NLP: BERT

Autoencoders and Transformers and LSTMs, oh my

Review/Recap

In our previous classes, we've talked about creating dictionaries and tokenizing text as an input into sentiment classifiers like VADER.

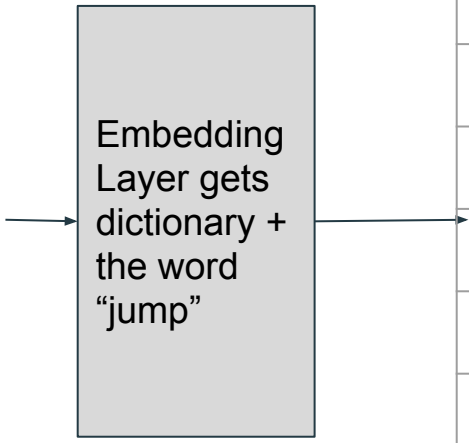
Today, we will delve into how those systems learn and take a quick look at a state of the art technology known as a Transformer.

Most of today will be geared toward understanding the intuition behind what we're doing. If you don't quite get the math as we're going, that's OK! Just understand the flow and you're good to go.

One hot encoding: Words into math

We've discussed tokenization and dictionary building in previous classes. To represent which word in a dictionary we're working with, a standard method is one hot encoding. Imagine a 1000 word dictionary:

Word	Token: word #
Apple	0
...	
Jump	500
...	
Zebra	999



List Position	Value
0	0
...	All zeros
500	1
...	More zeros
999	0

You can multiply together large lists of X and M

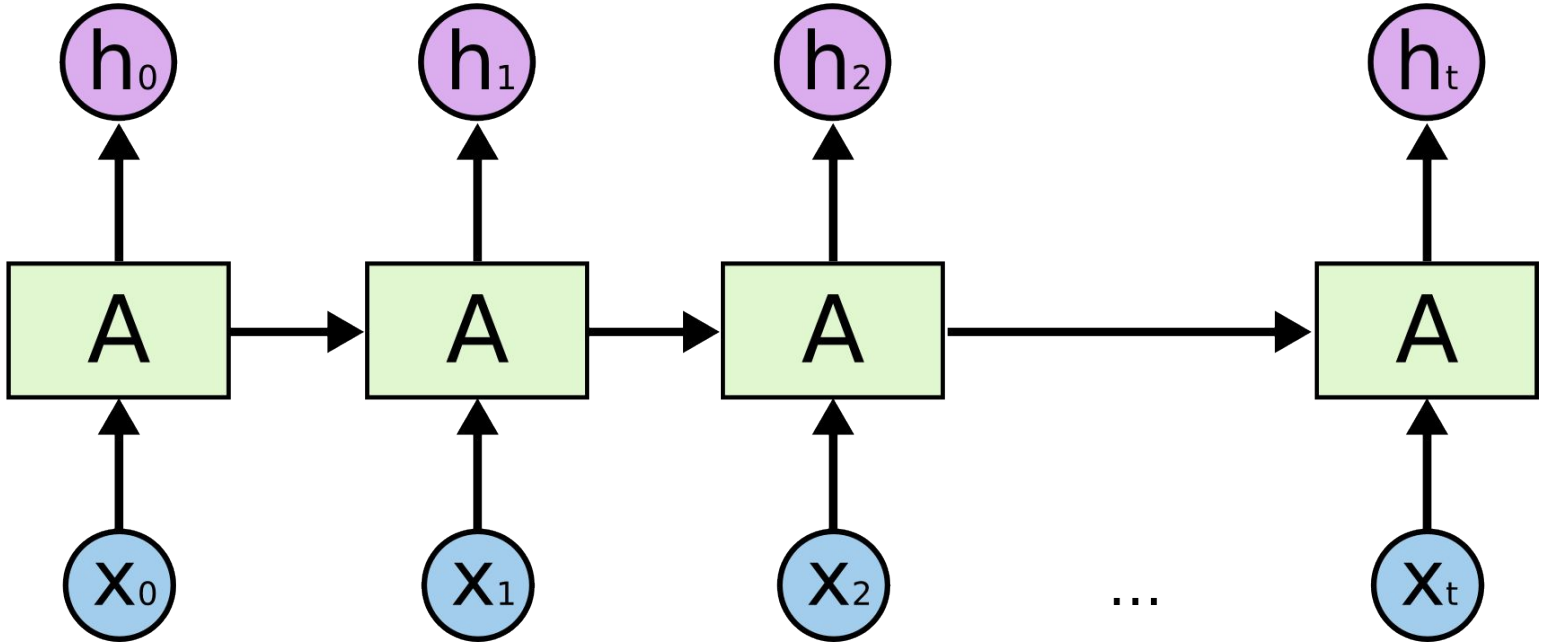
m0	0.25
m1	0.37
m2	0.01
...	
mN	?

*

x0	1
x1	0
x2	0
...	
xN	?

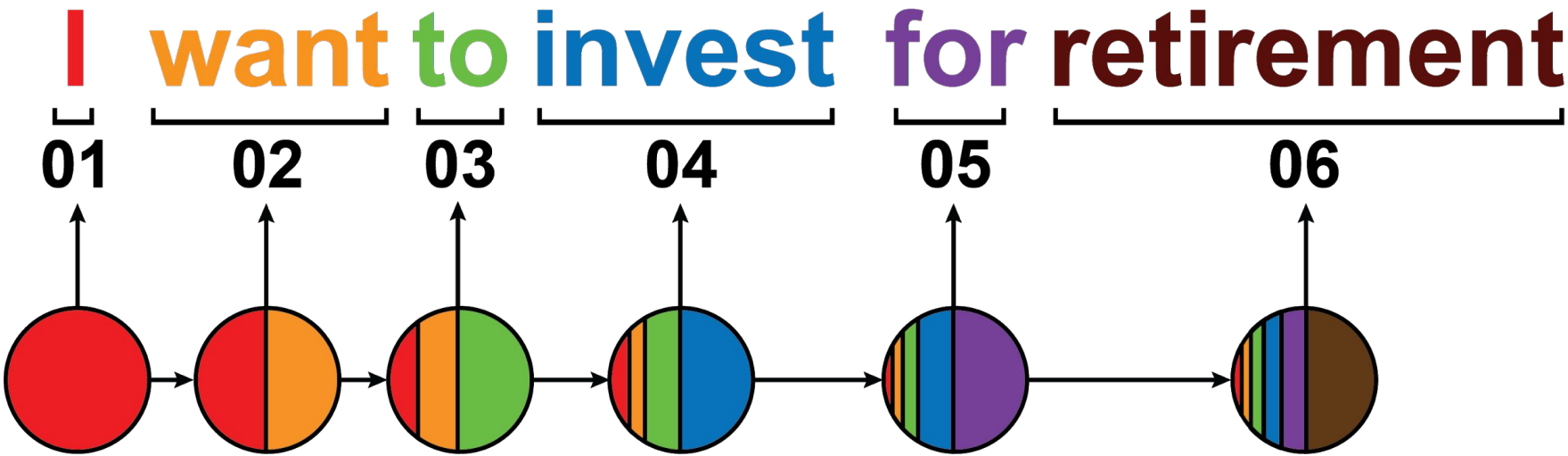
= (m0*x0) + (m1 * x1) + … (mN * xN)
= a single y answer

Unrolled LSTM



How Do RNNs Work?

The sentence is split into individual words. RNNs work sequentially, so we feed it one word at a time. By the final step, the RNN has encoded information from all the words in previous steps.



What does that look like in numbers?

Assume all coefficients (m in y=mx+b) are 0.5...

Original word	Converted to 1	Times coefficients	Result
to	1	0.5*0.5*0.5*0.5	0.0625
invest	1	0.5*0.5*0.5	0.125
for	1	0.5*0.5	0.25
Retirement	1	0.5	0.5

What happens if you ramp this up?

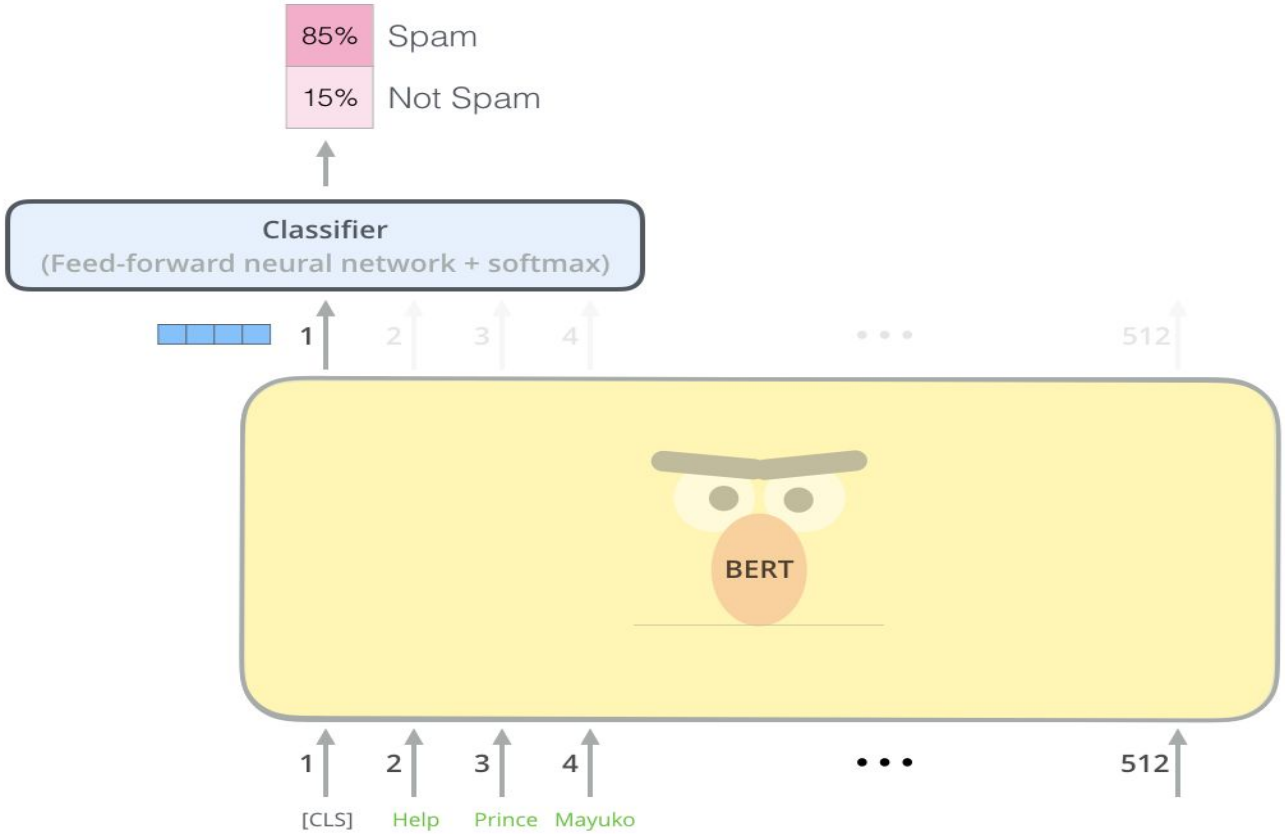
Way way way up… Like 300 million “m”s?

BERT

In 2018, Google released BERT, the **B**idirectional **E**ncoder **R**epresentation from **T**ransformers.

- Bert stacks a huge number of layers of these encoders on top of each other to produce as many as 768 outputs for each input sequence.
- These outputs can then be fed into a classifier model and produce very accurate predictions for a number of tasks, including sentiment analysis, spam detection, or predicting the next word/next sentence based on an input sentence.
- Bert includes its own tokenizing and embedding layers.
- Bert is a Pre-trained model, meaning you don't need to go through the time-consuming and resource-intensive process of training the model, but you can fine-tune it if you want.
- Trained on a huge corpus of books and wikipedia articles with a simple task of "Given an input sentence, what is the most likely next sentence?"

How you use Bert



Cite: http://jalammar.github.io/illustrated-bert/

Instructor Demo: Bert Tokenizer

Instructor review of a Jupyter Notebook that converts a text sentence into the proper format for the Bert Embedding layer.

Student activity

Prep data for BERT embedding layer

In this activity, students prep an input sentence for its trip through an embedding layer and onward to Bert.