Chinchilla scaling "law":

L = test loss

↳ avg. neg. log likelihood of test set

D = dataset size

↳ # of tokens

N = # model params

↳ embeddings, $\underbrace{W_q, W_k, W_v, W_z}_{\propto \text{ num layers}}$, W_{SOFTMAX}

C = compute budget

FLOPS(D, N) ↩ deterministic fn of model / data size

↳ floating point operations

goal: given a fixed FLOPS budget C

find

$$\underset{N, D \text{ s.t. } \mathrm{FLOPS}(N,D) = C}{\operatorname{argmin}} L(N, D)$$

$$L(N, D) = \frac{A}{N^{\alpha}} + \frac{B}{D^{\beta}} + E$$

contrib. of model size

contribution of data

loss of a perfect LM

---

trained two models w/ same compute C

- Gopher: 280B params, 300B tokens
- Chinchilla: 70B params, 1.4T tokens

L(Gopher) = 1.993

L(Chinchilla) = 1.936

Difference between RL and SFT:

in RL, we are maximizing expected reward

$\max E(R(y|x))$ where y is sampled from the model given prompt x

in SFT, we are minimizing loss of ground-truth y that is given to us

$\min -\log(p(y|x))$ where y is given to us

these are actually very similar!

given prompt x, I sample outputs Y_L and Y_W from the current model

$$R(y_L|x) = 0$$
$$R(Y_W|x) = 1$$

What if I do SFT over (x, Y_W)

$$L(\theta) = -\log p(Y_W|x)$$

$$\frac{dL}{d\theta} = \underbrace{-\frac{d}{d\theta} \log p(p_w | x)}$$

if I instead do RL, via REINFORCE
(Williams, 1992)

$$\frac{dL}{d\theta} E[R(y|x)] =$$

$$\cancel{0 \cdot \frac{d}{d\theta} \log p(y_L | x)} +$$

$$1 \cdot \frac{d}{d\theta} \log p(y_w | x)$$

$$= \frac{d}{d\theta} \log p(y_w | x)$$

$$\text{RL}: \quad \theta_{new} = \theta_{OLD} + \eta \frac{d}{d\theta} \log p(y_w | x)$$

$$\text{SFT}: \quad \theta_{new} = \theta_{OLD} - \eta \cdot \left[-\frac{d}{d\theta} \log p(y_w | x) \right]$$

↳ both methods increase $p(y_w | x)$

"on-policy"

↳ RL samples y from current model, while SFT generally uses y_w from an existing dataset ↳ "off-policy"