Building ML Models

Week 3: Auto Diff

Administrivia

Administrivia

- No meeting next week! (Fall Break from Studying)

Review

Review

- Created a math library
- Gradient Descent

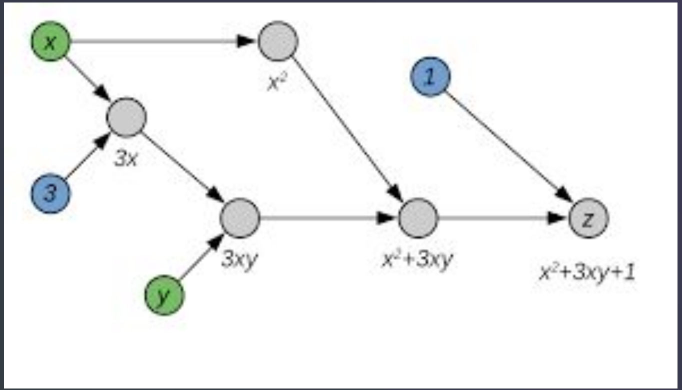
$$\frac{\partial f}{\partial x} = f_x, \frac{\partial f}{\partial y} = f_y$$

- **Taking the gradient with respect to neural network loss (!)**

Automatic Differentiation

Automatic Differentiation

- Recall:
  - Automatic differentiation: **recursive** function calls in Python = ANN function evaluation and differentiation
- **Chain rule** in calculus… what's this?



Chain rule

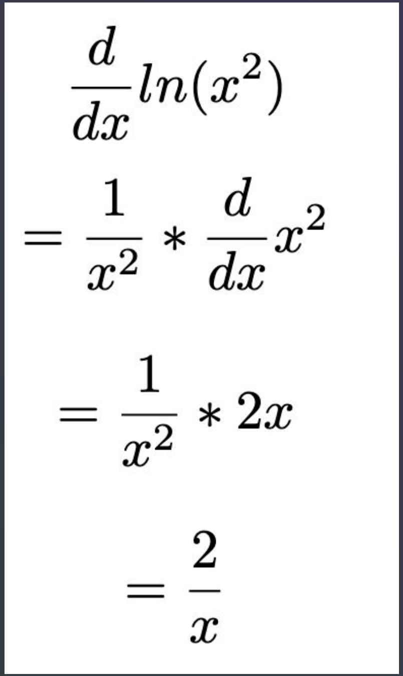
$$F'(x) = f'(g(x)) \cdot g'(x)$$

- Taking derivatives of nested function…
  - Must **also** capture derivative of inner function
- Outcome
  - Derivative of **outer** function, evaluated at **inner** function (eval x)
  - Times
  - Derivative of **inner** function, evaluated at x.

Automatic Differentiation in ANNs

- Previously, coded operations
  - Capable of recursive calls!

- **Now, let’s code functions**
  - These **must** take advantage of the chain rule.



$$\frac{d}{dx} ln(x^2)$$

$$= \frac{1}{x^2} * \frac{d}{dx} x^2$$

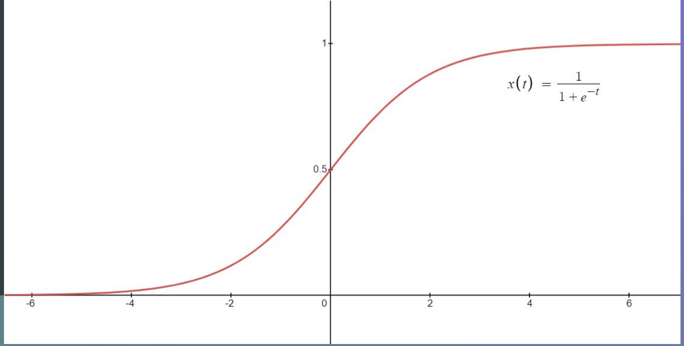
$$= \frac{1}{x^2} * 2x$$

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

Activation Functions

What is an Activation Function?

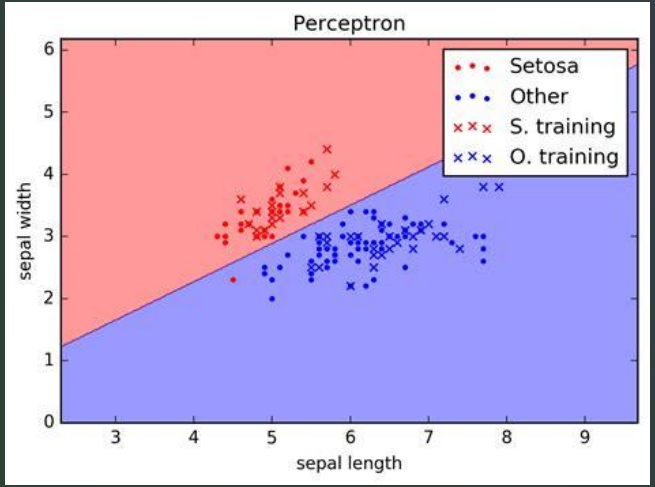
- Function we apply after applying the weights in a perceptron
- We’ve already seen one before: sigmoid

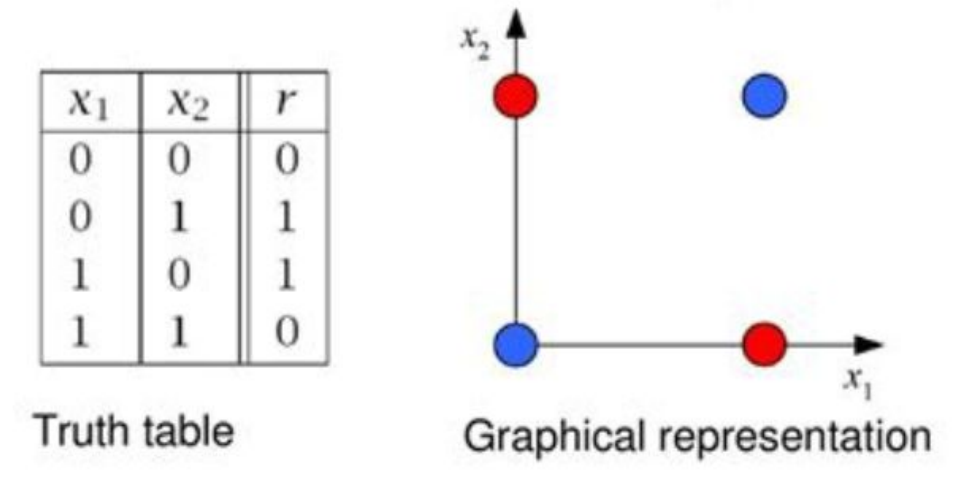


Why Do We Care?

- Introduces nonlinearity to our system
- If we didn’t have an activation function, we’d just be doing $y = x_1w_1 + x_2w_2+ \dots$ over and over again
- We can model more than just linear relationships
  - XOR Table

Linear Versus Nonlinear Classification



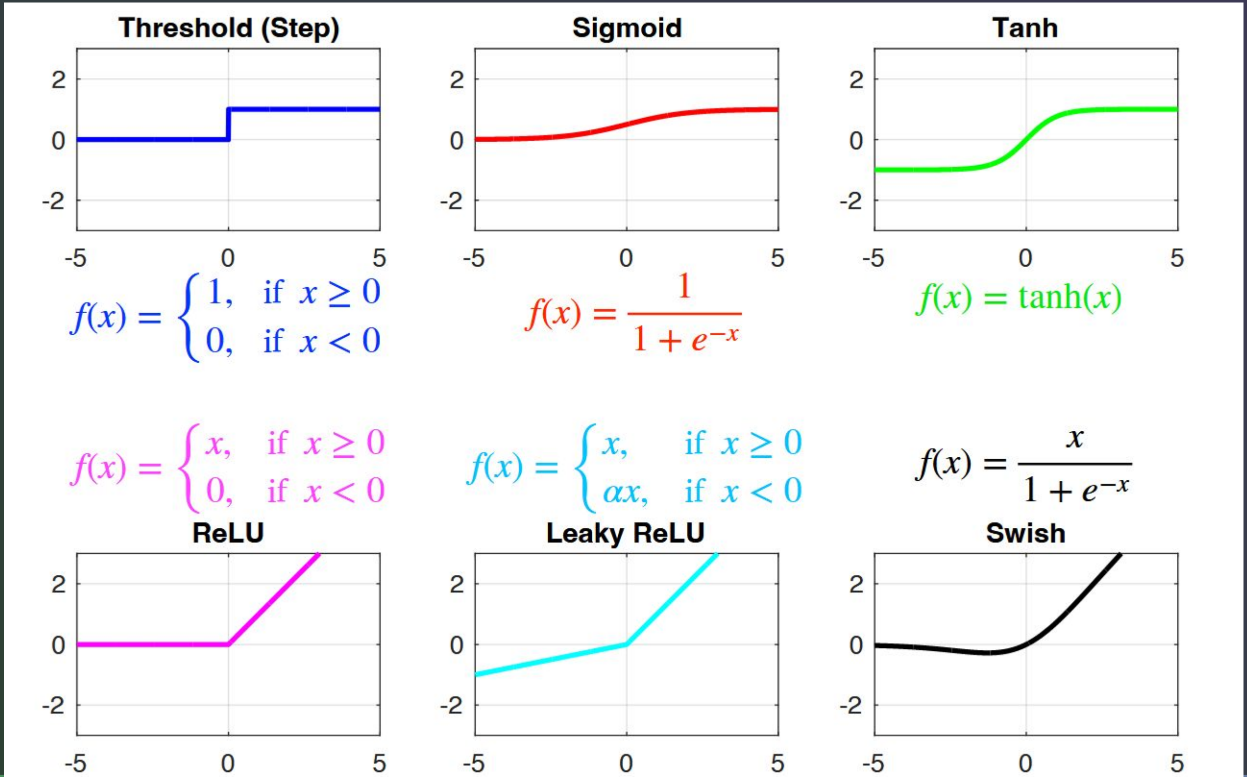


x_1	x_2	r
0	0	0
0	1	1
1	0	1
1	1	0

Truth table

Graphical representation

Common Activation Functions



Threshold (Step)

$$f(x) = \begin{cases} 1, & \text{if } x \geq 0 \\ 0, & \text{if } x < 0 \end{cases}$$

Sigmoid

$$f(x) = \frac{1}{1 + e^{-x}}$$

Tanh

$$f(x) = \tanh(x)$$

$$f(x) = \begin{cases} x, & \text{if } x \geq 0 \\ 0, & \text{if } x < 0 \end{cases}$$

ReLU

$$f(x) = \begin{cases} x, & \text{if } x \geq 0 \\ \alpha x, & \text{if } x < 0 \end{cases}$$

Leaky ReLU

$$f(x) = \frac{x}{1 + e^{-x}}$$

Swish

Let’s Code It!

Let’s Code It!

- GitHub
- Jupyter Notebook
- Notable changes from last week:
  - Provided for you: exponent, natlog.
    - ! use natlog to see an example of the chain rule in action
  - Python math overrides in expression base class