计算两层神经网络参数迭代的 JACOBI 矩阵：以 MNIST 数据集为例

ZEYU XIE[1], ANGXIU NI[2,3]

1. 变量定义

1.1. 输入输出.

(a) 学习率 $\alpha = 0.1$

(b) 一个 batch 的大小 $k = 60000$

(c) 输入层的输入 $X \in \mathbb{R}^{k\times 784}$

$$X = \begin{bmatrix} x^1 \\ x^2 \\ \vdots \\ x^k \end{bmatrix} = \begin{bmatrix} x_1^1 & x_2^1 & \cdots & x_{784}^1 \\ x_1^2 & x_2^2 & \cdots & x_{784}^2 \\ \vdots & \vdots & \ddots & \vdots \\ x_1^k & x_2^k & \cdots & x_{784}^k \end{bmatrix} \tag{1}$$

(d) 隐藏层的输入 $A_1 \in \mathbb{R}^{k\times 50}$

$$A_1 = \begin{bmatrix} a_1^1 & a_2^1 & \cdots & a_{50}^1 \\ a_1^2 & a_2^2 & \cdots & a_{50}^2 \\ \vdots & \vdots & \ddots & \vdots \\ a_1^k & a_2^k & \cdots & a_{50}^k \end{bmatrix} \tag{2}$$

(e) 隐藏层的输出 $Z_1 \in \mathbb{R}^{k\times 50}$

$$Z_1 = \begin{bmatrix} z_1^1 & z_2^1 & \cdots & z_{50}^1 \\ z_1^2 & z_2^2 & \cdots & z_{50}^2 \\ \vdots & \vdots & \ddots & \vdots \\ z_1^k & z_2^k & \cdots & z_{50}^k \end{bmatrix} \tag{3}$$

[1] Department of Mathematics, Tsinghua University, Beijing, China.

[2] Department of Mathematics, University of California, Irvine, USA

[3] Yau Mathematical Sciences Center, Tsinghua University, Beijing, China.

E-mail address: niangxiu@gmail.com.

Date: 2024 年 4 月 5 日.

(f) 输出层的输入 $A_2 \in \mathbb{R}^{k\times 10}$（$Z_1 = A_2$，是同一个矩阵）

$$A_2 = \begin{bmatrix} a'^1_1 & a'^1_2 & \cdots & a'^1_{10} \\ a'^2_1 & a'^2_2 & \cdots & a'^2_{10} \\ \vdots & \vdots & \ddots & \vdots \\ a'^k_1 & a'^k_2 & \cdots & a'^k_{10} \end{bmatrix} \tag{4}$$

(g) 输出层的输出 $Y \in \mathbb{R}^{k\times 10}$

$$Y = \begin{bmatrix} y^1 \\ y^2 \\ \vdots \\ y^k \end{bmatrix} = \begin{bmatrix} y^1_1 & y^1_2 & \cdots & y^1_{10} \\ y^2_1 & y^2_2 & \cdots & y^2_{10} \\ \vdots & \vdots & \ddots & \vdots \\ y^k_1 & y^k_2 & \cdots & y^k_{10} \end{bmatrix} \tag{5}$$

(h) 正确答案 $T \in \mathbb{R}^{k\times 10}$

$$T = \begin{bmatrix} t^1 \\ t^2 \\ \vdots \\ t^k \end{bmatrix} = \begin{bmatrix} t^1_1 & t^1_2 & \cdots & t^1_{10} \\ t^2_1 & t^2_2 & \cdots & t^2_{10} \\ \vdots & \vdots & \ddots & \vdots \\ t^k_1 & t^k_2 & \cdots & t^k_{10} \end{bmatrix} \tag{6}$$

1.2. **神经网络参数.**

(a) 输入层到隐藏层的权重 $W_1 \in \mathbb{R}^{784\times 50}$

$$W_1 = \begin{bmatrix} w^1_1 & w^1_2 & \cdots & w^1_{50} \\ w^2_1 & w^2_2 & \cdots & w^2_{50} \\ \vdots & \vdots & \ddots & \vdots \\ w^{784}_1 & w^{784}_2 & \cdots & w^{784}_{50} \end{bmatrix} \tag{7}$$

(b) 输入层到隐藏层的偏置 $b_1 \in \mathbb{R}^{50}$

$$b_1 = \begin{bmatrix} b^1_1 & b^1_2 & \cdots & b^1_{50} \end{bmatrix} \tag{8}$$

(c) 隐藏层到输出层的权重 $W_2 \in \mathbb{R}^{50\times 10}$

(9)
$$W_2 = \begin{bmatrix} w_1^1 & w_2^1 & \cdots & w_{10}^1 \\ w_1^2 & w_2^2 & \cdots & w_{10}^2 \\ \vdots & \vdots & \ddots & \vdots \\ w_1^{50} & w_2^{50} & \cdots & w_{10}^{50} \end{bmatrix}$$

(d) 隐藏层到输出层的偏置 $b_2 \in \mathbb{R}^{10}$

(10)
$$b_2 = \begin{bmatrix} b_2^1 & b_2^2 & \cdots & b_{10}^1 \end{bmatrix}$$

1.3. **神经网络参数的处理.**

(a) 第 t 步迭代的参数向量 $\theta^t \in \mathbb{R}^{39760}$ [1]

(11)
$$\theta^t = (\theta_1^t, \theta_2^t, \cdots, \theta_{39760}^t)$$

(b) 每一步迭代的参数向量的 Jacobi 矩阵 $Df(\theta^t) \in \mathbb{R}^{39760 \times 39760}$

(12)
$$Df(\theta^t) = \begin{bmatrix} \frac{\partial\theta_1^{t+1}}{\partial\theta_1^t} & \frac{\partial\theta_1^{t+1}}{\partial\theta_2^t} & \cdots & \frac{\partial\theta_1^{t+1}}{\partial\theta_{39760}^t} \\ \frac{\partial\theta_2^{t+1}}{\partial\theta_1^t} & \frac{\partial\theta_2^{t+1}}{\partial\theta_2^t} & \cdots & \frac{\partial\theta_2^{t+1}}{\partial\theta_{39760}^t} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial\theta_{39760}^{t+1}}{\partial\theta_1^t} & \frac{\partial\theta_{39760}^{t+1}}{\partial\theta_2^t} & \cdots & \frac{\partial\theta_{39760}^{t+1}}{\partial\theta_{39760}^t} \end{bmatrix}$$

2. 神经网络的结构

2.1. **输入层.** 输入层的维度为 784，即 28×28 的图片的 784 个像素。

2.2. **隐藏层.** 隐藏层的维度为 50。

先进行线性变换

(13)
$$A_1 = XW_1 + b_1$$

再经 $sigmoid$ 激活函数处理

[1]其中 $39760 = 784 \times 50 + 50 + 50 \times 10 + 10$

(14)

$$Z_1 = sigmoid(A_1) = sigmoid(\begin{bmatrix} a_1^1 & a_2^1 & \cdots & a_{50}^1 \\ a_1^2 & a_2^2 & \cdots & a_{50}^2 \\ \vdots & \vdots & \ddots & \vdots \\ a_1^k & a_2^k & \cdots & a_{50}^k \end{bmatrix}) = \begin{bmatrix} \frac{1}{1+\exp(a_1^1)} & \frac{1}{1+\exp(a_2^1)} & \cdots & \frac{1}{1+\exp(a_{50}^1)} \\ \frac{1}{1+\exp(a_1^2)} & \frac{1}{1+\exp(a_2^2)} & \cdots & \frac{1}{1+\exp(a_{50}^2)} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{1+\exp(a_1^k)} & \frac{1}{1+\exp(a_2^k)} & \cdots & \frac{1}{1+\exp(a_{50}^k)} \end{bmatrix}$$

2.3. **输出层.** 输出层的维度为 10

先进行线性变换

$$A_2 = Z_1 W_2 + b_2 \tag{15}$$

再经过 $softmax$ 层

(16)

$$Y = softmax(A_2) = softmax(\begin{bmatrix} {a'}^1 \\ {a'}^2 \\ \vdots \\ {a'}^k \end{bmatrix}) = \begin{bmatrix} \frac{\exp({a'}_1^1)}{\sum_{l=1}^{10}\exp({a'}_l^1)} & \frac{\exp({a'}_2^1)}{\sum_{l=1}^{10}\exp({a'}_l^1)} & \cdots & \frac{\exp({a'}_{10}^1)}{\sum_{l=1}^{10}\exp({a'}_l^1)} \\ \frac{\exp({a'}_1^2)}{\sum_{l=1}^{10}\exp({a'}_l^2)} & \frac{\exp({a'}_2^2)}{\sum_{l=1}^{10}\exp({a'}_l^2)} & \cdots & \frac{\exp({a'}_{10}^2)}{\sum_{l=1}^{10}\exp({a'}_l^2)} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\exp({a'}_1^k)}{\sum_{l=1}^{10}\exp({a'}_l^k)} & \frac{\exp({a'}_2^k)}{\sum_{l=1}^{10}\exp({a'}_l^k)} & \cdots & \frac{\exp({a'}_{10}^k)}{\sum_{l=1}^{10}\exp({a'}_l^k)} \end{bmatrix}$$

2.4. **损失函数.** 我们用交叉熵作为损失函数，即

$$L = -\frac{1}{k}\sum_{i=1}^{k}\sum_{j=1}^{10} y_j^i \log y_j^i \tag{17}$$

3. 梯度和 Jacobi 矩阵

第 t 步迭代的参数为

$$\theta^t = (W_1(t), b_1(t), W_2(t), b_2(t)) = (\theta_1^t, \theta_2^t, \cdots, \theta_{39760}^t) \tag{18}$$

此时对应的梯度为

$$grad(t) = (\frac{\partial L}{\partial \theta_1^t}, \frac{\partial L}{\partial \theta_2^t}, \cdots, \frac{\partial L}{\partial \theta_{39760}^t}) = (\frac{\partial L}{\partial W_1^t}, \frac{\partial L}{\partial b_1^t}, \frac{\partial L}{\partial W_2^t}, \frac{\partial L}{\partial b_2^t}) \tag{19}$$

神经网络迭代过程即

$$\theta^{t+1} = \theta^t - \alpha \cdot grad(t) \tag{20}$$

也即

$$
\begin{aligned}
W_1^{t+1} &= W_1^t - \alpha \cdot \frac{\partial L}{\partial W_1^t} \\
b_1^{t+1} &= b_1^t - \alpha \cdot \frac{\partial L}{\partial b_1^t} \\
W_2^{t+1} &= W_2^t - \alpha \cdot \frac{\partial L}{\partial W_2^t} \\
b_2^{t+1} &= b_2^t - \alpha \cdot \frac{\partial L}{\partial b_2^t}
\end{aligned} \tag{21}
$$

因此迭代的 Jacobi 矩阵为

$$
J(t) = \begin{bmatrix}
\frac{\partial \theta_1^{t+1}}{\partial \theta_1^t} & \frac{\partial \theta_1^{t+1}}{\partial \theta_2^t} & \cdots & \frac{\partial \theta_1^{t+1}}{\partial \theta_{39760}^t} \\
\frac{\partial \theta_2^{t+1}}{\partial \theta_1^t} & \frac{\partial \theta_2^{t+1}}{\partial \theta_2^t} & \cdots & \frac{\partial \theta_2^{t+1}}{\partial \theta_{39760}^t} \\
\vdots & \vdots & \ddots & \vdots \\
\frac{\partial \theta_{39760}^{t+1}}{\partial \theta_1^t} & \frac{\partial \theta_{39760}^{t+1}}{\partial \theta_2^t} & \cdots & \frac{\partial \theta_{39760}^{t+1}}{\partial \theta_{39760}^t}
\end{bmatrix} \tag{22}
$$

也即

$$
J(t) = \begin{bmatrix}
\frac{\partial W_1^{t+1}}{\partial W_1^t} & \frac{\partial W_1^{t+1}}{\partial b_1^t} & \frac{\partial W_1^{t+1}}{\partial W_2^t} & \frac{\partial W_1^{t+1}}{\partial b_2^t} \\
\frac{\partial b_1^{t+1}}{\partial W_1^t} & \frac{\partial b_1^{t+1}}{\partial b_1^t} & \frac{\partial b_1^{t+1}}{\partial W_2^t} & \frac{\partial b_1^{t+1}}{\partial b_2^t} \\
\frac{\partial W_2^{t+1}}{\partial W_1^t} & \frac{\partial W_2^{t+1}}{\partial b_1^t} & \frac{\partial W_2^{t+1}}{\partial W_2^t} & \frac{\partial W_2^{t+1}}{\partial b_2^t} \\
\frac{\partial b_2^{t+1}}{\partial W_1^t} & \frac{\partial b_2^{t+1}}{\partial b_1^t} & \frac{\partial b_2^{t+1}}{\partial W_2^t} & \frac{\partial b_2^{t+1}}{\partial b_2^t}
\end{bmatrix} \tag{23}
$$

4. Jacobi 矩阵的计算

4.1. 基本性质.

4.1.1. *sigmoid 函数的 Jacobi 矩阵.* 假设 $A \in \mathbb{R}^{n\times m}$ 是同一个矩阵，则 $\frac{\partial(sigmoid(A))}{\partial A}$ 的计算公式为

(24)

$$
\frac{\partial(sigmoid(A))}{\partial A} = diag\{\frac{d(sigmoid(x_1^1))}{dx_1^1}, \frac{d(sigmoid(x_2^1))}{dx_2^1}, \cdots, \frac{d(sigmoid(x_m^n))}{dx_m^n}\}
$$

4.1.2. *softmax 函数的 Jacobi 矩阵.* 假设 A 是同一个矩阵，则 $\frac{\partial(softmax(A))}{\partial A}$ 的计算公式为

$$
\frac{\partial(softmax(A))}{\partial A} = softmax(A) \cdot (I - softmax(A)) \tag{25}
$$

$$
\begin{aligned}
\frac{\partial W_1(t+1)}{\partial W_1(t)} &= \frac{\partial(W_1(t)-\alpha\cdot\frac{\partial L}{\partial W_1(t)})}{\partial W_1(t)} \\
&= I-\alpha\frac{\partial^2 L}{\partial W_1\partial W_1} \\
&= I-\alpha\frac{\partial}{\partial W_1}\left(\frac{\partial L}{\partial W_1}\right) \\
&= I-\alpha\frac{\partial}{\partial W_1}\left(\frac{\partial L}{\partial Y}\frac{\partial Y}{\partial A_2}\frac{\partial A_2}{\partial W_1}\right) \\
&= I-\alpha\frac{\partial}{\partial W_1}\left(\frac{\partial L}{\partial Y}\frac{\partial Y}{\partial A_2}\right)W_2^T Z_1^T \\
&= I-\alpha\left(\frac{\partial}{\partial W_1}\left(\frac{\partial L}{\partial Y}\right)\frac{\partial Y}{\partial A_2}+\frac{\partial L}{\partial Y}\frac{\partial}{\partial W_1}\left(\frac{\partial Y}{\partial A_2}\right)\right)W_2^T Z_1^T
\end{aligned} \tag{26}
$$

$$
\frac{\partial {W_1}'}{\partial b_1} = -\alpha\left(softmax'(A_2)W_2^T sigmoid'(A_1)\begin{bmatrix}1 & 1 & \cdots & 1\\ 1 & 1 & \cdots & 1\\ \vdots & \vdots & \ddots & \vdots\\ 1 & 1 & \cdots & 1\end{bmatrix}\right)X \tag{27}
$$

$$
\begin{aligned}
\frac{\partial {W_1}'}{\partial W_2} &= -\alpha\left(softmax'(A_2)W_2^T sigmoid'(A_1)\begin{bmatrix}x_1^1 & x_2^1 & \cdots & x_{784}^1\\ x_1^2 & x_2^2 & \cdots & x_{784}^2\\ \vdots & \vdots & \ddots & \vdots\\ x_1^k & x_2^k & \cdots & x_{784}^k\end{bmatrix}\right) \\
\frac{\partial {W_1}'}{\partial b_2} &= -\alpha\left(softmax'(A_2)W_2^T sigmoid'(A_1)\begin{bmatrix}1 & 1 & \cdots & 1\\ 1 & 1 & \cdots & 1\\ \vdots & \vdots & \ddots & \vdots\\ 1 & 1 & \cdots & 1\end{bmatrix}\right)
\end{aligned} \tag{28}
$$

$$
\begin{aligned}
\frac{\partial {b_1}'}{\partial W_1} &= -\alpha\left(softmax'(A_2)W_2^T sigmoid'(A_1)\begin{bmatrix}x_1^1 & x_2^1 & \cdots & x_{784}^1\\ x_1^2 & x_2^2 & \cdots & x_{784}^2\\ \vdots & \vdots & \ddots & \vdots\\ x_1^k & x_2^k & \cdots & x_{784}^k\end{bmatrix}\right) \\
\frac{\partial {b_1}'}{\partial b_1} &= -\alpha\left(softmax'(A_2)W_2^T sigmoid'(A_1)\begin{bmatrix}1 & 1 & \cdots & 1\\ 1 & 1 & \cdots & 1\\ \vdots & \vdots & \ddots & \vdots\\ 1 & 1 & \cdots & 1\end{bmatrix}\right)
\end{aligned} \tag{29}
$$

$$
\begin{aligned}
\frac{\partial {b_1}'}{\partial W_2} &= -\alpha \left(softmax'(A_2) W_2^T sigmoid'(A_1) \begin{bmatrix} x_1^1 & x_2^1 & \cdots & x_{784}^1 \\ x_1^2 & x_2^2 & \cdots & x_{784}^2 \\ \vdots & \vdots & \ddots & \vdots \\ x_1^k & x_2^k & \cdots & x_{784}^k \end{bmatrix} \right) \\
\frac{\partial {b_1}'}{\partial b_2} &= -\alpha \left(softmax'(A_2) W_2^T sigmoid'(A_1) \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & 1 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & 1 \end{bmatrix} \right)
\end{aligned} \tag{30}
$$

$$
\begin{aligned}
\frac{\partial {W_2}'}{\partial W_1} &= -\alpha \left(softmax'(A_2) \begin{bmatrix} a_1^1 & a_2^1 & \cdots & a_{50}^1 \\ a_1^2 & a_2^2 & \cdots & a_{50}^2 \\ \vdots & \vdots & \ddots & \vdots \\ a_1^k & a_2^k & \cdots & a_{50}^k \end{bmatrix} \right) \\
\frac{\partial {W_2}'}{\partial b_1} &= -\alpha \left(softmax'(A_2) \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & 1 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & 1 \end{bmatrix} \right)
\end{aligned} \tag{31}
$$

$$
\begin{aligned}
\frac{\partial {W_2}'}{\partial W_2} &= I - \alpha \frac{\partial^2 Y}{\partial W_2 \partial W_2} \\
&= I - \alpha \frac{\partial}{\partial W_2} \left(softmax'(A_2) A_1 \right) \\
&= I - \alpha \frac{\partial}{\partial W_2} \left(softmax'(A_2) \right) A_1 \\
&= I - \alpha \left(\frac{\partial softmax'(A_2)}{\partial A_2} \frac{\partial A_2}{\partial W_2} \right) A_1 \\
&= I - \alpha \left(softmax'(A_2) \frac{\partial A_2}{\partial W_2} \right) A_1 \\
&= I - \alpha \left(softmax'(A_2) \begin{bmatrix} a_1^1 & a_2^1 & \cdots & a_{50}^1 \\ a_1^2 & a_2^2 & \cdots & a_{50}^2 \\ \vdots & \vdots & \ddots & \vdots \\ a_1^k & a_2^k & \cdots & a_{50}^k \end{bmatrix} \right) A_1
\end{aligned} \tag{32}
$$

$$\frac{\partial {W_2}'}{\partial b_2} = -\alpha \left(softmax'(A_2) \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & 1 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & 1 \end{bmatrix} \right) \tag{33}$$

$$\begin{aligned} \frac{\partial {b_2}'}{\partial W_1} &= -\alpha \left(softmax'(A_2) \begin{bmatrix} a_1^1 & a_2^1 & \cdots & a_{50}^1 \\ a_1^2 & a_2^2 & \cdots & a_{50}^2 \\ \vdots & \vdots & \ddots & \vdots \\ a_1^k & a_2^k & \cdots & a_{50}^k \end{bmatrix} \right) \\ \frac{\partial {b_2}'}{\partial b_1} &= -\alpha \left(softmax'(A_2) \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & 1 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & 1 \end{bmatrix} \right) \end{aligned} \tag{34}$$

$$\begin{aligned} \frac{\partial {b_2}'}{\partial W_2} &= -\alpha \left(softmax'(A_2) \begin{bmatrix} a_1^1 & a_2^1 & \cdots & a_{50}^1 \\ a_1^2 & a_2^2 & \cdots & a_{50}^2 \\ \vdots & \vdots & \ddots & \vdots \\ a_1^k & a_2^k & \cdots & a_{50}^k \end{bmatrix} \right) \\ \frac{\partial {b_2}'}{\partial b_2} &= -\alpha \left(softmax'(A_2) \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & 1 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & 1 \end{bmatrix} \right) \end{aligned} \tag{35}$$

将以上 $4 \times 4 = 16$ 个 Jacobi 矩阵计算公式代入 $Df(W_1, b_1, W_2, b_2)$ 的定义，即得到 $Df(W_1, b_1, W_2, b_2)$ 的表达式