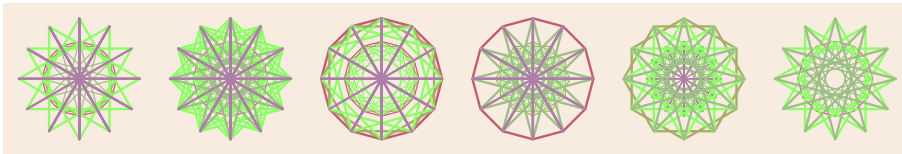


Example

Leon Tabak

06 January 2022

This work is licensed under CC BY 4.0. To view a copy of this license, visit <http://creativecommons.org/licenses/by/4.0/>.



```
1 import tensorflow as tf
2 from tensorflow import keras
3
4 def main():
5     # Keras comes with functions for fetching
6     # some widely used datasets
7
8     fashion_mnist = keras.datasets.fashion_mnist
9
10    # fetch training and test sets
11
12    # dataset is already divided into a training
13    # set and a test set
14
15    (X_train_full, y_train_full), (X_test, y_test) = \
16        fashion_mnist.load_data()
17
18    print( '\n **Size of MNIST Fashion training set** \n' )
19    print( X_train_full.shape )
20
21    print( '\n **Type of elements in MNIST Fashion dataset** \n' )
```

```

22     print( X_train_full.dtype )
23
24     # make validation set
25
26     # scale data—instead of integers in
27     # the interval [0, 255] represent pixel
28     # brightness with floating point values
29     # between 0.0 and 1.0
30
31     # validate with first 5000 records
32
33     X_validate = X_train_full[:5000]/255.0
34     y_validate = y_train_full[:5000]
35
36     # train with all remaining records
37     # in the full training set
38
39     X_train = X_train_full[5000:]/255.0
40     y_train = y_train_full[5000:]
41
42     # dataset contains images of 10 kinds of clothing
43
44     types_of_clothing = [
45         'T-shirt/top',
46         'Trouser',
47         'Pullover',
48         'Dress',
49         'Coat',
50         'Sandal',
51         'Shirt',
52         'Sneaker',
53         'Bag',
54         'Ankle boot'
55     ]
56
57     model = keras.models.Sequential([
58         keras.layers.Flatten( input_shape = [28, 28] ),
59         keras.layers.Dense( 300, activation = 'relu' ),
60         keras.layers.Dense( 100, activation = 'relu' ),
61         keras.layers.Dense( 10, activation = 'softmax' )
62     ])
63
64
65     print( '\n **Model summary** \n' )
66     print( model.summary() )
67

```

```

68     model.compile(
69         loss = 'sparse_categorical_crossentropy',
70         optimizer = 'sgd',
71         metrics = [ 'accuracy' ]
72     )
73
74     print( '\n **Fit the model** \n' )
75
76     history = model.fit(
77         X_train,
78         y_train,
79         epochs = 30,
80         validation_data = (X_validate, y_validate))
81
82     print( '\n **Evaluate model** \n' )
83     model.evaluate( X_test, y_test )
84
85     X_new = X_test[:3]
86     y_probabilities = model.predict( X_new )
87
88     print( '\n **Prediction probabilities** \n' )
89     print( y_probabilities.round(2) )
90
91     y_predictions = model.predict_classes( X_new )
92
93     print( '\n **Predictions** \n' )
94     print( y_predictions )
95
96
97     for i in y_predictions:
98         print( types_of_clothing[i] )
99
100 # end of main()
101
102 if __name__ == '__main__':
103     main()

```