A Relational Model for One-Shot Classification

Arturs Polis and Alexander Ilin

Aalto University, Espoo, Finland

Abstract. We show that a deep learning model with built-in relational inductive bias can bring benefits to sample-efficient learning, without relying on extensive data augmentation. The proposed one-shot classification model performs relational matching of a pair of inputs in the form of local and pairwise attention. Our approach solves *perfectly* the one-shot image classification Omniglot challenge. Our model exceeds human level accuracy, as well as the previous state of the art, with no data augmentation.

1 Introduction

Humans can often learn to distinguish previously unseen visual objects after being shown only a few example images of a novel object. Deep neural networks, on the other hand, typically require thousands of training images to generalize to new examples of classes seen during training. The field of *few-shot learning* aims to address this by developing models that can work with scarce data: only a few examples being available for training. One-shot classification is a special case of few-shot learning, where the system needs to learn to classify after seeing only one example of each class. Although deep learning models have made great progress in few-shot learning [1, 2, 3], reaching human level performance has been hard [4] even on simple datasets like Omniglot [5] which is one the main benchmarks for this task.

In this work, we propose to use a deep learning model with relational inductive bias [6] for the task of one-shot classification. The proposed model uses self-attention and cross-attention blocks of the Transformer [7] to compare a pair of images. Relational models with Transformer-based attention have recently been shown to succeed in vision tasks [8, 9, 10] and in this paper we show that this type of models can perform well in few-shot learning. Our proposed model is to the best of our knowledge the first deep learning model that achieves perfect (and better than human) accuracy in the most challenging Omniglot classification setup: one-shot within alphabet classification with 20 alternative classes without using data augmentation.

2 Omniglot challenge

Lake et al. [5] proposed Omniglot benchmark to measure progress of generalization from few examples in various tasks such as classification, generation of new examples and parsing objects into parts. Omniglot dataset contains characters from 50 alphabets, 1623 character classes in total, each character represented by only 20 hand-drawn instances, available both in image and pen trajectory formats (a few samples are shown in Fig. 1). The dataset is split into a 30-alphabet

Fig. 1: Omniglot data as images and points. Points that belong to the same pen stroke share color. Larger points mark the beginning, and empty circles – the end of the motion of the pen.

Table 1: Omniglot within alphabet one-shot, 20-way classification accuracy on the Evaluation set. Full and Min. denote the type of Background set used for training. We show the previously published results, when available, for: Human and BPL [5, 4], Prototypical Networks [3, 4] and ARC [11].

	Unaugmented		Augmented	
Model	Full	Min.	Full	Min.
Human	95.5	–	–	–
BPL	96.7	**95.8**	–	–
Prototypical Networks	86.3	69.9	94	–
ARC	–	–	97.75	–
ARC (full context)	–	–	98.5	–
AM (ours) – points	85.5	44.7	94	74.1
AM (ours) – images	97.5	79	98.25	95.1
AM (ours) – images (full context)	**100**	81.38	**99.5**	**98.5**

Background set (with 964 classes) used for training, and a 20-alphabet Evaluation set (with 659 classes). Recently, the "Minimal" [4] version of the training data has been proposed, containing two five-alphabet subsets of the Background set. The evaluation procedure of the N-way one-shot classification task consists of multiple episodes. Each episode is a classification task with N classes from the same alphabet. The classifier can use N *support* examples (one example per class) to assign N *query* examples to one of the N classes. The Background set can be used for training. For evaluation, we use pre-defined episodes from the Evaluation set with $N = 20$, which were proposed by the authors of Omniglot [5].

The authors of Omniglot developed a generative probabilistic model called Bayesian Program Learning (BPL)[1], as a non-neural baseline for Omniglot tasks. BPL has held the state of the art for the most challenging setup: the 20-way one-shot classification task without the use of data augmentations during training, it also generalizes very well from the Minimal training set (see Table 1). The authors argue [4] that the challenge remains unsolved because the human level on the *within-alphabet* classification has been out of reach for deep learning models that do not rely on heavy data augmentation. Most of the published research on Omniglot classification addresses a simpler *between-alphabets* classification task, in which testing episodes contain characters from different alphabets, making the examples easier to distinguish.

3 AttentiveMatcher: the proposed model

As humans, when we compare two unfamiliar images to see if they show the same item, we first look at each image to verify its contents, and then compare parts of each image to one other to see if all parts match. The same idea informs the approach presented in this paper: our model uses self-attention to learn the structure of each image, and cross-attention to match the corresponding
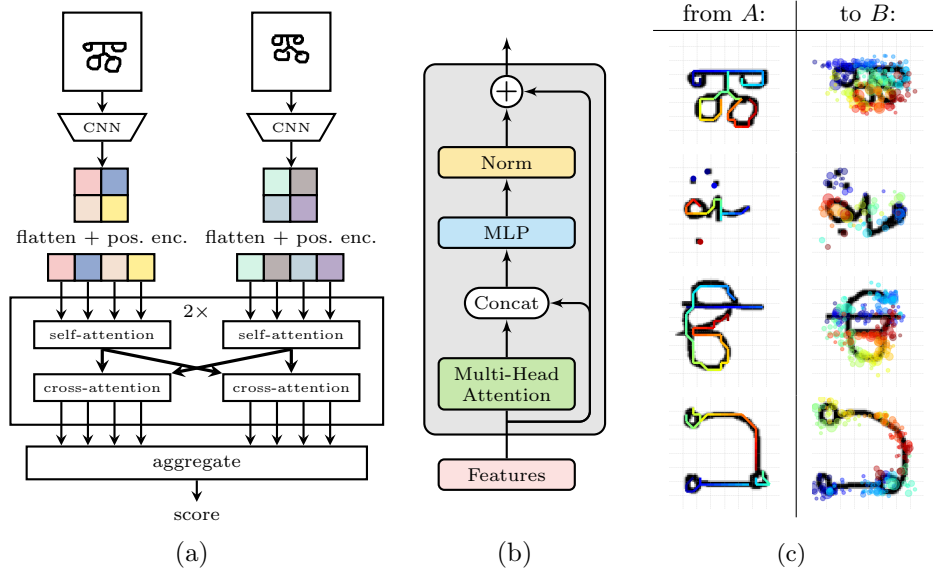
---

[1]https://github.com/brendenlake/BPL

Fig. 2: (a): The architecture of the proposed AttentiveMatcher model. (b): The attention layer used in our model. (c): Cross-attention strength from image A to image B, colors show which regions in A attend to which regions in B. The corresponding parts match despite being drawn quite differently.

parts between the two images, then the model aggregates this information to determine if images belong to the same class. With the above intuition in mind we call our model AttentiveMatcher (AM).

The training of AttentiveMatcher is performed on pairs of input, A and B, to predict whether they belong to the same class or not. The architecture of the model is illustrated in Fig. 2a. We use a convolutional network to embed the input images, obtaining a grid of 16×16 super-pixels, which we treat as 256 features. Similar to the hybrid model in [8] we use these features as inputs to the Transformer layers. First, we separately perform self-attention on the features coming from the two images. Next, we perform cross-attention between the features in the two inputs. We apply the above sequence of attention operations twice, in order for information in each feature to propagate across both inputs.

As the attention blocks, we use the Transformer [7] attention layers with the modifications inspired by [10]. Unlike in Transformer, where the outputs of the multi-head attention blocks are passed through a feed-forward block directly, we first concatenate the attention outputs with the input features before passing them to the feed-forward block followed by Layernorm [12] (see Fig. 2b). The feed-forward multilayer perceptron (MLP) is then followed by a residual connection. All the MLPs in our network have one ReLU-activated hidden layer.

After performing two rounds of self and cross-attention, we aggregate all the transformed local features into a single scalar score predicting the match between

A and B. The aggregation is inspired by the Graph Matching Networks [13]:

$$s = h\left(\sum_i \boldsymbol{\alpha}_i \odot f(\mathbf{z}_i)\right), \quad (\boldsymbol{\alpha}_i)_k = \frac{\exp g_k(\mathbf{z}_i)}{\sum_j \exp g_k(\mathbf{z}_j)}, \qquad (1)$$

where $\mathbf{z}_i$ are the local features from the two inputs, $(\boldsymbol{\alpha}_i)_k$ is the k-th element of K-dimensional vector $\boldsymbol{\alpha}_i$ and functions f, g_k, h are modeled with MLPs. The output layer of h has the sigmoid function, which results in score $s \in (0,1)$ treated as the probability that the two inputs belong to the same class. The training batches are constructed by sampling positive (same class) and negative (different classes) pairs from the training set. We use the binary cross-entropy loss to tune the parameters of the model.

The testing episodes are constructed by pairing each query input with each support input, resulting in $N \times N$ comparison pairs for each episode. Each query example is assigned a label of the support example with the highest score s. The model accuracy is defined as the fraction of the labels correctly assigned. To compare our model with ARC [11], which is the previous state of the art for the task, we also experimented in the scenario assumed by ARC when the query examples in one episode are known to belong to different classes. We used Hungarian matching [14] which assigns each support example to a unique query example in the episode, while maximizing overall score.

4   Experiments and results

We perform one-shot classification for 20-way Omniglot within alphabet task. Images are resized to 64×64 and passed through a four-layer convolutional network commonly used in few-shot models [3, 15]. We use kernel of size 3, with padding and the stride of 1, and we use max-pooling only after the first two convolutional layers. This results in a 16×16 feature map with $d_f = 128$ channels which we flatten into a sequence of 256 local features of size d_f that we use as input tokens to the Transformer layers. We use simple 1D positional encoding [7] for the inputs to the Transformer block and we use $K = 2048$ dimensions in the aggregation network in (1).

We run our experiments on both unaugmented and augmented data. When augmenting, we add rotations at $90°$, $180°$, $270°$ as extra training classes, similar to [3], resulting in 3856 classes. We use the batch size of 256 and we construct mini-batches such that the contribution of the positive examples in the loss is similar to the contribution of the negative examples. We use the Adam [16] optimizer with initial learning rate of 0.0001 reduced by factor of 0.1 every 100 epochs, and clip gradients at 2.5. While training, we validate by forming random within-alphabet episodes from the ten alphabets not in the test set. We stop the training once validation accuracy stops improving.

Additionally, we test our model with input characters represented as sequences of points. To extract points from Omniglot strokes, we follow the pre-processing done by BPL and linearly interpolate for equal spacing with 5 pixels

between neighbour points. We represent inputs as sequences of point coordinates. Each point is embedded using an MLP to size $d_f = 64$ and followed by dropout. We use $K = 1024$ in the aggregation network in (1).

Table 1 contains the obtained results. Our unaugmented model outperformed other published results, including BPL [5] and Prototypical Networks [3] and achieved perfect accuracy after applying Hungarian algorithm at test time, denoted by "full context" in the results. When training on the Minimal set, our model benefits from augmentation, and when full context is used, outperforms the previous state of the art, BPL. The model trained on points using data augmentation (three extra rotations per character) shows comparable accuracy to our image model, while the unaugmented results on points fall short of the corresponding image results. This is somewhat surprising, as one would assume that stroke trajectories contain the same amount of relevant information as the images. A likely reason for images performing better is the translation invariance induced by the convolutional network helping model to generalize better when compared to point-features prepared by simple MLPs.

The model allows us to visualize the attention, as shown in Fig. 2c. The cross-attention patterns obtained on Omniglot are meaningful: the classifier matches parts of its two inputs that represent the same strokes.

We also tested our model on natural images using the popular miniImageNet [1] benchmark. The model works well but does not exceed the state of the art. The performance of our model on natural images could also have been constrained by the small model size. We also studied the cross-attention patterns on the miniImageNet data and found that the model paid more attention to similarities in textures rather than semantically similar high-level parts and objects.

5   Conclusion

Our model obtains a perfect accuracy on the challenging Omniglot within alphabet classification task, exceeding both human level and BPL on unaugmented data, and to the best of our knowledge is the first model to do so. Our work shows that the relational inductive bias, in the form of local and pairwise attention, can be useful in solving few-shot learning problems. More work is needed to investigate the potential of this approach on natural images, where it is more challenging to develop meaningful relational representations in terms of objects and object parts. In addition, the relation between model size and interpretability of attention patterns for input matching should be explored. Working in this direction will drive the improvement in deep learning methods for efficient and scalable relational representation.

References

[1] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *Advances in neural information processing systems*, pages 3630–3638, 2016.

[2] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1126–1135. JMLR. org, 2017.

[3] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Advances in neural information processing systems*, pages 4077–4087, 2017.

[4] Brenden M Lake, Ruslan Salakhutdinov, and Joshua B Tenenbaum. The omniglot challenge: a 3-year progress report. *Current Opinion in Behavioral Sciences*, 29:97–104, 2019.

[5] Brenden M Lake, Ruslan Salakhutdinov, and Joshua B Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, 2015.

[6] Peter W Battaglia, Jessica B Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, et al. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*, 2018.

[7] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[8] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[9] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. *arXiv preprint arXiv:2104.14294*, 2021.

[10] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. *arXiv preprint arXiv:1911.11763*, 2019.

[11] Pranav Shyam, Shubham Gupta, and Ambedkar Dukkipati. Attentive recurrent comparators. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 3173–3181. JMLR. org, 2017.

[12] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[13] Yujia Li, Chenjie Gu, Thomas Dullien, Oriol Vinyals, and Pushmeet Kohli. Graph matching networks for learning the similarity of graph structured objects. *arXiv preprint arXiv:1904.12787*, 2019.

[14] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.

[15] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1199–1208, 2018.

[16] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.