Artificial Neural Networks

Andreas Robinson

12/19/2012

Introduction

- Artificial Neural Networks
  - Machine learning technique
    - Learning from past experience/data
    - Predicting/classifying novel data
  - Biologically motivated: human brain
  - 1-layer networks related to Support Vector Machines
    - Universal approximators

Motivating Quote

- “We are currently experiencing a second Neural Network Rennaissance (the first one happened in the 1980s and early 90s). In many applications, our deep NNs are now outperforming all other methods including the theoretically less general and less powerful *support vector machines* (which for a long time had the upper hand, at least in practice)”
  – Dr. Jurgen Schmidhuber
  – Between 2009-2012 Swiss AI lab has won 8 international pattern recognition contests and currently hold record for several machine learning benchmark datasets.

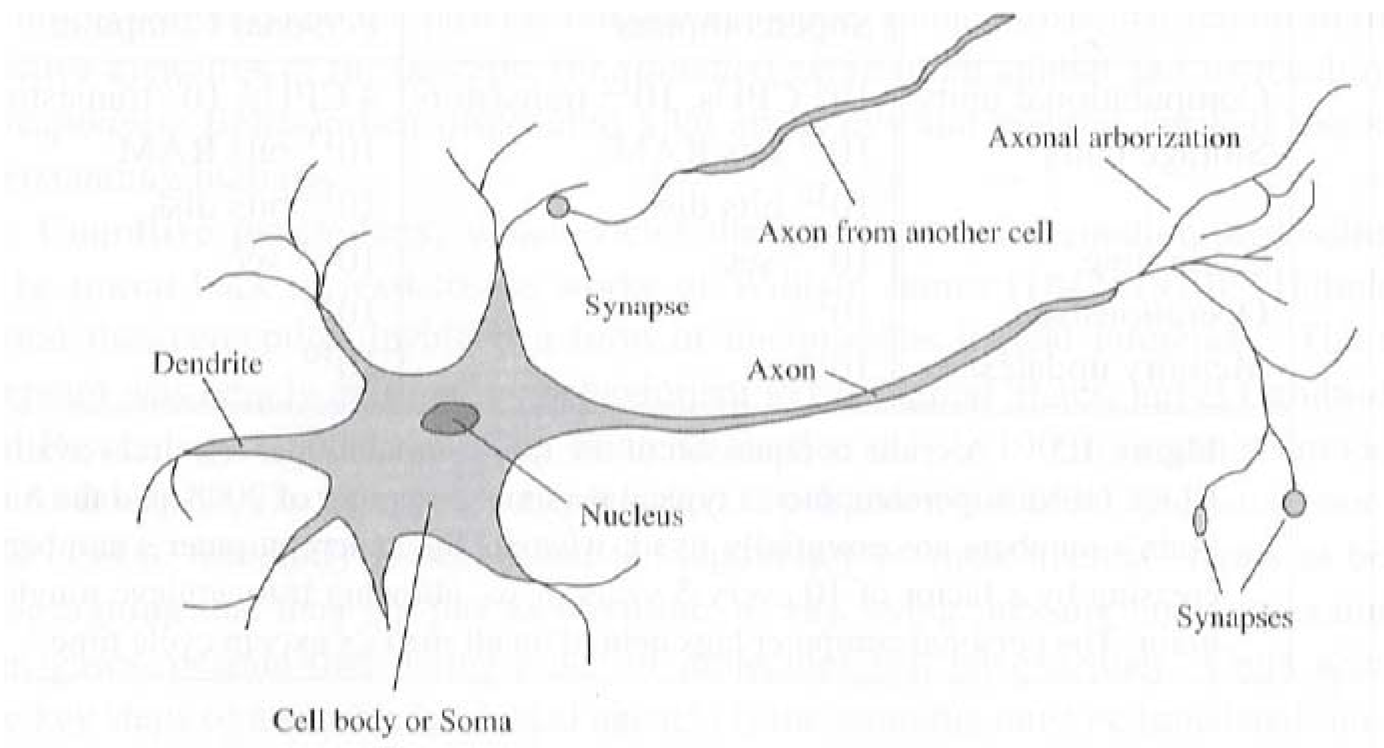
NN Example Applications

- Post Office OCR
  - Used to recognize hand-written zip code digits for the postal service.
  - Also, bank check readers
- DARPA grand challenge
  - Used by winning team as part of solution for extracting roads from aerial imagery
- DARPA Deep learning BAA – 2009 to present
  - Unsupervised deep architectures – automatic feature extraction
  - Much of the research -> deep neural networks and related approaches
- Goodrich Aerospace:
  - Learning telemetry mapping from shear ports
  - Pitot probes
- Primordial proposed as possible approach for Natick Phase 2 Land cover extraction
  - Deep neural networks
  - Previously have tried: SVMs, Max Likelihood, EM, region segmentation

Summary

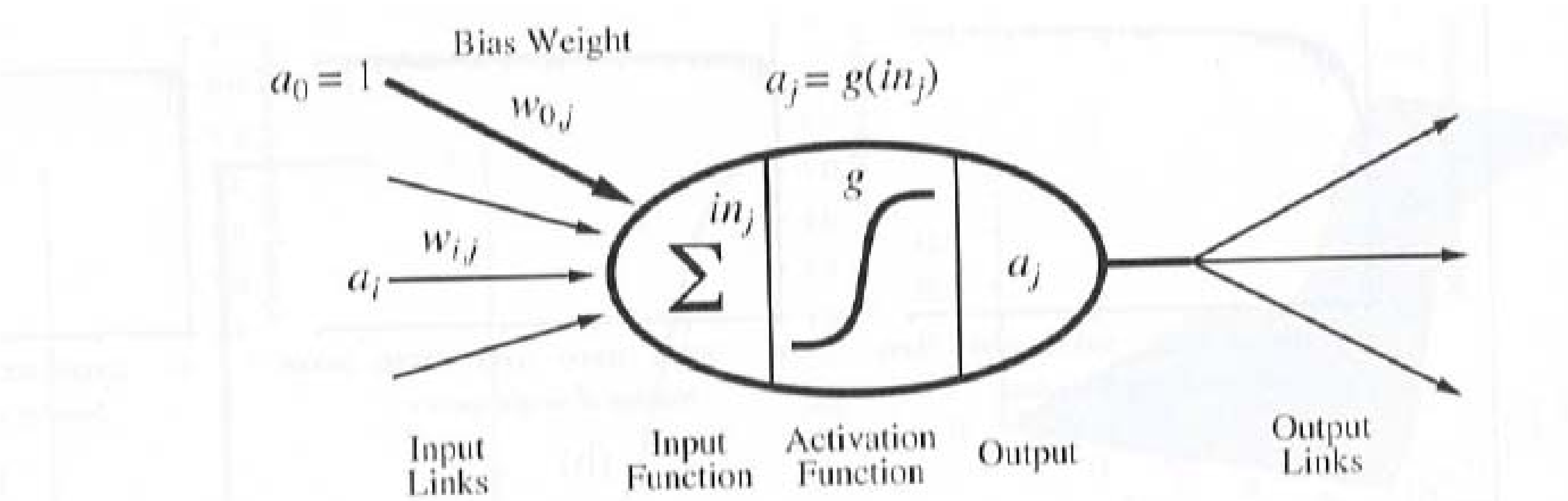
- Neurons
- Single-layer Networks
  - Perceptrons
  - 1950s-60s
- Multi-layer Networks (1-2 layers)
  - 1980s-90s
  - Demo
- Deep neural networks
- Recurrent neural networks (briefly)
- Competitions / Benchmarks
- Libraries

Biological Neuron



	Neurons	Synapses	Ops / Sec
Human Brain	10^{11}	10^{14}	10^{17}

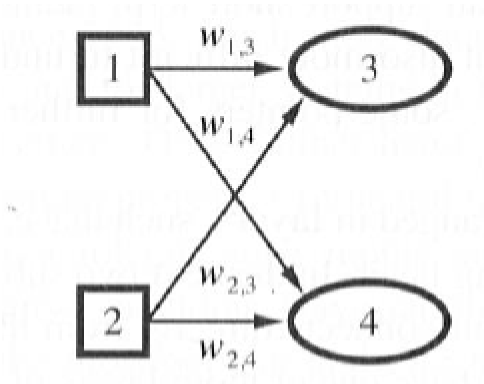
Artificial Neuron



$$a_j = g(in_j) = g\left(\sum_{i=0}^{n} w_{i,j}a_i\right)$$

Perceptrons

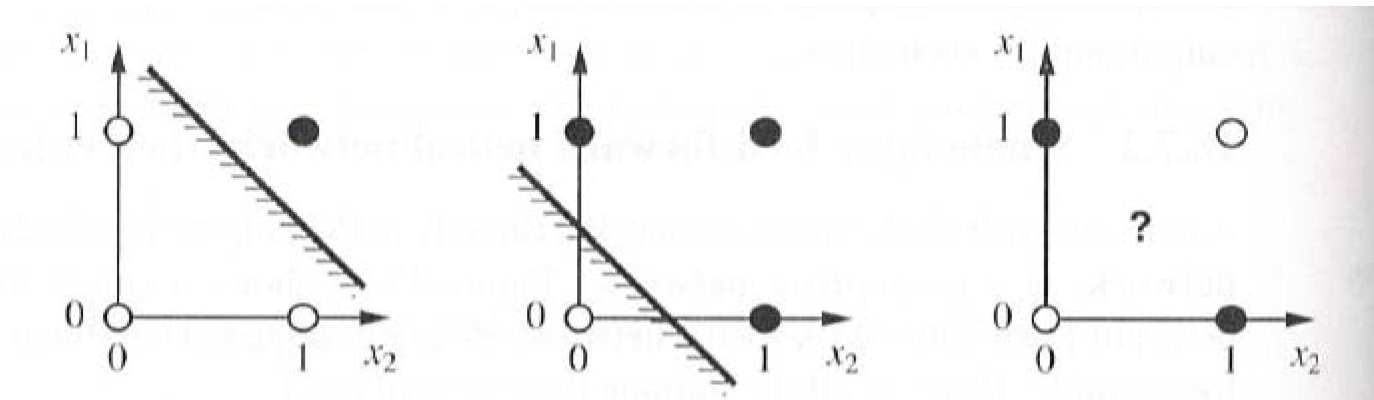
- Invented: Rosenblatt 1957
- Structure
  - Input/output layers
  - No “hidden” layers
  - Activation function
    - Hard threshold
- Feed forward
  - Shape of decision boundary?
- Learning rule
  - $W_i \leftarrow W_i + \text{alpha} * (y - h_w(\mathbf{x})) * x_i$



Perceptron Example:
Something missing?

Perceptron Limitations

- Linear separators
- Problematic cases
- Decline of neural network research
  - "Perceptrons" Minsky & Papert 1969
  - Also first AI winter

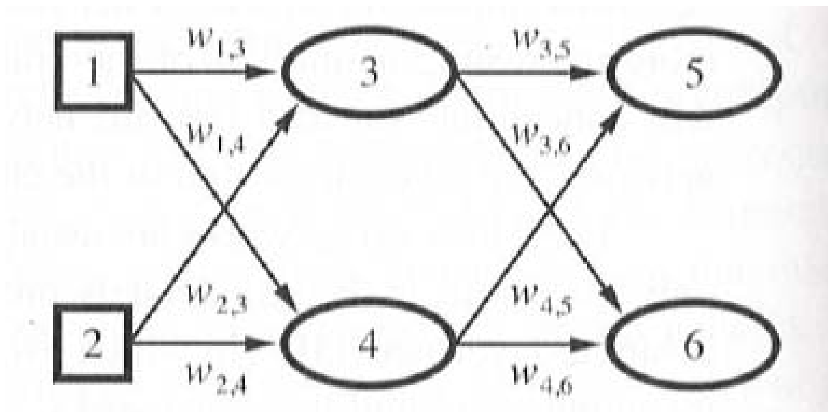


(a) x_1 **and** x_2 (b) x_1 **or** x_2 (c) x_1 **xor** x_2

Multilayer Neural Networks

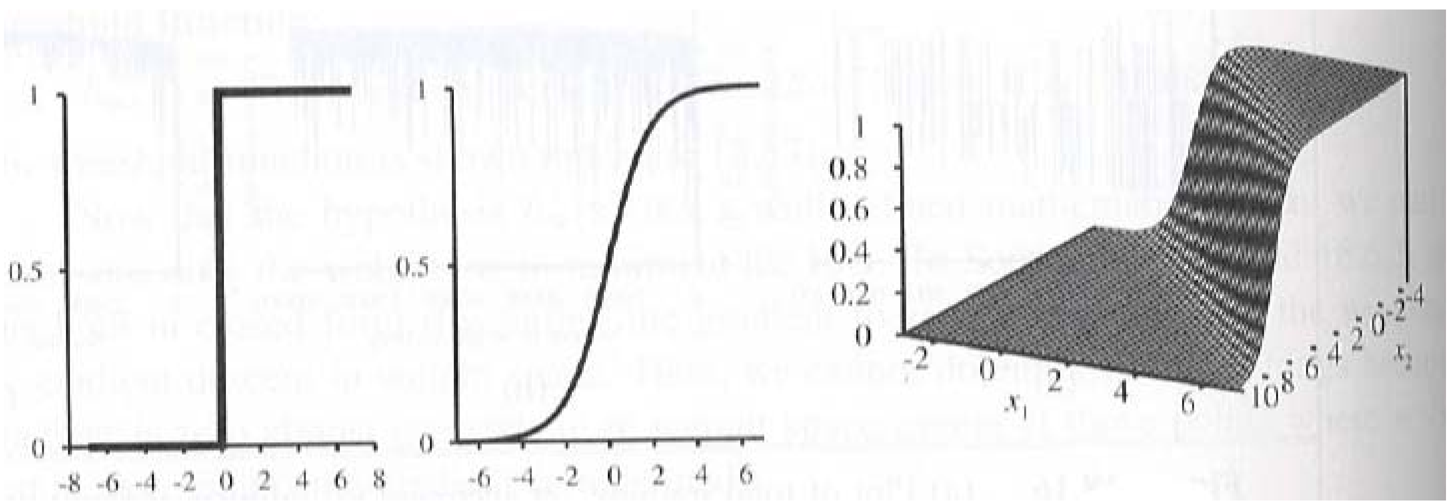
- Key innovation
  - Technique for training more than one layer
  - Back propagation
- Reinvigorated interest in neural nets
  - Back prop invented: 1969 [Ho]
  - Reinvented: 1974 [Werbos], 1985 [Park]
  - Widespread use: 1980s, early 90s
- Addressed key deficiencies that had been raised with perceptrons e.g. XOR
- Still feed-forward: typically 1-2 hidden layers

Multilayer Neural Network



Activation functions

- Hard threshold
- Support non-linearities
- Sigmoid
  - Differentiable



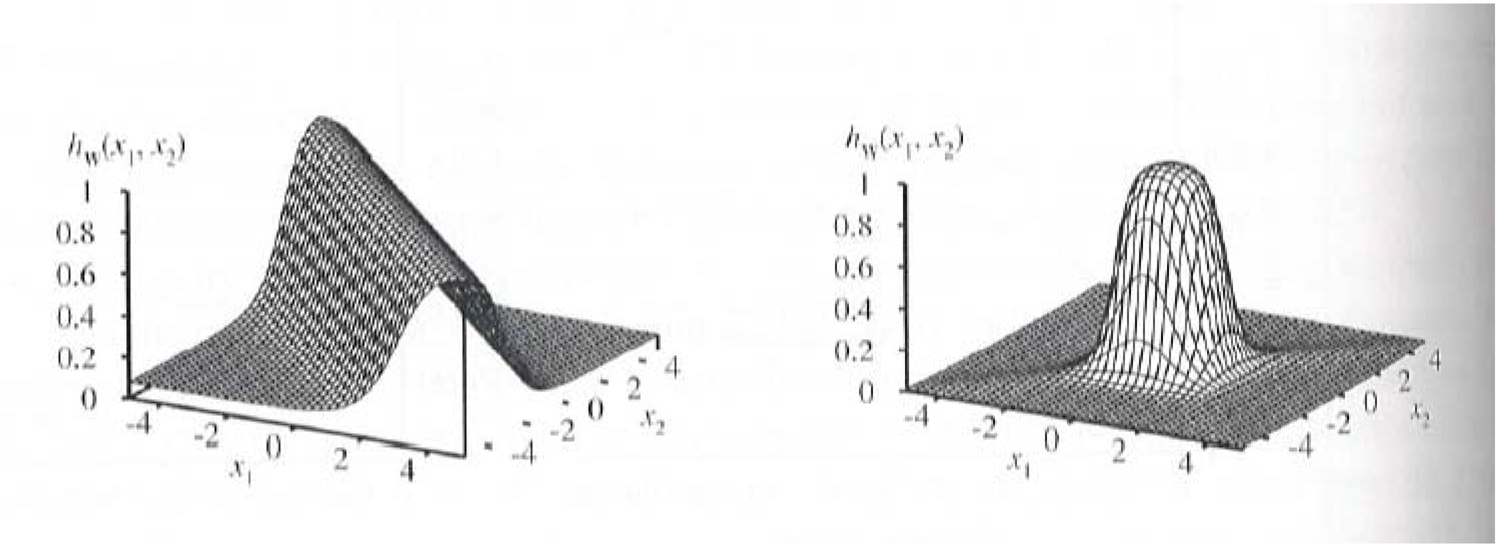
Training network

- Back propagation:
  – Gradient descent to minimize error on training sample
    - Adjust weights in direction that locally minimizes the error
    - Direction determined by gradient of error function (local partial derivatives) on training samples
      – (- dE/dW0, … , - dE/dW1)
  – Differentiable activation function, squared error => closed form derivative
  – Resulting weight update eqn, errors propagate back through network

Multilayer Networks - Theory

- Universal approximator
  - 1 hidden layer: Finite domain, continuous functions
- Local minima
  - Momentum
  - Retraining
- Typical topology
  - 1-2 layers
  - Regression vs classification (output activation)
- Determining the structure and parameters
  - Overfitting
  - Cross-validation
  - Early stopping
  - Regularization

Universal Approximator Visualization



Pre-processing

- Manual feature selection
  - HOGs, wavelets, shape descriptors, statistical properties, SIFT
  - Reduce dimensionality
  - Improve separability
- Segmentation – entity detection
- Training data deformations/invariants

Neural Network Demo

- Sample app
  - svn://bordeaux/source/ClassifySVM/
  - OpenCV
  - Number of features?
  - Adjustable Parameters:
    - Number of hidden nodes
    - Training iterations

Deep Networks

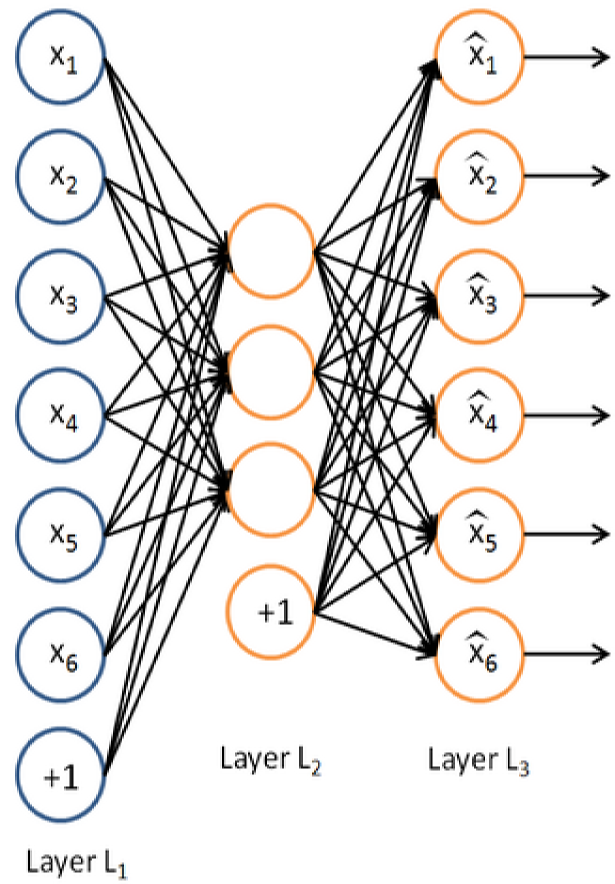
- Definition
  - Number of layers
  - Hierarchical structure
- Deep networks traditionally not used
  - Common phallacy:
    - 1 layer universal approximator
  - Lack of effective training algorithms; local minima, vanishing gradients, small training sets
- Automatic feature extraction
  - Lower layers
- Unsupervised:
  - Recent algorithms for training
  - E.g. Stacked auto-encoders, Restricted Boltzmann Machines (RBMs), etc
- Supervised
  - Convolutional networks
  - Unsupervised pre-training

Motivation – Deep Structures

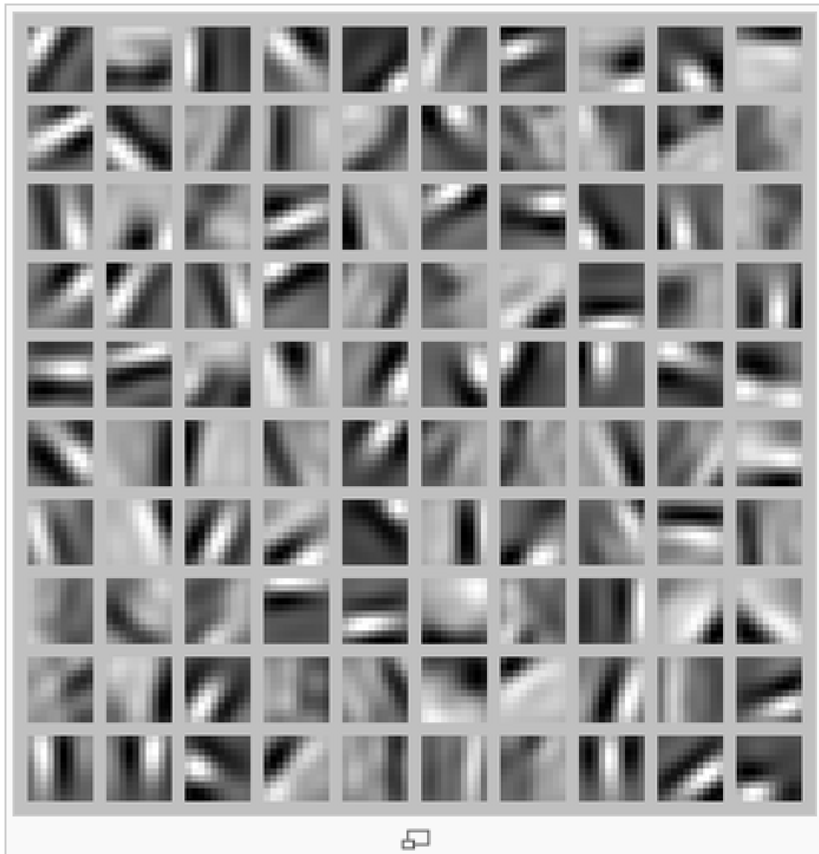
- Single Layer - Universal approximator
- Compact representations
  - "most functions representable compactly with a deep architecture would require a very large number of components if represented with a shallow one"*
  - Example: "For all k, there are depth k+1 circuits of linear size that require exponential size to simulate with depth k circuits"
    - Complexity in terms of number of bits or number of input nodes
  - Generalization
    - Lookup table: linear in sample size, exponential in bits
    - Sub-exponential representation => underlying pattern

Unsupervised Auto-encoders

- No labeled training samples
- Sparse auto-encoder
  - Learn compact representation
    - Input => input
    - Small number of hidden nodes, or bias in optimization towards zero-valued weights
  - Can use back-prop to train network



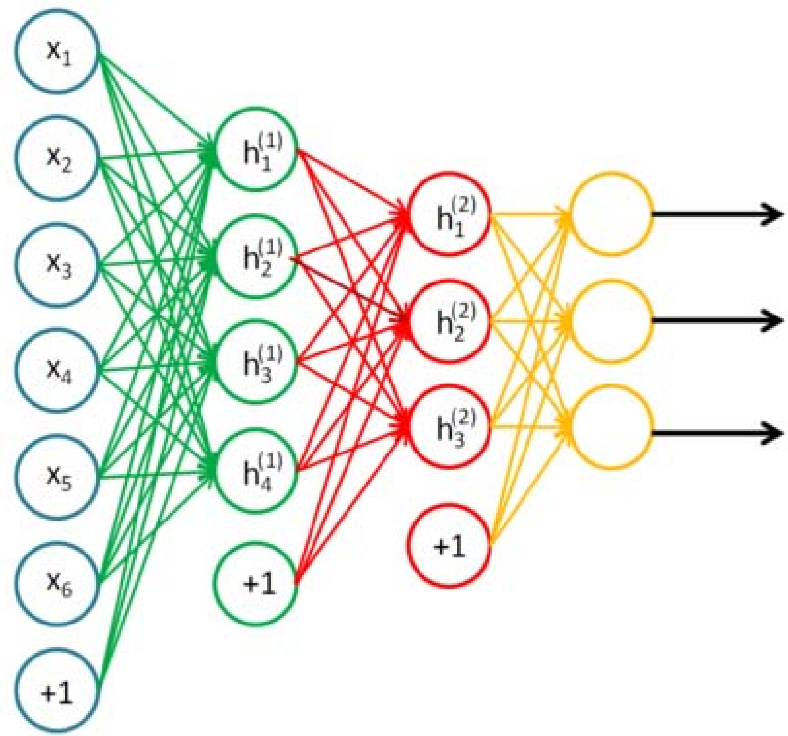
Auto-encoder Visualization



Images that maximize activation of each hidden node "feature"

Stacked Auto-encoders

- Stacked auto-encoders
  - Hierarchical, deep structure
  - Hidden nodes represent features
  - Low-level to more abstract features
    - Edges, shapes, faces
- Unsupervised pre-training



Google YouTube Classification

- Google team, 2012 unsupervised learning from YouTube images
  - Stacked Sparse Auto-encoders
    - 9 layers
    - 1 billion weight connections (compared to roughly 10^14 in brain)
  - 10 million images; random unlabeled YouTube frames
  - 1000 machine cluster (16,000 cores) trained for 3 days
  - Pooling and local contrast normalization
  - Local receptive field
- Who:
  - Jeff Dean, a Google technical fellow (et al)
  - Andrew Y. Ng, a Stanford computer scientist
- Current record on ImageNet database
  - Supervised pre-training
  - 20k object types
  - 70% better then previous best
  - 15.8% accuracy (random guess: .005%)
  - Challenging dataset
- LSVRC ImageNet 2012 Challenge (not Google):
  - Deep Convolutional Network (GPU)
  - Best team: 15.3% error (as opposed to accuracy); 26.1% runner up (SIFT features)
  - 1000 object types

– Deep learning quote winning team: “The point about this approach is that it scales beautifully. Basically you just need to keep making it bigger and faster, and it will get better. There’s no looking back now.” [Hinton]

Google Auto-encoder Visualization

Images that maximize activation of two hidden node "features"

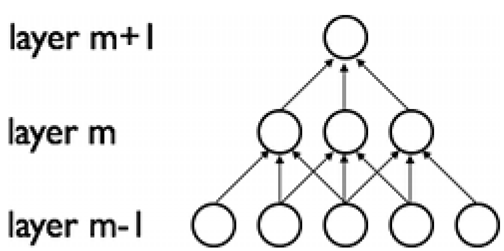
Deep Networks - Supervised

- Convolutional Neural Networks
  - Current top approach in many machine learning competitions/datasets
  - Biologically motivated
    - Visual cortex
    - Local receptive field
  - Shared weights
  - Reduced search space; translation invariance
  - Trained with back prop
  - Yan LeCun (90s)
- Unsupervised pre-training
  - E.g. stacked-auto-encoders
  - Google approach, non-convolutional
  - Swiss team down-played

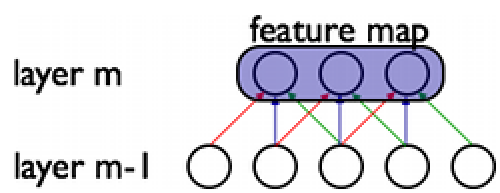
Convolutional Networks

- Local receptive field
  - Biological motivation
- Shared Weights (Convolutions)
  - Multiple feature maps per layer
  - Translation invariance
- Pooling (Max)
  - Downsampling

Local Receptive Field:



Shared Weights:



Deep Network Results

- Swiss AI lab (Dr. Schmidhuber)
  - Primarily: Convolutional Networks and Recurrent Networks (see next slide)
  - GPU implementations
  - In some cases, raw imagery rather than manual features
- Since 2009:
  - Lab has won 8 first prizes in visual pattern recognition contests
  - Including better than human performance in sign recognition (IJCNN 2011)
- Top performance in following benchmarks:
  - MNIST Handwritten Digits Benchmark ("1st human-competitive result in 2011")
    - 0.23% error
  - NORB Object Recognition Benchmark
  - CIFAR Image Classification Benchmark
  - The Weizmann & KTH Human Action Recognition Benchmarks

Recurrent neural networks

- Backwards connections (loops)
  - Human brain
  - Turing complete
    - Compact representations
- Top performance in several hand-writing recognition competitions
  - ICDAR 2009: the *Arabic Connected Handwriting Competition*, the *Handwritten Farsi/Arabic Character Recognition Competition*, and the *French Connected Handwriting Competition*
  - Same Swiss team as prev slide (Shmidhuber)

Some existing libraries

- **OpenCV:**
  - Basic shallow neural network implementation
  - Primarily a computer vision library
- **Fast Neural Network Library (FANN):**
  - C++ library for efficient feed-foward networks
  - http://leenissen.dk/fann/wp/
- **Pynnet:**
  - Python library for deep neural networks
  - Stacked auto-encoders, convolutional networks, recurrent networks, etc
  - http://code.google.com/p/pynnet/

References

- Russell & Norvig AI Textbook
- http://www.idsia.ch/~juergen/vision.html
- http://deeplearning.net/tutorial/lenet.html
- http://research.google.com/archive/unsupervised_icml2012.html
- http://ufldl.stanford.edu/wiki/index.php/Autoencoders_and_Sparsity
- http://ufldl.stanford.edu/wiki/index.php/Stacked_Autoencoders
- http://www.nytimes.com/2012/11/24/science/scientists-see-advances-in-deep-learning-a-part-of-artificial-intelligence.html?pagewanted=2&_r=1