Part 1. Handwriting assignment : (30%)

1. (10%)

Show that the logistic sigmoid function $y = \sigma(x) = \frac{1}{1+e^{-x}}$ satisfies the following properties:

(a) $\frac{\partial\sigma(x)}{x} = \boldsymbol{\sigma(x)\big(1 - \sigma(x)\big)}$ **(3%)**

Ans :

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

$$= \frac{d}{dx}\,(1+e^{-x})^{-1}$$

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

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

$$= \frac{1}{1+e^{-x}}\frac{e^{-x}}{1+e^{-x}}$$

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

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

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

$$= \sigma(x)\big(1-\sigma(x)\big)$$

(b) $\sigma(-x) = 1 - \sigma(x)$ (3%)

Ans :

$$\sigma(-x) = \frac{1}{1+e^{x}} = \frac{1}{1+e^{x}}\,\frac{e^{-x}}{e^{-x}}$$

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

(c) $x = \sigma^{-1}(y) = ln\left(\frac{y}{1-y}\right)$ (4%)

Ans :
To solve for x in terms of σ, we start by isolating the exponential term in the definition of $\sigma(x)$

$$y = \sigma(x) = \frac{1}{1+e^{-x}} = \frac{e^{x}}{1+e^{x}}$$

解出 x

$$y = \frac{e^{x}}{1+e^{x}} = y(1+e^{x}) = e^{x}$$

$$=> ye^{x} + y = e^{x}$$

$$=> ye^{x} - e^{x} = -y$$

$$= > e^{x}(y-1) = -y$$

$$=> e^{x} = \frac{-y}{y-1}$$

取e^{x}自然對數

$$x = ln\left(\frac{y}{1-y}\right)$$

因此

$$x = \sigma^{-1}(y) = ln\left(\frac{y}{1-y}\right) \text{ 得証}$$

2. (10%)

Let's take the derivative of the loss function with respect to the weight W_i:

$$\frac{\partial L(W)}{\partial W_i} = \frac{\partial}{\partial W_i}\left[-\sum_{n=1}^{N}\{y_n\, ln(\hat{y}_n) + (1 - y_n)ln(1 - \hat{y}_n)\}\right]$$

Using chain rule:

$$\frac{\partial L(W)}{\partial W_i} = -\sum_{n=1}^{N}\left(\frac{y_n}{\hat{y}_n} - \frac{1 - y_n}{1 - \hat{y}_n}\right)\left(\frac{\partial \hat{y}_n}{\partial W_i}\right)$$

Substituting in the expression for $\hat{y}_n$:

$$\frac{\partial L(W)}{\partial W_i} = -\sum_{n=1}^{N}\left(\frac{y_n}{\sigma\,(W^T\,\Phi_n)} - \frac{1 - y_n}{1 - \sigma\,(W^T\,\Phi_n)}\right)\frac{\partial}{\partial W_i}[\sigma\,(W^T\,\Phi_n)]$$

Using the chain rule again and the fact that the derivative of the sigmoid function is $\sigma(x)\big(1 - \sigma(x)\big)$:

$$\frac{\partial L(W)}{\partial W_i} = -\sum_{n=1}^{N}\left(\frac{y_n}{\sigma\,(W^T\,\Phi_n)} - \frac{1 - y_n}{1 - \sigma\,(W^T\,\Phi_n)}\right)\sigma\,(W^T\,\Phi_n)\,(1 - \sigma\,(W^T\,\Phi_n))\frac{\partial}{\partial W_i}[W^T\,\Phi_n]$$

Simplifying:

$$\frac{\partial L(W)}{\partial W_i} = -\sum_{n=1}^{N} (\hat{y}_n - y_n)\Phi_n$$

Thus, we have shown that the derivative of the binary cross-entropy loss function with respect to the weight W_i is given by:

$$\frac{\partial L(W)}{\partial W_i} = -\sum_{n=1}^{N} \left(\frac{y_n}{\hat{y}_n} - \frac{(1-y_n)}{1-\hat{y}_n}\right)\frac{\partial \hat{y}_n}{\partial \Phi_n}$$

$$= -\sum_{n=1}^{N} \left(\frac{y_n}{\hat{y}_n} - \frac{(1-y_n)}{1-\hat{y}_n}\right)\hat{y}_n(1-\hat{y}_n)\Phi_n$$

$$= -\sum_{n=1}^{N} \big(y_n(1-\hat{y}_n) - \hat{y}_n(1-y_n)\big)\,\Phi_n$$

$$= -\sum_{n=1}^{N}(\hat{y}_n - y_n)\,\Phi_n$$

This is exactly the same as the desired result, so we have shown that the derivative of the binary cross-entropy loss with respect to the weights is given by:

$$\nabla L(W) = -\sum_{n=1}^{N}(\hat{y}_n - y_n)\,\Phi_n \text{ 得証}$$

3. (10%)

(a) One commonly used loss function for this scenario is the Hinge loss function, defined as:

$$L(W) = \frac{1}{N}\sum_{n=1}^{N} max(0, 1 - y_n W^T\,\Phi_n)$$

This loss function penalizes the model when the predicted value

and the actual value have opposite signs, and encourages the model to increase the margin between the predicted value and the decision boundary.

(b)A good choice of activation function for the output in this case would be the hyperbolic tangent (tanh) function, which maps the output to the range of [-1,1]. This can be expressed as:

$$\boldsymbol{y = tanh(x) = \frac{e^{x} - \ e^{-x}}{e^{x} + \ e^{-x}}}$$

where x is the input to the activation function. The tanh function is continuous and differentiable, which makes it suitable for use in neural networks.