Machine Learning using Python

Bio

- Computer Vision Data Scientist
- PhD in Electrical and Computer Engineering
- CCC Information Services/ Nokia Bell labs
- Love photography!

General Notes:

Neda Hantehzadeh’s Email: nhante2@uic.edu

Office hours: 5:00 pm – 6:00 pm Wednesdays before class
Location: TBD

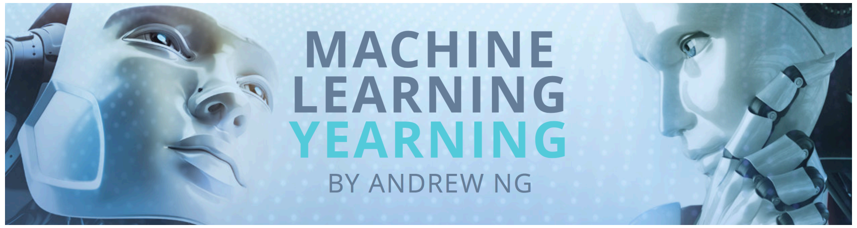
TA: Cheng Chen, Email: cchen224@uic.edu
Office hours: 2:00 pm-4:00 pm Mondays, UH 2401

Course Syllabus

- 1st week: Introduction to machine learning and the course
- 2nd week: Neural Networks
- 3rd week: Deep Learning
- 4th week: Deep Learning
- 5th Session: Mid-Term exam
- 6th session: SVM
- 7th session: Gradient Boosting
- 8th session: Final Project

Resources

1. http://www.mlyearning.org by Andrew Ng

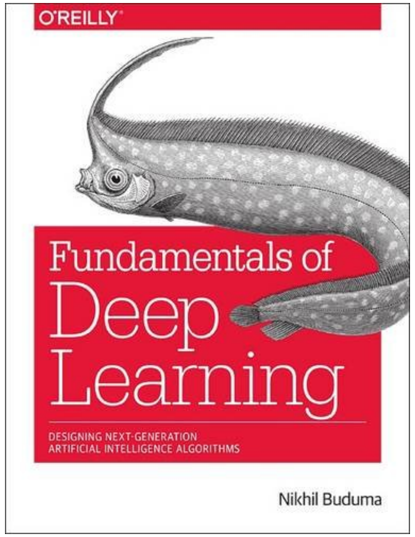


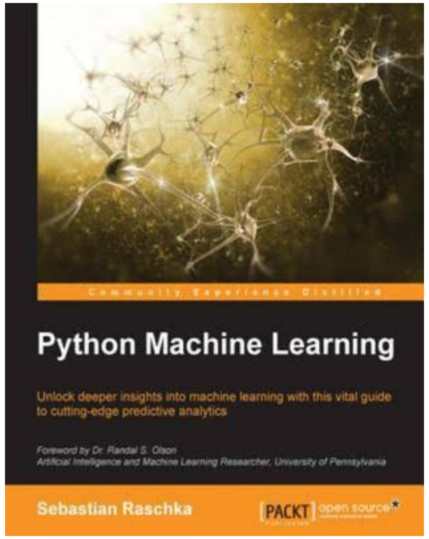
2. Python Machine Learning, Sebastian Raschka

3. Fundamentals of Deep Learning , Nikhil Buduma

   The currently working table of contents for this book is listed below:

   - The Neural Network
   - Training Feed Forward Neural Networks
   - Implementing Neural Networks in TensorFlow
   - Beyond Gradient Descent
   - Convolutional Neural Networks:
   - …..




Community Experience Distilled
Python Machine Learning
Unlock deeper insights into machine learning with this vital guide to cutting-edge predictive analytics
Foreword by Dr. Randal S. Olson
Artificial Intelligence and Machine Learning Researcher, University of Pennsylvania
Sebastian Raschka
PACKT open source

Course Goals

- Build multiple machine learning applications using Python
- Become familiar with Neural Networks and specifically Deep learning functions
- Gain experience with real data

Prerequisites

- Programming with Python, Numpy and Pandas and knowledge of data structures
- Comfortable with abstract mathematical concepts (probability, statistics, matrix algebra, …)
- General understanding of data mining and machine learning

Course Overview

- **Lectures:** Wednesday evenings 6-8:30 Burnham Hall 305
- **Homework:** week 3 and week 6
- **Exams:** Mid-term on week 5
- **Projects:** 4-6 person teams applying related machine learning algorithms learned during the course to real data from Kaggle including proposal, report and presentation

Grading

- Homework: (30%) 2 homework will be given (15% each)
- Mid-term: (30%)
- Final Project (30%)
- Attendance (10%)

Homework Assignments

- First homework data and questions will be provided on 3rd week
  - Deadline is the 4th week
  - Second homework will be given on the 6th week and the due date will be on 7th week
  - Every homework needs to be submitted individually, no sharing of code and/or solution
  - Copying homework solution from others will result in zero credit for that homework

1st week: Introduction

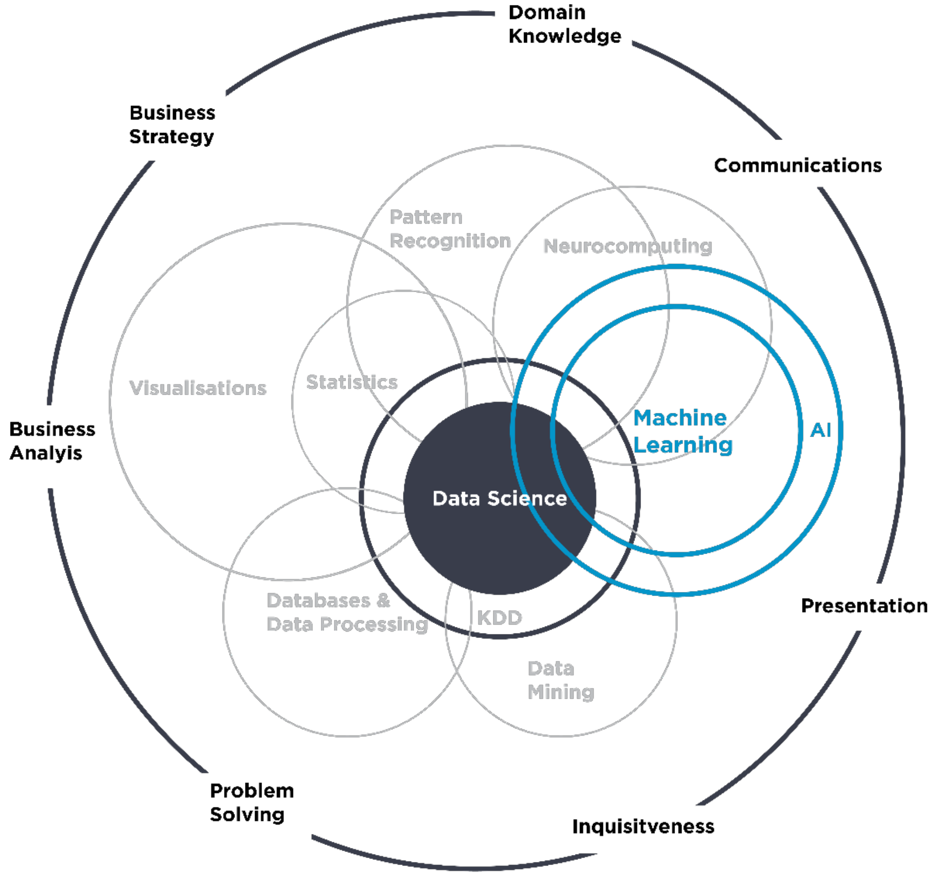
- What is the difference between data science and machine learning?
- Supervised vs Unsupervised
- What is Neural Network?
- Applications of Neural Networks
- Vectorization in Python
- Broadcasting in Python
- Instruction to get Python/Anaconda, Keras and required packages for coding

Machine Learning vs. Data Science

Data Mining + Machine Learning + Statistics + Visualization = Data Science

Machine learning is part of AI.
It is an art of creating algorithms which learn on their own instead of programing the machine for every task.

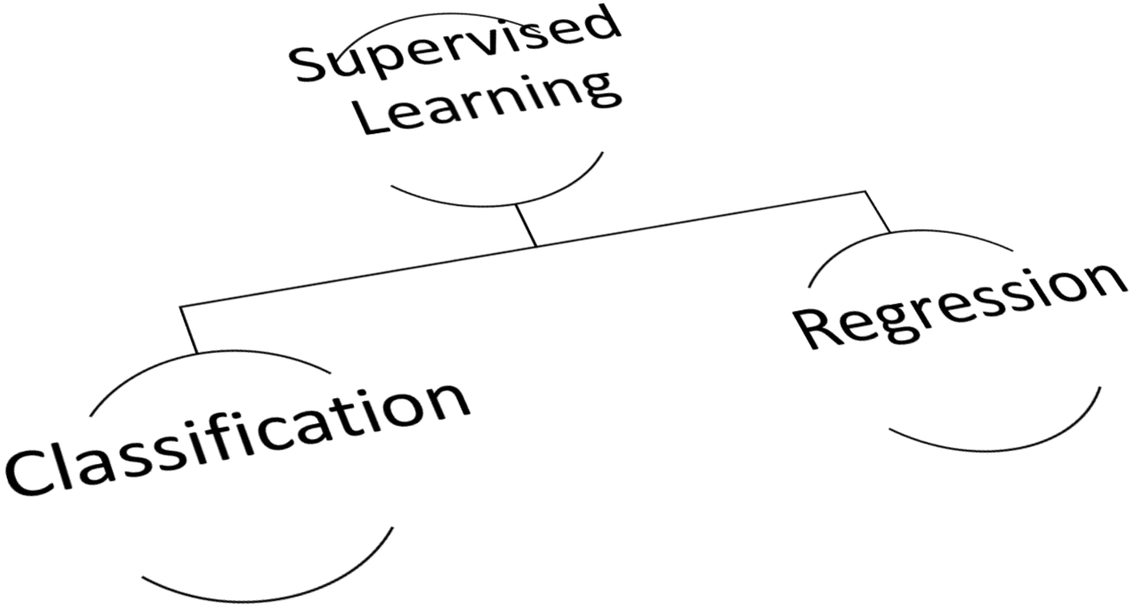
Machine learning is usually the ultimate goal of data mining for predicting from the data.



Source: http://www.kdnuggets.com/2014/06/data-science-skills-business-problems.html

What is Supervised Learning?

In supervised learning, we are given a data set and already know what our correct output should look like, having the idea that there is a relationship between the input and the output.

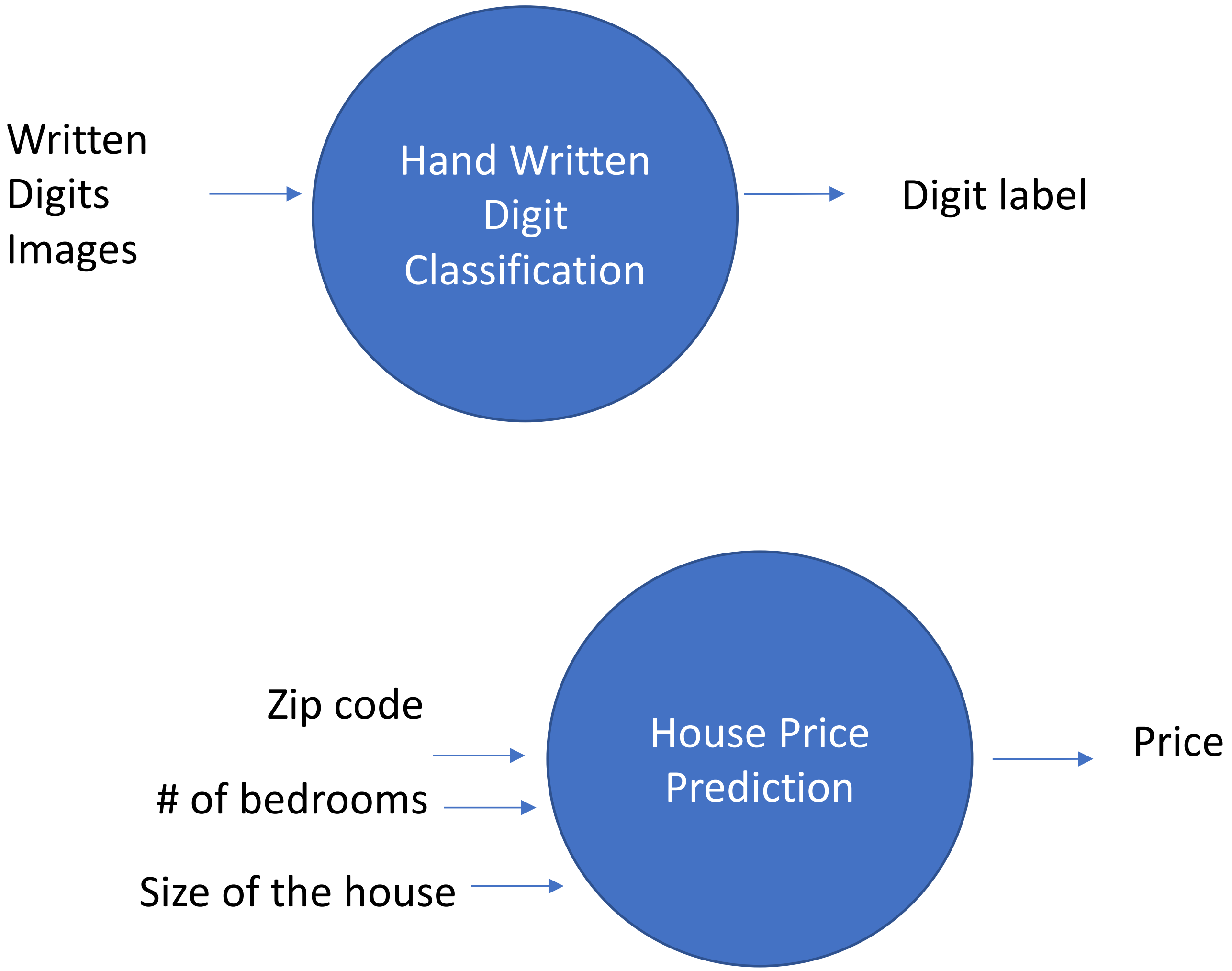


Supervised Learning

Classification

Regression

Supervised



Written Digits Images

Hand Written Digit Classification

Digit label

Zip code

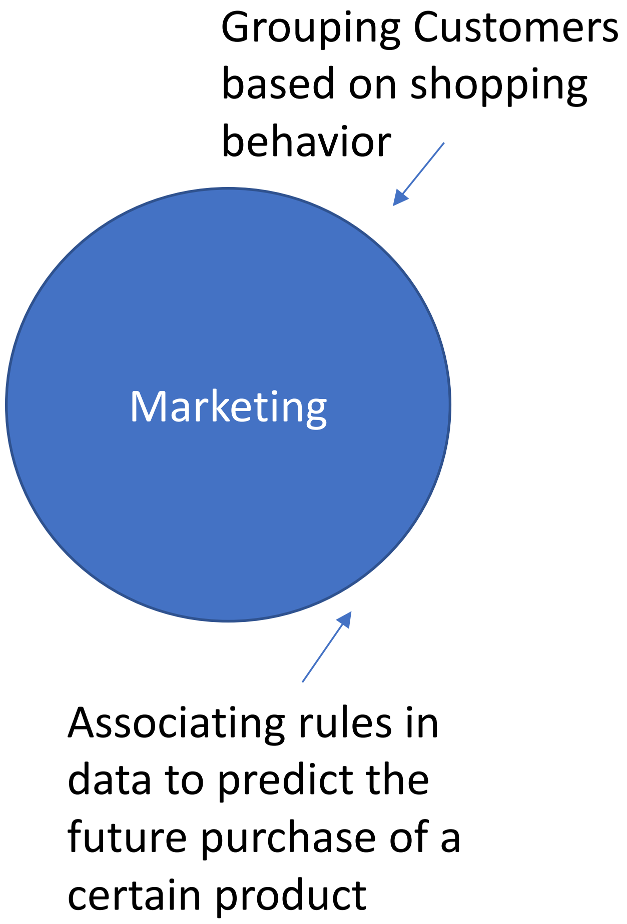
# of bedrooms

Size of the house

House Price Prediction

Price

Unsupervised

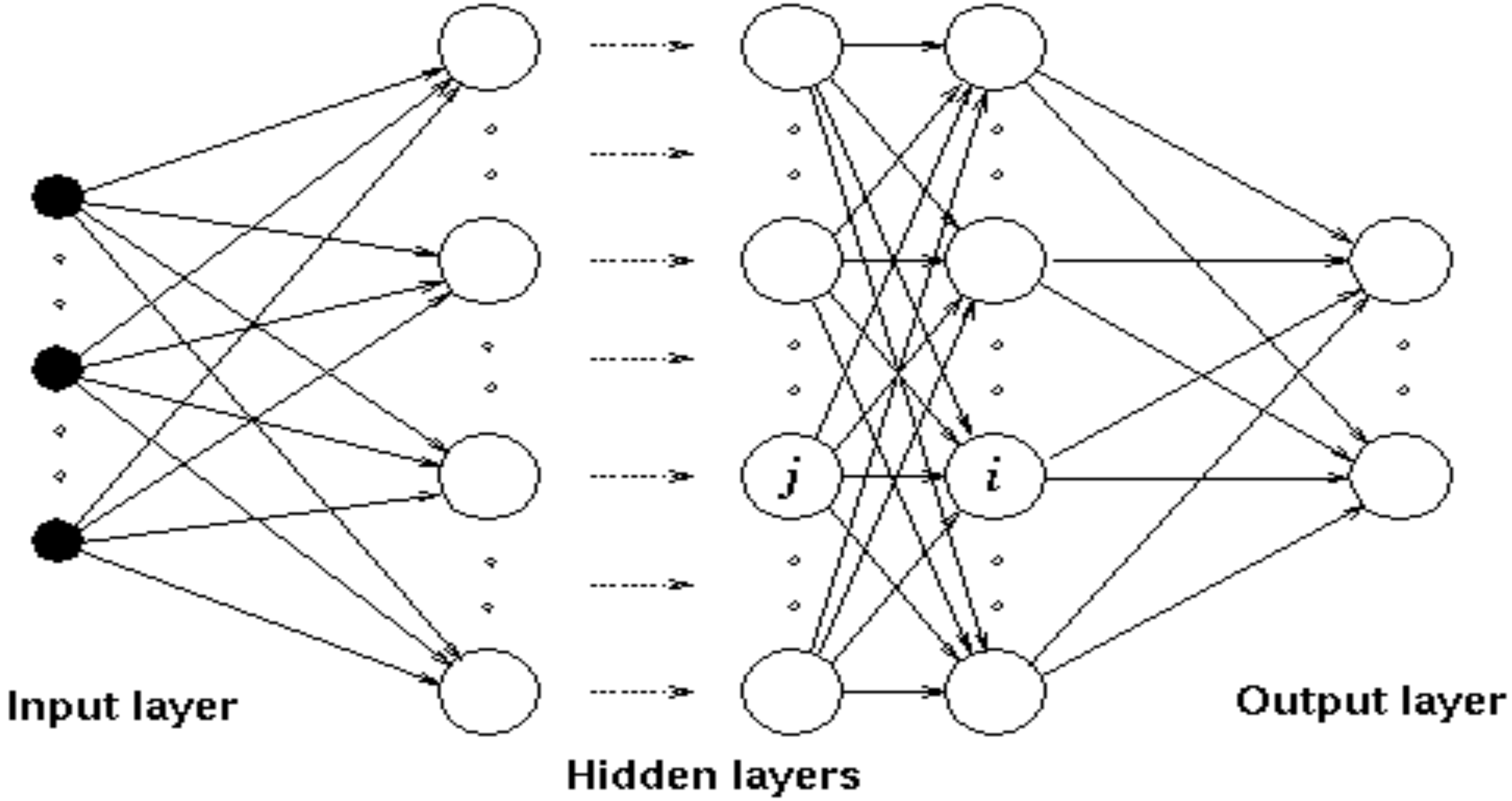


Grouping Customers based on shopping behavior

Marketing

Associating rules in data to predict the future purchase of a certain product

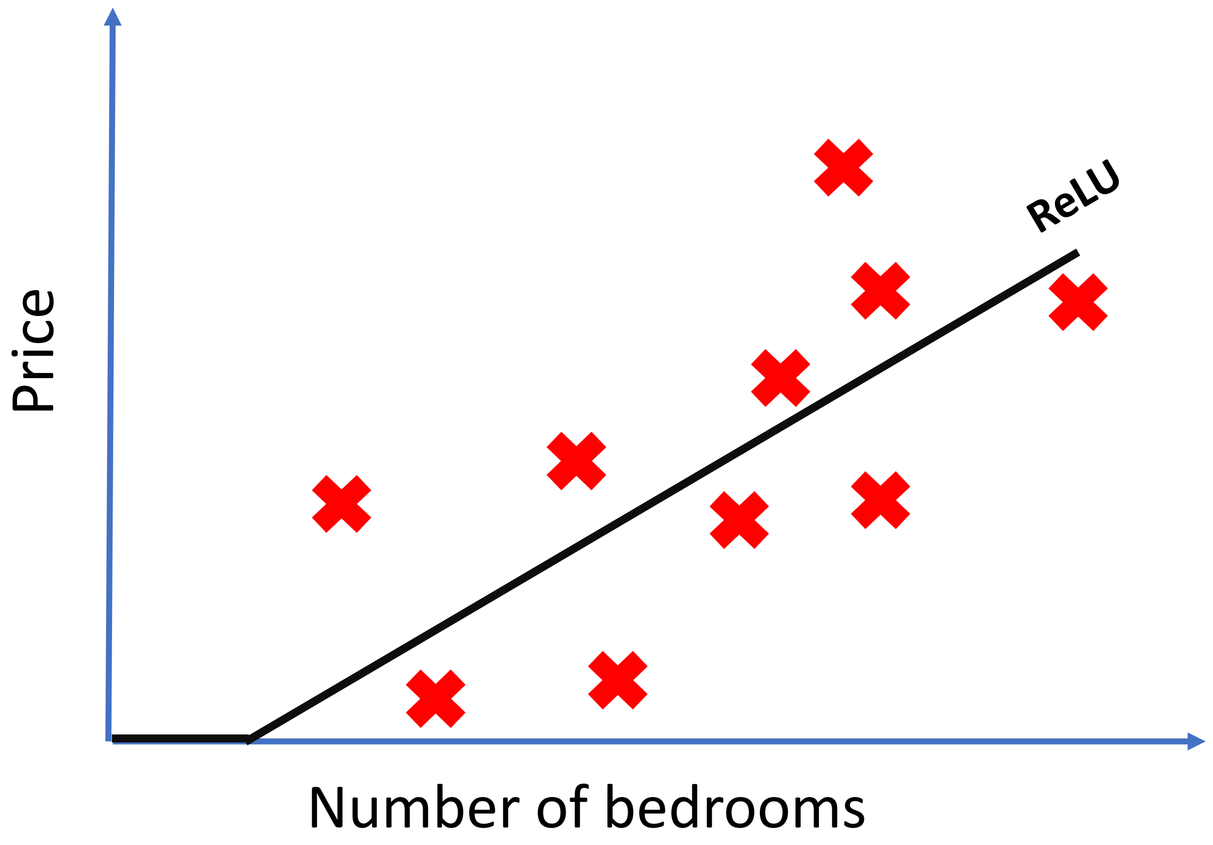
Examples of Supervised Learning Applications

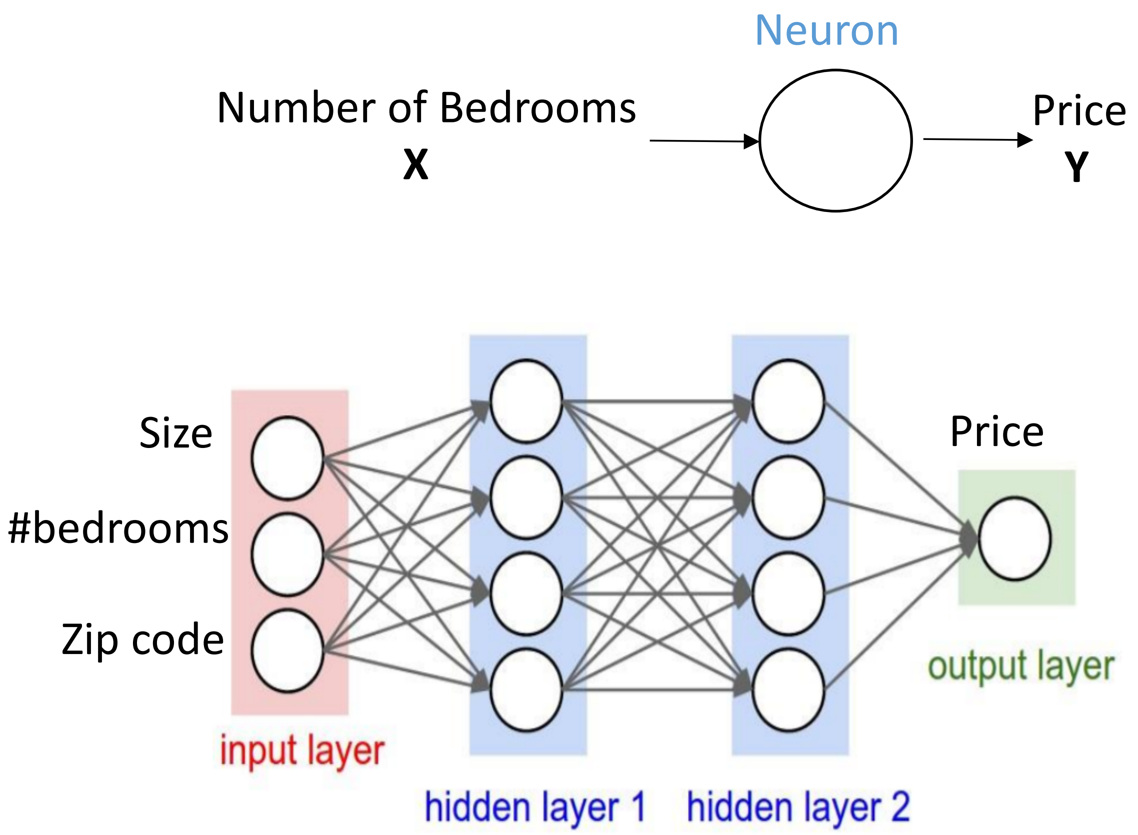


- Typical neural network: Multi Layer Perceptron (MLP)
- Real state, online Advertising, Health care,…

What is a neural network?

Housing Price Prediction





Examples of real applications related to NL

Input	**Output**	**Application**	**Type of Neural Network**
Images/video	Label of Image	Object detection	CNN
Voice/Sound	Text	Language Translation	RNN
Ad/user info	Click on Ad? (0/1)	Online Marketing	MLP
Home features	Price	Real State	MLP
Image, Radar info	Position of road and other cars	Autonomous driving	combination

Vectorization

What is vectorization?

$$z = w^t x + b$$

Logistic regression

$$\text{w=}\begin{bmatrix} \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \end{bmatrix} \qquad x = \begin{bmatrix} \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \end{bmatrix}$$

Non-Vectorized solution

```
$z = 0$
for $i\ in\ range$ (nx):
  $z += w[i] * x[i]$
$z += b$
```

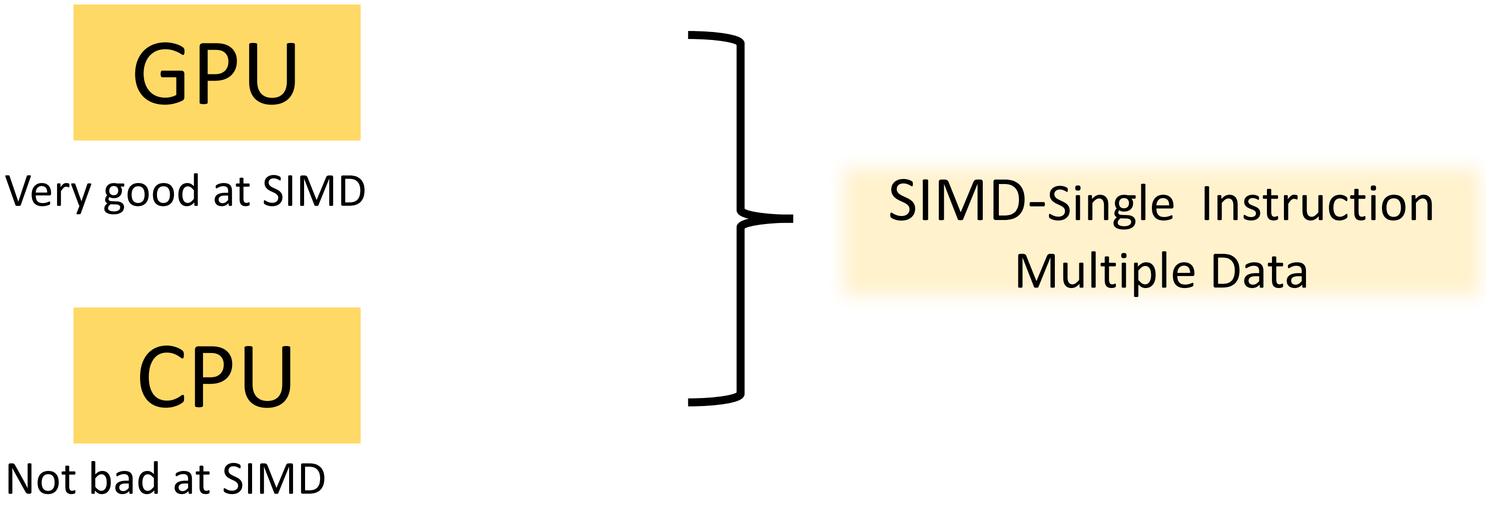
Vectorized solution

$$z = np.dot(w, x) + b$$

Vectorization

Lets look at example Python code!

Vectorization



GPU

Very good at SIMD

CPU

Not bad at SIMD

SIMD-Single Instruction Multiple Data

Can vectorization only be done using GPU?

- ❑Yes
- ❑NO

Broadcasting in Python

Crimes from Burglary, Robber, Assault in 100 randomly selected examples of different cities:

$$\begin{array}{cc} & \begin{array}{cccc}\text{Chicago} & \text{Boston} & \text{NY} & \text{LA}\end{array} \\ \begin{array}{c}\text{Burglary}\\ \text{Robber}\\ \text{Assault}\end{array} \Bigg\downarrow & \begin{bmatrix}56 & 20 & 50 & 46\\ 8 & 10 & 52 & 41\\ 5 & 2 & 15 & 8\end{bmatrix} \end{array} = \underset{(3\,,4)}{A}$$

Calculate the percentage of crime for each city, Can you do that without a for loop? Lets look at a sample code!

```
crime = A.sum(axis = 0)
percentage = 100*A/crime.reshape(1,4)
```

Broadcasting

$$\begin{bmatrix}1\\2\\3\\4\end{bmatrix} + \begin{bmatrix}100\\100\\100\\100\end{bmatrix}\cancel{100} = \begin{bmatrix}101\\102\\103\\104\end{bmatrix}$$

Broadcasting

$$\underset{(n\,,m)}{\begin{bmatrix}1 & 2 & 3\\4 & 5 & 6\end{bmatrix}} + \underset{\begin{bmatrix}100 & 200 & 300\\100 & 200 & 300\end{bmatrix}\;(m\,,n)}{\cancel{\begin{bmatrix}100 & 200 & 300\end{bmatrix}}\;(1\,,n)} = \begin{bmatrix}101 & 202 & 303\\104 & 205 & 306\end{bmatrix}$$

Bradcasting

$$\underset{(n\,,m)}{\begin{bmatrix}1&2&3\\4&5&6\end{bmatrix}} \quad + \quad \cancel{\begin{bmatrix}100\\200\end{bmatrix}}\ (m\,,1) \quad = \quad \begin{bmatrix}101&102&103\\204&205&206\end{bmatrix}$$

$$\begin{bmatrix}100&100&100\\200&200&200\end{bmatrix}(m\,,n)$$

Notes on Broadcasting/Numpy

Advantages of using Broadcasting:

- Makes the code efficient and flexible
- Faster in processing time
- Abstract and easy to use

Disadvantages:

- It can create errors with type and dimension if not carefully used in the code
- There are bugs!

Lets look at an example code!

Anaconda/Python and Keras

https://github.com/nhanteh/machine-learning-using-Python