5LSL0 Assignment 2: Nonlinear Models

Name	Student ID
Yiling Zhang	1756117
Liyuan Jiang	1704257

Lecturer: Rik Vullings, Ruud van Sloun
Nishith Chennakeshava, Hans van Gorp

Eindhoven, May 20, 2022

Learning nonlinear functions for regression and classification

Linear models

Q1

We know that $\mathbf{y} = \mathbf{X}\theta$ and $p(y_i;\theta) \sim \mathcal{N}(f(\mathbf{x}_i;\theta), 1)$ where

$$\boldsymbol{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix} \quad \boldsymbol{X} = \begin{bmatrix} [\mathbf{x}_1 & 1] \\ [\mathbf{x}_2 & 1] \\ \vdots & \vdots \\ [\mathbf{x}_m & 1] \end{bmatrix} \quad = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} & 1 \\ x_{21} & x_{22} & \cdots & x_{2n} & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} & 1 \end{bmatrix} \quad \theta = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \\ b \end{bmatrix}$$

The likelihood function can be expressed as $\mathcal{L}(\mathbf{y};\theta) = \prod_{i=1}^{m} \mathcal{N}(y_i | f(\mathbf{x}_i;\theta), 1)$, then the corresponding negative log-likelihood function is:

$$-l(\mathbf{y};\theta) = -\ln \mathcal{L}(\mathbf{y};\theta) = \frac{m}{2}\ln 2\pi + \sum_{i=1}^{m} \frac{(y_i - f(\mathbf{x}_i;\theta))^2}{2} = \frac{m}{2}\ln 2\pi + \frac{1}{2}(\boldsymbol{y} + \boldsymbol{X}\theta)^T(\boldsymbol{y} + \boldsymbol{X}\theta)$$

$\Rightarrow$

$$-\frac{\partial l(\mathbf{y};\theta)}{\partial \theta}|_{\theta=\hat{\theta}_{ML}} = \boldsymbol{X}^T(\boldsymbol{y} - \boldsymbol{X}\theta)|_{\theta=\hat{\theta}_{ML}} = 0 \tag{1}$$

The cost function is:

$$J(\boldsymbol{x}, y;\theta) = -l(\mathbf{y};\theta) = \frac{m}{2}\ln 2\pi + \frac{1}{2}(\boldsymbol{y} - \boldsymbol{X}\theta)^T(\boldsymbol{y} - \boldsymbol{X}\theta)$$

$\Rightarrow$

$$\frac{\partial J(\boldsymbol{x}, y;\theta)}{\partial \theta}|_{\theta=\theta^*} = \frac{\partial - l(\boldsymbol{y};\theta)}{\partial \theta}|_{\theta=\theta^*} = -\frac{\partial l(\boldsymbol{y};\theta)}{\partial \theta}|_{\theta=\theta^*} = \boldsymbol{X}^T(\boldsymbol{X}\theta - \boldsymbol{y})|_{\theta=\theta^*} = 0 \tag{2}$$

According to equation(1) and (2), we can get that the negative log-likelihood cost function will yield a maximum likelihood estimator (i.e. $\theta^* = \hat{\theta}_{ML}$) as the maximum likelihood is identical to the minimum negative log-likelihood, i.e. $\theta^* = \arg\min_{\theta}(J(\mathbf{y};\theta)) = \arg\min_{\theta}(-l(\mathbf{y};\theta)) = \arg\max_{\theta} l(\mathbf{y};\theta)) = \hat{\theta}_{ML}$.

Q2

Based on equation (2), we can get

$$\theta^* = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y} \tag{3}$$

Q3

Given the inputs $\boldsymbol{x} = \begin{bmatrix} 0 & 0 & 1 \\ 0.1 & 1 & 1 \\ 1 & 0.2 & 1 \\ 1 & 1 & 1 \end{bmatrix}$ and the outputs $f^*(\boldsymbol{x}) = \begin{bmatrix} 0 \\ 0.41 \\ 0.18 \\ 0.5 \end{bmatrix}$, we can get the optimal parameters based on equation (3):

$$\theta^* = (\boldsymbol{x}\boldsymbol{x}^{\boldsymbol{T}})^{-1}\boldsymbol{x}^{\boldsymbol{T}} f^*(\boldsymbol{x}) = \begin{bmatrix} 0.1000 \\ 0.4000 \\ 0.0000 \end{bmatrix} \iff \boldsymbol{w}^* = \begin{bmatrix} 0.1000 \\ 0.4000 \end{bmatrix}, \quad b = 0.0000$$

The resulting cost is $J(\boldsymbol{x}, f^*(\boldsymbol{x}); \theta^*) = \frac{m}{2}\ln 2\pi + \frac{1}{2}(f^*(\boldsymbol{x}) - \boldsymbol{x}\theta^*)^T(f^*(\boldsymbol{x}) - \boldsymbol{x}\theta^*) \approx 0.91894$. Mean squared error $(f^*(\boldsymbol{x}) - \boldsymbol{x}\theta^*)^T(f^*(\boldsymbol{x}) - \boldsymbol{x}\theta^*)/m \approx 1e^{-32}$. The process is well-described by our linear regression model since the MSE approximates 0.

Q4

Given $\boldsymbol{x} = \begin{bmatrix} 0 & 0 & 1 \\ 0.1 & 1 & 1 \\ 1 & 0.2 & 1 \\ 1 & 1 & 1 \end{bmatrix}$ and $\boldsymbol{y} = \begin{bmatrix} -0.0416 \\ 0.3610 \\ 0.1222 \\ 0.4733 \end{bmatrix}$, we can get the optimal parameters based on equation (3):

$$\theta^* = (\boldsymbol{x}\boldsymbol{x}^{\boldsymbol{T}})^{-1}\boldsymbol{x}^{\boldsymbol{T}} y = \begin{bmatrix} 0.1011 \\ 0.4107 \\ -0.050 \end{bmatrix} \iff \boldsymbol{w}^* = \begin{bmatrix} 0.1011 \\ 0.4107 \end{bmatrix}, \quad b = -0.050$$

The resulting cost with some arbitrary noisy sensor is $J(\boldsymbol{x}, f^*(\boldsymbol{x}); \theta^*) = \frac{1}{2}(f^*(\boldsymbol{x}) - \boldsymbol{x}\theta^*)^T(f^*(\boldsymbol{x}) - \boldsymbol{x}\theta^*) \approx 0.91915$, mean squared error $(f^*(\boldsymbol{x}) - \boldsymbol{x}\theta^*)^T(f^*(\boldsymbol{x}) - \boldsymbol{x}\theta^*)/m \approx 0.0001$, which is much larger than the cost obtained in Q3. In order to get better estimates, we can make use of more samples to suppress the influence of noise.

Q5

Given $p(\mathbf{y};\theta) \sim \mathcal{N}(X\theta, \Sigma)$ where the noise covariance matrix is:

$$\boldsymbol{\Lambda}^{-1} = \boldsymbol{\Sigma} = diag(\sigma_0, ..., \sigma_i, ..., \sigma_N)$$

we know

$$\mathcal{N}(\mathbf{X}\theta, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{\frac{N}{2}}} \frac{1}{|\Sigma|^{\frac{1}{2}}} e^{-\frac{1}{2}(\mathbf{y}-\mathbf{X}\theta)^T \boldsymbol{\Sigma}^{-1}(\mathbf{y}-\mathbf{X}\theta)}$$

then the NLL cost can be expressed as below:

$$J(\boldsymbol{x}, \boldsymbol{y}; \theta) = -\ln \mathcal{L}(\boldsymbol{y}; \theta)$$

$$= \frac{N}{2} \ln 2\pi + \frac{1}{2} \ln |\Sigma| + \frac{1}{2}(\mathbf{y} - \mathbf{X}\theta)^T \boldsymbol{\Sigma}^{-1}(\mathbf{y} - \mathbf{X}\theta)$$

where $|\Sigma|$ denotes the determinant of $\boldsymbol{\Sigma}$.
Let $\frac{\partial J}{\partial \theta} = 0$, then we can get

$$\mathbf{X}^T \boldsymbol{\Sigma}^{-1}(\mathbf{X}\theta - \mathbf{y}) = 0 \Rightarrow \theta^* = (\mathbf{X}^T \boldsymbol{\Lambda} \mathbf{X})^{-1} \mathbf{X}^T \boldsymbol{\Lambda} \mathbf{y}$$

The balance between the variances of the parameter estimates makes the weights relatively well-defined, which avoids some too large or small coefficients. If the weights are too large, the model would change dynamically, which would pick up too much local noise; on the contrary, if they are too small, the corresponding neurons are dead, which cannot learn features from data.

Q6

The inputs $\boldsymbol{x}_{XOR}$ and outputs $\boldsymbol{y}_{XOR}$ are defined as below:

$$\boldsymbol{x}_{XOR} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad \boldsymbol{y}_{XOR} = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

Based on equation (2), we can get

$$\boldsymbol{x}_{XOR}^T \boldsymbol{x}_{XOR} = \begin{bmatrix} 2 & 1 & 2 \\ 1 & 2 & 2 \\ 2 & 2 & 4 \end{bmatrix}, \quad \boldsymbol{x}_{XOR}^T \boldsymbol{y} = \begin{bmatrix} 1 \\ 1 \\ 2 \end{bmatrix} \Rightarrow \boldsymbol{\theta} = (\boldsymbol{x}_{XOR}^T \boldsymbol{x}_{XOR})^{-1} \boldsymbol{x}_{XOR}^T \boldsymbol{y} = \begin{bmatrix} 0 \\ 0 \\ 0.5 \end{bmatrix}$$

Thus, $\mathbf{w} = 0$, b $= 0.5$.

Nonlinear functions

Q7

- ReLU: $f(x) = max(0, x)$

$$\frac{\mathrm{d}f(x)}{\mathrm{d}x} = \begin{cases} 0, & \mathrm{x} \in (-\infty, 0) \\ 1, & \mathrm{x} \in (0, +\infty) \\ undefined, & \mathrm{x} = 0 \end{cases}$$

- Sigmoid: $f(x) = \sigma(x) = 1/(1 + exp(-x))$

$$\sigma(x)' = \sigma(x)^2 exp(-x) = \sigma(x)^2(1 - \sigma(x))/\sigma(x) = \sigma(x)(1 - \sigma(x)) \tag{4}$$

- Softmax: $f(\boldsymbol{x})_j = \frac{exp(x_j)}{\sum exp(x_i)}$, define $\sum exp(x_i)$ as Δ, then $\Delta'_{x_k} = exp(x_k)$

$$\begin{aligned} \frac{\partial f(\boldsymbol{x})_j}{\partial x_k} &= \frac{\delta(x_j - x_k)exp(x_j)\Delta - exp(x_j)\Delta'_{x_k}}{\Delta^2} \\ &= \begin{cases} f(\boldsymbol{x})_j(1 - f(\boldsymbol{x})_j)), & j = k \\ -f(\boldsymbol{x})_j f(\boldsymbol{x})_k, & j \neq k \end{cases} \end{aligned}$$

Q8

- ReLU: $f(x)' = 1$ when $x >> 0$
- Sigmoid: $\sigma(x)' \approx 0$ when $x >> 0$ and $\sigma(x)' = \sigma(x) - \sigma(x)^2 = \frac{1}{4} - (\sigma(x) - \frac{1}{2})^2 \in (0, \frac{1}{4}]$
- Softmax: since $f(\boldsymbol{x})_j \in (0, 1)$ only depends on the relative relation between different elements in the vector $\mathbf{x}$, the gradient is uncertain within the range of $(-1, 1)$ when we only know $\mathbf{x} >> \mathbf{0}$.

Shallow nonlinear models

Q9

As shown in Q6, a linear model $f(\mathbf{x}; \mathbf{w}, b) = \mathbf{w}^T\mathbf{x} + b$ is not able to represent the desired function XOR, while it can be solved by defining a mapping function that transform $\mathbf{x}$ nonlinearly into a space $\mathbf{h}$ before the next linear transformation as the nonlinearity enables a network to learn and to approximate arbitrary function mapping.

A linear mapping to a new space $\mathbf{h}$ does not suffice because the combination of two linear transformations is still linear, which is actually same as a linear model.

Q10

Given:

$$f(\mathbf{x}; \mathbf{W}^{(1)}, \mathbf{b}^{(1}, \mathbf{w}^{(2)}, b^{(2)}) = (\mathbf{w}^{(2)})^T max(0, \mathbf{W}^{(1)}\mathbf{x} + \mathbf{b}^{(1)}) + b^{(2)}$$

$$\mathbf{W}^{(1)} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}, \quad \mathbf{b}^{(1)} = \begin{bmatrix} 0 \\ -1 \end{bmatrix}, \quad \mathbf{w}^{(2)} = \begin{bmatrix} 1 \\ -2 \end{bmatrix}, \quad b^{(2)} = 0, \quad \mathbf{x} = \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix},$$

then the mapping into a latent space $\mathbf{h}$ should be:

$$\mathbf{h} = max(\mathbf{W}^{(1)}\mathbf{x} + \mathbf{b}^{(1)}) = max(0, \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix} + \begin{bmatrix} 0 \\ -1 \end{bmatrix}) = \begin{bmatrix} 0 & 2 \\ 0 & 1 \end{bmatrix}$$

$$f(\mathbf{h}) = (\mathbf{w}^{(2)})^T\mathbf{h} + b^{(2)} = \begin{bmatrix} 1 & -2 \end{bmatrix} \begin{bmatrix} h_1 \\ h_2 \end{bmatrix} = h_1 - 2h_2 = 0.5 \iff h_2 = \frac{h_1}{2} + 0.25 \qquad (5)$$

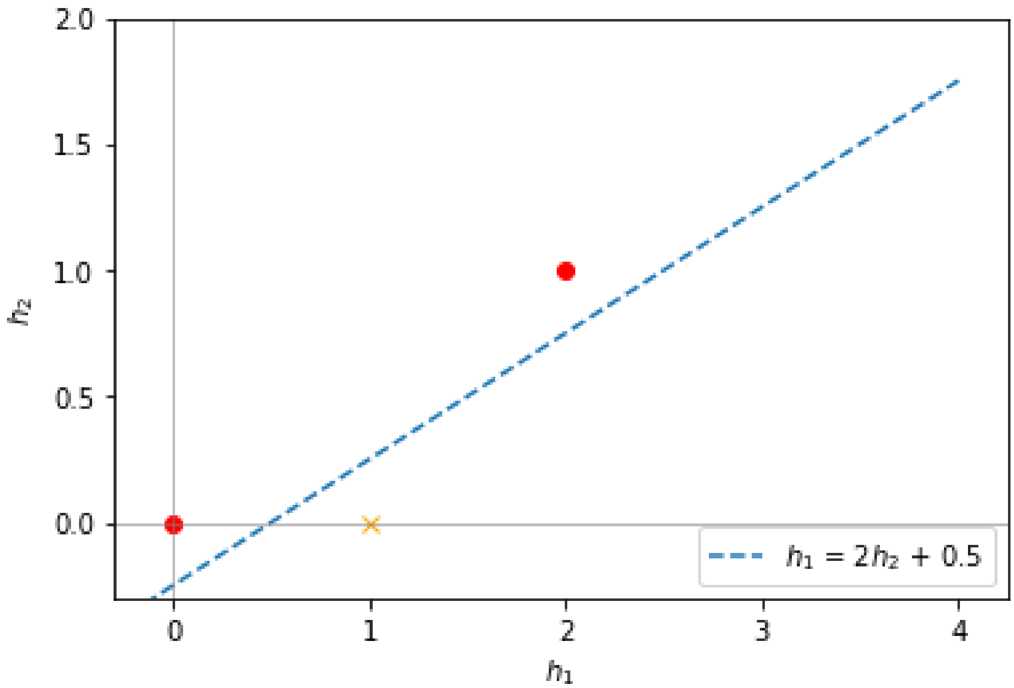

Figure 1: f($\mathbf{x}$) = 0.5 in the latent space $\mathbf{h}$

Q11

Substituting $\mathbf{h} = \begin{bmatrix} h_1 \\ h_2 \end{bmatrix} = max(0, \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ -1 \end{bmatrix}) = \begin{bmatrix} max(0, x_1 + x_2) \\ max(0, x_1 + x_2 - 1) \end{bmatrix}$ into the equation (5), we get the plot in the input space $\mathbf{x}$.

According to equation (5),

- if $x_1 + x_2 - 1 > 0$, then $h_1 = x_1 + x_2, h_2 = x_1 + x_2 - 1$,

$$x_1 + x_2 - 2(x_1 + x_2 - 1) = 0.5 \iff x_1 + x_2 = 1.5$$

- if $0 < x_1 + x_2 \leq 1$, then $h_1 = x_1 + x_2, h_2 = 0$:

$$x_1 + x_2 = 0.5$$

- if $x_1 + x_2 \leq 0$, then $h_1 = 0, h_2 = 0$, no activated.

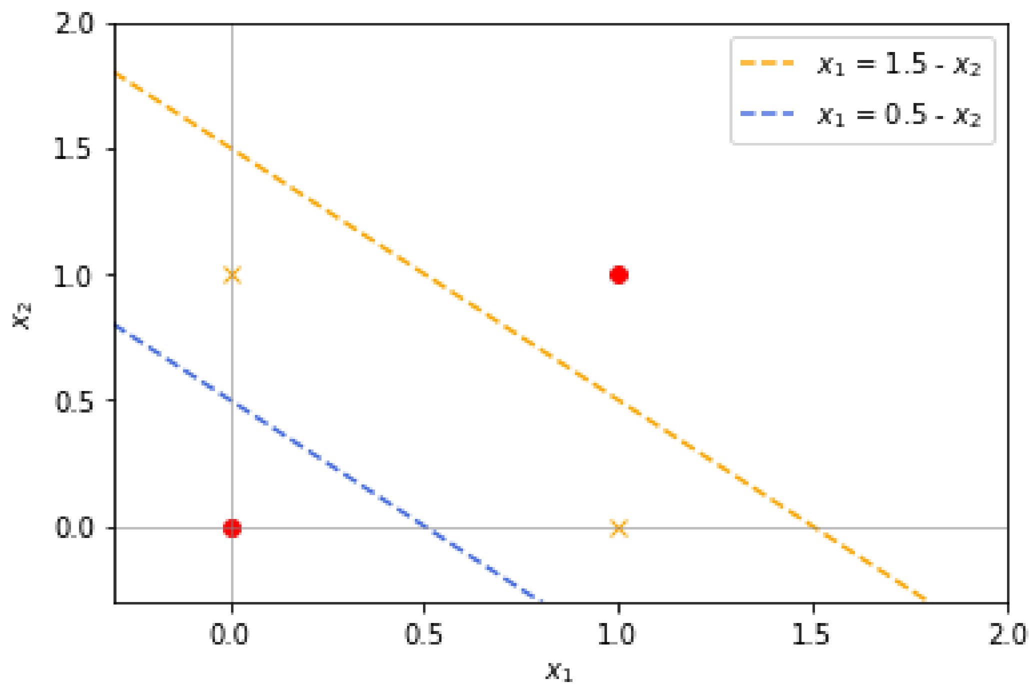

Figure 2: f($\mathbf{x}$)=0.5 in the input space $\mathbf{x}$

Binary classification with logistic regression

Q12

Softmax (i.e. $\frac{e^{x_i}}{\sum_{j=1}^{k} e^{x_j}}, i = 1, ..., k$) is useful for multi-class classification problems because it converts the output into the probability distribution that an input belongs to various categories. We can achieve multi-class classification by chose the class with the highest probability as the resulting label.

Q13

$$J = -\sum_{i=0}^{m-1} y^{(i)} \log(p^{(i)}) + (1-y^{(i)}) \log(1-p^{(i)})$$

$$\nabla_{p^{(k)}} J = -\frac{y^{(k)}}{p^{(k)}} + \frac{1-y^{(k)}}{1-p^{(k)}} \iff \nabla_p J = -\sum_{i=0}^{m-1} (\frac{y^{(i)}}{p^{(i)}} - \frac{1-y^{(i)}}{1-p^{(i)}})$$

Q14

Given:

$$p = \sigma(f(\mathbf{x};\mathbf{w})) = \sigma(\mathbf{w}^T\mathbf{x})$$

According to equation (4) in Q7:

$$\nabla_f p = \sigma(f(\mathbf{x};\mathbf{w}))(1-\sigma(f(\mathbf{x};\mathbf{w})))$$

Q15

$$f(\mathbf{x};\mathbf{w}) = \mathbf{w}^T\mathbf{x} \Rightarrow \nabla_{\mathbf{w}} f(\mathbf{x};\mathbf{w}) = \mathbf{x}$$

This step relates to Least Mean Square filter since both LMS and this step use the instantaneous estimate of gradient based on the input vector $\mathbf{x}$ as well.

Q16

$$\begin{aligned}
\nabla_w J = \frac{\partial J}{\partial w} &= \frac{\partial J}{\partial p}\frac{\partial p}{\partial f}\frac{\partial f}{\partial w} \\
&= \sum_{i=0}^{m-1} (-\frac{y^{(i)}}{p^{(i)}} + \frac{1-y^{(i)}}{1-p^{(i)}})\sigma(\mathbf{x}^{(i)};\mathbf{w})(1-\sigma(f(\mathbf{x}^{(i)};\mathbf{w}))\mathbf{x}^{(i)} \\
&= \sum_{i=0}^{m-1} (-\frac{y^{(i)}}{\sigma(\mathbf{w}^T\mathbf{x}^{(i)}} + \frac{1-y^{(i)}}{1-\sigma(\mathbf{w}^T\mathbf{x}^{(i)}})\sigma(\mathbf{w}^T\mathbf{x}^{(i)})(1-\sigma(\mathbf{w}^T\mathbf{x}^{(i)})\mathbf{x}^{(i)} \\
&= \sum_{i=0}^{m-1} (-(1-\sigma(\mathbf{w}^T\mathbf{x}^{(i)})y^{(i)} + (1-y^{(i)})\sigma(\mathbf{w}^T\mathbf{x}^{(i)}))\mathbf{x}^{(i)} \\
&= \sum_{i=0}^{m-1} (\sigma(\mathbf{w}^T\mathbf{x}^{(i)}) - y^{(i)})\mathbf{x}^{(i)}
\end{aligned}$$

Classification with a shallow nonlinear model

Q17

Given:

$$\begin{cases} J = -\sum_{i=0}^{m-1} y^{(i)} \log(p^{(i)}) + (1 - y^{(i)}) \log(1 - p^{(i)}) \\ p^{(i)} = \sigma(f^{(2)}(f^{(1)}(\mathbf{x}; \mathbf{W}^{(1)}, \mathbf{b}^{(1)}); \mathbf{w}^{(2)}, b^{(2)})) \\ f^{(2)} = (\mathbf{w}^{(2)})^T max(0, f^{(1)}) + b^{(2)} \\ f^{(1)} = \mathbf{W}^{(1)}\mathbf{x}^{(i)} + \mathbf{b}^{(1)} \end{cases}$$

Then:

$$\begin{cases} \nabla_p J = -\sum_{i=0}^{m-1} (\frac{y^{(i)}}{p^{(i)}} - \frac{1-y^{(i)}}{1-p^{(i)}}) \\ \nabla_{f^{(2)}} p^{(i)} = \sigma(f^{(2)})(1 - \sigma(f^{(2)})) \\ \nabla_{\mathbf{w}^{(2)}} f^{(2)} = max(0, f^{(1)}) \\ \nabla_{b^{(2)}} f^{(2)} = 1 \\ \nabla_{f^{(1)}} f^{(2)} = 0, \quad f^{(1)} < 0 \\ \nabla_{f^{(1)}} f^{(2)} = (\mathbf{w}^{(2)})^T, \quad f^{(1)} > 0 \\ \nabla_{\mathbf{b}^{(1)}} f^{(1)} = 1 \\ \nabla_{\mathbf{W}^{(1)}} f^{(1)} = \mathbf{x}^{(i)} \end{cases}$$

$$\begin{aligned} \nabla_{\mathbf{w}^{(2)}} J = \frac{\partial J}{\partial \mathbf{w}^{(2)}} &= \frac{\partial J}{\partial p^{(i)}} \frac{\partial p^{(i)}}{\partial f^{(2)}} \frac{\partial f^{(2)}}{\partial \mathbf{w}^{(2)}} \\ &= -\sum_{i=0}^{m-1} (\frac{y^{(i)}}{p^{(i)}} - \frac{1 - y^{(i)}}{1 - p^{(i)}}) \sigma(f^{(2)})(1 - \sigma(f^{(2)})) max(0, f^{(1)}) \\ &= \begin{cases} -\sum_{i=0}^{m-1} (y^{(i)} - \sigma(f^{(2)}))(\mathbf{W}^{(1)}\mathbf{x}^{(i)} + \mathbf{b}^{(1)}), \quad \text{if} \quad \mathbf{W}^{(1)}\mathbf{x}^{(i)} + \mathbf{b}^{(1)} \geq 0 \\ 0, \quad \text{if} \quad W^{(1)}x^{(i)} + b^{(1)} < 0 \end{cases} \\ \nabla_{\mathbf{W}^{(1)}} J = \frac{\partial J}{\partial \mathbf{W}^{(1)}} &= \frac{\partial J}{\partial p^{(i)}} \frac{\partial p^{(i)}}{\partial f^{(2)}} \frac{\partial f^{(2)}}{\partial f^{(1)}} \frac{\partial f^{(1)}}{\partial \mathbf{W}^{(1)}} \\ &= \begin{cases} -\sum_{i=0}^{m-1} (\frac{y^{(i)}}{p^{(i)}} - \frac{1-y^{(i)}}{1-p^{(i)}}) \sigma(f^{(2)})(1 - \sigma(f^{(2)}))(\mathbf{w}^{(2)})^T \mathbf{x}^{(i)}, \quad \text{if} \quad \mathbf{W}^{(1)}\mathbf{x}^{(i)} + \mathbf{b}^{(1)} \geq 0 \\ 0, \quad \text{if} \quad \mathbf{W}^{(1)}\mathbf{x}^{(i)} + \mathbf{b}^{(1)} < 0 \end{cases} \\ &= \begin{cases} -\sum_{i=0}^{m-1} (y^{(i)} - \sigma(f^{(2)}))(\mathbf{w}^{(2)})^T \mathbf{x}^{(i)}, \quad \text{if} \quad \mathbf{W}^{(1)}\mathbf{x}^{(i)} + \mathbf{b}^{(1)} \geq 0 \\ 0, \quad \text{if} \quad \mathbf{W}^{(1)}\mathbf{x}^{(i)} + \mathbf{b}^{(1)} < 0 \end{cases} \end{aligned}$$

$$\begin{aligned}
\nabla_{b^{(2)}} J &= \frac{\partial J}{\partial b^{(2)}} = \frac{\partial J}{\partial p^{(i)}} \frac{\partial p^{(i)}}{\partial f^{(2)}} \frac{\partial f^{(2)}}{\partial b^{(2)}} \\
&= -\sum_{i=0}^{m-1} (\frac{y^{(i)}}{p^{(i)}} - \frac{1-y^{(i)}}{1-p^{(i)}}) \sigma(f^{(2)})(1-\sigma(f^{(2)})) \\
&= -\sum_{i=0}^{m-1} (y^{(i)} - \sigma(f^{(2)})) \\
\nabla_{\mathbf{b}^{(1)}} J &= \frac{\partial J}{\partial \mathbf{b}^{(1)}} = \frac{\partial J}{\partial p^{(i)}} \frac{\partial p^{(i)}}{\partial f^{(2)}} \frac{\partial f^{(2)}}{\partial f^{(1)}} \frac{\partial f^{(1)}}{\partial \mathbf{b}^{(1)}} \\
&= \begin{cases} -\sum_{i=0}^{m-1} (\frac{y^{(i)}}{p^{(i)}} - \frac{1-y^{(i)}}{1-p^{(i)}}) \sigma(f^{(2)})(1-\sigma(f^{(2)}))(\mathbf{w}^{(2)})^T, & \text{if} \quad \mathbf{W}^{(1)} x^{(i)} + \mathbf{b}^{(1)} \geq 0 \\ 0, & \text{if} \quad \mathbf{W}^{(1)} x^{(i)} + \mathbf{b}^{(1)} < 0 \end{cases} \\
&= \begin{cases} -\sum_{i=0}^{m-1} (y^{(i)} - \sigma(f^{(2)}))(\mathbf{w}^{(2)})^T, & \text{if} \quad \mathbf{W}^{(1)} x^{(i)} + \mathbf{b}^{(1)} \geq 0 \\ 0, & \text{if} \quad \mathbf{W}^{(1)} x^{(i)} + \mathbf{b}^{(1)} < 0 \end{cases}
\end{aligned}$$

Therefore, the answers for $\frac{\partial J}{\partial \mathbf{w}^{(2)}}$, $\frac{\partial J}{\partial \mathbf{W}^{(1)}}$, $\frac{\partial J}{\partial b^{(2)}}$, and $\frac{\partial J}{\partial \mathbf{b}^{(1)}}$ should be B, A, C, B respectively.