

# Lab MLP

Paulo Roberto Farah  
Universidade Federal do Paraná (UFPR)  
Programa de Pós-Graduação em Informática (PPGInf)  
INFO 7004 – Aprendizagem de Máquina

## I. INTRODUÇÃO

*Multilayer perceptron (MLP)* é um conjunto de métodos de aprendizado supervisionado que têm como objetivo reconhecer padrões. É uma classe de redes neurais artificiais *feedforward* onde as conexões seguem em um único sentido, sem formar um círculo. Utiliza múltiplos classificadores leves e rápidos e assim, minimiza os erros de classificação.

Nesse contexto, o objetivo desse trabalho é avaliar os efeitos de quatro questões do método MLP. Primeiro, o impacto da quantidade de neurônios na camada escondida. Segundo, quais efeitos o aumento da quantidade de camadas escondidas produz nos resultados do classificador. Em seguida, avaliar diferentes valores de épocas de treinamento e, por fim, discutir qual arquitetura (simples ou complexa) entra em *overfitting* com mais facilidade.

## II. METODOLOGIA

A metodologia para a execução dessa atividade incluiu o planejamento dos experimentos, uma descrição da implementação e as instruções de instalação e execução do programa.

### A. Planejamento

A avaliação foi planejada com os seguintes valores dos parâmetros: **neurônios: 1, 10, 25, 50, 75 e 100, camadas: 1 e 2 e épocas: 1, 10, 50, 100, 150, 200.** Além disso, foi atribuído um valor fixo de *seed* para a rede com o objetivo de isolar o efeito de aleatoriedade do algoritmo.

Foram coletadas as métricas de desempenho de teste *acurácia*, *loss*, *matriz de confusão* e o *histórico* de acurácia de treino e teste.

Os experimentos foram executados em um computador com processador Intel i7 8<sup>th</sup> geração, com 16GB de memória e sistema operacional Windows 10. Foi utilizado python 3.6.6rc1 (64 bits) e IDE PyCharm para o desenvolvimento.

### B. Implementação

O programa `mlp2.py` foi modificado para executar os parâmetros definidos anteriormente, gravar os resultados nos arquivos `neuronios.csv`, `camadas.csv` e `epocas.csv`. Foram acrescentados os métodos `criar_modelo_uma_camada`, `criar_modelo_duas_camadas`, `gerar_resultados`, `plot` e `salvar_arquivo`.

### C. Instalação e Execução

A instalação dos pacotes necessários a execução do classificador pode ser feita com os seguintes comandos, descritos a seguir.

```
unzip mlp2.zip
cd mlp2
virtualenv venv
source venv/bin/activate
pip3 install -r requirements.txt
```

Para executar o programa:

```
python3 mlp2.py <treino> <teste>
```

## III. RESULTADOS

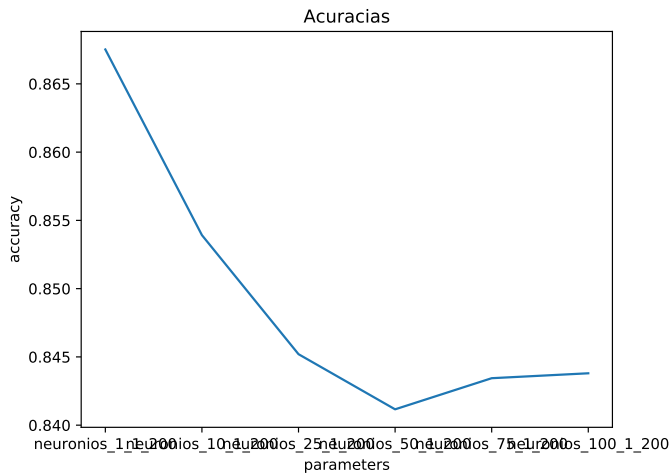
Os resultados são descritos e discutidos nesta seção. São mostrados os resultados da análise de neurônios, camadas, épocas e *overfitting* em relação à arquitetura da rede.

### A. Quantidade de neurônios na camada escondida

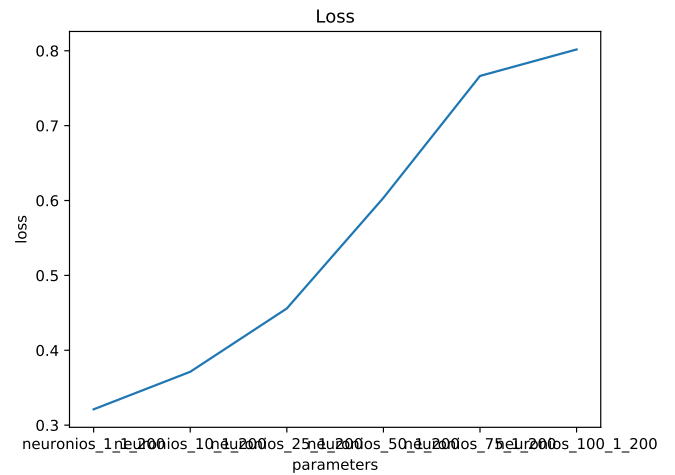
Nesta seção estão descritos os resultados da quantidade de neurônios na camada escondida. A Figura 1 mostra os valores de acurácia e perda (*loss*) para 1, 10, 25, 50, 75 e 100 neurônios com 1 camada escondida e 200 épocas. Pode-se observar uma melhor acurácia para 1 neurônio e a pior acurácia para 50 neurônios. A menor perda foi identificada para 1 neurônio também, mas a maior perda foi para 100 neurônios.

Na Figura 4 é mostrado o histórico das acurácias para os modelos com 1, 10, 25, 50, 75 e 100 neurônios, com 1 camada escondida e 200 épocas. Observa-se pelos resultados que, à medida que as épocas aumentam, a acurácia do treino aumenta e do teste reduz. Esse comportamento indica que, para esse conjunto de dados, o modelo apresenta *overfitting* no caso de 200 épocas. Dos valores analisados, apenas para 1 neurônio a acurácia não reduziu com o processamento das épocas.

Observa-se, a partir da Figura 5, a quantidade de verdadeiros e falsos positivos e negativos para cada um dos seis modelos de neurônios utilizados. Deste modo, caracteriza-se a diagonal principal como os acertos dos respectivos métodos. Nota-se que a quantidade de acertos na diagonal principal é maior para 25 neurônios para TP, com valor de 0,89 e para 1 e 50 neurônios para TF, com 0,86.

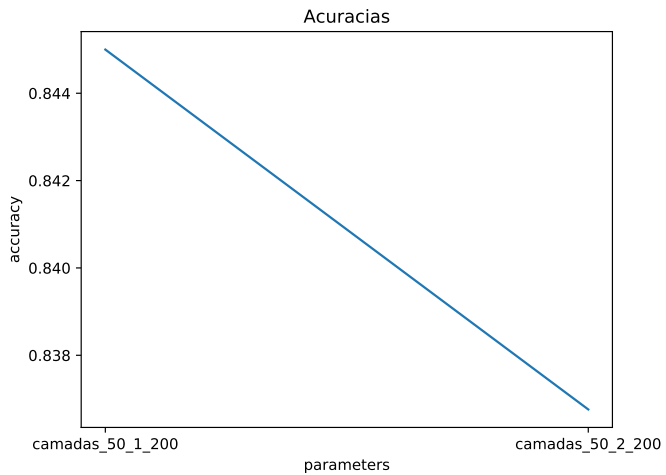


(a) Acurácia.

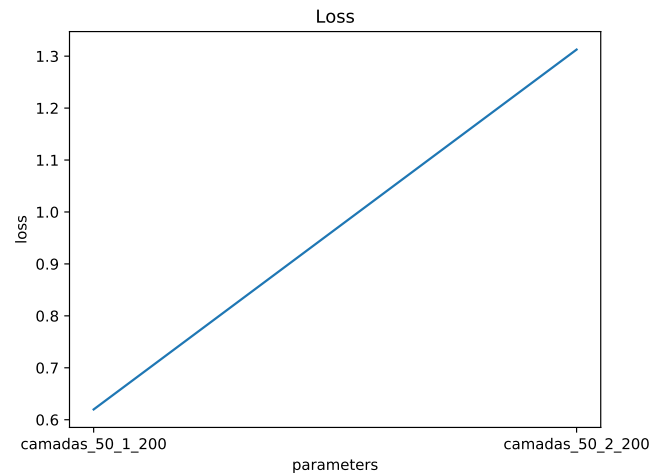


(b) Perda (Loss).

Figura 1: Acurácia e Loss dos valores de neurônios 1, 10, 25, 50, 75, 100 com 1 camada escondida e 200 épocas



(a) Acurácia.



(b) Perda (loss).

Figura 2: Acurácia e Loss dos valores de 1 e 2 camadas escondidas com 50 neurônios e 200 épocas.

### B. Quantidade de camadas escondidas

Nesta seção estão descritos os resultados da quantidade de camadas escondidas da rede neural. A Figura 2 mostra os valores de acurácia e perda (*loss*) para 1 e 2 camadas escondidas, 50 neurônios e 200 épocas. Pode-se observar uma melhor acurácia e menor perda para 1 camada.

Na Figura 6 é mostrado o histórico das acurácias para os modelos com 1 e duas camadas escondidas, 50 neurônios e 200 épocas. Observa-se para ambas que, à medida que as épocas aumentam, a acurácia do treino aumenta significativamente e do teste reduz no final. Esse comportamento indica que, para esse conjunto de dados, o modelo apresenta overfitting no caso de 200 épocas. O modelo com apenas 1 camada apresenta uma variabilidade menor do que o resultado com 2 camadas.

Observa-se, a partir da Figura 7, a quantidade de verdadeiros e falsos positivos e negativos para cada um dos seis modelos de neurônios utilizados. Deste modo, caracteriza-se a diagonal principal como os acertos dos respectivos métodos. Nota-se que a quantidade de acertos na diagonal principal é maior para 1 camada para TP, com valor de 0,85 e ocorre um empate para TF, com 0,84.

### C. Número de épocas de treinamento

Nesta seção estão descritos os resultados da quantidade de épocas de treinamento.

A Figura 3 mostram os valores de acurácia e perda (*loss*) para 1, 10, 50, 100, 150 e 200 épocas com 1 camada escondida e 50 neurônios. Pode-se observar uma melhor acurácia para

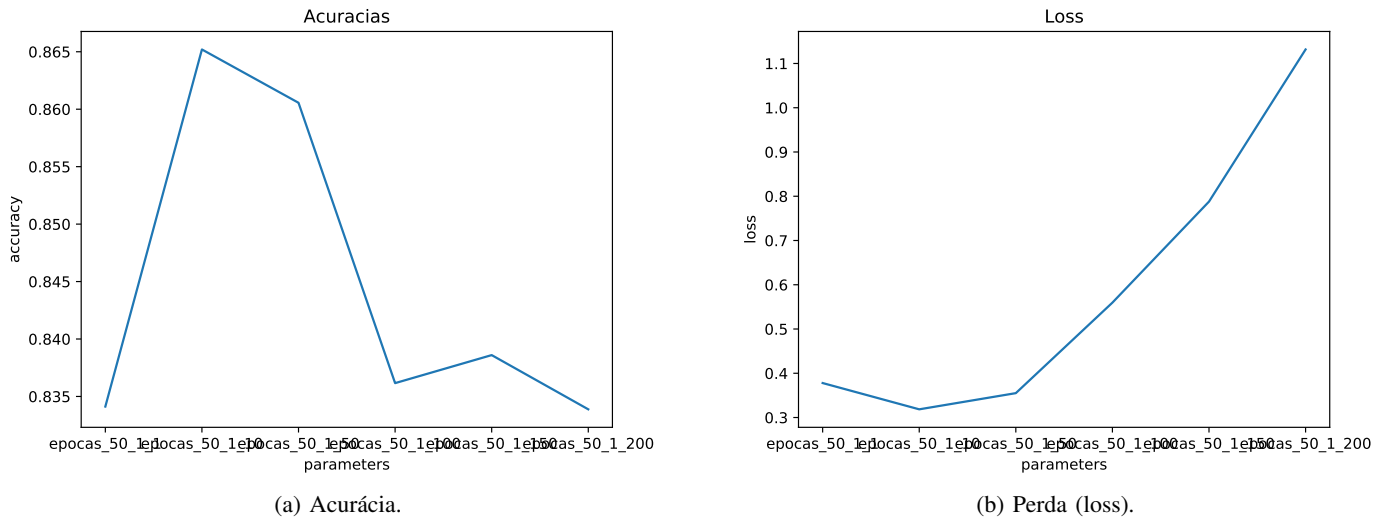


Figura 3: Acurácia e Loss dos valores de épocas 1, 10, 50, 100, 150 e 200 com 1 camada escondida e 50 neurônios.

10 épocas e a pior acurácia para 200 épocas. A menor perda foi identificada para 10 épocas também, e a pior perda foi para 200 neurônios.

Na Figura 8 é mostrado o histórico das acurácias para os modelos com 1, 10, 50, 100, 150 e 200 épocas, com 1 camada escondida e 50 neurônios. Com exceção dos resultados com 10 épocas, observa-se pelos que, à medida que as épocas aumentam, a acurácia do treino aumenta e do teste reduz. Esse comportamento indica que, para esse conjunto de dados, o modelo apresenta overfitting em quase todos os casos. Para 10 épocas, o modelo apresenta um resultado com melhor generalização, pois se aproxima do comportamento da base de aprendizagem.

Observa-se, a partir da Figura ??, a quantidade de verdadeiros e falsos positivos e negativos para cada um dos seis modelos de épocas utilizados. Nota-se que a quantidade de acertos na diagonal principal é maior para 10 e 50 épocas para TP, com valor de 0,89 e para 100 épocas para TF, com 0,91.

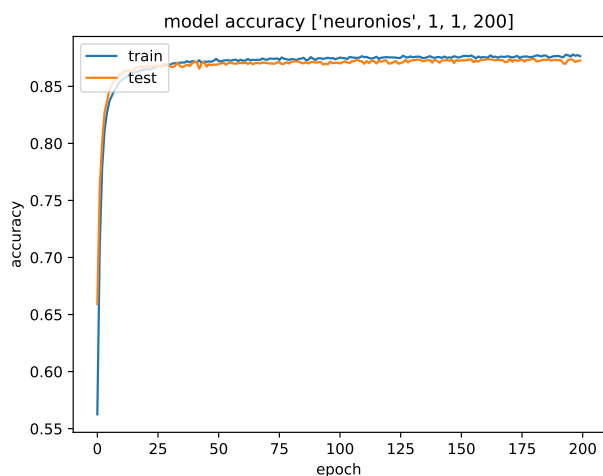
#### D. Overfitting da rede

De acordo com os resultados apresentados anteriormente, uma arquitetura mais simples, ou seja, com poucos neurônios, 1 camada escondida e 10 épocas apresenta maior capacidade de generalização. As arquiteturas com mais neurônios, mais camadas e mais épocas apresentaram uma redução em sua acurácia e maior perda para dados de teste. Em contrapartida, a base de aprendizagem apresentou uma melhora muito substancial, indicando assim overfitting.

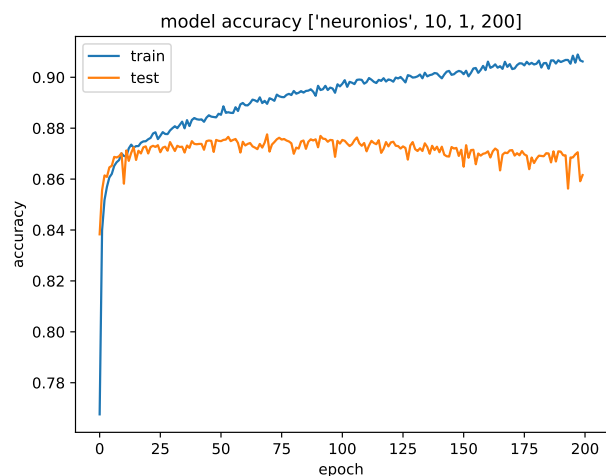
#### IV. CONCLUSÃO

O presente trabalho teve como objetivo principal a avaliação do desempenho dos parâmetros neurônios, camadas e épocas do modelo MLP. Foi avaliado o desempenho desses parâmetros com a base IMDB e os resultados indicaram que parâmetros menores, com 1 a 10 neurônios, 1 camada escondida e 10

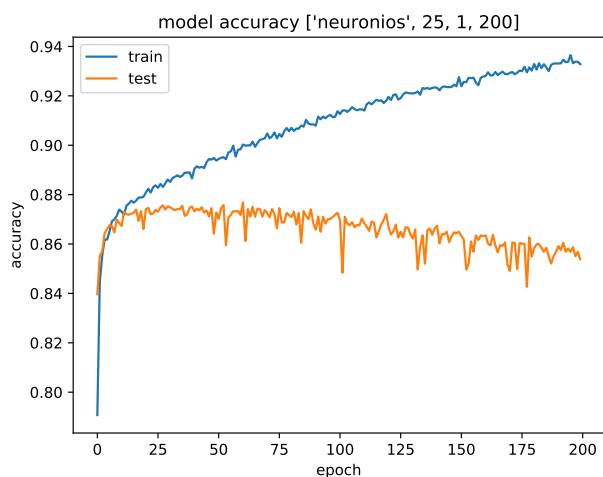
épocas apresentaram resultados melhores do que parâmetros com mais neurônios, camadas e épocas.



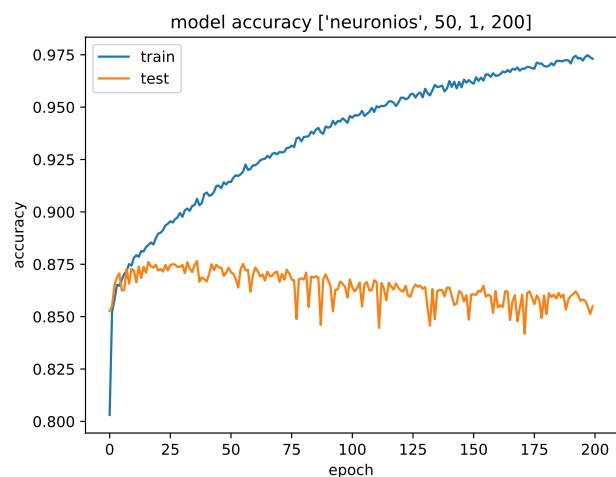
(a) 1 neurônio.



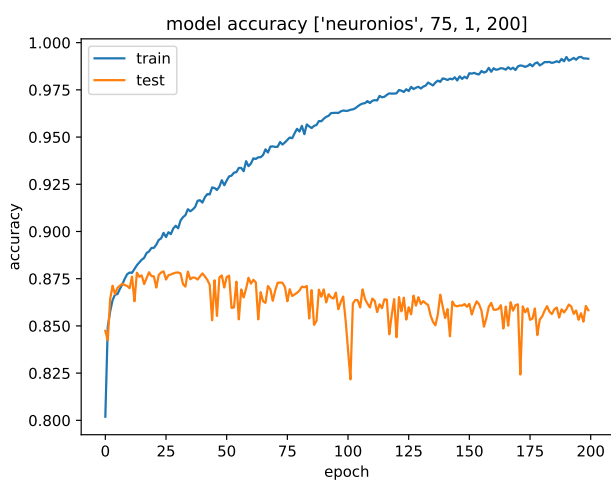
(b) 10 neurônios.



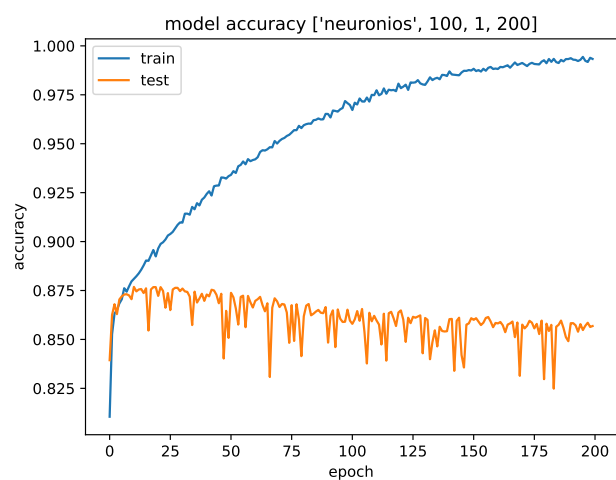
(c) 25 neurônios.



(d) 50 neurônios.

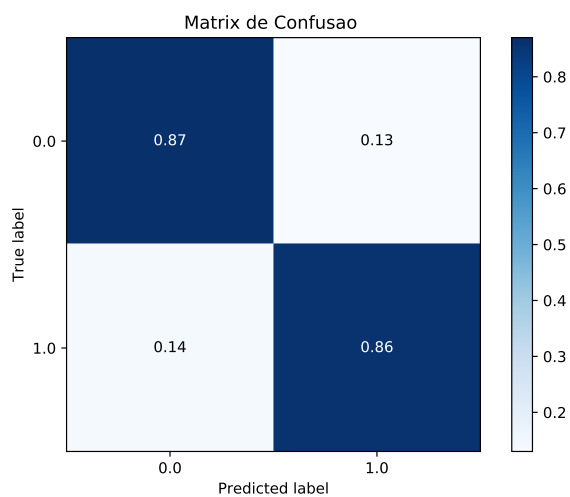


(e) 75 neurônios.

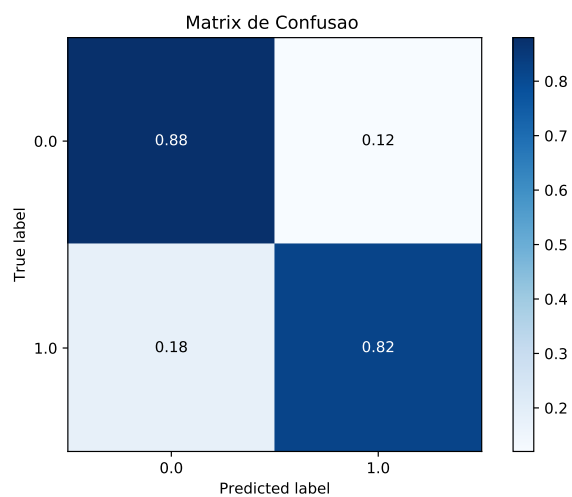


(f) 100 neurônios.

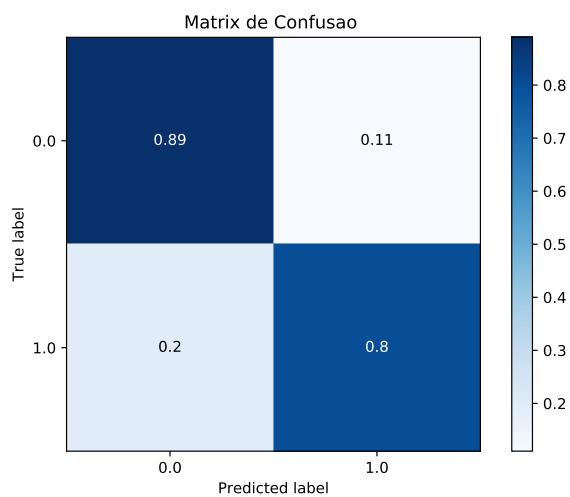
Figura 4: Histórico de acurácia dos modelos dos valores de neurônios 1, 10, 25, 50, 75, 100 com 1 camada escondida e 200 épocas.



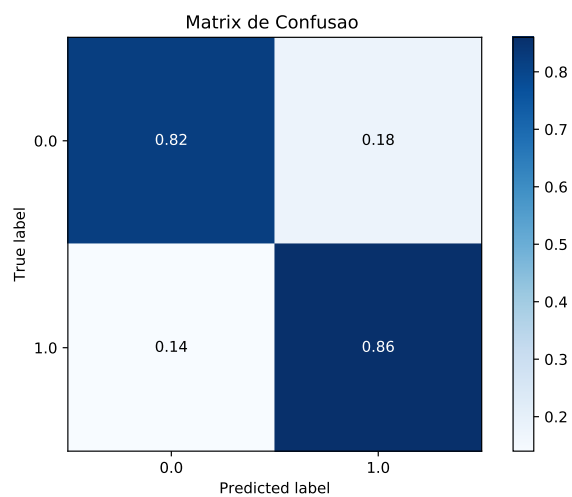
(a) 1 neurônio.



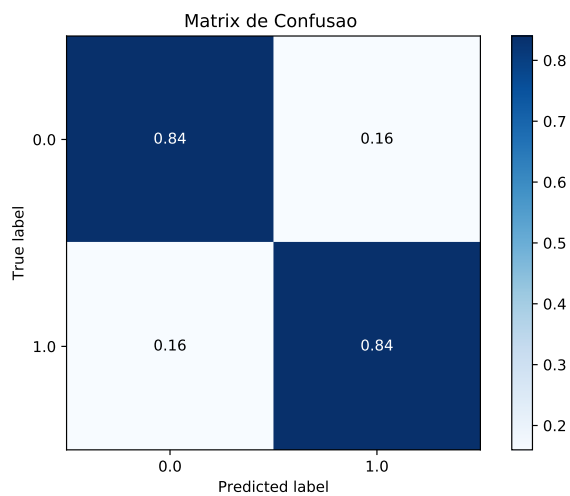
(b) 10 neurônios.



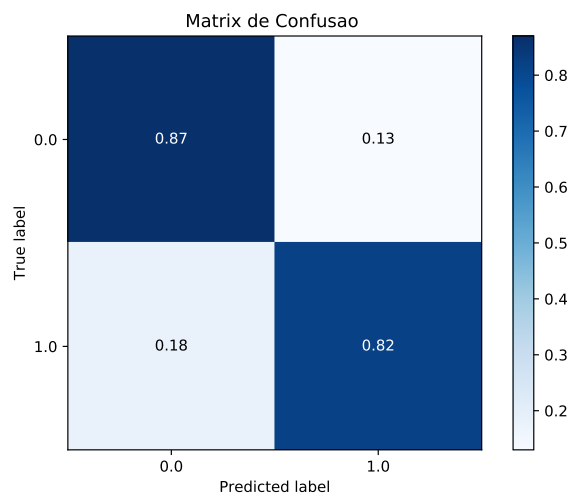
(c) 25 neurônios.



(d) 50 neurônios.

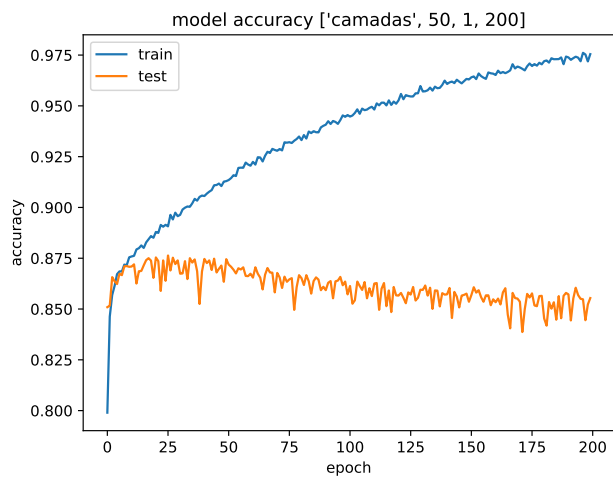


(e) 75 neurônios.

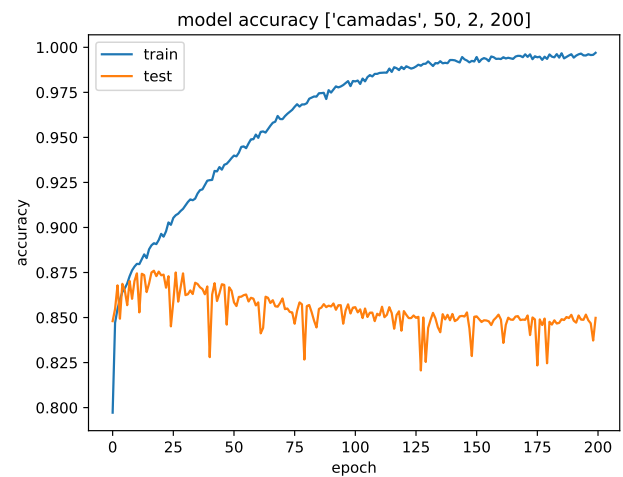


(f) 100 neurônios.

Figura 5: Matriz de confusão dos modelos dos valores de neurônios 1, 10, 25, 50, 75, 100 com 1 camada escondida e 200 épocas.

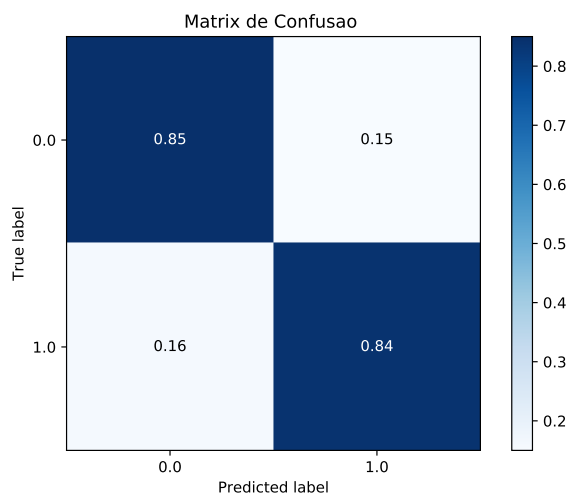


(a) 1 camada.

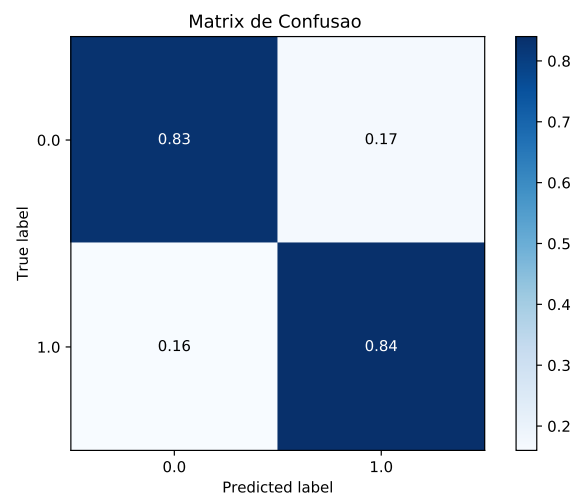


(b) 2 camadas.

Figura 6: Histórico de acurácia dos modelos dos valores de camadas escondidas 1 e 2, com 50 neurônios e 200 épocas.

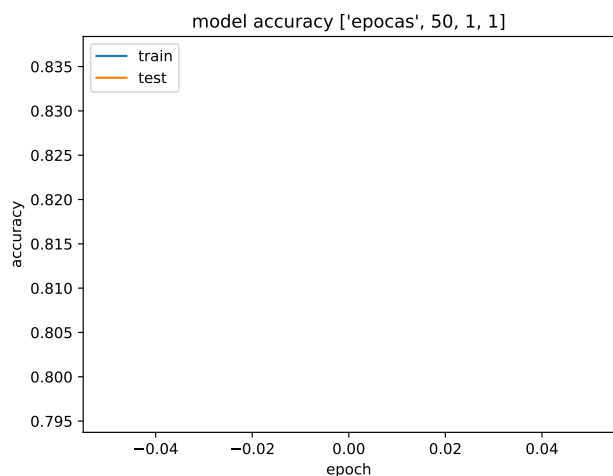


(a) 1 camada.

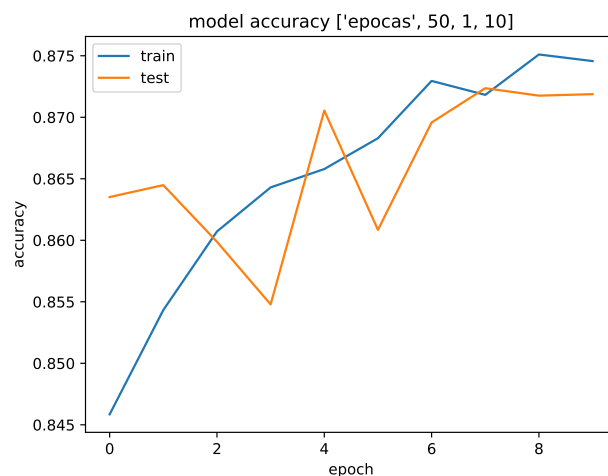


(b) 2 camadas.

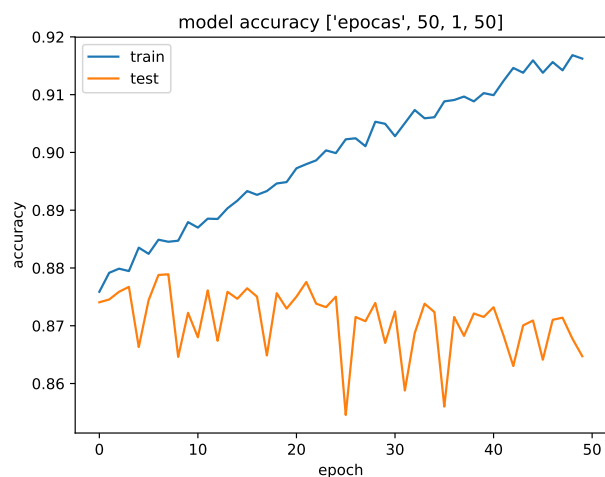
Figura 7: Matriz de confusão dos modelos dos valores de 50 neurônios com 1 e 2 camadas escondidas e 200 épocas.



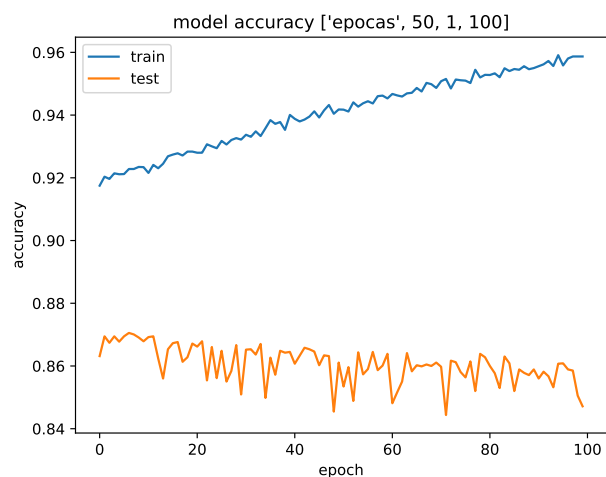
(a) 1 época.



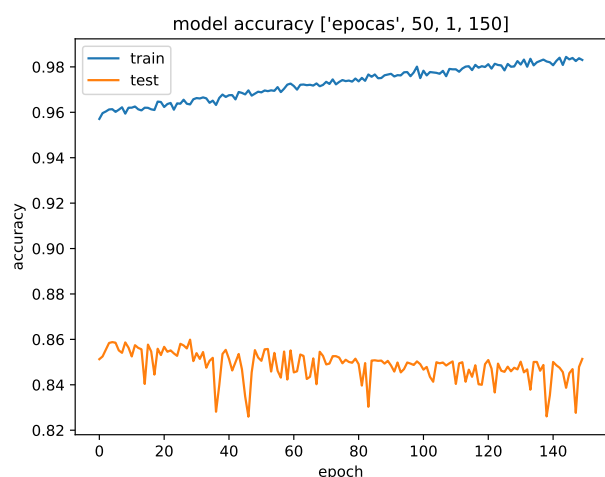
(b) 10 épocas.



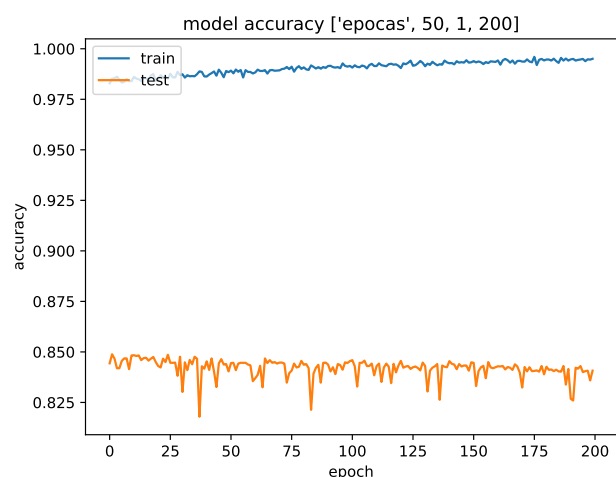
(c) 50 épocas.



(d) 100 épocas.

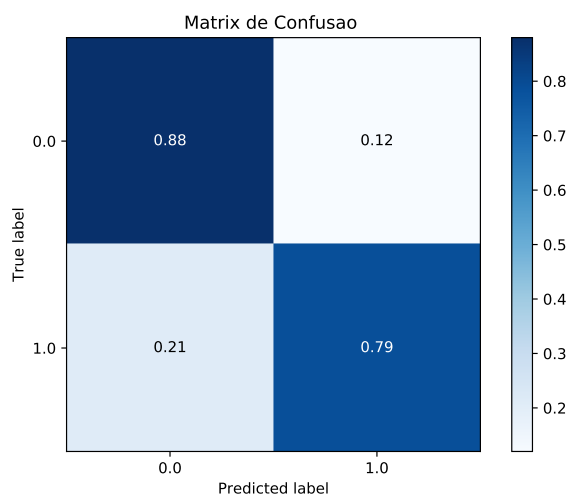


(e) 150 épocas.

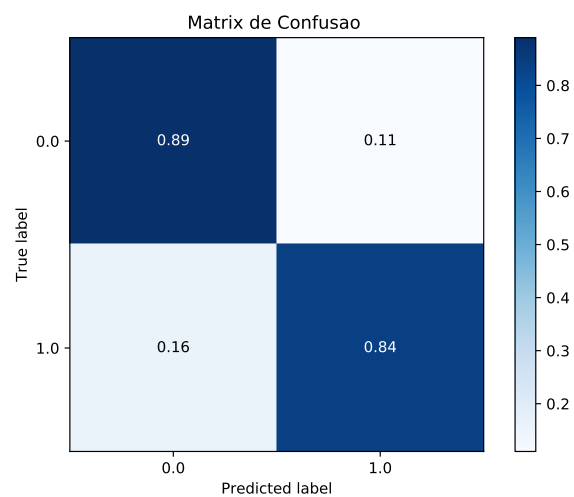


(f) 200 épocas.

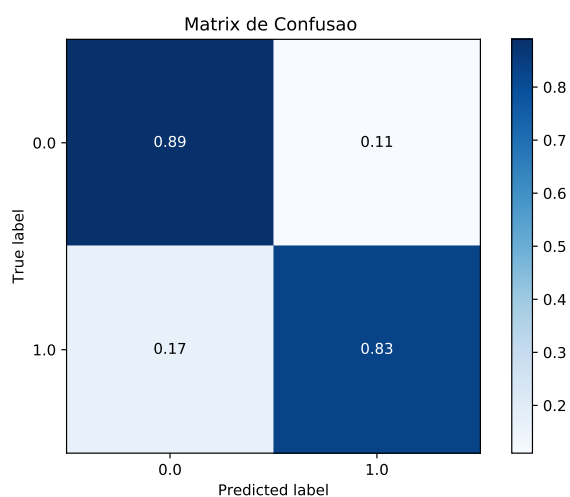
Figura 8: Histórico de acurácia dos modelos dos valores de épocas 1, 10, 50, 100, 150, 200 com 1 camada escondida e 50 neurônios.



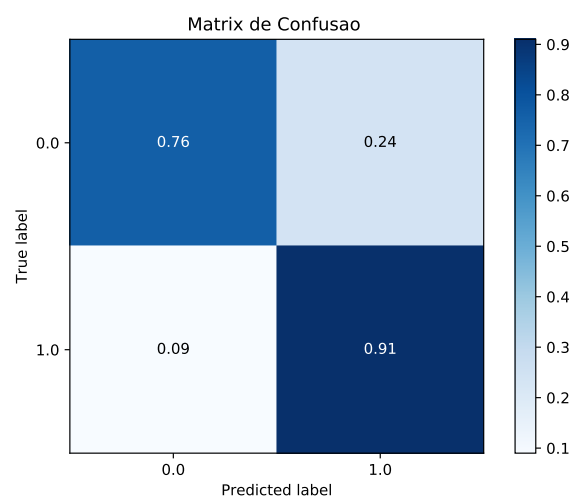
(a) 1 época.



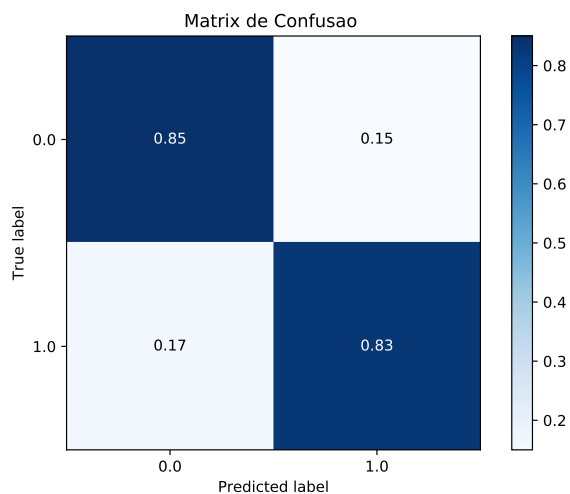
(b) 10 épocas.



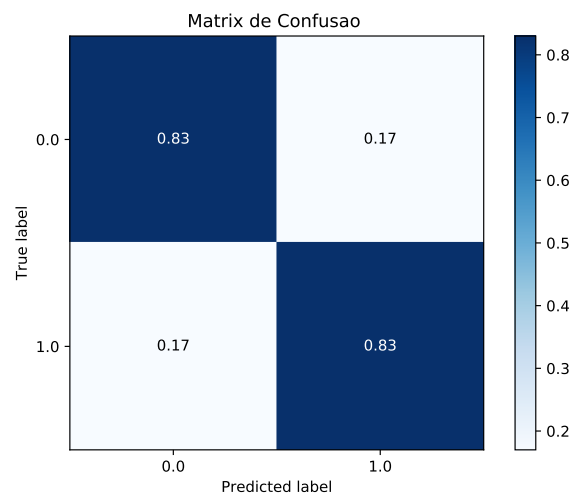
(c) 50 épocas.



(d) 100 épocas.



(e) 150 épocas.



(f) 200 épocas.

Figura 9: Matriz de confusão dos modelos dos valores de épocas 1, 10, 50, 100, 150, 200 com 1 camada escondida e 50 neurônios.