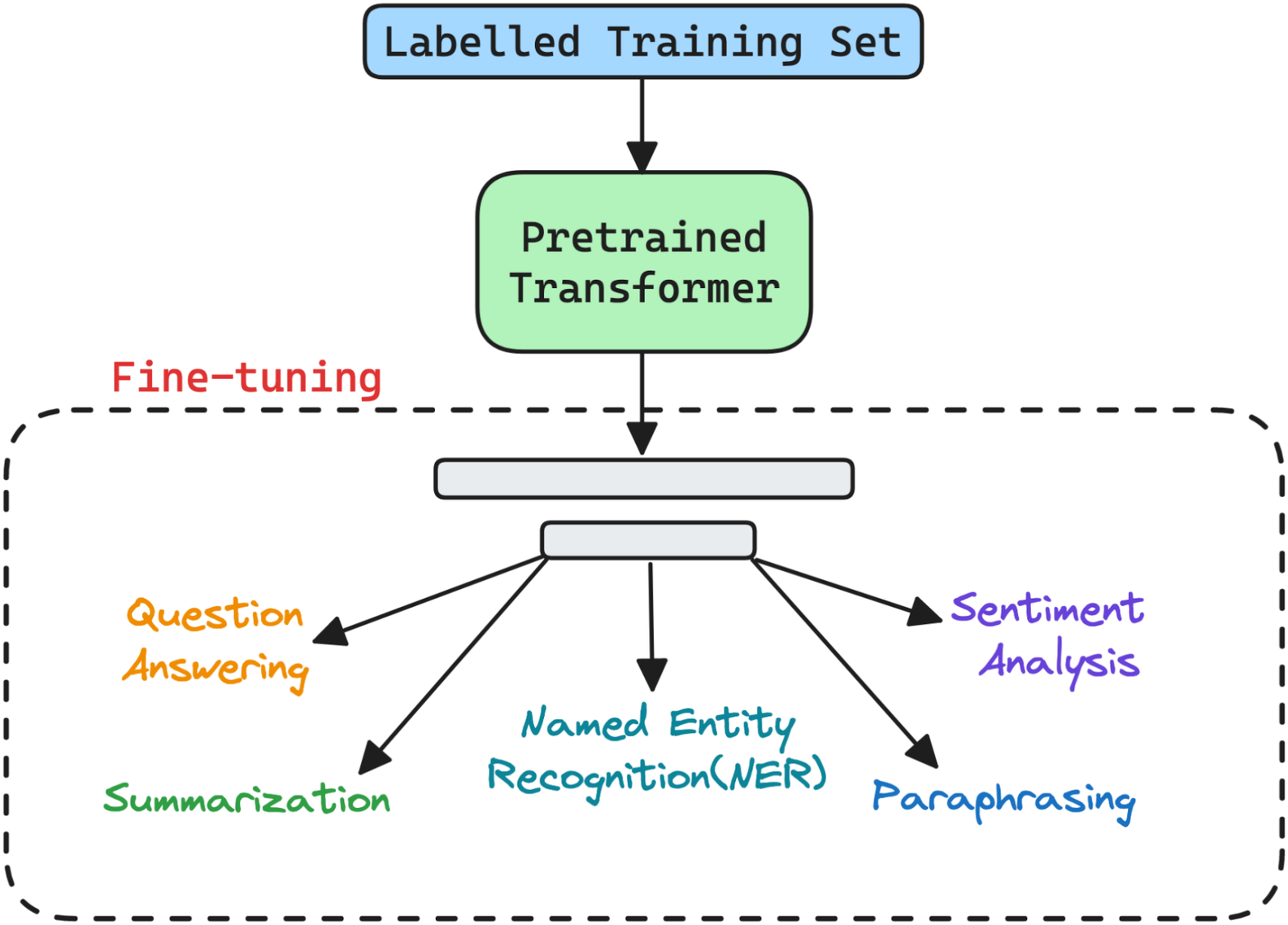
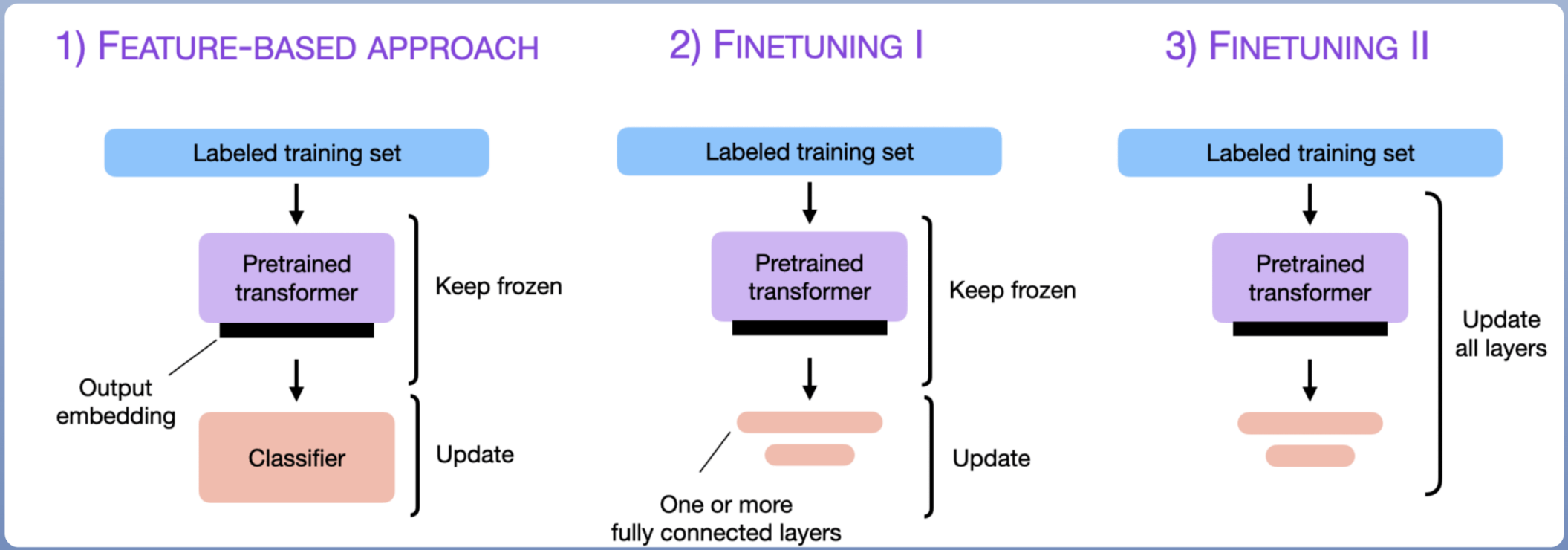
Fine-tuning an LLM!🚀



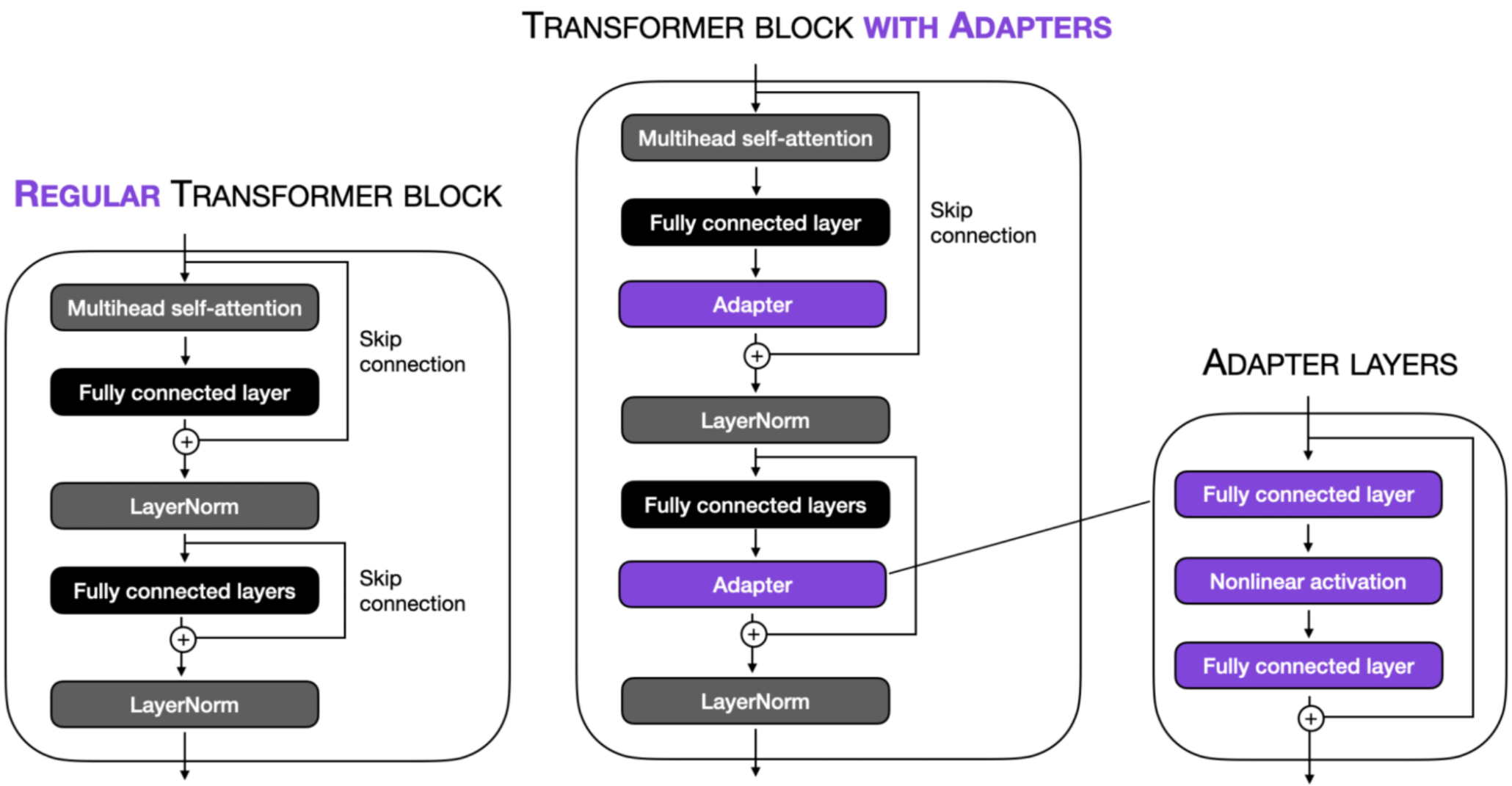
@akshay_pachaar

Fine-tuning Strategies



Transformer with Adapters

Only the adapter layers in each block are trainable

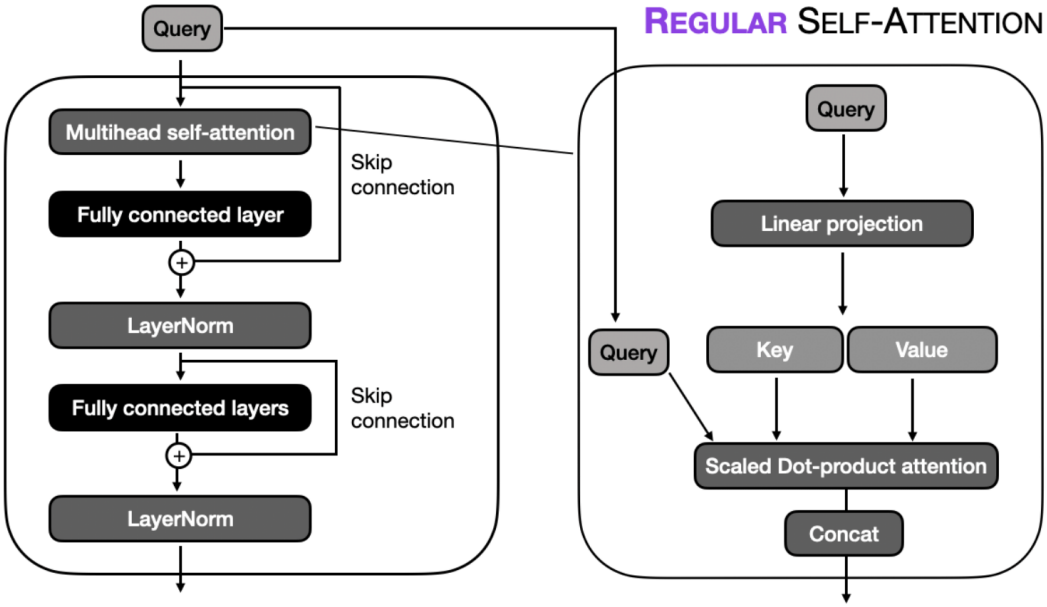


adapter.py

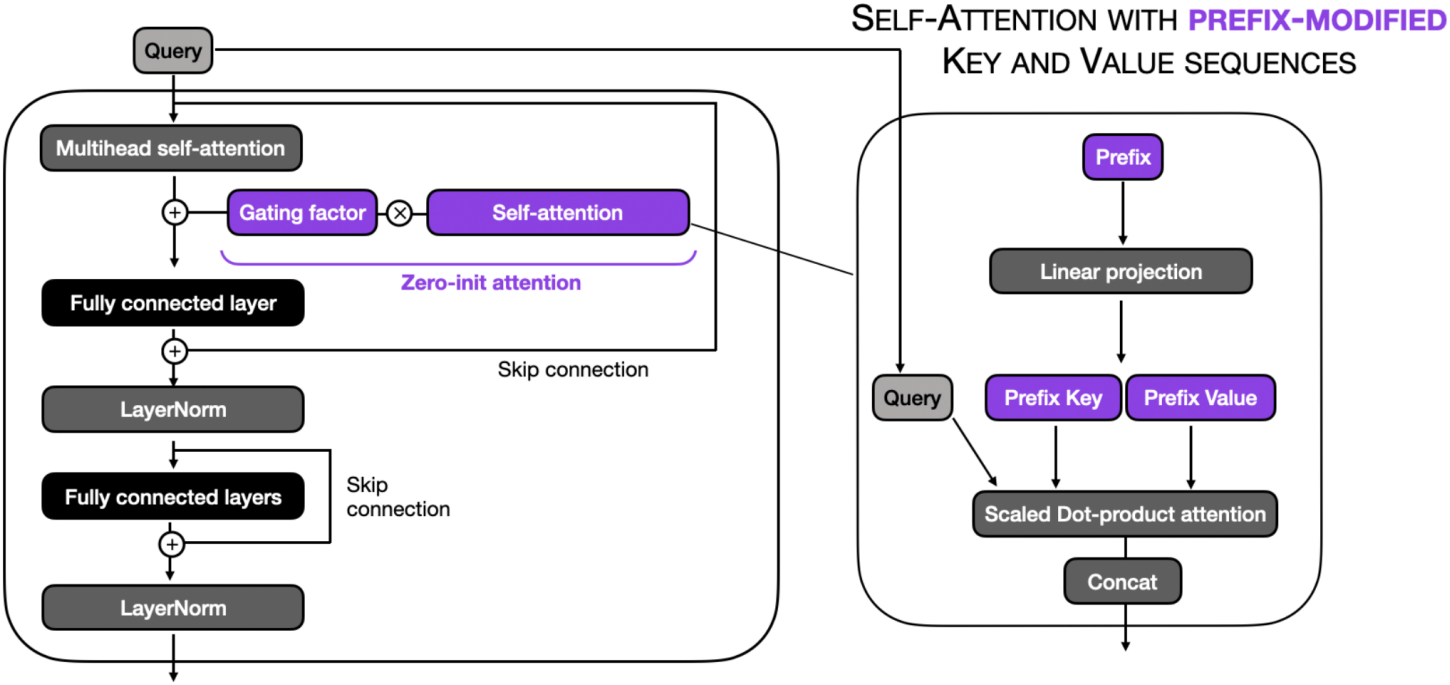
@akshay_pachaar

```python
def transformer_block_with_adapter(x):
residual = x
x = self attention(x)
x = AdapterLayers (x) # adapter
x = LayerNorm(x + residual)
residual = x
x = FullyConnectedLayers(x)
x = AdapterLayers(x) # adapter
x = LayerNorm(x + residual)
return X
```

Regular Transformer block



Transformer block with LLAMA Adapter



llama_adapter.py

```python
def transformer_block_with_llama_adapter(x, gating_factor, soft_prompt):
residual = x
y = zero_init_attention(soft_prompt, x) #llama-adapter: prepend prefix
x = self attention(x)
x = x + gating_factor * y #llama-adapter: apply zero_init_attention
x = LayerNorm(x + residual)
residual = x
x = FullyConnectedLayers (x)
x = LayerNorm(x + residual)
return X
```

@akshay_pachaar

That's a wrap!

If you interested in:

- Python 🐍
- Data Science 📈
- Machine Learning 🤖
- MLOps 🛠️
- NLP 🗣️
- Computer Vision 📹
- LLMs 🧠

Follow me on LinkedIn ✔️
Everyday, I share tutorials on above topics!

Cheers!! 🙂

@akshay_pachaar