# Lesson Overview

Welcome! In this lesson, you'll learn how to use PyTorch for building deep learning models. PyTorch was released in early 2017 and has been making a pretty big impact in the deep learning community. It's developed as an open source project by the Facebook AI Research team, but is being adopted by teams everywhere in industry and academia. In my experience, it's the best framework for learning deep learning and just a delight to work with in general. By the end of this lesson, you'll have trained your own deep learning model that can classify images of cats and dogs.

I'll first give you a basic introduction to PyTorch, where we'll cover **tensors** - the main data structure of PyTorch. I'll show you how to create tensors, how to do simple operations, and how tensors interact with NumPy.

Then you'll learn about a module called **autograd** that PyTorch uses to calculate gradients for training neural networks. Autograd, in my opinion, is amazing. It does all the work of backpropagation for you by calculating the gradients at each operation in the network which you can then use to update the network weights.

Next you'll use PyTorch to build a network and run data forward through it. After that, you'll define a loss and an optimization method to train the neural network on a dataset of handwritten digits. You'll also learn how to test that your network is able to generalize through **validation**.

However, you'll find that your network doesn't work too well with more complex images. You'll learn how to use pre-trained networks to improve the performance of your classifier, a technique known as **transfer learning**.

If you'd like to work through the notebooks on your own machine or otherwise outside the classroom, you can find the code in this repo.

See you in the lesson!