# Advanced Python for Neuroscientists
# Lecture 5: Neural network I

Summer 2022

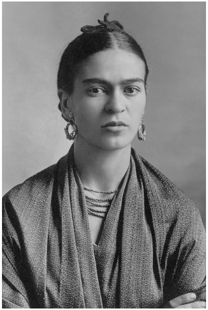
Princeton Neuroscience Institute
Instructors: Yisi Zhang & Bichan Wu

July 12, 2022

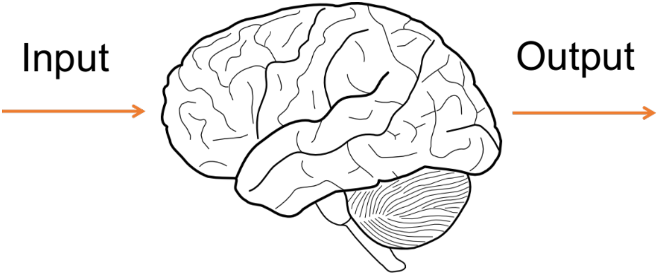
# Outline

- Motivations
- Model representation
- Gradient descent
- Word embedding

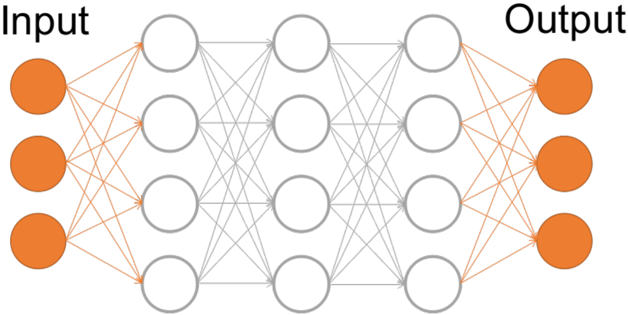
# 5.1 Motivation



# 5.1 Motivation



# 5.1 Motivation

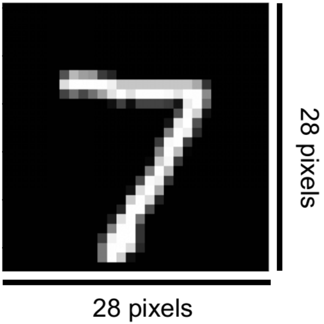


# 5.1 Motivation

How does machine recognize handwritten digits? – The MNIST database

# 5.1 Motivation

Each data is a 28x28 image

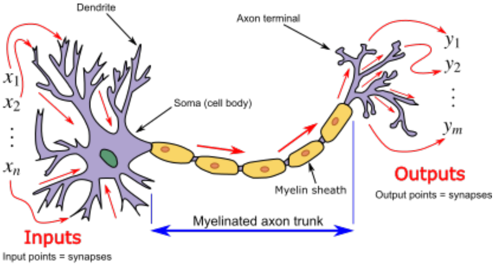



$28 \times 28 = 784$ pixels

x = [pixel 1 intensity, pixel 2 intensity, ..., pixel 784 intensity]$^T$

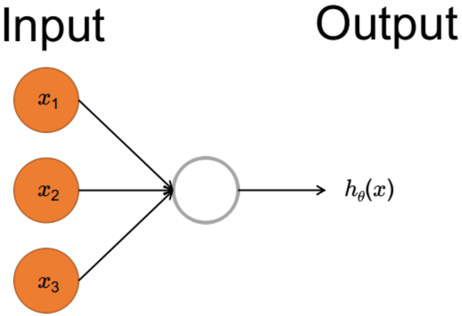
If we include quadratic terms, how many features do we get?

# 5.2 Model representation



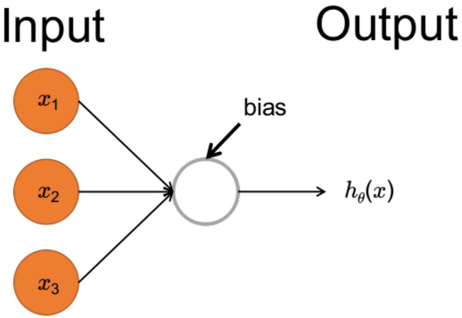
# 5.2 Model representation

The simplest model: logistic unit



# 5.2 Model representation

The simplest model: logistic unit




$$h_\theta(x) = \frac{1}{1+e^{-\theta^T x}}$$

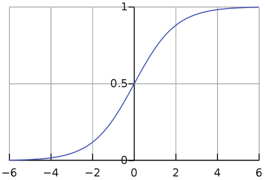
# 5.2 Model representation

The simplest model: logistic unit

$$z = \theta^T x = [\theta_0, \theta_1, \cdots, \theta_n] \begin{bmatrix} 1 \\ x_1 \\ \vdots \\ x_n \end{bmatrix} \tag{1}$$

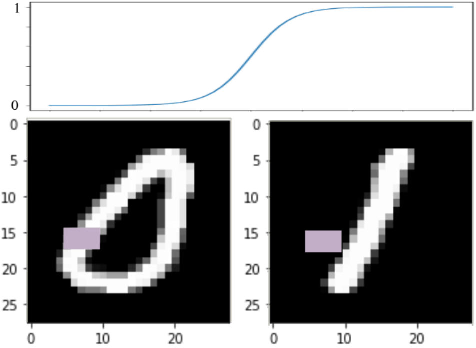
Sigmoid activation function

$g(z) = \frac{1}{1+e^{-z}}$



# 5.2 Model representation

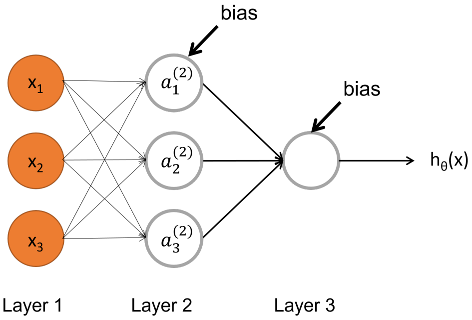
The simplest model: logistic unit




Now go to Colab exercise.

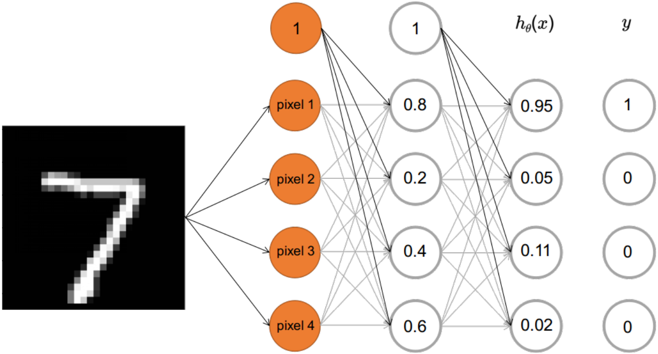
# 5.2 Model representation

## Neural network



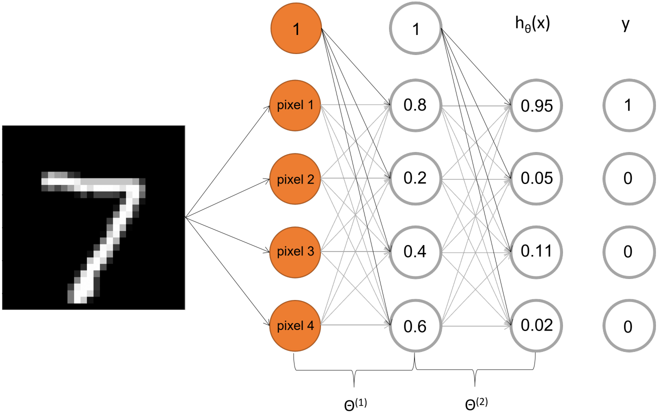
# 5.2 Model representation

## Neural network



# 5.2 Model representation

## Neural network



# 5.2 Model representation

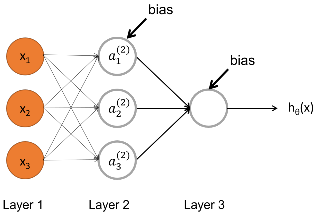
Softmax function

For multi-class neural networks, softmax function is used in the last layer:

$$\sigma(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$$

# 5.2 Model representation

Feedforward neural network




$a_i^{(j)}$ = "activation" of unit $i$ in layer $j$

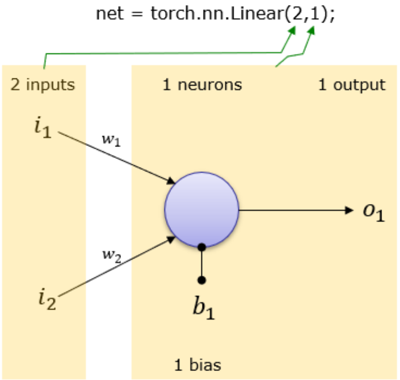
$\Theta^j$ = matrix of weights from layer $j$ to layer $j+1$

$$a_1^{(2)} = g(\Theta_{10}^{(1)} + \Theta_{11}^{(1)} x_1 + \Theta_{12}^{(1)} x_2 + \Theta_{13}^{(1)} x_3)$$
$$a_2^{(2)} = g(\Theta_{20}^{(1)} + \Theta_{21}^{(1)} x_1 + \Theta_{22}^{(1)} x_2 + \Theta_{23}^{(1)} x_3)$$
$$a_3^{(2)} = g(\Theta_{30}^{(1)} + \Theta_{31}^{(1)} x_1 + \Theta_{32}^{(1)} x_2 + \Theta_{33}^{(1)} x_3)$$

$$h_\theta(x) = a_1^{(3)} = g(\Theta_{10}^{(2)} + \Theta_{11}^{(2)} a_1^{(2)} + \Theta_{12}^{(2)} a_2^{(2)} + \Theta_{13}^{(2)} a_3^{(2)})$$

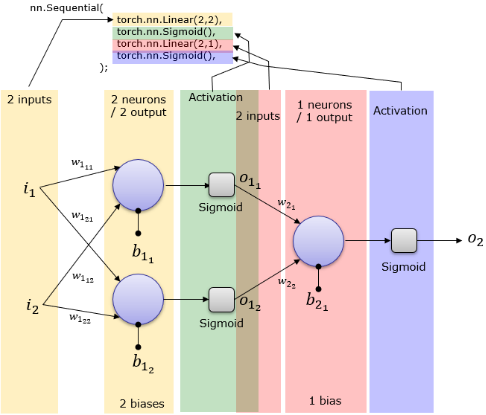
# 5.2 Model representation

Colab exercise: setup a single layer with inputs and outputs



# 5.2 Model representation

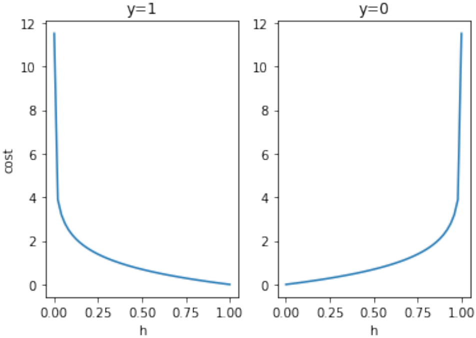
Colab exercise: setup a complete neural network with one hidden layer



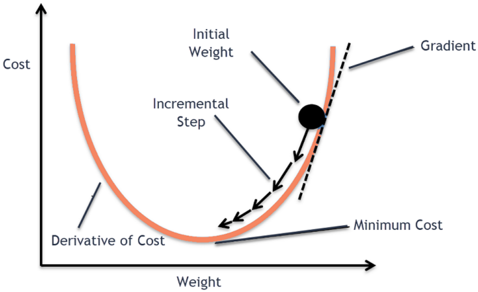
# 5.3 Gradient descent

Cost function for logistic regression:

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m} -y^{(i)}\log(h_\theta(x^{(i)})) - (1-y^{(i)})\log(1-h_\theta(x^{(i)}))$$



# 5.3 Gradient descent




$$\theta := \theta - \alpha \frac{\partial}{\partial \theta} J(\theta)$$

$\alpha$: learning rate

# 5.3 Gradient descent

Cost function for logistic regression:

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m} -y^{(i)}\log(h_\theta(x^{(i)})) - (1-y^{(i)})\log(1-h_\theta(x^{(i)}))$$

For scalar $x$ and $\theta$, $h_\theta(x) = g(\theta x) = \frac{1}{1+e^{-\theta x}}$, we have

$\frac{d}{d\theta}\log(h_\theta(x)) = x(1-g(\theta x))$, and $\frac{d}{d\theta}\log(1-h_\theta(x)) = -xg(\theta x)$

It is easy to prove that $\frac{\partial}{\partial\theta}J(\theta) = \frac{1}{m}X^T(g(X\theta)-y)$

Here $X = [x^{(1)}, \ldots, x^{(m)}]^T$ and $y = [y^{(1)}, \ldots, y^{(m)}]^T$

# 5.3 Gradient descent

Cost function for regularized logistic regression:

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m} -y^{(i)}\log(h_\theta(x^{(i)})) - (1-y^{(i)})\log(1-h_\theta(x^{(i)})) + \frac{\lambda}{2m}\sum_{j=1}^{n}\theta_j^2$$

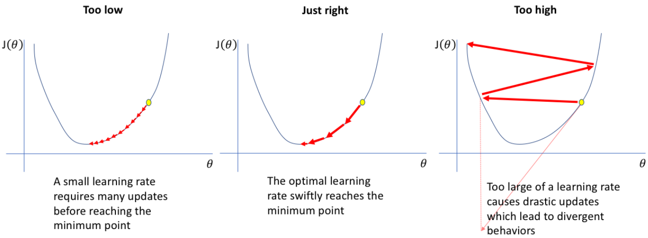
$$\frac{\partial}{\partial\theta}J(\theta) = \frac{1}{m}X^T(g(X\theta)-y) + \frac{\lambda}{m}\theta 1$$

where $\theta 1 = [0, \theta_1, ..., \theta_n]^T$

What if there are multiple outputs and hidden layers?

# 5.3 Gradient descent

## Learning rate



# 5.4 Word embedding

- Distributional semantics: "The distributional hypothesis in linguistics is derived from the semantic theory of language usage, i.e. words that are used and occur in the same contexts tend to purport similar meanings." (Wikipedia)

# 5.4 Word embedding

Example:

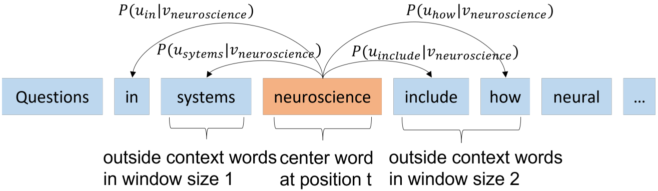
"Current neuroscience reports estimate that the human brain has over 85 million brain cells."

"The Sleep Society for Neuroscience offers insight into the scientific discoveries derived from REM studies as well as their implications."

"Williams starts university this autumn, studying neuroscience at London's UCL."

# 5.4 Word embedding

Compute $P(w_{t+j}|w_t)$



# 5.4 Word embedding

Minimize the objective function:

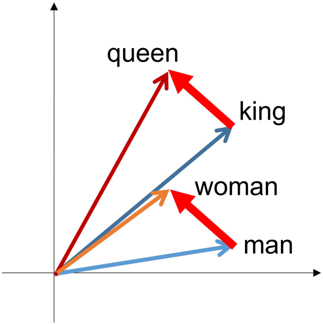
$$J(\theta) = -\frac{1}{T}\sum_{t=1}^{T}\sum_{-m\leq j\leq m} \log P(w_{t+j}|w_t;\theta)$$

Softmax function

$$P(o|c) = \frac{e^{u_o^T v_c}}{\sum_{w\in V} e^{u_w^T v_c}}$$

# 5.4 Word embedding

king + woman - man = queen



# Homework

- Make sure you understand all the exercises above
- Run through the codes here that should replicate all the figures
  https://github.com/yisiszhang/AdvancedPython/blob/main/colab/Lecture5.ipynb
- Try different neural network layouts
- Try different cost functions for gradient descent