EasyChair Preprint

№ 893

Improving Siamese Networks for One-Shot Learning Using Kernel-Based Activation Functions

Shruti Jadon and Aditya Acrot Srinivasan

August 19, 2020

Improving Siamese Networks for One-Shot Learning Using Kernel-Based Activation Functions

Shruti Jadon and Aditya Arcot Srinivasan

Department of Computer Science
University of Massachusetts Amherst
Amherst, MA, USA
sjadon@umass.edu,asrinivasan@cs.umass.edu

Abstract. The lack of a large amount of training data has always been the constraining factor in solving many problems in machine learning, making one-shot Learning one of the most intriguing ideas in machine learning. It aims to learn necessary objective information from one or only a few training examples. This process of learning in neural networks is generally accomplished by using a proper objective function (loss function) and embeddings extraction (architecture). In this paper, we discussed metric-based deep learning architectures for one-shot learning such as siamese neural networks [10] and present a method to improve on their accuracy using *Kafnets* (kernel-based non-parametric activation functions for neural networks) [17] by learning finer embeddings with relatively less number of epochs. Using kernel activation functions, we are able to achieve strong results that exceed ReLU-based deep learning models in terms of embedding structure, loss convergence, and accuracy. The project code with results can be found at `https://github.com/shruti-jadon/Siamese-Network-for-One-shot-Learning`.

Keywords: One-Shot Learning, Embeddings, Computer vision, Gaussian distribution, Loss function

1 Introduction

Humans learn new things with a very small set of examples—e.g. a child can generalize the concept of a 'Dog' from a single picture, but a machine learning system needs a lot of examples to learn its features. In simpler terms, when humans are introduced to a new concept, they seem to be able to grasp it quickly and extend extracted information to understand different variations of the concept [12]. Machine learning as a field has been highly successful at a variety of tasks such as classification, web search, image, and speech recognition. Oftentimes, however, these models do not do very well in the regime of low data. This is the primary motivation behind One-Shot Learning; to train a model with fewer examples but generalize to unfamiliar categories without extensive retraining.

Deep learning has played a major role in the advancement of machine learning, but it also requires large datasets. Different techniques such as regularization reduce overfitting in low data regimes, but do not solve the inherent problem that comes with fewer training examples. Furthermore, the large size of datasets leads to slow learning, requiring many weight updates using stochastic gradient descent. This is mostly due to the parametric aspect of the model, in which training examples need to be slowly learned by the model into its parameters. In contrast, many known non-parametric models like nearest neighbors do not require any training, but performance depends on a sometimes arbitrarily chosen distance metric like the L2 distance.

In computer vision, One-Shot Learning can be defined as an object classification problem. In usual machine learning-based models, we need large amount of data [9], whereas as humans we only need a few examples. One-shot learning is an approach to train a model to extract information about different categories from a few training images, just like humans do [6].

One way of addressing problems in One-Shot Learning is to develop specific features relevant to the domain of the problem; features that possess discriminative properties particular to a given target task. However, the problem with this approach is the lack of generalization that comes along with making assumptions about the structure of the input data. In this paper, we make use of an approach similar to [10] [7] while simultaneously evaluating different activation functions [8] that may be better suited to this task. The overall strategy we apply is twofold; train a discriminative deep learning model on a collection of data with similar/dissimilar pairs. Then, using the learned feature mappings, we can evaluate new categories. Since One-Shot Learning focuses on models which have a non-parametric approach of evaluation, we came across Kafnets [17] (kernel based non-parametric activation functions) that have shown initial promise in this domain of training neural networks using different forms of activation functions; so as to increase non-linearity, therefore decreasing the number of layers, and increasing the accuracy in a lot of cases. This paper has proposed two activation functions KAF and KAF2D, and focuses on their nature of continuity and differentiability. We have experimented with these activation functions and compared their effectiveness against traditional ones when used in the context of One-Shot learning.

1.1 Related Work

The research in one-shot learning has recently caught the attention of the machine learning community. The work resulting in the best accuracy for the image classification problem dates back to the 2000s by [6]. The authors have attempted to solve one-shot image classification problems using Bayesian framework [6][7]. Their approach involved taking help from previously learned class which can help in forecasting a future one.

Lake et al. [14] solved character recognition problem by proposing a solution known as Hierarchical BPL (Bayesian Program Learning). In [12] and [14], the authors present an approach where an image is deconstructed into several smaller

pieces to ascertain an explanation for the structure of pixels. However, the joint parameter space being very large leads to inference becoming intractable.

There have also been other methods that approach the problem of One-Shot Learning. Wu et al.[19] tackle path planning as a one-shot learning problem for robotic actuation. Maas and Kemp [16] use Bayesian networks on the Ellis Island passenger data to infer attributes. Lake et al. [11] use a generative Hidden Markov Model along with a Bayesian inference algorithm to try and identify unseen words in a speech recognition paradigm. Bertinetto et al. [2] learn parameters of a model from a single training image. In this, the network effectively learns to learn, later generalize across different tasks using other examples.

Another approach to solve one-shot learning is to learn proper embeddings/features using different forms of architecture such as siamese network [3]. Given a training example of a novel category, the network projects the features into a differentiating embedding space, where classification is performed using nearest neighbor. In such scenarios, Training requires processing images in a pair of similar and dissimilar categories based on distance [5].

Matching Networks [18] also looks at the problem of One-Shot Learning from embeddings perspective. As mentioned before, non-parametric alternatives like the Nearest Neighbors model choose an arbitrary distance function. The authors solve this problem by formulating a loss function that encompasses training a nearest neighbor like model end to end. In the image classification task, the generated output label $\hat{y}$ for a test example $\hat{x}$ is computed very similar to what you might see in the Nearest Neighbors algorithm. The method progresses by embedding both the training examples as well as given test example $\hat{x}$ in the same space, computing a cosine similarity-based metric, and then passing that through a softmax layer to generate a label. In this approach, the embedding extraction process for the training examples is done by bidirectional LSTM. For the test examples, an LSTM layer extracts the key features and gets merged with bidirectional LSTM using an attention module to generate fully contextual embeddings. The paper also benchmarks various approaches to one-shot learning that could be used as a reference for our results.

The approach that has been recently explored is the use of Deep Siamese Networks, which we borrow heavily from [10]. Convolutional neural networks have achieved exceptional results in many large-scale computer vision applications, particularly in image recognition tasks. Several factors make convolutional networks especially appealing. Local connectivity can greatly reduce the number of parameters in the model, which inherently provides some form of built-in regularization, although convolutional layers are computationally more expensive than standard non-linearities. Also, the convolution operation used in these networks has a direct filtering interpretation, where each feature map is convolved against input features to identify patterns as groupings of pixels. Thus, the outputs of each convolutional layer correspond to important spatial features in the original input space and offer some robustness to simple transforms. Further, we use a contrastive loss function as defined in [4]. The objective of the siamese architecture is not to classify input images, but to differentiate between

them. So, a classification loss function (such as cross-entropy) would not be the best fit. Instead, this architecture is better suited to use a contrastive function. Intuitively, this function just evaluates how well the network is distinguishing a given pair of images.

As One-Shot Learning focuses on models that have a non-parametric approach of evaluation, we came across work [17] on Kafnets (kernel-based non-parametric activation functions) that has worked in this domain of training neural networks using different forms of activation functions. Scardapane et al. [17] introduce a new family of activation functions that are based on Gaussian kernel expansion at every layer's neuron. Based on different properties of Gaussian kernel-based models, the authors proposed different types of kernel functions ranging from designing and initialization perspectives. They also presented activation functions, for higher dimension domain with the help of 2D Gaussian Kernels. The proposed Gaussian Kernel-based Activation functions can approximate any form of activation function distribution across layers, by maintaining their convex nature. They are continuous and differentiable across ranges, making it most suitable for stochastic gradient descent. In this paper, we focus on two activation functions, KAF and KAF2D, and the effects of implementing them in a siamese architecture for One-Shot Learning.

1.2 Dataset

For this experiment, we have used three main datasets: MNIST [15], AT&T Database of Faces [1] (formerly 'the ORL Database of Faces'), and Omniglot dataset [13]. All of the above datasets are available freely online and did not require any form of preprocessing. We chose MNIST because we first wanted to test our models with images with less information. Then to showcase generalization of our approach, we chose dataset with more features to extract from images and decided to work with the AT&T Database of Faces. The MNIST dataset is a collective dataset of 0-9 handwritten digits by various humans. It consists of a training set of 60,000 examples and a test set of 10,000 examples, but as we are experimenting on the One-Shot Learning approach, we have randomly sampled and chose only 500 images from the available dataset. Similar to MNIST, the AT&T Database of Faces consists of ten different images of each of the 40 different human subjects. For certain subjects, Images consists of variation in times, light, and expressions.

Omniglot is a dataset by [13] which is one of the most popular datasets used for one-shot learning. It is specially designed to compare and contrast the learning abilities of humans and machines. This dataset consists of handwritten characters of 50 languages (alphabets) with 1623 total characters. There are only 20 samples for each character, each drawn by a distinct individual. The dataset is divided into 2 sets: background set and evaluation set. Background set contains 30 alphabets (964 characters) and is used to train the model whereas, the remaining 20 alphabets are for pure evaluation purposes only (Fig 1).

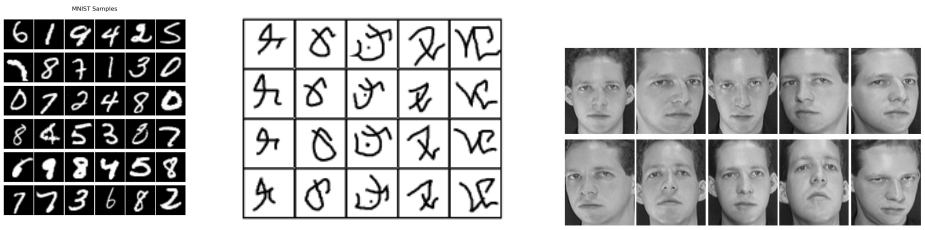

Fig. 1. Sample MNIST, Omniglot, and AT&T datasets

2 Methodology

For this work, we first implemented the basic architecture of Siamese Networks and Matching Networks using ReLU Activation function, and compared their learning abilities (Embeddings, and Optimization Ability) in comparison to Kernel Activation Functions. Siamese network [10] is an architecture with two parallel layers. In this architecture, instead of learning to classify its inputs using classification loss functions, the model learns to differentiate between two given inputs. It compares two inputs based on a similarity metric and checks whether they are the same or not. Similar to any deep learning architecture, a Siamese network also has two phases: the Training and Testing Phase. But usually, for a one-shot learning approach (as we won't have a lot of data points), we train the model architecture on a different dataset and test it for our less amount of dataset. To put in simpler terms, we learn image embeddings using a supervised metric-based approach with Siamese neural networks, then reuse that network's features for One-shot learning without fine-tuning or retraining (Fig 2).

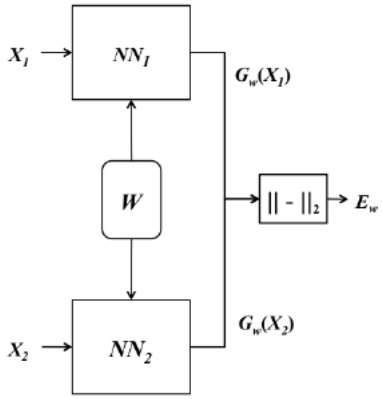

Fig. 2. Sample Siamese networks architecture

In this work, we have compared embeddings and optimization of Siamese Network architecture with ReLU Activation Function and Kernel Activation Functions. Through experimentation, we made 2 keen observations:

1. Embeddings learned using Kernel Activation functions were separated as compared to that in ReLU Function.
2. As Learning Capacity increased due to Kernel Activation Functions, it led to faster learning, i.e., better accuracy with less number of epochs.

One challenging task involved in this experiment is the implementation of activation functions. For this, we took the help of kernel [17] implementation of two Activation Functions, called KAF and KAF2D–KAF:Kernel Activation Function. Specifically, each activation function is modeled in terms of a Gaussian kernel expansion over D-dimension terms as
$g(s) = \sum_{i=1}^{D} \alpha_i \kappa(s, d_i)$ where $\{\alpha_i\}_{i=1:D}$ are mixing coefficients, $\{d_i\}_{i=1:D}$ are the called the dictionary elements, and $\kappa(.,.) : \mathbb{R} \to \mathbb{R}$ is 1D kernel function.

In simpler terms, we are trying to project every layer's output in the form of a Gaussian distribution and add them across D iterations. But If we observe, in general, Stochastic Gradient Training settings, D will increase as the number of training iterations increases. Therefore, we are fixing D to a number, and just varying mixing coefficients (α) of kernel functions. Stochastic Gradient descent works properly if a function is convex in nature; in matrix terms, the loss matrix needs to abide by the positive semi-definitive property. Kernel activation function also maintains the convexity of the function across layers, whereas ReLU causes the function to lose its convexity. For kernel function to maintain convexity, i.e;, have positive semi-definiteness property, it can have any possible α_i and d_i in

$$\sum_{i=1}^{D}\sum_{j=1}^{D} \alpha_i \alpha_j \kappa(d_i, d_j) \geq 0$$

As we are using 1D Gaussian kernel, $\kappa(s, d_i) = exp\{-\gamma(s - d_i)^2\}$ where $\gamma \epsilon \mathbb{R}$ is called the kernel bandwidth.

This gives us a proper derivatives for backpropagation as seen below:

$$\frac{\partial g(s)}{\partial \alpha_i} = \kappa(s, d_i)$$

$$\frac{\partial g(s)}{\partial s} = \sum_{i=1}^{D} \alpha_i \frac{\partial \kappa(s, d_i)}{\partial s}$$

Scardapane et al. [17] also consider a two-dimensional variant of the Proposed 1D Gaussian-based KAF (Kernel Activation Function), known as 2D-KAF. Roughly speaking, It can be viewed as Gaussian Mixture models, the 2D Gaussian kernels act as an independent pair of activation functions combined as a linear sum. Gaussian Mixture models' coefficient are learned using Expectation Maximization Algorithm, but can also be learned using stochastic gradient

descent algorithm. The combined 2D Gaussian Kernel equation can be defined as

$$g(s) = \sum_{i=1}^{D^2} \alpha_i \kappa(s, d_i)$$

here, $\kappa(s, d_i) = exp\left\{-\gamma \left\|s - d_i\right\|_2^2\right\}$ is a 2D Gaussian Kernel.

2.1 Siamese Network Architecture

Siamese Networks, as the name suggests, consist of 2 similar architectures. It's an architecture, which learns about the similarity between two inputs, and to ensure that we are calculating similarity scores on the same feature; Siamese Networks share equal weights among the two architectures.
We have experimented on Siamese Networks with 2 datasets: MNIST and AT&T.
For MNIST, we have coded a very simple 2 convolutional layer architecture:
Layer 1: Conv1 and Conv2;
Conv1: 1×20 followed by max pooling;
Conv2: 20×50 followed by max pooling;
Layer 2 : Fully Connected network with Activation Function. (50×500);
Layer 3 : Linear Layer (500×2)

As Face features are complex, for AT&T Face dataset, we have implemented dense-layered architecture:
Layer 1: Conv1, Conv2, and Conv3;
Conv1: 1×4 followed by Activation Function and max pooling;
Conv2: 4×4 followed by Activation Function and max pooling;
Conv3: 8X8 followed by Activation Function and max pooling;
Layer 2: Fully Connected network with Activation Function (100×100×8);
Layer 3: Linear Layer with Activation Function. (500×250);
Layer 4: Linear Layer with Activation Function. (250×5).

To ensure that we are learning proper differentiating features in 2 inputs, we need to use an objective function which trains network to learn contrasting properties. We have used Contrastive Loss function in both cases:

$$(1 - Y)\frac{1}{2}D_w^2 + (Y)\frac{1}{2}\left\{max(0, m - D_w)\right\}^2, and$$

$$D_w = \sqrt{\left\{G_w(X1) - G_w(X2)\right\}^2}$$

where G_w is the output of one of the sister networks. X1 and X2 is the input data pair.
For training, we have set the learning rate to 0.0005 and used Adam Optimizer.
We have also experimented on matching networks [18] using 1 D and 2 D Kernel Functions. For Matching Networks, we created 2 convoluted embedding extraction layer, followed by 3 fully connected linear layers and Bidirectional LSTMs for full contextual embeddings extraction.

3 Experiments and Results

For analysis of embeddings output, we compared intercluster versus intracluster distance of the projection of the features of different inputs, and we used silhouette score. We trained Siamese Network architecture on the MNIST dataset [7], with different activation functions [8], for 10 and 50 epochs. We have observed that the clustering score (silhouette score) in KAF2D was best, followed by KAF and ReLU, as displayed in Table 1. Silhouette Score ranges from -1 to 1, where close to 1 proves that the clusters obtained are good. There are 2 main reasons behind the improved performance:

1. For one-shot non-parametric activation functions can be of added value, as we have less amount of data. Though ReLU is a simple traditional non-parametric function, it doesn't add the capacity to architecture, rather it relies on added neural network layers. Whereas KAF and KAF2D are Gaussian kernel-based, therefore enable layers to capture better features of images.
2. Gaussian Distribution is considered as the best approximator when distribution is unknown, as stated by Central Limit Theorem. Therefore, for cases of fewer images, approximating the distribution of data after each layer will lead to better convergence.

We have also observed the training loss curve for all the cases, and see that with KAF, and KAF2D there were more fluctuations than ReLU, refer figure in Supplemetary Material.

We also experimented with the same architecture on the AT&T Face Similarity dataset. Like the output from the Siamese network, we obtained five-dimensional embeddings of the images in a plane; we then calculated pairwise distance, which was used by the metric to measure similarity.

Similar to the MNIST experiment, we ran Siamese Network architecture Model for Face-Similarity with KAF, KAF2D, and ReLU activation functions and observed that we were able to increase the similarity score just by using KAF and KAF2D. We also observed the behavior of the training loss curve with different functions. For ReLU, it converged much faster, which could be the reason for its efficiency, whereas KAF and KAF2D were fluctuating in the beginning but converged to a lower value of the loss at the end.

Table 1. Silhouette scores obtained for clusters in test set, 50 Epochs [MNIST Dataset]

Activation function	Silhouette score
ReLU	0.7766
KAF	0.8052
KAF2D	0.81641

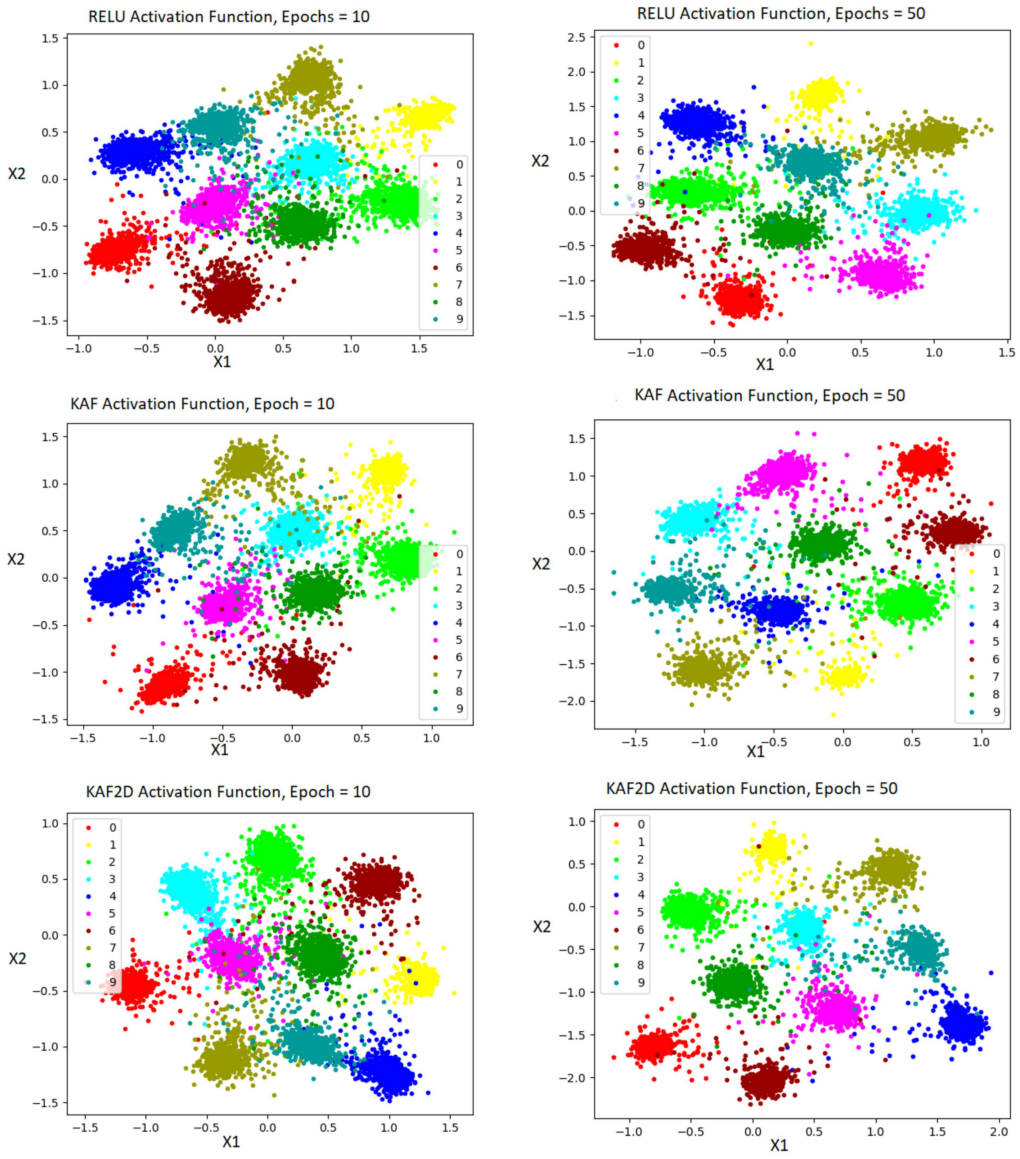

Fig. 3. Embeddings with ReLU, 1D, and 2D Gaussian Kernel activation functions for 10 and 50 epochs (MNIST dataset)

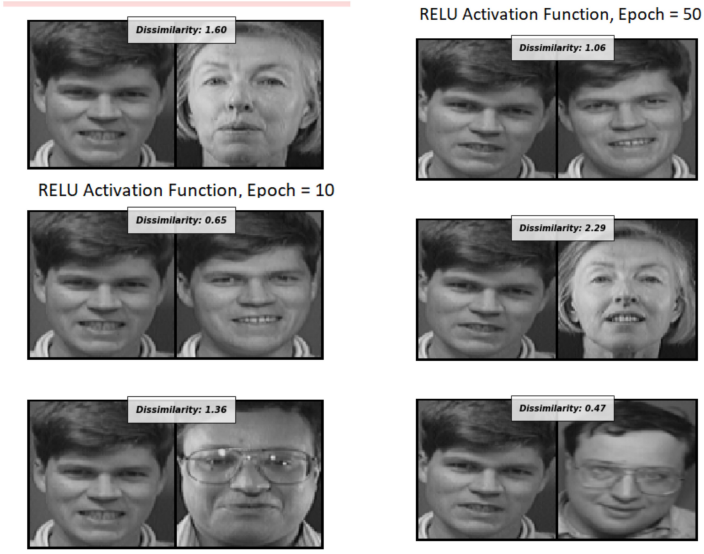

Fig. 4. Similarity scores for Faces with ReLU activation function (AT&T dataset)

Fig. 5. Similarity scores for faces with KAF activation function (AT&T dataset)

Table 2. Accuracies for matching networks on Omniglot dataset

Activation function	Accuracy(%)
ReLU	80.625
KAF	85.06
KAF2D	89.27

As a final experiment, we replicated the architecture of Matching Networks as in [18] with the KAF and KAF2D activation functions on the Omniglot dataset. we ran it for 1600 epochs with the results summarized in Table 2 (Figs 3,4, 5).

4 Discussion and Conclusions

In this paper, we presented a method to improve metrics-based one-shot learning approaches using kernel-based activation functions [17]. We have outlined our results comparing the performance of our networks to existing ReLU based architectures. After running experiments, we observe certain behavior related to activation functions as applied to the One-Shot Learning task:

1. We obtained better clusters (closely aligned) with KAF2D, followed by KAF and ReLU, for MNIST dataset.
2. The Training Loss Curve for AT&T Face dataset converged faster when using ReLU. When using KAF and KAF2D as activation functions, the loss fluctuated a bit in the beginning but provided a lower loss value at the end.
3. KAF takes around twice the training time of ReLU activation functions.
4. KAF2D takes around five times the training time of ReLU activation functions.
5. The results for Matching Networks architecture proved to be promising using Kernel-based activation functions.

We conclude that though new Activation Functions give better accuracy in matching networks, better similarity distance in AT&T dataset, and better intracluster scores for MNIST in less number of epochs, it takes a lot more time to converge as compared to ReLU.
Learning proper representations is an crucial part of any deep learning architecture, especially in case of one-shot learning when we have less amount of data. To learn the ability to learn, we need to improve out method of information extraction. In the above work, we proposed one method of data approximation using the Gaussian distribution activation function. For metrics-based one-shot learning, we can at least say that we need to learn proper approximation functions instead of adding more layers, as adding layers will lead to underfitting/overfitting the model considering less amount of data.

References

1. AT and T Laboratories Cambridge. The database of faces. *http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html*.
2. Luca Bertinetto, João F Henriques, Jack Valmadre, Philip Torr, and Andrea Vedaldi. Learning feed-forward one-shot learners. In *Advances in Neural Information Processing Systems*, pages 523–531, 2016.
3. Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. Signature verification using a" siamese" time delay neural network. In *Advances in Neural Information Processing Systems*, pages 737–744, 1994.
4. Sumit Chopra, Raia Hadsell, and Yann LeCun. Learning a similarity metric discriminatively, with application to face verification. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Sociey Conference on*, volume 1, pages 539–546. IEEE, 2005.
5. Haoqiang Fan, Zhimin Cao, Yuning Jiang, Qi Yin, and Chinchilla Doudou. Learning deep face representation. *arXiv preprint arXiv:1403.2802*, 2014.
6. Li Fei-Fei, Rob Fergus, and Pietro Perona. One-shot learning of object categories. *IEEE transactions on pattern analysis and machine intelligence*, 28(4):594–611, 2006.
7. S. Jadon. *Hands-on One-Shot Learning with Python: a Practical Guide to implementing Fast and Accurate Deep Learning Models with Fewer Training.* PACKT PUBLISHING LIMITED, 2019.
8. Shruti Jadon. Introduction to different activation functions for deep learning. *Medium, Augmenting Humanity*, 16, 2018.
9. Shruti Jadon and Mahmood Jasim. Video summarization using keyframe extraction and video skimming. *ArXiv*, abs/1910.04792, 2017.
10. Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. Siamese neural networks for one-shot image recognition. In *ICML Deep Learning Workshop*, volume 2, 2015.
11. Brenden Lake, Chia-ying Lee, James Glass, and Josh Tenenbaum. One-shot learning of generative speech concepts. In *Proceedings of the Cognitive Science Society*, volume 36, 2014.
12. Brenden Lake, Ruslan Salakhutdinov, Jason Gross, and Joshua Tenenbaum. One shot learning of simple visual concepts. In *Proceedings of the Cognitive Science Society*, volume 33, 2011.
13. Brenden M Lake, Ruslan Salakhutdinov, and Joshua B Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, 2015.
14. Brenden M Lake, Ruslan R Salakhutdinov, and Josh Tenenbaum. One-shot learning by inverting a compositional causal process. In *Advances in neural information processing systems*, pages 2526–2534, 2013.
15. Yann LeCun. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998.
16. Andrew Maas and Charles Kemp. One-shot learning with bayesian networks. Cognitive Science Society, 2009.
17. Simone Scardapane, Steven Van Vaerenbergh, and Aurelio Uncini. Kafnets: kernel-based non-parametric activation functions for neural networks. *arXiv preprint arXiv:1707.04035*, 2017.
18. Oriol Vinyals, Charles Blundell, Tim Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *Advances in Neural Information Processing Systems*, pages 3630–3638, 2016.

19. Di Wu, Fan Zhu, and Ling Shao. One shot learning gesture recognition from rgbd images. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*, pages 7–12. IEEE, 2012.

5 Supplementary Material

See Figs. 6, 7, 8

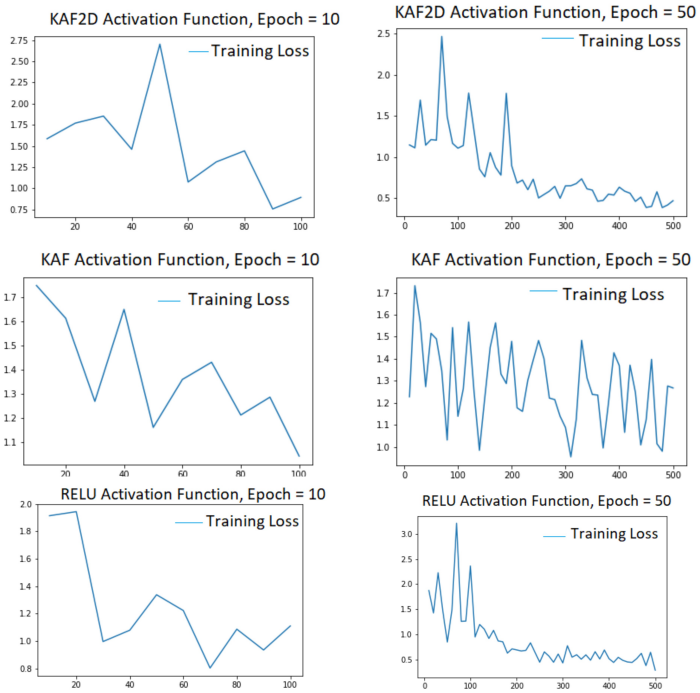

Fig. 6. Training Loss for Different Epochs AT&T Dataset

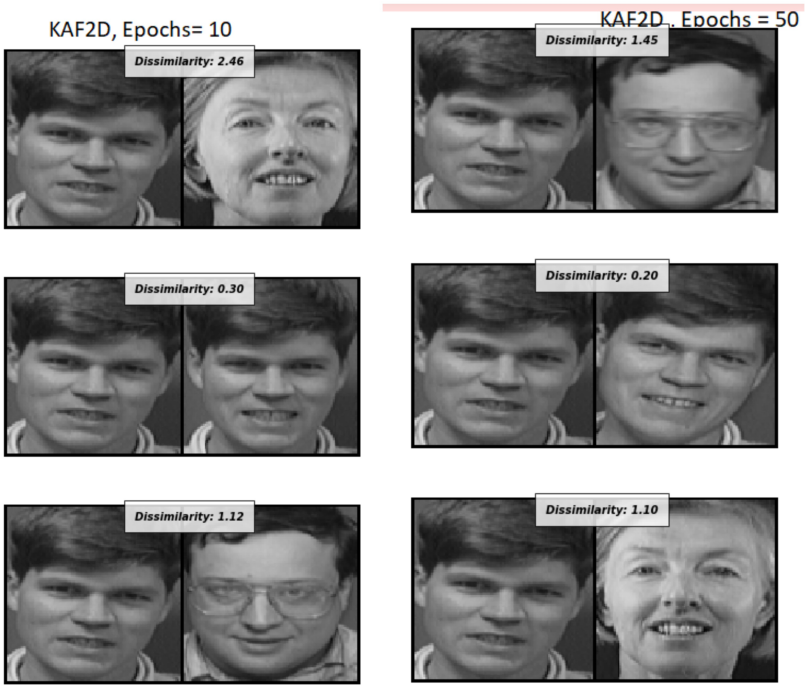

Fig. 7. Similarity Scores for Faces with KAF2D Activation Function [AT&T Dataset]

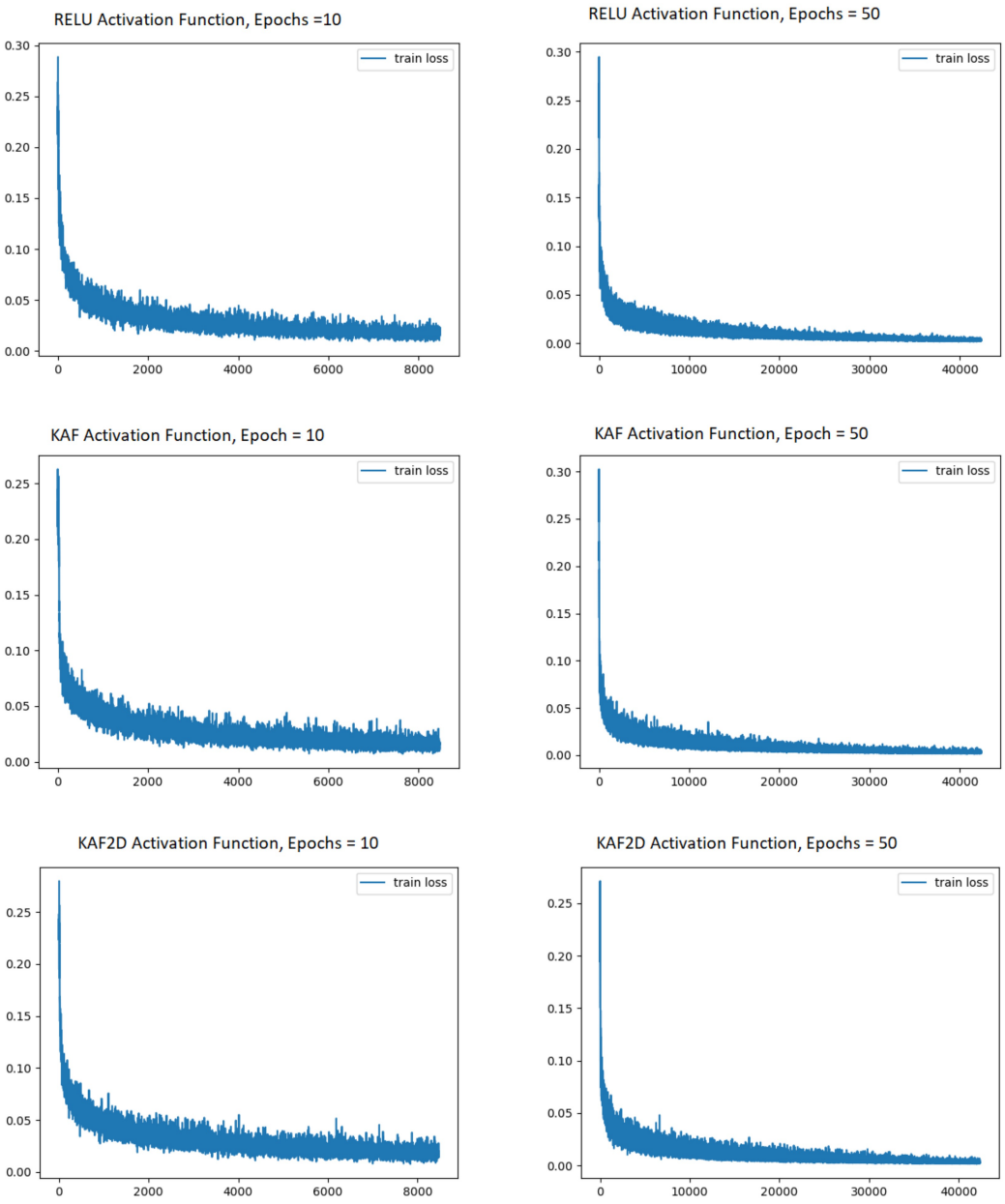

Fig. 8. Training Loss for MNIST dataset with different Epochs and Activation Functions.