

Exercício 10

Arthur Felipe Reis Souza

June 2, 2024

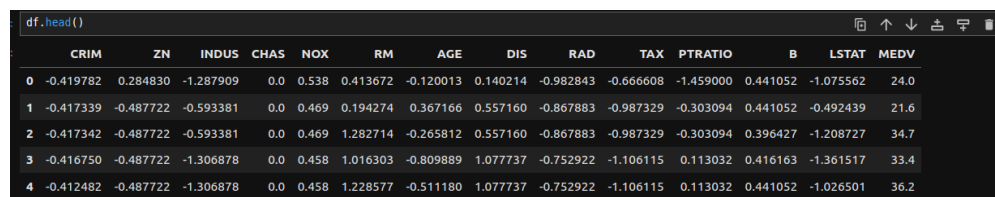
1 Introduction

O exercício semanal consiste em utilizar diferentes arquiteturas das redes MLP, de forma a resolver um problema de regressão e um problema de classificação. O dataset de classificação é o dataset Statlog, que consiste em um conjunto de características de um indivíduo que serão utilizadas para classificar a presença ou não presença de uma doença cardíaca. O dataset de regressão é o BostonHouse, que terá suas características utilizadas para estimar o valor médio de um imóvel em Boston.

A critério de comparação, no problema de classificação será utilizado dois algoritmos de Machine Learning para comparar o desempenho da rede MLP. Um será o classificador de Bayes e o outro o KNN.

2 Desenvolvimento

A primeira base de dados a ser utilizado é o BostonHouse, que inicialmente foi tratada usando o StandardScaler da biblioteca scikitlearn, normalizando todas as características do dataset. Após a normalização, foi utilizado o Grid-Search para encontrar o número de neurônios intermediários que levam ao modelo a uma melhor performance.



	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT	MEDV
0	-0.419782	0.284830	-1.287909	0.0	0.538	0.413672	-0.120013	0.140214	-0.982843	-0.666608	-1.459000	0.441052	-1.075562	24.0
1	-0.417339	-0.487722	-0.593381	0.0	0.469	0.194274	0.367166	0.557160	-0.867883	-0.987329	-0.303094	0.441052	-0.492439	21.6
2	-0.417342	-0.487722	-0.593381	0.0	0.469	1.282714	-0.265812	0.557160	-0.867883	-0.987329	-0.303094	0.396427	-1.208727	34.7
3	-0.416750	-0.487722	-1.306878	0.0	0.458	1.016303	-0.809889	1.077737	-0.752922	-1.106115	0.113032	0.416163	-1.361517	33.4
4	-0.412482	-0.487722	-1.306878	0.0	0.458	1.228577	-0.511180	1.077737	-0.752922	-1.106115	0.113032	0.441052	-1.026501	36.2

Figure 1: Dados normalizados BostonHouse.

O mesmo foi realizado para o dataset Statlog.

	age	sex	chest-pain	rest-bp	serum-chol	fasting-blood-sugar	electrocardiographic	max-heart-rate	angina	oldpeak	slope	major-vessels	thal
0	0.854167	1.0	1.000000	0.339623	0.447489	0.0	1.0	0.290076	0.0	0.387097	0.5	1.000000	0.00
1	0.791667	0.0	0.666667	0.198113	1.000000	0.0	1.0	0.679389	0.0	0.258065	0.5	0.000000	1.00
2	0.583333	1.0	0.333333	0.283019	0.308219	0.0	0.0	0.534351	0.0	0.048387	0.0	0.000000	1.00
3	0.729167	1.0	1.000000	0.320755	0.312785	0.0	0.0	0.259542	1.0	0.032258	0.5	0.333333	1.00
4	0.937500	0.0	0.333333	0.245283	0.326484	0.0	1.0	0.381679	1.0	0.032258	0.0	0.333333	0.00
...
265	0.479167	1.0	0.666667	0.735849	0.166667	1.0	0.0	0.694656	0.0	0.080645	0.0	0.000000	1.00
266	0.312500	1.0	0.333333	0.245283	0.312785	0.0	0.0	0.778626	0.0	0.000000	0.0	0.000000	1.00
267	0.562500	0.0	0.333333	0.433962	0.383562	0.0	1.0	0.625954	0.0	0.209677	0.5	0.000000	0.00
268	0.583333	1.0	1.000000	0.433962	0.150685	0.0	0.0	0.587786	0.0	0.064516	0.5	0.000000	0.75
269	0.791667	1.0	1.000000	0.622642	0.365297	0.0	1.0	0.282443	1.0	0.241935	0.5	1.000000	0.00

270 rows x 13 columns

Figure 2: Dados normalizados Statlog.

3 Resultados

3.1 Boston House

Os resultados para o dataset de regressão boston foram :

```
import statistics
print(f"The MSE for the test values is {np.mean(mse_scores)} +- {statistics.stdev(mse_scores)}")
The MSE for the test values is 285.2545589384348 +- 51.55878612102186
```

Figure 3: Resultado usando 5 neuronios na camada intermediária com o otimizador ADAM.

```
import statistics
print(f"The MSE for the test values is {np.mean(mse_scores)} +- {statistics.stdev(mse_scores)}")
The MSE for the test values is 36.78813105168576 +- 0.5946961898929474
```

Figure 4: Resultado usando 40 neuronios na camada intermediária com o otimizador ADAM.

```
import statistics

print(f"The MSE for the test values is {np.mean(mse_scores)} +- {statistics.stdev(mse_scores)}")

The MSE for the test values is 26.661896786760252 +- 1.0451228013004243
```

Figure 5: Resultado usando 10 neurônios na camada intermediária com o otimizador Gradiente Estocástico e a técnica de early stopping.

3.2 Statlog

No dataset Statlog, por ser de classificação, também será avaliado pela métrica AUC (Area Under the Curve). Essa métrica tem por objetivo avaliar o desempenho de um classificador variando os valores de threshold. Quanto mais próximo de 1, melhor é o desempenho do mesmo. No exercício, como dito anteriormente, foram utilizados 3 diferentes classificador, o classificador de Bayes, o KNN e uma rede MLP. Os resultados do desempenho estão mostrados abaixo :

```
[13]: print(f"The accuracy for knn is {np.mean(result_knn)} +- {statistics.stdev(result_knn)}")
      print(f"The accuracy for mlp is {np.mean(result_mlp)} +- {statistics.stdev(result_mlp)}")
      print(f"The accuracy for bayesian classifier is {np.mean(result_bay)} +- {statistics.stdev(result_bay)}")

The accuracy for knn is 0.8 +- 0.006295085819250355
The accuracy for mlp is 0.8235155159139104 +- 0.007817847011045699
The accuracy for bayesian classifier is 0.8414814814814815 +- 0.005465665091649087

[14]: X_train, X_test, y_train, y_test = train_test_split(X, y, train_size=0.8, random_state=42)
      y_pred_knn = neigh.fit(X_train, y_train).predict(X_test)
      y_pred_mlp = mlp.fit(X_train, y_train).predict(X_test)
      y_pred_bayes = bay.fit(X_train, y_train).predict(X_test)

[15]: from sklearn.metrics import roc_auc_score

      print(f"AUC score KNN : {roc_auc_score(y_test, y_pred_knn)}")
      print(f"AUC score MLP : {roc_auc_score(y_test, y_pred_mlp)}")
      print(f"AUC score BAYES : {roc_auc_score(y_test, y_pred_bayes)}")

AUC score KNN : 0.7184065934065934
AUC score MLP : 0.7925824175824177
AUC score BAYES : 0.8296703296703297
```

Figure 6: Resultado usando 100 neurônios na camada intermediária com o otimizador ADAM, KNN com k=5, e o classificador de bayes.

```
[37]: print(f"The accuracy for knn is {np.mean(result_knn)} +- {statistics.stdev(result_knn)}")
      print(f"The accuracy for mlp is {np.mean(result_mlp)} +- {statistics.stdev(result_mlp)}")
      print(f"The accuracy for bayesian classifier is {np.mean(result_bay)} +- {statistics.stdev(result_bay)}")

The accuracy for knn is 0.8155555555555555 +- 0.008151888404552674
The accuracy for mlp is 0.8333333333333333 +- 0.012345679012345694
The accuracy for bayesian classifier is 0.8414814814814815 +- 0.005465665091649087

[38]: X_train, X_test, y_train, y_test = train_test_split(X, y, train_size=0.8, random_state=42)
      y_pred_knn = neigh.fit(X_train, y_train).predict(X_test)
      y_pred_mlp = mlp.fit(X_train, y_train).predict(X_test)
      y_pred_bayes = bay.fit(X_train, y_train).predict(X_test)

[39]: from sklearn.metrics import roc_auc_score

      print(f"AUC score KNN : {roc_auc_score(y_test, y_pred_knn)}")
      print(f"AUC score MLP : {roc_auc_score(y_test, y_pred_mlp)}")
      print(f"AUC score BAYES : {roc_auc_score(y_test, y_pred_bayes)}")

AUC score KNN : 0.7554945054945056
AUC score MLP : 0.6978021978021978
AUC score BAYES : 0.8296703296703297
```

Figure 7: Resultado usando 10 neurônios na camada intermediária com o otimizador ADAM, KNN com k=15, e o classificador de bayes.

```

[61]: print(f"The accuracy for knn is {np.mean(result_knn)} +- {statistics.stdev(result_knn)}")
      print(f"The accuracy for mlp is {np.mean(result_mlp)} +- {statistics.stdev(result_mlp)}")
      print(f"The accuracy for bayesian classifier is {np.mean(result_bay)} +- {statistics.stdev(result_bay)}")

The accuracy for knn is 0.8266666666666665 +- 0.004552861460391586
The accuracy for mlp is 0.7666666666666666 +- 0.036330713585091775
The accuracy for bayesian classifier is 0.8414814814815 +- 0.005465665091649887

[62]: X_train, X_test, y_train, y_test = train_test_split(X, y, train_size=0.8, random_state=42)
      y_pred_knn = neigh.fit(X_train, y_train).predict(X_test)
      y_pred_mlp = mlp.fit(X_train, y_train).predict(X_test)
      y_pred_bayes = bay.fit(X_train, y_train).predict(X_test)

[63]: from sklearn.metrics import roc_auc_score

      print(f"AUC score KNN : {roc_auc_score(y_test, y_pred_knn)}")
      print(f"AUC score MLP : {roc_auc_score(y_test, y_pred_mlp)}")
      print(f"AUC score BAYES : {roc_auc_score(y_test, y_pred_bayes)}")

AUC score KNN : 0.7733516483518484
AUC score MLP : 0.7913736283736284
AUC score BAYES : 0.8296703296703297

```

Figure 8: Resultado usando 4 neurônios na camada intermediária com o otimizador ADAM, KNN com $k=25$, e o classificador de bayes.

4 Conclusão

Portanto, ao concluir com êxito o exercício semanal no dataset de regressão e de classificação, é possível afirmar que : No dataset Boston Housing, utilizando o mesmo otimizador, quanto maior o número de neurônios, menor o erro médio. Porém, ao alterar o otimizador para o gradiente estocástico e também utilizando a técnica de early stopping, o desempenho melhorou consideravelmente. No dataset Statlog, o classificador de bayes teve um desempenho relativamente melhor do que os outros dois algoritmos, mesmo realizando algumas variações nos hiperparâmetros do KNN e no MLP. O classificador de bayes é relativamente mais simples do que os outros dois algoritmos, e também teve uma melhor performance para essa base de dados.