TTIC 31230, Fundamentals of Deep Learning

David McAllester, April 2017

Interpreting Deep Networks

The Black Box Problem

The Human Black Box — Perception

Introspection is notoriously inadequate for AI.

Explain how you know there are upside down glasses in this picture.

The Human Black Box — Inference

Certain facts are obvious.

A king on empty chess board can reach every square (obvious).

A knight on an empty chess board can reach every square (true but not obvious).

The Human Black Box — Inference

Consider a graph with colored nodes.

If every edge is between nodes of the same color, then any path connects nodes of the same color.

Consider a swiss chocolate bar of 3×5 little squares.

How many breaks does it take to reduce this to fifteen unconnected squares?

Why Open the Box?

Insight for improving system design.

Insight into the reasons for system decisions.

Convincing people to accept system decisions.

Dimensionality Reduction

Visualizing the representation

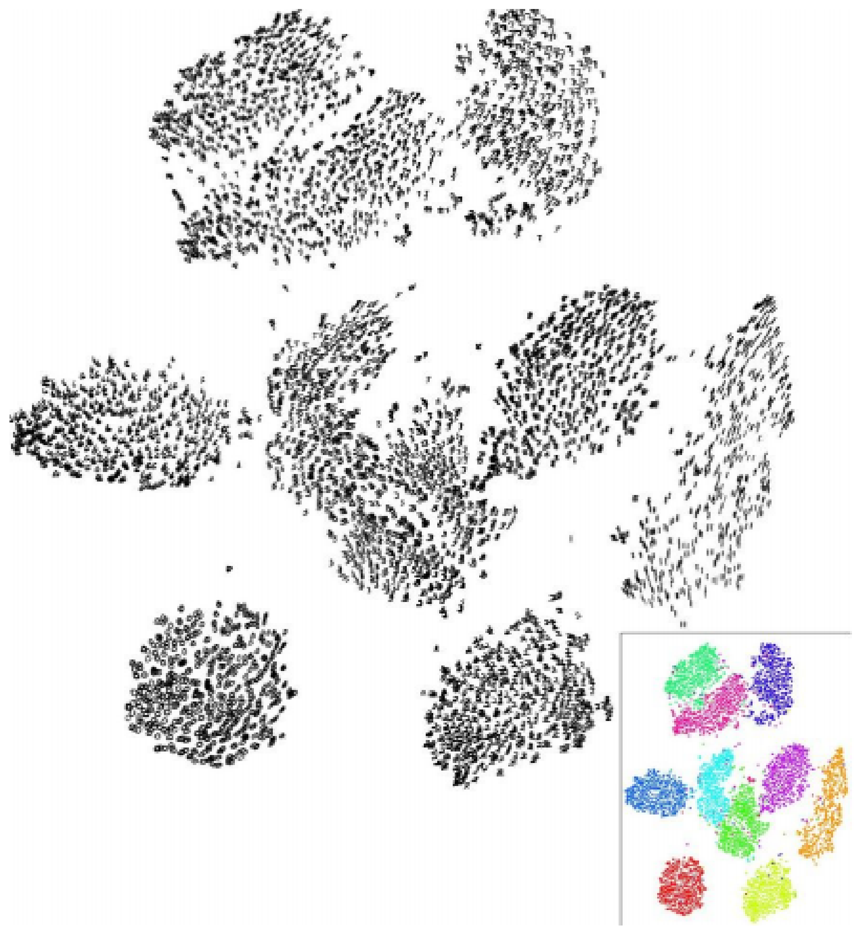
t-SNE visualization

[van der Maaten & Hinton]

Embed high-dimensional points so that locally, pairwise distances are conserved

i.e. similar things end up in similar places. dissimilar things end up wherever

Right: Example embedding of MNIST digits (0-9) in 2D



[Stanford CS231]

t-SNE

Consider high dimensional points x_1, ..., x_N.

$$P(j|i) = \frac{1}{Z_i} \exp\left(\frac{-||x_i - x_j||^2}{2\sigma_i^2}\right)$$

Set σ_i such that $H(P_i(:|i)) = H(P(:|j))$ for all i, j.

$$P(\{i,j\}) = \frac{P(i|j) + P(j|i)}{2N}$$

$$\text{we have} \quad \sum_j P(\{i,j\}) \geq \sum_j \frac{P(j|i)}{2N} = \frac{1}{2N}$$

t-SNE

Consider low dimensional points $y_1, \ldots, y_N$.

$$Q(\{i,j\}) = \frac{1}{Z}\left(\frac{1}{1+||y_i - y_j||^2}\right)$$

$$Y^* = \operatorname*{argmin}_Y H(P,Q)$$

t-SNE vs. Projection Modeling

t-SNE — $y(x)$ is defined by a table on the data points.

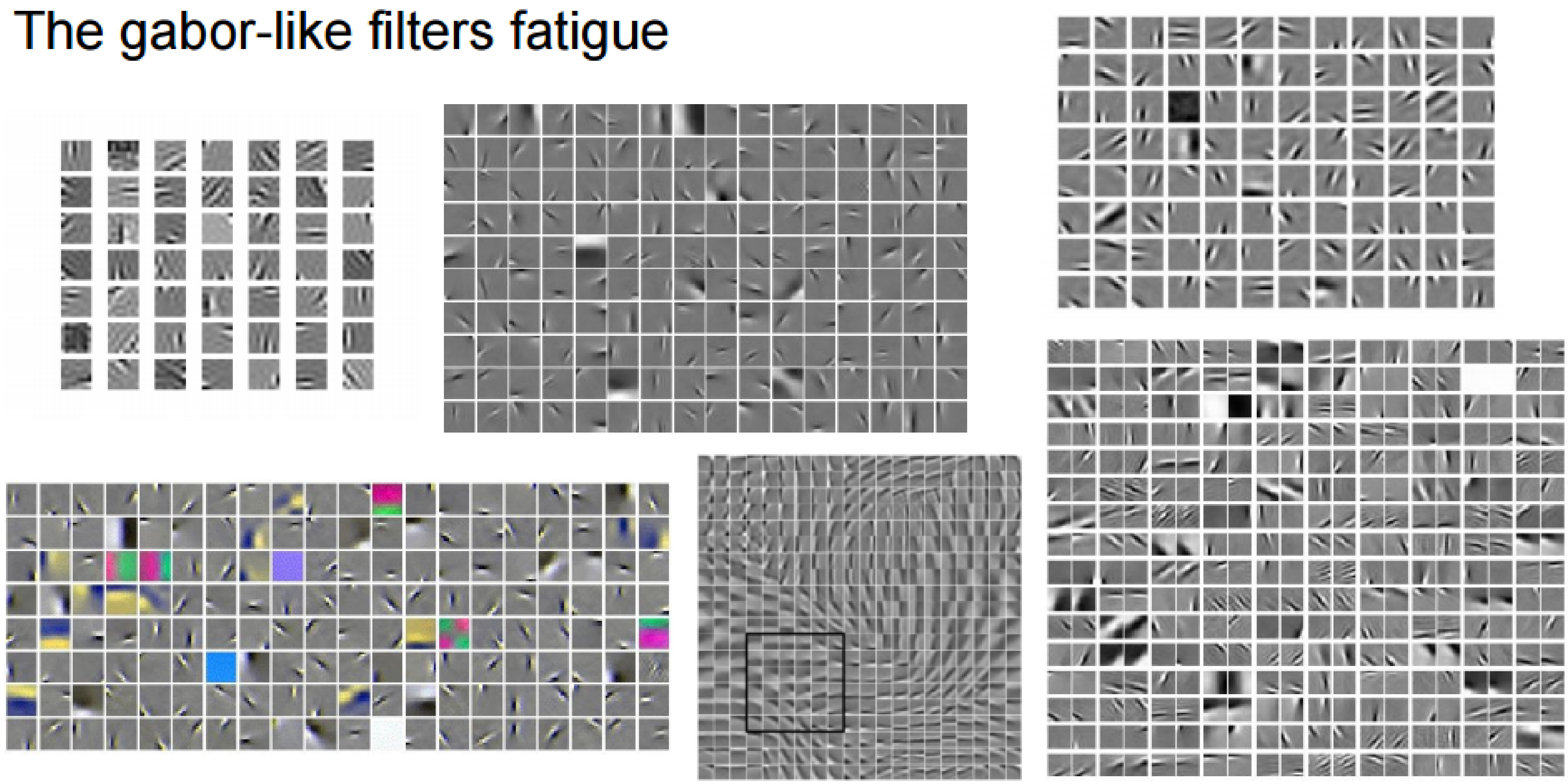
In PCA or Isomap we have $y_\Phi(x) \in \mathbb{R}^2$ for a parameterized function y_Φ.

Assume a (nearest neighbor) distribution on pairs (x_1, x_2) and consider random variables $y_1 = y_\Phi(x_1)$ and $y_2 = y_\Phi(x_2)$.

An idea (not PCA or Isomap) :

$$\Phi^* = \operatorname*{argmax}_{\Phi} I(y_1, y_2)$$

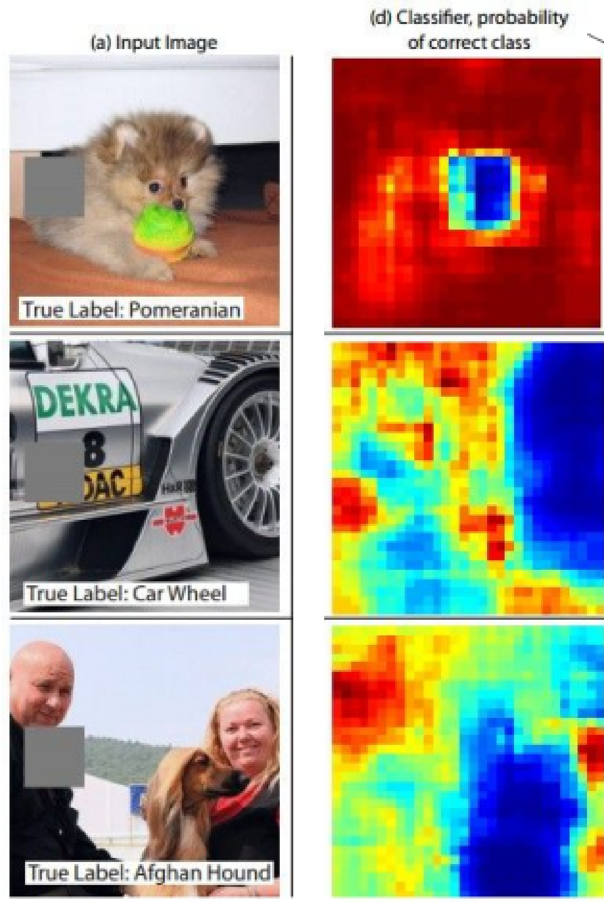
Visualizing the Filters



[Stanford CS231]

Occlusion experiments

[Zeiler & Fergus 2013]



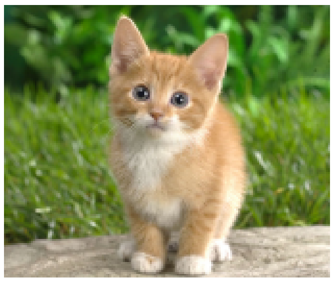
(as a function of the position of the square of zeros in the original image)

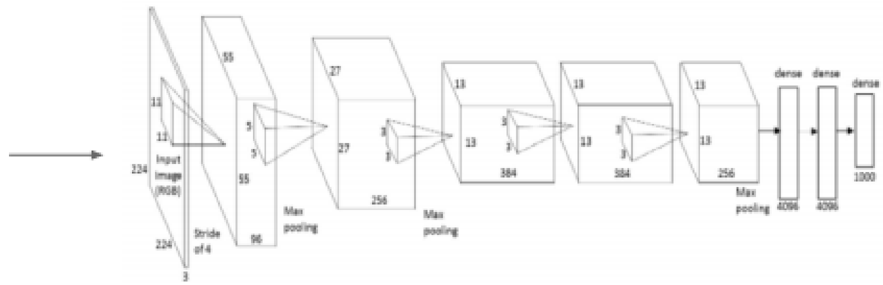
[Stanford CS231]

Backpropagation from Individual Neurons

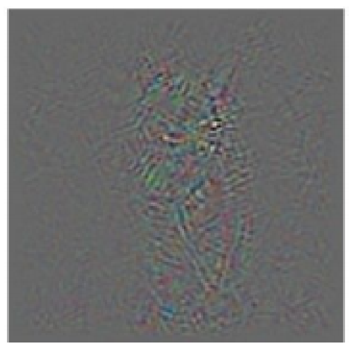
Deconv approaches

1. Feed image into net





2. Pick a layer, set the gradient there to be all zero except for one 1 for some neuron of interest
3. Backprop to image:



"Guided backpropagation:" instead

[Stanford CS231]

Guided Backpropagation

Rather than $\partial \ell / \partial x$ we are interested in $\partial \mathrm{neuron} / \partial x$.

We are interested in $\partial \mathrm{neuron} / \partial x$ where x in one color channel of one input pixel.

It turns out that $\partial \mathrm{neuron} / \partial x$ looks like image noise.

Instead we compute $x.\mathrm{ggrad}$ — a **guided** version of $\partial \mathrm{neuron} / \partial x$.

Guided Backpropagation

Guided backpropagation only considers computation paths that activate (as opposed to suppress) the neuron all along the activation path.

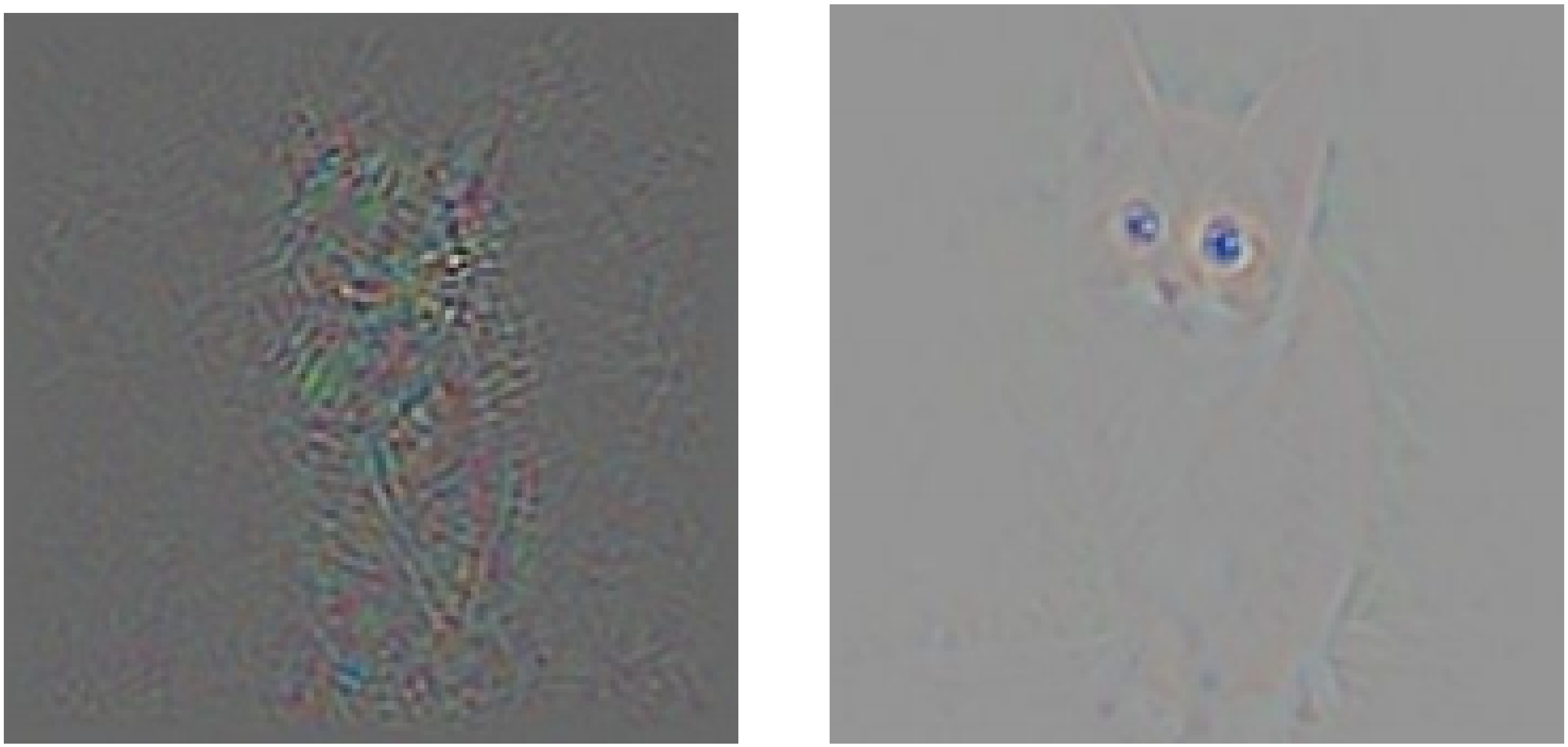
The backpropagation at activation functions is modified.

For a neuron y with $y = s(x)$ for activation function s:

$$x.\mathrm{ggrad} = I[y.\mathrm{ggrad} > 0]\; y.\mathrm{ggrad}\; ds/dx$$

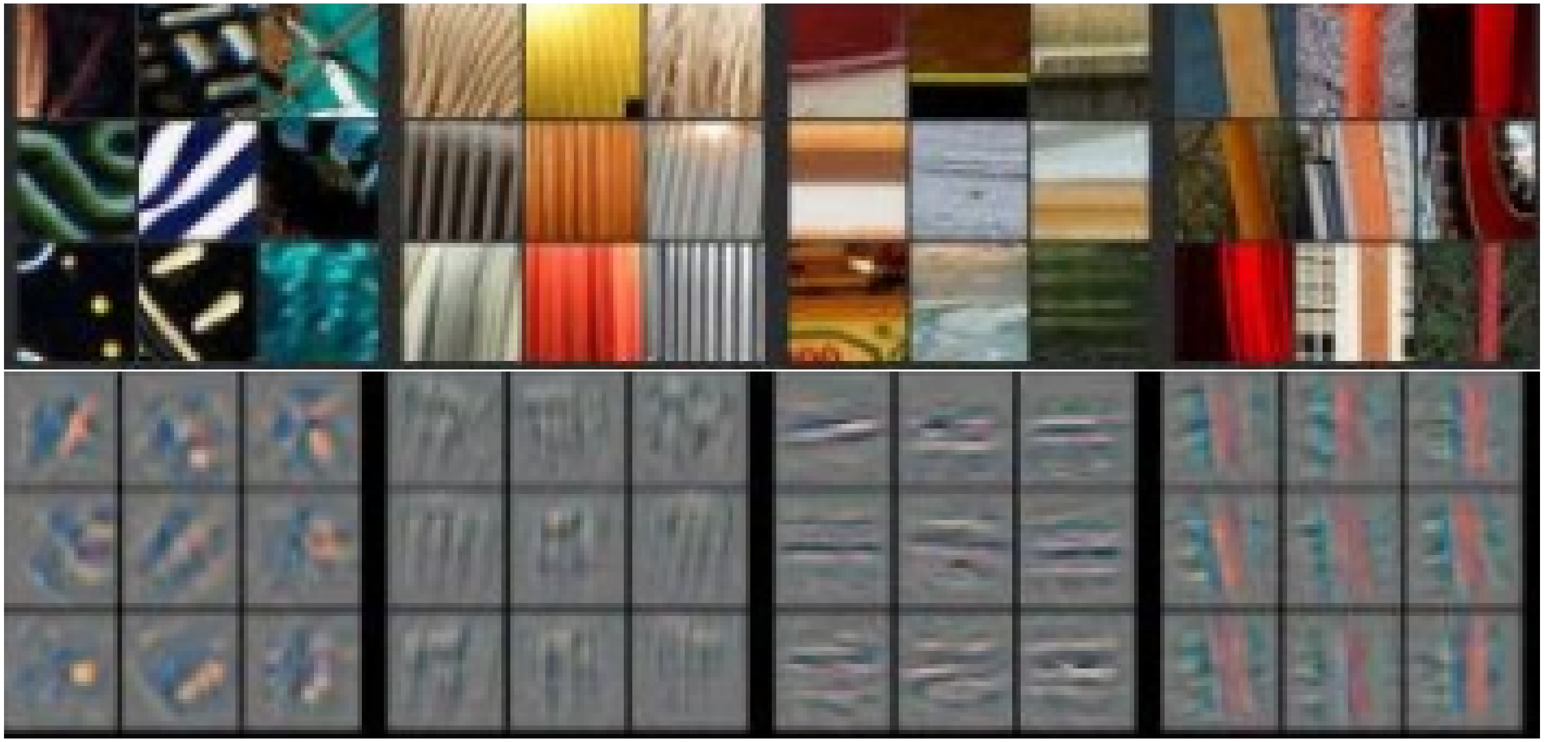
Guided Backpropagation

Guided Backpropagation



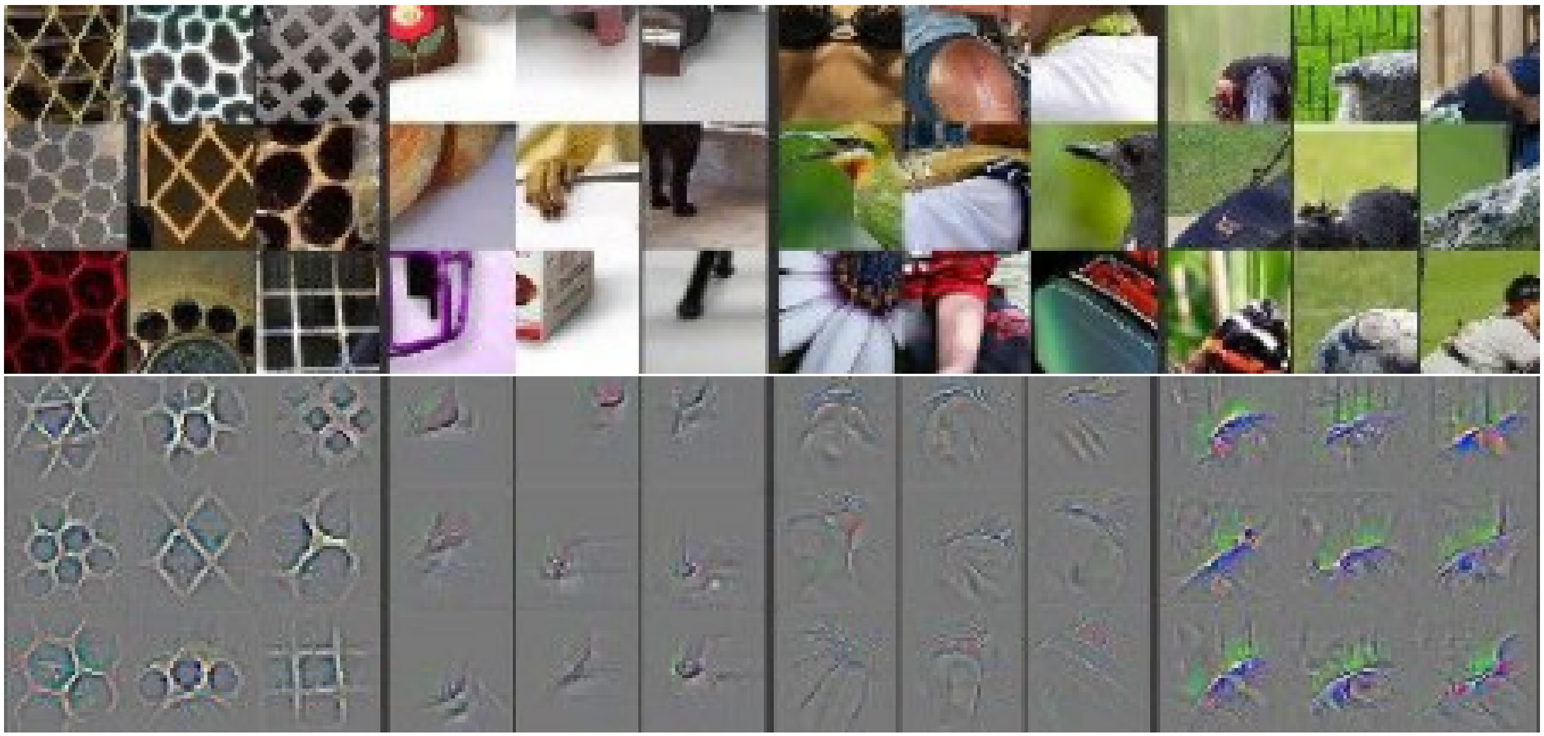
[Zeigler and Fergus 2013]

Guided Backpropagation Layer 2



[Zeigler and Fergus 2013]

Guided Backpropagation Layer 3



[Zeigler and Fergus 2013]

Guided Backpropagation Layer 4

[Zeigler and Fergus 2013]

Guided Backpropagation Layer 5

[Zeigler and Fergus 2013]

A Wheel or Face Detector

The nine strongest stimulators of the "wheel or face cell" are the following.

[Zeigler and Fergus 2013]

it's like "*vodka & potato*" classifier!

+

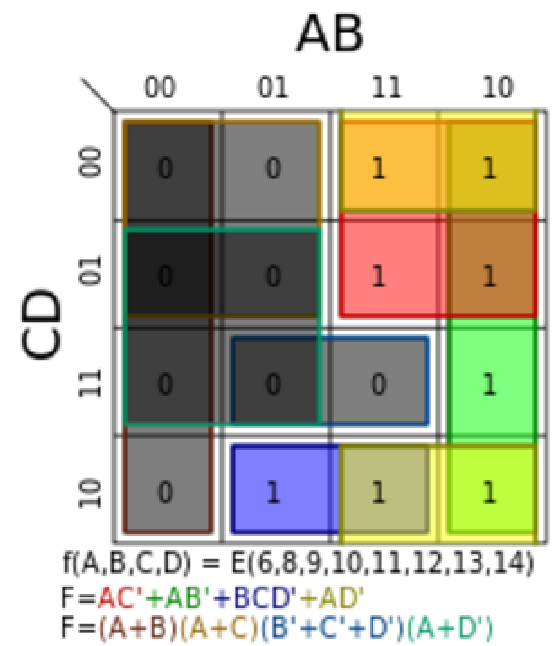
[Alyosho Efros]

The Kaurnaugh Model of DNNs

The Karnaugh map, also known as the K-map, is a method to simplify boolean algebra expressions.

Truth table of a function

	A	B	C	D	f(A, B, C, D)
0	0	0	0	0	0
1	0	0	0	1	0
2	0	0	1	0	0
3	0	0	1	1	0
4	0	1	0	0	0
5	0	1	0	1	0
6	0	1	1	0	1
7	0	1	1	1	0
8	1	0	0	0	1
9	1	0	0	1	1
10	1	0	1	0	1
11	1	0	1	1	1
12	1	1	0	0	1
13	1	1	0	1	1
14	1	1	1	0	1
15	1	1	1	1	0



$$
\begin{aligned}
F(A,B,C,D) &= AC' + AB' + BCD' + AD' \\
&= (A+B)(A+C)(B'+C'+D')(A+D')
\end{aligned}
$$

A Kaurnaugh Person Detector

Wheel or Face

Hand or Flower Hand or Flower

Leg or Tree Leg or Tree

The set of locally minimal models (circuits) could be vast (exponential) without damaging performance.

Is a Boolean circuit a distributed representation?

The Glass Model of SGD

Pysical glass (ordinary silica glass) is a metastable state — the ground state is quartz crystal.

As molten glass cools there is a temperature T_g (± 1 degrees C) at which it "solidifies" (the viscosity becomes huge).

This soldification process is very repeatable with a well defined final energy.

However, the local optimum achieved is presumably very different for each instance of cooling.

Identifying Channel Correspondences

Convergent Learning: Do Different Neural Networks Learn The Same Representations?, Li eta al., ICLR 2016.

Train Alexnet twice with different initializations to get net1 and net2.

For each convolution layer, each channel i of net1, and each channel j of net2, compute their correlation.

$$\rho_{i,j} = \mathrm{E}\left[\frac{(u_i - \mu_i)(u_j - \mu_j)}{\sigma_i \sigma_j}\right]$$

Semi-matching and Bipartite matching

Semi-matching: for each i in net1 find the best j in net2:

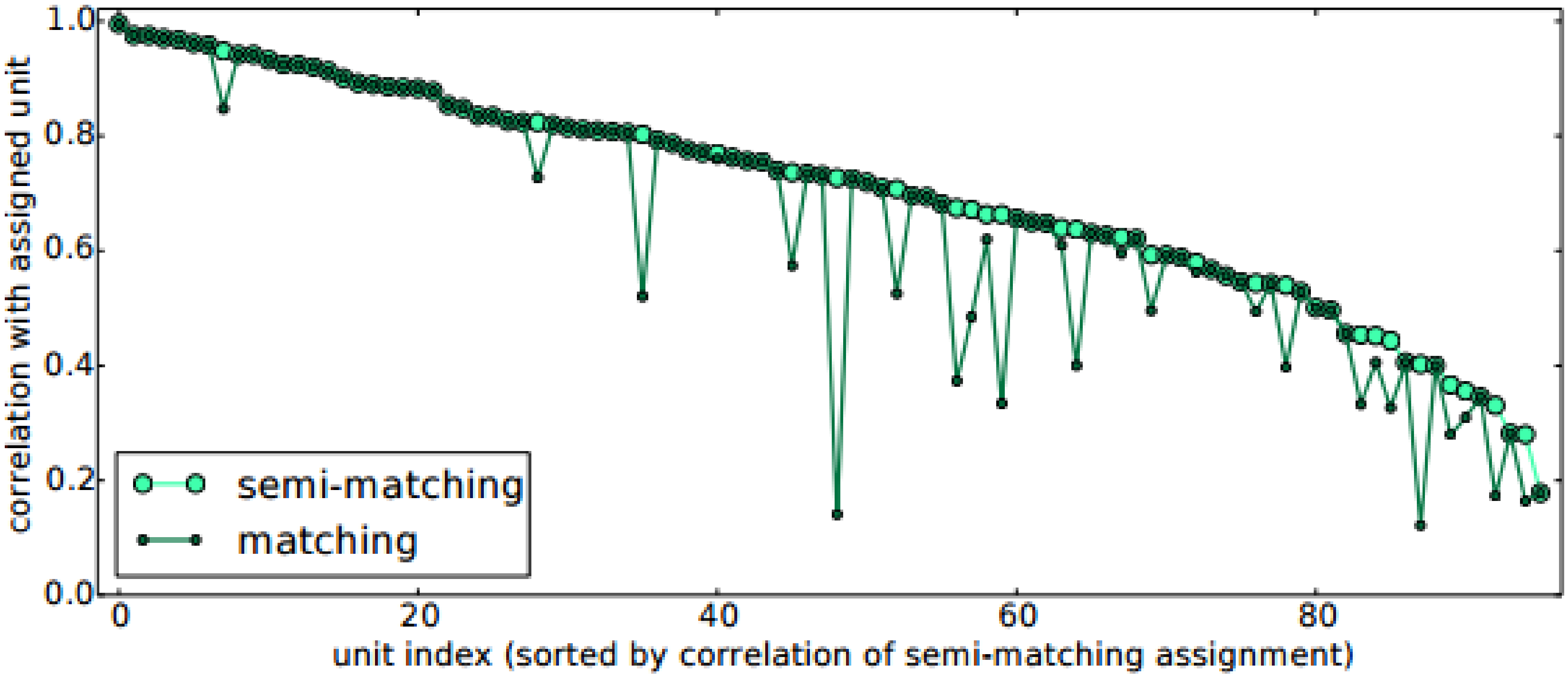
$$\hat{j}(i) = \operatorname*{argmax}_{j} \rho_{i,j}$$

Biparetitie Matching: Find the best one-to-one correspondence.

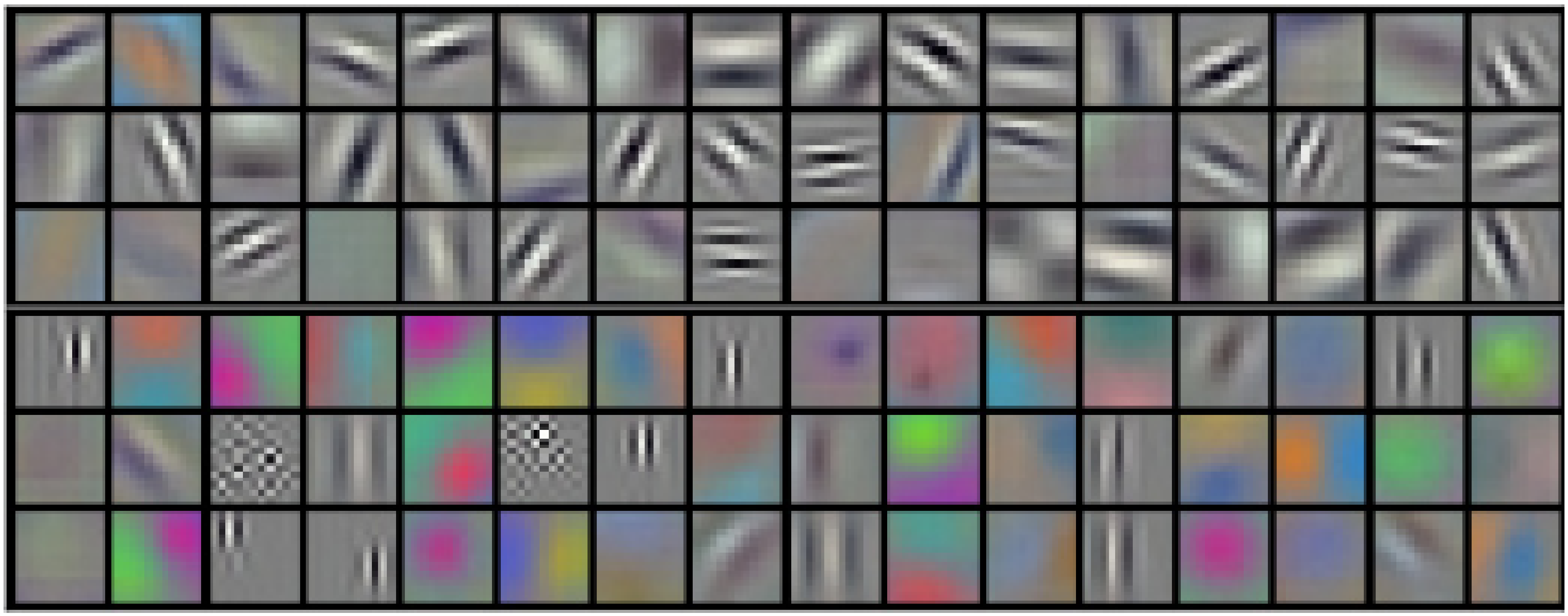
$$\hat{j} = \operatorname*{argmax}_{\hat{j} \text{ a bijection}} \sum_{i} \rho_{i,\hat{j}(i)}$$

Bipartite matching can be solved by a classical algorithm [Hopcroft and Karp, 1973]. John Hopcroft (age 77) is an author on this ICLR paper.

Correlations at Layer 1 (Wavelet Layer)

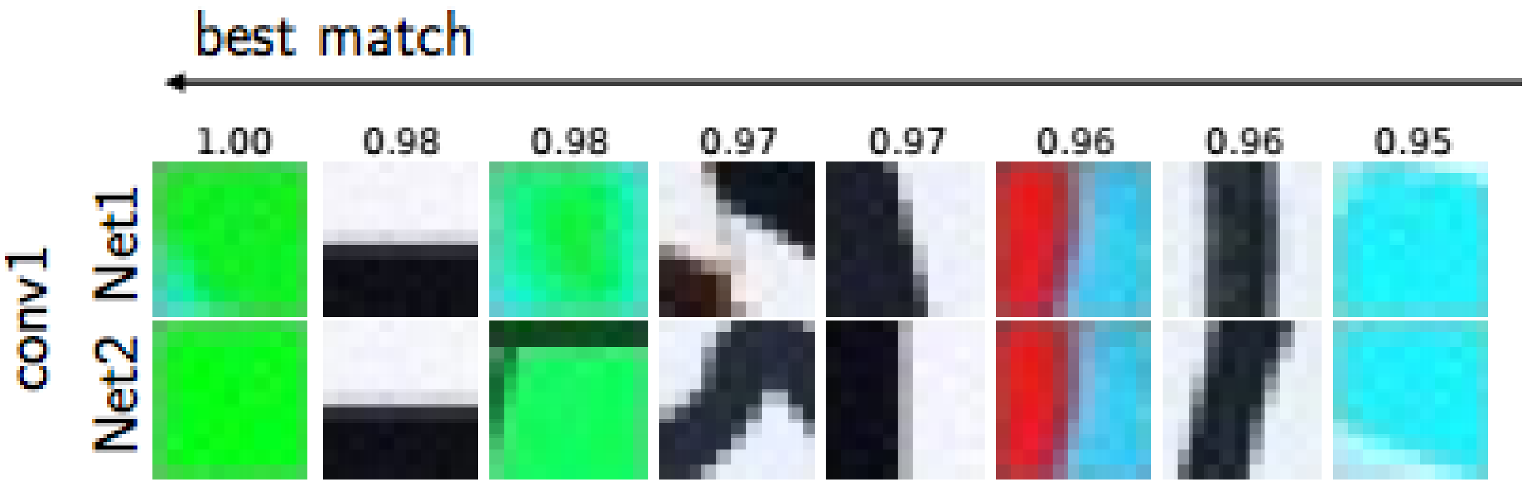


Alexnet Layer 1



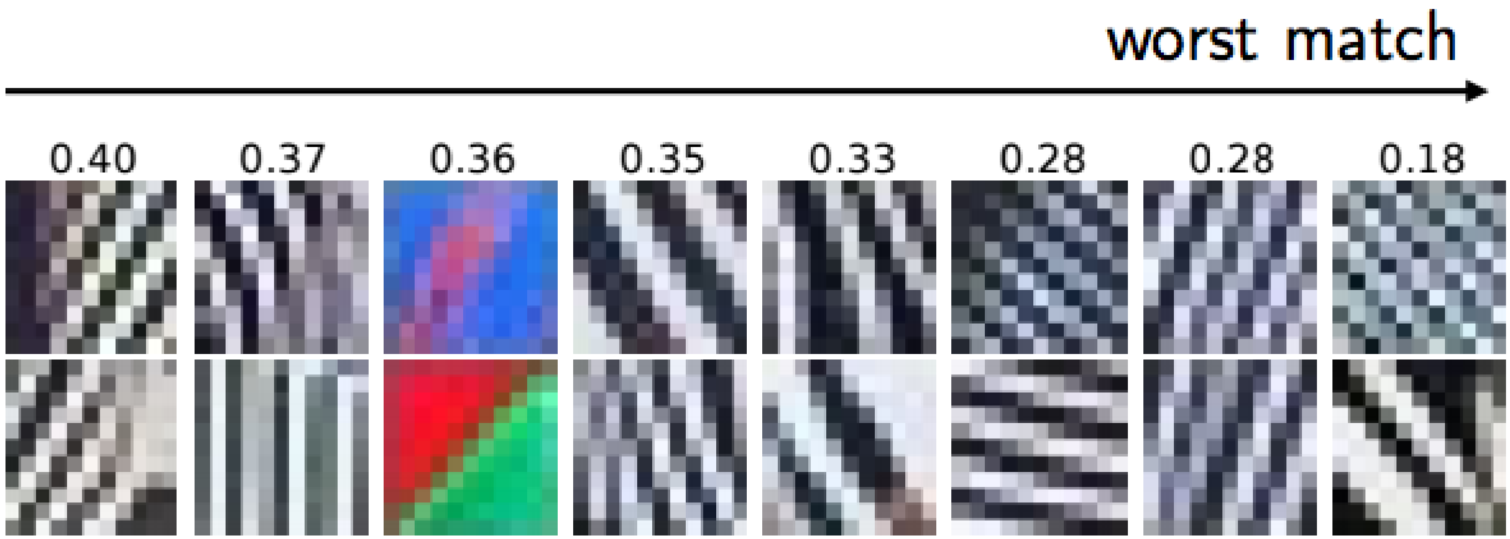
[Krizhevsky et al.]

Best Matches in Layer 1 semi-matching



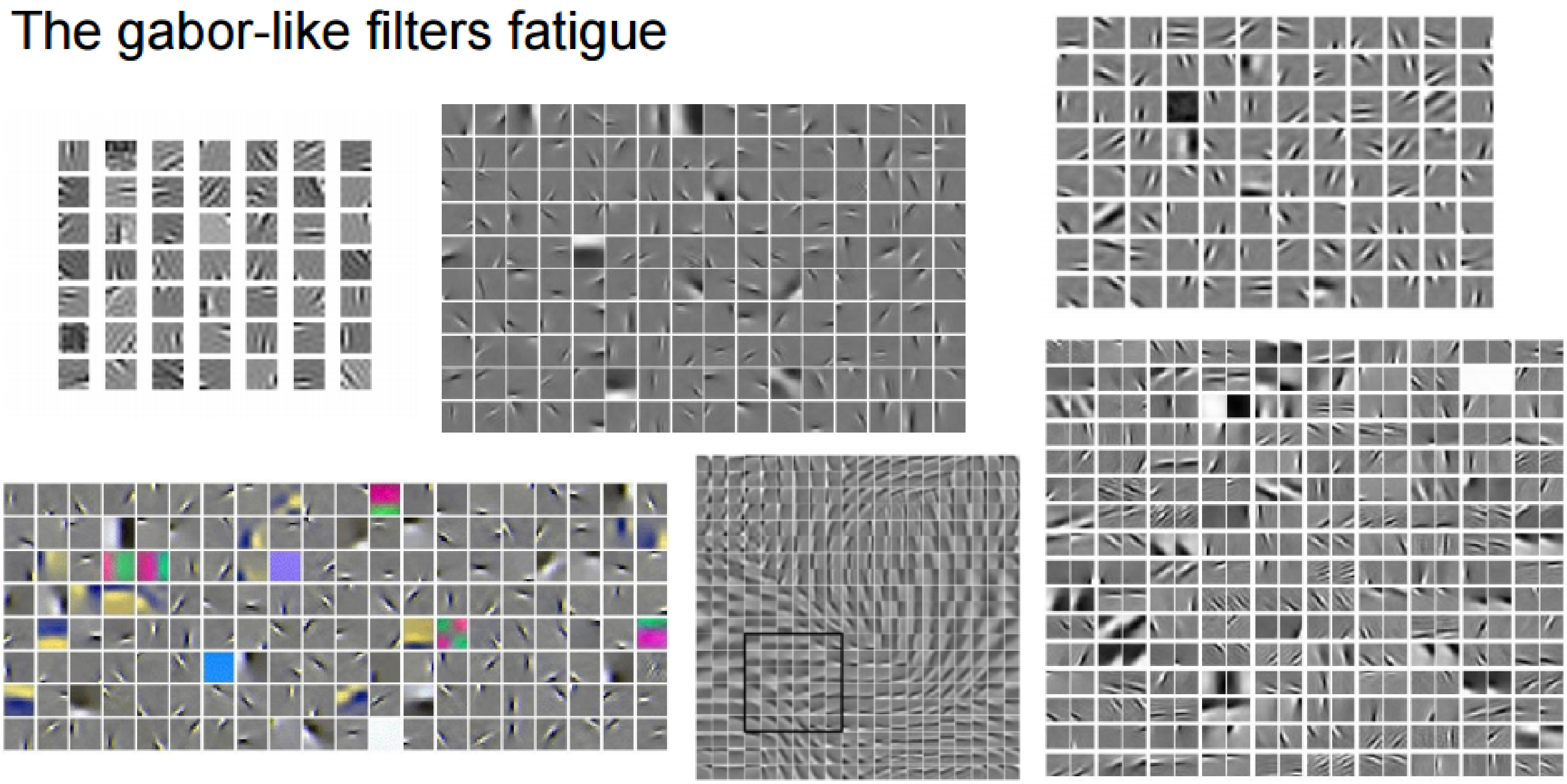
[Li et al.]

Worst Matches in Layer 1 semi-matching



[Li et al.]

Layer 1 in Other Networks



[Stanford CS231]

Regression Between Networks at Layer 1

Model each channel of net1 as a linear combination of channels of net2 using least squares regression.

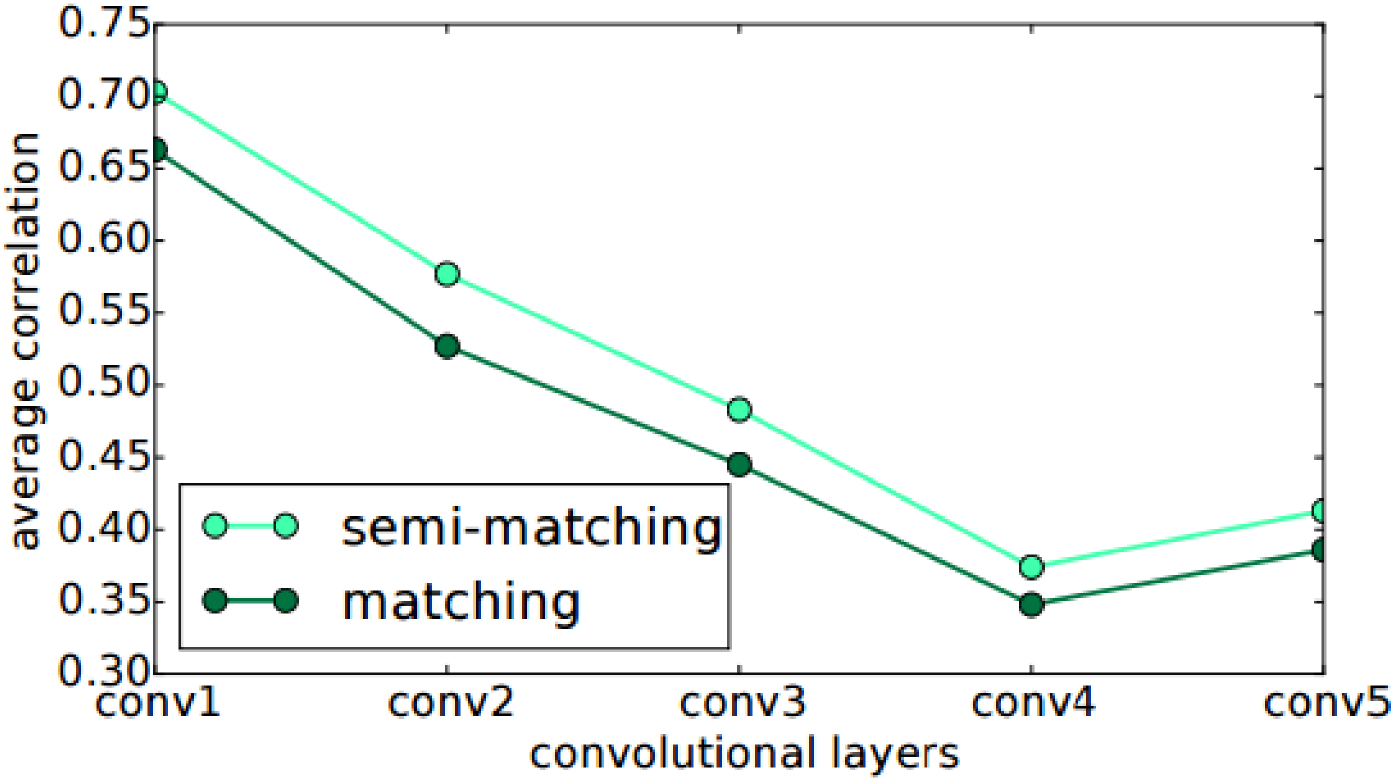
Before the regression each channel is normalized to have zero mean and channel variance.

No correlation would yield a square loss of 1.000.

No regularization gives a square loss of 0.170 and uses 96 channels in each prediction.

L1 regularization gives a square loss of 0.235 and uses 4.7 channels in each prediction.

Deeper Layers



In the regression experiment squared error was not significantly reduced at layers 3 through 5 even without regularization.

Model Compression

Deep Compression: Compressing Deep Neural Networks With Pruning, Trained Quantization and Huffman Coding, Han et al., ICLR 2016.

- Compressed Models can be downloaded to mobile devices faster and fit in lower-power CPU memory. (The motivation of this paper).
- Sparse models may be more interpretable than dense models.
- Model size is a measure of model complexity and can be viewed as a form of regularization.

VGG-16 is reduced by $49\times$ from 552MB to 11.3MB with no loss of accuracy.

Three Stages

- Sparsification by simple weight thresholding. ($10\times$ reduction).

- Trained Quantization ($6\times$ reduction).

- Huffman coding (40% reduction).

Quantization

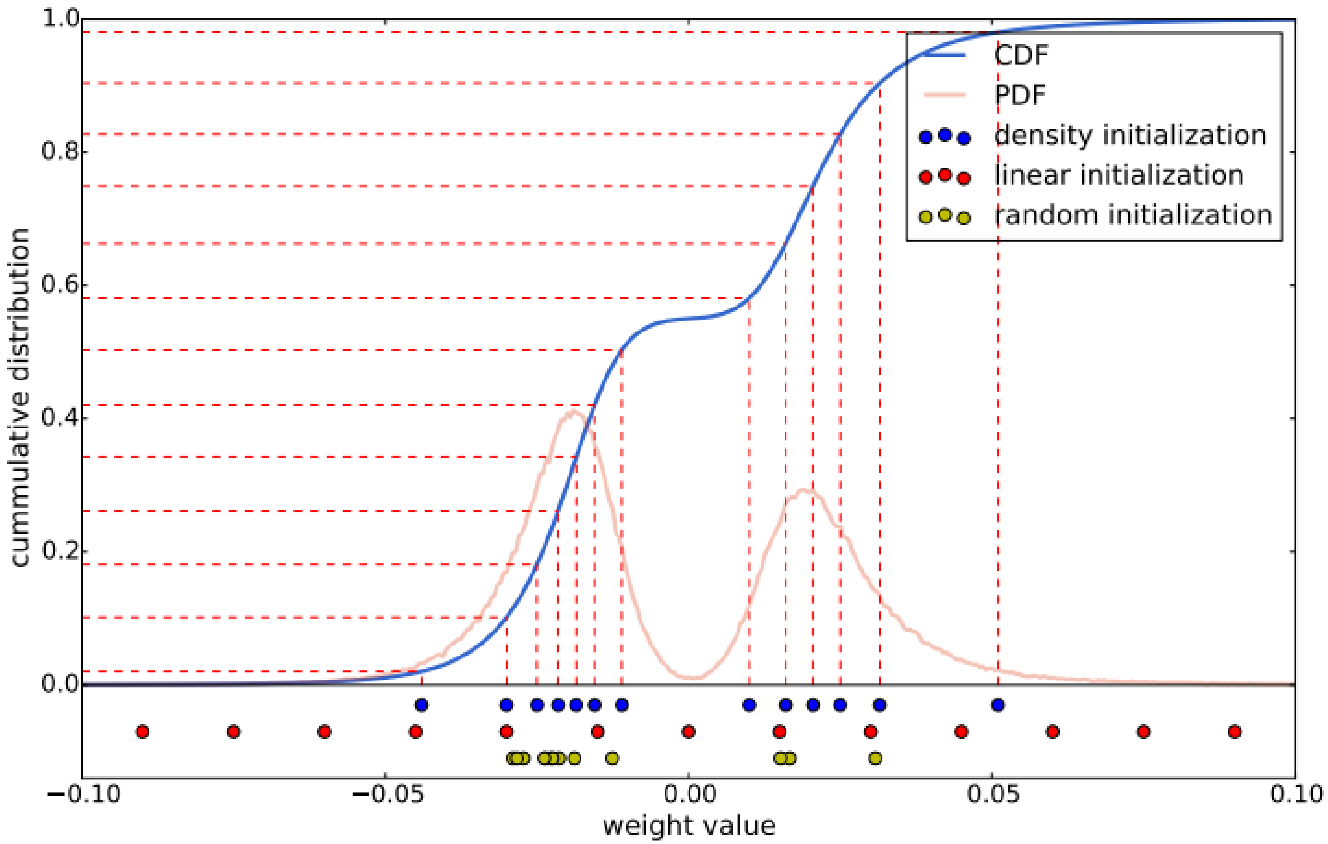
They use 5 bits of numerical precision for the weights.

This is done by having a table of the 32 possible weight values.

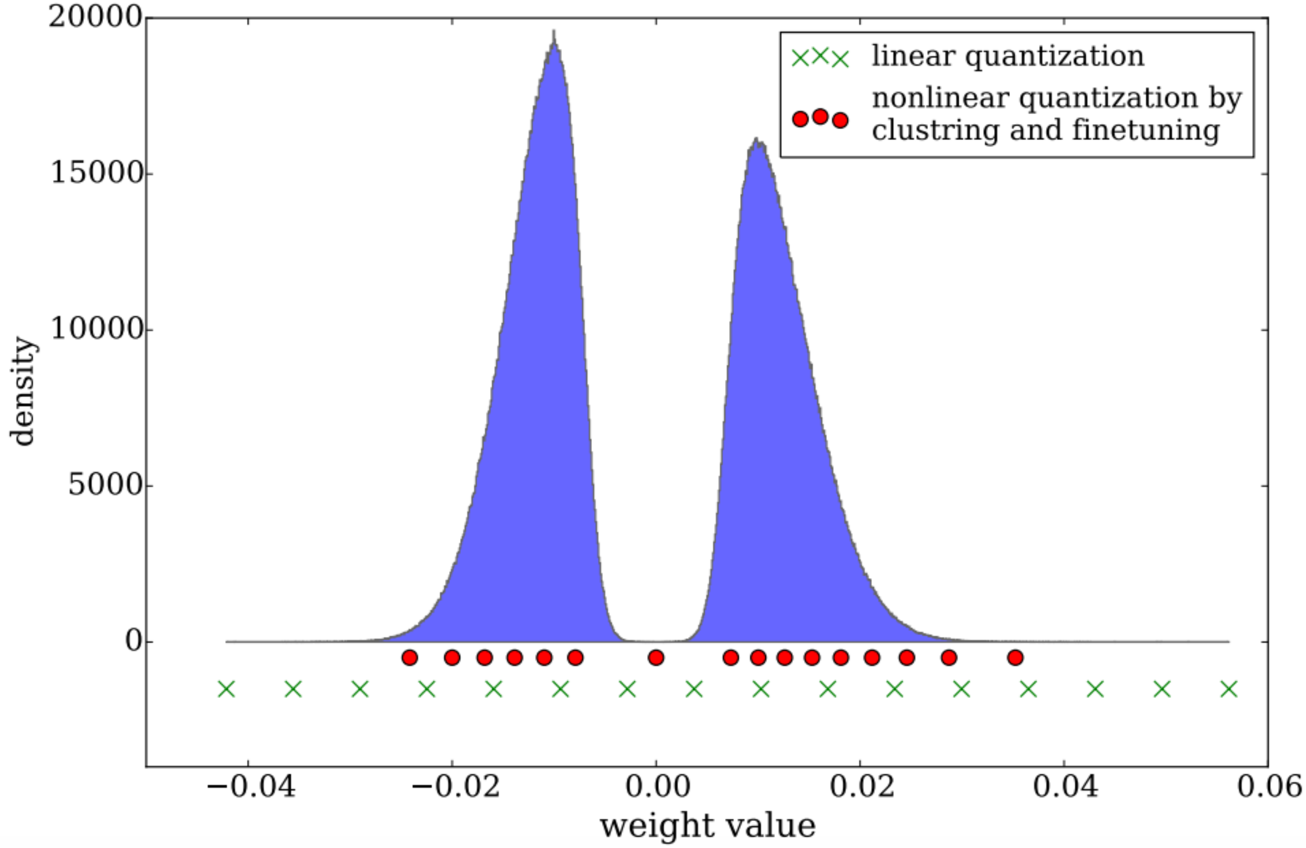
We have to cluster the weights into 32 groups and decide on a **centroid value** for each weight.

This is done with K-means clustering.

Initialization of Centroids



After Running K-means



Retrain to Adjust Centroids

Run over the data again doing backpropagation to adjust the table of the 32 possible weights.

This leaves the 5-bit code of each weight in the model unchanged.

Huffman Coding

Different 5-bit numerical codes have different frequencies.

This can be viewed as distribution over the 32 code words.

We can reduce the average number of bits per weight using fewer bits to code the more common weight values.

Huffman coding is applied to both the 5 bit weight coding and a three bit code used in the sparse representation of the weight matrices.

This results in about 5 bits per **nonzero** weight in a **sparse** coding of the weight matrices.

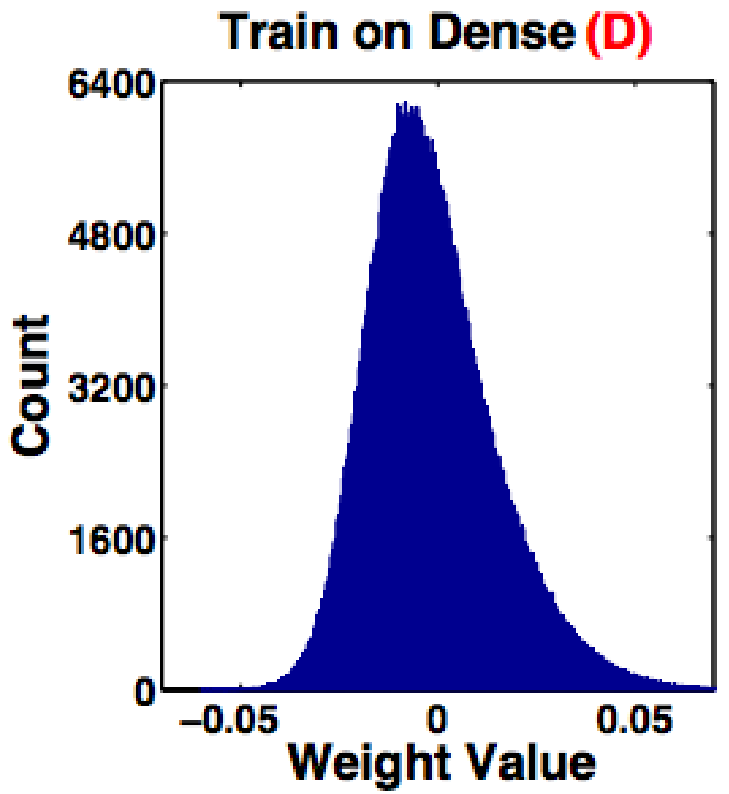
Dense-Sparse-Dense

DSD: Dense-Sparse-Dense Training for Deep Neural Networks, Han et al., ICLR 2017

1. Train a model.
2. Make the model sparse by weight thresholding.
3. Retrain the model holding the sparsity pattern fixed (still 32 bits per weight).
4. Go back to a dense model with all pruned weights initialized to zero.
5. Retrain the dense model.

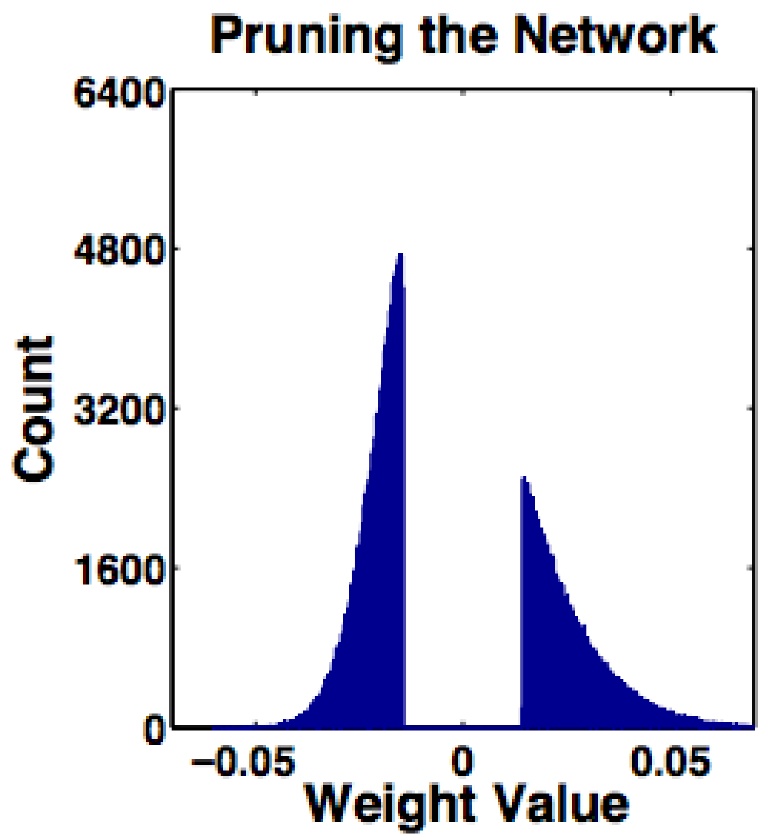
Results in significant performance improvements in a wide variety of models.

Step 1



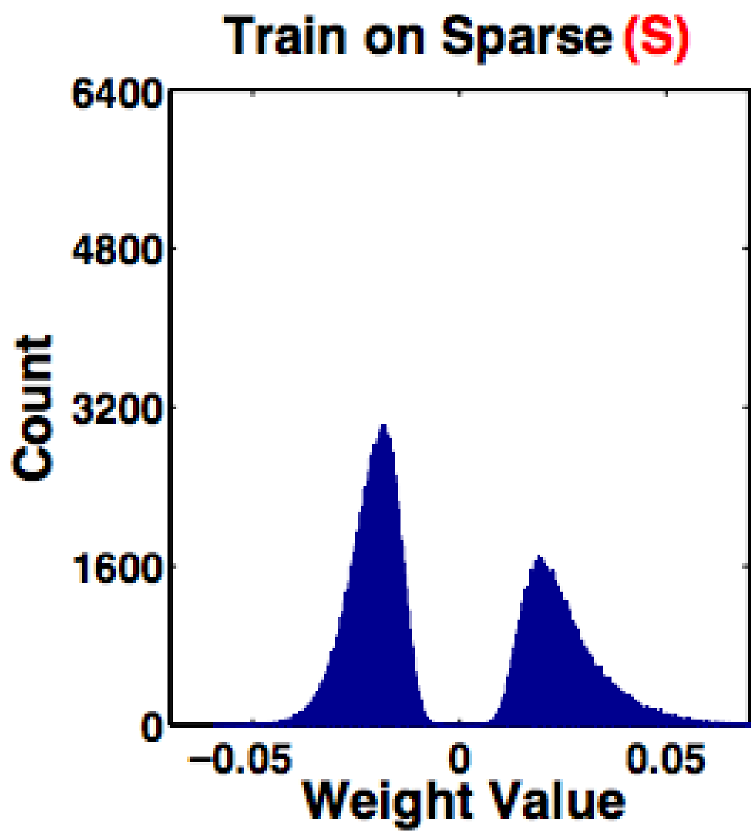
Train on Dense (D)
6400
4800
3200
1600
0
Count
−0.05
0
0.05
Weight Value

Step 2



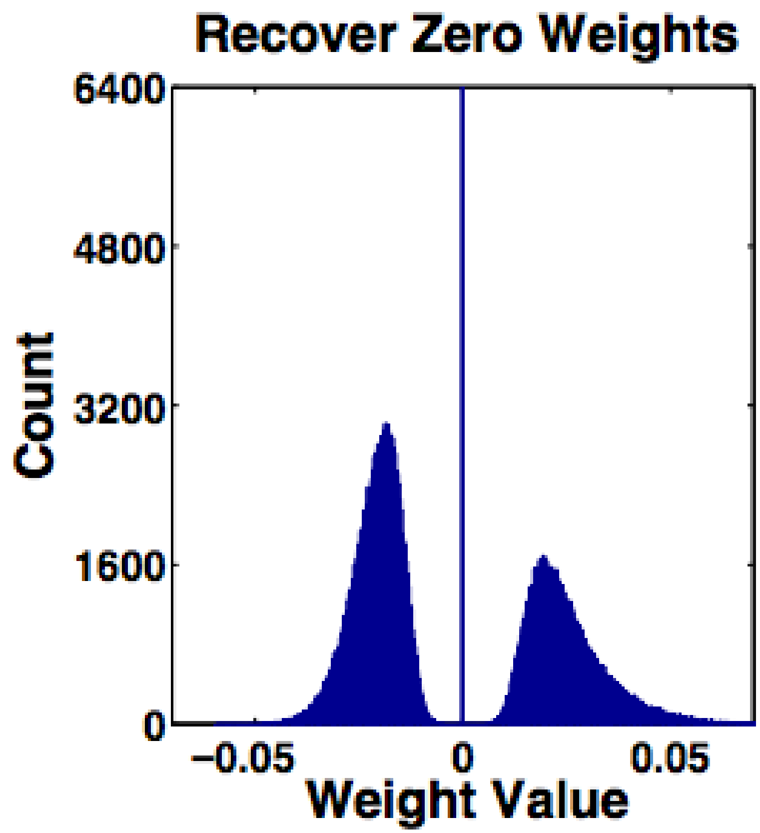
Pruning the Network
6400
4800
3200
1600
0
Count
−0.05
0
0.05
Weight Value

Step 3



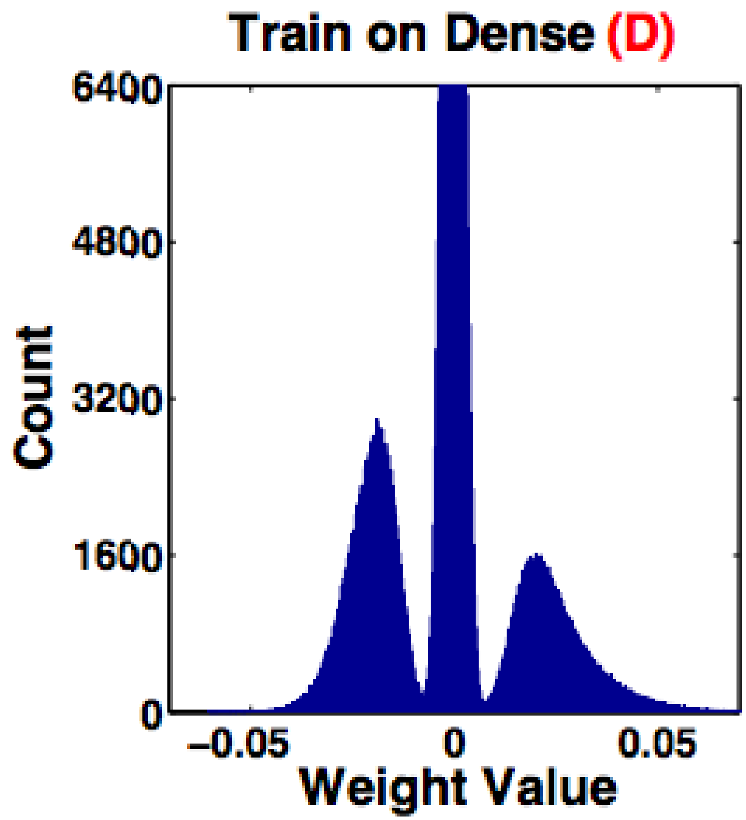
Train on Sparse (S)
6400
4800
3200
1600
0
Count
−0.05
0
0.05
Weight Value

Step 4



Recover Zero Weights
6400
4800
3200
1600
0
Count
−0.05
0
0.05
Weight Value

Step 5



Train on Dense (D)
6400
4800
3200
1600
0
Count
−0.05
0
0.05
Weight Value

Results

Neural Network	Domain	Dataset	Type	Baseline	DSD	Abs. Imp.	Rel. Imp.
GoogLeNet	Vision	ImageNet	CNN	31.1%[1]	**30.0%**	1.1%	3.6%
VGG-16	Vision	ImageNet	CNN	31.5%[1]	**27.2%**	4.3%	13.7%
ResNet-18	Vision	ImageNet	CNN	30.4%[1]	**29.2%**	1.2%	4.1%
ResNet-50	Vision	ImageNet	CNN	24.0%[1]	**22.9%**	1.1%	4.6%
NeuralTalk	Caption	Flickr-8K	LSTM	16.8[2]	**18.5**	1.7	10.1%
DeepSpeech	Speech	WSJ'93	RNN	33.6%[3]	**31.6%**	2.0%	5.8%
DeepSpeech-2	Speech	WSJ'93	RNN	14.5%[3]	**13.4%**	1.1%	7.4%

Attention as Explanation

A woman is throwing a frisbee in a park.

A little girl sitting on a bed with
a teddy bear.

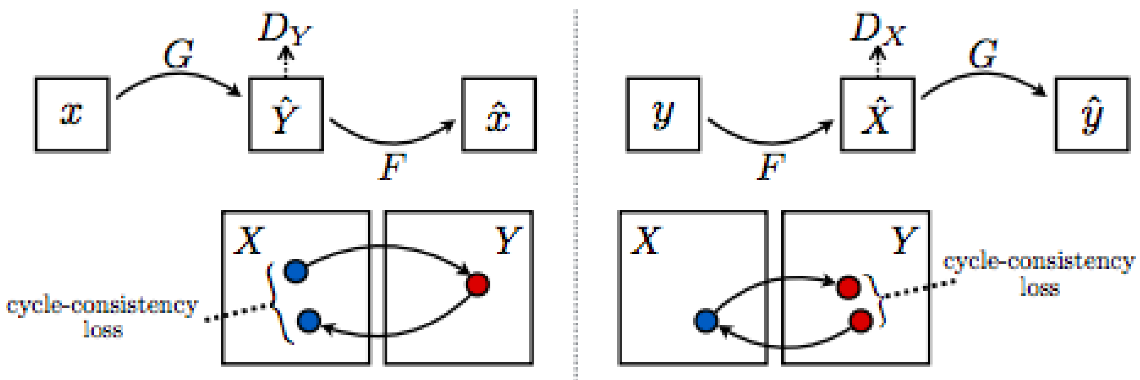
Xu et al. ICML 2015

Interpretation from Domain Correspondence

Kushner verliert den Zugang zu streng geheimen Informationen.

$$\Rightarrow$$

Kushner loses access to top-secret intelligence.



Causal Models are Explicitly Interpretable

Flu causes symptoms x, y z.

Strep causes symptoms x, y, u.

For the given information on the patient, the prior probability for flu is ...

Can Alpha Zero Explain Chess Moves?

I did x because if I did y they would do z and, in that case, ...

END