Team id no:PNT2022TMID27778

Adding Dense Layers

The name suggests that layers are fully connected (dense) by the neurons in a network layer. Each neuron in a layer receives input from all the neurons present in the previous layer. Dense is used to add the layers.

Task 1: Adding Hidden layers

This step is to add a dense layer (hidden layer). We flatten the feature map and convert it into a vector or single dimensional array in the Flatten layer. This vector array is fed it as an input to the neural network and applies an activation function, such as sigmoid or other, and returns the output.

```
In [ ]:
#add hidden layer
model.add(Dense(output_dim=150,init='uniform',activation='relu'))
```

Key terms:

- init is the weight initialization; initialization function is the network initialization function which sets all the weights and biases of a network to values suitable as a starting point for training.
- Units, which denotes is the number of neurons in the hidden layer.
- Activation function defines the output of input or set of inputs or in other terms defines node of the output of node that is given in inputs. They basically decide to deactivate neurons or activate them to get the desired output. It also performs a nonlinear transformation on the input to get better results on a complex neural network. Its nodes here just pass on the information (features) to the hidden layer.

You can add many hidden layers.

Task 2: Adding output layer

This step is to add a dense layer (output layer) where you will be specifying the number of classes your dependent variable has, activation function and weight initializer as the arguments. We use add () method to add dense layers. In this layer, no need of mentioning input dimensions as we have mentions them in the above layer itself.

```
In [ ]:
#add output layer
model.add(Dense(output_dim=1,activation='sigmoid',init='uniform'))
```