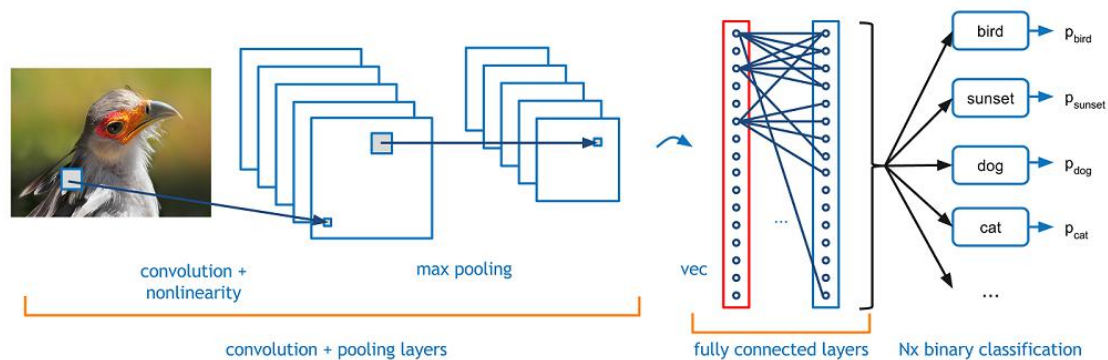


Machine Learning Lab 11

Convolutional Neural Networks

CNNs, like neural networks, are made up of neurons with learnable weights and biases. Each neuron receives several inputs, takes a weighted sum over them, pass it through an activation function and responds with an output. Unlike neural networks, where the input is a vector, here the input is a multi-channelled image and they are commonly applied to image processing tasks. They are also known as shift invariant or space invariant artificial neural networks (SIANN), based on their shared-weights architecture and translation invariance characteristics.



The dataset

For the convolutional neural networks I have performed my experiments on the Non-MNIST dataset, it is a dataset of characters and just like the MNIST dataset it is made of images of characters of the size of 28x28 pixels but the images are of the the english alphabets and not numbers. The images of these letters are in different fonts and the dataset contains 500,000 images in it, it contains 26 letters and each letter consists of 10 classes in itself.

""Download a file if not present, and make sure it's the right size.""

Experiment

In this experiment I implemented a convolutional neural network using tensorflow and applied it to the task of character classification and achieved an accuracy of 89.8%

I started with downloading the Non-MNIST dataset and and parsing through it's directory structure to extract the dataset and find the image. Next I load the image into the memory and preprocess them and save them into a pickle file.

Once the images are stored into a pickle file I load the pickle file and get the image in the form of numpy array from the pickle file. I randomize the data and reshape it to feed the neural network.

Next I create the CNN graph with tensorflow with two convolution layers, followed

by relu and one fully connected layer. The CNN uses the softmax cross entropy loss and a gradient descent optimizer to train itself. With this network I am able to achieve an accuracy of 88.5%, next I increase the stride of the neural network 2 for both the kernel as well as the max pooling layers and make the kernel size as 2 and the accuracy increases to 89.8%.

The code and plots can be found in the accompanying jupyter notebook.