

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

DEEP LEARNING TRONG KHOA HỌC DỮ LIỆU
LAB 3

MẠNG NEURAL TÍCH CHẬP (CNN)

Họ và tên : Lưu Quang Tiến Hoàng

MSSV : 20521342

Lớp : DS201.N11

Mục tiêu

- Ôn tập kiến thức cơ bản về mô hình mạng neural tích chập (CNN).
- Biết cách xây dựng mô hình mạng neural tích chập cơ bản và sử dụng một số mô hình mạng neural tích chập nổi tiếng.
- Áp dụng các mô hình trên vào bài toán phân loại chữ số viết tay (MNIST).

KHỞI ĐỘNG

(?) Kể tên một số mạng neural tích chập nổi tiếng mà bạn biết ?

Mạng neural tích chập nổi tiếng: LeNet-5, AlexNet, VGGNet, R-CNN...

(?) Trong một mạng neural tích chập, có thể có những loại lớp nào ?

- Convolution (conv)
- Pooling(pool)
- Fullyconnected (FC)

1. Các loại lớp trong mạng neural tích chập

(?) Thư viện Keras cung cấp những loại lớp nào ?

Convolution:

- Conv1D layer
- Conv2D layer
- Conv3D layer
- SeparableConv1D layer
- SeparableConv2D layer
- DepthwiseConv1D layer
- Conv1DTranspose layer
- Conv2DTranspose layer
- Conv3DTranspose layer

Pooling:

- MaxPooling1D layer
- MaxPooling2D layer
- MaxPooling3D layer
- AveragePooling1D layer
- AveragePooling2D layer
- AveragePooling3D layer
- GlobalMaxPooling1D layer
- GlobalMaxPooling2D layer
- GlobalMaxPooling3D layer
- GlobalAveragePooling1D layer
- GlobalAveragePooling2D layer
- GlobalAveragePooling3D layer

The base Layer class

Layer activations

Layer weight initializers

Layer weight regularizes

Layer weight constraints

Core layers

Recurrent layers

Preprocessing layer

Normalization layers

Regularization layers

Attention layers

Reshaping layers

Merging layers

Locally-connected layers

Activation layers

2. Chuẩn bị dữ liệu

3. Xây dựng mô hình mạng neural tích chập

(?) Sử dụng lệnh summary và cho biết cấu trúc của mô hình đã xây dựng ?

```
Model: "sequential_11"
```

Layer (type)	Output Shape	Param #
conv2d_28 (Conv2D)	(None, 26, 26, 32)	320
max_pooling2d_18 (MaxPooling2D)	(None, 13, 13, 32)	0
conv2d_29 (Conv2D)	(None, 11, 11, 64)	18496
max_pooling2d_19 (MaxPooling2D)	(None, 5, 5, 64)	0
flatten_11 (Flatten)	(None, 1600)	0
dense_21 (Dense)	(None, 10)	16010

```
=====  
Total params: 34,826  
Trainable params: 34,826  
Non-trainable params: 0  
=====
```

(?) Cho biết tổng số tham số (Total params) của mô hình đã xây dựng ?

Trả lời: 34826

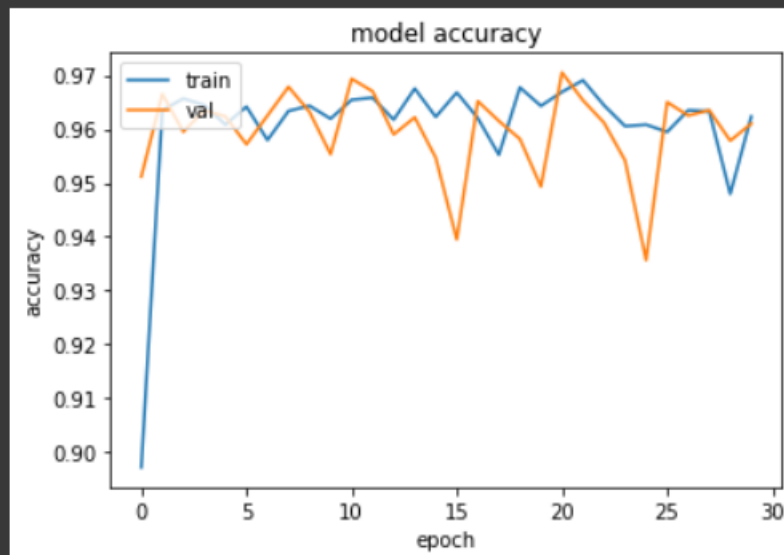
(?) Tiến hành huấn luyện mô hình, sau đó vẽ đồ thị học với accuracy và loss ?

```
[120] from tensorflow.keras.optimizers import Adam
      from tensorflow.keras.losses import CategoricalCrossentropy
      optimizer = Adam(learning_rate=0.01)
      loss = CategoricalCrossentropy()
      model.compile(optimizer=optimizer, loss=loss, metrics=['accuracy'])

[121] history = model.fit(X_train_expanded, y_train_new,
      batch_size=128, epochs=30,
      validation_data=(X_dev_expanded, y_dev_new))
```

Vẽ đồ thị học với accuracy

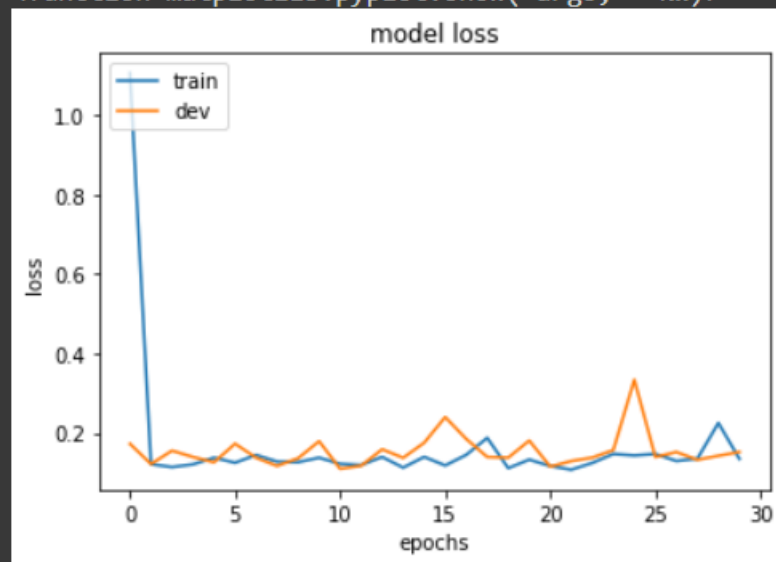
```
[209] import matplotlib.pyplot as plt
      plt.plot(history.history['accuracy'])
      plt.plot(history.history['val_accuracy'])
      plt.title('model accuracy')
      plt.ylabel('accuracy')
      plt.xlabel('epoch')
      plt.legend(['train', 'val'], loc = 'upper left')
      plt.show()
```



Vẽ đồ thị học với loss

```
✓ [210] import matplotlib.pyplot as plt  
0 giây  
plt.plot(history.history['loss'])  
plt.plot(history.history['val_loss'])  
plt.title('model loss')  
plt.ylabel('loss')  
plt.xlabel('epochs')  
plt.legend(['train', 'dev'], loc='upper left')  
plt.show
```

<function matplotlib.pyplot.show(*args, **kw)>



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

```
✓ [207] y_pred = model.predict(X_test_expanded)  
1 giây  
y_pred_label = np.argmax(y_pred,axis=-1)  
  
313/313 [=====] - 1s 2ms/step
```

```
✓ [208] from sklearn.metrics import accuracy_score  
0 giây  
accuracy_score(y_test, y_pred_label)*100  
  
95.88
```

(?) Tổng xác suất của các output tại lớp Softmax là bao nhiêu ?

Trả lời: Bằng 1

4. Xây dựng mạng LeNet-5

(?) Sử dụng lệnh summary và cho biết cấu trúc của mô hình đã xây dựng ?

\$4. Xây dựng mạng LeNet-5

```
[126] model2 = Sequential()
      model2.add(Conv2D(filters=6, kernel_size=(5,5), padding='same',
                        activation='relu', input_shape=(28,28,1)))
      model2.add(AveragePooling2D(strides=2, pool_size=(2, 2)))
      model2.add(Conv2D(filters=16, kernel_size=(5,5), padding='valid',
                        activation='relu'))
      model2.add(AveragePooling2D(strides=2, pool_size=(2, 2)))
      model2.add(Flatten())
      model2.add(Dense(120, activation='relu'))
      model2.add(Dense(84, activation='relu'))
      model2.add(Dense(10, activation='softmax'))
```

```
[172] model2.summary()
```

Model: "sequential_12"

Layer (type)	Output Shape	Param #
conv2d_30 (Conv2D)	(None, 28, 28, 6)	156
average_pooling2d_6 (AveragePooling2D)	(None, 14, 14, 6)	0
conv2d_31 (Conv2D)	(None, 10, 10, 16)	2416
average_pooling2d_7 (AveragePooling2D)	(None, 5, 5, 16)	0
flatten_12 (Flatten)	(None, 400)	0
dense_22 (Dense)	(None, 120)	48120
dense_23 (Dense)	(None, 84)	10164
dense_24 (Dense)	(None, 10)	850

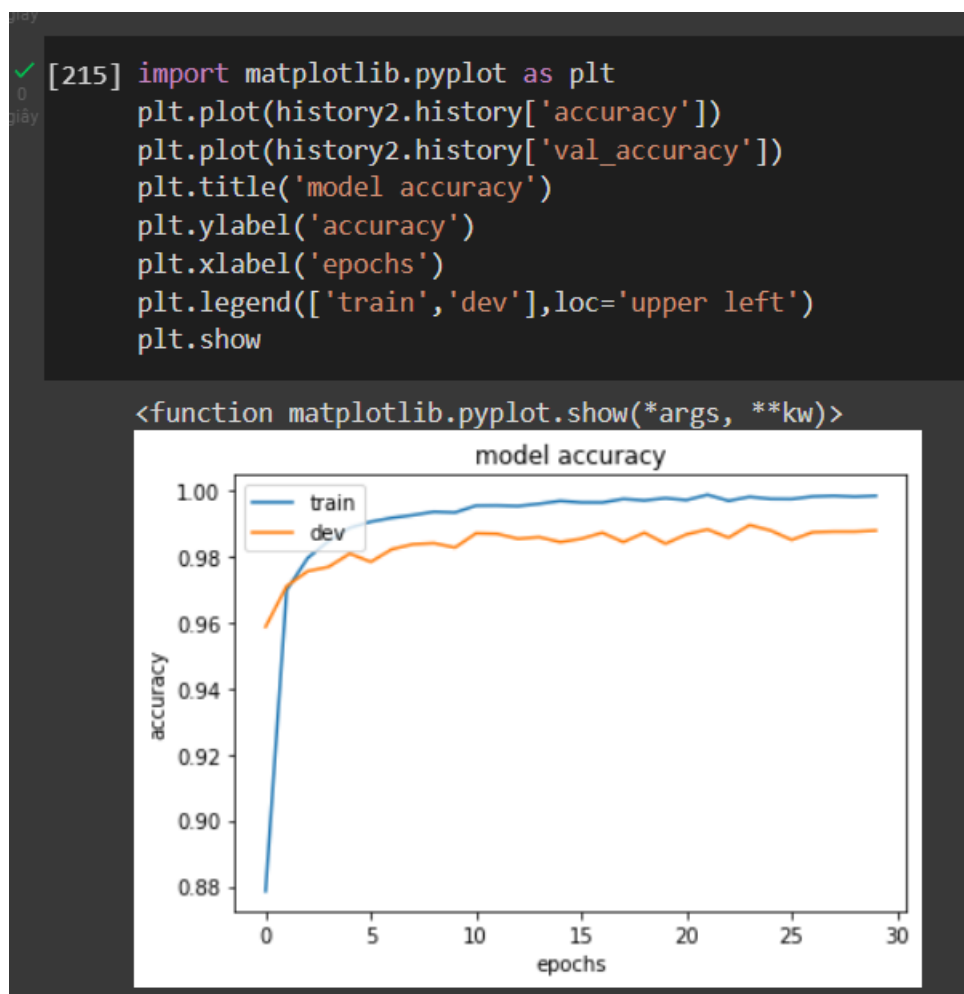
```
=====  
Total params: 61,706  
Trainable params: 61,706  
Non-trainable params: 0
```

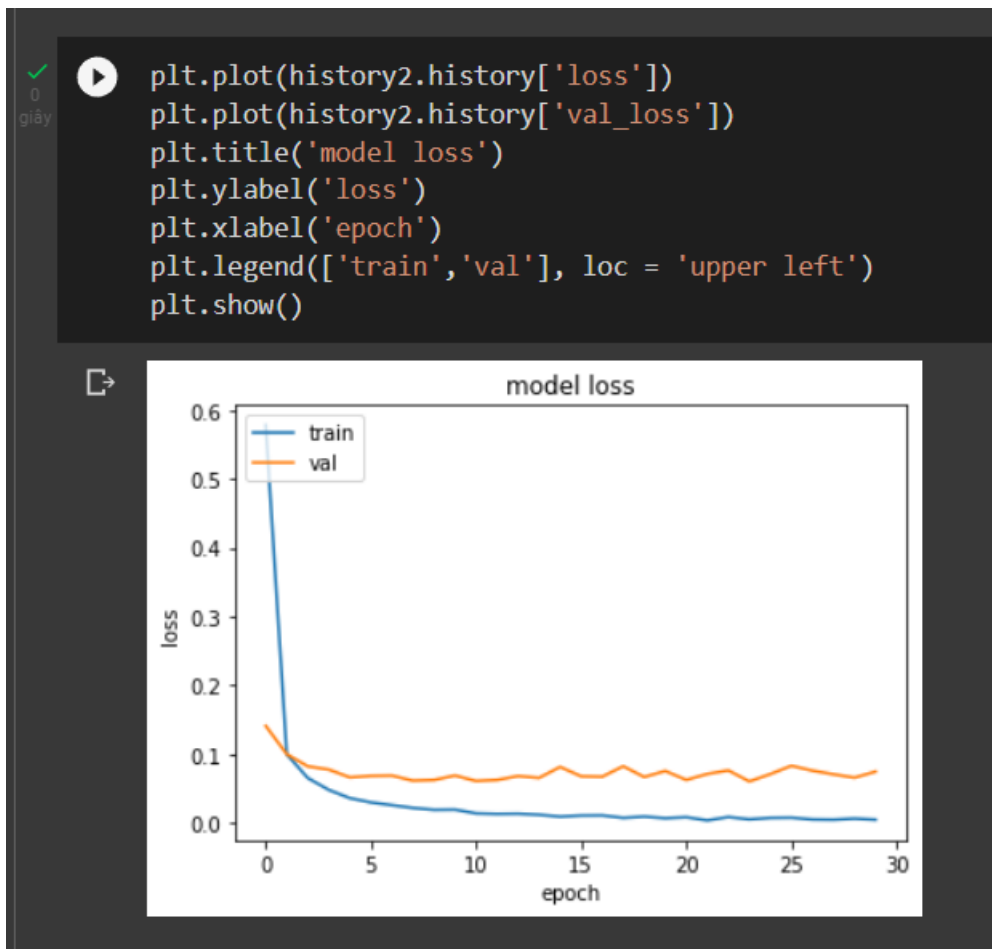
(?) Cho biết tổng số tham số (Total params) của mô hình đã xây dựng ?

Trả lời: 61706

(?) Tiến hành huấn luyện mô hình, sau đó vẽ đồ thị học với accuracy và loss ?

```
[127] optimizer = Adam(learning_rate=5e-4)
      loss = CategoricalCrossentropy()
      model2.compile(optimizer=optimizer, loss=loss, metrics=['accuracy'])
```





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

Trả lời: Độ chính xác dự đoán: 98.8%

1 giây

```
[217] y_pred = model2.predict(x_test_expanded)
      y_pred = np.argmax(y_pred , axis = -1)

313/313 [=====] - 1s 3ms/step
```

0 giây

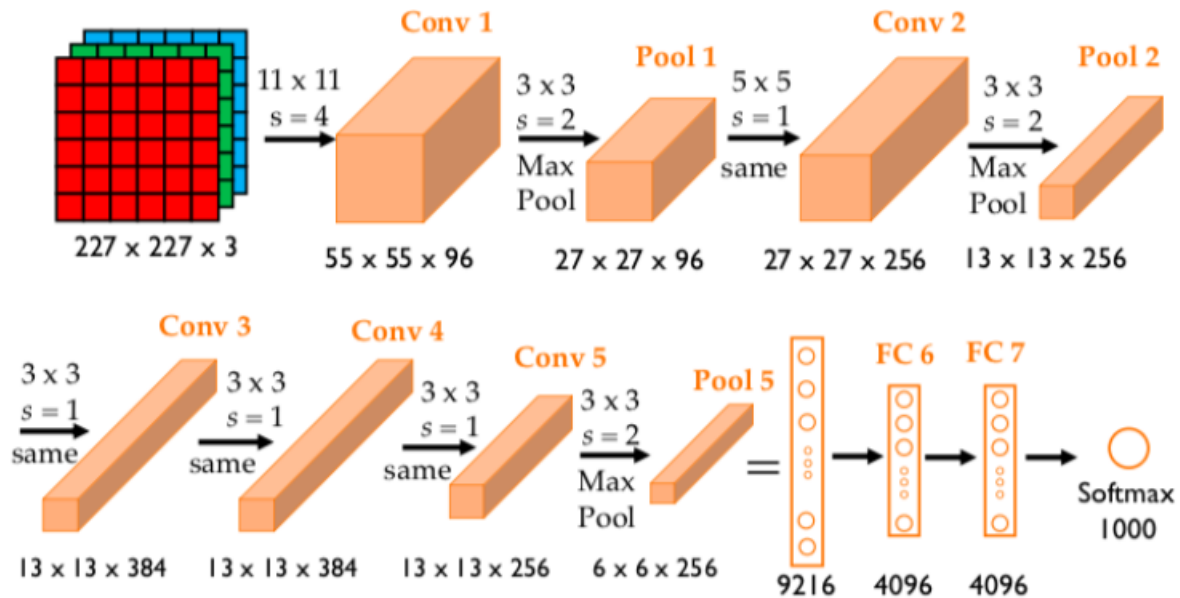
```
[218] accuracy_score(y_pred,y_test)*100

98.8
```

(?) Tổng xác suất của các output tại lớp Softmax là bao nhiêu ?

Trả lời: Bằng 1

5. Ví dụ áp dụng: Xây dựng mạng AlexNet



Minh họa kiến trúc mạng AlexNet

5. Ví dụ áp dụng: Xây dựng mạng AlexNet

```
[133] model3 = Sequential()
      model3.add(Conv2D(filters=96, kernel_size=(11,11),strides=(4,4),padding='same',activation='relu',input_shape=(28,28,1)))
      model3.add(MaxPooling2D(strides=(2,2), pool_size=(3, 3),padding='same'))
      model3.add(Conv2D(filters=256, kernel_size=(5,5), padding='same',activation='relu'))
      model3.add(MaxPooling2D(strides=(2,2), pool_size=(3, 3), padding='same'))
      model3.add(Conv2D(filters=384, kernel_size=(3,3),strides=(1,1), padding='same',activation='relu'))
      model3.add(Conv2D(filters=384, kernel_size=(3,3),strides=(1,1), padding='same',activation='relu'))
      model3.add(Conv2D(filters=256, kernel_size=(3,3),strides=(1,1), padding='same',activation='relu'))
      model3.add(MaxPooling2D(strides=(2,2), pool_size=(3, 3), padding='same'))
      model3.add(Flatten())
      model3.add(Dense(4096, activation='relu'))
      model3.add(Dense(4096, activation='relu'))
      model3.add(Dense(10, activation='softmax'))
```

- Sử dụng lệnh `summary` và cho biết cấu trúc của mô hình đã xây dựng.

```

✓ 0 giây
▶ model3.summary()
✕ Model: "sequential_13"

```

Layer (type)	Output Shape	Param #
conv2d_32 (Conv2D)	(None, 7, 7, 96)	11712
max_pooling2d_20 (MaxPooling2D)	(None, 4, 4, 96)	0
conv2d_33 (Conv2D)	(None, 4, 4, 256)	614656
max_pooling2d_21 (MaxPooling2D)	(None, 2, 2, 256)	0
conv2d_34 (Conv2D)	(None, 2, 2, 384)	885120
conv2d_35 (Conv2D)	(None, 2, 2, 384)	1327488
conv2d_36 (Conv2D)	(None, 2, 2, 256)	884992
max_pooling2d_22 (MaxPooling2D)	(None, 1, 1, 256)	0
flatten_13 (Flatten)	(None, 256)	0
dense_25 (Dense)	(None, 4096)	1052672
dense_26 (Dense)	(None, 4096)	16781312
dense_27 (Dense)	(None, 10)	40970

```

=====
Total params: 21,598,922
Trainable params: 21,598,922
Non-trainable params: 0

```

- Cho biết tổng số tham số (Total params) của mô hình đã xây dựng:
Trả lời: 21598922

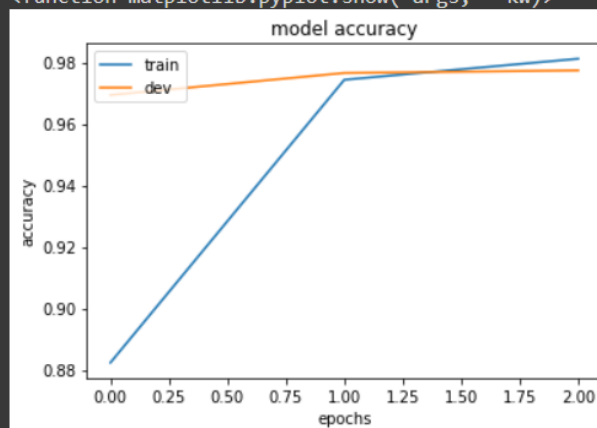
- Tiến hành huấn luyện mô hình với các thông số dưới đây, sau đó vẽ đồ thị học với accuracy và loss.
 - Hàm loss: CategoricalCrossentropy.
 - Hàm tối ưu: Adam, learning_rate=5e-4.
 - Độ đo đánh giá: Accuracy.
 - Batch_size: 256.
 - Epochs: 30.
- Trong quá trình huấn luyện có sử dụng kĩ thuật Early Stopping với các thông số sau:
 - monitor='val_loss'.
 - patience=1.

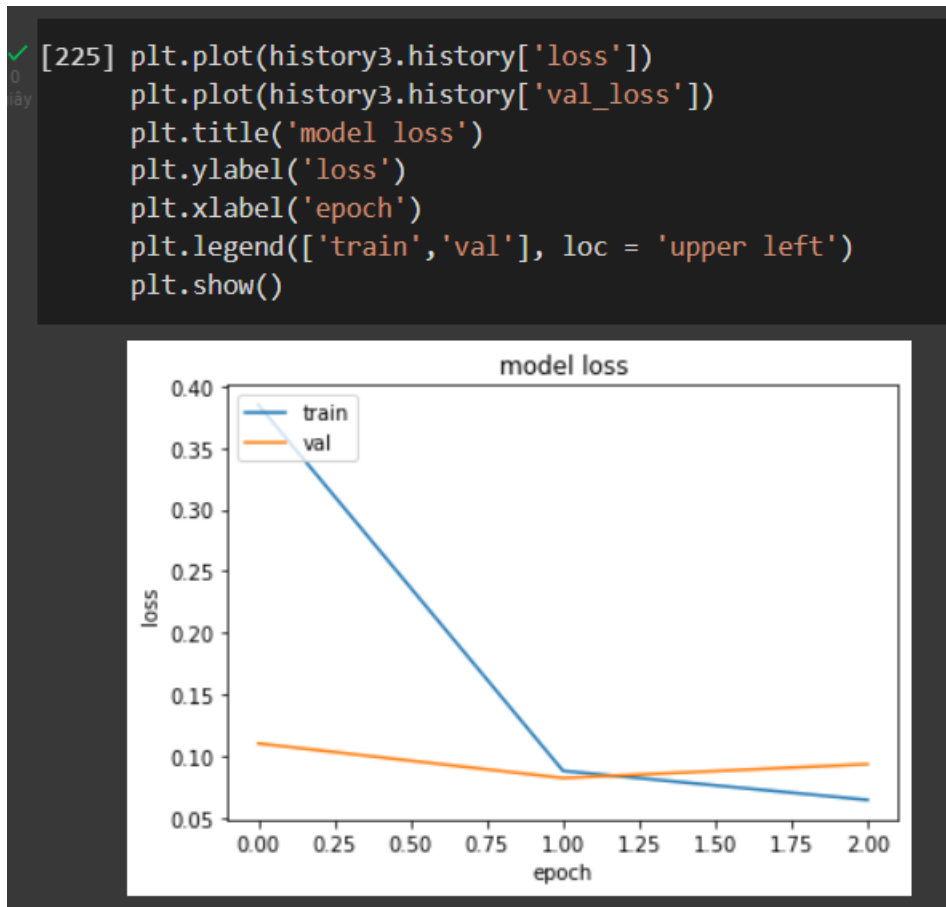
```
✓ [135] optimizer = Adam(learning_rate=5e-4)
0 giây
      loss = CategoricalCrossentropy()
      model3.compile(optimizer=optimizer, loss=loss, metrics=['accuracy'])
```

```
✓ [136] from keras.callbacks import EarlyStopping
0 giây
      callback =EarlyStopping (monitor='val_loss',patience=1)
```

```
✓ [224] import matplotlib.pyplot as plt
      plt.plot(history3.history['accuracy'])
      plt.plot(history3.history['val_accuracy'])
      plt.title('model accuracy')
      plt.ylabel('accuracy')
      plt.xlabel('epochs')
      plt.legend(['train','dev'],loc='upper left')
      plt.show
```

<function matplotlib.pyplot.show(*args, **kw)>





- **Đánh giá mô hình trên tập test và cho biết độ chính xác dự đoán.**

```
[226] y_pred = model3.predict(x_test_expanded)
      y_pred_label = np.argmax(y_pred,axis=-1)

      313/313 [=====] - 1s 3ms/step

[227] accuracy_score(y_test, y_pred_label)*100

      97.91
```

Trả lời: Độ chính xác là 97,91%

- Cho biết quá trình huấn luyện dừng lại tại epoch thứ mấy ? Bộ tham số của mô hình đạt giá trị tối ưu tại epoch thứ mấy ?

```
[223] history3 = model3.fit(X_train_expanded, y_train_new,
    batch_size=128, epochs=30,
    validation_data=(X_dev_expanded, y_dev_new), callbacks=callback)

Epoch 1/30
422/422 [=====] - 10s 22ms/step - loss: 0.3847 - accuracy: 0.8823 - val_loss: 0.1106 - val_accuracy: 0.9693
Epoch 2/30
422/422 [=====] - 8s 20ms/step - loss: 0.0884 - accuracy: 0.9743 - val_loss: 0.0826 - val_accuracy: 0.9765
Epoch 3/30
422/422 [=====] - 8s 19ms/step - loss: 0.0648 - accuracy: 0.9811 - val_loss: 0.0938 - val_accuracy: 0.9773
```

Quá trình huấn luyện dừng lại ở epoch thứ 3.

Bộ tham số của mô hình đạt giá trị tối ưu tại epoch thứ 2.