Perceptron

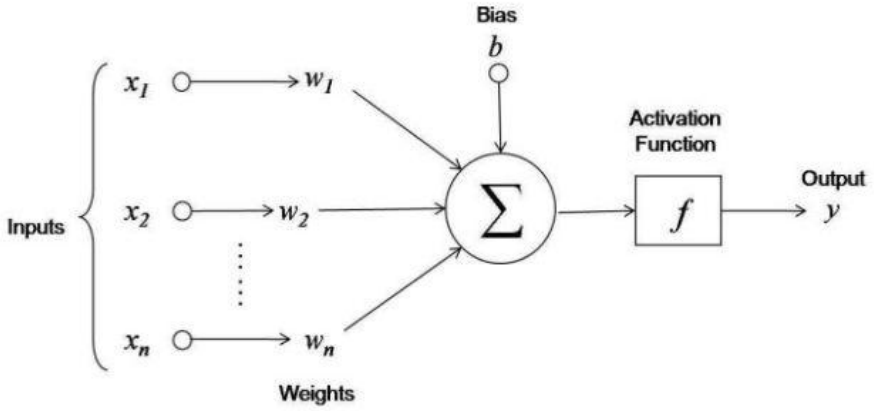
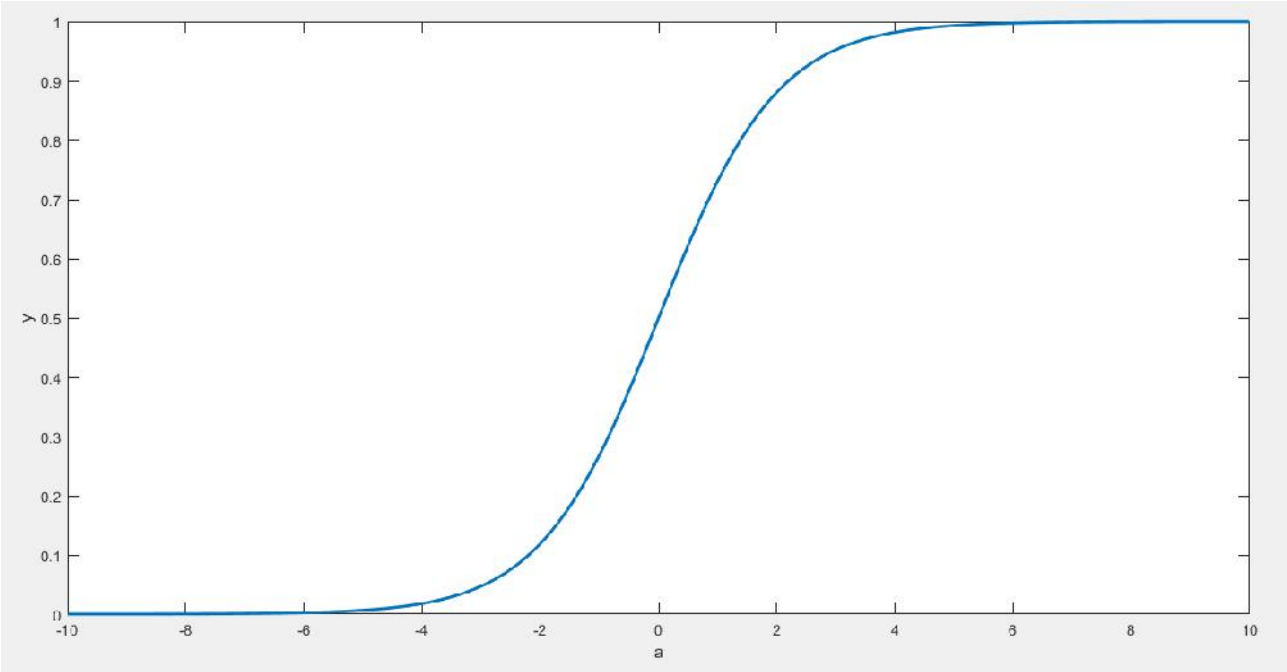


Image source: https://naadispeaks.wordpress.com/2017/11/08/artificial-neural-networks-with-net-in-azure-ml-studio/

Realize a network implementing a perceptron which takes $N_I = 10$ inputs $x_n$represented by using $b_x = 8$ bit. The network carries out the products between x_n and generic coefficients w_n and adds the result with a bias b. w_n and b are represented by using $b_w = 9$ bits.
x_n, $w_n$and b shall be considered in the range [-1,1] with the standard 2's complement representation.

The activation function of the perceptron shall be a **sigmoid**, shown in the next figure:



The output of the system shall be represented by using $b_o = 16$ bits.

Final report must presents:

- Introduction (algorithm description, possible applications, possible architectures, …)
- Architecture description (block diagram, I/O interface, etc.)
- VHDL code (with detailed comments)
- Test-plan and relevant Testbenches for the functional verification of the system
- Report the power consumption, the maximum clock frequency and the resource utilization on a Zync Xilinx FPGA. Explain possible logic synthesis warning messages
- Conclusion

Hint: Calling $a = b + \sum_{i=0}^{N_I - 1} w_n \cdot x_n$, a sigmoid produces an output $y = \frac{1}{1+e^{-a}}$.

Such function can be realized through dedicated architectures or through the standard Look-Up Table (lut) approach. Using the latter, you should fill such LUT by using the outputs which you expect for given inputs (for $a = 0$, the output shall be $y = 0.5$) . As requested, y shall be represented by using $b_o = 16$ bits using a positive representation. The LUT shall be designed to produce acceptable outputs for the full range of a. To do that, you shall calculate the combinations of b, w_n, x_n producing the highest positive and negative values of a and design the LUT in such range. How many outputs should you store?

Is there any symmetry of the sigmoid function that can be exploited to reduce the LUT size?