課題2レポート

情報学科 計算機コース 1029275871 芦田聖太

提出日　18/1/11

課題 2

1　ミニバッチ対応＆クロスエントロピー誤差の計算

ミニバッチを入力可能とするように改良し，さらにクロスエントロピー誤差を計算するプログラムを作成する。

仕様

- MNIST のテスト画像 10000 枚の中からランダムに B 枚をミニバッチとして取り出す。
- クロスエントロピー誤差の平均を標準出力に出力する。
- ニューラルネットワークの構造、重みは課題 1 と同じ。
- バッチサイズ B は自由に決める。（今回は 100 にする）
- ミニバッチを取り出す処理はランダムに行う。

2　設計方針

今回の設計では主に以下の 3 つのパートを構成することで、仕様を満たすことにした。

- ミニバッチを選択する
- ミニバッチをニューラルネットワークに適用する
- クロスエントロピーを計算する。

各パートについては実装とプログラムの説明の部分で詳しく解説していく。

3　実装とプログラムの説明

3.1　ミニバッチを選択する

np.random.choice で 0　9999 の 10000 個の数字の中から batch サイズ分の 100 個を選択する。minidata に X の中から選んだ 100 個の数字に対応するインデックスの画像データを格納する。minianwser にも対応するインデックスの正解ラベルを格納しておく。

main.py

```
batch = 100
choice = np.random.choice(len(X), batch, replace=False)
minidata = np.reshape(X[choice], (batch, row*row))
minidata = minidata.T
minianswer = Y[choice]
```

3.2　ミニバッチをニューラルネットに適用する

minidata を中間層への入力を計算する関数 mid に入力し、出力を minimidinput に格納する。シグモイド関数に minimidinput を入力し、出力を minimidout に格納する。出力層への入力を計算する関数 endend に minimidout を入力し、出力を minifininput に格納する。最後に、ソフトマックス関数に minifininput を入力し、出力を minifinout に格納する。以上で、ミニバッチのニューラルネットワークへの適用を完了した。

main.py

```
minimidinput = mid(minidata, middle, row, average, variance, seed)
minimidout = sigmoid.sigmoid(minimidinput)
minifininput = endend(minimidout, end, middle, average1, variance1, seed)
minifinout = softmax.softmax(minifininput)
```

3.3　クロスエントロピーを計算

a に softmax 関数の出力を、b に正解ラベルを入力する。hotone ベクトルにするために np.eye を用いて計算のために転地する。a の各要素で log をとり-1 をかけたものと hotone ベクトルを各要素ごとに掛け算を行い、和をとる。最後に、バッチサイズで割り平均をとり出力する。

CrossEntropy.py

```
def cross(a, b):
    hotone = np.eye(10)[b].T
    sum1 = np.sum(hotone * (np.log(a) * -1))
    ave = sum1 / b.shape[0]
    return ave
```

main.py

```
entropy = CrossEntropy.cross(minifinout, minianswer)
print(entropy)
```

4　実行結果

以下の 7.19260653892 の部分がクロスエントロピーとなっている。

```
input number : 10
6
7.19260653892
```

5　考察

5.1　工夫点

ミニバッチの選択を choice で行うことでスムーズにミニバッチを選択することができた。また、100 個のミニバッチを行列で表すことで、for 文を使わずに一回の処理でニューラルネットワークに入力することができたので早く処理を行うことができている。

5.2　問題点

各層への入力を mid や endend 関数で処理することで main.py から重みが見えなくなっているので、課題 3 以降ではこの関数を使うことができない。つまりあくまで課題 2 のための仕様になってしまっているということが後からではあるがわかった。