

課題 2

浦川樹

2021 年 12 月 24 日

1 課題内容

MNIST の学習データ 60000 枚の画像からランダムに B 枚をミニバッチとして取り出し、それらに対して順伝播を行う。その後クロスエントロピー誤差を計算する。

2 プログラムの仕様

2.1 プログラムの構成

one-hot vector への変換関数とミニバッチの実装は `utils` モジュールに、クロスエントロピー誤差の実装は `neural` モジュールに行った。それらを利用して課題 2 の実行の本体を `kadai2.py` に実装している。

2.2 one-hot vector への変換

クラスラベルの配列から、各ラベルを変換した one-hot vector の配列を返す関数 `to_categorical()` を実装した。各クラスラベルの値から対応する one-hot vector を構成し、それらを配列にして返している。

2.3 ミニバッチ

全入力データとそのラベルとバッチサイズを受け取り、バッチサイズ個の入力データとラベルからなるタプルを返す関数 `minibatch()` を実装した。`numpy.random.Generator.choice` を用いて入力データの配列とラベルの配列からそれぞれバッチサイズ分取り出している。取り出される入力データとラベルが対応するように、入力データの選択の乱数生成器のシードとラベルの選択の乱数生成器のシードを揃えている。

2.4 クロスエントロピー誤差

正解の one-hot vector の配列とニューラルネットワークの出力 $y^{(2)}$ の配列を受け取って、バッチごとのクロスエントロピー誤差の平均を計算する `cross_entropy()` を実装した。配列の各要素についてクロスエントロピー誤差を計算し、その平均をとっている。

3 実行結果

次のような結果が得られた．

```
cross entropy loss : 2.3796278725782947
```

4 工夫点

ミニバッチ作成と one-hot vector 生成はニューラルネットワーク以外で使用することもあるような処理なので neural モジュールではなく utils モジュールとした．また課題 1 と同様に Numpy の提供するベクトル演算をもちいて実装した．

5 問題点

ミニバッチ作成の乱数生成器のシードにシステム時刻を用いているが，ミニバッチを何度も作成するような場合シードに連続する整数を用いることになる．これが問題ないのかわからない．また，kadai2.py が kadai1.py に比べて体感できるレベルで遅くなっている．