Building ML Models

Week 2: Intro to Math

Overview

- Intro to Math
- Building a Math Library

Administrivia

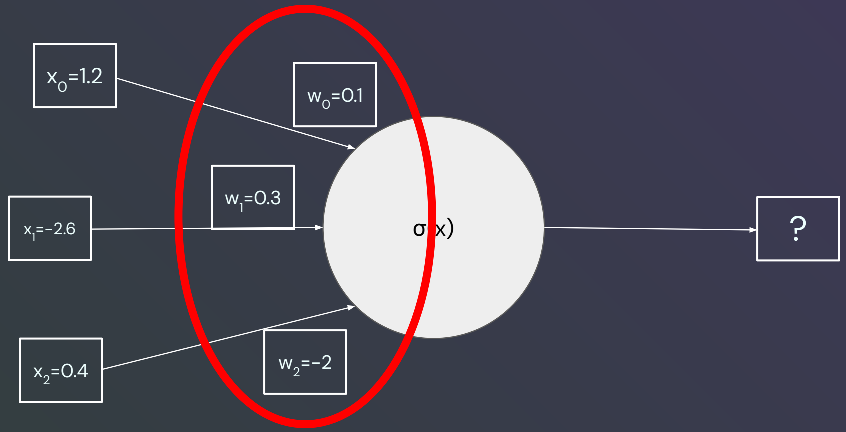
Administrivia

- Week 1 solution posted to GitHub
- Slack: we’re working on it
  - We’ve added everyone we could to the Slack, create an account with your UMich email if you haven’t already
  - If you still don’t have access, email one of us (kevincal@umich.edu, iheitman@umich.edu)
  - Hopefully next week we can fully transition to Slack

Intro to Math

Gradient Descent

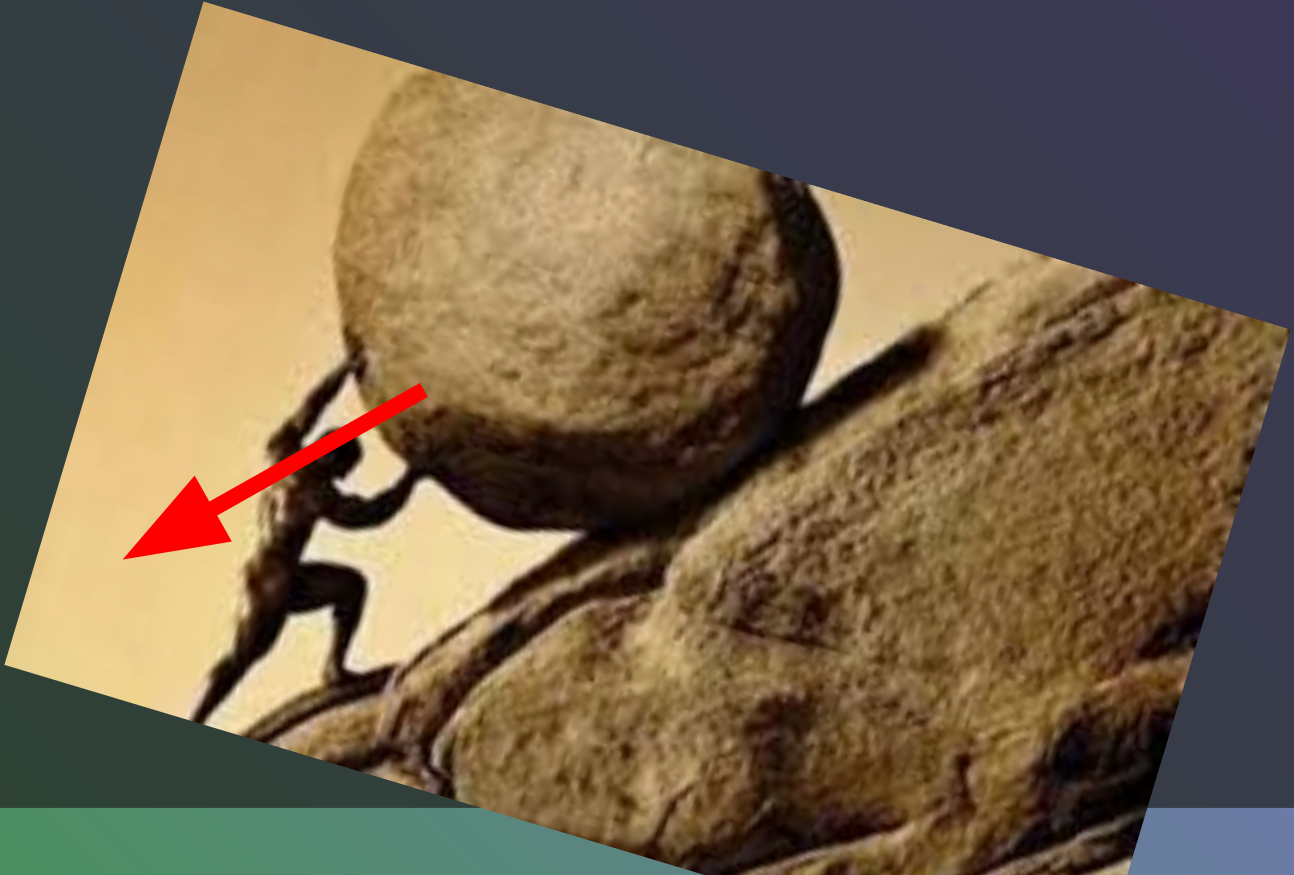
- Last week: perceptrons
- How does the perceptron “learn?”
  - **Continuously adjust weights**
  - Minimize loss (error)
- **How??**



Gradient Descent – Basics

Gradient Descent – Basics

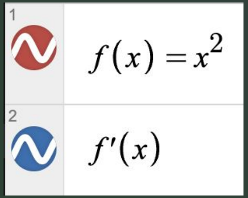
Gradient Descent – Basics

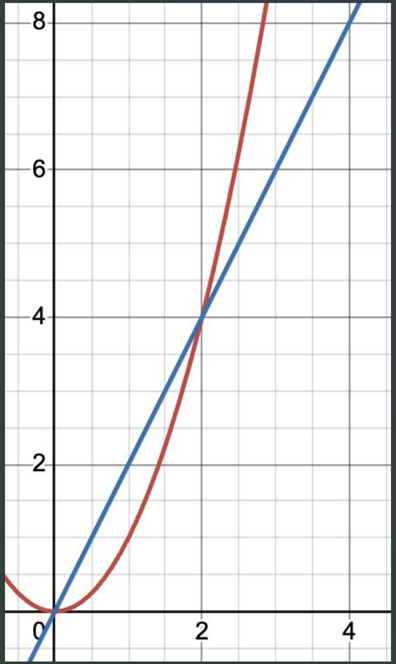


Gradient Descent – Basics

Gradient Descent – Basics

Gradient Descent – Basics

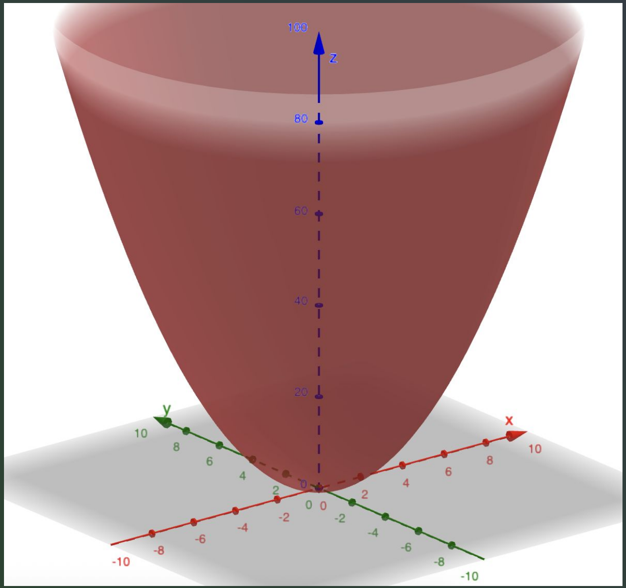




- Think back to calculus with the **derivative**
  - Instantaneous rate of change, or **slope**
  - **Steep slope = high f′(x)**
  - **Flat? f′(x) = 0**
- **Subtract slope = move closer to f′(x) = 0**

Gradient Descent – Multivariate

$f: z = x^2 + y^2$

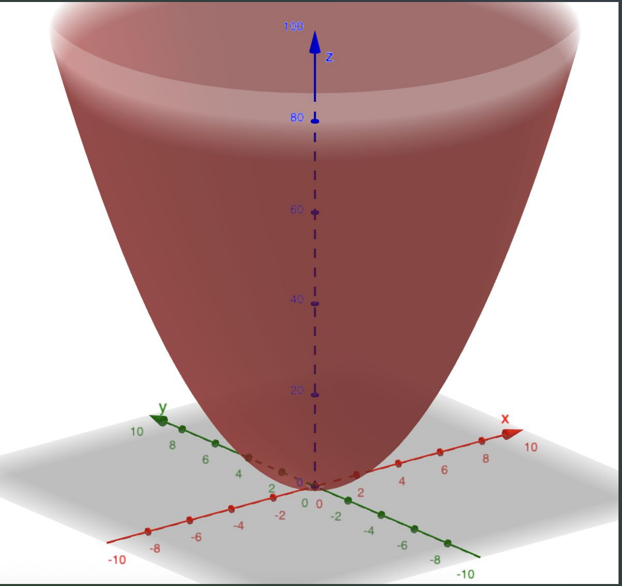


- Expand to 3 dimensions
  - Rate of change in z?
  - Hard to get with (x,y)
  - **Derivative in one dimension**
- Take the derivative **with respect** to x or y

Gradient Descent – Multi-Derivative

$$\frac{\partial f}{\partial x} = f_x, \frac{\partial f}{\partial y} = f_y$$

Gradient Descent – Multi-Derivative



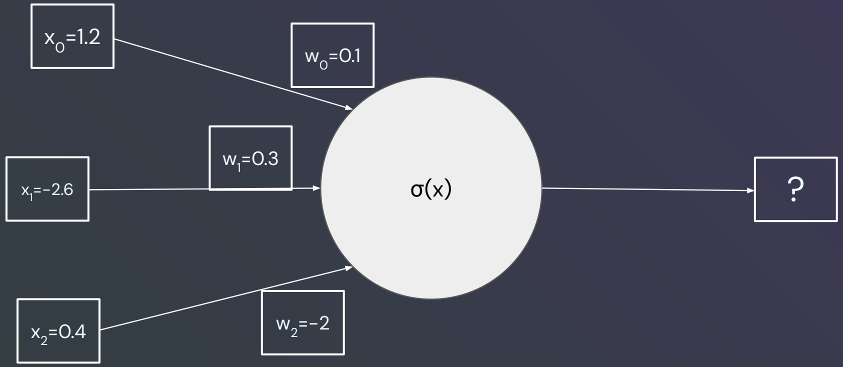
$$f(x, y) = x^2 + y^2$$

$$\frac{\partial f}{\partial x} = 2x$$

$$\frac{\partial f}{\partial y} = 2y$$

Gradient Descent – ML

- **Prediction for ANNs**
  - Train on labeled (for now) data, inputs & outputs
  - Need some way to eval fitness / **accuracy**



- **Mean Squared Error**
  - "MSE"

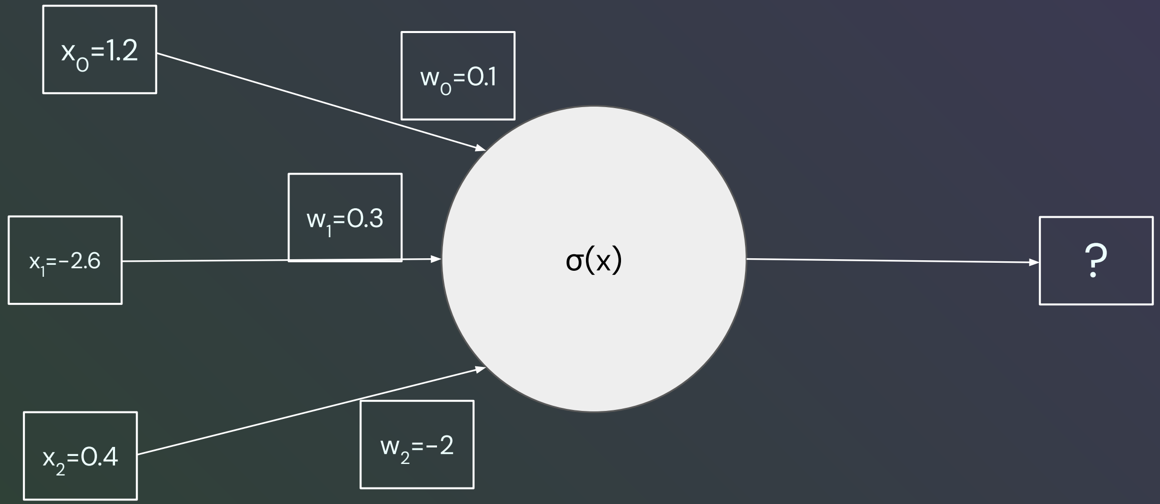
$$MSE(y, \hat{y}) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

Gradient Descent – ML

$$MSE(y, \hat{y}) = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

- **y vs y–hat**
  - y: **true** y-values in dataset, “target”
  - y-hat: **predicted** y-values
- This is a **function** of y, y–hat, components

Gradient Descent – ML



$$\hat{y} = i_1 \cdot w_i,\ \ i \in 1, 2, 3$$

$$\Delta w_i = \frac{\partial}{\partial w_i} MSE(y, \hat{y}),\ \ i \in 1, 2, 3$$

Building a Math Library

Implementation Details - Base Class

- One base class, `expression`
  - Represents a single variable, like `x`
- Other classes are derived from the `expression` base class

Implementation Details - Derived Class

- Each derived class represents some math operation (+, −, etc.)
- Implement 3 functions
- `__init__`: store the expression(s) passed into the `__init__` call
- `eval`: evaluate the result of evaluating the values passed in (usually some sort of recursive call to `eval`)
- `diff`: evaluate the result of differentiating the values passed in
- An example has been given to you in the form of the `addition` class

Let's Code It!

- GitHub Link
- Google Colab Link (make a copy of the file)