Build a DNN using Keras with `RELU` and `ADAM`

Load tensorflow

```
!pip install -U tensorflow --quiet
```

Collect Fashion mnist data from tf.keras.datasets

```
import tensorflow as tf
import pandas as pd
import numpy as np
tf.__version__
```

```
'2.0.0'
```

Change train and test labels into one-hot vectors

```
(x_train, trainY),(x_test, testY) = tf.keras.datasets.fashion_mnist.load_data()
```

```
print("Number of samples in Training are x_train and y_train ",(x_train.shape,trainY.shape))
```

```
Number of samples in Training are x_train and y_train  ((60000, 28, 28), (60000,))
```

```
print("Number of samples in Test are x_test and y_test",(x_test.shape,testY.shape))
```

```
Number of samples in Test are x_test and y_test ((10000, 28, 28), (10000,))
```

```
x_train.dtype, x_test.dtype
```

```
(dtype('uint8'), dtype('uint8'))
```

```
trainX = x_train.astype('float32')
testX = x_test.astype('float32')
```

```
trainX.dtype, testX.dtype
```

```
(dtype('float32'), dtype('float32'))
```

```
set(trainY)
```

```
{0, 1, 2, 3, 4, 5, 6, 7, 8, 9}
```

```
pd.value_counts(trainY)
```

```
9    6000
8    6000
7    6000
6    6000
5    6000
4    6000
3    6000
2    6000
1    6000
0    6000
dtype: int64
```

```
pd.value_counts(testY)
```

```
7    1000
6    1000
5    1000
4    1000
3    1000
2    1000
9    1000
1    1000
8    1000
0    1000
dtype: int64
```

```
trainX[0]
```

```
array([[  0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,
          0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,
          0.,   0.,   0.,   0.,   0.,   0.],
       [  0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,
          0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,
          0.,   0.,   0.,   0.,   0.,   0.],
       [  0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,
          0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,
          0.,   0.,   0.,   0.,   0.,   0.],
       [  0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,
          0.,   1.,   0.,   0.,  13.,  73.,   0.,   0.,   1.,   4.,   0.,
          0.,   0.,   0.,   1.,   1.,   0.],
       [  0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,
          0.,   3.,   0.,  36., 136., 127.,  62.,  54.,   0.,   0.,   0.,
          1.,   3.,   4.,   0.,   0.,   3.],
       [  0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,
          0.,   6.,   0., 102., 204., 176., 134., 144., 123.,  23.,   0.,
          0.,   0.,   0.,  12.,  10.,   0.],
       [  0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,
          0.,   0.,   0., 155., 236., 207., 178., 107., 156., 161., 109.,
         64.,  23.,  77., 130.,  72.,  15.],
       [  0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,
          1.,   0.,  69., 207., 223., 218., 216., 216., 163., 127., 121.,
        122., 146., 141.,  88., 172.,  66.],
       [  0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   1.,   1.,
          1.,   0., 200., 232., 232., 233., 229., 223., 223., 215., 213.,
        164., 127., 123., 196., 229.,   0.],
       [  0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,
          0.,   0., 183., 225., 216., 223., 228., 235., 227., 224., 222.,
        224., 221., 223., 245., 173.,   0.],
       [  0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,
          0.,   0., 193., 228., 218., 213., 198., 180., 212., 210., 211.,
        213., 223., 220., 243., 202.,   0.],
       [  0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   1.,   3.,
          0.,  12., 219., 220., 212., 218., 192., 169., 227., 208., 218.,
        224., 212., 226., 197., 209.,  52.],
       [  0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   6.,
          0.,  99., 244., 222., 220., 218., 203., 198., 221., 215., 213.,
        222., 220., 245., 119., 167.,  56.],
       [  0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   4.,   0.,
          0.,  55., 236., 228., 230., 228., 240., 232., 213., 218., 223.,
        234., 217., 217., 209.,  92.,   0.],
       [  0.,   0.,   1.,   4.,   6.,   7.,   2.,   0.,   0.,   0.,   0.,
          0., 237., 226., 217., 223., 222., 219., 222., 221., 216., 223.,
        229., 215., 218., 255.,  77.,   0.],
       [  0.,   3.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,  62., 145.,
        204., 228., 207., 213., 221., 218., 208., 211., 218., 224., 223.,
        219., 215., 224., 244., 159.,   0.],
       [  0.,   0.,   0.,   0.,  18.,  44.,  82., 107., 189., 228., 220.,
        222., 217., 226., 200., 205., 211., 230., 224., 234., 176., 188.,
        250., 248., 233., 238., 215.,   0.],
       [  0.,  57., 187., 208., 224., 221., 224., 208., 204., 214., 208.,
        209., 200., 159., 245., 193., 206., 223., 255., 255., 221., 234.,
        221., 211., 220., 232., 246.,   0.],
       [  3., 202., 228., 224., 221., 211., 211., 214., 205., 205., 205.,
        220., 240.,  80., 150., 255., 229., 221., 188., 154., 191., 210.,
        204., 209., 222., 228., 225.,   0.],
       [ 98., 233., 198., 210., 222., 229., 229., 234., 249., 220., 194.,
        215., 217., 241.,  65.,  73., 106., 117., 168., 219., 221., 215.,
        217., 223., 223., 224., 229.,  29.],
       [ 75., 204., 212., 204., 193., 205., 211., 225., 216., 185., 197.,
        206., 198., 213., 240., 195., 227., 245., 239., 223., 218., 212.,
        209., 222., 220., 221., 230.,  67.],
       [ 48., 203., 183., 194., 213., 197., 185., 190., 194., 192., 202.,
        214., 219., 221., 220., 236., 225., 216., 199., 206., 186., 181.,
        177., 172., 181., 205., 206., 115.],
       [  0., 122., 219., 193., 179., 171., 183., 196., 204., 210., 213.,
        207., 211., 210., 200., 196., 194., 191., 195., 191., 198., 192.,
        176., 156., 167., 177., 210.,  92.],
       [  0.,   0.,  74., 189., 212., 191., 175., 172., 175., 181., 185.,
        188., 189., 188., 193., 198., 204., 209., 210., 210., 211., 188.,
        188., 194., 192., 216., 170.,   0.],
       [  2.,   0.,   0.,   0.,  66., 200., 222., 237., 239., 242., 246.,
        243., 244., 221., 220., 193., 191., 179., 182., 182., 181., 176.,
        166., 168.,  99.,  58.,   0.,   0.],
       [  0.,   0.,   0.,   0.,   0.,   0.,   0.,  40.,  61.,  44.,  72.,
         41.,  35.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,
          0.,   0.,   0.,   0.,   0.,   0.],
       [  0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,
          0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,
          0.,   0.,   0.,   0.,   0.,   0.],
       [  0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,
          0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,
          0.,   0.,   0.,   0.,   0.,   0.]], dtype=float32)
```

```
trainX = trainX / 255
```

```
trainX[0]
```

```
array([[0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        ],
       [0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        ],
       [0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        ],
       [0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.00392157, 0.        , 0.        ,
        0.05098039, 0.28627452, 0.        , 0.        , 0.00392157,
        0.01568628, 0.        , 0.        , 0.        , 0.        ,
        0.00392157, 0.00392157, 0.        ],
       [0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.01176471, 0.        , 0.14117648,
        0.53333336, 0.49803922, 0.24313726, 0.21176471, 0.        ,
        0.        , 0.        , 0.00392157, 0.01176471, 0.01568628,
        0.        , 0.        , 0.01176471],
       [0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.02352941, 0.        , 0.4       ,
        0.8       , 0.6901961 , 0.5254902 , 0.5647059 , 0.48235294,
        0.09019608, 0.        , 0.        , 0.        , 0.        ,
        0.04705882, 0.03921569, 0.        ],
       [0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.60784316,
        0.9254902 , 0.8117647 , 0.69803923, 0.41960785, 0.6117647 ,
        0.6313726 , 0.42745098, 0.2509804 , 0.09019608, 0.3019608 ,
        0.50980395, 0.28235295, 0.05882353],
       [0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.00392157, 0.        , 0.27058825, 0.8117647 ,
        0.8745098 , 0.85490197, 0.84705883, 0.84705883, 0.6392157 ,
        0.49803922, 0.4745098 , 0.47843137, 0.57254905, 0.5529412 ,
        0.34509805, 0.6745098 , 0.25882354],
       [0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.00392157,
        0.00392157, 0.00392157, 0.        , 0.78431374, 0.9098039 ,
        0.9098039 , 0.9137255 , 0.8980392 , 0.8745098 , 0.8745098 ,
        0.84313726, 0.8352941 , 0.6431373 , 0.49803922, 0.48235294,
        0.76862746, 0.8980392 , 0.        ],
       [0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.7176471 , 0.88235295,
        0.84705883, 0.8745098 , 0.89411765, 0.92156863, 0.8901961 ,
        0.8784314 , 0.87058824, 0.8784314 , 0.8666667 , 0.8745098 ,
        0.9607843 , 0.6784314 , 0.        ],
       [0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.75686276, 0.89411765,
        0.85490197, 0.8352941 , 0.7764706 , 0.7058824 , 0.83137256,
        0.8235294 , 0.827451  , 0.8352941 , 0.8745098 , 0.8627451 ,
        0.9529412 , 0.7921569 , 0.        ],
       [0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.00392157,
        0.01176471, 0.        , 0.04705882, 0.85882354, 0.8627451 ,
        0.83137256, 0.85490197, 0.7529412 , 0.6627451 , 0.8901961 ,
        0.8156863 , 0.85490197, 0.8784314 , 0.83137256, 0.8862745 ,
        0.77254903, 0.81960785, 0.20392157],
       [0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.        ,
        0.02352941, 0.        , 0.3882353 , 0.95686275, 0.87058824,
        0.8627451 , 0.85490197, 0.79607844, 0.7764706 , 0.8666667 ,
        0.84313726, 0.8352941 , 0.87058824, 0.8627451 , 0.9607843 ,
        0.46666667, 0.654902  , 0.21960784],
       [0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.01568628,
        0.        , 0.        , 0.21568628, 0.9254902 , 0.89411765,
        0.9019608 , 0.89411765, 0.9411765 , 0.9098039 , 0.8352941 ,
        0.85490197, 0.8745098 , 0.91764706, 0.8509804 , 0.8509804 ,
        0.81960785, 0.36078432, 0.        ],
       [0.        , 0.        , 0.00392157, 0.01568628, 0.02352941,
        0.02745098, 0.00784314, 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.92941177, 0.8862745 , 0.8509804 ,
        0.8745098 , 0.87058824, 0.85882354, 0.87058824, 0.8666667 ,
        0.84705883, 0.8745098 , 0.8980392 , 0.84313726, 0.85490197,
        1.        , 0.3019608 , 0.        ],
       [0.        , 0.01176471, 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.24313726,
        0.5686275 , 0.8       , 0.89411765, 0.8117647 , 0.8352941 ,
        0.8666667 , 0.85490197, 0.8156863 , 0.827451  , 0.85490197,
        0.8784314 , 0.8745098 , 0.85882354, 0.84313726, 0.8784314 ,
        0.95686275, 0.62352943, 0.        ],
       [0.        , 0.        , 0.        , 0.        , 0.07058824,
        0.17254902, 0.32156864, 0.41960785, 0.7411765 , 0.89411765,
        0.8627451 , 0.87058824, 0.8509804 , 0.8862745 , 0.78431374,
        0.8039216 , 0.827451  , 0.9019608 , 0.8784314 , 0.91764706,
        0.6901961 , 0.7372549 , 0.98039216, 0.972549  , 0.9137255 ,
        0.93333334, 0.84313726, 0.        ],
       [0.        , 0.22352941, 0.73333335, 0.8156863 , 0.8784314 ,
        0.8666667 , 0.8784314 , 0.8156863 , 0.8       , 0.8392157 ,
        0.8156863 , 0.81960785, 0.78431374, 0.62352943, 0.9607843 ,
        0.75686276, 0.80784315, 0.8745098 , 1.        , 1.        ,
        0.8666667 , 0.91764706, 0.8666667 , 0.827451  , 0.8627451 ,
        0.9098039 , 0.9647059 , 0.        ],
       [0.01176471, 0.7921569 , 0.89411765, 0.8784314 , 0.8666667 ,
        0.827451  , 0.827451  , 0.8392157 , 0.8039216 , 0.8039216 ,
        0.8039216 , 0.8627451 , 0.9411765 , 0.3137255 , 0.5882353 ,
        1.        , 0.8980392 , 0.8666667 , 0.7372549 , 0.6039216 ,
        0.7490196 , 0.8235294 , 0.8       , 0.81960785, 0.87058824,
        0.89411765, 0.88235295, 0.        ],
       [0.38431373, 0.9137255 , 0.7764706 , 0.8235294 , 0.87058824,
        0.8980392 , 0.8980392 , 0.91764706, 0.9764706 , 0.8627451 ,
        0.7607843 , 0.84313726, 0.8509804 , 0.94509804, 0.25490198,
        0.28627452, 0.41568628, 0.45882353, 0.65882355, 0.85882354,
        0.8666667 , 0.84313726, 0.8509804 , 0.8745098 , 0.8745098 ,
        0.8784314 , 0.8980392 , 0.11372549],
       [0.29411766, 0.8       , 0.83137256, 0.8       , 0.75686276,
        0.8039216 , 0.827451  , 0.88235295, 0.84705883, 0.7254902 ,
        0.77254903, 0.80784315, 0.7764706 , 0.8352941 , 0.9411765 ,
        0.7647059 , 0.8901961 , 0.9607843 , 0.9372549 , 0.8745098 ,
        0.85490197, 0.83137256, 0.81960785, 0.87058824, 0.8627451 ,
        0.8666667 , 0.9019608 , 0.2627451 ],
       [0.1882353 , 0.79607844, 0.7176471 , 0.7607843 , 0.8352941 ,
        0.77254903, 0.7254902 , 0.74509805, 0.7607843 , 0.7529412 ,
        0.7921569 , 0.8392157 , 0.85882354, 0.8666667 , 0.8627451 ,
        0.9254902 , 0.88235295, 0.84705883, 0.78039217, 0.80784315,
        0.7294118 , 0.70980394, 0.69411767, 0.6745098 , 0.70980394,
        0.8039216 , 0.80784315, 0.4509804 ],
       [0.        , 0.47843137, 0.85882354, 0.75686276, 0.7019608 ,
        0.67058825, 0.7176471 , 0.76862746, 0.8       , 0.8235294 ,
        0.8352941 , 0.8117647 , 0.827451  , 0.8235294 , 0.78431374,
        0.76862746, 0.7607843 , 0.7490196 , 0.7647059 , 0.7490196 ,
        0.7764706 , 0.7529412 , 0.6901961 , 0.6117647 , 0.654902  ,
        0.69411767, 0.8235294 , 0.36078432],
       [0.        , 0.        , 0.2901961 , 0.7411765 , 0.83137256,
        0.7490196 , 0.6862745 , 0.6745098 , 0.6862745 , 0.70980394,
        0.7254902 , 0.7372549 , 0.7411765 , 0.7372549 , 0.75686276,
        0.7764706 , 0.8       , 0.81960785, 0.8235294 , 0.8235294 ,
        0.827451  , 0.7372549 , 0.7372549 , 0.7607843 , 0.7529412 ,
        0.84705883, 0.6666667 , 0.        ],
       [0.00784314, 0.        , 0.        , 0.        , 0.25882354,
        0.78431374, 0.87058824, 0.92941177, 0.9372549 , 0.9490196 ,
        0.9647059 , 0.9529412 , 0.95686275, 0.8666667 , 0.8627451 ,
        0.75686276, 0.7490196 , 0.7019608 , 0.7137255 , 0.7137255 ,
        0.70980394, 0.6901961 , 0.6509804 , 0.65882355, 0.3882353 ,
        0.22745098, 0.        , 0.        ],
       [0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.15686275, 0.23921569, 0.17254902,
        0.28235295, 0.16078432, 0.13725491, 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        ],
       [0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        ],
       [0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.        ,
```

```
testX = testX / 255
```

We have total of 10 classes in target variables

```
trainY = tf.keras.utils.to_categorical(trainY, num_classes=10)
testY = tf.keras.utils.to_categorical(testY, num_classes=10)
```

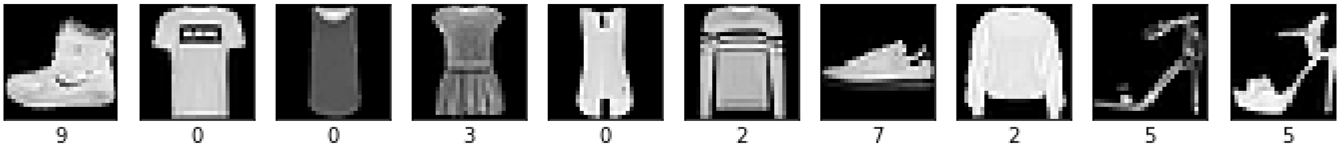
```
trainY[0]
```

```
array([0., 0., 0., 0., 0., 0., 0., 0., 0., 1.], dtype=float32)
```

```
np.argmax(trainY[0])
```

```
9
```

```
from matplotlib import pyplot as plt
plt.figure(figsize=(12,12))
for i in range(10):
  plt.subplot(1,10,i+1)
  plt.xticks([])
  plt.yticks([])
  plt.grid(False)
  plt.imshow(trainX[i],cmap='gray')
  plt.xlabel(np.argmax(trainY[i]))
plt.show()
```



Build the Graph

Initialize model, reshape & normalize data

```
#Clear out model from current memory
tf.keras.backend.clear_session()

#Initialize Sequential model
model = tf.keras.models.Sequential()

#Reshape data from 2D to 1D -> 28x28 to 784
model.add(tf.keras.layers.Reshape((784,),input_shape=(28,28,)))

#Normalize the data
model.add(tf.keras.layers.BatchNormalization())
```

Add two fully connected layers with 200 and 100 neurons respectively with `relu` activatio

```
#Add 1st hidden layer
model.add(tf.keras.layers.Dense(200, activation='relu'))
model.add(tf.keras.layers.BatchNormalization())
# Dropout
model.add(tf.keras.layers.Dropout(0.25))

#Add 2nd hidden layer
model.add(tf.keras.layers.Dense(100,  activation='relu'))
model.add(tf.keras.layers.BatchNormalization())
model.add(tf.keras.layers.Dropout(0.25))
```

Add the output layer with a fully connected layer with 10 neurons with `softmax` ac `categorical_crossentropy` loss and `adam` optimizer and train the network. And, r

```
#Add OUTPUT layer
model.add(tf.keras.layers.Dense(10, kernel_initializer='he_normal',
                                activation='softmax'))
#Compile the model
model.compile(optimizer='adam', loss='categorical_crossentropy',
              metrics=['accuracy'])
```

```
from datetime import datetime
logdir = "logs/scalars/" + datetime.now().strftime("%Y%m%d-%H%M%S")
tensorboard_callback = tf.keras.callbacks.TensorBoard(log_dir=logdir)
```

```
model.summary()
```

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
reshape (Reshape)            (None, 784)               0         
_________________________________________________________________
batch_normalization (BatchNo (None, 784)               3136      
_________________________________________________________________
dense (Dense)                (None, 200)               157000    
_________________________________________________________________
batch_normalization_1 (Batch (None, 200)               800       
_________________________________________________________________
dropout (Dropout)            (None, 200)               0         
_________________________________________________________________
dense_1 (Dense)              (None, 100)               20100     
_________________________________________________________________
batch_normalization_2 (Batch (None, 100)               400       
_________________________________________________________________
dropout_1 (Dropout)          (None, 100)               0         
_________________________________________________________________
dense_2 (Dense)              (None, 10)                1010      
=================================================================
Total params: 182,446
Trainable params: 180,278
Non-trainable params: 2,168
_________________________________________________________________
```

```
#Modelcheckpoint callback
ckpt = tf.keras.callbacks.ModelCheckpoint('mnist_v1.hdf5', save_best_only=True,
                                          monitor='val_loss', mode='min')


model.fit(trainX,trainY,
          validation_data=(testX,testY),
          epochs=10,
          batch_size=32, callbacks=[ckpt,tensorboard_callback])
```

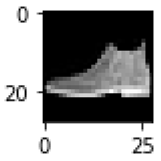
```
Train on 60000 samples, validate on 10000 samples
Epoch 1/10
60000/60000 [==============================] - 15s 254us/sample - loss: 0.5566 - accurac
Epoch 2/10
60000/60000 [==============================] - 13s 215us/sample - loss: 0.4295 - accurac
Epoch 3/10
60000/60000 [==============================] - 14s 230us/sample - loss: 0.3919 - accurac
Epoch 4/10
60000/60000 [==============================] - 14s 228us/sample - loss: 0.3737 - accurac
Epoch 5/10
60000/60000 [==============================] - 14s 228us/sample - loss: 0.3569 - accurac
Epoch 6/10
60000/60000 [==============================] - 17s 277us/sample - loss: 0.3374 - accurac
Epoch 7/10
60000/60000 [==============================] - 16s 272us/sample - loss: 0.3278 - accurac
Epoch 8/10
60000/60000 [==============================] - 16s 270us/sample - loss: 0.3225 - accurac
Epoch 9/10
60000/60000 [==============================] - 17s 292us/sample - loss: 0.3114 - accurac
Epoch 10/10
60000/60000 [==============================] - 16s 274us/sample - loss: 0.3032 - accurac
<tensorflow.python.keras.callbacks.History at 0x7f0c9df5e940>
```

```
y_pred = model.predict(testX)
print(y_pred)
```

```
[[1.1615418e-06 1.0253641e-06 1.7382665e-06 ... 1.0170623e-01
  2.9786319e-05 8.7610346e-01]
 [8.9308203e-05 3.5240062e-09 9.9870336e-01 ... 7.1629977e-08
  9.9528403e-09 5.6840744e-08]
 [1.5153137e-06 9.9999607e-01 1.6575723e-07 ... 7.7957178e-09
  2.5276262e-07 5.2019207e-09]
 ...
 [1.4396167e-03 4.4202224e-07 3.3199985e-04 ... 1.1025061e-05
  9.9482441e-01 3.1092497e-07]
 [1.7323366e-06 9.9984884e-01 2.2473982e-07 ... 1.6708621e-07
  6.4237577e-07 4.7100551e-07]
 [5.6212690e-07 5.7720405e-07 6.5057570e-06 ... 3.6024649e-02
  6.8448817e-06 5.0523122e-05]]
```

```
#Lets print the image as well
import matplotlib.pyplot as plt
plt.figure(figsize=(1,1))
plt.imshow(testX[0],cmap='gray')
```

```
<matplotlib.image.AxesImage at 0x7f0c8dc067f0>
```



```
np.argmax(testY[0])
```

```
9
```

```
np.argmax(y_pred[0])
```

```
9
```