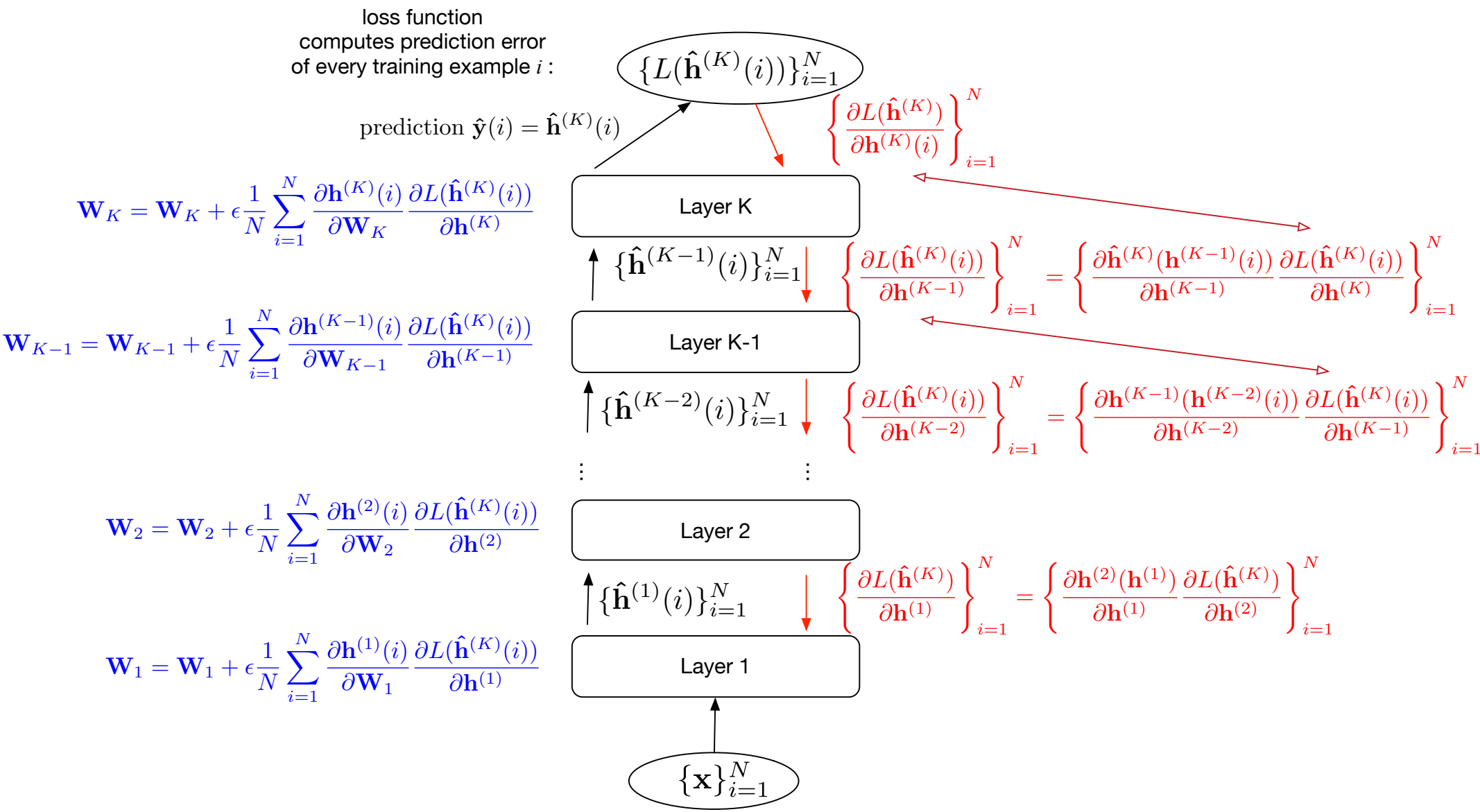
Data Mining & Machine Learning

CS57300
Purdue University

March 6, 2018

(Review) Forward & Backward Updates for Gradient Descent



Training Neural Networks

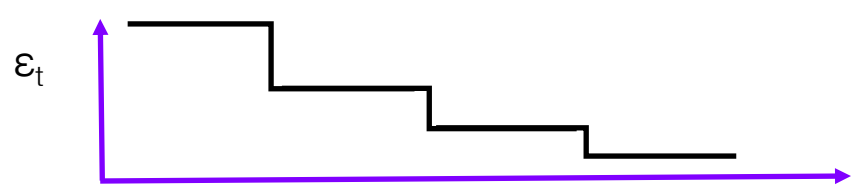
- Stochastic gradient descent (SGD) is important (mini-batch training)
- Learning rates are also important:
  - With constant learning rate ε, Neural Net (NN) gradient descent will likely not converge
  - Solutions:

    (1) SGD is a Robbins-Monro stochastic optimization method

    - Robbins-Monro approximations requires a decreasing learning rate. At step t learning rate is ε_t s.t.

$$\sum_t \epsilon_t = \infty \quad , \quad \sum_t \epsilon_t^2 < \infty$$

    - Works better with *plateaus* (keep ε_t constant for a while):



Reason: Markov chain convergence

    (2) Early stopping

    - Stop gradient descent with validation data…
      …if during training validation error keeps increasing, stop

$$\mathbf{W}_K = \mathbf{W}_K + \epsilon \frac{1}{N} \sum_{i=1}^{N} \frac{\partial \mathbf{h}^{(K)}(i)}{\partial \mathbf{W}_K} \frac{\partial L(\hat{\mathbf{h}}^{(K)}(i))}{\partial \mathbf{h}^{(K)}}$$

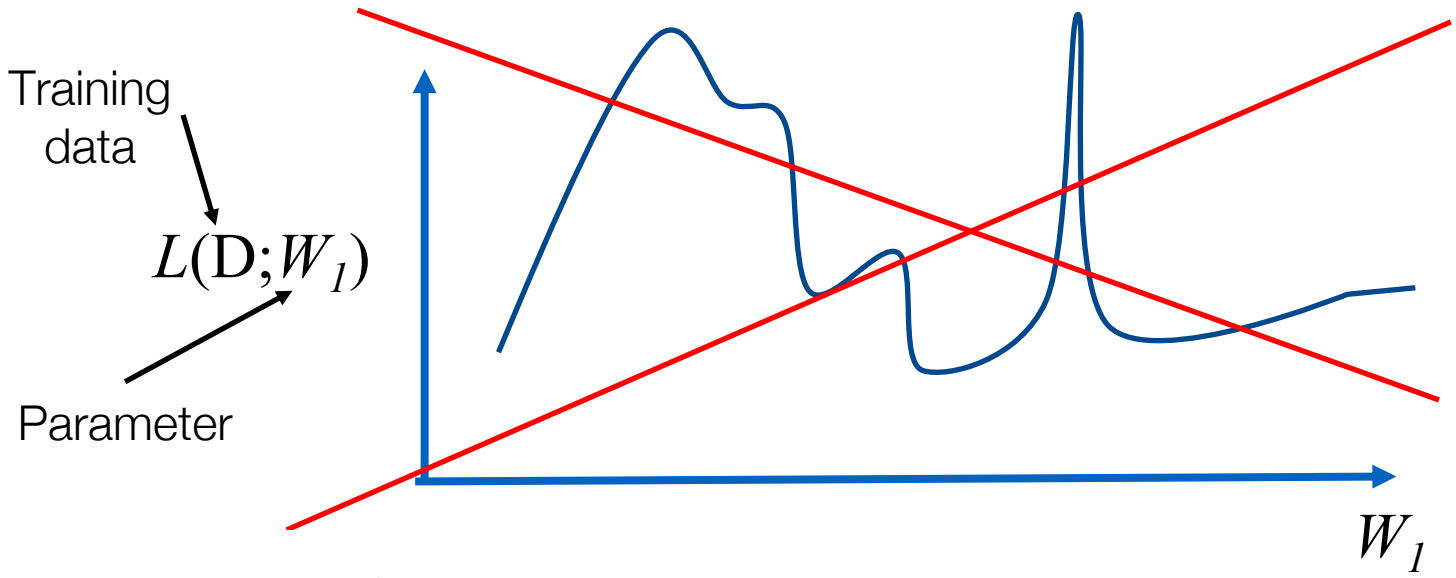
$$\mathbf{W}_{K-1} = \mathbf{W}_{K-1} + \epsilon \frac{1}{N} \sum_{i=1}^{N} \frac{\partial \mathbf{h}^{(K-1)}(i)}{\partial \mathbf{W}_{K-1}} \frac{\partial L(\hat{\mathbf{h}}^{(K)}(i))}{\partial \mathbf{h}^{(K-1)}}$$

$$\vdots$$

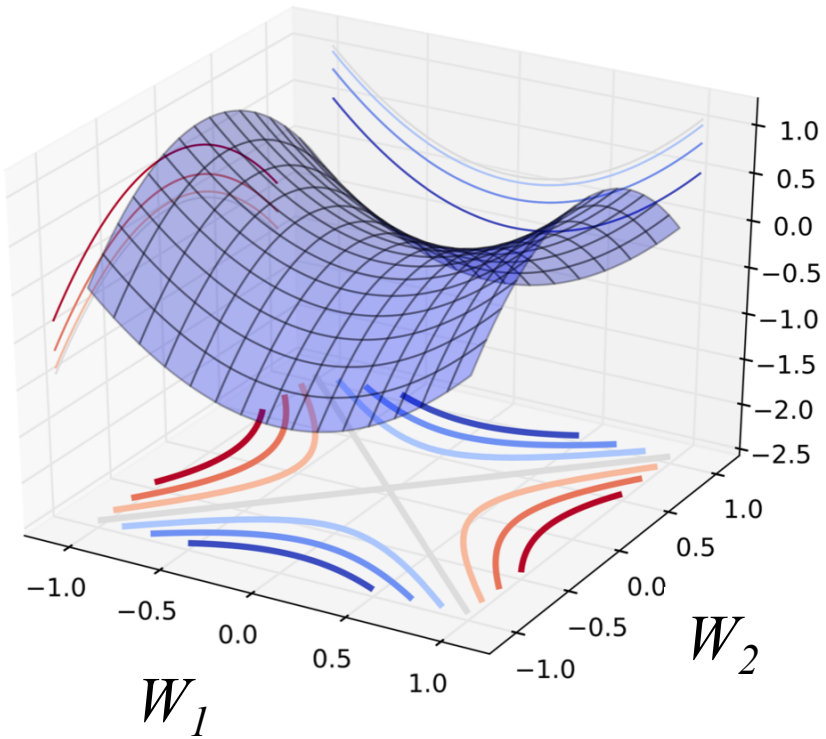
$$\mathbf{W}_2 = \mathbf{W}_2 + \epsilon \frac{1}{N} \sum_{i=1}^{N} \frac{\partial \mathbf{h}^{(2)}(i)}{\partial \mathbf{W}_2} \frac{\partial L(\hat{\mathbf{h}}^{(K)}(i))}{\partial \mathbf{h}^{(2)}}$$

$$\mathbf{W}_1 = \mathbf{W}_1 + \epsilon \frac{1}{N} \sum_{i=1}^{N} \frac{\partial \mathbf{h}^{(1)}(i)}{\partial \mathbf{W}_1} \frac{\partial L(\hat{\mathbf{h}}^{(K)}(i))}{\partial \mathbf{h}^{(1)}}$$

NN Score Functions are Non-convex



Not quite this



More like a lot
of saddle points...

Identifying and attacking the saddle point problem in high-dimensional non-convex optimization
Dauphin et al., NIPS 2014

Partial ack: Hugo Larochelle

Model Search Needs Wiggle Room…

- If dataset is created by labeling data using *K*-hidden units NN
  - Training another *K*-hidden units NN will likely fail
  - But training a K'-hidden unit NN may work, K' > K

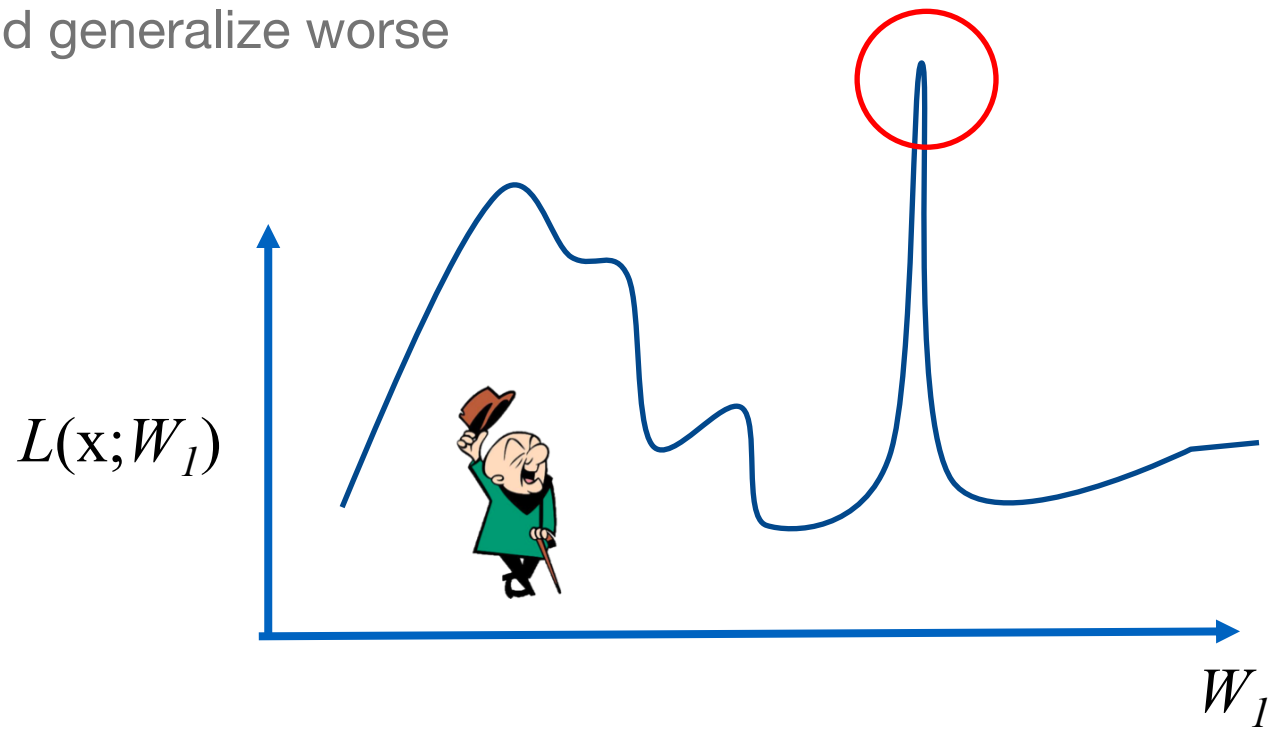
- But once trained, they can be compressed
  - Hinton, Geoffrey, Oriol Vinyals, and Jeff Dean. "Distilling the knowledge in a neural network," NIPS Deep Learning and Representation Learning Workshop, 2014

- As a corollary to the first point, first train a larger / deeper NN, then compress
  - *Do Deep Nets Really Need to be Deep?* Jimmy Ba, Rich Caruana, NIPS 2014

Partial ack: Hugo Larochelle

Training Works Best When Model Search is Bad at Its Job

- *On Large-Batch Training for Deep Learning: Generalization Gap and Sharp Minima* Keskar et al., ICLR 2017
  - Using large batch sizes tends to find sharp maximum and generalize worse

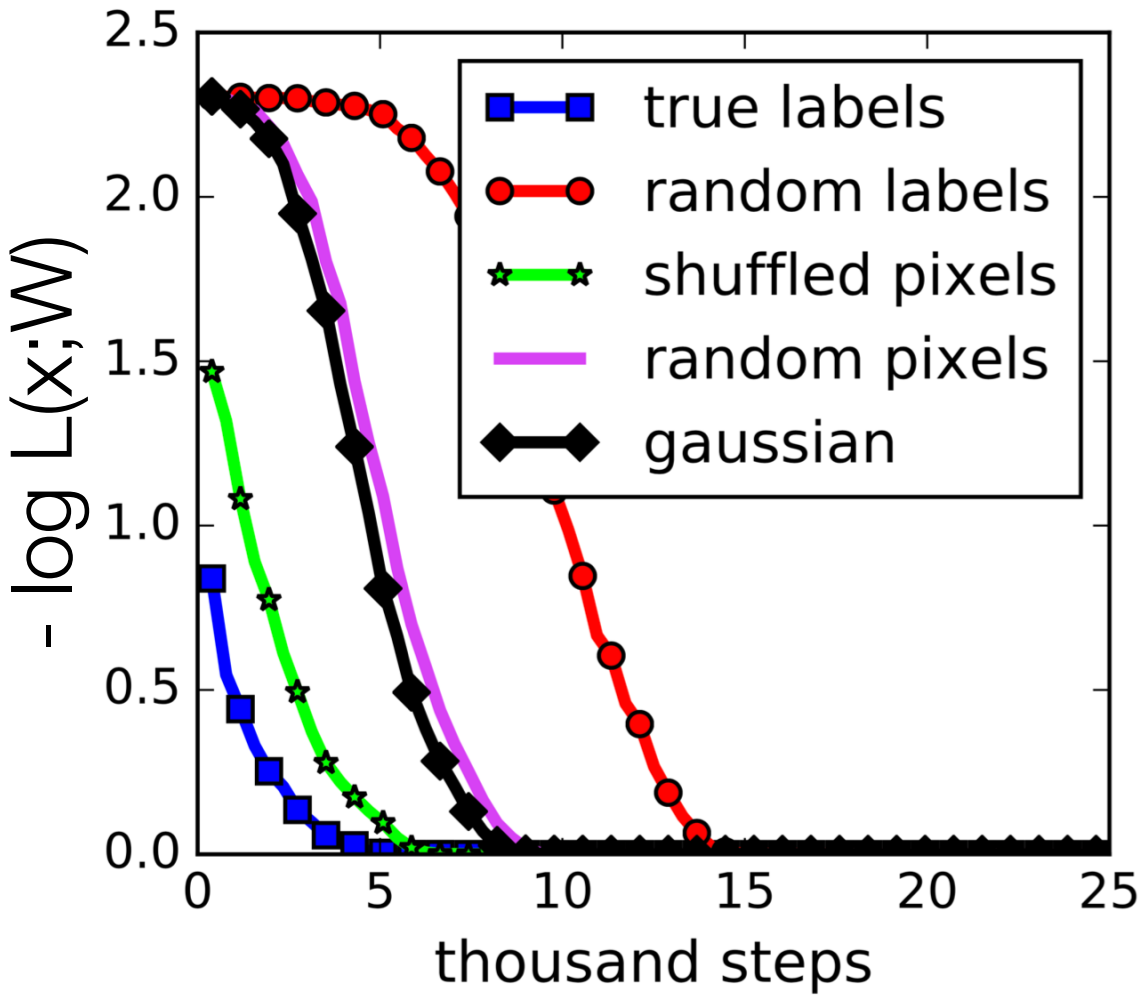


  - This means that Model Search is part of the design decision, it will not generalize well without taking the training algorithm into account

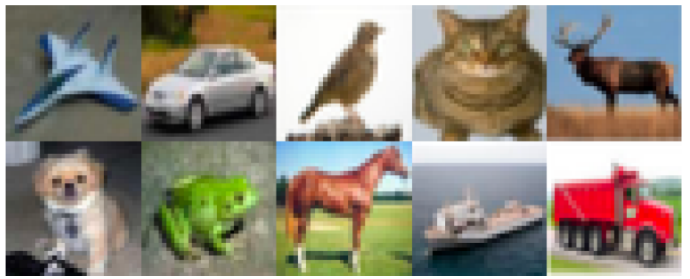
Partial ack: Hugo Larochelle

They Can Memorize Random Labels

- *Understanding Deep Learning Requires Rethinking Generalization*
  Zhang et al., ICLR 2017



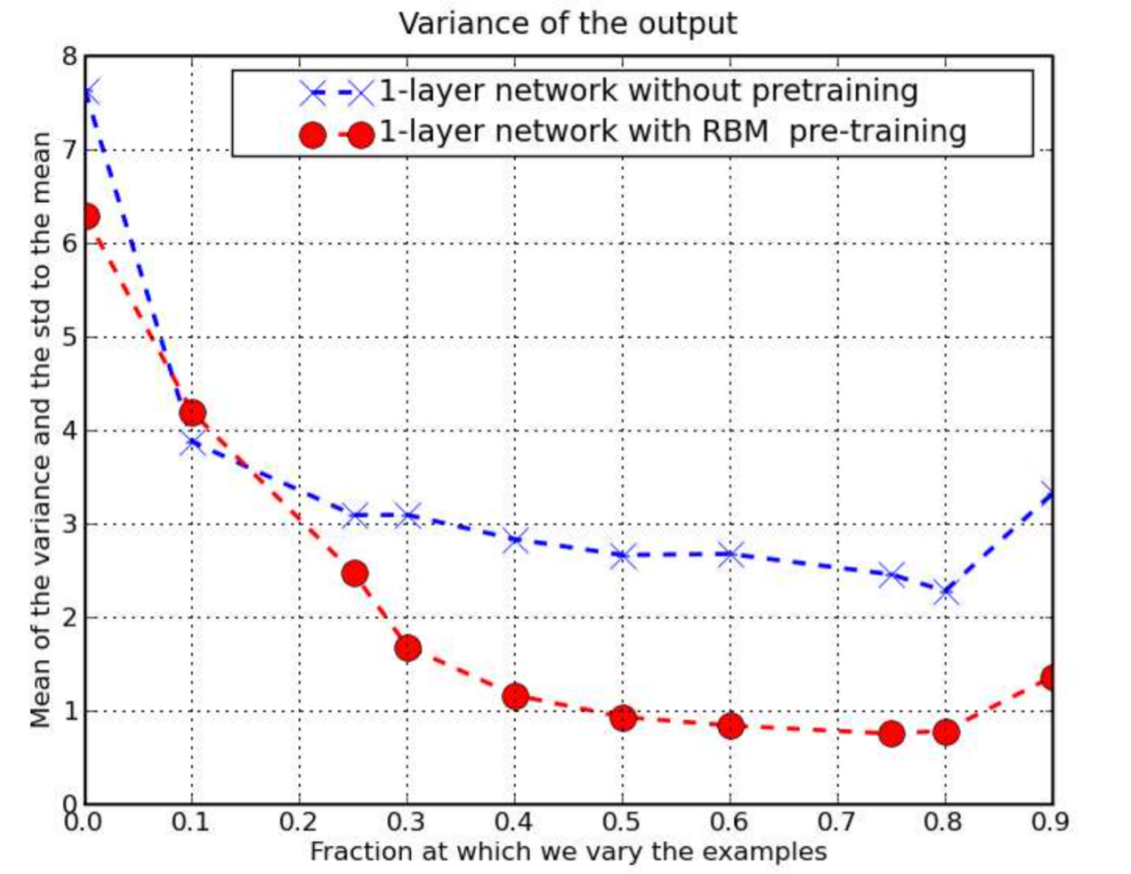
Task CIFAR10:
Label pictures into
10 categories



Partial ack: Hugo Larochelle

Search Is Influenced by Initialization & First Examples

- *Why Does Unsupervised Pre-Training Help Deep Learning*, Erhan et al., JMLR 2010



Partial ack: Hugo Larochelle

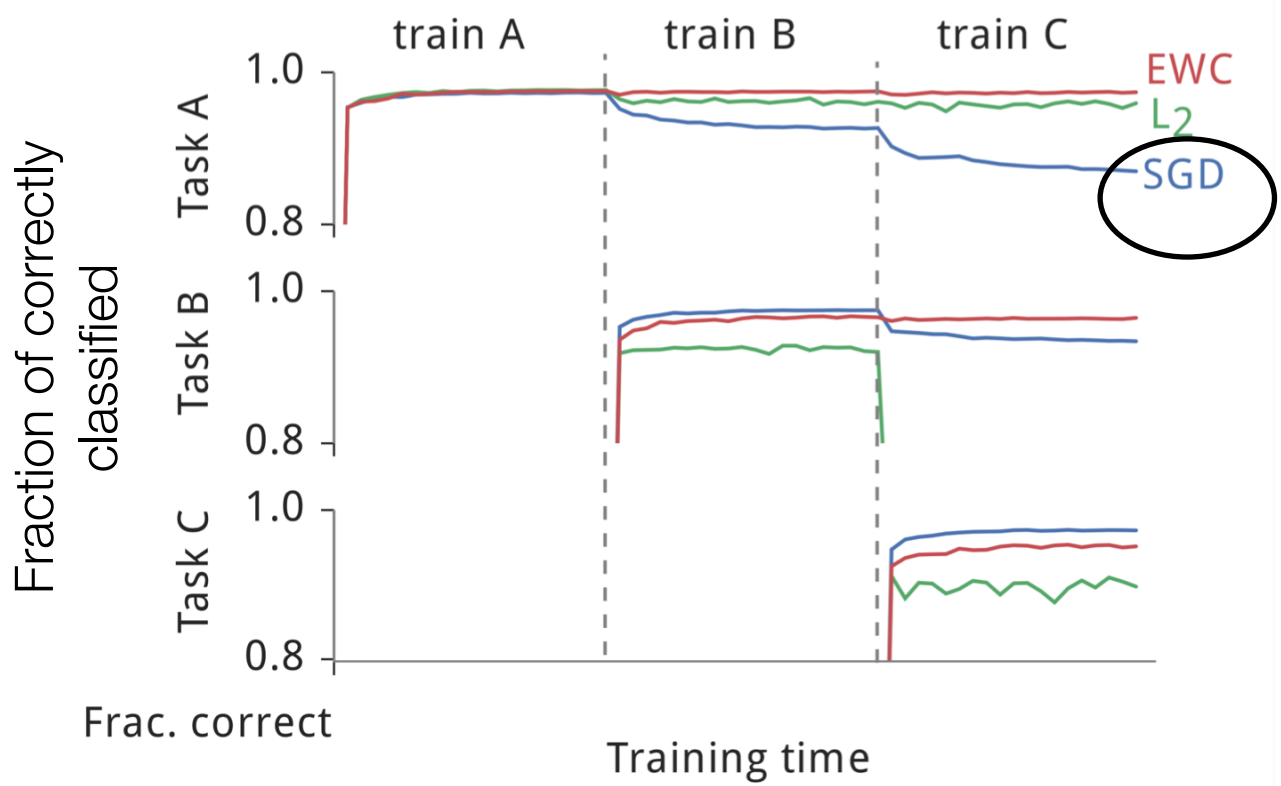
But They also Forget Quickly

- *Overcoming Catastrophic Forgetting in Neural Networks,* Kirkpatrick et al. PNAS 2017

- Consider dividing training into 3 distinct tasks for CIFAR10 (just for illustration):
  A = (boats, airplanes)
  B = (cats, dogs)
  C = (remaining classes)

- Use the Weights of last task to initialize new task

- Pay attention to SGD, blue curve



Partial ack: Hugo Larochelle

Real-World Example

- In what follows we will have a crash-course on Convolutional Neural Networks

- Goal is to use them in a real-life experiment

Convolutional Neural Networks (Motivation)

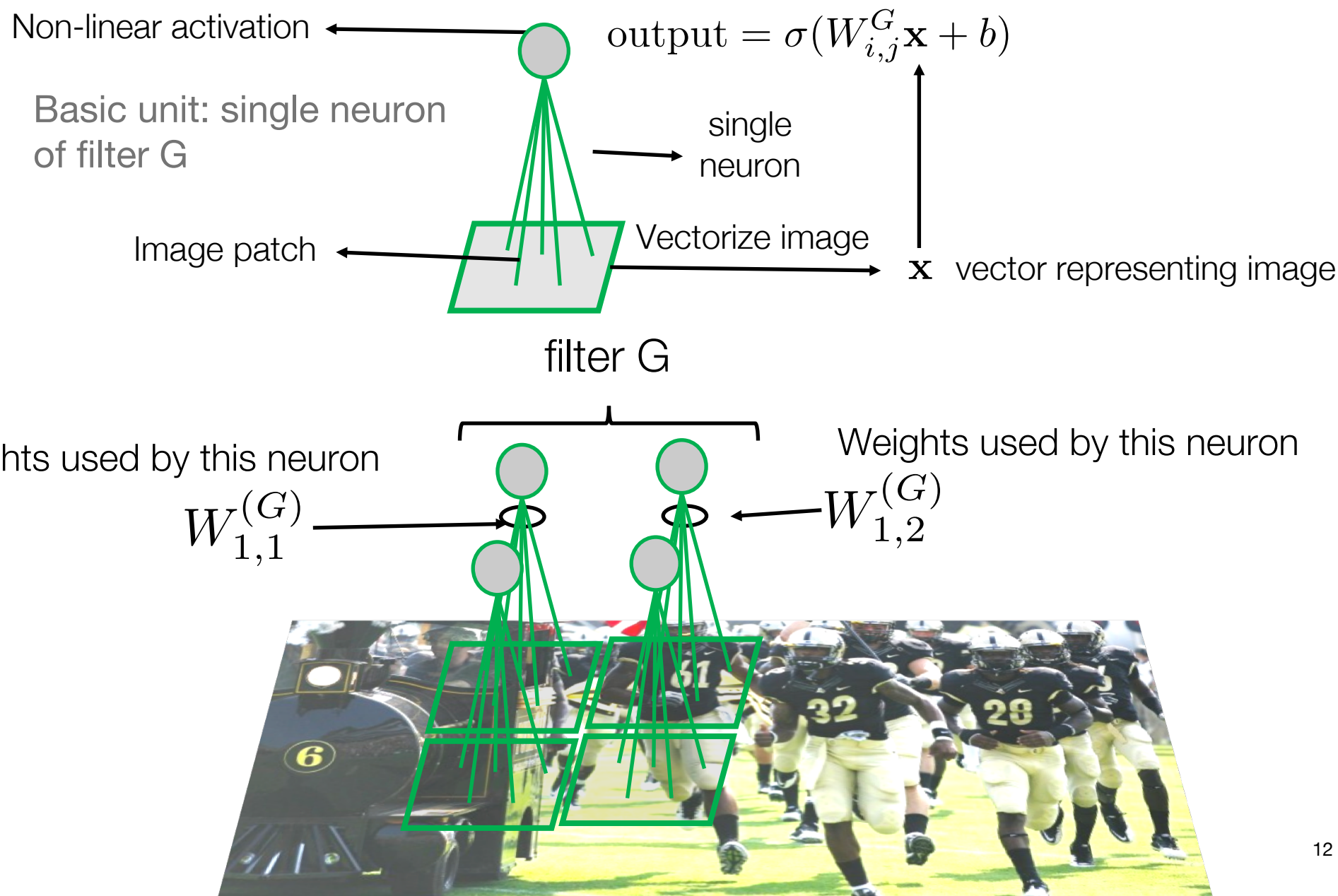
- Feedforward networks use image as input as a vector

- From image to vector… hard to account for spatial correlations in the vector representation (which pixels are next to each other?)

- Even harder to account for location and colors

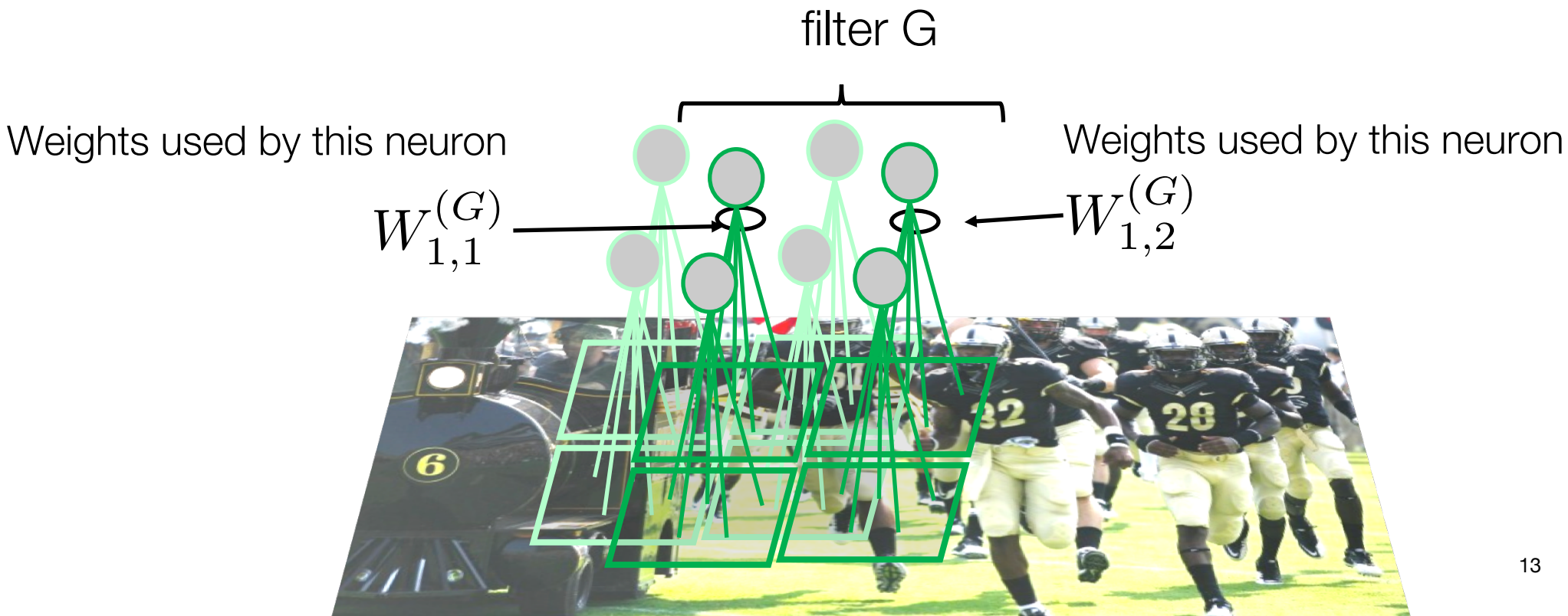
Convolutional Neural Network (CNN)

Transforms large image patch into one output



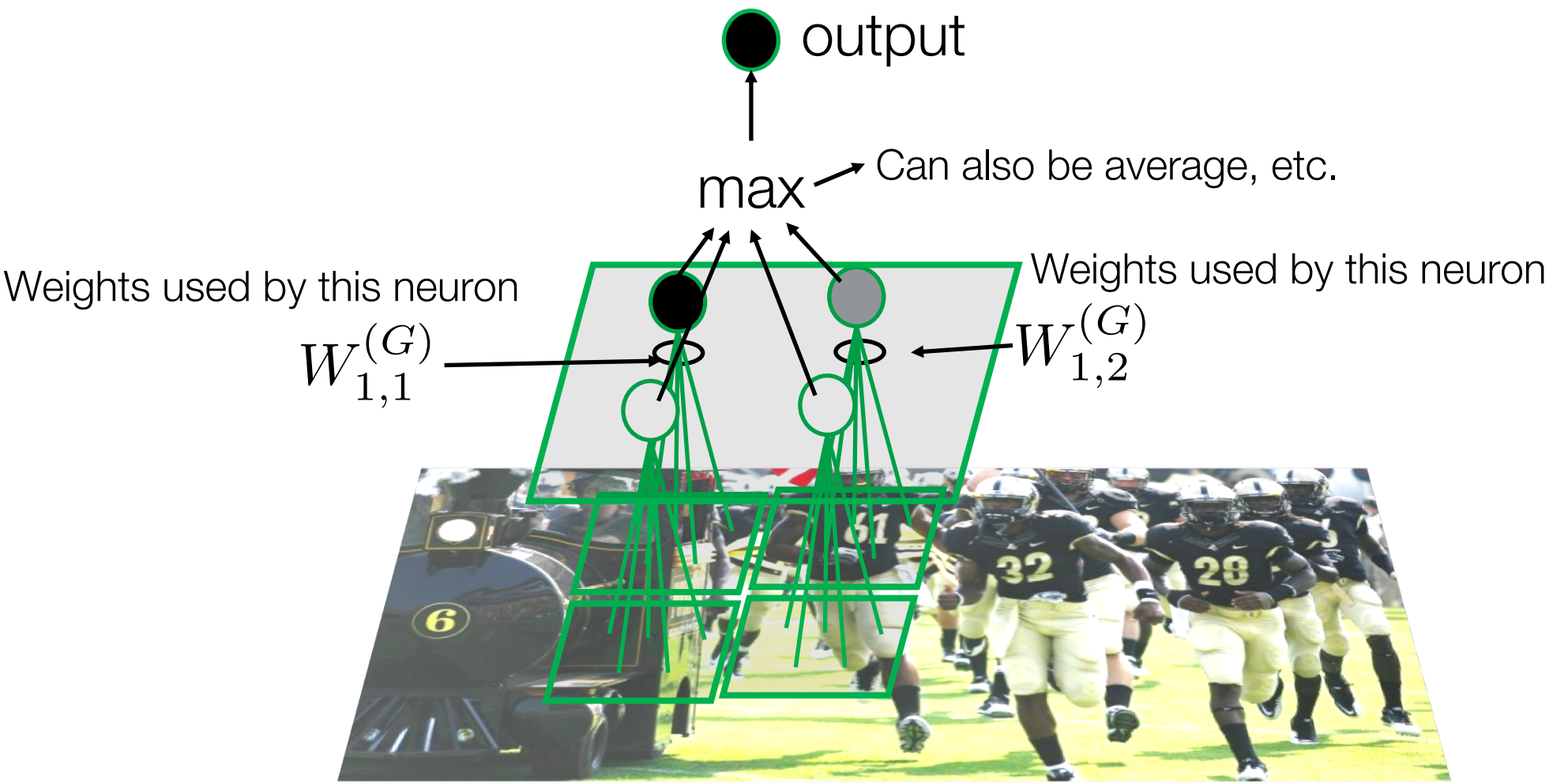
Convolutional Neural Network (CNN)

Cover the rest of the image by sliding the filter



Pooling

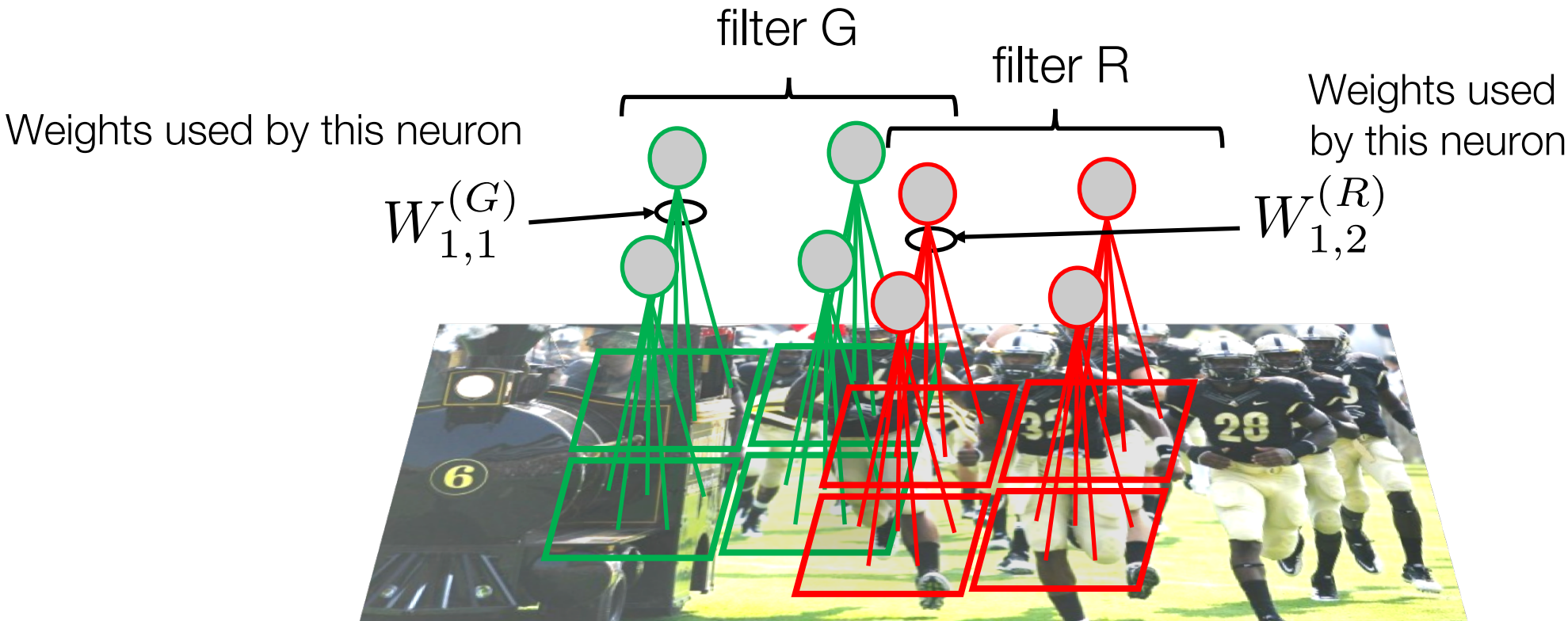
- Max pooling is a way to get a single output out of a filter



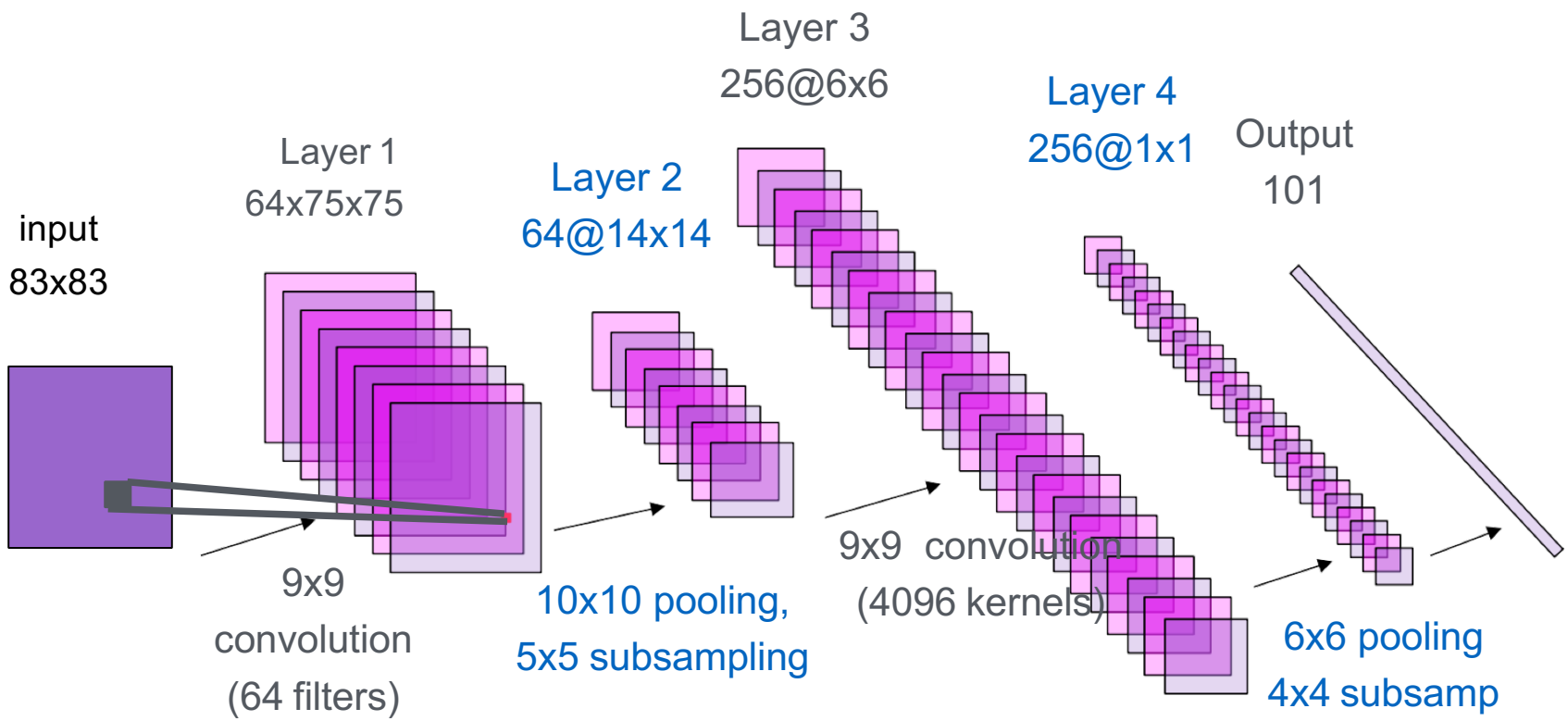
How to represent the image with different colors?

Convolutional Neural Network (CNN)

- We can use other filters
- For instance, one for each color
- We can also use more than one filter for each color



Convolutional Network (ConvNet)

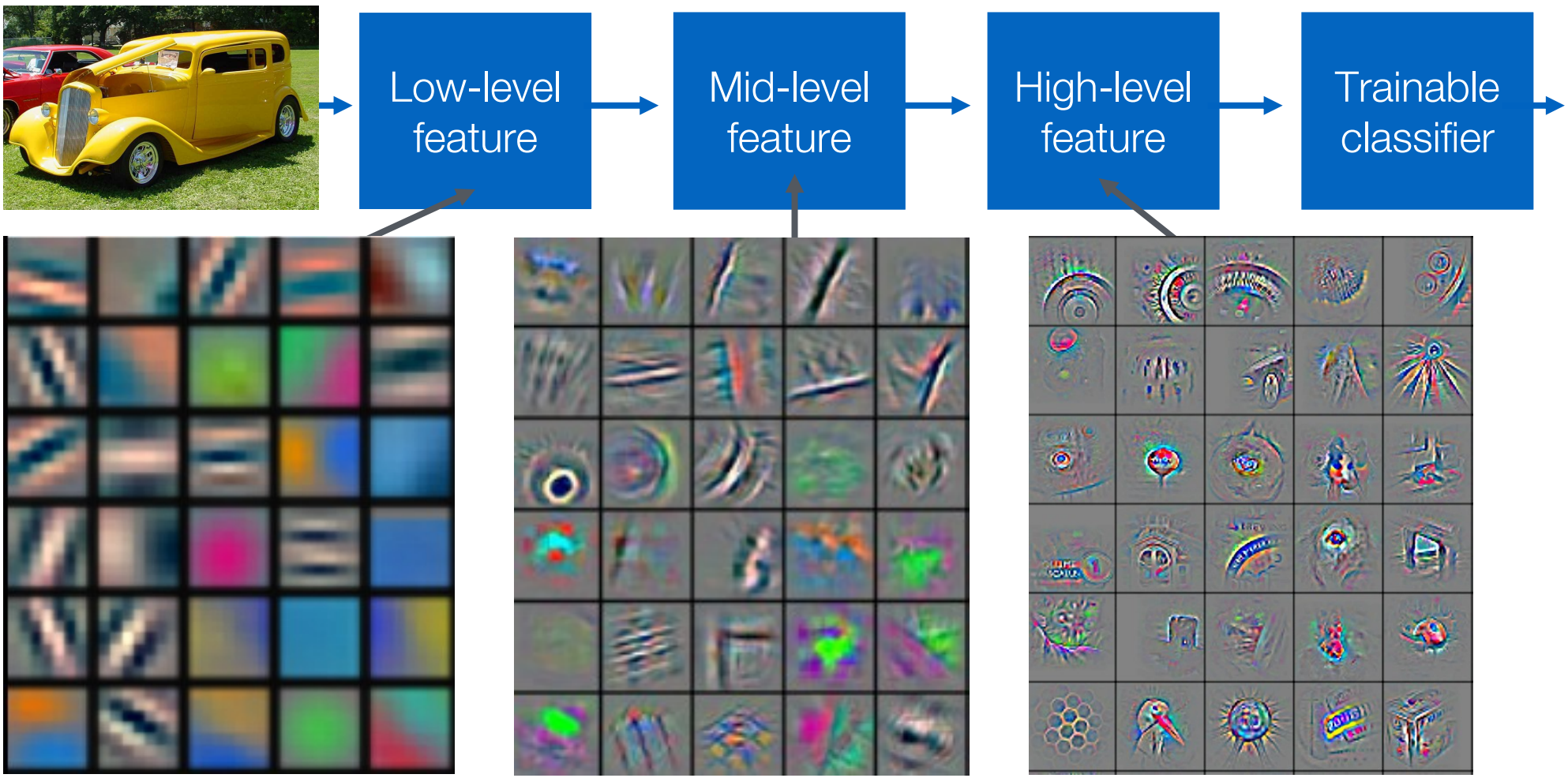


- Non-Linearity: sigmoid, rectified linear units (ReLU)
- Pooling: max, average,…
- Training: Image labels

Original: Y. LeCun, Facebook, modified by Ribeiro

Deep learning = learning hierarchical representations

It's deep if it has more than one stage of non-linear feature transformation



Feature visualization of convolutional net trained on ImageNet from [Zeiler & Fergus 2013]

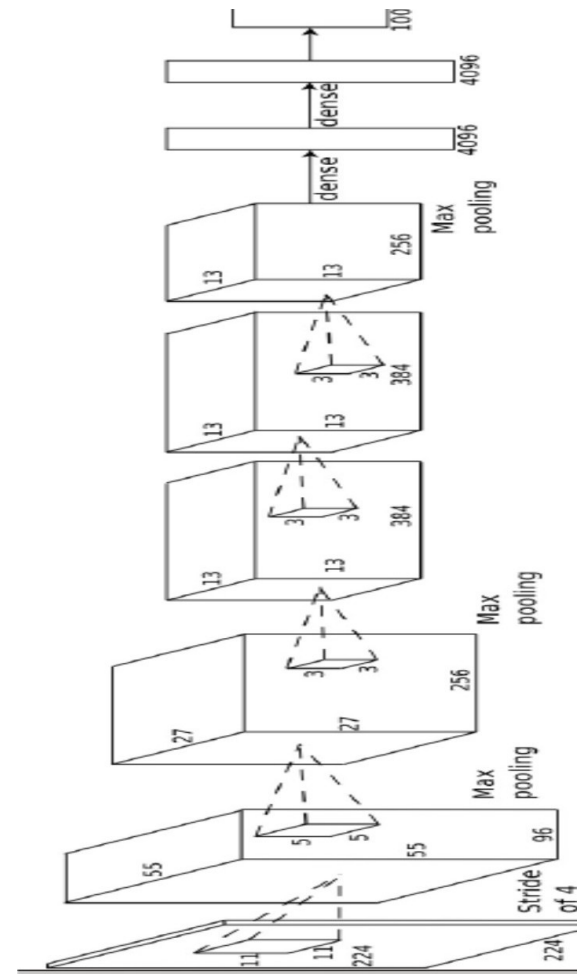
Credit: Y. LeCun, Facebook

Applications

Object Recognition [Krizhevsky, Sutskever, Hinton 2012]

– Won the 2012 ImageNet LSVRC. 60 Million parameters, 832M Matrix multiplication + accumulation operations

4M	FULL CONNECT	4Mflop
16M	FULL 4096/ReLU	16M
37M	FULL 4096/ReLU	37M
	MAX POOLING	
442K	CONV 3x3/ReLU 256fm	74M
1.3M	CONV 3x3ReLU 384fm	224M
884K	CONV 3x3/ReLU 384fm	149M
	MAX POOLING 2x2sub	
	LOCAL CONTRAST NORM	
307K	CONV 11x11/ReLU 256fm	223M
	MAX POOL 2x2sub	
	LOCAL CONTRAST NORM	
35K	CONV 11x11/ReLU 96fm	105M



Credit: Y. LeCun, Facebook

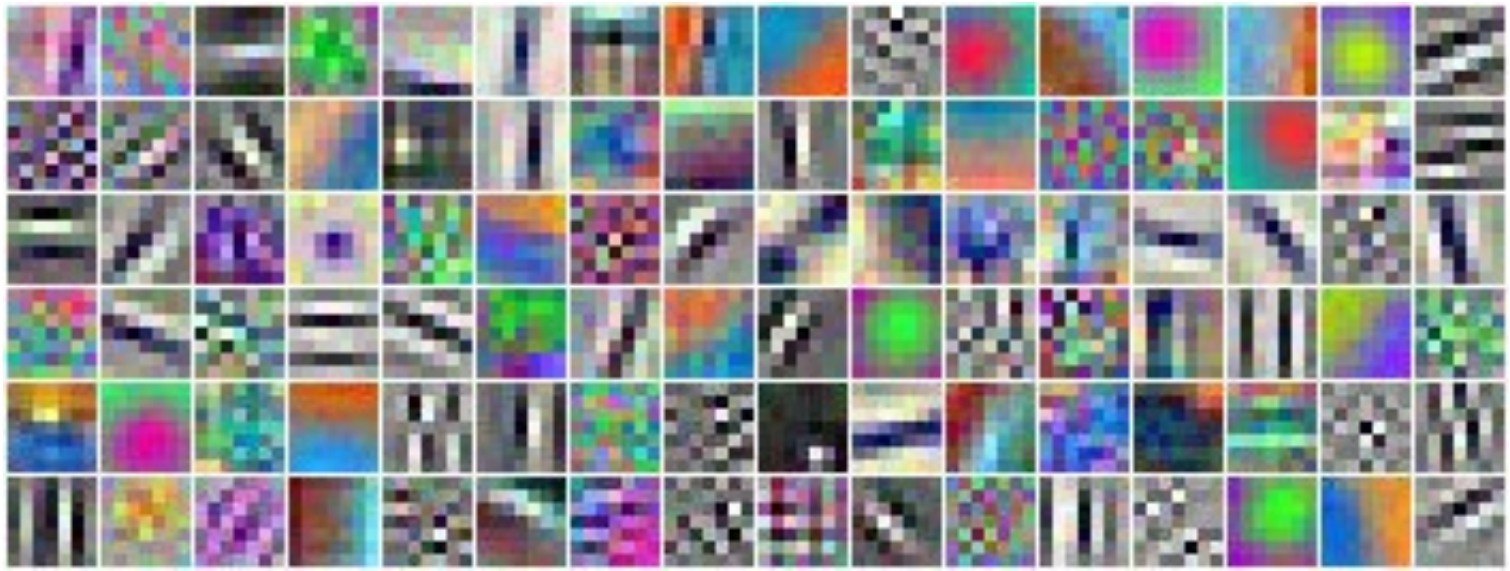
Object Recognition [Krizhevsky, Sutskever, Hinton 2012]

- Method: large convolutional net
  - 650K neurons, 832M synapses, 60M parameters
  - Trained with backprop on NVIDIA GPU
  - Trained "with all the tricks Yann came up with in the last 20 years, plus dropout" (Hinton, NIPS 2012)
  - Rectification, contrast normalization,...
- Error rate: 15% (whenever correct class isn't in top 5) Previous state of the art: 25% error
- **A revolution in computer vision**
- Acquired by Google in Jan 2013
- Deployed in Google+ Photo Tagging in May 2013

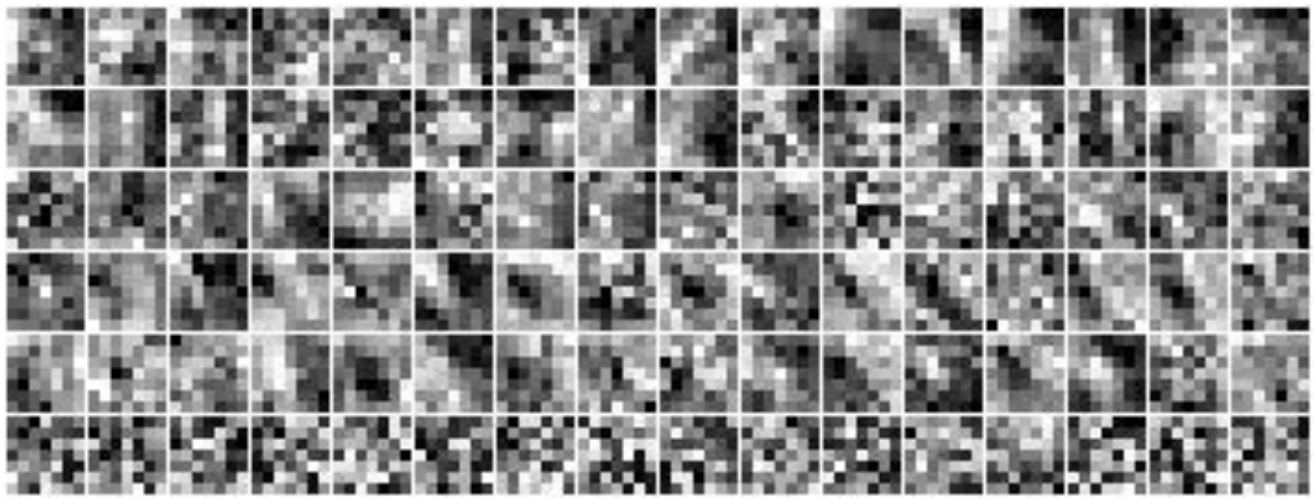
Credit: Y. LeCun, Facebook

Filters: Layer 1 (7x7) and Layer 2 (7x7)

– Layer 1: 3x96 filters, RGB->96 feature maps, 7x7 Filters, stride 2



– Layer 2: 96x256 filters, 7x7



Credit: Y. LeCun, Facebook