Week 4: Neural Networks and Large Language Models

Question

Think

Imagine that everything in the universe is determined by some function. What would that function be?

Question

Think

Imagine that everything in the universe is determined by some function. What would that function be?

Discussion prompt

Why might stacking many simple transformations be more powerful than using just one?

Neural networks as composed functions

Core idea

A neural network is a sequence of layers. Each layer transforms the input and passes it forward.

$$f(x) = \sigma(Wx + b)$$

- W is the weight matrix
- b is the bias
- σ is the activation function

Function composition

$$f(x) = f_3(f_2(f_1(x)))$$

Each layer builds on the previous one.

add an exercise finding the output

Why multiple layers matter

Deep learning

Deep learning means using many layers.

- early layers learn simple features
- later layers combine them into richer representations
- depth increases expressive power

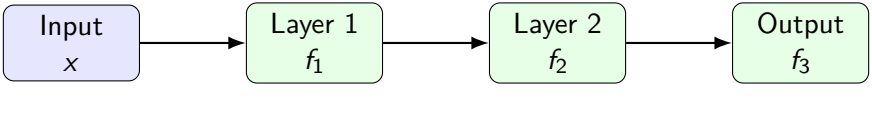
Key idea

A deeper network can represent more complex patterns than a shallow one.

Big idea

A neural network is not one complicated function; it is many simple functions stacked together.

A simple view of a neural network



Interpretation

Each layer changes the representation a little, but together they can learn something much richer.

Exercises: linear layers and ReLU

Exercise

Let

$$f_1(x, y) = \begin{pmatrix} x + y - 1 \\ x - y + 1 \end{pmatrix} \qquad \text{and} \qquad f_2(u, v) = \begin{pmatrix} 2u - v \\ u + v \end{pmatrix}$$

and let ReLU act componentwise:

$$\mathrm{ReLU}(z) = \max(0, z)$$

Compute:

$$\mathrm{ReLU}(f_2(\mathrm{ReLU}(f_1(1, 2))))$$

Challenge

What if we used the activation function tanh instead?

Large language models

Input representation

Words are transformed to tokens and then to vectors before the model processes them.

Core objective

Predict the next word, or more precisely, the next-token probability.

$$p(\text{next token} \mid \text{previous tokens})$$

- the model reads a sequence
- it transforms the sequence repeatedly
- it outputs a probability distribution over the vocabulary

Large language models: pre-training

Main idea

LLMs are trained on very large text collections before they are used for tasks.

Pre-training

During pre-training, the model learns patterns in language by predicting the next word or token.

- input: text converted into token IDs
- objective: predict the next token
- no human-written answers are needed for each example
- the model learns grammar, style, facts, and relationships from data

Attention and neural network layers

Neural network view

An LLM is built from many layers of functions:

$$f(x) = f_L(\cdots f_2(f_1(x))\cdots)$$

Each layer transforms the representation into something more useful.

Attention and neural network layers

Neural network view

An LLM is built from many layers of functions:

$$f(x) = f_L(\cdots f_2(f_1(x))\cdots)$$

Each layer transforms the representation into something more useful.

Attention

Attention helps the model decide which earlier words matter most for the current word.

- it helps the model capture relationships across a sentence
- it works using queries, keys, and values
- this idea came out in a paper in 2017

Key idea

Attention is a learned way focusing and contextualising the inputs.

Output: probabilities over words

Finally, we get:

$$P(\text{next token} \mid \text{previous tokens})$$

Interpretation

The model does not directly choose one word as certain truth. It gives a probability for each possible next token.

- the most likely token can be chosen
- or a token can be sampled from the distribution
- this is what makes generated text flexible and varied

Big picture

LLMs are large compositions of functions that transform sequences into probabilities.

How does what we learned come into it?

Input to output overview

Text $\rightarrow$ tokens $\rightarrow$ vectors $\rightarrow$ neural network with attention $\rightarrow$ probabilities $\rightarrow$ generated text

Tokenisation (unsupervised)

Separate the input into tokens.

Pre-training (self-supervised)

The model gets given a lot of text. It cuts off the last token and attempts to predict it. This is when the initial vectors for each token are determined.

Preference tuning (Reinforcement learning)

Humans give feedback on outputs and the model learns which responses are preferred.

Recap

Today

- neural networks are composed of layers of functions
- deep learning uses many layers to increase expressive power
- LLMs map text to vectors and predict the next token
- attention helps connect different parts of a sequence