Fundamentals of Deep Learning

Raymond Ptucha,
Rochester Institute of Technology
NVIDIA, Deep Learning Institute

Electronic Imaging 2020: SC20
January 28, 2019, 8:30am-12:45pm

IS&T International Symposium on
Electronic Imaging 2020
SCIENCE AND TECHNOLOGY

26–30 January 2020
Burlingame, California USA

1

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.

2

Agenda

- Part I- Intuition and Theory
  - 8:35-9:15pm: Introduction
  - 9:15-10:00pm: Convolutional Neural Networks
  - 10:00-10:40pm: Recurrent Neural Networks
- 10:40-11:00pm: Break
- Part II- Hands on
  - 11:00am-12:45pm: Hands-on exercises

4

Jordan Peele, BuzzFeed

https://www.youtube.com/watch?v=cQ54GDm1eL0

5

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!

LANDING AI

"AI (Artificial Intelligence) technology is now poised to transform every industry, just as electricity did 100 years ago. Between now and 2030, it will create an estimated $13 trillion of GDP growth."

Andrew Ng

Chairman and CEO, Landing AI

https://landing.ai/ai-transformation-playbook/?utm_source=MLYList&utm_medium=ButtonLink&utm_campaign=Playbook

Interest in Machine Learning Growing Faster Over Time

Interest over time for keywords "machine learning" , "deep learning"

https://trends.google.com/trends/explore?q=machine%20learning

Machine learning, cs229 is the most popular course at Stanford
Their deep learning class, cs231 went from 150 to 350 to 750 in 2015/16/17 respectively…

8

Pedestrian re-identification based on Tree branch network with local and global learning
Hui Li, Meng Yang, Zhihui Lai, Weishi Zheng, Zitong Yu
Comments: accepted by ICME2019(Oral)
Subjects: **Computer Vision and Pattern Recognition (cs.CV)**

[31] arXiv:1904.00386 [pdf, other]
PyramidBox++: High Performance Detector for Finding Tiny Face
Zhihang Li, Xu Tang, Junyu Han, Jingtuo Liu, Ran He
Subjects: **Computer Vision and Pattern Recognition (cs.CV)**

[32] arXiv:1904.00388 [pdf]
Multi-vision Attention Networks for On-line Red Jujube Grading
Xiaoye Sun, Liyan Ma, Gongyan Li
Subjects: **Computer Vision and Pattern Recognition (cs.CV)**

[33] arXiv:1904.00415 [pdf, other]
Self Supervised Occupancy Grid Learning from Sparse Radar for Autonomous Driving
Liat Sless, Gilad Cohen, Bat El Shlomo, Shaul Oron
Subjects: **Computer Vision and Pattern Recognition (cs.CV)**

[34] arXiv:1904.00420 [pdf, other]
Single Path One-Shot Neural Architecture Search with Uniform Sampling
Zichao Guo, Xiangyu Zhang, Haoyuan Mu, Wen Heng, Zechun Liu, Yichen Wei, Jian Sun
Subjects: **Computer Vision and Pattern Recognition (cs.CV)**

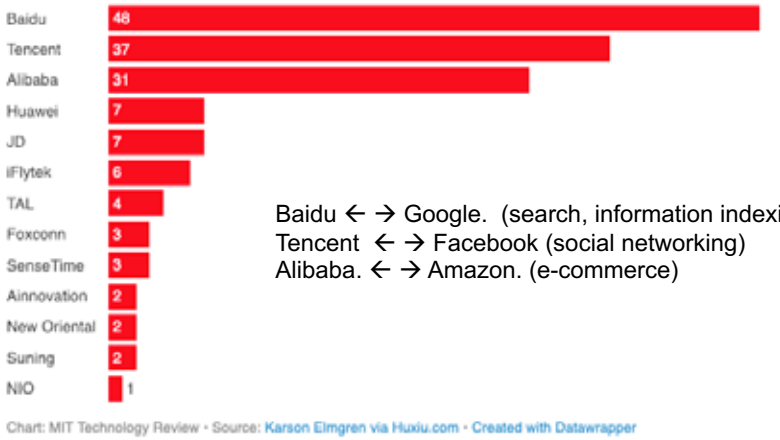
[35] arXiv:1904.00523 [pdf, other]
Toward Real-World Single Image Super-Resolution: A New Benchmark and A New Model
Jianrui Cai, Hui Zeng, Hongwei Yong, Zisheng Cao, Lei Zhang
Subjects: **Computer Vision and Pattern Recognition (cs.CV)**

[36] arXiv:1904.00537 [pdf, other]
Perceive Where to Focus: Learning Visibility-aware Part-level Features for Partial Person Re
Yifan Sun, Qin Xu, Yali Li, Chi Zhang, Yikang Li, Shengjin Wang, Jian Sun
Comments: 8 pages, 5 figures, accepted by CVPR2019

9

Who is Investing in Chinse AI Industry?

Number of companies funded Jan, 2019

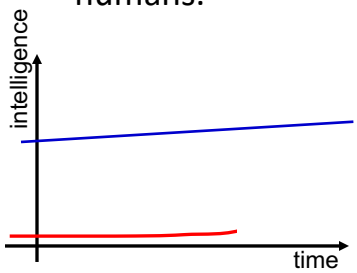


Baidu ← → Google. (search, information indexing)_
Tencent ← → Facebook (social networking)
Alibaba. ← → Amazon. (e-commerce)

https://www.technologyreview.com/s/612813/the-future-of-chinas-ai-industry-is-in-the-hands-of-just-three-companies/

The point of Singularity

- The point of singularity is when computers become smarter than humans.



Evolution of biology
Advancement of technology

Unleashing of Intelligence

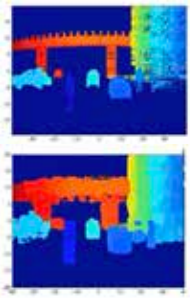
- Machines will slowly match, then quickly surpass human capabilities.
- Today it is exciting/scary/fun to drive next to an autonomous car.

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.
- 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!

Sample Autonomous Montage from YouTube



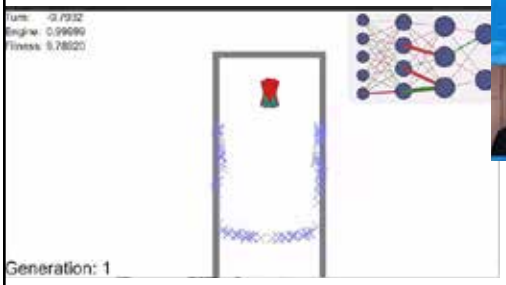
[1] 2018 IIHS statistics: https://www.iihs.org/iihs/topics/t/general-statistics/fatalityfacts/state-by-state-overview

2017: The Year of AI:

The Wall Street Journal, Forbes, and Fortune

NEC Face Recognition

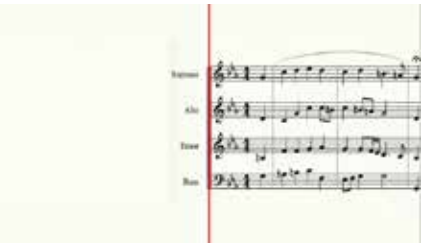
SONY Playstation Virtual Reality



Evolutionary Reinforcement Learning

2017: The Year of AI:

The Wall Street Journal, Forbes, and Fortune

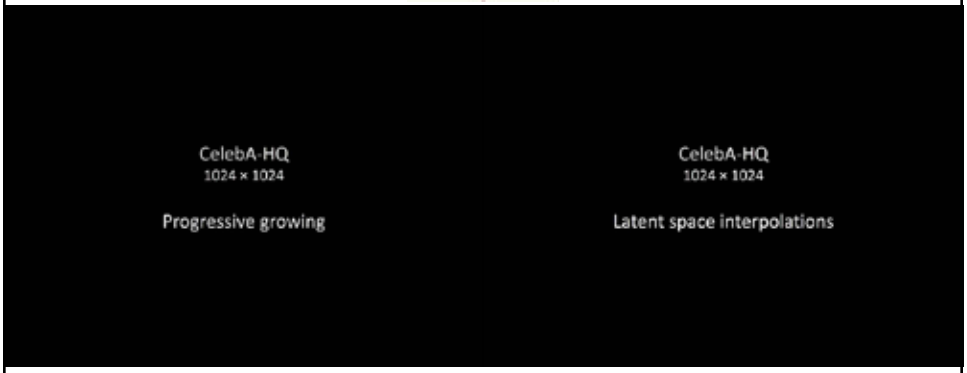


DeepBach

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

25

Awesome 2018 Technology

NVIDIA Drive

27

Awesome 2018 Technology

Email smart compose sentence completion

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

28

Awesome 2018 Technology

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

29

Awesome 2018 Technology

Giving Bruno Mars Dance Moves to Anyone

Android Companions

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

Anticipate More Change

- Fleets of autonomous vehicles may make car purchases a thing of the past.
- Experiences such as going to a concert, sports, travel will be replaced by virtual reality.
- Knowledge will be the new money:
  - Identify knowledge at the relevant time, then learn quickly
  - Need to be able to convince others your skills are valuable
  - Need to constantly learn throughout career

32

ML Trends

- If 2013/2014 were the year of CNNs, 2015/2016 were LSTMs, 2017/2018 was GANs.
- In 2017 AI fear mongering peaked, in 2018 press has come to terms with AI limitations.
- In 2018 focus started to turn to fairness, bias, interpretability.
- In 2019 we saw robustness, self-supervision, generation/detection of fake content.
- While current deep learning is not capable of achieving general AI, it may form building blocks.

33

AI Used in Court of Law

- Law enforcement has long been trying to predict predict who is more likely to commit a new crime or who is more dangerous.
- Correctional Offender Management Profiling for Alternative Sanctions (COMPAS) assigns a 1 (low) to 10 (high) risk.
- Many states, including New York, use this to decide if arrested individuals should be let go, held for bail, or locked up without bail.
- In a report published in 2010, NY said it 71% accurate at predicting crime activity (16K participants).

- https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing
- https://www.technologyreview.com/s/613508/ai-fairer-than-judge-criminal-risk-assessment-algorithm/?utm_source=newsletters&utm_medium=email&utm_campaign=the_algorithm.unpaid.engagement

34

Autonomous Driving

- Waymo (Google/Alphabet Inc.) purchased thousands of Chrysler Pacifica minivans- logged over 5M miles in 25 cities of autonomous driving. Commercial service begins in Phoenix this year. Deals with Jaguar and Honda in the works. In CA last year, reports the lowest number of times a driver takes over control and only 3 collisions in 350K miles. Can go full speed.
- GM is following closely with ~200 Chevy Bolt (SoftBank Vision Fund invested $2.25B), CA, AZ, MI. In CA last year, 22 collisions in 132K miles. Can only go up to 25 mph. Reports their autonomous car costs $200K (Chevy Bolt only $35K), with LiDAR sensors costing $30K (but estimate this will drop to hundreds of dollars soon).
- High profile fatalities to a pedestrian (Uber) and driver (Tesla) are giving consumer skepticism.
- Ride-hailing and ride-sharing are predicted to grow from $5B to $285B by 2030 (Goldman Sachs). Eliminating the driver could double operating margins.

35

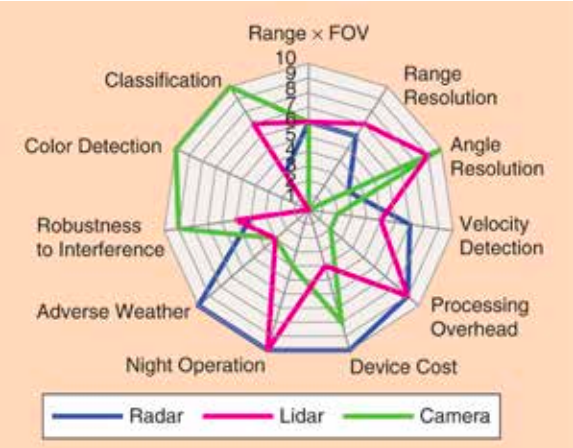
Autonomous Driving

- In Europe, Daimler seems to have the lead. Using NVIDIA technology, test cars can drive at Level 5 autonomy. Will offer Level 3 autonomy in the 2021 Mercedes lineup (car will mostly drive autonomous, but driver takes over as necessary).
- Other big players include Aptiv, Zoox, Renault-Nissan, Volkswagen, BMW, Toyota, Ford, Volvo, Hyuandai, Fiat Chrysler, Uber, Tesla, Baidu.

- Navigant Research (company that studies auto technology)
- https://medium.com/bloomberg/whos-winning-the-self-driving-car-race-39ed1aa58e93

36

Autonomous Driving

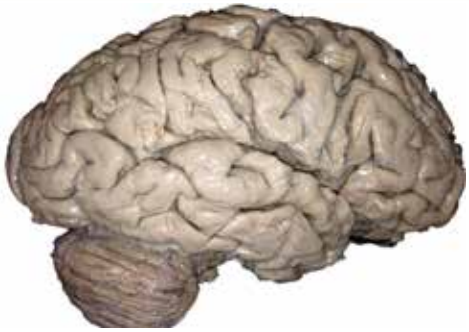


https://read.nxtbook.com/ieee/signal_processing/september_2019/interference_in_automotive_ra.html?mkt_tok=eyJpIjoiTVdWaE9URmlPVFF4TWp%E2%80%A6

*Note: Radar and Camera is actual data from current vehicles. LiDAR is estimated based upon beta models.

37

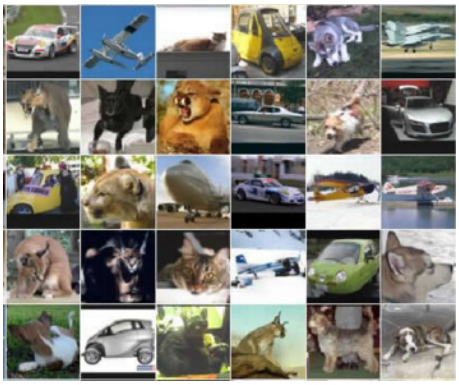
The Human Brain



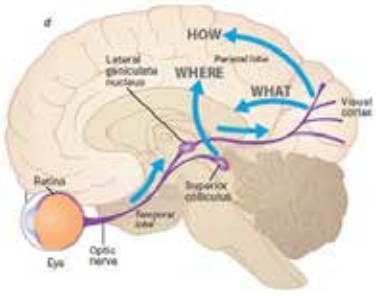
- We've learned more about the brain in the last 5 years than we have learned in the last 5000 years!
- It controls every aspect of our lives, but we still don't understand exactly how it works.

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:

43

The Brain on Pattern Recognition

Despite Changes in Occlusion:

44

The Brain on Pattern Recognition

Despite Changes in Size, Pose, Angle:

Tardar Sauce “Grumpy Cat”

45

The Brain on Pattern Recognition

Despite Changes in Background Clutter:

46

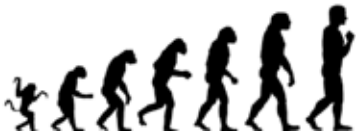
The Brain on Pattern Recognition

Despite Changes in Class Variation…

47

Teaching Computers to See

- It took evolution 540M years to develop the marvel of the eye-brain.



- Lets say a child collects a new image every 200msec.
- By age 3, this child has processed over 100M images.

$$(5\ images/sec)(60sec/min)(60min/hr)(12hr/day)(365days/yr)(3yrs) = 236M$$

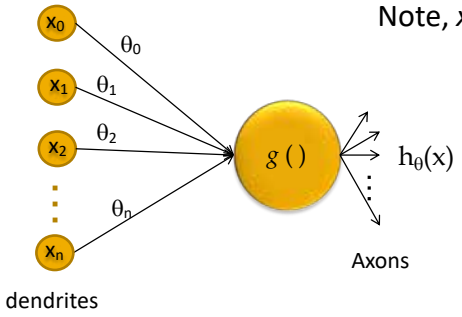
- Today's computers can do this in a few days…

48

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

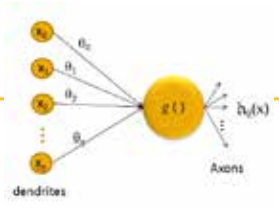
$$x = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \quad \theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \vdots \\ \theta_n \end{bmatrix}$$

Activation function

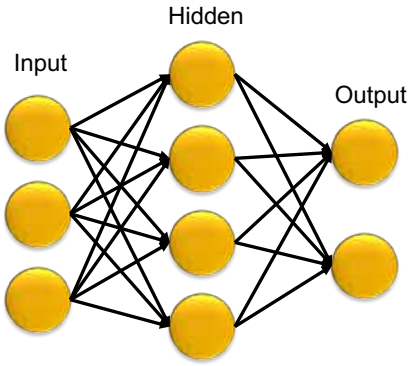
$$h_\theta(x) = g(x_0\theta_0 + x_1\theta_1 + \ldots + x_n\theta_n) = g\left(\sum_{i=0}^{n} x_i\theta_i\right)$$

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

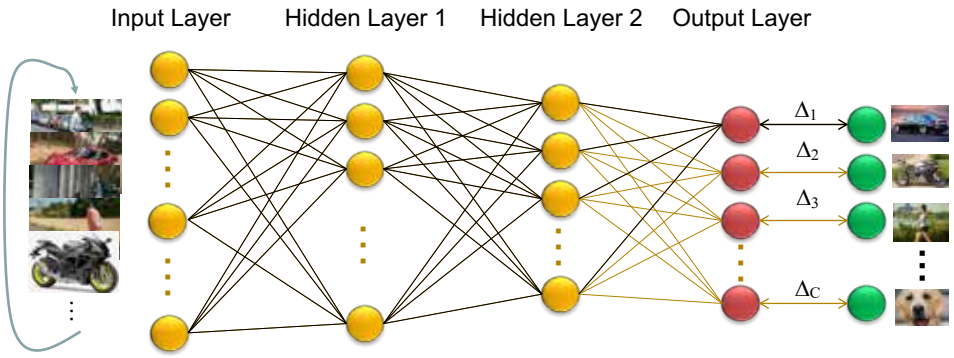
Artificial Neural Networks



- 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



A 20x20 image would have 400 input nodes

a nodes

b nodes

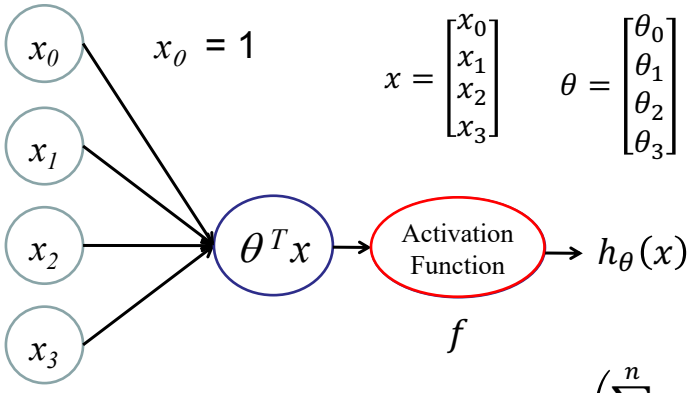
C nodes, where *C* is the number of classes

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

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

Neuron Model

Bias unit



$x_0 = 1$

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

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

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

Activation Function

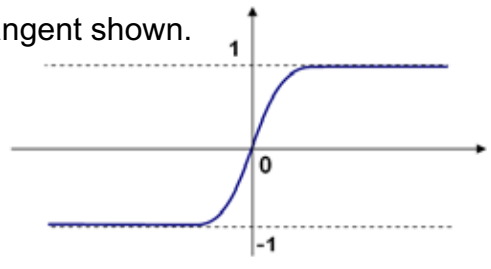
- Basic operation of an artificial neuron involves summing its weighted input signal and applying a threshold, called an activation function.
- If the sum is greater than threshold, fire, otherwise don't.
- A linear activation function is unbounded and limits the nonlinear properties of our net.
- Non-linear hyperbolic tangent shown.

Sigmoid:

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

Hyperbolic tangent:

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

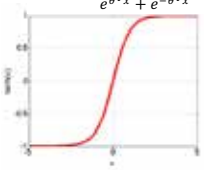


Activation Function Comparison

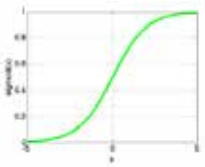
- Tanh
- Sigmoid

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

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

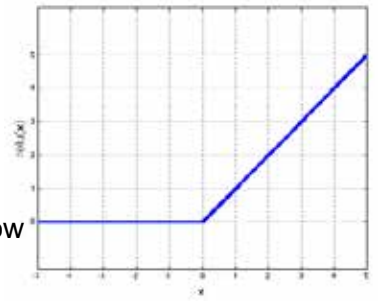


$h_\theta(x) = \frac{1}{1 + e^{-\theta^T x}}$



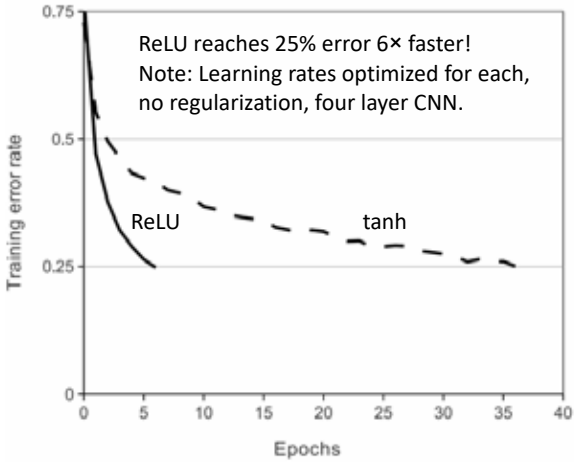
- Rectified Linear Units (ReLU)
  - Better for high dynamic range
  - Faster learning
  - Overall better result
  - Neurons can "die" if allowed to grow unconstrained

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



Tanh vs. ReLU on CIFAR-10 dataset [Krizhevsky'12]

CIFAR-10
Classify image as one of these ten classes:



- A four-layer convolutional neural network with ReLUs (solid line) reaches a 25% training error rate on CIFAR-10 six times faster than an equivalent network with tanh neurons (dashed line). The learning rates for each network were chosen independently to make training as fast as possible. No regularization of any kind was employed. The magnitude of the effect demonstrated here varies with network architecture, but networks with ReLUs consistently learn several times faster than equivalents with saturating neurons- **Krizhevsky et al., 2012**.

Other Activation Units

- ReLUs are the max of a linear unit and zero
- How about allowing a small slope on the negative side of the ReLU? (PReLU, He '15)
- How about take the max of a whole bunch of linear units? (Maxout, Goodfellow '13)
- How about using a small group of highly interconnected non-linear units as a basic module for backgropagation (Hinton)
  - Similar to a cortical column!
- How about learning parameters of activation function along with the weights? (Dushkoff '16)

ReLU vs Parametric ReLU (PReLU)

$$f(y_i) = \begin{cases} y_i & if\ y_i > 0 \\ 0 & if\ y_i \leq 0 \end{cases} \qquad f(y_i) = \begin{cases} y_i & if\ y_i > 0 \\ a_i y_i & if\ y_i \leq 0 \end{cases}$$

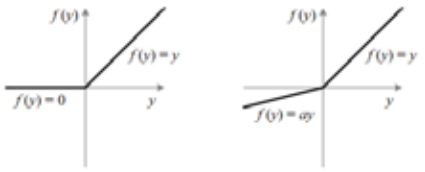


Figure 1. ReLU vs. PReLU. For PReLU, the coefficient of the negative part is not constant and is adaptively learned.

He et al., 2015

The subscript *i* just means PReLU is computed independently for each channel.

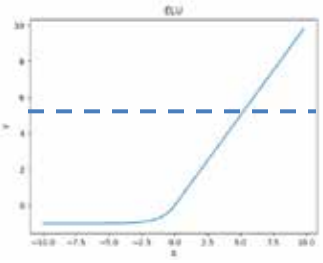
ELU vs. ReLU

ELU

With Exponential Linear Units (ELU), we can have a mean activation that is close to 0 and it is an exponential function. ELU does not saturate for large values of x. It is expressed as,

$$f(x) = \begin{cases} x & x >= 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$

Here α is a parameter. See: https://arxiv.org/abs/1511.07289



Note: output of activation functions are often capped to say 4 or 5

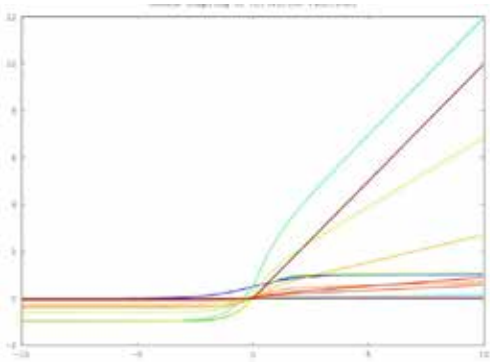
http://saikatbasak.in/sigmoid-vs-relu-vs-elu/

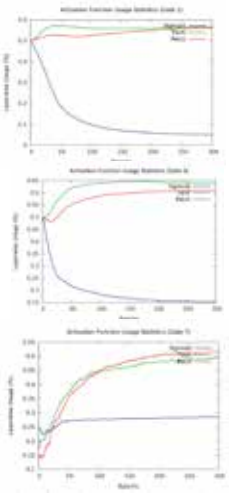
60

Learnable Activation Functions [Dushkoff '16]

If restrict to sigmoid, tanh, ReLU, early layers use tanh/ReLU about the same, then tanh more common, then in last layer, ReLU more common.

Random sampling of learned activation functions (used parametric function that can learn any generic sigmoid shape).

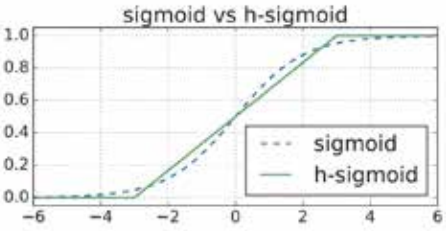


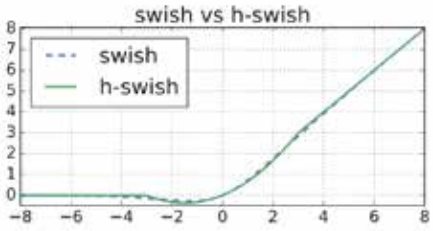


61

Swish Function [Howard '19]

- More non-linear variants…





- The h-variants are the hard variants…

Howard et al. "Searching for MobileNetV3," https://arxiv.org/pdf/1905.02244.pdf

62

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.

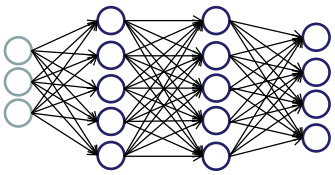
63

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).

Hyperparameters vs. parameters

- Hyperparameters are the tuning parameters for a nnet- say number of layers, nodes per layer, learning rate, momentum, regularization, etc.
- Parameters are the weights that are being learned. Ignoring bias terms, the below network has 3x5+5x5+5x4 = 60 parameters.
  - If we include bias terms, we have 4x5+6x5+6x4=74 learnable parameters.



Note: deep nets may contain 100M parameters with 20 layers!

Cost Function

Add regularization to prevent overfitting to the training set.

Logistic Regression Cost Function:

$$J(\theta) = \left[-\frac{1}{n} \sum_{i=1}^{n} y^{(i)} log\left(h_\theta\left(x^{(i)}\right)\right) + \left(1 - y^{(i)}\right) log\left(1 - h_\theta\left(x^{(i)}\right)\right) \right] + \frac{\lambda}{2n} \sum_{j=1}^{D} \theta_j^2$$

n = number training samples, *D* is the dimension of each sample

Foreach sample

Foreach class (or foreach output node)

C = number of classes
L = number of layers
s_l = number of nodes in layer *l*

Neural Network Cost Function:

$$J(\theta) = \left[-\frac{1}{n} \sum_{i=1}^{n} \sum_{c=1}^{C} y^{(i,c)} log\left(h_\theta\left(x^{(i)}\right)\right)_c + \left(1 - y^{(i,c)}\right) log\left(1 - h_\theta\left(x^{(i)}\right)\right)_c \right] +$$

$$\frac{\lambda}{2n} \sum_{l=1}^{L} \sum_{i=1}^{s_l} \sum_{j=1}^{s_{l-1}} \left(\theta_{ji}^{[l]}\right)^2$$

For *L* layers
Each layer *l* has s_l nodes
θ between layer *l-1* and layer *l*
Note: we don't regularize bias term

To From

Initialization of Weights

- We initialize weights to small ± values centered on 0.
- If they were all initialized to 0, the network wouldn't learn anything.
  - ie: the output of each node would be equal, therefore the update would update each term identically.
- If they were all initialized to large values (values > +1 or < -1), the inputs to each node would be saturated.
- If they were all initialized to small values around 0, we would be in the linear portion of the activation function.
  - W1 = 0.001*rand(length(h1),length(h2))

Initialization of Weights

- Although the uniform distribution is good, the more inputs to a node, the greater its variance.
- To set output distribution of all nodes equal (this empirically improves convergence), use "Xavier initialization"

h1=# input nodes

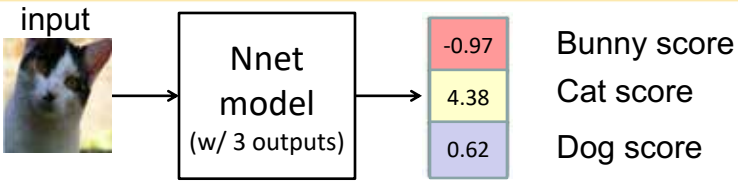
Transfer Learning

- Other solutions restrict weights: $-\epsilon_{init} < \theta_{ji}^{(l)} < \epsilon_{init}$

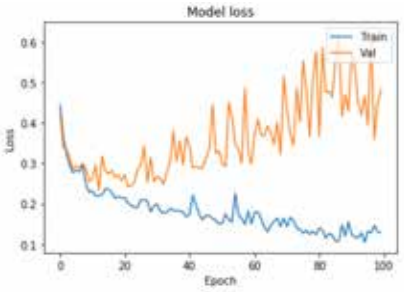
  $\epsilon_{init} = \sqrt{6}/\sqrt{s_l + s_{l+1}}$ s_l , s_{l+1} are the No. nodes in layers around $\theta^{(l)}$

  - For deep ReLU networks, He2015 showed:
    - **W1 = 0.001*rand(length(h1),length(h2))./sqrt(2*length(h1))**
  - Works best and is recommended for them

Multiclass Loss Functions



- The input image scores highest against cat, but is also somewhat similar to dog.
- How confident are you that the input image is a cat?
- How do we assign a loss function so that we can see how well we are doing during training?



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

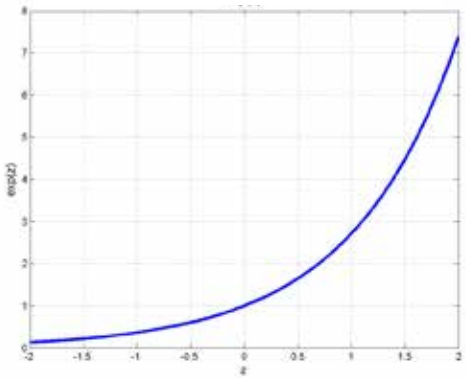
- Note: Only the output layer activation function changes- all hidden layer nodes activation functions would be the sigmoid/tanh/ReLU function.

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(z), where z is each output node



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

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

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

Most Common Loss Functions

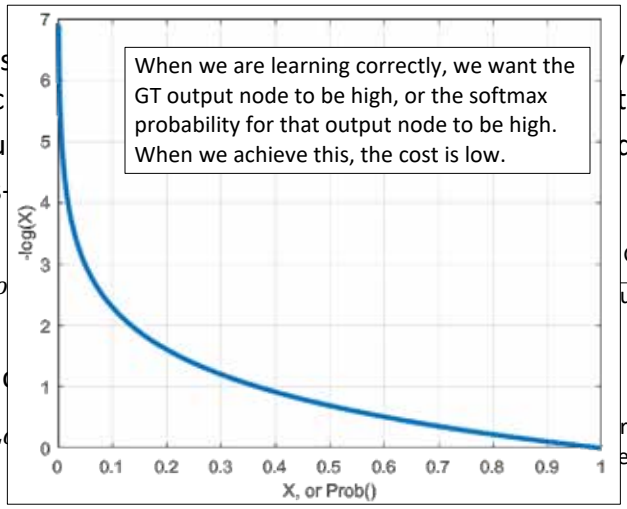
- The cos… from logistic… tion.
- For mu… ds:

1. Cross-…

$Loss^{(i)} = -lo…$ (GT node) / …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 = $\frac{\text{exp(output of GT node)}}{\text{Sum of exp(output) of all nodes}}$

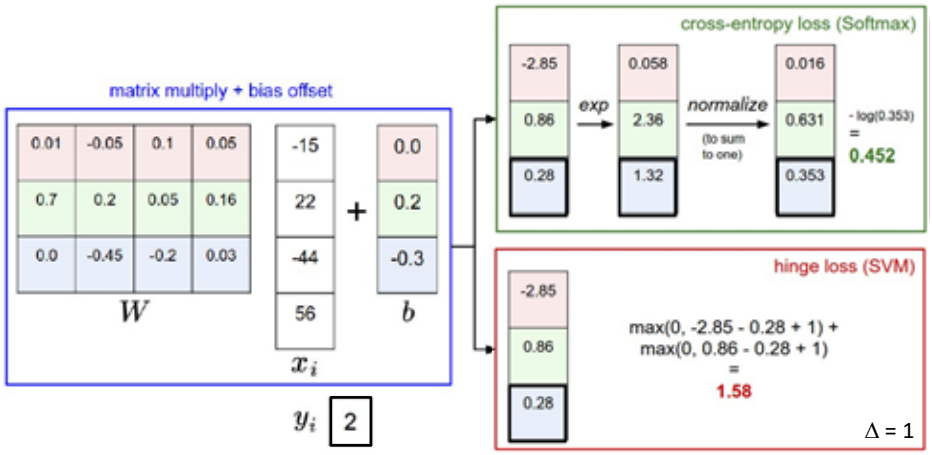
2. Multiclass SVM Loss (Weston Watkins formulation):

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

Sum of incorrect – correct classes

Loss Example

(based on cs231n, Li/Karpathy 2016)

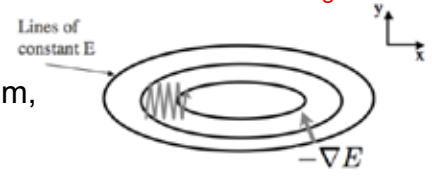


Local Minima

Marsland: Machine Learning

- Back propagation modifies the weights to minimize the cost function.
- It does this by approximating the gradient of the error and following it downhill.
- We could of course get stuck at a local minima.
- If we start near the global minimum, we often end up there.
- If we start at a local minimum, we often end up there!

Sometimes the direction of steepest descent is not in direction of global minima



Marsland: Machine Learning

Learning Rates

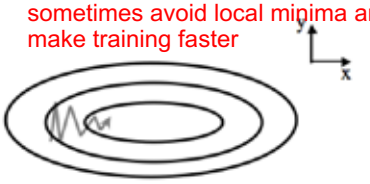
- Like gradient descent, we incorporate a learning rate so we don't take full steps in any one direction.
- In its simplest form, we have $w_{ij} = w_{ij} - \eta\, dw_{ij}$
- As training progresses, we generally anneal the learning rate over time, going from large to small steps. Three common approaches:
  1. Step decay: Either watch validation error and reduce learning rate whenever validation error stops improving, or automatically reduce by ~0.1 every ~20 epochs
  2. Exponential decay: $\alpha = \alpha_0 e^{-kt}$ (α_0 *and k are* hyperparameters, *t* is epoch.
  3. 1/*t* decay: $\alpha = \alpha_0/(1+kt)$ (α_0 *and k are* hyperparameters, *t* is epoch.

Momentum

Marsland: Machine Learning

- One solution is to train several networks, each at a different initialization.
- Another is to add a momentum term.
- Imagine a ball rolling down a hill- if it had little energy, it may get stuck in little dips.
- If it had a lot of energy, it would be more likely to skip over little dips, and keep rolling until it hopefully hits a global minimum.

Adding a momentum term can sometimes avoid local minima and make training faster



Marsland: Machine Learning

Momentum

Marsland: Machine Learning

- Momentum adds a contribution of our previous weight change to our current weight change.
- Change:

$$\Delta_{ij}^{(l)} = \Delta_{ij}^{(l)} + a_j^{(l)} \delta_i^{(l+1)}$$

- To:

$$\Delta_{ij}^{(l)}{}_t = \Delta_{ij}^{(l)}{}_{t-1} + a_j^{(l)} \delta_i^{(l+1)} + \alpha \Delta_{ij}^{(l)}{}_{t-1} \quad , \; 0 \le \alpha \le 1$$

Caution- these methods are quite effective, but more of an art than science!

The use of *t* and *t-1* emphasize the previous and current derivatives
A typical value of α is 0.9

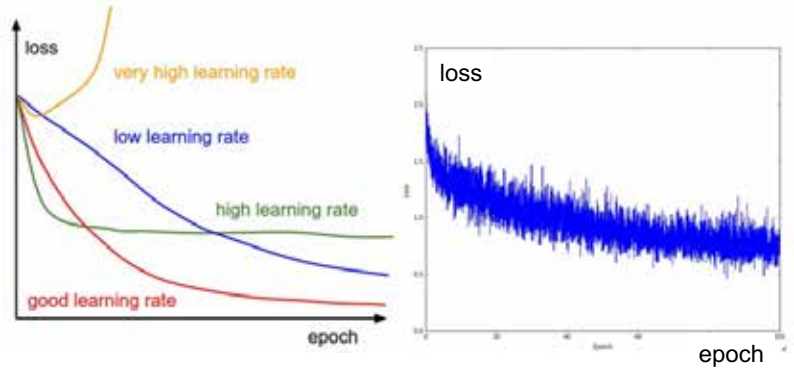
80

Sequential vs. Batch Training

- Back propagation can be done:
  - Sequential: one sample at a time,
    - Weights are shifted back and forth quite a bit
  - Group (minibatch): a group of samples at a time, or
  - Batch: all training samples at once
    - Weights are shifted in direction that makes most input samples better
    - Generally converges the fastest
- Recommended to use largest minibatch possible and stay within memory constraints of hardware.

81

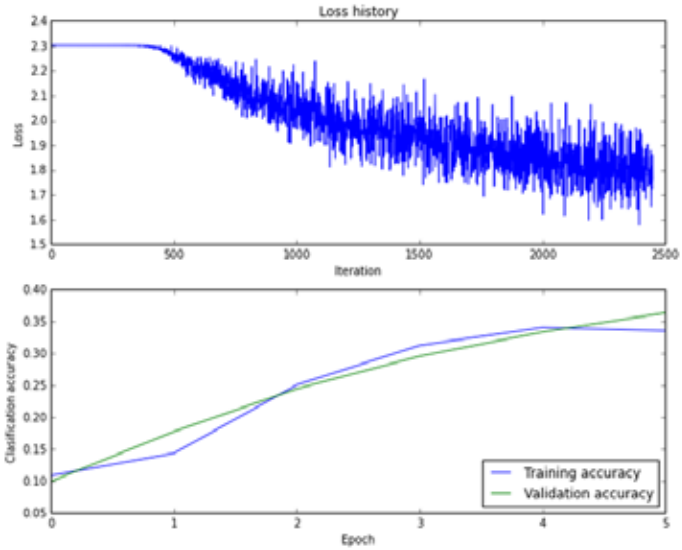
Examples of Learning Rate and Batch Size



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

84

Tuning Parameters

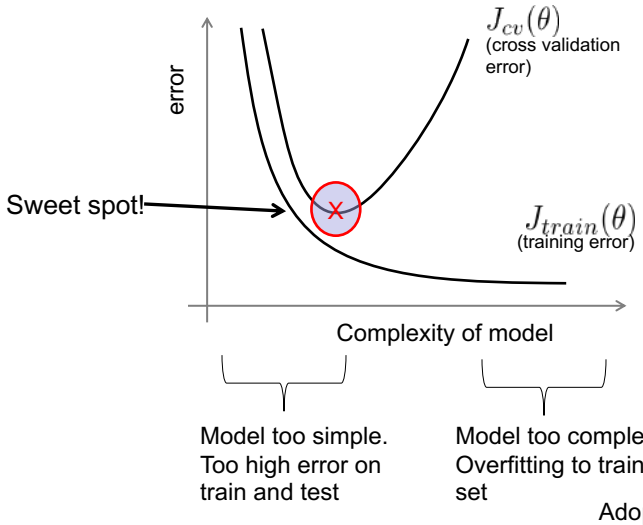


Curve too linear- increase learning rate.

Train and Test sets too similar- increase model complexity. Be careful, larger models susceptible to overfitting. Perhaps increase regularization.

85

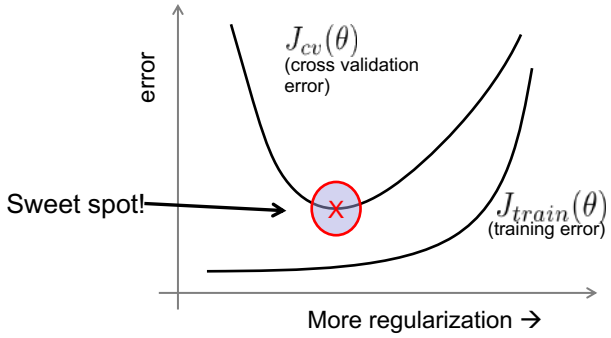
Bias (underfit) vs. Variance (overfit) errors



Adopted from: Andrew Ng, ML class

86

Regularization Tuning

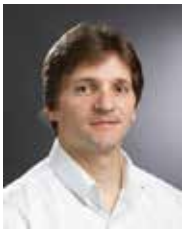


Adopted from: Andrew Ng, ML class

87

NVIDIA DEEP LEARNING INSTITUTE

For More Information: http://www.rit.edu/mil



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