Project 0: Convolutional Neural Networks

1 Objective

The ImageNet challenge initiated by Fei-Fei Li (2010) has been traditionally approached with image analysis algorithms such as SIFT with mitigated results until the late 90s. The advent of Deep Learning dawned with a breakthrough in performance which was gained by neural networks. Inspired by Yann LeCun et al. (1998) LeNet-5 model, the first deep learning model, published by Alex Krizhevsky et al. (2012) drew attention to the public by getting a top-5 error rate of 15.3% outperforming the previous best one with an accuracy of 26.2% using a SIFT model. This model, the so-called 'AlexNet', is what can be considered today as a simple architecture with five consecutive convolutional filters, max-pool layers, and three fully-connected layers.

This project is designed to provide you with first-hand experience on training a typical Convolutional Neural Network (ConvNet) model in a discriminative classification task. The model will be trained by Stochastic Gradient Descent, which is arguably the canonical optimization algorithm in Deep Learning.

2 Model

The ConvNet is a specific artificial neural network structure inspired by biological visual cortex and tailored for computer vision tasks. The ConvNet is a discriminative classifier, defined as a conditional (posterior) probability.

$$p(c \mid I; w) = \frac{\exp\left[f_c(I; w) + b_c\right]}{\sum_{c=1}^{C} \exp\left[f_c(I; w) + b_c\right]} \tag{1}$$

where $c \in \{1, 2, ..., C = 10\}$ is the class label of an input image I, and $f_c(I; w)$ is the scoring function for each category, and computed by a series of operations in the ConvNet structure. Figure 1 displays an structure (architecture) of the ConvNet.

1. **Convolution**. The convolution is the core building block of a ConvNet and consists of a set of learnable filters. Every filter is a small receptive field. For example, a typical filter on the first layer of a ConvNet might have size 5x5x3 (i.e., 5 pixels width and height, and 3 color channels). Figure 2 illustrates the filter convolution.

2. **Pooling**. Pooling is a form of non-linear down-sampling. Max-pooling is the most common. It partitions the input image into a set of non-overlapping rectangles and, for each such sub-region, outputs the maximum.

3. **Relu**. Relu is non-linear activation function $f(x) = max(0, x)$. It increases the nonlinear properties of the decision function and of the overall network without affecting the receptive fields of the convolution layer.

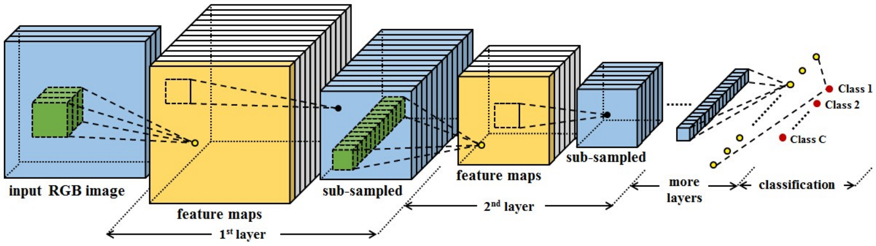

Figure 1: A ConvNet consists of multiple layers of filtering and sub-sampling operations for bottom-up feature extraction, resulting in multiple layers of feature maps and their sub-sampled versions. The top layer features are used for classification via multi-nomial logistic regression

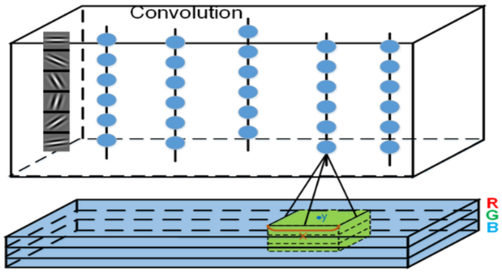

Figure 2: Filter convolution. Each node denotes a 3-channel filter at specific position. Each filter spatially convolves the image.

LeNet shown in Table 2 is a typical structure design taylored for the CIFAR-10 dataset. The Block1 has 32 filters and the filter size is 5x5x3. The Block5 is a logistic regression layer, in which the number of filters is the number of classes and the filter size should be the same as the output from Block4. The CIFAR-10 dataset consists of $60,000$ 32×32 color images in 10 classes, with $6,000$ images per class. There are $50,000$ training images and $10,000$ testing images. Figure 3 shows example images from 10 categories. The objective of classification is to predict the category of each image.

Block1	Block2	Block3	Block4	Block5
Conv 5x5x3x32 Pooling 3x3 Relu	Conv 5x5x32x32 Relu Pooling 3x3	Conv 5x5x32x64 Relu Pooling 3x3	Conv 4x4x64x64 Relu	Conv 1x1x64x10 Softmax

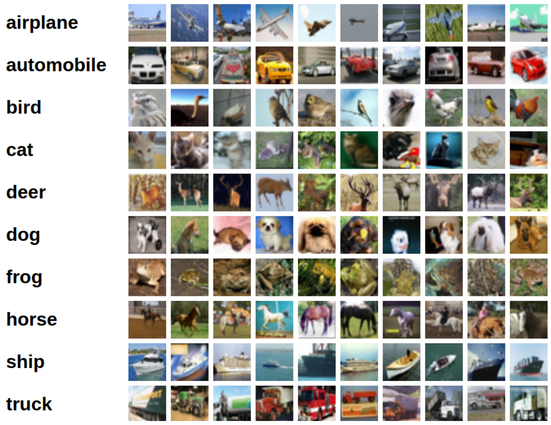

Figure 3: CIFAR-10 Dataset. CIFAR-10 contains 60,000 images in 10 classes.

3 Assignment

In this project, you will train variations of a LeNet in TensorFlow or PyTorch:

1. Optimization of LeNet.
   (a) Complete the functions $flatten()$, $convnet\_init()$, $convnet\_forward()$.
   (b) Adjust parameters $learning\_rate$, $epochs$ such that test-accuracy $> 70\%$.
   (c) Plot the training loss and test accuracy over epochs in two Figures.
2. Alteration of LeNet.
   (a) Keep the Block5, learn a ConvNet only with:
      i. Block1.
      ii. Block1 and Block2.
      iii. Block1, Block2 and Block3.
   (b) Compare the final test accuraries for (i., ii., iii.) in a Table. (**Hint**: You need to change the filter size in Block5 to match the output from previous block.)
3. Visualization of filters and activations.
   (a) Plot the learned 32 filters of the first convoluational layer in LeNet.
   (b) Plot the filter response maps for a given sample image of CIFAR-10.

Submission: (1) Distill your results into a report as pdf, and, (2) zip your code.