# Convotional Neural Networks

Khanh Nguyen

June 22, 2015

Convolutional neural networks (CNN) is one of the major successes that initiate the resurrection of neural networks in the last decade. The model revolutionizes the traditional framework in computer vision, which heavily depended on manually feature-engineering. CNNs automate feature-engineering and only takes raw image pixels as input but still outperforms traditional kernel-based models (e.g. SVM) by large margins. Particularly, CNNs participated in the ImageNet object recognition contest in 20XX and surprised the community by reducing the error rate a incredible XX%.

## 1 Motivation

Images are extremely high-dimension objects if we consider each pixel to be a feature. Using a fully-connected neural networks for an image would significantly increase the amount of computation. More seriously, such a model would have a gigantic set of paramters and thus is very susceptible to overfitting. To address this problem, CNNs are designed as sparsely connected neural networks. Each network layer is partitioned into a set of feature dectectors, often called *filters*. The input image can be considered as a filter itself. For RBG images, we have three filters per image, corresponding to values for red, green, and blue. The numbers of filters of the hidden layers are determined by the model. Those feature dectectors will act on every possible position of the images and produce *feature maps*, a mapping from positions to feature values. To compute the feature values, CNNs employs a special operation called *convolution*. For each position, a filter is convolved with the corresponding region of the image. The convolution operation is the main difference between CNNs and regular neural networks that operate on matrix multiplication. However, using a small set of filters is not enough to reduce the complexity of the model. A special technique, called *pooling*, is required to downsample the feature maps to map the network even sparser. Pooling is simply a mean of aggregating features in a local neighborhood.

Both derivation of the convolution and pooling operations can be derived easily. Therefore, CNNs can be train efficiently with backpropgation.

## 2 Convolution

Convolution is originally a mathemetical term that denotes this operation:

$$f * g = \int_{-\infty}^{\infty} f(t)g(x-t) \tag{1}$$

where $t$ is size of the convolution window.

Note that convolution is commutative, i.e. $f * g = g * f$. For 2D functions, convolution is defined as follows:

$$f * g = \int_{-\infty}^{\infty} f(x,y)g(m-x,n-y) \tag{2}$$

where $m, n$ are dimensions of the convolution window.

In the context of CNNs, we can imagine $f$ as the filter, $g$ as the image and $(m, n)$ as a position on the image's 2D grid. The filter acts as a sliding window that goes across the image (in both dimensions) and convolves with the correponding subimage. Let $k \times k$ be the size of the filter, $w \times h$ be the size the of the image. The convotion of the filter with the (larger) image produces a feature map of size $(w-k+1) \times (h-k+1)$.

## 3 Pooling