**Fundamentals of Machine Learning (Fall 2022)**

**Homework #3 (100 Pts, Due Date: Oct 30)**

**Student ID 2017314712**

**Name 김우진**

**Instruction:** We provide all codes and datasets in Python. Please write your code to complete the models, and submit your compressed file as ‘HW3\_STUDENTID\_YOURNAME.zip,’ including all the codes except for the ‘data’ directory and your document.

NOTE 1. You should install Pytorch. Please refer to [Start Locally | PyTorch](https://pytorch.org/get-started/locally/)

NOTE 2: You should write your codes in ‘EDIT HERE.’ It is not recommended to edit other parts.

1. Given a cubic polynomial function as the ground-truth function, implement the polynomial regression model in ‘PolynomialRegression.py.’

**(a) [10 pts] [Forward & Polynomial Features]** Implement **Forward** and **PolynomialFeatures** functions in ‘model/PolynomialRegression.py’ using the MSE loss. Note that the bias should be added to the input.

Answer: Fill in your code here. You also have to submit your code to i-campus.

|  |
| --- |
| self.optimizer.zero\_grad()                  n, \_ = batch\_x.shape                  mse\_loss = 1/(2\*n) \* ((batch\_y - pred) \*\* 2).sum()                  mse\_loss.backward()                  self.optimizer.step()  row, col = x.shape          features = np.zeros((row,self.degree + 1))            for i in range(0,row):              for j in range(0,self.degree + 1):                  features[i][j] = np.power(x[i],j)          features = features.astype(np.float32) |

**(b) [20 pts] [Overfitting]** Observe how the root mean squared error (RMSE) is changed in the training and test dataset by increasing the degree of the polynomial regression model and briefly explain the result.

|  |  |  |
| --- | --- | --- |
| **Degree** | **RMSE (Train)** | **RMSE (Test)** |
| **1** | 12.4416 | 17.1765 |
| **3** | 2.1514 | 2.7116 |
| **5** | 1.7499 | 2.7392 |

|  |
| --- |
| Degree 가 높아질수록 RMSE가 줄어드는 것을 확인할 수 있다. 하지만 너무 높아지면 오히려 과최적화가 된다는 것을 할수 있다. RMSE Test부분에서 Degree가 3일때가 5일때보다 낮은 것이 그 이유이다. |

1. Given a wine dataset, implement the softmax regression model in ‘model/SoftmaxClassification.py.’

Note 1. For detailed information on the wine dataset, please refer to [UCI Machine Learning Repository: Wine Data Set](https://archive.ics.uci.edu/ml/datasets/wine).

1. **[20 pts] [Softmax Classification]** Implement SoftmaxClassificaiton using PyTorch.

Answer: Fill in your code here. You also have to submit your code to i-campus.

|  |
| --- |
|  |

1. **[20 pts] [L2 Regularization]** Implmenet the L2 regularization, compare the performance of the model with/without the L2 regularization term, and briefly explain the result. A loss function with the L2 regularization is defined below.

**.**

|  |  |  |
| --- | --- | --- |
| **for L2 Regularization** | **Train accuracy** | **Test accuracy** |
| **0.0** | **58.4507** | **66.6667** |
| **0.01** | **73.2394** | **86.1111** |
| **0.1** | **80.9859** | **83.3333** |
| **1.0** | **61.9718** | **77.7778** |

Answer: Fill in your code here. You also have to submit your code to i-campus.

|  |
| --- |
| Lamda의 값이 커질수록 정확도가 증가되지만 어느일정이상 커지면 정확도가 떨어진다. 그래서 Lamda값을 잘 조절해서 넣어야한다. |

1. Implement the evaluation metrics in ‘evaluation/EvaluationMetric.py’ and report the model performance on the given dataset.

**(a) [10 pts] [Confusion Matrix]** Implement **\_set\_confusion\_matrix** method in the EvaluationMetric class and **draw the heatmap for each class** that is output as a result of the ‘3\_Evaluation\_main.py.’

Answer: Fill in your code here. You also have to submit your code to i-campus.

|  |
| --- |
|  |

**(b) [20 pts]** **[Evaluation Metrics]** Based on the confusion matrix, implement **precision**, **recall**,**f-measure**, and **marco-averaged** metrics for the overall performance.

Answer: Fill in your code here. You also have to submit your code to i-campus.

|  |
| --- |
|  |