It’s a Python-based scientific computing package targeted at two sets of audiences:

A replacement for NumPy to use the power of GPUs

a deep learning research platform that provides maximum flexibility and speed

Getting Started

In PyTorch, you define the graph as a class of type nn.module, and feed the input data through it. The code runs as the class is called. This has several advantages as well, because the code is easier to read and intuitive, and because of its runtime-execution model, it is easy to debug the code as the data passes through the model.

To train images, we use a convolutional neural network (CNN). The "neocognitron" was introduced by Kunihiko Fukushima in 1980. The neocognitron introduced the two basic types of layers in CNNs: convolutional layers, and downsampling layers. A convolutional layer contains units whose receptive fields cover a patch of the previous layer. The weight vector (the set of adaptive parameters) of such a unit is often called a filter. Units can share filters. Downsampling layers contain units whose receptive fields cover patches of previous convolutional layers. Such a unit typically computes the average of the activations of the units in its patch. This downsampling helps to correctly classify objects in visual scenes even when the objects are shifted.

An early model for visual pattern recognition applied to handwritten numerals is the *neocognitron*, developed in the 1980s by Kunihiko Fukushima. Building on the neurological pattern recognition research of neuropsychologists David Hubel and Torsten Wiesel, the neocognitron is a multilayered neural network model that mimic’s the brain’s own visual pattern recognition structure. The necognitron works in a series of layer pairs. Each layer pair consists of a *simple cell layer* responsible for recognizing pattern segments and a *complex cell layer* that handles pattern deformations and shifts.[[1]](#footnote-1) The initial layer pair uses digit images as input. Each subsequent layer pair then uses the output of the previous layer pair as input.

1. In 1989, during his Computer Science MS studies at Saint Joseph’s University, the author implemented the neocognitron in C on a Sun Microsystems workstation and proposed a design for parallelizing the neocognitron on a single instruction stream, multiple data stream (SIMD) architecture. [↑](#footnote-ref-1)