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**.

Follow along with the videos and work through the exercises in your own notebooks. If you get stuck, check out my solution videos and notebooks.

Get the notebooks

All the notebooks for this lesson are available from our deep learning repo on GitHub. You can find the files in the repo here:

```
https://github.com/udacity/deep-learning-v2-pytorch
```

Follow along in your notebooks to complete the exercises. I'll also be providing solutions to the exercises, both in videos and in the notebooks marked `(Solution)`.

Dependencies

These notebooks require PyTorch v0.4 or newer, and torchvision. The easiest way to install PyTorch and torchvision locally is by following the instructions on the PyTorch site. Choose the stable version, your appropriate OS and Python versions, and how you'd like to install it. You'll also need to install numpy and jupyter notebooks, the newest versions of these should work fine. Using the conda package manager is generally best for this,

```
conda install numpy jupyter notebook
```