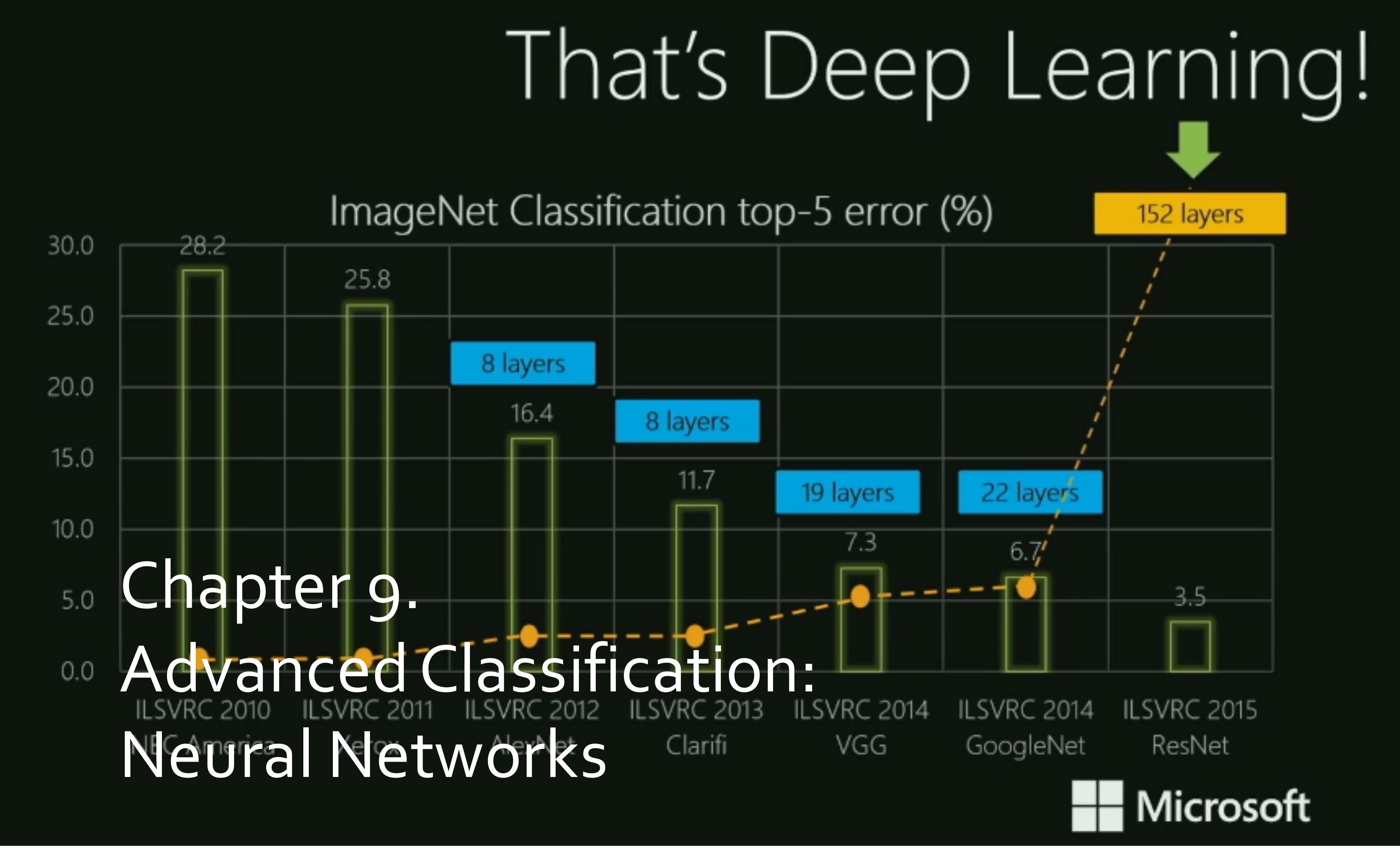


Chapter 9. Advanced Classification: Neural Networks

Meng Jiang

CSE 40647/60647 Data Science Fall 2017

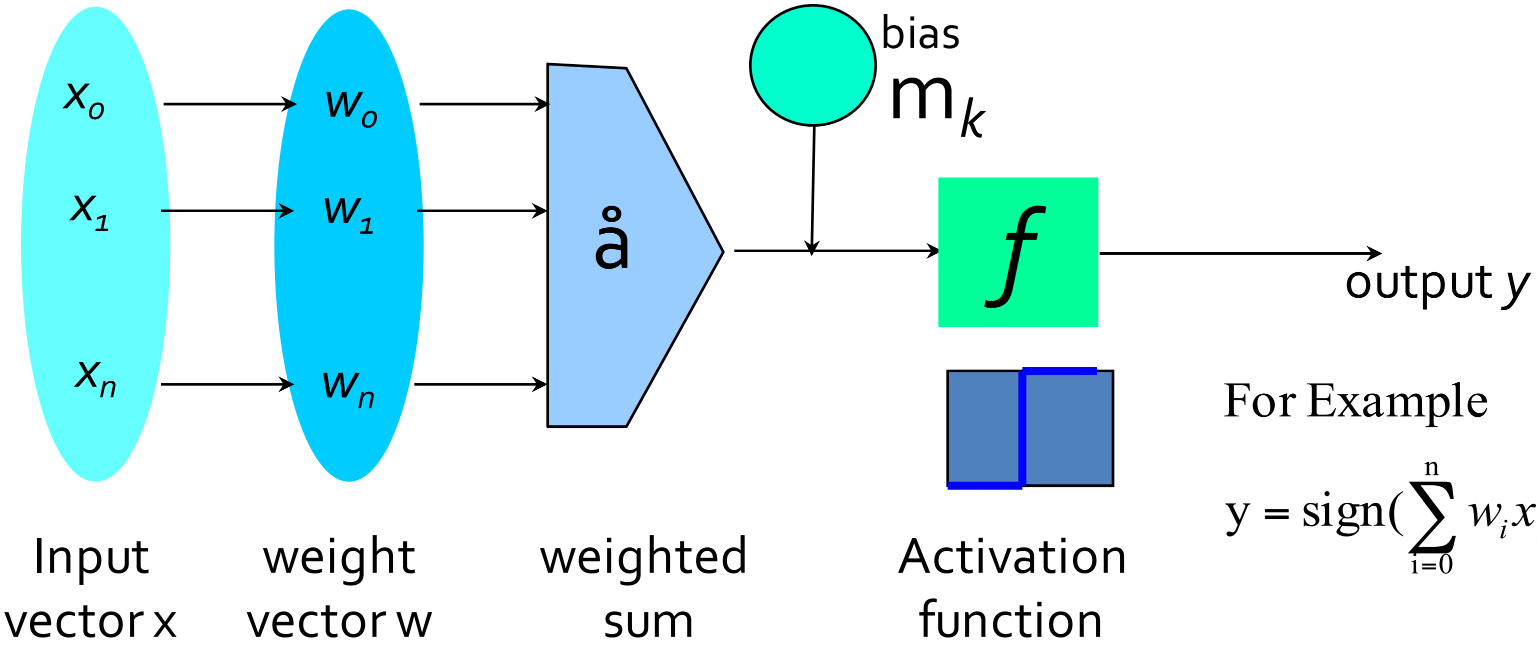
Introduction to Data Mining

Neural Network for Classification

- Started by psychologists and neurobiologists to develop and test computational analogues of neurons
- A neural network: A set of connected input/output units where each connection has a **weight** associated with it
  - During the learning phase, the **network learns by adjusting the weights** so as to be able to predict the correct class label of the input tuples
- Also referred to as **connectionist learning** due to the connections between units

Neuron: A Hidden/Output Layer Unit

- An *n*-dimensional input vector **x** is mapped into variable y by means of the scalar product and a nonlinear function mapping
- The inputs to unit are outputs from the previous layer. They are multiplied by their corresponding weights to form a weighted sum, which is added to the bias associated with unit. Then a nonlinear activation function is applied to it.

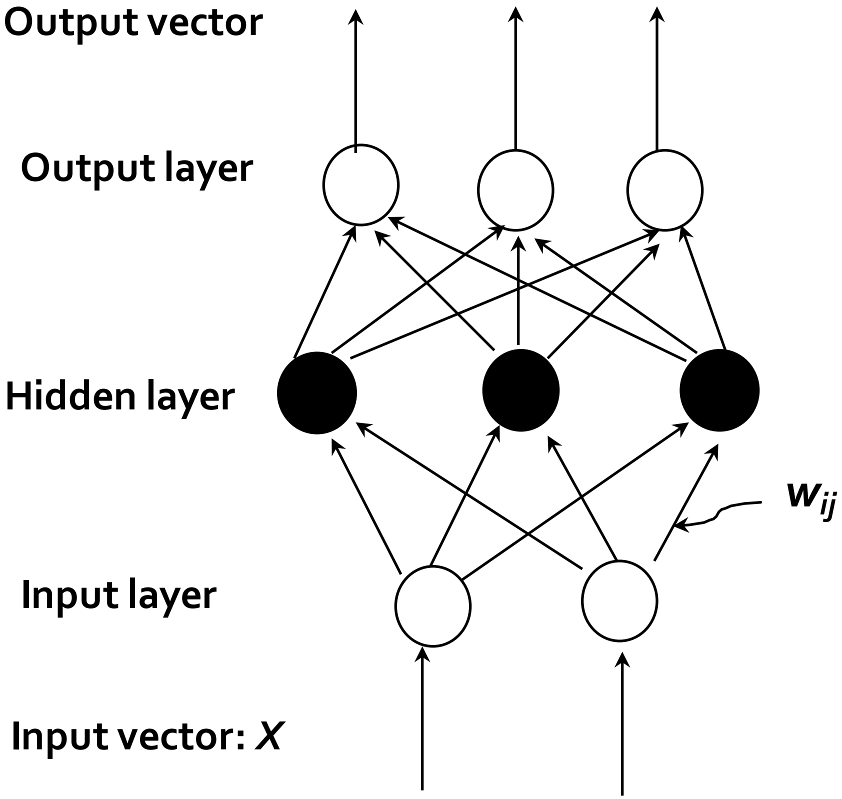


For Example

$$y = \mathrm{sign}\left(\sum_{i=0}^{n} w_i x_i - \mu_k\right)$$

A Multi-Layer Feed-Forward Neural Network

$$\boxed{w_j^{(k+1)} = w_j^{(k)} + \lambda(y_i - \hat{y}_i^{(k)})x_{ij}}$$



Output vector

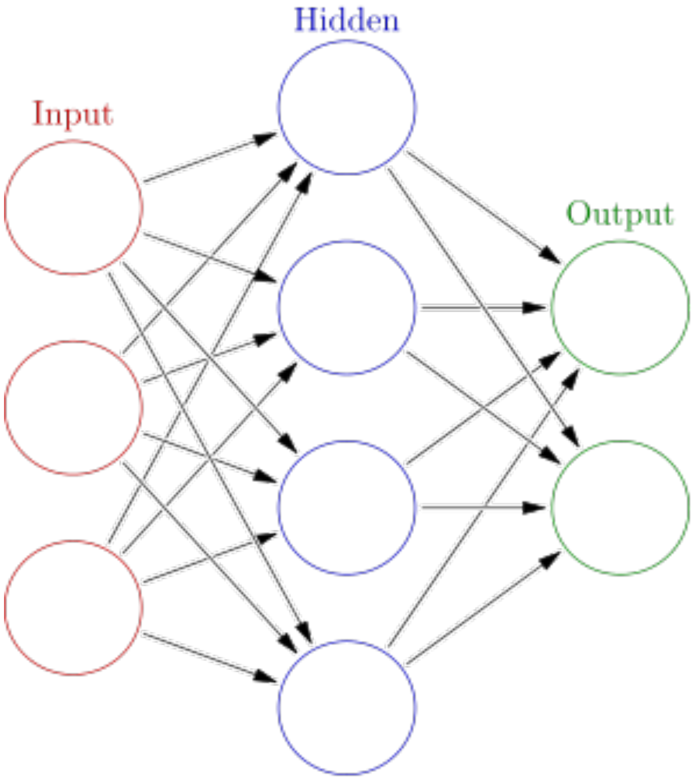
Output layer

Hidden layer

Input layer

Input vector: *X*

w_{ij}



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 (and what?), they can closely approximate any function

Defining a Network Topology

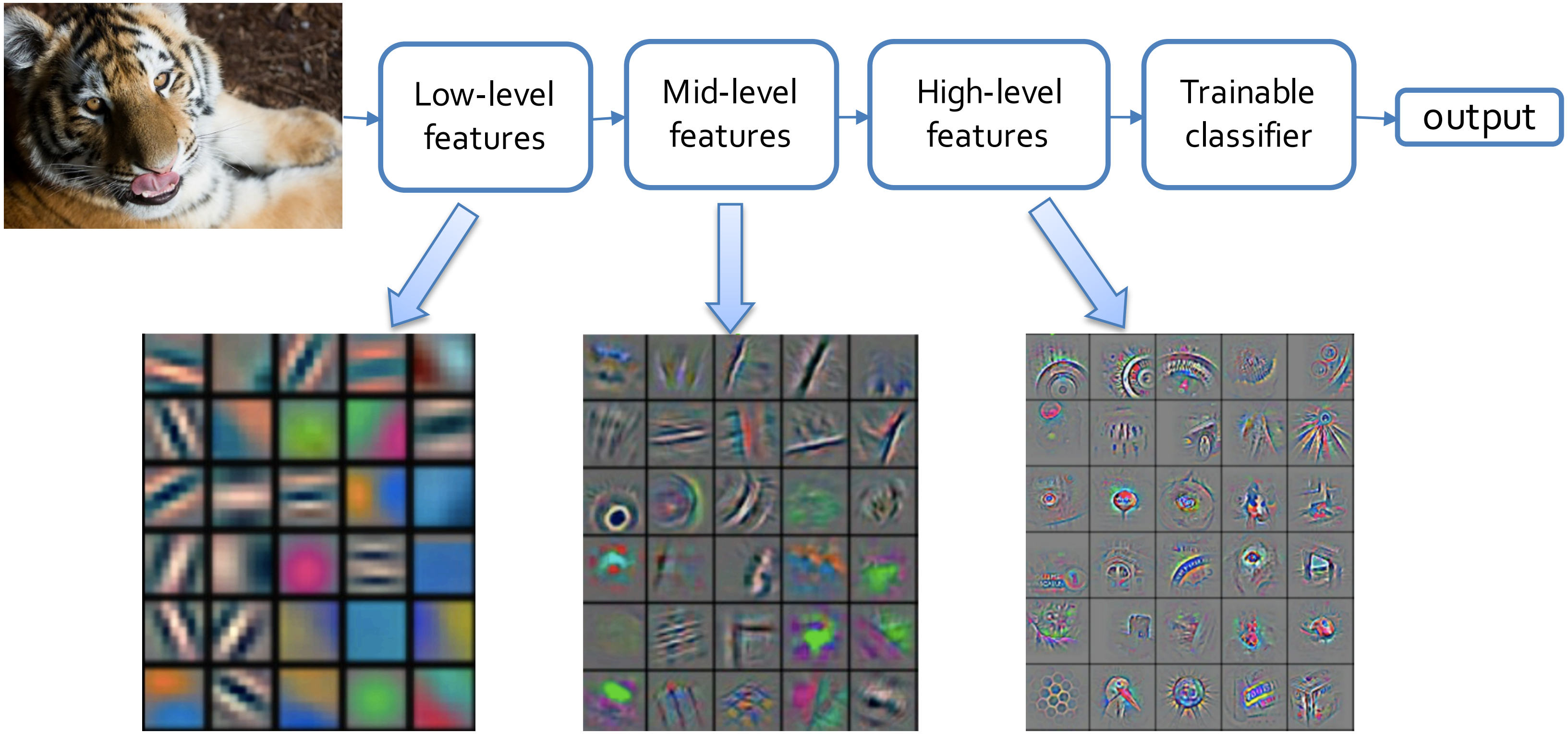
- Decide the **network topology**
  - Specify # of units in the *input layer*, # of *hidden layers* (if > 1), # of units in *each hidden layer*, and # of units in the *output layer*
- Normalize the input values for each attribute measured in the training tuples to [0.0—1.0]
- Once a network has been trained and its accuracy is **unacceptable**, repeat the training process with a *different network topology* or a *different set of initial weights*

From Neural Networks to Deep Learning

- Train networks with many layers (vs. shallow nets with just a couple of layers)
- Multiple layers work to build an improved feature space
  - First layer learns 1st order features (e.g., edges, …)
  - 2nd layer learns higher order features (combinations of first layer features, combinations of edges, etc.)
  - In current models, layers often learn in an unsupervised mode and discover general features of the input space—serving multiple tasks related to the unsupervised instances (image recognition, etc.)
  - Then final layer features are fed into supervised layer(s)
    - And entire network is often subsequently tuned using supervised training of the entire net, using the initial weightings learned in the unsupervised phase

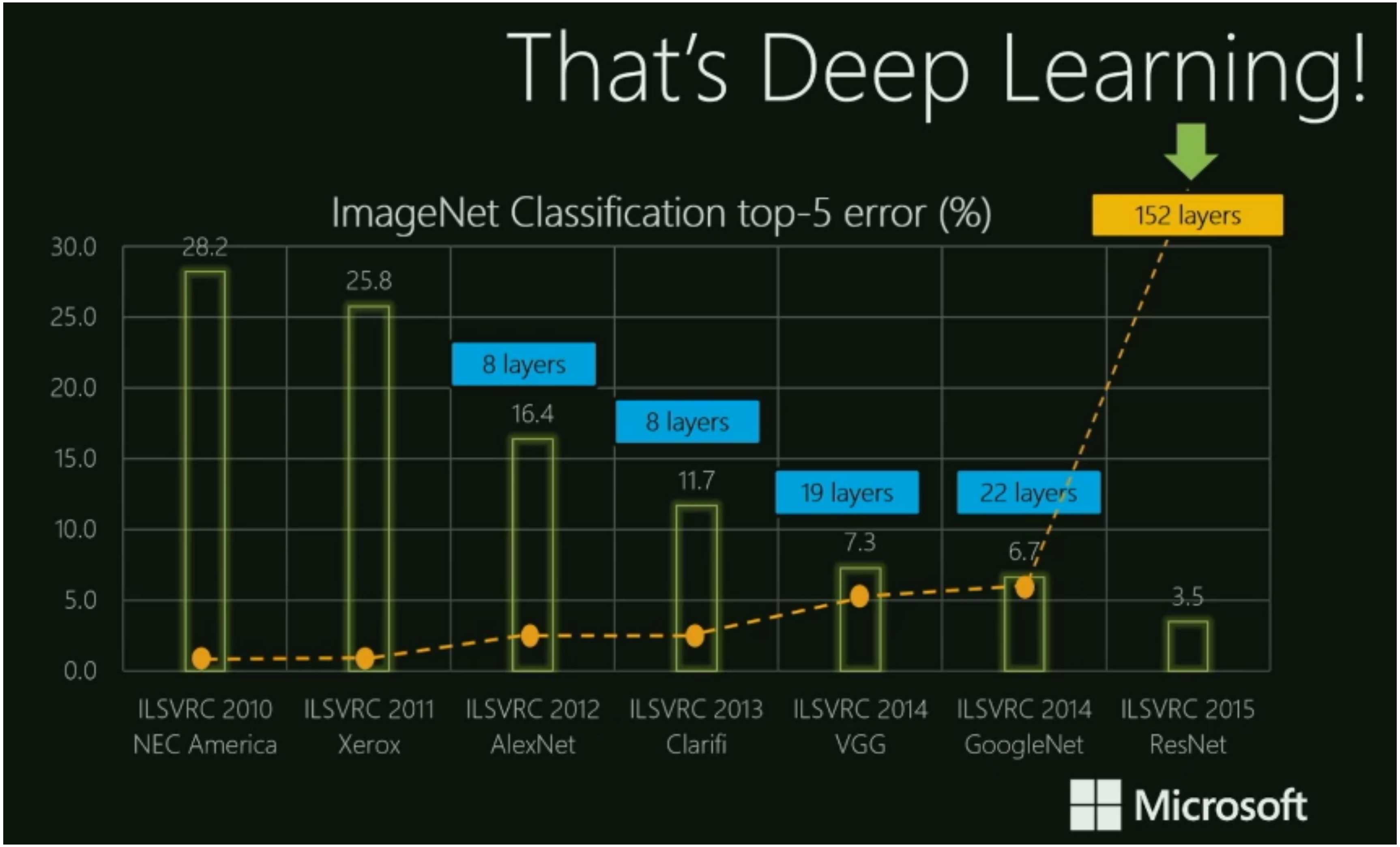
Deep Learning: Feature Visualization

- Deep learning (a.k.a. representation learning) seeks to learn rich hierarchical representations (i.e. features) automatically through multiple stage of feature learning process.



Feature visualization of convolutional net trained on ImageNet (Zeiler and Fergus, 2013)

Deep Learning on ImageNet



Limitations of Neural Networks

Random initialization + **densely connected networks** lead to:

- High cost
  - Each neuron in the neural network can be considered as a logistic regression.
  - Training the entire neural network is to train all the interconnected logistic regressions.
- Difficult to train as the number of hidden layers increases
  - Recall that logistic regression is trained by gradient descent.
  - In backpropagation, gradient is progressively getting more dilute. That is, below top layers, the correction signal δ_n is minimal.
- Stuck in local optima
  - The objective function of the neural network is usually not convex.
  - The random initialization does not guarantee starting from the proximity of global optima.

References

- C. M. Bishop, Neural Networks for Pattern Recognition. Oxford University Press, 1995
- L. Breiman, J. Friedman, R. Olshen, and C. Stone. Classification and Regression Trees. Wadsworth International Group, 1984
- C. J. C. Burges. A Tutorial on Support Vector Machines for Pattern Recognition. Data Mining and Knowledge Discovery, 2(2): 121-168, 1998
- N. Cristianini and J. Shawe-Taylor, Introduction to Support Vector Machines and Other Kernel-Based Learning Methods, Cambridge University Press, 2000
- H. Yu, J. Yang, and J. Han. Classifying large data sets using SVM with hierarchical clusters. KDD'03
- A. J. Dobson. An Introduction to Generalized Linear Models. Chapman & Hall, 1990
- R. O. Duda, P. E. Hart, and D. G. Stork. Pattern Classification, 2ed. John Wiley, 2001
- T. Hastie, R. Tibshirani, and J. Friedman. The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer-Verlag, 2001
- S. Haykin, Neural Networks and Learning Machines, Prentice Hall, 2008
- D. Heckerman, D. Geiger, and D. M. Chickering. Learning Bayesian networks: The combination of knowledge and statistical data. Machine Learning, 1995
- H. Cheng, X. Yan, J. Han & C.-W. Hsu, Discriminative Frequent Pattern Analysis for Effective Classification, ICDE‘07
- W. Cohen. Fast effective rule induction. ICML'95

References (cont.)

- H. Cheng, X. Yan, J. Han & P. S. Yu, Direct Discriminative Pattern Mining for Effective Classification, ICDE'08
- G. Cong, K. Tan, A. Tung & X. Xu. Mining Top-k Covering Rule Groups for Gene Expression Data, SIGMOD'05
- M. Deshpande, M. Kuramochi, N. Wale & G. Karypis. Frequent Substructure-based Approaches for Classifying Chemical Compounds, TKDE'05
- G. Dong & J. Li. Efficient Mining of Emerging Patterns: Discovering Trends and Differences, KDD'99
- W. Fan, K. Zhang, H. Cheng, J. Gao, X. Yan, J. Han, P. S. Yu & O. Verscheure. Direct Mining of Discriminative and Essential Graphical and Itemset Features via Model-based Search Tree, KDD'08
- W. Li, J. Han & J. Pei. CMAR: Accurate and Efficient Classification based on Multiple Class-association Rules, ICDM'01
- B. Liu, W. Hsu & Y. Ma. Integrating Classification and Association Rule Mining, KDD'98
- J. R. Quinlan and R. M. Cameron-Jones. FOIL: A midterm report. ECML'93
- Jingbo Shang, Wenzhu Tong, Jian Peng, and Jiawei Han, "DPClass: An Effective but Concise Discriminative Patterns-Based Classification Framework", SDM'16
- J. Wang and G. Karypis. HARMONY: Efficiently Mining the Best Rules for Classification, SDM'05
- X. Yin & J. Han. CPAR: Classification Based on Predictive Association Rules, SDM'03

Deep Learning Short Tutorial: CNNs

- Acknowledgement: Many of the pictures, results, and other materials are taken from:
  - Aarti Singh, Carnegie Mellon University
  - Andrew Ng, Stanford University
  - Barnabas Poczos, Carnegie Mellon University
  - Christopher Manning, Stanford University
  - Geoffrey Hinton, Google & University of Toronto
  - Richard Socher, MetaMind
  - Richard Turner, University of Cambridge
  - Yann LeCun, New York University
  - Yoshua Bengio, Universite de Montreal

nature
THE INTERNATIONAL WEEKLY JOURNAL OF SCIENCE
At last — a computer program that can beat a champion Go player PAGE 484
ALL SYSTEMS GO
CONSERVATION
SONGBIRDS À LA CARTE
Illegal harvest of millions of Mediterranean birds
PAGE 452
RESEARCH ETHICS
SAFEGUARD TRANSPARENCY
Don't let openness backfire on individuals
PAGE 459
POPULAR SCIENCE
WHEN GENES GOT 'SELFISH'
Dawkins's calling card forty years on
PAGE 462
NATURE.COM/NATURE
28 January 2016 £10
Vol. 529, No. 7587
04
9 770028 083095

In "Nature" 27 January 2016:

- "AlphaGo was not preprogrammed to play Go: rather, it learned using a general-purpose algorithm that allowed it to interpret the game's patterns."
- "...AlphaGo program applied **deep learning** in neural networks (convolutional NN) — brain-inspired programs in which connections between layers of simulated neurons are strengthened through examples and experience."

Deep Learning Today

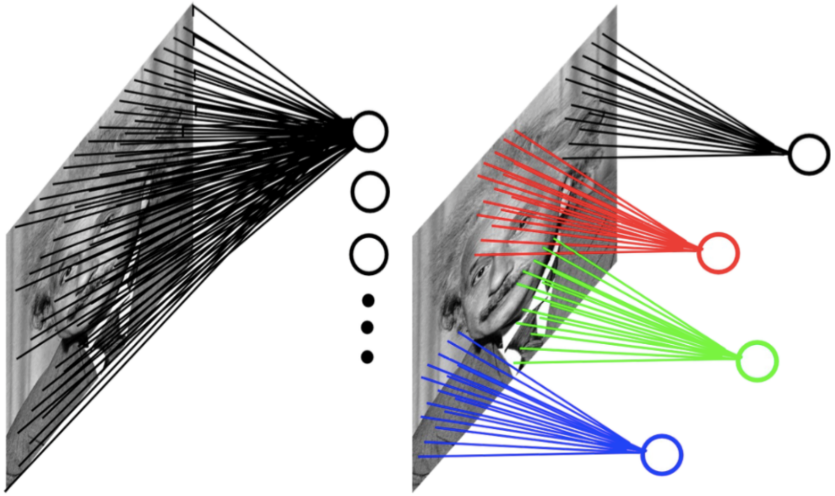
- Advancement in speech recognition
  - A few long-standing performance records were broken with deep learning methods
  - Microsoft and Google have both deployed DL-based speech recognition systems in their products
- Advancement in Computer Vision
  - Feature engineering is the bread-and-butter of a large portion of the CV community, which creates some resistance to feature learning
  - But the record holders on ImageNet and Semantic Segmentation are convolutional nets
- Advancement in Natural Language Processing
  - Fine-grained sentiment analysis, syntactic parsing
  - Language model, machine translation, question answering

Motivations for Deep Architectures

- Insufficient depth can hurt
  - With shallow architecture (SVM, NB, KNN, etc.), the required number of nodes in the graph (i.e. computations, and also number of parameters, when we try to learn the function) may grow very large.
  - Many functions that can be represented efficiently with a deep architecture cannot be represented efficiently with a shallow one.
- The brain has a deep architecture
  - The visual cortex shows a sequence of areas each of which contains a representation of the input, and signals flow from one to the next.
  - Note that representations in the brain are in between dense distributed and purely local: they are **sparse**: about 1% of neurons are active simultaneously in the brain.
- Cognitive processes seem deep
  - Humans organize their ideas and concepts hierarchically.
  - Humans first learn simpler concepts and then compose them to represent more abstract ones.
  - Engineers break-up solutions into multiple levels of abstraction and processing

Convolutional Neural Networks

- Input can have very high dimension. Using a fully-connected neural network would need a large amount of parameters.
- Inspired by the neurophysiological experiments conducted by [Hubel & Wiesel 1962], CNNs are a special type of neural network whose hidden units are only connected to local receptive field. The number of parameters needed by CNNs is much smaller.



Example: 200x200 image

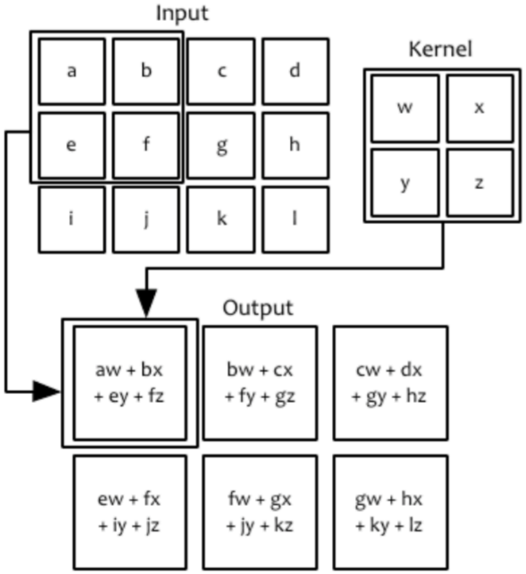
a) fully connected: 40,000 hidden units => 1.6 billion parameters

b) CNN: 5x5 kernel, 100 feature maps => 2,500 parameters

Convolution Operation in CNNs

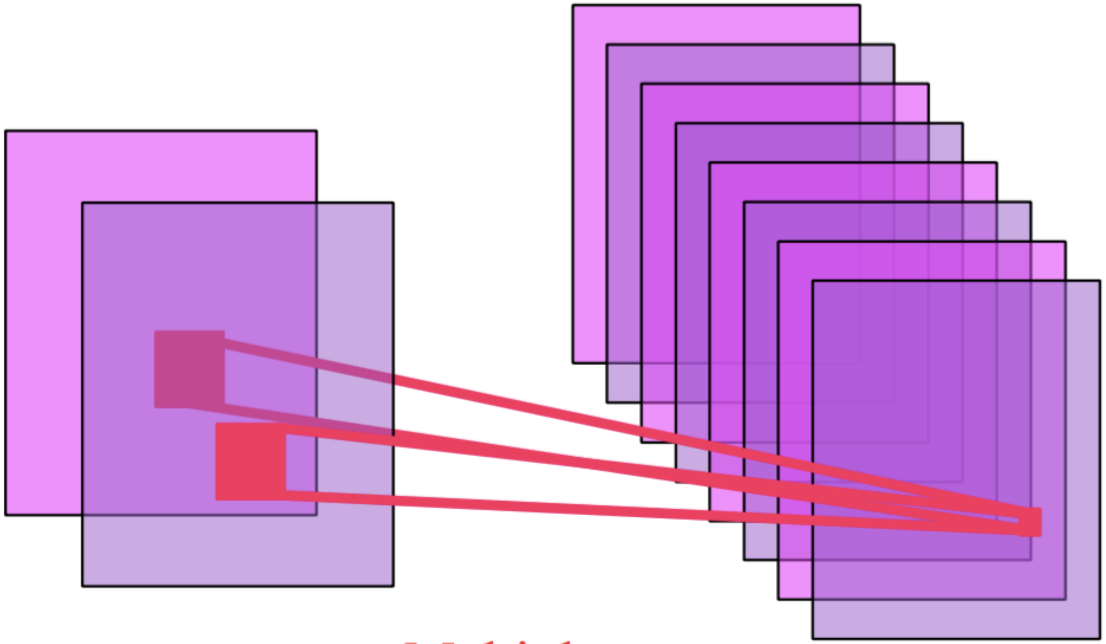
- Input: an image (2-D array) x
- Convolution kernel/operator(2-D array of learnable parameters): w
- Feature map (2-D array of processed data): s
- Convolution operation in 2-D domains:

$$s[i,j] = (x * w)[i,j] = \sum_{m=-M}^{M} \sum_{n=-N}^{N} x[i+m, j+n]\, w[m,n]$$



Multiple Convolutions

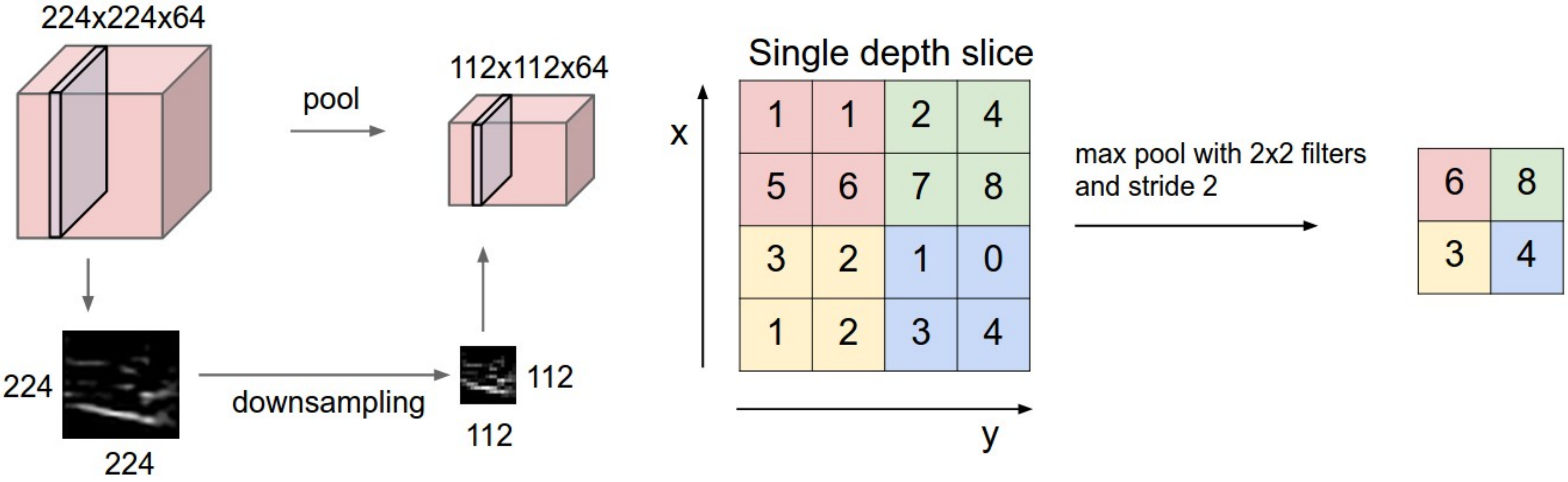
- Usually there are multiple feature maps, one for each convolution operator.



Multiple convolutions

Pooling Layer

- Intuition: to progressively reduce the spatial size of the representation to reduce the amount of parameters and computation in the network, and hence to also control overfitting
- Pooling partitions the input image into a set of non-overlapping rectangles and, for each such sub-region, outputs the maximum value of the features in that region.

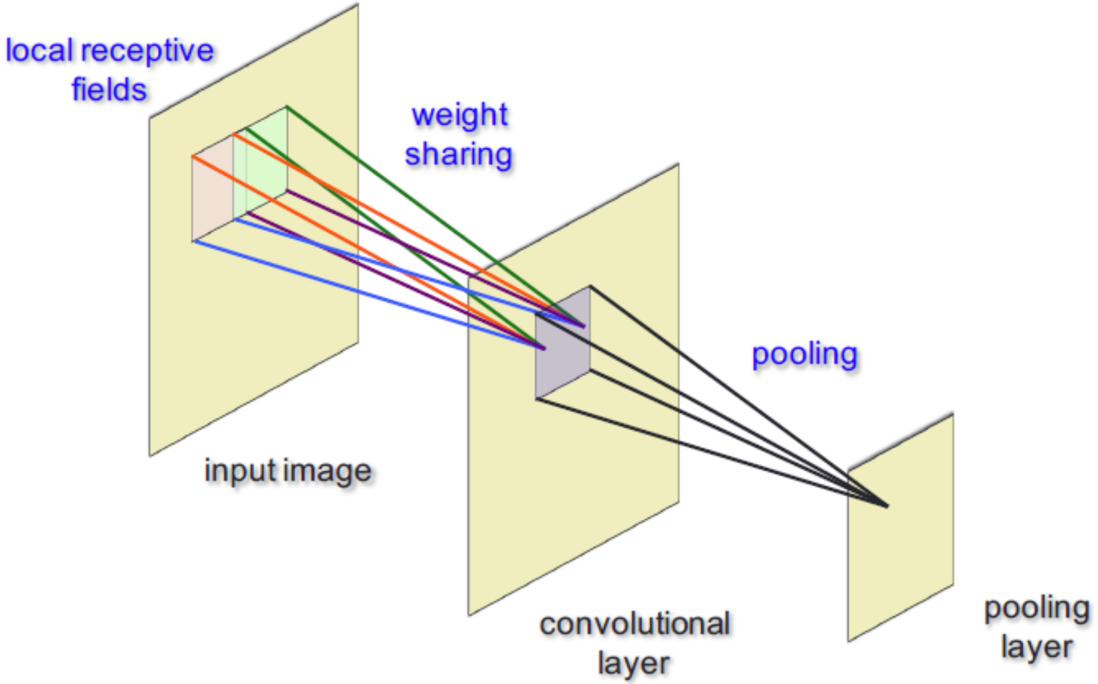


1	1	2	4
5	6	7	8
3	2	1	0
1	2	3	4

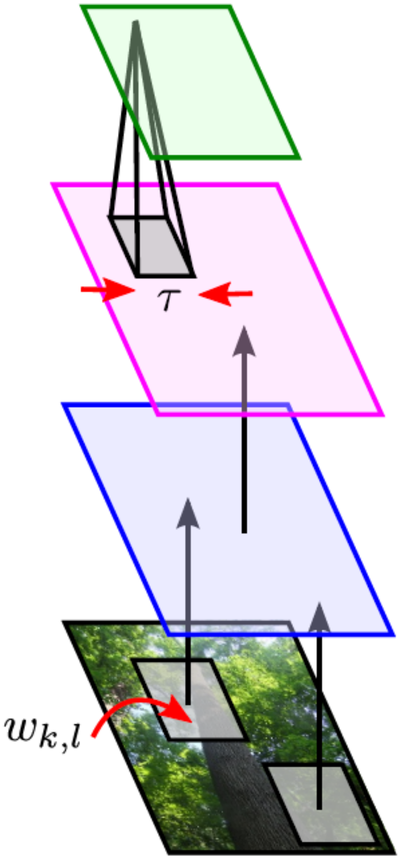
6	8
3	4

Pooling

- Common pooling operations:
  - Max pooling: reports the maximum output within a rectangular neighborhood.
  - Average pooling: reports the average output of a rectangular neighborhood (possibly weighted by the distance from the central pixel).



Deep CNN: Layers, Stages



$$x_{i,j} = \max_{|k|<\tau,|l|<\tau} y_{i-k,j-l}$$

mean or subsample also used

pooling stage

$$y_{i,j} = f(a_{i,j})$$

e.g. $f(a) = [a]_+$

$f(a) = \mathrm{sigmoid}(a)$

non-linear stage

$$a_{i,j} = \sum_{k,l} w_{k,l} z_{i-k,j-l}$$

only parameters

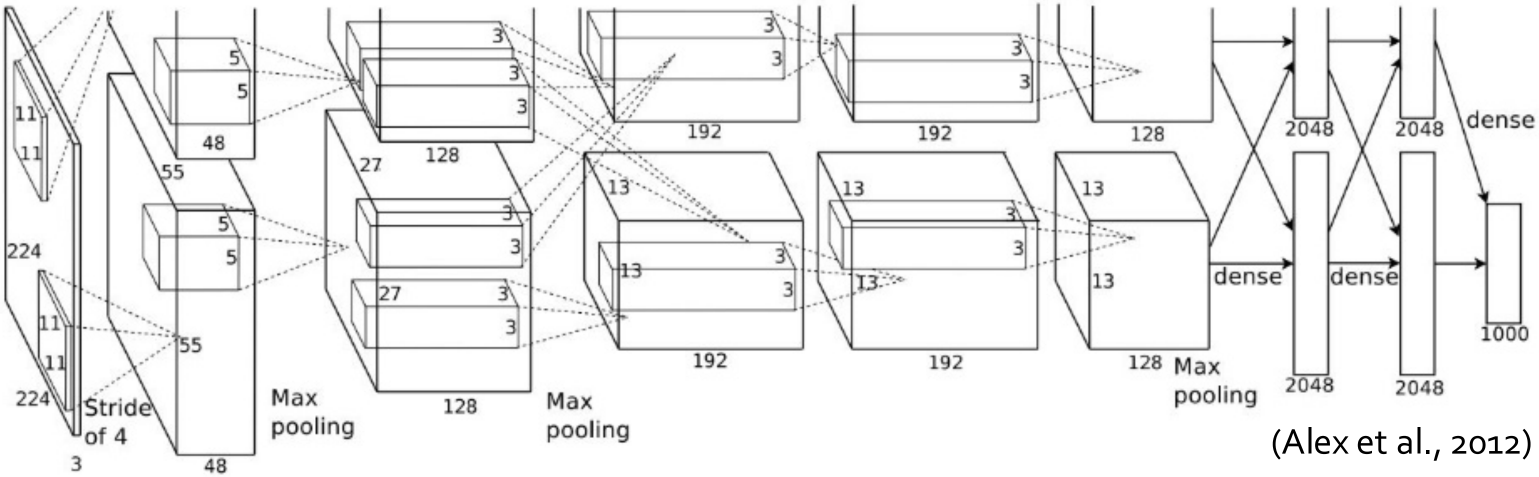
convolutional stage

$z_{i,j}$

input image

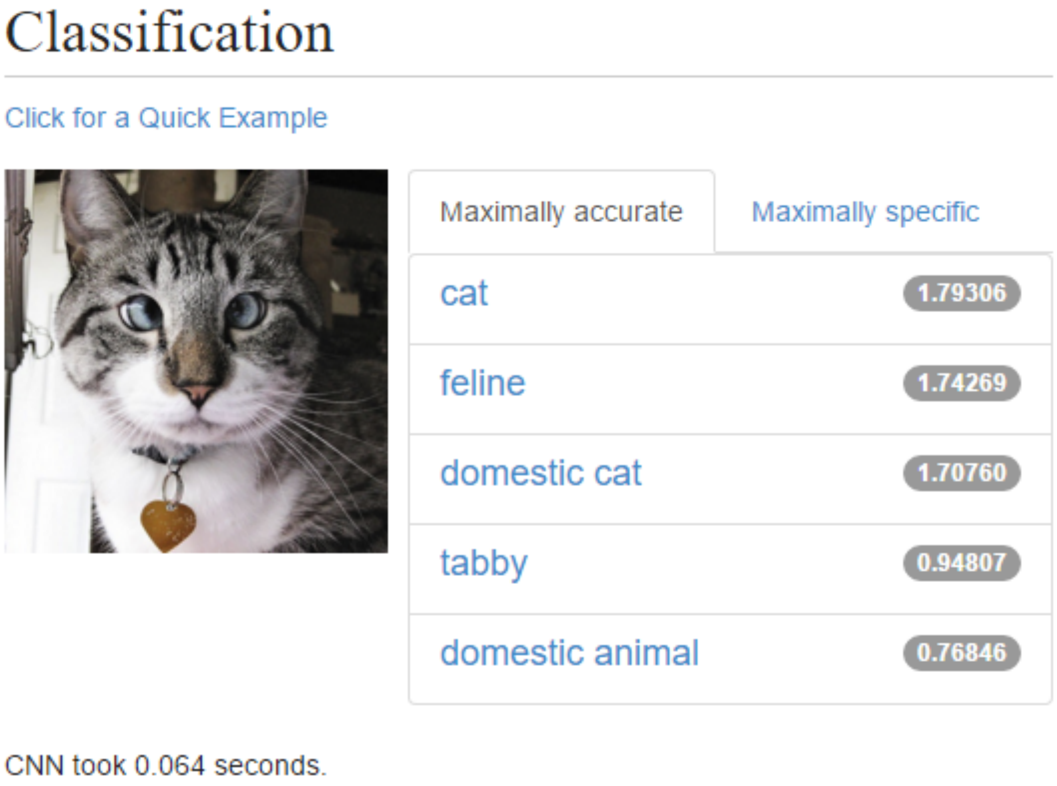
Deep CNN: Winner of ImageNet 2012

- Multiple feature maps per convolutional layer.
- Multiple convolutional layers for extracting features at different levels.
- Higher-level layers take the feature maps in lower-level layers as input.



(Alex et al., 2012)

Deep CNN for Image Classification



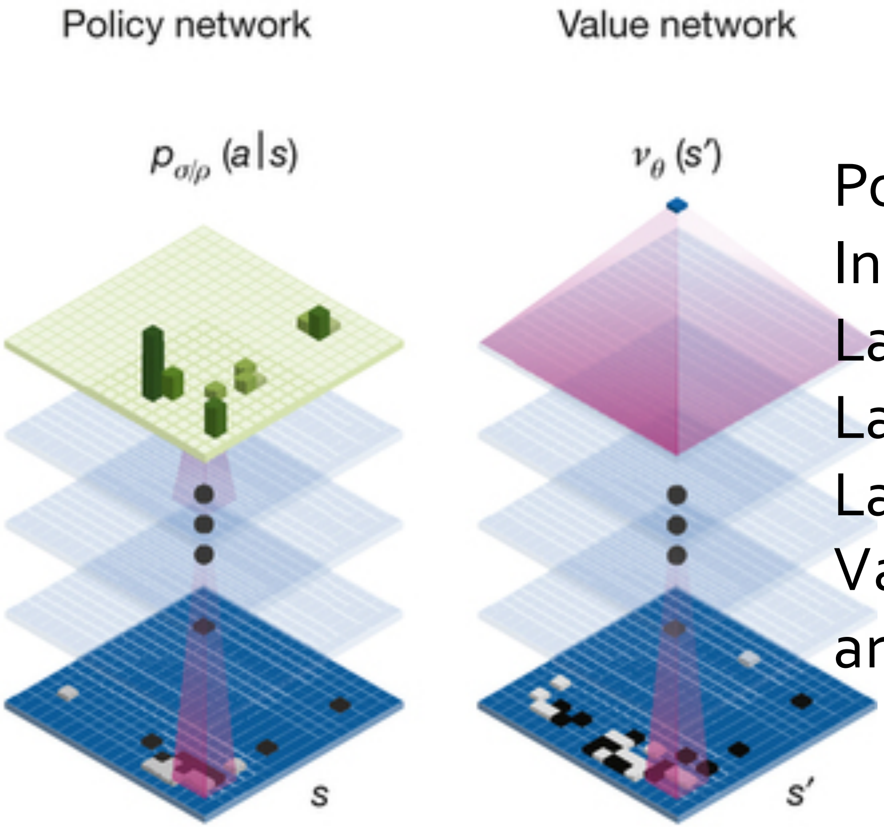
Classification

cat	1.79306
feline	1.74269
domestic cat	1.70760
tabby	0.94807
domestic animal	0.76846

CNN took 0.064 seconds.

Try out a live demo at
http://demo.caffe.berkeleyvision.org/

Deep CNN in AlphaGO



Policy network:
Input: 19x19, 48 input channels
Layer 1: 5x5 kernel, 192 filters
Layer 2 to 12: 3x3 kernel, 192 filters
Layer 13: 1x1 kernel, 1 filter
Value network has similar
architecture to policy network

(Silver et al, 2016)

Other Deep Learning Models

- Recurrent Neural Networks (RNNs)
  - Long Short-Term Memory (LSTM)
- (Deep) Reinforcement Learning
  - Deep Q-networks, Q learning
  - Policy-based
  - Value-based
  - Model-based

References

- Antol, S., Agrawal, A., Lu, J., Mitchell, M., Batra, D., Lawrence Zitnick, C., & Parikh, D. (2015). VQA: Visual question answering. In Proceedings of the IEEE International Conference on Computer Vision (pp. 2425-2433).
- Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. Science, 313(5786), 504-507.
- Schuster, M., & Paliwal, K. K. (1997). Bidirectional recurrent neural networks. Signal Processing, IEEE Transactions on, 45(11), 2673-2681.
- Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems (pp. 1097-1105).
- Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural computation, 9(8), 1735-1780.
- Mnih, Volodymyr, et al. "Playing atari with deep reinforcement learning."arXiv preprint arXiv:1312.5602 (2013).
- Mnih, Volodymyr, et al. "Human-level control through deep reinforcement learning." Nature 518.7540 (2015): 529-533.
- Silver, David, et al. "Mastering the game of Go with deep neural networks and tree search." Nature 529.7587 (2016): 484-489.
- Silver, D. (2015). Deep Reinforcement Learning [Powerpoint slides]. Retrieve from http://www.iclr.cc/lib/exe/fetch.php?media=iclr2015:silver-iclr2015.pdf
- Lecun, Y., & Ranzato, M. (2013). Deep Learning Tutorial [Powerpoint slides]. Retrieved from http://www.cs.nyu.edu/~yann/talks/lecun-ranzato-icml2013.pdf