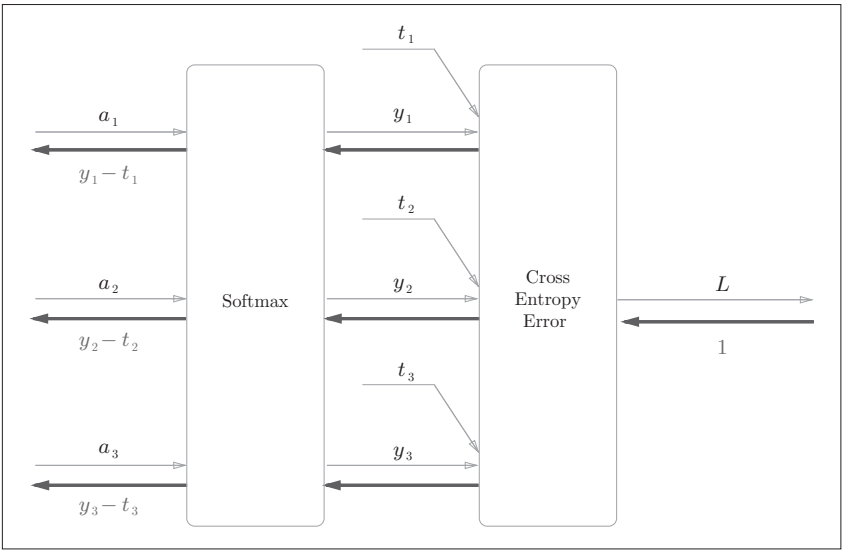
附录A
Softmax–with–Loss层的计算图

这里，我们给出softmax函数和交叉熵误差的计算图，来求它们的反向传播。softmax函数称为softmax层，交叉熵误差称为Cross Entropy Error层，两者的组合称为Softmax-with-Loss层。先来看一下结果，Softmax-with-Loss层可以画成图A-1所示的计算图。



图A-1　Softmax-with-Loss层的计算图

图A-1的计算图中假定了一个进行3类别分类的神经网络。从前面的层输入的是(a_1, a_2, a_3)，softmax层输出(y_1, y_2, y_3)。此外，教师标签是(t_1, t_2, t_3)，Cross Entropy Error层输出损失L。

如图A-1所示，在本附录中，Softmac-with-Loss层的反向传播的结果为$(y_1 - t_1, y_2 - t_2, y_3 - t_3)$。

A.1 正向传播

图A-1的计算图中省略了Softmax层和Cross Entropy Error层的内容。这里，我们来画出这两个层的内容。

首先是Softmax层。softmax函数可由下式表示。

$$y_k = \frac{\exp(a_k)}{\sum_{i=1}^{n} \exp(a_i)} \tag{A.1}$$

因此，用计算图表示Softmax层的话，则如图A-2所示。

图A-2的计算图中，指数的和（相当于式(A.1)的分母）简写为S，最终的输出记为(y_1, y_2, y_3)。

接下来是Cross Entropy Error层。交叉熵误差可由下式表示。

$$L = -\sum_{k} t_k \log y_k \tag{A.2}$$

根据式(A.2)，Cross Entropy Error层的计算图可以画成图A-3那样。

图A-3的计算图很直观地表示出了式(A.2)，所以应该没有特别难的地方。

下一节，我们将看一下反向传播。

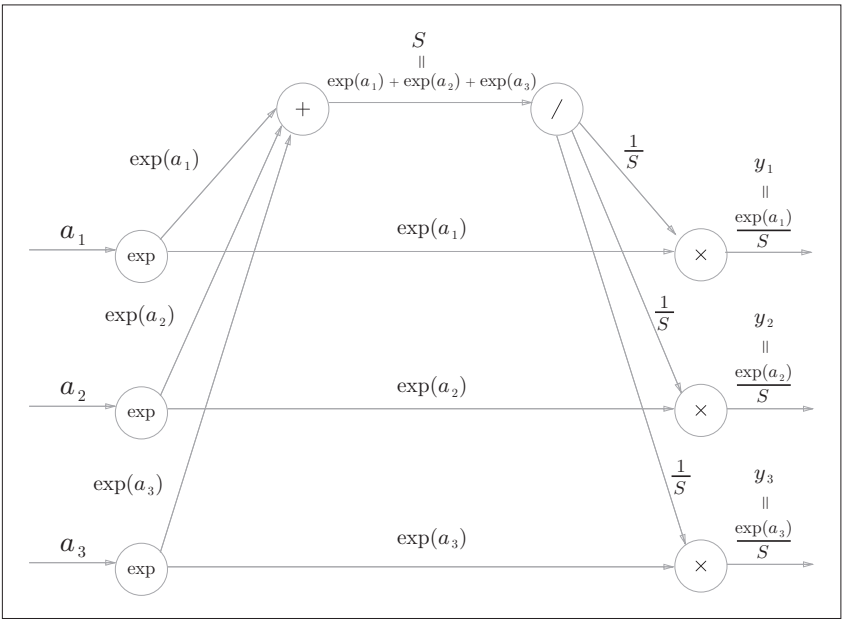


图 A-2 Softmax层的计算图（仅正向传播）

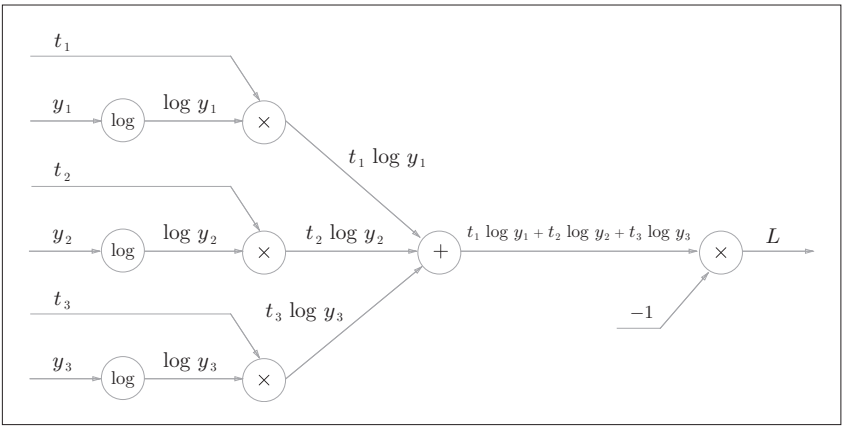
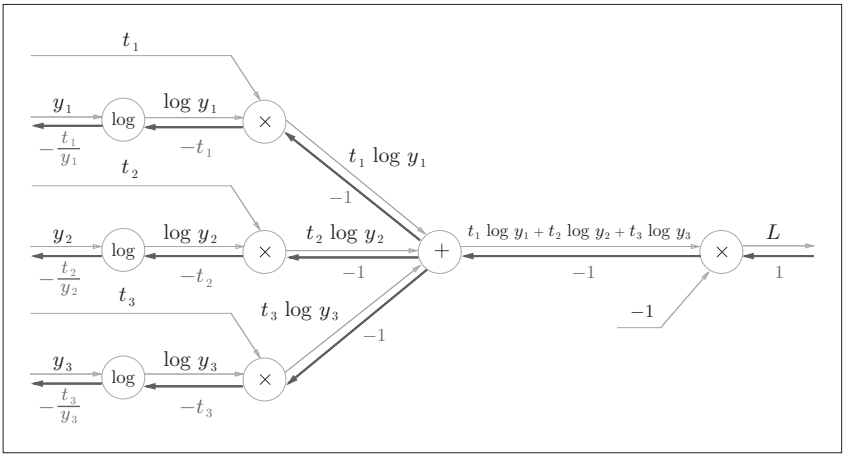


图 A-3 Cross Entropy Error层的计算图（仅正向传播）

A.2 反向传播

首先是Cross Entropy Error层的反向传播。Cross Entropy Error层的反向传播可以画成图A-4那样。



图A-4 交叉熵误差的反向传播

求这个计算图的反向传播时，要注意下面几点。

- 反向传播的初始值（图A-4中最右边的值）是1（因为 $\frac{\partial L}{\partial L} = 1$ ）。
- “×”节点的反向传播将正向传播时的输入值翻转，乘以上游传过来的导数后，再传给下游。
- “+”节点将上游传来的导数原封不动地传给下游。
- “log”节点的反向传播遵从下式。

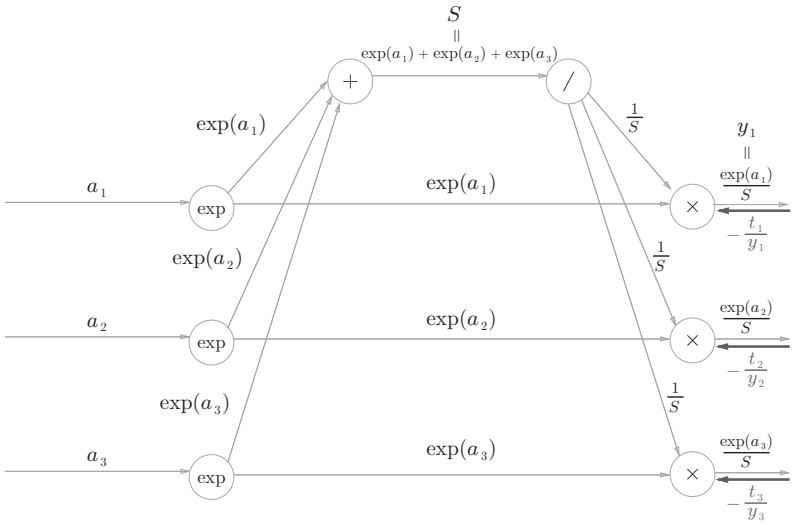
$$y = \log x$$

$$\frac{\partial y}{\partial x} = \frac{1}{x}$$

遵从以上几点，就可以轻松求得Cross Entropy Error的反向传播。结果$\left(-\frac{t_1}{y_1}, -\frac{t_2}{y_2}, -\frac{t_3}{y_3}\right)$是传给Softmax层的反向传播的输入。

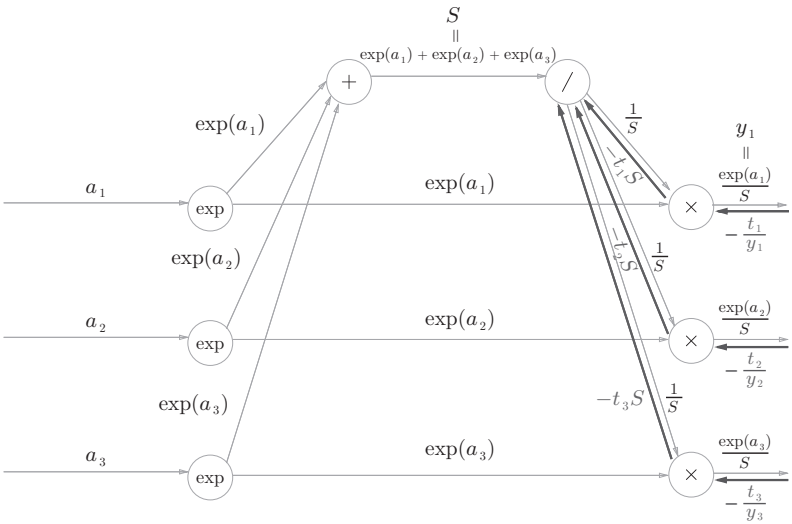
下面是Softmax层的反向传播的步骤。因为Softmax层有些复杂，所以我们来逐一进行确认。

步骤1



前面的层（Cross Entropy Error层）的反向传播的值传过来。

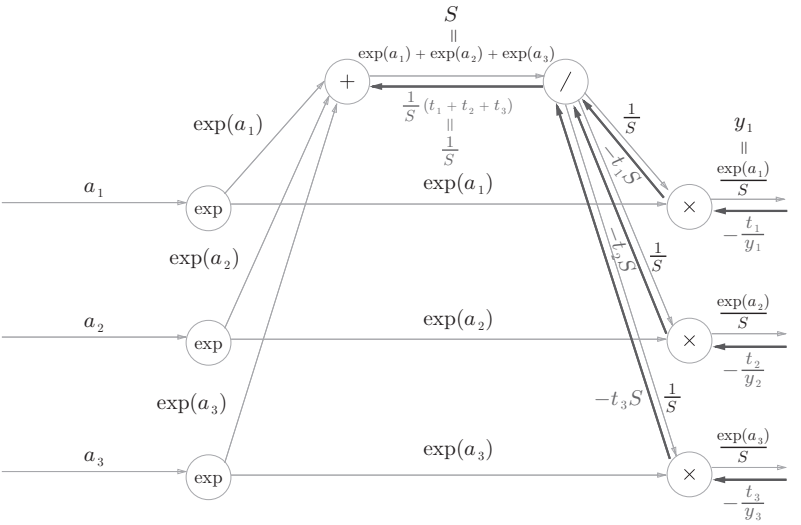
步骤2



“×”节点将正向传播的值翻转后相乘。这个过程中会进行下面的计算。

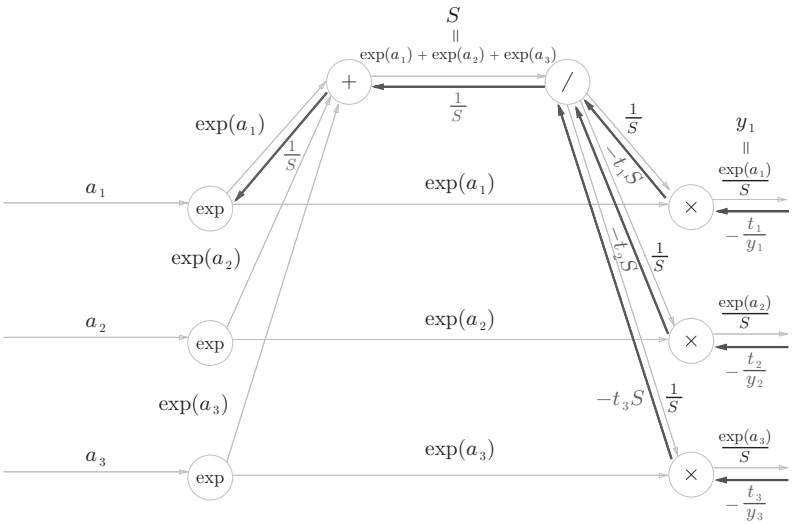
$$-\frac{t_1}{y_1}\exp(a_1) = -t_1\frac{S}{\exp(a_1)}\exp(a_1) = -t_1 S \tag{A.3}$$

步骤3



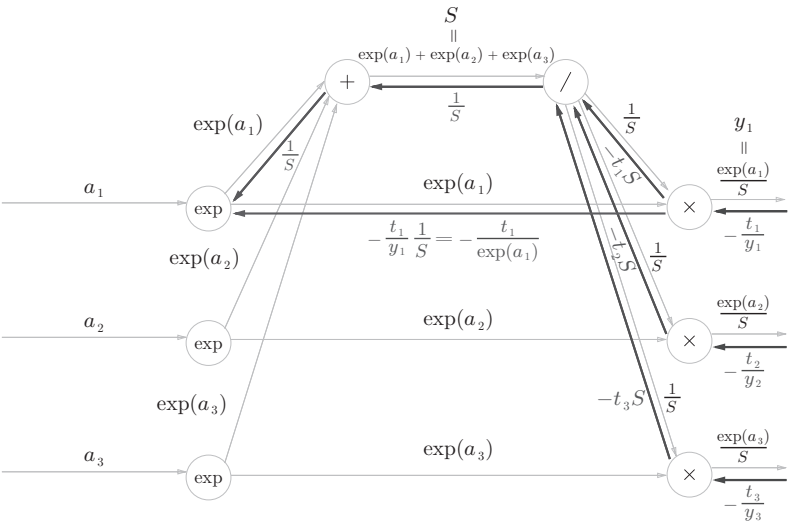
正向传播时若有分支流出，则反向传播时它们的反向传播的值会相加。因此，这里分成了三支的反向传播的值$(-t_1S, -t_2S, -t_3S)$会被求和。然后，还要对这个相加后的值进行“/”节点的反向传播，结果为$\frac{1}{S}(t_1 + t_2 + t_3)$。这里，$(t_1, t_2, t_3)$是教师标签，也是one-hot向量。one-hot向量意味着(t_1, t_2, t_3)中只有一个元素是1，其余都是0。因此，(t_1, t_2, t_3)的和为1。

步骤4



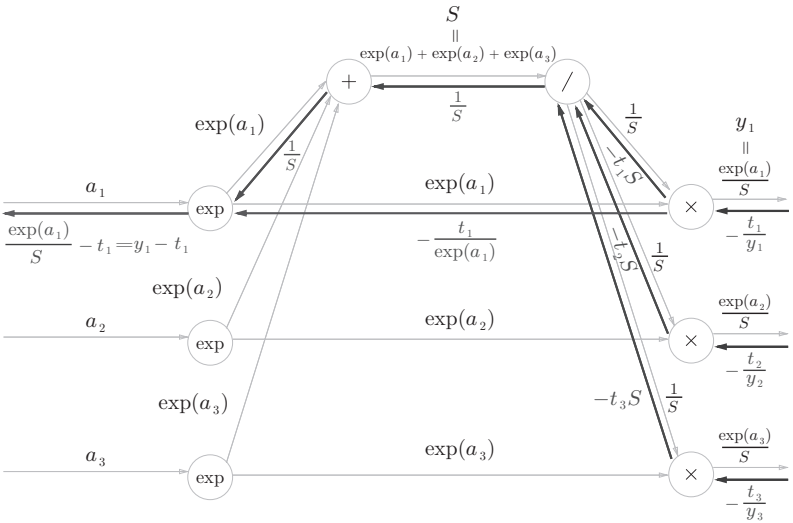
“+”节点原封不动地传递上游的值。

步骤 5



“×”节点将值翻转后相乘。这里，式子变形时使用了 $y_1 = \frac{\exp(a_1)}{S}$。

步骤6



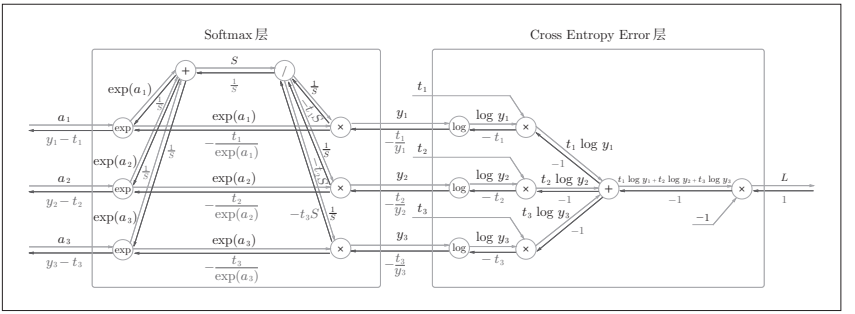
“exp”节点中有下面的关系式成立。

$$
\begin{aligned}
y &= \exp(x) \\
\frac{\partial y}{\partial x} &= \exp(x)
\end{aligned}
\tag{A.4}
$$

根据这个式子，向两个分支的输入和乘以$\exp(a_1)$后的值就是我们要求的反向传播。用式子写出来的话，就是$\left(\frac{1}{S}-\frac{t_1}{\exp(a_1)}\right)\exp(a_1)$，整理之后为$y_1-t_1$。综上，我们推导出，正向传播时输入是$a_1$的节点，它的反向传播是$y_1-t_1$。剩下的$a_2$、$a_3$也可以按照相同的步骤求出来(结果分别为$y_2-t_2$和$y_3-t_3$)。此外，除了这里介绍的3类别分类外，对于$n$类别分类的情况，也可以推导出同样的结果。

A.3 小结

上面，我们画出了Softmax-with-Loss层的计算图的全部内容，并求了它的反向传播。未做省略的Softmax-with-Loss层的计算图如图A-5所示。



图A-5 Softmax-with-Loss层的计算图

图A-5的计算图看上去很复杂，但是使用计算图逐个确认的话，求导（反向传播的步骤）也并没有那么复杂。除了这里介绍的Softmax-with-Loss层，遇到其他看上去很难的层（如Batch Normalization层）时，请一定按照这里的步骤思考一下。相信会比只看数学式更容易理解。