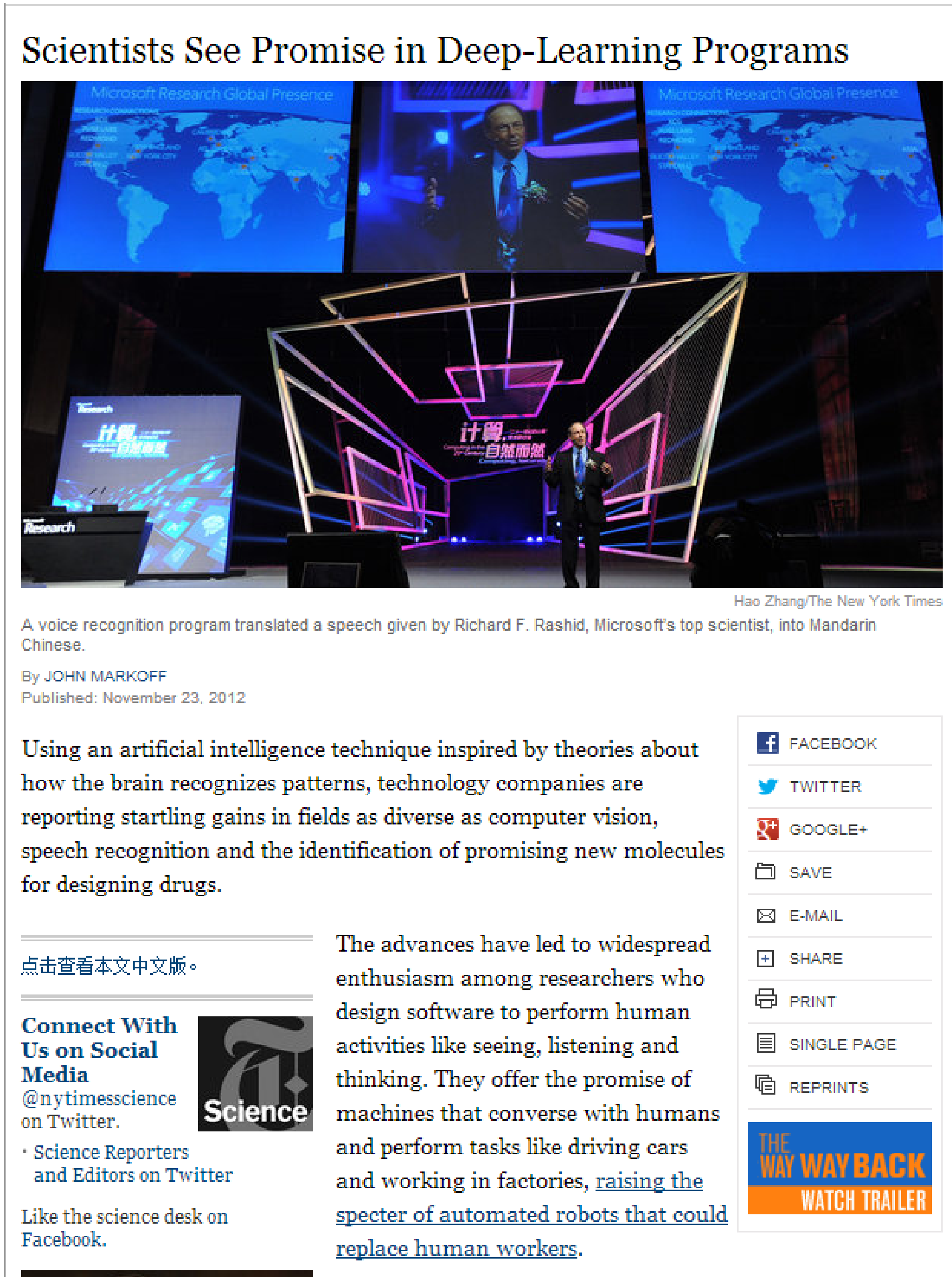
Deep Learning

Volker Tresp
Summer 2014



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。

Connect With Us on Social Media
@nytimesscience on Twitter.

- Science Reporters and Editors on Twitter

Like the science desk on Facebook.

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.

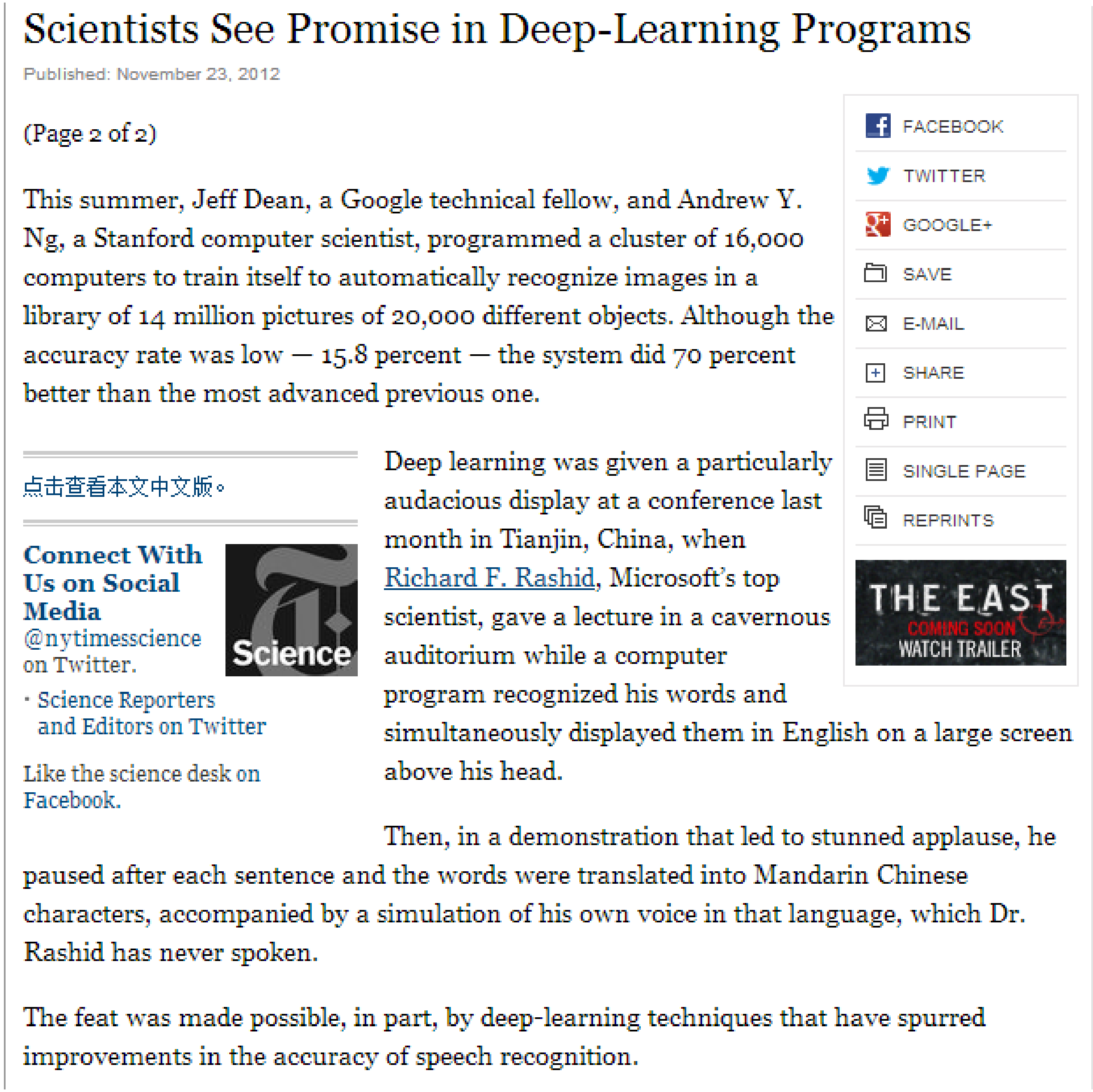
In the 1960s, some computer scientists believed that a workable artificial intelligence system was just 10 years away. In the 1980s, a wave of commercial start-ups collapsed, leading to what some people called the "A.I. winter."

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

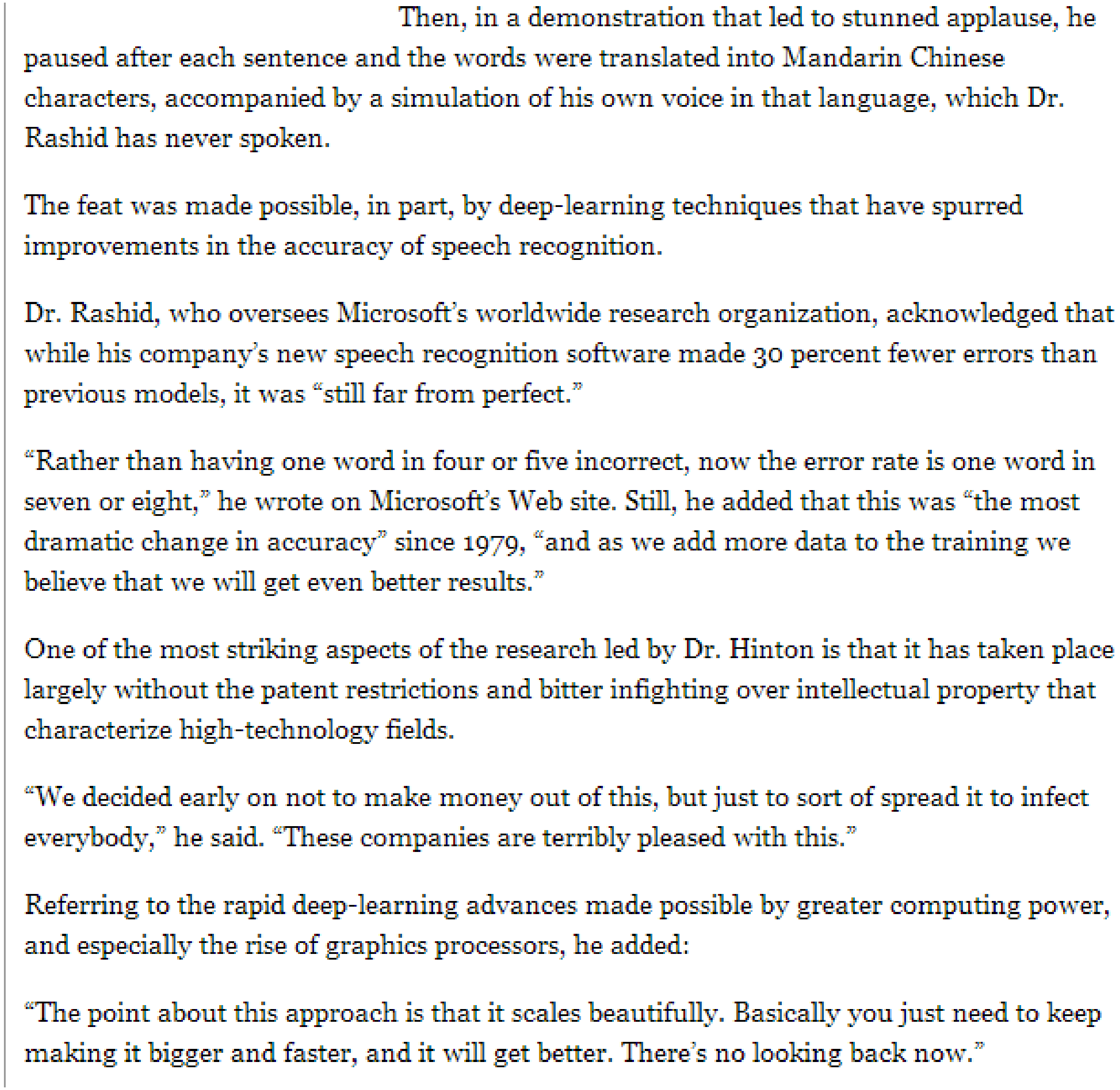
(Page 2 of 2)

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.

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."

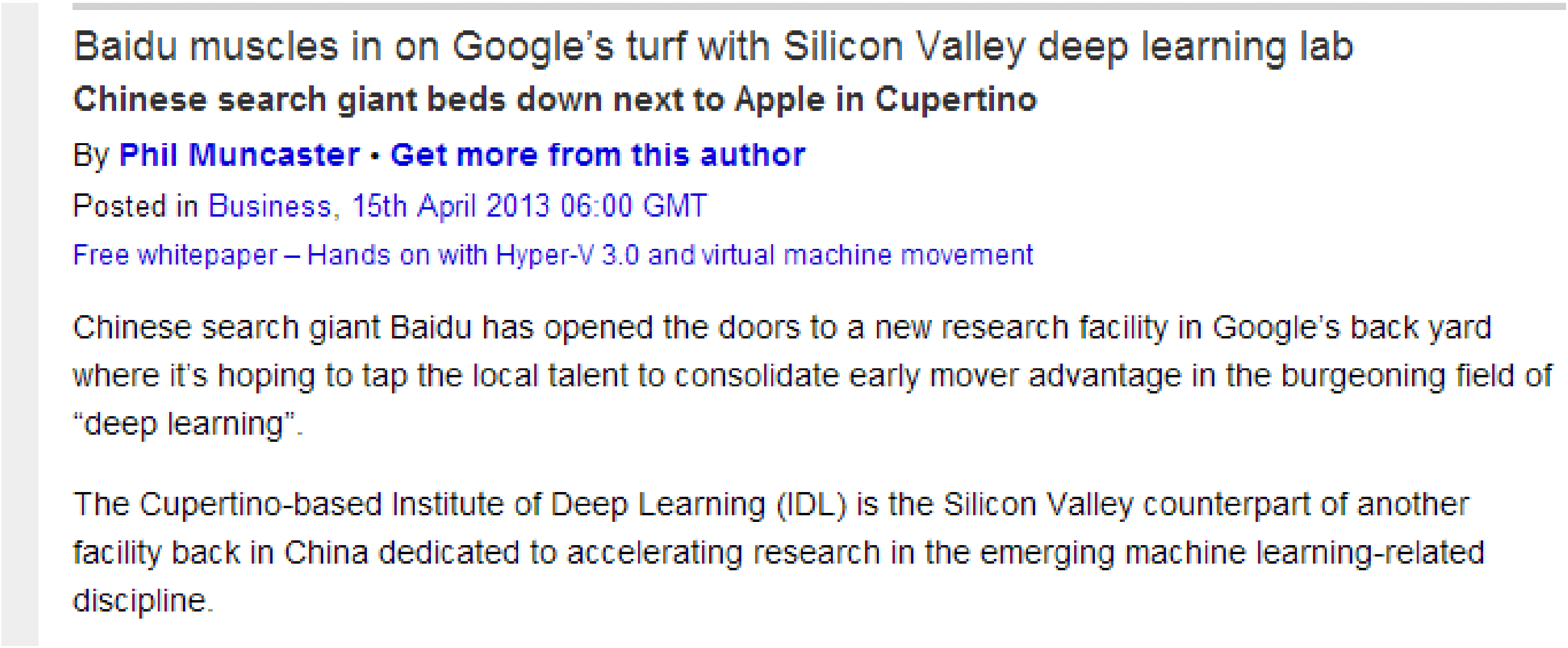
"Rather than having one word in four or five incorrect, now the error rate is one word in seven or eight," he wrote on Microsoft's Web site. Still, he added that this was "the most dramatic change in accuracy" since 1979, "and as we add more data to the training we believe that we will get even better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

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
- They achieved best results on many tasks/datasets
- What are the reasons?

Deep Learning Recipe (Hinton 2013)

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

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

Large Data Sets

- Only now data sets of appropriate size become available
- When decision boundaries are complex, a large data set describes the details
- Details can be captured with a complex (multi-layer) neural networks

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)

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

Weight Regularization

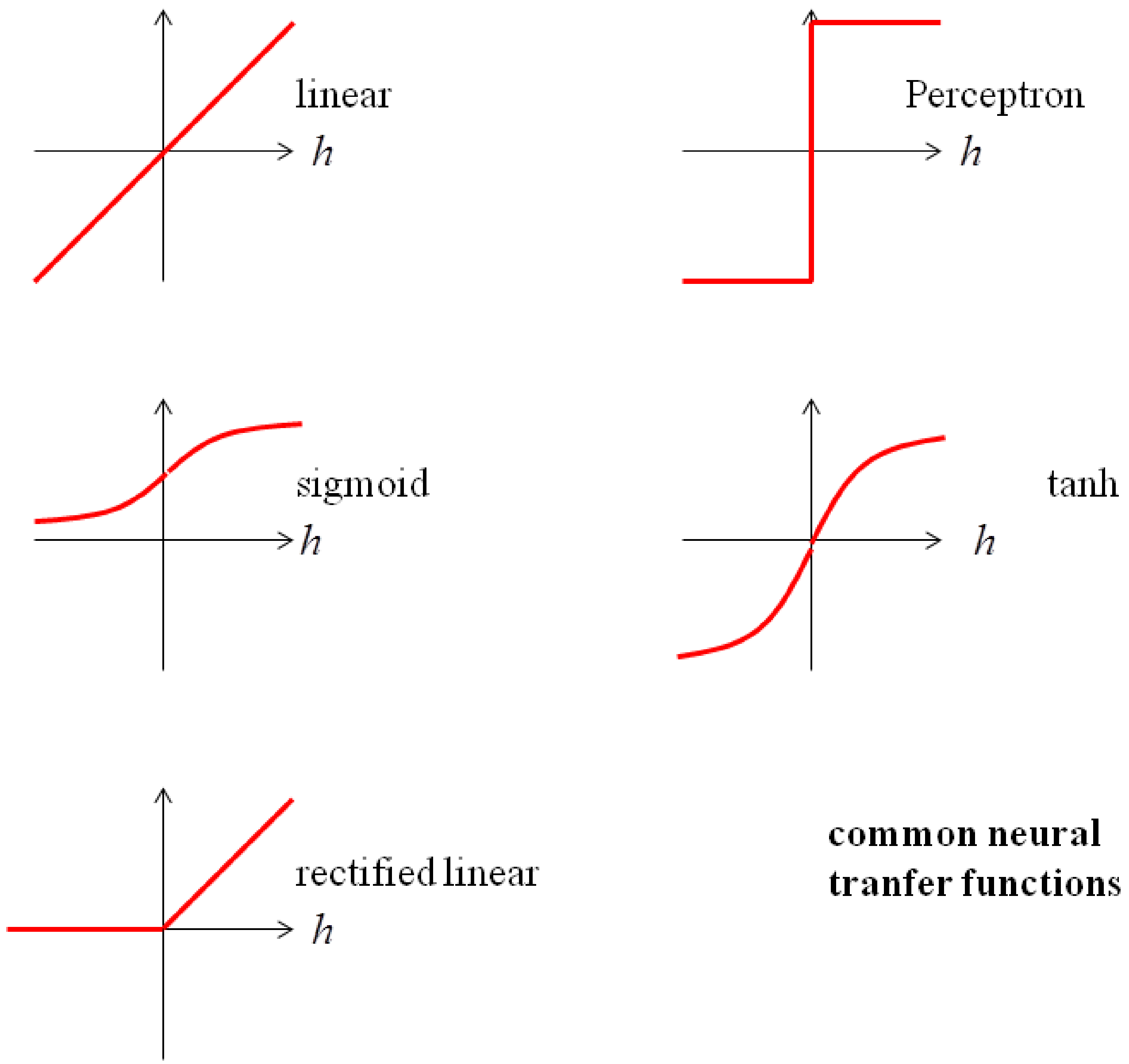
- Weight decay works

- But even better: for each neuron: normalize incoming weight vector to have the same maximum length. Thus if $\|w\| > \alpha$

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

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 a sparse solution



common neural tranfer functions

Initialize Weights with Unsupervised Learning

- Auto-Encoder
- Restricted Boltzmann Machine (RBM)

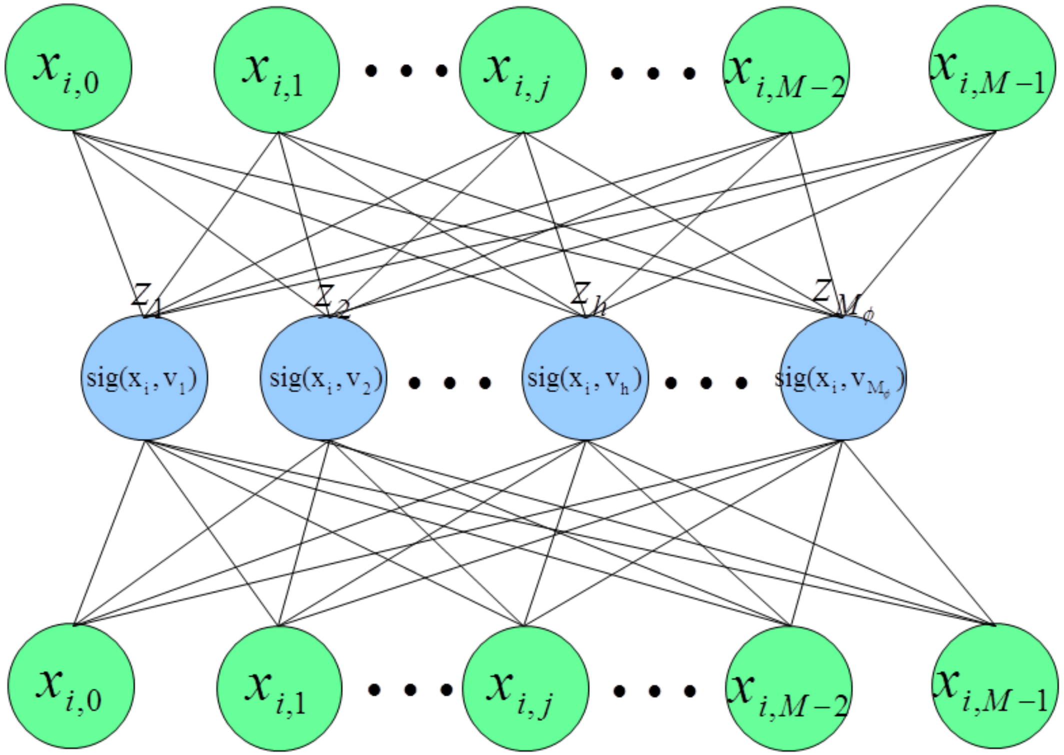
Auto-Encoder for Denoising

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

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

- The constraint is that the number of hidden units is smaller than M, thus a perfect reconstruction becomes impossible
- 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)



Restricted Boltzmann Machine

- 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

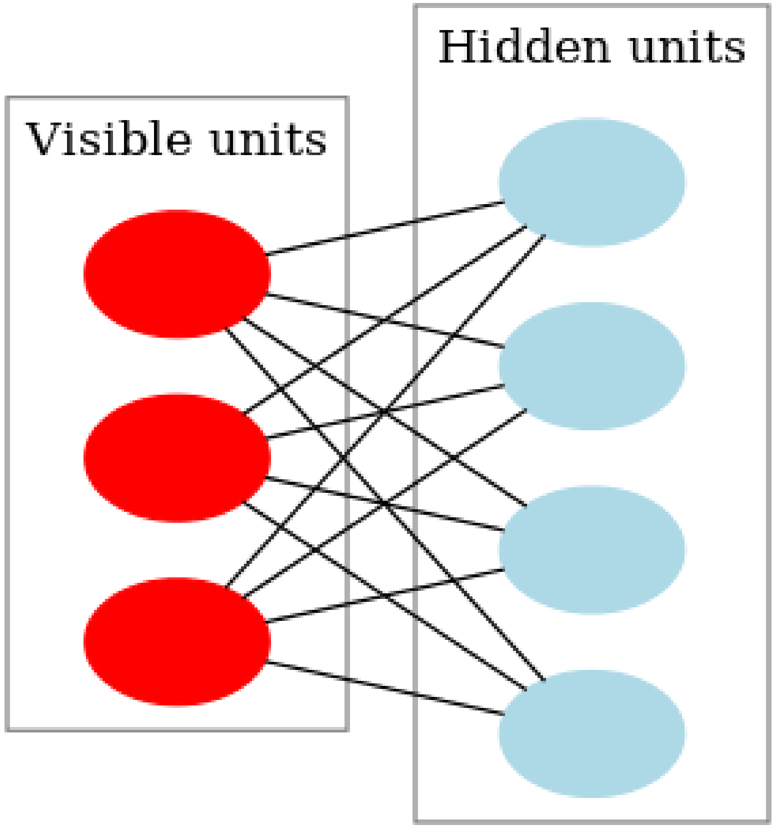
- As their name implies, RBMs are a variant of Boltzmann machines, with the restriction that their neurons must form a bipartite graph: The inputs are connected only to the hidden units, and the hidden units are only connected to the input units (weights are symmetrical: $w_{i,j} = w_{j,i}$)

- 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.

- Can even learn several layers by treating the previous hidden layer as data layer: thus a deep neural network can be initialized (after initialization, backprop is applied)

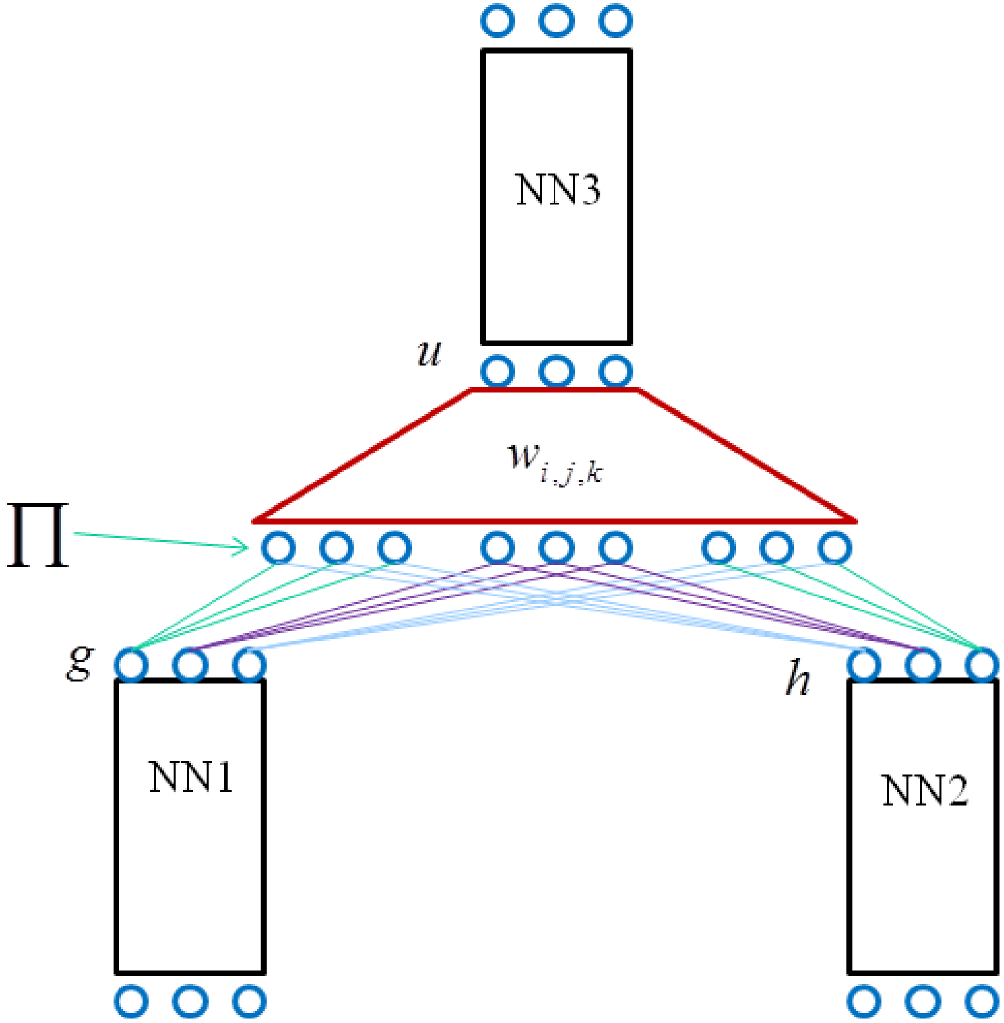
- But: if enough labelled training data is available, RBMs are not necessary, if weights are initialized in the right way

RBM



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
$w_{i,j,k}$
$\prod$
g
h
NN1
NN2

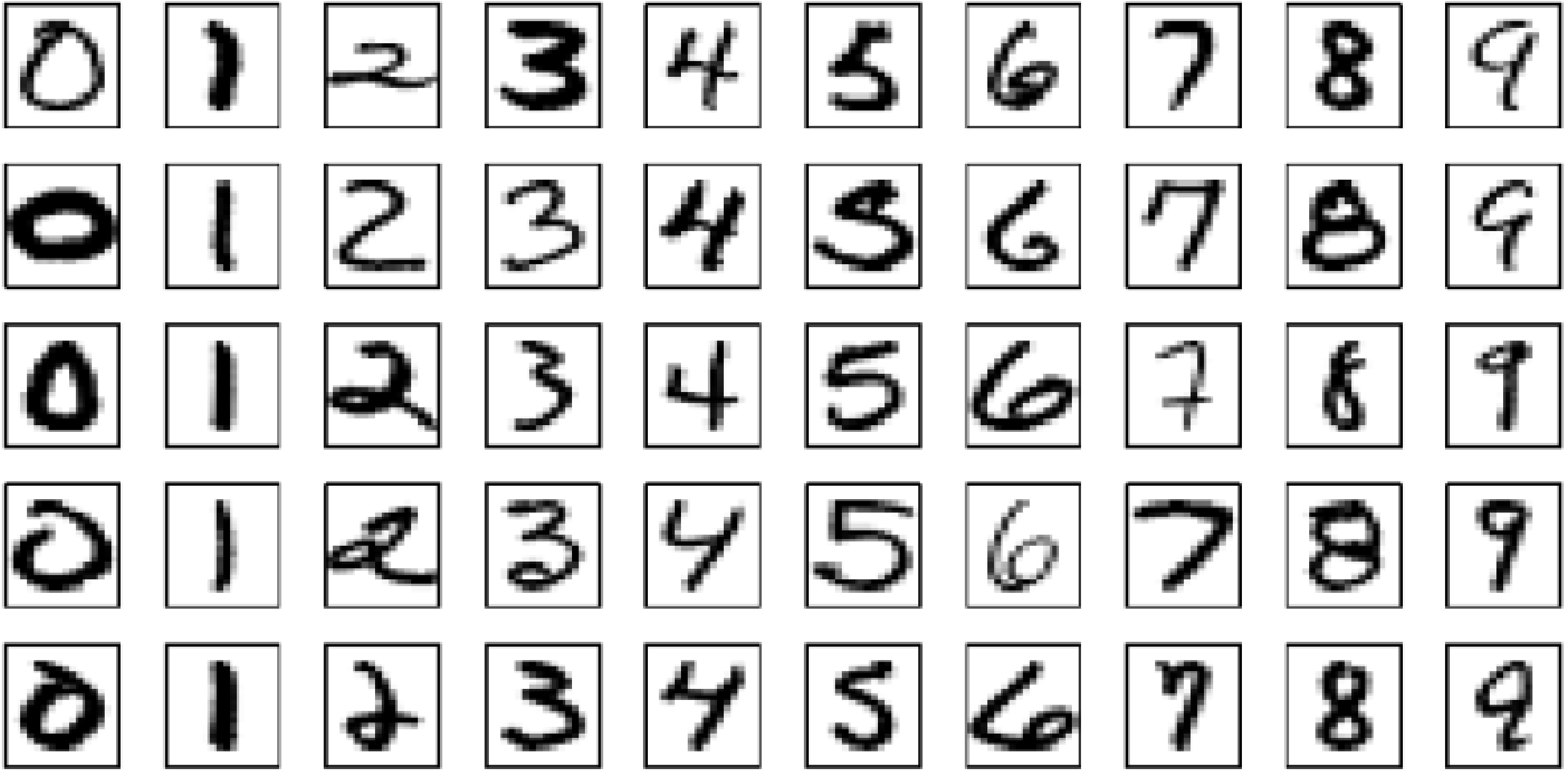
Android Server Architecture for Speech Recognition (2013)

- 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.

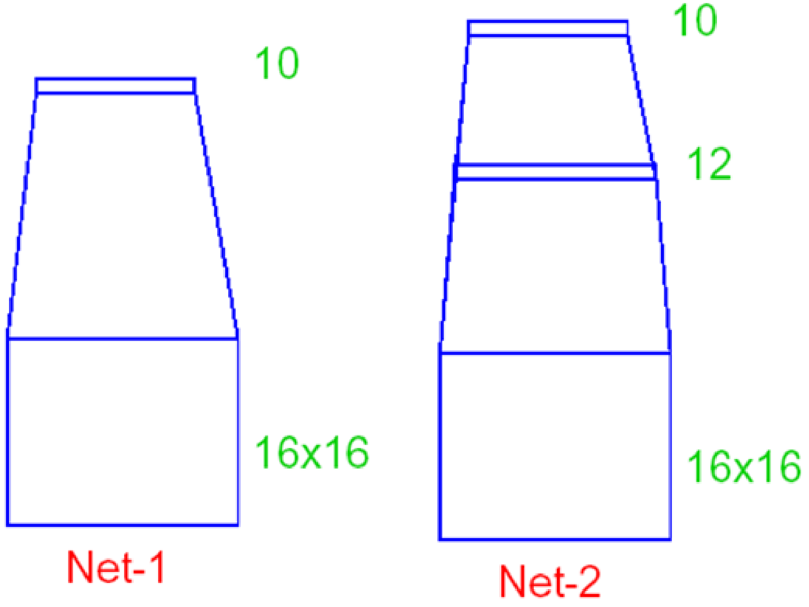
Convolutional Neural Networks (CNNs)

Recognition of Handwritten Digits



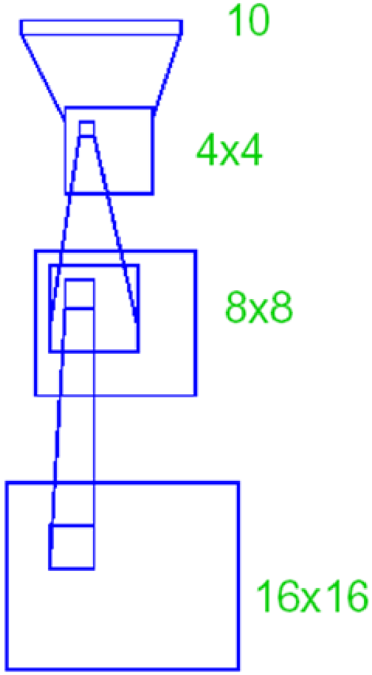
Recognition of Handwritten Digits using Neuronal Networks

- 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

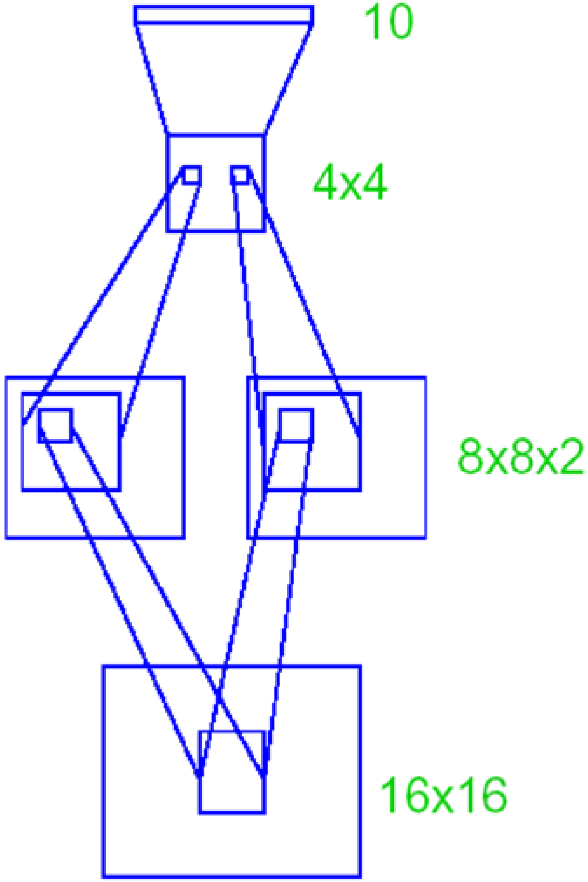
- IN the following variants, the complexity was reduced
- Net-3: Two hidden layers with local connectivity: motivated by the local receptive fields in the brain
  - Each of the 8×8 neurons in the first hiden layer is only connected to 3×3 input neurons from a receptive field
  - In the second hidden 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

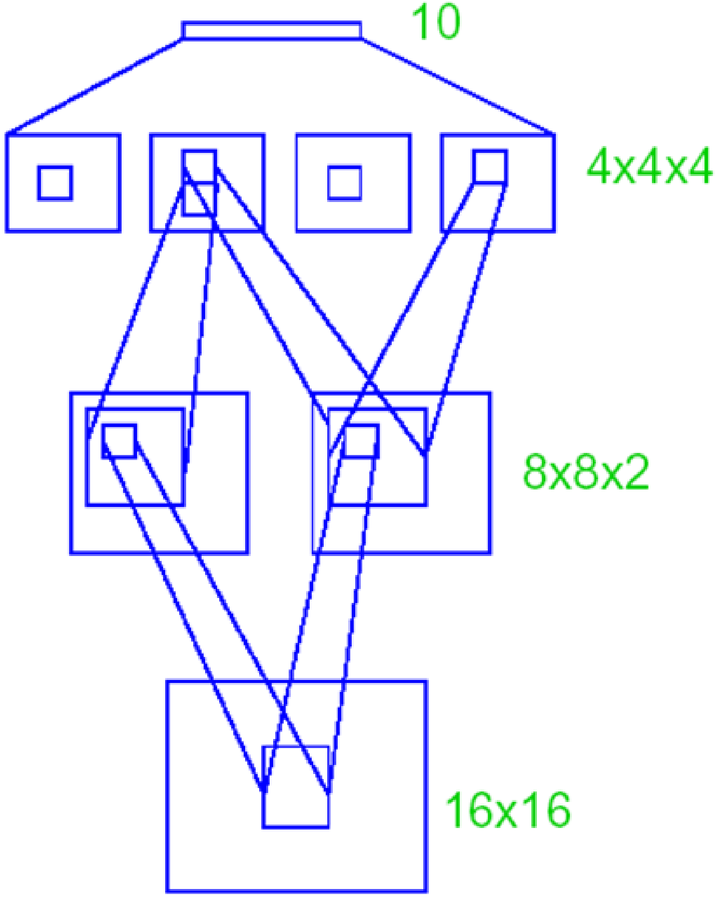
Neuronal Networks with Weight-Sharing (Net-4)

- 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)

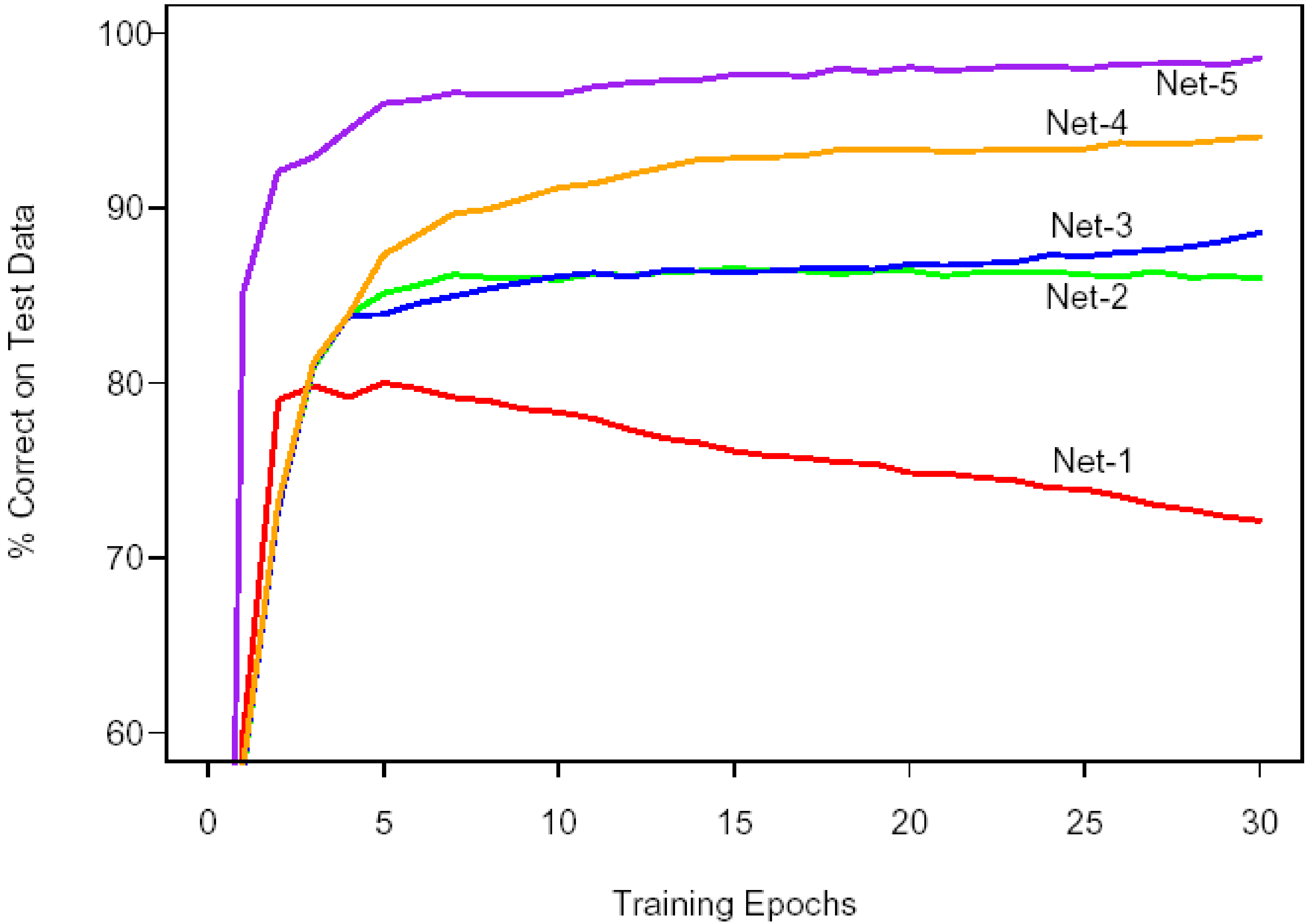
- Net-5: Two hidden layers with local connectivity and two layers of *weight-sharing*



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


% Correct on Test Data
100
90
80
70
60
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)

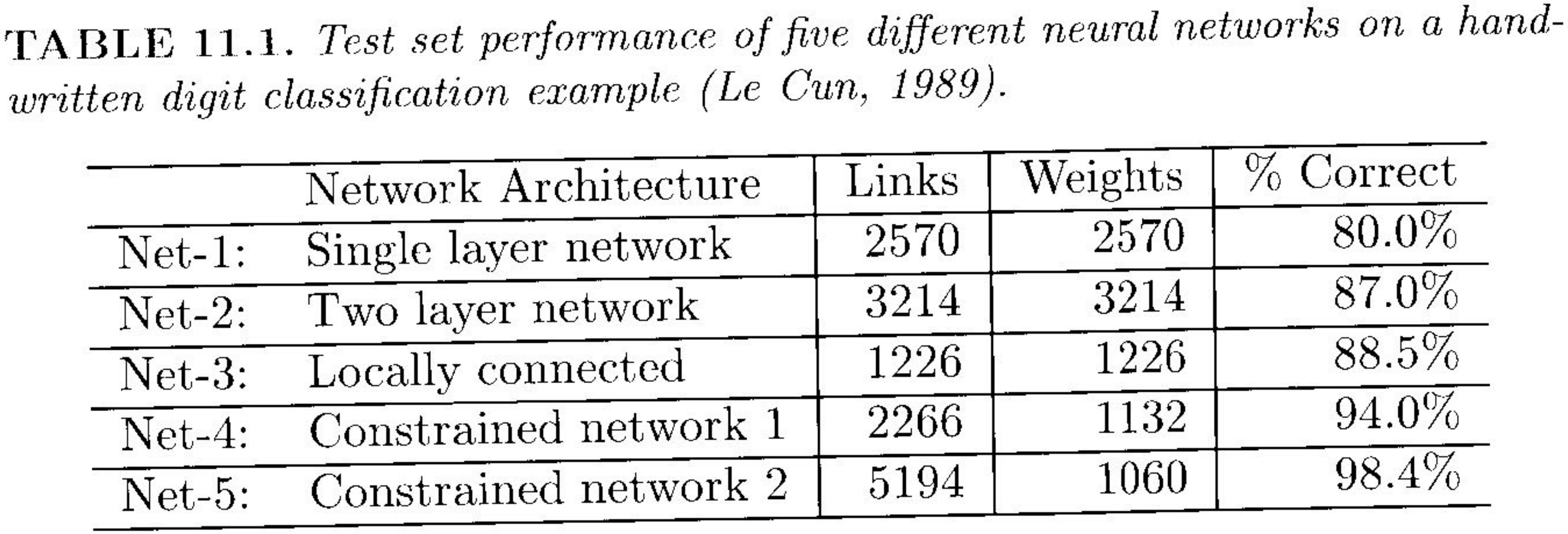


TABLE 11.1. *Test set performance of five different neural networks on a handwritten digit classification example (Le Cun, 1989).*

	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%

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.

Where from here?

- There will never be enough labelled data to learn it all
- The Google cat recognizer sees more cat images as any child and is not as good
- If one assumes that can 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, ...