

1. Pytorch入門

Tensorを使ってみよう

NumpyをTensorに置きかえたニューラルネットワークのプログラムを作ろう。
reg_2_tensor.pyの空欄を埋めてみてね。 reg_1_numpy.pyも参考にしてね。

ヒント：

行列積 [mm](#); 行列積 [matmul](#); @ [operators](#), [matmul](#), [issue](#); 切り捨て [clamp](#); 自乗 [torch.square](#); べき乗 [torch.pow](#); [clone](#)

Autogradを使ってみよう

Autogradを使って、プログラム（reg_3_autograd.py）を書き直してみよう。

ヒント：

[requires_grad](#); [backward](#); [no_grad](#)

nnを使ってみよう

Pytorchには様々なレイヤーがあります。下記のレイヤーを使って、プログラム（reg_4_nn.py）を書き直してみよう。

ヒント：

[nn](#); [Sequential](#); [Linear](#); [ReLU](#); [MSELoss](#); [parameters](#)

optimizerを使ってみよう

SGD, Adamを使ってみよう！reg_5_optim.pyを埋めてね。

ヒント：

[optim](#); [SGD](#); [Adam](#); [zero_grad](#); [step](#)

練習問題

1. 10層の全結合層のモデル (D_in->H, H->H, H->H, ..., H->D_out) を用いたプログラムを作成せよ.
2. 100層の全結合層のモデル (D_in->H, H->H, H->H, ..., H->D_out) を用いたプログラムを作成せよ.
3. cls_*.pyの空欄を埋めて, Classificationのプログラムを作成せよ.

ヒント :

- モデルを作る方法はいろいろあります. 考えてみてくださいね. [containers](#)
- softmaxとCrossEntropyLossはモジュールを使っても良いです.
[nn.CrossEntropyLoss](#); [functional.cross_entropy](#);
- (難しめ) CrossEntropyLossを自分で実装するなら, [ここ](#)を参考にしてね.
[Extending pytorch](#)