1: PRML Pg. 284, Ex. 5.1

Answer:

Goal is to find relation between sigmoid and hyperbolic tangent functions, then show that network parameters of Eq. 5.7 differ by linear transformations.

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

$$\tanh(a) = \frac{e^a - e^{-a}}{e^a + e^{-a}}$$

$$\tanh(a) = \frac{e^a - e^{-a} + e^{-a} - e^{-a}}{e^a + e^{-a}}$$

$$\tanh(a) = \frac{e^a + e^{-a}}{e^a + e^{-a}} + \frac{-2e^{-a}}{e^a + e^{-a}}$$

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

$$\tanh(a) = 1 - \frac{2e^{-a}}{e^a + e^{-a}} * \frac{e^a}{e^a}$$

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

$$\tanh(a) = 1 - 2\sigma(-2a)$$

Note the following equivalence, proved in PS2: $\sigma(-a) = -\sigma(a) + 1$

$$\tanh(a) = 1 - 2(-\sigma(2a) + 1)$$

$$\tanh(a) = 1 + 2\sigma(2a) - 2$$

$$\tanh(a) = 2\sigma(2a) - 1$$

The input to the kth node of the output layer, with weights $w_{kj}^{(2)}$ (M weights, each corresponding to a hidden node j connecting to node k), hidden layer input a_j, and bias $w_{k0}^{(2)}$ is given as follows, using Eq. 5.4 and Eq. 5.3:

$$a_k^{(t)} = \sum_{j=1}^{M} w_{kj}^{(2t)} \tanh(a_j^{(t)}) + w_{k0}^{(2t)}$$

Here, the activation function of the hidden layer is hyperbolic tangent. Rewriting with the sigmoid, tanh relation:

$$a_k^{(t)} = \sum_{j=1}^{M} w_{kj}^{(2t)} (2\sigma(2a_j^{(t)}) - 1) + w_{k0}^{(2t)}$$

$$a_k^{(t)} = \sum_{j=1}^{M} 2w_{kj}^{(2t)} \sigma(2a_j^{(t)}) - \sum_{j=1}^{M} w_{kj}^{(2t)} + w_{k0}^{(2t)}$$

On the other hand, if the activation function of the hidden layer is sigmoid, we have the following input to the output layer, from Eq. 5.4:

$$a_k^{(\sigma)} = \sum_{j=1}^{M} w_{kj}^{(2\sigma)} \sigma(a_j^{(\sigma)}) + w_{k0}^{(2\sigma)}$$

We can equate these two distinct output layer inputs with the following relations:

1) Between hidden layer inputs

$$2a_j^{(t)} = a_j^{(\sigma)}$$

2) Between weights from the hidden to output layers

$$2w_{kj}^{(2t)} = w_{kj}^{(2\sigma)}$$

3) Between biases

$$-\sum_{j=1}^{M} w_{kj}^{(2t)} + w_{k0}^{(2t)} = w_{k0}^{(2\sigma)}$$

This is almost to say that the two differently activated networks are the same given linear transformations of parameters, but relation 1 above linearly transforms hidden layer inputs. Luckily, Eq. 5.2 gives the following:

$$a_j^{(t)} = \sum_{i=1}^{D} w_{ji}^{(1t)} x_i + w_{j0}^{(1t)}$$

$$a_j^{(\sigma)} = \sum_{i=1}^{D} w_{ji}^{(1\sigma)} x_i + w_{j0}^{(1\sigma)}$$

Where x_i is one of D inputs, which create the hidden layer input a_j via the sum of the products of weight w_{ji}^1 and input x_i. In addition, there is a bias $w_{j0}^{(1)}$. Using relation 1, we find the following:

$$a_j^{(t)} = \sum_{i=1}^{D} w_{ji}^{(1t)} x_i + w_{j0}^{(1t)} = 2a_j^{(\sigma)} = \sum_{i=1}^{D} 2w_{ji}^{(1\sigma)} x_i + 2w_{j0}^{(1\sigma)}$$

So relation 1 can be rewritten as follows:

$$w_{ji}^{(1t)} = 2w_{ji}^{(1\sigma)}$$

and

$$w_{j0}^{(1t)} = 2w_{j0}^{(1\sigma)}$$

A neural network with hidden unit activation function tanh is therefore equivalent to a neural network with hidden unit activation sigmoid, given a linear transformation of the weights and biases between the input and hidden, and hidden and output layers.

2: PRML Pg. 284, Ex. 5.2

Answer:

Eq. 5.16, conditional target distribution:

$$p(\mathbf{t}|\mathbf{x},\mathbf{w}) = \mathcal{N}(\mathbf{t}|\mathbf{y}(\mathbf{x},\mathbf{w}),\beta^{-1}\mathbf{I})$$

Eq. 5.11, sum of squares error:

$$E(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\|\mathbf{y}(\mathbf{x}_n,\mathbf{w}) - \mathbf{t}_n\|^2$$

The likelihood function is the product of probabilities of the target vectors $\mathbf{t}_n$ (Eq. 5.16) for all N examples:

$$\prod_{n=1}^{N}\mathcal{N}(\mathbf{t}_n|\mathbf{y}(\mathbf{x}_n,\mathbf{w}),\beta^{-1}\mathbf{I})$$

We want to maximize the likelihood function in order to have a neural network with the highest probability of matching the target vector for a given example. Since logarithms are monotonic, maximizing the likelihood function is the same as maximizing the logarithm of the likelihood function:

$$\ln\prod_{n=1}^{N}\mathcal{N}(\mathbf{t}_n|\mathbf{y}(\mathbf{x}_n,\mathbf{w}),\beta^{-1}\mathbf{I})$$

By the rules of logarithms, the log of a product is equivalent to the sum of logs of the factors:

$$\ln\prod_{n=1}^{N}\mathcal{N}(\mathbf{t}_n|\mathbf{y}(\mathbf{x}_n,\mathbf{w}),\beta^{-1}\mathbf{I})$$

$$= \sum_{n=1}^{N}\ln\mathcal{N}(\mathbf{t}_n|\mathbf{y}(\mathbf{x}_n,\mathbf{w}),\beta^{-1}\mathbf{I})$$

Eq. 2.43:

$$\mathcal{N}(\mathbf{x}|\mu,\mathbf{\Sigma}) = \frac{1}{(2\pi)^{D/2}}\frac{1}{|\mathbf{\Sigma}|^{1/2}}\exp\left(-\frac{1}{2}(\mathbf{x}-\mu)^T\mathbf{\Sigma}^{-1}(\mathbf{x}-\mu)\right)$$

Here, $\mathbf{x} = \mathbf{t}_n$ is the normally distributed variable, $\mu = \mathbf{y}(\mathbf{x}_n,\mathbf{w})$ is its mean, and $\mathbf{\Sigma} = \beta^{-1}\mathbf{I}$ is its covariance matrix. Substituting this into the log-likelihood function:

$$\sum_{n=1}^{N}\ln\mathcal{N}(\mathbf{t}_n|\mathbf{y}(\mathbf{x}_n,\mathbf{w}),\beta^{-1}\mathbf{I})$$

$$= \sum_{n=1}^{N}\ln[\frac{1}{(2\pi)^{D/2}}\frac{1}{|\beta^{-1}\mathbf{I}|^{1/2}}\exp\left(-\frac{1}{2}(\mathbf{t}_n - \mathbf{y}(\mathbf{x}_n,\mathbf{w}))^T(\beta^{-1}\mathbf{I})^{-1}(\mathbf{t}_n - \mathbf{y}(\mathbf{x}_n,\mathbf{w}))\right)]$$

Again, the log of a product is equivalent to the sum of logs of the factors:

$$= \sum_{n=1}^{N}\ln[\exp\left(-\frac{1}{2}(\mathbf{t}_n - \mathbf{y}(\mathbf{x}_n,\mathbf{w}))^T(\beta^{-1}\mathbf{I})^{-1}(\mathbf{t}_n - \mathbf{y}(\mathbf{x}_n,\mathbf{w}))\right)] + \ln[\frac{1}{(2\pi)^{D/2}}\frac{1}{|\beta^{-1}\mathbf{I}|^{1/2}}]$$

The covariance matrix is independent of the example n, so just call the second term a constant C:

$$= \sum_{n=1}^{N} \ln[\exp{(-\frac{1}{2}(\mathbf{t}_n - \mathbf{y}(\mathbf{x}_n, \mathbf{w}))^T(\beta^{-1}\mathbf{I})^{-1}(\mathbf{t}_n - \mathbf{y}(\mathbf{x}_n, \mathbf{w})))}] + C$$

The natural log of an exponential is just the exponent:

$$= \sum_{n=1}^{N} -\frac{1}{2}(\mathbf{t}_n - \mathbf{y}(\mathbf{x}_n, \mathbf{w}))^T(\beta^{-1}\mathbf{I})^{-1}(\mathbf{t}_n - \mathbf{y}(\mathbf{x}_n, \mathbf{w})) + C$$

Note that the inverse of the identity matrix $\mathbf{I}$ is itself:

$$= \sum_{n=1}^{N} -\frac{1}{2}(\mathbf{t}_n - \mathbf{y}(\mathbf{x}_n, \mathbf{w}))^T(\beta\mathbf{I})(\mathbf{t}_n - \mathbf{y}(\mathbf{x}_n, \mathbf{w})) + C$$

β is the inverse variance of the Gaussian noise, and so is a number (not a vector) that is constant with respect to training example n, meaning it can be pulled out of the sum.

$$= -\frac{1}{2}\beta \sum_{n=1}^{N} (\mathbf{t}_n - \mathbf{y}(\mathbf{x}_n, \mathbf{w}))^T(\mathbf{I})(\mathbf{t}_n - \mathbf{y}(\mathbf{x}_n, \mathbf{w})) + C$$

Also, anything times the identity matrix $\mathbf{I}$ is itself.

$$= -\frac{1}{2}\beta \sum_{n=1}^{N} (\mathbf{t}_n - \mathbf{y}(\mathbf{x}_n, \mathbf{w}))^T(\mathbf{t}_n - \mathbf{y}(\mathbf{x}_n, \mathbf{w})) + C$$

By definition, $\mathbf{x}^T\mathbf{x} = \|\mathbf{x}\|^2$.

$$= -\frac{1}{2}\beta \sum_{n=1}^{N} \|(\mathbf{t}_n - \mathbf{y}(\mathbf{x}_n, \mathbf{w}))\|^2 + C$$

The first term on the left is related to the error function of Eq. 5.11 by a factor of -β. Maximizing the log-likelihood defined in the equation above, therefore, is equivalent to minimizing the squared sum of errors of Eq. 5.11.

3: PRML Pg. 285, Ex. 5.5

Answer:

A network output from node k has the following interpretation:

$$y_k(\mathbf{x}, \mathbf{w}) = p(t_k = 1|\mathbf{x})$$

The cross-entropy error function Eq. 5.24 is as follows, where the N examples are each denoted n, and the K output nodes are each denoted k:

$$E(\mathbf{w}) = -\sum_{n=1}^{N}\sum_{k=1}^{K} t_{nk} \ln y_k(\mathbf{x}_n, \mathbf{w})$$

In this question, the output is interpreted as the probability of $t_k = 1$, so the cross-entropy becomes the following:

$$E(\mathbf{w}) = -\sum_{n=1}^{N}\sum_{k=1}^{K} 1 \ln y_k(\mathbf{x}_n, \mathbf{w})$$

$$E(\mathbf{w}) = -\sum_{n=1}^{N}\sum_{k=1}^{K} \ln y_k(\mathbf{x}_n, \mathbf{w}) \tag{1}$$

Given our output interpretation, the probability of achieving a target $\mathbf{t}$ is as follows:

$$p(\mathbf{t}|\mathbf{w}_1, ..., \mathbf{w}_k) = \prod_{k=1}^{K} y_k^{t_k}$$

because the probability of a target t_k is given by the Bernoulli distribution Eq. 5.20, where $t_k = 1$ for our output interpretation:

$$p(t_k = 1|\mathbf{x}, \mathbf{w}) = y(\mathbf{x}, \mathbf{w})^1(1 - y(\mathbf{x}, \mathbf{w}))^{1-1}$$

$$p(t_k = 1|\mathbf{x}, \mathbf{w}) = y(\mathbf{x}, \mathbf{w})$$

For all N training examples, then, the probability of achieving the target matrix T, also known as the likelihood function, is as follows:

$$p(\mathbf{T}|\mathbf{w}_1, ..., \mathbf{w}_k) = \prod_{n=1}^{N}\prod_{k=1}^{K} y_k^{t_k}$$

We want to maximize this likelihood. Equivalently, since logs are monotonic, we can minimize the negative log of the likelihood:

$$-\ln \prod_{n=1}^{N}\prod_{k=1}^{K} y_k^{t_k}$$

By the rules of logs, the log of a product is equivalent to the sum of logs of the factors:

$$= -\sum_{n=1}^{N} \ln \prod_{k=1}^{K} y_k^{t_k}$$

Applying the same rule again:

$$= -\sum_{n=1}^{N}\sum_{k=1}^{K} \ln y_k^{t_k}$$

This negative log likelihood, which we want to minimize, is the same as the error function given above in Equation 1 (number refers to this document, not PRML), which we also want to minimize.

4: PRML Pg. 285, Ex. 5.6

Answer:

Eq. 5.21 error function, where there are N training examples, each labeled n.:

$$E(\mathbf{w}) = -\sum_{n=1}^{N} [t_n \ln y_n + (1 - t_n) \ln(1 - y_n)]$$

$$E(\mathbf{w}) = -\sum_{n=1}^{N} [t_n \ln \sigma(a_n) + (1 - t_n) \ln(1 - \sigma(a_n))]$$

The latter equation is for the case of a sigmoid activation function in the final layer. If there are K separate binary classifications to perform (K output layer units), each denoted k, then this error becomes the following:

$$E(\mathbf{w}) = -\sum_{n=1}^{N} \sum_{k=1}^{K} [t_{nk} \ln \sigma(a_{nk}) + (1 - t_{nk}) \ln(1 - \sigma(a_{nk}))]$$

Say we only consider the nth training example, whose error is as follows:

$$E(\mathbf{w}) = -\sum_{k=1}^{K} [t_k \ln \sigma(a_k) + (1 - t_k) \ln(1 - \sigma(a_k))]$$

Considering only the kth output, the error function becomes the following:

$$E(\mathbf{w}) = -t_k \ln \sigma(a_k) + (1 - t_k) \ln(1 - \sigma(a_k))$$

Differentiating this with respect to the kth activation, we find the following:

$$\frac{\partial E(\mathbf{w})}{\partial a_k} = -t_k \frac{\partial}{\partial a_k} \ln \sigma(a_k) + (1 - t_k) \frac{\partial}{\partial a_k} \ln(1 - \sigma(a_k))$$

where the target t_k can be pulled out of the derivative since it is independent of the activation a_k. Applying the chain rule and the fact that $\frac{d}{dx} \ln x = \frac{1}{x}$, we find the following:

$$\frac{\partial E(\mathbf{w})}{\partial a_k} = -t_k \frac{1}{\sigma(a_k)} \frac{\partial}{\partial a_k} \sigma(a_k) + (1 - t_k) \frac{1}{1 - \sigma(a_k)} \frac{\partial}{\partial a_k} (1 - \sigma(a_k))$$

Equation 4.88 gives us the following:

$$\frac{d\sigma(a_k)}{da_k} = \sigma(a_k)(1 - \sigma(a_k))$$

Substituting 4.88 into the error derivative...

$$\frac{\partial E(\mathbf{w})}{\partial a_k} = -t_k \frac{1}{\sigma(a_k)} \sigma(a_k)(1 - \sigma(a_k)) + (1 - t_k) \frac{1}{1 - \sigma(a_k)} \sigma(a_k)(1 - \sigma(a_k))$$

Simplifying...

$$\frac{\partial E(\mathbf{w})}{\partial a_k} = -t_k(1 - \sigma(a_k)) + \frac{1}{1 - \sigma(a_k)} \sigma(a_k)(1 - \sigma(a_k)) - \frac{t_k}{1 - \sigma(a_k)} \sigma(a_k)(1 - \sigma(a_k))$$

At this point we can use the original notation $\sigma(a_k) = y_k$...

$$\frac{\partial E(\mathbf{w})}{\partial a_k} = -t_k(1-y_k) + \frac{1}{1-y_k}y_k(1-y_k) - \frac{t_k}{1-y_k}y_k(1-y_k)$$

Simplifying again...

$$\frac{\partial E(\mathbf{w})}{\partial a_k} = -t_k(1-y_k) + y_k - t_k y_k$$

$$\frac{\partial E(\mathbf{w})}{\partial a_k} = -t_k + t_k y_k + y_k - t_k y_k$$

$$\frac{\partial E(\mathbf{w})}{\partial a_k} = -t_k + y_k$$

This is the same as Eq. 5.18:

$$\frac{\partial E}{\partial a_k} = y_k - t_k$$

5: PRML Pg. 285, Ex. 5.7

Answer:

The cross-entropy error function Eq. 5.24 is as follows, where the N examples are each denoted n, and the K output nodes are each denoted k:

$$E(\mathbf{w}) = -\sum_{n=1}^{N}\sum_{k=1}^{K} t_{nk} \ln y_k(\mathbf{x}_n, \mathbf{w})$$

For any training example n, the error function is given as follows:

$$E(\mathbf{w}) = -\sum_{k=1}^{K} t_{nk} \ln y_k(\mathbf{x}_n, \mathbf{w})$$

Differentiating this with respect to the output activation a_i, we can pull the target t_{nk} out of the derivative since it is independent of a_i.

$$\frac{\partial E(\mathbf{w})}{\partial a_i} = -\sum_{k=1}^{K} t_{nk} \frac{\partial}{\partial a_i} \ln y_k(\mathbf{x}_n, \mathbf{w})$$

Applying the chain rule and the fact that $\frac{d}{dx}\ln x = \frac{1}{x}$, we find the following:

$$\frac{\partial E(\mathbf{w})}{\partial a_i} = -\sum_{k=1}^{K} t_{nk} \frac{1}{y_k(\mathbf{x}_n, \mathbf{w})} \frac{\partial}{\partial a_i} y_k(\mathbf{x}_n, \mathbf{w})$$

By Eq. 5.25, the softmax function is given as follows:

$$y_k(\mathbf{x}, \mathbf{w}) = \frac{\exp(a_k(\mathbf{x}, \mathbf{w}))}{\sum_j \exp(a_j(\mathbf{x}, \mathbf{w}))}$$

Taking the derivative of softmax requires use of the product rule and chain rule, where the function is rewritten as follows:

$$y_k(\mathbf{x},\mathbf{w}) = \exp(a_k(\mathbf{x},\mathbf{w}))(\sum_j \exp(a_j(\mathbf{x},\mathbf{w})))^{-1}$$

Let's first do the case $k = i$. First, the product rule:

$$\frac{\partial y_k(\mathbf{x},\mathbf{w})}{\partial a_k} = \frac{\partial}{\partial a_k}[\exp(a_k(\mathbf{x},\mathbf{w}))](\sum_j \exp(a_j(\mathbf{x},\mathbf{w})))^{-1} + \exp(a_k(\mathbf{x},\mathbf{w}))\frac{\partial}{\partial a_k}[(\sum_j \exp(a_j(\mathbf{x},\mathbf{w})))^{-1}]$$

For the first term, recall that $\frac{\partial}{\partial x}e^x = e^x$:

$$\frac{\partial y_k(\mathbf{x},\mathbf{w})}{\partial a_k} = \exp(a_k(\mathbf{x},\mathbf{w}))(\sum_j \exp(a_j(\mathbf{x},\mathbf{w})))^{-1} + \exp(a_k(\mathbf{x},\mathbf{w}))\frac{\partial}{\partial a_k}[(\sum_j \exp(a_j(\mathbf{x},\mathbf{w})))^{-1}]$$

For the second term, use the chain rule:

$$\frac{\partial y_k(\mathbf{x},\mathbf{w})}{\partial a_k} = \exp(a_k(\mathbf{x},\mathbf{w}))(\sum_j \exp(a_j(\mathbf{x},\mathbf{w})))^{-1} - \exp(a_k(\mathbf{x},\mathbf{w}))(\sum_j \exp(a_j(\mathbf{x},\mathbf{w})))^{-2}\frac{\partial}{\partial a_k}[\sum_j \exp(a_j(\mathbf{x},\mathbf{w})]$$

The derivative in the second term is zero for all values of j except $j = k$, in which case again use $\frac{\partial}{\partial x}e^x = e^x$:

$$\frac{\partial y_k(\mathbf{x},\mathbf{w})}{\partial a_k} = \exp(a_k(\mathbf{x},\mathbf{w}))(\sum_j \exp(a_j(\mathbf{x},\mathbf{w})))^{-1} - \exp(a_k(\mathbf{x},\mathbf{w}))(\sum_j \exp(a_j(\mathbf{x},\mathbf{w})))^{-2}\exp(a_k(\mathbf{x},\mathbf{w}))$$

Rewriting this derivative and simplifying...

$$\frac{\partial y_k(\mathbf{x},\mathbf{w})}{\partial a_k} = \frac{\exp(a_k(\mathbf{x},\mathbf{w}))}{\sum_j \exp(a_j(\mathbf{x},\mathbf{w}))} - \frac{\exp(a_k(\mathbf{x},\mathbf{w}))^2}{(\sum_j \exp(a_j(\mathbf{x},\mathbf{w})))^2}$$

Substituting in the definition of the softmax (Eq. 5.25), we find the following for $k = i$:

$$\frac{\partial y_k(\mathbf{x},\mathbf{w})}{\partial a_k} = y_k(\mathbf{x},\mathbf{w}) - y_k(\mathbf{x},\mathbf{w})^2$$

Above, we found that the derivative of the error with respect to activation a_i is as follows:

$$\frac{\partial E(\mathbf{w})}{\partial a_i} = -t_{nk}\sum_{k=1}^{K}\frac{1}{y_k(\mathbf{x}_n,\mathbf{w})}\frac{\partial}{\partial a_i}y_k(\mathbf{x}_n,\mathbf{w})$$

Now substituting the derivative of the softmax function for $k = i$...

$$\frac{\partial E(\mathbf{w})}{\partial a_k} = -t_{nk}\frac{1}{y_i(\mathbf{x}_n,\mathbf{w})}(y_i(\mathbf{x},\mathbf{w}) - y_i(\mathbf{x},\mathbf{w})^2) - \sum_{k\neq i} t_{nk}\frac{1}{y_k(\mathbf{x}_n,\mathbf{w})}\frac{\partial}{\partial a_i}y_k(\mathbf{x}_n,\mathbf{w})$$

Now we must take the derivative of the softmax function when $k \neq i$:

$$\frac{\partial y_k(\mathbf{x},\mathbf{w})}{\partial a_i} = \frac{\partial}{\partial a_i}\exp(a_k(\mathbf{x},\mathbf{w}))(\sum_j \exp(a_j(\mathbf{x},\mathbf{w})))^{-1}$$

We can pull $\exp(a_k(\mathbf{x},\mathbf{w}))$ out of the derivative, because $k \neq i$:

$$\frac{\partial y_k(\mathbf{x},\mathbf{w})}{\partial a_i} = \exp(a_k(\mathbf{x},\mathbf{w}))\frac{\partial}{\partial a_i}(\sum_j \exp(a_j(\mathbf{x},\mathbf{w})))^{-1}$$

Applying the chain rule:

$$\frac{\partial y_k(\mathbf{x},\mathbf{w})}{\partial a_i} = -\exp(a_k(\mathbf{x},\mathbf{w}))(\sum_j \exp(a_j(\mathbf{x},\mathbf{w})))^{-2}\frac{\partial}{\partial a_i}\sum_j \exp(a_j(\mathbf{x},\mathbf{w}))$$

The derivative is only non-zero for one term: $i = j$. All other terms are zero.

$$\frac{\partial y_k(\mathbf{x},\mathbf{w})}{\partial a_i} = -\exp(a_k(\mathbf{x},\mathbf{w}))(\sum_j \exp(a_j(\mathbf{x},\mathbf{w})))^{-2}\exp(a_i(\mathbf{x},\mathbf{w}))$$

Rewriting...

$$\frac{\partial y_k(\mathbf{x},\mathbf{w})}{\partial a_i} = -\frac{\exp(a_k(\mathbf{x},\mathbf{w}))\exp(a_i(\mathbf{x},\mathbf{w}))}{(\sum_j \exp(a_j(\mathbf{x},\mathbf{w})))^2}$$

Substituting in the definition of the softmax (Eq. 5.25), we find the following:

$$\frac{\partial y_k(\mathbf{x},\mathbf{w})}{\partial a_i} = -y_k(\mathbf{x}_n,\mathbf{w})y_i(\mathbf{x}_n,\mathbf{w})$$

We can substitute this softmax derivative for $k \neq i$ into the error derivative, given again here:

$$\frac{\partial E(\mathbf{w})}{\partial a_k} = -t_{ni}\frac{1}{y_i(\mathbf{x}_n,\mathbf{w})}(y_i(\mathbf{x}_n,\mathbf{w}) - y_i(\mathbf{x}_n,\mathbf{w})^2) - \sum_{k\neq i} t_{nk}\frac{1}{y_k(\mathbf{x}_n,\mathbf{w})}\frac{\partial}{\partial a_i}y_k(\mathbf{x}_n,\mathbf{w})$$

$$\frac{\partial E(\mathbf{w})}{\partial a_k} = -t_{ni}\frac{1}{y_i(\mathbf{x}_n,\mathbf{w})}(y_i(\mathbf{x}_n,\mathbf{w}) - y_i(\mathbf{x}_n,\mathbf{w})^2) - \sum_{k\neq i} t_{nk}\frac{1}{y_k(\mathbf{x}_n,\mathbf{w})}(-y_k(\mathbf{x}_n,\mathbf{w})y_i(\mathbf{x}_n,\mathbf{w}))$$

We can cancel the y_k components of the second term and move y_i out of the sum, as y_i is independent of the sum over $k \neq i$.

$$\frac{\partial E(\mathbf{w})}{\partial a_k} = -t_{ni}\frac{1}{y_i(\mathbf{x}_n,\mathbf{w})}(y_i(\mathbf{x}_n,\mathbf{w}) - y_i(\mathbf{x}_n,\mathbf{w})^2) + y_i(\mathbf{x}_n,\mathbf{w})\sum_{k\neq i} t_{nk}$$

Simplifying...

$$\frac{\partial E(\mathbf{w})}{\partial a_k} = -t_{ni}(1 - y_i(\mathbf{x}_n,\mathbf{w})) + y_i(\mathbf{x}_n,\mathbf{w})\sum_{k\neq i} t_{nk}$$

$$\frac{\partial E(\mathbf{w})}{\partial a_k} = -t_{ni} + y_i(\mathbf{x}_n, \mathbf{w})t_{ni} + y_i(\mathbf{x}_n, \mathbf{w}) \sum_{k \neq i} t_{nk}$$

$$\frac{\partial E(\mathbf{w})}{\partial a_k} = -t_{ni} + y_i(\mathbf{x}_n, \mathbf{w}) \sum_{k} t_{nk}$$

By definition, the sum over all K target outputs must equal 1, as $\mathbf{t}_n$ is a probability vector:

$$\frac{\partial E(\mathbf{w})}{\partial a_k} = -t_{ni} + y_i(\mathbf{x}_n, \mathbf{w})$$

For any training example n and output node $i = k$, this is the same as Eq. 5.18:

$$\frac{\partial E}{\partial a_k} = y_k - t_k$$

6: PRML Pg. 285, Ex. 5.8

Answer:

$$\tanh(a) = \frac{e^a - e^{-a}}{e^a + e^{-a}}$$

$$\frac{d \tanh(a)}{da} = \frac{d}{da}[(e^a - e^{-a})(e^a + e^{-a})^{-1}]$$

By the product rule...

$$\frac{d \tanh(a)}{da} = \frac{d}{da}[(e^a - e^{-a})](e^a + e^{-a})^{-1} + (e^a - e^{-a})\frac{d}{da}[(e^a + e^{-a})^{-1}]$$

The first term can be evaluated using $\frac{d}{dx}e^x = e^x$ and the chain rule. The second term requires the chain rule as well.

$$\frac{d \tanh(a)}{da} = (e^a - (-e^{-a}))(e^a + e^{-a})^{-1} + (e^a - e^{-a})(-1)(e^a + e^{-a})^{-2}\frac{d}{da}[e^a + e^{-a}]$$

The second term can be evaluated using $\frac{d}{dx}e^x = e^x$ and the chain rule.

$$\frac{d \tanh(a)}{da} = (e^a + e^{-a})(e^a + e^{-a})^{-1} - (e^a - e^{-a})(e^a + e^{-a})^{-2}(e^a - e^{-a})$$

Simplifying...

$$\frac{d \tanh(a)}{da} = 1 - \frac{(e^a - e^{-a})^2}{(e^a + e^{-a})^2}$$

Substituting in the definition of hyperbolic tangent, we find the following:

$$\frac{d \tanh(a)}{da} = 1 - \tanh^2(a)$$

7: PRML Pg. 286, Ex. 5.18

Answer:

Goal is to find $\frac{\partial E_n}{\partial w_{ki}^{(s)}}$.

If we have skip-layer connections going directly from input to output, then the output of a node k is the following:

$$y_k = \sum_{j=0}^{M} w_{kj}^{(2)} z_j + \sum_{i=0}^{D} w_{ki}^{(s)} x_i \tag{2}$$

In the above expression, we have assumed that the activation function of the output layer is linear, i.e. $y_k = a_k$. The first term component $w_{kj}^{(2)}$ is the weight going from the jth hidden layer node to the kth output node, where there are M total hidden layer nodes. The second term component $w_{ki}^{(s)}$ is the skip-layer weight going from the ith input node to the kth output node, where there are D total input layer nodes, and it is assumed that each has skip-layer connections to each of the output nodes. In addition, the hidden layer node j's output z_j is defined as follows:

$$z_j = \tanh(a_j)$$

$$a_j = \sum_{i=0}^{D} w_{ji}^{(1)} x_i$$

where $w_{ji}^{(1)}$ is the weight going from the ith input layer node to the jth hidden layer node, and x_i is the ith input.

A sum-of-squares error for the nth example is defined as follows:

$$E_n = \frac{1}{2} \sum_{k=1}^{K} (y_k - t_k)^2$$

where there are K total output nodes.

Applying Eq. 5.50, the derivative we want to find is given as follows:

$$\frac{\partial E_n}{\partial w_{ki}^{(s)}} = \frac{\partial E_n}{\partial y_k} \frac{\partial y_k}{\partial w_{ki}^{(s)}}$$

From Eq. 5.51, $\delta_k \equiv \frac{\partial E_n}{\partial y_k}$

And from Eq. 5.54, $\delta_k = y_k - t_k$

Also, Eq. 5.54 follows from a straightforward derivative of the error function given above.

Given Eq. 2 (number refers to this document, not PRML)...

$$\frac{\partial y_k}{\partial w_{ki}^{(s)}} = x_i$$

The error we want to find is, therefore, the following:

$$\frac{\partial E_n}{\partial w_{ki}^{(s)}} = \delta_k x_i$$

$$\frac{\partial E_n}{\partial w_{ki}^{(s)}} = (y_k - t_k)x_i$$