CERN

[redacted]

Cours/Lecture Series

1993 – 1994 ACADEMIC TRAINING PROGRAMME

LECTURE SERIES

NEW DATES

SPEAKER	:	F. JAMES / CERN–CN
TITLE	:	Introduction to Neural Networks
TIME	:	31 January, 2 , 3 & 4 February from 11.00 to 12.00 hrs
PLACE	:	Auditorium

ABSTRACT

In this series of 4 lectures, the emphasis will be on solving real problems using multi-layer feed-forward networks. Using the general theory of inverse problems and recent theoretical results on computational complexity in neural networks, we try to develop ways of understanding in what sense a problem is solvable and what network architecture is necessary to solve it.

1. *Introduction and overview of Artificial Neural Networks.*

2, 3. *The Feed-forward Network as an Inverse Problem, and results on the computational complexity of network training.*

4. *Physics applications of neural networks.*

1994 / DSU-CP-CO
Distr. int. & ext.
200/700

February 8, 1994

CERN
ACADEMIC TRAINING PROGRAMME

Introduction to NEURAL NETWORKS

F. JAMES

Lectures given 31 Jan. - 4 Feb., 1994

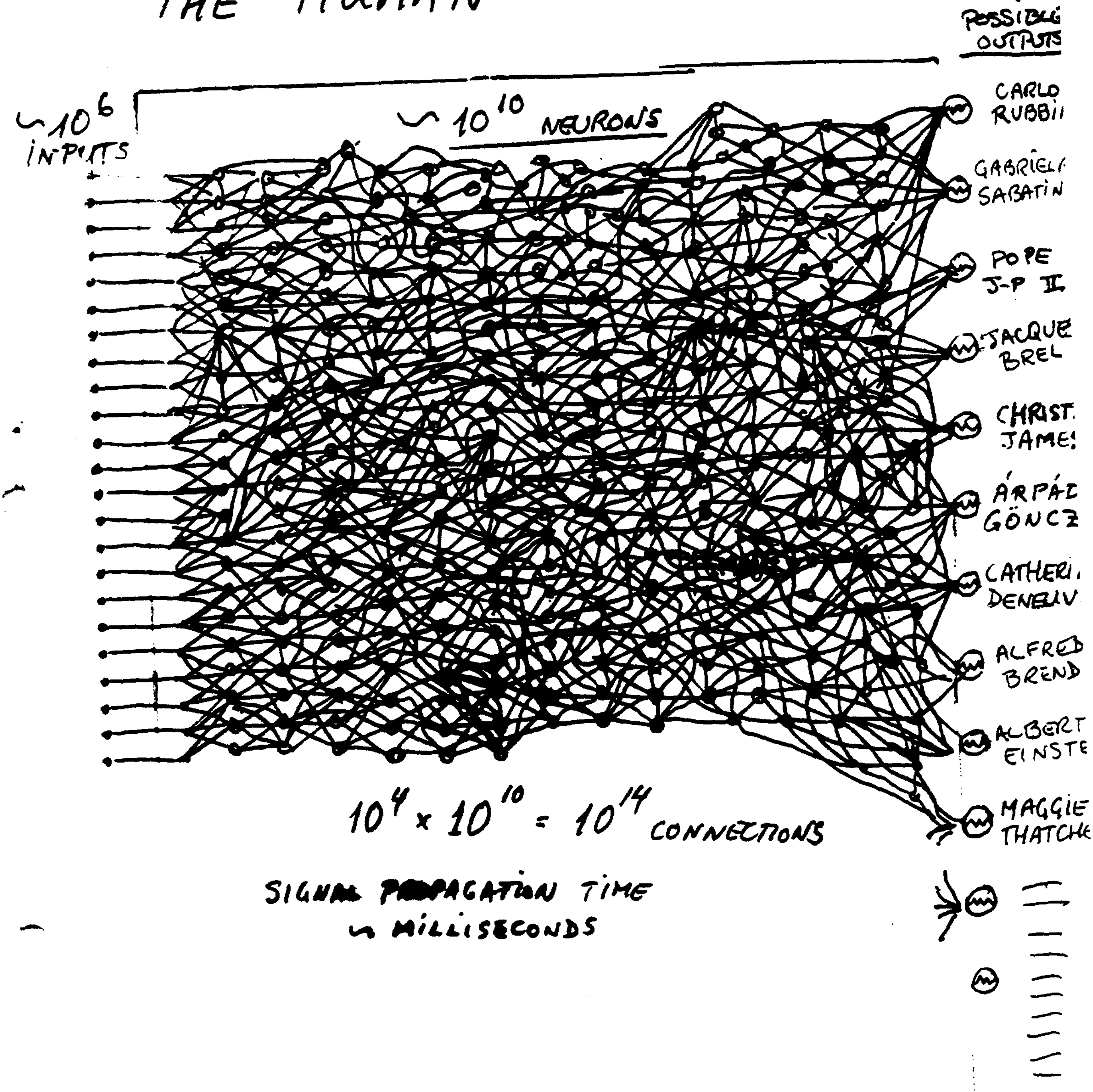
THE VON NEUMANN COMPUTER

— SEQUENTIAL EXECUTION OF INSTRUCTIONS FROM A STORE CONTAINING INSTRUCTIONS AND DATA.

— MOST OF THE STORE IS IDLE MOST OF THE TIME.

— WAS ONCE A MODEL OF HOW THE HUMAN BRAIN WORKS (WRONG!)

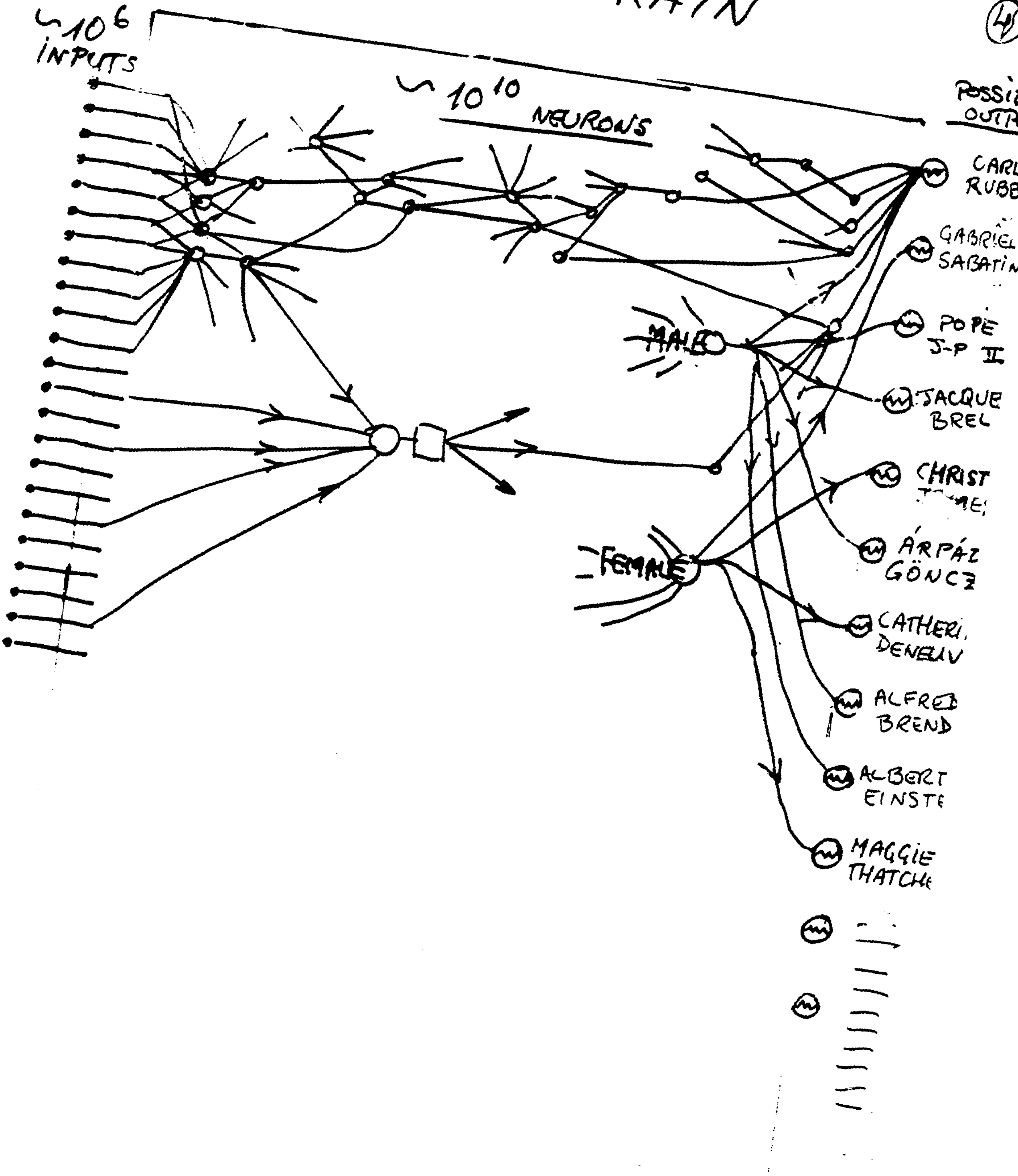
THE HUMAN BRAIN



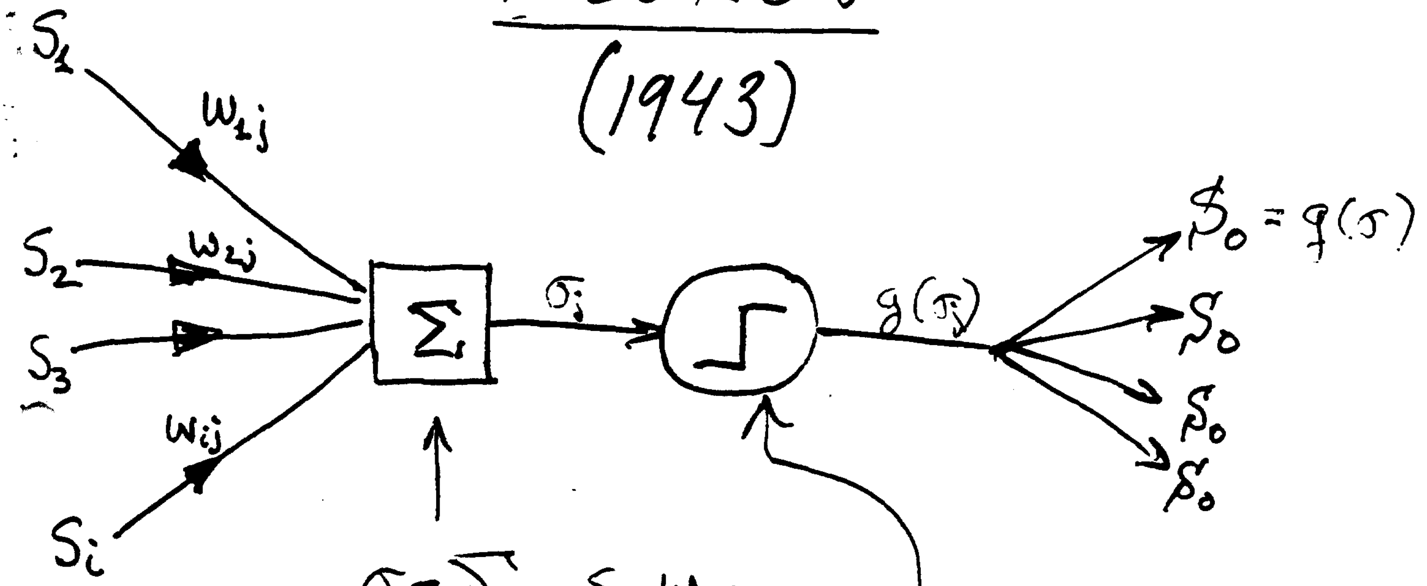
$10^4 \times 10^{10} = 10^{14}$ CONNECTIONS

SIGNAL PROPAGATION TIME
~ MILLISECONDS

THE HUMAN BRAIN



THE McCULLOCH-PITTS NEURON (1943)



$$\sigma_j = \sum_i S_i w_{ij}$$

MANY DEGREES OF FREEDOM

TRANSFER (THRESHHOLD) FUNCTION

$$g(\sigma) = \begin{cases} 1 & \text{IF } \sigma > \theta \\ 0 & \text{IF } \sigma < \theta \end{cases}$$

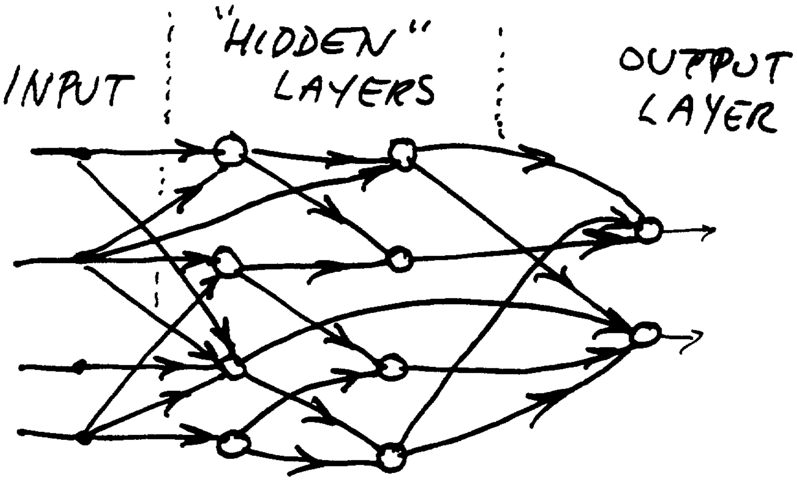
NON-LINEARITY

HISTORICAL REMARK:

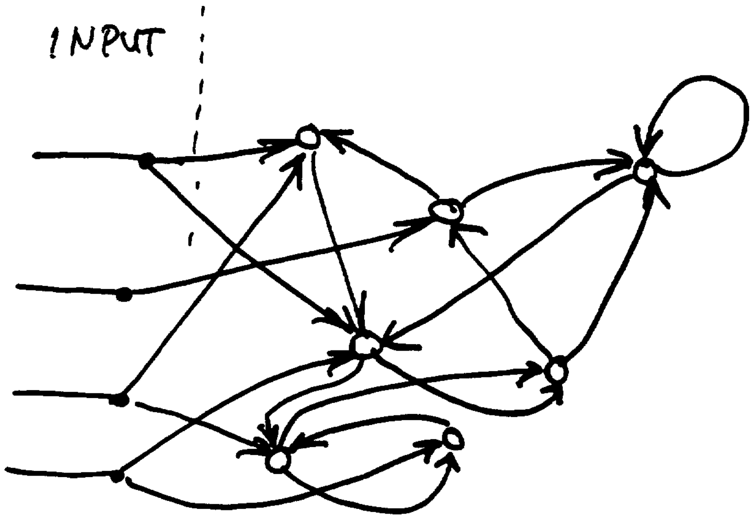
D. O. HEBB: "THE ORGANIZATION OF BEHAVIOR"

THIS EARLIEST SERIOUS STUDY OF LEARNING IN BIOLOGICAL SYSTEMS, APPEARED ONLY IN 1949

NETWORKS OF NEURONS



I. PERCEPTRON, OR FEED-FORWARD NETWORK



II. RECURRENT NETWORK

WHAT IS SO EXCITING ABOUT A.N.N.?

(1) INHERENTLY PARALLEL → FAST!

(2) CAN SOLVE MANY PROBLEMS BETTER THAN ANY OTHER KNOWN METHOD.

(3) LEARN FROM EXAMPLES - NO NEED TO UNDERSTAND

(4) UNIVERSAL PREDICTORS
- WEATHER
- THE STOCK MARKET!

(5) FAULT-TOLERANT, INSENSITIVE TO NOISE, GRACEFUL DEGRADATION

(6) IT IS SPECULATED THAT TRADITIONAL PROGRAMMING WILL BECOME ~IMPOSSIBLE AS COMPUTERS GET BIGGER

WHAT IS SO DEPRESSING ABOUT A.N.N. ?

50 YEARS LATER,

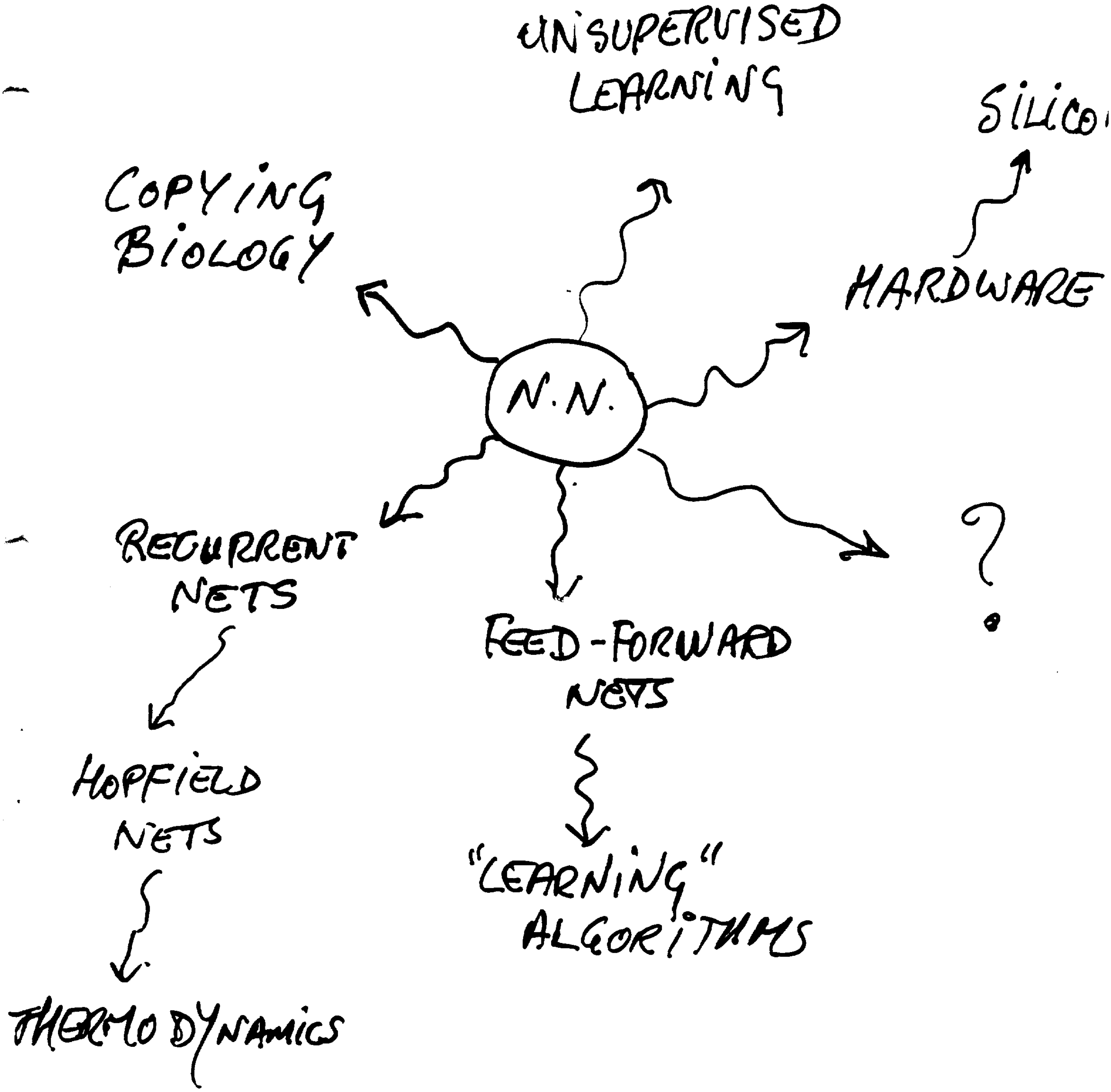
WE STILL CANNOT SOLVE

BIG PROBLEMS

(> 100 INPUTS)

WE WILL TRY TO UNDERSTAND IF THIS LIMIT IS FUNDAMENTAL

Research in Neural Networks



WHO IS DOING RESEARCH IN A. N. N. ?

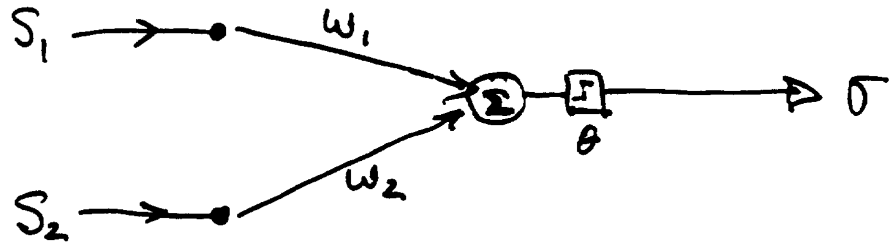
- ALMOST EVERYBODY !
- ANYBODY WITH A PC OR A MAC OR W.S. CAN MAKE HIS OWN A.N.N. TO PLAY WITH.
  SEE "MATHEMATICA" BOOK - 1994
- ENORMOUS TIME IS WASTED THIS WAY.

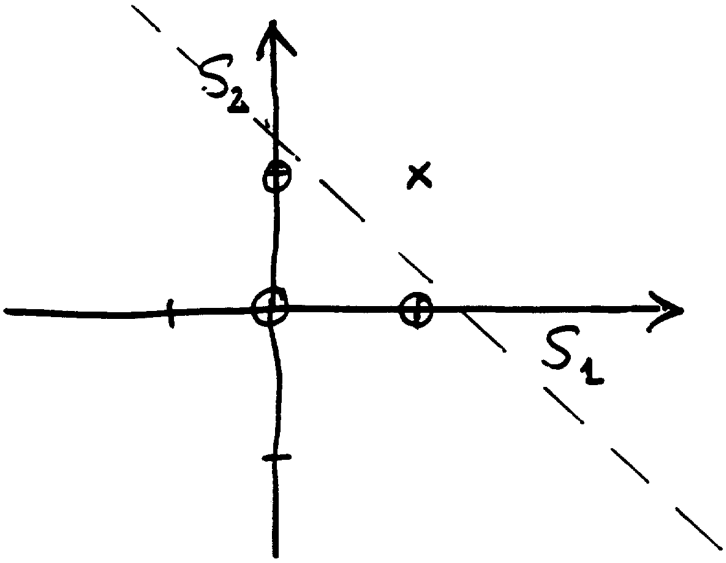
SERIOUS, ORGANIZED R&D

- NOT AT CERN !
  NO BOOKS, NO JOURNALS, NO GROUPS, NO MANDATES.
- M.I.T. IS THE "TEMPLE OF A.I. + N.N."
- EVERY COUNTRY HAS SOME REAL EXPERTS.
  USUALLY IN UNIVERSITIES
  - LUND, SWEDEN : C. PETERSON et al.
  - N.B.I., COPENHAGEN : B. LAUTRUP et al.
  - HELSINKI U. + T.U. : KOHONEN et. al.
  - ISRAËL, MANY GROUPS (D. AMIT → ROME)
- EEC SURVEY DEANNA - Report March 1993
  Database for European A.N.N. Activity

____ = PHYSICIST

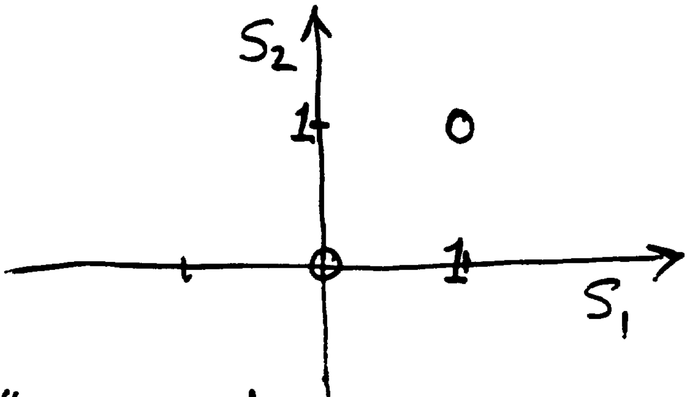
THE SIMPLEST PERCEPTRON





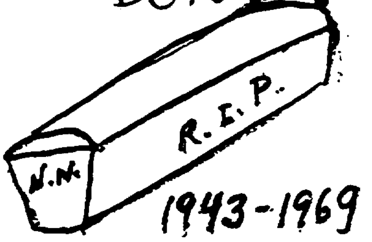
LINEARLY SEPARABLE

"AND" FUNCTION : $\begin{cases} w_1 = 3/4 \\ w_2 = 3/4 \\ \theta = 1 \end{cases}$

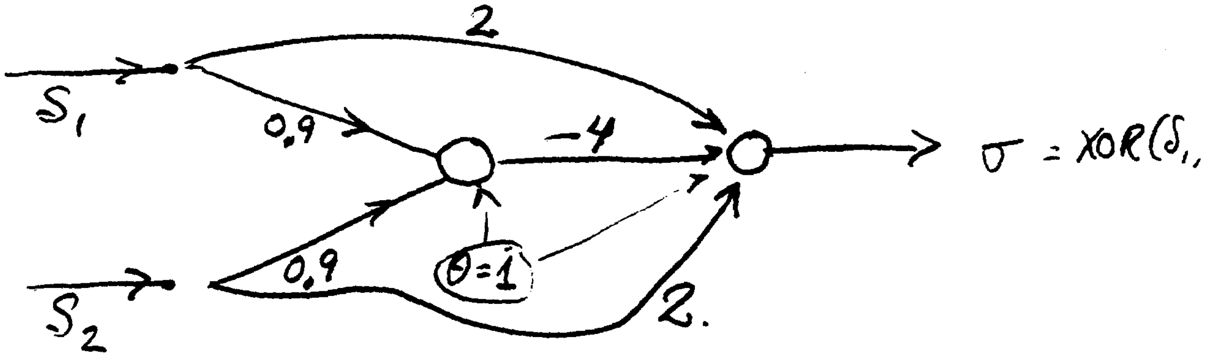


"XOR" FUNCTION

CANNOT BE DONE!



"XOR" with one "hidden layer"

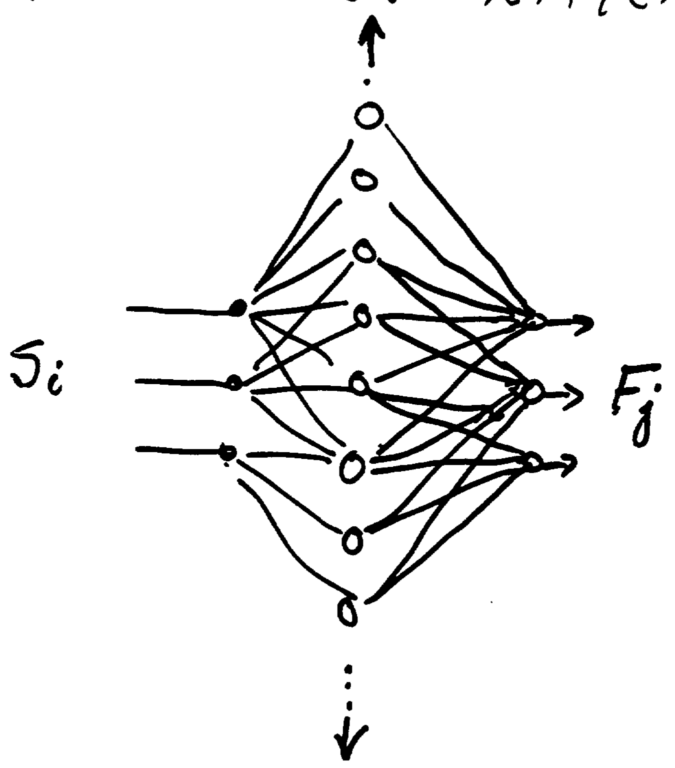


RENAISSANCE OF N.N. (~1989)

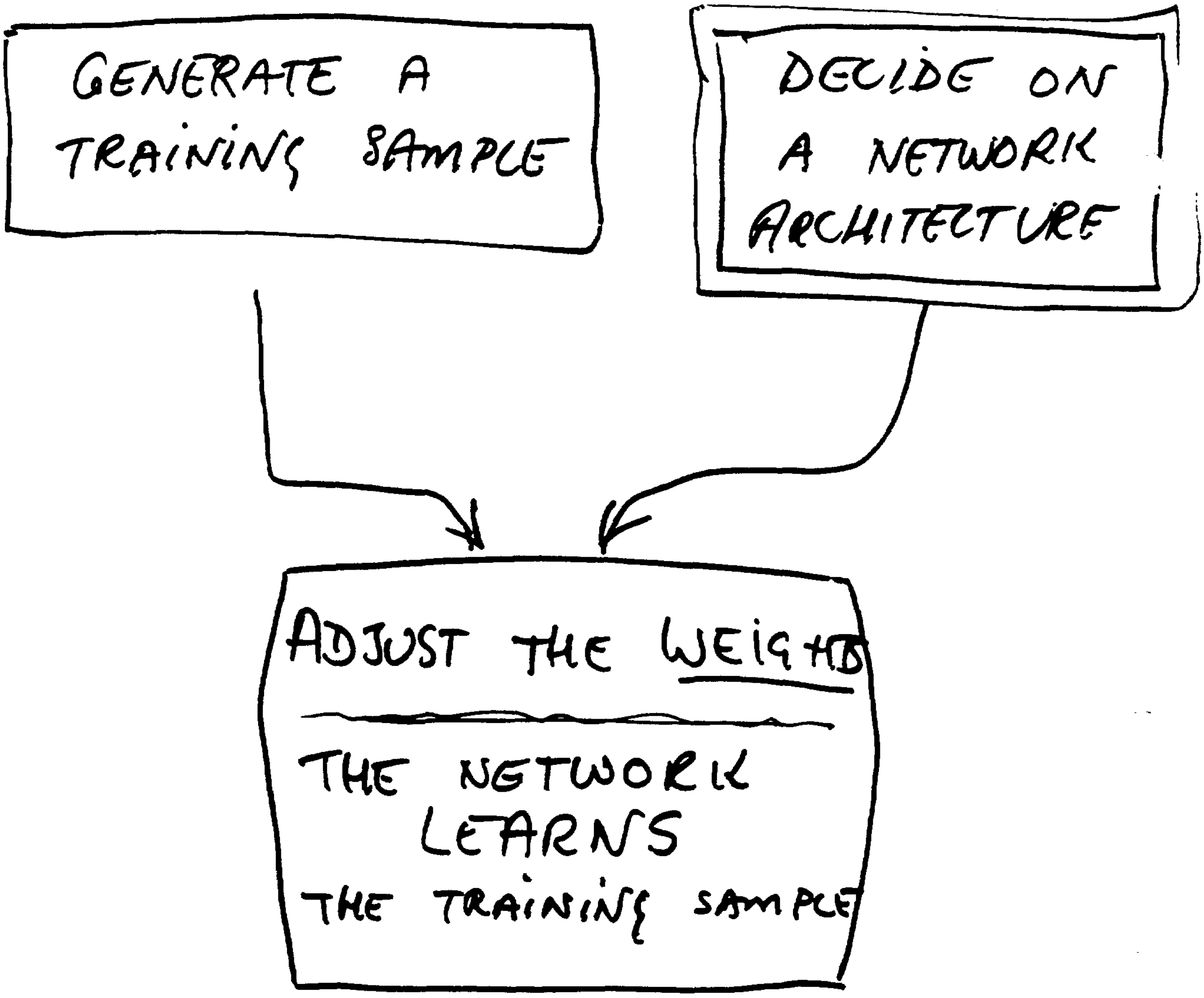
ONE "HIDDEN LAYER" CAN DO ANYTHING!

[CYBENKO, 1989 & HORNIK ET AL., 1989]

GIVEN ANY ARBITRARY INPUTS S_i AND ANY DESIRED OUTPUTS $\boxed{F_j(S_i)}$, IT IS POSSIBLE, WITH ONLY ONE HIDDEN LAYER, TO OBTAIN OUTPUTS ARBITRARILY CLOSE TO $F_j(S_i)$, PROVIDED THERE ARE ENOUGH UNITS IN THE HIDDEN LAYER.



HOW TO SOLVE A REAL PROBLEM



GENERATE A TRAINING SAMPLE

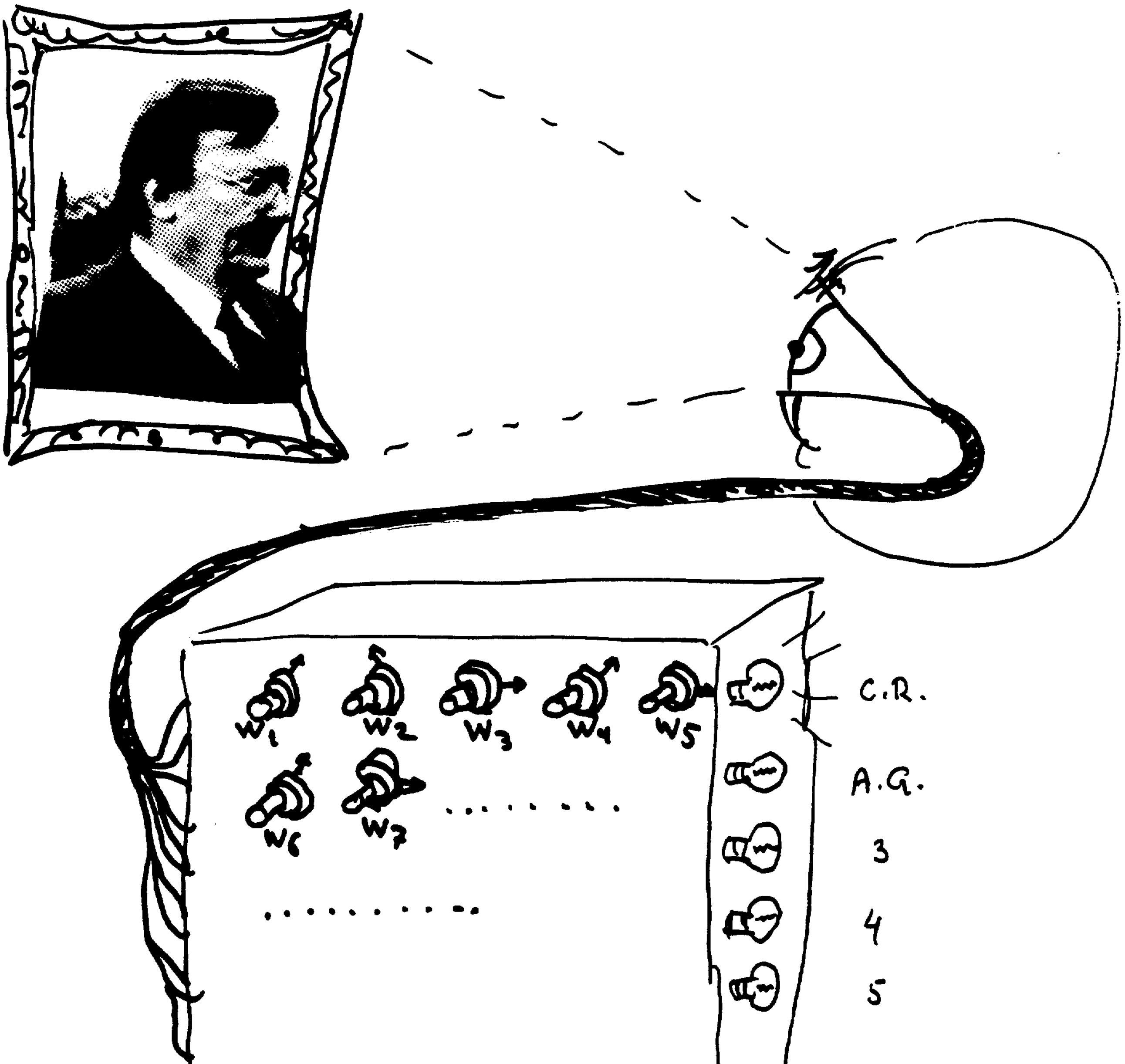
DECIDE ON A NETWORK ARCHITECTURE

ADJUST THE WEIGHTS

THE NETWORK LEARNS THE TRAINING SAMPLE

"SUPERVISED" LEARNING

How can a Network LEARN?



ADJUST ALL THE W_i

UNTIL ONLY THE RIGHT LIGHT GOES ON

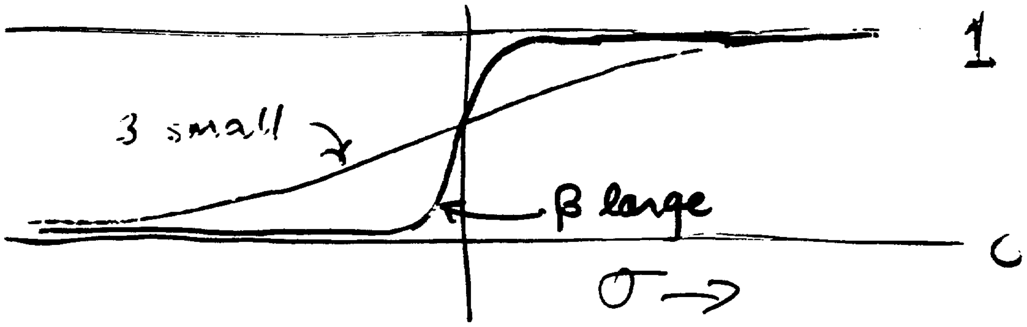
FOR LEARNING:

REPLACE SQUARE THRESHOLD

BY SIGMOID FUNCTION

(SMOOTH)

$$\boxed{g(\sigma) = \frac{1}{1 + e^{-2\beta\sigma}}} \quad (0<g<1)$$



$\frac{1}{\beta}$ is like a 'temperature'

LEARNING BY MINIMIZING CHISQUARE

(QUADRATIC COST FUNCTION)

$$\chi^2(\underline{w}) = \overbrace{\sum_k}^{\text{training sample}} \; \overbrace{\sum_j}^{\text{output neurons}} \Big(\underbrace{F_{kj}(\underline{w})}_{\text{neural net outputs}} - \underbrace{O_{kj}}_{\text{desired outputs}} \Big)^2.$$

- TO CALCULATE $\chi^2(\underline{w})$ REQUIRES

  k_{max} TRAVERSALS OF THE NETWORK

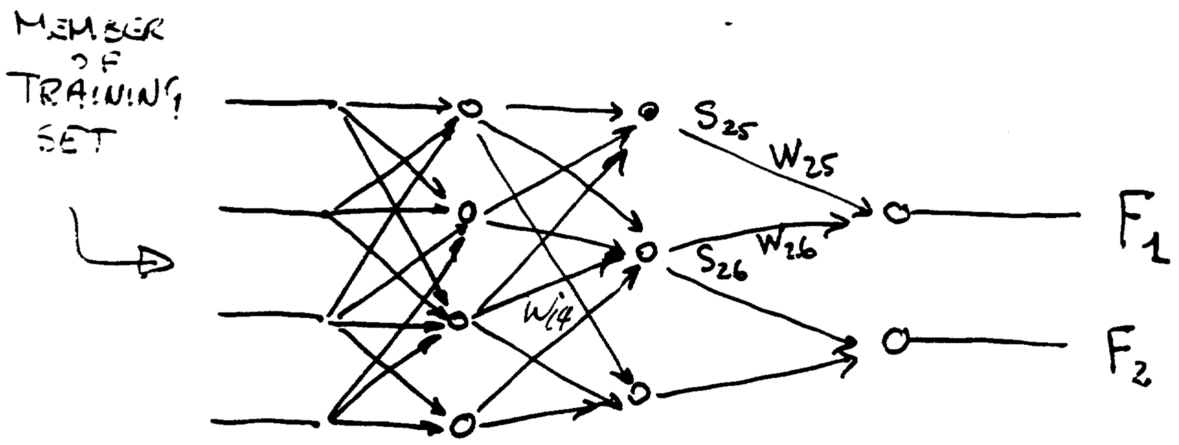
  (FEED-FORWARD NETWORK)

SIZE OF TRAINING SAMPLE.

WHAT IS REQUIRED TO CALCULATE THE DERIVATIVES

$$\frac{\partial \chi^2(\underline{w})}{\partial \underline{w}} \quad ?$$

USING THE "BACK-PROPAGATION" ALGORITHM GIVES YOU $\boxed{\text{ALL}}$ THE FIRST DERIVATIVES WITH LESS CALCULATION THAN TO GET χ^2 !



$$\chi^2 = \sum_i (F_i - O_i)^2$$

ACTUAL OUTPUTS

VECTOR OF DESIRED OUTPUTS

CONSIDER :

$$\frac{\partial}{\partial W_{26}} (F_1 - O_1)^2$$

$$= 2(F_1 - O_1) \cdot \frac{\partial F_1}{\partial w_{26}}$$

sigmoid

$$= \underbrace{2(F_1 - O_1) \cdot \frac{\partial}{\partial W_{26}} g(S_{25} W_{25} + S_{26} W_{26}}_{S_{26} \cdot g'(S_{25} W_{25} + S_{26} W_{2}}$$

BUT: $\boxed{g'(x) = 2\beta\, g(x) [1 - g(x)]}$

THE BACK-PROPAGATION ALGORITHM

1975 - 1985 -1985- 1986

(1) CALCULATE "CHI-SQUARE" - FORWARD SWEEP
THROUGH NETWORK

(2) CALCULATE ALL 1ST DERIVATIVES
BACKWARD SWEEP
THROUGH NETWORK

(3) MODIFY WEIGHTS BY AMOUNT "STEEPEST DESCENT"
PROPORTIONAL TO 1ST DERIVATIVE GRADIENT STEP
(STEP LENGTH IS A GUESS!)

(4) CALCULATE CHI-SQUARE
AT NEW POINT (NEW WEIGHTS)
IF CHISQURE GETS WORSE, CUT STEP SIZE
UNTIL CHISQUARE IMPROVES.

(5) GO TO (2)

THIS *NON-LINEAR MINIMIZATION*
TECHNIQUE IS VERY PRIMITIVE,
BUT IT SOMETIMES WORKS.

15

Computer Physics Communications 70 (1992) 167–182
North-Holland

Computer Physics
Communications

Pattern recognition in high energy physics with artificial neural networks – JETNET 2.0

Leif Lönnblad, Carsten Peterson and Thorsteinn Rögnvaldsson
Department of Theoretical Physics, University of Lund, Sölvegatan 14 A, S-223 62 Lund, Sweden

Received 27 August 1991

A F77 package of adaptive artificial neural network algorithms, JETNET 2.0, is presented. Its primary target is the high energy physics community, but it is general enough to be used in any pattern-recognition application area. The basic ingredients are the multilayer perceptron back-propagation algorithm and the topological self-organizing map. The package consists of a set of subroutines, which can either be used with standard options or be easily modified to host alternative architectures and procedures.

PROGRAM SUMMARY

Title of program: JETNET version 2.0

Catalogue number: ACGV

Program obtainable from: CPC Program Library, Queen's University of Belfast, N. Ireland (see application form in this issue)

Licensing provisions: none

Computer for which the program is designed: DECstation, SUN, Apollo, VAX, IBM and others with a F77 compiler
Computer: DECstation 3100; *Installation*: Department of Theoretical Physics, University of Lund, Sweden

Operating system under which the program has been tested: ULTRIX RISC 4.2

Programming language used: FORTRAN 77

Memory required to execute with typical data: ~ 90 kwords

No. of bits in a word: 32

Peripherals used: terminal for input, terminal or printer for output

No. of lines in distributed program, including test deck data, etc.: 3345

Keywords: pattern recognition, jet identification, artificial neural network

Nature of physical problem
High energy physics offers many challenging pattern-recognition problems. It could be separating photons from leptons based on calorimeter information or the identification of a quark based on the kinematics of the hadronic fragmentation products. Standard procedures for such recognition problems is the introduction of relevant cuts in the multi-dimensional data.

Method of solution
Artificial neural networks (ANN) have turned out to be a very powerful paradigm for automated feature recognition in a wide range of problem areas. In particular feed-forward multilayer networks are widely used due to their simplicity and good performance. JETNET 2.0 implements such a network with the back-propagation updating algorithm in

16

LU TP 93-29
CERN-TH.7135/94
December 1993

JETNET 3.0 – A Versatile Artificial Neural Network Package

Carsten Peterson and Thorsteinn Rögnvaldsson

Department of Theoretical Physics, University of Lund,
Sölvegatan 14 A, S-223 62 Lund, Sweden

Leif Lönnblad

Theory Division, CERN, CH 1211 Geneva 23, Switzerland

Submitted to **Computer Physics Communications**

PROGRAM SUMMARY

Title of Program: JETNET version 3.0

Catalogue number:

Program obtainable from: denni@thep.lu.se or via anonymous ftp from `thep.lu.se` in directory pub/`Jetnet/` or from `freehep.scri.fsu.edu` in directory `freehep/analysis/jetnet`.

Computer for which the programme is designed: DEC Alpha, DECstation, SUN, Apollo, VAX, IBM, Hewlett-Packard, and others with a F77 compiler

Computer: DEC Alpha 3000; installation: Department of Theoretical Physics, University of Lund, Lund, Sweden

Operating system: DEC OSF 1.3

ALTERNATIVES TO THE QUADRATIC COST FUNCTION χ^2

WHY χ^2 ? $= \sum_{kj} \left(F_{kj}(w) - O_{kj} \right)^2$

- TRADITIONAL, 'NATURAL', WELL-KNOWN
- OPTIMAL STATISTICAL PROPERTIES WHEN FITTING GAUSSIAN-DISTR. DATA (OR POISSON)

BAD PROPERTY FOR A.N.N.:

- AS WEIGHTS GET VERY FAR FROM BEST SOLUTION, χ^2 DOES NOT GET VERY Bi

WORST $\chi^2 \rightarrow$ constant, not ∞

(can get stuck)

THERE EXIST OTHER POSSIBILITIES

IN PARTICULAR: (DERIVED FROM INFORMATION THEORY)

$$E = \sum_{kj} \left[(1 + O_{kj}) \log\left(\frac{1 + O_{kj}}{1 + F_{kj}(w)} \right) + (1 - O_{kj}) \log\left(\frac{1 - O_{kj}}{1 - F_{kj}(w)} \right) \right.$$

THEN $E \rightarrow 0$ as $F_{kj}(w) \rightarrow O_{kj}$

BUT $E \rightarrow \infty$ as $F_{kj}(w) \rightarrow$ FAR FROM O_{kj}

taking $-1 \leq F_{kj}, O_{kj} \leq +1$

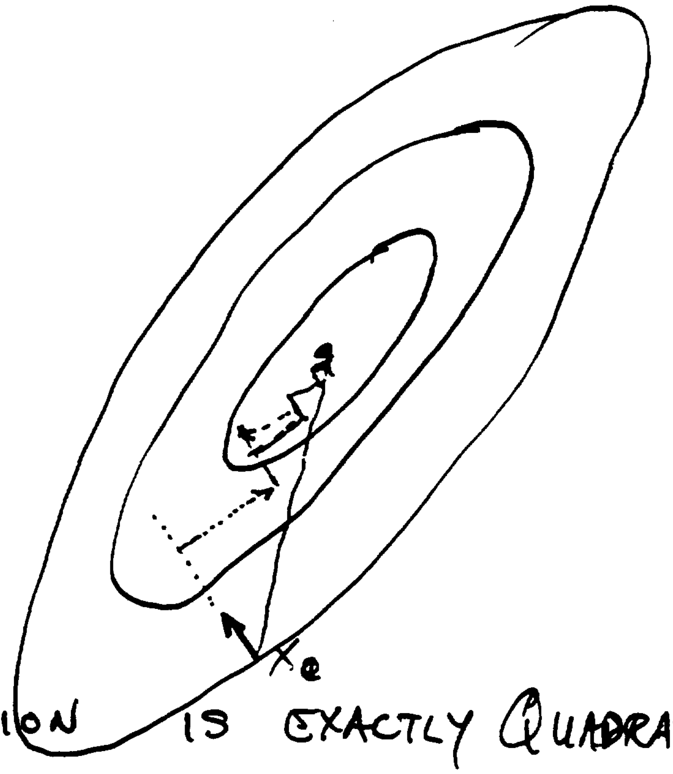
HOW BEST TO MINIMIZE "QUADRATIC" COST χ^2 ?

KNOWING ALL THE 1ST DERIV'S. IS A BIG HELP - USE IT WELL!

STEEPEST DESCENT

1) DOESN'T KNOW THE RIGHT STEP SIZE

2) CAN BE VERY SLOW, EVEN WITH THE RIGHT STEP SIZE.



(AND EVEN IF THE FUNCTION IS EXACTLY QUADRATIC!

TO IMPROVE IT

YOU NEED "SECOND-ORDER" INFORMATION

$$\underline{x}_1 = \underline{x}_0 - \underset{\sim}{G}^{-1} \cdot \underset{\sim}{g}$$

second-derivative matrix

gradient vector.

THIS GETS YOU STRAIGHT TO THE MINIMUM IF QUADRATIC!

<u>BUT</u> METHODS USING SECOND DERIVATIVE ARE **UNSTABLE**

IF $\underset{\sim}{G} \simeq 0$, $\underset{\sim}{G}^{-1} \cdot g \simeq \infty$

INFINITE STEP

IF $\underset{\sim}{G}$ NEGATIVE, $\underset{\sim}{G}^{-1} \cdot g$ GOES UPHILL

STEP IN OPPOSITE DIRECTION

ANOTHER __BUT__ : FULL SECOND-ORDER DERIVATIVES ARE TOO EXPENSIVE

$$\text{TIME \& STORAGE} \sim \frac{N^2}{2}$$

NUMBER OF CONNECTIONS

- THE PROGRAM __MINUIT__ DOES THIS - VERY POWERFUL ALGORITHMS USING G.
  GOOD FOR VERY SMALL PROBLEMS (<20 variables,
  SOPHISTICATED "ERROR ANALYSIS" - NOT NEEDED.

FOR VERY BIG N.N.,

- __AVOID__ ANYTHING THAT SCALES LIKE N^2.

HOW TO USE 2ND-ORDER INFORMATION WITHOUT $\left\{\begin{matrix}\text{CALCULATING}\\ \text{STORING}\end{matrix}\right\}$ N x N MATRIX?

CONJUGATE GRADIENTS METHOD.

(JETNET 3.0)

CONJUGATE GRADIENT METHOD
(FLETCHER + REEVES, ~1960)

TWO DIRECTIONS $\underline{d}_1$ and $\underline{d}_2$

ARE SAID TO BE CONJUGATE w.r.t. $\underset{\sim}{G}$

IF $\boxed{\underline{d}_1 \underset{\sim}{G} \underline{d}_2 = 0}$

CONSIDER A QUADRATIC FUNCTION

WITH $\dfrac{\partial^2 F}{\partial x_i \partial x_j} = \underset{\sim}{G}$ $\quad i, j = 1, \dots N$

FLETCHER + REEVES SHOWED THAT:

(1) YOU WILL MINIMIZE F EXACTLY WITH N LINEAR MINIMIZATIONS IF THE N LINES (DIRECTIONS) ARE CONJUGATE w.r.t. $\underset{\sim}{G}$

(2) IT IS POSSIBLE TO FIND CONJUGATE DIRECTIONS WITHOUT EVALUATING $\underset{\sim}{G}$ EXPLICITLY

CONJUGATE GRADIENT ALGORITHM

$$\underline{\Delta x} = \underline{x}_1 - \underline{x}_0$$ vector from x_0 to x_1

$$\underline{\Delta} g = g_1 - g_0$$ change in gradient

IF FUNCTION IS QUADRATIC:

$$\underset{\sim}{G} = \frac{\underline{\Delta} g}{\underline{\Delta x}} \Rightarrow \underline{\Delta g} = \underset{\sim}{G}\,\underline{\Delta x}$$

THEN ANY VECTOR d_1 ORTHOGONAL TO $\underline{\Delta} g$ IS:

$$\underline{d_1} \cdot \underline{\Delta} g = 0 = \underline{d_1}\,\underset{\sim}{G}\,\underline{\Delta x}$$

CONJUGATE TO Δx.

So: 1) CHOOSE $\underline{\Delta x}$ along $-g_0$ (1ST DIRECTION)

MINIMIZE ALONG $\underline{\Delta x}$ TO $\underline{x}_1, g_1$

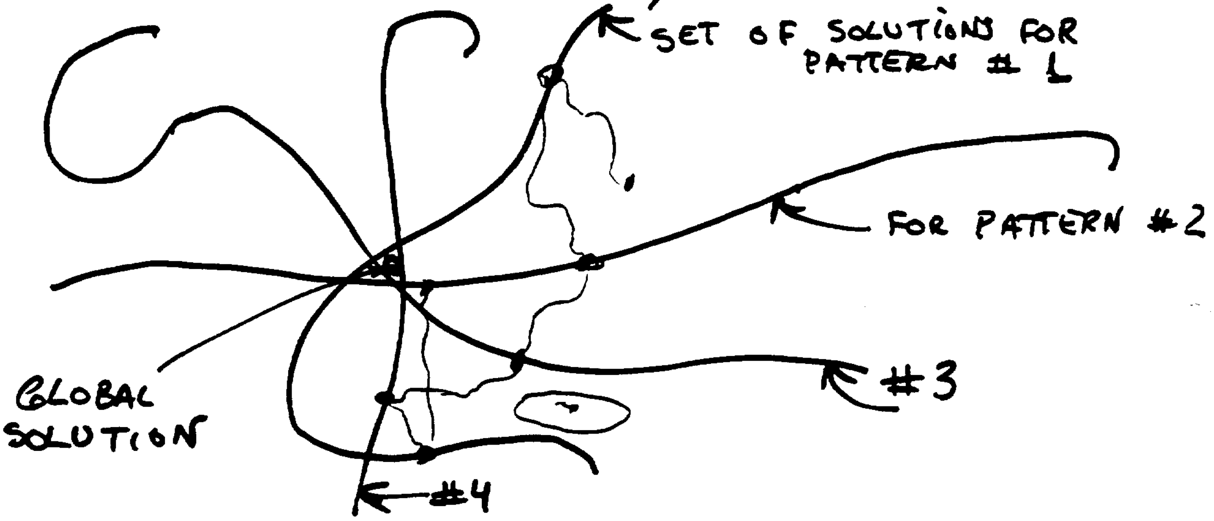
2) CHOOSE NEW $\underline{\Delta x}$ ORTHOGONAL TO $(g_1 - g_0)$ (2ND DIRECTION)

MINIMIZE ALONG NEW $\underline{\Delta x}$ TO $\underline{x}_2, g_2$

3) CONTINUE IN SAME WAY, ALL NEW DIRECTIONS WILL BE CONJUGATE

DIFFERENT STRATEGIES OF USING THE TRAINING SAMPLE:

(1) MINIMIZE χ^2 FOR EACH MEMBER (PATTERN) OF TRAINING SAMPLE, ONE BY ONE.



ONE COMPLETE CYCLE THROUGH TRAINING SAMPLE IS CALLED AN EPOCH.

(2) SUM χ^2 OVER A FEW PATTERNS (perhaps $p \sim$ member of inputs to network)

CYCLE OVER SETS (SUBSETS OF THE TRAINING SAMPLE)

(3) DEFINE χ^2 AS GLOBAL SUM OVER THE WHOLE TRAINING SAMPLE

LOCAL MINIMA ?

SIMULATED ANNEALING:

HIGH TEMPERATURE → LOW TEMPERATURE
(JUMPS OUT OF LOCAL MINIMA) (SETTLES INTO GLOBAL MINIMUM)

earning hard
most other signals in systems, even

24

ven time period represents value
vork
1024 connections)
s to transfer from input line to ouput

rately

verts summed values into output pulse

not guaranteed identical for different
ous limitation in many safety-critical

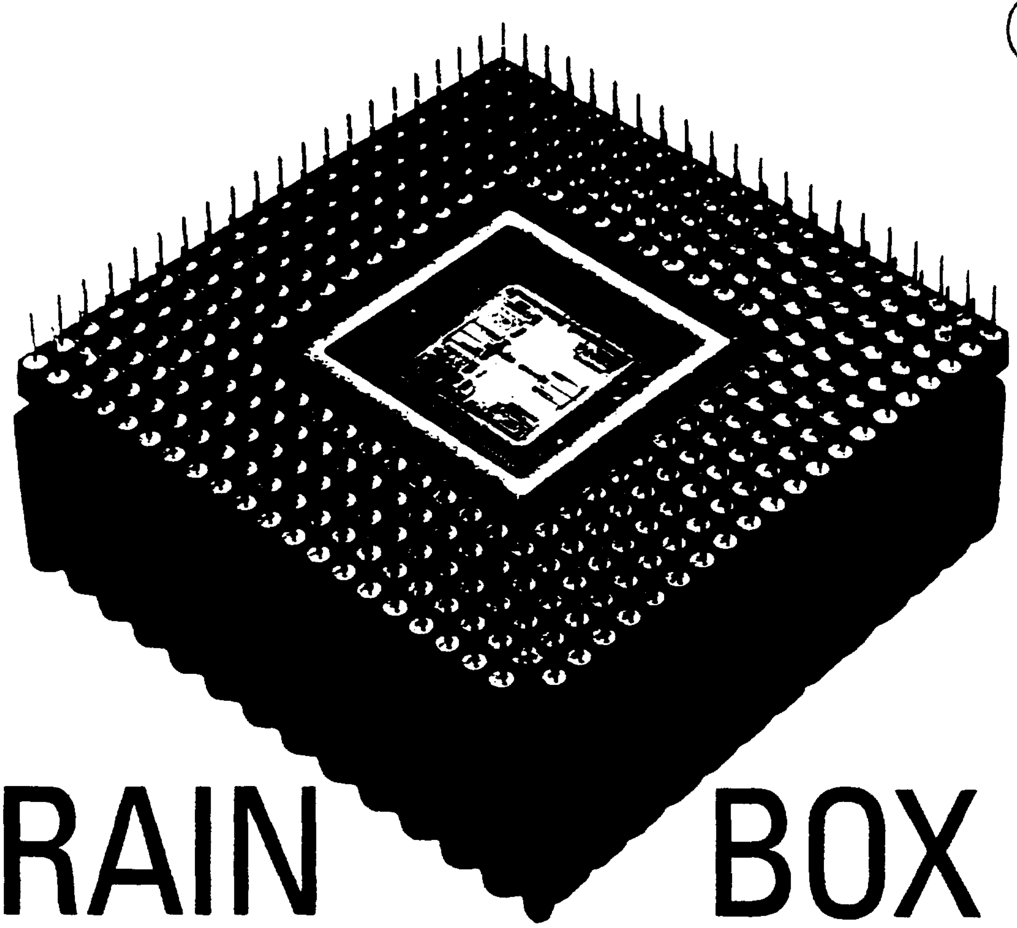
FROM:
J.G TAYLOR,
"The Promise of Neural
Networks."
Springer 1993

f) pRAMs

Hardware for a stochastic, non-linear, biologically realistic neuron
Digital devices built in 1988
Digital VLSI devices in 1990
Learning-in-hardware VLSI devices available now
Prototype analogue pRAM available now
Modular structure allows reconfigurable connectivity (ie various network architectures or genetic algorithms can be experimented with)
Interconnected modules allow the net to be expanded indefinitely
General-purpose 8-pRAM module (256 neurons) available now
Can interface to a workstation or PC

Table 3.1 Neural networks chips

Feature	Maxys Cct Tech	Intel 80170 ETANN 1991	Neural Semi	Adaptive Solutions	Advanced Micro Devices	pRAM KCL
Technology	digital	analogue	digital	digital	digital	digital
Number of Neurons	32	64	32	64	8	128
Number of Inputs	7	128	32	8	-	4
Accuracy	8-bits	~6-bits	16 levels	-	-	16-bits
Learning	Back-prop	Off-chip	Off-chip	yes	Off-chip	On-chip various
Expansion	32 modules (1024)	yes	yes	no	yes	4 links
Speed	50MHz (CLK)	200kHz	100kHz	100kHz?	50MHz (CLK)	200kHz
Cost	£10-50k	£500 chip £6000 system	-	£30k	£500 board (inc frame-grabber)	£2000
Availability	now	now	now	now	now	now



BRAIN BOX

SYNAPSE-1, designed to simulate the human brain, is the fastest neural computer in the world

By Dr. Ulrich Ramacher, Central Research and Development Department, Siemens AG, Munich, Germany

Until now, conventional computers have fallen far short in replicating what the human brain, nervous system and senses appear to achieve without any conscious effort. For example, even very powerful computers are still a long way from understanding natural language or recognizing faces. Yet automatic speech and image recognition, or making sensible guesses when knowledge is incomplete are the sort of things we will want our computers to do in the future.

Successful results have already been achieved in scientific and industrial research and development. For example, artificial neural networks can now recognize a wide range of handwritten letters and numbers, or predict the movement of exchange rates.

The major obstacle to the rapid development of new applications is the difficulty of simulating the neural learning process: the computing power required exceeds existing resources by at least five orders of magnitude.

A leap in computing power required

Even the fastest workstations take weeks - sometimes months - to perform such tasks: so a special computer is required which can simulate all types of neural networks and learning processes. Unlike computers based on the von Neumann model, a neural computer is equipped with a network of simple processors providing massively parallel processing capacity.

SYNAPSE-1 is the result of two years' work by the Central Research and Development Department at Siemens AG directed towards dramatically reducing the time needed to process neural applications. It can perform more than five billion fixed point operations (i.e. multiplying or adding 16-bit numbers) every second. With power on this scale, SYNAPSE-1 is not only 8 000 times faster than a powerful workstation, it is currently the world's fastest and

26

SIEMENS
NIXDORF

Siemens Nixdorf
Informationssysteme SA
International Accounts

Avenue des Baumettes 7
CH-1020 Renens

SIEMENS
NIXDORF

Peter Puhlmann
Dr. es sc. ingenieur dipl

Telephone (direct) 021/632 03 32
Telephone (central) 021/632 01 11
Telefax 021/635 86 82

Neurocomputer SYNAPSE-1

The capabilities of the human brain and senses are difficult to model by classical information technology. On the other hand, these capabilities like e.g. image and speech recognition, solution of complex optimization problems and especially learning capabilities become more and more important. Therefore, natural neural systems are modelled by artificial neural networks in order to tackle the above mentioned problem areas.

The structure of artificial neural networks is considerably different from the architecture of today's computer systems. Although neural networks can be simulated on conventional computers like workstations, only very simple cases can be explored because of the extremely long processing times. Especially the simulation of the neural learning process prevents fast developments of neural applications on such platforms.

Special purpose conputers that take into account the architecture of neural networks are much superior to those software solutions. The neurocomputer SYNAPSE-1 offers a performance that lies several orders of magnitude above that of a powerful workstation.

The power of SYNAPSE-1 (up to 3.2 billion connections or multiplications and accumulations per second) is resulting from a scalable multi-processor and memory architecture and from the neuro-signal-processor MA16 which executes the compute intensive operations of neural algorithms.

The system consists of the following hardware components:

- an array of 8 MA16 chips
- a data unit for non compute intensive operations
- a high-bandwidth memory
- a control unit for control and coordination.

Communication with host workstation and specialized input/output devices is implemented via a VME bus interface.

On the software side, the system is delivered with microprograms, operating software and workstation programming environment. Applications are programmed in the easy-to-use "neural Algorithms Programming Language" (nAPL) which supports the user in the development of his algorithms. nAPL is embedded in C++.

SYNAPSE-1 fulfills all the relevant requirements for a universal neurocomputer:

- compute power of at least 3 orders of magnitude above conventional computers, so that development times can be minimized

- support of small, medium and large neural networks, so that a complete area of potential applications can be implemented

- "general purpose" architecture, so that any of today's neural networks are supported and provision is made for future developments.

Technical data:

<u>MA16 array</u>
Peak performance 3.2 billion multiplications (16x16 bit) and accumulations (48 bit) per second,
Memory capacity 32 MBytes.
Transfer rates
- Backplane 0.8 GBytes/s (plus 100 MBytes/s parity)
- Data busses 0.8 GBytes/s
- Control busses 200 MBytes/s
- Address bus 56 MBytes/s,

Frequency rate 25 MHz

<u>W memory</u>
Capacity 128 MBytes

<u>Data Unit</u>
Peak performance 100 million integer operations per second,
Working memory 8 MBytes,
VME bus connection 4 Mbytes/s,
Y memory 8 MBytes,
Frequency rate 25 MHz

<u>Control Unit</u>
Peak performance sequencer 20 MIPS,
Peak performance Y address generator 25 million addresses per second,
Working memory 8 MBytes,
VME bus connection 4 MBytes/s,
Frequency rate 25 MHz

<u>Interface</u>
VME bus connection to workstation.

For more information, please contact:

Siemens Nixdorf
Informationssysteme SA
Division B
CH-1020 Renens

Tel.: (021) 632 01 11
Fax: (021) 635 86 82

LET US STEP BACK AND

CONSIDER THE GLOBAL PROBLEM

GLOBAL PROBLEM {
- CHOOSE NETWORK ARCHITECTURE
- CHOOSE SIZE OF TRAINING SAMPLE
- CHOOSE NUMBER OF NEURONS
- FIND VALUES OF WEIGHTS

AS AN <u>INVERSE</u> <u>PROBLEM</u>

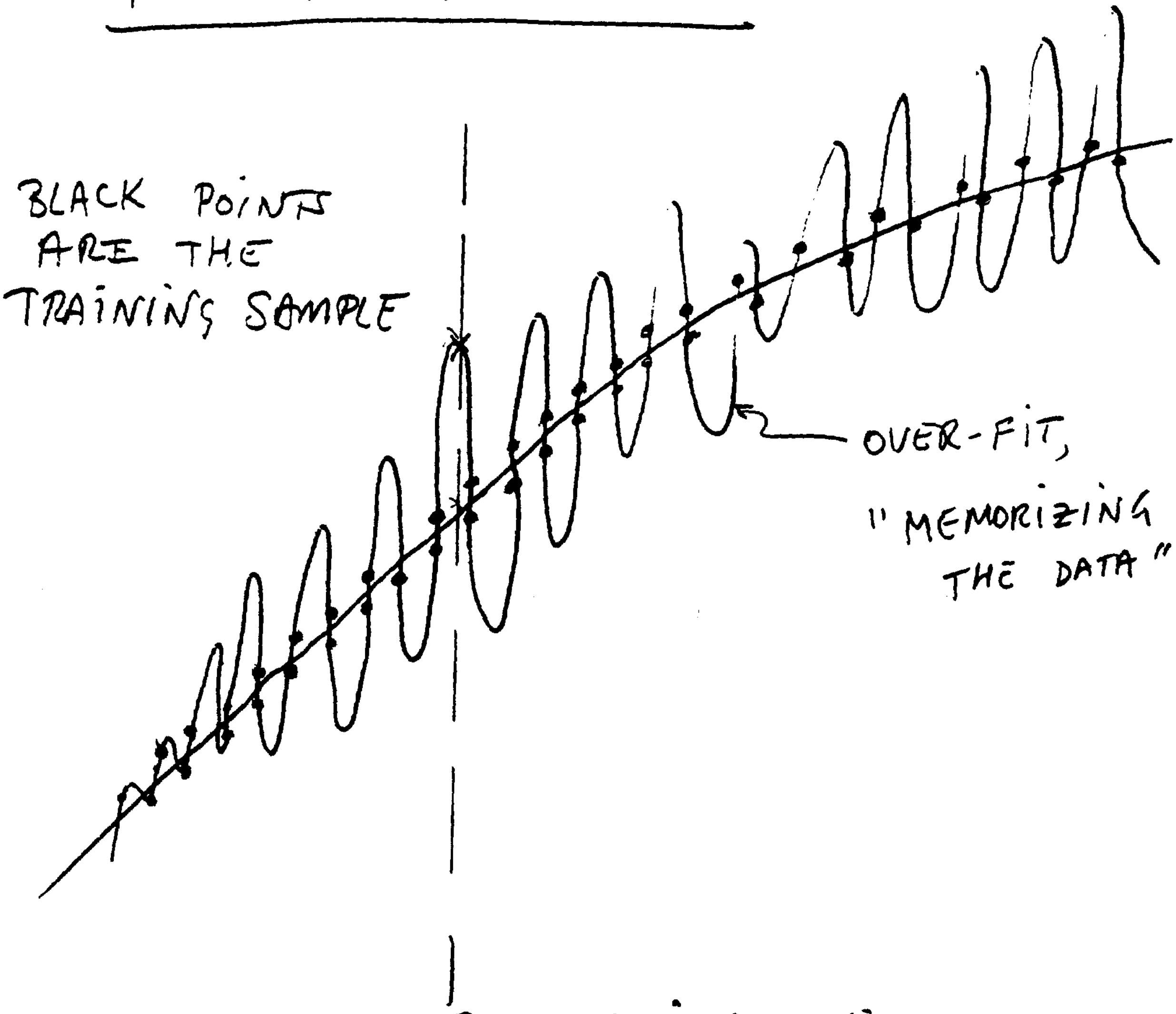
[INSPIRED BY WILLIAM PRESS, AND EXPLAINED IN: "NUMERICAL RECIPES IN FORTRAN", PAGES 795-817]

THEY ARE TREATING THE LINEAR INVERSE PROBLEM

WHEREAS <u>OURS IS NON-LINEAR</u>

BUT THERE IS MUCH IN COMMON

GENERALIZATION



BLACK POINTS ARE THE TRAINING SAMPLE

OVER-FIT, "MEMORIZING THE DATA"

GENERALIZATION

TO A PATTERN THE NETWORK HASN'T SEEN

PRESS ET AL

29

NUMERICAL RECIPES IN FORTRAN

We can combine these two points, for this conclusion: When a quadratic minimization principle is combined with a quadratic constraint, and both are positive, only *one* of the two need be nondegenerate for the overall problem to be well-posed. We are now equipped to face the subject of inverse problems.

The Inverse Problem with Zeroth-Order Regularization

Suppose that $u(x)$ is some unknown or underlying (u stands for both unknown and underlying!) physical process, which we hope to determine by a set of N measurements c_i, $i = 1, 2, \ldots, N$. The relation between $u(x)$ and the c_i's is that each c_i measures a (hopefully distinct) aspect of $u(x)$ through its own linear response kernel r_i, and with its own measurement error n_i. In other words,

$$c_i \equiv s_i + n_i = \int r_i(x)u(x)dx + n_i \tag{18.4.5}$$

(compare this to equations 13.3.1 and 13.3.2). Within the assumption of linearity, this is quite a general formulation. The c_i's might approximate values of $u(x)$ at certain locations x_i, in which case $r_i(x)$ would have the form of a more or less narrow instrumental response centered around $x = x_i$. Or, the c_i's might "live" in an entirely different function space from $u(x)$, measuring different Fourier components of $u(x)$ for example.

The *inverse problem* is, given the c_i's, the $r_i(x)$'s, and perhaps some information about the errors n_i such as their covariance matrix

$$S_{ij} \equiv \text{Covar}[n_i, n_j] \tag{18.4.6}$$

how do we find a good statistical estimator of $u(x)$, call it $\widehat{u}(x)$?

It should be obvious that this is an ill-posed problem. After all, how can we reconstruct a whole function $\widehat{u}(x)$ from only a finite number of discrete values c_i? Yet, whether formally or informally, we do this all the time in science. We routinely measure "enough points" and then "draw a curve through them." In doing so, we are making some assumptions, either about the underlying function $u(x)$, or about the nature of the response functions $r_i(x)$, or both. Our purpose now is to formalize these assumptions, and to extend our abilities to cases where the measurements and underlying function live in quite different function spaces. (How do you "draw a curve" through a scattering of Fourier coefficients?)

We can't really want every point x of the function $\widehat{u}(x)$. We do want some large number M of discrete points x_μ, $\mu = 1, 2, \ldots, M$, where M is sufficiently large, and the x_μ's are sufficiently evenly spaced, that neither $u(x)$ nor $r_i(x)$ varies much between any x_μ and $x_{\mu+1}$. (Here and following we will use Greek letters like μ to denote values in the space of the underlying process, and Roman letters like i to denote values of immediate observables.) For such a dense set of x_μ's, we can replace equation (18.4.5) by a quadrature like

$$c_i = \sum_\mu R_{i\mu} u(x_\mu) + n_i \tag{18.4.7}$$

where the $N \times M$ matrix $\mathbf{R}$ has components

$$R_{i\mu} \equiv r_i(x_\mu)(x_{\mu+1} - x_{\mu-1})/2 \tag{18.4.8}$$

30

18.4 Inverse Problems and the Use of A Priori Information

Later discussion will be facilitated by some preliminary mention of a couple of mathematical points. Suppose that $\mathbf{u}$ is an "unknown" vector that we plan to determine by some minimization principle. Let $\mathcal{A}[\mathbf{u}] > 0$ and $\mathcal{B}[\mathbf{u}] > 0$ be two positive functionals ~~of $\mathbf{u}$, so that we can try to determine~~ $\mathbf{u}$ by either

$$\text{minimize:} \quad \mathcal{A}[\mathbf{u}] \qquad \text{or} \qquad \text{minimize:} \quad \mathcal{B}[\mathbf{u}] \tag{18.4.1}$$

(Of course these will generally give different answers for $\mathbf{u}$.) As another possibility, now suppose that we want to minimize $\mathcal{A}[\mathbf{u}]$ subject to the *constraint* that $\mathcal{B}[\mathbf{u}]$ have some particular value, say b. The method of Lagrange multipliers gives the variation

$$\frac{\delta}{\delta \mathbf{u}}\left\{\mathcal{A}[\mathbf{u}] + \lambda_1(\mathcal{B}[\mathbf{u}] - b)\right\} = \frac{\delta}{\delta \mathbf{u}}\left(\mathcal{A}[\mathbf{u}] + \lambda_1\mathcal{B}[\mathbf{u}]\right) = 0 \tag{18.4.2}$$

where λ_1 is a Lagrange multiplier. Notice that b is absent in the second equality, since it doesn't depend on $\mathbf{u}$.

Next, suppose that we change our minds and decide to minimize $\mathcal{B}[\mathbf{u}]$ subject to the constraint that $\mathcal{A}[\mathbf{u}]$ have a particular value, a. Instead of equation (18.4.2) we have

$$\frac{\delta}{\delta \mathbf{u}}\left\{\mathcal{B}[\mathbf{u}] + \lambda_2(\mathcal{A}[\mathbf{u}] - a)\right\} = \frac{\delta}{\delta \mathbf{u}}\left(\mathcal{B}[\mathbf{u}] + \lambda_2\mathcal{A}[\mathbf{u}]\right) = 0 \tag{18.4.3}$$

with, this time, λ_2 the Lagrange multiplier. Multiplying equation (18.4.3) by the constant $1/\lambda_2$, and identifying $1/\lambda_2$ with λ_1, we see that the actual variations are exactly the same in the two cases. Both cases will yield the same one-parameter family of solutions, say, $\mathbf{u}(\lambda_1)$. As λ_1 varies from 0 to ∞, the solution $\mathbf{u}(\lambda_1)$ varies along a so-called *trade-off curve* between the problem of minimizing $\mathcal{A}$ and the problem of minimizing $\mathcal{B}$. Any solution along this curve can equally well be thought of as either (i) a minimization of $\mathcal{A}$ for some constrained value of $\mathcal{B}$, or (ii) a minimization of $\mathcal{B}$ for some constrained value of $\mathcal{A}$, or (iii) a weighted minimization of the sum $\mathcal{A} + \lambda_1\mathcal{B}$.

The second preliminary point has to do with *degenerate* minimization principles. In the example above, now suppose that $\mathcal{A}[\mathbf{u}]$ has the particular form

$$\mathcal{A}[\mathbf{u}] = |\mathbf{A}\cdot\mathbf{u} - \mathbf{c}|^2 \tag{18.4.4}$$

for some matrix $\mathbf{A}$ and vector $\mathbf{c}$. If $\mathbf{A}$ has fewer rows than columns, or if $\mathbf{A}$ is square but degenerate (has a nontrivial nullspace, see §2.6, especially Figure 2.6.1), then minimizing $\mathcal{A}[\mathbf{u}]$ will *not* give a unique solution for $\mathbf{u}$. (To see why, review §15.4, and note that for a "design matrix" $\mathbf{A}$ with fewer rows than columns, the matrix $\mathbf{A}^T\cdot\mathbf{A}$ in the normal equations 15.4.10 is degenerate.) *However*, if we add any multiple λ times a nondegenerate quadratic form $\mathcal{B}[\mathbf{u}]$, for example $\mathbf{u}\cdot\mathbf{H}\cdot\mathbf{u}$ with $\mathbf{H}$ a positive definite matrix, then minimization of $\mathcal{A}[\mathbf{u}] + \lambda\mathcal{B}[\mathbf{u}]$ *will* lead to a unique solution for $\mathbf{u}$. (The sum of two quadratic forms is itself a quadratic form, with the second piece guaranteeing nondegeneracy.)

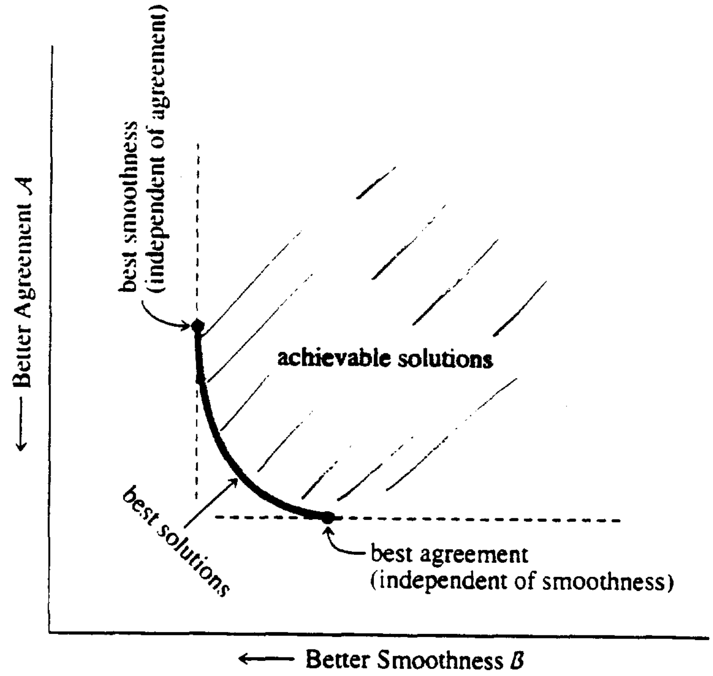

← Better Agreement $\mathcal{A}$
best smoothness
(independent of agreement)
achievable solutions
best solutions
best agreement
(independent of smoothness)
← Better Smoothness $\mathcal{B}$

Figure 18.4.1. Almost all inverse problem methods involve a trade-off between two optimizations: agreement between data and solution, or "sharpness" of mapping between true and estimated solution (here denoted $\mathcal{A}$), and smoothness or stability of the solution (here denoted $\mathcal{B}$). Among all possible solutions, shown here schematically as the shaded region, those on the boundary connecting the unconstrained minimum of $\mathcal{A}$ and the unconstrained minimum of $\mathcal{B}$ are the "best" solutions, in the sense that every other solution is dominated by at least one solution on the curve.

The value N is actually a surrogate for any value drawn from a Gaussian distribution with mean N and standard deviation $(2N)^{1/2}$ (the asymptotic χ^2 distribution). One might equally plausibly try two values of λ, one giving $\chi^2 = N + (2N)^{1/2}$, the other $N - (2N)^{1/2}$.

Zeroth-order regularization, though dominated by better methods, demonstrates most of the basic ideas that are used in inverse problem theory. In general, there are two positive functionals, call them $\mathcal{A}$ and $\mathcal{B}$. The first, $\mathcal{A}$, measures something like the agreement of a model to the data (e.g., χ^2), or sometimes a related quantity like the "sharpness" of the mapping between the solution and the underlying function. When $\mathcal{A}$ by itself is minimized, the agreement or sharpness becomes very good (often impossibly good), but the solution becomes unstable, wildly oscillating, or in other ways unrealistic, reflecting that $\mathcal{A}$ alone typically defines a highly degenerate minimization problem.

That is where $\mathcal{B}$ comes in. It measures something like the "smoothness" of the desired solution, or sometimes a related quantity that parametrizes the stability of the solution with respect to variations in the data, or sometimes a quantity reflecting *a priori* judgments about the likelihood of a solution. $\mathcal{B}$ is called the *stabilizing functional* or *regularizing operator*. In any case, minimizing $\mathcal{B}$ by itself is supposed to give a solution that is "smooth" or "stable" or "likely" — and that has nothing at all to do with the measured data.

(or any other simple quadrature — it rarely matters which). We will view equations (18.4.5) and (18.4.7) as being equivalent for practical purposes.

How do you solve a set of equations like equation (18.4.7) for the unknown $u(x_\mu)$'s? Here is a bad way, but one that contains the germ of some correct ideas: Form a χ^2 measure of how well a model $\widehat{u}(x)$ agrees with the measured data,

$$\begin{aligned}\chi^2 &= \sum_{i=1}^{N}\sum_{j=1}^{N}\left[c_i - \sum_{\mu=1}^{M} R_{i\mu}\widehat{u}(x_\mu)\right] S_{ij}^{-1}\left[c_j - \sum_{\mu=1}^{M} R_{j\mu}\widehat{u}(x_\mu)\right] \\ &\approx \sum_{i=1}^{N}\left[\frac{c_i - \sum_{\mu=1}^{M} R_{i\mu}\widehat{u}(x_\mu)}{\sigma_i}\right]^2\end{aligned} \qquad (18.4.9)$$

THE WEIGHTS

(compare with equation 15.1.5). Here $\mathbf{S}^{-1}$ is the inverse of the covariance matrix, and the approximate equality holds if you can neglect the off-diagonal covariances, with $\sigma_i \equiv (\text{Covar}[i,i])^{1/2}$.

Now you can use the method of singular value decomposition (SVD) in §15.4 to find the vector $\widehat{\mathbf{u}}$ that minimizes equation (18.4.9). Don't try to use the method of normal equations; since M is greater than N they will be singular, as we already discussed. The SVD process will thus surely find a large number of zero singular values, indicative of a highly non-unique solution. Among the infinity of degenerate solutions (most of them badly behaved with arbitrarily large $\widehat{u}(x_\mu)$'s) SVD will select the one with smallest $|\widehat{\mathbf{u}}|$ in the sense of

$$\sum_{\mu}[\widehat{u}(x_\mu)]^2 \quad \text{a minimum} \qquad (18.4.10)$$

(look at Figure 2.6.1). This solution is often called the *principal solution.* It is a limiting case of what is called *zeroth-order regularization,* corresponding to minimizing the sum of the two positive functionals

$$\text{minimize:} \quad \chi^2[\widehat{\mathbf{u}}] + \lambda(\widehat{\mathbf{u}}\cdot\widehat{\mathbf{u}}) \qquad (18.4.11)$$

in the limit of small λ. Below, we will learn how to do such minimizations, as well as more general ones, without the *ad hoc* use of SVD.

What happens if we determine $\widehat{\mathbf{u}}$ by equation (18.4.11) with a non-infinitesimal value of λ? First, note that if $M \gg N$ (many more unknowns than equations), then $\mathbf{u}$ will often have enough freedom to be able to make χ^2 (equation 18.4.9) quite unrealistically small, if not zero. In the language of §15.1, the number of degrees of freedom $\nu = N - M$, which is approximately the expected value of χ^2 when ν is large, is being driven down to zero (and, not meaningfully, beyond). Yet, we know that for the *true* underlying function $u(x)$, which has no adjustable parameters, the number of degrees of freedom and the expected value of χ^2 should be about $\nu \approx N$.

Increasing λ pulls the solution away from minimizing χ^2 in favor of minimizing $\widehat{\mathbf{u}}\cdot\widehat{\mathbf{u}}$. From the preliminary discussion above, we can view this as minimizing $\widehat{\mathbf{u}}\cdot\widehat{\mathbf{u}}$ subject to the *constraint* that χ^2 have some constant nonzero value. A popular choice, in fact, is to find that value of λ which yields $\chi^2 = N$, that is, to get about as much extra regularization as a plausible value of χ^2 dictates. The resulting $\widehat{u}(x)$ is called *the solution of the inverse problem with zeroth-order regularization.*

REGULARIZATION

- FIT THE DATA, BUT NOT NOISE
- AVOID OVERFITTING

(1) USE THE SIMPLEST NETWORK THAT WILL SOLVE THE PROBLEM

- REMOVE CONNECTIONS WITH SMALL WEIGHTS
- REMOVE NEURONS WITH SMALL OUTPUTS
- IF 2 WEIGHTS ARE HIGHLY CORRELATED, REMOVE ONE

[OPTIMAL BRAIN DAMAGE]

(2) CONSTRAIN THE WEIGHTS TO BE SMALL.

REGULARIZATION

MAKES

GENERALIZATION

INSTEAD OF PRUNING A N.N.
(BRAIN DAMAGE)

ANOTHER TECHNIQUE CONSISTS IN
SYNTHESIZING A NETWORK
STARTING FROM A MINIMAL SIZE
AND ADDING NODES.

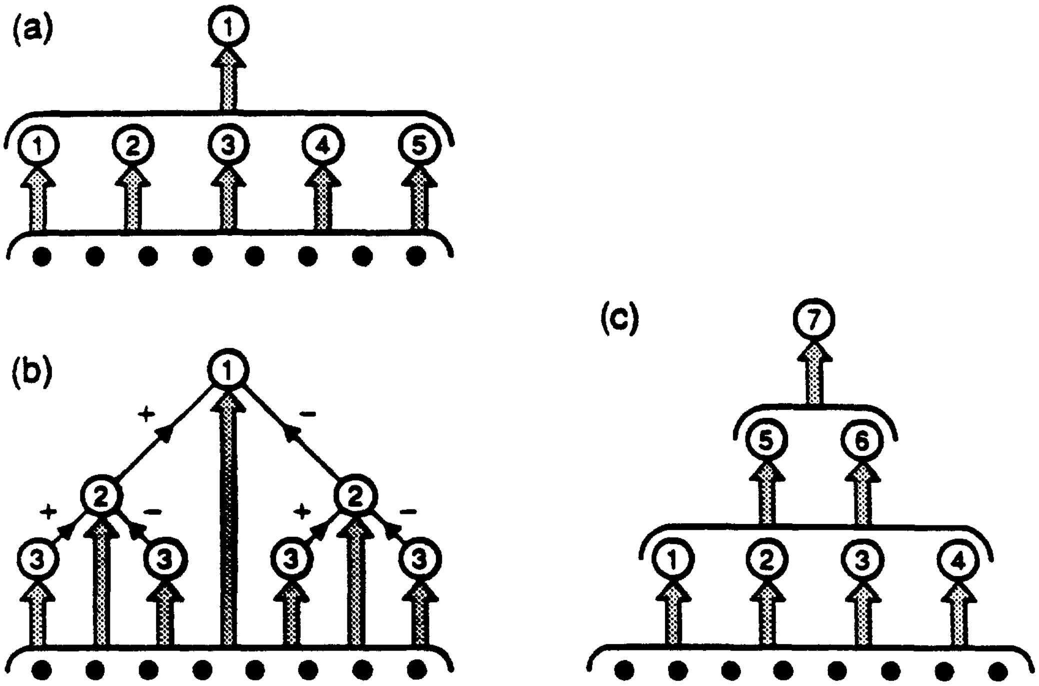
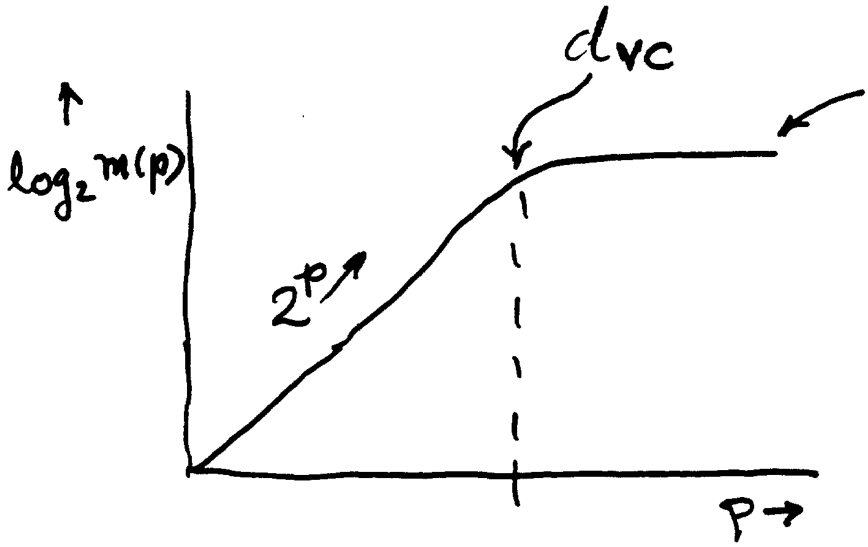


FIGURE 6.16 Network construction algorithms. The black dots are inputs, while the numbered circles are threshold units, *numbered in order of their creation.* Shaded arrows represent connections from all the units or inputs at the arrow's tail. (a) Marchand et al. [1990]. (b) Frean [1990]. (c) Mézard and Nadal [1989].

TRAINING SAMPLE SIZE p REQUIRED FOR GENERALIZATION ERROR $< \epsilon$

(SOMETIMES WE HAVE THE LUXURY OF BEING ABLE TO GENERATE A BIG SAMPLE)

THIS THEORY DEPENDS ON A PROPERTY OF THE NN-CONNECTIVITY CALLED THE:
VAPNIK-CHERVONENKIS DIMENSION OF THE NETWORK (1971)



growth rate $m(p)$ is the maximum number of different binary functions the NN could implement on a sample of size p.

TYPE OF N.N.	TRAINING SAMPLE SIZE p FOR ERROR ϵ
SINGLE PERCEPTRON WITH N INPUTS	$p > \dfrac{8N\log N}{\epsilon^2}$
FEED-FORWARD N.N. WITH M NODES, W weights	$p \leq \dfrac{W}{\epsilon}\log\dfrac{M}{\epsilon}$
FULLY-CONNECTED HIDDEN LAYER	$p \sim \dfrac{W}{\epsilon}$

HOW ACCURATE DO THE WEIGHTS HAVE TO BE?

THESE INTERESTING THEORETICAL RESULTS REFER TO N.N. WITH INTEGER WEIGHTS w_i

(1) $$w_i \leq (n+1)^{(n+1)/2} \Big/ 2^n \qquad i=0,\dots n$$

(2) $$w_i \geq n^{n/2} \Big/ 2^n \qquad i=0,\dots n$$

WHAT CAN WE LEARN FROM THE THEORY OF COMPUTATIONAL COMPLEXITY?

(1) HOW CAN WE EXPECT LEARNING TIME TO INCREASE AS A FUNCTION OF N.N. SIZE?

EXPONENTIALLY

COMPLEXITY THEORY SAYS: THIS IS AN Np - COMPLETE PROBLEM

THAT MEANS:

(1) IT IS AT LEAST AS HARD TO SOLVE AS SOME OTHER Np - COMPLETE PROBLEM

(2) NO ONE KNOWS HOW TO SOLVE ANY Np - COMPLETE PROBLEM IN POLYNOMIAL TIME

POLYNOMIAL TIME MEANS:

TIME INCREASES LIKE N^k for some finite k.

(2.1) FOR LARGE N.

(2.2) IN WORST CASE

(2.3) FOR EXACT (BEST) SOLUTION

EXPONENTIAL GROWTH MEANS: $T \gtrsim 2^{\sqrt{N}}$

100 neurons ⟶ 200 neurons

$$2^{\sqrt{100}} = 2^{10} \sim 1000 \text{ times longer.}$$

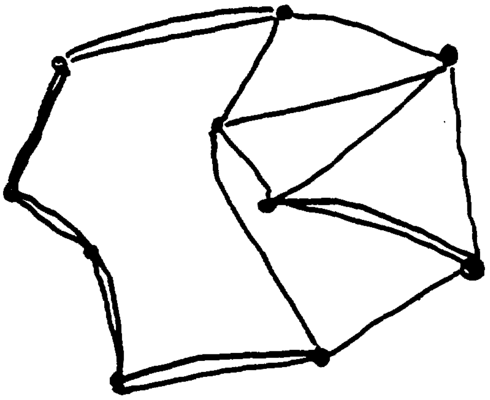
Np-complete problems are INSOLUBLE for large N.

BUT NP-COMPLETENESS IS ONLY

- IN WORST CASE
- FOR EXACT SOLUTION

EXAMPLE: THE FAMOUS TRAVELING SALESMAN:

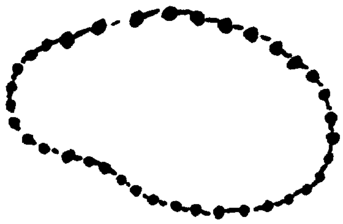
T.S. MUST VISIT ALL CITIES ONCE, MINIMIZING TOTAL TRAVELING.



BAD CASE

NP-COMPLETE,

BUT SOMETIMES <u>EASY</u>:



EASY CASE

SOMETIMES APPROXIMATE SOLUTIONS EXIST:

1. USING THE MINIMAL SPANNING TREE, A SOLUTION CAN BE FOUND FOR T.S.P. in $O(N^2)$ WHICH IS LESS THAN $\frac{3}{2}$ X LENGTH OF BEST SOLUTION.
2. INSERTION TECHNIQUE, $\sim \Theta(N^2)$

$$\frac{L_{\text{worst-case}}}{L_{\text{optimal}}} \leq \log_2(N) + 1$$

MANY OTHER METHODS GIVE GOOD APPROXIMATIONS IN POLYNOMIAL TIME

THE LESSON OF COMPLEXITY THEORY:

IF YOU WANT TO SOLVE BIG PROBLEM, MAKE SURE THAT EITHER:

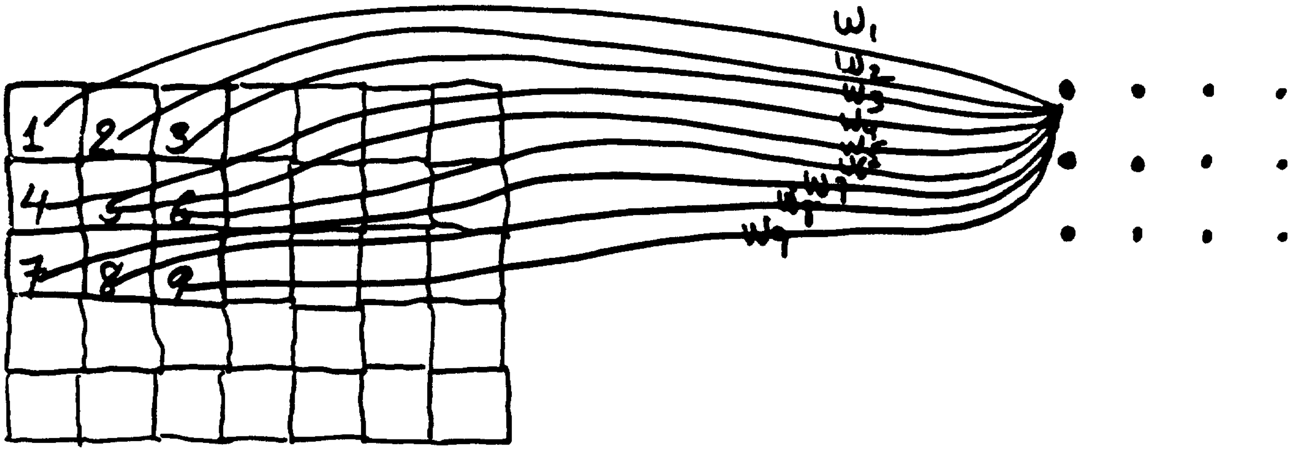
1) YOU ARE NOT IN "WORST CASE"

OR 2) APPROXIMATE SOLUTION IS GOOD ENOUGH

OR 3) NETWORK CAN BE DECOMPOSED/SIMPLIFIED SO THAT "EFFECTIVE SIZE" IS SMALL.

REDUCING THE COMPLEXITY: WEIGHT SHARING

HEP. DETECTOR
CALORIMETER



BY SYMMETRY: $w_1 = w_3 = w_7 = w_9$

$w_2 = w_4 = w_6 = w_8$

w_5

THIS IS FORESEEN IN JETNET

PERHAPS MORE IMPORTANT:

- DECOMPOSE THE PROBLEMS INTO PARTS
- IDENTIFY THE RÔLES OF INTERMEDIATE NEURONS, LAYERS

41

Artificial Neural Networks

Approximation and Learning Theory

Halbert White

with A. R. Gallant, K. Hornik, M. Stinchcombe, and J. Wooldridge

·y /

British Library.

this case, however. Where possible, notation follows that of White and Wooldridge. Other notation and definitions are as in Stinchcombe and White (1989b). We write $\mathscr{B}(\cdot)$ to denote the Borel σ-field generated by the open sets of the argument set, $\text{gr}(\cdot)$ denotes the graph of the indicated correspondence, and $\mathscr{A}(\cdot)$ is the collection of analytic sets of the indicated σ-field.

42

Theorem 4.1. (a) Let $(\Omega, \mathscr{F}, P)$ be a complete probability space and let (Θ, ρ) be a metric space. For $n = 1, 2, \ldots$, let Θ_n be a complete separable Borel subset of Θ and let $\hat{\Theta}_n : \Omega \Rightarrow \Theta$ be a correspondence with $\text{gr}\, \hat{\Theta}_n \in \mathscr{A}(\mathscr{F} \otimes \mathscr{B}(\Theta_n))$ such that for each ω in Ω $\hat{\Theta}_n(\omega) \subset \Theta_n$, and the set $\hat{\Theta}_n(\omega)$ is non-empty and compact. Let $Q_n : \Omega \times \Theta \to \bar{\mathbb{R}}$ be $\mathscr{F} \otimes \mathscr{B}(\Theta)$-measurable, and suppose that $Q_n(\omega, \cdot)$ is lower semicontinuous on Θ_n for each ω in Ω, $n = 1, 2, \ldots$.

Then for each $n = 1, 2, \ldots$ there exists a function $\hat{\theta}_n : \Omega \to \Theta_n$ measurable-$\mathscr{F}/\mathscr{B}(\Theta_n)$ (hence-$\mathscr{F}/\mathscr{B}(\Theta)$) such that $Q_n(\omega, \hat{\theta}_n(\omega)) = \min_{\theta \in \hat{\Theta}_n(\omega)} Q_n(\omega, \theta)$ for all ω in Ω.

(b) In addition, suppose $\{\underline{\Theta}_n\}$ and $\{\bar{\Theta}_n\}$ are increasing sequences of compact subsets of Θ such that $\cup_{n=1}^{\infty} \underline{\Theta}_n$ is dense in Θ and $\underline{\Theta}_n \subseteq \hat{\Theta}_n(\omega) \subseteq \bar{\Theta}_n$ for all ω in Ω, $n = 1, 2, \ldots$. Suppose there exists a function $\bar{Q} : \Theta \to \bar{\mathbb{R}}$ such that for all $\varepsilon > 0$

$$P[\omega : \sup_{\theta \in \bar{\Theta}_n} |Q_n(\omega, \theta) - \bar{Q}(\theta)| > \varepsilon] \to 0 \text{ as } n \to \infty, \tag{4.1}$$

Ω × Θn.
Part (a
measurab
statement
The obje
the limit
converger
permits tl
assumptic
bounding
ficiently c
element o
fast and t
constants
applicatio
We now
to corollar
dimension
entropy (K
metric ent
number of
the same a
exponentia
stochastic p

.4 or A.1(b),
+ ξ)] → 1 as
□
hen optimiza-
on provides a
In the limit,
true mapping
smaller is this

the previous
ined from the
2.1 of White
ed so as to be
not limited to
of White and

chcombe and
erated by the
the indicated
the indicated

ty space and
a complete
ondence with
n, and the set
$\mathscr{F} \otimes \mathscr{B}(\Theta)$-
ous on Θ_n for

n measurable-
$\Theta_n(\omega)$ $Q_n(\omega, \theta)$

and for $\theta_o \in \Theta$

$$\inf_{\theta \in \eta^c(\theta_o, \varepsilon)} \bar{Q}(\theta) - \bar{Q}(\theta_o) > 0, \tag{4.2}$$

where $\eta^c(\theta_o, \varepsilon) \equiv \{\theta \in \Theta : \rho(\theta, \theta_o) \geqslant \varepsilon\}$, and $\bar{Q}$ is continuous at θ_o. Then $\rho(\hat{\theta}_n, \theta_o) \xrightarrow{P} 0$. □

In our application, $(\Omega, \mathscr{F}, P)$ is the space on which the stochastic process $\{Z_t\}$ is defined. The properties of P determine whether $\{Z_t\}$ is an independent or a mixing sequence. The space Θ contains the object of interest (the unknown regression function θ_o), and ρ is a metric that measures distance in this space (weighted mean squared error). Q_n is the criterion function (squared error) optimized to arrive at an estimator $\hat{\theta}_n$. The set $\hat{\Theta}_n(\omega)$ over which optimization is carried out may depend on the data through ω; this permits treatment of cross-validation procedures. In some applications it may be natural to have Q_n defined only on the graph of $\hat{\Theta}_n(\omega)$ or on $\Omega \times \Theta_n$ instead of on all of $\Omega \times \Theta$ as we assume. Lemma 2.1 of Stinchcombe and White (1989b) establishes that defining Q_n on $\Omega \times \Theta$ results in no loss of generality, as there generally exists an appropriate measurable extension to $\Omega \times \Theta$ of Q_n originally defined on gr $\hat{\Theta}_n$ or $\Omega \times \Theta_n$.

Part (a) establishes the existence of a measurable estimator $\hat{\theta}_n$. Without measurability we cannot make probability statements about $\hat{\theta}_n$, such as statements about consistency. Part (b) establishes consistency of $\hat{\theta}_n$ for θ_o. The object θ_o is distinguished by its role as minimizer of $\bar{Q}$ (condition 4.2), the limit to which Q_n converges uniformly (condition 4.1). This uniform convergence can be verified in particular stochastic contexts; our next result permits this for i.i.d. and stationary mixing processes. The other notable assumption of part (b) is the existence of nonstochastic sets $\underline{\Theta}_n$ and $\bar{\Theta}_n$ bounding $\hat{\Theta}_n(\omega)$. The behavior of $\underline{\Theta}_n$ ensures that $\hat{\Theta}_n(\omega)$ becomes sufficiently dense in Θ, so that an element of $\hat{\Theta}_n(\omega)$ can well approximate an element of Θ. The behavior of $\bar{\Theta}_n$ ensures that $\hat{\Theta}_n(\omega)$ does not increase too fast and that Q_n converges to $\bar{Q}$ uniformly in an appropriate sense. The constants $\underline{q}_n$ and $\bar{q}_n$ of the previous section determine $\underline{\Theta}_n$ and $\bar{\Theta}_n$ in our application.

We now give a result permitting verification of condition (4.1), related to corollary 2 of Haussler (1989). Instead of using the concept of V-C dimension (Vapnik and Chervonenkis, 1971) we use the concept of metric

SOME RECENT GOOD BOOKS ON NEURAL NETWORKS

F. James, August, 1991

Amit, Daniel
Modelling Brain Function,
Cambridge University Press, 1989
(Amit is a theoretical physicist, but often adopts the biological approach.)

Anderson and Rosenfeld (eds.),
Neurocomputing: Foundations of Research
MIT Press, 1988
(A collection of important papers, expensive)

Beale and Jackson,
Neural Computing, an introduction,
Adam Hilger, 1990
(A good general introduction, paperback, not expensive.)

Block, H.D.
The Perceptron: A model for brain functioning,
Reviews of Modern Physics 34(1962), 123-135
(reprinted in Anderson and Rosenfeld)

Geszti, T.,
Physical Models of Neural Networks,
World Scientific, 1990

Hertz, Krogh and Palmer,
Introduction to the Theory of Neural Computation
Addison-Wesley, 1991
(probably the best book available for physicists)

Kohonen, T.,
Self-Organization and Associative Memory (3rd edition)
Springer-Verlag, 1989

McClelland and Rumelhart,
Parallel Distributed Processing, Vols. 1, 2 and 3,
MIT Press, 1988
(Vol. 1 gives the foundations and algorithms, vol. 2 is more biological, vol. 3 contains tutorial and software)

[redacted]

to continue that relationship. But the agreement leaves out the Pentium and

In the short-term, Pentium chips are still a small part of the market: Only 1% of all PCs shipped world-wide last year had

44

Researchers Make Advances In Microchip-Neuron Linkup

By JERRY E. BISHOP
Staff Reporter

Researchers said they took a key first step toward creating electronic microchips that use living brain cells.

The researchers said they had learned how to place embryonic brain cells in desired spots on silicon or glass chips and then induce the brain cells to grow along desired paths. The scientists hope to be able within the next six to 12 months to get the brain cells, or neurons, to grow connections to each other that will crudely mimic the circuitry that neurons form in the brain.

"I want to emphasize this is fundamental research," said biophysicist David L. Stenger of the Naval Research Laboratory in Washington, worried that the research might be misinterpreted as a fledgling effort to make an artificial brain.

"We're working with systems where you can investigate how neurons work," he said. As the researchers study how the biological neurons function and connect with each other on the chips, they should learn how to make better networks of artificial neurons that electronics and computer developers are currently trying to engineer, Dr. Stenger explained.

Possible Nerve-Gas Sensor

Nevertheless, their basic research could lead to some useful bioelectronic devices, said Dr. Stenger and his collaborator, chemist Jay Hickman of Science Applications International Corp. in McLean, Virginia, a "high-tech" contract-research company. For example, the pair are collaborating with researchers at Stanford University to develop a sensor composed of nerve cells on a chip that would sound the alarm when it detected a nerve poison, such as nerve gas used in chemical warfare, a bioelectronic variation of the "canary-in-the-coal mine" idea, Dr. Stenger said.

It also may be possible to eventually make "biochips" that drug makers could use to see if new compounds might interfere with, or perhaps enhance, functions like memory or learning.

Drs. Stenger and Hickman, who are funded largely by the Office of Naval Research, are developing the biochips in collaborations with scientists at the National Institutes of Health and at the University of California-Irvine.

Using DETA

The two scientists said they take a chip of silicon or glass and coat it with a single layer of molecules of a chemical that promotes the growth of neurons. They used one called DETA (for diethylenetriamine) in their experiments.

A microscopic "mask" is laid on the chip that shields both the spots where the scientists want the neurons to settle and the channels they want the neurons to follow to make connections. An ultraviolet laser removes the unshielded DETA, and the cleared sections are filled with a layer of molecules that discourage neuron growth.

Embryonic rat neurons are more or less sprinkled on the chip with the hope that those landing on the DETA spots will "take" and grow. "It's kind of like when you were a kid and you put glue on a piece of paper and sprinkled those sparkly things on it and then shook them off," Dr. Stenger explained. The neurons then send out connecting "wires" called dendrites and axons following the paths of DETA.

The neurons can be kept alive and functioning for months for study and experimentation, Dr. Stenger noted. At present, neuroscientists are able to study networks of brain cells outside the intact animal for only a matter of hours.

WALL ST. JOURNAL 1 FEB. '94

d of
ged
me
of
ted
ist.
lts
to
six
30,
ing
ter
ter
AA
aid
ths
for
ear
a
Ex-
in
.35
is
and
to
of
ro-
ue,
1%
on.
the
ose
ree
ult
ree
list
cy-
tal
on
ear
175
ith
ne
a
he
to
to
er
of
ny
ti,
of

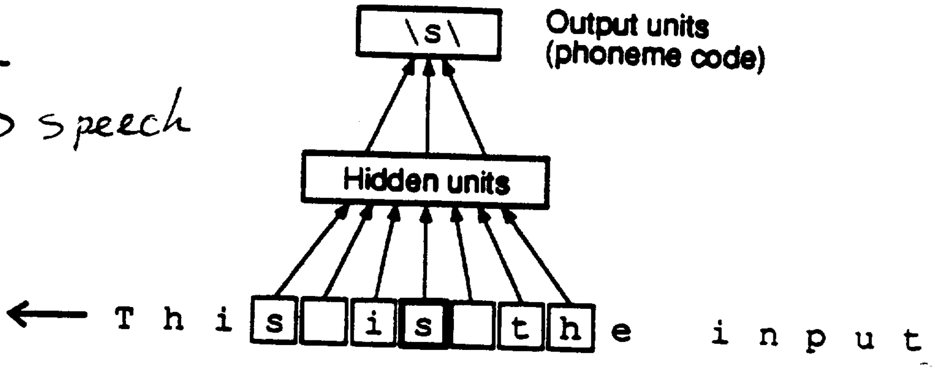
excised p
inspectors
quality doc
ing Albut
counts wer
The re
trouble ove
has prom
concern ab
production
producers
cost versio
keted at hi
ceutical fir
Copley
through a s
are prepar
Copley con
this matter
and will be
resolved."
The FDA
number of
have writte
ical monito
maintenanc
system use
terol. The
in Copley's
In early
a nationwi
0.5% conce
bacterial co
Susan Cr
in Rockville
comment on

SOME FAMOUS APPLICATIONS OF FEED-FORWARD A.N.N.

NETtalk

English text → speech

1987

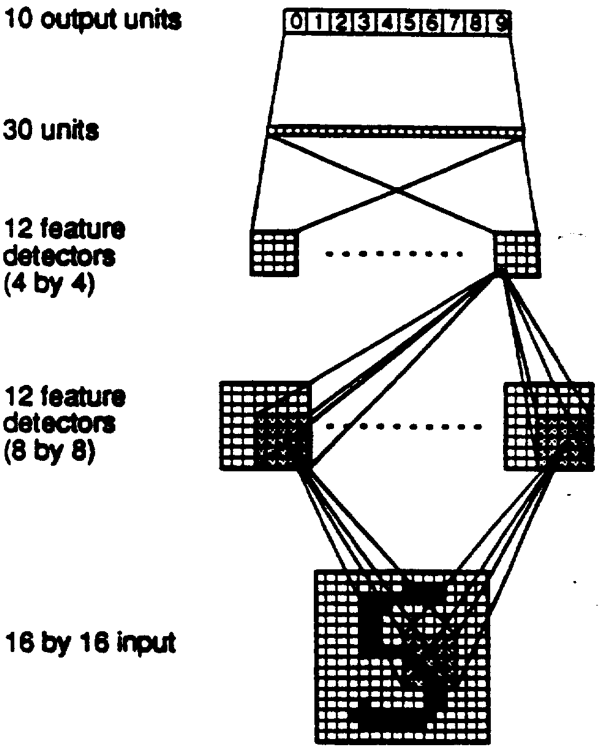


compare with

DEC-talk ⇐ a team of linguists for 10 years!

HANDWRITTEN ZIP-Code READING

1989



7300 digits training

2000 digits test

1% error on training set

test:	5% error no reject	1% error 12% reject

after pruning 3/4 of weights(!)

test: 1% error

9% reject

HUMANS DID NO BETTER!

46

Computer Physics Communications 67 (1991) 183-192
North-Holland

Computer Physics
Communications

Recognition of decays of charged tracks with neural network techniques

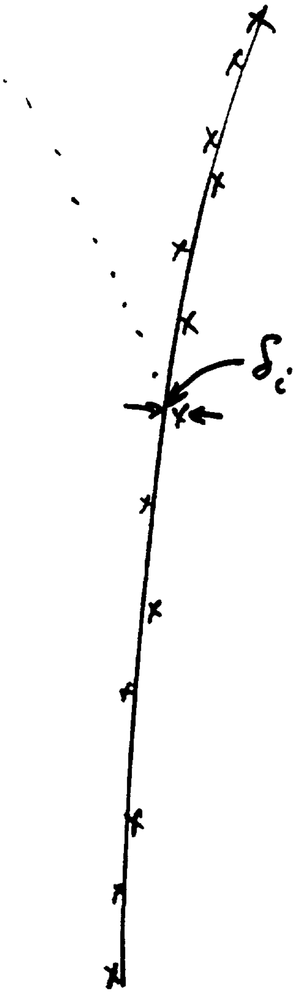
Georg Stimpfl-Abele

Laboratoire de Physique Corpusculaire, Université Blaise Pascal, Clermont-Ferrand, 63177 Aubière, France

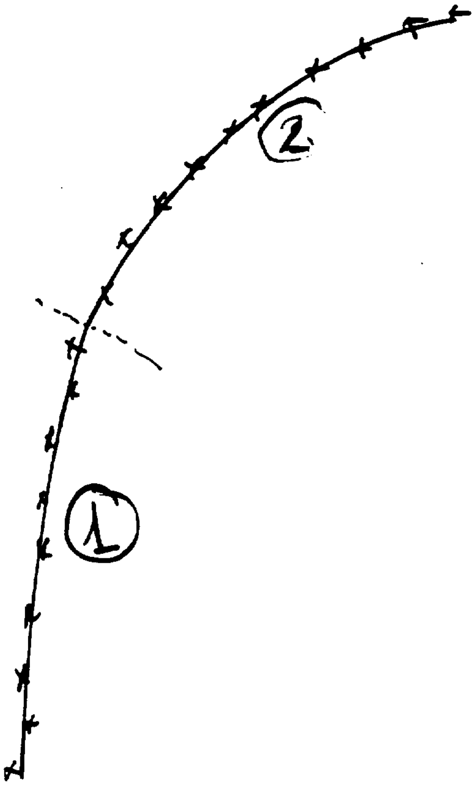
Received 10 May 1991

We developped neural-network learning techniques for the recognition of decays of charged tracks using a feed-forward network with error back-propagation. Two completely different methods are described in detail and their efficiencies for several NN architectures are compared with conventional methods. Excellent results are obtained.

$$\chi^2_{RESIDUALS} = \sum_{pts} \frac{\delta_i^2}{\sigma_i^2}$$



$$\chi^2_{PARAMETERS} = \sum_{PARAM.}^{5} \left(\frac{P_{i\textcircled{1}} - P_{i\textcircled{2}}}{\Delta P_i} \right)^2$$



network for

two hidden we obtained 10-10-1 the residual ap- :onfiguration he same ef- 2-1 layouts. not gain by concentrate one hidden

.bout 40000 them to the .d 150 times) in random ; effects (the of the input 'aster. Since · consuming

tracks and to 5% for DSIM tracks since the number of non-decaying tracks with bad fits is much larger for DSIM data.

The efficiency of the neural networks can directly be compared to the results of the analytical parameter-comparison method.

Table 3a
Kink-finding efficiencies (%) for FSIM data

E_π	3 GeV	5 GeV	10 GeV
χ^2_{res}	73.5	57.8	38.0
χ^2_{par}	79.8	68.0	50.2
N_{res}	75.2	68.5	50.1
N_{par}	79.0	69.1	51.6

Table 3b
Kink-finding efficiencies (%) for DSIM data

E_π	3 GeV	5 GeV	10 GeV
χ^2_{res}	53.1	35.8	14.3
χ^2_{par}	76.0	62.0	40.2
N_{res}	79.3	65.3	48.1
N_{par}	78.3	65.9	47.0

Conclusions: N.N. more efficient and also much faster even on a VAX!

48

Computer Physics Communications 74 (1993) 199–216
North-Holland

Computer Physics
Communications

Identification of b jets using neural networks

Graham Bahan and Roger Barlow

Department of Physics, Manchester Unicersity, Manchester, M13 9PL, UK

Received 5 June 1992

The problem of identifying *b* quark jets produced at LEP using a neural network technique has been studied. We find that networks perform better separation if they are given simple inputs, as opposed to inputs already combined into variables believed to be good for separation. Some first studies of systematic errors resulting from using neural network separation techniques are given.

1. Introduction

Neural networks with feed-forward topology are widely used in pattern recognition. In high energy physics they can be applied to the problem of recognising the nature of the quark producing a hadronic jet, specifically to discrimination between jets from heavy (b) and light (u, d, s, c) quarks, using only the general jet shape as input (as opposed to specific inputs such as high p_T leptons or large impact parameters). Some studies have already been done on this topic [1,2], examining the performance of networks trained and tested on samples of data generated by Monte Carlo programs, for which the nature of the jets is known. This note explores the possibility further, concentrating on the identification of b quark jets produced in the reaction $e^+ e^- \to Z \to q\bar{q}$ at a centre of mass energy of 91 GeV.

In such an application the most important question is what the best input variables are to use with the network; in particular, whether to use constructed shape variables known from experience to be useful for b identification, to give the network the best possible chance of success, or to use unprocessed ("raw") event information, trusting to the network training algorithm to find the best way of combining them. Gottschalk and Nolty [1] suggest that entering the shape of the event in a largely unprocessed format (they used a hypothetical calorimeter) is preferable to using more sophisticated variables, but this study was done at a centre of mass energy of 14 GeV, where the distinction is much clearer than at 91 GeV, and it is not clear that the situations are comparable.

In the next section we discuss how the separating power of a network can be quantitatively measured, particularly when using it to perform a statistical separation. This gives an objective and unambiguous definition of the extent to which one network performance is "better" than another. Then in section 3 we compare the performance of networks using 3 different philosophies for input variables: highly preprocessed variables (following the suggestions of ref. [2]), secondary jet clusters, and raw momenta. Section 4 studies the size and topology of network needed to bring out the results. In section 5 we discuss systematic effects that may result from the use of such a network, by using samples from different Monte Carlo models.

Correspondence to: R. Barlow, Department of Physics, Manchester University, Manchester, M13 9PL, UK.

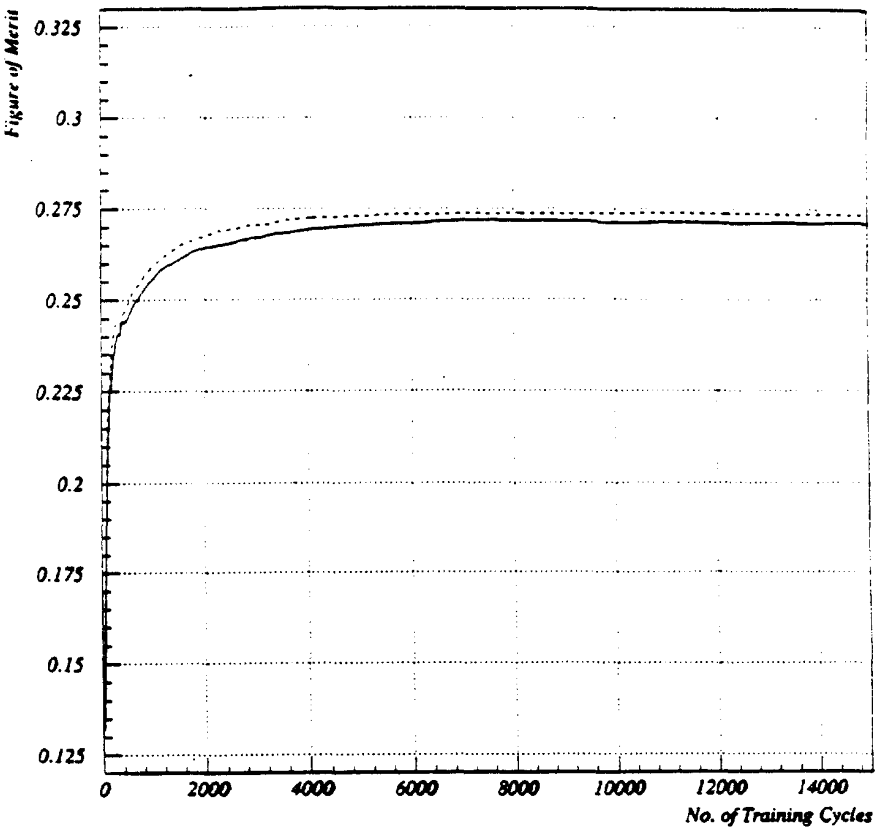


Fig. 5. The learning curve for the secondary clustering inputs.

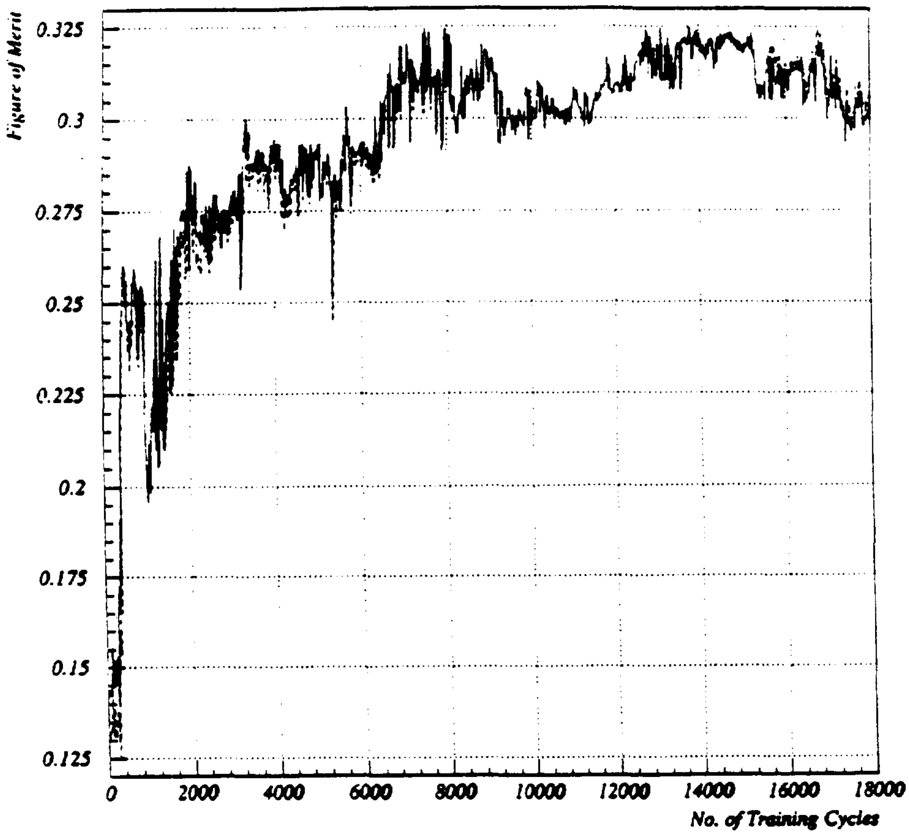


Fig. 7. The learning curve for raw inputs.

50

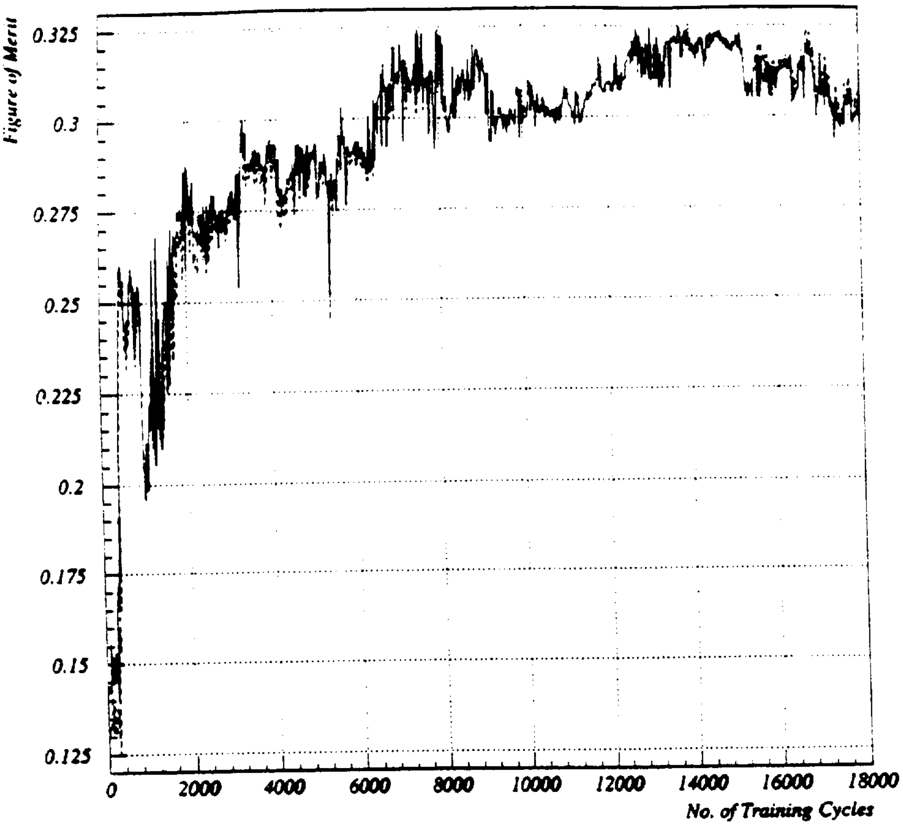
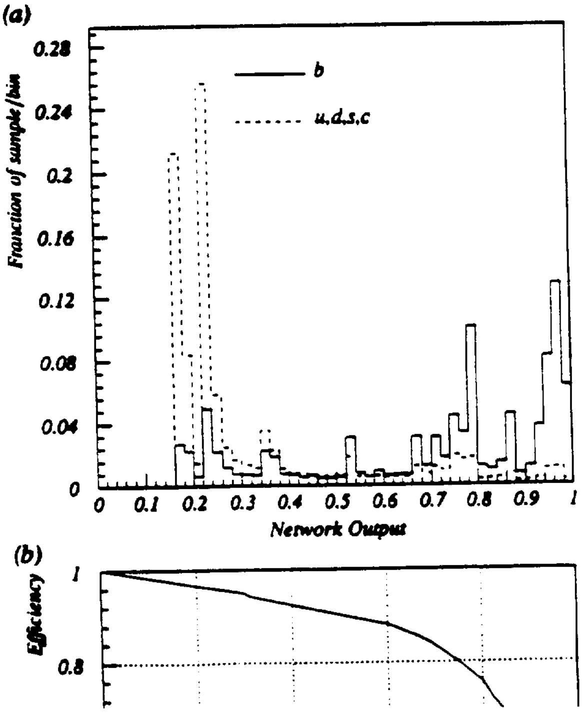


Fig. 7. The learning curve for raw inputs.



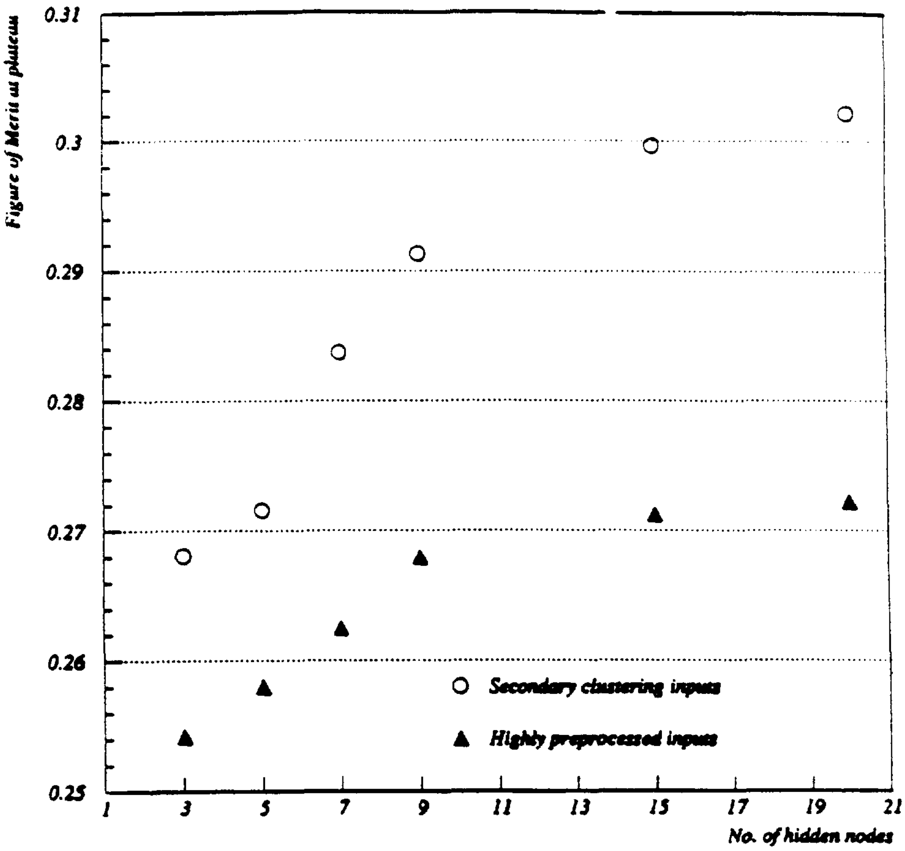


Fig. 9. Effect of the number of nodes in a single hidden layer on the performance of networks using different inputs.

network is much longer, and there are still features of the output which are unpleasant and not well understood.

We recommend the use of the figure of merit, F, as an unambiguous parameter describing the discriminating power of the network, enabling meaningful comparisons to be made between different networks, or different versions of the same network.

Acknowledgements

We are indebted to Leif Lönnblad for providing us with the JETNET program, and to Terry Wyatt for useful discussions.

Appendix. Evaluating F

To evaluate F for the results of a particular network one does not have access to the parent functions $h(x)$ and $l(x)$ but to the two Monte Carlo event samples. These must be histogrammed in bins $i = 1, 2, 3 \ldots N_B$ and F estimated as

$$\hat{F} = \alpha\bar{\alpha} \sum \frac{(h_i - l_i)^2}{\alpha h_i + \bar{\alpha} l_i}.$$

52

Computer Physics Communications 78 (1993) 1–16
North-Holland

Computer Physics
Communications

Higgs search and neural-net analysis

Georg Stimpfl-Abele [a] and Pablo Yepes [b,1]

[a] *Laboratoire de Physique Corpusculaire, Université Blaise Pascal, Clermont-Ferrand, F-63177 Aubiere, France*
[b] *Rice University, Houston TX 77251-1892, USA*

Received 8 June 1993; in revised form 22 August 1993

Two aspects of neural-net analysis are addressed: the application of neural nets to physics analysis and the analysis of neural nets. Feed-forward nets with error back-propagation are applied to the search for the standard Higgs boson at LEP 200. New methods to select the most efficient variables in such a classification task and to analyse the nets are presented. The sensitivity of the nets for systematic effects is studied extensively. The efficiencies of the neural nets are found to be significantly better than those of standard methods.

1. Introduction

The search for new elementary particles is among the most important tasks in high energy physics. In general, events containing new particles are produced along with a much larger number of conventional events. Hence an analysis looking for new phenomena needs a filtering process intended to separate signal and background events.

In this study a traditional filtering method, using standard one-dimensional cuts, is compared with a neural net (NN) approach in the search for the Higgs boson at LEP 200. The following two mass hypotheses are chosen: 70 and 90 GeV/c^2. The lower mass represents the easier case because the signal is higher and the backgrounds can be better discriminated. The higher mass is rather challenging since it is just below the Z mass.

Standard feed-forward nets with one hidden layer and error back-propagation are used. Emphasis is given to the selection of the best input-variables by analysing their utility inside the net. Systematic effects are studied in detail.

The physics case is discussed in section 2, followed by a description of the generation and the preselection of the input data. Section 4 is dedicated to the standard analysis based on one-dimensional cuts. Section 5 contains the technical details of the net generation, like architecture and learning procedure, and a description of the methods developed to analyse neural nets and to select the best input-variables. The performance of the NNs in the Higgs search is demonstrated in section 6. Systematic effects are studied extensively in section 7 in order to test the reliability of the methods. Finally, conclusions are given.

2. Higgs production and backgrounds at LEP 200

The Standard Model of particle physics [1] is the commonly accepted theory to explain the interactions among elementary particles. This model predicts the existence of the Higgs boson, H, responsible for the so-called Symmetry Breaking mechanism [2]. During recent years the Large Electron Positron collider (LEP) at CERN, operating with a center-of-mass energy (E_{cm}) around 91 GeV (LEP 100), has per-

Correspondence to: G. Stimpfl-Abele, PPE division, CERN, CH-1211 Geneva 23, Switzerland; E-mail address: stimpfl@cernvm.cern.ch

[1] E-mail address: yepes@physics.rice.edu

as function of the number of input
ses.

H$_{70}$	H$_{90}$
19.1	8.7
15.8	8.7
15.5	8.7
14.2	8.4
13.2	8.4
13.2	8.2
12.6	8.0
10.9	8.0

arged tracks ($N_{\rm trk}$).

event is fitted with the WW

ıass of all secondary tracks

ıngles between the jets if the
to three jets (S_θ).
of the most energetic electron

f secondaries in the *Higgs* jets
fitted with the HZ hypothesis

re the inputs for N$_{10}$. The in-
r nets N$_n$ are the variables 1 to

ng between the number of in-
the performance two nets from
for further study. The net with
for high performance and the
nputs (N$_7$) but still good per-

with section 4 shows that all
the standard analysis are con-
ıbove. There are two new vari-
$P^{\rm e}_{\rm max}$ (9). Adding them to the
does not improve the statisti-
To allow for a direct compari-

Table 7
Signal and background events left for standard and NN analyses at $m_{\rm H}$ = 70 GeV/c^2 and their statistical significance.

Reaction	Cuts	N$_{\rm st}$	N$_7$	N$_{10}$
$e^+e^- \to$ H$_{70}$ Z	60.5	29.5	23.6	40.9
$e^+e^- \to q\bar{q}gg$	22.3	2.4	1.9	3.7
$e^+e^- \to W^+W^-$	25.0	3.2	1.4	4.0
$e^+e^- \to$ Z Z	1.4	0.4	0.2	0.6
Total background	48.7	6.0	3.5	8.3
Statistical significance	8.7	12.0	12.6	14.2
Min. luminosity [pb^{-1}]	330.	174.	157.	124.

Table 8
Signal and background events left for standard and NN analyses at $m_{\rm H}$ = 90 GeV/c^2 and their statistical significance.

Reaction	Cuts	N$_{\rm st}$	N$_7$	N$_{10}$
$e^+e^- \to$ H$_{90}$ Z	38.0	30.5	26.4	22.0
$e^+e^- \to q\bar{q}gg$	11.9	5.0	3.0	1.1
$e^+e^- \to W^+W^-$	11.8	3.9	2.6	1.6
$e^+e^- \to$ Z Z	11.4	6.6	5.3	4.2
Total background	35.1	15.5	10.9	6.9
Statistical significance	6.4	7.7	8.0	8.4
Min. luminosity [pb^{-1}]	610.	422.	391.	354.

the three nets as function of the cut on the output for both masses. At least two background events above the cut are demanded to avoid big statistical fluctuations. The threshold at 0 is due to the preselection. The statistical significances raise almost linearly with the cut, reaching their maximum around 0.9 .

The number of accepted signal and background events of the three nets and the statistical significance are shown in tables 7 and 8 for the analysis at 70 and 90 GeV/c^2, respectively. The cut on the NN output is chosen such that the significance is maximal. The results of the standard cuts (table 3) are included to ease the

[redacted]

54

Finding Gluon Jets with a Neural Trigger

Leif Lönnblad,[a] Carsten Peterson,[b] and Thorsteinn Rögnvaldsson[c]

Department of Theoretical Physics, University of Lund, Sölvegatan 14A, S-22362 Lund, Sweden

(Received 6 April 1990)

Using a neural-network classifier we are able to separate gluon from quark jets originating from Monte Carlo-generated e^+e^- events with 85%-90% accuracy.

PACS numbers: 13.87.Fh, 12.38.Qk, 13.65.+i

In this Letter, we demonstrate how to separate gluon and quark jets using a neural-network identifier. Being able to distinguish the origin of a jet of hadrons is important from many perspectives. It can shed experimental light on the confinement mechanism in terms of detailed studies on the so-called string effect[1] and related issues. Also, a fairly precise identification of the gluon jet is required for establishing the existence of the three-gluon coupling in e^+e^- annihilation.[2] To date the gluon-jet identification has been done by making various cuts on the kinematic variables ranging from just identifying the jet with smallest energy as the gluon jet[1] to more elaborate schemes.[3,4] Such procedures are often based on the entire event rather than just a single isolated jet. It would be desirable to focus on the latter alternative given that in many situations "global" quantities like total jet energies are less well known. One such example is jets produced in high-p_T hadron-hadron collisions.

A straightforward method for identifying the jets would be to find the functional mapping between the observed hadronic kinematical information and the feature (quark or gluon). This reduces the problem from an expert's exercise to a "black box" fitting procedure. This is exactly what the neural-network approach aims at. It has the advantage over other fitting schemes in that it is very general, inherently parallel, and easy to implement in custom-made hardware with its simple processor structure. The latter feature is very important for real-time triggering.

We confine our studies to Monte Carlo-generated e^+e^- events using the Lund Monte Carlo model. To some extent this induces a "chicken-and-egg" effect to our studies; some of the physics one wants to study is already there. This effect can be minimized by limiting ourselves to kinematical quantities that are most model independent, e.g., considering the fastest particles only.

Although this paper is limited to the separation of gluon and quark jets, it is clear that the methodology could be used in a variety of different triggering situations.

The neural-network learning algorithm.—The basic ingredients in a neural network are *neurons* n_i and connectivity *weights* ω_{ij}. For feature recognition problems like ours the neurons are often organized in a feed-forward layered architecture (see Fig. 1) with input (x_k), hidden (h_j), and output (y_i) nodes. Each neuron performs a weighted sum of the incoming signals and thresholds this sum with a "sigmoid" function $g(x) = 0.5[1+\tanh(x)]$. For the hidden and output neurons one has

$$h_j = g(a_j/T)\,, \qquad (1)$$

$$y_i = g(a_i/T)\,, \qquad (2)$$

where the "temperature" T sets the slope of g and the weighted input sums a_j and a_i are given by $\sum_k \omega_{jk} x_k$ and $\sum_j \omega_{ij} h_j$, respectively.

The hidden nodes have the task of correlating and building up an "internal representation" of the patterns to be learned. Training the network corresponds to changing the weights ω_{ij} such that a given input parameter $\mathbf{x}^{(p)}$ gives rise to an output (feature) value $\mathbf{y}^{(p)}$ that equals the desired output or target value $\mathbf{t}^{(p)}$. A frequently used procedure for accomplishing this is the *back-propagation* learning rule[5] where the error function

$$E = \tfrac{1}{2} \sum_p \sum_i (y_i^{(p)} - t_i^{(p)})^2 \qquad (3)$$

is minimized. Changing ω_{ij} by gradient descent corresponds to[5]

$$\Delta\omega_{ij} = -\eta \delta_i h_j + \alpha \Delta\omega_{ij}^{\text{old}} \qquad (4)$$

for the hidden to output layers, where δ_i is given by

$$\delta_i = (y_i - t_i) g'(a_i)\,. \qquad (5)$$

Correspondingly, for the input to hidden layers one has

$$\Delta\omega_{jk} = -\eta \sum_i \omega_{ij} \delta_i g'(a_j) x_k + \alpha \Delta\omega_{jk}^{\text{old}}\,. \qquad (6)$$

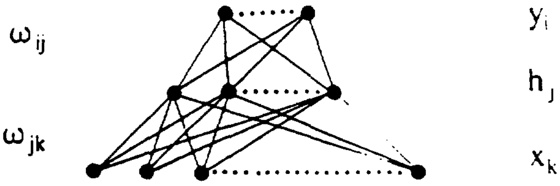


FIG. 1. A feed-forward neural network with one layer of hidden units.

Nuclear Physics B349 (1991) 675-702
North-Holland

55

USING NEURAL NETWORKS TO IDENTIFY JETS

Leif LÖNNBLAD*, Carsten PETERSON** and Thorsteinn RÖGNVALDSSON***

Department of Theoretical Physics, University of Lund, Sölvegatan 14A, S-22362 Lund, Sweden

Received 29 June 1990

A neural network method for identifying the ancestor of a hadron jet is presented. The idea is to find an efficient mapping between certain observed hadronic kinematical variables and the quark-gluon identity. This is done with a neuronic expansion in terms of a network of sigmoidal functions using a gradient descent procedure, where the errors are back-propagated through the network. With this method we are able to separate gluon from quark jets originating from Monte Carlo generated e^+e^- events with ~85% approach. The result is independent of the MC model used. This approach for isolating the gluon jet is then used to study the so-called string effect.

In addition, heavy quarks (b and c) in e^+e^- reactions can be identified on the 50% level by just observing the hadrons. In particular we are able to separate b-quarks with an efficiency and purity, which is comparable with what is expected from vertex detectors. We also speculate on how the neural network method can be used to disentangle different hadronization schemes by compressing the dimensionality of the state space of hadrons.

1. Introduction

During the last couple of years there has been an upsurge in interest for brain-style computing in terms of artificial neural networks (NN). The origin of this enthusiasm is the power this new computational paradigm has shown for a wide variety of real-world feature recognition applications. Not only is the performance of the NN promising but the entire approach is very appealing with its adaptiveness and robustness. Another attractive feature is the inherent parallelism in neural networks and the feasibility of making custom made hardware with fast execution times and thereby facilitating real-time performance.

High-energy physics contain many feature recognition problems, ranging from low-level trigger conditions in experimental setups, to extraction of theoretically relevant quantities in collected data. Needless to say, the demand for efficient feature extraction procedures will become more acute with increasing luminosity and energy. In a previous paper [1] preliminary results for gluon-quark separation

* thepll@seldc52 (bitnet) leif@thep.lu.se (internet)
** thepcap@seldc52 (bitnet) carsten@thep.lu.se (internet)
*** thepdr@seldc52 (bitnet) denni@thep.lu.se (internet)

TABLE 7
Results for the $r = \langle n_2 \rangle / \langle n_1 \rangle$ measured on a sample of 10000 MC events (ARIADNE) for different methods of identifying the gluon jet: using the true gluon jet of the MC, the NN approach and the "smallest jet" approach

"traditional method

	MC truth	Neural network	Smallest jet
Success rate	100%	74%	67%
$r = \langle n_2 \rangle / \langle n_1 \rangle$	2.16 ± 0.03	2.00 ± 0.02	1.60 ± 0.02
$r\ (m_{\mathrm{had}} \geqslant m_K)$	3.0 ± 0.1	2.7 ± 0.1	1.9 ± 0.1

56

Computer Physics Communications 78 (1993) 17-22
North-Holland

Computer Physics
Communications

A comparison between a neural network and the likelihood method to evaluate the performance of a transition radiation detector

R. Bellotti [a], M. Castellano [b], C. De Marzo [a], N. Giglietto [b],
G. Pasquariello [c] and P. Spinelli [a]

[a] *Dipartimento di Fisica, Università di Bari, and INFN, Sez. di Bari, Via Amendola 173, 70126 Bari, Italy*
[b] *INFN, Sez. di Bari, Via Amendola 173, 70126 Bari, Italy*
[c] *Istituto Elaborazione Segnali e Immagini - C.N.R., Via Amendola 166/5, 70126 Bari, Italy*

Received 16 June 1993; in revised form 16 August 1993

A classification system able to evaluate the performances of a transition radiation detector prototype for electrons/hadrons discrimination is presented. It is based both on a layered feed-forward neural network trained using back-propagation and a likelihood ratio technique. The information fed into the classification system consists of the number of hits detected by each multiwires proportional chamber of the detector. The best results are obtained by the neural network approach that successfully identifies 4.0 GeV/c electrons with an hadron contamination of about 4×10^{-3} at 98% acceptance efficiency.

1. Introduction

Several detectors have been proposed, so far, for particle identification in high energy physics. Transition radiation detectors (TRDs) can be used to discriminate particles with a different γ Lorentz factor [1]: they have been employed for high energy particle identification in experiments at CERN [2,3] and Fermilab [4,5] and in the future also in LHC (see e.g. ref. [6]) and SSC experiments (see e.g. ref. [7]). Moreover, TRDs can be used to distinguish positrons from protons up to 1 TeV energy in cosmic ray space experiments [8-10].

Pattern recognition methods, involving powerful statistical and neurocomputing techniques, are taken into account in high energy physics to study the classification power of detectors. For this purpose, the output patterns of the detectors, produced by known particle-classes, are stored in tagged data files. They are examined by a classification system in order to carry out explicit relationship among data in terms of mapping a pattern from pattern space into class-membership space. A measure of the separability of the classes in a suitable feature space provides the detector classification power.

In this paper is presented a pattern recognition system based on a back-propagation neural network and likelihood ratio algorithm to evaluate the performances of a TRD developed [10,11] for electrons/hadrons discrimination in cosmic ray experiments. In the next section the classification system is presented and the neural architecture and the likelihood ratio test are described; in the third section the application to experimental data and test results are discussed.

2. The pattern classification system

The TRD prototype [10] consists of 10 modules, each one made of a carbon fiber radiator followed by a $Xe-CH_4$ filled proportional cham-

Correspondence to: M. Castellano, INFN, Sez. di Bari, Via Amendola 173, 70126 Bari, Italy.

57

fixed momentum, by means of Cherenkov counters, lead glass scintillators and a standard module electronic logic. A mean tagging inefficiency for the electrons is evaluated of about 10^{-4} [10].

The TRD data, according to the scheme in fig. 1, are taken at the different beam momenta available. Typical data for pions and electrons are shown in fig. 3. The pion events are mainly detected with a small number of hits in the whole detector, while the electron patterns have some counts in each chamber. Using the same data sets, i.e. about 5000 events for each beam momentum, the N_0 and L feature spaces are generated, as shown in fig. 4 for 4 GeV/c momentum only. Better accumulation around 0 and 1 of the feature values is achieved in the N_0 space with respect to the L one. This behaviour is well emphasized in fig. 5 where pion contamination versus electron acceptance is computed according to the procedure described in the second section. As a result, the NN technique reaches the minimum pion contamination value around 95% of electron acceptance efficiency. The table 1 summerizes pion contamination against acceptance at three different momentum data set for both methods. In the NN method a very low contamination is already achieved at full acceptance. This performance is never shown by other methods [12,19,20] and it can be useful to employ TRDs in space cosmic ray experiments [9,10] where, searching for rare events with the constraint of short duration exposures, an acceptance as large as possible is required.

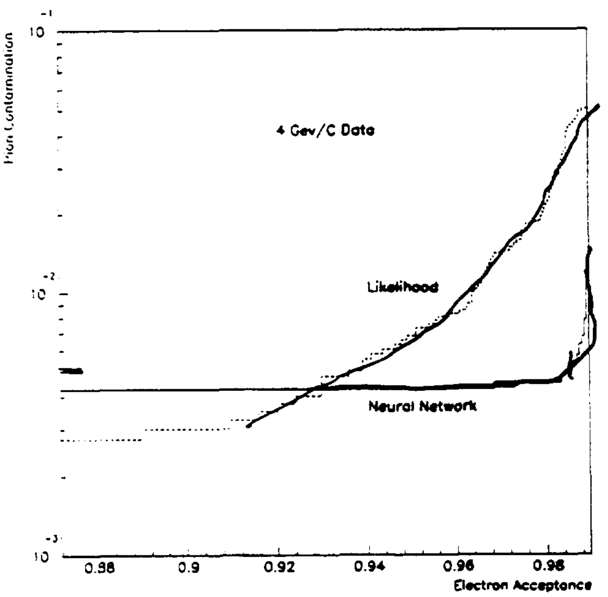


Fig. 5. Pion contamination versus electron acceptance for 4 GeV/c data: the TRD performance evaluated by the likelihood ratio technique (dotted line) and the neural network (solid line).

Table 1
Pion contamination ($\times 10^{-3}$) evaluated for different electron acceptance by the likelihood ratio test and neural network techniques.

Acceptance	1 GeV		2 GeV		4 GeV	
	L	NN	L	NN	L	NN
90%	25.3	15.1	14.8	6.0	3.0	4.2
95%	83.4	26.7	21.9	6.0	6.8	4.2
97%	134.9	26.7	46.6	8.1	14.1	4.5
98%	236.3	26.7	97.9	90.0	21.4	4.5

4. Conclusions

A comparison between statistical and neural approaches to evaluate the TRD performances for the electron/pion discrimination problem has been presented. The ability of the net to explore many hypotheses simultaneously with respect to the likelihood ratio statistical technique has played a fundamental role in obtaining better results. Finally, from the practical view point, the parallel distributed processing model of the net could be hardware implemented using neuron-like components to speed up the particle classification task.

References

[1] B. Dolgoshein, Nucl. Instrum. Methods A 326 (1993) 434.
[2] J. Cobb et al., Nucl. Instrum. Methods 140 (1977) 413.
[3] A. Vacchi, Nucl. Instrum. Methods A 252 (1986) 235.
[4] A. Denisov et al., preprint Fermilab-Conf-84/134-E (1984).
[5] D. Errede et al., preprint Fermilab-Conf-89/170-E (1989).
[6] Proc. ECFA-CERN Workshop on the Large Hadron Collider in the LEP-Tunnel, ECFA 90-133, Vol. I (1990).

58

Computer Physics Communications 78 (1993) 23–28
North-Holland

Computer Physics
Communications

Selection of optimal subsets of tracks with a feed-back neural network

Rudolf Frühwirth

Institut für Hochenergiephysik der Österreichischen Akademie der Wissenschaften, Vienna, Austria

Received 6 July 1993

A feed-back neural network is described which solves the problem of finding an optimal set of compatible nodes in a compatibility graph. The properties of the network are explored with random graphs. The performance is illustrated with simulated data of the DELPHI detector.

1. Introduction

An important constraint on any track reconstruction algorithm is given by the fact that any two tracks in the final selection have to be *compatible*, i.e. must not share data. The selection algorithm therefore has to solve a *mization problem* by selecting a of compatible tracks. It has

Table 2
Classification of simulated tracks.

Algorithm	Data set I neural net	 standard	Data set II neural net	 standard
Simulated tracks	5000	5000	10000	10000
Reconstructable tracks	4331	4331	8279	8279
Tracks in class 1	3867	3712	5518	4874
Tracks in class 2	240	251	903	850
Tracks in class 3	94	149	1176	1625
Processing time [CPU s]	483	518	2067	2538

que. with the largest sum quality indicators (QIs) will be called the best *maximal set*.

In our specific application we have found it useful to define the track QI as a combination of track angle and chi-square probability. The procedure is described in section 4.

The problem of finding maximal compatible sets in a compatibility graph has been analyzed in considerable detail in ref. [1]. By modifi- the Hopfield net

]. The details of the simulation performance of the network algorithm are presented in section 4.

2. Architecture of the network

The relation of compatibility in a set of tracks can be mapped on a compatibility graph in which

Correspondence to: R. Frühwirth, Nikolsdorfer Gasse 18, A-1050 Wien, Austria (permanent address).

59

Test of agreement between two multidimensional empirical distributions

LLuís Garrido[1,2] , **Vicens Gaitan**[2], **Miquel Serra-Ricart**[3]
(1) Departament d'Estructura i Constituens de la Materia
Universitat de Barcelona
Diagonal 647, E-08028 Barcelona, Spain
(2) Institut de Física d'Altes Energies
Universitat Autònoma de Barcelona
E-08193 Bellaterra (Barcelona), Spain

(3) Instituto de Astrofísica de Canarias
E-38200 La Laguna (Tenerife), Spain

January 20, 1994

Abstract

We present a method to test the agreement between two multidimensional empirical distributions which is not restricted to work with projections in fewer dimensions due to the lack of data, and with the relevant fact that it is free of binning. The method, which can be successfully implemented on layered neural nets, gives a lower bound value on any estimator that measures the inconsistency between the two distributions.

(Submitted to Computer Physics Communications)

1 Introduction

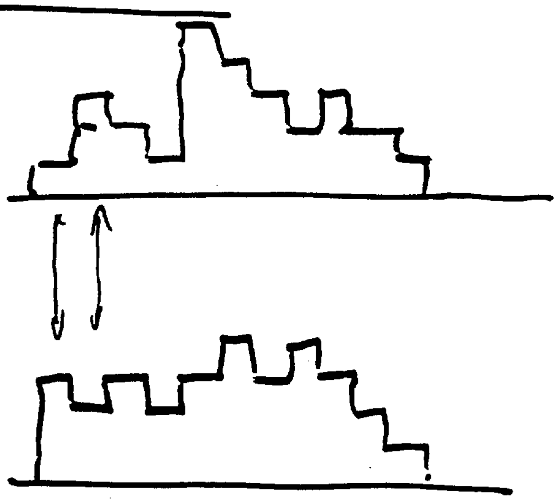
An everyday task in all areas of science is the comparison of theoretical models with experimental data. In some cases the theoretical models cannot be checked directly because their explicit analytical form does not exist (e.g.: the models are the result of a large number of calculations with complex algorithms) and/or there are additional effects not included in the model (e.g.: detector effects during the acquisition of the experimental data). A commonly used strategy to overcome this problem is to generate Monte Carlo events according to the theoretical model and to fold them with the additional effects. The result is a simulated data sample (MC) that can be checked against the experimental data.

TEST OF STATISTICAL AGREEMENT BETWEEN 2 DATA SETS.

(WERE THEY DRAWN FROM THE SAME PROBABILITY DENSITY DISTRIBUTION?)

IF DATA IS ONE-DIMENSIONAL

- CHISQUARE REQUIRES BINNING



- KOLMOGOROV TEST REQUIRES ORDERING

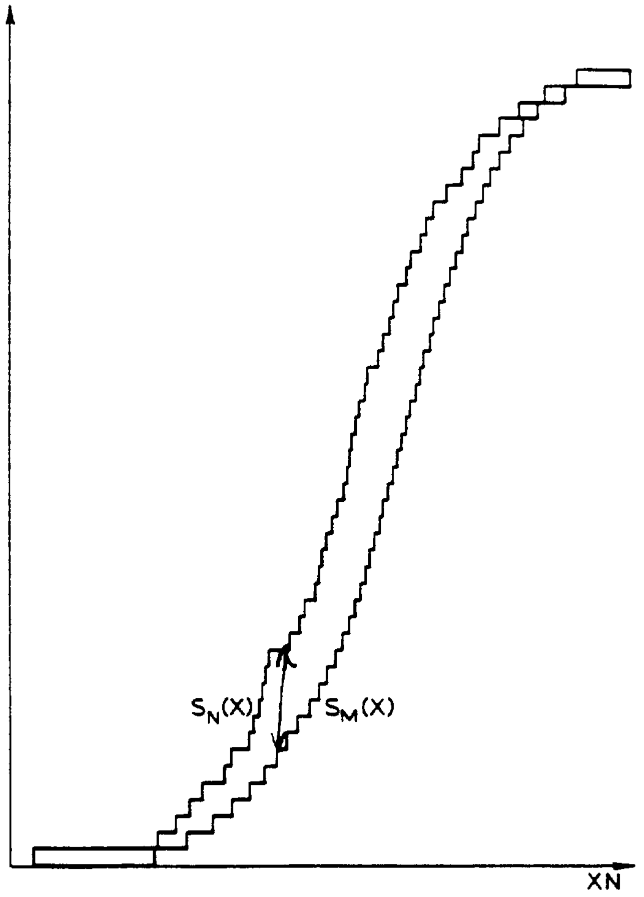


Fig. 11.3 Example of two cumulative distributions, $S_N(X)$ and $S_M(X)$, of the type (11.9).

WHEN YOU MEASURE MORE THAN ONE VARIABLE FOR EACH DATA POINT, THE DATA BECOME MULTI-DIMENIONAL AND THE TEST FOR AGREEMENT

SHOULD BECOME MORE POWERFUL.

BUT

WE DON'T KNOW HOW TO USE THE EXTRA INFORMATION

- CHISQUARE : BINS BECOME TOO BIG. LOSE RESOLUTION

- KOLMOGOROV : CANNOT ORDER DATA IN > 1 DIMENSION

ANY PROJECTION THROWS AWAY INFORMATION

UNLESS

THE PROJECTION IS BASED UPON THE DIFFERENCE BETWEEN THE TWO DATA SETS!

THE GAITAN-GARRIDO PROJECTION

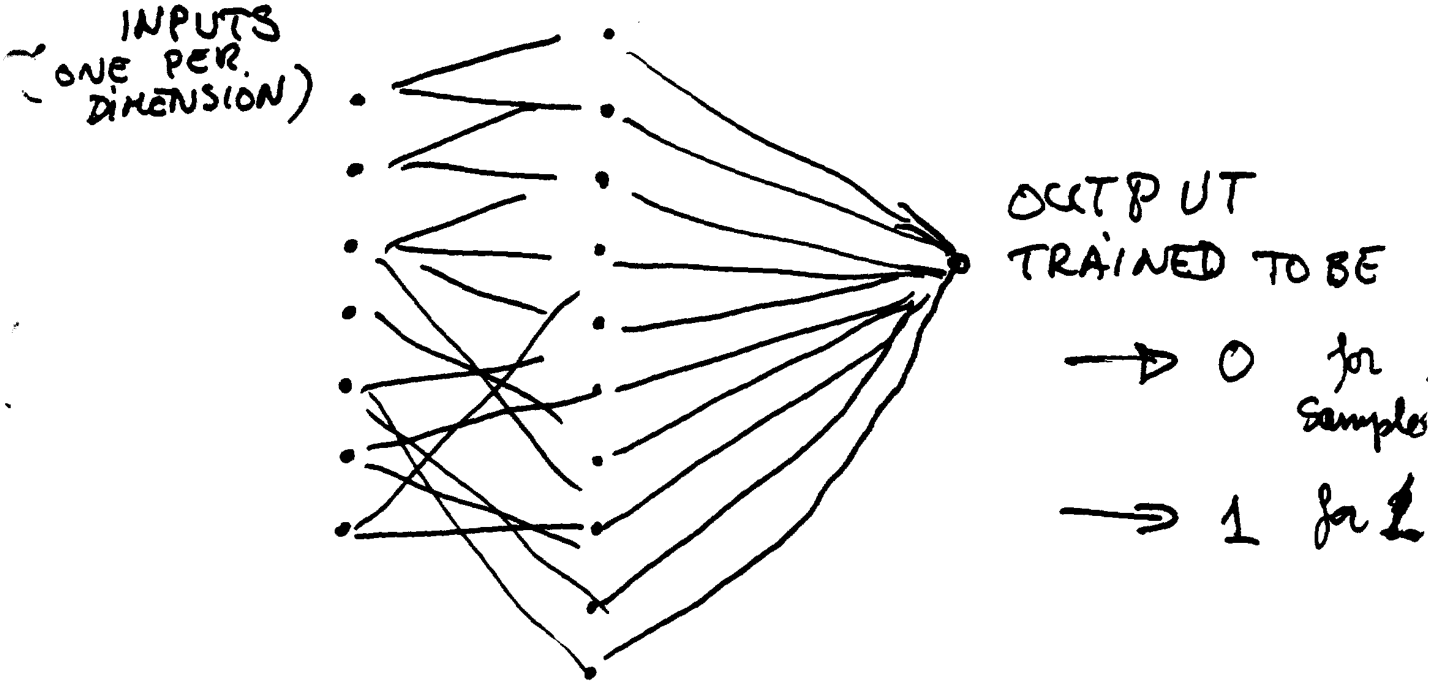
PROJECT ONTO THE VARIABLE

$$K = \frac{P_1(\underline{x})}{P_1(\underline{x}) + P_2(\underline{x})}$$

P_1, P_2 = underlying probability densities from which samples 1 and 2 were taken (UNKNOWN!)

HOW TO ESTIMATE K ?

NEURAL NETWORK, OF COURSE



THIS NET MAPS EVERY DATA ITEM TO A VALUE OF K