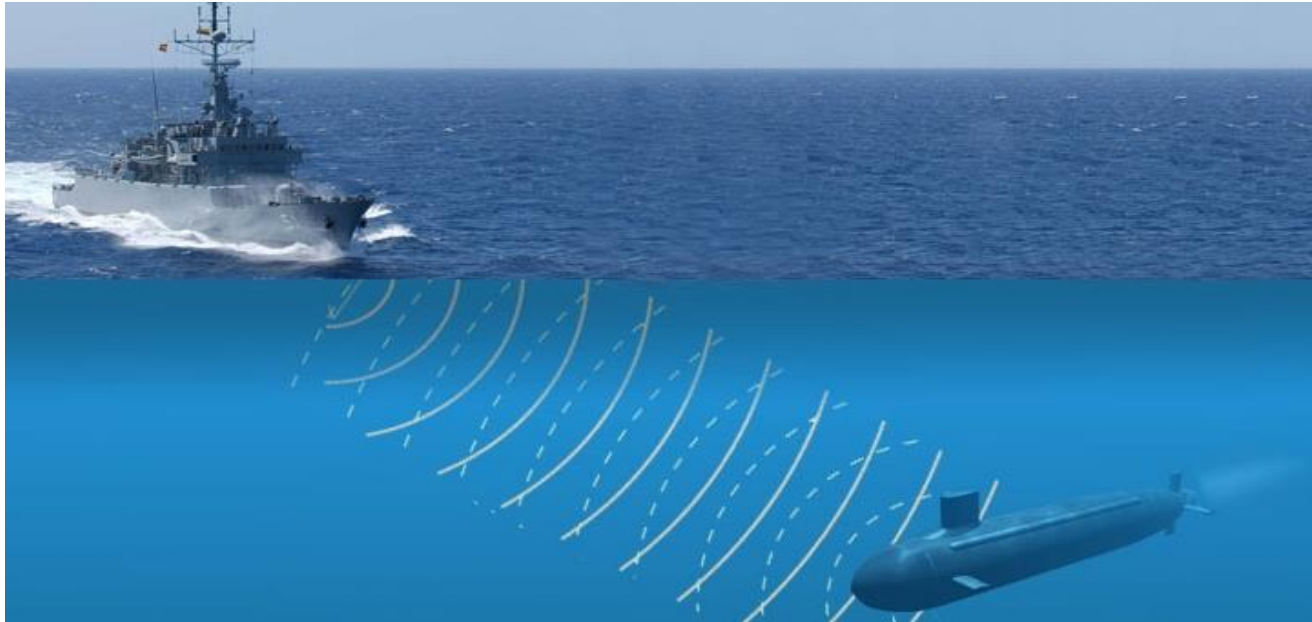


Applications of a feedforward MLP:
Sonar classification, NetTalk,
ALVINN and recognition of handwritten characters

Lubica Benuskova

Beňušková Ľ. Umelé neurónové siete. In: Návrát P. a kol.,
Umelá inteligencia (3. vydanie), Vydavateľstvo STU, 2015

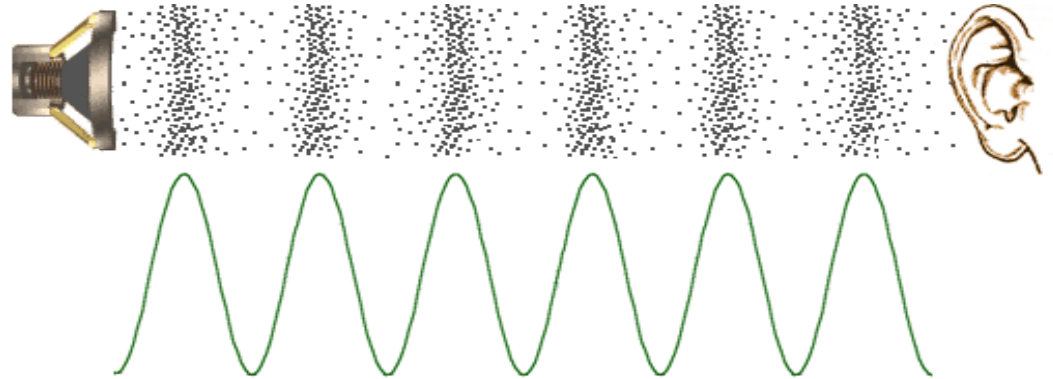
Classification of sonar signals



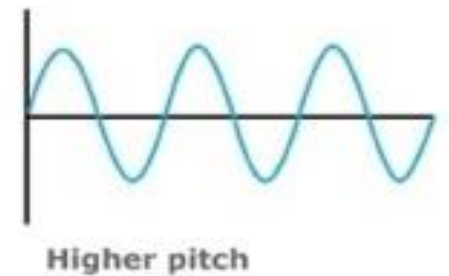
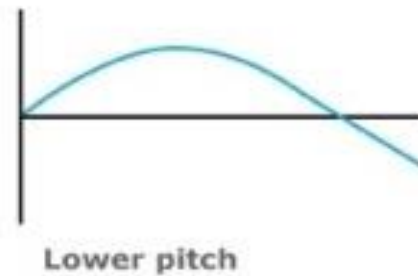
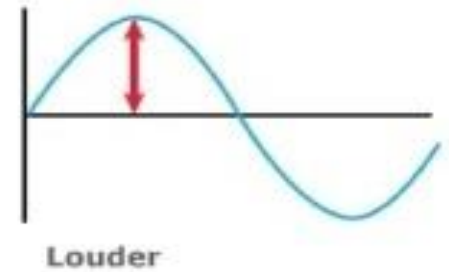
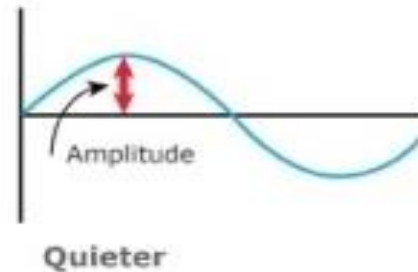
- SONAR (SOund Navigation And Ranging): the sonar emits a sound and then senses the return sound bounced back from an object.
- Task: to train a MLP to discriminate between sonar signals bounced off a metal cylinder (a mine, submarine) and those bounced off a roughly cylindrical rock (**Authors: Gorman and Sejnowski 1988**)

Sound waves

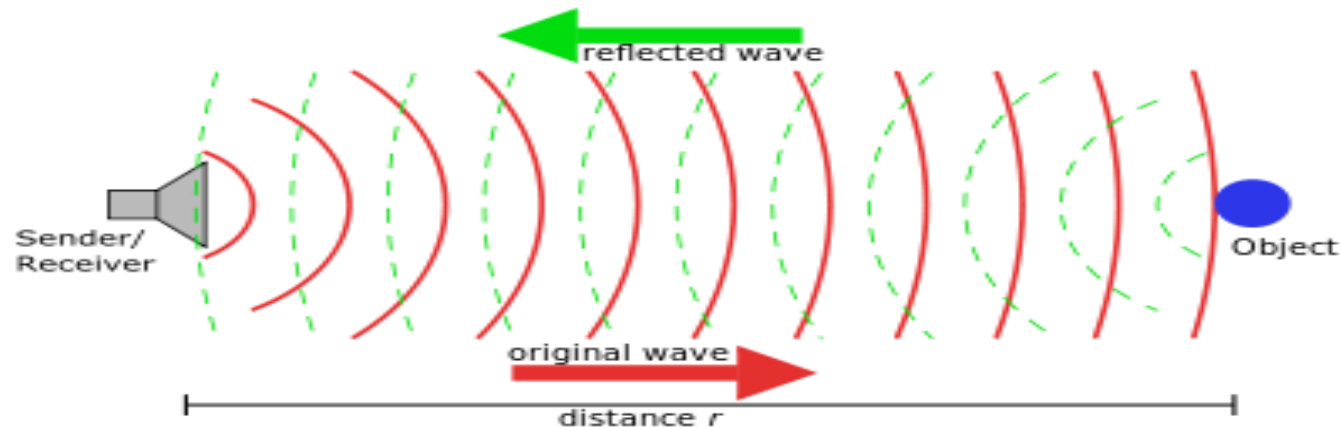
- Sound is a wave:



- The amplitude and frequency (number of cycles per second) determine the loudness and pitch, respectively.



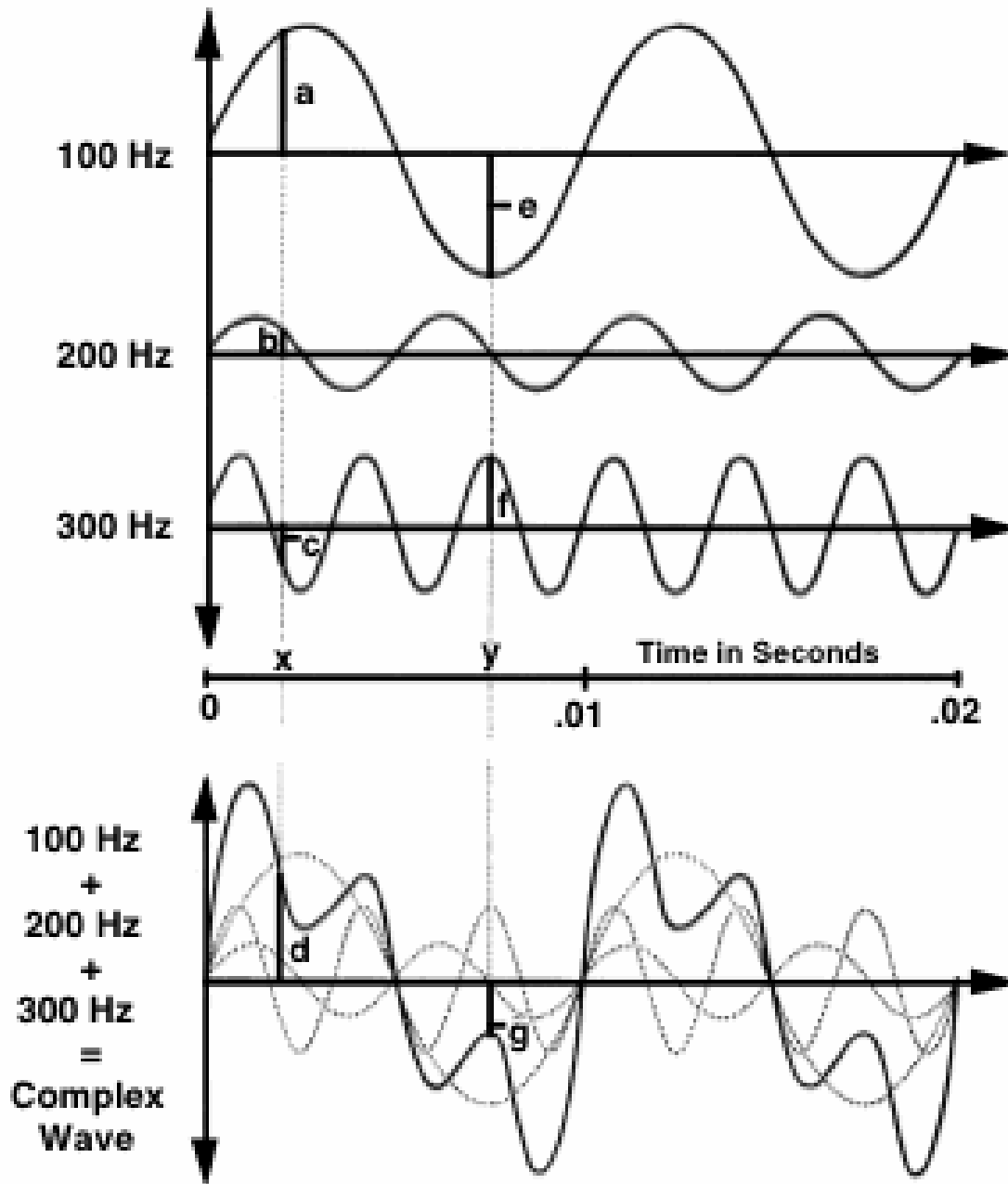
Set up of sonar experiment



- In the experiment with MLP, sonar signals were measured from the same distance but different angles ($0 - 90$ or $0 - 180$) on real objects (metal cylinders and rocks) that were all about the same size.
- The bounced back sound wave was a complex sound wave:



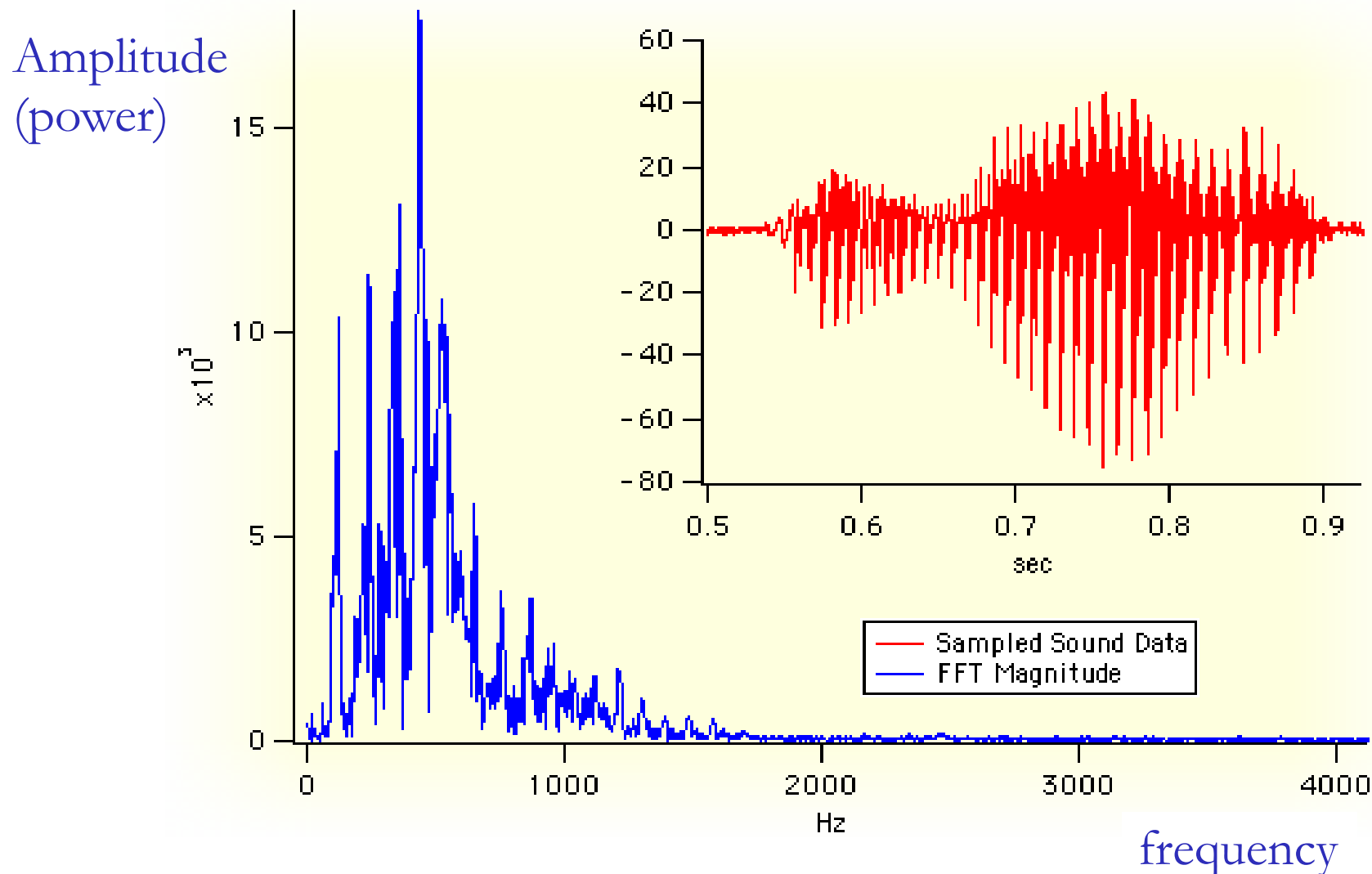
Fourier analysis – principle



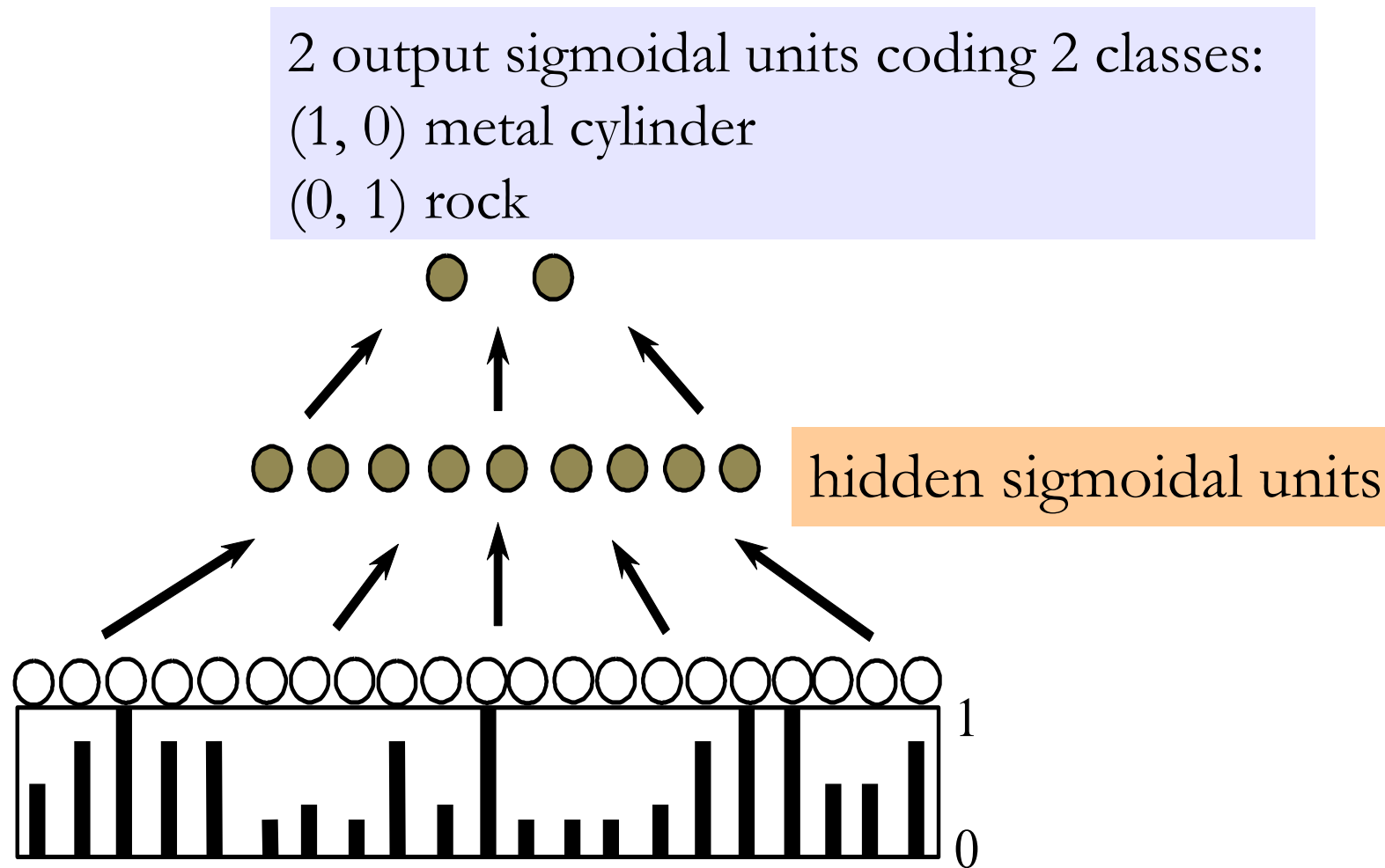
- Fourier (read Furrye) analysis is a decomposition of an original composite wave into elementary sine waves, each with a different amplitude (power) and frequency.

Fourier analysis – example

- Thus a results of Fourier analysis is a graph of amplitudes as a function of elementary frequency in the sound wave.



MLP architecture for sonar classification



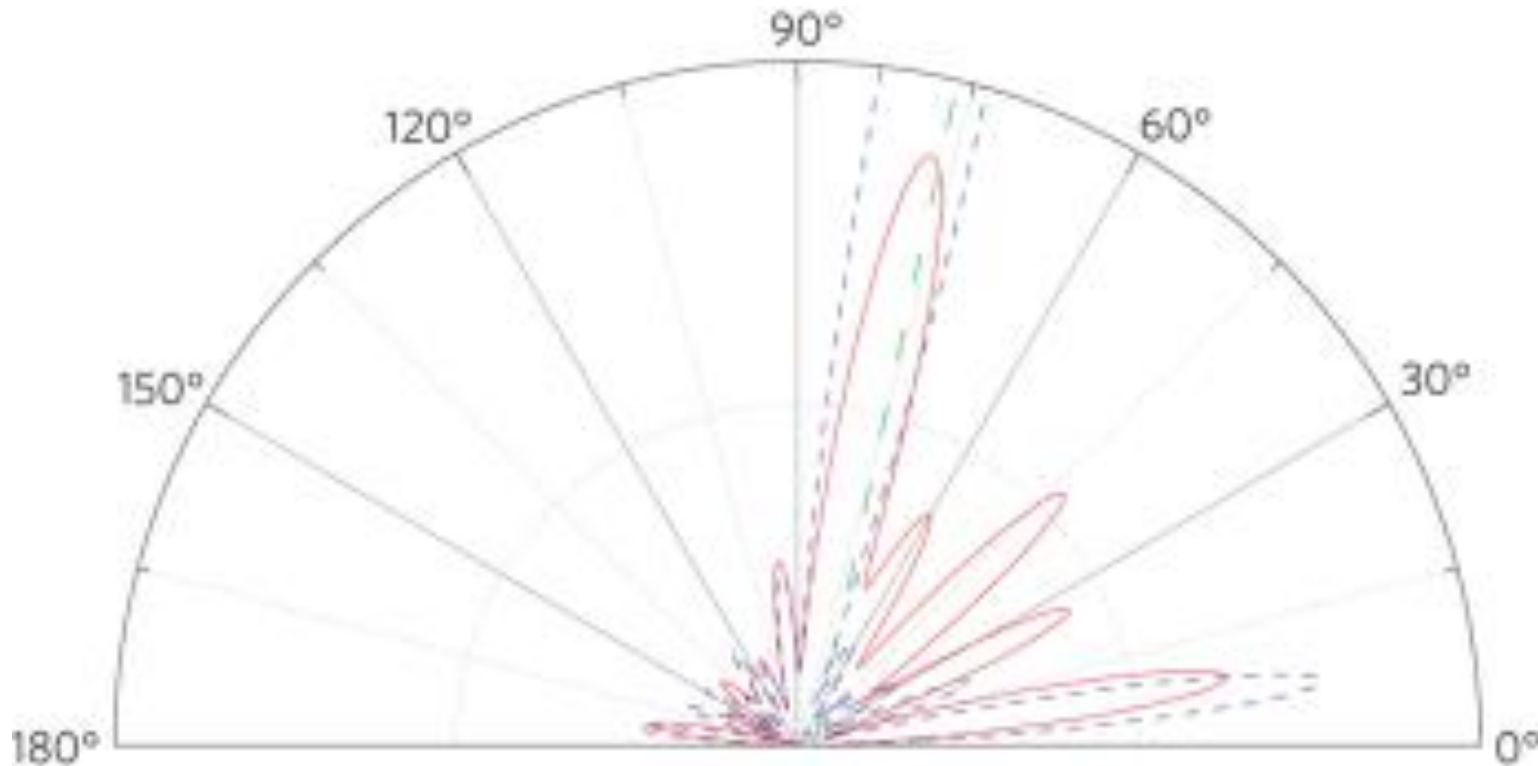
- Input: power spectrum of normalized intensities for 60 frequency bands
- Each input pattern is a vector of 60 numbers in the range 0.0 to 1.0.
- Each number represents the energy within a particular frequency band, integrated over a certain period of time (e.g. 1 second)

Sonar net – training parameters

- The goal of training: to train the MLP to discriminate between sonar signals bounced off a metal cylinder (representing mine, or submarine) and those bounced off a roughly cylindrical rock (Gorman and Sejnowski 1988)
- Parameters of error backpropagation rule
 - Learning speed $\alpha = 2$, initial weights $[-0.3, +0.3]$.
- Training and testing sets consisted of these underwater data:
 - 111 patterns were obtained by bouncing sonar signals off a metal cylinder at various angles (spanning 90°).
 - 97 patterns from rocks under similar conditions (span 180°).

Testing

- The data were split into the training set and the test set according to random rule or "aspect-angle dependent" rule.
- In the random rule, the samples for testing were chosen at random.
- In the "aspect-angle dependent" rule, the samples for testing were selected so that the distribution of angles in the train and test set was the same.



Sonar net – each number is average from 13 runs

Random

#Hidden Units	% Right on Training set	Std. Dev.	% Right on Test Set	Std. Dev.
0	89.4	2.1	77.1	8.3
2	96.5	0.7	81.9	6.2
3	98.8	0.4	82.0	7.3
6	99.7	0.2	83.5	5.6
12	99.8	0.1	84.7	5.7
24	99.8	0.1	84.5	5.7

Angle dependent

Hidden Units	% Right on Training set	Std. Dev.	% Right on Test Set	Std. Dev.
0	79.3	3.4	73.1	4.8
2	96.2	2.2	85.7	6.3
3	98.1	1.5	87.6	3.0
6	99.4	0.9	89.3	2.4
12	99.8	0.6	90.4	1.8
24	100.0	0.0	89.2	1.4

Sonar net – comparison

- The nearest neighbour classifier on the same data gave 82.7% probability of correct classification (they evaluated Euclidean distance from the test signal to $K=2$ closest training neighbours and the closest one determined the class).
- Three trained human subjects were each tested on 100 signals, chosen at random from the set of all signals. Their responses ranged between 88% and 97% correct.
- Conclusion: simple net performed equally well as the worst human.

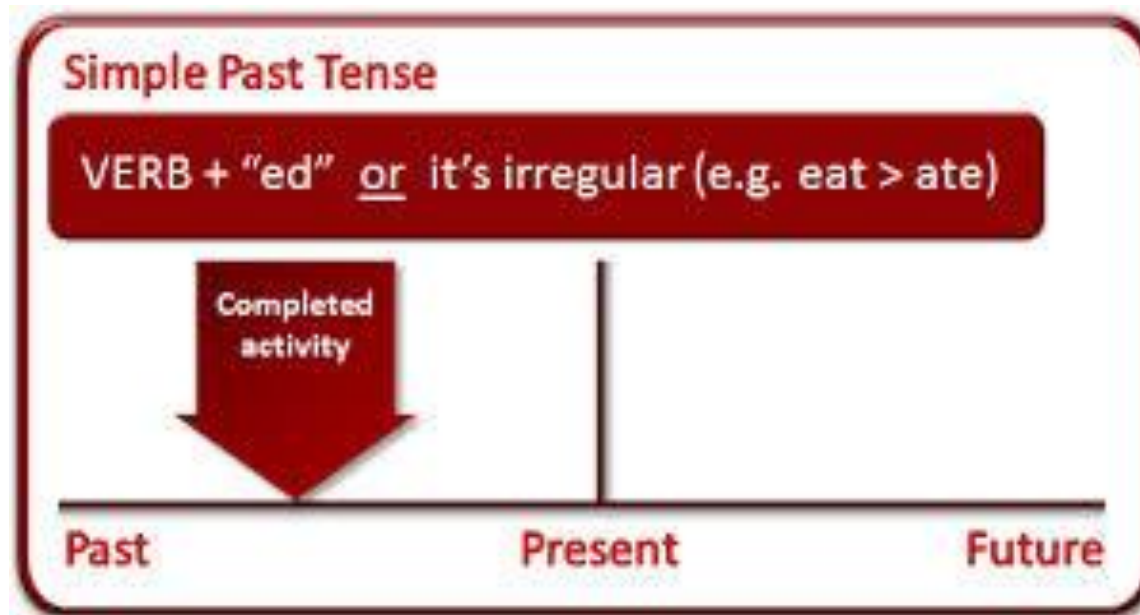
NETtalk (Sejnowski and Rosenberg, 1987)

- NETtalk – converts written English text into speech.
- Pronunciation: correspondence between letters and phonemes (elementary speech sounds):

Phoneme (sound)	Examples	Graphemes (written patterns)		Phoneme (sound)	Examples	Graphemes (written patterns)	
/b/	<u>b</u> anana, bu <u>bb</u> les	b	bb	/s/	<u>s</u> un, mou <u>se</u>	s	ss, ce, se, c, sc
/c/	<u>c</u> ar, du <u>ck</u>	c	k, ck, q, ch	/t/	<u>t</u> urtle, li <u>tt</u> le	t	tt
/d/	<u>d</u> inosaur, pu <u>dd</u> le	d	dd,	/v/	<u>v</u> olcano, ha <u>lv</u> e	v	ve
/f/	<u>f</u> ish, gi <u>raff</u> e	f	ff, ph, gh	/w/	<u>w</u> atch, <u>que</u> en	w	wh, u
/g/	<u>g</u> uitar, go <u>gg</u> les	g	gg,	/x/	fo <u>x</u>	x	
/h/	<u>h</u> elicopter	h		/y/	yo-yo	y	
/j/	<u>j</u> ellyfish, fri <u>dg</u> e	j	g, dge, ge	/z/	<u>z</u> ip, plea <u>se</u>	z	zz, ze, s, se
/l/	<u>l</u> eaf, be <u>ll</u>	l	ll, le	/sh/	<u>sh</u> oes, televi <u>si</u> on	sh, ch, si, ti	
/m/	<u>m</u> onkey, ha <u>mm</u> er	m	mm, mb	/ch/	<u>ch</u> ildren, sti <u>ch</u>	ch, tch	
/n/	<u>n</u> ail, <u>kn</u> ot	n	nn, kn	/th/	mo <u>th</u> er	th	
/p/	<u>p</u> umpkin, pu <u>pp</u> ets	p	pp	/th/	<u>th</u> ong	th	
/r/	<u>r</u> ain, <u>wr</u> ite	r	rr, wr	/ng/	<u>ng</u> , a <u>nk</u> le	ng, n	

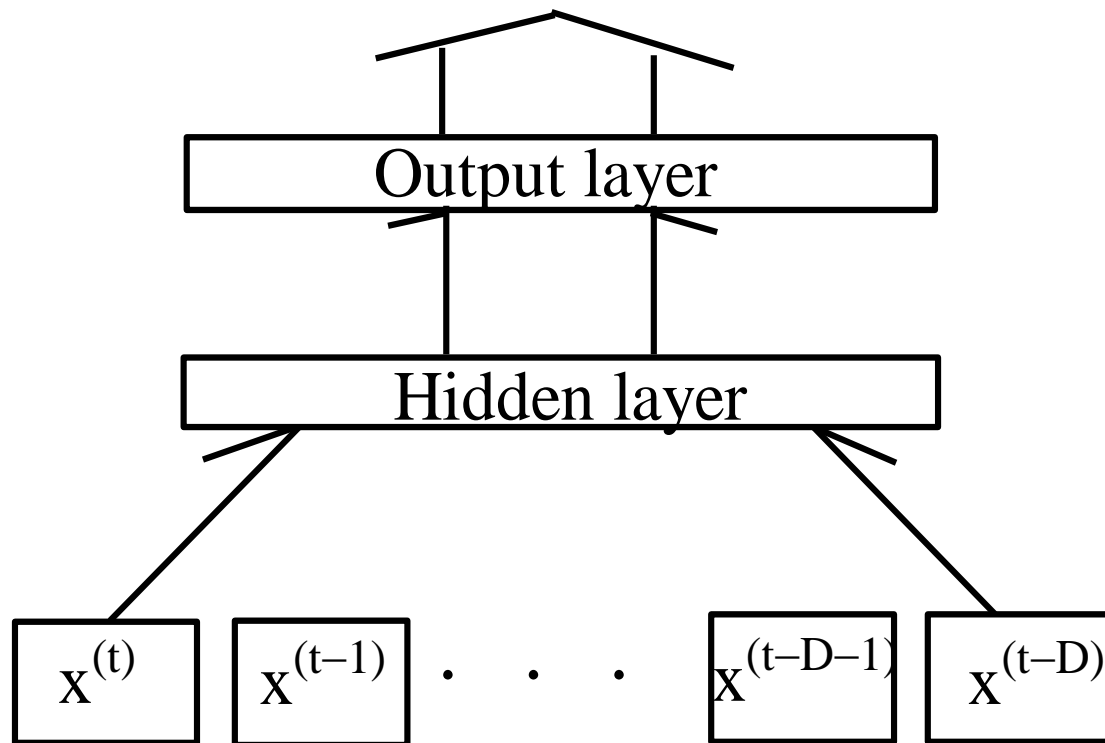
NETtalk (Sejnowski and Rosenberg, 1987)

- NETtalk – converts English text into speech
- English is difficult, irregular spelling (ceiling vs. sealing, etc.)
- Problem of reconciling rules and exceptions is a difficult one
 - For instance forming the past tense of verbs, etc.



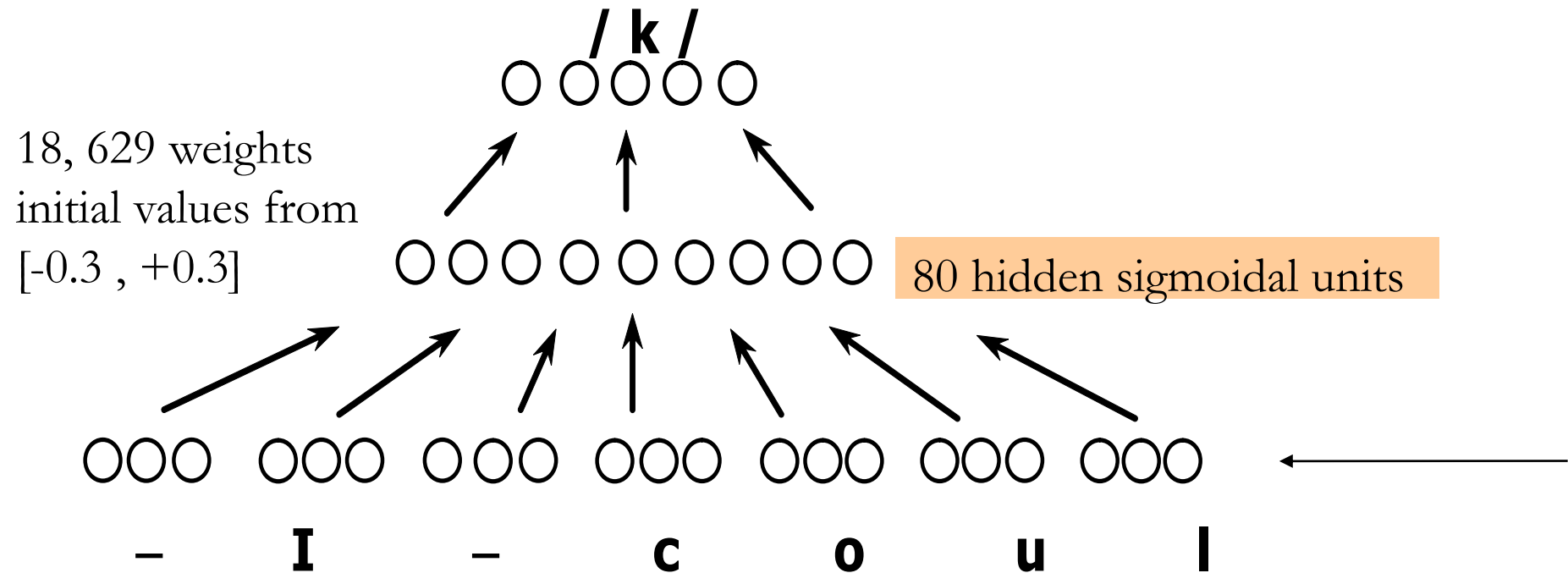
NETtalk architecture

- NETtalk – converts English text into speech.
- Moving window architecture, the so-called **TDNN** – temporal delay neural network, where $x(t)$ is the symbol (letter) at time t .



NETtalk - architecture

- 26 output sigmoidal units coding 26 phonemes, connected to a digital speech synthesiser
- each unit coded 21 articulatory features, 5 additional units encoded stress & boundaries
- desired output is the phoneme for the letter in the middle input group



- A moving window of letters in the text fed through 203 inputs
- 7 groups of units
- Each group contained 29 elements coding 26 letters by 1-hot code plus 3 additional units that coded punctuation, continuation and word boundaries

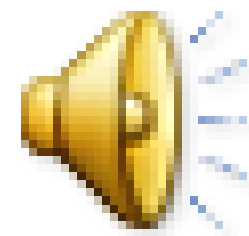
NETtalk – training method

- Text stepped through the window letter by letter (text was a record of a speech of a child in the first grade);
- At each step MLP computed a phoneme code for the letter in the centre in the context of 3 letters before and after;
- After each word the weights were adjusted by error-backpropagation;
- After each input, the inner/dot product of the output \mathbf{o} and desired output \mathbf{y} was computed: the phoneme with the largest value won:

$$\mathbf{y} \cdot \mathbf{o} = \sum_{i=1}^n y_i o_i$$

NETtalk – learning stages on child's set

- 1st stage: Distinctions between vowels (a, e, ...) and consonants (b, k, ...) – However, the net predicted the same vowel for all vowels and the same consonant for all consonants – “babbling” stage
- 2nd stage: recognition of word boundaries, the output resembles pseudo-words
 - After 10 epochs the text was understandable
- When the net made an error, it substituted phonemes that sounded similar (even corrected errors in the train set)
 - <http://cnl.salk.edu/Media/nettalk.mp3>



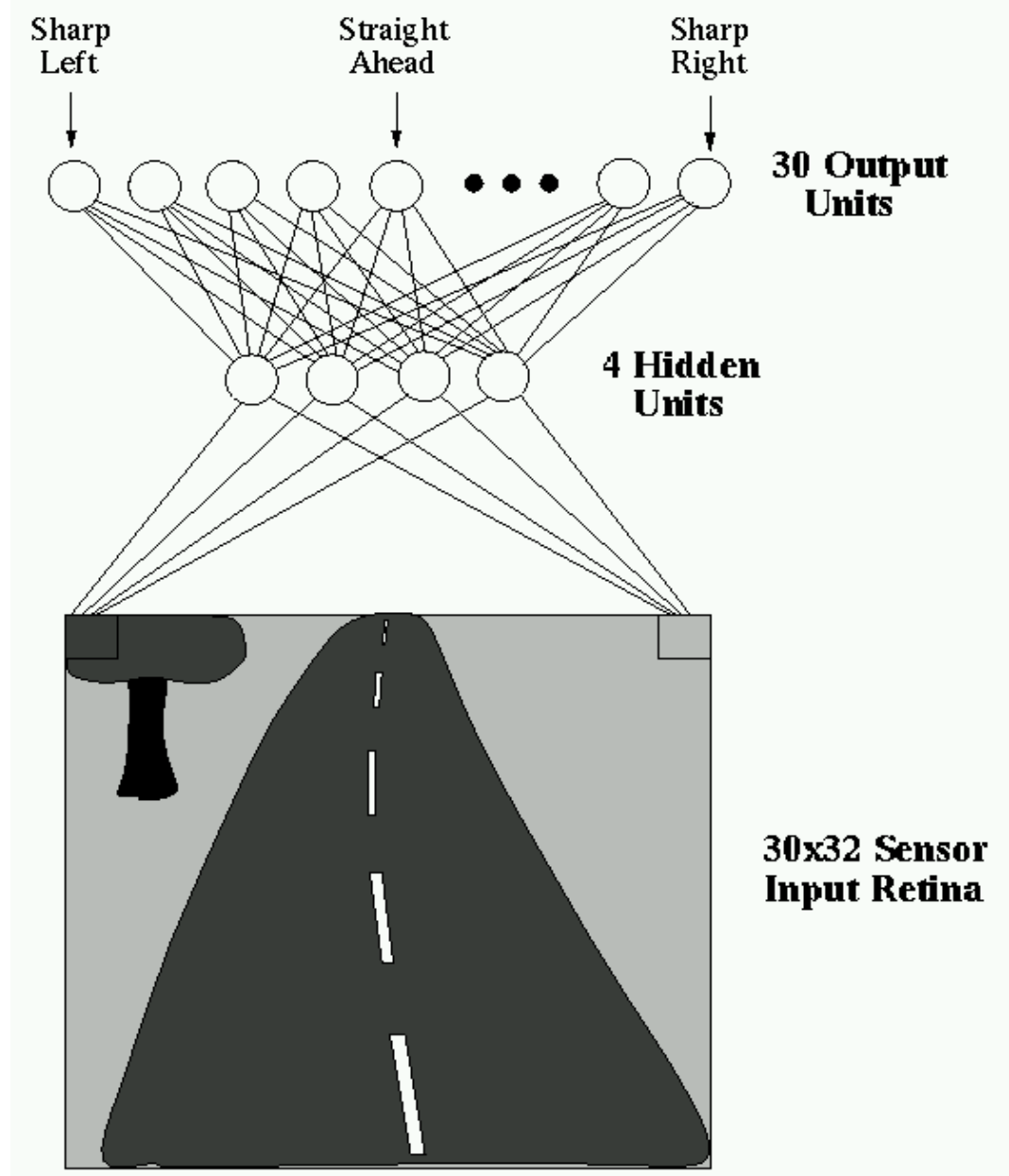
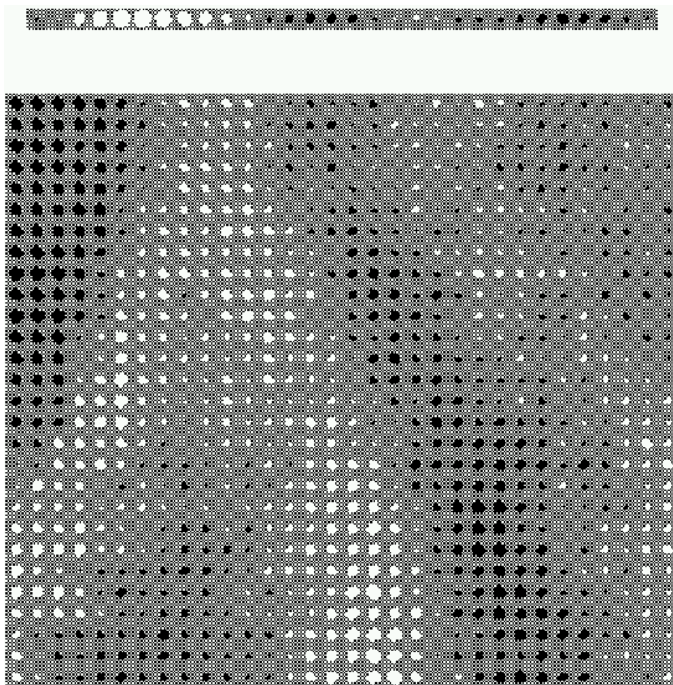
NETtalk – conclusions

- Illustration of many aspects of learning language
 - It starts with no apriori knowledge about pronunciation
 - It could have been trained on any language
 - It went through distinct stages during learning
- The information was coded in a distributed manner over neurons and weights, i.e. no single unit or synapse was essential
 - As a consequence, degraded gracefully with increasing damage
- It accomplishes in one stage what occurs in two stages: children learn to talk first, then they learn to read.

ALVINN

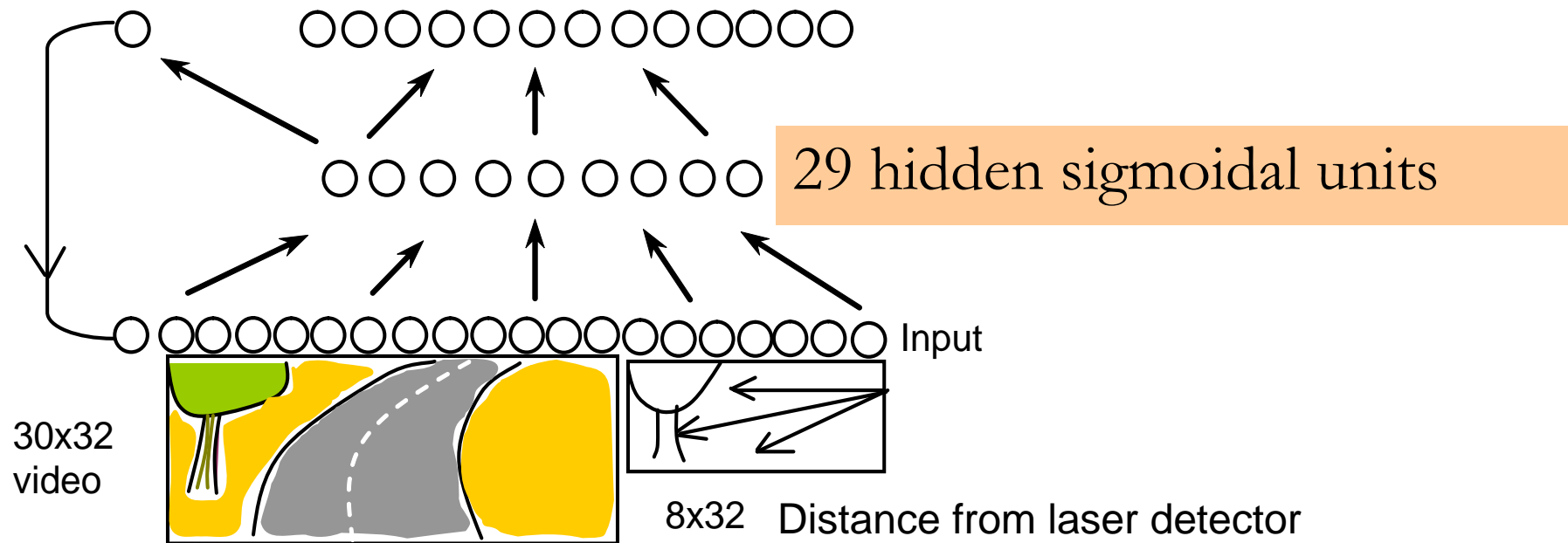
- **Autonomous Land Vehicle In a Neural Net** learns to control the vehicle by “watching” a driver (Pomerleau 1989).
- ALVINN's architecture consists of a single hidden layer back-propagation network.
 - The input layer of the network is a 30 times 32 unit "retina", which receives input from the car's video camera.
 - Each input unit is fully connected to five hidden units, which are in turn fully connected to a layer of 30 output units.
- The output layer is a linear representation of the direction the vehicle should travel in order to keep the car on the road.

ALVINN



Improved architecture of ALVINN

45 directional linear neurons that code an angle of a steering wheel + one special neuron indicates the light contrast between the road and the non-road



- Input: from the blue part of light spectrum (better contrast)
- Input value = intensity of a blue light in the range 0.0 to 1.0 in each pixel.
- 1217 input values in total.

ALVINN – training

- The training set consisted of 1200 snapshots of the road in the park and the corresponding angle of a steering wheel.
- After $\frac{1}{2}$ hour, vehicle was tested in the park.
- Training was repeated several times. Every time errors were included in the new training set with the correct answer (i.e. angle of steering).
- Result: moving at a speed 5 km/h, the vehicle could master also unknown roads in the park.
- This was really a first simple attempt. No traffic, low speed, etc.

Autonomous cars today

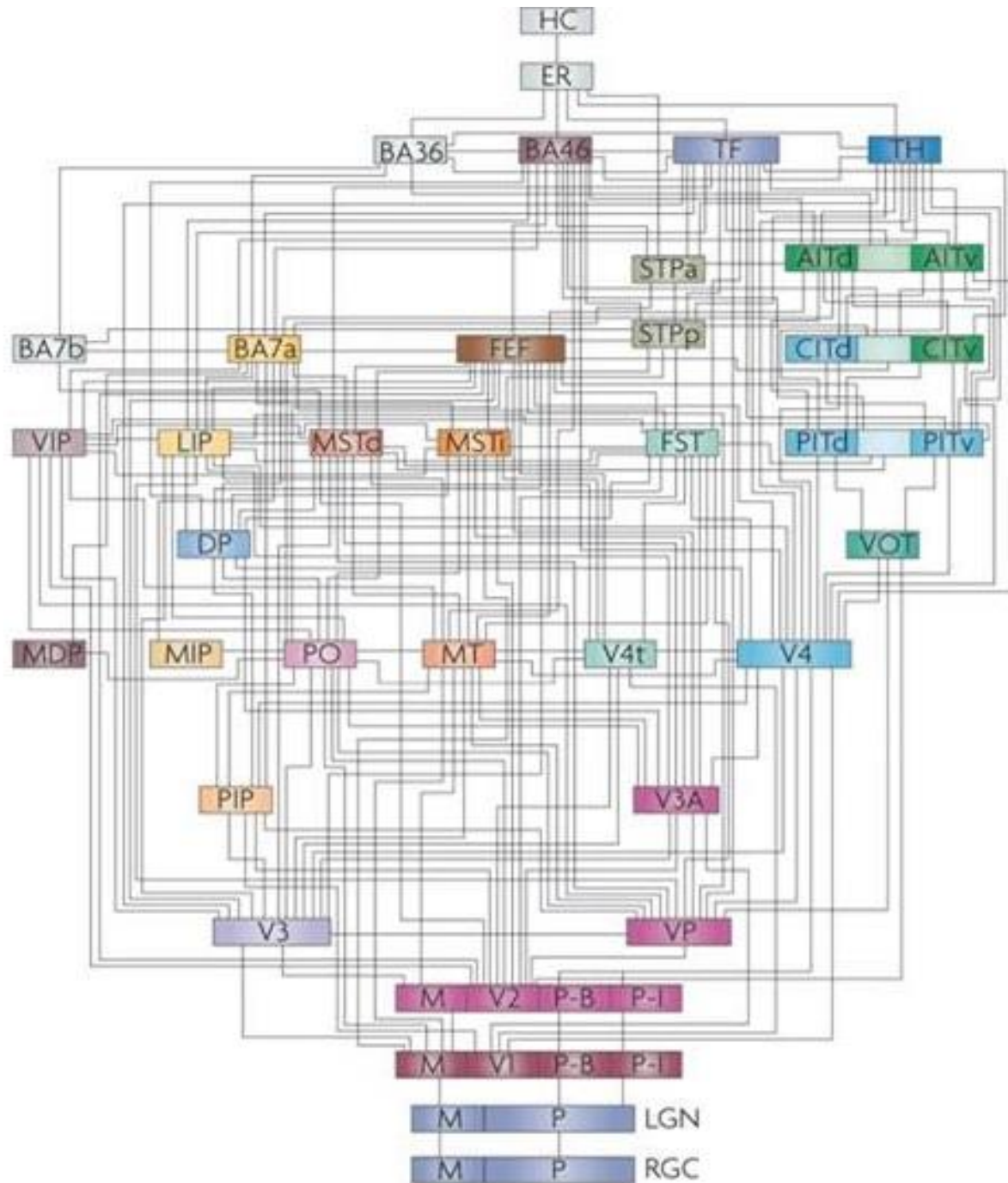
- Today there are races and other challenges for unmanned autonomous systems driving a car, which are capable to drive at 50 km/h, avoid obstacles, obey traffic rules, etc.
- For instance see DARPA urban and grand challenge (2007):
 - <http://www.youtube.com/watch?v=-xibwwNVLgg> (40 min)
 - <http://www.youtube.com/watch?v=735hAuQ0tAM> (2.5 min)
- The Defense Advanced Research Projects Agency (DARPA) is an agency of the U.S. Department of Defense responsible for the development of new technologies for use by the military. They pioneer development of many military technologies involving AI.
 - On the positive side (ARPANET) was one of the world's first operational packet switching networks, and the progenitor of the global Internet.

Neural networks for visual recognition

- All these AI systems driving a car have algorithms for visual recognition
- People are very good at recognizing shapes
 - It's intrinsically difficult and computers are bad at it
- Our perceptual systems are very good at dealing with *invariances*
 - Translation invariance, rotation invariance, scaling invariance
 - deformation, contrast, lighting, rate
 - ...
- We are so good at this that its hard to appreciate how difficult it is.
 - Its one of the main difficulties in making computers perceive.
 - We still don't have generally accepted solutions...

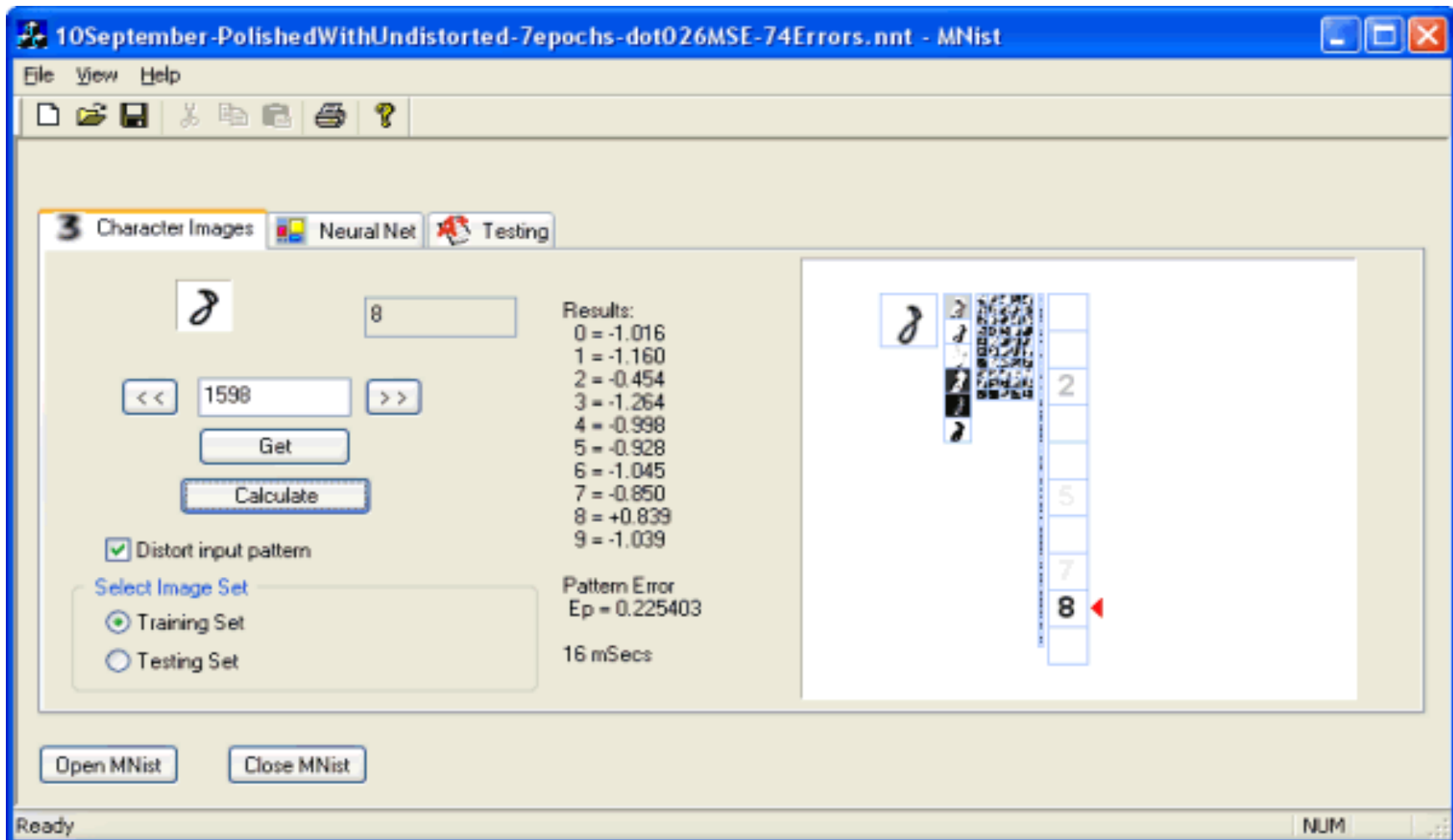
Visual system of the brain

- Here is a map of forward and recurrent neural connectivity between all the **about 40 visual** brain areas.
- Figure source: Felleman, S. J., Essen, D. C. Distributed hierarchical processing in the primate cerebral cortex. Cerebral Cortex vol. 1, 1–46 (1991).
- The visual system does everything – shape, motion, where, what, colour, 3D, etc.



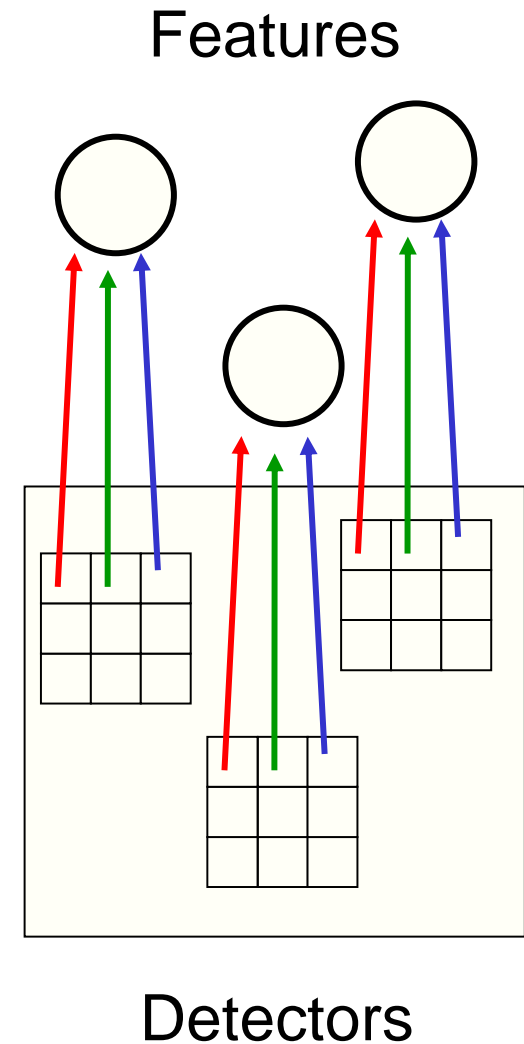
Applying MLP to shape recognition

- Task – to teach an MLP to recognise single handwritten characters.
- Source of following ideas & images is the prize winner in competition MFC/C++ 2006: <http://www.codeproject.com/Articles/16650/Neural-Network-for-Recognition-of-Handwritten-Digi> (by By Mike O'Neill, 2006)



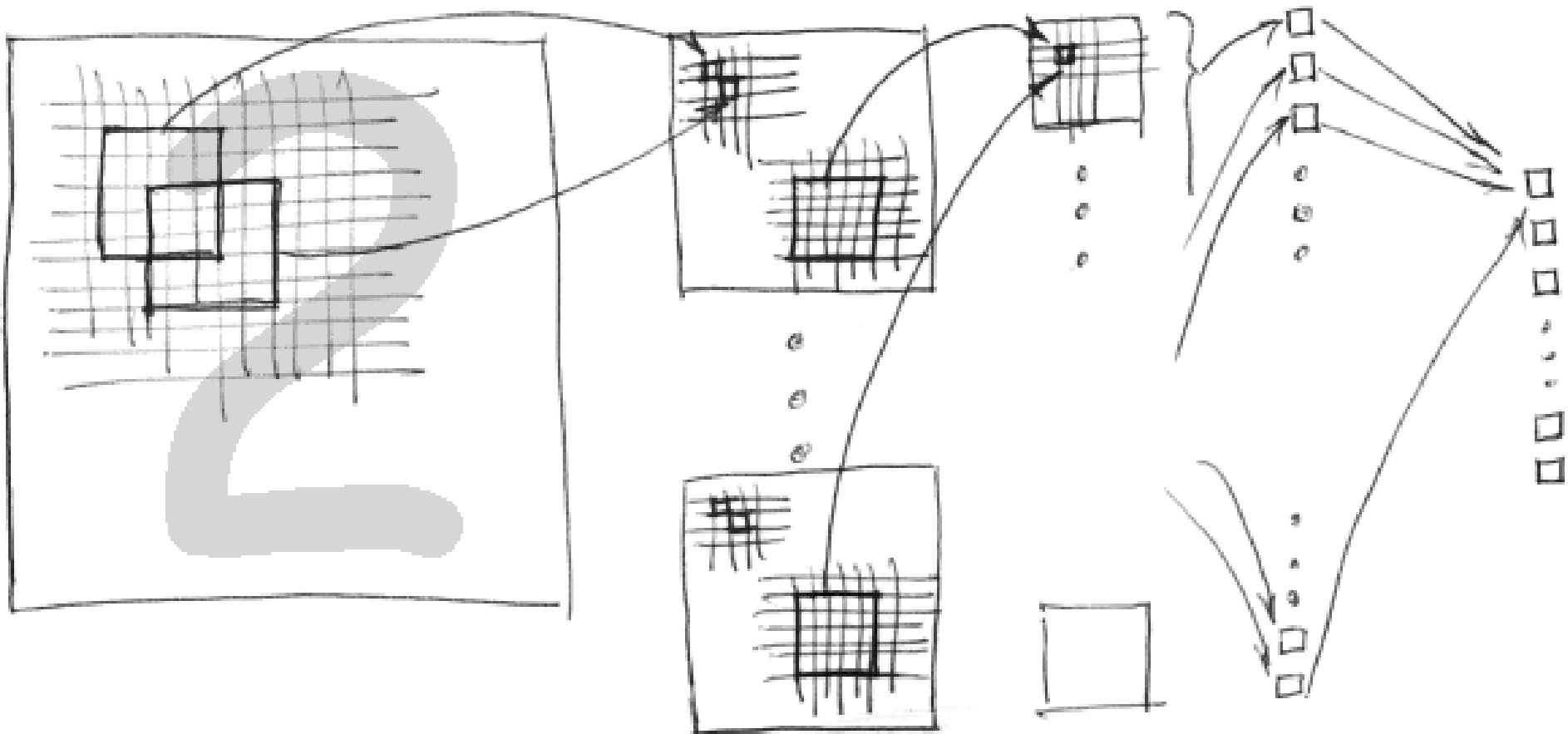
The replicated feature approach

- What we know is that the brain uses many different copies of the same feature detector.
 - The copies all have slightly different positions.
 - Could also replicate across scale and orientation.
- The brain also uses several different feature types, each with its own replicated pool of detectors.
 - Allows each patch of image to be represented in several ways.



Convolutional neural network

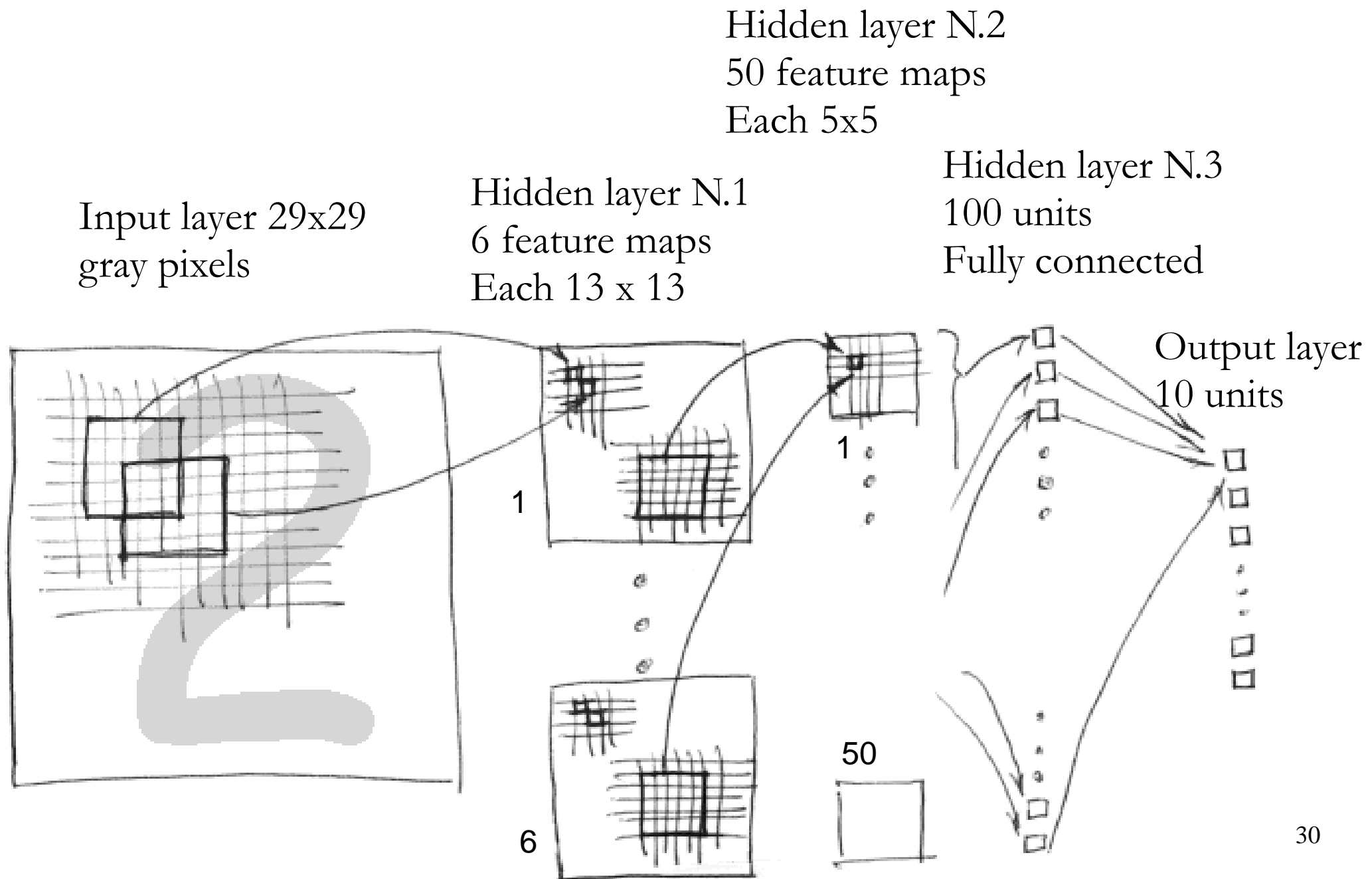
- This MLP is a five-layer convolutional neural network. The input layer takes grayscale data of a 29x29 image of a handwritten digit, and the output layer is composed of ten neurons of which exactly one neuron has a target value of +1 corresponding to the answer while all other nine have target value of -1.



Shared weights

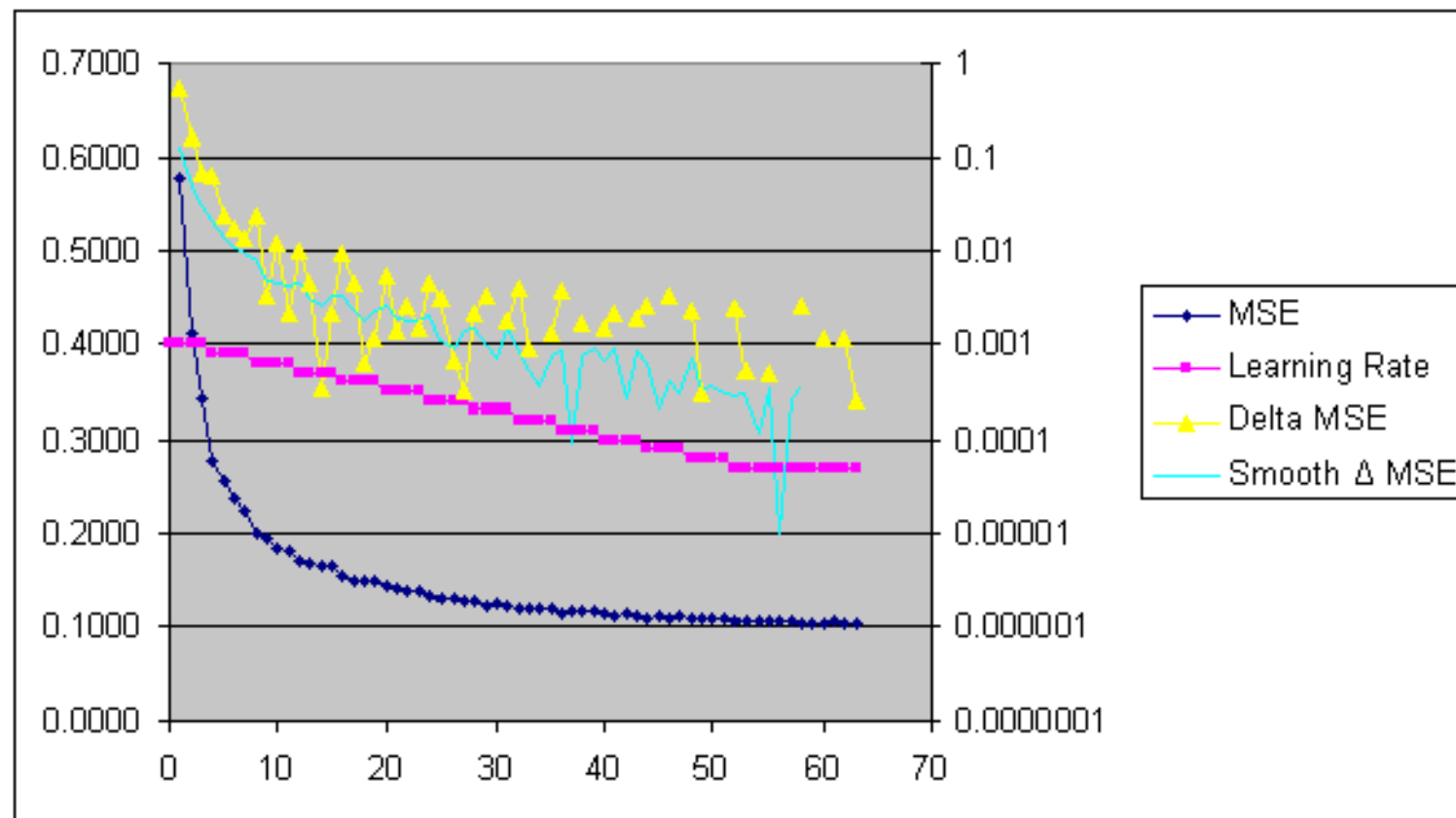
- These networks are also known as “shared weight” neural networks.
- The idea is that a small kernel window is moved over neurons from a prior layer. In this network, a kernel is sized to 5x5 elements.
- Each element in the 5x5 kernel window has a weight independent of that of another element, so there are 25 weights (plus the bias term).
- This kernel is shared across all elements in the prior layer, hence the name “shared weight”.
- The full description and code can be found at:
<http://www.codeproject.com/Articles/16650/Neural-Network-for-Recognition-of-Handwritten-Digi>

Architecture of layers

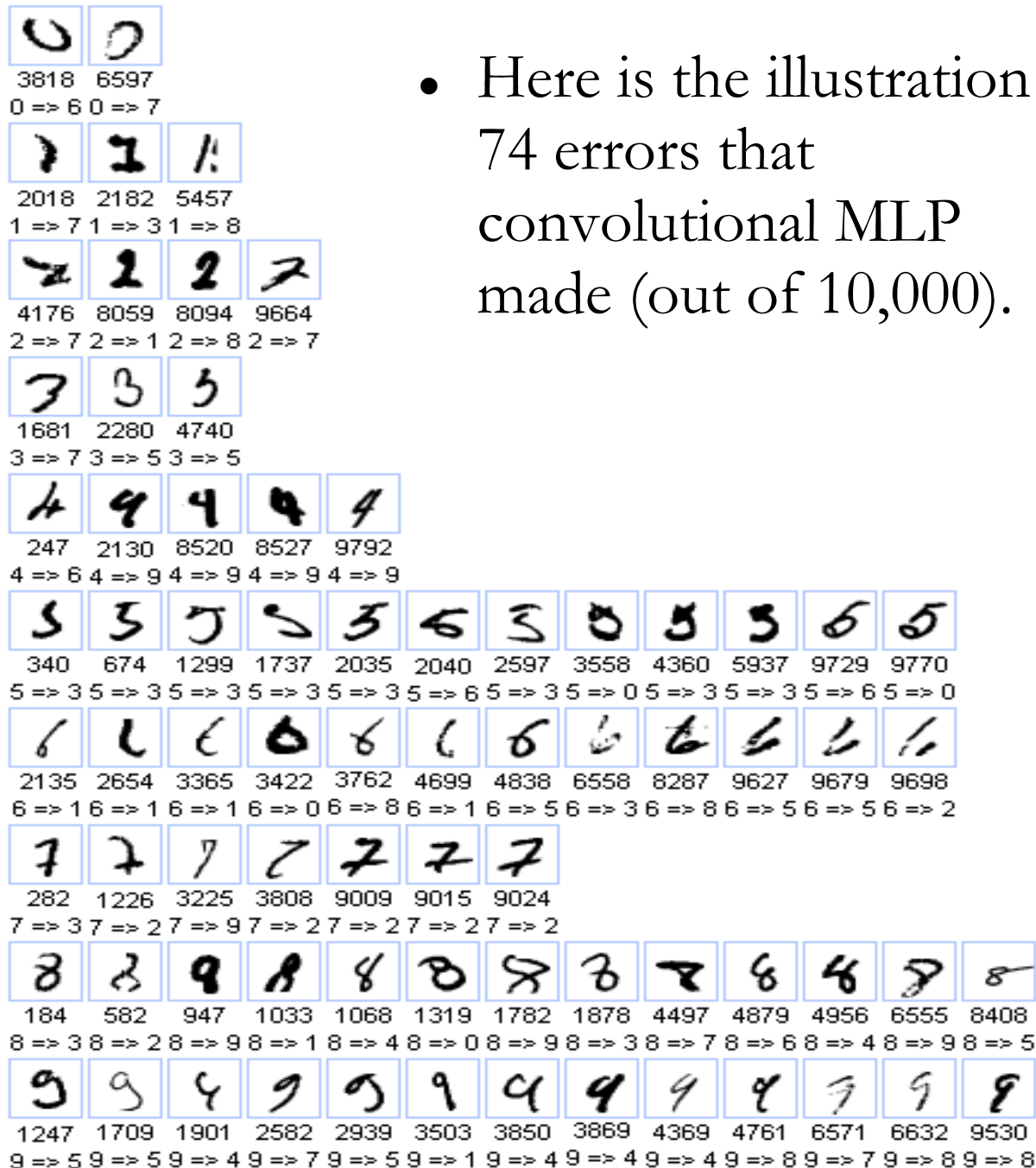


Results of training and testing

- The plot shows the evolution of the MSE (the mean squared error E) during training. X-axis is the number of epochs. Note, the learning rate is not constant is purposefully decreased.
- After training, there were only 74 mis-recognitions out of the 10,000 patterns in the testing set, for an error rate of 0.74%.



Errors in testing

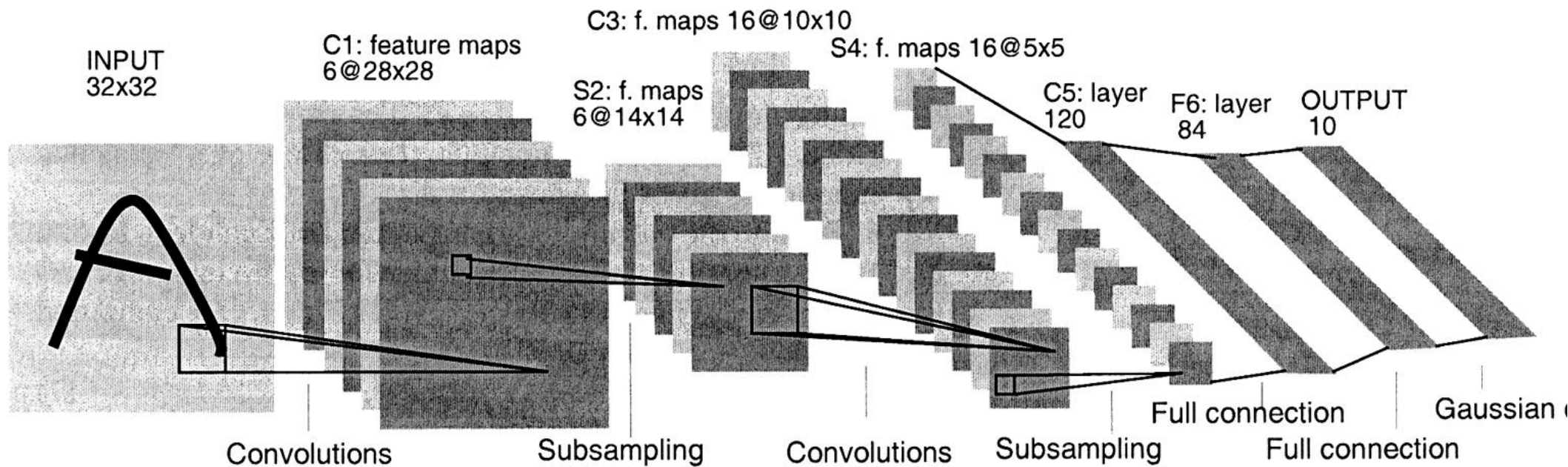


- Here is the illustration of 74 errors that convolutional MLP made (out of 10,000).

LeNet for digit recognition

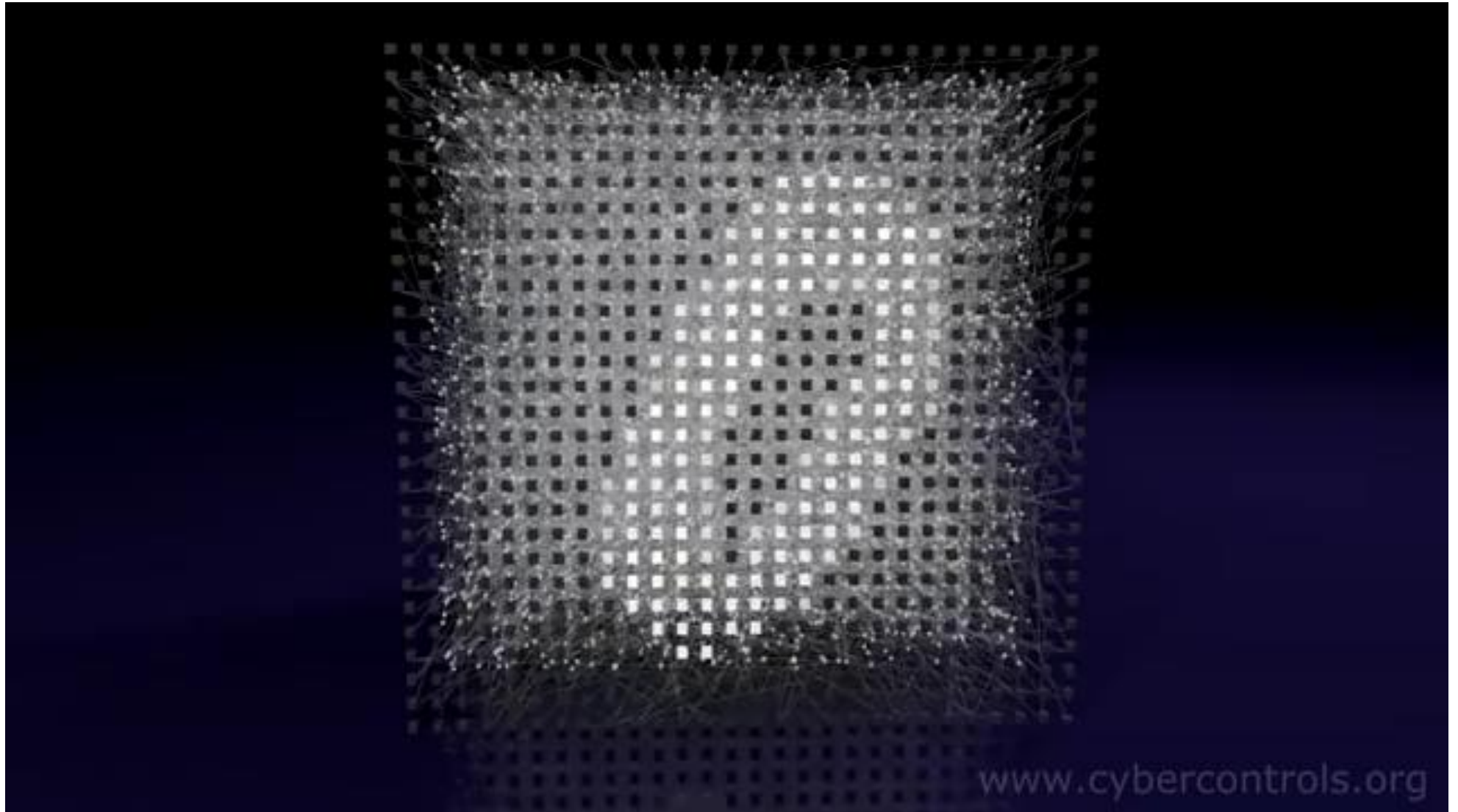
- This ANN was inspired by the work of Patrice Simard et al. & Yann LeCun, who is the Chief AI Scientist at Facebook/Meta.
- Yann LeCun and co-workers (in 1989) developed a really good recognizer for handwritten digits by using back-propagation in a feed-forward net with:
 - Many hidden layers
 - Many pools of replicated units in each layer.
 - Averaging of the outputs of nearby replicated units.
 - Net can cope with several characters at once even if they overlap.
- Look at LeCun's AI stuff at <http://yann.lecun.com>

LeNet – architecture and training



- LeNet has 1256 units and 64,660 connections
- After 23 epochs the success rate = 98.6% on the train set of 7291 handwritten digits and 95% on the test set consisting of 2007 new digits.

Animation



Summary

- Geoffrey E. Hinton (<http://www.cs.toronto.edu/~hinton/>) is credited for pioneering the development of a general model called **deep belief network** (which the convolutional architectures are specific cases of).
- More details at http://www.scholarpedia.org/article/Deep_belief_networks
- Numerous applications of MLP, DBN and other ANN in many areas:
 - Bioinformatics
 - Physics
 - Computer science
 - Engineering
 - Transportation
 - Telecommunications
 - Economics
 - ...