

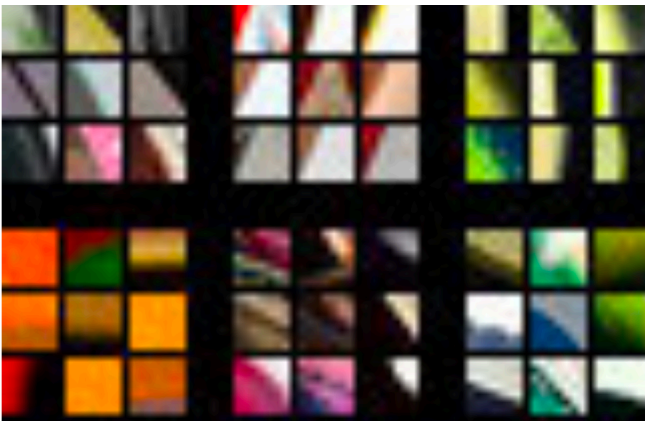
Theory of Hierarchical Learning via Gradient Descent

Margalit Glasgow
mglasgow@stanford.edu

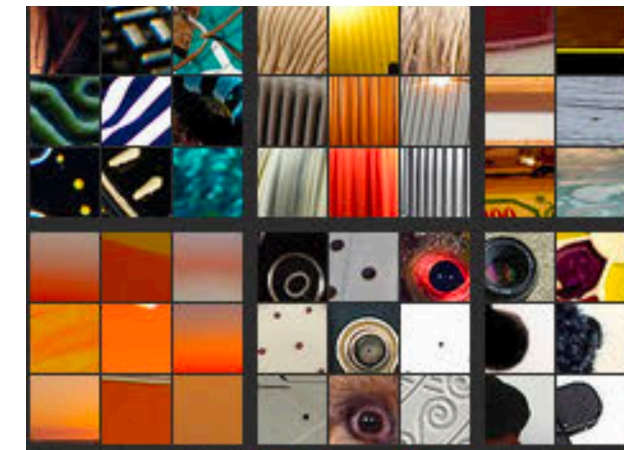
Abstract

In this project, we set the stage for theoretical study of hierarchical learning on neural networks via gradient descent. Our main goal is to design good synthetic toy settings for studying this non-trivial setting of learning. We define hierarchical learning, and consider various desired properties of a hierarchical learning problem which make it meaningfully hierarchical, likely to be amenable for analysis, and difficult enough that existing tools in the theory literature will not succeed. We then design some toy problems, and show empirically that they have these properties.

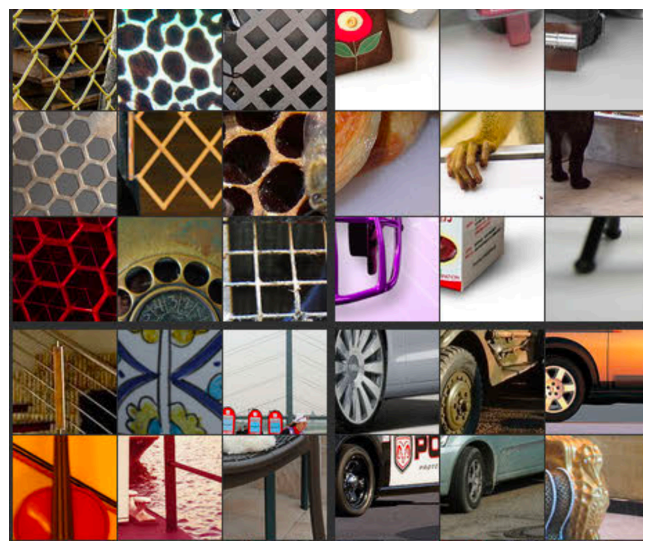
Motivation: Understanding Hierarchical Learning



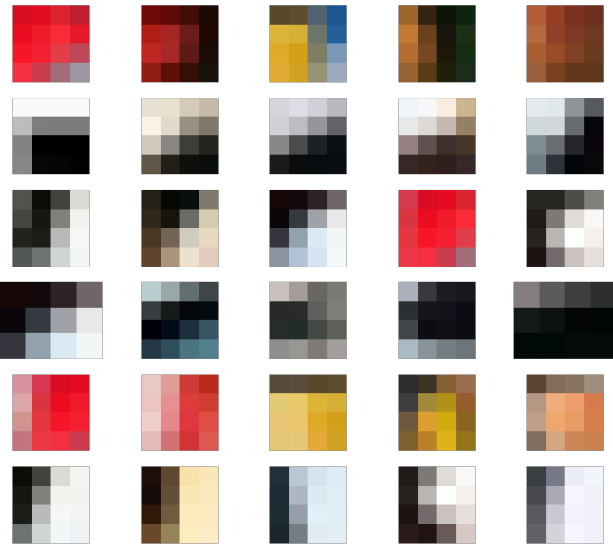
First Layer Convolutions



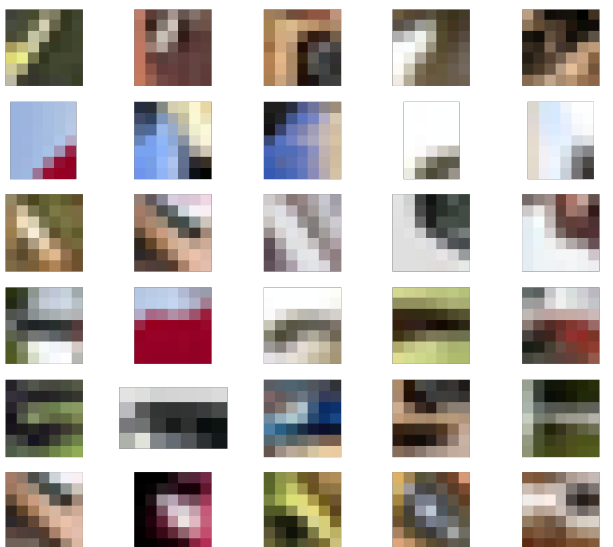
Second Layer Convolutions



Third Layer Convolutions



First Layer Convolutions



Second Layer Convolutions

Left: Patches maximally activating convolutional filters in layers 1-3 of 8 Layer CNN in [1].
Right: Patches maximally activating filters in layers 1-2 of Simple 4 Layer CNN.

Defining Hierarchical Learning

Example

Dog: Two Circles (Eyes) + Triangle (Nose) below
Truck: Circle (wheel) next to smooth patch.

Low Level Features: Circles, Triangles, Smooth patch

Higher Level Features: [Spacial] Combinations Low-level features.

Definition: Hierarchical Learning Problem

A C -class level H hierarchical learning problem $\mathcal{F}$ is a sequence of function classes $\mathcal{F}_1, \dots, \mathcal{F}_H$. Each instance $f \in \mathcal{F}$ is a sequence of functions $\{f_i \in \mathcal{F}_i\}_{i \in [H]}$, which on input $x \in \mathcal{X}$ outputs the composition $f_H(f_{H-1}(\dots f_1(x))) \in \mathbb{R}^C$.

In each function class $\mathcal{F}_i$, the functions map from some domain $\mathcal{X}_{i-1} \rightarrow \mathcal{X}_i$, where $\mathcal{X}_0 = \mathcal{X}$, and $\mathcal{X}_H = \mathbb{R}^C$.

Property 1 (Learning Required at All Levels). *If we only train any one layer of the neural network, we cannot achieve good train and test error.*

Property 2 (Low-Level Feature Interpretability). *Most first layer trained weights are correlated with a single underlying feature. This correlation is statistically significant compared to the correlations of the untrained weights with the underlying features.*

Property 3 (Kernel Regimes Inapplicable). *If we train via SGD and achieve good train and test error, then the trained network W_t travels far from its initialization W_0 .*

Property 4 (Layer-Wise-Training Fails). *If we train layer by layer from the bottom up for a suitable number of epochs, we do not achieve good test error.*

Measured as
 $\|W_t - W_0\| / \|W_0\| > 1$

Examples of 1-D Toy Problems Designed

	Class 1	Class 2
Problem 1	F1 in any location	F2 in any location
Problem 2	F1 AND F2	F1 XOR F2
Problem 3	F1 OR F2 ... OR Fk	F1 OR F2 ... OR Fk
Problem 4	(F1 AND F2) OR (F3 AND F4), sequentially	(F1 AND F3) OR (F2 AND F4), sequentially

Feature 1

Feature 2



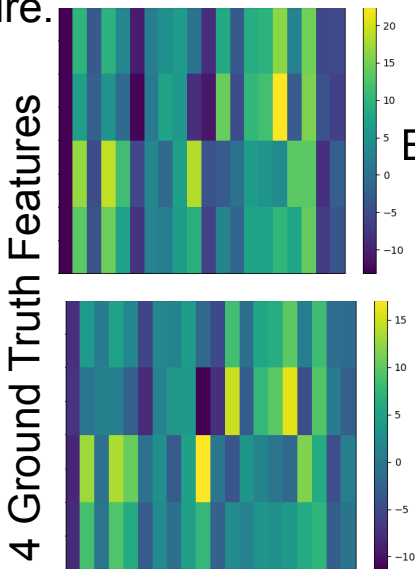
Example from Problem 4, Class 1

Results: Which Properties were Achieved by the Designed Problems?

	Epochs to 0.1 Test Err	Prop. 1? (Must Train all layers)	Prop. 2? (Interpretable)	Prop. 3? (Dev from Init)	Prop. 4? (Layer-W-Training)
Problem 1	4	No	No	No	No
Problem 2	50	Yes	No	Yes	Yes
Problem 3	44	Yes	Yes	Yes	Yes
Problem 4	46	Sort-of	Yes	Yes	No
CIFAR-10 on CNN	(Trained to 0.3 error)	Yes	Yes	Yes	Yes

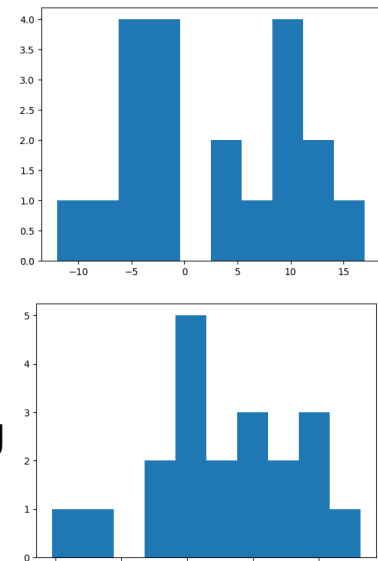
Alignment between 20 trained filters and each ground truth feature.

Histogram over 20 trained filters of max alignment with a feature.



Before Training

After Training



References 20 Filters

[1] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks.