Capstone Proposal

Machine Learning Nanodegree

Image Recognization using Convolution Neural Network

Domain knowledge **:**

Image recognization is the ability of computer to interpret a thing from images, documents and various other sources. The core idea came from the computer vision(how do a computer recognise its a car or human or dog).Computer Vision has become ubiquitous in our society, with applications in search, image understanding, apps, mapping, medicine, drones, and self-driving cars. Core to many of these applications are visual recognition tasks such as image classification, localization and detection. Recent developments in neural network approaches have greatly advanced the performance of these state-of-the-art visual recognition systems.

GPU-based implementations of this approach won many pattern recognition contests, including the IJCNN 2011 Traffic Sign Recognition Competition,the ISBI 2012 Segmentation of neuronal structures in EM stacks challenge,the ImageNet Competition and others.

Deep, highly nonlinear neural architectures similar to the neocognitron and the "standard architecture of vision", inspired by simple and complex cells, were pre-trained by unsupervised methods by Hinton.A team from his lab won a 2012 contest sponsored by Merck to design software to help find molecules that might identify new drugs.

Reference: www.en.wikipedia.org/wiki/image-recognisation

Problem Statement :

We have to predict what is there in an image. The dataset contains images of frogs,automobiles,horse and 7 others images which are used to train the data. This is a supervised learning problem.It falls under classification problem containing 10 classes(0-9).Our model is tested by its ability to correctly predict different real world and unseen data

Dataset and Inputs :

I have selected the cifar-10 dataset which has 50,000 training data and 10,000 testing data.The digits have been size-normally and spreaded on image .It consists of rgb(color) images and size of the image is 32*32 pixels.

The dataset is available in keras module ,everyone can use it by writing by writing the underneath code

Source Code :

```
from keras.datasets import cifar10
```

We can load the data by load_data() function.It returns two tuples:

X_train,X_test:unit8 grayscale image data with shape(50000,3,32,32).

Y_train,y_test:unit8 array of digit labels(0-9)with shape (10,000,)

Reference: https://keras.io/datasets/

Their names give clear info about what they do.X_train and y_train are used for training and X_test and y_test used for testing and evaluation of our model.

Solution Statement :

I have already mentioned this problems comes under supervised learning and classification in depth,there are lot of algorithms available for classication the data namely NaiveBayes,RandomForest,Logistic regression,decision tree,neural nets and many more.The algorithm I have chosen is Convolution Neural Network .I think its better because there are so many fields in an image and the data set is not too big and inorder to get maximum accuracy I think Neural Nets are the best choice.Neural network is the first choice when we are doing image processing.

Convolution neural network is a type of feed-forward artificial neural network most commonly applied for visual imagery.As our problem belong to visual imagery we can use CNN here.

Reference:https://en.wikipedia.org/wiki/Convolution_neural_network

Important Layers required for the architecture of CNN:

1. **Convolution Layer**: This layer takes an 3d-tensor or 4d tensor as input generally(height,width,channels).This layer applies filters such that one image can be made into some small images(naively) ,The size of the filters gives rise to locally connected structure which are convolved with the image to produce feature maps.
2. **Pooling Layer**: It performs a down sampling operation along the dimensions.It is generally used to reduce the memory.Generally we use Maxpooling .
3. **ReLU Layer**: The rectified linear unit apply elementwise activation function
4. **Dense layer** :It is mainly used for predicting

The model is trained with CNN on the training data and then evaluated against test data.

Benchmark Model :

The benchmark model is chosen from one of the result of (benchmark query of cifar on google) image.The benchmark model solves the same image recognition problem mentioned in this project by using a Simple Vector Machines.It uses the cifar-10 dataset only and it achieved 40% accuracy score for the best SVM.The same result can be measured for our model also and we compare those two.

Reference : https://houxianxu.github.io/implementation/SVM.html

Evaluation Metrics :

Accuracy is measured as ratio of correctly classified samples to the overall samples.The cross-entropy will be used instead of mean squared error.Cross entropy can be used as an error measure when a network outputs can be of as representing independent hypothesis.The node activations can be used for representing probability that our hypothesis is true.The output vector represents a probability distribution and our error measure(Cross entropy) which indicates the distance between what the network believes and what the actual is.Cross-entropy is more advantageous in which the targets are 0 and 1.Cross-entropy tends to allow error to change the weights even when nodes saturate(their derivatives close to asymptotic 0)

Reference:http://www.cse.unsw.edu.au/~billw/cs9444/crossentropy.html

Project Design :

Initially the datset is loaded.The loaded data is explored and visualised using numpy(shape of the datsets) and matplotlib modules (plot an image) to understand data.Exploring and understanding the data will help us how to attack and whether any preprocessing is required for the inputs or not.If preprocessing is required we should do it.Once the input data is clean and perfect,we will built our architecture as discussed in solution statement.After building the architecture it must be compiled.The compiled model is used for training and evaluated using accuracy score with testing data.If we are satisfied with the accuracy we can stop and we can compare with benchmark model.If not then we will refine the model by increasing more layers or even using transfer learning.Now our model is evaluated against completely unseen data(images) to measure its performance.

Workflow

1. Loading the data
2. Exploring and Visualization
3. Preprocessing
4. Building the architecture
5. Compiling and training
6. Refining our model
7. Compiling and training
8. Accuracy Evaluation
9. Performance on unseen data