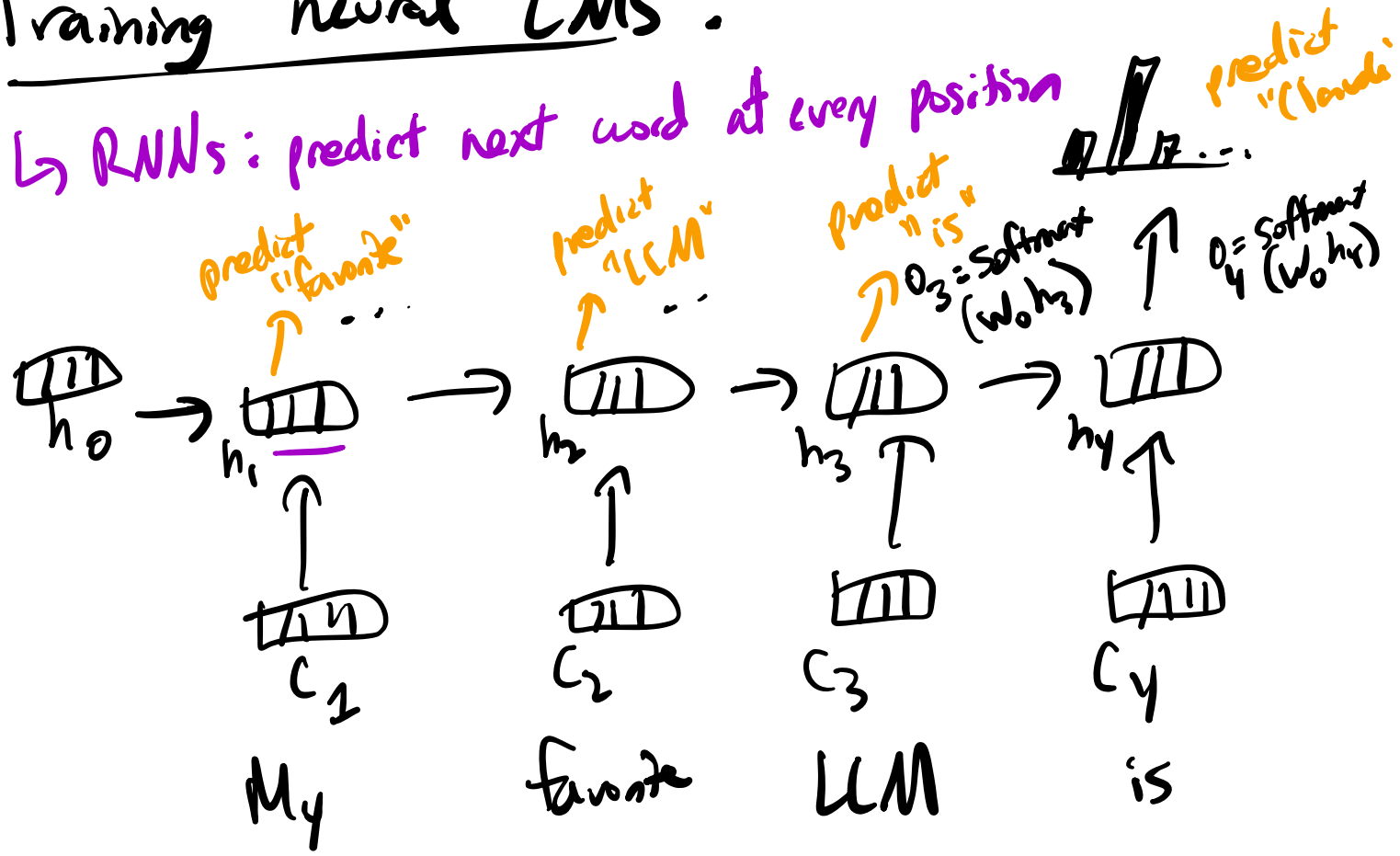
Training neural LMs:

↳ RNNs: predict next word at every position



↳ What are my parameters θ?

$$\theta_{RNN} = \{E, W_o, W_h, W_e\}$$

↳ θ_{RNN} begins as completely random numbers

↳ Goal: Adjust θ_{RNN} such that we make better predictions of the next word

Gradient descent for neural LMs:

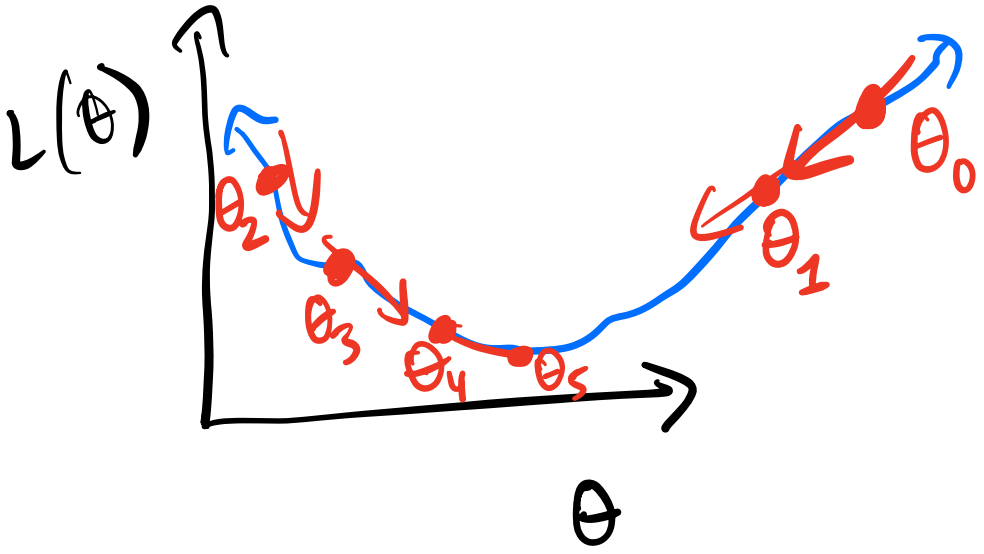
1. Define a loss fn $L(\theta)$ that tells us how bad the model is doing at next word prediction
   - ↳ smooth, differentiable
   - ↳ NLL for LM
2. Compute the gradient $\frac{dL}{d\theta}$.

   The gradient tells us the direction of steepest ascent.
   - ↳ how much would L change if we increase θ_i by a very small amount: $\frac{dL}{d\theta_i}$;
   - ↳ RNN: $\frac{dL}{d\theta} : \left\{ \frac{dL}{dE}, \frac{dL}{dW_0}, \frac{dL}{dW_h}, \frac{dL}{dW_e} \right\}$
3. Take a step in the direction of the negative gradient (decrease L)

$$\theta_{new} = \theta_{OLD} - \eta \frac{dL}{d\theta}$$

↳ learning rate, "step size"

↳ LLM optimizers: Adam, AdamW, Muon



Step 1:
Loss fn: NLL

prefix: My favorite LLM is

target: Claude → observed in training data

max P(Claude | My favorite LLM is)

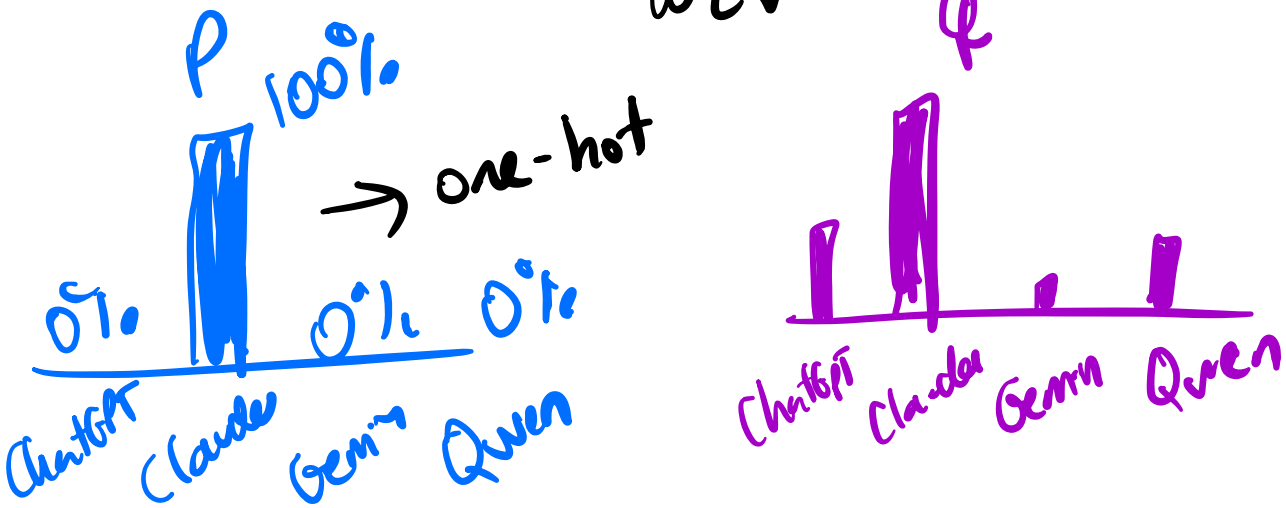
min $-\log P(\uparrow)$

Why is this also called cross-entropy loss?

↳ target / reference distribution p

predicted / model distribution q

$$CE(p, q) = -\sum_{w \in V} p(w) \log q(w)$$



CE loss simplifies

$$-\log q(\text{Claude} \mid \text{My fave LLM is})$$

Step 2: backpropagation:

↳ chain rule of calculus

$$\frac{d}{dx} g(f(x)) = \frac{dg}{df} \cdot \frac{df}{dx}$$

↳ in NLM training, we apply the chain rule going backwards from the output layer

↳ caching of intermediate partial derivatives / products

---

a toy example

↳ tiny scalar network (no matrices)

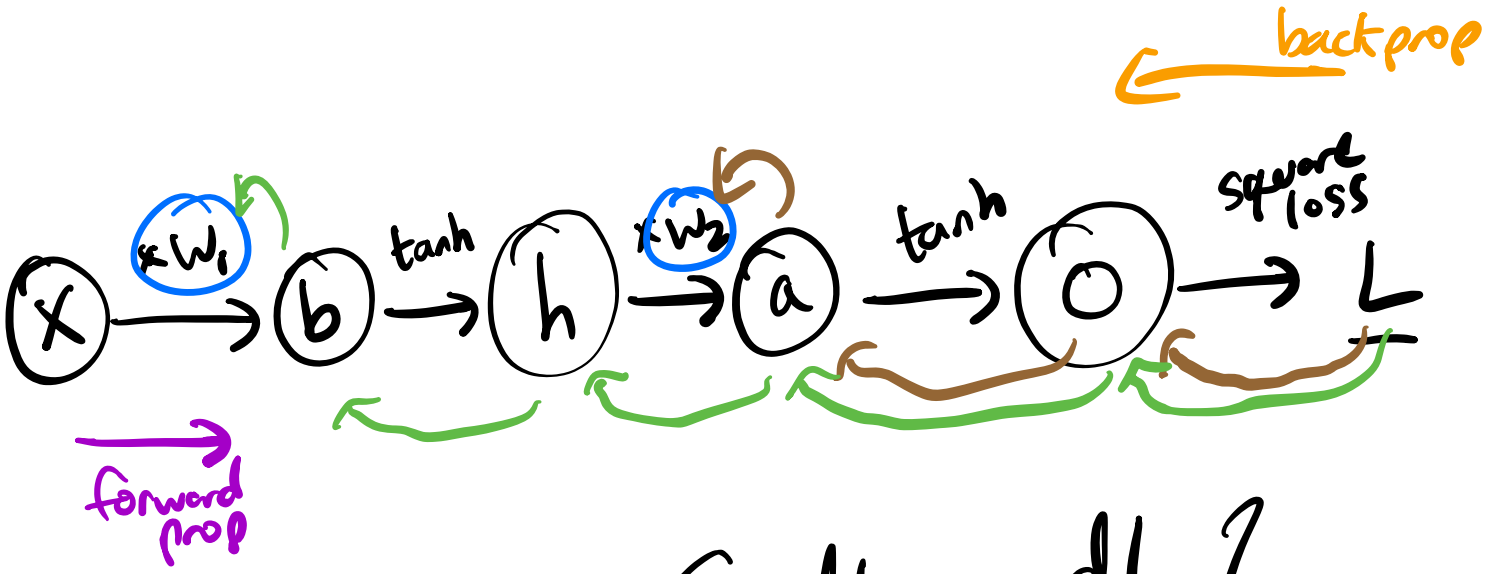
↳ square loss instead of NLL

input (x, y)

hidden $h = \tanh(w_1 x)$, $b = w_1 x$

output $o = \tanh(w_2 h)$, $a = w_2 h$

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



gradient $\frac{dL}{d\theta} : \left\{ \frac{dL}{dw_1}, \frac{dL}{dw_2} \right\}$

$$\frac{dL}{dw_2} = \frac{dL}{do} \cdot \frac{do}{da} \cdot \frac{da}{dw_2}$$

$$\frac{d}{dx}\tanh(x) = 1 - \tanh^2(x)$$

$$-(y-o)\cdot(1-o^2)\cdot h$$

$$\frac{dL}{dw_1} = \underbrace{\frac{dL}{do} \cdot \frac{do}{da}}_{\text{same product for } \frac{dL}{dw_1}, \frac{dL}{dw_2}, \text{ cached}} \cdot \frac{da}{dh} \cdot \frac{dh}{db} \cdot \frac{db}{dw_1}$$

Step 3: update params

$$w_{1_{new}} = w_{1_{old}} - \eta \frac{dL}{dw_2}$$

$$w_{2_{new}} = w_{2_{old}} - \eta \frac{dL}{dw_2}$$