图像识别

In [1]:

```
import keras
from keras.datasets import cifar10
from keras.models import Sequential
from keras.layers import Dense, Dropout, Flatten, Conv2D, MaxPooling2D
from pathlib import Path

import matplotlib.pyplot as plt
%matplotlib inline
```

```
Using TensorFlow backend.
```

In [2]:

```
# 加载数据集
(x_train, y_train), (x_test, y_test) = cifar10.load_data()

# 数据集归一化(0-1之间)
x_train = x_train.astype('float32')
x_test = x_test.astype('float32')
x_train /= 255
x_test /= 255
```

In [3]:

```
# 标签转换为 One-Hot 编码
y_train = keras.utils.to_categorical(y_train, 10)
y_test = keras.utils.to_categorical(y_test, 10)
```

In [4]:

```
# 定义CNN网络结构
model = Sequential()

model.add(Conv2D(32, (3, 3), padding='same', input_shape=(32, 32, 3), activation="relu"))
model.add(Conv2D(32, (3, 3), activation="relu"))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))

model.add(Conv2D(64, (3, 3), padding='same', activation="relu"))
model.add(Conv2D(64, (3, 3), activation="relu"))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Flatten())
model.add(Dropout(0.25))

model.add(Dense(512, activation="relu"))
model.add(Dropout(0.5))

model.add(Dense(10, activation="softmax"))
```

In [5]:

```
# 编译CNN模型
model.compile(
    loss='categorical_crossentropy',
    optimizer='adam',
    metrics=['accuracy']
)
model.summary()
```

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 32, 32, 32)	896
conv2d_2 (Conv2D)	(None, 30, 30, 32)	9248
max_pooling2d_1 (MaxPooling2	(None, 15, 15, 32)	0
dropout_1 (Dropout)	(None, 15, 15, 32)	0
conv2d_3 (Conv2D)	(None, 15, 15, 64)	18496
conv2d_4 (Conv2D)	(None, 13, 13, 64)	36928
max_pooling2d_2 (MaxPooling2	(None, 6, 6, 64)	0
flatten_1 (Flatten)	(None, 2304)	0
dropout_2 (Dropout)	(None, 2304)	0
dense_1 (Dense)	(None, 512)	1180160
dropout_3 (Dropout)	(None, 512)	0
dense_2 (Dense)	(None, 10)	5130

Total params: 1,250,858
Trainable params: 1,250,858
Non-trainable params: 0

In [6]:

```
# 训练CNN模型
history = model.fit(
    x_train,
    y_train,
    batch_size=64,
    epochs=30,
    validation_data=(x_test, y_test),
    shuffle=True
)
```

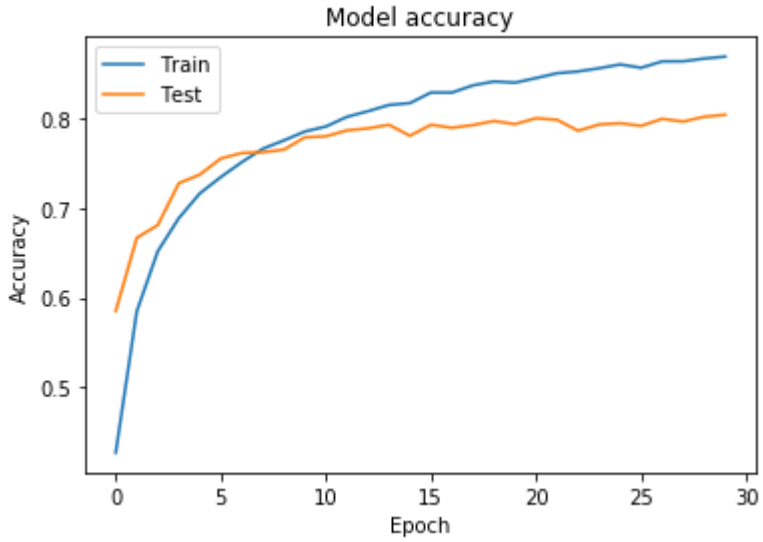
```
Train on 50000 samples, validate on 10000 samples
Epoch 1/30
50000/50000 [==============================] - 129s 3ms/step - loss: 1.5606 - acc:
0.4269 - val_loss: 1.1651 - val_acc: 0.5851
Epoch 2/30
50000/50000 [==============================] - 128s 3ms/step - loss: 1.1701 - acc:
0.5845 - val_loss: 0.9537 - val_acc: 0.6667
Epoch 3/30
50000/50000 [==============================] - 129s 3ms/step - loss: 0.9903 - acc:
0.6517 - val_loss: 0.9160 - val_acc: 0.6809
Epoch 4/30
50000/50000 [==============================] - 128s 3ms/step - loss: 0.8856 - acc:
0.6886 - val_loss: 0.8036 - val_acc: 0.7276
Epoch 5/30
50000/50000 [==============================] - 128s 3ms/step - loss: 0.8081 - acc:
0.7163 - val_loss: 0.7633 - val_acc: 0.7373
Epoch 6/30
50000/50000 [==============================] - 128s 3ms/step - loss: 0.7533 - acc:
0.7346 - val_loss: 0.7121 - val_acc: 0.7552
Epoch 7/30
50000/50000 [==============================] - 128s 3ms/step - loss: 0.7075 - acc:
0.7512 - val_loss: 0.6923 - val_acc: 0.7614
Epoch 8/30
50000/50000 [==============================] - 127s 3ms/step - loss: 0.6626 - acc:
0.7661 - val_loss: 0.7022 - val_acc: 0.7622
Epoch 9/30
50000/50000 [==============================] - 127s 3ms/step - loss: 0.6394 - acc:
0.7754 - val_loss: 0.6910 - val_acc: 0.7651
Epoch 10/30
50000/50000 [==============================] - 127s 3ms/step - loss: 0.6102 - acc:
0.7854 - val_loss: 0.6560 - val_acc: 0.7789
Epoch 11/30
50000/50000 [==============================] - 128s 3ms/step - loss: 0.5913 - acc:
0.7910 - val_loss: 0.6373 - val_acc: 0.7800
Epoch 12/30
50000/50000 [==============================] - 128s 3ms/step - loss: 0.5630 - acc:
0.8017 - val_loss: 0.6324 - val_acc: 0.7866
Epoch 13/30
50000/50000 [==============================] - 128s 3ms/step - loss: 0.5472 - acc:
0.8080 - val_loss: 0.6208 - val_acc: 0.7889
Epoch 14/30
50000/50000 [==============================] - 127s 3ms/step - loss: 0.5256 - acc:
0.8150 - val_loss: 0.6246 - val_acc: 0.7928
Epoch 15/30
50000/50000 [==============================] - 126s 3ms/step - loss: 0.5075 - acc:
0.8171 - val_loss: 0.6732 - val_acc: 0.7806
Epoch 16/30
50000/50000 [==============================] - 125s 3ms/step - loss: 0.4896 - acc:
```

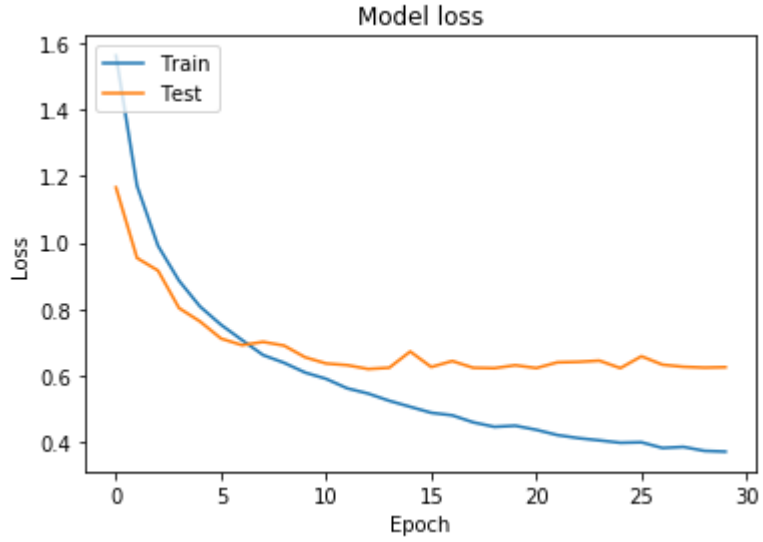
```
0.8289 - val_loss: 0.6269 - val_acc: 0.7929
Epoch 17/30
50000/50000 [==============================] - 126s 3ms/step - loss: 0.4820 - acc:
0.8288 - val_loss: 0.6447 - val_acc: 0.7895
Epoch 18/30
50000/50000 [==============================] - 125s 3ms/step - loss: 0.4609 - acc:
0.8369 - val_loss: 0.6244 - val_acc: 0.7926
Epoch 19/30
50000/50000 [==============================] - 125s 3ms/step - loss: 0.4475 - acc:
0.8412 - val_loss: 0.6233 - val_acc: 0.7971
Epoch 20/30
50000/50000 [==============================] - 125s 2ms/step - loss: 0.4508 - acc:
0.8399 - val_loss: 0.6321 - val_acc: 0.7936
Epoch 21/30
50000/50000 [==============================] - 126s 3ms/step - loss: 0.4385 - acc:
0.8450 - val_loss: 0.6236 - val_acc: 0.8002
Epoch 22/30
50000/50000 [==============================] - 125s 3ms/step - loss: 0.4228 - acc:
0.8504 - val_loss: 0.6408 - val_acc: 0.7984
Epoch 23/30
50000/50000 [==============================] - 125s 3ms/step - loss: 0.4136 - acc:
0.8524 - val_loss: 0.6422 - val_acc: 0.7862
Epoch 24/30
50000/50000 [==============================] - 126s 3ms/step - loss: 0.4068 - acc:
0.8560 - val_loss: 0.6458 - val_acc: 0.7932
Epoch 25/30
50000/50000 [==============================] - 129s 3ms/step - loss: 0.3997 - acc:
0.8602 - val_loss: 0.6235 - val_acc: 0.7946
Epoch 26/30
50000/50000 [==============================] - 128s 3ms/step - loss: 0.4011 - acc:
0.8565 - val_loss: 0.6590 - val_acc: 0.7917
Epoch 27/30
50000/50000 [==============================] - 128s 3ms/step - loss: 0.3840 - acc:
0.8636 - val_loss: 0.6338 - val_acc: 0.7995
Epoch 28/30
50000/50000 [==============================] - 127s 3ms/step - loss: 0.3873 - acc:
0.8638 - val_loss: 0.6274 - val_acc: 0.7965
Epoch 29/30
50000/50000 [==============================] - 127s 3ms/step - loss: 0.3753 - acc:
0.8668 - val_loss: 0.6250 - val_acc: 0.8018
Epoch 30/30
50000/50000 [==============================] - 127s 3ms/step - loss: 0.3730 - acc:
0.8690 - val_loss: 0.6263 - val_acc: 0.8041
```

In [7]:

```
# 模型在训练集和验证集上的预测准确率变化曲线
plt.plot(history.history['acc'])
plt.plot(history.history['val_acc'])
plt.title('Model accuracy')
plt.ylabel('Accuracy')
plt.xlabel('Epoch')
plt.legend(['Train', 'Test'], loc='upper left')
plt.savefig("accuracy.png")
plt.show()

# 模型在训练集和验证集上的损失函数曲线
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('Model loss')
plt.ylabel('Loss')
plt.xlabel('Epoch')
plt.legend(['Train', 'Test'], loc='upper left')
plt.savefig("loss.png")
plt.show()
```





In [8]:

```
# 保存CNN网络结构
model_structure = model.to_json()
f = Path("model_structure.json")
f.write_text(model_structure)

# 保存训练好的权重参数
model.save_weights("model_weights.h5")
```