Review of cross attn:

- lots of blocks
- unmasked
- self-attn

Encoder

Decoder

- add cross attn output to self-attn output (residual)
- masked self-attn

- e_1, e_2, e_3
- f_1, f_2, f_3
- <sos>
Using neural LMs for transfer learning

Let's consider the task of sentiment analysis.

2013:

\[ \text{this movie is great} \]

- We initialize \( c_i \) with pretrained word embeddings.
- We softmax predict positive word embeddings.
- We train from scratch.

2018:

- Issues with above: forcing the model to learn composition from a small labeled dataset is a bad idea.
- What if instead of only transferring word embs, we repurpose a neural LM to solve the downstream task.
ELMo: embeddings from language models

Step 1: pretrain an RNN LM on lots of data

Step 2: freeze LM parameters, use its representations as input to a task-specific model
ELMo architecture:

**Forward LM:**

```
<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>this</td>
<td>movie</td>
<td>is</td>
<td>great</td>
</tr>
</tbody>
</table>
```

This hidden state only knows about "this movie is".

**Backwards LM**

```
<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>great</td>
<td>is</td>
<td>movie</td>
<td>this</td>
</tr>
</tbody>
</table>
```

This hidden state only knows about "great is".
forward/backward LM is pretty clumsy
can we replace these w/ a single model

ELMo ⇒ BERT
2018 2019

- 2 unidirectional LMs ⇒ one masked LM
- recurrent NNs ⇒ Transformers
- fine-tune the entire model instead of keeping the LM frozen
- pretrain on way more data w/ a way bigger model
Masked language modeling:

- predict "opened"
- softmax layer
- many Transformer blocks
- opened opened opened

Students [MASK] their books

- each representation at every time step is fully contextualized!
  - "aware" of both past + future words
- same training obj. as basic LM