Approaches for Identifying & Characterizing Faults in Logic Simulation for Silicon Development Processes

Mubeen Khan (mkhan3), Prawal Shrestha (prawals)

Summary

Application of supervised learning in pre-silicon validation for fault characterization.

Device Under Test

Direct Digital Frequency Synthesizer

- Harmonically pure digital sine wave generator in fixed point q1.14 format.
- Use cases include variable frequency tone generation for modulation/demodulation functions in high speed baseband processors for mobile wireless devices.
- Advantages over conventional techniques (a) Extremely fast frequency switching (b) Capable of very fine output resolution increasing O(2^n) with the size of the internal phase accumulator (c) Continuity of phase when frequency is changed; minimizes transients & allows to control the phase of the generated signal.

Design Parameters

- Frequency Control W1 (slow varying input)
- Accumulator Control W2 (fast varying input)
- Coarse ROM configuration
- Fine ROM configuration

Output

- q1.14 fixed pt. Sine. Converted to [-1,1] real

Test Application

- High speed frequency modulation

Data Set

- Collection of suitable design matrix data from actual in-circuit testing that would lend itself well to our experimentation was a formidable challenge.
- For this study data has been collected from bit-exact cycle-exact simulation of the actual DUT.
- Design matrix and the corresponding output for the training data set was generated from a bit-exact cycle-exact hardware description language simulation.
- Fit-accuracy of the learned parameters was put to the test using unseen but realistic test data which was generated using a random simulation seed.

Models

Supervised Learning

- K-Means++ Clustering
  - Conceived at the outset as a mechanism for estimation it was instead found to be useful for frequency classification of the output samples
- Polynomial Regression & LWR
  - Both techniques were deployed but were found unsatisfactory for modelling the output of the circuit
- Neural Network
  - A NN with 2 hidden layers was able to estimate the training data with accuracy deemed sufficient for isolating basic faults
  - tanh was used as activation function for all layers
  - Output node generates real estimate, [-1,1], which is well suited for this activation function. Tanh catalyzes training by virtue of generating stronger gradients.
  - Dataset was conditioned for zero-mean & unit variance
  - Interpret loss from the model output for detection and classification of anomalies

Loss Function:

$$J = \left(\frac{1}{2m}\sum_{i=i}^{m}(\hat{y}-y)^2\right) + \lambda\left(||W^{[3]}||^2 + ||W^{[2]}||^2 + ||W^{[1]}||^2\right)$$

Back Propagation:

$$\delta^{[3]} = \nabla_{z^{[3]}} J = \frac{1}{2m}\sum_{i=1}^{m}\left(\nabla_{z^{[3]}}\left(\hat{y}^{(i)} - y^{(i)}\right)^2\right) = \frac{1}{m}\sum_{i=1}^{m}\left(\left(\hat{y}^{(i)} - y^{(i)}\right)\circ\left(a^{[3](i)}\right)'\right)$$

$$\delta^{[2]} = \left(W^{[3]T}\delta^{[3]}\right)\circ\left(a^{[2]}\right)' = \left(W^{[3]T}\delta^{[3]}\right)\circ\left(1-(a^{[2]})^2\right)$$

$$\delta^{[1]} = \left(W^{[2]T}\delta^{[2]}\right)\circ\left(a^{[1]}\right)' = \left(W^{[2]T}\delta^{[2]}\right)\circ\left(1-(a^{[1]})^2\right)$$

Forward Propagation:

$$\begin{array}{ll}
z^{[1]} = W^{[1]}x^{(i)} + b^{[1]} & W^{[1]} \in R^{K_1\times n}, \quad x^{(i)} \in R^{n\times 1}, \quad b^{[1]} \in R^{K_1\times 1} \\
a^{[1]} = f(z^{[1]}) & z^{[1]} \in R^{K_1\times 1} \\
z^{[2]} = W^{[2]}a^{[1]} + b^{[2]} & W^{[2]} \in R^{K_2\times K_1}, \quad a^{[1]} \in R^{K_1\times 1}, \quad b^{[2]} \in R^{K_2\times 1} \\
a^{[2]} = g(z^{[2]}) & z^{[2]} \in R^{K_2\times 1} \\
z^{[3]} = W^{[3]}a^{[2]} + b^{[3]} & W^{[3]} \in R^{1\times K_2}, \quad a^{[2]} \in R^{K_2\times 1}, \quad b^{[3]} \in R^{1\times 1} \\
a^{[3]} = h(z^{[3]}) & z^{[3]} \in R^{1\times 1} \\
\hat{y}^{(i)} = a^{[3]} & a^{[2]} \in R^{K\times 1}
\end{array}$$

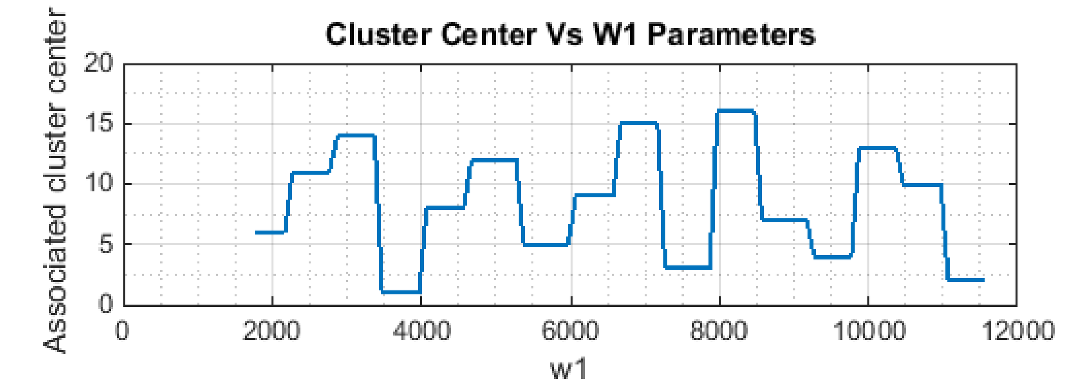
Future Work

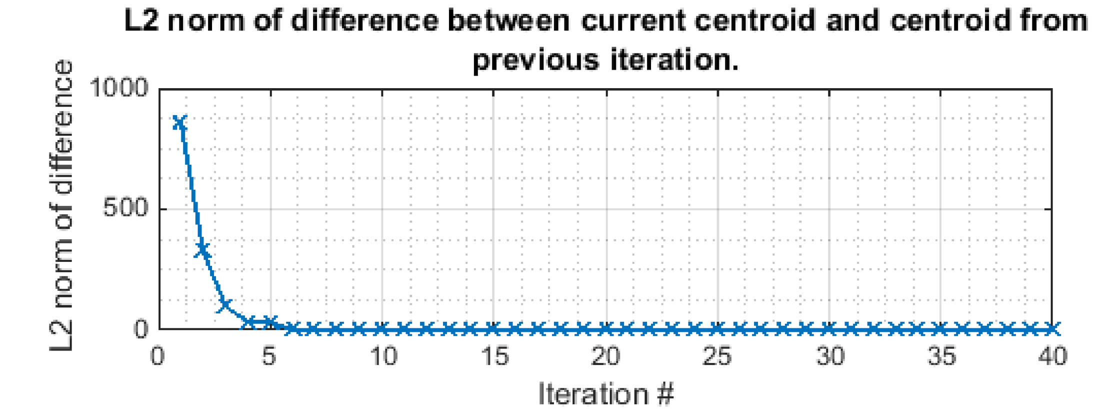
- Network Configuration Optimization – Prune nodes with insignificant weights for computational performance
- Fine tuning output estimation
- Error classification. Present capability identifies basic forms of error only. Recognize following distinct signatures
  - Classify phase truncation or acc overflow distortion
  - Distortion due to data compression in the ROM
  - Distortion due to finite precision of the look-up table

Results

K-Means++ Clustering

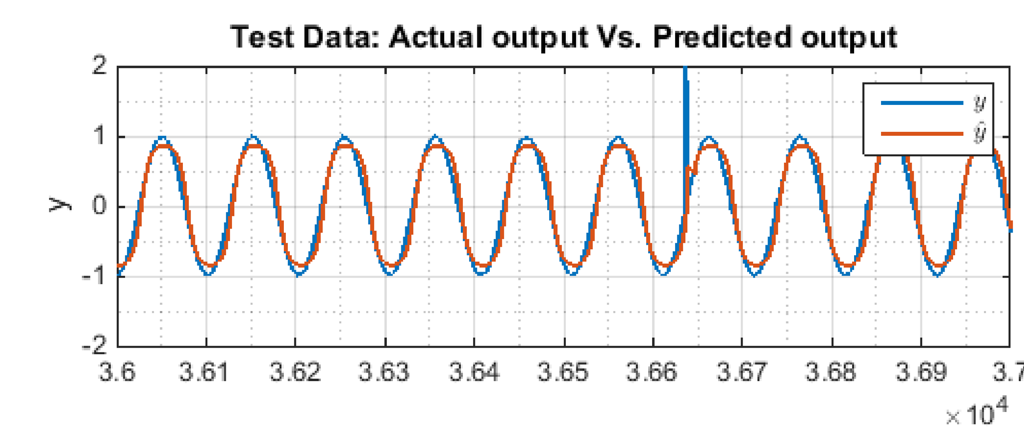
- Cluster center converged after around 5 iterations
- Cluster centers stay constant for continuous range of w1 values. Frequencies are grouped together and so model can be used for output frequency verification.

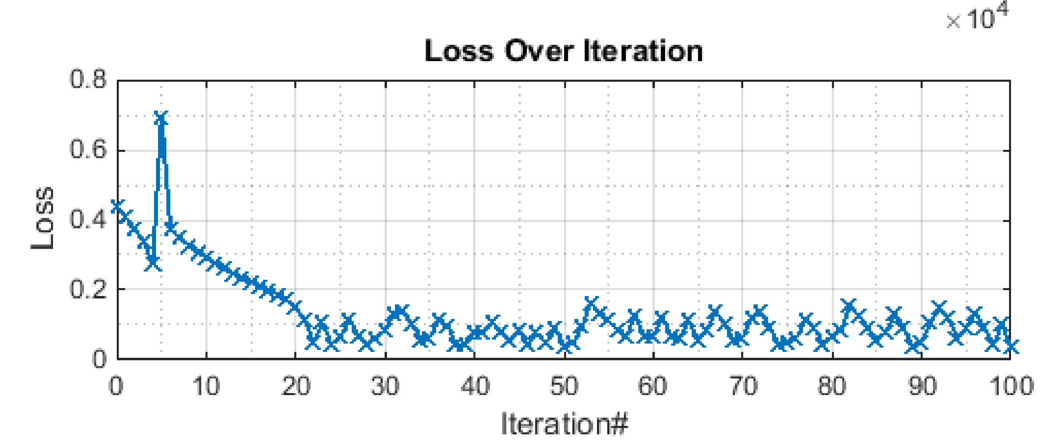




Neural Network

- NN converges after around 20 batch gradient updates.
- Model applied on test data was able to roughly predict the output.





References

[1] V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: A survey. ACM Computing Surveys, 41(3), 2009.
[2] A. DeOrio, D. S. Khudia, and V. Bertacco. Post-silicon bug diagnosis with inconsistent executions. In Proc. ICCAD, 2011.
[3] O. Guzey, L.-C. Wang, J. R. Levitt, , and H. Foster. Increasing the efficiency of simulation-based functional
[4] D. Josephson. Manic depression of u-proc debug. In Proc. ITC, 2002.
[5] S.-B. Park, A. Bracy, H. Wang, & S. Mitra. BLoG: Post-silicon bug localization in processors using bug localization graphs. DAC, 2010.
[6] D. Arthur & S. Vassilvitskii. K-means++ The Advantages of Careful Seeding.