Introduction to Deep Learning for Facial and Gesture Understanding

Part I: Introduction

DEEP LEARNING INSTITUTE
www.nvidia.com/dli

Raymond Ptucha,
Rochester Institute of Technology, USA

Tutorial-2
May 14, 2019, 2-6pm

14th IEEE International Conference on Automatic Face and Gesture Recognition
FG2019

NVIDIA DLI Account

- Congratulations on registering for the F&G'19 Tutoral-2: "Introduction to Deep Learning for Facial and Gesture Understanding".
- Navigate to:
- courses.nvidia.com/dli-event
  - Browser Recommendation: Chrome
- Event code:
  FG19_CV2.0_AMBASSADOR_MAY19
- Create an Account

Fair Use Agreement

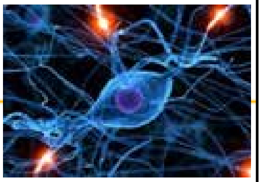
This agreement covers the use of all slides in this document, please read carefully.

- You may freely use these slides, if:
  - You send me an email telling me the conference/venue/company name in advance, and which slides you wish to use.
  - You receive a positive confirmation email back from me.
  - My name (R. Ptucha) appears on each slide you use.

Agenda

- **Part I: Introduction**
- Part II: Convolutional Neural Nets
- Part III: Fully Convolutional Nets
- Break
- Part IV: Facial Understanding
- Part V: Recurrent Neural Nets
- Hands-on with NVIDIA DIGITS

Machine Learning



- Machine learning is giving computers the ability to analyze, generalize, think/reason/behave like humans.
- Machine learning is transforming medical research, financial markets, international security, and generally making humans more efficient and improving quality of life.
- Inspired by the mammalian brain, deep learning is machine learning on steroids- bigger, faster, better!

Unleashing of Intelligence

- Machines will slowly match, then quickly surpass human capabilities.
- Today it is exciting/scary/fun to drive next to an autonomous car.

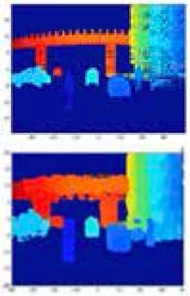
Tomorrow it may be considered irresponsible for a human to relinquish control from a car that has faster reaction times, doesn’t drink/text/get distracted/tired, and is communicating with surrounding vehicles and objects.

https://www.designnews.com/electronics-test/bmw-will-use-innoviz-lidar-its-autonomous-vehicles/117710752958701

- U.S. car fatality is about 1.16 deaths per 100M miles[1].
- Current driverless technology requires human intervention every few dozen to few thousand miles.

Sample Autonomous Montage from YouTube

- Investing billions of $$ to close the gap:
  1. Redundancy (vision, LiDAR, RADAR)
  2. Smarter decision making:
     usefulness vs. safety, i.e. can be safe going very slow, but will take too long to arrive at destination!



[1] 2018 IIHS statistics: https://www.iihs.org/iihs/topics/t/general-statistics/fatalityfacts/state-by-state-overview

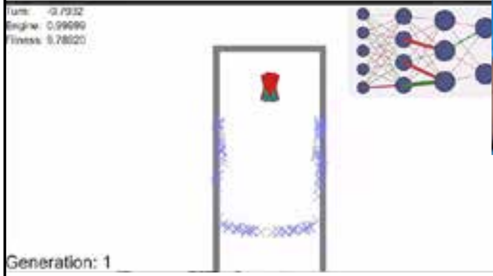
---

2017: The Year of AI:

The Wall Street Journal, Forbes, and Fortune

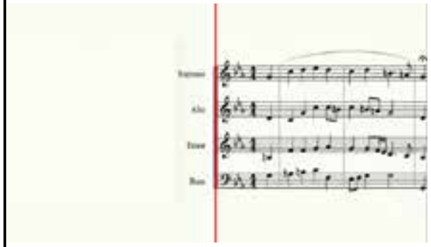
NEC Face Recognition

SONY Playstation Virtual Reality



Evolutionary Reinforcement Learning

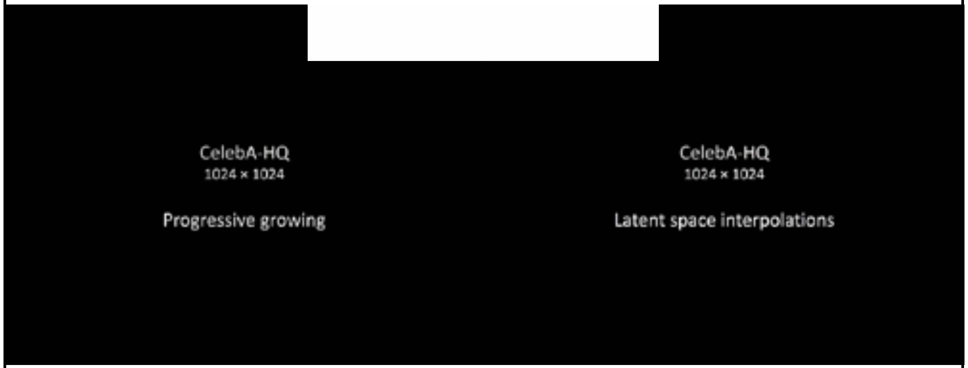
2017: The Year of AI:

The Wall Street Journal, Forbes, and Fortune

DeepBach

NVIDIA Autonomous Car
Detection & Segmentation

YOLO v2 Object Detection

Awesome 2018 Technology



http://research.nvidia.com/sites/default/files/pubs/2017-10_Progressive-Growing-of/karras2017gan-paper.pdf

Awesome 2018 Technology

NVIDIA Drive

Awesome 2018 Technology

Email smart compose sentence completion

https://ai.googleblog.com/2018/05/smart-compose-using-neural-networks-to.html?m=1

Awesome 2018 Technology

Goggle Duplex: https://www.youtube.com/watch?v=D5VN56jQMWM

Awesome 2018 Technology

Giving Bruno Mars Dance Moves to Anyone

Android Companions

Erica, from Hiroshi Ishiguro

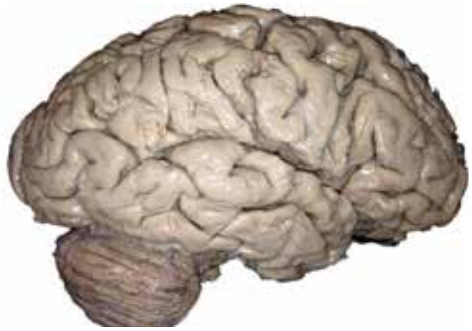
2019 and Beyond

Closing the Gap between Man and Machine

GauGAN: https://www.youtube.com/watch?v=p5U4NgVGAwg

Tesla AutoPilot, V2:
https://www.youtube.com/watch?v=_1MHGUC_BzQ

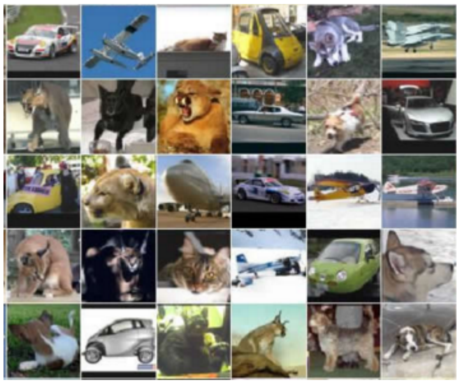
The Human Brain



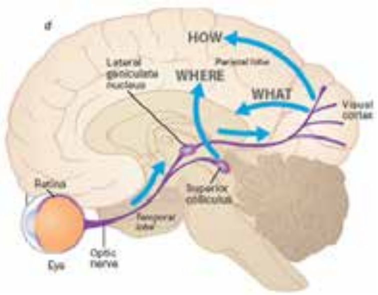
- We’ve learned more about the brain in the last 5 years than we have learned in the last 5000 years!
- It controls every aspect of our lives, but we still don’t understand exactly how it works.

The Brain on Pattern Recognition

- Airplane, Cat, Car, Dog



STL-10 dataset



http://thebraingeek.blogspot.com/2012/08/blindsight.html

The Brain on Pattern Recognition

Despite Changes in Deformation:

The Brain on Pattern Recognition

Despite Changes in Occlusion:

The Brain on Pattern Recognition

Despite Changes in Size, Pose, Angle:

Tardar Sauce “Grumpy Cat”

The Brain on Pattern Recognition

Despite Changes in Background Clutter:

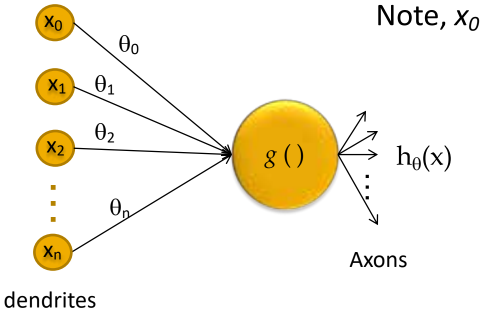
The Brain on Pattern Recognition

Despite Changes in Class Variation…

Neural Nets on Pattern Recognition

- Instead of trying to code simple intuitions/rules on what makes an airplane, car, cat, and dog…
- We feed neural networks a large number of training samples, and it will automatically learn the rules!
- We will learn the magic behind this today!

Artificial Neuron



Note, x_0 is the bias unit, x_0=1

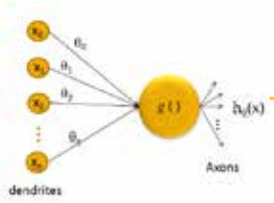
$$x = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \qquad \theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \vdots \\ \theta_n \end{bmatrix}$$

Activation function

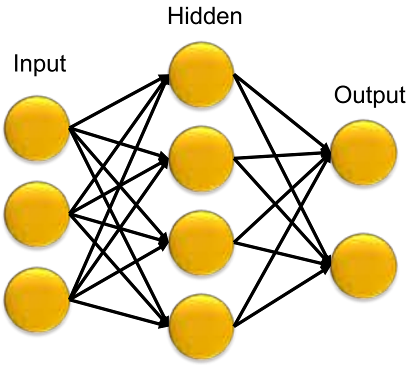
$$h_\theta(x) = g(x_0\theta_0 + x_1\theta_1 + \ldots + x_n\theta_n) = g\left(\sum_{i=0}^{n} x_i\theta_i\right)$$

$$h_\theta(x) = g(\theta^T x)$$

Artificial Neural Networks

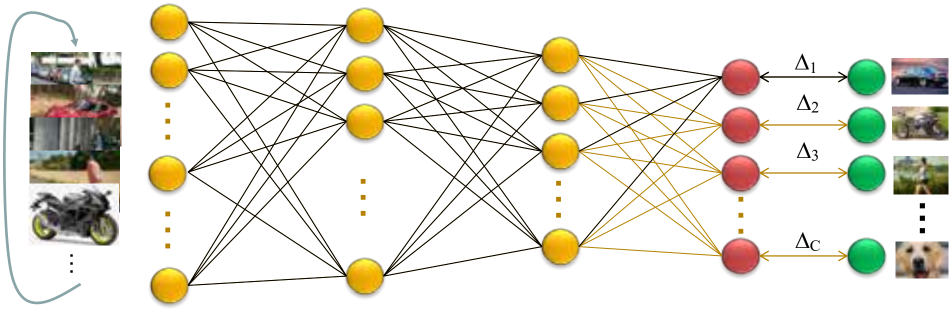


- Artificial Neural Network (ANN) – A network of interconnected nodes that “mimic” the properties of a biological network of neurons



4-Layer ANN Fully Connected Topology

Input Layer Hidden Layer 1 Hidden Layer 2 Output Layer



A 20x20 image would have 400 input nodes

a nodes

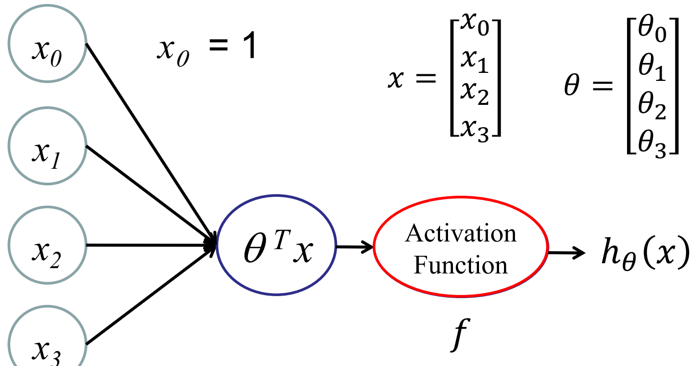
b nodes

C nodes, where *C* is the number of classes

Backpropagation (~1985) uses $\frac{\partial \Delta}{\partial w}$ for learning

Learning happens in the weights- each line is a weight.

Neuron Model



Bias unit

x_0 = 1

x_0

x_1

x_2

x_3

$$x = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix} \qquad \theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \theta_3 \end{bmatrix}$$

$\theta^T x$ → Activation Function → $h_\theta(x)$

f

$$h_\theta(x) = g(x_0\theta_0 + x_1\theta_1 + \ \dots + x_n\theta_n) = g\left(\sum_{i=0}^{n} x_i\theta_i\right)$$

$$h_\theta(x) = g(\theta^T x)$$

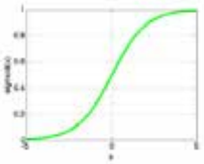
Activation Function Comparison

- Tanh
- Sigmoid

Gradient of both saturates at zero. Sigmoid also non-zero centered, so in practice tanh performs better.

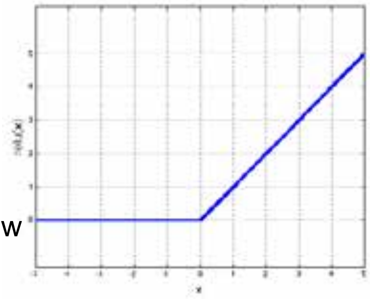
$$h_\theta(x) = \frac{e^{\theta^T x} - e^{-\theta^T x}}{e^{\theta^T x} + e^{-\theta^T x}}$$

$$h_\theta(x) = \frac{1}{1 + e^{-\theta^T x}}$$



- Rectified Linear Units (ReLU)
  - Better for high dynamic range
  - Faster learning
  - Overall better result
  - Neurons can "die" if allowed to grow unconstrained

$$h_\theta(x) = max(0, x)$$



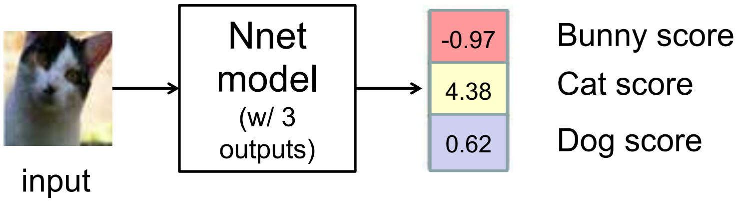
Where Do Weights Come From?

- The weights in a neural network need to be learned such that the errors are minimized.
- Just like logistic regression, we can write a cost function.
- Similar to gradient descent, we can write an iterative procedure to update weights, with each iteration decreasing our cost.
- These iterative methods may be less efficient than a direct analytical solution, but are easier to generalize.

Backpropagation

- We need to solve weights of a network so that the error is minimized.
- Weights can be refined by changing each weight by an amount proportional to the partial derivative of the error with respect to each weight.
- Partial derivatives can be calculated by iteratively changing each weight and measuring the corresponding change in error.
- Hard to do with millions of weights!
- In 1986, a technique called back-propagation was introduced (D. E. Rumelhart, G. E. Hinton, and R. J. Williams "Learning representations by back-propagating errors," J. *Nature* 323, 533-536, 1986).

Multiclass Loss Functions



- The input image scores highest against cat, but is also somewhat similar to dog.
- How do we assign a loss function?

Activation Function of Output Layer

- Sigmoid returns 0 or 1 for each output node.
- What if you wanted a confidence interval?
- Use a linear activation function for regression: $a^{(l)}=z^{(l)}$
- Softmax often used for classification:

$$a_c^{(L)} = h_\theta(x)_c = g\left(z_c^{(L)}\right) = \frac{exp\left(z_c^{(L)}\right)}{\sum_{c=1:C} exp\left(z_c^{(L)}\right)}$$

← *exp()* of each output node

← Sum of all output nodes

- Note: Only the output layer activation function changes- all hidden layer nodes activation functions would be the sigmoid/tanh/ReLU function.

input

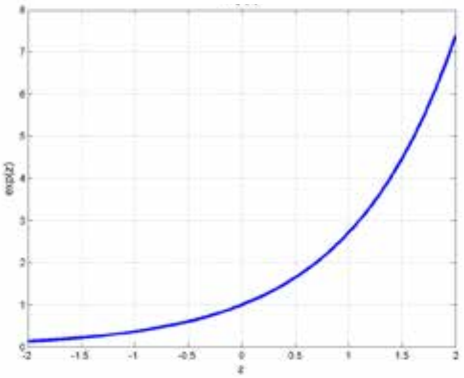
Nnet model (w/ 3 outputs)

-0.97	Bunny score	0.38	0.46
4.38	Cat score	79.8	97.3
0.62	Dog score	1.86	2.26

$$exp\left(z_c^{(L)}\right)$$

$$\frac{exp\left(z_c^{(L)}\right)}{\sum_{c=1:C} exp\left(z_c^{(L)}\right)}\times 100$$

exp(z), where z is each output node



exp() take in values +/- values and returns all positive values.

Softmax Example

input

Nnet model (w/ 3 outputs)

0.1	Bunny score
2	Cat score
1	Dog score

0.1
2
1

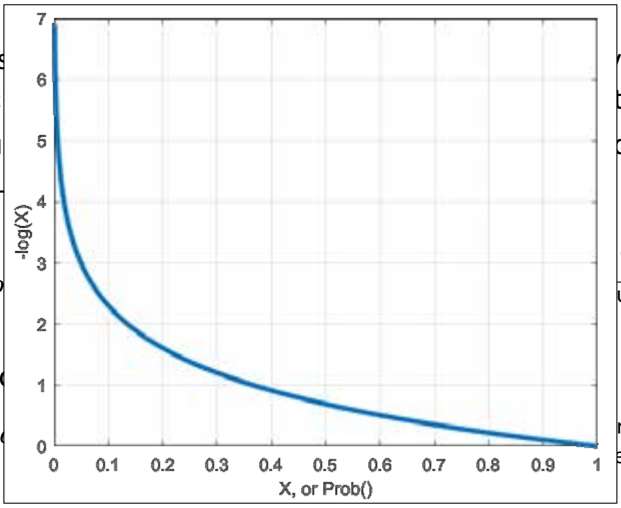
softmax

$$\frac{exp\left(z_c^{(L)}\right)}{\sum_{c=1:C} exp\left(z_c^{(L)}\right)}$$

0.1
0.7
0.2

Probabilities!

Most Common Loss Functions



- The cos from logistic tion.
- For mu ds:

1. Cross-

GT node)

$Loss^{(i)} = -lo$ ut) of all nodes

2. Multi

L rect – es

Most Common Loss Functions

- The cost function we previously used was a direct copy from logistic regression and works great for binary classification.
- For multi-class, there are two popular data loss methods:

1. Cross-entropy loss, which uses softmax:

$$Loss^{(i)} = -log\left(\frac{exp\left(out_{yi}^{(i)}\right)}{\sum_{c=1:C} exp\left(out_c^{(i)}\right)}\right)$$

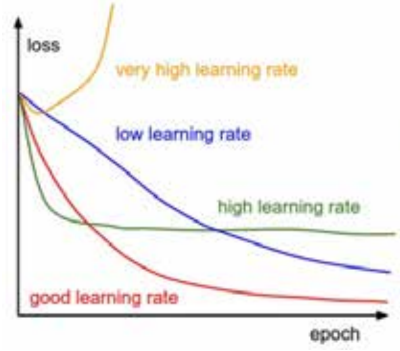
Loss for sample *i* = $\frac{\text{exp(output of GT node)}}{\text{Sum of exp(output) of all nodes}}$

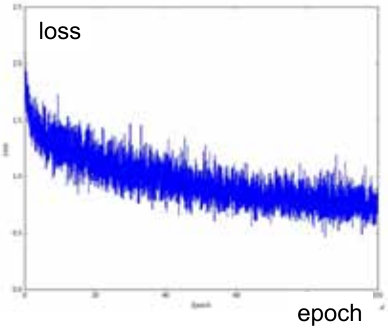
2. Multiclass SVM Loss (Weston Watkins formulation):

$$Loss^{(i)} = \sum_{j \neq yi} max(0, out_j - out_{yi} + \Delta)$$

Sum of incorrect – correct classes

Examples of Learning Rate and Batch Size

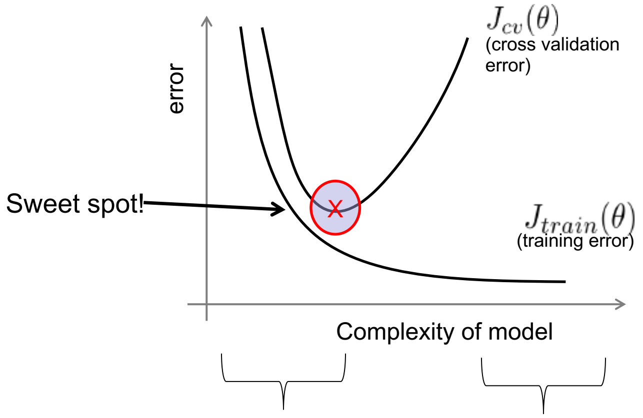




This batch size could be made a little larger to shrink the variance

---

Bias (underfit) vs. Variance (overfit) errors

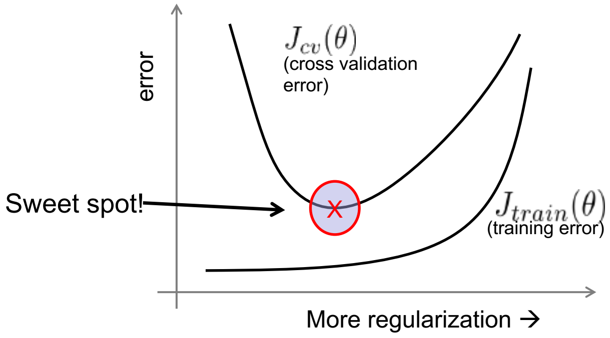


Model too simple. Too high error on train and test

Model too complex. Overfitting to training set

Adopted from:
Andrew Ng, ML class

Regularization Tuning



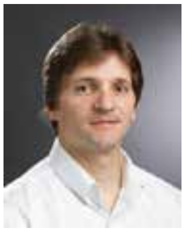
Adopted from:
Andrew Ng, ML class

DEEP LEARNING INSTITUTE

www.nvidia.com/dli

https://www.rit.edu/mil

R·I·T



Raymond W. Ptucha
Assistant Professor, Computer Engineering
Director, Machine Intelligence Laboratory
Rochester Institute of Technology

Email: rwpeec@rit.edu
Phone: +1 (585) 797-5561
Office: GLE (09) 3441