Similarity Learning for Matching Hand Written Digits

Waheed Moonwar *, Mohammad Rakibul Hasan Mahin †, Fahmid Bin Kibria ‡
Aurnab Biswas §, Fatiha Ishrar Chowdhury ¶, Sifat E Jahan ‖, Annajiat Alim Rasel **
Department of Computer Science and Engineering, Brac University, Dhaka, Bangladesh
*waheed.moonwar@g.bracu.ac.bd, †mohammad.rakibul.hasan.mahin@g.bracu.ac.bd, ‡fahmid.bin.kibria@g.bracu.ac.bd
§aurnab.biswas@g.bracu.ac.bd, ¶fatiha.ishrar.chowdhury@g.bracu.ac.bd, ‖sifat.jahan@bracu.ac.bd, **annajiat@gmail.com

***Abstract*—Using deep learning to match handwritten digits can be a big help to the banking system and the forensic system. Modern machine learning methods like content-based information retrieval and deep learning can be used on these types of images because they can handle very large data sets and use them to find hidden structure and make accurate predictions. This information could make the traditional text matching work better, but it would also make the process harder to figure out. In this work, we propose a way to match handwritten digits using transfer learning. Transfer learning lets the model use what it already knows to get better predictions.**

***Index Terms*—similarity learning, transfer learning, hand digit recognition**

I. Introduction

It's helpful to be able to quickly identify items with similar properties. One efficient method that can be used for this purpose is similarity searching. Computer scientists can use contrastive learning to construct similarity methods for similarity searching.

When similarity models are trained, they produce embeddings in which items are placed in a metric space. This makes it so that similar things are close to each other and different things are farther away. This is related, both intuitively and mathematically, to word embeddings, which you already know about. For example, Vienna and Amsterdam are close to each other, as are mayonnaise and mustard. However, these two groups are more spread out than the other two.

Learning a pairwise similarity measure from data is a fundamental task in machine learning. Pair distances underlie classification methods like nearest neighbors and kernel machines, and similarity learning has important applications for "query-by-example" in information retrieval. For instance, a user may wish to find images that are similar to (but not identical copies of) an image she has; a user watching an online video may wish to find additional videos about the same subject. In all these cases, we are interested in finding a semantically-related sample, based on the visual content of an image, in an enormous search space. Learning a relatedness function from examples could be a useful tool for such tasks.

II. Related Work

There has been a lot of research into learning similarity via relative relevance, and some new methods have been developed to tackle learning on a wide scale. There are two primary categories of similarity learning methods when dealing with limited amounts of data. The first method, called learning Mahalanobis distances, is learning a linear projection of the data into a new space (often one with less dimensions) in which the distance between any two points is specified in terms of the Euclidean distance. The methods of Fisher's Linear Discriminant Analysis (LDA), Relevant Component Analysis (RCA) [1], Supervised Global Metric Learning (GML) [2], Large Margin Nearest Neighbor (LMNN) [3], and Metric Learning via Collapsed Classes (MLCC) [4] are all examples of this type of analysis (MLCC). The learnt metric may also be subject to other limitations, such as sparseness [**?**]. For a more in-depth analysis, see out [5].

To enhance the functionality of kernel-based classifiers, a second family of methods, known as learning kernels, is employed. Without a large amount of data, it is difficult to learn a whole kernel matrix nonparametrically. Alternatively, some studies [6] have proposed learning a weighted sum of pre-defined kernels, with the weights being learned from data. This has been found to perform poorly compared to a uniform weighting of the kernels in several scenarios . The research presented in [7] goes a step further by learning a weighting over local distance functions for each image in the training set. Dimensionality reduction has also been applied to the study of non linear picture similarity learning, as in [8].

III. Proposed Methodology

A. Similarity Learning

For AI researchers, "similarity learning" is a sub-field of supervised machine learning. Learning a similarity function that quantifies the degree to which two things are connected is the purpose of this technique, which is related to regression and classification. It can be used in rating and recommendation systems, in addition to visual identification detection, facial verification, and speech verification. When investigating the variety of applications that emerge from the use of data science, computer vision, and deep learning techniques, one crucial fact becomes apparent: many of these techniques are built on the principle of measuring similarity between any two vectors. These vectors can stand in as symbols for the things under consideration. Since this is the case, similarity measurement becomes crucial in the development of machine learning or deep learning methods. In a similar manner, whether a supervised or unsupervised learning job must be completed, an algorithm is needed to do so effectively.

B. Deep Learning Models

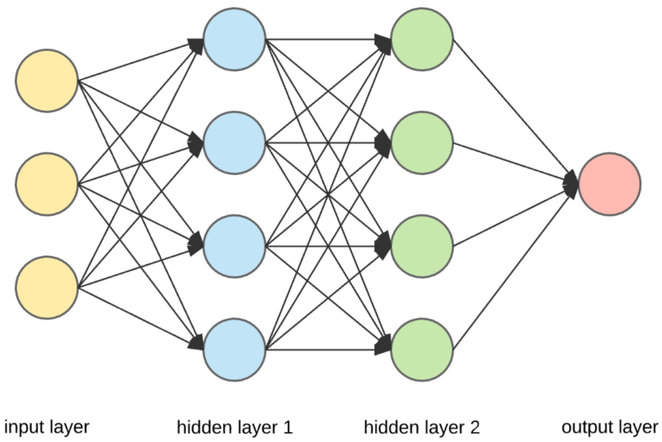

Fig. 1. ANN Architecture

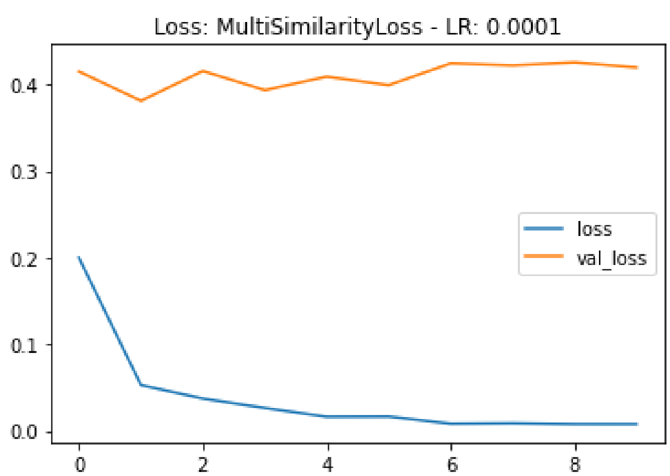

Fig. 2. Loss Curve for ANN

1) Artificial Neural Network: As an alternative to statistical methods for the classification of remote sensing data, the use of artificial neural networks (ANN) is becoming increasingly common [9] in today's modern world. In order to classify the land use and land cover of Landsat images, the primary focus of this study is on two algorithms of artificial neural networks (ANN), namely back propagation and k-means. The image classification task was addressed by these algorithms using a variety of different methodologies.
2) Convolutional Neural Network: One of the most popular models for use in deep learning is a convolutional neural network. When compared to other models, Convolutional neural networks have proven to be particularly effective at classifying images. As part of this paper, they developed a

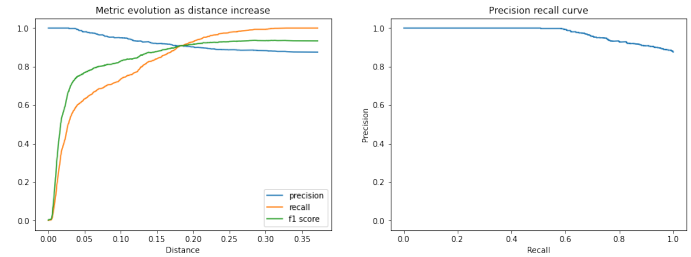

Fig. 3. Metric Evolution & Precision Recall for ANN

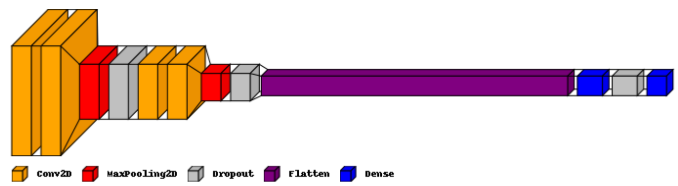

Fig. 4. CNN Architecture

basic Convolutional neural network for classifying images. An basic [10] Convolutional neural network did the job of classifying the images. The minist [11] and cifar-10 databases serve as the foundation for their experiments. They also investigated various techniques of learning rate set and various optimization algorithms for solving optimal parameters of the influence on picture classification, all on the basis of the Convolutional neural network.

IV. Experiments

A. Experimental Setup and Training Details

Initial batch size is 36(6 classes * 6 examples per class). We used 10 epochs because of less resources. Loss rate was 0.0001.

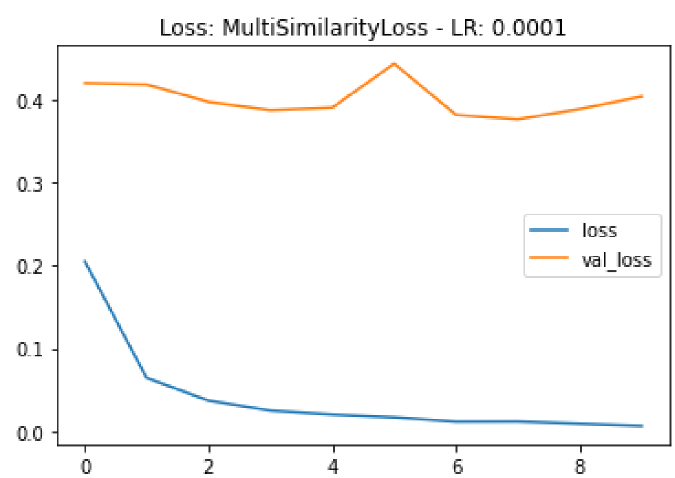

Fig. 5. Loss Curve for CNN

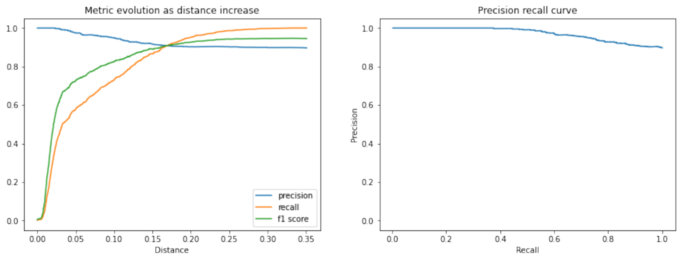

Fig. 6. Metric Evolution & Precision Recall for CNN

```
Epoch 1/10
1000/1000 [==============================] - 12s 7ms/step - loss: 0.2000 - val_loss: 0.4149
Warmup complete
Epoch 2/10
1000/1000 [==============================] - 7s 7ms/step - loss: 0.0529 - val_loss: 0.3812
Epoch 3/10
1000/1000 [==============================] - 7s 7ms/step - loss: 0.0373 - val_loss: 0.4155
Epoch 4/10
1000/1000 [==============================] - 7s 7ms/step - loss: 0.0264 - val_loss: 0.3935
Epoch 5/10
1000/1000 [==============================] - 7s 7ms/step - loss: 0.0164 - val_loss: 0.4090
Epoch 6/10
1000/1000 [==============================] - 6s 6ms/step - loss: 0.0166 - val_loss: 0.3992
Epoch 7/10
1000/1000 [==============================] - 7s 7ms/step - loss: 0.0083 - val_loss: 0.4243
Epoch 8/10
1000/1000 [==============================] - 7s 7ms/step - loss: 0.0088 - val_loss: 0.4220
Epoch 9/10
1000/1000 [==============================] - 7s 7ms/step - loss: 0.0079 - val_loss: 0.4254
Epoch 10/10
1000/1000 [==============================] - 6s 6ms/step - loss: 0.0078 - val_loss: 0.4196
```

Fig. 7. Epoch used for ANN

B. Dataset

The MNIST database of handwritten digits, has a training set of 60,000 examples, and a test set of 10,000 examples. It's a small part of the full NIST collection. The digits are now uniformly sized and centered in a specified image size. This standard dataset of handwritten images was first made accessible in 1999 and has been used as a standard for evaluating classification methods ever since. Even as more advanced machine learning methods become available, MNIST continues to serve as a trusted resource for both researchers and students.

C. Performance Evaluation & Discussion

From Fig. 3 and Fig. 6 represents the metric evolution of ANN architecture and CNN architecture respectively. Both of them have similar performance even though, CNN performance is slightly better. We prefer CNN in this case for matching digits.

```
Epoch 1/10
1000/1000 [==============================] - 8s 7ms/step - loss: 0.2049 - val_loss: 0.4203
Epoch 2/10
1000/1000 [==============================] - 7s 7ms/step - loss: 0.0644 - val_loss: 0.4186
Epoch 3/10
1000/1000 [==============================] - 7s 7ms/step - loss: 0.0370 - val_loss: 0.3976
Epoch 4/10
1000/1000 [==============================] - 7s 7ms/step - loss: 0.0248 - val_loss: 0.3875
Epoch 5/10
1000/1000 [==============================] - 7s 7ms/step - loss: 0.0200 - val_loss: 0.3906
Epoch 6/10
1000/1000 [==============================] - 7s 7ms/step - loss: 0.0167 - val_loss: 0.4438
Epoch 7/10
1000/1000 [==============================] - 7s 7ms/step - loss: 0.0115 - val_loss: 0.3819
Epoch 8/10
1000/1000 [==============================] - 7s 7ms/step - loss: 0.0116 - val_loss: 0.3765
Epoch 9/10
1000/1000 [==============================] - 7s 7ms/step - loss: 0.0089 - val_loss: 0.3889
Epoch 10/10
1000/1000 [==============================] - 7s 7ms/step - loss: 0.0063 - val_loss: 0.4041
```

Fig. 8. Epoch used for CNN

Models	Binary Accuracy Score	Recall	f1 - score	Precision
ANN	0.869	0.993143	0.935918	0.884929
CNN	0.897	1	0.946702	0.898798

TABLE I
COMPARATIVE PERFORMANCE EVALUATION.

Loss Curves are more stable in CNN's case. In the precision-recall curve, when the recall reaches almost 100% precision drops, it can be seen in both of the architectures.

V. CONCLUSION AND FUTURE WORK

Recently, a large number of research has been done to examine the use of CNN, ANN and RNN as deep learning methods. According to that, works have been classified to the advancement, model design and training process. Multiple papers are reviewed during the process to evaluate individual methods based on their experimental results. Moreover, these individual type of works help to reach to the goal where the advantages and disadvantages of each methods of deep learning is clarified. Numerous researches on similarity, detection, retrieval and verification can be done on various objects studying these cases.

REFERENCES

[1] A. bar hillel, T. Hertz, N. Shental, and D. Weinshall, "Learning distance functions using equivalence relations," vol. 1, 07 2003, pp. 11–18.

[2] E. Xing, M. Jordan, S. J. Russell, and A. Ng, "Distance metric learning with application to clustering with side-information," *Advances in neural information processing systems*, vol. 15, 2002.

[3] K. Q. Weinberger, J. Blitzer, and L. Saul, "Distance metric learning for large margin nearest neighbor classification," *Advances in neural information processing systems*, vol. 18, 2005.

[4] A. Globerson and S. Roweis, "Metric learning by collapsing classes," *Advances in neural information processing systems*, vol. 18, 2005.

[5] L. Yang and R. Jin, "Distance metric learning: A comprehensive survey," *Michigan State Universiy*, vol. 2, no. 2, p. 4, 2006.

[6] G. Lanckriet, N. Cristianini, P. Bartlett, L. Ghaoui, and M. Jordan, "Learning the kernel matrix with semi-definite programming." vol. 5, 01 2002, pp. 323–330.

[7] A. Frome, Y. Singer, F. Sha, and J. Malik, "Learning globally-consistent local distance functions for shape-based image retrieval and classification," in *2007 IEEE 11th International Conference on Computer Vision*. IEEE, 2007, pp. 1–8.

[8] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, vol. 2. IEEE, 2006, pp. 1735–1742.

[9] N. A. Mahmon and N. Ya'acob, "A review on classification of satellite image using artificial neural network (ann)," in *2014 IEEE 5th Control and System Graduate Research Colloquium*, 2014, pp. 153–157.

[10] T. Guo, J. Dong, H. Li, and Y. Gao, "Simple convolutional neural network on image classification," in *2017 IEEE 2nd International Conference on Big Data Analysis (ICBDA)*, 2017, pp. 721–724.

[11] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.