• Multi-layer perceptrons (MLP) constitute the most famous and most employed type of neural networks [1].

• At first, they were introduced in the context of classification, and it was soon recognized that they had the property of universal approximation capability.

• The introduction of the back-propagation algorithm made them flexible enough for many applications.
• Their basic structure for prediction is:

\[ x_{n-1} \rightarrow 1 \]
\[ x_{n-2} \rightarrow 2 \]
\[ x_{n-p} \rightarrow C \]

\[ \text{out} \]
MLP AND PREDICTION (2)

- The output of the ith neuron in the intermediate (hidden) layer is:

\[ y_i = f \left( \theta_j + \sum_{j=1}^{p} w_{ij} x_{n-j} \right) \]

with \( f(.) \) a sigmoidal function such as \( f(u) = \tanh(au) \).

- The final output is a weighted sum of the outputs in the hidden layer.
• As for the polynomial predictor, one considers a NAR formulation:

\[ x_n = g(x_{n-1}, \ldots, x_{n-p}) + \epsilon_n \]

and one estimates \( g(.) \) in the least-square sense using the data at hand. That is, one uses the universal approximation capability of MLPs.

• Of course, the input vector to predict \( x_n \) is

\[ \mathbf{x}_n = [x_{n-1}, x_{n-2}, \ldots, x_{n-p}] . \]
• It is possible to show that, when $p$ increases, the size of the MLP increases more slowly than that of a polynomial predictor. But:
  – It is almost impossible to relate the values of the MLP parameters to the characteristic of the predicted signal (black box effect).
  – Convergence of the back-propagation algorithm is sometimes problematic.
  – Adaptive prediction is not very efficient.
MLP AND PREDICTION (5)

- Example: prediction of RR intervals ($p = 3, 5$ neurons in the hidden layer). ESR: -16 dB
MLP AND PREDICTION (6)

- Generalization: prediction on another part of the signal. ESR = -13 dB
• Radial basis functions (RBF) constitute another type of neural networks, issued from interpolation theory.

• By some aspects, RBF combine the advantages of polynomial models and MLP:
  – Universal approximation property,
  – Estimation of parameters is simple,
  – There is a physical interpretation of these parameters.
• Let us suppose we have $N$ pairs $\{x_i, y_i = g(x_i)\}, \ i = 1, \ldots, N$, with $x_i$ $p$-dimensional vectors, $y_i$ scalars, and $g(.)$ a continuous function.

• the interpolation of $g(.)$ from these $N$ pairs is an ill-posed one, because there is usually no information on $g(.)$ between these pairs.

• A sensible approach, called regularization, has been developed to tackle this type of problem.
In this approach, one hypothesizes that $g(.)$ should not oscillate erraticaly, and should present some degree of smoothness. This translates into defining an interpolating function $G(.)$ minimizing a composite criterion:

$$C(G) = \sum_{i=1}^{N} [y_i - G(x_i)]^2 + \rho \|PG\|^2$$

with $P$ a differential operator.
RBF AND INTERPOLATION (3)

• Parameter $\rho$ balances the relative influence of the two terms. The first one deals with the fidelity with respect to the data, the second with interpolation smoothness.

• It is possible to show that, if $P$ is an infinite sum of differential operators of increasing degree, and rotation and translation invariant, then $G(.)$ is:

$$G(x) = \sum_{i=1}^{N} w_i \phi(\|x - x_i\|) \quad \text{with} \quad \phi(s) = \exp\left(-\frac{s^2}{2\beta^2}\right)$$
• Parameter $\beta$ must be specified with respect to the regularization parameter $\rho$. If there is a true interpolation then one must have:

$$G(x_i) = y_i, \quad i = 1, \ldots, N$$

and one must solve the following linear system:

$$
\begin{bmatrix}
\phi(\|x_1 - x_1\|) & \cdots & \phi(\|x_1 - x_N\|)\\
\vdots & \ddots & \vdots \\
\phi(\|x_N - x_1\|) & \cdots & \phi(\|x_N - x_N\|)
\end{bmatrix}
\begin{bmatrix}
w_1 \\
\vdots \\
w_N
\end{bmatrix}
= 
\begin{bmatrix}
y_1 \\
\vdots \\
y_N
\end{bmatrix}
$$
Thus a matrix equation:

\[ A\mathbf{w} = \mathbf{y} \]

- It is possible to show that if all vectors \( \{x_i\} \) are different, matrix \( A \) is always invertible, and it will always be possible to determine the coefficients \( \{w_i\} \) of \( \mathbf{w} \).
- The value of \( \beta \) defines the behavior of \( G(.) \) between the \( \{x_i\} \). The larger it is, the smoother \( G(.) \) is.
• Example

True function (not known).

\( \beta = 5 \)

\( \beta = 2 \)

\( \beta = 0.5 \)
• This approach of interpolation presents some limitations:
  – The fact that the interpolating function must include all pairs \( \{x_i, y_i = g(x_i)\} \) makes the approach highly sensitive to noise (erroneous values) and thus prone to overfitting.
  – When the number of pairs increases \( G(.) \) becomes soon complex.
The idea that has been proposed to create a new type of neural networks is quite simple. One considers a slightly different formulation:

$$ G(x) = \sum_{j=1}^{M} w_j \phi\left( \|x - c_j\| \right) \text{ avec } \phi(s) = \exp\left( -\frac{s^2}{2 \beta^2} \right) $$

with $M < N$, and vectors $\{c_j\}$ called centers, which are not constrained to be a subset of the $\{x_i\}$. Function $G(.)$ will now approximate $g(.)$, typically in the least-square sense.
• To sum up, the RBF network approximates locally $g(.)$ in the neighborhood of each center, and merges these local approximations to create a global one.

• The nice feature of RBF networks is that, once the centers and $\beta$ have been defined, the determination of coefficients $\{w_j\}$ is simple, since $G(.)$ depends linearly on them.
• If least-square estimation is used, one must solve:

\[
\begin{bmatrix}
\phi(\|x_1 - c_1\|) & \cdots & \phi(\|x_1 - c_M\|) \\
\phi(\|x_2 - c_1\|) & \cdots & \phi(\|x_2 - c_M\|) \\
\vdots & & \vdots \\
\phi(\|x_N - c_1\|) & \cdots & \phi(\|x_N - c_M\|)
\end{bmatrix}
\begin{bmatrix}
w_1 \\
\vdots \\
w_M
\end{bmatrix}
= 
\begin{bmatrix}
y_1 \\
\vdots \\
y_N
\end{bmatrix}
+ 
\begin{bmatrix}
e_1 \\
\vdots \\
e_N
\end{bmatrix}
\]

that is \( A w = y + e \). \( A \) is now an \( N \times M \) matrix.

• One may add a column of "1" to \( A \) if \( y \) is significantly non zero-mean.
• When minimizing $\|A\mathbf{w} - \mathbf{y}\|$, one performs of course an orthogonal projection of vector $\mathbf{y}$ on the subspace generated by the columns of $A$.

• Since matrix $A$ may be ill-conditioned it is better to use a robust estimation scheme, typically SVD.

• But of course centers should not be too close to each other.
• Note first there is no really optimal way to select the centers. Also, we will come back later to the problem of determining the number of centers.

• One of the firsts methods proposed consisted in selecting centers randomly (uniformly), in the hypercube defined by the \( \{ x_i \} \). This is not very efficient, because the spatial distribution of the \( \{ x_i \} \) is not taken into account.
A second approach consists in selecting randomly the centers among the \( \{x_i\} \). This is already better, since statistically those centers are representative of the spatial distribution of the \( \{x_i\} \). That is, there are more centers in the regions of space where the \( \{x_i\} \) are clustered. However, the constraint remains that the centers are a subset of the \( \{x_i\} \).
• A solution yielding better performance is to select centers using *learning vector quantization* (LVQ), which is used in other contexts too to find representatives of vector sets. LVQ works as follows:

1. The initial centers are chosen \( \{ c_j \} \) at random in the hypercube defined by the \( \{ x_i \} \).
2. Vectors \( \{x_i\} \) are presented, generally for several epochs. At iteration \( k \), the following correction is applied to center \( c_j \):

\[
c_j(k+1) = c_j(k) + \alpha(k)[x(k) - c_j(k)]
\]

if

\[
\|x(k) - c_j(k)\| < \|x(k) - c_m(k)\|, \quad m \neq j
\]

with \( x(k) \) the vector presented at iteration \( k \). The net effect is to draw \( c_j \) closer to \( x(k) \), and thus possibly to a cluster.
Parameter $\alpha(k)$ must decrease with $k$, generally using:

$$\alpha(k) = \alpha_0 \left( 1 - \frac{k}{K} \right)$$

$K$ total number of iterations, and $\alpha_0$ a constant (typically $\alpha_0 = 0.05$).

- It is to be noted that LVQ is sensitive to the initial center selection.
• **Example:** LVQ for $M = 5$ centers, vectors $\{x_i\}$ drawn from two bi-dimensional Gaussian pdf.
Another scheme, less sensitive to initial center choice, is *Lloyd-Max vector quantization algorithm*. Centers \( \{c_j\} \) (sometimes called in this context *centroids*) are chosen such that they minimize a distortion criterion:

\[
D = \sum_{j=1}^{M} \sum_{x_i \in V_j} \| x_i - c_j \|^2
\]
Subsets $V_j$ contain vectors $x_j$ such that $c_j$ is the center closest to them. So to speak, $c_j$ must be a good representative of $V_j$. Lloyd-Max algorithm works as follows

1. Initial random selection of $\{c_j\}$ among the $\{x_i\}$

2. Update:

$$c_j \rightarrow c_j = \text{mean}(x_i \in V_j)$$

3. If distortion $D$ stabilizes, stop, otherwise back to 2.
Example: Lloyd-Max for $M = 5$ centers, vectors $\{x_i\}$ drawn from two bi-dimensional Gaussian pdf.

- data
- centroids
• Once the centers \( \{c_j\} \) have been chosen, parameter \( \beta \) can be determined.

• This is the same as for interpolation: too large a \( \beta \) means too strong an interaction (overlap) between the local approximations. Too small a \( \beta \) means a "bumpy" approximation.

• A good empirical rule is:

\[
\beta = \frac{d}{\sqrt{M}} \quad \text{with} \quad d = \max \|c_i - c_j\|, \quad 1 \leq i, j \leq M
\]
MULTIPLE $\beta$

- Nothing imposes to use the same value for $\beta$ for each RBF.
- It is even intuitively appealing that $\beta$ be smaller in regions with many centers, since the spatial influence of the RBFs corresponding to these centers should be smaller.
- A good empirical rule for $\beta_j$ corresponding to center $c_j$ is:

$$\beta_j = \frac{1}{N \sqrt{8}} \sum_{i=1}^{N} \left\| x_i - c_j \right\|$$
• As for MLP, one considers a NAR formulation:

\[ x_n = g(x_{n-1}, \ldots, x_{n-p}) + \varepsilon_n \]

and one estimates \( g(.) \) in the least-square sense using the data at hand. That is, one uses the universal approximation capability of RBF networks.

• Of course, the input vector to predict \( x_n \) is

\[ x_n = [x_{n-1}, x_{n-2}, \ldots, x_{n-p}] \].
Example: prediction of RR intervals ($p = 3, 20$ RBF, multiple $\beta$). ESR: -10 dB
• Generalization: prediction on another part of the signal. ESR = -4 dB
• Center selection, which conditions the choice of parameter $\beta$ also, has an important role in RBF network performance.

• LVQ and Lloyd-Max algorithms are intuitively appealing. There are however other interesting schemes [3].

• Again, we note that solving $Aw = y + e$ in the least squares sense means performing an orthogonal projection of $y$ on the linear subspace generated by the columns of $A$. 
• One can apply the same procedure as for term selection in polynomial prediction, that is a Gram-Schmidt orthogonalization on selected columns (orthogonal least squares, OLS).

• At each iteration, one adds in $A$ the column yielding maximum error reduction. This column is defined by a pair $(c_j, \beta_j)$, and center $c_j$ is selected among the vectors $\{x_i\}$.
One proceeds as follows:

1) The first center is selected by examining all the \( \{x_i\} \), and keeping the smallest least squares error.

2) Successive centers are chosen among the remaining \( \{x_i\} \), with an orthogonalization of \([\phi(x_1), \ldots, \phi(x_N)]^T\) with respect to the terms already selected. In this way the center giving maximum error reduction is spotted.
• **Example:** prediction of RR intervals \((p = 3, 5\) RBF, multiple \(\beta\)). ESR: -11 dB

Signal Prediction
- samples corresponding to selected \(x_i\)
How many RBFs?

\[ MDL = N \cdot \ln \sigma_e^2 + M \cdot \ln N + \text{?} \]

- **Data**
- **Model**
  - linear parameters only!
- **Model**
  - nonlinear parameters
Clustering of the State Space (Sunspot Series)

- Clustering of the State Space
- Sunspot Series

- Scattered data points
- Axes labeled: $x(n-2)$ vs. $x(n-1)$
- Graph indicating MSE (in %) vs. Added Center Noise

- Noise added to the data
- Graph shows fluctuations in MSE across different noise levels

Signal Processing Laboratory
Swiss Federal Institute of Technology, Lausanne
Mackey-Glass Time Series

\[ x(n-1), x(n-2) \]

\[ \text{MSE (in \%)} \]

\[ 0.04, 0.05, 0.06, 0.07, 0.08, 0.09, 0.1, 0.11 \]

\[ 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1, 1.1, 1.2, 1.3, 1.4 \]

\[ 0.1, 0.15, 0.2, 0.25, 0.3 \]

Added Center Noise
State-space Quantization

2 bits
4 bits
6 bits
The Coding of a Center

1 1 0 1 1 0

\[ x(n-1) \]

\[ x(n-2) \]
The Best Subset of Centers (4 bits)
The Best Subset of Centers (6 bits)
3-D State Space Quantization (3 bits)
3-D State Space Quantization (6 bits)
MDL-Criterion for RBF Networks

\[
\text{MDL} = N \cdot \ln \sigma_e^2 + M \cdot (\ln N + B \cdot \ln 2)
\]

B: Number of bits to encode one center
Prediction Comparison (Sunspot)

![Graph showing relative MSE vs. Nodes for different bit sizes and LVQ. The graph indicates a decrease in relative MSE as the number of nodes increases, with distinct points for 6 bits, 4 bits, 8 bits, and LVQ.]
Prediction Comparison (Mackey-Glass)

![Graph showing relative MSE vs Nodes for different bit lengths (4 bits, 6 bits, 8 bits) and LVQ.](image)

- **Nodes**: 5 to 11
- **relative MSE**: 0 to 0.2

- **4 bits**
- **6 bits**
- **8 bits**

**LVQ**
• It is of course possible to derive an adaptive version of RBF network prediction.

• In what follows, reference to time index $k$ will not be explicit.

$w(i)$ ith component of coefficient vector $w$

$\phi(i)$ ith RBF

$\phi$ RBF vector

• instantaneous square error is thus

$$\varepsilon^2 = [y - G(x)]^2 = [y - w^T\phi]^2$$
• In LMS, the updates are:

\[ w \rightarrow w - \mu_1 \frac{\partial \varepsilon^2}{\partial w} = w - 2\mu_1 \varepsilon \frac{\partial \varepsilon}{\partial w} = w + 2\mu_1 \varepsilon \frac{\partial G(x)}{\partial w} \]

\[ = w + 2\mu_1 \varepsilon \phi \]

\[ c_j \rightarrow c_j + 2\mu_2 \varepsilon \frac{\partial G(x)}{\partial c_j} = c_j + 2\mu_2 \varepsilon w(j) \phi(j)(x - c_j) \]

\[ \beta \rightarrow \beta + 2\mu_3 \varepsilon \frac{\partial G(x)}{\partial \beta} = \beta + \frac{2\mu_3 \varepsilon}{\beta^3} \sum_{i=1}^{M} \|x - c_i\|^2 \phi(i) \]
• Example: prediction of RR intervals ($p = 3, 5$ RBF, unique $\beta$). ESR: -8 dB

**Signal Prediction**

*Multiple epochs before convergence*
• Remarks:
  – Better than batch algorithm, but not impressive
  – Tuning of adaptation parameters is uneasy (except for $w$ for which it is possible to normalize by $\phi^T \phi$)

• But maybe the gradient approach is not suited to center and $\beta$ parameter update. For instance, nothing guarantees that the mean square error has a unique minimum.
• Why not *decouple* the update of $\mathbf{w}$, for which LMS should work well, from that of the centers and $\beta$?

• One might use an LVQ-style update for the centers, then modify $\beta$ by using the empirical rule based on the distance between centers.

• But it is necessary to define a link between the two update schemes.
1. One first computes the modification on $c_i$, the center closest to $x$ with LVQ (but $\alpha$ must remain constant to preserve adaptability):

$$c_i \rightarrow c_i + \Delta c_i = c_i + \alpha(x - c_i)$$

2. If the maximum distance between centers changes, $\beta$ should become $\beta + \Delta \beta$, by using the empirical rule for unique $\beta$.

The output changes from $G(x)$ to $G(x) + \Delta G(x)$. 
• But: \[ \Delta G(x) \approx (\Delta c_i)^T \frac{\partial G(x)}{\partial c_i} + \Delta \beta \frac{\partial G(x)}{\partial \beta} \]

• Thus:

\[
\frac{\partial (G(x) + \Delta G(x))}{\partial w} \approx \frac{\partial G(x)}{\partial w} + (\Delta c_i)^T \frac{\partial G(x)}{\partial w \partial c_i} + \Delta \beta \frac{\partial G(x)}{\partial w \partial \beta}
\]

• And the instantaneous square error becomes:

\[ \varepsilon^2 = [y - G(x) - \Delta G(x)]^2 \]

The coefficient vector \( w \) is updated with:

\[
w \rightarrow w - \mu_1 \frac{\partial \varepsilon^2}{\partial w} = w + 2 \mu \varepsilon \frac{\partial (G(x) + \Delta G(x))}{\partial w}
\]
With some computations:

\[
(\Delta c_i)^T \frac{\partial G(x)}{\partial w \partial c_i} = \frac{\alpha}{\beta^2} \| x - c_i \|^2 \phi(i) \begin{bmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix}
\]

\[
\frac{\partial G(x)}{\partial w \partial \beta} = \frac{1}{\beta^3} \begin{bmatrix} \| x - c_1 \|^2 \phi(1) \\ \vdots \\ \| x - c_M \|^2 \phi(M) \end{bmatrix}
\]
Example: prediction of RR intervals ($p = 3, 5$ RBF, unique $\beta$). ESR: -10.7 dB
• Remarques:
  – Prediction performance improves in the non-stationary intervals.
  – It is a lot easier to tune $\mu$ for $w$ and $\alpha$ for the centers.
  – This approach is easily modified for the multiple $\beta$ case.
REFERENCES


