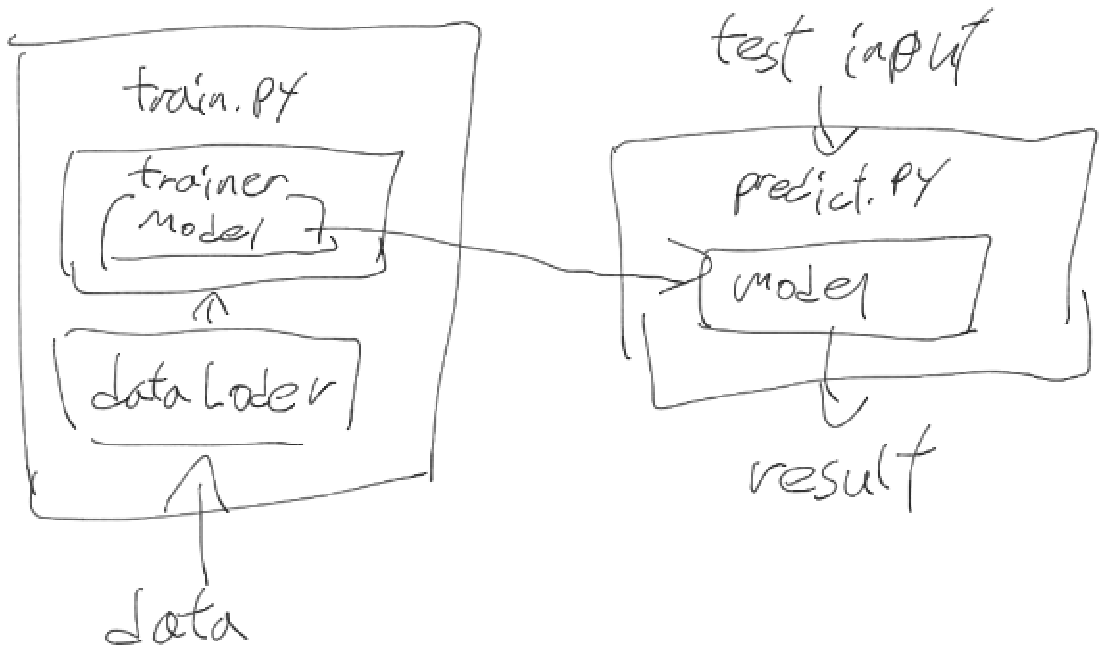
Fastcampus

오리엔테이션 및 간단한 설명

Model.py : 모델 Architecture 정의된 클래스
trainer.py: 모델 학습하기 위한 코드
dataloder.py: 전처리 및 input 을 만들어주는
train.py : hyper-param 받고 선언 학습
Predict.py : Model과 test input을 바탕으로 추론



☆ keras에서 torch로 넘어가기
pytorch 공부하기

Mnist
문제정의: 784(28×28) → f → 10개의 숫자로 분류
image

class ImageClassifier(nn.Module):

def __init__:
input, output 정의

$\boxed{\text{sequental}}$ model
Linear
Leakyrelu
Batchnorm
Softmax(dim=-1)

input size
(batchsize, image vector)
-1이 아닐경우 전체 배치에 대해 축조 -1이면
하나의 이미지로만 축조

def foward:
y = self.layers(x)
return y

trainer file
class Trainer():

init
Model, Optimizer, lossfucn
정의

_train
self.model.train()
data random shuffle

torch.randperm
들어온 input의 차원까지의 수를 랜덤하게 return
f(4) -> [0,3,1,2]

torch.index_select
param = input : input Tensor
dim : index를 바꿀 차원
index : 1차원 배열

index = [0,2]
dim = [0]
이면 output input 0번과 2번만 남김

index 번호

index = [1,3,2,0]
dim = [0]
이면 input tensor가

$$\begin{bmatrix} \text{input}[1] \\ \text{input}[3] \\ \text{input}[2] \\ \text{input}[0] \end{bmatrix}$$

이순서로 바뀜

batch_size로 나누기

batch마다 계산
- forward
- loss 계산
- back probagation
- optimizer
- total loss에 더하기 (float 형으로 바꾸기)

return total_loss/len(x)

_validate

self.model.eval

data shuffle

batch 계산

· forward

· loss 계산

· total_loss += loss

[(batch,784),(bs,1)]

def train(self, t, v, config)

loss -train] — 호출
     -validate]

lowest loss > validate_loss
    best_model 가지고 있기

ephoch 끝나면
    best model로

—train file

def define_argparser
    arg 정의 및 코드 (ex —gpu-id)

def main

device 선언

Mnist load 및 reshape (.view() 사용) flatten

train/val split

shuffle

Model 선언

Adam optimizer

CrossentropyLoss -> crit

trainer 선언

trainer.train

model save

util.py

load mnist

Mnist data load

전처리 ON

강의 끝

위에꺼 대로 Mnist 다시 짜보기

→ 오늘 숙제