Introduction

- soft-computing techniques : neural networks
- applications

Maurizio Valle

Soft computing techniques: neural networks

Lippmann 1987

The rationale underlying Neural Networks is to achieve good **overall** performance via **dense** interconnection of **simple computational elements**. To this respect, the NN structure loosely resembles that of biological nervous systems. Computational elements or nodes (i.e. neurons) are typically **non-linear**.

The simplest neuron sums N weighted inputs (the inputs are weighted by synapses) and passes the result forward (to other neurons) through a non-linearity: each neuron is characterised by an internal threshold and/or by the type of the non-linearity (usually the tanh() or the sigmoid() function).

Soft computing techniques: neural networks

To solve complex cognitive tasks, Neural Networks are much more "**trainable** than explicitly programmable": NN models **explore many competing hypotheses simultaneously**. **The values of the weights represent the net's knowledge**.

They can be accomplished, for a given task, through a procedure, usually called "**training**" or "**learning**".

In general, NN models are specified by the **net topology, neuron characteristics and learning rules**.

Soft computing techniques: neural networks

Cauwenberghs 1999

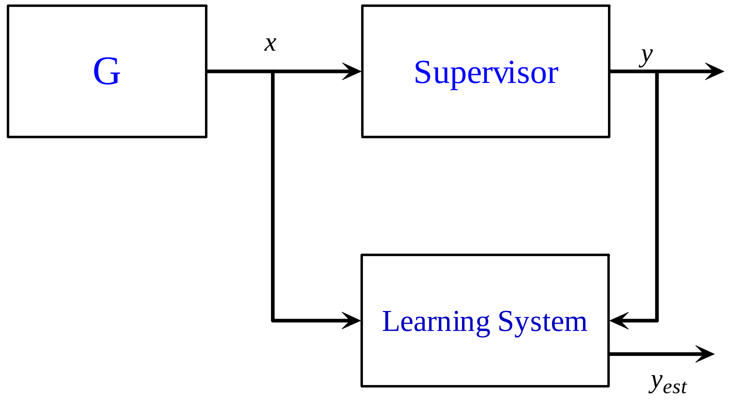
A general framework for **learning**. Given a system with a set of adjustable parameters (i.e. the array **w**, the "adaptive elements") adjust each parameter Wj,i (i.e. generic element of the array **w**) to "optimise/maximise" some previously defined performance index or to "minimise" a previously defined error index $\varepsilon(\mathbf{w})$.

The basic difference between the general form of **adaptation** and **learning** lies in the way **the system uses past experience** in trying to respond effectively to previously unseen, although similar, input stimuli.

The learning task aims at **generalising** beyond the specifics of the presented input samples, and at minimising the expected value of $\varepsilon(\mathbf{w})$ from the underlying statistics of the training samples: $\mathbf{w} = \text{argmin } E(\varepsilon(\mathbf{w}))$.

Soft computing techniques: neural networks

Vapnik 1999



Soft computing techniques: neural networks

- G: generates random vectors $x \in R^n$ drawn from a **fixed** but **unknown** pdf *F(x)*
- Supervisor: returns an output value *y* to every input vector *x*, according to a conditional distribution function ***F(y|x)***, **fixed** but **unknown**
- Learning System: is capable of implementing a set of functions $y_{es} = f(x,\mathbf{w})$

Soft computing techniques: neural networks

The **selection** of the desired function is based on a **training set** of *l* **independent and identically distributed observations** drawn according to $F(x,y)=F(x)F(y|x)$: $(\mathbf{x}_1,\mathbf{y}_1),\ldots,(\mathbf{x}_l,\mathbf{y}_l)$

The **problem of learning** is that of choosing from the set of functions $f(x,\mathbf{w})$ the one that best approximates the supervisor's response i.e. that minimizes E (ε(**w**)) in the situation where the joint probability density function ***F(x,y)*** is **unknown** and the **only available information is contained in the training set**.

Soft computing techniques: neural networks

- Pattern recognition

Let the supervisor's output $y=\{0,1\}$ and also $y_{es} = f(x,\mathbf{w}) = \{0,1\}$.

The problem of learning is to find a function $f(x,\mathbf{w})$ which minimizes:

$$\varepsilon(w) = \int L(y, f(x,w))dF(x,y)$$

where:

$$L(y,f(x,w)) = \begin{cases} 0 \text{ if } y = y_{es} \\ 1 \text{ if } y \neq y_{es} \end{cases}$$

then the problem of learning is to find a function that minimizes the probability of classification error when the probability measure ***F(x,y)*** is **unknown** but the **learning set is known**.

Soft computing techniques: neural networks

- Regression estimation

Let the supervisor function y be a **real value** and let $f(x,\mathbf{w})$ be a set of real functions that contains the *regression function* $f(x,w_0)=\int y\,dF(y\,|\,x)$

The regression function is the one that minimizes:

$$\varepsilon(w)=\int\left(y-f(x,w)\right)^2 dF(x,y)$$

The problem of learning is to find a function that minimizes the ε(w) when the probability measure ***F(x,y)*** is **unknown** but the **learning set is known**.

Soft computing techniques: neural networks

- Density estimation

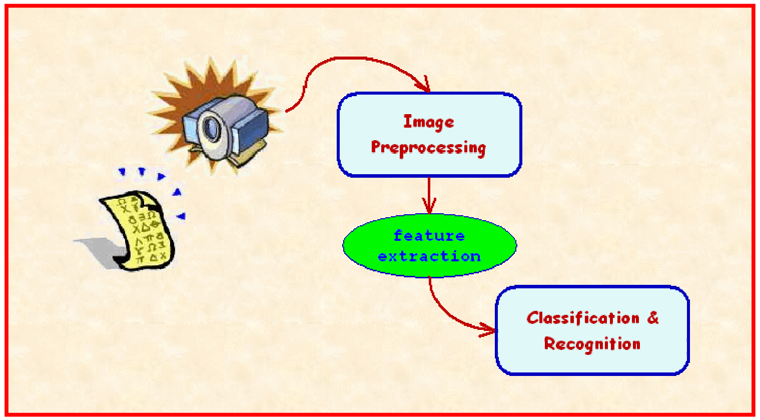
Finally consider the problem of density estimation from the set of density functions p$(x,\mathbf{w})$.

The problem of learning is to find the density function which minimizes:

$$\varepsilon(w)=\int p(x,w)\,dF(x,y)$$

The problem of learning is to find a function that minimizes the ε(w) when the probability measure ***F(x)*** is **unknown** but indipendent and identically distribuited data $x_1, \ldots, x_n$ are given

Applications (1): Optical Character Recognition (OCR)



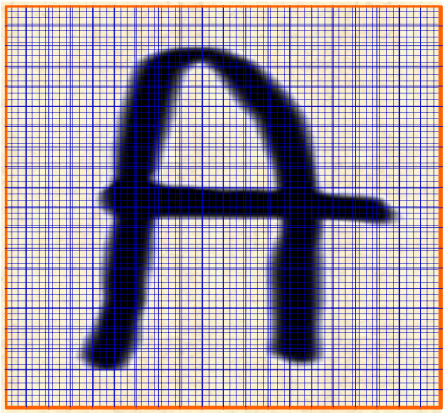
Applications (1): Portable OCR Systems

- Pen scanners **with OCR capabilities are emerging as a new segment of portable equipments**

⇒ They can still be improved in:

- robustness against variations of fonts, contrast and hand speed
- recognition of symbols, punctuation and handwriting
- speed
- accuracy
- power efficiency
- connectivity
- etc.

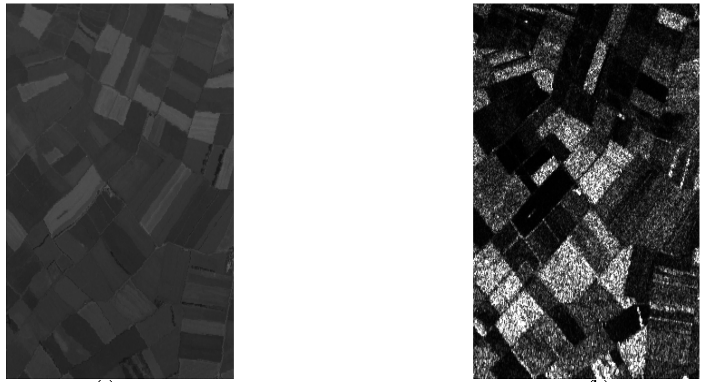
Applications (1): Optical Character Recognition



20 × 20 = 400 pixels

64 × 64 = 4096 pixels

Applications (2): classification of remote sensing images



(a) (b)

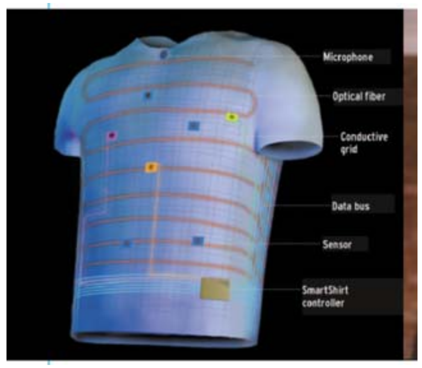
Multisensor image used for the experiments: (a) channel 9 of ATM sensor; (b) channel L-HV (L-band, HV-polarization) of the SAR sensor.

Applications (3): ECG data reconstruction (Murray et al. 2004)

(a) **(b)**

The ECG traces, of (a) normal heartbeats (b) ectopic beats, sampled from training data (solid line) and reconstructed by the trained CRBM (dashed line) after 20 steps of Gibbs sampling from initially-random input.

Applications (4): smart sensors and multisensors



A flexible data bus in Sensatex's SmartShirt prototype carries signals from various sensors plugged into connectors in the shirt to a controller at the waist.An optical fiber woven through the shirt can detect penetration by a bullet.

Red wires supply voltage, green wires carry data, and blue wires are ground for Infineon's demonstrator smart carpet motion-detection module. A capacitive sensor in the module detects when a green wire is touched, which lights the red LED.

IEEE Spectrum Oct. 2003