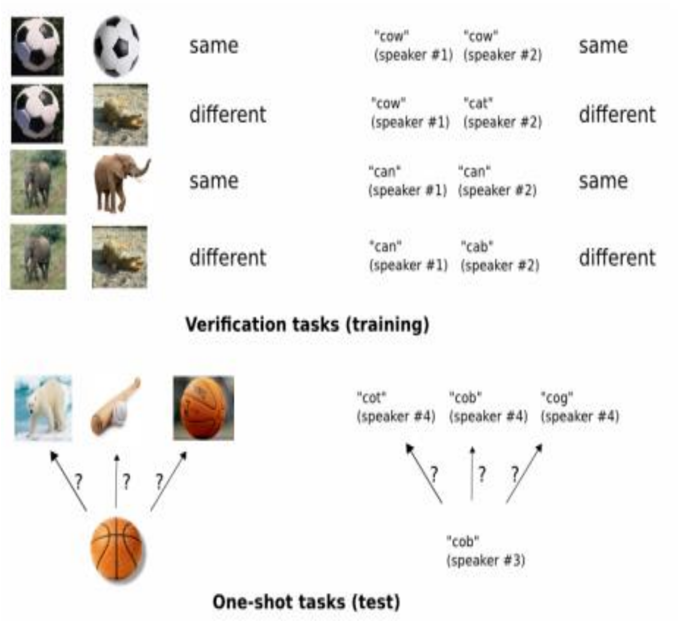
Siamese Neural Networks for One-shot Image Recognition

Authors: Gregory Koch, Richard Zemel, Ruslan Salakhutdinov

Abstract

The process of learning good features for machine learning applications can be very computationally expensive and may prove difficult in cases where little data is available. A prototypical example of this is *the one-shot learning* setting, in which we must correctly make predictions given only a single example of each new class.In this paper, authors explore a method for learning *siamese neural networks* which employ a unique structure to naturally rank similarity between inputs.



Machine learning has been successfully used to achieve state-ofthe-art performance in avariety of applications such as web search, spam detection, caption generation, and speech and image recognition. However, these algorithms often break down when forced to make predictions about data for which little supervised information is available.One particularly interesting task is classification under the restriction that we may only observe a single example of each possible class before making a prediction about a test instance. This is called one-shot learning and it is the primary focus of our model presented in this work.

Approach:

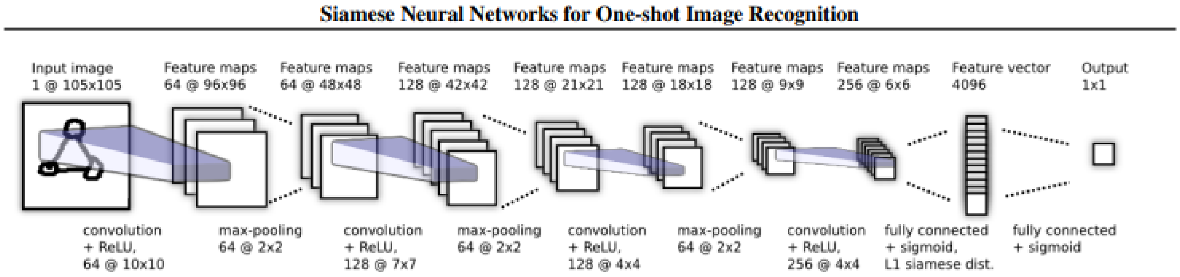
In this experiment, the authors restrict their attention to character recognition, although the basic approach can be replicated for almost any modality. For this domain, they employ large siamese convolutional neural networks which a) are capable of learning generic image features useful for making predictions about unknown class distributions even when very few examples from these new distributions are available; b) are easily trained using standard optimization techniques on pairs sampled from the source data; and c) provide a competitive approach that does not rely upon domain-specific knowledge by instead exploiting deep learning techniques.

To develop a model for one-shot image classification, the aim is to first learn a neural network that can discriminate between the class-identity of image pairs, which is the standard verification task for image recognition. The verification model learns to identify input pairs according to the probability that they belong to the same class or different classes. This model can then be used to evaluate new images, exactly one per novel class, in a pairwise manner against the test image. The pairing with the highest score according to the verification network is then awarded the highest probability for the one-shot task.

Deep Siamese Network for Image Verification:

A siamese neural network consists of twin network which accept distinct inputs but are joined by an energy function at the top. Same weight and parameters are used in both the networks.

The model consists of a sequence of convolutional layers, each of which uses a single channel with filters of varying size and a fixed stride of 1. The number of convolutional filters is specified as a multiple of 16 to optimize performance. The network applies a *ReLU activation function* to the output feature maps, optionally followed by maxpooling with a filter size and stride of 2.



Thus the kth filter map in each layer takes the following form:

a (k) 1,m = max-pool(max(0,W(k) l−1,l ? h1,(l−1) + bl), 2)

a (k) 2,m = max-pool(max(0,W(k) l−1,l ? h2,(l−1) + bl), 2)

where Wl−1,l is the 3-dimensional tensor representing the feature maps for layer l and we have taken * to be the valid convolutional operation corresponding to returning of only those output units which were the results of complete overlap between each convulutional filter and the input feature maps.

The units in the final convolutional layer are flattened into a single vector. This convolutional layer is followed by a fully-connected layer, and then one more layer computing the induced distance metric between each siamese twin, which is given to a single sigmoidal output unit.

Loss Function:

We assume y(x (i) 1 , x (i) 2) = 1 whenever x1 and x2 are from the same character class and y(x (i) 1 , x (i) 2) = 0 otherwise. We impose a regularized cross-entropy objective on our binary classifier of the following form:

L(x $^{(i)}$ 1 , x $^{(i)}$ 2) = y(x $^{(i)}$ 1 , x $^{(i)}$ 2) log p(x $^{(i)}$ 1 , x $^{(i)}$ 2)+ (1 − y(x $^{(i)}$ 1 , x $^{(i)}$ 2)) log (1 − p(x $^{(i)}$ 1 , x $^{(i)}$ 2)) + λ T |w|^{2}

Optimization:

We fix a minibatch size of 128 with learning rate ηj , momentum µj , and L2 regularization weights λj defined layer-wise, so that our update rule at epoch T is as follows:

w (T) kj (x (i) 1 , x (i) 2) = w (T) kj + Δw (T) kj (x (i) 1 , x (i) 2) + 2λj |wkj | Δw (T) kj (x (i) 1 , x (i) 2) = −ηj∇w (T) kj + µjΔw (T −1) kj , where ∇wkj is the partial derivative with respect to the weight between the jth neuron in some layer and the kth neuron in the successive layer.

We initialized all network weights in the convolutional layers from a normal distribution with zero-mean and a standard deviation of 10−2 . Biases were also initialized from a normal distribution, but with mean 0.5 and standard deviation 10−2. learning rates were decayed uniformly across the network by 1 percent per epoch.

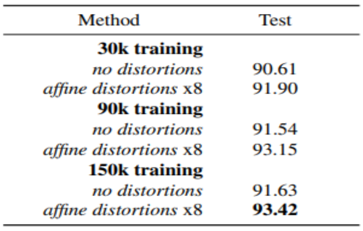
Hyperparameter optimization:- For learning schedule and regularization hyperparameters, we set the layerwise learning rate ηj ∈ [10−4 , 10−1], layer-wise momentum µj ∈ [0, 1], and layer-wise L2 regularization penalty λj ∈ [0, 0.1]. For network hyperparameters, we let the size of convolutional filters vary from 3x3 to 20x20, while the number of convolutional filters in each layer varied from 16 to 256 using multiples of 16. Fully-connected layers ranged from 128 to 4096 units, also in multiples of 16.

Experiments:- The Omniglot data set was collected by Brenden Lake and his collaborators at MIT via Amazon’s Mechanical Turk to produce a standard benchmark for learning from few examples in the handwritten character recognition domain. Omniglot contains examples from 50 alphabets ranging from well-established international languages like Latin and Korean to lesser known local dialects. The number of letters in each alphabet varies considerably from about 15 to upwards of 40 characters. All characters across these alphabets are produced a single time by each of 20 drawers Lake split the data into a 40 alphabet background set and a 10 alphabet evaluation set.

Verification:- To train our verification network, we put together three different data set sizes with 30,000, 90,000, and 150,000 training examples by sampling random same and different pairs. We set aside sixty percent of the total data for training: 30 alphabets out of 50 and 12 drawers out of 20.

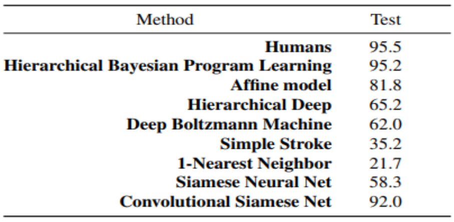
To monitor performance during training, we used two strategies. First, we created a validation set for verification with 10,000 example pairs taken from 10 alphabets and 4 additional drawers.

Table 1. Accuracy on Omniglot verification task (siamese conv lutional neural net)



Method	Test
30k training	
no distortions	90.61
affine distortions x8	91.90
90k training	
no distortions	91.54
affine distortions x8	93.15
150k training	
no distortions	91.63
affine distortions x8	**93.42**

Table 2. Comparing best one-shot accuracy from each type of network against baselines.



Method	Test
Humans	95.5
Hierarchical Bayesian Program Learning	95.2
Affine model	81.8
Hierarchical Deep	65.2
Deep Boltzmann Machine	62.0
Simple Stroke	35.2
1-Nearest Neighbor	21.7
Siamese Neural Net	58.3
Convolutional Siamese Net	92.0

One-shot Learning:- Once we have optimized a siamese network to master the verification task, we are ready to demonstrate the discriminative potential of our learned features at one-shot learning. Suppose we are given a test image x, some column vector which we wish to classify into one of C categories. We are also given some other images {xc} C c=1, a set of col umn vectors representing examples of each of those C categories. We can now query the network using x, xc as our input for a range of c = 1, . . . , C. 2 Then predict the class corresponding to the maximum similarity. C ∗ = argmaxcp (c).

To empirically evaluate one-shot learning performance, Lake developed a 20-way within-alphabet classification task in which an alphabet is first chosen from among those reserved for the evaluation set, along with twenty characters taken uniformly at random. Two of the twenty drawers are also selected from among the pool of evaluation drawers. These two drawers then produce a sample of the twenty characters. Each one of the characters produced by the first drawer are denoted as test images and individually compared against all twenty characters from the second drawer, with the goal of predicting the class corresponding to the test image from among all of the second drawer's characters. This process is repeated twice for all alphabets, so that there are 40 one-shot learning trials for each of the ten evaluation alphabets. This constitutes a total of 400 one-shot learning trials, from which the classification accuracy is calculated. The one-shot results are given in Table 2.

MNIST One-shot Trial:- The Omniglot data set contains a small handful of samples for every possible class of letter; for this reason, the original authors refer to it as a sort of "MNIST transpose", where the number of classes far exceeds the number of training instances. We thought it would be interesting to monitor how well a model trained on Omniglot can generalize to MNIST, where we treat the 10 digits in MNIST as an alphabet and then evaluate a 10-way oneshot classification task. We followed a similar procedure to Omniglot, generating 400 one-shot trials on the MNIST test set, but excluding any fine tuning on the training set. All 28x28 images were upsampled to 35x35, then given to a reduced version of our model trained on 35x35 images from Omniglot which were downsampled by a factor of 3. We also evaluated the nearest-neighbor baseline on this task. Table 3 shows the results from this experiment.

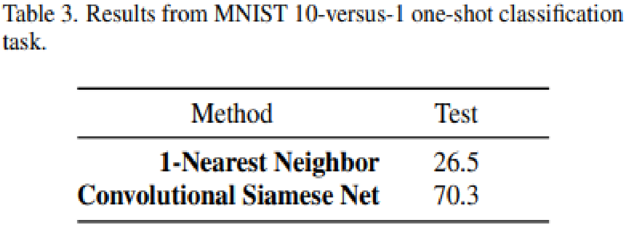


Table 3. Results from MNIST 10-versus-1 one-shot classification task.

Method	Test
1-Nearest Neighbor	26.5
Convolutional Siamese Net	70.3

The nearest neighbor baseline provides similar performance to Omniglot, while the performance of the convolutional network drops by a more significant amount. However, we are still able to achieve reasonable generalization from the features learned on Ominglot without training at all on MNIST.

Conclusions:- presented a strategy for performing one-shot classification by first learning deep convolutional siamese neural networks for verification. We outlined new results comparing the performance of our networks to an existing state-of-the-art classifier developed for the Omniglot data set. Our networks outperform all available baselines by a significant margin and come close to the best numbers achieved by the previous authors. We have argued that the strong performance of these networks on this task indicate not only that human-level accuracy is possible with our metric learning approach, but that this approach should extend to one-shot learning tasks in other domains, especially for image classification.

This algorithm can be extended to exploit the data about individual stroke trajectories to produce final computed distortions.