**Neural Network Analysis of IMDB Dataset**

**Assignment -2**

**Harini Kavipurapu**

**Abstract**

In this present paper, an attempt has been made to work deep into a proved neural network-based sentiment analysis model on the IMDB database and extend it further. Experiments conducted included using different architecture and implementing different techniques, such as modifying the number of hidden layers, and the number of hidden units in each layer, the different forms of activation functions and then the loss functions . Due to the overfitting problem, several regularization techniques were used in this experiment, including the use of dropout in attempts to achieve better results on the validation set.

**Introduction**

Neural networks suggest large enhancements in the automated sentiment categorization process. This work sought to implement and improve an architecture of a neural network, for classifying movie reviews as either positive or negative from the IMDB dataset. In modifications, changes in model performance were made through several methods as discussed in the assignment.

**Data Preparation**

* **Dataset Loading**

For the textual data, IMDB dataset comprising of 50,000 movie reviews has been used and it was loaded by using Keras’s imdb.load\_data(num\_words=10000) function. To manage the input size and dimensionality the vocabulary was restricted to the last 10000 words found in the corpus.

**Model Architecture**

**Layer Configuration**

* **The neural network model was experimented with using different architectures:** 1, 2, and 3 Hidden Layers: Different types of hidden layers were tried as Keras allows one, two and three hidden layers to check the impact of the same to validate and test the model.
* **Hidden Units:** The experiment was performed on a network with a configuration of two hidden layers, where 32 and 64 units were added to each layer to determine the best configuration.
* **Activation Functions:** relu and tanh is available as activation functions, relu is new style activation, and tanh is old school style activation.
* **Loss Function:** The binary\_crossentropy loss function was then compared with the mse loss function. This determines how it affects performance on a binary classification problem.
* **Regularization Techniques:** An L2 regularization was added to the weights to discourage large weights and make the model more generalizable and to add dropout layers so that that the model does not over learn.

**Compilation and Training**

**Optimization and Loss**

* **Optimizer:** The model was trained using RMSprop as the model optimizer, this is because all these optimize for sparse gradients.
* **Loss Function:** During the experimentation both binary\_crossentropy and mse loss functions were evaluated binary\_crossentropy is preferable for tasks of this type of binary classification.
* **Training Procedure:** The training of the model took 20 epochs using a batch size of 512. Cross validation method was utilized to check for overfitting and to predict future unknown data sets.

**Evaluation and Results**

**Performance Metrics**

* **Hidden Layers:** The model with two hidden layers gave the best result in terms of both training accuracy and validation accuracy. The model with only one layer was not as effective in the problem, the three-layer model possibly over-fitting the problem after a few epochs.
* **Hidden Units:** Taking measures to increase the number of hidden units to 64 for training improved on the performance.
* **Activation Function:** Out of the two, the model with relu surpassed the other, especially with validation accuracy. The model with tanh activation was not as good in generalization as the relu-based models.
* **Loss Function:** The binary\_crossentropy proved to be much more effective at producing reduced accuracy and loss than mse which is less appropriate for binary classification.
* **Regularization and Dropout:** When dropout layers were applied with regularizing by L2, it appeared that overfitting is avoided, and the validation sets show more realistic accuracy.

**Training and Validation Accuracy**

* In training and validation, the models achieved the accuracy of 89.76% and 89.48% respectively.
* The model using the two layers of 64 units, relu activation, and binary cross entropy training accuracy of the model was 93.85% while validation accuracy was 87.61%.
* The highest score with dropout and L2 with the test accuracy of 86.80% and the test loss of 0.1313 was received by the best-chosen model.

**Summary of Findings**

* **Effect of Hidden Layers:** The model with more layers (three hidden layers) and more parameters overfit the data, despite having more layers. A less complicated architecture with two hidden layers gave a better generalization.
* **Effect of Hidden Units:** The increase in hidden units to 64 saw improvement in the learning capability of the model.
* **Effect of Activation and Loss Function:** As for the activation function output layer must be binary\_crossentropy, since it is binary classification and so the best activation function as the result of experiment is relu.
* **Regularization:** Both dropouts, and L2 regularization did help in reducing overfitting and improved results on both the validation and test set.

**Graphical Analysis**

The training and validation curves came out to show that: With increase in epochs, both the training loss and validation loss reduced, and the training accuracy and validation accuracy increased. Indeed, the model with regularization showed flatter convergence curve, which proved better generalization was achieved.

**Conclusion**

The final model was able to distinguish between positive and negative movie reviews successfully: the accuracy of the model in the test set was 86,80%. These experiments showed that modifying hidden layers, modifying number of hidden units; applying appropriate form of regularization, plays a crucial role in its performance. It would be worthwhile for similar work to be done using other optimization methods and complex network topologies to improve performance and minimize. Capital intensity.