Augmenting Nearest Neighbor-Based Algorithms With Siamese Neural Networks

Gene Lewis
Stanford University
glewis17@cs.stanford.edu

Wilbur Yang
Stanford University
wilbury@cs.stanford.edu

Abstract

While nearest neighbor-based algorithms perform well on low-dimensional datasets with well-defined semantic similarity metrics, they perform poorly on high-dimensional datasets like images. The main reason for this disparity is the lack of semantic meaning in traditional distance measures such as cosine or Jaccard similarity when they are applied to high dimensional data. By learning a distance metric for a specific dataset using Siamese layers in convolutional neural networks (CNNs), we can boost the performance of nearest neighbor algorithms by using the learned distance metric. Due to nearest neighbor-based algorithms' invariance to the number of classes, we suspected that a boost in the suitability of the similarity metric will vastly improve their performance, even on large datasets where vanilla CNN classifiers may not perform very well. We investigated the trade-offs of using a boosted K-Nearest Neighbor (KNN) classifier with learned distance metric as compared to a KNN classifier with vanilla L_2 distance metric on CIFAR-10. We found that the KNN classifier with learned distance metric outperforms the vanilla classifier and scales better with the choice of K.

1. Introduction

1.1. Convolutional Neural Classifiers

A breakthrough in image classification came with the introduction of Convolutional Neural Networks (CNNs)[6], where an image is passed into a nested series of functions and convolved with filters, producing a distribution over the output classes. However, these classifiers become more and more difficult to train as the number of potential classes increases relative to the data.

1.2. Nearest-Neighbor Classifiers

Nearest-neighbor approaches to classification are relatively agnostic to the number of predictive classes. This is essentially because all of the predictive power of neighbor-based algorithms is concentrated in the representative power of the dataset and the suitability of the distance metric used to calculate "closest neighbors" to a test point of interest. Standard distance metrics such as Euclidean distance usually perform poorly for high-dimensional data such as images, and in large data settings such as ImageNet[9]. It has been proposed that an adaptive or learned distance metric is more suitable for increasing nearest-neighbor performance for algorithms. We are therefore investigating whether using a neural network that learns a distance metric from data can be utilized to boost the performance of nearest-neighbors classifiers while circumventing the curse of dimensionality.

The input to our Siamese network is a pair of images. Our network then computes a measure of distance between them, and outputs a scalar representing this estimated distance. For our K-Nearest Neighbors classifier, the input is a stacked array of images and our output is an integer in the range of the number of classes representing our prediction of the class of the image.

2. Related Work

Our paper draws inspiration from Chopra *et al.* [2], which uses a Siamese network to learn a distance metric between images of faces in the AT&T and FERET datasets. In this paper, a contrastive metric allows for discriminative learning by mapping the input pair of images to a low-dimensional target space, in which a distance function is applied (a la energy-based models, or EBMs [7]) and a contrastive loss is computed. Hadsell *et al.* [4], its spiritual successor, describes slightly different distance and loss functions, which we employ in our experiments. Nair *et al*. builds on top of [2] by specifying the use of ReLU layers, in particular the Noisy ReLU (NReLU) activation as a beneficial option in a distance metric learning Siamese architecture.

Approaches taken to learn a distance metric vary widely. For instance, Weinberger *et al.* [11] takes a slightly different approach on the same overarching problem of metric learning and instead poses the problem as an application of semidefinite programming.

The state-of-the-art in metric learning is seen in Bell *et*

al. [1], where widely successful CNNs such as AlexNet and GoogLeNet were used in learning a distance metric of crowdsourced product images on a massive scale. In particular, evaluation of the metric involved testing whether or not the k-nearest neighbors of an in-situ cropped image contains the corresponding iconic (target) image. Furthermore, the similarities found in the k nearest neighbors found are qualitatively compelling and not always superficial.

3. Methods

3.1. Siamese Neural Networks

We implemented a Siamese neural network to learn a distance metric from inputs of paired images. The Siamese neural network is a horizontal concatenation of two identical CNNs such that each branch of the network sees only half of the input image pair. The final outputs are combined using the L_2 norm so that a L_2 distance of 0 denotes the closest classification and any deviation denotes a larger distance. This output distance is taken to be the "energy" corresponding to the input [4] [7].

The standard formula for this distance is as follows:

$$D_W(X_1, X_2) = ||G_W(X_1) - G_W(X_2)||_2$$

where G_W is the learned mapping that takes input images to their representation in the metric embedding, parametrized by the learned weight matrix W, D_W is the distance or energy, and X_1 and X_2 are the two input images.

To train this Siamese neural network, we learn on a contrastive loss function that in essence pulls together pairs of similar images and pushes apart pairs of dissimilar images. A dissimilar image within a set margin in the learned embedding contributes positively to the loss. As seen in [4], the contrastive term in the loss function is crucial for preventing convergence to the collapsed solution (that is, trivial weights) of the EBM. The specific loss function is reproduced below:

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

where W is the matrix of shared weights learned by the Siamese network, Y is the class label (either 0 - same class, or 1 - different classes), and D_W is defined above. The complete architecture of our network is shown in Figure 1.

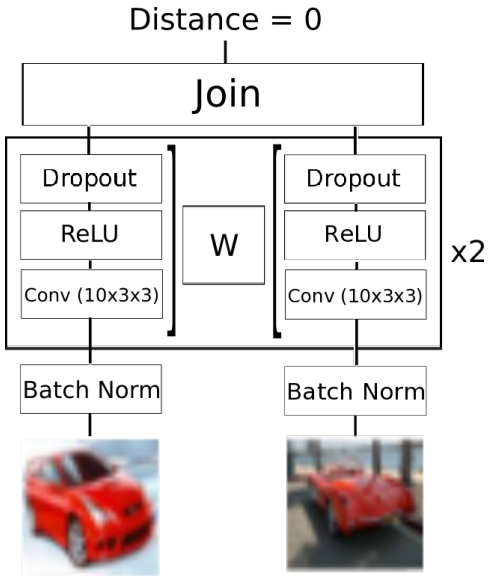

Figure 1. The architecture of our Siamese network. W represents the shared weight matrix between the identical halves of the Siamese architecture.

3.2. K-Nearest Neighbors

The K-Nearest Neighbors classifier trains by memorizing the training data points and the corresponding classifications. Given a new test image and some distance metric, the classifier calculates the distance between the test point and all the training points, taking the k closest and choosing the corresponding classification that appears most frequently.

This approach usually works well when the training data set can reliably be used to perform inference on the test set and when the distance metric used to calculate "nearest" images is suitable. One of the most straightforward distances is the L_2-norm (also known as Euclidean) distance:

$$d(\vec{x}, \vec{y}) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

However, we require a more semantically suitable distance metric when working with KNN on image datasets.

3.3. Siamese-Boosted K-Nearest Neighbors

As previously discussed, the core component of the K-Nearest Neighbors algorithm depends on the distance metric used to classify training points as "closer" or "further" from a given test point. Also as previously discussed, a Siamese neural network has the ability to estimate a learned distance between two input images. In this work, we study the effects of combining these techniques by using the forward pass of a trained Siamese neural network as our distance metric in K-Nearest Neighbors.

4. Dataset/Features

4.1. CIFAR-10

We used the CIFAR-10 dataset due to ease of access, dataset size constraints, and easily interpretable prediction results. The CIFAR-10 dataset utilizes 60,000 color images

of size 32x32 pixels across 10 classes with 6,000 images per class. These classes are a diverse range of animate and inanimate objects: airplanes, automobiles, birds, cats, deer, dogs, frogs, horses, ships, and trucks. This dataset is a standard across the literature and commonly used to ensure that learning algorithms with convolutional layers work in the first place [5].

4.2. Augmenting with Pairs

Training our Siamese network requires inputs of pairs of images. Our dataset was generated from the CIFAR-10 dataset. For each image $\mathcal{I}$ in the dataset, we created two examples by pairing that image with one image with the same label, $\mathcal{I}_G$, and one image with a different label, $\mathcal{I}_I$. Adopting the terminology of [2], the former is a "genuine" pair while the latter is an "impostor" pair. Thus, we have generate the pairs $(\mathcal{I}, \mathcal{I}_G)$ and $(\mathcal{I}, \mathcal{I}_I)$. Since CIFAR-10 by default has 50,000 training examples and 10,000 test examples, we generated 100,000 training examples and 2,000 test examples. Because each image is $3 \times 32 \times 32$ floats, the training set is 3.1×10^8 floats, or 9.8×10^9 bytes in total. We found that a dataset of this size is about the maximum we are able to fit in memory on a machine with 16 GB of RAM.

Because a batch normalization layer was part of our CNN architecture, there was no need to normalize the input images manually in a preprocessing step.

5. Experiments/Results/Discussions

5.1. Overfit Siamese Network

We first trained a large Siamese neural network with four Convolution layers followed by a fully connected layer. Each convolution layer is composed of stacks of 10 filters with spatial extent 3×3 apiece. All weights of all layers for all networks were initialized using Glorot initialization [3]. Additionally, each convolution layer is followed by a ReLU[8] activation layer.

The results of training this network for 30 epochs are shown in 3. We note that although the training loss decreases steadily over the period of training, the validation loss steadily increases. This behavior is often indicative of the model overfitting to the data; to investigate, we plotted a similarity matrix encoding the distances between examples from two similar classes (in this case, planes and automobiles), to examine the discriminative strength of the model.

The similarity matrix is shown in 4; dark patches represent images with large distance, and light patches represent images with small distance. If our Siamese network has been properly learning, we expect the bottom left and top right quadrants of the matrix to be light in color since these regions correspond to comparing two images taken from the same class. In contrast, we expect the top left and bottom right quadrants of the matrix to have a darker hue, since these regions correspond to comparing two images taken from different classes. However, note that we expect the difference to be only slightly pronounced, since the selected classes are visually quite similar.

When we examine 4, we note that we do see the tiled pattern we expect, but that the strong contrast suggests that the network has acquired a degree of overfit in discriminating between these classes.

5.2. Smaller Siamese Network

To combat the evidence of overfitting observed in the previous section, we take a combination of measures. First, we reduce the number of parameters in the model by cutting the number of convolutional layers from four to two. Next, we implement early stopping since over-training can lead to overfitting; through cross-validation, we chose our early stopping time to be after 10 epochs. Finally, we introduced a Dropout[10] layer after every activation, with Dropout probability of 0.1.

When we examine the training and validation loss for our smaller Siamese network in 5, we find that the training and validation losses now diverge much less and, instead, both converge to around 0.23. Similarly, when we examine the similarity matrix 6 for the small Siamese network, we find that the differences between dark and light regions are fuzzier and closer in color, indicating that the network can still make a similarity distinction but does not overplay the differences in the classes. Thus, we conclude that our smaller Siamese network is relatively free from overfit.

5.3. Vanilla K-Nearest Neighbors Classifier

For our baseline K-Nearest Neighbors classifier, we trained on 20,000 random images from CIFAR-10 taken uniformly from the classes. We then tested on 200 validation points, using L_2 distance to find the distance between each validation point and all 20,000 training points. We then cross-validated our choice of K for the optimal number of neighbors, show in 7.

Note that the performance of L_2-distance K-Nearest Neighbors generally decreases as K increases. This trend can be interpreted conceptually as expanding a hypersphere around a test point with radius defined by L_2-distance and encountering increasingly more training images that lie near the test image in L_2 space, but not in our semantic space, leading to misclassification error. This drop in performance as K increases suggests that L_2 distance is not an appropriate distance metric for the semantic space that the images live in.

This notion is strengthened by examining the confusion matrix shown in 8. Here, the axes of our matrix each consist of the ten output classes of CIFAR-10, and each cell corresponds to a count of test images. For each test image,

Figure 2. Example of a genuine and imposter pair

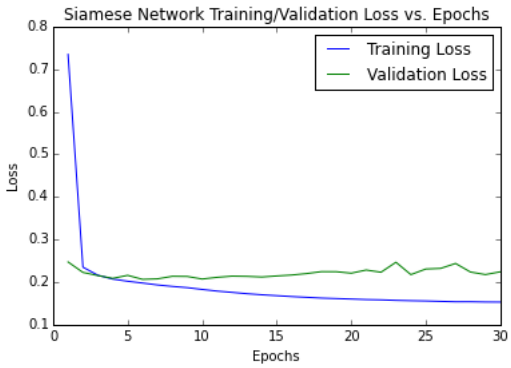

Figure 3. Training and Validation loss for a large Siamese network

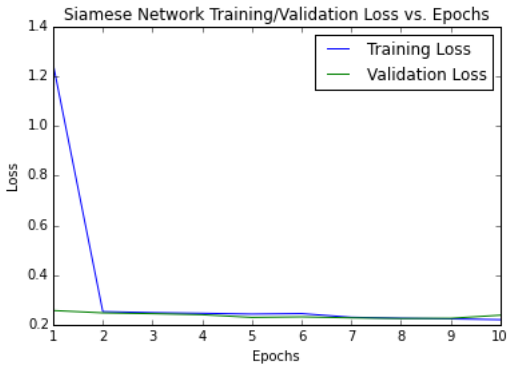

Figure 5. Training and Validation loss for a smaller Siamese network

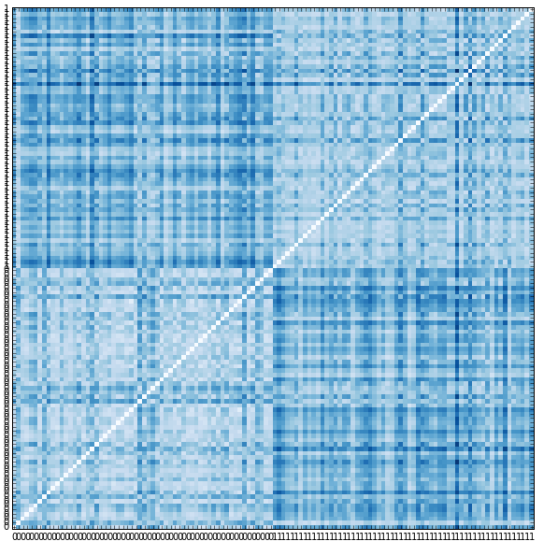

Figure 4. Similarity imatrix for large Siamese net with 60 examples taken from classes 0 (plane) and 1 (automobile). In this plot, lighter means more similar.

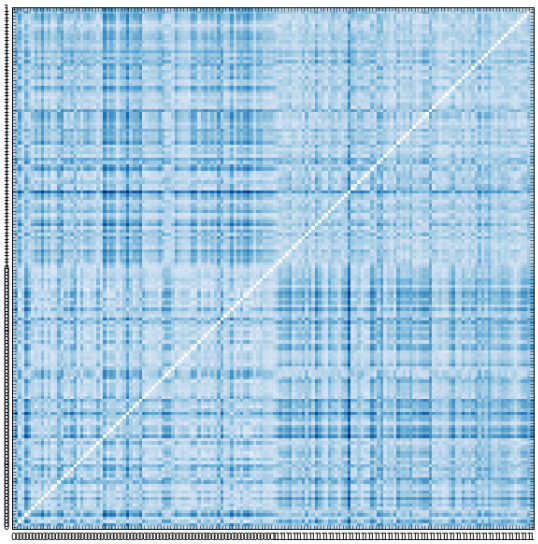

Figure 6. Similarity matrix for small Siamese net with 60 examples taken from classes 0 (plane) and 1 (automobile). In this plot, lighter means more similar.

we have its actual class label and the class label predicted by K-Nearest Neighbors; the cell that corresponds to these

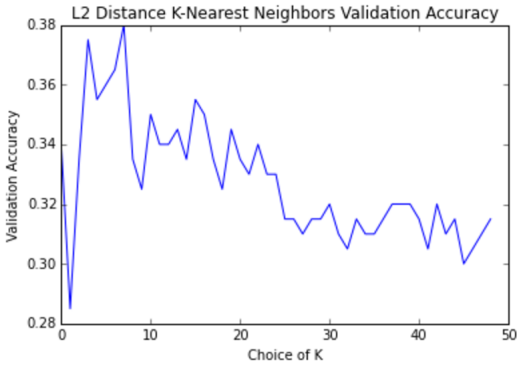

Figure 7. Accuracy of KNN with L_2 distance for different choices of K.

indices has its count incremented by one. Thus, our confusion matrix gives us a succinct visual representation of which classes are commonly misclassified as what.

Note that CIFAR-10 also has an interesting structure; classes that correspond to vehicles are given by the first two and final two indices, while the middle six indices correspond to various types of wildlife. Thus, we expect large number of errors in the regions of the matrix where each of these regions overlap with itself: the corners of the matrix, corresponding to vehicle confusion, and the center of the matrix, corresponding to wildlife confusion. Finally, we expect a good classifier to have a strong diagonal axis, corresponding to correct classifications.

When we examine 8, we find evidence that L_2 KNN is indeed a weak classifier for images. The region of confusion is very widely spread across the matrix, with a weak diagonal axis and little distinction between the expected areas of error compared to regions where we expect little to no error. This uniformity of error reinforces the evidence that L_2 is not an appropriate distance metric for the semantic image space, as there is little discernible pattern in the errors that arise.

5.4. Boosted K-Nearest Neighbors Classifier

Similarly to our baseline K-Nearest Neighbors classifier, we trained on 20,000 random images from CIFAR-10 taken uniformly from the classes. We then tested on 200 validation points, using our small trained Siamese network to find the distance between each validation point and all 20,000 training points. We then cross-validated our choice of K for the optimal number of neighbors, show in 10.

We note with satisfaction that, in contrast to the performance of L_2-distance K-Nearest Neighbors, the performance of Siamese-Network Boosted K-Nearest Neighbors generally increases as K increases. Returning to our thought experiment from before, this trend can be interpreted conceptually as expanding a hypersphere around a test point with radius defined by our learned metric distance

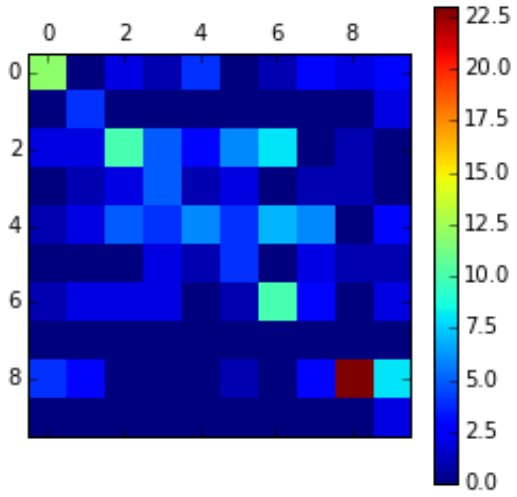

Figure 8. Confusion matrix across 10 classes for KNN with L_2 distance, K=8

Figure 9. Misclassification examples for KNN with the L_2 distance metric. Images on the left are test images, images on the right are the corresponding closest image.

and encountering increasingly more training images that lie in our semantic space, leading to stronger and more confident classifications as the size of our sphere (proportional to K) increases. This increase in performance as K increases suggests that, in contrast to our baseline, our Siamese network did indeed learn to model an appropriate distance metric for the semantic space of the images.

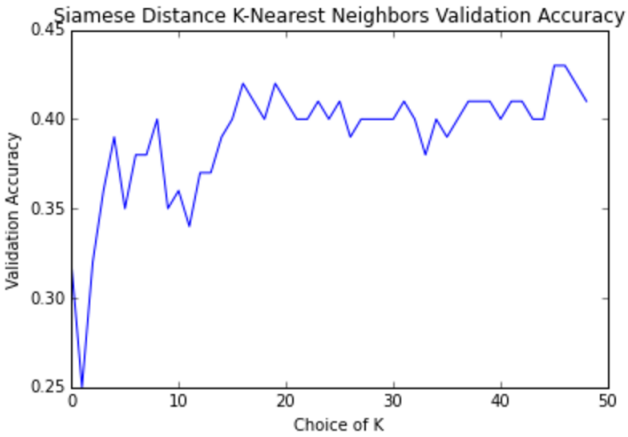

Figure 10. Accuracy of KNN with a learned distance metric for different choices of K.

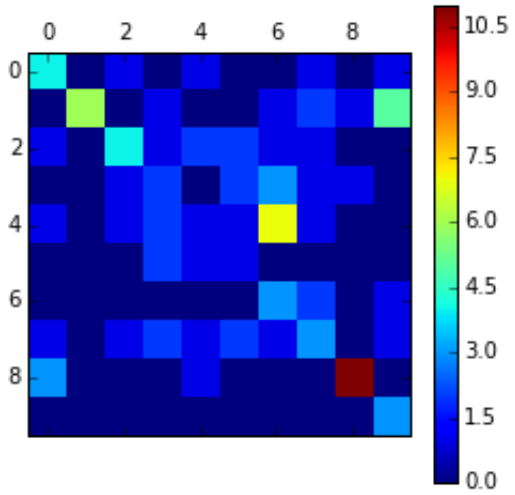

Figure 11. Confusion matrix across 10 classes for KNN with a learned distance metric, K=46.

When we examine 11, we find evidence that our Siamese-Boosted KNN is indeed a strong classifier for images. The region of confusion conforms much more closely with our expected pattern discussed in the previous section, with a relatively strong diagonal axis, a higher proportion of the error concentrated in the corners and centers, and a very low proportion of error in the center top, bottom, left, and right of the matrix. This stronger error pattern reinforces the evidence that the Siamese distance metric is an appropriate distance metric for the semantic image space, as the error pattern structure implies that the Siamese distance is capable of at least making the high-level distinction between vehicles and wildlife, which the baseline distance was unable to do.

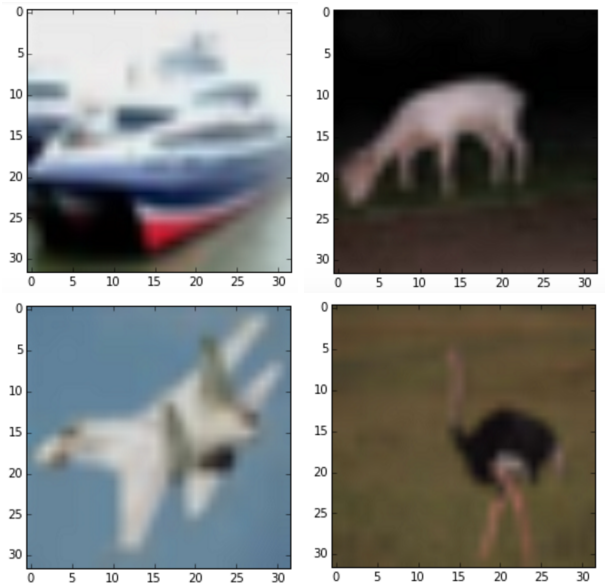

Figure 12. Misclassification examples for Siamese-Boosted KNN. Images on the left were test images, images on the right were the corresponding closest image.

5.5. Error Analysis

We now examine misclassification examples for our baseline and Siamese K-Nearest Neighbor classifiers. For our baseline, the figure given in 9 displays two misclassifications, where each row is a test image/train image pair such that the right training image is the predicted closest image to the left test image. We note that the errors seem to be fairly interpretable: both image pairs have very similar pixel colors in similar places, with the ship/dog pair both having a white background with a darker center image, and the deer/frog pair both having a green background with a brown center. We recall that L_2 distance takes the square root of the sum of the squares of the differences between two images; with that in mind, it makes sense that the raw pixel values between these images align closely. This analysis strongly reinforces the idea that L_2 is a poor semantic metric, operating on the technical level of pixels and, at best, shapes, instead of understanding the semantic contrastive differences between (most starkly) a ship and a dog.

The figure given in 12 also displays two misclassifications in a similar manner to the last. Here, we note the errors do not lend themselves so easily to a surface level interpretation; neither image pair has similarly colored pixels, nor do they have similarly shaped dominant objects in the same part of the frame. Thus, the nature of this error seems to run deeper than mere similar pixel value. Revisiting our argument for why the confusion matrix in 11 is valid evidence for the superiority of the Siamese-Boosted KNN approach over the baseline, we note that CIFAR-10 can be broken down into two broad categories consisting

of vehicles and wildlife, which gave rise to a particular expected error pattern in the corresponding confusion matrix of a reasonable classifier. Coming back to our misclassification examples, we note that each of the test examples is some kind of vehicle, and has been misclassified as some kind of animal; thus, we see that the nature of this error is more closely related to a matter of semantics. This type of error is immensely interesting, and further lends credence to the idea that our Siamese network has learned a distance that conforms to the data manifold.

6. Conclusion/Future Work

In this work, we have thoroughly exhibited the ways in which a Siamese network can be used to boost the power of the ubiquitous K-Nearest Neighbors classifier. We have shown that standard distance metrics fail to account for the special manifold of the data, as evidenced by surface level misclassification errors, random confusion matrix error patterns, and decreasing accuracy as K increases. We have also shown that , in contrast, using a distance metric learned using a Siamese network reverses all of these indicators by having deeper misclassification errors, a more predictable confusion matrix error pattern, and an increasing accuracy as K increases; it thus has strong evidence for learning an appropriate distance metric that works well with K-Nearest Neighbors.

Unfortunately, due to computational constraints, we were unable to explore how our approaches work on much larger datasets. Given extra time and resources, we believe that our positive results here indicate that our approach does indeed help circumvent the curse of dimensionality by learning to work with it, and so could be competitive with modern techniques on massive data sets with a large target prediction space. We would also be interested in learning a more sophisticated distance model, or investigating parallelism techniques to make model-distance based approaches competitive in speed.

References

[1] S. Bell and K. Bala. Learning visual similarity for product design with convolutional neural networks. *ACM Transactions on Graphics (TOG)*, 34(4):98, 2015.

[2] S. Chopra, R. Hadsell, and Y. 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.

[3] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *International conference on artificial intelligence and statistics*, pages 249–256, 2010.

[4] R. Hadsell, S. Chopra, and Y. LeCun. Dimensionality reduction by learning an invariant mapping. In *Computer vision and pattern recognition, 2006 IEEE computer society conference on*, volume 2, pages 1735–1742. IEEE, 2006.

[5] A. Krizhevsky. Learning multiple layers of features from tiny images, 2009.

[6] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[7] Y. LeCun, S. Chopra, R. Hadsell, M. Ranzato, and F. Huang. A tutorial on energy-based learning. *Predicting structured data*, 1:0, 2006.

[8] V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 807–814, 2010.

[9] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[10] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 2014.

[11] K. Q. Weinberger and L. K. Saul. Distance metric learning for large margin nearest neighbor classification. *The Journal of Machine Learning Research*, 10:207–244, 2009.