IIT Bombay

Robotics Assignment

Air Lab

Semantic Segmentation-Supervised SNN

Author:
Pulkit Katdare

Date:
November 30,2015

Abstract

Semantic segmentation (or pixel classification) associates one of the pre-defined class labels to each pixel. Here a Humble effort has been made on my part to semantically classify an image using Machine Learning Algorithms.

1 Introduction

Semantic Segmentation is an active research area in the field of Robotic Vision and has been proven very effective in path planning Algorithms. Conventional Semantic Segmentation Algorithms involves exclusive use of Second or Generation Neural Networks along with Convolution to reduce the size of Image.Here an effort has been made to implement Semantic Segmentation using two methods

1. Implementation using ANN

2.Spiking Neural Networks Model (SNN)

3.Bag of Words Model

The main difference between Regression and Classification is the fact that Classification is used to predict a class and regression is used to predict continous values

Training set: a set of examples used for learning: to fit the parameters of the classifier we would use the training set to find the "optimal" weights with the back-prop rule

Validation set: a set of examples used to tune the parameters of a classifier .We would use the validation set to find the "optimal" number of hidden units or determine a stopping point for the back-propagation algorithm

Test set: a set of examples used only to assess the performance of a fully-trained classifier In the MLP case, we would use the test to estimate the error rate after we have chosen the final model After assessing the final model on the test set.

The Problem is not said to be solved as 99% accuracy depends on the application where it is used.For example in satellite imagery you may not need a 99 % accuracy always but instead in a quadcopter you may.

Supervised Learning in supervised learning the data set is given classifiers to classify whereas in **Unsupervised learning** the data set determines it's classification on it's own. In supervised learning you need dedicated set of data for it's classifiers.Unsupervised learning on the other hand have a high computational cost as it determines unknown classifiers and many a times it may even perform an undesired classification.

Hyperparameters are parameters that are set when the neural network is not even optimised.i e Setting the hyperparamter in equivalent to model selection in a neural network.where as parameters are just the variables around which neural networks work

Generally a Gradient based optimistion is not possible because generally the gradient of the hyperparameter is not available but with the advent of [5] efforts have been made in this direction to implement it.

2 Implementation using ANN

2.1 Motivation

Initially the advent of Artificial Neural Networks started with the advent of LeNet. The Hand Digit Recognition network developed by Prof Yann LeCunn's lab,New York. Here, an effort has been made to understand semantic Segmentation using LeNet for semantic Segmentation.

2.2 Implementation

In the LeNet we have around 60,000 28 X 28 images being trained in the network and then later it is used to predict values of the digits. Here as the number of images are less . We have many small patches of 28 X 28 in the each of the Images and then trained them on the LeNet Architecture

2.3 Implementation and Results

There are basically two approaches tried:
1 One where the whole Image is divided into 32 X 32 patches
In this case the accuracy in the prediction is coming out to be 80.375 %
2 One where we take out as many as 28 X 28 patches in the image
In this case the accuracy in the prediction is 72.6 %
Now Further work in this field is to try to take settled weights of the network and only in the last layer we apply the concept of spiking Neural Networks to make it more adaptive
The network in this case is not robust but the accuarcy can be definitely improved

3 Spiking Neural Network

3.1 Motivation

Spiking Neural Network or SNN as they are more popularly called are the newest thing in Neural Network Research and even touted as the Third Generation Network in the neural network research. With the advent of Neuro Grid[1] True NorthChip[2] . These are event based Chips which are extremely energy efficient as well as turing complete. I believe as SNN tries to emulate the brain architecture in the near future these are most likely going to replace traditional clock based architecture.

3.2 Implementation

Initially thought process behind tackling the problem was similar to Character Recognition using SNN[3].The idea was to take image samples of all the objects that needed to be semantically classified(That has to be picked up manually from the sample images). For example: Road , sign etc and make the network learn from those images and the output matrix over that whole image is that particular number it corresponds to that particular number. Take for instance if the image of Road is being learned and the sample images are of 28 X 28 size then in the final output all the 28 X 28 pixel will be labeled as '8' .

This implementation was inspired by a paper on motion detection using SNN[4] . Here instead of Unsupervised Learning as quoted in the paper I tried to implement a Supervised Learning of the same, where the architecture included the image layer(512 X 640 X 3) and a single layer above it which had 10 neurons (Ideally neuron 'n' in this layer spikes only when 'n' numbered segment is detected) . Where I tried to change weights using two weight change rules 1. Constant weight update 2. exponential weight change.

3.3 Results

In both implementations as mentioned above the weight change rule as suggested in the section above leads to a lots of variations of weights which finally leads to inaccuarate predictions and error as high as 90 percent.

The possible reasons behind this may be the fact that in all the papers referred above the object of interest seems to be a closed object (for example : A Ball) but in the problem statement mentioned in the assignment the 'objects' are all open ended like sky. So the Weights fail to converge well and hence a wrong prediction.Another possible reasons maybe the fact that all the objects or labels mentioned in the problem statement above follow some sort distribution. for example take 'sky' and 'road' that needs to be labeled. Now suppose weights are set for detection of a ' Road' but while predicting if 'sky' appears then that particular network will surely spike for a 'sky' as intensity of 'sky' image is more it will have to spike in that case.

3.4 Conclusions and further Work

To overcome all the problems mentioned above I have designed a supervised network where there are two parameters to be . The first one is the settling potential E_L and the second being the weights ofcourse

The Settling Potential is being set because of the fact that here the identification is not just based on the neuron that spikes but based on the fact that the neuron which spikes the minimum will be the identification.

4 Bag of Words Model

4.1 Motivation

The Motivation behind this came from the fact that in order to classify an image in 'sky' 'Roads' etc we can take an image of road and compare it with our desired pixel by comparing it with neighbourhood our desired pixel. Later after some search this turned out to be most basic semantic segmentation algorithm named 'Bag of Words'

4.2 Implementation

Here as the Image of the objects were not given. I have basically decided to pick up these images matrix by scouting through each and every pixel of the png image and took the corresponding neighbourhood of RGB Image to be the object matrix of that corresponding object. Finally averaging out such Image matrices over all such object matrix detected gives me the final image matrix.

4.3 Results and Conclusion

Here it was observed that if the neighbourhood matrix is taken to be 21 X 21 the error came out to be around 67% if the neighbourhood matrix is taken to be 31 X 31 then the error comes out to be around 50% and if the neighbourhood matrix is taken to be 41 X 41 then the error comes to around 30%
However, We observed that the method is computationally expensive and instead convolution networks may be implemented instead.

5 Introduction

There are four forms of representation of Angles in typical Vision Applications:

1. Euler Angles:This just has a Net Rotation as the Vector sum of Rotation around all the Three Principal Axis. Namely X,Y,Z Axis or pitch, yaw, Roll

2. Quarternion : Quarternion in very similar to what is an interpolation of 3D version of representation of complex Numbers

3. Axis Angle Representation : This is very similar to quarternions except that it uses a very complicated Rodriguez formula to compute angular rotations

4. Rotation matrices : Simple 3 X 3 Rotation Matrix

6 The Task

6.1 Advantages and Disadvantages

Advantage and Disadvantage of Euler Angle Representation

One of the important Disadvantages of Euler Angle Representation is the fact many a times it leads to a condition called as gimbal lock. A Condition which leads to system losing one degree of Freedom.

Another Disadvantage of the Euler Angle Representation is the fact that it gives a Rotation Matrix which is multiplication of Three Different Matrices and the order of the rotation has to be pre determined in advanced

Advantages of The Euler Angle maybe that it is very intuitive to begin with and can be used quite easily. On the other hand it is computationally expensive

One advantage of using **Unit Quaternion** Representation is the Fact that

it is more compact as compared to Matrix Representation of the Rotation which significantly reduces degree of complexity in the problem For Example: Unit Quarternion only needs 4 Parameters as compared to 9 (3 X 3) needed by the Rotation Matrices

Also one of the other advantage of using Unit Quarternion Representation is the fact that it is used heavily when the camera motion has to move smoothly in a feed for examples: Video Games

One Disadvantage can be that Quarternion are not that intutive and they are generally used when there are random angle rotation around some axis

Axis Angle Representation is an idealized representation of Unit Quaternion Representation Advantages:Very much useful for small rotation angles

Disadvantages: When two angles are to needed to give a combined Rotation. Axis Angle Rotation fails there miserably

The Axis Angle Representation is minimal and hence does not need much information Doesn't suffer gimbal lock Axis angle requires conversion to other form

6.2 Rotation Matrices

$$R(a) = \begin{bmatrix} 0.707 & -0.353 & 0.612 \\ 0.707 & 0.353 & -0.612 \\ 0 & 0.866 & 0.500 \end{bmatrix}$$

$$R(b) = \begin{bmatrix} 0.500 & 0.000 & -0.866 \\ 0.000 & 1.000 & 0.000 \\ 0.866 & 0.000 & 0.500 \end{bmatrix}$$

6.3 Rotation Matrix to Quarternions

$$Q(a) = \begin{bmatrix} 0.800 \\ 0.000 \\ 0.000 \\ 0.4619 \end{bmatrix}$$

$$Q(b) = \begin{bmatrix} 0.8660 \\ 0.000 \\ 0.000 \\ 0.000 \end{bmatrix}$$

6.4 Composition

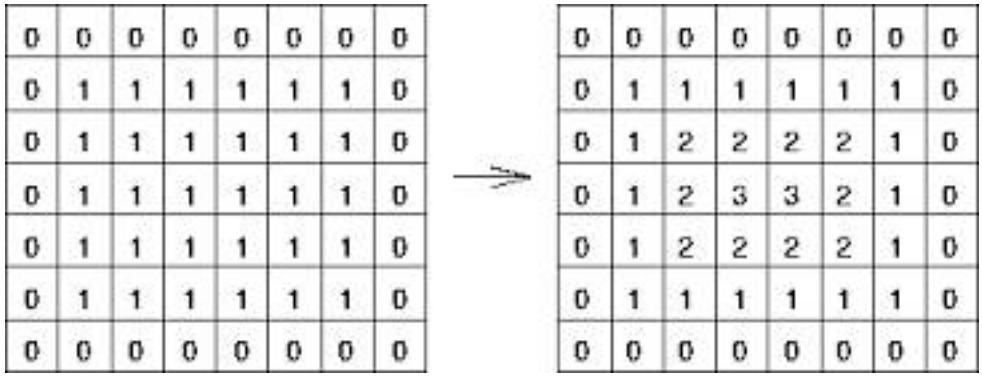
Composition of Q(a) and Q(b) is not commutative at all. Intutively it is also quite clear because this problem is very similar to having a convention of Euler Angle Rotation.

In computation of Q_f and QQ_a will turn out to be the same because as we saw earlier Q(a) and Q(b) are commutative so it never violates that rule in the question

7 Distance Transform

7.1 Introduction

Morphological Distance Transform is essentially a measure which indicates how far is the body from the nearest obstacle.This is generally a binary image consisting of either ones and zeros and our job is to take the nearest neighbor from it and compute the distance from that neighbor and determine the corresponding image.



Generally there are many types of metrics while taking Distance Transform of an Image
1 Manhattan Distance Transform
This type of metric includes Distance being the absolute difference in their Cartesian coordinates
2Euclidean Distance Transform
This involves our usual euclidean distance metric
3Chess Board Distance Transform
This is basically the L infinity norm

7.2 Implementation

In this particular part of the assignment I have decided to go with implementing euclidean distance transform. Two approaches were tried here in this case
1In which we go on manually finding out the obstacles coordinates and then we go one finding the distance of any point from that obstacle and find the minimum from all of them
2 Here we take an advantage of open CV Functions where we find out the contours in the map and then using point polygon test (A function in opencv) we find out the minimum of distances from each of these contours.

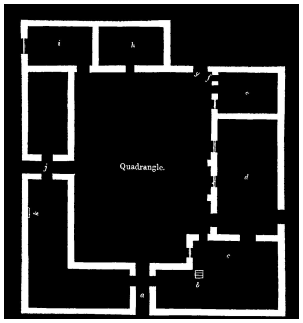


Figure 1: Orignal Image

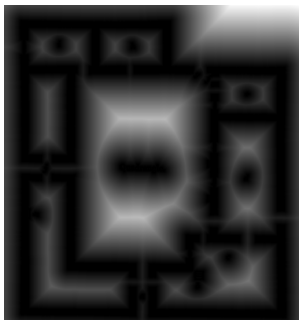


Figure 2: Distance Transformed Image

8 Motion Planning

8.1 Introduction

"Motion planning (also known as the navigation problem or the piano mover's problem) is a term used in robotics for the process of breaking down a desired movement task into discrete motions that satisfy movement constraints and possibly optimize some aspect of the movement." The algorithms for motion planning in this case has been to use Grid Based search for the algorithm.

8.2 Implementation

Here there are two types of approaches tried:
1.Djiskstaw's Algorithm
The map is generally considered nodes and traversable nodes as it's edges so basically to select the next node we basically take the minimum of the one which has less cost to the goal approach
2.A^* Algorithm
In A^* Algorithm basically the next node is selected as the one which has the minimum total cost to the goal i.e historic cost + expected cost.
It is expected that for high speed searches A^* works the best but it may give out wrong answers.In this code it was observed that it doesn't give out the same path and there was no factor time in this case.
There was also another approach tried but as it turns out it only suitable for obstacles which are simple for example a Block but fails for concave obstacles.

Note: Here initial point has been taken as [144,144] and the goal is taken as [811,920]. As the goal coordinates given in the question lies on the obstacle it didn't work correctly.

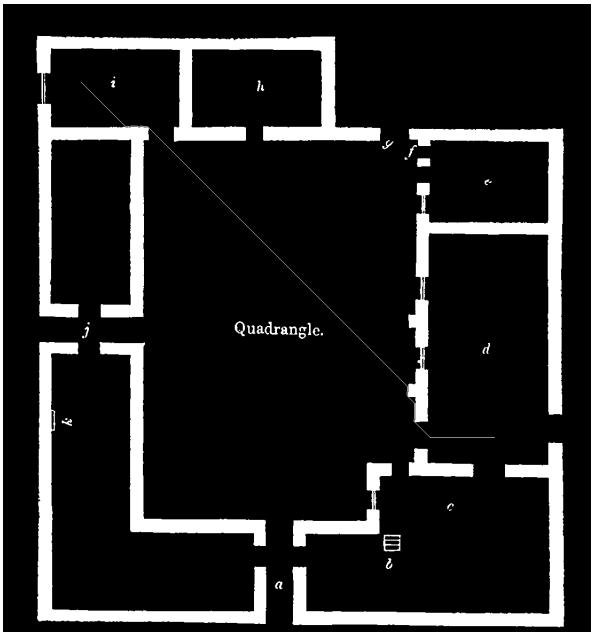


Figure 3: Path of the Image

9 Refrences

1.Neurogrid: A Mixed-Analog-Digital Multichip System for Large-Scale Neural Simulations
2. https://www.sciencemag.org/content/345/6197/668.abstract
3. Unsupervised features extraction from asynchronous silicon retina through Spike-Timing-Dependent Plasticity
4. Character Recognition using Spiking NeuralNetworks
5. Gradient Based hyperparameter optimization
6.Python PIL, OpenCv,Numpy,Matplotlib Libraries in Python