# 1 Gradient Search Procedure

## 1.1 Feedforward neural networks

## 1.2 Backpropagation neural networks

## 1.3 General form of Feed-forward and Back-propagation

### 1.3.1 Notation

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

$$\overrightarrow{X}_\ell = \begin{array}{cc} & D \\ N & \begin{pmatrix} & & \\ & {}^{n}(x_\ell)_d & \\ & & \end{pmatrix} \end{array}$$

$$\overrightarrow{W}_\ell = \begin{array}{cc} & H \\ D & \begin{pmatrix} \quad (w_\ell)_h^d \quad \end{pmatrix} \end{array}$$

$$\vec{b}_\ell = \begin{array}{c} H \\ \begin{pmatrix} \quad (b_\ell)_h \quad \end{pmatrix} \end{array}$$

$$\overrightarrow{a}_\ell = \begin{array}{cc} & H \\ N & \begin{pmatrix} \quad {}^n(a_\ell)_h = {}^n(x_\ell)_d \cdot (w_{\ell-1})_h^d + (b_{\ell-1})_h \quad \end{pmatrix} \end{array}$$

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

#### 1.3.1.1 Feed-Forward Neural Network

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

$$\overrightarrow{X}_{\ell-1} \mapsto \overrightarrow{a}_{\ell-1} = \overrightarrow{X}_{\ell-1}\overrightarrow{W}_{\ell-2} \to \overrightarrow{X}_\ell = f_{\ell-1}(\overrightarrow{a}_{\ell-1}) \mapsto \overrightarrow{a}_\ell = \overrightarrow{X}_\ell\overrightarrow{W}_{\ell-1} \to \hat{y} = \mathrm{softmax}(\overrightarrow{a}_\ell) \tag{1.2}$$

We denote the loss function as,

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

#### 1.3.1.2 Back-propagation Neural Network

Let us start by considering the last layer weights $(w_{\ell-1})_h^d$ and perform the derivative on the loss function

$$\frac{\partial}{\partial (w_{\ell-1})_h^d} L = \frac{\partial L}{\partial\, {}^n(a_\ell)_h} \frac{\partial\, {}^n(a_\ell)_h}{\partial (w_{\ell-1})_h^d} = \frac{\partial L}{\partial\, {}^n(a_\ell)_h}\, {}^n(x_\ell)_d = {}^n(\delta_\ell)_h \cdot {}^n(x_\ell)_d = {}_n(x_\ell^T)^d \cdot {}^n(\delta_\ell)_h \quad (1.4)$$

For the ease of notation, we denote ${}^n(\delta_\ell)_h$ as the error signal in layer $\ell$. Now, we derivative of $(w_{\ell-2})_h^d$ on the loss function,

$$\frac{\partial}{\partial (w_{\ell-2})_h^d} L = \frac{\partial L}{\partial\, {}^n(a_{\ell-1})_h} \frac{\partial\, {}^n(a_{\ell-1})_h}{\partial (w_{\ell-2})_h^d} \quad (1.5)$$

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

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

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

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

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

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

So,

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

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

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

The gradient of bias is similar with the above proof,

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

The loss function, full gradients and L2 regularization,

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