

Relatório da Prova 1 de Aprendizado de Máquina

César Augusto Julio da Silva

Instituto de Computação
Universidade Federal do Rio de Janeiro (UFRJ) – Rio de Janeiro, RJ – Brasil
cesarsilva06@dcc.ufrj.br

Abstract. *This report contains an analysis of the results linked to the code created in accord with the guidelines related to the replacement exam of the first exam of the machine learning course, lectured by the teacher João Carlos.*

Resumo. *Este relatório contém uma análise dos resultados vinculados ao código criado de acordo com as diretrizes relacionadas ao exame substitutivo da primeira prova da matéria de aprendizado de máquina, lecionada pelo professor João Carlos.*

1. Introdução

O objetivo deste relatório é complementar o código criado no *Google Collaboratory* com os gráficos e informações relevantes e análises dos resultados essenciais do projeto.

2. Primeiro Teste

O primeiro teste consiste de uma implementação básica de uma rede neural utilizando as diretrizes propostas pelo professor como parâmetros e fazendo uma análise dos *scores* e das funções de perda, finalizando com a amostra dos mesmos e um gráfico das funções.

	K	Score de Treinamento	Score de Teste	Ultima Perda
0	1	0.731481	0.703704	0.510589
1	2	0.851852	0.851852	0.48916
2	3	0.842593	0.740741	0.475293
3	4	0.861111	0.740741	0.483593
4	5	0.814815	0.962963	0.513673

Figura 1. Dados de Score e Perda.

É possível observar que apesar do score de Teste subir a perda final, em geral, se mantém constante próximo de 0.5, enquanto o score de treinamento oscila ao redor de 0.8.

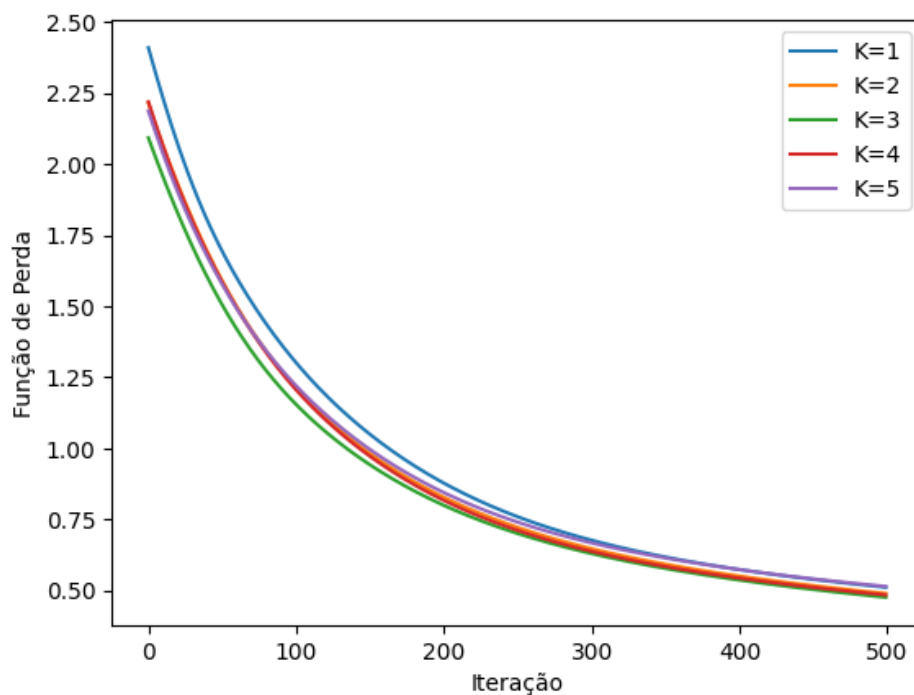


Figura 2. Gráfico das Funções de Perdas de cada Fold.

Aqui no gráfico podemos observar que todas funções tem uma queda abrupta até aproximadamente a iteração 150, diminuindo suas inclinações até a iteração 500, a qual todas chegam juntas (com uma leve subida no $K = 5$). Na minha opinião, isso pode ser um sinal da rede estar próxima de um overfitting, já que a perda não está mais diminuindo, apesar do score de treinamento continuar subindo.

3. Alteração dos Parâmetros

As alterações escolhidas foram três: modificação do número de iterações, mudança no número de neurônios (mantendo uma camada) e troca do número de camadas (mantendo dois neurônios).

Esses parâmetros foram escolhidos pois, respectivamente: é muito provável que 500 não fosse o suficiente, dado que a perda não decrescia mais; é possível que mais neurônios tragam um score melhor nos primeiros folds já que dois é um número bem pequeno (menor do que até o padrão da própria função nativa da linguagem); e o aumento de camadas foi escolhido, sinceramente, pois não sabia prever qual fenômeno aconteceria apesar de estar inclinado a uma melhoria no desempenho (alcance de um score mais alto, mais rápido, e com menos perda).

3.1. Mudança no número de iterações.

Foi escolhido alterar para 4 valores extras: 100, 1000 (mil), 10000 (dez mil) e 100000 (cem mil) iterações e averiguar as mudanças nas mesmas. A escolha dos números foi arbitrária

Numero de Iteracoes: 100				
	K	Score de Treinamento	Score de Teste	Ultima Perda
0	1	0.324074	0.518519	1.306895
1	2	0.37963	0.37037	1.217762
2	3	0.407407	0.259259	1.159064
3	4	0.361111	0.407407	1.211018
4	5	0.388889	0.296296	1.22512
Numero de Iteracoes: 1000				
	K	Score de Treinamento	Score de Teste	Ultima Perda
0	1	0.935185	0.925926	0.345125
1	2	0.944444	0.925926	0.319323
2	3	0.935185	0.888889	0.301115
3	4	0.925926	0.888889	0.317847
4	5	0.925926	1	0.346879
Numero de Iteracoes: 10000				
	K	Score de Treinamento	Score de Teste	Ultima Perda
0	1	0.972222	0.962963	0.161179
1	2	0.962963	0.962963	0.141851
2	3	0.972222	0.962963	0.131627
3	4	0.972222	0.925926	0.147076
4	5	0.953704	1	0.15375
Numero de Iteracoes: 100000				
	K	Score de Treinamento	Score de Teste	Ultima Perda
0	1	0.972222	0.962963	0.161179
1	2	0.962963	0.962963	0.141851
2	3	0.972222	0.962963	0.131627
3	4	0.972222	0.925926	0.147076
4	5	0.953704	1	0.15375

Figura 3. Tabelas com dados das mudanças de iterações.

O primeiro dos parâmetros indica uma alta perda e um baixo score, tanto no teste quanto no treinamento. Isso demonstra que a rede está em underfitting devido ao pequeno número de iterações não permiti-la adequar-se de forma correta.

A partir daí, vemos um grande aumento tanto no score de treinamento quanto no de teste, ao mesmo tempo que vemos uma queda drástica na perda. Em algum instante, provavelmente antes da iteração 10000, a rede já conseguiu tirar o máximo de

treinamento e por isso os dados da tabela de 10000 iterações estão iguais a anterior. Assim, pode-se deduzir que o número ótimo de iterações está antes de dez mil.

3.2. Mudança no número de neurônios

Aqui, foram escolhidos o valor inicial e foram elevados ao cubo cada um dos conseguintes, formando então os valores: 2, 8, 32 e 128.

Numero de Neuronios: 2				
	K	Score de Treinamento	Score de Teste	Ultima Perda
0	1	0.731481	0.703704	0.510589
1	2	0.851852	0.851852	0.48916
2	3	0.842593	0.740741	0.475293
3	4	0.861111	0.740741	0.483593
4	5	0.814815	0.962963	0.513673
Numero de Neuronios: 8				
	K	Score de Treinamento	Score de Teste	Ultima Perda
0	1	0.925926	0.962963	0.412755
1	2	0.981481	0.962963	0.368213
2	3	0.972222	0.925926	0.34911
3	4	0.962963	0.962963	0.36383
4	5	0.981481	0.962963	0.415864
Numero de Neuronios: 32				
	K	Score de Treinamento	Score de Teste	Ultima Perda
0	1	0.972222	0.962963	0.242247
1	2	0.972222	0.962963	0.222408
2	3	0.972222	0.925926	0.207406
3	4	0.972222	0.925926	0.230035
4	5	0.972222	1	0.240308
Numero de Neuronios: 128				
	K	Score de Treinamento	Score de Teste	Ultima Perda
0	1	0.972222	1	0.094347
1	2	0.990741	0.925926	0.078112
2	3	0.990741	0.962963	0.069548
3	4	0.981481	0.925926	0.082279
4	5	0.972222	1	0.093883

Figura 4. Tabelas com dados da mudança de neurônios.

Vamos pular a primeira execução (é a mesma do teste inicial). O fenômeno que pode ser observado em todas é o alto score praticamente em todos os conseguintes mas uma perda decrescente inversamente proporcional ao número de neurônios.

Contudo, não há muita oscilação dentro de um mesmo teste.

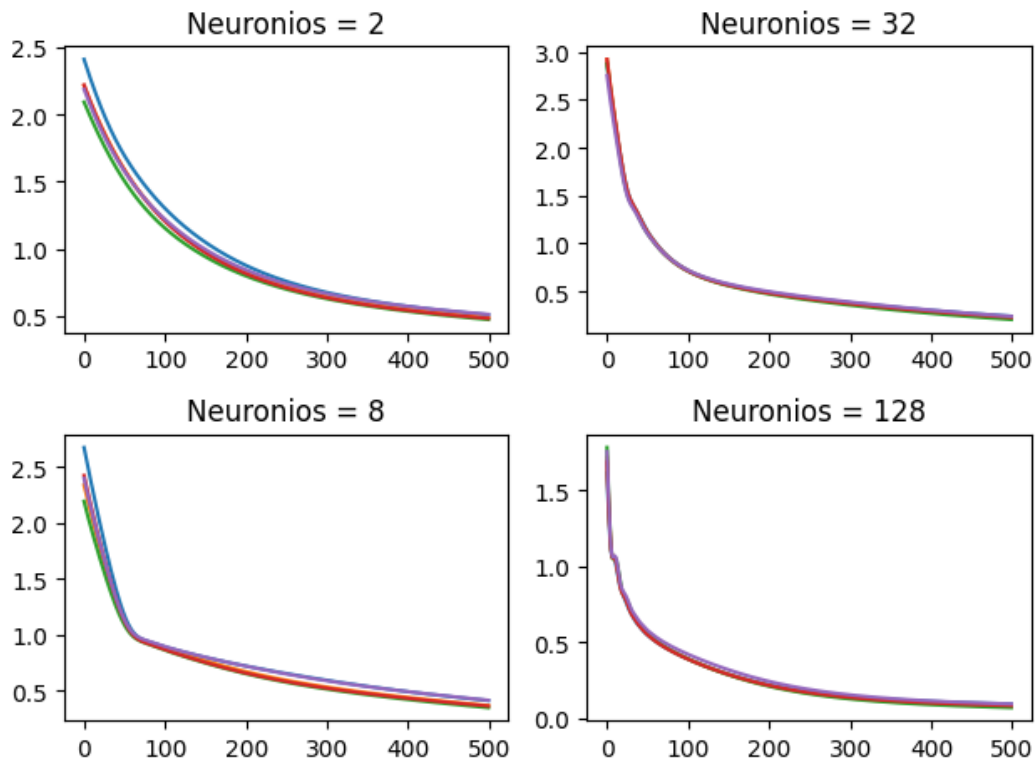


Figura 5. Gráficos de Perda dos testes com alteração no número de neurônios.

Esses gráficos foram desenhados para tentar observar o que ocorria no processo. Apesar das curvas serem similares, o fato de não vermos uma progressão no aumento do score de treinamento é uma boa indicação de *overfitting*. Um questionamento que tenho é se poderíamos ver uma explicação mais clara do que ocorreria caso o dataset usado fosse maior.

3.3. Mudança no número de camadas

A última modificação efetuada foi um aumento no número de camadas, de 1 para 4, gradualmente.

Numero de Camadas: 1				
	K	Score de Treinamento	Score de Teste	Ultima Perda
0	1	0.731481	0.703704	0.510589
1	2	0.851852	0.851852	0.48916
2	3	0.842593	0.740741	0.475293
3	4	0.861111	0.740741	0.483593
4	5	0.814815	0.962963	0.513673
Numero de Camadas: 2				
	K	Score de Treinamento	Score de Teste	Ultima Perda
0	1	0.666667	0.62963	0.642784
1	2	0.62963	0.555556	0.648019
2	3	0.611111	0.62963	0.656787
3	4	0.611111	0.740741	0.65966
4	5	0.638889	0.518519	0.652534
Numero de Camadas: 3				
	K	Score de Treinamento	Score de Teste	Ultima Perda
0	1	0.638889	0.555556	0.709882
1	2	0.759259	0.814815	0.653959
2	3	0.824074	0.740741	0.618219
3	4	0.787037	0.62963	0.652098
4	5	0.722222	0.925926	0.689651
Numero de Camadas: 4				
	K	Score de Treinamento	Score de Teste	Ultima Perda
0	1	0.814815	0.666667	0.469709
1	2	0.925926	0.925926	0.466874
2	3	0.898148	0.814815	0.454462
3	4	0.953704	1	0.458657
4	5	0.916667	0.962963	0.488112

Figura 6. Tabelas com dados dos testes de alteração do número de camadas.

Nesse último experimento, um fenômeno esquisito ocorre: ocorre uma oscilação bem grande nos scores de teste e treinamento.

Entre um *fold* e outro, há melhoria seguida de piora seguida de melhoria. Curiosamente, os testes 2 e 3 (número de camadas 2 e 3) tem uma piora no score treinamento em comparação com testes 1 e 4 (número de camadas 1 e 4). O mesmo acontecimento se mostra na perda, com o aumento da última registrada nesses testes em relação aos outros. Apesar de meus instintos estarem certos com relação a uma melhoria no score em geral (o teste 4 tem um desempenho bem melhor que o teste 1) há o questionamento se, para esse *dataset* específico, existe um número de camadas adequado ideal.

4. Análise do melhor resultado

Dado isso, comparamos os melhores de cada com o teste inicial.

Numero de Neuronios: 2				
	K	Score de Treinamento	Score de Teste	Ultima Perda
0	1	0.731481	0.703704	0.510589
1	2	0.851852	0.851852	0.48916
2	3	0.842593	0.740741	0.475293
3	4	0.861111	0.740741	0.483593
4	5	0.814815	0.962963	0.513673
Numero de Iteracoes: 10000				
	K	Score de Treinamento	Score de Teste	Ultima Perda
0	1	0.972222	0.962963	0.161179
1	2	0.962963	0.962963	0.141851
2	3	0.972222	0.962963	0.131627
3	4	0.972222	0.925926	0.147076
4	5	0.953704	1	0.15375
Numero de Camadas: 4				
	K	Score de Treinamento	Score de Teste	Ultima Perda
0	1	0.814815	0.666667	0.469709
1	2	0.925926	0.925926	0.466874
2	3	0.898148	0.814815	0.454462
3	4	0.953704	1	0.458657
4	5	0.916667	0.962963	0.488112
Numero de Neuronios: 128				
	K	Score de Treinamento	Score de Teste	Ultima Perda
0	1	0.972222	1	0.094347
1	2	0.990741	0.925926	0.078112
2	3	0.990741	0.962963	0.069548
3	4	0.981481	0.925926	0.082279
4	5	0.972222	1	0.093883

Figura 7. Comparação entre os melhores resultados

Dentre os que estão na figura acima, a minha escolha do “melhor” vai para o de 4 camadas, pois apesar de ter uma perda mais alto que os outros é muito difícil que ambos os testes de 10000 (dez mil) iterações e o de 128 neurônios não estejam em *overfitting*, mesmo com a divisão de 90:10.

Assim, a execução final foi feita no de 4 camadas, testando os 10% separados para validação. O score exibido foi 1.0 , mostrando que a rede conseguiu adequar-se ao *dataset*.

Referências

da Silva, J. C. P. (2023) “ICP363 - Introdução ao Aprendizado de Máquina. Aula 10 - Redes Neurais”. Disponível em: https://drive.google.com/drive/folders/16Fb9L-WNqZYHTlkv-4R_wpyZc5saNp0F
 sklearn.neural_network.MLPClassifier. Scikit Learn. Disponível em: https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier

er.html#sklearn.neural_network.MLPClassifier.predict

Creating multiple subplots using plot.subplots. **Matplotlib.** Disponível em:
https://matplotlib.org/stable/gallery/subplots_axes_and_figures/subplots_demo.html

How is it possible that validation loss is increasing while validation accuracy is increasing as well. **Stack Exchange.** Disponível em:
<https://stats.stackexchange.com/questions/282160/how-is-it-possible-that-validation-loss-is-increasing-while-validation-accuracy>