Design Method of Analog Sigmoid Function and its Approximate Derivative

Lylia Thiziri Chabane, Germain Pham, Paul Chollet, Patricia Desgreys

▶ **To cite this version:**

Lylia Thiziri Chabane, Germain Pham, Paul Chollet, Patricia Desgreys. Design Method of Analog Sigmoid Function and its Approximate Derivative. XXXVI Conference on Design of Circuits and Integrated Systems, Nov 2021, Vila do Conde (virtual), Portugal. hal-03331347

HAL Id: hal-03331347

https://telecom-paris.hal.science/hal-03331347

Submitted on 19 Jul 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Design Method of Analog Sigmoid Function and its Approximate Derivative

Lylia Thiziri Chabane, Dang-Kièn Germain Pham, Paul Chollet, Patricia Desgreys
LTCI, Télécom Paris, Institut Polytechnique de Paris, Palaiseau, France
lchabane@telecom-paris.fr

***Abstract*—In this paper, we propose to implement the sigmoid function, which will serve as an activation function of the neurons of a Multi Layer Perceptron (MLP) network, as well as its approximate derivative using an analog circuit. Several implementations have already been proposed in the literature, in particular, by Lu et al. (2000), which offers both a configurable and simple circuit realized in $1.2\,\mu\mathrm{m}$ technology. In this paper we demonstrate the circuit design of a sigmoid function based on Lu et al. using $65\,\mathrm{nm}$ technology in order to reduce energy consumption and circuit area. The design is based on an in-depth theoretical analysis of the circuit and validated by circuit level simulations. The main contributions of the paper are a modification of topology of the circuit in order to meet the required nonlinear response of the circuit and the extraction of the DC power consumption of the resulting circuit.**

***Index Terms*—Activation function, analog CMOS circuit, approximate derivative, backpropagation, multi-layer perceptron, sigmoid function.**

I. Introduction

In the current digital age, Artificial Intelligence (AI) spreads and impacts all areas of modern society. If AI has experienced such a revolution, it is thanks to the development of digital hardware. Today, Moore's Law, which predicts the improvement of digital microprocessors, is now facing the physical limits of matter. In addition, the Von-Neumann architecture, is characterized by a significant energy loss due to the flow of data between memory and processor, and is not the best solution for implementing neural networks such as mutli layer perceptrons (MLPs). Therefore, many researchers are moving towards the realization of neural networks implemented by means of analog circuits [1].

Different technologies have been considered to implement the behavior of the neuron with varying degrees of success. In [2], a memristor was added to the feedback loop of an operational amplifier to create a pseudo sigmoid function, e.g. a linear amplifier transfer function with bounded upper and lower voltage rails. Authors in [3] explain that the magnetic texture of spintronics such as domain walls and skyrmions, can implement *leaky integrate and fire* neurons.

In this paper, standard CMOS technology is preferred due to its maturity, low-cost and well-established fabrication processes. Therefore, the following review focuses on papers proposing implementations of sigmoid functions with CMOS circuits.

978-1-6654-2116-4/21/$31.00©2021 European Union

In 1993, Bogason [4] proposed a circuit based on two differential pairs in $2.4\,\mu\mathrm{m}$ CMOS technology driven by a digital `MODE` signal. Depending on the logic state of `MODE`, the circuit generates the sigmoid function or the derivative function. The drawback is therefore that it only generates one function at a time and requires additional driver circuitry. In 1994, Annema [5] proposed a circuit that generated both functions at the same time, which avoids the driving circuit. The proposed circuit is also based on a differential pair. However, the parameters of the function (gain factor and threshold) are fixed and cannot be changed. In 1997, Al-Ruwaihi [6] proposed a programmable sigmoidal activation function generator circuit which was manufactured by MOSIS in $2.0\,\mu\mathrm{m}$ CMOS technology. Changing a voltage was enough to vary the gain factor of the function, thus offering a wide choice of possibilities. However, the circuit suffers from significant nonlinear distortions.

In 2000, Lu, Shi and Chen [7] proposed a circuit using $1.2\,\mu\mathrm{m}$ technology which is configurable and able to generate both a sigmoid and an approximate derivative at the same time using CMOS technology. This circuit, thanks to its simplicity, offers a very low energy consumption which is the main feature to be improved to address future challenges of analog AI computing architectures. In addition, this simplicity will ease the design of large and complex networks. We propose here an improved version of this circuit realized in $65\,\mathrm{nm}$. This technology node is preferred here because it supports adequate supply voltage ($1.2\,\mathrm{V}$), which is more suited to analog processing. Our major contribution is to improve the circuit of [7] in order to be compatible with the chosen technology node. In addition, to the best of our knowledge, power consumption for such elementary circuit has not been discussed yet, which is an essential feature for the development of analog based neuromorphic systems. In this paper, we provide the DC power consumption obtained from circuit simulations.

In section II, we do a reminder on MLP as well as backpropagation, which emphasizes the need to implement the activation function as well as its derivative. In section III, we start by making an overall description of the circuit which is made up of two sub-circuits, then we focus on the two sub-circuits operation in more detail by explaining the modifications as well as the compromises made to obtain the desired signals at the output. Finally we present the simulation results obtained with Cadence Virtuoso ADE [8].

II. Multi Layer Perceptron and Backpropagation

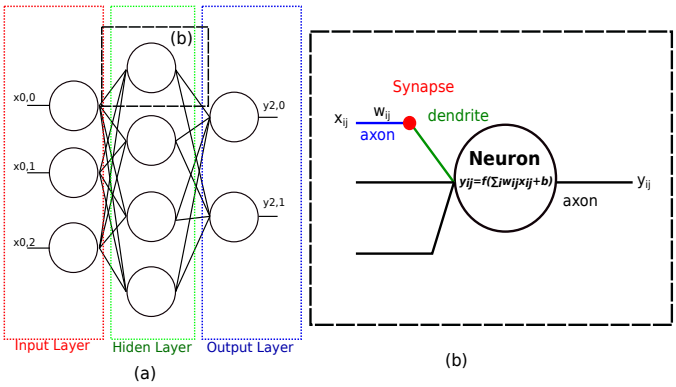

Fig. 1: Simple Multi Layer Perceptron (MLP) : (a) Example of a neural network with one hidden layer, (b) Connections to a neuron. x_{ij}, w_{ij}, $f(.)$ and b are the inputs of the neuron, the dendritic weights, the activation function, and the bias respectively

In the brain, the neuron is the basic unit of the nervous system. Neurons are interconnected with each other: their inputs, called *dendrites*, have a weighting role via the *synapses*; their output, known as *axon*, redistributes the *activity* level of the neuron to other neurons. In artificial systems, they are modeled as shown in Fig.1.(a). At the level of the neuron (Fig.1.(b)) the weighted sum resulting from the dendrites passes through the activation function which introduces "non-linearity". Various activation functions can be used and the most commonly used is the sigmoid function which is given by:

$$f(x) = \frac{1}{1 + \exp(-\alpha x + \Theta)} \tag{1}$$

where α is called the *gain factor* and Θ is called the *threshold*. In order to ensure the learning process of the MLP network, we use a *supervised* learning method by minimizing a quadratic error function defined as:

$$Error = \frac{1}{2}(y_{actual} - y_{target})^2 \tag{2}$$

$$y_{actual} = f\left(\sum_{ij} x_{ij} w_{ij}\right) \tag{3}$$

This objective function is used to tune the synaptic weights using a *gradient descent* based optimization method. This method, known as "backpropagation algorithm" consists in calculating the gradient of the error with respect to the weight that is to be tuned [9]. The weight update rule is given by:

$$w_{IJ} \leftarrow w_{IJ} - \frac{\partial Error}{\partial w_{IJ}} \tag{4}$$

with w_{IJ} the weight that is being updated. Recalling the *chain rule*, the gradient of the error can be expanded to:

$$\frac{\partial Error}{\partial w_{IJ}} = \frac{\partial Error}{\partial y_{calculated}} \times \frac{\partial y_{calculated}}{\partial \sum_{ij} x_{ij} w_{ij}} \times \frac{\partial \sum_{ij} x_{ij} w_{ij}}{\partial w_{IJ}} \tag{5}$$

$$\frac{\partial Error}{\partial w_{IJ}} = (y_{actual} - y_{target}) \times f'\left(\sum_{i} x_{ij} w_{ij}\right) \times x_{IJ} \tag{6}$$

It can be noted from Eq. (6) that, the value and the expression of the derivative of the activation function are required to calculate the weight update. The computation of these values in analog circuits is actually one of the major challenge to implement end-to-end analog MLP networks that include both inference and teaching phases with analog computation.

III. Circuit Description

A. Overall view

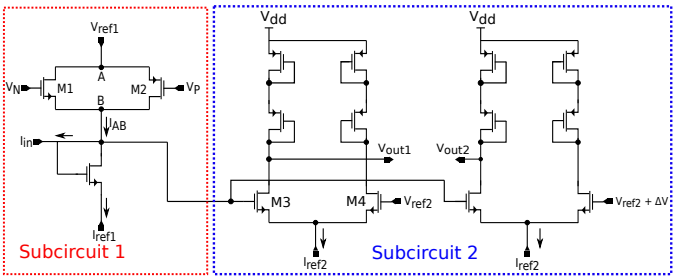

Fig. 2: Circuit schematic of the generator [7]

Fig. 2 shows the circuit of Lu et al. as published in [7]. The input of `subcircuit 1` is the current I_{in} which is also the input of the circuit in general. This current represents the weighted sum of the inputs of the neuron. The voltage V_B, the output of `subcircuit 1`, is a function of the currents I_{in} and I_{ref1}, and the voltage $V_N - V_P$. In [7], $V_N - V_P$ tunes the *gain factor* of the sigmoid function at the output V_{out1}. I_{ref1} tunes the horizontal position, previously mentionned as the *threshold*.

`Subcircuit 2` generates both the sigmoid function at the output V_{out1} and also its approximate derivative via the voltage difference $V_{out1} - V_{out2}$. `Subcircuit 2` is driven by the voltage V_B and outputs the two voltages V_{out1} and V_{out2}. In the following, we develop the design methodology used to migrate the initial circuit of [7] to 65 nm technology.

Note also that the *bulk* terminal of all the transistors is connected to the *source* terminal of the transistor in order to simplify the formulas.

B. Subcircuit1 : The Sigmoidal Biasing Circuit

As introduced earlier, `subcircuit 1` sets the voltage V_B which drives the input of a differential pair that produces a sigmoid function. This voltage carries several information such as the gain factor and the threshold of the activation function, which is modified by the voltage $V_N - V_P$ and the current I_{ref1} respectively, and obviously the value of the input signal representing the weighted sum coming from a synaptic network carried by the current I_{in}.

In section III-C, we show that the input of the circuit needs a sufficiently wide dynamic range to allow an appropriate response of the circuit. However, in order to guarantee proper operation of the circuit and obtain an equivalent resistance

which varies linearly with the voltage $V_N - V_P$ between the nodes A and B, the two transistors $M1$ and $M2$ must always remain in the linear region, which means that:

$$V_{AB} < V_N - V_B - V_{thn} \quad (7)$$
$$V_{AB} < V_A - V_P - |V_{thp}| \quad (8)$$

where V_{AB} represents the maximum voltage between the *drain* and the *source* nodes of the two transistors and V_{thn} and V_{thp} the *threshold voltages* of the NMOS and PMOS transistors respectively. Eq. (7) and (8) yield to the following conditions:

$$V_A + V_{thn} < V_N < V_{dd} \quad (9)$$
$$0 < V_P < V_B - |V_{thp}|. \quad (10)$$

Maximizing the dynamic range of V_{AB} requires to maximize V_A and to minimize V_B. Eq. (9) and (10) show that maximizing V_A and minimizing V_B, reduces the allowed dynamic range of V_N and V_P. The tradeoff here is to provide enough dynamic range for implementing a full sigmoid function, while leaving enough room to control the voltage difference $V_N - V_P$ which encodes the gain factor.

Now we compute the function that governs the voltage V_B. In the triode region, the currents of the circuit are [10]:

$$I_N = \beta_N \left((V_N - V_B - V_{thn}) V_{AB} - \frac{1}{2} V_{AB}^2 \right) \quad (11)$$
$$I_P = \beta_P \left((V_A - V_P - |V_{thp}|) V_{AB} - \frac{1}{2} V_{AB}^2 \right) \quad (12)$$
$$I_{AB} = I_N + I_P \quad (13)$$

with $I_{(N,P)}$ and $\beta_{(N,P)}$ the drain current and the transconductances parameter of each transistor. Transistors $M1$ and $M2$ are sized to have $\beta_N = \beta_P = \beta$ and the quadratic term is neglected in our derivation for sake of simplicity. Consequently, the circuit between nodes A and B is equivalent to a variable resistor R_{AB} given by:

$$R_{AB} = \frac{V_{AB}}{I_{AB}} \approx \frac{1}{\beta[(V_N - V_P) - (V_{thn} + |V_{thp}|)]} \quad (14)$$

R_{AB} is therefore expected to be essentially independent from V_{AB}. However, this is not exactly true due to the dependence of the threshold voltages to potentials V_A, V_B.

On the other hand, voltage V_B is given by:

$$V_B = V_{ref1} - V_{AB} \implies V_B = V_{ref1} - R_{AB} I_{AB} \quad (15)$$

which can be further developed as:

$$V_B \approx V_{ref1} - \frac{Iin + I_{ref1}}{\beta[(V_N - V_P) - (V_{thn} + |V_{thp}|)]} \quad (16)$$

Now, the dynamic range of V_B has to be analyzed in order to guarantee that the transistors always remain in the linear/ohmic zone. In our case, V_B varies between $450\,\mathrm{mV}$ and $750\,\mathrm{mV}$ with V_{ref1}= $750\,\mathrm{mV}$ so that $V_{AB} < 0.3\,\mathrm{V}$; otherwise, transistors will go into saturation. Further numerical results are provided in section III-D.

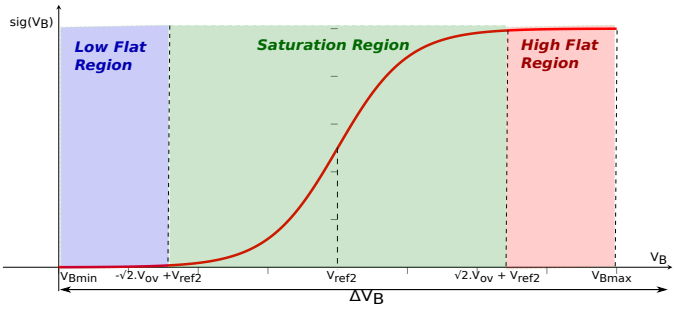

Fig. 3: The desired sigmoidal function with its three regions: Low Flat Region, Saturation Region, High Flat Region.

C. Subcircuit2 : Generator of sigmoidal function and its approximate derivative

Now, we analyze the effect of the driving voltage V_B on the differential pairs. The expected I/O characteristics of the differential pair is shown in Fig. 3. It implements a full sigmoid function that is defined over the voltage range of ΔV_B (recall that $V_B \in [0.45\,\mathrm{V}; 0.75\,\mathrm{V}]$). First, we set the voltage $V_{ref2} = 0.6\,\mathrm{V}$ which corresponds to the mean of V_B. Then, the "Voltage Amplification" A_V and the "Over-drive Voltage at equilibrium" V_{OV} are set so that: 50% of ΔV_B corresponds to the "Saturation Region", 25% for "Low Flat Region" and 25% "High Flat Region". ΔV, as shown in Figs. 2 and 4, is a small fixed voltage that shifts V_{out2} with respect to V_{out1}; as shown in [7], the difference $V_{out2} - V_{out1}$ provides an approximate of the sigmoid derivative. Because the range of V_B is quite

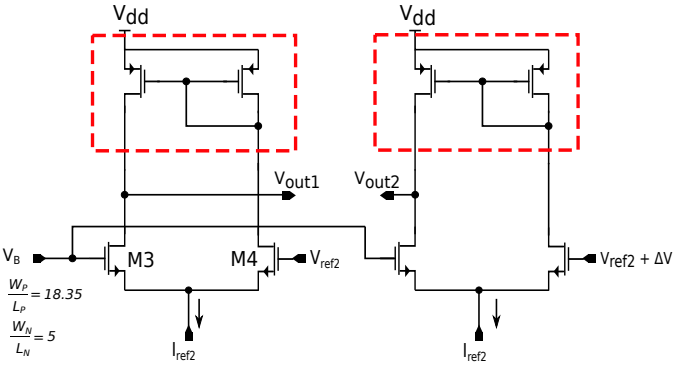

Fig. 4: The subcircuit2 "Generator of sigmoidal function and its approximate derivative"

small (about $0.3\,\mathrm{V}$ as mentionned earlier), the voltage gain of the differential pair must be maximized in order to ensure nonlinear output response. The circuit topology proposed by [7] is not able to provide a gain large enough to meet the targeted nonlinear response. This gain limitation is due to the supply voltage reduction (from $3.3\,\mathrm{V}$ to $1.2\,\mathrm{V}$).

The circuit is modified as shown in Fig. 4, by replacing the PMOS "diode connected" transistors with a "current mirror". Note that here also the *bulk* terminal is connected to the *source* terminal of the transistor. In this case, the "Voltage

Amplification" A_V is equal to [11]:

$$A_V = -gm_{Eq,N} \times (ro_N || ro_P) \tag{17}$$

ro_N and ro_P are the "ON" state resistances of the transistors and $gm_{Eq,N}$ is the transconductance of the NMOS transistor at equilibrium (e.g. in common mode) defined as [11]:

$$gm_{Eq,N} = \sqrt{\mu_N C_{OX} \frac{W_N}{L_N} I_{ref2}}. \tag{18}$$

It corresponds to the slope of the characteristics shown in Fig. 3 for $V_B = Vref2$. The "Overdrive Voltage at equilibrium" V_{OV} is given by:

$$V_{OV} = \sqrt{\frac{I_{ref2}}{\mu_N C_{OX} \frac{W_N}{L_N}}}. \tag{19}$$

V_{OV} defines the saturation region shown in Fig. 3 and is set according to the dynamic range of V_B. Therefore, the targeted nonlinear response is obtained by tuning the ratio $\frac{W_N}{L_N}$ and the current I_{ref2}.

D. Simulation Results

The proposed circuit was simulated with Virtuoso with Vdd =1.2 V, V_{ref1} = 0.75 V, 0.75 V < $V_N - V_P$ < 1.2 V, V_{ref2} = 0.60 V, ΔV = 10 mV and I_{ref2} = 10 µA. Fig. 5

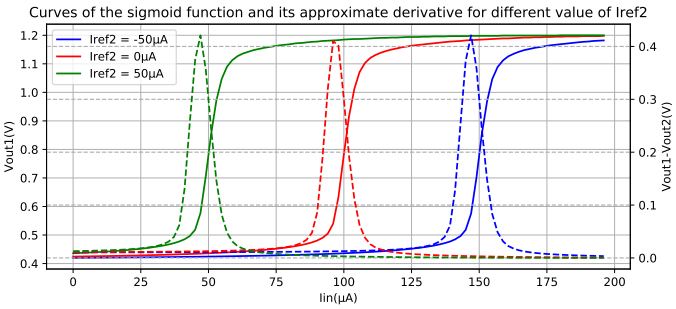

Fig. 5: Sigmoidal function (continuous line) and its approximate derivative (broken line) for different value of I_{ref1}

shows the obtained sigmoidal function and the approximate derivative as function of input current I_{in}. Also, the current I_{ref1} properly sets the threshold (horizontal shift) of the sigmoid. The sigmoid function obtained varies from 0.4 V to 1.2 V and is centered in 0.8 V, while the true sigmoid function varies from 0 to 1 and is centered at 0.5. For next work, we could consider a circuitry which would lower the voltage at the center of 0.8 V to $\frac{V_{dd}}{2}$ and would amplify our output signal to match the desired function.

According to [7], the variation of $V_N - V_P$ is supposed to modify the gain factor of the function. Fig. 6 shows the derivative curve calculated from the sigmoid functions and shows a variation of the maximum gain with respect to $V_N - V_P$. This variation is quite limited and is attributed to the small dynamic range of the equivalent resistance R_{AB}. For now, the required dynamic range of R_{AB} is still under analysis to validate this circuit feature. In addition, it should be noted that

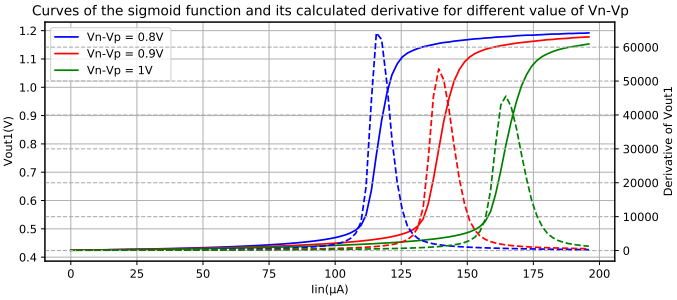

Fig. 6: Sigmoidal function (continuous line) and its calculated derivative (broken line) for different value of $V_N - V_P$

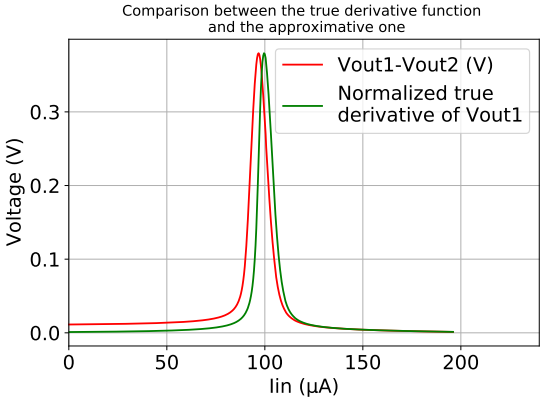

Fig. 7: Comparison between the true derivative function and the approximative one

the variation of $V_N - V_P$ causes an undesirable shift in the function.

Fig. 7 compares the approximate derivative obtained with the differential voltage $V_{out1} - V_{out2}$ and the true derivative of $Vout1$, computed in Virtuoso's *calculator* with the `deriv` function. The two functions are close with a slight horizontal shift of the approximate derivative compared to the true one. Future work will be to experiment the level of tolerance of the neural network to the error made on the derivative of the activation function at the time of training so that it can converge. The results of these experiments will allow us to better assess the quality and the required accuracy of the derivative to properly train a regular MLP and demonstrate the efficiency of this circuit for on-chip analog learning under nominal operating conditions and taking into account the impact of PVT variations.

We have calculated the power of this circuit as:

$$P = V_{ref1} \times I_{ref1} + V_{dd} \times I_{dd} \tag{20}$$

where I_{ref1} and I_{dd} were the current delivered by the voltage generator V_{ref1} and V_{dd} respectively. The calculated value from the DC simulations is P = 171 µW. It is worthnoting that [12] summarizes power consumptions of current accelerators and highlights the fact that conventional learning systems require more than 100 W. It is expected that neuromorphic systems based on the proposed analog circuit provide similar

learning capabilities as conventional approaches while reducing the power consumption.

IV. Conclusion

In this paper, we propose an improved version of the circuit described in the paper by Lu et al. The proposed circuit topology allows to appropriately produce a full sigmoid function and its approximate derivative using a $65\,\mathrm{nm}$ technology. The main objective is to minimize both energy consumption and also circuit area with the ultimate target of implementing large scale deep neural networks. The calculated power consumption of the circuit is $171\,\mu\mathrm{W}$. This power estimation is useful for dimensioning future neuromorphic networks based on analog circuits and for demonstrating the potential of new computational architectures which are currently under development.

References

[1] T. P. Xiao, C. H. Bennett, B. Feinberg, S. Agarwal, and M. J. Marinella, "Analog architectures for neural network acceleration based on non-volatile memory," *Applied Physics Reviews*, vol. 7, no. 3, 2020.

[2] C. Yakopcic, M. Z. Alom, and T. M. Taha, "Memristor crossbar deep network implementation based on a convolutional neural network," in *2016 International Joint Conference on Neural Networks (IJCNN)*, 2016, pp. 963–970.

[3] J. Grollier, D. Querlioz, K. Camsari, K. Everschor-Sitte, S. Fukami, and M. D. Stiles, "Neuromorphic spintronics," *Nature electronics*, vol. 3, no. 7, 2020.

[4] G. Bogason, "Generation of a neuron transfer function and its derivatives," *Electronics Letters*, vol. 29, no. 21, 1993.

[5] A. J. Annema, "Hardware realisation of a neuron transfer function and its derivative," *Electronics letters*, vol. 30, no. 7, 1994.

[6] K. Al-Ruwaihi, "Cmos analogue neurone circuit with programmable activation functions utilising mos transistors with optimised process/device parameters," *IEE Proceedings-Circuits, Devices and Systems*, vol. 144, no. 6, 1997.

[7] C. Lu, B. Shi, and L. Chen, "Analogue circuit realization of a programmable sigmoidal function and its derivative for on-chip bp learning," in *IEEE APCCAS 2000. 2000 IEEE Asia-Pacific Conference on Circuits and Systems. Electronic Communication Systems. (Cat. No.00EX394)*, 2000, pp. 626–629.

[8] Cadence, "Virtuoso analog design environment," https://www.cadence.com/en_US/home/tools/custom-ic-analog-rf-design/circuit-design/virtuoso-analog-design-environment.html.

[9] B. M. Wilamowski, "Neural network architectures and learning algorithms," *IEEE Industrial Electronics Magazine*, vol. 3, no. 4, 2009.

[10] A. S. Sedra and K. C. Smith, *Microelectronic Circuits*, 7th ed. Oxford University Press, 2015.

[11] B. Razavi, *Microelectronics*. Wiley, 2015.

[12] A. Reuther, P. Michaleas, M. Jones, V. Gadepally, S. Samsi, and J. Kepner, "Survey of machine learning accelerators," in *2020 IEEE High Performance Extreme Computing Conference (HPEC)*, 2020, pp. 1–12.