**Neural Networks**

**Introduction to Neural Network:**

Neural networks, a subset of machine learning and form the foundation of deep learning, draw inspiration from the human brain's signaling process. They consist of layers of interconnected nodes, including an input layer, hidden layers, and an output layer. Each node has its own set of weights and a threshold. When the output of a node surpasses its threshold, it activates and passes data to the next layer. Each node has an input, weights, bias, and an activation function. Neurons are the fundamental units of artificial neural networks. Weights are the numerical values assigned to the connections linking neurons. The activation function is a mathematical formula that calculates the neuron's output based on its inputs. Bias is a constant value added to the neuron's input to enhance its performance. Data flows through the network in a feedforward manner, with one node's output becoming the next node's input. During the training phase, the nodes in a neural network receive input data from the input layers in the form of tensors. These nodes then perform operations on the data and pass it on to the next layer until a final output is produced by the output layer. To make accurate predictions, random weights are initially assigned to each of the nodes, which are then adjusted during the iteration process to minimize the difference between the predicted and actual values.

**IMDB Dataset:**

The IMDB dataset is a compilation of 50,000 reviews from the Internet Movie Database that are extremely polarized. The dataset is split into two sets - training and testing purposes. Both sets have an equal number of positive and negative reviews. Keras provides the IMDB dataset, which comprises preprocessed reviews and their corresponding labels. The reviews are made up of a sequence of words, and the labels indicate whether they are positive (1) or negative (0). Each integer in the sequence corresponds to a specific word in the dictionary. The IMDB model consists of two layers, each with 16 units. The activation function used is "Relu" and the loss function is "binary cross-entropy".

Here are the different models and their corresponding accuracy scores:

|  |  |
| --- | --- |
| Model | Accuracy |
| Basic | 88.38 |
| Adding Layers | 87.30 |
| Reducing Layers | 88.68 |
| Increased Units to 32 | 87.62 |
| Increased units to 64 | 86.46 |
| MSE loss function | 88.14 |
| Tanh activation function | 87.40 |

From the above table, it is observed that the accuracy varies with the different parameters.

Below is a summary of the performance of various models trained on the IMDb dataset.

Basic Model has the Accuracy of 88.38%. The basic model is a simple neural network architecture that achieved a good accuracy score.

When extra layers were added to the neural network, the accuracy slightly decreased (Adding Layers model with accuracy of 87.30%). This suggests that increasing complexity by adding more layers didn't provide a significant accuracy improvement. If there are too many hidden layers, the network may become too specialized to the training dataset, resulting in performing poorly on unseen data.

In contrast, when the the layers are decresed the accuracy improved (Reduced Layers model with accuracy of 88.68%), that a simpler model architecture, with fewer hidden layers, can perform well for this dataset.

it is important to note that If there are too few hidden layers, the network may not be able to capture the complex relationships between the inputs and outputs.

Increasing the number of units (neurons) in the hidden layers to 32 didn't result in a significant accuracy boost (Increased Units to 32 model with accuracy of 87.62%). Further Increase in the neurons that is ,increasing the units to 64 led to a decrease in accuracy ( Increased Units to 64 model with accuracy of 86.46%). This indicates that an overly complex model might overfit the training data, causing poorer generalization to the test data.

Expanding the number of units in the network can help the model learn more information, leading to the identification of complex patterns and ultimately improving accuracy. However, this can also increase the likelihood of overfitting, which occurs when the model performs poorly on test data due to having learned too much from the training data.

Changing the loss function to Mean Squared Error (MSE) from binary cross-entropy didn't substantially affect accuracy. Binary cross-entropy is typically more suitable for binary classification tasks like sentiment analysis. MSE Loss Function model gave the Accuracy of 88.14%).When working with classification models such as the IMDB dataset, the most suitable loss function to use is binary cross entropy. This is because MSE loss function is mainly used for regression models. If the MSE loss function is used for a classification model, it may not accurately address the model's objective. Predicted error is not measured as the difference between true and predicted values, but as the number of misclassified samples. Consequently, the accuracy of the model decreases.

Switching the activation function to hyperbolic tangent (tanh) instead of ReLU didn't significantly impact accuracy. Activation functions can influence information processing, but their effect depends on the problem and dataset. In a classification model, the choice between Relu and tanh activation function depends on the architecture and problem type.

ReLU is a popular choice for most applications due to its simplicity, computational efficiency, and ability to mitigate vanishing gradients. It is often used as the default activation function in hidden layers. Tanh activation function, on the other hand, is symmetrical to the origin, making it useful for data with positive and negative values. tanh can be useful when you need outputs centered around zero.

Regularization is a crucial technique in neural networks to prevent overfitting, it occurs when a model performs good on the training data but poorly on unseen data and allows it to perform well both in the training and testing data.

The use of L2 regularization in machine learning encourages the model to have smaller weights. This is achieved by adding up a penalty term to the loss function, which helps in reducing the complexity of the model. The L2 regularization term is calculated by summing the squares of all weight values in the network and adding it to the loss function.

A graph of a graph showing the loss of a training and validation

Description automatically generated with medium confidence

Dropout is one of the regularization techniques that randomly drops (sets to zero) a fraction of neurons during each forward and backward pass of training. It introduces randomness and prevents neurons from co-adapting too much. A random subset of neurons is deactivated, and their outputs are set to zero. This forces the model to learn more robust features by relying on different combinations of neurons.

A graph of training and validation

Description automatically generated

We can observe from both plots that the validation loss increases with the increase in the epochs from a certain point after reaching the optimum point, both in the L2\_regularization and dropout techniques. Whereas training loss decreases with an increase in the epochs.