



















› Course Shortcuts

› Student Lounge

› Q&A

› Explore a Neural Network

▼ Use Convolutional Neural Networks for Image Analysis

-  Module Introduction: Use Convolutional Neural Networks for Image Analysis
-  Neural Networks for Images
-  Formalize Convolution
-  Review Convolutional Networks
-  Define the Convolutional Operation
-  Explore Image Convolution
-  Identify Patterns in Images
-  Explore Pooling Layers
-  Visualize a Convolutional Network in Action
-  Stabilize Training with Batch Normalization
-  Formalize Batch Normalization
-  Preserve Information with Residual and Dense Networks
-  ConvNet Cheat Sheet
-  PyTorch Tutorial
-  Implement a Convolutional Network with PyTorch
-  Module Wrap-up: Use Convolutional Neural Networks for Image Analysis

› Analyze Sequence Data with Neural Sequence Models

› Course Resources

Deep Learning and Neural Networks >

## Review Convolutional Networks

Images are one of the most common modes of data. Making sense of images is one of the fundamental challenges of artificial intelligence. Popular applications include visual object recognition (e.g., of traffic signs in self-driving cars) or face recognition.

However, image data is very high dimensional. For example, a 1 megapixel image contains 3 million numbers. Images are usually represented as a  $3 \times \text{height} \times \text{width}$  (or  $\text{height} \times \text{width} \times 3$ ) tensor (or 3D matrices). The dimension with 3 elements is known as the color channel, and for a usual image we have 3 color channels for red, green, and blue. The other two dimensions capture the coordinates of each pixel in the image. For gray-scale images, we have a  $\text{height} \times \text{width}$  or  $\text{height} \times \text{width} \times 1$  image, since the color of the pixel can be captured by a single number describing its darkness.

If we have an image of size  $3 \times 1000 \times 1000$ , then this image essentially has 3 million dimensions ( $3 \times 1000 \times 1000$ ). A naive multilayer perception (i.e., plain neural network without convolutional layers) trained on such images will have many weights and nodes, so will need a huge number of photos to train it.

Not only are images high dimensional, but neighboring pixels in an image are also highly correlated. More concretely, an image pixel does not really contain a lot of information, but a block of image pixels is often much more meaningful. To understand this, consider an image of a tiger. Looking at a single pixel does not give you any information about the image, but by looking at a block of images (e.g., several neighboring pixels that make a patch of fur with orange and black stripes), you can guess that the image contains a tiger. If we vectorize an image before inputting it into a multilayer Perceptron, we end up with one long 1 million dimensional vector and essentially lose much of the spatial relation between neighboring pixels.

This is where convolutional neural networks (ConvNets) come into play. On a high level, ConvNets try to learn small filters that can detect small regional patterns at the first few layers. The vector output of these layers might have higher values if a particular pattern is present, and lower otherwise. In later layers, these outputs are passed into multilayer Perceptrons for classification.

In this module, we will explore different components of ConvNets and how we can piece them together to develop a special type of neural network that is fascinatingly well-suited for image data.

### ☆ Key Points

Images are represented as 3-dimensional tensors, of dimensions  $\text{height} \times \text{width} \times 3$ , where the last dimension captures the color channels.

Convolutional neural networks have the ability to filter local regions of images to recognize patterns.

◀ Previous

Next ▶