Name: Venkata Sai Reddy Peddireddy

Assignment: Neural Network

Git Hub link:
https://github.com/venkatasai8120/AML_Assignments.git

Neural Network summary Report

The purpose of this assignment was to study how architecture and the choice of network training influence the performance of a classifier on the IMDB dataset. The dataset consists of 25,000 reviews for training and 25,000 reviews for testing. All words were converted into a 10,000-dimensional multi-hot vector.
I trained 11 models using the RMSprop optimizer, varying the number of layers used, the number of hidden units, activation functions, the loss function, as well as including Dropout and L2 regularization.

Model	**Layers & Units**	**Activation**	**Regularization**	**Dropout**	**Loss Function**	**Validation Accuracy**	**Validation Loss**	**Test Accuracy**	**Test Loss**
M1	16-16	ReLU	None	0.0	BCE	0.8708	0.5246	0.8594	0.5652
M2	16	ReLU	None	0.0	BCE	0.8752	0.3755	0.8659	0.4012
M3	16-16-16	ReLU	None	0.0	BCE	0.8682	0.6066	0.8602	0.6486
M4	32	ReLU	None	0.0	BCE	0.8758	0.3987	0.8661	0.4213
M5	64	ReLU	None	0.0	BCE	0.8763	0.4080	0.8663	0.4327
M6	128	ReLU	None	0.0	BCE	0.8750	0.4057	0.8680	0.4279
M7	16	ReLU	None	0.0	MSE	0.8772	0.0890	0.8707	0.0948
M8	16	Tanh	None	0.0	BCE	0.8731	0.4145	0.8639	0.4434
M9	16	ReLU	None	0.5	BCE	0.8831	0.3371	0.8738	0.3615
M10	16	ReLU	L2 (1e-3)	0.0	BCE	0.8777	0.3637	0.8722	0.3776
M11	64-32	ReLU	L2 (1e-4)	0.5	BCE	0.8826	0.6263	0.8719	0.6684

Model Architecture:

Neural network architecture, including the number of hidden layers and the number of units in each hidden layer, determines the learning capacity of a neural network. For model 2 (one hidden layer with 16 units) loss on validation is low (37.55), but accuracy is moderate (87.52%). The two hidden-layer version (Model 1) with 16 units has similar accuracy, but a higher validation loss (52.46), showing slight overfitting. The Model 3 (three layers) had higher loss meaning that adding more layers was unnecessary and only made the model more complicated. However, later runs (Models 4-6) gain only slightly, suggesting little difference after 64 units. Overall, simpler models tend to yield better results on this dataset.

Activation Functions:

Activation functions make the model non-linear They let it learn patterns that are complex. Most models used ReLU. Model 8 used Tanh in contrast. The ReLU models had faster training time, higher accuracy, and a lower loss overall. Tanh (Model 8) didn't help either: its validation loss was 41.45.For sentiment classification, ReLU is the best default. In later models, if you have dead neurons or unstable training, try Leaky ReLU or Batch Normalization.

Loss Functions:

Loss functions determine how the network modifies the weights. Most models used Binary Cross entropy as loss function, recommended for binary classification (positive vs negative reviews). Model 7 had a better loss when using MSE, although the accuracy did not improve compared to other models. It also shows BCE still performs better on text classification, since it relies on differences.

Regularization (L2 and Dropout):

Regularization prevents excessive fitting by stopping the network as it memorizes the training data. Model 9 used Dropout (0.5). Model 9 had the best performance at Val = 88.31% and Test = 87.38%. Model 10 and Model 11, with L2 regularization, exceeded the model’s lacking regularization. Dropout or L2 regularization techniques help training converge and improve generalization. Dropout from 0.3 to 0.5 is recommended. The recommendation includes L2 weight decay to balance the training and validation performance.

Final Insights:

After evaluating all 11 models, Model 9 (1 Hidden Layer, 16 Units, ReLU, Dropout 0.5, Binary cross entropy) was selected as the best model. This model had the best validation accuracy (88.31%) and the best validation loss (33.71) of all models it generalized well to the validation set. More layers (Model 3) did not yield better results. A stable and accurate solution was achieved using ReLU/Binary Cross entropy. Dropout was most effective at reducing overfitting. This experiment's results further support the idea that simple networks regularize appropriately and thus outperform deeper nets. Researchers may try other optimizers like Adam. They could also try more techniques like Batch Normalization and Leaky ReLU to improve stability or performance.