mônade

Smol but Mighty

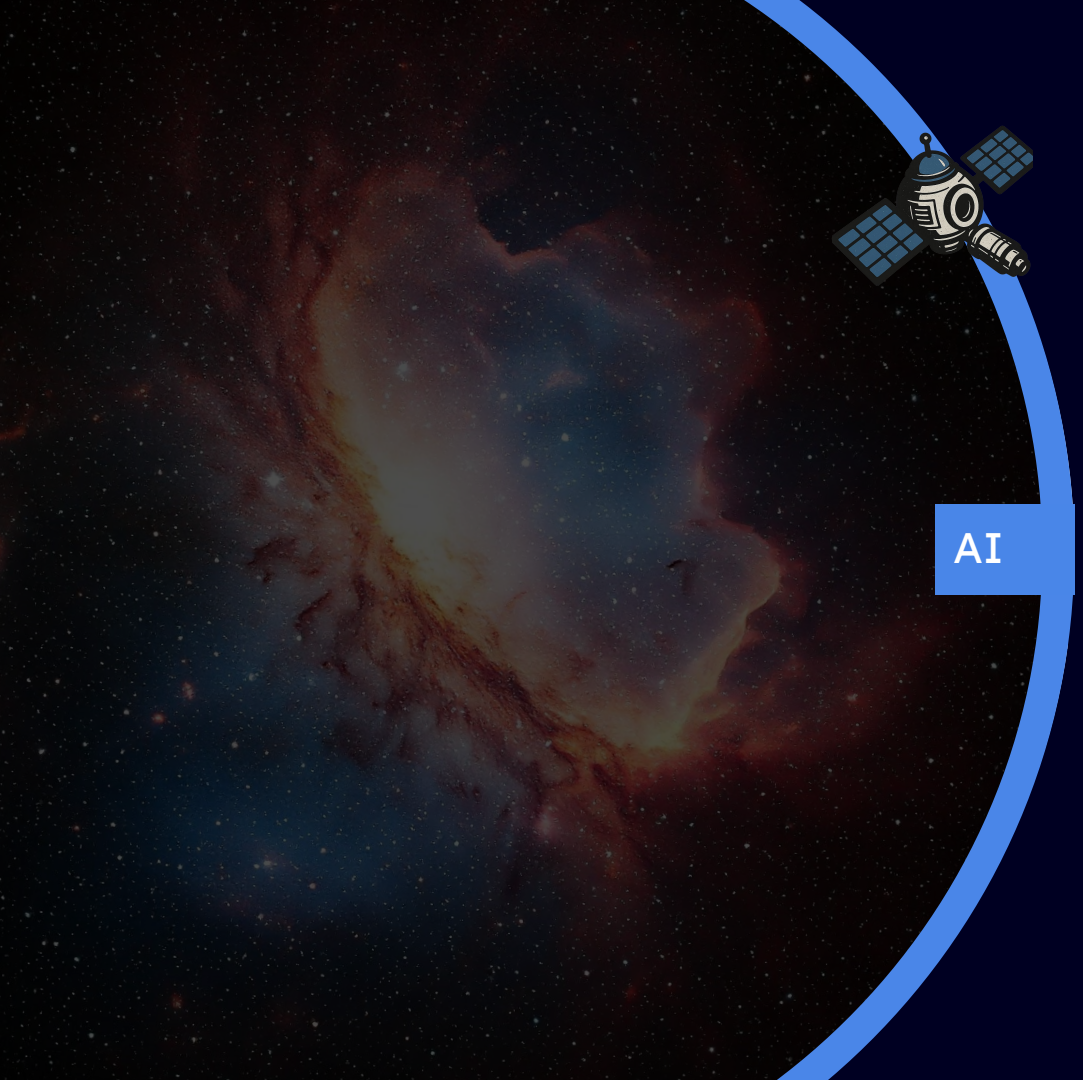
Addestra il tuo Mini Language Model

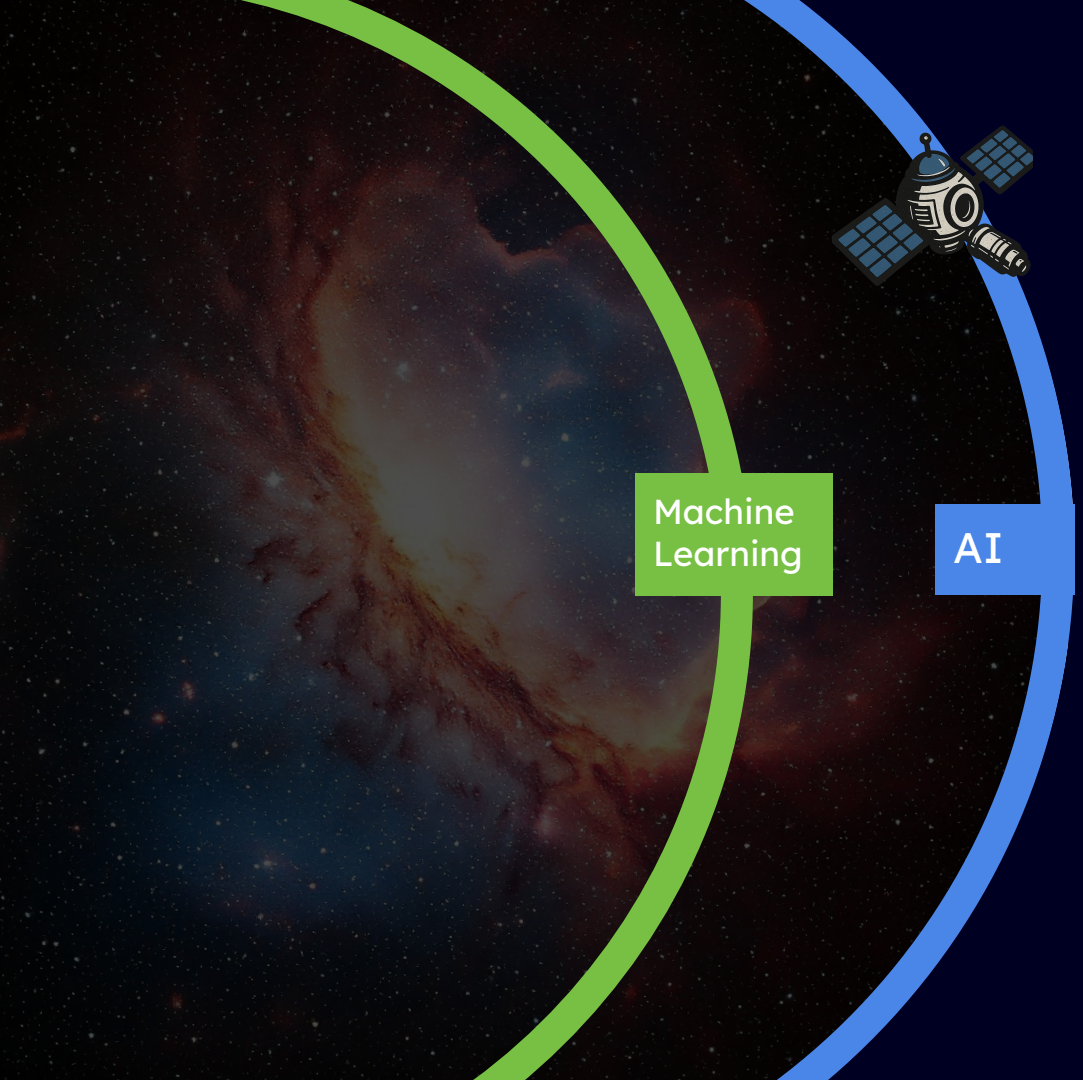
Piero Dotti

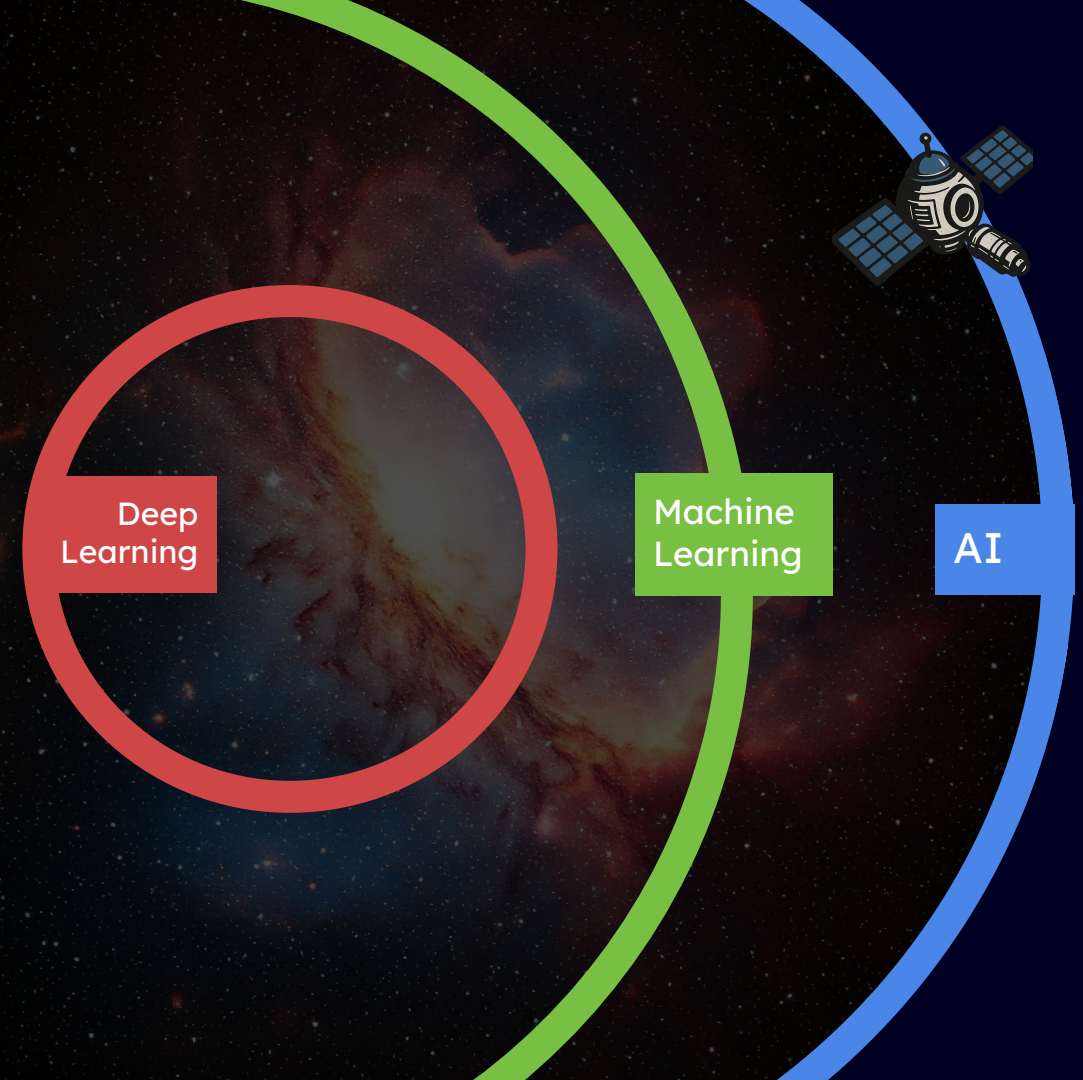
CTO @ Monade & Elysia

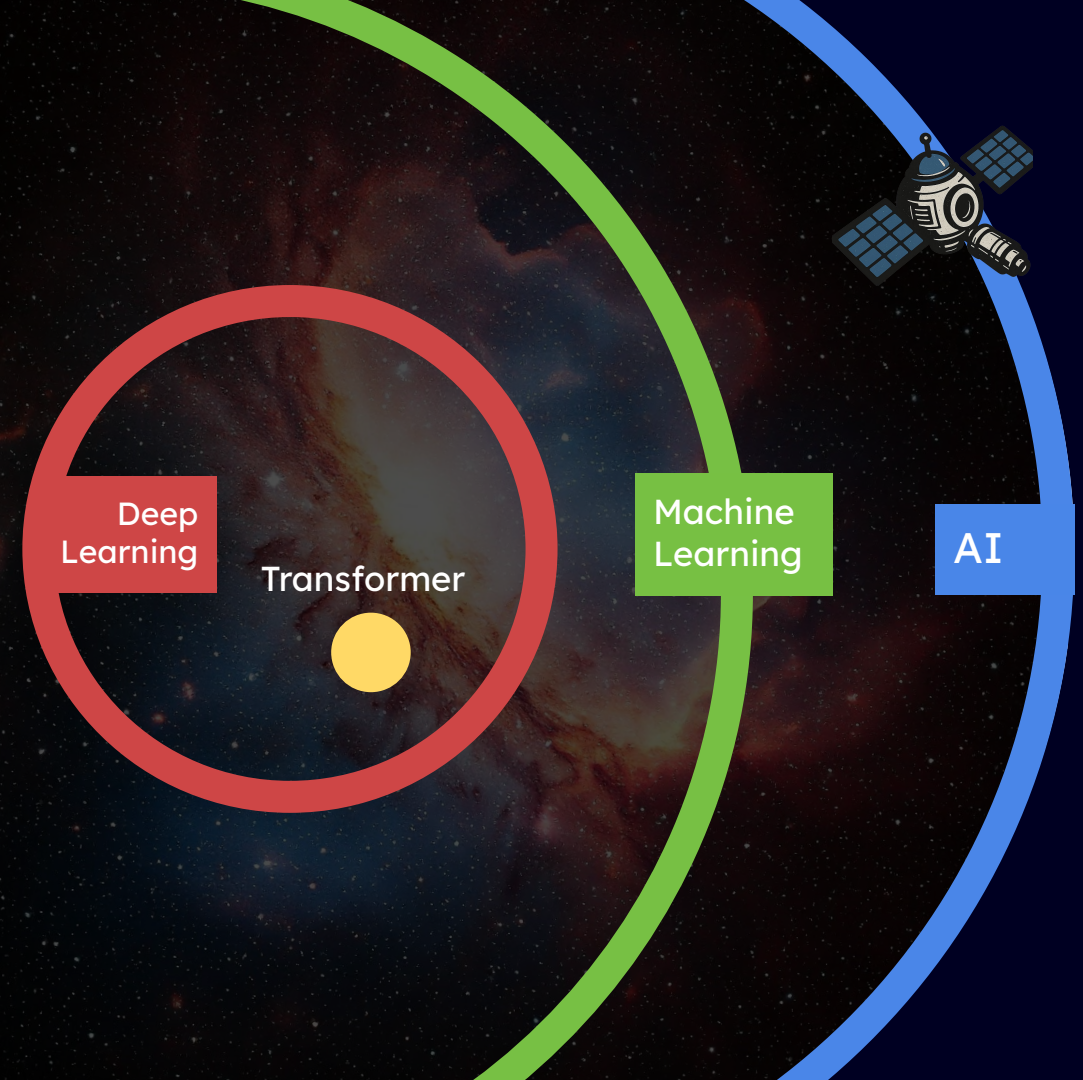
Nicolò Festa

Developer @ Monade


AI


Machine
Learning
AI


Deep Learning
Machine Learning
AI

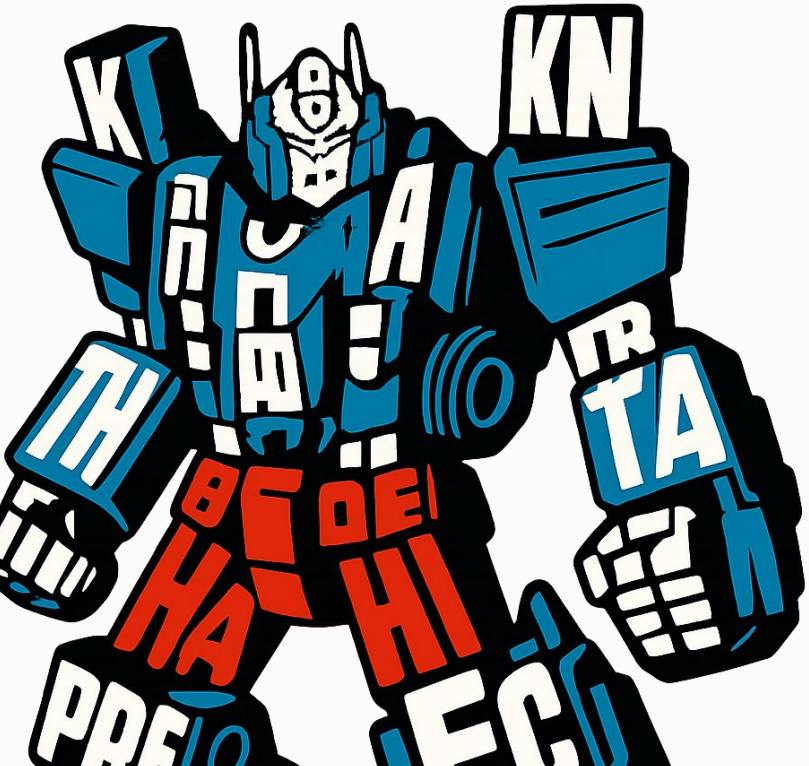

Deep Learning
Transformer
Machine Learning
AI

Transformer

2017 - Attention is all you need (Google)

Nascono per risolvere task di traduzione

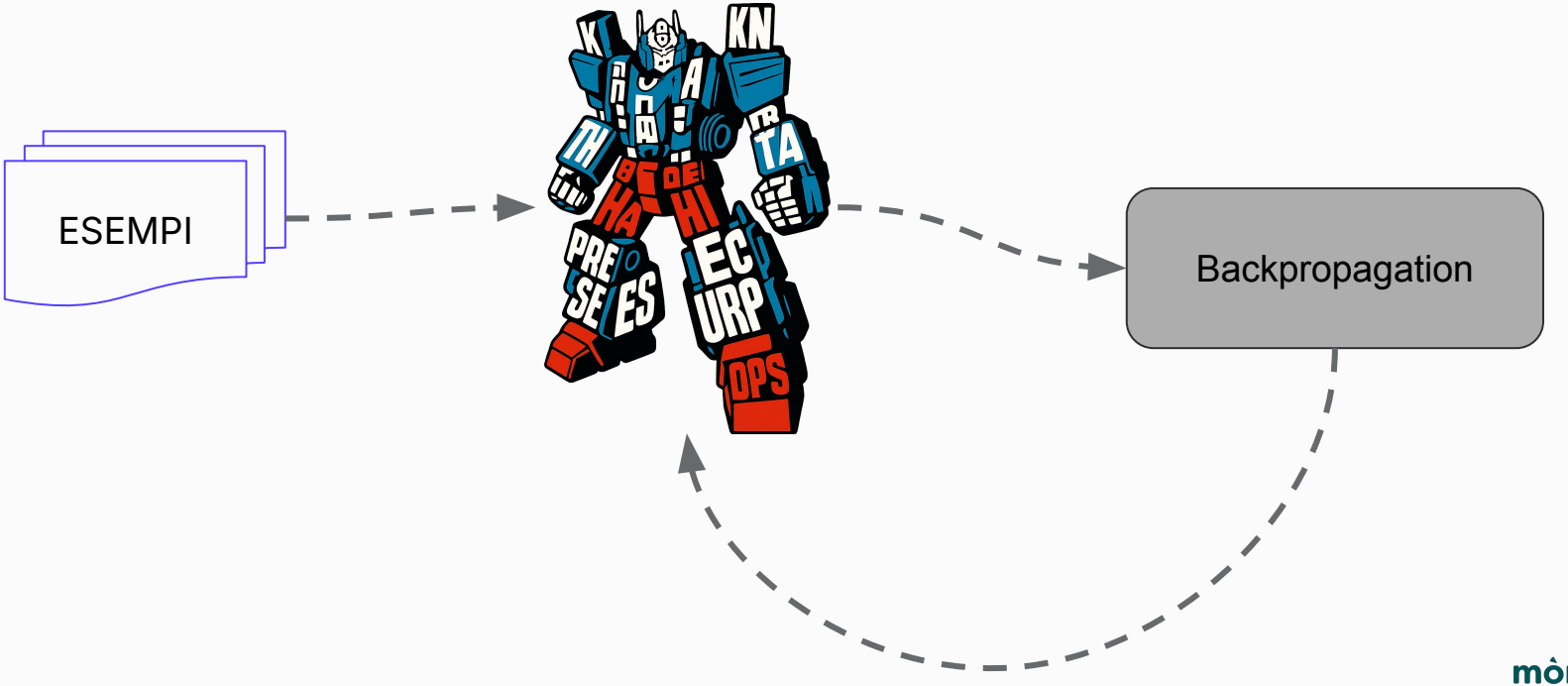
Ma che cos’è un Transformer?

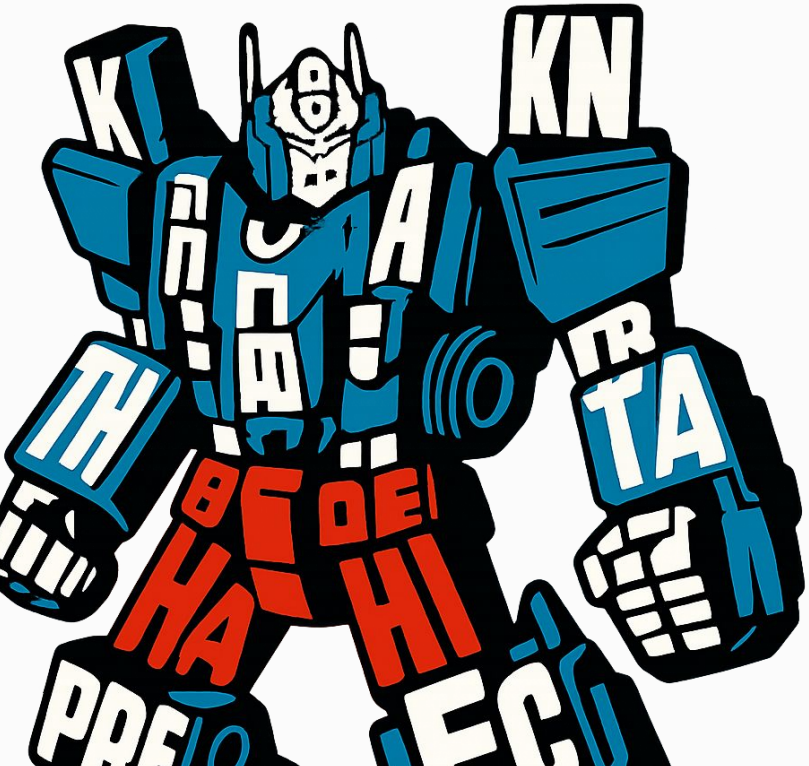


Modello Linguistico

- Rete neurale
- Addestrata su grandi quantità di testo.

Training Rete Neurale



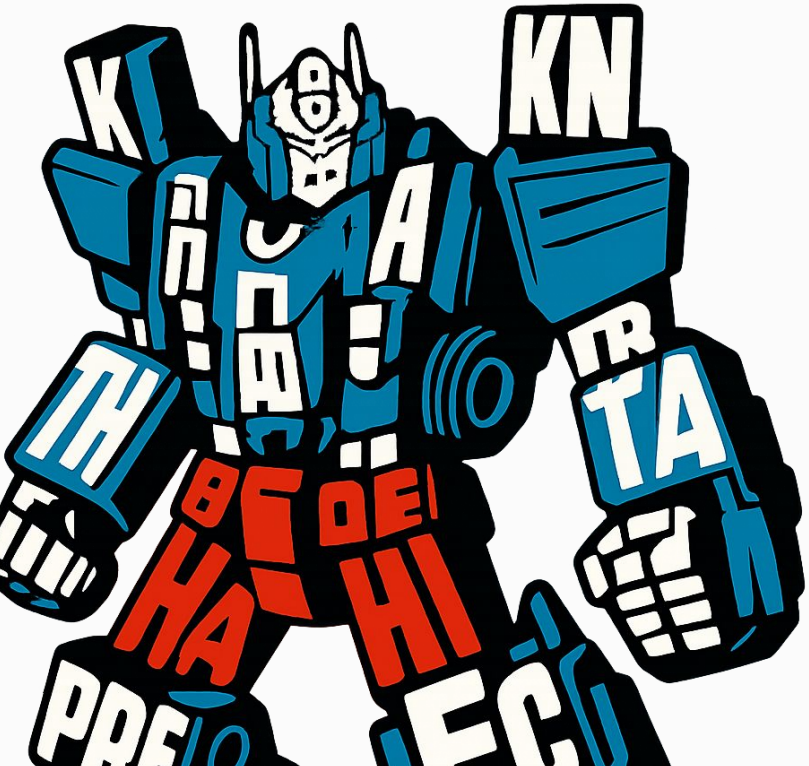


Self-supervised Learning

1 Si prende una frase:
`"Il gatto dorme sul divano."`

2 Si maschera una o più parole:
`"Il gatto dorme sul [MASK]."`

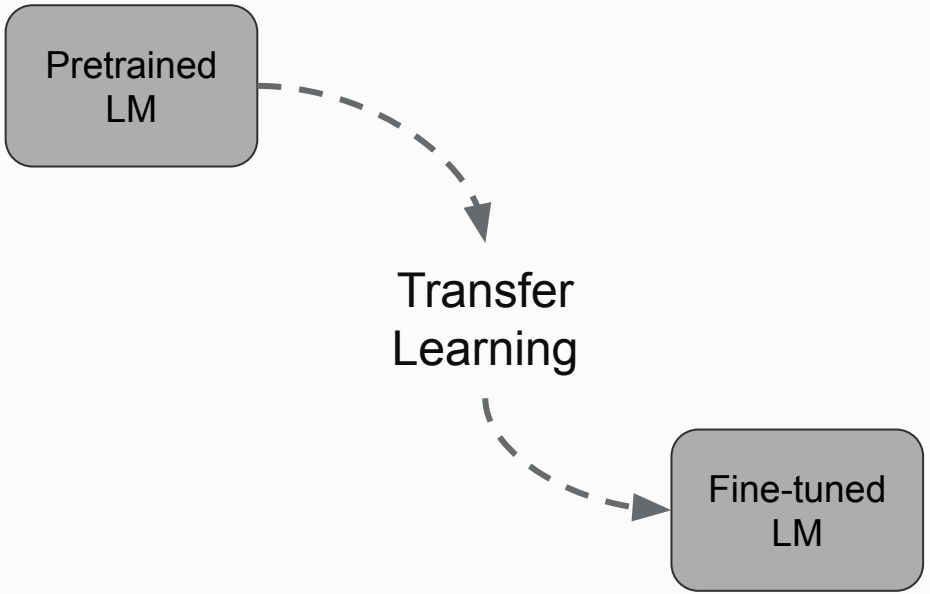
3 Il modello deve predire la parola mancante usando il contesto:
`"Il gatto dorme sul divano."`

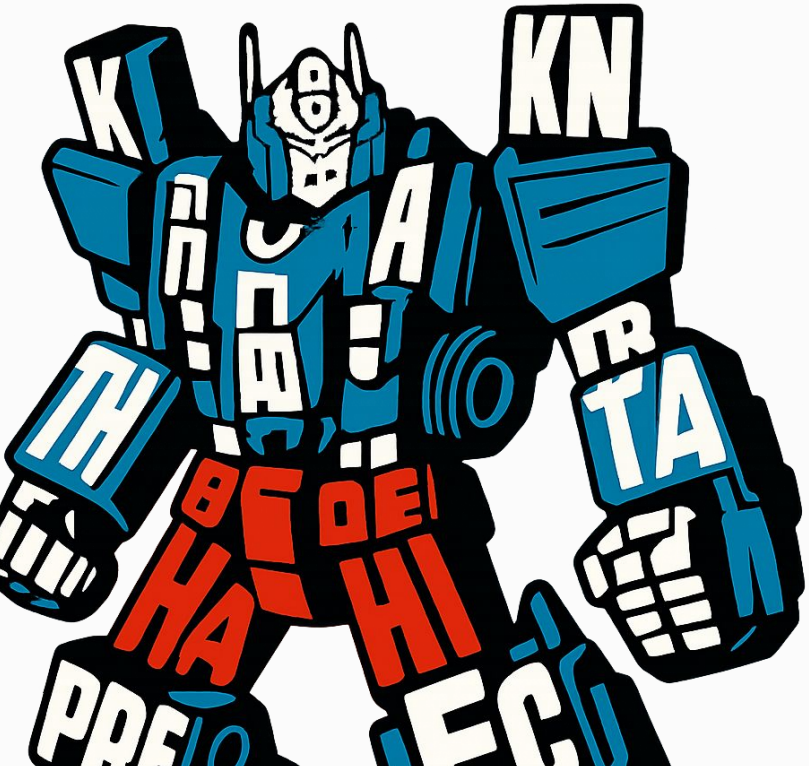


Pre Training

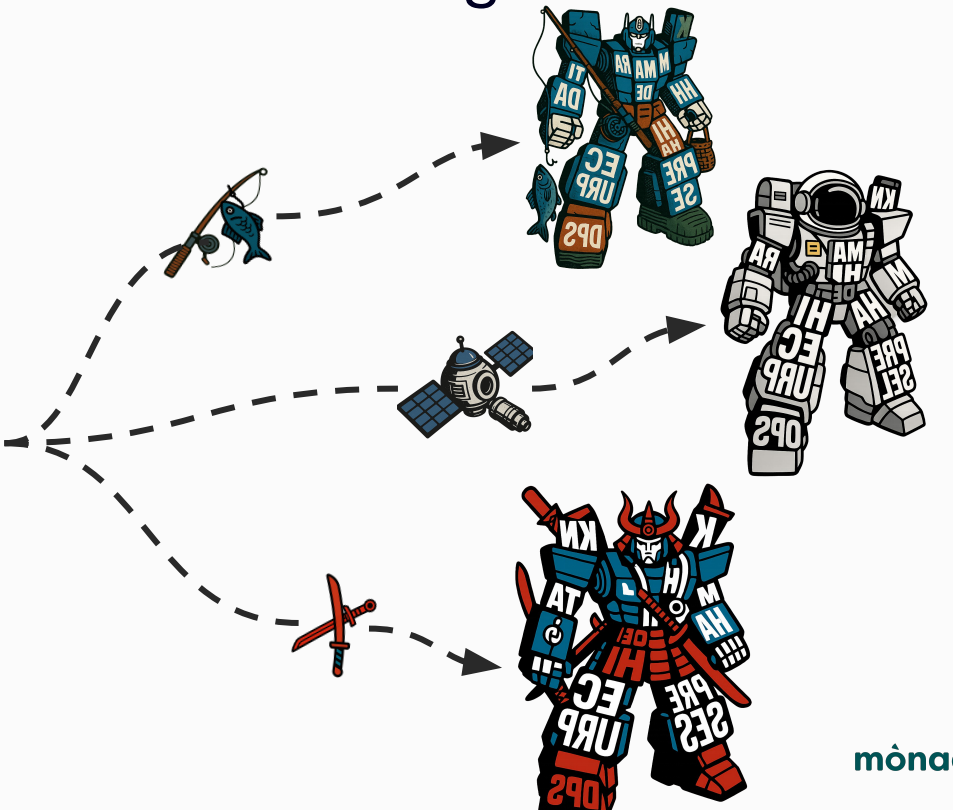
- Pre-trained Model
- Rappresentazione statistica del linguaggio

Fine-Tuning

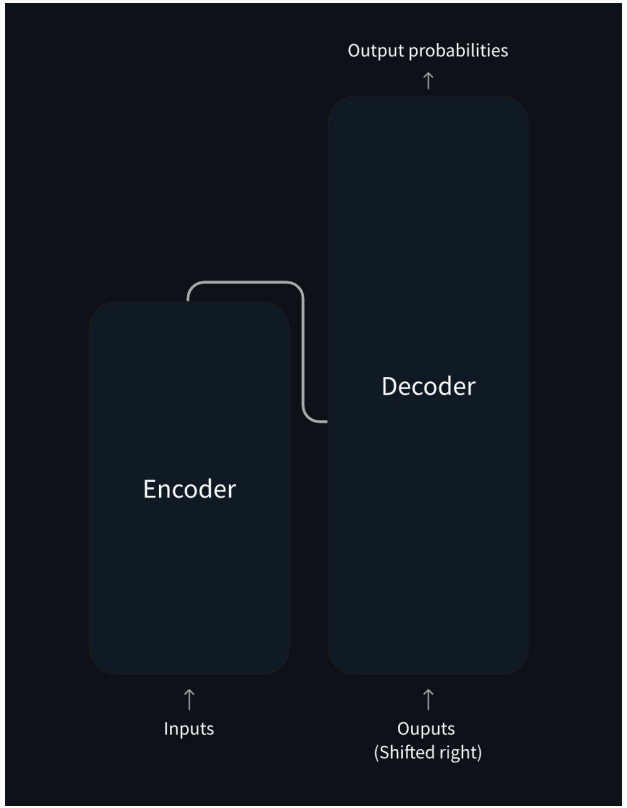




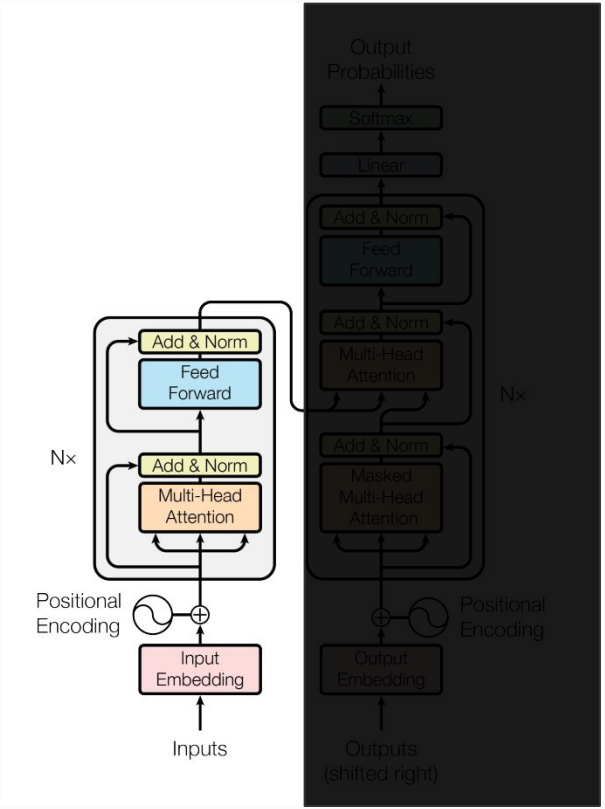
Fine-Tuning



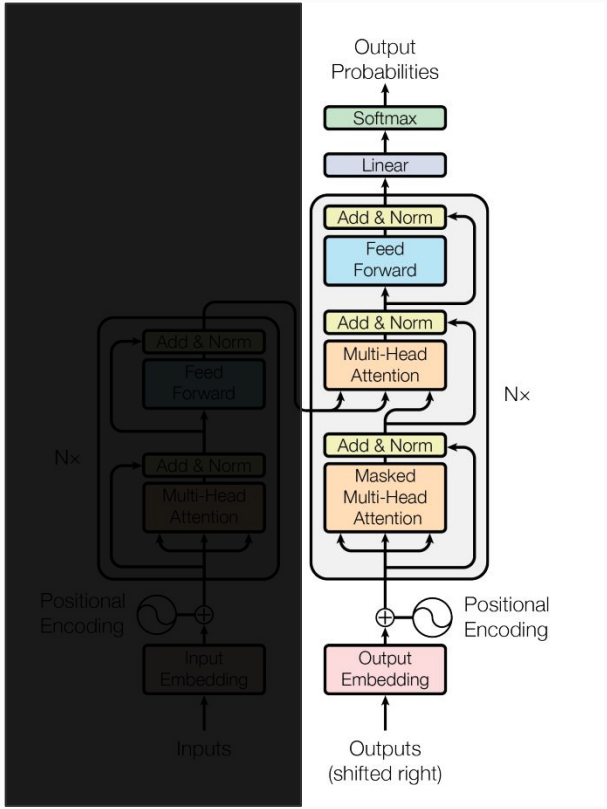
Smontiamo un Transformer


Output probabilities
Decoder
Encoder
Inputs
Ouputs
(Shifted right)

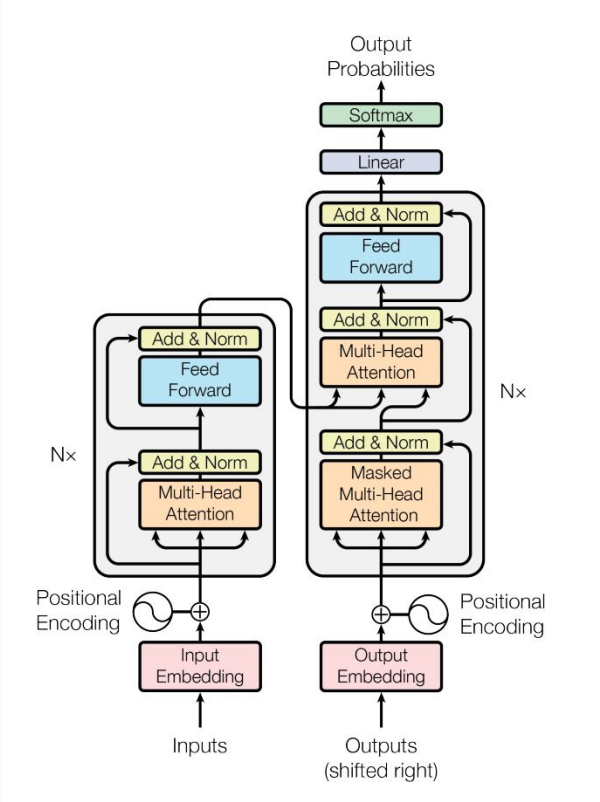
Encoder


Output Probabilities
Softmax
Linear
Add & Norm
Feed Forward
Add & Norm
Multi-Head Attention
Add & Norm
Masked Multi-Head Attention
Nx
Positional Encoding
Output Embedding
Outputs (shifted right)
Add & Norm
Feed Forward
Add & Norm
Multi-Head Attention
Nx
Positional Encoding
Input Embedding
Inputs

Decoder


Output Probabilities
Softmax
Linear
Add & Norm
Feed Forward
Add & Norm
Multi-Head Attention
Add & Norm
Masked Multi-Head Attention
Nx
Positional Encoding
Output Embedding
Outputs (shifted right)
Add & Norm
Feed Forward
Add & Norm
Multi-Head Attention
Nx
Positional Encoding
Input Embedding
Inputs

Encoder-Decoder



Tokenizer

Il tokenizer prende una stringa di testo ("Il gatto salta") e la spezza in unità chiamate token. A seconda del tipo di tokenizer, questi token possono essere:

parole intere → es. `["Il", "gatto", "salta"]`

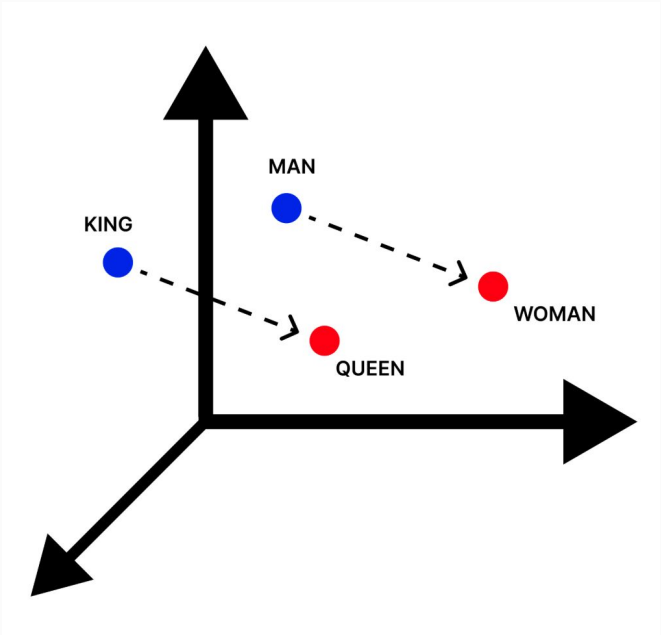
sottoparti di parole (subword) → es. `["Il", "gatt", "o", "salt", "a"]`

singoli caratteri → es. `["I", "l", " ", "g", "a", "t", "t", "o", ...]`

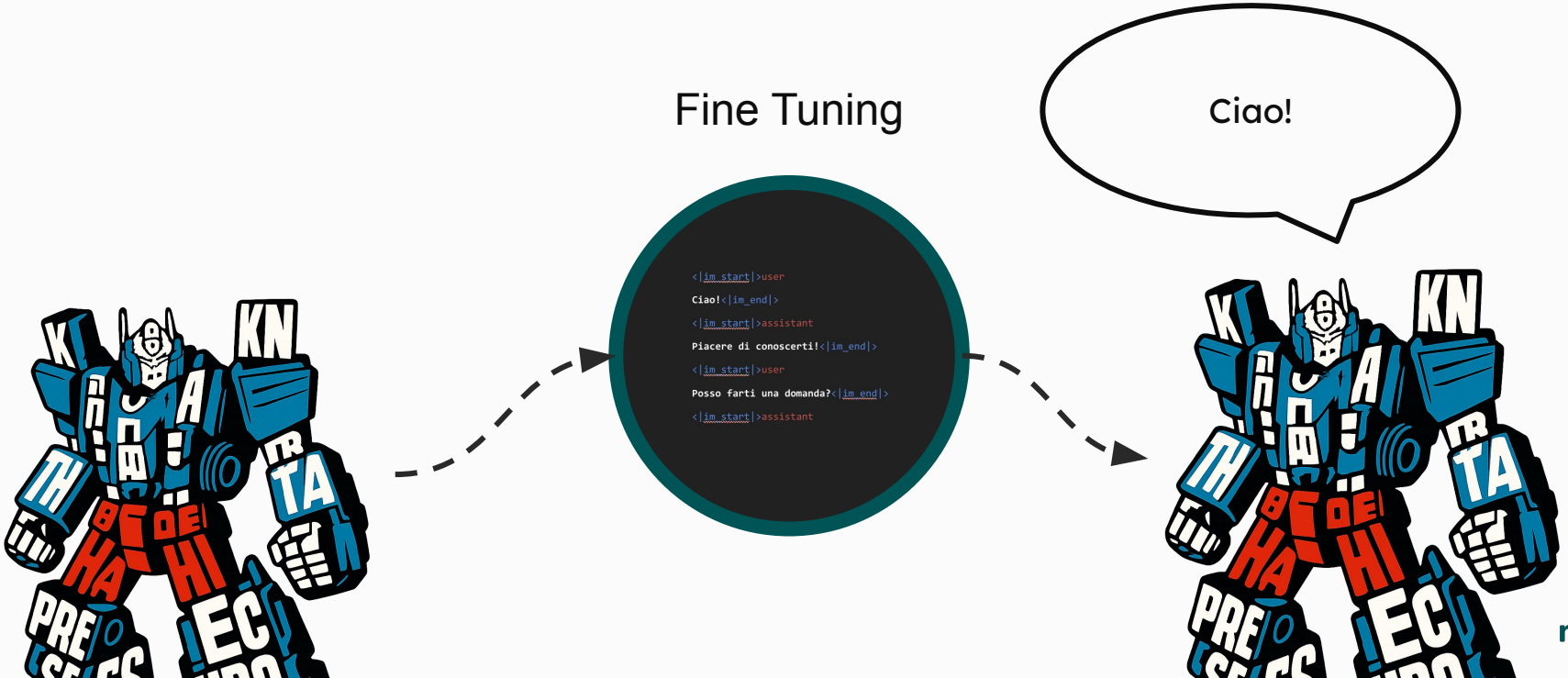
Word Embedding

Rappresentazione di parole tramite vettori N-dimensionali.

Concetti simili sono vicini nello spazio.



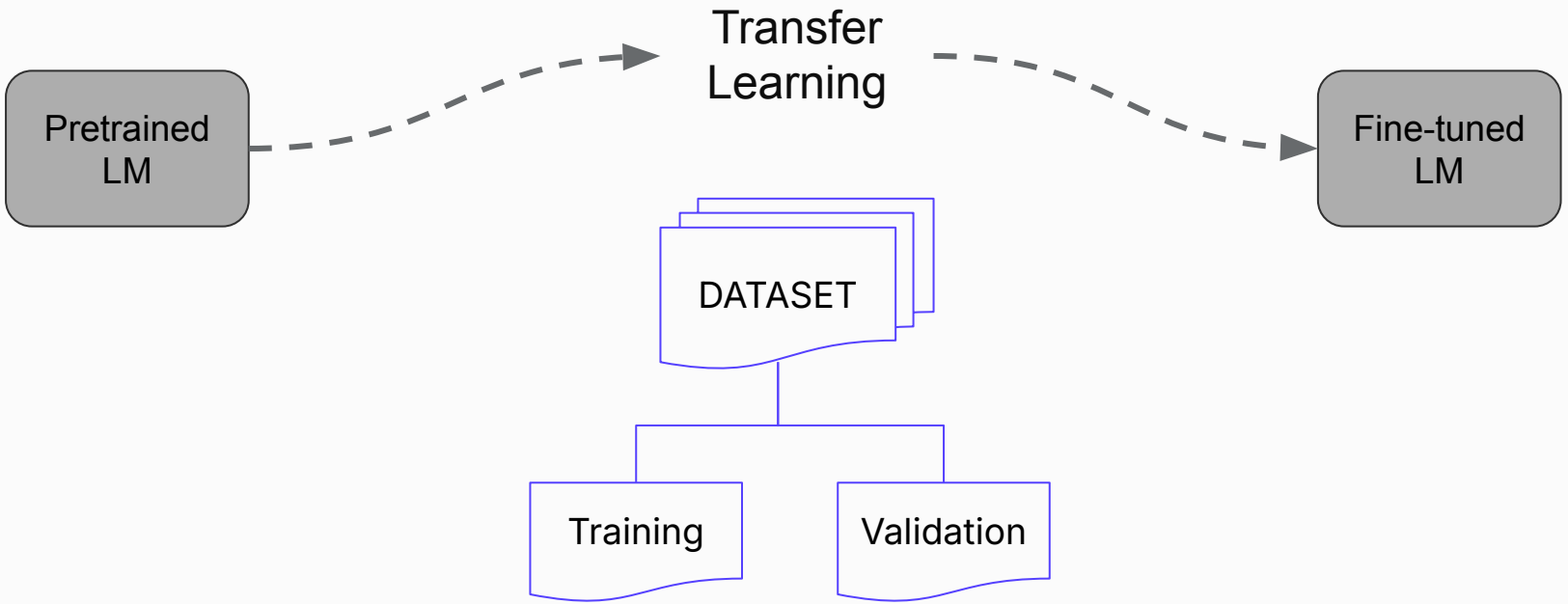
Instruct Model



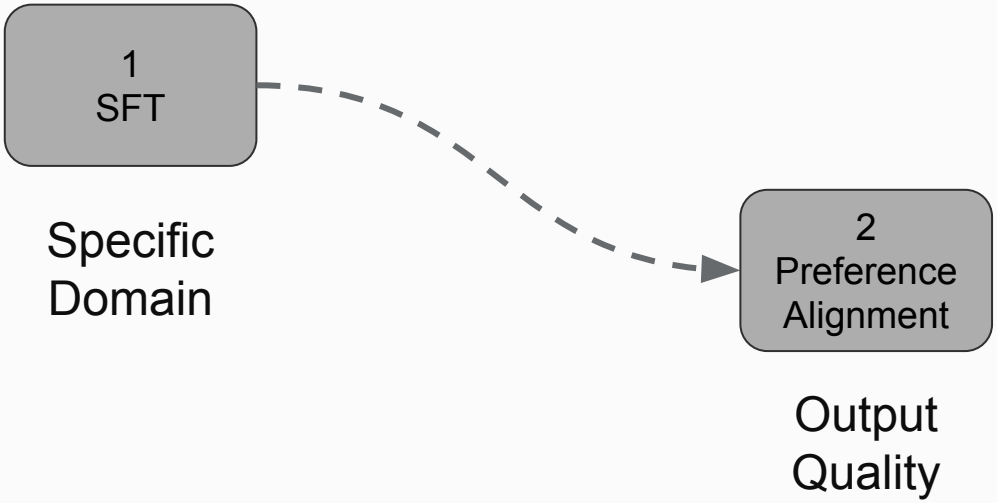
Chat Template

```
<|im_start|>user
Ciao!<|im_end|>
<|im_start|>assistant
Piacere di conoscerti!<|im_end|>
<|im_start|>user
Posso farti una domanda?<|im_end|>
<|im_start|>assistant
```

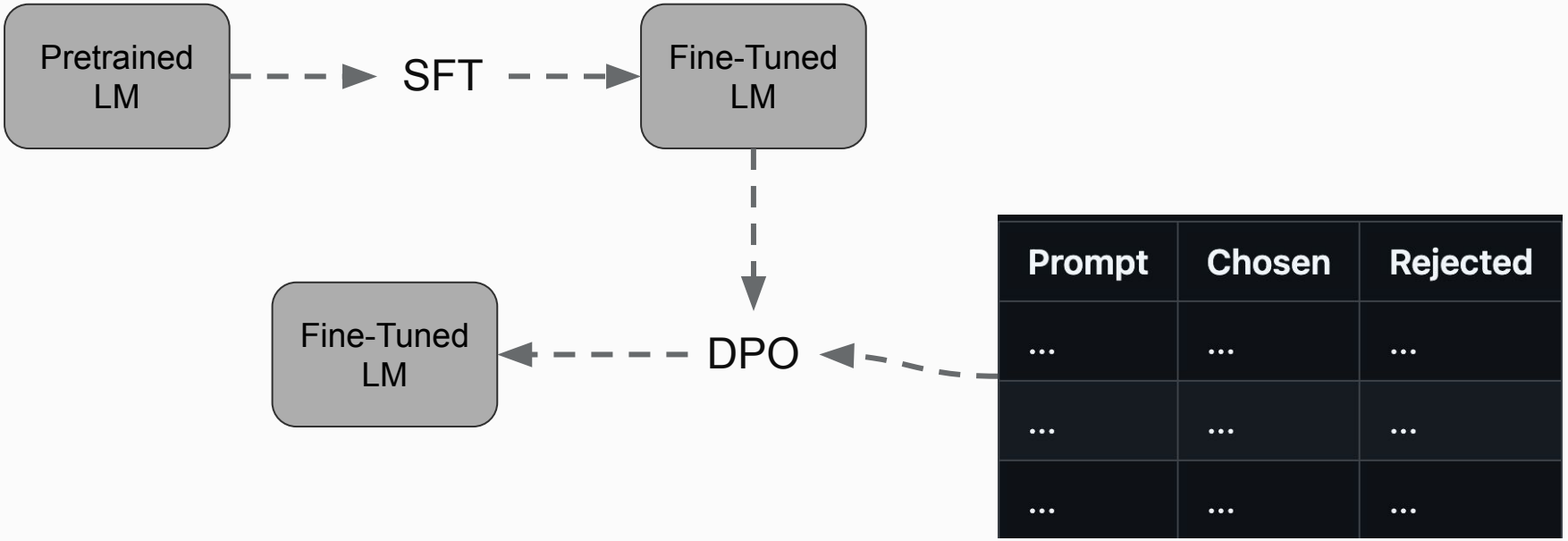
SFT
Supervised Fine-Tuning



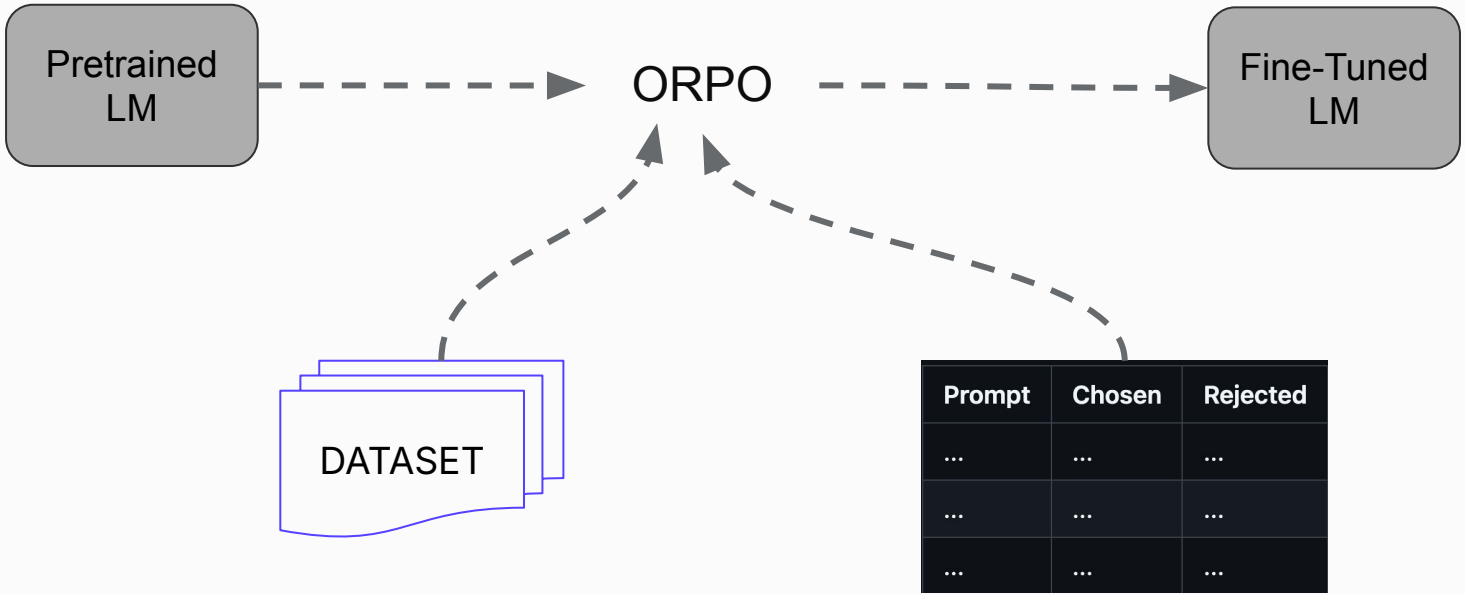
Preference Alignment



DPO
Direct Preference Optimization



ORPO
Odds Ratio Preference Optimization



PEFT
Parameter-Efficient Fine-Tuning

Fine-tuning Tradizionale

- Aggiorna tutti i parametri del modello.
  Per modelli grandi richiede enormi risorse GPU.
- Dobbiamo salvare copie separate dei modelli.
- Rischio di perdere capacità iniziali del modello di partenza.

PEFT

- Modifica solo una piccola parte del modello.
- Devo salvare solo i parametri modificati.
- Non perdo le capacità iniziali del modello.

LoRA
Low-Rank Adaption

- Congela i parametri del modello di partenza.
- Riduce il numero dei parametri addestrabili fino al 90%.
- Inietta matrici di decomposizione addestrabili.

$$\begin{pmatrix} a_{1,1} & a_{1,2} & a_{1,3} & a_{1,n} \\ a_{2,1} & a_{2,2} & a_{2,3} & a_{2,n} \\ a_{3,1} & a_{3,2} & a_{3,3} & a_{3,n} \\ a_{m,1} & a_{m,2} & a_{m,3} & a_{m,n} \end{pmatrix}$$

$$\begin{pmatrix} a_{1,1} & a_{1,2} & a_{1,3} & a_{1,n} \\ a_{2,1} & a_{2,2} & a_{2,3} & a_{2,n} \\ a_{3,1} & a_{3,2} & a_{3,3} & a_{3,n} \\ a_{m,1} & a_{m,2} & a_{m,3} & a_{m,n} \end{pmatrix}$$

Let’s Code!

github.com/monade/smol-but-mighty