

Neural Networks

Machine Learning – CSE546

Carlos Guestrin

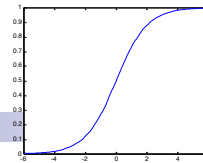
University of Washington

December 2, 2014

©Carlos Guestrin 2005-2014

1

Logistic regression



- $P(Y|X)$ represented by:

$$P(Y = 1 | x, W) = \frac{1}{1 + e^{-(w_0 + \sum_i w_i x_i)}} \\ = g(w_0 + \sum_i w_i x_i)$$

- Learning rule – MLE:

$$\frac{\partial \ell(W)}{\partial w_i} = \sum_j x_i^j [y^j - P(Y^j = 1 | x^j, W)] \\ = \sum_j x_i^j [y^j - g(w_0 + \sum_i w_i x_i^j)]$$

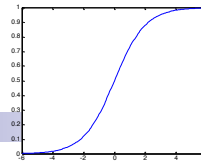
$$w_i \leftarrow w_i + \eta \sum_j x_i^j \delta^j$$

$$\delta^j = y^j - g(w_0 + \sum_i w_i x_i^j)$$

©Carlos Guestrin 2005-2014

2

Perceptron as a graph

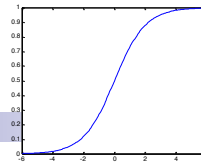


$$g(w_0 + \sum_i w_i x_i) = \frac{1}{1 + e^{-(w_0 + \sum_i w_i x_i)}}$$

©Carlos Guestrin 2005-2014

3

Linear perceptron classification region



$$g(w_0 + \sum_i w_i x_i) = \frac{1}{1 + e^{-(w_0 + \sum_i w_i x_i)}}$$

©Carlos Guestrin 2005-2014

4

Perceptron, linear classification, Boolean functions

- Can learn x_1 AND x_2
- Can learn x_1 OR x_2
- Can learn any conjunction or disjunction

©Carlos Guestrin 2005-2014

5

Perceptron, linear classification, Boolean functions

- Can learn majority
- Can perceptrons do everything?

©Carlos Guestrin 2005-2014

6

Going beyond linear classification

- Solving the XOR problem

©Carlos Guestrin 2005-2014

7

Hidden layer

- Perceptron: $out(\mathbf{x}) = g(w_0 + \sum_i w_i x_i)$

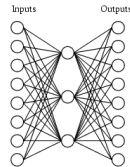
- 1-hidden layer:

$$out(\mathbf{x}) = g\left(w_0 + \sum_k w_k g\left(w_0^k + \sum_i w_i^k x_i\right)\right)$$

©Carlos Guestrin 2005-2014

8

Example data for NN with hidden layer



A target function:

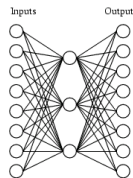
Input	Output
10000000	→ 10000000
01000000	→ 01000000
00100000	→ 00100000
00010000	→ 00010000
00001000	→ 00001000
00000100	→ 00000100
00000010	→ 00000010
00000001	→ 00000001

Can this be learned??

9

Learned weights for hidden layer

A network:



Learned hidden layer representation:

Input	Hidden Values	Output
10000000	→ .89 .04 .08	→ 10000000
01000000	→ .01 .11 .88	→ 01000000
00100000	→ .01 .97 .27	→ 00100000
00010000	→ .99 .97 .71	→ 00010000
00001000	→ .03 .05 .02	→ 00001000
00000100	→ .22 .99 .99	→ 00000100
00000010	→ .80 .01 .98	→ 00000010
00000001	→ .60 .94 .01	→ 00000001

10

NN for images

The diagram shows a neural network with four output nodes labeled 'left', 'strt', 'right', and 'up'. These nodes are connected to a hidden layer of three nodes, which in turn connects to an input layer of 30x32 pixels. Below the network is a grayscale image of a man's face wearing sunglasses. Below that are four smaller grayscale images showing the same man's face from different angles: left profile, front, right profile, and top-down. The text 'Typical input images' is centered below these four images.

90% accurate learning head pose, and recognizing 1-of-20 faces

11

Weights in NN for images

The diagram shows the same neural network structure as slide 11. To the right of the network, under the heading 'Learned Weights', are four grayscale weight filters. Each filter is a 30x32 pixel patch. The first filter is a vertical gradient, the second is a horizontal gradient, the third is a diagonal gradient, and the fourth is a vertical gradient. Below these filters are three grayscale images of the man's face from different angles: left profile, front, and right profile. Below these images are four smaller grayscale images showing the same man's face from different angles: left profile, front, right profile, and top-down. The text 'Typical input images' is centered below these four images.

©Carlos Guestrin 2005-2014

12

Forward propagation for 1-hidden layer - Prediction

- 1-hidden layer:

$$out(\mathbf{x}) = g \left(w_0 + \sum_k w_k g \left(w_0^k + \sum_i w_i^k x_i \right) \right)$$

©Carlos Guestrin 2005-2014

13

Gradient descent for 1-hidden layer – Back-propagation: Computing $\frac{\partial \ell(W)}{\partial w_k}$

$$\ell(W) = \frac{1}{2} \sum_j [y^j - out(\mathbf{x}^j)]^2$$

Dropped w_0 to make derivation simpler

$$out(\mathbf{x}) = g \left(\sum_{k'} w_{k'} g \left(\sum_{i'} w_{i'}^{k'} x_{i'} \right) \right)$$

$$\frac{\partial \ell(W)}{\partial w_k} = \sum_{j=1}^m -[y^j - out(\mathbf{x}^j)] \frac{\partial out(\mathbf{x}^j)}{\partial w_k}$$

©Carlos Guestrin 2005-2014

14

Gradient descent for 1-hidden layer – Back-propagation: Computing $\frac{\partial \ell(W)}{\partial w_i^k}$

$$\ell(W) = \frac{1}{2} \sum_j [y^j - out(\mathbf{x}^j)]^2$$

Dropped w_0 to make derivation simpler

$$out(\mathbf{x}) = g \left(\sum_{k'} w_{k'} g \left(\sum_{i'} w_{i'}^{k'} x_{i'} \right) \right)$$

$$\frac{\partial \ell(W)}{\partial w_i^k} = \sum_{j=1}^m -[y - out(\mathbf{x}^j)] \frac{\partial out(\mathbf{x}^j)}{\partial w_i^k}$$

Multilayer neural networks

Forward propagation – prediction

- Recursive algorithm
- Start from input layer
- Output of node V_k with parents U_1, U_2, \dots :

$$V_k = g\left(\sum_i w_i^k U_i\right)$$

©Carlos Guestrin 2005-2014

17

Back-propagation – learning

- Just stochastic gradient descent!!!
- Recursive algorithm for computing gradient
- For each example
 - Perform forward propagation
 - Start from output layer
 - Compute gradient of node V_k with parents U_1, U_2, \dots
 - Update weight w_i^k

©Carlos Guestrin 2005-2014

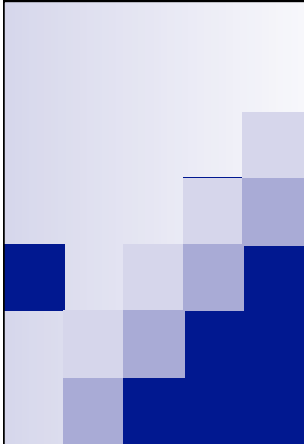
18

Many possible response/link functions

- Sigmoid
- Linear
- Exponential
- Gaussian
- Hinge
- Max
- ...

Poster Session

- Thursday Dec 4, 2:30-4:30pm
 - Please arrive 15mins early to set up
- Everyone is expected to attend
- Prepare a poster
 - We provide poster board (32"x40") and pins
 - Both one large poster and several pinned pages are OK
- Capture
 - Problem you are solving
 - Data you used
 - ML methodology
 - Results
- Prepare a 2-minute speech about your project
- Two instructors will visit your poster separately
- You'll be graded on 3 criteria:
 - Scope: how much stuff you did
 - Technical depth: how challenging it was to do your project (and whether your methodology was correct)
 - Presentation: how you share what you did



Convolutional Neural Networks & Application to Computer Vision

Machine Learning – CSEP546

Carlos Guestrin

University of Washington

December 2, 2014

©Carlos Guestrin 2005-2014

21

Contains slides from...



- LeCun & Ranzato
- Russ Salakhutdinov
- Honglak Lee

©Carlos Guestrin 2005-2014

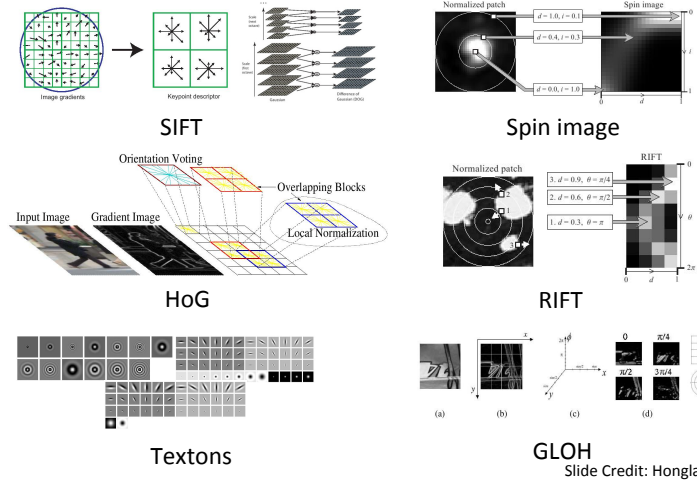
22

Neural Networks in Computer Vision

- Neural nets have made an amazing come back
 - Used to engineer high-level features of images

- Image features:

Some hand-created image features



Scanning an image with a detector

- Detector = Classifier from image patches:

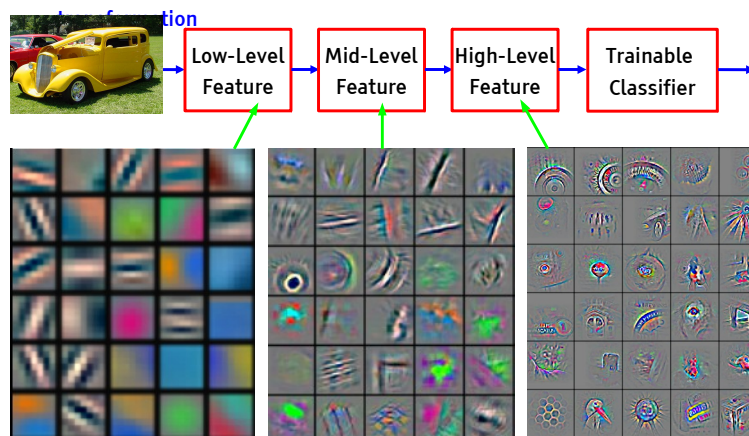
- Typically scan image with detector:



©Carlos Guestrin 2005-2014

25

Using neural nets to learn non-linear features

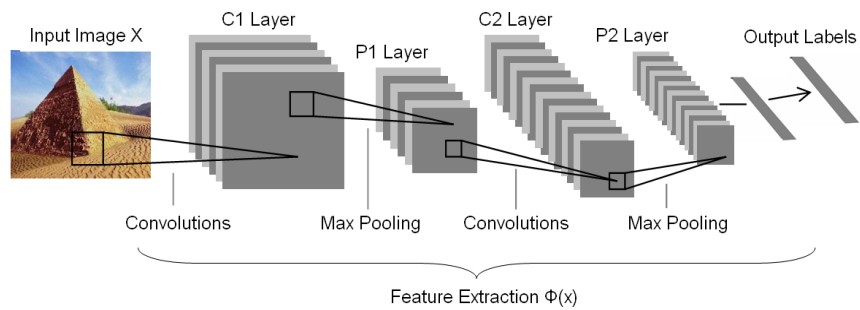


Feature visualization of convolutional net trained on ImageNet from [Zeiler & Fergus 2013]

©Carlos Guestrin 2005-2014

26

But, many tricks needed to work well...



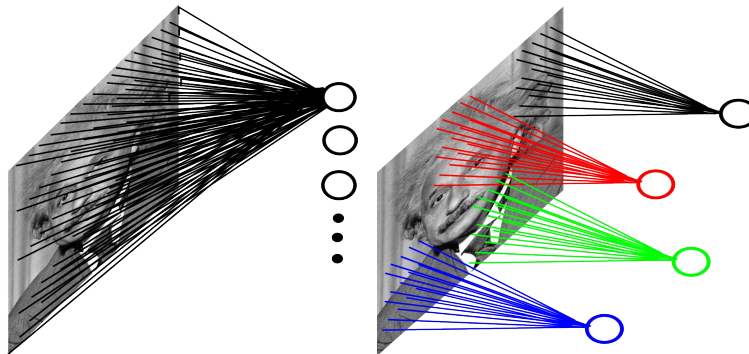
©Carlos Guestrin 2005-2014

27

Convolution Layer

Example: 200x200 image

- ▶ Fully-connected, 400,000 hidden units = 16 billion parameters
- ▶ Locally-connected, 400,000 hidden units 10x10 fields = 40 million params
- ▶ Local connections capture local dependencies



©Carlos Guestrin 2005-2014

28

Parameter sharing

- Fundamental technique used throughout ML
- Neural net without parameter sharing:

- Sharing parameters:

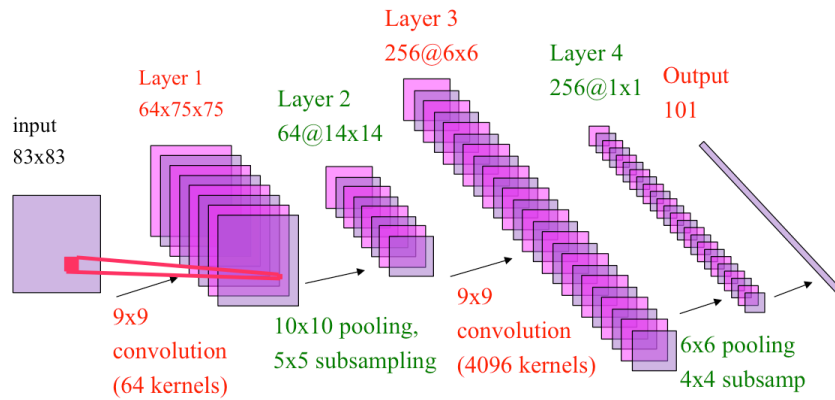
Pooling/Subsampling

- Convolutions act like detectors:



- But we don't expect true detections in every patch
- Pooling/subsampling nodes:

Example neural net architecture



©Carlos Guestrin 2005-2014

31

Sample results

Traffic Sign Recognition (GTSRB)

- ▶ German Traffic Sign Reco Bench
- ▶ 99.2% accuracy



House Number Recognition (Google)

- ▶ Street View House Numbers
- ▶ 94.3 % accuracy



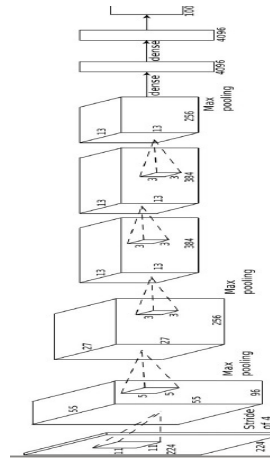
©Carlos Guestrin 2005-2014

32

Example from Krizhevsky, Sutskever, Hinton 2012

■ Won the 2012 ImageNet LSVRC. 60 Million parameters, 832M MAC ops

4M	FULL CONNECT	4Mflop
16M	FULL 4096/ReLU	16M
37M	FULL 4096/ReLU	37M
	MAX POOLING	
442K	CONV 3x3/ReLU 256fm	74M
1.3M	CONV 3x3ReLU 384fm	224M
884K	CONV 3x3/ReLU 384fm	149M
	MAX POOLING 2x2sub	
307K	LOCAL CONTRAST NORM	
	CONV 11x11/ReLU 256fm	223M
	MAX POOL 2x2sub	
	LOCAL CONTRAST NORM	
35K	CONV 11x11/ReLU 96fm	105M

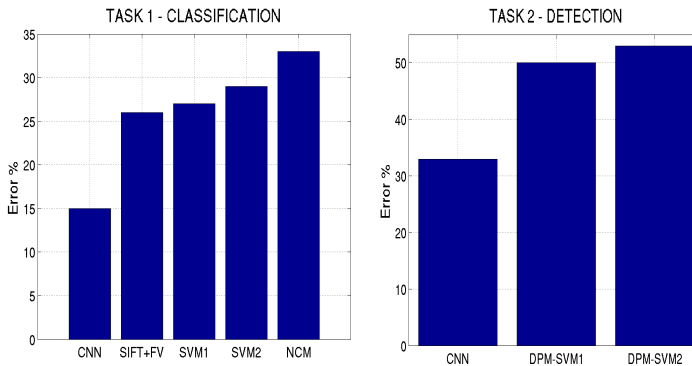


©Carlos Guestrin 2005-2014

33









Results by Krizhevsky, Sutskever, Hinton 2012

- ImageNet Large Scale Visual Recognition Challenge
- 1000 categories, 1.5 Million labeled training samples






































©Carlos Guestrin 2005-2014

34

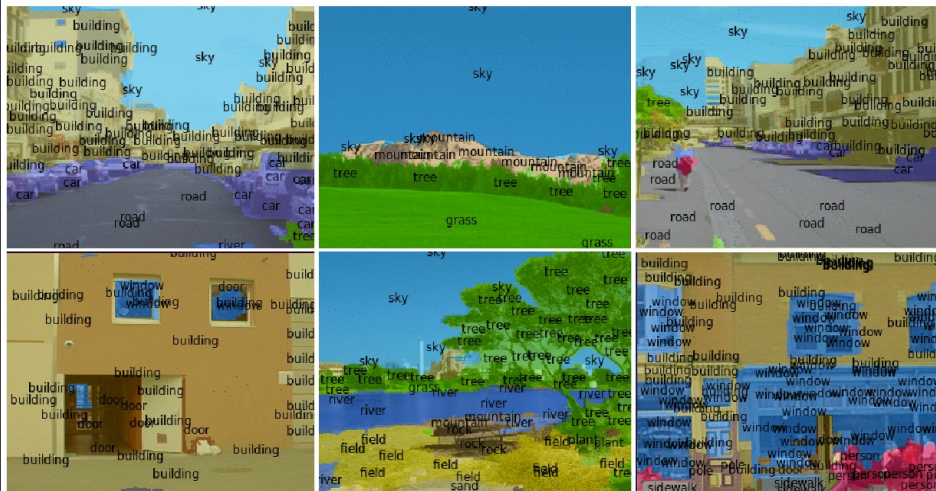
			
mite	container ship	motor scooter	leopard
<ul style="list-style-type: none"> mite black widow cockroach tick starfish 	<ul style="list-style-type: none"> container ship lifeboat amphibian fireboat drilling platform 	<ul style="list-style-type: none"> motor scooter go-kart moped bumper car golfcart 	<ul style="list-style-type: none"> leopard jaguar cheetah snow leopard Egyptian cat
			
grille	mushroom	cherry	Madagascar cat
<ul style="list-style-type: none"> convertible grille pickup beach wagon fire engine 	<ul style="list-style-type: none"> agaric mushroom jelly fungus gill fungus dead-man's-fingers 	<ul style="list-style-type: none"> dalmatian grape elderberry ffordshire bullterrier currant 	<ul style="list-style-type: none"> squirrel monkey spider monkey titi indri howler monkey

©Carlos Guestrin 2005-2014 35

TEST IMAGE	RETRIEVED IMAGES					
						
						
						
						
						

©Carlos Guestrin 2005-2014 36

Application to scene parsing

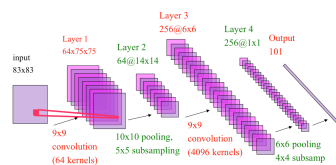


[Farabet et al. ICML 2012, PAMI 2013]

©Carlos Guestrin 2005-2014

37

Learning challenges for neural nets



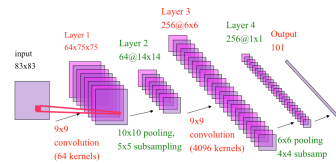
- Choosing architecture
- Slow per iteration and convergence

- Gradient “diffusion” across layers
- Many local optima

©Carlos Guestrin 2005-2014

38

Random dropouts



- Standard dropout:

$$w_i \leftarrow w_i + \eta \sum_j x_i^j \delta^j$$

- Random dropouts: randomly choose edges not to update:

- Functions as a type of “regularization”... helps avoid “diffusion” of gradient

©Carlos Guestrin 2005-2014

39

Revival of neural networks

- Neural networks fell into disfavor in mid 90s -early 2000s
 - Many methods have now been rediscovered ☺
- Exciting new results using modifications to optimization techniques and GPUs
- Challenges still remain:
 - Architecture selection feels like a black art
 - Optimization can be very sensitive to parameters
 - Requires a significant amount of expertise to get good results

©Carlos Guestrin 2005-2014

40