

Practical AI and LLM for Devs

Goals

1. Get hands-on with major concepts, techniques and tools in the LLM world: RAG, Agents, Fine-tuning, hosting, Hugging Face, Pinecone, Streamlit, etc
2. Build meaningful intuition for strengths/weaknesses of different solutions across diverse, real-world use cases
3. Learn in an accelerated/intense way with a community of serious (and fun) peers

Agenda

- Designing LLM Solutions and Syllabus
- How LLMs work
- Overview of Neural Networks
- Group Project: Mixer
- Individual Project: Week 1

Designing LLM Solutions and Syllabus

Designing LLM solutions

Step 1: Start w/ ChatGPT 4 + prompts w/ one-shot or multi-shot

Step 2: Use RAG to load in more specific context

Step 3: Decompose prompt into agents for specialization OR access to APIs

Step 4: Build a dataset for fine-tuning

Step 5: Fine-tune models

Syllabus

Week 1: Exploring neural networks

Week 2: Embeddings & RAG

Week 3: Agents and Tools

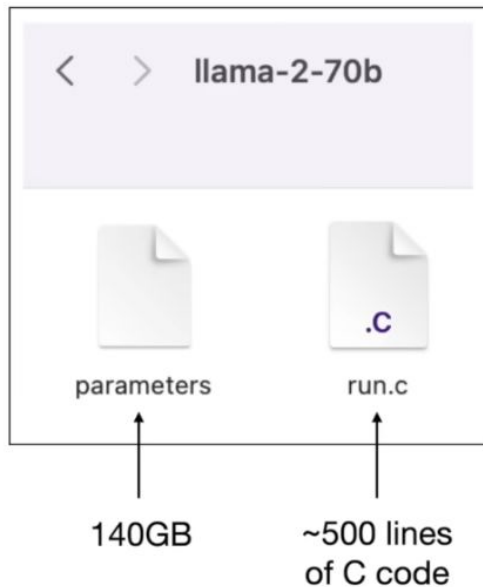
Week 4: Building training sets and evaluation

Week 5: Fine-tuning models

Week 6: TBD

How LLMs Work

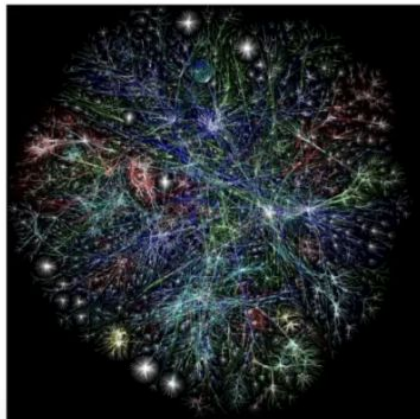
What are LLMs made of?



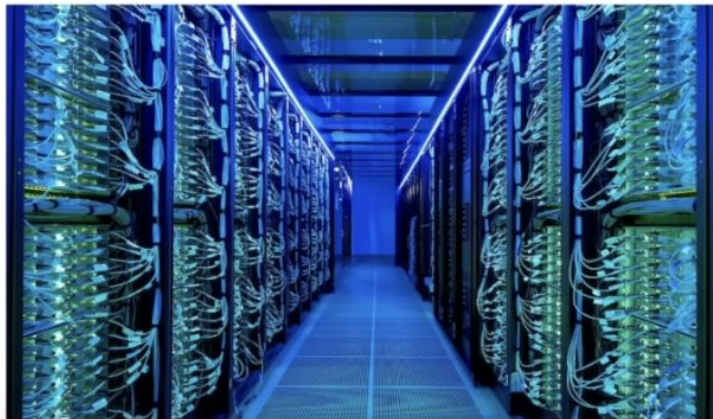
- Parameters are 2 byte numbers
- C implementation adds/multiplies the numbers

Training them is more involved.

Think of it like compressing the internet.



Chunk of the internet,
~10TB of text



6,000 GPUs for 12 days, ~\$2M
~1e24 FLOPS



parameters.zip

~140GB file

*numbers for Llama 2 70B

After training...

- This is called the “base model”
- The knowledge is in the network, but the output is not really coherent
- Re-hydrates the embedded knowledge in a familiar form, but hallucinates details

After finetuning you have an Assistant:

<USER>

Can you help me with this code? It seems like there is a bug.
`print("hello world)`

<ASSISTANT>

It looks like you forgot to close the string passed to the function `print`. You have to add a closing quote to properly terminate the string. Here is the corrected function:

```
print("hello world")
```

Let me know if I can help with anything else!


- Training data is in Question / Answer format
- Quality of training data matters!
- Scale is ~100k conversations (originally authored by people)
- With modern models, we can bootstrap more training data w/ human/AI collaboration

How does it work?

- We know exactly how the math of the transformer architecture works
- We know how to adjust the parameters during training to reduce error
- We don't know how the parameters collaborate to work as well as it does

Overview of Neural Networks


Kaggle Machine Learning competitions

 KAGGLE · GETTING STARTED PREDICTION COMPETITION · ONGOING

Submit Prediction ...

Titanic - Machine Learning from Disaster

Start here! Predict survival on the Titanic and get familiar with ML basics



[Overview](#) [Data](#) [Code](#) [Models](#) [Discussion](#) [Leaderboard](#) [Rules](#) [Team](#) [Submissions](#)

Dataset Description

Overview

The data has been split into two groups:

- training set (train.csv)
- test set (test.csv)

Files

3 files

Size

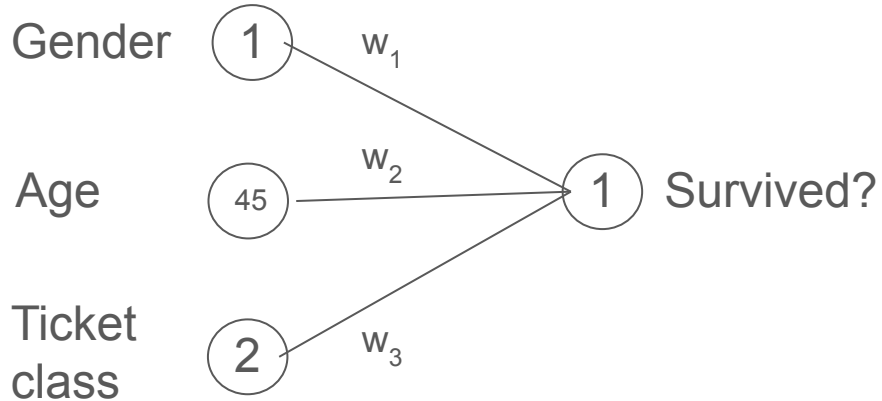
93.08 kB

Type

CSV

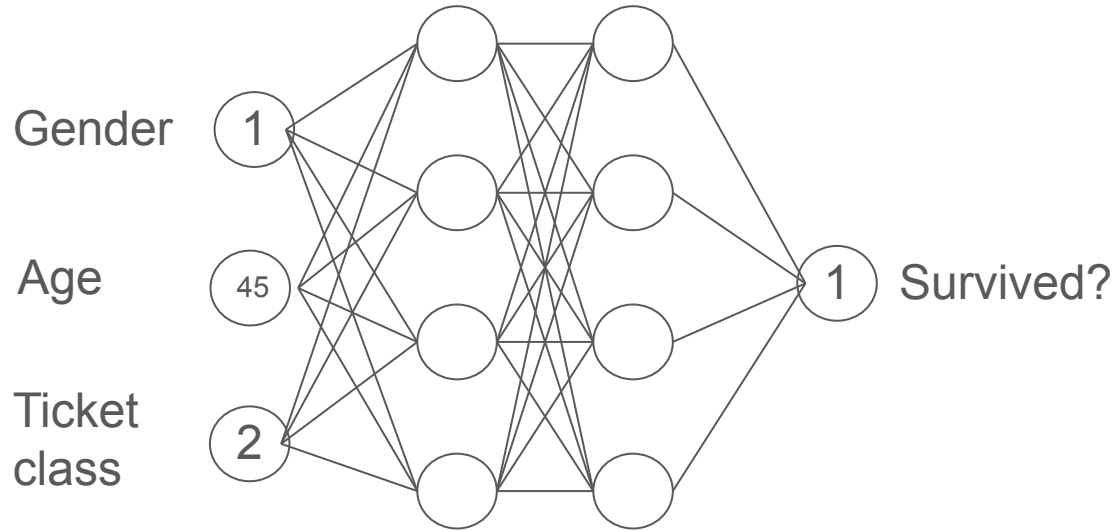
License

Titanic - Predicting survival with the Perceptron



- Invented in 1950s
- Worked okay, but not that well
- Classification or regression
- Weights were trained by iteratively reducing error
- Neurons had an activation function (ReLU - rectified linear unit)

Deep Learning - Adding hidden layers



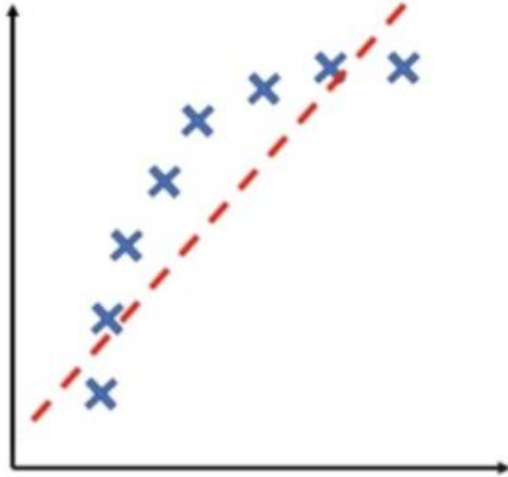
- Adding hidden layers inspired by synapses in the brain
- Weights trained via back propagation (stochastic gradient descent)
- Worked surprisingly well!
Not very explainable

Training process

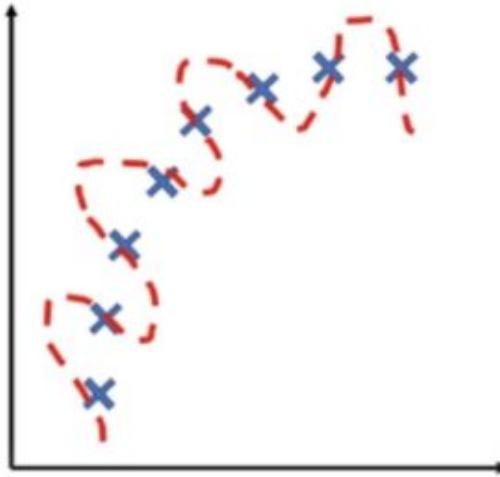
- Each run through of the training set is called an epoch
- For each training item, calculate the loss function, and decrease the loss by tweaking each weight by the learning rate
- Train using a training set, final check against a validation set
- Too many epochs leads to overfitting

Overfitting

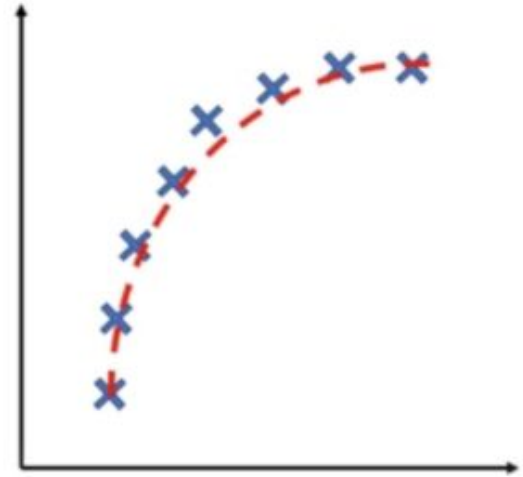
Underfitting



Overfitting



Ideal Balance



Network architectures

- Feedforward architecture
- RNN - great for time series
- CNN - great for images
- Transformers - led to ChatGPT

A few final neural network concepts...

- Training, validation, and test sets
- Transfer learning
- Quantization (e.g., 4 bit instead of 16 bit)
- Parameter Efficient Fine Tuning (PEFT)
 - LoRA (QLoRA)

Group Project: Mixer

Group Project: Week 1

- Mixer with the class, fill out [preferences survey](#)
- Receive groupings
- Meet your group
- List top project ideas

Individual Project: Week 1

Wrapping Up

Wrapping Up

- Individual project
 - Submit your jupyter notebook by Sunday, midnight PST
 - Absences: submit a video presentation for your lighting talk (in lieu of live presentation)
- Group project
 - Fill out survey tonight
 - Meet and list top project ideas
- After class email
 - Discord
 - Lecture video