KerasMultiClass_MNIST_Fashion

```
#import keras
import numpy as np
import tensorflow as tf
from tensorflow import keras
from keras.layers import Dense, Activation, Dropout, Flatten
from keras.models import Sequential
from keras.optimizers import Adam
```

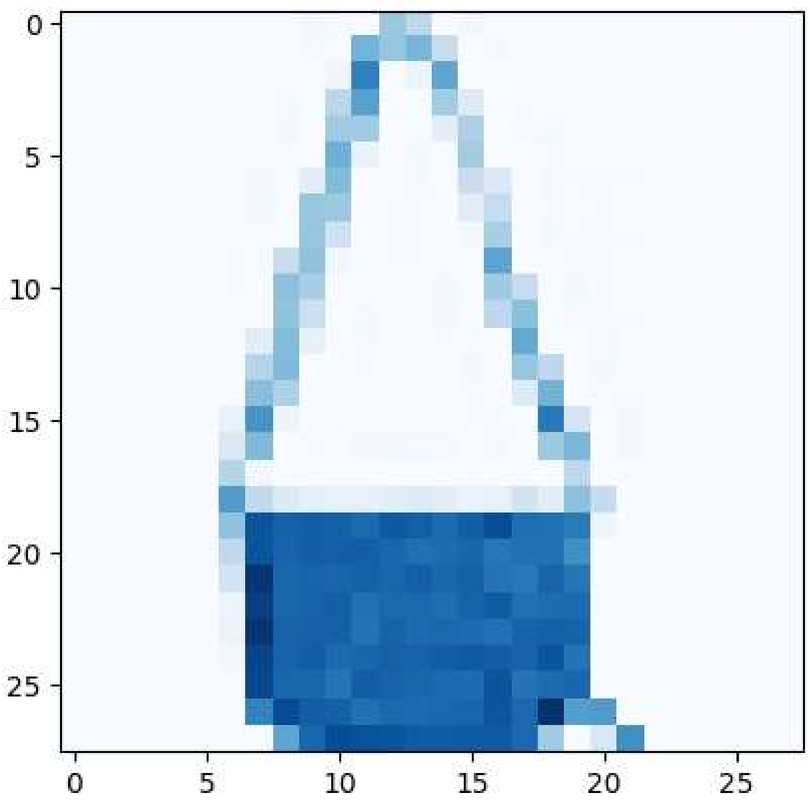
```
(x_train, y_train), (x_test, y_test) = tf.keras.datasets.fashion_mnist.load_data()
```

```
print(x_train.shape)
print(x_test.shape)
print(y_train)
```

```
(60000, 28, 28)
(10000, 28, 28)
[9 0 0 ... 3 0 5]
```

```
import matplotlib.pyplot as plt
image_index = 1248
print(y_train[image_index])
plt.imshow(x_train[image_index], cmap = 'Blues')
plt.show()
```

```
8
```



```
model = Sequential([
    Flatten(input_shape=(28,28)),
    Dense(2048, activation = 'relu'),
    Dense(1024, activation = 'relu'),
    Dense(1024, activation = 'relu'),
    Dense(512, activation = 'relu'),
    Dense(10, activation = 'softmax'),
])
```

```
model.summary()
```

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #   
=================================================================
 flatten_1 (Flatten)         (None, 784)               0         
                                                                 
 dense_5 (Dense)             (None, 2048)              1607680   
                                                                 
 dense_6 (Dense)             (None, 1024)              2098176   
                                                                 
 dense_7 (Dense)             (None, 1024)              1049600   
                                                                 
 dense_8 (Dense)             (None, 512)               524800    
                                                                 
 dense_9 (Dense)             (None, 10)                5130      
                                                                 
=================================================================
Total params: 5285386 (20.16 MB)
Trainable params: 5285386 (20.16 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

```
model.compile(optimizer = 'adam', loss = 'sparse_categorical_crossentropy', metrics = ['a
```

```
history = model.fit(x_train, y_train, epochs = 25, verbose = 2)
print(history.epoch, history.history['accuracy'][-1])
```

```
Epoch 1/25
1875/1875 - 6s - loss: 0.2475 - accuracy: 0.9105 - 6s/epoch - 3ms/step
Epoch 2/25
1875/1875 - 6s - loss: 0.2617 - accuracy: 0.9076 - 6s/epoch - 3ms/step
Epoch 3/25
1875/1875 - 5s - loss: 0.2470 - accuracy: 0.9115 - 5s/epoch - 3ms/step
Epoch 4/25
1875/1875 - 6s - loss: 0.2444 - accuracy: 0.9126 - 6s/epoch - 3ms/step
Epoch 5/25
1875/1875 - 6s - loss: 0.2481 - accuracy: 0.9109 - 6s/epoch - 3ms/step
Epoch 6/25
1875/1875 - 6s - loss: 0.2421 - accuracy: 0.9127 - 6s/epoch - 3ms/step
Epoch 7/25
1875/1875 - 5s - loss: 0.2613 - accuracy: 0.9104 - 5s/epoch - 3ms/step
Epoch 8/25
1875/1875 - 6s - loss: 0.2397 - accuracy: 0.9158 - 6s/epoch - 3ms/step
Epoch 9/25
```

```
1875/1875 - 6s - loss: 0.2315 - accuracy: 0.9167 - 6s/epoch - 3ms/step
Epoch 10/25
1875/1875 - 6s - loss: 0.2549 - accuracy: 0.9135 - 6s/epoch - 3ms/step
Epoch 11/25
1875/1875 - 6s - loss: 0.2432 - accuracy: 0.9162 - 6s/epoch - 3ms/step
Epoch 12/25
1875/1875 - 6s - loss: 0.2255 - accuracy: 0.9191 - 6s/epoch - 3ms/step
Epoch 13/25
1875/1875 - 6s - loss: 0.2285 - accuracy: 0.9182 - 6s/epoch - 3ms/step
Epoch 14/25
1875/1875 - 5s - loss: 0.2257 - accuracy: 0.9184 - 5s/epoch - 3ms/step
Epoch 15/25
1875/1875 - 6s - loss: 0.2320 - accuracy: 0.9173 - 6s/epoch - 3ms/step
Epoch 16/25
1875/1875 - 6s - loss: 0.2205 - accuracy: 0.9214 - 6s/epoch - 3ms/step
Epoch 17/25
1875/1875 - 6s - loss: 0.2510 - accuracy: 0.9160 - 6s/epoch - 3ms/step
Epoch 18/25
1875/1875 - 5s - loss: 0.2150 - accuracy: 0.9220 - 5s/epoch - 3ms/step
Epoch 19/25
1875/1875 - 6s - loss: 0.2153 - accuracy: 0.9215 - 6s/epoch - 3ms/step
Epoch 20/25
1875/1875 - 6s - loss: 0.2656 - accuracy: 0.9105 - 6s/epoch - 3ms/step
Epoch 21/25
1875/1875 - 5s - loss: 0.2111 - accuracy: 0.9244 - 5s/epoch - 3ms/step
Epoch 22/25
1875/1875 - 6s - loss: 0.2209 - accuracy: 0.9217 - 6s/epoch - 3ms/step
Epoch 23/25
1875/1875 - 5s - loss: 0.2091 - accuracy: 0.9249 - 5s/epoch - 3ms/step
Epoch 24/25
1875/1875 - 6s - loss: 0.2151 - accuracy: 0.9241 - 6s/epoch - 3ms/step
Epoch 25/25
1875/1875 - 5s - loss: 0.4296 - accuracy: 0.9174 - 5s/epoch - 3ms/step
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23
```

```
model.evaluate(x_test, y_test)
```

```
313/313 [==============================] - 1s 2ms/step - loss: 0.5325 - accuracy: 0.8
[0.5325161814689636, 0.8888000249862671]
```