IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 5, Issue 4, April 2016

Prediction of Facial Key points in Images Using Neural Networks

Manish Bhelande[1], Aadharsh Krishnan[2], Akhilesh Bharadwaj[3], Niraj Palecha[4], Yash Tawade[5]

Assistant Professor, Department of Information Technology, Shah & Anchor Kutchhi Engineering College, Chembur[1]

UG Student, Department of Information Technology, Shah & Anchor Kutchhi Engineering College, Chembur[2,3,4,5]

Abstract: Detecting facial key point positions on images is a challenging task since facial features differ significantly from one individual to another. Even for a certain individual, there is an occurrence of wide variations due to factors such as size, position, viewing angle, and illumination effects. In this paper, we present a system that trains and compares multiple neural networks and try to optimize their learning rate constantly. This juxtaposes the different levels of accuracy obtained in predicting the facial key points in images even with a wide array of significantly varying facial features. Our method uses a simple three-layer neural network and distinct variations of convolutional neural networks.

Keywords: facial key points, neural networks, hyper-parameter optimization, deep learning, convolution networks.

I. INTRODUCTION

The dataset we are working with is taken from an ongoing contest - Facial Keypoints Detection [1] on Kaggle, which was made available by Dr. Yoshua Bengio of the University of Montreal. Training dataset consists of 7049 images with coordinates of 15 keypoints. Test dataset consists of 1783 images. All images are 96x96 pixels. Training, testing, and validation of a neural network and deep neural networks with a large number of layers is a time-consuming process. Therefore, for implementing and training the neural networks much faster, we utilized GPU for data-intensive calculations. Theano [2][3] is a Python library that enables dynamic generation of optimized C code that can be executed much faster on a CUDA capable GPU.

II. RELATED WORK

Cascaded Convolutional Networks

Yi Sun, Xiaogang Wang & Xiaoou Tang [4] designed a method for determining the positions of keypoints on facial images with the help of a meticulously planned 3 level convolutional neural network. At every level, the outputs of the different neural networks are combined for an accurate prediction. With the help of convolutional network deep structures, they extract global high level facial features easily for the entire face region at the initialization stage itself. This constitutes towards high accuracy prediction of the keypoints. Moreover, since they train the networks to predict every keypoint at the same time sequentially, geometric constraints are thus encoded implicitly. But even with high accuracy and reliable prediction, it has been noticed that this technique has a limitation that does not enable inputs to large regions for the initial prediction.

Color Image Face Detection

Rein-Lien Hsu et al. [5] proposed an algorithm for facial detection in digital color images even with different Illumination conditions and high background complexity. It utilizes an illumination compensation method and also a nonlinear color transformation for detecting skin patches spanning the entire image. It then generates face entities according to spatial arrangement of the detected skin regions. The authors suggest creating boundary maps for verification of the face candidates but they still face difficulty in high-luma and low-luma skin tones in color images.

DropConnect Regularization

Li Wan, Matthew Zeiler et al. [6] introduced the concept of DropConnect, which is a generalized version of DropOut method. The paper proposes this new technique where, instead of each connection, each output unit can be dropped with an alternate probability of (1 - p). The authors use DropConnect to regularize large fully connected network layers within each neural network. For training purposes, an arbitrarily selected subset of weights within each network layer is set to zero. Each unit therefore receives an arbitrary subset of units as input from the previous layer. It is noticed that the dynamic sparsity is beset on the weights and not on the output vectors of previous network layers. Finally, it derives a bound on performance based on generalization for both DropOut and DropConnect.

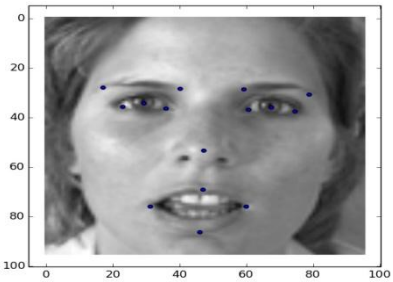

Fig. 1 : Model's facial keypoint prediction on a test digital image.

III. SYSTEM DESIGN

Simple 3 Layer Network or Single Hidden Layer Network

We trained a simple fully connected neural network with a single hidden layer with initially 50 neurons in the hidden layer. The first of the three layers is the input layer with 9216 neurons which are fed with 9216 pixel values for each image.

The output layer was defined with 30 neurons representing the 15 keypoints with x and y coordinate for each keypoint.

The weights for this network were randomly initialized and updated on each iteration (epoch) with an optimization technique known as Nesterov's Accelerated Gradient Descent (NAG).

The training of a neural network is handled by tweaking some hyper-parameters. Hyper-parameters such as Learning rate and Momentum are associated in training a neural network that is optimized using NAG.

The objective function used is Mean Squared Error(MSE) as this a regression task. The training phase is executed for 400 times which optimized weights at the end of each iteration. In training the network, we reached a minimum of 2.989 for MSE with this simple method.

Convolutional Neural Network

Convolutional neural networks are a major reason for the recent breakthrough in computer vision. This approach is different from the network involving fully connected layers. Convolutional layers use local connectivity and pool sharing which decrease the number of parameters. A unit in a convolutional layer connects a 2-dimensional matrix of neurons from the previous layer.

The network implemented consists of 3 convolutional layers and 2 fully connected layers. Each convolutional layer is followed by a max-pooling layer. We found that at around 1000 epochs, the network reaches 1.95 MSE and doesn't improve much further.

Optimizing hyper-parameters

To train the networks we initialized the hyper-parameters, Learning rate as 0.1 and Momentum as 0.9. These parameters are used by the optimization method to update the weights for the next iteration. Using static hyper-parameters is not an efficient approach. Changing these dynamically as the number of iterations increase is an approach suggested by Ilya Sutskever et al. [7].

The learning rate is decreased linearly with the number of iterations. Because, when we start training the model we are farther away from an optimal state. Momentum, on the other hand, is increased. These changes make the training much faster and compared to the convolutional neural network with static hyper-parameters, this approach stops improving at around 750 iterations.

IV. CONCLUSION

Thus restating our proposed thesis and summarizing the main points of this paper, we conclude stating that we have tested two neural networks, namely a single hidden layer network and a conventional neural network. Both of the networks have been trained using hyper-parameters in order to automatically update the weights for the next iteration. However, on testing, it has been observed that even though the latter network takes much more time to be trained, it reaches a much better MSE than that of its former counterpart hence proving to be a much more viable option.

REFERENCES

[1]. Kaggle - Facial keypoints detection [Online]. Available: https://www.kaggle.com/c/facial-keypoints-detection
[2]. F. Bastien, P. Lamblin, R. Pascanu, J. Bergstra, I. Goodfellow, A. Bergeron, N. Bouchard, D. Warde-Farley and Y. Bengio. “Theano: new features and speed improvements”. NIPS 2012 deep learning workshop.
[3]. J. Bergstra, O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, G. Desjardins, J. Turian, D. Warde-Farley and Y. Bengio. “Theano: A CPU and GPU Math Expression Compiler”. Proceedings of the Python for Scientific Computing Conference (SciPy) 2010. June 30 - July 3, Austin, TX
[4]. Yi Sun, Xiaogang Wang, and Xiaoou Tang. 2013. Deep Convolutional Network Cascade for Facial Point Detection. In Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR ’13). IEEE Computer Society, Washington, DC, USA, 3476-3483.
[5]. Rein-Lien Hsu, M. Abdel-Mottaleb and A. K. Jain,” Face detection in color images,” in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no. 5, pp. 696-706, May 2002.
[6]. Wan, Li, Matthew Zeiler, Sixin Zhang, Yann L. Cun, and Rob Fergus.”Regularization of neural networks using DropConnect.” In Proceedings of the 30th International Conference on Machine Learning (ICML-13), pp. 1058-1066. 2013.
[7]. Sutskever, Ilya, James Martens, George Dahl, and Geoffrey Hinton. "On the importance of initialization and momentum in deep learning." In Proceedings of the 30th international conference on machine learning (ICML-13), pp. 1139-1147. 2013.
[8]. Michael A. Nielsen,” Neural Networks and Deep Learning”, Determination Press 2015

DOI 10.17148/IJARCCE.2016.5425