EasyChair Preprint

№ 893

Siamese Networks for One Shot Learning using Kernel Based Activation functions

Shruti Jadon and Aditya Acrot Srinivasan

April 12, 2019

Siamese Networks for One Shot Learning using Kernel Based Activation functions

Shruti Jadon
Department of Computer Science
University of Massachusetts Amherst
Amherst, MA

Aditya Arcot Srinivasan
Department of Computer Science
University of Massachusetts Amherst
Amherst, MA

Abstract

The lack of a large amount of training data has always been the constraining factor in solving a lot of problems in machine learning, making One Shot Learning one of the most intriguing ideas in machine learning. It aims to learn information about object categories from one, or only a few, training examples, and for certain image classification tasks, has successfully been able to get results comparable to human beings. This project aims to deal with understanding the architecture of One Shot Learning using Siamese neural networks [1] and improve on their performance using Kafnets (kernel-based non-parametric activation functions for neural networks) [4]. We also intend to evaluate these activation functions for advanced one shot learning models(Matching Networks).

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 particular, when presented with stimuli, people seem to be able to understand new concepts quickly and then recognize variations on these concepts in future percepts [7]. Machine learning as a field has been highly successful at a variety of tasks such as classification, web search, image and speech recognition. Often times 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 an important role in the advancement of machine learning, but it also requires large datasets. Different techniques such as regularization reduces 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[1].

One-shot learning is an object categorization problem in computer vision. Whereas most machine learning based object categorization algorithms require training on hundreds or thousands of images and very large datasets, one-shot learning aims to learn information about object categories from one, or only a few, training images [10]. This is called one-shot learning and forms the basis of our work in this project.

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 project, we make use of an approach

similar to [1] while simultaneously evaluating different activation functions that may be better suited to this task. The overall strategy we apply is two fold; 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 nonparametric approach of evaluation, we came across Kafnets [4] (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 implemented these activations and compared their effectiveness against traditional ones when used in the context of One Shot learning.

2 Related Work

The research in one shot learning has not yet caught much attention of the machine learning community. The work resulting in the best accuracy for the image classification problem dates back to the 2000's by Li Fei-Fei et al. The authors have developed a variational bayesian framework [10] for one shot image classification using the premise that a previously learned class can help in forecasting a future one.

Lake et al. [8] tackled the problem of character recognition by proposing a method called Hierarchical Bayesian Program Learning. In [7] and [8], 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 lead to inference becoming intractable.

There have also been other methods that approach the problem of One Shot Learning. [16] tackle path planning as a one shot learning problem for robotic actuation. [15] use Bayesian networks on the Ellis Island passenger data to infer attributes. [9] use a generative Hidden Markov Model along with a Bayesian inference algorithm to try and identify unseen words in a speech recognition paradigm. [14] predicts the parameters of a neural network from a single exemplar image. The network that effectively learns to learn, generalizing across tasks defined by different exemplars.

A different approach to one-shot learning is to learn an embedding space, which is typically done with a siamese network [5]. Given an exemplar of a novel category, classification is performed in the embedding space by a simple rule such as nearest-neighbor. Training is usually performed by classifying pairs according to distance [6].

Another technique that looks at the problem of One Shot Learning is by use of matching networks or bi-directional LSTMs [2]. As mentioned before, non parametric alternatives like the Nearest Neighbours model choose an arbitrary distance function. The authors solve this problem by formulating a loss function that encompasses in training a nearest neighbour 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 Nearest Neighbors algorithm. The method progresses by embedding both the training examples as well as given test example $\hat{x}$, compute a cosine similarity based metric as the "match", and then pass that through a softmax to get normalized mixing weights to generate a label. The embedding process for the training examples make use of a bidirectional LSTM over the examples. For the test examples, is a an LSTM that processes for a fixed amount (K time steps) and at each point also attends over the examples in the training set. The encoding is the last hidden state of the LSTM. The paper also benchmarks various approaches to one shot learning could be used a reference for our results.

The approach that has been recently explored is the use of Deep Siamese Networks which we borrow from heavily [1]. 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 nonlinearities. 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 [3]. 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 which have a non parametric approach of evaluation, we came across a recent paper [4] on Kafnets (kernel based non-parametric activation functions) that has worked in this domain of training neural networks using different forms of activation functions. [4] introduce a novel family of flexible activation functions that are based on an inexpensive kernel expansion at every neuron. Leveraging over several properties of kernel-based models, the authors propose multiple variations for designing and initializing these kernel activation functions (KAFs), including a multidimensional scheme allowing to non linearly combine information from different paths in the network. The resulting KAFs can approximate any mapping defined over a subset of the real line, either convex or nonconvex. Furthermore, they are smooth over their entire domain, linear in their parameters, and they can be regularized using any known scheme. In this project, we focus on two activation functions, KAF and KAF2D and the effects of implementing them in a siamese architecture for One Shot Learning.

3 Data-Sets

For this project, We have used two main datasets: MNIST [11] and the AT&T Database of Faces [12] (formerly 'The ORL Database of Faces'). Further we also use the 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 we thought it would be appropriate to pick a dataset with more features to extract from images and decided to work with the AT&T Database of Faces.

The MNIST dataset consists of handwritten digits, has a training set of 60,000 examples, and a test set of 10,000 examples. It is a subset of a larger set available from NIST. The digits have been size-normalized and centered in a fixed-size image.

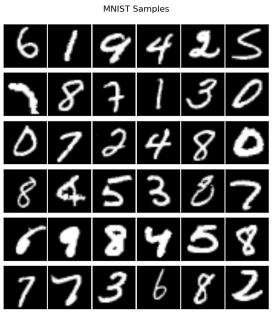

Figure 1: Sample MNIST Dataset

The AT&T Database of Faces consists of ten different images of each of 40 distinct subjects. For some subjects, the images were taken at different times, varying the lighting, facial expressions (open / closed eyes, smiling / not smiling) and facial details (glasses / no glasses). All the images were taken against a dark homogeneous background with the subjects in an upright, frontal position (with tolerance for some side movement).

Omniglot is a dataset by (Lake et al., 2015) that is specially designed to compare and contrast the learning abilities of humans and machines. The dataset contains handwritten characters of 50 languages (alphabets) with 1623 total characters. The dataset is divided into a background set and

Figure 2: Sample At&T Face Dataset

an evaluation set. Background set contains 30 alphabets (964 characters) and only this set should be used to perform all learning (e.g. hyper-parameter inference or feature learning). The remaining 20 alphabets are for pure evaluation purposes only. Each character is a 105 x 105 greyscale image. There are only 20 samples for each character, each drawn by a distinct individual.

4 Methodology

In this project, our first step was to train siamese networks to recognize similar and different forms of images with one or two examples of each form. Later we replace the the loss function layer, and instead use the last layer output as embeddings in a fixed dimensional space. Finally we check for the nearest data point and assign the appropriate category. We have used pytorch library for the implementation of Siamese Networks for Image classification. To maintain context, we take cues from the implementation of Matching Networks which use an LSTM architecture. One challenging task involved the implementation of activation functions. We implemented two Activation Functions, called KAF and KAF2D. KAF(Kernel Activation Function) Specifically, each activation function is modelled in terms of a kernel expansion over D terms as:

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

where, $\{\alpha_i\}_{i=1:D}$ are the mixing coefficients, $\{d_i\}_{i=1:D}$ are the called the dictionary elements, and $\kappa(.,.) : R \to R$ is 1D kernel function.

In kernel methods, the dictionary elements are generally selected from the training data. In a stochastic optimization setting, this means that D would grow linearly with the number of training iterations, unless some proper strategy for the selection of the dictionary is implemented. To simplify our treatment, a simplified case where the dictionary elements are fixed has been considered, where we only adapt the mixing coefficients. This has the additional benefit that the resulting model is linear in its adaptable parameters, and can be efficiently implemented for a mini-batch of training data using highly-vectorized linear algebra routines. Note that there is a vast literature on kernel methods with fixed dictionary elements, particularly in the field of Gaussian processes.

The kernel function need only respect the positive semi-definiteness property, i.e., for any possible choice of α_i and d_i we have that:

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

In this paper, they have used 1D Gaussian kernel defined as:

$$\kappa(s, d_i) = exp\{-\gamma(s - d_i)^2\}$$

where $\gamma \epsilon R$ is called the kernel bandwidth. This model has very straightforward derivatives for back-propagation as seen below:

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

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

[4] also considers a two-dimensional variant of the proposed KAF, denoted as 2D-KAF. Roughly speaking, the 2D-KAF acts on a pair of activation values, instead of a single one, and learns a two dimensional function to combine them. It can be seen as a generalization of a two-dimensional max-out neuron, which is instead constrained to output the maximum value among the two inputs. The equation is given 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.

After implementing Activation Functions, the next step was to implement Siamese Networks for Classification. The architecure is detailed in the next sections.

4.1 Architecture of Siamese Networks

For MNIST, we have coded a very simple 2 convolutional layer architecture:
Layer 1 : Conv1 and Conv2
Conv1: 1X20 followed by maxpooling
Conv2: 20X50 followed by maxpooling
Layer 2 : Fully Connected network with Activation Function. (50X500)
Layer 3 : Linear Layer (500X2)

For AT&T Face Dataset, we have implemented dense layered Architecture:

Layer 1 : Conv1, Conv2, and Conv3
Conv1: 1X4 followed by Activation Function and Max Pooling.
Conv2: 4X4 followed by Activation Function and Max Pooling.
Conv3: 8X8 followed by Activation Function and Max Pooling.
Layer 2 : Fully Connected network with Activation Function (100X100X8).
Layer 3 : Linear Layer with Activation Function. (500X250).
Layer 4 : Linear Layer with Activation Function. (250X5).
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.
We have set learning rate to 0.0005, and used the Adam Optimizer.

We have also changed the architecture for matching networks a bit so as to adapt the KAF2D Activation Function, but mostly used what was suggested in [2]. It consists of 4 convoluted layers, followed by 3 Fully connected linear layers and then Bidirectional LSTMs. This also uses the contrastive loss function.

5 Experiments and Results

We ran Siamese Network Architecture on the MNIST Dataset, with different activation functions, each 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. The reason, it is working better could be about the Activation function. Firstly, for one shot we need non-paramteric activation functions only, as we have less amount of data. RELU is simple traditional non-paramteric function, whereas KAF, and KAF2D were kernel based, so they could have contributed in capturing features of images in better way.

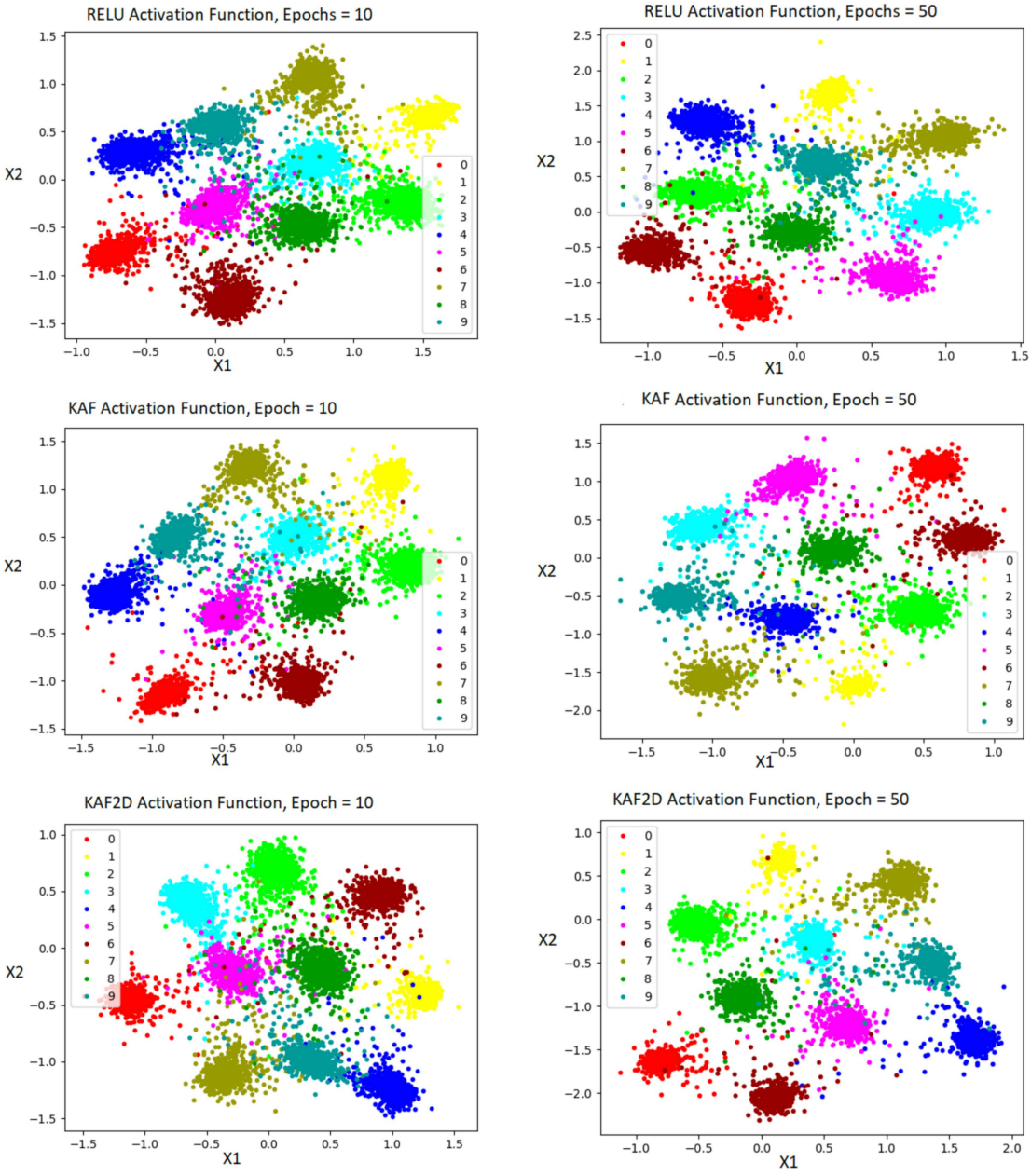

Figure 3: Embeddings with different Activation Function for different Epochs [MNIST Dataset]

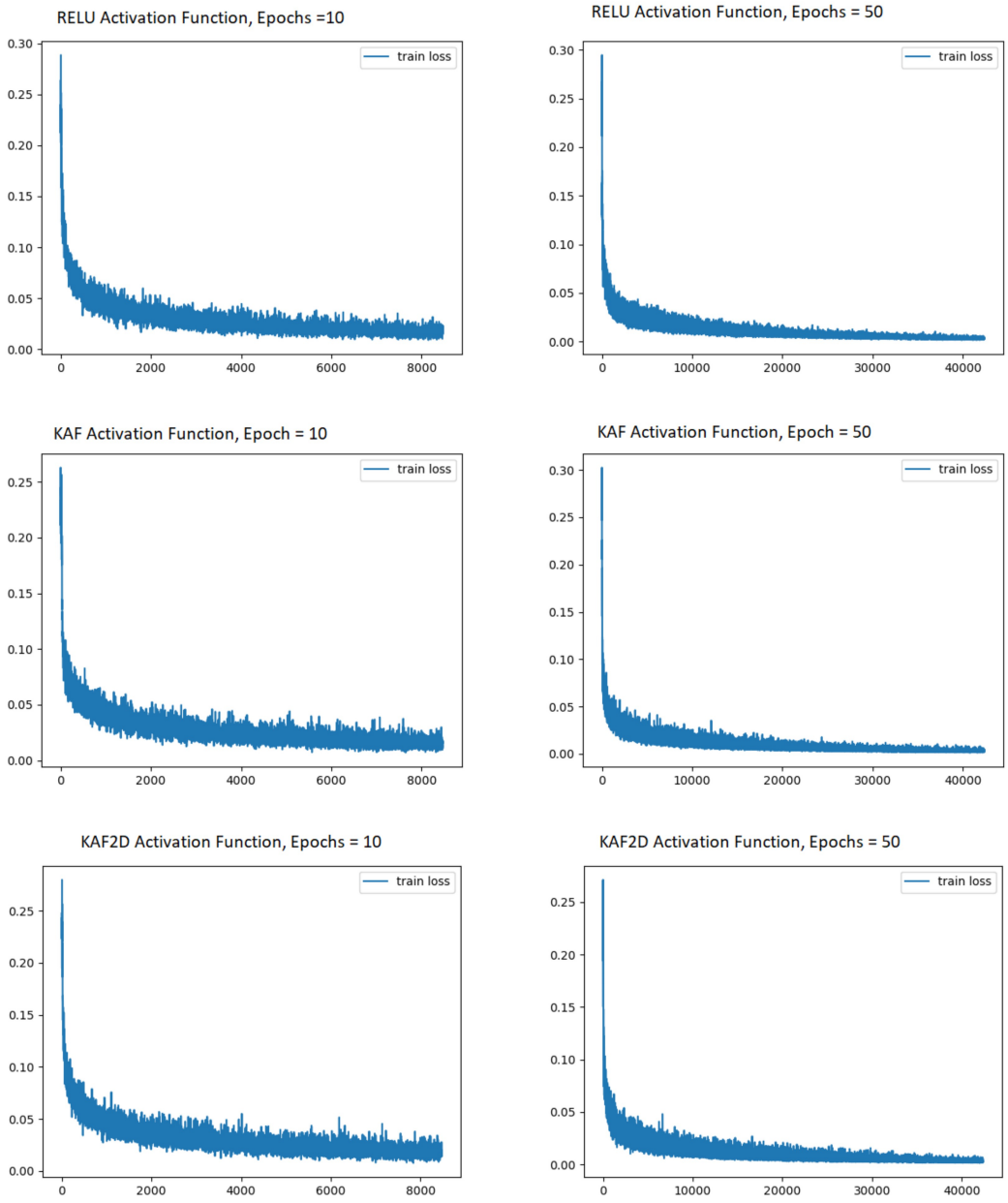

Figure 4: Training Loss for MNIST dataset with different Epochs and Activation Functions.

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.

We also experimented with the same architecture on the At&T Face Similarity dataset. As 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 the metric to measure similarity. Similar to the MNIST experiment, we ran it with KAF, KAF2D, and RELU activation function, and observed that we were able to increase the closeness of it, using KAF and KAF2D. We also observed the behavior of training loss curve with different functions. What we observed is, that for RELU, it converged much faster, which could be reason of its efficiency. Whereas KAF, and KAF2D were fluctuating in the beginning, but converged to a lower value of loss at the end.

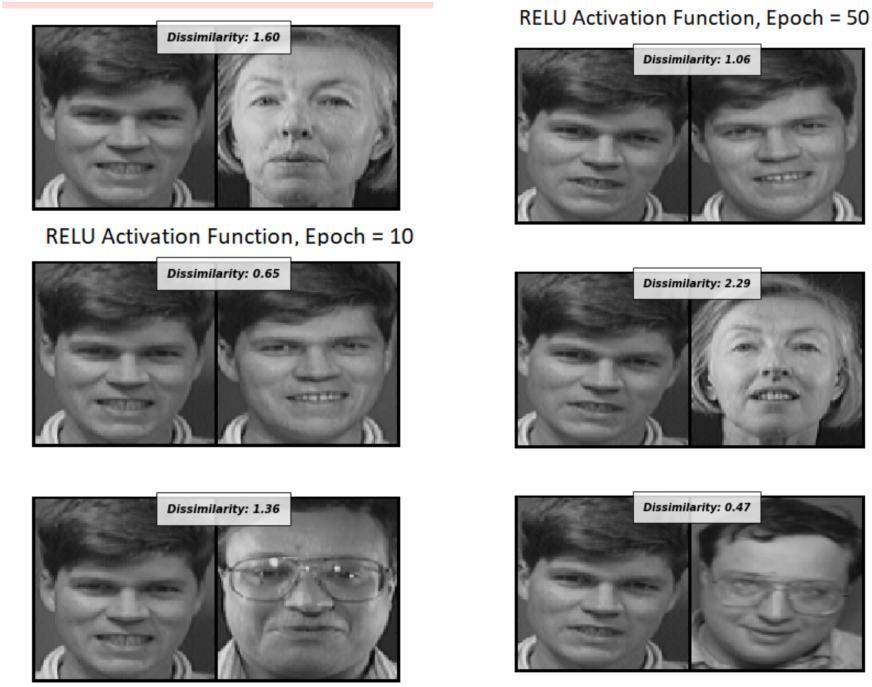

Figure 5: Similarity Scores for Faces with RELU Activation Function [At&T Dataset]

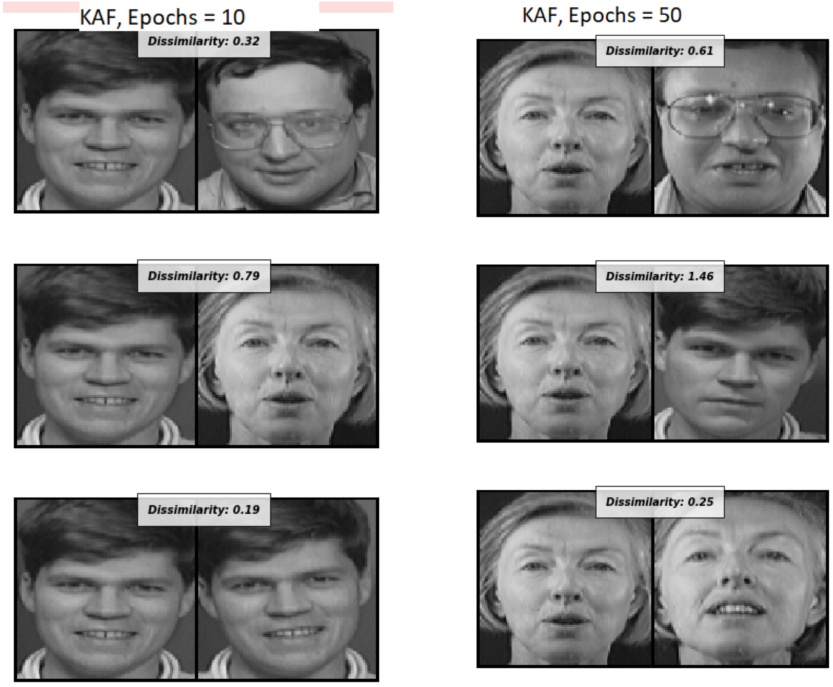

Figure 6: Similarity Scores for Faces with KAF Activation Function [At&T Dataset]

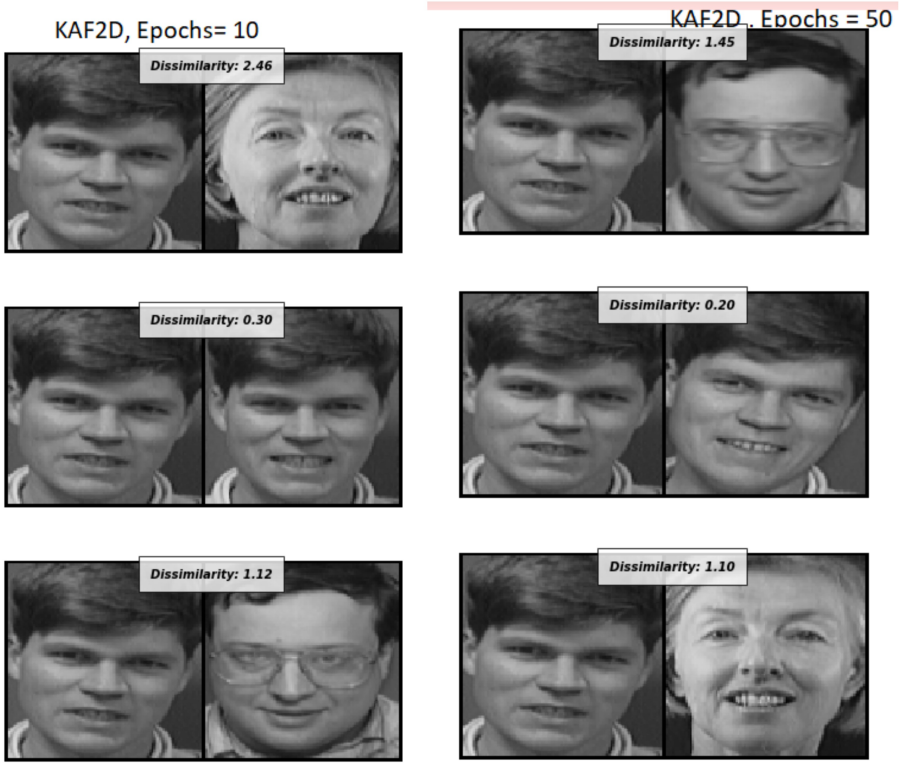

Figure 7: Similarity Scores for Faces with KAF2D Activation Function [At&T Dataset]

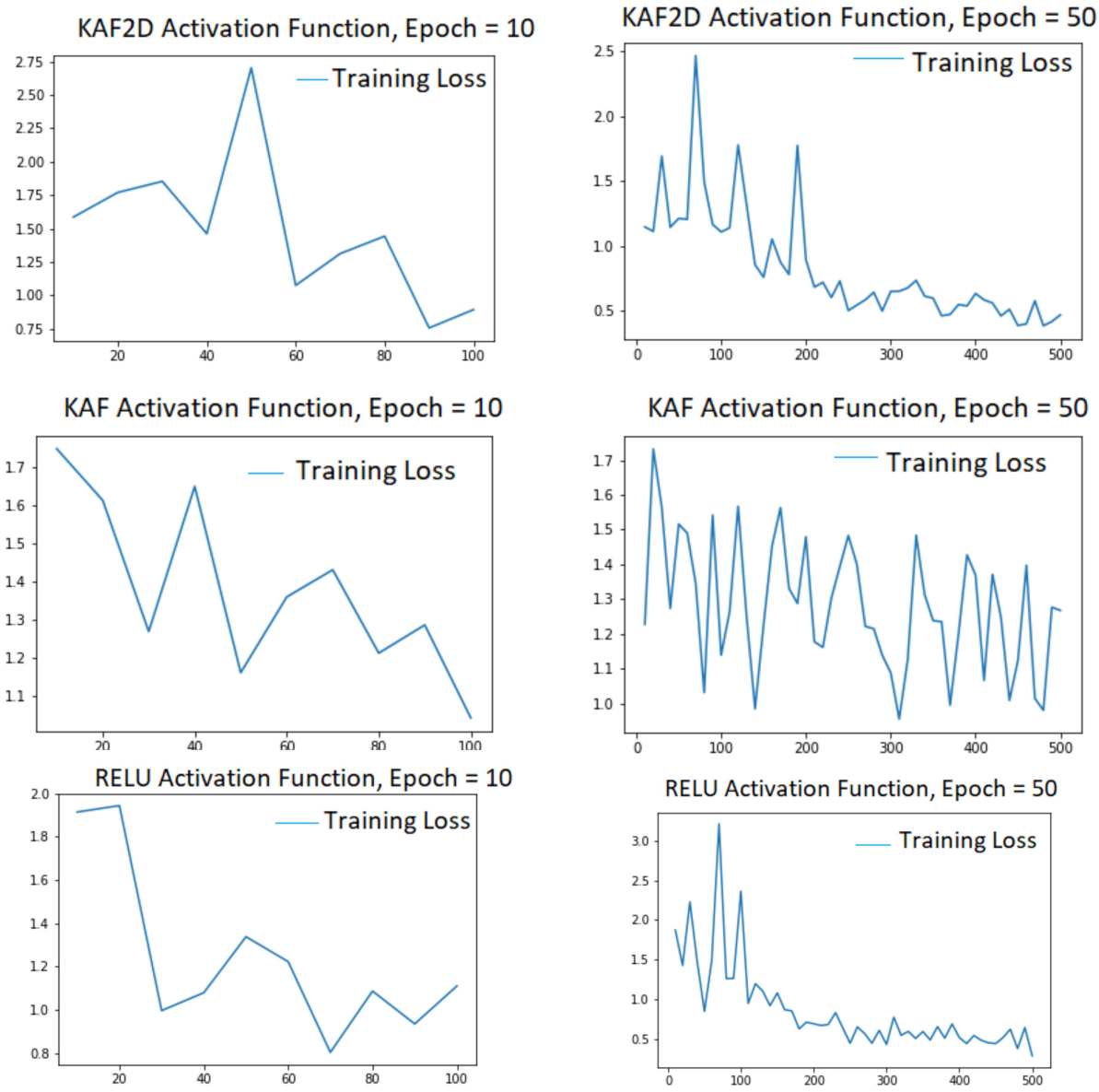

Figure 8: Training Loss for Different Epochs AT&T Dataset

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

Table 2: Accuracies for Matching Networks on Omniglot Dataset

Activation Function	Accuracy
RELU	91.18%
KAF	92.06%
KAF2D	89.27% Intermediate

5.1 Accuracies obtained on Matching Networks

As a final experiment, we replicated the architecture of Matching Networks as in [2] with the KAF and KAF2D activation functions on the Omniglot Dataset. It ran for 1600 Epochs with the results summarized in Table 2.

6 Discussion and Conclusions

In this project, we present kernel based activation functions [4] to improve the performance of one-shot classification by applying it to learn deep convolutional siamese neural networks for image classification. We have outlined our results comparing the performance of our networks to existing RELU based Architectures.

After running some experiments, we observe certain behavior related to activation functions as applied to the One Shot learning task:

1. KAF takes around twice the training time of RELU activation functions.
2. KAF2D takes around five times the training time of RELU activation functions.
3. We obtained better clusters (closely aligned) with KAF2D, followed by KAF and RELU, for MNIST Dataset.
4. 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.
5. The Results for Matching Networks Architecture proved to be promising using KAF based activation functions, but we weren't able to completely calculate the accuracy for KAF2D, as we ran out of AWS credits. For 1000 epochs we obtained 89.27% max.

We conclude that the new Activation Functions did giving better accuracy in matching networks, better similarity distance in AT&T dataset, and better intra cluster scores for MNIST, they took a lot more time to converge as compared to RELU.
In the future, we can test this approach for Neural Turning Machines Algorithm of One Shot Learning, but as mentioned it will require more resources. For now, we can conclude, that these activation functions show promise, but if used in complex and large architectures, it will cost a lot of resources and time. We can use them for small architectures, just like with MNIST's (2 convolutional Layer) and AT&T's (3 convolutional Layer). But when it comes to Matching Networks Architecture(which involves LSTMs) we can trade off accuracy for flexibility in time and resources.

References

[1] Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. Siamese neural networks for one-shot image recognition. In *ICML Deep Learning Workshop*, volume 2, 2015.

[2] 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.

[3] 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 Society Conference on*, volume 1, pages 539–546. IEEE, 2005.

[4] Simone Scardapane, Steven Van Vaerenbergh, and Aurelio Uncini. Kafnets: kernel-based non-parametric activation functions for neural networks. *arXiv preprint arXiv:1707.04035*, 2017.

[5] 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.

[6] Haoqiang Fan, Zhimin Cao, Yuning Jiang, Qi Yin, and Chinchilla Doudou. Learning deep face representation. *arXiv preprint arXiv:1403.2802*, 2014.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] Yann LeCun. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998.

[12] ATT Laboratories Cambridge. The database of faces. *http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html*.

[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] 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.

[15] Andrew Maas and Charles Kemp. One-shot learning with bayesian networks. Cognitive Science Society, 2009.

[16] 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.