Assignment - 6

Kethan Challa
230317.

(Q1)

① → logistic regression uses probabilities above, below thresholds to classify, but perceptron after refining from ~~loss funct~~ weight optimisation, directly gives output 0(0), 1 binary.

② → weights get updated in perceptron if there is misclassification. In logistic regression there is loss function, it is optimised by gradient descent.

Tweak :

1) Change the activation Function : to $\boxed{\text{sigmoid}}$

2) Use cross-entropy loss function and reduce the loss to update weights.

(Q2). Email → 2 chances of outputs → 1 neuron [0/1]

Digit classification → 10 possible → 10 neurons.

→ Act Fln, At the end, we need to update weights, sigmoid [0 to 1] output will be better and easy to train.

since differentiable, can use GD. to update

→ We can use $\boxed{\text{SoftMax}}$, which gives probability per every DIGIT

3) a) $\boxed{\text{input } X}$

$\Rightarrow$ (total no. of inputs in the batch) $\times 10 \Rightarrow n \times 10$

b) $W_h \Rightarrow 10 \times 50$

$b_h \Rightarrow 1 \times 50$ } $\rightarrow$ Result : $Z = XW_h + b_h$

c) $W_o \Rightarrow 50 \times 3$

$b_o \Rightarrow 3$

d) $\boxed{\text{Output matrix } Y}$

$\hookrightarrow (n \times 10) \times (10 \times 50) \times (50 \times 3)$

$\Rightarrow (n \times 3)$

e) $Y = (\text{ReLU}(\text{ReLU}(XW_h + b_h)W_o + b_o))$

Q4) $E(w) = -\sum_{i}^{N} \{ t_n \ln y_n + (1-t_n)\ln(1-y_n) \}$

$$\frac{\partial E(w)}{\partial a_k} = \frac{\partial E_k}{\partial y_k} \cdot \frac{\partial y_k}{\partial a_k}$$, where $y_k = \sigma(a_k)$

$\Rightarrow y_k = \frac{1}{1+e^{-a_k}}$

$\Rightarrow e^{-a_k} \cdot y_k + y_k = 1$

$\Rightarrow e^{-a_k} = \frac{1-y_k}{y_k}$

$\Rightarrow -a_k = \ln\left(\frac{1-y_k}{y_k}\right)$

$\Rightarrow a_k = \ln\left(\frac{y_k}{1-y_k}\right)$

1) $$\frac{\partial E_k}{\partial y_k} = \frac{-t_k}{y_k} - \frac{1-t_k}{1-y_k}$$

$$= \frac{y_k - t_k}{y_k(1-y_k)}$$

2) $$\frac{\partial y_k}{\partial a_k} = \frac{1}{\frac{\partial a_k}{\partial y_k}} = \frac{1}{\frac{1}{y_k} + \frac{1}{1-y_k}}$$

$$\frac{\partial y_k}{\partial a_k} = y_k(1-y_k)$$

LHS $\Rightarrow$ $$\frac{\partial E_k}{\partial a_k} = \frac{y_k - t_k}{y_k(1-y_k)} \cdot y_k(1-y_k) = y_k - t_k$$ // RHS.

5) Input $\Rightarrow 4 \times 4$
Filter $\Rightarrow 2 \times 2$.
Output $\Rightarrow 3 \times 3$.

$\rightarrow$ Stride = 1.

$\rightarrow$

$-4+0+0+36 = 32$	$-10+0+24+0 = 14$	$6+0+0-24 = -18$
$0+0-4-18 = -22$	$-12+0-12+0 = -24$	$0+0+0+12 = 12$
$2+0+20+0 = 22$	$6+0+0+0 = 6$	$0+0+0+18 = 18$

$= \begin{bmatrix} 32 & 14 & -18 \\ -22 & -24 & 12 \\ 22 & 6 & 18 \end{bmatrix}$

Q6) Stride = 2
~~Strided~~
Input-padded = $(0, x_1, x_2, x_3, x_4, 0)$
Filter = (w_1, w_2, w_3).

$\rightarrow$ 2 output Activations $y_1 = 0 \cdot w_1 + x_1 w_2 + x_2 w_3$
$y_2 = x_2 w_1 + x_3 w_2 + x_4 w_3$.

$\rightarrow$ we know
$y = Ax$

$x = \begin{bmatrix} 0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ 0 \end{bmatrix}$ $y = \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} \Rightarrow A = \begin{pmatrix} w_1 & w_2 & w_3 & 0 & 0 & 0 \\ 0 & 0 & w_1 & w_2 & w_3 & 0 \end{pmatrix}$

Transpose $\rightarrow$ input $y = (y_1, y_2)$
filter (w_1, w_2, w_3)

$z = A^T y \Rightarrow \begin{bmatrix} w_1 & 0 \\ w_2 & 0 \\ w_3 & w_1 \\ 0 & w_2 \\ 0 & w_3 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} w_1 y_1 \\ w_2 y_1 \\ w_3 y_1 + w_1 y_2 \\ w_2 y_2 \\ w_3 y_2 \\ 0 \end{bmatrix}$

Q7) $X \Rightarrow$ input $\Rightarrow$ $\begin{bmatrix} 1 & 1 & 2 & 1 \\ 2 & 1 & 2 & 0 \\ 3 & 2 & 4 & 0 \\ 1 & 1 & 2 & 1 \end{bmatrix}$

Encoder:

$W_1 = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ -1 & 0 & 1 & 0 \end{bmatrix}$ $\quad$ $W_2 = \begin{bmatrix} 1 & 0 & 1 \\ -1 & 1 & 0 \end{bmatrix}$

$b_1 = \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix}$ $\quad$ $b_2 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$

$\cancel{W_2 X + b_2} =$

$\Downarrow$

$W_1 X + b_1 = \begin{bmatrix} 2 & 2 & 4 & 2 \\ 5 & 3 & 6 & 1 \\ 1 & 0 & 1 & -1 \end{bmatrix}$

$\mathrm{ReLU}(W_1 X + b_1) = \begin{bmatrix} 2 & 2 & 4 & 2 \\ 5 & 3 & 6 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix}$

$\Downarrow$

P.

$\Rightarrow$ $W_2 P + b_2 = \begin{bmatrix} 3 & 2 & 5 & 2 \\ 3 & 1 & 2 & -1 \end{bmatrix}$

$\mathrm{ReLU}(W_2 P + b_2) = \begin{bmatrix} 3 & 2 & 5 & 2 \\ 3 & 1 & 2 & 0 \end{bmatrix}$

$\Downarrow$

Z

Decoder:

$W_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & -1 \end{bmatrix}$ $\quad$ $b_1 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$ $\quad \Rightarrow \quad W_1 Z + b_1 = \begin{bmatrix} 3 & 2 & 5 & 2 \\ 4 & 2 & 3 & 1 \\ 0 & 1 & 3 & 2 \end{bmatrix}$

$\Downarrow$

$W_2 = \begin{bmatrix} 1 & 0 & -1 \\ 1 & -1 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$ $\quad$ $b_2 = \begin{bmatrix} 0 \\ 0 \\ 1 \\ -3 \end{bmatrix}$ $\quad$ $\mathrm{ReLU}(W_1 Z + b_1) = \begin{bmatrix} 3 & 2 & 5 & 2 \\ 4 & 2 & 3 & 1 \\ 0 & 1 & 3 & 2 \end{bmatrix}$

$\Downarrow$

A

$Y = W_2 A + b_2 = \begin{bmatrix} 3 & -1 & 2 & 0 \\ -1 & 0 & 2 & 1 \\ 1 & 2 & 4 & 3 \\ 1 & 0 & 3 & 0 \end{bmatrix}$

$\Downarrow$

Y

$$y - y' = \begin{bmatrix} 3 & 1 & 2 & 0 \\ -1 & 0 & 2 & 1 \\ 1 & 2 & 4 & 3 \\ 1 & 0 & 3 & 0 \end{bmatrix} - \begin{bmatrix} 1 & 1 & 2 & 1 \\ 2 & 1 & 2 & 0 \\ 3 & 2 & 4 & 1 \\ 1 & 1 & 2 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 2 & 0 & 0 & -1 \\ -3 & -1 & 0 & 1 \\ -2 & 0 & 0 & 2 \\ 0 & -1 & 1 & -1 \end{bmatrix}$$

↳ Reconstruction loss.

MSE loss

↳ Gradients $\frac{\partial L}{\partial y} = (y - y') \times 2$

$$= \begin{bmatrix} 4 & 0 & 0 & -2 \\ -6 & -2 & 0 & 2 \\ -4 & 0 & 0 & 4 \\ 0 & -2 & 2 & -2 \end{bmatrix}$$