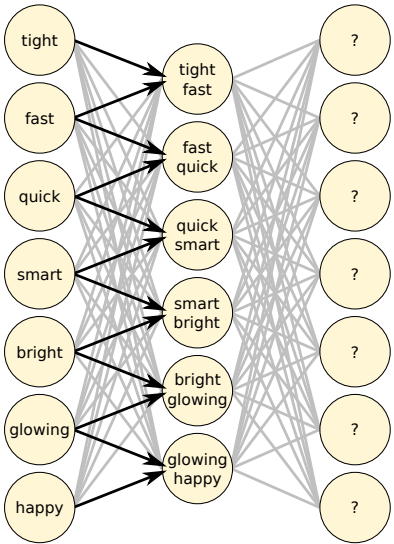
How is autocomplete like and how is it unlike ChatGPT?

Jim Pivarski

Princeton University – IRIS-HEP

July 9, 2024

1st difference: LLMs use neural networks



We made a database of exact words.

Using a neural network, LLMs encode sequences of meanings, not the exact words (or letters).

This makes it more sensitive to both synonyms and multi-word context.

From Andrej Karpathy's *Unreasonable Effectiveness of RNNs*

Cell sensitive to position in line:

```
The sole importance of the crossing of the Berezina lies in the fact
that it plainly and indubitably proved the fallacy of all the plans for
cutting off the enemy's retreat and the soundness of the only possible
line of action--the one Kutuzov and the general mass of the army
demanded--namely, simply to follow the enemy up. The French crowd fled
at a continually increasing speed and all its energy was directed to
reaching its goal. It fled like a wounded animal and it was impossible
to block its path. This was shown not so much by the arrangements it
made for crossing as by what took place at the bridges. When the bridges
broke down, unarmed soldiers, people from Moscow and women with children
who were with the French transport, all--carried on by vis inertiae--
pressed forward into boats and into the ice-covered water and did not,
surrender.
```

Cell that turns on inside quotes:

```
"You mean to imply that I have nothing to eat out of.... On the
contrary, I can supply you with everything even if you want to give
dinner parties," warmly replied Chichagov, who tried by every word he
spoke to prove his own rectitude and therefore imagined Kutuzov to be
animated by the same desire.

Kutuzov, shrugging his shoulders, replied with his subtle penetrating
smile: "I meant merely to say what I said."
```

Cell that robustly activates inside if statements:

```
static int __dequeue_signal(struct sigpending *pending, sigset_t *mask,
    siginfo_t *info)
{
 int sig = next_signal(pending, mask);
 if (sig) {
  if (current->notifier) {
   if (sigismember(current->notifier_mask, sig)) {
    if (!(current->notifier)(current->notifier_data)) {
     clear_thread_flag(TIF_SIGPENDING);
     return 0;
    }
   }
  }
  collect_signal(sig, pending, info);
 }
 return sig;
}
```

A large portion of cells are not easily interpretable. Here is a typical example:

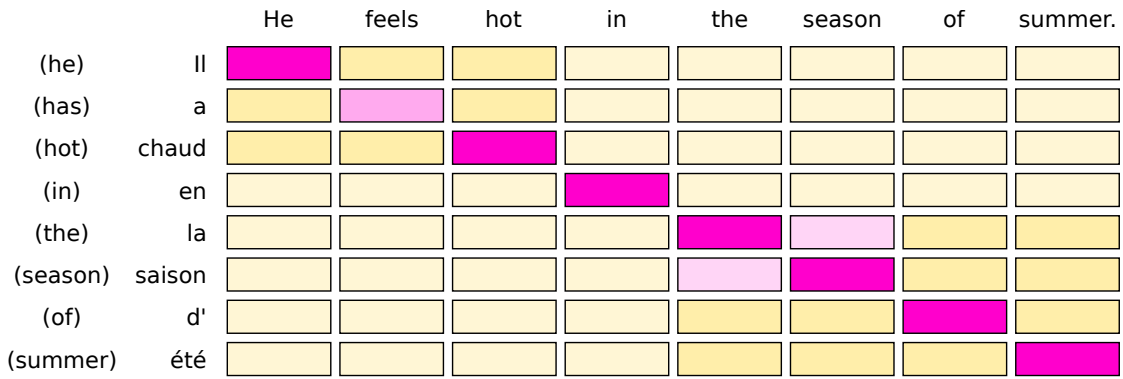
```
/* Unpack a filter field's string representation from user-space
 * buffer. */
char *audit_unpack_string(void **bufp, size_t *remain, size_t len)
{
 char *str;
 if (!*bufp || (len == 0) || (len > *remain))
  return ERR_PTR(-EINVAL);
 /* Of the currently implemented string fields, PATH_MAX
  * defines the longest valid length.
  */
```

Activation of individual cells in neural networks that generate text.

blue is $+1$, red is -1.

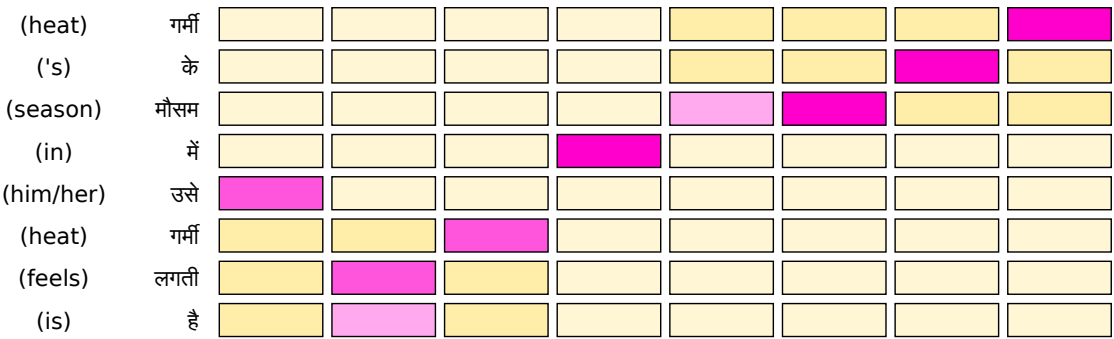
Some cells are gates that turn on and off submodels with different correlations, but most cells don't act on their own at all.

2nd difference: attention mechanism (from language translation)



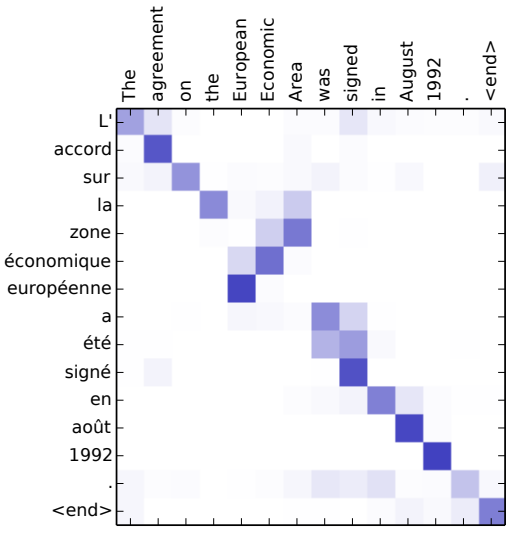
Our context window was exactly 5 tokens long. LLMs train an "attention" distribution that varies in size and shape for each token.

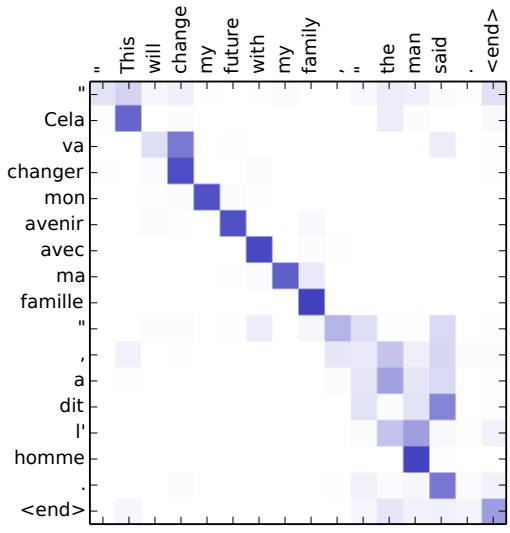
2nd difference: attention mechanism (from language translation)



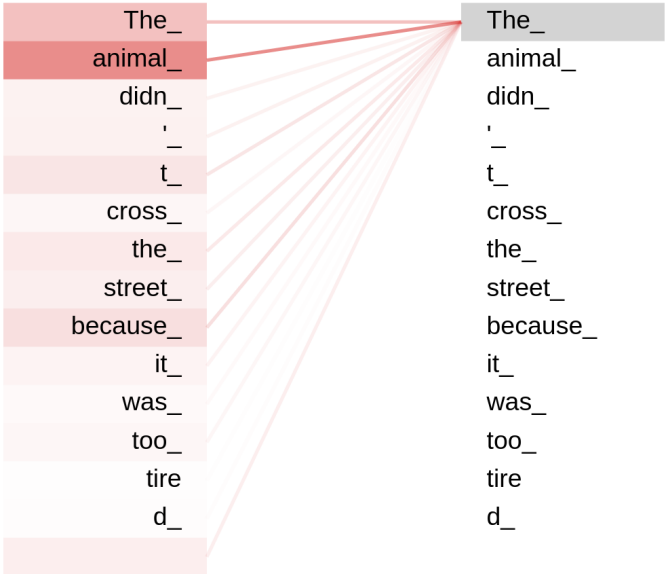
Our context window was exactly 5 tokens long. LLMs train an "attention" distribution that varies in size and shape for each token.

From D. Bahdanau, K. Cho, Y. Bengio (2014)





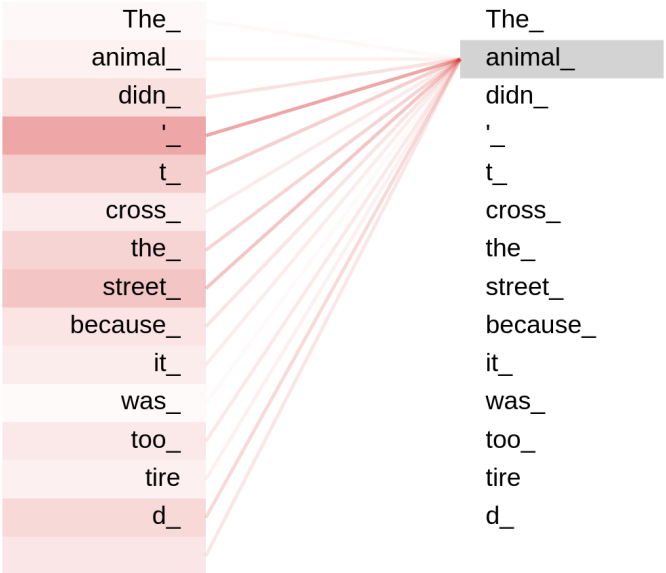
Self-attention (from “Tensor2Tensor Intro” on Google Colab)



The attention mechanism, when applied to a single stream of text (not language translation), provides long-term context, i.e. memory.

Notice that “it” pays the most attention to the two nouns, but more to “animal” than “street.”

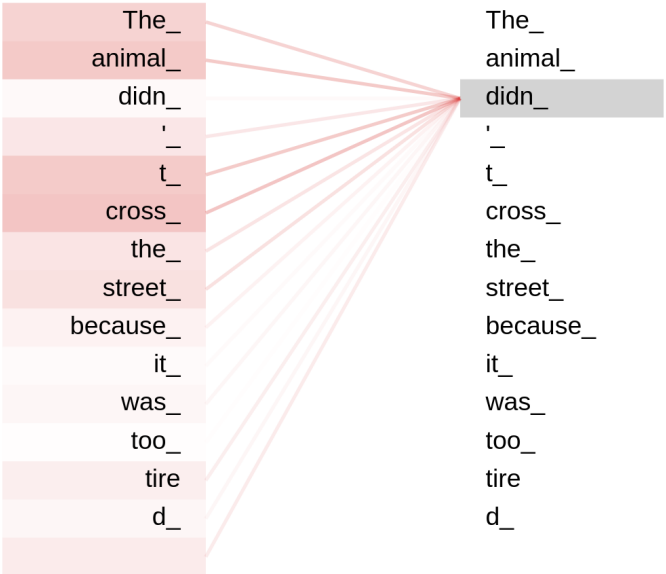
Self-attention (from “Tensor2Tensor Intro” on Google Colab)



The attention mechanism, when applied to a single stream of text (not language translation), provides long-term context, i.e. memory.

Notice that “it” pays the most attention to the two nouns, but more to “animal” than “street.”

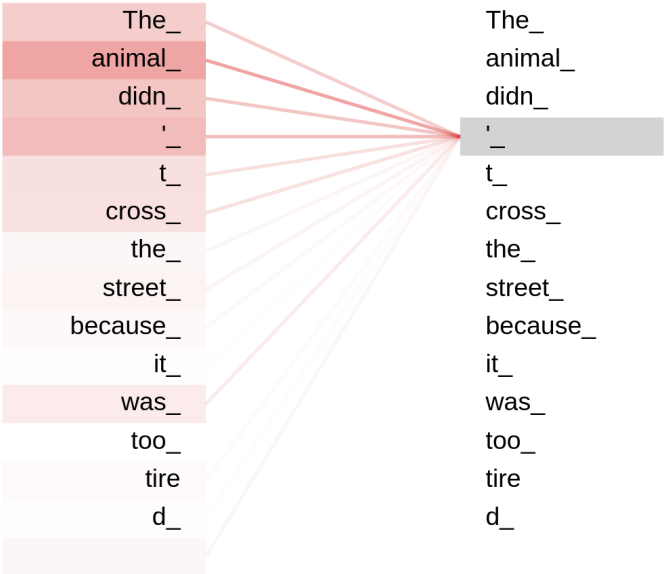
Self-attention (from "Tensor2Tensor Intro" on Google Colab)



The attention mechanism, when applied to a single stream of text (not language translation), provides long-term context, i.e. memory.

Notice that "it" pays the most attention to the two nouns, but more to "animal" than "street."

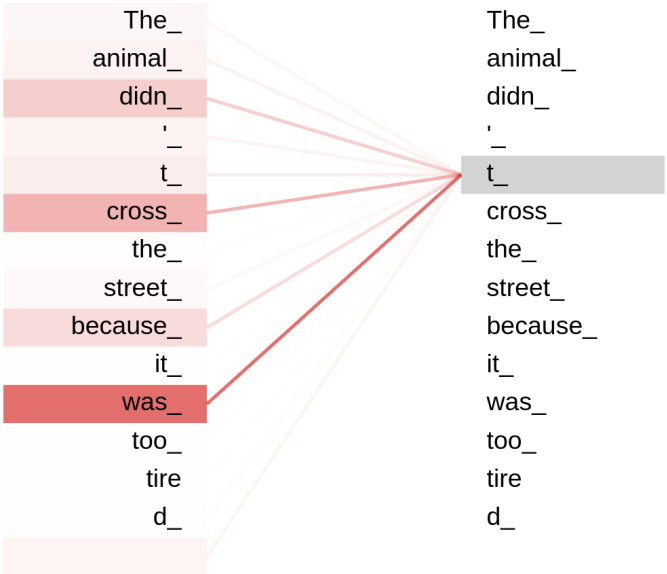
Self-attention (from “Tensor2Tensor Intro” on Google Colab)



The attention mechanism, when applied to a single stream of text (not language translation), provides long-term context, i.e. memory.

Notice that “it” pays the most attention to the two nouns, but more to “animal” than “street.”

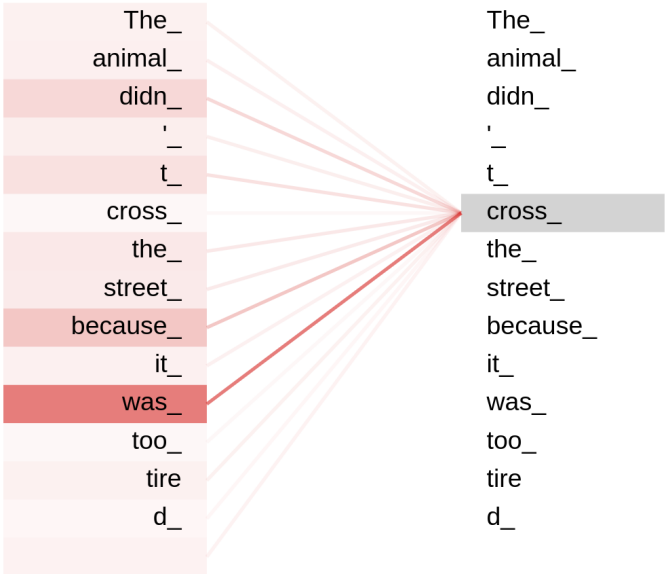
Self-attention (from “Tensor2Tensor Intro” on Google Colab)



The attention mechanism, when applied to a single stream of text (not language translation), provides long-term context, i.e. memory.

Notice that “it” pays the most attention to the two nouns, but more to “animal” than “street.”

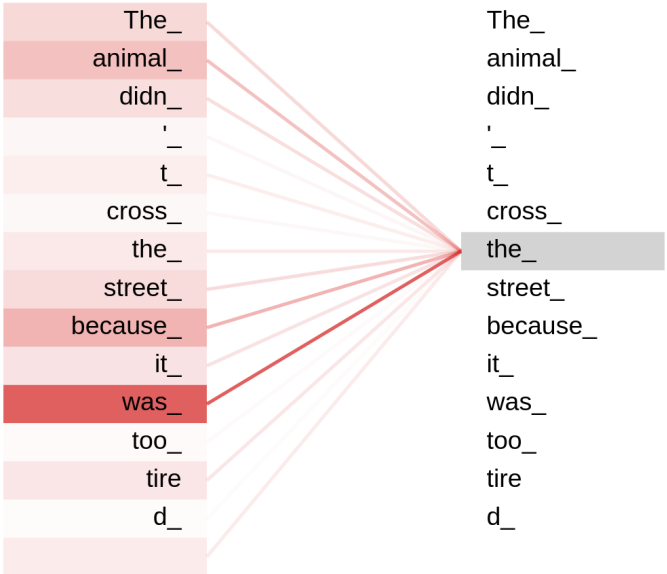
Self-attention (from “Tensor2Tensor Intro” on Google Colab)



The attention mechanism, when applied to a single stream of text (not language translation), provides long-term context, i.e. memory.

Notice that “it” pays the most attention to the two nouns, but more to “animal” than “street.”

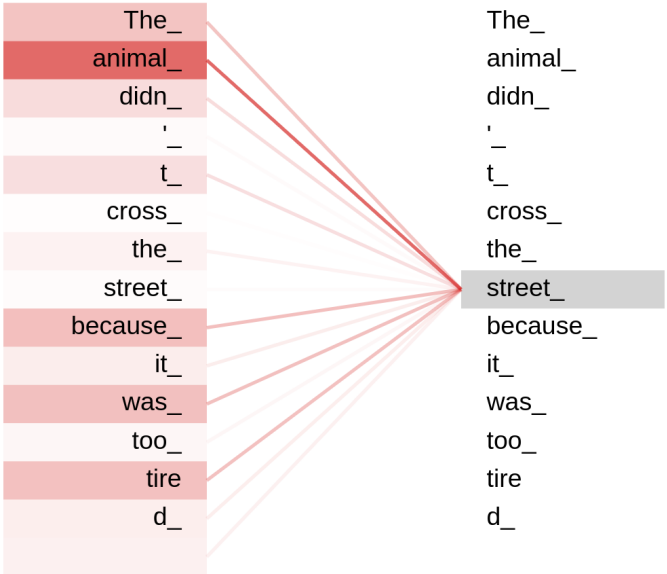
Self-attention (from "Tensor2Tensor Intro" on Google Colab)



The attention mechanism, when applied to a single stream of text (not language translation), provides long-term context, i.e. memory.

Notice that "it" pays the most attention to the two nouns, but more to "animal" than "street."

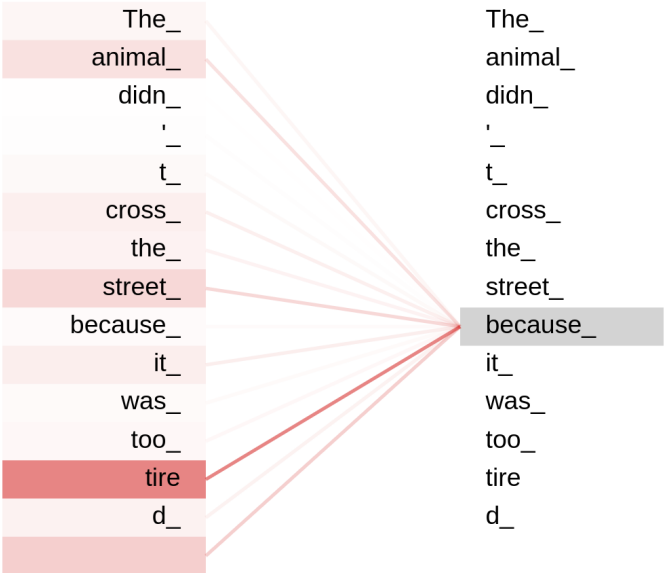
Self-attention (from "Tensor2Tensor Intro" on Google Colab)



The attention mechanism, when applied to a single stream of text (not language translation), provides long-term context, i.e. memory.

Notice that "it" pays the most attention to the two nouns, but more to "animal" than "street."

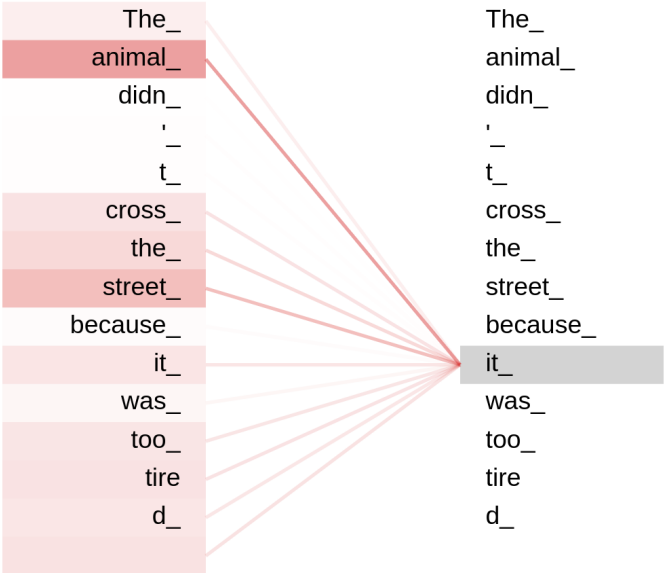
Self-attention (from “Tensor2Tensor Intro” on Google Colab)



The attention mechanism, when applied to a single stream of text (not language translation), provides long-term context, i.e. memory.

Notice that “it” pays the most attention to the two nouns, but more to “animal” than “street.”

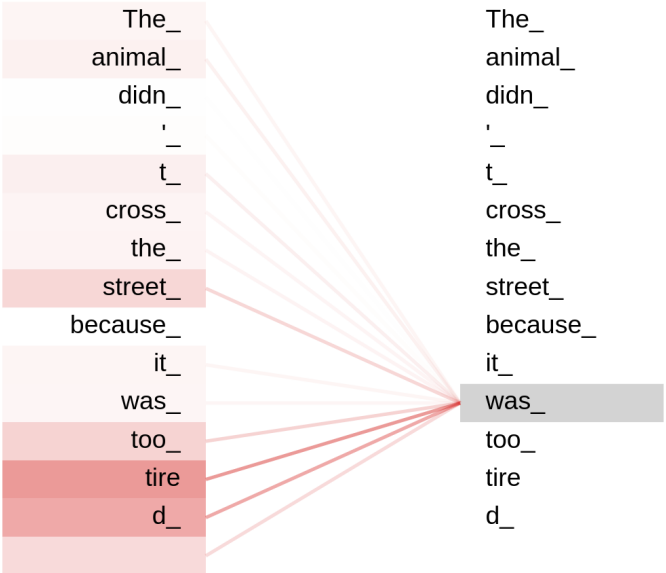
Self-attention (from "Tensor2Tensor Intro" on Google Colab)



The attention mechanism, when applied to a single stream of text (not language translation), provides long-term context, i.e. memory.

Notice that "it" pays the most attention to the two nouns, but more to "animal" than "street."

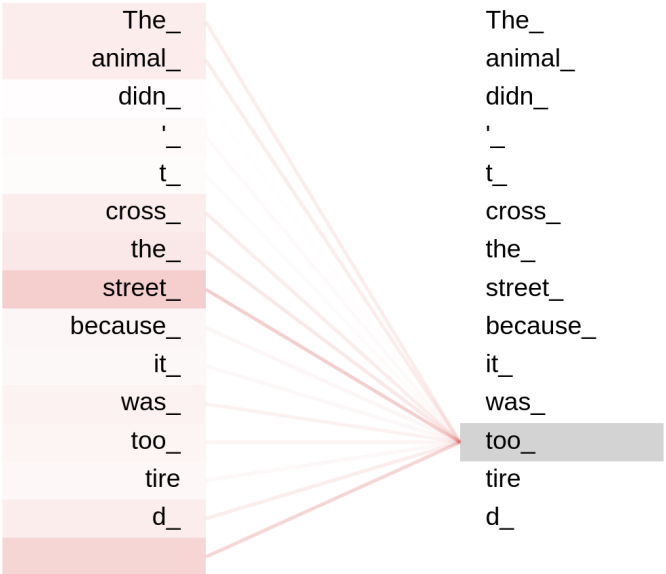
Self-attention (from “Tensor2Tensor Intro” on Google Colab)



The attention mechanism, when applied to a single stream of text (not language translation), provides long-term context, i.e. memory.

Notice that “it” pays the most attention to the two nouns, but more to “animal” than “street.”

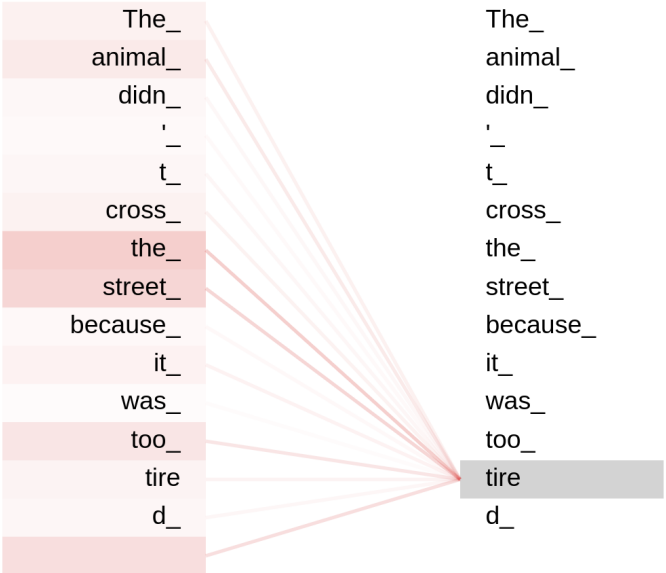
Self-attention (from “Tensor2Tensor Intro” on Google Colab)



The attention mechanism, when applied to a single stream of text (not language translation), provides long-term context, i.e. memory.

Notice that “it” pays the most attention to the two nouns, but more to “animal” than “street.”

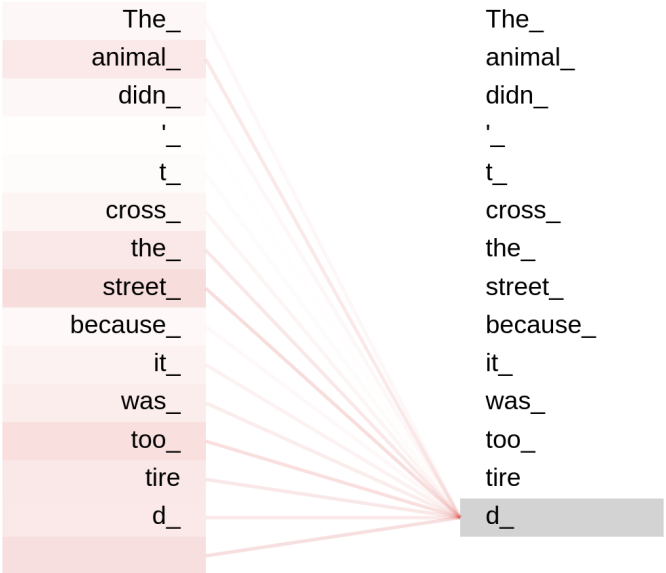
Self-attention (from "Tensor2Tensor Intro" on Google Colab)



The attention mechanism, when applied to a single stream of text (not language translation), provides long-term context, i.e. memory.

Notice that "it" pays the most attention to the two nouns, but more to "animal" than "street."

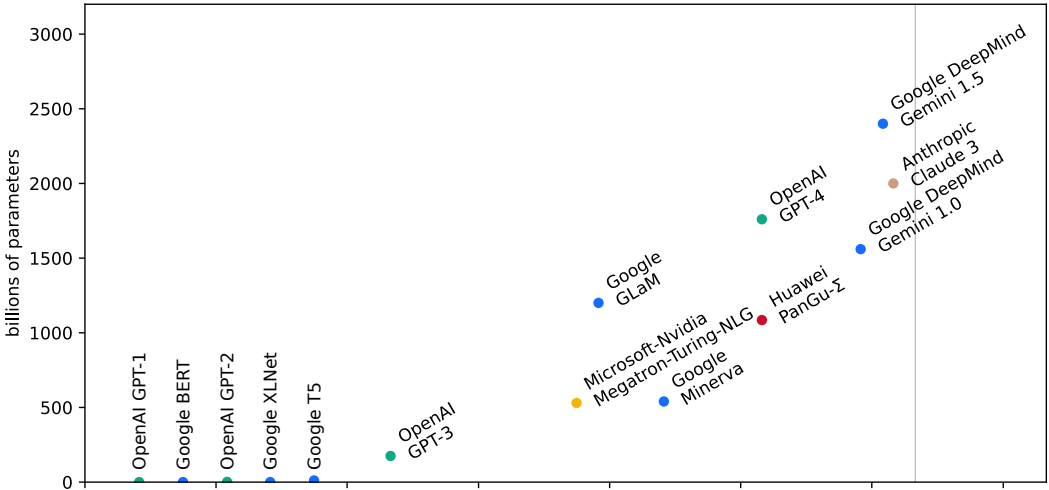
Self-attention (from “Tensor2Tensor Intro” on Google Colab)



The attention mechanism, when applied to a single stream of text (not language translation), provides long-term context, i.e. memory.

Notice that “it” pays the most attention to the two nouns, but more to “animal” than “street.”

3rd difference: LLMs are huge



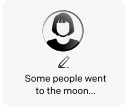
4th difference: chat LLMs are fine-tuned for usefulness

Step 1

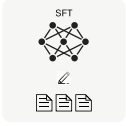
Collect demonstration data, and train a supervised policy.

A prompt is sampled from our prompt dataset.

A labeler demonstrates the desired output behavior.



This data is used to fine-tune GPT-3 with supervised learning.



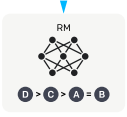
Step 2

Collect comparison data, and train a reward model.

A prompt and several model outputs are sampled.

A labeler ranks the outputs from best to worst.

This data is used to train our reward model.



Step 3

Optimize a policy against the reward model using reinforcement learning.

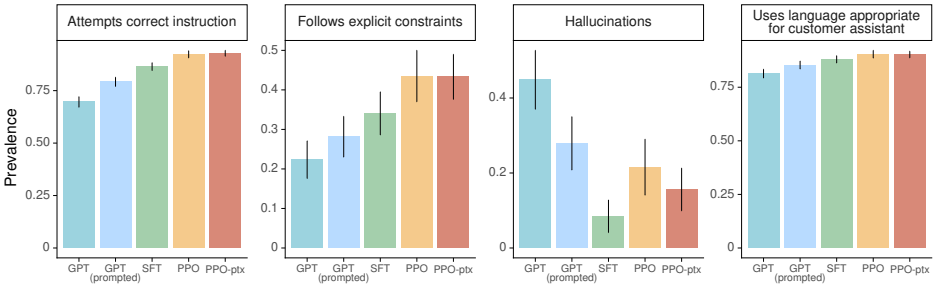
A new prompt is sampled from the dataset.

The policy generates an output.

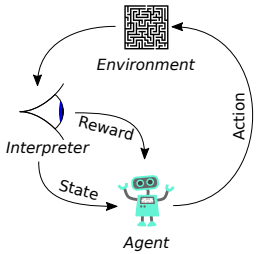
The reward model calculates a reward for the output.

The reward is used to update the policy using PPO.

From L. Ouyang et al. /OpenAI (2022)



After the big model (GPT-3) was trained on raw text completions, the neural network weights were re-fitted to optimize for helpfulness, as defined by 40 paid users (InstructGPT, and similarly for ChatGPT).



How is autocomplete like and how is it unlike ChatGPT?

Similarities:

- Like our Shakespearean autocomplete engine, LLMs are **models** of word sequences that have been **fitted** to **measurements** of text written by humans (Common Crawl, WebText, Wikipedia, public domain books...).

How is autocomplete like and how is it unlike ChatGPT?

Similarities:

- Like our Shakespearean autocomplete engine, LLMs are **models** of word sequences that have been **fitted** to **measurements** of text written by humans (Common Crawl, WebText, Wikipedia, public domain books...).
- They generate new text with word **correlations** similar to what they've seen.

How is autocomplete like and how is it unlike ChatGPT?

Similarities:

- Like our Shakespearean autocomplete engine, LLMs are **models** of word sequences that have been **fitted** to **measurements** of text written by humans (Common Crawl, WebText, Wikipedia, public domain books. . .).
- They generate new text with word **correlations** similar to what they've seen.

Differences:

- But LLMs are deep neural networks, not nearest-matches of n-grams.

How is autocomplete like and how is it unlike ChatGPT?

Similarities:

- Like our Shakespearean autocomplete engine, LLMs are **models** of word sequences that have been **fitted** to **measurements** of text written by humans (Common Crawl, WebText, Wikipedia, public domain books. . .).
- They generate new text with word **correlations** similar to what they've seen.

Differences:

- But LLMs are deep neural networks, not nearest-matches of n-grams.
- They use the "attention" mechanism to correlate relevant word pairs over large distances, to stay on topic.

How is autocomplete like and how is it unlike ChatGPT?

Similarities:

- Like our Shakespearean autocomplete engine, LLMs are **models** of word sequences that have been **fitted** to **measurements** of text written by humans (Common Crawl, WebText, Wikipedia, public domain books...).
- They generate new text with word **correlations** similar to what they've seen.

Differences:

- But LLMs are deep neural networks, not nearest-matches of n-grams.
- They use the "attention" mechanism to correlate relevant word pairs over large distances, to stay on topic.
- The training datasets and the numbers of parameters are huge.

How is autocomplete like and how is it unlike ChatGPT?

Similarities:

- Like our Shakespearean autocomplete engine, LLMs are **models** of word sequences that have been **fitted** to **measurements** of text written by humans (Common Crawl, WebText, Wikipedia, public domain books. . .).
- They generate new text with word **correlations** similar to what they've seen.

Differences:

- But LLMs are deep neural networks, not nearest-matches of n-grams.
- They use the "attention" mechanism to correlate relevant word pairs over large distances, to stay on topic.
- The training datasets and the numbers of parameters are huge.
- Chat-bots have been fine-tuned by human trainers for usefulness.

But those differences in scale and methodology don't change the fact that LLMs are fundamentally **word correlation models**.

But those differences in scale and methodology don't change the fact that LLMs are fundamentally **word correlation models**.

We know (roughly) what went into them.

But those differences in scale and methodology don't change the fact that LLMs are fundamentally **word correlation models**.

We know (roughly) what went into them.

Therefore, when someone asks, "can ChatGPT think like us?" they're really asking whether we're *not* word correlation models.

Bibliography

Blogs

- Andrej Karpathy; Unreasonable effectiveness of recurrent neural networks: https://karpathy.github.io/2015/05/21/rnn-effectiveness/
- Andrej Karpathy's web demos: https://cs.stanford.edu/people/karpathy/convnetjs/
- Jay Alammar; Visualizing machine learning: https://jalammar.github.io/
- Huiming Song; Detailed comparison of OpenAI GPTs: https://songhuiming.github.io/pages/2023/05/28/gpt-1-gpt-2-gpt-3-instructgpt-chatgpt-and-gpt-4-summary/
- Gregory Roberts; LLM training data sources: https://gregoreite.com/drilling-down-details-on-the-ai-training-datasets/

Academic papers

- Attention mechanism: https://arxiv.org/abs/1409.0473 (2014)
- Transformer model: https://arxiv.org/abs/1706.03762 (2017)
- Human fine-tuning: https://arxiv.org/abs/2203.02155 (2022)