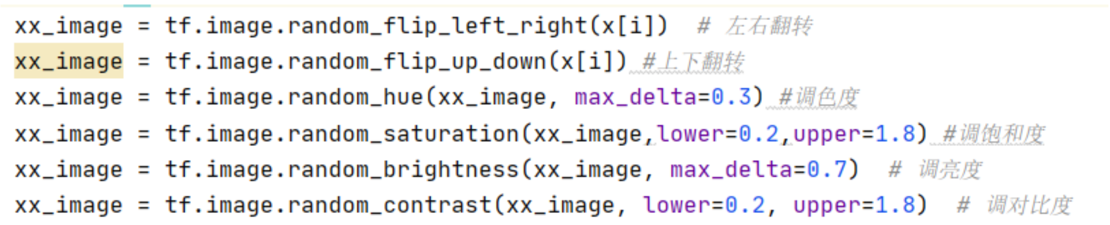
注：因为选择的数据集为 cifar10 与 cifar100，步骤很相似，下面仅以 cifar10 为例作报告

1. 测试数据生成方法

使用数据增量方法，对图像进行随机变换，调用 tensorflow.image 内置的图像变换函数进行测试数据的生成



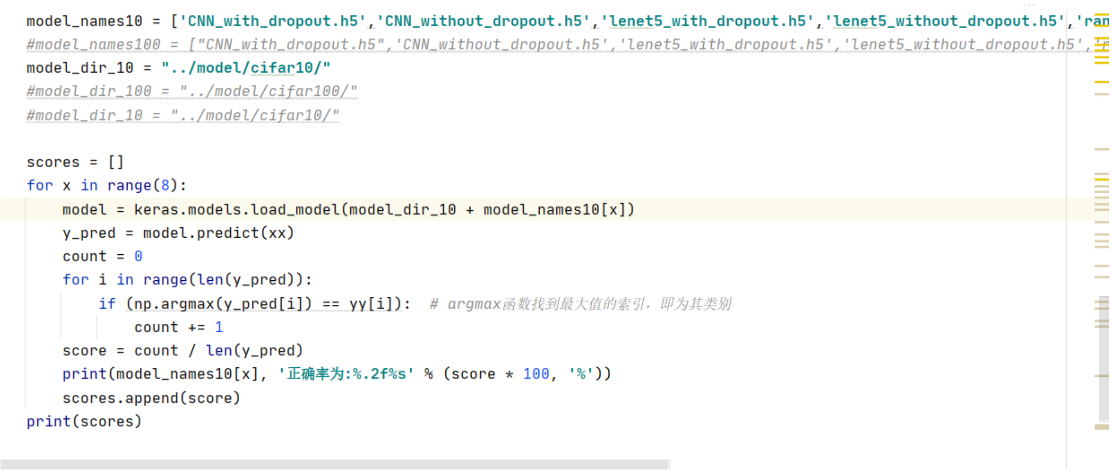
```
xx_image = tf.image.random_flip_left_right(x[i])  # 左右翻转
xx_image = tf.image.random_flip_up_down(x[i]) #上下翻转
xx_image = tf.image.random_hue(xx_image, max_delta=0.3) #调色度
xx_image = tf.image.random_saturation(xx_image,lower=0.2,upper=1.8) #调饱和度
xx_image = tf.image.random_brightness(xx_image, max_delta=0.7)  # 调亮度
xx_image = tf.image.random_contrast(xx_image, lower=0.2, upper=1.8)  # 调对比度
```

调色度方法中，max_delta 表示图像色像的取值在[0,0.3]之间，亮度同理

调饱和度方法中，lower 和 upper 表示饱和度的上下界，对比度同理

2. 测试数据质量度量方法

使用深度学习模型



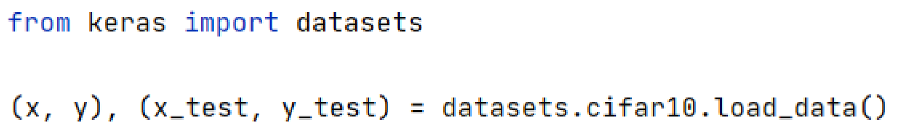
```
model_names10 = ['CNN_with_dropout.h5','CNN_without_dropout.h5','lenet5_with_dropout.h5','lenet5_without_dropout.h5','ran
#model_names100 = ["CNN_with_dropout.h5",'CNN_without_dropout.h5','lenet5_with_dropout.h5','lenet5_without_dropout.h5','r
model_dir_10 = "../model/cifar10/"
#model_dir_100 = "../model/cifar100/"
#model_dir_10 = "../model/cifar10/"

scores = []
for x in range(8):
    model = keras.models.load_model(model_dir_10 + model_names10[x])
    y_pred = model.predict(xx)
    count = 0
    for i in range(len(y_pred)):
        if (np.argmax(y_pred[i]) == yy[i]):  # argmax函数找到最大值的索引，即为其类别
            count += 1
    score = count / len(y_pred)
    print(model_names10[x], '正确率为:%.2f%s' % (score * 100, '%'))
    scores.append(score)
print(scores)
```

用 argmax 函数找到最大值的索引，就是它的类别，以此为依据判断正确率

3. 项目代码实现的流程图介绍

- 加载数据

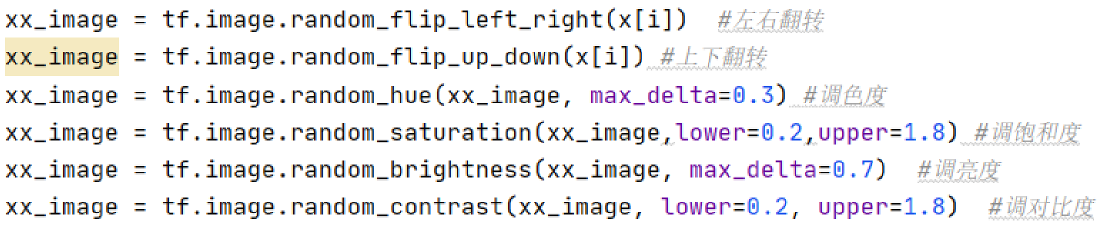
直接使用 keras.datasets 加载数据集



```
from keras import datasets

(x, y), (x_test, y_test) = datasets.cifar10.load_data()
```

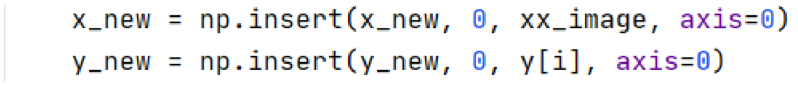
- 数据扩增

调用 tensorflow 方法进行数据扩增

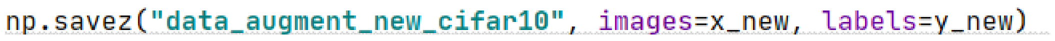


```
xx_image = tf.image.random_flip_left_right(x[i])  #左右翻转
xx_image = tf.image.random_flip_up_down(x[i]) #上下翻转
xx_image = tf.image.random_hue(xx_image, max_delta=0.3) #调色度
xx_image = tf.image.random_saturation(xx_image,lower=0.2,upper=1.8) #调饱和度
xx_image = tf.image.random_brightness(xx_image, max_delta=0.7)  #调亮度
xx_image = tf.image.random_contrast(xx_image, lower=0.2, upper=1.8)  #调对比度
```

- 数据保存



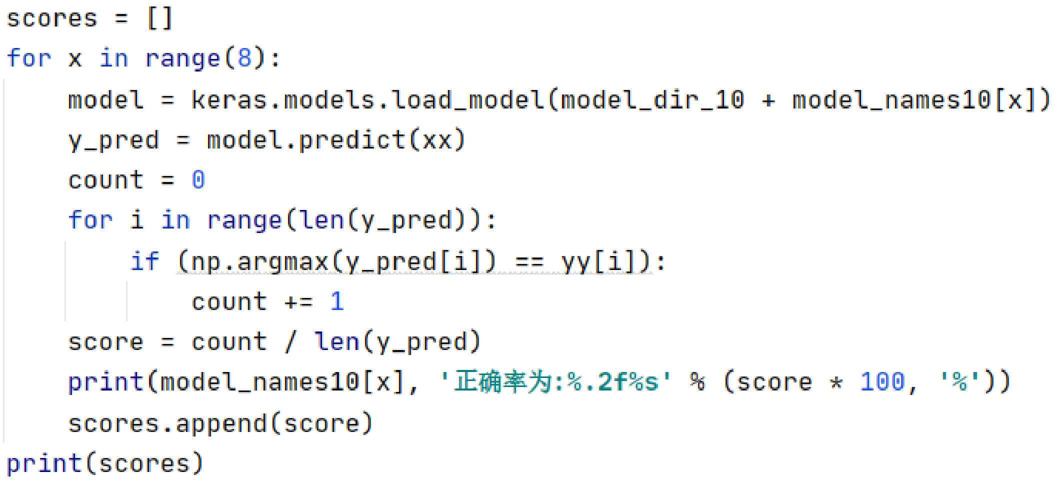
```
    x_new = np.insert(x_new, 0, xx_image, axis=0)
    y_new = np.insert(y_new, 0, y[i], axis=0)
```



```
np.savez("data_augment_new_cifar10", images=x_new, labels=y_new)
```

向原始训练集的开头插入随机变换后的图像，再向原始训练集的标签插入随机变换后的图像的标签，之后将多个数组保存到文件中

- 数据质量测试



```
scores = []
for x in range(8):
    model = keras.models.load_model(model_dir_10 + model_names10[x])
    y_pred = model.predict(xx)
    count = 0
    for i in range(len(y_pred)):
        if (np.argmax(y_pred[i]) == yy[i]):
            count += 1
    score = count / len(y_pred)
    print(model_names10[x], '正确率为:%.2f%s' % (score * 100, '%'))
    scores.append(score)
print(scores)
```

使用深度学习模型测试正确率