Neural Networks

S.no	Hidden Layers & Activation Used	Dense Units	Regularizations	Dropouts	Loss and Accuracy in Validation
1.	2 & “relu” (From original code)	16	None	None	loss: 0.2774 - Accuracy: 0.8880
2.	1 & "relu."	16	None	None	loss: 0.2884 - Accuracy: 0.8862
3.	1 & "relu."	32	None	None	loss: 0.2902 - Accuracy: 0.8833
4.	2 & "relu."	32	None	None	loss: 0.2793 - Accuracy: 0.8892
5.	1& "relu."	64	None	None	loss: 0.2909 - Accuracy: 0.8833
6.	2 & "relu."	64	None	None	loss-0.2842 accuracy: 0.8865
7.	1 &"tanh."	16	None	None	loss:0.0864 - accuracy:0.8877
8.	2 &” tanh."	16	None	None	loss:0.0835 - accuracy:0.8857
9.	3 &” tanh."	16	None	None	loss:0.1063 - accuracy:0.8603
10.	1 &” tanh."	32	None	None	loss:0.0905 - accuracy: 0.8800

Neural Networks

S.no	Hidden Layers & Activation Used	Dense Units	Regularizations	Dropouts	Loss and Accuracy in Validation
11.	2 &” tanh."	32	None	None	loss:0.0867 - Accuracy:0.8827 .
12.	3 &” tanh."	32	None	None	loss:0.0881 - Accuracy:0.8847
13.	1 &” tanh."	64	None	None	loss:0.0918 - Accuracy:0.8837
14.	2 &” tanh."	64	None	None	loss:0.0872 - accuracy:0.8840
15.	3 &” tanh."	64	None	None	loss:0.0854 - accuracy:0.8845
16.	2 & “tanh."	64	L2 = 0.001	None	loss:0.0881 - accuracy: 0.8825
17.	2 & “tanh”	64	L1_L2(L1=0.001, L2=0.01)	None	loss:0.5691 - accuracy: 0.8477
18.	2 & “tanh”	32	None	0.5	loss:0.0899 - accuracy:0.8763
19.	2 & “tanh."	64	None	0.5	loss:0.0849 - accuracy:0.8075

Summary:

- All the above models have been run with the following parameters to compile the model: optimizer='rmsprop', loss='mse function', metrics=['accuracy']).

  1. Except for models 1-6, run using the binary cross-entropy loss function. In binary classification, where the goal is to predict the proper label for each sample, the loss function "mse" cannot be employed, even though it exhibits minimal validation loss for each epoch. Instead, "mse" is used for regression problems to predict continuous variables.
- Using the “tanh” function instead of “relu” shows a steep increase in validation loss as high as 0.98 with no impact on the validation accuracy.
- On the current dataset, changing hidden from 2 to 1 & 3 does not significantly impact, with loss varying between 0.27 and accuracy at 0.88.
- Changing dense units from 16 to 32 and 64 shows a slight increase in validation loss for each epoch and no significant change in the accuracy.
- The concept of regularization helps control the data's overfitting by adding a penalty to the loss function with large weights. In the current scenario, L2 = 0.001 regularizations were applied, where the cost added is proportional to the square of the value of the weight coefficients.
- The effect of regularization L2 is clearly shown as the validation loss is at the peak of 0.6887. However, the validation accuracy is still at the constant of 0.88
- The concept of regularization can be applied in combination with L1 and L2 to mitigate overfitting. The current regularization was set as L1_L2(L1=0.001, L2=0.01).

- Both training and validation sets have proportionated loss and accuracy.
- Drop-out is another method of regularization to control the overfitting of data.
- Since earlier models showed improvement with L2 and dropout regularization, we applied L2 and dropout together to check the improvement of the model. The application of both regularizations shows no significant performance improvement, as the model's accuracy is constant at 0.88.

Conclusion:

Upon comparison of the model using different parameters, the regularization method using drop-out showed improvement in accuracy over other models at 0.8763, with a loss on the validation set being 0.5691. We can use the drop-out method as a hyperparameter to tune the model to increase its curacy of the model.

How this project was completed in detail:

Changing how many levels are hidden:

A single hidden layer minimizes the model's parameter count and makes it easier to understand. That lessens the model's ability to recognize intricate data patterns, however. As a result, the validation and test accuracy may be reduced compared to the initial model with two hidden layers.

b) Three hidden layers: This improves the model's potential to identify intricate patterns in the data. If the model is not correctly regularized, it might raise the possibility of overfitting. If the model can avoid overfitting, the validation and test accuracy might be better than it was with the two hidden layers in the original model.

Changing how many hidden units are present:

32 units: This limits the model's ability to recognize complicated data patterns. As a result, compared to the 64-unit original model, the validation and test accuracy may need to be more accurate.

b) 64 units: This was the model's first set of unit numbers. Model capacity and simplicity are well-balanced. The original model's validation and test accuracy may hold.

Changes to the loss function include:

> mse loss function, a different kind of loss function that calculates the mean squared difference between the output that is anticipated and the actual output. Regression issues frequently employ it. As in the IMDB example, it might not be

the ideal loss function for binary classification situations. In comparison to the initial model using the loss function of binary crossentropy, the validation and test accuracy may be reduced.

The following are some options for changing the activation function:

tanh activation function. This symmetric activation function was well-liked in the early days of neural networks. The relu activation function can be replaced with this. The vanishing gradient issue, however, could make it less useful for deep neural networks. Compared to the original model with the relu activation function, the validation and test accuracy may need to be more accurate.

Using regularization methods:

a) A penalty component is added to the loss function using L1 regularization, which pushes the model to learn light weights. With less overfitting and better generalization, this can be helpful. Compared to the original model, the validation and test accuracy could increase.

b) L2 regularization: This increases the loss function's penalty term, which motivates the model to pick tiny weights. Moreover, it can help to lessen overfitting and increase generality. Compared to the original model, the test and validation accuracy could increase.

b) Dropout regularization: This method randomly removes part of the neural network's units during training. This can lessen overfitting and increase generality. Compared to the original model, the test and validation accuracy could increase.

The performance of a neural network model may be enhanced using various techniques. The exact situation at hand and the qualities of the data will determine which strategy is best. It is crucial to assess the model's performance using the right metrics and compare it to other models to ensure the change is appreciable.