Learning to Compare: Relation Network for Few-Shot Learning

Flood Sung Yongxin Yang[3] Li Zhang[2] Tao Xiang[1] Philip H.S. Torr[2] Timothy M. Hospedales[3]
[1]Queen Mary University of London [2]University of Oxford [3]The University of Edinburgh
floodsung@gmail.com t.xiang@qmul.ac.uk
{lz, phst}@robots.ox.ac.uk {yongxin.yang, t.hospedales}@ed.ac.uk

Abstract

We present a conceptually simple, flexible, and general framework for few-shot learning, where a classifier must learn to recognise new classes given only few examples from each. Our method, called the Relation Network (RN), is trained end-to-end from scratch. During meta-learning, it learns to learn a deep distance metric to compare a small number of images within episodes, each of which is designed to simulate the few-shot setting. Once trained, a RN is able to classify images of new classes by computing relation scores between query images and the few examples of each new class without further updating the network. Besides providing improved performance on few-shot learning, our framework is easily extended to zero-shot learning. Extensive experiments on five benchmarks demonstrate that our simple approach provides a unified and effective approach for both of these two tasks.

1. Introduction

Deep learning models have achieved great success in visual recognition tasks [22, 15, 35]. However, these supervised learning models need large amounts of labelled data and many iterations to train their large number of parameters. This severely limits their scalability to new classes due to annotation cost, but more fundamentally limits their applicability to newly emerging (eg. new consumer devices) or rare (eg. rare animals) categories where numerous annotated images may simply never exist. In contrast, humans are very good at recognising objects with very little direct supervision, or none at all *i.e.*, few-shot [23, 9] or zero-shot [24] learning. For example, children have no problem generalising the concept of "zebra" from a single picture in a book, or hearing its description as looking like a stripy horse. Motivated by the failure of conventional deep learning methods to work well on one or few examples per class, and inspired by the few- and zero-shot learning ability of humans, there has been a recent resurgence of interest in machine one/few-shot [8, 39, 32, 18, 20, 10, 27, 36, 29] and zero-shot [11, 3, 24, 45, 25, 31] learning.

Few-shot learning aims to recognise novel visual categories from very few labelled examples. The availability of only one or very few examples challenges the standard 'fine-tuning' practice in deep learning [10]. Data augmentation and regularisation techniques can alleviate overfitting in such a limited-data regime, but they do not solve it. Therefore contemporary approaches to few-shot learning often decompose training into an auxiliary meta learning phase where transferrable knowledge is learned in the form of good initial conditions [10], embeddings [36, 39] or optimisation strategies [29]. The target few-shot learning problem is then learned by fine-tuning [10] with the learned optimisation strategy [29] or computed in a feed-forward pass [36, 39, 4, 32] without updating network weights. Zero-shot learning also suffers from a related challenge. Recognisers are trained by a single example in the form of a class description (c.f., single exemplar image in one-shot), making data insufficiency for gradient-based learning a challenge.

While promising, most existing few-shot learning approaches either require complex inference mechanisms [23, 9], complex recurrent neural network (RNN) architectures [39, 32], or fine-tuning the target problem [10, 29]. Our approach is most related to others that aim to train an effective metric for one-shot learning [39, 36, 20]. Where they focus on the learning of the transferrable embedding and pre-define a fixed metric (e.g., as Euclidean [36]), we further aim to *learn* a transferrable deep metric for comparing the relation between images (few-shot learning), or between images and class descriptions (zero-shot learning). By expressing the inductive bias of a *deeper* solution (multiple non-linear learned stages at both embedding and relation modules), we make it easier to learn a generalisable solution to the problem.

Specifically, we propose a two-branch Relation Network (RN) that performs few-shot recognition by learning to compare *query* images against few-shot labeled *sample* images. First an *embedding module* generates representations of the query and training images. Then these embeddings are compared by a *relation module* that determines if they

are from matching categories or not. Defining an episode-based strategy inspired by [39, 36], the embedding and relation modules are meta-learned end-to-end to support few-shot learning. This can be seen as extending the strategy of [39, 36] to include a *learnable non-linear* comparator, instead of a fixed linear comparator. Our approach outperforms prior approaches, while being simpler (no RNNs [39, 32, 29]) and faster (no fine-tuning [29, 10]). Our proposed strategy also directly generalises to zero-shot learning. In this case the sample branch embeds a single-shot category description rather than a single exemplar training image, and the relation module learns to compare query image and category description embeddings.

Overall our contribution is to provide a clean framework that elegantly encompasses both few and zero-shot learning. Our evaluation on four benchmarks show that it provides compelling performance across the board while being simpler and faster than the alternatives.

2. Related Work

The study of one or few-shot object recognition has been of interest for some time [9]. Earlier work on few-shot learning tended to involve generative models with complex iterative inference strategies [9, 23]. With the success of discriminative deep learning-based approaches in the data-rich many-shot setting [22, 15, 35], there has been a surge of interest in generalising such deep learning approaches to the few-shot learning setting. Many of these approaches use a meta-learning or learning-to-learn strategy in the sense that they extract some transferrable knowledge from a set of auxiliary tasks (meta-learning, learning-to-learn), which then helps them to learn the target few-shot problem well without suffering from the overfitting that might be expected when applying deep models to sparse data problems.

Learning to Fine-Tune The successful MAML approach [10] aimed to meta-learn an initial condition (set of neural network weights) that is good for fine-tuning on few-shot problems. The strategy here is to search for the weight configuration of a given neural network such that it can be effectively fine-tuned on a sparse data problem within a few gradient-descent update steps. Many distinct target problems are sampled from a multiple task training set; the base neural network model is then fine-tuned to solve each of them, and the success at each target problem after fine-tuning drives updates in the base model – thus driving the production of an easy to fine-tune initial condition. The few-shot optimisation approach [29] goes further in meta-learning not only a good initial condition but an LSTM-based optimizer that is trained to be specifically effective for fine-tuning. However both of these approaches suffer from the need to fine-tune on the target problem. In contrast, our approach solves target problems in an entirely feed-forward manner with no model updates required, making it more convenient for low-latency or low-power applications.

RNN Memory Based Another category of approaches leverage recurrent neural networks with memories [27, 32]. Here the idea is typically that an RNN iterates over an examples of given problem and accumulates the knowledge required to solve that problem in its hidden activations, or external memory. New examples can be classified, for example by comparing them to historic information stored in the memory. So 'learning' a single target problem can occur in unrolling the RNN, while learning-to-learn means training the weights of the RNN by learning many distinct problems. While appealing, these architectures face issues in ensuring that they reliably store all the, potentially long term, historical information of relevance without forgetting. In our approach we avoid the complexity of recurrent networks, and the issues involved in ensuring the adequacy of their memory. Instead our learning-to-learn approach is defined entirely with simple and fast feed forward CNNs.

Embedding and Metric Learning Approaches The prior approaches entail some complexity when learning the target few-shot problem. Another category of approach aims to learn a set of projection functions that take query and sample images from the target problem and classify them in a feed forward manner [39, 36, 4]. One approach is to parameterise the weights of a feed-forward classifier in terms of the sample set [4]. The meta-learning here is to train the auxiliary parameterisation net that learns how to paramaterise a given feed-forward classification problem in terms of a few-shot sample set. Metric-learning based approaches aim to learn a set of projection functions such that when represented in this embedding, images are easy to recognise using simple nearest neighbour or linear classifiers [39, 36, 20]. In this case the meta-learned transferrable knowledge are the projection functions and the target problem is a simple feed-forward computation.

The most related methodologies to ours are the prototypical networks of [36] and the siamese networks of [20]. These approaches focus on learning embeddings that transform the data such that it can be recognised with a *fixed* nearest-neighbour [36] or linear [20, 36] classifier. In contrast, our framework further defines a relation classifier CNN, in the style of [33, 44, 14] (While [33] focuses on reasoning about relation between two objects in a same image which is to address a different problem.). Compared to [20, 36], this can be seen as providing a learnable rather than fixed metric, or non-linear rather than linear classifier. Compared to [20] we benefit from an episodic training strategy with an end-to-end manner from scratch, and compared to [32] we avoid the complexity of set-to-set RNN embedding of the sample-set, and simply rely on pooling [33].

Zero-Shot Learning Our approach is designed for few-

shot learning, but elegantly spans the space into zero-shot learning (ZSL) by modifying the sample branch to input a single category description rather than single training image. When applied to ZSL our architecture is related to methods that learn to align images and category embeddings and perform recognition by predicting if an image and category embedding pair match [11, 3, 43, 46]. Similarly to the case with the prior metric-based few-shot approaches, most of these apply a fixed manually defined similarity metric or linear classifier after combining the image and category embedding. In contrast, we again benefit from a deeper end-to-end architecture including a learned non-linear metric in the form of our learned convolutional relation network; as well as from an episode-based training strategy.

3. Methodology

3.1. Problem Definition

We consider the task of few-shot classifier learning. Formally, we have three datasets: a training set, a support set, and a testing set. The support set and testing set share the same label space, but the training set has its own label space that is disjoint with support/testing set. If the support set contains K labelled examples for each of C unique classes, the target few-shot problem is called C-way K-shot.

With the support set only, we can in principle train a classifier to assign a class label $\hat{y}$ to each sample $\hat{x}$ in the test set. However, due to the lack of labelled samples in the support set, the performance of such a classifier is usually not satisfactory. Therefore we aim to perform meta-learning on the training set, in order to extract transferrable knowledge that will allow us to perform better few-shot learning on the support set and thus classify the test set more successfully.

An effective way to exploit the training set is to mimic the few-shot learning setting via *episode* based training, as proposed in [39]. In each training iteration, an episode is formed by randomly selecting C classes from the training set with K labelled samples from each of the C classes to act as the *sample* set $\mathcal{S} = \{(x_i, y_i)\}_{i=1}^{m}$ ($m = K \times C$), as well as a fraction of the remainder of those C classes' samples to serve as the *query* set $\mathcal{Q} = \{(x_j, y_j)\}_{j=1}^{n}$. This sample/query set split is designed to simulate the support/test set that will be encountered at test time. A model trained from sample/query set can be further fine-tuned using the support set, if desired. In this work we adopt such an episode-based training strategy. In our few-shot experiments (see Section 4.1) we consider one-shot ($K = 1$, Figure 1) and five-shot ($K = 5$) settings. We also address the $K = 0$ zero-shot learning case as explained in Section 3.3.

3.2. Model

One-Shot Our Relation Network (RN) consists of two modules: an *embedding* module f_φ and a *relation* module g_ϕ, as illustrated in Figure 1. Samples x_j in the query set $\mathcal{Q}$, and samples x_i in the sample set $\mathcal{S}$ are fed through the embedding module f_φ, which produces feature maps $f_\varphi(x_i)$ and $f_\varphi(x_j)$. The feature maps $f_\varphi(x_i)$ and $f_\varphi(x_j)$ are combined with operator $\mathcal{C}(f_\varphi(x_i), f_\varphi(x_j))$. In this work we assume $\mathcal{C}(\cdot, \cdot)$ to be concatenation of feature maps in depth, although other choices are possible.

The combined feature map of the sample and query are fed into the relation module g_ϕ, which eventually produces a scalar in range of 0 to 1 representing the similarity between x_i and x_j, which is called relation score. Thus, in the C-way one-shot setting, we generate C relation scores $r_{i,j}$ for the relation between one query input x_j and training sample set examples x_i,

$$r_{i,j} = g_\phi(\mathcal{C}(f_\varphi(x_i), f_\varphi(x_j))), \quad i = 1, 2, \ldots, C \tag{1}$$

K-shot For K-shot where $K > 1$, we element-wise sum over the embedding module outputs of all samples from each training class to form this class' feature map. This pooled class-level feature map is combined with the query image feature map as above. Thus, the number of relation scores for one query is always C in both one-shot or few-shot setting.

Objective function We use mean square error (MSE) loss (Eq. (2)) to train our model, regressing the relation score $r_{i,j}$ to the ground truth: matched pairs have similarity 1 and the mismatched pair have similarity 0.

$$\varphi, \phi \leftarrow \underset{\varphi,\phi}{\operatorname{argmin}} \sum_{i=1}^{m} \sum_{j=1}^{n} (r_{i,j} - \mathbf{1}(y_i == y_j))^2 \tag{2}$$

The choice of MSE is somewhat non-standard. Our problem may seem to be a classification problem with a label space $\{0, 1\}$. However conceptually we are predicting relation scores, which can be considered a regression problem despite that for ground-truth we can only automatically generate $\{0, 1\}$ targets.

3.3. Zero-shot Learning

Zero-shot learning is analogous to one-shot learning in that one datum is given to define each class to recognise. However instead of being given a support set with one-shot image for each of C training classes, it contains a semantic class embedding vector v_c for each. Modifying our framework to deal with the zero-shot case is straightforward: as a different modality of semantic vectors is used for the *support* set (*e.g.* attribute vectors instead of images), we use a

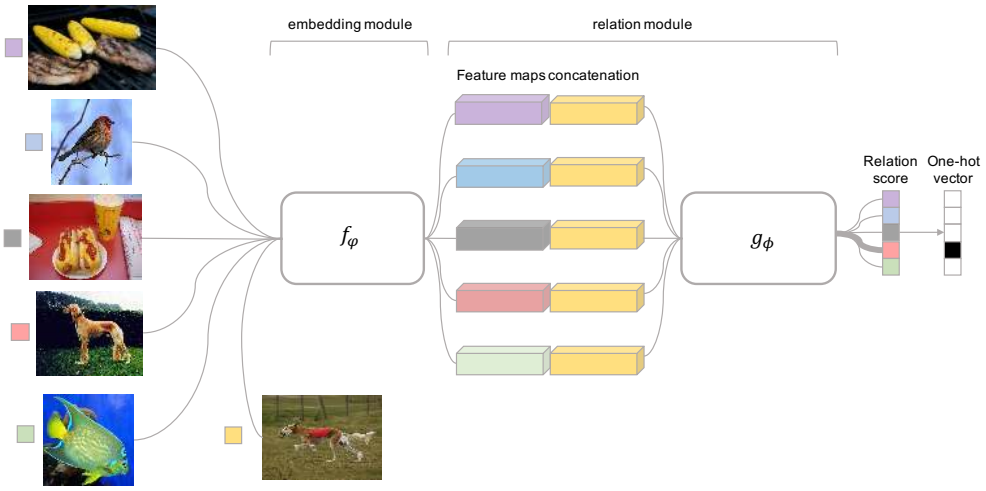


Figure 1: Relation Network architecture for a 5-way 1-shot problem with one query example.

second heterogeneous embedding module f_{φ_2} besides the embedding module f_{φ_1} used for the image *query* set. Then the relation net g_ϕ is applied as before. Therefore, the relation score for each query input x_j will be:

$$r_{i,j} = g_\phi(\mathcal{C}(f_{\varphi_1}(v_c), f_{\varphi_2}(x_j))), \quad i = 1, 2, \ldots, C \quad (3)$$

The objective function for zero-shot learning is the same as that for few-shot learning.

3.4. Network Architecture

As most few-shot learning models utilise four convolutional blocks for embedding module [39, 36], we follow the same architecture setting for fair comparison, see Figure 2. More concretely, each convolutional block contains a 64-filter 3×3 convolution, a batch normalisation and a ReLU nonlinearity layer respectively. The first two blocks also contain a 2×2 max-pooling layer while the latter two do not. We do so because we need the output feature maps for further convolutional layers in the relation module. The relation module consists of two convolutional blocks and two fully-connected layers. Each of convolutional block is a 3×3 convolution with 64 filters followed by batch normalisation, ReLU non-linearity and 2×2 max-pooling. The output size of last max pooling layer is $\mathcal{H} = 64$ and $\mathcal{H} = 64 * 3 * 3 = 576$ for Omniglot and *mini*ImageNet respectively. The two fully-connected layers are 8 and 1 dimensional, respectively. All fully-connected layers are ReLU except the output layer is Sigmoid in order to generate relation scores in a reasonable range for all versions of our network architecture.

The zero-shot learning architecture is shown in Figure 3. In this architecture, the DNN subnet is an existing network (e.g., Inception or ResNet) pretrained on ImageNet.

4. Experiments

We evaluate our approach on two related tasks: few-shot classification on Omniglot and *mini*Imagenet, and zero-shot classification on Animals with Attributes (AwA) and Caltech-UCSD Birds-200-2011 (CUB). All the experiments are implemented based on PyTorch [1].

4.1. Few-shot Recognition

Settings Few-shot learning in all experiments uses Adam [19] with initial learning rate 10^{-3} , annealed by half for every 100,000 episodes. All our models are end-to-end trained from scratch with no additional dataset.

Baselines We compare against various state of the art baselines for few-shot recognition, including neural statistician [8], Matching Nets with and without fine-tuning [39], MANN [32], Siamese Nets with Memory [18], Convolutional Siamese Nets [20], MAML [10], Meta Nets [27], Prototypical Nets [36] and Meta-Learner LSTM [29].

4.1.1 Omniglot

Dataset Omniglot [23] contains 1623 characters (classes) from 50 different alphabets. Each class contains 20 samples drawn by different people. Following [32, 39, 36], we augment new classes through $90°$, $180°$ and $270°$ rotations of existing data and use 1200 original classes plus rotations for training and remaining 423 classes plus rotations for testing. All input images are resized to 28×28.

Training Besides the K sample images, the **5-way 1-shot** contains 19 query images, the **5-way 5-shot** has 15 query images, the **20-way 1-shot** has 10 query images and the **20-way 5-shot** has 5 query images for each of the C sampled classes in each training episode. This means for

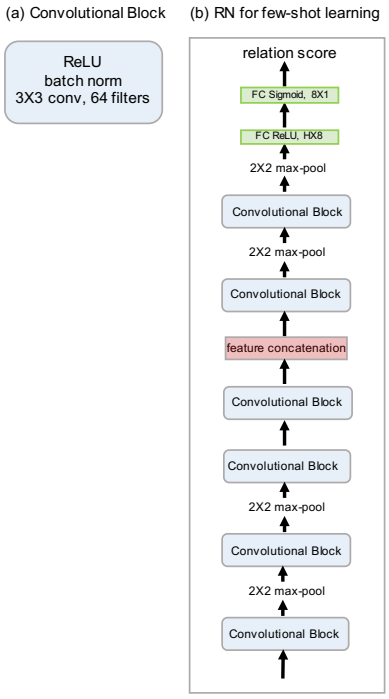


Figure 2: Relation Network architecture for few-shot learning (b) which is composed of elements including convolutional block (a).

example that there are $19 \times 5 + 1 \times 5 = 100$ images in one training episode/mini-batch for the 5-way 1-shot experiments.

Results Following [36], we computed few-shot classification accuracies on Omniglot by averaging over 1000 randomly generated episodes from the testing set. For the 1-shot and 5-shot experiments, we batch one and five query images per class respectively for evaluation during testing. The results are shown in Table 1. We achieved state-of-the-art performance under all experiments setting with higher averaged accuracies and lower standard deviations, except 5-way 5-shot where our model is 0.1% lower in accuracy than [10]. This is despite that many alternatives have significantly more complicated machinery [27, 8], or fine-tune on the target problem [10, 39], while we do not.

4.1.2 *mini*ImageNet

Dataset The *mini*Imagenet dataset, originally proposed by [39], consists of 60,000 colour images with 100 classes, each having 600 examples. We followed the split introduced by [29], with 64, 16, and 20 classes for training, validation and testing, respectively. The 16 validation classes is used for monitoring generalisation performance only.

Training Following the standard setting adopted by most existing few-shot learning work, we conducted 5 way 1-shot and 5-shot classification. Beside the K sample images, the

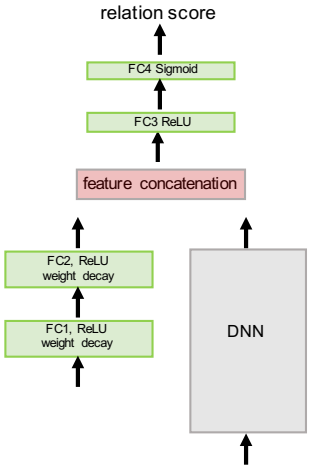


Figure 3: Relation Network architecture for zero-shot learning.

5-way 1-shot contains 15 query images, and the **5-way 5-shot** has 10 query images for each of the C sampled classes in each training episode. This means for example that there are $15 \times 5 + 1 \times 5 = 80$ images in one training episode/mini-batch for 5-way 1-shot experiments. We resize input images to 84×84. Our model is trained end-to-end from scratch, with random initialisation, and no additional training set.

Results Following [36], we batch 15 query images per class in each episode for evaluation in both 1-shot and 5-shot scenarios and the few-shot classification accuracies are computed by averaging over 600 randomly generated episodes from the test set.

From Table 2, we can see that our model achieved state-of-the-art performance on 5-way 1-shot settings and competitive results on 5-way 5-shot. However, the 1-shot result reported by prototypical networks [36] reqired to be trained on 30-way 15 queries per training episode, and 5-shot result was trained on 20-way 15 queries per training episode. When trained with 5-way 15 query per training episode, [36] only got $46.14 \pm 0.77\%$ for 1-shot evaluation, clearly weaker than ours. In contrast, all our models are trained on 5-way, 1 query for 1-shot and 5 queries for 5-shot per training episode, with much less training queries than [36].

4.2. Zero-shot Recognition

Datasets and settings We follow two ZSL settings: the *old* setting and the new *GBU* setting provided by [42] for training/test splits. Under the *old* setting, adopted by most existing ZSL works before [42], some of the test classes also appear in the ImageNet 1K classes, which have been used to pretrain the image embedding network, thus violating the zero-shot assumption. In contrast, the new *GBU* setting ensures that none of the test classes of the datasets appear in the ImageNet 1K classes. Under both settings, the

Model	Fine Tune	5-way Acc.		20-way Acc.	
		1-shot	5-shot	1-shot	5-shot
MANN [32]	N	82.8%	94.9%	-	-
CONVOLUTIONAL SIAMESE NETS [20]	N	96.7%	98.4%	88.0%	96.5%
CONVOLUTIONAL SIAMESE NETS [20]	Y	97.3%	98.4%	88.1%	97.0%
MATCHING NETS [39]	N	98.1%	98.9%	93.8%	98.5%
MATCHING NETS [39]	Y	97.9%	98.7%	93.5%	98.7%
SIAMESE NETS WITH MEMORY [18]	N	98.4%	99.6%	95.0%	98.6%
NEURAL STATISTICIAN [8]	N	98.1%	99.5%	93.2%	98.1%
META NETS [27]	N	99.0%	-	97.0%	-
PROTOTYPICAL NETS [36]	N	98.8%	99.7%	96.0%	98.9%
MAML [10]	Y	$98.7 \pm 0.4\%$	$\mathbf{99.9 \pm 0.1\%}$	$95.8 \pm 0.3\%$	$98.9 \pm 0.2\%$
RELATION NET	N	$\mathbf{99.6 \pm 0.2\%}$	$\mathbf{99.8 \pm 0.1\%}$	$\mathbf{97.6 \pm 0.2\%}$	$\mathbf{99.1 \pm 0.1\%}$

Table 1: Omniglot few-shot classification. Results are accuracies averaged over 1000 test episodes and with 95% confidence intervals where reported. The best-performing method is highlighted, along with others whose confidence intervals overlap. '-': not reported.

Model	FT	5-way Acc.	
		1-shot	5-shot
MATCHING NETS [39]	N	$43.56 \pm 0.84\%$	$55.31 \pm 0.73\%$
META NETS [27]	N	$49.21 \pm 0.96\%$	-
META-LEARN LSTM [29]	N	$43.44 \pm 0.77\%$	$60.60 \pm 0.71\%$
MAML [10]	Y	$48.70 \pm 1.84\%$	$63.11 \pm 0.92\%$
PROTOTYPICAL NETS [36]	N	$49.42 \pm 0.78\%$	$\mathbf{68.20 \pm 0.66\%}$
RELATION NET	N	$\mathbf{50.44 \pm 0.82\%}$	$65.32 \pm 0.70\%$

Table 2: Few-shot classification accuracies on *mini*Imagenet. All accuracy results are averaged over 600 test episodes and are reported with 95% confidence intervals, same as [36]. For each task, the best-performing method is highlighted, along with any others whose confidence intervals overlap. '-': not reported.

test set can comprise only the unseen class samples (conventional test set setting) or a mixture of seen and unseen class samples. The latter, termed generalised zero-shot learning (GZSL), is more realistic in practice.

Two widely used ZSL benchmarks are selected for the *old* setting: **AwA** (Animals with Attributes) [24] consists of 30,745 images of 50 classes of animals. It has a fixed split for evaluation with 40 training classes and 10 test classes. **CUB** (Caltech-UCSD Birds-200-2011) [40] contains 11,788 images of 200 bird species with 150 seen classes and 50 disjoint unseen classes. Three datasets [42] are selected for *GBU* setting: **AwA1**, **AwA2** and **CUB**. The newly released AwA2 [42] consists of 37,322 images of 50 classes which is an extension of AwA while AwA1 is same as AwA but under the *GBU* setting.

Semantic representation For **AwA**, we use the continuous 85-dimension class-level attribute vector from [24], which has been used by all recent works. For **CUB**, a continuous 312-dimension class-level attribute vector is used.

Implementation details Two different embedding modules are used for the two input modalities in zero-shot learning. Unless otherwise specified, we use Inception-V2 [38, 17] as the query image embedding DNN in the old and conventional setting and ResNet101 [16] for the GBU and generalised setting, taking the top pooling units as image embedding with dimension $D = 1024$ and 2048 respectively. This DNN is pre-trained on ILSVRC 2012 1K classification without fine-tuning, as in recent deep ZSL works [25, 30, 45]. A MLP network is used for embedding semantic attribute vectors. The size of hidden layer FC1 (Figure 3) is set to 1024 and 1200 for AwA and CUB respectively, and the output size FC2 is set to the same dimension as the image embedding for both datasets. For the relation module, the image and semantic embeddings are concatenated before being fed into MLPs with hidden layer FC3 size 400 and 1200 for AwA and CUB, respectively.

We add weight decay (L2 regularisation) in FC1 & 2 as there is a hubness problem [45] in cross-modal mapping for ZSL which can be best solved by mapping the semantic feature vector to the visual feature space with regularisation. After that, FC3 & 4 (relation module) are used to compute the relation between the semantic representation (in the visual feature space) and the visual representation. Since the hubness problem does not existing in this step, no L2 regularisation/weight decay is needed. All the ZSL models are trained with weight decay 10^{-5} in the embedding network. The learning rate is initialised to 10^{-5} with Adam [19] and then annealed by half every 200,000 iterations.

Results under the old setting The conventional evaluation for ZSL followed by the majority of prior work is to assume that the test data all comes from unseen classes. We evaluate this setting first. We compare 15 alternative approaches in Table 3. With only the attribute vector used as the sample class embedding, our model achieves competitive result on AwA and state-of-the-art performance on the more challenging CUB dataset, outperforming the most related alternative prototypical networks [36] by a big margin. Note that only inductive methods are considered. Some re-

Model	F	SS	AwA 10-way 0-shot	CUB 50-way 0-shot
SJE [3]	F_G	A	66.7	50.1
ESZSL [31]	F_G	A	76.3	47.2
SSE-RELU [46]	F_V	A	76.3	30.4
JLSE [47]	F_V	A	80.5	42.1
SYNC-STRUCT [6]	F_G	A	72.9	54.5
SEC-ML [5]	F_V	A	77.3	43.3
PROTO. NETS [36]	F_G	A	-	54.6
DEVISE [11]	N_G	A/W	56.7/50.4	33.5
SOCHER *et al.* [37]	N_G	A/W	60.8/50.3	39.6
MTMDL [43]	N_G	A/W	63.7/55.3	32.3
BA *et al.* [25]	N_G	A/W	69.3/58.7	34.0
DS-SJE [30]	N_G	A/D	-	50.4/ 56.8
SAE [21]	N_G	A	84.7	61.4
DEM [45]	N_G	A/W	**86.7**/78.8	58.3
RELATION NET	N_G	A	84.5	**62.0**

Table 3: Zero-shot classification accuracy (%) comparison on AwA and CUB (hit@1 accuracy over all samples) under the old and conventional setting. SS: semantic space; A: attribute space; W: semantic word vector space; D: sentence description (only available for CUB). F: how the visual feature space is computed; For non-deep models: F_O if overfeat [34] is used; F_G for GoogLeNet [38]; and F_V for VGG net [35]. For neural network based methods, all use Inception-V2 (GoogLeNet with batch normalisation) [38, 17] as the DNN image imbedding subnet, indicated as N_G.

cent methods [48, 12, 13] are tranductive in that they use all test data at once for model training, which gives them a big advantage at the cost of making a very strong assumption that may not be met in practical applications, so we do not compare with them here.

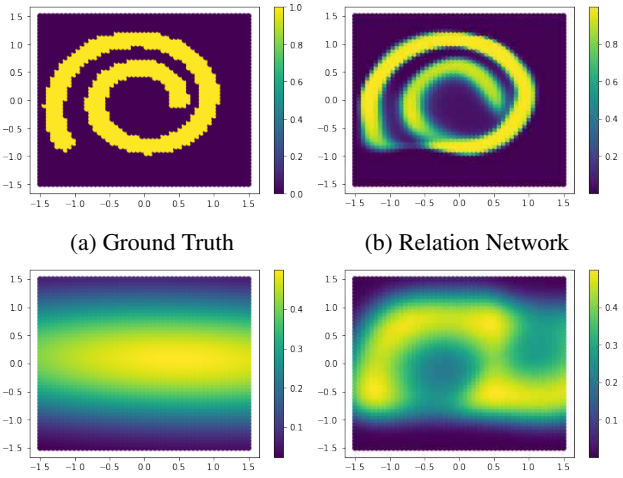
Results under the GBU setting We follow the evaluation setting of [42]. We compare our model with 11 alternative ZSL models in Table 4. The 10 shallow models results are from [42] and the result of the state-of-the-art method DEM [45] is from the authors' GitHub page[1]. We can see that on AwA2 and CUB, Our model is particularly strong under the more realistic GZSL setting measured using the harmonic mean (H) metric. While on AwA1, our method is only outperformed by DEM [45].

5. Why does Relation Network Work?

5.1. Relationship to existing models

Related prior few-shot work uses fixed pre-specified distance metrics such as Euclidean or cosine distance to perform classification [39, 36]. These studies can be seen as distance metric learning, but where all the learning occurs in the feature embedding, and a fixed metric is used given the learned embedding. Also related are conventional metric learning approaches [26, 7] that focus on learning a shallow (linear) Mahalanobis metric for a fixed feature representation. In contrast to prior work's fixed metric or fixed features and shallow learned metric, Relation Network can be seen as both learning a deep embedding *and* learning a deep non-linear metric (similarity function)[2]. These are mutually tuned end-to-end to support each other in few short learning.

Why might this be particularly useful? By using a flexible function approximator to learn similarity, we learn a good metric in a data driven way and do not have to manually choose the right metric (Euclidean, cosine, Mahalanobis). Fixed metrics like [39, 36] assume that features are solely compared element-wise, and the most related [36] assumes linear separability after the embedding. These are thus critically dependent on the efficacy of the learned embedding network, and hence limited by the extent to which the embedding networks generate inadequately discriminative representations. In contrast, by deep learning a non-linear similarity metric jointly with the embedding, Relation Network can better identify matching/mismatching pairs.



(a) Ground Truth (b) Relation Network (c) Metric Learning (d) Metric + Embedding

Figure 4: An example relation learnable by Relation Network and not by non-linear embedding + metric learning.

5.2. Visualisation

To illustrate the previous point about adequacy of learned input embeddings, we show a synthetic example where existing approaches definitely fail and our Relation Network can succeed due to using a deep relation module. Assuming 2D query and sample input embeddings to a relation module, Fig. 4(a) shows the space of 2D sample inputs for a fixed 2D query input. Each sample input (pixel) is colored according to whether it matches the fixed query or not. This

[1] https://github.com/lzrobots/DeepEmbeddingModel_ZSL

[2] Our architecture does not guarantee the self-similarity and symmetry properties of a formal similarity function. But empirically we find these properties hold numerically for a trained Relation Network.

	AwA1				AwA2				CUB			
Model	ZSL T1	GZSL u	GZSL s	GZSL H	ZSL T1	GZSL u	GZSL s	GZSL H	ZSL T1	GZSL u	GZSL s	GZSL H
DAP [24]	44.1	0.0	88.7	0.0	46.1	0.0	84.7	0.0	40.0	1.7	67.9	3.3
CONSE [28]	45.6	0.4	88.6	0.8	44.5	0.5	90.6	1.0	34.3	1.6	**72.2**	3.1
SSE [46]	60.1	7.0	80.5	12.9	61.0	8.1	82.5	14.8	43.9	8.5	46.9	14.4
DEVISE [11]	54.2	13.4	68.7	22.4	59.7	17.1	74.7	27.8	52.0	23.8	53.0	32.8
SJE [3]	65.6	11.3	74.6	19.6	61.9	8.0	73.9	14.4	53.9	23.5	59.2	33.6
LATEM [41]	55.1	7.3	71.7	13.3	55.8	11.5	77.3	20.0	49.3	15.2	57.3	24.0
ESZSL [31]	58.2	6.6	75.6	12.1	58.6	5.9	77.8	11.0	53.9	12.6	63.8	21.0
ALE [2]	59.9	16.8	76.1	27.5	62.5	14.0	81.8	23.9	54.9	23.7	62.8	34.4
SYNC [6]	54.0	8.9	87.3	16.2	46.6	10.0	90.5	18.0	55.6	11.5	70.9	19.8
SAE [21]	53.0	1.8	77.1	3.5	54.1	1.1	82.2	2.2	33.3	7.8	57.9	29.2
DEM [45]	**68.4**	**32.8**	84.7	**47.3**	**67.1**	**30.5**	86.4	45.1	51.7	19.6	54.0	13.6
RELATION NET	68.2	31.4	**91.3**	46.7	64.2	30.0	**93.4**	**45.3**	**55.6**	**38.1**	61.1	**47.0**

Table 4: Comparative results under the GBU setting. Under the conventional ZSL setting, the performance is evaluated using per-class average Top-1 (**T1**) accuracy (%), and under GZSL, it is measured using **u** = **T1** on unseen classes, **s** = **T1** on seen classes, and **H** = harmonic mean.

represents a case where the output of the embedding modules is not discriminative enough for trivial (Euclidean NN) comparison between query and sample set. In Fig. 4(c) we attempt to learn matching via a Mahalanobis metric learning relation module, and we can see the result is inadequate. In Fig. 4(d) we learn a further 2-hidden layer MLP embedding of query and sample inputs as well as the subsequent Mahalanobis metric, which is also not adequate. Only by learning the full deep relation module for similarity can we solve this problem in Fig. 4(b).

In a real problem the difficulty of comparing embeddings may not be this extreme, but it can still be challenging. We qualitatively illustrate the challenge of matching two example Omniglot query images (embeddings projected to 2D, Figure 5(left)) by showing an analogous plot of real sample images colored by match (cyan) or mismatch (magenta) to two example queries (yellow). Under standard assumptions [39, 36, 26, 7] the cyan matching samples should be nearest neighbours to the yellow query image with some metric (Euclidean, Cosine, Mahalanobis). But we can see that the match relation is more complex than this. In Figure 5(right), we instead plot the same two example queries in terms of a 2D PCA representation of each query-sample pair, as represented by the relation module's penultimate layer. We can see that the relation network has mapped the data into a space where the (mis)matched pairs are linearly separable.

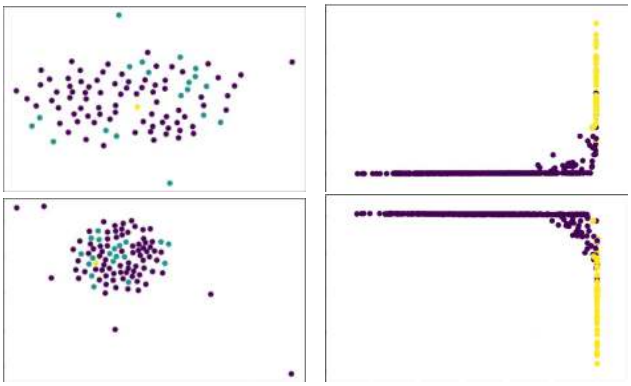


Figure 5: Example Omniglot few-shot problem visualisations. Left: Matched (cyan) and mismatched (magenta) sample embeddings for a given query (yellow) are not straightforward to differentiate. Right: Matched (yellow) and mismatched (magenta) relation module pair representations are linearly separable.

6. Conclusion

We proposed a simple method called the Relation Network for few-shot and zero-shot learning. Relation network learns an embedding and a deep non-linear distance metric for comparing query and sample items. Training the network end-to-end with episodic training tunes the embedding and distance metric for effective few-shot learning. This approach is far simpler and more efficient than recent few-shot meta-learning approaches, and produces state-of-the-art results. It further proves effective at both conventional and generalised zero-shot settings.

Acknowledgements This work was supported by the ERC grant ERC-2012-AdG 321162-HELIOS, EPSRC grant Seebibyte EP/M013774/1, EPSRC/MURI grant EP/N019474/1, EPSRC grant EP/R026173/1, and the European Union's Horizon 2020 research and innovation program (grant agreement no. 640891). We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPU and the EPSRC funded Tier 2 facility, JADE used for this research.

References

[1] Pytorch. https://github.com/pytorch/pytorch. 4

[2] Z. Akata, F. Perronnin, Z. Harchaoui, and C. Schmid. Label-embedding for image classification. *TPAMI*, 2016. 8

[3] Z. Akata, S. Reed, D. Walter, H. Lee, and B. Schiele. Evaluation of output embeddings for fine-grained image classification. In *CVPR*, 2015. 1, 3, 7, 8

[4] L. Bertinetto, J. F. Henriques, J. Valmadre, P. H. S. Torr, and A. Vedaldi. Learning feed-forward one-shot learners. In *NIPS*, 2016. 1, 2

[5] M. Bucher, S. Herbin, and F. Jurie. Improving semantic embedding consistency by metric learning for zero-shot classiffication. In *ECCV*, 2016. 7

[6] S. Changpinyo, W.-L. Chao, B. Gong, and F. Sha. Synthesized classifiers for zero-shot learning. In *CVPR*, 2016. 7, 8

[7] D. Chen, X. Cao, L. Wang, F. Wen, and J. Sun. Bayesian face revisited: A joint formulation. In *ECCV*. Springer Berlin Heidelberg, 2012. 7, 8

[8] H. Edwards and A. Storkey. Towards a neural statistician. *ICLR*, 2017. 1, 4, 5, 6

[9] L. Fei-Fei, R. Fergus, and P. Perona. One-shot learning of object categories. *TPAMI*, 2006. 1, 2

[10] C. Finn, P. Abbeel, and S. Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*, 2017. 1, 2, 4, 5, 6

[11] A. Frome, G. S. Corrado, J. Shlens, S. Bengio, J. Dean, T. Mikolov, et al. Devise: A deep visual-semantic embedding model. In *NIPS*, 2013. 1, 3, 7, 8

[12] Y. Fu, T. M. Hospedales, T. Xiang, Z. Fu, and S. Gong. Transductive multi-view embedding for zero-shot recognition and annotation. In *ECCV*, 2014. 7

[13] Y. Fu and L. Sigal. Semi-supervised vocabulary-informed learning. In *CVPR*, 2016. 7

[14] X. Han, T. Leung, Y. Jia, R. Sukthankar, and A. C. Berg. Matchnet: Unifying feature and metric learning for patch-based matching. In *CVPR*, 2015. 2

[15] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 1, 2

[16] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 6

[17] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015. 6, 7

[18] Ł. Kaiser, O. Nachum, A. Roy, and S. Bengio. Learning to remember rare events. *ICLR*, 2017. 1, 4, 6

[19] D. Kingma and J. Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 4, 6

[20] G. Koch, R. Zemel, and R. Salakhutdinov. Siamese neural networks for one-shot image recognition. In *ICML Workshop*, 2015. 1, 2, 4, 6

[21] E. Kodirov, T. Xiang, and S. Gong. Semantic autoencoder for zero-shot learning. In *CVPR*, 2017. 7, 8

[22] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012. 1, 2

[23] B. Lake, R. Salakhutdinov, J. Gross, and J. Tenenbaum. One shot learning of simple visual concepts. In *CogSci*, 2011. 1, 2, 4

[24] C. H. Lampert, H. Nickisch, and S. Harmeling. Attribute-based classification for zero-shot visual object categorization. *PAMI*, 2014. 1, 6, 8

[25] J. Lei Ba, K. Swersky, S. Fidler, and R. Salakhutdinov. Predicting deep zero-shot convolutional neural networks using textual descriptions. In *ICCV*, 2015. 1, 6, 7

[26] T. Mensink, J. Verbeek, F. Perronnin, and G. Csurka. Metric learning for large scale image classification: Generalizing to new classes at near-zero cost. In *ECCV*, 2012. 7, 8

[27] T. Munkhdalai and H. Yu. Meta networks. In *ICML*, 2017. 1, 2, 4, 5, 6

[28] M. Norouzi, T. Mikolov, S. Bengio, Y. Singer, J. Shlens, A. Frome, G. S. Corrado, and J. Dean. Zero-shot learning by convex combination of semantic embeddings. In *ICLR*, 2014. 8

[29] S. Ravi and H. Larochelle. Optimization as a model for few-shot learning. In *ICLR*, 2017. 1, 2, 4, 5, 6

[30] S. Reed, Z. Akata, B. Schiele, and H. Lee. Learning deep representations of fine-grained visual descriptions. In *CVPR*, 2016. 6, 7

[31] B. Romera-Paredes and P. Torr. An embarrassingly simple approach to zero-shot learning. In *ICML*, 2015. 1, 7, 8

[32] A. Santoro, S. Bartunov, M. Botvinick, D. Wierstra, and T. Lillicrap. Meta-learning with memory-augmented neural networks. In *ICML*, 2016. 1, 2, 4, 6

[33] A. Santoro, D. Raposo, D. G. Barrett, M. Malinowski, R. Pascanu, P. Battaglia, and T. Lillicrap. A simple neural network module for relational reasoning. In *NIPS*, 2017. 2

[34] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv preprint arXiv:1312.6229*, 2013. 7

[35] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *ICLR*, 2015. 1, 2, 7

[36] J. Snell, K. Swersky, and R. S. Zemel. Prototypical networks for few-shot learning. In *NIPS*, 2017. 1, 2, 4, 5, 6, 7, 8

[37] R. Socher, M. Ganjoo, C. D. Manning, and A. Ng. Zero-shot learning through cross-modal transfer. In *NIPS*, 2013. 7

[38] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *CVPR*, 2015. 6, 7

[39] O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra, et al. Matching networks for one shot learning. In *NIPS*, 2016. 1, 2, 3, 4, 5, 6, 7, 8

[40] C. Wah, S. Branson, P. Perona, and S. Belongie. Multiclass recognition and part localization with humans in the loop. In *ICCV*, 2011. 6

[41] Y. Xian, Z. Akata, G. Sharma, Q. Nguyen, M. Hein, and B. Schiele. Latent embeddings for zero-shot classification. In *CVPR*, 2016. 8

[42] Y. Xian, C. H. Lampert, B. Schiele, and Z. Akata. Zero-shot learning-a comprehensive evaluation of the good, the bad and the ugly. *arXiv preprint arXiv:1707.00600*, 2017. 5, 6, 7

[43] Y. Yang and T. M. Hospedales. A unified perspective on multi-domain and multi-task learning. In *ICLR*, 2015. 3, 7

[44] S. Zagoruyko and N. Komodakis. Learning to compare image patches via convolutional neural networks. In *CVPR*, 2015. 2

[45] L. Zhang, T. Xiang, and S. Gong. Learning a deep embedding model for zero-shot learning. In *CVPR*, 2017. 1, 6, 7, 8

[46] Z. Zhang and V. Saligrama. Zero-shot learning via semantic similarity embedding. In *ICCV*, 2015. 3, 7, 8

[47] Z. Zhang and V. Saligrama. Zero-shot learning via joint latent similarity embedding. In *CVPR*, 2016. 7

[48] Z. Zhang and V. Saligrama. Zero-shot recognition via structured prediction. In *ECCV*, 2016. 7