



Fakultät für Informatik
Lehrstuhl für Bildverarbeitung und Mustererkennung

Image Classification using the deep learning caffe framework

Neeraj Sujan - 03656452

Praktikum *Maschninelles Lernen für Anwendung des Computersehens* SoSe2015

Submission: 08. May 2015

Image Classification using the Caffe framework.

Neeraj Sujan

Fakultät für Informatik

Technische Universität München

Email: n.p.sujan@tum.de

Abstract—The report [?] discusses the popular caffe framework to classify images using deep learning algorithms. The training model is based on the popular imagenet example and the data from the ilsvrc 2012 challenge. The images used for classification were provided by the chair of pattern recognition affiliated to the faculty of computer science, TU München. The paper also discusses the popular back propagation algorithm.

I. INTRODUCTION

Neural Networks: Neural Networks or Artificial Neural Networks, as it is popularly known, are a family of machine learning algorithms, that builds upon the popular single layer perceptron, developed by Rosenblatt. Neural Networks works on the principle of multiple hidden layer with input and output neurons. Each hidden layer contains an activation function, such as a sigmoidal function to make the network as efficient as possible. As in other classification algorithms, the goal of the network remains to minimize the error function, where the error function is given as follows

A. Caffe Framework

The Caffe framework [1] from UC Berkeley is designed to let researchers and novice learners create and explore convolutional neural networks and other Deep Neural Networks easily. Caffe provides state-of-the-art modeling for advancing and deploying deep learning in research and industry with support for a wide variety of architectures and efficient implementations of prediction and learning. The Caffe framework follows a layered architecture, following a b

B. Back Propagation Algorithm

The following four steps describe the Backpropagation algorithm (taken from Bishop's Book)

1. Apply an input vector x_n to the network and forward propagate through the network using

$$a_j = \sum_i w_{ji} z_i \quad (1)$$

$$z_j = h(a_j) \quad (2)$$

to find the activation of all the hidden and output units.

2. Evaluate the δ_k for all the output units using

$$\delta_k = y_k - t_k \quad (3)$$

3. Backpropagate the δ' using

$$\delta_j = h'(a_j) \sum_k w_{kj} \delta_k \quad (4)$$

to obtain δ_j for each hidden unit in the network.

4. Use the below equation to evaluate the required derivatives

$$\frac{\partial E_n}{\partial w_{ji}} = \delta_j z_i \quad (5)$$

Modification : If the input neuron is modified such that the activation function is not sigmoidal, then the following changes in the back-propagation algorithm will take place.

In equation (2), $z_j = h(a_j) = f(\sum_i w_{ji} z_i + b)$. Equation (4) will change correspondingly to $\delta_j = f'(\sum_i w_{ji} z_i + b) \sum_k w_{kj} \delta_k$

1) *Experiment*: The figure below shows the 8 images that were to be classified using the ilsvrc2 2012 training model.

Probability of class : In a multi-output classification problem, the probability if a class plays a very important role. The probability of a class indicates the likelihood of occurrence of a particular class. Given a particular unknown input, the corresponding output vector O , then the estimated probability that it belongs to each class is given as follows.

$$P_{co}(c | O) = \frac{p(c | O)}{\sum_{c'} p(c' | O)}$$

Image	Class	Probability of the class	Entropy
1	8(hen)	0.155564	4.50067
2	315(mantis,mantid)	0.234804	3.29147
3	421(bannister,banister,balustrade,balusters,handrail)	0.100069	4.36765
4	639(maillot,tank suit)	0.0476254	4.9167
5	673(mouse, computer mouse)	0.259835	3.07259
6	947 (mushroom)	0.376555	1.64067
7	834 (suit, suit of clothes)	0.642554	1.10757
8	980 (volcano)	0.172035	3.90447

C. Results

As shown in the above table, the model was able to classify the 6th image correctly to a mushroom, whereas for all other images it gave a very good guess deciphering part of the image. For instance, in image 4, the model was able to predict the suit, whereas in the Alexnet model, the prediction was a tie. The image in example 2 looks like a grasshopper, hence the the model predicted it to be a mantis (Gottesanbeterin in German),whereas the image actually is a plant found in tropical countries but due to the colour and shape of the plant, it was predicted to be a mantis. The model also managed to decipher the hand-rail in the 3rd image which was a very complex images with many features like a slide, people, rails etc. The 8th image was a picture of a factory with some kind of heavy machinery, whereas the model predicted it to be a volcanic eruption, due to the fire shown in the image. The model performed sub-optimally for the 1st,4th and the 5th image and it classified them to be a hen, tank suit and a computer mouse respectively.

Cropping: On cropping the image there was a slight improvement

II. CONCLUSION

The conclusion goes here.

REFERENCES

- [1] H. Kopka and P. W. Daly, *A Guide to L^AT_EX*, 3rd ed. Harlow, England: Addison-Wesley, 1999.
- [2] J. Denker and Yann leCunn, *Transforming Neural-Net Output Levels to Probability Distributions*. AT & T Bell Laboratories, Holmdel,NJ 0773