Back Propagation of Neural Network

Masayuki Tanaka

2013/09/06

1 Notations

Let's consider four layer network as shown in Fig. 1, for example. The output of output nodes can be expressed as:

$$o_k = f(n_k)\,, \tag{1}$$
$$n_k = \sum_j w_{jk} o_j\,, \tag{2}$$

where w_{jk} is the weight of edge jk, $f(n_k)$ is the sigmoid function, and other symbols are shown in Fig. 1. For simplification, the bias term is ommited.

2 Back Propagation

2.1 Loss functions

Loss functions for a certain training samples can be expressed as:

$$L_{\mathrm{SquarL}} = \frac{1}{2}\sum_k (t_j - o_j)^2\,, \tag{3}$$
$$L_{\mathrm{CrossEntropy}} = -\sum_k \left[t_j \log o_j + (1-t_j)\log(1-o_j)\right]\,, \tag{4}$$

where t_j is the ground truth output.

2.2 Derivative of the output layer

Let's derive the derivative of the loss function with respect to w_{jk}.

$$\frac{\partial L}{\partial w_{jk}} = \frac{\partial L}{\partial o_k}\frac{\partial o_k}{\partial n_k}\frac{\partial n_k}{\partial w_{jk}} = \delta_k \frac{\partial n_k}{\partial w_{jk}}\,, \tag{5}$$

where

$$\delta_k = \frac{\partial L}{\partial n_k} = \frac{\partial L}{\partial o_k}\frac{\partial o_k}{\partial n_k} \tag{6}$$

From Eq. (2),

$$\frac{\partial \partial n_k}{\partial w_{jk}} = o_j\,. \tag{7}$$

Then, the derivative can be expressed with δ_k

$$\frac{\partial L}{\partial w_{jk}} = \delta_k o_j\,. \tag{8}$$

For the square loss function,

$$\begin{aligned}
\delta_k &= (o_k - t_k) o_k (1 - o_k)\,, && (9)\\
\frac{\partial L}{\partial o_k} &= (o_k - t_k)\,, && (10)\\
\frac{\partial o_k}{\partial n_k} &= f'(o_k) \;=\; o_k(1 - o_k)\,. && (11)
\end{aligned}$$

For the cross entropy loss function,

$$\begin{aligned}
\delta_k &= o_k - t_k\,, && (12)\\
\frac{\partial L}{\partial o_k} &= -\frac{t_k}{o_k} - \frac{1 - t_k}{1 - o_k} \;=\; \frac{o_k - t_k}{o_k(1 - o_k)}\,, && (13)\\
\frac{\partial o_k}{\partial n_k} &= f'(o_k) \;=\; o_k(1 - o_k)\,. && (14)
\end{aligned}$$

2.3 Derivative of the hidden layers

Let's derive the derivative of the loss function with respect to w_{ij}.

$$\begin{aligned}
\frac{\partial L}{\partial w_{ij}} &= \frac{\partial L}{\partial n_j} \times \frac{\partial n_j}{\partial w_{ij}} && (15)\\
&= \left[\sum_k \frac{\partial L}{\partial n_k}\frac{\partial n_k}{\partial o_j}\right] \frac{\partial o_j}{\partial n_j} \times \frac{\partial n_j}{\partial w_{ij}} && (16)\\
&= \left[\sum_k \delta_k w_{jk}\right] o_j(1 - o_j) \times \frac{\partial n_j}{\partial w_{ij}} && (17)\\
&= \delta_j \times o_i && (18)
\end{aligned}$$

where

$$\delta_j = \frac{\partial L}{\partial n_j} = \left[\sum_k \delta_k w_{jk}\right] o_j(1 - o_j) \qquad (19)$$

Let's derive the derivative of the loss function with respect to w_{hi}.

$$\begin{aligned}
\frac{\partial L}{\partial w_{hi}} &= \frac{\partial L}{\partial n_i} \times \frac{\partial n_i}{\partial w_{hi}} && (20)\\
&= \left[\sum_j \frac{\partial L}{\partial n_j}\frac{\partial n_j}{\partial o_i}\right] \frac{\partial o_i}{\partial n_i} \times \frac{\partial n_i}{\partial w_{hi}} && (21)\\
&= \left[\sum_j \delta_j w_{ij}\right] o_i(1 - o_i) \times \frac{\partial n_i}{\partial w_{hi}} && (22)\\
&= \delta_i \times o_h && (23)
\end{aligned}$$

where

$$\delta_i = \frac{\partial L}{\partial n_i} = \left[\sum_j \delta_j w_{ij}\right] o_i(1 - o_i) \qquad (24)$$

For hidden layer, the form of the derivatives do not depend on the loss function.

3 Back Propagation for Fast Dropout

3.1 Closed form approximation

In the dropout case, the output of the node can be approximated [1] as

$$o_k = E_{\boldsymbol{Z}}\left[\sigma\left(\sum_j z_{jk} w_{jk} o_j\right)\right] \simeq \sigma(n'_k)\,, \tag{25}$$

where

$$n'_k = \frac{\mu_k}{\sqrt{1+s_k^2\pi/8}}\,, \tag{26}$$

$$\mu_k = \sum_j p_{jk} w_{jk} o_j = p\sum_j w_{jk} o_j\,, \tag{27}$$

$$s_k^2 = \sum_j p_{jk}(1-p_{jk}) w_{jk}^2 o_j^2 = p(1-p)\sum_j w_{jk}^2 o_j^2\,. \tag{28}$$

3.2 Derivatives of output layer

$$\frac{\partial L}{\partial w_{jk}} = \frac{\partial L}{\partial \mu_k}\frac{\partial \mu_k}{\partial w_{jk}} + \frac{\partial L}{\partial s_k^2}\frac{\partial s_k^2}{\partial w_{jk}} = \delta_k \frac{\partial \mu_k}{\partial w_{jk}} + \tau_k \frac{\partial s_k^2}{\partial w_{jk}} \tag{29}$$

where

$$\delta_k = \frac{\partial L}{\partial \mu_k} = \frac{\partial L}{\partial o_k}\frac{\partial o_k}{\partial n'_k}\frac{\partial n'_k}{\partial \mu_k} \tag{30}$$

$$\tau_k = \frac{\partial L}{\partial s_k^2} = \frac{\partial L}{\partial o_k}\frac{\partial o_k}{\partial n'_k}\frac{\partial n'_k}{\partial s_k^2} \tag{31}$$

$$\frac{\partial \mu_k}{\partial w_{jk}} = p o_j \tag{32}$$

$$\frac{\partial s_k^2}{\partial w_{jk}} = 2p(1-p) w_{jk} o_j^2 \tag{33}$$

For square loss function,

$$\delta_k = \frac{\partial L}{\partial o_k}\frac{\partial o_k}{\partial n'_k}\frac{\partial n'_k}{\partial \mu_k} = (o_k - t_k) \times o_k(1-o_k) \times \frac{1}{(1+s_k^2\pi/8)^{1/2}} \tag{34}$$

$$\tau_k = \frac{\partial L}{\partial o_k}\frac{\partial o_k}{\partial n'_k}\frac{\partial n'_k}{\partial s_k^2} = (o_k - t_k) \times o_k(1-o_k) \times \left[-\frac{\pi}{16}\frac{\mu}{(1+s_k^2\pi/8)^{3/2}}\right] \tag{35}$$

For cross entropy loss function,

$$\delta_k = \frac{\partial L}{\partial o_k}\frac{\partial o_k}{\partial n'_k}\frac{\partial n'_k}{\partial \mu_k} = (o_k - t_k) \times \frac{1}{(1+s_k^2\pi/8)^{1/2}} \tag{36}$$

$$\tau_k = \frac{\partial L}{\partial o_k}\frac{\partial o_k}{\partial n'_k}\frac{\partial n'_k}{\partial s_k^2} = (o_k - t_k) \times \left[-\frac{\pi}{16}\frac{\mu}{(1+s_k^2\pi/8)^{3/2}}\right] \tag{37}$$

3.3 Derivatives of hidden layers

$$\frac{\partial L}{\partial w_{ij}} = \frac{\partial L}{\partial \mu_j}\frac{\partial \mu_j}{\partial w_{ij}} + \frac{\partial L}{\partial s_j^2}\frac{\partial s_j^2}{\partial w_{ij}} = \delta_j \frac{\partial \mu_j}{\partial w_{ij}} + \tau_j \frac{\partial s_j^2}{\partial w_{ij}} \tag{38}$$

$$= \left[\sum_k \left\{\frac{\partial L}{\partial \mu_k}\frac{\partial \mu_k}{\partial o_j} + \frac{\partial L}{\partial s_k^2}\frac{\partial s_k^2}{\partial o_j}\right\}\right]\frac{\partial o_j}{\partial n'_j}\frac{\partial n'_j}{\partial w_{ij}} \tag{39}$$

$$= \left[\sum_k \left\{\frac{\partial L}{\partial \mu_k}\frac{\partial \mu_k}{\partial o_j} + \frac{\partial L}{\partial s_k^2}\frac{\partial s_k^2}{\partial o_j}\right\}\right]\frac{\partial o_j}{\partial n'_j}\left(\frac{\partial n'_j}{\partial \mu_j}\frac{\partial \mu_j}{\partial w_{ij}} + \frac{\partial n'_j}{\partial s_j^2}\frac{\partial s_j^2}{\partial w_{ij}}\right) \tag{40}$$

$$\begin{cases} \frac{\partial L}{\partial \mu_k} &= \delta_k \\ \frac{\partial L}{\partial s_k^2} &= \tau_k \\ \frac{\partial \mu_k}{\partial o_j} &= pw_{ij} \\ \frac{\partial s_k^2}{\partial o_j} &= 2p(1-p)w_{ij}^2 o_j \\ \frac{\partial n'_j}{\partial \mu_j} &= \frac{1}{(1+s_j^2\pi/8)^{1/2}} \\ \frac{\partial n'_j}{\partial s_j^2} &= -\frac{\pi}{16}\frac{\mu}{(1+s_j^2\pi/8)^{3/2}} \\ \frac{\partial o_j}{\partial n'_j} &= o_j(1-o_j) \end{cases} \tag{41}$$

$$\delta_j = \left[\sum_k \left\{\frac{\partial L}{\partial \mu_k}\frac{\partial \mu_k}{\partial o_j} + \frac{\partial L}{\partial s_k^2}\frac{\partial s_k^2}{\partial o_j}\right\}\right]\frac{\partial o_j}{\partial n'_j}\frac{\partial n'_j}{\partial \mu_j} = \alpha_j \frac{o_j(1-o_j)}{(1+s_j^2\pi/8)^{1/2}} \tag{42}$$

$$\tau_j = \left[\sum_k \left\{\frac{\partial L}{\partial \mu_k}\frac{\partial \mu_k}{\partial o_j} + \frac{\partial L}{\partial s_k^2}\frac{\partial s_k^2}{\partial o_j}\right\}\right]\frac{\partial o_j}{\partial n'_j}\frac{\partial n'_j}{\partial s_j^2} = -\alpha_j \frac{\pi}{16}\frac{o_j(1-o_j)}{(1+s_j^2\pi/8)^{3/2}} \tag{43}$$

$$\alpha_j = \left[\sum_k \left\{\delta_k \times pw_{jk} + \tau_k \times 2p(1-p)w_{jk}^2 o_j\right\}\right] \tag{44}$$

References

[1] Sida Wang and Christopher Manning, Fast dropout training, Proc. International Conference on Machine Learning (ICML), pp. 118–126, 2013.

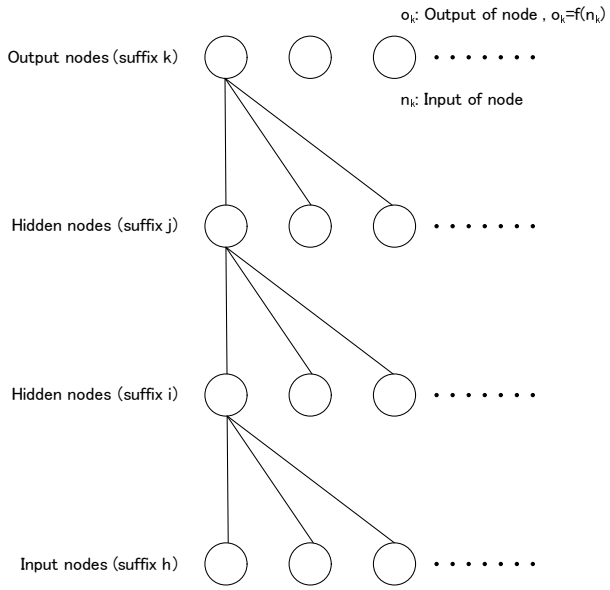

Figure 1: Notations