Speeding-Up Complex RF IC Sizing Optimizations with a Process, Voltage and Temperature Corner Performance Estimator using Deep ANNs

Pedro José Vaz
Instituto de Telecomunicações
Instituto Superior Técnico – Universidade de Lisboa, Lisboa, Portugal
pedro.jose.vaz@tecnico.ulisboa.pt

Abstract—**Despite the advances of the state-of-the-art on electronic design automation tools, nowadays, most of the automatic circuit sizing processes require a high amount of time to conclude, due to the increasing complexity of radio-frequency (RF) blocks, integration technologies and stringent specifications under several different process, voltage, and temperature (PVT) corner conditions. This work explores an innovative approach to automatic circuit sizing of RF integrated circuit (IC) blocks using deep learning techniques and, more specifically, artificial neural networks (ANNs), to complement and improve an optimization-based sizing loop. In order to find all the relevant performance figures of a certain circuit sizing solution, the optimization loop simulates and evaluates the desired circuit topology under different process fabrication dispersions, as well as voltage and temperature variations, which are also known as PVT corner analysis. The ANN architecture proposed in this work is a regression-only model. The goal of this model is to estimate all the relevant circuit performances in PVT corners using, as input features, the circuit's sizing, and the accurate performance figures in typical conditions, and thus, speeding-up the optimization process by bypassing the time-consuming circuit simulation. This model will complement and speed-up the optimization loop of the AIDA tool. The proposed PVT estimator was integrated on the AIDA tool and tested on three different circuit sizing optimizations: a class C/D voltage-controlled oscillator (VCO) for 3.5-to-4.8 GHz and 2.3-to-2.5 GHz ranges, and an ultralow power class B/C VCO. The results obtained show that the PVT estimator was able to reduce the workload of the circuit simulator up to 78.5%, while achieving a total optimization speed-up factor of 2.92.**

Keywords—Artificial Neural Networks, Automatic Sizing Optimization, Circuit Sizing, Deep Learning, Electronic Design Automation, Radio-Frequency Integrated Circuits

I. Introduction

Over the last few years, the electronics industry has experienced a massive increase in the demand for smaller and more complex integrated systems, mainly due to the rise of portable devices. Now more than ever, developers are faced with the challenge of creating more powerful systems while ensuring smaller size and low power consumption. Technologies such as the internet of things (IoT) or 5th generation broadband (5G), will join millions of devices and sensors together, enabling great advances in education, healthcare, transportation, agriculture, amongst many other areas. All these applications continuously gather an increasing amount of data, posing unprecedented challenges to each element of the networks. Due to this, today's market demands high communication rates, large bandwidths, and ultralow-power consumptions, in which radio frequency (RF) integrated circuits (ICs) play a critical role. Most of these systems implement a combination of both analog/RF and digital circuits in the same chip, i.e., a Mixed-Signal (MS) Systems-on-Chip (SoC).

In addition, for IoT and 5G, the design of RF and mm-Wave ICs in deep nanometric technologies is becoming extremely difficult because of their high complexity, demanding performances, and their need to be designed and manufactured at minimal costs under strict time-to-market constraints.

To aid designers overcome the many difficulties encountered in manual sizing of analog/RF IC blocks, several optimization-based sizing approaches emerged. These electronic design automation (EDA) tools use several algorithms that explore the design space effectively, rather than iterating over designer-defined analytical equations. They can be used along with performance models that can capture several circuit characteristics of RF circuits. However, despite its increased computational effort, utilizing foundry-provided device models and a circuit simulator as an evaluation engine resulted in the most accurate and generally adopted approach. Most of the commercially available solutions that use the simulation-based architecture, e.g., Cadence's Virtuoso GXL [1] or MunEDA's DNO/GNO [2], still take a restrictive single-objective approach being used to semi-automate the manual sizing design process. Consequently, simulation-based techniques are a continuous subject of research of the community to face the most recent design challenges.

However, there are some problems that must be mandatorily addressed. The optimization-based sizing loop requires a considerable time to complete. For example, in [3], the optimization-based sizing loop of a class C/D voltage-controlled oscillator (VCO), performed with a population of 512 elements and 200 generations, took 50 hours to complete, and in [4], with a population of 256 elements and just 100 generations the optimization-based sizing loop of a class B/C VCO, took 367 hours to complete. These optimizations were carried in a machine with an Intel-Xeon E5-2630-v3@2.40 GHz with 64 GB of RAM using 8 cores for parallel evaluation. For a developer that needs to meet the stringent time-to-market constraints, such optimizations' durations may not be viable, and thus, a speed-up is necessary.

In an optimization loop, the candidate sizing solution's relevant performances are found by simulating and evaluating the circuit under process fabrication dispersion and voltage and temperature variations, also known as PVT corner analysis, which take most of the computational effort. Therefore, this work proposes:

- Accelerating the analog/RF IC optimization-based sizing loop of the AIDA tool [5] using artificial neural networks (ANNs) to complement the simulation process, and therefore reducing the simulator workload;

- The model aims at estimating circuit performance in the PVT corners using the circuit's sizing and performance on typical conditions as input features;
- The model used for a particular circuit topology can be reused for optimizations with completely different targets of the same circuit topology (plug-and-play functionalities);
- The structure of the model used for a particular VCO circuit topology can be reused for different VCO circuit topologies (plug-and-train functionalities).

To test these objectives, the PVT estimator was integrated on the AIDA tool and tested on three different circuit sizing optimizations: a class C/D VCO for 3.5-to-4.8 GHz and 2.3-to-2.5 GHz ranges, and finally, an ultralow power class B/C VCO.

This document is organized as follows. In Section II, the related work regarding acceleration of IC sizing optimizations using ANNs is presented. In Section III, the PVT Estimator is described along with its development. In Section IV, the experimental results are shown and discussed. Finally in Section V, the conclusions of the work are presented.

II. Related Work

Today, ANNs are quite popular in the ML world due to the increased amount of data and computing power available. These two factors prevented researchers from using them altogether in academic settings. Now with faster computer-processing, ANNs can be found in image processing, speech recognition and other areas where large amounts of data are available.

ANNs can build effective end-to-end ML systems and can be used in EDA for modeling [6], synthesis [7], layout generation [8] or even fault testing [9].

In [10], a neural network-based methodology is used to estimate the performance parameters of CMOS Operational Amplifiers (Amp-Op) topologies. Training data of the model was directly generated through SPICE simulations to provide accurate and reliable data to the system. The execution time using ANN models was about 10 sec for each configuration, totaling 80 sec for all configurations, which represents a speed-up factor of 2000.

In [11], an ANN with two hidden layers is used to replace a SPICE simulator. A rough POF can be found in a reasonable time with multi-objective optimization (MOO), but a high-quality one requires a lot of simulator iterations which results in long synthesis times. After a MOO phase to obtain a low-quality pareto optimal fronts (POF), the process switches to a faster single-objective optimization (SOO) to complete the POF making it smoother and more continuous. At this phase the SPICE simulator was also replaced by an ANN which reduced the synthesis time even further. The training data for the ANN was the data obtained in the MOO phase.

In [12], a similar method is used to accelerate a simulation-based circuit synthesizer through the use of ANNs to determine circuit performances instead of a SPICE simulator. Instead of training the ANN with simulation data beforehand and simply replacing the simulator with the trained ANN, the simulation-based synthesizer is left unchanged for some generations of the optimization loop and only after the ANN replaces the SPICE simulator. Unlike other conventional algorithms, all the data generated in the first phase is used as training data for the ANN instead of being discarded. The main innovation of this approach is that there is no separate data acquisition step to train the ANNs used therefore makes it possible be used for every new topology without loss of generality for all analog circuits.

In [13] ANNs are used to improve the sample efficiency for several large circuits regarding their post-layout performance parameters. ANNs are used as an oracle, where, given two different circuit sizing solutions, the ANN will predict which design performs better for each individual parameter, requiring a sub-ANN for each parameter. This discriminator achieves at least two orders of magnitude in sample efficiency which represents a big reduction in number of simulations required.

All these approaches to reduce the execution time of optimization-based sizing use ANNs to replace or complement the circuit simulator. The execution time is greatly reduced by avoiding time-consuming circuit simulations, however, to recover the accuracy lost, in [10][14] at later stages of the optimization the circuit simulator is reestablished. Furthermore, the ANN models are trained over the entire design space, which spends valuable resources modelling and evaluating large regions of unusable design combinations. In [15] the ANNs were also trained to replace the simulator, but the previous issue is somewhat addressed by applying data mining techniques to build a model that capture only significant regions of the performance space visited during automatic synthesis.

A brief overview of the works previously discussed is presented in Table I.

In this work, the popular and powerful usage of an ANN to enhance the optimization-based sizing by complementing the circuit simulator will be applied. This method will be used in circuit topologies more complex than the ones previously discussed in the literature, and, despite the marginal loss of accuracy when compared with the circuit simulator, the speed-up gained using this approach will surely boost the optimization performance.

TABLE I. Speeding-up simulation-based sizing with ANNs Overview

Reference	Speed-up factor (up to)	Maximum Error	Number of Layers	Method
G. Wolfe, 2003 [10]	≈2000	60%	3	Complement/ Replace simulator
T. O. Çakıcı, 2020 [11]	29.7	1.55%	3	Replace simulator
G. İslamoğlu, 2019 [12]	2.8	0.77%	4 & 5	Semi-replace simulator
K. Hakhamaneshi, 2019 [13]	n/r	n/r	4	Replace simulator
G. Alpaydin, 2003 [14]	n/r	n/r	3	Complement simulator
Hongzhou Liu, 2002 [15]	n/r	≈10000%	3	Replace simulator

n/r – not reported

III. PVT Estimator

Considering the main objectives of this work and the related work presented and discussed, it is proposed the elaboration of a PVT corner performance estimator using deep ANNs, complementing the circuit simulator to be used during sizing optimization. The ANNs will receive as input the performance figures respective to the typical conditions, i.e., TT conditions, obtained via accurate circuit simulation and the candidate circuit sizing solutions, and will predict the performances for the remaining PVT corners. In the optimization-loop of the simulation-based sizing, the PVT estimator will be located after the circuit simulator, as depicted in Fig. 1.

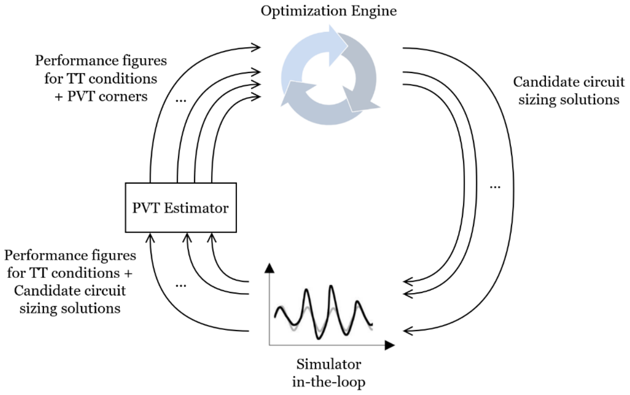
Optimization Engine
Performance figures for TT conditions + PVT corners
...
Candidate circuit sizing solutions
...
PVT Estimator
Performance figures for TT conditions + Candidate circuit sizing solutions
...
Simulator in-the-loop

Fig. 1. Location of the PVT estimator in the AIDA optimization loop

A. Case Study

The development of the proposed tool will take part in the sizing of a complex dual-mode class C/D VCO presented in [3], which is presented in Fig. 2. In that work, instead of achieving the desired performance parameters with sequential SOOs, a single many-objective sizing optimization, described as "everything at once" optimization, is proposed to achieve the best performance boundaries while finding the optimal tradeoffs. The circuit simulator performed a multi-corner analysis and the optimization followed a worst-case corner criteria on top of a worst-case tuning range optimization, taking into account two different tuning modes, b_{0000} and b_{1111}. The results pushed the circuit to its performance limits, reducing to almost half of the power consumption of the original design and showed its potential for ultralow-power with more than 93% reduction.

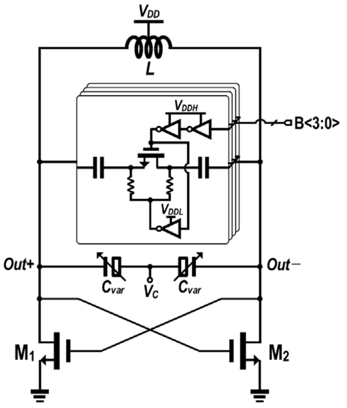

Fig. 2. Dual-mode class-C/D VCO schematic. Reprinted from [3].

In the optimizations carried, there were 28 optimization variables that affect the sizing of 43 devices. The full list can be found in Table II.

TABLE II. Optimization Variables

Variable	Units	Min.	Grid	Max.
ind_radius	μm	15	5	90
ind_nturns	-	1	1	6
ind_spacing	μm	2	1	4
ind_width	μm	3	1	30
mccl, m1l	nm	60	20	240
mccw, m1w	μm	0.6	0.2	6
mccnf, m1nf	-	1	1	32
mccm	-	1	1	100
moscapw	μm	0.4	0.2	3.2
moscapl	μm	0.2	0.2	3.2
mimvw, mimvl, mim1w	μm	2	0.2	20
r1l, r2l, r3l, r4l	μm	1	0.2	10
r1m, r2m, r3m, r4m	-	1	1	20
nfn1, nfn2, nfp1, nfp2	-	1	1	100

A total of 18 performances were considered and three optimizations were performed with populations of 512 elements optimized for 1000 generations. Of all the sizing solutions, the POFs of the three optimizations provided, in total, 769 optimal sizing solutions. Each optimization took approximately 100 hours to complete in an Intel-Xeon CPU E5-2630-v3@2.40 GHz with 64 GB of RAM workstation using eight cores for parallel evaluation, resulting in 300 hours total, i.e., more than 12 days. Once again, the main goal of the proposed PVT Estimator in this work will be to reduce this execution time to an acceptable range.

B. Dataset

The source of the dataset will be simulated performances and associated sizing parameters generated by an optimization-based sizing of the circuit in Section III.A. In total, 9 different testbench variations will be considered (TT, FF, FS, SF, SS, 300mV, 400mV, m40dC and 85dC) that produce 10 different performance figures each, and, due to the worst-case tuning range optimization (two tuning modes are evaluated, $b_{0000}$and b_{1111}), each sizing must be simulated 18 times, providing a total of 180 simulated performance figures. The full list of testbench variations can be seen in Table III and the list of performances in Table IV. The principal objective of this optimization is to minimize both power and phase noise at 10 MHz in both tuning modes while imposing value constraints on 7 measured performances, in both tuning modes as well. These optimization constraints and objectives are shown in Table V.

TABLE III. List of testbench variations: TT and PVT corners

Name	Process	Voltage	Temperature
TT	TT	0.35 V	25ºC
FF	FF	0.35 V	25ºC
FS	FS	0.35 V	25ºC
SF	SF	0.35 V	25ºC
SS	SS	0.35 V	25ºC
300mV	TT	0.3 V	25ºC
400mV	TT	0.4 V	25ºC
m40dC	TT	0.35 V	-40ºC
85dC	TT	0.35 V	85ºC

The dataset will have a total number of 48 features, where 28 of them are the optimization variables plus 20 performance figures of the simulation in TT conditions in two different modes, i.e., b_{0000} and b_{1111}, and a total of 160 labels, i.e., the performance figures of the remaining corner variations in two different tuning modes.

TABLE IV. PERFORMANCES CONSIDERED FOR TT AND PVT CORNERS

Measure	Units	Description
f_{osc}	GHz	Oscillation frequency
PN@10kHz	dBc/Hz	Phase noise at 10KHz
PN@100kHz	dBc/Hz	Phase noise at 100KHz
PN@1MHz	dBc/Hz	Phase noise at 1MHz
PN@10MHz	dBc/Hz	Phase noise at 10MHz
power	mW	Power consumption
FOM@10kHz	dBc/Hz	Figure-of-merit at 10KHz
FOM@100kHz	dBc/Hz	Figure-of-merit at 100KHz
FOM@1MHz	dBc/Hz	Figure-of-merit at 1MHz
FOM@10MHz	dBc/Hz	Figure-of-merit at 10MHz

TABLE V. OPTIMIZATION CONSTRAINTS AND OBJECTIVES

Tuning mode	Measure	Units	Optimization Constraint	Optimization Objective
b_{0000}	f_{osc}	GHz	≥ 4.8	
	PN@10kHz	dBc/Hz	≤ -49	
	PN@100kHz	dBc/Hz	≤ -76	
	PN@1MHz	dBc/Hz	≤ -98	
	PN@10MHz	dBc/Hz	≤ -119	minimize
	power	mW	n/d	minimize
	FOM@10MHz	dBc/Hz	≥ 180	
b_{1111}	f_{osc}	GHz	≤ 3.9	
	PN@10kHz	dBc/Hz	≤ -55	
	PN@100kHz	dBc/Hz	≤ -82	
	PN@1MHz	dBc/Hz	≤ -103	
	PN@10MHz	dBc/Hz	≤ -124	minimize
	power	mW	n/d	minimize
	FOM@10MHz	dBc/Hz	≥ 180	

C. *ANNs Structure*

The structure of the ANNs used will be fully connected. Each ANN will estimate the performance figures of a specific corner for a specific tuning mode, so the output layer will have 10 neurons. Each ANN will receive as inputs, the sizing of the circuit, which consists of 28 optimization variables, and 10 performance figures of the simulation in TT conditions for its corresponding tuning mode, which means that the input layer will have a total 38 neurons. The number of hidden layers and number of nodes per hidden layer will be determined in the tuning phase. The structure of the ANN implemented for corner FF and tuning mode b_{0000} is shown in Fig. 3.

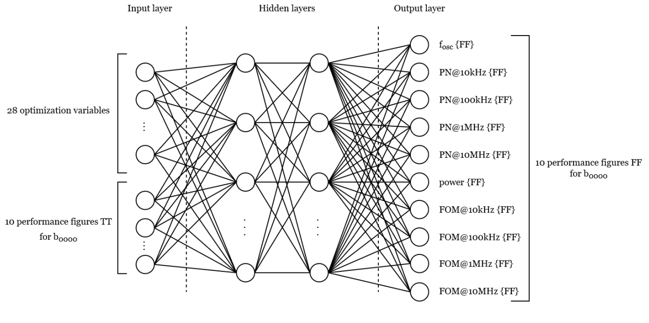

Fig. 3. ANN structure for corner FF and tuning mode b_{0000}

D. *Tuning Phase*

Throughout this part of the work the language used to program the ANN was Python, using both Tensorflow [16] and Keras [17] as ML libraries.

The starting point of the ANN architecture is the one described in Section III.C, where the output layer contains 10 nodes, one for each performance parameter of a certain combination of tuning mode and PVT corner variation. The optimization was performed for 9 testbench variations which 8 represent the PVT corners, and 2 tuning modes, and thus, in total, 16 different ANNs will be required.

The dataset contains 92115 data entries composed by, as described in Section III.B, 48 features where 28 represent the optimization variables and the other 20 represent the TT performance figures, 10 for each tuning mode. As for labels, the dataset contains 160 performance figures of the remaining PVT corner variations in two different tuning modes.

For each different ANN, it is only needed the performance figures of one combination of corner variations and tuning mode, so firstly the dataset had to be divided in 16 different datasets where each dataset represents a different corner combination. To increase model accuracy, only the TT performance figures that represent the same tuning mode as the labels are kept, so the final dataset structure only contains 38 features and 10 labels.

Some data entries have *null* values on the features and/or labels, representing sizing solutions that the simulator couldn't produce a meaningful performance figure. These entries had to be removed from each dataset to provide the best possible data to the ANNs along with duplicated rows.

Finally, the outliers present in each dataset must be removed. To do this, for each performance figure, the 1% lowest and highest values were cut from the dataset, alongside their entire row of data. The final sizes of the dataset for each PTV corner ANN can be found in Table VI.

TABLE VI. DIMENSIONS OF DATASETS FOR THE TRAINING OF THE DIFFERENT ANNS

Corner	Tuning mode			
	b_{0000}	% total	b_{1111}	% total
FF	81377	88.34	79300	86.09
FS	81842	88.85	81200	88.15
SF	82247	89.29	76794	83.37
SS	73456	79.74	48742	52.91
300mV	77608	84.25	69017	74.92
400mV	81865	88.87	81342	88.30
m40dC	81182	88.13	77103	83.70
85dC	80707	87.62	82028	89.05

Finally, all the 16 datasets were randomized, and split into two datasets:

- Training dataset: 90% of the original dataset;
- Test dataset: Remaining 10% of the original dataset.

The training dataset will be used to train the ANNs while the test dataset will be used to test the models. As for the tuning of the ANN architecture phase, the tuning parameters phase was only performed in the ANN regarding the corner FF with tuning mode b_{0000}.

With the dataset ready, the ANN's tuning parameter phase starts. As evaluation methods for the ANN's training, three following metrics were used. Mean Squared Error (MSE), defined as:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{1}$$

Mean Absolute Error (MAE), defined by:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - x_i| \tag{2}$$

And finally, Mean Absolute Percentage Error (MAPE), defined by:

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \tag{3}$$

Due to the nature of the values that are trying to be predicted with this ANN, the error between predicted value and actual value must be small. Because of this, an error value lower than 1% (for the case of the MAPE) was chosen as a reasonable target to achieve for the accuracy of the ANN.

Despite having three different metrics as evaluation methods, the loss function of the ANN throughout the tuning phase was the MSE because this is one of the most popular loss functions in regression problems like the one addressed in this work. A similar approach was used to choose the optimizer, with Adam being pointed as one of the most popular methods.

Regarding the batch size of the training phase, 128 was chosen as an appropriate value considering the size of the training dataset used.

As for data normalization, normalization for a range of 1 to 2 was used. This range was chosen, over the typical 0 to 1, due to its influence in MAPE values, making them explode in value due to values close to 0 in the denominator of the MAPE formula in (3).

The first hyperparameters to be tuned were: number of layers, number of neurons per layer, and learning rate. To determine the adequate number of layers for the model, two studies were made, the first one using 2 hidden layers for the model, and the second 3 hidden layers.

The parameter values considered for the 2 hidden layers study were: 200, 320, 440 and 560 for the size of hidden layer 1; 200, 300, 400 and 500 for the size of hidden layer 2; and 0.00005, 0.0001, 0.0005 and 0.001 for the learning rate. The different combinations of these three parameters were studied and the lowest values were obtained for: 440 neurons in hidden layer 1, 400 neurons in hidden layer 2 and 0.0001 for the learning rate.

In the 3 hidden layers study, the learning rate was set to 0.0001 considering it was the best value of the previous study. The parameter values considered were: 200, 320 and 440 for the size of hidden layer 1; 200, 300 and 400 for the size of hidden layer 2; and 200, 300 and 400 for the size of hidden layer 3. The best error results with 3 hidden layers were 18% to 30% higher than the best results with 2 hidden layers. Considering this fact there was no need to increase the number of hidden layers of the ANN, so no further study was required. The best parameter values of the 2 hidden layer study will be used in the final model.

The next parameter to be tuned was the activation function of the neurons located in the hidden layers. Four different activation functions were considered: sigmoid function, ReLU function [18], leaky ReLU function [19] and ELU function [20]. The activation function that generates the lowest error in almost all metrics was the ReLU function, so this function will be used in the final model.

The last parameter to be tuned was the dropout rate and the different values considered were: 0%, 5%, 10%, 20% and 30%. Dropout rates of 5%, 10% and 20% showed the best and relatively similar metric values between them while the values for dropout rate of 0% and 30% show a decrease in accuracy of the ANN. The worst results came from having no dropout rate at all which reveals the necessity of this regularization technique. A dropout rate of 20% was chosen for the final model, given that the ANN presents the best results with this value.

E. Final Model

All the parameters of the ANN are now tuned to achieve the best performance possible and a brief summary of the final model is shown in Table VII along with the metrics of the training phase at each epoch in Fig. 4 and Fig. 5.

TABLE VII. SUMMARY OF ANN

Parameter	Value
Input layer size	38
Hidden layer 1 size	440
Hidden layer 2 size	400
Output layer size	10
Loss function	Mean Squared Error
Optimizer	Adam
Batch size	128
Hidden layers activation function	ReLU
Learning rate	0.0001
Dropout rate	20%
Training epochs	300
Validation split	20%

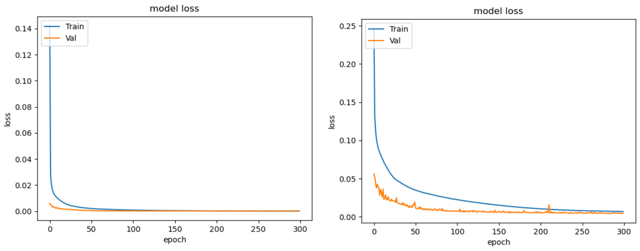

Fig. 4. MSE loss function (left) and MAE loss function (right)

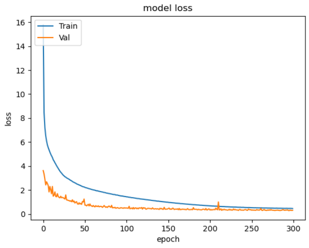

Fig. 5. MAPE loss function

The final values of the 3 metrics for all 16 ANNs are shown in Table VIII. As can be seen by the MAPE values of both training and test loss, the final model in all ANNs is presenting better performance than the initial goal of 1% error.

TABLE VIII. FINAL MODEL METRIC VALUES

Tuning mode	**Corner**	**Training loss**			**Test loss**		
		MSE ($\times10^{-4}$)	**MAE ($\times10^{-3}$)**	**MAPE**	**MSE ($\times10^{-4}$)**	**MAE ($\times10^{-3}$)**	**MAPE**
b_{0000}	**FF**	0.5346	3.5950	0.2344	1.5345	4.4429	0.2919
	FS	0.5046	3.7093	0.2438	1.5623	4.3444	0.2873
	SF	0.7096	3.7298	0.2441	1.2777	4.2840	0.2810
	SS	0.9830	4.3566	0.2833	1.6931	4.9340	0.3236
	300mV	0.7763	5.3615	0.3524	1.5838	5.9087	0.3919
	400mV	0.6531	4.5662	0.3061	1.0625	4.9738	0.3339
	m40dC	0.6195	4.0734	0.2665	1.1569	4.3965	0.2890
	85dC	0.6881	4.4254	0.2965	1.1435	4.8061	0.3237
b_{1111}	**FF**	1.2872	5.4180	0.3638	2.3003	5.9296	0.3987
	FS	0.7244	4.0943	0.2727	1.7098	4.7493	0.3163
	SF	0.8058	4.3929	0.2858	1.7478	5.0881	0.3324
	SS	1.1053	7.8992	0.5777	2.8484	11.5884	0.8384
	300mV	1.8979	7.0264	0.4585	3.4236	8.1042	0.5304
	400mV	1.7348	7.4566	0.4828	3.1930	8.4154	0.5486
	m40dC	5.9536	9.3559	0.6163	8.4010	10.8045	0.7134
	85dC	1.0614	4.5612	0.3037	1.7748	5.1120	0.3418

F. Controlled PVT Estimator

With the ANNs for each corner and tuning mode tuned and ready to be used, the next step of this work is to integrate the PVT estimator into the AIDA loop. The location of the PVT estimator is presented again in Fig. 6.

In the first step of the optimization loop several circuit sizing solutions are proposed by the optimization engine (number of solutions depends on the population defined). In the original AIDA loop, the simulator evaluates each of these solutions for all TT conditions and PVT corners, and outputs all the evaluated performances. The optimization engine receives these evaluations and ranks the solutions (population) according to their compliance with the objectives and constraints set for the current optimization problem.

In the modified AIDA loop, the simulator will only need to evaluate the solutions for TT conditions. Each of the ANNs will receive, as input, the sizing solution and, according to its tuning mode, the performance figures respective to the TT conditions (previously evaluated by the simulator). With the inputs defined, each ANN will output the performance figures corresponding to a specific corner and tuning mode, so the performance figures for all PVT corners can be sent to the optimization engine for further ranking. These performance figures are a mix of simulated performance figures (TT corners) and predicted performance figures (remaining PVT corners).

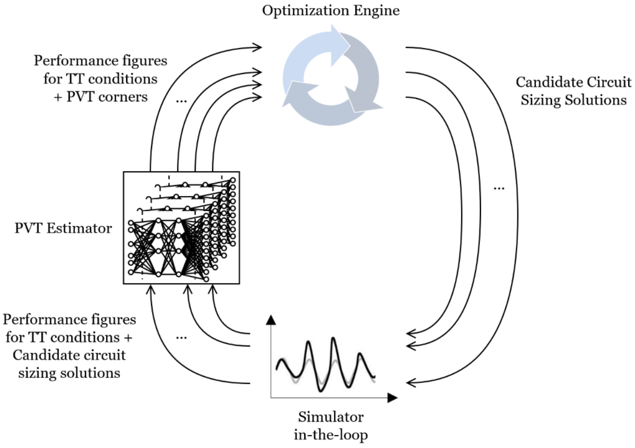

Fig. 6. Location of the PVT estimator in the AIDA optimization loop

When complementing the simulator with a fast estimation model, the optimization process may be erroneously guided to unrealistic design space regions, and, therefore, the tradeoff of when to predict or simulate candidate sizing solutions is delicate. Therefore, some control has to be introduced in the loop to guide the optimization to feasible solution regions. In order to achieve this, a simple error controller for each ANN used in the modified loop was implemented. A brief flowchart of one generation of the modified loop with the new controller is shown in Fig. 7.

At each generation, before both candidate sizing solutions and TT performance figures are sent to the ANNs, first they pass through a controller that will choose which ANNs will operate at the current generation, i.e., which PVT corner/tuning mode combination will be simulated or predicted.

First, the controller sends 20% of the candidate sizing solutions to be simulated and predicted at the same time. For PVT corner/tuning mode combination, with the output of the simulator, the controller checks if there are more feasible solutions than unfeasible solutions. If there are more unfeasible solutions than feasible solutions or the same number, the PVT corner/tuning mode combination will be simulated (instead of predicted) in that generation for the remaining candidate solutions. This acts as the primary filter to prevent the optimization of entering unfeasible regions.

If there are more feasible solutions than unfeasible it passes to the next step. Here the controller uses the output of the feasible solutions from the simulator, i.e., simulated performances (from the previous step) and compares them to the corresponding predicted performances. The error between each performance is calculated and the average error of all points is obtained. If the error (MAPE) is higher than 5% the combination will be simulated in that generation for the rest of the candidate solutions. If is equal or lower than 5% the corresponding ANN will predict the PVT corner/tuning mode combination in that generation for the rest of the candidate solutions.

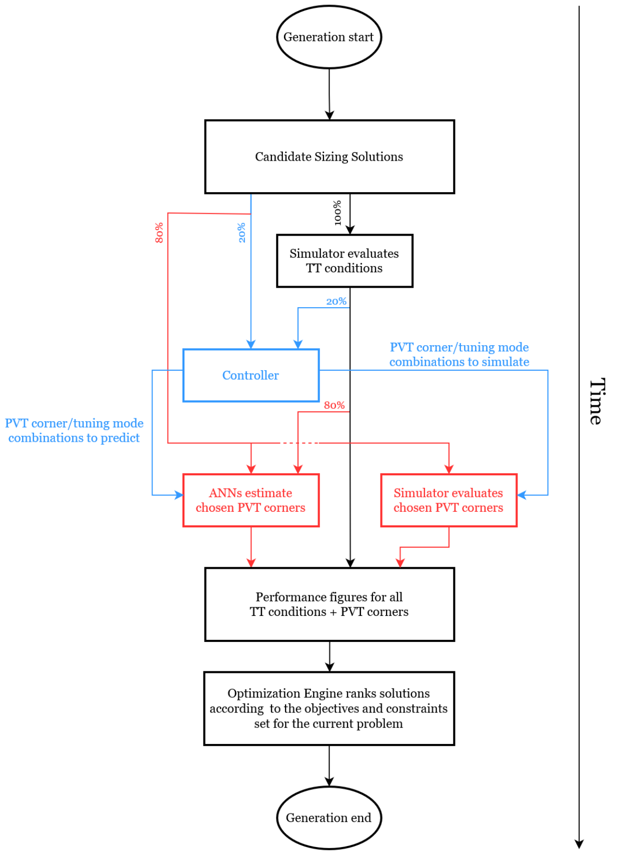
Generation start
Candidate Sizing Solutions
100%
80%
20%
Simulator evaluates TT conditions
20%
Controller
PVT corner/tuning mode combinations to simulate
PVT corner/tuning mode combinations to predict
80%
ANNs estimate chosen PVT corners
Simulator evaluates chosen PVT corners
Performance figures for all TT conditions + PVT corners
Optimization Engine ranks solutions according to the objectives and constraints set for the current problem
Generation end
Time

Fig. 7. Flow of a generation of the controlled modified AIDA loop (modification in red and controller in blue)

IV. Results

A. Class-C/D VCO from 3.9-to-4.8 GHz

An optimization problem with the objectives and constraints of Table V was set using the simulation-based sizing tool from [5] enhanced with the controlled PVT Estimator (CPVTE). As in the original optimization, the random population was set to 256 elements and the number of generations to 330. The speed-up obtained with the CPVTE is computed using the percentage of usage of each ANN throughout the optimization. Fig. 8 shows that 78.5% of the PVT performances were predicted instead of simulated. The PVT corners that were mostly simulated correspond to those less represented in the dataset (Table VI), e.g., SS and 300mV in b1111. By the end of this process, the POF has 20 solutions, as shown in Fig. 9. To attest their feasibility, these were fully simulated, resulting in one feasible solution and two additional solutions with only 1 out of 160 specifications violated, showing that the CPVTE could direct the sizing loop to feasible performance regions.

Additionally, 20 generations were performed without the CPVTE, ensuring that all solutions have simulator-grade accuracy, producing a final POF of 30 solutions. These fronts are superimposed in Fig. 9, alongside the optimization of Section

III.B. While the optimization with CPVTE did not reach the same power values as traditional simulation-based, it dominates in terms of PN, with almost 2-dBc/Hz improvement. By the end of the 350th generation, the total speed-up factor obtained using the CPVTE is 2.92×, more than 16 days of computational effort, while still achieving highly competitive sizing solutions.

B. Plug-and-play from 2.3-to-2.5 GHz

To test its plug-and-play functionalities, the CPVTE previously trained for the dataset of Section III.B is now applied in an optimization with a distant set of targets. The specifications of Table V were changed to fosc[b1111] ≥ 2.5 GHz, and fosc[b0000] ≤ 2.3 GHz, and the PNs and FOMs in all carriers were tightened by 5-dBc/Hz. The population size was kept to 256, the number of generations to 180 with CPVTE plus 20 without it. The speed-up obtained with the CPVTE is shown in Fig. 10, where 74.5% of the PVT performances were predicted. It is essential to highlight that the ANNs were trained in a dataset mainly composed of sizing solutions oscillating between 3.9-to-4.8-GHz. Still, the two control phases decide to bypass approximately every 3 in 4 simulations. By the end of this process, the POF has 5 solutions, as shown in Fig. 11. After complete simulation, two of those solutions are borderline feasibility, with only 2 out of 160 specifications marginally violated, proving that the optimization was guided to a good performance space.

After the last 20 generations without CPVTE, the optimization found 5 feasible solutions, shown in Fig. 11. To benchmark these results, a traditional simulation-based optimization was carried for the same population size and number generations, taking approximately 350 hours, and producing 13 sizing solutions, which are superimposed in Fig. 11. As in the previous experiment, the optimization with CPVTE explored design space regions that led to improved PNs. This fact is reinforced when analyzing the solutions at the 120th, 160th and 180th generations that already show a trend towards the PN spectrum of the tradeoff. The total speed-up factor obtained using the CPVTE is 2.48×, almost 9 days of computational effort, with no ANN tuning or training required.

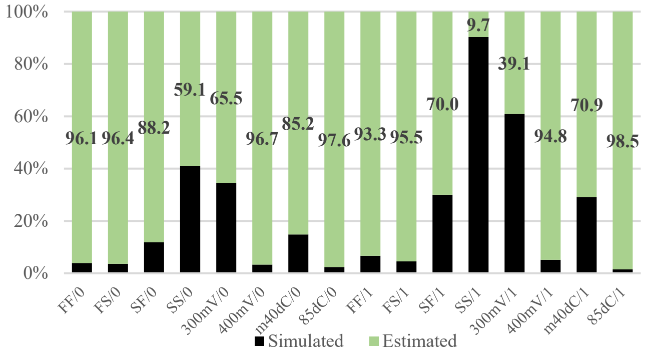

Fig. 8. Evaluations simulated *vs* estimated on the 3.9-to-4.8GHz optimization

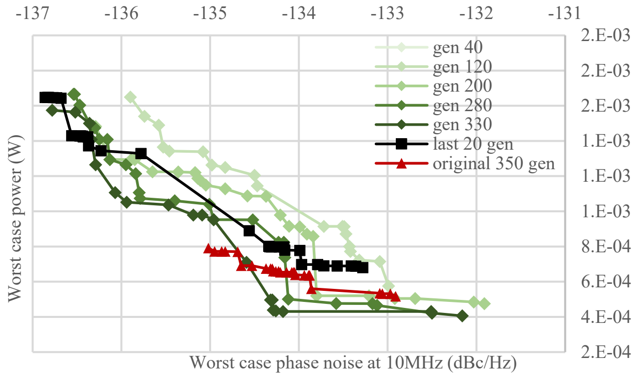

Fig. 9. Evolution of the POFs on the 3.9-to-4.8GHz optimization.

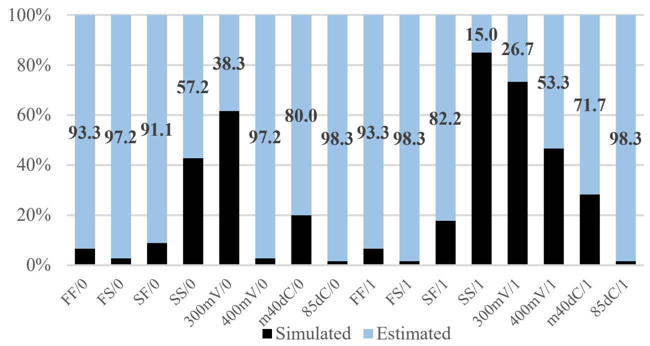

Fig. 10. Evaluations simulated *vs* estimated on the 2.3-to-2.5GHz optimization.

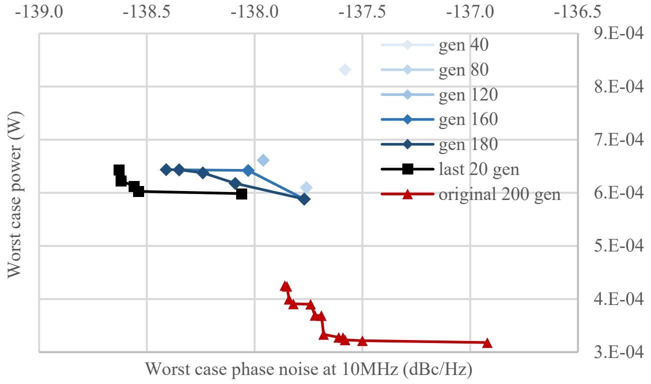

Fig. 11. Evolution of the POFs on the 2.3-to-2.5GHz optimization.

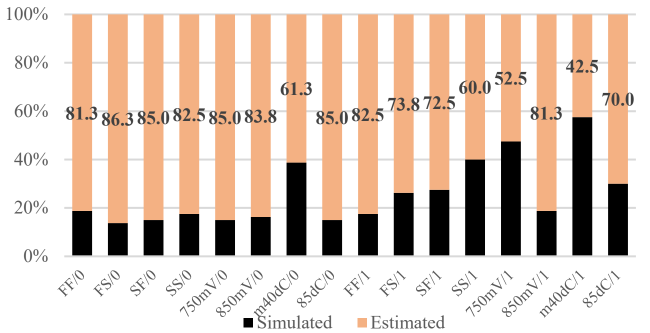

Fig. 12. Evaluations simulated vs estimated on the class B/C VCO optimization.

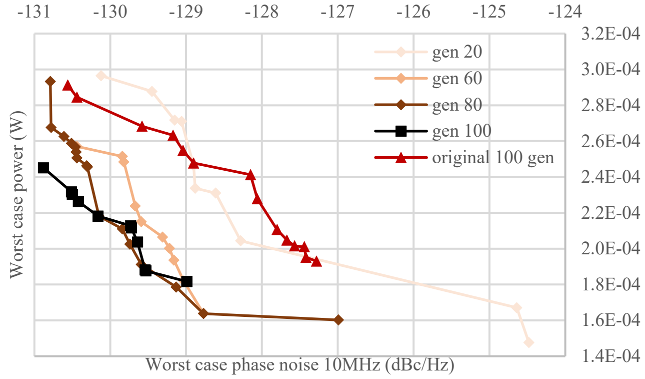

Fig. 13. Evolution of the POFs on the class B/C VCO optimization.

C. Plug-and-train Ultralow-Power Class B/C VCO

To test its plug-and-train functionalities, the CPVTE ANNs structure previously tuned for the dataset of Section III.B is now reused in the CPVTE ANNs' training phase for an optimization of a different VCO circuit topology, i.e., an ultralow-power Class B/C VCO. An optimization problem with the objectives and constraints detailed in [4] was set using the simulation-based sizing tool from [5] enhanced with the CPVTE. The population size was set to 256 and the number of generations to 80 with CPVTE plus 20 without it. The speed-up obtained with the CPVTE is shown in Fig. 12, where 74.1% of the PVT performances were predicted. By the end of this process, the POF has 16 solutions, as shown in Fig. 13. After complete simulation, five of those solutions are close to feasibility, with only 6 out of 160 specifications marginally violated, proving that the optimization was guided to a good performance space.

After the last 20 generations without CPVTE, the optimization found 12 feasible solutions, shown in Fig. 13. To benchmark these results, a traditional simulation-based optimization was carried for the same population size and number generations, taking approximately 636 hours and producing 14 sizing solutions, which are superimposed in Fig. 13. As in the previous experiments, the optimization with CPVTE explored design space regions that led to improved PNs, however, in this one, the optimization was capable of finding more competitive solutions in terms of power also. This fact is reinforced when analyzing the solutions at the 60th and 80th generations that already show better results than the original optimization. The total speed-up factor obtained using the CPVTE is 2.11×, almost 14 days of computational effort, with no ANN tuning required.

V. Conclusions

In this work, it is presented an approach towards the acceleration of analog/RF IC optimization-based sizing loop with the help of a PVT corner performance estimator, using multiple ANNs, to complement the simulation process, therefore reducing the simulator workload.

For the development of the PVT estimator, an optimization-based sizing of a Class C/D VCO for 3.9-to-4.8 GHz was used as case study, gathering the necessary data to train the ANNs and to ascertain if the results of the estimations, before integration in the optimization loop, were adequate. All ANNs showed that the estimation error results were adequate so the integration process and discussing of final results were performed.

Three different circuit optimizations were used to test the PVT estimator. The first one was the same optimization based-sizing of a class C/D VCO for 3.9-to-4.8 GHz used for the development of the PVT estimator. The PVT estimator reduced 78.5% of the simulator workload, lowering the total optimization run time by 16 and a half days (original run took 25 and a half days to complete). The final solution results showed similar performances to the original optimization, and therefore, proving that the PVT estimator is capable of finding adequate PVT corner performances. The second optimization was an optimization-based sizing of the same circuit topology as the previous one, although, the range in which the VCO operates was changed to 2.3-to-2.5 GHz and the optimization constraints were tightened. The PVT estimator reduced 74.5% of the simulator workload, lowering the total optimization run time by 8 and a half days (original run took 14 and a half days to complete). Feasible solutions were found at the end of the optimization using the PVT estimator, proving its capability of being reused for optimizations with completely different targets of the same circuit topology its ANNs were trained to, therefore demonstrating its plug-and-play functionalities. The third and final experiment was an optimization-based sizing of a different VCO circuit topology, i.e., an ultralow power class B/C VCO. The same structure of the ANNs used in the two previous tests, was reused to train the ANNs in this optimization. The PVT estimator reduced 74.1% of the simulator workload, lowering the total optimization run time by 14 and a half days (original run took 26 and a half days to complete). Feasible solutions with better performances than the original optimization, were found at the end of the optimization using the PVT estimator, proving the capability of its ANNs reusage for a different VCO circuit topology, therefore demonstrating its plug-and-train functionalities.

References

[1] Cadence. Accessed: Out. 12, 2021. [Online]. Available: http://www.cadence.com.

[2] MunEDA. Accessed: Out. 12, 2021. [Online]. Available: http://www.muneda.com.

[3] R. Martins, N. Lourenço, N. Horta, J. Yin, P. Mak and R. P. Martins, "Many-Objective Sizing Optimization of a Class-C/D VCO for Ultralow-Power IoT and Ultralow-Phase-Noise Cellular Applications," in *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 27, no. 1, pp. 69-82, Jan. 2019.

[4] R. Martins *et al*., "Design of a 4.2-to-5.1 GHz Ultralow-Power Complementary Class-B/C Hybrid-Mode VCO in 65-nm CMOS Fully Supported by EDA Tools," in *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 67, no. 11, pp. 3965-3977, Nov. 2020.

[5] R. Martins, N. Lourenço, S. Rodrigues, J. Guilherme and N. Horta, "AIDA: Automated analog IC design flow from circuit level to layout," *2012 International Conference on Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design (SMACD)*, Seville, 2012, pp. 29-32.

[6] A. Suissa and et. al., "Empirical method based on neural networks for analog power modeling," in *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 29, no. 5, pp. 839–844, April 2010.

[7] N. Kahraman and T. Yildirim, "Technology independent circuit sizing for fundamental analog circuits using artificial neural networks," in Ph.D. Research in Microelectronics and Electronics (PRIME). IEEE, pp. 1–4, 2008.

[8] K. Zhu and et. al., "Genius route: A new analog routing paradigm using generative neural network guidance," in *Proceedings of International Conference on Computer Aided Design (ICCAD)*, 2019.

[9] M. Andraud, H. Stratigopoulos, and E. Simeu, "One-Shot Non-Intrusive Calibration Against Process Variations for Analog/RF Circuits," IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 63, no. 11, pp. 2022–2035, Sept. 2016.

[10] G. Wolfe and R. Vemuri, "Extraction and use of neural network models in automated synthesis of operational amplifiers," in *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 22, no. 2, pp. 198-212, Feb. 2003.

[11] T. O. Çakıcı, G. İslamoğlu, Ş. N. Güzelhan, E. Afacan and G. Dündar, "Improving POF Quality in Multi Objective Optimization of Analog ICs via Deep Learning," *2020 European Conference on Circuit Theory and Design (ECCTD)*, Sofia, Bulgaria, pp. 1-4, 2020.

[12] G. İslamoğlu, T. O. Çakici, E. Afacan and G. Dündar, "Artificial Neural Network Assisted Analog IC Sizing Tool," *2019 16th International Conference on Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design (SMACD)*, Lausanne, Switzerland, pp. 9-12, 2019.

[13] K. Hakhamaneshi, N. Werblun, P. Abbeel and V. Stojanović, "BagNet: Berkeley Analog Generator with Layout Optimizer Boosted with Deep Neural Networks," *2019 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, Westminster, CO, USA, pp. 1-8, 2019.

[14] G. Alpaydin, S. Balkir and G. Dundar, "An evolutionary approach to automatic synthesis of high-performance analog integrated circuits," in *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 3, pp. 240-252, June 2003.

[15] Hongzhou Liu, A. Singhee, R. A. Rutenbar and L. R. Carley, "Remembrance of circuits past: macromodeling by data mining in large analog design spaces," *Proceedings 2002 Design Automation Conference (IEEE Cat. No.02CH37324)*, New Orleans, LA, USA, pp. 437-442, 2002.

[16] Tensorflow. Accessed: Out. 12, 2021. [Online]. Available: www.tensorflow.org

[17] Keras. Accessed: Out. 12, 2021. [Online]. Available: https://github.com/fchollet/keras

[18] Aurlien Gron. 2017. Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems (1st. ed.). O'Reilly Media, Inc. ISBN 978-1491962299.

[19] B. Xu, N. Wang, T. Chen and M. Li, "Empirical Evaluation of Rectified Activations in Convolutional Network ", *Cornell University arXiv:1505.00853*, November 2015.

[20] D. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs) ", *Cornell University arXiv:1511.07289,* November 2015.