1 April 30

1.1 Feed-forward and Back-propagation of Neural Network

1.1.1 Notation

- $\overrightarrow{X}_\ell$: denotes the input data matrix in layer ℓ with N input size and D input dimensional size.
- $\overrightarrow{W}_\ell$: denotes the weight matrix in layer ℓ with D input dimensional size and H output layer size.
- $\vec{b}_\ell$: denotes the bias vector in layer ℓ with H output layer size.
- $\overrightarrow{a}_\ell$: denotes the output matrix before activation from layer ℓ.
- $f_\ell(\cdot)$: denotes the activation function in layer ℓ.

$$\overrightarrow{X}_\ell = \begin{matrix} \\ N \end{matrix} \begin{matrix} \overset{D}{\overset{\downarrow}{}} \\ \end{matrix} \begin{pmatrix} {}^n(x_\ell)_d \end{pmatrix}$$

$$\overrightarrow{W}_\ell = \begin{matrix} \\ D \end{matrix} \begin{matrix} \overset{H}{\overset{\downarrow}{}} \\ \end{matrix} \begin{pmatrix} (w_\ell)_h^d \end{pmatrix}$$

$$\vec{b}_\ell = \begin{matrix} \overset{H}{\overset{\downarrow}{}} \end{matrix} \begin{pmatrix} (b_\ell)_h \end{pmatrix}$$

$$\overrightarrow{a}_\ell = \quad N \left(\begin{array}{c} {}^{H} \\ {}^{n}(a_\ell)_h = {}^{n}(x_\ell)_d \cdot (w_\ell)_h^d + (b_\ell)_h \end{array} \right)$$

$$\overrightarrow{X}_{\ell+1} = f_\ell(\overrightarrow{a}_\ell) \tag{1.1}$$

1.1.1.1 Feed-Forward Neural Network

For a two layer fully-connected neural network , the network has the following architecture:

$$\overrightarrow{X}_{\ell-1} \mapsto \overrightarrow{a}_{\ell-1} = \overrightarrow{X}_{\ell-1}\overrightarrow{W}_{\ell-1} \to \overrightarrow{X}_\ell = f_{\ell-1}(\overrightarrow{a}_{\ell-1}) \mapsto \overrightarrow{a}_\ell = \overrightarrow{X}_\ell\overrightarrow{W}_\ell \to \hat{y} = \text{softmax}(\overrightarrow{a}_\ell) \tag{1.2}$$

We denote the loss function as,

$$L = loss(y, \hat{y}) = \sum_n \sum_h loss({}^{n}y_h, {}^{n}\hat{y}_h) \tag{1.3}$$

1.1.1.2 Back-propagation algorithm

Let us start by considering the last layer weights $(w_\ell)_h^d$ and perform the derivative on the loss function

$$\frac{\partial}{\partial (w_\ell)_h^d} L = \frac{\partial L}{\partial\, {}^{n}(a_\ell)_h} \frac{\partial\, {}^{n}(a_\ell)_h}{\partial (w_\ell)_h^d} = \frac{\partial L}{\partial\, {}^{n}(a_\ell)_h}\, {}^{n}(x_\ell)_d = {}^{n}(\delta_\ell)_h \cdot {}^{n}(x_\ell)_d = {}_{n}(x_\ell^T)^d \cdot {}^{n}(\delta_\ell)_h \tag{1.4}$$

For the ease of notation, we denote ${}^{n}(\delta_\ell)_h$ as the error signal in layer ℓ. Now, we derivative of $(w_{\ell-1})_h^d$ on the loss function,

$$\frac{\partial}{\partial (w_{\ell-1})_h^d} L = \frac{\partial L}{\partial\, {}^{n}(a_{\ell-1})_h} \frac{\partial\, {}^{n}(a_{\ell-1})_h}{\partial (w_{\ell-1})_h^d} \tag{1.5}$$

$$= \frac{\partial L}{\partial\, {}^{n}(a_{\ell-1})_h}\, {}^{n}(x_{\ell-1})_d \tag{1.6}$$

$$= {}_{n}(x_{\ell-1}^T)^d \cdot {}^{n}(\delta_{\ell-1})_h \tag{1.7}$$

For a two layer($\ell = 2$) fully connected neural network, the error signal for the last layer has the below form,

$$\begin{aligned} {}^{n}(\delta_\ell)_h &= \frac{\partial L}{\partial {}^{n}(a_\ell)_h} \\ &= loss'(y, \hat{y})\hat{y}' \\ &= loss'(y, \hat{y}) \odot f'_\ell({}^{n}(a_\ell)_h) \end{aligned} \tag{1.8}$$

For the error signal in the first layer,

$$\begin{aligned} {}^{n}(\delta_{\ell-1})_h &= \frac{\partial L}{\partial {}^{n}(a_{\ell-1})_h} \\ &= \sum_d \frac{\partial L}{\partial {}^{n}(a_\ell)_d} \cdot \frac{\partial {}^{n}(a_\ell)_d}{\partial {}^{n}(a_{\ell-1})_h} \\ &= \sum_d {}^{n}(\delta_\ell)_d \cdot \frac{\partial {}^{n}(a_\ell)_d}{\partial {}^{n}(a_{\ell-1})_h} \end{aligned}$$

We'll show how to proof $\frac{\partial {}^{n}(a_\ell)_d}{\partial {}^{n}(a_{\ell-1})_h}$. For ${}^{n}(a_\ell)_d$, we know that

$$\begin{aligned} {}^{n}(a_\ell)_d &= {}^{n}(x_\ell)_h \cdot (w_\ell)^h_d + (b_\ell)_d \\ &= f_{\ell-1}({}^{n}(a_{\ell-1})_h)(w_\ell)^h_d + (b_\ell)_d \end{aligned}$$

So,

$$\frac{\partial {}^{n}(a_\ell)_d}{\partial {}^{n}(a_{\ell-1})_h} = f'_{\ell-1}({}^{n}(a_{\ell-1})_h)(w_\ell)^h_d \tag{1.9}$$

Finally, the complete form of the error signal in the first layer,

$$\begin{aligned} {}^{n}(\delta_{\ell-1})_h &= f'_{\ell-1}({}^{n}(a_{\ell-1})_h) \sum_d {}^{n}(\delta_\ell)_d \cdot (w_\ell)^h_d \\ &= f'_{\ell-1}({}^{n}(a_{\ell-1})_h) \odot {}^{n}(\delta_\ell)_d \cdot (w_\ell^T)^d_h \end{aligned}$$

The gradient of bias is similar with the above proof,

$$\frac{\partial}{\partial (b_\ell)_h} L = \frac{\partial L}{\partial {}^{n}(a_\ell)_h} \frac{\partial {}^{n}(a_\ell)_h}{\partial (b_\ell)_h} = \sum_n {}^{n}(\delta_\ell)_h$$

$$\frac{\partial}{\partial (b_{\ell-1})_h} L = \frac{\partial L}{\partial {}^{n}(a_{\ell-1})_h} \frac{\partial {}^{n}(a_{\ell-1})_h}{\partial (b_{\ell-1})_h} = \sum_n {}^{n}(\delta_{\ell-1})_h$$

The loss function, full gradients and L2 regularization,

$$
\begin{aligned}
L &= loss(y, \hat{y}) + \frac{\lambda}{2}(\overrightarrow{W}_{\ell}^{2} + \overrightarrow{W}_{\ell-1}^{2}) \\
\frac{\partial}{\partial (w_\ell)_h^d} L &= {}_n(x_\ell^T)^d \cdot {}^n(\delta_\ell)_h + \lambda (w_\ell)_h^d \\
&= {}_n(x_\ell^T)^d (loss'(y, \hat{y}) \odot f'_\ell({}^n(a_\ell)_h)) + \lambda (w_\ell)_h^d \\
\frac{\partial}{\partial (w_{\ell-1})_h^d} L &= {}_n(x_{\ell-1}^T)^d \, {}^n(\delta_\ell)_h + \lambda (w_{\ell-1})_h^d \\
&= {}_n(x_{\ell-1}^T)^d \cdot (f'_{\ell-1}({}^n(a_{\ell-1})_h) \odot ({}^n(\delta_\ell)_d \cdot (w_\ell^T)_h^d)) + \lambda (w_{\ell-1})_h^d \\
\frac{\partial}{\partial (b_\ell)_h} L &= \frac{\partial L}{\partial \, {}^n(a_\ell)_h} \frac{\partial \, {}^n(a_\ell)_h}{\partial (b_\ell)_h} = \sum_n {}^n(\delta_\ell)_h \\
\frac{\partial}{\partial (b_{\ell-1})_h} L &= \frac{\partial L}{\partial \, {}^n(a_{\ell-1})_h} \frac{\partial \, {}^n(a_{\ell-1})_h}{\partial (b_{\ell-1})_h} = \sum_n {}^n(\delta_{\ell-1})_h
\end{aligned}
$$