

課題 3

浦川樹

2022 年 1 月 13 日

1 課題内容

MNIST の学習データ 60000 枚の画像から 3 層ニューラルネットワークのパラメータの学習を行う。

2 誤差逆伝播法による誤差関数の勾配計算

バッチ中のあるデータ b に対するクロスエントロピー誤差 E_b は

$$E_b = \sum_{k=1}^C -y_{b,k} \log y_{b,k}^{(2)}$$

で与えられる。よってバッチ全体の誤差 E は

$$E = \frac{1}{B} \sum_{b \in \mathcal{B}} E_b$$

である。ここで \mathcal{B} はバッチのデータ全体の集合を意味する。さて b をバッチ中のあるデータとし、次のような場合を考える。

$$\begin{aligned} a_{b,j}^{(m)} &= \sum_{i=0}^{\mathcal{D}} w_{ji}^{(m)} z_{b,i} \quad (j = 1, \dots, \mathcal{M}) \\ y_{b,j}^{(m)} &= h(a_{b,j}^{(m)}) \\ a_{b,k}^{(m+1)} &= \sum_{j=0}^{\mathcal{M}} w_{kj}^{(m+1)} y_{b,j}^{(m)} \quad (k = 1, \dots, \mathcal{C}) \end{aligned}$$

ただし $z_{b,0} = y_{b,0}^{(m)} = 0$ とする。このとき次の式が成り立つ。

$$\frac{\partial E}{\partial W^{(m)}} = \left(\frac{\partial E_b}{\partial a_{b,j}^{(m)}} \right) \cdot (z_{b,i})$$

$$\left(\frac{\partial E_b}{\partial a_{b,j}^{(m)}} \right) = \left(h'(a_{b,j}^{(m)}) \right) \otimes \left(\tilde{W}^{(m+1)} \cdot \left(\frac{\partial E_b}{\partial a_{b,k}^{(m+1)}} \right) \right)$$

ここで $\frac{\partial E}{\partial W^{(m)}}$ は (j,i) 成分が $\left(\frac{\partial E}{\partial w_{ji}^{(m)}} \right)$ の $(\mathcal{M}+1) \times \mathcal{D}$ 行列, $\left(\frac{\partial E_b}{\partial a_{b,j}^{(m)}} \right)$ は (j,b) 成分で表示された $(\mathcal{M}+1) \times B$ 行列, $(z_{b,i})$ は (b,i) 成分で表示された $B \times \mathcal{D}$ 行列である。また $\left(h'(a_{b,j}^{(m)}) \right)$ は (j,b) 成分で表示された $\mathcal{M} \times B$

行列, $\left(\frac{\partial E_b}{\partial a_{b,k}^{(m+1)}}\right)$ は (k,b) 成分で表示された $\mathcal{C} \times B$ 行列であり,

$$\tilde{W}^{(m+1)} = \begin{pmatrix} w_{11}^{(m+1)} & w_{12}^{(m+1)} & \cdots & w_{1\mathcal{M}}^{(m+1)} \\ w_{21}^{(m+1)} & w_{22}^{(m+1)} & \cdots & w_{2\mathcal{M}}^{(m+1)} \\ \vdots & \vdots & \ddots & \vdots \\ w_{\mathcal{C}1}^{(m+1)} & w_{\mathcal{C}2}^{(m+1)} & \cdots & w_{\mathcal{C}\mathcal{M}}^{(m+1)} \end{pmatrix}$$

である. \otimes は行列のアダマール積を表す. この 2 式を用いると損失関数の勾配を計算することができる.

3 プログラムの仕様

誤差関数の勾配計算や重みの更新は neural モジュールの `fit()` 関数に実装した. それを `kadai3.py` から呼び出している.

4 実行結果

次のような結果が得られた.

```
epoch 0 cross entropy: 0.6849489297103114
epoch 1 cross entropy: 0.33892783681748456
epoch 2 cross entropy: 0.2723342287194881
epoch 3 cross entropy: 0.24425845073320412
epoch 4 cross entropy: 0.21721248400020834
epoch 5 cross entropy: 0.1975543220241951
epoch 6 cross entropy: 0.18297456555145214
epoch 7 cross entropy: 0.17012985644519665
epoch 8 cross entropy: 0.158478132640478
epoch 9 cross entropy: 0.1461636032973061
epoch 10 cross entropy: 0.14117950727287137
epoch 11 cross entropy: 0.13281330008682546
epoch 12 cross entropy: 0.1251475363629807
epoch 13 cross entropy: 0.11953131104941685
epoch 14 cross entropy: 0.11382739492039515
```

5 工夫点

教科書に記載されている損失関数の勾配計算がよくわからなくなったので, できるだけ一般化した場合の逆伝播を自力で計算した. その結果ニューラルネットワークを多層に拡張した場合でも用いることのできる式を得た.

6 問題点

学習の速度が遅い。デモの際に皆さんの学習の様子を見たが、私のより学習が早い上に正解率も高い方がいた。