Credits

Many of the pictures, results, and other materials are taken from:

Ruslan Salakhutdinov
Joshua Bengio
Geoffrey Hinton
Yann LeCun

Deep Convolutional Networks

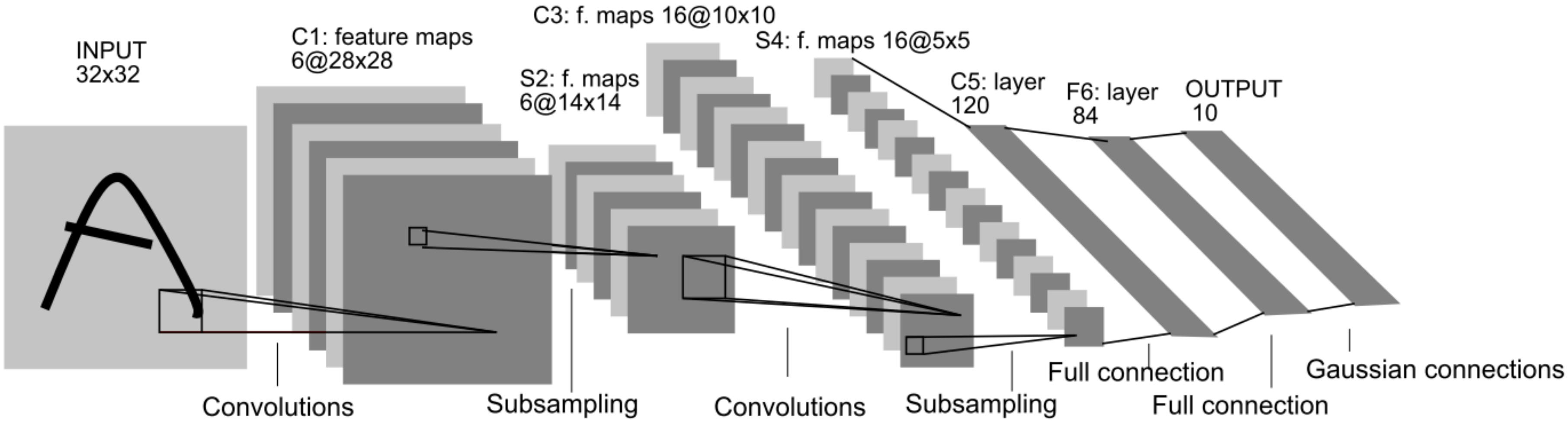
Compared to standard feedforward neural networks with similarly-sized layers,

- CNNs have much fewer connections and parameters
- and so they are easier to train,
- while their theoretically-best performance is likely to be only slightly worse.

LeNet 5

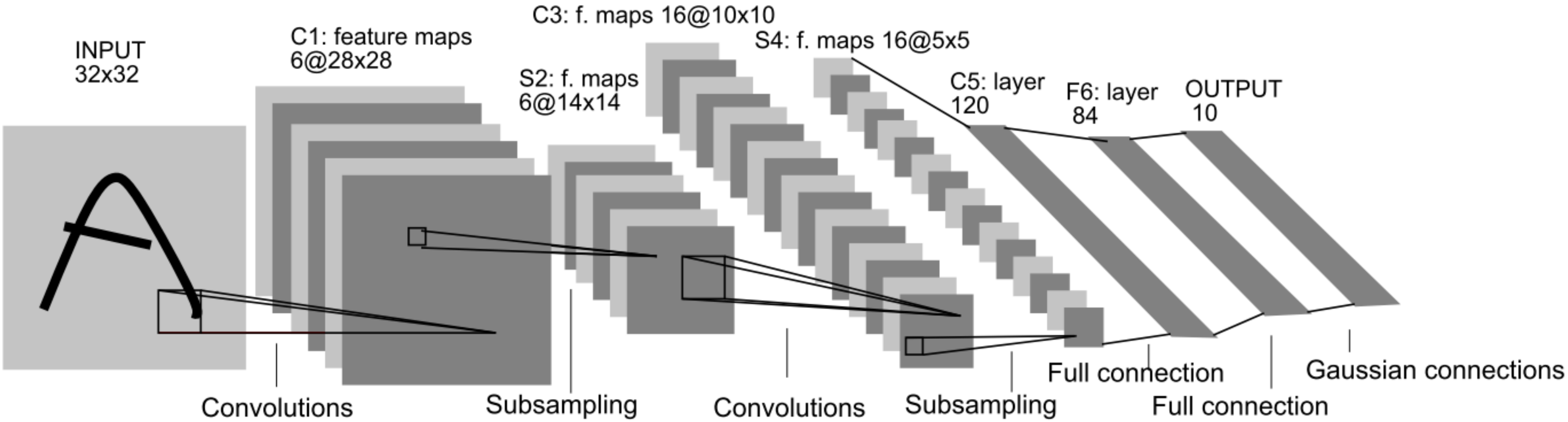
Y. LeCun, L. Bottou, Y. Bengio and P. Haffner: **Gradient-Based Learning Applied to Document Recognition**, *Proceedings of the IEEE, 86(11):2278-2324, November **1998***

LeNet 5, LeCun 1998



- Input: 32x32 pixel image. Largest character is 20x20
  (All important info should be in the center of the receptive field of the highest level feature detectors)
- Cx: Convolutional layer
- Sx: Subsample layer
- Fx: Fully connected layer
- Black and White pixel values are normalized:
  E.g. White = -0.1, Black =1.175 (Mean of pixels = 0, Std of pixels =1)

LeNet 5, Layer C1



C1: Convolutional layer with 6 feature maps of size 28x28. $C1_k$ (k=1…6)

Each unit of C1 has a 5x5 receptive field in the input layer.

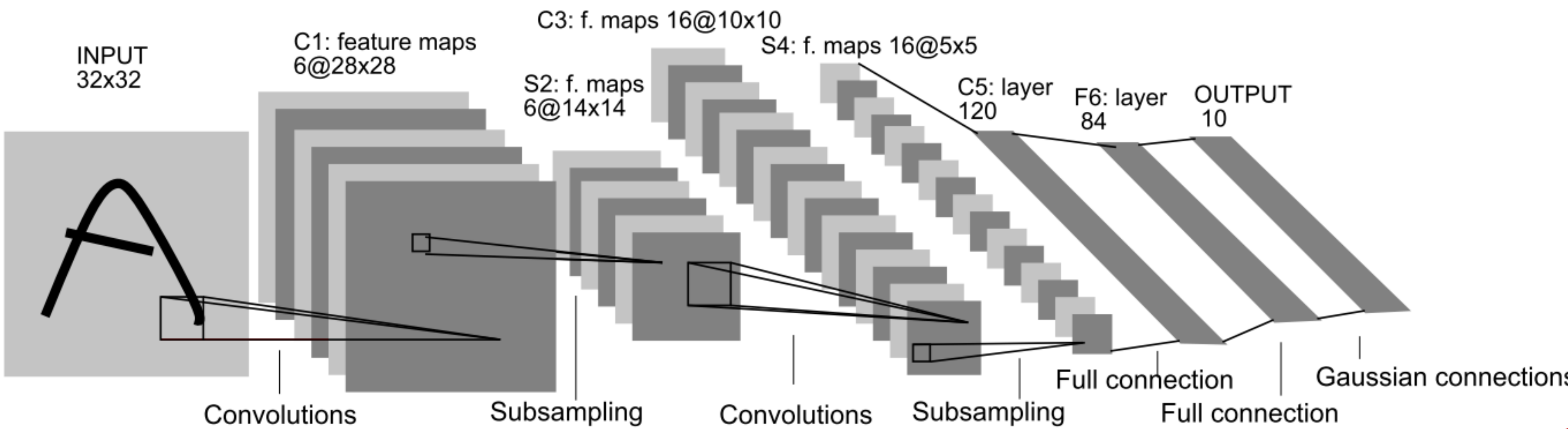
- Topological structure
- Sparse connections
- Shared weights

(5*5+1)*6=156 parameters to learn

Connections: 28*28*(5*5+1)*6=122304

If it was fully connected we had (32*32+1)*(28*28)*6 parameters

LeNet 5, Layer S2



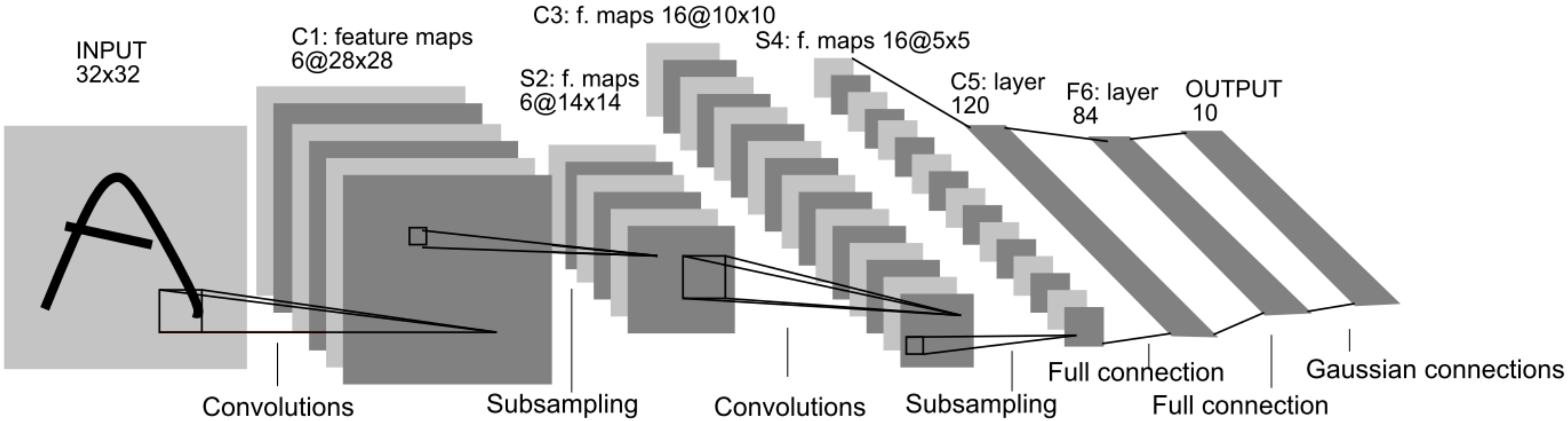
S2: Subsampling layer with 6 feature maps of size 14x14

2x2 nonoverlapping receptive fields in C1

Layer S2: 6*2=12 trainable parameters.

Connections: 14*14*(2*2+1)*6=5880

LeNet 5, Layer C3



- C3: Convolutional layer with 16 feature maps of size 10x10
- Each unit in C3 is connected to several! 5x5 receptive fields at identical locations in S2

Layer C3:

1516 trainable parameters.

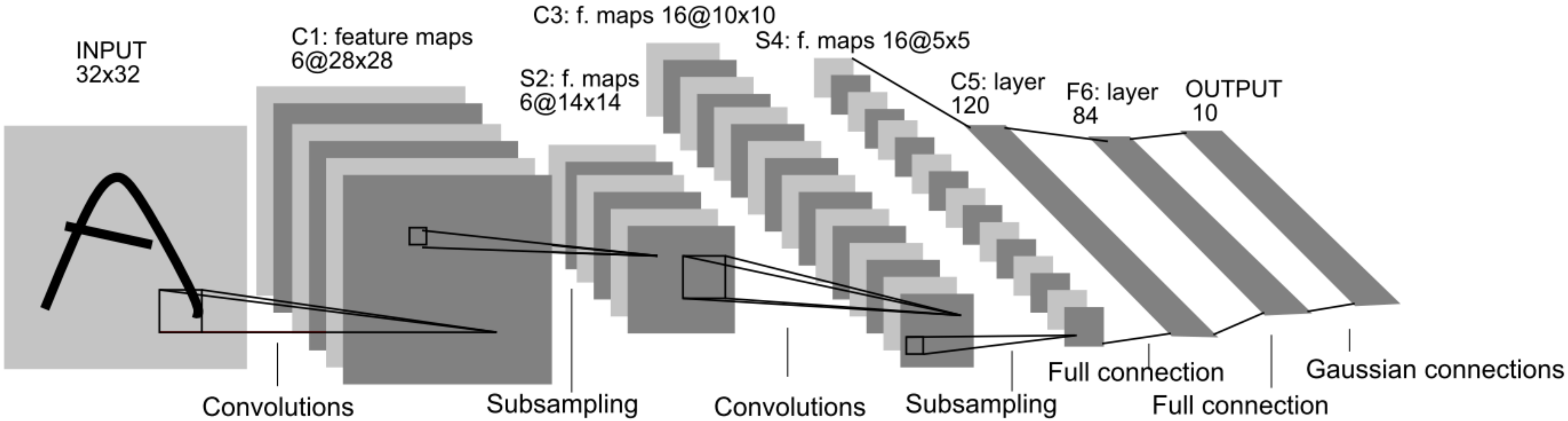
Connections: 151600

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	X				X	X	X			X	X	X	X		X	X
1	X	X				X	X	X			X	X	X	X		X
2	X	X	X				X	X	X			X		X	X	X
3		X	X	X			X	X	X	X			X		X	X
4			X	X	X			X	X	X	X		X	X		X
5				X	X	X			X	X	X	X		X	X	X

TABLE I

EACH COLUMN INDICATES WHICH FEATURE MAP IN S2 ARE COMBINED BY THE UNITS IN A PARTICULAR FEATURE MAP OF C3.

LeNet 5, Layer S4

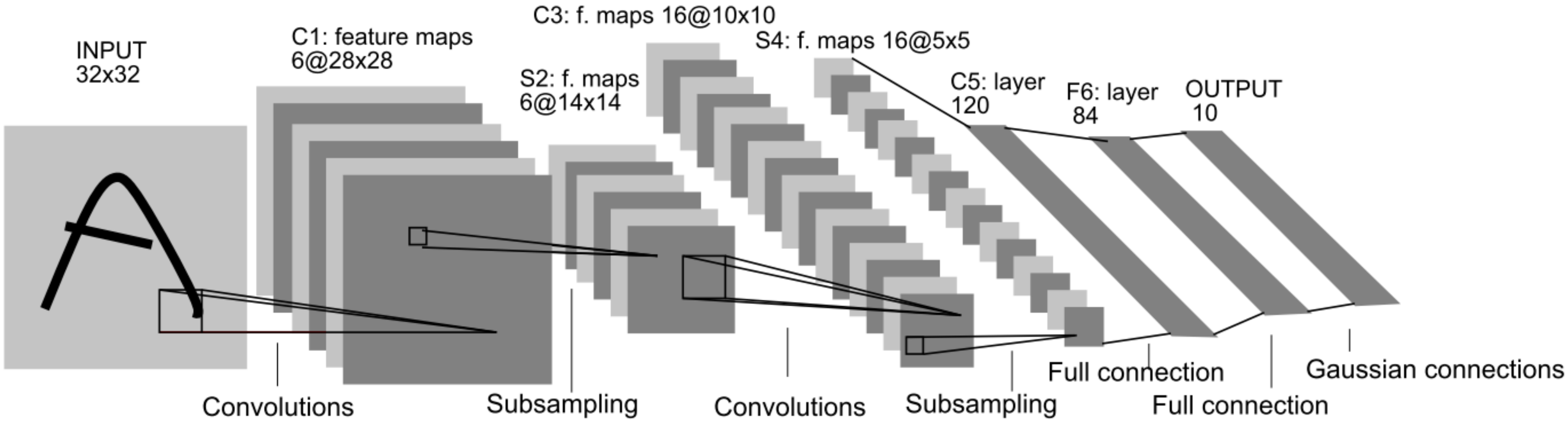


- S4: Subsampling layer with 16 feature maps of size 5x5
- Each unit in S4 is connected to the corresponding 2x2 receptive field at C3

Layer S4: 16*2=32 trainable parameters.

Connections: 5*5*(2*2+1)*16=2000

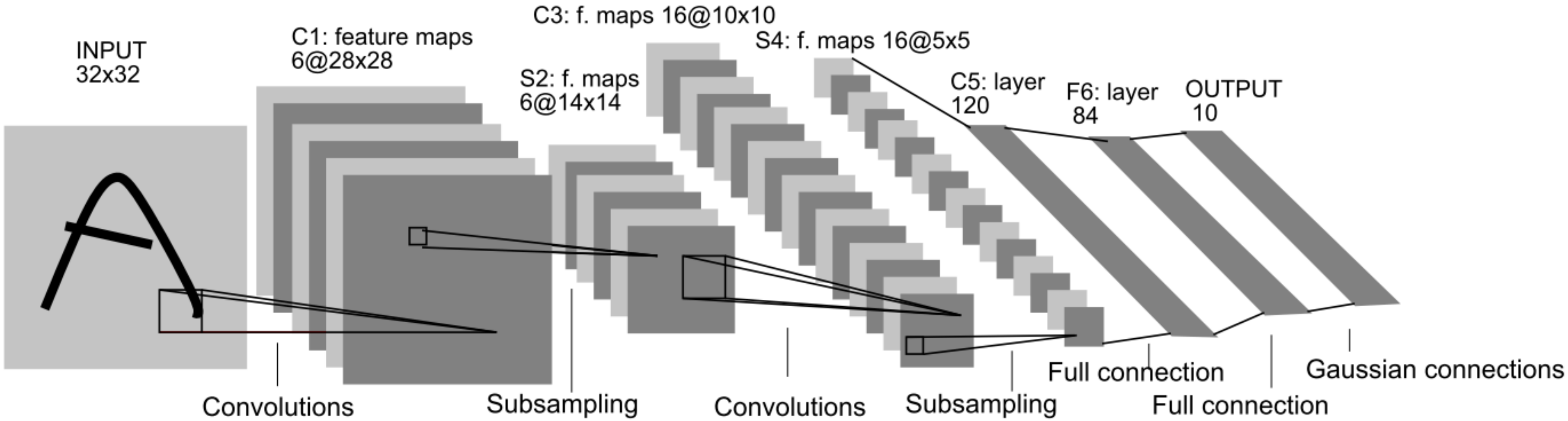
LeNet 5, Layer C5



- C5: Convolutional layer with 120 feature maps of size 1x1
- Each unit in C5 is connected to all 16 5x5 receptive fields in S4

Layer C5: 120*(16*25+1) = 48120 trainable parameters and connections (Fully connected)

LeNet 5, Layer C5



Layer F6: 84 fully connected units. 84*(120+1)=10164 trainable parameters and connections.

Output layer: 10RBF (One for each digit)

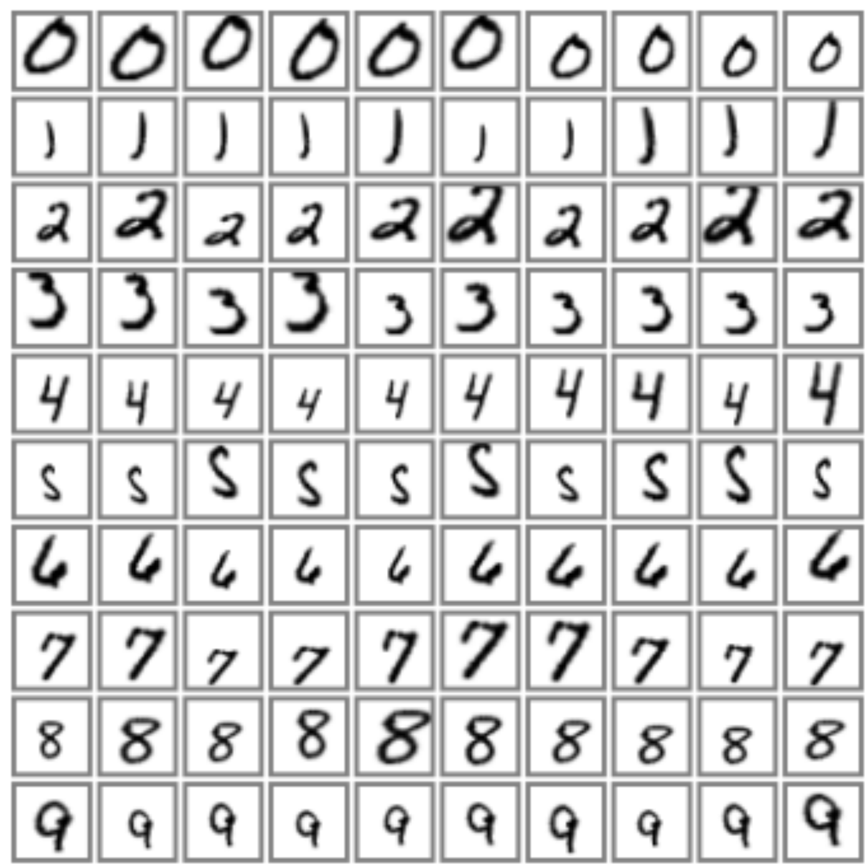
84=7x12, stylized image

Weight update: Backpropagation

MINIST Dataset

60,000 original datasets

Test error: 0.95%



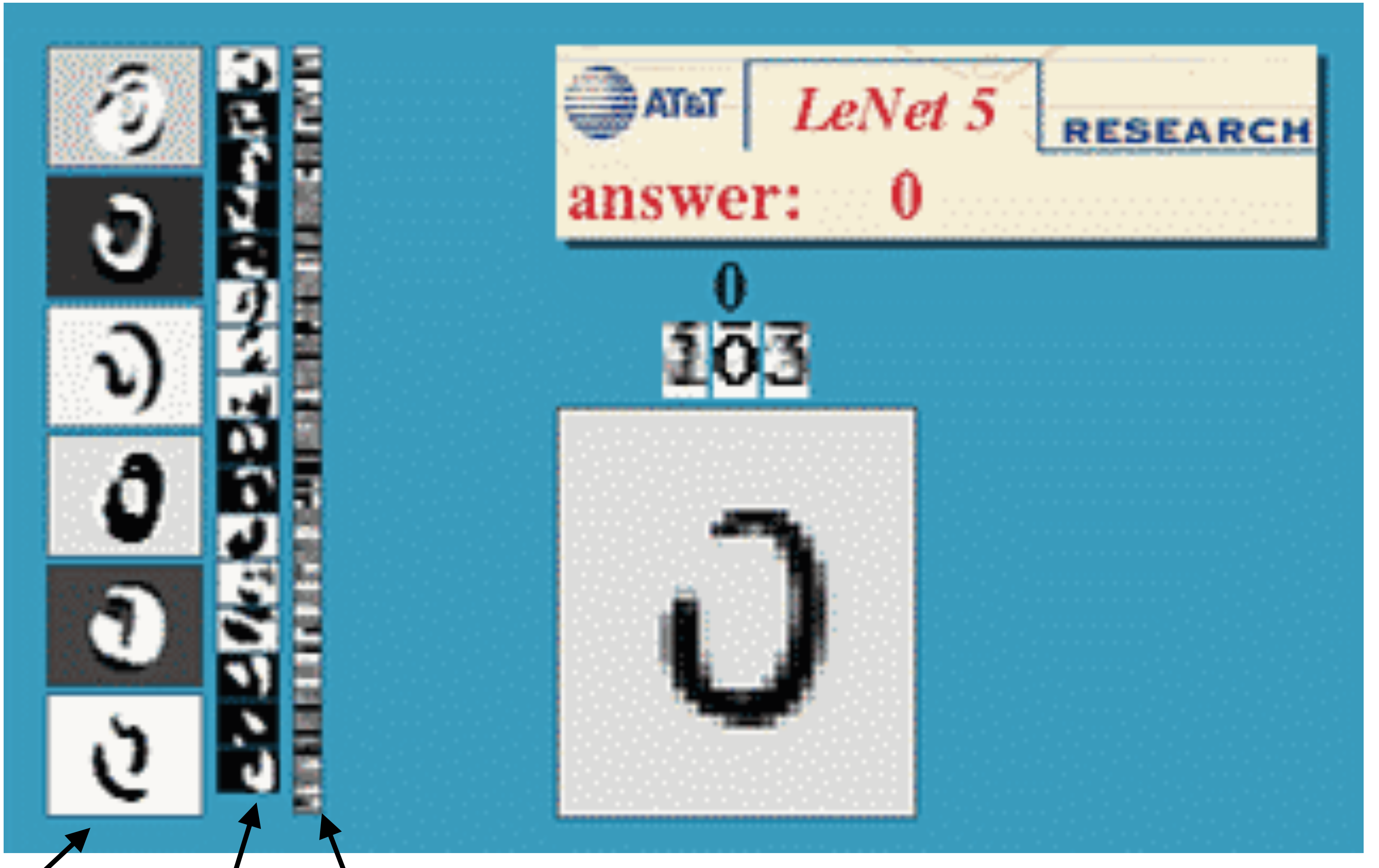
540,000 artificial distortions

+ 60,000 original

Test error: 0.8%

Misclassified examples

LeNet 5 in Action

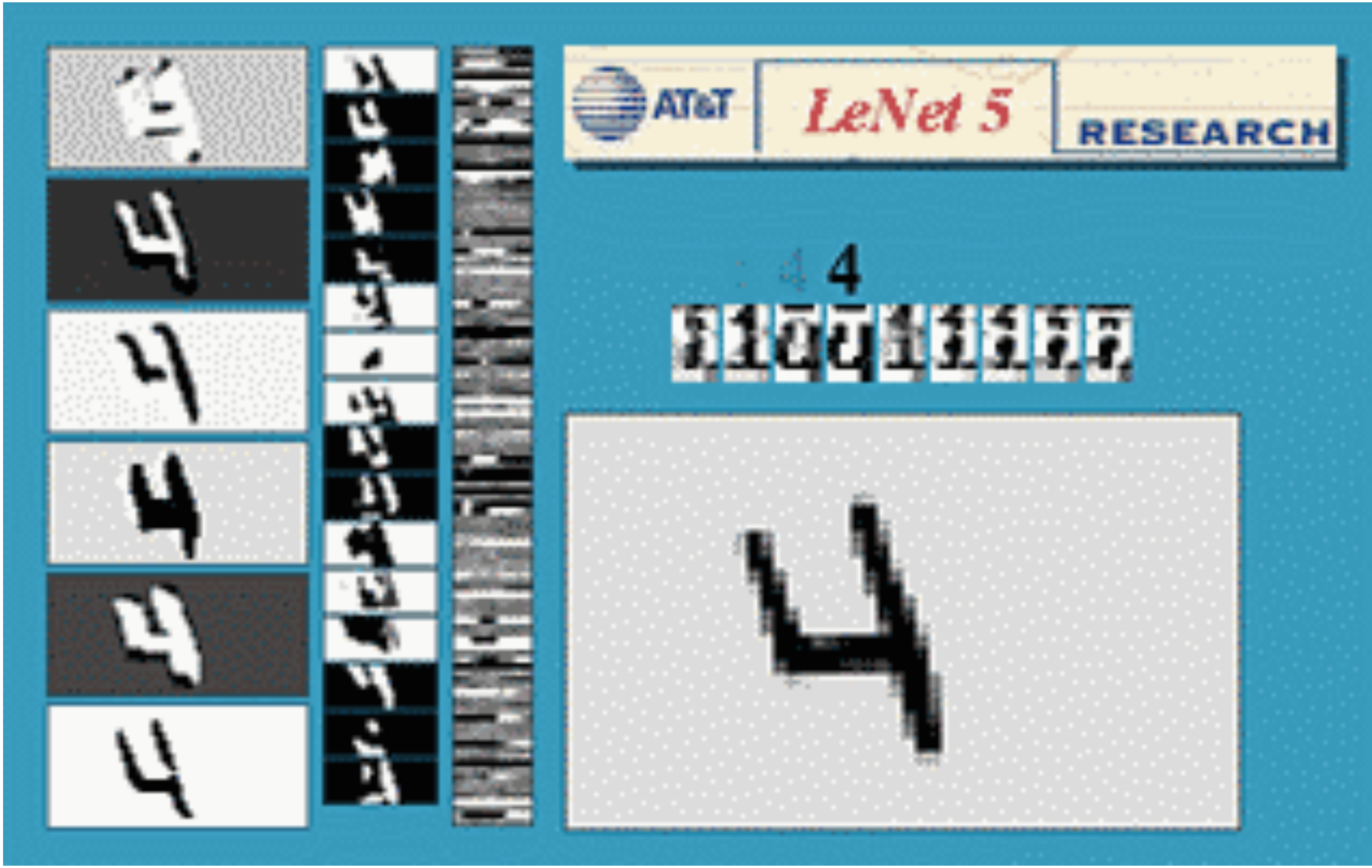


C1 C3 S4

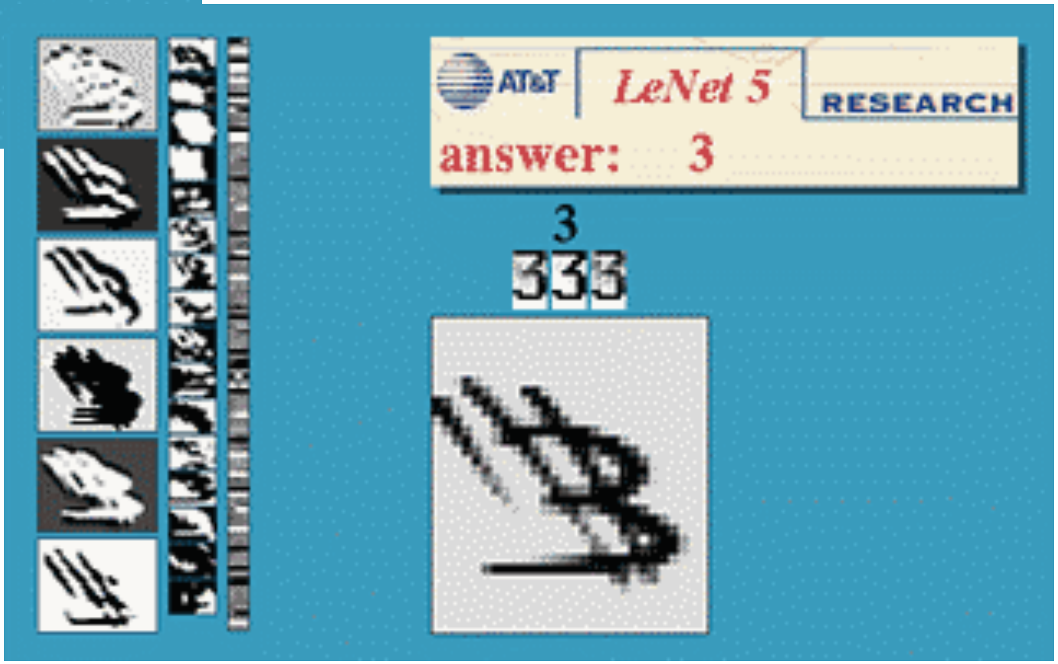
Input

LeNet 5, Shift invariance

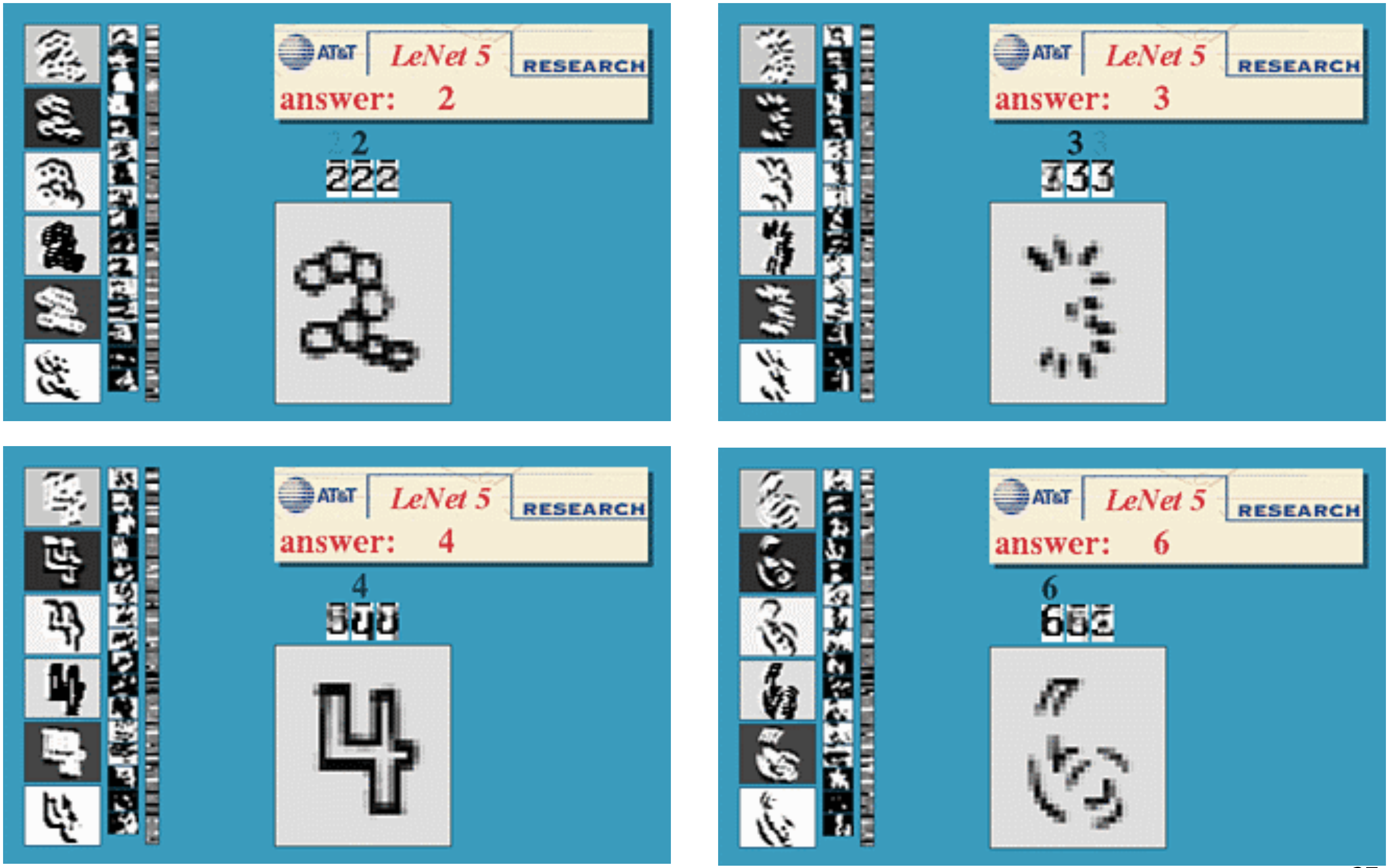
LeNet 5, Rotation invariance



LeNet 5, Nosie resistance



LeNet 5, Unusual Patterns



ImageNet Classification with Deep Convolutional Neural Networks

Alex Krizhevsky, Ilya Sutskever, Geoffrey Hinton,

Advances in Neural Information Processing Systems 2012

ImageNet

- 15M images
- 22K categories
- Images collected from Web
- Human labelers (Amazon's Mechanical Turk crowd-sourcing)
- ImageNet Large Scale Visual Recognition Challenge (ILSVRC-2010)
  - 1K categories
  - 1.2M training images (~1000 per category)
  - 50,000 validation images
  - 150,000 testing images
- RGB images
- Variable-resolution, but this architecture scales them to 256x256 size

ImageNet

Classification goals:

- Make 1 guess about the label (Top-1 error)
- make 5 guesses about the label (Top-5 error)

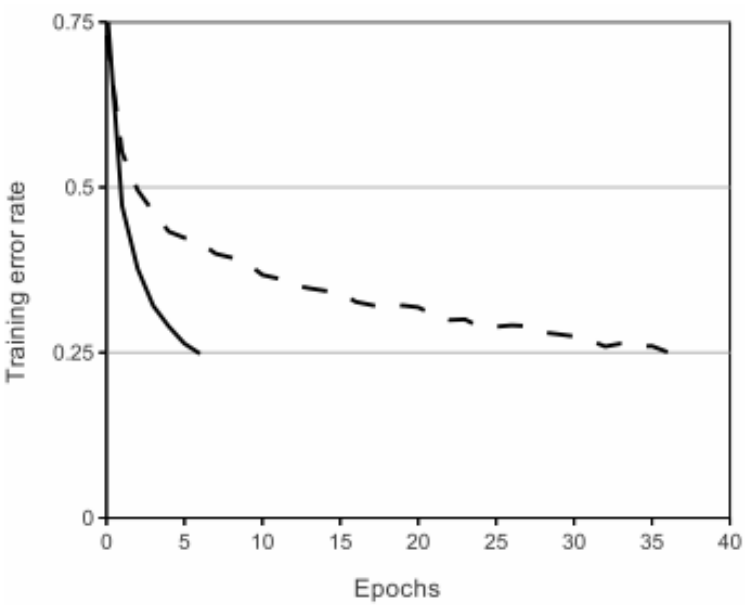
The Architecture

Typical nonlinearities: $f(x) = \tanh(x)$

$$f(x) = (1 + e^{-x})^{-1}$$

Here, however, Rectified Linear Units (ReLU) are used: $f(x) = \max(0, x)$

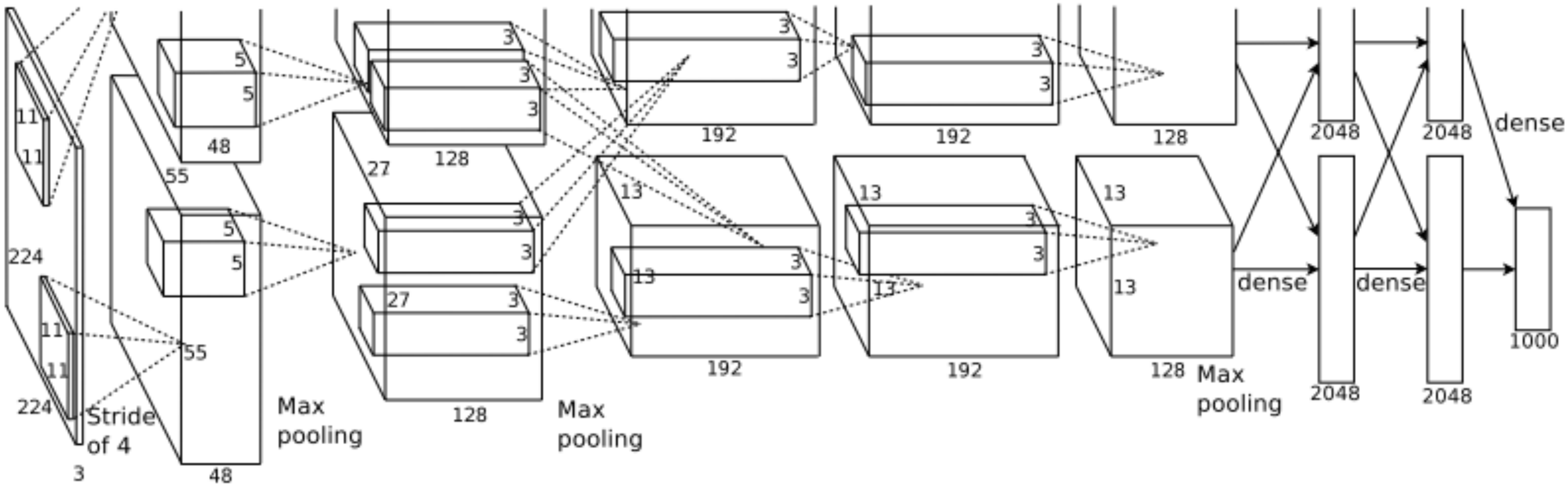
Empirical observation: Deep convolutional neural networks with ReLUs train several times faster than their equivalents with tanh units



A four-layer convolutional neural network with ReLUs (solid line) reaches a 25% training error rate on CIFAR-10 six times faster than an equivalent network with tanh neurons

(dashed line)

The Architecture



The first convolutional layer filters the 224×224×3 input image with 96 kernels of size 11×11×3 with a stride of 4 pixels (this is the distance between the receptive field centers of neighboring neurons in the kernel map. 224/4=56

The pooling layer: form of non-linear down-sampling. Max-pooling partitions the input image into a set of rectangles and, for each such sub-region, outputs the maximum value

The Architecture

- Trained with stochastic gradient descent
- on two NVIDIA GTX 580 3GB GPUs
- for about a week

- 650,000 neurons
- 60,000,000 parameters
- 630,000,000 connections
- 5 convolutional layer, 3 fully connected layer
- Final feature layer: 4096-dimensional

Data Augmentation

The easiest and most common method to **reduce overfitting** on image data is to artificially **enlarge the dataset** using label-preserving transformations.

We employ two distinct forms of data augmentation:

- image translation
- horizontal reflections
- changing RGB intensities

Dropout

- We know that combining different models can be very useful (Mixture of experts, majority voting, boosting, etc)

- Training many different models, however, is very time consuming.

The solution:

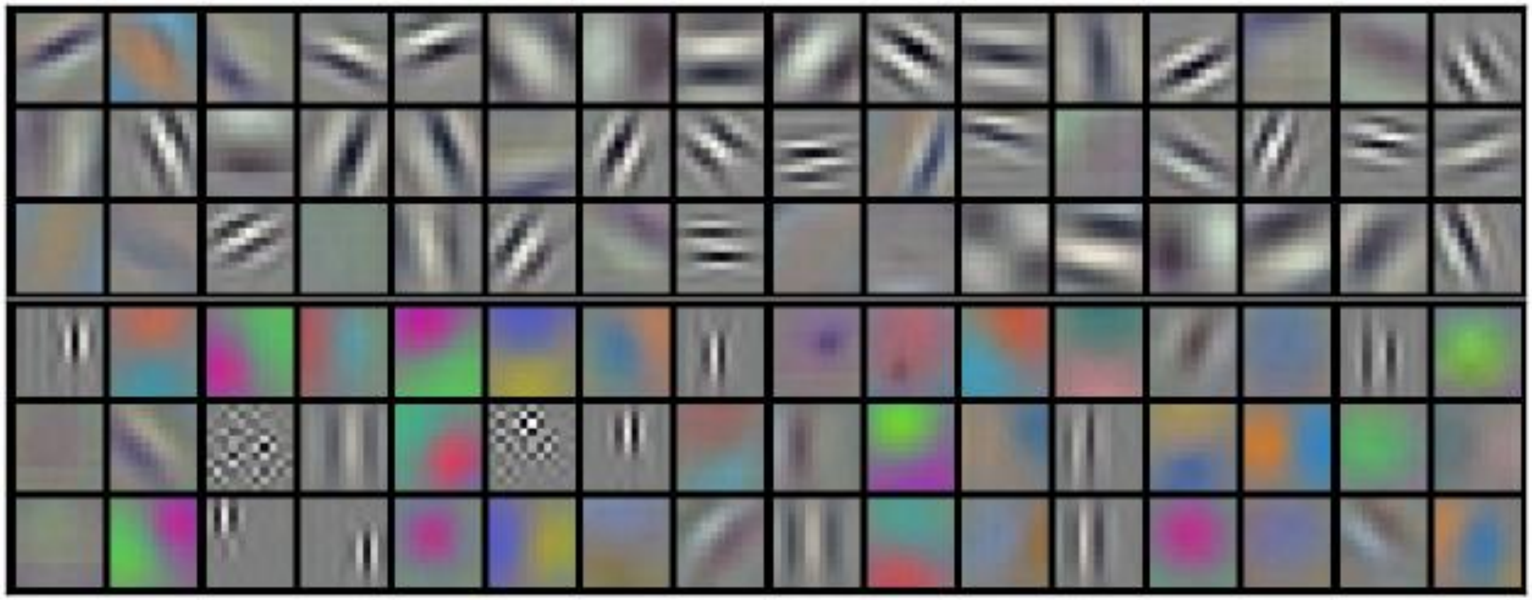
Dropout: set the output of each hidden neuron to zero w.p. 0.5.

Dropout

Dropout: set the output of each hidden neuron to zero w.p. 0.5.

- The neurons which are “dropped out” in this way do not contribute to the forward pass and do not participate in backpropagation.
- So every time an input is presented, the neural network samples a different architecture, but all these architectures share weights.
- This technique reduces complex co-adaptations of neurons, since a neuron cannot rely on the presence of particular other neurons.
- It is, therefore, forced to learn more robust features that are useful in conjunction with many different random subsets of the other neurons.
- Without dropout, our network exhibits substantial overfitting.
- Dropout roughly doubles the number of iterations required to converge.

The first convolutional layer



96 convolutional kernels of size 11×11×3 learned by the first convolutional layer on the 224×224×3 input images.

The top 48 kernels were learned on GPU1 while the bottom 48 kernels were learned on GPU2

Looks like Gabor wavelets, ICA filters…

Results

Results on the test data:

top-1 error rate: 37.5%

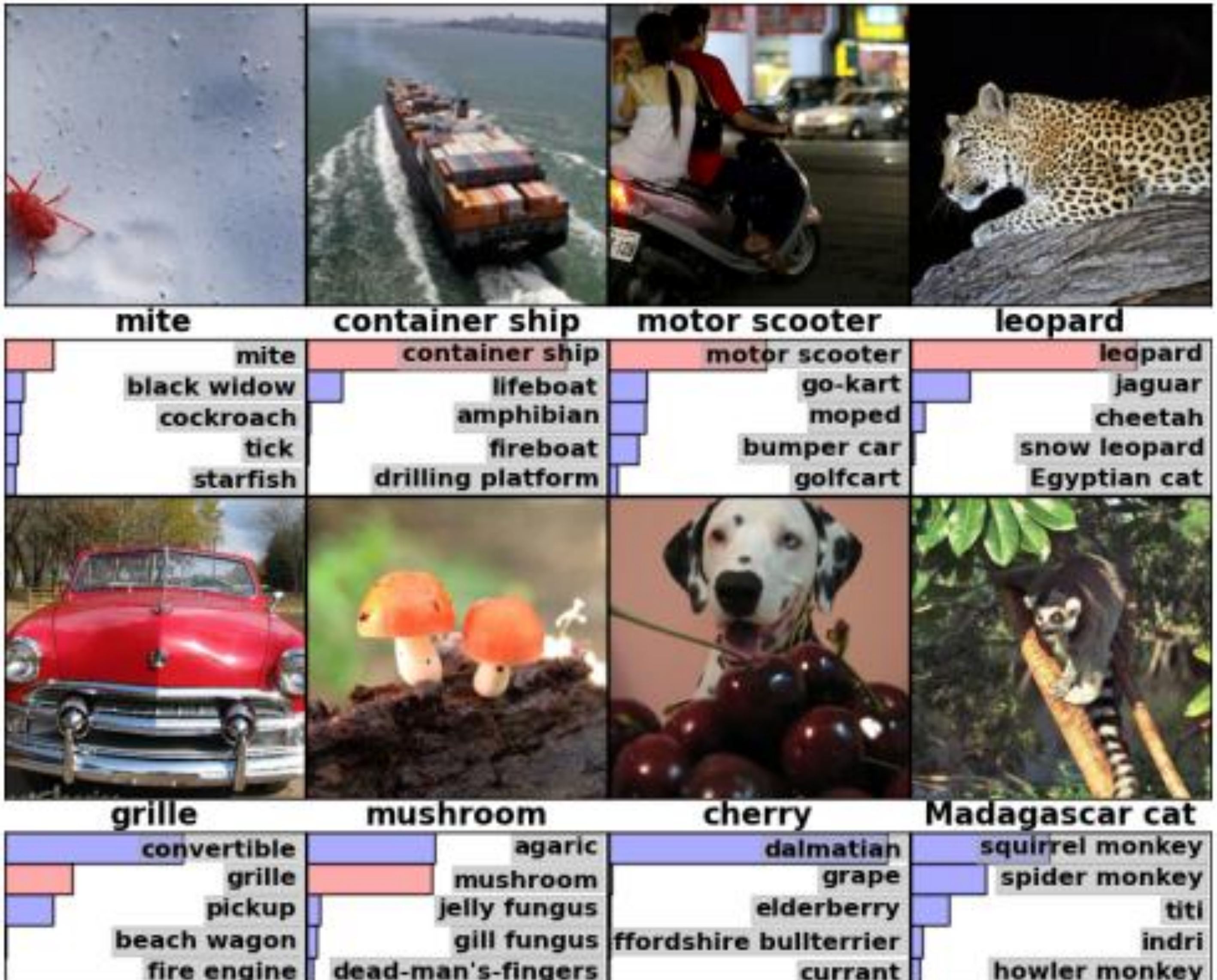
top-5 error rate: 17.0%

ILSVRC-2012 competition:

15.3% accuracy

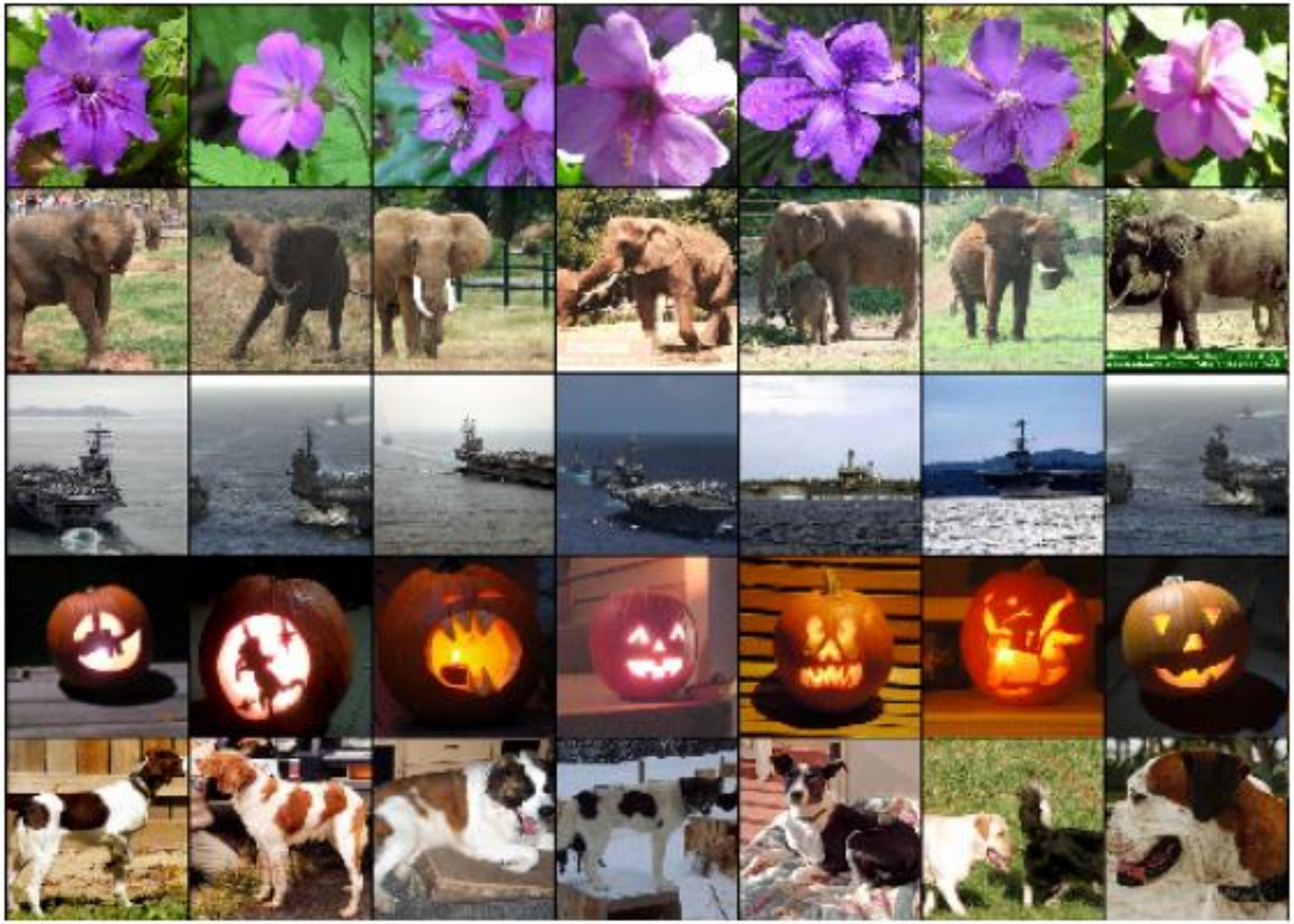
2nd best team: 26.2% accuracy

Results



mite
mite
black widow
cockroach
tick
starfish
container ship
container ship
lifeboat
amphibian
fireboat
drilling platform
motor scooter
motor scooter
go-kart
moped
bumper car
golfcart
leopard
leopard
jaguar
cheetah
snow leopard
Egyptian cat
grille
convertible
grille
pickup
beach wagon
fire engine
mushroom
agaric
mushroom
jelly fungus
gill fungus
dead-man's-fingers
cherry
dalmatian
grape
elderberry
ffordshire bullterrier
currant
Madagascar cat
squirrel monkey
spider monkey
titi
indri
howler monkey

Results: Image similarity



Test column

six training images that produce feature vectors in the last hidden layer with the smallest Euclidean distance from the feature vector for the test image.