Backpropagation

Recall: The chain rule

$$\frac{d}{dw} C(z(w))$$

$$\frac{dC}{dw} = \frac{dC}{dz}\frac{dz}{dw}$$

$$\frac{d}{dw} C(z_1(w), z_2(w), z_3(w), \ldots)$$

$$\frac{dC}{dw} = \sum_{i=1}^{N} \frac{dC}{dz_i}\frac{dz_i}{dw}$$

Definitions:

L: # of layers

N^m: Dimensionality of layer m

W^m: weight matrix for layer m $\mathbb{R}^{N^m \times N^{m-1}}$

b^m: bias vector for layer m $\mathbb{R}^{N^m}$

σ^m: nonlinearity for layer m

z^m: "preactivation" for layer m $z^m = W^m a^{m-1} + b^m$
"linear mix"

a^m: "activation" for layer m $a^m = \sigma^m(z^m)$

a^0: input to the model (x)

y: Target output

C: Cost (loss) function $C(a^L, y)$

Want $\frac{dC}{dW^m}$ for all m.

Backprop will give $\frac{dC}{dW^m}$ given:

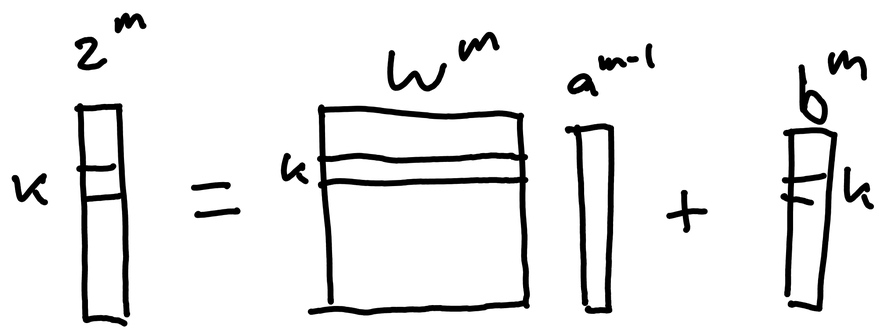
$$\frac{dC}{da^L} \quad \text{and} \quad \frac{da^m}{dz^m}$$

↳ immediately known based on design

$$\frac{dC}{dW^m_{ij}} = \sum_{k=1}^{N^m} \frac{dC}{dz^m_k} \frac{dz^m_k}{dW^m_{ij}}$$

Recall: $z^m_k = \sum_{\ell=1}^{N^m} W^m_{k\ell} a^{m-1}_\ell + b^m_k$

$$z^m = W^m a^{m-1} + b^m$$



if $i \neq k$, $\frac{dz^k_m}{dW^m_{ij}} = 0$

When $i = k$.

$$\frac{dz_i^m}{dw_{ij}^m} = \frac{d}{dw_{ij}^m}\left(\sum_{e=1}^{N^m} w_{ie}^m a_e^{m-1} + b_i^m\right)$$

$$= a_j^{m-1}$$

$$\rightarrow \frac{dz_k^m}{dw_{ij}^m} = \begin{cases} 0, & i \neq k \\ a_j^{m-1}, & i = k \end{cases}$$

$$\rightarrow \frac{dC}{dw_{ij}^m} = \frac{dC}{dz_i^m} a_j^{m-1}$$

$$\rightarrow \frac{dC}{dw} = \frac{dC}{dz^m} a^{m-1\,T}$$

We still need $\frac{dC}{dz^m}$.

For $m = L$, $\frac{dC}{dz_k^L} = \frac{dC}{a_k^L}\frac{da_k^L}{dz_k^L}$

both known

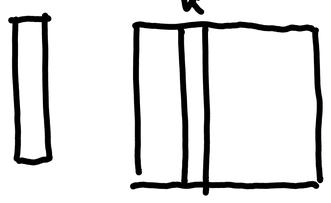
For $m < L$, we have

$$\frac{dC}{dz_k^m} = \frac{dC}{da_k^m}\left(\frac{da_k^m}{dz_k^m}\right) \quad \text{(known)}$$

$$= \left(\sum_{l=1}^{N^{m+1}} \frac{dC}{dz_l^{m+1}}\frac{dz_l^{m+1}}{da_k^m}\right)\frac{da_k^m}{dz_k^m}$$

$$= \left(\sum_{l=1}^{N^{m+1}} \frac{dC}{dz_l^{m+1}}\frac{d}{da_k^m}\left(\sum_{h=1}^{N^m} w_{lh}^{m+1} a_h^m + b_l^{m+1}\right)\right)\frac{da_k^m}{dz_k^m}$$

$$= \left(\sum_{l=1}^{N^{m+1}} \frac{dC}{dz_l^{m+1}} w_{lk}^{m+1}\right)\frac{da_k^m}{dz_k^m}$$



$$\frac{dC}{dz^m} = \left(w^{m+1\,T}\frac{dC}{dz^{m+1}}\right) \circ \frac{da_k}{dz_k}$$

How to compute $\frac{dC}{d \text{ any weight}}$:

1) $\frac{dC}{dz^L} = \frac{dC}{da^L} \frac{da^L}{dz^L}$ (all known)

2) Recursively compute $\frac{dC}{dz^m}$ for $m = L-1, L-2, \ldots$

$$\frac{dC}{dz^m} = \left(W^{m+1^T} \frac{dC}{dz^{m+1}} \right) \circ \frac{da^m}{dz^m}$$

known; previous step; known

3) $\frac{dC}{dW^m} = \frac{dC}{dz^m} a^{m-1^T}$

from step 2; from forward pass

ex $a^m = \frac{1}{1+\exp(-z^m)}$ (all m including $m=L$)

$$C(y, a^L) = (y-1)\log(1-a^L) - y\log(a^L)$$

$$\frac{dC}{dz^L} = \frac{dC}{da^L}\frac{da^L}{dz^L}$$ } deriv of sigmoid

↳ deriv of x-entropy

$$= \left(\frac{a^L - y}{a^L(1-a^L)}\right) a^L(1-a^L)$$

$$= a^L - y$$

$$\frac{dC}{dz^{L-1}} = \left(w^{L^T}\frac{dC}{dz^L}\right) \circ \frac{da^{L-1}}{dz^{L-1}}$$

$$w^{L^T}(a^L - y) \circ a^{L-1}(1-a^{L-1})$$

$$\frac{dC}{dw^{L-1}} = \frac{dC}{dz^{L-1}} a^{L-2^T}$$

$$= w^{L^T}(a^L - y) \circ a^{L-1}(1-a^{L-1})\, a^{L-2^T}$$

$$\frac{dC}{dz_m} = \left(W^{m+1^T} \frac{dC}{dz^{m+1}}\right) \circ \frac{da^m}{dz^m}$$

↑ repeated matrix multiplies

↑ repeated mult. of nonlinearity deriv.

This can cause gradients to "explode" (go to ∞) or "vanish" (go to 0) as we propagate back.

Parameter Initialization

$$\frac{dC}{dz^m} = \left(w^{m+1\,T}\frac{dC}{dz^{m+1}}\right) \circ \frac{da^m}{dz^m}$$

Want to avoid vanishing/exploding gradients.

Can control the weight matrix initialization.

"initialize" → choose a distribution to sample initial values from.

Assume: $o = wx \qquad x \in \mathbb{R}^{n_{in}}$

1. No nonlinearities.
2. $x_i \sim N(\emptyset, 1)$
3. $w_i \sim N(\emptyset, \sigma^2)$

↑ what should we choose for sigma?

$o = \sum_i w_i x_i$

$\mathbb{E}[o] = \emptyset$

$Var[o] = \mathbb{E}[o^2] - \cancel{(\mathbb{E}[o])^2}$

$= \sum_i \mathbb{E}[w_i^2 x_i^2]$

$= \sum_i \mathbb{E}[w_i^2]\mathbb{E}[x_i^2] = n_{in}\sigma^2$

Forward pass: If we want variance of activations ≈ 1, set $n_{in}\sigma^2 = 1$

Backward pass: Want $n_{out}\sigma^2 = 1$.

Can't satisfy both in $n_{in} \neq n_{out}$.

So average as a compromise:

$$(n_{in}\sigma^2 + n_{out}\sigma^2)/2 = 1$$

$$\sigma = \sqrt{\frac{2}{n_{in}+n_{out}}}$$

"Xavier" or "Glorot" initialization

Autograd

ex $h = ReLU(w_h x + b_h)$

$o = w_o h + b_o$

$L = (y - o)^2$

We want $\frac{dL}{dw_o}, \frac{dL}{db_o}, \frac{dL}{dw_h}, \frac{dL}{db_h}$

We can always use the chain rule.

Break down into individual operation

$m_h = w_h x$

$z_h = m_h + b_h$

$h = ReLU(z_h)$

$m_o = w_o h$

$o = m_o + b_o$

$e = y - o$

$L = e^2$

Now we can write:

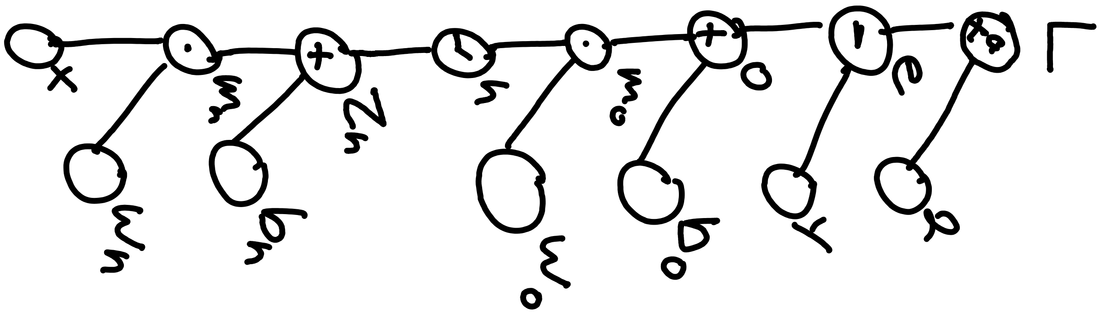
$$\frac{dL}{dw_n} = \frac{dL}{de}\frac{de}{do}\frac{do}{dm_o}\frac{dm_o}{dh}\frac{dh}{dz_n}\frac{dz_n}{dm_n}\frac{dm_n}{dw_n}$$

At each step, we're computing the derivative of a single simple operation.

1. What operations were applied
2. Derivatives of those operations

So, we:

1. Define a set of operation
2. Define their derivatives
3. Keep track of what operations were applied in a "computational graph"


x
·
w_n
b_n
+
z_n
1
y_0
·
w_0
+
b_0
1
y_1
+
b_2
L