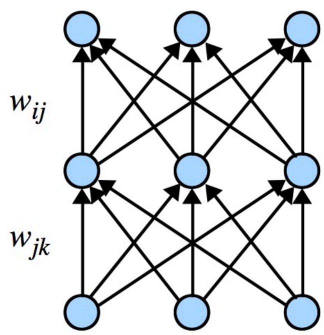
Derivation of the Backpropagation Learning Rule

Notation



"i-units" $\quad a_i$ = activation of output unit i

b_i = bias of output unit i

"j-units" $\quad a_j$ = activation of hidden unit j

b_j = bias of hidden unit j

"k-units" $\quad x_k$ = activation of input unit k

w_{ij} = connection weight from hidden unit j to output unit i

w_{jk} = connection weight from input unit k to hidden unit j

y_i = target value for output unit i

Forward pass: given input pattern $\boldsymbol{x}$, compute output activations

1. Sum of all incoming activity received by hidden unit j: $\quad z_j = \left(\sum_k w_{jk}\, x_k\right) + b_j$
2. Activation of hidden unit j: $\quad a_j = \sigma(z_j) \quad$ where σ is the sigmoid function $\sigma(z) = \dfrac{1}{1+e^{-z}}$
3. Sum of all incoming activity received by output unit i: $\quad z_i = \left(\sum_j w_{ij}\, a_j\right) + b_i$
4. Activation of output unit i: $\quad a_i = \sigma(z_i)$

Quadratic cost function

Cost over all n patterns in the dataset $\{\boldsymbol{x}^{(p)} \rightarrow \boldsymbol{y}^{(p)}\}$, as a function of the weights and biases:

$$C = \frac{1}{n} \sum_p \sum_i \tfrac{1}{2}\,({y_i}^{(p)} - {a_i}^{(p)})^2$$

In the interest of clarity, we will drop the (p) superscripts from ${y_i}^{(p)}$, ${a_i}^{(p)}$, ${x_k}^{(p)}$, etc. from now on, and assume that there is just one input pattern in the dataset ($n = 1$). The generalization to multiple patterns is straightforward, in which case C will just include more terms in the summation.

In general, if F is a function of x, think of $\frac{\partial F}{\partial x}$ as meaning "the influence x has on F". If y depends on x, then x's influence can act "through y"

$$\frac{\partial F}{\partial x} = \frac{\partial y}{\partial x} \times \frac{\partial F}{\partial y} \qquad \text{(chain rule)}$$

Hidden → Output Weights

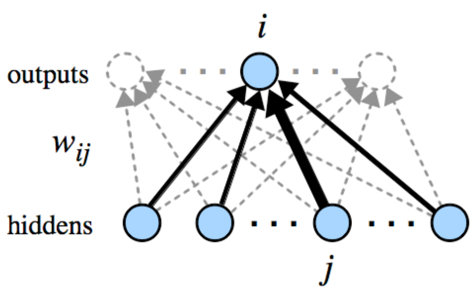
We update each weight so as to move in the opposite direction of the cost gradient:

$$\Delta w_{ij} = -\eta \, \frac{\partial C}{\partial w_{ij}} \qquad \text{where the constant } \eta > 0 \text{ is the learning rate}$$

Calculating $\frac{\partial C}{\partial w_{ij}}$ will give us a learning rule for the hidden → output weights:

$$\frac{\partial C}{\partial w_{ij}} = \frac{\partial z_i}{\partial w_{ij}} \times \frac{\partial a_i}{\partial z_i} \times \frac{\partial C}{\partial a_i}$$

$$\boxed{\begin{array}{c}\text{influence of}\\ w_{ij} \text{ on } C\end{array}} = \boxed{\begin{array}{c}\text{influence of}\\ w_{ij} \text{ on } z_i\end{array}} \times \boxed{\begin{array}{c}\text{influence of}\\ z_i \text{ on } a_i\end{array}} \times \boxed{\begin{array}{c}\text{influence of}\\ a_i \text{ on } C\end{array}}$$



$$z_i = \left(\sum_j w_{ij}\, a_j\right) + b_i \qquad a_i = \sigma(z_i) = \frac{1}{1+e^{-z_i}}$$

$$\frac{\partial z_i}{\partial w_{ij}} = \frac{\partial}{\partial w_{ij}}\left[\left(\sum_j w_{ij}\, a_j\right) + b_i\right] = \frac{\partial}{\partial w_{ij}}\Big[w_{i1}\, a_1 + w_{i2}\, a_2 + \ldots + w_{ij}\, a_j + \ldots + b_i\Big] = \boxed{a_j}$$

$$\frac{\partial a_i}{\partial z_i} = \sigma'(z_i) = \frac{0 \cdot (1+e^{-z_i}) - (-e^{-z_i}) \cdot 1}{(1+e^{-z_i})^2} = \frac{e^{-z_i}}{(1+e^{-z_i})^2} \qquad \text{using the quotient rule}$$

$$= \frac{(1+e^{-z_i}) - 1}{(1+e^{-z_i})^2} = \frac{1}{1+e^{-z_i}} - \frac{1}{(1+e^{-z_i})^2} = \frac{1}{1+e^{-z_i}}\left(1 - \frac{1}{1+e^{-z_i}}\right) = \boxed{a_i\,(1-a_i)}$$

$$C = \sum_i \tfrac{1}{2}\,(y_i - a_i)^2 = \tfrac{1}{2}\,(y_1 - a_1)^2 + \tfrac{1}{2}\,(y_2 - a_2)^2 + \ldots + \tfrac{1}{2}\,(y_i - a_i)^2 + \ldots$$

$$\frac{\partial C}{\partial a_i} = 2 \cdot \tfrac{1}{2}\,(y_i - a_i) \cdot (-1) = \boxed{a_i - y_i}$$

Therefore

$$\frac{\partial C}{\partial w_{ij}} = \frac{\partial z_i}{\partial w_{ij}} \frac{\partial a_i}{\partial z_i} \frac{\partial C}{\partial a_i} = a_j \cdot a_i\,(1-a_i)\cdot(a_i - y_i) = (a_i - y_i)\,a_i\,(1-a_i)\,a_j$$

For convenience, we define $\delta_i = (a_i - y_i)\,a_i\,(1-a_i)$ and rewrite the above equation as

$$\frac{\partial C}{\partial w_{ij}} = \delta_i\,a_j$$

which gives the rule for calculating the weight change for the hidden $\rightarrow$ output weight w_{ij}:

$$\Delta w_{ij} = -\eta\,\frac{\partial C}{\partial w_{ij}} = -\eta\,\delta_i\,a_j$$

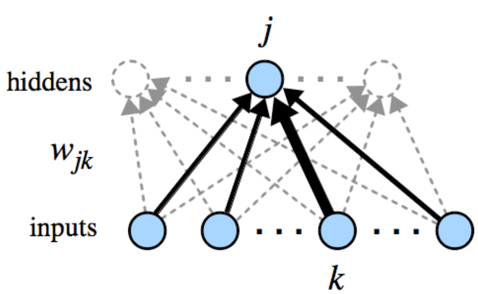
Update Rule for Hidden $\rightarrow$ Output Layer:
$\delta_i = (a_i - y_i)\,a_i\,(1-a_i)$
$\Delta w_{ij} = -\eta\,\delta_i\,a_j \qquad \Delta b_i = -\eta\,\delta_i$

Input $\rightarrow$ Hidden Weights

The learning rule for the input $\rightarrow$ hidden weights is: $\Delta w_{jk} = -\eta\,\dfrac{\partial C}{\partial w_{jk}}$

$$\frac{\partial C}{\partial w_{jk}} = \frac{\partial z_j}{\partial w_{jk}} \times \frac{\partial a_j}{\partial z_j} \times \frac{\partial C}{\partial a_j}$$

$$\boxed{\text{influence of } w_{jk} \text{ on } C} = \boxed{\text{influence of } w_{jk} \text{ on } z_j} \times \boxed{\text{influence of } z_j \text{ on } a_j} \times \boxed{\text{influence of } a_j \text{ on } C}$$

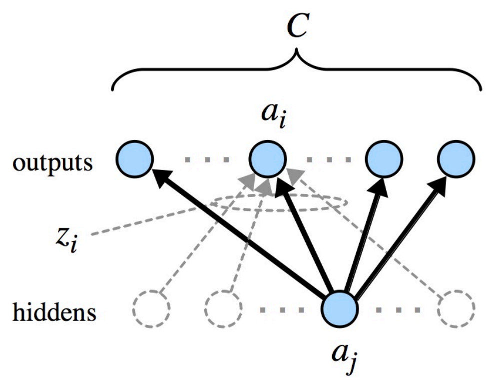


$$z_j = \left(\sum_k w_{jk}\,x_k\right) + b_j \qquad a_j = \sigma(z_j) = \frac{1}{1+e^{-z_j}}$$

$$\frac{\partial z_j}{\partial w_{jk}} = \frac{\partial}{\partial w_{jk}}\left[\left(\sum_k w_{jk}\,x_k\right) + b_j\right] = \frac{\partial}{\partial w_{jk}}\left[w_{j1}\,x_1 + w_{j2}\,x_2 + \ldots + w_{jk}\,x_k + \ldots + b_j\right] = \boxed{x_k}$$

$$\frac{\partial a_j}{\partial z_j} = \sigma'(z_j) = \boxed{a_j\,(1-a_j)}$$

What about $\frac{\partial C}{\partial a_j}$? This is the influence that hidden unit j's activation has on the total cost. This activation feeds into all i-units, each of which influences the cost C:



$$\frac{\partial C}{\partial a_j} = \sum_i \frac{\partial z_i}{\partial a_j} \times \frac{\partial a_i}{\partial z_i} \times \frac{\partial C}{\partial a_i} = \sum_i \boxed{\text{influence of } a_j \text{ on } z_i} \times \boxed{\text{influence of } z_i \text{ on } a_i} \times \boxed{\text{influence of } a_i \text{ on } C}$$

We already calculated $\frac{\partial a_i}{\partial z_i}$ and $\frac{\partial C}{\partial a_i}$ earlier: $\frac{\partial a_i}{\partial z_i} = \boxed{a_i\,(1-a_i)}$ $\frac{\partial C}{\partial a_i} = \boxed{a_i - y_i}$

So all that remains is to calculate $\frac{\partial z_i}{\partial a_j}$, using the fact that $z_i = \left(\sum_j w_{ij}\,a_j\right) + b_i$

$$\frac{\partial z_i}{\partial a_j} = \frac{\partial}{\partial a_j}\left[\left(\sum_j w_{ij}\,a_j\right) + b_i\right] = \frac{\partial}{\partial a_j}\left[w_{i1}\,a_1 + w_{i2}\,a_2 + \ldots + w_{ij}\,a_j + \ldots + b_i\right] = \boxed{w_{ij}}$$

Therefore

$$\frac{\partial C}{\partial a_j} = \sum_i \frac{\partial z_i}{\partial a_j}\,\frac{\partial a_i}{\partial z_i}\,\frac{\partial C}{\partial a_i} = \sum_i w_{ij} \cdot a_i\,(1-a_i) \cdot (a_i - y_i) = \sum_i w_{ij}\,(a_i - y_i)\,a_i\,(1-a_i)$$

which, using our earlier definition of $\delta_i = (a_i - y_i)\,a_i\,(1-a_i)$, we can rewrite as

$$\frac{\partial C}{\partial a_j} = \boxed{\sum_i w_{ij}\,\delta_i}$$

We now have all of the pieces needed to complete our calculation of $\frac{\partial C}{\partial w_{jk}}$, that is, the influence of the input $\rightarrow$ hidden weight w_{jk} on the total cost C:

$$\frac{\partial C}{\partial w_{jk}} = \frac{\partial z_j}{\partial w_{jk}} \times \frac{\partial a_j}{\partial z_j} \times \frac{\partial C}{\partial a_j}$$

$$\boxed{\text{influence of } w_{jk} \text{ on } C} = \boxed{x_k} \times \boxed{a_j\,(1-a_j)} \times \boxed{\sum_i w_{ij}\,\delta_i}$$

In summary,

$$\frac{\partial C}{\partial w_{jk}} = x_k \, a_j \, (1 - a_j) \left(\sum_i w_{ij} \, \delta_i \right) = \left(\sum_i w_{ij} \, \delta_i \right) a_j \, (1 - a_j) \, x_k$$

For convenience, we define $\delta_j = \left(\sum_i w_{ij} \, \delta_i \right) a_j \, (1 - a_j)$ and rewrite the above equation as

$$\frac{\partial C}{\partial w_{jk}} = \delta_j \, x_k$$

which gives the rule for calculating the weight change for the input → hidden weight w_{jk}:

$$\Delta w_{jk} = -\eta \, \frac{\partial C}{\partial w_{jk}} = -\eta \, \delta_j \, x_k$$

Update Rule for Input → Hidden Layer:

$$\delta_j = \left(\sum_i w_{ij} \, \delta_i \right) a_j \, (1 - a_j)$$

$$\Delta w_{jk} = -\eta \, \delta_j \, x_k \qquad \Delta b_j = -\eta \, \delta_j$$

Backward pass: given output activations, backpropagate error and update weights

1. Compute delta value for each output unit i: $\delta_i = (a_i - y_i) \, a_i \, (1 - a_i)$
2. Compute delta value for each hidden unit j: $\delta_j = \left(\sum_i w_{ij} \, \delta_i \right) a_j \, (1 - a_j)$
3. Compute weight and bias changes for hidden → output layer: $\Delta w_{ij} = -\eta \, \delta_i \, a_j \quad \Delta b_i = -\eta \, \delta_i$
4. Compute weight and bias changes for input → hidden layer: $\Delta w_{jk} = -\eta \, \delta_j \, x_k \quad \Delta b_j = -\eta \, \delta_j$
5. Update hidden → output weights and biases: $w_{ij} = w_{ij} + \Delta w_{ij} \qquad b_i = b_i + \Delta b_i$
6. Update input → hidden weights and biases: $w_{jk} = w_{jk} + \Delta w_{jk} \qquad b_j = b_j + \Delta b_j$

Momentum

The momentum parameter $0 \leq \alpha \leq 1$ controls how much the previous weight/bias change at time $t - 1$ contributes to the current change at time t (these equations replace steps 3 and 4 above):

$\Delta w_{ij}(t) = -\eta \, \delta_i \, a_j + \alpha \, \Delta w_{ij}(t-1)$ for the hidden → output weights

$\Delta w_{jk}(t) = -\eta \, \delta_j \, x_k + \alpha \, \Delta w_{jk}(t-1)$ for the input → hidden weights

$\Delta b_i(t) = -\eta \, \delta_i + \alpha \, \Delta b_i(t-1)$ for the output unit biases

$\Delta b_j(t) = -\eta \, \delta_j + \alpha \, \Delta b_j(t-1)$ for the hidden unit biases