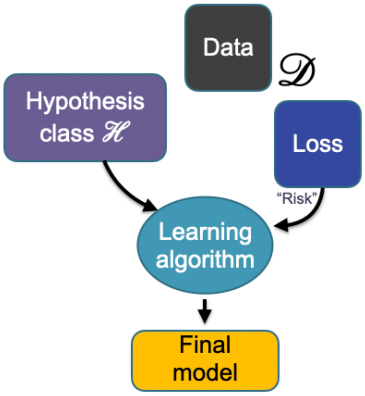
Monday, March 23, 2026

Lecture 6: Automatic differentiation

Recap: How have we been filling this picture in?

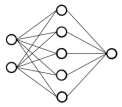


Finite differences

$$\frac{\partial f(\vec{\theta})}{\partial \theta_i} = \lim_{\varepsilon \to 0} \frac{f(\cdots, \theta_i + \varepsilon, \cdots) - f(\cdots, \theta_i, \cdots)}{\varepsilon}$$

Q for you: For the NN we just built … how many times would we have to evaluate it to get $\nabla_\theta \mathcal{L}$ with finite differences?

$$z = W_1 X + b_1$$
$$h = \text{ReLU}(z_1)$$
$$y_{\text{pred}} = W_2 h + b_2$$



$$W_1 \in \mathbb{R}^{16 \times 5}, b_1 \in \mathbb{R}^{16}$$
$$W_2 \in \mathbb{R}^{16}, b_2 \in \mathbb{R}$$

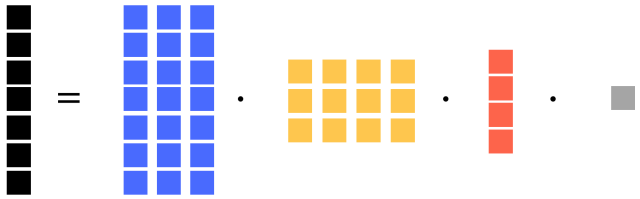
Consider the bollowing program:

$$\vec{z} = f(g(h(x)))$$

$$\frac{dz_i}{dx} = \frac{df_i}{dg_j} \cdot \frac{dg_j}{dh_k} \cdot \frac{dh_k}{dx}$$

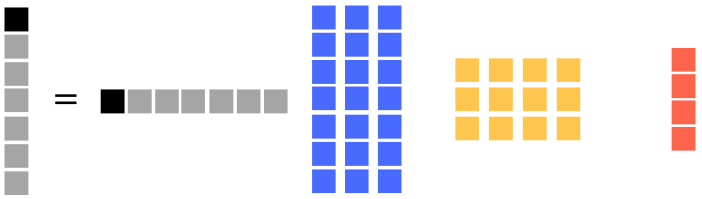
Forward mode automatic differentiation

$$\delta\vec{z} = J_{f(g)} \cdot J_{g(h)} \cdot J_{h(x)} \cdot 1$$



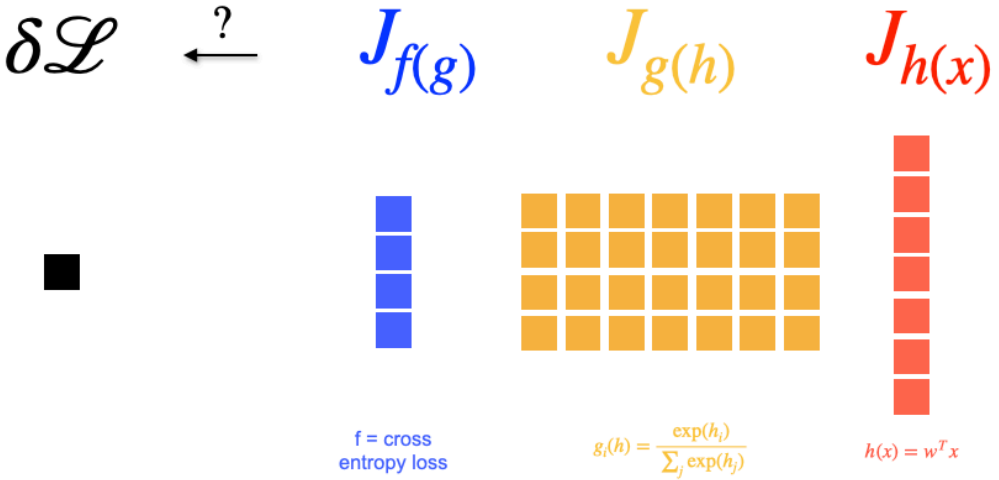
Reverse mode automatic differentiation

$$\delta\vec{z} = e_i^T \cdot J_{f(g)} \cdot J_{g(h)} \cdot J_{h(x)}$$



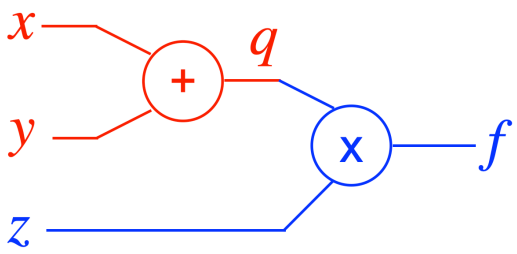
Which is better for ML?

In ML, we often have “many inputs” , “one output” type problems, e.g, $x \in \mathbb{R}^d$ and $\mathscr{L} \in \mathbb{R}$. Is forward or reverse mode more efficient? (Circle one)



Computational graphs

Interactive exercise: Calculate the gradients $\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}$ at the point (x,y,z) = (5, -2, 4).



With Jacobians

Goal: Find the gradients at the point $W = \begin{pmatrix} 0.1 & 0.5 \\ -0.3 & 0.8 \end{pmatrix}, x = \begin{pmatrix} 0.2 \\ 0.4 \end{pmatrix}$