**Optimizing Neural Networks on the IMDB Dataset**

**Assignment - 2**

Abstract

Sentiment analysis of the IMDB dataset has been subjected to this research in order to understand how the accuracy of a neural network model can be improved. Different structures of neural network were tried out, where parameters that changed included the number of hidden layers, the number of units in the hidden layers, the nature of the activation and loss functions. Other standard techniques including dropout and L2 Regularizations were also applied to enhance the validation accuracy of the model that was being used. The experiments proved fruitful; the best model was the one obtained with the parameters: test accuracy equaled to 86.80%, and test loss that equaled 0,1313.

Introduction

They have played a greater role in automated sentiment classification that enhances the efficiency in precisely matched dataset of sentiment such as IMDB movie review dataset. The aim of this project was to improve an existing binary classification neural network model (positive/negative reviews classification) based on different architectural and hyper parameters tuning techniques. The objective was to improve the model’s accuracy, eliminate the model’s ability to memorize the train data set and be able to predict unseen data by tweaking through the hidden layers, activation functions, regularization techniques and several others.

Data Preparation

Dataset Loading

IMDB dataset of 50,000 movie reviews was read using Keras’s built-in function imdb.load\_data(num\_words=10000) under which only 10,000 most frequently occurring words were considered. This preprocessing was performed mainly to control the size of input and unneccessary complexity of the model.

Model Architecture

Layer Configuration

The model's architecture was tested with several configurations to identify the best-performing model:

Hidden Layers: Test of the two factors; = Models with 1, 2 and 3 layers of hidden layers were used in order to see how depth provided a relation to accuracy.

Hidden Units: Changed the number of hidden units per layer to 16, 32, 64 and 256 to see how this affects performance.

Activation Functions: Two activation functions were tested: relu, which dominates contemporary neural networks, and tanh, used in the early days of big nets.

Loss Functions: To determine whether binary\_crossentropy is suitable for use in binary classification, it was compared with mse (mean squared error).

Regularization Techniques: L2 regularization was incorporated to minimize high weight values that cause overfitting; the dropout layers were added to increase generality.

Compilation and Training

Optimization and Loss

Optimizer: We chose the RMSprop as it optimally addresses sparse gradients.

Loss Functions: In the experiments, both binary\_crossentropy and mse were applied with the binary\_crossentropy used for binary classification.

Training Procedure: It also took 20 iterations of training to complete, using a batch size of 512. In addition, the independent validation set was set aside from the learning dataset so that the extent of over- fitting could be tracked.

Evaluation and Results

Performance Metrics

Hidden Layers: Among the models tested, the model with two hidden layers provided the highest accuracy of both training and validation data. The single-layer approach gave the lowest accuracy, and the three-layer model exhibited some level of overtraining.

Hidden Units: When changed the number of hidden units to 64 it was found that the training and the validation accuracies were the highest. It is for this reason that when using the 256 hidden units, overfitting took place.

Activation Functions: It is seen that the model with relu activation function was better than the model with tanh in training accuracy and most significantly the validation accuracy.

Loss Functions: Binary\_crossentropy gave better accuracy and less loss than mse, this correlates with the fact that binary classification is optimal for binary\_crossentropy.

Regularization and Dropout: I was able to control overfitting that allowed for higher validation accuracy by using dropout layers and L2 regularization.

Training and Validation precision

The most successful model, with two hidden layers, 64 hidden units, relu activation, and binary\_crossentropy loss function, achieved:

Training Accuracy: 93.85%

Validation Accuracy: 87.61%

Test Accuracy: 86.80%

Test Loss: 0.1313

Summary of Findings

Hidden Layers: The two-layer model was confirmed to be the most effective one unless the number of layers was maximized which resulted in overtraining. One layer was not enough to address the complexity of the data being presented.

Hidden Units: As the number of hidden units was increased to 64, the flushing capacity of this model was enhanced and further enhancement such as addition of 256 units was realized to distort over enhancements.

Activation and Loss Functions: Choosing relu as an activation function for the last layer of the model combined with binary cross entropy function was the best configuration when it was compared with other permutations.

Regularization: Thus the dropout and L2 regularization techniques completed the task to minimize the overfitting of the model and brought beneficial changes to validate the model with better performance on the validation set.

Graphical Analysis

The training as well as the validation set figures depicted that they are having a reduced loss per epoch and an improved accuracy per epoch. Adding the techniques like regularization led the models to converge smoother and generalize well than the models without these techniques.

Conclusion

From this work it proved that one can build optimized models in terms of architecture of neural networks for sentiment classification tasks. The best of the model was achieved 86.80% on the testing set conducive to the rationale behind choosing right no. of hidden layers, no. of hidden units/ nodes, activation functions & the type of regularization to be applied. As a continuation of this work, it may be useful to consider different types of networks, for example, convolutional or recurrent, and examine how their use affects classification performance and training time.