Discriminative k-shot learning using probabilistic models

Matthias Bauer[*,†,‡] **Mateo Rojas-Carulla**[*,†,‡] **Jakub Bartłomiej Świątkowski**[†]
Bernhard Schölkopf[‡] **Richard E. Turner**[†]
[†]Department of Engineering, University of Cambridge, Cambridge, UK
[‡]Max Planck Institute for Intelligent Systems, Tübingen, Germany
[*] MB and MR contributed equally to this work

1 Introduction

K-shot learning has enjoyed a recent resurgence in the academic community [1–5]. Current state-of-the-art methods use complex deep learning architectures and claim that learning good systems for k-shot learning requires *episodic training on simulated data for a specific task and number of shots*. In contrast, this paper proposes a general framework based upon the combination of a deep feature extractor, trained on batch classification, and traditional probabilistic modelling. It subsumes two existing approaches in this vein [5, 6], and is motivated by similar ideas from multi-task learning [7]. We show that even a simple probabilistic model achieves state-of-the-art on a standard k-shot learning dataset by a large margin.

Our basic setup is as follows: a convolutional neural network (CNN) is trained on a large labelled training set. This learns a rich representation of images at the top hidden layer of the CNN. Accumulated knowledge about classes is embodied in the top layer softmax weights of the network. This information is extracted by training a probabilistic model on these weights. K-shot learning can then 1) use the representation of images provided by the CNN as input to a new softmax function, and 2) learn the new softmax weights by combining prior information about their likely form derived from the original dataset with the k-shot likelihood.

2 Probabilistic k-shot learning

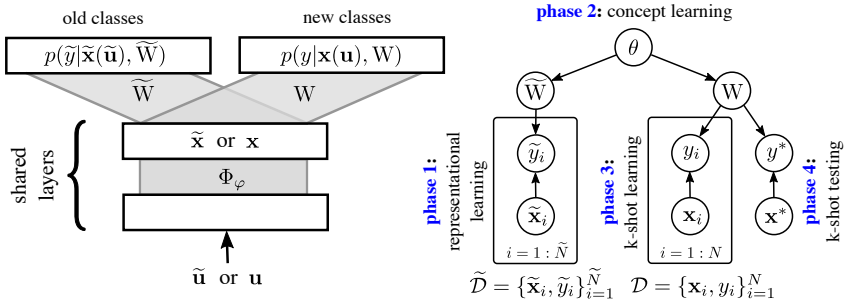

Figure 1: *left:* Shared feature extractor Φ_φ and separate top linear layers W and $\widetilde{\mathrm{W}}$ with corresponding softmax units on old and new classes. *right:* Graphical model for probabilistic k-shot learning.

K-shot learning task. We receive a large dataset $\widetilde{\mathcal{D}} = \{\widetilde{\mathbf{u}}_i, \widetilde{y}_i\}_{i=1}^{\widetilde{N}}$ of images $\widetilde{\mathbf{u}}_i$ and labels $\widetilde{y}_i \in \{1, \dots, \widetilde{C}\}$ and a small dataset $\mathcal{D} = \{\mathbf{u}_i, y_i\}_{i=1}^{N}$ of C new classes, $y_i \in \{\widetilde{C}+1, \widetilde{C}+C\}$, with k images from each new class. Our goal is to construct a model that can leverage the information in $\widetilde{\mathcal{D}}$ and $\mathcal{D}$ to predict well on unseen images $\mathbf{u}^*$ from the new classes; the performance is evaluated against ground truth labels y^*.

Our framework comprises four phases that we refer to as 1) *representational learning*, 2) *concept learning*, 3) *k-shot learning*, and 4) *k-shot testing*, cf. Fig. 1 *(right)*.

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

Feature extractor and representational learning. We use a CNN Φ_φ based on ResNet-34 [8] or VGG [9] as feature extractor. Its last hidden layer activations are mapped to two sets of softmax output units corresponding to the $\widetilde{C}$ classes in the large dataset $\widetilde{\mathcal{D}}$ and the C classes in the small dataset $\mathcal{D}$, respectively. These separate mappings are parametrized by weight matrices $\widetilde{\mathrm{W}}$ for the old and W for the new classes. For *representational learning (phase 1)* the large dataset $\widetilde{\mathcal{D}}$ is used to train the CNN Φ_φ using standard deep learning optimisation approaches. The CNN remains fixed from then on.

Probabilistic modelling. The next goal is to build a probabilistic method for k-shot prediction that transfers structure from the trained softmax weights $\widetilde{\mathrm{W}}$ to the new k-shot softmax weights W and combines it with the k-shot training examples. Given a test image's feature representation $\mathbf{x}^* = \Phi(\mathbf{u}^*)$ during *k-shot testing (phase 4)*, the prediction for the new label y^* is found by averaging the softmax outputs over the posterior distribution of the softmax weights given the two datasets,

$$p(y^* \,|\, \mathbf{x}^*, \mathcal{D}, \widetilde{\mathcal{D}}) = \int p(y^* \,|\, \mathbf{x}^*, \mathrm{W}) p(\mathrm{W} \,|\, \mathcal{D}, \widetilde{\mathcal{D}}) \mathrm{d}\mathrm{W}. \tag{1}$$

To this end, we consider a general class of probabilistic models in which the two sets of softmax weights are generated from shared hyperparameters θ, so that $p(\widetilde{\mathrm{W}}, \mathrm{W}, \theta) = p(\theta)p(\widetilde{\mathrm{W}}|\theta)p(\mathrm{W}|\theta)$, see Fig. 1 *(right)*. In this way, the large dataset $\widetilde{\mathcal{D}}$ contains information about θ that in turn constrains the new softmax weights W. We further assume that there is very little uncertainty in $\widetilde{\mathrm{W}}$ once the large initial training set is observed and so a maximum a posteriori (MAP) estimate, as returned by standard deep learning, suffices. As a consequence of this approximation and the structure of the model, the original data $\widetilde{\mathcal{D}}$ are not required for the k-shot learning phase. Instead, the weights learned from these data, $\widetilde{\mathrm{W}}^{\mathrm{MAP}}$, can themselves be treated as observed data, which induce a predictive distribution over the k-shot weights $p(\mathrm{W}|\widetilde{\mathrm{W}}^{\mathrm{MAP}})$ via Bayes' rule. We refer to this step as *concept learning (phase 2)*.

During *k-shot learning (phase 3)* we treat this predictive distribution as our new prior on the weights and again use Bayes' rule to combine it with the softmax likelihood of the k-shot training examples $\mathcal{D}$ to obtain a new posterior over the weights that now also incorporates $\mathcal{D}$,

$$p(\mathrm{W} \,|\, \mathcal{D}, \widetilde{\mathcal{D}}) \approx p(\mathrm{W} \,|\, \mathcal{D}, \widetilde{\mathrm{W}}^{\mathrm{MAP}}) \propto p(\mathrm{W} \,|\, \widetilde{\mathrm{W}}^{\mathrm{MAP}}) \prod_{n=1}^{N} p(y_n|\mathbf{x}_n, \mathrm{W}). \tag{2}$$

Finally, we approximate Eq. (2) by its MAP estimate $\mathrm{W}^{\mathrm{MAP}}$, so that the integral in Eq. (1) becomes $p(y^* \,|\, \mathbf{x}^*, \mathcal{D}, \widetilde{\mathcal{D}}) \approx p(y^* \,|\, \mathbf{x}^*, \mathcal{D}, \widetilde{\mathrm{W}}^{\mathrm{MAP}}) \approx p(y^* \,|\, \mathbf{x}^*, \mathrm{W}^{\mathrm{MAP}})$.

Our method. We use a simple Gaussian model $p(\mathbf{w}|\theta) = \mathcal{N}(\mathbf{w}|\mu, \Sigma)$ with its conjugate Normal-inverse-Wishart prior $p(\theta) = p(\mu, \Sigma) = \mathcal{NIW}(\mu, \Sigma \,|\, \mu_0, \kappa_0, \Lambda_0, \nu_0)$, and estimate MAP solutions for the parameters $\theta^{\mathrm{MAP}} = \{\mu^{\mathrm{MAP}}, \Sigma^{\mathrm{MAP}}\}$. The approximations discussed lead to $p(\mathrm{W} \,|\, \widetilde{\mathcal{D}}) \approx p(\mathrm{W} \,|\, \widetilde{\mathrm{W}}^{\mathrm{MAP}}) = \mathcal{N}(\mathrm{W} \,|\, \mu^{\mathrm{MAP}}, \Sigma^{\mathrm{MAP}})$.

Relationship to logistic regression. Standard L_2-regularised logistic regression corresponds to the MLE solution of the softmax likelihood $p(y_n \,|\, \mathbf{x}_n, \mathrm{W}) = \mathrm{softmax}(\mathrm{W}\mathbf{x}_n)$ with L_2 penalty on the weights with inverse regularisation strength $1/C_{\mathrm{reg}}$. Its solution corresponds to the MAP solution of a model with isotropic Gaussian prior on the weights with zero mean: $p(\mathrm{W} \,|\, \mathcal{D}) \propto \mathcal{N}(\mathrm{W}|0, \frac{1}{2}C_{\mathrm{reg}}\mathrm{I}) \prod_{n=1}^{N} p(y_n \,|\, \mathbf{x}_n, \mathrm{W})$ and is closely related to the above Gaussian model with MAP inference. However, the probabilistic framework has several advantages: i) modelling assumptions and approximations are made explicit, ii) it is strictly more general and can incorporate non-zero means μ^{MAP}, iii) the probabilistic interpretation provides a principled way of choosing the regularisation constant using the trained weights $\widetilde{\mathrm{W}}$: $C_{\mathrm{reg}} = 2\sigma^2_{\widetilde{\mathrm{W}}}$, where $\sigma^2_{\widetilde{\mathrm{W}}}$ is the empirical variance of the weights $\widetilde{\mathrm{W}}^{\mathrm{MAP}}$. In k-shot learning, alternative (frequentist) methods such as cross-validation suffer in the face of the small number of k-shot examples, and are not applicable in 1-shot learning at all.

3 Experiments

Dataset. *mini*ImageNet has become a standard testbed for k-shot learning and is derived from the ImageNet ILSVRC12 dataset [10] by extracting 100 of the 1000 classes. Each class contains 600 images downscaled to 84×84 pixels. We use the 100 classes (64 train, 16 validation, 20 test) proposed by [11]. As our approach does not require a validation set, we use both the training and validation data for the representational learning.

Method	1-shot	5-shot
ResNet-34 + Isotropic Gauss (ours)	$\mathbf{56.3 \pm 0.4\%}$	$\mathbf{73.9 \pm 0.3\%}$
Matching Networks (reimpl., 1-shot)	$46.8 \pm 0.5\%$	-
Matching Networks (reimpl., 5-shot)	-	$62.7 \pm 0.5\%$
Meta-Learner LSTM [11]	$43.4 \pm 0.8\%$	$60.6 \pm 0.7\%$
Prototypical Nets (1-shot) [4]	$49.4 \pm 0.8\%$	$65.4 \pm 0.7\%$
Prototypical Nets (5-shot) [4]	$45.1 \pm 0.8\%$	$68.2 \pm 0.7\%$

Table 1: Accuracy on 5-way classification on *mini*ImageNet. Our best method, an isotropic Gaussian model using ResNet-34 features, consistently outperforms all competing methods by a wide margin.

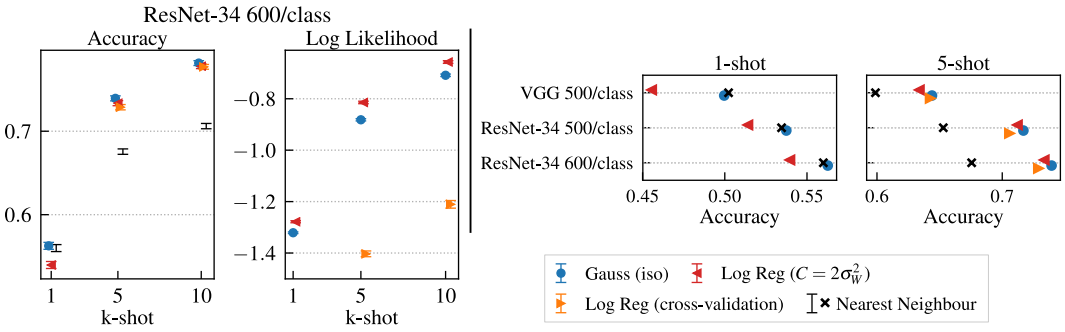

Figure 2: Results on *mini*ImageNet. *(left):* Results using ResNet-34 as feature extractor. *(right):* Comparison of different network architectures and training set sizes on the k-shot learning task.

Overall k-shot performance. We report performance on the *mini*ImageNet dataset in Tab. 1 and Fig. 2. Our best method uses a modified ResNet-34 with 256 features, trained with all 600 examples per training class, as feature extractor and a simple isotropic Gaussian model on the weights for concept learning. Despite its simplicity, our method achieves state-of-the-art and beats prototypical networks by a wide margin of about 6%. The baseline methods using the same feature extractor are also state-of-the-art compared to prototypical networks and both logistic regressions show comparable accuracy to our methods except for on 1-shot learning. In terms of log-likelihoods, Log Reg ($C = 2\sigma_W^2$) fares slightly better, whereas Log Reg (cv) is much worse.

Deeper features lead to better k-shot learning. We investigate the influence of different feature extractors of increasing complexity in Fig. 2 *(right)*: i) VGG style network (500 train images per class), ii) ResNet-34 (500 per class), and iii) a ResNet-34 (all 600 per class). We find that the complexity of the feature extractor as well as training set size consistently correlate with the accuracy at k-shot time. For instance, on 5-shot, Gauss (iso) achieves a significant increase of almost 10%. Importantly, this result implies that training specifically for k-shot learning is not necessary for achieving high generalisation performance on this k-shot problem. On the contrary, training a powerful deep feature extractor on batch classification using all of the available training data, then building a simple probabilistic model using the learned features and weights achieves state-of-the-art. While not presented in this paper, the obtained classifiers are well calibrated at k-shot time, and generalise well to an online setting.

4 Conclusion

We present a probabilistic framework for k-shot learning that exploits the powerful features and class information learned by a neural network on a large training dataset. Probabilistic models are then used to transfer information in the network weights to new classes. Experiments on *mini*ImageNet using a simple Gaussian model within our framework achieve state-of-the-art for 1-shot and 5-shot learning by a wide margin and, at the same time, return well calibrated predictions. This finding is contrary to the current belief that episodic training is necessary to learn good k-shot features and puts the success of recent complex deep learning approaches to k-shot learning into context. The new approach is flexible and extensible, being applicable to general discriminative models and k-shot learning paradigms. The Gaussian model is closely related to regularised logistic regression, but provides a principled and fully automatic way to regularise. This is particularly important in k-shot learning, as it is a low-data regime, in which cross-validation performs poorly and where it is important to train on all available data, rather than using validation sets.

5 Acknowledgements

We thank Antreas Antoniou and Amos Storkey for their implementation of Matching Networks. MB acknowledges funding by a Qualcomm European Scholarship in Technology. RET thanks Google and EPSRC grants EP/M0269571 & EP/L000776/1.

References

[1] Brenden Lake, Ruslan Salakhutdinov, and Joshua Tenenbaum. "Human-level concept learning through probabilistic program induction". In: *Science* 350.6266 (2015).

[2] Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. "Siamese neural networks for one-shot image recognition". In: *Deep Learning workshop, International Conference of Machine Learning* (2015).

[3] Oriol Vinyals, Charles Blundell, Tim Lillicrap, Daan Wierstra, et al. "Matching networks for one shot learning". In: *Advances in Neural Information Processing Systems*. 2016.

[4] Jake Snell, Kevin Swersky, and Richard Zemel. "Prototypical Networks for Few-shot Learning". In: *arXiv e-print: 1703.05175* (2017).

[5] Nitish Srivastava and Ruslan R Salakhutdinov. "Discriminative transfer learning with tree-based priors". In: *Advances in Neural Information Processing Systems*. 2013.

[6] Jordan Burgess, James Robert Lloyd, and Zoubin Ghahramani. "One-Shot Learning in Discriminative Neural Networks". In: *NIPS Bayesian Deep Learning workshop* (2016).

[7] Bart Bakker and Tom Heskes. "Task Clustering and Gating for Bayesian Multitask Learning". In: *Journal of Machine Learning Research* 4 (2003).

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep Residual Learning for Image Recognition". In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016. eprint: 1512.03385.

[9] Karen Simonyan and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition". In: *arXiv e-print:1409.1556* (2014).

[10] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. "ImageNet Large Scale Visual Recognition Challenge". In: *International Journal of Computer Vision (IJCV)* 115.3 (2015). DOI: 10.1007/s11263-015-0816-y.

[11] Sachin Ravi and Hugo Larochelle. "Optimization as a model for few-shot learning". In: *International Conference on Learning Representations*. Vol. 1. 2. 2017.