

AI and PyTorch

What is PyTorch?

↳ a ML library used for Computer vision, deep learning research, and natural language processing

↓
allows computers to understand, interpret, and generate human language

What is AI?

↳ Computer Systems designed to perform tasks that typically require human intelligence (learning, problem-solving, decision-making, and understanding language)

** If you can just write some software rules/instructions to solve the problem do that, use ML for more complex scenarios

When to use deep learning:

- Problems with long lists of rules
- Continually changing environments
- Discovering insights within large collections of data

Machine Learning vs Deep learning

↓

- Structured data

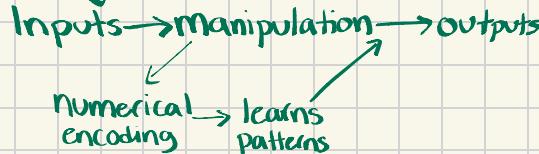
- Algorithm: gradient boosted machine

↳ combines weak models to create a strong predictive model

↓

UNSTRUCTURED data

Algorithm: neural networks



** Use creativity to determine what to use in your system

AI field that allows computers to 'see', interpret, and derive meaning from images/videos

↓

developing and applying multi-layered Neural networks to learn complex patterns from raw data

↓

Computing systems inspired by the structure and function of the human brain, using a layered network of interconnected nodes to process data and learn patterns

Types of Learning

- ↳ Supervised learning: data and labels
- ↳ Unsupervised & Self-supervised learning: just data
- ↳ Transfer Learning: takes a learned pattern and transfers it to another model

PyTorch

Why use it?

↳ helps you run your ML code accelerated on a GPU / TPU

↓
Graphics Processing Unit
Used to crunch numbers very fast (think of video game graphics)
→ you need a lot of numerical calculations)

↓
Tensor Processing Unit

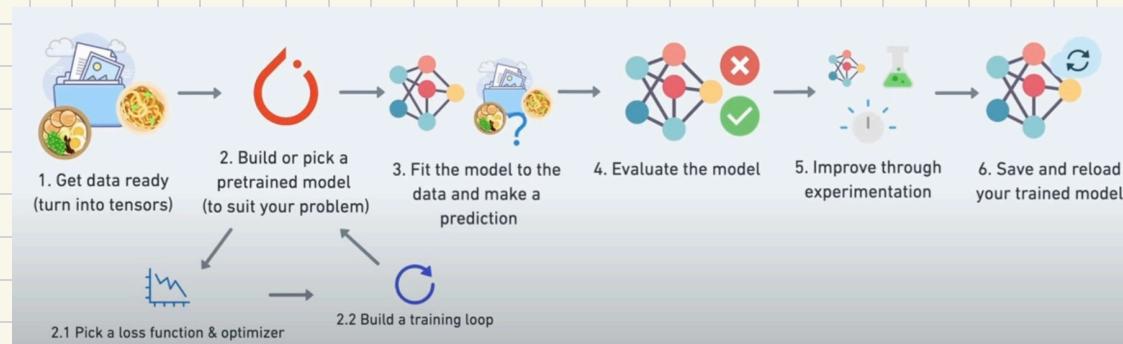
encoded data in some numerical representation
input → tensor → pattern → tensor → output

* PyTorch allows you to leverage a GPU through an interface CUDA

↓
Nvidia parallel computing platform and API (application programming interface) that allows software to use the GPU

environments where multiple processors work simultaneously to solve problems faster by dividing tasks and executing them concurrently on hardware

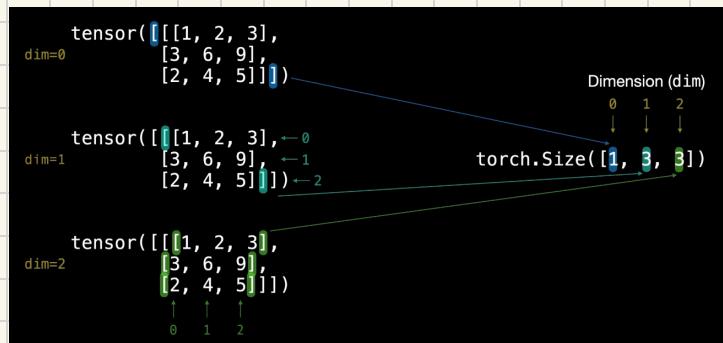
PyTorch Workflow



Coding With PyTorch

* see github google collab files for practice
Code and notes

* `torch.Tensor(#)` is a multidimensional matrix containing elements of a single data type



Name	What is it?	Number of dimensions	Lower or upper (usually/example)
scalar	a single number	0	Lower (a)
vector	a number with direction (e.g. wind speed with direction) but can also have many other numbers	1	Lower (y)
matrix	a 2-dimensional array of numbers	2	Upper (q)
tensor	an n-dimensional array of numbers	can be any number, a 0-dimension tensor is a scalar, a 1-dimension tensor is a vector	Upper (x)

Random Tensors

Starts with random #s → looks at data → updates random nums → looks at data → updates random

Tensor Datatypes