eznf框架

开发流程

开发时采用`dev`分支，开发完毕后将`dev`分支向`main`分支提交pull request。

`push`前记得先`git pull origin dev`或者`git pull origin main`

项目需求

Tensor

`eznf/tensor/tensor.py`

- 支持GPU
- 支持list、ndarray的相互转换
- 重载运算符
- 实现backward

Module

`eznf/nn/modules`

基类：Module

- 支持GPU
- 实现构造函数
- 实现`forward()` -> 调用functional里面的计算式

Linear

ReLu

Sigmoid

Tanh

Softmax

Cov1d

Cov2d

Cov3d

MaxPool

AvgPool

MSELoss

CrossEntropyLoss

Hebb

感知机

(时间允许：BatchNorm, Dropout)

Functional

`eznf/nn/functional.py`

基类：被使用于module

- 支持GPU
- 错误检测：输入类型是否为tensor，如果不满足，`raise ValueError('xxx')`

linear

reLu

sigmoid

tanh

softmax

cov1d

cov2d

cov3d

maxPool

avgPool

mse_loss

cross_entropy

Autograd

`eznf/autograd`

function

- 计算图构建
- 反向传播

Optimizer

`eznf/optim`

- 支持GPU

SGD

Adam

DataSet

`eznf/dataset`

MNIST

- `__init__(self, shuffle:bool, path)`
- `get(self, train_size) -> x_train, x_test, y_train, y_test`

Cifar

- `__init__(self, shuffle:bool, path)`
- `get(self, train_size) -> x_train, x_test, y_train, y_test`

Visualization

`eznf/visualization`

VTrain

- `__init__(self, model, x_train, x_test, y_train, y_test)`
- `train(aplha, epoch, optim, criterian)`
    - 画出loss曲线
    - 画出训练集和测试集的准确率曲线

Evaluation

- `__init__(self, model, x_train, x_test, y_train, y_test, n)`
- `CMplot()`
    - 根据所给的类别数n绘制混淆矩阵

- `ROCplot()`
  - 根据所给的类别数n绘制ROC曲线
- `PRplot`
  - 根据所给的类别数n绘制PR曲线

Utils

`eznf/utils/utils.py`

from_numpy

ones

zeros

empty

to_numpy

to_list

框架需要的一些辅助函数

前期分工

接口先按照pytorch的tensor做测试

Tensor

`李帅`

Utils

`刘天一`

Visualization

`秦臻远`

DataSet

陈腾

Functional

李浩宇

```
### DataLoader

`eznf/dataload`

* 支持GPU

* 封装训练集和测试集

（支持batch）
```