Training Large Convolutional Neural Networks

Rob Fergus

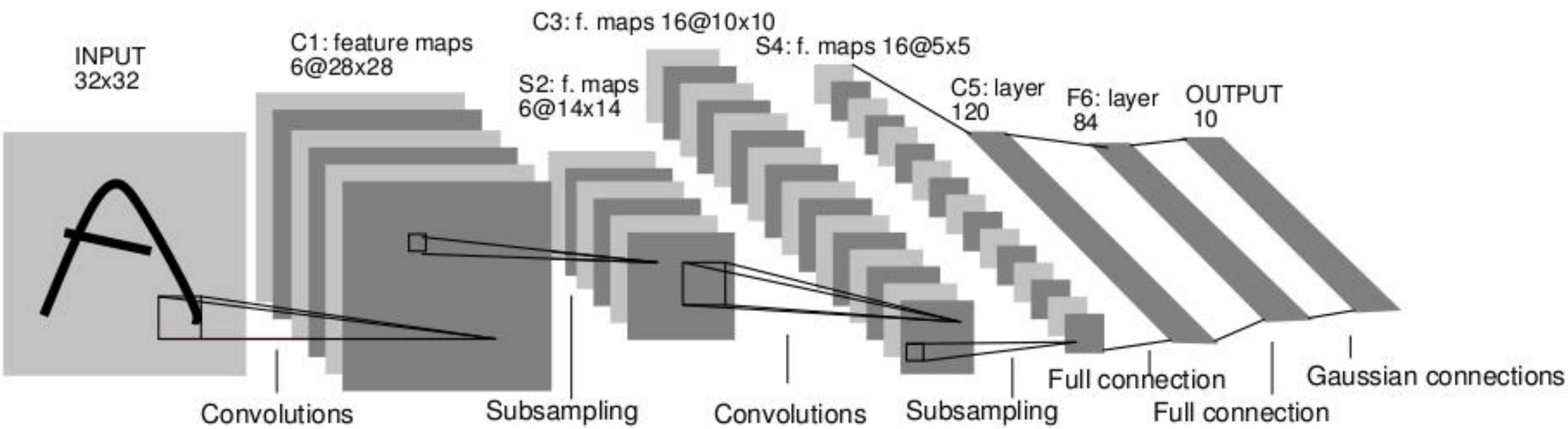
Dept. of Computer Science, Courant Institute,
New York University

Overview

- All about LeCun's Convolutional Neural Networks
  - LeCun et al. 1998
- Krizhevsky, Sutskever & Hinton NIPS 2012
- Stochastic Regularization methods
  - DropOut [Hinton et al. 2012]
  - Other related methods

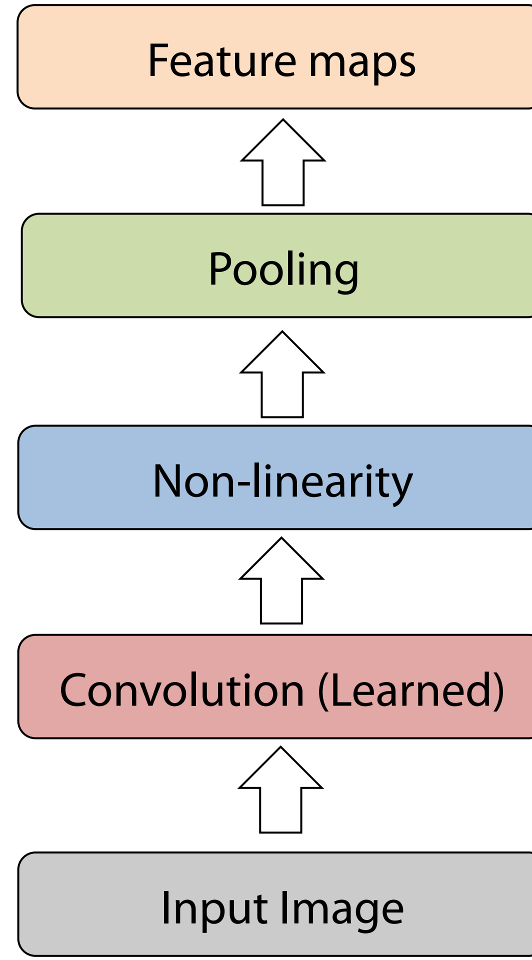
Convolutional Neural Networks

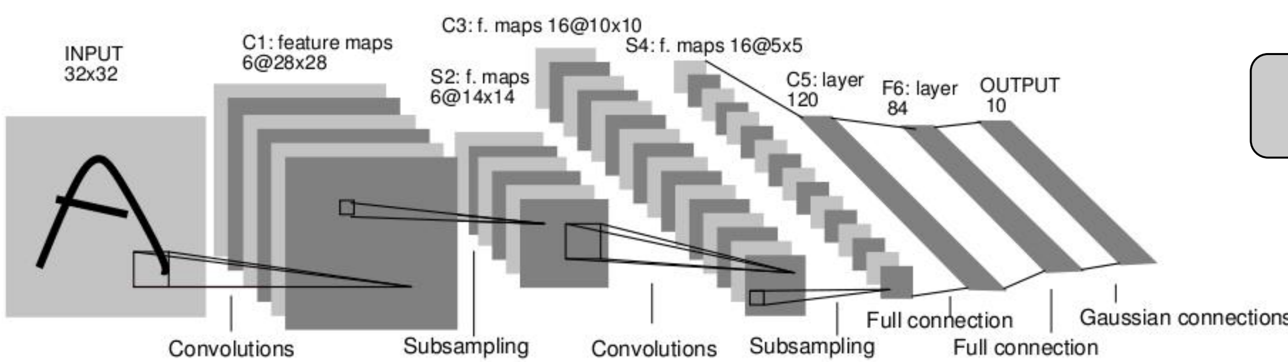
- LeCun et al. 1998
- Very successful on MNIST digits
- But didn't work so well on Caltech 101 (why?)



Recap of Convnets

- Feed-forward:
  - Convolve input
  - Non-linearity (rectified linear)
  - Pooling (local max)
- Supervised
- Train convolutional filters by back-propagating classification error

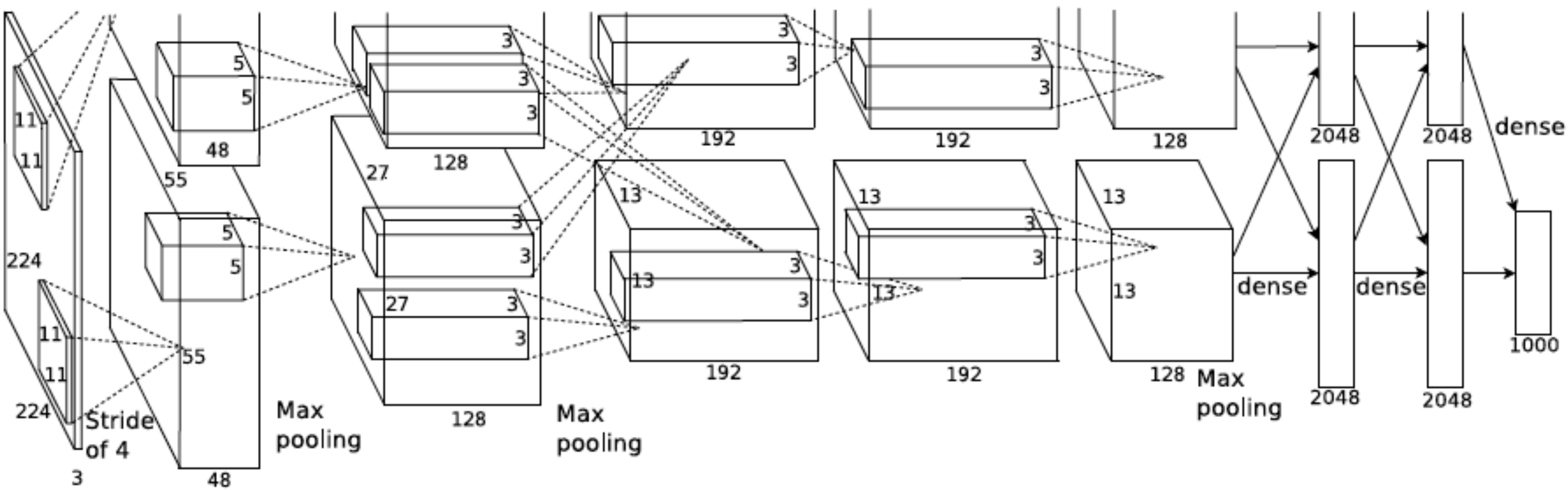




LeCun et al. 1998

Krizhevsky et al. [NIPS2012]

- Same model as LeCun'98 but:
  - bigger model
  - more data
  - GPU implementation



- 7 hidden layers, 650,000 neurons, 60,000,000 parameters
- Trained on 2 GPUs for a week

IMAGENET Large Scale Visual Recognition Challenge 2012 (ILSVRC2012)

Held in conjunction with PASCAL Visual Object Classes Challenge 2012 (VOC2012)

Back to Main page

All results

Task 1

Team name	Filename	Error (5 guesses)	Description
SuperVision	test-preds-141-146.2009-131-137-145-146.2011-145f.	0.15315	Using extra training data from ImageNet Fall 2011 release
SuperVision	test-preds-131-137-145-135-145f.txt	0.16422	Using only supplied training data
ISI	pred_FVs_wLACs_weighted.txt	0.26172	Weighted sum of scores from each classifier with SIFT+FV, LBP+FV, GIST+FV, and CSIFT+FV, respectively.
ISI	pred_FVs_weighted.txt	0.26602	Weighted sum of scores from classifiers using each FV.

Show Alex's Slides

Regularizing Neural Nets

- Neural Networks are good at classifying large labeled datasets
- Large capacity is essential: more layers and more units
- But without regularization, model with millions or billions of parameters can easily overt
- Existing regularization methods:
  - L1 or L2 penalty
  - Bayesian methods
  - Early stopping of training

Stochastic Regularization

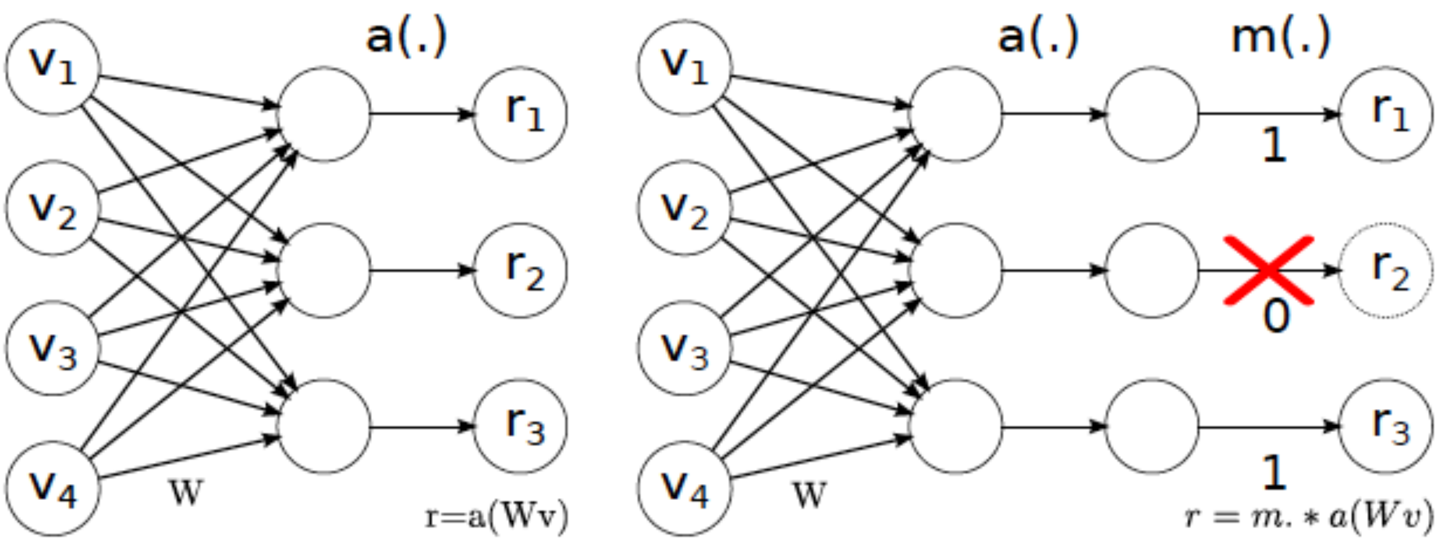
- Deliberately add noise into network
- DropOut [Hinton et al. 2012]
- Recent follow-on work:
  - DropConnect [Wan et al. 2013]
  - Stochastic Pooling [Zeiler & Fergus 2013]
  - MaxOut [Goodfellow 2013]

Review of DropOut Network [Hinton et al. 2012]

- Stochastic dropping of units
- Each element of a layer's output is kept with probability p, otherwise being set to 0 with probability $(1 - p)$
- Input v, weights W, activation function $a(.)$, output r and DropOut mask m:

$$r = m .* a(Wv)$$

- For *every* training example at *every* epoch has different mask m



Normal Network DropOut Network

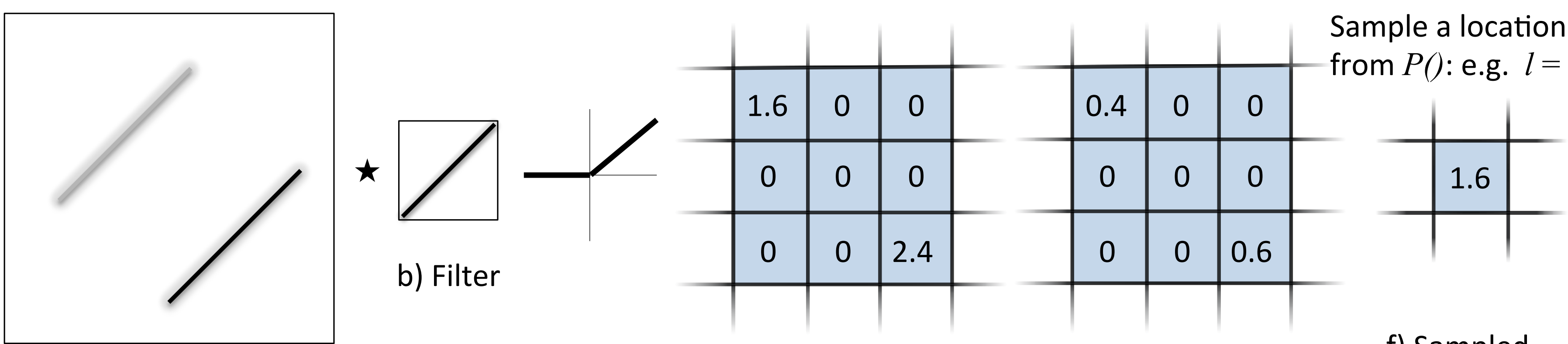
Show Li's Slides

What about Convolution Layers?

- DropOut/DropConnect hurts on these
- MaxOut [Goodfellow et al. 2013]
  - Take max over group of feature maps
- Stochastic Pooling [Zeiler & Fergus 2013]

Stochastic Pooling: Training

- Compute activations a_i : (≥ 0)
- Normalize to sum to 1 -> $p_i = \dfrac{a_i}{\sum_{k \in R_j} a_k}$
- Sample location, l, from multinomial
- Use activation from the location: $s = a_l$

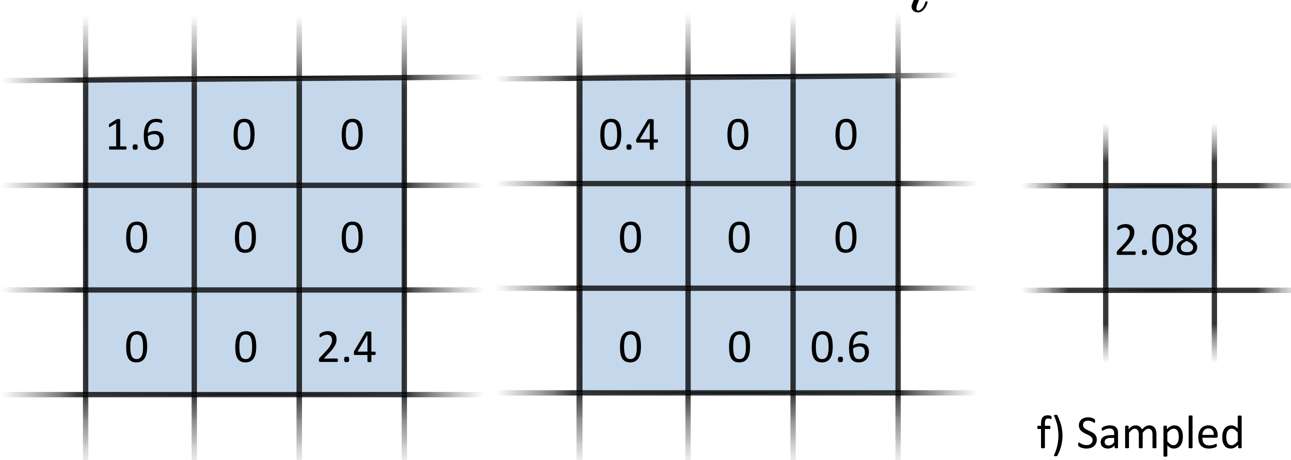


a) Image b) Filter c) Rectified Linear d) Activations, a_i e) Probabilities, p_i f) Sampled Activation, s

Stochastic Pooling: Inference

- Sampling adds noise at test time
- Could sample multiple locations … too slow
- Instead, scale activations by probabilities:

$$s = \sum_i p_i a_i$$

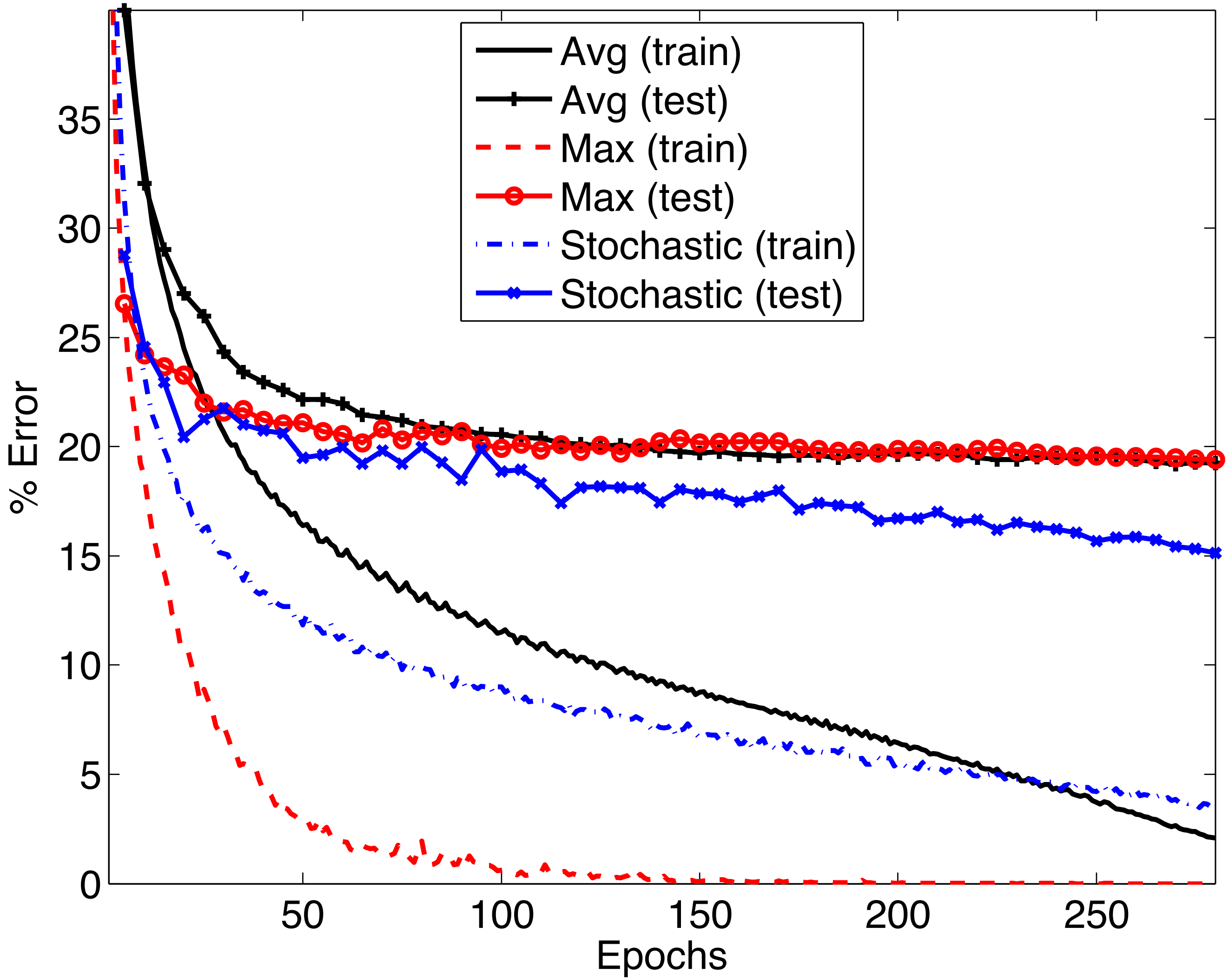


d) Activations, a_i e) Probabilities, p_i f) Sampled Activation, s

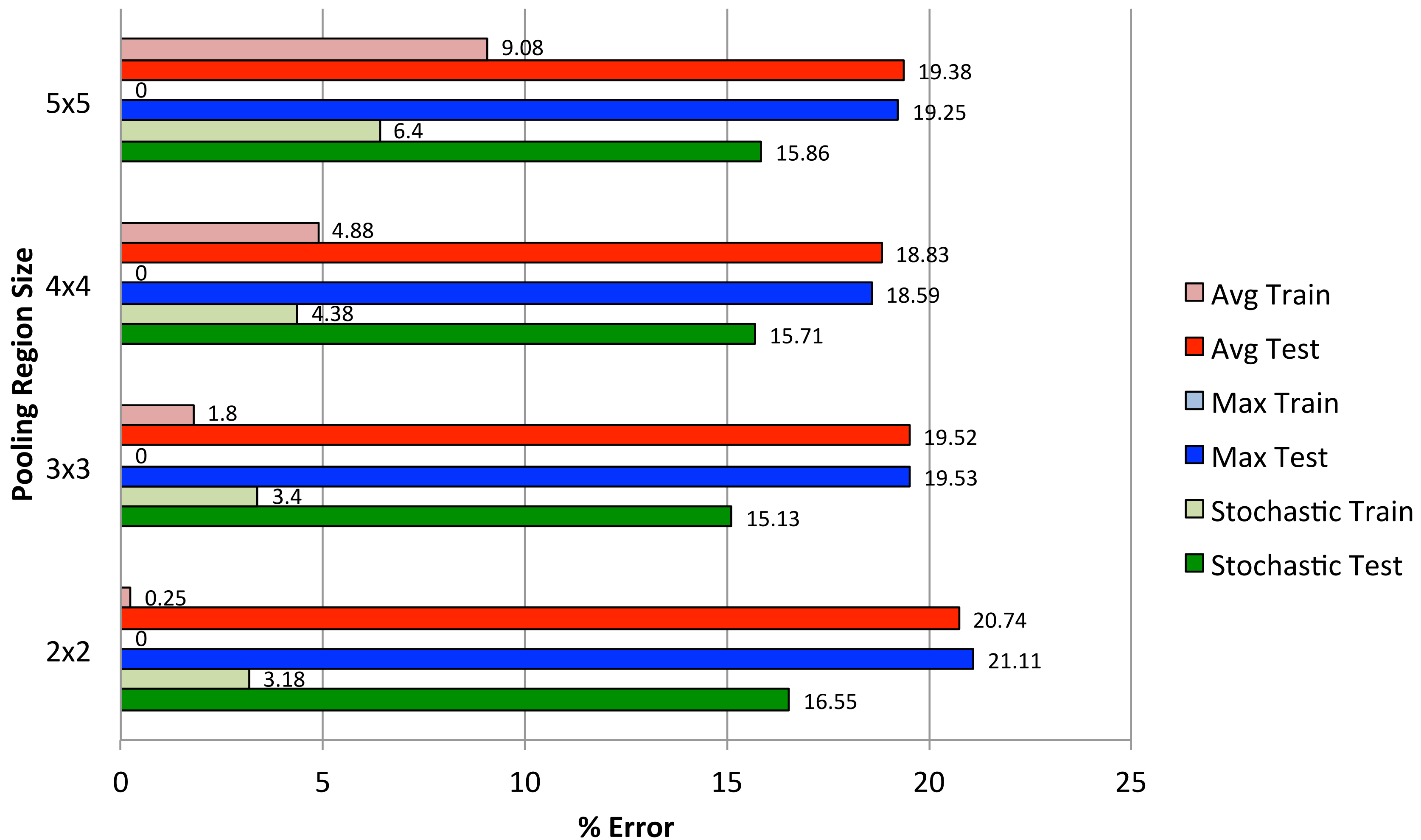
Example:

$$\begin{aligned} 2.08 = 0.4 \times 1.6 &+ \\ 0 \times 0 &+ \\ \ldots &+ \\ 0.6 \times 2.4 & \end{aligned}$$

Convergence and Overfitting: CIFAR-10



Effects of Pooling Size



CIFAR-10 Results

	Train Error %	Test Error %
Multi-Stage Conv. Net + 2-layer Classifier [12]	–	5.03
Multi-Stage Conv. Net + 2-layer Classifer + padding [12]	–	4.90
64-64-64 Avg Pooling	1.83	3.98
64-64-64 Max Pooling	0.38	3.65
64-64-64 Stochastic Pooling	1.72	3.13
64-64-128 Avg Pooling	1.65	3.72
64-64-128 Max Pooling	0.13	3.81
64-64-128 Stochastic Pooling	1.41	**2.80**

Train/Test combinations

Train Method	Test Method	Train Error %	Test Error %
Stochastic Pooling	Probability Weighting	3.20	**15.20**
Stochastic Pooling	Stochastic Pooling	3.20	17.49
Stochastic Pooling	Stochastic-10 Pooling	3.20	15.51
Stochastic Pooling	Stochastic-100 Pooling	3.20	**15.12**
Stochastic Pooling	Max Pooling	3.20	17.66
Stochastic Pooling	Avg Pooling	3.20	53.50
Probability Weighting	Probability Weighting	0.0	19.40
Probability Weighting	Stochastic Pooling	0.0	24.00
Probability Weighting	Max Pooling	0.0	22.45
Probability Weighting	Avg Pooling	0.0	58.97
Max Pooling	Max Pooling	0.0	19.40
Max Pooling	Stochastic Pooling	0.0	32.75
Max Pooling	Probability Weighting	0.0	30.00
Avg Pooling	Avg Pooling	1.92	19.24
Avg Pooling	Stochastic Pooling	1.92	44.25
Avg Pooling	Probability Weighting	1.92	40.09

Conclusions

- Big Convnets work really well for classification
- Around half error of existing methods
- Stochastic regularization important to achieve these results
- Future work: detection