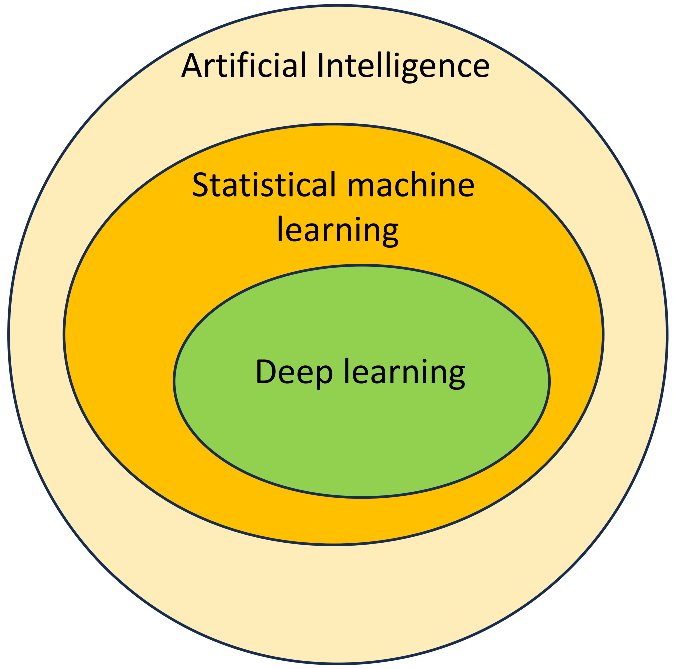
Machine Learning for Robotics: **Introduction**

Prof. Navid Dadkhah Tehrani

<u>Artificial Intelligence (AI):</u>
Born at a workshop at Dartmouth college in 1956. It is simulation of human intelligence processes by machines, especially computer systems, enabling them to perform tasks such as learning, reasoning, problem-solving, and understanding natural language.

<u>Machine Learning (ML):</u>
deals with the statistical inference problem of finding a predictive function based on data.
<u>Deep learning :</u>
make use of the deep neural network as opposed to using classical machine learning algorithms (e.g. k-mean clustering, decision tree, linear regression, …)

<u>Artificial General Intelligence (AGI):</u>
Multi-purpose AI mimicking human intelligence across a variety of tasks as opposed to narrow AI that's only specialized in one task.



Why Deep Learning?

The performance of the deep learning models get better as we feed them more data.

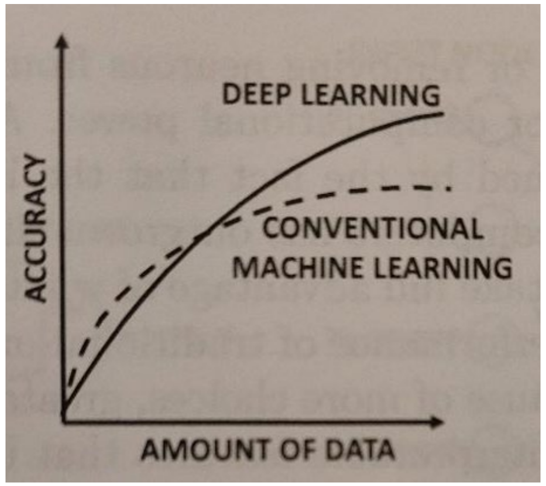


Image from: C. Aggarwal, Neural Networks and Deep learning

Some applications of Deep Learning in Robotics

- Image classification
- Image semantic segmentation
- Depth estimation from camera
- Object detection.
- Object tracking.
- Model prediction
- Reinforcement learning
- Imitation Learning
- Fault/Anomaly detection :
  can be employed for detecting anomalies or faults in robotic systems
  by capturing the normal relationships and dependencies within the system
  and identifying deviations from the norm.

Non robotics applications:

- Email spam detection
- Fingerprint identification
- Stock market prediction
- Face detection
- Reading checks at bank’s ATM machine
- Amazon’s product recommendation
- Netflix movie recommendation
- medical diagnosis
- Siri or Alexa speech recognition
- Large Language models
- Text translation

Datasets

Researchers in each field in deep learning make their own dataset and often time release it to the public so more researcher can develop algorithms on the data set.

Some of the famous datasets:
(pretty much a dataset exist nowadays for anything you want to do!).

- ImageNet: 14 million images, annotated in 20,000 categories
  https://www.image-net.org/
- CIFAR-10 (CIFAR-100): consists of 60k, 32x32 color images in 10 (100) classes.
  https://www.cs.toronto.edu/~kriz/cifar.html

- COCO: 330K images, 1.5M objects, annotated in 91 categories
  https://cocodataset.org/

- nuScences: 1,000 driving scenes, containing 1.4M images, 1.4M bounding boxes and 390K lidar sweeps
  https://www.nuscenes.org/nuscenes

- Kitti**:** 100,000 images across multiple hours of driving. 7.5K in object detection benchmark
  https://www.cvlibs.net/datasets/kitti/

Synthetic datasets:

They are generated by running high fidelity simulation of vehicles with high quality graphic engines such as Unreal engine or Unity.

Mid-Air dataset: 79 minutes of drone flight in simulator with stereo camera and IMU/GPS.

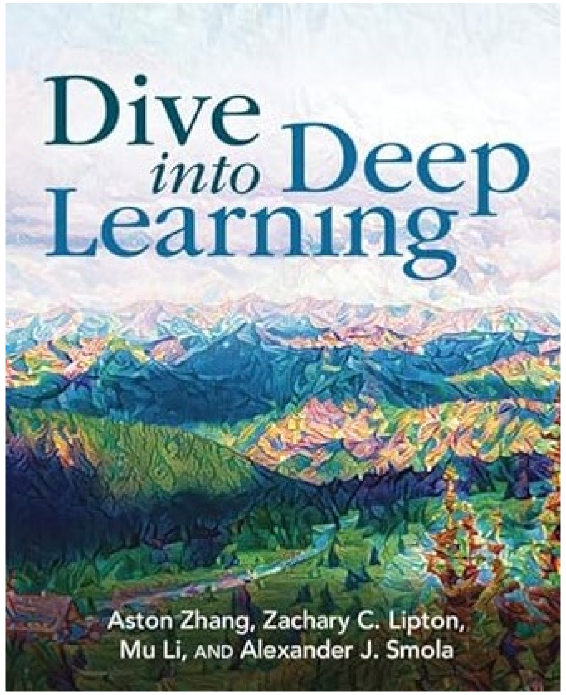
https://midair.ulg.ac.be/

Reading Material for the Course:

This course focuses 100 percent on deep learning. We do not cover classical machine learning.

Textbook: Deep dive into deep learning, by A. Zhang, Z. Lipton, M. Li , A. Smola, 2023

Also available online via https://d2l.ai/


Dive into Deep Learning
Aston Zhang, Zachary C. Lipton,
Mu Li, AND Alexander J. Smola

The book covers the basic material that are prerequisites for advanced concepts.

For robotics application we mainly use research paper.

Tools used in this course:

- Python
- Pytorch → https://pytorch.org/tutorials/
- Numpy
- Matplotlib

PyTorch Released in 2017

Pytorch API is closer to Numpy. Based on trend in the recent research papers, it seems that the deep learning community likes Pytorch better.

Recommended IDE: Visual Studio Code Visual Studio Code

multi-platform, multi-language support: Python, Julia, C/C++, etc,
powerful debugger.

Tensor

Tensors are basically high dimensional arrays.
In deep learning everything is done with tensors.
All the data is first converted to tensors, and then goes into a neural networks.
Neural network itself is of collection of tensors with various operation such addition, multiplication, normalization, nonlinearity, ….
The output is a tensor/tensors of desired output.

Q: why can’t we do all of this in numpy?
A: Pytorch has GPU support, which means that we can load the data and neural network in the GPU and take advantage of parallel processing.

GPUs have parallelization for linear algebra operations.
Also Pytorch has automatic differentiations which is used in training of neural networks.

Q: Since Pytorch is in in Python, isn’t it too slow?
A: The core functionality of Pytorch is written in C++, with python binding (i.e. C++ backend).
And the parts that are in native Python are written very efficiently.

According to a study, Pytorch is only responsible for about 10 percent of the slowdowns. Which means that even if we use a full C++ API we might only get 10 percent performance improvement.
This not substantial compared to difficulty of working with C++ API.

What is Cuda:
Cuda is a C++ library for Nvidia GPUs.

Similarly ROCm is GPU library for AMD processors. And arm compute library has GPU support for Arm processors.

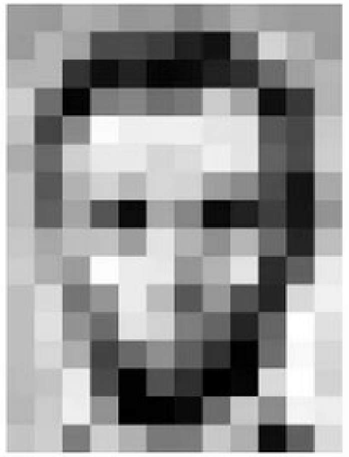
Cuda is the most popular one, but other companies are also trying to get into this market.

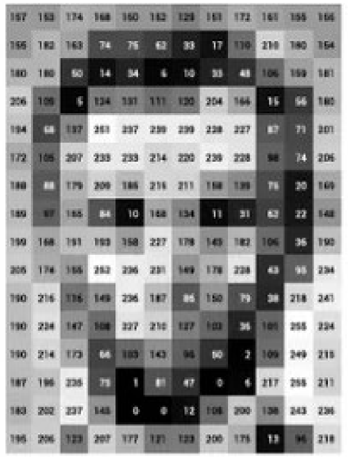
Rank 0 tensor: scalar

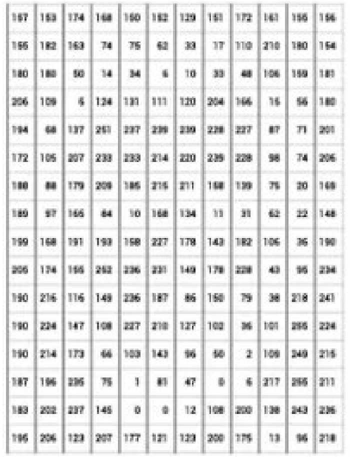
Rank 1 tensor: vector

Rank 2 tensor (2D tensor):

Gray scale image:

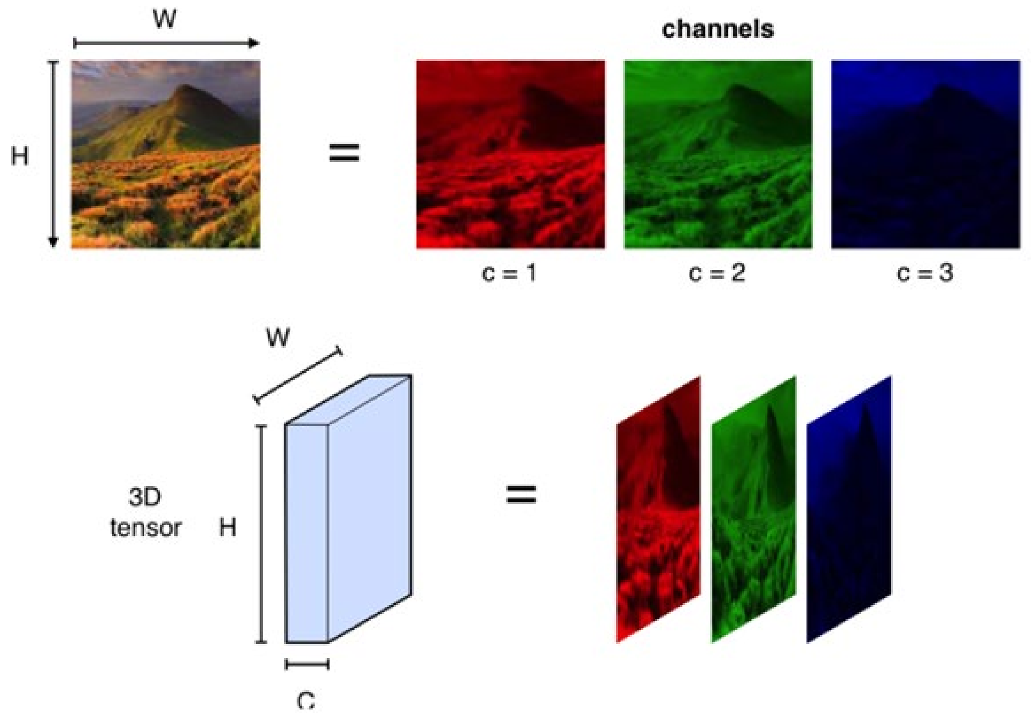






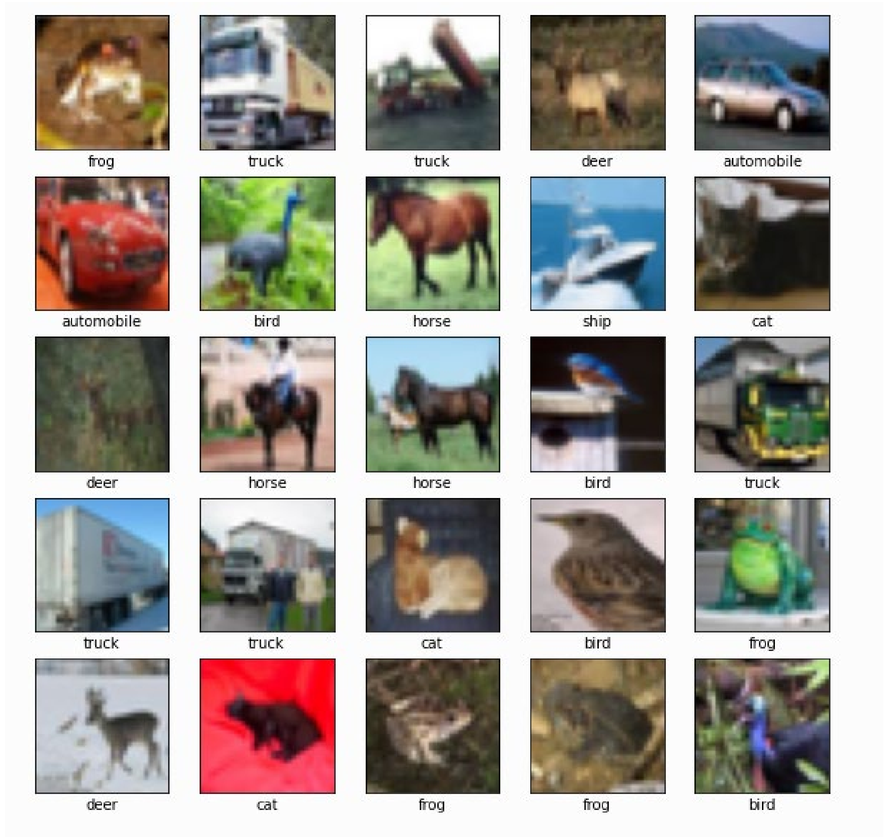
157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	86	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	95	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218

Rank 3 tensors (3D tensors)



Rank 4 Tensor (4D tensor)

Batch of RGB images:



From CIFAR-10 dataset