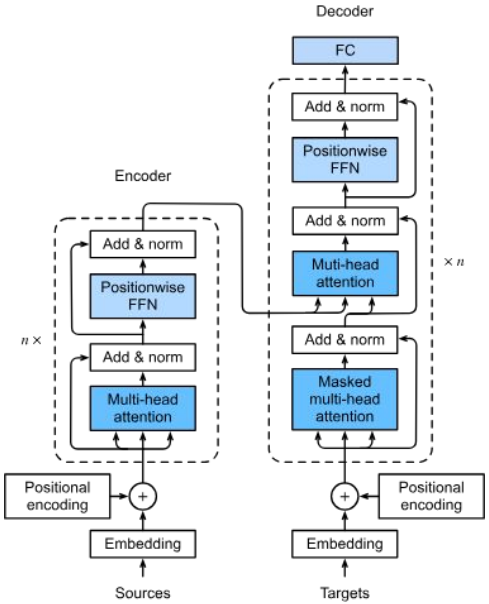
Language is simpler than you think:
Modeling Language with Hidden Markov Models

Larry & Josh

Our Goal

- Over the past few years, large language models have taken off (ChatGPT, Gemini, Grok, etc.)
- These are predictive models using probabilities, and are based off of the transformer architecture, which is REALLY complicated
- **Let's try to create a really simple probability based language model and see what it can do**



Transformer Structure

Image Credit: https://machinelearningmastery.com/the-transformer-model/

Idea #1

- We have a string of characters and we want to predict the next one:

```
The quick brown fox jum_
```

- Let’s just pretend that the next character is completely dependent on the previous one, so if we look at the probability that the next character is ‘p’:

P(next character is ‘p’ | string) ⇒ P(next character is ‘p’ | previous character)

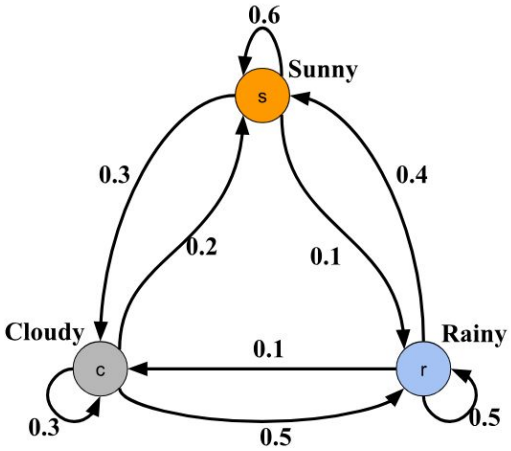
- Then, this system can be modeled with a **Markov Model**

Idea #1

- W
- Le
- Th

What's a Markov Model?

- A model of a random system where future states depend completely on the current state
- Consists of states S and transition probabilities P
- Our states are the characters in our string, the transitions are the next character, and we can infer P based on statistics of a dataset



Idea #1

- We have a string of characters and we want to predict the next one:

  `The quick brown fox jum_`

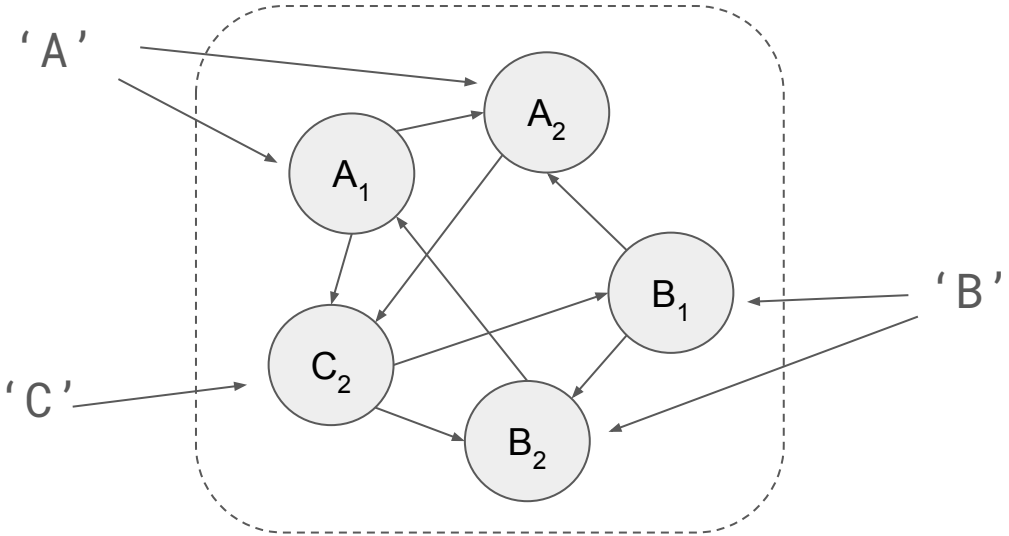
- Let's just pretend that the next character is completely dependent on the previous one:

  P(next character | string) = P(next character | previous character)

- Then, this system can be modeled with a **Markov Model**
- **Obvious problem: the next character depends on more than just the current one**

Idea #2

- Let’s keep the markov model but add more states and assume that characters don’t correspond one-to-one to each state
- Rather, let us assign many states to each character:



What we see:

‘T’	‘h’	‘e’	‘ ’	‘q’

Possible sequence of states:

T_1	h_6	e_3	$‘ ’_5$	q_5

Idea #2

- Let’s keep the markov model but add more states and assume that characters don’t correspond one-to-one to each state
- Rather, let us assign many states to each character.
- This is known as a **Hidden Markov Model**[1]
- Now, the current state depends on both the current character (as it has to be one of the states that map to the current character), as well as previous states (which determines which of these states is most likely)

[1] Technically a clone-structured HMM as the emission matrix is binary

Training Details

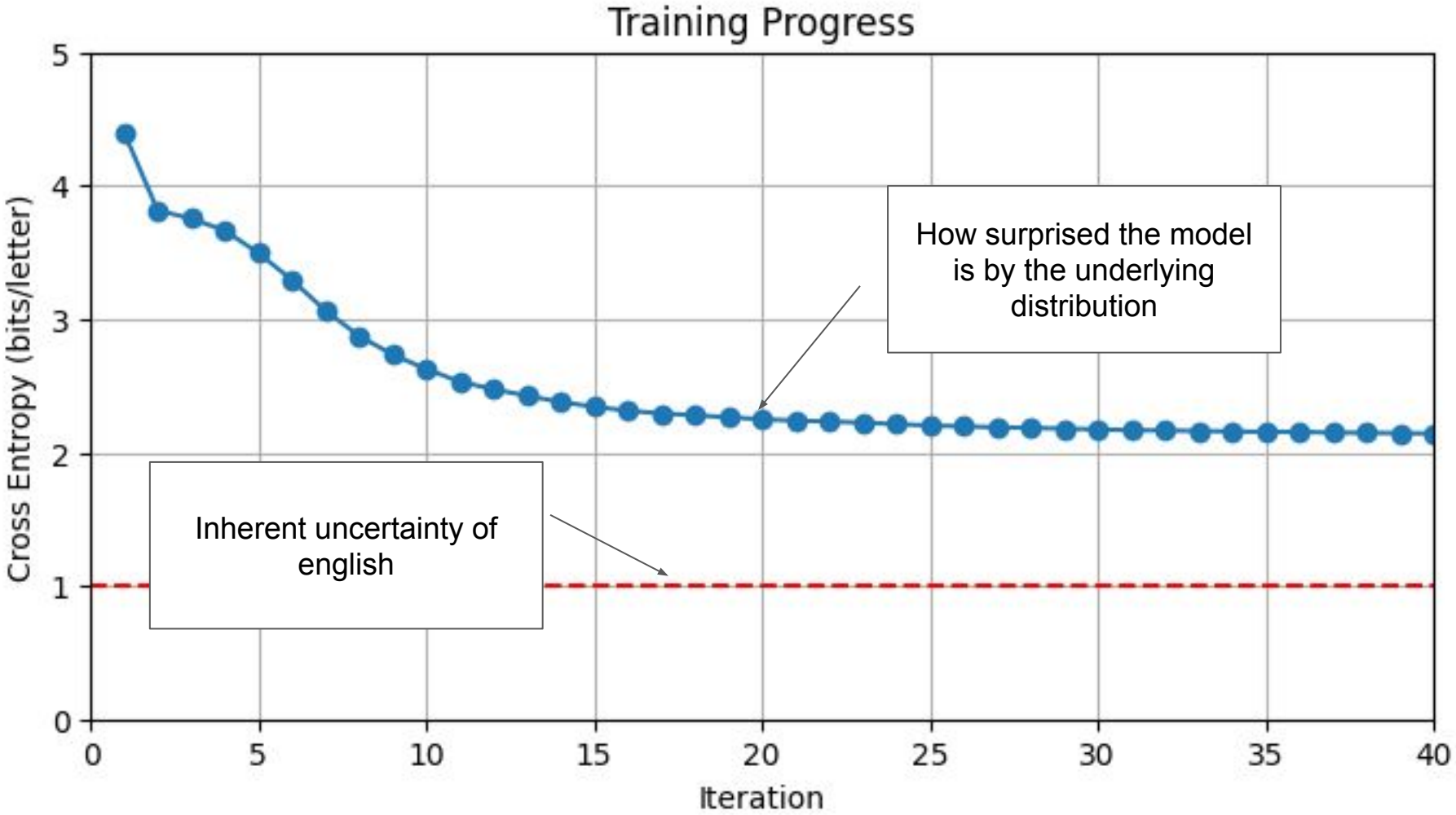
- We don’t know which of the hidden states are active so we can’t collect statistics directly
- Rather, we use an iterative process called Viterbi training:
  - First, guess the transition probabilities and using that, figure out which hidden states are most likely based on the dataset
  - Then, update the transition probabilities based on the statistics of those hidden states

Let’s make the model!

- We’ll use a simpler language dataset, `tinystories`, which is based on the words and sentences known by a 5 year old
- Total number of states: 3000 (allocated proportionally to character frequency)
- Dataset length: 30,000 characters
- Number of training iterations: 40
- Sampling strategy: topk, k=2

```
Spot saw the shiny car and
said, "Wow, Kitty, your car is
so bright and clean!" Kitty
smiled and replied, "Thank you,
Spot. I polish it every day."
After playing with the car,
Kitty and Spot felt thirsty.
They found a small pond with
clear water. They drank the
water and felt very happy. They
played together all day and
became...
```

Dataset sample


Training Progress
Cross Entropy (bits/letter)
Iteration
How surprised the model is by the underlying distribution
Inherent uncertainty of english
0
1
2
3
4
5
0
5
10
15
20
25
30
35
40

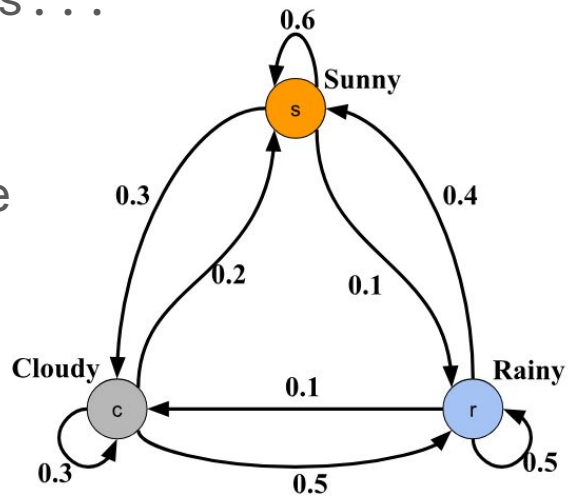
Training Results

Some sample results from running the model:

- `There felt said, "I watched Sue. Tom got and ...`
- `Lucy you withing to played him happy. ...`
- `After mom was a bird. Tom wasked and his...`

In conclusion, the model was able to learn:

- Spelling of common english words and word structure
  - Even inventing words like “`Withing`” and “`Wasked`"
- Punctuation and grammar (over short durations)
- **All from a simple markov model!**



Extended results

There was song his ball day, but he happy that the cared and saw him wed to played saw a tree the big train, Max. Mia was a big tree and Sue said, "I can park to park to played his frient ought a little book and his best and had a bird. Sue look the specause but him wed to the spider was a little big tree the truck the candle boy nail the can play withing to see he happy out it and said, "I haved to played his balwaying he happy to the sut they saw her mommy wend, Max. Tim saw a big tree and Sam said, "Tom truck the book and saw a big tree and said, "I wanted to sly was a little giratter fore to played Sam was a big lime, they like the candy said, "Tom was so happy. Sue. There was about under was a bird wasked and Sam store was a litty wasked to play to sly was a big leank you was a lat he happy to parknow to play and store felt very much. They liked and Sue story with the candy hill day, but the sut help the caked and saide and Sue liked and hung his because!"