Similarity of Pre-trained and Fine-tuned Representations

Thomas Goerttler [1] Klaus Obermayer [1 2]

Abstract

In transfer learning, only the last part of the networks - the so-called head - is often fine-tuned. Representation similarity analysis shows that the most significant change still occurs in the head even if all weights are updatable. However, recent results from few-shot learning have shown that representation change in the early layers, which are mostly convolutional, is beneficial, especially in the case of cross-domain adaption. In our paper, we find out whether that also holds true for transfer learning. In addition, we analyze the change of representation in transfer learning, both during pre-training and fine-tuning, and find out that pre-trained structure is *unlearned* if not usable.

1. Introduction

Pre-trained weights are often reused when deep learning models are trained. There are several reasons to use weights from a model having been pre-trained on a larger dataset: the training is faster, less time is needed, the computational costs are decreased, and the performance improves, especially when only having a small dataset. For computer vision, using on ImageNet pre-trained weights have become the standard practice, as the weights encode information coming from millions of images in diverse domains.

However, it is still not fully revealed why transfer learning works so well. It is assumed that the first part of the network is only dealing with low-level features (e.g., edges and blobs) and, later on, high-level features (e.g., objects) (Zhuang et al., 2021). Therefore, only fine-tuning the head (which are the fully connected layers at the end) is often enough to adapt to the novel but still similar tasks. This is also an implicit regularization as reducing the parameter also prevents overfitting. (Raghu et al., 2020) discovered that in a few-shot learning problem, a meta-learner trained to adapt fast to a novel task hardly changes the representation of early layers during fine-tuning if the task comes from the same distribution as the training tasks. However, Oh et al. (2021) found out that, especially in the case of cross-domain adaption, where the fine-tuning task does not come from the same distribution as in training, also an adaptation of earlier layers is very beneficial. Neyshabur et al. (2020) investigated what is transferred in transfer learning by shuffling the blocks of inputs. They confirmed that lower layers are responsible for more general features and that a network with pre-trained weights stays in the same basin of solution during fine-tuning.

This paper analyses representation obtained by models having initialized, pre-trained, and fine-tuned weights. We compare their corresponding representation and analyze their similarity when applying them to structured, unstructured, domain, and cross-domain tasks. We find out that structure in the data is encoded in the early convolutional layers. If the transfer task is unstructured, it *unlearns* the information. If the data comes from the same domain or a cross-domain, it exploits it.

2. Similarity Analysis of Representation

To analyze what happens during pre-training and fine-tuning, we compare the representation similarity of different layers of the model. Representation similarity techniques are widely used in computational neuroscience and machine learning. In neuroscience, most often the RSA (representation similarity analysis) is used, e.g., to compare a computational or behavioral model with the brain response (Kriegeskorte et al., 2008). In deep learning, most often centered kernel alignment (CKA) (Kornblith et al., 2019) is used to compare the representation (Raghu et al., 2020; Oh et al., 2021; Kornblith et al., 2021; Neyshabur et al., 2020). RSA and CKA are similar and are instances of a more general approach discussed by Dwivedi et al. (2020). In a first step, representations associated with each pair of inputs - which can be normalized - are compared (e.g., by Euclidean distances or calculating the scalar product). In a second step, the resulting Gram matrices of a layer can be compared to other gram matrices with another similarity measure.

[1] Chair of Neural Information Processing, Technische Universitat Berlin, Germany [2] Bernstein Center for Computational Neuroscience Berlin, Germany. Correspondence to: Thomas Goerttler <thomas.goerttler@tu-berlin.de>.

Workshop of Updatable Machine Learning at ICML 2022, Baltimore, Maryland, USA, PMLR 162, 2022. Copyright 2022 by the author(s).

In our experiments, we use the 4-block-conv architecture which has been proposed by Vinyals et al. (2016) and also used in Oh et al. (2021). The architecture consists of 4 modules with 3x3 convolutions and 64 filters. These are followed by batch normalization, a ReLU activation function, and a pooling layer (2x2). The output of the fourth block is flattened and fully connected with the output layer. We use the stochastic gradient optimizer with a learning rate of 0.001 and a momentum of 0.9.

We apply the network to the two standard datasets, CIFAR-10 and SVHN. Every run is repeated five times (each with a different seed), and the standard error of the results is indicated.

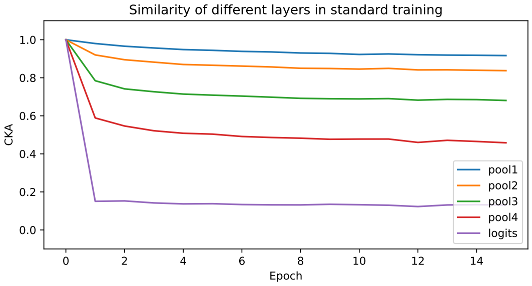


Figure 1. Similarity of the representations of the the 4-block-conv architecture applied on CIFAR-10 during training.

3. Pre-trained Networks

To understand how transfer learning learns, we first need to analyze what is learned in the standard deep learning, which serves in transfer learning as pre-training. This gives a good overview of how networks generally learn and also the option to be directly compared to fine-tuning.

In our first experiment, we look at the representation change of the layers to the random initialization and control how it evolves during training (see Figure 1). Similar to Goerttler & Obermayer (2021), we confirm that the representations of early layers change less than the later ones. This implies that less adaptation in the early layer is common for neural networks. Therefore, one has to be careful to interpret smaller adaptations in early layers directly but instead has to compare them always with another setup.

3.1. Label Generation

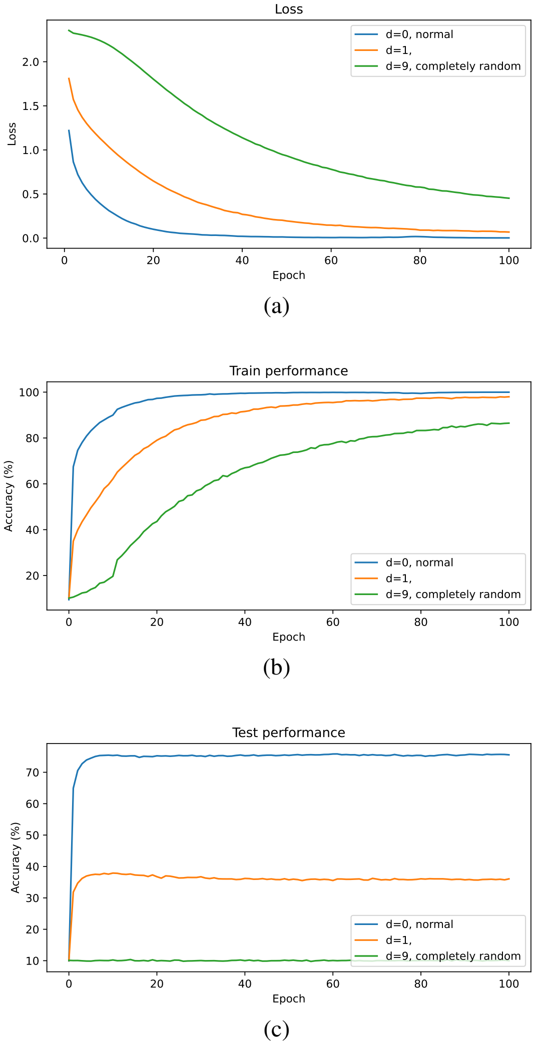
Since the scalar value of the similarity score of CKA is difficult to classify, we propose an experiment on random labels. For that, we also introduce partially random labels. If we have a labeled dataset with D classes which are named with numbers from 0 to $D-1$, we define a partially random label y_d with a degree of randomness $d \in \{0, 1, ..., D-1\}$ as:

$$y_d = (y + Y) \mod D \tag{1}$$

where Y is a random variable with a probability function of

$$Pr(y = Y) = \begin{cases} \dfrac{1}{d+1} & \text{if } y \in \mathbb{N}_0 \text{ and } y \leq d \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

Every original label is uniformly distributed over $d + 1$ labels, such that the closer d is to D, the more random the labels are, and less structure remains in the problem. A choice of $d = 0$ is equivalent to the original dataset, whereas $d = D - 1$ means that the dataset is completely random.



(a)

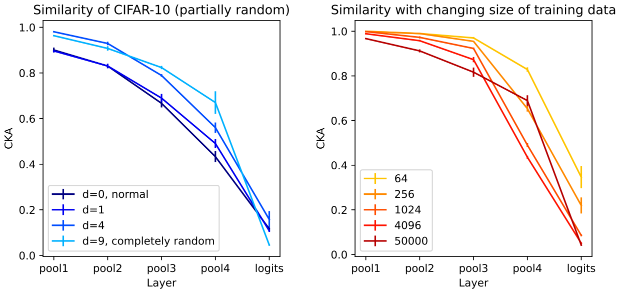
(b)

(c)

Figure 2. This figure depicts the training statistics of experiments on CIFAR-10 with (partially) random labels: in (a) the loss, in (b) the training accuracy and in (c) the test accuracy.

3.2. Decrease of Structure in Training Data

Before we analyze the similarity, we want to look at the training statistics, which are depicted in Figure 2. First of all, in Figure 2(a) be seen that if the data is more random and therefore has less structure, the loss decreases slower. This makes sense, as the overall problem is more difficult, and it has to memorize the labels and cannot take advantage of the shared structure in the data. Nevertheless, if the models are trained long enough, the training accuracy in the experiments on the randomized data is almost as high as when trained on the randomized data (see Figure 2(b)). It is really interesting that the same neural network on the one side can generalize if data is structured, but on the other side, memorize if the data does not allow to learn structure. This even holds when the networks are largely overparameterized (Poggio et al. (2018)). As a sanity check, we depict the test accuracy in Figure 2(c), which of course, only learns something when there is structure. The test accuracy on CIFAR-10 is not matching the state-of-the-art. However, the focus of the paper is not to beat it but to understand representation change in transfer learning.



(a) (b)

Figure 3. Comparision of the similarity of the experiment on CIFAR-10 with (partially) random labels (3(a)). On Figure 3(b) the change of representation can be analyzed with respect to the numbers of sample in training size for random labels.

Looking at the results of Figure 3(a), it can be observed that if the structure in the data decreases, there is less similarity change overall and especially in the early layers. From that, we conclude that memorization is mainly happening in the last layer. However, the clearer the structure is in the training set, the more the earlier layers are used.

We are also interested in how the representations change with the size of the data when training on completely random labels. In Figure 3(b) we see that if there are only a few samples, memorization does not need anything else than the last layer. The more data we have, the more are also earlier layers used.

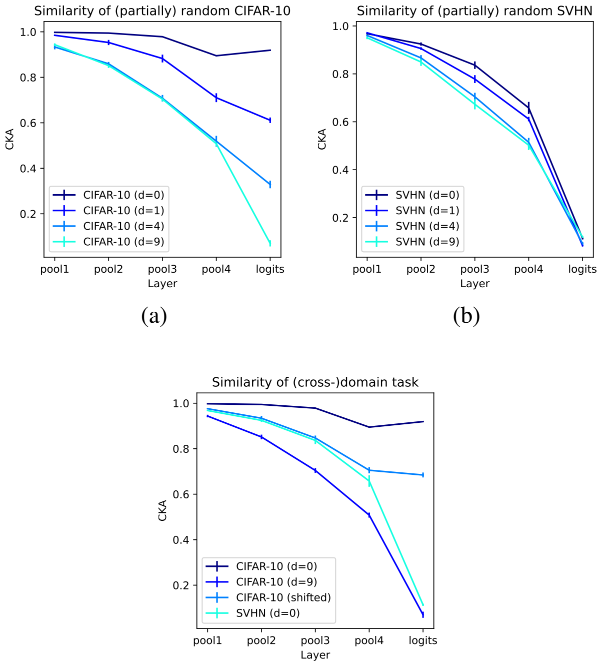
Overall, we can summarize so far that a larger change in early layers coheres with the learned structure.

4. Fine-tuned Networks

Instead of using randomly initialized weights, transfer learning takes advantage of pre-trained weights. In this section, we analyze the representation change of models during fine-tuning. For all the following comparisons, we pre-trained all models on the training set of CIFAR-10 (5 different seeds). We fine-tuned several versions of the CIFAR-10 test data and the SVHN dataset. next to (partially) random versions (see Section 3), we also shifted the labels, such the new labels Y_s are

$$y_S = (y + s) \mod D \qquad (3)$$

where $s \in 1, ..., D-1$. This represents a similar dataset that lies in the same domain, but the novel labels have to be relearned. For all fine-tuned tasks, we used 5000 samples for training and 5000 for testing[1].



(a) (b)

(c)

Figure 4. Representation change of fine-tuned networks, which were initilized by pre-trained weights.

4.1. (Partially) Random Labels

Looking at the results of fine-tuning the (partially) random labels on CIFAR-10 (Figure 4(a)), the opposite of the results in standard deep learning happens (see Section 3.2). The more random the data is, the more the representation changes. It makes sense that for non-random labels, hardly

[1] also in SVHN, although there is, in theory, more data available, we want to be consistent

anything changes in early layers as the data from the problem comes from the same distribution as in pre-training. However, the learned structure is *unlearned* when there is no structure in the dataset because the labels are random. Interestingly the representation from the complete random experiment is higher than when it was trained on random weights. We repeated the experiment also on SVHN and a randomized version of it (Figure 4(b)) to shift the data distribution. But also here, the same is observed, and the more randomness, the more change in representation.

Therefore, we propose that if a model has learned a specific structure of the data, this is only used when the structure is actually needed.

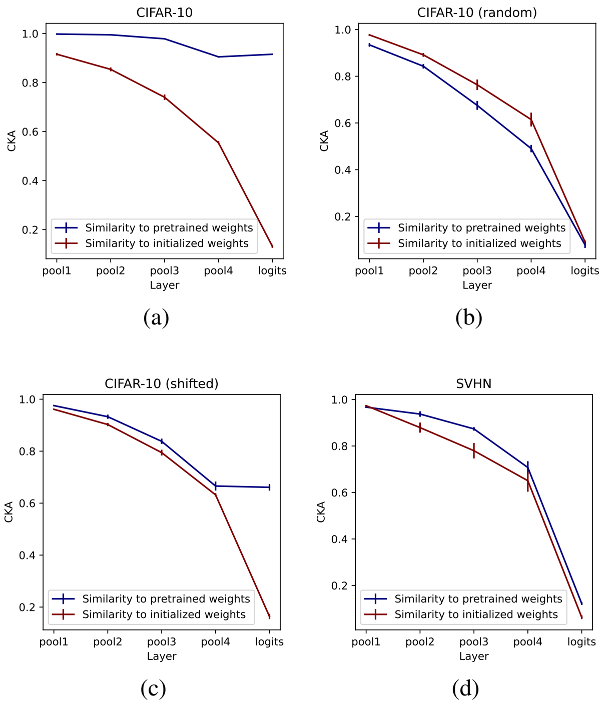
4.2. Domain vs. Cross-domain

In Figure 4(c), the results of fine-tuning on a domain (shifted) and a cross-domain (SVHN) tasks are depicted. The representation change in early layers is larger than when fine-tuned on CIFAR-10 but significantly less than when fine-tuned on random labels. This indicates that the models make use of pre-trained weights. Compared to domain adaption, a cross-domain adaptation relies more on change in the penultimate layer, which has already been mentioned by Oh et al. (2021).

4.3. Pre-trained vs Pre-initilized

Our models are trained in two phases. In the first phase, it transforms the random weights into pre-trained ones, and in the second, it turns the pre-trained weights into fine-tuned ones. In these experiments, we compare the representation change of the fine-tuned weights to both the random initialization and the representation obtained by the pre-trained weights.

In Figure 5(a), we can observe that for CIFAR-10, the fine-tuned representations are closer to the pre-trained than to the pre-initialized one. This makes sense, as the task during pre-training was on data coming from the same distribution, so very similar. Interestingly, when having complete random labels, this is different (see Figure 5(b)). The representations obtained after applying the final model to the data are more similar to the complete random initialization than to the pre-trained weights on which the fine-tune task started. This confirms our claim that if the data is not meaningful and structured directly, it *unlearns* learned structure, as this is not helpful in memorization. However, when the model has learned the structure, it helps to fine-tune and is strongly used when applied to a task that can take advantage of it. In Figure 5(c) and Figure 5(d), we see that the fine-tune task exploits the pre-trained weights when applied not on almost the same, but on a domain, respectively, a cross-domain task. This is the case because the final representation is more similar to the pre-trained one than to the initial one, although the differences between the two similarities are not as far away as when the task is almost the same. From the results on random labels (Figure 5(b)), we know that this is not automatically the case. The tasks have to share some underlying structure, which goes beyond the similar input in the input layers. Even if the input is the same, random weights destructure the dataset such that pre-trained encoded information is not useful.



(a)

(b)

(c)

(d)

Figure 5. Comparision of the representation change with respect to the representations of initialized and pre-trained weights.

5. Conclusion

In this paper, we did several experiments to understand how transfer learning takes advantage of pre-trained weights. We used CKA to analyze the representation and reveal the representation change during pre-training and fine-tuning. We found out that memorization happens mainly in the last layer, if possible. Early layers do not change. However, if memorization is learned on a pre-trained model, it *unlearns* the structure and changes the early representation. Domain and cross-domain tasks exploit pre-trained representation and take advantage of it. When solving cross-domain tasks, the earlier layers have to change more than when solving domain tasks to adapt better to the new input data.

References

Dwivedi, K., Huang, J., Cichy, R. M., and Roig, G. Duality diagram similarity: A generic framework for initialization selection in task transfer learning. In Vedaldi, A., Bischof, H., Brox, T., and Frahm, J. (eds.), *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XXVI*, volume 12371 of *Lecture Notes in Computer Science*, pp. 497–513. Springer, 2020.

Goerttler, T. and Obermayer, K. Exploring the similarity of representations in model-agnostic meta-learning. *Learning-To-Learn Workshop at the 9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*, 2021.

Kornblith, S., Norouzi, M., Lee, H., and Hinton, G. E. Similarity of neural network representations revisited. In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pp. 3519–3529. PMLR, 2019.

Kornblith, S., Chen, T., Lee, H., and Norouzi, M. Why do better loss functions lead to less transferable features? In Ranzato, M., Beygelzimer, A., Dauphin, Y. N., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 28648–28662, 2021.

Kriegeskorte, N., Mur, M., and Bandettini, P. A. Representational similarity analysis-connecting the branches of systems neuroscience. *Frontiers in Systems Neuroscience*, 2:4, 2008.

Neyshabur, B., Sedghi, H., and Zhang, C. What is being transferred in transfer learning? In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

Oh, J., Yoo, H., Kim, C., and Yun, S. BOIL: towards representation change for few-shot learning. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.

Poggio, T. A., Kawaguchi, K., Liao, Q., Miranda, B., Rosasco, L., Boix, X., Hidary, J., and Mhaskar, H. N. Theory of deep learning III: explaining the non-overfitting puzzle. *CoRR*, abs/1801.00173, 2018.

Raghu, A., Raghu, M., Bengio, S., and Vinyals, O. Rapid learning or feature reuse? towards understanding the effectiveness of MAML. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*, 2020.

Vinyals, O., Blundell, C., Lillicrap, T., Kavukcuoglu, K., and Wierstra, D. Matching networks for one shot learning. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pp. 3630–3638, 2016.

Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., Xiong, H., and He, Q. A comprehensive survey on transfer learning. *Proc. IEEE*, 109(1):43–76, 2021.