

DS201 - Deep Learning trong Khoa học dữ liệu

Lab 3. MẠNG NEURAL TÍCH CHẬP (CNN)

MSSV: 20521196

Họ tên: Nguyễn Mạnh Đức

Lớp: DS201.N11.1

0. Khởi động

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

⇒ MobileNetV1, VGG, Highway Networks, Wide ResNet, PolyNet, Pyramidal Net, ZfNet, Depth based CNNs, DenseNet

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

⇒ Lớp Convolutional (Kernel).

⇒ Lớp Pooling.

⇒ Lớp Dense (Fully-connected).

1. Các loại lớp trong mạng neural tích chập

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

The base Layer class: Layer class, weights property, trainable_weights property, non_trainable_weights property, add_weight method, trainable property, get_weights method, set_weights method, get_config method, add_loss method, add_metric method, losses property, metrics property, dynamic property.

Layer activations: relu function, sigmoid function, softmax function, softplus function, softsign function, tanh function, selu function, elu function, exponential function.

Layer weight initializers: RandomNormal class, RandomUniform class, TruncatedNormal class, Zeros class, Ones class, GlorotNormal class, GlorotUniform class, HeNormal class, HeUniform class, Identity class, Orthogonal class, Constant class, VarianceScaling class.

Layer weight regularizers: L1 class, L2 class, L1L2 class, OrthogonalRegularizer class.

Layer weight constraints: MaxNorm class, MinMaxNorm class, NonNeg class, UnitNorm class, RadialConstraint class.

Core layers: Input object, Dense layer, Activation layer, Embedding layer, Masking layer, Lambda layer.

Convolution layers: Conv1D layer, Conv2D layer, Conv3D layer, SeparableConv1D layer, SeparableConv2D layer, DepthwiseConv2D layer, Conv1DTranspose layer, Conv2DTranspose layer, Conv3DTranspose layer.

Pooling layers: MaxPooling1D layer, MaxPooling2D layer, MaxPooling3D layer, AveragePooling1D layer, AveragePooling2D layer, AveragePooling3D layer, GlobalMaxPooling1D layer, GlobalMaxPooling2D layer, GlobalMaxPooling3D layer, GlobalAveragePooling1D layer, GlobalAveragePooling2D layer, GlobalAveragePooling3D layer.

Recurrent layers: LSTM layer, GRU layer, SimpleRNN layer, TimeDistributed layer, Bidirectional layer, ConvLSTM1D layer, ConvLSTM2D layer, ConvLSTM3D layer, Base RNN layer.

Preprocessing layers: Text preprocessing, Numerical features preprocessing layers, Categorical features preprocessing layers, Image preprocessing layers, Image augmentation layers.

Normalization layers: BatchNormalization layer, LayerNormalization layer, UnitNormalization layer.

Regularization layers: Dropout layer, SpatialDropout1D layer, SpatialDropout2D layer, SpatialDropout3D layer, GaussianDropout layer, GaussianNoise layer, ActivityRegularization layer, AlphaDropout layer,

Attention layers: MultiHeadAttention layer, Attention layer, AdditiveAttention layer.

Reshaping layers: Reshape layer, Flatten layer, RepeatVector layer, Permute layer, Cropping1D layer, Cropping2D layer, Cropping3D layer, UpSampling1D layer, UpSampling2D layer, UpSampling3D layer, ZeroPadding1D layer, ZeroPadding2D layer, ZeroPadding3D layer.

Merging layers: Concatenate layer, Average layer, Maximum layer, Minimum layer, Add layer, Subtract layer, Multiply layer, Dot layer.

Locally-connected layers: LocallyConnected1D layer, LocallyConnected2D layer.

Activation layers: ReLU layer, Softmax layer, LeakyReLU layer, PReLU layer, ELU layer, ThresholdedReLU layer.

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 ?

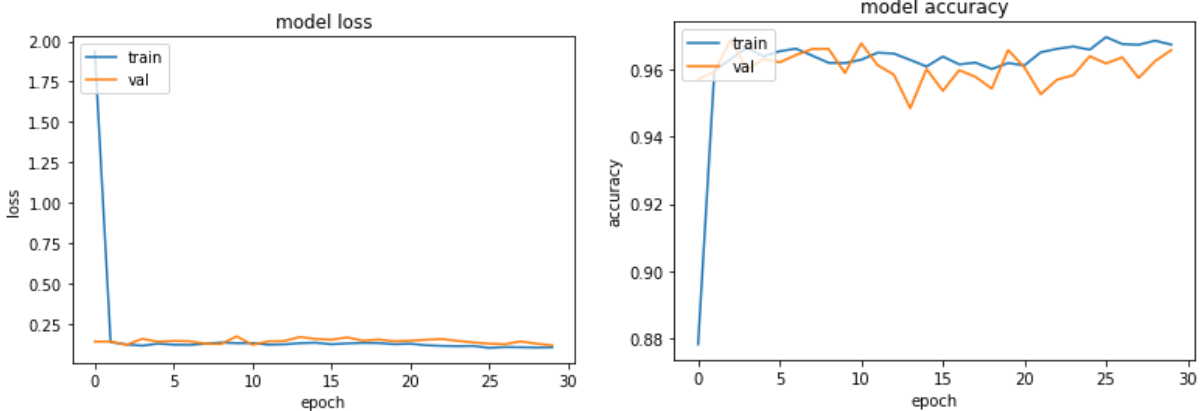
📄 Model: "sequential"

Layer (type)	Output Shape	Param #
=====		
conv2d (Conv2D)	(None, 26, 26, 32)	320
max_pooling2d (MaxPooling2D)	(None, 13, 13, 32)	0
conv2d_1 (Conv2D)	(None, 11, 11, 64)	18496
max_pooling2d_1 (MaxPooling2D)	(None, 5, 5, 64)	0
flatten (Flatten)	(None, 1600)	0
dense (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 ?

⇒ **34826**

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



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

```
313/313 [=====] - 1s 2ms/step  
96.23
```

⇒ **96.23%**

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

⇒ **1**

4. Xây dựng mạng LeNet-5

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

Model: "sequential_9"

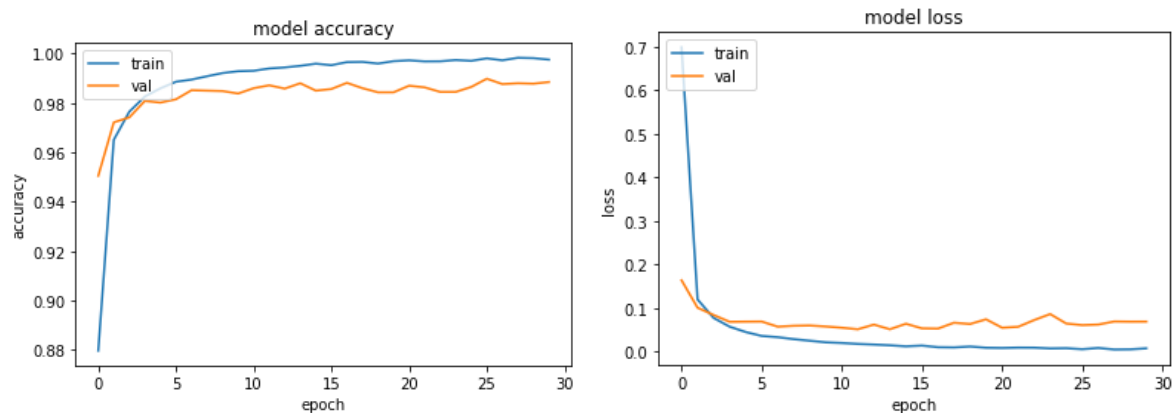
Layer (type)	Output Shape	Param #
conv2d_18 (Conv2D)	(None, 28, 28, 6)	156
average_pooling2d_8 (Average Pooling2D)	(None, 14, 14, 6)	0
conv2d_19 (Conv2D)	(None, 10, 10, 16)	2416
average_pooling2d_9 (Average Pooling2D)	(None, 5, 5, 16)	0
flatten_9 (Flatten)	(None, 400)	0
dense_17 (Dense)	(None, 120)	48120
dense_18 (Dense)	(None, 84)	10164
dense_19 (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 ?

⇒ **61706**

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



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

313/313 [=====] - 1s 2ms/step
98.85000000000001


⇒ 98.85%

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

⇒ 1

5. Ví dụ áp dụng: Xây dựng mạng AlexNet

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

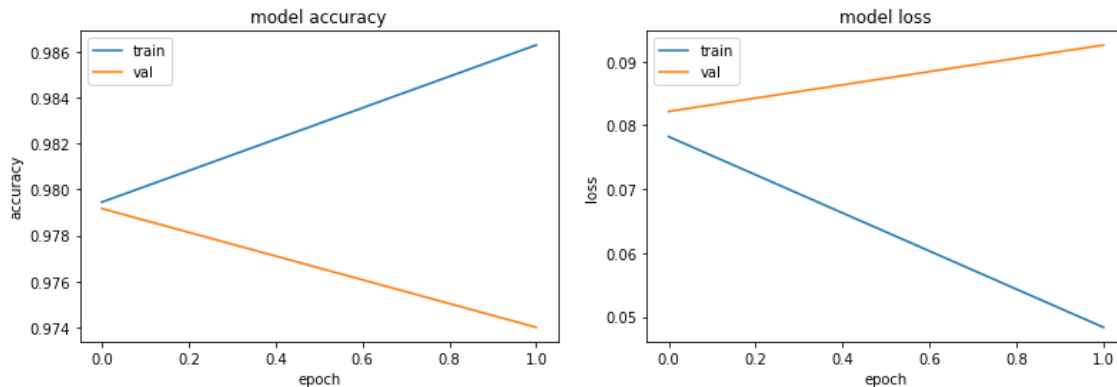
0s  Model: "sequential_10"

Layer (type)	Output Shape	Param #
=====		
conv2d_20 (Conv2D)	(None, 7, 7, 96)	11712
max_pooling2d_10 (MaxPooling2D)	(None, 4, 4, 96)	0
conv2d_21 (Conv2D)	(None, 4, 4, 256)	614656
max_pooling2d_11 (MaxPooling2D)	(None, 2, 2, 256)	0
conv2d_22 (Conv2D)	(None, 2, 2, 384)	885120
conv2d_23 (Conv2D)	(None, 2, 2, 384)	1327488
conv2d_24 (Conv2D)	(None, 2, 2, 256)	884992
max_pooling2d_12 (MaxPooling2D)	(None, 1, 1, 256)	0
flatten_10 (Flatten)	(None, 256)	0
dense_20 (Dense)	(None, 4096)	1052672
dense_21 (Dense)	(None, 4096)	16781312
dense_22 (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 ?

⇒ 21598922

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



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

✕ 313/313 [=====] - 1s 3ms/step
97.68

⇒ 97.68%

(?) 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 ?

```
Epoch 1/30  
211/211 [=====] - 7s 31ms/step - loss: 0.0782 - accuracy: 0.9794 - val_loss: 0.0822 - val_accuracy: 0.9792  
Epoch 2/30  
211/211 [=====] - 6s 29ms/step - loss: 0.0483 - accuracy: 0.9863 - val_loss: 0.0926 - val_accuracy: 0.9740
```

⇒ Thứ 2

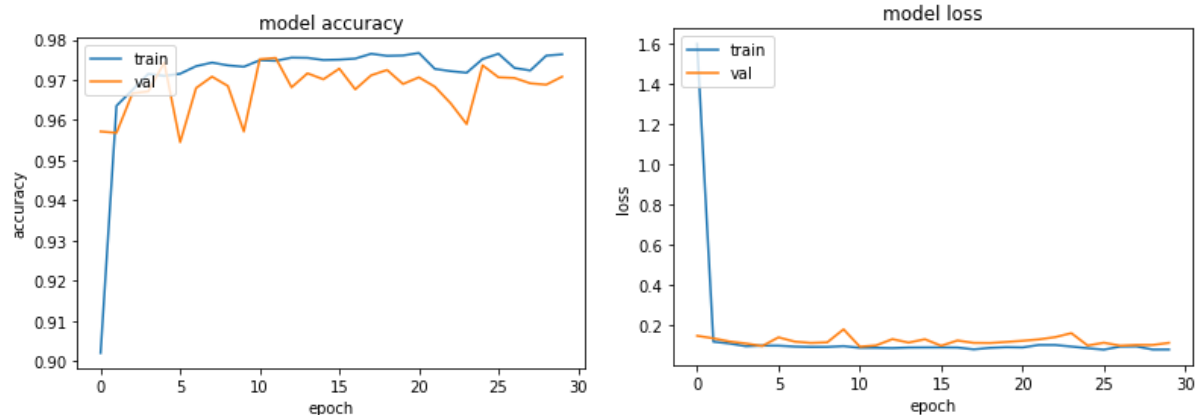
(?) Tổng xác suất của các output tại lớp Softmax là bao nhiêu ?

⇒ 1

BÀI TẬP

Câu 1

- Xây dựng lại mô hình mạng CNN như hướng dẫn ở Mục 3, tuy nhiên sử dụng hàm kích hoạt ReLU cho hai lớp CONV.
- Tiến hành huấn luyện mô hình, sau đó vẽ đồ thị học với accuracy và loss.



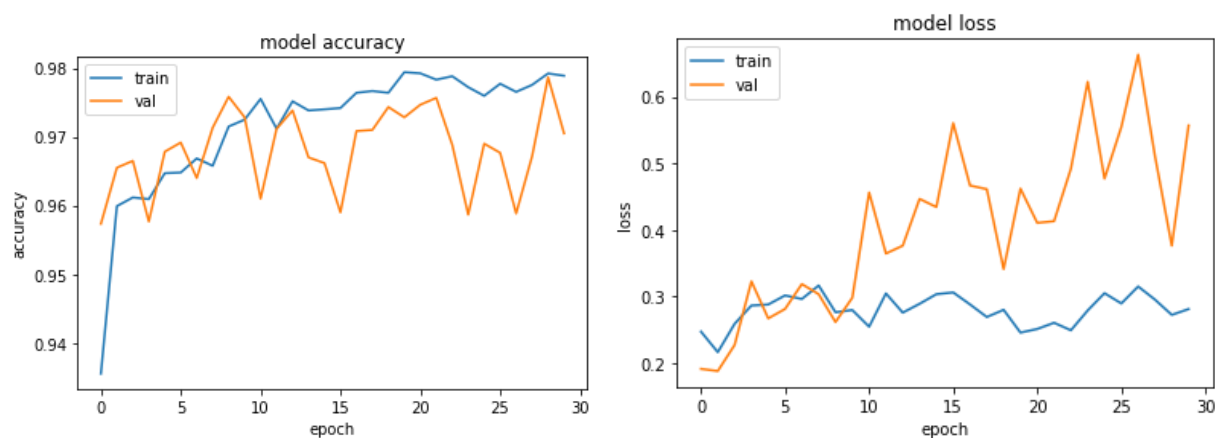
c. Đánh giá mô hình trên tập test và cho biết độ chính xác dự đoán.

```
313/313 [=====] - 1s 3ms/step
97.3
```

Bài tập 2

a. Xây dựng lại mô hình mạng CNN như hướng dẫn ở Mục 3, tuy nhiên sử dụng hàm kích hoạt Tanh cho hai lớp CONV.

b. Tiến hành huấn luyện mô hình, sau đó vẽ đồ thị học với accuracy và loss.



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

```
313/313 [=====] - 1s 2ms/step
97.46000000000001
```

Bài tập 3

a. Thực hiện tương tự như Bài tập 1, nhưng thêm lớp Dropout với giá trị $p = 0.5$ vào trước lớp Dense.

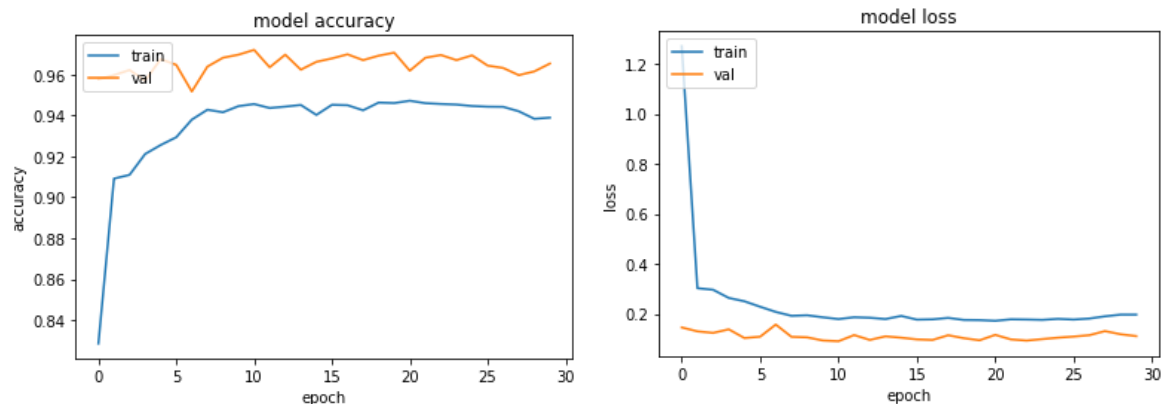
b. Sử dụng lệnh summary và cho biết cấu trúc của mô hình đã xây dựng.

Model: "sequential_14"

Layer (type)	Output Shape	Param #
conv2d_31 (Conv2D)	(None, 26, 26, 32)	320
max_pooling2d_19 (MaxPooling2D)	(None, 13, 13, 32)	0
conv2d_32 (Conv2D)	(None, 11, 11, 64)	18496
max_pooling2d_20 (MaxPooling2D)	(None, 5, 5, 64)	0
flatten_14 (Flatten)	(None, 1600)	0
dropout (Dropout)	(None, 1600)	0
dense_25 (Dense)	(None, 10)	16010

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

c. Tiến hành huấn luyện mô hình, sau đó vẽ đồ thị học với accuracy và loss.



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

313/313 [=====] - 1s 3ms/step
96.66

e. Lưu lại mô hình thành file có định dạng .h5.

BÀI TẬP MỞ RỘNG

Bài tập 4*

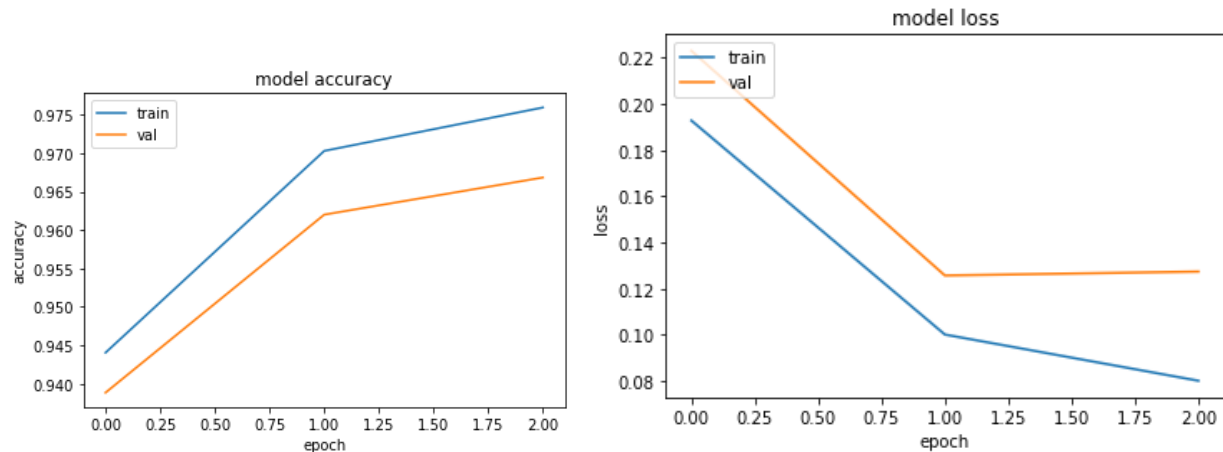
a. Tìm hiểu cấu trúc và xây dựng lại mạng ZFNET.

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

Model: "sequential_15"

Layer (type)	Output Shape	Param #
conv1 (Conv2D)	(None, 11, 11, 96)	4800
pool1 (MaxPooling2D)	(None, 6, 6, 96)	0
bn_conv1 (BatchNormalizatio n)	(None, 6, 6, 96)	384
conv2 (Conv2D)	(None, 1, 1, 256)	614656
pool2 (MaxPooling2D)	(None, 1, 1, 256)	0
bn_conv2 (BatchNormalizatio n)	(None, 1, 1, 256)	1024
conv3 (Conv2D)	(None, 1, 1, 512)	1180160
conv4 (Conv2D)	(None, 1, 1, 1024)	4719616
conv5 (Conv2D)	(None, 1, 1, 512)	4719104
pool3 (MaxPooling2D)	(None, 1, 1, 512)	0
flatten_15 (Flatten)	(None, 512)	0
dense_26 (Dense)	(None, 4096)	2101248
dense_27 (Dense)	(None, 4096)	16781312
dense_28 (Dense)	(None, 10)	40970
=====		
Total params: 30,163,274		
Trainable params: 30,162,570		
Non-trainable params: 704		

c. Tiến hành huấn luyện mô hình với các thông số dưới đây, sau đó vẽ đồ thị học với accuracy và loss.



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

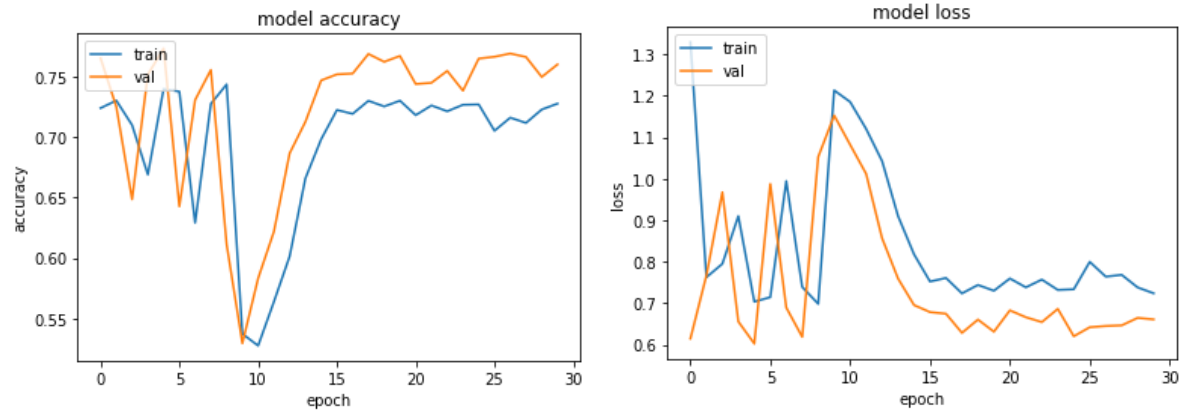
```
➡ 313/313 [=====] - 2s 6ms/step
96.59
```

Bài tập 5*

a. Thực hiện tương tự Bài tập 3 (Không cần thực hiện câu e), nhưng với bộ dữ liệu Fashion MNIST.

```
➡ Model: "sequential_16"
```

Layer (type)	Output Shape	Param #
=====		
conv2d_33 (Conv2D)	(None, 26, 26, 32)	320
max_pooling2d_21 (MaxPooling2D)	(None, 13, 13, 32)	0
conv2d_34 (Conv2D)	(None, 11, 11, 64)	18496
max_pooling2d_22 (MaxPooling2D)	(None, 5, 5, 64)	0
flatten_16 (Flatten)	(None, 1600)	0
dropout_1 (Dropout)	(None, 1600)	0
dense_29 (Dense)	(None, 10)	16010
=====		
Total params: 34,826		
Trainable params: 34,826		
Non-trainable params: 0		



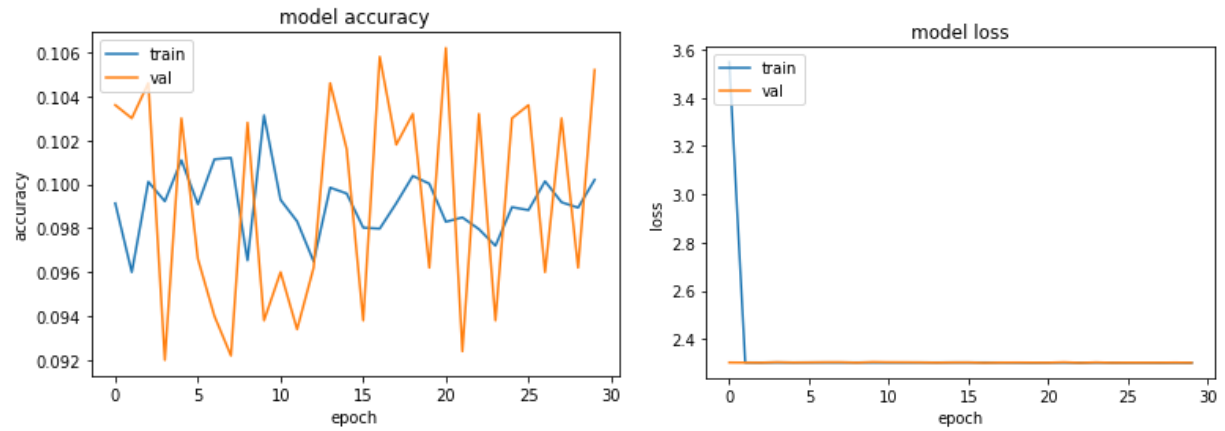
313/313 [=====] - 1s 2ms/step
75.69

b. Thực hiện tương tự Bài tập 3 (Không cần thực hiện câu e), nhưng với bộ dữ liệu CIFAR-10.

Model: "sequential_18"

Layer (type)	Output Shape	Param #
conv2d_37 (Conv2D)	(None, 30, 30, 32)	896
max_pooling2d_25 (MaxPooling2D)	(None, 15, 15, 32)	0
conv2d_38 (Conv2D)	(None, 13, 13, 64)	18496
max_pooling2d_26 (MaxPooling2D)	(None, 6, 6, 64)	0
flatten_18 (Flatten)	(None, 2304)	0
dropout_3 (Dropout)	(None, 2304)	0
dense_31 (Dense)	(None, 10)	23050

=====
Total params: 42,442
Trainable params: 42,442
Non-trainable params: 0



313/313 [=====] - 1s 2ms/step
10.05

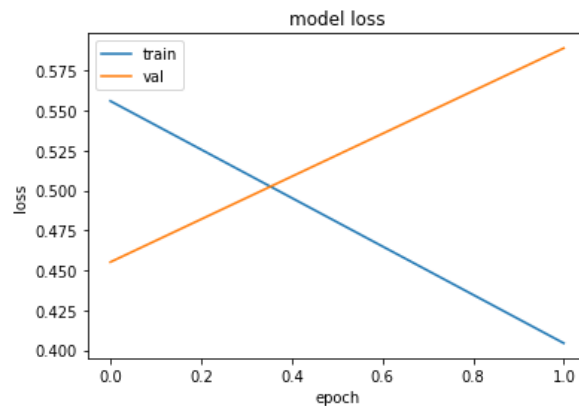
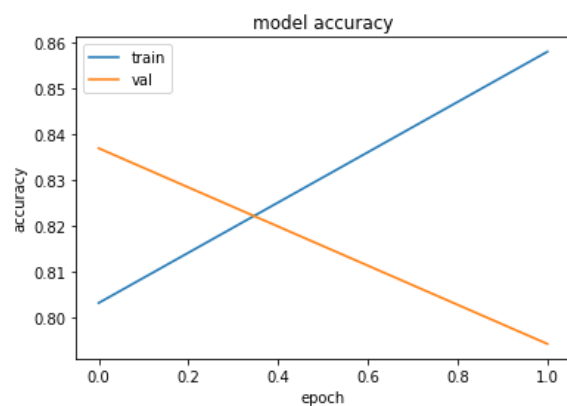
Bài tập 6*

- Thực hiện tương tự Bài tập 4, nhưng với bộ dữ liệu Fashion MNIST.

Model: "sequential_19"

Layer (type)	Output Shape	Param #
conv1 (Conv2D)	(None, 11, 11, 96)	4800
pool1 (MaxPooling2D)	(None, 6, 6, 96)	0
bn_conv1 (BatchNormalization)	(None, 6, 6, 96)	384
conv2 (Conv2D)	(None, 1, 1, 256)	614656
pool2 (MaxPooling2D)	(None, 1, 1, 256)	0
bn_conv2 (BatchNormalization)	(None, 1, 1, 256)	1024
conv3 (Conv2D)	(None, 1, 1, 512)	1180160
conv4 (Conv2D)	(None, 1, 1, 1024)	4719616
conv5 (Conv2D)	(None, 1, 1, 512)	4719104
pool3 (MaxPooling2D)	(None, 1, 1, 512)	0
flatten_19 (Flatten)	(None, 512)	0
dense_32 (Dense)	(None, 4096)	2101248
dense_33 (Dense)	(None, 4096)	16781312
dense_34 (Dense)	(None, 10)	40970

=====
Total params: 30,163,274
Trainable params: 30,162,570
Non-trainable params: 704



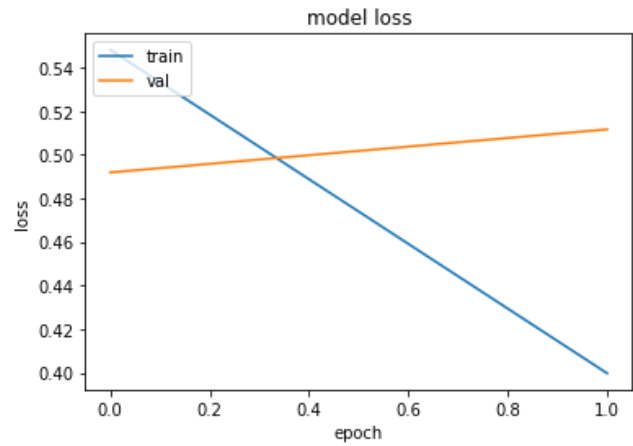
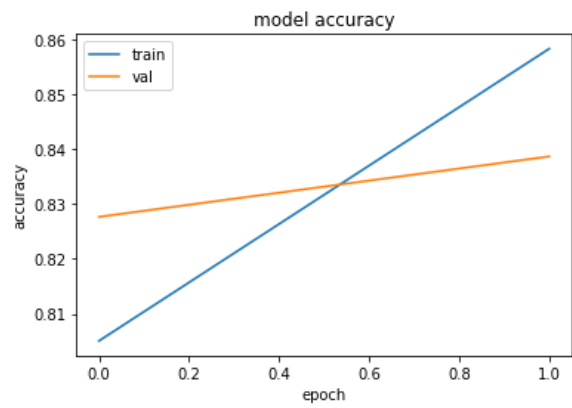
```
313/313 [=====] - 2s 5ms/step
78.81
```

b. Thực hiện tương tự Bài tập 4, nhưng với bộ dữ liệu CIFAR-10.

```
Model: "sequential_20"
```

Layer (type)	Output Shape	Param #
conv1 (Conv2D)	(None, 11, 11, 96)	4800
pool1 (MaxPooling2D)	(None, 6, 6, 96)	0
bn_conv1 (BatchNormalizatio n)	(None, 6, 6, 96)	384
conv2 (Conv2D)	(None, 1, 1, 256)	614656
pool2 (MaxPooling2D)	(None, 1, 1, 256)	0
bn_conv2 (BatchNormalizatio n)	(None, 1, 1, 256)	1024
conv3 (Conv2D)	(None, 1, 1, 512)	1180160
conv4 (Conv2D)	(None, 1, 1, 1024)	4719616
conv5 (Conv2D)	(None, 1, 1, 512)	4719104
pool3 (MaxPooling2D)	(None, 1, 1, 512)	0
flatten_20 (Flatten)	(None, 512)	0
dense_35 (Dense)	(None, 4096)	2101248
dense_36 (Dense)	(None, 4096)	16781312
dense_37 (Dense)	(None, 10)	40970

```
=====  
Total params: 30,163,274  
Trainable params: 30,162,570  
Non-trainable params: 704
```



313/313 [=====] - 2s 6ms/step
83.08