

計算機科学実験及演習 4 画像認識 レポート 2

高橋駿一 2018 年度入学 1029-30-3949

提出日: 2020 年 10 月 22 日

課題 2

課題内容

[課題 1] のコードをベースに、ミニバッチ (=複数枚の画像) 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。なお、本レポートは [課題 1] からの差分について記述した。

作成したプログラムの説明

ソースコード 1: ミニバッチのインデックスを作成

```
1 def create_batch(X):
2     batch_size = 100
3     np.random.seed(seed=32)
4     batch_index = np.random.choice(len(X), batch_size)
5     return batch_index
```

ミニバッチのインデックスを作成する関数である。訓練データを受け取り、0 以上訓練データのサイズ以下の整数値を `batch_size` の値の個数取得した。なお、実行ごとに同じ結果を得るために `seed` 値は 32 で固定している。

ソースコード 2: ミニバッチを活用可能な入力層の処理

```
1 def input_layer2(X):
2     batch_index = create_batch(X)
3     input_images = X[batch_index]
4     image_size = input_images[0].size
5     class_num = 10
6     input_vector = input_images.reshape(100,image_size)
7     return input_vector, image_size, batch_index, class_num
```

ミニバッチを入力可能とするように改良した、入力層の処理を行うための関数である。入力データのインデックスを標準入力から取得するのではなく、`create_batch(X)` によって取得し、`input_images` としてミニバッチを作成した。さらに、ミニバッチの状態で各画像データを `image_size` 次元に変更するために `input_images` をバッチサイズ (100) \times `image_size` の行列に変形した。

ソースコード 3: クロスエントロピー誤差の計算

```
1 def cross_entropy_loss(y_pred, y_ans):
2     B = len(y_pred)
3     E = 1 / B * np.sum((-1) * y_ans * np.log(y_pred))
4     return E
```

クロスエントロピー誤差を計算するための関数である。ニューラルネットワークの出力 `y_pred` と正解のクラスを one-hot vector 表記にした `y_ans` を入力として受け取り、クロスエントロピー誤差を計算する。なお、ミニバッチの状態で本計算を実行できる。

ソースコード 4: 課題 2 の実行

```
1 input_vec, image_size, batch_index, class_sum = input_layer2(test_X)
2 batch_label = train_Y[batch_index]
3 y_ans = np.identity(10)[batch_label]
4 # print(batch_label)
5 # print('input', image_size, batch_index, class_sum)
6 W1, b1 = preprocessing(image_size, 30, image_size)
7 y1 = matrix_operation(W1, input_vec, b1)
8 # print('matrix', y1)
9 y1 = sigmoid(y1)
10 # print('sigmoid', y1)
11 W2, b2 = preprocessing(30, class_sum, 30)
12 a = matrix_operation(W2, y1, b2)
13 # print('a', a)
14 y2 = softmax(a)
15 # print(y2)
16 binary_y = postprocessing(y2)
17 # print(binary_y)
18 E = cross_entropy_loss(y2, y_ans)
19 print(E)
```

ここまでで作成した関数を活用して課題 2 の処理を実行した。なお, `input_layer2` で取得したミニバッチのインデックス `batch_index` によって正解のクラス `batch_label` を受け取り, これを one-hot vector 表記に変更したものを `y_ans` としてクロスエントロピー誤差の計算に活用した。

実行結果

実行の結果, クロスエントロピー誤差の 6.912088284247902 が標準出力に出力された。コメントアウトを外すことで各段階でも正しく動作していることが確認できた。

工夫点

- この後の課題でも活用しやすくするために機能別で関数を実装した。
- クロスエントロピー誤差を計算する `cross_entropy_loss` で `for` 文を使わずミニバッチのまま処理できるように記述した。

問題点

- 関数を細かく分けたことでバッチサイズを一箇所で管理できおらず, `create_batch(X)` 内の `batch_size` とは別で `input_layer2(X)` 内で `input_vector` を取得するためにバッチサイズである 100 をそのまま記述してしまっている。