

Machine Learning HW4

Fall 2020

Study Group: Hsuan-Cheng Chen, Cheng-Hao Chou

PoKang Chen

1.(a)

Machine Learning

(a) $X \in \mathbb{R}^{N \times D_m}$
 $W \in \mathbb{R}^{D_m \times D_{out}}$ $b \in \mathbb{R}^{D_{out}}$ $Y \in \mathbb{R}^{N \times D_{out}}$

$$y^{(i)} = W^T x^{(i)} + b \in \mathbb{R}^{D_{out}}$$

$$Y = XW + B, \quad B = \mathbb{R}^{N \times D_{out}} = \begin{bmatrix} b_1 \\ b_1 \\ \vdots \end{bmatrix}$$

Diagram illustrating the dimensions of the variables and their derivatives:

- $\frac{\partial L}{\partial W} \in \mathbb{R}^{D_m \times D_{out}}$
- $\frac{\partial L}{\partial b} \in \mathbb{R}^{D_{out}}$
- $\frac{\partial L}{\partial X} \in \mathbb{R}^{N \times D_m}$
- $\frac{\partial L}{\partial y} \in \mathbb{R}^{N \times D_{out}}$

Diagram illustrating the dimensions of the variables and their derivatives:

- $\frac{\partial L}{\partial W_{i,j}} \in \mathbb{R}^{D_m \times D_{out}}$
- $\frac{\partial L}{\partial W_{i,j}} \in \mathbb{R}^{D_m \times D_{out}}$
- $\frac{\partial L}{\partial W_{i,j}} \in \mathbb{R}^{D_m \times D_{out}}$

$$y^{(i)} = W^T x^{(i)} + b \Rightarrow y_m^{(n)} = W_m^T x^{(i)} + b_m$$

$$\frac{\partial L}{\partial W_{i,j}} = \sum_n \sum_m \frac{\partial L}{\partial y_m^{(n)}} \frac{\partial y_m^{(n)}}{\partial W_{i,j}} = \sum_n \sum_m \frac{\partial L}{\partial y_m^{(n)}} \frac{W_m^T x^{(i)} + b_m}{\partial W_{i,j}} = \sum_n \frac{\partial L}{\partial y_m^{(n)}} x_i^{(n)} \quad (m \rightarrow j)$$

$$= \sum_n x_i^{(n)} \cdot \frac{\partial L}{\partial y_j^{(n)}} = x_i^{(1)} \frac{\partial L}{\partial y_j^{(1)}} + x_i^{(2)} \frac{\partial L}{\partial y_j^{(2)}} + \dots + x_i^{(N)} \frac{\partial L}{\partial y_j^{(N)}} = \frac{\partial L}{\partial y_j}$$

$$\Rightarrow \frac{\partial L}{\partial W} = \boxed{X^T \frac{\partial L}{\partial Y}} \quad \text{①}$$

$$\frac{\partial L}{\partial b_j} = \sum_n \sum_m \frac{\partial L}{\partial y_m^{(n)}} \frac{\partial y_m^{(n)}}{\partial b_j} = \sum_n \frac{\partial L}{\partial y_j^{(n)}} \cdot 1 = \frac{\partial L}{\partial y_j^{(1)}} + \frac{\partial L}{\partial y_j^{(2)}} + \dots + \frac{\partial L}{\partial y_j^{(N)}} = J \cdot \frac{\partial L}{\partial Y}$$

for that $J = [1 \dots 1]_{N \times 1} \Rightarrow \frac{\partial L}{\partial b} = \frac{\partial L}{\partial Y} \cdot J$

$$\frac{\partial L}{\partial x_j^{(i)}} = \sum_n \sum_m \frac{\partial L}{\partial y_m^{(n)}} \frac{\partial y_m^{(n)}}{\partial x_j^{(i)}} = \sum_n \sum_m \frac{\partial L}{\partial y_m^{(n)}} \frac{\partial (W_m^T x^{(i)} + b_m)}{\partial x_j^{(i)}} = \sum_n \sum_m \frac{\partial L}{\partial y_m^{(n)}} W_{m,j}$$

$$= \sum_m \frac{\partial L}{\partial y_m^{(i)}} \cdot W_{m,j} = \frac{\partial L}{\partial y_1^{(i)}} \cdot W_{1,j} + \frac{\partial L}{\partial y_2^{(i)}} \cdot W_{2,j} + \dots + \frac{\partial L}{\partial y_m^{(i)}} \cdot W_{m,j}$$

$$\therefore \frac{\partial L}{\partial X} = \boxed{\frac{\partial L}{\partial Y} \cdot W^T} = N \times D_m \quad \text{②}$$

1.(b)

(b)

$$Y = \text{ReLU}(X).$$

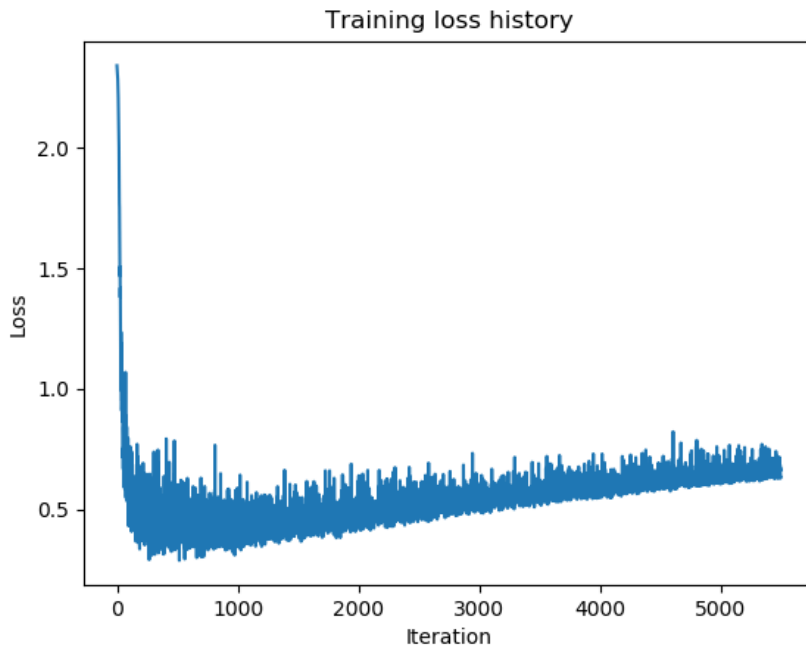
$$N \left[\begin{array}{c} \\ \text{Dim} \end{array} \right]$$

$$\frac{\partial L}{\partial X} = \frac{\partial L}{\partial Y} \frac{\partial Y}{\partial X} = \begin{cases} X \geq 0, & \frac{\partial L}{\partial Y} \times 1 \\ X < 0, & \frac{\partial L}{\partial Y} \times 0 \end{cases}$$

1.(c)

2.

Test accuracy 0.9736



3.

(Epoch 0 / 10) train acc: 0.137000; val_acc: 0.176600

(Epoch 1 / 10) train acc: 0.932000; val_acc: 0.940200

(Epoch 2 / 10) train acc: 0.979000; val_acc: 0.957000

(Epoch 3 / 10) train acc: 0.980000; val_acc: 0.963400

(Epoch 4 / 10) train acc: 0.980000; val_acc: 0.964800

(Epoch 5 / 10) train acc: 0.985000; val_acc: 0.971400

(Epoch 6 / 10) train acc: 0.995000; val_acc: 0.971000

(Epoch 7 / 10) train acc: 0.993000; val_acc: 0.971200

(Epoch 8 / 10) train acc: 0.998000; val_acc: 0.973600

(Epoch 9 / 10) train acc: 1.000000; val_acc: 0.975200

(Epoch 10 / 10) train acc: 0.999000; val_acc: 0.976400

Test accuracy 0.9715

4

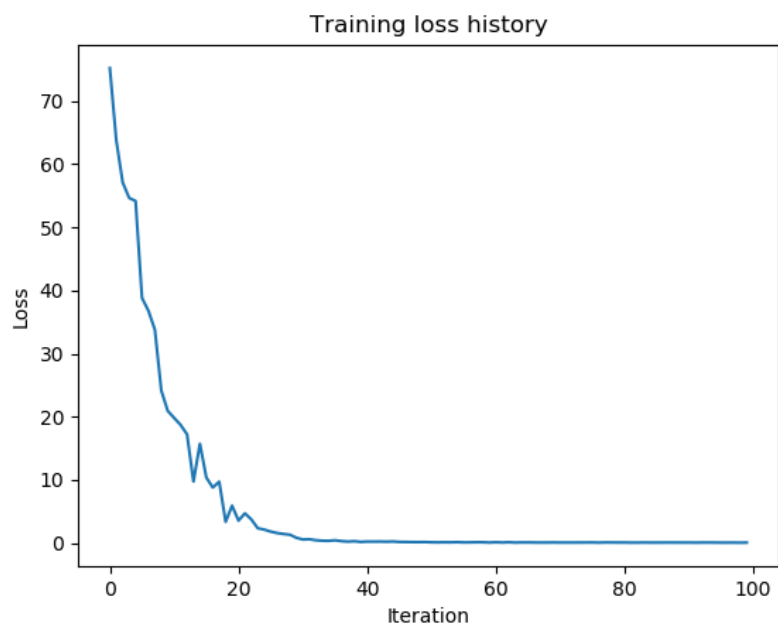
(Iteration 1 / 100) loss: 75.200942
(Iteration 11 / 100) loss: 19.807634
(Iteration 21 / 100) loss: 3.560278
(Iteration 31 / 100) loss: 0.569345
(Iteration 41 / 100) loss: 0.246043
(Iteration 51 / 100) loss: 0.130600
(Iteration 61 / 100) loss: 0.135817
(Iteration 71 / 100) loss: 0.087732
(Iteration 81 / 100) loss: 0.095701
(Iteration 91 / 100) loss: 0.086511

val
a man in a man on a <UNK> <UNK> <UNK> <END>
GT:<START> a woman <UNK> a girl who is holding a suitcase <END>



val
eautiful brown teddy at a a on a <UNK> of some <UNK> the other side <EN
GT:<START> a brown teddy bear sits on a desk next to a computer <END>





5(a) Finetune

Best val Acc: 0.928105

5(b) Freeze

Freeze the parameters in pre-trained model and train the final fc layer
Performance of pre-trained model without finetuning

Best val Acc: 0.960784