

# Open the Magic Box of Deep Learning for Image Classification

PhD Candidate: Zhe Li  
Advisor: Prof. Tianbao Yang

The University of Iowa

Friday 8<sup>th</sup> September, 2017

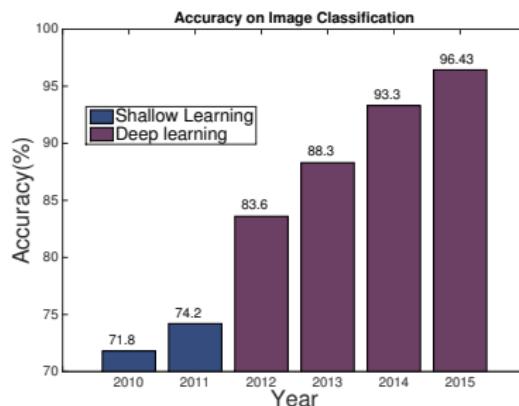
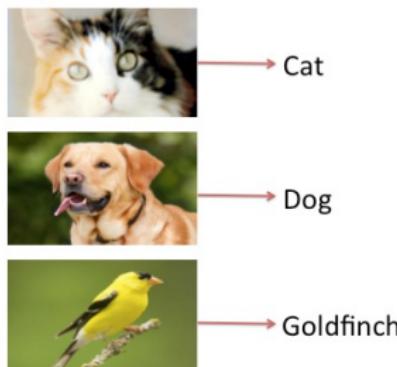
# The magic of deep learning

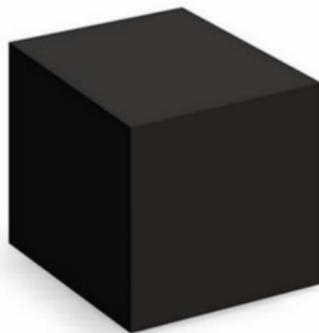
- AlphaGo Game



# The magic of deep learning

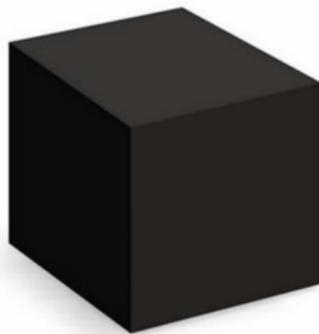
## • Image Classification





• Bob:

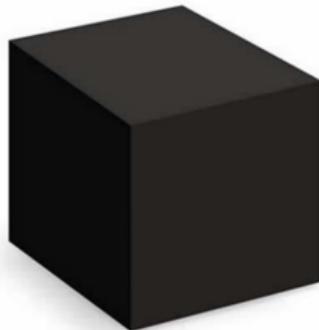
Magic Box of Deep Learning.  
But, it is black



What is inside?

• Bob:

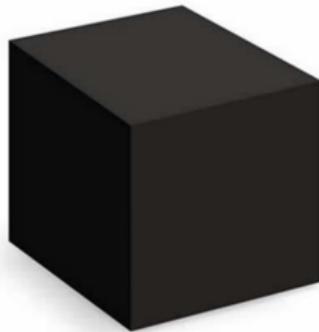
Magic Box of Deep Learning.  
But, it is black



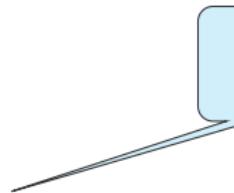
Magic Box of Deep Learning.  
But, it is black

- Bob:
- Allice:

How do  
it work



- Bob:
- Allice:
- A PhD student:



Magic Box of Deep Learning.  
But, it is black

1 What is inside?

2 How does it work?

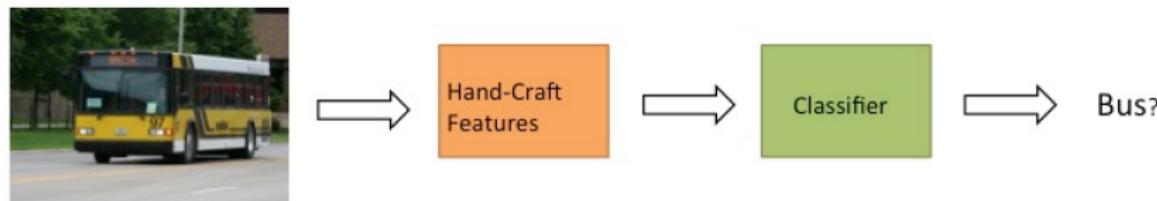
3 Can we make it better?

# Outline

- 1 What is inside?
- 2 How does it work?
- 3 Can we make it better?

# What is deep learning?

- Shallow Learning:



# What is deep learning?

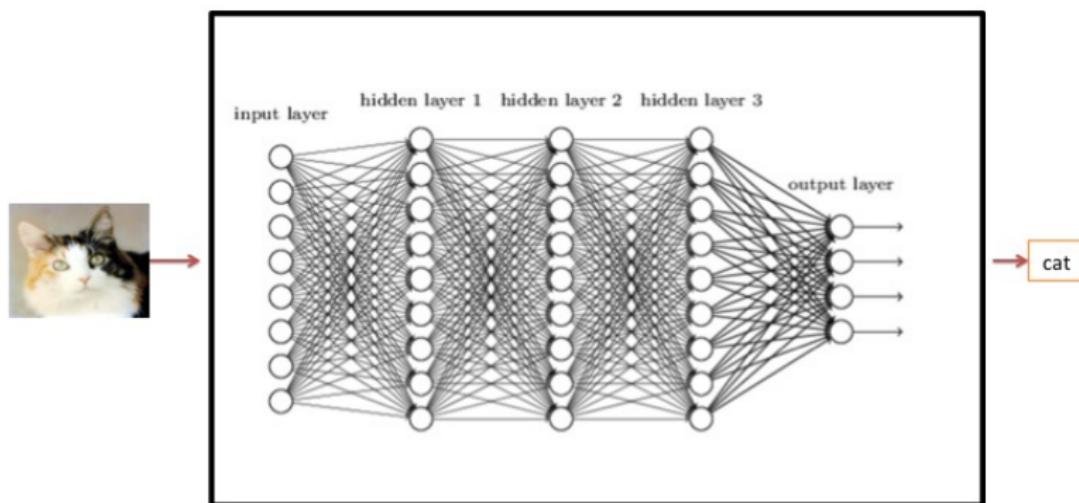
- How our brain learns:



- Billions neurons and connections
- Electrical signal transmitted through axon
- Activity of a neuron depends on inputs ("jennifer neurons")

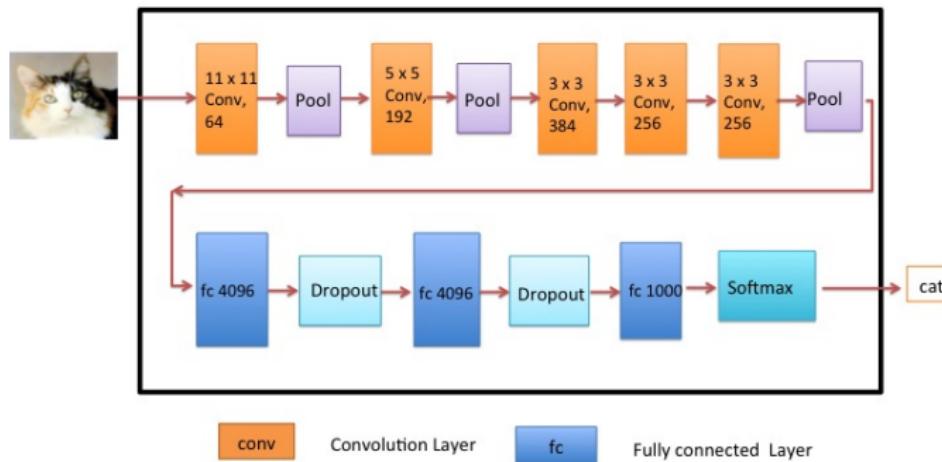
# What is deep learning?

- Deep Learning (Inspiration from human brain):



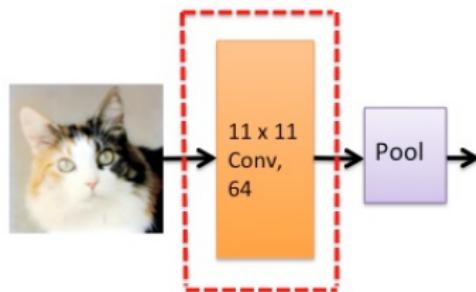
# What is inside?

The classical example: AlexNet [A Krizhevsky, et .al, 2012]



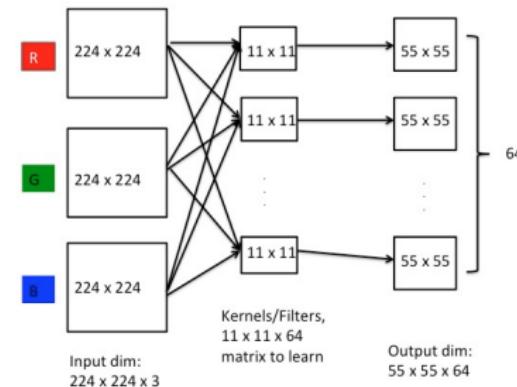
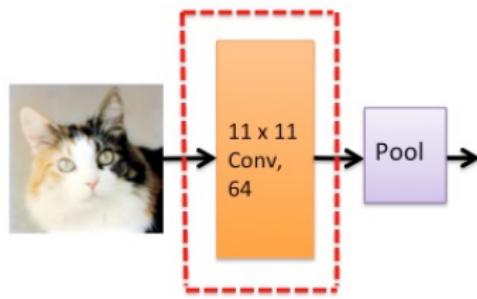
# What is inside?

Convolution Layer [Y Lecun, et .al, 1998]



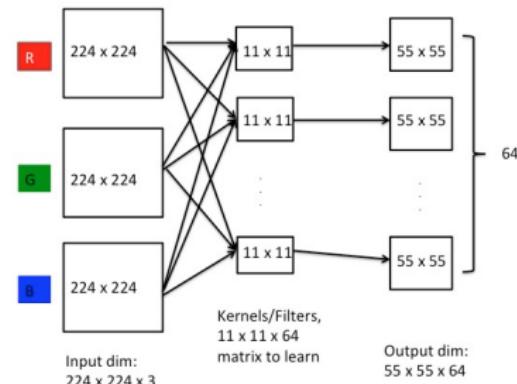
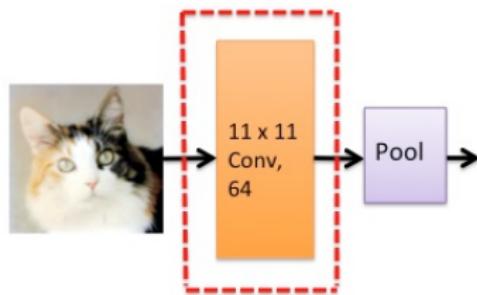
# What is inside?

Convolution Layer [Y Lecun, et .al, 1998]



# What is inside?

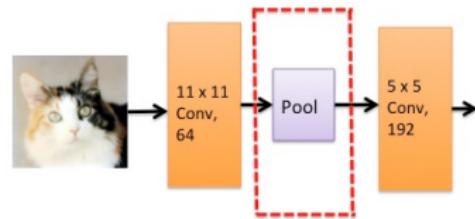
Convolution Layer [Y Lecun, et .al, 1998]



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

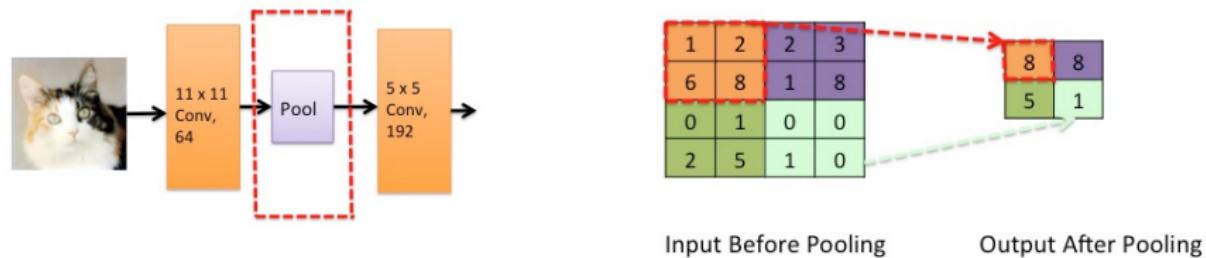
# What is inside?

## Pooling Layer



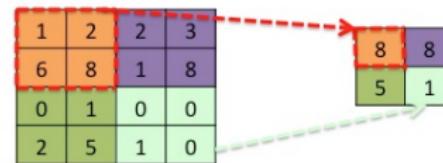
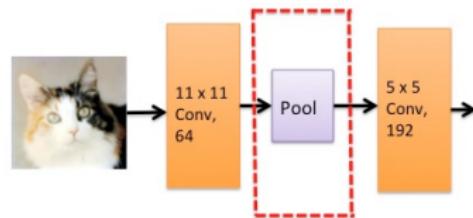
# What is inside?

## Pooling Layer



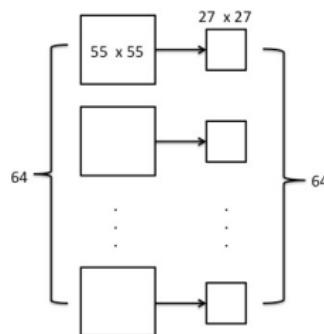
# What is inside?

## Pooling Layer



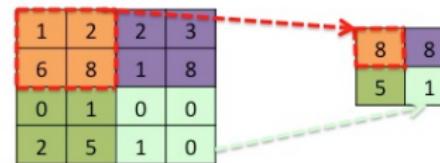
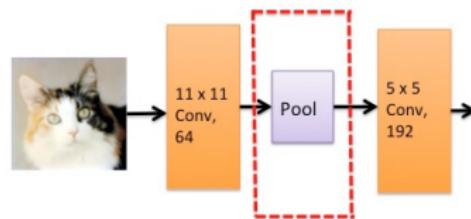
Input Before Pooling

Output After Pooling



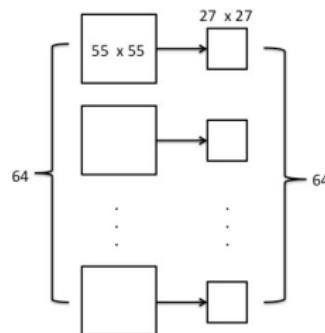
# What is inside?

## Pooling Layer



Input Before Pooling

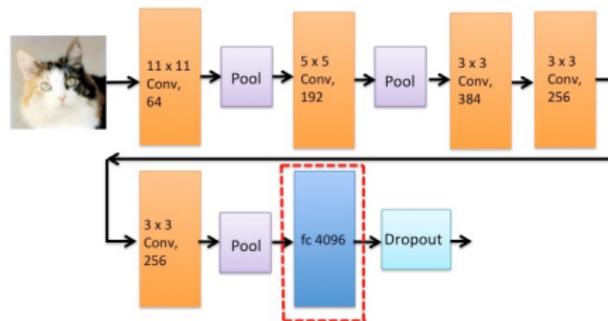
Output After Pooling



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

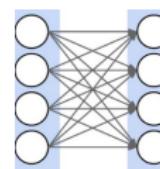
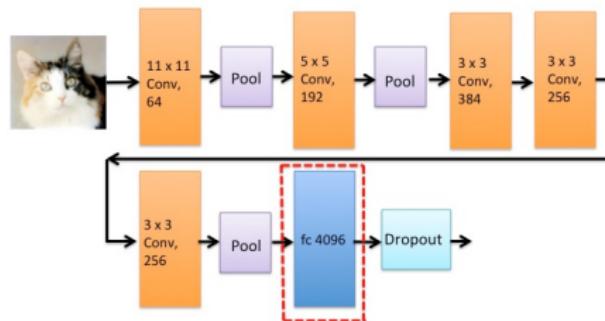
# What is inside?

## Fully Connected Layer



# What is inside?

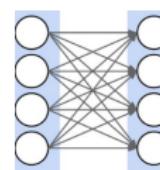
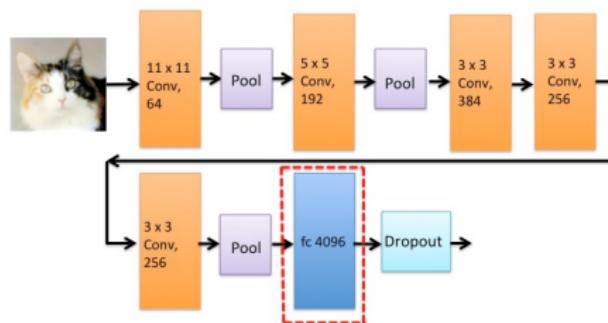
## Fully Connected Layer



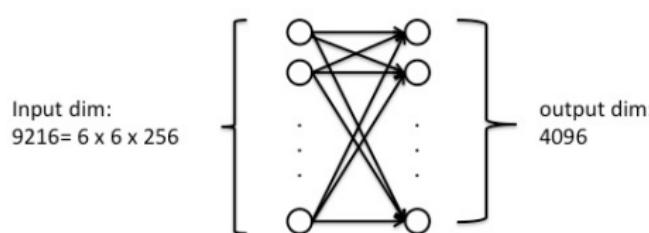
- 4 × 4 weight matrix

# What is inside?

## Fully Connected Layer

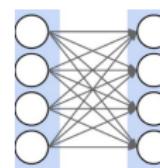
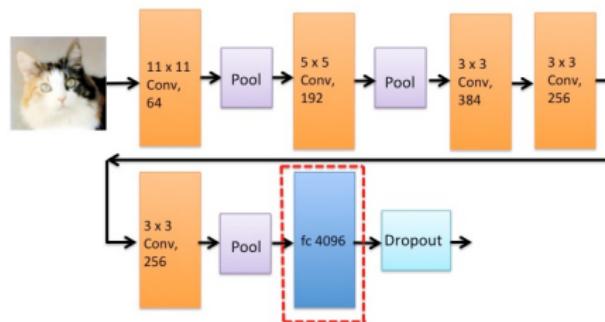


• 4 × 4 weight matrix



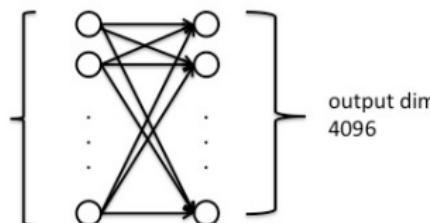
# What is inside?

## Fully Connected Layer



•  $4 \times 4$  weight matrix

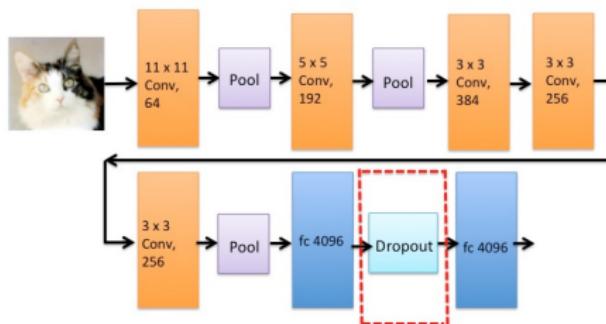
Input dim:  
 $9216 = 6 \times 6 \times 256$



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

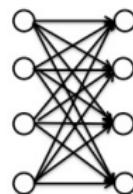
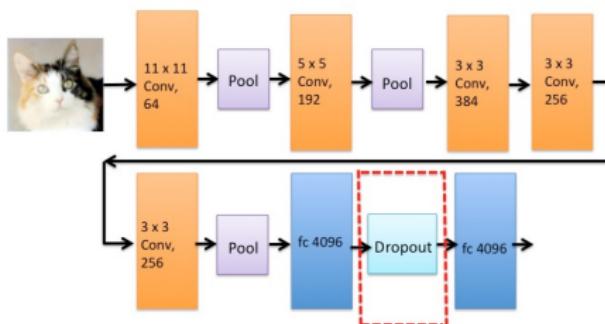
# What is inside?

Dropout Layer[G E Hinton et al, 2012]

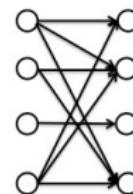


# What is inside?

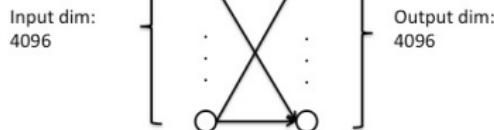
## Dropout Layer[G E Hinton et al, 2012]



Before applying  
dropout



After applying  
dropout



input dim	4096
output dim	4096
num of parameters	0

# What is inside?

Summary of this magic box:

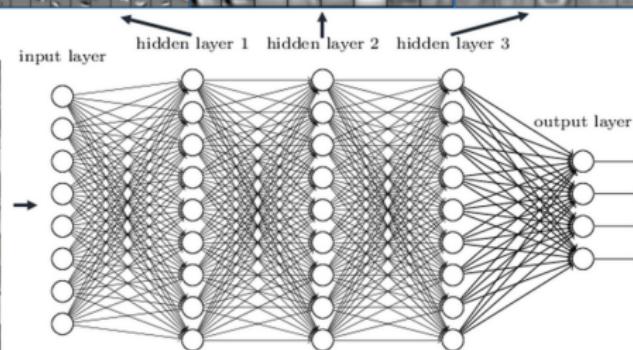
- 8 layers with weights to learn
  - 5 convolution layers
  - 3 fully connected layers
- 3 pooling layers, 2 dropout layers
- Num of parameters: 58.7 millions

# Outline

- 1 What is inside?
- 2 How does it work?
- 3 Can we make it better?

# How does it work?

Deep neural networks learn hierarchical feature representations



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

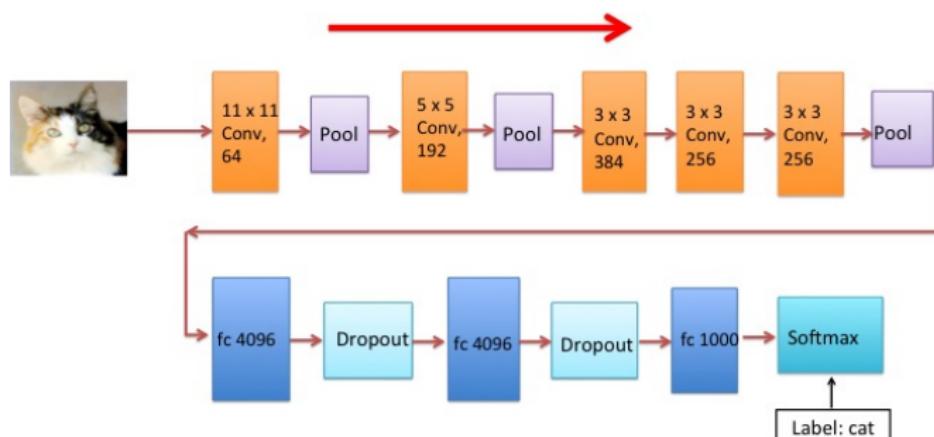
# How does it work?

Set up parameters (Optimization Algorithm)

- Initialize the parameter  $w$  in neural network
- Forward to compute the loss
- Back Propagation to update the parameter  $w$

# How does it work?

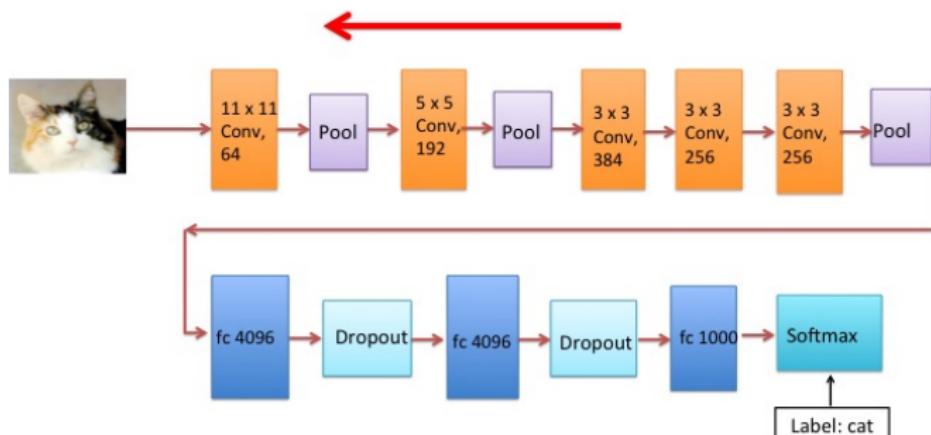
Forward to compute the loss



# How does it work?

Back Propagation to update the parameter  $w$

- chain rule for computing gradient



# How does it work?

## Implementation - Software Level

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

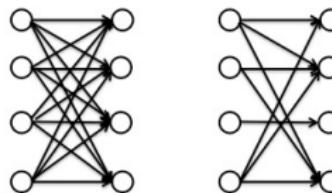


# Outline

- 1 What is inside?
- 2 How does it work?
- 3 Can we make it better?

# Can we make it better?

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



Before applying dropout

After applying dropout

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

# Can we make it better?

## Improved Dropout

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

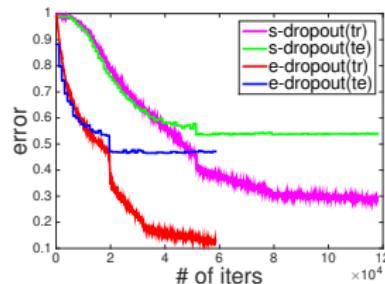


Figure: Evolutional dropout vs standard dropout on CIFAR100 datasets for deep learning

# Q&A

If you have

- **a couple** questions, you may ask now!
- **several** questions, we may talk after this seminar!
- **tons of** questions,

# Thank You!