ISSN XXXX XXXX © 2018 IJESC

Research Article Volume 8 Issue No.4

Similarity Amongst Us: An Invariant Transformational Mapping

Neeraj Garg[1], Raunaq Jain[2], Somin Wadhwa[3]
Associate Professor[1], Student[2, 3]
Department of CSE
Maharaja Agrasen Institute of Technology, Delhi, India

Abstract:
Conventional wisdom dictates that deep neural networks are really good at learning from high dimensional data like images or spoken language, but only when they have huge amounts of labelled examples to train on. On the contrary, humans are capable of one-shot learning - if you take a human who's never seen a spatula before, and show thema single picture of a spatula, they will probably beable to distinguish spatulas from other kitchen utensils with astoundingly high precision. Our model is given a tiny labelled training set S,which has N examples, each vectors of the same dimension with a distinct label y. Unlike conventional learning systems where the loss function is a sum over samples, using the Contrastive Loss Function.

Keywords: One-Shot Learning, Deep Neural Networks

I. INTRODUCTION

The ability to learn with as little data as possible in a 'human-like' manner is a desirable characteristic in machine learning problems especially pertaining to neural calculation requiring large amount of compute to run. This formed the basis of FaceNet[1]. Idea was to map images in a compact Euclidean space where distances form a direct mapping to the facial similarity. The approach was based on altering the loss function proposed by *Hadsell et all* commonly known as the Contrastive Loss function. It is used to learn the parameters W of a parameterized function GW, in such a way that neighbors are pulled together and non-neighbors are pushed apart. Prior knowledge can be used to identify the neighbors for each training data point. This is achieved through the use of an ingenious approach where we try to learn the similarity between the example points in an embedding space. Similar points are brought closer to each other while dissimilar points are kept away. Euclidean distance is used as a metric to find the similarity or the dissimilarity between the points in the embedding space. A different type of artificial neural network architecture, known as Siamese Neural Network, to find the similarity between the inputs. It is a shared neural network wherein the pair of inputs is passed through together which is combined in the last layers of the network. We analyze the different loss functions which have been used till now in the form of a survey. Siamese Neural Networks have been used in various domains, such as image classification, song prediction, and language prediction. They are very popular for different verification tasks, such as online signature verification, face verification, etc. It has also been used for one-shot image recognition. To illustrate such an architecture we take up the problem of Face Recognition. Traditional machine learning based approaches usually work in a similar manner wherein they first extract features from the input, say image features, and then applying a suitable metric on the extracted features to indicate the similarity between any two images. Since similarity measurement and feature extraction are two independent tasks, the system performance is usually suboptimal when compared with a system employing convolutional neural networks. Hence, deep learning is attempted to automatically learn features and metrics. We demonstrate that approach pertaining to Siamese Architecture used along with Contrastive Loss can learn a low dimensional mapping given the neighbourhood relationships that may come from a set of available data points. In our case the images.

II. MATERIALS AND METHODS

1. **AT&T Face Database:** AT&T Facial Database contains forty categories of faces with ten images of each. All in all, it contains about four-hundred images. Using this dataset provides the appropriate amount of data to test and evaluate in a 'one-shot' manner under our evaluation criteria. Purposely keeping per class number of images to be low. The facial database used is available at the Cambridge Repository[3].
2. **Convolutional Neural Network:** Siamese nets were first introduced in the early 1990's for similarity matching to solve signature verification task. A siamese neural network consists of a twin network joined together at the top layer by a function. This function is used to calculate a metric using the higher level feature representation. A siamese neural network accepts two inputs at the same time which are passed through the network whose weights are tied. This causes any two similar images to not be mapped to very different locations in the embedding space. A distance function is generally used as the function at the top. The distance is minimized for similar pairs while maximized, till above a margin, for dissimilar pairs. We use a convolutional neural networks because it has been shown to achieve exceptional results in many computer vision applications, particularly in image recognition tasks. All our experiments were performed in python using PyTorch- a CUDA accelerated fast GPU based library.
3. **preprocessing:** In image processing and computer vision , anisotropic diffusion, also called Perona Malik diffusion, is a technique aiming at reducing image noise without removing

significant parts of the image content, typically edges, lines or other details that are important for the interpretation of the image. Anisotropic diffusion resembles the process that creates a scale-space, where an image generates a parameterized family of successively more and more blurred images based on a diffusion process. Each of the resulting images in this family is given as a convolution c between the image and a 2D isotropic Gaussian filter, where the width of the filter increases with the parameter. This diffusion process is a linear and space invariant transformation of the original image. Anisotropic diffusion is a generalization of this diffusion process: it produces a family of parameterized images, but each resulting image is a combination between the original image and a filter that depends on the local content of the original image.

4. Contrastive Loss Function: A meaningful mapping from high to low dimensional space which maps similar input vectors to nearby points on the output manifold and dissimilar vectors to distant points. A loss function whose minimization can produce such a function is Contrastive Lose Function. Unlike conventional learning systems where the loss function is a sum over samples, the loss function here runs over pairs of samples. Intuitively, this function just evaluates how well the network is distinguishing a given pair of input vectors.

Let $X_1, X_2 \in I$ be a pair of input vectors shown to the system and let Y be a binary label assigned to this pair. If X_1 and X_2 are deemed similar then $Y = 0$, and if they are deemed dissimilar then $Y = 1$. Define the parameterized distance function to be learned D_W between X_1, X_2 as the euclidean distance between the outputs of G_w, that is,

$$D_W(\overrightarrow{X_1}, \overrightarrow{X_2}) = \left\| G_w(\overrightarrow{X_1}) - G_w(\overrightarrow{X_2}) \right\|_2$$

Equation 1.0

To shorten notation, $D_W(\overrightarrow{X_1}, \overrightarrow{X_2})$ is written D_w. Then the contrastive loss function is

$$(1 - Y)\frac{1}{2}(D_W)^2 + (Y)\frac{1}{2}\{max(0, m - D_W)\}^2$$

Equation 2.0

m is a margin whose value is greater than 0. The margin denotes a radius around $D_W(X)$. It indicates that dissimilar pairs that are beyond this margin will not contribute to the loss function. Simply minimizing the distance between all the similar pairs will lead to a collapsed solution, since $D_w$and the loss L could then be made zero by setting $G_w$to be a constant.

III. RESULTS AND DISCUSSIONS

1. Analysis

One major issue which we find with both the contrastive loss is the use of margin to keep negative points from accumulating close together to the positive points. The margin causes the loss function to lose a ton of information. The problem is that the loss can never get below 0. So, as long as the negative value is further than the positive value plus the margin there will be no gain for the algorithm as it wouldn't condense the positive and the anchor anymore. This makes the positive points to cloud around the anchor instead of clustering as close as possible, thereby losing a lot of information.

This problem of information loss is faced by the contrastive loss function too.

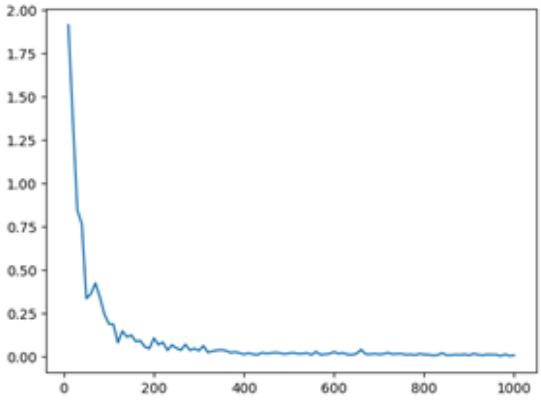

Figure.1. Training Loss Vs Number of Epochs

Figure.2. Normalized similarity as predicted.

After selection of features, we created scatter plots of the independent variables against our target i.e. salary to determine their individual effects on the same (S2-S5 Fig.). Figure 1 represents the loss at each epoch of training. We can clearly figure out that the network stabilizes significantly soon. The model was able to clearly distinguish the similar and dissimilar pairs of faces of the AT&T dataset as seen from Figure 2 and Figure 3.

2. Assessment of performance

With this premise, we intended to design a strategy that could be meaningfully implemented for the prediction of salary. Along with mean absolute error and root mean square error, we have also made use of residual plots for a number of algorithm used in the analysis (Support vector machines, Random forests (n-estimator = 100), Lasso regression (a = 0.001), Ridge regression (a = 0.1)). Results (Table 1) indicate that Ridge and Lasso regression (Figure 3 & 4) fair out marginally better than Random forest (Figure 5), with Support Vector Machines following closely behind.

Figure.3. Degree of dissimarity as normalized by the model.

IV. SUMMARY AND CONCLUSIONS

These are very interesting findings and it is somewhat surprising that it works so well. Future work can explore how far this idea can be extended. Presumably there is a limit as to how much the v2 embedding can improve over v1, while still being compatible. Additionally it would be interesting to train small networks that can run on a mobile phone and are compatible to a larger server side model. An obvious drawback of our model is the fact that, as the support set S grows in size, the computation for each gradient update becomes more expensive. Although there are sparse and sampling-based methods to alleviate this, much of our future efforts will concentrate around this limitation. Further, as exemplified in the Image Net dogs subtask, when the label distribution has obvious biases (such as being fine grained), our model suffers. We feel this is an area with exciting challenges which we hope to keep improving in future work [4-5].

V. ACKNOWLEDGEMENTS

N.G., R.J. and S.W. would like to thank the Department of Computer Science and Engineering at MAIT, for providing the computational facilities required to complete the project and their continuous support.

VII. REFERENCES

[1]. Florian Schroff, Dmitry Kalenichenko, James Philbin (2015), FaceNet: A Unified Embedding for Face Recognition and Clustering.*arXiv:1503.03832*

[2]. Raia Hadsell, Sumit Chopra, Yann LeCun (2006) Dimensionality Reduction by Learning an Invariant Mapping. *Computer Vision and Pattern Recognition*

[3]. AT&T Facial Database, http://www.cl.cam. ac.uk /research/ dtg/attarchive/facedatabase.html

[4]. Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, Daan Wierstra (2017) *Matching Networks for One Shot Learning, arXiv:1606.04080v2*

[5]. Yao-Hung Hubert Tsai, Ruslan Salakhutdinov (2018), Improving One-Shot Learning through Fusing Side Information. *arXiv:1710.08347*.