Session 2

- Case Studies – 15 mins
- Recap of Session 1
- Python Syntax
- Numpy
- Fun experiment – Chatbot using Slack and Google Dialogflow

Session 2

- Case Studies – 15 mins
- Recap of Session 1
- Python Syntax
- Numpy
- Fun experiment – Chatbot using Slack and Google Dialogflow

Quora - identify duplicate Qs

- An important product principle for Quora is that there should be a single question page for each logically distinct question.
  - the queries

    "What is the most populous state in the USA?" and

    "Which state in the United States has the most people?"

    should not exist separately on Quora because the intent is identical.
- Currently Quora uses Random Forest and they launched a competition on Kaggle to see if they could get better approaches than the one in use.
- Winning team used DeepLearning approach to NLP
- https://www.kaggle.com/c/quora-question-pairs

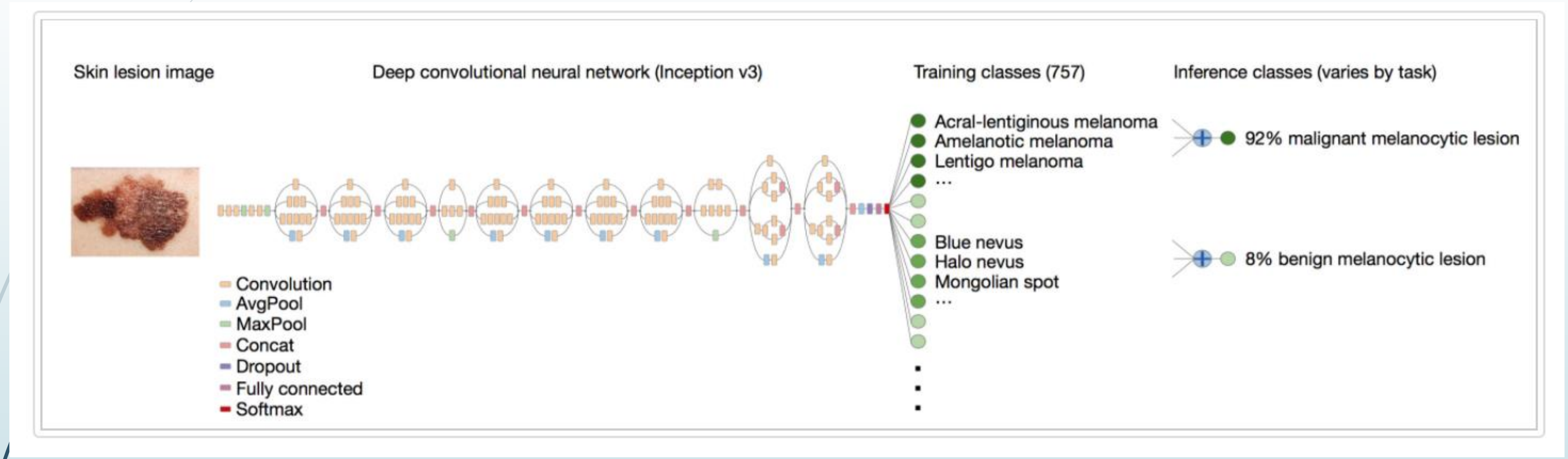
Dermatologist-level classification of skin cancer

- five-year survival rate
  - Early detection 97%
    Late detection 14%

- Techniques used
  - DeepLearning (CNN – Convolutional Neural Networks) with image pixels as raw values
  - Transfer learning – take a pretrained model and change last few layers to your specific problem and retrain last few layers

https://cs.stanford.edu/people/esteva/nature/

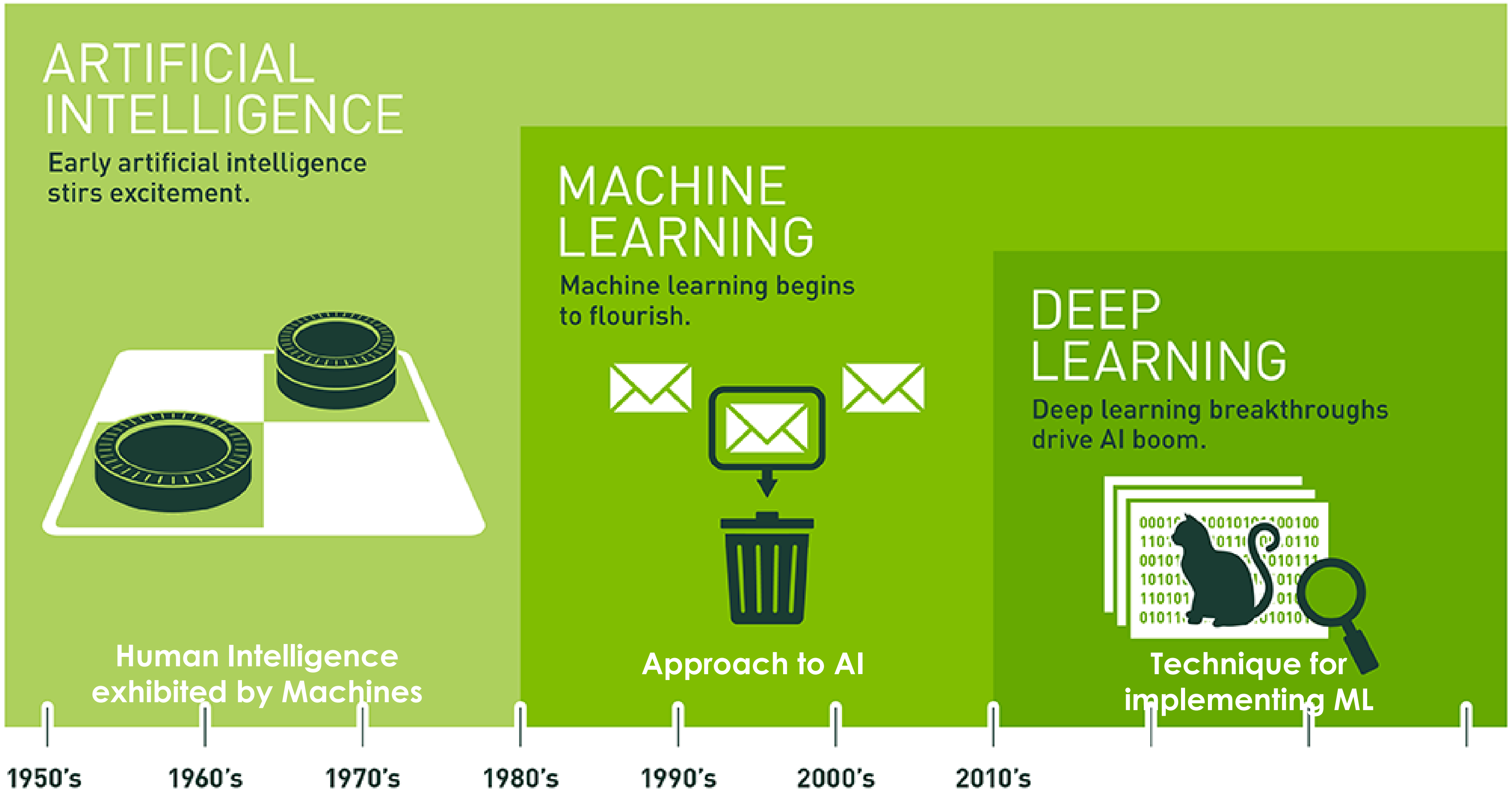
Dermatologist-level classification of skin cancer



Inception V3 architecture from Google (aka GoogLeLet) – variant of winner of ILSVRC 2014 (we saw VGG16 from Oxford which was runners up in 2014)
The trained network from Google was used and only last few layers were modified for Skin Cancer detection

Session 2

- Case Studies – 15 mins
- Recap of Session 1
- Python Syntax
- Numpy
- Fun experiment – Chatbot using Slack and Google Dialogflow


ARTIFICIAL INTELLIGENCE
Early artificial intelligence stirs excitement.
MACHINE LEARNING
Machine learning begins to flourish.
DEEP LEARNING
Deep learning breakthroughs drive AI boom.
Human Intelligence exhibited by Machines
Approach to AI
Technique for implementing ML
1950's
1960's
1970's
1980's
1990's
2000's
2010's

Since an early flush of optimism in the 1950s, smaller subsets of artificial intelligence – first machine learning, then deep learning, a subset of machine learning – have created ever larger disruptions.

Source: https://blogs.nvidia.com/blog/2016/07/29/whats-difference-artificial-intelligence-machine-learning-deep-learning-ai/

Best Defn of DeepLearning Problems

Tough to describe ….

but easy to do for Humans

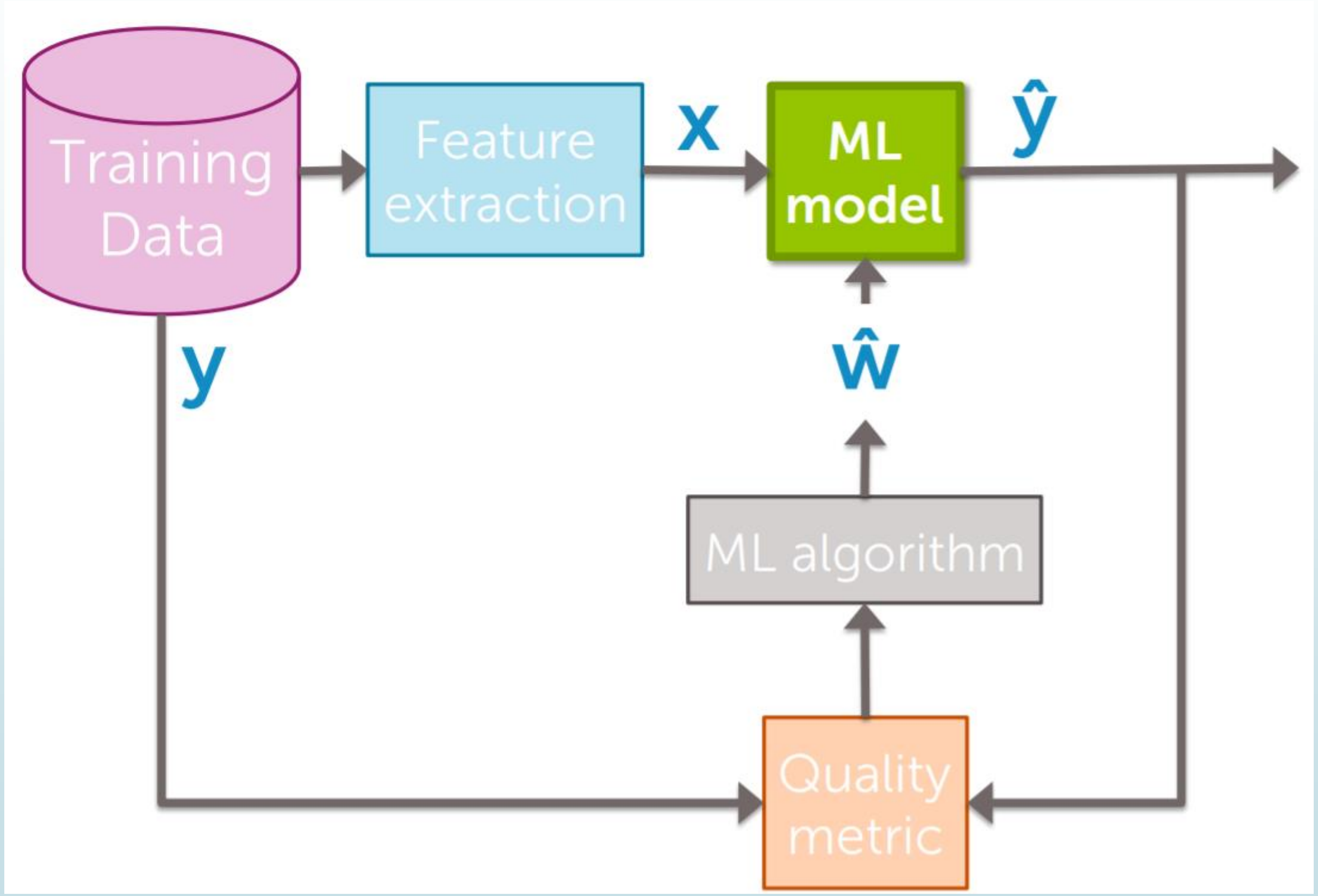
Why now?

- Higher Computing power with special purpose hardware (GPU)
- Availability of high volume data auto generated in digital world
- Newer Advances in Computational Models

Learning types

- Supervised Learning
  - Given X^{Train} and Y^{Train}
  - Learn relationship: X^{Train} -> Y^{Train}
  - Use relationship to predict output for new set of inputs

So what are we trying to do?

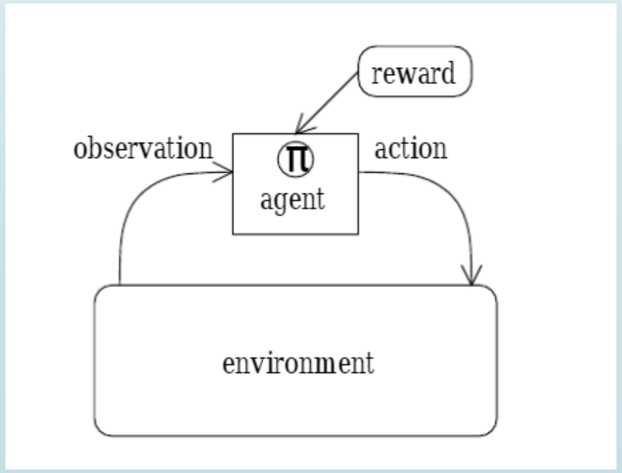


Learning types

- Unsupervised Learning
  - Given X^{Train}
  - Try to find common patterns to segment /cluster the data
  - The trained system predicts the segment/cluster a new data belongs to

Learning types

- Reinforcement Learning
  - Learn from responses system gets for some action taken and slowly become more intelligent in taking right steps
  - Example – chess playing program trained to beat human champs
  - One of the hottest areas



Session 2

- Case Studies – 15 mins
- Recap of Session 1
- Python Syntax
- Numpy
- Fun experiment – Chatbot using Slack and Google Dialogflow

Linear Algebra

- This is a refresher. Assumes some working knowledge of vectors/matrices – you can refer to various online courses to get a deeper understanding

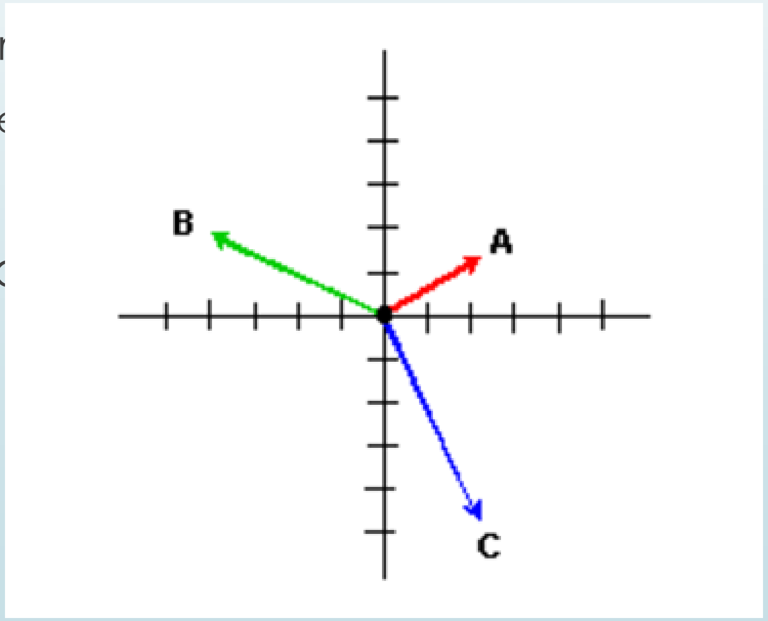
- We will use numpy to learn as we go along
  - You import it using statement

```
import numpy as np
```

- Material is largely based on materials from internet specifically - http://www.scipy-lectures.org

What is a vector

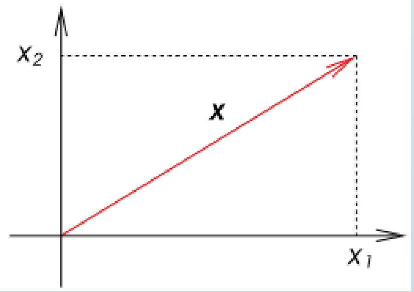
- What is a vector?
  - Simple answer – a single dimensional array of objects – usually numbers
  - [4.5, 3.0, -1.5]   [True, False, False, True], [5, 4, 1, 7]
  - Example of vector/ar
    - Voice signal sample
    - Pixels of an image
- Geometrical interpreta



What is a vector(2)

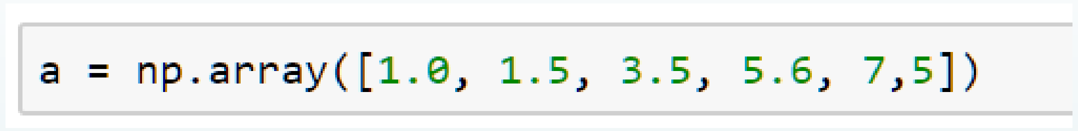
- We usually use vector to represent one single input
  - Say we are trying to predict house price based on #bedrooms, #area, #distance_school
  - So two inputs could be represented as
    - [3, 1345, 5.1]   and [2.5, 1100, 2.3]
  - So we have two inputs with each having 3 values (i.e. 3 dimensional vector)
  - Mathematically vectors are shown as tall columns – vector and transpose

$$\mathbf{x} = \begin{pmatrix} x_1 \\ \vdots \\ x_m \end{pmatrix}, \quad \mathbf{x}^T = (x_1, \ldots, x_m)\,.$$

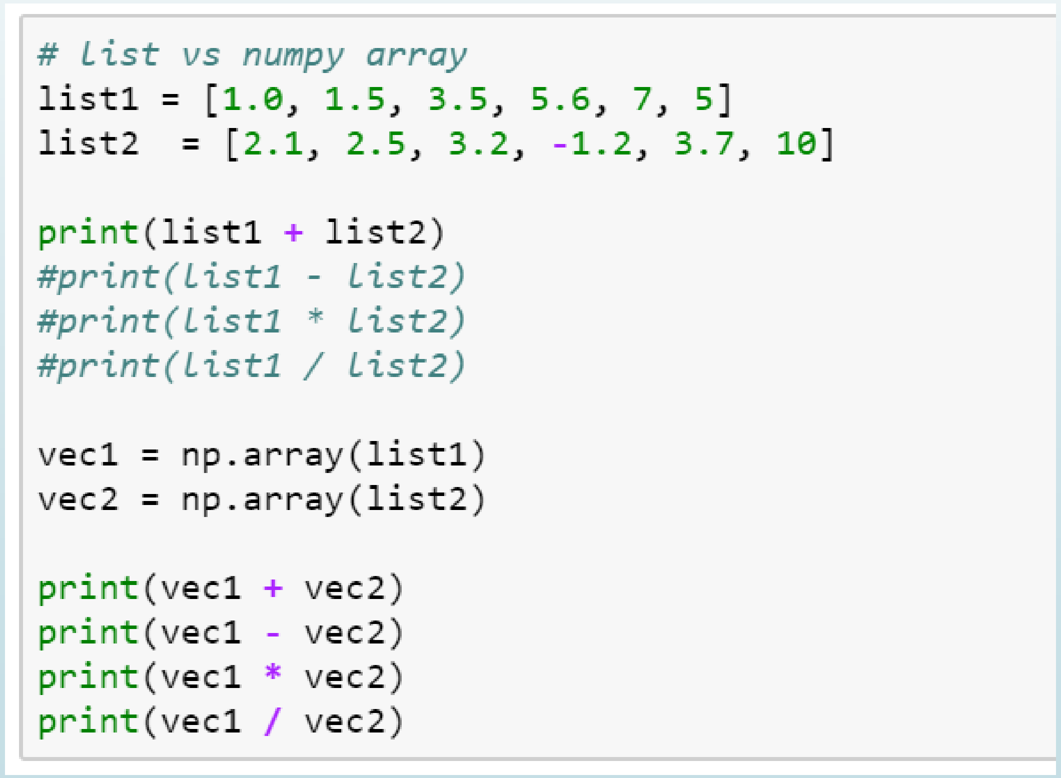


What is a vector(3)

- You can create it using following command
- Numpy Vectors/arrays are light-weight wrappers over python lists and it does alter some basic behavior of list operators.



```
a = np.array([1.0, 1.5, 3.5, 5.6, 7,5])
```

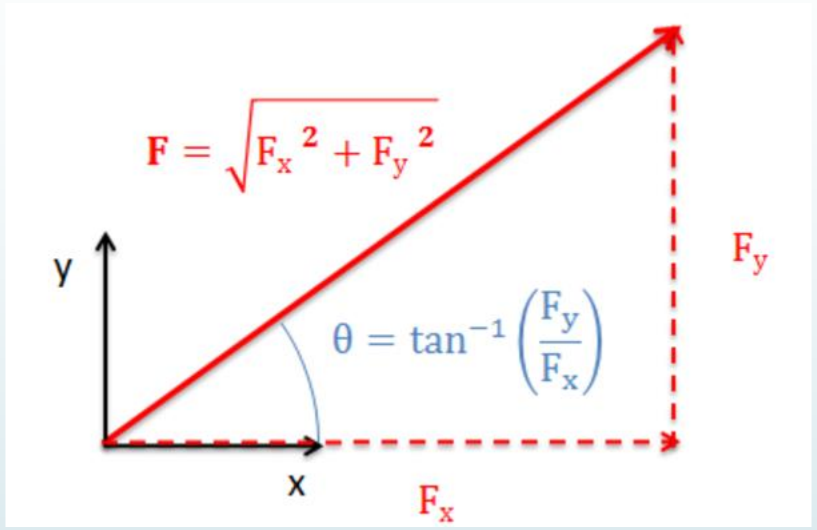


```
# list vs numpy array
list1 = [1.0, 1.5, 3.5, 5.6, 7, 5]
list2  = [2.1, 2.5, 3.2, -1.2, 3.7, 10]

print(list1 + list2)
#print(list1 - list2)
#print(list1 * list2)
#print(list1 / list2)

vec1 = np.array(list1)
vec2 = np.array(list2)

print(vec1 + vec2)
print(vec1 - vec2)
print(vec1 * vec2)
print(vec1 / vec2)
```

Vector Magnitude

- Magnitude:

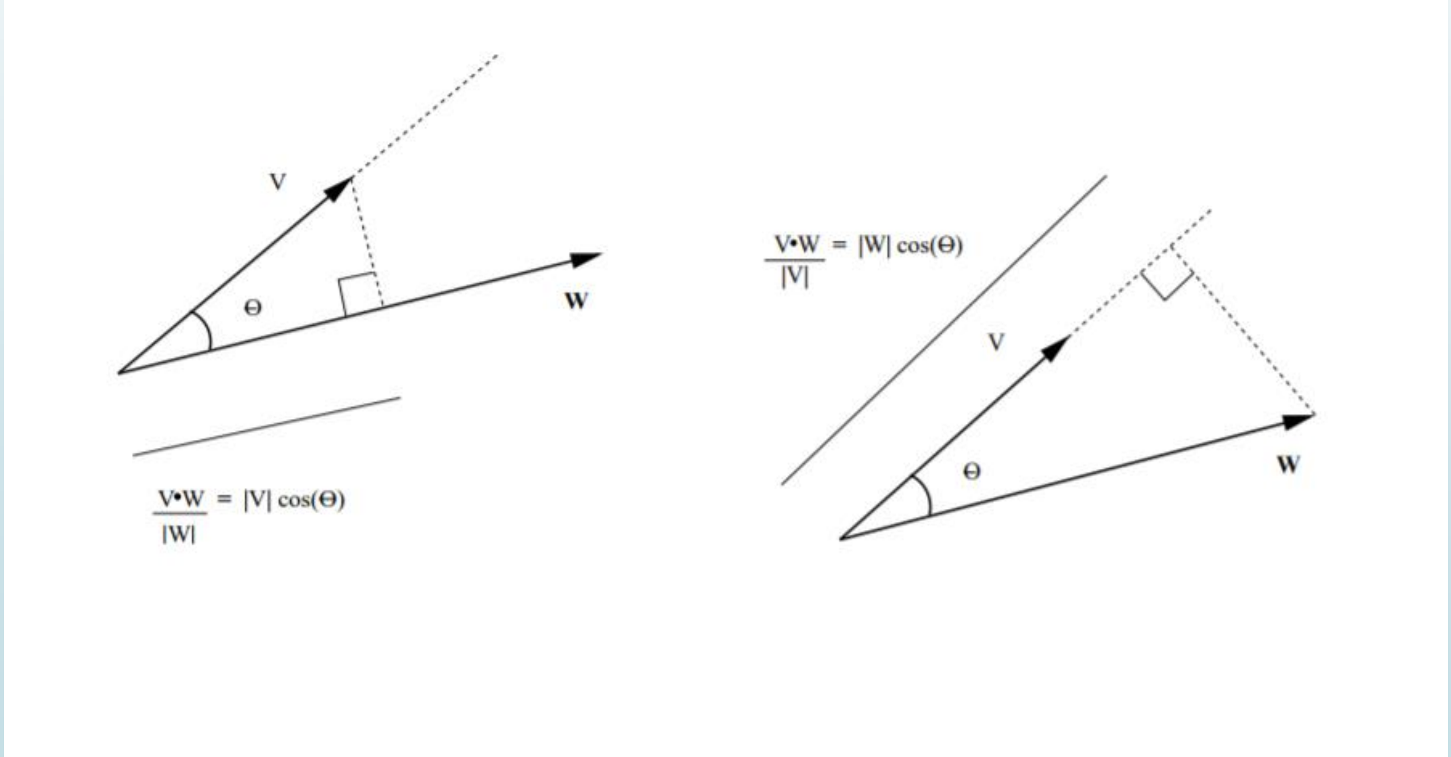
- Exercise create a new vector and find magnitude of that vector in num
- Unit vectors
  - What is it? – normalize norm to 1
  - How to do in python?


$\mathbf{F} = \sqrt{F_x{}^2 + F_y{}^2}$
y
$\theta = \tan^{-1}\left(\frac{F_y}{F_x}\right)$
F_y
x
F_x

Vector Inner/dot/scalar product

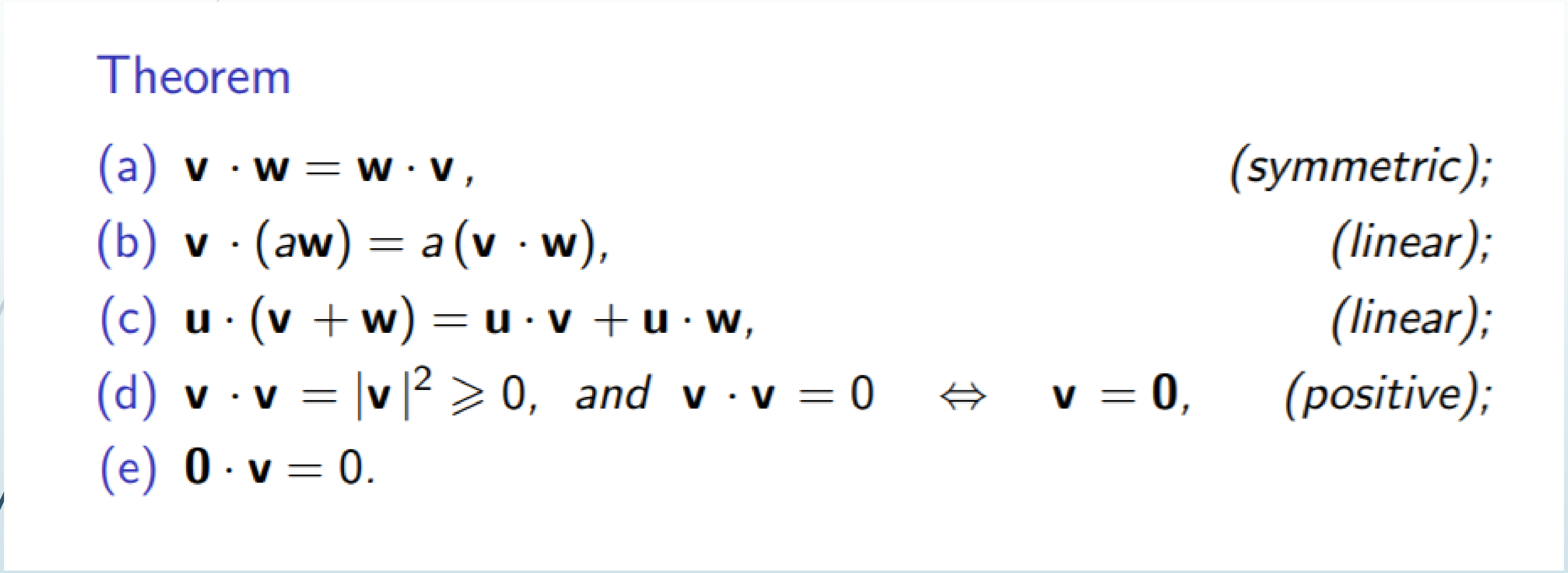
- Mathematically:
  - Multiply corresponding elements and add them together
  - |x| . |y| . Cos(angle)
  - Projection of
    - x on y or
    - y on x

$$\mathbf{x}^T\mathbf{y} = (x_1, \ldots, x_m)\begin{pmatrix} y_1 \\ \vdots \\ y_m \end{pmatrix} = x_1y_1 + \ldots + x_my_m.$$



V
Θ
W
$\frac{V \cdot W}{|W|} = |V| \cos(\Theta)$
$\frac{V \cdot W}{|V|} = |W| \cos(\Theta)$
V
Θ
W

Properties of dot product



Theorem

(a) $\mathbf{v} \cdot \mathbf{w} = \mathbf{w} \cdot \mathbf{v}$, *(symmetric);*

(b) $\mathbf{v} \cdot (a\mathbf{w}) = a\,(\mathbf{v} \cdot \mathbf{w})$, *(linear);*

(c) $\mathbf{u} \cdot (\mathbf{v} + \mathbf{w}) = \mathbf{u} \cdot \mathbf{v} + \mathbf{u} \cdot \mathbf{w}$, *(linear);*

(d) $\mathbf{v} \cdot \mathbf{v} = |\mathbf{v}|^2 \geqslant 0, \ \textit{and} \ \mathbf{v} \cdot \mathbf{v} = 0 \quad \Leftrightarrow \quad \mathbf{v} = \mathbf{0}$, *(positive);*

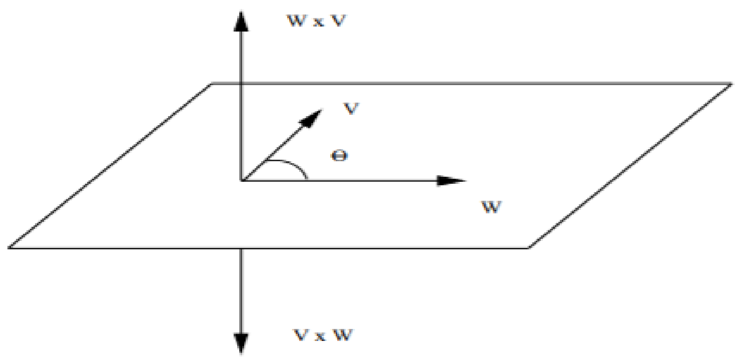
(e) $\mathbf{0} \cdot \mathbf{v} = 0.$

Cross product

Definition

Let **v** , **w** be vectors in $\mathbb{R}^3$ having length $|\mathbf{v}|$ and $|\mathbf{w}|$ with angle in between θ, where $0 \leq \theta \leq \pi$. The *cross product* of **v** and **w**, denoted as $\mathbf{v} \times \mathbf{w}$, is a vector perpendicular to both **v** and **w**, pointing in the direction given by the right hand rule, with norm

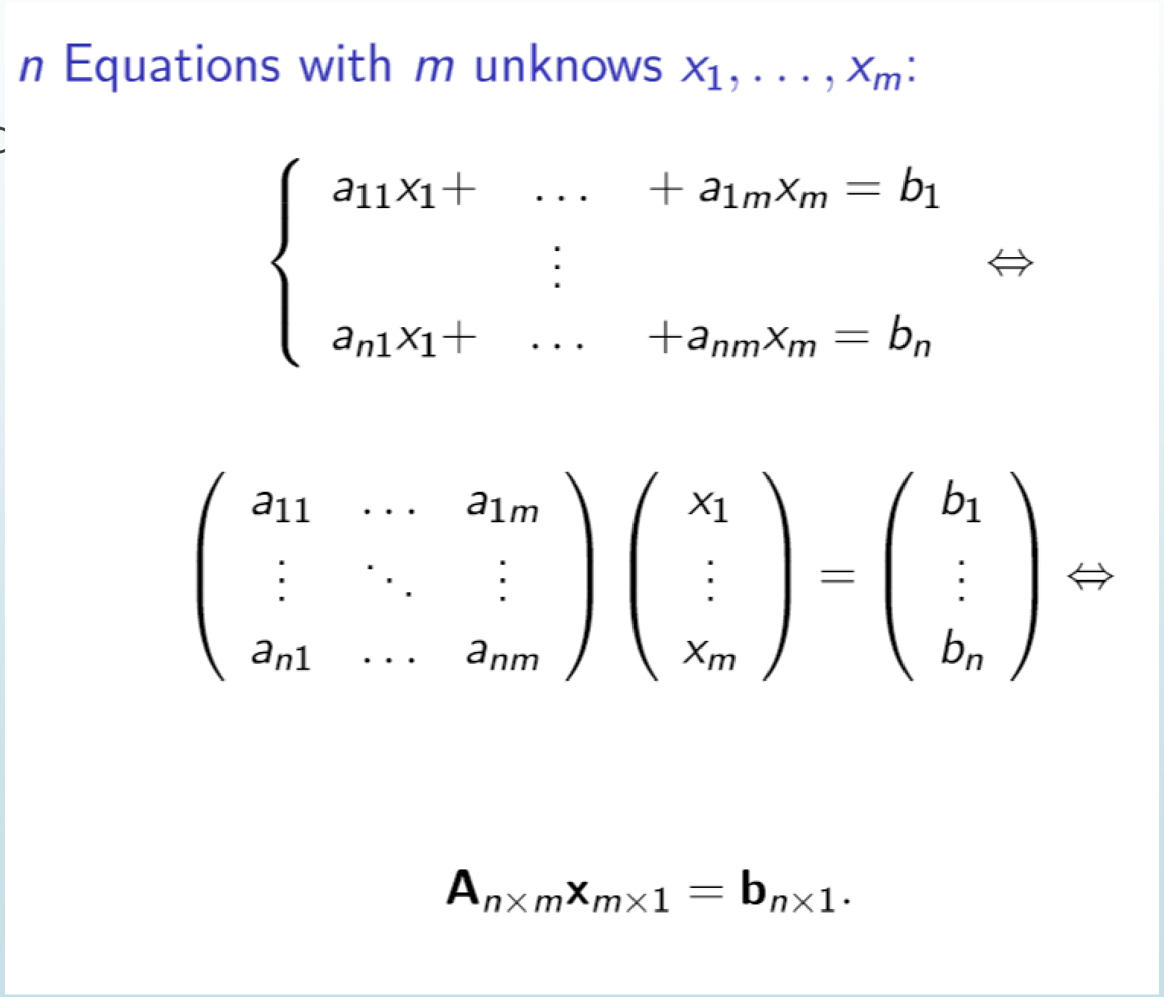
$$|\mathbf{v} \times \mathbf{w}| = |\mathbf{v}|\,|\mathbf{w}| \sin(\theta).$$



Cross product vectors are perpendicular to the original vectors.

Matrix

- Two dimensional array of dc
- Motivation?
- Numpy?



n Equations with m unknows $x_1, \dots, x_m$:

$$\left\{ \begin{array}{cccc} a_{11}x_1 + & \dots & + a_{1m}x_m = b_1 \\ & \vdots & \\ a_{n1}x_1 + & \dots & + a_{nm}x_m = b_n \end{array} \right. \Leftrightarrow$$

$$\begin{pmatrix} a_{11} & \dots & a_{1m} \\ \vdots & \ddots & \vdots \\ a_{n1} & \dots & a_{nm} \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_m \end{pmatrix} = \begin{pmatrix} b_1 \\ \vdots \\ b_n \end{pmatrix} \Leftrightarrow$$

$$\mathbf{A}_{n \times m} \mathbf{x}_{m \times 1} = \mathbf{b}_{n \times 1}.$$

Matrix-vector multiplication

- If A is a matrix of size (mxn)
- If B is a vector of size “n” i.e. matrix of size (nx1)
- Result of AB will be of size (mx1)

$$\mathbf{AB} = \begin{pmatrix} a & b & c \\ p & q & r \\ u & v & w \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} ax + by + cz \\ px + qy + rz \\ ux + vy + wz \end{pmatrix},$$

- Numpy?

Matrix-Matrix multiplication

- If A is a matrix of size (mxn)
- If B is a matrix of size (nxk)
- Result of AB will be of size (mxk)

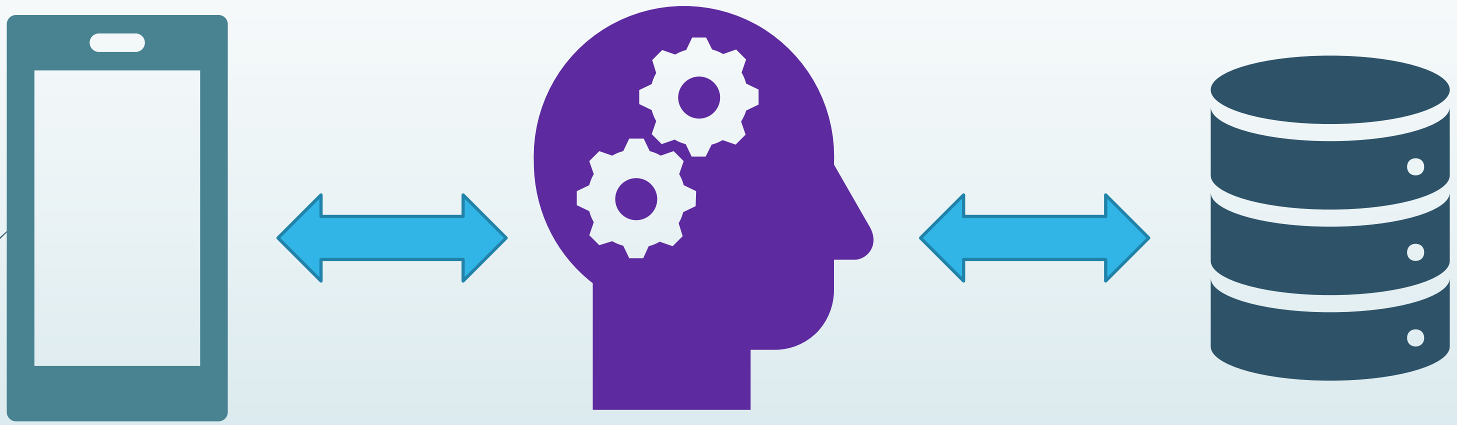
$$\mathbf{AB} = \begin{pmatrix} a & b & c \\ p & q & r \\ u & v & w \end{pmatrix} \begin{pmatrix} \alpha & \beta & \gamma \\ \lambda & \mu & \nu \\ \rho & \sigma & \tau \end{pmatrix} = \begin{pmatrix} a\alpha + b\lambda + c\rho & a\beta + b\mu + c\sigma & a\gamma + b\nu + c\tau \\ p\alpha + q\lambda + r\rho & p\beta + q\mu + r\sigma & p\gamma + q\nu + r\tau \\ u\alpha + v\lambda + w\rho & u\beta + v\mu + w\sigma & u\gamma + v\nu + w\tau \end{pmatrix}$$

- Numpy?

Session 2

- Case Studies – 15 mins
- Recap of Session 1
- Python Syntax
- Numpy
- Fun experiment – Chatbot using Slack and Google Dialogflow

ChatBots



UI
slack, custom,
Google, Messenger

Dialogflow
Entities, intents,
context

Services
Fulfillment
Application

Steps

- Create account with Google Dialogflow (www.dialogflow.com)
- Create a new Bot
- Enable small talk
- Create an entity Tea with 3/4 types of teas
- Create one or two Intents
- Enable integration and test with Slack