Assignment of chapter 6

Chengzhang Yang 16337272 Intelligent Science and Technology

3.

(a) $dn_H + n_H c + nd$

(b)

hidden layer -> output layer

> $$\begin{aligned}\frac{\partial J}{\partial W_{kj}^{(2)}} &= \frac{\partial J}{\partial z}\frac{\partial z}{\partial net_k}\frac{\partial net_k}{\partial W_{kj}^{(2)}} \\ &= (t_k - z_k)f'(net_k)y_i\end{aligned}$$

input layer -> hidden layer

> $$\frac{\partial J}{\partial W_{ji}^{(1)}} = \frac{\partial J}{\partial y_j}\frac{\partial y_j}{\partial net_j}\frac{\partial net_j}{\partial W_{ji}^{(1)}}$$
>
> $$\begin{aligned}\frac{\partial J}{\partial y_j} &= -\sum_{k=1}^{c}(t_k - z_k)\frac{\partial z_k}{\partial y_j} \\ &= -\sum_{k=1}^{c}(t_k - z_k)f'(net_k)W_{kj}^{(2)}\end{aligned}$$
>
> $$\frac{\partial J}{\partial W_{ji}^{(1)}} = -\sum_{k=1}^{c}(t_k - z_k)f'(net_k)W_{kj}^{(2)}f'(net_j)x_i$$

Only consider multiplication, the computation complexity is $O(m_e(n_H c + dn_H c)) = O(m_e dn_H c)$

(c) $O(nm_e dn_H c)$

8.

(a) $dn_H + n_H c + c$

(b)

$$y_j = f(\sum_{i=1}^{d} W_{ji}^{(1)}x_i)$$

$$z_k = f(\sum_{j=1}^{n_H} y_j W_{kj}^{(2)} + W_{k0})$$

Suppose activation function is an odd function

$$
y'_j = f(\sum_{i=1}^{d} -W_{ji}^{(1)} x_i) = -f(\sum_{i=1}^{d} W_{ji}^{(1)} x_i) = -y_j
$$

$$
z'_k = f(\sum_{j=1}^{n_H} (-y_j)(-W_{kj}^{(2)}) + W_{k0}) = z_k
$$

(c)

Exchange the sign of the weights, we can form 2^{n_H} different units.

Exchange the location of the weights, we can construct $n_H!$ different units.

Totally, we can construct $n_H!2^{n_H}$ different units.

When $n_H = 10$, the value of $n_H!2^{n_H}$ is approximately 3×10^9. The accurate value is 3715891200.

9.

begin initialize n_H, w, η, $m \leftarrow 0$

do $m \leftarrow m + 1$

$x^m \leftarrow$ next pattern

$w_{ji}^{(1)} \leftarrow w_{ji} + \eta \delta_j x_i; w_{kj}^{(2)} \leftarrow w_{kj} + \eta \delta_k y_i$

return w

end

10.

(a)

$$
\begin{aligned}
f'(net) &= \frac{-ae^{anet}}{1+e^{anet}} \frac{1}{1+e^{anet}} \\
&= a(f(net) - 1)f(net)
\end{aligned}
$$

(b)

$$
\begin{aligned}
tanh'(net) &= \frac{be^{-bnet}(1+e^{-bnet}) + b(1-e^{-bnet})e^{-bnet}}{(1+e^{-bnet})^2} \\
&= 2b\frac{e^{-bnet}}{(1+e^{-bnet})^2} \\
tanh(x)^2 &= a^2 \frac{1+e^{-2bx} - 2e^{-bx}}{1+e^{-2bx} + 2e^{-bx}} \\
tanh(x)^2 - a^2 &= a^2 \frac{-4e^{-bx}}{(1+e^{-bx})^2} \\
\therefore atanh'(net) &= \frac{b}{2a}(a - tanh(net))(a + tanh(net))
\end{aligned}
$$

12.

When all the weight from input layer to hidden layer are identical. In initial state, all hidden neurons have the same value. From equation

$$\frac{\partial J}{\partial W_{kj}^{(2)}} = (t_k - z_k)f'(net_k)y_i$$

where all y has identical value, we know that, in back propagation procedure, $W_{ki}^{(2)}, i = 1, 2, \ldots n_H$ has the same gradient with net_k. On the other hand

$$\frac{\partial J}{\partial W_{ji}^{(1)}} = -\sum_{k=1}^{c}(t_k - z_k)f'(net_k)W_{kj}^{(2)}f'(net_j)x_i$$

Since $W^{(2)}$ could be initialized randomly, after the first back propagation, value of $W^{(1)}$ may be different. After that, The neural network can work as other general network. Note that, if $W^{(2)}$ is also initialized with identical weight, the neural network will no longer work since gradient of all $W^{(1)}$ are the same. After first back propagation, all value of $W^{(1)}$ are still identical. Thereafter, a circle construct. That's to say the power of the hidden layer is identical as a single neuron. Generally, if we initialize n th previous layers, we need to take n times back propagation to make the multi layer network work normally.

26.

(a) Same as **10.(b)**

(b)

$$f(net) = \begin{cases} -a & , net = -\infty \\ 0 & , net = 0 \\ a & , net = \infty \end{cases}$$

$$f'(net) = \begin{cases} 0 & , net = -\infty \\ \frac{ab}{2} & , net = 0 \\ 0 & , net = \infty \end{cases}$$

$$f''(net) = \begin{cases} 0 & , net = -\infty \\ 0 & , net = 0 \\ 0 & , net = \infty \end{cases}$$