Kaggle v_2

November 5, 2020

0.0.1 Reference: https://www.tensorflow.org/tutorials/keras/classification

[1]:

```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
import matplotlib.pyplot as plt
import tensorflow as tf
from tensorflow import keras
```

[2]:

```
data_train = pd.read_csv('./data/fashion-mnist_train.csv')
data_test = pd.read_csv('./data/fashion-mnist_test.csv')
```

[41]:

```
X_train = data_train.values[:, 1:]
Y_train = data_train.values[:, 0]
X_test = data_test.values[:, 1:]
Y_test = data_test.values[:, 0]

X_train, X_val, Y_train, Y_val = train_test_split(X_train, Y_train, test_size=0.
 ↪2, random_state=0)
```

[42]:

```
X_train = X_train.reshape(X_train.shape[0], 28, 28)
X_test = X_test.reshape(X_test.shape[0], 28, 28)
X_val = X_val.reshape(X_val.shape[0], 28, 28)
```

[43]:

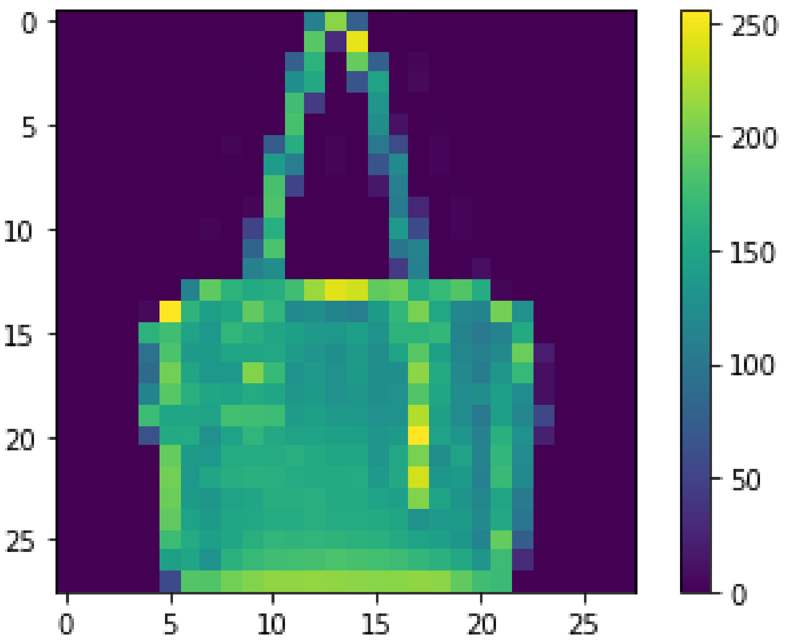
```
X_train.shape
```

[43]:

```
(48000, 28, 28)
```

[44]:

```
plt.figure()
plt.imshow(X_train[80])
plt.colorbar()
plt.grid(False)
plt.show()
```



```
[45]: X_train, X_val, X_test = X_train.astype('float32'), X_val.astype('float32'), X_test.astype('float32'),
      X_train /= 255
      X_val /= 255
      X_test /= 255
```

```
[46]: class_names = ['T-shirt/top', 'Trouser', 'Pullover', 'Dress', 'Coat',
                     'Sandal', 'Shirt', 'Sneaker', 'Bag', 'Ankle boot']
      plt.figure(figsize=(10,10))
      for i in range(25):
          plt.subplot(5,5,i+1)
          plt.xticks([])
          plt.yticks([])
          plt.grid(False)
          plt.imshow(X_train[i], cmap=plt.cm.binary)
          plt.xlabel(class_names[Y_train[i]])
      plt.show()
```

[47]:

```
model = keras.Sequential([
    keras.layers.Flatten(input_shape=(28, 28)),
    keras.layers.Dense(128, activation='relu'),
    keras.layers.Dense(10, activation='softmax')
])
```

[48]:

```
model.compile(optimizer='adam',
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])
```

[50]:

```
model.fit(X_train, Y_train, epochs=10)
```

```
Epoch 1/10
```

```
1500/1500 [==============================] - 1s 991us/step - loss: 0.5213 -
accuracy: 0.8164
Epoch 2/10
1500/1500 [==============================] - 2s 1ms/step - loss: 0.3903 -
accuracy: 0.8593
Epoch 3/10
1500/1500 [==============================] - 3s 2ms/step - loss: 0.3498 -
accuracy: 0.8731
Epoch 4/10
1500/1500 [==============================] - 2s 1ms/step - loss: 0.3229 -
accuracy: 0.8830
Epoch 5/10
1500/1500 [==============================] - 1s 605us/step - loss: 0.3069 -
accuracy: 0.8864
Epoch 6/10
1500/1500 [==============================] - 1s 746us/step - loss: 0.2888 -
accuracy: 0.8931
Epoch 7/10
1500/1500 [==============================] - 2s 1ms/step - loss: 0.2762 -
accuracy: 0.8975
Epoch 8/10
1500/1500 [==============================] - 1s 764us/step - loss: 0.2658 -
accuracy: 0.9018
Epoch 9/10
1500/1500 [==============================] - 1s 633us/step - loss: 0.2551 -
accuracy: 0.9050
Epoch 10/10
1500/1500 [==============================] - 2s 1ms/step - loss: 0.2467 -
accuracy: 0.9074
```

[50]:

```
<tensorflow.python.keras.callbacks.History at 0x7f462a1c8e90>
```

[52]:

```python
val_loss, val_acc = model.evaluate(X_val, Y_val, verbose=2)
test_loss, test_acc = model.evaluate(X_test, Y_test, verbose=2)
print('accuracy on validation:', val_acc)
print('accuracy on test:', test_acc)
```

```
375/375 - 0s - loss: 0.3129 - accuracy: 0.8864
313/313 - 0s - loss: 0.3068 - accuracy: 0.8888
accuracy on validation: 0.8864166736602783
accuracy on test: 0.8888000249862671
```