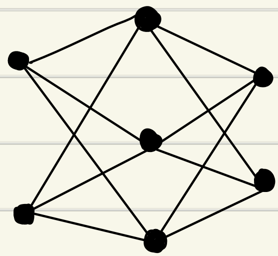
Low Rank Adaptation (LoRA)

LoRA fine tunes a model by adding new trainable parameters



consider a neural network

$$x \rightarrow h(x)$$

input x

hidden layer function of x

$h(x) = W_0 x$

W_0 = weight matrix
x = input vector

$W_0 \in R^{d \times k}$
$x \in R^{k \times 1}$
$h(x) \in R^{d \times 1}$

→ so what happens if we retrain all parameters?

suppose our weight matrix has shape 1000×1000
so this would mean changing 1000000 parameters!

so how does LoRA help us?

$$\underbrace{h(x) = W_0 x}_{\text{old term}} + \underbrace{\Delta W x}_{\text{new term!}}$$

where $\Delta W x = BA$ where B & A are matricies

$$\therefore h(x) = W_0 x + BAx$$
$$h(x) = (W_0 + BA)x$$

$W_0, \Delta W \in \mathbb{R}^{d \times k}$

$B \in \mathbb{R}^{d \times r}$

$A \in \mathbb{R}^{r \times k}$

r is called the intrinsic rank of the model

$h(x) \in \mathbb{R}^{d \times 1}$

the reason this works & we get efficiency gains because this r is a lot smaller than d and k

$$(\overbrace{W_0}^{\text{frozen}} + \overbrace{BA}^{\text{trainable}})\, x = h(x)$$

B and A contain far fewer terms than W_0

so eg if $d = 1000$
$k = 1000$
$r = 2$

$\rightarrow (d \times r) + (k \times r)$
$= (1000 \times 2) + (1000 \times 2)$
$=$ 4000 trainable parameters

significantly reduces the parameters we have to train