How transferable are features in deep neural networks?

Link: https://arxiv.org/abs/1411.1792

Summary

- The first layer in a deep neural network trained on images tend to learn general features like Gabor filters and color blobs while the last layer features are specific to the task. So there should be a transition from general to specific features.
- When transferring features to a different task we can either choose to fine tune(retrain) the first *n* layers or keep them frozen. If the target dataset is quite small and number of parameters in the first *n* layers is large, fine-tuning may result in overfitting so we usually leave those features frozen. But if the target dataset is large or number of parameter is small we can fine tune to improve performance.
- When features are kept frozen two issues cause performance degradation: (1) the higher layer neurons are highly specialized for their original task so they don't generalize well to the target task and perform poorly (2) the neurons on the neighboring layers co-adapt during training(i.e., their values are optimized based on surrounding layer neurons) is such a way that cannot be rediscovered when one layer is frozen. The second effect dominates in the intermediate layers and is diminished in later layers where first effect takes dominates.
- Transferring features from almost any number of layers can produce a boost in generalization performance after fine tuning to a new dataset.
- Transferability of features decreases as the distance between the base task and target task increases particularly when transferring higher layers if features are kept frozen (paper doesn't show the result when features are fine-tuned on distant task).
- Using frozen features from a distant task for first *n* layers preforms significantly better than using random filters for the same number of layers(and freezing them, training the rest of the network).