Layer-Wise Contrastive Unsupervised Representation Learning

Stephen Zhao
stephen.zhao@mail.utoronto.ca

1 Introduction

Existing machine learning models typically require large amounts of labeled data, which can be costly to obtain. Feature representations learned in an unsupervised manner let us leverage existing unlabeled data, with immediate applications in transfer learning, such as weight initialization for faster learning, or better performance if labeled data for the downstream task is scarce. Representations learned in an unsupervised manner are also likely better than those from a supervised task if the downstream task focuses on features irrelevant to the upstream supervised training task. In this work, we focus on the set of unsupervised learning approaches that Arora et al. (2019) [1] call "contrastive unsupervised representation learning". We use a layer-wise adaptation, starting within the context of images. We train feature representations of semantically similar images (i.e. nearby patches within the same image) to be closer than that of unrelated images (e.g. random patches), to learn convolutional neural network filters from unlabeled data.

We hypothesize our layer-wise approach will be faster than end-to-end training, particularly if we only transfer a few layers. Works such as Yosinski et al. (2014) [10] demonstrate that in certain multi-layer neural network architectures, the first few layers learn more "general" features that are most transferable, suggesting that we only need to learn and transfer a few layers. Grinberg et al. (2019) [5] demonstrate that local Hebbian learning of first-layer filters can be almost as good as end-to-end learning, but at a much faster training speed. Motivated by these results, we believe we can achieve similar performance in the transfer learning setting to end-to-end learning but with faster training time.

2 Related Work

Unsupervised Representation Learning Arora et al. (2019) [1] use the term "Contrastive Learning" for methods that contrast pairs of semantically similar text or images with negative (for example, random) samples. In the context of unsupervised representation learning, the objective is for representations of the positive (similar) pair to be closer to each other than the negative pair. Their theoretical framework involves first drawing from a distribution of latent classes (unobserved semantically meaningful categories), and then drawing from an associated distribution of possible inputs (e.g. images) conditional on the latent class. In this setup, the loss in contrastive unsupervised representation learning is an upper bound on the lowest possible loss on a downstream supervised task involving the same set of latent classes. Thus, the feature representations that minimize loss on the unsupervised task are useful for the supervised task.

Contrastive learning with visual data is often done by comparing image patches. Wang & Gupta (2015) [8] work with videos, generating samples by tracking patches of motion. The first and last tracked frames become the positive pair, and can be contrasted with a random patch. Patches are passed through a siamese-triplet network involving both convolutional and fully-connected layers, and loss is based on the cosine distance of final feature representations. Doersch et al. (2015) [3] instead consider two input image patches, a center patch and one of eight possible neighbour patches; the training task is to classify the spatial location of the neighbouring patch.

Dosovitskiy et al. (2014) [4] approach the idea of extracting information from semantically similar image patches by applying a series of transformations (translation, scaling, rotation, contrast) to a sample patch to generate a class. Generated classes then form a batch to which classification algorithms can be applied.

Layer-Wise Learning Compared to standard end-to-end learning, layer-wise learning can provide faster training. Grinberg et al. (2019) [5] use a local Hebbian learning rule for unsupervised learning of the first layer weights. They achieve almost as good results on transfer learning tasks with much less wall-clock training time, concluding that general features, learned independently from lower layers, work well for the first layer.

Transfer Learning Works such as Yosinski et al. (2014) [10] and Lee et al. (2017) [7] demonstrate that, at least in certain convolutional network architectures, the first few layers contain general features that are most useful for transfer, and that transferring a few lower layers is almost as good as transferring all layers for fine-tuning. This suggests that a fast layer-wise training algorithm could achieve additional time savings in transfer learning.

3 Methodology

Implementation Details Given a dataset of images, we sample adjacent patches (overlapping, drawn from a larger patch) and pass them through a single layer of convolutional filters, followed by max-pooling. The resulting embeddings are passed through a triplet loss function, where the contrastive distance is based on either a cosine similarity or Euclidean norm metric. [1]

We found that small patches, such as the minimum 7x7 for a 6x6 filter with 2x2 max-pooling, worked as well for the first layer as larger patch sizes, such as 10x10. This is consistent with an interpretation of our patch sampling method as similar (in expectation) to applying a convolutional layer on a larger patch. Max-pooling is essential for learning filters such as edge detectors, as it provides translation invariance.

Preliminary Results Using CIFAR-100, with labels removed, as the unsupervised learning training set, we learn first-layer 6x6 convolutional filters that look similar to those learned from supervised training (compare Figure 1 and Figure 2 in the appendix). We test these filters in a transfer learning setting, training a simple classifier (2 convolutional layers and 2 fully connected layers) on CIFAR-10, where the first layer filters are the ones learned in the unsupervised setting, and are either 1) frozen or 2) fine-tuned. In the fine-tuning setting, the filters outperform random initialization (Figure 3). It would probably be difficult to beat supervised transfer in this case given that features useful for CIFAR-100 are likely also useful for CIFAR-10.

Next Steps Respectable performance in this simple environment motivates us to look at more complex datasets, as well as deeper, wider, or more modern network architectures. Regarding network architecture, the current state-of-the-art on CIFAR-10 appears to be an AmoebaNet-B (18, 512) architecture achieving 99.0% accuracy [6]. ResNets are also very popular, providing competitive results for example in [9] and [2]. We could start with common architectures used for testing unsupervised representation learning in the past such as AlexNet (e.g. in [8]) and VGG (e.g. in [3], also used by Arora et al. [1]), before moving on to more modern architectures.

Other important next steps include multiple layers of layer-wise training and transfer, as well as comparison with end-to-end unsupervised learning. Additional future directions could include extensions to auxiliary tasks, looking at layer-wise supervised representation learning, extension to a semi-supervised learning setup, and possibly an extension beyond images to more general settings where we can define notions of semantic similarity.

Acknowledgement

Thanks to Silviu Pitis for guidance, advice, and mentorship throughout this project.

References

[1] S. Arora, H. Khandeparkar, M. Khodak, O. Plevrakis, and N. Saunshi. A theoretical analysis of contrastive unsupervised representation learning, 2019.

[2] T. DeVries and G. W. Taylor. Improved regularization of convolutional neural networks with cutout, 2017.

[3] C. Doersch, A. Gupta, and A. A. Efros. Unsupervised visual representation learning by context prediction. *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1422–1430, 2015.

[4] A. Dosovitskiy, P. Fischer, J. T. Springenberg, M. A. Riedmiller, and T. Brox. Discriminative unsupervised feature learning with exemplar convolutional neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38:1734–1747, 2014.

[1] Code is available here: `https://github.com/Silent-Zebra/shallow-conv`

[5] L. Grinberg, J. Hopfield, and D. Krotov. Local unsupervised learning for image analysis. *arXiv preprint arXiv:1908.08993*, 2019.

[6] Y. Huang, Y. Cheng, A. Bapna, O. Firat, M. X. Chen, D. Chen, H. Lee, J. Ngiam, Q. V. Le, Y. Wu, and Z. Chen. Gpipe: Efficient training of giant neural networks using pipeline parallelism, 2018.

[7] J. Y. Lee, F. Dernoncourt, and P. Szolovits. Transfer learning for named-entity recognition with neural networks. *arXiv preprint arXiv:1705.06273*, 2017.

[8] X. Wang and A. Gupta. Unsupervised learning of visual representations using videos. *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 2794–2802, 2015.

[9] Y. Yamada, M. Iwamura, T. Akiba, and K. Kise. Shakedrop regularization for deep residual learning, 2018.

[10] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson. How transferable are features in deep neural networks? In *NIPS*, 2014.

A Appendix: Visualization and Results

Figure 1: Unsupervised Filters

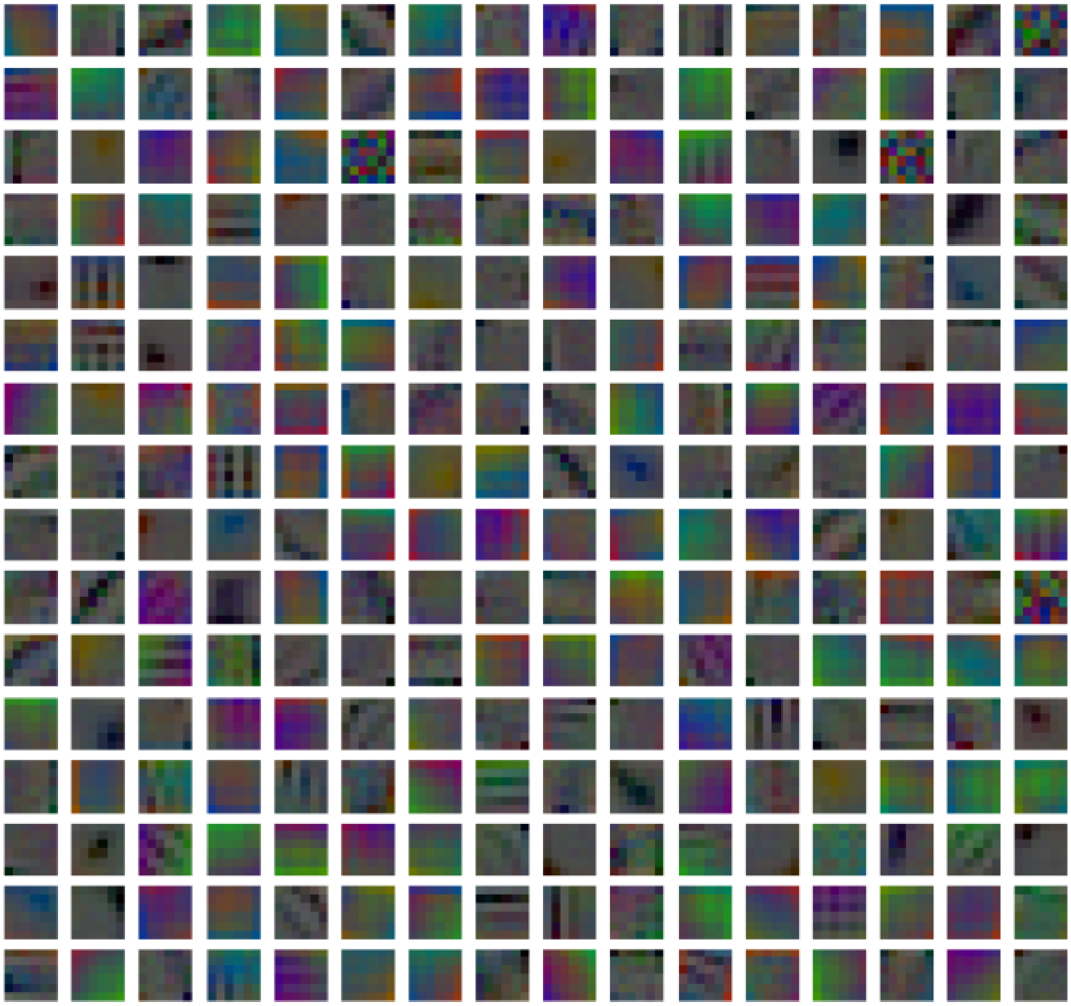


Figure 2: Supervised Filters

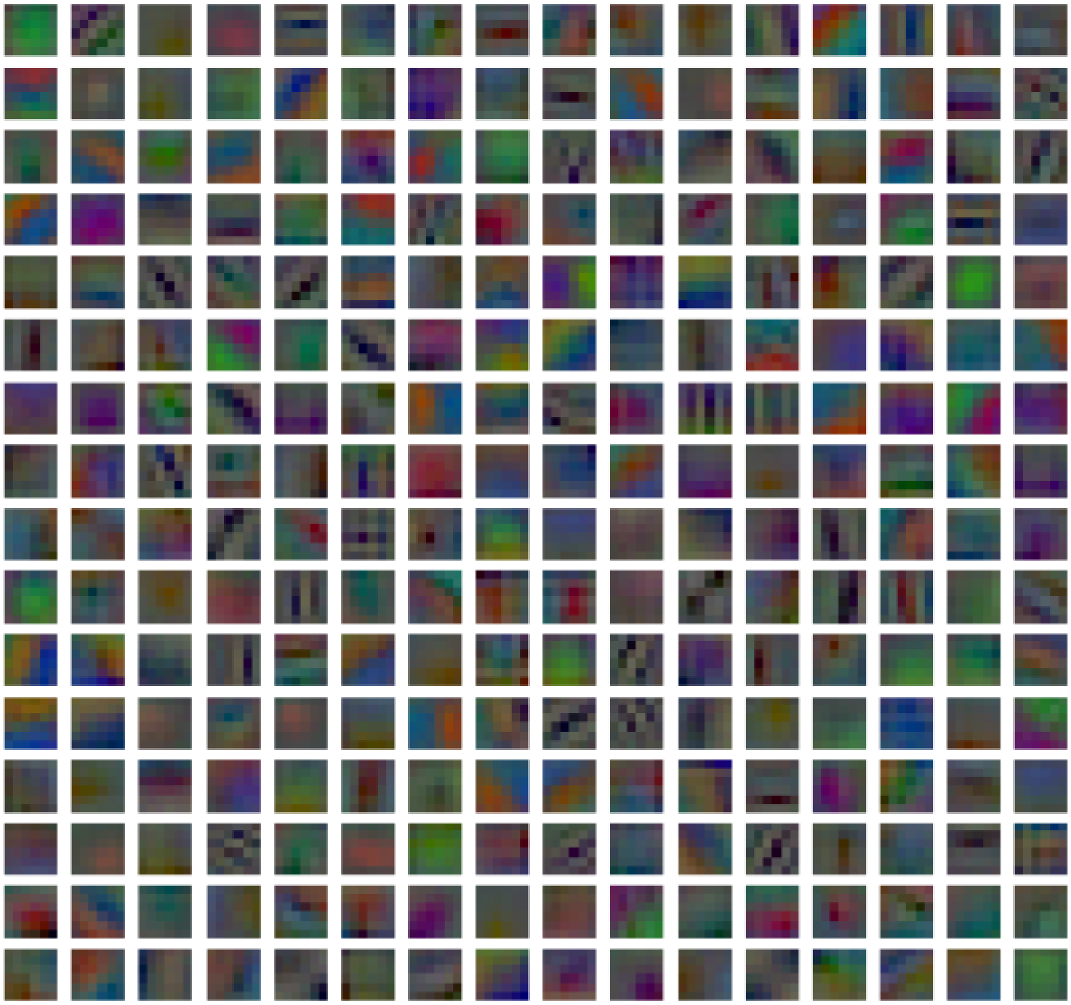
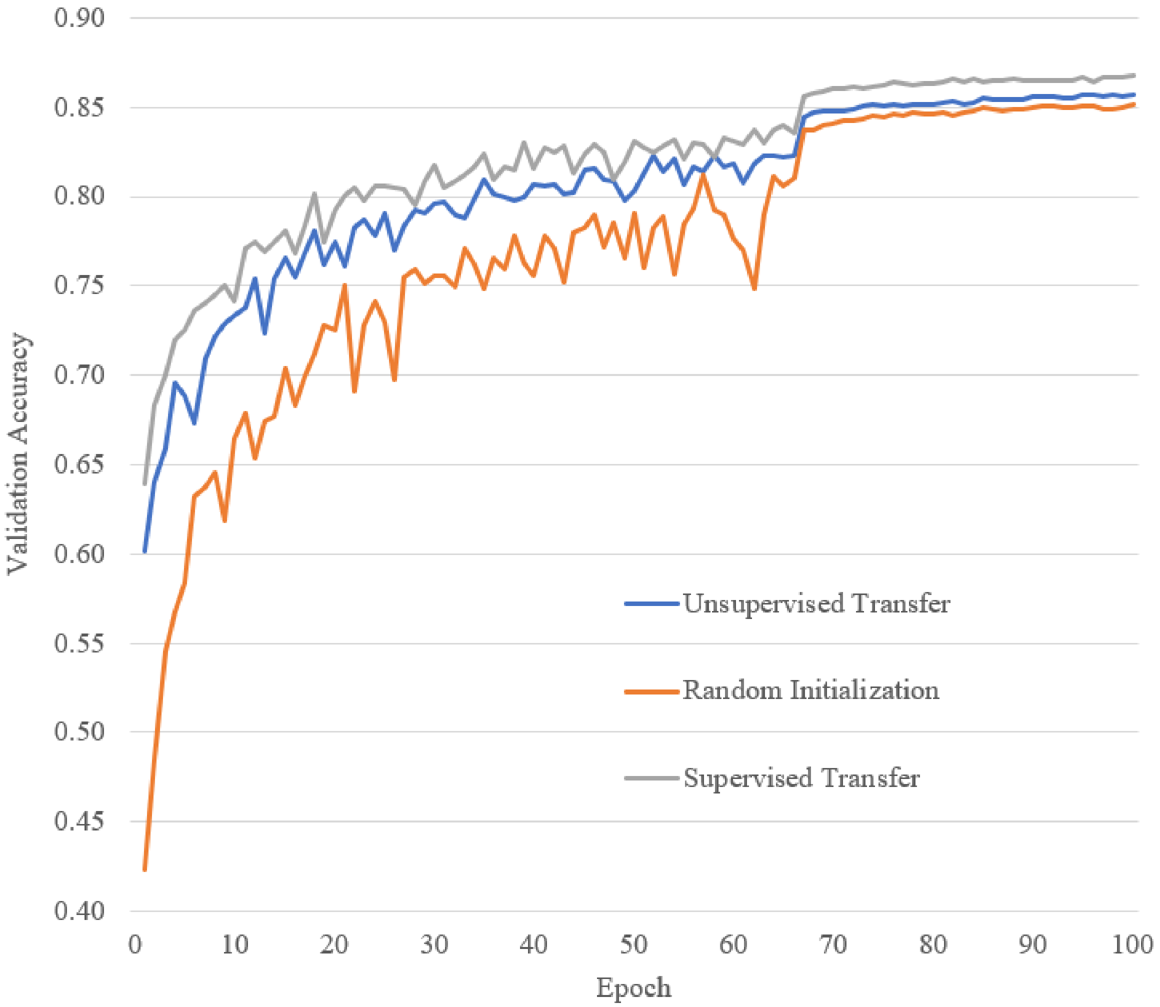


Figure 3: Performance on CIFAR-10. The learning rate is 0.001 for the first two-thirds of the training, and is 0.0001 afterwards. Results are averaged over 3 independent runs for each line (keeping transferred filters constant across runs, but allowing for fine-tuning).


Validation Accuracy
Epoch
Unsupervised Transfer
Random Initialization
Supervised Transfer