Intro. to Artificial Intelligence (75911) – Spring 2024

Instructor Dr Syed Ali Raza

Project Report

Members:

Shaikh Muhammad Sharjeel - 26932

Abdul Wasay Imran – 27126

14 May 2024

IMAGE CLASSIFICATION USING ANN/MLP

## Problem Statement:

Classifying images from the [CIFAR-10](https://www.cs.toronto.edu/~kriz/cifar.html) dataset, into 10 different categories: airplane, automobile, bird, cat, deer, dog, frog, horse, ship and truck.

## Methodology:

We built a Artificial Neural Network consisting of 4 layers: an input layer (with 1024 inputs), an output layer (with 10 nodes) and two hidden layers (of 16 nodes each), to achieve results as accurate as possible on the hardware we had. We used forward and backpropagation using ReLU as activation function and softmax as output. We only made use of Python’s numpy and math libraries, without relying on pre-built frameworks like Tensorflow and Pytorch.

### Limitations:

Our model is just a basic neural network with few nodes. The CIFAR dataset contains 10 classes which are very different, and to properly classify them, we need a more complex and advanced model, like CNN or RNN. Hence our accuracy is low. Nonetheless we were able to implement what we learned in class, and we got a good understanding of neural networks.

We could have solved this by using a simpler dataset, but we decided to go with CIFAR to see how much we can push ourselves.

## Implementation Details:

### Collecting Data:

We’ve only made use of the training data from the [CIFAR-10](https://www.cs.toronto.edu/~kriz/cifar.html) dataset, which we in turn split for training and testing our model. Then we converted the color images to gray-scaled images. This was done as a prerequisite to our project to gather the data required. For that, a code file (Conversion.py) is attached with submission for conversion of CIFAR-10 dataset into a desired .csv file containing a total of 50000 32x32 grey-scaled images, where each image is stored as a linear array having 1024 values between 0 and 255. (We did not normalize in the csv file because it turned out saving as a floating point number caused file size to increase to 1.2GB compared to 200MB)

### Designing the Neural Network:

We started off with a single hidden layer model of the neural network with various number of nodes. Initially we set the number of nodes to 10 which gave us a very poor accuracy of 15%. Increasing it to 16 and then 32 didn’t improve the results at all. Then we drastically increased it to 512 nodes which showed signs of better accuracy of 17% (a 2% increase) but the available resources didn’t allow for the computation to proceed smoothly.

We then moved on to an alternative solution by adding another hidden layer. We tried a combination of 512 and 256 nodes for the first and second hidden layer respectively, to get an accuracy of 20%. Upon further tweaking our model with two hidden layers, we concluded that it gave the best accuracy of 36% in a practical amount of time for 16 nodes each in the hidden layers with 5000 iterations.

### Results:

We wrapped up our project with the final best accuracy of 36% with a 4-layer Neural Network.

### Plans:

We could use better hardware to train the same model with more nodes and hidden layers. We could also learn and use researched image classification techniques. We can increase the image size from 32x32 to 256x256 for better results. We can also modify the GGUI so that it can work on image outside the dataset.

### Evaluation:

We are confident that we made the most out of what we’ve learned so far and with the resources that were available to us, to achieve results which are beyond satisfactory for a dataset like this.