Cifar-10 实验总结

训练展示：

测试代码	网络结构	nb_epoch	loss	acc	val_loss	val_acc	训练时间/s
cifar10_cnn_dl	参考结构	50	0.7723	0.7372	0.6074	0.7926	1552
cifar10_cnn_dl3	增加两个 Dropout(0.25)	50	0.9537	0.6733	0.7407	0.7419	1616
cifar10_cnn_dl4	‘same’->’valid’	50	0.8481	0.7127	0.6728	0.7722	1345
cifar10_cnn_dl5	Dropout:0.25->0.5	50	1.0443	0.6389	0.9716	0.6649	1600
cifar10_cnn_dl6	卷积核：3*3->5*5 最后一个激活函数：’relu’->’ tanh’ SGD: lr = 0.01 -> lr = 0.0065	50	0.5928	0.7960	0.5906	0.8037	2387
cifar10_cnn_dl7	卷积核：3*3->5*5 最后一个激活函数：’relu’->’ tanh’ SGD: lr = 0.01 -> lr = 0.0035	50	0.4500	0.8430	0.4769	0.8403	2362
cifar10_cnn_dl8	卷积核：3*3->5*5 最后一个激活函数：’relu’->’ tanh’ SGD: lr = 0.01 -> lr = 0.0035 增加卷积层：第二部分	50	0.4463	0.8466	0.5207	0.8308	2677
cifar10_cnn_dl9	卷积核：3*3->5*5 最后一个激活函数：’relu’->’ tanh’ SGD: lr = 0.01 -> lr = 0.0035 增加卷积层：第二部分 增加 nb_epoch：50->100	100	0.3652	0.8751	0.4865	0.8552	5291

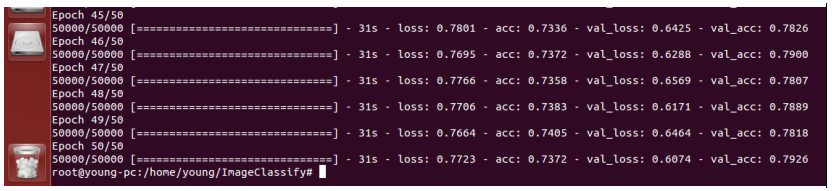
cifar10_cnn_dl 网络结构：

```
model = Sequential()

model.add(Convolution2D(32, 3, 3, border_mode='same',
                        input_shape=(img_channels, img_rows, img_cols)))
model.add(Activation('relu'))
model.add(Convolution2D(32, 3, 3))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))

model.add(Convolution2D(64, 3, 3, border_mode='same'))
model.add(Activation('relu'))
model.add(Convolution2D(64, 3, 3))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))

model.add(Flatten())
model.add(Dense(512))
model.add(Activation('relu'))
model.add(Dropout(0.5))
model.add(Dense(nb_classes))
model.add(Activation('softmax'))

# let's train the model using SGD + momentum (how original).
sgd = SGD(lr=0.01, decay=1e-6, momentum=0.9, nesterov=True)
model.compile(loss='categorical_crossentropy', optimizer=sgd)
```

cifar10_cnn_dl3 网络结构：

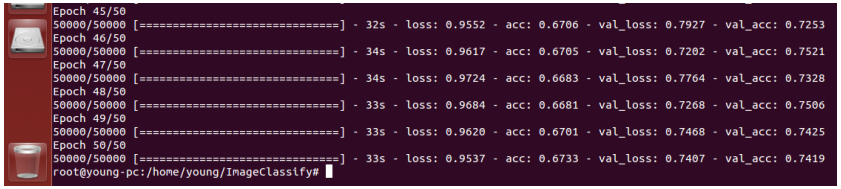
```
model = Sequential()

model.add(Convolution2D(32, 3, 3, border_mode='same',
                        input_shape=(img_channels, img_rows, img_cols)))
model.add(Activation('relu'))
model.add(Dropout(0.25))

model.add(Convolution2D(32, 3, 3))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))

model.add(Convolution2D(64, 3, 3, border_mode='same'))
model.add(Activation('relu'))
model.add(Dropout(0.25))

model.add(Convolution2D(64, 3, 3))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))

model.add(Flatten())
model.add(Dense(512))
model.add(Activation('relu'))
model.add(Dropout(0.5))
model.add(Dense(nb_classes))
model.add(Activation('softmax'))


# let's train the model using SGD + momentum (how original).
sgd = SGD(lr=0.01, decay=1e-6, momentum=0.9, nesterov=True)
model.compile(loss='categorical_crossentropy', optimizer=sgd)
```

cifar10_cnn_dl 结果：



```
Epoch 45/50
50000/50000 [==============================] - 31s - loss: 0.7801 - acc: 0.7336 - val_loss: 0.6425 - val_acc: 0.7826
Epoch 46/50
50000/50000 [==============================] - 31s - loss: 0.7695 - acc: 0.7372 - val_loss: 0.6288 - val_acc: 0.7900
Epoch 47/50
50000/50000 [==============================] - 31s - loss: 0.7766 - acc: 0.7358 - val_loss: 0.6569 - val_acc: 0.7807
Epoch 48/50
50000/50000 [==============================] - 31s - loss: 0.7706 - acc: 0.7383 - val_loss: 0.6171 - val_acc: 0.7889
Epoch 49/50
50000/50000 [==============================] - 31s - loss: 0.7664 - acc: 0.7405 - val_loss: 0.6464 - val_acc: 0.7818
Epoch 50/50
50000/50000 [==============================] - 31s - loss: 0.7723 - acc: 0.7372 - val_loss: 0.6074 - val_acc: 0.7926
root@young-pc:/home/young/ImageClassify#
```

far10_cnn_dl3 结果：



```
Epoch 45/50
50000/50000 [==============================] - 32s - loss: 0.9552 - acc: 0.6706 - val_loss: 0.7927 - val_acc: 0.7253
Epoch 46/50
50000/50000 [==============================] - 34s - loss: 0.9617 - acc: 0.6705 - val_loss: 0.7202 - val_acc: 0.7521
Epoch 47/50
50000/50000 [==============================] - 34s - loss: 0.9724 - acc: 0.6683 - val_loss: 0.7764 - val_acc: 0.7328
Epoch 48/50
50000/50000 [==============================] - 33s - loss: 0.9684 - acc: 0.6681 - val_loss: 0.7268 - val_acc: 0.7506
Epoch 49/50
50000/50000 [==============================] - 33s - loss: 0.9620 - acc: 0.6701 - val_loss: 0.7468 - val_acc: 0.7425
Epoch 50/50
50000/50000 [==============================] - 33s - loss: 0.9537 - acc: 0.6733 - val_loss: 0.7407 - val_acc: 0.7419
root@young-pc:/home/young/ImageClassify#
```

增加两个 Dropout(0.25)效果并没有原来的好。

cifar10_cnn_dl 网络结构：

```
model = Sequential()

model.add(Convolution2D(32, 3, 3, border_mode='same',
                        input_shape=(img_channels, img_rows, img_cols)))
model.add(Activation('relu'))
model.add(Convolution2D(32, 3, 3))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))

model.add(Convolution2D(64, 3, 3, border_mode='same'))
model.add(Activation('relu'))
model.add(Convolution2D(64, 3, 3))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))

model.add(Flatten())
model.add(Dense(512))
model.add(Activation('relu'))
model.add(Dropout(0.5))
model.add(Dense(nb_classes))
model.add(Activation('softmax'))

# let's train the model using SGD + momentum (how original).
sgd = SGD(lr=0.01, decay=1e-6, momentum=0.9, nesterov=True)
model.compile(loss='categorical_crossentropy', optimizer=sgd)
```

cifar10_cnn_dl4 网络结构：

```
model = Sequential()

model.add(Convolution2D(32, 3, 3, border_mode='valid',
                        input_shape=(img_channels, img_rows, img_cols)))
model.add(Activation('relu'))
model.add(Convolution2D(32, 3, 3))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))

model.add(Convolution2D(64, 3, 3, border_mode='valid'))
model.add(Activation('relu'))
model.add(Convolution2D(64, 3, 3))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))

model.add(Flatten())
model.add(Dense(512))
model.add(Activation('relu'))
model.add(Dropout(0.5))
model.add(Dense(nb_classes))
model.add(Activation('softmax'))


# let's train the model using SGD + momentum (how original).
sgd = SGD(lr=0.01, decay=1e-6, momentum=0.9, nesterov=True)
model.compile(loss='categorical_crossentropy', optimizer=sgd)
```

cifar10_cnn_dl 结果：

```
Epoch 45/50
50000/50000 [==============================] - 31s - loss: 0.7801 - acc: 0.7336 - val_loss: 0.6425 - val_acc: 0.7826
Epoch 46/50
50000/50000 [==============================] - 31s - loss: 0.7695 - acc: 0.7372 - val_loss: 0.6288 - val_acc: 0.7900
Epoch 47/50
50000/50000 [==============================] - 31s - loss: 0.7766 - acc: 0.7358 - val_loss: 0.6569 - val_acc: 0.7807
Epoch 48/50
50000/50000 [==============================] - 31s - loss: 0.7706 - acc: 0.7383 - val_loss: 0.6171 - val_acc: 0.7889
Epoch 49/50
50000/50000 [==============================] - 31s - loss: 0.7664 - acc: 0.7405 - val_loss: 0.6464 - val_acc: 0.7818
Epoch 50/50
50000/50000 [==============================] - 31s - loss: 0.7723 - acc: 0.7372 - val_loss: 0.6074 - val_acc: 0.7926
root@young-pc:/home/young/ImageClassify#
```

参数改为‘valid’效果并没有原来的好。

far10_cnn_dl4 结果：

```
Epoch 45/50
50000/50000 [==============================] - 27s - loss: 0.8531 - acc: 0.7086 - val_loss: 0.7642 - val_acc: 0.7399
Epoch 46/50
50000/50000 [==============================] - 27s - loss: 0.8503 - acc: 0.7098 - val_loss: 0.7138 - val_acc: 0.7631
Epoch 47/50
50000/50000 [==============================] - 27s - loss: 0.8571 - acc: 0.7074 - val_loss: 0.6711 - val_acc: 0.7671
Epoch 48/50
50000/50000 [==============================] - 27s - loss: 0.8458 - acc: 0.7118 - val_loss: 0.6916 - val_acc: 0.7676
Epoch 49/50
50000/50000 [==============================] - 27s - loss: 0.8563 - acc: 0.7087 - val_loss: 0.6693 - val_acc: 0.7725
Epoch 50/50
50000/50000 [==============================] - 26s - loss: 0.8481 - acc: 0.7127 - val_loss: 0.6728 - val_acc: 0.7722
root@young-pc:/home/young/ImageClassify#
```

cifar10_cnn_dl 网络结构：

```
model = Sequential()

model.add(Convolution2D(32, 3, 3, border_mode='same',
                        input_shape=(img_channels, img_rows, img_cols)))
model.add(Activation('relu'))
model.add(Convolution2D(32, 3, 3))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))

model.add(Convolution2D(64, 3, 3, border_mode='same'))
model.add(Activation('relu'))
model.add(Convolution2D(64, 3, 3))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))

model.add(Flatten())
model.add(Dense(512))
model.add(Activation('relu'))
model.add(Dropout(0.5))
model.add(Dense(nb_classes))
model.add(Activation('softmax'))

# let's train the model using SGD + momentum (how original).
sgd = SGD(lr=0.01, decay=1e-6, momentum=0.9, nesterov=True)
model.compile(loss='categorical_crossentropy', optimizer=sgd)
```

cifar10_cnn_dl5 网络结构：

```
model = Sequential()

model.add(Convolution2D(32, 3, 3, border_mode='same',
                        input_shape=(img_channels, img_rows, img_cols)))
model.add(Activation('relu'))
model.add(Convolution2D(32, 3, 3))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.5))

model.add(Convolution2D(64, 3, 3, border_mode='same'))
model.add(Activation('relu'))
model.add(Convolution2D(64, 3, 3))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.5))

model.add(Flatten())
model.add(Dense(512))
model.add(Activation('relu'))
model.add(Dropout(0.5))
model.add(Dense(nb_classes))
model.add(Activation('softmax'))

# let's train the model using SGD + momentum (how original).
sgd = SGD(lr=0.01, decay=1e-6, momentum=0.9, nesterov=True)
model.compile(loss='categorical_crossentropy', optimizer=sgd)
```

cifar10_cnn_dl 结果：

```
Epoch 45/50
50000/50000 [==============================] - 31s - loss: 0.7801 - acc: 0.7336 - val_loss: 0.6425 - val_acc: 0.7826
Epoch 46/50
50000/50000 [==============================] - 31s - loss: 0.7695 - acc: 0.7372 - val_loss: 0.6288 - val_acc: 0.7900
Epoch 47/50
50000/50000 [==============================] - 31s - loss: 0.7766 - acc: 0.7358 - val_loss: 0.6569 - val_acc: 0.7807
Epoch 48/50
50000/50000 [==============================] - 31s - loss: 0.7706 - acc: 0.7383 - val_loss: 0.6171 - val_acc: 0.7889
Epoch 49/50
50000/50000 [==============================] - 31s - loss: 0.7664 - acc: 0.7405 - val_loss: 0.6464 - val_acc: 0.7818
Epoch 50/50
50000/50000 [==============================] - 31s - loss: 0.7723 - acc: 0.7372 - val_loss: 0.6074 - val_acc: 0.7926
root@young-pc:/home/young/ImageClassify#
```

参数前两个 Dropout(0.25)改为 Dropout(0.5)效果并没有原来的好。

ifar10_cnn_dl5 结果：

```
Epoch 45/50
50000/50000 [==============================] - 33s - loss: 1.0353 - acc: 0.6414 - val_loss: 1.0040 - val_acc: 0.6592
Epoch 46/50
50000/50000 [==============================] - 31s - loss: 1.0330 - acc: 0.6430 - val_loss: 0.9596 - val_acc: 0.6632
Epoch 47/50
50000/50000 [==============================] - 31s - loss: 1.0426 - acc: 0.6411 - val_loss: 1.0929 - val_acc: 0.6225
Epoch 48/50
50000/50000 [==============================] - 31s - loss: 1.0524 - acc: 0.6384 - val_loss: 0.9028 - val_acc: 0.6948
Epoch 49/50
50000/50000 [==============================] - 31s - loss: 1.0478 - acc: 0.6402 - val_loss: 0.8498 - val_acc: 0.7119
Epoch 50/50
50000/50000 [==============================] - 31s - loss: 1.0443 - acc: 0.6389 - val_loss: 0.9716 - val_acc: 0.6649
root@young-pc:/home/young/ImageClassify#
```

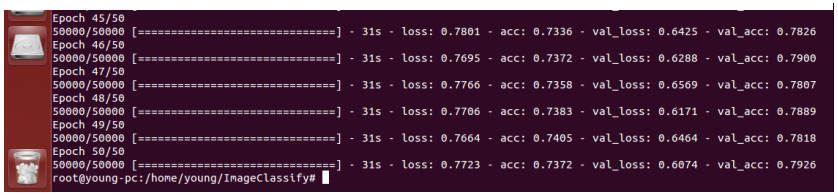
cifar10_cnn_dl 网络结构：

```
model = Sequential()

model.add(Convolution2D(32, 3, 3, border_mode='same',
                        input_shape=(img_channels, img_rows, img_cols)))
model.add(Activation('relu'))
model.add(Convolution2D(32, 3, 3))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))

model.add(Convolution2D(64, 3, 3, border_mode='same'))
model.add(Activation('relu'))
model.add(Convolution2D(64, 3, 3))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))

model.add(Flatten())
model.add(Dense(512))
model.add(Activation('relu'))
model.add(Dropout(0.5))
model.add(Dense(nb_classes))
model.add(Activation('softmax'))

# let's train the model using SGD + momentum (how original).
sgd = SGD(lr=0.01, decay=1e-6, momentum=0.9, nesterov=True)
model.compile(loss='categorical_crossentropy', optimizer=sgd)
```

cifar10_cnn_dl6 网络结构：

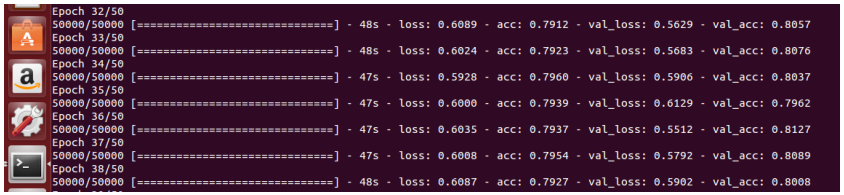
```
model = Sequential()

model.add(Convolution2D(32,5, 5, border_mode='same',
                        input_shape=(img_channels, img_rows, img_cols)))
model.add(Activation('relu'))
model.add(Convolution2D(32, 5, 5))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))

model.add(Convolution2D(64, 5, 5, border_mode='same'))
model.add(Activation('relu'))
model.add(Convolution2D(64, 5, 5))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))

model.add(Flatten())
model.add(Dense(512))
model.add(Activation('tanh'))
model.add(Dense(nb_classes))
model.add(Activation('softmax'))

# let's train the model using SGD + momentum (how original).
sgd = SGD(lr=0.0065, decay=1e-6, momentum=0.9, nesterov=True)
model.compile(loss='categorical_crossentropy', optimizer=sgd)
```

cifar10_cnn_dl 结果：



```
Epoch 45/50
50000/50000 [==============================] - 31s - loss: 0.7801 - acc: 0.7336 - val_loss: 0.6425 - val_acc: 0.7826
Epoch 46/50
50000/50000 [==============================] - 31s - loss: 0.7695 - acc: 0.7372 - val_loss: 0.6288 - val_acc: 0.7900
Epoch 47/50
50000/50000 [==============================] - 31s - loss: 0.7766 - acc: 0.7358 - val_loss: 0.6569 - val_acc: 0.7807
Epoch 48/50
50000/50000 [==============================] - 31s - loss: 0.7706 - acc: 0.7383 - val_loss: 0.6171 - val_acc: 0.7889
Epoch 49/50
50000/50000 [==============================] - 31s - loss: 0.7664 - acc: 0.7405 - val_loss: 0.6464 - val_acc: 0.7818
Epoch 50/50
50000/50000 [==============================] - 31s - loss: 0.7723 - acc: 0.7372 - val_loss: 0.6074 - val_acc: 0.7926
root@young-pc:/home/young/ImageClassify#
```

ifar10_cnn_dl6 结果：



```
Epoch 32/50
50000/50000 [==============================] - 48s - loss: 0.6089 - acc: 0.7912 - val_loss: 0.5629 - val_acc: 0.8057
Epoch 33/50
50000/50000 [==============================] - 48s - loss: 0.6024 - acc: 0.7923 - val_loss: 0.5683 - val_acc: 0.8076
Epoch 34/50
50000/50000 [==============================] - 47s - loss: 0.5928 - acc: 0.7960 - val_loss: 0.5906 - val_acc: 0.8037
Epoch 35/50
50000/50000 [==============================] - 47s - loss: 0.6000 - acc: 0.7939 - val_loss: 0.6129 - val_acc: 0.7962
Epoch 36/50
50000/50000 [==============================] - 47s - loss: 0.6035 - acc: 0.7937 - val_loss: 0.5512 - val_acc: 0.8127
Epoch 37/50
50000/50000 [==============================] - 47s - loss: 0.6008 - acc: 0.7954 - val_loss: 0.5792 - val_acc: 0.8089
Epoch 38/50
50000/50000 [==============================] - 48s - loss: 0.6087 - acc: 0.7927 - val_loss: 0.5902 - val_acc: 0.8008
```

修改卷积核大小为 5*5，并降低 SGD 的 lr 值为 0.0065，效果比原来的好，但是在 nb_epoch 大于 35 之后出现明显的过拟合，训练成功率也在下降。

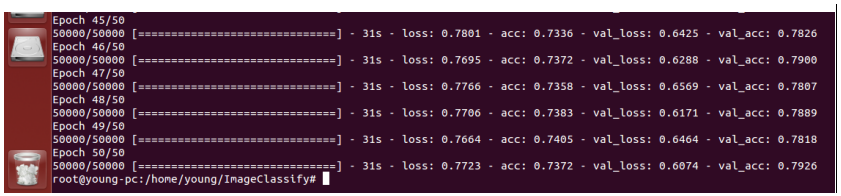
cifar10_cnn_dl 网络结构：

```
model = Sequential()

model.add(Convolution2D(32, 3, 3, border_mode='same',
                        input_shape=(img_channels, img_rows, img_cols)))
model.add(Activation('relu'))
model.add(Convolution2D(32, 3, 3))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))

model.add(Convolution2D(64, 3, 3, border_mode='same'))
model.add(Activation('relu'))
model.add(Convolution2D(64, 3, 3))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))

model.add(Flatten())
model.add(Dense(512))
model.add(Activation('relu'))
model.add(Dropout(0.5))
model.add(Dense(nb_classes))
model.add(Activation('softmax'))

# let's train the model using SGD + momentum (how original).
sgd = SGD(lr=0.01, decay=1e-6, momentum=0.9, nesterov=True)
model.compile(loss='categorical_crossentropy', optimizer=sgd)
```

cifar10_cnn_dl7 网络结构：

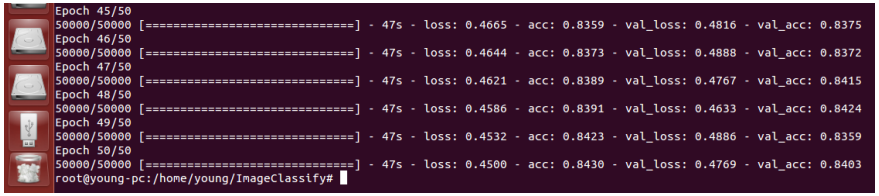
```
model = Sequential()

model.add(Convolution2D(32,5, 5, border_mode='same',
                        input_shape=(img_channels, img_rows, img_cols)))
model.add(Activation('relu'))
model.add(Convolution2D(32, 5, 5))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))

model.add(Convolution2D(64, 5, 5, border_mode='same'))
model.add(Activation('relu'))
model.add(Convolution2D(64, 5, 5))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))

model.add(Flatten())
model.add(Dense(512))
model.add(Activation('tanh'))
model.add(Dense(nb_classes))
model.add(Activation('softmax'))

# let's train the model using SGD + momentum (how original).
sgd = SGD(lr=0.0035, decay=1e-6, momentum=0.9, nesterov=True)
model.compile(loss='categorical_crossentropy', optimizer=sgd)
```

cifar10_cnn_dl 结果：



```
Epoch 45/50
50000/50000 [==============================] - 31s - loss: 0.7801 - acc: 0.7336 - val_loss: 0.6425 - val_acc: 0.7826
Epoch 46/50
50000/50000 [==============================] - 31s - loss: 0.7695 - acc: 0.7372 - val_loss: 0.6288 - val_acc: 0.7900
Epoch 47/50
50000/50000 [==============================] - 31s - loss: 0.7766 - acc: 0.7358 - val_loss: 0.6569 - val_acc: 0.7807
Epoch 48/50
50000/50000 [==============================] - 31s - loss: 0.7706 - acc: 0.7383 - val_loss: 0.6171 - val_acc: 0.7889
Epoch 49/50
50000/50000 [==============================] - 31s - loss: 0.7664 - acc: 0.7405 - val_loss: 0.6464 - val_acc: 0.7818
Epoch 50/50
50000/50000 [==============================] - 31s - loss: 0.7723 - acc: 0.7372 - val_loss: 0.6074 - val_acc: 0.7926
root@young-pc:/home/young/ImageClassify#
```

0_cnn_dl7 结果：



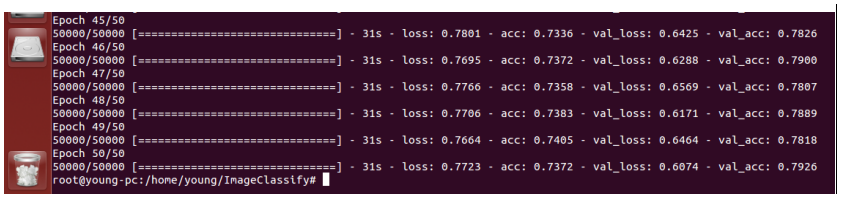
```
Epoch 45/50
50000/50000 [==============================] - 47s - loss: 0.4665 - acc: 0.8359 - val_loss: 0.4816 - val_acc: 0.8375
Epoch 46/50
50000/50000 [==============================] - 47s - loss: 0.4644 - acc: 0.8373 - val_loss: 0.4888 - val_acc: 0.8372
Epoch 47/50
50000/50000 [==============================] - 47s - loss: 0.4621 - acc: 0.8389 - val_loss: 0.4767 - val_acc: 0.8415
Epoch 48/50
50000/50000 [==============================] - 47s - loss: 0.4586 - acc: 0.8391 - val_loss: 0.4633 - val_acc: 0.8424
Epoch 49/50
50000/50000 [==============================] - 47s - loss: 0.4532 - acc: 0.8423 - val_loss: 0.4886 - val_acc: 0.8359
Epoch 50/50
50000/50000 [==============================] - 47s - loss: 0.4500 - acc: 0.8430 - val_loss: 0.4769 - val_acc: 0.8403
root@young-pc:/home/young/ImageClassify#
```

修改卷积核大小为 5*5，并降低 SGD 的 lr 值为 0.0035，效果比之前的更好，且没有出现过拟合现象。

cifar10_cnn_dl 网络结构：

```
model = Sequential()

model.add(Convolution2D(32, 3, 3, border_mode='same',
                        input_shape=(img_channels, img_rows, img_cols)))
model.add(Activation('relu'))
model.add(Convolution2D(32, 3, 3))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))

model.add(Convolution2D(64, 3, 3, border_mode='same'))
model.add(Activation('relu'))
model.add(Convolution2D(64, 3, 3))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))

model.add(Flatten())
model.add(Dense(512))
model.add(Activation('relu'))
model.add(Dropout(0.5))
model.add(Dense(nb_classes))
model.add(Activation('softmax'))

# let's train the model using SGD + momentum (how original).
sgd = SGD(lr=0.01, decay=1e-6, momentum=0.9, nesterov=True)
model.compile(loss='categorical_crossentropy', optimizer=sgd)
```

cifar10_cnn_dl 实验结果：

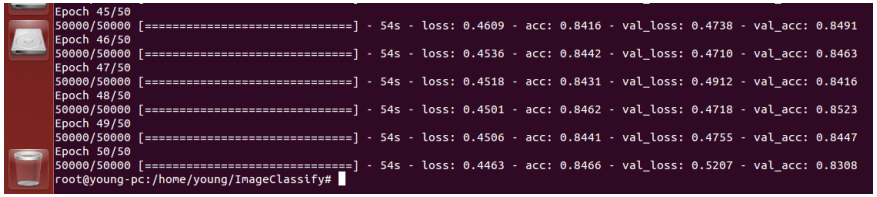


```
Epoch 45/50
50000/50000 [==============================] - 31s - loss: 0.7801 - acc: 0.7336 - val_loss: 0.6425 - val_acc: 0.7826
Epoch 46/50
50000/50000 [==============================] - 31s - loss: 0.7695 - acc: 0.7372 - val_loss: 0.6288 - val_acc: 0.7900
Epoch 47/50
50000/50000 [==============================] - 31s - loss: 0.7766 - acc: 0.7358 - val_loss: 0.6569 - val_acc: 0.7807
Epoch 48/50
50000/50000 [==============================] - 31s - loss: 0.7706 - acc: 0.7383 - val_loss: 0.6171 - val_acc: 0.7889
Epoch 49/50
50000/50000 [==============================] - 31s - loss: 0.7664 - acc: 0.7405 - val_loss: 0.6464 - val_acc: 0.7818
Epoch 50/50
50000/50000 [==============================] - 31s - loss: 0.7723 - acc: 0.7372 - val_loss: 0.6074 - val_acc: 0.7926
root@young-pc:/home/young/ImageClassify#
```

cifar10_cnn_dl8 网络结构：

```
model = Sequential()

model.add(Convolution2D(32,5, 5, border_mode='same',
                        input_shape=(img_channels, img_rows, img_cols)))
model.add(Activation('relu'))
model.add(Convolution2D(32, 5, 5))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))

model.add(Convolution2D(64, 5, 5, border_mode='same'))
model.add(Activation('relu'))
model.add(Convolution2D(64, 5, 5))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))

model.add(Convolution2D(64, 5, 5, border_mode='same'))
model.add(Activation('relu'))
model.add(Convolution2D(64, 5, 5, border_mode='same'))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))

model.add(Flatten())
model.add(Dense(512))
model.add(Activation('tanh'))
model.add(Dense(nb_classes))
model.add(Activation('softmax'))

# let's train the model using SGD + momentum (how original).
sgd = SGD(lr=0.0035, decay=1e-6, momentum=0.9, nesterov=True)
model.compile(loss='categorical_crossentropy', optimizer=sgd)
```

cifar10_cnn_dl8 实验结果：



```
Epoch 45/50
50000/50000 [==============================] - 54s - loss: 0.4609 - acc: 0.8416 - val_loss: 0.4738 - val_acc: 0.8491
Epoch 46/50
50000/50000 [==============================] - 54s - loss: 0.4536 - acc: 0.8442 - val_loss: 0.4710 - val_acc: 0.8463
Epoch 47/50
50000/50000 [==============================] - 54s - loss: 0.4518 - acc: 0.8431 - val_loss: 0.4912 - val_acc: 0.8416
Epoch 48/50
50000/50000 [==============================] - 54s - loss: 0.4501 - acc: 0.8462 - val_loss: 0.4718 - val_acc: 0.8523
Epoch 49/50
50000/50000 [==============================] - 54s - loss: 0.4506 - acc: 0.8441 - val_loss: 0.4755 - val_acc: 0.8447
Epoch 50/50
50000/50000 [==============================] - 54s - loss: 0.4463 - acc: 0.8466 - val_loss: 0.5207 - val_acc: 0.8308
root@young-pc:/home/young/ImageClassify#
```

修改卷积核大小为 5*5，并降低 SGD 的 lr 值为 0.0035，且增加了第二部分卷基层，效果相对来说是最好的了，而且迭代 50 次后还没出现过拟合，仍具有

上升趋势。

cifar10_cnn_dl 网络结构：

```
model = Sequential()

model.add(Convolution2D(32, 3, 3, border_mode='same',
                        input_shape=(img_channels, img_rows, img_cols)))
model.add(Activation('relu'))
model.add(Convolution2D(32, 3, 3))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))

model.add(Convolution2D(64, 3, 3, border_mode='same'))
model.add(Activation('relu'))
model.add(Convolution2D(64, 3, 3))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))

model.add(Flatten())
model.add(Dense(512))
model.add(Activation('relu'))
model.add(Dropout(0.5))
model.add(Dense(nb_classes))
model.add(Activation('softmax'))

# let's train the model using SGD + momentum (how original).
sgd = SGD(lr=0.01, decay=1e-6, momentum=0.9, nesterov=True)
model.compile(loss='categorical_crossentropy', optimizer=sgd)
```

cifar10_cnn_dl9 网络结构：

```
model = Sequential()

model.add(Convolution2D(32,5, 5, border_mode='same',
                        input_shape=(img_channels, img_rows, img_cols)))
model.add(Activation('relu'))
model.add(Convolution2D(32, 5, 5))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))

model.add(Convolution2D(64, 5, 5, border_mode='same'))
model.add(Activation('relu'))
model.add(Convolution2D(64, 5, 5))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))

model.add(Convolution2D(64, 5, 5, border_mode='same'))
model.add(Activation('relu'))
model.add(Convolution2D(64, 5, 5, border_mode='same'))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))

model.add(Flatten())
model.add(Dense(512))
model.add(Activation('tanh'))
model.add(Dense(nb_classes))
model.add(Activation('softmax'))

# let's train the model using SGD + momentum (how original).
sgd = SGD(lr=0.0035, decay=1e-6, momentum=0.9, nesterov=True)
model.compile(loss='categorical_crossentropy', optimizer=sgd)
```

cifar10_cnn_dl 实验结果：

```
Epoch 45/50
50000/50000 [==============================] - 31s - loss: 0.7801 - acc: 0.7336 - val_loss: 0.6425 - val_acc: 0.7826
Epoch 46/50
50000/50000 [==============================] - 31s - loss: 0.7695 - acc: 0.7372 - val_loss: 0.6288 - val_acc: 0.7900
Epoch 47/50
50000/50000 [==============================] - 31s - loss: 0.7766 - acc: 0.7358 - val_loss: 0.6569 - val_acc: 0.7807
Epoch 48/50
50000/50000 [==============================] - 31s - loss: 0.7706 - acc: 0.7383 - val_loss: 0.6171 - val_acc: 0.7889
Epoch 49/50
50000/50000 [==============================] - 31s - loss: 0.7664 - acc: 0.7405 - val_loss: 0.6464 - val_acc: 0.7818
Epoch 50/50
50000/50000 [==============================] - 31s - loss: 0.7723 - acc: 0.7372 - val_loss: 0.6074 - val_acc: 0.7926
root@young-pc:/home/young/ImageClassify#
```

cifar10_cnn_dl9 实验结果：

```
Epoch 95/100
50000/50000 [==============================] - 54s - loss: 0.3681 - acc: 0.8751 - val_loss: 0.4609 - val_acc: 0.8599
Epoch 96/100
50000/50000 [==============================] - 54s - loss: 0.3681 - acc: 0.8744 - val_loss: 0.4603 - val_acc: 0.8589
Epoch 97/100
50000/50000 [==============================] - 54s - loss: 0.3678 - acc: 0.8757 - val_loss: 0.4828 - val_acc: 0.8539
Epoch 98/100
50000/50000 [==============================] - 54s - loss: 0.3683 - acc: 0.8729 - val_loss: 0.4669 - val_acc: 0.8591
Epoch 99/100
50000/50000 [==============================] - 54s - loss: 0.3689 - acc: 0.8730 - val_loss: 0.5160 - val_acc: 0.8482
Epoch 100/100
50000/50000 [==============================] - 54s - loss: 0.3652 - acc: 0.8751 - val_loss: 0.4865 - val_acc: 0.8552
young@young-pc:~/ImageClassify$
```

小结：实验发现，在 cifar10_cnn_dl8 的基础上，将 nb_epoch 改为 100，得到了实验最好结果，训练成功率是 87.51%，测试成功率是 85.52%。

测试展示：

测试代码	网络结构	测试数据	loss	acc	测试时间/s
cifar10_cnn_dl	参考结构	Cifar10 TestSet: 10000	0.6074	0.7926	1
cifar10_cnn_dl8	卷积核：3*3->5*5 最后一个激活函数：’relu’->’ tanh’ SGD: lr = 0.01 -> lr = 0.0035 增加卷积层：第二部分	Cifar10 TestSet: 10000	0.5207	0.8308	2
cifar10_cnn_dl9	卷积核：3*3->5*5 最后一个激活函数：’relu’->’ tanh’ SGD: lr = 0.01 -> lr = 0.0035 增加卷积层：第二部分 增加 nb_epoch：50->100	Cifar10 TestSet: 10000	0.4865	0.8552	2

cifar10_cnn_dl 网络结构：

```
model = Sequential()

model.add(Convolution2D(32, 3, 3, border_mode='same',
                        input_shape=(img_channels, img_rows, img_cols)))
model.add(Activation('relu'))
model.add(Convolution2D(32, 3, 3))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))

model.add(Convolution2D(64, 3, 3, border_mode='same'))
model.add(Activation('relu'))
model.add(Convolution2D(64, 3, 3))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))

model.add(Flatten())
model.add(Dense(512))
model.add(Activation('relu'))
model.add(Dropout(0.5))
model.add(Dense(nb_classes))
model.add(Activation('softmax'))

# let's train the model using SGD + momentum (how original).
sgd = SGD(lr=0.01, decay=1e-6, momentum=0.9, nesterov=True)
model.compile(loss='categorical_crossentropy', optimizer=sgd)
```

cifar10_cnn_dl 测试结果：

cifar10_cnn_dl8 网络结构：

```
model = Sequential()

model.add(Convolution2D(32,5, 5, border_mode='same',
                        input_shape=(img_channels, img_rows, img_cols)))
model.add(Activation('relu'))
model.add(Convolution2D(32, 5, 5))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))

model.add(Convolution2D(64, 5, 5, border_mode='same'))
model.add(Activation('relu'))
model.add(Convolution2D(64, 5, 5))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))

model.add(Convolution2D(64, 5, 5, border_mode='same'))
model.add(Activation('relu'))
model.add(Convolution2D(64, 5, 5, border_mode='same'))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))

model.add(Flatten())
model.add(Dense(512))
model.add(Activation('tanh'))
model.add(Dense(nb_classes))
model.add(Activation('softmax'))

# let's train the model using SGD + momentum (how original).
sgd = SGD(lr=0.0035, decay=1e-6, momentum=0.9, nesterov=True)
model.compile(loss='categorical_crossentropy', optimizer=sgd)
```

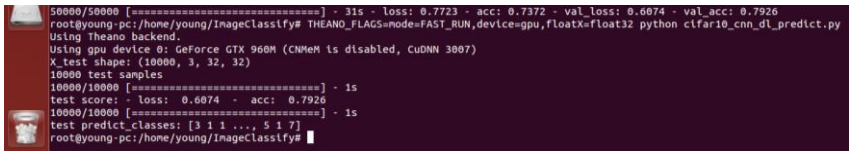
cifar10_cnn_dl8 测试结果：

小结：实验表明，改进过的结构可以将测试成功率提高 4 个百分点，效果还是不错的。

cifar10_cnn_dl 网络结构：

```
model = Sequential()

model.add(Convolution2D(32, 3, 3, border_mode='same',
                        input_shape=(img_channels, img_rows, img_cols)))
model.add(Activation('relu'))
model.add(Convolution2D(32, 3, 3))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))

model.add(Convolution2D(64, 3, 3, border_mode='same'))
model.add(Activation('relu'))
model.add(Convolution2D(64, 3, 3))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))

model.add(Flatten())
model.add(Dense(512))
model.add(Activation('relu'))
model.add(Dropout(0.5))
model.add(Dense(nb_classes))
model.add(Activation('softmax'))

# let's train the model using SGD + momentum (how original).
sgd = SGD(lr=0.01, decay=1e-6, momentum=0.9, nesterov=True)
model.compile(loss='categorical_crossentropy', optimizer=sgd)
```

cifar10_cnn_dl 测试结果：



cifar10_cnn_dl9 网络结构：

```
model = Sequential()

model.add(Convolution2D(32,5, 5, border_mode='same',
                        input_shape=(img_channels, img_rows, img_cols)))
model.add(Activation('relu'))
model.add(Convolution2D(32, 5, 5))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))

model.add(Convolution2D(64, 5, 5, border_mode='same'))
model.add(Activation('relu'))
model.add(Convolution2D(64, 5, 5))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))

model.add(Convolution2D(64, 5, 5, border_mode='same'))
model.add(Activation('relu'))
model.add(Convolution2D(64, 5, 5, border_mode='same'))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))

model.add(Flatten())
model.add(Dense(512))
model.add(Activation('tanh'))
model.add(Dense(nb_classes))
model.add(Activation('softmax'))

# let's train the model using SGD + momentum (how original).
sgd = SGD(lr=0.0035, decay=1e-6, momentum=0.9, nesterov=True)
model.compile(loss='categorical_crossentropy', optimizer=sgd)
```

cifar10_cnn_dl9 测试结果：

小结：实验表明，改进过的结构可以将测试成功率提高 6 个百分点，效果很明显。

附录：

1. 文档中所有源文件都在文件夹 ImageClassifySummary 里，实验结果截图在 resultimage 文件夹下。
2. 保存的模型和测试结果均以其对应源代码文件名区别。例如，源代码为 cifar10_cnn_dl.py，训练保存的模型文件为 my_model_architecture_dl.json，模型权重为 my_model_weights_dl.h5，测试代码为 cifar10_cnn_dl_predict.py，测试预测结果为 test_predict_dl.txt，测试分类结果为 test_predict_classes_dl.txt。其他源文件的测试结果文件名与之类似。
3. 测试平台参数：

   CPU：Inter(R) Core(TM) i7-6700HQ CPU @ 2.60

   GPU：GeForce GTX 960M

   Cuda：7.5

   CuDNN：7.0

   Python：2.7.6

   Theano：0.8.1

   Keras：0.3.3