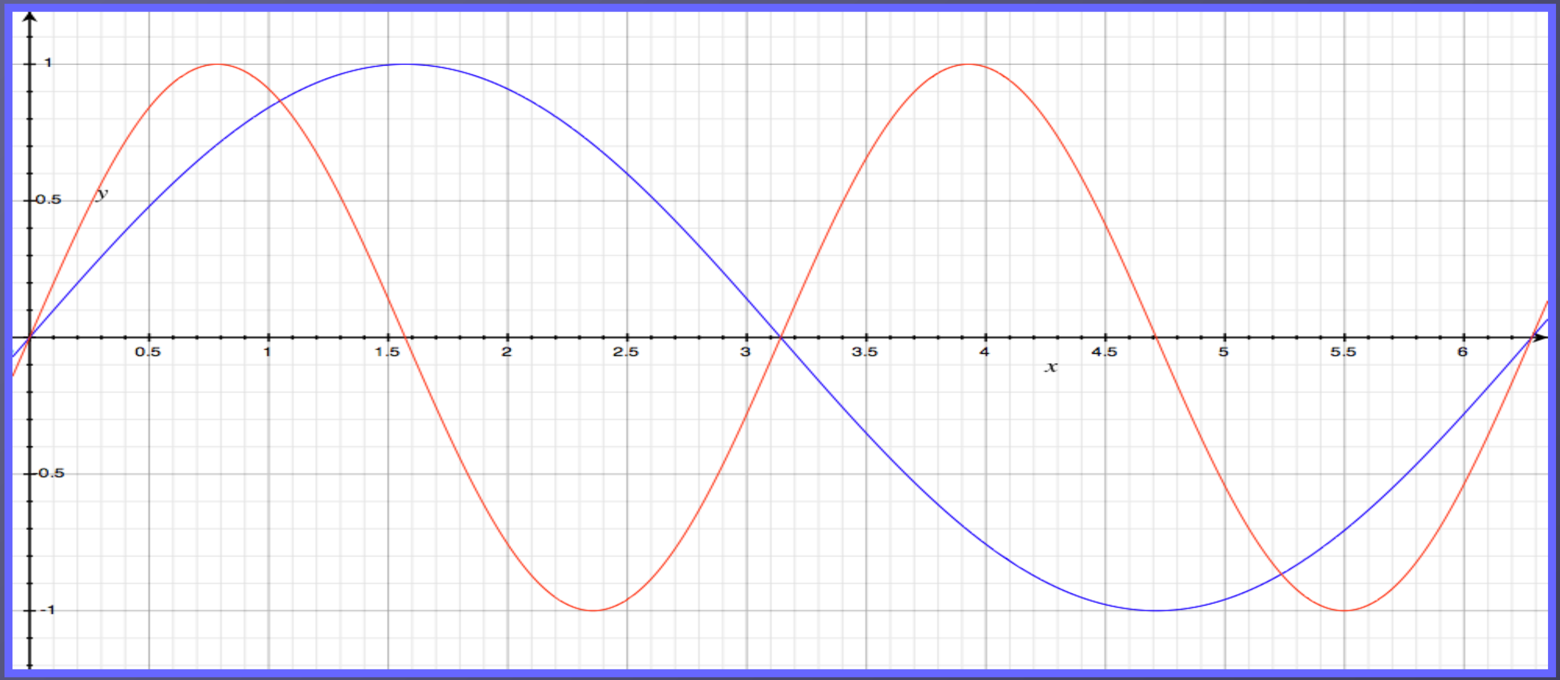
Motivation for using TDNN

- Feed forward
  - Information only flows one way
  - One input pattern produces (same) one output
  - No sense of time (or memory of previous state)
- Time delayed
  - Nodes represent previous states
  - Information flow is multidirectional
  - Sense of time and memory of previous state(s)

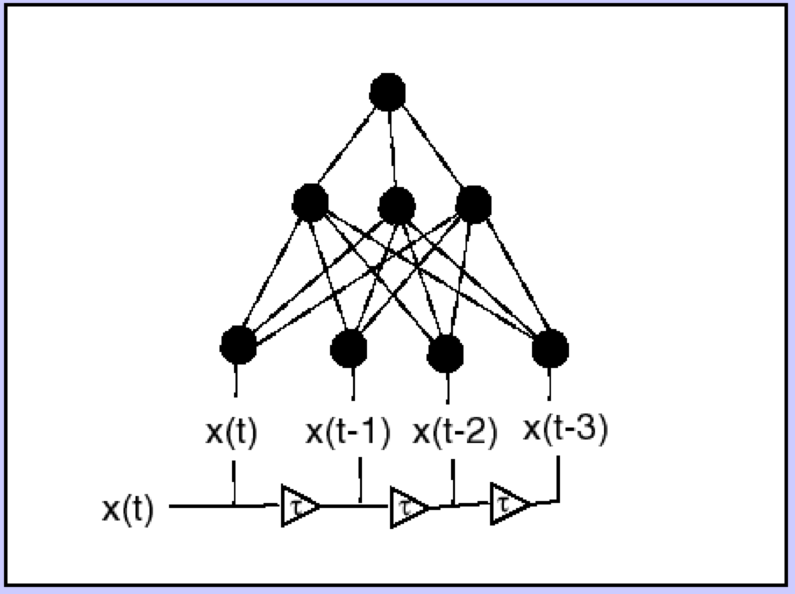
Example: Frequency Doubler Testbench

- Input signal: sin(x)
- Target (desired) output: sin(2x)
- We note that without memory, accurate input-output mapping would not be possible



Tapped Delay Lines / Time Delay Neural Networks (TDNN)

- One of the simplest ways of performing sequence recognition
  - Allows conventional backpropagation algorithms to be used
- Downsides: Memory is limited by length of tapped delay line
  - If a large number of input units are needed then computation can be slow and many examples are needed



TDNN (cont.)

- A simple extension to this is to allow non-uniform sampling

$$\bar{x}_i(t) = x(t - \omega_i)$$

where ω_i is the integer delay associated with component i.

- Another approach is for each "input" to really be a convolution of the original input sequence

$$\bar{x}_i(t) = \sum_{\tau=1}^{t} c_i(t-\tau)x(\tau)$$

In the case of the delay line memories:

$$c_i(t-\tau) = \begin{cases} 1 & \text{if } t = \omega_i \\ 0 & \text{else} \end{cases}$$

Learning time sequences

- There are many tasks that require learning a temporal sequence of events
- These problems can be broken into 3 distinct types of tasks
  - **Sequence Recognition:** Produce a particular output *pattern* when a specific input sequence is seen. Applications: speech recognition
  - **Sequence Reproduction:** Generate the rest of a sequence when the network sees only part of the sequence. Applications: Time series prediction (stock market, sun spots, etc)
  - **Temporal Association**: Produce a particular output *sequence* in response to a specific input sequence. Applications: speech generation