Question 1. Answer

1.1 The goal is to predicting the value of a binary variable y, which is the case of Bernoulli distribution. therefore the output layer should have Sigmoid units as the output activation function.

1.2 It represents a probability between 0 and 1.

1.3 Overall loss function is the summation of the loss functions over all examples(one example in this question) :$L_{CE}(f(\boldsymbol{x}, \boldsymbol{\theta}), y) = -(ylog(g) + (1-y)log(1-g))$.

1.4 $\frac{\partial L_{CE}(f(\boldsymbol{x},\boldsymbol{\theta}),y)}{\partial a(\boldsymbol{x},\boldsymbol{\theta})} = \frac{\partial(-ylog(g)-(1-y)log(1-g)))}{\partial a} = \frac{\partial(-ylog(g)-(1-y)log(1-g)}{\partial g}\frac{\partial g}{\partial a} = (\frac{-y}{g} + \frac{1-y}{1-g})(g(1-g)) = g - y$

1.5 Overall loss function is the summation of the loss functions over all examples(one example in this question) : $L_{MSE}(f(\boldsymbol{x}, \boldsymbol{\theta}), y) = -(y - f(\boldsymbol{x}, \boldsymbol{\theta}))^2$

1.6 $\frac{\partial L_{MSE}(f(\boldsymbol{x},\boldsymbol{\theta}),y)}{\partial a(\boldsymbol{x},\boldsymbol{\theta})} = -\frac{\partial(y-f(\boldsymbol{x},\boldsymbol{\theta}))^2}{\partial a} = -\frac{\partial(y-g)^2}{\partial g}\frac{\partial g}{\partial a} = 2(y-g) * g * (1-g)$.

1.7 From the above results,the cross-entropy loss is more appropriate loss function for binary classification. The gradient calculated for the pre-activation layer is simpler than the one with MSE, which makes it easier to optimize. Another point is that the MSE loss funcion is not concave, the gradient is hard to resolve given its Hessian matrix is $\begin{pmatrix} 2 & -2 \\ -2 & 2 \end{pmatrix}$ which is non semi-positive definite.

Question 2. Answer

2.1 $S(\boldsymbol{x}+c)_i = \frac{e^{x_i+c}}{\sum_j e^{x_j+c}} = \frac{e^{x_i}e^c}{\sum_j e^{x_j}e^c} = \frac{e^c e^{x_i}}{e^c \sum_j e^{x_j}} = \frac{e^{x_i}}{\sum_j e^{x_j}} = S(\boldsymbol{x})_i$, where c is a scalar constant.

2.2 when $\boldsymbol{x}$ is a 2-dimensional vector, we can represent $\boldsymbol{x}$ as $(\boldsymbol{x}_1, \boldsymbol{x}_2)$. and in this case, we represent $\boldsymbol{z}$ as an affine transformation of $\boldsymbol{x}$, such that $\begin{pmatrix} z_0 \\ z_1 \end{pmatrix} = \begin{pmatrix} \beta_0^T \\ \beta_1^T \end{pmatrix} \boldsymbol{x}$. Let $\mathbf{C}$ represents the class categories as (C_0, C_1). $P(C_i|x) = S(z_i) = \frac{e^{z_i}}{e^{z_0}+e^{z_1}}$. $P(C_1|x) = S(z_1) = \frac{e^{z_1}}{e^{z_0}+e^{z_1}} = \frac{1}{e^{z_0-z_1}+1} = \frac{1}{e^{-z'}+1} = \sigma(z')$,where $z= -(z_0 - z_1)$. $S(z_0) = \frac{e^{z_0}}{e^{z_0}+e^{z_1}} = 1 - \frac{e^{z_1}}{e^{z_0}+e^{z_1}} = 1 - \frac{1}{e^{z_0-z_1}+1} = 1 - \frac{1}{e^{-z'}+1} = 1 - \sigma(z')$.Therefore, in the special case where j =2, we can find softmax predict one of the class as $\sigma(z)$ and the other $1 - \sigma(z)$.

2.3 when $\boldsymbol{x}$ is a K-dimensional vector, we can represent $\boldsymbol{x}$ as $(\boldsymbol{x}_1, \boldsymbol{x}_2 ... \boldsymbol{x}_K)$. and in this case, we represent $\boldsymbol{y}$ as an affine transformation of $\boldsymbol{x}$, such that

$$\begin{pmatrix} y_1 \\ y_2 \\ .... \\ y_k \end{pmatrix} = \begin{pmatrix} \beta_1^T \\ \beta_2^T \\ .... \\ \beta_k^T \end{pmatrix} \boldsymbol{x}$$

for $c \subseteq 1, 2, ..., k$,$P(y_i = c|x) = S(y_i) = \frac{e^{y_i}}{\sum_{i=1}^{K} e_i^y} = \frac{e^{\beta_c * X_i}}{e^{\sum_{i=1}^{K} e^{\beta_k * X_i}}}$. Since for all $y_i, \sum_{k=1}^{K} p(y_i = k) = 1$, hence, one of the $p(y_i = c)$ must be determined by the rest probabilities where $k \neq c$. So there are only K-1 β for $\boldsymbol{x}$.
According to what explained above and the translation-invariant property of softmax, we can substract a constant "a" from the $\boldsymbol{\beta}$. The original function can be represented as $S(y_i) = S(y_i - a) = \frac{e^{y_i - a}}{\sum_{i=1}^{K} y_i - a} = \frac{e^{(\beta_c - a) * X_i}}{e^{\sum_{i=1}^{K} e^{(\beta_k - a) * X_i}}}$.
Here, we let $a = -\beta_1$,so that we force the β_1 to be zero, which taking account the fact that only K-1 categories probability needed. Hence,for $\beta_i^{'} = \beta_i - \beta_1$, $S(y_i) = \frac{e^{(\beta_c^{'}) * X_i}}{e^{0 * X_i} + \sum_{i=2}^{K} e^{(\beta_k^{'}) * X_i}} = \frac{e^{y_i^{'}}}{e^0 + \sum_{i=2}^{K} e^{y_k^{'}}}$, where $\boldsymbol{y} = [0, y_2^{'}, y_3^{'} ... y_K^{'}]$. By change the notation, we get $\boldsymbol{y} = [0, y_1, y_2 .... y_{K-1}]$.

2.4 $J_{softmax}(\boldsymbol{x}) = \frac{\partial S_i}{\partial a_j} = \frac{\partial \frac{e^{a_i}}{\sum_{k=1}^{N} e_k^a}}{\partial a_j}$. Following the quotient rule $f(x) = \frac{g(x)}{h(x)}$,we have $f'(x) = \frac{g'(x)h(x) - h'(x)g(x)}{[h(x)]^2}$. Here, we set $g_i = e_i^a, h_i = \sum_{k=1}^{N} e^{a_k}$. For the derivative of h_i, it is e_j^a for any a_j ;for g_i, the derivative w.r.t a_j is e_{a_j} only when $i = j$, otherwise it is 0.

Replace $\sum_{k=1}^{N} e_k^a$ as $\sum$, when $i = j$,$\frac{\partial \frac{e^{a_i}}{\sum_{k=1}^{N} e_k^a}}{\partial a_j} = \frac{e^{a_i} \sum - e_j^a e_i^a}{\sum^2} = \frac{e^{a_i}}{\sum} \frac{\sum - e_i^a}{\sum} = S_i(1 - S_j)$.

for $i \neq j$,$\sum_{k=1}^{N} e_k^a$ as $\sum$, when $i = j$,$\frac{\partial \frac{e^{a_i}}{\sum_{k=1}^{N} e_k^a}}{\partial a_j} = \frac{0 - e_j^a e_i^a}{\sum^2} = -\frac{e^{a_j}}{\sum} \frac{\sum e_i^a}{\sum} = S_j S_i$. Put the two cases together, we have $S_i((i = j) - S_j) = Diag(\boldsymbol{p}) - \boldsymbol{p}\boldsymbol{p}^T$.

Question 3. Answer

Let $a = \sum_{i=1}^{D} \omega_{ji}^{(1)} x_i + \omega_{j0}^{(1)}$, then

$$tanh(a) = \frac{e^a - e^{-a}}{e^a + e^{-a}} = 1 - \frac{2}{e^{2a}+1} = 1 - 2\sigma(-2a) = 1 - 2(1-\sigma(2a)) = 2\sigma(2a) - 1$$

which means

$$\sigma(2a) = \frac{1}{2}\left(tanh(a) + 1\right)$$

$$\sigma(a) = \frac{1}{2}\Big(tanh(\frac{a}{2}) + 1\Big)$$

Therefore

$$y(x, \Theta, \sigma)_k = \sum_{j=1}^{M} \omega_{kj}^{(2)} \sigma\left(\sum_{i=1}^{D} \omega_{ji}^{(1)} x_i + \omega_{j0}^{(1)}\right) + \omega_{k0}^{(2)}$$

=

$$\sum_{j=1}^{M} \omega_{kj}^{(2)} \frac{1}{2}\left(tanh\left(\sum_{i=1}^{D} \frac{1}{2}\omega_{ji}^{(1)} x_i + \frac{1}{2}\omega_{j0}^{(1)}\right) + 1\right) + \omega_{k0}^{(2)}$$

=

$$\sum_{j=1}^{M} \frac{1}{2}\omega_{kj}^{(2)} tanh\left(\sum_{i=1}^{D} \frac{1}{2}\omega_{ji}^{(1)} x_i + \frac{1}{2}\omega_{j0}^{(1)}\right) + \left(\sum_{j=1}^{M} \frac{1}{2}\omega_{kj}^{(2)} + \omega_{k0}^{(2)}\right)$$

where

$$\Theta' = (\tilde{\omega}^{(1)}, \tilde{\omega}^{(2)}) = (\frac{1}{2}\omega^{(1)}, \frac{1}{2}\omega^{(2)})$$

$$\tilde{\omega}_{k0}^{(2)} = \frac{1}{2}\sum_{j=1}^{M} \omega^{(2)} + \omega_{k0}^{(2)}$$

Question 4. Answer

4.1

$$\text{Input} = 128 * 128 * 3$$

$$\text{First convolution layer output} = \frac{128 + 2 * 3 - 8}{2} + 1 = 64; \text{Output dim=64*64*32}$$

$$\text{Max pooling layer Output} = \frac{64 - 2}{2} + 1 = 32; \text{Output dim=32*32*32}$$

$$\text{Third convolution layer output} = \frac{32 + 2 * 1 - 3}{1} + 1 = 32; \text{Output dim=32*32*64}$$

The output of the last layer contains 32*32*64=65536 scalars.

4.2

$$3 * 3 * 32 * 64 = 18432$$

4.3
- kernel flip : $\left[2, 0, 1\right]$
- Valid convolution : $[1 * 2 + 2 * 0 + 3 * 1, 2 * 2 + 3 * 0 + 4 * 1] = [5, 8]$
- Full convolution :

$$\text{Input padding}[0, 0, 1, 2, 3, 4, 0, 0]$$

$$\text{Convolution}$$

$$[0{*}2{+}0{*}0{+}1{*}1, 0{*}2{+}1{*}0{+}2{*}1, 1{*}2{+}2{*}0{+}3{*}1, 2{*}2{+}3{*}0{+}4{*}1, 3{*}2{+}4{*}0{+}0{*}1, 4{*}2{+}0{*}0{+}0{*}1]$$

$=$

$$[1, 2, 5, 8, 6, 8]$$

- Same convolution :

$$\text{Input padding}[0, 1, 2, 3, 4, 0]$$

$$\text{Convolution}$$

$$[0 * 2 + 1 * 0 + 2 * 1, 1 * 2 + 2 * 0 + 3 * 1, 2 * 2 + 3 * 0 + 4 * 1, 3 * 2 + 4 * 0 + 0 * 1]$$

$=$

$$[2, 5, 8, 6]$$

Question 5. Answer

5.1

$$\boldsymbol{x} = \begin{pmatrix} x_1 \\ x_2 \\ .... \\ x_d \end{pmatrix}$$

After first valid convolution :

$$\boldsymbol{a}^1 = \begin{pmatrix} x_1 \\ x_2 \\ .... \\ x_{d-1} \\ x_d \end{pmatrix} * \begin{pmatrix} w_1^1, w_2^1, ...w_{k_1-1}^1, w_{k_1}^1 \end{pmatrix} = \begin{pmatrix} x_1 w_1^1 + x_2 w_2^1 + ...x_{1+k_1-1} w_{k_1}^1 \\ x_2 w_1^1 + x_3 w_2^1 + ...x_{2+k_1-1} w_{k_1}^1 \\ \cdots \\ \\ x_{n-1} w_1^1 + x_n w_2^1 + ...x_{n+k_1-2} w_{k_1}^1 \\ x_n w_1^1 + x_{n+1} w_2^1 + ...x_{n+k_1-1} w_{k_1}^1 \end{pmatrix}$$

After second valid convolution :

$$\boldsymbol{a}^2 = \begin{pmatrix} h_1^1 \\ h_2^1 \\ .... \\ h_{d-k_1}^1 \\ h_{d-k_1+1}^1 \end{pmatrix} * \begin{pmatrix} w_1^2, w_2^2, ...w_{k_2-1}^2, w_{k_2}^2 \end{pmatrix} = \begin{pmatrix} h_1^1 w_1^2 + h_2^1 w_2^2 + ...h_{1+k_2-1}^1 w_{k_2}^1 \\ h_2^1 w_1^2 + h_3^1 w_2^2 + ...h_{2+k_2-1}^1 w_{k_2}^2 \\ \cdots \\ \\ h_{n-1}^1 w_1^2 + h_n^1 w_2^2 + ...h_{n+k_2-2}^1 w_{k_2}^2 \\ h_n^1 w_1^2 + h_{n+1}^1 w_2^2 + ...h_{n+k_2-1}^1 w_{k_2}^2 \end{pmatrix}$$

In the first valid convolution, $n + k_1 - 1 = d$, therefore, $|\boldsymbol{a}^1| = d - k_1 + 1$. The following Relu activation does not change the output shape. So $|\boldsymbol{h}^1|$ has same $d - k_1 + 1$ dimension. For second convolution, $n + k_2 - 1 = d - k_1 + 1$,, we can get $|\boldsymbol{a}^2| = d - k_1 + 1 - k_2 + 1 = d - k_1 - k_2 + 2$.

5.2 $\frac{\partial a_i^2}{\partial h_n^1}$ can be represented as following array by fixing the row as $\partial a_i^2, i \in (1, 2, \cdots, d - k_1 - k_2 + 2)$ and each element as ∂h_n^1 in the array while increasing by 1 till $d - k_1 + 1$.

$$\begin{pmatrix} \frac{\partial(h_1^1 w_1^2 + h_2^1 w_2^2 + ...h_{1+k_2-1}^1 w_{k_2}^1)}{\partial h_1^1} & \cdots & \frac{\partial(h_1^1 w_1^2 + h_2^1 w_2^2 + ...h_{1+k_2-1}^1 w_{k_2}^1)}{\partial h_{d-k_1+1}^1} \\ \frac{\partial(h_2^1 w_1^2 + h_3^1 w_2^2 + ...h_{2+k_2-1}^1 w_{k_2}^2)}{\partial h_1^1} & \cdots & \frac{\partial(h_2^1 w_1^2 + h_3^1 w_2^2 + ...h_{2+k_2-1}^1 w_{k_2}^2)}{\partial h_{d-k_1+1}^1} \\ \cdots & & \\ \frac{\partial(h_{d-k_1}^1 w_1^2 + h_{d-k_1+1}^1 w_2^2 + ...h_{d-1}^1 w_{k_2}^2)}{\partial h_1^1} & \cdots & \frac{\partial(h_{d-k_1-k_2}^1 w_1^2 + h_{d-k_1-k_2+1}^1 w_2^2 + ...h_{d-k_1}^1 w_{k_2}^2)}{\partial h_{d-k_1+1}^1} \\ \frac{\partial(h_{d-k_1-k_2+1}^1 w_1^2 + h_{d-k_1-k_2+2}^1 w_2^2 + ...h_{d-k_1+1}^1 w_{k_2}^2)}{\partial h_1^1} & \cdots & \frac{\partial(h_{d-k_1-k_2+1}^1 w_1^2 + h_{d-k_1-k_2+2}^1 w_2^2 + ...h_{d-k_1+1}^1 w_{k_2}^2)}{\partial h_{d-k_1+1}^1} \end{pmatrix}$$

$$=$$

$$\begin{pmatrix} w_1^2 & w_2^2 & \cdots & w_{k_2}^2 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & w_1^2 & w_2^2 & \cdots & w_{k_2}^2 & 0 & 0 & 0 & \cdots & 0 & 0 \\ & & & & & \cdots & & & & & \\ 0 & 0 & 0 & 0 & 0 & \cdots & w_1^2 & w_2^2 & \cdots & w_{k_2}^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & \cdots & 0 & w_1^2 & w_2^2 & \cdots & w_{k_2}^2 \end{pmatrix}$$

Hence,for any n, if $n < i$ or $n > k_2 + i$, the respective derivative will be 0. For the rest n, since $\boldsymbol{a}_n^2 = (\boldsymbol{h}^1 * \boldsymbol{w}^2)_n = \sum_{j=1}^{k_2} h_{n+j-1}^1 w_j$. $\frac{\partial a_{n+j-1}^2}{\partial h_n^1} = w_j^2$. Therefore, $\frac{\partial a_i^2}{\partial h_n^1} = w_j^2$, where $i = n + j - 1$.

5.3 Given that

$$(\nabla_{\boldsymbol{h}^1} L)_n = \sum_{i=1}^{|\boldsymbol{a}^2|} (\nabla_{\boldsymbol{a}^2} L)_i \frac{\partial a_i^2}{\partial h_n^1}$$

$$(\nabla_{\boldsymbol{h}^1} L)_n$$

$=$

$$\sum_{i=1}^{|\boldsymbol{a}^2|} \frac{\partial L}{\partial a^2_{n+j-1}} \frac{\partial a^2_{n+j-1}}{\partial h^1_n}$$

$=$

$$\sum_{i=1}^{|\boldsymbol{a}^2|} \frac{\partial L}{\partial a^2_{n+j-1}} w^2_j$$

$=$

$$\begin{pmatrix} \frac{\partial L}{\partial a_1^2} \frac{\partial a_1^2}{\partial h_1^1} + \frac{\partial L}{\partial a_2^2} \frac{\partial a_2^2}{\partial h_1^1} .... \frac{\partial L}{\partial a^2_{|d-k_1-k_2+2|}} \frac{\partial a^2_{|d-k_1-k_2+2|}}{\partial h_1^1} \\ \frac{\partial L}{\partial a_1^2} \frac{\partial a_1^2}{\partial h_2^1} + \frac{\partial L}{\partial a_2^2} \frac{\partial a_2^2}{\partial h_2^1} .... \frac{\partial L}{\partial a^2_{d-k_1-k_2+2}} \frac{\partial a^2_{d-k_1-k_2+2}}{\partial h_2^1} \\ .... \\ \frac{\partial L}{\partial a_1^2} \frac{\partial a_1^2}{\partial h^1_{d-k_1+1}} + ... + \frac{\partial L}{\partial a^2_{d-k_1-k_2+1}} \frac{\partial a^2_{d-k_1-k_2+1}}{\partial h^1_{d-k_1+1}} + \frac{\partial L}{\partial a^2_d - k_1 - k_2 + 2} \frac{\partial a^2_{d-k_1-k_2+2}}{\partial h^1_{d-k_1+1}} \end{pmatrix}$$

By replacing the results we have from Q5.2,we have

$$(\nabla_{\boldsymbol{h}^1} L)_n = \begin{pmatrix} \frac{\partial L}{\partial a_1^2} w_1^2 + \frac{\partial L}{\partial a_2^2} 0 \cdots \frac{\partial L}{\partial a^2_{|a^2|}} 0 \\ \frac{\partial L}{\partial a_1^2} w_2^2 + \frac{\partial L}{\partial a_2^2} w_1^2 \cdots \frac{\partial L}{\partial a^2_{|a_2|}} 0 \\ \cdots \\ \frac{\partial L}{\partial a_1^2} 0 + \cdots + \frac{\partial L}{\partial a^2_{d-k_1-k_2+1}} 0 + \frac{\partial L}{\partial a^2_d - k_1 - k_2 + 2} w^2_{k_2} \end{pmatrix}$$

By full convolution definition,

$$(\nabla_{\boldsymbol{a}^2} L)_n \,\tilde{*}\, \text{flip}(\boldsymbol{w}^2)$$

$=$

$$\sum_{j=1}^{k} \frac{\partial L}{\partial a^2_{n+j-k}} w^2_{flip(j)}$$

$=$

$$\sum_{j=1}^{k} \frac{\partial L}{\partial a^2_{n+j-k_2}} w^2_{k_2-j+1}$$

Here, we replace n as the dimensions of $|h^1|$ given the nature of full convolution.

$$\begin{pmatrix} \frac{\partial L}{\partial a^2_{2-k_2}} w^2_{k_2} + \frac{\partial L}{\partial a^2_{3-k_2}} w^2_{k_2-1} \cdots \frac{\partial L}{\partial a^2_1} w^2_1 \\ \frac{\partial L}{\partial a^2_{3-k_2}} w^2_{k_2} + \frac{\partial L}{\partial a^2_{4-k_2}} w^2_{k_2-1} \cdots + \frac{\partial L}{\partial a^2_1} w^2_2 + \frac{\partial L}{\partial a^2_2} w^2_1 \\ \cdots \\ \frac{\partial L}{\partial a^2_{d-k_1-k_2+2}} w^2_{k_2} + \frac{\partial L}{\partial a^2_{d-k_1-k_2+3}} w^2_{k_2-1} + \cdots \frac{\partial L}{\partial a^2_{d-k_1-k_2+1+k_2}} w^2_1 \end{pmatrix}$$

Since $n + j - k_2 < 0$ or $n + j > d - k_1 + 1 + k_2$, the corresponding $|a_i|$ will be padded with 0. which means the derivative will be 0. Hence,

$$(\nabla_{\boldsymbol{a}^2} L)_n \tilde{*} \operatorname{flip}(\boldsymbol{w}^2) = \begin{pmatrix} 0 + 0 + \cdots \frac{\partial L}{\partial a_1^2} w_1^2 \\ 0 + 0 + \cdots \frac{\partial L}{\partial a_2^2} w_2^2 + \frac{\partial L}{\partial a_1^2} w_1^2 \\ \cdots \\ \frac{\partial L}{\partial a_{d-k_1-k_2+2}^2} w_{k_2}^2 + 0 + 0 \cdots \end{pmatrix}$$

We can see this matrix is equivalent as $(\nabla_{\boldsymbol{h}^1} L)_n$ above. Therefore, $\nabla_{\boldsymbol{h}^1} L = \nabla_{\boldsymbol{a}^2} L \tilde{*} \operatorname{flip}(\boldsymbol{w}^2)$ (full convolution).

5.4

$$\boldsymbol{a}^1 = \begin{pmatrix} x_1 \\ x_2 \\ .... \\ x_{d-1} \\ x_d \end{pmatrix} \tilde{*} \begin{pmatrix} w_1^1, w_2^1, \cdots, w_{k_1-1}^1, w_{k_1}^1 \end{pmatrix} = \begin{pmatrix} x_{2-k_1} w_1^1 + x_{3-k_1} w_2^1 + \cdots + x_1 w_{k_1}^1 \\ x_{3-k_1} w_1^1 + x_{4-k_1} w_2^1 + \cdots + x_2 w_{k_1}^1 \\ \cdots \\ \\ x_{n-k_1} w_1^1 + x_{n+1-k_1} w_2^1 + \cdots + x_{n-1} w_{k_1}^1 \\ x_{n+1-k_1} w_1^1 + x_{n+2-k_1} w_2^1 + \cdots + x_n w_{k_1}^1 \end{pmatrix}$$

$$\boldsymbol{a}^2 = \begin{pmatrix} h_1^1 \\ h_2^1 \\ \cdots \\ h_{d+k_1-2}^1 \\ h_{d+k_1-1}^1 \end{pmatrix} \tilde{*} \begin{pmatrix} w_1^2, w_2^2, \cdots, w_{k_2-1}^2, w_{k_2}^2 \end{pmatrix} = \begin{pmatrix} h_{2-k_2}^1 w_1^1 + h_{3-k_2}^1 w_2^1 + \cdots + h_1^1 w_{k_2}^1 \\ h_{3-k_2}^1 w_1^1 + h_{4-k_2}^1 w_2^1 + \cdots + h_2^1 w_{k_2}^1 \\ \cdots \\ \\ h_{n-k_2}^1 w_1^1 + h_{n+1-k_2}^1 w_2^1 + \cdots + h_{n-1}^1 w_{k_2}^1 \\ h_{n+1-k_2}^1 w_1^2 + h_{n+2-k_2}^1 w_2^2 + \cdots + h_n^1 w_{k_2}^1 \end{pmatrix}$$

For the first full convolution : since $n + 1 - k_1 = d$,therefore, $|\boldsymbol{a}^1| = d + k_1 - 1$; The relu activation does not change the output shape,So $|\boldsymbol{h}^1|$ has same $d + k_1 - 1$ dimensional.; For second full convolution, since $n + 1 - k_2 = d + k_1 - 1$(or $|\boldsymbol{a}^2| + 1 - k_2 = d + k_1 - 1$), we can get $|\boldsymbol{a}^2| = d + k_1 - 1 + k_2 - 1 = d + k_1 + k_2 - 2$.

5.5 $\frac{\partial a_i^2}{\partial h_n^1}$can be represented as following array by fixing the row as $\partial a_i^2, i \in (1, 2, \cdots, d + k_1 + k_2 - 2)$ and each element as ∂h_n^1 in the array while increasing by 1 till $d + k_1 - 1$).

$$\begin{pmatrix} \frac{\partial(h_{2-k_2}^1 w_1^1 + h_{3-k_2}^1 w_2^1 + \cdots + h_1^1 w_{k_2}^1)}{\partial h_1^1} & \frac{\partial(h_{2-k_2}^1 w_1^1 + h_{3-k_2}^1 w_2^1 + \cdots + h_1^1 w_{k_2}^1)}{\partial h_2^1} & \cdots & \frac{\partial((h_{2-k_2}^1 w_1^1 + h_{3-k_2}^1 w_2^1 + \cdots + h_1^1 w_{k_2}^1)}{\partial h_{d+k_1-1}^1} \\ \frac{\partial h_{3-k_2}^1 w_1^1 + h_{4-k_2}^1 w_2^1 + \cdots + h_2^1 w_{k_2}^1)}{\partial h_1^1} & \frac{\partial(h_{3-k_2}^1 w_1^1 + h_{4-k_2}^1 w_2^1 + \cdots + h_2^1 w_{k_2}^1)}{\partial h_2^1} & \cdots & \frac{\partial(h_{3-k_2}^1 w_1^1 + h_{4-k_2}^1 w_2^1 + \cdots + h_2^1 w_{k_2}^1)}{\partial h_{d+k_1-1}^1} \\ \\ & \cdots \\ \frac{\partial(h_{d+k_1-2}^1 w_1^1 + h_{d+k_1-1}^1 w_2^1 + \cdots + h_{d+k_1+k_2-3}^1 w_{k_2}^1)}{\partial h_1^1} & \cdots & \frac{\partial(h_{d+k_1-2}^1 w_1^1 + h_{d+k_1-1}^1 w_2^1 + \cdots + h_{d+k_1+k_2-3}^1 w_{k_2}^1)}{\partial h_{d+k_1-1}^1} \\ \frac{\partial(h_{d+k_1-1}^1 w_1^1 + h_{d+k_1}^1 w_2^1 + \cdots + h_{d+k_1+k_2-2}^1 w_{k_2}^1)}{\partial h_1^1} & \cdots & \frac{\partial(h_{d+k_1-1}^1 w_1^1 + h_{d+k_1}^1 w_2^1 + \cdots + h_{d+k_1+k_2-2}^1 w_{k_2}^1)}{\partial h_{d+k_1-1}^1} \end{pmatrix}$$

$$= \begin{pmatrix} w_{k_2}^2 & 0 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ w_{k_2-1}^2 & w_{k_2}^2 & 0 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & & & & & & & & \\ 0 & 0 & 0 & 0 & \cdots & \cdots & \cdots & w_1^2 & w_2^2 \\ 0 & 0 & 0 & 0 & \cdots & \cdots & \cdots & 0 & w_1^2 \end{pmatrix}$$

where the column name is $i = (1, 2, \cdots, d+k_1+k_2-2)$, and the row heading is $n = (1, 2, \cdots, d+k_1-1)$
Like what is shown in the matrix above, if $n < k_2-j$, $\frac{\partial a_i^2}{\partial h_{n+j-k_2}^1} = 0$. if $i > n+j-1$ or $n > i+k_2-1$, the respective derivative will be 0. For the rest , According to the full convolution definition, $a_n^2 = (\boldsymbol{h}^1 \tilde{*} \boldsymbol{w}^2)_n = \sum_{j=1}^{k_2} h_{n+j-k} w_j$. $\frac{\partial a_i^2}{\partial h_{n+j-k_2}^1} = w_j$; , where $i = n + j - k_2$

5.6

$$(\nabla_{\boldsymbol{h}^1} L)_n = \sum_{i=1}^{|\boldsymbol{a}^2|} (\nabla_{\boldsymbol{a}^2} L)_i \frac{\partial a_i^2}{\partial h_n^1}$$

By replacing the result from Q5.4, we get

$$\begin{pmatrix} \frac{\partial L}{\partial a_1^2} w_{k_2}^2 & +\frac{\partial L}{\partial a_2^2} w_{k_2-1}^2 & +\cdots & +\cdots & +\frac{\partial L}{\partial a_{|a^2|}^2} 0 \\ \frac{\partial L}{\partial a_1^2} 0 & +\frac{\partial L}{\partial a_2^2} w_{k_2}^2 & +\frac{\partial L}{\partial a_3^2} w_{k_2-1}^2 & +\cdots & +\frac{\partial L}{\partial a_{|a^2|}^2} 0 \\ \cdots & & & & \\ \frac{\partial L}{\partial a_1^2} 0 & +\frac{\partial L}{\partial a_2^2} 0 & +\cdots & +\frac{\partial L}{\partial a_{|a^2-1|}^2} w_1^2 & +\frac{\partial L}{\partial a_{|a^2|}^2} 0 \\ \frac{\partial L}{\partial a_1^2} 0 & +\frac{\partial L}{\partial a_2^2} 0 & +\cdots & +\frac{\partial L}{\partial a_{|a^2-1|}^2} w_2^2 & +\frac{\partial L}{\partial a_{|a^2|}^2} w_1^2 \end{pmatrix}$$

$$=$$

$$\begin{pmatrix} \frac{\partial L}{\partial a_1^2} w_{k_2}^2 & +\frac{\partial L}{\partial a_2^2} w_{k_2-1}^2 & +\cdots & +\frac{\partial L}{\partial a_{k_2}^2} w_1^2 \\ \frac{\partial L}{\partial a_2^2} w_{k_2-1}^2 & +\frac{\partial L}{\partial a_3^2} w_{k_2}^2 & +\cdots & +\frac{\partial L}{\partial a_{k_2+1}^2} w_1^2 \\ \cdots & & & \\ \frac{\partial L}{\partial a_{|a^2|-k_2}^2} w_{k_2}^2 & +\frac{\partial L}{\partial a_{|a^2|-k_2+1}^2} w_{k_2-1}^2 & +\cdots & +\frac{\partial L}{\partial a_{|a^2|}^2} w_1^2 \end{pmatrix}$$

by valid convolution definition

$$(\nabla_{\boldsymbol{a}^2} L)_n \tilde{*} \operatorname{flip}(\boldsymbol{w}^2)$$

$$=$$

$$\sum_{j=1}^{k} \frac{\partial L}{\partial a_{n+j-1}^2} w_{flip(j)}^2$$

$$=$$

$$\sum_{j=1}^{k} \frac{\partial L}{\partial a_{n+j-1}^2} w_{k_2-j+1}^2$$

$$=$$

$$\begin{pmatrix} \frac{\partial L}{\partial a_1^2} w_{k_2}^2 + \frac{\partial L}{\partial a_2^2} w_{k_2-1}^2 \cdots \frac{\partial L}{\partial a_{n+k_2-1}^2} w_1^2 \\ \frac{\partial L}{\partial a_2^2} w_{k_2}^2 + \frac{\partial L}{\partial a_3^2} w_{k_2-1}^2 \cdots \frac{\partial L}{\partial a_{n+k_2-1}^2} w_1^2 \\ \cdots \\ \frac{\partial L}{\partial a_{|a^2|-k_2}^2} w_{k_2}^2 + \dots \frac{\partial L}{\partial a_{|a^2|-1}^2} w_2^2 + \frac{\partial L}{\partial a_{|a^2|}^2} w_1^2 \end{pmatrix}$$

Therefore $\nabla_{\boldsymbol{h}^1} L = \nabla_{\boldsymbol{a}^2} L * \text{flip}(\boldsymbol{w}^2)$ (valid convolution)