Analog Electronics Neural Networks: Analog Computing combined with Digital Data Processing Revisited

Stefan Bosse[1], Björn Lüssem[2]
[1]University of Koblenz, Dept. Computer Science, Koblenz, [2]University of Bremen, Faculty of Physics / Electrical Engineering, Bremen

Abstract — **Analog computing had its prime between the 1960 and 1970 years. With the raise of powerful digital computers analog computing using transistor and OPAMP circuits vanished nearly completely, but gained an increasing interest in recent years again. In this work, we will consider in particular analog ANN that are considered as co-processors for digital systems. We will show that the training result of an ANN using digital algorithms can be transferred on analog transistor circuits. But this process is still a challenge and can fail. So we discuss the limitations and possible solutions to generate and create analog ANN (AANN).**

Keywords — Non-destructive Testing, Damage Diagnostics, X-ray Radiography, Computer Tomography, Feature Detection, Machine Learning

1. Introduction

For decades signal processing was performed with analog electronics, including the era of analog computers, e.g., used for solving differential equations. In the last five decades, most analog circuits were substituted by digital electronic systems. Artificial Neural Networks (ANN) were originally inspired by analog systems, and implemented originally with analog electronics, but limited to one perceptron. Today they are computed by discretized digital computers. In this work, analog ANN (AANN) should be considered as co-processors and investigated with respect to their digital counterparts.

The motivation of this work is manifolded. Considering highly miniaturized and embedded sensor nodes based on digital silicon electronic (e.g., by using a microcontroller), providing less than 20 kB RAM and integer arithmetic only, computations of ANN are possible by transforming floating-point arithmetic models to scaled integer models without loss of accuracy (details can be found in [7]). But from a resource point of view with respect to digital logic, the computation of a fully connected ANN with N nodes requires roughly estimated about N^2*k transistors for storage (k is about 4-6) and M transistors for 8-bit arithmetic and code processing logic (M is about $10•10^3$-$100•10^3$). A weighted analog electronics summer circuit requires l+1 resistors and a difference amplifier with about 4-8 transistors (at least 2). An approximated non-linear transfer function, e.g., the sigmoid function, can be built from at least two transistors [1], and typically less than 20 transistors [2] if the gradient of the function is computed, too. The hyperbolic tangents function can be implemented with only two diodes [3]. Such small circuits are well suited for printed (organic) transistor electronics replacing more and more silicon electronics, but still limiting circuits to a size of about 100 transistors and posing reduced stability, reproducibility, and statistical variance (of the entire circuits).

In our work we address the following research questions to the computational ANN sub-domain:

1. Can AANN be trained with a digital node graph and floating point arithmetic performing gradient-based error optimization and finally be converted to an analog circuit approximation (assuming ideal operation-

al amplifiers)?

2. Can AANN be trained with a digital node graph and floating point arithmetic performing gradient-based error optimization and finally be converted to an analog circuit approximation assuming non-ideal circuits, especially with transistor-reduced circuits?
3. Are organic transistors suitable?

The implementation and approximation error of simple non-linear activation functions using transistor electronics are investigated and discussed. Instead using real analog electronics, we will substitute the circuits by a simulation model using the *spice3f* simulator [8], particularly the *ngspice* version [9]. We will consider different model abstraction levels, starting with ideal operational amplifier (voltage controlled voltage sources), then using approximated real OPAMP models, and finally introducing transistor circuits with models of organic transistors [6].

The next sections introduce the analog artificial neural network architecture and its electronic circuits with a short discussion of limitations. A short introduction in the digital twin ANN is given with modifications necessary for the digital.analog transformation process. An experimental section follows which applies the proposed transformation process to the IRIS benchmark dataset. Finally, the results are discussed and summarizing the lessons learned.

2. Analog Neural Networks

With respect to analog computing, we have to distinguish and consider:

- Different transistor technologies, e.g., Bipolar, JFET, OFET/OTFT, OECT;
- Operational amplifier (OPAMP) circuits with a minimal number of components (transistors);
- Non-linear transfer functions, e.g., logistic regression (sigmoid) or hyperbolic tangents, and their implementation with a minimal number of components;
- Transfer functions and characteristic curves of OPAMP/sigmoid circuits
- A composed neuron (perceptron) circuit;
- A full ANN.

We will start for sake of simplicity with traditional bipolar transistor circuits. The minimal number of transistors for an OPAMP and the sigmoid function is three without compromising usability, easy design procedure, and stability.

The circuit for a three-transistor OPAMP is shown in Fig. 1 posing a nearly linear transfer curve (with hard clipping), and a similar circuit for the smooth clipping non-linear sigmoid function implementation in Fig. 2. Both circuits base on a differential NPN transistor pair, followed by an current and voltage amplifying PNP transistor or a current amplifying NPN transistor, respectively. To achieve an output voltage range of nearly [-10V,10V], the power supply of the OPAMP3 circuit is set to [-10V,15V].

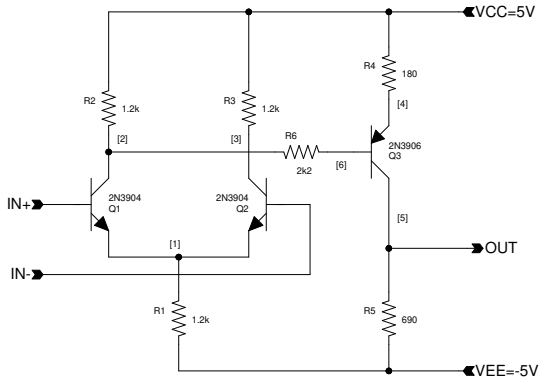

Fig. 1. OPAMP3 circuit (three transistor operational amplifier) using commonly used NPN and PNP bipolar transistors

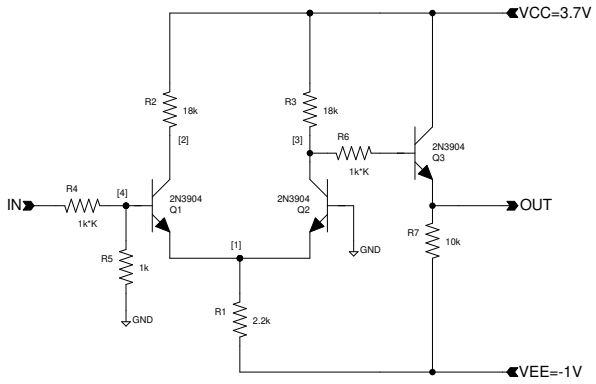

Fig. 2. SIGMOID3 circuit (three transistors) implementing the sigmoid function (output range 0-3V) using commonly used NPN bipolar transistors. The input resistor determines the *k* factor (sensitivity or x-range scaling).

The sigmoid three-transistor circuit has different x- and y-scaling compared with the mathematical function, but conforms with high accuracy to the scaled mathematical function, as shown in Fig. 3. The x-scaling can be set by the input resistor multiplication factor *k*. The y-scale is always approximately in the value range [0.05V,2.9V]. The SIGMOID3 circuit needs a slightly odd power supply [-1V,3.7V].

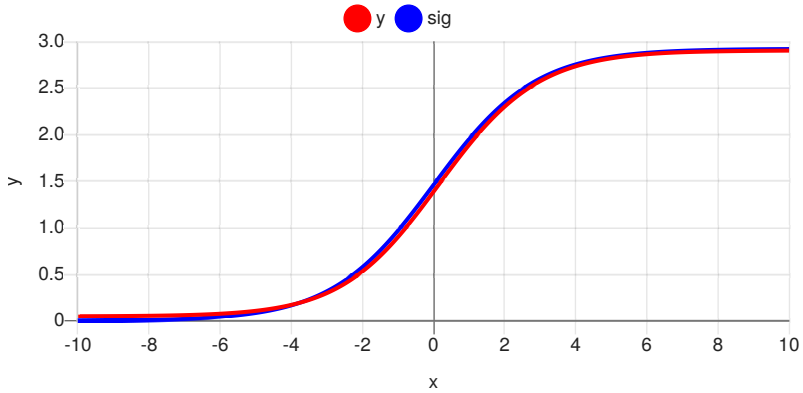

Fig. 3. Plot of analog SIGMOID3 (y) output and mathematical (sig) function with k=50, sigmoid x-scaling=0.7 and y-scaling=2.9.

Having defined the elementary cells OPAMP3 and SIGMOID3 of a neural network we can compose neurons (one perceptron), layers of neurons, and entire networks. An ANN is described by the layer-network structure and parameters (weights, bias). Weights and bias values can be positive or negative. In principle, a common difference amplifier can be used. But we will have commonly more than one negative and positive input, making the parametrization of such circuits difficult (negative and positive gain can not be indecently controlled). Therefore, we split the input path of a neuron into two paths, one for negative weights and negative bias (if any), and one for positive weights and bias (if any), finally merged by a unity gain difference amplifier. The entire architecture of a neuron is shown in Fig. 4.

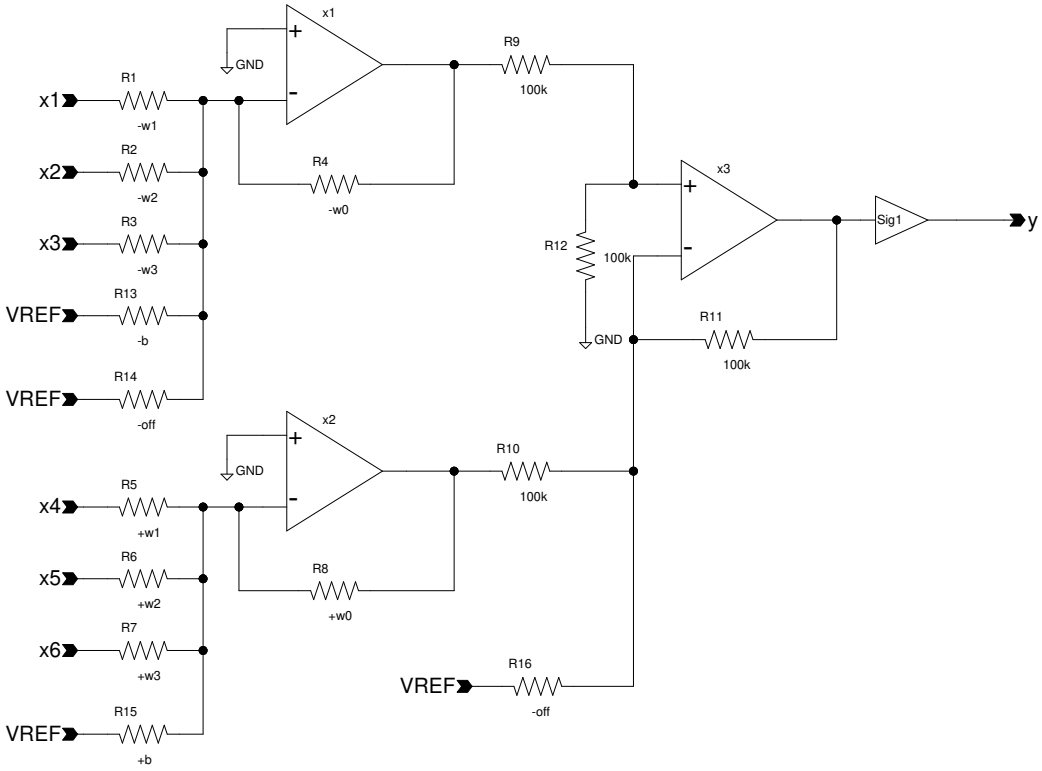

Fig. 4. Single perceptron (neuron) circuit using one OPAMP3 circuit for all negative weights and negative bias, one OPAMP3 circuit for positive weights and positive bias (mutual exclusive), one difference OPAMP3 circuit combining both temporary outputs, and finally applying the sigmoid function.

Due to the current-controlled current-source model of a bipolar transistor and current flows from base to emitter/collector the gain of such a simplified OPAMP will be lower as compared with the gain of a mathematical ideal OPAMP. This gain mismatch (representing the weight of a neuron) requires a correction of the input resistor with a function depending on the original computed input resistor r_{i} value in relation to the feedback resistor r_{f}:

$$\begin{aligned} G &= G_0 \cdot F(r_f, r_i) \\ F(r_f, r_i) &= [0.6, 0.9] \end{aligned} \tag{1}$$

Additionally, there is a significant output offset of such simplified OPAMP circuit (up to 3V), which must be compensated by a feeding a compensation current flow via a resistor into the inverting input node. The offset voltage depends on the feedback resistor value and the accumulative (parallel) resistor of all input resistors connected to the inverting input node. The dependency is extended if the non-inverting input is not grounded (as in the case of the difference amplifier, but fortunately having constant gain and resistor networks).

Assuming a fixed feedback resistor of an OPAMP r_{f}, e.g. 100 kΩ, an input resistor of the inverting

OPAMP node is computed by:

$$r_i = G\left(r_f, \frac{r_f}{w_i}\right) \tag{2}$$

3. Transformation Methods

An analog circuit is basically an undirected mesh graph of current nodes, i.e., an electronic circuit has no real dedicated input and output ports. A digital feed-forward neural network, in contrast, is a directed graph of functional nodes. There are basically two analog architectures which must be distinguished by a digital-to-analog transformation process:

1. Circuits with ideal OPAMPs and transfer functions (sigmoid), i.e., a theoretical and mathematical model of a neuron, which can use original ANN models and training methods and a direct digital-to-analog mapping model;
2. Circuits with non-ideal OPAMPs and transfer functions requiring modified models and training algorithms and an advanced digital-to-analog mapping model.

The first approach can be sub-divided into unconstrained ideal OPAMP and nearly-ideal but constrained real OPAMPS. An ideal OPAMP is characterized by an infinite open loop gain and infinite input and output value ranges. A real OPAMP has a finite high open loop gain (> 100000), but limited output value ranges, given by the supply voltages, e.g., [-10V,10V]. The digital-to-analog transformation of ideal OPAMP circuits is trivial and not further considered in this work. The transformation of real OPAMP circuits or non-ideal OPAMP circuits is a challenge.

The challenges are:

1. Limited open loop gain (50-100) creating a limit of the weights (< 50);
2. Intermediate values can exceed the output range of OPAMPs and clipping occurs;
3. The input- and output-range of non-linear transfer functions (e.g., sigmoid) is different from the mathematical version.
4. Real OPAMP circuits pose non-linearity (distortion) and highly relevant output offset voltages (Δ).
5. Composed circuits with bipolar transistors pose complex side effects and further deviation from ideal OPAMP circuits due to the current-controlled current-source operational model of such transistors.

To reflect the limitations and deviation of reduced transistor circuits compared with ideal OPAMP models, the neuron architecture in the digital model must be modified, as shown in Fig. 5. Additional clipping and scaling blocks are added to the weighted summation function and the non-linear sigmoid transfer function. Due to the limited open loop gain of the considered OPAMP3 circuit, weight parameter clipping is added, too.

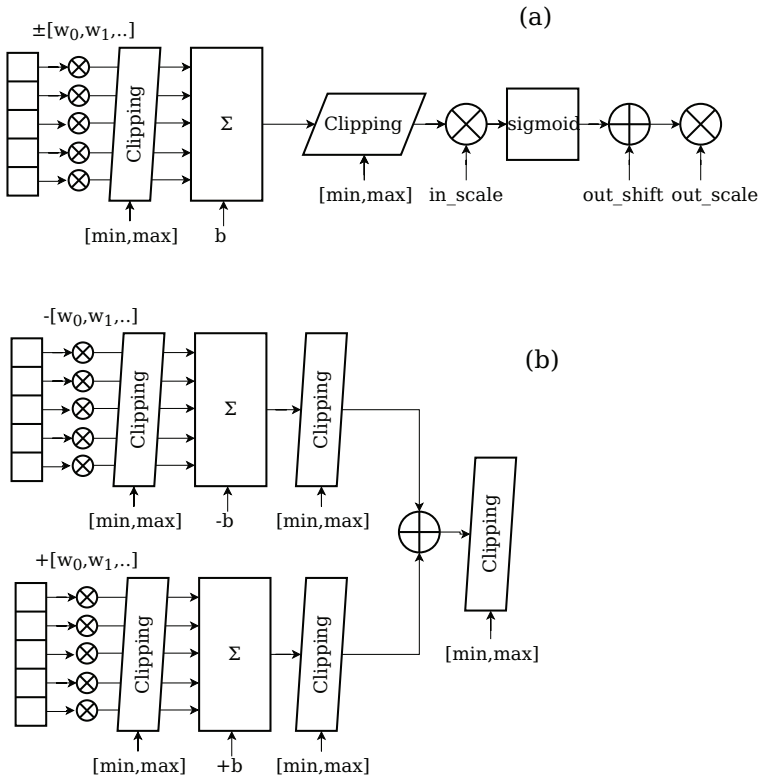

Fig. 5. Modified digital neuron architectures with clipping and scaling (a) Simplified (b) With separate negative and positive weight paths

The ANN is trained with scaled and normalized data by using the digital modified network model and commonly available training algorithms, e.g., ADAM, SGD, and so on. The intermediate value and weight parameter clipping introduces distortion in the training process, but results commonly still in a satisfying model parameter optimization and prediction error minimization. We assume a 1:1 digital-analog value mapping, i.e., a digital (mathematical) value 1 is corresponding to a voltage of 1 V.

The clipping parameters and the x-scaling of the transfer functions must be chosen carefully. In model architecture (a) even if there is an output clipping comparable to the electronic circuit behavior, there can be intermediate value overflows in one or both weight amplification branches. Higher mathematical values are not an issue for digital computations, but with a 1:1 digital-analog mapping the absolute limits are given by the power suppüly voltages of the transistor circuits.

Therefore, the non-linear sigmoid function should be highly sensitive (i.e., low k values and high x-scaling). But deviations of analog circuits like offset voltages can shift the transfer curve to their 0/1 boundaries resulting in saturated nodes not present in the digital model. A suitable compromise must be found on an iterative base.

The analog sigmoid function has a fixed output value range of about [0V,3V]. To reduce the risk of intermediate network values higher than the clipping range, the digital model is trained with a sigmoid value range [0,1], finally reducing all weights connected to the output of a sigmoid function by a facto of three.

We used the JavaScript ConvNetJS software framework ([11], consisting of one file) to apply our modifications. ConvNetJS provides advanced trainers and a broad range of network architectures, but is still very com-

pact and easy to maintain. The main modification was the replacement of a commonly used practice to express the gradient function g (of the transfer function f) as a function of the original transfer function, e.g., in the case of the sigmoid function $y=sigmoid(x)$ this is $y(y-1)$, i.e., computing the gradients from the output values y. Instead, we modified the gradient computation by computing the gradient as a function of the input x. Finally, we added weight (filter) and output clipping.

The analog circuit is directly synthesized from the trained digital model. The synthesizer *aanngen* has to perform:

- (spice) net-list generation,
- rescaling,
- resistor computation from weight and bias parameters under amplification correction and connecting them to the appropriate sub-circuits (OPAMP3 OPN/OPP sub-circuits for negative and positive weights/bias, respectively),
- adding and connecting sigmoid analog sub-circuits,
- adding offset correction resistors based on computed circuit components.

The layer structure, weight and bias parameters from the trained digital model is exported in JSON format and processed by the synthesizer program. Currently, the synthesizer creates a ngspice net-list with simulation control statements testing the analog circuit with test data.

4. Example: Benchmark IRIS dataset classification

An ANN with a layer structure of [4,3,3] and scaled sigmoid transfer functions were trained with the benchmark IRIS data set consisting of 151 data instances. Because this study is only a proof of concept and comparison of a digital and an analog circuit model, training and test were performed with the entire data set.

The input vector x consists of four physical parameters (length, width, petal length and width), which are normalized to the range [0,1] independently, finally correlating to the analog voltage ranges [0V,1V]. The three species classes are one-hot encoded (y). There is no soft-max layer at the output of the ANN model due to a lack of analog circuits implementing an interconnected multi-node function accurately. The first input layer is not present in the analog circuit (it is a pass-through layer).

The training was performed with the ADAM optimize, α=0.02,γ=0.5, and a batch size of 5. The filter clipping was set to 5, the output scaling (of the summation function) was set to [-10,10]. A typical model parameter set achieved after 10000 single training iterations (by selecting training instances randomly) is shown in Fig. 6. The classification results of the (clipped) digital model compared with the results from analog circuit are shown in Fig. 7. The circuit was simulated by using *ngspice* with altered settings of the input vector x.

The results shows that the transformation process from a digital to an analog model succeeded. The average classification error increased (from 3% to 10%), but the overall accuracy of the analog model is still good and comparable to the digital model. Due to the limitations of the used simple and minimalistic circuits the results are better than expected. Offsets and gains deviations were not fully compensated in the analog model.

```
====== Weights layer 1 ======
[4.77,-2.39,4.10,0.66]
[0.74,1.47,-5.00,-5.00]
[-4.94,-1.72,-4.52,-4.29]
====== Weights layer 2 ======
[-4.85,4.55,-1.28]
[4.53,4.36,-3.90]
[1.89,-4.34,-3.33]
====== Bias layer 1 ======
```

```
[-1.36,5.34,-5.03]
====== Bias layer 2 ======
[-2.22,-6.71,0.29
```

Fig. 6. Parameters of the digital IRIS classification model

```
              Prediction                          Prediction
              C     A     B                       C     A     B
Reference  C  50    0     0          Reference  C  50    0     0
           A  0     51    0                     A  0     51    0
           B  4     0     46                    B  15    0     35

N  : [50,51,50] (151)                N  : [50,51,50] (151)
TP : [50,51,46] (147)                TP : [50,51,35] (136)
TN : [97,100,101] (298)              TN : [86,100,101] (287)
FP : [4,0,0] (4)                     FP : [15,0,0] (15)
FN : [0,0,4] (4)                     FN : [0,0,15] (15)
Unique    : [C,A,B]                  Unique    : [C,A,B]
Error     : [0.00,0.00,0.08] (0.03)  Error     : [0.00,0.00,0.30] (0.10)
Accuracy  : [1.00,1.00,0.92] (0.97)  Accuracy  : [1.00,1.00,0.70] (0.90)
Precision : [0.93,1.00,1.00] (0.97)  Precision : [0.77,1.00,1.00] (0.90)
Recall    : [1.00,1.00,0.92] (0.97)  Recall    : [1.00,1.00,0.70] (0.90)
F1 Score  : [0.96,1.00,0.96] (0.97)  F1 Score  : [0.87,1.00,0.82] (0.90)
-----------------------------------  -----------------------------------
Digital Model                        Analog Model
```

Fig. 7. Comparison of prediction results of the (modified and clipped) digital and the analog circuit model (Classes: A=setosa, B=versicolor, C=virginica)

5. Discussion

Due to the non-ideal analog circuit behavior compared with mathematical ideal operational amplifiers there is an increasing accumulative error with an increased output deviation and prediction errors, finally reducing the safety margin in classification. The non-ideal behavior bases on:

1. Output offset of transistor circuits (OPAMP3) and offset correction coefficient base on resistors networks of entire sub-circuit;
2. Lowered gain (which must be corrected) on inverting input and gain correction coefficient depends on resistor networks;
3. Limited gain due to low open gain factor;
4. Drift due temperature variation;
5. Transistor parameter variations (e.g., *hfe*);
6. Deviation of the SIGMOID3 transfer curve from the mathematical (scaled) sigmoid function.

The y-scaling of the sigmoid is fixed, but the x-scaling can be freely chosen. A small x-scaling decreases the output values of the summation circuit, but increase the sensitivity to offset errors. A larger x-scaling results in the opposite relationship.

Offsets and gains deviations were not fully compensated in the analog model using approximated and simplified calculation models derived from simulation, but the analog model is still usable. But with an increasing number of layers (and neurons per layer), the value errors accumulates and can lower the model ac-

curacy until uselessness.

6. From Silicon to Organic Printed Electronics

The future goal of this work is to transform and implement digital computation in analog organic printed electronics. We can distinguish different organic transistor technologies:

1. Organic Thin-film Field-effect Transistors (OFET,OTFT) [6]
2. Organic Electrochemical Transistors (OECT) [10]

We started with silicon bipolar transistors to show the principle possibility to transform digital to analog models. We were able to create neural circuits sufficiently close to the digital model behavior with a minimal number of transistors. The next step is the replacement of BJT transistors with voltage-controlled JFET transistors and finally with OFET and OECT transistors. But some selected IV curve characteristics show the next significant challenge. The JFTE and BCT curves are comparable with respect to steepness (gain) and JFET circuits are well understood. The OFET curve [6] shows a totally different behavior with respect to steepness and scale. In [10] the authors presented OECT transistors with a much more promising behavior maybe suitable to create OPAMP3 and SIGMOID3 circuits.

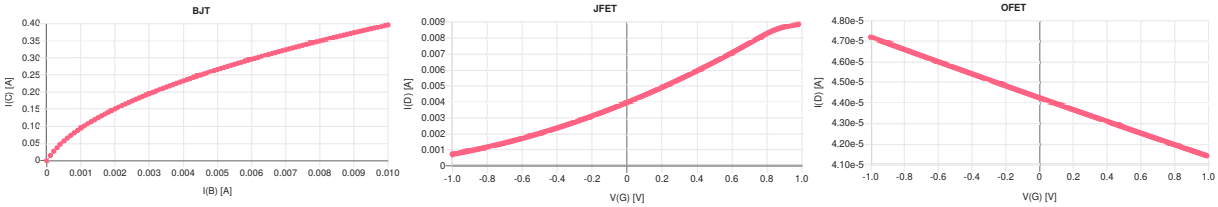

Fig. 8. Simulated transistor IV curve characteristics (V_{CE}=V_{DS}=10V) for different transistor technologies: (a) Silicon NPN-BJT (b) Silicon N-JFET (c) p-OFET (spice model from [6])

Assuming a typical size of organic printed transistors of about 200 × 200 µm [12], the circuit presented in this work with 66 transistors and 75 resistors would cover an area of about 2 × 2 mm, sufficiently small to be integrated in material-integrated sensor nodes [13].

7. Conclusions

We could show that digital ANN models can be transformed into analog circuits with minimal transistor counts. The presented analog transistor circuits OPAMP3 and SIGMOID3 are elementary cells and building blocks for neurons and neural networks. Each sub-circuit requires only 3 transistors. We tested and evaluated out approach with the IRIS benchmark dataset. We found that the digital model must be modified to reflect real circuit clipping (saturation) and limited open loop gain (limiting teh maximal weights). The presented AANN example circuit consists of 16 OPAMP3 components and 6 SIGMOID3 components, in total 66 transistors and 75 resistors. Although, the average classification error increased from 3 to 10%, the overall model accuracy is preserved.

To conclude: The digital-to-analog transformation of ANN is possible, but the imperfections and correction of the simplified transistor circuits are limiting factors, especially for larger networks. We propose to use surrogate ML models of the sub-circuits for different parameter settings and IO characteristics derived and trained from simulation and integrated in the digital model training. The seems inevitable if OFET and OECT

transistor technologies with much higher degree of imperfections ares used.

8. Acknowledgments

The authors expressly acknowledge the financial support of the research work on this article within the Research Unit 3022 “Ultrasonic Monitoring of Fibre Metal Laminates Using Integrated Sensors” (Project number: 418311604) by the German Research Foundation (Deutsche Forschungsgemeinschaft (DFG)).

9. References

[1] https://acidbourbon.wordpress.com/2024/02/16/implementing-a-sigmoid-function-in-analog-circuitry, accessed on-line 1.9.2024

[2] https://telecom-paris.hal.science/hal-03331347/document, accessed on-line 1.9.2024

[3] https://electronics.stackexchange.com/questions/657902/what-are-the-activation-functions-that-can-be-generated-using-op-amps-and-filter, accessed on-line 1.9.2024

[4] Xu, S.; Li, X.; Xie, C.; Chen, H.; Chen, C.; Song, Z. A High-Precision Implementation of the Sigmoid Activation Function for Computing-in-Memory Architecture. Micromachines 2021, 12, 1183. https://doi.org/10.3390/mi12101183

[5] https://github.com/sudharsan2000/analog-NN, accessed on-line 1.9.2024

[6] Valletta, A., Demirkol, A. S., Maira, G., Frasca, M., Vinciguerra, V., Occhipinti, L. G., ... & Fortunato, G. (2016). A compact SPICE model for organic TFTs and applications to logic circuit design. IEEE Transactions on Nanotechnology, 15(5), 754-761., https://core.ac.uk/download/pdf/96707259.pdf

[7] S. Bosse, A Virtual Machine Platform Providing Machine Learning as a Programmable and Distributed Service for IoT and Edge On-Device Computing: Architecture, Transformation, and Evaluation of Integer Discretization, Algorithms. 2024; 17(8):356. https://doi.org/10.3390/a17080356

[8] Quarles, T., Newton, A. R., Pederson, D. O., & Sangiovanni-Vincentelli, A. (1994). Spice 3 version 3f5 user’s manual

[9] ngspice, https://ngspice.sourceforge.io, accessed on-line 1.10.2024

[10] P. R. Paudel, V. Kaphle, D. Dahal, R. K. RadhaKrishnan, B. Lüssem, Tuning the Transconductance of Organic Electrochemical Transistors. Adv. Funct. Mater. 2021, 31, 2004939. https://doi.org/10.1002/adfm.202004939

[11] ConvNetJS, accessed on-line, https://cs.stanford.edu/people/karpathy/convnetjs/, 2024

[12] P. A. Ersman, Screen printed digital circuits based on vertical organic electrochemical transistors, Flex. Print. Electron, vol. 2, 2017.

[13] S. Bosse, D. Lehmhus, W. Lang, M. Busse (Ed.), Material-Integrated Intelligent Systems: Technology and Applications, Wiley, ISBN: 978-3-527-33606-7 (2018)

10. Appendix

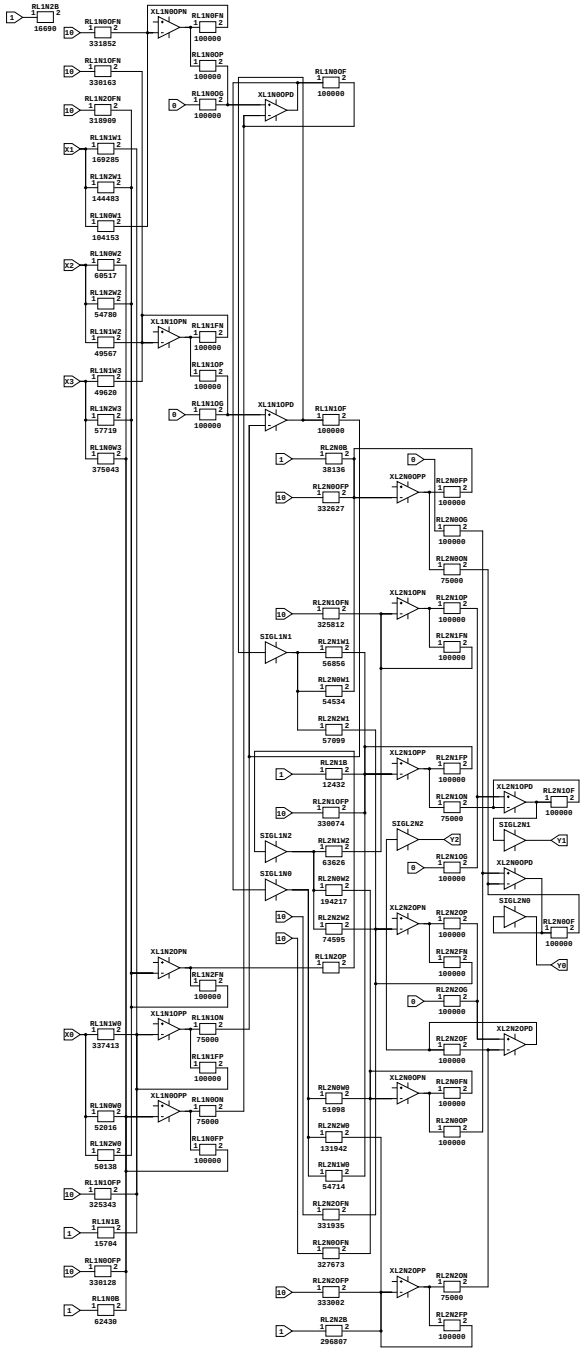

Fig. 9. AANN circuit (for IRIS dataset classification)

```
*
*  SPICE model for 2N3904/MMBT3904 transistor
```

```
*
.model 2N3904 NPN(IS=4.639E-15 NF=0.9995 ISE=2.091E-14 NE=1.6 BF=160.1 IKF=0.12
+                 VAF=98.69 NR=1.001 ISC=3.257E-12 NC=1.394 BR=5.944 IKR=0.06
+                 VAR=19.29 RB=1 IRB=1E-6 RBM=1 RE=0.3614 RC=1.755 XTB=0
+                 EG=1.11 XTI=3 CJE=5.631E-12 VJE=0.7002 MJE=0.3385
+                 TF=3.001E-10 XTF=27 VTF=1.461 ITF=0.2723 PTF=0 CJC=4.949E-12
+                 VJC=0.5969 MJC=0.1928 XCJC=0.864 TR=9.4E-8 CJS=0 VJS=0.75
+                 MJS=0.333 FC=0.5582)
.model 2N3906 PNP(IS=1E-14 VAF=100
+  BF=200 IKF=0.4 XTB=1.5 BR=4
+  CJC=4.5E-12 CJE=10E-12 RB=20 RC=0.1 RE=0.1
+  TR=250E-9   TF=350E-12 ITF=1 VTF=2 XTF=3 Vceo=40  Icrating=200m)
```

Fig. 10. Spice model of bipolar transistors

```
*              IN+ IN- OUT
.subckt OPAMP3 IN1 IN2 OUT VP=5 VN=-5
V1 VSS 0 dc {VP}
V3 VEE 0 dc {VN}
R1 1 VEE 1200
R2 VSS 2 1200
R3 VSS 3 1200
R4 VSS 4 180
R5 5 VEE 690
R6 2 6 2000
Q1 2 IN1 1 2N3904
Q2 3 IN2 1 2N3904
Q3 5 6 4 2N3906
R7 5 OUT 10
*.op
.ends
```

Fig. 12. Spice model of OPAMP3 circuit

```
*               IN OUT K
.subckt SIGMOID IN OUT K=10.0 VP=3.7 VN=-1.0
V1 VSS 0 dc {VP}
V2 VEE 0 dc {VN}
R1 1 VEE 2200
R2 VSS 2 18000
R3 VSS 5 18000
R4 IN 4 {K*1000}
R5 4 0 1000
Q1 2 4 1 2N3904
Q2 5 0 1 2N3904
* C B E
R6 5 6 10000
Q3 VSS 6 OUT 2N3904
R7 OUT VEE 10000
.ends
```

Fig. 12. Spice model of SIGMOID3 circuit

```
VREF1 REF 0 1
```

```
VREF10 REF10 0 10
* X Input Vector (assuming normalized voltages 0-1V
VX0 INX0 0 DC 0
VX1 INX1 0 DC 0
VX2 INX2 0 DC 0
VX3 INX3 0 DC 0
* Layer 1
* Layer 1 Node 0
RL1N0W0 INX0 OPL1N0PT 17702
RL1N0W1 INX1 OPL1N0NT 35445
RL1N0W2 INX2 OPL1N0PT 20566
RL1N0W3 INX3 OPL1N0PT 128439
RL1N0B REF OPL1N0NT 62430
RL1N0OFN REF10 OPL1N0NT 330226
* OPAMP W- L1N0
RL1N0FN OPL1N0NOUT OPL1N0NT 100000
* OPAMP W-: IN+ IN- OUT
XL1N0OPN 0 OPL1N0NT OPL1N0NOUT OPAMP3 VP=15 VN=-10
RL1N0OFP REF10 OPL1N0PT 318610
* OPAMP W+ L1N0
RL1N0FP OPL1N0POUT OPL1N0PT 100000
* OPAMP W+: IN+ IN- OUT
XL1N0OPP 0 OPL1N0PT OPL1N0POUT OPAMP3 VP=15 VN=-10
* DIFF(-,+)
RL1N0OP OPL1N0NOUT OPL1N0DP 100000
RL1N0ON OPL1N0POUT OPL1N0DN 75000
RL1N0OF OPL1N0DOUT OPL1N0DN 100000
RL1N0OG OPL1N0DP 0 100000
RL1N0OFD REF10 OPL1N0DN 2800000
* OPAMP DIFF(W- - W+): IN+ IN- OUT
XL1N0OPD OPL1N0DP OPL1N0DN OPL1N0DOUT OPAMP3 VP=15 VN=-10
* SIGMOID
XL1N0OSG OPL1N0DOUT OUTL1N0 SIGMOID K=10
* Layer 1 Node 1
RL1N1W0 INX0 OPL1N1PT 115552
RL1N1W1 INX1 OPL1N1PT 57719
RL1N1W2 INX2 OPL1N1NT 16777
RL1N1W3 INX3 OPL1N1NT 16887
RL1N1B REF OPL1N1PT 15704
RL1N1OFN REF10 OPL1N1NT 318803
* OPAMP W- L1N1
RL1N1FN OPL1N1NOUT OPL1N1NT 100000
* OPAMP W-: IN+ IN- OUT
XL1N1OPN 0 OPL1N1NT OPL1N1NOUT OPAMP3 VP=15 VN=-10
RL1N1OFP REF10 OPL1N1PT 325708
* OPAMP W+ L1N1
RL1N1FP OPL1N1POUT OPL1N1PT 100000
* OPAMP W+: IN+ IN- OUT
XL1N1OPP 0 OPL1N1PT OPL1N1POUT OPAMP3 VP=15 VN=-10
* DIFF(-,+)
RL1N1OP OPL1N1NOUT OPL1N1DP 100000
RL1N1ON OPL1N1POUT OPL1N1DN 75000
RL1N1OF OPL1N1DOUT OPL1N1DN 100000
RL1N1OG OPL1N1DP 0 100000
RL1N1OFD REF10 OPL1N1DN 2800000
* OPAMP DIFF(W- - W+): IN+ IN- OUT
XL1N1OPD OPL1N1DP OPL1N1DN OPL1N1DOUT OPAMP3 VP=15 VN=-10
* SIGMOID
XL1N1OSG OPL1N1DOUT OUTL1N1 SIGMOID K=10
* Layer 1 Node 2
RL1N2W0 INX0 OPL1N2NT 17063
```

```
RL1N2W1 INX1 OPL1N2NT 49262
RL1N2W2 INX2 OPL1N2NT 18642
RL1N2W3 INX3 OPL1N2NT 19643
RL1N2B REF OPL1N2NT 16690
RL1N2OFN REF10 OPL1N2NT 308362
* OPAMP W- L1N2
RL1N2FN OPL1N2NOUT OPL1N2NT 100000
* OPAMP W-: IN+ IN- OUT
XL1N2OPN 0 OPL1N2NT OPL1N2NOUT OPAMP3 VP=15 VN=-10
* NO D-AMP needed
RL1N2OF OPL1N2NOUT OPL1N2DOUT 0
* SIGMOID
XL1N2OSG OPL1N2DOUT OUTL1N2 SIGMOID K=10
* Layer 2
* Layer 2 Node 0
RL2N0W0 OUTL1N0 OPL2N0NT 51098
RL2N0W1 OUTL1N1 OPL2N0PT 54534
RL2N0W2 OUTL1N2 OPL2N0NT 194217
RL2N0B REF OPL2N0NT 38136
RL2N0OFN REF10 OPL2N0NT 327673
* OPAMP W- L2N0
RL2N0FN OPL2N0NOUT OPL2N0NT 100000
* OPAMP W-: IN+ IN- OUT
XL2N0OPN 0 OPL2N0NT OPL2N0NOUT OPAMP3 VP=15 VN=-10
RL2N0OFP REF10 OPL2N0PT 332627
* OPAMP W+ L2N0
RL2N0FP OPL2N0POUT OPL2N0PT 100000
* OPAMP W+: IN+ IN- OUT
XL2N0OPP 0 OPL2N0PT OPL2N0POUT OPAMP3 VP=15 VN=-10
* DIFF(-,+)
RL2N0OP OPL2N0NOUT OPL2N0DP 100000
RL2N0ON OPL2N0POUT OPL2N0DN 75000
RL2N0OF OPL2N0DOUT OPL2N0DN 100000
RL2N0OG OPL2N0DP 0 100000
RL2N0OFD REF10 OPL2N0DN 2800000
* OPAMP DIFF(W- - W+): IN+ IN- OUT
XL2N0OPD OPL2N0DP OPL2N0DN OPL2N0DOUT OPAMP3 VP=15 VN=-10
* SIGMOID
XL2N0OSG OPL2N0DOUT OUTL2N0 SIGMOID K=10
* Layer 2 Node 1
RL2N1W0 OUTL1N0 OPL2N1PT 54714
RL2N1W1 OUTL1N1 OPL2N1PT 56856
RL2N1W2 OUTL1N2 OPL2N1NT 63626
RL2N1B REF OPL2N1NT 12432
RL2N1OFN REF10 OPL2N1NT 325812
* OPAMP W- L2N1
RL2N1FN OPL2N1NOUT OPL2N1NT 100000
* OPAMP W-: IN+ IN- OUT
XL2N1OPN 0 OPL2N1NT OPL2N1NOUT OPAMP3 VP=15 VN=-10
RL2N1OFP REF10 OPL2N1PT 330074
* OPAMP W+ L2N1
RL2N1FP OPL2N1POUT OPL2N1PT 100000
* OPAMP W+: IN+ IN- OUT
XL2N1OPP 0 OPL2N1PT OPL2N1POUT OPAMP3 VP=15 VN=-10
* DIFF(-,+)
RL2N1OP OPL2N1NOUT OPL2N1DP 100000
RL2N1ON OPL2N1POUT OPL2N1DN 75000
RL2N1OF OPL2N1DOUT OPL2N1DN 100000
RL2N1OG OPL2N1DP 0 100000
RL2N1OFD REF10 OPL2N1DN 2800000
* OPAMP DIFF(W- - W+): IN+ IN- OUT
```

```
XL2N1OPD OPL2N1DP OPL2N1DN OPL2N1DOUT OPAMP3 VP=15 VN=-10
* SIGMOID
XL2N1OSG OPL2N1DOUT OUTL2N1 SIGMOID K=10
* Layer 2 Node 2
RL2N2W0 OUTL1N0 OPL2N2PT 131942
RL2N2W1 OUTL1N1 OPL2N2NT 57099
RL2N2W2 OUTL1N2 OPL2N2NT 74595
RL2N2B REF OPL2N2PT 296807
RL2N2OFN REF10 OPL2N2NT 331935
* OPAMP W- L2N2
RL2N2FN OPL2N2NOUT OPL2N2NT 100000
* OPAMP W-: IN+ IN- OUT
XL2N2OPN 0 OPL2N2NT OPL2N2NOUT OPAMP3 VP=15 VN=-10
RL2N2OFP REF10 OPL2N2PT 333002
* OPAMP W+ L2N2
RL2N2FP OPL2N2POUT OPL2N2PT 100000
* OPAMP W+: IN+ IN- OUT
XL2N2OPP 0 OPL2N2PT OPL2N2POUT OPAMP3 VP=15 VN=-10
* DIFF(-,+)
RL2N2OP OPL2N2NOUT OPL2N2DP 100000
RL2N2ON OPL2N2POUT OPL2N2DN 75000
RL2N2OF OPL2N2DOUT OPL2N2DN 100000
RL2N2OG OPL2N2DP 0 100000
RL2N2OFD REF10 OPL2N2DN 2800000
* OPAMP DIFF(W- - W+): IN+ IN- OUT
XL2N2OPD OPL2N2DP OPL2N2DN OPL2N2DOUT OPAMP3 VP=15 VN=-10
* SIGMOID
XL2N2OSG OPL2N2DOUT OUTL2N2 SIGMOID K=10
```

Fig. 13. Spice model of the entire synthesized AANN for the IRIS dataset classification