IFT6390 HW3 Theory Q1

1 a) $\text{sign}(x) = \mathbb{1}_{x>0} - \mathbb{1}_{x<0}$

1 b) $$g(x) = \begin{cases} x, & \text{when } x > 0 \\ 0, & \text{when } x \leq 0 \end{cases}$$

$$\nabla g(x) = \frac{d}{dx} g(x) = \begin{cases} 1, & \text{when } x > 0 \\ 0, & \text{when } x < 0 \end{cases}$$

Since the Heaviside step function is defined as:

$$H(x) = \begin{cases} 1, & \text{when } x > 0 \\ 0.5, & \text{when } x = 0 \\ 0, & \text{when } x < 0 \end{cases}$$

then $\nabla g(x) = H(x)$ for $x < 0$ and $x > 0$.

1 c)

$$g(x) = x\,H(x)$$

Alternatively,

$$g(x) = (x+1)^{H(x)} - 1$$

1 d) $$a(x) = \frac{1}{1 + e^{-kx}}$$

when k is large.

$$\lim_{k\to\infty} a(x) = \lim_{k\to\infty}\left(\frac{1}{1+e^{-kx}}\right)$$

when $x > 0$,

$$\lim_{k\to\infty}\left(\frac{1}{1+e^{-kx}}\right) = \lim_{k\to\infty}\left(\frac{1}{1+0}\right)$$

$$= 1$$

when $x = 0$,

$$\lim_{k\to\infty}\left(\frac{1}{1+e^{-kx}}\right) = \lim_{k\to\infty}\left(\frac{1}{1+1}\right)$$

$$= \frac{1}{2}$$

when $x < 0$,

$$\lim_{k\to\infty}\left(\frac{1}{1+e^{-kx}}\right) = \lim_{k\to\infty}\left(\frac{1}{1+\infty}\right)$$

$$= 0$$

Since at large k,

$$a(x) = \begin{cases} 1, & \text{when } x > 0 \\ 0.5, & \text{when } x = 0 \\ 0, & \text{when } x < 0 \end{cases}$$

then $H(x)$ can be approximated by $a(x)$ when k is large.

(e) $\sigma'(x) = \frac{d}{dx}\left(\frac{1}{1+e^{-x}}\right)$

$$= -\left(1+e^{-x}\right)^{-2}\left(-e^{-x}\right)$$

$$= \frac{1}{1+e^{-x}}\left(\frac{e^{-x}}{1+e^{-x}}\right)$$

$$= \frac{1}{1+e^{-x}}\left(\frac{1+e^{-x}-1}{1+e^{-x}}\right)$$

$$= \frac{1}{1+e^{-x}}\left(1-\frac{1}{1+e^{-x}}\right)$$

$$= \sigma(x)\left(1-\sigma(x)\right)$$

Let $x \in \mathbb{R}^n$, hence $\sigma(x): \mathbb{R}^n \to \mathbb{R}^n$

$$J_{\sigma(x)} = \begin{bmatrix} \frac{\partial \sigma}{\partial x_1}, & \frac{\partial \sigma}{\partial x_2}, & \cdots & \frac{\partial \sigma}{\partial x_n} \end{bmatrix}$$

$$= \begin{bmatrix} \frac{\partial \sigma(x_1)}{\partial x_1} & \frac{\partial \sigma(x_1)}{\partial x_2} & \cdots & \frac{\partial \sigma(x_1)}{\partial x_n} \\ \frac{\partial \sigma(x_2)}{\partial x_1} & \frac{\partial \sigma(x_2)}{\partial x_2} & \cdots & \frac{\partial \sigma(x_2)}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial \sigma(x_n)}{\partial x_1} & \frac{\partial \sigma(x_n)}{\partial x_2} & \cdots & \frac{\partial \sigma(x_n)}{\partial x_n} \end{bmatrix} \in \mathbb{R}^{n \times n}$$

$$= \begin{bmatrix} \sigma(x_1)(1-\sigma(x_1)) & 0 & \cdots & 0 \\ 0 & \sigma(x_2)(1-\sigma(x_2)) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma(x_n)(1-\sigma(x_n)) \end{bmatrix}$$

$$\Rightarrow \left(J_{\sigma(x)}\right)_{i,j} = \sigma(x_i)(1-\sigma(x_i))\,\delta_{i=j}, \text{ where}$$

$i \in \{1, 2, \cdots, n\}$

$j \in \{1, 2, \cdots, n\}$

f) $\ln \sigma(x) = -\text{softplus}(-x)$

LHS $= \ln\left(\frac{1}{1+\exp(-x)}\right) \Rightarrow -\ln(1+\exp(-x))$

$\Rightarrow -\ln(\text{softplus}(-x)) = \text{RHS}$

g) $\text{softplus}(x) - \text{softplus}(-x) = x$

$$\text{LHS} = \ln(1+\exp(x)) - \ln(1+\exp(-x))$$
$$= \ln(1+\exp(x)) - \ln\left(1+\frac{1}{\exp(x)}\right)$$
$$= \ln\left(1+\exp(x)\right) - \ln\left(\frac{1+\exp(x)}{\exp(x)}\right)$$
$$= \ln\left(\frac{(1+\exp(x))}{(1+\exp(x))} \times \exp(x)\right)$$
$$= \ln(\exp(x))$$
$$= x = \text{RHS}$$

h) $S(x+c) = S(x)$

LHS $= S(x+c)_i = \frac{\exp(x_i+c)}{\sum_j \exp(x_j+c)} = \frac{\exp(x_i)\exp(c)}{\exp(c)\sum_j \exp(x_j)} = \frac{\exp(x_i)}{\sum_j \exp(x_j)}$

$= S(x) = \text{RHS}$

i) $S(xc)$

LHS $= S(xc)_i = \frac{\exp(x_i c)}{\sum_j \exp(x_j c)} = \frac{\exp(x_i)^c}{\sum_j \exp(x_j)^c}$

if $c=0 \Rightarrow S(xc) = \frac{1}{\sum 1} = \frac{1}{K}$, where K = no. of classes

$c=1 \Rightarrow S(xc) = S(x)$

As c increases, $S(xc)$ gets more skewed

j) $\frac{\partial S(x)_i}{\partial x_j} = S(x_i)(\delta_{i=j} - S(x)_j)$

$S(x_i) = \frac{\exp(x_i)}{\sum \exp(x_j)} \Rightarrow \partial S(x_j) = \exp(x_i) \times \frac{-1}{(\sum \exp(x_j))^2} \times (\exp(x_j))$

$= \frac{\exp(x_i)}{\sum \exp(x_j)} \times - \frac{\exp(x_j)}{\sum \exp(x_j)}$

$= S(x)_i \times -S(x)_j$ if $j \neq i$

or $= S(x)_i (1 - S(x)_i)$ if $j = i$

$= S(x)_i (\delta_{i=j} - S(x)_j)$ - RHS

k) Let Jacobian of softmax = J

$\Rightarrow J_{ij} = \frac{\partial S(x)_i}{\partial x_j}$ if $i = j \Rightarrow J_{ii} = \frac{\partial S(x)_i}{\partial x_j} = S(x)_i(-S(x)_i)$
$= S(x)(-S(x))^T$

$i \neq j \Rightarrow J_{ij} = \frac{\partial S(x)_i}{\partial x_j} = S(x_i)(1 - S(x_j))$
$= S(x)(1 - S(x))^T$

$\therefore J = S(x)(1 - S(x))^T - \text{diag}(S(x))$

l) $\nabla_u \log S(x(u)) = \nabla_u x(u)_i - E_j[\nabla_u x(u)_j]$

$\log S(x(u))_i = \log(\exp(x(u)_i)) - \log(\sum_j \exp(x(u)_j))$
$= x(u)_i - \log(\sum_j \exp(x(u)_j))$

$\nabla_u \log S(x(u))_i = \nabla_u x(u)_i - \nabla_u \log(\sum_j \exp(x(u)_j))$
$= \nabla_u x(u)_i - \frac{1}{\sum_j \exp(x(u_j))} \times \sum_j \exp(x(u_j)) \nabla_u x(u)_j$ — (1)

$E_j[\nabla_u x(u)_j] = \sum_j \nabla_u x(u)_j S(x(u))_j = \sum \nabla_u x(u)_j \frac{\exp(x(u)_j)}{\sum \exp(x(u)_j)}$ — (2)

$\Rightarrow \nabla_u \log S(x(u))_i = \nabla_u x(u)_i - \frac{\sum \nabla_u x(u)_j \exp(x(u)_j)}{\sum \exp(x(u)_j)}$

$= \nabla_u x(u)_i - E_j[\nabla_u x(u)_j]$

m) $L(x,c) = \sum_{i=1}^{K} -c_i \log y_i$ where $y = S(x)$

Gradient of cross-entropy loss,

$$\nabla_u L(x,c) = \sum_{i=1}^{K} -c_i \nabla_u \log S(x|u)_i$$

$$= \sum -c_i (\nabla_u x(u)_i - E_j [\nabla_u x(u)_j])$$

$$= -\sum_{i=1}^{K} c_i \nabla_u x(u)_i + E_j [\nabla_u x(u)_j] \sum_{i=1}^{K} c_i$$

Since c is a one-hot vector, $\sum c_i = 1$

$$\Rightarrow \nabla_u L(x,c) = \sum c_i \nabla_u x(u)_i + E_j [\nabla_u x(u)_j] \times 1$$

Q2

a) $W^{(1)} \in R^{d_h \times d}$

Dimension of $b^{(1)} = d_h \times 1$

Formula of $h^a = W^{(1)} x + b^{(1)}$

where $x \in R^d$

$$h_j^a = \sum_{k=1}^{d} w_{jk}^{(1)} x_k + b_j$$

$h^s = softplus(h^a)$ where $\begin{cases} h^s \in R^{d_h} \\ h^a \in R^{d_h} \end{cases}$

$h_j^s = \ln(1 + \exp(h_j^a))$

$h^s = \left[\ln(1+\exp(h_1^a)) \quad \ln(1+\exp(h_2^a)) \quad \ldots \quad \ln(1+\exp(h_{dh}^a))\right]^T$

b) Dimensions of $W^{(2)} = m \times d_h$

$b^{(2)} = m \times 1$

The pre-activation output vector denoted by o^a is defined as:

$$o^A = W^{(2)} h^s + b^{(2)} \quad ; \quad o^A \in R^m$$

Activation function $= \phi(o^A) = \text{softmax}(o^A)$

$$o^s = \phi(o^A) = \text{softmax}(W^{(2)} h^s + b^{(2)})$$

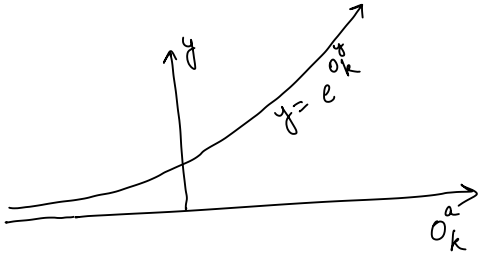
where, $o^s \in R^m$

$$o_k^a = \sum_{j=1}^{d_h} w_{kj}^{(2)} h_j^s + b_k^{(2)}$$

$$o_k^s = \frac{e^{o_k^a}}{\sum_{l=1}^{m} e^{o_l^a}}$$

c) $O^s = \text{softmax}(O^a)$; $O^s \in R^m$

$$O_k^s = \frac{e^{O_k^a}}{\sum_{j=1}^{m} e^{O_j^a}} \quad \forall \; k \in \{1, 2 \dots m\}$$



$\forall \; O_k^a \in R, \quad e^{O_k^a} > 0$

therefore, $\sum_{j=1}^{m} e^{O_j^a} > 0$

Since both numerator & denominator term are > 0

hence $\forall \; k \in \{1, 2 \dots m\} \quad O_k^s > 0$

$$\sum_{k=1}^{m} O_k^s = \sum_{k=1}^{m} \left(\frac{e^{O_k^a}}{\sum_{l=1}^{m} e^{O_l^a}} \right)$$

$$= \frac{1}{\sum_{l=1}^{m} e^{O_l^a}} \left(\sum_{k=1}^{m} e^{O_k^a} \right) = \frac{\sum_{k=1}^{m} e^{O_k^a}}{\sum_{l=1}^{m} e^{O_l^a}} = 1$$

Hence the sum of the vector $O^s = 1$

Since each element O_k^s represents probability values, their sum must be equal to 1 and should be individually positive.

d) A binary classifier with softmax activation would consist of 2 output neurons. such that,

$$O_1^s = P(y=0) = \frac{e^{O_1^a}}{e^{O_1^a} + e^{O_2^a}} = \frac{1}{1 + e^{(O_2^a - O_1^a)}}$$

$$O_2^s = P(y=1) = \frac{e^{O_2^a}}{e^{O_1^a} + e^{O_2^a}} = \frac{1}{1 + e^{(O_1^a - O_2^a)}}$$

Both probability values are of the form

$$O_k^s = \frac{1}{1 + e^{-z}}$$

which is same as sigmoid activation function.

and, $O_2^s = 1 - O_1^s$

Thus it can be concluded:

- Softmax activation is same as sigmoid for binary classification.
- Softmax is an extension of sigmoid for multi class classification problems.

e) $L_{mSE}(\sigma(o^A), y) = (\sigma(o^a) - y)^2$

where $L_{mSE} \in R$, $y \in \{0, 1\}$

For binary classification, $o^a \in R$

therefore $\frac{\partial L_{mSE}}{\partial o^a} \in R$

$$\frac{\partial L_{mSE}}{\partial o^a} = \frac{\partial (\sigma(o^a) - y)^2}{\partial o^a}$$

$$= 2(\sigma(o^a) - y) \frac{\partial \sigma(o^a)}{\partial o^a}$$

where,

$$\frac{\partial \sigma(o^a)}{\partial o^a} = \sigma(o^a)(1 - \sigma(o^a))$$

thus,

$$\boxed{\frac{\partial L_{mSE}}{\partial o^a} = 2(\sigma(o^a) - y)\sigma(o^a)(1 - \sigma(o^a))}$$

f)

$$L_{CE}(\sigma(o^a), y) = -y \log(\sigma(o^a)) - (1-y) \log(1 - \sigma(o^a))$$

where $\sigma(o^a) \in R$, $y \in \{0, 1\}$, $L_{CE} \in R$

thus $\frac{\partial L_{CE}}{\partial o^a} \in R$

$$\frac{\partial L_{CE}}{\partial o^a} = \frac{-y}{\sigma(o^a)} \cdot \frac{\partial \sigma(o^a)}{\partial o^a} - \frac{(1-y)}{1 - \sigma(o^a)} \cdot \frac{\partial (1 - \sigma(o^a))}{\partial o^a}$$

since $\frac{\partial \sigma(o^a)}{\partial o^a} = \sigma(o^a)(1 - \sigma(o^a))$

Plugging it into above expression we get,

$$\frac{\partial L_{CE}}{\partial o^a} = -y(1 - \sigma(o^a)) + (1-y)\sigma(o^a)$$

$$\boxed{\frac{\partial L_{CE}}{\partial o^a} = \sigma(o^a) - y} \qquad \frac{\partial L_{CE}}{\partial o^a} \in R$$

g) $L_{CE} = -y \log \sigma(o^A) - (1-y) \log(1-\sigma(o^A))$

$$\frac{\partial L_{CE}}{\partial o^a} = \sigma(o^a) - y$$

$$L_{MSE} = (\sigma(o^A) - y)^2 \qquad \frac{\partial L_{MSE}}{\partial o^a} = 2(\sigma(o^a) - y)\,\sigma(o^a)(1-\sigma(o^a))$$

Why cross entropy is better loss function than MSE?

1) Loss penalties are higher in CE for misclassification as compared to MSE. We can see this with an example:

Let $\sigma(o^A) = [0.1, 0.9]$

$y = [1, 0]$

$L_{CE} = -\log(0.1) = 1$ ✓

$L_{MSE} = (0.1-1)^2 = 0.81$

A larger value of loss ensures that the gradients computed are not squashed to zero particularly for deep networks. We can see this below.

$$\frac{\partial L_{CE}}{\partial o^a} = \sigma(o^a) - y = -0.8$$

$$\frac{\partial L_{MSE}}{\partial o^a} = 2(\sigma(o^a)-y)\,\sigma(o^a)(1-\sigma(o^a)) = -0.162$$

gradients get squashed in MSE loss as compared to CE

2) Convex objective. → $\frac{\partial^2 L_{CE}}{\partial^2 o^a} = \sigma(o^a)(1-\sigma(o^a)) > 0$

$\forall\, o^a \in R$

Convex objectives are easier to minimize thus its preferred loss function over MSE

h) $L(x, y) = -\log o^s_y(x) \quad ; x \in R^d \quad , y \in R$

For softmax activation, o^s_y is defined as:

$$o^s_y = \frac{e^{o^a_y}}{\sum_{k=1}^{m} e^{o^a_k}}$$

Thus we get,

$$L(x, y) = -\log\left(\frac{e^{o^a_y}}{\sum_{k=1}^{m} e^{o^a_k}}\right)$$

$$= -\log\left(e^{o^a_y}\right) + \log\left(\sum_{k=1}^{m} e^{o^a_k}\right)$$

$$\boxed{L(x, y) = -o^a_y + \log \sum_{k=1}^{m} e^{o^a_k}}$$

i) $\hat{R} = \frac{1}{n} \sum_{i=1}^{n} L(x^i, y^i)$

Set of Parameters θ associated are:

$W^{(1)}$ Dimension = $d_h \times d$

$b^{(1)}$ Dimension = $d_h \times 1$

$W^{(2)}$ Dimension = $m \times d_h$

$b^{(2)}$ Dimension = $m \times 1$

Total scalar Parameter $n_\theta = d_h(d+1) + m(d_h+1)$

$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}}\, \hat{R}$$

$$\boxed{\hat{\theta} = \underset{\theta}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^{n} L(x^i, y^i)}$$

j) $$\theta_{t+1} = \theta_t - \alpha \frac{\partial \hat{R}}{\partial \theta}$$

$$\theta_{t+1} = \theta_t - \frac{\alpha}{n} \sum_{i=1}^{n} \frac{\partial L(x^i, y^i)}{\partial \theta}$$

n = size of training dataset

α = learning rate ; $\alpha > 0$

k) $$\hat{R}_{reg}(\theta) = \hat{R}(\theta) + \lambda \|\theta\|_2^2$$

$$\frac{\partial \hat{R}_{reg}(\theta)}{\partial \theta} = \frac{\partial \hat{R}(\theta)}{\partial \theta} + 2\lambda \|\theta\|$$

Plugging this back in gradient descent equation

$$\theta_{t+1} = \theta_t - \alpha \left[\frac{1}{n} \sum_{i=1}^{n} \frac{\partial L(x^i, y^i)}{\partial \theta} + 2\lambda \|\theta\| \right]$$

i) $L(\mathbf{x}, y) = -\log o_y^s$

From the expression of loss derived in part-(h), we get

$$L(x, y) = -o_y^a + \log \sum_{k=1}^{m} e^{o_k^a}$$

$$\frac{\partial L}{\partial o^a} \in R^m$$

Computing partial derivative wrt any 'k' we get

$$\frac{\partial L}{\partial o_{k=y}^a} = -1 + \frac{1}{\sum_{j=1}^{m} e^{o_j^a}} \left(0 + \dots e^{o_y^a} + \dots 0\right)$$

$$\frac{\partial L}{\partial o_{k=y}^a} = \frac{e^{o_y^a}}{\sum_{j=1}^{m} e^{o_j^a}} - 1 = o_y^s - 1$$

for $k \neq y$, $\frac{\partial o_y^a}{\partial o_{k \neq y}^a} = 0$

hence,

$$\frac{\partial L}{\partial o^a_{k \neq y}} = 0 + \frac{1}{\sum_{j=1}^{m} e^{o^a_j}} \left(e^{o^a_k} \right)$$

$$= o^s_{k \neq y}$$

therefore, $$\frac{\partial L}{\partial o^a_k} = \begin{cases} o^s_k - 1 & ; \ k = y \\ o^s_k & ; \ k \neq y \end{cases}$$

In vector form we can thus write

$$\frac{\partial L}{\partial o^A} = o^s - \text{onehot}_m(y)$$

where $\text{onehot}_m(y) =$ $\begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix}$ $\leftarrow y^{th}$ index

$$\frac{\partial L}{\partial o^A} \in R^m \quad ; \quad o^s \in R^m$$

m)
$$\frac{\partial L}{\partial w_{kj}^{(2)}} = \frac{\partial L}{\partial o_k^a} \frac{\partial o_k^a}{\partial w_{kj}^{(2)}}$$

From part-(l) we know that

$$\frac{\partial L}{\partial o_k^a} = \begin{cases} o_k^s - 1 & ; \ k = y \\ o_k^s & ; \ k \neq y \end{cases}$$

$$o^a = w^{(2)} h^s + b^{(2)} \quad ; \quad w^{(2)} \in R^{m \times d_h}$$
$$b^{(2)} \in R^m$$
$$h^s \in R^{d_h}$$

$$o_k^a = \sum_{j=1}^{d_h} w_{kj}^{(2)} h_j^s + b_k$$

thus, $\frac{\partial o_k^a}{\partial w_{kj}^{(2)}} = h_j^s$

$$\frac{\partial o_k^a}{\partial b_k} = 1$$

substituting the values derived we get

$$\frac{\partial L}{\partial w_{kj}^{(2)}} = \begin{cases} (o_k^s - 1)\, h_j^s & ; \quad k = y \\ o_k^s\, h_j^s & ; \quad k \neq y \end{cases}$$

$\in \; k \in \{1, 2 \dots m\} \;\; \& \;\; j \in \{1, 2 \dots d_h\}$

$$\frac{\partial L}{\partial b_k^{(2)}} = \begin{cases} (o_k^s - 1) & k = y \\ o_k^s & k \neq y \end{cases}$$

n) $$\frac{\partial L}{\partial W^{(2)}} = \frac{\partial L}{\partial o^a} \frac{\partial o^a}{\partial W^{(2)}}$$

$$\frac{\partial L}{\partial o^a} = o^s - \text{onehot}_m(y)$$

$L \in R \quad o^a \in R^m \Rightarrow \frac{\partial L}{\partial o^a} \in R^m$

$W^{(2)} \in R^{m \times d_h}, \quad o^a \in R^m$

$\frac{\partial o^a}{\partial W^{(2)}}$ is a <u>tensor</u> with dim $= m \times m \times d_h$

where $$\frac{\partial o^a}{\partial W^{(2)}} = \begin{bmatrix} \frac{\partial o_1^a}{\partial W^{(2)}} \\ \vdots \\ \frac{\partial o_m^a}{\partial W^{(2)}} \end{bmatrix}_{m \times 1}$$

for any $k \in \{1, 2 \ldots m\}$ $\frac{\partial o_k^a}{\partial W^{(2)}} \in R^{m \times d_h} = \begin{bmatrix} o^T \\ \vdots \\ h^{sT} \\ \vdots \\ o^T \end{bmatrix}_{m \times d_h} \leftarrow k^{th}$ index ; $o^T \in R^{d_h}$

where, $h^{s^T} = \underbrace{\begin{bmatrix} h_1^s & h_2^s & \cdots & h_{d_h}^s \end{bmatrix}^T}_{1 \times d_h}$ & $0^T = \underbrace{\begin{bmatrix} 0 & 0 & \cdots & 0 \end{bmatrix}^T}_{1 \times d_h}$

Combining we get

$\frac{\partial L}{\partial W^{(2)}}$ dimension $= m \times d_h$

$\frac{\partial L}{\partial o^a} \in R^m$

$\frac{\partial o^a}{\partial W^{(2)}}$ dimension $= m \times m \times d_h$

The vector-tensor product upon multiplication is equivalent to the below matrix product

$$\frac{\partial L}{\partial W^{(2)}} = \underbrace{\begin{bmatrix} o_1^s \\ \vdots \\ o_y^s - 1 \\ \vdots \\ o_m^s \end{bmatrix}}_{m \times 1} \underbrace{\begin{bmatrix} h_1^s & h_2^s & \cdots & h_{d_h}^s \end{bmatrix}}_{1 \times m}$$

$$\boxed{\frac{\partial L}{\partial W^{(2)}} = \underbrace{\left(o^s - \text{onehot}_m(y)\right)}_{m \times 1} \underbrace{h^{s^T}}_{1 \times m}}$$

$$\frac{\partial L}{\partial b^{2}} = \frac{\partial L}{\partial o^{q}} \frac{\partial o^{q}}{\partial b^{2}}$$

$\frac{\partial L}{\partial o^{q}}$ dimension $= 1 \times m$, $b^{(2)} \in R^{m}$

$o^{q} \in R^{m} \Rightarrow \frac{\partial o^{q}}{\partial b^{(2)}}$ dimension $= m \times m$

we know $\frac{\partial o_{k}^{q}}{\partial b_{k}^{2}} = 1$

therefore $\frac{\partial o^{q}}{\partial b^{(2)}} = I_{m \times m} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \cdots & 1 & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix}$

$$\boxed{\frac{\partial L}{\partial b^{2}} = \underbrace{\left(o^{s} - \text{onehot}_{m}(y)\right)^{T}}_{1 \times m} I_{m \times m}}$$

c) $$\frac{\partial L}{\partial h_j^s} = \sum_{k=1}^{m} \frac{\partial L}{\partial o_k^a} \frac{\partial o_k^a}{\partial h_j^s}$$

$$\frac{\partial L}{\partial o_k^a} = \begin{cases} o_k^s - 1 & ; \ k = y \\ o_k^s & ; \ k \neq y \end{cases}$$

$$o_k^a = \sum_{j=1}^{d_h} w_{kj}^2 h_j^s + b_k^{(2)}$$

$$\frac{\partial o_k^a}{\partial h_j} = w_{kj}^{(2)}$$

therefore, $$\boxed{\frac{\partial L}{\partial h_j^s} = \sum_{k=1}^{m} o_{k \neq y}^s w_{kj}^{(2)} + (o_y^s - 1) w_{yj}^{(2)}}$$

d) Dimensions

$$\frac{\partial L}{\partial h^s} = 1 \times d_h$$

$$\frac{\partial L}{\partial o^a} = 1 \times m$$

$$\frac{\partial o^a}{\partial h^s} = m \times d_h$$

$$\frac{\partial L}{\partial o^a} = o^s - \text{onehot}_m(y) \quad ; \quad o^s \in R^m \quad \text{onehot}_m(y) \in R^m$$

$$\frac{\partial o_k^a}{\partial h_j^s} = W_{kj}^{(2)} \quad \forall \quad k \in \{1, 2 \dots m\}, \; j \in \{1, 2 \dots d_h\}$$

$$\frac{\partial o^a}{\partial h^s} = W^{(2)} \quad ; \quad W^{(2)} \in R^{m \times d_h}$$

$$\boxed{\frac{\partial L}{\partial h^s} = \underbrace{\left(o^s - \text{onehot}_m(y)\right)^T}_{1 \times m} W^{(2)}_{m \times d_h}}$$

2)

$$\frac{\partial L}{\partial h_j^a} = \frac{\partial L}{\partial h_j^s} \frac{\partial h_j^s}{\partial h_j^a}$$

$$h_j^s = \ln\left(1 + \exp(h_j^a)\right)$$

$$\frac{\partial h_j^s}{\partial h_j^a} = \frac{1}{1 + \exp(h_j^a)} \cdot \exp(h_j^a)$$

In general $\forall\, i, j \in \{1, 2 \ldots d_h\}$

$$\frac{\partial h_i^s}{\partial h_j^a} = \begin{cases} \frac{1}{1 + \exp(-h_j^a)} & i = j \\ 0 & i \neq j \end{cases}$$

from part-(0) we derived earlier,

$$\frac{\partial L}{\partial h_j^s} = \sum_{k=1}^{m} O_{k \neq y}^s W_{kj}^{(2)} + \left(O_y^s - 1\right) W_{yj}^{(2)}$$

therefore we get

$$\boxed{\frac{\partial L}{\partial h_j^a} = \left(\sum_{k=1}^{m} O_{k \neq y}^s W_{kj}^{(2)} + \left(O_y^s - 1\right) W_{yj}^{(2)} \right) \frac{1}{1 + \exp(-h_j^a)}}$$

r) $\frac{\partial L}{\partial h^a} = \frac{\partial L}{\partial h^s} \frac{\partial h^s}{\partial h^a}$

$\frac{\partial L}{\partial h^s} = \underset{1 \times m}{\left(0^s - \text{onehot}_m(y)\right)^T} W^{(2)}_{m \times d_h}$ [from part-(b)]

Dimensions: $\frac{\partial L}{\partial h^s} = 1 \times d_h$

$W^{(2)} \rightarrow m \times d_h$

$$\frac{\partial h^s_i}{\partial h^a_j} = \begin{cases} \frac{1}{1 + \exp(-h^a_j)} & i = j \\ 0 & i \neq j \end{cases}$$

$\frac{\partial h^s}{\partial h^a} = \text{diag}(H)$

where $H = \left[\frac{1}{1+e^{-h^a_1}} \quad \frac{1}{1+e^{-h^a_2}} \quad \cdots \quad \frac{1}{1+e^{-h^a_{d_h}}}\right]^T_{d_h \times 1}$

Dimensions

$\frac{\partial h^s}{\partial h^a} = d_h \times d_h$

$h^a, h^s \in R^{d_h}$

$$\frac{\partial L}{\partial h^a} = \underbrace{\left(o^s - \text{onehot}_m(y)\right)^T}_{1 \times d_h} \underbrace{W^{(2)}}_{m \times d_h} \underbrace{\text{diag}(H)}_{d_h \times d_h}$$

$\frac{\partial L}{\partial h^a}$ dimension $= 1 \times d_h$

IFT6390 HW3 Theory Q3

3. a) $h = \left\lfloor \frac{i + 2p - k}{s} \right\rfloor + 1$,

where i = input height
p = padding,
k = kernel size,
s = stride

Let h_1, h_2 and h_3 be the height of the first, second and third layers respectively.

$$h_1 = \left\lfloor \frac{128 + 2 \times 3 - 8}{2} \right\rfloor + 1$$

$$= 64$$

Since input image is 128×128, and 32 kernels were convolved over it, the dimensions of the first layer is $32 \times 64 \times 64$.

Since the second layer is a 2×2 max-pooling with stride 2,

$$h_2 = \left\lfloor \frac{64}{2} \right\rfloor = 32.$$

$\Rightarrow$ The dimensions of second layer is 32×32 32.

Since third layer convolves 64 3×3 kernels with stride 1 and zero-padding of 1,

$$h_3 = \left\lfloor \frac{32 + 2 \times 1 - 3}{1} \right\rfloor + 1$$

$$= 32$$

$\Rightarrow$ The dimensions of third (last) layer is $64 \times 32 \times 32$.

3b). Number of parameters $= 64 \times 3 \times 3 \times 32$
$= 18432$

3c) Let x be $[1, 1, 4, 4, 4, 1, 1]$
k be $[1/4, 1/4, 1/4]$

Let x^{full} which is x with padding for full convolution.
be $[0, 0, 1, 1, 4, 4, 4, 1, 1, 0, 0]$

Full convolution:

$$(x^{full} * k)_i = \sum_p x^{full}_{i+p} k_p$$

$$(x^{full} * k) = [0.25, 0.5, 1.5, 2.25, 3, 2.25, 1.5, 0.5, 0.25]$$

Valid convolution:

Let x^{valid} be x with padding for valid convolution, which is simply x (no padding).

$$(x^{valid} * k)_i = \sum_p x^{valid}_{i+p} k_p$$

$$(x^{valid} * k) = [1.5, 2.25, 3, 2.25, 1.5]$$

Same convolution:

Let x^{same} be $[0, 1, 1, 4, 4, 4, 1, 1, 0]$

$$(x^{same} * k)_i = \sum_p x^{same}_{i+p} k_p$$

$$(x^{same} * k) = [0.5, 1.5, 2.25, 3, 2.25, 1.5, 0.5]$$

3d) The convolution operation effectively computes the average of the values in x within the window of the kernel. Since the kernel size is 3, every 3 consecutive values in x are averaged. It is an average pooling convolution.