ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

DEEP LEARNING TRONG KHOA HỌC DỮ LIỆU
LAB 2

TỐI ƯU MÔ HÌNH MẠNG NEURAL

Họ và tên : Lưu Quang Tiến Hoàng
MSSV : 20521342
Lớp : DS201.N11

Mục tiêu

- Ôn tập kiến thức cơ bản về các phương pháp tối ưu mô hình mạng neural.
- Cài đặt thử nghiệm một số phương pháp tối ưu mô hình mạng neural.
- Áp dụng mô hình đã tối ưu vào bài toán phân loại ảnh trang phục.

(?) Liên hệ kiến thức đã học, hãy kể tên:

- Một số loại tham số (parameters) có thể có của mô hình mạng neural ?
- Một số loại siêu tham số (hyperparameters) có thể có của mô hình mạng neural ?
- Một số phương pháp tối ưu mô hình mạng neural đã học ?

Một số tham số: weight, bias,…

Một số loại siêu tham số: Learning rate, batch size, epochs,…

Một số phương pháp: Gradient Descent, Adam,…

(?) Sử dụng lệnh summary để xem cấu trúc của mô hình và cho biết kết quả ?

```
[115] model.summary()

Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense (Dense)               (None, 784)               615440

 dense_1 (Dense)             (None, 10)                7850

=================================================================
Total params: 623,290
Trainable params: 623,290
Non-trainable params: 0
_________________________________________________________________
```

Mô hình có 2 lớp và có 623290 tham số.

I, Load dữ liệu:

(?) Hãy khảo sát bộ dữ liệu Fashion MNIST và cho biết:

- Tập train và tập test có bao nhiêu ảnh ?
- Mỗi ảnh trong tập train và tập test có kích thước bao nhiêu ?
- Tập train có bao nhiêu nhãn và liệt kê tên các nhãn ?

```
[126] X_train.shape

      (60000, 28, 28)

[127] X_test.shape

      (10000, 28, 28)
```

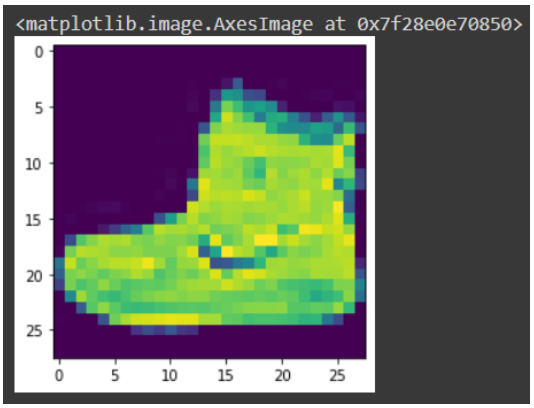
Tập train có 60000 ảnh còn tập test có 10000 ảnh.

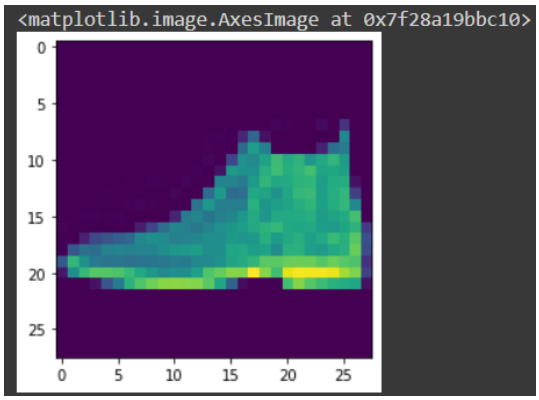
Chúng đều có kích thước 28x28.

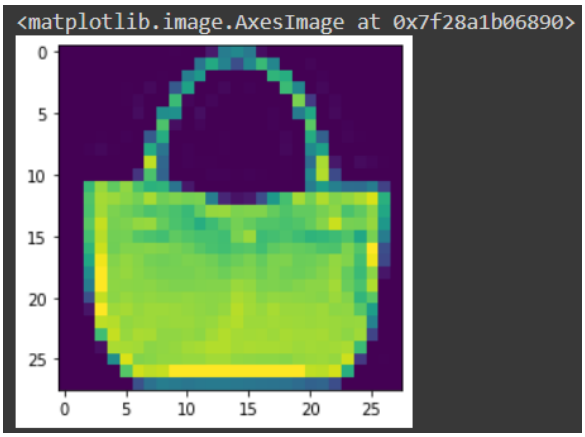
Có 10 nhãn được đánh số từ 0-9.

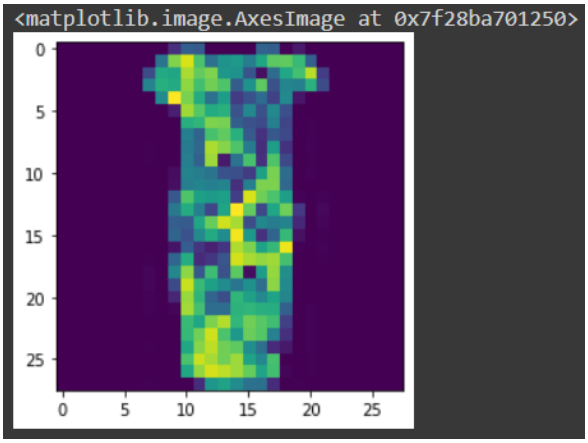
```
labels=list(set(y_train))

labels

[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

(?) Sử dụng thư viện Matplotlib để trực quan các ảnh sau: X_train[0], X_train[50], X_test[100], X_test[1000] ?









II, Chuẩn bị dữ liệu:

III, Xây dựng và huấn luyện mô hình:

(?) Kể tên một số hàm kích hoạt do thư viện Keras cung cấp ?

Sigmoid, tanh, linear, relu, maxout, leaky relu,…

(?) Sử dụng lệnh summary để xem cấu trúc của mô hình đã xây dựng:

- Cho biết kết quả thực thi câu lệnh ?
- Cho biết tổng số tham số của mô hình ?

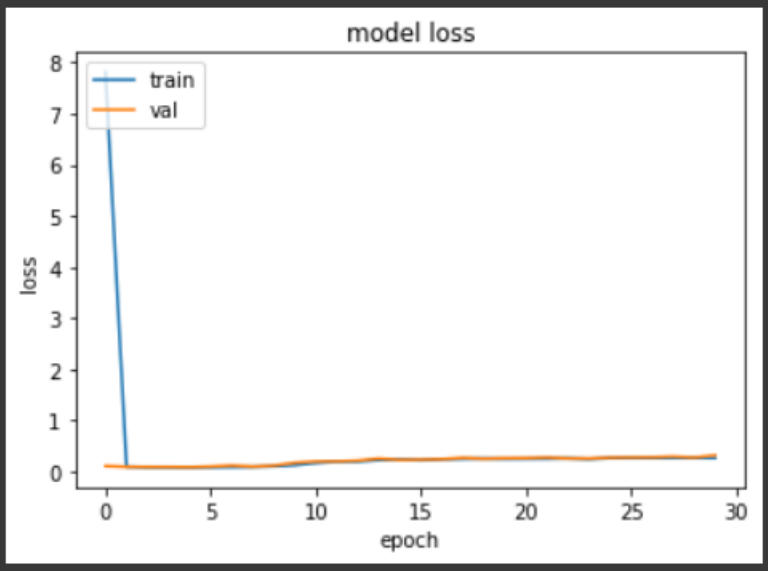
Kết quả thực thi câu lệnh:

```
[241] model_1.add(Dense(784,input_shape=(784,),activation ='relu'))
      model_1.add(Dense(10,input_shape=(10,),activation ='sigmoid'))

[242] model_1.summary()

      Model: "sequential_32"
      _________________________________________________________________
       Layer (type)                 Output Shape              Param #
      =================================================================
       dense_71 (Dense)             (None, 784)               615440

       dense_72 (Dense)             (None, 10)                7850

      =================================================================
      Total params: 623,290
      Trainable params: 623,290
      Non-trainable params: 0
      _________________________________________________________________
```

Tổng tham số mô hình là 623290.

(?) Vẽ đồ thị học với **loss** ?

```
import matplotlib.pyplot as plt
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('model loss')
plt.ylabel('loss')
plt.xlabel('epoch')
plt.legend(['train', 'val'], loc='upper left')
plt.show()
```



IV, Hàm mất mát (LOSS)

V, Chuẩn hóa mô hình (REGULARIZATION)

(?) Viết công thức chuẩn hóa L1 và L2 cho tham số W ?

Công thức chuẩn hóa L1:

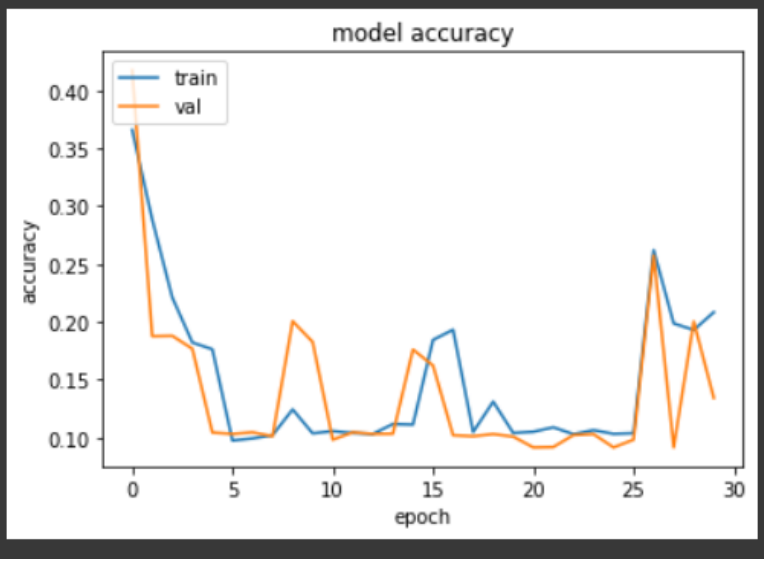
$$
\begin{aligned}
w_{\text{new}} &= w - \eta \frac{\partial L_1}{\partial w} \\
&= w - \eta \cdot \left[2x(wx + b - y) + \lambda \frac{d|w|}{dw} \right] \\
&= \begin{cases} w - \eta \cdot \left[2x(wx + b - y) + \lambda \right] & w > 0 \\ w - \eta \cdot \left[2x(wx + b - y) - \lambda \right] & w < 0 \end{cases}
\end{aligned}
$$

Công thức chuẩn hóa L2:

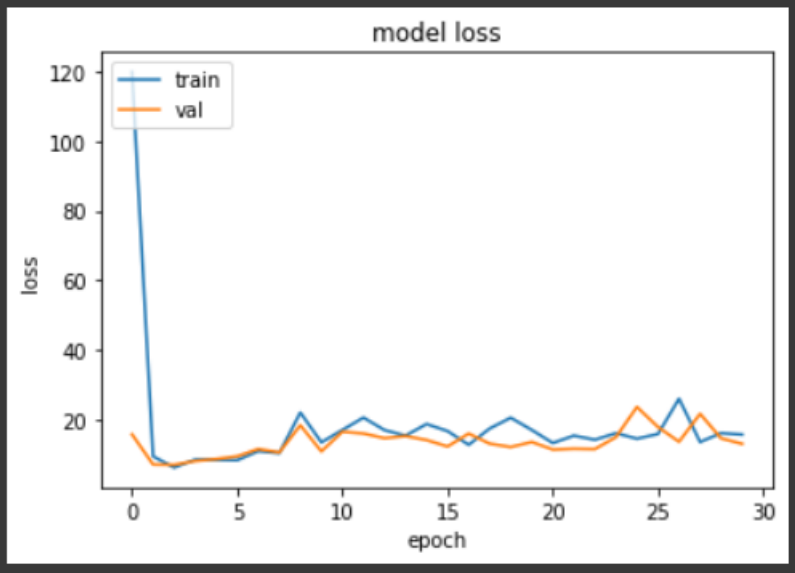
$$
\begin{aligned}
w_{\text{new}} &= w - \eta \frac{\partial L_2}{\partial w} \\
&= w - \eta \cdot \left[2x(wx + b - y) + 2\lambda w \right]
\end{aligned}
$$

(?) Huấn luyện lại mô hình, sau đó vẽ đồ thị học với **accuracy** và với **loss** ?

```
plt.plot(history2.history['accuracy'])
plt.plot(history2.history['val_accuracy'])
plt.title('model accuracy')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(['train', 'val'], loc='upper left')
plt.show()
```



```
plt.plot(history2.history['loss'])
plt.plot(history2.history['val_loss'])
plt.title('model loss')
plt.ylabel('loss')
plt.xlabel('epoch')
plt.legend(['train', 'val'], loc='upper left')
plt.show()
```



(?) Đánh giá mô hình trên tập test và cho biết độ chính xác dự đoán ?

Độ chính xác của mô hình đánh giá trên tập test là 14.17%

```
[297] y_pred_2 =model2.predict(X_test_reshaped)
      y_pred_label_2 =np.argmax(y_pred_2,axis=1)
      accuracy_score(y_test,y_pred_label_2)*100

      313/313 [==============================] - 0s 1ms/step
      14.17
```

VI, Khởi tạo tham số (Parameter Initialization)

(?) Thử khởi tạo tham số 0 và tham số 1 cho mô hình, sau đó huấn luyện và xem sự ảnh hưởng của việc khởi tạo tham số đối với mô hình như thế nào ?

```
[302] from tensorflow.keras.initializers import GlorotUniform
      from tensorflow import zeros, ones

      model4=Sequential()

[304] model4.add(Dense(784, input_shape=(784, ), kernel_initializer=zeros, bias_initializer=zeros, activation='relu'))
      model4.add(Dense(10, activation='sigmoid'))
      model4.compile(optimizer=optimizer, loss=loss, metrics=['accuracy'])

[305] model5 = Sequential()
      model5.add(Dense(784, input_shape=(784, ), kernel_initializer=ones, bias_initializer=ones, activation='relu'))
      model5.add(Dense(10, activation='sigmoid'))
      model5.compile(optimizer=optimizer, loss=loss, metrics=['accuracy'])

[306] history4 = model4.fit(X_train_reshaped, y_train_new, validation_data=(X_dev_reshaped, y_dev_new), batch_size=128, epochs=30)
      plt.plot(history4.history['accuracy'])
      plt.plot(history4.history['val_accuracy'])
      plt.title('model accuracy')
      plt.ylabel('accuracy')
      plt.xlabel('epoch')
      plt.legend(['train', 'val'], loc='upper left')
      plt.show()
```

```
history5 = model5.fit(X_train_reshaped, y_train_new, validation_data=(X_dev_reshaped, y_dev_new), batch_size=128, epochs=30)
plt.plot(history5.history['accuracy'])
plt.plot(history5.history['val_accuracy'])
plt.title('model accuracy')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(['train', 'val'], loc='upper left')
plt.show()
```

Đánh giá mô hình :

```
y_pred_5 = model5.predict(X_test_reshaped)
y_pred_5 = np.argmax(y_pred_5, axis=-1)
print('Accuracy is ', accuracy_score(y_test, y_pred_5)*100)
```

```
313/313 [==============================] - 1s 2ms/step
Accuracy is  10.0
```

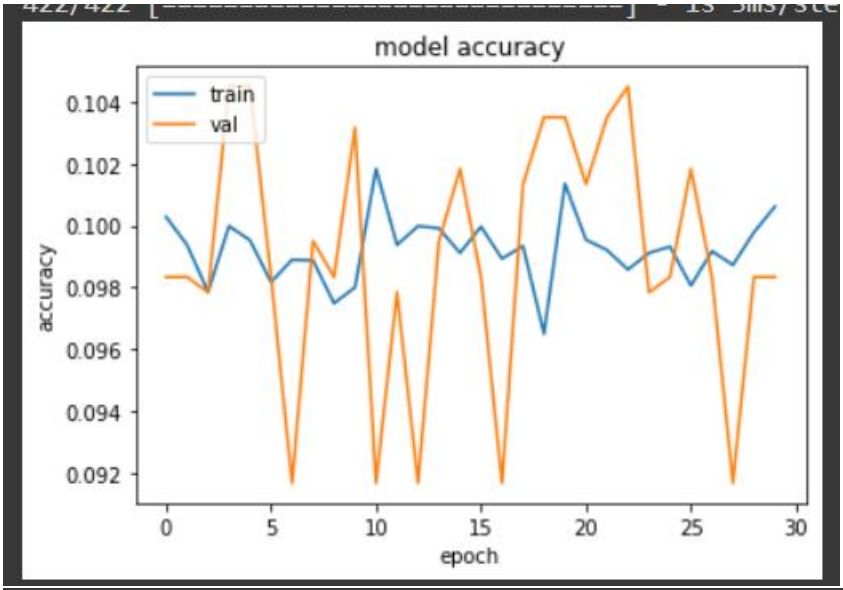
Nhận xét: Việc khởi tạo tham số giúp mô hình đạt độ chính xác thấp hơn.

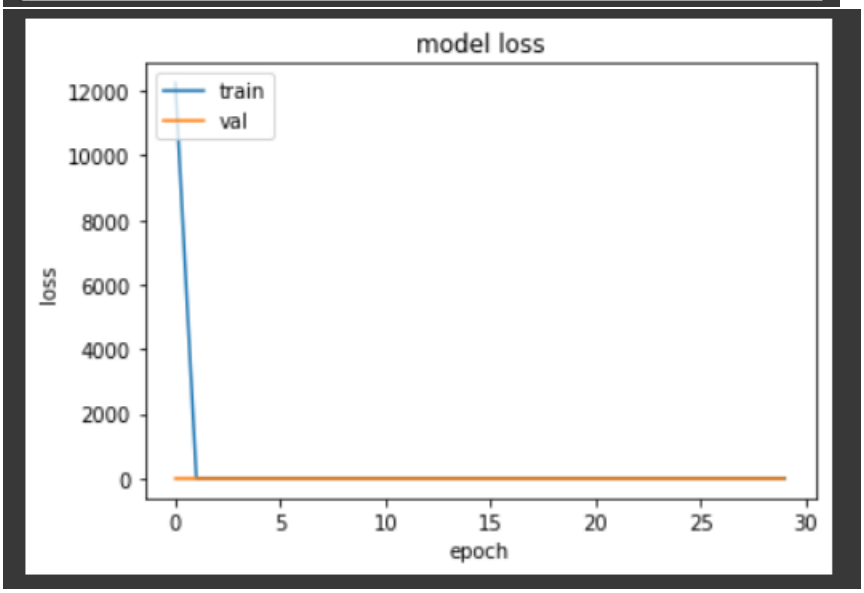
VII, Các thuật toán tối ưu:

(?) Kể tên một số thuật toán tối ưu do thư viện Keras cung cấp ?

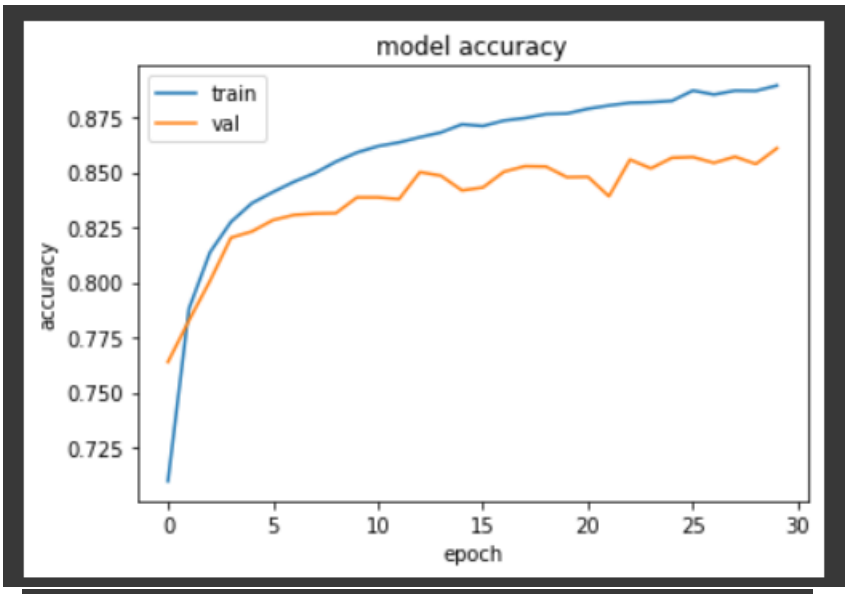
SGD, RMSprop, Adam, Nadam, Adamax.

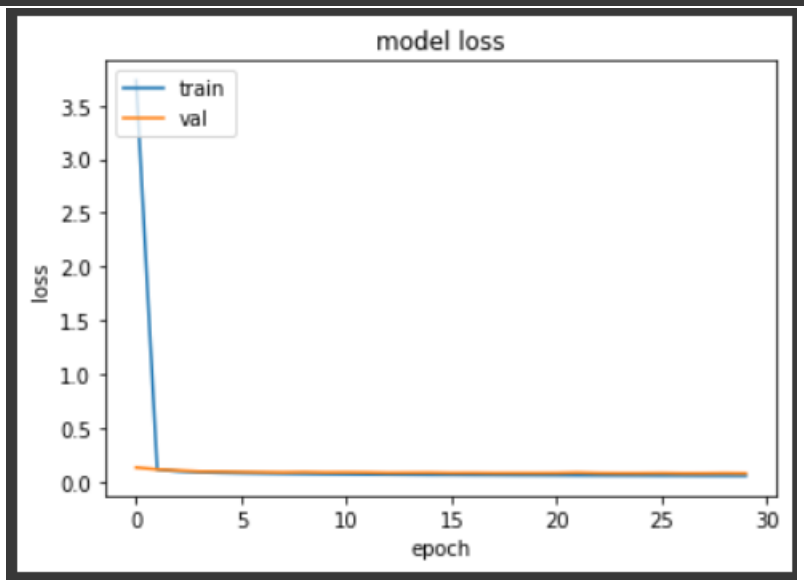
(?) Huấn luyện mô hình, sau đó vẽ đồ thị học với **accuracy** và với **loss** ?



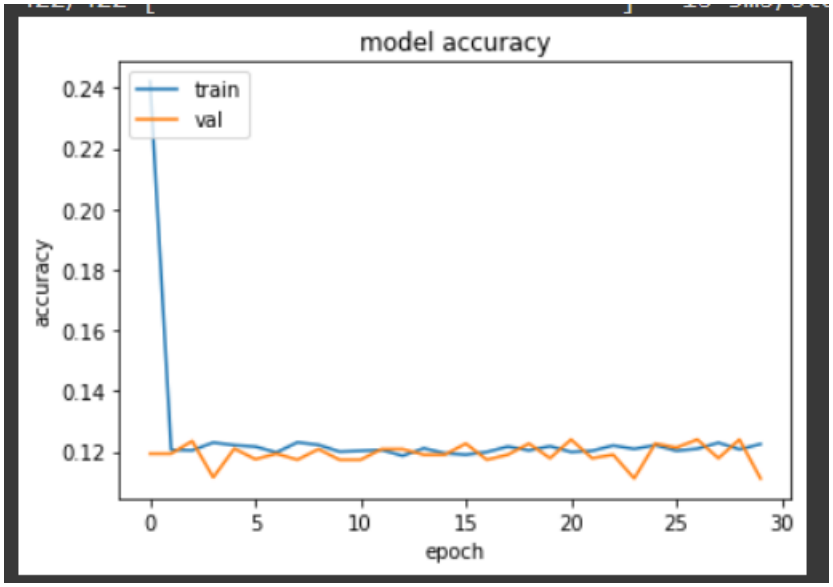


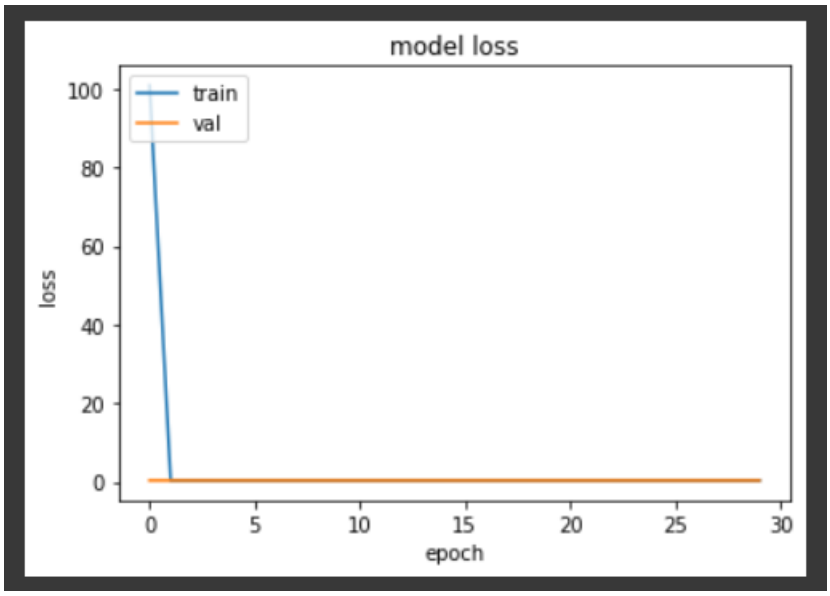
(?) Huấn luyện mô hình, sau đó vẽ đồ thị học với **accuracy** và với **loss** ?



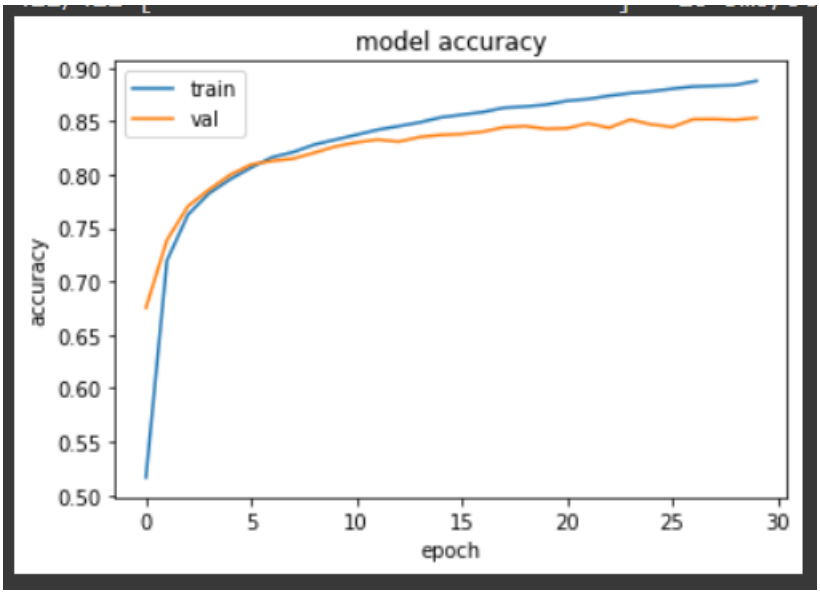


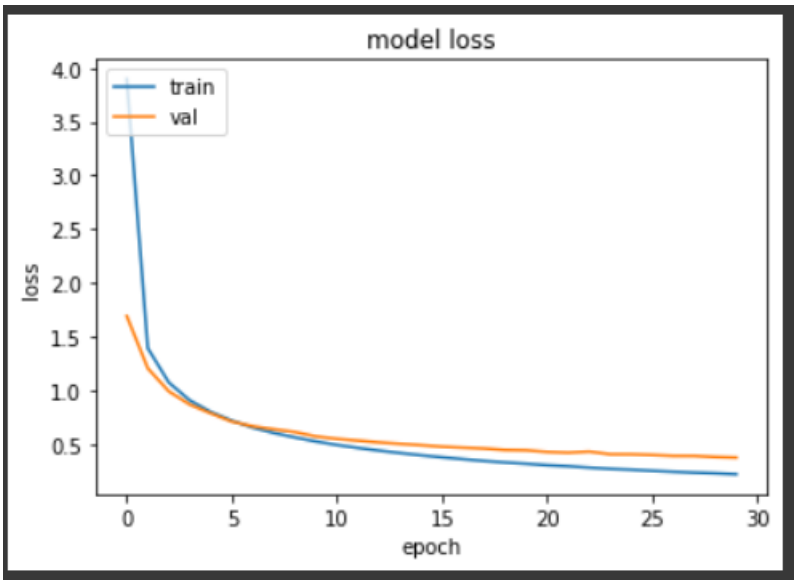
(?) Huấn luyện mô hình, sau đó vẽ đồ thị học với **accuracy** và với **loss** ?





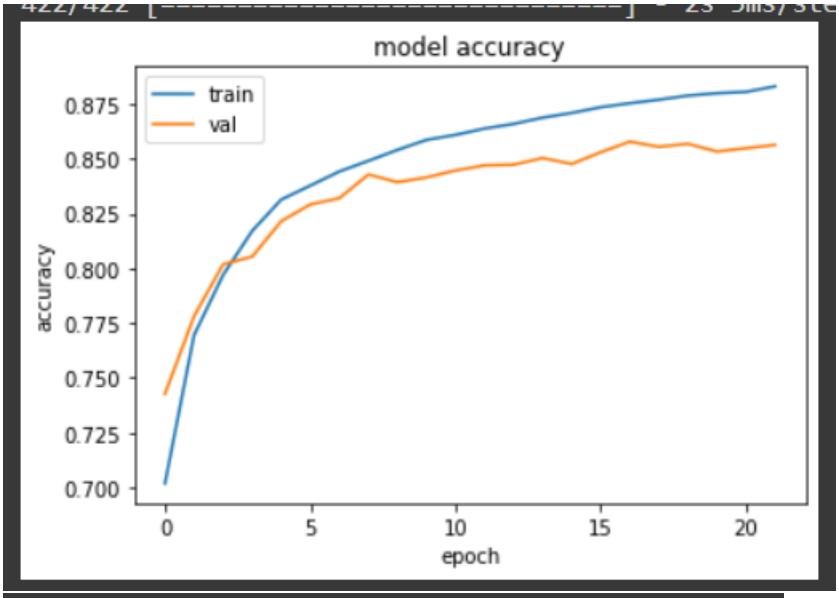
(?) Huấn luyện mô hình, sau đó vẽ đồ thị học với **accuracy** và với **loss** ?

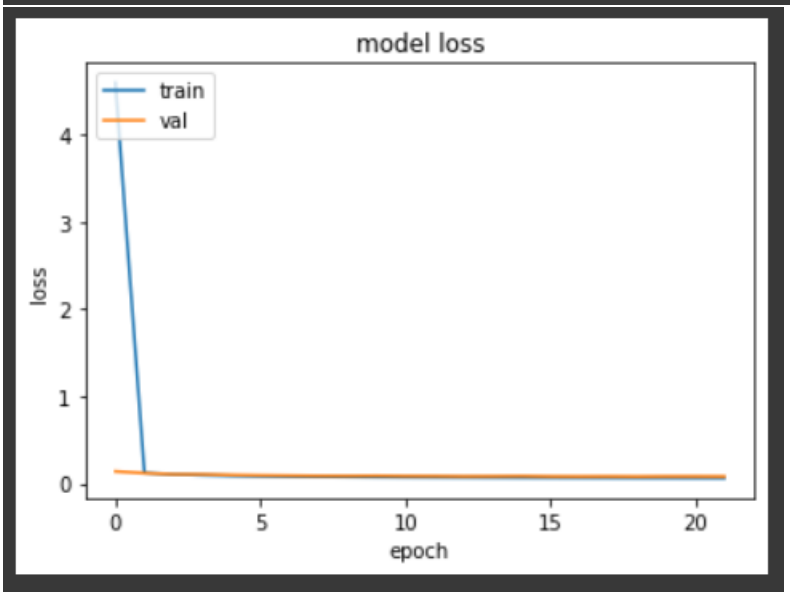




(?) Huấn luyện mô hình, sau đó thực hiện các yêu cầu sau:

- Vẽ đồ thị học với **accuracy** và với **loss** ?
- Từ đồ thị học, hãy cho biết quá trình huấn luyện dừng lại sau bao nhiêu epochs và val_loss đạt giá trị nhỏ nhất tại epoch thứ mấy?





Nhận xét: Quá trình huấn luyện dừng lại sau 22 epochs và val_loss đạt giá trị nhỏ nhất tại epoch thứ 20.

(?) Với mô hình đã được tối ưu:

- Đánh giá mô hình trên tập test và nêu độ chính xác dự đoán ?
- Vẽ ma trận nhầm lẫn của mô hình và nêu nhận xét ?

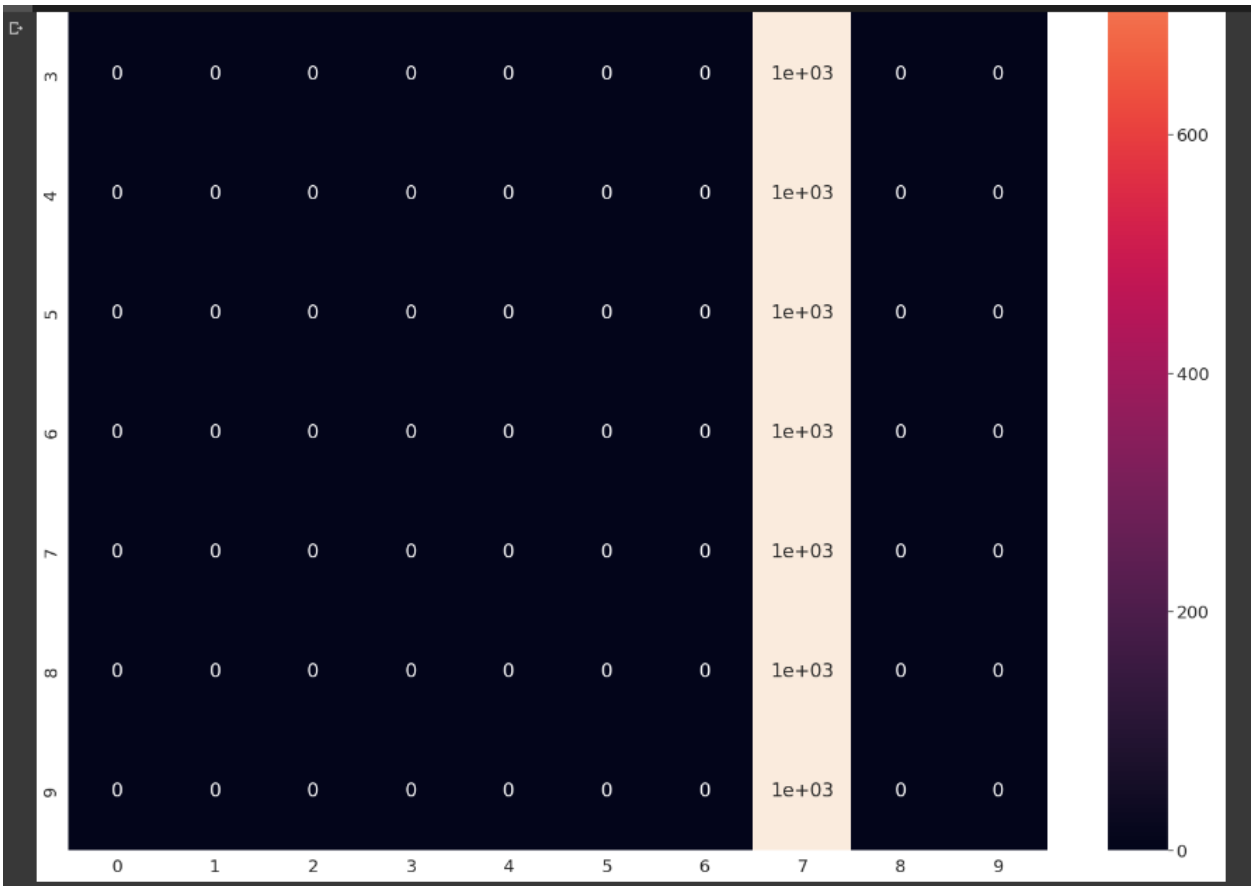
```
from sklearn.metrics import accuracy_score
from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay
y_pred_8 = model8.predict(X_test_reshaped)
y_pred_8 = np.argmax(y_pred_8, axis=-1)
print('Accuracy is ', accuracy_score(y_test, y_pred_8)*100)
```

```
313/313 [==============================] - 0s 1ms/step
Accuracy is  84.31
```

Ma trận nhầm lẫn:

```
import seaborn as sn
import matplotlib.pyplot as plt
cf = confusion_matrix(y_test, y_pred)
print(cf)
df_cm = pd.DataFrame(cf, index = [i for i in "0123456789"],
                     columns = [i for i in "0123456789"])
print(df_cm)
# sn.heatmap(df_cm, annot=True, cbar=False, linewidths=1, cmap='green')
# plt.xlabel('Predictions', fontsize=18)
# plt.ylabel('Actuals', fontsize=18)
# plt.title('Confusion Matrix', fontsize=18)
plt.figure(figsize = (25,25))
sn.set(font_scale=1.8)
sn.heatmap(df_cm, annot=True)
plt.show()
```

Nhận xét: Mô hình dự đoán sai nhiều ở các nhãn 4 và 6


3 0 0 0 0 0 0 0 1e+03 0 0
4 0 0 0 0 0 0 0 1e+03 0 0
5 0 0 0 0 0 0 0 1e+03 0 0
6 0 0 0 0 0 0 0 1e+03 0 0
7 0 0 0 0 0 0 0 1e+03 0 0
8 0 0 0 0 0 0 0 1e+03 0 0
9 0 0 0 0 0 0 0 1e+03 0 0
0 1 2 3 4 5 6 7 8 9
600
400
200
0