HOW YOU – YES, YOU! – CAN TRAIN AN LLM*

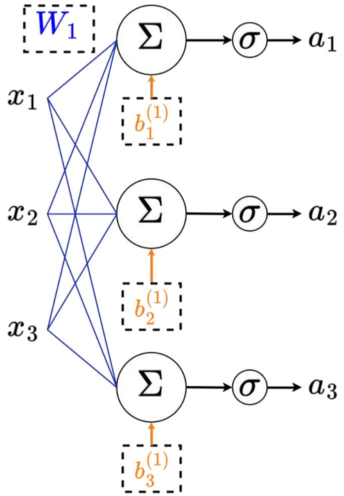
Sam Bowyer

*(SORT OF...)

What even is a neural network?

- Basic neural network layer

$$f(x; W, b) = \sigma(Wx + b)$$

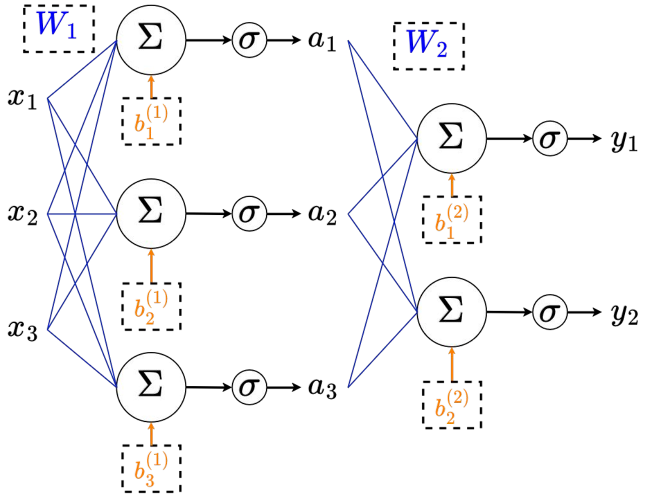


What even is a neural network?

- Basic neural network layer

$$f(x; W, b) = \sigma(Wx + b)$$

- Compose L layers $f = f_1 \circ f_2 \cdots \circ f_L$



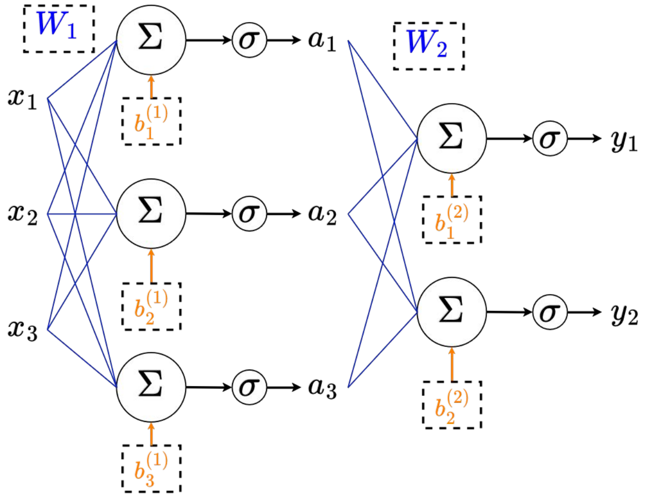
What even is a neural network?

- Basic neural network layer

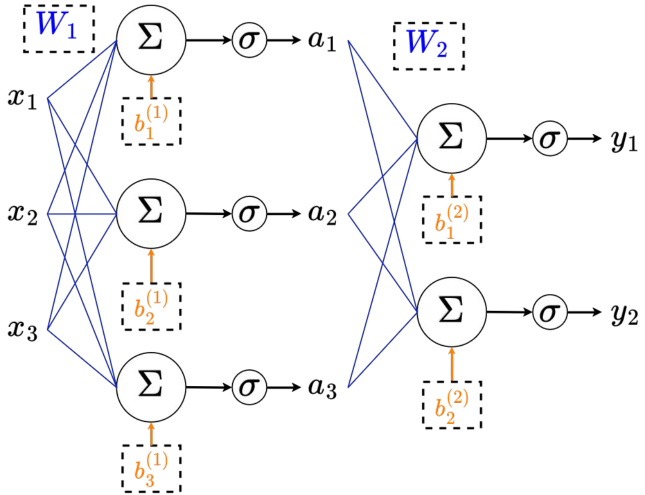
$$f(x; W, b) = \sigma(Wx + b)$$

- Compose L layers $f = f_1 \circ f_2 \cdots \circ f_L$

- View the function as a distribution over outputs $p(y|x;\theta) = f(x;\theta)$



What even is a neural network?



- Basic neural network layer

$$f(x; W, b) = \sigma(Wx + b)$$

- Compose L layers $f = f_1 \circ f_2 \cdots \circ f_L$

- View the function as a distribution over outputs $p(y|x;\theta) = f(x;\theta)$
- Parameters $\theta = \{W_l, b_l | l = 1, \ldots, L\}$ trained via maximum likelihood

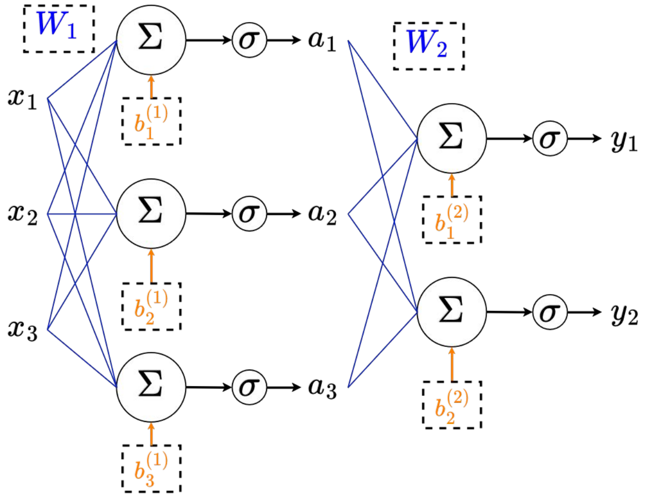
$$\theta^* = \arg\max_\theta p(\mathcal{Y}|\mathcal{X};\theta)$$

What even is a neural network?

- Basic neural network layer

$$f(x; W, b) = \sigma(Wx + b)$$

- Compose L layers $f = f_1 \circ f_2 \cdots \circ f_L$

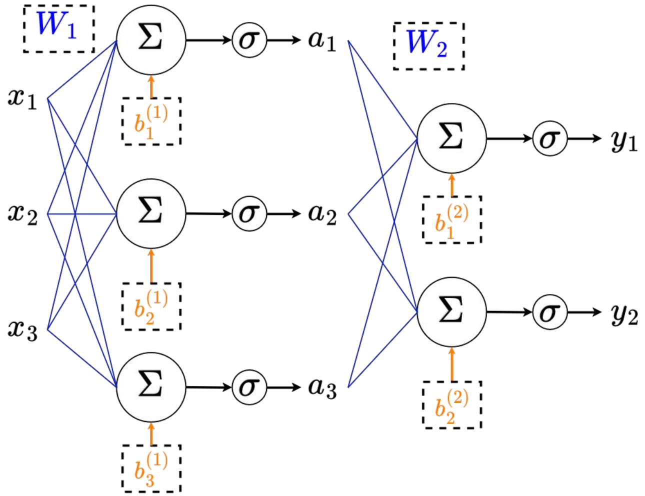


- View the function as a distribution over outputs $p(y|x;\theta) = f(x;\theta)$
- Parameters $\theta = \{W_l, b_l | l = 1, \ldots, L\}$ trained via maximum likelihood

$$\theta^* = \arg\max_\theta p(\mathcal{Y}|\mathcal{X};\theta)$$

  - Via gradient ascent-type optimisation $\theta \leftarrow \theta + \eta \nabla_\theta p(\mathcal{Y}|\mathcal{X};\theta)$

What even is a neural network?



- Basic neural network layer

$$f(x; W, b) = \sigma(Wx + b)$$

- Compose L layers $f = f_1 \circ f_2 \cdots \circ f_L$

- View the function as a distribution over outputs $p(y|x;\theta) = f(x;\theta)$
- Parameters $\theta = \{W_l, b_l | l = 1, \ldots, L\}$ trained via maximum likelihood

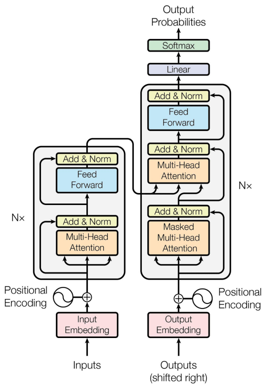
$$\theta^* = \arg\max_\theta p(\mathcal{Y}|\mathcal{X};\theta)$$

  – Via gradient ascent-type optimisation $\theta \leftarrow \theta + \eta \nabla_\theta p(\mathcal{Y}|\mathcal{X};\theta)$

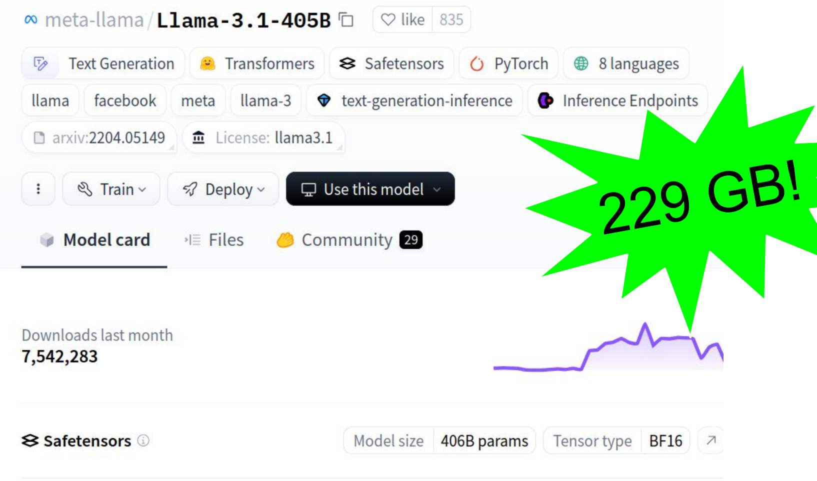
- In an LLM: x - start of some text; y - the rest of the text

Large Language Models are LARGE

- TONS of parameters

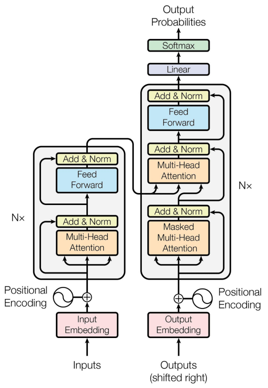


Attention Is All You Need.
Vaswani et al. (2017)

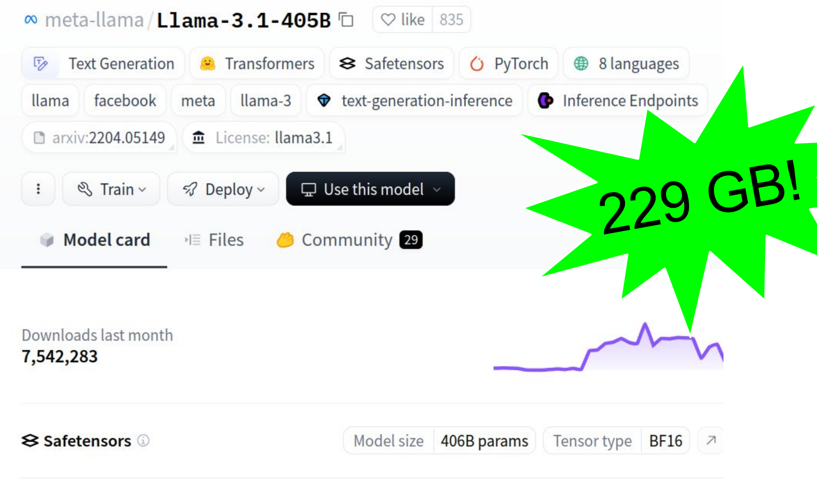


Large Language Models are LARGE

- TONS of parameters
- Incredibly flexible learners

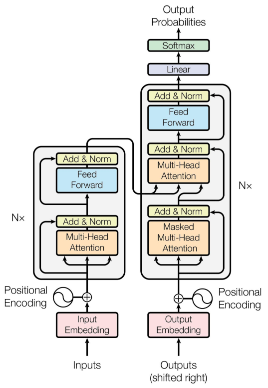


Attention Is All You Need.
Vaswani et al. (2017)

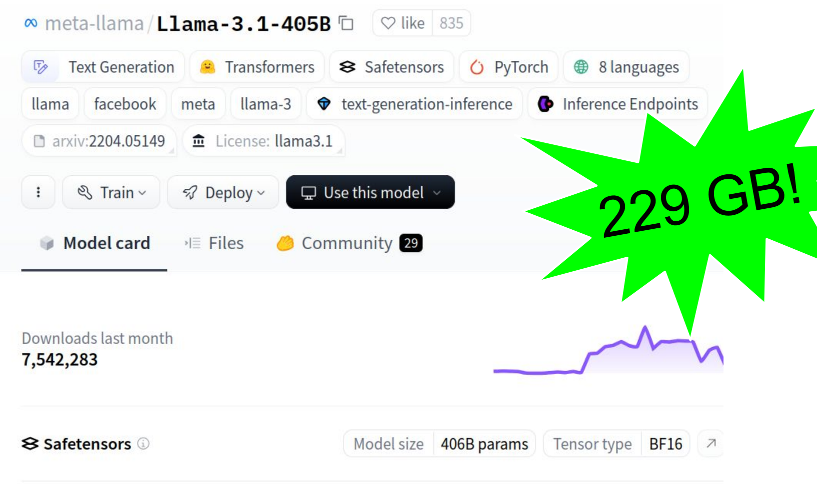


Large Language Models are LARGE

- TONS of parameters
- Incredibly flexible learners
- …but also a pain to train, requiring:

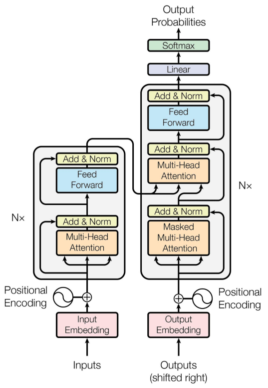


Attention Is All You Need.
Vaswani et al. (2017)

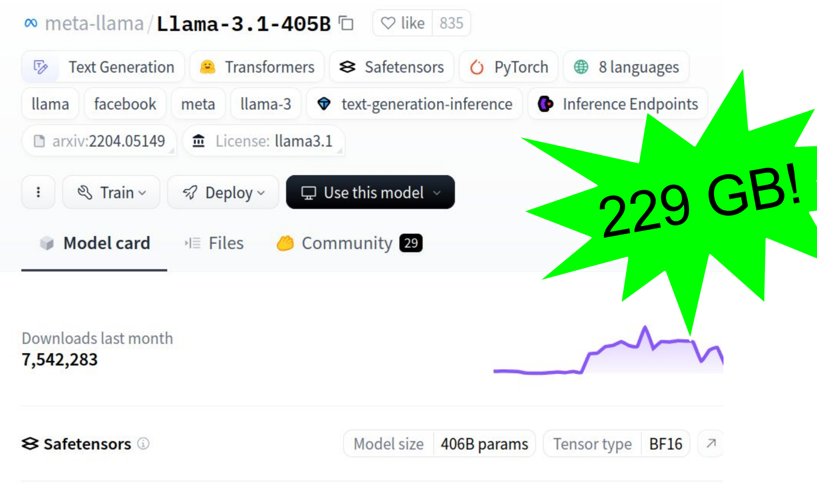


Large Language Models are LARGE

- TONS of parameters
- Incredibly flexible learners
- …but also a pain to train, requiring:
  - Gigantic training set $\mathcal{D}$

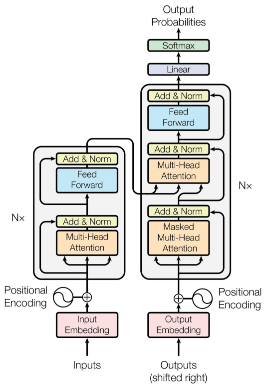


Attention Is All You Need.
Vaswani et al. (2017)

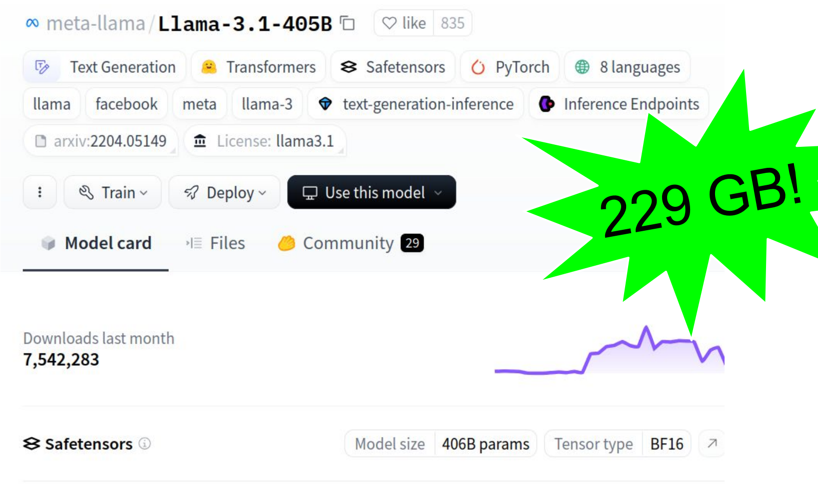


Large Language Models are LARGE

- TONS of parameters
- Incredibly flexible learners
- …but also a pain to train, requiring:
  - Gigantic training set $\mathcal{D}$
  - Top of the range hardware

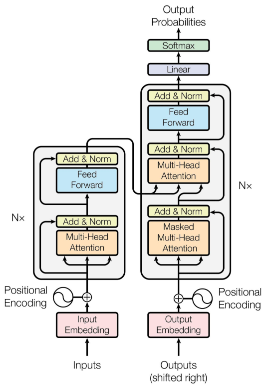


Attention Is All You Need.
Vaswani et al. (2017)

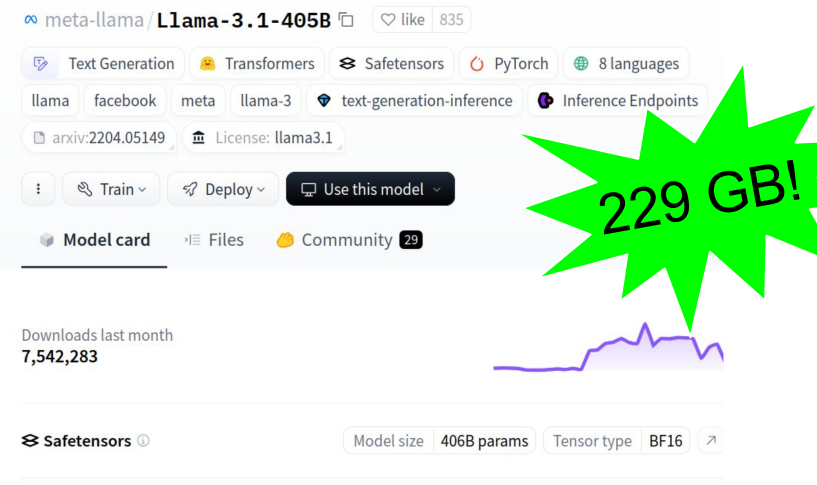


Large Language Models are LARGE

- TONS of parameters
- Incredibly flexible learners
- …but also a pain to train, requiring:
  - Gigantic training set $\mathcal{D}$
  - Top of the range hardware
  - Lots of memory

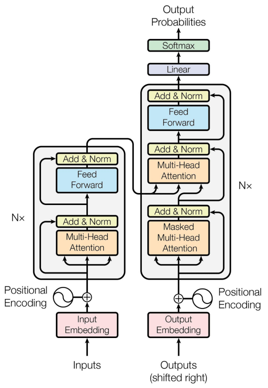


Attention Is All You Need.
Vaswani et al. (2017)

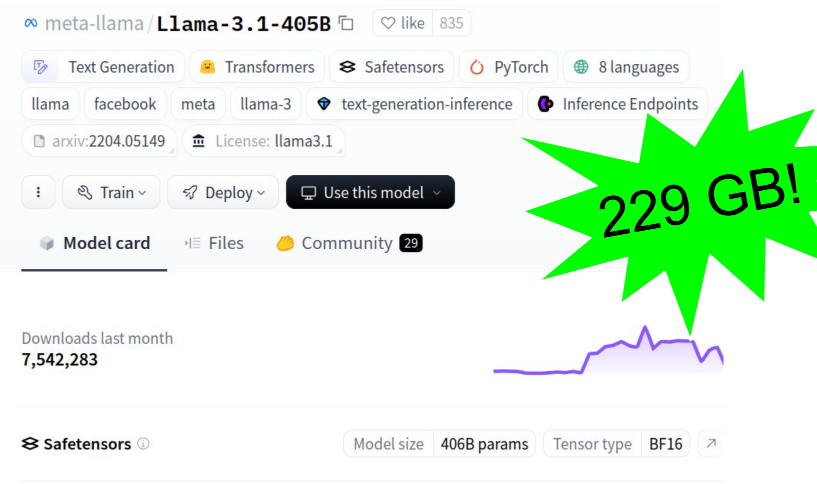


Large Language Models are LARGE

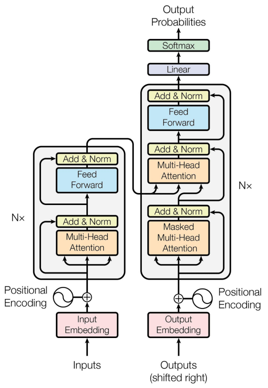
- TONS of parameters
- Incredibly flexible learners
- …but also a pain to train, requiring:
  – Gigantic training set $\mathcal{D}$
  – Top of the range hardware
  – Lots of memory
  – Lots of time



Attention Is All You Need.
Vaswani et al. (2017)

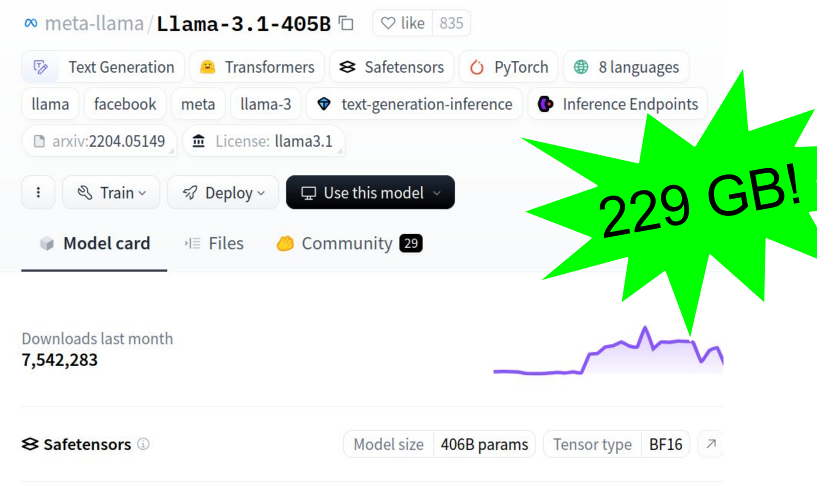


Large Language Models are LARGE



Attention Is All You Need.
Vaswani et al. (2017)

- TONS of parameters
- Incredibly flexible learners
- …but also a pain to train, requiring:
  - Gigantic training set $\mathcal{D}$
  - Top of the range hardware
  - Lots of memory
  - Lots of time
  - Lots of skilled (and patient!) engineers



2021-11-28 1:50am ET [Stephen]: 12.27

Looks like 26 tried to immediately upload a checkpoint and failed its cp commands! Then it took another step, lowered its scalar, and tried uploading again! And again! The humanity! We're already at loss scale 0.25.

So you don't actually want to train one from scratch...

So you don't actually want to train one from scratch...

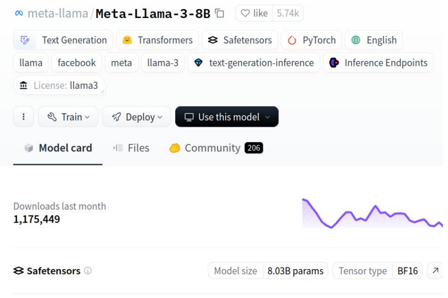
- Instead, take a pretrained 'foundation model' and *finetune* it on your specific data $\mathcal{D}_{\mathrm{FT}}$

ChatGPT

Claude

Meta AI

Hugging Face



So you don't actually want to train one from scratch...

- Instead, take a pretrained 'foundation model' and *finetune* it on your specific data $\mathcal{D}_{\text{FT}}$

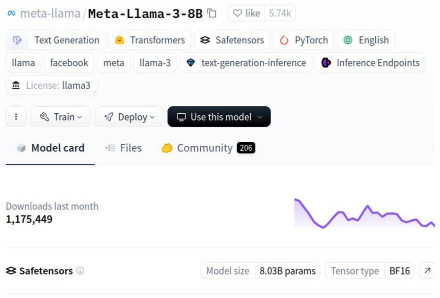
- *Full Finetuning* $\theta^* = \arg\max_{\theta} p(\mathcal{Y}_{\text{FT}} | \mathcal{X}_{\text{FT}}; \theta)$

ChatGPT

Claude

Meta AI

Hugging Face



So you don't actually want to train one from scratch…

- Instead, take a pretrained 'foundation model' and *finetune* it on your specific data $\mathcal{D}_{\mathrm{FT}}$
- *Full Finetuning* $\theta^* = \arg\max_{\theta} p(\mathcal{Y}_{\mathrm{FT}} | \mathcal{X}_{\mathrm{FT}}; \theta)$
- *Partial Finetuning*: freeze some parameters $\theta_{\mathrm{frozen}} \subset \Theta$ and not others $\theta_{\mathrm{FT}} \subset \Theta$

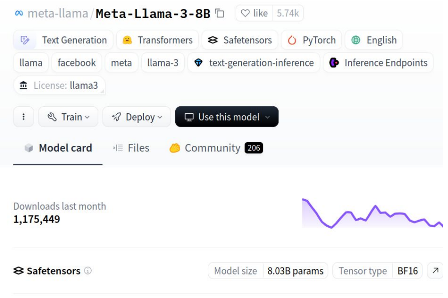
$$\theta^*_{\mathrm{FT}} = \arg\max_{\theta_{\mathrm{FT}}} p(\mathcal{Y}_{\mathrm{FT}} | \mathcal{X}_{\mathrm{FT}}; \theta_{\mathrm{frozen}} \cup \theta_{\mathrm{FT}})$$

ChatGPT

Claude

Meta AI

Hugging Face



And you don’t even really want to finetune the whole thing…

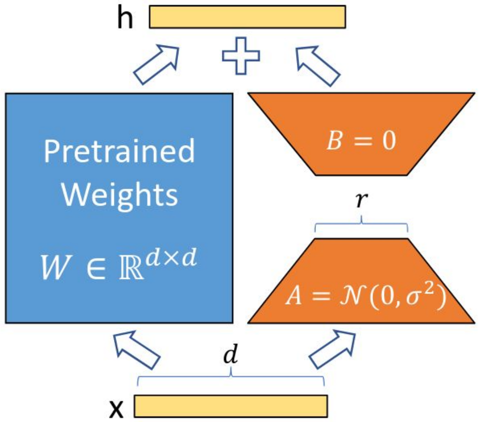
And you don’t even really want to finetune the whole thing…

- Most parameters $W \in \mathbb{R}^{d \times d}$ only need a small nudge in the right direction

And you don't even really want to finetune the whole thing…

- Most parameters $W \in \mathbb{R}^{d \times d}$ only need a small nudge in the right direction
- **LoRA**: freeze all parameters and train some additive low-rank matrices $A, B^T \in \mathbb{R}^{r \times d}, r \ll d$

$$W = W_{\text{frozen}} + BA$$



LoRA: Low-Rank Adaptation of Large Language Models. Hu et al. (2021)

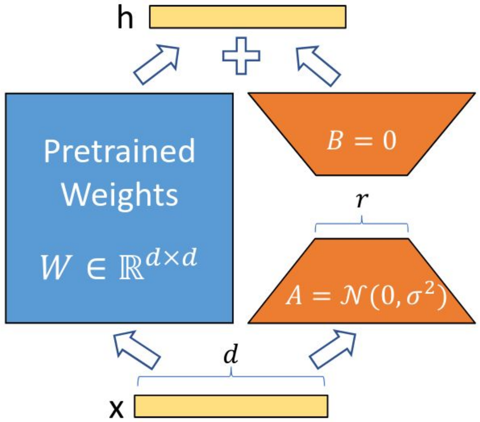
And you don't even really want to finetune the whole thing...

- Most parameters $W \in \mathbb{R}^{d \times d}$ only need a small nudge in the right direction

- **LoRA**: freeze all parameters and train some additive low-rank matrices $A, B^T \in \mathbb{R}^{r \times d}, r \ll d$

$$W = W_{\text{frozen}} + BA$$

  – Only requires training $2dr$ parameters instead of d^2



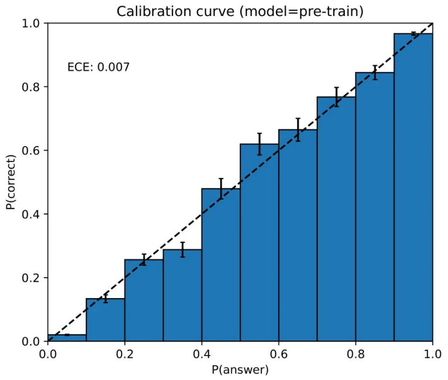
LoRA: Low-Rank Adaptation of Large Language Models. Hu et al. (2021)

(But finetuning *can* cause problems…)

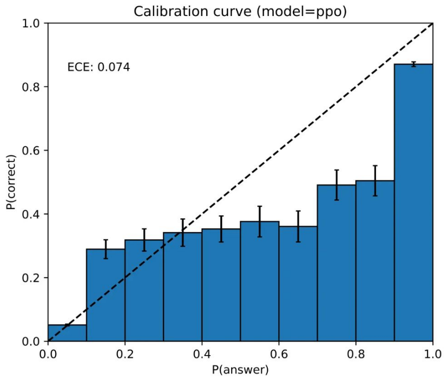
(But finetuning *can* cause problems...)

- Finetuning often makes models overconfident

(Before finetuning)



(After finetuning)



GPT-4 Technical Report. OpenAI (2023)

(But finetuning *can* cause problems...)

- Finetuning often makes models overconfident

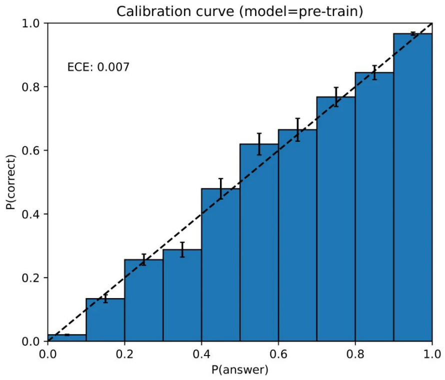
- Potential solution: rather than just a point estimate

$$\theta^*_{\mathrm{FT}} = \arg\max_{\theta_{\mathrm{FT}}} p(\mathcal{Y}_{\mathrm{FT}} | \mathcal{X}_{\mathrm{FT}}; \theta_{\mathrm{FT}})$$

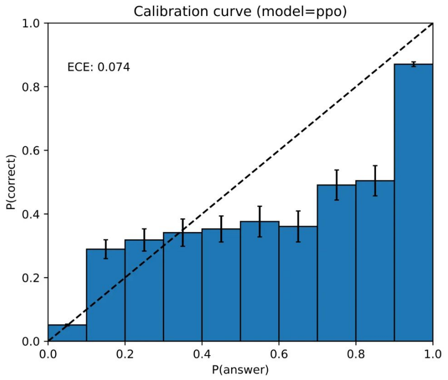
  find the whole posterior distribution

$$p(\theta_{\mathrm{FT}} | \mathcal{Y}_{\mathrm{FT}}, \mathcal{X}_{\mathrm{FT}})$$

(Before finetuning)



(After finetuning)



GPT-4 Technical Report. OpenAI (2023)

(But finetuning *can* cause problems…)

- Finetuning often makes models overconfident

- Potential solution: rather than just a point estimate

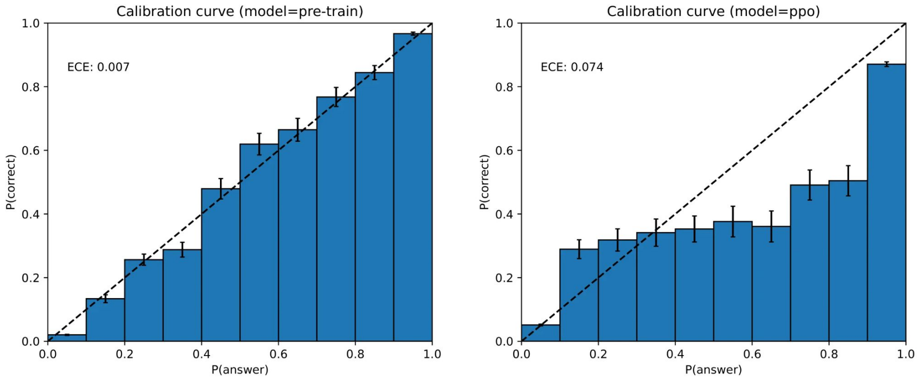
$$\theta^*_{\mathrm{FT}} = \arg\max_{\theta_{\mathrm{FT}}} p(\mathcal{Y}_{\mathrm{FT}} | \mathcal{X}_{\mathrm{FT}}; \theta_{\mathrm{FT}})$$

  find the whole posterior distribution

$$p(\theta_{\mathrm{FT}} | \mathcal{Y}_{\mathrm{FT}}, \mathcal{X}_{\mathrm{FT}})$$

- Use knowledge of this distribution to correct the model's overconfidence

(Before finetuning) (After finetuning)



GPT-4 Technical Report. OpenAI (2023)