LMS TO DEEP LEARNING: HOW DSP ANALYSIS ADDS DEPTH TO LEARNING

Paul Gorday, Nurgün Erdöl, Hanqi Zhuang
Outline

• Motivation
• Pulse Amplitude Modulation
• Demodulator Architecture
• Neural Network Topology & Training
• Simulation Results
• Conclusions
Motivations

• Since its inception, the LMS technique has been tied to learning systems (ADALINE = ADAptive Linear Neuron*)

• LMS has been extended to nonlinear neural networks and deep learning systems through the backpropagation algorithm

• Analysis and understanding are still catching up

• For communications, simple demodulators provide a useful frame of reference for evaluating learning techniques

• Our goal in this paper was a fundamental starting point similar to communication textbooks

Pulse Amplitude Modulation

- Starting point for developing maximum a posteriori (MAP) and maximum likelihood (ML) classification strategies in most communications textbooks

\[ m(n) = \sum_{k} d_k p(n - kT) \]

- Nyquist pulse shapes (Root-Raised Cosine) → orthonormal basis expansion
- Gaussian filtered pulse → non-orthogonal basis expansion
Demodulation

Data can be recovered using basis analysis equations

\[ m(n) = \sum_k d_k p(n - kT) \quad d_k = \sum_n m(n) q(n - kT) \]

→ \( \{p(n-kT)\} \) and \( \{q(n-kT)\} \) must be biorthogonal sets

[Graphs showing Nyquist and Gaussian pulses]
Flexible FSK Learning Demodulator

**Demodulator Architecture**

- Present synchronized instances of the received signal to a neural network for classification

![Diagram of Demodulator Architecture]

- **Modulator**
  - $m(n)$
  - $x(n)$
- **Neural Network**
  - $z(n)$
- **Delay Line**
- **Decision**
- **Output Symbol**

![Graph of Modulation vs Sample Index]
Neural Network Training

• Supervised learning with backpropagation
• Cost function is mean-squared error
• Neural network outputs interpreted as posterior probabilities
• Batch-mode stochastic gradient descent (SGD)
  • 1000 instances per batch
  • 2,000 epochs
  • Learning rate, $\eta = 0.1$
• Training data included noise ($E_b/N_0 = 7$ dB, SER~0.001)
Elementary Feed-Forward Neural Network

- Single neuron with $tanh$ activation function
- Nonlinear version of matched filter (correlation detector)
Results – Feed-Forward Network

- Nyquist pulse shape: FFNN learned orthogonal (matched filter) weights
- Gaussian-filtered pulse: network learned biorthogonal weights
Flexible FSK Learning Demodulator

Elementary Convolutional Neural Network

- Three-node convolutional layer with stride equal to symbol spacing
- All neurons use \textit{tanh} activation function
- Nonlinear version of matched filter plus zero-forcing equalizer

Diagram:
- Inputs: \( x \) (Delay Line Contents)
- Outputs: \( y \) (Network Output)
- Error: \( e \)
- Training Error: \( d \)
- Desired output: \{+/- 1\}
Results – Convolutional Network

- Gaussian-filtered pulse: CNN learned same solution as FFNN
- Performs slightly better than matched filter plus ZF equalizer
Elementary Recurrent Neural Network

- Feed back previous soft symbol decision
- Single neuron uses $tanh$ activation function
- Nonlinear version of matched filter plus decision-feedback equalizer

![Diagram of Elementary Recurrent Neural Network]

- Network Output $y$
- Training Error $e$
- Desired output $d$ {+/- 1}
Results – Recurrent Network

- Gaussian-filtered pulse: RNN learned matched filter shape near the feedback symbol, biorthogonal shape where no symbol information is available
- Performed slightly better than CNN and FFNN
Conclusions

- Communications: Elementary neural networks learned matched filter or simple equalizer solutions
- Signal processing: Training process drives system toward orthogonal/biorthogonal solution
- Data driven: Signal-to-noise level during training affects balance between noise and orthogonality
- Activation: $\tanh$ served mainly to dampen learning rate
- Cost function: MSE focuses on signal quality, cross-entropy focuses on decision probabilities (equivalent in this simple case)