

Deep Learning

An historical perspective

Alessandro Ferrari
alessandroferrari87@gmail.com

Have you ever complained about demanding teachers?

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
PROJECT MAC

Artificial Intelligence Group
Vision Memo. No. 100.

July 7, 1966

THE SUMMER VISION PROJECT

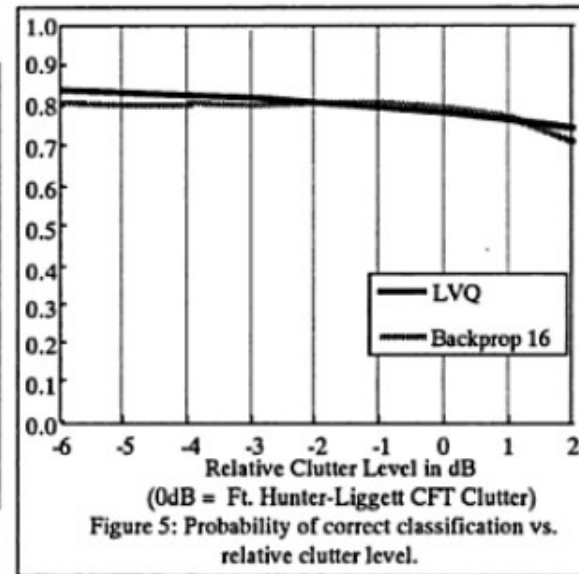
Seymour Papert

The summer vision project is an attempt to use our summer workers effectively in the construction of a significant part of a visual system. The particular task was chosen partly because it can be segmented into sub-problems which will allow individuals to work independently and yet participate in the construction of a system complex enough to be a real landmark in the development of "pattern recognition".

High hopes...

		Training		Testing	
		Tank	Truck	Tank	Truck
LVQ	Tank	0.892	0.108	0.795	0.205
	Truck	0.113	0.887	0.211	0.789
MLP 16 Hidden	Tank	0.824	0.176	0.774	0.226
	Truck	0.118	0.882	0.193	0.807
MLP 3 Outputs	Tank	0.968	0.032	0.812	0.188
	Truck	0.119	0.881	0.216	0.784

Table 1: Neural Networks Performance Summary
Confusion Matrices



4.0 RESULTS:

The Multi-Level Perceptron and the Learning Vector Quantizer were capable of classifying tanks and trucks to roughly 80%. A summary of confusion matrices for a sampling of some networks is given in Table 1. Robustness to clutter levels was high, varying little from -6dB attenuated to +2dB amplified clutter. Figure 5 shows the performance as a function of clutter for the MLP and LVQ networks.

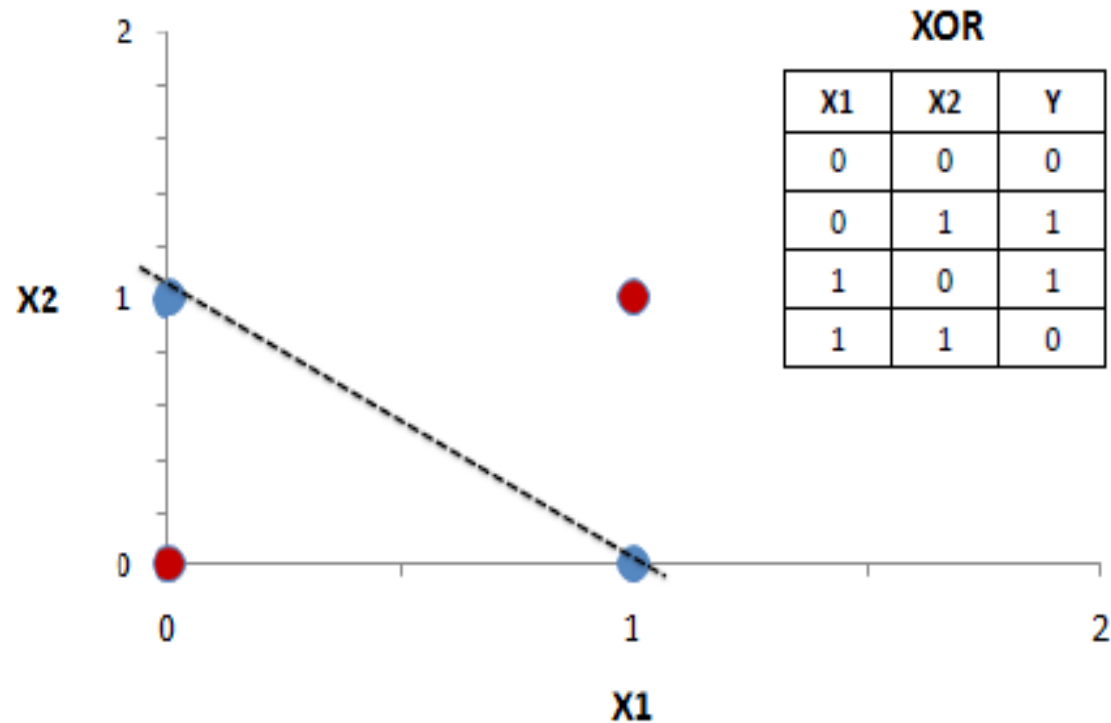
5.0 CONCLUSIONS:

Neural networks are a viable technology approach to classifying tanks from trucks using MMW profile signatures. Of particular significance is the demonstrated robustness to clutter corruption. Tanks and trucks were classified to ~80% confidence level even when high clutter was included. This level of performance held for all possible target aspects. Both MLP and LVQ networks exhibited comparable success.

Neural network implementation of 2-category target classification using high-range resolution polarimetric MMW radar, Pyka (WCNN'93, Portland)

Reality...

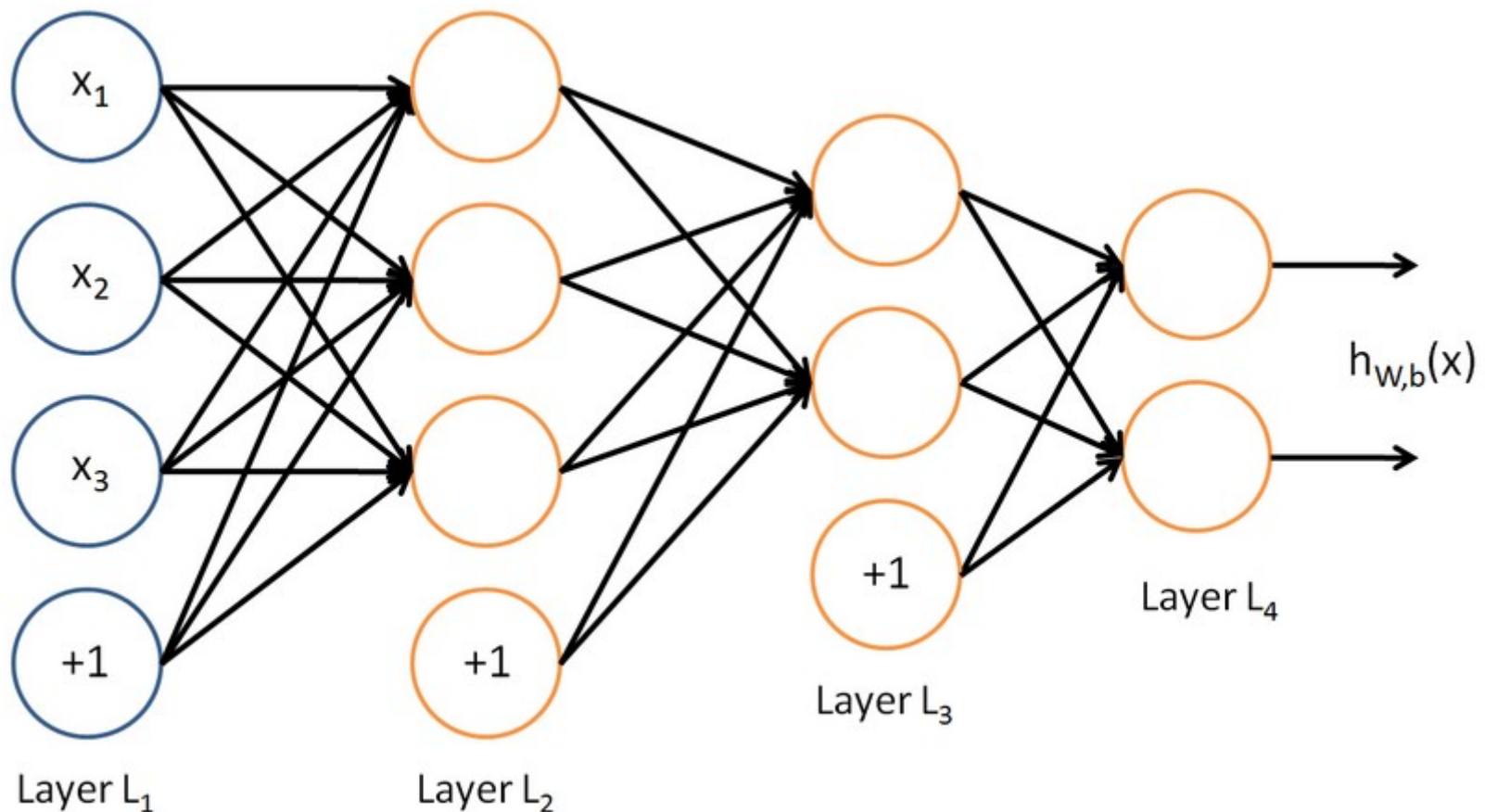
A single perceptron can model just linearly separable data. So, it cannot model even a simple operation such as a XOR. (Minsky & Papert 1969)



*It does not hold for multi layered architectures.

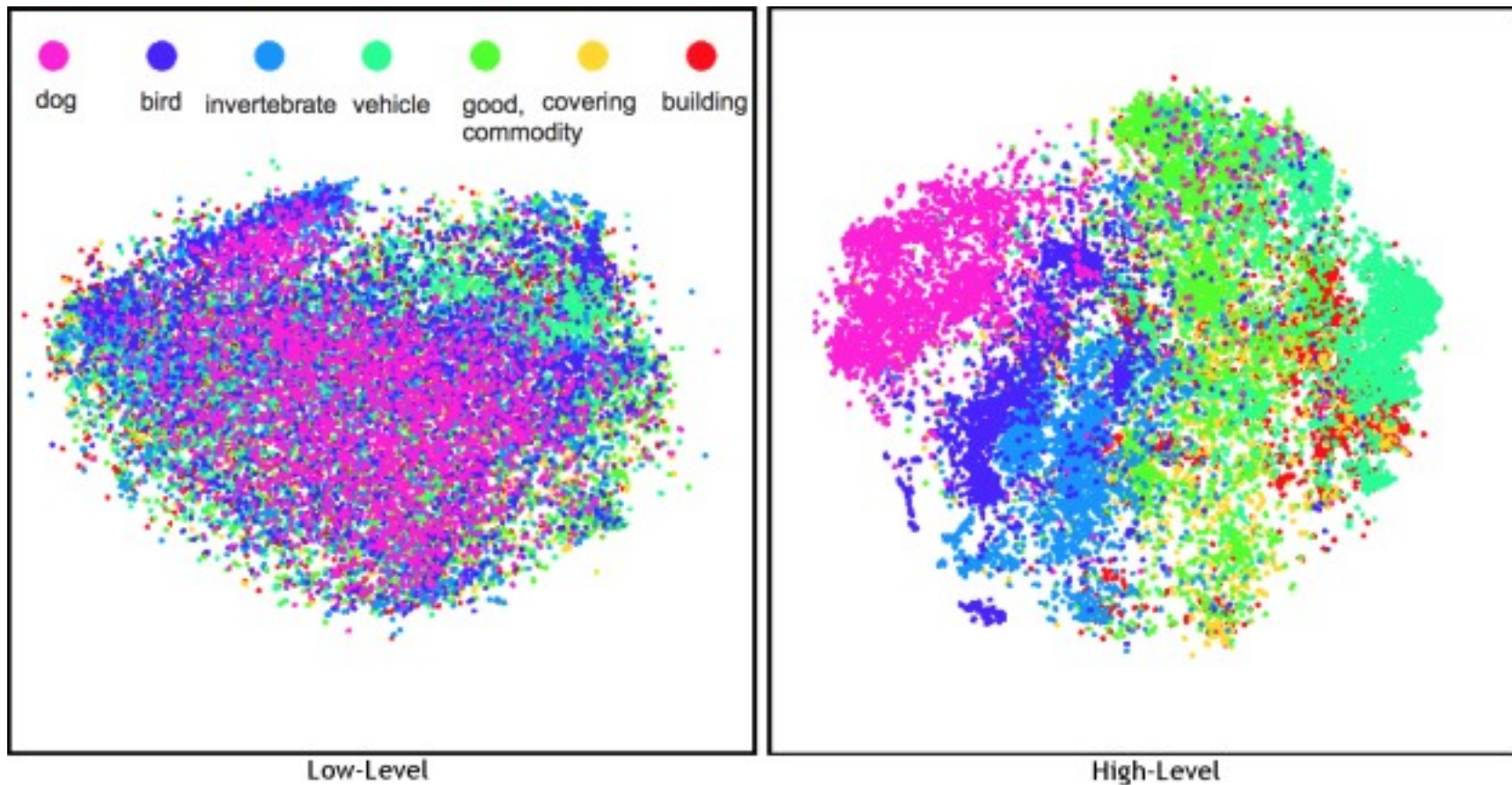
The basics: **Neural Networks (I)**

Definition: Algorithms that attempt to learn a hierarchical and **convenient** representation of data.



The basics: **Neural Networks (II)**

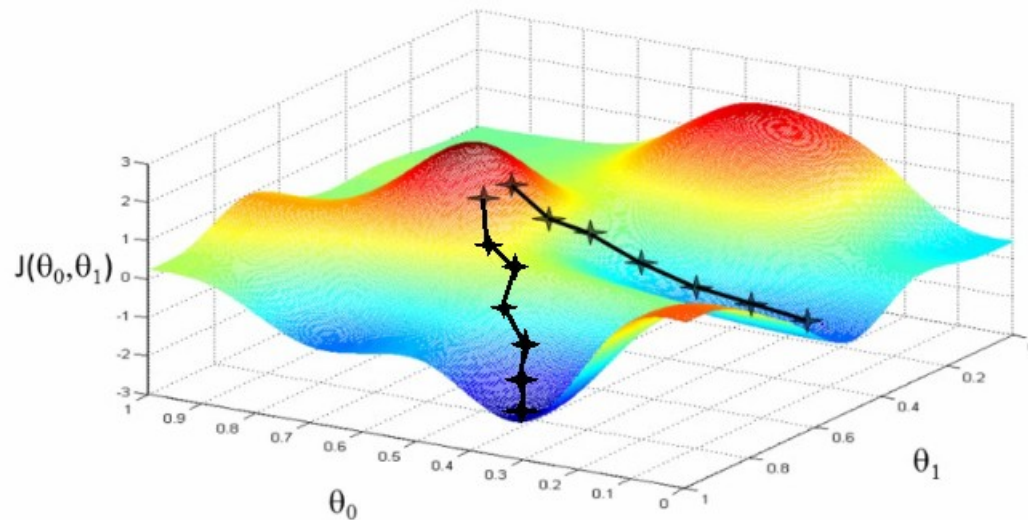
Why should you care about **learning representation**?



Making sense of data. Useful for **classification, clustering, samples generation.**

The basics: **Optimization (I)**

Definition: an optimization problem consists of **minimizing** (or maximizing) a real **function** by systematically choosing **input values** from within an allowed set and computing the value of the function.



The basics: **Optimization (II)**

- Is machine learning just optimization?

Optimization is just about finding **good minima**.

- This is just half part of the story.

Classification is about attaining **low test classification** error.

- Necessary to learn a function that **generalize** well to new unseen samples.

The basics: **cost function (or loss)**

Definition: the cost function is the function to be minimized by the optimization function. Example for classification: cost of the learned function output respect with training set labels.

$$J(W, b; x, y) = \frac{1}{2} \cdot \sqrt{\sum_{k=0}^N (h_{W,b}(x_k) - y_k)^2} + \frac{\lambda}{2} \cdot \sum_{l=1}^L \sum_{i=1} \sum_{j=1} w_{l,i,j}^2$$

Optimization is about:

$$W', b' = \operatorname{argmin}_{W, b} J(W, b; x, y)$$

The basics: **Weight Decay** for regularization



The basics: **Objective of learning**

$$E_{test} - E_{train} = k \cdot (h / P)^\alpha$$

k = constant

h = capacity

P = number of training samples

α = number between 0.5 and 1.0

“Statistical mechanics of learning from examples” Seung *et.al.*, 1992

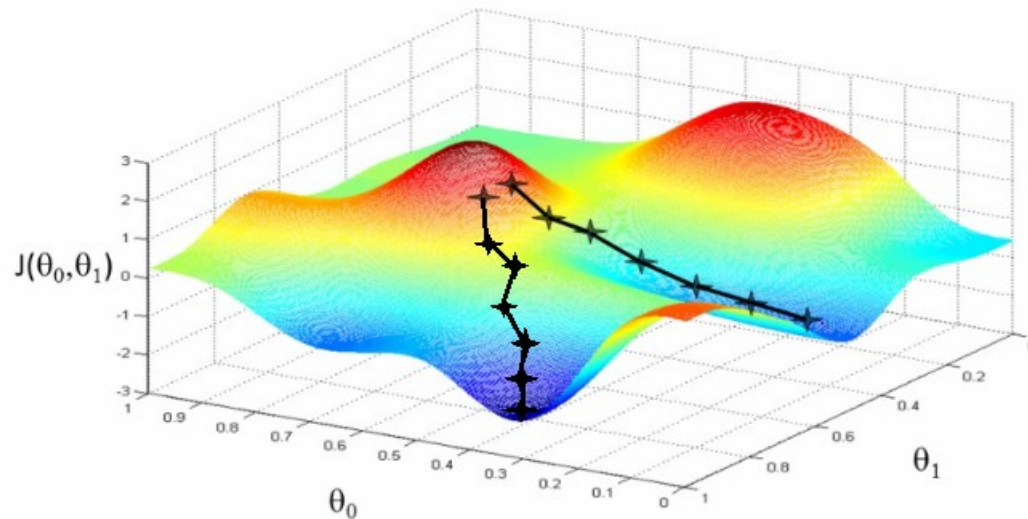
“Measuring the vc-dimension of a learning machine” Vapnik *et.al.*, 1994

“Learning curves: asymptotic values and rate of convergence” Cortes *et.al.*, 1994

The basics: **Gradient Descent**

W is iteratively adjusted according to:

$$W_k = W_{k-1} - \varepsilon \cdot \frac{\partial E(W)}{\partial W}$$



The basics: **Back-Propagation algorithm**

1) Perform a feedforward pass, computing the activations for layers L2, L3, up to the output layer Ln, using the equations defining the forward propagation steps.

2) For the Ln layer:

$$\delta_i^{(n_l)} = \frac{\partial}{\partial z_i^{(n_l)}} \frac{1}{2} \|y - h_{W,b}(x)\|^2 = -(y_i - a_i^{(n_l)}) \cdot f'(z_i^{(n_l)})$$

3) For $l = L-1, L-2, L-3, \dots, 2$:

$$\delta_i^{(l)} = \left(\sum_{j=1}^{s_{l+1}} W_{ji}^{(l)} \delta_j^{(l+1)} \right) f'(z_i^{(l)})$$

4) Compute the partial derivative:

$$\frac{\partial}{\partial W_{ij}^{(l)}} J(W, b; x, y) = a_j^{(l)} \delta_i^{(l+1)}$$
$$\frac{\partial}{\partial b_i^{(l)}} J(W, b; x, y) = \delta_i^{(l+1)}.$$

http://ufldl.stanford.edu/wiki/index.php/Backpropagation_Algorithm

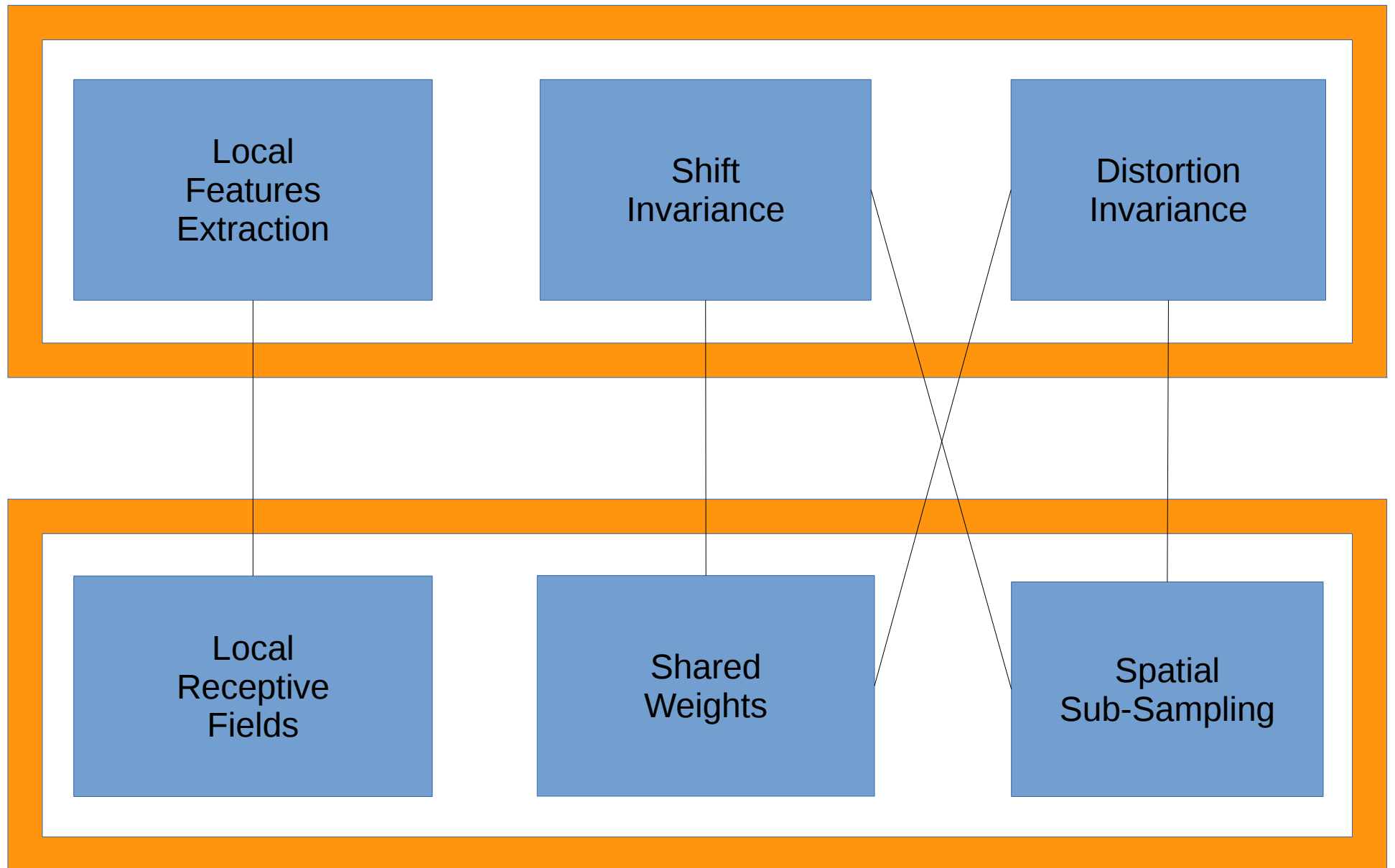
<http://karpathy.github.io/neuralnets/>

Typical fully connected layer

Flaws:

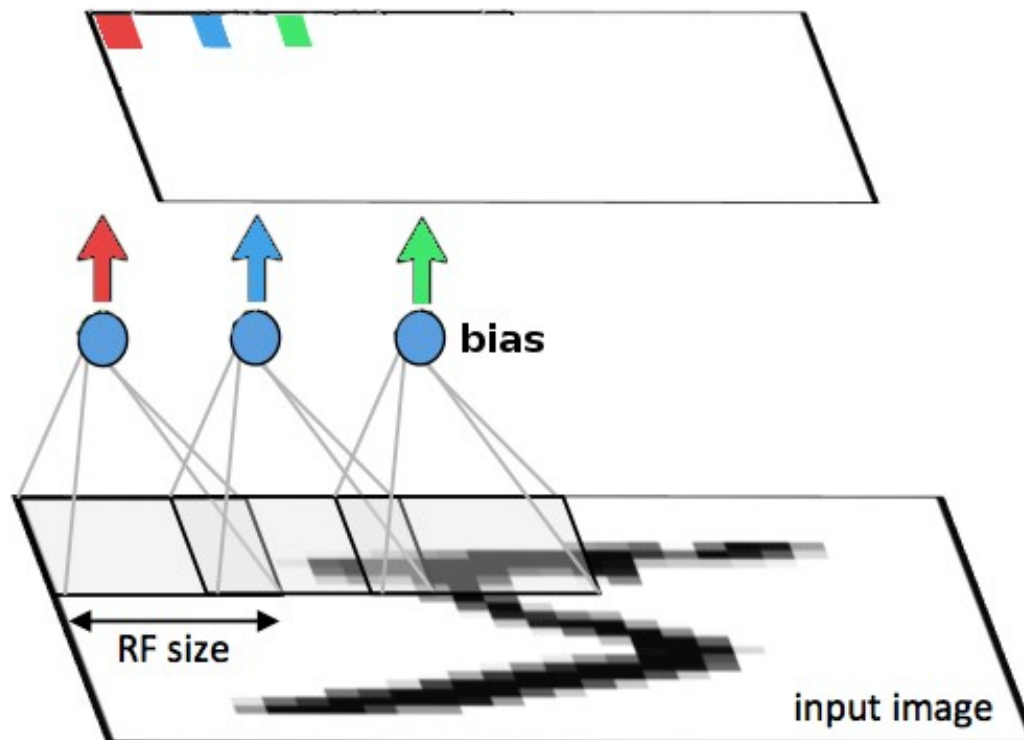
- Images contain thousand of pixels, 1st f.c. layer will contain many thousand weights
 - Prone to overfitting
 - Slower learning
- No built-in invariance to translation and local distortion of the input:
 - Ideally, this can be achieved by learning multiple units with similar weight in different locations;
- Prior structure of the input is completely ignored, images have highly spatially correlated 2D structure:
 - In f.c. layers input variables can be represented in any (fixed) order and the outcome does not change.

Ideal properties...



Local Receptive Fields

- Neurons can extract elementary visual features such as oriented edges, end-points, corners
- These features are then combined to sub-sequent layers in order to detect high order features



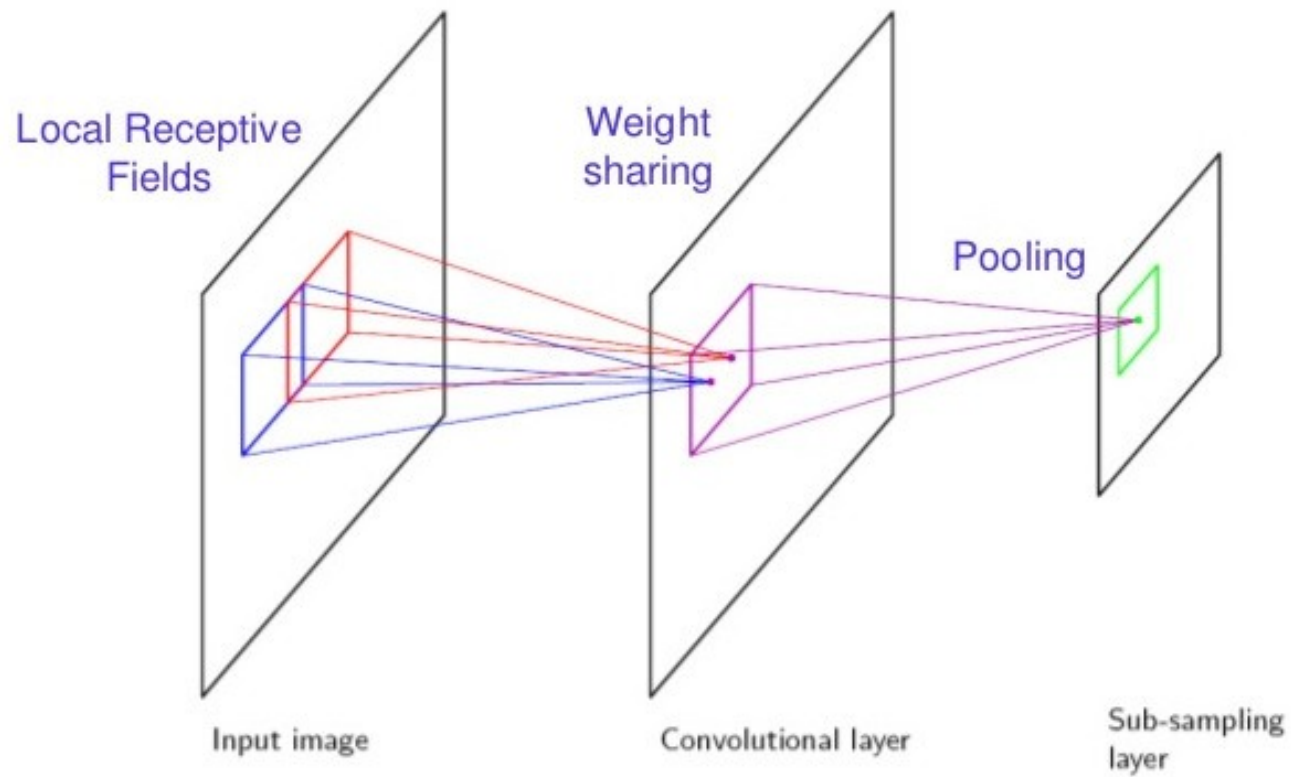
Shared weights

- Distortions or shifts may cause the position of salient features to vary;
- Elementary features detector are likely to be useful across the entire image;
- Feature map: a set of units whose receptive fields are constrained to have identical weights vectors;
- Adjacent local receptive fields overlap in the same features map;
- A set of features maps compose a convolutional layer.

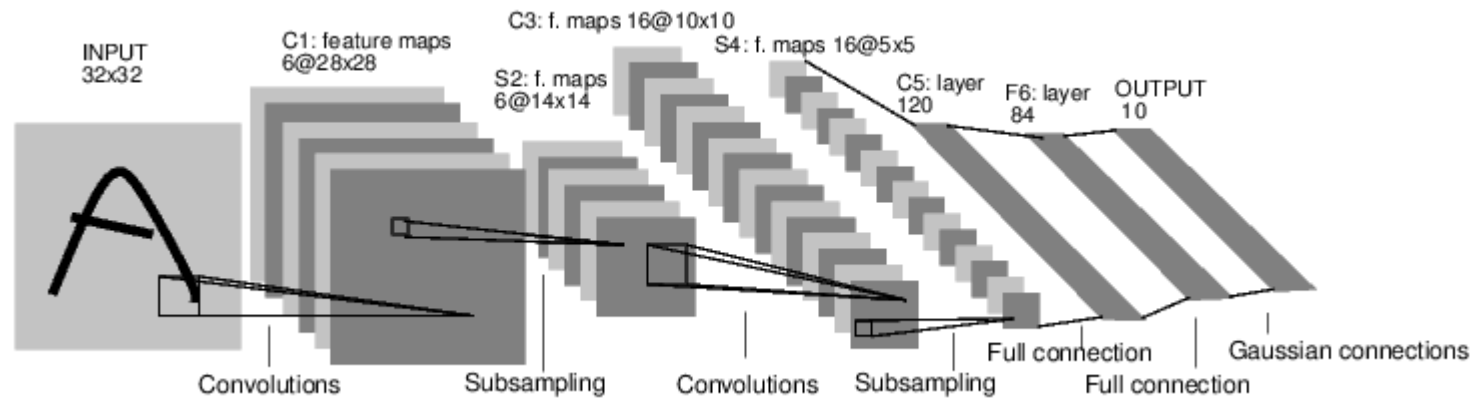
Pooling Layer (I)

- Once a feature has been detected, exact position can be harmful, because exact position can vary from instance to instance of the object;
- Approximate position relative to other features is relevant;
- Less parameters;
- Max-pooling act as a best features selector;

Pooling Layer (II)



LeNet 5



Stochastic Gradient

VS

Full-Batch Gradient

- Full-Batch Gradient: gradients are accumulated over the entire training set, exact gradient estimation;
- (Batch) Stochastic Gradient: a partial “noisy” gradient is evaluated on the basis of a single training sample, parameters are updated using the approximated gradient.

1. Jackel bets (one fancy dinner) that by March 14, 2000, people will understand quantitatively why big neural nets working on large databases are not so bad. (Understanding means that there will be clear conditions and bounds)

Vapnik bets (one fancy dinner) that Jackel is wrong.

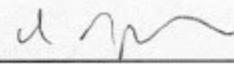
But .. If Vapnik figures out the bounds and conditions, Vapnik still wins the bet.

2. Vapnik bets (one fancy dinner) that by March 14, 2005, no one in his right mind will use neural nets that are essentially like those used in 1995.


Jackel bets (one fancy dinner) that Vapnik is wrong



V. Vapnik 3/14/95



L. Jackel 3/14/95



Witnessed by Y. LeCun 3/14/95

Beside MNIST, training deep ConvNets was fairly hard

- Training requires a lot of computation and data, too much for the computational power available back then;
- Vanishing gradients;
- Poor weights initialization scheme and “noisy” estimated updates with batch gradient descent easily get stuck in poor local minima;
- Full-batch is feasible just on small datasets...

Unsupervised pre-training...

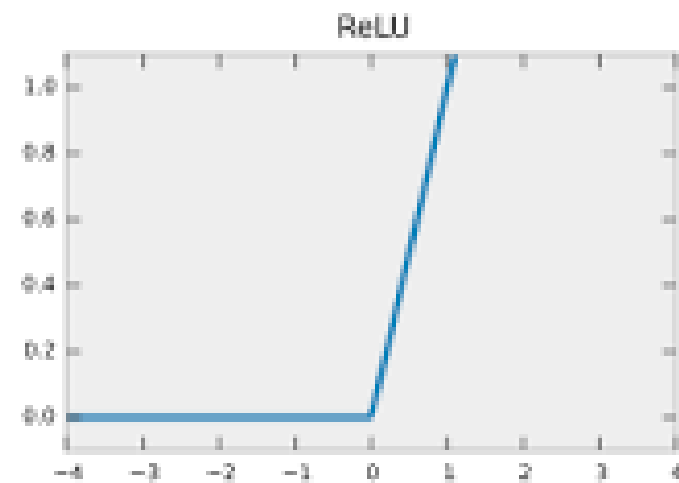
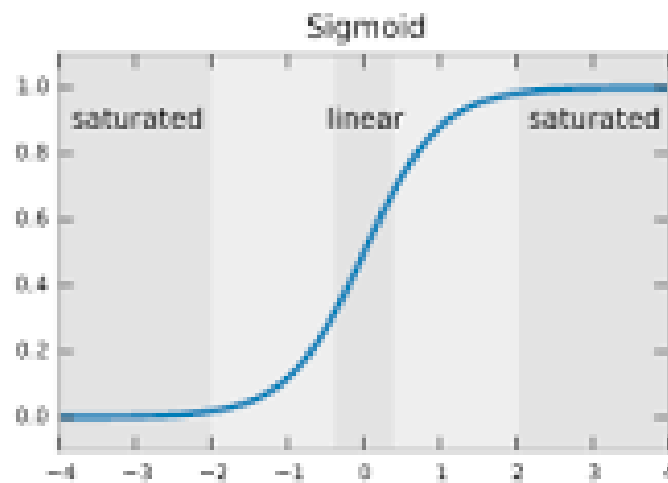
“A Fast Learning Algorithm for Deep Belief Nets”, Hinton & Osindero, 2006

Basically, initialization of weights for the supervised phase by means of stacked Restricted Boltzmann machines...

Expensive, but it showed that training deep architecture is possible!

Rectified Linear Units

“Rectified Linear Units Improve Restricted Boltzmann Machines”, Nair, Hinton, 2010



Momentum

“On the importance of initialization and momentum in deep learning”, *Sutskever, Martens, Dahl, Hinton*, 2011

$$V_{t+1} = \mu \cdot V_t - \alpha \cdot \varepsilon \cdot \frac{\partial E(W)}{\partial W}$$

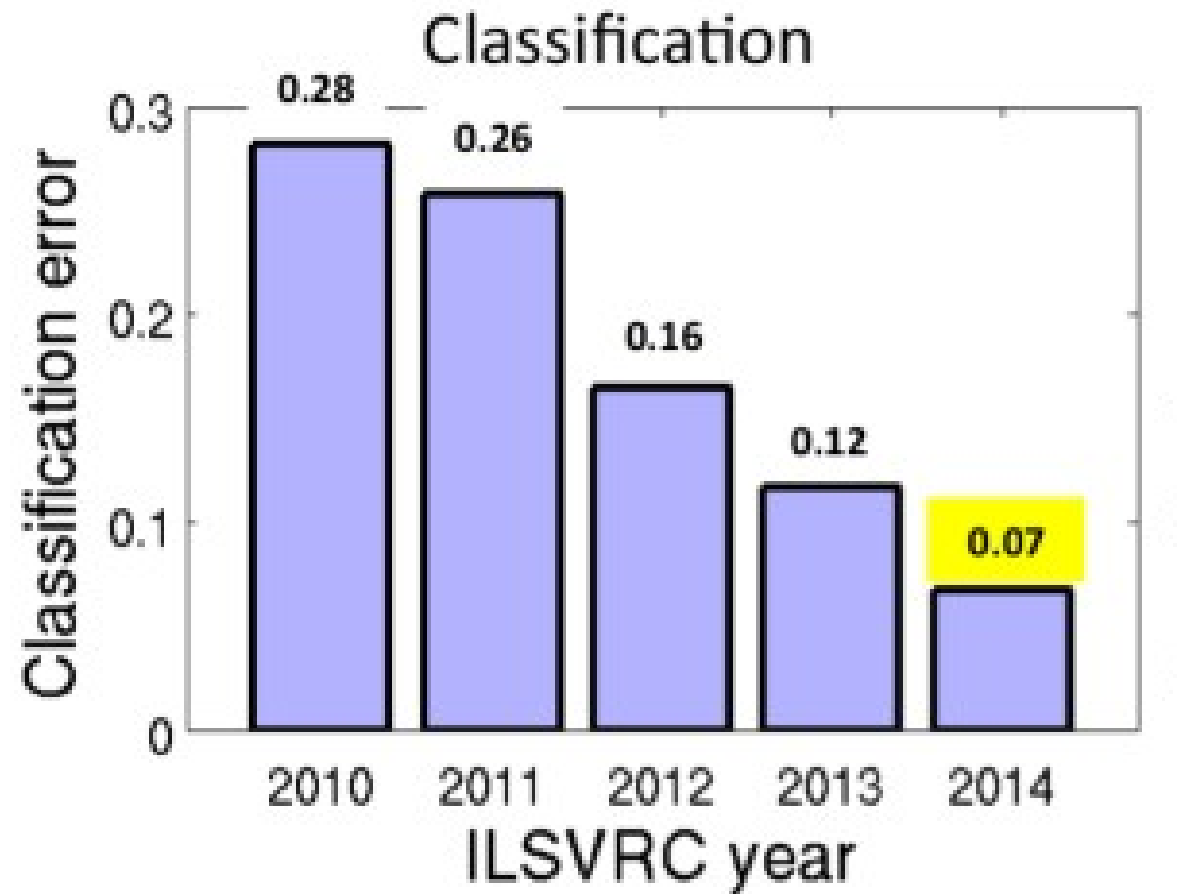
$$W_{t+1} = W_t + v_{t+1}$$

Weights initialization

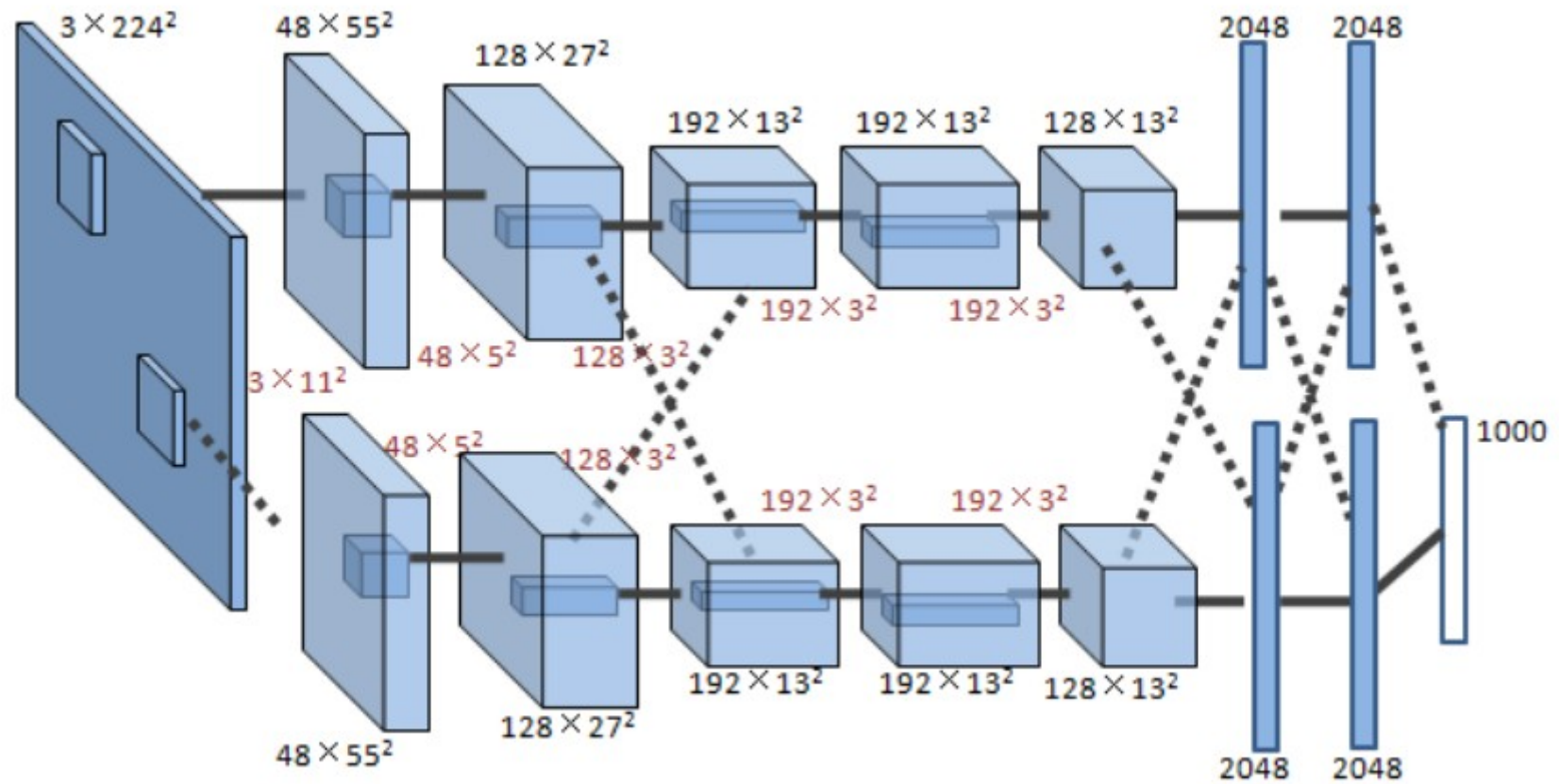
“Understanding the difficulty of training deep feedforward neural networks”, *Glorot, Bengio, 2011*

$$w_{i,j,l} = U\left[-6 \frac{\text{fan}_{\text{input}} + \text{fan}_{\text{out}}}{2}, +6 \frac{\text{fan}_{\text{input}} + \text{fan}_{\text{out}}}{2}\right]$$

From ILSVC 2012...



AlexNet



Hand-crafted vs Learned Representations



Traditional Pattern Recognition: Fixed/Handcrafted Feature Extractor



Mainstream Modern Pattern Recognition: Unsupervised mid-level features



Deep Learning: Representations are hierarchical and trained



Credit to Yann LeCun for the slide

Hand-crafted **shallow** HoG/SIFT representation (I)

What is this?

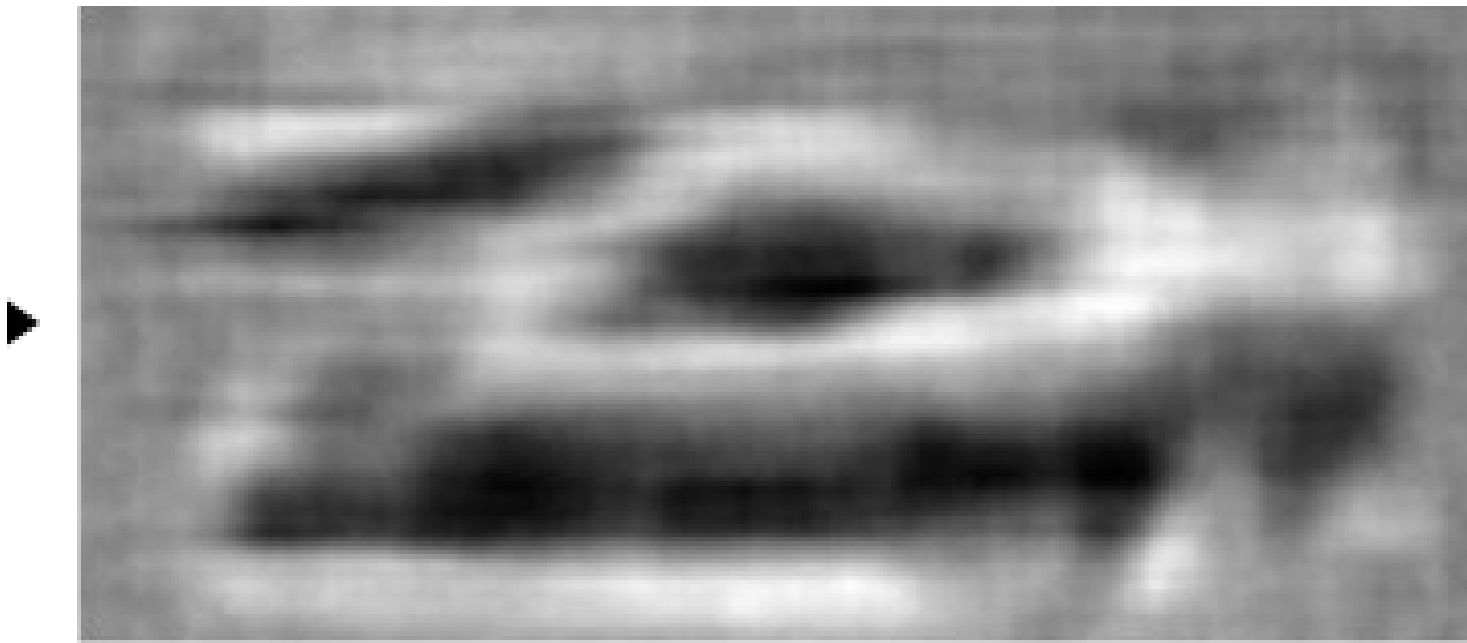


Hand-crafted **shallow** HoG/SIFT representation (II)



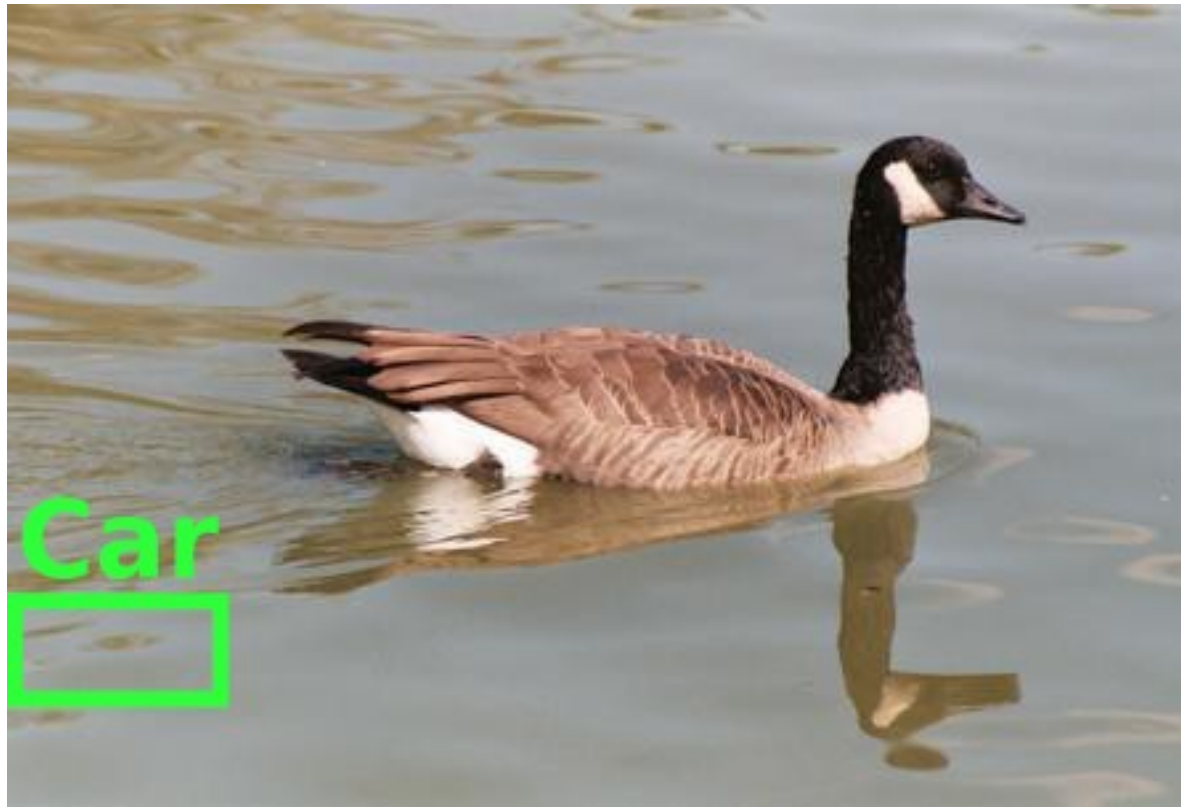
Hand-crafted **shallow** HoG/SIFT representation (III)

What is this?



Hand-crafted **shallow** HoG/SIFT representation (III)

Not really...



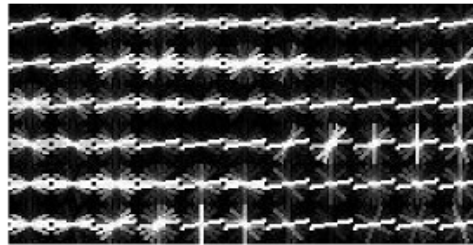
C. Vondrick, A. Khosla, T. Malisiewicz, A. Torralba. "HOGgles: Visualizing Object Detection Features" ICCV 2013

Hand-crafted **shallow** HoG/SIFT representation (III)

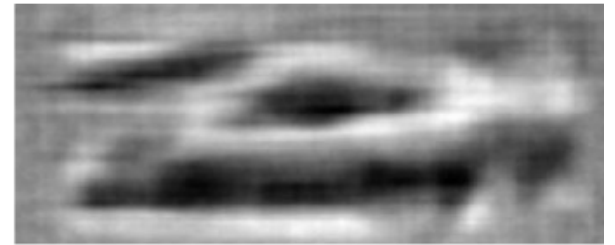
Low-level cue representation that fails to capture effectively high level abstractions.



Car Detection



HOG Features



Our Visualization

Conversely, deep learned features..



Face detector



Human body detector



Cat detector

Thank you!

Any questions?