M-Lab CARTE AI Workshop 2025

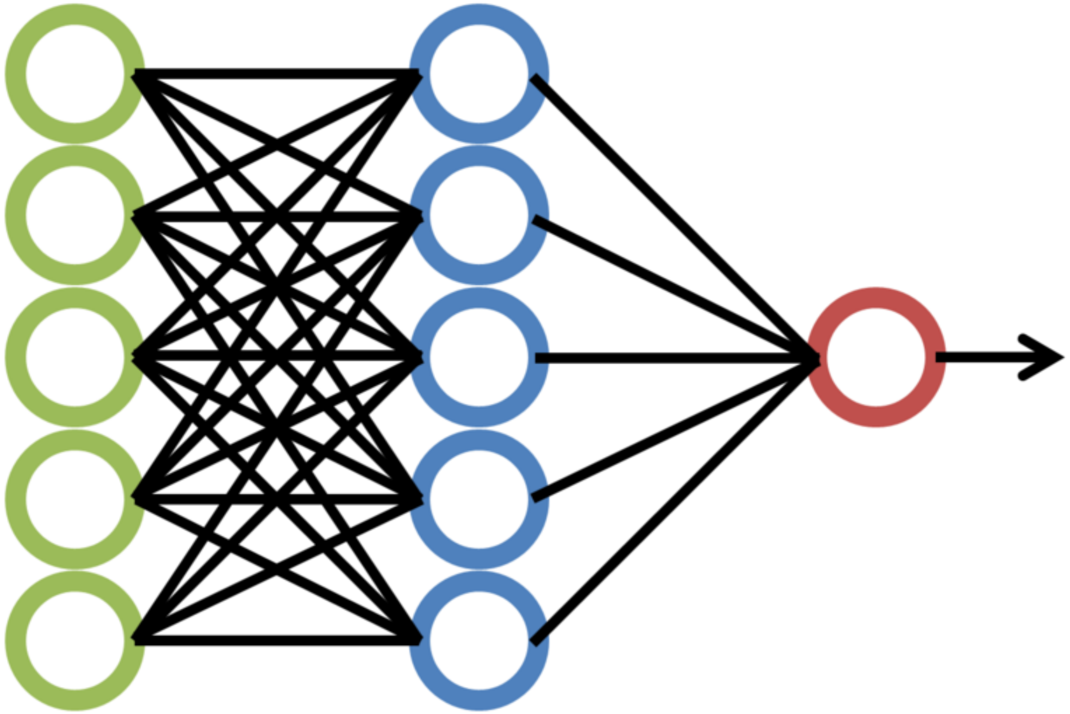
Neural Networks & Optimization

Overview

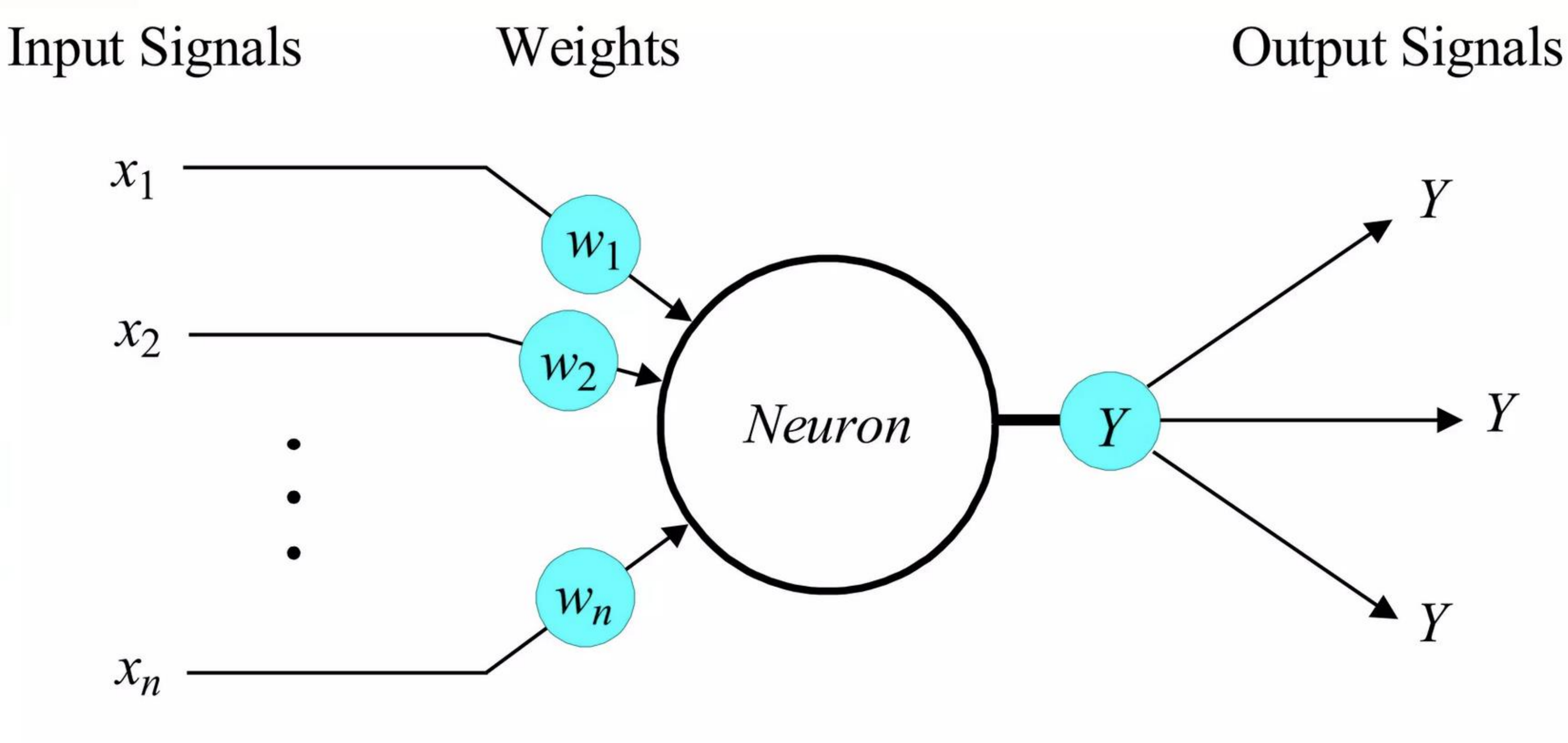
- This afternoon we will discuss the fundamentals of neural networks
- Neural networks drive nearly all modern AI tools
- We will also look at optimization
- Optimization is an example of a non-Machine Learning area of Artificial Intelligence

What is a neural network?

- Complex structure of interconnected computing nodes (neurons)
- Can identify patterns and trends in complex data
- NNs operate on the principle of “learning” from data, using a process that mimics how biological brains learn

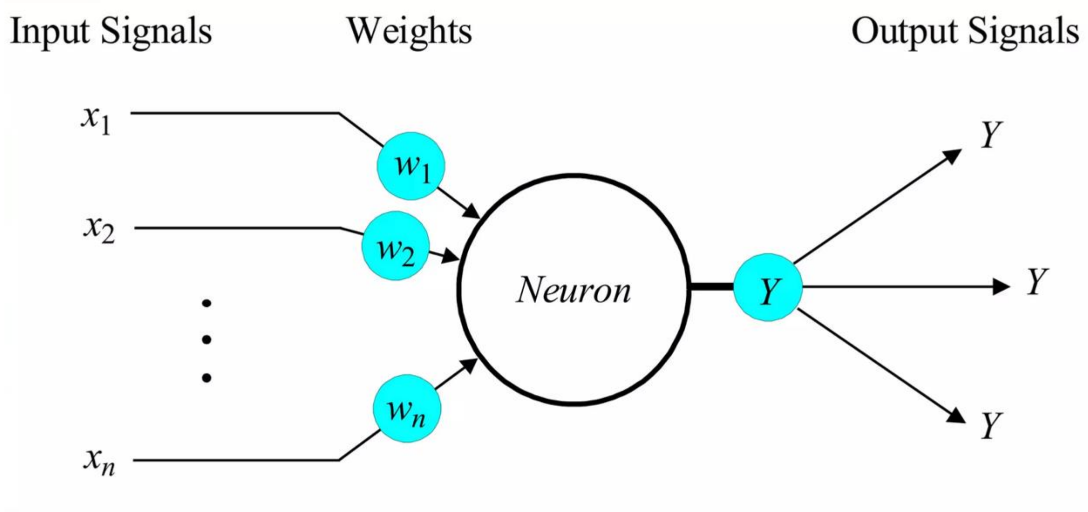


View inside an artificial neuron



View inside an artificial neuron

- Behaves like a linear regression model:
- $w_1 x_1 + w_2 x_2 + \ldots + w_n x_n$
- Weights correspond to how much the neuron “cares” about each input



Visualization

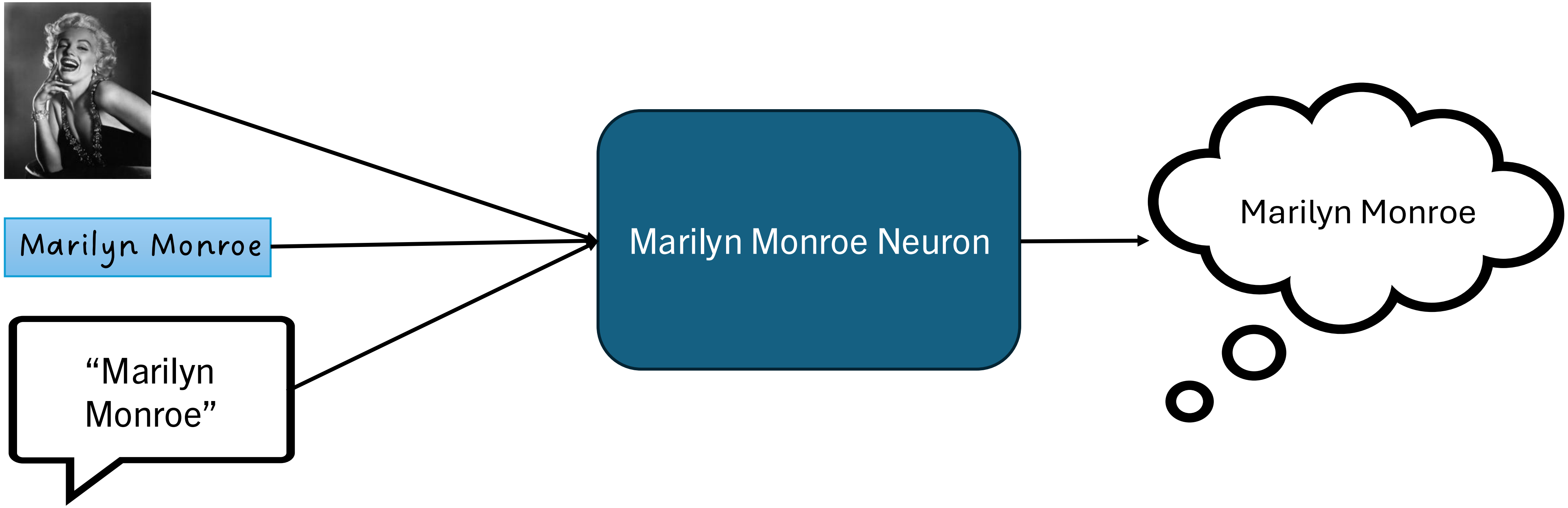
https://neuron.carte.training/

The Marilyn Monroe neuron

- Study conducted on patients with epilepsy
- Researchers use specialized equipment to measure the “excitement” of individual neurons in a patient’s brain
- Measuring a neuron, the researchers showed patients a series of images
- In each patient, they found around five neurons that fired when the patient looked at a specific person

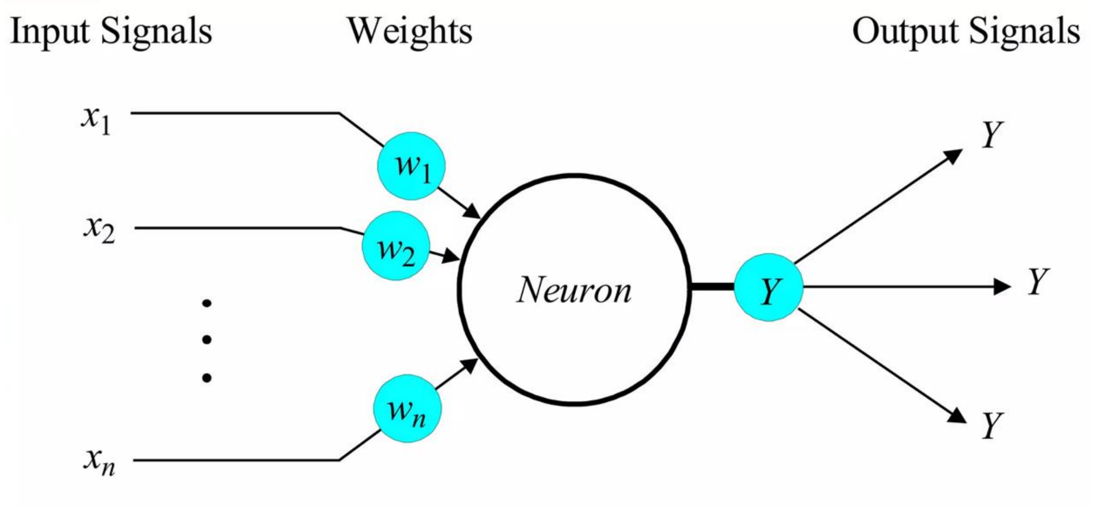
Back to the brain: the Marilyn Monroe Neuron

- Once a “celebrity” neuron was identified, the researchers wanted to know if it would still fire for representations other than images



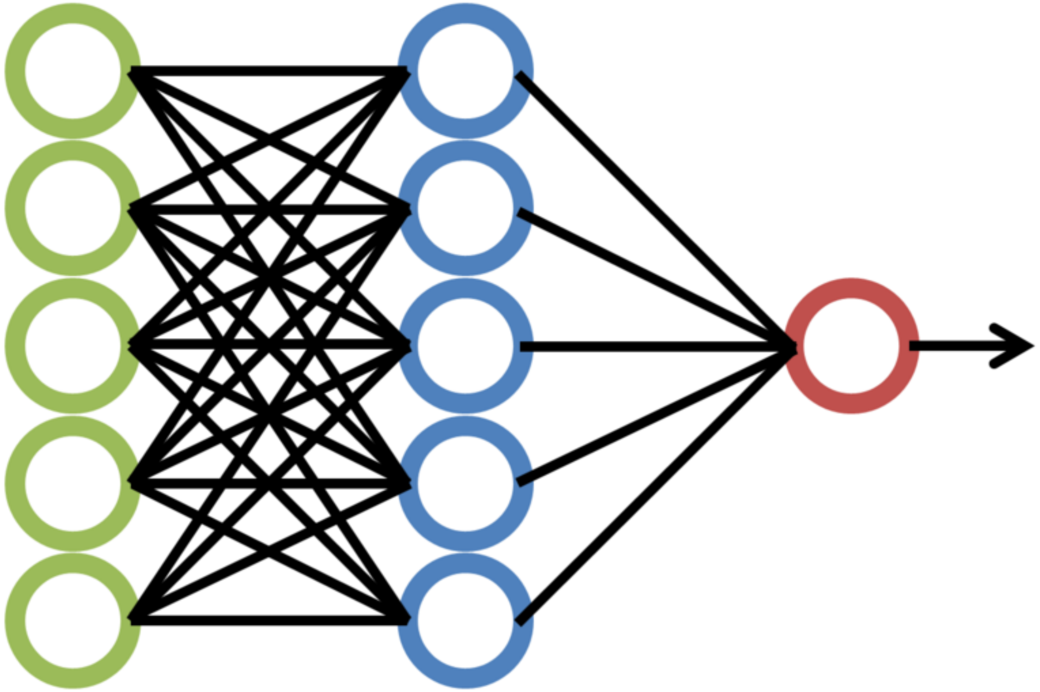
Marilyn Monroe ANN

- Weights would be high from neurons that react to different representations of Marilyn Monroe
- Weights would be low for neurons that react to other people, or concepts



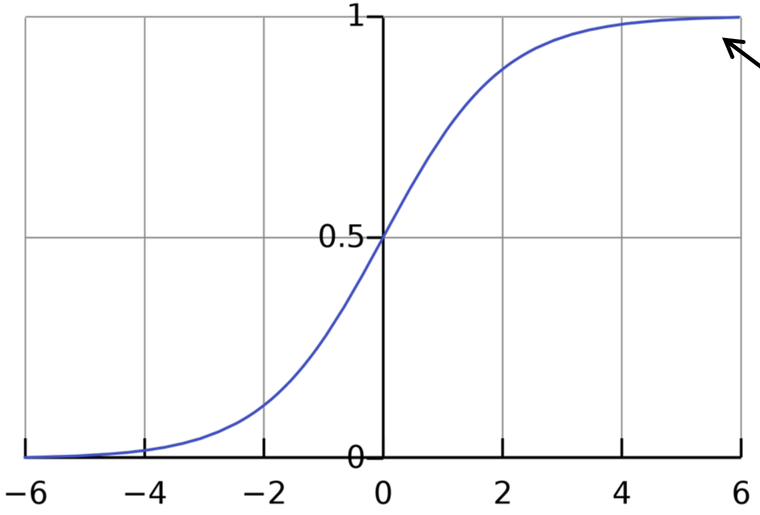
ANNs

- Each neuron considers the responses of the neurons in the previous layer
- It learns to pay attention to the neurons that are excited about what it's excited about
- Ignores the neurons that are excited about other things



Activation Functions

- Basic approach: when I see enough activity, I get excited
- Below threshold: 0
- Above threshold: 1
- More useful: gradually increase excitement as we see more activity
- In practice: many different activation functions!



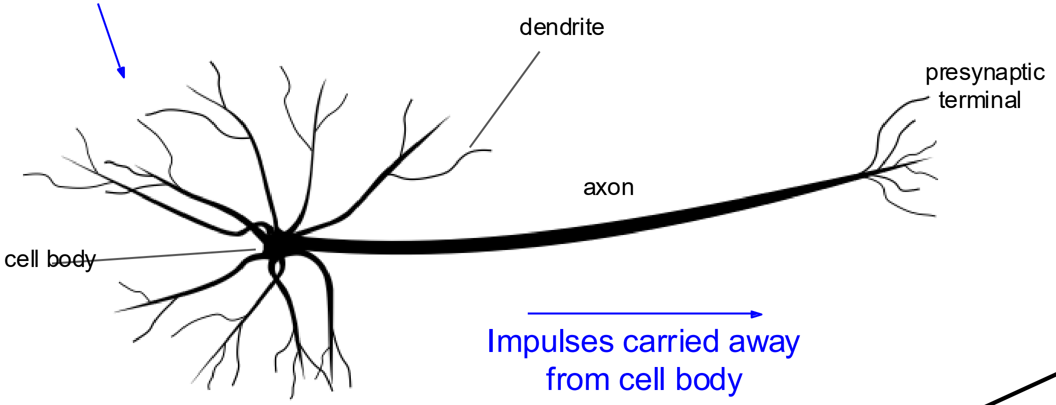
Desired value gets max output

The Neuron Metaphor

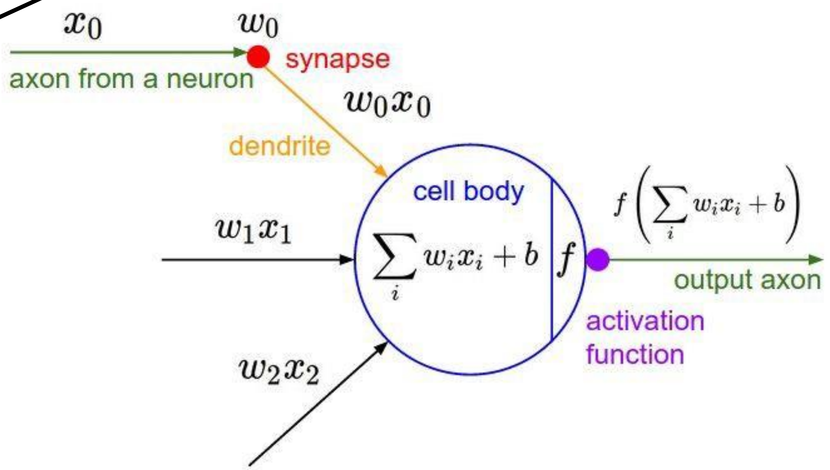
- Neural networks were inspired by our understanding of the brain and how neurons interact.
- An artificial neuron in a neural network takes in multiple inputs, applies a function to them, and generates an output – mirroring the basic functionality of a biological neuron.
- This analogy has been extremely useful for explaining and visualizing how these artificial structures work.

The Neuron Metaphor

Impulses carried toward cell body

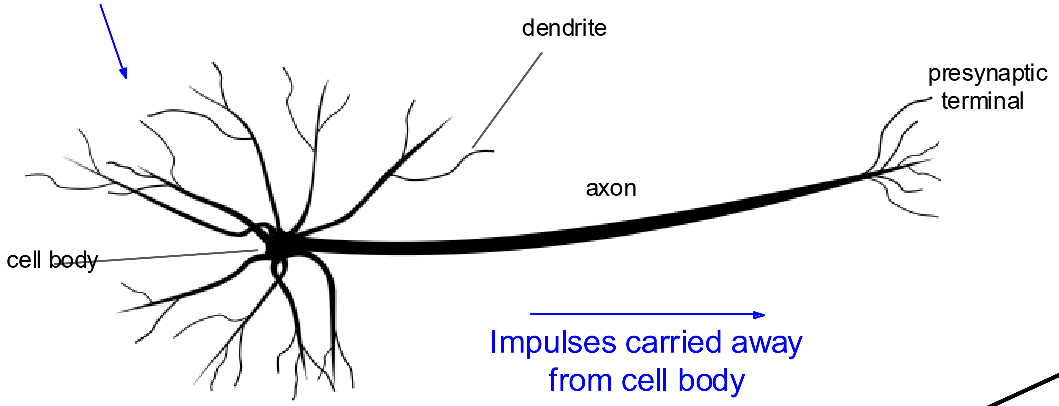


Impulses carried away from cell body

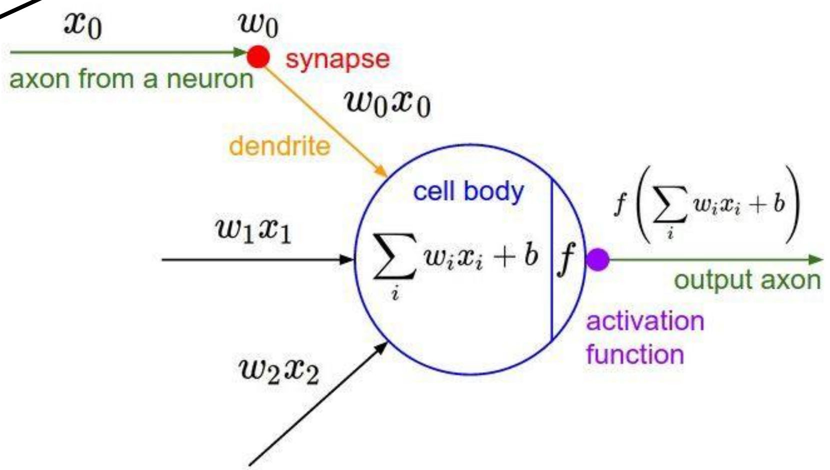


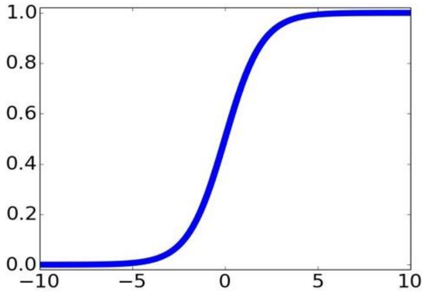
The Neuron Metaphor

Impulses carried toward cell body



Impulses carried away from cell body





sigmoid activation function

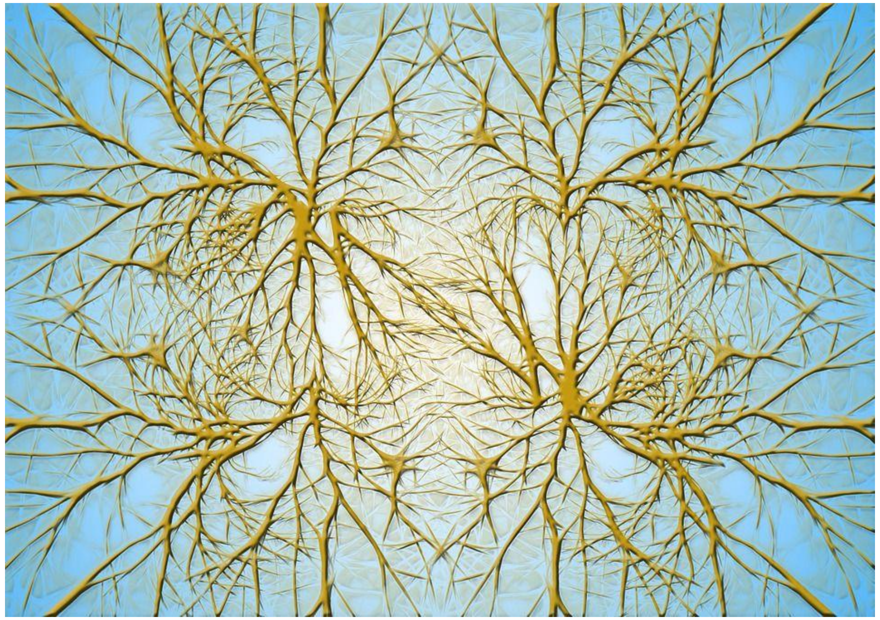
$$\frac{1}{1+e^{-x}}$$

Training the Network

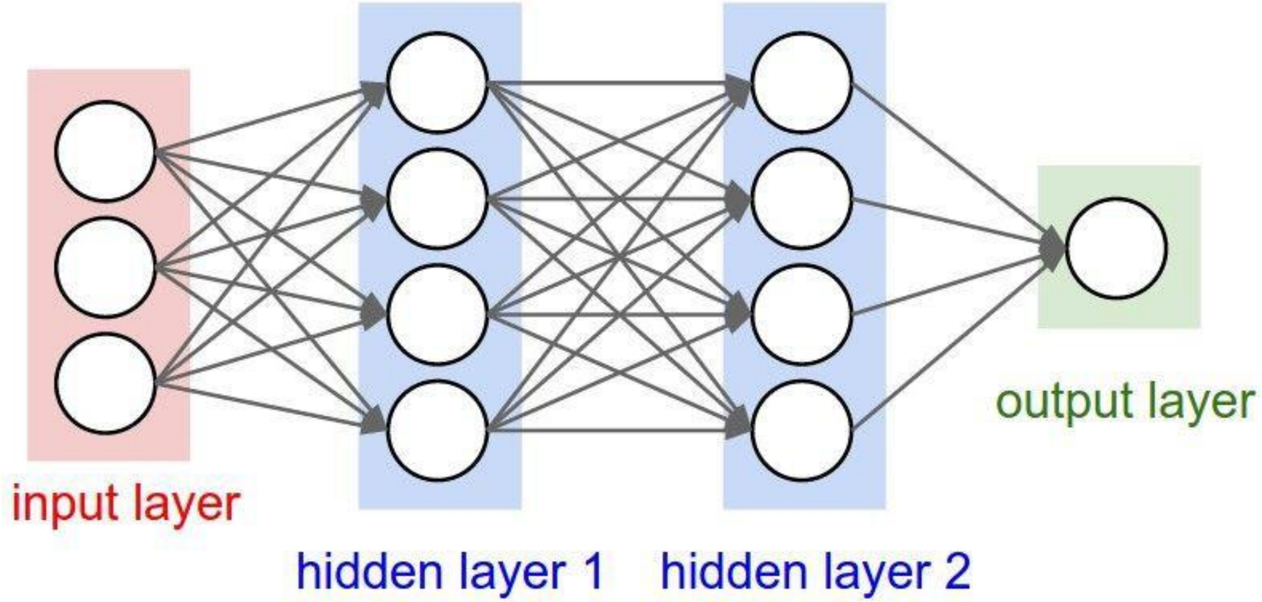
- Find parameters that **minimize the total error**
- **Loss** for a given sample is the total error in predictions made
- Going through the network, the predictions are dependent on the settings of the parameters
- We have a mathematical function representing the network
- A way of measuring how ”good” it is

The Metaphor Breaks Down

<u>Biological Neurons:</u>
Complex connectivity patterns



<u>Neurons in a neural network:</u>
Organized into regular layers for computational efficiency



The Metaphor Breaks Down

- Biological neurons are vastly more complex: they use a mixture of electrical and chemical signals, have complex temporal dynamics, and can restructure their own connections.
- The brain is not just a feed-forward network: it has many complex feedback loops, which are not typically found in artificial neural networks.
- The brain isn't easily divided into distinct layers, as we do in artificial neural networks.

The Metaphor Breaks Down

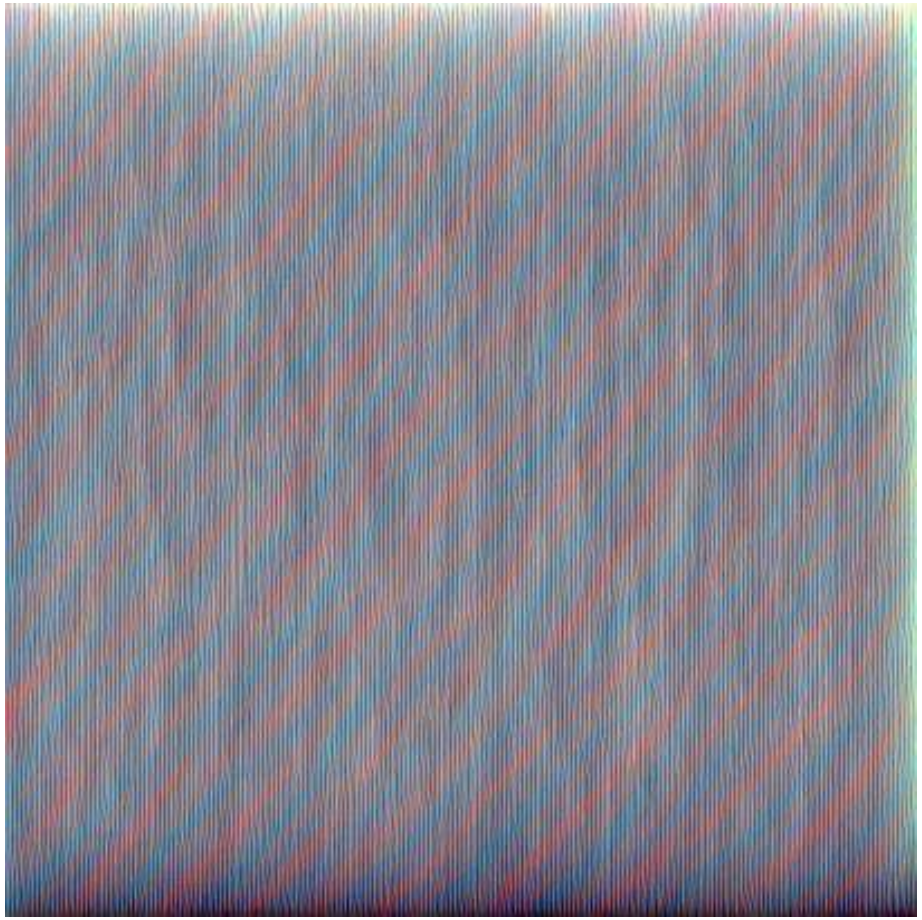
- Over-reliance on the analogy can lead to misunderstandings about how neural networks function and their capabilities.
- This can lead to unrealistic expectations about what neural networks can do, or to overgeneralizations about their functioning.
- For instance, claiming a neural network "thinks" or "understands" like a human brain is misleading.
- To further progress, it's important to view artificial neural networks as mathematical/statistical tools, and not overstate the comparison to the human brain.

Going past the fully connected network

- In many image tasks, we want to be able to recognize something regardless of where it is in the image
- For fully-connected networks, the order of the inputs is fixed
- No “shift invariance”

Going past the fully connected network

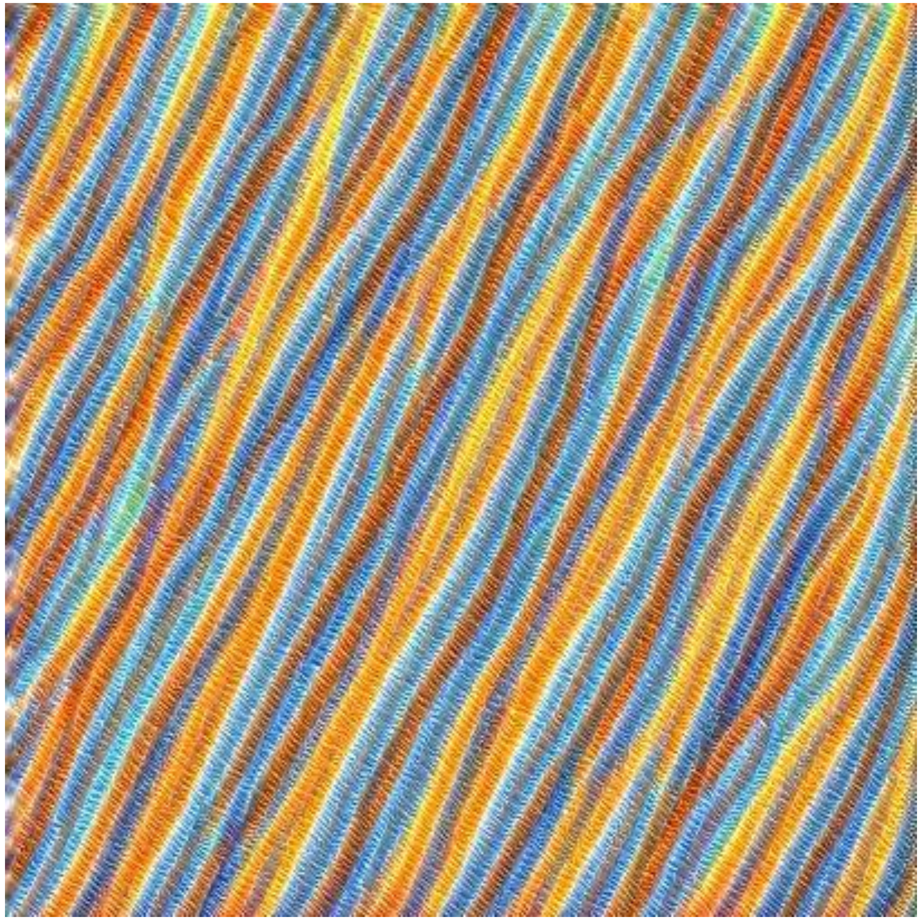
- In the 1950s and 60s, researchers showed that the brain contains neurons which respond to specific patterns, regardless of where they appear
- Combinations of very basic patterns can then be recognized as a more complicated one!



VGG-16, neuron in layer 7

Going past the fully connected network

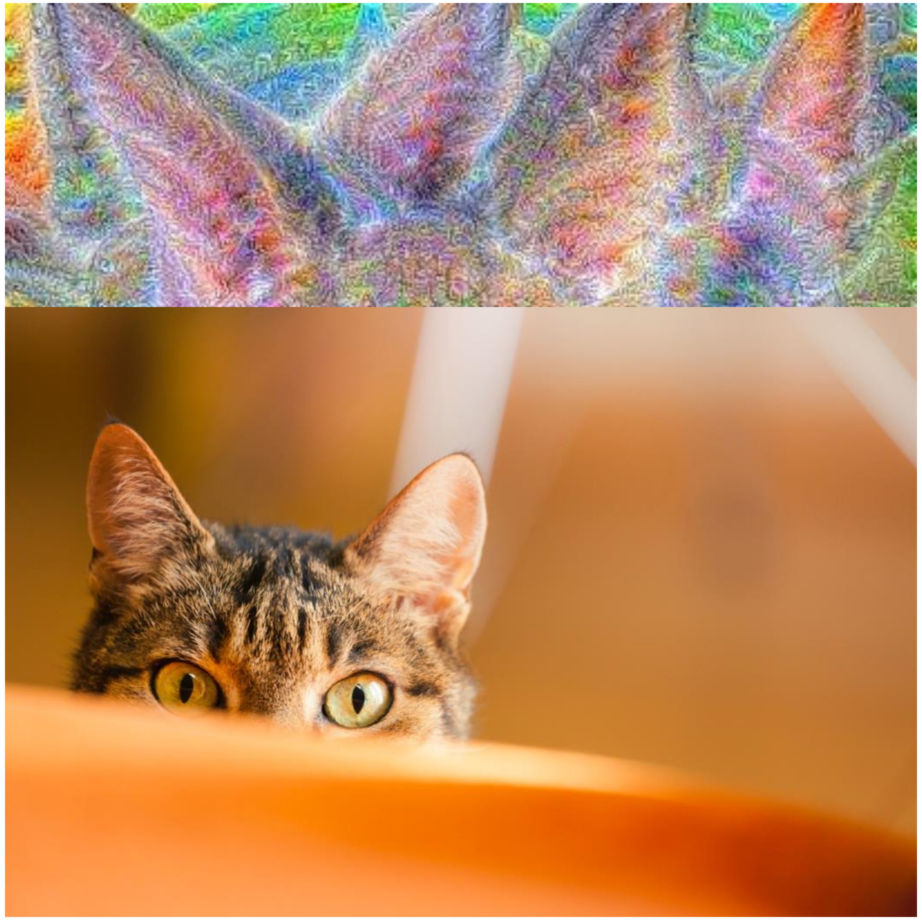
- In the 1950s and 60s, researchers showed that the brain contains neurons which respond to specific patterns, regardless of where they appear
- Combinations of very basic patterns can then be recognized as a more complicated one!



VGG-16, neuron in layer 14

Going past the fully connected network

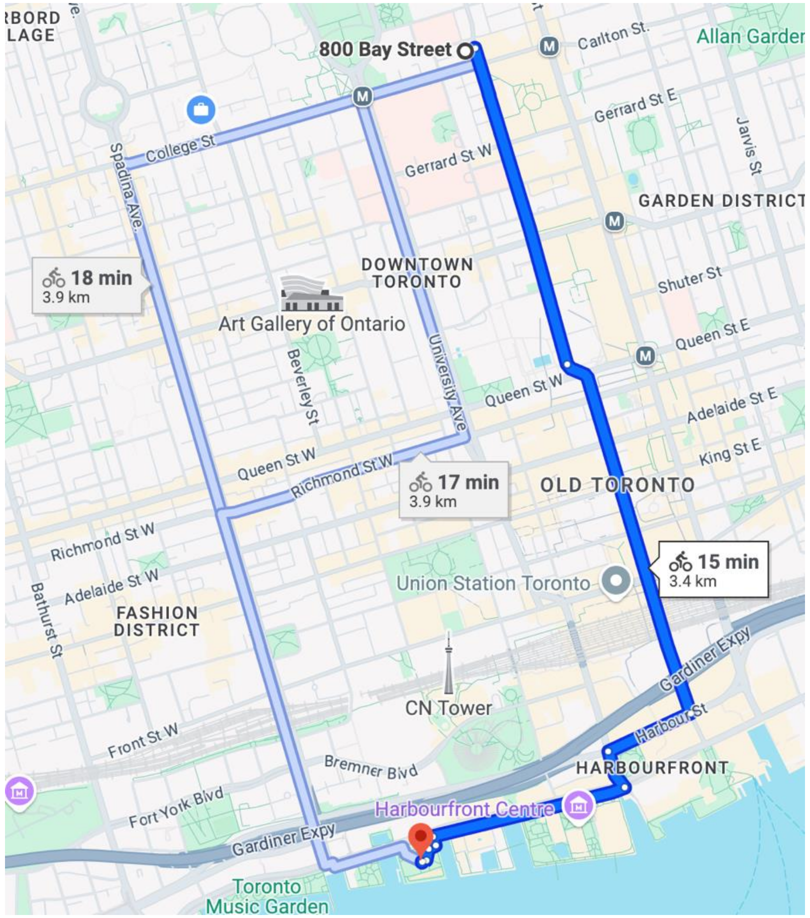
- In the 1950s and 60s, researchers showed that the brain contains neurons which respond to specific patterns, regardless of where they appear
- Combinations of very basic patterns can then be recognized as a more complicated one!



VGG-16, neuron in layer 40

What is Optimization?

- Major field within Data Analytics, Operations Research and Management Science
- Basic idea: find the values of the decision variables that maximize (or minimize) the objective value, while staying within the constraints
- How do I find the shortest route to bike to the harbourfront, without breaking traffic laws?



What is Optimization?

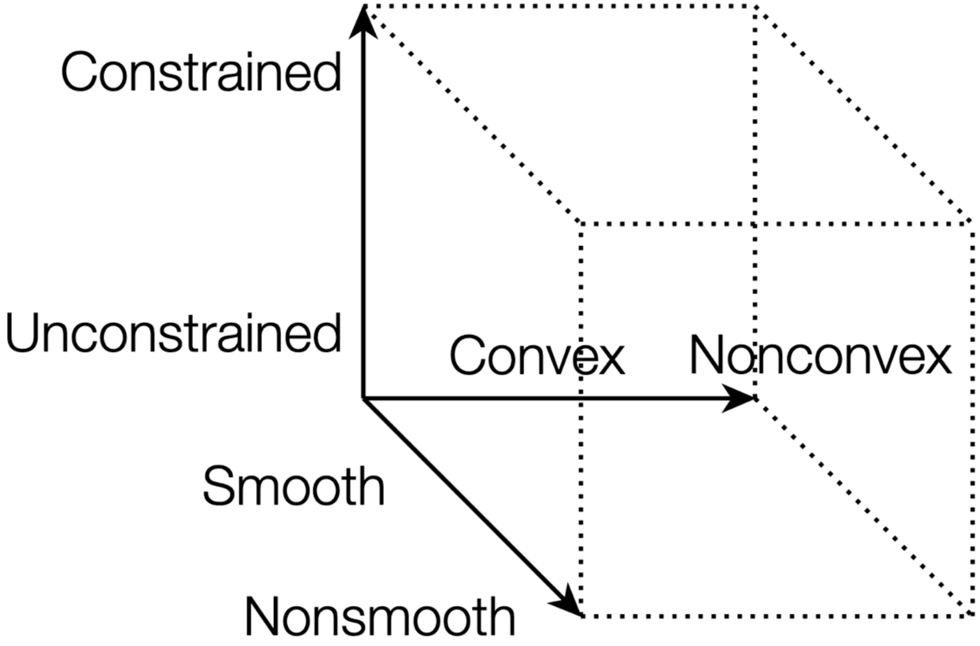
- In machine learning, we usually want to minimize the result of a loss function
- A huge number of ML problems can be solved using optimization
  - e.g. regression, classification, maximum likelihood
- If we can use optimization, we get access to powerful tools which can find our answer

GUROBI OPTIMIZATION

Google OR-Tools

Classes of optimization problem

- Many different types of problem can be framed as an optimization problem
- Three main distinctions help to define them
- Constrained vs Unconstrained
- Convex vs Nonconvex
- Smooth vs Nonsmooth (less important)



Constrained vs Unconstrained

- Constraints are conditions on what answers are acceptable
- When finding the shortest driving route, you are really finding the shortest *legal* driving route
- When scheduling employees, have to factor in their availability

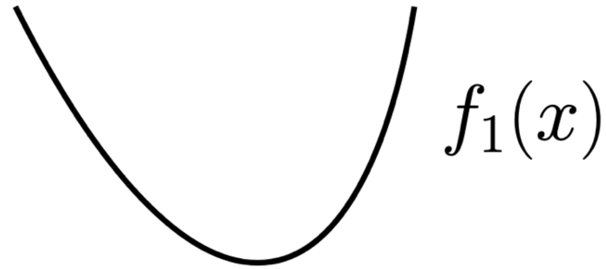
$$\underset{x}{\text{minimize}} \quad f(x)$$

vs

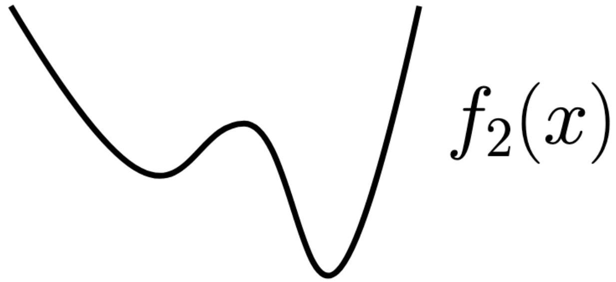
$$\begin{aligned} \underset{x}{\text{minimize}} \quad & f(x) \\ \text{subject to} \quad & g_i(x) \leq 0, \quad i = 1, \dots, m \\ & h_i(x) = 0, \quad i = 1, \dots, p \end{aligned}$$

Convex vs Nonconvex

- A function is convex if there is exactly one “bottom” point – the global minimum
- This makes the problem much easier to solve because as long as the error is decreasing, you are getting closer to the best answer
- If the function is nonconvex, you can be “tricked” by a local minimum



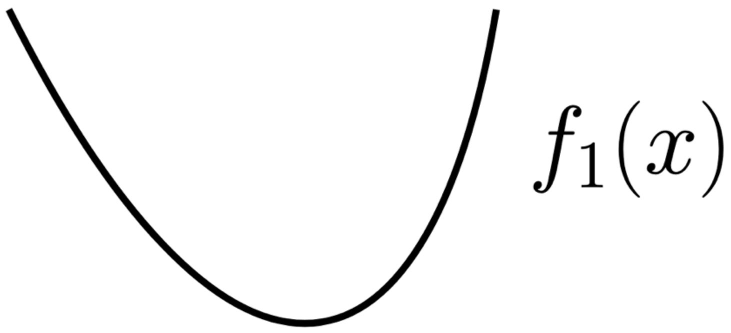
Convex function



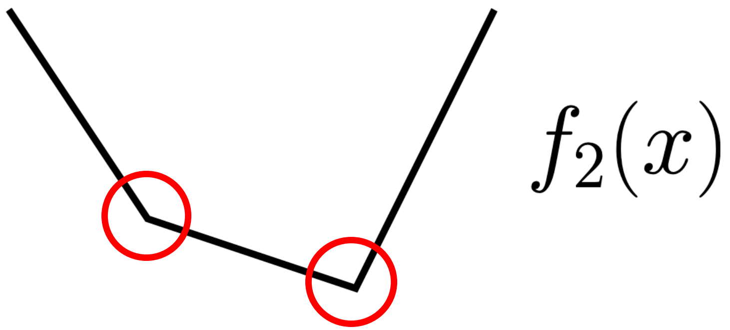
Nonconvex function

Smooth vs Nonsmooth

- Many modelling methods depend on calculating the derivative of the error – this tells us how to change our answer to get closer to the minimum
- If the function is nonsmooth, there are points (red) where it is not possible to differentiate



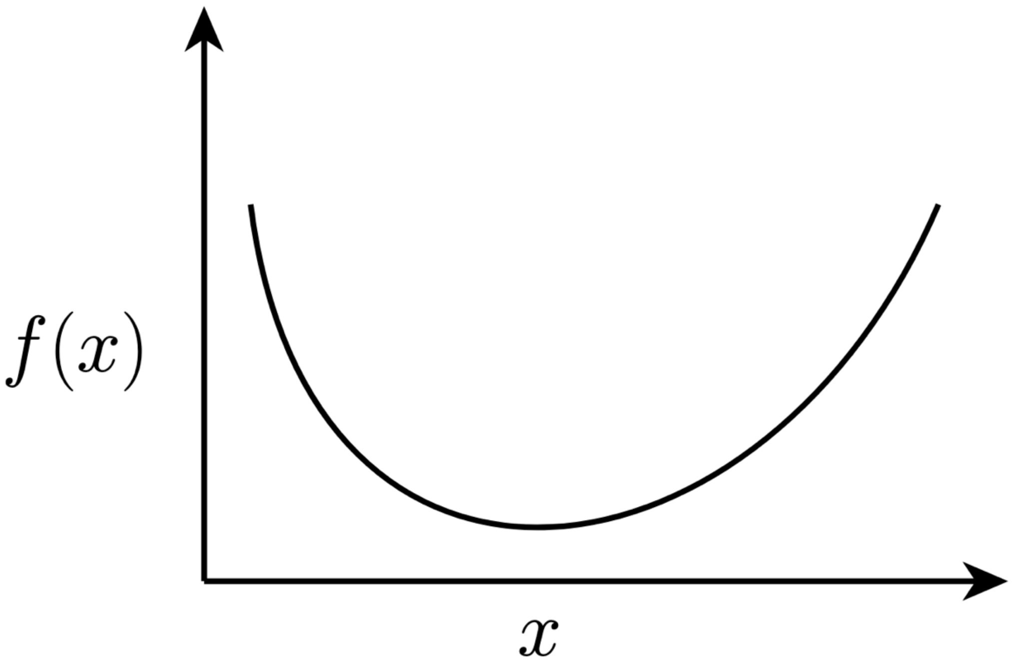
Smooth function



Nonsmooth function

Solving an optimization problem

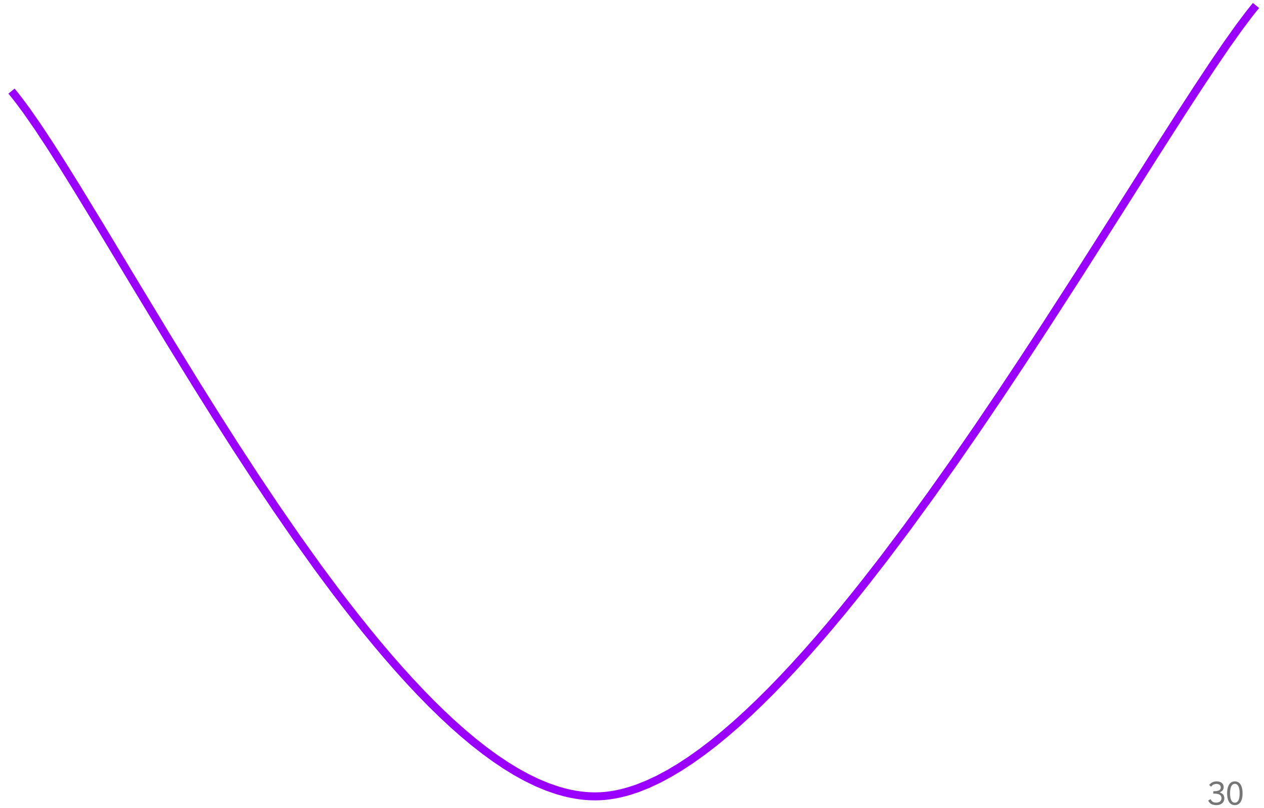
- Let's start with the simplest case: unconstrained, convex, smooth function
- We just need to find the point where the curve is flat (i.e. derivative is zero) - this is the minimum
- If the function is very simple, we can just calculate this value directly
- Otherwise, we can use gradient descent



Gradient descent

For some loss function $L(\mathbf{w})$, gradient $\nabla L(\mathbf{w})$ points towards in direction of steepest ascent.

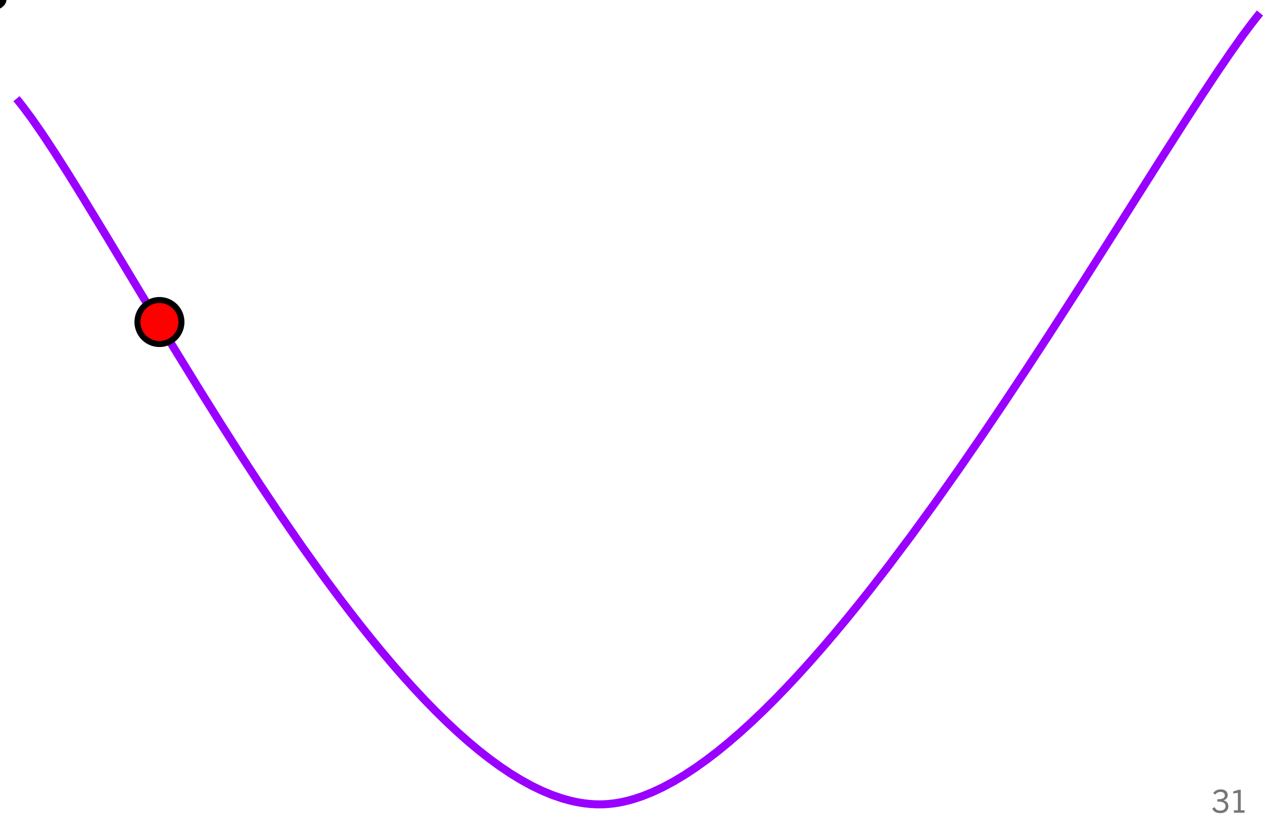
In 1d, either points left or right



Gradient descent

For some loss function $L(\mathbf{w})$, gradient $\nabla L(\mathbf{w})$ points towards in direction of steepest ascent.

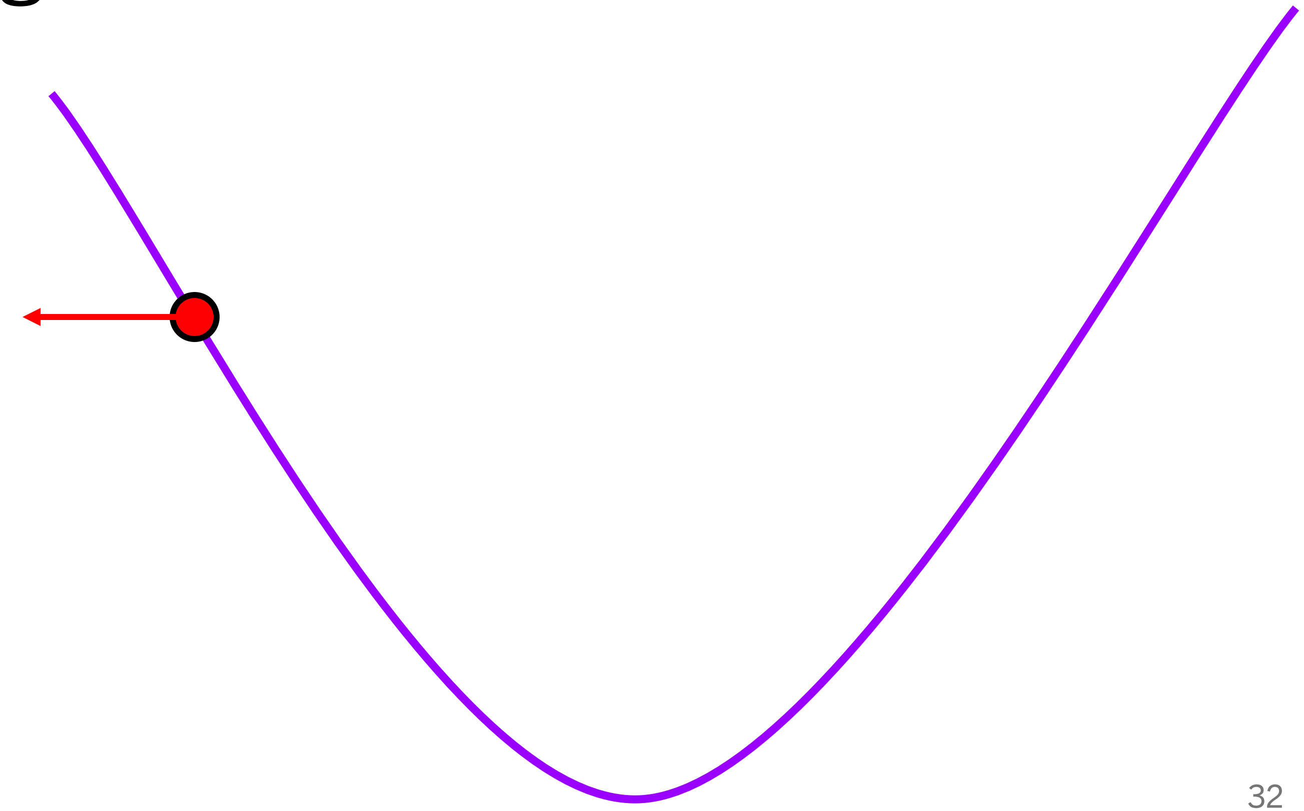
In 1d, either points left or right



Gradient descent

For some loss function L(**w**), gradient ∇L(**w**) points towards in direction of steepest ascent.

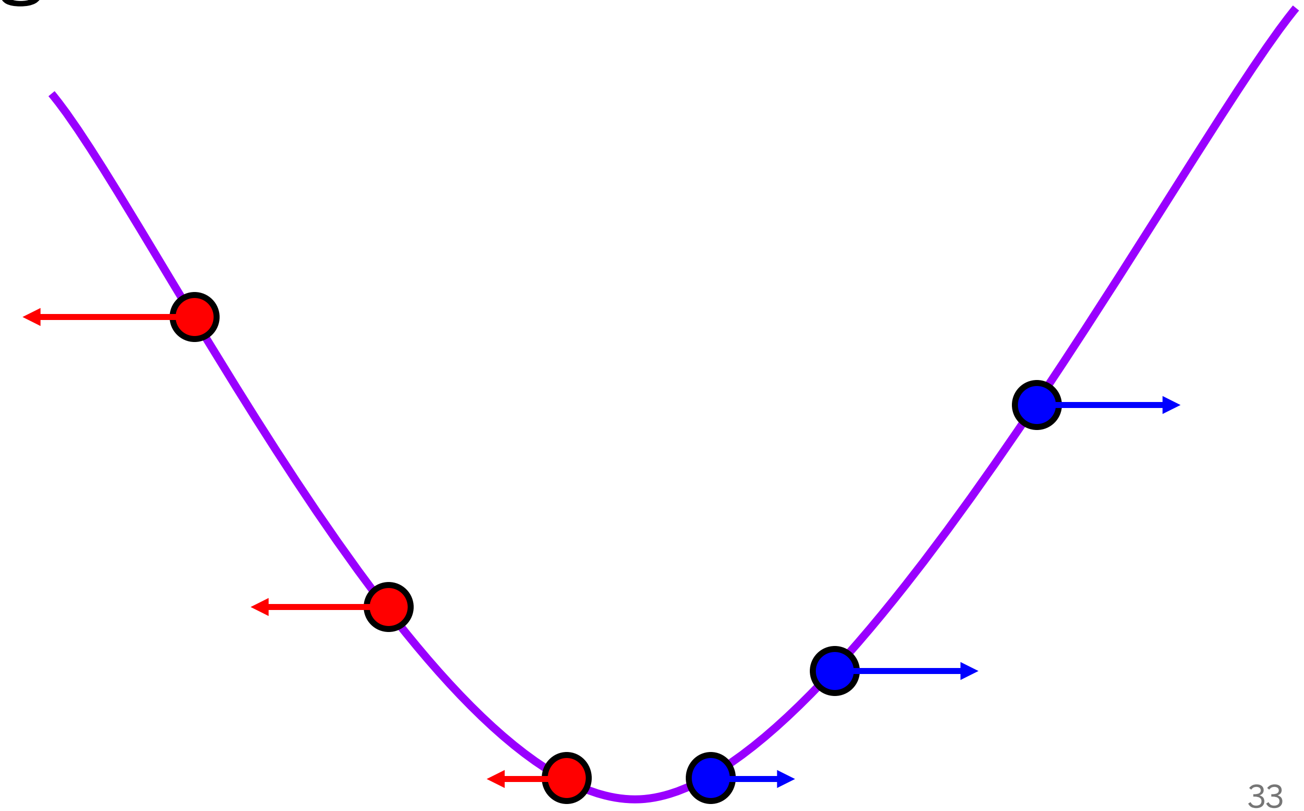
In 1d, either points left or right



Gradient descent

For some loss function L(**w**), gradient ∇L(**w**) points towards in direction of steepest ascent.

In 1d, either points left or right



Gradient descent

For some loss function L(**w**), gradient ∇L(**w**) points towards in direction of steepest ascent.

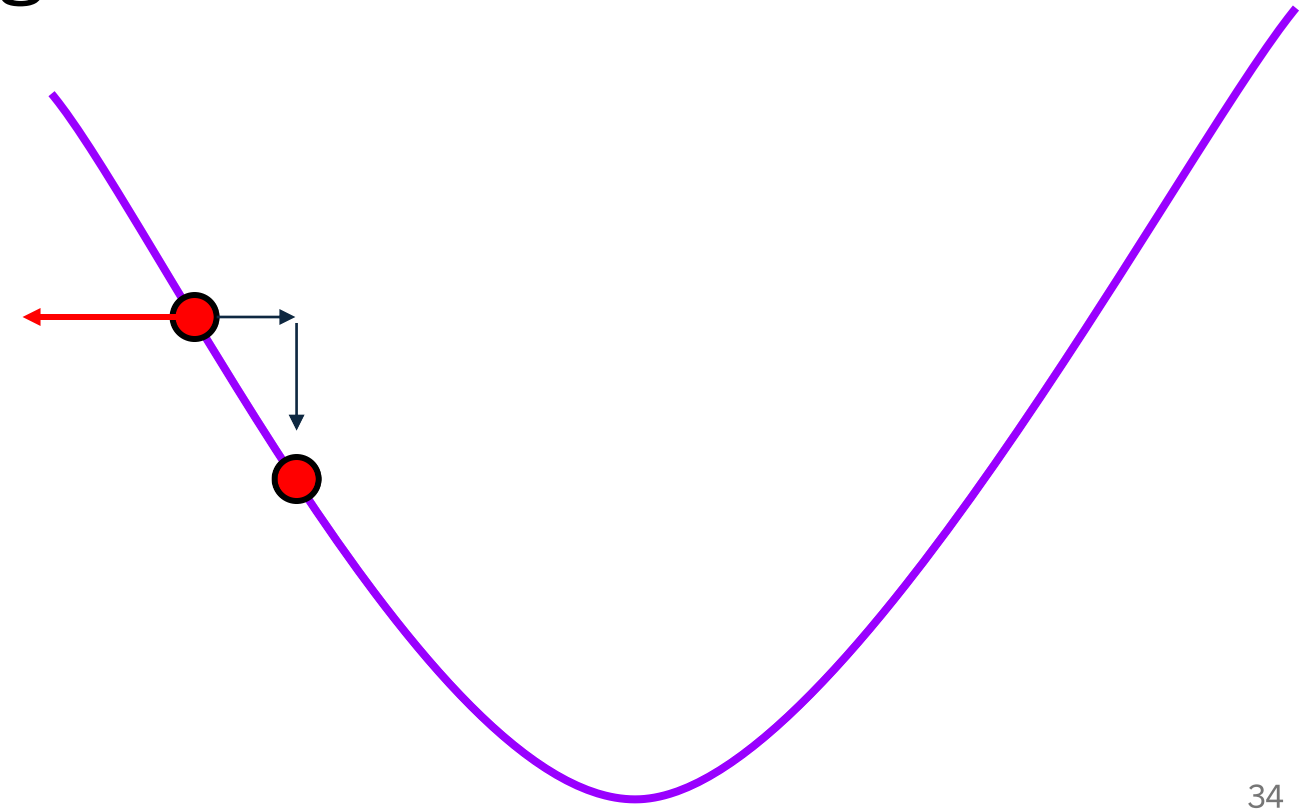
In 1d, either points left or right

Algorithm:

Take derivative

Move slightly in other direction

Repeat



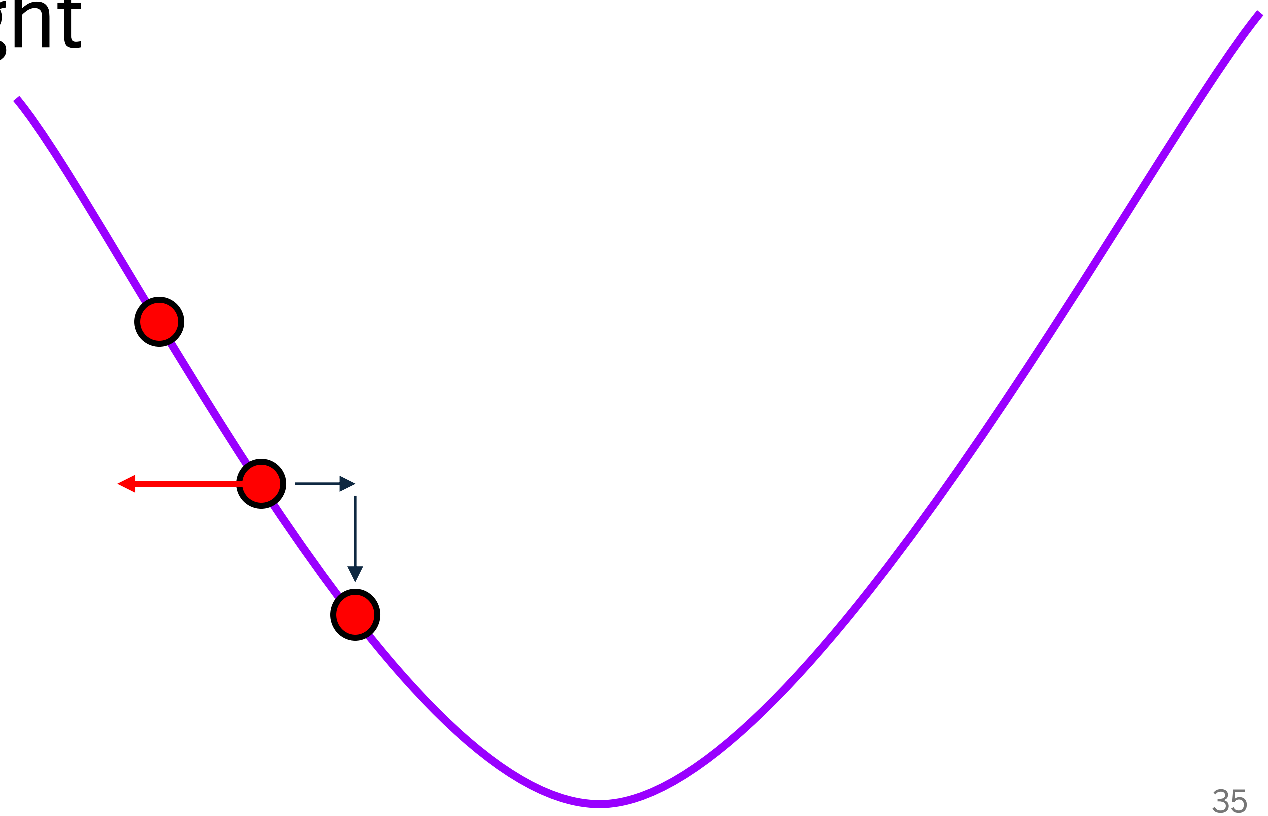
Gradient descent

For some loss function L(**w**), gradient ∇L(**w**) points towards in direction of steepest ascent.

In 1d, either points left or right

Algorithm:

Take derivative
Move slightly in other direction
Repeat



Gradient descent

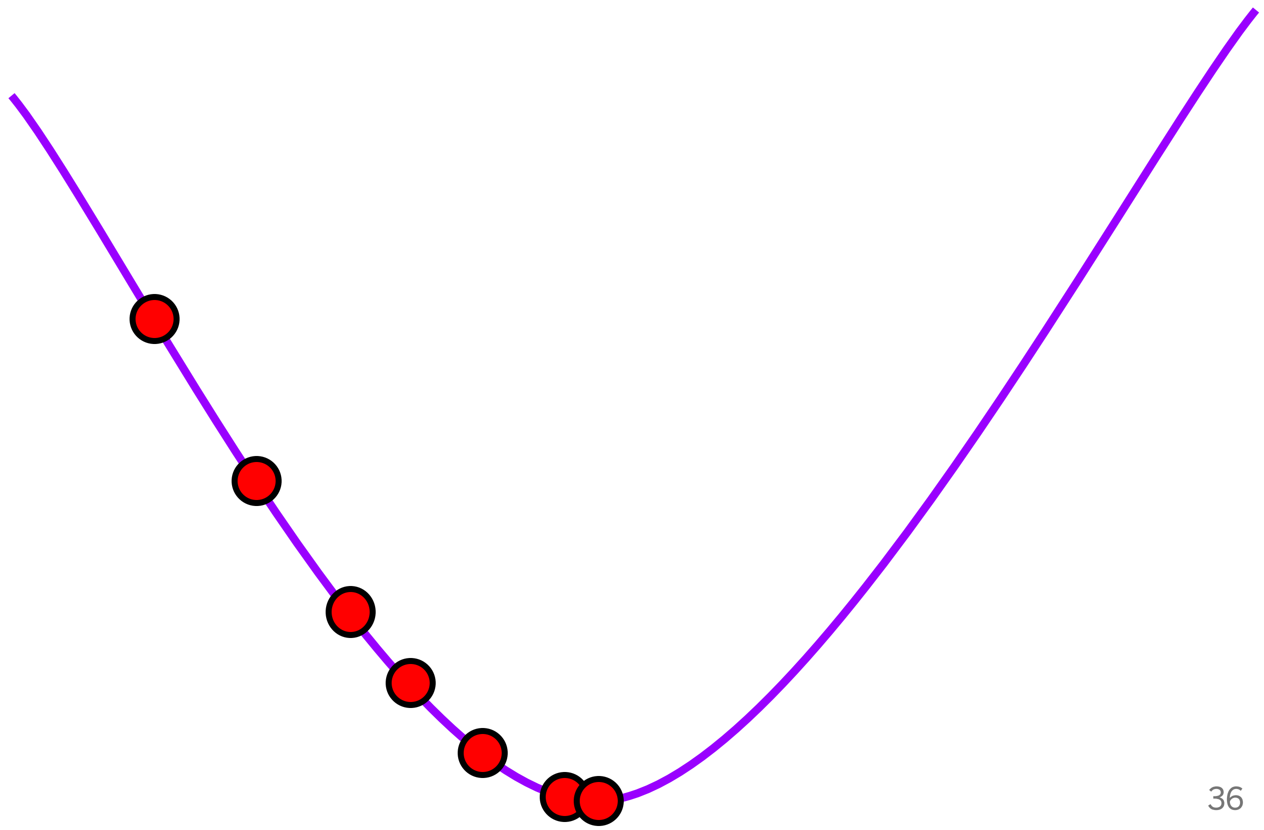
Algorithm:

Take derivative

Move slightly in other direction

Repeat

End up at local optima

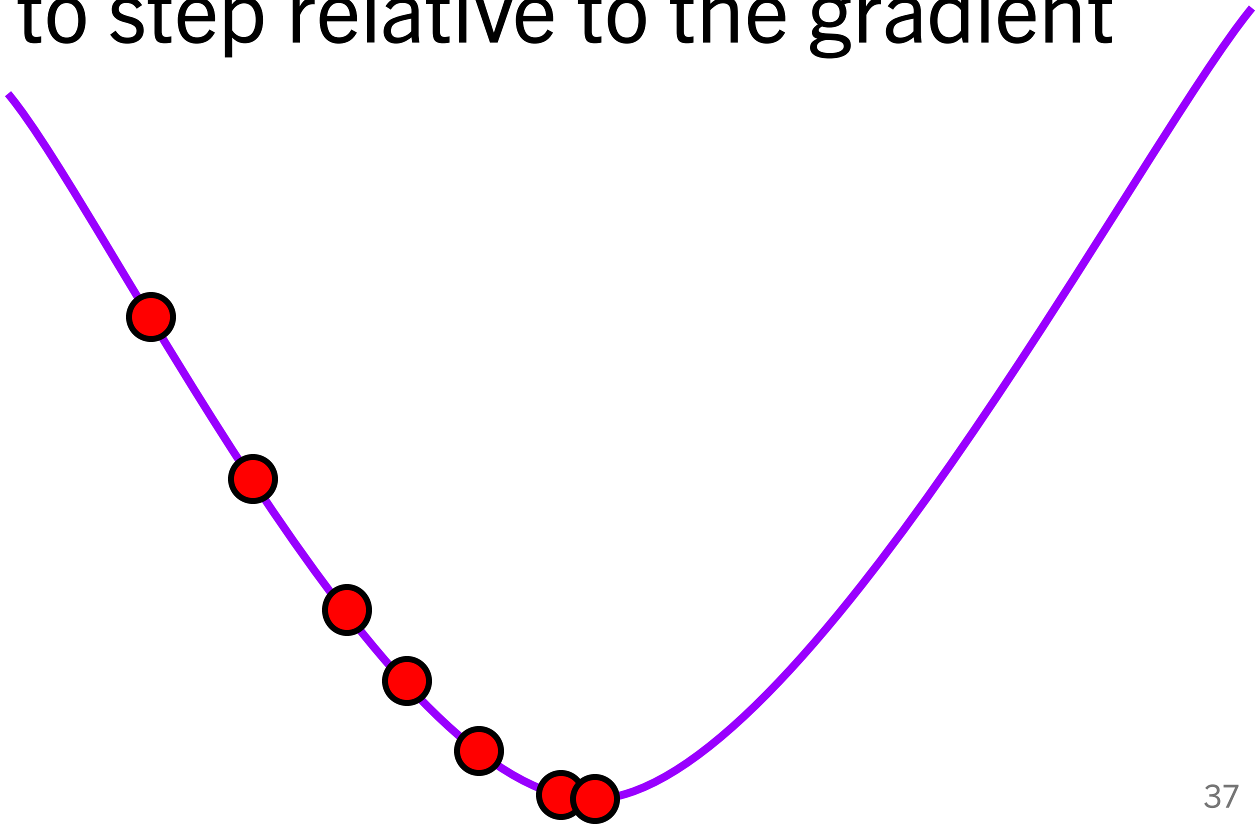


Gradient descent

Formally:

$\mathbf{w}_{t+1} = \mathbf{w}_t - \eta \nabla L(\mathbf{w})$

Where η is *step size*, how far to step relative to the gradient



Gradient Descent Visualization

https://gradient.carte.training/