Introduction to
Multilayer Perceptrons

Marco Gori
University of Siena
An introduction to Multilayered Neural Networks

Outline of the course

• Motivations and biological inspiration
• Multilayer perceptrons: architectural issues
• Learning as function optimization
• Backpropagation
• The applicative perspective
An introduction to Multilayered Neural Networks

A difficult problem for knowledge-based solutions

An “A” perceived by a webcam

How can I provide a satisfactory statement to associate the picture with an “A”?
Emulation of the brain?

... Or simply inspiration?

I just want to point out that the componentry used in The memory may be entirely different from the one that Underlines the basic active organs. John von Neumann, 1958

... Inspiration at the level of neurons
... Hard to go beyond!
An introduction to Multilayered Neural Networks

Artificial neurons

\[ \sigma(\cdot) = \tanh(\cdot) \]

Sigmoidal units

Radial basis units
Supervised and reinforcement learning

Reinforcement info: reward/punishment
Supervised info: specific target
An introduction to Multilayered Neural Networks

Directed Acyclic Graph Architecture

Partial ordering on the nodes

Feedforward architecture

Multilayer architecture
An introduction to Multilayered Neural Networks

Forward Propagation

Let $S$ be any topological sorting of the nodes and let $pa[v]$ be the parents of node $v$

\[
\text{foreach } v \in S \\
\quad x_v = \sigma \left( \sum_{z \in pa[v]} w_{v,z} x_z \right)
\]
Boolean Functions

\[ f(u_1, \ldots, u_m) = \bigvee_h m_h \]

\[ = \bigvee_h \bigwedge_k \chi(u_k(h)) \]
Boolean Functions by MLP

- Every Boolean function can be expressed in the first canonical form
- Every minterm is a linearly-separable function (one on a hypercube’s vertex)
- OR is linearly-separable
- Similar conclusions using the second canonical form.

Every Boolean function can be represented by an MLP with two layers: minterms at the first layer, OR at the second one.
Membership functions

A set function is defined by

$$f_U(u) = 1 \quad \text{for all} \quad u \in U$$

Convex set by MLP
Membership for complex domains

non-connected domains

non-convex domains
An introduction to Multilayered Neural Networks

Membership functions (Lippman ASSP’87)

- Every hidden unit is associated with a hyperplane
- Every convex set is associated with units in the first hidden layer
- Every non-connected or non-convex set can be represented by a proper combination (at the second hidden layer) of units representing convex sets in the first hidden layer

Basic statement: Two hidden layer to approximate any set function
Universal Approximation

Given \( f : \mathbb{R}^m \rightarrow \mathbb{R}^p : u \rightarrow f(u) \)

and \( d = f(u) \quad \epsilon > 0 \)

find \( w \in \Omega \)

\[
\int_{U \times D} \| f(u) - x(w, u) \|^2 \, du \, dd < \epsilon.
\]

One hidden layer (with “enough” hidden units)!
Supervised Learning

Consider the triple \( \{ \mathcal{L}, \mathcal{N}, E(\cdot) \} \)

where

\[
L = \{ (u_\sigma, d_\sigma), \sigma \in \Delta U \}
\]

\[
E(\omega) = \sum_{u \in U} e(u)
\]

Error due to the mismatch between \( x(\omega, u) \) and \( d(u) \)

\[
E_2 = \frac{1}{2} \sum_{u \in U} (x(\omega, u) - d(u))^2
\]
An introduction to Multilayered Neural Networks

Gradient Descent

\[ \frac{d\omega}{dt} = -\mu \nabla \omega E \]

- The optimization may involve a huge number of parameters – even one million (Bourlard 1997)
- The gradient heuristics is the only one which is meaningful in such huge spaces
- The trajectory ends up in local minima of the error function.
- How is the gradient calculated? (very important issue!)
Backpropagation


\[ \frac{\partial E}{\partial w_{ij}} = \sum_{u \in U} \frac{\partial e(u)}{\partial w_{ij}} \]

\[ e_{ij} = \frac{\partial e}{\partial w_{ij}} = \frac{\partial e}{\partial a_i} \cdot \frac{\partial a_i}{w_{ij}} = \delta_i \cdot x_j, \]
An introduction to Multilayered Neural Networks

Backpropagation (con’t)

\[
\begin{align*}
\text{if } \ i \in O \text{ then } \delta_i &= \sigma'(a_i)(x_i - d_i) \\
\text{else } \delta_i &= \sum_{k \in \text{ch}[i]} \frac{\partial x_i}{\partial a_i} \cdot \frac{\partial a_k}{\partial x_i} \cdot \frac{\partial e}{\partial a_k} \\
&= \sigma'(a_i) \sum_{k \in \text{ch}[i]} w_{k,i} \cdot \delta_k
\end{align*}
\]
An introduction to Multilayered Neural Networks

Backpropagation (con’t)

$S_d$ any topologic sorting induced by $< d$

$S_i$ topologic sorting induced by the inverse $< i$

FORWARD $\forall i \in S_d$ $x_i = \sigma \left( \sum_{j \in \text{pa}[i]} w_{i,j} x_j \right)$

BACKWARD $\forall i \in S_i$

\begin{align*}
\text{begin} \\
\text{if } i \in O \text{ then } \delta_i &= \sigma'(a_i)(x_i - d_i) \text{ else} \\
\delta_i &= \sigma'(a_i) \sum_{k \in \text{ch}[i]} w_{k,i} \cdot \delta_k \\
e_{i,j} &= \delta_i \cdot x_j \\
\text{end}
\end{align*}
An introduction to Multilayered Neural Networks

Batch-learning

We calculate the “true-gradient”

$$\frac{\partial E}{\partial w_{ij}} = \sum_{u \in U} \frac{\partial e(u)}{\partial w_{ij}}$$

The weight updating takes place according within
The classic framework of numerical analysis
On-line learning

Weight updating after presenting each example …

\[ w_{ij}(k + 1) = w_{ij}(k) - \mu \frac{\partial e(k)}{\partial w_{ij}} \]

momentum term to filter out abrupt changes …

\[ w_{ij}(k + 1) = w_{ij}(k) - \mu \frac{\partial e(k)}{\partial w_{ij}} + \eta w_{ij}(k) \]
Backprop Heuristics

• Weight initialization
  – Avoid small weights (no Backprop of the delta errors)
  – Avoid large weights (neuron saturation)

• Learning rate
  – The learning rate can change during the learning (higher when the gradient is small)
  – There is no “magic solution”!
Backprop Heuristics (con’t)

- Activation function: symmetric vs asymmetric
- Target values to avoid saturation
- Input normalization
- The input variables (coordinates) should be incorrelated
- Normalization w.r.t. to the fan-in of the first hidden layer
An introduction to Multilayered Neural Networks

The problem of local minima

simple example of sub-optimal learning

local minima as symmetric configurations ...
Basic results

- No local minima in the case of linearly separable patterns (Gori & Tesi, IEEE-PAMI92)
- No local minima if the number of hidden units is equal to the number of examples (Yu et al, IEEE-TNN95)
- A general comment:
  - The theoretical investigations on this problem have not relevant results for the design of the networks.
Complexity issues: Backprop is optimal

Classic numerical algorithms require

$O(W^2)$ for the computation of a single partial derivative and
$\Theta(W)$ for the whole gradient computation

Backprop requires $\Theta(W)$ for the whole gradient computation

e.g. in the case of 10,000 parameters, we need 10,000 FPO versus 100 millions FPO!!! This is one of the main reasons of the success of Backpropagation!
An introduction to Multilayered Neural Networks

Crossvalidation penalties to limit large weights

Courtesy MathWorks

Marco Gori - IEEE Expert Now Course
The applicative perspective: pattern recognition
An introduction to Multilayered Neural Networks

MLP as classifiers: a simple pre-processing

input

Preprocessed input

output
The cascade
Autoassociator-based classifiers

Motivation: better behavior for pattern verification
Gori & Scarselli, IEEE-PAMI-98
Some successful applications

• Airline market assistance, BehavHeuristics Inc
• Automated Real Estate Appraisal Systems - HNC Software
• OCR - Caere, Audre Recognition System
• Path planning, NeuroRoute - Protel
• Electronic nose - AromaScan Inc
• Quality control - Anheuser-Busch, Dunlop
• Banknote acceptor BANK, DF Elettronica Florence