- **2-1 数据读取机制DataLoader与Dataset**
  - 1. 人民币二分类
    - 数据
      - 数据收集：Im

g，Label

- 数据划分：train、valid、test
- 数据读取：DataLoader
  - **Samp**

Index：生成索引，样本的序号

- **DataSet**：Img，Label

根据索引读取图

片或标签
    - 数据预处理：transforms
- 2. DataLoader与Dataset
    - Data

Loader

- torch.utils.data.DataLoader
  - **构建可迭代的数据装载器**
    - dat

aset：Dataset类，决定数据从哪读取及如何读取

- batchsize：批大小
- num_wor

ks：是否多进程读取数据。采用多进程可以加快读取数据的速度，减少时间

- shuffle：每个epo

ch是否乱序

- drop_last：当样本数不能被batchsize整除时，是否舍弃最后一批数据

- **Epoch：所有训练样本都已输入到模型中，称为一个Epoch**
- Iteration：一批样本输入到

模型中，称之为一个Iteration

- Batchsize：批大小，决定一个Epoch有多少个It

eration

- y样本总数：80，Batchsize：8；1 Epoch = 10 Itera

tion

- torch.utils.data.Dataset
  - Dataset抽象类，所有自定义的Da

tasset需要继承它，并且复写

- __getitem__()
- getitem：接收一个索引，返回一

个样本

- 数据读取
  - 1. 读哪些数据？
    - Sampler输出的Index
  - 2. 从哪读数据？
    - Dataset

t中的data_dir

- 3.

  怎么读数据?

  - Dataset中的getitem

    - 数据读取机制的流程图

      -

for循环中使用Dataloader，

- 进入Dataloader之后根据是否使用多进程进入我们单

进程或者多进程的DataLoaderIter，

- 之后使用Sampler获取索引Index，
- 拿到索

引之后给到DatasetFetcher，

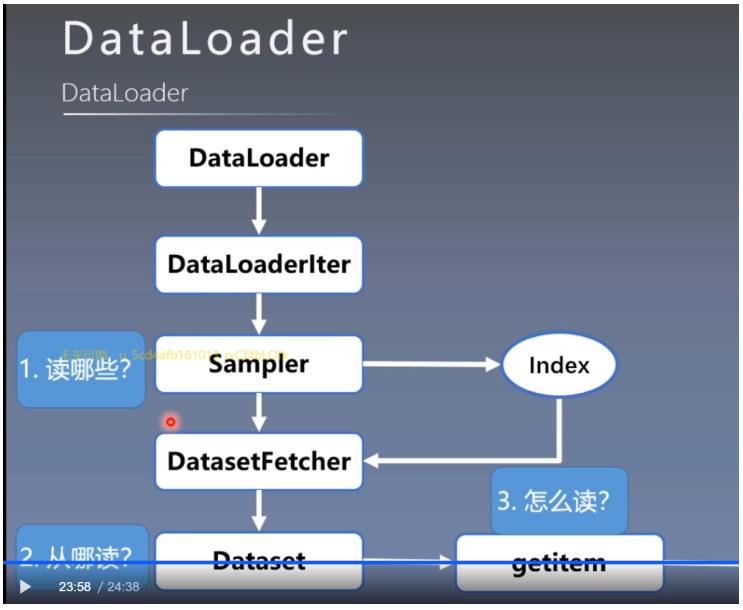
- 在DatasetFetcher中会调用Dataset，

根据我们给定的索引通过getitem函数从硬盘中读取相应的图片Img和标签Label。

- 当读取了

一个batchSize大小的数据之后，通过一个collate_fn将这些数据整理成一个Batc

hDataa的形式，然后就可以输入到模型当中去训练了。



- 结语
  - 这次

课程，学习了纸币二分类模型训练及数据读取模块DataLoader与DataSet的概念。

- 下次课

程，将学习PyTorch的数据预处理模块----transforms。

- **2-2 数据预处理train**

sforms模块机制

- 目录
  - 1. transforms运行机制
  - 2. 数据标准化--transfor

- ms.normalize
  - torchvision：计算机视觉工具包
    - torchvision.tra

nsforms:

常用的图像预处理方法

- torchvision.datasets:

常用数据集的

dataset实现，MNIST，CIFAR-10，ImageNet等

- torchvision.m

odel：常用的模型预训练，AlexNet，VGG，ResNet，GoogLeNet等

- torc

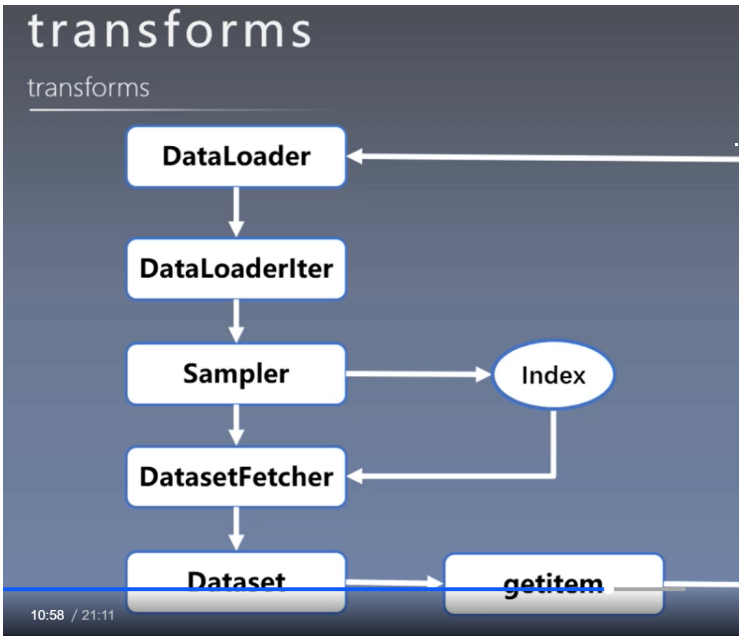
h v i s i o n . t r a n s f o r m s : 常用的图像预处理方法

- 数据中心化
- 数据标准化
- 缩放、裁剪、旋转

、翻转、填充
- 噪声添加、灰度变换、线性变换、仿射变换
- 亮度、饱和度及对比度变换

1. transf

orms运行机制



- 2. 数据标准化--transforms.No

r
m
a
l
i
z
e

- t
  r
  a
  n
  s
  f
  o
  r
  m
  s
  .
  N
  o
  r
  m
  a
  l
  i
  z
  e
  (
  m
  e
  a
  n
  ,

  s
  t
  d
  ,

  i
  n
  p
  l
  a
  c
  e
  =

F a l s e)

- 功能：逐channel的对图像进行标准化。（标准化是指将数据的均值变为0，标准差变

为1）

- output = (input - mean) / std
  - mean：各通道的均值
  - std：

各通道的标准差
- inplace：是否原地操作

- 为什么要对数据进行标准化？
  - 因为对数据进行标准化后，可

以加快模型的收敛。

- 如果我们的训练数据有一个良好的分布的话，可以加速我们模型的收敛

- 结语
  - 本次课学习

了数据预处理transforms的流程与机制以及数据标准化normalize

- 下次课程，我们将会

学习PyTorch的transforms的各种方法

- **2-3 二十二种transforms数据预**

处理方法

- transforms图像增强（一）
- 目录
  - 1. 数据增强 Data Augmentati

on

- 2. transforms ---裁剪
- 3. transforms ---翻转和旋转

o

1. 数据增强

- 数据增强又称为数据增广，数据扩增，它是对训练集进行变换，使训练集更丰富，从而让模

型更具泛化能力。对训练集进行一系列的操作变换，增加训练集的数量，使得训练集更丰富，让模型有更多

的数据去学习去训练，从而提高模型的泛化能力。

- 2. transforms ---裁剪Crop

1) transforms.CenterCrop

- **调试Tricks：**

  Debug模式下的conso

Ie可以通过代码来操作数据，查看、修改等。

- 3. transforms ---翻转和旋转结语

本次学习了数据预处理transforms的数据增强方法

裁剪、翻转和旋转

- 下次课，学习P

yTorch的transforms其他数据增强方法。学习自定义transforms方法

学会自定义transforms方法。

目录

- 3. 自定义transforms方法
  - 1. transforms.transforms方法操作

transform

图像变换

- 1. Pad 对图像边缘进行填充...
- 7. transfor

ms.Lambda

- 功能：用户自定义lambda方法
- lambda：lambda匿名函数

2. tr

ansforms---

transforms方法操作

- transforms的选择操作
  - 3.

自定义

transforms方法

- 自定义transforms要素：
  - 1. 仅接收一个参数，返回一个参数。
  -

2. 注意上下游的输出与输入。

- 结语。下次课程将学习PyTorch的模型模块.
- **本周课后作业**。1.

采用步进(Step into)的调试方法从for i, data in enumerate(

train_loader)

这一行代码开始，进入到每一个被调用函数，直到进入RMBDataset

t类中的__getitem__函数，记录从for循环到RMBDataset的__getite

m
_
_
所设计的类与函数？

- 例如：

- 第一步：for i, data in enumerate

(train_loader)

- 第二步：DataLoader类，__iter__函数

- 第三步

：
*
*
*
类
，
*
*
*
函
数

- 第
n
步
：
R
M
B
D
a
t
a
s
e
t
类
，
_
_
g
e
t
i
t
e
m
_
_
函
数
- 打
卡
要
求

：文字简答

- 训练RMB二分类模型，熟悉数据读取机制，并且从kaggle中下载猫狗二分类训练数据，

自己编写一个DogCatDataset，使得pytorch可以对猫狗二分类训练集进行读取。

- 数据

下载：https://www.kaggle.com/c/dogs-vs-cats-redux-

kernels-edition/data