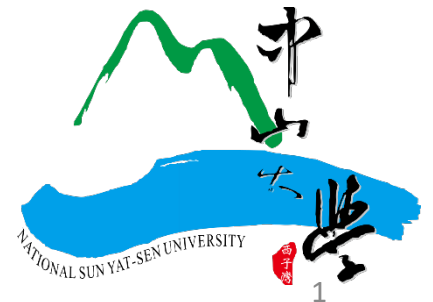


# Assignment 3a

魏家博 (Chia-Po Wei)

Department of Electrical Engineering  
National Sun Yat-sen University





# Outline

1. Fashion-MNIST with Keras
  - Model Fitting
  - Save Checkpoints

# Fashion MNIST



- 70,000 images in 10 classes
- Each class has 7,000 images
- Each grayscale image is 28x28
- 60,000 images for training
- 10,000 images for testing

0	T-shirt/top
1	Trouser
2	Pullover
3	Dress
4	Coat
5	Sandal
6	Shirt
7	Sneaker
8	Bag
9	Ankle boot

## Steps for Assignment 3a

1. Open [assignment\\_3a.ipynb](#)

2. Use `show_image_grids()` to see samples of Fashion-MNIST

Understand your training data is **very important** for the success of machine learning.

3. Normalize training and test data

```
train_images = train_images / 255.0  
test_images = test_images / 255.0
```

After normalization, the value of each pixel is between [0,1]

4. Create a Keras Sequential Model

```
model = keras.Sequential([  
    keras.layers.Flatten(input_shape=input_shape),  
    keras.layers.Dense(128, activation=tf.nn.relu),  
    keras.layers.Dense(output_shape, activation=tf.nn.softmax)  
])
```

You need to determine `input_shape` and `output_shape`.

## Steps for Assignment 3a

5. Use `model.summary()` to see the network structure of the model

6. Compile the Keras model

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

7. [Optional] Create TensorBoard callback

```
tb_callback = TensorBoard(log_dir='log_fashion')
```

8. Create checkpoint feedback

```
checkpoint_path = "ckpt_fashion/cp-{epoch:04d}.ckpt"  
cp_callback = ModelCheckpoint(checkpoint_path, save_weights_only=True,  
                             verbose=1, period=5)
```

## Steps for Assignment 3a

### 9. Fit the model

```
model.fit(train_images, train_labels, epochs=20, callbacks =  
[cp_callback, tb_callback])
```

### 10. Compute and print the test accuracy

```
test_loss, test_acc = model.evaluate(test_images, test_labels)
```

### 11. [Optional] Type the following command in Anaconda Prompt (make sure that log\_fashion/ is in the current directory)

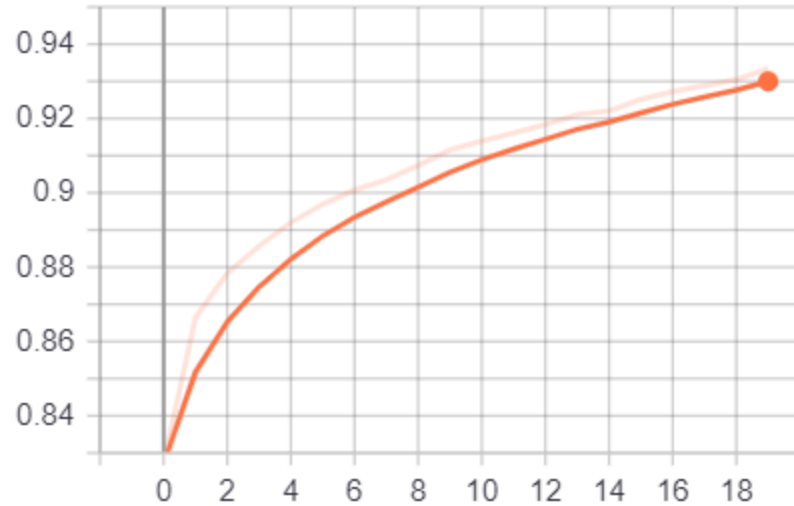
```
tensorboard.exe --logdir log_fashion
```

Navigate your web browser to `localhost:6006` to view the TensorBoard

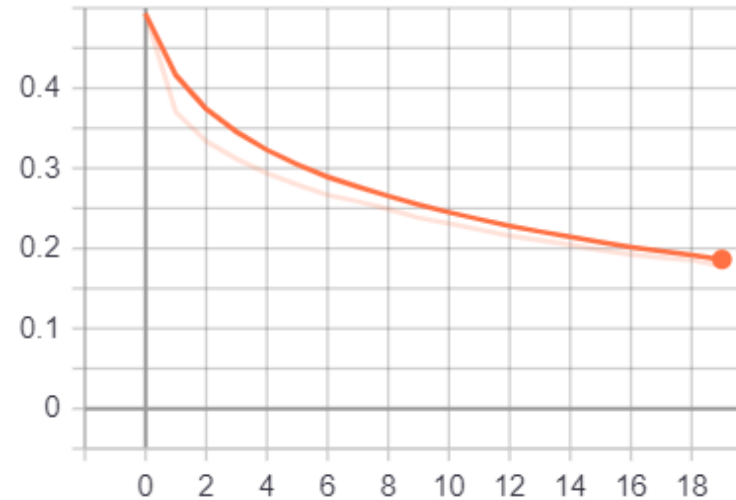
## Steps for Assignment 3a

TensorBoard will display the following two figures.

epoch\_acc



epoch\_loss



12. Compute and report the test accuracy using ckpt saved at different training epochs, `model.load_weights(weight_path)`

```
loss, acc = model.evaluate(test_images, test_labels)
```

where `weight_path` is one of the following

```
ckpt_fashion/cp-0005.ckpt
```

```
ckpt_fashion/cp-0010.ckpt
```

```
ckpt_fashion/cp-0015.ckpt
```

```
ckpt_fashion/cp-0020.ckpt
```

## Steps for Assignment 3a

### 13. Load the ckpt saved at epoch = 20.

Use `prob = model.predict(test_images)` to get the probability.

`prob.shape` is (10000,10)

`prob[i, :]` is the probability of `test_images[i, :, :]`

Compute predictions based on `prob`. For example,

if `prob[i, :]` is [0, 0.4, 0.6, 0, 0, 0, 0, 0, 0, 0],

then `predictions[i]` should be 2 (the index corresponding to the largest probability). You can use `np.argmax()` to compute predictions.

Compute and print the test accuracy by comparing `predictions` with `test_labels`. If `predictions[i]` equals `test_labels[i]`, then this means the prediction of `i`th test image is correct, otherwise the prediction is incorrect. Hint: Use the `==` operator.

此步驟不需要使用迴圈，如使用迴圈會被扣分。

此步驟計算得到之 test accuracy 與用 `model.evaluate()` 得到的 test accuracy 一致。



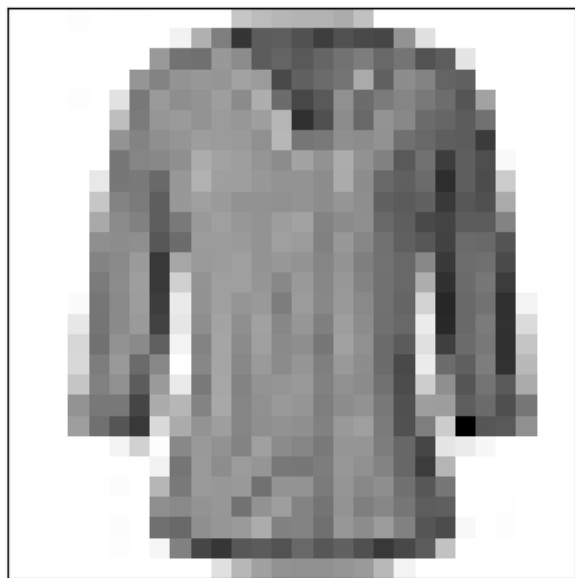
## Steps for Assignment 3a

14. Display the prediction result of `test_image[4]` and `np.fliplr(test_images[4])` as in the following two figures (左圖是 `test_image[4]` 預測結果，右圖是對 `test_image[4]` 左右翻轉後之預測結果).

Use `prob = model.predict(test_images[4:5])`, because `model.predict()` is assumed to predict multiple images at once.

`test_images[4:5].shape` is `(1,28,28)`

`test_images[4].shape` is `(28,28)`



Shirt 55%  
(Shirt is the true label)



Coat 83%  
(Shirt is the true label)

```
plt.grid(False)
plt.xticks([])
plt.yticks([])
plt.imshow(img, cmap=plt.cm.binary)
plt.show()
```

Use `plt.xlabel(description)` to display to xlabel, where `description` contains the words we wish to add to the image.

## Questions for Assignment 3a

1. How many training epochs in `model.fit()` are required such that the training loss is converged? (epochs=20 其實還未收斂，需畫出 loss vs. epochs 曲線)  
<https://machinelearningmastery.com/display-deep-learning-model-training-history-in-keras/>
2. Adjust the batch\_size in `model.fit()`. What will happen if we change batch\_size from 32 to 512? (訓練時間是增加或減少？測試正確率是上升或下降？原因為何？)
3. What is the difference between the following two options in `model.compile()`?

```
loss='categorical_crossentropy'
```

```
loss='sparse_categorical_crossentropy'
```

Use an example to illustrate the difference.