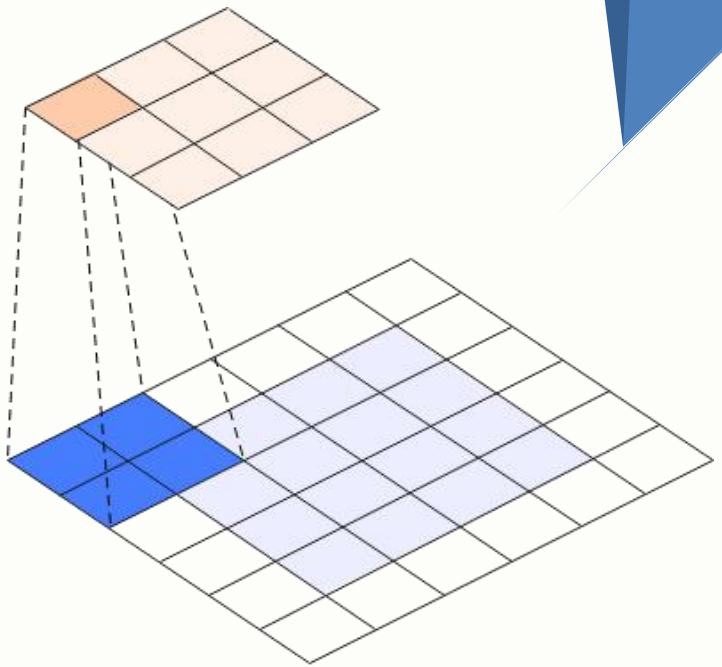
第04章 图像数据集和数据预处理

欧 新 宇

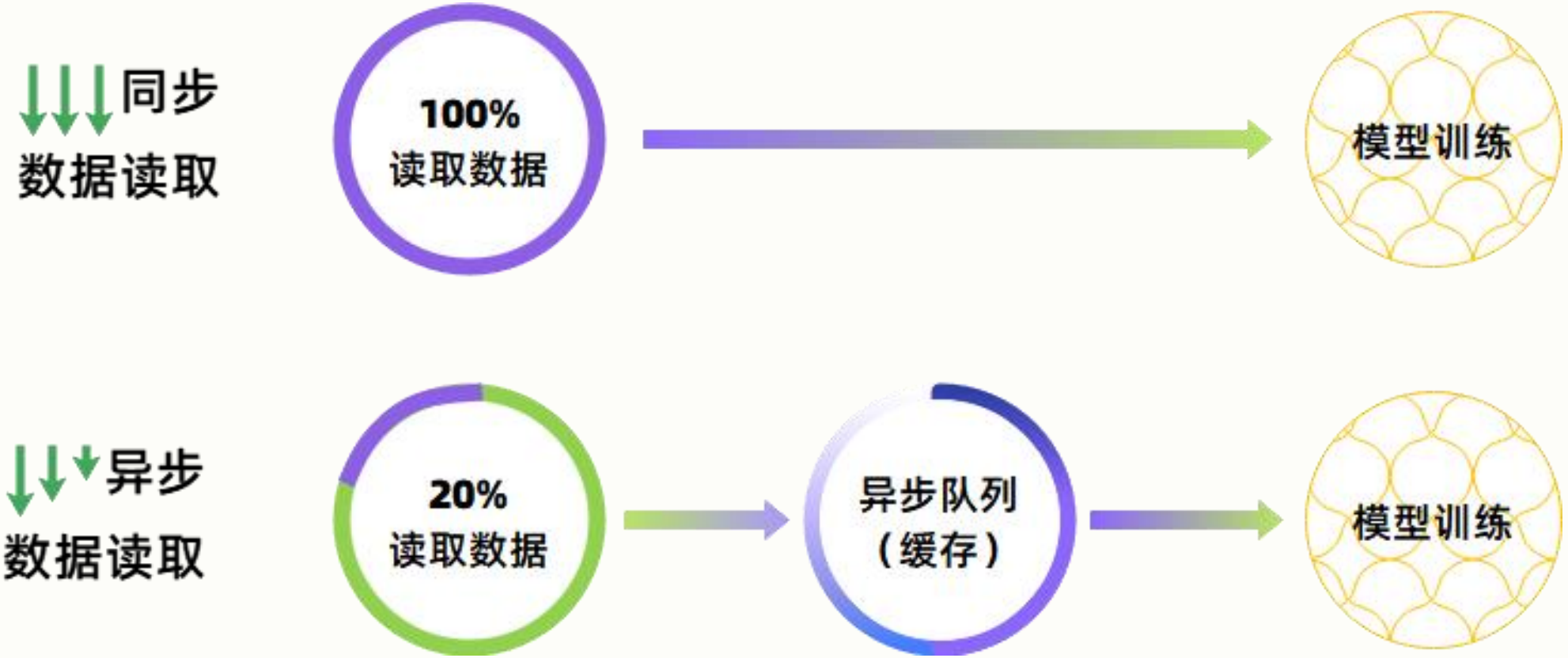
Thanks to LiMu-CS329 and LiFeifei-CS231n



数据读取（Data Reading）

数据读取方法简介

同步数据读取和异步数据读取



数据读取方法简介

数据读取的基本流程

定义必要库及全局参数

os,cv2,json,numpy,paddle；数据集路径；图像基本参数

定义数据集

从数据列表中获取数据，对图像进行数据规约和数据增广

创建数据读取器

利用数据集类实例化一个数据读取器，返回图像矩阵、类别标签和样本数量

创建小批量数据迭代读取器

使用高阶API创建小批量数据迭代读取器。配置批次大小、打乱、末尾丢弃

异步数据读取

```python
# codes04007_asynchronous_initialization
import os
import cv2
import json
import numpy as np
import matplotlib.pyplot as plt
import sys
sys.path.append(r'D:\WorkSpace\DeepLearning\WebsiteV2')    # 定义课程自定义模块保存位置
from codes.paddle import common, datasets
import paddle
import paddle.vision.transforms as T

# 1. 定义数据集基本信息
dataset_name      = 'Zodiac'
dataset_path      = 'D:\\Workspace\\ExpDatasets\\'
dataset_root_path = os.path.join(dataset_path, dataset_name)

# 2. 图像基本信息
args = {
'input_size': [3, 227, 227],              # 定义图像输入模型时的尺寸
'mean_value': [0.485, 0.456, 0.406],      # Imagenet均值
'std_value': [0.229, 0.224, 0.225],       # Imagenet标准差
}
```

1. 定义必要库

本课程自建库，可以用来实现一些特定功能，方便代码重用

① 定义必要库及全局参数

os,cv2,json,numpy,paddle；
数据集路径；图像基本参数

2. 定义数据集根路径

3. 定义超参数，包括数据集基本信息、数据增广和数据规约配置超参数

异步数据读取

定义数据集 ②

从数据列表中获取数据，对图像进行数据规约和数据增广

```python
# codes04008_asynchronous_create_dataset
class DatasetZodiac(paddle.io.Dataset):
    # 1. 初始化数据集，并将样本和标签映射到列表中
    def __init__(self, dataset_root_path, mode='test'):
    # 2. 定义数据获取函数，返回单条数据（样本数据、对应的标签）
    def __getitem__(self, index):
    # 3. 定义样本总数获取函数
    def __len__(self):
```

1. 初始化数据集，定义数据列表读取方式、数据规约和数据增广方法。

2. 读取数据、执行数据增广，并返回图像矩阵和标签数组。

3. 返回样本总数

```python
    # 读取数据列表文件，将每一行都按照路径和标签进行拆分成两个字段的序列，并将序列依次保存至data序列中
    # 1) 若列表信息长度为2，则表示包含路径和标签信息。
    # 2) 若列表信息长度为1，则表示只包含路径，不包含标签。一般正式的测试文件都只包含路径，不包含标签。
    with open(os.path.join(dataset_root_path, mode+'.txt')) as f:
        for line in f.readlines():
            info = line.strip().split('\t')                 # 拆分从列表文件中读取到数据信息
            image_path = info[0].strip()                    # 信息的[0]位置为路径
            if len(info) == 2:                              # 判断信息的长度，若包含标签则写入image_label
                image_label = info[1].strip()
            elif len(info) == 1:                            # 判断信息的长度，若不包含标签，则用"-1"表示
                image_label = -1
            self.data.append([image_path, image_label])             # 将路径和标签写入[data]容器

    # 对训练数据和验证、测试数据采用不同的数据预处理方法
    # 1) train和trainval：执行随机裁剪，并完成标准化预处理
    # 2) train和trainval：直接执行尺度缩放，并完成标准化预处理
    inputSize = self.args['input_size'][1:3] if len(self.args['input_size'])==3 else self.args['input_size']
    if self.isTransforms == 0:
        self.transforms = T.Compose([                       # 0) 必要数据规约
            T.Resize(inputSize),            # 直接尺度缩放
            T.ToTensor(),                                   # 转换成Paddle规定的Tensor格式
        ])
    elif self.isTransforms == 1 or (self.isTransforms == 2 and mode in ['val', 'test']):
        self.transforms = T.Compose([                       # 1) 基本数据预处理，不含数据增广
            T.Resize(inputSize),            # 直接尺度缩放
            T.ToTensor(),                                   # 转换成Paddle规定的Tensor格式
            T.Normalize(mean=self.args['mean_value'],       # 均值方差归一化
                        std=self.args['std_value'])
        ])
    elif self.isTransforms == 2 and mode in ['train', 'trainval']:
        self.transforms = T.Compose([                       # 2) 训练数据预处理，包含数据增广
            T.Resize((256, 256)),                           # 直接尺度缩放
            T.RandomResizedCrop(inputSize),  # 随机裁剪
            T.RandomHorizontalFlip(prob=0.5),               # 水平翻转
            T.RandomRotation(15),                           # 随机旋转
            T.ColorJitter(brightness=0.4,                   # 色彩扰动：亮度、对比度、饱和度和色度
                          contrast=0.4,
```

从数据集列表中获取图像路径和标签

根据任务需求不同定义三种不同的数据规约和数据增广方法

异步数据读取

③ 创建数据读取器和数据迭代读取器

④ 利用数据集类实例化一个数据读取器和小批量数据迭代读取器，返回小批次图像的图像矩阵、类别标签

程序清单4-9　创建异步数据读取器

```
# codes04009_asynchronous_create_reader
dataset_train = DatasetZodiac(dataset_root_path, args=args, mode='train')
dataset_val = DatasetZodiac(dataset_root_path, args=args, mode='val')
dataset_trainval = DatasetZodiac(dataset_root_path, args=args, mode='trainval')
dataset_test = DatasetZodiac(dataset_root_path, args=args, mode='test')
```

1. 实例化数据集类，并从硬盘上读取数据

程序清单4-13　创建异步数据迭代读取器

```
# codes04013_asynchronous_create_dataLoader
train_reader = paddle.io.DataLoader(dataset_train, batch_size=64, shuffle=True,drop_last=True)
val_reader = paddle.io.DataLoader(dataset_val, batch_size=64, shuffle=False, drop_last=False)
trainval_reader = paddle.io.DataLoader(dataset_trainval, batch_size=64, shuffle=True, drop_last=True)
test_reader = paddle.io.DataLoader(dataset_test, batch_size=64, shuffle=False, drop_last=False)
```

仅训练集和训练验证集要求强制打乱

2. 拆分数据读取器，按批次进行输出，并打乱数据

欧老师的联系方式

读万卷书 行万里路 只为最好的修炼

QQ：14777591 (宇宙骑士)

Email：ouxinyu@alumni.hust.edu.cn

Website: http://ouxinyu.cn

Tel：18687840023