CS 1674: Intro to Computer Vision

Introduction

Prof. Seong Jae Hwang
University of Pittsburgh
January 19, 2021
Course Info

• Course website: http://pitt.edu/~sjh95/cs1674_s21/
• Instructor: Seong Jae Hwang (sjh95@pitt.edu)
• Office: https://pitt.zoom.us/my/seongjae
• Class: Tue/Thu, 1:15pm-2:30pm
• Office hours: TBD
• Hangout room: (ID on Canvas)
About the Instructor

1989: Born in South Korea

2011: BS in CS at University of Illinois-Urbana Champaign

2013: MS in Robotics at University of Pennsylvania

2019: PhD in Robotics at University of Pennsylvania
About the Instructor

2010: Motorola, IL
Android App Development

2011: Yahoo!, IL
Hadoop Scheduling Analysis

2014: Zepp Labs, CA
Computer Vision Scientist
(golf club head trajectory tracking)

2018, 2019: Google Research, CA
Research Intern (8M YouTube Video classification, Automated ML Pipeline)
About the TA

• **TA (grader):** TBD
• **Office:** Zoom TBD
• **Office hours:** TBD
  – Do this Doodle by the end of Monday:
  – Mark *all* times you’re available during *any* week
Textbooks & Slides

Textbooks
- Computer Vision: Algorithms and Applications by Richard Szeliski
- Visual Object Recognition by Kristen Grauman and Bastian Leibe
- More resources available on course webpage

Slides
- Huge thanks to Prof. Adriana Kovashka at Pitt
- Your notes from class are your best study material, slides are not complete with notes
- Slides will be uploaded
  - I will “write” on the lecture notes during class, but this will not be uploaded (you will find them in the class recordings though).
Programming Language

- We’ll use Matlab
- It can be downloaded for free from MyPitt
- We’ll do a short tutorial; ask TA if you need further help
Course Structure

• Lectures
• Near-daily quizzes (4% of total grade)
  – 0.25% each, 20 mins to finish but really takes 3 mins
• Participation in class (8%)
  – Responding to the instructor's or others' questions
  – Asking and answering questions during class (mic or chat)
  – Answering others' questions on Canvas
  – Bringing in relevant articles you saw in the news (I may mention them in class)
  – Coming to office hours
  – Any way to engage and participate!
• Weekly programming assignments: (9 x 8%)
• Two essays: (2 x 8%)
Zoom Etiquette

• Please turn your camera on if you are joining class on Zoom
  – Will **not** affect your grade or anything if you don’t, just my wish to create a lively class
Health and Well-being
(In case we move to in-class)

• Follow university policies
• Please be kind to others and be patient
• Wear your mask over your nose and mouth
• Please reach out to the instructor if you want to talk about anything privately, and don’t hesitate to ask for help
• Use hangout channel: talk to your peers about anything
Warning #1

• This class is **a lot of work**

• I’ve opted for shorter, more manageable HW assignments, but there is a lot of them

• I expect you’d be spending **6-8 hours** on homework each week

• ... But you get to understand algorithms and concepts in detail!
Warning #2

• Some parts will be **hard** and require that you pay close attention!
• We will have low-stakes near-daily quizzes to help you stay focused and help me know how you are doing
• Use instructor’s and TA’s office hours

• ... You will learn a lot!
What you may be able to

• Understand how machines see and understand photos and images
• Have basic knowledge about some classic to modern computer vision techniques
• See why and how modern computer vision techniques (e.g., deep learning) have evolved in such ways
• See through the "magic" of modern CV
• Identify common pitfalls and mistakes ML/CV practitioners may make and learn how to avoid them
• Identify or at least guess the family of ML/CV models that modern products could be using
• Find appropriate set of techniques and methods for the task of your interest
• Start getting into advanced computer vision and deep learning techniques
• Identify the subareas/topics in CV that interest you
Policies and Schedule

http://pitt.edu/~sjh95/cs1674_sp21

Breakout: Read, discuss, come up with questions, report back
Questions?
Plan for Today

• Blitz introductions
• What is computer vision?
  — Why do we care?
  — What are the challenges?
  — What is recent research like?
• Overview of topics
• Review and tutorial
  — Linear algebra
  — Matlab
Blitz introductions (15 sec)

• What is your name?
• One of the followings:
  – What one thing outside of school are you passionate about?
  – What do you hope to get out of this class? Why CV?
  – What do you want to do if the COVID-19 ends and we go back to normal?
Computer Vision
What is computer vision?

Is camera = computer vision?

"We see with our brains, not with our eyes" (Oliver Sacks and others)
What is computer vision?

• You can collect or may already have images and videos

• But you need to **automatically understand** them to:
  – Allow a machine to **recognize** objects, people, scenes, and activities
  – **Mine**, **search**, and **interact** with visual data
  – **Compute** properties and **navigate** within the 3D world using visual data
  – **Generate** realistic synthetic visual data

Adapted from Kristen Grauman
Perception and interpretation

The Wicked Twister
Ferris wheel
Cedar Point
Lake Erie
amusement park

How to do? What are the challenges?

How to do? What are the challenges?

Objects
Activities
Scenes
Locations
Text / writing
Faces
Gestures
Motions
Emotions...

The Wicked Twister
Ferris wheel
Cedar Point
Lake Erie

How to do? What are the challenges?
Uniqueness of CV

• We can “see” the information
• This means your intuition has a good chance to be a good solution!
  – What you have just mentioned have all been or are actively being studied in the field.
• So share your thoughts and intuition with us
  – Many times, they will be correct and very reasonable
• I will try to provide strengths and weaknesses (rather than right-or-wrong)
Prepare you to be a CV scientist

1. What is the **objective**?
2. What is the **challenge**?
3. What **information/data** do you need/have?
4. What is the mathematical **formulation/task**?
5. What **method** can solve the mathematical formulation?

• This course is about thinking about this process together and learning techniques that enable these.
• When you face a new CV problem, you can answer these in optimal (**may be not perfectly**) way.
• The process still holds in advanced CV courses: You just learn slightly more advanced topics in each step.
Visual search, organization

Query → Image or video archives → Relevant content

How to do? What are the challenges?
Measurement

Real-time stereo

Structure from motion

Multi-view stereo for community photo collections

How to do? What are the challenges?
Generation

How to do? What are the challenges?

Karras et al., “Progressive Growing of GANs for Improved Quality, Stability, and Variation”, ICLR 2018
Related disciplines

- Artificial intelligence
- Machine learning
- Cognitive science
- Graphics
- Image processing
- Algorithms
Computer Vision vs. Computer Graphics

Vision: Understand 2D image as an object

Graphics: Understand 3D object as an image
Why vision?

• Images and video are everywhere!

- Personal photo albums
- Movies, news, sports
- Surveillance and security
- Medical and scientific images

Adapted from Lana Lazebnik

144k hours uploaded to YouTube daily
4.5 mil photos uploaded to Flickr daily
10 bil images indexed by Google
Why vision?

- As image sources multiply, so do applications
  - Relieve humans of boring, easy tasks
  - Perception for robotics / autonomous agents
  - Organize and give access to visual content
  - Description of content for the visually impaired
  - Human-computer interaction
  - Fun applications (e.g. art styles to my photos)

Breakout: Come up with 3 specific applications, discuss which ones most useful, report back.
Things that work well
Faces and digital cameras

Camera waits for everyone to smile to take a photo [Canon]

Setting camera focus via face detection
Face recognition
Optical character recognition

INPUT: IMAGE

OUTPUT: TEXT

MATLAB OCR

Exploring photo collections

Photo Tourism
Exploring photo collections in 3D

(a)  (b)  (c)

Snavely et al.
Safety & security

Navigation, driver safety

Monitoring pool (Poseidon)

Pedestrian detection
MERL, Viola et al.

Surveillance
Interactive systems

Shotton et al.
Video-based interfaces

Human joystick
NewsBreaker Live

Assistive technology systems
Camera Mouse
Boston College

YouTube Link
Vision for medical & neuroimages

Image guided surgery
MIT AI Vision Group

fMRI data
Golland et al.
Things that need more work

The latest at CVPR, ICCV, ECCV

CVPR = IEEE/CVF Conference on Computer Vision and Pattern Recognition
ICCV = IEEE/CVF International Conference on Computer Vision
ECCV = European Conference on Computer Vision
Our ability to detect objects has gone from 34 mAP in 2008 to 73 mAP at 7 FPS (frames per second) or 63 mAP at 45 FPS in 2016.

**YOLO: Real-Time Object Detection**

You only look once (YOLO) is a system for detecting objects on the Pascal VOC 2012 dataset. It can detect the 20 Pascal object classes:

- person
- bird, cat, cow, dog, horse, sheep
- aeroplane, bicycle, boat, bus, car, motorbike, train
- bottle, chair, dining table, potted plant, sofa, tv/monitor
Accurate object detection in real time

<table>
<thead>
<tr>
<th></th>
<th>Pascal 2007 mAP</th>
<th>Speed</th>
</tr>
</thead>
<tbody>
<tr>
<td>DPM v5</td>
<td>33.7</td>
<td>0.07 FPS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>14 s/img</td>
</tr>
<tr>
<td>R-CNN</td>
<td>66.0</td>
<td>0.05 FPS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>20 s/img</td>
</tr>
<tr>
<td>Fast R-CNN</td>
<td>70.0</td>
<td>0.5 FPS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2 s/img</td>
</tr>
<tr>
<td>Faster R-CNN</td>
<td>73.2</td>
<td>7 FPS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>140 ms/img</td>
</tr>
<tr>
<td>YOLO</td>
<td>69.0</td>
<td>45 FPS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>22 ms/img</td>
</tr>
</tbody>
</table>

50 mph -> 73 ft/sec
- 22ms/img -> 1 frame every 1.6 ft
- 140ms/img -> 1 frame every 10.22 ft

Recognition in novel modalities

Figure 6: Qualitative Results. YOLO running on sample artwork and natural images from the internet. It is mostly accurate although it does think one person is an airplane.
Learning from weak supervision

“Less-than-ideal but not totally useless information”
Unsupervised learning
Understanding activities and intents
Understanding how well activity performed

"Stretch Hands"

"Lower Feet"

Quality of Action: 86.5 / 100
Imagining motion in static images
Decoding physics from video
Figure 1: Our MovieQA dataset contains 14,944 questions about 408 movies. It contains multiple sources of information: plots, subtitles, video clips, scripts, and DVS transcriptions. In this figure we show example QAs from *The Matrix* and localize them in the timeline.
Understanding stories in film

Video Clip

Legend
- Character
- Attribute
- Relationship
- Interaction
- Summary Int.
- Topic
- Reason
- Timestamp

Scene: Field Road
Situation: Bullying
Description:
As Jenny and Forrest are on the road, three boys start throwing rocks at Forrest. Jenny urges him to run from them. While Forrest runs, his leg braces fall apart.
Understanding advertisements is more challenging than simply recognizing physical content from images, as ads employ a variety of strategies to persuade viewers.

We collect an advertisement dataset containing 64,832 images and 3,477 videos, each annotated by 3-5 human workers from Amazon Mechanical Turk.

<table>
<thead>
<tr>
<th>Image</th>
<th>Topic</th>
<th>204,340</th>
<th>Strategy</th>
<th>20,000</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sentiment</td>
<td>102,340</td>
<td>Symbol</td>
<td>64,131</td>
<td></td>
</tr>
<tr>
<td>Q+A Pair</td>
<td>202,090</td>
<td>Slogan</td>
<td>11,130</td>
<td></td>
</tr>
<tr>
<td>Video</td>
<td>Topic</td>
<td>17,345</td>
<td>Fun/Exciting</td>
<td>15,380</td>
</tr>
<tr>
<td>Sentiment</td>
<td>17,345</td>
<td>English?</td>
<td>17,374</td>
<td></td>
</tr>
<tr>
<td>Q+A Pair</td>
<td>17,345</td>
<td>Effective</td>
<td>16,721</td>
<td></td>
</tr>
</tbody>
</table>

Here are some sample annotations in our dataset.

- What’s being advertised in this image?
  - Cars, automobiles

- What sentiments are provoked in the viewer?
  - Amused, Creative, Impressed, Youthful, Conscious

- What strategies are used to persuade viewer?
  - Symbolism, Contrast, Straightforward, Transferred qualities

- What should the viewer do, and why should they do this?
  - I should buy Volkswagen because it can hold a big bear.
  - I should buy VW SUV because it can fit anything and everything in it.
  - I should buy this car because it can hold everything I need.

More information available at [http://cs.pitt.edu/~kovashka/ads](http://cs.pitt.edu/~kovashka/ads)
Reasoning and acting: Embodied question answering
Image generation

Figure 1. Examples of generated images from text descriptions. Left: captions are from zero-shot (held out) categories. Right: captions are from training set categories.

Reed et al., ICML 2016

Figure 3: Generated bedrooms after five epochs of training. There appears to be evidence of visual under-fitting via repeated noise textures across multiple samples such as the base boards of some of the beds.
Image generation
Transferring art styles

DeepArt.io – try it for yourself!
Video generation
Object eraser
Computer vision is not solved

- Deep learning makes excellent use of massive data (labeled for the task of interest?)
  - But it’s hard to understand how it does so, makes it hard to fix when it doesn’t work well
  - It doesn’t work well when massive data is not available and your task is different than tasks for which data is available
  - We can recognize objects with 97% accuracy but reasoning about relationships and intent is harder
Seeing AI

YouTube link

Microsoft Cognitive Services: Introducing the Seeing AI project
Why is vision difficult?

• Ill-posed problem: real world much more complex than what we can measure in images
  – 3D $\rightarrow$ 2D
  – Motion $\rightarrow$ static

• Impossible to literally “invert” image formation process with limited information
  – Need information outside of this particular image to generalize what image portrays (e.g. to resolve occlusion)
What the computer gets

Why is this problematic?

Adapted from Kristen Grauman and Lana Lazebnik
Challenges: many nuisance parameters

Illumination  
Object pose  
Clutter  

Occlusions  
Intra-class appearance  
Viewpoint  

Think again about the pixels...
Challenges: intra-class variation
Challenges: importance of context
Challenges: Complexity

- Thousands to millions of pixels in an image
- 3,000-30,000 human recognizable object categories
- 30+ degrees of freedom in the pose of articulated objects (humans)
- Billions of images indexed by Google Image Search
- 1.424 billion smart camera phones sold in 2015
- About half of the cerebral cortex in primates is devoted to processing visual information [Felleman and van Essen 1991]
Challenges: Limited supervision

Less

Unlabeled, multiple objects

Classes labeled, some clutter

More

Cropped to object, parts and classes

Kristen Grauman
Challenges: Vision requires reasoning

What color are her eyes?
What is the mustache made of?

How many slices of pizza are there?
Is this a vegetarian pizza?

Is this person expecting company?
What is just under the tree?

Does it appear to be rainy?
Does this person have 20/20 vision?
Evolution of datasets

- Challenging problem → active research area

PASCAL:
20 categories, 12k images

ImageNet:
22k categories, 14mil images

Microsoft COCO:
80 categories, 300k images
CV scientist mindset checklist

1. What is the **objective**?
2. What is the **challenge**?
3. What **information/data** do you need/have?
4. What is the mathematical **formulation/task**?
5. What **method** can solve the mathematical formulation?
Some Visual Recognition Problems: Why are they challenging?
Recognition: What objects do you see?
Detection: Where are the cars?
Activity: What is this person doing?
Scene: Is this an indoor scene?
Instance: Which city? Which building?
Visual question answering:
Why is there a carriage in the street?
Again: to prepare you to be a CV scientist

1. What is the **objective**?
2. What is the **challenge**?
3. What **information/data** do you need/have?
4. What is the mathematical **formulation/task**?
5. What **method** can solve the mathematical formulation?

- This course is about thinking about this process together and learning techniques that enable these.
- When you face a new CV problem, you can answer these in optimal (may be not perfectly!) way.
- The process still holds in advanced CV courses: You just learn slightly more advanced topics in each step.
Overview of topics

1. Objective
2. Challenge
3. Data/Information
4. Formulation/task
5. Method
Features and filters

1. Objective
2. Challenge
3. Data/Information
4. Formulation/task
5. Method

• Transforming and describing images; textures, colors, edges
Features and filters

- Detecting distinctive + repeatable features
- Describing images with local statistics
Grouping and fitting

1. Objective
2. Challenge
3. Data/Information
4. Formulation/task
5. Method

- Clustering, segmentation, fitting; what parts belong together?
Multiple views

- Multi-view geometry, matching, invariant features, stereo vision

Hartley and Zisserman

Fei-Fei Li

Kristen Grauman
Image categorization

• Fine-grained recognition

Generalist  Insect catching  Grain eating  Coniferous-seed eating  Nectar feeding

Chiseling  Dip netting  Surface skimming  Scything  Probing

Aerial fishing  Pursuit fishing  Scavenging  Raptorial  Filter feeding

Visipedia Project
Image categorization

- Material recognition

[Bell et al. CVPR 2015]
Image categorization

- Image style recognition

Flickr Style: 80K images covering 20 styles.

Wikipaintings: 85K images for 25 art genres.

[Karayev et al. BMVC 2014]
Visual recognition and SVMs

- Recognizing objects and categories, learning techniques

Adapted from Kristen Grauman
Convolutional neural networks (CNNs)

- State-of-the-art on many recognition tasks

Krizhevsky et al., NIPS 2012

Yosinski et al., ICML DL workshop 2015
Recurrent neural networks

- Sequence processing, e.g. question answering

1. Objective
2. Challenge
3. Data/Information
4. Formulation/task
5. Method
Motion and tracking

• Tracking objects, video analysis
Pose and actions

- Automatically annotating human pose (joints)
- Recognizing actions in first-person video
Linear algebra review
What are images? (in Matlab)

• Matlab treats images as matrices of numbers
• To proceed, let’s talk very briefly about how images are formed
Image formation

Illumination (energy) source

Scene element

Imaging system

(Internal) image plane (film)
Digital images

• **Sample** the 2D space on a regular grid
• **Quantize** each sample (round to nearest integer)

**FIGURE 2.17** (a) Continuous image projected onto a sensor array. (b) Result of image sampling and quantization.
Digital images

- **Sample** the 2D space on a regular grid
- **Quantize** each sample (round to nearest integer)
- What does quantizing signal look like?

Image thus represented as a matrix of integer values.

Adapted from S. Seitz
Digital color images

© 2000 How Stuff Works

Slide credit: Kristen Grauman
Digital color images

Color images, RGB color space:
Split image into three channels

Adapted from Kristen Grauman
**Images in Matlab**

- Color images represented as a matrix with multiple channels (=1 if grayscale)
- Suppose we have a NxM RGB image called “im”
  - im(1,1,1) = top-left pixel value in R-channel
  - im(y, x, b) = y pixels down, x pixels to right in the b\(^{th}\) channel
  - im(i, j, k) = row i, column j, channel k (Matlab and matrix convention)
  - im(N, M, 3) = bottom-right pixel in B-channel
- imread(filename) returns a uint8 image (values 0 to 255)
  - Convert to double format with double or im2double

```
0.92  0.93  0.94  0.97  0.62  0.37  0.85  0.97  0.93  0.92  0.99  
0.95  0.89  0.82  0.89  0.56  0.31  0.75  0.92  0.81  0.95  0.91  
0.89  0.72  0.51  0.55  0.51  0.42  0.57  0.41  0.49  0.49  0.91  
0.96  0.95  0.88  0.94  0.56  0.46  0.91  0.87  0.90  0.97  0.95  
0.71  0.81  0.81  0.87  0.57  0.37  0.80  0.88  0.89  0.79  0.85  
0.49  0.62  0.60  0.58  0.50  0.60  0.58  0.50  0.61  0.45  0.33  
0.86  0.84  0.74  0.58  0.51  0.39  0.73  0.92  0.91  0.49  0.74  
0.96  0.67  0.54  0.85  0.48  0.37  0.88  0.90  0.94  0.82  0.93  
0.69  0.49  0.56  0.66  0.43  0.42  0.77  0.73  0.71  0.90  0.99  
0.79  0.73  0.90  0.67  0.33  0.61  0.69  0.79  0.73  0.93  0.97  
0.91  0.94  0.89  0.49  0.41  0.78  0.78  0.77  0.89  0.99  0.93  
```

Row \( \rightarrow \)

Column \( \downarrow \)

Adapted from Derek Hoiem
Quick Demo

• **imtool**
  
  – *Image Processing and Computer Vision Toolbox*
  
    • Free to install during the installation or after under App
Vectors and Matrices

• Vectors and matrices are just collections of ordered numbers that represent something: movements in space, scaling factors, word counts, movie ratings, pixel brightnesses, etc.

• We’ll define some common uses and standard operations on them.
Vector

• A column vector $\mathbf{v} \in \mathbb{R}^{n \times 1}$ where

$$
\mathbf{v} = 
\begin{bmatrix}
  v_1 \\
  v_2 \\
  \vdots \\
  v_n
\end{bmatrix}
$$

• A row vector $\mathbf{v}^T \in \mathbb{R}^{1 \times n}$ where

$$
\mathbf{v}^T = 
\begin{bmatrix}
  v_1 & v_2 & \ldots & v_n
\end{bmatrix}
$$

$T$ denotes the transpose operation
Vector

• You’ll want to keep track of the orientation of your vectors when programming in MATLAB.
• You can transpose a vector $V$ in MATLAB by writing $V'$. 
Vectors have two main uses

- Vectors can represent an offset in 2D or 3D space
- Points are just vectors from the origin
  - Ex: xy coordinate
- Data can also be treated as a vector
- Such vectors don’t have a geometric interpretation, but calculations like “distance” still have value.
Example: Feature representation

- A vector representing *measurable characteristics* of a data sample we have
- E.g. a glass of juice can be represented via its
  - color = {yellow=1, red=2, green=3, purple=4} and
  - taste = {sweet=1, sour=2}
- A given glass $x_i$ can be represented as a vector:
  - $x_i = [3 \ 2]$ represents *sour, green* juice
- For *D features*, this defines a $D$-dimensional space
- Mathematically measure *similarity* between samples
Example: Feature representation

E.g. a glass of juice can be represented via its

\( \text{color} = \{ \text{yellow}=1, \text{red}=2, \text{green}=3, \text{purple}=4 \} \)

and \( \text{taste} = \{ \text{sweet}=1, \text{sour}=2 \} \)

L2 distance:
\[
\begin{align*}
    d(x_1, x_2) &= \sqrt{4+0} \\
    d(x_1, x_3) &= \sqrt{0+1} \\
    d(x_2, x_3) &= \sqrt{4+1}
\end{align*}
\]

L1 distance:
\[
\begin{align*}
    d(x_1, x_2) &= 2+0 \\
    d(x_1, x_3) &= 0+1 \\
    d(x_2, x_3) &= 2+1
\end{align*}
\]
Euclidean distance

Definition  [edit]

The Euclidean distance between points \( \mathbf{p} \) and \( \mathbf{q} \) is the length of the line segment connecting them (\( \overline{pq} \)).

In Cartesian coordinates, if \( \mathbf{p} = (p_1, p_2, \ldots, p_n) \) and \( \mathbf{q} = (q_1, q_2, \ldots, q_n) \) are two points in Euclidean \( n \)-space, then the Euclidean distance \( d \) from \( \mathbf{p} \) to \( \mathbf{q} \), or from \( \mathbf{q} \) to \( \mathbf{p} \) is given by the Pythagorean formula:[1]

\[
d(\mathbf{p}, \mathbf{q}) = d(\mathbf{q}, \mathbf{p}) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \cdots + (q_n - p_n)^2}
\]

\[
= \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2}.
\]

(1)

The position of a point in a Euclidean \( n \)-space is a Euclidean vector. So, \( \mathbf{p} \) and \( \mathbf{q} \) may be represented as Euclidean vectors, starting from the origin of the space (initial point) with their tips (terminal points) ending at the two points. The Euclidean norm, or Euclidean length, or magnitude of a vector measures the length of the vector:[1]

\[
\|
\mathbf{p}
\| = \sqrt{p_1^2 + p_2^2 + \cdots + p_n^2} = \sqrt{\mathbf{p} \cdot \mathbf{p}},
\]

where the last expression involves the dot product.
Norms

• L1 norm

\[ \| x \|_1 := \sum_{i=1}^{n} |x_i| \]

• L2 norm

\[ \| x \| := \sqrt{x_1^2 + \cdots + x_n^2} \]

• L^p norm (for real numbers \( p \geq 1 \))

\[ \| x \|_p := \left( \sum_{i=1}^{n} |x_i|^p \right)^{1/p} \]
Matrix

- A matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ is an array of numbers with size $m \downarrow$ by $n \rightarrow$, i.e. $m$ rows and $n$ columns.

$$
\mathbf{A} = \begin{bmatrix}
a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\
a_{21} & a_{22} & a_{23} & \cdots & a_{2n} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
a_{m1} & a_{m2} & a_{m3} & \cdots & a_{mn}
\end{bmatrix}
$$

- If $m = n$, we say that $\mathbf{A}$ is square.
Matrix Operations

• Addition

\[
\begin{bmatrix}
 a & b \\
 c & d \\
\end{bmatrix} + \begin{bmatrix}
 1 & 2 \\
 3 & 4 \\
\end{bmatrix} = \begin{bmatrix}
 a + 1 & b + 2 \\
 c + 3 & d + 4 \\
\end{bmatrix}
\]

– Can only add a matrix with matching dimensions, or a scalar.

\[
\begin{bmatrix}
 a & b \\
 c & d \\
\end{bmatrix} + 7 = \begin{bmatrix}
 a + 7 & b + 7 \\
 c + 7 & d + 7 \\
\end{bmatrix}
\]

• Scaling

\[
\begin{bmatrix}
 a & b \\
 c & d \\
\end{bmatrix} \times 3 = \begin{bmatrix}
 3a & 3b \\
 3c & 3d \\
\end{bmatrix}
\]
Matrix Multiplication

• Let $X$ be an $a \times b$ matrix, $Y$ be an $b \times c$ matrix
• Then $Z = X \ast Y = XY$ is an $a \times c$ matrix
• Second dimension of first matrix, and first dimension of second matrix have to be the same, for matrix multiplication to be possible
• Practice: Let $X$ be an $10 \times 5$ matrix. Let’s factorize it into 3 matrices: if $X = ABC$, then what could be the dimensions (sizes) of $A$, $B$, $C$?
Matrix Multiplication

• The product $X = AB$ is:

• Each entry $X(i, j)$ is $i'$th row of $A$ dot product with is $j'$th column of $B$. 
Matrix Multiplication

- Example:

\[
\begin{bmatrix}
0 & 2 \\
4 & 6 \\
\end{bmatrix} \times \begin{bmatrix}
1 & 3 \\
5 & 7 \\
\end{bmatrix}
\]

Each entry of the matrix product is made by taking the dot product of the corresponding row in the left matrix, with the corresponding column in the right one.

\[
0 \cdot 3 + 2 \cdot 7 = 14
\]
Inner Product (Dot Product)

- Multiply corresponding entries of two vectors and add up the result

\[ x^T y = \begin{bmatrix} x_1 & \ldots & x_n \end{bmatrix} \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} = \sum_{i=1}^{n} x_i y_i \quad \text{(scalar)} \]

- \( x^T y = x \cdot y = |x||y|\cos(\text{angle between } x \text{ and } y) \)

- If \( B \) is a unit vector, then \( A \cdot B \) gives the length of \( A \) which lies in the direction of \( B \) (projection)

(if \( B \) is unit-length hence norm is 1)
Different Types of Product

- \( x, y = \) column vectors (nx1)
- \( X, Y = \) matrices (mxn)
- \( x, y = \) scalars (1x1)

- \( x^T y = x \cdot y = \) inner product (1xn \( \times \) nx1 = scalar)
- \( x \otimes y = xy^T = \) outer product (nx1 \( \times \) 1xn = matrix)

- \( X * Y = XY = \) matrix product
- \( X .* Y = \) element-wise product
Matrix Operations

- **Transpose** – flip matrix, so row 1 becomes column 1

\[
\begin{bmatrix}
0 & 1 & \ldots
\end{bmatrix}
\]

\[
\begin{bmatrix}
0 & 1 \\
2 & 3 \\
4 & 5
\end{bmatrix}^T = \begin{bmatrix}
0 & 2 & 4 \\
1 & 3 & 5
\end{bmatrix}
\]

- A useful identity:

\[
(ABC)^T = C^T B^T A^T
\]
Matrix Operations

• MATLAB example:

\[ AX = B \]

\[ A = \begin{bmatrix} 2 & 2 \\ 3 & 4 \end{bmatrix}, \quad B = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \]

\[
\begin{array}{l}
>> x = A \backslash B \\
x =
\end{array}
\]

\[
\begin{bmatrix}
1.0000 \\
-0.5000
\end{bmatrix}
\]
Matrix Operation Properties

• Matrix addition is commutative and associative
  – $A + B = B + A$
  – $A + (B + C) = (A + B) + C$

• Matrix multiplication is associative and distributive but not commutative
  – $A(B*C) = (A*B)C$
  – $A(B + C) = A*B + A*C$
  – $A*B \neq B*A$
Special Matrices

• Identity matrix $\mathbf{I}$
  – Square matrix, 1’s along diagonal, 0’s elsewhere
  – $\mathbf{I} \cdot \text{[another matrix]} = \text{[that matrix]}$

$$
\begin{bmatrix}
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 1
\end{bmatrix}
$$

• Diagonal matrix
  – Square matrix with numbers along diagonal, 0’s elsewhere
  – A diagonal $\cdot \text{[another matrix]}$ scales the rows of that matrix

$$
\begin{bmatrix}
3 & 0 & 0 \\
0 & 7 & 0 \\
0 & 0 & 2.5
\end{bmatrix}
$$
Special Matrices

• Symmetric matrix

$$A^T = A$$

$$\begin{bmatrix}
1 & 2 & 5 \\
2 & 1 & 7 \\
5 & 7 & 1
\end{bmatrix}$$
Prepare your linear algebra skill

• Linear algebra will keep appear throughout the course
• If you are new to linear algebra, check the tutorial: http://cs229.stanford.edu/section/cs229-linalg.pdf
• If you are not new, check the tutorial for review: http://cs229.stanford.edu/section/cs229-linalg.pdf
• All the linear algebra you need for this course: http://cs229.stanford.edu/section/cs229-linalg.pdf
Matlab
Matlab tutorial

Main tutorial script: http://pitt.edu/~sjh95/cs1674_s21/lec1/tutorial.m

example of a function: http://pitt.edu/~sjh95/cs1674_s21/lec1/myfunction.m

example of a function 2: http://pitt.edu/~sjh95/cs1674_s21/lec1/myotherfunction.m

Please cover whatever we don’t finish at home.
Other tutorials and exercises

• https://people.cs.pitt.edu/~milos/courses/cs2750-Spring2020/Lectures/matlab_tutorial.pdf

• http://www.math.udel.edu/~braun/M349/Matlab_probs2.pdf