Artificial Neural Networks for Fault Diagnostics and Prognostics in Nuclear Power Plants Components and Systems

Piero Baraldi
Politecnico di Milano
Fault Diagnostics and Prognostics

Nuclear Power Plant Component

Measured signals

Detect

Diagnose

Predict

Anomalous operation

Normal operation

Malfunctioning type (classes)

Remaining Useful Life (RUL)
Data-driven fault diagnostics
In This Lecture

1. Fault diagnostics: what is it?
2. Procedural steps for developing a fault diagnostic system
3. Artificial Neural Networks
Fault diagnostics: What is it

Fault diagnostics: What is it

Forcing functions

Industrial system

Transient

Measured signals

Transient

Fault detection early recognition

Danger!

Fault diagnosis: (correct assignment)
Fault diagnostics: transient classification

Fault Class 1
Fault Class 2
... 
Fault Class c

Fault Class 1
Fault Class 2
... 
Fault Class c

Measured signals

x1
x2
x3

X1(t)
X2(t)
X3(t)

Transient

Diagnostic System

Classifier
In This Lecture

1. Fault diagnostics: what is it?
2. Procedural steps for developing a fault diagnostic system
3. Objectives of a fault diagnostic system
4. Unsupervised classification methods
5. Supervised classification methods
1. Identify fault/degradation classes:
   - System Analysis (FMECA, Event Tree Analysis, ...)
   - Good engineering sense of practice
     \[ \{C_1, C_2\} \]
   - Analysis of the historical data

\[ x_1 = \text{signal 1} \]
\[ x_2 = \text{signal 2} \]
Empirical Approach to Fault Classification

1. Identify fault/degradation classes
2. Collect data couples: \( \langle \text{measurements}, \text{class} \rangle \)

- **Historical data**
  - Fault transients in industrial systems may be rare
  - Information on the class of the fault originating the transient is not always available

\[ x_1 = \text{representative signal 1} \]
\[ x_2 = \text{representative signal 2} \]
Example: turbine transients in a Nuclear Power Plant

Available data

• 5 signals (operational conditions) + 7 signals (turbine behaviour)
• 149 shut-down transients from a nuclear power plant
• Types of anomalies are unknown
Empirical Approach to Fault Classification

1. Identify fault classes
2. Collect data couples: <measurements, fault class>

- Historical plant data
  - Fault transients in industrial systems may be rare
  - Information on the class of the fault originating the transient is not always available

- Physics-based simulator
Empirical Approach to Fault Classification

1. Identify fault classes
2. Collect data couples: \(<\text{measurements, fault class}>\)

3. Develop the empirical classification algorithm using as \textit{training set} the collected data couples: \(<\text{measurements, fault classes}>\)

Measured signals

\begin{align*}
X_1(t) & \quad \text{Fault Class 1} \\
X_2(t) & \quad \text{Fault Class 2} \\
X_3(t) & \quad \text{Fault Class c}
\end{align*}
Data-driven fault prognostics approaches

• Machine learning and data mining algorithms
  o Support Vector Machines
  o K-Nearest Neighbour Classifier
  o Fuzzy similarity-based approaches
    o Artificial Neural Networks
  o Neurofuzzy Systems
  o Relevant Vector Machines
  o …
- Prognostics
  - What is it?
  - Sources of information
  - Prognostic approaches
Fault prognostics: What is it?

Predict

Remaining Useful Life (RUL)
○ Prognostics
  ○ What is it?
  ○ Sources of information
  ○ Prognostic approaches
Prognostics: Sources of Information

- Life durations of a set of similar components
- A physics-based model of the degradation
- Degradation trajectories of similar components

- Threshold of failure
- Current degradation trajectory
- External/operational conditions
Component: turbine blade
Degradation mechanism: creeping
Component: turbine blade
Degradation mechanism: creeping

Degradation indicator: blade elongation $\chi(t) = \frac{\text{Length}(t) - \text{initial length}}{\text{initial length}}$
Sources of information for prognostics

• Life durations of a set of similar components which have already failed:
  \[ T_1, T_2, \ldots, T_n \]
Sources of information for prognostics

- Life durations of a set of similar components which have already failed
- Threshold of failure: $x^{th}$

![Diagram](image)

Fault Initiation
Sources of information for prognostics

- Life durations of a set of similar components which have already failed
- Threshold of failure: $x^{th}$

"A blade is discarded when the elongation, $x$, reaches 1.5%"
Sources of information for prognostics

- Life durations of a set of similar components which have already failed
- Threshold of failure
- A sequence of observations collected from the degradation initiation to the present time (current degradation trajectory): $z_1, z_2, \ldots, z_k$
Sources of information for prognostics

- Life durations of a set of similar components which have already failed
- Threshold of failure
- A sequence of observations collected from the degradation initiation to the present time (current degradation trajectory): $z_1, z_2, \ldots, z_k$

Elongation measurements = past evolution of the degradation indicator
Sources of information for prognostics

- Life durations of a set of similar components which have already failed
- Threshold of failure
- A sequence of observations collected from the degradation initiation to the present time (current degradation trajectory)
- Degradation trajectories of similar components

\[ z(t) \]

Fault Initiation
Sources of information for prognostics

- Life durations of a set of similar components which have already failed
- Threshold of failure
- A sequence of observations collected from the degradation initiation to the present time (current degradation trajectory)
- Degradation trajectories of similar components
- Information on external/operational conditions (past – present - future)

Past, present and future time evolution of: $u_1, u_2, \ldots, u_k, u_{k+1}, \ldots$
Sources of information for prognostics

- Life durations of a set of similar components which have already failed
- Threshold of failure
- A sequence of observations collected from the degradation initiation to the present time (current degradation trajectory)
- Degradation trajectories of similar components
- Information on external/operational conditions (past – present - future)

\[ u_1 = T = \text{temperature} \]
\[ u_2 = \theta_r = \text{rotational speed} \]
Sources of information for prognostics

- Life durations of a set of similar components which have already failed
- Threshold of failure
- A sequence of observations collected from the degradation initiation to the present time (current degradation trajectory)
- Degradation trajectories of similar components
- Information on external/operational conditions (past – present - future)
Sources of information for prognostics

- Life durations of a set of similar components which have already failed
- Threshold of failure
- A sequence of observations collected from the degradation initiation to the present time (current degradation trajectory)
- Degradation trajectories of similar components
- Information on external/operational conditions (past – present - future)
Sources of information for prognostics

- Life durations of a set of similar components which have already failed
- Threshold of failure
- A sequence of observations collected from the degradation initiation to the present time (current degradation trajectory)
- Degradation trajectories of similar components
- Information on external/operational conditions (past – present - future)
- A physics-based model of the degradation process

\[ x_k = f_k(x_{k-1}, \ldots, x_1, u_{k-1}, \omega_{k-1}) \]
Sources of information for prognostics

- Life durations of a set of similar components which have already failed
- Threshold of failure
- A sequence of observations collected from the degradation initiation to the present time (current degradation trajectory)
- Degradation trajectories of similar components
- Information on external/operational conditions (past – present - future)
- Measurement equation
- A physics-based model of the degradation process

**Norton law for creep growth**

\[
\frac{dx}{dt} = A \exp\left(-\frac{Q}{RT}\right) \varphi^n
\]

- \(x\) = blade elongation
- \(T\) = temperature
- \(\varphi = K\theta_r^2\) = applied stress
- \(\theta_r\) = rotational speed
- \(A, Q, n\) = equipment inherent parameters
- **Arrhenius law**
- **External/operational conditions**
- Prognostics
  - What is it?
  - Sources of information
  - Prognostic approaches
Prognostic approaches

- Experience-Based
  - Life durations of a set of similar components

- Model-Based
  - A physics-based model of the degradation
  - Degradation trajectories of similar components

- Data-Driven
  - Threshold of failure
  - Current degradation trajectory
  - External/operational conditions

- Timeline and or other relevant details
Prognostic approaches

Life durations of a set of similar components

Data-Driven

Model-Based

Experience-Based

- Measurement equation
- A physics-based model of the degradation

- Degradation trajectories of similar components

- Threshold of failure
- Current degradation trajectory
- External/operational conditions

Hybrid
Data driven fault prognostics

- **When?**
  - Understanding of first principles of component degradation is not comprehensive
  - System sufficiently complex that developing an accurate physical model is prohibitively expensive

- **Advantages:**
  - Quick and cheap to be developed
  - They can provide system wide-coverage

- **Disadvantages**
  - They require a substantial amount of data for training:
    - Degradation data are difficult to collect for new or highly reliable systems (running systems to failure can be a lengthy and costly process)
    - Quality of the data can be low due to difficulty in the signal measurements
Artificial Neural Networks
The problem

\[ y = g(x) \]

\( g \) is unknown!
- \( g \) is highly non linear
- \( g \) is complex

How to build an empirical model using input/output data?

\[ x_1^{(1)}, x_2^{(1)}, x_3^{(1)} \mid t_1^{(1)}, t_2^{(1)} \]
\[ x_1^{(2)}, x_2^{(2)}, x_2^{(2)} \mid t_1^{(2)}, t_2^{(2)} \]
\[ \ldots \]
What are Artificial Neural Networks (ANN)?
**What Are (Artificial) Neural Networks?**

- Multivariate, non-linear interpolators capable of reconstructing the underlying complex I/O nonlinear relations by combining multiple simple functions.

- An artificial neural network is composed by several simple **computational units** (also called nodes or neurons).
The computational unit

- Computational unit (node, neuron)
  The output ($u$) of a node is the result of a (possibly nonlinear) transformation ($f$) on the input variables ($y_1, ..., y_n$) transmitted along the links of the graph.

$$y = \sum_{k=1}^{n} y_k$$
$$u = f(y)$$

$f = \text{activation function can be:}$
- Linear $f(y) = y$
- Sigmoidal $f(y) = 1/(1+e^y)$
- Others

$$f' = f(1 - f)$$
The computational unit: other representations

\[ y_1 \ y_2 \ \cdots \ y_n \]

represented as

\[ f \]

or

\[ u \]
An artificial neural network is composed by several simple computational units (also called nodes or neurons) directionally connected by weighted connections.

Connection between nodes $i$ and $j$

$$y_{mj} = w_{mj}u_j$$

$w_{mj}$ = synapsis weight between nodes $j$ and $m$
An artificial neural network is composed by several simple **computational units** (also called nodes or neurons) directionally connected by **weighted connections** organized in a proper architecture.
An example of ANN architecture

→ The multilayered feedforward ANN
Number of connections
$11 \times 7 + 8 \times 5 = 117$
**INPUT LAYER:**
each $k$-th node ($(k=1, 2, ..., n_i)$) receives the value of the $k$-th component of the input vector $\vec{x}$ and delivers the same value.

**HIDDEN LAYER:**
each $j$-th node ($(j=1, 2, ..., n_h)$) receives

and delivers $\sum_{k=1}^{n_i} x_k w_{jk} + w_{j0}$ with $f^h$ typically sigmoidal.
**OUTPUT LAYER:**

Each $l$-th node ($l=1, 2, \ldots, n_o$) receives

$$\sum_{j=1}^{n_h} u_j^h w_{lj} + w_{l0}$$

and delivers $u_l^o = f\left(\sum_{j=1}^{n_h} u_j^h w_{lj} + w_{l0}\right)$ $f$ typically linear or sigmoidal.
Why are we speaking of “Neural Networks”?

- Biological basis of ANN
- Human brain is a densely interconnected network of approximately $10^{11}$ neurons, each connected to, on average, $10^4$ others.
- Neuron activity is excited or inhibited through connections to other neurons.
Biological Motivation: the neuron

- The *dendrites* provide input signals to the cell.
- The *axon* sends output signals to cell 2 via the axon terminals. These axon terminals merge with the dendrites of cell 2.
What is the mathematical basis behind Artificial Neural Networks?
Neural networks are universal approximators of multivariate non-linear functions.

**KOLMOGOROV (1957):**
For any real function \( f(x_1, x_2, \ldots, x_n) \) continuous in \([0,1]^n\), \( n \geq 2 \), there exist \( n(2n+1) \) functions \( \psi_{ml}(\xi) \) continuous in \([0,1]\) such that

\[
f(x_1, x_2, \ldots, x_n) = \sum_{l=1}^{2n+1} \phi_l \left( \sum_{m=1}^{n} \psi_{ml}(x_m) \right)
\]

where the \( 2n+1 \) functions \( \phi_l \)'s are real and continuous.

Thus, a total of \( n(2n+1) \) functions \( \psi_{ml}(\xi) \) and \( 2n+1 \) functions \( \phi_l \) of one variable represent a function of \( n \) variables.
Neural networks are universal approximators of multivariate non-linear functions.

**CYBENKO (1989):** Let $\sigma(\cdot)$ be a sigmoidal continuous function. The linear combinations

$$\sum_{j=0}^{N} \alpha_j \sigma \left( \sum_{i=1}^{n} x_i w_{ij} + \vartheta_j \right)$$

are dense in $[0,1]^n$.

In other words, any function $f: [0,1]^n \rightarrow \mathbb{R}$ can be approximated by a linear combination of sigmoidal functions. **Notice that the theorem does not specify the values of $N$, $w_{ij}$ and $\alpha_j$!**
How to train an ANN?

→ Setting the ANN parameters
The ANN parameters to be set

- Once the ANN architecture has been fixed (number of layers, number of nodes for layer), the only parameters to be set are the *synapsis weights* \((w_{lj}, w_{jk})\).
Available input/output patterns

\[ x_1^{(1)}, x_2^{(1)} \mid t^{(1)} \]
\[ x_1^{(2)}, x_2^{(2)} \mid t^{(2)} \]
\[ \cdots \]
\[ x_1^{(p)}, x_2^{(p)} \mid t^{(p)} \]
\[ \cdots \]
\[ x_1^{(np)}, x_2^{(np)} \mid t^{(np)} \]
Training Objective: minimize the *average squared output deviation error* (also called Energy Function):

\[
E = \frac{1}{2n_p n_o} \sum_{p=1}^{n_p} \sum_{l=1}^{n_o} (u_{pl}^o - t_{pl})^2
\]

- TRUE \(l\)-th output of the \(p\)-th training pattern
- ANN \(l\)-th output of the \(p\)-th training pattern
The Training objective: graphical representation

- $E$ is a function of the ANN outputs
- $E$ is a function of the synapsis weights $[w_{jk}]$
The error back propagation algorithm

1. Initialize weights to random values:
   \[ w_{jk}^{(0)} = \text{rand} \]

2. While \( E \) is small:
   - Update \( w_{jk}^{(i)} \) using the gradient descent method:

   \[
   w_{jk}^{(i+1)} = w_{jk}^{(i+1)} - \eta \frac{\partial E}{\partial w_{jk}}
   \]

Learning coefficient gradient \( \nabla E(w) = \left( \frac{\partial E}{\partial w_1}, \frac{\partial E}{\partial w_2} \right) \)
Error Backpropagation (hidden-output)

- Updating of $w_{lj}$:

$$w_{lj}^{(i+1)} = w_{lj}^{(i)} - \eta \frac{\partial E}{\partial w_{lj}}$$

with

$$E = \frac{1}{2n_p n_o} \sum_{p=1}^{n_p} \sum_{l=1}^{n_o} (u_{pl}^o - t_{pl})^2$$

- Without loss of generality, set $n_p = 1$

$$E = \frac{1}{2n_o} \sum_{l=1}^{n_o} (u_l^o - t_l)^2$$

$$\frac{\partial E}{\partial w_{lj}} = \frac{2(u_l^o - t_l)}{2n_o} \frac{\partial u_l^o}{\partial w_{lj}}$$
Error Backpropagation (hidden-output)

\[ E = \frac{1}{2n_o} \sum_{l=1}^{n_o} (u_l^o - t_l)^2 \]

\[
\frac{\partial E}{\partial w_{lj}} = \frac{2}{2n_o} \frac{(u_l^o - t_l)}{\partial w_{lj}} \]

Output layer neuron

\[ y_{l1}^o \]
\[ y_{lj}^o \]
\[ y_{ln_h}^o \]

\[ f(\ ) \]

\[ u_l^o \]
Error Backpropagation (hidden-output)

\[ E = \frac{1}{2n_o} \sum_{l=1}^{n_o} (u_l^o - t_l)^2 \]

\[ \frac{\partial E}{\partial w_{lj}} = \frac{2(u_l^o - t_l)}{2n_o} \frac{\partial u_l^o}{\partial w_{lj}} = \frac{(u_l^o - t_l)}{n_o} \frac{\partial u_l^o}{\partial y_l^o} \frac{\partial y_l^o}{\partial w_{lj}} \]

\[ = \frac{(u_l^o - t_l)}{n_o} f'(y_l^o) \frac{\partial}{\partial w_{lj}} \sum_{j'=1}^{n_h} y_{lj'}^o \]

Output layer neuron
Error Backpropagation (hidden-output)

\[ E = \frac{1}{2n_o} \sum_{l=1}^{n_o} (u^o_l - t_l)^2 \]

\[ \frac{\partial E}{\partial w_{lj}} = \frac{2(u^o_l - t_l)}{2n_o} \frac{\partial u^o_l}{\partial w_{lj}} = \frac{(u^o_l - t_l)}{n_o} \frac{\partial u^o_l}{\partial y^o_l} \frac{\partial y^o_l}{\partial w_{lj}} \]

\[ = \frac{(u^o_l - t_l)}{n_o} \frac{\partial u^o_l}{\partial y^o_l} \frac{\partial (\sum_{j'=1}^{n_h} y^o_{lj'})}{\partial w_{lj}} = \frac{(u^o_l - t_l)}{n_o} \frac{\partial u^o_l}{\partial y^o_l} \frac{\partial y^o_l}{\partial w_{lj}} \]

\[ y^o_{lj} = w_{lj} \cdot u^h_j \]
Error Backpropagation (hidden-output)

- Updating of $w_{lj}$:

$$w_{lj}^{(i+1)} = w_{lj}^{(i)} - \eta \frac{\partial E}{\partial w_{lj}}$$

with

$$E = \frac{1}{2n_p n_o} \sum_{p=1}^{n_p} \sum_{l=1}^{n_o} (u_{pl}^o - t_{pl})^2$$

- Without loss of generality, set $n_p = 1$

$$E = \frac{1}{2n_o} \sum_{l=1}^{n_o} (u_l^o - t_l)^2$$

$$\frac{\partial E}{\partial w_{lj}} = \frac{2(u_l^o - t_l)}{2n_o} \frac{\partial u_l^o}{\partial w_{lj}} = \frac{(u_l^o - t_l)}{n_o} \frac{\partial u_l^o}{\partial y_l^o} \frac{\partial y_l^o}{\partial w_{lj}} = \frac{(u_l^o - t_l)}{n_o} f'(y_l^o) u_j^h$$
Error Backpropagation (hidden-output)

- Updating of $w_{lj}$:

\[
\begin{align*}
    w_{lj}^{(i+1)} &= w_{lj}^{(i+1)} - \eta \frac{\partial E}{\partial w_{lj}} \\
    \text{with} \quad \frac{\partial E}{\partial w_{lj}} &= \frac{(u_l^o - t_l)}{n_o} f'(y_l^o) u_j^h = \frac{1}{n_o} \delta_i u_j^h \\
    \delta_i &= (u_l^o - t_l) f'(y_l^o) \\
    \Delta w_{lj}^{(i)} &= \frac{1}{n_o} \delta_i u_j^h \\
    w_{lj}^{(i+1)} &= w_{lj}^{(i)} - \eta \frac{1}{n_o} \delta_i u_j^h
\end{align*}
\]
Error Backpropagation (hidden-output)

- Updating of $w_{lj}$:
  
  $$w_{lj}^{(i+1)} = w_{lj}^{(i+1)} - \eta \frac{\partial E}{\partial w_{lj}}$$

  with
  
  $$\frac{\partial E}{\partial w_{lj}} = \frac{(u_l^o - t_i)}{n_o} f'(y_i^o)u_j^h = \frac{1}{n_o} \delta_i u_j^h$$

  $$w_{lj}^{(i+1)} = w_{lj}^{(i)} + \Delta w_{lj}^{(i)} + \alpha \Delta w_{lj}^{(i-1)}$$
Error Backpropagation (input-hidden)

- Updating of $w_{kj}$:

$$w_{kj}^{(i+1)} = w_{kj}^{(i)} - \eta \frac{\partial E}{\partial w_{kj}}$$

with

$$E = \frac{1}{2n_Pn_O} \sum_{p=1}^{n_p} \sum_{l=1}^{n_o} (u_{pl}^o - t_{pl})^2$$

- Without loss of generality, set $n_p = 1$

$$\frac{\partial E}{\partial w_{kl}} = \frac{1}{2n_O} \sum_{l=1}^{n_o} \frac{2(u_l^o - t_l)}{2n_o} \frac{\partial u_l^o}{\partial w_{kj}} = \frac{1}{n_o} \sum_{l=1}^{n_o} (u_l^o - t_l) \frac{\partial u_l^o}{\partial y_l^o} \frac{\partial y_l^o}{\partial u_j^h} \frac{\partial u_j^h}{\partial y_j^h} \frac{\partial y_j^h}{\partial w_{kj}} =$$

$$= \sum_{l=1}^{n_o} \frac{(u_l^o - t_l)}{n_o} f'(y_l^o) w_{jl} f'(y_j^h) u_k^i = u_k^i f'(y_j^h)$$
Error Backpropagation (hidden-input)

Similarly to the updating of the output-hidden weights,

Updating weight $w_{jk}$ (hidden-input connections)

$$
\Delta \bar{w}_{jk} = -\eta \frac{\partial E}{\partial \bar{w}_{jk}}
$$

\[\frac{\partial E}{\partial \bar{w}_{jk}} = \frac{1}{n_o} \sum_{l=1}^{n_o} \phi_i^+ (u_i^o - t_l) \frac{\partial u_i^o}{\partial y_l^o} \frac{\partial y_l^o}{\partial y_j^h} \frac{\partial y_j^h}{\partial u_j^h} \frac{\partial y_j^h}{\partial \bar{w}_{jk}}\]

\[= \frac{1}{n_o} \sum_{l=1}^{n_o} \phi_i^+ (u_i^o - t_l) f'(y_i^o) w_{lj} f'(y_j^h) u_k^i\]

\[= -\frac{1}{n_o} \delta_j u_k^i \quad \bar{\delta}_j = f'(y_j^h) \sum_{l=1}^{n_o} \delta_l w_{lj}\]

$$
\Delta \bar{w}_{jk} = \frac{1}{n_o} \eta \bar{\delta}_j u_k^i
$$

\[\Delta \bar{W}_{jk} (n) = \frac{1}{n_o} \eta \bar{\delta}_j u_j^i + \alpha \Delta \bar{W}_{jk} (n-1)\]

Learning coefficient  Momentum
Utilization of the Neural Network

After training:

- Synaptic weights fixed
- New input $\rightarrow$ retrieval of information in the weights $\rightarrow$ output

Capabilities:

- Nonlinearity of sigmoids $\rightarrow$ NN can learn nonlinear mappings
- Each node independent and relies only on local info (synapses)

Parallel processing and fault-tolerance
CONCLUSIONS [ANN]

**Advantages:**
- No physical/mathematical modelling efforts.
- Automatic parameters adjustment through a training phase based on available input/output data. Adjustments to obtain the best interpolation of the functional relation between input and output.

**Disadvantages:**
“black box” : difficulties in interpreting the underlying physical model.
ANN for fault diagnostics
Objective

- Build and train a neural network to classify different malfunctions in the plant component
Input/Output patterns

- **Input:** signal measurements
- **Output:** number (label) of the class of the failure
Application 1

Transient classification in a Feedwater System of a Boiling Water Reactor (BWR)
Boiling Water Reactor
Secondary System
Outputs

- Class 1: Leakage through the second high-pressure preheater
- Class 2: Leakage in the first high-pressure preheater to the drain tank
- Class 3: Leakage through the first high-pressure preheater drain back-up valve to the condenser
- Class 4: Leakage through high-pressure preheaters bypass valve
- Class 5: Leakage through the second high-pressure preheater drain back-up valve to the feedwater tank
### Variables inputs

<table>
<thead>
<tr>
<th>Variable</th>
<th>Signal</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Position level for control valve EA1</td>
<td>%</td>
</tr>
<tr>
<td>2</td>
<td>Position level for control valve EB1</td>
<td>%</td>
</tr>
<tr>
<td>3</td>
<td>Temperature drain before VB3</td>
<td>°C</td>
</tr>
<tr>
<td>4</td>
<td>Temperature feedwater after EA2 train A</td>
<td>°C</td>
</tr>
<tr>
<td>5</td>
<td>Temperature feedwater after EB2 train B</td>
<td>°C</td>
</tr>
<tr>
<td>6</td>
<td>Temperature drain 6 after VB1</td>
<td>°C</td>
</tr>
<tr>
<td>7</td>
<td>Temperature drain 5 after VB2</td>
<td>°C</td>
</tr>
<tr>
<td>8</td>
<td>Position level control valve before EA2</td>
<td>%</td>
</tr>
<tr>
<td>9</td>
<td>Position level control valve before EB2</td>
<td>%</td>
</tr>
<tr>
<td>10</td>
<td>Temperature feedwater before EB2 train B</td>
<td>°C</td>
</tr>
</tbody>
</table>
Measurement:
36 sampling instants in [80, 290]s, one each 6s.
**Input/Output patterns**

- **Input:** The class assignment is performed dynamically as a two-step time window of the measured signals shifts to the successive time $t+1$

- **Output:** number of the class to which the variables belong
A training set was constructed containing 8 transients for each class.

For a transient of a given class the 35 patterns used to train the network take the following form:

\[
\begin{array}{cccccccc}
  x_1(2) & x_1(1) & x_2(2) & x_2(1) & \ldots & x_{10}(2) & x_{10}(1) & \text{class} \\
  \vdots & \vdots & \vdots & \vdots & \ldots & \vdots & \vdots & \vdots \\
  x_1(t) & x_1(t-1) & x_2(t) & x_2(t-1) & \ldots & x_{10}(t) & x_{10}(t-1) & \text{class} \\
  x_1(t+1) & x_1(t) & x_2(t+1) & x_2(t) & \ldots & x_{10}(t+1) & x_{10}(t) & \text{class} \\
  \vdots & \vdots & \vdots & \vdots & \ldots & \vdots & \vdots & \vdots \\
  x_1(36) & x_1(35) & x_2(36) & x_2(35) & \ldots & x_{10}(36) & x_{10}(35) & \text{class} \\
\end{array}
\]
How to set the ANN architecture

- **Parameters to be set:**
  - Number of neurons in the hidden layer ($n_h$)
  - Learning coefficient
  - Momentum

- **Method:**
  - Divide the training data into:
    - A training set (it will be used to find the synapsis weight)
    - A validation set (it will be used to find the optimal value of the parameters)
  - Proceed by trial-and-error
Network architecture

- $N_h, \eta, \alpha$ optimized with grid optimization

- Optima values
  - $N_h : 17$
  - $\eta : 0.55$
  - $\alpha : 0.6$
Test results

Neural Network and True values. TEST DATA

input pattern

class

input pattern

class

input pattern

class

input pattern

class

input pattern

3a 3b 3c 3d 3e
- The real-valued outputs provided by the neural network closely approximate the class integer values since early times.
- As time progresses, the network output class assignment approaches more and more closely the true class integer.
New transients

- Response of the neural network performance desirable when a new transient is found
  - Identification that something happens
  - Classification as don’t know
New transients

- New class
  - Class 6: Steam line valve to the second high-pressure preheater closing

Class 6: Valve VA6
New transients

- Close to Class 2 but with a larger deviation
- Value provided by the neural network close to 0
- the large output error is due to the fact that some of the inputs which are fed into the network fall outside the ranges of the input values of the training transients.
Conclusions

- A neural network has been constructed and trained to classify different malfunctions in the secondary system of a boiling water reactor.
- The network uses as input patterns the information provided by signals that are monitored in the plant and provides as output an estimate of the class assignment.
- When a transient belonging to a different class is fed to the network, it is capable of detecting that the pattern does not belong to any of the classes for which it was trained.
ANN for fault prognostics
ANN for fault prognostics

- Possible approaches
  - Learning directly from data the component RUL
  - Modeling cumulative damage (health index) and then extrapolating out to a damage (health) threshold
ANN for fault prognostics

- Possible approaches
  - Learning directly from data the component RUL
  - Modeling cumulative damage (health index) and then extrapolating out to a damage (health) threshold
Direct RUL Prediction: The model

\[
\begin{align*}
S_1 & \rightarrow x_1(t) \rightarrow RUL \\
& \rightarrow x_1(t-1) \\
& \rightarrow \ldots \\
& \rightarrow x_1(t-m) \\
S_n & \rightarrow x_n(t) \rightarrow RUL \\
& \rightarrow x_n(t-1) \\
& \rightarrow \ldots \\
& \rightarrow x_n(t-m)
\end{align*}
\]
Direct RUL Prediction: The model

\[
\begin{align*}
S_1 & \rightarrow x_1(t) \\
\vdots & \\
S_n & \rightarrow x_n(t) \\
\end{align*}
\]

\[
\begin{align*}
& \quad \rightarrow \quad \rightarrow x_1(t-1) \\
& \quad \rightarrow \quad \rightarrow x_n(t-1) \\
\end{align*}
\]

Class

ANN
Information necessary to develop the model

- Signal measurements for a set of $N$ similar components from degradation onset to failure

Similar Component 1:
Data available

\[
\begin{bmatrix}
    x_1^{(1)}(1) & \ldots & x_n^{(1)}(1) \\
    \vdots & \ddots & \vdots \\
    x_1(t_f^{(1)}) & \ldots & x_n(t_f^{(1)})
\end{bmatrix}
\]
ANN training data

Input

\[
\begin{bmatrix}
  x_1(t) \ldots x_1(t-m) \ldots x_n(t) \ldots x_n(t-m)
\end{bmatrix}
\]

Output

\[
\begin{bmatrix}
  t_f^{(1)} - m \\
  t_f^{(1)} - m - 1 \\
  \vdots \\
  x_1^{(1)}(t_f^{(1)} - m - 1)
\end{bmatrix}
\]

\[
RUL
\]

Trajectory 1

\[
x_1(t)
\]

\[
x_1(t) \rightarrow t
\]
Information necessary to develop the model

- Signal measurements for a set of $N$ similar components from degradation onset to failure

\[
\begin{pmatrix}
  x_1^{(1)}(1) & \ldots & x_n^{(1)}(1) \\
  \vdots & \ddots & \vdots \\
  x_1(t_f^{(1)}) & \ldots & x_n(t_f^{(1)}) \\
\end{pmatrix}
\]

\[
\begin{pmatrix}
  x_1^{(N)}(1) & \ldots & x_n^{(N)}(1) \\
  \vdots & \ddots & \vdots \\
  x_1(t_f^{(N)}) & \ldots & x_n(t_f^{(N)}) \\
\end{pmatrix}
\]
### ANN training data

#### Input

<table>
<thead>
<tr>
<th>$x_1(t)$</th>
<th>...</th>
<th>$x_1(t - m)$</th>
<th>...</th>
<th>$x_n(t)$</th>
<th>...</th>
<th>$x_n(t - m)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_1^{(1)}(m)$</td>
<td>$x_1^{(1)}(1)$</td>
<td>$x_n^{(1)}(m)$</td>
<td>$x_n^{(1)}(1)$</td>
<td>$t_f^{(1)} - m$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$x_1^{(1)}(m + 1)$</td>
<td>$x_1^{(1)}(2)$</td>
<td>$x_n^{(1)}(m + 1)$</td>
<td>$x_n^{(1)}(2)$</td>
<td>$t_f^{(1)} - m - 1$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>$x_1^{(1)}(t_f^{(1)} - 1)$</td>
<td>$x_1^{(1)}(t_f^{(1)} - m - 1)$</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>1</td>
</tr>
</tbody>
</table>

#### Output

<table>
<thead>
<tr>
<th>$x_1^{(N)}(m)$</th>
<th>$x_1^{(N)}(1)$</th>
<th>$x_n^{(N)}(m)$</th>
<th>$x_n^{(N)}(1)$</th>
<th>$t_f^{(N)} - m$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_1^{(N)}(m + 1)$</td>
<td>$x_1^{(N)}(2)$</td>
<td>$x_n^{(N)}(m + 1)$</td>
<td>$x_n^{(N)}(2)$</td>
<td>$t_f^{(N)} - m - 1$</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>$x_1^{(N)}(t_f^{(N)} - 1)$</td>
<td>$x_1^{(N)}(t_f^{(N)} - m - 1)$</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
Where to study

- All slides
- Paper: A Review of Prognostics and Health Management Applications in Nuclear Power Plants by Jamie Coble, Pradeep Ramuhalli, Leonard Bond, J. Wesley Hines, and Belle Upadhyaya
- Document: Neural Network Modeling