数理工学実験
テーマ8　ニューラルネットワークによる機械学習

2022年2月2日 提出

工学部情報学科数理工学コース２年
1029-32-7314　岡本淳志

1　課題 22

$f(u) = u^2$ を用いて順に求めていくと、

$$\begin{aligned} u^{(1)} &= x \\ h^{(1)} &= f(u^{(1)}) = x^2 \\ u^{(2)} &= w^{(2)} h^{(1)} = w_2 x^2 \\ h^{(2)} &= f(u^{(2)}) = w_2^2 x^4 \\ u^{(3)} &= w^{(3)} h^{(2)} = w_2^2 w_3 x^4 \\ \hat{y} &= f^{(3)}(u^{(3)}) = w_2^4 w_3^2 x^8 \\ \mathcal{L}(\hat{y}) &= \hat{y}^2 = w_2^8 w_3^4 x^{16} \end{aligned} \tag{1}$$

である。また、

$$\frac{\partial \mathcal{L}}{\partial w_2} = 8 w_2^7 w_3^4 x^{16}, \ \frac{\partial \mathcal{L}}{\partial w_3} = 4 w_2^8 w_3^3 x^{16} \tag{2}$$

である。

この式（2）の値を逆誤差伝播法を用いて計算する。

$$\begin{aligned} \delta^{(3)} &= (2\hat{y})(2u^{(3)}) = 4 w_2^6 w_3^3 x^{12} \\ \delta^{(2)} &= \delta^{(3)} w^{(3)} (2u^{(2)}) = 8 w_2^7 w_3^4 x^{14} \end{aligned} \tag{3}$$

これを用いると、

$$\begin{aligned} \delta^{(3)} h^{(2)} &= 4 w_2^8 w_3^3 x^{16} \\ &= \frac{\partial \mathcal{L}}{\partial w_3} \\ \delta^{(2)} h^{(1)} &= 8 w_2^7 w_3^4 x^{16} \\ &= \frac{\partial \mathcal{L}}{\partial w_2} \end{aligned} \tag{4}$$

以上より、偏微分を計算することなく、簡単な代入操作を繰り返し、$\frac{\partial \mathcal{L}}{\partial w_2}$, $\frac{\partial \mathcal{L}}{\partial w_3}$ を求めることができた。これが、逆誤差伝播法である。

2　課題 23

シグモイド関数

$$f(u) = \frac{1}{1 + e^{-u}} \tag{5}$$

に関して、

$$f'(u) = \frac{e^u}{(1 + e^u)^2}, \quad f''(u) = \frac{-e^{2u} + e^u}{(1 + e^u)^3} = \frac{e^u(-e^u + 1)}{(1 + e^u)^3} \tag{6}$$

である。$u > 0$ のとき、$f''(u) < 0$ で $f'(u)$ は単調減少。$u < 0$ のとき、$f''(u) > 0$ で $f'(u)$ は単調増加。よって、$f'(u)$ は $u = 0$ で最大値 $f'(0) = 0.25$ をとる。これより、活性化関数にシグモイド関数を用いると勾配消失を起こす原因となりやすい。

一方で、ReLU

$$f(u) = \begin{cases} u & (u \geq 0) \\ 0 & (u < 0) \end{cases} \tag{7}$$

に関して、

$$f'(u) = \begin{cases} 1 & (u \geq 0) \\ 0 & (u < 0) \end{cases} \tag{8}$$

で、$f'(u)$ は一定である。

このことから、ReLU は勾配消失を起こす可能性が低いと言える。

3 課題 24

結果は以下の表のようになった。

表 1:

損失関数	損失:loss	精度:acc
クロスエントロピー	2.502	0.9304
MSE	0.0107	0.9328

よってこの問題において、損失関数をクロスエントロピーにした場合と MSE にした場合とでは、精度はほぼ変わらないが損失は MSE の場合の方が小さくなることが分かった。

4 課題 27

隠れ層は一つにした。結果は以下の表のようになった。

表 2:

ノードの数	損失:loss	精度:acc	実行時間 (s)
10	0.2299	0.9356	20.8820
10^2	0.0379	0.9898	29.1658
10^3	7.0967e-05	1.0000	24.1539
10^4	0.0119	0.9963	82.6326
10^5	-	-	-

ノードの数を 10^5 にしたときは途中で実行が止まったため計算できなかったが、Epoch 1/10 の実行スピードから見ても、ノードの数が 10^4 の場合よりも実行に時間がかかることが分かった。

以上より、この問題についてノードの数を 10^3 にするのが適切であると分かった。

5 課題 28

エポックを増やすと最初のうち（エポックが小さいうち）は損失が下がり、精度が上がっていった。あるエポックの数（今回はエポック 13）からその変化は単調には見られなくなり、損失と精度は上下しはじめた。しかし、全体的に見るとエポックの数が増えるにつれて損失は下がり、精度は上がっていった。（すなわち、エポック 12 よりエポック 100 付近の方が損失は小さく、精度は高くなっていた。）

このことから、単にエポックを増やせば良い結果が得られるとは一概には言えないと分かった。このことは、過学習が起こってしまうことが大きな要因の一つであると考えた。

6 課題 29

結果は以下の表のようになった。

表 3:

最適化:optimizer	損失:loss	精度:acc	実行時間 (s)
GradientDescent	0.2011	0.9424	27.1282
Adadelta	1.3862	0.7729	41.8935
Adagrad	0.3417	0.9034	41.8053
Adam	0.0141	0.9955	30.2502
Ftrl	2.3018	0.1124	42.2380
RMSProp	0.0162	0.9955	35.3913

このように、最適化の手法によって損失や精度、実行時間に差があることが分かった。問題に適した手法を用いるには、いくつかの最適化手法を試すのが賢明であると思った。

7　参考文献

- 数理工学実験テキスト
- Cloud LaTeX
  https://cloudlatex.io
- LaTeX コマンド一覧（リスト）
  https://medemanabu.net/latex/latex-commands-list/