**PyTorch for Deep Learning Bootcamp**

**1. Introduction**

Machine learning is **turning things (data) into numbers** and **finding patterns** in those numbers. The computer does the finding of patterns with code and machine, we write the code. In the sphere which is artificial intelligence there is machine learning which contains deep learning.

Traditional programming: Inputs + Rules = Output (MAKES)

Machine learning algorithm: Inputs (featurs) + Outputs (label) = Rules (FIGURES OUT)

Why use machine learning (or deep learning)?

Why not? Or, for a complex problem can you think of all the rules?

Probably not.

If you can build a **simple rule-based** system that doesn’t require machine learning, do that.

What deep learning is **good** for?

**Problems with long lists of rules**, when the traditional approach fails, machine/deep learning may help.

**Continually changing environments**, deep learning can adapt (‘learn’) to new scenarios.

**Discovering insights within large collections of data**, can you imagine trying to hand-craft rules for what 101 different kinds of food look like?

What deep learning is **not good** for (typically)?

**When you need explainability,** the patterns learned by a deep learning model are typically uninterpretable by a human.

**When the traditional approach is a better option**, if you can accomplish what you need with a simple rule-based system.

**When errors are unacceptable**, since the outputs of deep learning model aren’t always predictable.

**When you don’t have much data**, deep learning models usually require a fairly large amount of data to produce great results (typically, not always).

For **structured data** (like tables with rows and columns) you use **machine learning**, it uses a **gradient boosted machine** as an algorithm,and for **unstructured data** (images, audio) you use **deep learning,** which uses neural networks for algorithm.

With PyTorch we create neural networks like: **fully connected neural networks** and **Convolutional Neural Networks** (**CNN**). In some cases you can use both.

What are neural networks?

We have **inputs** (images, text or audio), **numerical encoding**, before data gets used with a neural network it needs to be turned into numbers (tensors), **learns representation**, choose the best neural network for your problem(patterns/features/weights), **rappresentation outputs** (tensors) and **outputs** (labels).

**Anatomy of neural networks**

**Input layer**, data goes here, **hidden layer(s)**, learns patterns in data, **output layer**, outputs learned representation or prediction probabilities.

Each layer is usually combination of **linear (straight line)** and/or **non-linear (not straight line)** functions.

**Types of learning**

**Supervised learning** (data and labels)**, unsupervised learning & self-supervised learning** (just data) and **transfer learning** (learns from patterns of other models)**.**

What is deep learning actually used for?

Use cases: **recommendation,** then **translation** and **speech recognition** are part of the **sequence to sequence (seq2seq),** and **computer vision** plus **natural language processing (NLP)** are part of the **regression** and **classification** respectively**.**

What is PyTorch?

Is the most popular deep learning framework

Write fast deep learning code in python (able to run on a GPU/many GPUs)

Able to access many pre-built deep learning models (Torch Hub/torchvision.models)

Whole stack: preprocess and model data, deploy model in your application/cloud

Originally designed and used in-house by Facebook/Meta (now open-source and used by companies such as Tesla, Microsoft and OpenAI).

What is a tensor?

They are the result of numerical encoding of the inputs and result of the neural network before labeling them, they are like an array of numbers.