

实验一 图像任务

姓名：张辰菁 班级：计算机2103 学号：2215015048

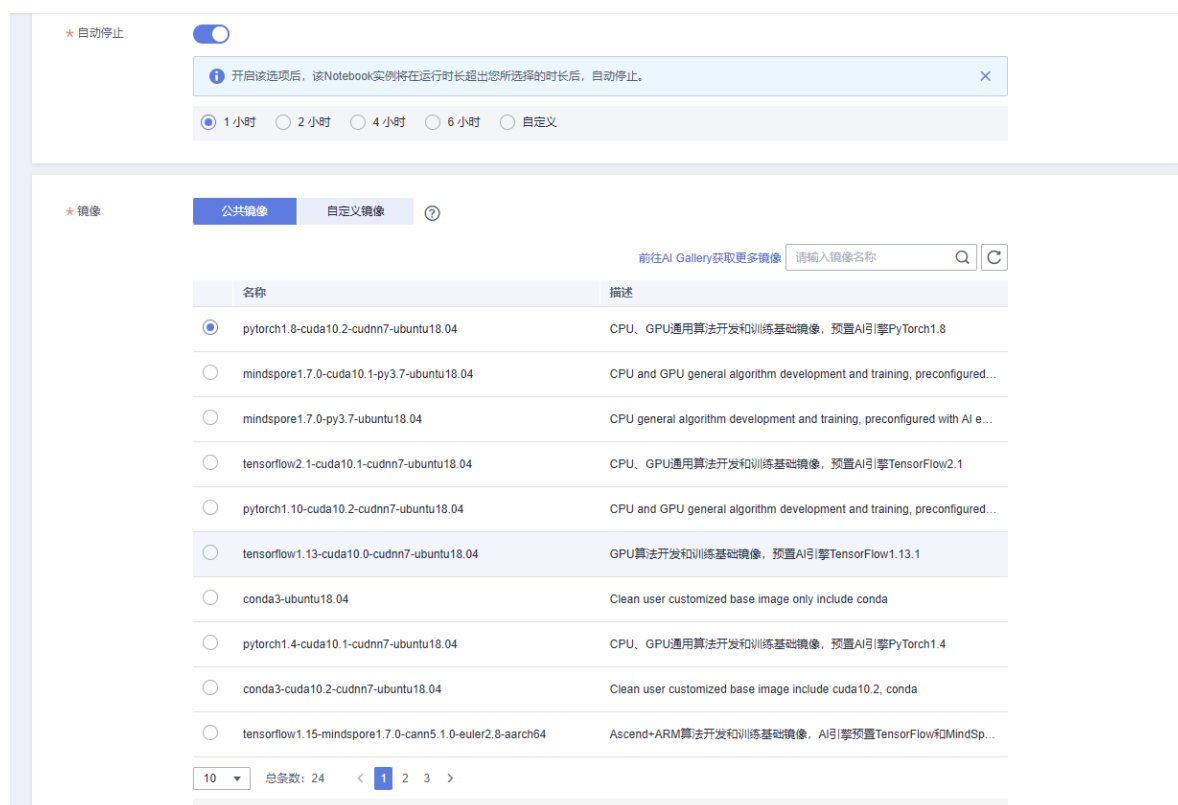
一、实验目的

- 1.了解华为云上Mindspore资源申请方式；
- 2.掌握Mindspore中数据集选择、下载、搭建的方法；
- 3.以图像为例子，学习Mindspore中数据处理与增强的方法。

二、实验过程

1.华为云资源申请

- 1.登录www.huaweicloud.com，点击右上角的控制台，在上方搜索框内搜索“Modelarts”进入Modelarts控制台
- 2.点击左侧总览下的开发环境-Notebook进入Notebook控制台，点击右上角“创建”创建Notebook



在镜像中选择mindspore1.7.0-cuda10.1-py3.7-ubuntu18.04（或者其他mindspore），选择适宜的类型和CPU/GPU规格后，点击“立即创建”创建环境，等待几分钟后在Notebook控制台中打开。

2.数据集选择与下载

可以选择Mnist, Cifar10, ImageNet, FashionMNIST等数据集。

使用本地下载再上传或者直接wget在Notebook内下载。

3.数据集搭建

使用mindspore.dataset库中的函数来及进行数据集搭建。对于不同的数据集，有对应不同的函数，如Cifar10数据集使用的是

```
mindspore.dataset.Cifar10Dataset(DATA_DIR_CIFAR10, num_samples( (即批大小) =>
```

函数。

使用dataset.create_dict_iterator()来对已经构建好的数据集中的数据进行遍历并存储到一个list中

```
image_list, label_list = [], []
for data in dataset1.create_dict_iterator():
    image_list.append(data['image'])
    label_list.append(data['label'])
```

在随后的打印、处理图片时，使用list中的数据。

4.数据处理与增强

使用matplotlib.pyplot (as plt) 库中的函数来打印图片。

先使用

```
plt.figure(figsize=())
```

来规定每一张图片打印出来的大小，避免打印成批图片时图片挤在一起或是太大而无法正常显示；

对于成批图片，可以使用for循环，plt.subplot(row,column,pos)来分别打印每张图片到对应位置。

```
count=0
row = 8      # 画布行数
column = 8   # 画布列数
pos = 1
plt.figure(figsize=(14,14))
for i in range(row):
    for j in range(column):
        plt.subplot(row, column, pos)                # 显示位置
        plt.imshow(image_list[count].asnumpy(), cmap=plt.cm.gray) # 显示内容
        plt.title(labels[int(label_list[count].asnumpy())])
        # 显示标题
        plt.axis('off')
        pos = pos + 1
        count= count + 1
    pos = column * (i + 1) + 1
```

对于单张图片，设定好figsize后，直接使用plt.imshow()即可

数据增强操作则使用下列库中的函数：

```
import mindspore.dataset.vision.c_transforms as c_trans
import mindspore.dataset.transforms as c2
```

实现单个处理方式处理整个数据集，使用C和C2中的函数，将一个变量赋值为一个处理函数，然后使用dataset_name.map()函数。

```
random_crop = c_trans.RandomCrop([10, 10])
dataset2 = dataset1.map(operations=random_crop, input_columns=["image"])
```

实现多个处理方式处理整个数据集，则给变量transforms赋值为多个处理函数，然后使用Compose函数将处理函数整合，最后使用dataset.map()将函数作用于整个数据集：

```
transforms = [
    #翻转
    c_trans.RandomHorizontalFlip(prob=1.0),
    #裁剪
    c_trans.RandomCrop((32,32),(2,2,2,2)),
    #反相
    #c_trans.Invert(),
    #旋转180°
    C.RandomRotation(degrees=(180,180), expand=False, center=(15.5,15.5))
]
# 通过Compose操作将transforms列表中函数作用于数据集图片

compose_trans = Compose(transforms)
dataset2 = dataset1.map(operations=compose_trans, input_columns=["image"])
```

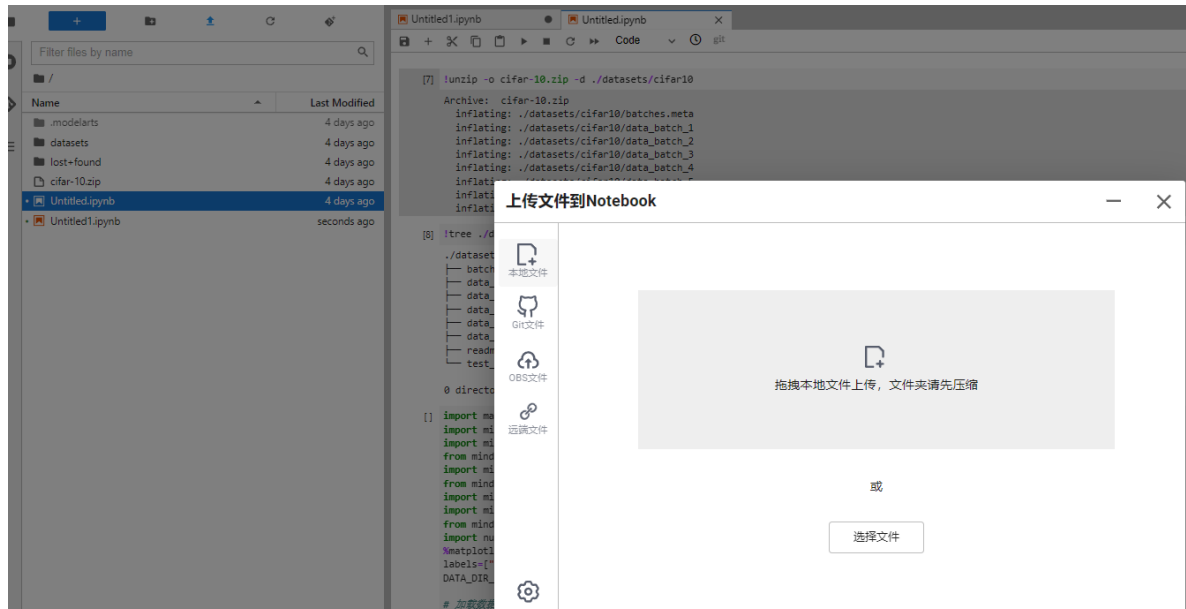
三、实验内容

1.数据集的使用和读取

本次实验以Cifar10数据集为例，学习在Mindspore上使用和读取数据集的方法。

1.1 本地下载数据集，上传至Notebook

可以先在本机上下载好数据集的压缩包，然后点击箭头拖拽本地文件进行上传。



上传后，新建.ipynb文件，使用

```
!unzip -o cifar-10.zip -d ./datasets/cifar10
```

进行解压。

1.2 直接在notebook里下载数据集

可以使用wget指令直接下载

```
!wget -N https://obs.dualstack.cn-north-4.myhuaweicloud.com/mindspore-website/notebook/datasets/cifar10.zip
```

解压方式同上。

1.3 创建一个数据集，读取一批batch大小为64的数据，并将它们打印到一个图片里

```
import matplotlib.pyplot as plt
import mindspore.dataset as ds
import mindspore.dataset.vision as vision
from mindspore.dataset.vision import Inter
import mindspore.nn as nn
from mindspore import dtype as mstype
import mindspore.dataset.vision.c_transforms as C
import mindspore.dataset.transforms as C2
from mindspore import context
import numpy as np
%matplotlib inline
labels=
["airplane","automobile","bird","cat","deer","dog","frog","horse","ship","truck"
]
DATA_DIR_CIFAR10 = "./datasets/cifar10/"
```

```
# 从Cifar10中选取64张图片构建数据集dataset1
dataset1 = ds.Cifar10Dataset(DATA_DIR, num_samples=64, shuffle=True)
#将dataset1中数据存放到列表中
image_list, label_list = [], []
for data in dataset1.create_dict_iterator():
    image_list.append(data['image'])
    label_list.append(data['label'])

num_samples = len(image_list)
count=0
row = 8      # 画布行数
column = 8   # 画布列数
pos = 1
#for循环打印这64张图片到一张图片
plt.figure(figsize=(14,14))
for i in range(row):
    for j in range(column):
        plt.subplot(row, column, pos)                # 显示位置
        plt.imshow(image_list[count].asnumpy(), cmap=plt.cm.gray) # 显示内容
        plt.title(labels[int(label_list[count].asnumpy())])
        plt.axis('off')
        pos = pos + 1
        count= count + 1
    pos = column * (i + 1) + 1
```

2.数据的切片

在这批数据里选择一个图片，单独打印这张图片。

构建好数据集后，使用

```
image_list, label_list = [], []
for data in dataset1.create_dict_iterator():
    image_list.append(data['image'])
    label_list.append(data['label'])
```

将数据存至列表，然后直接通过下标方式进行切片。

```
image=image_list[0].asnumpy()
label=labels[int(label_list[0].asnumpy())]
plt.imshow(image) # 显示内容
plt.title(label)
```

3.数据处理方法的使用

先定义transforms变量，将处理函数赋值给transforms：

```
transforms = [
    #翻转
    c_trans.RandomHorizontalFlip(prob=1.0),
    #裁剪
    c_trans.RandomCrop((32,32),(2,2,2,2)),
    #反相
    #c_trans.Invert(),
    #旋转180°
    C.RandomRotation(degrees=(180,180), expand=False, center=(15.5,15.5))
]
```

然后通过Compose操作将transforms列表中函数作用于数据集图片

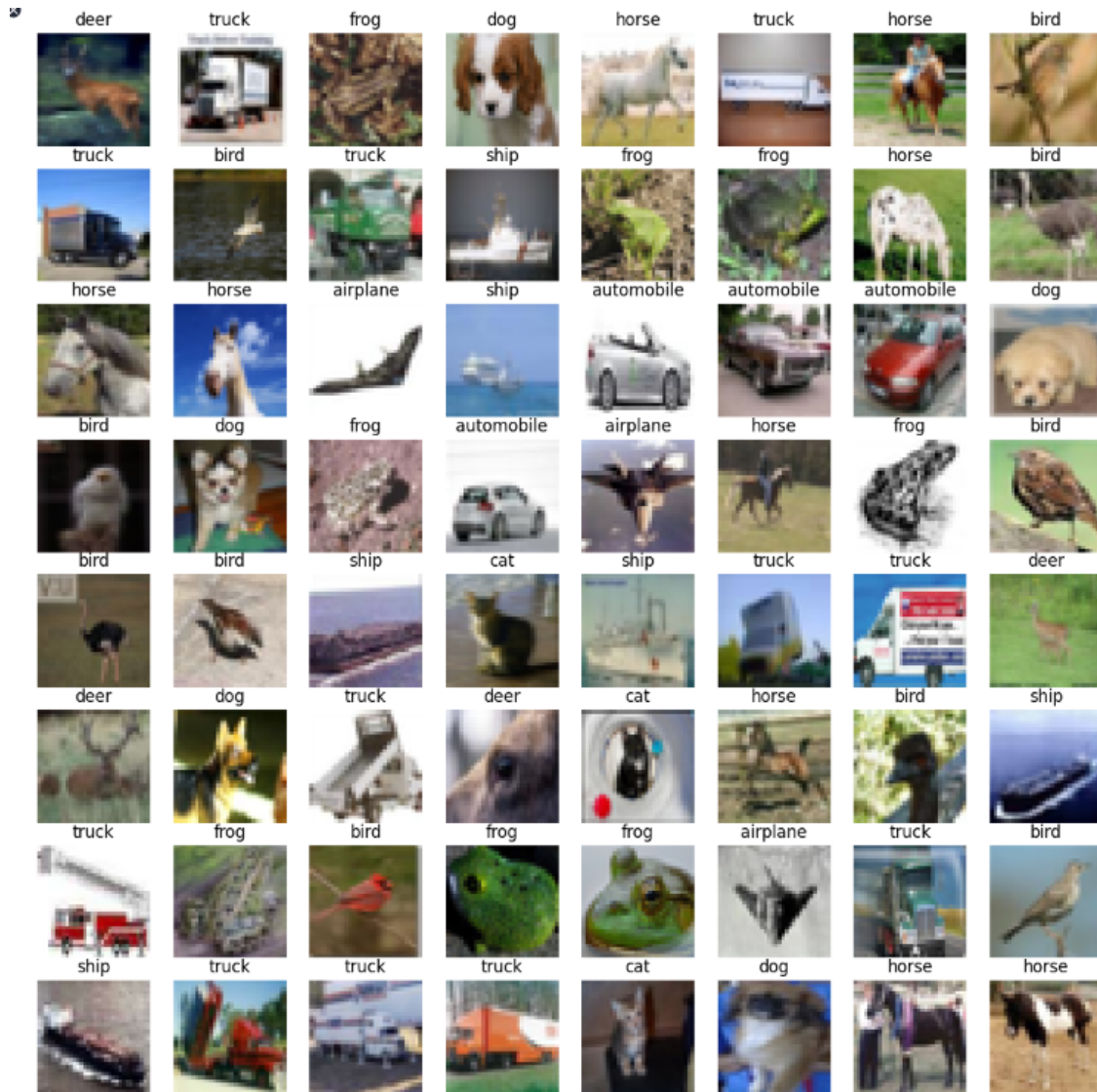
```
# 通过Compose操作将transforms列表中函数作用于数据集图片
compose_trans = Compose(transforms)
dataset2 = dataset1.map(operations=compose_trans, input_columns=["image"])
```

最后从dataset2中选取图片查看处理结果：

```
image_list3=[]
label_list3=[]
for data in dataset2.create_dict_iterator():
    image_list3.append(data['image'])
    label_list3.append(data['label'])
image_list2.append(image_list3[0])
label_list2.append(label_list3[0])
plt.figure(figsize=(300,300))
num_samples = len(image_list2)
for i in range(num_samples):
    plt.subplot(1, len(image_list), i + 1)
    plt.imshow(image_list2[i].asnumpy())
    plt.title(labels[int(label_list2[i].asnumpy())])
```

四、实验结果

1.创建一个数据集，读取一批batch大小为64的数据，并将它们打印到一个图片里

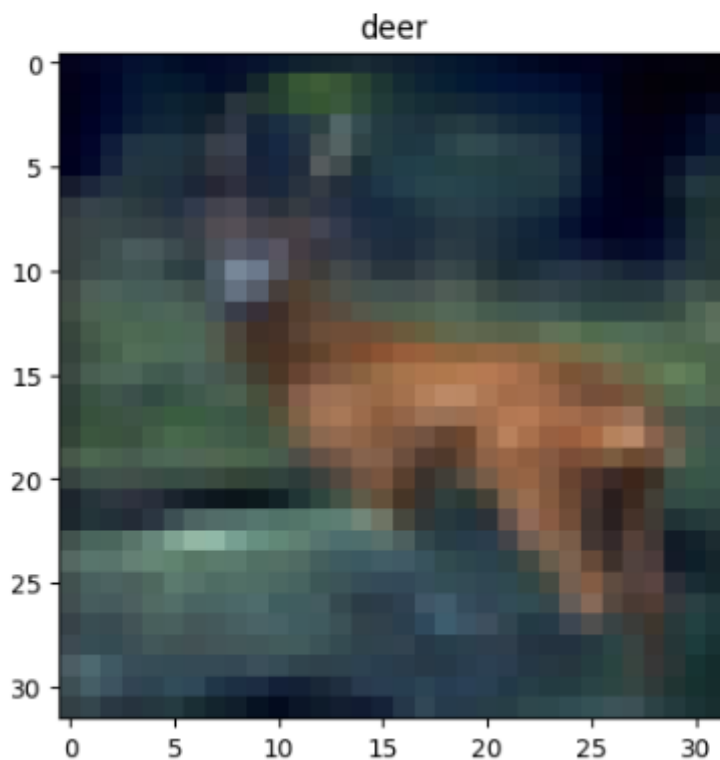


2.在这批数据里选择一个图片，单独打印这张图片。

83
ns

```
image=image_list[0].asnumpy()  
label=labels[int(label_list[0].asnumpy())]  
plt.imshow(image) # 显示内容  
plt.title(label)
```

```
Text(0.5, 1.0, 'deer')
```

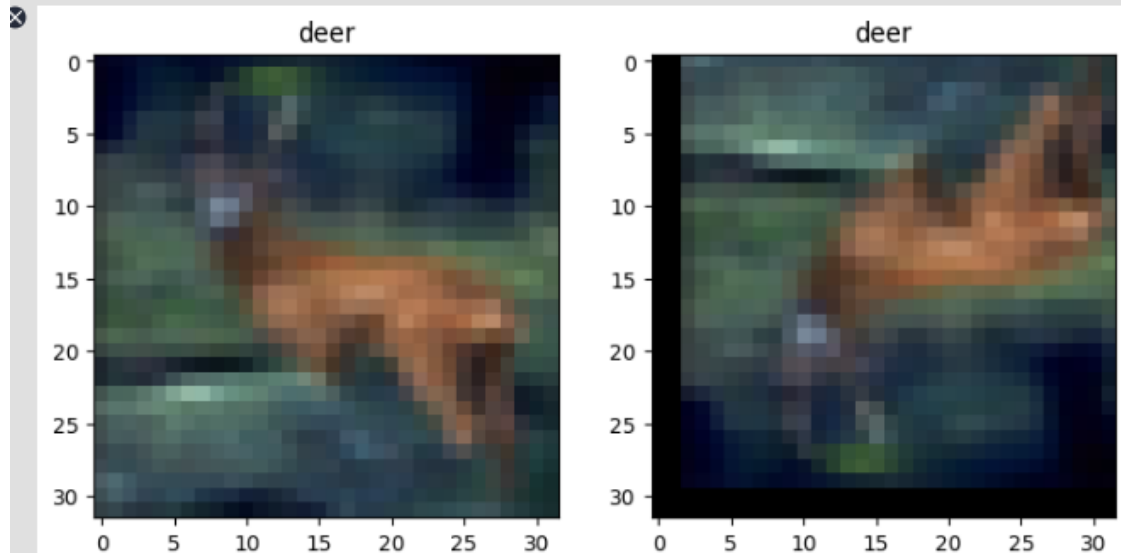


3.选择旋转、翻转、裁剪的其中两种方法分别进行处理，并将原图和处理后的图像打印到一个图片里。

```

image_list3=[]
label_list3=[]
for data in dataset2.create_dict_iterator():
    image_list3.append(data['image'])
    label_list3.append(data['label'])
image_list2.append(image_list3[0])
label_list2.append(label_list3[0])
plt.figure(figsize=(300,300))
num_samples = len(image_list2)
for i in range(num_samples):
    plt.subplot(1, len(image_list), i + 1)
    plt.imshow(image_list2[i].asnumpy())
    plt.title(labels[int(label_list2[i].asnumpy())])

```



五、总结

- 1.本次实验通过查阅MindSpore手册，完成了对MindSpore中数据集构建和数据读取、数据切片、数据处理的学习。
- 2.Mindspore中对于数据的读取、处理都有库函数可以使用，在使用时，注意查阅官方手册，多尝试即可。相关链接：[MindSpore](#)

六、完整源代码

```
!unzip -o cifar-10.zip -d ./datasets/cifar10
```

```
!tree ./datasets/cifar10
```

```

import matplotlib.pyplot as plt
import mindspore.dataset as ds
import mindspore.dataset.vision as vision
from mindspore.dataset.vision import Inter
import mindspore.nn as nn
from mindspore import dtype as mstype
from mindspore import context
import numpy as np
import mindspore.dataset.vision.c_transforms as c_trans
import mindspore.dataset.transforms as c2
%matplotlib inline

```



```
labels=
["airplane", "automobile", "bird", "cat", "deer", "dog", "frog", "horse", "ship", "truck"
]
DATA_DIR_CIFAR10 = "./datasets/cifar10/"
```

```
dataset1 = ds.Cifar10Dataset(DATA_DIR, num_samples=64, shuffle=True)

image_list, label_list = [], []
for data in dataset1.create_dict_iterator():
    image_list.append(data['image'])
    label_list.append(data['label'])

num_samples = len(image_list)
count=0
row = 8      # 画布行数
column = 8   # 画布列数
pos = 1
plt.figure(figsize=(14,14))
for i in range(row):
    for j in range(column):
        plt.subplot(row, column, pos)                # 显示位置
        plt.imshow(image_list[count].asnumpy(), cmap=plt.cm.gray) # 显示内容
        plt.title(labels[int(label_list[count].asnumpy())])
        # 显示标题
        plt.axis('off')
        pos = pos + 1
        count= count + 1
pos = column * (i + 1) + 1
```

```
image=image_list[0].asnumpy()
label=labels[int(label_list[0].asnumpy())]
plt.imshow(image) # 显示内容
plt.title(label)
```

```
# 打印数据增强操作后图片的形状、标签
image_list2, label_list2 = [], []
image_list2.append(image_list[0])
label_list2.append(label_list[0])
transforms = [
    #翻转
    c_trans.RandomHorizontalFlip(prob=1.0),
    #裁剪
    c_trans.RandomCrop((32,32), (2,2,2,2)),
    #反相
    #c_trans.Invert(),
    #旋转180°
    C.RandomRotation(degrees=(180,180), expand=False, center=(15.5,15.5))
]
# 通过Compose操作将transforms列表中函数作用于数据集图片
compose_trans = Compose(transforms)
dataset2 = dataset1.map(operations=compose_trans, input_columns=["image"])

image_list3=[]
```

```
label_list3=[]
for data in dataset2.create_dict_iterator():
    image_list3.append(data['image'])
    label_list3.append(data['label'])
image_list2.append(image_list3[0])
label_list2.append(label_list3[0])
plt.figure(figsize=(300,300))
num_samples = len(image_list2)
for i in range(num_samples):
    plt.subplot(1, len(image_list), i + 1)
    plt.imshow(image_list2[i].asnumpy())
    plt.title(labels[int(label_list2[i].asnumpy())])
```