Machine Learning Engineer Nanodegree

Capstone Proposal

Zhenghe Jin
October 18st, 2016

Domain Background

In this project, I would like to work on visual recognition via convolutional neural networks. This classical problem in computer vision, image processing, and machine vision is that of determining whether or not the image data contains some specific object, feature, or activity. Different varieties of the recognition problem are described as object recognition, identification and detection：

- Object recognition (also called object classification) – one or several pre-specified or learned objects or object classes can be recognized, usually together with their 2D positions in the image or 3D poses in the scene. Blippar, Google Goggles and LikeThat provide stand-alone programs that illustrate this functionality.
- Identification – an individual instance of an object is recognized. Examples include identification of a specific person's face or fingerprint, identification of handwritten digits, or identification of a specific vehicle.
- Detection – the image data are scanned for a specific condition. Examples include detection of possible abnormal cells or tissues in medical images or detection of a vehicle in an automatic road toll system. Detection based on relatively simple and fast computations is sometimes used for finding smaller regions of interesting image data which can be further analyzed by more computationally demanding techniques to produce a correct interpretation.

Currently, the best algorithms for such tasks are based on convolutional neural networks. An illustration of their capabilities is given by the ImageNet Large Scale Visual Recognition Challenge; this is a benchmark in object classification and detection, with millions of images and hundreds of object classes. Performance of convolutional neural networks, on the ImageNet tests, is now close to that of humans. The best algorithms still struggle with objects that are small or thin, such as a small ant on a stem of a flower or a person holding a quill in their hand. They also have trouble with images that have been distorted with filters (an increasingly common phenomenon with modern digital cameras). By contrast, the conventional treatment by multilayer perceptrons or SVM rely on and-crafted features instead of learned features through convolutional networks, will be limited by the size of features of the images with raw input pixel information of more than a million. Specifically, our problem will be training a learning model to automatically recognize one or multiple digits within an image. This can be widely applied to large scale problem from identifying hard written digits to recognizing house numbers for unmanned delivery such as Amazon Prime Air.

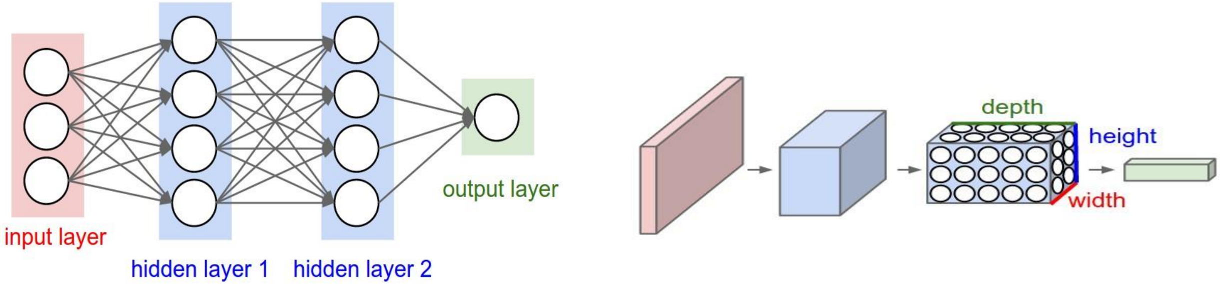

Figure 1: Left: A regular 3-layer Neural Network. Right: A ConvNet arranges its neurons in three dimensions (width, height, depth), as visualized in one of the layers. Every layer of a ConvNet transforms the 3D input volume to a 3D output volume of neuron activations. In this example, the red input layer holds the image, so its width and height would be the dimensions of the image, and the depth would be 3 (Red, Green, Blue channels). [Ref 1]

Problem Statement

Recognizing numbers on images might be accomplished by human eyes with little problem, but it can be of great difficulty for computers, and this is the place when machine learning and convolution neural networks come into play.

The convolution neural network (ConvNet) is a special artificial neural network. Unlike a regular Neural Network, the layers of a ConvNet have neurons arranged in 3 dimensions: width, height, depth. A simple ConvNet is a sequence of layers, and every layer of a ConvNet transforms one volume of activations to another through a differentiable function. Three main types of layers to build ConvNet architectures: Convolutional Layer, Pooling Layer, and Fully-Connected Layer These layers are then stacked to form a full ConvNet architecture. The details of ConvNet are discussed through the Stanford CS231n lecture notes [Ref 1].

Datasets and Inputs

The dataset we will be using to train and test our model are from the SVHN dataset[1], which is a good large scale dataset collected from house numbers in Google Street View. SVHN is a real-world image dataset for developing machine learning and object recognition algorithms with minimal requirement on data preprocessing and formatting. It can be seen as similar in flavor to MNIST (e.g., the images are of small cropped digits), but incorporates an order of magnitude more labeled data (over 600,000 digit images) and comes from a significantly harder, unsolved, real world problem (recognizing digits and numbers in natural scene images). SVHN is obtained from house numbers in Google Street View images.

For each image in the dataset, there are one or multiple numerical digits and each digit below to one class (10 classes for “0” to “9”). The dataset comes in two formats, one is the original images with character level bounding boxes, the other is MNIST-like 32-by-32 images centered around a single character. The dataset is splitted into three sets, train, test and extra. For the project, I will be mainly use train and test for training and testing, while include data from extra set to improve the accuracy. Additional shuffering will be applied for randomizing the inputs.

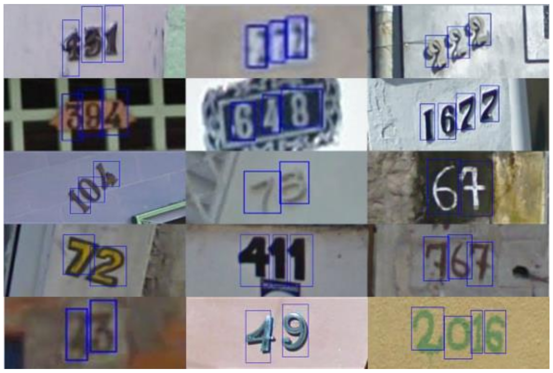

Figure 2: sample image from the SVHN datasets.

Solution Statement

In this project, a 7-layer ConvNet will be implemented, this architecture resembles the classical LeNet [Ref 2], which is the first successful application of ConvNet developed by Yann LeCun in

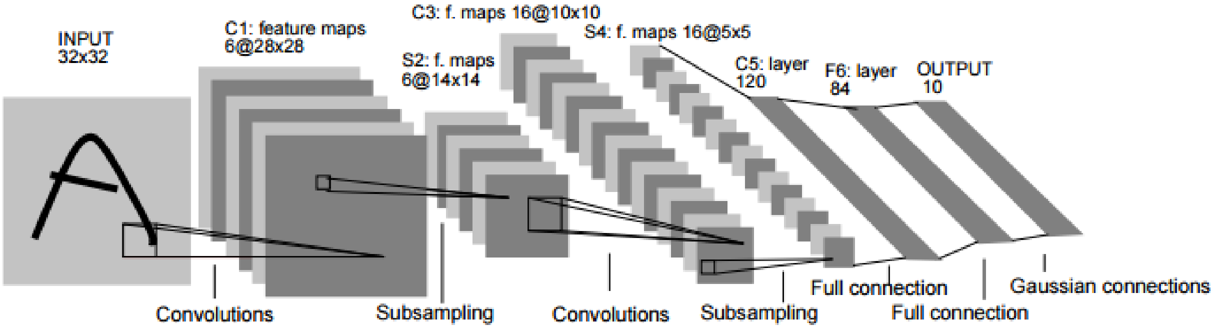

Figure 2: Architecture of LeNet-5, a ConvNet, here for digits’ recognition. [Ref 2]

[1] Details of the SVHN dataset are described at http://ufldl.stanford.edu/housenumbers/

1990's. The LeNet architecture was used to read zip codes, digits, etc. Actual parameters will be changed, in order to achieve better accuracy.

Benchmark Model

In comparison with the convolution neural network for image recognition, there are more direct but less efficient ways such as regular Neural Network or SVM for classification. All these models will be evaluated by the test accuracy as well as time complexity. The hand-crafted features will be the greyscale of the resized image, which will be difficult to learn. There are still some high level features such as the local binary patterns (LBP) which is a powerful feature for texture classification.

Evaluation Metrics

Similar with the evaluation metric for benchmark model, the performance of the ConvNet will be evaluated by the test accuracy as well as time complexity. Scalability can also be important.

Project Design

As a machine learning project with appropriate size of dataset, the project will be spited into two parts. The first is data analysis which transform the dataset into clean data. The second part will be performing the "learning" task for the highest score of accuracy.

For the data cleaning, the raw image will be reshaped to the same size as well as greyscaled since the color of image doesn't affect the output, i.e. the numbers on the image. Training sets and testing sets will be selected and shuffered from the SVHN dataset for the learning.

For the training, the LeNet will be implemented with initial parameters preset by the literature. Then the size of the network will be tuned to increase the training accuracy. Regulation will be introduced and parameter tuned to improve the testing accuracy.

References

1. http://cs231n.github.io/convolutional-networks/

2. LeCun, Yann, et al. "Gradient-based learning applied to document recognition." Proceedings of the IEEE 86.11 (1998): 2278-2324.