ImageNet Classification with Deep Convolutional Neural Networks

Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton

Presented by
Tugce Tasci, Kyunghee Kim

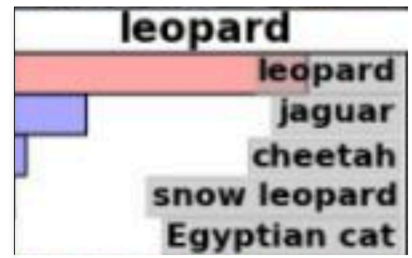
05/18/2015

Outline

- Goal
- DataSet
- Architecture of the Network
- Reducing overfitting
- Learning
- Results
- Discussion

Goal

Classification



ImageNet

- Over 15M labeled high resolution images
- Roughly 22K categories
- Collected from web and labeled by Amazon Mechanical Turk

http://image-net.org/

ILSVRC

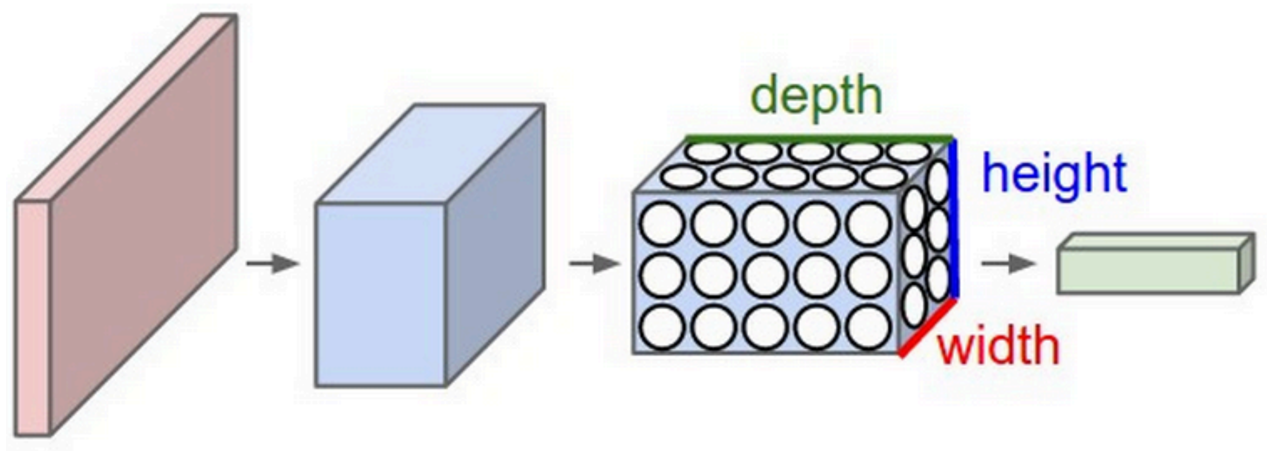
- Annual competition of image classification at large scale
- 1.2M images in 1K categories
- Classification: make 5 guesses about the image label

EntleBucher

Appenzeller

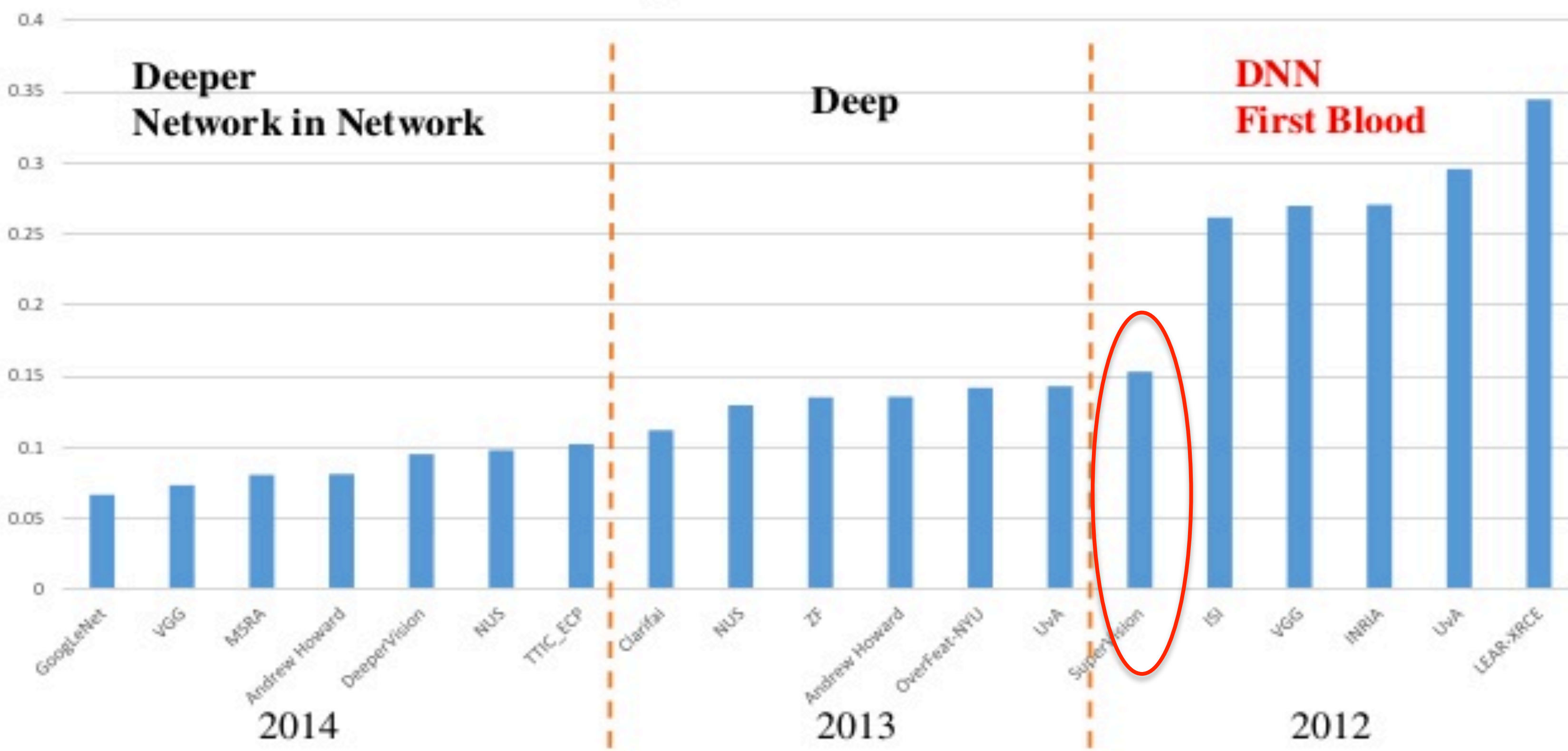
Convolutional Neural Networks

- Model with a large learning capacity
- Prior knowledge to compensate all data we do not have



ILSVRC

ImageNet Classification error throughout years and groups



Li Fei-Fei: ImageNet Large Scale Visual Recognition Challenge, 2014 http://image-net.org/

SuperVision (SV)

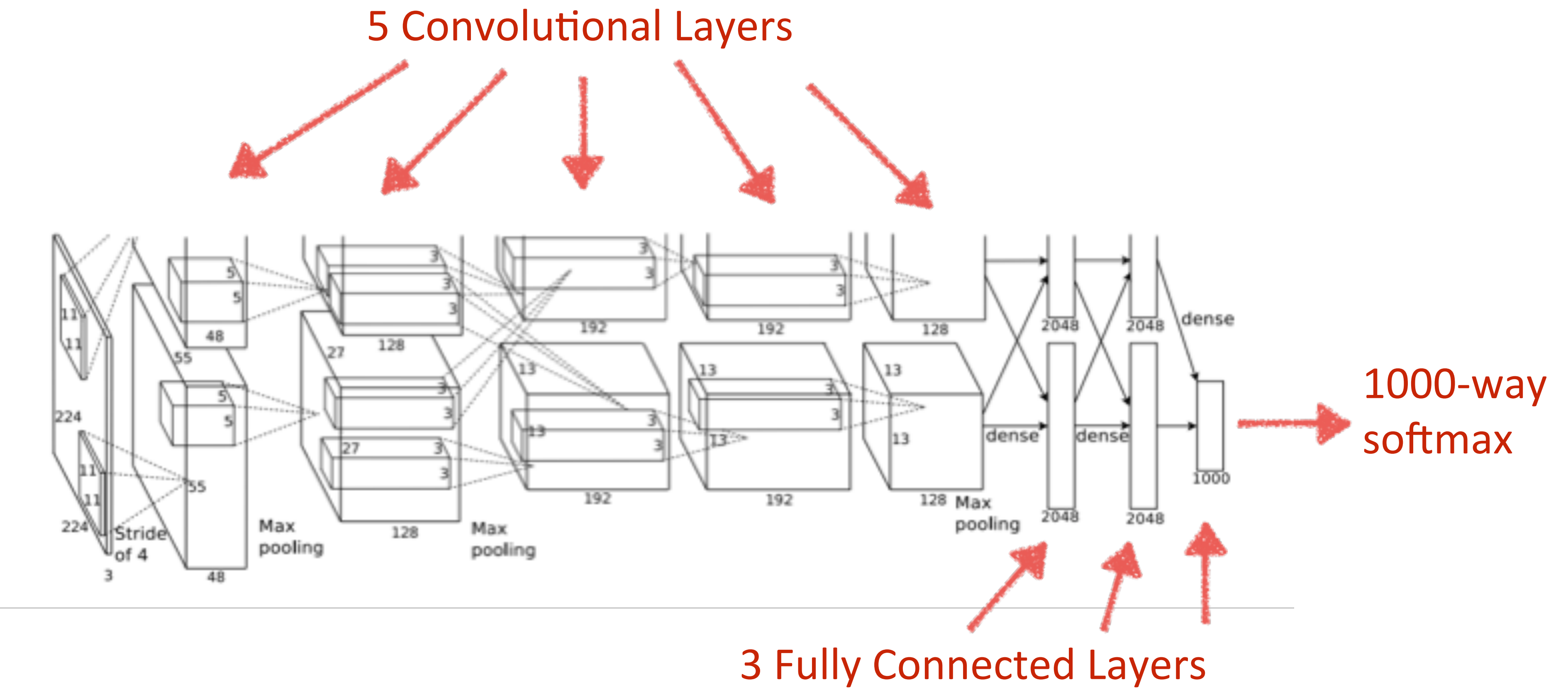
Image classification with deep convolutional neural networks

- 7 hidden “weight” layers
- 650K neurons
- 60M parameters
- 630M connections

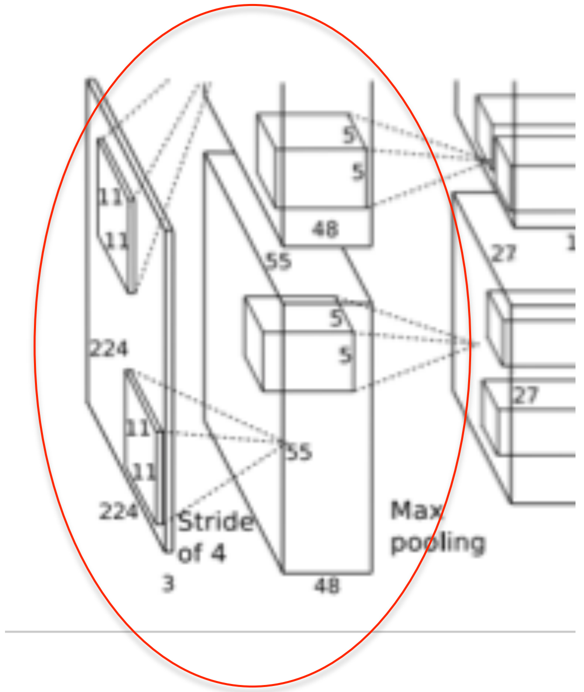
- Rectified Linear Units, overlapping pooling, dropout trick
- Randomly extracted 224x224 patches for more data

http://image-net.org/challenges/LSVRC/2012/supervision.pdf

Architecture

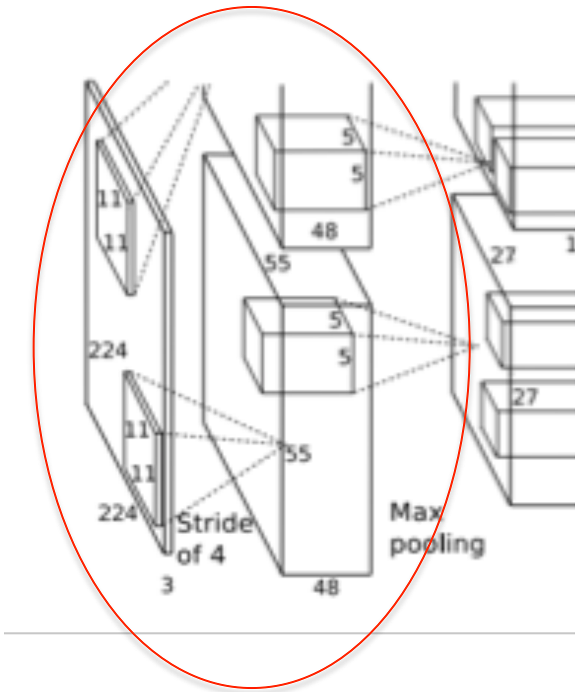


Layer 1 (Convolutional)



- Images: 227x227x3
- F (receptive field size): 11
- S (stride) = 4
- Conv layer output: 55x55x96

Layer 1 (Convolutional)



- 55*55*96 = 290,400 neurons
- each has 11*11*3 = 363 weights and 1 bias
- 290400 * 364 = 105,705,600 paramaters on the first layer of the AlexNet alone!

Architecture

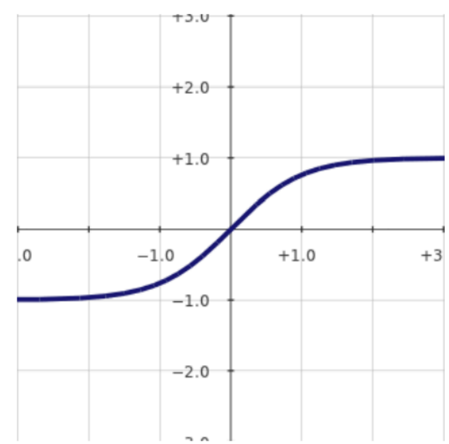
RELU Nonlinearity

- Standard way to model a neuron

$$f(x) = tanh(x) \quad \text{or} \quad f(x) = (1 + e^{-x})^{-1}$$

Very slow to train

f(x) = tanh(x)

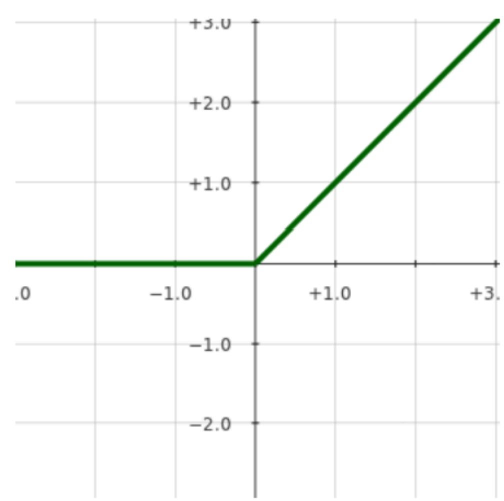


- Non-saturating nonlinearity (RELU)

$$f(x) = max(0, x)$$

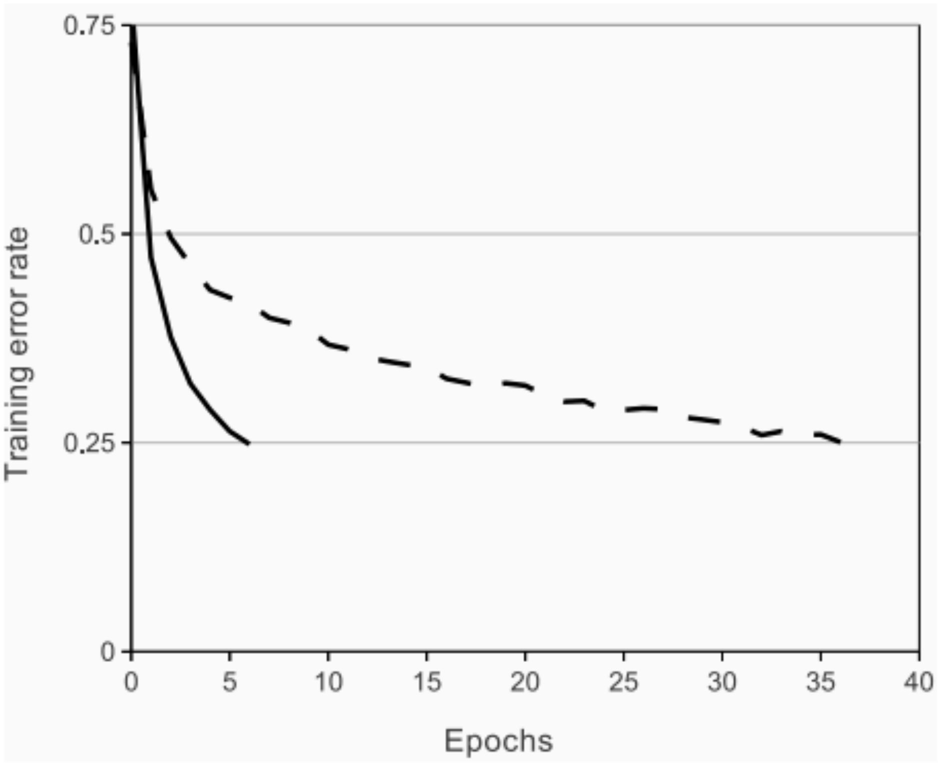
Quick to train

f(x) = max(0, x)



Architecture

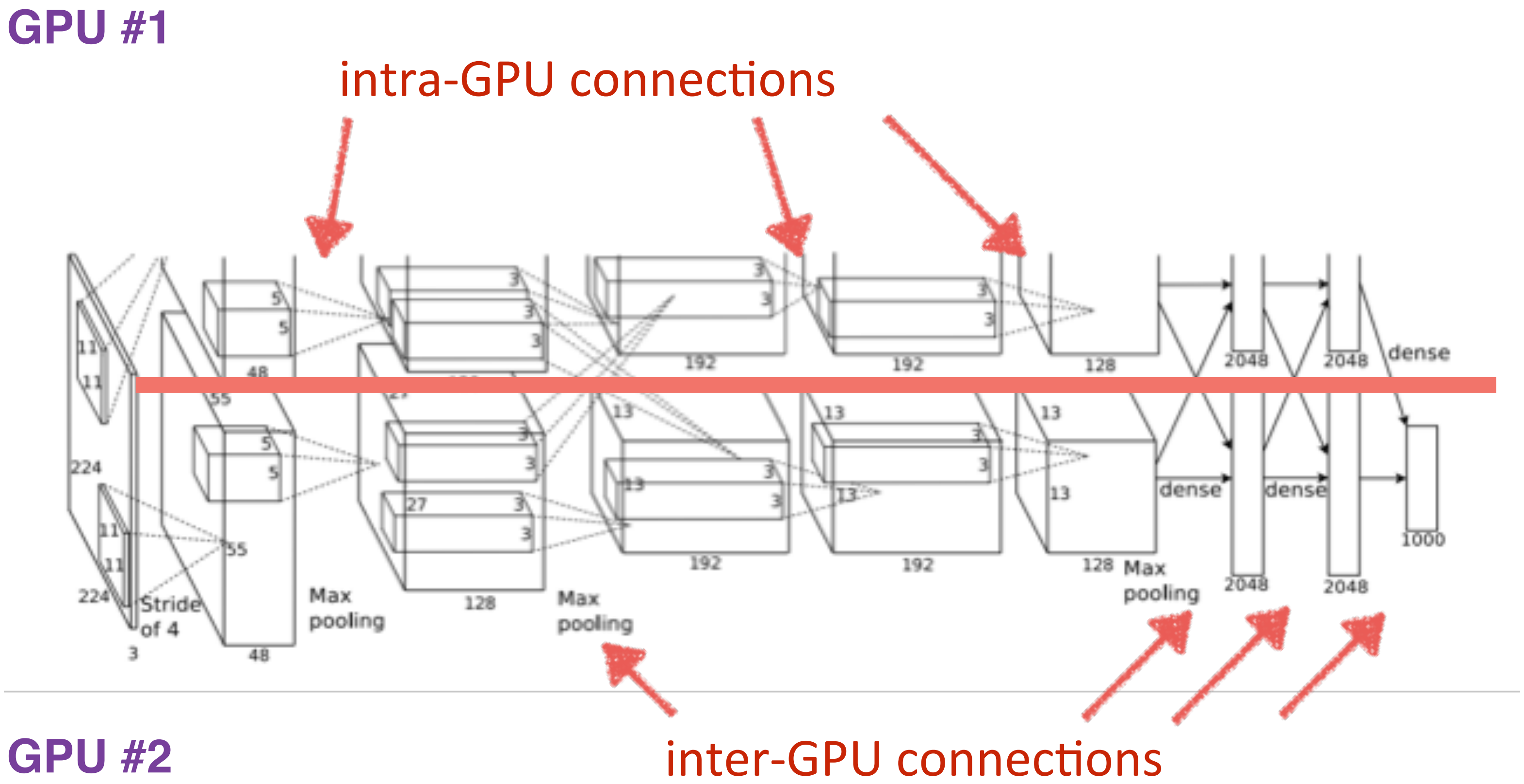
RELU Nonlinearity



A 4 layer CNN with ReLUs (solid line) converges six times faster than an equivalent network with tanh neurons (dashed line) on CIFAR-10 dataset

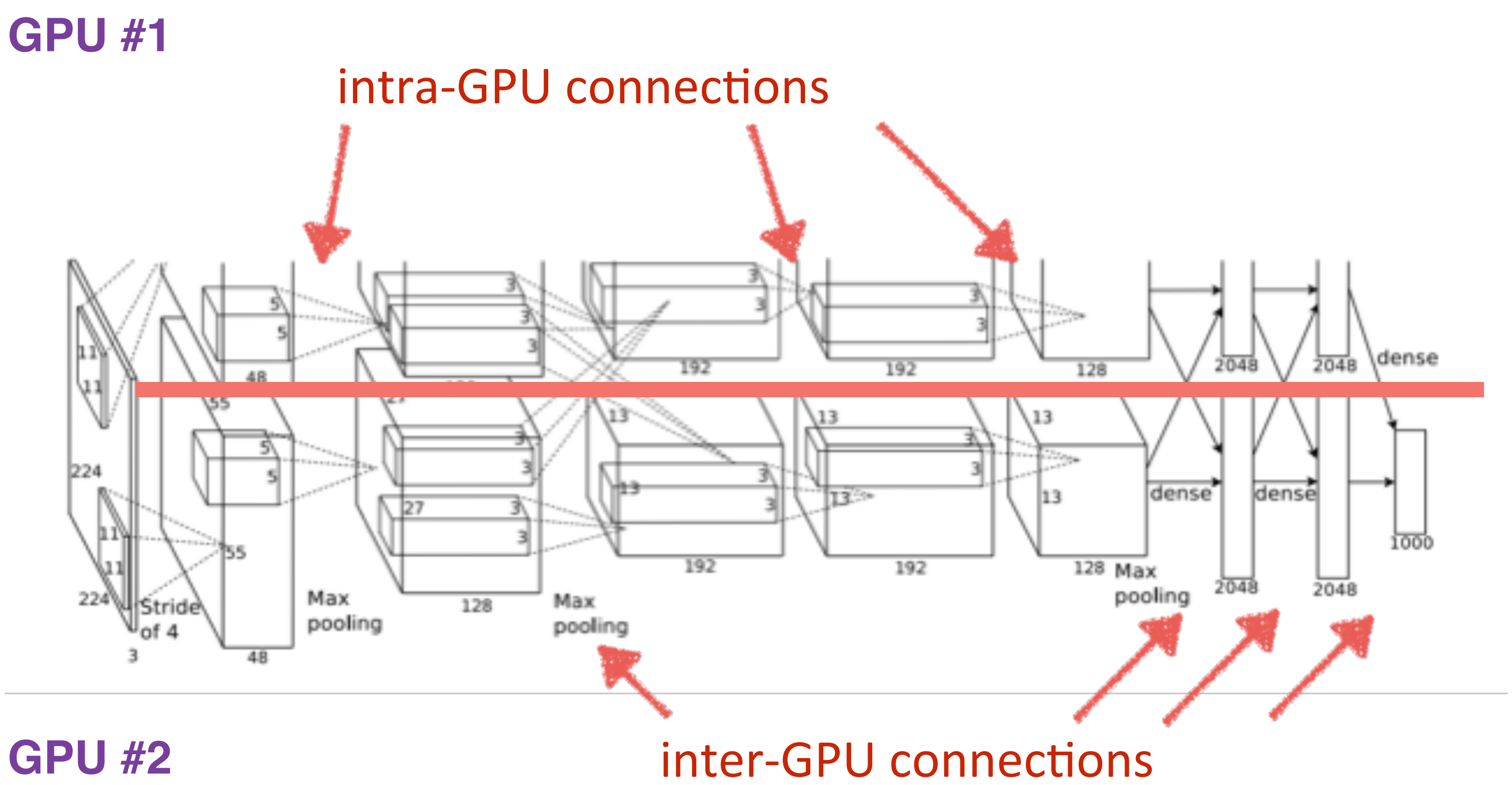
Architecture

Training on Multiple GPUs



Architecture

Training on Multiple GPUs

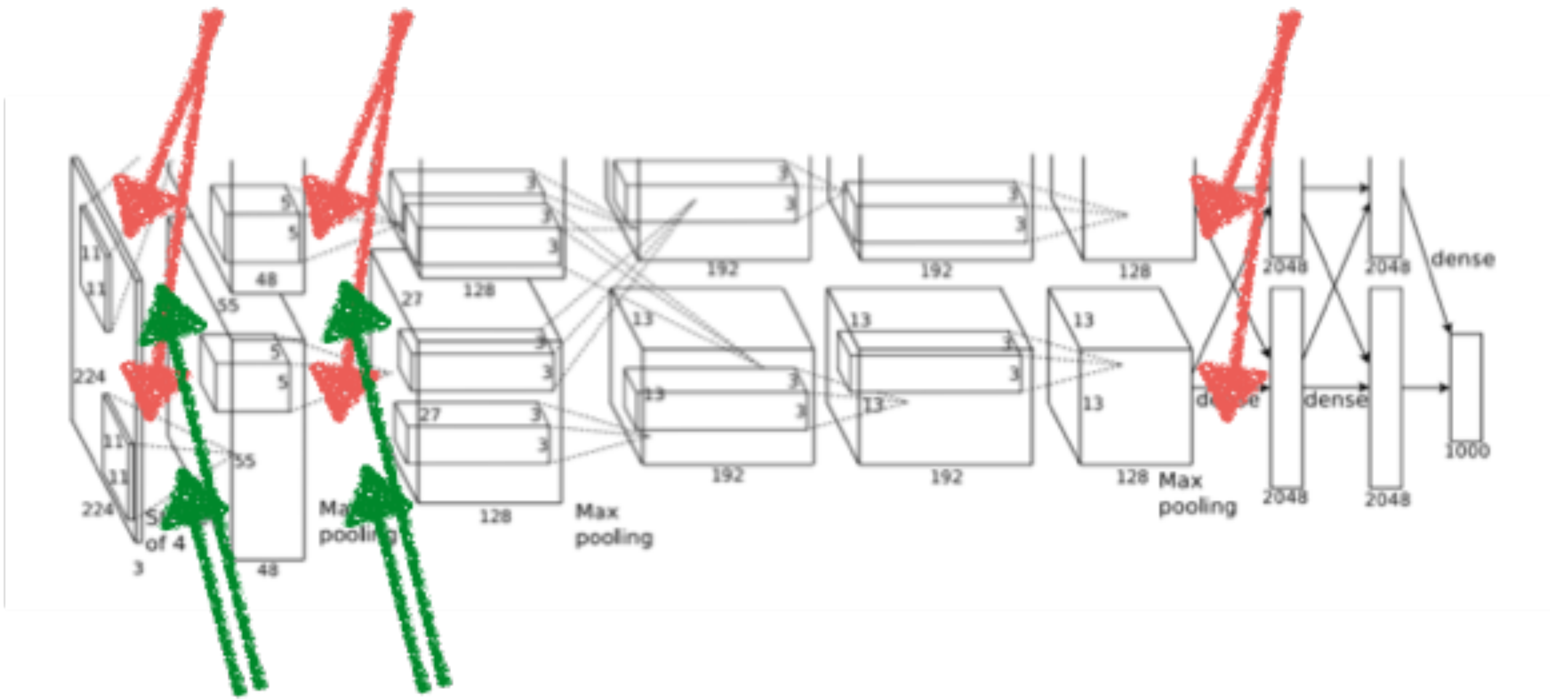


Top-1 and Top-5 error rates decreases by 1.7% & 1.2% respectively, comparing to the net trained with one GPU and half neurons!!

Architecture

Overlaping Pooling

Max-pooling layers



Response normalization layers

Architecture

Local Response Normalization

- No need to input normalization with ReLUs.
- But still the following local normalization scheme helps generalization.

$$b_{x,y}^{i} = a_{x,y}^{i} \Big/ \left(k + \alpha \sum_{j=\max(0,i-n/2)}^{\min(N-1,i+n/2)} \left(a_{x,y}^{j}\right)^2 \right)^{\beta}$$

Response-normalized activity

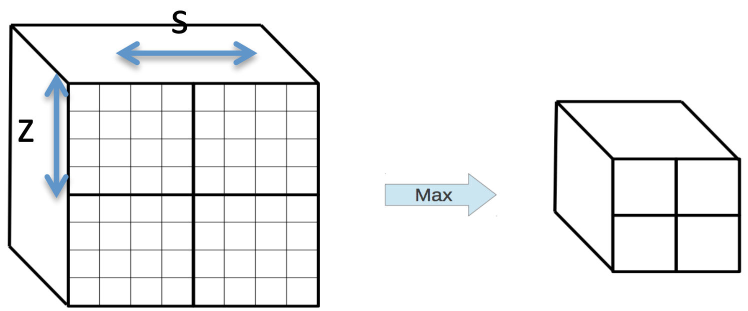
Activity of a neuron computed by applying kernel I at position (x,y) and then applying the ReLU nonlinearity

- Response normalization reduces top-1 and top-5 error rates by 1.4% and 1.2% , respectively.

Architecture

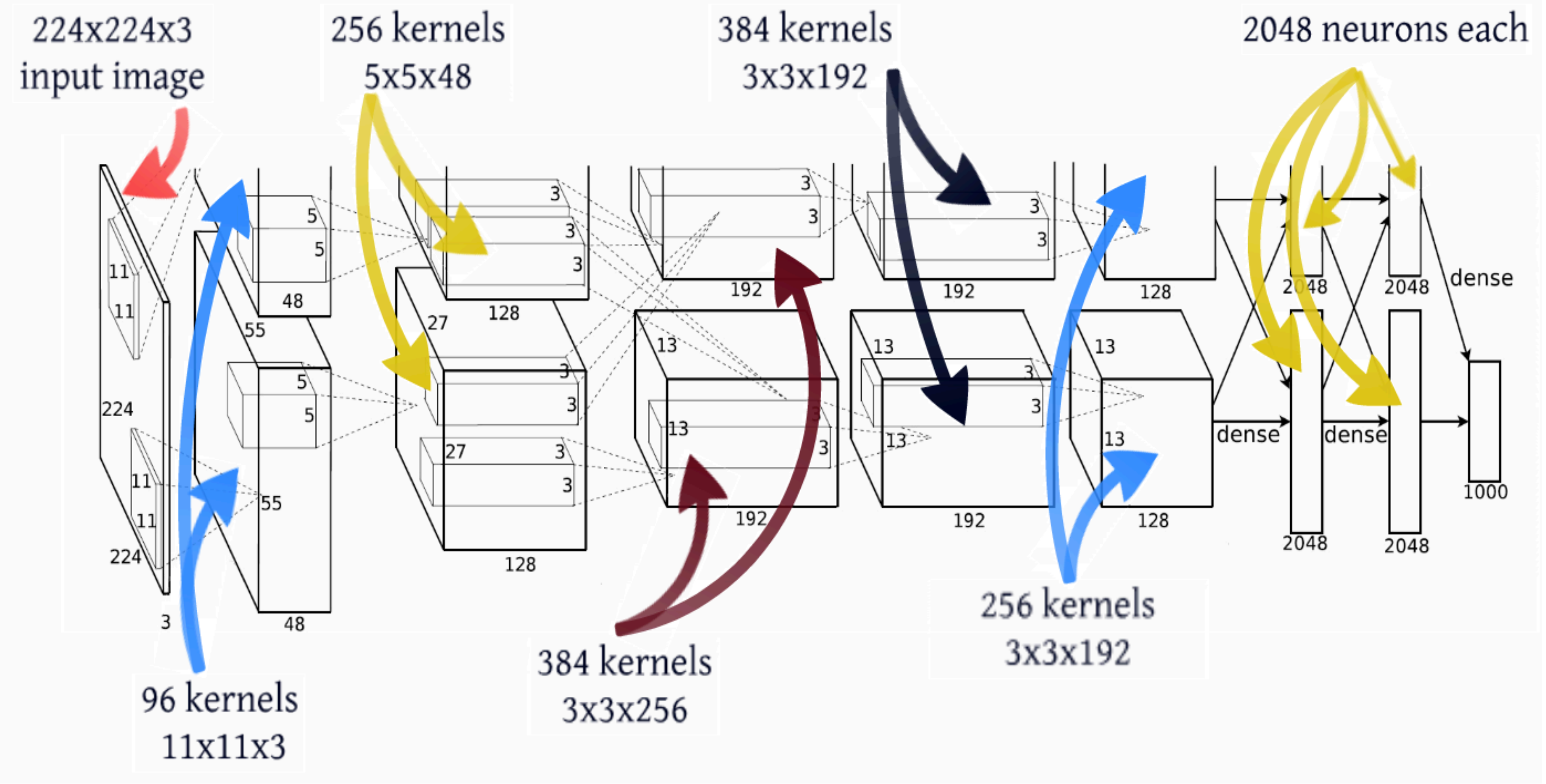
Overlaping Pooling

- Traditional pooling (s = z)

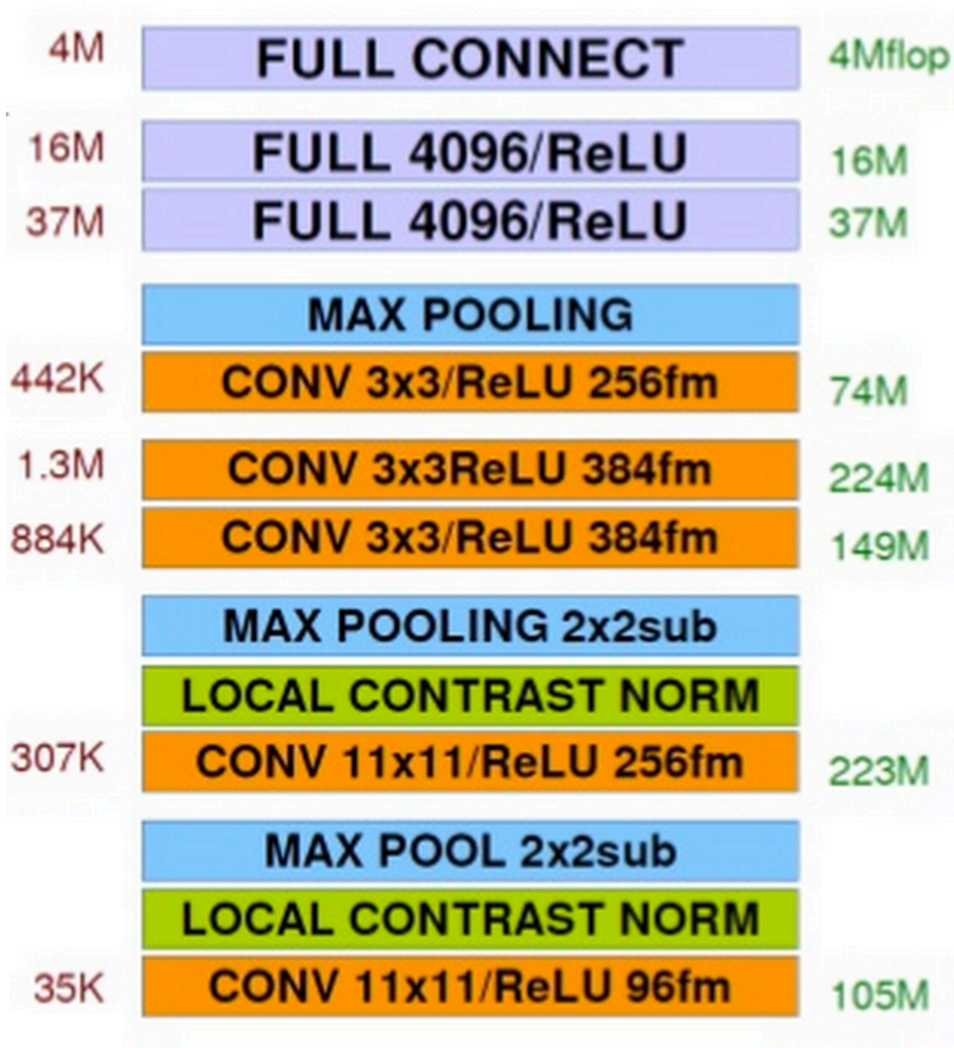


- s < z ➔ overlapping pooling
- top-1 and top-5 error rates decrease by 0.4% and 0.3%, respectively, compared to the non-overlapping scheme s = 2, z = 2

Architecture



Architecture Overview



Outline

- Introduction
- DataSet
- Architecture of the Network
- Reducing overfitting
- Learning
- Results
- Discussion

Reducing Overfitting

Data Augmentation

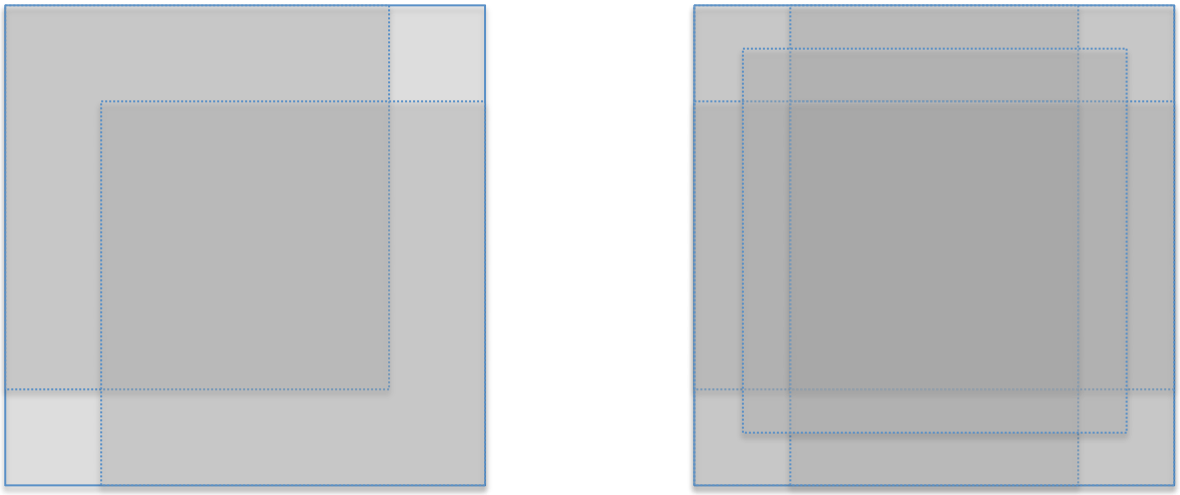
- 60 million parameters, 650,000 neurons
  → Overfits a lot.

- Crop 224x224 patches (and their horizontal reflections.)

Reducing Overfitting

Data Augmentation

- At test time, average the predictions on the 10 patches.



Reducing Overfitting

- Softmax

$$L = \frac{1}{N}\sum_i -\log\left(\frac{e^{f_{y_j}}}{\sum_j e^{f_j}}\right) + \lambda\sum_k\sum_l W_{k,l}^2$$

j = 1…1000

$P(y_i \mid x_i; W)$ Likelihood

- No need to calibrate to average the predictions over 10 patches.

cf. SVM

$$L = \frac{1}{N}\sum_i\sum_{j\neq y_i}\left[\max(0, f(x_i;W)_j - f(x_i;W)_{y_i} + \Delta) + \lambda\sum_k\sum_l W_{k,l}^2\right]$$

Slide credit from [Stanford CS231N Lecture 3]

Reducing Overfitting

Data Augmentation

- Change the intensity of RGB channels
- $$I_{xy} = [I_{xy}^R,\ I_{xy}^G,\ I_{xy}^B]^T$$

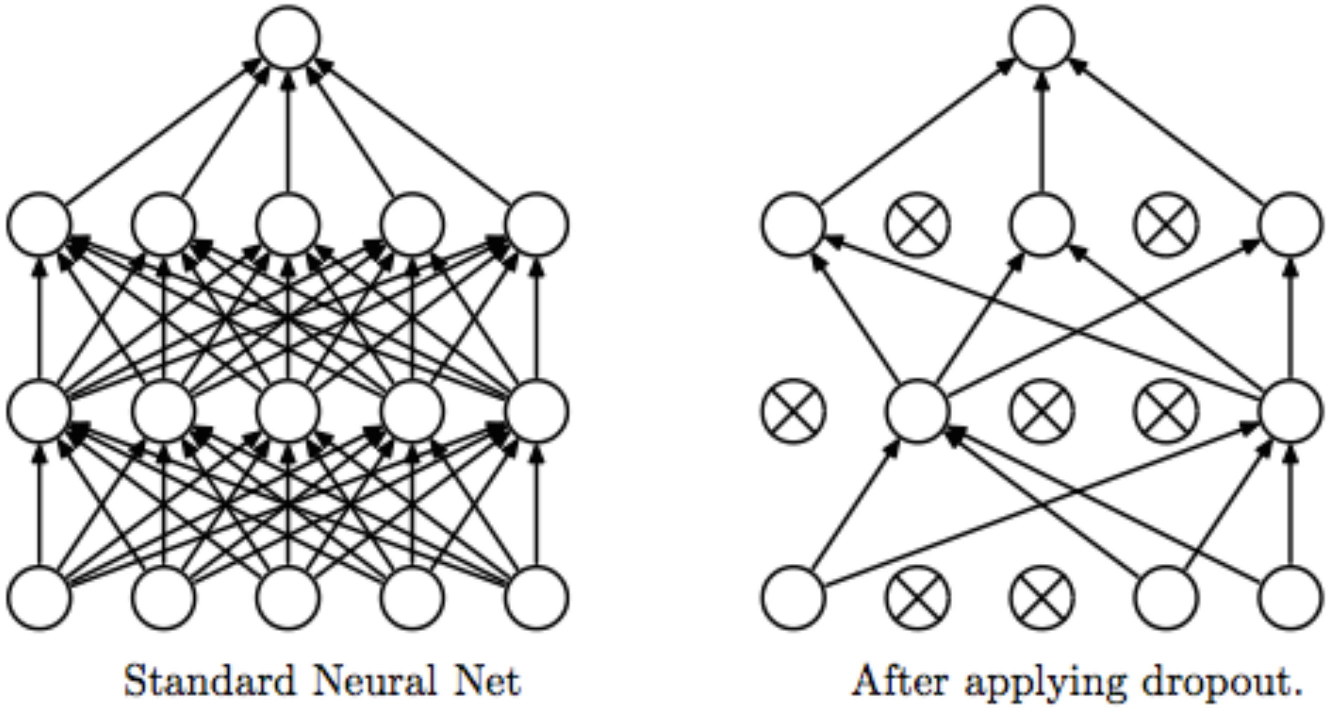
  add multiples of principle components

  $$[\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3][\alpha_1\lambda_1, \alpha_2\lambda_2, \alpha_3\lambda_3]^T$$

  $$\alpha_i \sim N(0,\ 0.1)$$

Reducing Overfitting

Dropout



- With probability 0.5
- last two 4096 fully-connected layers.

Figure credit from [Srivastava et al.]

Stochastic Gradient Descent Learning

Momentum Update

momentum(damping parameter)

Learning rate (initialized at 0.01)

$$v_{i+1} := 0.9 \cdot v_i - 0.0005 \cdot \epsilon \cdot w_i - \epsilon \cdot \left\langle \frac{\partial L}{\partial w}\Big|_{w_i} \right\rangle_{D_i}$$

$$w_{i+1} := w_i + v_{i+1}$$

weight decay

Gradient of Loss
w.r.t weight
Averaged over batch

Batch size: 128

- The training took 5 to 6 days on two NVIDIA GTX 580 3GB GPUs.

Results : ILSVRC-2010

Model	**Top-1**	**Top-5**
Sparse coding [2]	*47.1%*	*28.2%*
SIFT + FVs [24]	*45.7%*	*25.7%*
CNN	**37.5%**	**17.0%**

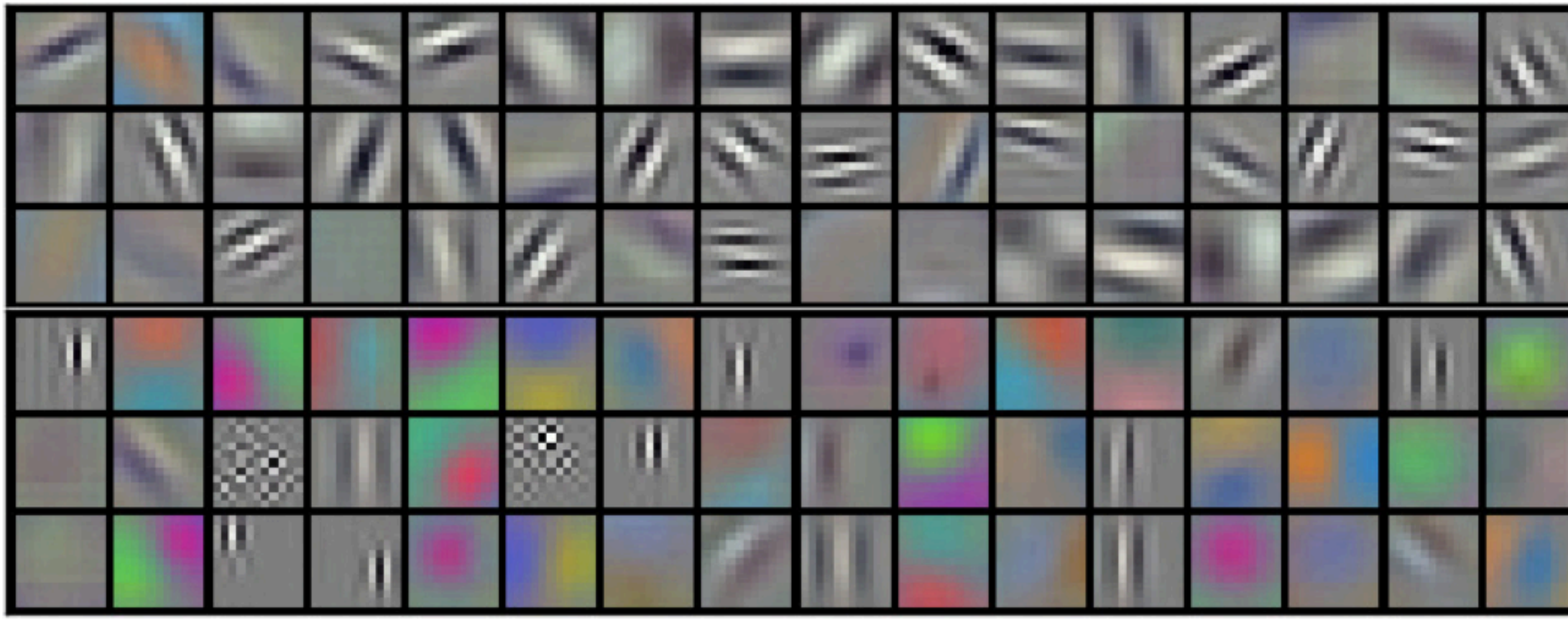
Table 1: Comparison of results on ILSVRC-2010 test set. In *italics* are best results achieved by others.

Results : ILSVRC-2012

Model	Top-1 (val)	Top-5 (val)	Top-5 (test)
SIFT + FVs [7]	—	—	*26.2%*
1 CNN	40.7%	18.2%	—
5 CNNs	38.1%	16.4%	**16.4%**
1 CNN*	39.0%	16.6%	—
7 CNNs*	36.7%	15.4%	**15.3%**

Table 2: Comparison of error rates on ILSVRC-2012 validation and test sets. In *italics* are best results achieved by others. Models with an asterisk* were "pre-trained" to classify the entire ImageNet 2011 Fall release. See Section 6 for details.

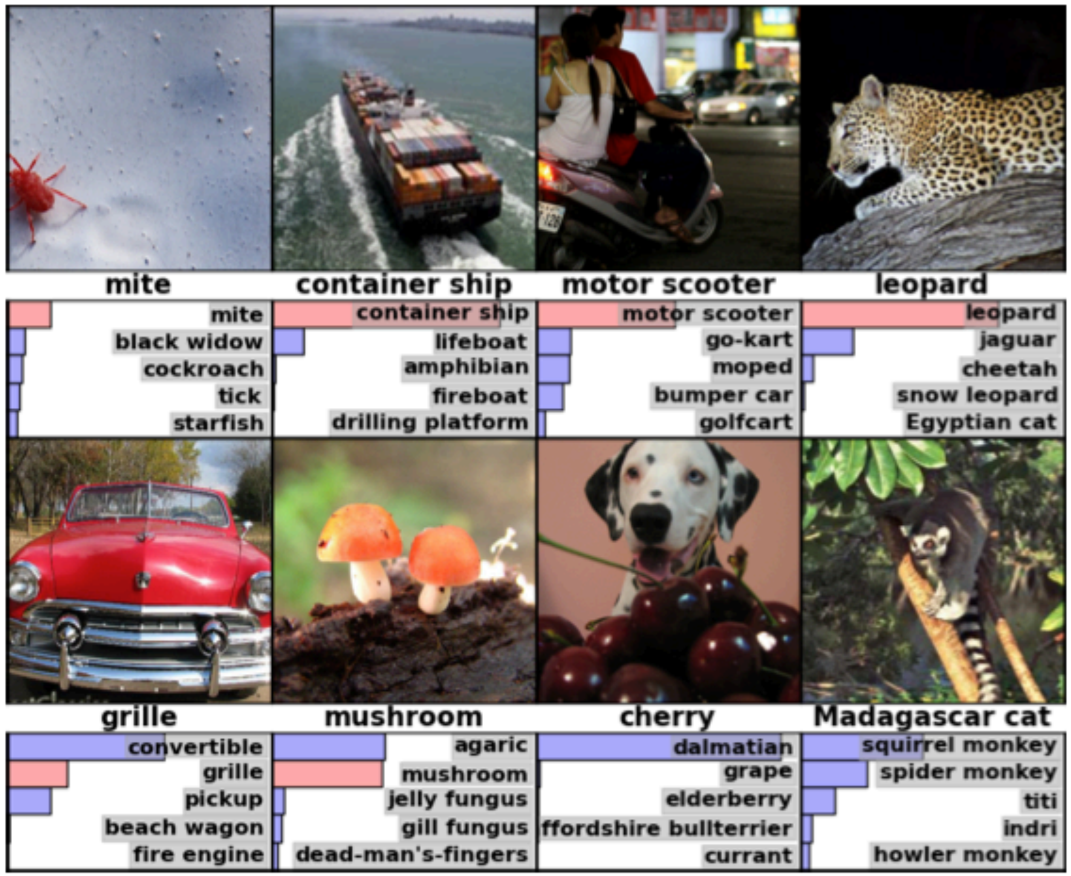
96 Convolutional Kernels



- 11 x 11 x 3 size kernels.
- top 48 kernels on GPU 1 : color-agnostic
- bottom 48 kernels on GPU 2 : color-specific.

Why?

Eight ILSVRC-2010 test images



Five ILSVRC-2010 test images

The output from the last 4096 fully-connected layer :

4096 dimensional feature.

Discussion

- Depth is really important.

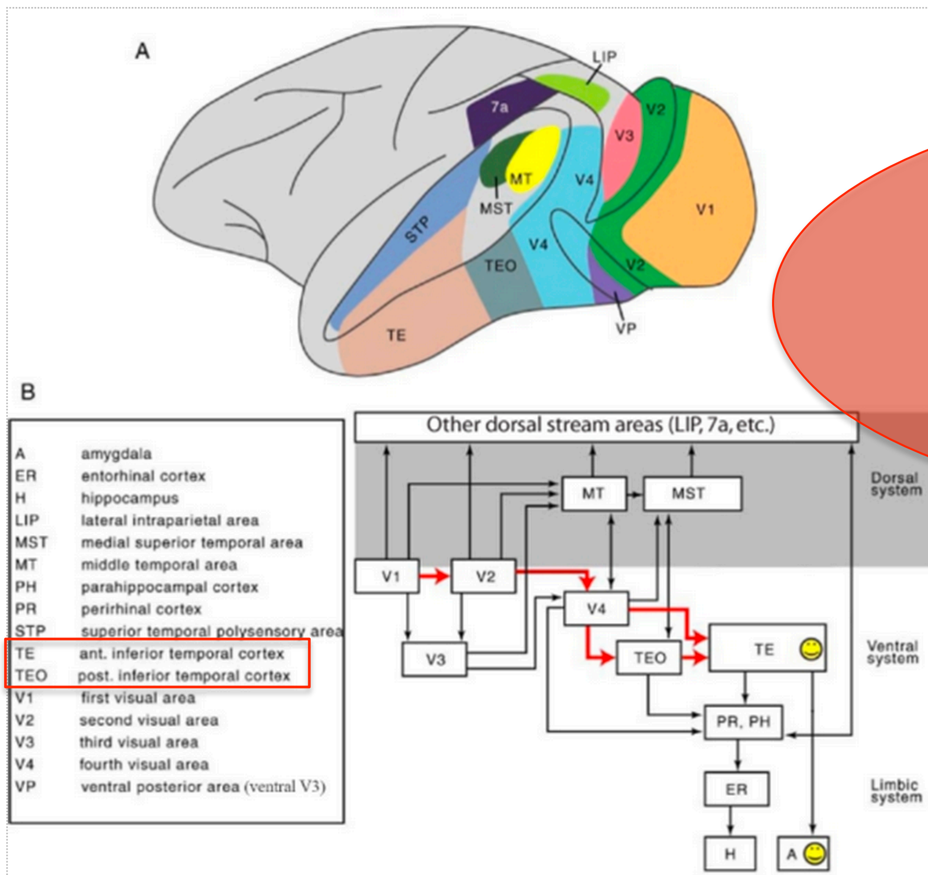
  removing a single convolutional layer degrades the performance.

  K. Simonyan, A. Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. Technical report, 2014.

  → 16-layer model, 19-layer model. 7.3% top-5 test error on ILSVRC-2012

Discussion

- Still have many orders of magnitude to go in order to match the infero-temporal(IT) pathway of the human visual system.



Convolutional Neural Networks? vs. Convolutonal Networks?

Figure adapted from Gross, C. G., Rodman, H. R., Gochin, P. M., and Colombo, M. W. (1993). Inferior temporal cortex as a pattern recognition device. In “Computational Learning and Cognition” (E. Baum, ed.), pp. 44–73. Society for Industrial and Applied Mathematics, Philadelphia.

Discussion

- Classification on video.

  video sequences provide temporal structure missing in static images.

  K. Simonyan, A. Zisserman.
  Two-Stream Convolutional Networks for Action Recognition in Videos. NIPS 2014.

  → separating two pathways for spatial and temporal networks analogous to the ventral and dorsal pathways.