DS 4440

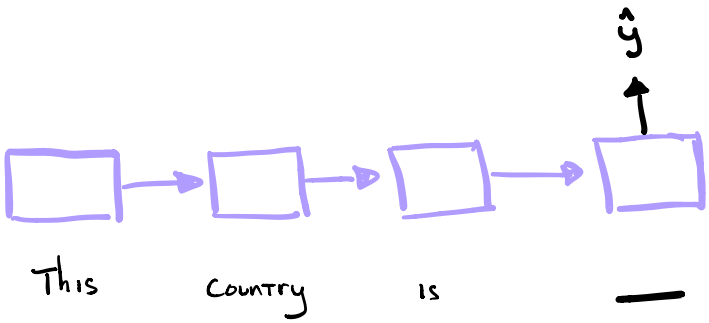
RNNs → Contextual embeddings

→ Transformers

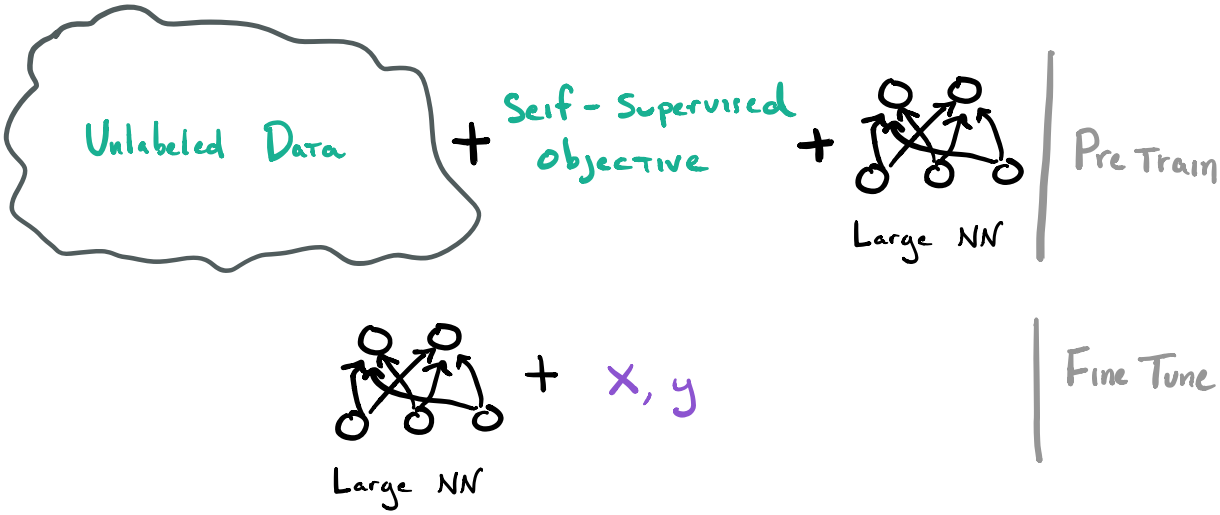
Sequence modeling

$$P(x_t \mid x_{t-1}, \dots, x_1)$$

We will use language modeling (LM) as a motivating example.



One reason we might care about LM is for pretraining via self-supervision.



Contextualized word vectors

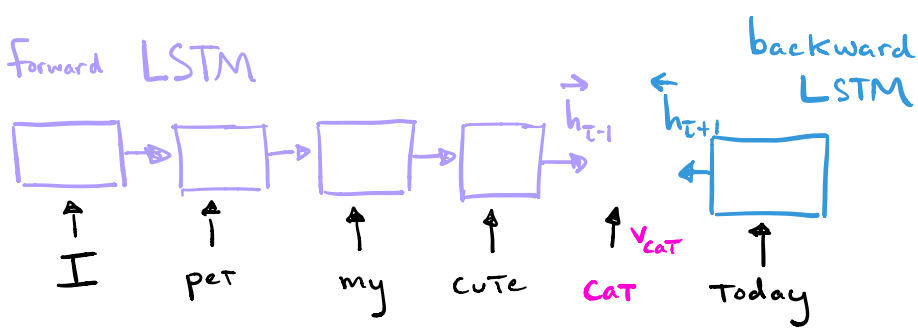
Recall (lecture 8) that something like W2V induces embeddings for words (discrete inputs)

"cat" → [x y z] Static; always [x y z]

"cute cat" → [x y z] "fat cat" → [x y z]

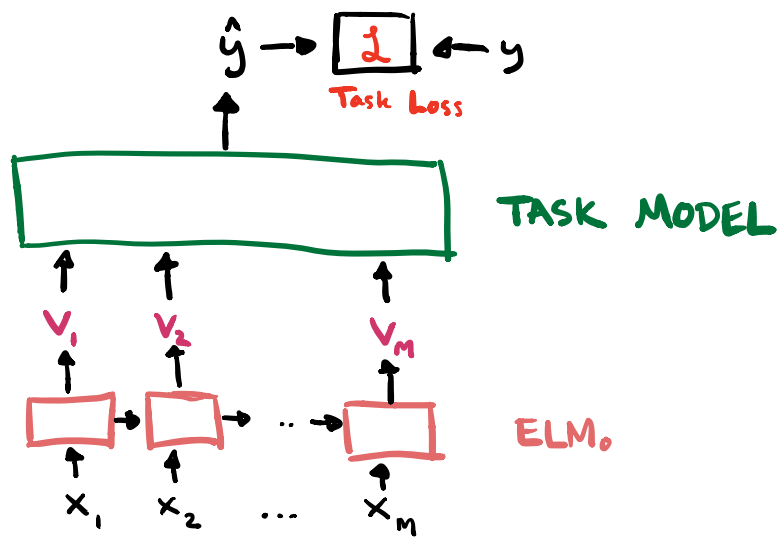
Idea Train an RNN for language modeling, then extract contextualized embeddings from it.

ELMo Embeddings from Language Models (Peters et al., '18)



Jointly optimize $\underbrace{P(\text{cat} \mid v_{cat}, \vec{h}_{t-1})}_{\text{forward LSTM params} + v_{cat} \text{ (shared)} + \text{SoftMax (shared)}} + P(\text{cat} \mid v_{cat}, \overleftarrow{h}_{t+1})$

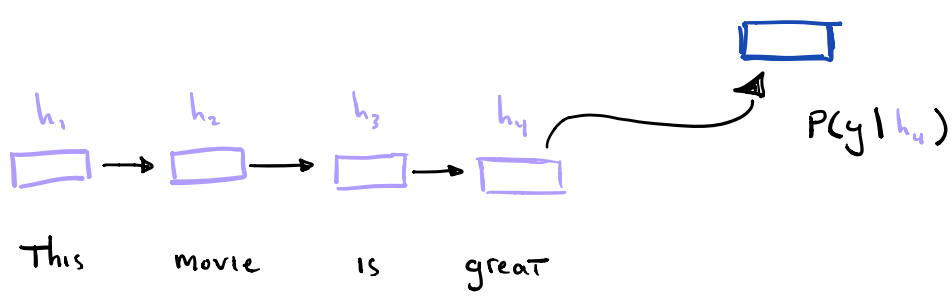
(Pre-) Train this over millions of sentences, then adapt to a target task.



Ok so ELMo still uses RNNs. One drawback to this: This necessitates sequential processing which is slow.

Transformers do away with this by instead using repeated application of self-attention.

Attention



$M = 4$

Idea Weight hidden states

$$a_j = \underset{\langle 1 \times h \rangle \langle h \times 1 \rangle}{W_a \cdot h_j} \qquad \bar{h} = \sum_{j=1}^{m} \alpha_j h_j$$

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

Self-Attention

Let's generalize the above: Assume we have a set of keys K and values V. Given a query q,

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

$$o = \sum_j \alpha_j v_j$$

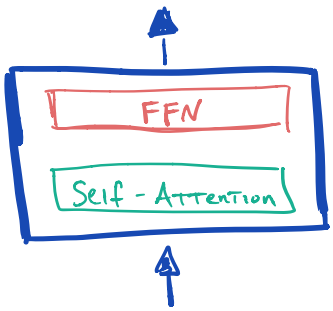
$$= (k_j \cdot q) / \underbrace{\sqrt{d}}_{\text{Scale by dims}} \quad \Big\} \text{ dot-product attention}$$

$$= v^T \tanh(W_k k_j + W_q \cdot q) \quad \Big\} \text{ MLP attention}$$

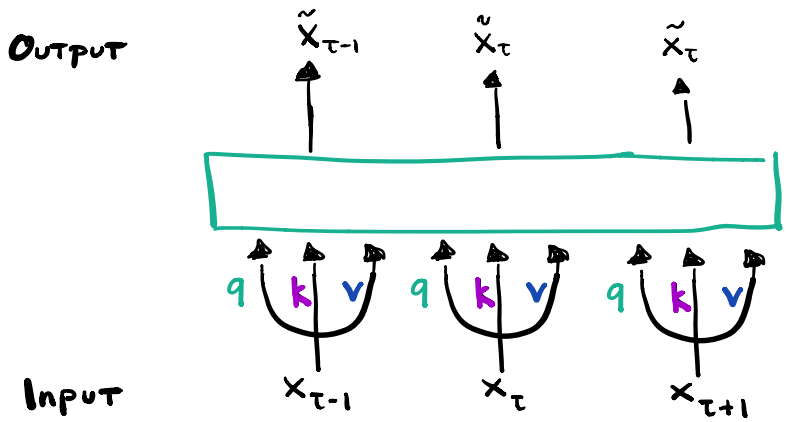
Now, **Transformers!**

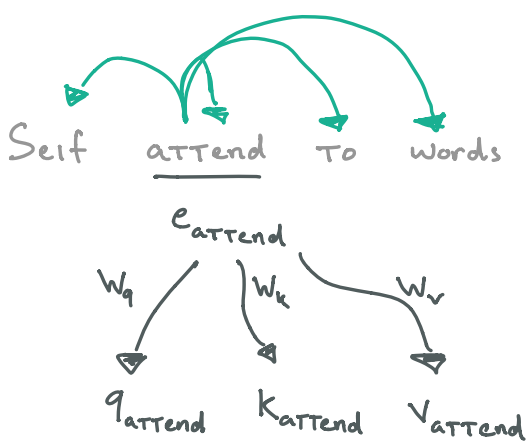
[Vaswani, et al., 2017]

Idea Replace recurrence with repeated blocks of Self-attention + feed fwd layer



Self Attention





	attend
Self	$s(k_{self}, q_{attend})$
attend	$s(k_{attend}, q_{attend})$
to	$s(k_{to}, q_{attend})$
words	$s(k_{words}, q_{attend})$

$$\begin{bmatrix} \\ \\ \\ \\ \end{bmatrix} \alpha^{attend}$$

$$\tilde{x}_{attend} = \sum_{j=1}^{4} \alpha_j^{attend} v_j$$

See CoLab!