.6980**, 2014.
- Loce, R. P.; Bala, R. ; Trivedi, M. **Computer Vision and Imaging in Intelligent Transportation Systems**. John Wiley & Sons, 2017.
- Maas, A. L.; Hannun, A. Y. ; Ng, A. Y. **Rectifier nonlinearities improve neural network acoustic models**. In: Proc. icml, p. 3, 2013.
- Dianne Medeiros and Helio Cunha and Martin Andreoni Lopez and Luiz Magalhaes and Edelberto Silva and Alex Vieira and Natalia Fernandes. **Análise de dados em redes sem fio de grande porte: Processamento em fluxo em tempo real, tendências e desafios**. In: SBRC 2019 - Minicurso, mai 2019.
- Menon, A.; Mehrotra, K.; Mohan, C. K. ; Ranka, S. Characterization of a class of sigmoid functions with applications to neural networks. **Neural Networks**, v.9, n.5, p. 819–835, 1996.
- De Myttenaere, A.; Golden, B.; Le Grand, B. ; Rossi, F. Mean absolute percentage error for regression models. **Neurocomputing**, v.192, p. 38–48, 2016.
- Oda, T.; Obukata, R.; Yamada, M.; Hiyama, M.; Barolli, L. ; Takizawa, M. **A neural network based user identification for tor networks: Comparison analysis of different activation functions using friedman test**. In: Network-Based Information Systems (NBIS), 2016 19th International Conference on, p. 480–487. IEEE, 2016.
- Principe, J. C.; Euliano, N. R. ; Lefebvre, W. C. **Neural and adaptive systems: fundamentals through simulations**, volume 672. Wiley New York, 2000.
- Qolomany, B.; Maabreh, M.; Al-Fuqaha, A.; Gupta, A. ; Benhaddou, D. **Parameters optimization of deep learning models using particle swarm optimization**. In: 2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC), p. 1285–1290. IEEE, 2017.
- Fu, R.; Zhang, Z. ; Li, L. **Using LSTM and GRU neural network methods for traffic flow prediction**. In: 2016 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC), p. 324–328, Nov 2016.
- Rosenblatt, F. The perceptron: a probabilistic model for information storage and organization in the brain. **Psychological review**, v.65, n.6, p. 386, 1958.
- Russell, I. The delta rule. **University of Hartford, West Hartford. Accessed**, v.5, 2012.
- Russell, S. J.; Norvig, P. **Artificial intelligence: a modern approach**. Malaysia; Pearson Education Limited,, 2016.

-
- Sazli, M. H. A brief review of feed-forward neural networks. **Communications, Faculty of Science, University of Ankara**, v.50, n.1, p. 11–17, 2006.
- Tan, T. G.; Teo, J. ; Anthony, P. **A comparison of two sigmoidal-type activation functions in video game controller evolution**. In: Sustainable Utilization and Development in Engineering and Technology (STUDENT), 2011 IEEE Conference on, p. 90–94. IEEE, 2011.
- Zhang, Y.; Ke, Z. **On hyperbolic sine activation functions used in znn for time-varying matrix square roots finding**. In: Systems and Informatics (ICSAI), 2012 International Conference on, p. 740–744. IEEE, 2012.

A Apêndice

Neste capítulo se encontra os restantes dos gráficos gerados para os modelos que não foram classificados entre os 2 melhores de cada cenário.

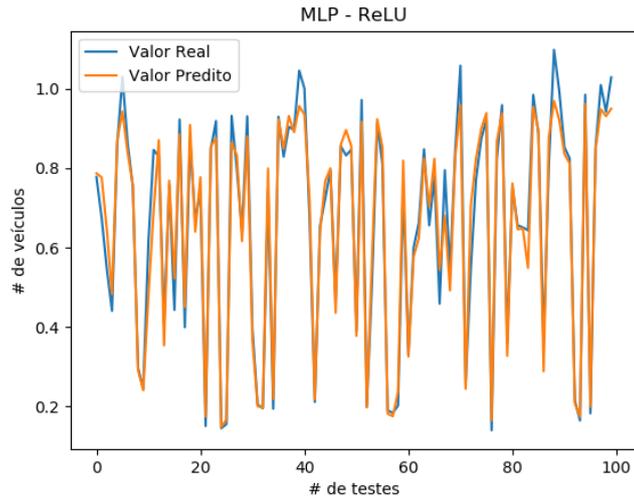


Figura A.1: Comparando valor predito com valor real para modelo MLP com função ReLU. Linha azul corresponde ao valor real e linha laranja ao valor predito. Cenário: Tráfego de Veículos.

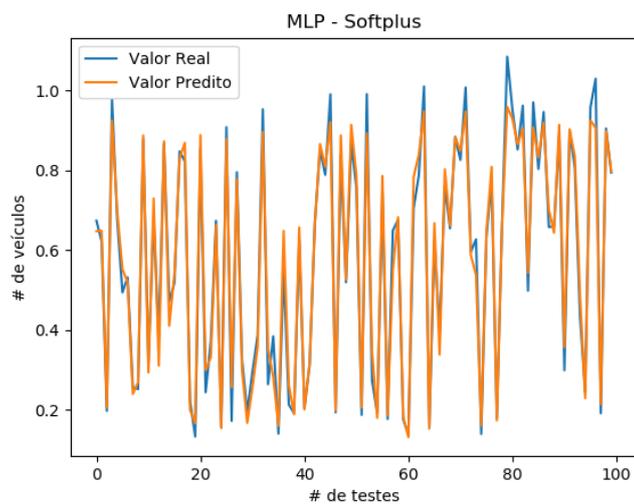


Figura A.2: Comparando valor predito com valor real para modelo MLP com função *Softplus*. Linha azul corresponde ao valor real e linha laranja ao valor predito. Cenário: Tráfego de Veículos.

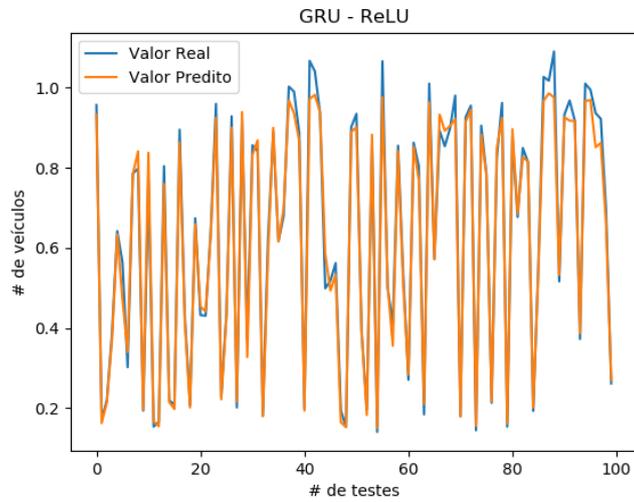


Figura A.3: Comparando valor predito com valor real para modelo GRU com função ReLU. Linha azul corresponde ao valor real e linha laranja ao valor predito. Cenário: Tráfego de Veículos.

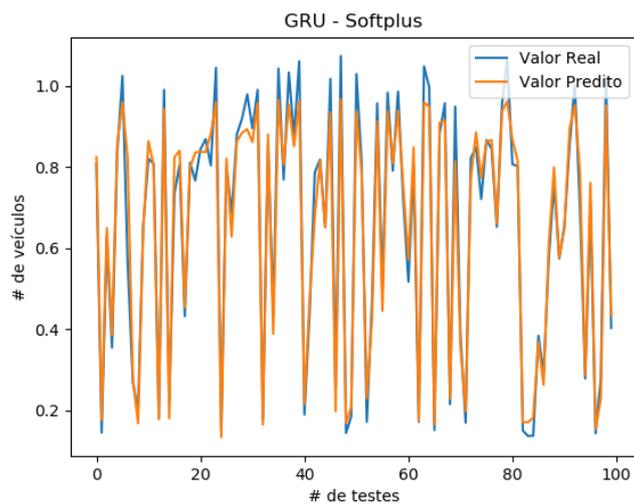


Figura A.4: Comparando valor predito com valor real para modelo GRU com função *Softplus*. Linha azul corresponde ao valor real e linha laranja ao valor predito. Cenário: Tráfego de Veículos.

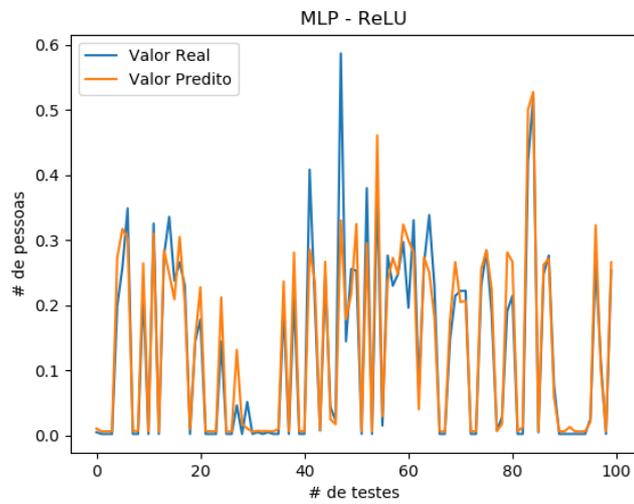


Figura A.5: Comparando valor predito com valor real para modelo MLP com função ReLU. Linha azul corresponde ao valor real e linha laranja ao valor predito. Cenário: Rede sem fio. Local: Engenharia

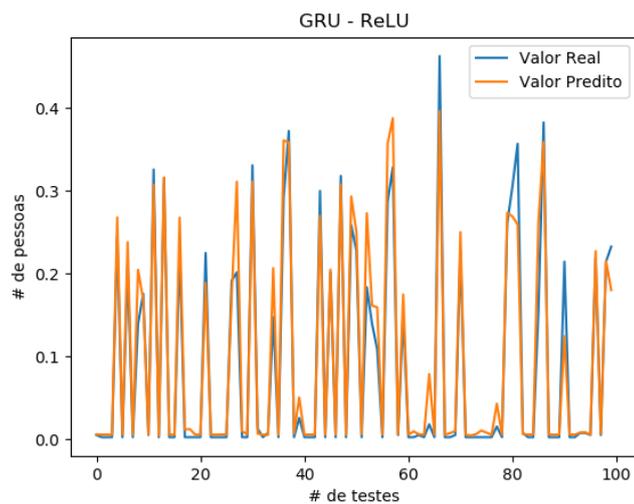


Figura A.6: Comparando valor predito com valor real para modelo GRU com função ReLU. Linha azul corresponde ao valor real e linha laranja ao valor predito. Cenário: Rede sem fio. Local: Engenharia

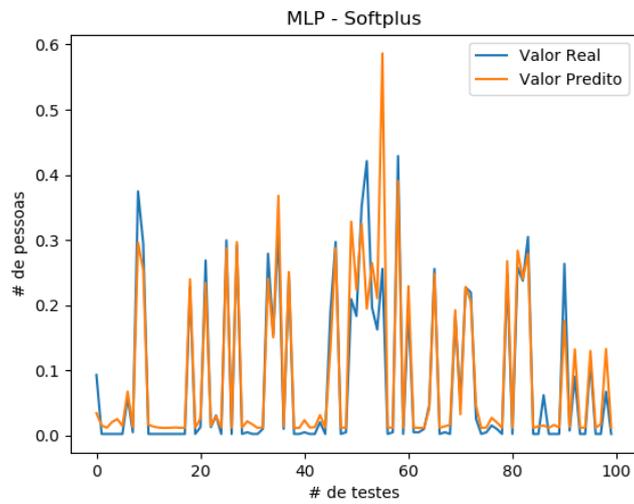


Figura A.7: Comparando valor predito com valor real para modelo MLP com função *Softplus*. Linha azul corresponde ao valor real e linha laranja ao valor predito. Cenário: Rede sem fio. Local: Engenharia

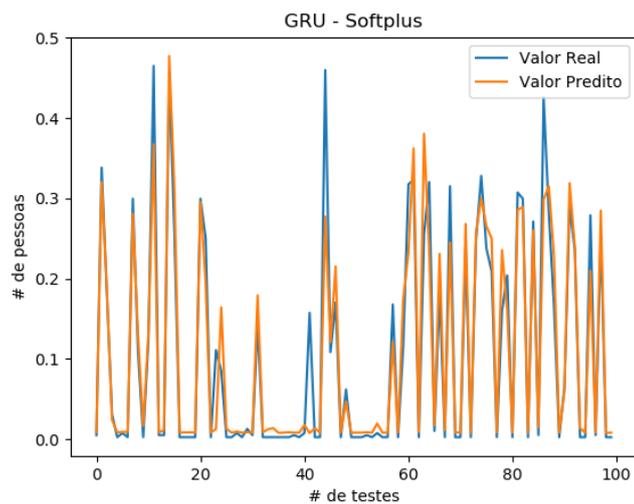


Figura A.8: Comparando valor predito com valor real para modelo GRU com função *Softplus*. Linha azul corresponde ao valor real e linha laranja ao valor predito. Cenário: Rede sem fio. Local: Engenharia

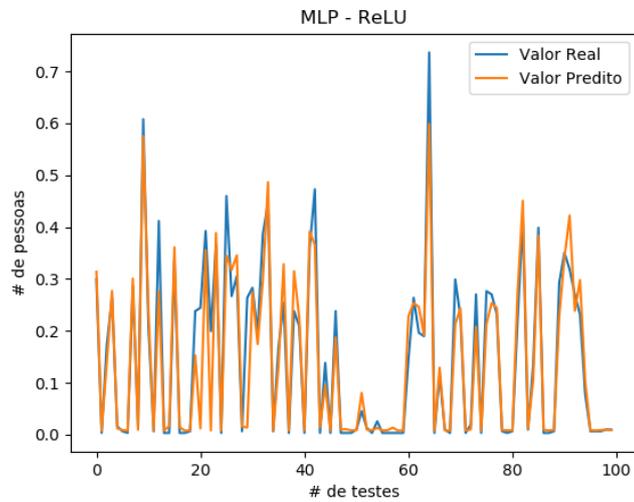


Figura A.9: Comparando valor predito com valor real para modelo MLP com função ReLU. Linha azul corresponde ao valor real e linha laranja ao valor predito. Cenário: Rede sem fio. Local: ICE

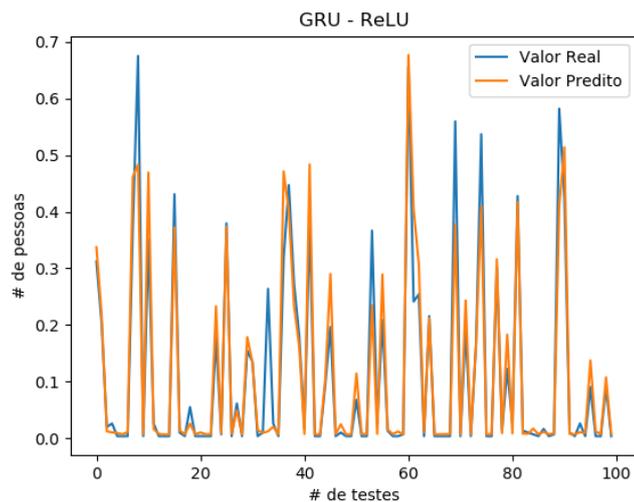


Figura A.10: Comparando valor predito com valor real para modelo GRU com função ReLU. Linha azul corresponde ao valor real e linha laranja ao valor predito. Cenário: Rede sem fio. Local: ICE

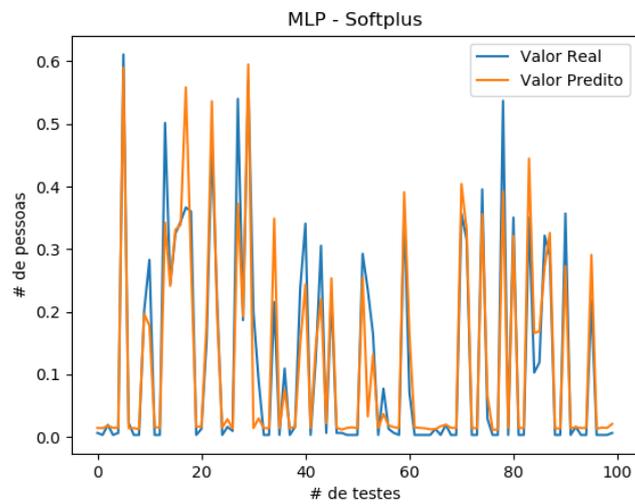


Figura A.11: Comparando valor predito com valor real para modelo MLP com função *Softplus*. Linha azul corresponde ao valor real e linha laranja ao valor predito. Cenário: Rede sem fio. Local: ICE

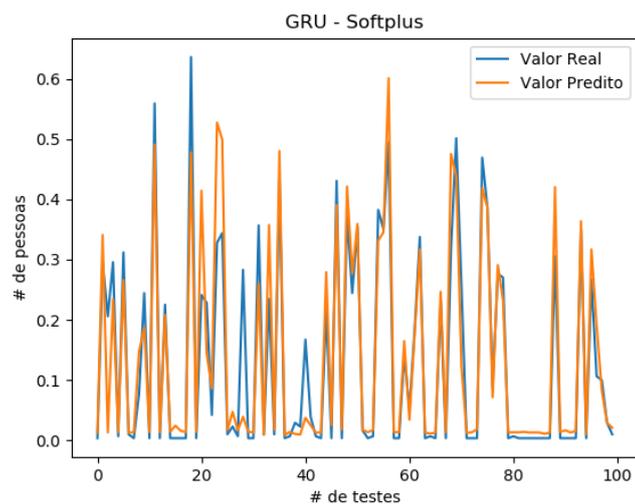


Figura A.12: Comparando valor predito com valor real para modelo GRU com função *Softplus*. Linha azul corresponde ao valor real e linha laranja ao valor predito. Cenário: Rede sem fio. Local: ICE

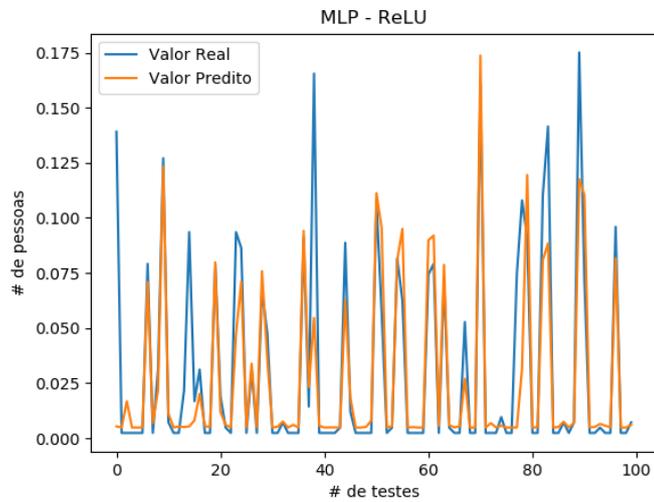


Figura A.13: Comparando valor predito com valor real para modelo MLP com função ReLU. Linha azul corresponde ao valor real e linha laranja ao valor predito. Cenário: Rede sem fio. Local: IAD

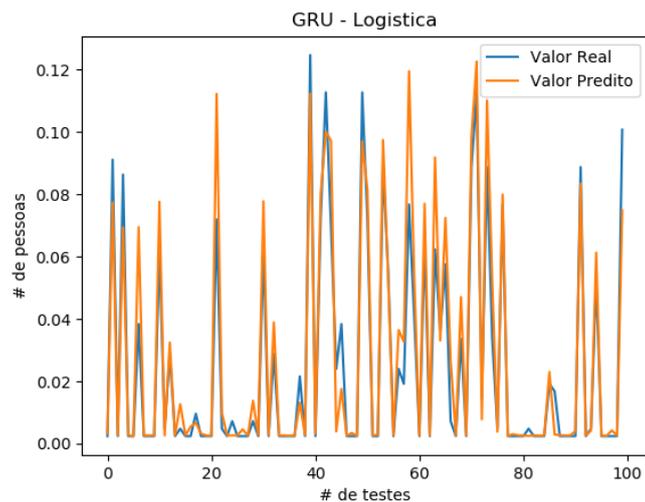


Figura A.14: Comparando valor predito com valor real para modelo GRU com função Logística. Linha azul corresponde ao valor real e linha laranja ao valor predito. Cenário: Rede sem fio. Local: IAD

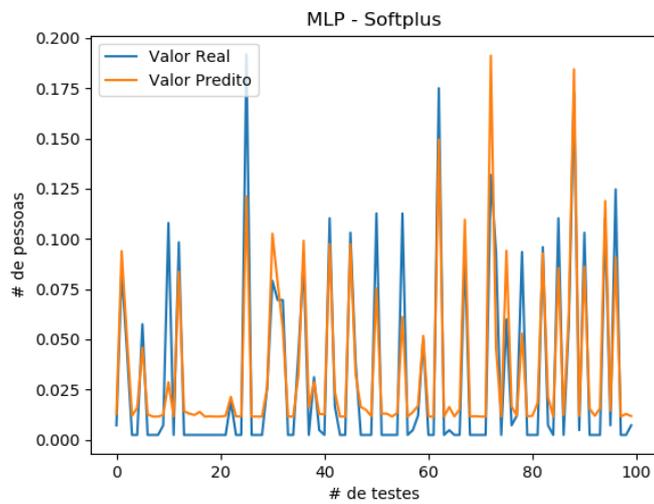


Figura A.15: Comparando valor predito com valor real para modelo MLP com função *Softplus*. Linha azul corresponde ao valor real e linha laranja ao valor predito. Cenário: Rede sem fio. Local: IAD

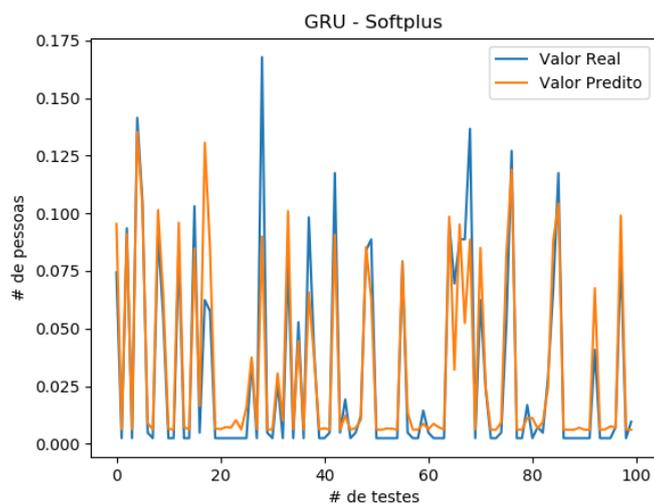


Figura A.16: Comparando valor predito com valor real para modelo GRU com função *Softplus*. Linha azul corresponde ao valor real e linha laranja ao valor predito. Cenário: Rede sem fio. Local: IAD