

Discussion 7

The Chain rule

$$\begin{aligned}
 & \left[\mathbf{f}(\mathbf{g}(\mathbf{h}(\mathbf{i}(\mathbf{j}(\mathbf{x})))) \right]' \\
 &= \\
 & \mathbf{f}'(\mathbf{g}(\mathbf{h}(\mathbf{i}(\mathbf{j}(\mathbf{x})))) \\
 & \quad * \mathbf{g}'(\mathbf{h}(\mathbf{i}(\mathbf{j}(\mathbf{x})))) \\
 & \quad \quad * \mathbf{h}'(\mathbf{i}(\mathbf{j}(\mathbf{x}))) \\
 & \quad \quad \quad * \mathbf{i}'(\mathbf{j}(\mathbf{x})) \\
 & \quad \quad \quad \quad * \mathbf{j}'(\mathbf{x}) \\
 & \quad \quad \quad \quad \quad * \mathbf{1}
 \end{aligned}$$


fig: The chain rule (source:
Wikipedia)

This week's resources:

- 3Blue1Brown: Backpropagation, intuitively video
- 3Blue1Brown: Backpropagation calculus video
- Artem Kirsanov: The Most Important Algorithm in Machine Learning video

1 Backprop in Practice: Staged Computation

We will apply backpropagation for the function $f(x, y, z) = (x + y)z$.

(a) Decompose f into two simpler functions.

(b) Draw the computation graph that represents f following the composition from part (a).

- (c) Write the steps of the forward pass and backward pass through the network.
- (d) Update your network drawing with the intermediate values in the forward and backward pass using the inputs $x = -2$, $y = 5$, and $z = -4$.

2 A Two Layer Neural Network

Consider a neural network with two fully connected layers **but no nonlinearities**. The network maps input $\mathbf{x} \in \mathbb{R}^2$ to output $y \in \mathbb{R}$ according to the following functions:

$$\mathbf{h} = \mathbf{W}_1 \mathbf{x}, \quad y = \mathbf{W}_2 \mathbf{h},$$

where $\mathbf{W}_1 \in \mathbb{R}^{2 \times 2}$ and $\mathbf{W}_2 \in \mathbb{R}^{1 \times 2}$. We define the loss as the mean squared error:

$$L = \frac{1}{2}(y - t)^2,$$

where t is the target output.

- (a) Write the steps of the forward pass and backward pass through the network. *Hint:* You may find the following vector calculus identities useful:

$$(i) \frac{\partial}{\partial \mathbf{x}}(\mathbf{y}^\top \mathbf{x}) = \mathbf{y}, \quad (ii) \frac{\partial}{\partial \mathbf{A}}(\mathbf{y}^\top \mathbf{A} \mathbf{x}) = \mathbf{y} \mathbf{x}^\top, \quad (iii) \frac{\partial}{\partial \mathbf{x}}(\mathbf{A} \mathbf{x}) = \mathbf{A},$$

$$(iv) \frac{\partial L}{\partial \mathbf{x}} = \frac{\partial L}{\partial y} \frac{\partial y}{\partial \mathbf{x}}, \quad (v) \frac{\partial L}{\partial \mathbf{x}} = \left(\frac{\partial \mathbf{y}}{\partial \mathbf{x}} \right)^\top \frac{\partial L}{\partial \mathbf{y}}, \quad (vi) \frac{\partial L}{\partial \mathbf{A}} = \sum_{i,j} \frac{\partial L}{\partial Y_{ij}} \frac{\partial Y_{ij}}{\partial \mathbf{A}}$$

where y is a scalar, $\mathbf{x}$ and $\mathbf{y}$ are vectors, $\mathbf{A}$ is a matrix, and Y_{ij} is the ij th element of $\mathbf{Y}$.

- (b) Write out y directly in terms of $\mathbf{x}$ and the weights $\mathbf{W}_1, \mathbf{W}_2$.
- (c) Suppose you insert a nonlinearity $h(\cdot)$ after the first layer so that $\mathbf{h} = h(\mathbf{W}_1\mathbf{x})$. Why does this change what the network can represent? Illustrate using a simple example with ReLU.