Stochastic Gradient Descient

CPSC 383: Explorations in Artificial Intelligence and Machine Learning
Fall 2025

Jonathan Hudson, Ph.D.
Associate Professor (Teaching)
Department of Computer Science
University of Calgary

August 27, 2025

UNIVERSITY OF CALGARY

Review

Training a neural net is is the process of adjusting _____to minimize _____across all of the _____data.

Training

Definition

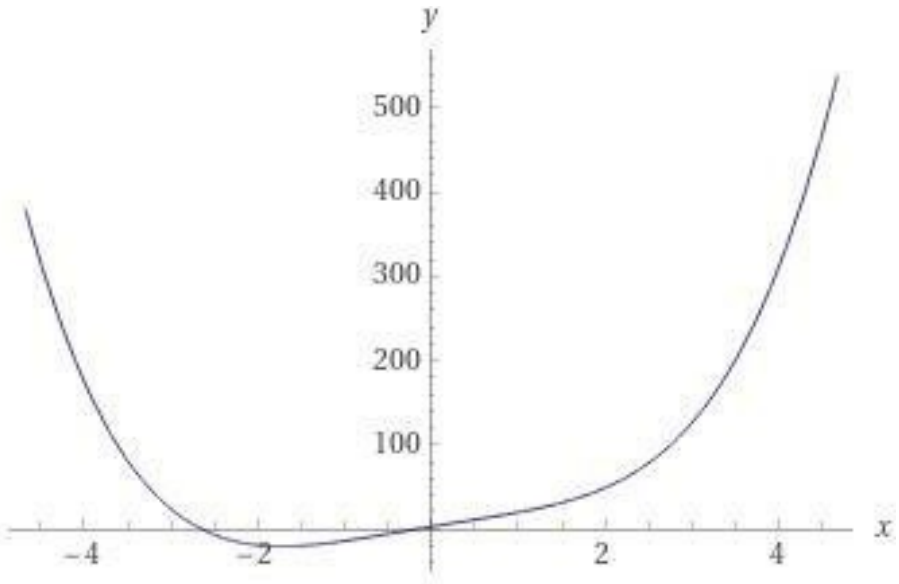
Optimization is a set of problems that involve selecting the “best” element from some set of values.

- Usually values maximizing or minimizing some objective function

Optimization problems can be **discrete** or **continuous**.

Example

- Suppose you wanted to find the “best” value of x, i.e. the one that minimizes the cost function $f(\mathrm{x}) = x^4 - x^2 + 17x + 3$ shown below.



Optimization

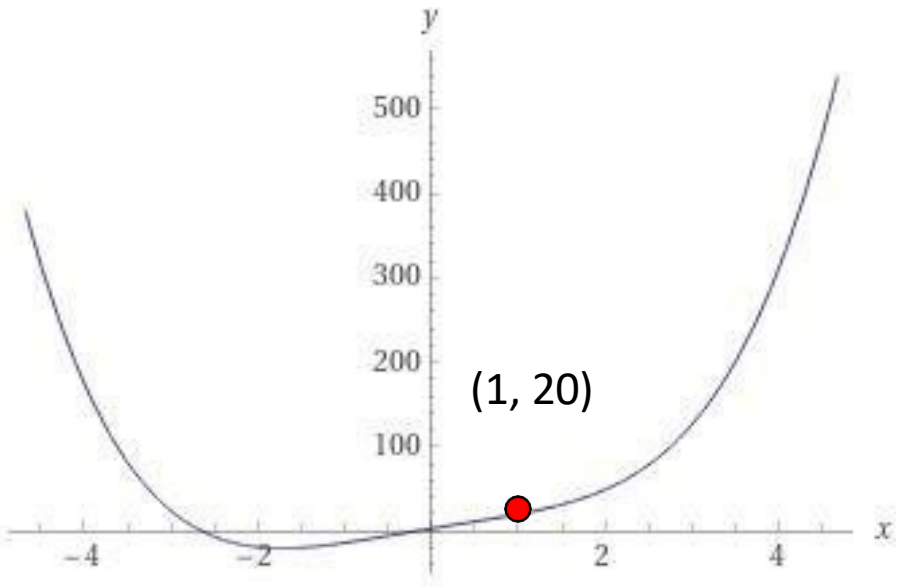
In general, optimization problems are **intractable**, i.e. an exact solution is either computationally expensive or impossible to obtain.

However, in practice we are usually happy with a “good enough” solution that is easier to find.

Optimization

Idea: what if you just want to find something better than what you currently have?

$$f(x) = x^4 - x^2 + 17x + 3$$

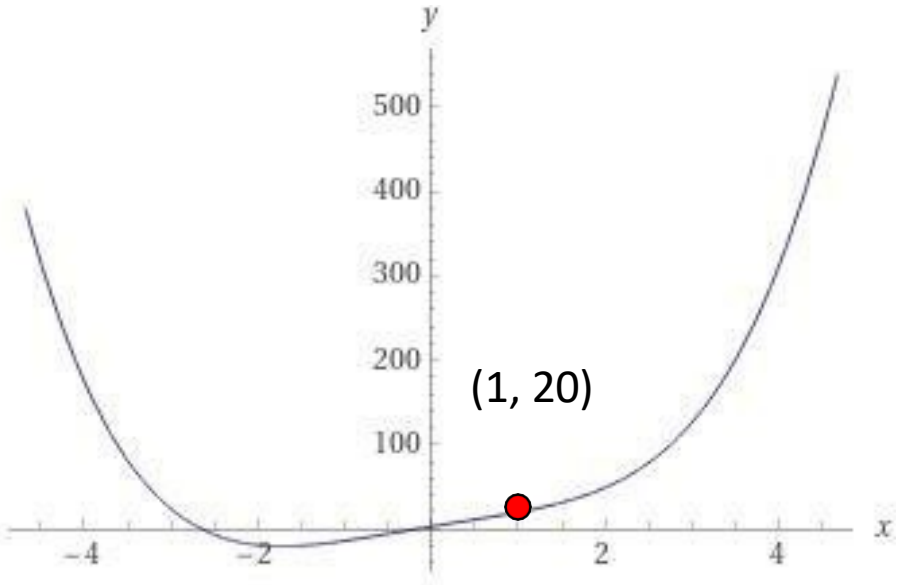


Optimization

Idea: what if you just want to find something better than what you currently have?

$$f(x) = x^4 - x^2 + 17x + 3$$

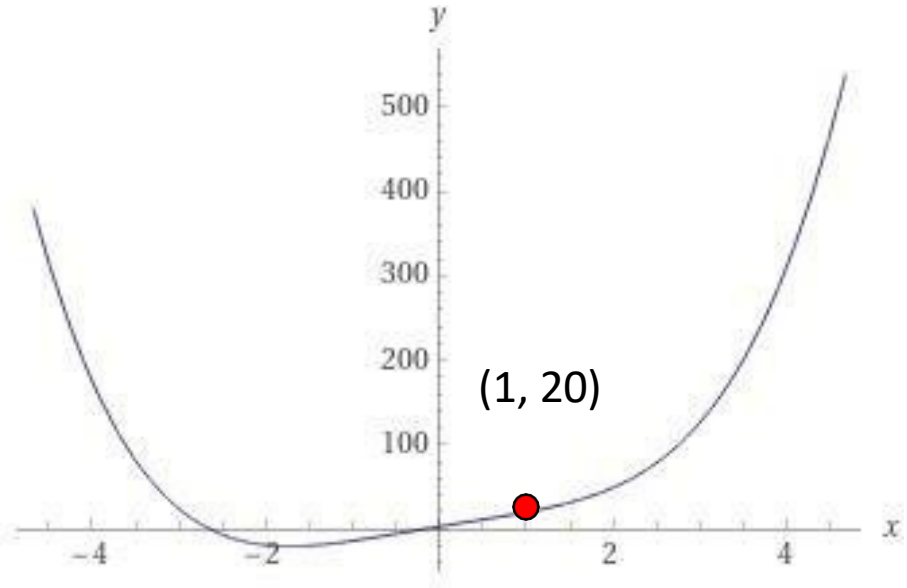
$$f'(x) = 4x^3 - 2x + 17$$



Optimization

Idea: what if you just want to find something better than what you currently have?

$$f(x) = x^4 - x^2 + 17x + 3$$
$$f'(x) = 4x^3 - 2x + 17$$



$f'(1) = 19 > 0$
so move left!

Optimization

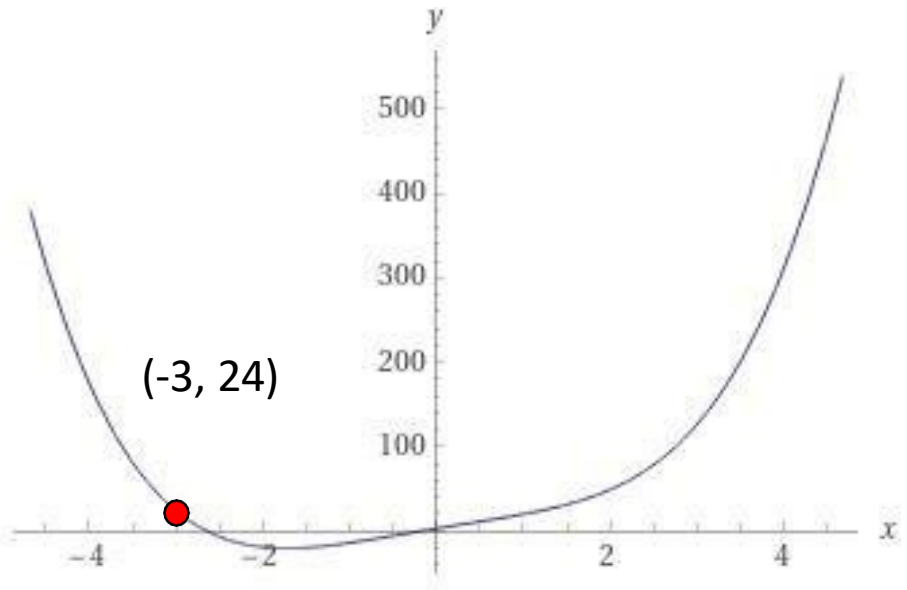
Idea: what if you just want to find something better than what you currently have?

$$f(x) = x^4 - x^2 + 17x + 3$$

$$f'(x) = 4x^3 - 2x + 17$$

$f'(-3) = -85 < 0$

so move right!



UNIVERSITY OF CALGARY

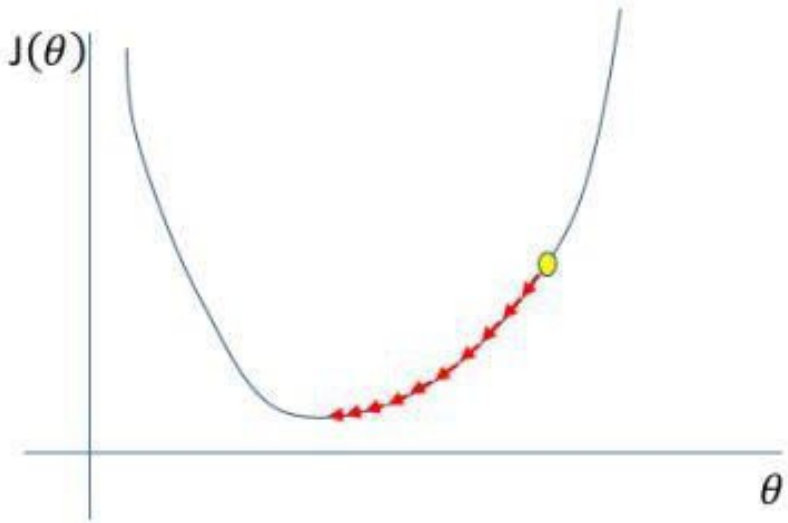
Learning rate

The step size in this approach is called the **learning rate**, η.

$$x = x - \eta * sign\left(f'(x)\right)$$ to minimize

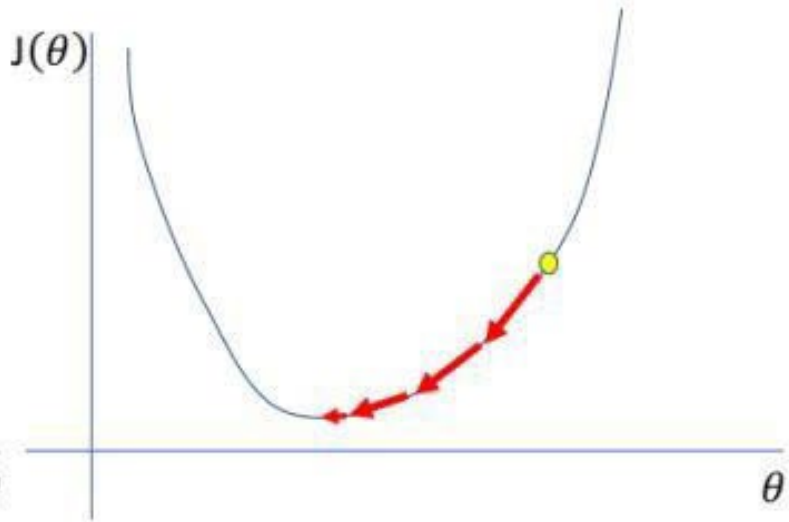
$$x = x + \eta * sign\left(f'(x)\right)$$ to maximize

Too low



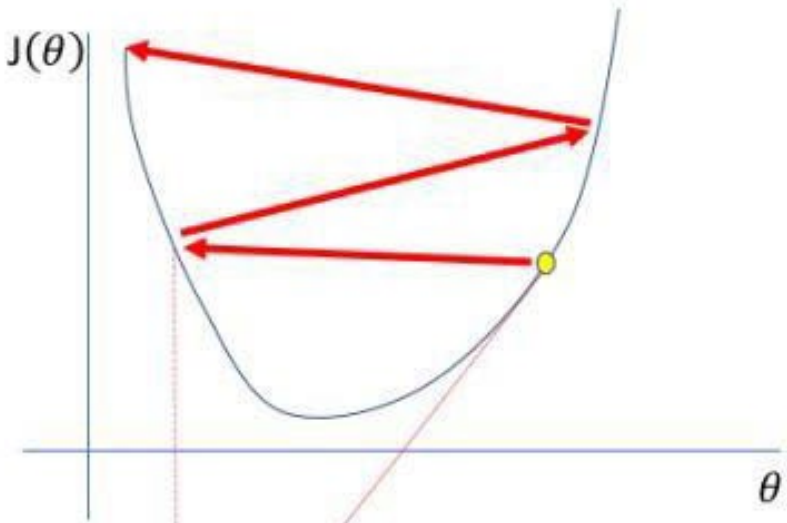
A small learning rate requires many updates before reaching the minimum point

Just right



The optimal learning rate swiftly reaches the minimum point

Too high



Too large of a learning rate causes drastic updates which lead to divergent behaviors

Gradients

Gradient

The derivative at a point in k dimensions is called the **gradient**, which is a vector that points in the direction of steepest ascent.

$$f'(x_1, x_2, \ldots x_k) = \left(\frac{\partial y}{\partial x_1}, \frac{\partial y}{\partial x_2}, \cdots \frac{\partial y}{\partial x_k}\right)$$


gradient of f

Definition

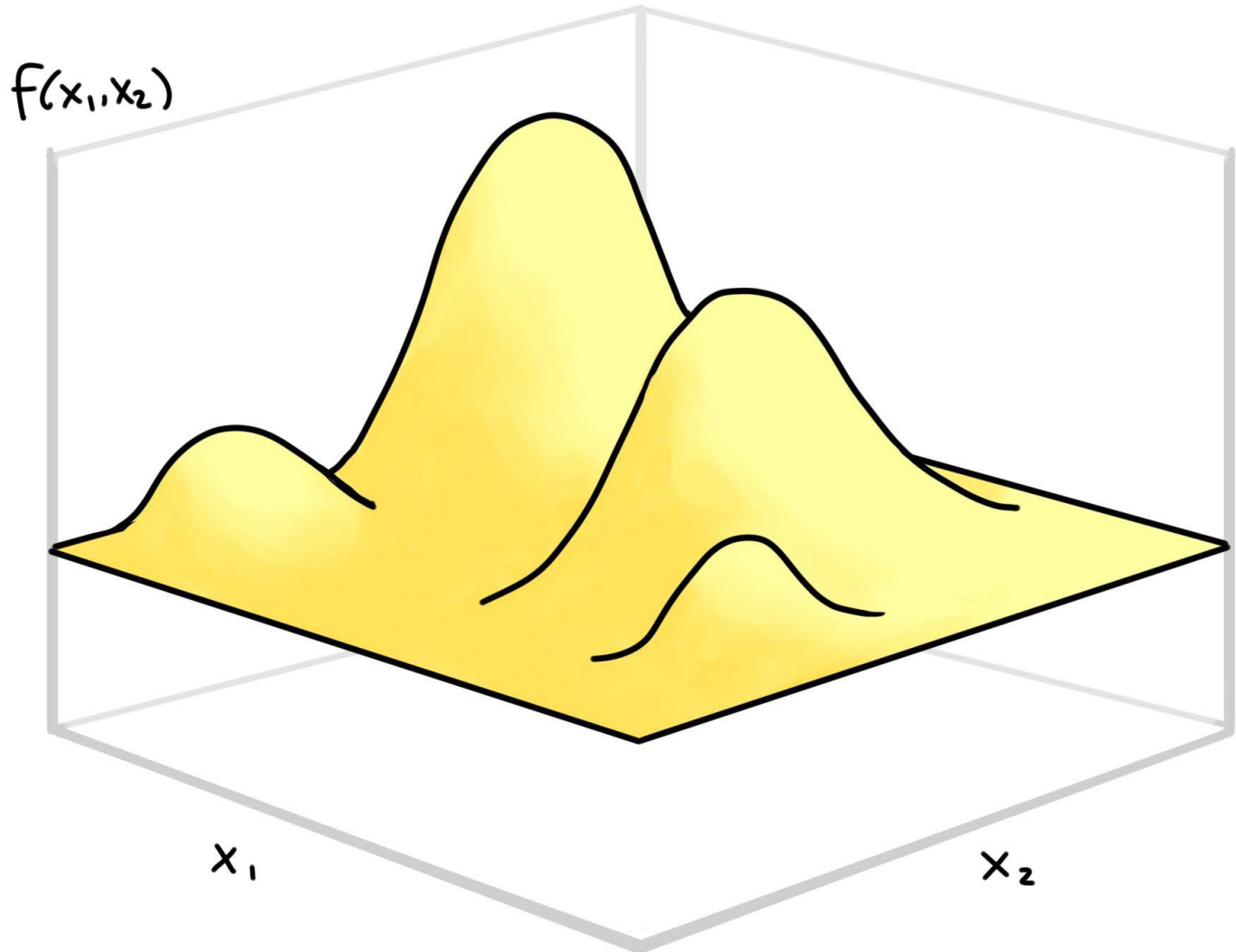
Gradient descent is an iterative algorithm for minimizing a differentiable multivariable function.

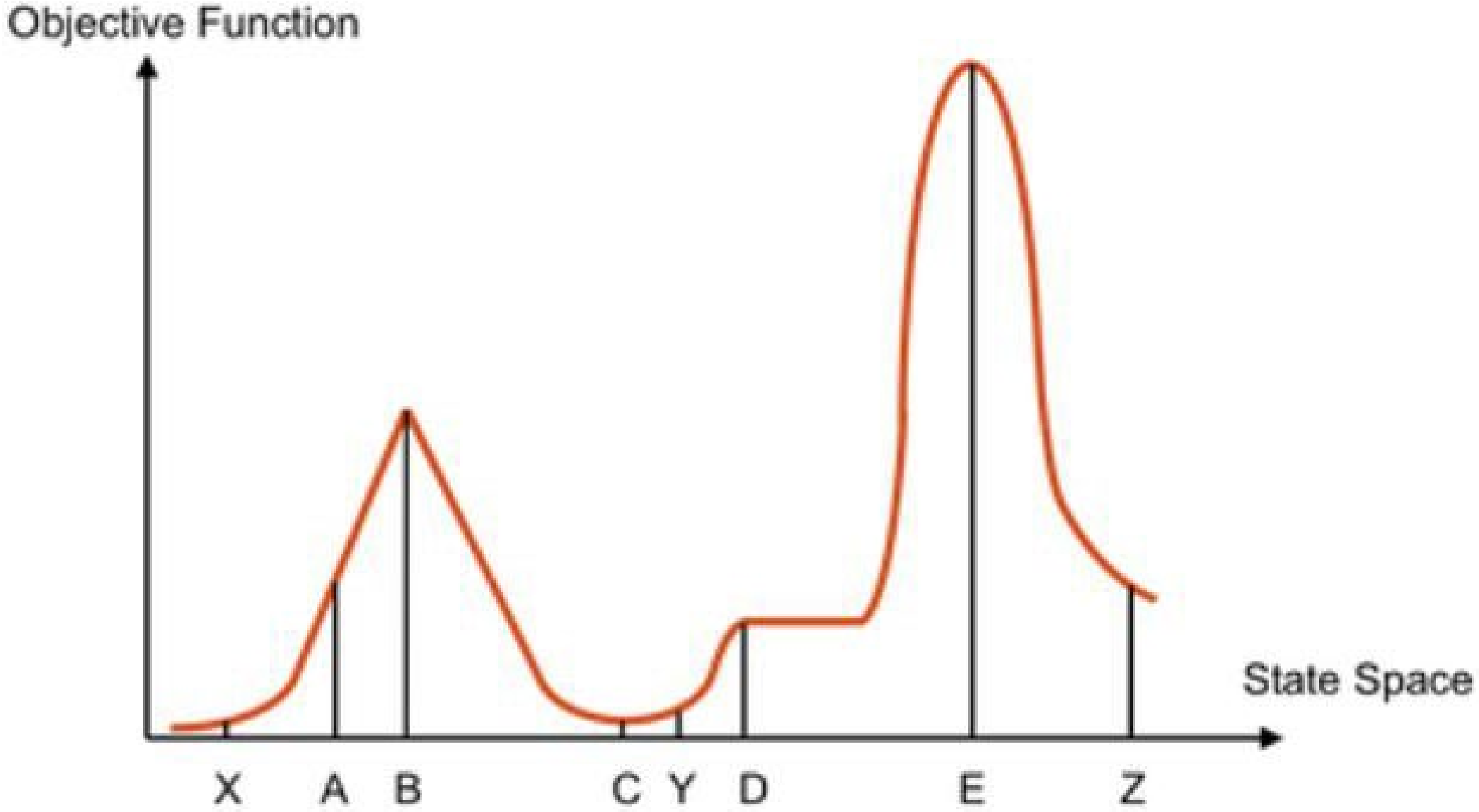
1. Pick a random starting point
2. Try to find the minimum value by taking a step of size η in the direction of the gradient
3. Repeat step 2 until you run out of steps

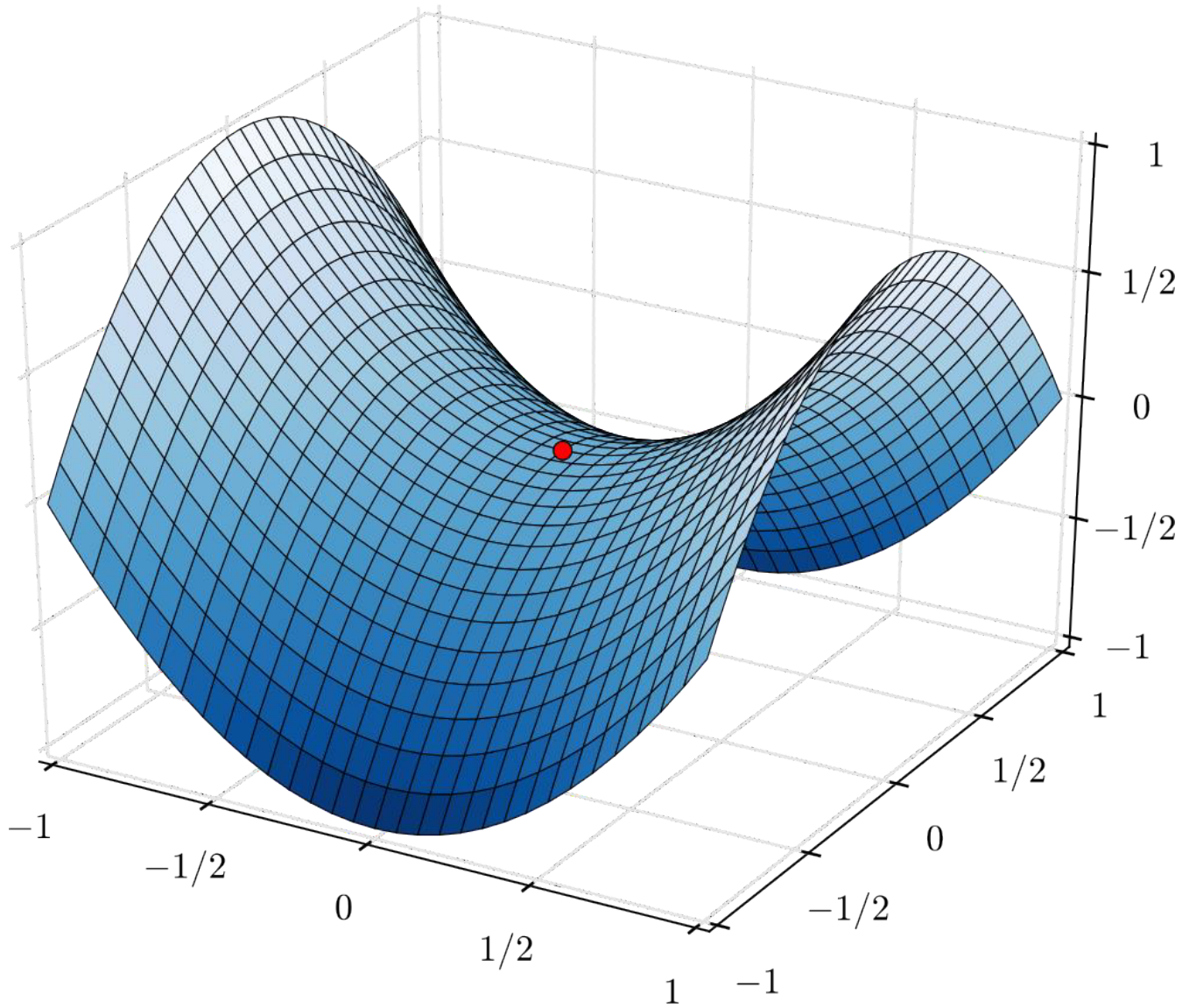
Gradient descent

Gradient descent is very sensitive to both your random starting point and the shape (topography) of the space you are exploring.

…i.e. it doesn't always work well.


f(x1,x2)
x1
x2


Objective Function
State Space
X
A
B
C
Y
D
E
Z


1
1/2
0
−1/2
−1
−1
−1/2
0
1/2
1
−1
−1/2
0
1/2
1

Note

Taking small steps against the gradient to try and find the global minimum is an example of following a **heuristic**.

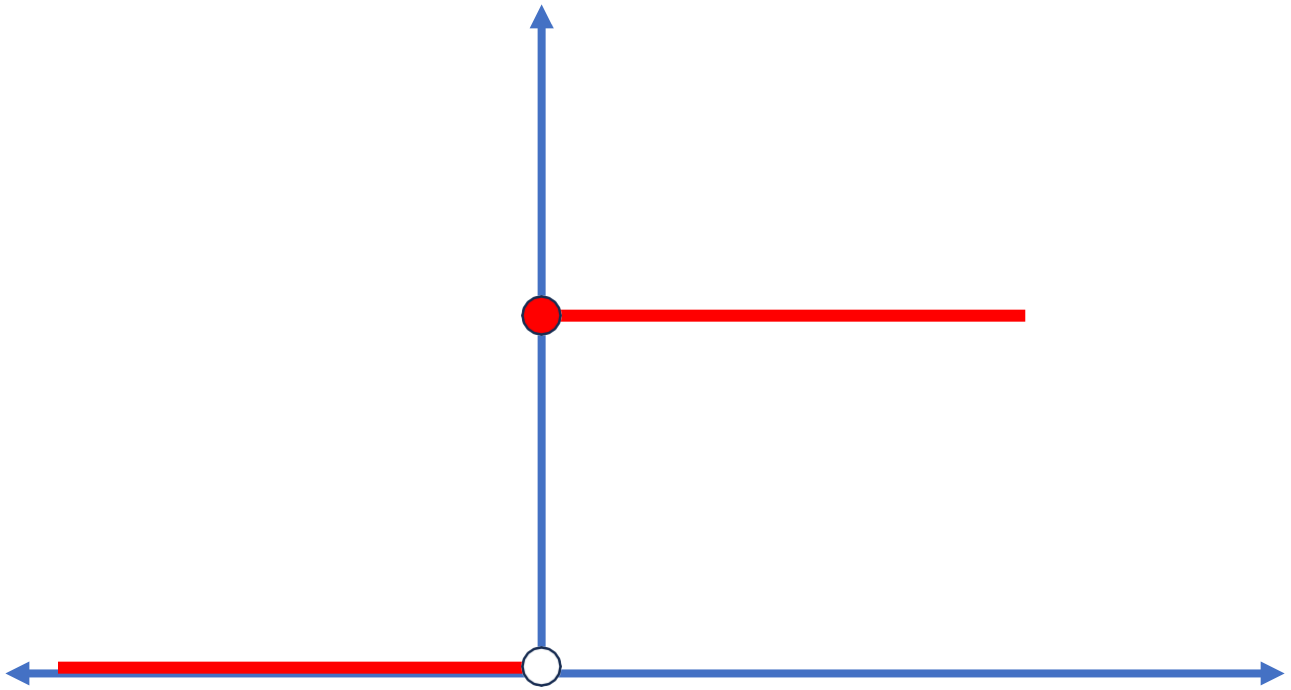
And as we saw, there are always bad cases where any heuristic makes things worse, or cases where it is simply not helpful.

Activation Functions

Activation functions

One big problem with our current activation function is that it is not **continuous** (and the derivative is either flat or DNE).

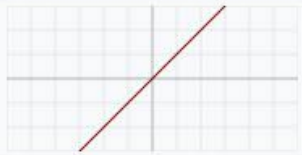
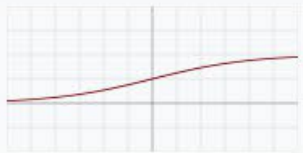
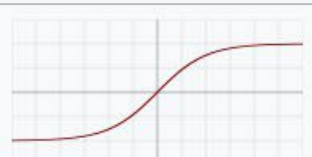
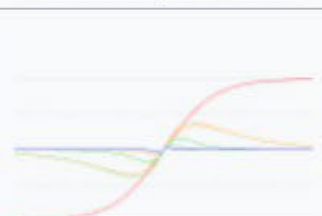
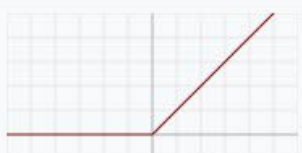
- This does not work well for our optimization!



Activation functions

To handle this, we have some alternative activation functions:

- **Rectified linear unit (ReLU):** $f(z) = \begin{cases} z & \text{if } z > 0 \\ 0 & \text{otherwise} \end{cases}$
- **Sigmoid (logistic) function**: $f(z) = \frac{1}{1+e^{-z}}$

Name	Plot	Function, $g(x)$	Derivative of g, $g'(x)$	Range	Order of continuity
Identity		x	1	$(-\infty, \infty)$	C^∞
Binary step		$\begin{cases} 0 & \text{if } x < 0 \\ 1 & \text{if } x \geq 0 \end{cases}$	0	$\{0, 1\}$	C^{-1}
Logistic, sigmoid, or soft step		$\sigma(x) \doteq \frac{1}{1 + e^{-x}}$	$g(x)(1 - g(x))$	$(0, 1)$	C^∞
Hyperbolic tangent (tanh)		$\tanh(x) \doteq \frac{e^x - e^{-x}}{e^x + e^{-x}}$	$1 - g(x)^2$	$(-1, 1)$	C^∞
Soboleva modified hyperbolic tangent (smht)		$\operatorname{smht}(x) \doteq \frac{e^{ax} - e^{-bx}}{e^{cx} + e^{-dx}}$		$(-1, 1)$	C^∞
Rectified linear unit (ReLU)[13]		$(x)^+ \doteq \begin{cases} 0 & \text{if } x \leq 0 \\ x & \text{if } x > 0 \end{cases}$ $= \max(0, x) = x\mathbf{1}_{x>0}$	$\begin{cases} 0 & \text{if } x < 0 \\ 1 & \text{if } x > 0 \end{cases}$	$[0, \infty)$	C^0

https://en.wikipedia.org/wiki/Activation_function

Activation functions

- A good activation function needs to be:
- Non-linear
- Easy to compute
- Continuous and differentiable (almost everywhere)
- Numerically stable, especially when combined with your loss function

Altogether

Neural nets

The goal of training is to find $w_1, w_2, \ldots w_n, b_1, b_2, \ldots b_m$ that minimize:

$$T(w_1, w_2, \ldots w_n, b_1, b_2, \ldots b_m) = \frac{1}{m} \sigma_{i=0}^{m} L(g_i, a_i),$$

where g_i is the guess output by our neural net for sample i.

Neural nets

The true objective function $T\left(w_1, w_2, \ldots w_n, b_1, b_2, \ldots b_m\right)$ depends simultaneously on all of the training samples.

- This is a lot of information/computation to factor in all at once

Neural nets

Instead, we usually prefer to improve $L(g_i, a_i)$ for a single random sample (x_i, y_i) at a time.

This approach is called **stochastic gradient descent**

- Can prove this is almost as good as the original

Epochs

In practice, we usually shuffle the training data and then iterate through the samples, improving the loss on each one in turn.

- Seems to be just as good as picking randomly each time

A single pass through the training data of this type is called an **epoch**.

Next…convolutional neural networks

Jonathan Hudson, Ph.D.
jwhudson@ucalgary.ca
https://cspages.ucalgary.ca/~jwhudson/

UNIVERSITY OF CALGARY