Machine Learning
Industrial Vibration Monitoring
Demonstration

John Edwards
jedwards@numerix-dsp.com

DSP NUMERIX

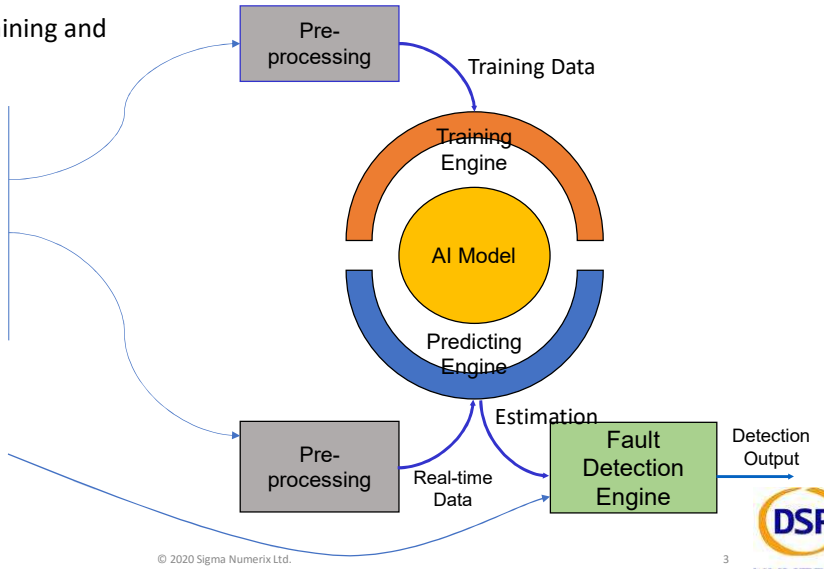
Agenda

- High level view of Machine Learning for vibration monitoring
- Demonstration of the trained network detecting vibration modes
- Architectural overview
- Downloading, compiling and using SigLib and the ML demonstration code

DSP NUMERIX

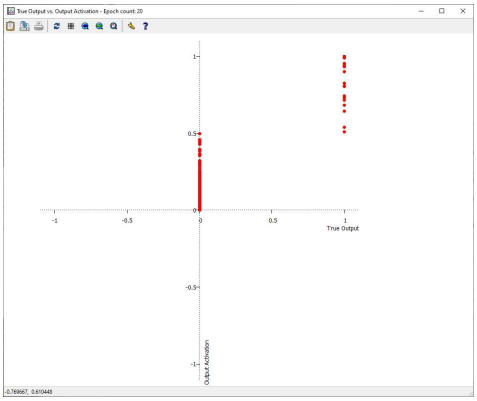
Machine Vibration Monitoring Application Overview

- Multiple stages to both training and inferring
  - Signal preparation
    - Filtering and noise removal
    - Level normalization
  - Feature engineering
    - Spectrum analysis
    - Cepstrum analysis
    - Peak detection
    - Zero crossing detection
  - Feature recognition
    - Event characterization
    - Fault characterization



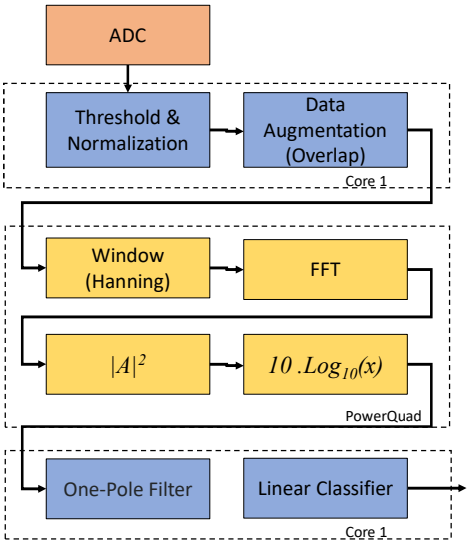
Let's Start With A Demonstration Then We'll See How It Works …

Scatter Diagram For The Demonstration System



Real-time demonstration on a Raspberry Pi

Application Flow Diagram

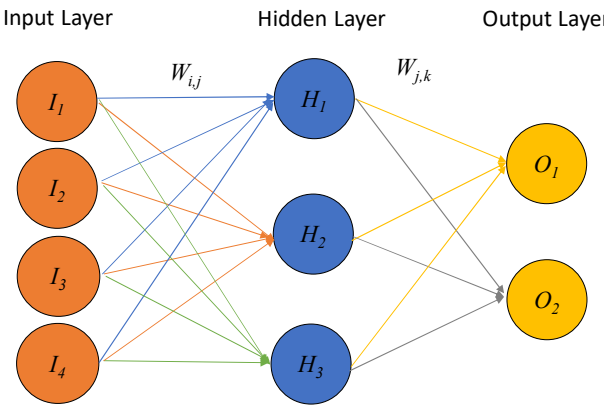


- Threshold & Normalization
  - Do not classify if below energy level threshold
- Data Augmentation
  - Duplication of source data, with overlap
  - Useful when reduced length training dataset available
- One-Pole Filter
  - Element wise low-pass filtering of frequency bins
  - Averages out the data going to the classifier
- Linear Classifier
  - With dot-product kernels on the PowerQuad

DSP NUMERIX

5

Multi-Layer Backpropagation Perceptron Network Solution



- Convolutional Neural Network
  - The cross-multiplications (coloured) are standard DSP dot-product operations

- Sample Rate
  - 16 kHz
- Input Layer Length (FFT Length /2)
  - 128
    - FFT Length = 256
    - We are just using the baseband magnitude
- Hidden Layer Length
  - 25
- Model Size
  - Using 8 bit quantized node weights
  - 2k bytes + 500 bytes per additional output category
  - Requires about 2 mins of audio, for each category, to train the model
    - Multiple epochs also helps (typically ≈5)
- Model Accuracy
  - >99% *
    - * Depends on the similarity (cross-correlation) between the signals

DSP NUMERIX

6

Building And Using The ML Example - I

- Download the free evaluation version of SigLib from: https://numerix-dsp.com/eval/
  - This includes the full source code for all of the examples used here
- Extract the .zip file into your installation folder e.g.: ~/siglib
- Set the environment variables for your compiler
  - Example setup scripts are available in the root SigLib folder
  - More details are available in the SigLib introduction video here: https://numerix-dsp.com/siglib/siglib_introduction.html
- The Machine Learning training and validation examples are in the folder:
  - siglib/Examples/CExamples/MachineLearning
- The training audio files are in:
  - siglib/Examples/CExamples/MachineLearning/Original
  - The training files for this example are:
    - MotorMode1.wav
    - MotorMode2.wav

Building And Using The ML Example - II

- Open a command prompt in the ML example folder:
  - siglib/Examples/CExamples/MachineLearning
- Execute the scripts to build and run the training and validation programs:
  - Under Windows:
    - mbr_single_category_motorMode.bat
  - Under Linux:
    - ./gbr_single_category_motorMode.sh
- The trained files generated are placed in:
  - siglib/Examples/CExamples/MachineLearning/weightCoefficientsFiles
- The trained header that is created is:
  - weightsFloatMotorMode1_MotorMode2.h

Building And Using The ML Example - III

- The example application is “pa_ml.c” and is located in the folder:
  - siglib/Examples/PortAudioExamples
- Copy the file weightsFloatMotorMode1_MotorMode2.h (from Step II) into this folder
- To run the example, use the following commands:
  - Windows:
    - mbr pa_ml
  - Linux:
    - ./lbr.sh pa_ml
  - Raspberry Pi:
    - ./rbr.sh pa_ml
- Notes:
  - The difference between the Linux and Raspberry Pi builds is that the Raspberry Pi build includes code to drive an SPI LED array
  - If you do not have the SPI LED array then just use the standard Linux build script on your Raspberry Pi
  - This example will:
    - Use the default input audio device for your computer
    - Give best results when using a high quality microphone. It is not recommended to use a microphone built into a laptop although, depending on the laptop, this can give satisfactory results

NUMERIX
http://www.numerix-dsp.com
John Edwards BEng (Hons) CEng MIEEE
AI And DSP Consultant
jedwards@numerix-dsp.com
https://www.linkedin.com/in/johnedwards/

Thank You Very Much For Watching

Please Do Contact Me If You Have Any Questions