Chapter 1

Introducing deep learning and the PyTorch Library

To execute succesful deep learning:

- We need a way to ingest whatever data we have at hand.
- We somehow need to define the deep learning machine.
- We must have an automated way, training, to obtain useful representations and make the machine produce desired outputs.

Tensor: A multidimensional array that shares many similarities with NumPy Arrays

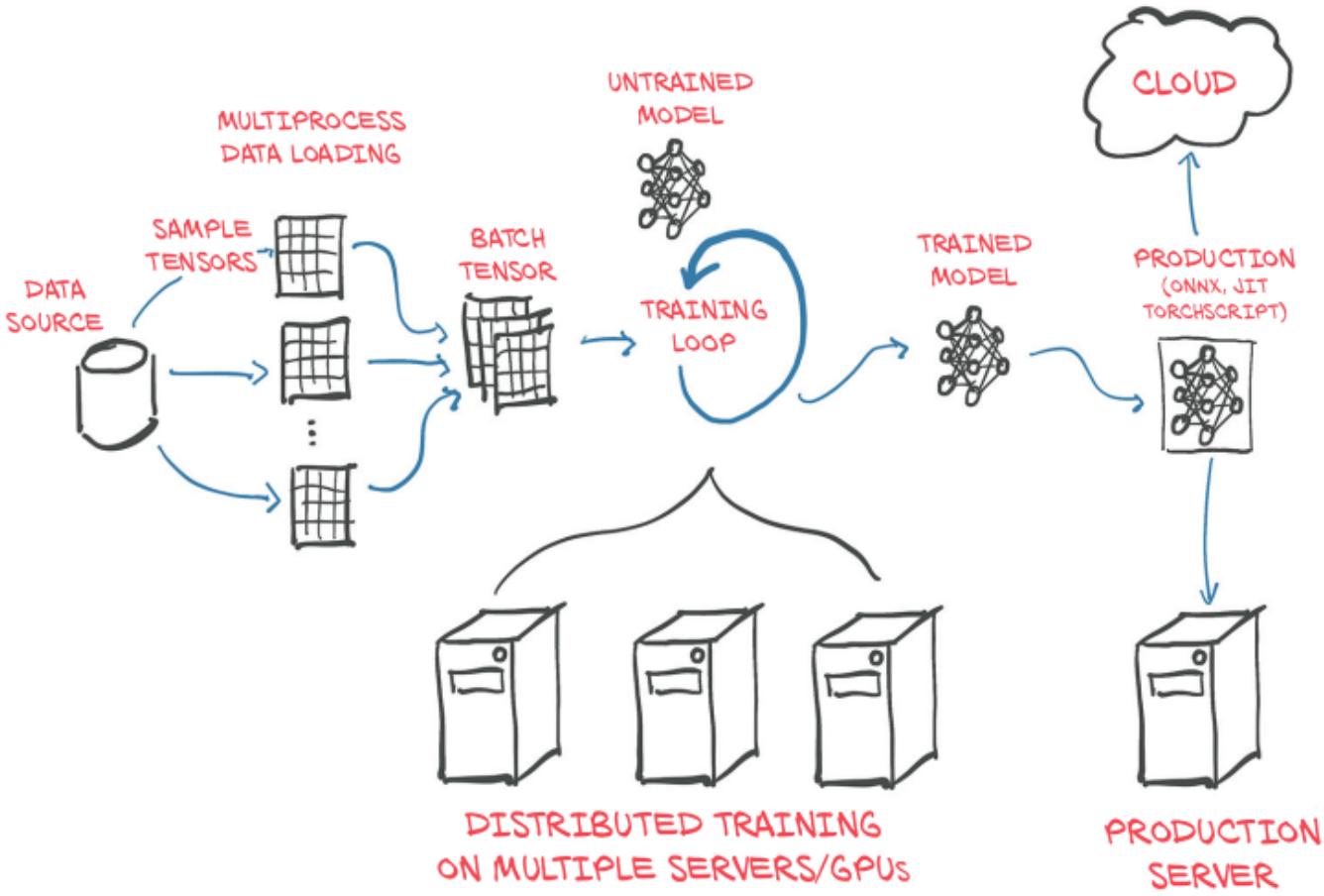

Figure 1.2 Basic, high-level structure of a PyTorch project, with data loading, training, and deployment to production

`torch.nn.parallel.Distributed-DataParallel` and the `torch.distributed` submodule can be employed to use the additional hardware, if necessary.

Using TorchScript, PyTorch can serialize a model into a set of instructions that can be invoked independently from Python: say, from C++ programs or on mobile devices.

DAWNBench (https://dawn.cs.stanford.edu/benchmark/index.html)

- Provides benchmarks on training time and cloud computing costs related to common deep learing tasks on publicly available datasets.

Google Colaboratory (https://colab.research.google.com)

- Cloud platform, offer GPU-enabled Jupyter Notebooks with PyTorch preinstalled (free quota)

We are going to be making heavy use of Jupyter Notebooks for our example code

Full working code for all listings from the book can be found at:

- The book's website (www.manning.com/books/deep-learning-with-pytorch)
- Our repository in github (https://github.com/deep-learning-with-pytorch/dlwpt-code)