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

Load tensorflow

```
from __future__ import absolute_import, division, print_function, unicode_literals
import tensorflow as tf
from matplotlib import pyplot as plt
import numpy as np
```

```
import tensorflow.compat.v1 as tf1
tf1.disable_v2_behavior()
```

Collect Fashion mnist data from tf.keras.datasets

```
tf.keras.datasets.fashion_mnist.load_data()
```

```
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/train-
32768/29515 [================================] - 0s 0us/step
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/train-
26427392/26421880 [==============================] - 0s 0us/step
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/t10k-l
8192/5148 [===============================================] - 0s 0us/step
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/t10k-i
4423680/4422102 [==============================] - 0s 0us/step
((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, 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, 0, ..., 0, 0, 0],
          ...,
          [0, 0, 0, ..., 0, 0, 0],
          [0, 0, 0, ..., 0, 0, 0],
          [0, 0, 0, ..., 0, 0, 0]]], dtype=uint8),
```

```
 array([9, 0, 0, ..., 3, 0, 5], dtype=uint8)),
(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, 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, 0, ..., 0, 0, 0],
         ...,
         [0, 0, 0, ..., 0, 0, 0],
         [0, 0, 0, ..., 0, 0, 0],
         [0, 0, 0, ..., 0, 0, 0]]], dtype=uint8),
```

bold text#### Change train and test labels into one-hot vectors

```
import numpy as np
from keras.utils import np_utils
from keras.preprocessing.image import ImageDataGenerator
```

```
x_train = x_train.reshape(x_train.shape[0], 28, 28, 1).astype('float32')
x_test = x_test.reshape(x_test.shape[0], 28, 28, 1).astype('float32')
```

```
x_train.shape
```

```
(60000, 28, 28, 1)
```

```
x_test.shape
```

```
(10000, 28, 28, 1)
```

```
y_train = np_utils.to_categorical(y_train, 10)
y_test = np_utils.to_categorical(y_test, 10)
```

```
y_train.shape
```

```
(60000, 10)
```

```
y_train[0].shape
```

```
(10,)
```

```
y_train[0]
```

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

```
y_test.shape
```

```
(10000, 10)
```

```
y_test[0].shape
```

```
(10,)
```

```
y_test[0]
```

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

```
x_train.shape
```

```
(60000, 28, 28, 1)
```

Build the Graph

Initialize model, reshape & normalize data

```
import keras
from keras.models import Sequential
from keras.layers import Dense, Activation, Dropout, Flatten, Reshape
from keras.layers import Convolution2D, MaxPooling2D

x_train=x_train.astype("float32")


x_test=x_test.astype("float32")


x_train /= 200
x_test /= 200


x_train[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.   ],
       [0.005],
       [0.   ],
       [0.   ],
       [0.065],
       [0.365],
       [0.   ],
       [0.   ],
       [0.005],
       [0.02 ],
       [0.   ],
       [0.   ],
       [0.   ],
       [0.   ],
       [0.005],
       [0.005],
       [0.   ]],

      [[0.   ],
       [0.   ],
       [0.   ],
       [0.   ],
       [0.   ],
       [0.   ],
       [0.   ],
       [0.   ],
       [0.   ],
```

```
       [0.   ],
       [0.   ],
       [0.   ],
       [0.   ],
       [0.015],
       [0.   ],
       [0.18 ],
       [0.68 ],
       [0.635],
       [0.31 ],
       [0.27 ],
       [0.   ],
       [0.   ],
       [0.   ],
       [0.005],
       [0.015],
       [0.02 ],
       [0.   ],
       [0.   ],
       [0.015]],

      [[0.   ],
       [0.   ],
       [0.   ],
       [0.   ],
       [0.   ],
       [0.   ],
       [0.   ],
       [0.   ],
       [0.   ],
       [0.   ],
       [0.   ],
       [0.   ],
       [0.   ],
       [0.03 ],
       [0.   ],
       [0.51 ],
       [1.02 ],
       [0.88 ],
       [0.67 ],
       [0.72 ],
       [0.615],
       [0.115],
       [0.   ],
       [0.   ],
       [0.   ],
       [0.   ],
       [0.06 ],
       [0.05 ],
       [0.   ]],

      [[0.   ],
       [0.   ],
       [0.   ],
       [0.   ],
       [0.   ],
       [0.   ],
       [0.   ],
       [0.   ],
```

```
       [0.   ],
       [0.   ],
       [0.   ],
       [0.   ],
       [0.   ],
       [0.   ],
       [0.775],
       [1.18 ],
       [1.035],
       [0.89 ],
       [0.535],
       [0.78 ],
       [0.805],
       [0.545],
       [0.32 ],
       [0.115],
       [0.385],
       [0.65 ],
       [0.36 ],
       [0.075]],

      [[0.   ],
       [0.   ],
       [0.   ],
       [0.   ],
       [0.   ],
       [0.   ],
       [0.   ],
       [0.   ],
       [0.   ],
       [0.   ],
       [0.   ],
       [0.   ],
       [0.005],
       [0.   ],
       [0.345],
       [1.035],
       [1.115],
       [1.09 ],
       [1.08 ],
       [1.08 ],
       [0.815],
       [0.635],
       [0.605],
       [0.61 ],
       [0.73 ],
       [0.705],
       [0.44 ],
       [0.86 ],
       [0.33 ]],

      [[0.   ],
       [0.   ],
       [0.   ],
       [0.   ],
       [0.   ],
       [0.   ],
       [0.   ],
       [0.   ],
       [0.   ],
```