Data Science

Deriving Knowledge from Data at Scale

Valentine Fontama

11 August 2016

Neural Networks and Deep Learning

Agenda

- Artificial Neural Networks
- Hands on with Neural Networks
- Deep Learning

Neural Networks

http://www.youtube.com/watch?v=GrugzF0-V3I

Neural Networks and Deep Learning

Neural Networks and Deep Learning is a free online book. The book will teach you about:

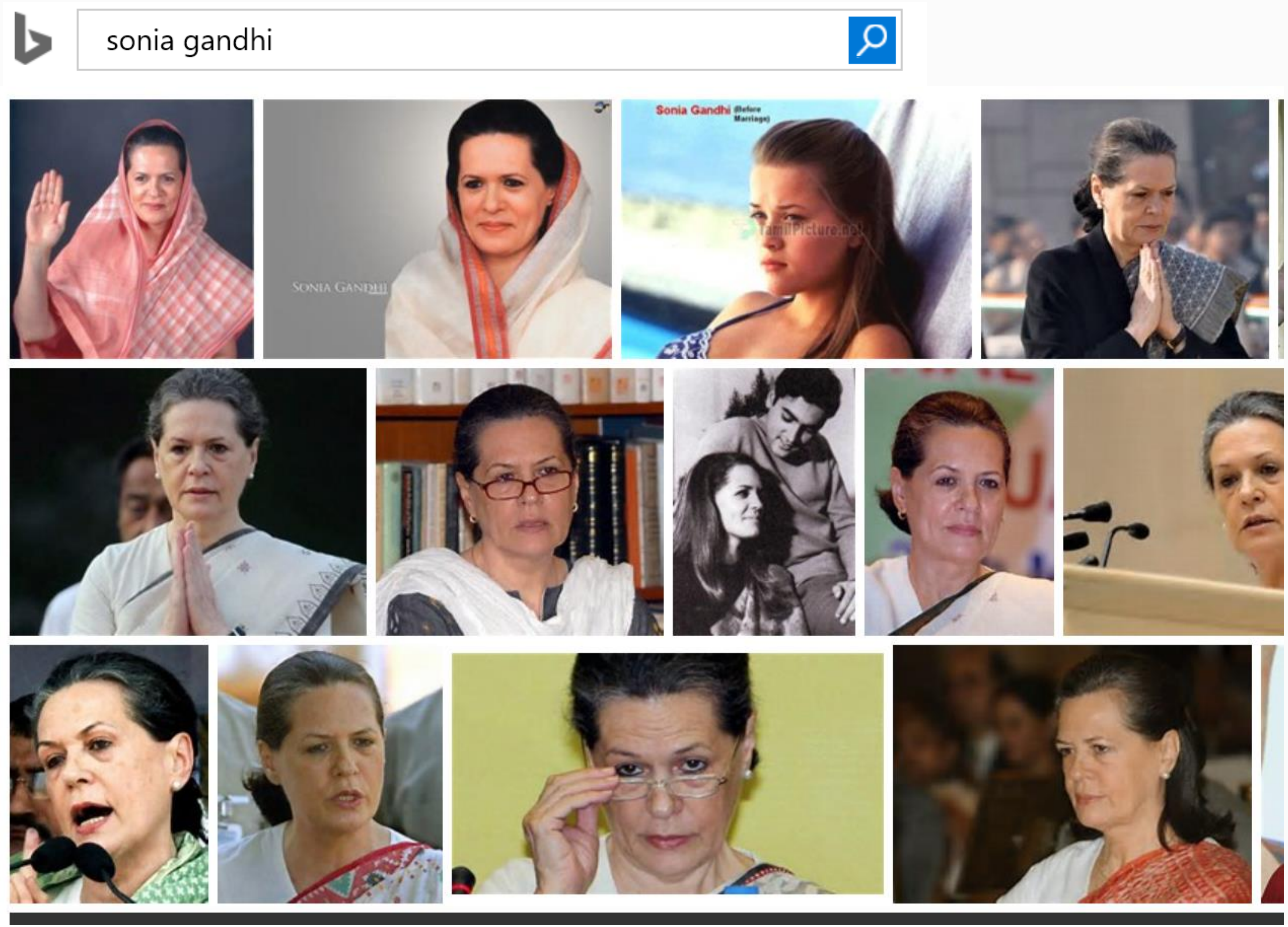
- Neural networks, a beautiful biologically-inspired programming paradigm which enables a computer to learn from observational data
- Deep learning, a powerful set of techniques for learning in neural networks

Neural networks and deep learning currently provide the best solutions to many problems in image recognition, speech recognition, and natural language processing. This book will teach you many of the core concepts behind neural networks and deep learning.

For more details about the approach taken in the book, see here. Or you can jump directly to Chapter 1 and get started.

Neural Networks and Deep Learning
- What this book is about
- On the exercises and problems
- Using neural nets to recognize handwritten digits
- How the backpropagation algorithm works
- Improving the way neural networks learn
- A visual proof that neural nets can compute any function
- Why are deep neural networks hard to train?
- Deep learning
- Appendix: Is there a *simple* algorithm for intelligence?
- Acknowledgements
- Frequently Asked Questions

Free online book by Michael Nielsen



Results include a photo of Actress, Reese Witherspoon

Query Image

Nearest Neighbor Images from the Index

Prev

DNN

Prev

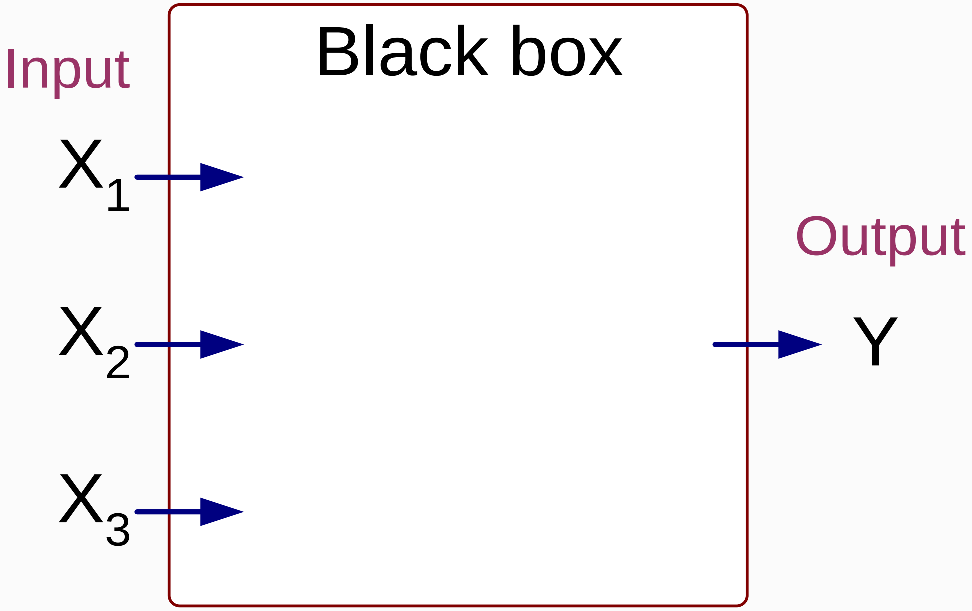
DNN

Prev

DNN

Artificial Neural Networks (ANN)

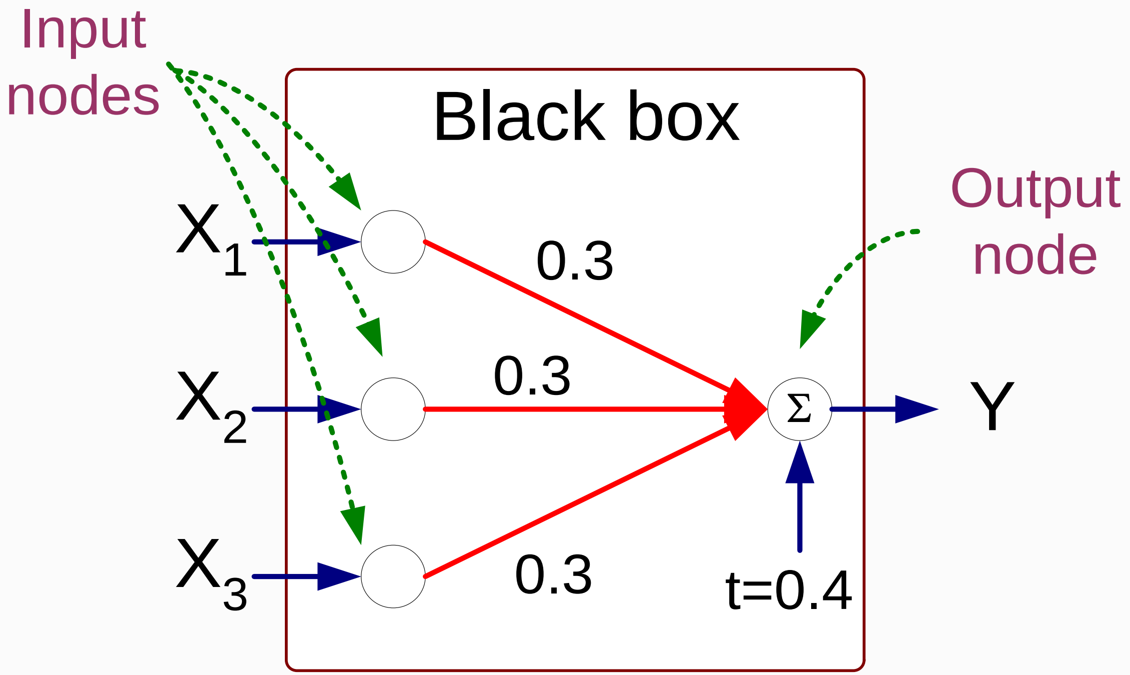
X_1	X_2	X_3	Y
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1
0	0	1	0
0	1	0	0
0	1	1	1
0	0	0	0



Output Y is 1 if at least two of the three inputs are equal to 1.

Artificial Neural Networks (ANN)

X_1	X_2	X_3	Y
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1
0	0	1	0
0	1	0	0
0	1	1	1
0	0	0	0

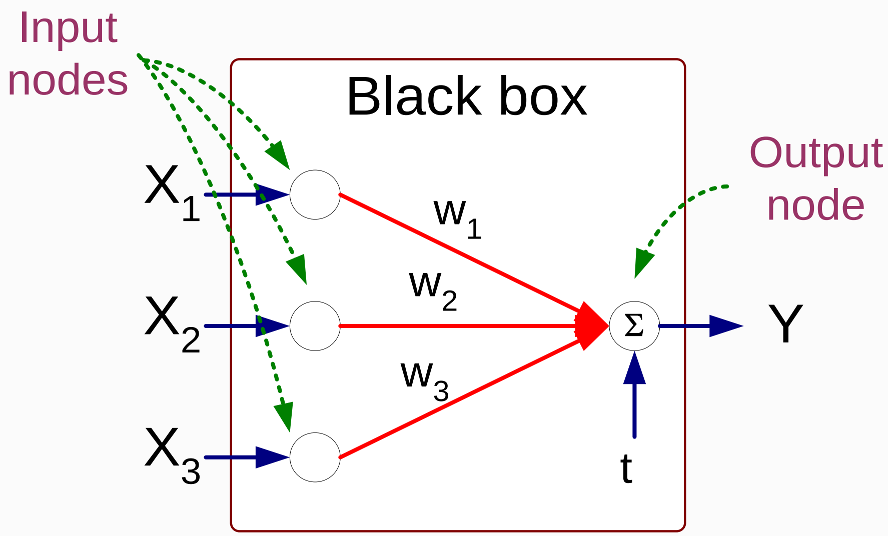


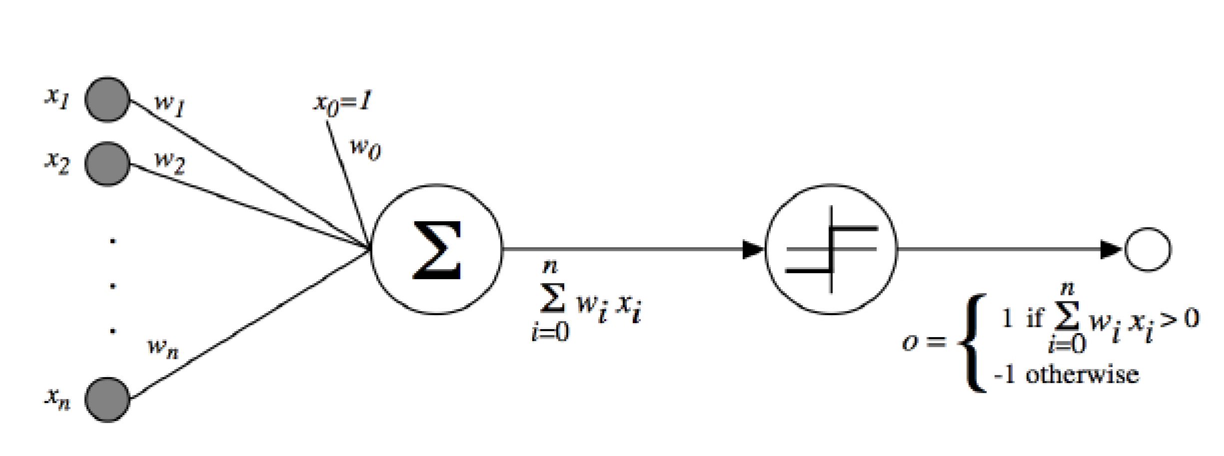
$$Y = I(0.3X_1 + 0.3X_2 + 0.3X_3 - 0.4 > 0)$$

$$\text{where } I(z) = \begin{cases} 1 & \text{if } z \text{ is true} \\ 0 & \text{otherwise} \end{cases}$$

Artificial Neural Networks (ANN)

- Model is an assembly of inter-connected nodes and weighted links;
- Output node sums up each of its input value according to the weights of its links;
- Compare output node against some threshold t





Perceptron Model

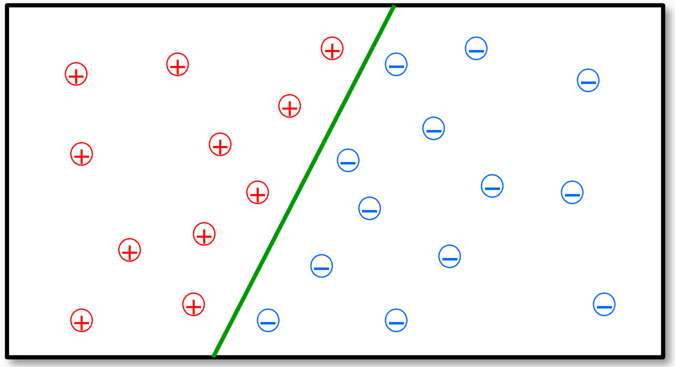
$$Y = I\left(\sum_i w_i X_i - t\right) \quad \text{or}$$

$$Y = sign\left(\sum_i w_i X_i - t\right)$$

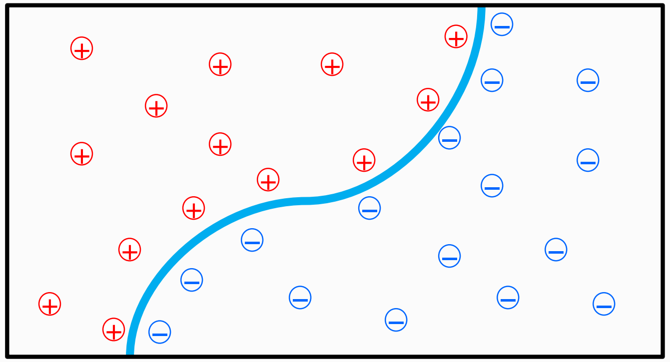
First Generation Neural Networks: The Perceptron

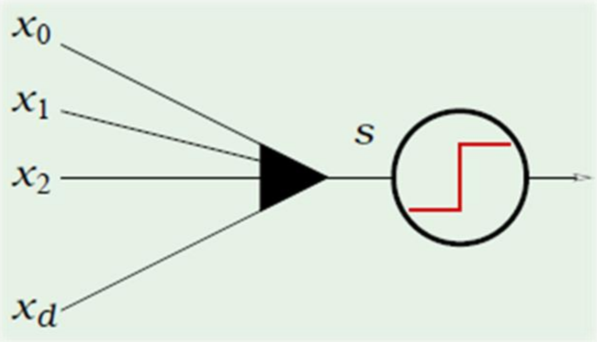
- One of the earliest ML algorithms (Rosenblatt 1958).
- Online linear binary classification algorithm.
- Determines a hyperplane (line in $\mathbb{R}^2$, plane in $\mathbb{R}^3$,...) separating the points for the two classes.

Linearly separable data:

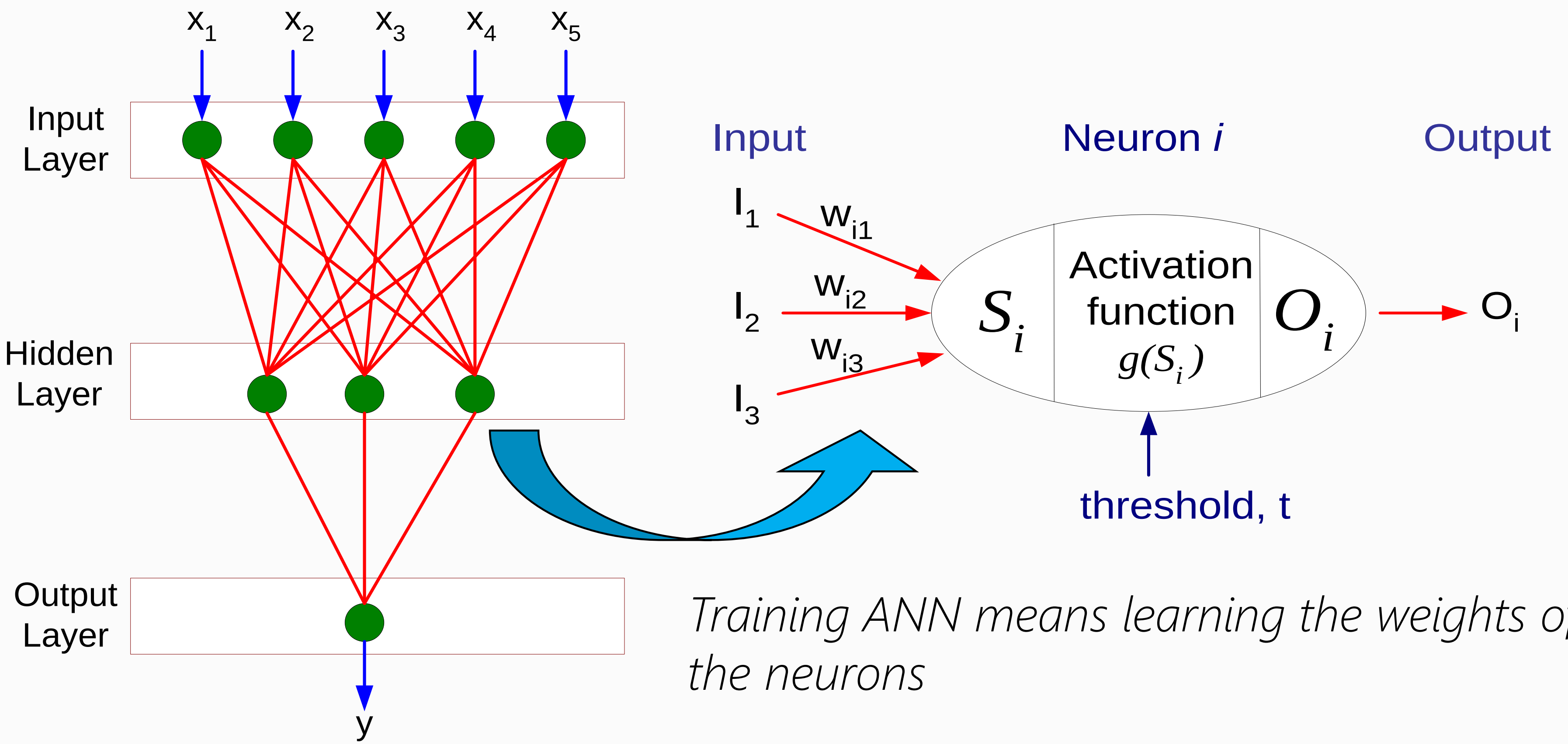


Non-linearly separable data:





General Structure of ANN

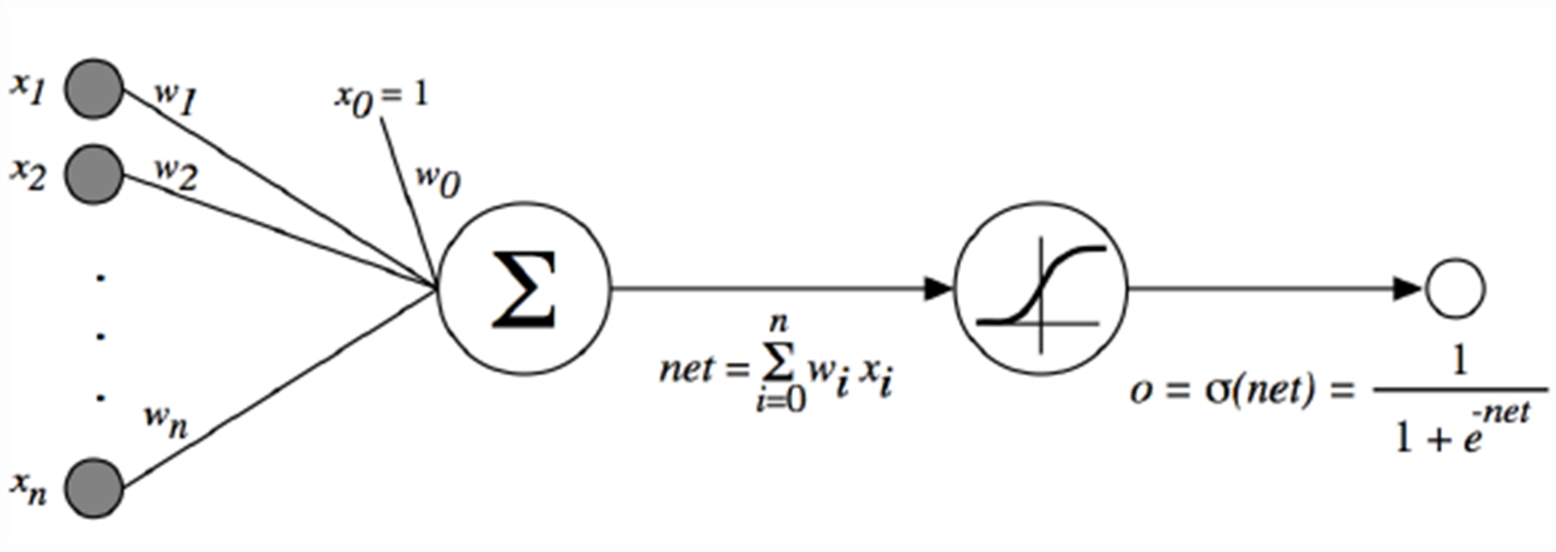


Training ANN means learning the weights of the neurons

- Perceptrons only have two layers: the input layer and the output layer
- Perceptrons only have one output unit;
- Multi-layer ANNs consist of an input layer, hidden layer, and output layer
- Multi-layer neural networks can have several output units.

Second Generation Neural Networks: Multi-Layer Perceptron

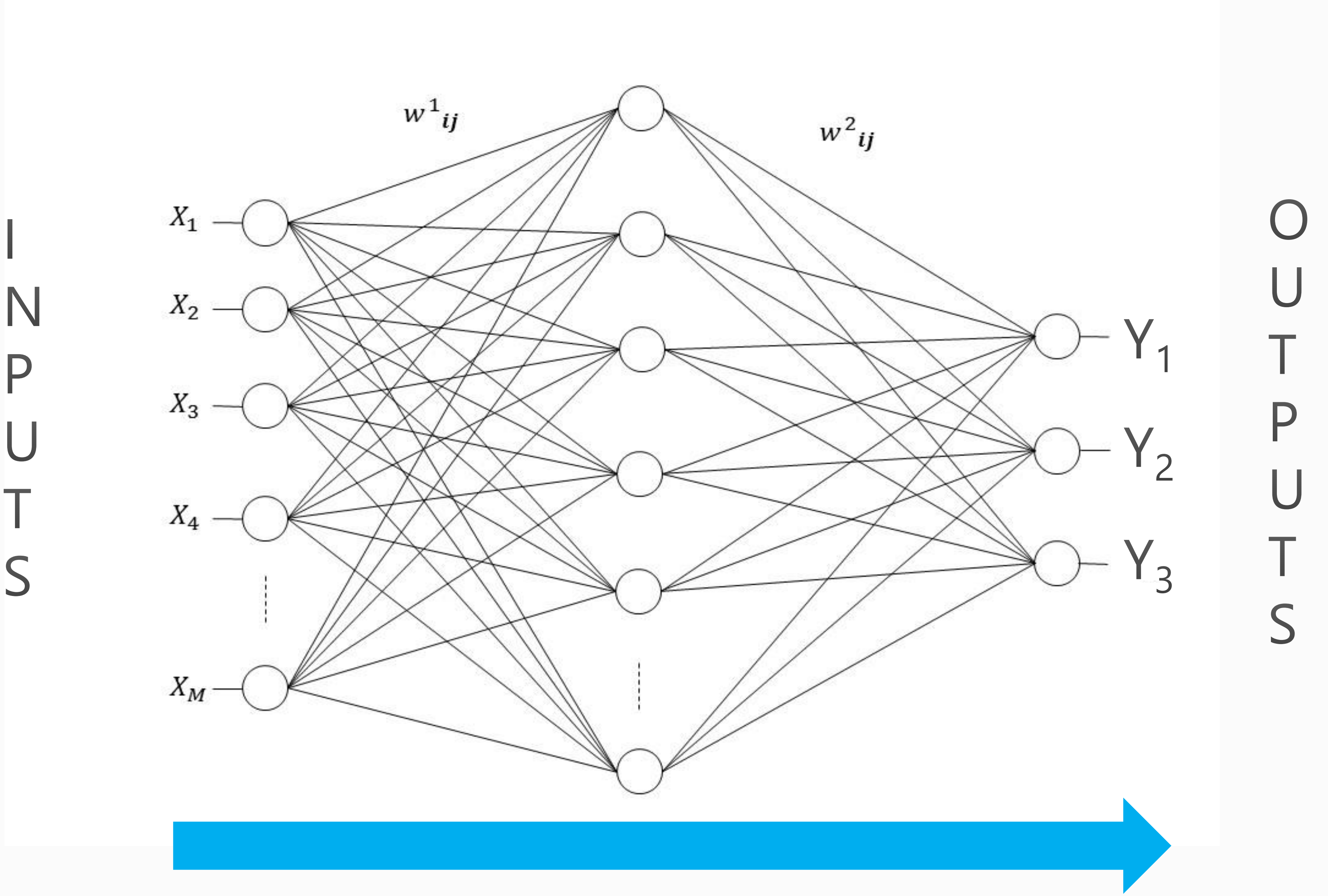
- The units of the hidden layer function as input units to the next layer
- However, each layer of linear units still produce only linear functions
- The step function in perceptrons is another choice, but it is not differentiable, and therefore not suitable for gradient descent search
- Solution: the sigmoid function, a non-linear, differentiable threshold function



How A Multi-Layer Neural Network Works

- The inputs to the network correspond to the attributes measured for each training tuple
- Inputs are fed simultaneously into the units making up the input layer
- They are then weighted and fed simultaneously to a hidden layer
- The number of hidden layers is arbitrary, although usually only one
- The weighted outputs of the last hidden layer are input to units making up the output layer, which emits the network's prediction
- The network is feed-forward: None of the weights cycles back to an input unit or to an output unit of a previous layer
- From a statistical point of view, networks perform nonlinear regression: Given enough hidden units and enough training samples, they can closely approximate any function

Feedforward Phase of Learning



Define the Network Topology

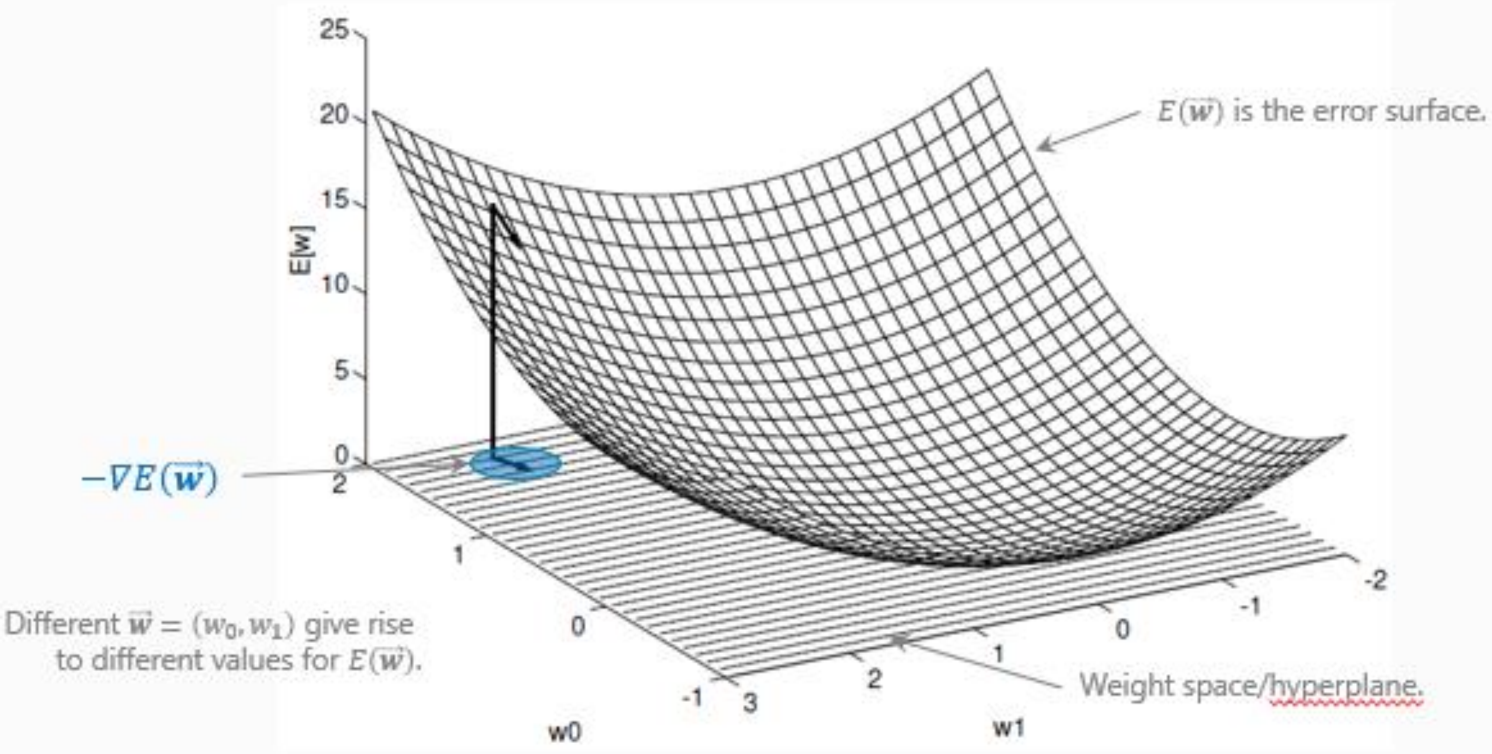
- Decide the network topology: Specify # of units in the *input layer*, *# of hidden layers* (if > 1), # of units in *each hidden layer*, and # of units in *output layer*
- Normalize the input values for each attribute measured in the training tuples to [0.0 - 1.0]
- One input unit per domain value, each initialized to 0
- Output, if for classification and more than two classes, one output unit per class is used
- Once a network has been trained and its accuracy is unacceptable, repeat the training process with a *different network topology* or a *different set of initial weights*

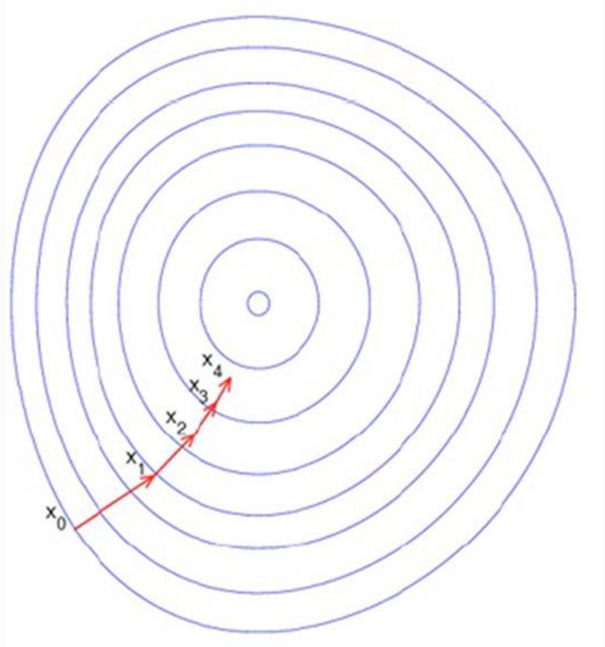
Math Fact

The gradient of the error: $$\nabla E(\vec{\boldsymbol{w}}) = \left[\frac{\partial E}{\partial w_0}, \frac{\partial E}{\partial w_1}, \dots, \frac{\partial E}{\partial w_d}\right]$$

(a vector in weight space) specifies the direction of the argument that leads to the steepest increase for the value of the error.

The negative of the gradient gives the direction of the <u>steepest decrease</u>.





Algorithm for Learning ANN

- Initialize the weights (w_0, w_1, …, w_k)

- Adjust the weights in such a way that the output of ANN is consistent with class labels of training examples
  - Objective function:

$$E = \sum_i \left[Y_i - f(w_i, X_i)\right]^2$$

  - Find the weights w_i's that minimize the above objective function
    - e.g., backpropagation algorithm

Backpropagation

- Iteratively process a set of training examples & compare the network's prediction with the actual known target value
- For each training example, the weights are modified to minimize the mean squared error between the network's prediction and the actual target value
- Modifications are made in the "backwards" direction: from the output layer, through each hidden layer down to the first hidden layer, hence "backpropagation"
- Steps
  - Initialize weights to small random numbers, associated with biases
  - Propagate the inputs forward (by applying activation function)
  - Backpropagate the error (by updating weights and biases)
  - Terminating condition (when error is very small, etc.)

Artificial Neural Networks

Known samples (historical data) are used to “train” the network.

Input data (x_i) are assigned weights (w_i) and combined in the “hidden” layer – like a set of linear regressions. These sets can then combined in additional layers – like regressions of regressions.

The sum of data and weights are transformed (“squashed”) to the range of the training data and error is measured.

A supervised training algorithm uses output error to adjust network weights to minimize errors.

Artificial Neural Networks

Stochastic Gradient Descent

- Converging to a minimum can be quite slow (i.e. it can take thousands of steps). Increasing the learning rate can improve this but can lead to overstepping minima
- If there are multiple local minima in the error surface, gradient descent can get stuck in one of them and not find the global minimum
- Gradient descent updates weights after summing over all training examples
- Stochastic gradient descent alleviates these difficulties
  - Randomly shuffle the data set;
  - Run small batch of training samples through the algorithm;
  - Stochastic (or incremental) gradient descent updates weights incrementally after calculating error for batch;
  - Continue until end of training set is reached
  - Related to Noise, Jitter, Simulated Annealing, etc.

Practical Considerations

- A good Backpropagation net requires more than the core of the learning algorithms. Many parameters must be carefully selected to ensure a good performance.
- Although the deficiencies of Backpropagation nets cannot be completely cured, some of them can be eased by some practical means.
- Initial weights (and biases)
  - Random, [-0.05, 0.05], [-0.1, 0.1], [-1, 1]
  - Normalize weights for hidden layer (v_ij) (Nguyen-Widrow)
    - Randomly assign v_ij for all hidden units V_j
    - For each V_j, normalize its weight by $\boldsymbol{v}_{ij} = \boldsymbol{\beta} \cdot \boldsymbol{v}_{ij} / \|\boldsymbol{v}_{.j}\|_2$ and $\boldsymbol{\beta} = 0.7\sqrt[n]{\boldsymbol{p}}$

      where $\|\boldsymbol{v}_{.j}\|_2$ is the normalization factor

      where $\boldsymbol{p} = \#$ of hiddent nodes, $\boldsymbol{n} = \#$ of input nodes
    - Avoid bias in weight initialization: $\|\boldsymbol{v}_{.j}\|_2 = \boldsymbol{\beta}$ after normalizat ion

Training samples

- Quality and quantity of training samples determines the quality of learning results
- Samples must be <u>good representatives</u> of the problem space
  - Random sampling
  - Proportional sampling (with prior knowledge of the problem space)
- # of training patterns needed:
  - There is no theoretically ideal number. Following is a rule of thumb
  - W: total # of weights to be trained (depends on net structure)
    e: desired classification error rate
  - If we have P = W/e training patterns, and we can train a net to correctly classify (1 – e/2)P of them,
  - Then this net would (in a statistical sense) be able to correctly classify a fraction of 1 – e input patterns drawn from the same sample space
  - Example: W = 80, e = 0.1, P = 800. If we can successfully train the network to correctly classify (1 – 0.1/2)*800 = 760 of the samples, we would believe that the net will work correctly 90% of time with other input.

Over-Training/Over-Fitting

- Trained net fits very well with the training samples (total error $E \approx 0$), but not with new input patterns
- Over-training may become serious if
  - Training samples were not obtained properly
  - Training samples have noise

Control over-training for better generalization
- Either divide the samples into two sets
    - \- 90% into training set: used to train the network
    - \- 10% into test set: used to validate training results

  periodically test the trained net with test samples, stop training when test results start to deteriorating, or
  - Use Cross-validation

- Stop training early (before $E \approx 0$)
- Add noise to training samples: x:t becomes x+noise:t

Hyperparameters

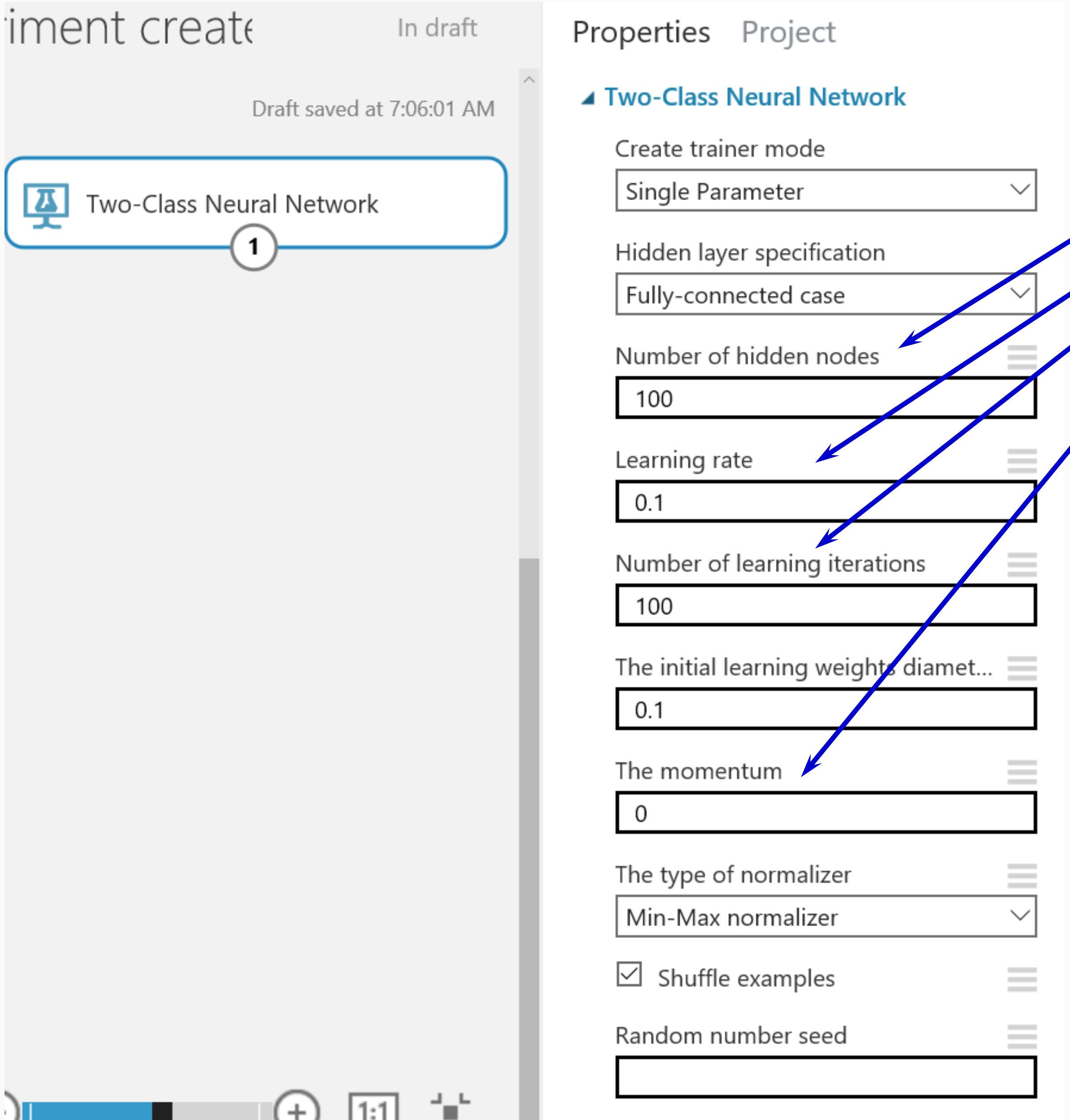
Gradient descent

- Initial weights
- Learning rate schedule
- Batch size
- Momentum
- Stopping criteria

Tuning

- Split data set into training, cross-validation (cv), and test
- Fit model on training set
- Tune hyperparameters on Cross-Validation set
- Evaluate on test set

Neural Networks in Azure ML



Azure ML provides user control of training parameters:

- # of hidden nodes
- Learning rate
- # of iterations or epochs (“training time”)
- Controls on varying changes to increments (“momentum”) and weight decay

Some problems with backpropagation

The amount of information that each training case provides about the weights is at most the log of the number of possible output labels.

- So to train a big net we need lots of labeled data.

In nets with many layers of weights the backpropagated derivatives either grow or shrink multiplicatively at each layer.

- Learning is tricky either way.

Dumb gradient descent is not a good way to perform a global search for a good region of a very large, very non-linear space.

- So deep nets trained by backpropagation are rare in practice.

A solution to all of these problems

- Use greedy unsupervised learning to find a sensible set of weights one layer at a time. Then fine-tune with backpropagation.
- Greedily learning one layer at a time scales well to really deep networks.
- Most of the information in the final weights comes from modeling the distribution of input vectors.
  - The precious information in the labels is only used for the final fine-tuning.
- We do not start backpropagation until we already have sensible weights that already do well at the task.
  - So the fine-tuning is well-behaved and quite fast.

Neural Network as a Classifier

Strength

- High tolerance to noisy data
- Ability to classify untrained patterns
- Well-suited for continuous-valued inputs *and outputs*
- Successful on an array of real-world data, e.g., hand-written letters
- Algorithms are inherently parallel
- Techniques have been developed for the extraction of rules from trained neural networks

Weakness

- Long training time
- Require a number of parameters typically best determined empirically, e.g., the network topology or "structure."
- Poor interpretability: Difficult to interpret the symbolic meaning behind the learned weights and of "hidden units" in the network

Neural Networks - Example

Credits: Based on the example by Frauke Günther and Stefan Fritsch

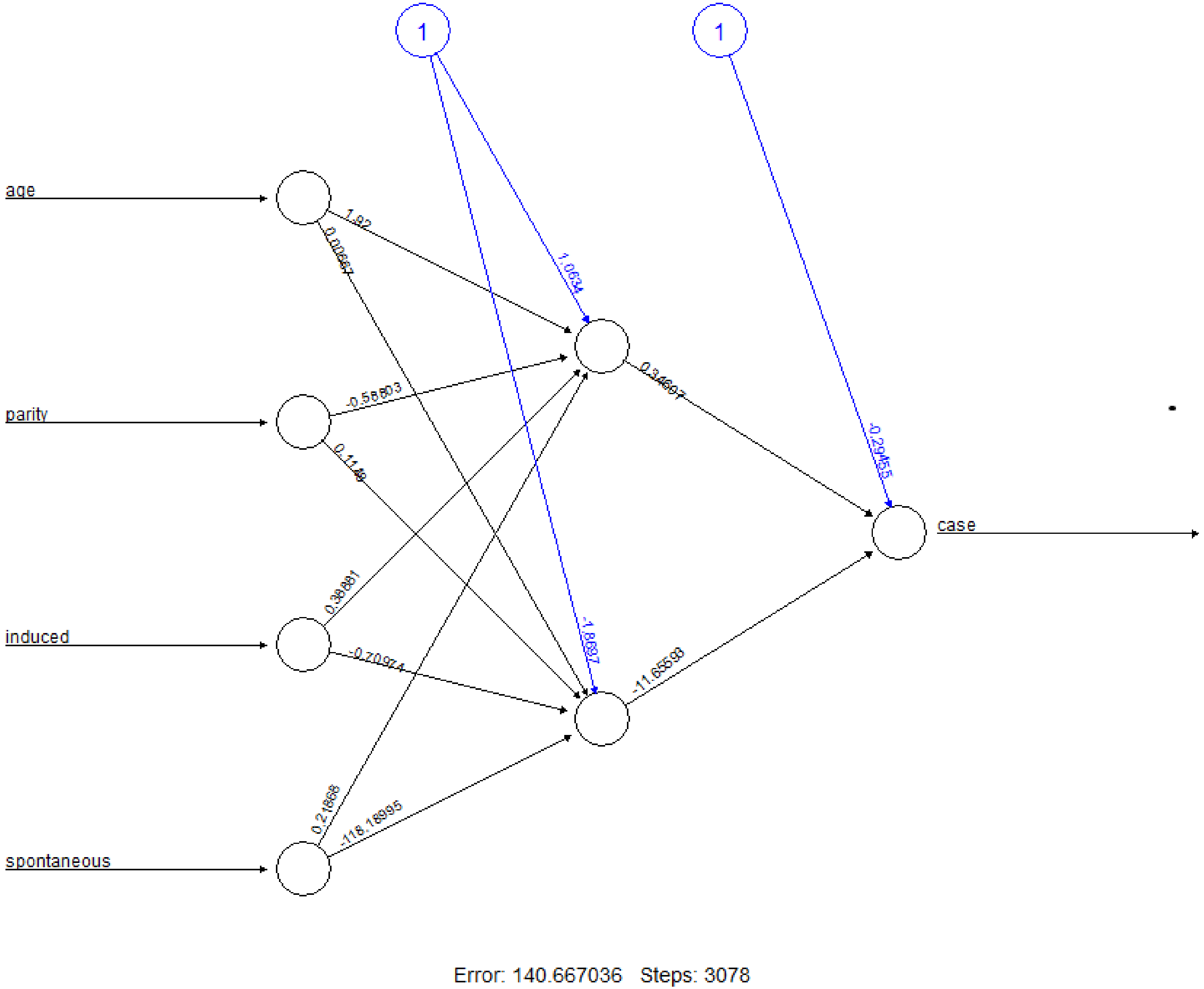
Data

```
> head(infert)
  education age parity induced case spontaneous stratum pooled.stratum
1    0-5yrs  26      6       1    1           2       1              3
2    0-5yrs  42      1       1    1           0       2              1
3    0-5yrs  39      6       2    1           0       3              4
4    0-5yrs  34      4       2    1           0       4              2
5   6-11yrs  35      3       1    1           1       5             32
6   6-11yrs  36      4       2    1           1       6             36
```

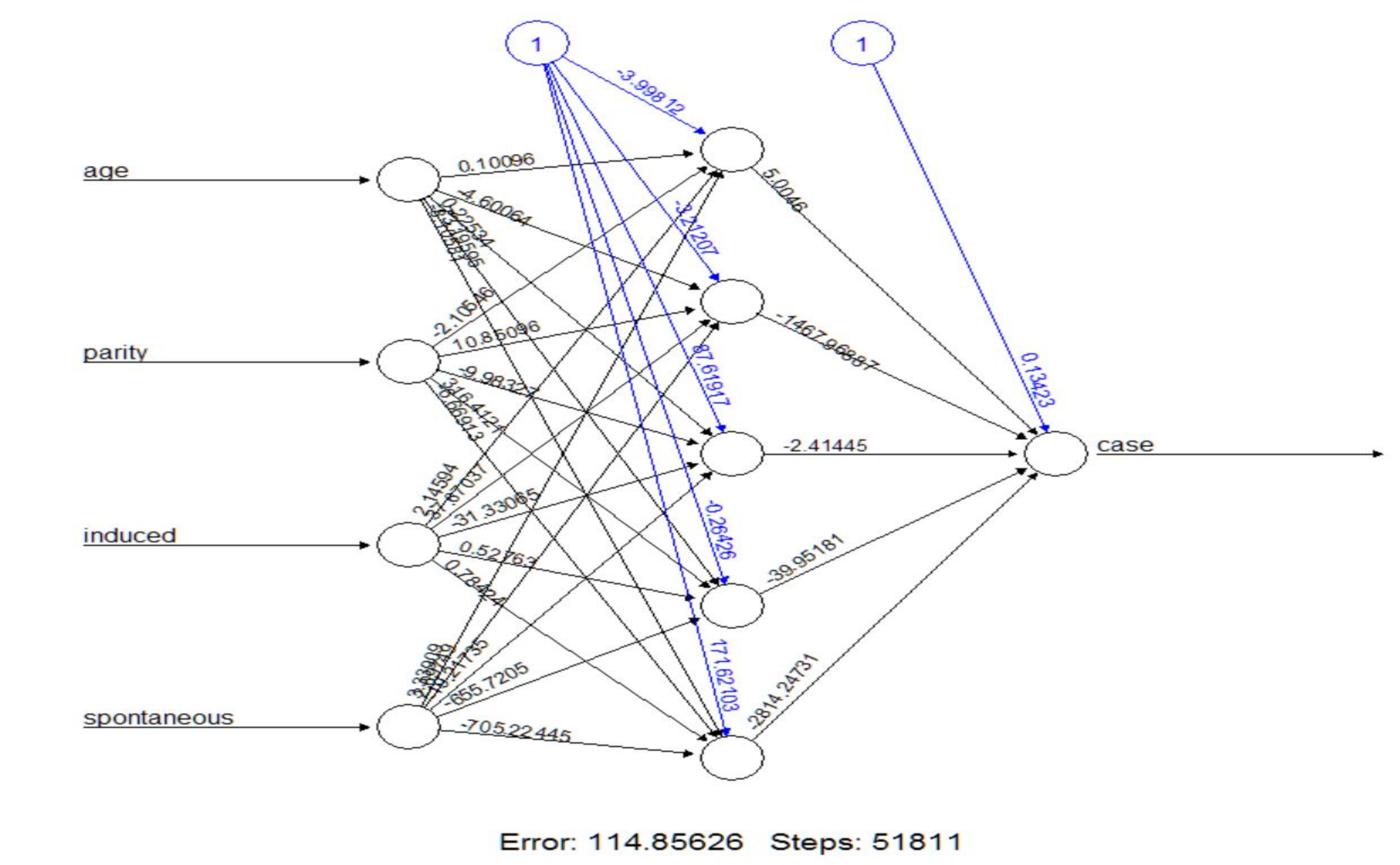
Infert - Contains data of a case-control study that investigated infertility after spontaneous and induced abortion (Trichopoulos et al., 1976).

248 observations, 83 women, who were infertile (cases), and 165 women, who were not infertile (controls).

```r
library(neuralnet)

head(infert)

## Train a Neural network with two hidden neurons
nn <- neuralnet(
  case~age+parity+induced+spontaneous,
  data=infert, hidden=2, err.fct="ce",linear.output=FALSE)

##nn
##Visualize the neual network
plot(nn)


nn$result.matrix

covaresult <- cbind(nn$covariate,nn$net.result[[1]])
dimnames(covaresult) <- list(NULL,c("age","parity","induced","spontaneous","nn-output"))
head(covaresult)
```


1
1
age
parity
induced
spontaneous
case
1.92
1.0634
-0.58603
0.38881
-0.70974
-1.8697
0.21868
-118.18995
-0.29455
Error: 140.667036 Steps: 3078

- Now change "hidden" parameter to 5
- What do you observe in the performance and results?



SVM vs. Neural Network

SVM

- Deterministic algorithm
- Nice generalization properties
- Hard to learn – learned in batch mode using quadratic programming techniques
- Using kernels one can learn very complex functions (hypothesis space)

Neural Network

- Nondeterministic algorithm
- Generalizes well but doesn't have strong mathematical foundation
- Can easily be learned in incremental fashion
- To learn complex functions—use multilayer perceptron (nontrivial)

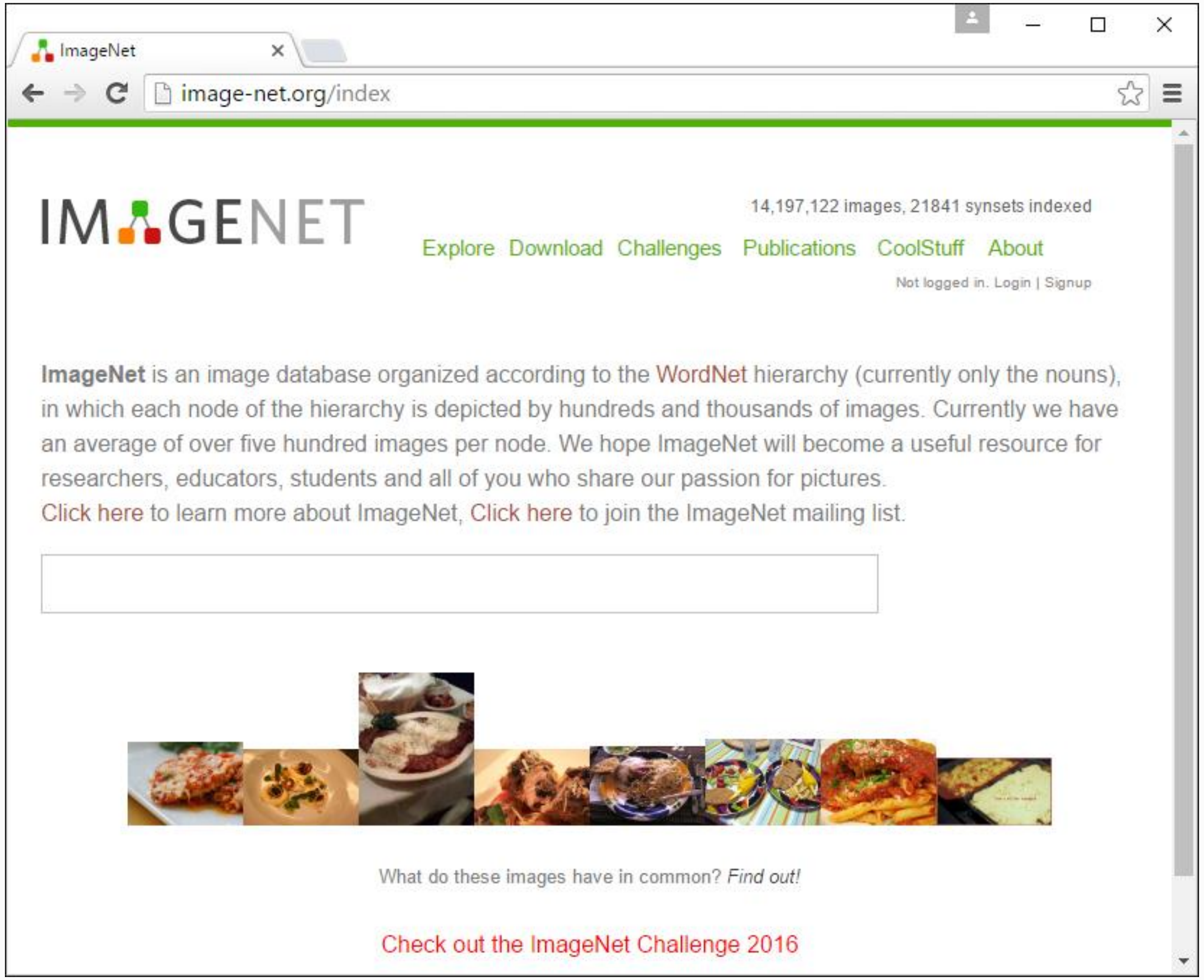
Third Generation Neural Networks: Deep Neural Networks

Goal

Credits:Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton, ImageNet Classification with Deep Convolutional Neural Networks

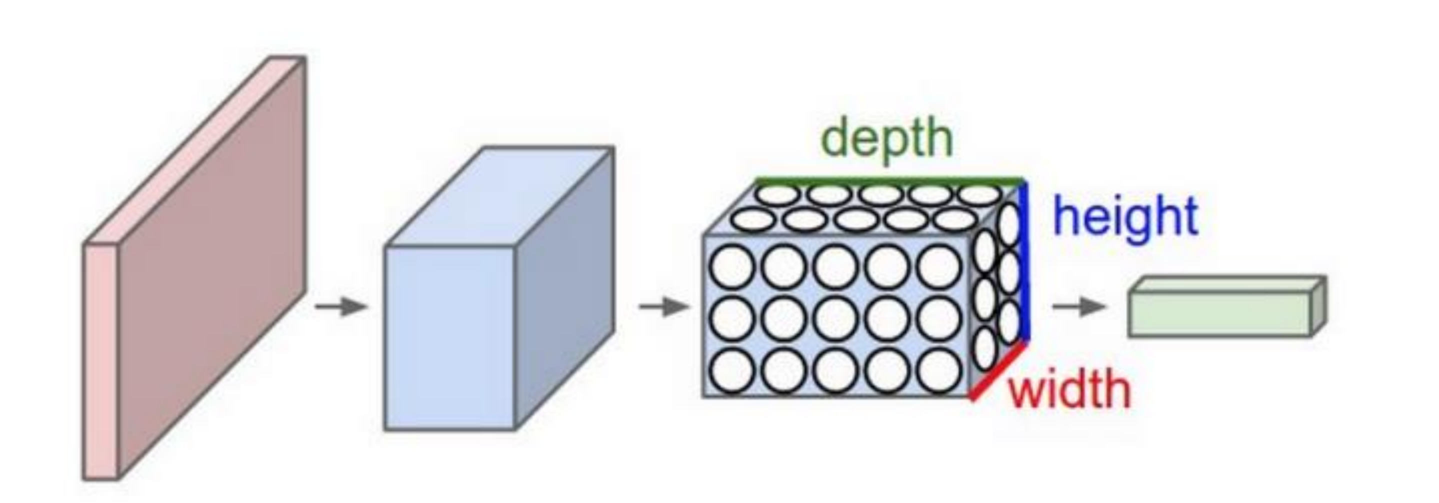
ImageNet

- 15M High-resolution images (labeled)
- 22K Categories
- Web images, labeled using Amazon Mechanical Turk

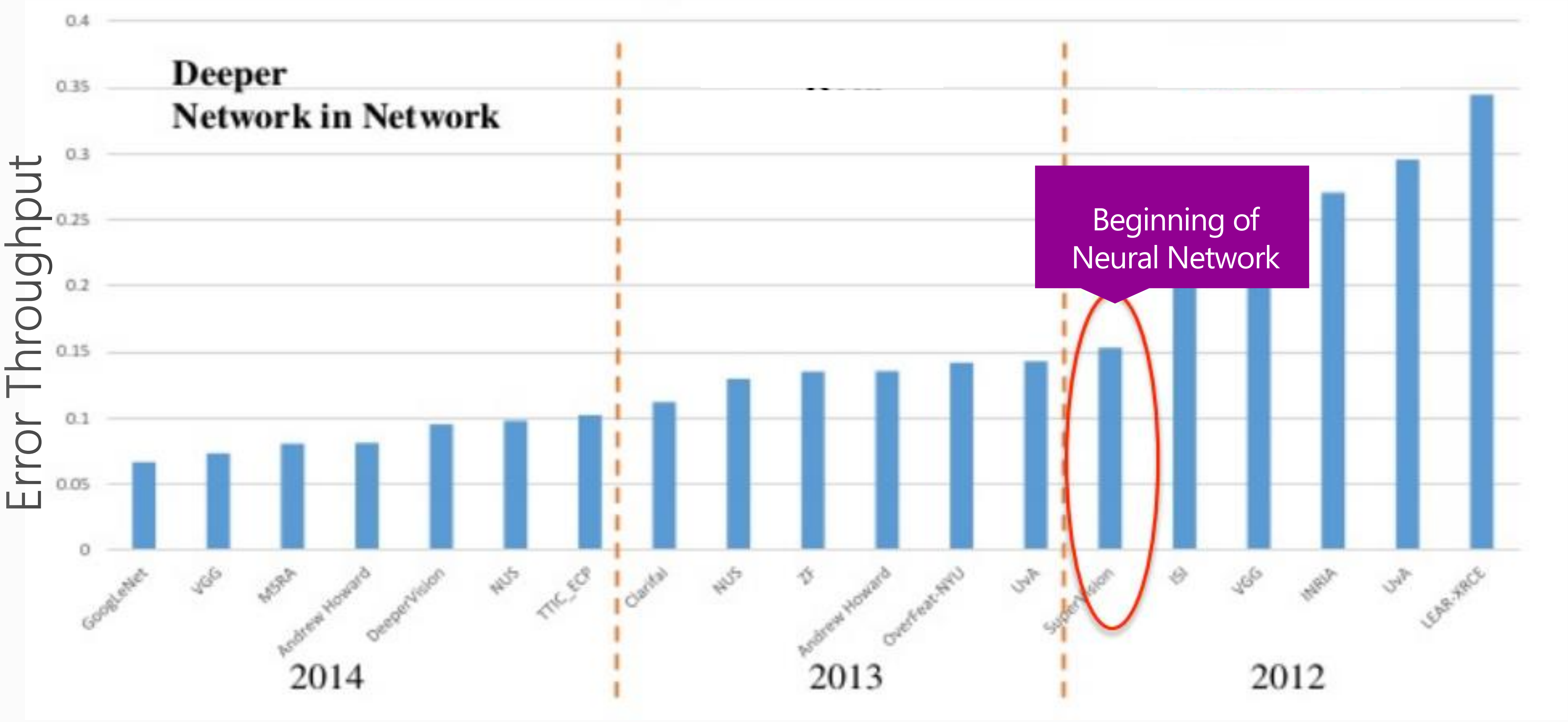


Convolutional Neural Networks

- Large Learning Capacity Model
- Prior knowledge to compensate for data that are not available



ImageNet – Achievements over the years



Credits: Slides from this point

Based on the ICML Tutorial 2004 by Professor Deng Li, MSR (with permissions to use this)

Inside Microsoft Research

Advancing the state of the art in computing

Microsoft Research | News center | Research areas | About | Contact

DNN Research Improves Bing Voice Search

Inside Microsoft Research 17 Jun 2013 10:00 AM **4**

Like 22

Tweet 17

Posted by Rob Knies

We live in a society obsessed with speed. Whether it's download times on a mobile phone or Usain Bolt's time in the 100 meters, the faster the better. We also live during an era when accuracy has become not just preferable but essential. The technological marvels of the 21st century demand it.

Speed=good. Accuracy=good. Put them together, and you've got a leap forward, such as **recent advancements in Bing Voice Search** for **Windows Phone** that enable customers to get faster, more accurate results

from Data at Scale

Impact of deep learning in speech technology

Microsoft

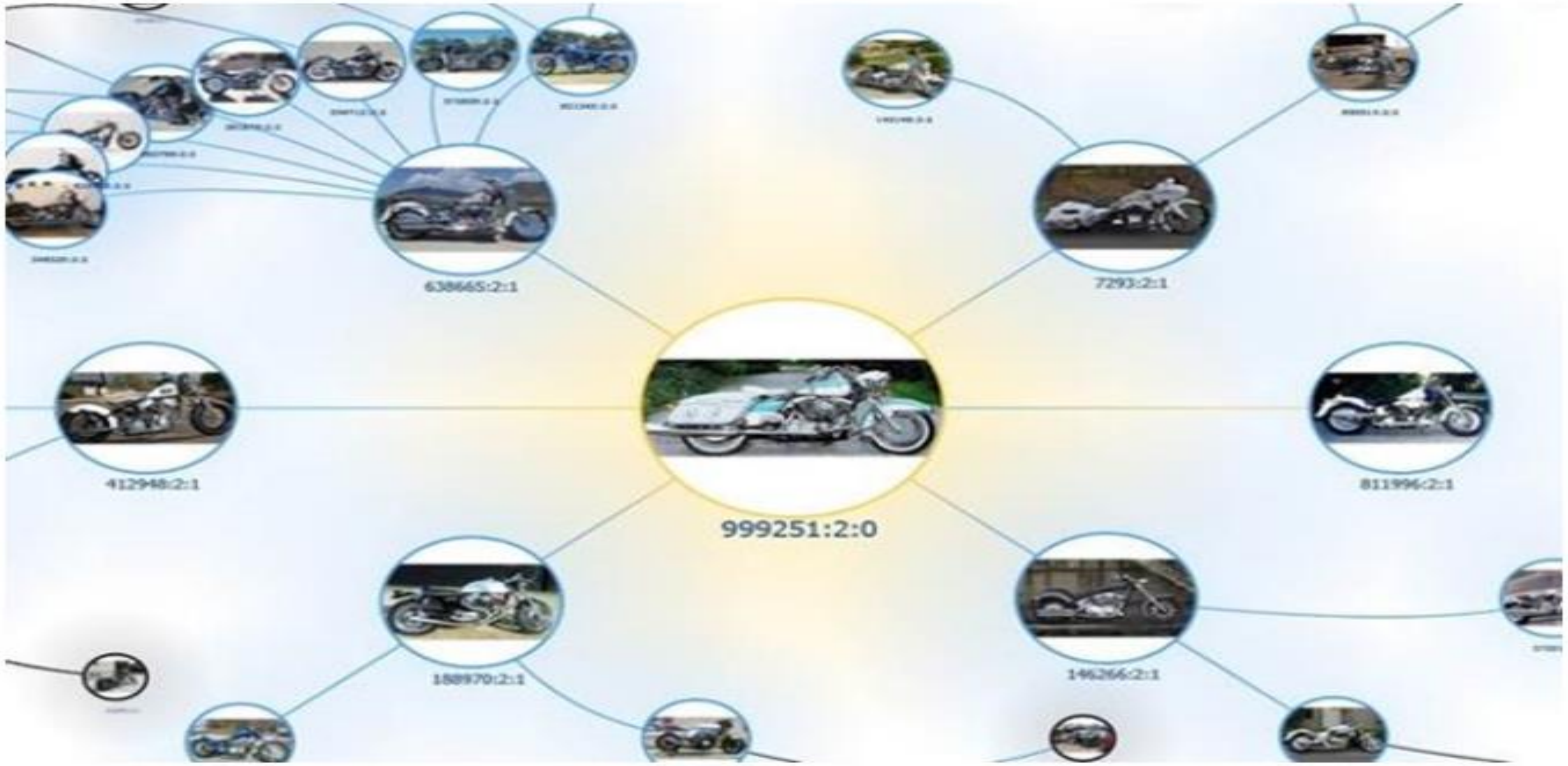
The Fire Hose

Covering the news of the day at Microsoft

TechNet Blogs » The Fire Hose » Bing helps you find better images in your searches with 'deep learning'

Bing helps you find better images in your searches with 'deep learning'

22 Nov 2013 9:21 AM



MIT Technology Review

September 20, 2013

COMPUTING NEWS

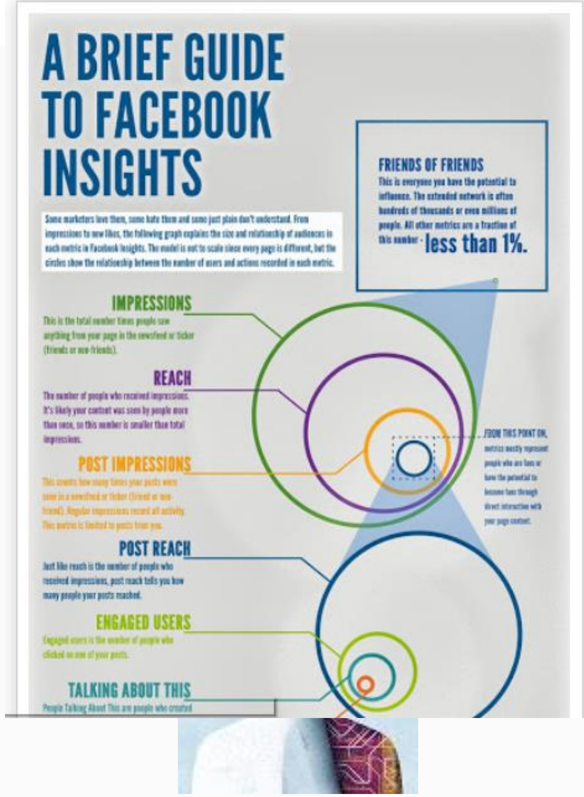
8 COMMENTS

Facebook Launches Advanced AI Effort to Find Meaning in Your Posts

A technique called deep learning could help Facebook understand its users and their data better.

By Tom Simonite on September 20, 2013

……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.



MIT Technology Review

BUSINESS NEWS

8 COMMENTS

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 based in

BloombergBusinessweek
Technology

Acquisitions

The Race to Buy the Human Brains Behind Deep Learning Machines

By Ashlee Vance | January 27, 2014

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."

WIRED GEAR SCIENCE ENTERTAINMENT BUSINESS SECURITY DESIGN OPINION MAGAZINE VIDEO INSIDER SUBSCRIBE

INNOVATION INSIGHTS | community content | featured | blog

Deep Learning's Role in the Age of Robots

BY JULIAN GREEN, JETPAC 05.02.14 2:56 PM

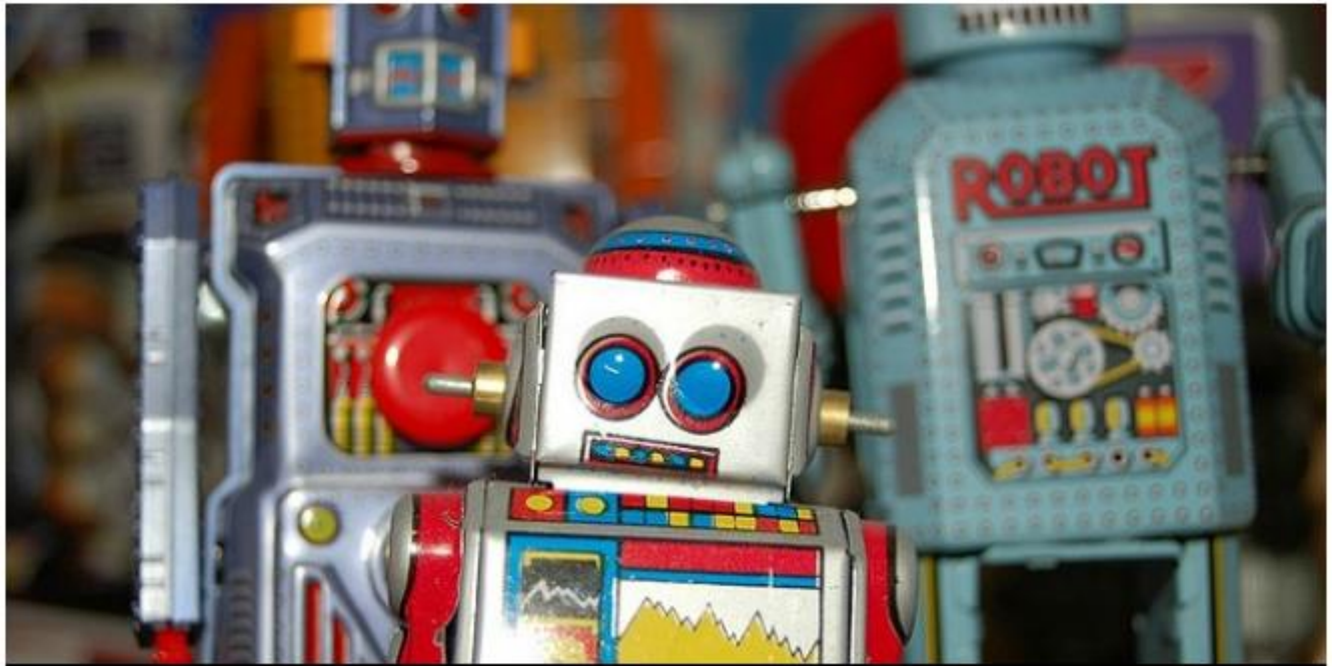


Image: jeffedoe/Flickr

Can robots see as well as humans? That's a question the biggest companies around are trying to answer.

MIT Technology Review

NEWS & ANALYSIS | FEATURES | VIEWS | MULTIMEDIA | DISCUSSIONS | TOPICS | POPULAR: 10 BREAKTHROUGH TECHNOLOGIES

China's Baidu Bets on Deep Learning

MIT Technology Review - 3 days ago

Deep learning makes it possible for machines to process large amounts of date using simulated networks of simple neurons, crudely modeled ...

Baidu snatches Google's **deep-learning** visionary, Andrew ...

VentureBeat - 3 days ago

DARPA is working on its own deep-learning project for natural-language processing

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

2 Comments +1

A▼ A▲

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.*

photo: DARPA

When it comes to large organizations working on artificial intelligence systems for understanding language, there's Google, Microsoft, Yahoo and ... the Defense Advanced Research Projects Agency. The agency, better known as DARPA, is working on a project it calls Deep Exploration and Filtering of Text, or DEFT, in order to analyze textual data at a scale beyond what humans could do by themselves.

GIGAOM EVENTS RESEARCH

Apple Cleantech Cloud Data Europe Media Mobile Podcas

artificial intelligence / machine-learning / open source

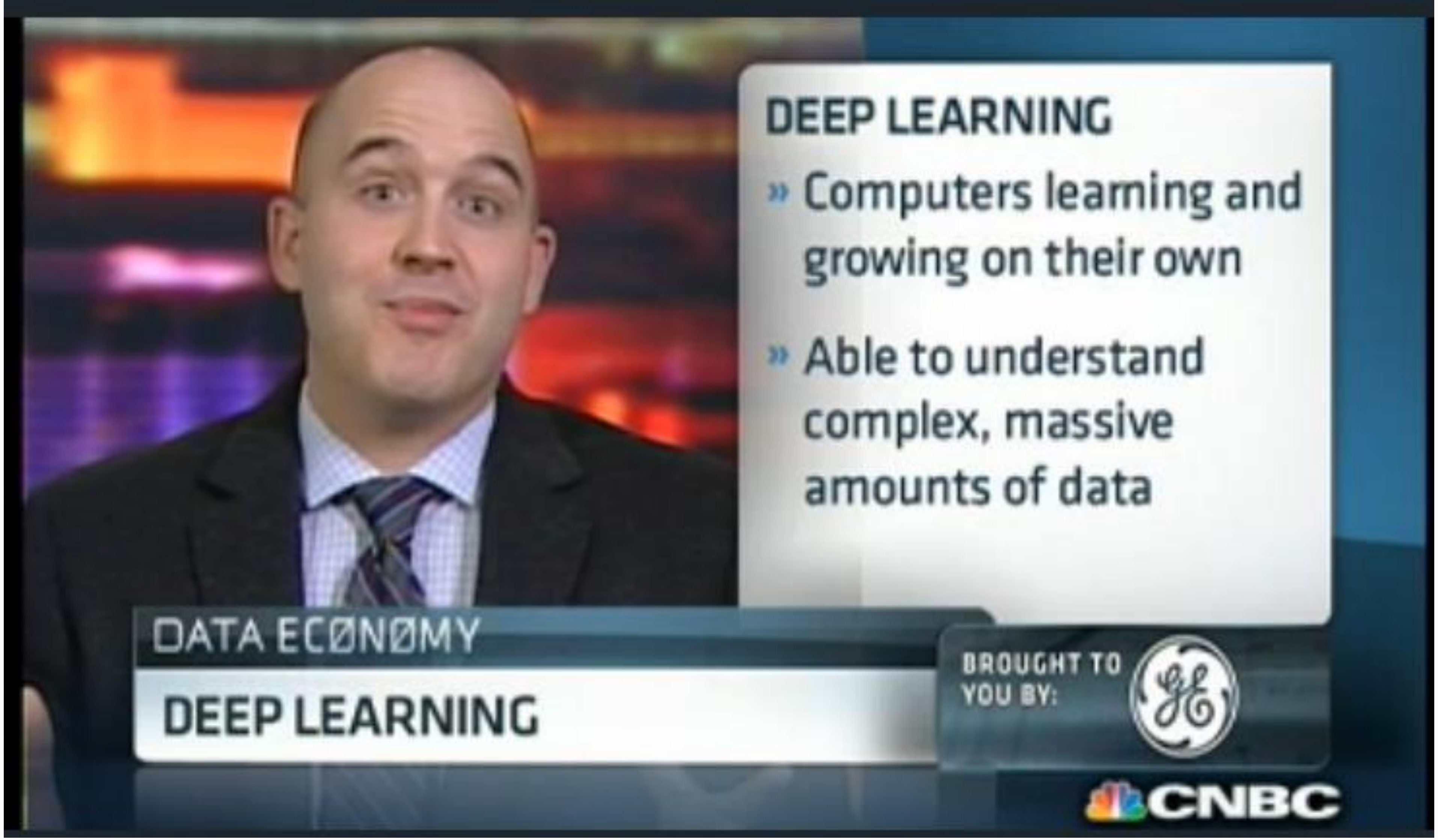
A startup called Skymind launches, pushing open source deep learning

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

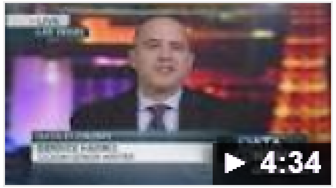
No Comments +1

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

photo: deviantART / rajasegar



[Is **Deep Learning**, the 'holy grail' of big data? - CNBC - Video](video.cnbc.com/gallery/?video=3000192292)

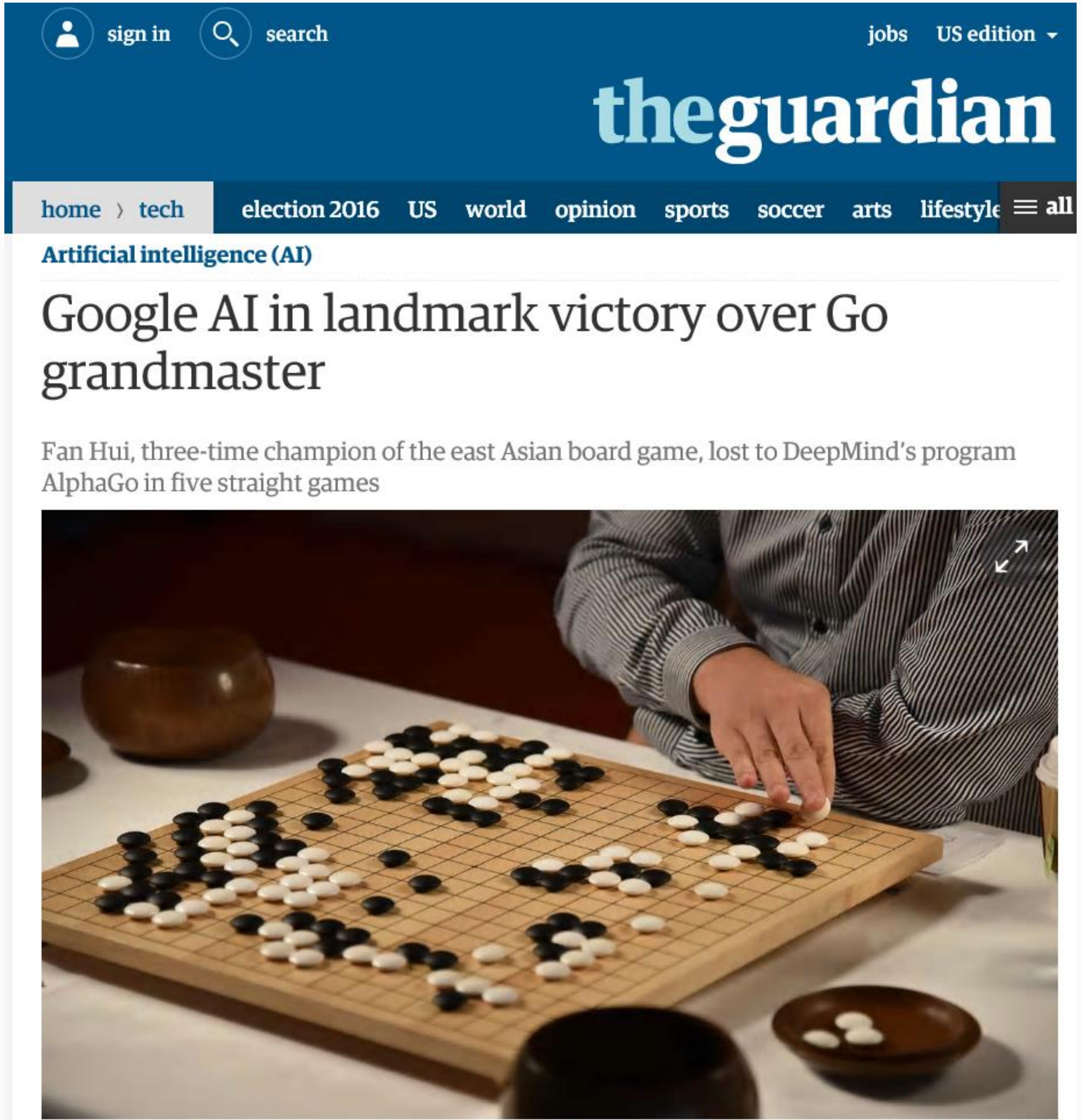


video.cnbc.com/gallery/?video=3000192292

Aug 22, 2013

Derrick Harris, GigaOM, explains how "**Deep Learning**" computers are able to process and understand ...

January 2016: Google's AlphaGo Beats European Go Champion


sign in search jobs US edition

theguardian

home › tech election 2016 US world opinion sports soccer arts lifestyle ≡ all

Artificial intelligence (AI)

Google AI in landmark victory over Go grandmaster

Fan Hui, three-time champion of the east Asian board game, lost to DeepMind's program AlphaGo in five straight games

March 2016: Google’s AlphaGo Beats World Go Champion

Artificial intelligence (AI)

Google's AlphaGo AI defeats human in first game of Go contest

Machine takes 1-0 lead in historic five-game matchup between computer program developed by DeepMind and world's best Go player Lee Sedol

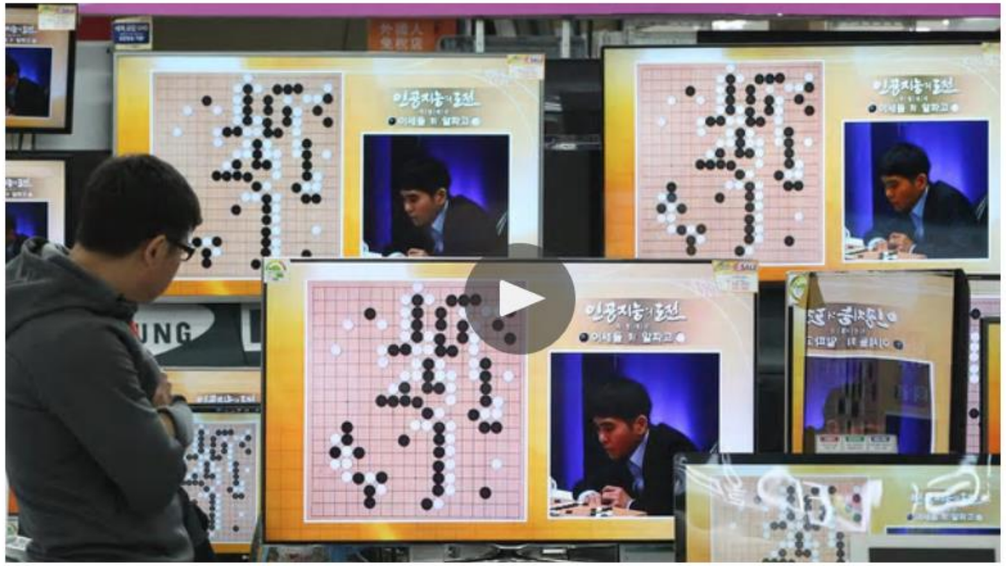
Steven Borowiec in Seoul

Wednesday 9 March 2016 04.14 EST

This article is 2 months old

Shares 6,850 | Comments 392

Save for later



Google's AlphaGo computer wins the first game of Go in a tournament against the world champion, Lee Sedol

Lee Sedol started with a bow, a traditional Korean gesture of respect for an opponent who could neither see him nor sense his presence.

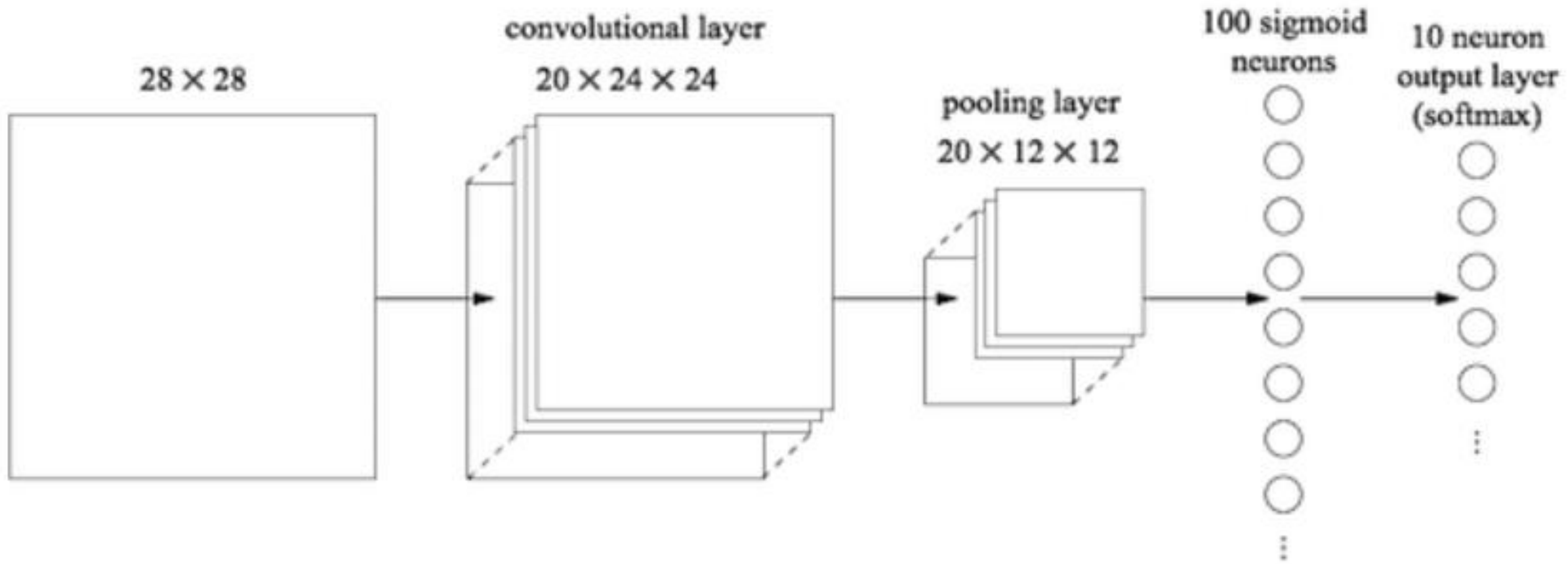
The world champion at Go - an ancient Chinese board game - looked nervous. His eyes darted from side to side. He took a sip of water, and made his first move.

Lee could be forgiven some nerves: his opponent was AlphaGo, an artificial-intelligence program designed by Google DeepMind, their five-game series billed as a landmark face-off between human and computer. “History is really being made here,” said commentator Chris Garlock, as the first game in the series

How Deep Learning Works

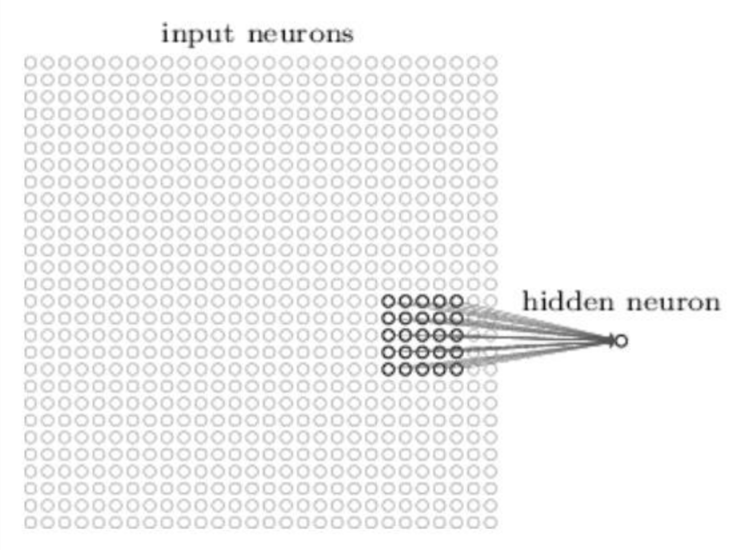
- We know that deep neural networks with many hidden layers are more powerful than shallow networks
- However, deep networks with fully connected nodes are harder to train than shallow networks
- Also, fully connected networks do not capture the spatial structure of images
- Convolutional nets addresses and offers a mode efficient way to train deep neural networks
- Most Deep Neural Networks are based on Convolution

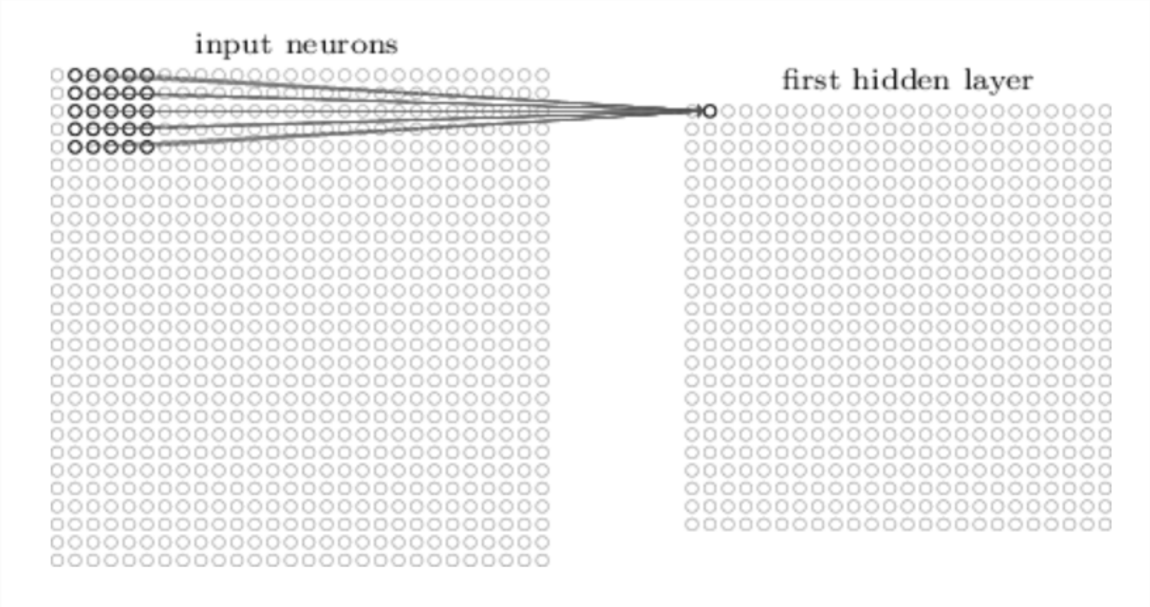
Basic Architecture of a Convolutional Net



- A Convolution net has at least one convolution layer, one pooling layer and at least one fully connected layer
- Convolution layer applies filtering to extract specific features from the raw data
- Pooling layer summarizes the features
- Fully connected layer learns the prediction problem through Backpropagation algorithm.

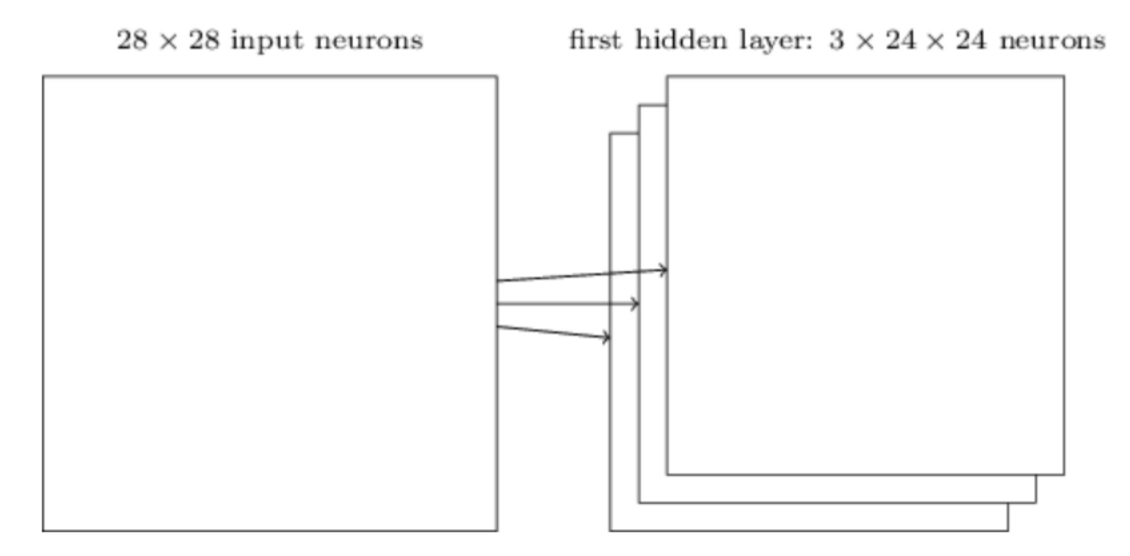
How does Convolution Work?





- The Convolution layer uses a small window to extract local features from the raw data:
  - E.g. it can use a 5X5 pixel window to analyze one small part of a 28X28 image
- Each of the 5X5 input is connected to the hidden node in the convolution layer
- We slide the 5X5 window across the raw data to scan the whole image
- If we slide by 1 pixel the hidden layer will have 24X24 pixels

How does Convolution Work?



- Each of the 24X24 hidden nodes in the Convolution layer shares the same weights and bias
- Each hidden layer in the convolution layer is a ***feature map*** that learns one feature of the image
- The shared 5X5 weights and bias are also called the ***kernel*** or ***filter***
- Assuming a 5X5, kernel, the output of the *j,k*th hidden node is:

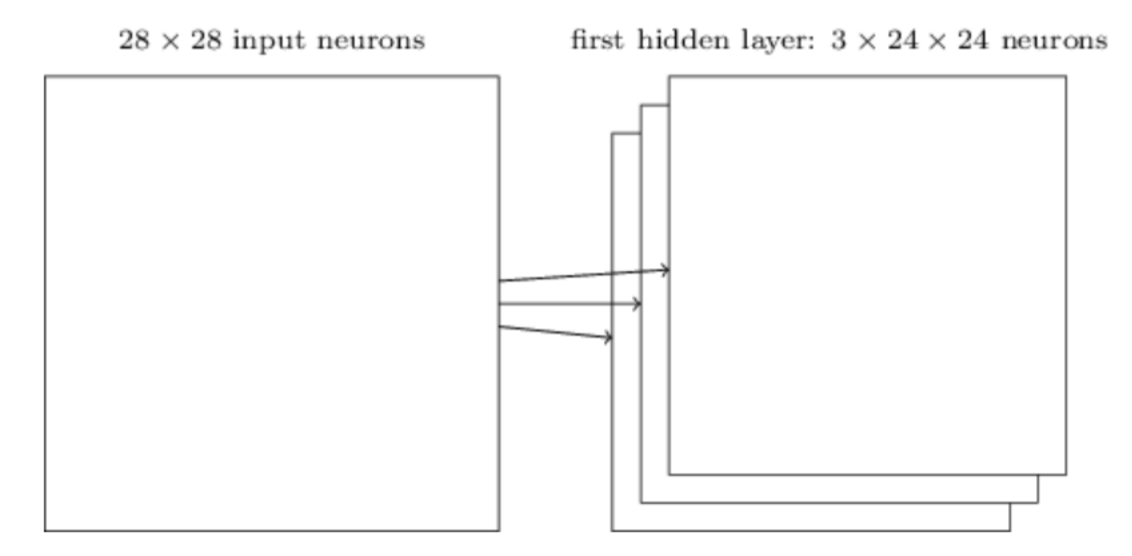
$$\sigma\left(b + \sum_{l=0}^{4} \sum_{m=0}^{4} w_{l,m} a_{j+l,k+m}\right)$$

- Where σ is the activation function (e.g. sigmoidal); $w_{l,m}$ is the 5X5 array of shared weights, and $a_{x,y}$ is the input value at the position *x,y*
- The Convolution layer can have several feature maps e.g. 3 feature maps to learn 3 different features of the image

Why Convolution?

- The name convolution comes from the convolution operation
- We can rewrite the previous equation as

$$a^1 = \sigma(b + w * a^0),$$

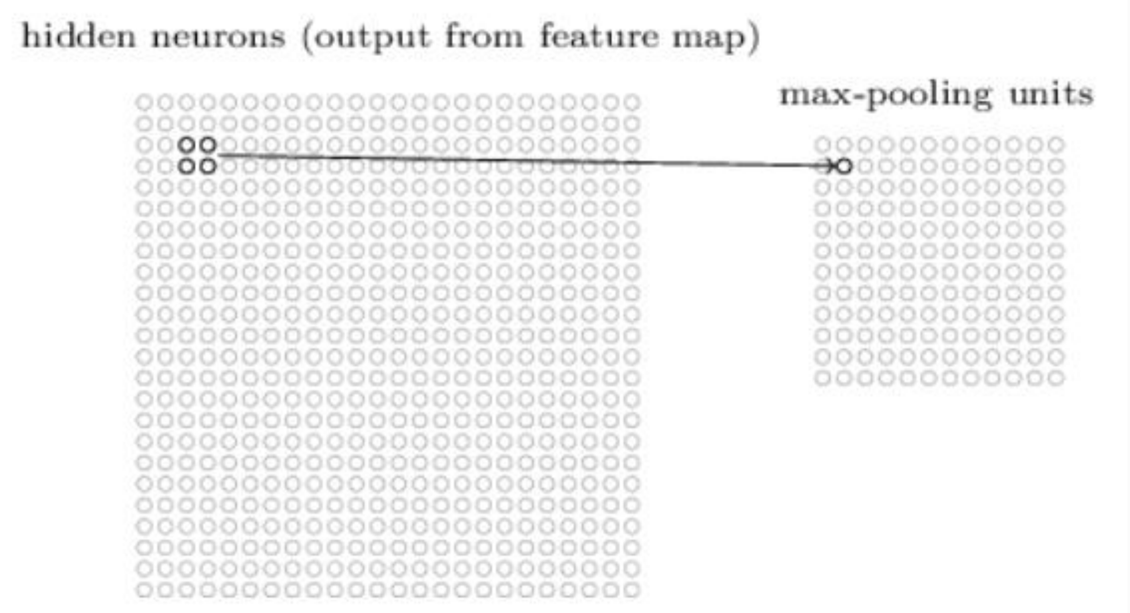


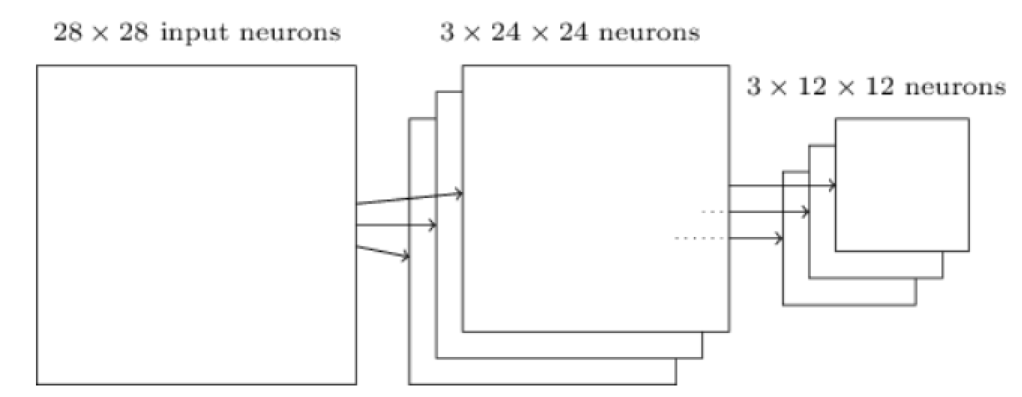
Where:

- a^1 represents the output activations from one feature map,
- a^0 are the inputs
- * is the convolution operation

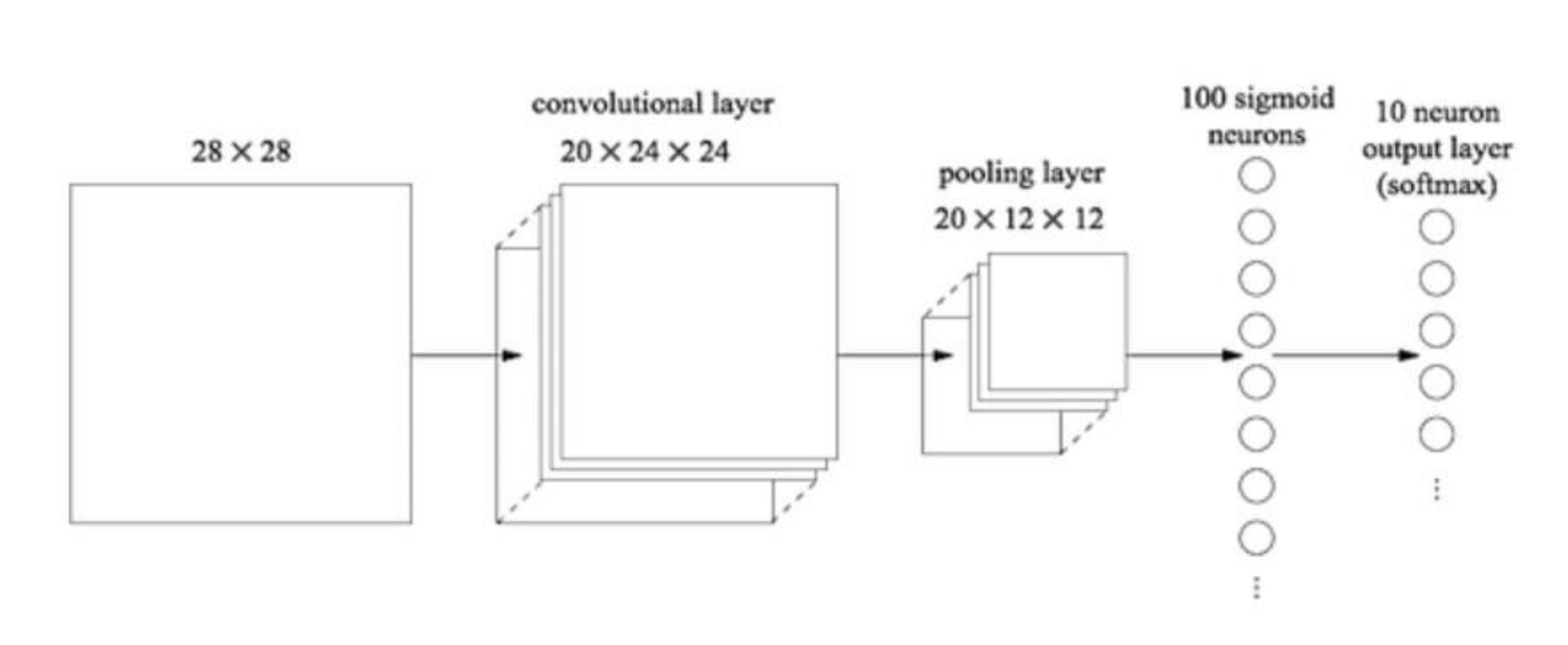
How does Pooling Work?

- The pooling layer appears after the convolution layer
- Each pooling layer compresses the output of the convolution layer
- For example, each pooling node can summarize a 2X2 region in the feature map
- In Max-pooling, each pooling node outputs the maximum of a 2X2 region from the convolution layer
- There is one pooling layer for each feature map in the convolution layer. Hence, there will be 3 pooling layers for 3 feature maps
- Pooling can also use L2 norm or simple average of the 2X2 window



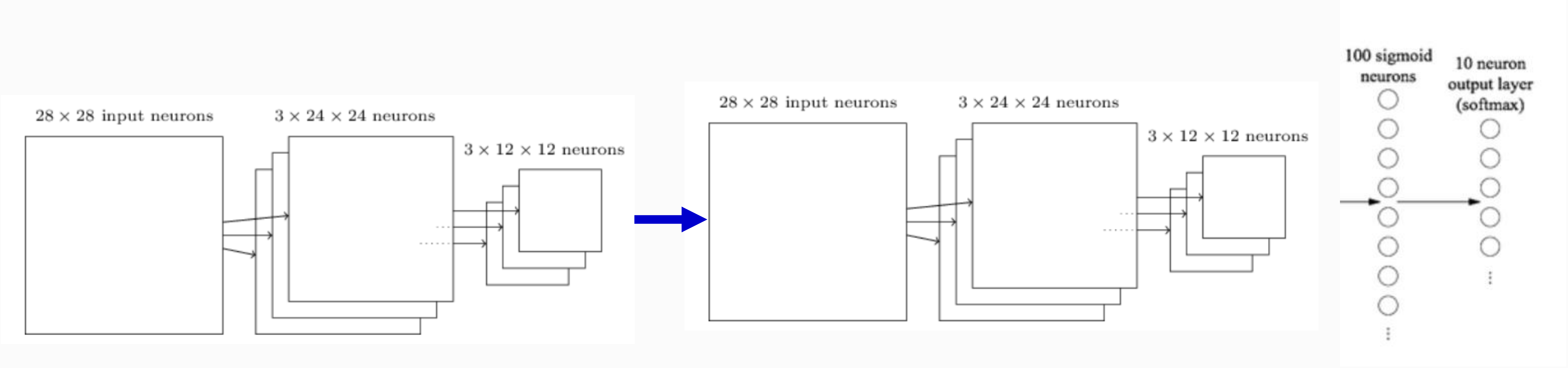


Completing the Convolutional Net



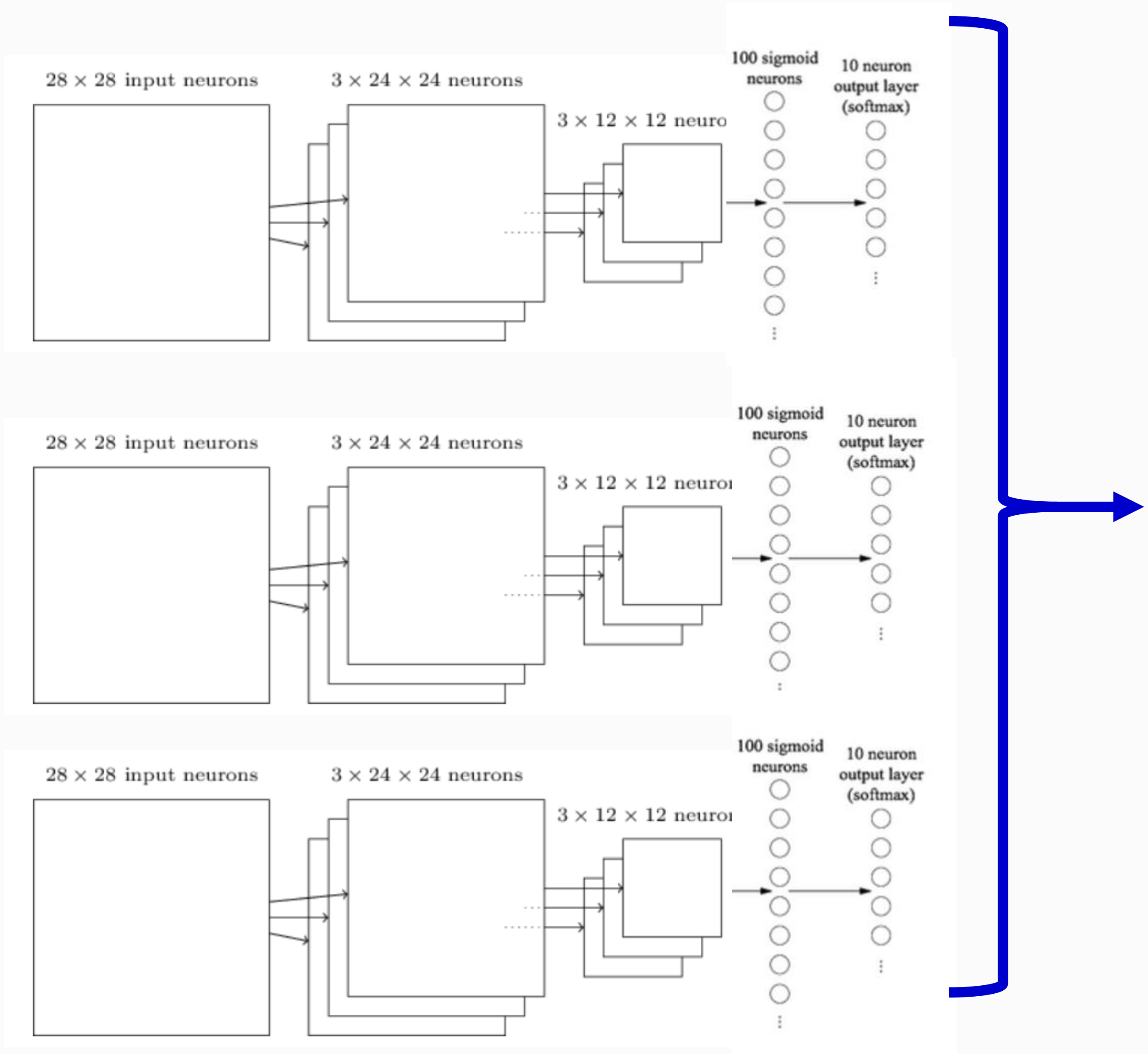
- The final stage of a convolutional net is a fully connected layer
- The inputs to the fully connected net are the outputs of the Max-pooling nodes
- We use a variant of Gradient Descent algorithm to train the fully connected network.

Multiple Convolution and Pooling Layers



- A convolutional net can have multiple convolution and pooling layers feeding each other
- This allows the network to learn more abstract features of the raw data

Ensemble Models with Convolutional Nets



- You can also create an ensemble model using convolutional nets as base learners
- Use majority vote to get the output
- Can perform better than individual convolution nets!

Examples

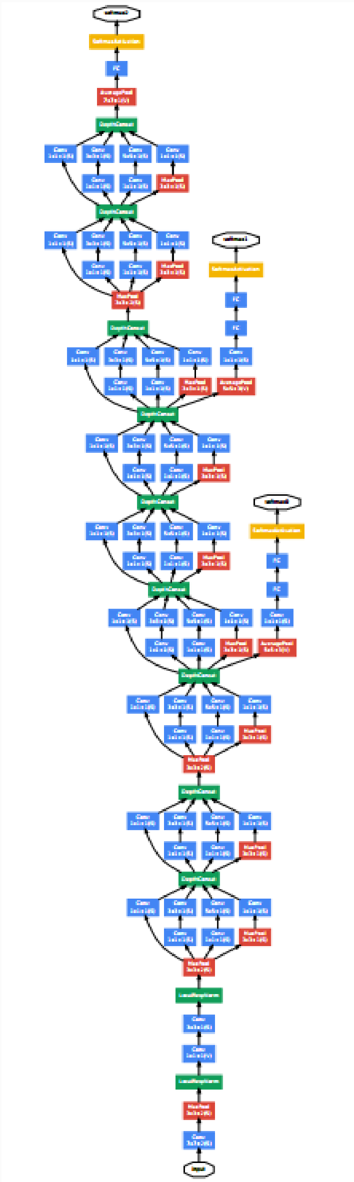
ImageNet Large-Scale Visual Recognition Challenge 2014 (ILSVRC14)

Task: Classifying the image into one of
1000 leaf-node categories in the Imagenet hierarchy.

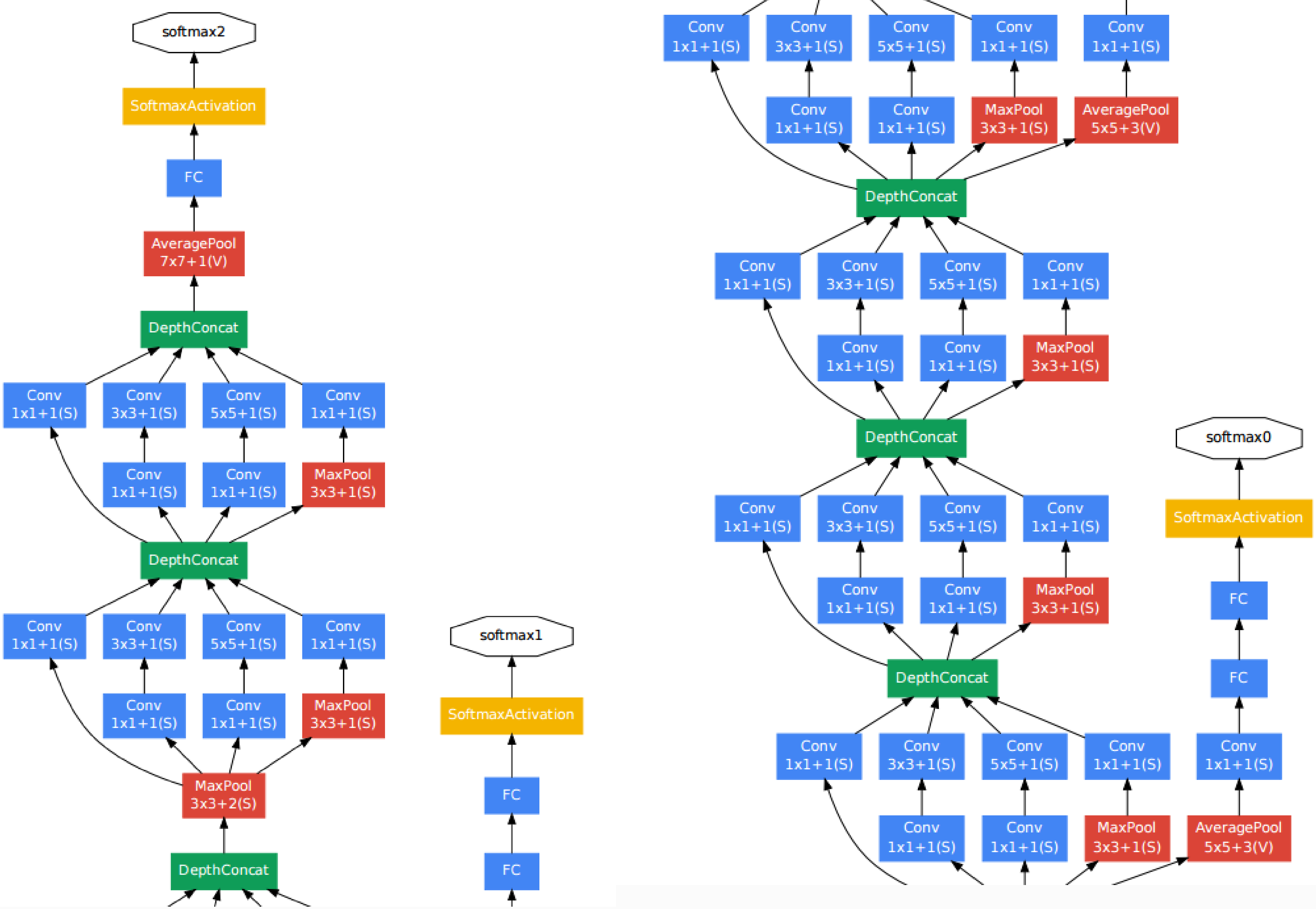
1.2 million images for training,
50,000 for validation and 100,000 images for testing

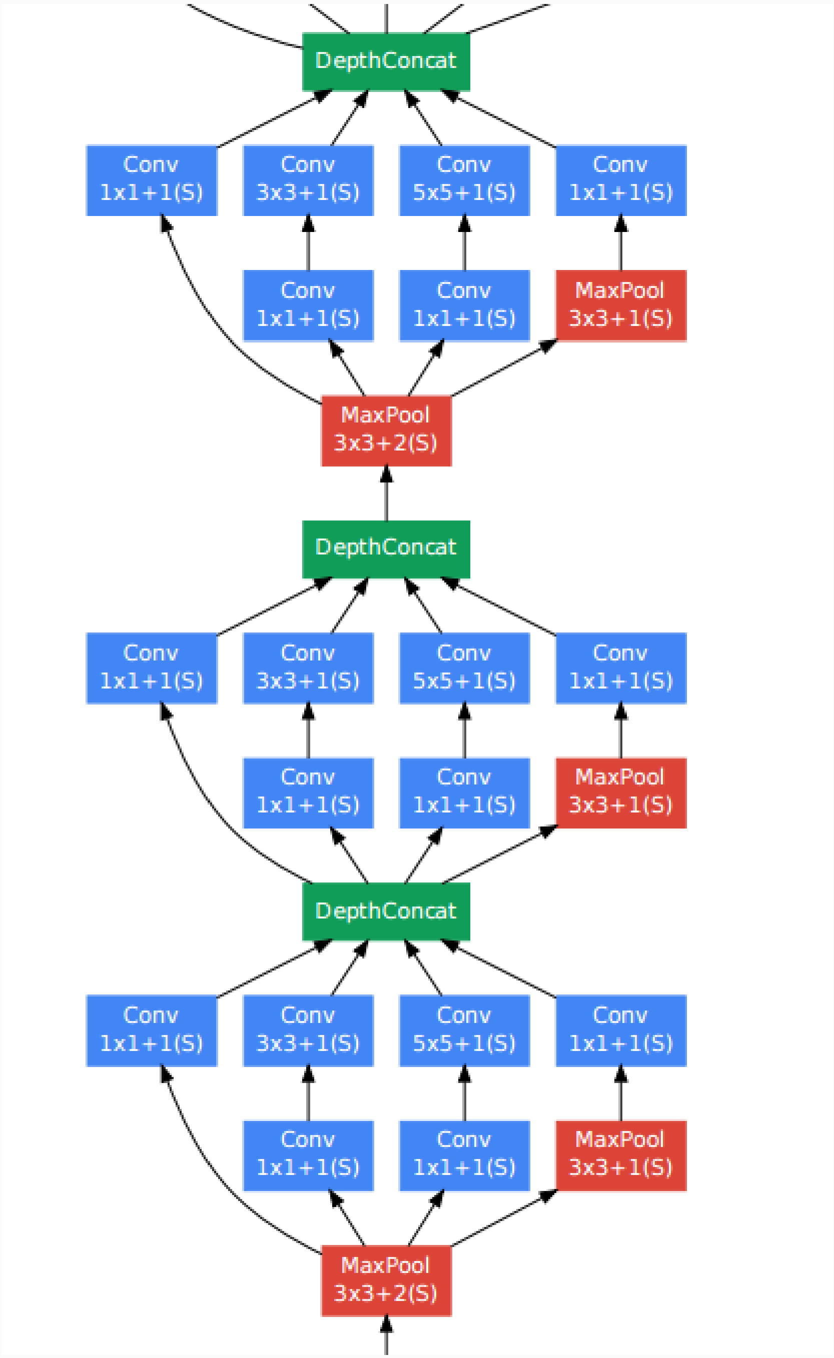
Team	Year	Place	mAP	external data	ensemble	approach
UvA-Euvision	2013	1st	22.6%	none	?	Fisher vectors
Deep Insight	2014	3rd	40.5%	ImageNet 1k	3	CNN
CUHK DeepID-Net	2014	2nd	40.7%	ImageNet 1k	?	CNN
GoogLeNet	2014	1st	43.9%	ImageNet 1k	6	CNN

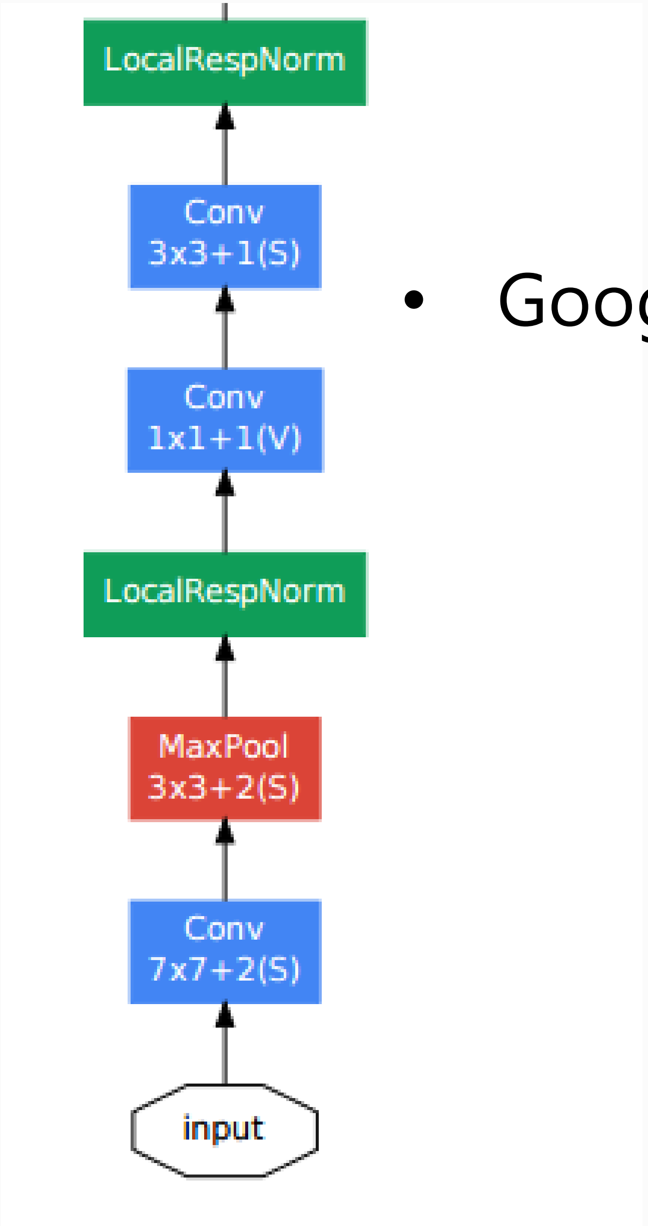
http://arxiv.org/pdf/1409.4842.pdf



GoogLeNet


softmax2
SoftmaxActivation
FC
AveragePool
7x7+1(V)
DepthConcat
Conv
1x1+1(S)
Conv
3x3+1(S)
Conv
5x5+1(S)
Conv
1x1+1(S)
Conv
1x1+1(S)
Conv
1x1+1(S)
MaxPool
3x3+1(S)
DepthConcat
Conv
1x1+1(S)
Conv
3x3+1(S)
Conv
5x5+1(S)
Conv
1x1+1(S)
Conv
1x1+1(S)
Conv
1x1+1(S)
MaxPool
3x3+1(S)
MaxPool
3x3+2(S)
DepthConcat
softmax1
SoftmaxActivation
FC
FC
Conv
1x1+1(S)
Conv
3x3+1(S)
Conv
5x5+1(S)
Conv
1x1+1(S)
Conv
1x1+1(S)
Conv
1x1+1(S)
Conv
1x1+1(S)
MaxPool
3x3+1(S)
AveragePool
5x5+3(V)
DepthConcat
Conv
1x1+1(S)
Conv
3x3+1(S)
Conv
5x5+1(S)
Conv
1x1+1(S)
Conv
1x1+1(S)
Conv
1x1+1(S)
MaxPool
3x3+1(S)
DepthConcat
Conv
1x1+1(S)
Conv
3x3+1(S)
Conv
5x5+1(S)
Conv
1x1+1(S)
Conv
1x1+1(S)
Conv
1x1+1(S)
MaxPool
3x3+1(S)
DepthConcat
Conv
1x1+1(S)
Conv
3x3+1(S)
Conv
5x5+1(S)
Conv
1x1+1(S)
Conv
1x1+1(S)
Conv
1x1+1(S)
MaxPool
3x3+1(S)
softmax0
SoftmaxActivation
FC
FC
Conv
1x1+1(S)
AveragePool
5x5+3(V)





- GoogLeNet was 22 layers deep!

- A team of Microsoft Researcher won the ImageNet Large-Scale Visual Recognition Challenge 2015

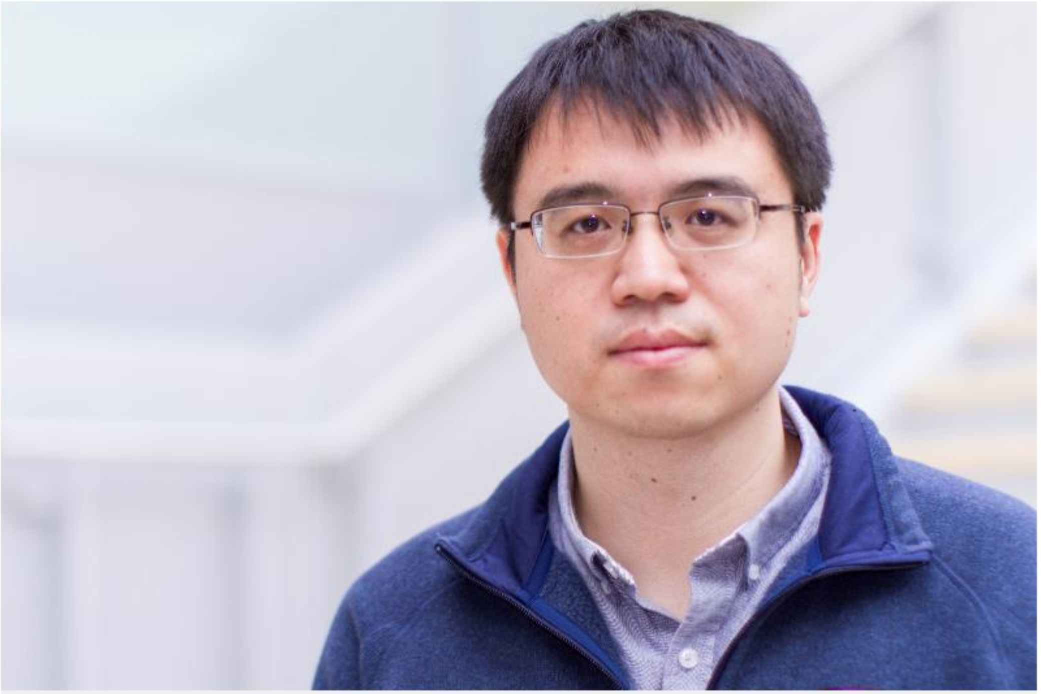
- Their solution used a Deep Convolutional net with 152 layers and an Ensemble approach

More at:

- http://arxiv.org/abs/1512.03385

The Fire Hose   Microsoft On the Issues

Microsoft researchers win ImageNet computer vision challenge

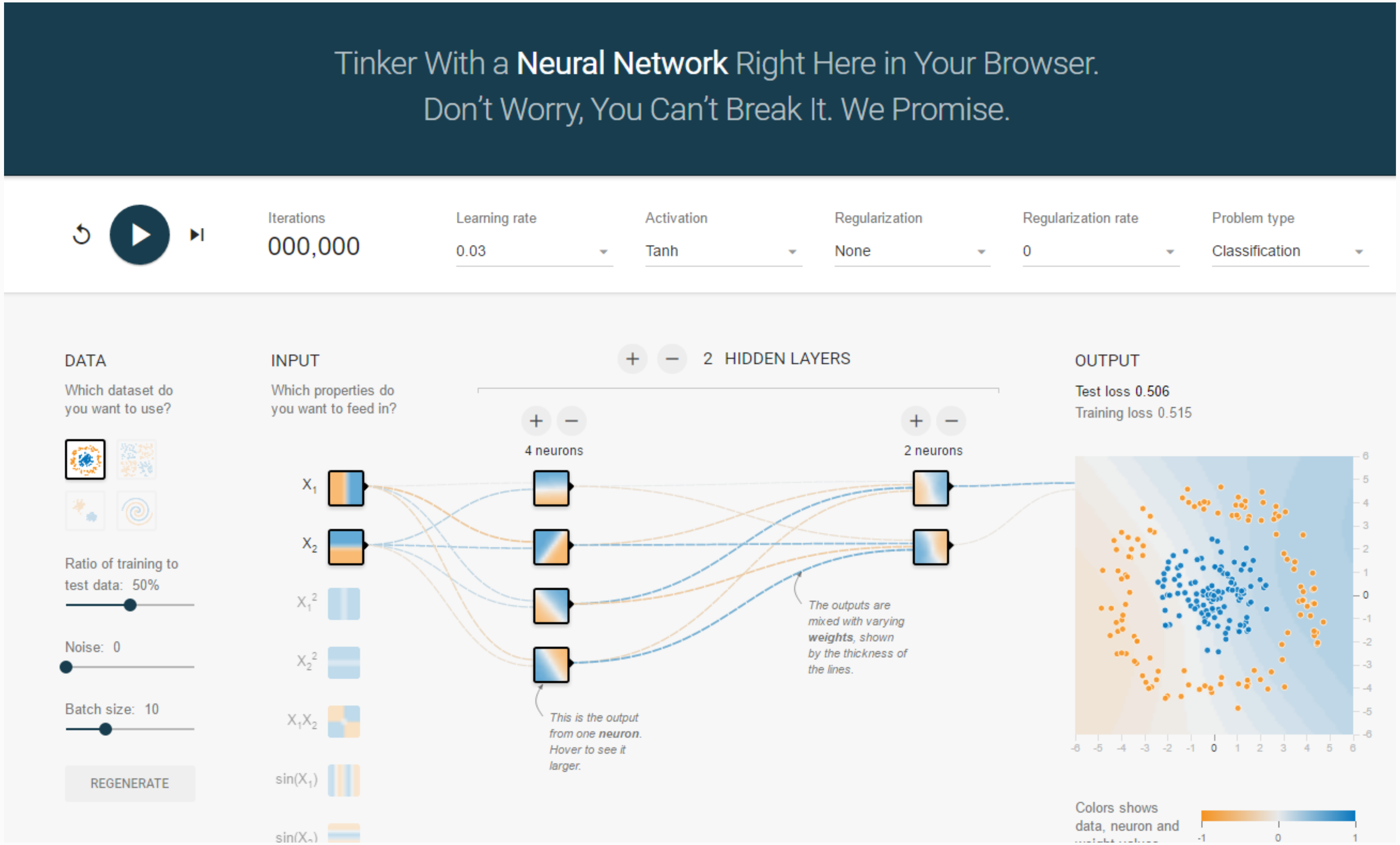


Jian Sun, a principal research manager at Microsoft Research, led the image understanding project. Photo: Craig Tuschhoff/Microsoft.

Posted December 10, 2015 By **Allison Linn**   f 184   in 175

Microsoft researchers on Thursday announced a major advance in technology designed to identify the objects in a photograph or video, showcasing a system whose accuracy meets and sometimes exceeds human-level performance.

Microsoft's new approach to recognizing images also took first place in several major categories of image recognition challenges Thursday, beating out many other competitors



http://playground.tensorflow.org/

Resources

Neural Networks

http://www.youtube.com/watch?v=GrugzF0-V3I

Machine Learning Applications for Data Center Optimization

Jim Gao, Google

Abstract

The modern data center (DC) is a complex interaction of multiple mechanical, electrical and controls systems. The sheer number of possible operating configurations and nonlinear interdependencies make it difficult to understand and optimize energy efficiency. We develop a neural network framework that learns from actual operations data to model plant performance and predict PUE within a range of 0.004 +/- 0.005 (mean absolute error +/- 1 standard deviation), or 0.4% error for a PUE of 1.1. The model has been extensively tested and validated at Google DCs. The results demonstrate that machine learning is an effective way of leveraging existing sensor data to model DC performance and improve energy efficiency.

1. Introduction

The rapid adoption of Internet-enabled devices, coupled with the shift from consumer-side computing to SaaS and cloud-based systems, is accelerating the growth of large-scale data centers (DCs). Driven by significant improvements in hardware affordability and the exponential growth of Big Data, the modern Internet company encompasses a wide range of characteristics including personalized user experiences and minimal downtime. Meanwhile, popular hosting services such as Google Cloud Platform and Amazon Web Services have dramatically reduced upfront capital and operating costs, allowing companies with smaller IT resources to scale quickly and efficiently across millions of users. These trends have resulted in the rise of large-scale DCs and their corresponding operational challenges.

One of the most complex challenges is power management. Growing energy costs and environmental responsibility have placed the DC industry under increasing pressure to improve its operational efficiency. According to Koomey, DCs comprised 1.3% of the global energy usage in 2010 [1]. At this scale, even relatively modest efficiency improvements yield significant cost savings and avert millions of tons of carbon emissions.