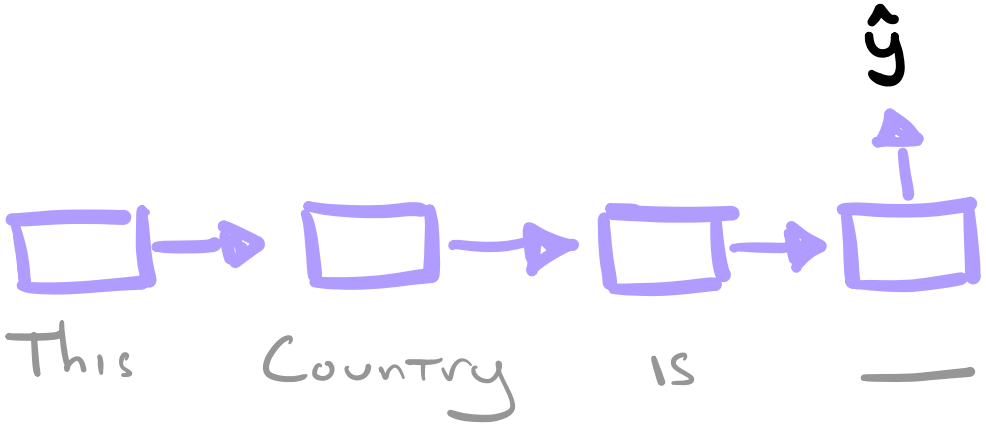
DS 4440 RNNs → Contextual embeddings → Transformers

Autoregressive generative model:

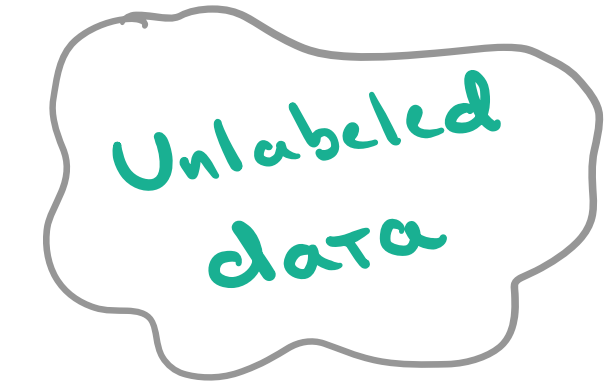
$$P(x_T | x_{T-1} \cdots x_1)$$

$$P(x) = \prod_T P(x_T | x_{<T})$$

e.g. Language Modeling.

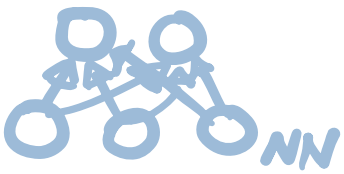


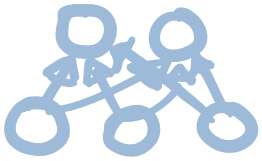
Language Modeling is a kind of Self-Supervised learning, often used for Pre-Training



e.g. the Internet

+ Self Supervised loss | Pretrain





+ $\mathcal{L}$ and (x, y) | Fine-Tune

OR

(x, y) | In Context learning (later)

Contextualized Word Vectors

Recall (9.30 lecture) that W2V yields embeddings of words

"Cat" → W2V → [x y z] | Static; always x y z

"Cute Cat" → [x y z]

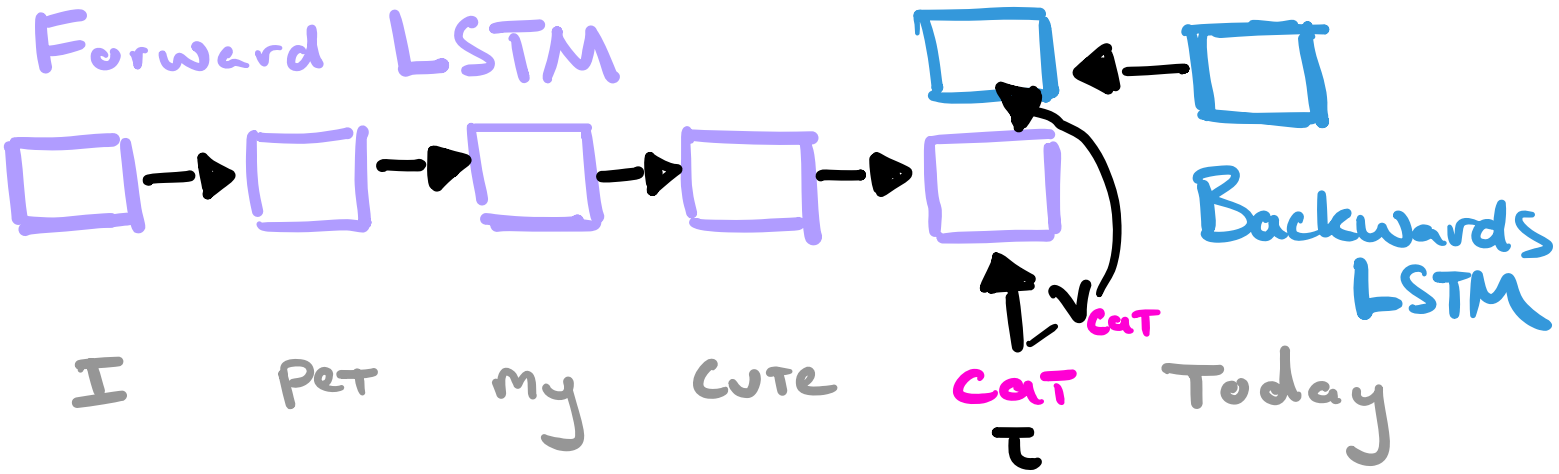
"fat Cat" → [x y z]

IDEA Train an RNN for language modeling; extract contextualized embeddings

ELMo

Embeddings from Language Models

[Peters et al., '18]



$$\mathcal{L} = -P(cat \mid v_{cat}, \overrightarrow{h}_t) - P(cat \mid v_{cat}, \overleftarrow{h}_t)$$

Self Supervised! LM

$$ELMo_{cat} = [v_{cat}; \overrightarrow{h}_t; \overleftarrow{h}_t]$$

Can now use $ELMo_{CAT}$ vectors as features for downstream tasks

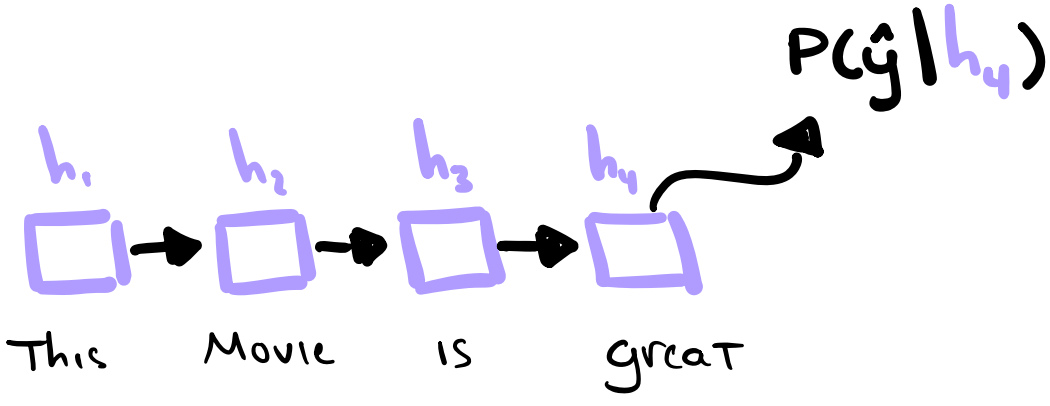
fine-Tuning!

This works pretty well! But, ELMo is based on RNNs and so it's Slooooooow due to Sequential processing

Enter Transformers which dive away with recurrence using Self-attention.

(Vanilla)

Attention!



Idea Weight hidden states

$$a_j = \underset{\langle 1 \times h \rangle \langle h \rangle}{W_a \cdot h_j}$$

$$\bar{h} = \sum_{j=2}^{n} \alpha_j h_j$$

$\alpha = \text{SoftMax}(a)$ | $P(y|\bar{h})$

Self-Attention

Generalizing: Assume keys k and queries q.

$$a_j = S(k_j, q) \qquad \alpha = \text{SoftMax}(a)$$

$$\bar{h} = \sum_j \alpha_j v_j$$

e.g. $(k_j \cdot q)/\sqrt{d}$ "dot product Attn"

Scale by dims

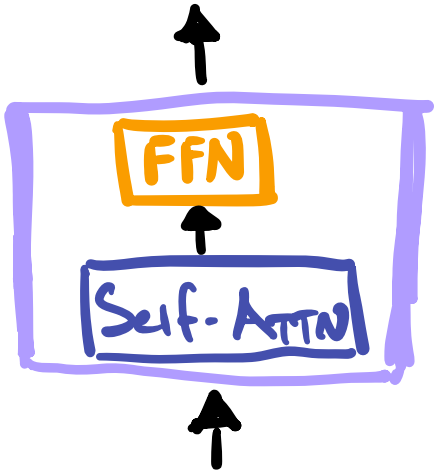
or

$$v_j^T \tanh(W_k k_j + W_q q)$$ "MLP Attn"

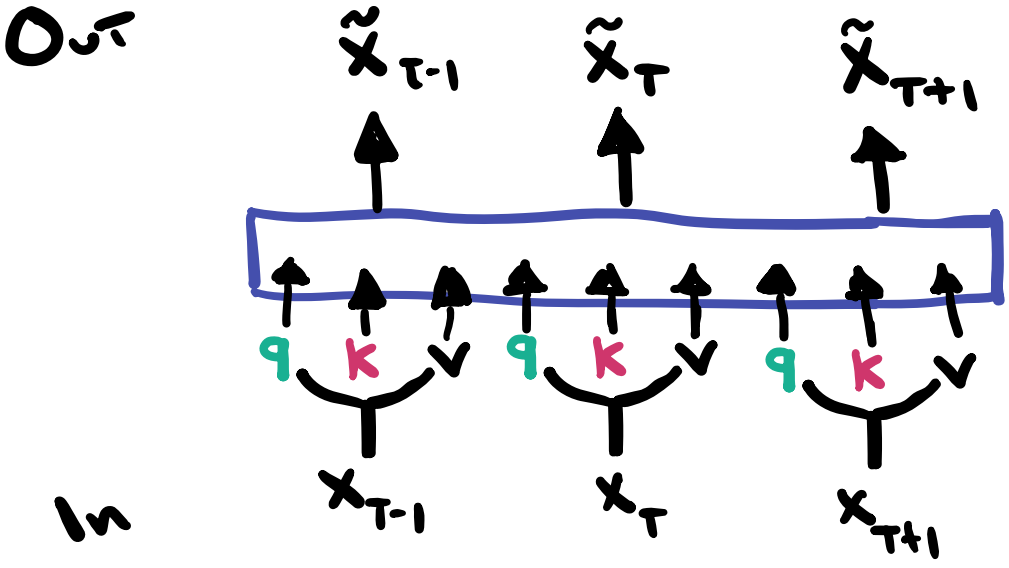
Transformers

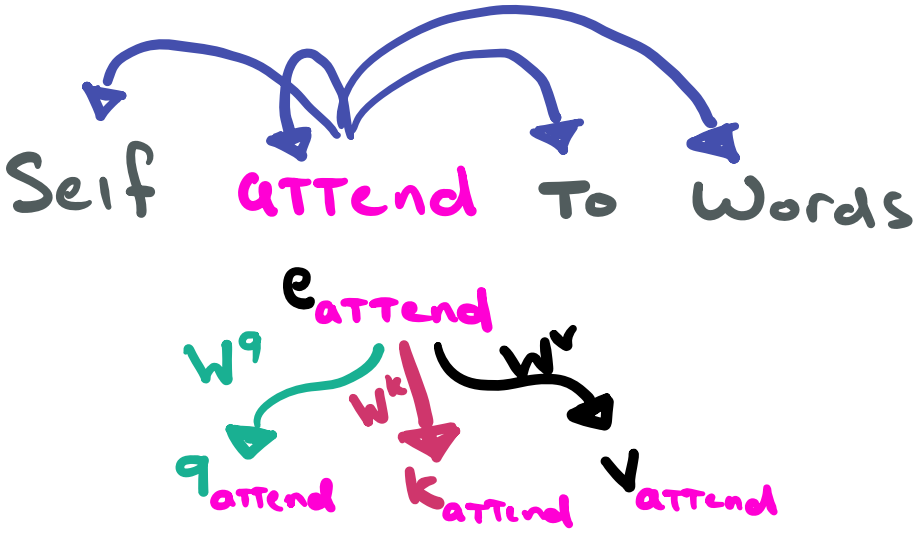
[Vaswani et al '17]

Replace recurrence with repeated blocks of self-attention.



Transformer Block





attend

Self $s(K_{self}, q_{attend})$

attend $s(K_{attend}, q_{attend})$

to $s(K_{to}, q_{attend})$

Words $s(K_{words}, q_{attend})$

$\rightarrow sm([\quad])$

α^{attend}

$$\tilde{x}_{attend} = \sum_j \alpha_j^{attend} v_j$$

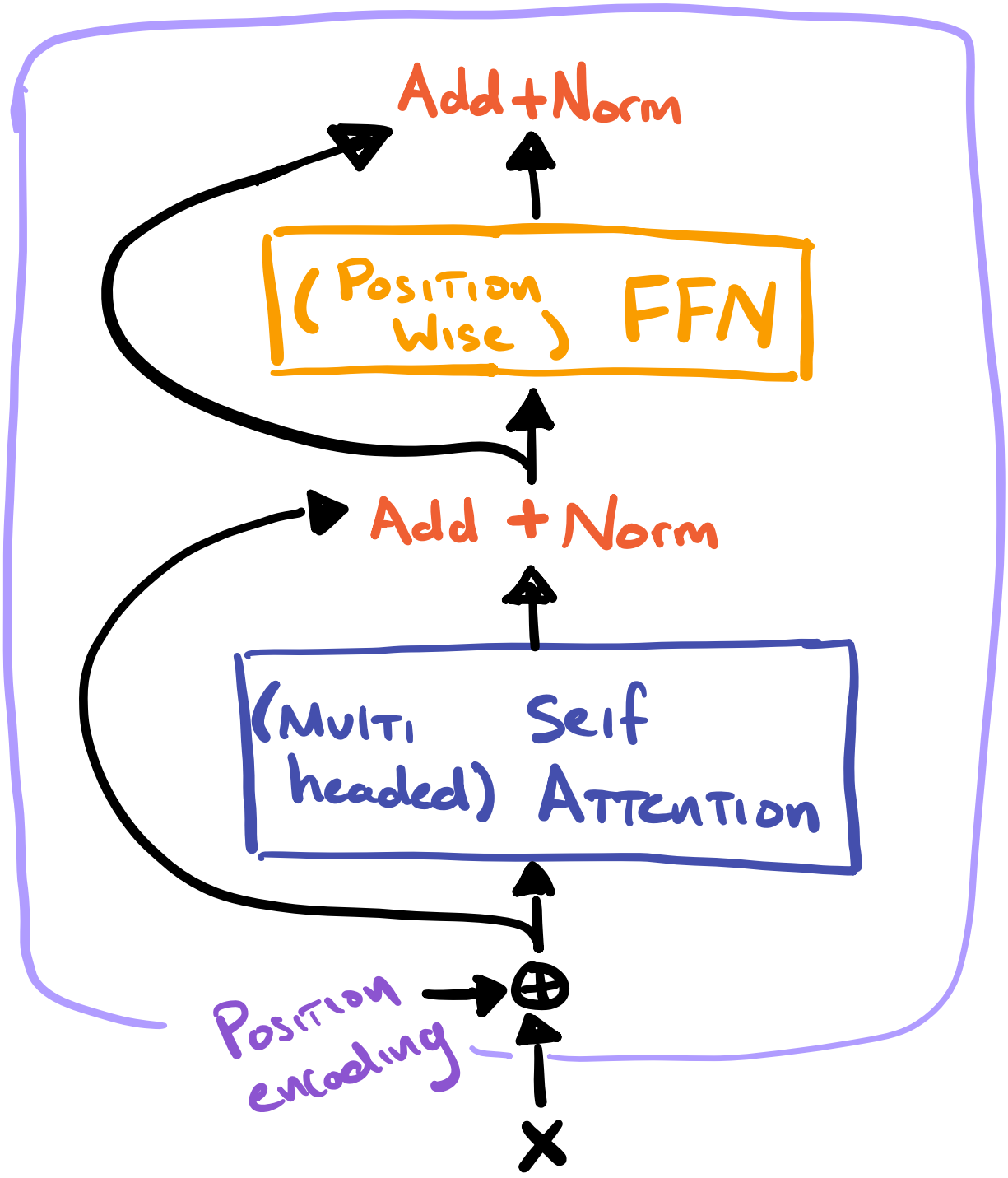
(See CoLab exercise!)

In Multi-headed Attention we define L attention heads with independent params W_q^l, W_k^l, W_v^l.

Outputs are combined (e.g., $\oplus$).

After Self Attention, FFN (Position Wise)

$$FFN(\tilde{x}_j)$$



Another detail is add + Norm which provides a residual shortcut and imposes a layer norm, which is like batch norm (10/7 lecture) but transposed.

Positional encodings address the issue of token positions (why is this needed?)