**Tutorial 2 NLP**

**Name:** Hitesh Vaidya

**UID #:** 87467572

**Objective:**

To study tokenization and embedding using bag or character and bag or words and test performance of MLP using 3 learning algorithms

**Methodology:**

As mentioned by professor in the class, two tokenization approaches were used and tested with MLP. A grid search of parameters was performed on the MLP. As mentioned by professor, no tokenization or grid search was expected with random MLP update and Feedback Alignment.

**Character tokenization on IMDB dataset used with MLP:**

Using Bag of Characters embedding, following set of parameters were used to perform grid search on MLP,

# List of dimensions in all the layers of the network

# The input size is determined by the dimension of the bag-of-characters vector.

hidden\_layers = [1, 2, 3]

hidden\_sizes = [128, 256, 512]

output\_size = 2

# activation

activations = ['ReLU', 'Tanh', 'LeakyReLU']

# learning rate

learning\_rates = [0.001, 0.0005, 0.0001]

# batch size

batch\_sizes = [32, 64, 128]

# optimizer

optims = ['SGD', 'Adam', 'RMSProp']

epochs = 10

trials = 60

A function named generate\_params() is used to generate a combination of parameters from the above shown options. Every set of parameters combination is tested on random seed which is set using set\_new\_seed(trial\_number) function. For every combination of parameters, we perform 3 trials and select the metrics based on the best test accuracy score. Therefore, we performed about 60\*3 = 180 trials in total for character tokenization.

The vocabulary size for character tokenization was 134 which was also the sequence length of the input for MLP. The test accuracy values for character tokenization ranged from 0.55 to 0.60 for all combinations of parameters.

The metrics of all the trials are recorded in file named **char\_token\_experiments\_log.csv**

The **best performing configuration**,

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| seed | layers | activation | optimizer | learning\_rate | test\_accuracy | test\_loss | precision | recall |
| **1082806321** | **1** | **LeakyReLU** | **Adam** | **0.001** | **0.611** | **0.6872** | **0.5958** | **0.6905** |

For the above parameters we performed 3 trials again with different seed in each run and following are the mean and std values of the metrics

loss for best parameters, mean: 0.6695, std: 0.0147

accuracy for best parameters, mean: 0.6014, std: 0.0103

precision for best parameters, mean: 0.5896, std: 0.0234

recall for best parameters, mean: 0.6990, std: 0.1112

**Word Tokenization on IMDB dataset used with MLP**

The bag of words tokenization was performed by setting char\_level=False in the char\_level\_tokenization() function provided. Similar to char tokenization experiments, grid search was performed on word level tokenization using following set of parameters,

# List of dimensions in all the layers of the network

# The input size is determined by the dimension of the bag-of-words vector.

hidden\_layers = [1, 2, 3]

hidden\_sizes = [512, 256, 128]

output\_size = 2

# activation

activations = ['ReLU', 'Tanh', 'LeakyReLU']

# learning rate

learning\_rates = [0.001, 0.0005, 0.0001]

# batch size

batch\_sizes = [32,64,128]

# optimizer

optims = ['SGD', 'Adam', 'RMSProp']

epochs = 10

for seed and parameter combination we used the same functions as used for char level tokenization. As well as, same number of trials were performed. The size of vocabulary for word level tokenization was 80169 and same is the size of sequence length. The test accuracy values ranged from about 0.53 to 0.87.

The metrics of all the trials are recorded in file named **word\_token\_experiments\_log.csv**

The **best performing model** had following set of parameters,

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| seed | layers | Activation | optimizer | learning\_rate | test\_accuracy | test\_loss | precision | recall |
| **8.99E+08** | **1** | **ReLU** | **RMSProp** | **0.001** | **0.8781** | **0.3134** | **0.878** | **0.8783** |

For the above parameters we performed 3 trials again with different seed in each run and following are the mean and std values of the metrics

loss for best parameters, mean: 0.3420, std: 0.0157

accuracy for best parameters, mean: 0.8664, std: 0.0111

precision for best parameters, mean: 0.8442, std: 0.0337

recall for best parameters, mean: 0.9025, std: 0.0286