Deep Learning

Volker Tresp
Summer 2016

Scientists See Promise in Deep-Learning Programs

Hao Zhang/The New York Times

A voice recognition program translated a speech given by Richard F. Rashid, Microsoft's top scientist, into Mandarin Chinese.

By JOHN MARKOFF
Published: November 23, 2012

Using an artificial intelligence technique inspired by theories about how the brain recognizes patterns, technology companies are reporting startling gains in fields as diverse as computer vision, speech recognition and the identification of promising new molecules for designing drugs.

The advances have led to widespread enthusiasm among researchers who design software to perform human activities like seeing, listening and thinking. They offer the promise of machines that converse with humans and perform tasks like driving cars and working in factories, raising the specter of automated robots that could replace human workers.

点击查看本文中文版。

Keith Penner

A student team led by the computer scientist Geoffrey E. Hinton used deep-learning technology to design software.

The technology, called deep learning, has already been put to use in services like Apple's Siri virtual personal assistant, which is based on Nuance Communications' speech recognition service, and in Google's Street View, which uses machine vision to identify specific addresses.

But what is new in recent months is the growing speed and accuracy of deep-learning programs, often called artificial neural networks or just "neural nets" for their resemblance to the neural connections in the brain.

"There has been a number of stunning new results with deep-learning methods," said Yann LeCun, a computer scientist at New York University who did pioneering research in handwriting recognition at Bell Laboratories. "The kind of jump we are seeing in the accuracy of these systems is very rare indeed."

Artificial intelligence researchers are acutely aware of the dangers of being overly optimistic. Their field has long been plagued by outbursts of misplaced enthusiasm followed by equally striking declines.

In the 1960s, some computer scientists believed that a workable artificial intelligence system was just 10 years away. In the 1980s, a wave of commercial start-ups collapsed, leading to what some people called the "A.I. winter."

But recent achievements have impressed a wide spectrum of computer experts. In October, for example, a team of graduate students studying with the University of Toronto computer scientist Geoffrey E. Hinton won the top prize in a contest sponsored by Merck to design software to help find molecules that might lead to new drugs.

From a data set describing the chemical structure of thousands of different molecules, they used deep-learning software to determine which molecule was most likely to be an effective drug agent.

The achievement was particularly impressive because the team decided to enter the contest at the last minute and designed its software with no specific knowledge about how the molecules bind to their targets. The students were also working with a relatively small set of data; neural nets typically perform well only with very large ones.

"This is a really breathtaking result because it is the first time that deep learning won, and more significantly it won on a data set that it wouldn't have been expected to win at," said Anthony Goldbloom, chief executive and founder of Kaggle, a company that organizes

Scientists See Promise in Deep-Learning Programs

Published: November 23, 2012

This summer, Jeff Dean, a Google technical fellow, and Andrew Y. Ng, a Stanford computer scientist, programmed a cluster of 16,000 computers to train itself to automatically recognize images in a library of 14 million pictures of 20,000 different objects. Although the accuracy rate was low — 15.8 percent — the system did 70 percent better than the most advanced previous one.

点击查看本文中文版。

Deep learning was given a particularly audacious display at a conference last month in Tianjin, China, when Richard F. Rashid, Microsoft's top scientist, gave a lecture in a cavernous auditorium while a computer program recognized his words and simultaneously displayed them in English on a large screen above his head.

Then, in a demonstration that led to stunned applause, he paused after each sentence and the words were translated into Mandarin Chinese characters, accompanied by a simulation of his own voice in that language, which Dr. Rashid has never spoken.

The feat was made possible, in part, by deep-learning techniques that have spurred improvements in the accuracy of speech recognition.

Then, in a demonstration that led to stunned applause, he paused after each sentence and the words were translated into Mandarin Chinese characters, accompanied by a simulation of his own voice in that language, which Dr. Rashid has never spoken.

The feat was made possible, in part, by deep-learning techniques that have spurred improvements in the accuracy of speech recognition.

Dr. Rashid, who oversees Microsoft's worldwide research organization, acknowledged that while his company's new speech recognition software made 30 percent fewer errors than previous models, it was "still far from perfect."

"Rather than having one word in four or five incorrect, now the error rate is one word in seven or eight," he wrote on Microsoft's Web site. Still, he added that this was "the most dramatic change in accuracy" since 1979, "and as we add more data to the training we believe that we will get even better results."

One of the most striking aspects of the research led by Dr. Hinton is that it has taken place largely without the patent restrictions and bitter infighting over intellectual property that characterize high-technology fields.

"We decided early on not to make money out of this, but just to sort of spread it to infect everybody," he said. "These companies are terribly pleased with this."

Referring to the rapid deep-learning advances made possible by greater computing power, and especially the rise of graphics processors, he added:

"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."

Baidu muscles in on Google's turf with Silicon Valley deep learning lab

Chinese search giant beds down next to Apple in Cupertino

By **Phil Muncaster** • **Get more from this author**

Posted in Business, 15th April 2013 06:00 GMT

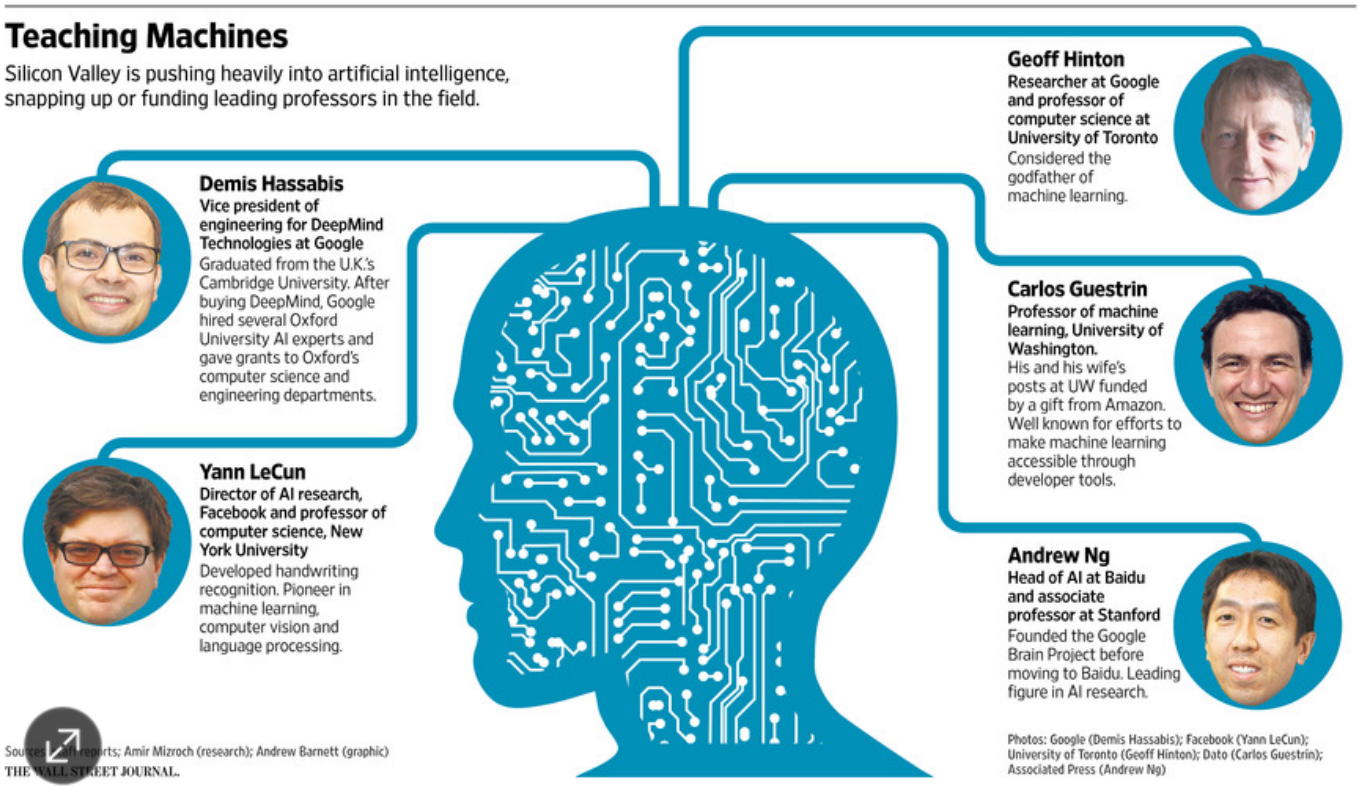
Chinese search giant Baidu has opened the doors to a new research facility in Google's back yard where it's hoping to tap the local talent to consolidate early mover advantage in the burgeoning field of "deep learning".

The Cupertino-based Institute of Deep Learning (IDL) is the Silicon Valley counterpart of another facility back in China dedicated to accelerating research in the emerging machine learning-related discipline.

TECH

Artificial-Intelligence Experts Are in High Demand

Tech firms, universities stock research centers amid push in hot area of computer science



By AMIR MIZROCH

Updated May 1, 2015 5:41 a.m. ET

15 COMMENTS

When the University of Washington's computer-science department wanted to poach artificial-intelligence expert Carlos Guestrin from Carnegie Mellon, it turned to

Neural Network Winter and Revival

- While Machine Learning was flourishing, there was a Neural Network winter (late 1990's until late 2000's)
- Around 2010 there was a revival which made neural networks again extremely popular; it was restarted by Geoffrey Hinton, Yann LeCun, and Yoshua Bengio
- Yann LeCun (New York University and Facebook Artificial Intelligence Research) and Yoshua Bengio (Université de Montréal) are two world-class researchers who never stopped working with neural networks. Geoffrey Hinton (co-inventor of the MLP, Boltzmann machines and others) (Google and University of Toronto) says it all got re-started with the 2006 paper "A fast learning algorithm for deep belief nets" by Hinton, Osindero, and Teh
- In Europe: Jürgen Schmidhuber at IDSIA
- Deep networks achieved best results on many tasks/datasets

Schmidhuber: "Deep Learning Conspiracy"

Jürgen Schmidhuber

Kai Yu

Yu Kai, head of Baidu's Institute of Deep Learning (IDL), demonstrates the smart bike project, DuBike, at the company's headquarters in Beijing. Photo: Simon Song

What Belongs to Deep Learning

1. In general: NNs with many large hidden layers
2. Any RNN network (last lecture)
3. Convolutional Neural Networks (CNNs)
4. Representation Learning

Deep Learning Recipe (Hinton 2013)

What are the reasons?

1. Take a large data set
2. Take a Neuronal Network with many (e.g., 7) large (z.B. 1000 nodes/layer) layers
3. Optional: Use GPUs
4. Train with Stochastic Gradient Decent (SGD)
5. Except for the output layer use *rectified linear units*: $\max(0, h)$
6. Regularize with *drop-out*
7. Optional: Initialize weights with unsupervised learning
8. If the input is spatial (e.g., a picture), use convolutional networks (*weight sharing*) with *max-pooling*

Important Benefits

- A deep network learns complex application-specific features
- To model complex functions, shallow networks either require many (M_ϕ) basis functions or many (N) kernels. A deep architecture can achieve an efficient representation with fewer ressources in a hierarchical layered structure

1: Large Data Set

- When decision boundaries are complex, a large data set describes the details
- Details can be captured with a complex (multi-layer) neural networks
- 20 million utterances for training the acoustic model for speech recognition
- 10 million random 200x200 pixel thumbnail images taken from YouTube content for training an object recognition system (cat detector)

2: Large Networks

- It has been possible to train small to medium size problems since the early 1990s
- In deep learning people work with really large Neural Networks. Example: 10 layers, 1000 neurons/layer

3: Graphical Processing Units (GPUs)

- GPUs are highly suited for the kind of number crunching, matrix/vector math involved in deep Neural Networks. GPUs have been shown to speed up training algorithms by orders of magnitude
- Their highly parallel structure makes them more effective than general-purpose CPUs for algorithms where processing of large blocks of data is done in parallel
- General-Purpose Computing on Graphics Processing Units (GPGPU) is the utilization of a graphics processing unit (GPU), which typically handles computation only for computer graphics, to perform computation in applications traditionally handled by the central processing unit (CPU)

4: Stochastic Gradient Descent SGD

- Often regular SGD is used where the gradient is calculated on a single training pattern
- "Minibatch SGD" works identically to SGD, except that we more than one training example is used to make each estimate of the gradient
- AdaGrad (adaptive gradient algorithm) is often used for learning rates to be adaptively altered.

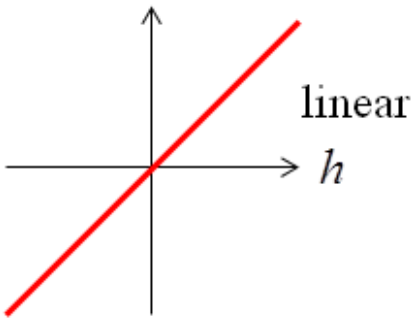
$$w_j := w_j - \frac{\eta}{\sqrt{G_{j,j}}} g_j$$

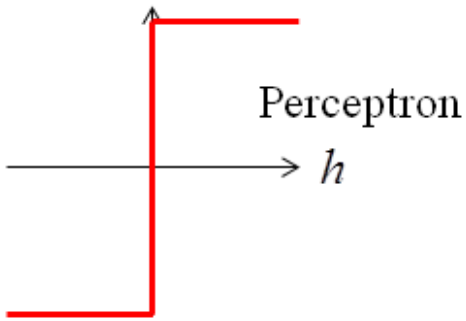
$$G_{j,j} = \sum_{\tau=1}^{t} g_{\tau,j}^2.$$

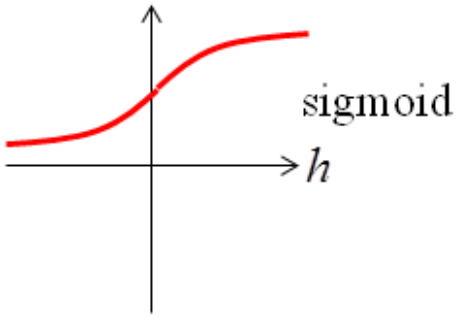
- Extreme parameter updates get dampened, while parameters that get few or small updates receive higher learning rates (g is the notation for the gradient)

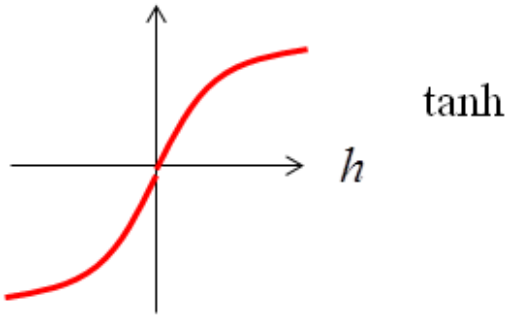
5: Rectified Linear Function

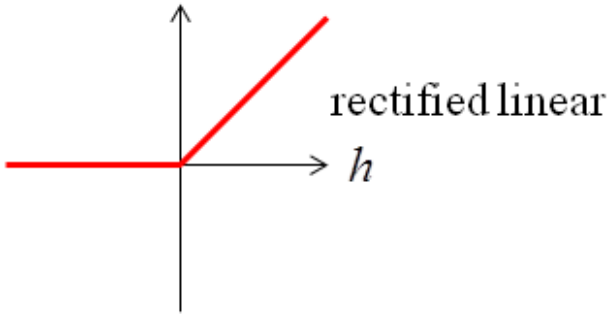
- Rectified Linear Function (ReL) is $max(0, h)$
- Can be motivated in the following way: summing up the response of identical neurons (same input and output weights) where only the threshold/bias is varying. This become similar to a rectified linear neuron
- Reduces the effects of the vanishing gradient problem with sigmoid neurons! They learn much faster!
- Seems odd since some neurons become insensitive to the error, but a sufficient number stays active
- Leads to sparse gradients and to a sparse solution
- For training classification tasks, the output layer has sigmoidal activation functions and the cross-entropy cost function is used











common neural tranfer functions

6A: Drop-Out Regularization

- For each training instance: first remove 50% of all hidden units, randomly chosen. Only calculate error and do adaptation on the remaining network
- For testing (prediction): use all hidden units but multiply all outgoing weights by $1/2$ (gives you same expectation but no variance)
- This is like a committee machine, where each architecture is a committee member, but committee member share weights. It supposedly works like calculating the geometric mean: average the log of the predictions (and then take the exponential over the average)
- Works better than stopped learning! No stopping rule required!
- Can even do drop-out in the input layer, thus different committee members see different inputs!
- Hinton: *use a large enough neural network so that it overfits on your data and then regularize using drop out*

6B: Weight Regularization

- Weight decay works
- But even better: for each neuron, normalize the incoming weight vector to have the same maximum length. Thus if $\|\mathbf{w}\| > \alpha$

$$\mathbf{w} \rightarrow \alpha \frac{1}{\|\mathbf{w}\|} \mathbf{w}$$

7: Initialize Weights with Unsupervised Learning

- Auto-Encoder
- Restricted Boltzmann Machine (RBM) for Deep Boltzmann Machines (DBMs) and Deep Belief Networks (DBNs)

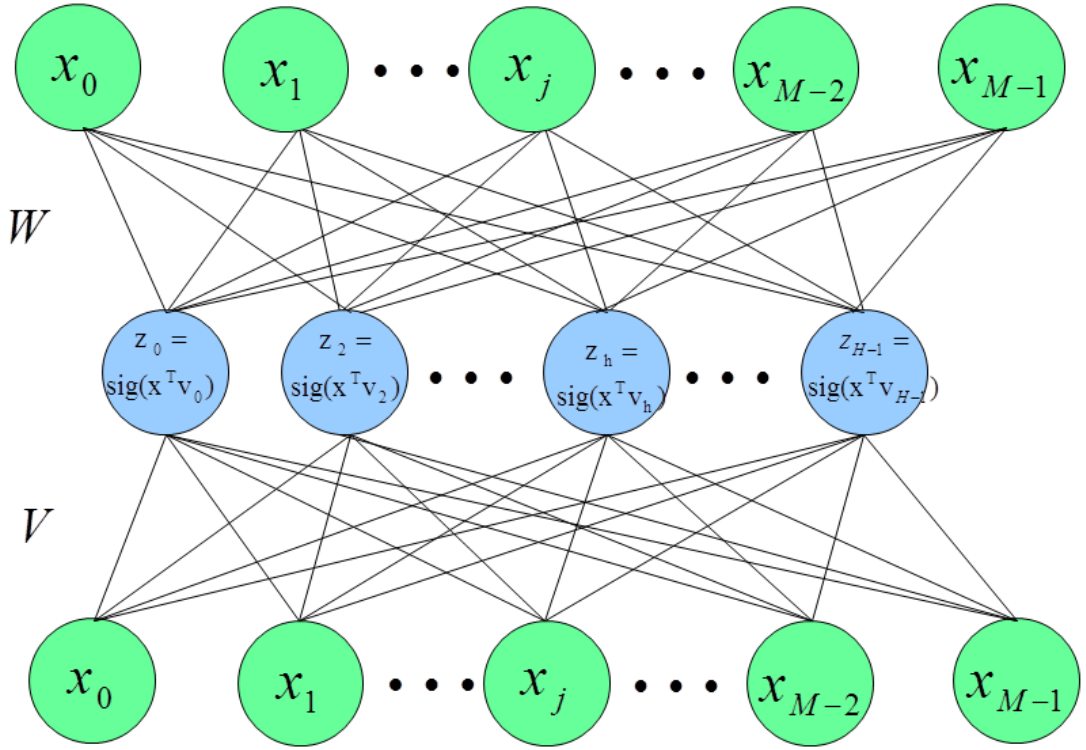
Auto-Encoder

- As the term auto encoder indicates, the goal is to learn the identity $\mathbf{y} = \mathbf{x}$ (M-dimensional vectors)

$$NN(\mathbf{x}) \rightarrow \mathbf{x}$$

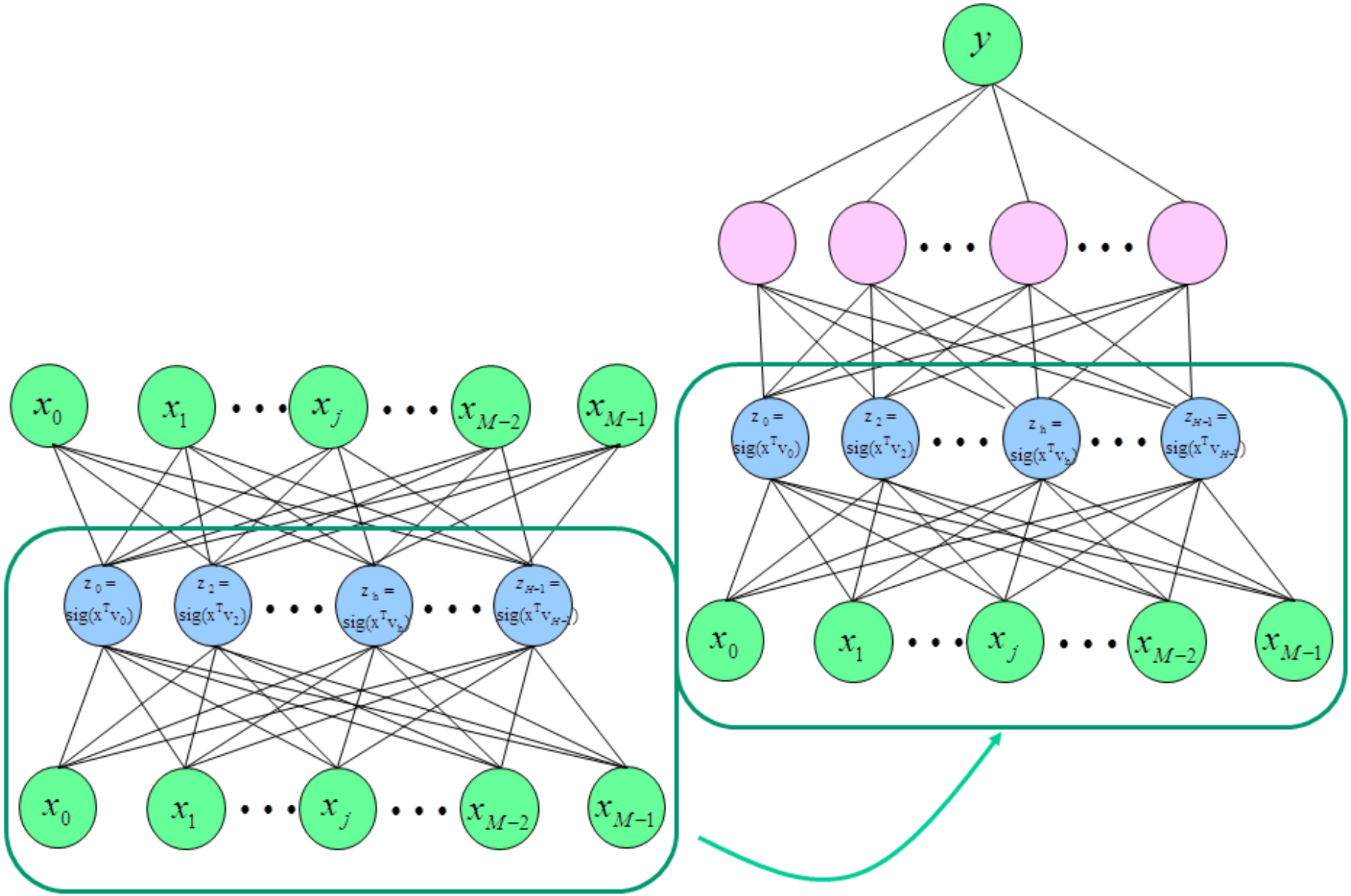
- The constraint is that the number of hidden units is smaller than M, thus a perfect reconstruction becomes impossible; the output of the hidden layer is considered a *representation* or an *embedding* of the input vector (representation learning, embedding learning)
- The output of an auto-encoder layer is the hidden representation $\mathbf{z}$ (see figure)
- The linear equivalent would be a Principal Component Analysis (PCA), although the auto encoder does not require orthonormality and finds a representation in between a component analysis and a cluster analysis

Auto-Encoder (Bottleneck Neural Network)



Auto-Encoder in Neural Networks

- Consider that one has available a large number of unlabelled training data $x_1, x_2, \ldots, x_U$ and a much smaller number of labelled data $(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)$ with $N << U$
- Then the unlabelled data can be used to train the auto encoder. The input-to-hidden layer is then copied into the network for predicting y. Then the complete network is adapted using backprop. Thus the auto encoder provides a clever initialization of the weights
- The auto-encoder can also be used in additional hidden layers in the neural network (Stacked Denoising Autoencoders (SdA))


y
x_0
x_1
x_j
x_{M-2}
x_{M-1}
$z_0 = \text{sig}(x^T v_0)$
$z_2 = \text{sig}(x^T v_2)$
$z_h = \text{sig}(x^T v_h)$
$z_{H-1} = \text{sig}(x^T v_{H-1})$

Denoising Autoencoder

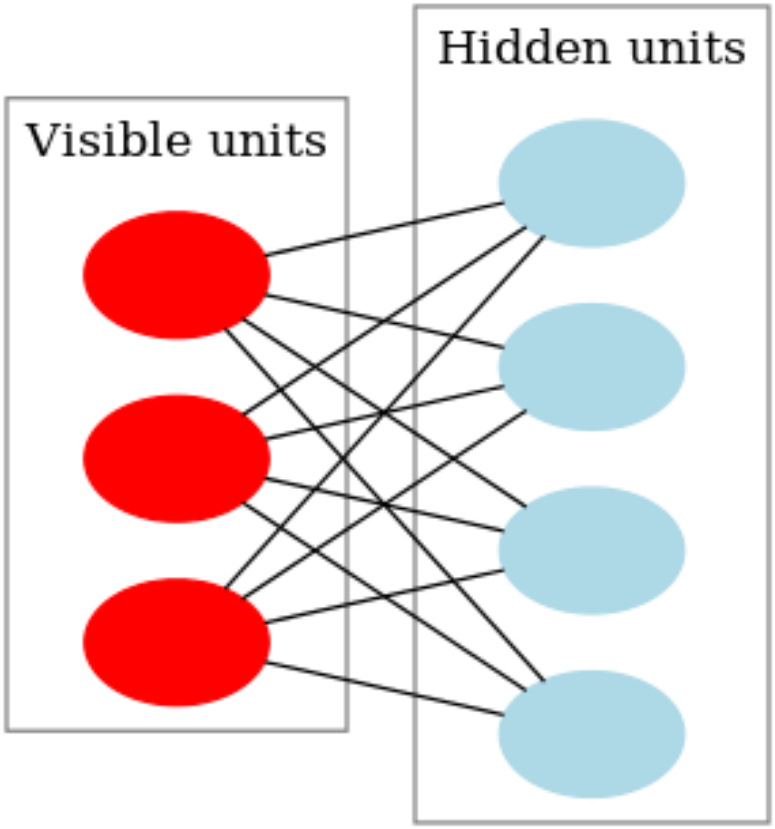
- $$NN(\mathbf{x} + noise) \rightarrow \mathbf{x}$$

Restricted Boltzmann Machines (RBMs), Deep Belief Networks (DBNs), Deep Boltzmann Machines (DBMs)

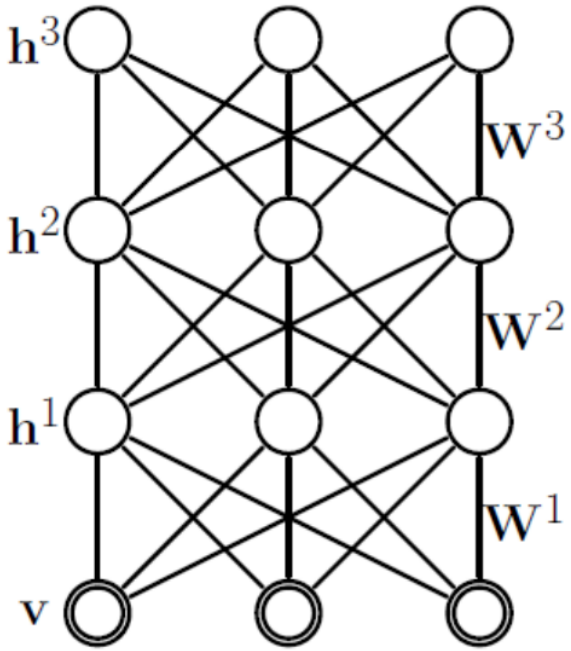
- A restricted Boltzmann machine (RBM) is a generative stochastic neural network that can learn a probability distribution over its set of inputs, similar to an auto encoder
- As their name implies, RBMs are a variant of **Boltzmann machines**, with the **restriction** that their neurons must form a bipartite graph: The inputs are connected only to the hidden units, and the hidden units are only connected to the input units (weights are symmetrical: $w_{i,j} = w_{j,i}$)
- Thus, as the auto encoder, the RBM learns a latent representation of the input vectors. But the number of latent components can be larger than the number of inputs, so the latent representation found is a combination of a component analysis and a cluster analysis. Training is performed with the contrastive divergence (CD) algorithm.
- The RBM can also be used in several hidden layers by treating the previous hidden layer as data layer. The layered structure is is sometimes referred to as a **Deep Belief Network** (DBN).

- A deep neural network can be initialized as a DBN. After initialization, backprop is applied.
- Variant: Deep Boltzmann Machine (DBM)

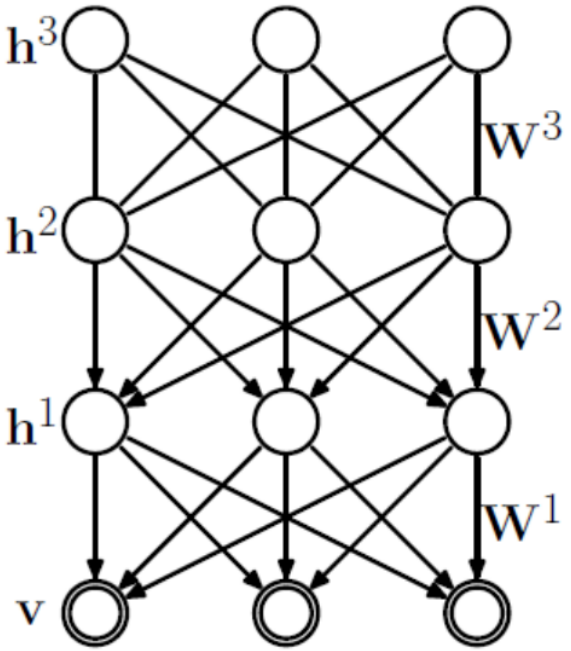
RBM



DBM versus RBN



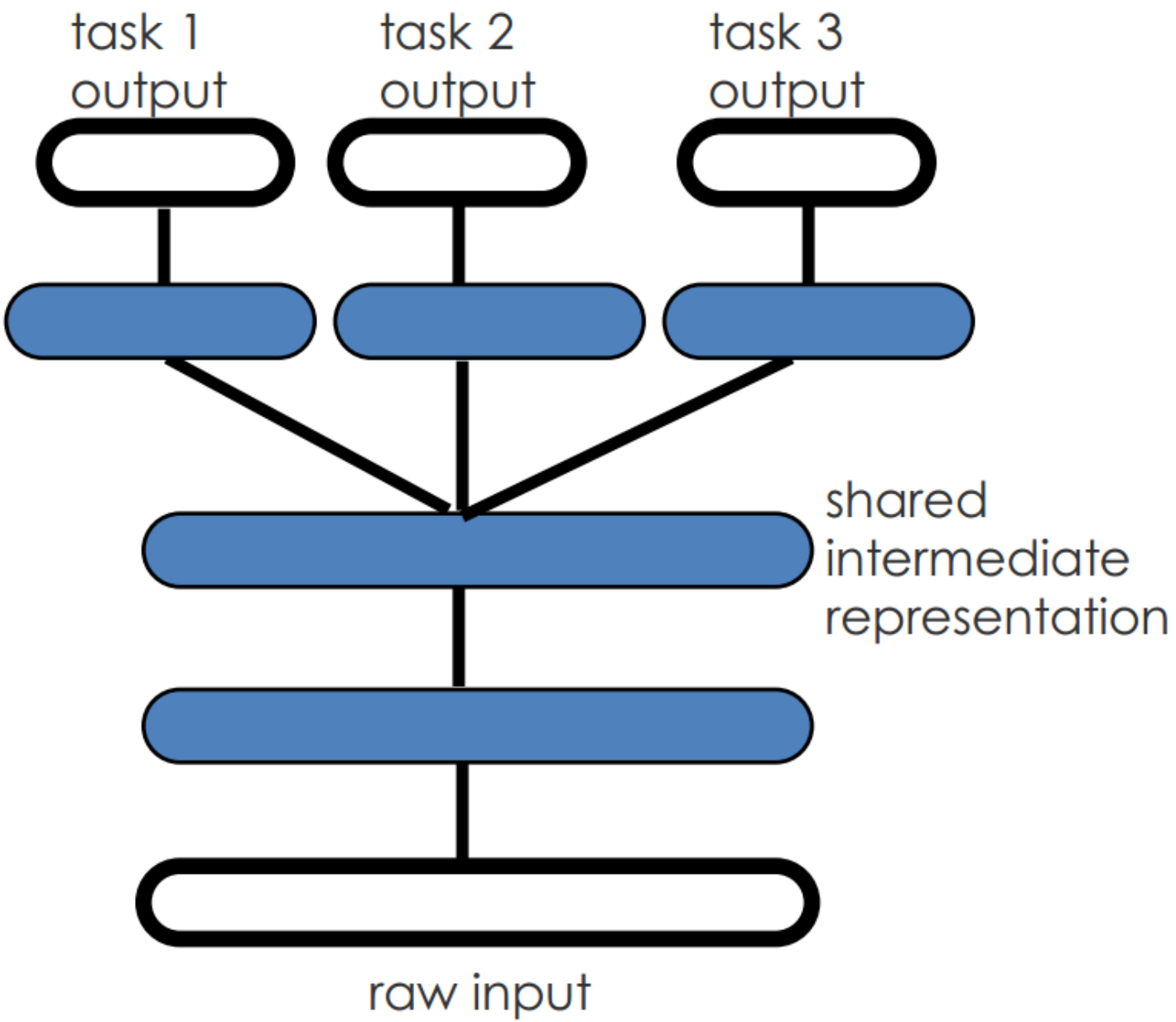
Deep Boltzmann Machine



Deep Belief Network

Multitask Learning

- Recall that in the auto encoder approach, the same representation was used to train both the auto encoder and the classification task. This idea can be generalized in many ways. The idea is to learn several tasks by sharing common representations
- Another advantage is that a new task can be learned much faster!


task 1
output
task 2
output
task 3
output
shared
intermediate
representation
raw input

Facebook's Deep Face: Face Recognition as Multi-Task Learning

- Build a deep learning NN to classify many face images from 4000 persons. Thus there are 4000 outputs, one for each person
- The next to last layer is used as a representation for any face image (also for faces and persons not in the training set)
- How do I use this net for new persons and faces?
- Calculate the distance between a new face and labeled faces from your Facebook friends based on the representation in the next to last layer
- Note that here, the representation is close to the output layer
- Much effort is spent in the input layers to normalize the facial images
- C: convolutional layer. M: max-pooling layer. The subsequent layers (L4, L5 and L6) are locally connected, like a convolutional layer they apply a filter bank, but every location in the feature map learns a different set of filters.

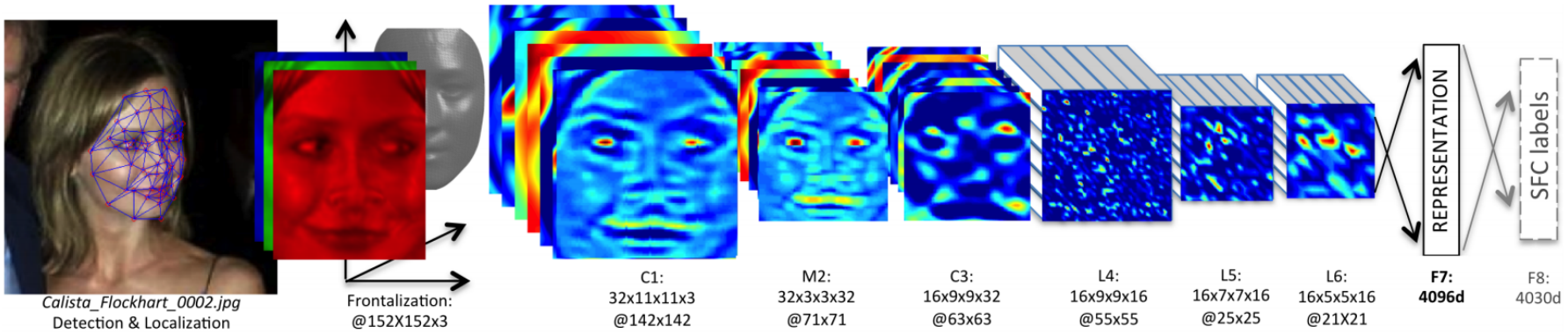
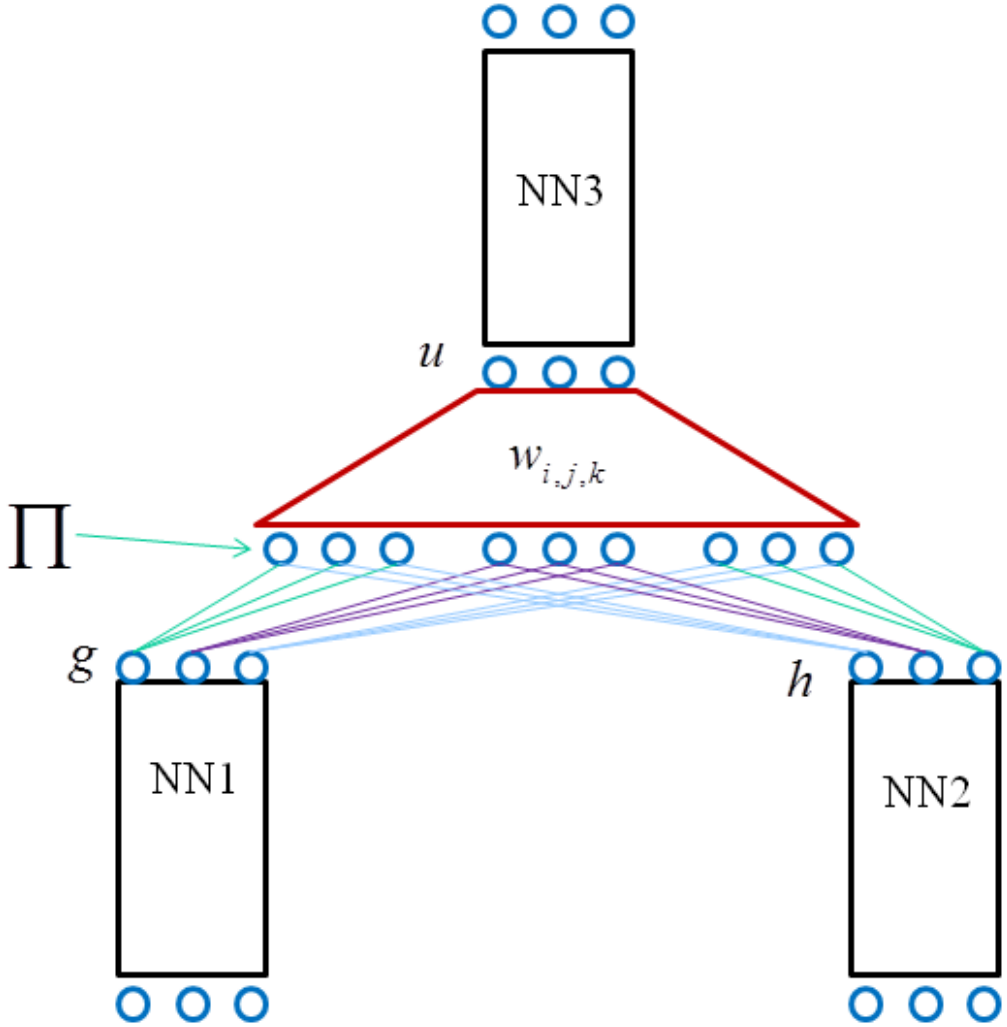


Figure 2. **Outline of the *DeepFace* architecture**. A front-end of a single convolution-pooling-convolution filtering on the rectified input, followed by three locally-connected layers and two fully-connected layers. Colors illustrate feature maps produced at each layer. The net includes more than 120 million parameters, where more than 95% come from the local and fully connected layers.

Multiplicative Couplings for Joining Information Sources

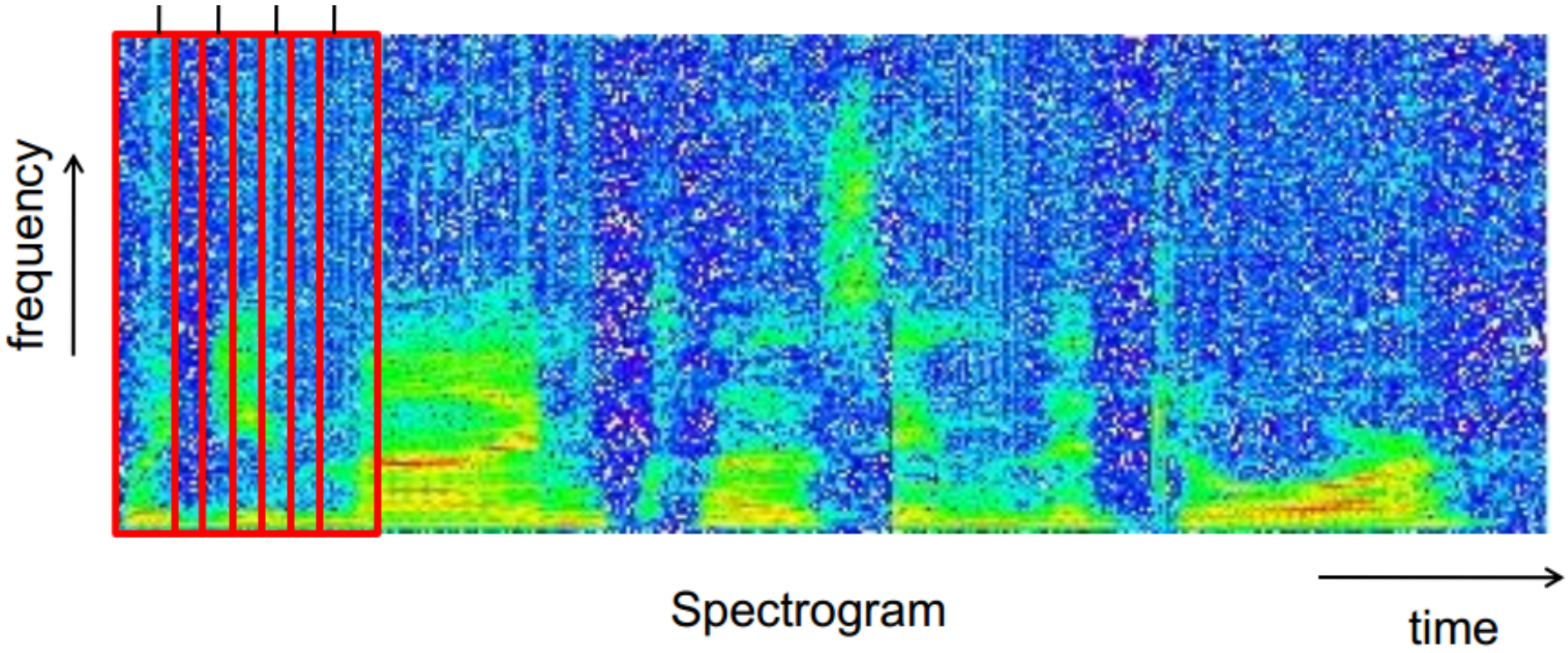
- Given two latent vector representations g and h. Let u be the next higher hidden layer
- The coupling is assumed multiplicative
- $u_i = \sum_j \sum_k w_{i,j,k}\, g_j h_k$
- The weights can be presented as a three-way tensor (note, that the weight has three indices) and the operations then be written as a product of a tensor with the two vectors


NN3
u
$w_{i,j,k}$
Π
g
h
NN1
NN2

Android Server Architecture for Speech Recognition (2013)

- Part of speech recognition with Hidden Markov Models (HMMs): predict a state in the HMM (State) using a frequency representation of the acoustic signal in a time window (Frame)
- The Neural Network is trained to learn $P(State|Frame)$
- 4-10 layers, 1000-3000 nodes / layer, no pre-training
- Rectified linear activations: y=max(0,x)
- Full connectivity between layers,
- Softmax output (cross-entropy cost function) (see lecture on linear classifiers)
- Features:
  - 25ms window of audio, extracted every 10ms.
  - log-energy of 40 Mel-scale filterbanks, stacked for 10-30 frames.

- Training time: 2-3 weeks using GPUs!
- Online: Android uses the server solution. Offline: Small Neural Network on the Smart Phone
- Advantage: Speaker independent! Now used by Google, Microsoft, IBM, replacing Gaussian mixture models (30% reduction in error)
- Even more improvement on the task of object recognition in images (from 26% error to 16% error)) using 1.2 million training images. With convolutional neural networks.


frequency
Spectrogram
time

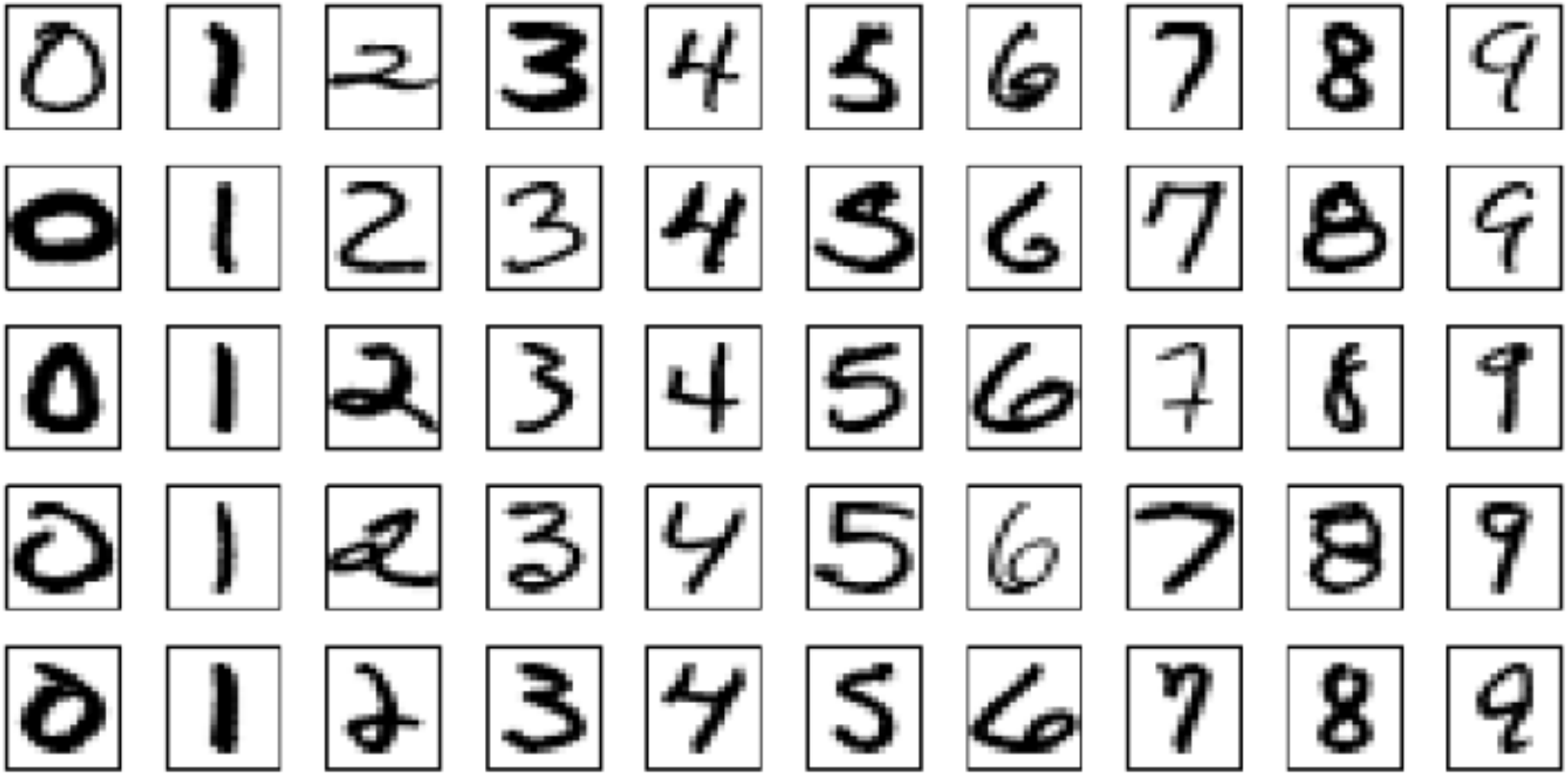
task	Hours of training data	Deep net+HMM	GMM+HMM same data	GMM+HMM more data
Switchboard	309	16.1	23.6	17.1 (2k hours)
English Broadcast news	50	17.5	18.8	
Bing voice search	24	30.4	36.2	
Google voice input	5870	12.3		16.0 (lots more)
Youtube	1400	47.6	52.3	

How many Training Data Points for Deep Learning to Work?

- As of 2015, a rough rule of thumb is that a supervised deep learning algorithm will generally achieve acceptable performance with around 5,000 labeled examples per category, and will match or exceed human performance when trained with a dataset containing at least 10 million labeled examples. Working successfully with datasets smaller than this is an important research area, focusing in particular on how we can take advantage of large quantities of unlabeled examples, with unsupervised or semi-supervised learning.

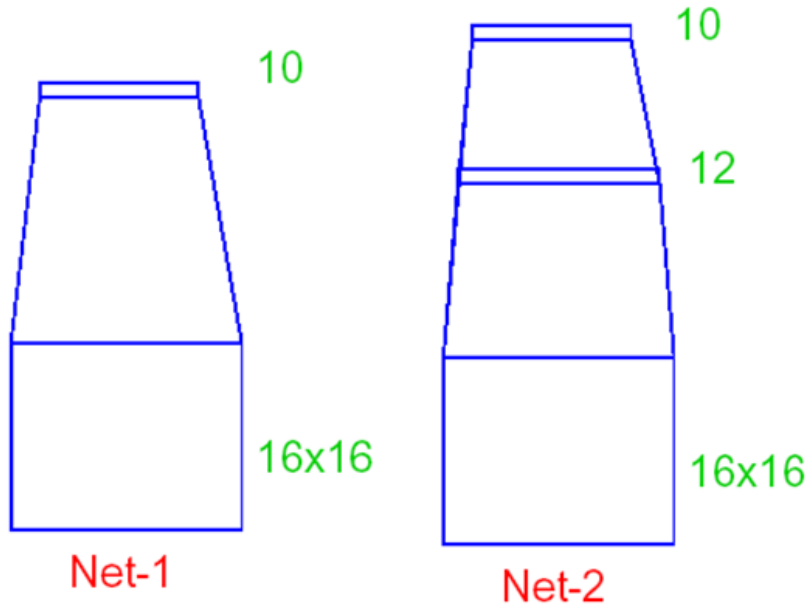
8: Convolutional Neural Networks (CNNs)

Recognition of Handwritten Digits



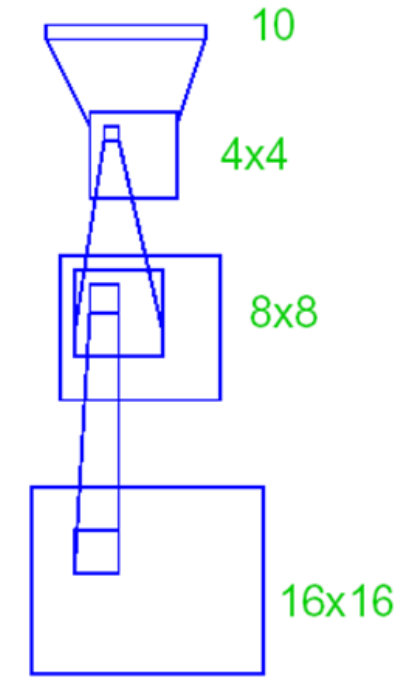
Recognition of Handwritten Digits using Neuronal Networks

- Example: 16×16 grey-valued pictures; 320 training images, 160 test images
- Net-1: No hidden layer: corresponds to 10 Perceptrons, one for each digit
- Net-2: One hidden layer with 12 nodes; fully connected ("normal MLP")


10
16x16
Net-1
10
12
16x16
Net-2

Neuronal Network with local connectivity: Net-3

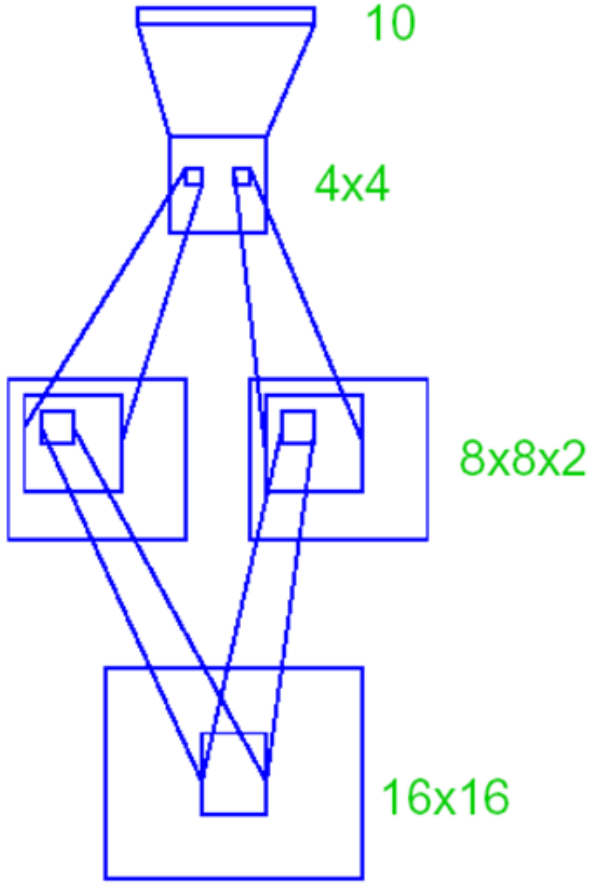
- In the following variants, the complexity was reduced
- Net-3: Two hidden layers with local connectivity (but no weight sharing yet): motivated by the local receptive fields in the brain
  - Each of the 8×8 neurons in the first hiden layer is only connected to 3×3 input neurons from a receptive field
  - In the second hidden layer, each of the 4×4 neurons is connected to 5×5 neurons in the first hidden layer
  - Net-3 has less than 50% of the weights of Net-2, but more neurons



Net-3
Local Connectivity

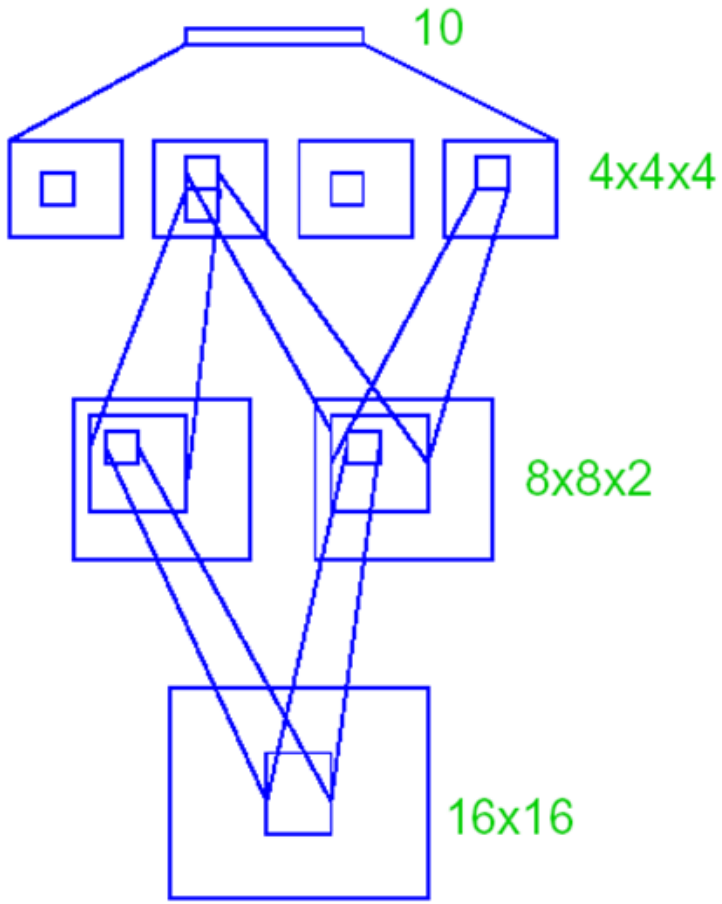
Neuronal Networks with Weight-Sharing (Net-4)

- Net-4: Two hidden layers with local connectivity and *weight-sharing*
- All receptive fields in the left 8×8 block have the same weights; the same is true for all neurons in the right 8×8 block
- The 4×4 block in the second hidden layer, as before


10
4x4
8x8x2
16x16

Neural Networks with Weight Sharing (Net-5)

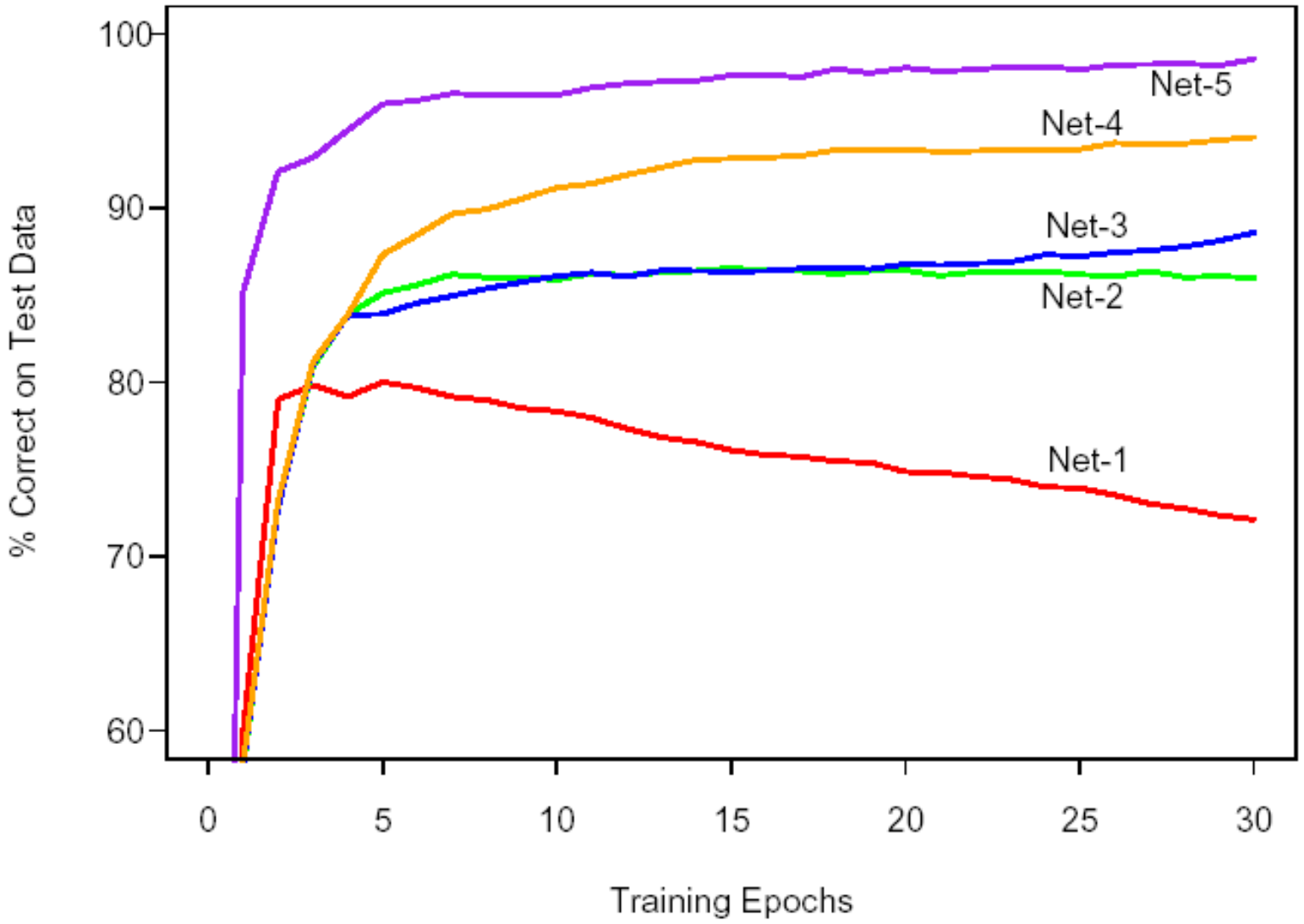
- Net-5: Two hidden layers with local connectivity and two layers of *weight-sharing*



Net-5

Learning Curves

- One training epoch is one pass through all data
- The following figure shoes the performance on the test set
- Net-1: One sees overfitting with increasing epochs
- Net-5: Shows best results without overfitting


100
90
80
70
60
% Correct on Test Data
0
5
10
15
20
25
30
Training Epochs
Net-5
Net-4
Net-3
Net-2
Net-1

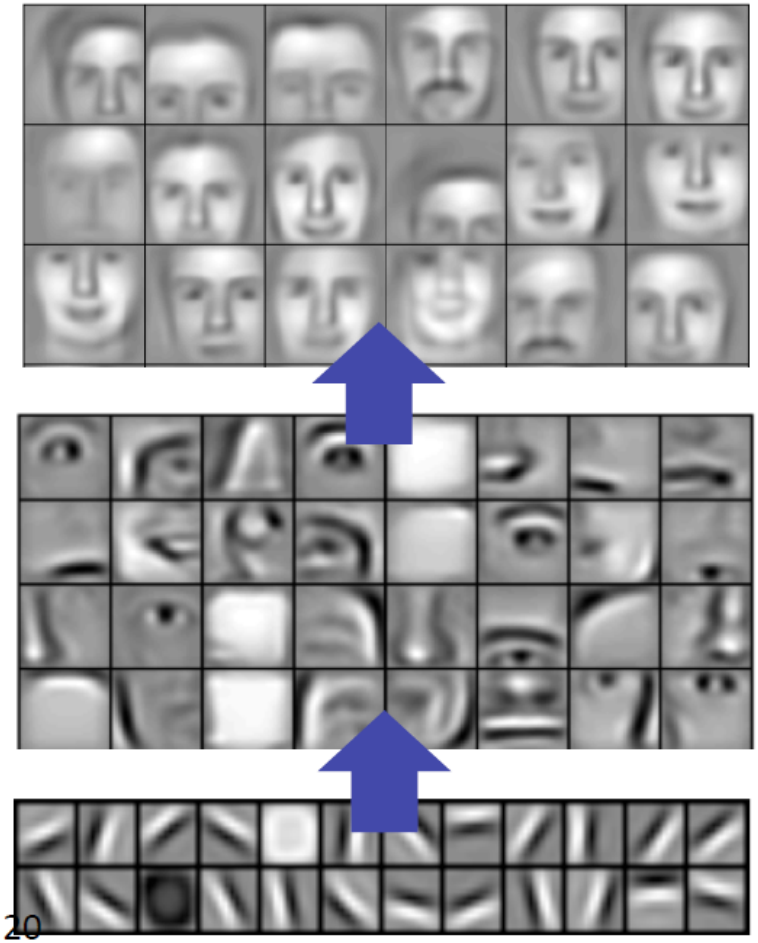
Statistics

- Net-5 has best performance. The number of free parameters (1060) is much smaller than the total number of parameters (5194)

TABLE 11.1. *Test set performance of five different neural networks on a handwritten digit classification example (Le Cun, 1989).*

	Network Architecture	Links	Weights	% Correct
Net-1:	Single layer network	2570	2570	80.0%
Net-2:	Two layer network	3214	3214	87.0%
Net-3:	Locally connected	1226	1226	88.5%
Net-4:	Constrained network 1	2266	1132	94.0%
Net-5:	Constrained network 2	5194	1060	98.4%

Successive model layers learn deeper intermediate representations

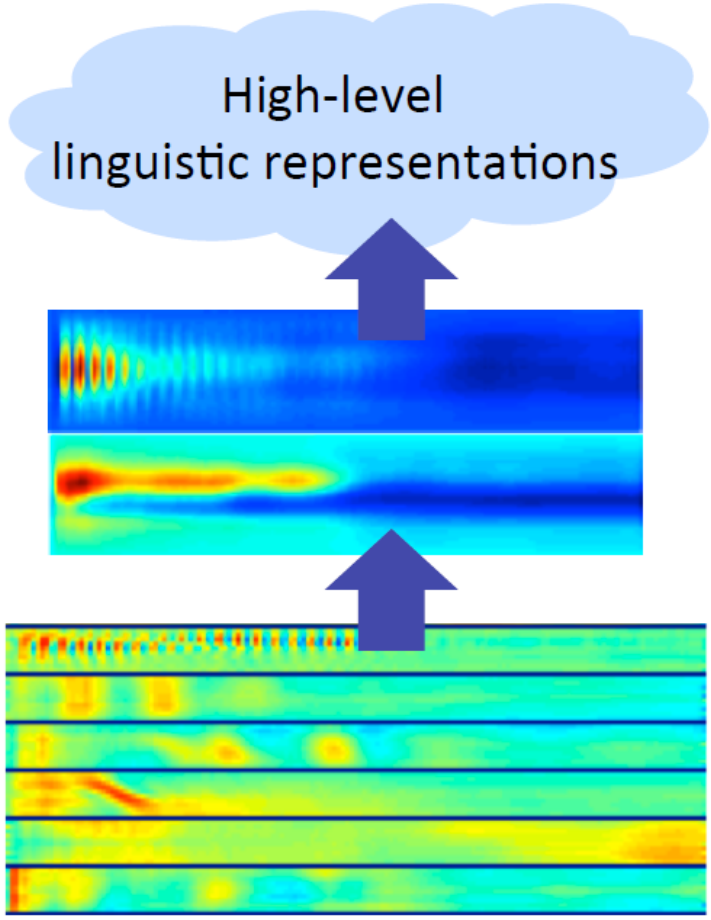


Layer 3

Parts combine
to form objects

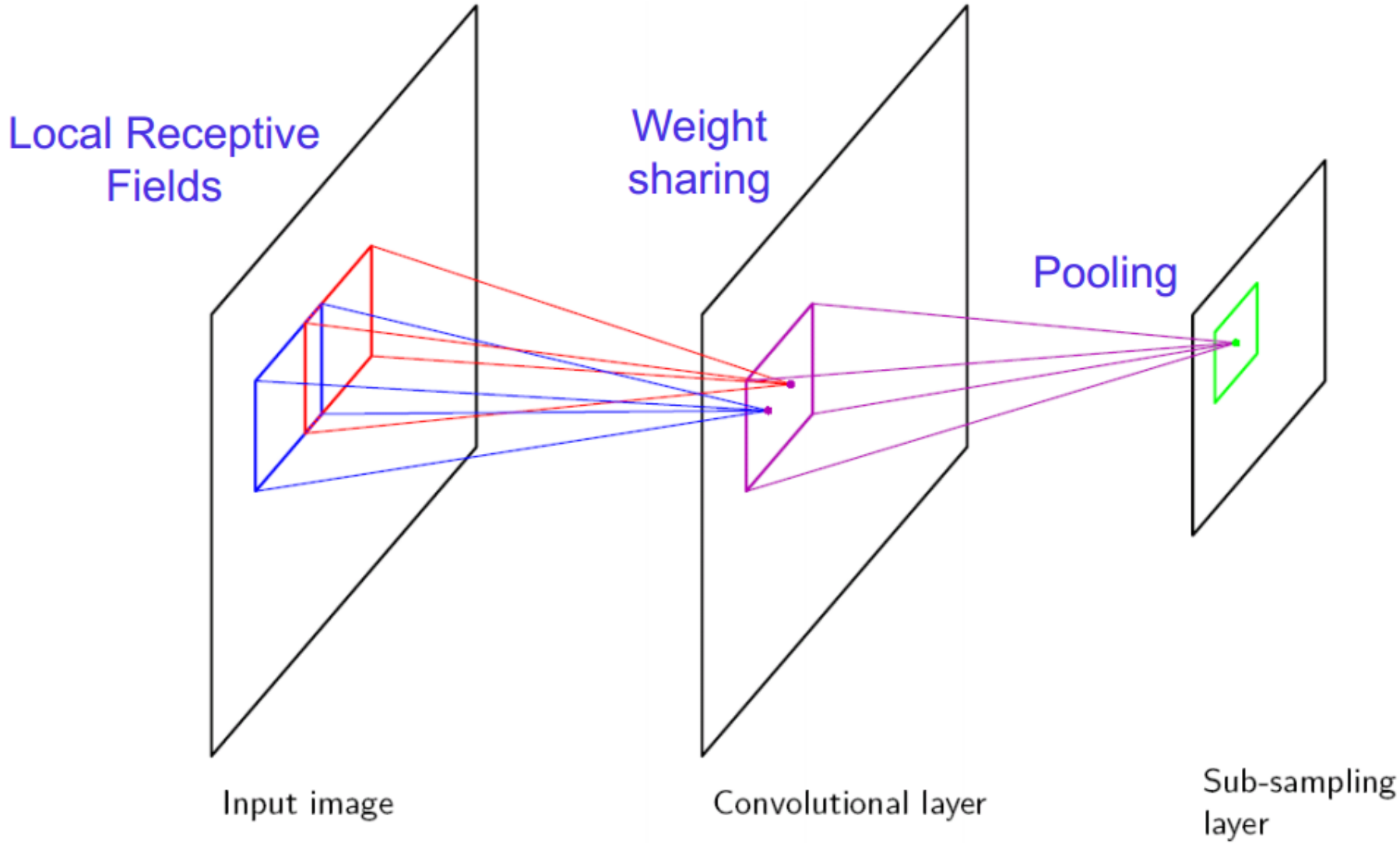
Layer 2

Layer 1



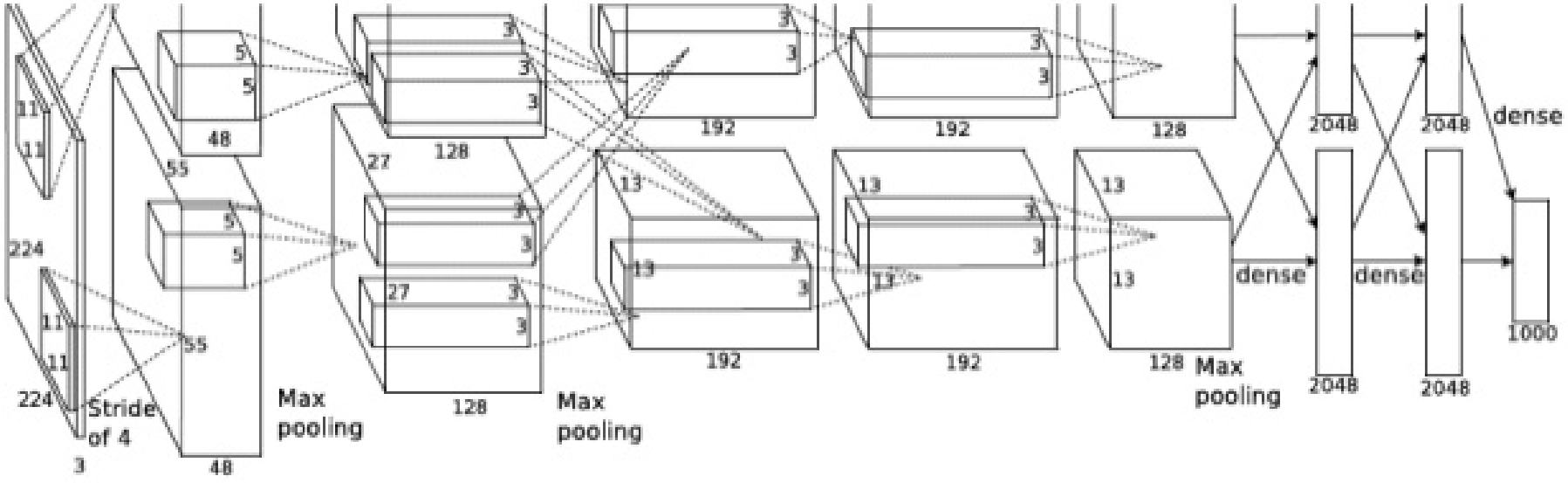
Pooling

- For example, one could compute the mean (or max) value of a **particular feature** over a **region of the image**. These summary statistics are much lower in dimension (compared to using all of the extracted features) and can also improve results (less over-fitting). We aggregation operation is called this operation pooling, or sometimes **mean pooling** or **max pooling** (depending on the pooling operation applied).
- Max-pooling is useful in vision for two reasons: (1) it reduces the computational complexity for upper layers and (2) it provides a form of translation invariance
- Since it provides additional robustness to position, max-pooling is thus a "smart" way of reducing the dimensionality of intermediate representations.


Local Receptive Fields
Weight sharing
Pooling
Input image
Convolutional layer
Sub-sampling layer

AlexNet

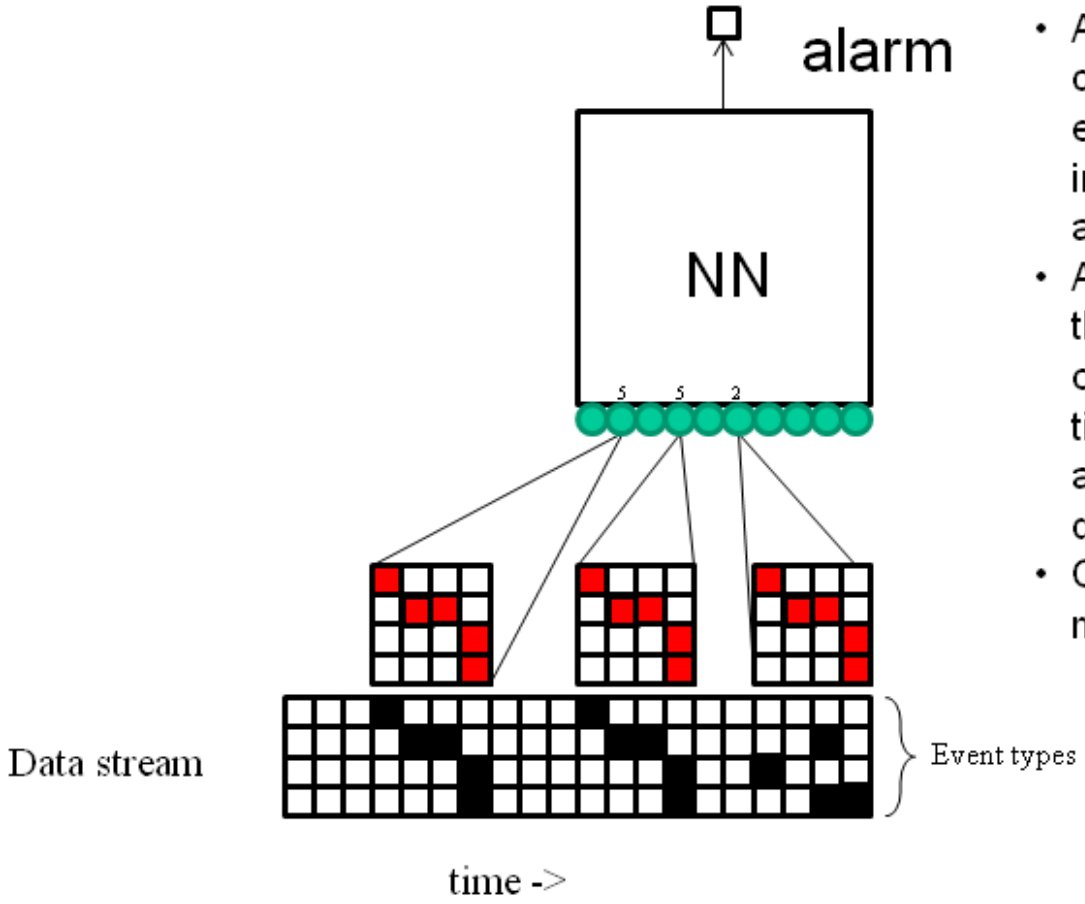
- Similar framework to LeCun'98 but:
  - Bigger model (7 hidden layers, 650,000 units, 60,000,000 params)
  - More data (10^6 vs. 10^3 images)
  - GPU implementation (50x speedup over CPU)
    - Trained on two GPUs for a week



A. Krizhevsky, I. Sutskever, and G. Hinton,
ImageNet Classification with Deep Convolutional Neural Networks, NIPS 2012

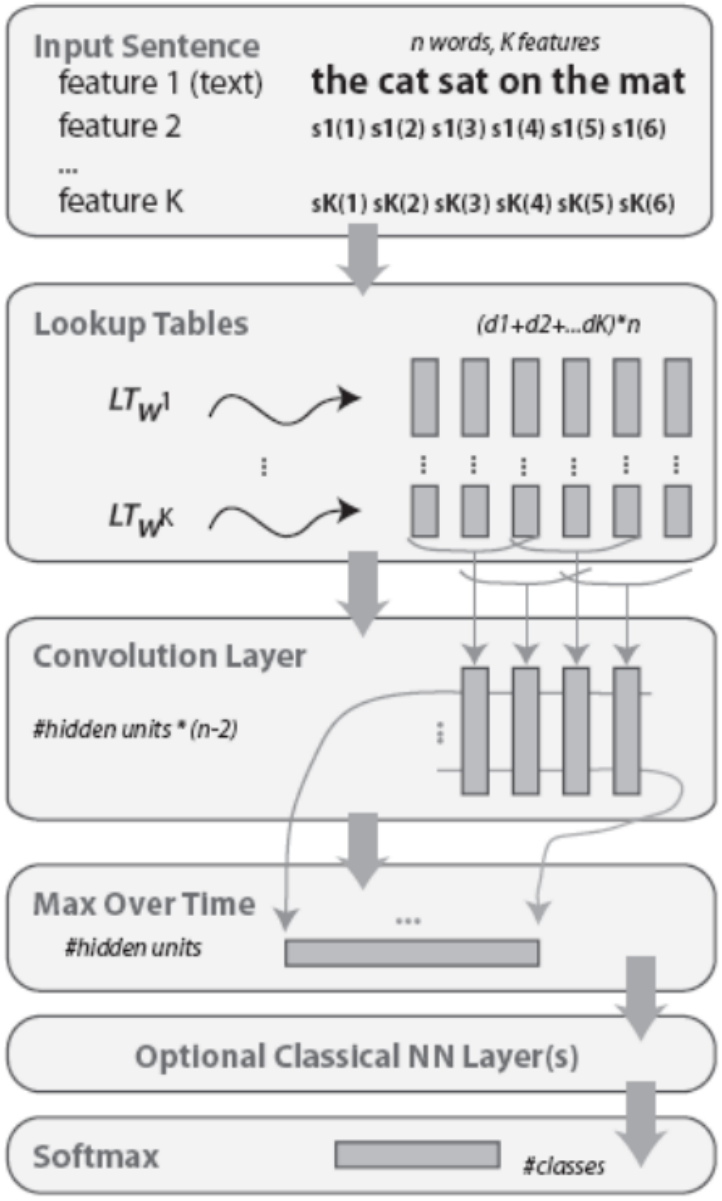
CNN for Time Series

- Convolutional neural networks can also be useful for time-series modeling



- A TDNN could learn that if a certain pattern happened exactly a certain time interval in the past, then sound an alarm
- A CNN TDNN could learn that if a certain pattern occurred *somewhere* in the time window, then sound an alarm (independent on the data outside of the window)!
- Compare: sequential pattern mining / stream mining

General Deep Architecture for NLP



(Collobert and Weston, 2009)

Basic features (e.g., word, capitalization, relative position)

Embedding by lookup table

Convolution (i.e., how each word is relevant to its context?)

Max pooling

Supervised learning

Comments

- Convolutional Deep Neural Networks

Where from here?

- There will never be enough labelled data to learn it all
- The Google cat recognizer sees more cat images than any child and is not as good
- If one assumes that cat features are not encoded genetically, then unsupervised learning. i.e., understanding the world's statistics might do the job! First attempts: RBM, all sorts of Clustering, auto encoders, ...

Tools

- **Torch7** is used at facebook, Deep Mind and several groups at Google (based on LuaJIT which is similar to Python)
- **GP-GPU-CUDA:** Facebook, NYU, and Google/Deep Mind all have custom CUDA back-ends for fast/parallel convolutional network training (CUDA (Compute Unified Device Architecture) is a parallel computing platform and programming model implemented by the graphics processing units (GPUs) that they produce. CUDA gives program developers direct access to the virtual instruction set and memory of the parallel computational elements in CUDA GPUs)
- **Theano:** Python library. Popular in the deep learning community
- **Deeplearning4j** is an open source deep learning library for Java and the Java Virtual Machine
- **Caffe** is a deep learning framework made with expression, speed, and modularity in mind. It is developed by the Berkeley Vision and Learning Center (BVLC) and by community contributors. Yangqing Jia created the project during his PhD at UC Berkeley.

Successes

- Microsoft Speech (Richard Rashid, Chief Research Officer)Chief Research Officer Rick Rashid demonstrates a speech recognition breakthrough via machine translation that converts his spoken English words into computer-generated Chinese language.
- Google: Android Speech Recognition: Maps; Image+ (cats etc.); improve Google translate (ongoing project); Google used the (deep learning) program to update its Google+ photo-search software in May 2013.
- Apple: SIRI (the iPhone's voice-activated digital assistant, Siri, relies on deep learning.)
- Facebook: DeepFace, of the steps detect-align-represent-classify, the representation step is done by a DNNs. Asked whether two unfamiliar photos of faces show the same person, a human being will get it right 97.53 percent of the time. New software developed by researchers at Facebook can score 97.25 percent on the same challenge, regardless of variations in lighting or whether the person in the picture is directly facing the camera.

- ImageNet (Hinton et al.;) classify 1.2 Mio images ImageNet LSVRC contest in 1000 differen classes
- Kaggle: Merck molecular activation (predict useful drug candidates. The task was to trawl through database entries on more than 30,000 small molecules, each of which had thousands of numerical chemical-property descriptors, and to try to predict how each one acted on 15 different target molecules. Dahl and his colleagues won $22,000 with a deep-learning system. “We improved on Merck’s baseline by about 15%,” he says.)