REPORT

ASSIGNMENT-02 NEURAL NETWORKS

ADVANCED MACHINE LEARNING:

STEPS THAT ARE DONE BEFORE BUILDING UP THE MODEL:

- The code collects the training and test sets from the IMDB dataset after loading it from Keras. It specifically uses the imdb.load data() function to load the dataset and return the training and test sets as tuples of NumPy arrays. The num words parameter is set to 10,000, so only the 10,000 words that appear most frequently in the reviews are going to make up the dataset. This is a common strategy to minimize the quantity of input data and exclude words that are hardly used and might not add much to the categorization task.
- The code then accesses train data[0] and outputs the first movie review from the training set. The output is a list of integers representing the words in the review, where each integer's value represents a word's position in the dictionary of the 10,000 most common words.
- The resulting maximum value i.e., 9999 represents the total number of unique words in the dataset since the integer values represent the index of a word in a dictionary of the 10,000 most common words. The output of this code offers us a general sense of the neural network's input layer's size, which will be used to categorize the movie reviews.
- A string containing the original text of the first movie review in the training set is the decoded review variable that is returned. This code is beneficial in comprehending the structure of the input data and ensuring that the input has been loaded and precompiled correctly.
- Target values for the neural network's evaluation and training will be arrays. The network's objective is to discover a mapping between the one-hot encoded input sequences (x train and x test) and the appropriate sentiment labels (y train and y test).

BUILDING THE MODEL:

This model consists of 4 fully connected layers.

The new model uses **dropout and regularization** and contains four dense layers with 64 neurons each. The output layer utilizes a "**sigmoid**" activation function, while the hidden levels use a "**tanh**" activation function. The optimizer is "**adam**," and the loss function is "**mse**". Using a batch size of 512, the model is trained on the training set for 4 epochs. Loss and accuracy are the evaluation measures that are used.

A neural network model that also was trained on a sentiment analysis task and used the IMDB movie reviews dataset's training and validation accuracy and loss is presented in the output above. The model is trained using 512-batch training over 20 epochs.

- The model is composed of an output layer with a single neuron and a sigmoid activation function, four fully connected levels (referred to as "Dense layers"), and two dropout layers (referred to as "Dropout layers"). 64 neurons make up the first layer, and the hyperparameter kernel regularizer is set to regularizers.L2 (0.005). By compelling the network to apply smaller weights, this provides an L2 regularization penalty to the layer's weights to try prevent overfitting.
- The hyperbolic tangent activation function is utilized in the second and third layers, that each contain 64 neurons. There is no regularization provided to the weights of these layers.
- Then also added the Dropout layers with a dropout rate of 0.5 are the fourth and fifth layers. Dropout is a regularization method that, during training, erratically removes (i.e., sets to 0) a portion of the output values from the preceding layer. As a result, the network is prompted to acquire more resilient and generalizable features, which helps prevent overfitting.

Overall, by outlining the essential components for optimization, evaluation, and errors calculation, the compile technique gets the model ready for training.

Compiling:

It is usual practice to evaluate the model's performance during training and prevent overfitting by setting aside a validation set. To track the model's performance, it is trained on the training set and assessed on the validation set after each iteration.

Summary:

- ➢ The validation accuracy grows from 0.8222 to 0.8761 over the same time span, whereas the training accuracy rises from 0.7329 in the first epoch to 0.9385 in the final epoch. This may indicate that the model is not overfitting and is instead learning how to appropriately classify the reviews.
- ➢ The validation loss falls from 0.2947 to 0.1257 over the same time period, and the training loss decreases from 0.5242 in the first epoch to 0.0781 in the final epoch. This indicates that the model is increasingly more certain of its predictions and is steadily improving at detecting and classifying the reviews.
- ➢ The model is assumed to be performing well overall on this activity, according to its accuracy of 0.8761 on the validation set. It is important to remember that the model's performance can change depending on the individual use case and dataset.
- ➢ The output indicates that the model achieved a test loss of 0.1313 and a test accuracy of 0.8680.

- The first value 0.1313 represents the test loss of the model, while the second value 0.8680 represents the test accuracy of the model.

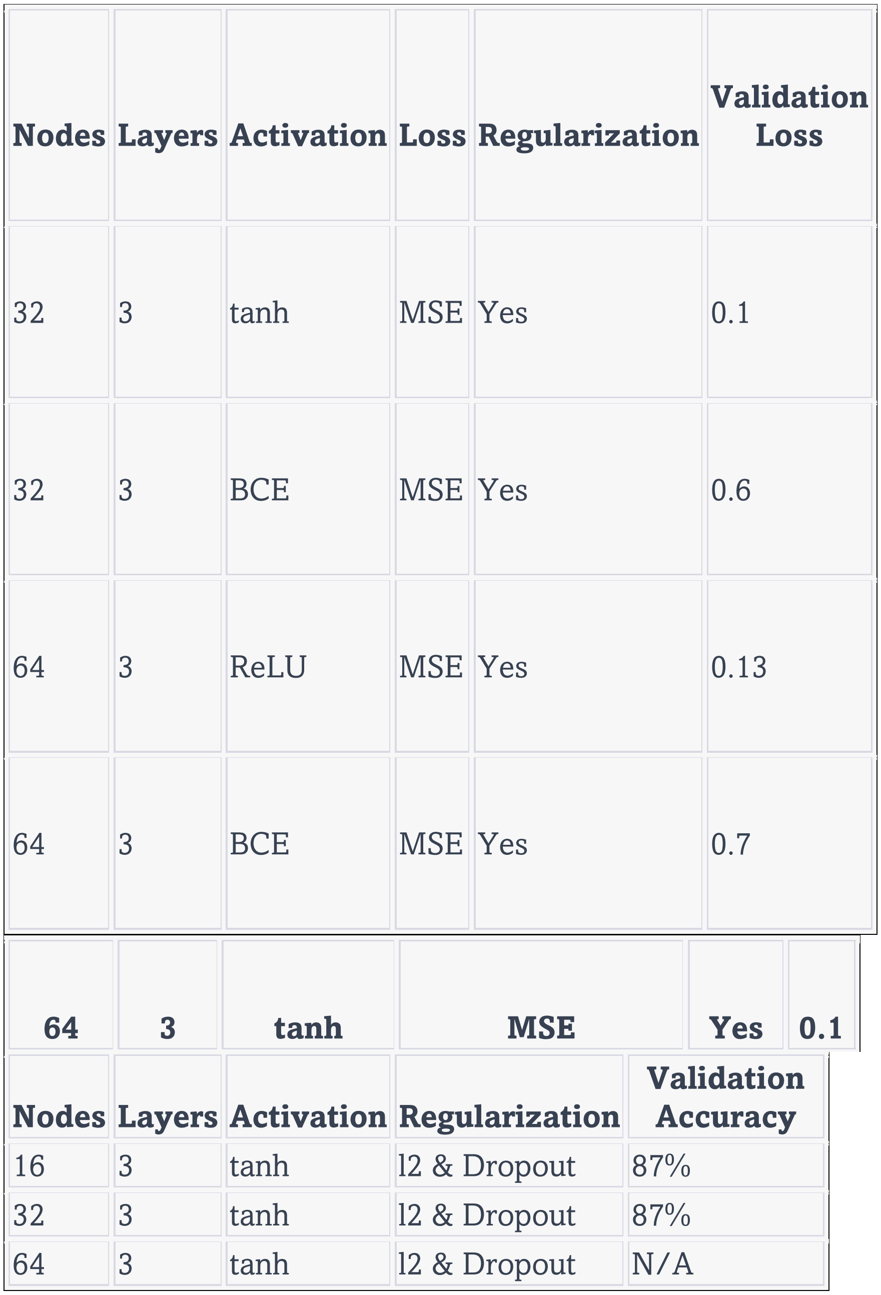
The results variable includes the evaluation metrics of the model on the test dataset after retraining. The mean squared error (MSE) loss on the test dataset, which is listed as the first number in the list, is 0.1656, and the model's accuracy on the test dataset is listed as the second value, at 0.8727.

Adam: The optimizer "adam" has been utilized in the test set. Adam is an optimization algorithm that is computationally efficient, resistant to the assignment of hyperparameters, and changes the learning rate for each component based on its historical gradients. My model's accuracy using Adam Optimize is 88%.

RMSprop: it is an optimization algorithm that modifies the learning rate for each parameter using a trend line of the squared gradients. In the presence of distracting gradients, this can help avoid oscillations and limiting learning. In processes like voice recognition, natural language processing, and image recognition, RMSprop is frequently utilized. With the rmsprop optimizer, the test set's reliability is 86%.

Summary of validation loss function in a table:

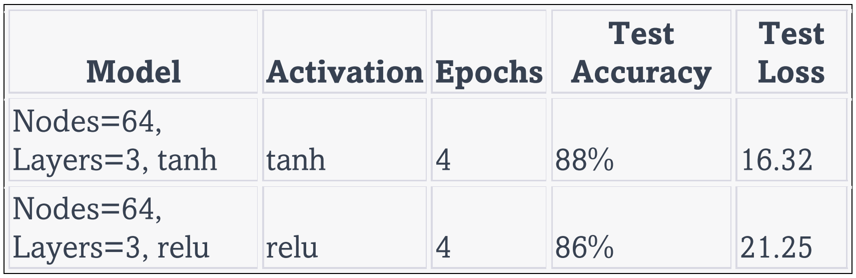
Nodes	Layers	Activation	Loss	Regularization	Validation Loss
16	3	tanh	MSE	Yes	0.12
16	3	BCE	MSE	Yes	0.58
32	3	ReLU	MSE	Yes	0.13
32	3	BCE	MSE	Yes	0.4



Nodes	Layers	Activation	Loss	Regularization	Validation Loss
32	3	tanh	MSE	Yes	0.1
32	3	BCE	MSE	Yes	0.6
64	3	ReLU	MSE	Yes	0.13
64	3	BCE	MSE	Yes	0.7
64	**3**	**tanh**	**MSE**	**Yes**	**0.1**

Nodes	Layers	Activation	Regularization	Validation Accuracy
16	3	tanh	l2 & Dropout	87%
32	3	tanh	l2 & Dropout	87%
64	3	tanh	l2 & Dropout	N/A

Validation and training accuracy



Model	Activation	Epochs	Test Accuracy	Test Loss
Nodes=64, Layers=3, tanh	tanh	4	88%	16.32
Nodes=64, Layers=3, relu	relu	4	86%	21.25

Test accuracy