Fine-tuning Deep Neural Networks in Continuous Learning Scenarios - Supplementary Material -

Christoph Käding[1,2], Erik Rodner[1,2], Alexander Freytag[1,2], and Joachim Denzler[1,2]

[1]Computer Vision Group, Friedrich Schiller University Jena, Germany
[2]Michael Stifel Center Jena, Germany

Abstract. This document contains additional empirical analyses for the paper "Fine-tuning Deep Neural Networks in Continuous Learning Scenarios". The provided information is not necessary to understand the main paper. Although some of the evaluated parameter effects are well known, we present our evaluations for completeness of our study. The following aspects are covered: (i) an additional evaluation of continuous fine-tuning with updates of samples from known classes only (Section S1), (ii) the empirical effect of varying learning rates on continuous fine-tuning (Section S2), (iii) the empirical effect of different batch sizes for SGD steps on continuous fine-tuning (Section S3), (iv) the effect of continuous fine-tuning for different numbers of layers (Section S4), (v) continuous learning of upper layers with varying update influence (Section S5), (vi) a detailed analysis of the influence of sample weights for updates (Section S6), (vii) continuous learning with noisy labels (Section S7), and (viii) further visualizations of shifts for attention regions (Section S8).

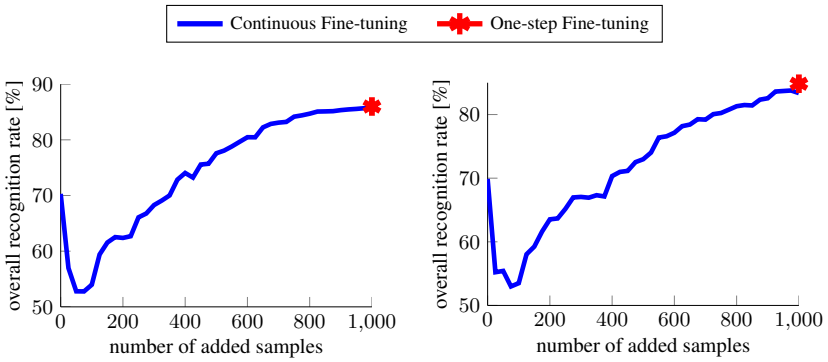
S1 Additional Experiments for Fine-tuning with Known Classes

Section 4 of the main paper introduces three fine-tuning scenarios. While the experimental evaluation of the main paper focuses on the more general case (C2), we want to additionally present an evaluation for case (C1). In scenario (C2) additional training samples of known as well as new classes are added continuously to an initial training set. In contrast to this, scenario (C1) only considers additional samples of known classes.

To evaluate continuous fine-tuning in this setting, we initially trained a CNN with ten classes and three initial known samples each using the MS-COCO dataset [1]. As additional training data 100 random samples per class are added in batches of 25 samples. We perform 10 epochs in every update step and average performance on the corresponding test set over 9 runs. The remaining parameters are similar to the ones given in Section 5.1 and 5.2.

Corresponding results for fine-tuning of only the last two fully connected layers as well as all layers using a BVLC AlexNet [2] are shown in Fig. S1. It can be seen that

This research was supported by grant DE 735/10-1 of the German Research Foundation (DFG).



(a) last two layers (b) all layers

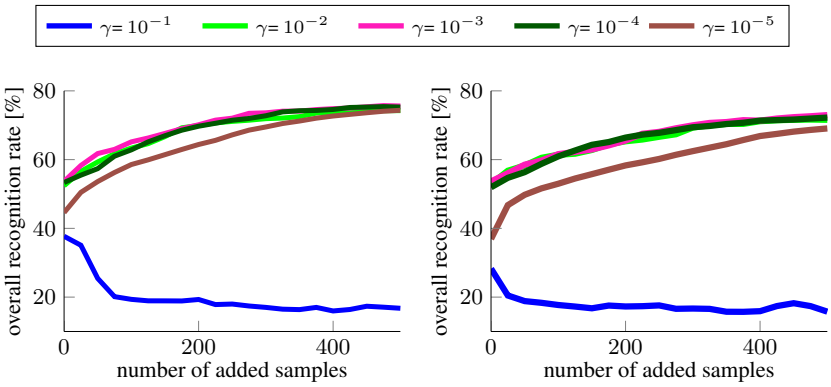
Fig. S1: Obtained accuracy by continuously updating with samples of known classes.

the obtained accuracy drops clearly after adding the first additional samples in both cases. We attribute this behavior to the imbalance of class samples after the first updates. After more samples are added and the training data become more balanced, the performance rises towards the accuracy obtained by the one-step fine-tuning. We conclude that continuous fine-tuning can also be applied to scenarios where only samples of known classes are added over time. Therefore we mainly study the more interesting scenario (C2) covering also the occurrence of so far unknown classes.

S2 Comparison of Different Learning Rates

It is well-known that choosing an appropriate learning rate γ for SGD optimization is crucial for successful learning, *e.g.,* as evaluated by Wilson et al. [3]. If the learning rate is too small, the optimization is unlikely to sufficiently discover the parameter space and can hardly diverge from the initial solution. On the other hand, a large value for learning rates can lead to exploding gradients which prevents the convergence during learning due to numerical issues. Therefore, we also analyzed the effect of different learning rates on the success of continuous fine-tuning. As mentioned in Section 5 of the main submission, we observed no surprising results and excluded them from the main paper. The following analysis closes this gap and presents the results.

We follow the experimental setup of the main paper as described in Section 5.1. Continuous fine-tuning is done for the last two fully-connected layers with an SGD batch size of $|\mathcal{S}| = 64$. The results with different values for the learning rate γ are shown in Fig. S2. As can be seen, the results are clearly in line with previous findings for standard learning or one-step fine-tuning scenarios [3]: if the learning rate is too small (brown curve), the SGD optimization does not converge to a suitable optimum. Similarly, if the learning rate is too large (blue curve), the SGD optimization does not converge at all. Based on this evaluation, we selected a learning rate of 0.001 for all remaining experiments.



(a) MS-COCO (b) Stanford40Actions

Fig. S2: Comparison of different learning rates γ for continuous fine-tuning.

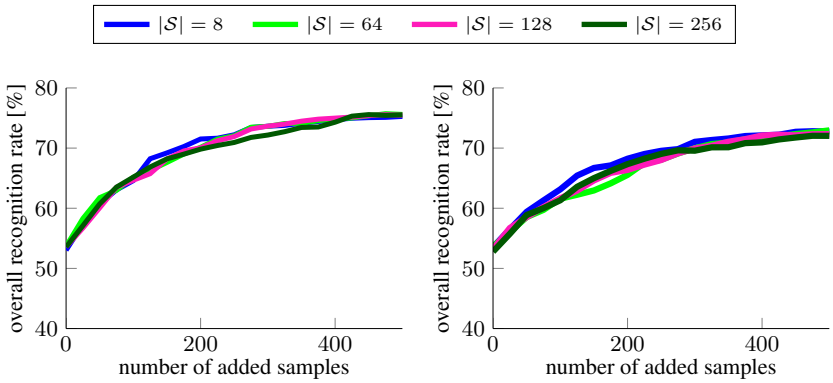
S3 Comparison of Different SGD Mini-batch Sizes

Besides adjusting the learning rate carefully, the optimization process can further be influenced by choosing an appropriate size $|\mathcal{S}|$ for the mini-batches $\mathcal{S}$. A detailed analysis for the standard training of neural networks was already presented in [3]. Hence, we were interested in investigating the impact of different batch sizes for continuous fine-tuning. As for the previous experiment, we excluded our findings from the main submission due to the lack of space and the unsurprisingness of results. Instead, we present our findings in the following.

For this analysis, we follow the previous setup and fix all parameters except the mini-batches' size $|\mathcal{S}|$. As before, we present experiments when the last two fully-connected layers are continuously fine-tuned. Results are shown in Fig. S3. As can be seen, the choice of the batch size has no significant impact on the success of continuous fine-tuning. Therefore, we use a mini-batch size of $|\mathcal{S}| = 64$ as default choice for all remaining experiments. An exception is the evaluation of continuous fine-tuning with a fixed number of SGD update steps (Fig. 4 and Fig. 5 in the main submission) where each batch is comprised of only $|\mathcal{S}| = 25$ examples.

S4 Comparison of Continuous Fine-tuning for Different Layers

In all previously presented evaluations (in the main submission as well as in this supplementary material document), we either investigated continuous fine-tuning for all layers of a given network or only for the last two layers. These settings were inspired by the common practice for one-step fine-tuning, where either all layers are adapted (if enough data is available) or weights of early layers are fixed and only the final classifier-related layers are adapted. Nonetheless, continuous fine-tuning applies to the general settings of adapting an arbitrary subset of parameters given novel data. In the next experiment, we evaluate continuous fine-tuning when parameters of more and more layers are adapted.



(a) MS-COCO (b) Stanford40Actions

Fig. S3: Impact of SGD batch size $|\mathcal{S}|$ on continuous fine-tuning.

For the evaluation, we keep the experimental setup of the previous sections and only change the number of frozen layers (*i.e.,* layers where parameter values are fixed during the optimization). Results are shown in Fig. S4. To our surprise, it can clearly be seen that continuous fine-tuning is possible for all investigated settings and obtained accuracies hardly differ. Hence, we conclude that our presented results for either learning the last two or all layers can be safely transferred to scenarios where different layer sets are adapted.

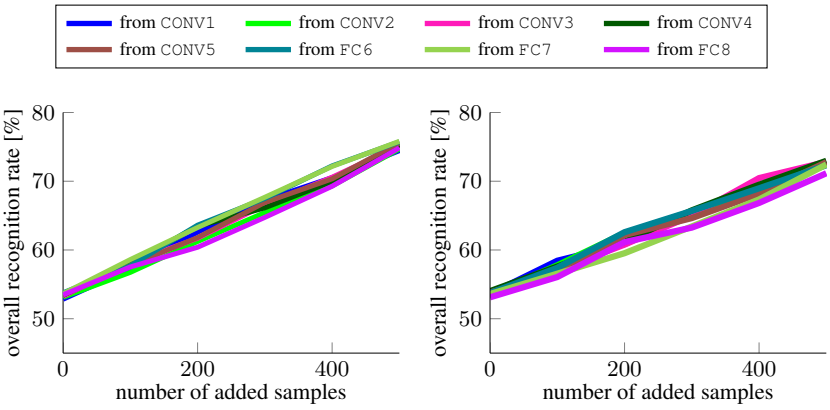
S5 Experiments of Section 5.4 with only two Learnable Layers

In Section 5.4 of the main submission, we investigated how continuous fine-tuning with fixed SGD batch sizes can be improved by different sampling priorities for mini-batch sampling. To this end, we varied the weighting parameter λ which resembles the trade-off between preferring known or novel data during sampling. The presented experiments were obtained for the scenario where parameters of all layers are continuously fine-tuned. For the sake of completeness, we present in the following the same evaluation but adapt only parameters of the last two fully-connected layers.

We keep the experimental setup identical to the one in Section 5.4. Results are shown in Fig. S5. In can be seen that the obtained results are similar to those of the main submission (compare against Fig. 4). This further underlines our conclusion that results can be transferred to continuous fine-tuning scenarios with different numbers of learnable layers.

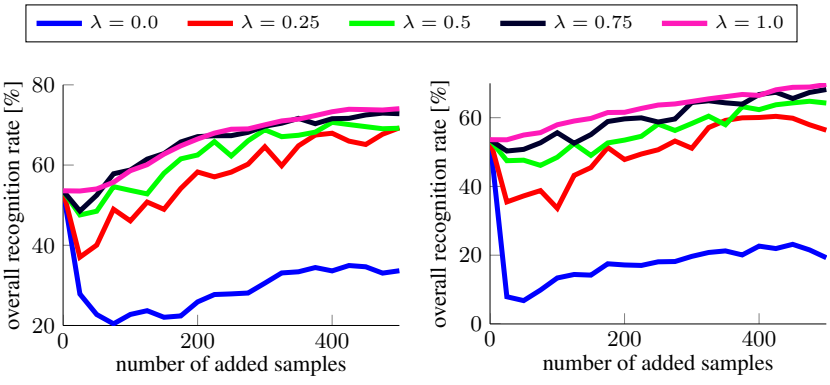
S6 Experiments of Section 5.4 with Detailed Accuracy Analysis

In the previous evaluation and the results presented in Section 5.4 of the main paper, we investigated continuous fine-tuning with fixed SGD batch sizes and priority sampling for mini-batch selection. From the results, we clearly observed that sampling purely



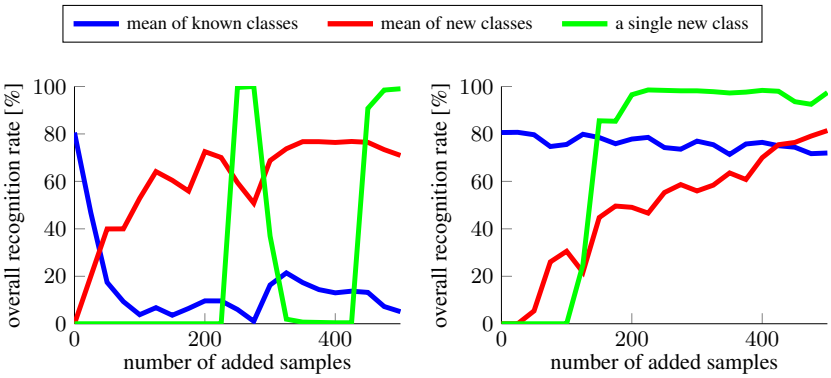
(a) MS-COCO (b) Stanford40Actions

Fig. S4: Impact of number of layers to be fine-tuned on continuous fine-tuning.



(a) MS-COCO (b) Stanford40Actions

Fig. S5: Continuous fine-tuning of all layers with different choices of λ.



(a) $\lambda = 0.0$ (b) $\lambda = 1.0$

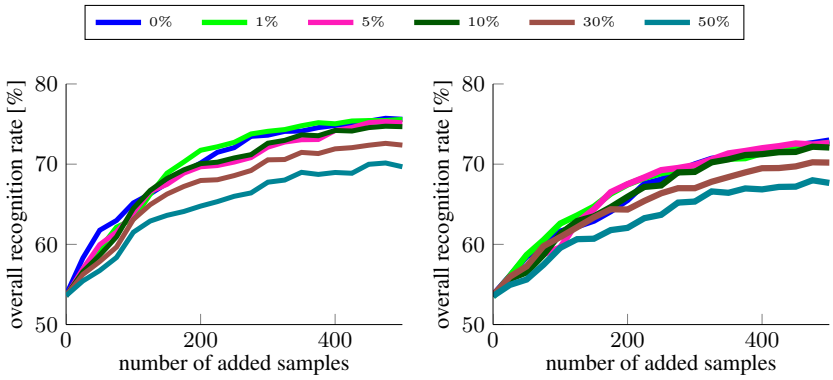
Fig. S6: Overall accuracy separately evaluated for known and novel categories under different settings for the weighting parameter λ in mini-batch sampling. Blue curve shows the overall accuracy averaged over all test examples from known categories. Similarly, red and green curves correspond to overall accuracies evaluated on examples of all novel categories or a single novel category, respectively.

novel data (*i.e.,* $\lambda = 0.0$) severely decreased the overall accuracy. In the following, we further investigate why this happened.

Our hypothesis was that running SGD with data solely from the novel category leads to dramatic overfitting even when only few SGD steps are computed. In contrast, we expect that sampling additionally known data serves as regularizer. To test our hypothesis, we evaluate the models learned in Section 5.4 for $\lambda = 0.0$ and $\lambda = 1.0$ separately on examples of known categories and novel categories. Results for a single run on the MS-COCO dataset are shown in Fig. S6. As can be seen, sampling only novel data for mini-batches (*i.e.,* $\lambda = 0.0$) leads to a drastic decrease of accuracy for known classes (blue curve). While the mean accuracy with respect to novel data increases (red curve), the accuracy of individual novel categories undergoes drastic changes over time (green curve shows results for a single novel category). In contrast to that, for $\lambda = 1.0$, the accuracy rises for new classes (red curve) and keeps nearly constant for the already known ones (blue curve). For a single novel category (green curve), a high accuracy is obtained as soon as examples of this class are added. We conclude that known data prevent the network from overfitting to newly added data during continuous fine-tuning.

S7 Experiments of Section 5.5 with only two Learnable Layers

In contrast to controlled academic environments, real world learning scenarios are often faced with label noise, *i.e.,* non-perfect data annotations. A common example is active learning with unreliable annotators, *e.g.,* if annotators are uncertain, lack knowledge, or simply do mistakes. Therefore, we analyzed the effect of label noise on continuous fine-tuning in Section 5.5 of the main paper. The presented experiments were obtained



(a) MS-COCO (b) Stanford40Actions

Fig. S7: Impact of label noise in update sets on the accuracy obtained by continuous fine-tuning.

for the scenario where parameters of all layers are continuously fine-tuned. We now present the same evaluation but adapt parameters of the last two fully-connected while keeping remaining parameters fixed. Results are shown in Fig. S7. As can be seen, the results are comparable to those of the main submission (compare against Fig. 6 in the main paper). Again, we observe that small amounts of noise do not strongly harm the accuracy obtained by continuous fine-tuning of CNNs.

S8 Additional Visualizations for Section 6

In addition to Section 6 of the main paper, we show further visualizations to investigate the network's changes during continuous fine-tuning. The experimental setup as well as the technique for computing the network's region of attention are kept as described in Section 6 of the main submission. Here, we present more examples which we obtained by visualizing the attention shift of a single filter from the `CONV5` layer. Visualizations with intuitive region shifts towards the action-related objects are shown in Fig. S8. Examples with no region shifts are shown in Fig. S9. Finally, Fig. S10 shows examples where the attention regions shifted towards contextual related areas.

Fig. S8: Visualization of attention shift of a single filter of the CONV5 layer during continuous fine-tuning with categories from the Stanford40Actions dataset. We show examples where the attention region of the network before the respective category became known (magenta box) shifted *towards the action-related objects* after the availability of the category (cyan box).

Fig. S9: Visualization of attention shift of a single filter of the CONV5 layer during continuous fine-tuning with categories from the Stanford40Actions dataset. We show examples where the attention region of the network before the respective category became known (magenta box) *did not move significantly* after the availability of the category (cyan box).

Fig. S10: Visualization of attention shift of a single filter of the CONV5 layer during continuous fine-tuning with categories from the Stanford40Actions dataset. We show examples where the attention region of the network before the respective category became known (magenta box) shifted *towards contextual related areas* after the availability of the category (cyan box).

References

1. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: ECCV. (2014) 740–755
2. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: NIPS. (2012) 1097–1105
3. Wilson, D.R., Martinez, T.R.: The general inefficiency of batch training for gradient descent learning. Neural Networks **16** (2003) 1429–1451