Structural Health Monitoring Using Statistical Pattern Recognition

Supervised Learning Methods

Keith Worden and Graeme Manson

Presented by
Keith Worden

The Structural Health Monitoring Process

1. Operational evaluation
2. Data acquisition & networking
3. Feature selection & extraction
4. Probabilistic decision making
   • Neural Networks
   • Support Vector Machines
Outline

• Supervised learning
• Neural networks
• Classification and regression
• Damage location on an aircraft wing
• Support vector machines

Supervised Learning

• Algorithms from machine learning where class labels are available for data.
• Applicable to problems involving classification or regression.
• In SHM context, SL applies when data from damaged structures or systems are available.
• SL allows additional diagnostic capability to just detection.
• Main algorithms over the years have been based on Artificial Neural Networks.
• Development of discipline of statistical learning theory has generated new learners like Support Vector Machines.
Lecture Overview

- Pattern Processing
- Probability Theory
- Novelty Detection
- Classification
- Regression
- Neural Networks

Damage Identification Hierarchy

- Level One: Damage Detection
  - Novelty detection
  - discussed in previous lecture.

- Level Two: Damage Location
  - May be classification (damage area), or regression (damage coordinates) problem.

- Level Three: Damage Type
  - Will be classification (crack, delamination, corrosion)

- Level Four: Damage Level
  - May be classification (discrete level), or regression (continuous level) problem.

- Level Five: Prediction
  - Combines information from pattern recognition with physical models.
Biological Inspiration

- Conventional serial processing (von Neumann) computers are superior at certain tasks, the brain is far superior at others:
  - Image recognition
  - Speech recognition
  - Co-ordination
  - Data fusion
- The reason for this superiority is the massively parallel nature of the brain.

10 billion ($10^{10}$) neurons in the human brain, each connected to around 10,000 other neurons.

Each neuron receives electrochemical inputs from other neurons at the dendrites. If the sum of these electrical inputs is sufficiently powerful to activate the neuron (exceeds a threshold), it transmits an electrochemical signal along the axon, and passes this signal to the other neurons whose dendrites are attached.
Artificial Neuron

- McCulloch and Pitts modelled the behaviour of a single neuron in 1943. They called this mathematical model a perceptron.

- The McCulloch-Pitts (MCP) model is the simplest possible neural network model and its power is restricted by its simplicity.

- A set of inputs $x_i$ are multiplied by weights $w_i$ and the sum of these weighted inputs is calculated. If this weighted sum is greater than the neuron’s threshold $\beta$ then the neuron fires ($y = 1$). If not, the neuron does not fire ($y = 0$).

Artificial Neuron

- A single perceptron is capable of representing some simple Boolean functions whereby the input variables $x_i$ and the output variable $y$ can only take values 0 or 1.

- Consider the two input case:

$$y = 0 \text{ if } w_1x_1 + w_2x_2 \geq \beta \text{ else } y = 1 \text{ if } w_1x_1 + w_2x_2 < \beta$$

- Values can be found for $w_1$ and $w_2$ to allow representation of any linearly separable logic function:

<table>
<thead>
<tr>
<th></th>
<th>$w_1$</th>
<th>$w_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>AND</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>$w_1$</th>
<th>$w_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>OR</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
Artificial Neuron

• Not all Boolean functions are linearly separable e.g. consider the XOR function:

<table>
<thead>
<tr>
<th>XOR</th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

\[ y = 0 \text{ if } w_1 x_1 + w_2 x_2 \geq \beta \text{ else } y = 1 \text{ if } w_1 x_1 + w_2 x_2 < \beta \]

• Examining the four possible combinations of \( x_1 \) and \( x_2 \) gives:

\[
\begin{align*}
0 &\geq \beta \quad (x_1 = 0, x_2 = 0, y = 0) \\
0 &< \beta \quad (x_1 = 0, x_2 = 1, y = 1) \\
1 &< \beta \quad (x_1 = 1, x_2 = 0, y = 0) \\
1 &\geq \beta \quad (x_1 = 1, x_2 = 1, y = 0)
\end{align*}
\]

for which no values of \( w_1 \) and \( w_2 \) exist.

Feedforward Networks

• Although the applicability of a single perceptron is limited, by arranging many perceptrons in various configurations and applying training mechanisms, one can actually perform tasks that are hard to implement using conventional serial processing approaches.

• For the problems of classification and regression encountered in health monitoring, the most commonly used architecture is the feedforward network.

• Data progresses from input layer through to an output layer via network connections. Single or several layers of connections can be used.
Multi-Layer Perceptron Networks

- Most common type of neural network is the Multi-Layer Perceptron (MLP) network.
- Many layers of weights can be used but network with two layers of weights is capable of *universal approximation* in sense that one can approximate any continuous function to arbitrary accuracy.
- Network weights are set during supervised learning procedure using a training data set with a corresponding set of target vectors. Error between the actual network outputs and the target outputs is minimised using a technique known as backpropagation.

\[ a_j^{(1)} = \sum_{i=1}^{d} w_{ji}^{(1)} x_i + b_j^{(1)} \quad j = 1, \ldots, M \]
\[ z_j = \tanh(a_j^{(1)}) \quad j = 1, \ldots, M \]

\[ a_k^{(2)} = \sum_{i=1}^{M} w_{kj}^{(2)} z_j + b_k^{(2)} \quad k = 1, \ldots, c \]
Multi-Layer Perceptron Networks V

\[ y_k = a_k^{(2)} \quad k = 1, \ldots, c \]

- **Linear activation function** (regression)

Multi-Layer Perceptron Networks VI

\[ y_k = \frac{\exp(a_k^{(2)})}{\sum_{l=1}^{c} \exp(a_l^{(2)})} \quad k = 1, \ldots, c \]

- **Softmax activation function** (classification)
Network Training

- In supervised learning we will have a set of input vectors \( \{ x_n \} \) where \( n = 1, \ldots, N \) together with a corresponding set of target vectors \( \{ t_n \} \).

- The aim of the neural network is to give the desired output for a given input.

- The goal is to find the values of the network weights, \( w \), which minimise the error function

\[
E(w) = \frac{1}{2} \sum_{n=1}^{N} \| y(x_n, w) - t_n \|^2
\]

- The network weights are randomly initialised and the error function is evaluated.

Network Training II

- The derivatives of the error function with respect to each weight are calculated and the weights are then adjusted accordingly to reduce the error. This process is repeated until some stopping criterion is satisfied.

- A validation data set is used to decide upon the best network parameters (number of hidden nodes, stopping time, etc.) and to avoid overtraining.

- A third data set, the testing set, is then passed through the network to assess its performance.
Example - Regression

- The problem consists of one input variable X and one target variable T with data generated by sampling X at equal intervals and then generating target data by computing SIN(2*PI*X) and adding Gaussian noise. A two-layer network with linear outputs is trained by minimising a sum-of-squares error function using the scaled conjugate gradient optimizer.

- Noisy sine wave

Example - Regression

- Create an MLP network with one input node, one output node and 3 hidden nodes. Train the network using the input and output data using scaled conjugate gradient (back propagation) to optimise the network weights and biases. Calculate network output for a new range of X test values and plot against data and true function.

- Network Prediction
Classification: The 1 of M Rule

- If we want to make contact with Bayesian classification (and we do because it’s optimal), we want to estimate *a posteriori* probabilities.
- The 1 of M training strategy can be proved to give this outcome (with some assumptions).

1. Choose same number of network outputs as classes.
2. During training, when data is presented at the input, set the output corresponding to the desired class to unity and the others to zero.
3. During recall, the network will respond at each output with the posterior probability!
4. Choose the highest output.

Example - Classification

- Aim of second demo is to show how an MLP can model a decision boundary.
- Figure shows the data sampled from a mixture of three Gaussian distributions, the first of which is labelled with blue circles and the other two are labelled red crosses.
- Bayesian decision boundary shown which gives a classification rate of 81.6%.

Training data
Example - Classification

- Create an MLP network with two input nodes (X and Y coordinates), two output nodes (Class 1 and Class 2) and 4 hidden nodes. Train the network using the input and output data using scaled conjugate gradient (back propagation) to optimise the network weights and biases.
• Experimental validation of SHM methodology based on novelty detection.
• Philosophy of programme was to develop methods which are robust enough to be successful on real aircraft structures.
• Starboard wing of trainer aircraft.
• Damage: Simulated by removal of inspection panels.

• Nine panels chosen. Percentage of wing area from 0.1% to 1.3%.
• Variability in panel fixing conditions likely to be an issue.
Data Capture

- Base measurements were transmissibilities - previously successful.
- Three transmissibility groups, each with central transducer. Transmissibility measurement associated with each panel.

Data Capture

- Wing excited using electrodynamic shaker on bottom surface of wing. White Gaussian excitation generated within acquisition system and amplified.
- DIFA Scadas acquisition system controlled by LMS software used to measure transmissibilities.
- 1024 spectral lines recorded for each transmissibility between 1024 Hz. and 2048 Hz.
Damage Location on an Aircraft

Data Capture

- 25 configurations considered - two removals of each panel interspersed with seven normal (undamaged) conditions. This allowed investigation of variability of the normal and damaged conditions between panel removals.

- One 16-average transmissibility recorded for feature selection followed by 100 single-shot transmissibilities. In total, 700 one-shot normal conditions and 200 one-shot measurements for each damage condition. Only magnitudes stored.

Feature Selection

- For most novelty detection methods, relatively low-dimensional features are required to construct a model of normality from the undamaged data.

- In this case, a feature is a region of the transmissibility which unambiguously separates normal condition from one, or more, of the ‘damage’ conditions.

- 16-average normal transmissibilities were visually compared, in turn, to the two 16-average transmissibilities for each damage condition.
Damage Location on an Aircraft

Novelty Detection

• Nine ‘best’ features chosen to form novelty detectors - one for damage in each panel.

Panel 3 Removed

Panel 7 Removed

• Performance of novelty detectors checked for all damage states. Each detector substantially ‘fired’ for only the panel for which it was intended.

Damage Location on an Aircraft

Sensors (acceleration data)

Pre-Processing (calculate transmissibility functions)

Feature Extraction (region of transmissibility function)

Pattern Recognition (novelty detection)

Decision (damaged or undamaged based on novelty index)

• Data to decision process for damage detection phase of work.
Damage Location on an Aircraft

Neural Network Classification

• Multi-Layer Perceptron (MLP) neural network used to map novelty indices obtained from the transmissibility features to the damage location.
• Data divided into training, validation and testing sets.
• Input and output layers had nine nodes. Training set used to establish network weights. Structure and training time etc. optimised by selecting conditions which gave lowest validation set error.
• Testing set gives quantification of success of network.
**Damage Location on an Aircraft**

**Neural Network Classification**

- Best network: 10 hidden units and 150,000 presentations of data. Validation error: 0.155. Training error: 0.158.

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>61</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>52</td>
<td>7</td>
<td>4</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>60</td>
<td>0</td>
<td>61</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>6</td>
<td>0</td>
<td>8</td>
<td>52</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>4</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>58</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>62</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>15</td>
<td>47</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Confusion Matrix

- Correct classification of 86.5% of patterns. Panels 3 and 6 due to weak feature. Panels 8 and 9 due to both detectors firing when either panel removed.

---

**Support Vector Machines (without Maths)**

- Developed in the context of *Statistical Learning Theory* (Vapnik).
- Basic principle is to control generalisation error in problems.
- SLT theory very complex and is largely concerned with producing bounds on generalisation error.
- Based on structural risk minimisation rather than empirical risk minimisation.
- Naturally trades model complexity against performance.
- Main algorithm associated with SLT is *Support Vector Machine*.
- Maximum margin classifier.
Support Vector Machines II

- Neural network seeks to minimise classification error.
- If two classes are separable, NN will establish an arbitrary decision boundary.
- This can generate testing errors when decision boundary is placed too close to one class.

Support Vector Machines III

- Support vector machine places decision boundary where it maximises distance to classes – maximum margin.
- Data points on margin are the support vectors.
- Generalisation error is controlled by number of support vectors algorithm can handle sparse data.
- Algorithm can also be applied to non-separable and nonlinearly separable classes.
Summary

- Supervised learning is the form of learning used when class labels for data are known.
- It is appropriate to higher levels of damage identification when examples of data from the damaged condition are known.
- Problems can be of classification or regression.
- Neural networks provide a powerful approach to both classification and regression.
- Support vector machines provide an alternative which may be attractive when data are sparse – as is often the case in SHM problems.

References

- C.M. Bishop. Neural Networks for Pattern Recognition, Oxford University Press, 1998.