# Lecture 24 — Large Language Models

Jeff Zarnett
jzarnett@uwaterloo.ca

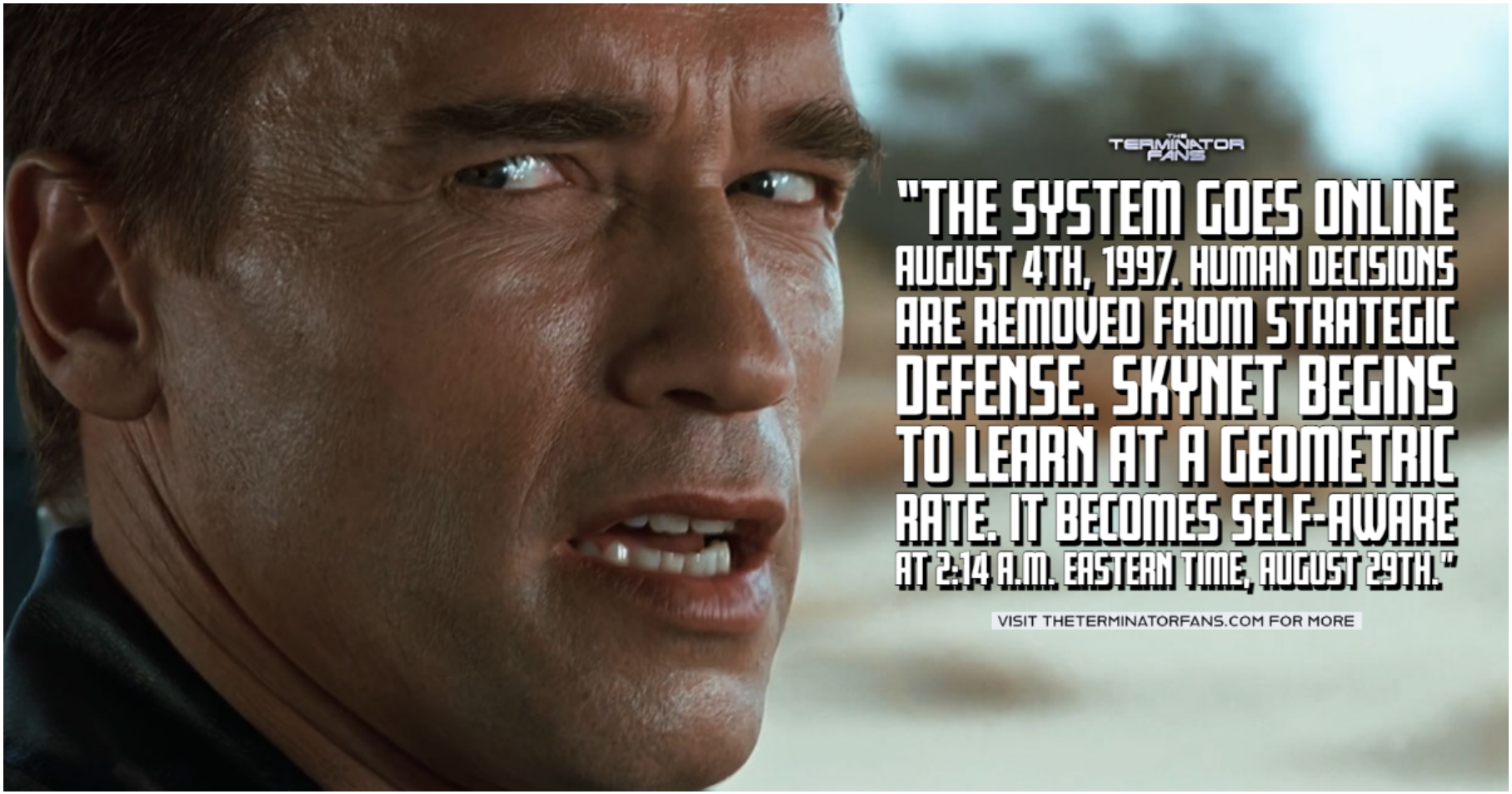
Department of Electrical and Computer Engineering
University of Waterloo

September 18, 2023

# In the Beginning

In November of 2022, OpenAI introduced ChatGPT to the world...

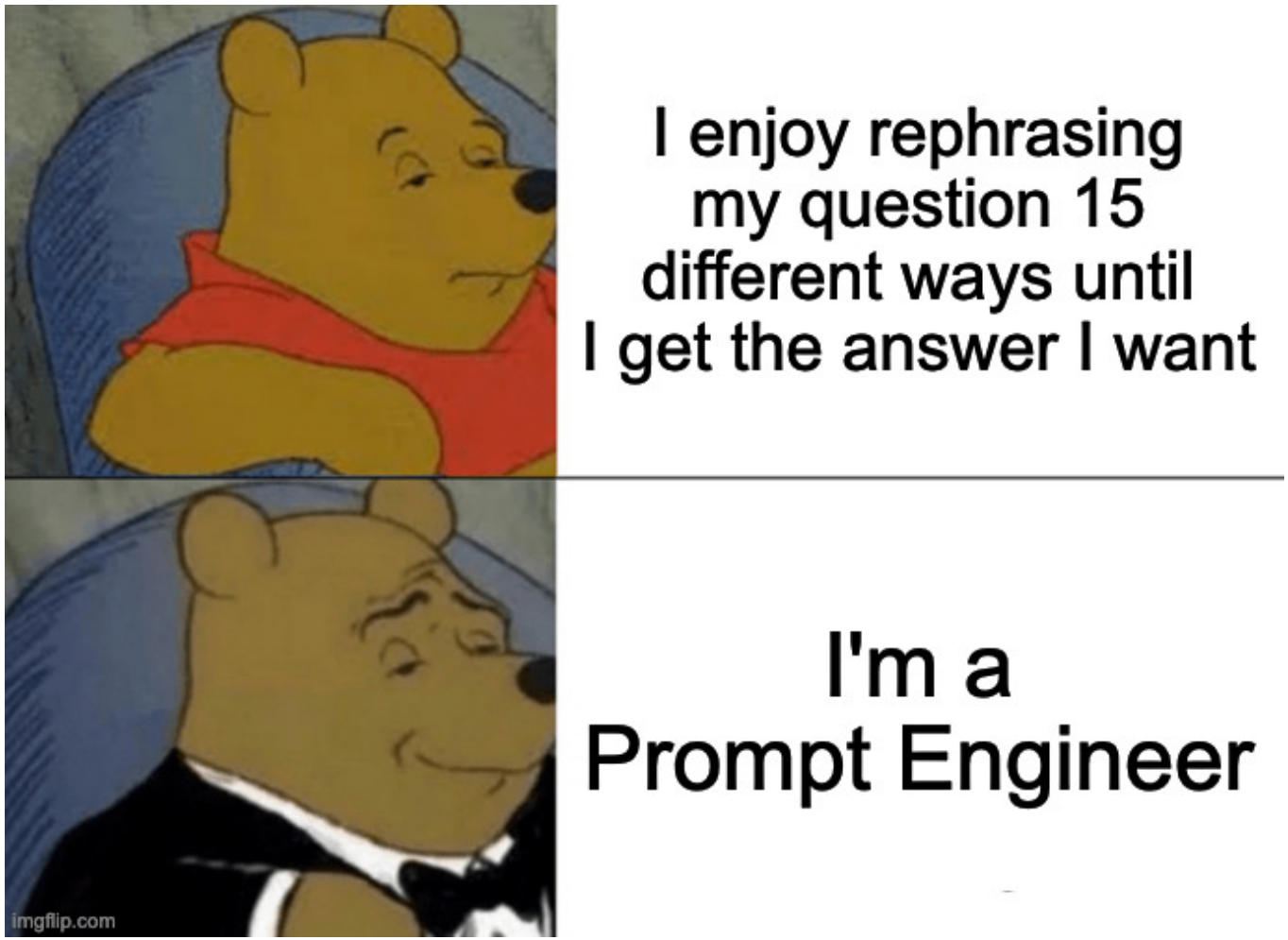
# Reaction 1


THE TERMINATOR FANS
"THE SYSTEM GOES ONLINE AUGUST 4TH, 1997. HUMAN DECISIONS ARE REMOVED FROM STRATEGIC DEFENSE. SKYNET BEGINS TO LEARN AT A GEOMETRIC RATE. IT BECOMES SELF-AWARE AT 2:14 A.M. EASTERN TIME, AUGUST 29TH."
VISIT THETERMINATORFANS.COM FOR MORE

# Reaction 2


STARTUP TECH SUPPOSED TO DISRUPT THEIR INDUSTRY OF CHOICE
CHATGPT
imgflip.com

# Reaction 3


I enjoy rephrasing my question 15 different ways until I get the answer I want
I'm a Prompt Engineer
imgflip.com


The "buy my course" starter pack
*Is banned from PayPal*
*Doesn't want you to know that that his money really comes from selling his course*
"I'm a mentor"
$2,000 Per Month SMMA
fulfillment by amazon
AliExpress
shopify
WOO COMMERCE
facebook business
INSTAGRAM FOR BUSINESS
AT 16
"Not saturated"
"Are you tired of working a 9-5?"
"college is a waste of money"
$1,000 Per DAY PASSIVE INCOME
*not a rental
$5,000 ON SHOPIFY
WITH NO EXPERIENCE
"you gotta find a Niche"
*Usally in his late teens*
"Five easy steps"
*Sets up an "us vs them" dynamic*

# ChatGPT

Such large language models have existed before, but ChatGPT ended up a hit because it's pretty good at being "conversational".

This is referred to as Natural Language Processing (NLP).

Just because it gives you an answer, doesn’t mean the answer is correct or true.


AI is subject to hallucinations.

# Remember, The Law...

Legally and professionally, the engineer is responsible for understanding how a given tool works & verifying the output is reasonable & correct.

# Prepared in Advance?

Part of what makes the GPT-3 and GPT-4 models better at producing output that matches our expectations is that it relies on pre-training.

This course is not one on neural networks, large language models, AI, or similar.

But performance is relevant here!

# Parameters

One factor that matters in how good a model is: parameters.

Bigger is *usually* better...
    But requires more computational and memory resources.

We can maybe tune some options!

# Let’s Try Some Optimization

This section is based on a guide from “Hugging Face”.


You may have guessed by the placement of this topic in the course material that the GPU is the right choice for how to generate or train a large language model.

# About that Name

Just... uh... don't confuse Hugging Face with Facehugger...

# Why a GPU?

In this case we're talking about Transformers.

There are three main groups of optimizations that it does:

- Tensor Contractions
- Statistical Normalizations
- Element-Wise Operators

We also need to consider what's in memory.

# Optimize What?

We can focus on how to generate a model that gives answers quickly...

Or we can focus on how to generate or train the model quickly.

# Make a Fast Model

Use more space to reduce CPU usage, optimize for common cases, speculate...

Some of these are more fun than others: given a particular question, can you guess what the followup might be?

# Make a Model Fast

Why would we customize some LLM?

Don’t send your data to OpenAI...

Specialize for your workload.

# Batch Size

Our first major optimization, and perhaps the easiest to do, is the batch size.

# Let’s Review the Code

The code is a little too large for the slides so let’s go over it in a code editor.

The bert-large-uncased model is about 340 MB.

It’s uncased because it makes no distinction between capitals and lower-case letters, e.g., it sees “Word” and “word” as equivalent.

# Let’s Try Training

```
jzarnett@ecetesla0:~/github/ece459/lectures/live-coding/L24$ python3 dummy_data.py
Starting up. Initial GPU utilization:
GPU memory occupied: 0 MB.
Initialized Torch; current GPU utilization:
GPU memory occupied: 417 MB.
Some weights of BertForSequenceClassification were not initialized
from the model checkpoint at  bert-large-uncased and are newly initialized:
['classifier.bias', 'classifier.weight']
You should probably TRAIN this model on a down-stream task to be able
to use it for predictions and inference.
GPU memory occupied: 1705 MB.
torch.cuda.OutOfMemoryError: CUDA out of memory. Tried to allocate 20.00 MiB (GPU 0;
7.43 GiB total capacity; 6.90 GiB already allocated; 16.81 MiB free; 6.90 GiB
reserved in total by PyTorch) If reserved memory is >> allocated memory try setting
max_split_size_mb to avoid fragmentation.  See documentation for Memory Management
and PYTORCH_CUDA_ALLOC_CONF
```

# Let’s See the Problem

I asked `nvidia-smi`…

```
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 470.199.02   Driver Version: 470.199.02   CUDA Version: 11.4     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|                               |                      |               MIG M. |
|===============================+======================+======================|
|   0  Tesla P4            Off  | 00000000:17:00.0 Off |                    0 |
| N/A   42C    P0    23W /  75W |      0MiB /  7611MiB |      1%      Default |
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                                  |
|  GPU   GI   CI        PID   Type   Process name                  GPU Memory |
|        ID   ID                                                   Usage      |
|=============================================================================|
|  No running processes found                                                 |
+-----------------------------------------------------------------------------+
```

# Kirk to Engineering...

7 611 MB is not enough for this model...


Problem is we don’t have any bigger VRAM cards.

# Fine, I Guess

What I actually did next was change to a smaller version of the model, `bert-base-uncased`.

It's significantly smaller (110 MB) and something the card could handle.