Neural Nets for Adaptive Filter and Adaptive Pattern Recognition

Brian Young
btyoung@gmail.com
CSCE 636

10 February 2010

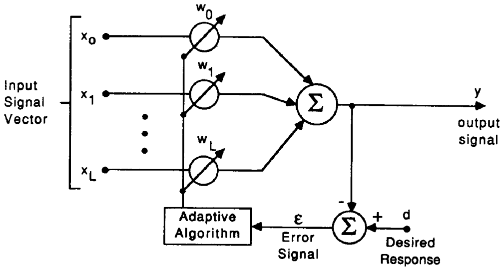
Outline

Article Context

- Published 1988 in IEEE Journals
- Bernard Widrow and Rodney Winter
  - Widrow was an EE professor at Stanford
  - Background in adaptive filtering and control
  - Developed the LMS Algorithm
- Specific algorithm isn't referenced often.

Adaptive Combiners and Filters

The Adaptive Linear Combiner (ALC)

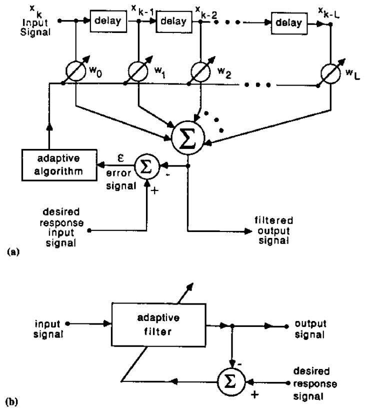


$$y = \mathbf{w}^T\mathbf{x}$$
$$\epsilon = d - y$$

- Weighted inputs are summed
- Adaptive algorithm works to minimize error ϵ
- Essentially an SLP with a single linear output

Adaptive Combiners and Filters

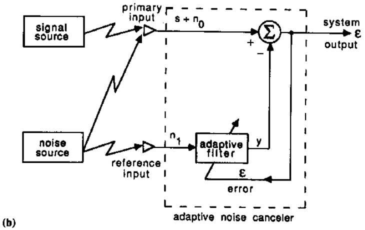
The Adaptive Filter (AF)



- Digitizes a single input to feed into an ALC
- Requires some knowledge of the required output
- In general this requires some data that can be correlated to the unknown output
- Well-used in industrial applications – not a 'toy' technique

Adaptive Combiners and Filters

Adaptive Filter example - Noise removal



- Adaptive filter attempts to minimize output noise from an external source
- Filter algorithm tries to minimize *total* output square magnitude
- Filter only has information on noise, so can only reduce the noise

$$
\begin{aligned}
E[\epsilon^2] &= E[s^2] + 2E[s(n_0 - y)] + E[(n_0 - y)^2] \\
&= E[s^2] + E[(n_0 - y)^2]
\end{aligned}
$$

- n_0 is related to n_1, but not necessarily the same
- The filter finds the the mapping from n_0 to n_1

Minimal Disturbance and the LMS Algorithm

Minimimal Disturbance Principle

- Consider a system with more variables than constraints
- Infinitely many solutions that can fulfill constraint
- How do you pick a solution?

Assuming that the current set of parameters is not too far from the solution, the best choice is the one that minimizes the change in the adaptive parameters.

Minimal Disturbance and the LMS Algorithm

Example - Smartphone accelerometer calibration

3 Sources of Error:

- Scaling error
- Offset error
- Non-orthogonality

$$\begin{bmatrix} \hat{x} \\ \hat{y} \\ \hat{z} \end{bmatrix} = \begin{bmatrix} S_{11} & S_{12} & S_{13} \\ 0 & S_{22} & S_{23} \\ 0 & 0 & S_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

Subject to:

$$f(\hat{\mathbf{x}}) = \hat{x}^2 + \hat{y}^2 + \hat{z}^2 - 1 = 0$$

Now we minimize

$$H = \sum \Delta S_{ij}^2 + \sum \Delta b_i^2 + \lambda f(\hat{\mathbf{x}} + \Delta \hat{\mathbf{x}})$$

Taking partials

$$\frac{\partial H}{\partial \Delta S_{ij}} = 2\Delta S_{ij} + 2\lambda x_i \left(\hat{x}_j + \Delta \hat{x}_j\right) = 0 \quad \rightarrow \quad \Delta S_{ij} \approx -\lambda x_i \hat{x}_j$$

$$\frac{\partial H}{\partial \Delta b_i} = 2\Delta b_i + 2\lambda \left(\hat{x}_i + \Delta \hat{x}_i\right) = 0 \quad \rightarrow \quad \Delta b_i \approx -\lambda \hat{x}_i$$

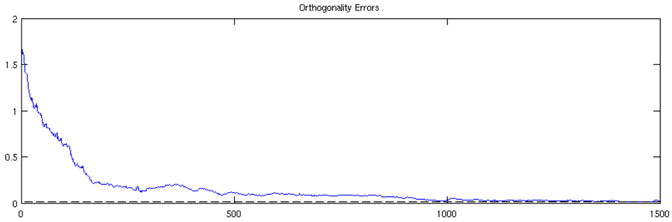
Minimal Disturbance and the LMS Algorithm

Example - Smartphone accelerometer calibration

Now solve for the λ that sets $f(\hat{\mathbf{x}} + \Delta\hat{\mathbf{x}})$ to zero:

$$\lambda^2 \Delta\hat{\mathbf{x}}^T \Delta\hat{\mathbf{x}} + 2\lambda\hat{\mathbf{x}}^T \Delta\hat{\mathbf{x}} + \hat{\mathbf{x}}^T \hat{\mathbf{x}} = 0$$

This is a quadratic equation and easily solved. Apply a multiplicative learning factor α and make changes:



Minimal Disturbance and LMS Algorithm

LMS Algorithm

Generalizing this for the system $\mathbf{y} = \mathbf{W}\mathbf{x}$, with teaching vector $\mathbf{d}$, we define a cost function

$$J = \sum \sum \Delta \mathbf{W}^2 + \boldsymbol{\epsilon}^T \boldsymbol{\epsilon}$$

$$\boldsymbol{\epsilon} = \mathbf{d} - (\mathbf{W} + \Delta \mathbf{W})\mathbf{x}$$

Taking the gradient WRT $\Delta \mathbf{W}$:

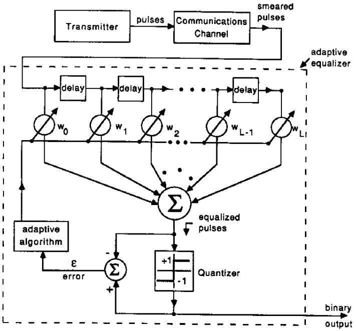
$$\frac{\partial J}{\partial \Delta \mathbf{W}} = 2\Delta \mathbf{W} + 2\boldsymbol{\epsilon}\mathbf{x}^T = 0$$

Applying a learning rate of α yields the *delta rule*:

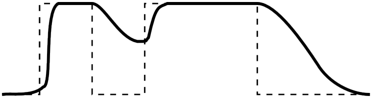
$$\Delta \mathbf{W} = -\alpha \boldsymbol{\epsilon} \mathbf{x}^T$$

Adalines

Adaptive Channel Equalizer



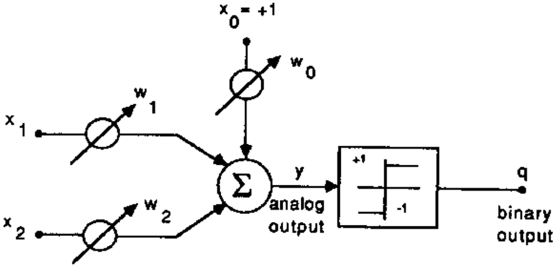
Sample Signal:



- Transmission lines ‘smear’ signals with an unknown impulse response
- Directly quantizing the sample signal would not yield the right result
- Placing a quantizer outside of an adaptive filter can improve the precision
- Quadruples data rate
- Looking even more like a neural net

Adalines

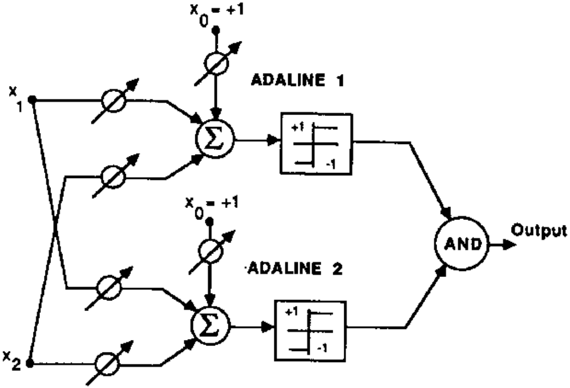
If we strip away all of the application specific components of the Adaptive Channel Equalizer, we get the Adaptive Linear Neuron (Adaline):



- Simply an ALC with a quantizer
- Note that errors off of y, not q
- Allows it to become "more" correct

Madalines

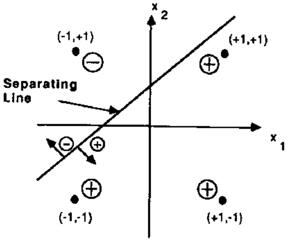
Combining multiple Adalines is a natural extension, creating a Madaline:

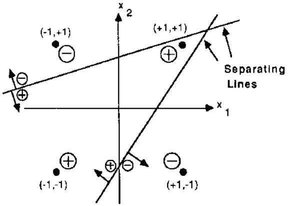


Note how the second layer is fixed

Madalines

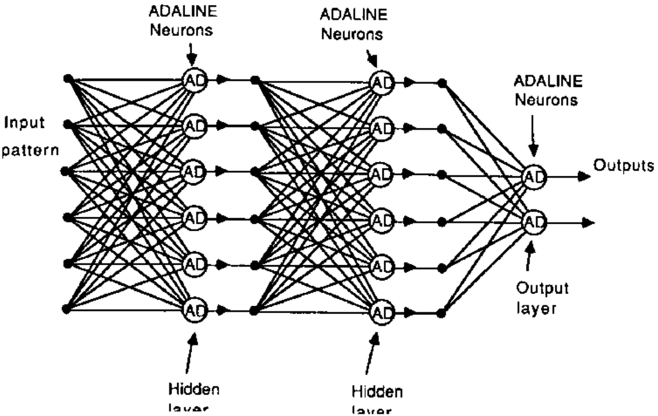
This creates multiple intersecting hyperplanes. In a 2-D boolean space:





Madalines

Not a far leap to an MLP



"Invented" a neural network without mentioning the brain!

The Hammer Principle

When you have a hammer...

- Here we have derived a multi-layer perceptron from an adaptive filter
- AFs are human-engineered solutions rather than a biological model
- However, this approach leads the author to over-engineer applications, as we will see

Also note that minimal disturbance is a new take on least-squares minimization.

Madaline Rule II

Another problem with backpropogation

- The backpropogation algorithm requires a differentiable threshold function.
- This can make digital implementation difficult
- Paper proposes a method that allows training of hidden units with an ideal step function

This already begs the question "Is this necessary" given that backpropogation has been well implemented and is the *de facto* standard.

Solution Algorithm

Methodology

1. Apply a teaching input with known desired output
2. In the first layer, select the neuron with the output closest to zero
3. Scale the input weights in the direction that will cause it to ‘flip’
4. Propogate changes forward
5. If change reduces error then proceed to next teaching input
6. Otherwise reverse changes and try the neuron next closest to zero
7. If all single neurons have been varied, begin modifying subsets of 2,3,... neurons

Solution Algorithm

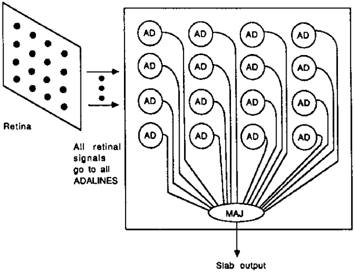
Critique

- Discrete and emperical rather than a rigorous mathematical method
- No evidence that it can be expressed as a least-squares minimization (or something similar)
- Ultimately inelegant compared to other training methods
- Are there cases where the sigmoid function is extremely inconvenient?

Example Problem

Slab proprocessor

The author demonstrates his training algorithm by developing a network that classifies input symbols without regard to rotation, translation or scaling.



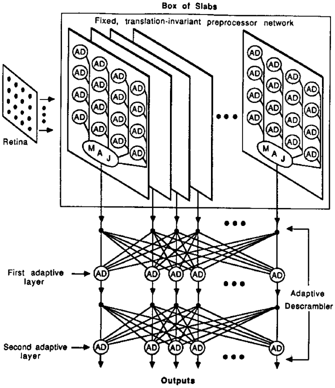
Done with slabs that:

- Are hard-wired connective networks
- Remove translation, rotation, etc.
- Each neuron in a slab is a convolution of the image with a pattern

- The convolution kernel is altered by the translation, rotation, etc. in all other neurons
- Thus majority votetaker will be the same regardless of change made
- However, this loses information about the actual pattern

Example Problem

Descrambling the Preprocessor Output



- Multiple slabs with random kernels will contain all information about the image, short of that specifically removed
- However, it will be 'scrambled' image – all of the information is there, but how do you interpret it?
- A fully adjustable 2-layer perceptron is used to descramble the inputs from the preprocessors
- This is trained using MRII
- These systems are stacked to handle the three individual tasks

Example Problem

Critique

- Maintains 12 layers, when typically only 3 are mathematically required to characterize any problem
- Forces an ungainly series of over-engineered structures rather than letting the training algorithm find internal representations
- An FFT would be a better way to handle translation and rotation – a straightforward MLP would likely generalize to a case that looks like an FFT
- Seems to be stuck in first-generation network thinking
- However, it does represent the modularity Hinton discussed
- Reduces validity of the confirmation of MRII rule since this is a very specialized case

Conclusions

- Neural nets can be derived as an array of adaptive filters
- This construction is functionally equivalent to other derivations
- This variation of perspective:
  - Can lead to new insights
  - Can limit creativity
  - Suggest multiple perspectives to avoid staid thinking
- LMS Algorithm based on Minimal Disturbance
  - Different approach than typical gradient descent
  - Ultimately shown to be another expression of least squares
- Specific Methodology not popularized
  - More discrete and inelegant than typical learning methods
  - However, later MRIII method shown to be functionally equivalent to backprop without explicit sigmoid