Recent Developments in One Shot Learning
Group no. 5

1. Akshat Jindal - 150075 - akshatj@iitk.ac.in
2. Mrinaal Dogra - 150425 - mrinaald@iitk.ac.in
3. Raktim Mitra - 150562 - raktim@iitk.ac.in
4. Rohit Bose - 150596 - rohitkb@iitk.ac.in
5. Shivam Utreja - 150682 - sutreja@iitk.ac.in

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, we firstly explore *Siamese Networks*, a unique twin-network setting which is capable of checking how *similar* any two inputs are. The observed shortcomings of the model inspire us to explore a more robust model, *Matching Networks*, a neural network which uses recent advances in attention and memory that enable rapid learning. We then extend the models to the MNIST in addition to the standard Omniglot dataset they are trained on to check their robustness and achieve promising results.

1 Problem Motivation

With the advent of deep learning approaches to tasks, we have begun achieving human-level accuracy on many previously poorly explored tasks. Conventional wisdom says that deep neural networks are really good at learning from high dimensional data like images or spoken language, but only when they have huge amounts of labelled examples to train on. Data augmentation and regularization techniques alleviate overfitting in low data regimes, but do not solve it. Furthermore, learning is still slow and based on large datasets, requiring many weight updates using stochastic gradient descent. Humans on the other hand, are capable of one-shot learning i.e. learn new concepts with very little supervision – e.g. a child can generalize the concept of "giraffe" from a single picture in a book.
This ability to rapidly learn from very little data seems like it's obviously desirable for machine learning systems to have because collecting and labelling data is expensive and thus motivates us to explore one-shot classification in detail.
Overall, research into one-shot learning algorithms is fairly immature and has received limited attention by the machine learning community.This motivates us to explore the domain of one-shot learning, exploring various approaches to one-shot classification and analyzing the ingenuity as well as shortcomings of these approaches.

2 Problem Statement

Before we try to solve any problem, we should first precisely state what the problem actually is, so here is the problem of one-shot classification expressed symbolically:

Our model is given a small labelled set $\mathbb{S}$, which has $\mathbf{N}$ examples, each vectors of the same dimension with a distinct label $\mathbf{y}$.

$$\mathbb{S} = \{(\boldsymbol{x_1}, y_1), ..., (\boldsymbol{x_n}, y_n)\}$$

It is also given $\boldsymbol{x'}$, the test example it has to classify. Since exactly one example in the support set has the right class, the aim is to correctly predict which $\hat{y} \in \mathbb{S}$ is the same as $\boldsymbol{x'}$'s label, y .

3 Previous Methods

- **Variational Bayesian Framework**[1][2]: The work in one-shot learning originated in the early 2000s by *Li Fei-Fei et al.* The authors developed a variational Bayesian framework, and used the premise that previously learned classes could be leveraged to help forecast future ones when very few examples are available from a given class.
- **Bayesian Network approach**[3]: *Mass et al.* proposed an approach using Bayesian networks where the network learns a hyperparameter for each distribution in the network, and which specifies whether it is a non-deterministic or near-deterministic one. The authors used this approach to one-shot learning problems based on a real-world database of immigration records, and showed that it outperformed the standard Bayesian network approaches for one shot-learning problems.
- **Hierarchical Bayesian Program Learning(HBPL)**[4]: *Lake et al.* approahed the problem of one-shot learning from the cognitive science perspective. The authors presented a Hierarchical Bayesian model based on compositionality and causality to address one-shot learning for character recognition.

3.1 Limitations of these Methods

- In some methods, like the HBPL[4] approach, the authors used a lot of metadata to learn their models. Such metadata includes stroke data, sub-strokes data, etc. These models learn about the process(strokes) through which a character is being generated. However such features for data might not be available while testing for a new test image.

4 Novel Developments

Most work in the field of one-shot learning addresses the problem in a highly domain specific way. The methods require highly domain specific knowledge of what features and inference procedures to use. Hence, these methods end up being fragile in nature; breaking when applied to a one-shot learning problem in a different setting, or even for a different dataset in a similar setting.
The works of *Koch et al.*[5] and *Vinyals et al.*[6] aim at developing robust, general purpose one-shot learning techniques which can be applied to a fairly wide range of domains, without much domain knowledge.

4.1 Using Siamese Neural Networks[5]

The Siamese neural network architecture is nothing but a pair of identical neural networks with their corresponding weights identical. This network takes a pair of distinc inputs, each input fed to one of the twins. The output (highest level feature vector) of each twin is combined by a function which computes the level of similarity between these vectors (normalized from $(0, 1)$). While training, pairs of inputs drawn from the same class are given a similarity score of 1, and those drawn from different classes are given a similarity score of 0. Traditionally, these networks were used to solve image tasks like signature verification.

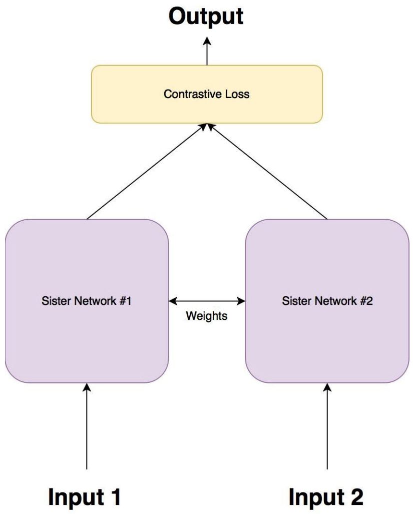

Figure 1: General siamese network architecture.

4.1.1 Approach

The goal of this approach is to learn fairly generic image features in a supervised manner, which can be reused in a one-shot setting without any retraining. The model goes about learning these generic features by training a deep Siamese convolutional neural network on pairs of images labelled "same" or "different", depending on whether they are drawn from the same class or different classes.
The subset of classes used during training is kept completely separate from those used for testing. In effect, every class that the network encounters during testing, it sees for the first time. During testing, the network compares the test image with exactly one image from each new class. It then gives a similarity score to each such pair. It then predicts the class of the test image same as that of the image with which it had the highest similarity score (above some threshold).

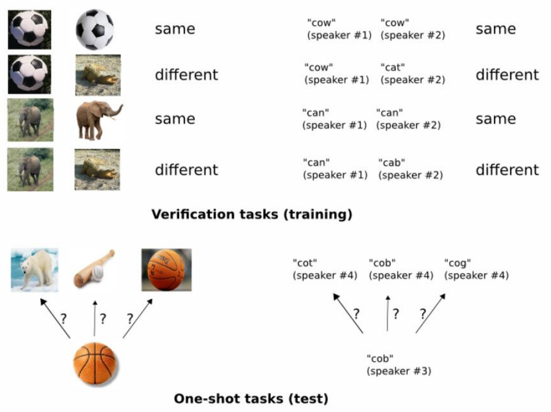

Figure 2: Training and testing the Siamese network for one-shot learning.

Note that all the steps of the above description are in line with the one-shot learning scenario. The parameters of the model are fixed using standard learning and optimization techniques. No domain specific knowledge is used.
The results in the paper mainly revolve around image (character) data. However, it can be reproduced for other domains, applying the same basic idea with minor domain-specific tweaking to the architecture.

4.1.2 Architecture details

The Siamese network model architecture which we used is the same as the one described in the *Koch et al.*[5]. In the paper a convolutional neural network was used. 3 Blocks of Conv-RELU-Max Pooling are used followed by a Conv-RELU connected to a fully-connected layer with a sigmoid function. This layer produces the feature vectors that will be fused by the L1 weighed distance layer. The output is fed to a final layer that outputs a value between 1 and 0 (same class or different class). To assess the best architecture, they had used a Bayesian hyper-parameter tuning. The best architecture is depicted in Figure 3

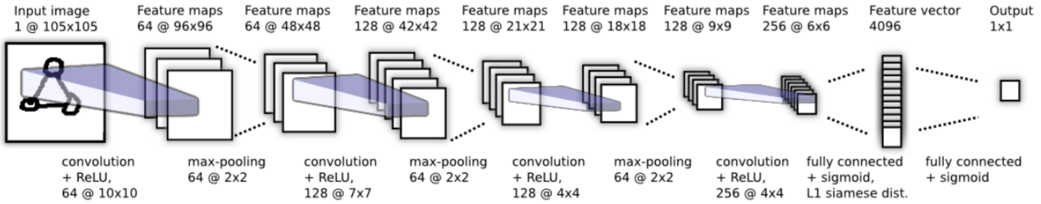

Figure 3: Architecture achieved after tuning in [5]. Siamese twin not depicted.

4.2 Matching Networks

4.2.1 Approach

Augmenting neural networks with external memory has been focus to a number of recent models e.g. memory networks[7] and pointer networks[8]. These models feature a differentiable attention mechanism which model P(A | B) where A and/or B are two sets or sequences. The Matching Networks model [6] casts this setting to a One Shot Learning setting using the set to set model of the above framework. The main novelty of this model lies in reinterpreting a well studied framework (neural networks with external memories) to do one-shot learning.

4.2.2 Model Description

The model maps a support set of k examples of image-label pairs $S = \{(x_i, y_i)\}_{i=1}^{k}$ to a classifier $c_S(\hat{x})$. The mapping is defined as $P(\hat{y}|\hat{x}, S)$, parameterized by a neural network. When a new support set $\hat{S}$ comes, the parametric neural network defined by P is used to make predictions. In simple terms the model looks like this:

$$\hat{y} = \sum_{i=1}^{k} a(\hat{x}, x_i) y_i$$

Here, x_i, y_i are labels from the support set S, and it is just a linear combination of y_i upon a kernel on X x X. a acts as an attention mechanism and the y_i act as memories bound to the corresponding x_i. One key point is unlike other previous attention mechanisms this a is non -parametric i.e. as the support set S grows so does the memory used. This raw usage of support set makes the C_S very flexible and adaptive to new support sets.

Attention Mechanism

Let's look at the attention kernel a in a bit more depth. A common function that is related to attention models and kernels is softmax over cosine distances. The matching network uses a very similar one:

$$a(\hat{x}, x_i) = e^{cos_d(f(\hat{x}), g(\hat{(}x_i)))} / \sum_{j=1}^{k} e^{cos_d(f(\hat{x}), g(\hat{(}x_j)))}$$

Here f and g are functions modelled by neural network and in various situations they can be different (even f = g). These are parametrised variously for different deep convolutional neural networks.

Full Context Embeddings

The most unusual aspect of matching networks is reinterpretation of memory augmentation to do one shot learning. The functions f and g are the neural network functions that embed each x_i from support set. The matching network model proposes embedding the elements of the set through a function which takes as input the full set S in addition to x_i, i.e. g becomes $g(x_i, S)$, calling it full context embedding. This becomes useful if some x_i is very close to some x_j The model also proposes use of bidirectional LSTM to encode x_i in the context of super set S.

4.3 Siamese network with LSTM

We tried one shot learning with Siamese networks and matching networks. Matching networks performed much better than vanilla Siamese. Matching networks uses bidirectional LSTM for context encoding. LSTM itself also does well in classification as we found out by running it on MNIST. So, we hypothesized that the Siamese Network results might improve if we augment the siamese pairs with LSTMs, it might perform better. The approach taken in this method was to fit a Siamese network composed of two LSTM (Long Short Term Memory) networks on each side and then compare their outputs. We passed pixels of images (from Omniglot dataset) column by column to the twin networks. This was an attempt to retain the spatial information of an image.

5 Data

5.1 Omniglot Dataset

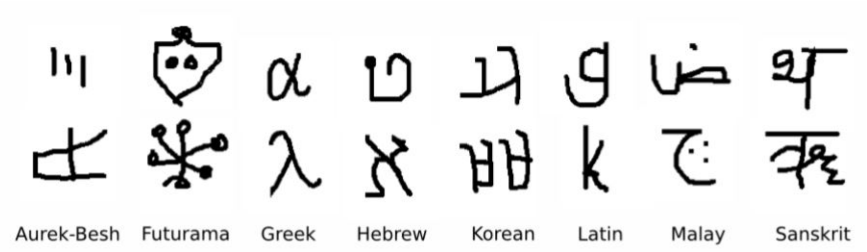

Figure 4: Instances of alphabets from various languages in the Omniglot dataset.

The omniglot dataset is a set of hand-drawn characters from the alphabets of various languages. It is meant to be a benchmark for learning from limited data, in the hand-written character recognition domain. It consists of all characters from alphabets of 50 different languages, ranging from famous international languages like Greek, to even a few fictitious languages like Klingon.

The alphabets have already been split into a 40 alphabet *background* set and a 10 alphabet *evaluation* set. The background set is used for training, validating and testing the model during the hyper-parameter tuning phase while building the model. The evaluation set is only used to measure the performance of the final, tuned model on the one-shot classification task.

Each of the characters from all the alphabets has been drawn by 20 different artists, single time each. Therefore, in this dataset, the number of classes (≈ 1500) far outnumber the number of samples per class (20). This is why it is also referred to as the "MNIST transpose" dataset.

5.2 MNIST Dataset

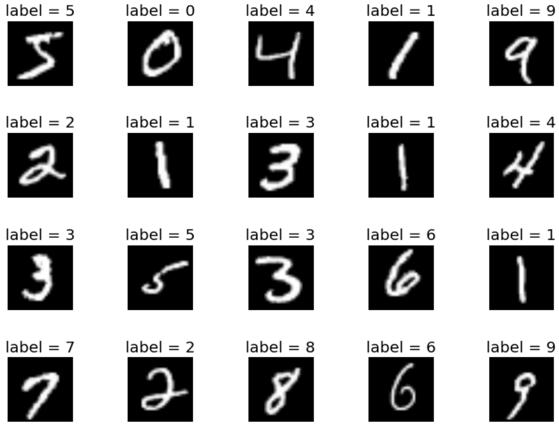

Figure 5: Examples of handwritten digits and their labels from the MNIST dataset

The MNIST database contains labelled, handwritten digits (0 through 9) from approximately 250 writers. The training set has $60,000$ examples while the test set has $10,000$. The set of writers for the training set and the test set is disjoint. The original black and white images have been normalised to fit 20×20 pixel box, while preserving the aspect ratio. The images were centered in a 28x28 image by computing the center of mass of the pixels, and translating the image so as to position this point at the center of the 28x28 field.

5.3 Data Preprocessing and Augmentation

We have augmented the training images before feeding them to the networks with affine transformations. These transformations were introduced to increase the amount of dataset as well as to reduce the probability of over-fitting. We used the following transformations:

- **Rotation**: The images are rotated in the range of $[-15°, 15°]$.
- **Shear**: The images are sheared with a factor of $[-17.2°, 17.2°]$.
- **Zoom**: The images are scaled(zoomed) with a factor in the range of of $[0.8, 2]$.
- **Shift**: The images are shifted in the xy-frame with a range of $[-5, 5]$ pixels.

6 Implementation details

6.1 Siamese Network

- **Optimizers**: In contrast to the *Koch et al.*[5] where the authors used a modified version of the SGD algorithm, we used the Adam optmizer in our final implementation. Apart from Adam, we also tried the standard SGD(Stochastic Gradient Descent) and AdaGrad optimizers in our model.
- **Weight and Bias Initialization of layers**: The weights for all the layers in the network are initialized from a normal distribution with mean 0 and a standard deviation of 10^{-2}. The biases for layers are initialized from a normal distribution with a mean of 0.5 and a standard deviation of 10^{-2}
- **Layer-wise kernel regularization**: We have also kept different kernel l2 regularization parameter for each convolutional layer and the dense layer.

6.2 Matching Network

Matching Network is the most state of the art model of one shot learning. Due to its implementational complexity, we did not implement it from scratch. For our convenience we used an available implementation (`https://github.com/AntreasAntoniou/MatchingNetworks`) of matching network for testing and tweaking.

6.3 Siamese Network with LSTM

The LSTM takes inputs as columns of the image matrix, and outputs a 64 dimensional vector that is fed in to a dense layer which outputs a 128 dimensional vector. The l1 norm of this vector is used for the distance to be fed into the sigmoid layer. As mentioned earlier, we tried to combine two LSTMs like in a Siamese network.

- **Optimizers**: We used the Adam optimizer in our implementation.
- **Weight and Bias Initialization of layers**: The weights for all the layers in the network are initialized from a normal distribution with mean 0 and a standard deviation of 10^{-2}. The biases for layers are initialized from a normal distribution with a mean of 0.5 and a standard deviation of 10^{-2}
- **Layer-wise kernel regularization**: We have used l2 regularization in kernels.

7 Results

All the following results represents accuracy on 20-way one shot classification task.

	Accuracy %
Siamese Net without image augmenatation on Omniglot	33.75
Siamese Net with Default SGD optimiser on Omniglot	66.53
Siamese Net with Adagrad optimiser Omniglot	63.28
Siamese Net with Adam optimiser Omniglot	68.75

Table 1: Experimental Results with Siamese Network

As we implemented the Siamese network a little bit differently from the one discussed in *Koch et al.*[5], we were not able to reach the accuracies as were presented in the paper. Nonetheless, the results are promising enough to improve on tuning the model further.

	Accuracy %
Matching Network without full context embedding on Omniglot	81.25
Matching Network with full context embedding on Omniglot	85.63
Matching Network without full context embedding on MNIST	84.38

Table 2: Experimental Results with Matching Network

Matching networks is the latest work in this field. It uses the whole support set to recognize new test set of images. This idea of reinterpretation of memory augmentation to do one shot learning is clearly proving to be far ahead of other models in terms of accuracy.

	Accuracy %
LSTM standard classification accuracy	97
LSTM on one-shot learning [9]	12.7
Siamese Network with augmented LSTM for one shot	10.32

Table 3: Experimental Results with Siamese Network augmented LSTM

As we can see, our augmentation of siamese network with LSTM led to a very poor accuracy of around 10%. Surprisingly, though a normal LSTM worked pretty well for classifying elements of the MNIST dataset (is an entirely single, non-Siamese fashion) with a staggeringly high accuracy of 97%.

8 Possibilities for Future Work

- We would like to do more cross dataset testing without retraining, starting with the MNIST dataset. The model developed on the Omniglot dataset can be reused on the MNIST dataset by rescaling the MNIST images.
- We would like to have our own implementation of Matching networks which we were unable to accomplish in this scope.
- Our augmentation of Siamese network with LSTM is performing poorly and we have to figure out the reasons for the same, and if possible, remedy them.

9 Acknowledgements

One-shot learning is a very challenging task which we have realised first hand. It was enjoyable to read various research papers and brain-storming. Although our own novel trials did not produce satisfactory results, it has been a steep learning experience. We thank Prof. Arnab Bhattacharya for giving us this opportunity.

References

[1] L. Fe-Fei, Fergus, and Perona, "A bayesian approach to unsupervised one-shot learning of object categories," in *Proceedings Ninth IEEE International Conference on Computer Vision*, pp. 1134–1141 vol.2, Oct 2003.

[2] L. Fei-Fei, R. Fergus, and P. Perona, "One-shot learning of object categories," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, pp. 594–611, Apr. 2006.

[3] A. I. R. Maas and C. Kemp, "One-shot learning with bayesian networks," 2009.

[4] B. M. Lake, R. R. Salakhutdinov, and J. Tenenbaum, "One-shot learning by inverting a compositional causal process," in *Advances in Neural Information Processing Systems 26* (C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, eds.), pp. 2526–2534, Curran Associates, Inc., 2013.

[5] G. Koch, T. EDU, R. Zemel, and R. Salakhutdinov, "Siamese neural networks for one-shot image recognition,"

[6] O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra, *et al.*, "Matching networks for one shot learning," in *Advances in Neural Information Processing Systems*, pp. 3630–3638, 2016.

[7] A. Miller, A. Fisch, J. Dodge, A.-H. Karimi, A. Bordes, and J. Weston, "Key-value memory networks for directly reading documents," *arXiv preprint arXiv:1606.03126*, 2016.

[8] O. Vinyals, M. Fortunato, and N. Jaitly, "Pointer networks," in *Advances in Neural Information Processing Systems*, pp. 2692–2700, 2015.

[9] A. Santoro, S. Bartunov, M. Botvinick, D. Wierstra, and T. Lillicrap, "One-shot learning with memory-augmented neural networks," *arXiv preprint arXiv:1605.06065*, 2016.