Comparison of loss functions in One shot learning using Siamese Neural networks

Muhammad Moeed Khalid
Computer Science
University of Windsor
Windsor ON, Canada
khalid21@uwindsor.ca

Ahmed Shafeek
Computer Science
University of Windsor
Windsor ON, Canada
ashafeek@uwindsor.ca

***Abstract*—Data gathering and formulation is still a big problem in the machine learning world. While using Deep Neural Networks with datasets such as Imagenet give high-end accuracies, that is largely due to the fact that they have had millions of images to train on. In most real world scenarios gathering such enormous amounts of data is not feasible. One of the ways to counter this problem is One shot learning, where we try and make predictions based on N number of images of a class. We do so by using N-Way One shot learning using Siamese Neural Networks. We compare the accuracies across different distance and loss functions.**

***Index Terms*—Siamese networks, one shot learning, twin neural networks, contrastive, cross entropy**

I. Introduction

Machine learning specifically deep learning algorithms are more powerful and more accurate than most regular algorithms. This has been proven time and again [1, 2]. In fact more and more machine learning algorithms are getting human-esque accuracy this days. But this cannot be credited to machine Learning algorithms alone. Most of these algorithms would perform poorly relative to their current results had the datasets sizes been smaller [3].

One way that scientists have wanted to solve this problem for a long time is via incorporation of more statistical methodologies to the existing deep neural networks [4]. This has become more prevalent with the introduction of Siamese Neural networks[5]. In Siamese neural networks (also know as *twin* neural networks) we use two identical Neural networks or deep neural networks in tandem, they have the same weights, the have the same hyper-parameters, the only thing they differ in are the input vectors that they are working on. These images are a combination of pairs that either belong to the same or different class. At the end of both the networks rather than calculating probabilities we calculate the distances between both the output vectors and assign them ranking based on that. This problem has been tackled in recent times using One shot learning as well, where the model is trained on only sample of a class rather than hundreds of thousands if images of that class. Hence the name *one-shot* learning [6].

Using Siamese Neural networks with one shot learning [7] and comparing and contrasting different loss functions with different distance functions gives us and idea of what loss functions pairs well with what distance functions. We will be working on an initially built Siamese neural network that was using only the L1 distance and Binary Cross Entropy. We extend it to further use L2 distance as well as Contrastive loss function to make a comparison between the different pairs of loss and distance functions.

II. Motivation and Problem Statement

Machine learning algorithms specifically neural networks are becoming more and more popular as they consistently outperform their counterparts. But this comes with its drawbacks such as: They need huge amounts of data to train on; Their complex structure means they need quite a while to train and the humongous amounts of data adds to that computational complexity. We want to tackle this issue by converting a classification problem to a distance calculation problem using Siamese Neural networks.

III. Related Work

The concept of Siamese Neural networks was first introduced by Bromley et al [5], where they used it detect the forgery of signatures. This is the most basic type of siamese neural network which used two input vectors of signatures, calculated their distance and gave a binarized output on whether or not the signature was forged.

After this a lot of work has been done with Siamese Neural Networks, specially in Image recognition as it is a very computationally intensive field and gathering a large number of Images is not always a feasible approach [5, 7].

Siamese Neural Networks are used in Natural Language processing problems as well. Kumar et al. used a Siamese recurrent Neural Network that had two bidirectional recurrent neural networks to detect click-bait from around 20,000 posts which is a relatively small number of posts according to the current Norm of Natural Language Processing Fields.

Looking at the success Siamese Neural networks have on images, it is only natural that we apply the same concepts on videos as well. Ryoo et al. [8] uses low resolution videos that are around 12x16 pixels, Their approach for classification of objects in low resolution videos using Multi-SNN outperforms other similar methods.

One shot learning is combined with Siamese Neural Networks as a very feasible approach to tackling this smaller dataset

issue [7]. One shot learning is a relatively new field with more and more work being done on advanced problems such as COVID-19 as well [9].

IV. Methods, Algorithms, Solutions

We are going to be comparing our results from N-way one shot learning using Siamese neural networks with K-NN which we consider as a baseline for our experiments. We also generate random weights to compare our results against as that gives us an idea that our models are in actuality learning something rather than giving superficial results.

A. K-Nearest Neigbour

K-NN is a machine learning method that classifies points based on their similarity to each other [10]. Where k is the number of points the algorithm can consider while making its decisiion using the following fromula:

$$C(\hat{x}) = argmin||\hat{x} - x_c|| \quad (1)$$

By tradition and common practice we use Eucledian distance whislt we look for points but that can be modified according to the need of the situation. For the sake of our experiments we are using 1 as the value of our k in our K-NN algorithm.

B. Siamese Neural Networks

Statistics play a significant role in the creation and optimization of neural networks [11]. But there have always been efforts to enhance this co relation between statistics and Neural networks as it may give us more control to fiddle around with neural networks. Siamese Neural Network (also known as a twin neural network) is a big leap in that direction [4]. A Siamese neural networks consists of two identical neural networks that work on (in most cases) a pair of input vectors to output a hidden feature vector for each of them. These hidden feature vectors are updated again and again by comparing them with the ground truth and making adjustments accordingly. This method is known as back-propagation [12].

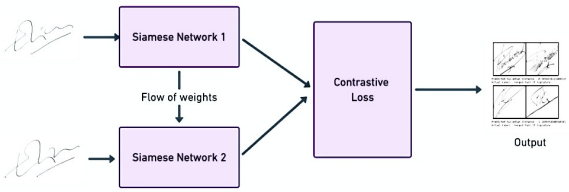
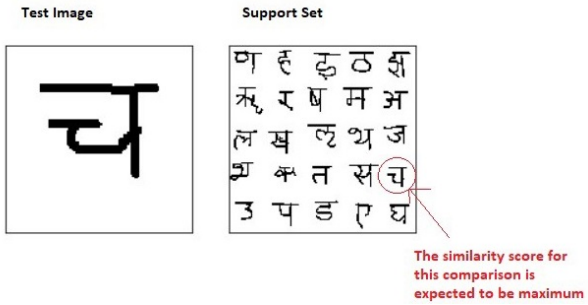


Fig. 1. Siamese Neural networks

Once we have output vectors for both our inputs we can use them to calculate any measures we want to such as Precision, Recall or ROC Curve etc. These vectors are then used to calculate their distances from each other. Any distance function such as Manhattan (L1) [13] or Euclidean (L2) distance [14] can be used for this purpose.
Now we need to define the loss function that we want to use for this purpose. These can be paired with any loss functions that we want but for our experimentation we wanted to stuck with Binary Cross Entropy [15] and Contrastive loss[4] to find out which one works better in our case.

C. One Shot Learning

Using Siamese Neural Network we use one shot learning rather than classification. In a way that converts our problem from a classification problem to a distance problem.
In one shot learning we train our network from one input only [6]. This is contrary to the usual methodology of training Neural networks which require hundreds of thousands of inputs to train on. How does this exactly work?

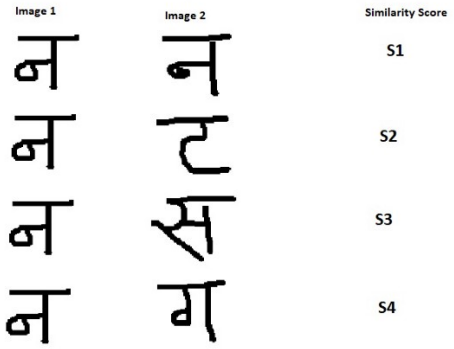


,

Fig. 3. One shot Learning

Rather than taking one input image and classify it based on the output score of the network, our network takes in two images and will give them a score based on their similarity. This is usually follows by an output binarizer that normalizes the score between 0 and 1.

1) N-Way One Shot Learning: An extension of One Shot learning is N-Way one shot learning where one input image is paired with multiple images out of which only one of the images is positive and the rest of them are negative.



,

Fig. 4. 4-way One shot Learning

Lets say we want to do 4-way one shot learning, we will create 4 pair of images and calculate their scores. Intrinsically the positive pair of images should have better score than the rest of the 3 images.

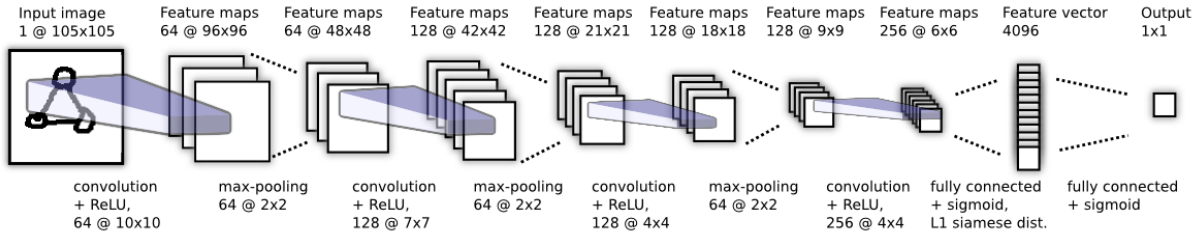


Fig. 2. Architecture of our Neural Network

D. Euclidean distance

Also known as L2 distance, the Euclidean distance between two points is the distance of a line drawn between those two points [14]. It is calculated by calculating the difference between the co-ordinates of two points and taking their absolute value. The simplest form of Euclidean distance is given as.

$$d(p,q) = |q - p| \tag{2}$$

But we usually generalize this formula to cater to higher dimensions and it is given as:

$$d(p,q) = \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2} \tag{3}$$

E. Manhattan Distance

Also known as L1 distance, the Manhattan distance between 2 points is calculated by taking the difference of the Cartesian points in each dimension and adding them [13]. The formula for Manhattan distance is given as:

$$\sum_{i=1}^{n} |x_i - y_i| \tag{4}$$

F. Binary Cross Entropy

Binary Cross Entropy loss is a loss function that we use to calculate the distance between the true and predicted values[15]. It follows the logic that if the predicted class probability is close to the actual value it will be assigned the value of 0, otherwise it will be assigned the value of 1. The formula for cross entropy is given as:

$$\sum_{i=1}^{2} t_i \log(p_i) \tag{5}$$

G. Contrastive Loss

Contrastive loss is another loss function that we use in our experiments that calculates the loss by taking the difference between a positive and a negative sample, and two positive samples and then compares them. We use an extra parameter called margin, denoted by τ, which is the degree to which we want the input vector to resemble the test vector. The formula for Contrastive loss is given as:

$$y * Sim(y, \hat{y}) + (1 - y) * max(\tau - Sim(y, \hat{y}), 0)^2 \tag{6}$$

V. Experiments and Discussion

A. Dataset

We perform our experiments on the Omniglot dataset [16]. The dataset contains 1623 handwritten characters from 50 different alphabets. These characters were drawn on Amazon's Turk by 20 different people. This dataset was created for the purpose of One shot learning so it fits our needs well.

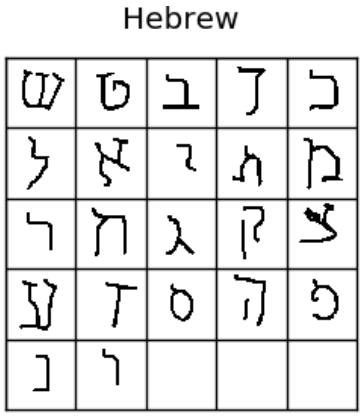


Fig. 5. Example (Hebrew language) image from the Omniglot Dataset

,

B. Experiment, Framework, Performance Measures

1) Experiment:

2) Framework and Architecture: The framework we decided to use in this paper is the same as used in Goch et al. [7]. We have max-pool and three convolutional layers in our network. Where as the output layers are couple of fully connected Sigmoid layers that use either L1 or L2 distances and then binarizes the output to give a result of either 0 or 1.

C. Results and Discussion

We tested our Siamese neural network with the by using Contrastive Loss with Euclidean distance and Binary cross entropy loss with Manhattan distance. We then compare the results.

These calculations were done with batch size of 32 and 1000 epochs. The original paper from which we were following the parameters and the hyperparameters used 20,000 epochs and different batch sizes, but we did not have the computational power to run such a large number of iterations, so instead we based our experiments on a fixed number of batch size and epochs.

TABLE I
COMPARISON OF ACCURACY AND LOSSES OF DIFFERENT METHOD COMBINATIONS

Distance + Loss Function	Loss	Accuracy (%)			
		3-way	11-way	20-way	mean[c]
L1 + Binary Cross Entropy	0.90	88.0	78.0	61.0	75.6
L2 + Contrastive Loss	0.047	64.0	64.0	44.0	57.3

We achieved a mean accuracy of 75% and a loss of 0.90 using L1 loss with Binary cross entropy. This accuracy was significantly higher than that of L2 distance paired with contrastive loss with a margin of 0.02. But the total loss achieved by the latter was 0.004 which was exponentially lesser than that of L1 + BCE.

1) L1 with Binary Cross Entropy: In 1000 iterations, the loss significantly reduced from 1.7 to 0.90, and the accuracy for 20 way one shot learning also increased from 48.4% to 61%. More iterations would have probably resulted in a more significant increase but computational limitations hindered us from doing that.

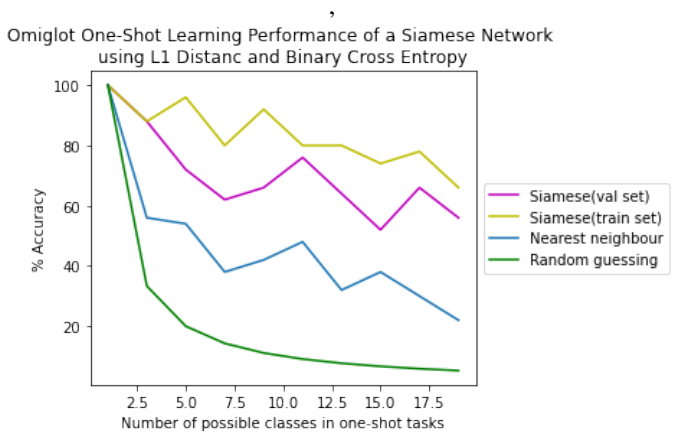


Fig. 6. Comparison of accuracy b/w Siamese Neural Network using L1 distance and Binary Cross Entropy Loss Function With 1-NN and Random Guessing

2) L2 + Contrastive Loss: In 1000 iterations, the loss significantly reduced from 0.7 to 0.04, where the accuracy increased from 8% to 44%. One might think that the lower the loss the higher the accuracy will be, but the actual concept is actually counter intuitive and is not always the case. This direct relation between the accuracy and loss whilst using this network made it evident that the model we had did not suit well with this combination of loss function and distance function.

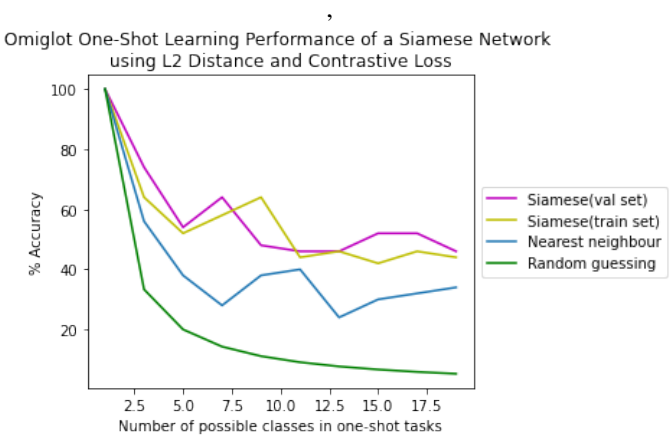


Fig. 7. Comparison of accuracy b/w Siamese Neural Network using L2 distance and Contrastive Loss Function With 1-NN and Random Guessing

VI. CONCLUSION AND FUTURE WORK

We concluded from our work that just because Contrastive loss is a more commonly paired loss function with Siamese Neural network these days does not mean it will always perform well as compared to the more common loss functions such as Binary Cross Entropy. It has a high dependency on the architecture of the neural network that we are using as well as the hyperparameters we are using to train and test it.
For future work we want to extend this experiment to triplet loss as well and see how that weighs in against these distance and loss functions.

REFERENCES

[1] A. Valier, "Who performs better? avms vs hedonic models," *Journal of Property Investment & Finance*, 2020.

[2] C. F. Tsai and S. P. Wang, "Stock price forecasting by hybrid machine learning techniques," in *Proceedings of the international multiconference of engineers and computer scientists*, vol. 1, no. 755, 2009, p. 60.

[3] J. G. A. Barbedo, "Impact of dataset size and variety on the effectiveness of deep learning and transfer learning for plant disease classification," *Computers and electronics in agriculture*, vol. 153, pp. 46–53, 2018.

[4] D. Chicco, "Siamese neural networks: An overview," *Artificial Neural Networks*, pp. 73–94, 2021.

[5] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, and R. Shah, "Signature verification using a" siamese" time delay neural network," *Advances in neural information processing systems*, vol. 6, pp. 737–744, 1993.

[6] O. Vinyals, C. Blundell, T. Lillicrap, K. Kavukcuoglu, and D. Wierstra, "Matching networks for one shot learning," *arXiv preprint arXiv:1606.04080*, 2016.

[7] G. Koch, R. Zemel, and R. Salakhutdinov, "Siamese neural networks for one-shot image recognition," in *ICML deep learning workshop*, vol. 2. Lille, 2015.

[8] M. Ryoo, K. Kim, and H. Yang, "Extreme low resolution activity recognition with multi-siamese embedding learn-

ing," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.

[9] V. M. Aradhya, M. Mahmud, D. Guru, B. Agarwal, and M. S. Kaiser, "One-shot cluster-based approach for the detection of covid–19 from chest x–ray images," *Cognitive Computation*, pp. 1–9, 2021.

[10] L. E. Peterson, "K-nearest neighbor," *Scholarpedia*, vol. 4, no. 2, p. 1883, 2009.

[11] H. S. Stern, "Neural networks in applied statistics," *Technometrics*, vol. 38, no. 3, pp. 205–214, 1996.

[12] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[13] S. Craw, *Manhattan Distance*. Boston, MA: Springer US, 2017, pp. 790–791. [Online]. Available: https://doi.org/10.1007/978-1-4899-7687-1_511

[14] P.-E. Danielsson, "Euclidean distance mapping," *Computer Graphics and image processing*, vol. 14, no. 3, pp. 227–248, 1980.

[15] P.-T. De Boer, D. P. Kroese, S. Mannor, and R. Y. Rubinstein, "A tutorial on the cross-entropy method," *Annals of operations research*, vol. 134, no. 1, pp. 19–67, 2005.

[16] B. M. Lake, R. Salakhutdinov, and J. B. Tenenbaum, "Human-level concept learning through probabilistic program induction," *Science*, vol. 350, no. 6266, pp. 1332–1338, 2015. [Online]. Available: https://science.sciencemag.org/content/350/6266/1332