

Relatório EP5

Isabela Blucher - 9298170

18 de Junho de 2019

Implementação

Montagem do dataset

O dataset usado neste EP é composto por exemplos do dataset do MNIST que pertencem às classes 0, 1, 2, 3 e 4. Para selecionar os exemplos para o dataset de treinamento + validação, foram selecionados todos os exemplos que pertencem a uma classe específica, e destes foram selecionado 500 de forma aleatória, usando funções do `numpy`.

Após a seleção dos exemplos usados no dataset, as imagens foram redimensionadas de vetores bidimensionais de 28×28 para um vetor unidimensional com 784 elementos. Os elementos desse vetor unidimensional foram normalizados para o intervalo $[0, 1]$ pois foram divididos por 255, o valor máximo possível para os pixels.

5-fold cross-validation

Com o dataset montado, foi possível implementar o 5-fold cross-validation. Como os dados foram previamente embaralhados, os folds foram separados pegando quintos sucessivos do dataset de treinamento e colocando eles em uma lista.

Então um loop que itera 5 vezes (o número de folds definido) seleciona um fold específico baseado em seu índice na lista mencionada para ser o dataset de teste da validação cruzada, enquanto todos os outros folds são usados como o dataset de treino da validação cruzada.

Modelos

Os modelos utilizados são implementações disponíveis no `scikit-learn`. Após a separação dos dados, os modelos são treinados com os dados de treino da validação cruzada e depois usados nos dados de teste do fold para obter um valor para a acurácia daquele fold. A acurácia calculada é inserida em lista, que é usada para calcular o valor da cross-validation accuracy de cada modelo.

Informações relevantes

- Shape de `X_train` e `Y_train`: (2500, 784) e (2500, 1)
- Shape de `X_test` e `Y_test`: (5139, 784) e (5139, 1)
- Valor mínimo e máximo de `X_train` e `X_test`: 0 e 1, após a normalização dos pixels
- Valor mínimo e máximo de `Y_train` e `Y_test`: 0 e 4, que representam as classes dos exemplos
- Número de exemplos de cada classe e acurácia obtida: como os folds são montados aleatoriamente, para cada execução cada classe terá uma quantidade e uma acurácia diferentes. Um exemplo de uma execução pode ser encontrado na Tabela 1.
- Cross-validation accuracy: para uma execução os valores obtidos foram 0.9768 (SVM) e 0.9684 (MLP).
- Características do conjunto de teste: o número de exemplos de cada classe no conjunto de teste é [980, 1135, 1032, 1010, 982].
- Acurácia do algoritmo escolhido em respeito ao conjunto de teste: 0.9863786728935591

Tabela 1: Resultados validação cruzada

	Número de exemplos	Acurácia SVM	Acurácia MLP
0	[105, 104, 101, 112, 78]	0.978	0.968
1	[86, 92, 111, 106, 105]	0.976	0.964
2	[109, 112, 91, 100, 88]	0.982	0.978
3	[105, 99, 89, 92, 115]	0.966	0.96
4	[95, 93, 108, 90, 114]	0.982	0.972

Desempenho final

Ao executar o cross-validation na SVM com parâmetros $\gamma = 0.05$ e $C = 5$ (valores de referência) e no Multi-Layer Perceptron com duas camadas ocultas de 100 nós cada e um learning rate inicial de 0.001 o melhor resultado foi obtido com a SVM.

A acurácia final obtida no conjunto de teste foi 0.9863786728935591. A matriz de confusão resultante está na Figura 1. A figura foi plotada usando um script disponível na documentação do `scikit-learn` [1].

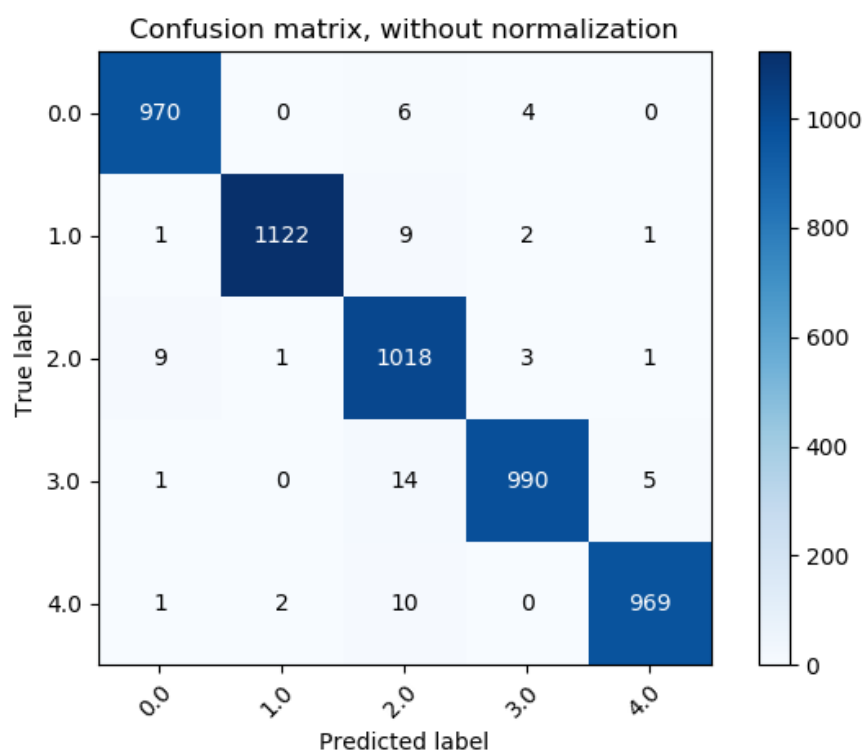


Figura 1: Matriz de confusão para a SVM sobre o conjunto de teste

Referências

- [1] https://scikit-learn.org/stable/auto_examples/model_selection/plot_confusion_matrix.html#sphx-glr-auto-examples-model-selection-plot-confusion-matrix-py