Gaussian Prototypical Networks for Few-Shot Learning on Omniglot

Stanislav Fort
Stanford University
Stanford, CA 94305, USA
sfort1@stanford.edu

Abstract We propose a novel architecture for k-shot classification on the Omniglot dataset. Building on prototypical networks, we extend their architecture to what we call *Gaussian prototypical networks*. Prototypical networks learn a map between images and embedding vectors, and use their clustering for classification. In our model, a part of the encoder output is interpreted as a confidence region estimate about the embedding point, and expressed as a Gaussian covariance matrix. Our network then constructs a direction and class dependent distance metric on the embedding space, using uncertainties of individual data points as weights. We show that Gaussian prototypical networks are a preferred architecture over vanilla prototypical networks with an equivalent number of parameters. We report results consistent with state-of-the-art performance in 1-shot and 5-shot classification both in 5-way and 20-way regime on the Omniglot dataset. We explore artificially down-sampling a fraction of images in the training set, which improves our performance. Our experiments therefore lead us to hypothesize that Gaussian prototypical networks might perform better in less homogeneous, noisier datasets, which are commonplace in real world applications.

Introduction Humans are able to learn to recognize new object categories on a single or small number of examples. This has been demonstrated in a wide range of activities from hand-written character recognition [Lake et al., 2011], and motor control [Braun et al., 2009], to acquisition of high level concepts [Lake et al., 2015]. Replicating this kind of behavior in machines is the motivation for studying few-shot learning.

In general, deep learning models have a very high functional expressivity, and rely on being slowly, iteratively trained in a supervised regime on large amounts of data. An influence of a particular example within the training set is therefore small, as the training is designed to capture the general structure of the dataset. This prevents rapid introduction of new classes after training. In contrast, few-shot learning requires very fast adaptation to new data. In particular, k-shot classification refers to a regime where classes unseen during training must be learned using k labeled examples.

In this paper, we develop a novel architecture we refer to as *Gaussian prototypical network*. We build on prototypical networks from Snell et al. [2017], and matching networks from Vinyals et al. [2016]. Our contribution lies in allowing the network to express, handle and utilize confidence estimates in individual data points, and therefore to deal with uncertainty. We show that our architecture is a preferred way of using additional trainable parameters compared to vanilla prototypical networks. We reach, to our knowledge, results consistent with state-of-the-art on the Omniglot dataset [Lake et al., 2015]. We show that partially down-sampling our training set improves performance, as the network learns to utilize confidence estimates better. We therefore hypothesize that Gaussian prototypical networks might perform better in less homogeneous, noisier datasets. Such applications might include facial recognition of a previously unknown person.

Methods Prototypical networks map images into embedding vectors in a high-dimensional space, and use their clustering for classification. Our model maps an image into an embedding vector, and an estimate of the image quality. Together with the embedding vector, a confidence region around it is predicted, characterized by a Gaussian covariance matrix.

Second workshop on Bayesian Deep Learning (NIPS 2017), Long Beach, CA, USA.

We use a multi-layer convolutional neural network without an explicit, final fully connected layer. The output of the encoder is a concatenation of an embedding vector $\vec{x} \in \mathbb{R}^D$ and the relevant components of the covariance matrix $\Sigma \in \mathbb{R}^{D \times D}$. Therefore

$$\mathrm{encoder}_{\mathrm{Gauss}}(W) : I \in \mathbb{R}^{H \times W \times C} \rightarrow [\vec{x}, \vec{s}_{\mathrm{raw}}] \in \left[\mathbb{R}^D, \mathbb{R}^{D_S}\right] , \tag{1}$$

where D_S is the dimensionality of the predicted components of the covariance matrix, and D is the dimensionality of the embedding space. We explore two variants of the Gaussian prototypical network: a) *radius*, in which a single number is predicted per image, and b) *diagonal*, where the diagonal of the covariance matrix is predicted. The encoder consists of 4 convolutional blocks described in Eq. 2 stacked together.

$$3 \times 3\,\mathrm{CNN} \rightarrow \mathrm{batch\,normalization} \rightarrow \mathrm{ReLU} \rightarrow 2 \times 2\,\mathrm{max\,pool} \tag{2}$$

Our best-performing models have 128, 256, 512, and 512 3×3 filters in the 4 blocks respectively, with the embedding space dimensionality of 512 for the radius case, and 256 for the diagonal case.

We explore three variants of the Gaussian prototypical network:

a) **Radius** covariance estimate. $D_S = 1$ and only a single real number $s_{\mathrm{raw}} \in \mathbb{R}^1$ is generated per image. As such the covariance matrix has the form $\Sigma = \mathrm{diag}\,(\sigma, \sigma, \ldots, \sigma)$, where σ is calculated from the raw encoder output s_{raw}. The confidence estimate is therefore not direction-sensitive.

b) **Diagonal** covariance estimate. $D_S = D$ and the dimension of the covariance estimate is the same as of the embedding space. $\vec{s}_{\mathrm{raw}} \in \mathbb{R}^D$ is generated per image. Therefore the covariance matrix has the form $\Sigma = \mathrm{diag}\,(\vec{\sigma})$, where $\vec{\sigma}$ is calculated from the raw encoder output $\vec{s}_{\mathrm{raw}}$. This allows the network to express direction-dependent confidence about a data point.

c) **Full** covariance estimate. A full covariance matrix is output per data point. This method proved to be needlessly complex for the tasks given and therefore was not explored further.

Table 1: The results of our experiments as compared to other papers. To our knowledge, our models perform consistently with state-of-the-art in 1-shot and 5-shot classification both in 5-way and 20-way regime on the Omniglot dataset.

	20-way		5-way	
Model	1-shot	5-shot	1-shot	5-shot
Matching networks [Vinyals et al., 2016]	93.8%	98.5%	98.1%	98.9%
Matching networks [Vinyals et al., 2016]	93.5 %	98.7 %	97.9%	98.7%
Neural statistician [Edwards and Storkey, 2016]	93.2 %	98.1 %	98.1%	99.5%
Prototypical network [Snell et al., 2017]	96.0 %	98.9 %	98.8%	**99.7%**
Finn et al. [2017]			98.7 ± 0.4%	99.9 ± 0.3%
Munkhdalai and Yu [2017]			98.9 %	
TCML [Mishra et al., 2017]	**97.64 ± 0.30%**	**99.36 ± 0.18%**	**98.96 ± 0.20%**	**99.75 ± 0.11%**
Gauss (radius) (ours)	97.02 ± 0.40%	**99.16 ± 0.11%**	**99.02 ± 0.11%**	99.66 ± 0.04%
Gauss (radius) damage (ours)	96.94 ± 0.31%	**99.29 ± 0.09%**	**99.07 ± 0.07%**	**99.73 ± 0.04%**

A key component of the model is the episodic training regime described in Snell et al. [2017]. A subset of classes is chosen from the total number of classes in the training set. For each of these, *support* and *query* examples are chosen at random. The encoded embeddings of the support examples are used to define the class *prototype* – the typical embedding vector for a given class. Confidence estimates are used as weights. A total class covariance matrix is formed by combining covariance matrices of individual data points belonging to it. It is then employed to construct a **direction and class dependent distance metric on the embedding space**, and used to classify *query* images based on proximity to clusters of *support* images. The distance $d_c(i)$ from a class prototype c to a query point i is calculated as

$$d_c^2(i) = (\vec{x}_i - \vec{p}_c)^T S_c \,(\vec{x}_i - \vec{p}_c) \ , \tag{3}$$

where the inverse of class covariance matrix S_c acts as a metric. A diagram of the network function is shown in Figure 1.

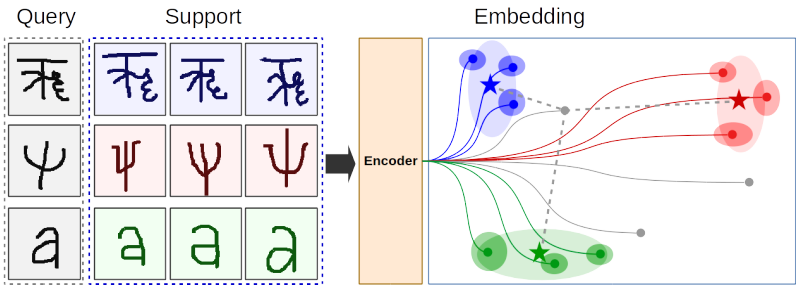

Figure 1: A diagram of the function of the Gaussian prototypical network. An encoder maps an image into a vector in the embedding space (dark circles), and covariance matrix (dark ellipses). Support images are used to define the prototype (stars) and the covariance matrix (light ellipses) of a class. Distances between prototypes and encoded query images (modified by the total covariance of a class) are used to classify them.

Usage of covariance estimate In order to validate our assumption that the Gaussian prototypical network outperforms the vanilla version due to its ability to predict covariances of individual embedded images and therefore the possibility to down-weight them, we studied the distribution of predicted elements of covariance matrices for our best performing network for undamaged and damaged test data. The network was trained on partially down-sampled training data.

For the undamaged test set, the vast majority of covariance estimates took the same value, indicating that the network did not use its ability to down-weight data points. However, for a partially down-sampled test set, the distribution of magnitudes of covariance estimates got significantly broader. We interpret this as a confirmation that the network learned to put less emphasis on down-sampled images. A comparison of both distributions is shown in Figure 2.

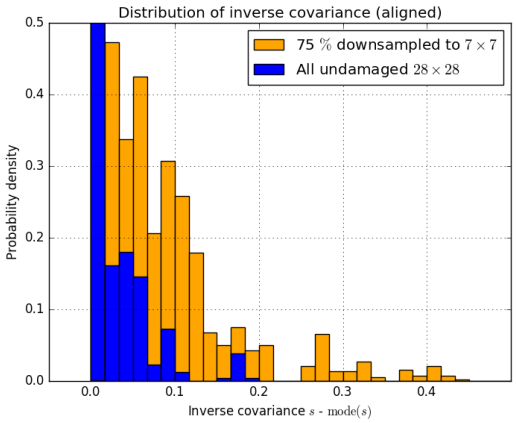

Figure 2: Predicted covariances for the original test set and a partially down-sampled version of it. The Gaussian network learned to down-weight damaged examples by predicting a higher s, as apparent from the heavier tail of the yellow distribution.

Results The comparison of our models to results from literature is presented in Table 1. We report, to our knowledge, results consistent with state-of-the-art in 1-shot and 5-shot classification both in 5-way and 20-way regime on the Omniglot dataset. Especially for 5-way classification, our results are very close to perfect performance. We managed to get better accuracies (in particular for $(k > 1)$-shot classification) by artificially down-sampling fractions of our training dataset, encouraging the network to fully utilize covariance estimates. We hypothesize that the ability to learn the embedding as well as its uncertainty would be even more beneficial for poorer-quality datasets, which are commonplace in real world applications. There, down-weighting some data points might be crucial for faithful classification.

Acknowledgments We would like to thank Ben Poole, Yihui Quek, and Justin Johnson (all at Stanford University) for useful discussions. A part of this work was done as a class project for the Stanford University CS 231N: *Convolutional Neural Networks for Visual Recognition*, which provided Google Cloud credit coupons that partially supported our GPU usage.

References

Brenden M. Lake, Ruslan Salakhutdinov, Jason Gross, and Joshua B. Tenenbaum. One shot learning of simple visual concepts. 2011.

Daniel A. Braun, Ad Aertsen, Daniel M. Wolpert, and C. Mehring. Motor task variation induces structural learning. *Current Biology*, 19(4):352–357, 2009.

B. M. Lake, R. Salakhutdinov, and J. B. Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 2015.

Jake Snell, Kevin Swersky, and Richard S. Zemel. Prototypical networks for few-shot learning. *CoRR*, abs/1703.05175, 2017. URL `http://arxiv.org/abs/1703.05175`.

Oriol Vinyals, Charles Blundell, Tim Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 3630–3638. Curran Associates, Inc., 2016. URL `http://papers.nips.cc/paper/6385-matching-networks-for-one-shot-learning.pdf`.

H. Edwards and A. Storkey. Towards a Neural Statistician. *ArXiv e-prints*, June 2016.

C. Finn, P. Abbeel, and S. Levine. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. *ArXiv e-prints*, March 2017.

T. Munkhdalai and H. Yu. Meta Networks. *ArXiv e-prints*, March 2017.

N. Mishra, M. Rohaninejad, X. Chen, and P. Abbeel. Meta-Learning with Temporal Convolutions. *ArXiv e-prints*, July 2017.