How the backpropagation algorithm works

Srikumar Ramalingam

School of Computing

University of Utah

Reference

Most of the slides are taken from the second chapter of the online book by Michael Nielson:

- neuralnetworksanddeeplearning.com

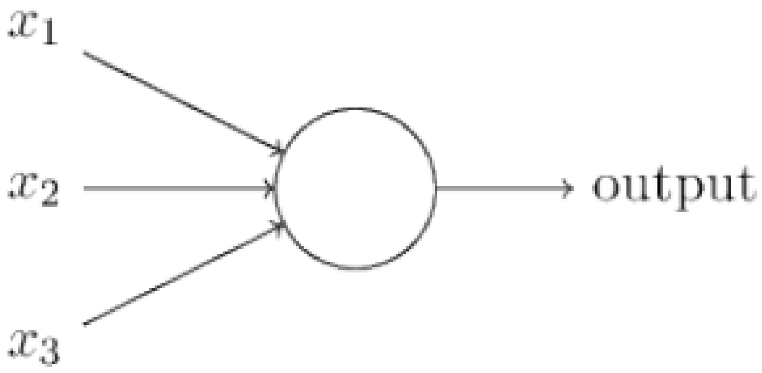
Introduction

- First discovered in 1970.
- First influential paper in 1986:

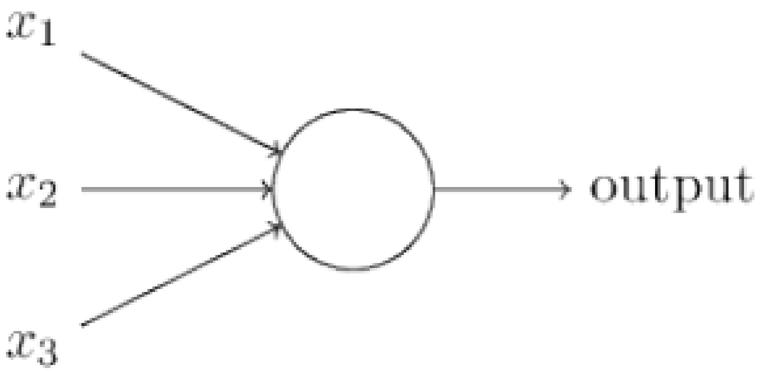
Rumelhart, Hinton and Williams, Learning representations by back-propagating errors, Nature, 1986.

Perceptron (Reminder)

$$\text{output} = \begin{cases} 0 & \text{if } w \cdot x + b \leq 0 \\ 1 & \text{if } w \cdot x + b > 0 \end{cases}$$



Sigmoid neuron (Reminder)

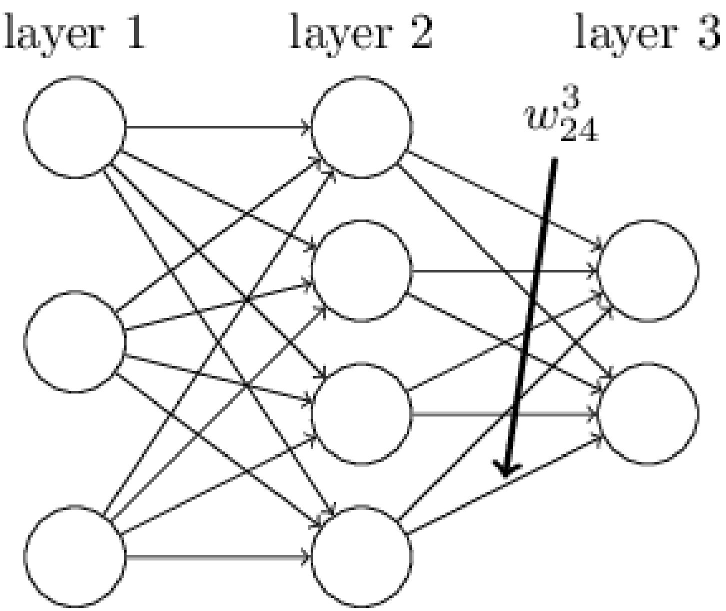


- A sigmoid neuron can take real numbers (x_1, x_2, x_3) within 0 to 1 and returns a number within 0 to 1. The weights (w_1, w_2, w_3) and the bias term b are real numbers.

Sigmoid function $\sigma(z) \equiv \frac{1}{1+e^{-z}}$

$\sigma(0) = 0.5,$
$\sigma(-\infty) = 0,$
$\sigma(\infty) = 1$

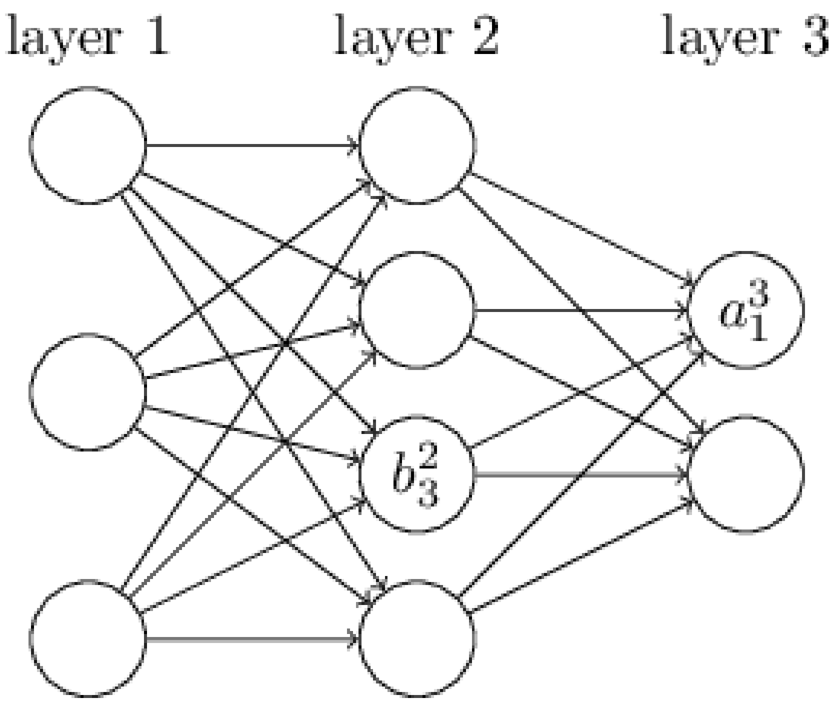
Matrix equations for neural networks



w^l_{jk} is the weight from the k^{th} neuron in the $(l-1)^{\text{th}}$ layer to the j^{th} neuron in the l^{th} layer

- The indices j and k seem a little counter-intuitive!
- Notations are used in this manner to enable matrix multiplications.

Layer to layer relationship



$$a_j^l = \sigma(z_j^l)$$

$$z_j^l = \sum_k w_{jk}^l a_k^{l-1} + b_j^l$$

$$a_j^l = \sigma\left(\sum_k w_{jk}^l a_k^{l-1} + b_j^l\right)$$

Examples:

$$a_1^3 = \sigma(z_1^3), a_2^3 = \sigma(z_2^3)$$

$$z_j^3 = \sum_{k=1\ to\ 4} w_{jk}^3 a_k^2 + b_j^3\ , j \in \{1,2\}$$

$$z_j^2 = \sum_{k=1\ to\ 4} w_{jk}^2 a_k^1 + b_j^2, j \in \{1,..,4\}$$

- b_j^l is the bias term in the j_{th} neuron in the l_{th} layer.
- a_j^l is the activation in the j_{th} neuron in the l_{th} layer.
- z_j^l is the weighted input to the j_{th} neuron in the l_{th} layer.

Cost function from the network

Groundtruth for each input

Output activation vector for a specific training sample x.

$$C = \frac{1}{2n} \sum_x \|y(x) - a^L(x)\|^2$$

# of input samples

for each input sample

Input vector x

Backpropagation and stochastic gradient descent

- The goal of the backpropagation algorithm is to compute the gradients $\frac{\partial C}{\partial w}$ and $\frac{\partial C}{\partial b}$ of the cost function C with respect to each and every weight and bias parameters. Note that backpropagation is only used to compute the gradients.

$$C = \frac{1}{2n} \sum_x \|y(x) - a^L(x)\|^2$$

- Stochastic gradient descent is the training algorithm.

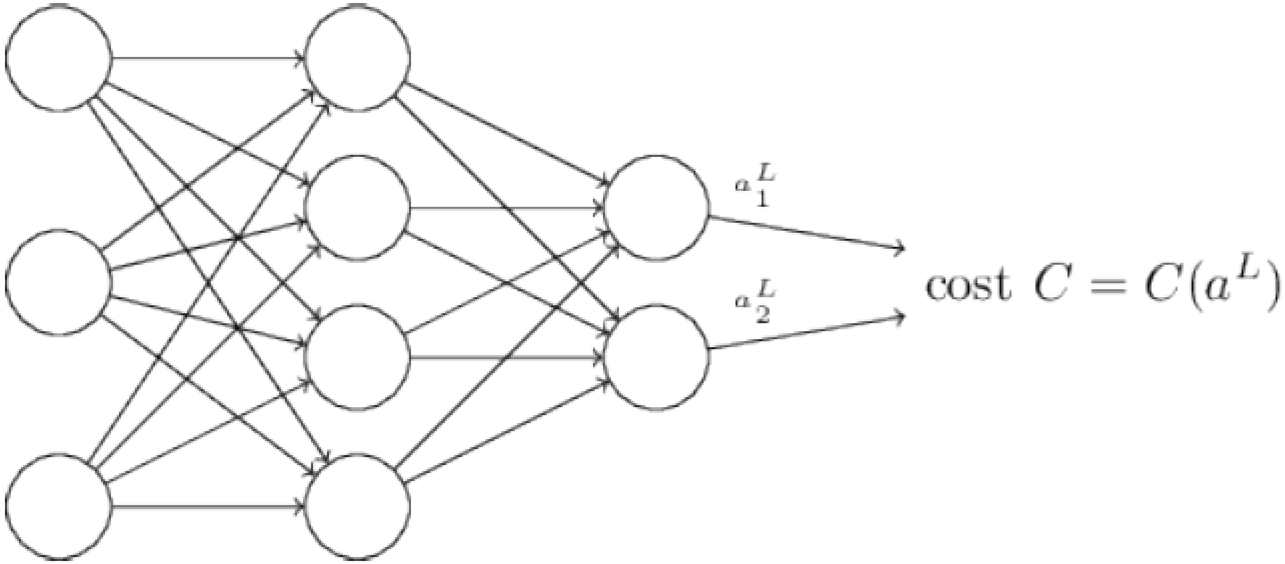
Assumptions on the cost function

1. We assume that the cost function can be written as the average over the cost functions from individual training samples: $C = \frac{1}{n}\sum_x C_x$. The cost function for the individual training sample is given by $C_x = \frac{1}{2}|y(x) - a^L(x)|^2$.

- why do we need this assumption? Backpropagation will only allow us to compute the gradients with respect to a single training sample as given by $\frac{\partial C_x}{\partial w}$ and $\frac{\partial C_x}{\partial b}$. We then recover $\frac{\partial C}{\partial w}$ and $\frac{\partial C}{\partial b}$ by averaging the gradients from the different training samples.

Assumptions on the cost function (continued)

2. We assume that the cost function can be written as a function of the output from the neural network. We assume that the input x and its associated correct labeling $y(x)$ are fixed and treated as constants.



Hadamard product

- Let s and t are two vectors. The Hadamard product is given by:

$$s \odot t$$

$$(s \odot t)_j = s_j t_j$$

$$E.g., \begin{bmatrix}1\\2\\3\end{bmatrix} \odot \begin{bmatrix}2\\2\\2\end{bmatrix} = \begin{bmatrix}2\\4\\6\end{bmatrix}$$

$$\begin{bmatrix}1\\2\end{bmatrix} \odot \begin{bmatrix}3\\4\end{bmatrix} = \begin{bmatrix}1*3\\2*4\end{bmatrix} = \begin{bmatrix}3\\8\end{bmatrix}$$

Such elementwise multiplication is also referred to as schur product.

Backpropagation

- Our goal is to compute the partial derivatives $\frac{\partial C}{\partial w_{jk}^l}$ and $\frac{\partial C}{\partial b_j^l}$.
- We compute some intermediate quantities while doing so:

$$\delta_j^l = \frac{\partial C}{\partial z_j^l}$$

Four equations of the BP (backpropagation)

Summary: the equations of backpropagation (L is the total number of layers)

1) $\delta_j^L = \frac{\partial C}{\partial a_j^L} \frac{\partial a_j^L}{\partial z_j^L}$ $\qquad BP1$

2) $\delta_j^l = \sum_k (w_{kj}^{l+1} \delta_k^{l+1}) \sigma'(z_j^l)$ $\qquad BP2$

3) $\frac{\partial C}{\partial b_j^l} = \delta_j^l$ $\qquad BP3$

4) $\frac{\partial C}{\partial w_{jk}^l} = a_k^{l-1} \delta_j^l$ $\qquad BP4$

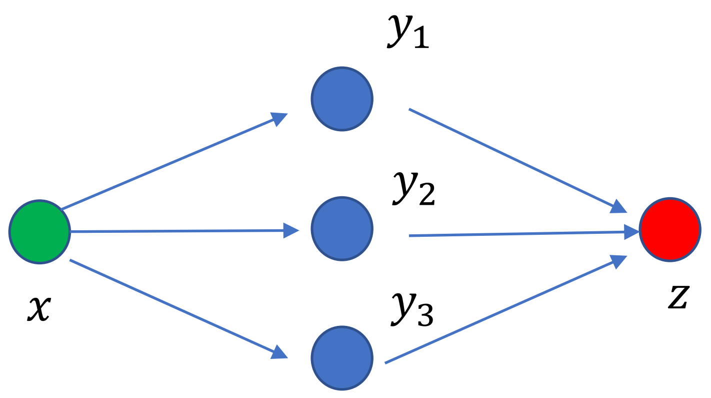
Chain Rule in differentiation

- In order to differentiate a function $\mathrm{z} = f\big(g(x)\big)$ w.r.t x, we can do the following:

Let $\mathrm{y} = g(x), \;\; z = f(y), \; \frac{dz}{dx} = \frac{dz}{dy} \times \frac{dy}{dx}$

Chain Rule in differentiation (computation graph)

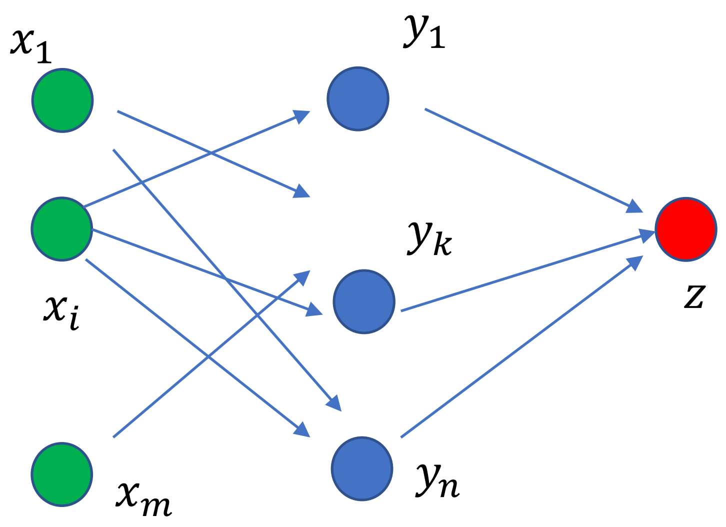
$$\frac{\partial z}{\partial x} = \sum_{\substack{j: x \in Parent(y_j), \\ y_j \in Ancestor\,(z)}} \frac{\partial z}{\partial y_j} \frac{\partial y_j}{\partial x}$$



Chain Rule in differentiation (vector case)

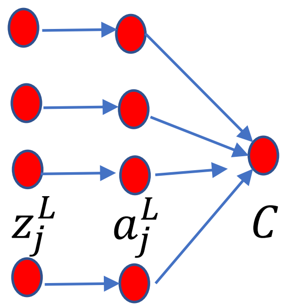
Let $x \in \mathbb{R}^m$, $y \in \mathbb{R}^n$, g maps from $\mathbb{R}^m$ to $\mathbb{R}^n$, and f maps from $\mathbb{R}^n$ to $\mathbb{R}$. If $y = g(x)$ and $z = f(y)$, then

$$\frac{\partial z}{\partial x_i} = \sum_k \frac{\partial z}{\partial y_k} \frac{\partial y_k}{\partial x_i}$$



BP1

$$C = \frac{1}{2n} \sum_x \|y(x) - a^L(x)\|^2$$



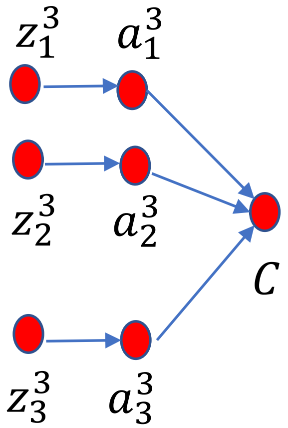
Variable association for applying vector chain rule

To Show: $\boxed{\delta_j^L = \frac{\partial C}{\partial a_j^L} \frac{\partial a_j^L}{\partial z_j^L}}$

Here L is the last layer. We get this result by applying chain rule once.

$$\delta_j^L = \frac{\partial C}{\partial z_j^L} = \frac{\partial C}{\partial a_j^L} \frac{\partial a_j^L}{\partial z_j^L}.$$

Examples for BP1



$$\delta_1^3 = \frac{\partial C}{\partial z_1^3} = \frac{\partial C}{\partial a_1^3}\frac{\partial a_1^3}{\partial z_1^3},$$

$$\delta_2^3 = \frac{\partial C}{\partial z_2^3} = \frac{\partial C}{\partial a_2^3}\frac{\partial a_2^3}{\partial z_2^3},$$

$$\delta_3^3 = \frac{\partial C}{\partial z_3^3} = \frac{\partial C}{\partial a_3^3}\frac{\partial a_3^3}{\partial z_3^3},$$

$$\delta^3 = \frac{\partial C}{\partial a^3} \odot \frac{\partial a^3}{\partial z^3} = \frac{\partial C}{\partial a^3} \odot \sigma'(z^3),$$

$$\delta^L = \frac{\partial C}{\partial a^L} \odot \sigma'(z^L)$$

Derivates of Sigmoid activation function

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

$$\frac{d\sigma(z)}{dz} = \sigma'(z) = (-1)(1+e^{-z})^{-2}(-1)e^{-z}$$

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

$$= \frac{1}{1+e^{-z}}\left(1 - \frac{1}{1+e^{-z}}\right)$$

$$= \sigma(z)(1-\sigma(z))$$

Derivates of quadratic objective function

$$C = \frac{1}{2}|y - a^L|^2 = \frac{1}{2}\left(\left(\mathrm{y_1} - \mathrm{a_1^L}\right)^2 + \left(\mathrm{y_2} - \mathrm{a_2^L}\right)^2 + \cdots + \left(\mathrm{y_n} - \mathrm{a_n^L}\right)^2\right)$$

$$\frac{\partial C}{\partial a_j^L} = \left(y_j - a_j\right)$$

$$\frac{\partial C}{\partial a^L} = \begin{bmatrix} (y_1 - a_1) \\ (y_2 - a_2) \\ \cdot \\ \cdot \\ (y_n - a_n) \end{bmatrix}$$

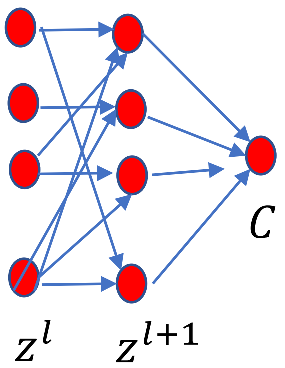
BP2

$$\boxed{\delta_j^l = \sum_k (w_{kj}^{l+1}\delta_k^{l+1})\sigma'(z_j^l)}$$

Proof:

$$\delta_j^l = \frac{\partial C}{\partial z_j^l} = \sum_k \frac{\partial C}{\partial z_k^{l+1}} \frac{\partial z_k^{l+1}}{\partial z_j^l} = \sum_k \frac{\partial z_k^{l+1}}{\partial z_j^l} \delta_k^{l+1}$$

$$z_k^{l+1} = \sum_j w_{kj}^{l+1} a_j^l + b_k^l = \sum_j w_{kj}^{l+1} \sigma(z_j^l) + b_k^l$$



Variable association for applying vector chain rule

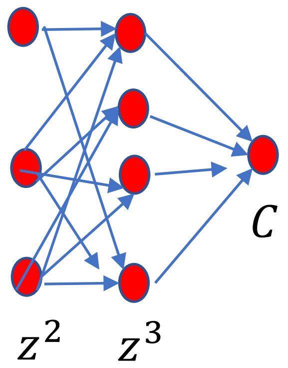
By differentiating we have:

$$\frac{\partial z_k^{l+1}}{\partial z_j^l} = w_{kj}^{l+1}\sigma'(z_j^l)$$

$$\delta_j^l = \sum_k (w_{kj}^{l+1}\delta_k^{l+1})\sigma'(z_j^l)$$

Vectorized notation: $\delta^l = (w^{l+1})^T \delta^{l+1} \odot \sigma'(z^l)$

BP2 Example



Variable association for applying vector chain rule

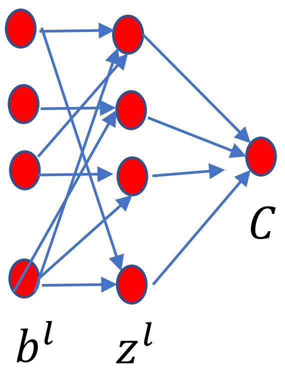
$$\delta_1^2 = \frac{\partial C}{\partial z_1^2} = \sum_{k=1}^{4} \frac{\partial C}{\partial z_k^3} \frac{\partial z_k^3}{\partial z_1^2} = \sum_{k=1}^{4} \delta_k^3 \frac{\partial z_k^3}{\partial z_1^2}$$

$$z_k^3 = \sum_{j=1}^{3} w_{kj}^3 \sigma(z_j^2) + b_k^3 \;, \frac{\partial z_k^3}{\partial z_1^2} = w_{k1}^3 \sigma'(z_1^2)$$

$$\delta_1^2 = \sum_{k=1}^{4} \delta_k^3 \frac{\partial z_k^3}{\partial z_1^2} = \sum_{k=1}^{4} \delta_k^3 w_{k1}^3 \sigma'(z_1^2) = (\delta_1^3 w_{11}^3 + \delta_2^3 w_{21}^3 + \delta_3^3 w_{31}^3 + \delta_4^3 w_{41}^3)\, \sigma'(z_1^2)$$

$$\delta_2^2 = \sum_{k=1}^{4} \delta_k^3 w_{k1}^3 \sigma'(z_2^2) = (\delta_1^3 w_{12}^3 + \delta_2^3 w_{22}^3 + \delta_3^3 w_{32}^3 + \delta_4^3 w_{42}^3)\, \sigma'(z_2^2)$$

BP3



Variable association for applying vector chain rule

$$\frac{\partial C}{\partial b_j^l} = \delta_j^l$$

Proof:

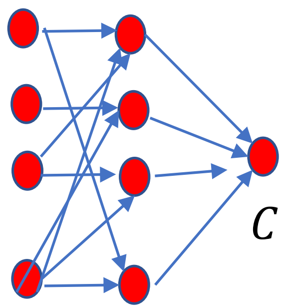
$$\frac{\partial C}{\partial b_j^l} = \sum_k \left(\frac{\partial C}{\partial z_k^l} \frac{\partial z_k^l}{\partial b_j^l} \right) \;\; = \frac{\partial C}{\partial z_j^l} \frac{\partial z_j^l}{\partial b_j^l}, \text{ the other terms } \frac{\partial z_k^l}{\partial b_j^l} \text{ vanish when } \; j \neq k.$$

$$= \delta_j^l \frac{\partial \left(\sum_k w_{jk} a_k^{l-1} + b_j^l \right)}{\partial b_j}$$

$$= \delta_j^l$$

BP3 Example

$$z^3 = w^3 a^2 + b^3$$

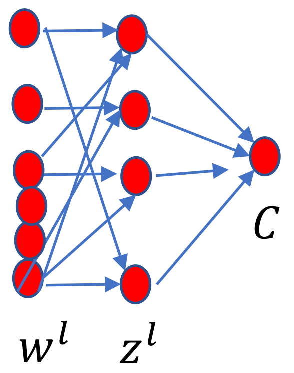


Variable association for
applying vector chain rule

$$\frac{\partial C}{\partial b_1^3} = \delta_1^3 = \sum_{k=1}^{3} \frac{\partial C}{\partial z_k^3} \frac{\partial z_k^3}{\partial b_1^3} = \frac{\partial C}{\partial z_1^3} \frac{\partial z_1^3}{\partial b_1^3}$$

$$= \delta_1^l \, \frac{\partial (\sum_{k=1}^{3} w_{1k}^3 a_k^2 + b_1^3)}{\partial b_1^3} = \delta_1^l \qquad \frac{\partial C}{\partial b^3} = \delta^l$$

BP4



Variable association for applying vector chain rule

$$\frac{\partial C}{\partial w_{jk}^l} = a_k^{l-1}\delta_j^l$$

Proof:

$$\frac{\partial C}{\partial w_{jk}^l} = \sum\nolimits_m \frac{\partial C}{\partial z_m^l}\frac{\partial z_m^l}{\partial w_{jk}^l}$$

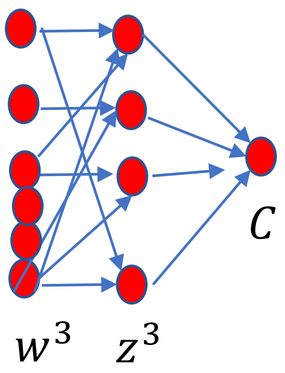
$$= \frac{\partial C}{\partial z_j^l}\ \frac{\partial z_j^l}{\partial w_{jk}} \text{ and the other terms } \frac{\partial z_j^l}{\partial w_{jk}} \text{ when } m \neq j.$$

$$= \delta_j^l \frac{\partial\left(\sum_k w_{jk}^l a_k^{l-1} + b_j^l\right)}{\partial w_{jk}}$$

$$= \delta_j^l\ a_k^{l-1}$$

BP4 Example

$$\frac{\partial C}{\partial w_{12}^3} = a_2^2 \delta_1^3$$



Variable association for applying vector chain rule

Proof:

$$\frac{\partial C}{\partial w_{12}^3} = \sum_m \frac{\partial C}{\partial z_m^3} \frac{\partial z_m^3}{\partial w_{12}^3} = \frac{\partial C}{\partial z_1^3} \frac{\partial z_1^3}{\partial w_{12}^3} = \delta_1^3 \frac{\partial \left(\sum_k w_{12}^3 a_2^2 + b_1^3 \right)}{\partial w_{12}} = \delta_1^3 a_2^2$$

The backpropagation algorithm

1. **Input x:** Set the corresponding activation a^1 for the input layer.

2. **Feedforward:** For each $l = 2, 3, \ldots, L$ compute $z^l = w^l a^{l-1} + b^l$ and $a^l = \sigma(z^l)$.

3. **Output error δ^L:** Compute $$\delta_j^L = \frac{\partial C}{\partial a_j^L} \frac{\partial a_j^L}{\partial z_j^L}.$$

4. **Backpropagate the error:** For each $l = L-1, L-2, \ldots, 2$ compute $$\delta_j^l = \sum_k (w_{kj}^{l+1} \delta_k^{l+1}) \sigma'(z_j^l)$$

5. **Output:** The gradient of the cost function is given by $\frac{\partial C}{\partial w_{jk}^l} = a_k^{l-1} \delta_j^l$ and $\frac{\partial C}{\partial b_j^l} = \delta_j^l$.

The word "backpropagation" comes from the fact that we compute the error vectors δ_j^l in the backward direction.

Gradients using finite differences

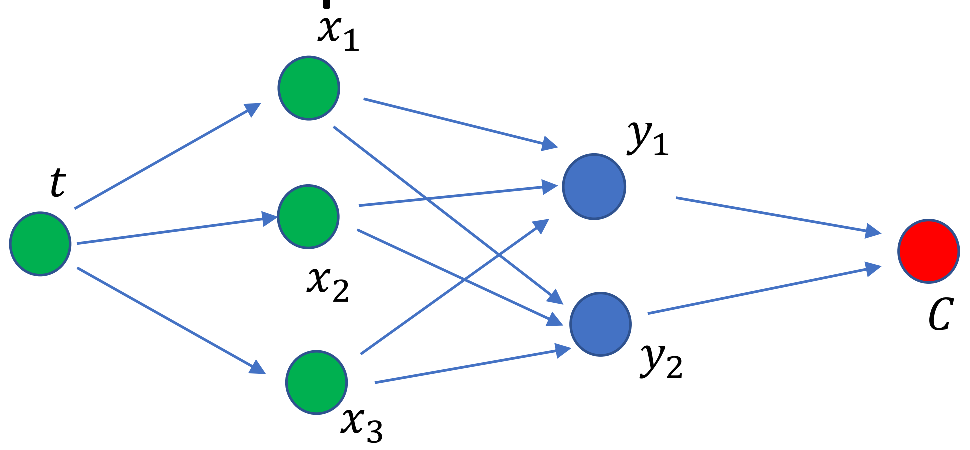
$$\frac{\partial C}{\partial w_j} \approx \frac{C(w + \epsilon e_j) - C(w)}{\epsilon}$$

Here ϵ is a small positive number and e_j is the unit vector in the jth direction.
Conceptually very easy to implement.
In order to compute this derivative w.r.t one parameter, we need to do one forward pass – for millions of variables we will have to do millions of forward passes.

- Backpropagation can get all the gradients in just one forward and backward pass – forward and backward passes are roughly equivalent in computations.

The derivatives using finite differences would be a million times slower!!

BP interpretation



$$\frac{\partial C}{\partial x_i} = \left(\frac{\partial y_1}{\partial x_i}\frac{\partial C}{\partial y_1} + \frac{\partial y_2}{\partial x_i}\frac{\partial C}{\partial y_2}\right), \qquad \frac{\partial C}{\partial t} = (\frac{\partial x_1}{\partial t}\frac{\partial C}{\partial x_1} + \frac{\partial x_2}{\partial t}\frac{\partial C}{\partial x_2} + \frac{\partial x_3}{\partial t}\frac{\partial C}{\partial x_3})$$

$$\frac{\partial C}{\partial t} = (\frac{\partial x_1}{\partial t}\left(\frac{\partial y_1}{\partial x_1}\frac{\partial C}{\partial y_1} + \frac{\partial y_2}{\partial x_1}\frac{\partial C}{\partial y_2}\right) + \frac{\partial x_2}{\partial t}\left(\frac{\partial y_1}{\partial x_2}\frac{\partial C}{\partial y_1} + \frac{\partial y_2}{\partial x_2}\frac{\partial C}{\partial y_2}\right) + \frac{\partial x_3}{\partial t}\left(\frac{\partial y_1}{\partial x_3}\frac{\partial C}{\partial y_1} + \frac{\partial y_2}{\partial x_3}\frac{\partial C}{\partial y_2}\right)) =$$
$$\frac{\partial x_1}{\partial t}\frac{\partial y_1}{\partial x_1}\frac{\partial C}{\partial y_1} + \frac{\partial x_1}{\partial t}\frac{\partial y_2}{\partial x_1}\frac{\partial C}{\partial y_2} + \frac{\partial x_2}{\partial t}\frac{\partial y_1}{\partial x_2}\frac{\partial C}{\partial y_1} + \frac{\partial x_2}{\partial t}\frac{\partial y_2}{\partial x_2}\frac{\partial C}{\partial y_2} + \frac{\partial x_3}{\partial t}\frac{\partial y_1}{\partial x_3}\frac{\partial C}{\partial y_1} + \frac{\partial x_3}{\partial t}\frac{\partial y_2}{\partial x_3}\frac{\partial C}{\partial y_2}$$

Backpropagation – the big picture

$$\Delta C \approx \sum_{mnp\ldots q} \frac{\partial C}{\partial a^L_m} \frac{\partial a^L_m}{\partial a^{L-1}_n} \frac{\partial a^{L-1}_n}{\partial a^{L-2}_p} \ldots \frac{\partial a^{l+1}_q}{\partial a^l_j} \frac{\partial a^l_j}{\partial w^l_{jk}} \Delta w^l_{jk}$$

- To compute the total change in C we need to consider all possible paths from the weight to the cost.

$$\frac{\partial C}{\partial w^l_{jk}} = \sum_{mnp\ldots q} \frac{\partial C}{\partial a^L_m} \frac{\partial a^L_m}{\partial a^{L-1}_n} \frac{\partial a^{L-1}_n}{\partial a^{L-2}_p} \ldots \frac{\partial a^{l+1}_q}{\partial a^l_j} \frac{\partial a^l_j}{\partial w^l_{jk}}$$

- We are computing the rate of change of C w.r.t a weight w.
- Every edge between two neurons in the network is associated with a rate factor that is just the ratio of partial derivatives of one neurons activation with respect to another neurons activation.
- The rate factor for a path is just the product of the rate factors of the edges in the path.
- The total change is the sum of the rate factors of all the paths from the weight to the cost.

Thank You

DERIVATIVE RULES

$$\frac{d}{dx}\left(x^n\right)=nx^{n-1}$$

$$\frac{d}{dx}\left(a^x\right)=\ln a\cdot a^x$$

$$\frac{d}{dx}\left(f(x)\cdot g(x)\right)=f(x)\cdot g'(x)+g(x)\cdot f'(x)$$

$$\frac{d}{dx}\left(\frac{f(x)}{g(x)}\right)=\frac{g(x)\cdot f'(x)-f(x)\cdot g'(x)}{\left(g(x)\right)^2}$$

$$\frac{d}{dx}\left(f(g(x))\right)=f'(g(x))\cdot g'(x)$$

$$\frac{d}{dx}\left(\ln x\right)=\frac{1}{x}$$

$$\frac{d}{dx}\left(\sin x\right)=\cos x$$

$$\frac{d}{dx}\left(\cos x\right)=-\sin x$$

$$\frac{d}{dx}\left(\tan x\right)=\sec^2 x$$

$$\frac{d}{dx}\left(\cot x\right)=-\csc^2 x$$

$$\frac{d}{dx}\left(\sec x\right)=\sec x\tan x$$

$$\frac{d}{dx}\left(\csc x\right)=-\csc x\cot x$$

$$\frac{d}{dx}\left(\arcsin x\right)=\frac{1}{\sqrt{1-x^2}}$$

$$\frac{d}{dx}\left(\arctan x\right)=\frac{1}{1+x^2}$$

$$\frac{d}{dx}\left(\operatorname{arc\,sec} x\right)=\frac{1}{x\sqrt{x^2-1}}$$

$$\frac{d}{dx}\left(\sinh x\right)=\cosh x$$

$$\frac{d}{dx}\left(\cosh x\right)=\sinh x$$

Source: http://math.arizona.edu/~calc/Rules.pdf

INTEGRAL RULES

$$\int x^n dx = \frac{1}{n+1} x^{n+1} + c, \quad n \neq -1$$

$$\int a^x dx = \frac{1}{\ln a} a^x + c$$

$$\int \frac{1}{x} dx = \ln|x| + c$$

$$\int \frac{dx}{\sqrt{1-x^2}} = \arcsin x + c$$

$$\int \frac{dx}{1+x^2} = \arctan x + c$$

$$\int \frac{dx}{x\sqrt{x^2-1}} = \operatorname{arc\,sec} x + c$$

$$\int \sin x dx = -\cos x + c$$

$$\int \cos x dx = \sin x + c$$

$$\int \sec^2 x dx = \tan x + c$$

$$\int \sinh x dx = \cosh x + c$$

$$\int \csc^2 x dx = -\cot x + c$$

$$\int \sec x \tan x dx = \sec x + c$$

$$\int \csc x \cot x dx = -\csc x + c$$

$$\int \cosh x dx = \sinh x + c$$

Source: http://math.arizona.edu/~calc/Rules.pdf