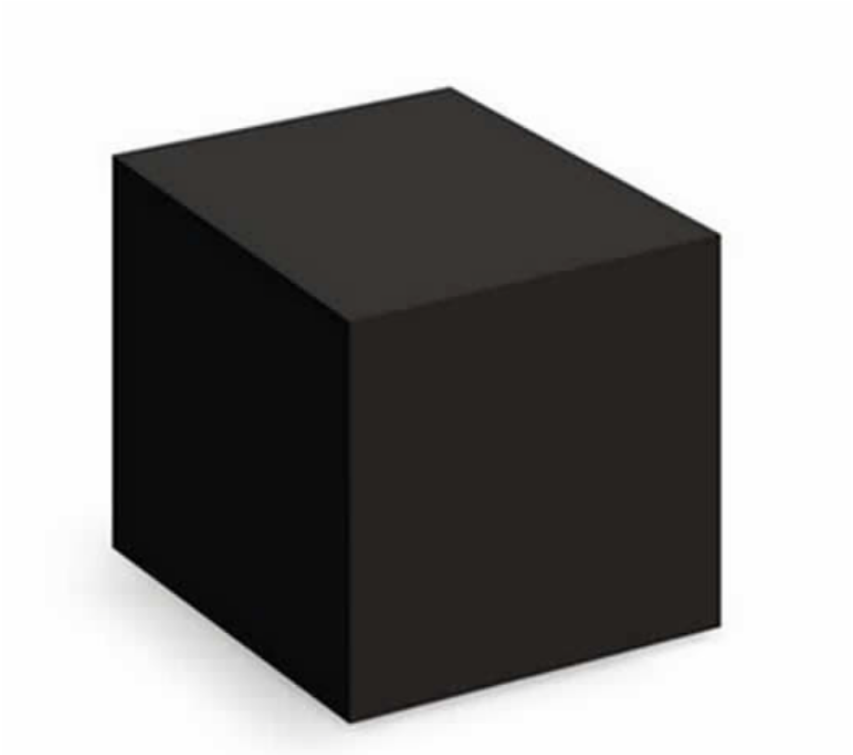
Open the Magic Box of Deep Learning for Image Classification

Zhe Li

The University of Iowa

Monday 11th September, 2017



Is this black box magic?

- 1,000 different items: cat, dog, car, truck, $\cdots$

Is this black box magic?

- 1,000 different items: cat, dog, car, truck, $\cdots$
- How about randomly guess when you have 5 chances?

Is this black box magic?

- 1,000 different items: cat, dog, car, truck, $\cdots$
- How about randomly guess when you have 5 chances?
  - Accuracy percentage: 0.5%

Is this black box magic?

- 1,000 different items: cat, dog, car, truck, $\cdots$
- How about randomly guess when you have 5 chances?
  - Accuracy percentage: 0.5%
- What about the traditional shallow learning?

Is this black box magic?

- 1,000 different items: cat, dog, car, truck, $\cdots$
- How about randomly guess when you have 5 chances?
  - Accuracy percentage: 0.5%
- What about the traditional shallow learning?
  - Accuracy percentage: 74.3%
- What about this magic box?

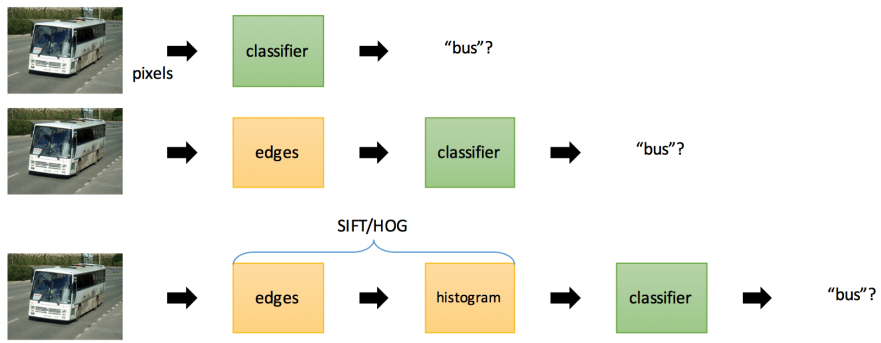
Is this black box magic?

- 1,000 different items: cat, dog, car, truck, $\cdots$
- How about randomly guess when you have 5 chances?
  - Accuracy percentage: 0.5%
- What about the traditional shallow learning?
  - Accuracy percentage: 74.3%
- What about this magic box?
  - Accuracy percentage: 84.7%(the-state-of-the-art).

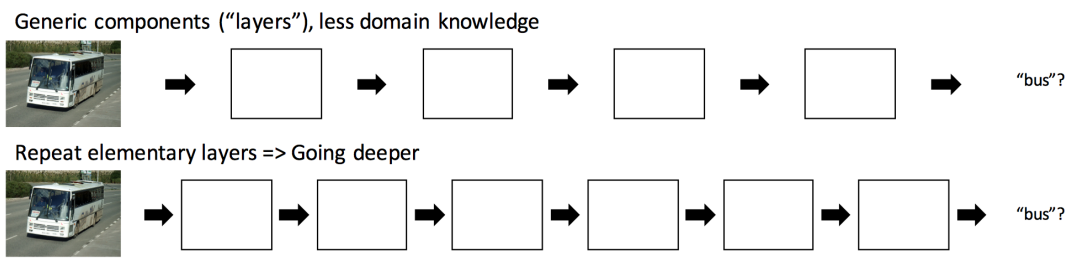
Outline

What is deep learning?

- Shallow Learning:

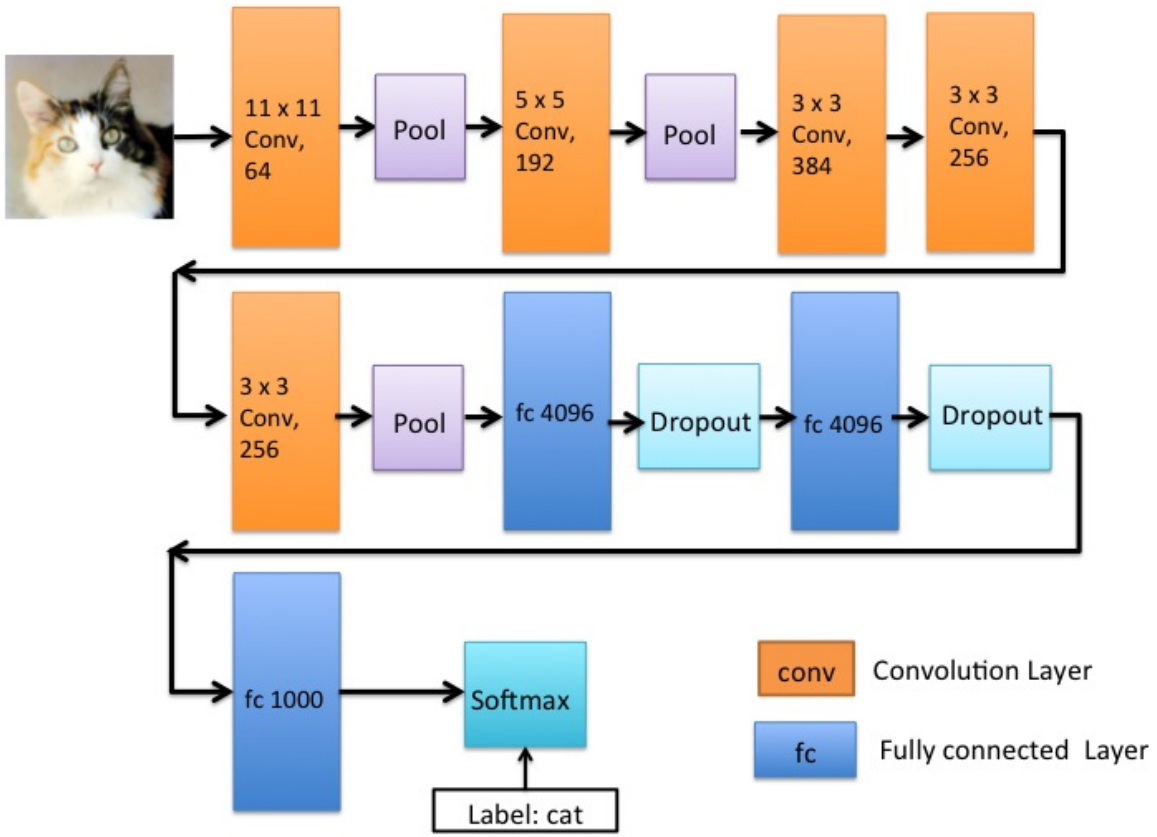


- Deep Learning:

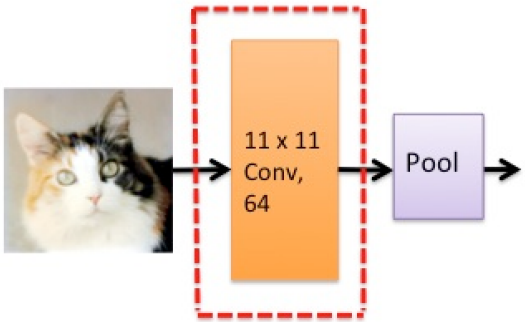


Outline

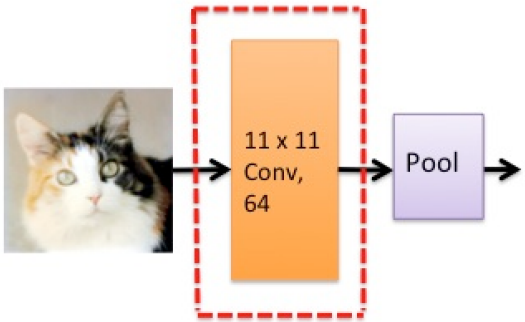
AlexNet [A Krizhevsky et al, 2012]

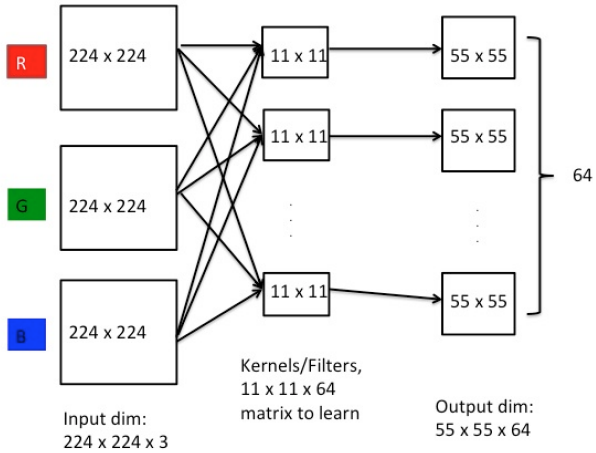


Convolution Layer [Y Lecun et al, 1998]

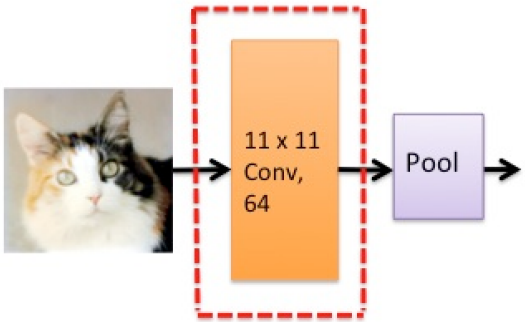


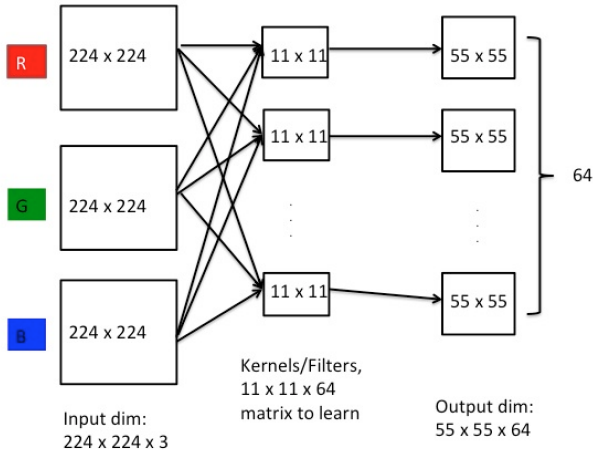
Convolution Layer [Y Lecun et al, 1998]

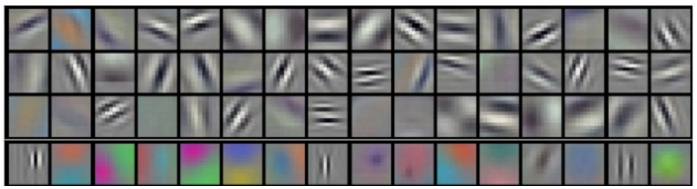




Convolution Layer [Y Lecun et al, 1998]

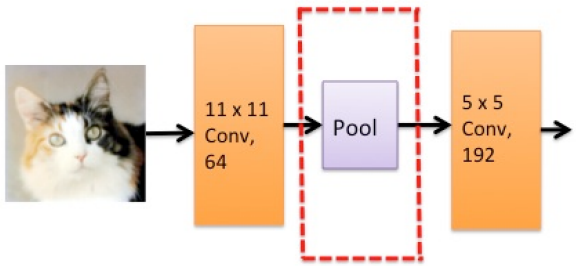




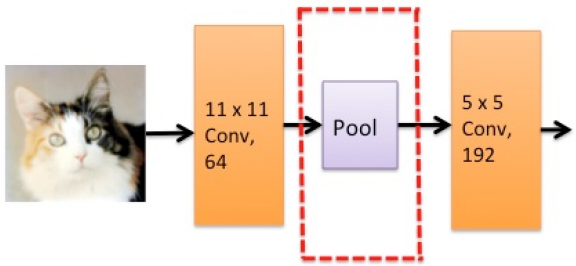


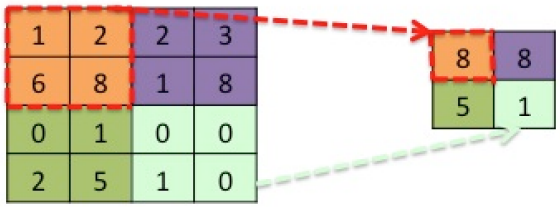
Input Dim	$224 \times 224 \times 3$
Output Dim	$55 \times 55 \times 64$
Num of parameters	$11 \times 11 \times 64 + 64$

Pooling Layer



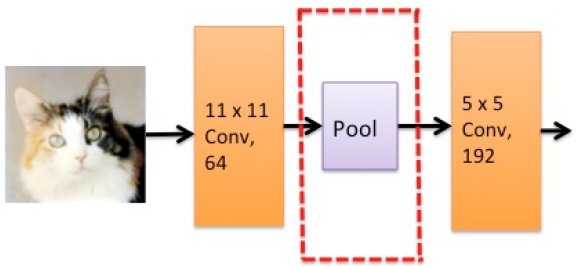
Pooling Layer

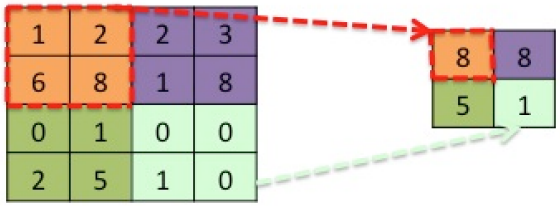




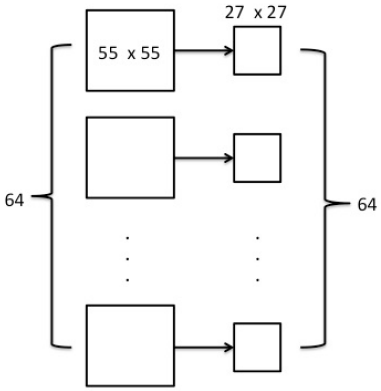
Input Before Pooling Output After Pooling

Pooling Layer

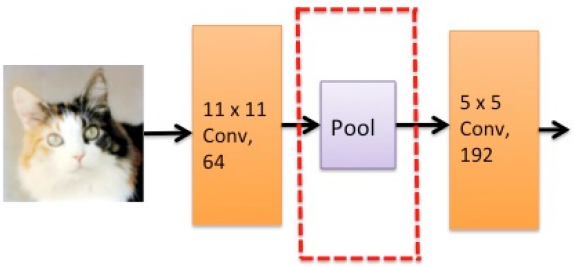


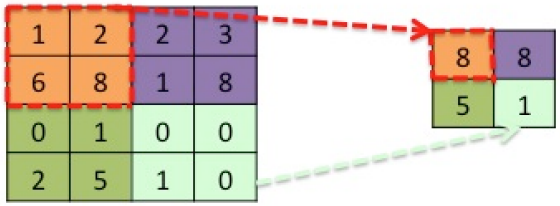


Input Before Pooling Output After Pooling

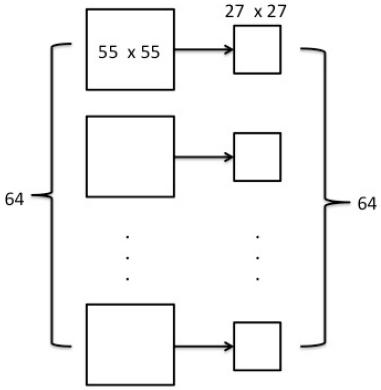


Pooling Layer



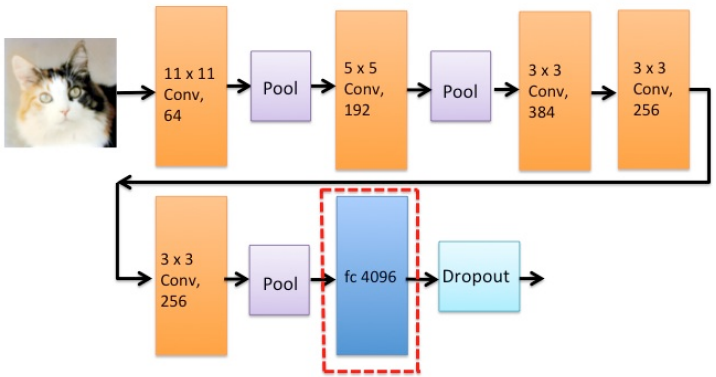


Input Before Pooling Output After Pooling

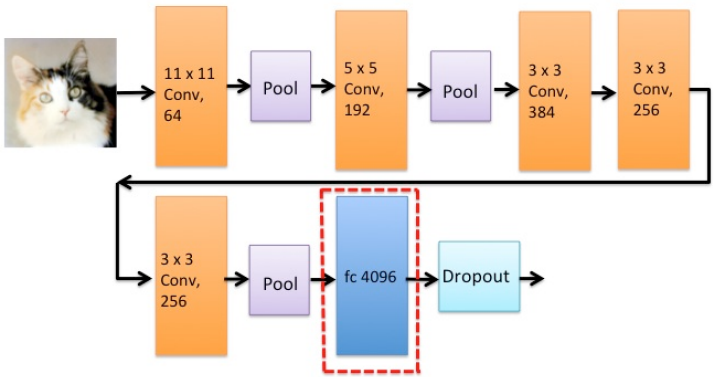


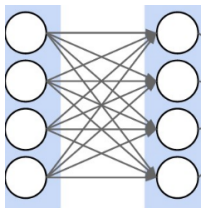
Input Dim	$55 \times 55 \times 64$
Output Dim	$27 \times 27 \times 64$
Num of parameters	0

Fully Connected Layer



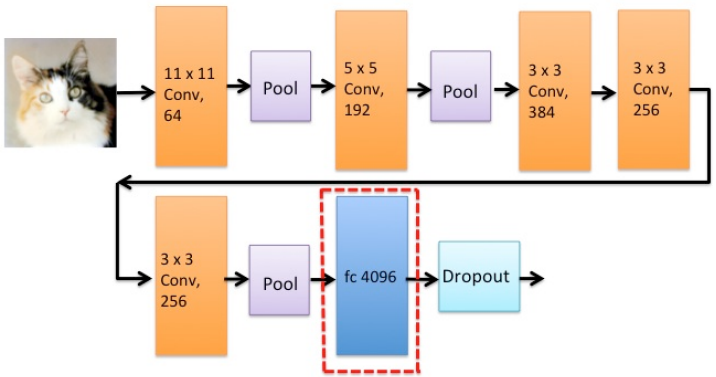
Fully Connected Layer

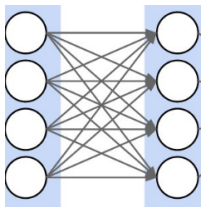




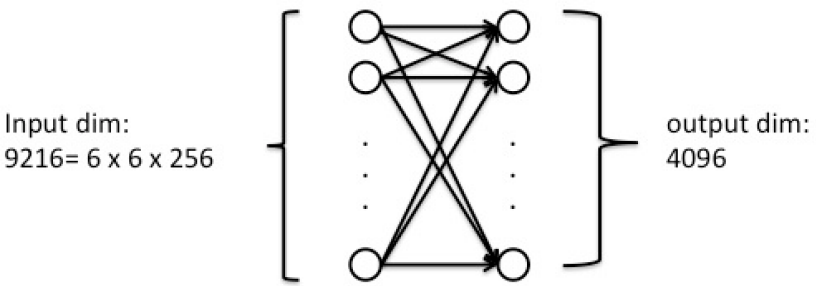
- 4×4 weight matrix

Fully Connected Layer

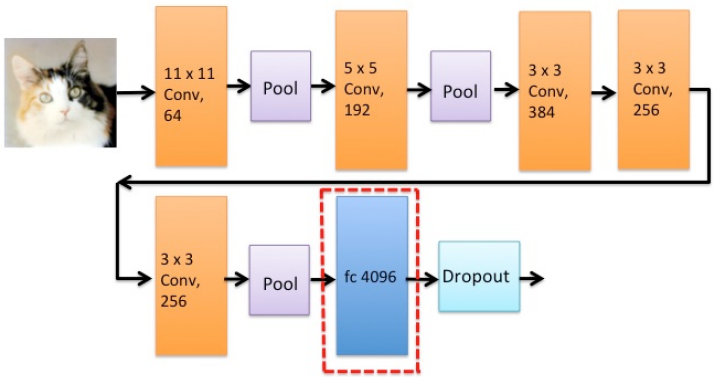


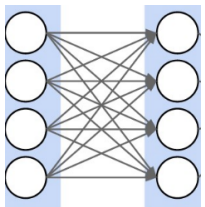


- 4×4 weight matrix

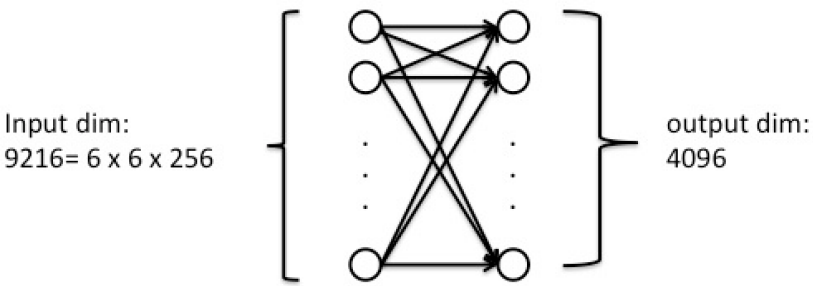


Fully Connected Layer



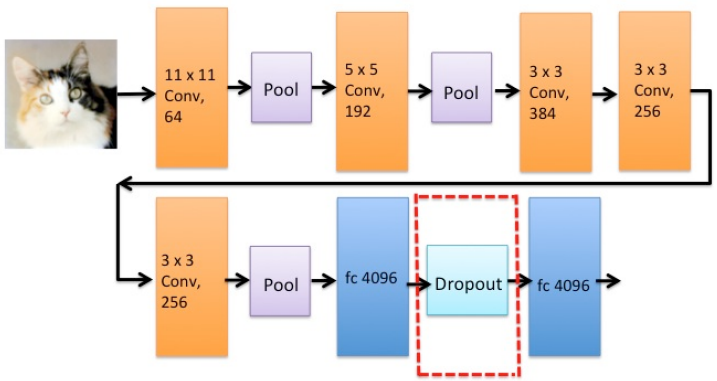


- 4×4 weight matrix

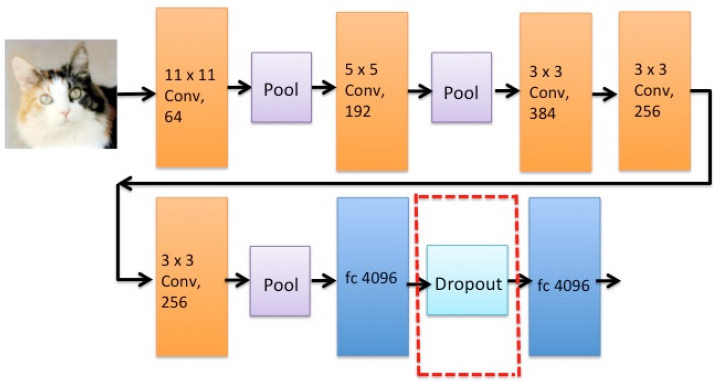


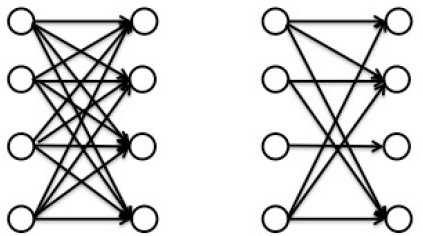
input dim	$6 \times 6 \times 256$
output dim	4096
num of parameters	$6 \times 6 \times 256 \times 4096$

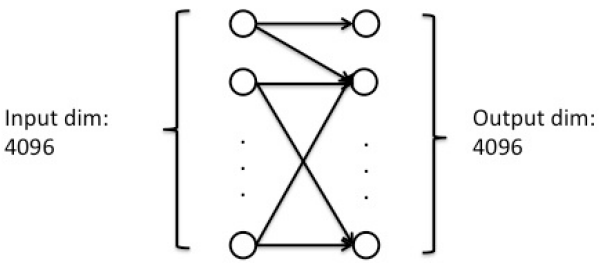
Dropout Layer[G E Hinton et al, 2012]



Dropout Layer[G E Hinton et al, 2012]







input dim	4096
output dim	4096
num of parameters	0

Summary of Each Layer

	Input dim	Output dim	Num of parameters
conv1	$224 \times 224 \times 3$	$55 \times 55 \times 64$	$11 \times 11 \times 64 + 64$

Summary of Each Layer

	Input dim	Output dim	Num of parameters
conv1	$224 \times 224 \times 3$	$55 \times 55 \times 64$	$11 \times 11 \times 64 + 64$
pool1	$55 \times 55 \times 64$	$27 \times 27 \times 64$	0

Summary of Each Layer

	Input dim	Output dim	Num of parameters
conv1	$224 \times 224 \times 3$	$55 \times 55 \times 64$	$11 \times 11 \times 64 + 64$
pool1	$55 \times 55 \times 64$	$27 \times 27 \times 64$	0
conv2	$27 \times 27 \times 64$	$27 \times 27 \times 192$	$5 \times 5 \times 192 + 192$

Summary of Each Layer

	Input dim	Output dim	Num of parameters
conv1	$224 \times 224 \times 3$	$55 \times 55 \times 64$	$11 \times 11 \times 64 + 64$
pool1	$55 \times 55 \times 64$	$27 \times 27 \times 64$	0
conv2	$27 \times 27 \times 64$	$27 \times 27 \times 192$	$5 \times 5 \times 192 + 192$
pool2	$27 \times 27 \times 192$	$13 \times 13 \times 192$	0

Summary of Each Layer

	Input dim	Output dim	Num of parameters
conv1	$224 \times 224 \times 3$	$55 \times 55 \times 64$	$11 \times 11 \times 64 + 64$
pool1	$55 \times 55 \times 64$	$27 \times 27 \times 64$	0
conv2	$27 \times 27 \times 64$	$27 \times 27 \times 192$	$5 \times 5 \times 192 + 192$
pool2	$27 \times 27 \times 192$	$13 \times 13 \times 192$	0
conv3	$13 \times 13 \times 192$	$13 \times 13 \times 384$	$3 \times 3 \times 384 + 384$

Summary of Each Layer

	Input dim	Output dim	Num of parameters
conv1	$224 \times 224 \times 3$	$55 \times 55 \times 64$	$11 \times 11 \times 64 + 64$
pool1	$55 \times 55 \times 64$	$27 \times 27 \times 64$	0
conv2	$27 \times 27 \times 64$	$27 \times 27 \times 192$	$5 \times 5 \times 192 + 192$
pool2	$27 \times 27 \times 192$	$13 \times 13 \times 192$	0
conv3	$13 \times 13 \times 192$	$13 \times 13 \times 384$	$3 \times 3 \times 384 + 384$
conv4	$13 \times 13 \times 384$	$13 \times 13 \times 256$	$3 \times 3 \times 256 + 256$

Summary of Each Layer

	Input dim	Output dim	Num of parameters
conv1	$224 \times 224 \times 3$	$55 \times 55 \times 64$	$11 \times 11 \times 64 + 64$
pool1	$55 \times 55 \times 64$	$27 \times 27 \times 64$	0
conv2	$27 \times 27 \times 64$	$27 \times 27 \times 192$	$5 \times 5 \times 192 + 192$
pool2	$27 \times 27 \times 192$	$13 \times 13 \times 192$	0
conv3	$13 \times 13 \times 192$	$13 \times 13 \times 384$	$3 \times 3 \times 384 + 384$
conv4	$13 \times 13 \times 384$	$13 \times 13 \times 256$	$3 \times 3 \times 256 + 256$
conv5	$13 \times 13 \times 256$	$13 \times 13 \times 256$	$3 \times 3 \times 256 + 256$

Summary of Each Layer

	Input dim	Output dim	Num of parameters
conv1	$224 \times 224 \times 3$	$55 \times 55 \times 64$	$11 \times 11 \times 64 + 64$
pool1	$55 \times 55 \times 64$	$27 \times 27 \times 64$	0
conv2	$27 \times 27 \times 64$	$27 \times 27 \times 192$	$5 \times 5 \times 192 + 192$
pool2	$27 \times 27 \times 192$	$13 \times 13 \times 192$	0
conv3	$13 \times 13 \times 192$	$13 \times 13 \times 384$	$3 \times 3 \times 384 + 384$
conv4	$13 \times 13 \times 384$	$13 \times 13 \times 256$	$3 \times 3 \times 256 + 256$
conv5	$13 \times 13 \times 256$	$13 \times 13 \times 256$	$3 \times 3 \times 256 + 256$
pool3	$13 \times 13 \times 256$	$6 \times 6 \times 256$	0

Summary of Each Layer

	Input dim	Output dim	Num of parameters
conv1	$224 \times 224 \times 3$	$55 \times 55 \times 64$	$11 \times 11 \times 64 + 64$
pool1	$55 \times 55 \times 64$	$27 \times 27 \times 64$	0
conv2	$27 \times 27 \times 64$	$27 \times 27 \times 192$	$5 \times 5 \times 192 + 192$
pool2	$27 \times 27 \times 192$	$13 \times 13 \times 192$	0
conv3	$13 \times 13 \times 192$	$13 \times 13 \times 384$	$3 \times 3 \times 384 + 384$
conv4	$13 \times 13 \times 384$	$13 \times 13 \times 256$	$3 \times 3 \times 256 + 256$
conv5	$13 \times 13 \times 256$	$13 \times 13 \times 256$	$3 \times 3 \times 256 + 256$
pool3	$13 \times 13 \times 256$	$6 \times 6 \times 256$	0
fc1	$9216 = 6 \times 6 \times 256$	4096	9216×4096

Summary of Each Layer

	Input dim	Output dim	Num of parameters
conv1	$224 \times 224 \times 3$	$55 \times 55 \times 64$	$11 \times 11 \times 64 + 64$
pool1	$55 \times 55 \times 64$	$27 \times 27 \times 64$	0
conv2	$27 \times 27 \times 64$	$27 \times 27 \times 192$	$5 \times 5 \times 192 + 192$
pool2	$27 \times 27 \times 192$	$13 \times 13 \times 192$	0
conv3	$13 \times 13 \times 192$	$13 \times 13 \times 384$	$3 \times 3 \times 384 + 384$
conv4	$13 \times 13 \times 384$	$13 \times 13 \times 256$	$3 \times 3 \times 256 + 256$
conv5	$13 \times 13 \times 256$	$13 \times 13 \times 256$	$3 \times 3 \times 256 + 256$
pool3	$13 \times 13 \times 256$	$6 \times 6 \times 256$	0
fc1	$9216 = 6 \times 6 \times 256$	4096	9216×4096
dropout1	4096	4096	0

Summary of Each Layer

	Input dim	Output dim	Num of parameters
conv1	$224 \times 224 \times 3$	$55 \times 55 \times 64$	$11 \times 11 \times 64 + 64$
pool1	$55 \times 55 \times 64$	$27 \times 27 \times 64$	0
conv2	$27 \times 27 \times 64$	$27 \times 27 \times 192$	$5 \times 5 \times 192 + 192$
pool2	$27 \times 27 \times 192$	$13 \times 13 \times 192$	0
conv3	$13 \times 13 \times 192$	$13 \times 13 \times 384$	$3 \times 3 \times 384 + 384$
conv4	$13 \times 13 \times 384$	$13 \times 13 \times 256$	$3 \times 3 \times 256 + 256$
conv5	$13 \times 13 \times 256$	$13 \times 13 \times 256$	$3 \times 3 \times 256 + 256$
pool3	$13 \times 13 \times 256$	$6 \times 6 \times 256$	0
fc1	$9216 = 6 \times 6 \times 256$	4096	9216×4096
dropout1	4096	4096	0
fc2	4096	4096	4096×4096

Summary of Each Layer

	Input dim	Output dim	Num of parameters
conv1	$224 \times 224 \times 3$	$55 \times 55 \times 64$	$11 \times 11 \times 64 + 64$
pool1	$55 \times 55 \times 64$	$27 \times 27 \times 64$	0
conv2	$27 \times 27 \times 64$	$27 \times 27 \times 192$	$5 \times 5 \times 192 + 192$
pool2	$27 \times 27 \times 192$	$13 \times 13 \times 192$	0
conv3	$13 \times 13 \times 192$	$13 \times 13 \times 384$	$3 \times 3 \times 384 + 384$
conv4	$13 \times 13 \times 384$	$13 \times 13 \times 256$	$3 \times 3 \times 256 + 256$
conv5	$13 \times 13 \times 256$	$13 \times 13 \times 256$	$3 \times 3 \times 256 + 256$
pool3	$13 \times 13 \times 256$	$6 \times 6 \times 256$	0
fc1	$9216 = 6 \times 6 \times 256$	4096	9216×4096
dropout1	4096	4096	0
fc2	4096	4096	4096×4096
dropout2	4096	4096	0

Summary of Each Layer

	Input dim	Output dim	Num of parameters
conv1	$224 \times 224 \times 3$	$55 \times 55 \times 64$	$11 \times 11 \times 64 + 64$
pool1	$55 \times 55 \times 64$	$27 \times 27 \times 64$	0
conv2	$27 \times 27 \times 64$	$27 \times 27 \times 192$	$5 \times 5 \times 192 + 192$
pool2	$27 \times 27 \times 192$	$13 \times 13 \times 192$	0
conv3	$13 \times 13 \times 192$	$13 \times 13 \times 384$	$3 \times 3 \times 384 + 384$
conv4	$13 \times 13 \times 384$	$13 \times 13 \times 256$	$3 \times 3 \times 256 + 256$
conv5	$13 \times 13 \times 256$	$13 \times 13 \times 256$	$3 \times 3 \times 256 + 256$
pool3	$13 \times 13 \times 256$	$6 \times 6 \times 256$	0
fc1	$9216 = 6 \times 6 \times 256$	4096	9216×4096
dropout1	4096	4096	0
fc2	4096	4096	4096×4096
dropout2	4096	4096	0
fc3	4096	1000	4096×1000

Summary of Each Layer

	Input dim	Output dim	Num of parameters
conv1	$224 \times 224 \times 3$	$55 \times 55 \times 64$	$11 \times 11 \times 64 + 64$
pool1	$55 \times 55 \times 64$	$27 \times 27 \times 64$	0
conv2	$27 \times 27 \times 64$	$27 \times 27 \times 192$	$5 \times 5 \times 192 + 192$
pool2	$27 \times 27 \times 192$	$13 \times 13 \times 192$	0
conv3	$13 \times 13 \times 192$	$13 \times 13 \times 384$	$3 \times 3 \times 384 + 384$
conv4	$13 \times 13 \times 384$	$13 \times 13 \times 256$	$3 \times 3 \times 256 + 256$
conv5	$13 \times 13 \times 256$	$13 \times 13 \times 256$	$3 \times 3 \times 256 + 256$
pool3	$13 \times 13 \times 256$	$6 \times 6 \times 256$	0
fc1	$9216 = 6 \times 6 \times 256$	4096	9216×4096
dropout1	4096	4096	0
fc2	4096	4096	4096×4096
dropout2	4096	4096	0
fc3	4096	1000	4096×1000
softmax	1000	1	0

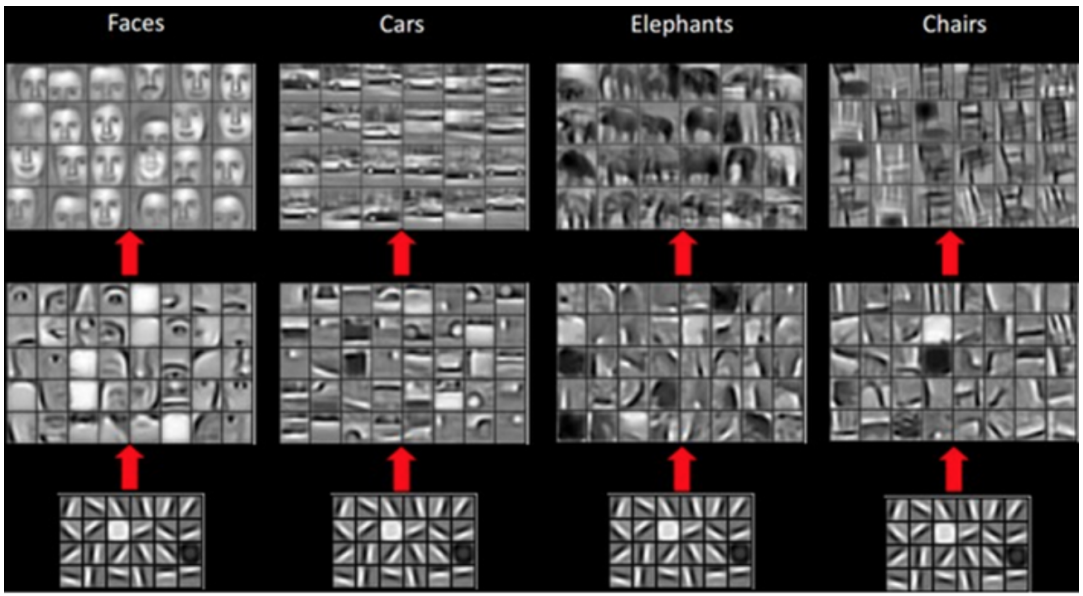
- Totally, the number of parameters: 58,643,712.

Summary of neural network structure

How big is this neural network?

- 8 layers with weights to learn
  - 5 convolution layers
  - 3 fully connected layers
- 3 pooling layers, 2 dropout layers
- Num of parameters: 58.7 millions

Why does this neural network structure work?



- The higher the layer is, the more abstract feature that layer generates.

How to design the neural network structure?

- Totally, experience
- But, the successful layer pattern:
  (conv $\rightarrow$ pool?)*M $\rightarrow$ (fc $\rightarrow$ dropout?)*N

How to design the neural network structure?

- Totally, experience
- But, the successful layer pattern:
  (conv $\rightarrow$ pool?)*M $\rightarrow$ (fc $\rightarrow$ dropout?)*N

You may have questions:

- How many layers?
- Convolution layer: how many kernels? kernel size? and so on
- Pooling Layer: window size? stride?
- Fully Connected Layer: output size?

How to design the neural network structure?

- Totally, experience
- But, the successful layer pattern:
  (conv $\rightarrow$ pool?)*M $\rightarrow$ (fc $\rightarrow$ dropout?)*N

You may have questions:

- How many layers?
- Convolution layer: how many kernels? kernel size? and so on
- Pooling Layer: window size? stride?
- Fully Connected Layer: output size?

Answers:

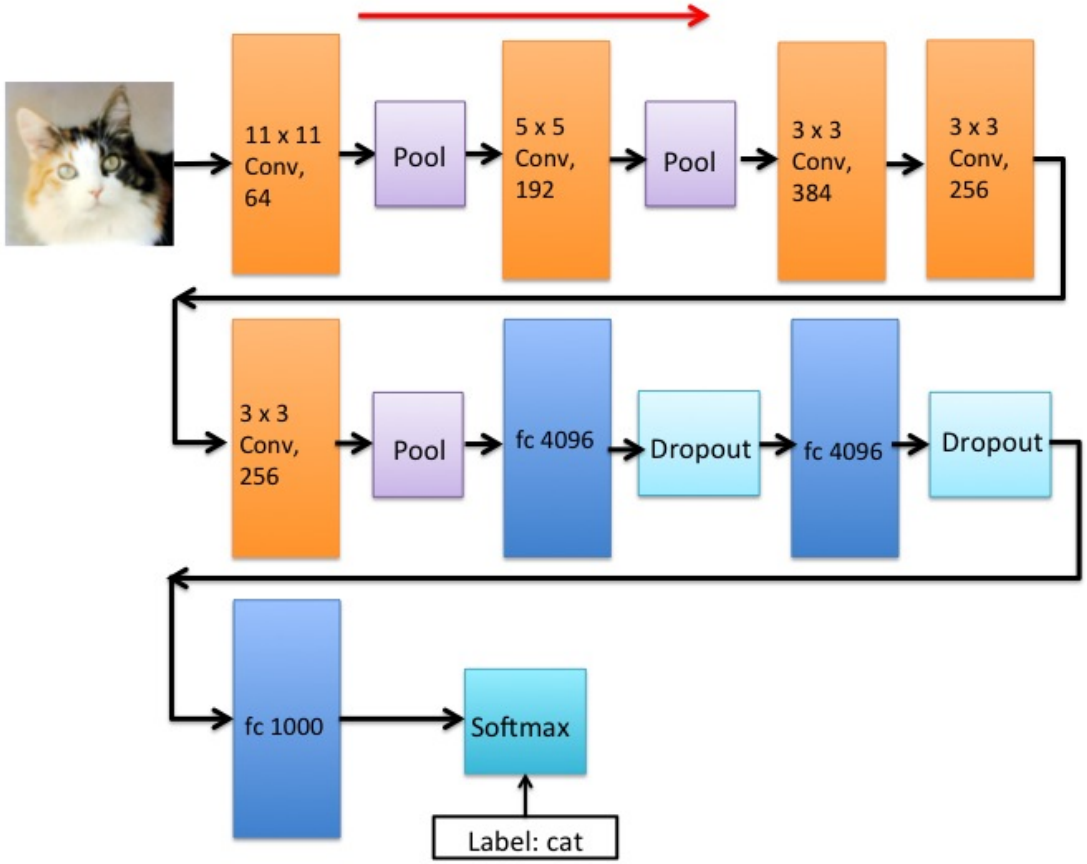
- Graduate student tuning

Outline

Optimization

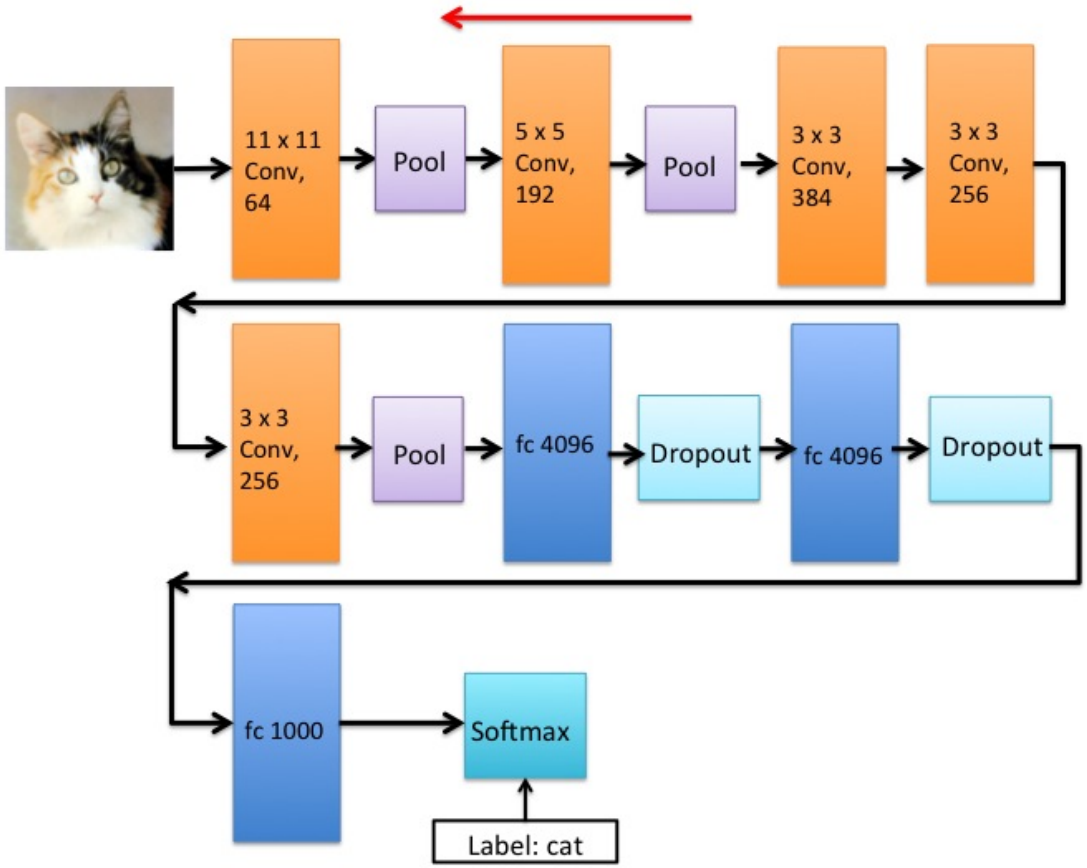
- Initialize the parameter **w** in neural network
- Forward to compute the Loss
- Back Propagation to update the parameter **w**
- Mini-batch + Stochastic Gradient Descent

Forward to compute the loss



Back Propagation to update the parameter **w**

- chain rule for computing gradient



MiniBatch + Stochastic gradient descent

Let $\mathbf{x}_i$ is i^{th} image, y_i is the label associated with i^{th} image.

- Gradient descent

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \frac{\eta_t}{n} \sum_{i=1}^{n} \nabla f(\mathbf{w}_t; \mathbf{x}_i, y_i)$$

MiniBatch + Stochastic gradient descent

Let $\mathbf{x}_i$ is i^{th} image, y_i is the label associated with i^{th} image.

- Gradient descent

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \frac{\eta_t}{n} \sum_{i=1}^{n} \nabla f(\mathbf{w}_t; \mathbf{x}_i, y_i)$$

- Stochastic gradient descent

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \eta_t \nabla f(\mathbf{w}_t; \mathbf{x}_j, y_j)$$

MiniBatch + Stochastic gradient descent

Let $\mathbf{x}_i$ is i^{th} image, y_i is the label associated with i^{th} image.

- Gradient descent

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \frac{\eta_t}{n} \sum_{i=1}^{n} \nabla f(\mathbf{w}_t; \mathbf{x}_i, y_i)$$

- Stochastic gradient descent

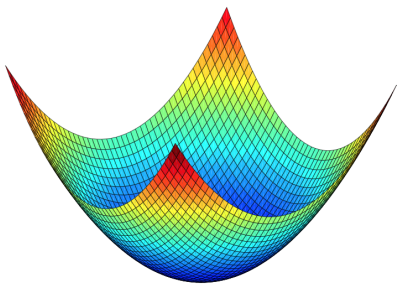
$$\mathbf{w}_{t+1} = \mathbf{w}_t - \eta_t \nabla f(\mathbf{w}_t; \mathbf{x}_j, y_j)$$

- Mini-batch stochastic gradient descent

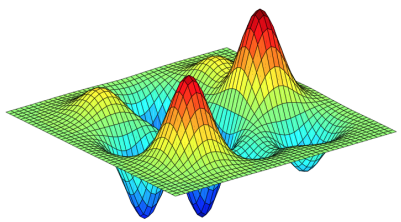
$$\mathbf{w}_{t+1} = \mathbf{w}_t - \frac{\eta_t}{|\mathcal{B}|} \sum_{j \in \mathcal{B}} \nabla f(\mathbf{w}_t; \mathbf{x}_i, y)$$

Why is optimizing hard in deep learning?

- Non-convexity (Less well-established theory)



(a) convex

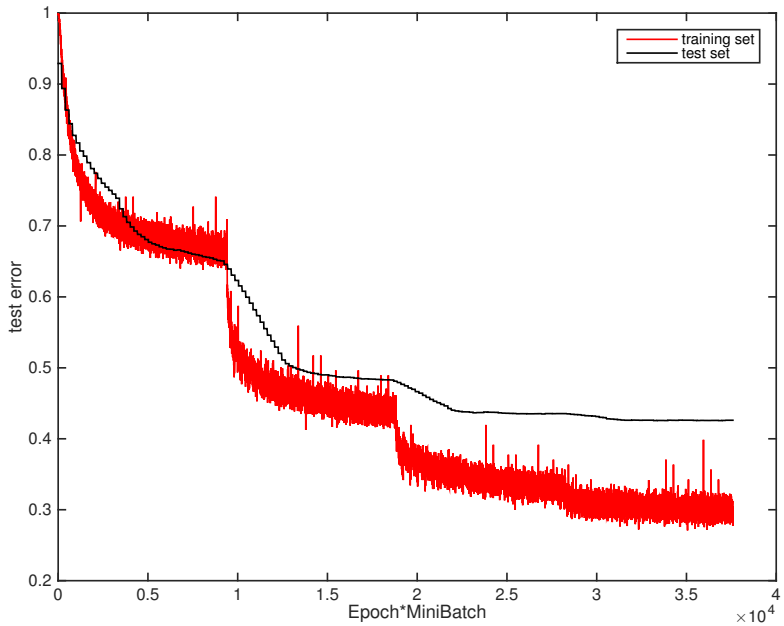


(b) non-convex

- Large parameters (High computational cost)
  - 58.7 million dimensional weight
- Large dataset (High computation cost)
  - 1.4 million images

Practical Strategy

- Decrease the learning rate after a number of iterations
  - Learning rate: $0.01 \rightarrow 0.001 \rightarrow 0.0001 \rightarrow 0.00001$



Practical Pain

- Tuning parameters

Practical Pain

- Tuning parameters

You may have questions:

- What is the initial learning rate?
- When should the learning rate be decreased?
- How many times should learning rate be decreased?
- What is the batch size?

Practical Pain

- Tuning parameters

You may have questions:

- What is the initial learning rate?
- When should the learning rate be decreased?
- How many times should learning rate be decreased?
- What is the batch size?

Answers:

- Graduate student descent algorithm

Outline

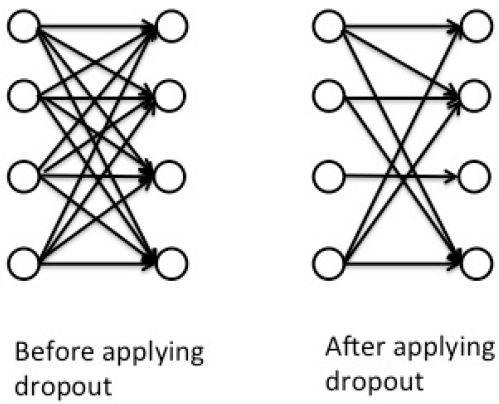
Implementation

- Open source Library: Caffe, tensorflow, cuda-convnet, $\cdots$
- Module-based, easily build new neural network structure


Outline

Dropout

- Recall Dropout Layer: Uniformly at randomly drop out features.



- Is uniformly dropout optimal?
  - Answered the above question in our NIPS 2016 paper.

Improved Dropout

- Dropping out the output of the neuron based on multinomial distribution computed from the training data.

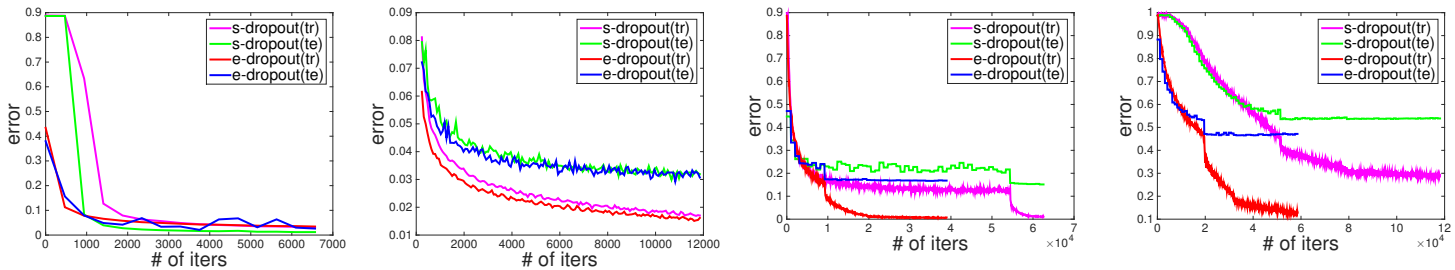


Figure: Evolutional dropout vs standard dropout on four benchmark datasets for deep learning: MNIST, SVHN, CIFAR10, CIFAR100

Where is the idea of the improved dropout from?

- Theoretically, we can prove in shallow learning that based on the following distribution

$$p_i = \frac{\sqrt{\mathrm{E}_{\mathcal{D}}[x_i^2]}}{\sum_{j=1}^{d} \sqrt{\mathrm{E}_{\mathcal{D}}[x_j^2]}}, i = 1, \ldots, d$$

  dropout can leads to a lower test error, where x_i is the i^{th} feature.
- Can we compute the above distribution?
  - No, we don't know the feature distribution.

Where is the idea of the improved dropout from?

- Empirically, use the following distribution:

$$p_i = \frac{\sqrt{\frac{1}{n}\sum_{j=1}^{n}[[\mathbf{x}_j]_i^2]}}{\sum_{i'=1}^{d}\sqrt{\frac{1}{n}\sum_{j=1}^{n}[[\mathbf{x}_j]_{i'}^2]}}$$

- How about the empirical performance on shallow learning?

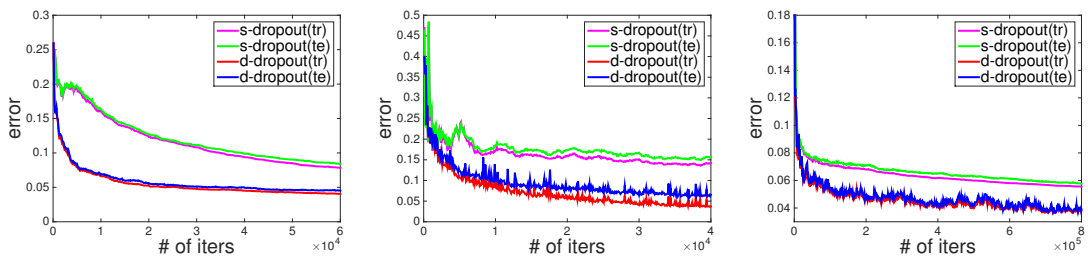


Figure: data-dependent dropout vs standard dropout on three data sets (real-sim, news20, RCV1) for logistic regression

Where is the idea of the improved dropout from?

Could we use this idea to deep learning?

- Yes, but not efficient to compute the distribution in deep learning.
- compute the distribution on the mini-batch data points on each iteration.

Improved dropout

- Four different datasets MINST, SVHN, CIFAR10, CIFAR100

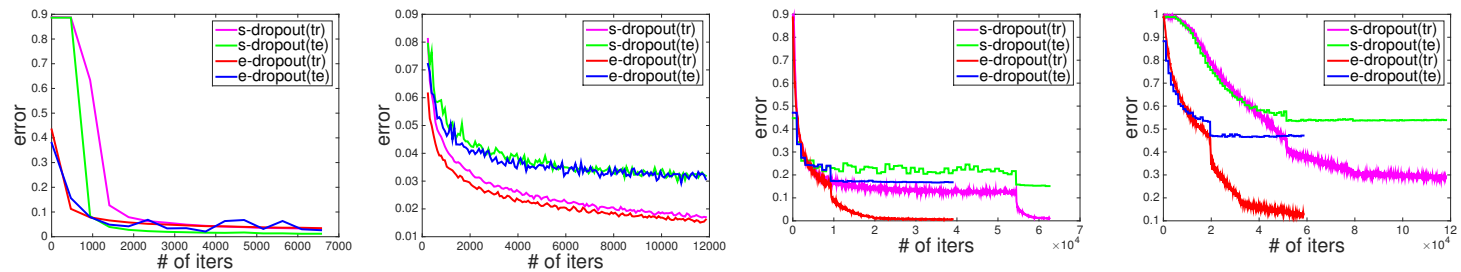


Figure: Evolutional dropout vs standard dropout on four benchmark datasets for deep learning

Thank You!