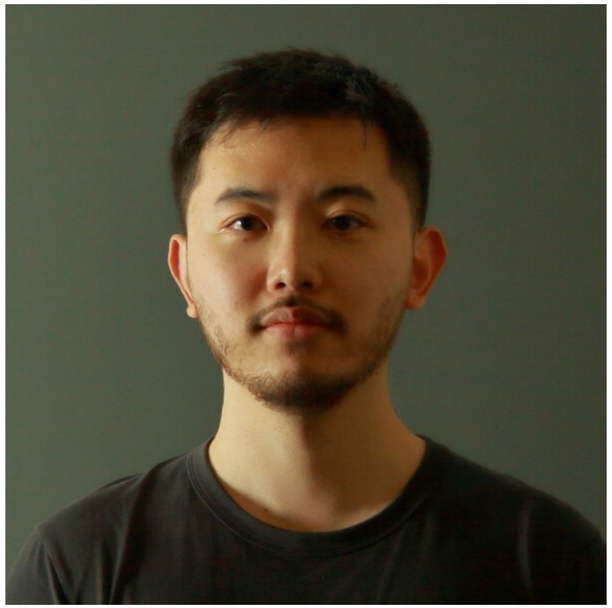
Machine Learning for Computational Linguistics

Lecture 1: Introduction to the course & Linear Algebra

Srikanth Madikeri, 20.09.2023

Course organization

Tutors



Fei Gao

Yuliia Frund

Tutorat every Tuesday between 12h15 and 13h45 (AND-3-02/06)

Course organization

Evaluation

- Portfolio: 25% exercises, 75% final exam
- 4 exercises in total. Tentative release dates
  - 4th October, 25th October, 8th November, 29th November
  - 2 weeks to submit
  - OK to do in groups of 2, but must be declared
- **WARNING**: Final exam is on 20th December, not 17th December

What is Machine Learning?

Machine learning

83 languages

Article Talk Read Edit View history Tools

From Wikipedia, the free encyclopedia

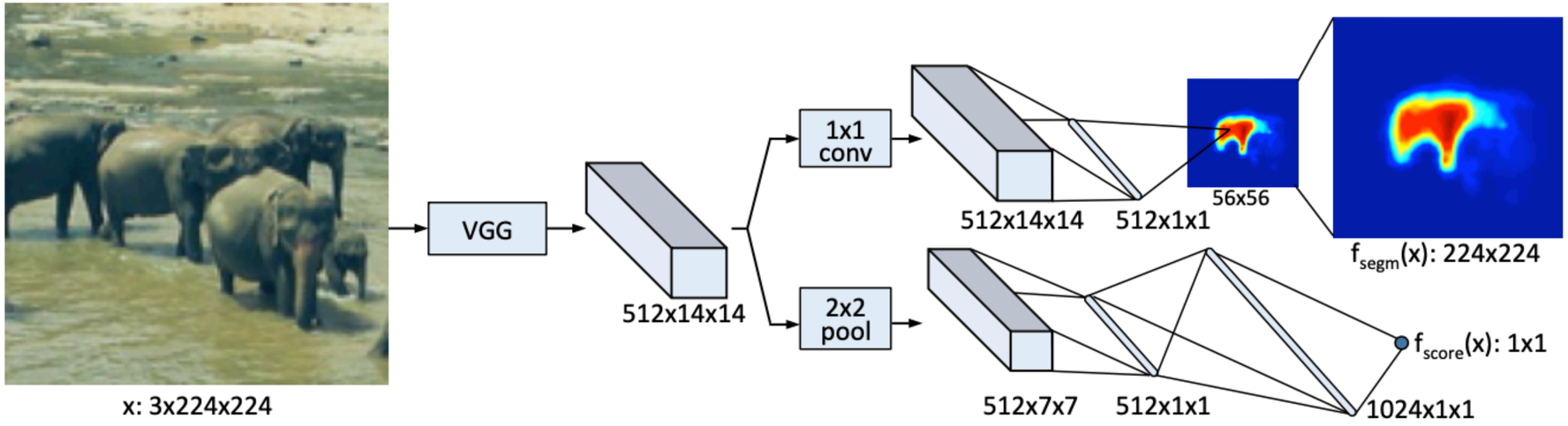
Machine learning (**ML**) is a field of study in artificial intelligence concerned with the development and study of statistical algorithms that can learn from data and generalize to unseen data and thus perform tasks without explicit instructions.[1] Recently, artificial neural networks have been able to surpass many previous approaches in performance.[2]

Keywords: statistical algorithms, data, artificial neural networks

Example: Machine Translation
Google Translate

- An example of a task with a simple definition
  - Humans find it easy to define what they want to do
  - Hard to automate
- Improvements in underlying technology: Rule based ==> Statistical machine learning ==> Deep Neural networks
  - Rule-based: You have to come up with all possible rules (and exceptions)!
    - E.g. French has too many exceptions
  - Statistical machine learning: Need to choose a different model for different task
  - Deep Neural Networks: Recent approaches converge to a unified model for many tasks

Object segmentation



Popular examples

- ChatGPT: Large Language Models with reinforcement learning
- Autopilot in cars
- Voice assistants in your phones

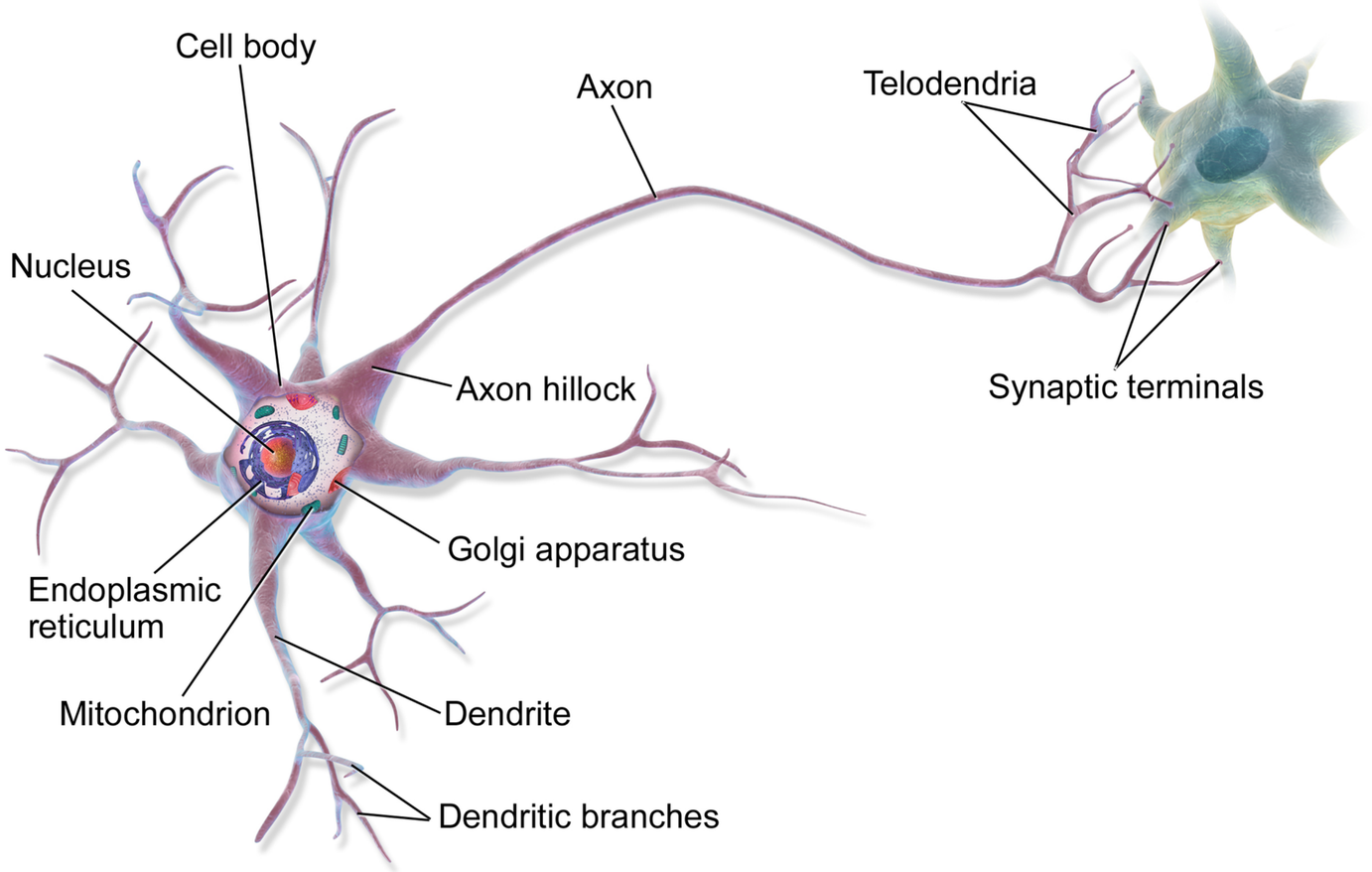
The Learning problem

- Define the task. E.g. Translate text from English to French
- Training set: consists of data (X) and targets (Y)
  - E.g. Machine Translation (En -> Fr) How are you —> Ça va?
  - X: How are you?
  - Y: Ça va?
  - Problem: Learn a mapping that transforms X to Y
- Learning: Learn the map from many examples of (X, Y)
- Inference: Given X, *predict* Y

Approaching the learning problem

- We are humans. So, how do we solve the problem?
- Formulate it in a way that we can convert it into a computer program
  - Check if a feasible solution exists

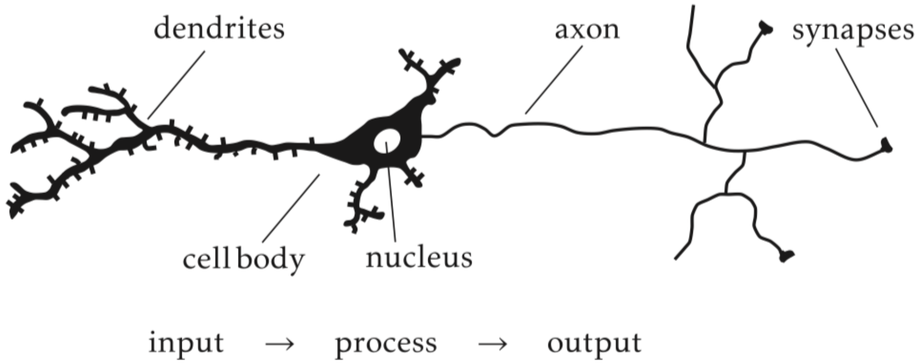
Neurons in the Human Brain



A multipolar neuron (image from Wikipedia)

- Brain contains billions of neuron cells
- They are interconnected
- Each neuron has many incoming connections and outgoing connections
- Myelination: pathways get stronger the more they are used

Spiking neurons



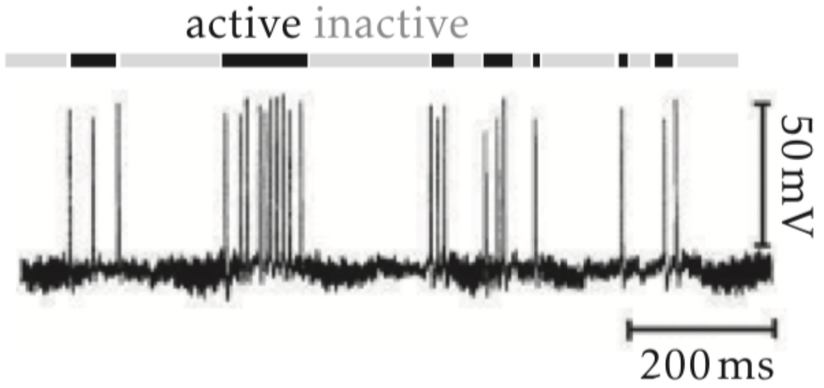
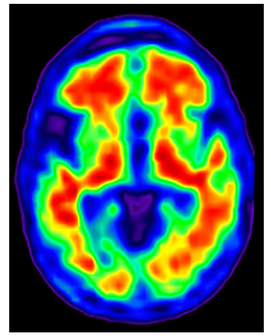


Image from Mehlig 2019

- (Over-)Simplified image of the neuron
- Viewed as a processing unit
- Information flows when it is active
  - Spike of voltage for a brief moment
  - Spike implies ***activity***
- Check chapter 1 from Mehlig's textbook for a nice introduction



A computation unit based on the neuron

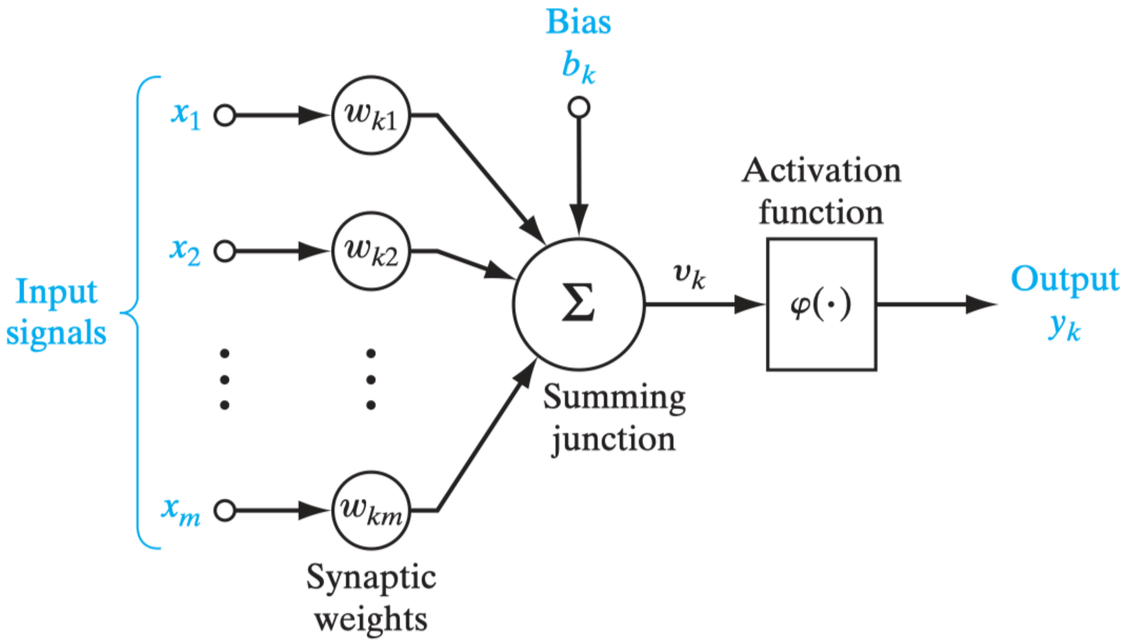
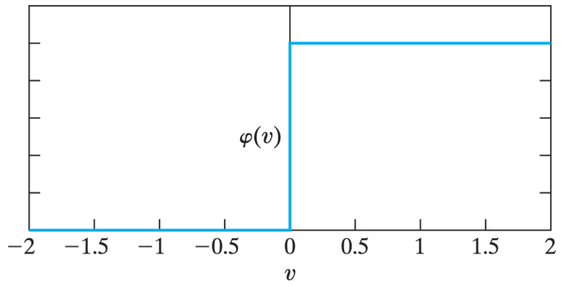


FIGURE 5 Nonlinear model of a neuron, labeled k.



Thresholding function

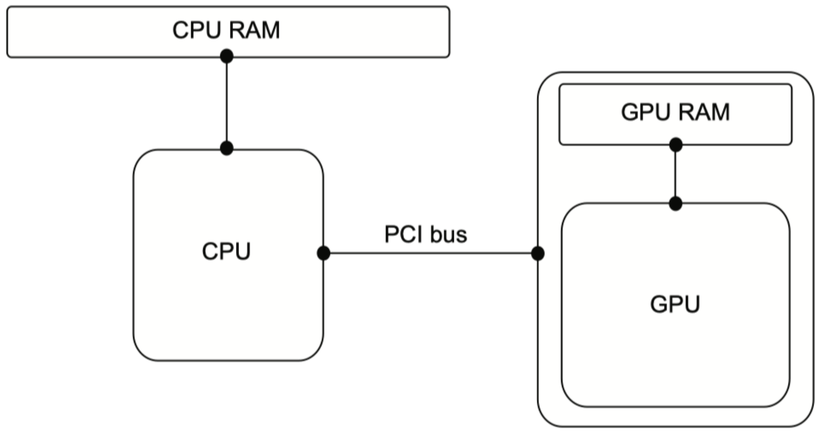
Source: Haykin 3e

Relevant material added to OLAT

AI revolution

- The foundations for the methods we use have already existed for decades
  - Mcculoh and Pitts neuron in the 60s
  - Perceptron from the 60s
  - Multilayer Perceptron (MLP) from the 80s
- What's new?
  - More data: internet has become a great source of data
  - Better hardware: GPUs!
  - Democratization of data and code: Github, Huggingface

CPUs vs GPUs



How CPU and GPU exist in the same system

CPU: Central Processing Unit

GPU: Graphic Processing Unit

GPUs are highly efficient at executing parallel code

Poor choice to execute serial code

How consumer grade GPU looks physically

Topics covered in the course

- We will cover only deep learning techniques (course title is a misnomer)
- Foundations of deep learning
- Model architectures: Transformers, RNNs, LSTMs, Encoder-decoder
- LLM, Instruction fine-tuning
- Adapters

What will not be covered

- Reinforcement learning
- CNNs (in-depth)
- Probabilistic models (e.g. GMMs, HMMs, Variational learning)

Textbooks and references

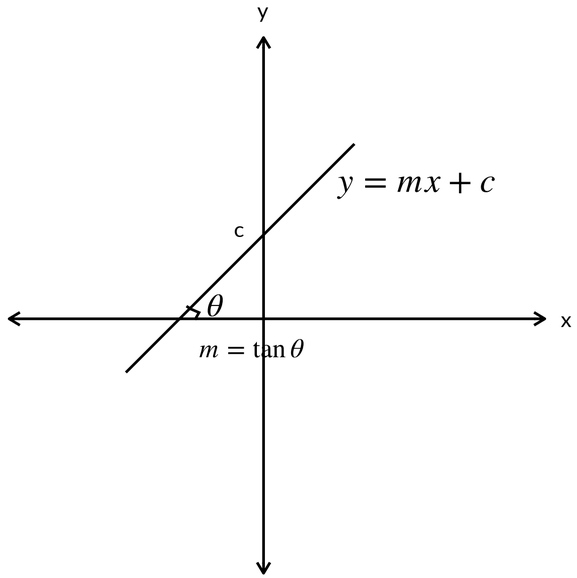
- For PyTorch programming better to follow online videos and Github repositories
- Main reference: Speech and Language processing by Jurafsky and Martin (3rd edition)
  - https://web.stanford.edu/~jurafsky/slp3/
- Other resources:
  - D2l.ai
  - Deep learning https://www.deeplearningbook.org/

Course Evaluation

- 25% exercises
  - OK to do in groups of 2, but must be declared
  - Source of materials should be declared (e.g. slide number, text book section, ChatGPT, educated guess)
- Tutorials on Tuesday 12h15 — 13h30 (AND-3-02/06)
- 75% final exam (20th December NOT 17th December)
  - Digital exam

Linear Algebra

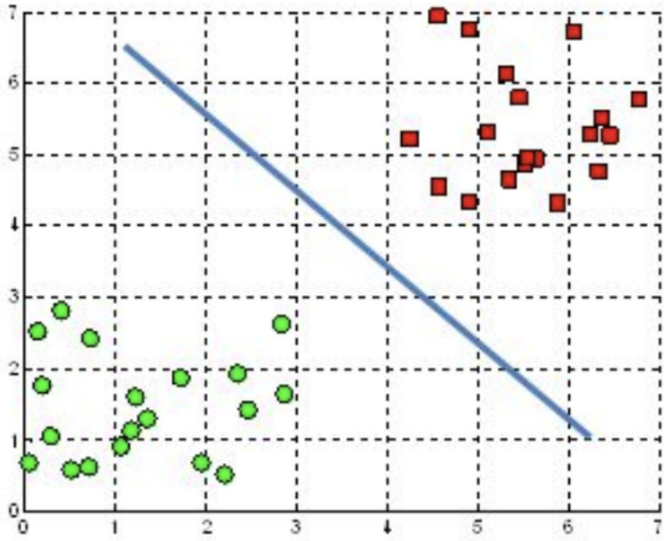
Recap



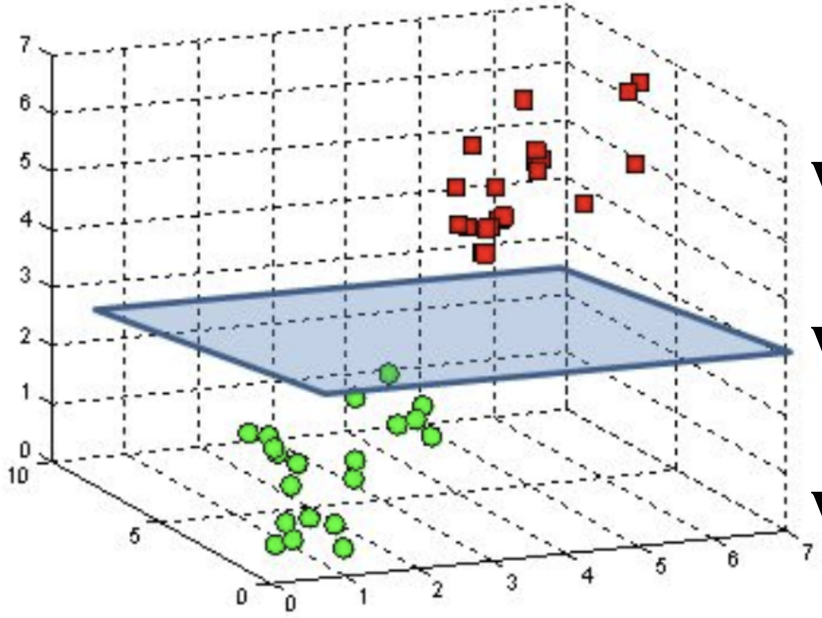
- Algebra of lines
- What is a line
  - In 1 dimension: $y = mx + c$
  - m: slope
  - c : intercept
- This is a 1 dimensional line. In 2 dimension we get a plane
- In n-dimensions we have a hyperplane

Hyperplanes

A hyperplane in $\mathbb{R}^2$ is a line



A hyperplane in $\mathbb{R}^3$ is a plane



$$\mathbf{w}^T\mathbf{x} + b > 0$$

$$\mathbf{w}^T\mathbf{x} + b = 0$$

$$\mathbf{w}^T\mathbf{x} + b < 0$$

- Equation of hyperplane: $\mathbf{w}^T\mathbf{x} + b = 0$
- Where $\mathbf{w}, \mathbf{x}$ are vectors in $n-$dimensions. This is also stated as $\mathbf{x} \in \mathbb{R}^n$
  - We use bold small letters for vectors, regular font for scalars
- T is **transpose** operation (next slide)
- b is a scalar

Dot product example

- Let us look at a 5-dimensional example

$$\mathbf{x} = \begin{bmatrix} 1.0 \\ 2.0 \\ 3.0 \\ 4.0 \\ 5.0 \end{bmatrix} \quad \mathbf{w} = \begin{bmatrix} 10.0 \\ 5.0 \\ 0.0 \\ -2.0 \\ -2.0 \end{bmatrix}$$

$$\mathbf{w}^T = [10.0 \quad 5.0 \quad 0.0 \quad -2.0 \quad -2.0]$$

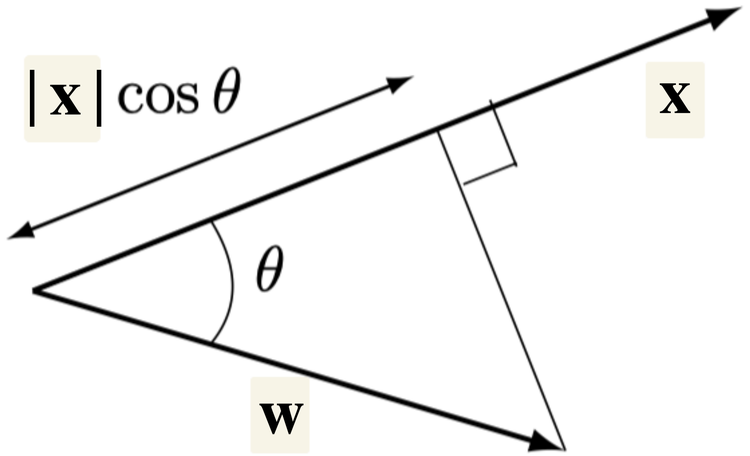
$$\mathbf{x} = \begin{bmatrix} 1.0 \\ 2.0 \\ 3.0 \\ 4.0 \\ 5.0 \end{bmatrix}$$

$$\mathbf{w}^T\mathbf{x} = 10.0 \times 1.0 + 2.0 \times 5.0 + 3.0 \times 0.0 + \ldots$$

$$\mathbf{w}^T\mathbf{x} = 10.0 + 10.0 + 0.0 + \ldots$$

$$\mathbf{w}^T\mathbf{x} = -1$$

Geometric interpretation



$$\mathbf{w}^T\mathbf{x} = |\mathbf{w}||\mathbf{x}|\cos\theta$$

- The part $\mathbf{x}$ that goes in the direction of $\mathbf{w}$ (or vice versa!) i.e. **projection** of $\mathbf{x}$ onto $\mathbf{w}$
- This is why it can be used to check similarity between two vectors!
- Two vectors are **orthogonal** if their dot product is 0

Vectors (contd.)

- Vectors are sequences of scalars
- dimensions of a vector = number of vector elements
- Convention is to write vector as a column a.k.a column vector

$$\mathbf{w} = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \\ w_5 \end{bmatrix} \quad \mathbf{w}^T = [w_1 \quad w_2 \quad w_3 \quad w_4 \quad w_5] \quad \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix}$$

$$\mathbf{w}^T\mathbf{x} = \begin{matrix} w_1 & w_2 & w_3 & w_4 & w_5 \\ \times & \times & \times & \times & \times \\ x_1 & x_2 & x_3 & x_4 & x_5 \end{matrix}$$

$$\mathbf{w}^T\mathbf{x} = w_1x_1 + w_2x_2 + w_3x_3 + w_4x_4 + w_5x_5$$

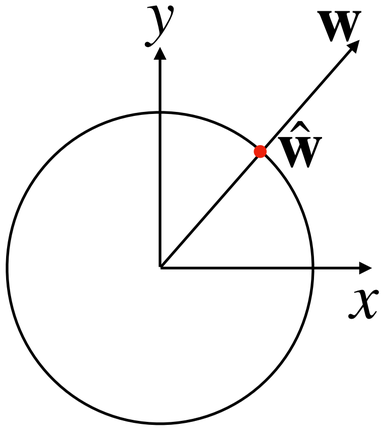
$$dim(\mathbf{w}) = 5$$

$$dim(\mathbf{w}^T\mathbf{x}) = 1$$

Magnitude and direction of vectors

- A vector can be decomposed into two components
- Magnitude: length of the vector
- Direction of the vector. It has magnitude 1.
  - Obtained by dividing the vector by its magnitude: a.k.a length normalization

$$\mathbf{w} = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \\ w_5 \end{bmatrix} \qquad |\mathbf{w}| = \sqrt{w_1^2 + w_2^2 + \ldots w_5^2} \qquad \hat{\mathbf{w}} = \begin{bmatrix} w_1/|\mathbf{w}| \\ w_2/|\mathbf{w}| \\ w_3/|\mathbf{w}| \\ w_4/|\mathbf{w}| \\ w_5/|\mathbf{w}| \end{bmatrix}$$



Special vectors

$$\mathbf{0} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \end{bmatrix}$$

Zero vector

$$\mathbf{1} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ \vdots \end{bmatrix}$$

One vector

$$\mathbf{e}_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad \mathbf{e}_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad \mathbf{e}_i = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 1 \\ \vdots\ 0 \end{bmatrix}$$

Standard basis vector

Foundation for word embeddings

Vector operations

Scalar multiplication

$$\mathbf{x} = \begin{bmatrix} 1.0 \\ 2.0 \\ 3.0 \\ 4.0 \\ 5.0 \end{bmatrix} \qquad \alpha\ \mathbf{x} = \begin{bmatrix} \alpha 1.0 \\ \alpha 2.0 \\ \alpha 3.0 \\ \alpha 4.0 \\ \alpha 5.0 \end{bmatrix} \qquad 10\mathbf{x} = \begin{bmatrix} 10.0 \\ 20.0 \\ 30.0 \\ 40.0 \\ 50.0 \end{bmatrix}$$

α is a scalar

Addition of two vectors

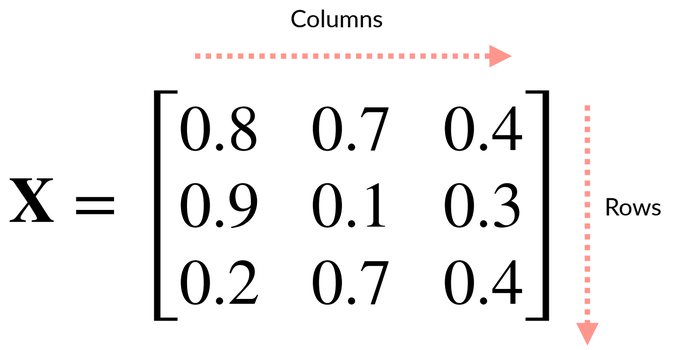
is element-wise addition

$$\mathbf{y} = \begin{bmatrix} 1.5 \\ 2.2 \\ 3.3 \\ 2.0 \\ -5.0 \end{bmatrix} \qquad \mathbf{x} + \mathbf{y} = \begin{bmatrix} 1.0 + 1.5 \\ 2.0 + 2.2 \\ 3.0 + 3.3 \\ 4.0 + 2.0 \\ 5.0 - 5.0 \end{bmatrix} = \begin{bmatrix} 2.5 \\ 4.2 \\ 6.3 \\ 6.0 \\ 0.0 \end{bmatrix}$$

Matrices

C

- Two-dimensional arrangement of values
  - First dimension is row, number of rows denoted by m
  - Second dimension is column, number of columns denoted by n
- We use capital bold letters to denote matrices
  - Sometimes it is helpful to subscript the shape of the matrix



$$\mathbf{X} = \begin{bmatrix} 0.8 & 0.7 & 0.4 \\ 0.9 & 0.1 & 0.3 \\ 0.2 & 0.7 & 0.4 \end{bmatrix}$$

$$\mathbf{X} \in \mathbb{R}^{3,3} \qquad 3 \times 3 \text{ matrix}$$

Generally we say $\mathbf{X} \in \mathbb{R}^{m,n}$

$$\mathbf{X} = \begin{bmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \\ x_{31} & x_{32} & x_{33} \end{bmatrix}$$

Matrix with scalar variables

- When n=1, we get back column vector
- When m=1, we get row vector

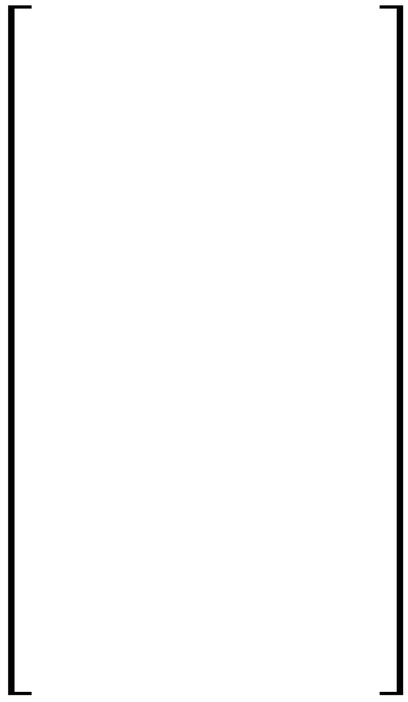
Matrix shapes

$$\begin{bmatrix} 0.8 & 0.7 & 0.4 \\ 0.9 & 0.1 & 0.3 \\ 0.2 & 0.7 & 0.4 \end{bmatrix}$$

Square $m = n$

$$\begin{bmatrix} 0.8 & 0.7 & 0.4 & -0.2 & -0.1 \\ 0.9 & 0.1 & 0.3 & -0.6 & 0.6 \\ 0.2 & 0.7 & 0.4 & 0.9 & 0.2 \end{bmatrix}$$

Rectangular $m \neq n$



Tall & skinny

$m \gg n$

Short & fat $m \ll n$

Matrix Operations

Addition of two matrices

$$\begin{bmatrix} 0.8 & 0.7 & 0.4 \\ 0.9 & 0.1 & 0.3 \\ 0.2 & 0.7 & 0.4 \end{bmatrix} + \begin{bmatrix} 0.2 & -0.7 & 0.6 \\ 0.1 & 0.1 & 0.3 \\ -0.2 & 0.3 & 0.4 \end{bmatrix}$$

$$= \begin{bmatrix} 0.8+0.2 & 0.7-0.7 & 0.4+0.6 \\ 0.9+0.1 & 0.1+0.1 & 0.3+0.3 \\ 0.2-0.2 & 0.7+0.3 & 0.4+0.4 \end{bmatrix} = \begin{bmatrix} 1.0 & 0.0 & 1.0 \\ 1.0 & 0.2 & 0.6 \\ 0.0 & 1.0 & 0.8 \end{bmatrix}$$

Both matrices need to be of same dimensions

Matrix operations

Scalar multiplication

$$2 \times \begin{bmatrix} 0.8 & -0.7 & 0.4 \\ -0.9 & 0.1 & 0.3 \\ 0.2 & 0.7 & 0.4 \end{bmatrix} = \begin{bmatrix} 1.6 & -1.4 & 0.8 \\ -1.8 & 0.2 & 0.6 \\ 0.4 & 1.4 & 0.8 \end{bmatrix}$$

Matrix operations

Trace and Transpose

Trace $tr(\mathbf{A})$

Sum of diagonal elements

$$\begin{bmatrix} 0.8 & 0.7 & 0.4 \\ 0.9 & 0.1 & 0.3 \\ 0.2 & 0.7 & 0.4 \end{bmatrix}$$

$$0.8 + 0.1 + 0.4$$

$$\begin{bmatrix} 0.8 & 0.7 & 0.4 & -0.2 & -0.1 \\ 0.9 & 0.1 & 0.3 & -0.6 & 0.6 \\ 0.2 & 0.7 & 0.4 & 0.9 & 0.2 \end{bmatrix}$$

$$0.8 + 0.1 + 0.4$$

Transpose $\mathbf{A}^T$

Swap rows and columns

$$\begin{bmatrix} 0.8 & 0.9 & 0.2 \\ 0.7 & 0.1 & 0.7 \\ 0.4 & 0.3 & 0.4 \end{bmatrix}$$

$$\begin{bmatrix} 0.8 & 0.9 & 0.2 \\ 0.7 & 0.1 & 0.7 \\ 0.4 & 0.3 & 0.4 \\ \vdots & \vdots & \vdots \end{bmatrix}$$

Symmetric matrix: $\mathbf{A} = \mathbf{A}^T$

$$\begin{bmatrix} 0.8 & 0.9 & 0.2 \\ 0.9 & 0.1 & 0.7 \\ 0.2 & 0.7 & 0.4 \end{bmatrix}$$

Linear independence

A set of vectors are linearly dependent if they can be scaled and added to obtain 0.

Otherwise, they are linearly independent

(1,0,0) , (0,1,0) and (0,0,1) linearly independent.

(1,0,1) , (2,-1,1) , (3,-1,2) are linearly dependent.

Matrix rank

- Consider the row (or column) vectors in a matrix
- The rank of the matrix is the size of the largest set of linearly independent vectors

$$\mathbf{X} = \begin{bmatrix} 1 & 2 & 3 \\ 0 & 2 & 2 \\ 1 & 4 & 5 \end{bmatrix}$$

- Rank of the above matrix is 2
- Only row 1 and 2 are linearly independent

Matrix multiplication

- Let $\mathbf{X}_1$ be a $m \times n$ matrix
- Let $\mathbf{X}_2$ be $a \times b$ matrix
- $\mathbf{X}_1$ can be multiplied with $\mathbf{X}_2$ only if $a = n$
  - The resulting matrix $\mathbf{X}_1\mathbf{X}_2$ is $m \times b$

$$\mathbf{X}_1 = \begin{bmatrix} 2 & 4 \\ 3 & 2 \\ 1 & 2 \end{bmatrix} \quad \mathbf{X}_2 = \begin{bmatrix} 3 & 2 \\ 1 & 1 \end{bmatrix} \quad \mathbf{X}_1\mathbf{X}_2 = \begin{bmatrix} 2 & 4 \\ 3 & 2 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} 3 & 2 \\ 1 & 1 \end{bmatrix} =$$

Special matrices

Diagonal matrix

$$\mathbf{D} = \begin{bmatrix} 0.8 & 0.0 & 0.0 \\ 0.0 & 0.1 & 0.0 \\ 0.0 & 0.0 & 0.4 \end{bmatrix}$$

When a matrix is pre-multiplied by a diagonal matrix, i.e $\mathbf{DX}$ it scales the rows

When it is post-multiplied, that is $\mathbf{XD}$ it scales the columns

Identity matrix

$$\mathbf{I}_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

3×3 identity matrix

$$\mathbf{I}_n = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \dots & 0 \\ 0 & \vdots & \vdots & \dots & ? \end{bmatrix}$$

$n \times n$ identity matrix

Any matrix multiplied with $\mathbf{I}$ is the matrix itself

$$\mathbf{AI} = \mathbf{IA} = \mathbf{A}$$

Matrix identities

- Distributive law $\mathbf{A}(\mathbf{B}+\mathbf{C}) = \mathbf{AB} + \mathbf{AC}$
- Associative law $\mathbf{A} + (\mathbf{B} + \mathbf{C}) = (\mathbf{A} + \mathbf{B}) + \mathbf{C}$
- Transpose after product: $(\mathbf{AB})^T = \mathbf{B}^T\mathbf{A}^T$

For more fun with matrix identities see “Matrix cookbook”

Matrix inverse

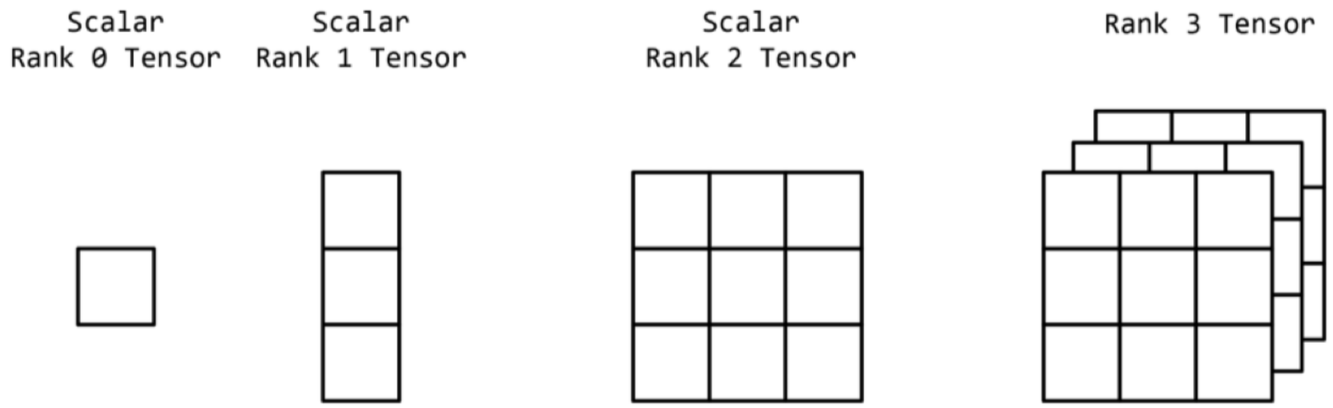
$$\mathbf{A}\mathbf{A}^{-1} = \mathbf{A}^{-1}\mathbf{A} = \mathbf{I}$$

- Only square matrices can be inverted
- Only full-rank matrices can be inverted

Tensors

A stack of matrices

- Multidimensional arrangement of numbers



- PyTorch uses tensors as a fundamental representation of data

Linear algebra topics covered in the next class

- Geometric Matrix-vector product
- Eigen values
- Singular Value Decomposition