ASIC IMPLEMENTATION OF A PRE-TRAINED NEURAL NETWORK FOR ECG FEATURE EXTRACTION

Huruy Tekle Tefai, Hani Saleh, Temesghen Tekeste, Mahmoud Alqutayri, Baker Mohammad
System On Chip Center
Khalifa University

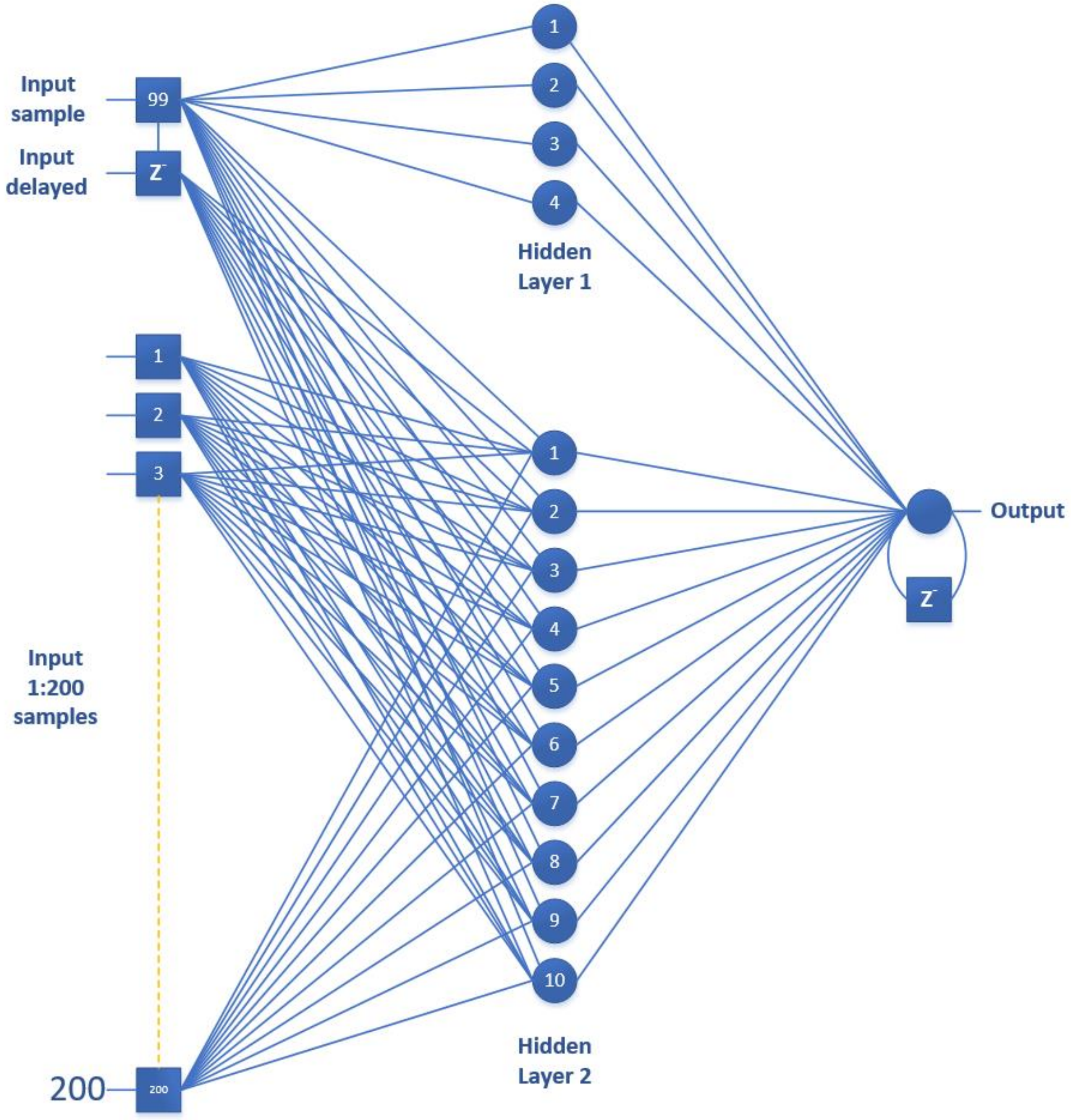
2020 IEEE International Symposium on Circuits and Systems
Virtual, October 10-21, 2020

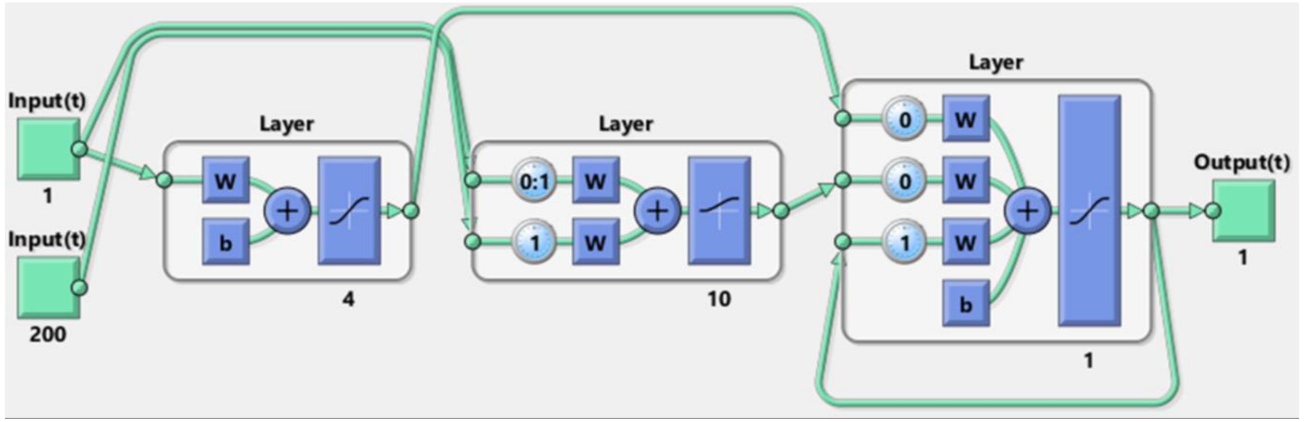
Background information and Motivation

- Success of ANN applications in ECG feature extraction
- It removes the need for hand-crafted features
- Deep Neural Network (1D CNN and LSTM) are common for 1 dimensional data
- It is growing in terms of accuracy and complexity
- The computation involved is heavy for deep networks
- Shallow networks are also capable of learning complex features similar to deep networks [1]
- Hardware implementation of QRS detection using shallow architecture neural network

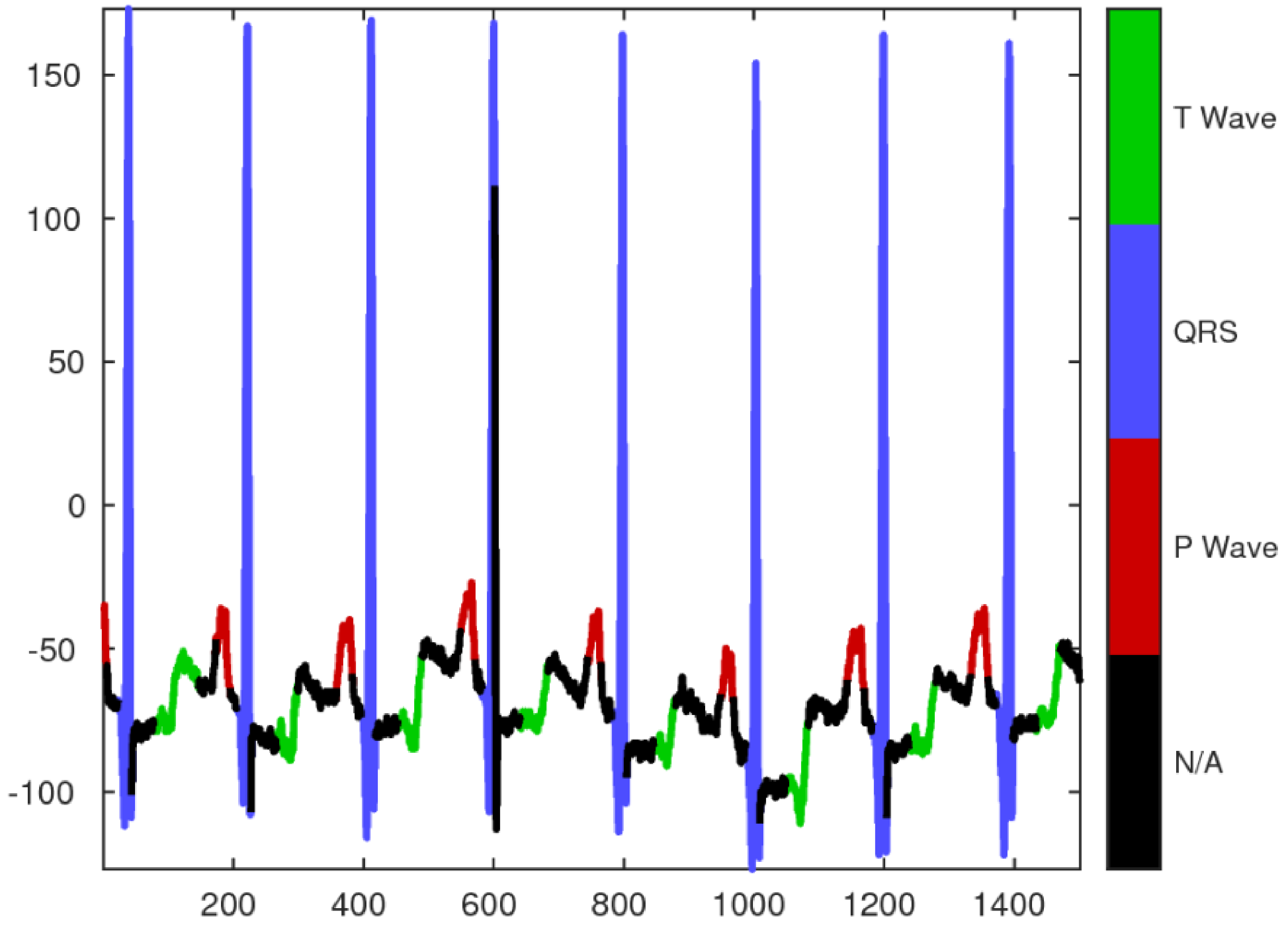
[1] Ba, Jimmy, and Rich Caruana. ”Do deep nets really need to be deep?.” Advances in neural information processing systems. 2014

Proposed neural network model





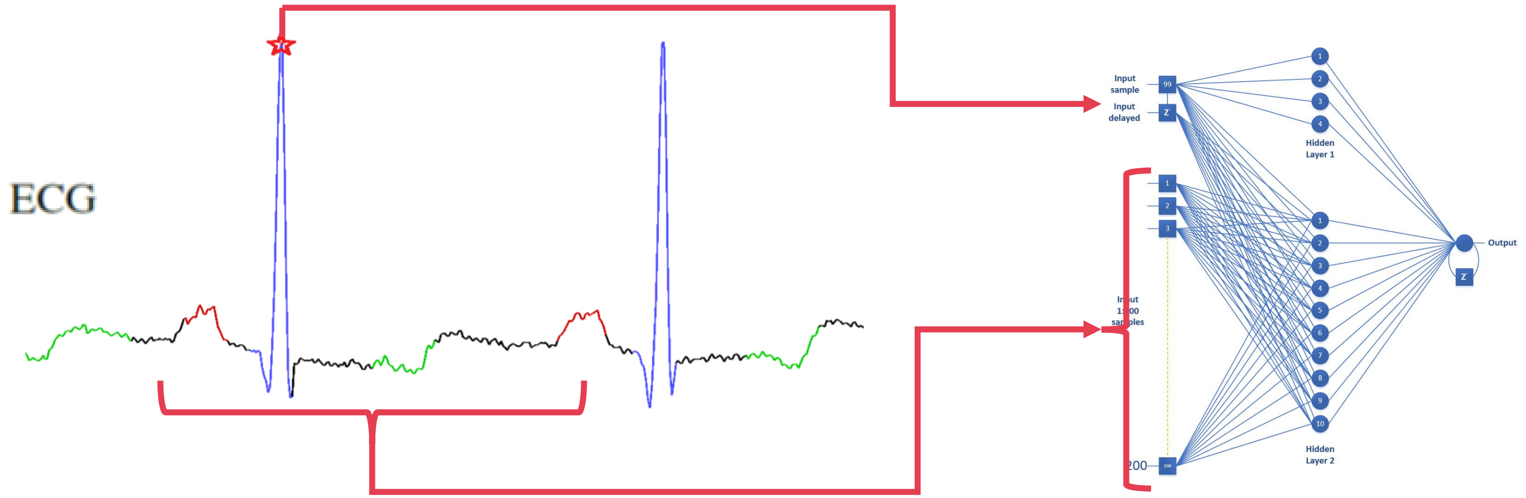
Dataset: PhysioNET labeled data

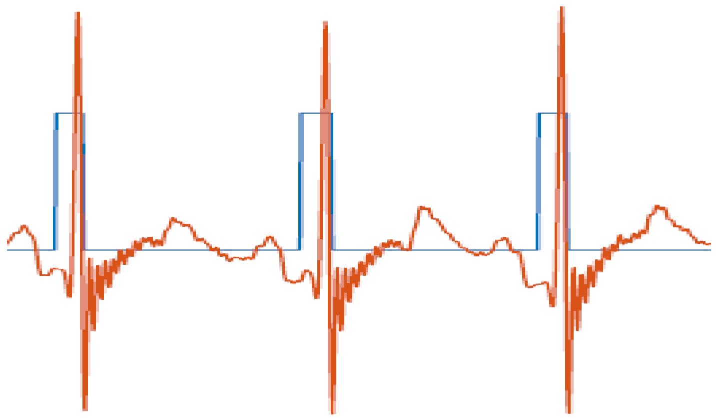


1	'n/a'	-79
2	'n/a'	-77
3	'n/a'	-79
4	'n/a'	-81
5	'n/a'	-80
6	'n/a'	-83
7	'QRS'	-84
8	'QRS'	-86
9	'QRS'	-82
10	'QRS'	-82

Dataset: Data Preparation

- **Input: A matrix of two vectors**



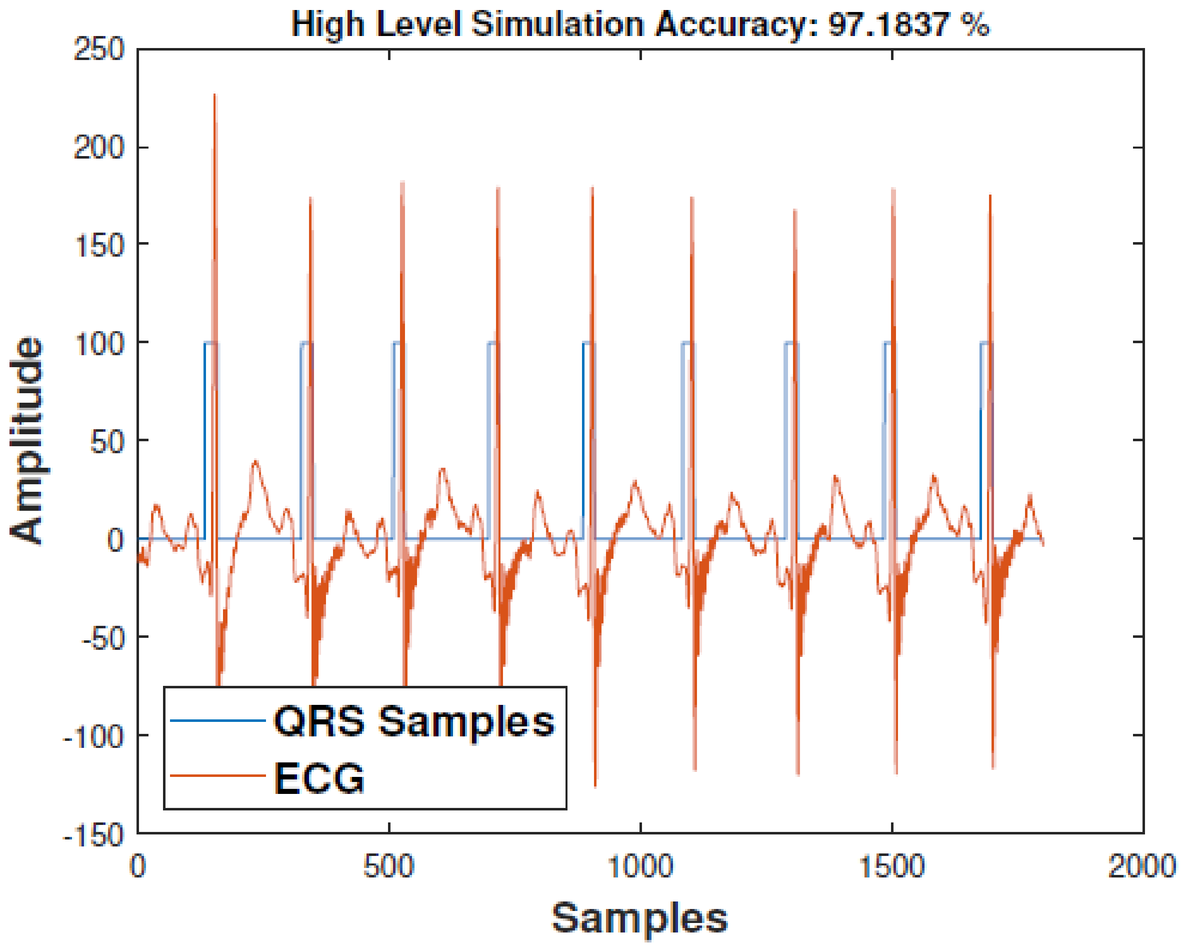


Label vector in blue

$$\begin{bmatrix} \mathrm{ECG(i)} \\ \begin{bmatrix} \mathrm{ECG(i\text{-}100)} & \ldots & \mathrm{ECG(i)} & \ldots & \mathrm{ECG(i+99)} \end{bmatrix} \end{bmatrix}$$

where *i* is mid sample index of the running window

High level accuracy



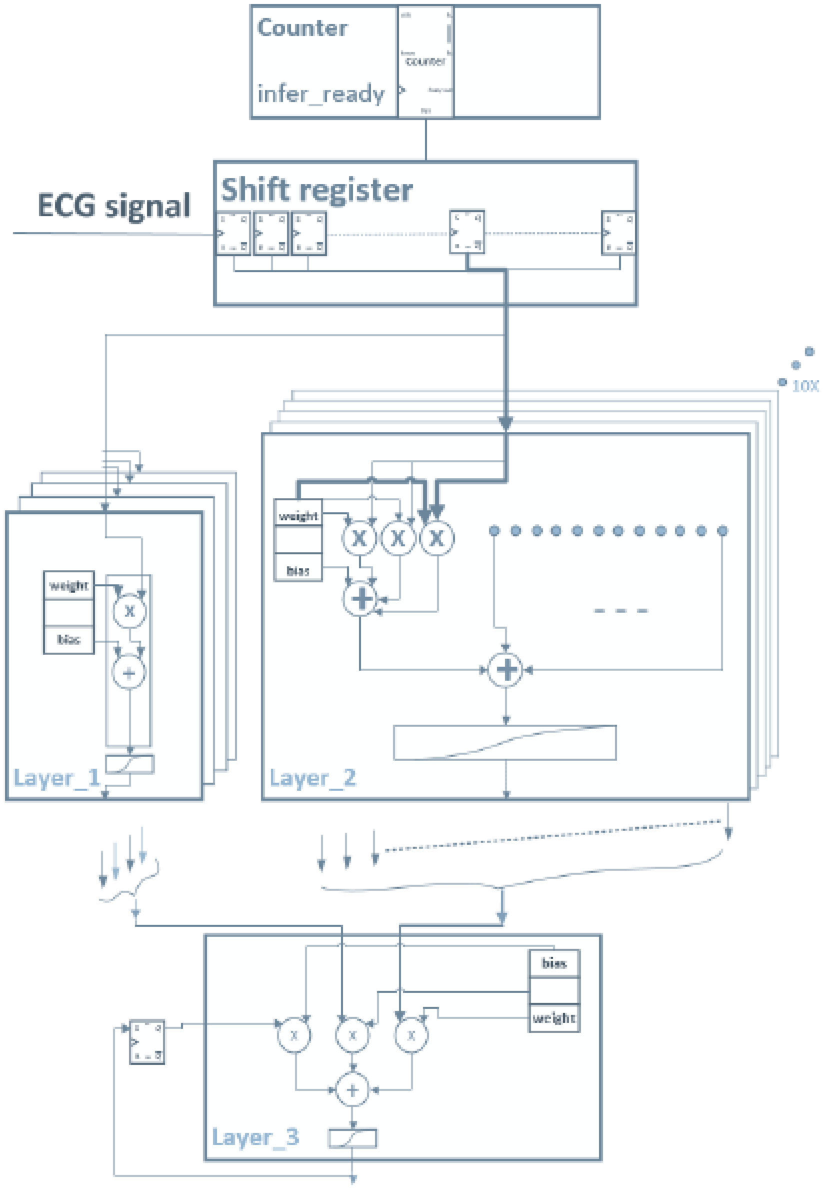
High level accuracy

- In addition to the QRS feature, the network was analyzed for P & T wave detection

Percentage	Train Data			Test Data		
	Acc	Sen	PPR	Acc	Sen	PPR
QRS	97.18	96.98	82.81	96.51	95.76	79.47
P wave	97.67	98.23	88.57	95.71	91.17	84.34
T wave	94.04	94.62	86.08	85.04	79.87	74.83

Hardware Implementation

- **RTL for the inference phase**
- **Use parameters obtained in training**
- **Convert parameters to a fixed point representation**
- **The design was simulated for functional verification**



No of parameters = 202x10 + (4+4) + 14 + 1+1 = **2044**

Category	*Value*
Internal Power (mW)	14.45
Net Switching Power (mW)	1.58
Leakage Power (W)	184.71
Total Power (mW)	**16.03**

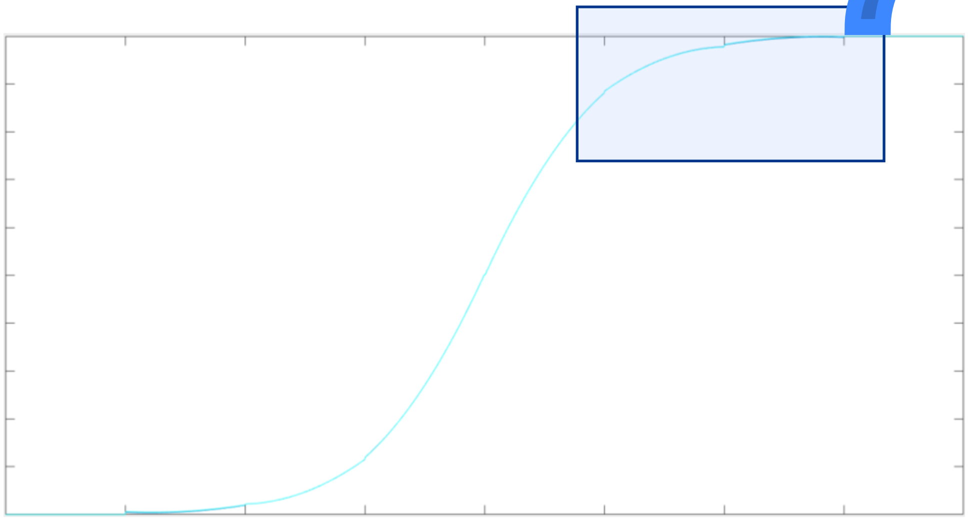
Hardware Implementation

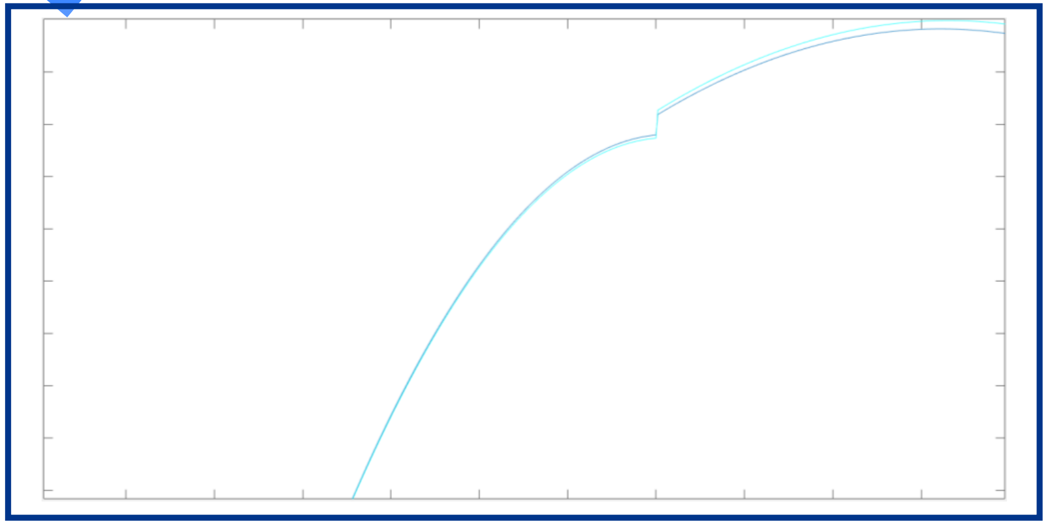
$$y = \frac{1}{1 + e^{-x}}$$

Sigmoid

$$y) = \begin{cases} 0, & \text{if } x \leq -6 \\ 0.1998 + 0.0703 * x + 0.0063 * x^2, & \text{if } -6 < x \leq -4 \\ 0.3803 + 0.1756 * x + 0.0215 * x^2, & \text{if } -4 < x \leq -2 \\ 0.5020 + 0.2727 * x + 0.0406 * x^2, & \text{if } -2 < x \leq 0 \\ 0.4982 + 0.2723 * x - 0.0404 * x^2, & \text{if } 0 < x \leq 2 \\ 0.6179 + 0.1768 * x - 0.0217 * x^2, & \text{if } 2 < x \leq 4 \\ 0.7968 + 0.0718 * x - 0.0064 * x^2, & \text{if } 4 < x \leq 6 \\ 0.9923 + 0.0021 * x - 0.0001 * x^2, & \text{if } 6 < x \leq 8 \\ 1, & \text{if } x > 8 \end{cases}$$

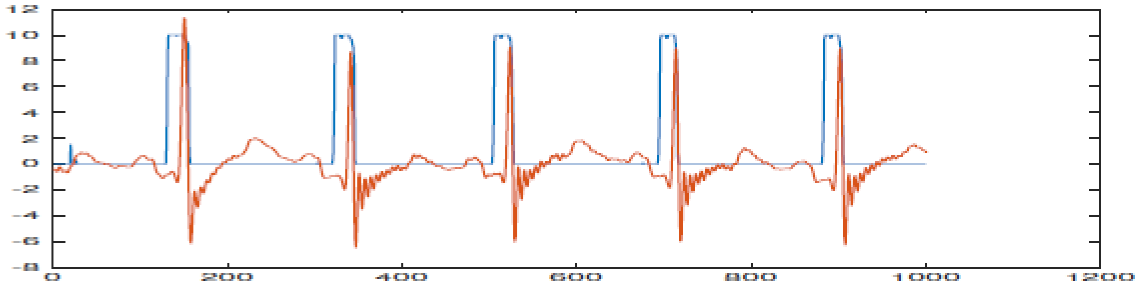
Polynomial approximation



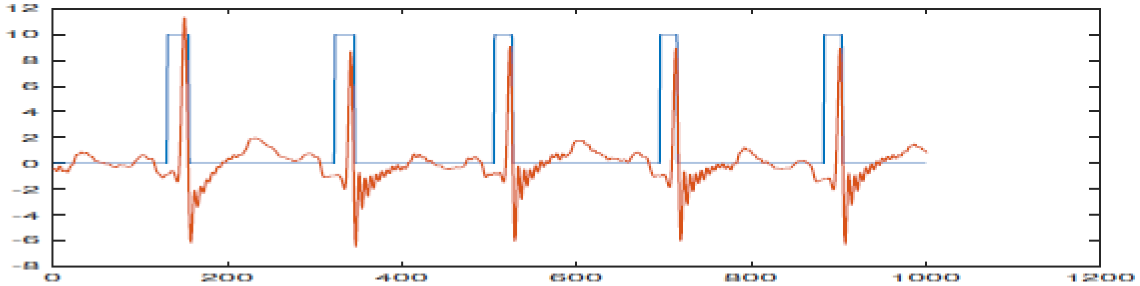


Simulation Result

- **Hardware Performance output before and after applying threshold**



(a) Before



(b) After

Conclusion

- **Shallow networks can**
  - Learn useful relationship between input and output
  - Achieve accuracy comparable to deeper networks

Thank You!