Power, Performance and Area Modelling Techniques for System-on-Chips using Transfer Learning

Abhishek Sengupta
Master of Science in Communications Engineering
Technical University of Munich
Munich,Germany
ge79car.sengupta@tum.de

***Abstract*—Power,Performance and Area (PPA) are the most crucial factors to be considered while developing an System-On-Chip to improve performance, durability and reliability . This paper primarily focuses on an accurate PPA estimation using Transfer Learning where learning from previous Machine Learning Models is taken into consideration for accurate predictions. Transfer Learning helps in achieving a high prediction accuracy with a constrained number of training samples. A comparative study of the different models is also presented in this paper. The predictions are based on the RTL design parameters of the hardware system. An accuracy of up to 97% can be achieved with the proposed models.**

***Index Terms*—Power, Performance, Area, PPA, Machine Learning, Transfer Learning, System-On-Chip.**

I. Introduction

Power,Performance and Area of a System-On-Chip should be taken into consideration through all the design phases. Contemporary processing systems are embedded with diversified components like CPU's, GPU's and have various design alternatives like frequency, memory registers, number of cores. These factors need to be considered in order to achieve an optimum PPA in the final design. In classical approaches, the PPA of a hardware is measured at the end of the design process. This can lead to the measured PPA not complying with the expected PPA and eventually hamper the functionality of the hardware. Therefore, getting an estimate of the PPA at every design phase is crucial. Hence, Machine Learning models in combination with Transfer Learning can be used to predict the PPA of a particular hardware design stage based on the design parameters. The future design stages can be based on the results obtained and the design strategies can be modified to comply with the expected PPA. As demonstrated in [5], Transfer Learning helps us in predicting PPA with an accuracy of more than 98%.

The most extensive phase in designing PPA models is gathering data. In this paper, the concept of Transfer Learning is used to train new models by using the learning from previous models.In Transfer Learning, as demonstrated in [13], a base model is trained initially on a base dataset and task, and then the learned features are transferred to a second target model to be trained on a target dataset and task. Transfer Learning attains high accuracy if the features of the dataset of the base and target models are well related. Hence, the data needed for training new PPA models is significantly reduced since some learned features are transferred to the new model. The model can be trained on the PPA measurements that is obtained by running applications on test hardware and capturing the change in the PPA measurements on the variation of hardware parameters like clock frequency, memory capacity like RAM, number of processors.

Section II of this paper deals with the State of the Art of PPA Analysis using Machine Learning. Section III deals with the different Machine Learning models, where as Section IV dives into the parameters and model generation and test framework. Section V deals with the dataset and Section VI provides an insight into the experiments and results.The paper concludes with the Conclusion in Section VII and the References.

II. State of the art of PPA Analysis using Machine Learning

PPA Modelling is critical in every design phase of all of System-On-Chips. Different Machine Learning techniques have been used over time for PPA predictions. PPA predictions have been made at all design phases to develop an overall optimized circuit.

Previously Convolutional Neural Networks(CNN's) have been used in the prediction of the power consumed by a chip in [10]. Also, a Machine Learning framework has been developed to estimate the performance of an FPGA based on its design in [11] . Neural Networks have also been deployed to calculate PPA for hardware optimization in [7].

However, for these models to be highly efficient and achieve a higher prediction accuracy, a high amount of training data is required. Hence, in this paper, Transfer Learning models are demonstrated which achieve high accuracy with a constrained amount of data by sharing it's learned parameters to a second model. The Transfer learning models are built using Neural Networks and Gradient Boost Regressor as shown in [5] and Decision Trees as demonstrated in [12]. This paper also aims

to deliver a comparative study of the different models based on their degree of accuracy. The PPA models described in this paper attains an overall accuracy of 97%.

III. PPA Modelling with Machine Learning

Here, a few Machine Learning models like Neural Networks, Decision Trees and Gradient Boost Regressor that can be used for PPA Modelling are demonstrated.The focus is on using Transfer Learning, that is to re-use trained models for PPA predictions of different System-On-Chips. Later, a comparative study of the different models based on their degree of accuracy is presented in Section VI.

A. Neural Network

A Neural Network(NN) is a widely used Machine Learning Model for achieving a high degree of accuracy as explained in [3]. There exists different forms of NN such as Feed Forward Neural Networks, Convolutional Neural Networks, Recurrent Neural Networks etc. As shown in Fig. 1, The NN is a Feed Forward Neural Network which has an input layer, hidden layers and an output layer. Each layer contains many neurons which are mathematical functions that has learnable weights and biases and are connected with the neurons of the other layers. Each neuron is connected to a non-linear activation function as shown in [14] which is added in order to help the NN learn complex patterns in the data. The activation function helps in adding non-linearity to the data and also helps in restricting the output from a neuron to a certain limit. Training a Neural Network refers to finding the appropriate Weights of the Neural Connections by feedback loops called Gradient Backward Propagation as described in [15].

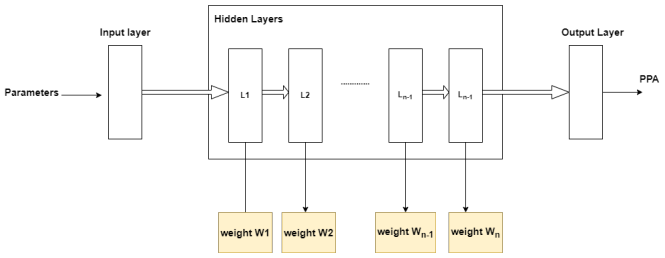

Fig. 1. Neural Network with no shared layers

A Neural Network can share the learning among various similar hardware problems because learning is in the form of weights. Hence when a NN is trained for a specific problem, some of the layers can be used to train another NN for a similar model. This paper demonstrates the use of Transfer Learning, where a base model with a shared set of weights, and a shared set of layers which are trained specifically for each unique hardware design problem. A Neural Network with a shared base model and a single trainable hidden layer for different hardware designs is shown in Fig 2.

In this way, the amount of samples and training time required for PPA predictions can be minimized. This concept of Transfer Learning where a base model is used is based on the fact that the internal structure of the hardware may be similar to some extent and so the data collected for training also has similar features to some extent.

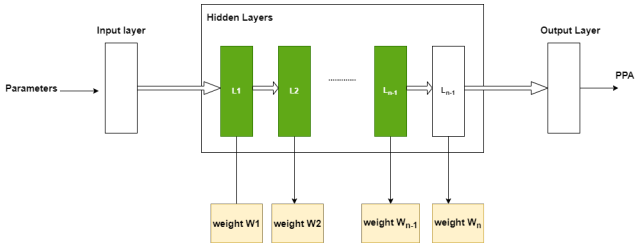

Fig. 2. Neural Network with the shared layers depicted in green

B. Decision Trees

Decision Tree, as shown in [2], is an algorithm that contains conditional control statements. It has a tree like structure that branches out from a root leaf node and forms many leaf nodes where decisions are made. In PPA estimation, a Decision Tree can make predictions based on conditional statements like if the number of cores in a processor is greater than a certain amount, or the instructions executed in a circuit in unit time is greater than a threshold. Decision Trees are a very reliable model of prediction based on internal calculations of Entropy and Information Gain.

Entropy is an Information Theory metric by which the amount of uncertainty in a group of observations is measured. Given a dataset with N classes, the Entropy can be measured as :

$$E = -\sum_{i=1}^{N} p_i log_2 p_i$$

where p_i is the probability of randomly selecting an example in class i.

Information Gain is a parameter that determines the quality of splitting. The Information Gain can be calculated as :

$$InformationGain = Entropy_{parent} - Entropy_{children}$$

The parent node can be defined as the node where splitting occurs and the children nodes are formed after splitting at the parent node. The less the entropy or uncertainty of the child, the more the Information Gain. Hence, to make accurate decisions the branch with the highest Information Gain is chosen.A previously built Decision Tree can be used as base model and new branches for prediction can be created on the base model as shown in [12] by the new set of training data that is obtained specific to a hardware problem.

C. Gradient Boost Regressor

Gradient Boosting, as explained in [5], works on the principle that the best possible next model when combined with the ensemble of the previous models, can lead to a better prediction accuracy. It leads to the minimization of the overall prediction errors.For implementing the Gradient Boosting Algorithm, a loss function needs to be optimized, a weak learning model to make predictions and an additive model to add to the

weak learning model in order to minimize the loss function. Hence, the Gradient Boost Regressor, iteratively combines the weak learners like Decision Trees into a strong learner.

IV. Model Parameters and Model Generation

A. Parameters used for PPA Modelling

The PPA Modelling of a System-On-Chip depends on the architectural design of the chip. Hence to calculate the PPA, the Register Transfer Level(RTL) Implementation, RTL synthesis parameters such as clock period, enable signals and circuit delay are taken into account. Based on these parameters at a particular design stage, the PPA can be predicted at that particular stage of hardware development.

B. PPA Model Generation Framework

A Framework to generate the PPA model is shown in Fig. 3. In the first stage, design data from RTL is obtained. A sampling space is generated using Latin Hypercube Sampling (LHS) which is a statistical method for generating random samples of data from a multidimensional distribution as described in [8]. Once the samples are generated, the RTL is synthesized to obtain the PPA estimation for all the combinations of parameters from the sample space. The parameters of the extracted PPA and the PPA estimation can be used as the training data to train the model and the final model is obtained.

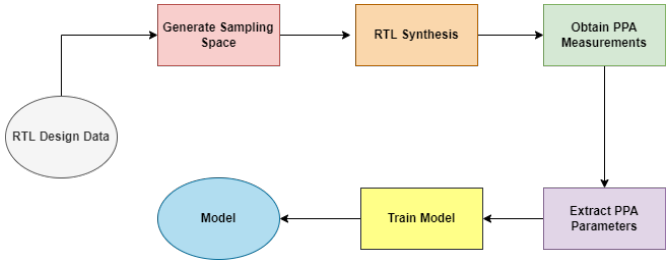

Fig. 3. PPA Model Generation Framework

C. PPA Model Testing Framework

A Framework to test the generated PPA model is being shown in Fig. 4. This framework determines whether the extracted PPA parameters of the test hardware are accurate enough for the hardware design to move to the next stage. Firstly, the RTL design data is taken and then the model generated by the PPA Model Generation Framework as shown in Fig. 3 is tested with the RTL data. If the accuracy target is met, the next stage of hardware development starts, and if it is not met the RTL design data of the current stage needs to be changed and the process needs to start afresh.

V. Dataset to create and evaluate models

The RTL Design data contains parameters like number of combinatorial and sequential blocks, registers, finite state machine's and clock signal measurements. The RTL synthesis is done for numerous combinations of these parameters and the PPA measurements are extracted for each of the synthesized RTL's. These PPA measurements serve as the training data

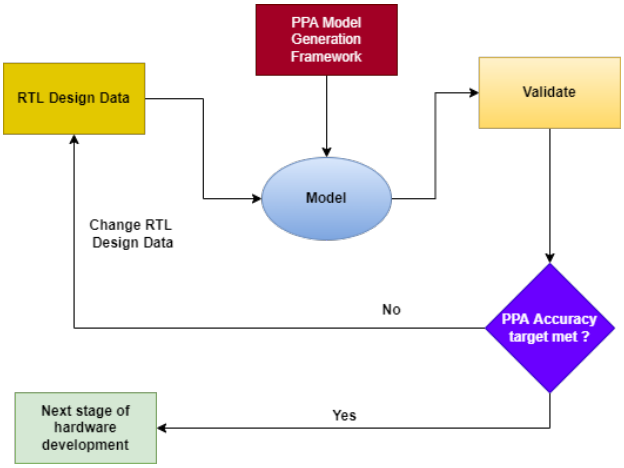

Fig. 4. PPA Model Testing Framework

in relation to that particular RTL design from where it was extracted. This data is used to train our model and later make PPA predictions based on a given RTL synthesis.

VI. Experiments and Results

A. Machine Learning Model Formulations for PPA Predictions

PPA Models are created with the RTL Design data. The models are created with 200 samples. 60% of the data is used as the training set, 20% used for validation and the remaining 20% as the test set. The model accuracy is measured in terms of Mean Absolute Error (MAE) which can be defined as the Arithmetic Average over the absolute errors. The MAE can be calculated as:

$$MAE = -\frac{100}{N}\sum_{i=1}^{N} abs(\frac{y - y_1}{y})$$

where y is the obtained value after PPA prediction and y_1 is the actual value from the test set.

TABLE I
MODELS EVALUATION

Accuracy Parameters	Machine Learning Model		
	Neural Network	*Decision Tree*	*GB*
Dynamic Power	96.21	96.42	96.75
Critical Path	97.13	96.64	96.91
Area	97.70	97.64	97.16
PPA	97.16	97.43	97.20

Table I shows the accuracy parameters that is obtained after implementing different Machine Learning Models. The Accuracy Parameters are calculated as:

$$AccuracyParameters = 100\% - MAE$$

It can be concluded that the Decision Tree has the highest PPA accuracy while the Neural Network with the shared layers has the least. However, all the three models have an overall PPA prediction of accuracy of more than 97%.

The Decision Tree has the highest prediction accuracy for Area and the least for Critical Path. While the Gradient Boost Regressor has a consistent prediction accuracy for all the accuracy parameters, the Neural Network has a prediction accuracy with Dynamic Power at 96.21% and Area at 97.70%.

B. Comparing the prediction accuracy of the Machine Learning Models with number of samples

A comparison of the PPA Prediction Accuracy and Training Samples for the different Transfer Learning models is shown in figure 5. The figure is based on the evaluation results obtained in [5]. The models are trained for different number of samples and then the Prediction Accuracy is obtained. Once the graph is obtained, the highest Prediction Accuracy for a least number of training samples can be inferred. This will help to reduce the amount of samples needed and is beneficial in case of practical chip design scenarios.

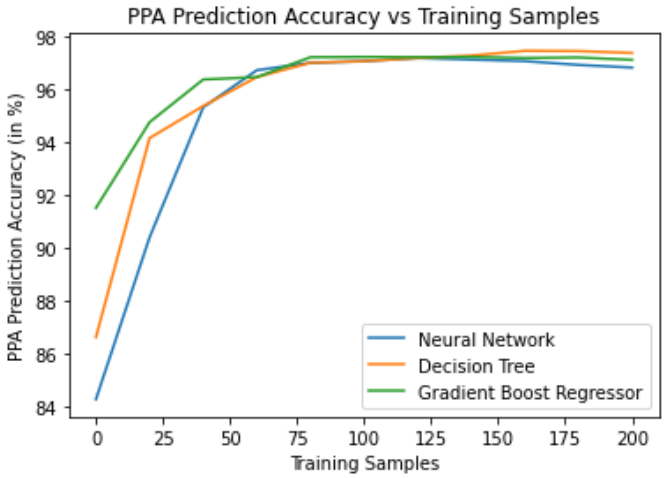

Fig. 5. PPA Prediction Accuracy vs Training Samples [5]

As can be seen from the Fig. 5 that the Gradient Boost Algorithm has a higher Prediction Accuracy than the other two models if the number of samples is close to 50.However, for a large number of training samples, the Decision Tree performs better than the Neural Network and Gradient Boost Regressor. By analyzing the curves, the optimum amount of training samples needed to make each model predict with the highest accuracy can be chosen.

VII. CONCLUSION

This work presented some Machine Learning models to predict the Power, Performance and Area of System-On-Chips hardware design. The models are successful in obtaining a high PPA prediction accuracy. Decision Trees turn out to be the best model with an accuracy of 97.43%. However, Neural Networks and Gradient Boost Regressor both perform well at an accuracy of 97.16% and 97.20% respectively.

Using Machine Learning models to predict the PPA at every design phase based on RTL parameters and hardware components, is better than the traditional approach of measuring the PPA at the end of the hardware design phase is beneficial economically in terms of resource usage. This paper also explores the possibility of using Transfer Learning in PPA predictions. Transfer Learning helps in reducing the amount of data that is needed to train our models and also helps in reducing time complexity by using pre-trained models. Also, Transfer Learning helps us in attaining higher prediction accuracy using a fewer number of new training samples. In case of Neural Networks, with the number of training samples nearly equal to 100, the prediction accuracy increases proportionally to the number of shared layers as shown in [5].

REFERENCES

[1] Y. Zhou, H. Ren, Y. Zhang, B. Keller, B. Khailany, and Z. Zhang, "Power inference using machine learning," in 2019 56th ACM/IEEE Design Automation Conference (DAC), 2019, pp. 16.

[2] S. van den Elzen and J. J. van Wijk, "BaobabView: Interactive construction and analysis of decision trees," 2011 IEEE Conference on Visual Analytics Science and Technology (VAST), 2011, pp. 151-160, doi: 10.1109/VAST.2011.6102453.

[3] P. D. Wasserman and T. Schwartz, "Neural networks. II. What are they and why is everybody so interested in them now?," in IEEE Expert, vol. 3, no. 1, pp. 10-15, Spring 1988, doi: 10.1109/64.2091.

[4] F. Last, M. Haeberlein, and U. Schlichtmann, Predicting memory compiler performance outputs using feed-forward neural networks, ACM Trans. Des. Autom. Electron. Syst., vol. 25, no. 5, Jul. 2020.

[5] W. R. Davis, P. Franzon, L. Francisco, B. Huggins and R. Jain, "Fast and Accurate PPA Modeling with Transfer Learning," 2021 IEEE/ACM International Conference On Computer Aided Design (ICCAD), 2021, pp. 1-8, doi: 10.1109/ICCAD51958.2021.9643533.

[6] A. Natekin and A. Knoll, Gradient boosting machines, a tutorial, Frontiers in neurorobotics, vol. 7, p. 21, 12 2013.

[7] J. Kwon and L. P. Carloni, Transfer learning for design-space exploration with high-level synthesis, in Proceedings of the 2020 ACM/IEEE Workshop on Machine Learning for CAD, ser. MLCAD 20. New York, NY, USA: Association for Computing Machinery, 2020, p. 163168.

[8] J.C. Helton, F.J. Davis, Latin hypercube sampling and the propagation of uncertainty in analyses of complex systems, Reliability Engineering System Safety, Volume 81, Issue 1, 2003, Pages 23-69, ISSN 0951-8320, https://doi.org/10.1016/S0951-8320(03)00058-9.

[9] J. L. Greathouse and G. H. Loh, Machine learning for performance and power modeling of heterogeneous systems, in 2018 IEEE/ACM International Conference on Computer-Aided Design (ICCAD), 2018, pp. 16.

[10] Y. Zhou, H. Ren, Y. Zhang, B. Keller, B. Khailany, and Z. Zhang, Primal: Power inference using machine learning, in 2019 56th ACM/IEEE Design Automation Conference (DAC), 2019, pp. 16.

[11] Z. Lin, J. Zhao, S. Sinha, and W. Zhang, Hl-pow: A learning-based power modeling framework for high-level synthesis, in 2020 25th Asia and South Pacific Design Automation Conference (ASP-DAC), 2020, pp. 574580.

[12] J. w. Lee and C. Giraud-Carrier, "Transfer Learning in Decision Trees," 2007 International Joint Conference on Neural Networks, 2007, pp. 726-731, doi: 10.1109/IJCNN.2007.4371047.

[13] S. J. Pan and Q. Yang, "A Survey on Transfer Learning," in IEEE Transactions on Knowledge and Data Engineering, vol. 22, no. 10, pp. 1345-1359, Oct. 2010, doi: 10.1109/TKDE.2009.191.

[14] Q. Liu and J. Wang, "A One-Layer Recurrent Neural Network With a Discontinuous Hard-Limiting Activation Function for Quadratic Programming," in IEEE Transactions on Neural Networks, vol. 19, no. 4, pp. 558-570, April 2008,doi:10.1109/TNN.2007.910736

[15] J. Leonard, M.A. Kramer, Improvement of the backpropagation algorithm for training neural networks,Computers Chemical Engineering,Volume 14, Issue 3, 1990,Pages 337-341,ISSN,0098-1354,https://doi.org/10.1016/0098-1354(90)87070-6.