

Relatório - Projeto 1

Aprendizagem Profunda

Brian A. Nunes - IST1105399

Mariana Serrão - IST1105045

December 23, 2022

1 Questão 1

1.1 Exercício 1

a)

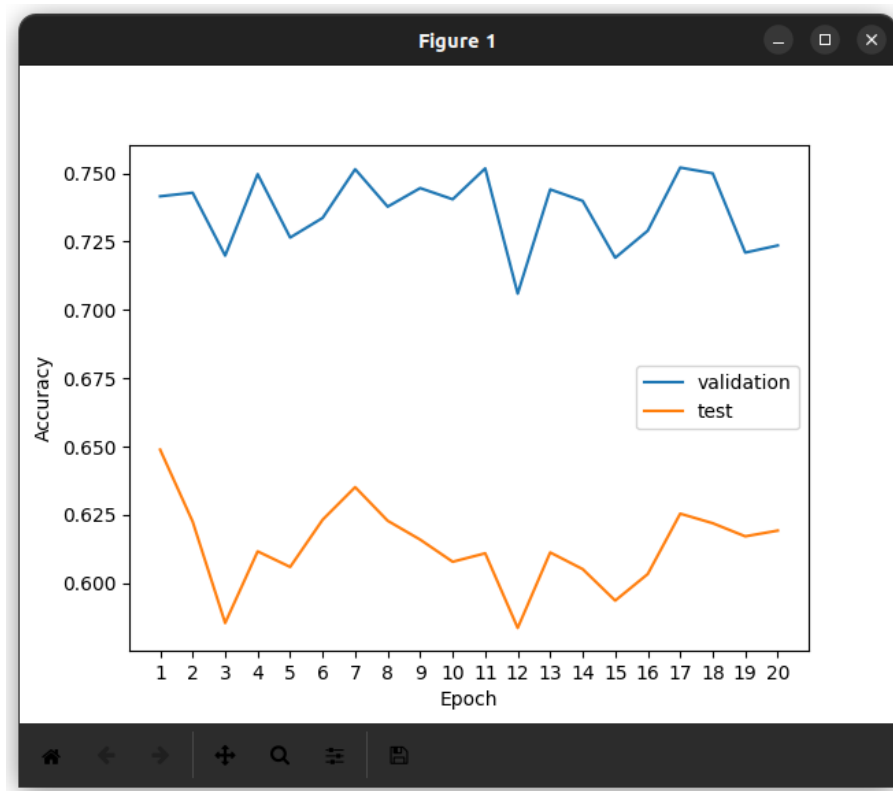


Gráfico 1: Acurácia com *Perceptron* por *epoch*

b)

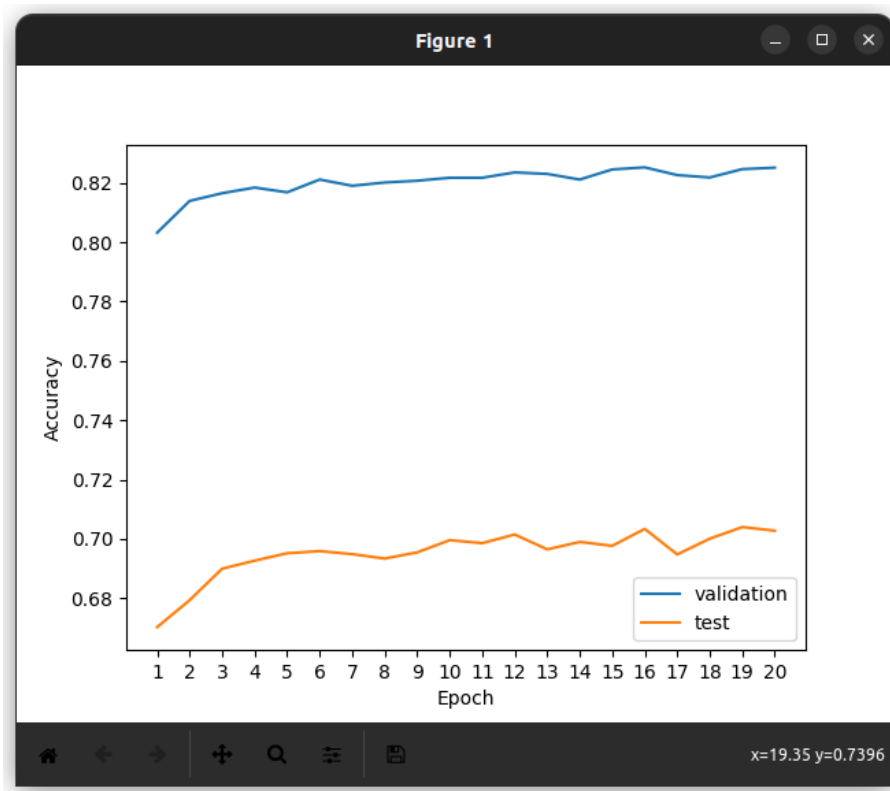


Gráfico 2: Acurácia com *Logistic Regression* por *epoch*

1.2 Exercício 2

a) *Perceptrons* isoladamente conseguem traçar retas em planos, o que já é uma boa ferramenta para a tomada de decisões com um computador, mas no que se refere a XOR's os *perceptrons* falham em entregar resultado satisfatório, entretanto com a utilização de camadas de neurônios que se alimentam é possível contornar esse problema. Nesta tarefa, em específico, a compreensão de imagens é demasiada complexa para que um simples neurônio consiga fazer uma boa distinção entre os caracteres escritos, já com a utilização de uma rede neuronal é possível alcançar níveis muito mais elevados de precisão.

b) Vide código.

2 Questão 2

2.1 Exercício 1

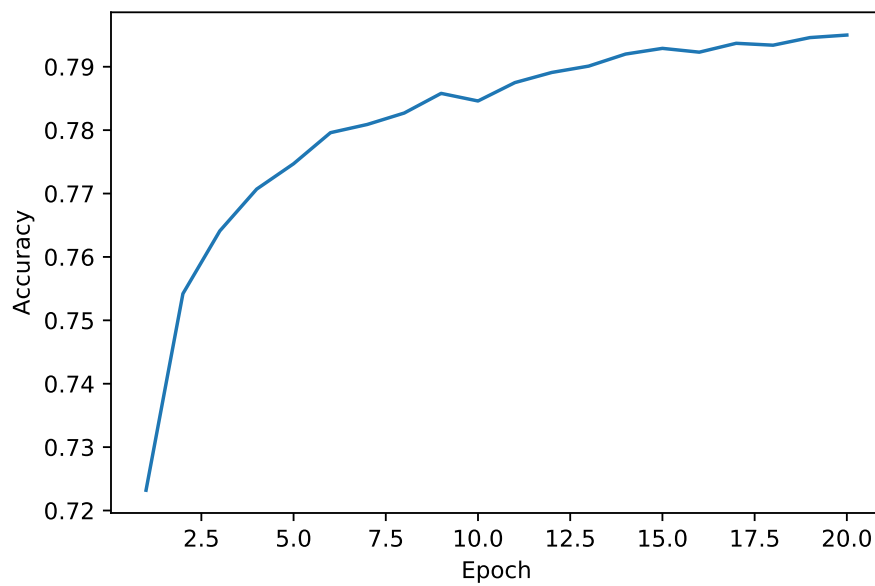


Gráfico 3: *Validation Accuracy* por *epoch* com *Learning Rate* igual a 0.001

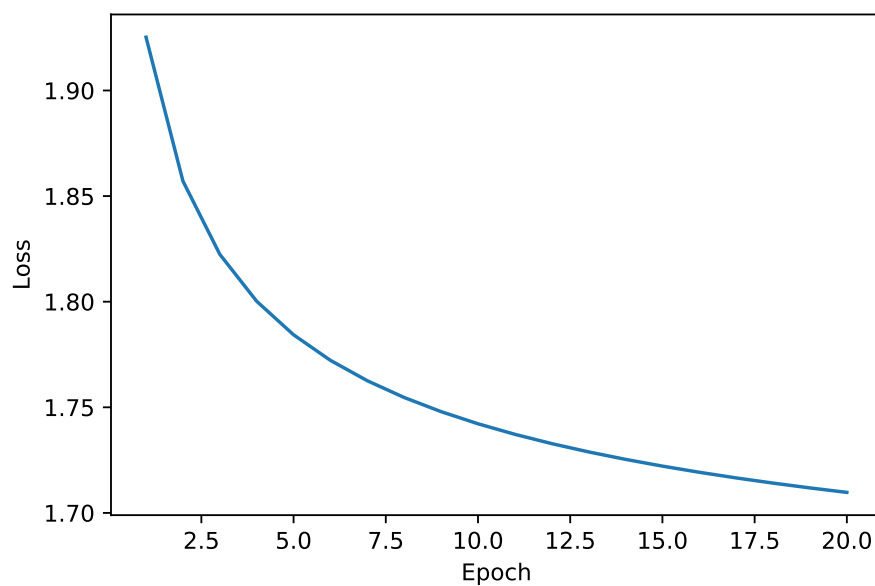


Gráfico 4: *Trainig Loss* por *epoch* com *Learning Rate* igual a 0.001

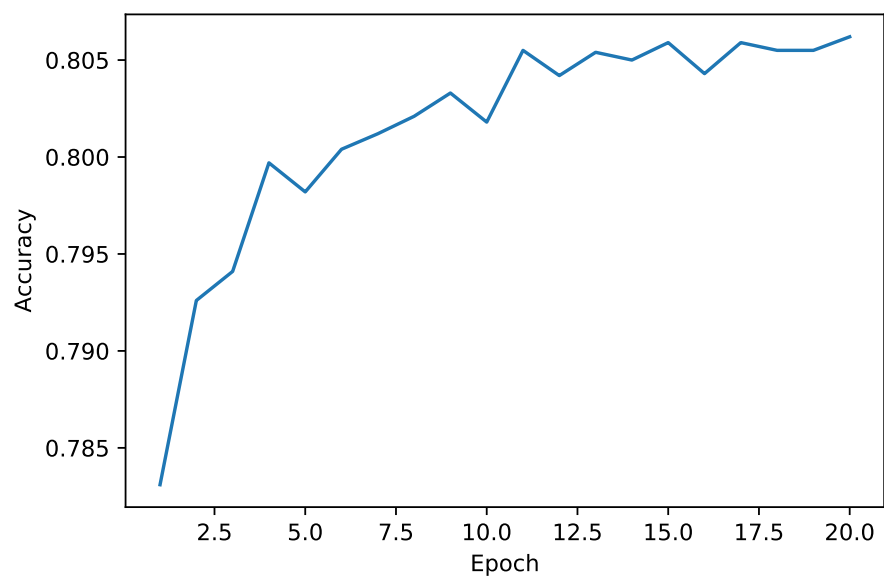


Gráfico 5: *Validation Accuracy* por *epoch* com *Learning Rate* igual a 0.01

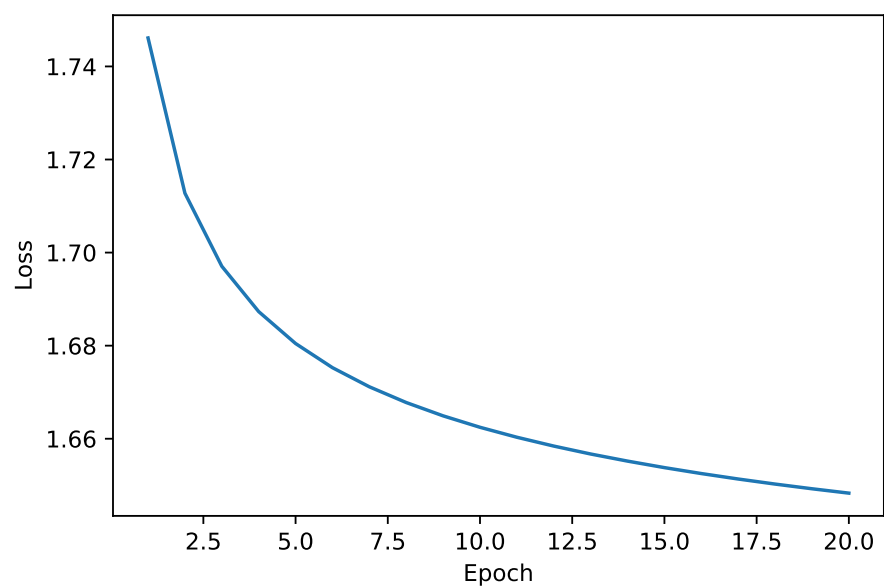


Gráfico 6: *Training Loss* por *epoch* com *Learning Rate* igual a 0.01

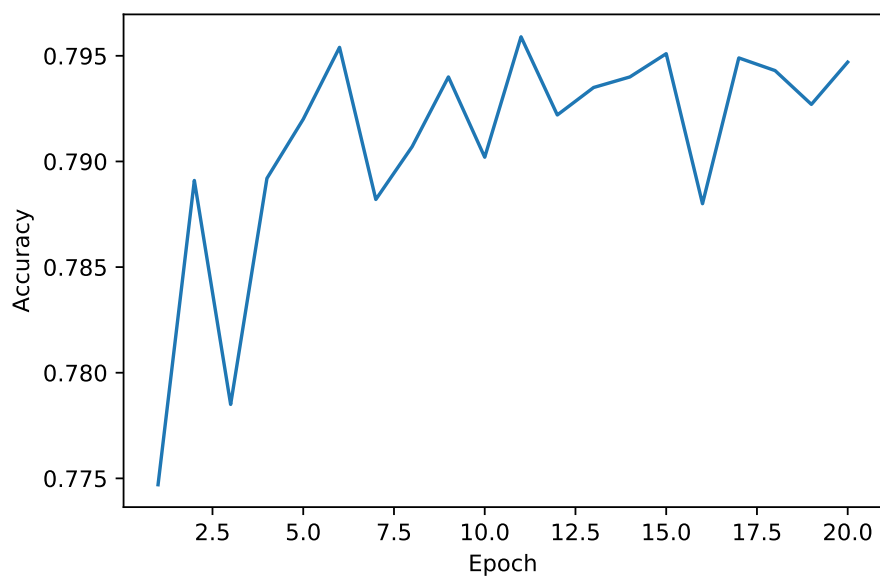


Gráfico 7: *Validation Accuracy* por *epoch* com *Learning Rate* igual a 0.1

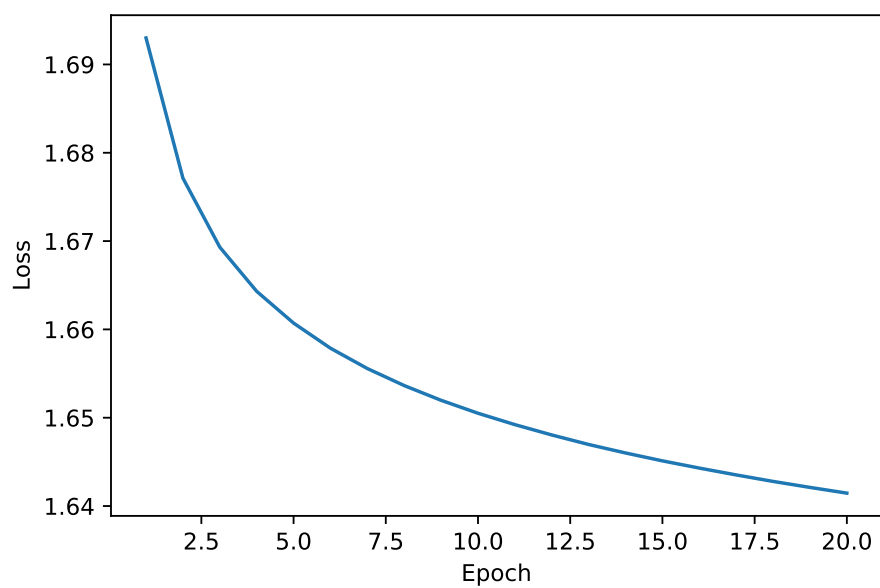


Gráfico 8: *Trainig Loss* por *epoch* com *Learning Rate* igual a 0.1

Learning Rate	0.001	0.01	0.1
Final Training Loss	1.7097	1.6483	1.6415
Final Validation Accuracy	0.795	0.8062	0.7947
Final Test Accuracy	0.6504	0.6641	0.6331

Tabela 1: Performance por *Learning Rate*

Como é possível ver na Tabela 1, a melhor configuração atingida foi com um *Learning Rate* igual a 0.01. As acurácias de validação e teste desse modelo foram, respectivamente: 0.8062 e 0.6641

2.2 Exercício 2

Para ajustar os hiperparâmetros, o modelo foi iniciado de acordo com os padrões definidos no enunciado. Em seguida, foram feitos testes com cada um dos *Learning Rates* propostos para as função de ativação *relu* e *tanh*.

ReLU

Learning Rate	0.001	0.01	0.1
Final Training Loss	0.7879	0.3513	0.2166
Final Validation Accuracy	0.8656	0.9389	0.9435
Final Test Accuracy	0.7449	0.8593	0.8701

Tabela 2: Performance por *Learning Rate*, utilizando função de ativação *relu*

Tanh

Learning Rate	0.001	0.01	0.1
Final Training Loss	0.6124	0.4162	0.2808
Final Validation Accuracy	0.8305	0.9128	0.9318
Final Test Accuracy	0.702	0.8289	0.8514

Tabela 3: Performance por *Learning Rate*, utilizando função de ativação *tanh*

Já que a função *relu* apresentou uma performance melhor que a *tanh* para todos os *Learning Rates* testados, foram feitos testes mais extensivos utilizando-a como função de ativação. Nas Tabelas 4, 5 e 6 estão os resultados do modelo utilizando diferentes valores de *Dropout Probability*; variando também o *Learning Rate*.

Learning Rate = 0.001

Dropout Probability	0.3	0.5
Final Training Loss	0.7879	0.863
Final Validation Accuracy	0.8656	0.8601
Final Test Accuracy	0.7449	0.7317

Tabela 4: Performance por *Dropout Probability*, utilizando função de ativação *relu*

Pode-se observar nas Tabelas 4 a 6 que, novamente, a performance com *Learning Rate* igual a 0.1 se manteve superior às demais. Dessa forma, os seguintes testes foram realizados com LR fixo, variando o tamanho das *hidden layers* e a *dropout probability* (Tabelas 7 e 8).

Learning Rate = 0.01

Dropout Probability	0.3	0.5
Final Training Loss	0.3513	0.443
Final Validation Accuracy	0.9389	0.9312
Final Test Accuracy	0.8593	0.8407

Tabela 5: Performance por *Dropout Probability*, utilizando função de ativação *relu***Learning Rate = 0.1**

Dropout Probability	0.3	0.5
Final Training Loss	0.2166	0.3676
Final Validation Accuracy	0.9435	0.9352
Final Test Accuracy	0.8701	0.8472

Tabela 6: Performance por *Dropout Probability*, utilizando função de ativação *relu***Dropout Probability = 0.3**

Hidden Size	100	200
Final Training Loss	0.2166	0.1315
Final Validation Accuracy	0.9435	0.9564
Final Test Accuracy	0.8701	0.8966

Tabela 7: Performance por *Hidden Size*, utilizando função de ativação *relu***Dropout Probability = 0.5**

Hidden Size	100	200
Final Training Loss	0.3676	0.2299
Final Validation Accuracy	0.9352	0.9514
Final Test Accuracy	0.8472	0.8818

Tabela 8: Performance por *Hidden Size*, utilizando função de ativação *relu*

Adicionalmente, foram feitos testes menos extensos utilizando a função de ativação *tanh*, que apresentou menor acurácia. A pesquisa foi feita utilizando seu *Learning Rate* de melhor performance: 0.1, variando *Dropout Probability* e em seguida *Hidden Size*. (Tabelas 9 e 10)

Como esperado, o modelo com função de ativação *tanh* manteve-se com performance inferior ao com *relu*. Assim, a melhor configuração atingida teve os seguintes hiperparâmetros: **Learning Rate = 0.1, Dropout Probability = 0.3, Hidden Size = 200 e Activation Function = ReLU**. Este modelo resultou em uma acurácia de validação igual a 0.9564 e uma acurácia de teste igual a 0.8966.

Learning Rate = 0.1

Dropout Probability	0.3	0.5
Final Training Loss	0.2808	0.419
Final Validation Accuracy	0.9318	0.9056
Final Test Accuracy	0.8514	0.8178

Tabela 9: Performance por *Dropout Probability*, utilizando função de ativação *tanh*

Dropout Probability = 0.3

Dropout Probability	100	200
Final Training Loss	0.2808	0.2007
Final Validation Accuracy	0.9318	0.9463
Final Test Accuracy	0.8514	0.8755

Tabela 10: Performance por *Dropout Probability*, utilizando função de ativação *tanh*

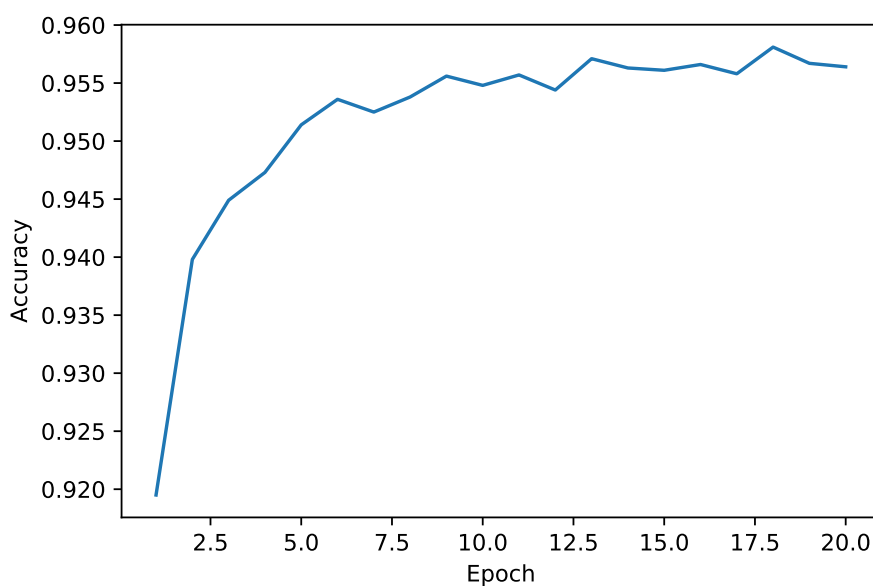


Gráfico 9: *Validation Acuracy* por *epoch* com *Single Layer Feed-Foward Neural Network*

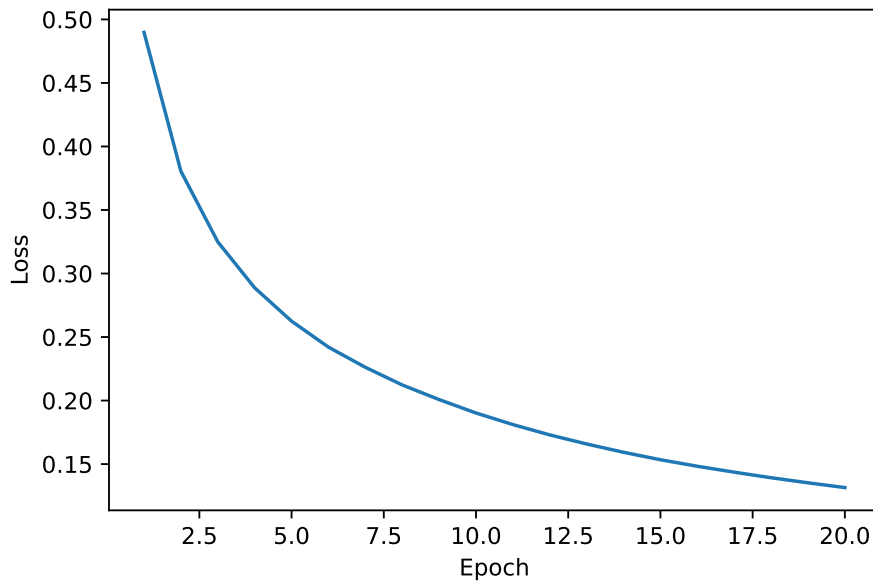


Gráfico 10: *Trainig Loss por epoch com Single Layer Feed-Foward Neural Network*

2.3 Exercício 3

Neste exercício, foram criados dois novos modelos, com os hiperparâmetros *default* do Exercício 2, contendo duas e três *layers*. Entretanto, o melhor resultado manteve-se o do modelo do exercício anterior, com apenas uma *layer*.

A performace dos modelos com 2 e 3 *layers* com os hiperparâmetros *default* do Exercício 2, estão na tabela 11. Adicionalmente foram traçados seus gráficos de *training loss* e *validation accuracy* por *epoch*.

Layers	2	3
Final Training Loss	0.5700	0.9068
Final Validation Accuracy	0.8925	0.8136
Final Test Accuracy	0.7817	0.6816

Tabela 11: Performance por quantidade de *layers*

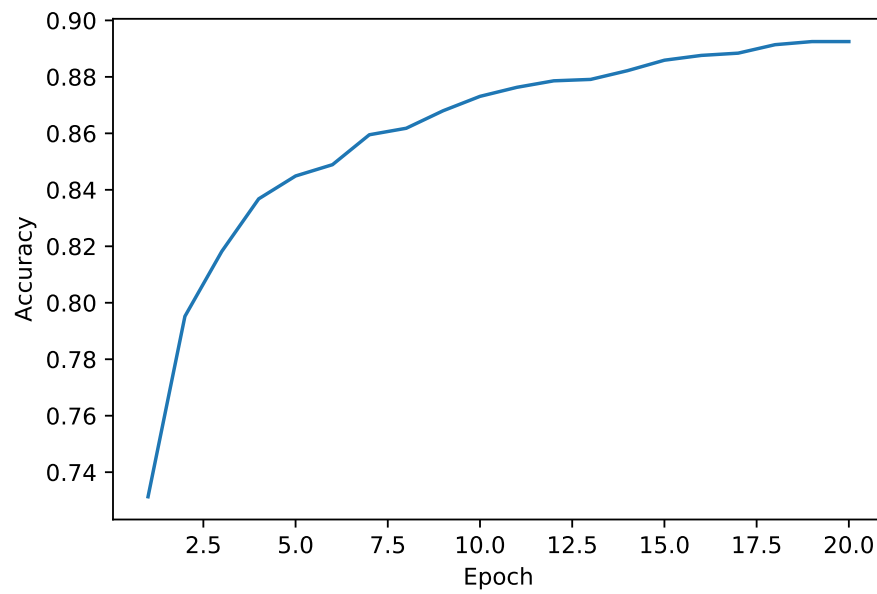


Gráfico 11: *Validation Accuracy* por *epoch* com *Double Layer Feed-Foward Neural Network*

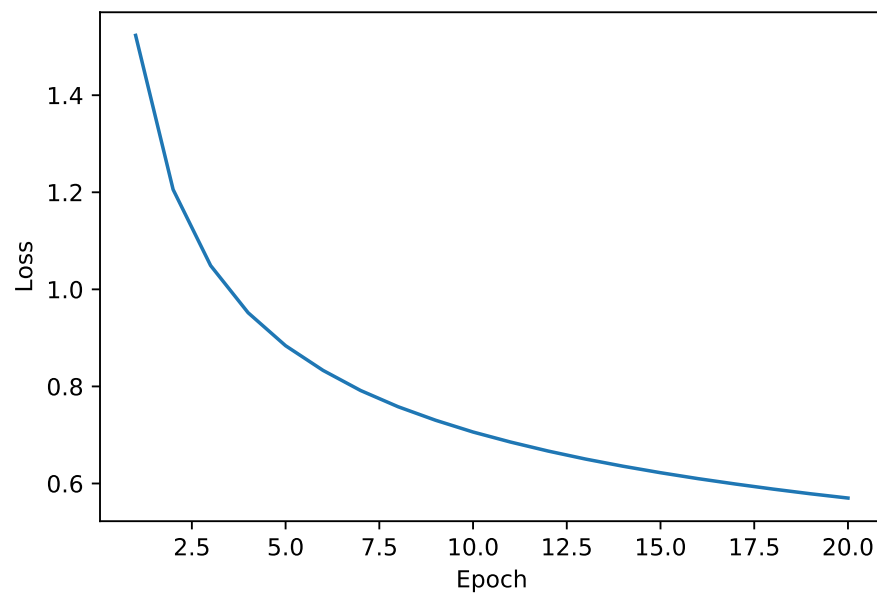


Gráfico 12: *Trainig Loss* por *epoch* com *Double Layer Feed-Foward Neural Network*

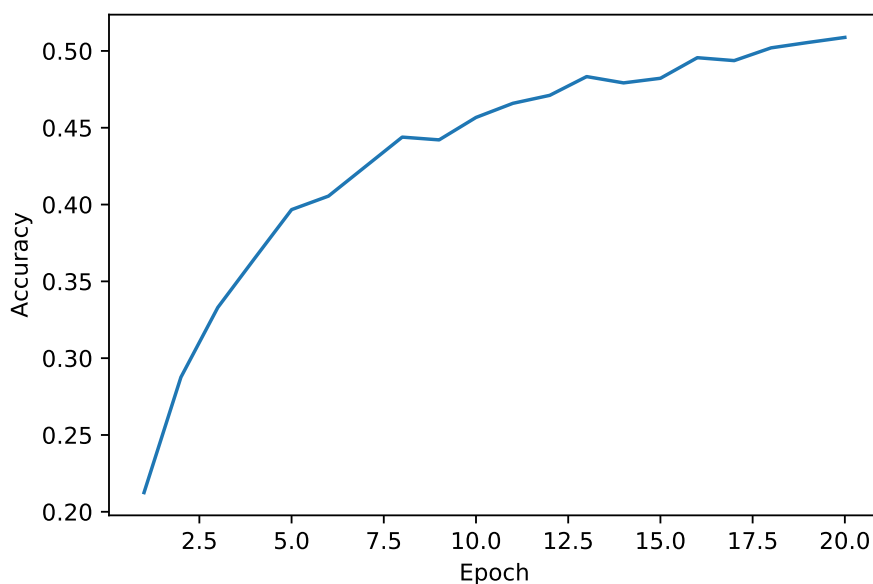


Gráfico 13: *Validation Accuracy* por *epoch* com *Triple Layer Feed-Foward Neural Network*

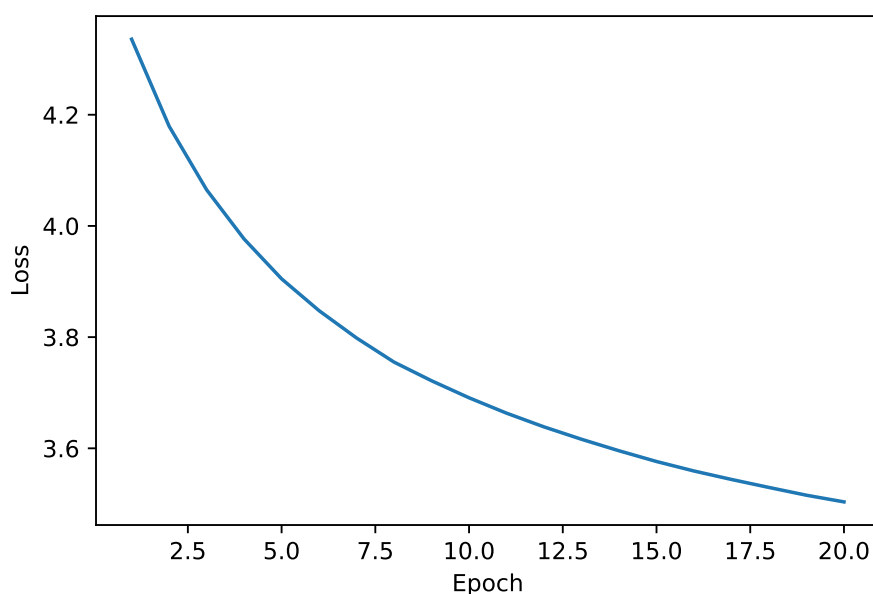


Gráfico 14: *Trainig Loss* por *epoch* com *Triple Layer Feed-Foward Neural Network*

3 Responsabilidades

O projeto foi produzido em conjunto, sendo assim, todo o trabalho foi revisado e feito e/ou auxiliado pelos dois membros. Em mais detalhes, Brian foi responsável, em maior parte, pela primeira questão, enquanto Mariana pela segunda, mas ambos estiveram presente no desenvolvimento de todo o trabalho. Já para a terceira questão, que a dupla considerou a mais difícil, o trabalho foi feito juntos não havendo participação maior ou menor de nenhuma parte. No que se refere ao relatório, Brian foi responsável pela estruturação do documento em

LaTeX, e preenchimento das partes referentes a primeira questão, por fim, Mariana o finalizou preenchendo as partes referentes a segunda e terceira questão.