



**UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE TECNOLOGIA - ITEC
FACULDADE DE ENGENHARIA ELÉTRICA E BIOMÉDICA
CURSO DE ENGENHARIA ELÉTRICA**

RODRIGO GOMES DUTRA 201607140080

**RELATÓRIO DE REDES NEURAIS APLICADAS PARA A PREDIÇÃO DA
VELOCIDADE DO VENTO**

**Belém
2019**

SUMÁRIO

1	INTRODUÇÃO	2
1.1	BANCO DE DADOS	2
2	DESENVOLVIMENTO	4
2.1	RESULTADOS DA MELHOR TOPOLOGIA	4
3	CONCLUSÃO	6
	REFERÊNCIAS	7

1 INTRODUÇÃO

Este relatório é relativo à atividade proposta em sala de aula na disciplina de Inteligência computacional, essa atividade consiste em aplicar uma rede neural para a resolução de um problema de séries temporais. A rede neural será aplicada em um banco de dados de velocidade do vento, com o objetivo de prever o próximo valor da velocidade do vento dado as entradas definidas no modelo.

O banco de dados consiste em dados de velocidade do vento o qual será detalhado mais adiante nesse documento.

O software que será utilizado para os fins de gerar a estrutura da rede neural será o Python3(ROSSUM, 1995), com auxílio das bibliotecas Tensorflow(ABADI et al., 2016) e Scikit(PEDREGOSA et al., 2011), as quais foram utilizadas respectivamente para gerar a estrutura da rede neural e manipular a base de dados.

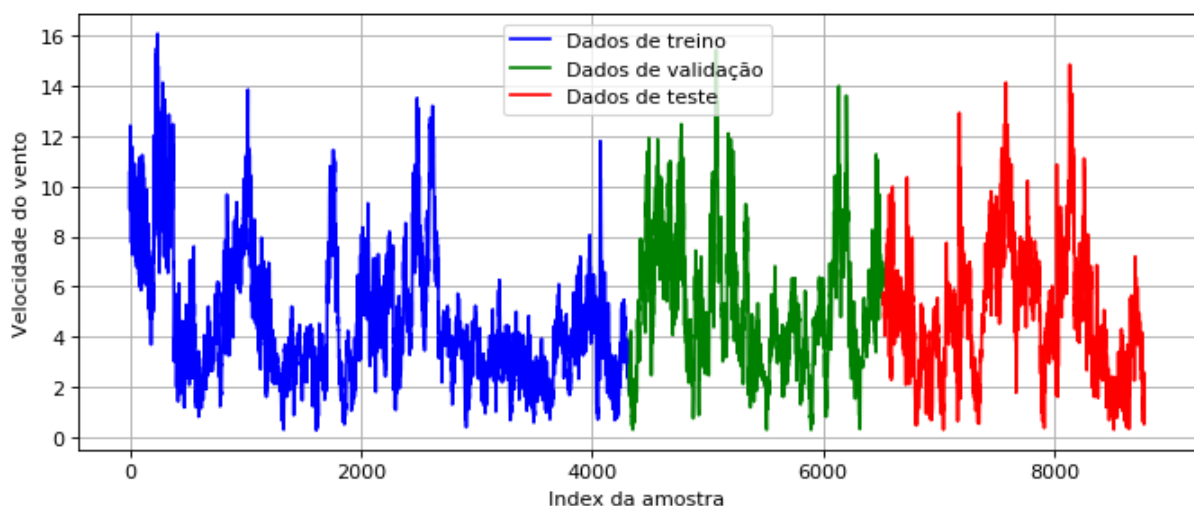
1.1 BANCO DE DADOS

O banco de dados utilizado foi consiste em dados de velocidade do vento em m/s coletados em 2 meses, abril e maio.

Tabela 1 – Distribuição de dados

	Lote de dados	Porcentagem	Abril	Maio	Total
0	Treino	49.20273348519362	4320	0	4320
1	Validação	25.056947608200456	0	2200	2200
2	Teste	25.72892938496583	0	2259	2259
3	Total	100.0	4320	4460	8780

Figura 1 – A série temporal



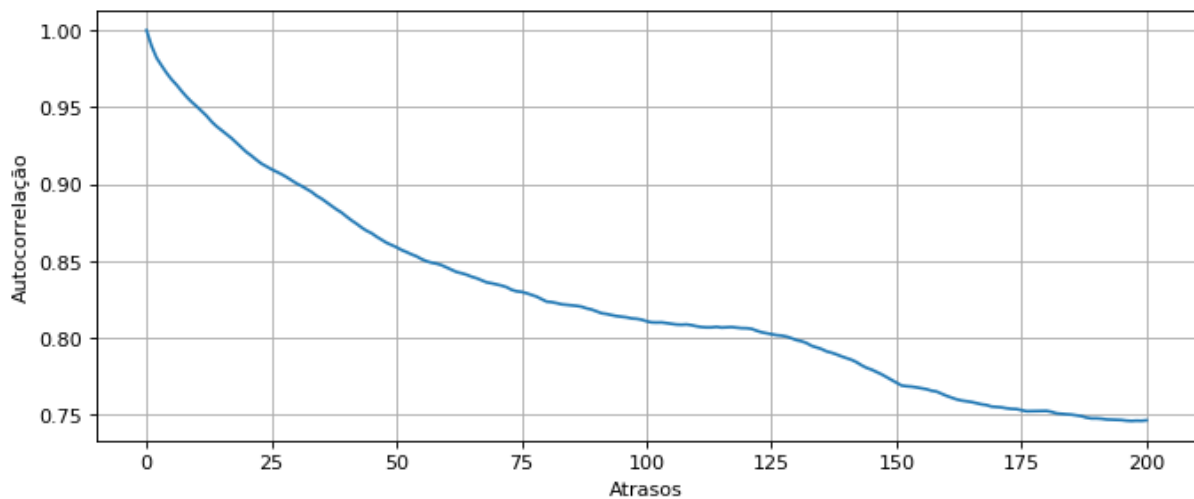
A distribuição de dados demonstrado na tabela 1, demonstra que os dados de abril foram utilizados somente para treino, enquanto os dados de maio foram utilizados para validação e teste.

Para a normalização da base de dados foi necessário utilizar uma equação que limitasse os dados em um escopo de 0 a 1, de forma que o maior número positivo deverá assumir o valor de 1, enquanto que o menor número irá assumir o valor 0

$$x'_i = \frac{(x_i - x_{min}) * (L_{max} - L_{min})}{x_{max} - x_{min}} + L_{min} \quad (1.1)$$

A fim de avaliar as entradas de um sistema baseado em séries temporais primeiramente é necessário analisar a correlação com a própria série, dado que a série é uni variada, para isso foi realizado o gráfico de autocorrelação utilizando os dados de treino:

Figura 2 – Autocorrelação da série temporal



A partir do gráfico da autocorrelação linear da série temporal é possível definir as entradas do sistema, visto que para que um atraso no tempo da série temporal ser uma entrada viável esta deverá possuir uma forte autocorrelação com o passo no tempo 0. Assim, foi definido que as entradas do sistema seriam os atrasos de até o numero 125, de forma que o ponto 125 do gráfico indica quando a autocorrelação deixa a marca de 0.8.

Dessa forma a escopo de um sistema baseado em séries temporais pode ser definido como :

$$x_n = MLP(x_n - 1, x_n - 2, x_n - 3, \dots, x_n - 125) \quad (1.2)$$

Assim as entradas da base de dados de treino, validação e teste foram janelas com 125 amostras atrasadas em relação a amostra que representa o valor real.

2 DESENVOLVIMENTO

Para realizar a análise e classificação do banco de dados de vinhos utilizando a redes neurais foi criado um código utilizado o Python3 (ROSSUM, 1995).

Foi utilizado o algoritmo de Adam para o treinamento da rede neural, com uma taxa de aprendizagem de, 0.001 e decaimento por época de 0.0005. Utilizando-se dos dados de treinamento definidos na Tabela 1. Para a função de erro, depois de realizado-se a análise da melhor função baseado-se no desempenho, foi utilizado a função do Erro quadrático médio.

A fim de estudar o comportamento da rede neural conforme o número de neurônios na camada escondida, foi necessário mais que uma tentativa para cada quantidade de neurônios, para assim tentar tirar influencia do caráter randômico no erro inerente em relação a saída da rede neural e a saída real do banco de dados para as mesmas entradas.

Nesse sentido foram realizadas 2 repetições de criação da rede neural variando-se o número de neurônios, utilizando-se dos valores: 1;5;10;15;30 de neurônios, de forma que em cada repetição era salvo os erros de teste relativas a atual iteração. Dessa forma, foi possível gerar uma tabela com os valores de erro médio quadrático coletados.

Tabela 2 – MSE vs número de neurônios

Neurônios	MSE 1ª repetição	MSE 2ª repetição	MSE médio
5	2.529916225261784	2.1159133353513444	2.322914780306564
10	1.9011647226797765	2.0167100242797735	1.958937373479775
15	1.7312334709983244	1.6864874023466125	1.7088604366724685
30	1.6861438731937628	1.538252724963972	1.6121982990788672

2.1 RESULTADOS DA MELHOR TOPOLOGIA

A melhor topologia seguindo os resultados da tabela 2 foi a topologia com 30 neurônios na camada escondida, o comportamento dessa topologia pode ser melhor observado através de seus resultados de treinamento e teste, visualizando-se a evolução do gráfico do erro e o gráfico de teste.

Figura 3 – Gráfico do erro

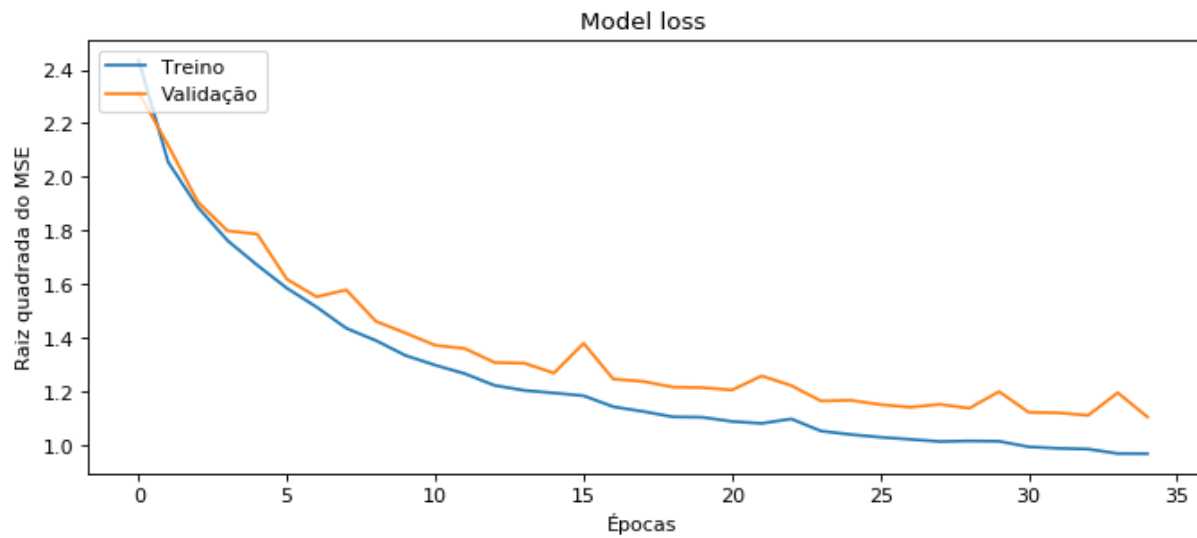
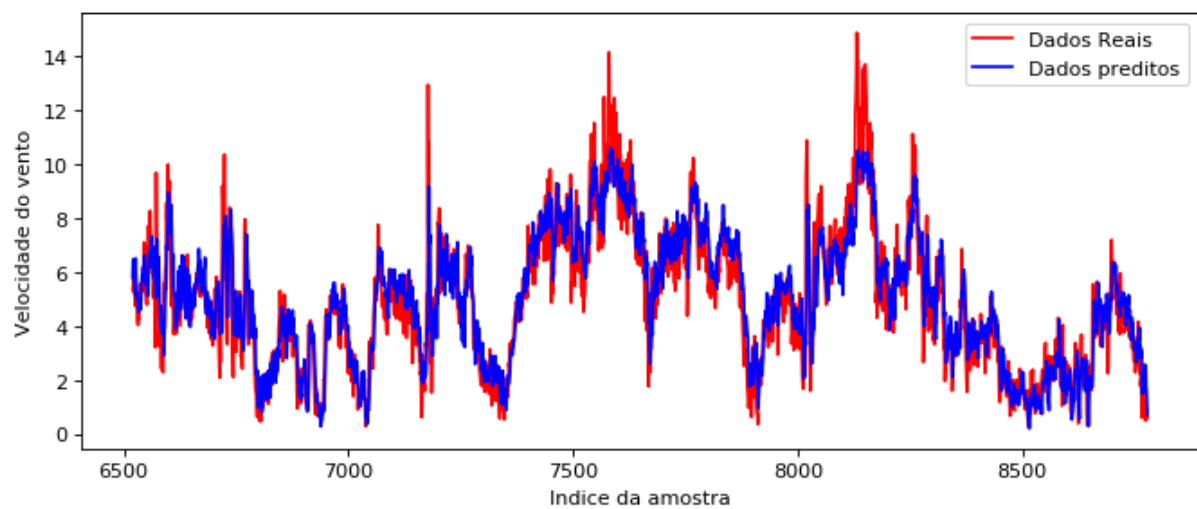


Figura 4 – Resultados da saída da rede



3 CONCLUSÃO

A partir dos resultados das rede neural no problema de séries temporais proposto é possível concluir que avaliar o número de neurônios de uma rede neural é bastante importante para ampliar a acurácia da rede, de forma que foi possível reduzir o erro médio quadrático em praticamente 1 comparando-se a topologia com 5 e 30 neurônios.

De posse dos resultados da melhor topologia pode-se concluir também que esta apresentou uma boa precisão, porém ao observar a figura 4 é notório que a rede neural não conseguiu prever com muita precisão os comportamentos de pico da série temporal. Dessa forma, outras análises e topologias poderiam ser feitas a fim de melhorar esse comportamento.

É possível também concluir a importância de modelos de séries temporais para o problema proposto, visto que um modelo opitimo poderia ser aplicado para previsões meteorológicas do tipo e além disso poderia ser utilizado para análises da velocidade do vento para a viabilização de projetos de geradores eólicos, prevendo assim também a potencia gerada pelo gerador eólico.

REFERÊNCIAS

ABADI, M. et al. Tensorflow: A system for large-scale machine learning. In: **12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)**. [S.l.: s.n.], 2016. p. 265–283.

PEDREGOSA, F. et al. Scikit-learn: Machine learning in python. **Journal of machine learning research**, v. 12, n. Oct, p. 2825–2830, 2011.

ROSSUM, G. Python reference manual. CWI (Centre for Mathematics and Computer Science), 1995.