Task 1: ANN

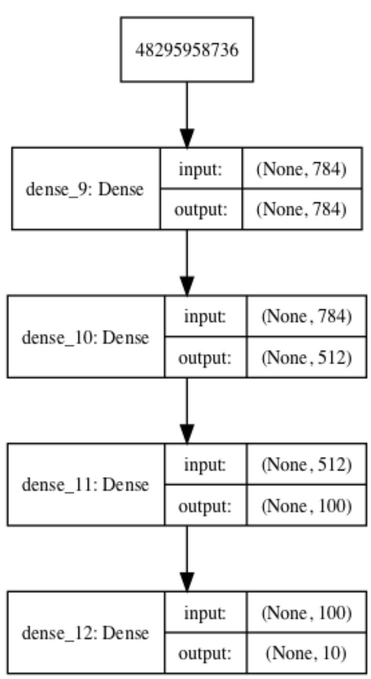
```
model = models.Sequential()
model.add(layers.Dense(784, activation = "tanh",input_shape=(784, )))
model.add(layers.Dense(512, activation = "sigmoid"))
model.add(layers.Dense(100, activation = "relu"))
model.add(layers.Dense(10, activation = "softmax"))
model.summary()
from keras.utils import plot_model
plot_model(model, to_file='model_plot.png', show_shapes=True, show_layer_names=True)
from IPython.display import Image,SVG
Image(filename='model_plot.png')
```

Layer (type)	Output Shape	Param #
dense_9 (Dense)	(None, 784)	615440
dense_10 (Dense)	(None, 512)	401920
dense_11 (Dense)	(None, 100)	51300
dense_12 (Dense)	(None, 10)	1010

Total params: 1,069,670

Trainable params: 1,069,670

Non-trainable params: 0



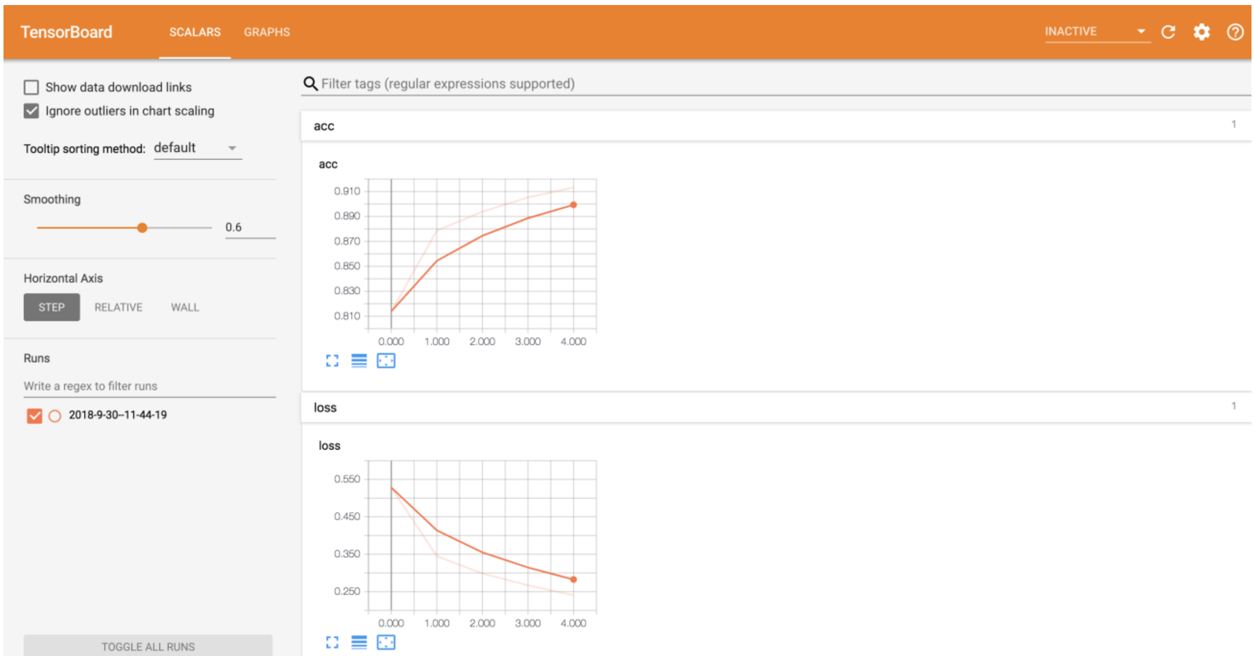
```
import time
t = time.localtime(time.time())
timeStamp = str(t.tm_year) + '-' + str(t.tm_mon) + '-' + str(t.tm_mday) + '--' + str(t.tm_hour) + '-'+str(t.tm_min) + '-
'+str(t.tm_sec)
timeStamp
tBoard = TensorBoard(log_dir='Graph/{}'.format(timeStamp))
model.compile(
 optimizer = "adam",
 loss = "categorical_crossentropy",
 #loss = "sparse_categorical_crossentropy",
 metrics = ["accuracy"]
)
result = model.fit(train_x,train_y,
                batch_size=minibatch,
                epochs=epochs,
                validation_split = 0.2,
                shuffle=True,
                callbacks=[tBoard])
```

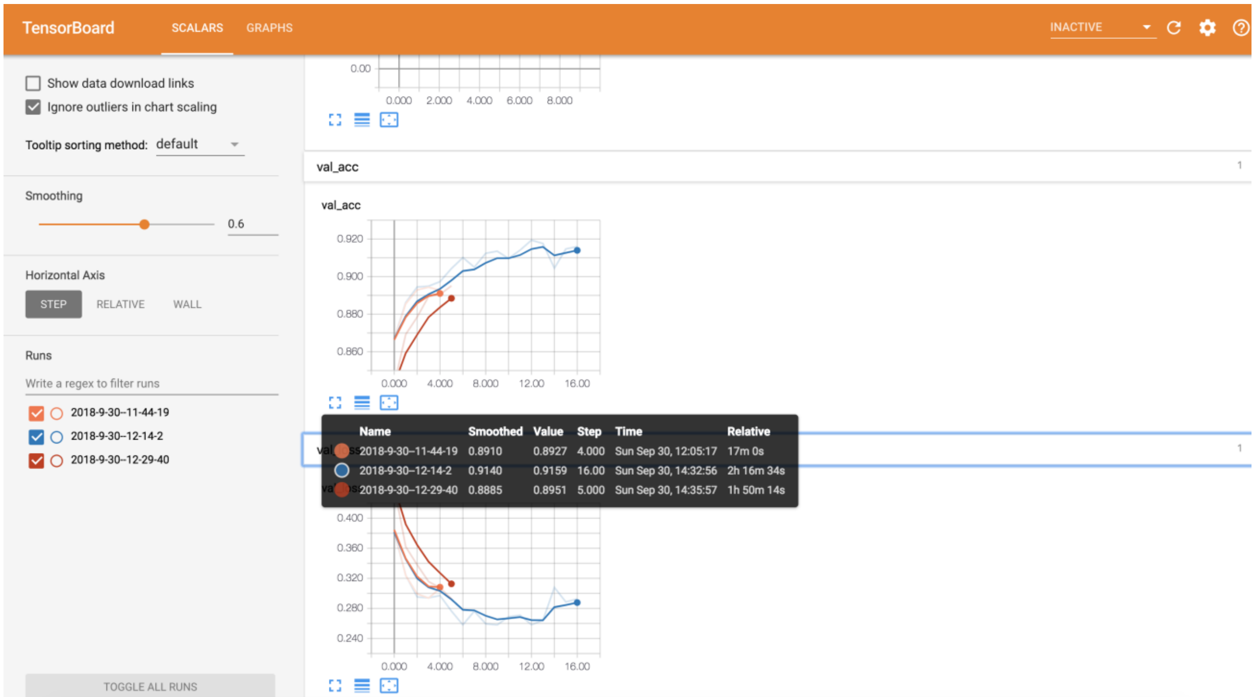
```
Train on 38400 samples, validate on 9600 samples
Epoch 1/50
38400/38400 [==============================] - 8s 215us/step - loss: 0.593
7 - acc: 0.7877 - val_loss: 0.4671 - val_acc: 0.8307
Epoch 2/50
38400/38400 [==============================] - 7s 189us/step - loss: 0.405
7 - acc: 0.8526 - val_loss: 0.3901 - val_acc: 0.8619
Epoch 3/50
38400/38400 [==============================] - 8s 204us/step - loss: 0.370
4 - acc: 0.8646 - val_loss: 0.3831 - val_acc: 0.8645
Epoch 4/50
38400/38400 [==============================] - 7s 195us/step - loss: 0.349
0 - acc: 0.8705 - val_loss: 0.3844 - val_acc: 0.8609
Epoch 5/50
38400/38400 [==============================] - 8s 207us/step - loss: 0.325
9 - acc: 0.8789 - val_loss: 0.3492 - val_acc: 0.8716
Epoch 6/50
38400/38400 [==============================] - 8s 196us/step - loss: 0.309
4 - acc: 0.8846 - val_loss: 0.3421 - val_acc: 0.8758
Epoch 7/50
38400/38400 [==============================] - 8s 205us/step - loss: 0.290
1 - acc: 0.8921 - val_loss: 0.3153 - val_acc: 0.8855
Epoch 8/50
38400/38400 [==============================] - 7s 194us/step - loss: 0.278
4 - acc: 0.8956 - val_loss: 0.3361 - val_acc: 0.8807
Epoch 9/50
38400/38400 [==============================] - 8s 200us/step - loss: 0.267
9 - acc: 0.8997 - val_loss: 0.3401 - val_acc: 0.8761
Epoch 10/50
38400/38400 [==============================] - 7s 193us/step - loss: 0.254
5 - acc: 0.9046 - val_loss: 0.3103 - val_acc: 0.8886
Epoch 11/50
38400/38400 [==============================] - 8s 202us/step - loss: 0.240
9 - acc: 0.9089 - val_loss: 0.3154 - val_acc: 0.8872
Epoch 12/50
```

```
38400/38400 [==============================] - 7s 195us/step - loss: 0.232
9 - acc: 0.9135 - val_loss: 0.3214 - val_acc: 0.8844
Epoch 13/50
38400/38400 [==============================] - 7s 194us/step - loss: 0.225
3 - acc: 0.9147 - val_loss: 0.3078 - val_acc: 0.8918
Epoch 14/50
38400/38400 [==============================] - 8s 198us/step - loss: 0.211
0 - acc: 0.9200 - val_loss: 0.3090 - val_acc: 0.8870
Epoch 15/50
38400/38400 [==============================] - 7s 193us/step - loss: 0.200
7 - acc: 0.9241 - val_loss: 0.3220 - val_acc: 0.8898
Epoch 16/50
38400/38400 [==============================] - 8s 199us/step - loss: 0.198
6 - acc: 0.9255 - val_loss: 0.3096 - val_acc: 0.8872
Epoch 17/50
38400/38400 [==============================] - 7s 193us/step - loss: 0.184
2 - acc: 0.9308 - val_loss: 0.3043 - val_acc: 0.9008
Epoch 18/50
38400/38400 [==============================] - 7s 195us/step - loss: 0.175
9 - acc: 0.9336 - val_loss: 0.3160 - val_acc: 0.8894
Epoch 19/50
38400/38400 [==============================] - 8s 198us/step - loss: 0.176
2 - acc: 0.9335 - val_loss: 0.3714 - val_acc: 0.8829
Epoch 20/50
38400/38400 [==============================] - 8s 197us/step - loss: 0.160
7 - acc: 0.9395 - val_loss: 0.3260 - val_acc: 0.8921
Epoch 21/50
38400/38400 [==============================] - 7s 191us/step - loss: 0.155
0 - acc: 0.9410 - val_loss: 0.3360 - val_acc: 0.8879
Epoch 22/50
38400/38400 [==============================] - 8s 198us/step - loss: 0.144
7 - acc: 0.9456 - val_loss: 0.3339 - val_acc: 0.8892
Epoch 23/50
38400/38400 [==============================] - 7s 191us/step - loss: 0.139
2 - acc: 0.9460 - val_loss: 0.3231 - val_acc: 0.8954
Epoch 24/50
38400/38400 [==============================] - 8s 198us/step - loss: 0.131
8 - acc: 0.9502 - val_loss: 0.3495 - val_acc: 0.8921
Epoch 25/50
38400/38400 [==============================] - 8s 198us/step - loss: 0.119
7 - acc: 0.9545 - val_loss: 0.3489 - val_acc: 0.8951
Epoch 26/50
38400/38400 [==============================] - 8s 218us/step - loss: 0.113
5 - acc: 0.9566 - val_loss: 0.3450 - val_acc: 0.8987
Epoch 27/50
38400/38400 [==============================] - 8s 205us/step - loss: 0.109
3 - acc: 0.9591 - val_loss: 0.3762 - val_acc: 0.8872
Epoch 28/50
38400/38400 [==============================] - 8s 197us/step - loss: 0.105
0 - acc: 0.9603 - val_loss: 0.3745 - val_acc: 0.8937
Epoch 29/50
38400/38400 [==============================] - 7s 191us/step - loss: 0.104
4 - acc: 0.9599 - val_loss: 0.3598 - val_acc: 0.8965
Epoch 30/50
```

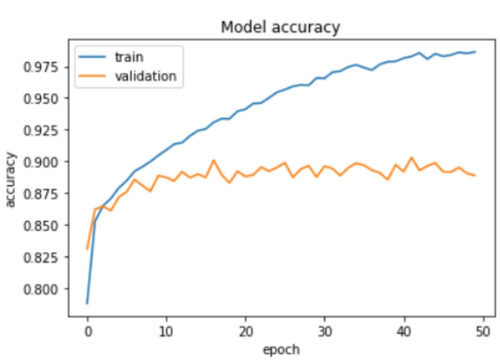
```
38400/38400 [==============================] - 8s 199us/step - loss: 0.090
6 - acc: 0.9658 - val_loss: 0.4232 - val_acc: 0.8875
Epoch 31/50
38400/38400 [==============================] - 7s 193us/step - loss: 0.093
6 - acc: 0.9655 - val_loss: 0.3786 - val_acc: 0.8960
Epoch 32/50
38400/38400 [==============================] - 8s 196us/step - loss: 0.079
1 - acc: 0.9704 - val_loss: 0.3860 - val_acc: 0.8943
Epoch 33/50
38400/38400 [==============================] - 7s 192us/step - loss: 0.076
8 - acc: 0.9711 - val_loss: 0.4313 - val_acc: 0.8886
Epoch 34/50
38400/38400 [==============================] - 8s 202us/step - loss: 0.068
3 - acc: 0.9745 - val_loss: 0.3878 - val_acc: 0.8948
Epoch 35/50
38400/38400 [==============================] - 8s 204us/step - loss: 0.065
8 - acc: 0.9762 - val_loss: 0.3906 - val_acc: 0.8984
Epoch 36/50
38400/38400 [==============================] - 7s 194us/step - loss: 0.067
3 - acc: 0.9740 - val_loss: 0.4091 - val_acc: 0.8968
Epoch 37/50
38400/38400 [==============================] - 8s 206us/step - loss: 0.074
6 - acc: 0.9720 - val_loss: 0.4290 - val_acc: 0.8929
Epoch 38/50
38400/38400 [==============================] - 8s 206us/step - loss: 0.061
3 - acc: 0.9766 - val_loss: 0.4413 - val_acc: 0.8907
Epoch 39/50
38400/38400 [==============================] - 8s 201us/step - loss: 0.058
0 - acc: 0.9786 - val_loss: 0.5043 - val_acc: 0.8854
Epoch 40/50
38400/38400 [==============================] - 8s 197us/step - loss: 0.055
9 - acc: 0.9790 - val_loss: 0.4395 - val_acc: 0.8972
Epoch 41/50
38400/38400 [==============================] - 8s 201us/step - loss: 0.052
2 - acc: 0.9814 - val_loss: 0.4476 - val_acc: 0.8917
Epoch 42/50
38400/38400 [==============================] - 8s 198us/step - loss: 0.046
4 - acc: 0.9829 - val_loss: 0.4555 - val_acc: 0.9029
Epoch 43/50
38400/38400 [==============================] - 8s 212us/step - loss: 0.040
5 - acc: 0.9856 - val_loss: 0.5131 - val_acc: 0.8927
Epoch 44/50
38400/38400 [==============================] - 8s 215us/step - loss: 0.051
9 - acc: 0.9807 - val_loss: 0.4901 - val_acc: 0.8963
Epoch 45/50
38400/38400 [==============================] - 8s 207us/step - loss: 0.040
9 - acc: 0.9849 - val_loss: 0.4803 - val_acc: 0.8986
Epoch 46/50
38400/38400 [==============================] - 8s 213us/step - loss: 0.047
1 - acc: 0.9829 - val_loss: 0.5238 - val_acc: 0.8917
Epoch 47/50
38400/38400 [==============================] - 8s 218us/step - loss: 0.042
7 - acc: 0.9840 - val_loss: 0.5253 - val_acc: 0.8915
Epoch 48/50
```

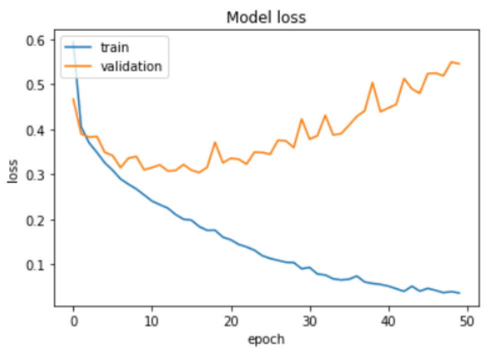
```
38400/38400 [==============================] - 8s 208us/step - loss: 0.037
5 - acc: 0.9861 - val_loss: 0.5192 - val_acc: 0.8951
Epoch 49/50
38400/38400 [==============================] - 8s 213us/step - loss: 0.039
9 - acc: 0.9852 - val_loss: 0.5495 - val_acc: 0.8904
Epoch 50/50
38400/38400 [==============================] - 9s 226us/step - loss: 0.036
8 - acc: 0.9864 - val_loss: 0.5460 - val_acc: 0.8887
```






```
plt.plot(result.history['acc'])
plt.plot(result.history['val_acc'])
plt.title('Model accuracy')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(['train', 'validation'], loc='upper left')
plt.show()
# summarize history for loss
plt.plot(result.history['loss'])
plt.plot(result.history['val_loss'])
plt.title('Model loss')
plt.ylabel('loss')
plt.xlabel('epoch')
plt.legend(['train', 'validation'], loc='upper left')
plt.show()
```






```
test_result = model.evaluate(test_x, test_y, verbose=0)
print('Classification loss:', test_result[0])
print('Test Accuracy:', test_result[1])
```

```
Classification loss: 0.5589288444047172
Test Accuracy: 0.8840833333333333
```

```
images,labels = ld.load('test')
test_x = images
target_test_y = labels
test_x = test_x.astype('float32')
test_x = test_x / 255
test_y = to_categorical(target_test_y)

y_pred = model.predict(test_x)
y_pred = np.argmax(np.round(y_pred),axis=1)
from sklearn.metrics import classification_report
print(classification_report(target_test_y, y_pred))
```

```
[[0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 ...
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]] [9 2 1 ... 8 1 5]
```

	precision	recall	f1-score	support
0	0.81	0.84	0.82	1000
1	0.96	0.99	0.97	1000
2	0.75	0.84	0.79	1000
3	0.91	0.84	0.87	1000
4	0.86	0.74	0.80	1000
5	0.99	0.96	0.97	1000
6	0.71	0.73	0.72	1000
7	0.91	0.98	0.94	1000
8	0.98	0.97	0.97	1000
9	0.98	0.93	0.95	1000
avg / total	0.88	0.88	0.88	10000

from sklearn.metrics import confusion_matrix
cm = confusion_matrix(target_test_y, y_pred)
print(cm)

```
[[840   6  19  15   3   0 114   0   3   0]
 [  3 986   0   7   0   0   3   0   1   0]
 [ 19   1 841  11  45   0  83   0   0   0]
 [ 37  29  28 841  28   0  35   0   2   0]
 [ 10   1 150  25 743   0  68   0   3   0]
 [  0   0   0   1   0 958   0  26   2  13]
 [121   6  74  22  38   0 733   0   6   0]
 [  0   0   0   0   0   7   0 984   0   9]
 [  7   2   8   4   2   2   3   5 967   0]
 [  1   0   0   1   0   5   0  68   0 925]]
```

```
y_classes = ["T-shirt/top","Trouser","Pullover","Dress","Coat","Sandal","S
hirt","Sneaker","Bag","Ankle boot"]
import seaborn as sns
sns.heatmap(cm, annot=True, fmt='d',xticklabels=y_classes, yticklabels=y_c
lasses)
plt.ylabel('Actual')
plt.xlabel('Predicted')
plt.show()
```