

演習 Python, Pytorchでのニューラルネットワーク実装

1/8の進め方

担当者は全員、

各自がFashionMNIST（ファッション雑貨画像データ）の識別を全結合ニューラルネットワークを使用して行い、その推定精度（accuracy）の高さを競う。

授業時間までに各自でネットワーク構造や学習過程を工夫して精度を高め、その**工夫の意図**と得られた結果について一人40分程度、パワーポイントやKeynote等を使用して説明する。コードのスクショも載せて、どういうコーディングをしたかも報告する。

最も良い精度が出た工夫のみを説明するのではなく、そこまでの過程で得られた知見も報告する。表や図で比較があるとベター。

表の例：学習率/学習アルゴリズムを変えたときの精度の変化

最適化アルゴリズム	Accuracy
SGD	60.52
Adam	72.89

表の例：バッチ正則化を入れることによる精度の変化

	バッチ正則化あり	ドロップアウトあり	いずれもなし
Accuracy			
ドロップアウト			

疑問や実装上困ったことなどもあれば最後に述べること。

実行の詳細

実行環境は任せるが、環境構築不要なGoogle colabが便利。

配布したquickstart_tutorial.ipynbが基本コードとなっている（Pytorch公式で提供されているものに少し中井が手を加えた。コード外の文章が参考に、必要に応じて公式ページを参照すること。）。このコードを自身のやりたい工夫に応じて書き換えることで簡単に実行できる。工夫を加えるたびに新たなファイルとして保存しておく方が結果をまとめやすい。

Python, Pytorchの経験はそれぞれなので、必要に応じて以下のコードも参考にすること。

- ・ 1_intro.ipynb：Pythonの基本的な使い方
- ・ 3_pytorch_tutorial.ipynb：Pytorchの基本的な使い方
- ・ 4_ANDfunction.ipynb：ニューラルネットワークでAND関数を学習
- ・ 5_mnist.ipynb：ニューラルネットワークで手書き数字画像の識別タスク

困ったら中井まで連絡を。数十分手が止まるくらいなら気軽に相談してください。

目的・目標

ここまで本でやってきた内容をコードに落とし込んで理解を深めることを目的としている。

また、今後研究では自身でコードについて調べたり工夫するプロセスが重要になるため、その練習をすることも目的のひとつ。

互いのコードをレビューして、参考になるものがあれば以後取り入れる。