Deep Learning
Models, Algorithms, and Applications in Computer Vision

Boqing Gong
May 17th, 2016

A tutorial on Research for Under Graduates 2016
at the Center for Research in Computer Vision, University of Central Florida
Deep Learning in The Press ...

• BEGIN: Slides by Dr. Li Deng
Deep Learning

With massive amounts of computational power, machines can now recognize objects and translate speech in real time. Artificial intelligence is finally getting smart.

Temporary Social Media

Messages that quickly self-destruct could enhance the privacy of online communications and make people freer to be spontaneous.

Prenatal DNA Sequencing

Reading the DNA of fetuses will be the next frontier of the genomic revolution. But do you really want to know about the genetic problems or musical aptitude of your unborn child?

Additive Manufacturing

Skeptical about 3-D printing? GE, the world's largest manufacturer, is on the verge of using the technology to make jet parts.

Baxter: The Blue-Collar Robot

Rodney Brooks' latest creation is easy to interact with, but the complex innovations behind the robot show just how hard it is to get along with people.

Memory Implants

A maverick neuroscientist believes he has deciphered the code by which the brain forms long-term memories. Next: testing a prosthetic implant for people suffering from long-term memory loss.

Smart Watches

The designers of the Pebble watch realized that a mobile phone is more useful if you don't have to take it out of your pocket.

Ultra-Efficient Solar Power

Doubling the efficiency of a solar cell would completely change the economics of renewable energy. Nanotechnology just might make it possible.

Big Data from Cheap Phones

Collecting and analyzing information from simple cell phones can provide surprising insights into how people move and behave – and even help us understand the spread of diseases.

Supergrids

A new high-power circuit breaker could finally make highly efficient DC power grids practical.
Scientists See Promise in Deep-Learning Programs
John Markoff
November 23, 2012

Rich Rashid in Tianjin, October, 25, 2012
Enabling Cross-Lingual Conversations in Real Time

The success of the team’s progress to date was on display May 27, in a talk by Microsoft CEO Satya Nadella in Rancho Palos Verdes, Calif., during the Code Conference. During Nadella’s conversation with Kara Swisher and Walt Mossberg of the Re/code tech website relating to a new

The path to the Skype Translator gained
even more momentum with an
encounter in the
autumn of 2010. Seide and colleague Kiran
Thamban had developed a system
they called The
Translator Telephone
for live speech-to-text
and text-to-speech
trans

calls.

Li Deng (left) and Geoff Hinton.

A core development that enables Skype translation came from Redmond researcher Li Deng. He invited Geoff Hinton, a professor at the University of Toronto, to visit Redmond in 2009 to work on new neural-network learning methods, based on a couple of seminal papers from Hinton and his collaborators in 2006 that had brought new
Bing helps you find better images in your searches with ‘deep learning’

22 Nov 2013 9:21 AM
Impact of deep learning in speech technology
Facebook’s foray into deep learning sees it following its competitors Google and Microsoft, which have used the approach to impressive effect in the past year. Google has hired and acquired leading talent in the field (see “10 Breakthrough Technologies 2013: Deep Learning”), and last year created software that taught itself to recognize cats and other objects by reviewing stills from YouTube videos. The underlying deep learning technology was later used to slash the error rate of Google’s voice recognition services (see “Google’s Virtual Brain Goes to Work”). Researchers at Microsoft have used deep learning to build a system that translates speech from English to Mandarin Chinese in real time (see “Microsoft Brings Star Trek’s Voice Translator to Life”). Chinese Web giant Baidu also recently established a Silicon Valley research lab to work on deep learning.
Is Google Cornering the Market on Deep Learning?

A cutting-edge corner of science is being wooed by Silicon Valley, to the dismay of some academics.

By Antonio Regalado on January 29, 2014

How much are a dozen deep-learning researchers worth? Apparently, more than $400 million.

This week, Google reportedly paid that much to acquire DeepMind Technologies, a startup and in
intelligence projects. “DeepMind is bona fide in terms of its research capabilities and depth,” says Peter Lee, who heads Microsoft Research.

According to Lee, Microsoft, Facebook (FB), and Google find themselves in a battle for deep learning talent. Microsoft has gone from four full-time deep learning experts to 70 in the past three years. “We would have more if the talent was there to be had,” he says. “Last year, the cost of a top, world-class deep learning expert was about the same as a top NFL quarterback prospect. The cost of that talent is pretty remarkable.”
Deep Learning’s Role in the Age of Robots

BY JULIAN GREEN, JETPAC 05.02.14  2:56 PM

Image: JetEdoe/Flickr

Can robots see as well as humans? That’s a question the biggest companies around are trying to answer.
Chinese Search Giant Baidu Hires Man Behind the “Google Brain”

Leading AI researcher Andrew Ng, previously associated with Google, will lead a new effort by China’s Baidu to create software that understands the world.

By Tom Simonite on May 16, 2014

Baidu has long been referred to as “China’s Google” because it dominates Web search in the country. Today the comparison grew more apt: Baidu has opened a new artificial-intelligence research lab in Silicon Valley that will be overseen by
DARPA is working on its own deep-learning project for natural-language processing

by Derrick Harris  MAY. 2, 2014 - 10:49 AM PDT

SUMMARY: The Defense Advanced Research Projects Agency, or DARPA, is building a set of technologies to help it better understand human language so it can analyze speech and text sources and alert analysts of potentially useful information.
A startup called Skymind launches, pushing open source deep learning

by Derrick Harris  JUN. 2, 2014 - 10:03 AM PDT

SUMMARY: Skymind is providing commercial support and services for an open source project called deeplearning4j. It's a collection of approaches to deep learning that mimic those developed by leading researchers, but tuned for enterprise adoption.
DEEP LEARNING

» Computers learning and growing on their own

» Able to understand complex, massive amounts of data

DATA ECONOMY

DEEP LEARNING

Brought to you by:

CNBC

Is Deep Learning, the 'holy grail' of big data? - CNBC - Video
video.cnbc.com/gallery/?video=3000192292
Aug 22, 2013
Derrick Harris, GigaOM, explains how "Deep Learning" computers are able to process and understand ...
Deep Learning in The Press ...

• END: Slides by Dr. Li Deng
Deep Learning in Academia

• Deep neural networks
• Deep generative models
• Learning & Inference
• Optimization & Regularization
• Applications (in computer vision, speech, robotics, graphics, natural language process, etc.)
• Tools
• Tricks
Object Detection

Image credits: sub_o@StackOverFlow, Sheng et al., and Divvala et al.
Deep Learning in Vision

Object detection performance, PASCAL VOC 2010

- DPM (2010): 33.4
- segDPM (2014): 40.4
- RCNN (2014): 53.7
- RCNN* (Oct 2014): 62.9
- segRCNN (Jan 2015): 67.2
- Fast RCNN (Jun 2016): 70.8
Object Recognition

Correct label: Correct predictions

Correct label: Incorrect predictions
ImageNet 1K Competition (Fall 2012)

![Bar chart showing performance in the ImageNet 1K Competition. The bars are for LEAR-XRCE, U. of Amsterdam, XRCE/INRIA, Oxford, ISI, and Deep CNN (University of Toronto team).](chart_image)

- LEAR-XRCE: Error rate
- U. of Amsterdam: Error rate
- XRCE/INRIA: Error rate
- Oxford: Error rate
- ISI: Error rate
- Deep CNN (University of Toronto team): Error rate
ImageNet 1K Competition (Fall 2013)
After no improvement for 10+ years by the research community...

...MSR reduced error from ~23% to <15% (and under 7% for Rick’s demo)!

Li Deng (MSRR), Dong Yu (MSRR), & Geoffrey Hinton (Utoronto);
Frank Seide (MSRA)
Data and machine learning

Performance

Amount of data

New AI methods (deep learning)

Most learning algorithms

Andrew Ng
The Big Data Era!

- Facebook: 140 billion images, 12M new images/hr
- Flickr: 6 billion images, 1.8M added daily
- Instagram: 30 billion images, 70M added daily
- YouTube: 300 hours of video uploaded per minute
- World: 3.8 trillion photos, 10% taken in 2013

80~90% of total IP traffic will be visual! — Cisco

Slide credit: Alexei Efros
In the remaining 1.5 hours ...  

- Deep neural networks  
- Deep generative models  
- Learning & Inference  
- Optimization & Regularization  
- Applications (in computer vision, speech, robotics, graphics, natural language process, etc.)  
- Tools  
- Tricks
Neuron: The Building Block
Biological neurons

- Human brains has about 10 billion neurons
- Each connected to 10K other neurons
- A neuron fires if the sum of electrochemical inputs exceeds some threshold

*Image credit: cs.stanford.edu/people/eroberts*
Artificial neurons --- perceptrons

- Introduced by Rosenblatt in 1958
- The **basic building block** for (not all) neural networks

\[
y = \varphi \left( \sum_{i=1}^{n} w_i x_i + b \right) = \varphi (w^T x + b)
\]

\(\varphi(\cdot)\): activation function

Image credit: www.hiit.fi/u/ahonkela/dippa/node41.html
Popular activation functions

\[ \varphi(\cdot) : \text{activation function} \]

\[ \varphi(x) = \begin{cases} 
0 & \text{if } x < 0 \\
1 & \text{if } x \geq 0 
\end{cases} \]

\[ \varphi(x) = \frac{1}{1 + \exp(-x)} \]

\[ \varphi(x) = \tanh(x) = \frac{\exp(x) - \exp(-x)}{\exp(x) + \exp(-x)} \]

\[ \varphi(x) = \begin{cases} 
0 & \text{if } x < 0 \\
x & \text{if } x \geq 0 
\end{cases} \]
Artificial neurons --- perceptrons

- AND
- OR
- NOT
- XOR

- Support Vector Machines
- Logistic regression
- Linear regression

Image credit: www.hiit.fi/u/ahonkela/dippa/node41.html
Let’s build neural networks!
Constructing neural networks from neurons

- Human brains has about 10 billion neurons
- Each connected to 10K other neurons
- A neuron fires if the sum of electrochemical inputs exceeds some threshold
Basic network structures

- **Feed-forward** networks
- **Recurrent** neural networks

*Image credit: http://mesin-belajar.blogspot.com/2016/01/a-brief-history-of-neural-nets-and-deep_84.html*
A Simple Neural Network

Also,
Neuron (offset “+1” omitted)
Support Vector Machine (linear $h$)
Logistic regression (logistic $h$)
Linear regression
Principal component analysis
Etc.
Neural Network
Neural Network

Elman Neural Network

Jordan Neural Network

Hopfield Neural Network
Convolutional Neural Network

ConvNet diagram from Torch Tutorial
Convolutional Neural Networks (CNN)
CNN: Building blocks

$\tau$

$w_{k,l}$

$z_{i,j}$

$a_{i,j} = \sum_{k,l} w_{k,l} z_{i-k,j-l}$

only parameters

$y_{i,j} = f(a_{i,j})$

\[ f(a) = [a]_+ \]

\[ f(a) = \text{sigmoid}(a) \]

$\tau$

$|k| < \tau, |l| < \tau$

$\max_{k,l} y_{i-k,j-l}$

pooling stage

mean or subsample also used

By Dr. Richard E. Turner
Convolution

Center element of the kernel is placed over the source pixel. The source pixel is then replaced with a weighted sum of itself and nearby pixels.

Source pixel

Convolution kernel (emboss)

New pixel value (destination pixel)

Figure from VImage
Convolution

32x32x3 image

- Height: 32
- Width: 32
- Depth: 3
Convolution

32x32x3 image

5x5x3 filter

Convolve the filter with the image i.e. “slide over the image spatially, computing dot products”
Convolution

32x32x3 image
5x5x3 filter $w$

1 number:
the result of taking a dot product between the filter and a small 5x5x3 chunk of the image (i.e. $5*5*3 = 75$-dimensional dot product + bias)

$$w^T x + b$$

Fei-Fei Li & Andrej Karpathy & Justin Johnson
Convolution

32x32x3 image
5x5x3 filter

convolve (slide) over all spatial locations

activation map
consider a second, green filter

Convolution

32x32x3 image
5x5x3 filter

convolve (slide) over all spatial locations

activation maps
Convolutional Neural Network

ConvNet diagram from Torch Tutorial
CNN: Building blocks

\[ x_{i,j} = \max_{|k|<\tau,|l|<\tau} y_{i-k,j-l} \]

Mean or subsample also used

\[ y_{i,j} = f(a_{i,j}) \]

e.g. \( f(a) = [a]_+ \)

\[ f(a) = \text{sigmoid}(a) \]

\[ a_{i,j} = \sum_{k,l} w_{k,l} z_{i-k,j-l} \]

only parameters

By Dr. Richard E. Turner
Convolutional Neural Network

ConvNet diagram from Torch Tutorial
CNN: Building blocks

\[ x_{i,j} = \max_{|k|<\tau,|l|<\tau} y_{i-k,j-l} \]

mean or subsample also used

pooling stage

\[ y_{i,j} = f(a_{i,j}) \]

non-linear stage

e.g. \[ f(a) = [a]^+ \]

\[ f(a) = \text{sigmoid}(a) \]

\[ a_{i,j} = \sum_{k,l} w_{k,l} z_{i-k,j-l} \]

only parameters

convolutional stage

input image

By Dr. Richard E. Turner
Pooling

A pooling operator operates on individual feature channels, coalescing nearby feature values into one by the application of a suitable operator.

Common choices include max-pooling (using the max operator) or sum-pooling (using summation).

Max-pooling is defined as:

\[
y_{i',j'} = \max_{i,j \in \Omega(i',j')} x_{ij}
\]
Pooling

224x224x64 -> pool -> 112x112x64

downsampling

Single depth slice:

<table>
<thead>
<tr>
<th>1</th>
<th>1</th>
<th>2</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
</tbody>
</table>

max pool with 2x2 filters and stride 2

<table>
<thead>
<tr>
<th>6</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>4</td>
</tr>
</tbody>
</table>
Convolutional Neural Network

ConvNet diagram from Torch Tutorial
Why pooling layer?

• ??
Why pooling layer?

- Reducing sizes of feature maps → More manageable
- Handling (small) translation
- Handling (small) scaling and (small) rotation
Another convolution layer

ConvNet diagram from Torch Tutorial
Convolutional Neural Networks
History & State of the art
A bit of history:

**Hubel & Wiesel,**  
1959
RECEPTIVE FIELDS OF SINGLE NEURONES IN THE CAT'S STRIATE CORTEX

1962
RECEPTIVE FIELDS, BINOCULAR INTERACTION AND FUNCTIONAL ARCHITECTURE IN THE CAT'S VISUAL CORTEX

1968...

Fei-Fei Li & Andrej Karpathy & Justin Johnson
Why use hierarchical multi-layered models?

Argument 1: visual scenes are hierarchically organised

```
object
  └── object parts
    └── primitive features
        └── input image

trees
  └── bark, leaves, etc.
    └── oriented edges
        └── forest image
```

By Dr. Richard E. Turner
Why use hierarchical multi-layered models?

Argument 2: biological vision is hierarchically organised

- objects
  - object parts
    - primitive features
      - input image
  - trees
    - bark, leaves, etc.
    - oriented edges
      - forest image
    - V4: different textures
      - V1: simple and complex cells
      - Inferotemporal cortex
    - photo-receptors retina
Why use hierarchical multi-layered models?

Argument 3: shallow architectures are inefficient at representing deep functions

single layer neural network implements: $x = f_\theta(z)$

shallow networks can be computationally inefficient

networks we met last lecture with large enough single hidden layer can implement any function 'universal approximator'

however, if the function is 'deep' a very large hidden layer may be required
<table>
<thead>
<tr>
<th>Layer 1</th>
<th>Layer 2</th>
</tr>
</thead>
</table>

Justifying the arguments (?)
2012: The AlexNet
15 million labeled images with 22,000 categories

Labeled by humans using…

Amazon’s Mechanical Turk crowdsourcing tool
pay people to manually label large datasets
ImageNet Large-Scale Visual Recognition Challenge (ILSVRC)

Subset of the full ImageNet dataset

1.2 million training images
50,000 validation, 150,000 test images
1000 different classes!

~1000 images in each category (not an unreasonable number for MIA)
amazing **Results** to motivate why we learn about this approach

On test (unseen) data:
- top-1 error = 37.5% (prev best was 45.7%)
- top-5 error = 15.3% (second-best 26.2%)
Results

Correct label

Correct predictions

Incorrect? Predictions
ImageNet 1K Competition (Fall 2012)

Error

LEAR-XRCE
U. of Amsterdam
XRCE/INRIA
Oxford
ISI
SuperVision

Deep CNN
Univ. Toronto team
ImageNet 1K Competition (Fall 2013)
2013: VGGNet
2014: GoogLeNet
2015: ResNet
Questions thus far?
Question: how to set the weights for each neuron?!
Milestones: In the remaining 1.5 hours ...

- Deep neural networks
- Deep generative models
- *Learning* & Inference
- *Optimization* & Regularization
- Applications (*in* computer vision, speech, robotics, graphics, natural language process, etc.)
- Tools
- Tricks
Learning weights (model parameters) (1)

• Is equivalent to

\[ \Theta \]  

Empirical Risk Minimization (ERM)

Choose one hypothesis \( h \in \mathcal{H} \) to approximate concept \( c \)

• where,

Binary classification concept: \( c : \mathcal{X} \to \mathcal{Y} = \{0, 1\} \)

Hypotheses \( \mathcal{H} = \{\text{NET}(\Theta) | \Theta_d \in \mathbb{R}\} \)

• Questions: \( c \) is unknown

\( c \in \mathcal{H} \)?
Learning the model parameters $\Theta$ (2)

• Is equivalent to

Choose one hypothesis $h \in \mathcal{H}$ to approximate concept $c$

• Can be implemented by

$$\Theta^* \leftarrow \arg \min_{\Theta} \quad R(\Theta) \quad \leftarrow \text{Called the generalization risk}$$

$$R(\Theta) = Pr(\text{NET}(x; \Theta) \neq y) = \mathbb{E}_{(x,y) \sim P_{XY}} [\text{NET}(x; \Theta) \neq y]$$

$P_{XY}$ is the underlying distribution of $(x,y)$
Learning the model parameters $\Theta$ (3)

- Is equivalent to

Choose one hypothesis $h \in \mathcal{H}$ to approximate concept $c$

- by,

$$
\Theta^* \leftarrow \arg\min_{\Theta} \mathbb{E}_{(x,y) \sim P_{XY}} [\text{NET}(x; \Theta) \neq y]
$$
Unknown underlying distribution $P_{XY}$

$$\Theta^* \leftarrow \arg \min_{\Theta} \mathbb{E}_{(x,y) \sim P_{XY}} [\text{NET}(x; \Theta) \neq y]$$
Unknown underlying distribution $P_{XY}$
→ Empirical risk minimization

$$
\Theta^* \leftarrow \arg\min_{\Theta} \quad \mathbb{E}_{(x,y) \sim P_{XY}}[\text{NET}(x; \Theta) \neq y]
$$

$$
\hat{\Theta} \leftarrow \arg\min_{\Theta} \quad \frac{1}{n} \sum_{i=1}^{n} [\text{NET}(x_i; \Theta) \neq y_i]
$$

$\hat{\Theta} \rightarrow \Theta^*$ given many training data $(x_i, y_i), i = 1, 2, \cdots, n$
Optimization ... and we are done!

\[ \hat{\Theta} \leftarrow \arg \min_{\Theta} \frac{1}{n} \sum_{i=1}^{n} \left[ \text{NET}(x_i; \Theta) \neq y_i \right] \]
Optimization ... and we are done?

- Challenge: non-differentiable loss function

\[
\hat{\Theta} \leftarrow \arg \min_{\Theta} \frac{1}{n} \sum_{i=1}^{n} [\text{NET}(x_i; \Theta) \neq y_i]
\]

\[\text{0-1 Loss function}\]
Optimization ... with a new loss function

- Squired loss is differentiable everywhere
- It is cheap to optimize using gradient descent

\[ \hat{\Theta} \leftarrow \arg \min_{\Theta} \frac{1}{n} \sum_{i=1}^{n} (y_i - \text{NET}(x_i; \Theta))^2 \]

Squared Loss

- Other loss functions
  - Hinge loss, Cross-entropy loss, Exponential Loss, RankNet loss, etc.
SGD: Stochastic gradient descent

\[ \hat{\Theta} \leftarrow \arg \min_{\Theta} \frac{1}{n} \sum_{i=1}^{n} (y_i - \text{NET}(x_i; \Theta))^2 \]

Squared Loss

\[ \frac{\partial \text{Loss}}{\partial \Theta} \]

Backpropagation
Milestones: In the remaining 1.5 hours ...

• **Deep neural networks**
• Deep generative models
• **Learning & Inference**  
  [Optional]
• **Optimization & Regularization**
• Applications (*in computer vision, speech, robotics, graphics, natural language process, etc.*)
• Tools
• Tricks
Next: Recurrent neural networks (RNN)

- Feed-forward networks
- Recurrent neural networks

Why RNN?

- Feed-forward networks
  - Model static input-output concept
    - No time series
  - Exists a single forward direction

- CNN

- Recurrent neural networks
  - Model dynamic state transition
    - Time & sequence data
  - Exists feedback connections

- LSTM
Why RNN? (cont’d)

• Markov models
  • Model dynamic state transition
    • Time & sequence data
  • Markov (short-range) dependency
  • Moderately sized states

• Recurrent neural networks
  • Model dynamic state transition
    • Time & sequence data
  • Long-range dependency
  • Exponentially expressive states
Why RNN? (applications in computer vision)

• Image caption generator

Image credits: Karpathy, Andrej, and Li Fei-Fei
Why RNN? (applications in computer vision)

• Image caption generator
• Sentence embedding

Why RNN? (applications in computer vision)

- Image caption generator
- Sentence embedding
- Word embedding

Image credits: Karpathy, Andrej, and Li Fei-Fei
Why RNN? (applications in computer vision)

• Image caption generator
• Sentence embedding
• Word embedding
• Activity recognition

Why RNN? (applications in computer vision)

- Image caption generator
- Sentence embedding
- Word embedding
- Activity recognition
- Video representation

Srivastava, Nitish, Elman Mansimov, and Ruslan Salakhutdinov.
Why RNN? (applications in computer vision)

- Image caption generator
- Sentence embedding
- Word embedding
- Activity recognition
- Video representation
- Visual Question Answering
- Early action detection
- Human dynamics
- Scene labeling

- Language modeling
- Machine translation
Next: Recurrent neural networks (RNN)

• RNN
• Vanishing and exploding gradients
• Long short-term memory (LSTM)
• Gated Recurrent Unit (GRU)
RNN

• The processing occurs in discrete steps
(Discrete-time) RNN

• At a single time step $t$:
(Discrete-time) RNN

- Three time steps and beyond

Image credits: Richard Socher
(Discrete-time) RNN

- Three time steps and beyond

- In math

\[
x_1, \ldots, x_{t-1}, x_t, x_{t+1}, \ldots, x_T
\]

\[
h_t = \sigma \left( W^{(h h)} h_{t-1} + W^{(h x)} x[t] \right)
\]

\[
\hat{y}_t = \text{softmax} \left( W^{(S)} h_t \right)
\]

Image credits: Richard Socher
(Discrete-time) RNN

- Three time steps and beyond
- A layered feedforward net
- Tied weights for different time steps
- Conditioning (memorizing?) on all previous input
- Memory being cheap to save in RAM

Image credits: Richard Socher
(Discrete-time) RNN

• Compare with holistically nested network

Difference from feedforward network:
- Tied weights for RCNN
- #layers not fixed for RCNN
- Input at every layer for RCNN
- Less RAM for RCNN

Image credits: Richard Socher
(Discrete-time) RNN

- Three time steps and beyond
- Expressive in modeling sequences
- Training by backpropagation
  - Unstable
  - Vanishing & exploding gradients
  - Troublesome in learning long-term dependencies
- Training by other methods?
  - Alternatives exist
  - Hard to use

Image credits: Richard Socher
Next: Recurrent neural networks (RNN)

• RNN
• Vanishing and exploding gradients
• RNN advanced: Long short-term memory (LSTM)
• RNN advanced: Gated Recurrent Unit (GRU)
LSTM (Long Short-Term Memory)

- RNN

- LSTM

Overwrite the hidden states
→ multiplicative gradients

Add to the cell states
→ additive gradients

Image credits: http://colah.github.io/posts/2015-08-Understanding-LSTMs/
LSTM in a nutshell

An LSTM contains:
- Forget gate
- Additive operations $\rightarrow$ additive gradients
- Input gate
- Output gate
- Memory cell

It does not contain:
- Input $x$
- Hidden states
- Output $y$

\[
\begin{align*}
    i_t &= \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i) \\
    f_t &= \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f) \\
    c_t &= f_t c_{t-1} + i_t \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \\
    o_t &= \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o) \\
    h_t &= o_t \tanh(c_t)
\end{align*}
\]
LSTM vs. GRU (gated recurrent unit)
Remarks

- Plain RNN
  - Powerful in modeling sequences
  - Hard to capture long dependencies: vanishing & exploding gradients

- LSTMs & GRUs
  - Able to capture long dependencies
  - New memory cells: additive updates weighted by gates
  - Multiplicative gradients \( \rightarrow \) Additive gradients

- Applications in vision: video classification, image captioning, VQA, etc.
- Ignore math, and gain intuitive understanding
Stacked RNNs
Often better performance

Stacked LSTMs / GRUs
Bidirectional RNN → Often better performance
RNN is often used as Encoder and/or Decoder

Kyunghyun Cho et al. 2014
Milestones: In the remaining 1.5 hours ...

- Deep neural networks: ??
- Deep generative models: not covered
- **Learning** & Inference: ??
- **Optimization** & Regularization: ??
- Applications (*in computer vision, speech, robotics, graphics, natural language process, etc.*)
- Tools: ??
- Tricks
Popular deep learning tools

- Pylearn2
- Theano (+ Keras)
- Caffe
- Torch
- Cuda-convnet
- Deeplearning4j
- TensorFlow
- CNTK
- Chainer
Milestones: In the remaining 1.5 hours ...

• Deep neural networks: ??
• Deep generative models: not covered
• **Learning** & Inference: ??
• **Optimization** & Regularization: ??
• Applications (*in computer vision, speech, robotics, graphics, natural language process, etc.*)
• Tools: ??
• **Tricks**
A few tips/tricks
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

- Non-saturating nonlinearity (RELU)
  \[ f(x) = \max(0, x) \]
  Quick to train
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.
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

Data Augmentation

- Change the intensity of RGB channels

\[
I_{xy} = [I^R_{xy}, I^G_{xy}, I^B_{xy}]^T
\]

- add multiples of principle components

\[
[p_1, p_2, p_3][\alpha_1 \lambda_1, \alpha_2 \lambda_2, \alpha_3 \lambda_3]^T
\]

\[
\alpha_i \sim N(0, 0.1)
\]

Reduces the top-1 error rate by over 1%
Dropout

- Set to 0 the output of a neuron with 0.5 probability
- Reduces complex co-adaptation and forces to learn more robust features
- Done in last two layers
Reducing Overfitting

Dropout

- With probability 0.5
- last two 4096 fully-connected layers.
Stochastic Gradient Descent Learning

Updating the weights

**Momentum**: Adds a fraction of the previous weight update to the current one (increases speed of convergence)

\[
v_{i+1} := 0.9 \cdot v_i - 0.0005 \cdot \epsilon \cdot w_i - \epsilon \cdot \left( \frac{\partial L}{\partial w} \right)_{w_i}^{D_i}
\]

**Weight decay**: penalizes large weights (as weight increases, changes less)

\[
w_{i+1} := w_i + v_{i+1}
\]

**Batch size** = how many examples in an iteration \(i\) to train the model on = 128 examples

Average over the \(i\)th batch \(D_i\) of the derivative of the objective with respect to \(w\), evaluated at \(w_i\)
Tech Details - Preprocessing

An input image (256x256) Minus sign The mean input image


• A compiled list of than 100 papers: [http://www.cs.ucf.edu/~bgong/CAP6412/papers.md.html](http://www.cs.ucf.edu/~bgong/CAP6412/papers.md.html)