

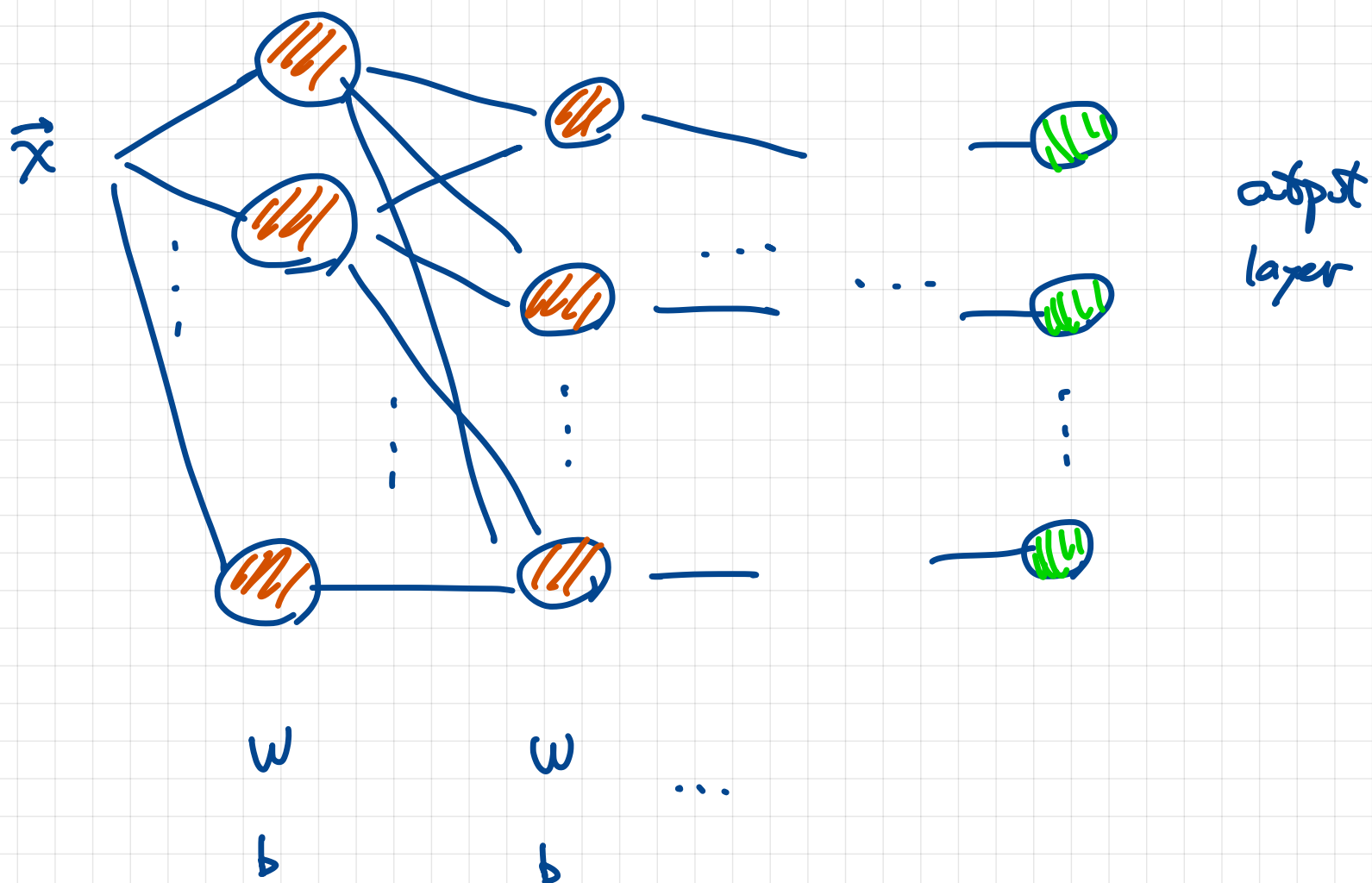
Short Takes 331

Machine learning:
Backpropagation



Machine learning: backpropagation.

- A neural network ...



... is ultimately a function designed in a particular way (layers, activation functions, etc) with parameters (weights & biases) to be "learned".

- One defines a cost function to be minimized via stochastic gradient descent.

$$C(w, b) = \sum_{i=1}^N C_i(w, b),$$

given data $\{(\vec{x}_1, \vec{y}_1), (\vec{x}_2, \vec{y}_2), \dots, (\vec{x}_N, \vec{y}_N)\}$

e.g. $C_i = (\vec{y}_i - \vec{a}(\vec{x}_i, w, b))^2$

→ We need $\vec{\nabla} C_i$!

Let's see...

$$\frac{\partial C}{\partial w_{ij}^{(k)}} = \sum_m \frac{\partial C}{\partial a_m} \frac{\partial a_m}{\partial w_{ij}^{(k)}} \quad , \quad \text{but...}$$

Diagram illustrating the partial derivative of the cost function C with respect to a weight $w_{ij}^{(k)}$. The weight is associated with a specific layer (indicated by k), neuron index i in that layer, and input index j . The derivative is expressed as a sum over network outputs a_m , where each term is the derivative of the cost with respect to the output a_m multiplied by the derivative of a_m with respect to the weight $w_{ij}^{(k)}$.

$$a = f(\dots f(w^{(2)} f(\underbrace{w^{(1)} \bar{x} + b^{(1)}}_{z^{(1)}}) + b^{(2)}) + b^{(3)} \dots)$$

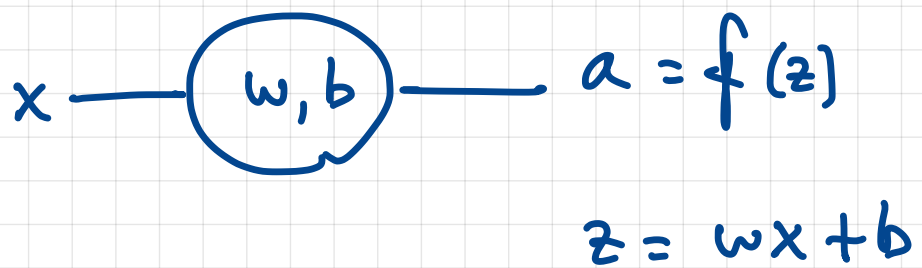
Diagram illustrating the nested structure of the network output a . It shows a sequence of layers: an input \bar{x} is processed by a layer with weights $w^{(1)}$ and bias $b^{(1)}$ to produce an intermediate output $z^{(1)}$. This $z^{(1)}$ is then processed by a second layer with weights $w^{(2)}$ and bias $b^{(2)}$ to produce another intermediate output $z^{(2)}$. Finally, $z^{(2)}$ is processed by a third layer with bias $b^{(3)}$ to produce the final output a . The overall function is $a = f(\dots f(w^{(2)} f(z^{(1)} + b^{(2)}) + b^{(3)} \dots)$.

- We won't tackle this fully, but work out two simple cases...

$$C = C(a)$$

$$a = a(w)$$

$$= f(\underbrace{wx + b}_{z})$$



$$\Rightarrow \frac{\partial C}{\partial w} = \frac{\partial C}{\partial a} \frac{\partial a}{\partial w}$$

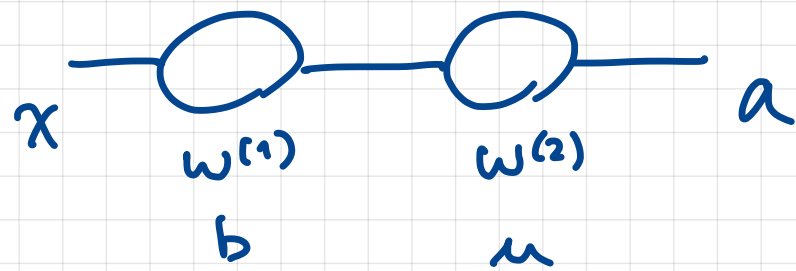
$$\frac{\partial a}{\partial w} = \frac{\partial f}{\partial z} \cdot \frac{\partial z}{\partial w}$$

$$\frac{\partial z}{\partial w} = x$$

Note: $\frac{\partial f}{\partial z}$ is fully determined by the activation function and $z = wx + b$.

- Slightly more complicated: two layers

$$a(w) = f(w^{(2)} \underbrace{f(w^{(1)}x + b)}_{z^{(1)}} + \mu)$$



$$\rightarrow \bullet \frac{\partial C}{\partial w^{(2)}} = \frac{\partial C}{\partial a} \frac{\partial a}{\partial w^{(2)}}$$

$$\frac{\partial a}{\partial w^{(2)}} = \frac{\partial f}{\partial z^{(2)}} \cdot \frac{\partial z^{(2)}}{\partial w^{(2)}}$$

fully determined by layer 2

$$\frac{\partial z^{(2)}}{\partial w^{(2)}} = f(z^{(1)}) \quad \checkmark$$

$$= \frac{\partial C}{\partial a} \frac{\partial f}{\partial z^{(2)}} f(z^{(1)})$$

$$\bullet \frac{\partial C}{\partial w^{(1)}} = \frac{\partial C}{\partial a} \frac{\partial a}{\partial w^{(1)}}$$

$$\frac{\partial a}{\partial w^{(1)}} = \frac{\partial f}{\partial z^{(2)}} \frac{\partial z^{(2)}}{\partial w^{(1)}}$$

$$\frac{\partial z^{(2)}}{\partial w^{(1)}} = w^{(2)} \frac{\partial f}{\partial z^{(1)}} \cdot \frac{\partial z^{(1)}}{\partial w^{(1)}}$$

$$\frac{\partial z^{(1)}}{\partial w^{(1)}} = x$$

$$\rightarrow = \frac{\partial C}{\partial a} \frac{\partial f}{\partial z^{(2)}} w^{(2)} \frac{\partial f}{\partial z^{(1)}} \cdot x$$

The gist of back propagation is that we perform far fewer operations if we calculate the gradient of the last layer first and then use the intermediate steps to calculate the gradient of the previous layer, and so on.

If we did forward propagation, we would be computing the same elements many times.

