**Summary:**

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

Significance:

* My approach towards the problem, initially learned the importance of keras Sequential model which is stack of layers for building neural network.
* It contains the important imports like, layers, Dense, Dropouts and Regularizers required to design a neural network
* These are required imports

from tensorflow import keras

from tensorflow.keras import layers

from tensorflow.keras.layers import Dense

from tensorflow.keras.layers import Dropout

from tensorflow.keras import regularizers

* I implemented 2, 3 layered and 6 layered neural network with 16 ,64, 64 hidden neurons to check the performance.
* One important thing that I observed is no matter how many layers we stacked up, it will train and gives the almost same performance once it reaches the threshold.

model = keras.Sequential()

it follows the structure like input layer 🡪 hidden layers 🡪 output layer

The above line initializes the Sequential model where we can build up stack of layers.

model.add(Dense(64, activation="tanh"))

* The above line signifies that we are adding a new hidden layer with 64 dense units using tanh activation function.
* When I say 64 hidden units, we can assume that they are 64 neuron are being created in the layer to learn the data which is in form of vectors
* Activation function is also called transfer function, if the output range of function is limited, just sigmoid squashed the value to 1 which is above to it. They are non linear functions.

model.add(Dropout(0.5))

* The importance of Dropout is useful when we have a scenario of overfitting. The above line says, hey please randomly dropout some of my neurons since it causes overfitting. When I say 0.5, that implies 50 percent of my neurons are dropped out.
* I have tried using L1 and L2 regularizes but that does not cause much effect rather it diminishes performance. I think the model Is saturated and the best validation accuracy we can get a range of 86 or 87 percentage.
* Replaced binary\_crossentrophy with mean square error to check the performance metrics on the loss.
* The result is that validation loss has performed good. Initially when using binary\_crossentrophy the validation loss is 0.5 but when using mse it reduces to 0.1.
* Relu being the top among sigmoid and tanh function because of vanishing gradient problem. In this context tanh performs same as tanh.

|  |  |  |
| --- | --- | --- |
| Combinations | Training accuracy | Validation accuracy |
| 2 dense layers  16 hidden units  Tanh activation function  Optimizers = adam | 99.63 | 87.01 |
| 3 dense layers  64 hidden units  Tanh activation function  Dropouts (0.5)  Regularizers  Optimizers = adam | 98 | 86.73 |
| 6 dense layers  64 hidden units  Tanh activation function  Dropouts (0.5)  Optimizers = adam  regularizers | 98.75 | 86.12 |