**Neural Network Assignment-1**

1. Modify an existing neural network model to improve performance?

There are several ways to modify an existing neural network model to improve its performance. Here are a few:

* Adjust the architecture: The architecture of a neural network includes the number of layers, the number of neurons in each layer, and the types of layers. Experiment with changing the number of layers or neurons, or adding or removing layers, to find a configuration that works better for your data.
* Change the activation function: The activation function controls the output of each neuron in the network. Try using a different activation function, such as the rectified linear unit (ReLU) or the hyperbolic tangent, to see if it improves the model's performance.
* Modify the loss function: The loss function is used to calculate the error between the predicted output and the true output. If your model is not performing well, try using a different loss function, such as mean squared error (MSE) or categorical cross-entropy, to see if it improves the model's accuracy.
* Regularize the model: Regularization techniques can be used to prevent overfitting and improve the generalization of the model. You can try adding dropout layers, L1 or L2 regularization, or early stopping to improve the model's performance.
* Increase the amount of data: If your model is not performing well, try increasing the amount of data that you are training on. This can help the model learn more patterns and generalize better to new data.
* Fine-tune the model: If you are using a pre-trained model, you can fine-tune it on your specific task by training it on your data for a few more epochs. This can help the model learn to better recognize patterns that are specific to your data.

These are just a few ways to modify an existing neural network model to improve its performance. Depending on your specific task, other modifications may be more appropriate.

1. Explain how different approaches affect the performance of the model?
   * Increasing the complexity of the model
   * Modifying the loss function
   * Regularizing the model
   * Increasing the amount of data
   * Changing the activation function

**For the IMDB example that we discussed in class, do the following:**

1. You used two hidden layers. Try using one or three hidden layers and see how doing so   
affects validation and test accuracy.

The number of hidden layers is an important hyperparameter that can affect the performance of the model.

# One hidden layer

Graphical user interface, text

Description automatically generated with medium confidence

Chart, line chart

Description automatically generated

In one hidden layer the validation loss: 0.2767 and validation accuracy: 0.8893 which is highest in Epoch 5.

#Three hidden layers

Text

Description automatically generated with low confidenceChart, line chart

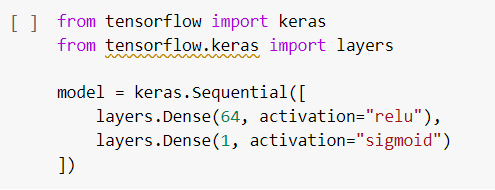
Description automatically generated

1. Try using layers with more hidden units or fewer hidden units: 32 units, 64 units, and so   
   on?

Graphical user interface, text, application

Description automatically generated

Validation loss: 0.2782 validation accuracy: 0.8899



Validation loss: 0.2754 – validation accuracy: 0.8887

1. Try using the mse loss function instead of binary\_crossentropy.

Text

Description automatically generated

Validation loss: 0.0838 validation accuracy: 0.8863

1. Try using the tanh activation (an activation that was popular in the early days of neural networks) instead of relu.

A picture containing text, indoor

Description automatically generated

1. Use any technique we studied in class, and these include regularization, dropout, etc., to   
    get your model to perform better on validation.

**some techniques to improve model performance on validation data:**

1. Regularization: Regularization techniques like L1, L2 or Elastic Net regularization can be used to prevent overfitting. Regularization reduces the complexity of the model and helps it generalize better. Regularization can be applied to various layers of the model.
2. Dropout: Dropout is a regularization technique that randomly drops out a fraction of the neurons during training. This helps to prevent overfitting and improve generalization. Dropout can be added after every layer in the network.
3. Batch Normalization: Batch normalization is a technique used to normalize the inputs of each layer in a neural network. It helps to stabilize the distribution of inputs and can speed up the training process. Batch normalization can be added after each layer in the network.
4. Early stopping: Early stopping is a technique used to prevent overfitting by monitoring the validation loss during training. The model training is stopped when the validation loss stops improving.
5. Data Augmentation: Data augmentation is a technique used to increase the size of the training dataset by creating new synthetic examples. This helps the model to generalize better and reduces overfitting.
6. Hyperparameter Tuning: Tuning the hyperparameters like learning rate, number of epochs, batch size, activation functions etc., can help in improving the performance of the model on validation data.

These techniques can be used in isolation or in combination to improve the performance of the model on the validation data. It is important to experiment with these techniques to find the optimal combination that works best for the specific problem and dataset.