notation

E is error ($\equiv$ loss)
o is output, of a neuron, after activation
s is sum of weight * underlying layer outputs, before activation
w is weight
o^2 is output of second layer
o_i^2 is output of node i in layer 2
w_{ji}^2 is weight from node j in layer 1 to node i in layer 2
layers are arranged as: 0 is input layer, then layer 1, layer 2 etc
$a(x)$ is activation function
y_i^* is label i, ie the ground truth for node i, in final output layer
i_i is input to node i in input layer

overall

$$\frac{\partial E}{\partial w_{ji}^{l-1}} = \frac{\partial E}{\partial o_i}\frac{\partial o_i}{\partial s_i}\frac{\partial s_i}{\partial w_{ji}}$$

$$= \frac{\partial \text{loss}}{\partial o_i^l}\frac{\partial \text{activation}}{\partial s_i^l}o_j^{l-1}$$

$$= \frac{\partial \text{loss}}{\partial s_i^l}o_j^{l-1}$$

Recursion:

$$\frac{\partial E}{\partial s_i^{l-1}} = \sum_k \frac{\partial E}{\partial s_k^l}\frac{\partial s_k^l}{\partial o_i^{l-1}}\frac{\partial o_i^{l-1}}{\partial s_i^{l-1}}$$

$$= \frac{\partial \text{activation}^{l-1}}{\partial s_i^{l-1}}\sum_k (\text{error from above})w_{ik}^l$$

loss

Squared error

$$E = \sum_i \frac{1}{2}(o_i - y_i^*)^2$$

$$\frac{\partial E}{\partial o_i} = o_i - y_i^*$$

Cross-entropy

$$E = -\sum_i (y_i^* \log o_i + (1 - y_i^*)\log(1 - o_i))$$

$$\frac{\partial E}{\partial o_i} = \frac{o_i - y_i^*}{o_i(1 - o_i)}$$

Multinomial cross-entropy

$$E = -\sum_i y_i^* \log o_i$$

$$\frac{\partial E}{\partial o_i} = -\frac{y_i^*}{o_i}$$

activation

sigmoid

$$o_i = \sigma(s_i)$$

$$\frac{\partial o_i}{\partial s_i} = o_i(1 - o_i)$$

tanh

$$o_i = \tanh(s_i)$$

$$\frac{\partial o_i}{\partial s_i} = 1 - (o_i)^2$$

relu

linear

softmax

scratch

backpropagation

$$y^*(t) = f[x(t)] + \epsilon$$

log likelihood:

$$\log L = \sum_t -\frac{1}{2}||y^*(t) - y(t)||^2$$

For output unit:

$$\frac{\partial \log L(n)}{\partial w_{ij}(n)} = \sum_t \frac{\partial \log L(t)}{\partial y_i(t)} \frac{\partial y_i(t)}{\partial s_i(t)} \frac{\partial s_i(t)}{\partial w_{ij}(n)}$$

For hidden unit:

$$\frac{\partial \log L(n)}{\partial w_{ij}(n)} = \sum_t \frac{\partial \log L(t)}{\partial h_i(t)} \frac{\partial h_i(t)}{\partial s_i(t)} \frac{\partial s_i(t)}{\partial w_{ij}(n)}$$

For output units:

$$\frac{\partial L(t)}{\partial y_i(t)} = y_i^*(t) - y_i(t)$$

For linear output units:

$$y_i(t) = s_i(t) = \sum_j w_{ij}(n) h_j(t)$$

$$\frac{\partial y_i(t}{\partial s_i(t)} = 1$$

$$\frac{\partial s_i(t)}{\partial w_{ij}(n)} = h_j(t)$$

$$w_{ij}(n+1) = w_{ij}(n) - \alpha \frac{\partial \log L(n)}{\partial w_{ij}(n)}$$

For hidden unit:

$$\frac{\partial \log L(t)}{\partial w_{ij}(t)} = \sum_k \frac{\partial \log L(t)}{\partial s_k(t)} w_{ki}(n)$$

sigmoid

- based on http://www.ics.uci.edu/~pjsadows/backpropderivation.pdf

cross entropy for single sample is

$$E = -\sum_k (y_k^* \log(o_k) + (1 - y_k^*) \log(1 - o_k))$$

Activation function is logistic:

$$o_i = \frac{1}{1 + \exp(-s_i)}$$

where s_i is sum at a node:

$$s_i = \sum_j (w_{ji} o_j)$$

we want:

$$\frac{\partial E}{\partial w_{ji}}$$

$$= \frac{\partial E}{\partial o_i} \frac{\partial o_i}{\partial s_i} \frac{\partial s_i}{\partial w_{ji}}$$

$$\frac{\partial E}{o_i} = -\frac{y_i^*}{o_i} + \frac{1 - y_i^*}{1 - o_i}$$

$$= -\frac{y_i^* - y_i^* o_i - o_i + y_i^* o_i}{o_i(1 - o_i)}$$

$$= \frac{o_i - y_i^*}{o_i(1 - o_i)}$$

$$\frac{\partial o_i}{\partial s_i} = o_i(1 - o_i)$$

$$\frac{\partial E}{\partial s_i} = o_i - y_i^*$$

$$\frac{\partial E}{\partial s_i^{l-1}} = \sum_k \frac{\partial E}{\partial s_k^l} \frac{\partial s_k^l}{\partial o_i^{l-1}} \frac{\partial o_i^{l-1}}{\partial s_i^{l-1}}$$

$$\frac{\partial E}{\partial s_i^{l-2}} = \sum_k \sum_j \frac{\partial E}{\partial s_k^l} \frac{\partial s_k^l}{\partial o_j^{l-1}} \frac{\partial o_j^{l-1}}{\partial s_j^{l-1}} \frac{\partial s_j^{l-1}}{\partial o_i^{l-2}} \frac{\partial o_i^{l-2}}{\partial s_i^{l-2}}$$

$$= \sum_k \sum_j \frac{\partial E}{\partial s_k^l} w_{kj}^l \frac{\partial o_j^{l-1}}{\partial s_j^{l-1}} w_{ji}^{l-1} \frac{\partial o_i^{l-2}}{\partial s_i^{l-2}}$$

classification

class has multinomial distribution:

$$p(y^*(t)|x(t)) = \prod_{k=1}^{K} y_k(t)^{y_k^*(t)}$$

log likelihood:

$$\log L = \sum_t \sum_k y_k^*(t) \log y_k(t)$$

output units use softmax activation:

$$y_i(t) = \frac{\exp[s_i(t)]}{\sum_k \exp[s_k(t)]}$$

$$\frac{\partial \log L(n)}{\partial w_{ij}(n)} = \sum_t \frac{\partial \log L(t)}{\partial s_i(t)} \frac{\partial s_i(t)}{\partial w_{ij}(n)}$$

$$\frac{\partial \log L(t)}{\partial s_i(t)} = \sum_k \frac{\partial \log L(t)}{\partial y_k(t)} \frac{\partial y_k(t)}{\partial s_i(t)}$$

$$= \sum_k \frac{y_k^*(t)}{y_k(t)} y_k(t)(\delta_{ik} - y_i(t))$$

$$\frac{\partial s_i(t)}{\partial w_{ij}(n)} = h_j(t)$$

Softmax:

$$y_i = \frac{\exp s_{n,i}}{\sum_k \exp s_{n,k}}$$

$$\frac{\exp(s_{n,i})/\exp(\max_j s_{n,j})}{\sum_k \exp(s_{n,k})/\exp(\max_j s_{n,j})}$$

$$= \frac{\exp(s_{n,i} - \max_j(s_{n,j}))}{\sum_k \exp(s_{n,k} - \max_j(s_{n,j}))}$$

Cross entropy for softmax (?):

$$= -\sum_n \sum_k y_{n,k}^* \log y_{n,k}$$

where $y^*_{n,k}$ should be 0, except for $y^*_{n,l_n} = 1$, where l_n is the label of instance n

sigmoid, mse (I'm trying myself, might not be correct...)

$$L = \sum_k \frac{1}{2}(y^*_k - o_k)^2$$

$$o_i = \frac{1}{1 + \exp(-s_i)}$$

$$s_i = \sum_k w_{ki} o_k$$

$$\frac{\partial L}{\partial w_{ji}} = \frac{\partial L}{\partial o_i} \frac{\partial o_i}{\partial s_i} \frac{\partial s_i}{\partial w_{ji}}$$

$$\frac{\partial L}{\partial o_i} = 2\frac{1}{2}(y^*_i - o_i)(-1)$$

$$= (o_i - y^*_i)$$

$$\frac{\partial o_i}{\partial s_i} = o_i(1 - o_i)$$

$$\frac{\partial s_i}{\partial w_{ji}} = o_j$$

$$\frac{\partial L}{\partial w_{ji}} = (o_k - y^*_k) o_i (1 - o_i) o_j$$

We want to reduce L slightly.

- therefore, we should modify w_{ji} slightly:

$$w_{ji}(n+1) = w_{ji}(n) - \alpha \frac{\partial L}{\partial w_{ji}}$$

$$= w_{ji}(n) - \alpha (o_i - y^*_i) o_i (1 - o_i) o_j$$

$$= w_{ji}(n) - \alpha \text{error}(o_i, y^*_i) \text{derivative}(o_i) \text{output}^{l-1}(o^{l-1}_j)$$

$$\frac{\partial L}{\partial w^{l-1}_{kj}} = \sum_i \frac{\partial L}{\partial o^l_i} \frac{\partial o^l_i}{\partial s^l_i} \frac{\partial s^l_i}{o^{l-1}_j} \frac{\partial o^{l-1}_j}{\partial s^{l-1}_j} \frac{\partial s^{l-1}_j}{\partial w^{l-1}_{kj}}$$

$$s_i = \sum_j o^{l-1}_j w^l_{ji}$$

$$\frac{\partial s^l_i}{o^{l-1}_j} = w^l_{ji}$$

So,

$$\frac{\partial L}{\partial w^{l-1}_{kj}} = \sum_i (o^l_i - y^*_i) o^l_i (1 - o^l_i) w^l_{ji} o^{l-1}_j (1 - o^{l-1}_j) o^{l-2}_k$$

$$= o^{l-1}_j (1 - o^{l-1}_j) o^{l-2}_k \sum_i (o^l_i - y^*_i) o^l_i (1 - o^l_i) w^l_{ji}$$

$$= (\sum_i \text{error}^l \text{deriv}(o_i^l) w_{ji}^l) \text{deriv}(o_j^{l-1}) \text{output}(o_k^{l-2})$$

$$= \text{apparent error}_j^{l-1}(o_i^l, w_{ji}^l, y_i^*) \text{deriv}^{l-1}(o_j^{l-1}) \text{output}^{l-2}(o_k^{l-2})$$

where

$$\text{apparent error}_j^{l-1}(o_i^l, w_{ji}^l, y_i^*) = \sum_i (o_i^l - y_i^*) o_i^l (1 - o_i^l) w_{ji}^l$$

$$= \sum_i (\text{error}_i^l(o_i^l, y_i^*) \text{deriv}_i^l(o_i^l) w_{ji}^l)$$

$$= \sum_i (\frac{\partial L}{\partial o_i^l} \frac{\partial o_i^l}{\partial s_i^l} w_{ji}^l)$$

$$= \sum_i (\frac{\partial L}{\partial o_i^l} \frac{\partial o_i^l}{\partial s_i^l} \frac{\partial s_i^l}{\partial o_{ji}^{l-1}})$$

$$= \sum_i \frac{\partial L}{\partial o_{ji}^{l-1}}$$

$$= \sum_i \text{apparent error}_i^l \frac{\partial o_i^l}{\partial o_{ji}^{l-1}}$$

$$= \sum_i \text{apparent error}_i^l \frac{\partial o_i^l}{\partial s_i^l} \frac{\partial s_i^l}{\partial o_{ji}^{l-1}}$$

$$= \sum_i \text{apparent error}_i^l \frac{\partial o_i^l}{\partial s_i^l} w_{ji}^l$$

eg

$$\frac{\partial o_i^l}{\partial s_i^l} \frac{\partial s_i^l}{\partial o_{ji}^{l-1}} = o_i^l (1 - o_i^l) w_{ji}^l$$

or, for tanh

$$\frac{\partial o_i^l}{\partial s_i^l} \frac{\partial s_i^l}{\partial o_{ji}^{l-1}} = (1 - (o_i^l)^2) w_{ji}^l$$

Error functions

Cross entropy:

$$L = -\sum_k (y_k^* \log(o_k) + (1 - y_k^*) \log(1 - o_k))$$

$$\frac{\partial L}{\partial o_i} = -\frac{y_i^*}{o_i} - \frac{(1 - y_i^*)(-1)}{1 - o_i}$$

$$= -\frac{y_i^*}{o_i} + \frac{1 - y_i^*}{1 - o_i}$$

$$= \frac{-y_i^* + y_i^* o_i + o_i - y_i^* o_i}{o_i (1 - o_i}$$

$$= \frac{o_i - y_i^*}{o_i (1 - o_i)}$$

Cross-entropy, multiclass:

$$L = -\sum_i y_i^* \log(o_i)$$

$$\frac{\partial L}{\partial o_i} = -\frac{y_i^*}{o_i}$$

when $j = i$:

$$\frac{\partial o_i}{\partial s_j} = o_i(1 - o_i)$$

when $j! = i$:

$$\frac{\partial o_i}{\partial s_j} = -o_i o_j$$

Generalizing:

$$\frac{\partial o_i}{\partial s_j} = \delta_{i,j} o_i - o_i o_j$$

$$= o_i(\delta_{i,j} - o_j)$$

$$\frac{\partial E}{\partial s_i} = o_i - y_i^*$$

Squared error:

$$L = \sum_k \frac{1}{2}(y_k^* - o_k)^2$$

$$\frac{\partial L}{\partial o_i} = o_i - y_i^*$$

Activation functions

Sigmoid:

$$o_i = \frac{1}{1 + \exp(-s_i)}$$

$$\frac{\partial o_i}{\partial s_i} = o_i(1 - o_i)$$

Linear:

$$o_i = s_i$$

$$\frac{\partial o_i}{\partial s_i} = 1$$

softmax:

$$o_i = \frac{\exp(s_i)}{\sum_k \exp(s_k)}$$

$$\frac{\partial o_i}{\partial s_i} =$$

errors backprop

My current method:

$$\text{error}^{l-1} = \sum_j \text{error}_j^l \frac{\partial o_j^l}{\partial o_i^{l-1}}$$

$$= \sum_j \frac{\partial L}{\partial o_j^l} \frac{\partial o_j^l}{\partial s_j^l} \frac{\partial s_j^l}{\partial o_i^{l-1}}$$

eg

$$= \sum_i \frac{\partial L}{\partial o_i^l} (1 - (o_i^l)^2)) w_{ji}^l$$

$$\frac{\partial L}{\partial w_{ji}^l} = \frac{\partial L}{\partial o_i^l} \frac{\partial o_i^l}{\partial s_i^l} \frac{\partial s_i^l}{\partial w_{ji}^l}$$

eg

$$= \frac{\partial L}{\partial o_i^l} (1 - (o_i^l)^2) o_j^{l-1}$$

Peter Sadowski "notes on backpropagation" method:

$$\frac{\partial L}{\partial s_i^{l-1}} = \sum_j \frac{\partial L}{\partial s_j^l} \frac{\partial s_j^l}{\partial o_i^{l-1}} \frac{\partial o_i^{l-1}}{\partial s_i^{l-1}}$$

eg

$$= \sum_j \frac{\partial L}{\partial s_j^l} w_{ij}^l (1 - (o_i^{l-1})^2)$$

$$\frac{\partial L}{\partial w_{ji}^l} = \frac{\partial L}{\partial s_i^l} \frac{\partial s_i^l}{\partial w_{ji}^l}$$

$$= \frac{\partial L}{\partial s_i^l} o_j^{l-1}$$

numerical validation

$$w_{ji}(n+1) = w_{ji}(n) + \alpha \frac{\partial L}{\partial w_{ji}}$$

So

$$\frac{\partial L}{\partial w_{ji}} = \frac{w_{ji}(n+1) - w_{ji}(n)}{\alpha}$$

$$L(n+1) \approx L(n) + \Delta w_{ji} \frac{\partial L}{\partial w_{ji}}$$

$$\Delta(w_{ji}) = \alpha \frac{\partial L}{\partial w_{ji}}$$

Therefore,

$$\frac{\partial L}{\partial w_{ji}} = \frac{\Delta w_{ji}}{\alpha}$$

And so,

$$\Delta L \approx \frac{(\Delta w_{ji})^2}{\alpha}$$

errors backprop compared to possible layer configurations
using $\frac{\partial E}{\partial s_i^l}$:

example configuration:
- expected outputs, y_i^*, $L = \sum_i \frac{1}{2}(y_i^* - o_i^{l=4})^2$
- softmax, $o_i^{l=4}$, $o_i^{l=4} = a(s_1^{l=4}, s_2^{l=4}, .., s_k^{l=4})$, $s_i^{l=4} = o_i^{l=3}$
- fully connected, linear, $o_i^{l=3}$, $s_i^{l=3}$, $w_{ji}^{l=3}$, $o_i^{l=3} = s_i^{l=3}$, $s_i^{l=3} = \sum_k o_k^{l=2} w_{ki}^{l=3}$
- convolutional, relu, $o_i^{l=2}$, $s_i^{l=2}$, $w_{ji}^{l=2}$, $s_i^{l=2} = \sum_k o_k^{l=1} w_{ki}^{l=2}$, $o_i^{l=2} = a(s_i^{l=2})$
- convolutional, relu, $o_i^{l=1}$, $s_i^{l=1}$, $w_{ji}^{l=1}$, $s_i^{l=1} = \sum_k o_k^{l=0} w_{ki}^{l=1}$, $o_i^{l=1} = a(s_i^{l=1})$
- input $o_i^{l=0}$

softmax:

$$\frac{\partial L}{\partial o_i^{l=4}} = o_i^{l=4} - y_i^*$$

$$\frac{\partial o_i^{l=4}}{\partial s_j^{l=4}} = o_i(\delta_{i,j} - o_j)$$

$$\frac{\partial s_i^{l=4}}{\partial o_i^{l=3}} = 1$$

$$\frac{\partial o_i^{l=3}}{\partial s_i^{l=3}} = 1$$

$$\frac{\partial s_i^{l=3}}{\partial w_{ji}^{l=3}} = o_j^{l=2}$$

$$\frac{\partial s_i^{l=3}}{\partial o_j^{l=2}} = w_{ji}^{l=3}$$

$$\frac{\partial o_i^{l=2}}{\partial s_i^{l=2}} =$$

example configuration:
- expected outputs, $y_i^* \rightarrow$ squared error $E = \sum_k \frac{1}{2}(o_k - y_k^*)^2$
- fully connected layer, $o_i^2 = a^2(s_i^2)$, $a^2(x) = \tanh(x)$ $\frac{\partial \tanh(x)}{\partial x} = 1 - (\tanh(x))^2$, $s_i^2 = \sum_k o_k^1 w_{ki}^2$
- fully connected layer, $o_i^1 = a^1(s_i^1)$, $a^1(x) = \tanh(x)$, $s_i^1 = \sum_k o_k^0 w_{ki}^1$
- input layer, $o_i^0 = i_i$

$$\frac{\partial E}{\partial o_i^2} = o_i - y_i^*$$

$$\frac{\partial E}{\partial s_i^2} = (o_i - y_i^*)(1 - (o_i^2)^2)$$

example configuration:
- expected outputs, $y_i^* \rightarrow$ cross-entropy error $E = -\sum_k (y_k^* \log o_k^2 + (1 - y_k^*) \log(1 - o_k^2))$
- fully connected layer, $o_i^2 = a^2(s_i^2)$, $a^2(x) = \tanh(x)$ $\frac{\partial \tanh(x)}{\partial x} = 1 - (\tanh(x))^2$, $s_i^2 = \sum_k o_k^1 w_{ki}^2$

- fully connected layer, $o_i^1 = a^1(s_i^1)$, $a^1(x) = \tanh(x)$, $s_i^1 = \sum_k o_k^0 w_{ki}^1$
- input layer, $o_i^0 = i_i$

$$\frac{\partial E}{\partial o_i^2} = \frac{o_i^2 - y_i^*}{o_i^2(1 - o_i^2)}$$

$$\frac{\partial E}{\partial s_i^2} = \frac{o_i^2 - y_i^*}{o_i^2(1 - o_i^2)}(1 - (o_i^2)^2)$$

example configuration:
- expected outputs, $y_i^* \rightarrow$ cross-entropy error $E = -\sum_k (y_k^* \log o_k^2 + (1 - y_k^*) \log(1 - o_k^2))$
- fully connected layer, $o_i^2 = a^2(s_i^2)$, $a^2(x) = \sigma(x)$ $\frac{\partial \sigma(x)}{\partial x} = \sigma(x)(1 - \sigma(x))$, $s_i^2 = \sum_k o_k^1 w_{ki}^2$
- fully connected layer, $o_i^1 = a^1(s_i^1)$, $a^1(x) = \tanh(x)$, $s_i^1 = \sum_k o_k^0 w_{ki}^1$
- input layer, $o_i^0 = i_i$

$$\frac{\partial E}{\partial o_i^2} = \frac{o_i^2 - y_i^*}{o_i^2(1 - o_i^2)}$$

$$\frac{\partial E}{\partial s_i^2} = \frac{o_i^2 - y_i^*}{o_i^2(1 - o_i^2)} o_i^2(1 - o_i^2) = o_i^2 - y_i^*$$

example configuration:
- expected outputs, y_i^*, $\rightarrow$ cross-entropy error $E = -\sum_k y_k^* \log o_k^2$
- softmax, o_i^4, $o_i^4 = a(s_1^4, s_2^4, .., s_k^4)$, $s_i^{l=4} = o_i^3$
- fully connected, linear, o_i^l, s_i^l, w_{ji}^l, $o_i^l = s_i^l$, $s_i^{l=3} = \sum_k o_k^2 w_{ki}^3$
- convolutional, relu, o_i^l, s_i^l, w_{ji}^l, $s_i^l = \sum_k o_k^1 w_{ki}^2$, $o_i^2 = a(s_i^2)$
- convolutional, relu, o_i^1, s_i^1, w_{ji}^1, $s_i^1 = \sum_k o_k^0 w_{ki}^1$, $o_i^1 = a(s_i^1)$
- input $o_i^0 = i_i$