Power, Performance and Area Modelling Techniques for System on Chips using Machine 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 Machine Learning algorithms and providing a comparative study of the different models. The predictions are based on the RTL design parameters of the hardware system. Some models are implemented with Transfer Learning to achieve a high accuracy with a constrained number of training samples. 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 countless design alternatives like frequency, memory registers, number of cores.These factors need to be considered at the earliest design phase so that these does not affect the final design. Hence, we use Machine Learning models at various stages of the design process to predict the PPA of that stage and the future design stages based on the hardware design at that stage. The ML model can be trained on the PPA measurements that we obtain 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. An approximation of the PPA at a preliminary design phase can assist us in optimizing the PPA of the chip. The paper also uses the concept of Transfer Learning in order to use a miniumum amount of data to train the model and hence save on both time and resources.

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 has been a very critical thing in the design 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, various Machine Learning frameworks have been developed to estimate the performance of an FPGA based on its design in [12] . Neural Networks have also been deployed to calculate PPA for hardware optimization in [7].

This paper demonstrates the generation of PPA models using Machine Learning algorithms and to delivers a comparative study of the different models based on their degree of accuracy. The paper also implements Transfer Learning to deal with constrained training samples.

III. Machine Learning Models for PPA Modelling

Here, the various Machine Learning algorithms that can be used for PPA Modelling are demonstrated. We focus on using Transfer Learning, that is to re-use trained models for PPA predictions of different System-On-Chips. Later, we go on to do a comparative study of the different models based on their degree of accuracy.

A. Neural Network

A Neural Network(NN) is one of the most widely used Machine Learning Models.As shown in Fig. 1, The NN consists of an input layer, hidden layers and an output layer. Each layer contains many neurons which are connected with the neurons of the other layers.Each connection has a weight and on being trained it learns from the data. Each layer has an activation function 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 restricted to a certain limit.

A Neural Network can share the learning among various similar hardware problems because learning is in the form of

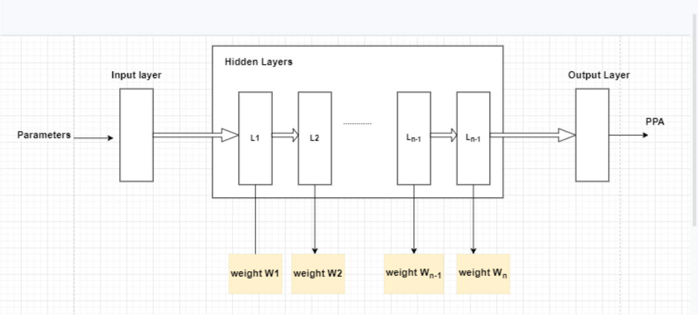

Fig. 1. Neural Network with no shared layers

weights. This is where the phenomenon of Transfer Learning comes into play. This paper demonstrates the design of a base model with the same set of weights, and a 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.

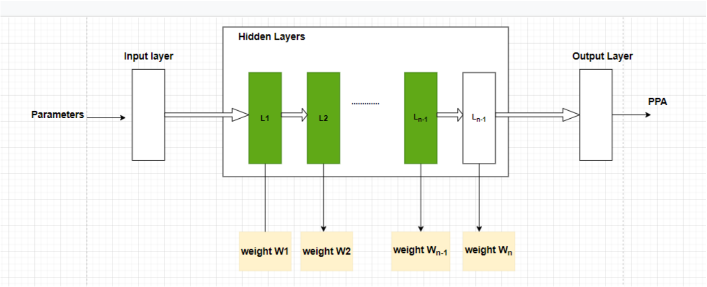

Fig. 2. Neural Network with the shared layers depicted in green

In this way, we can limit the amount of samples and training time required for PPA predictions.This phenomenon of Transfer Learning where we use a base model is based on the fact that the internal structure of the hardware may be similar to some extent.

B. Decision Trees

Decision Tree 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, 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 helps in determining 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 of the child, the more the Information Gain. Hence, to make accurate decisions we choose the branch with the highest Information Gain.

C. Gradient Boost Regressor

Gradient Boosting 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, we need a loss function which is 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 power, we take into account the Register Transfer Level(RTL) Implementation, RTL synthesis parameters such as clock period, enable signals, circuit delay. Based on these parameters at a particular design stage, we can predict the PPA at that particular stage and in the future design stages.

B. PPA Model Generation Framework

A Framework to generate the PPA model is shown in Fig. 3. In the first stage, we take up the design data from the RTL. Then a sample space is generated using a Latin Hyper-Cube(LHC). Once the samples are generated, we run the RTL Synthesis tool to obtain the PPA for all the combinations of parameters from the sample space. The parameters of the extracted PPA can be used in the training set to train the model and the final model is obtained.

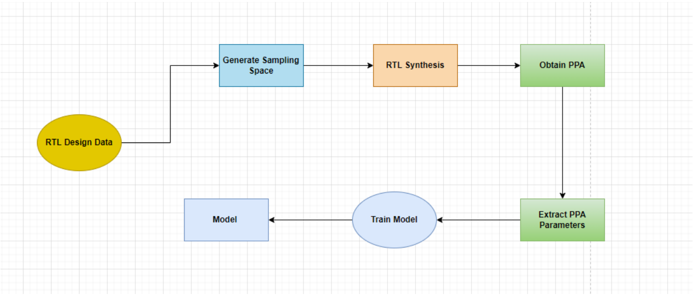

Fig. 3. PPA Model Generation Framework

C. PPA Model Testing Framework

A Framework to test the generated PPA model is being shown in Fig. 4. This model helps in determining whether the extracted PPA parameters of the test hardware are accurate enough for the hardware design to move to the next stage. Firstly, we take up the design data and perform the RTL Synthesis. Then we obtain the PPA and extract its parameters. If the accuracy target is met, we proceed to the next stage of the design process, and if it is not met the RTL design data needs to be changed and the process needs to start afresh.

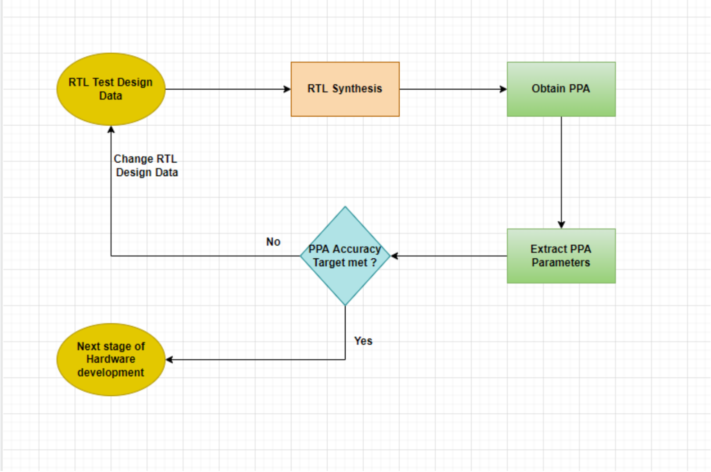

Fig. 4. PPA Model Testing Framework

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 in relation to that particular RTL design from where it was extracted. We use this data 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 without Transfer Learning

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

[a] Sample of a Table footnote.

Table I shows the accuracy parameters that we obtain after implementing different Machine Learning Models. The Accuracy Parameters are calculated as 100% - MAE. We can see 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%.

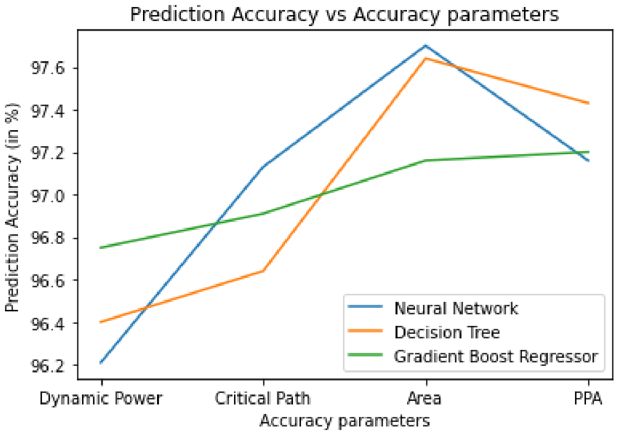

Fig. 5. PPA Prediction Accuracy vs Accuracy Parameters

A comparison of the PPA Prediction Accuracy and Accuracy Parameters for the different Machine Learning models is shown in Fig. 5. We can see that the Neural Network and the Gradient Boost Regressor have almost the same PPA Prediction Accuracy. 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 wide range of 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 without Transfer Learning.

A comparison of the PPA Prediction Accuracy and Training Samples for the different Machine Learning models is shown in Fig5. We wanted to train the models for a different number of samples and relate it to the Prediction Accuracy. Once we obtain the graph, we can infer the highest Prediction Accuracy for a least number of training samples. This will help to reduce the amount of samples we need and is beneficial in case of practical chip design scenarios.

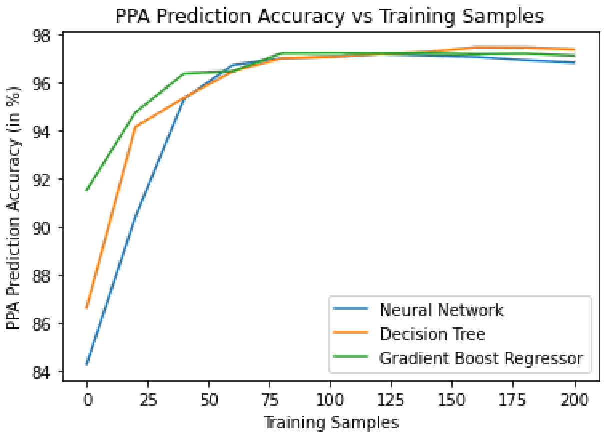

Fig. 6. PPA Prediction Accuracy vs Training Samples

We can see from the Fig. 6 that the Gradient Boost Algorithm has a higher Prediction Accuracy than the other two models if we have close to 50 samples.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, we can choose the optimum amount of training samples needed to make each model predict with the highest accuracy.

C. Machine Learning Model Predictions with Transfer Learning

In case of Neural Networks, this paper proposes to use Transfer Learning which means that the Neural Network can learn from previous models. The Neural Network a base model with the same set of weights, and a set of layers which are trained specifically for each unique hardware design problem. A comparison of the PPA Prediction Accuracy and the number of shared layers for different number of training samples is shown in Fig. 7. This effectively reduces the number of samples and training time needed by the Neural Network to predict with a high accuracy.

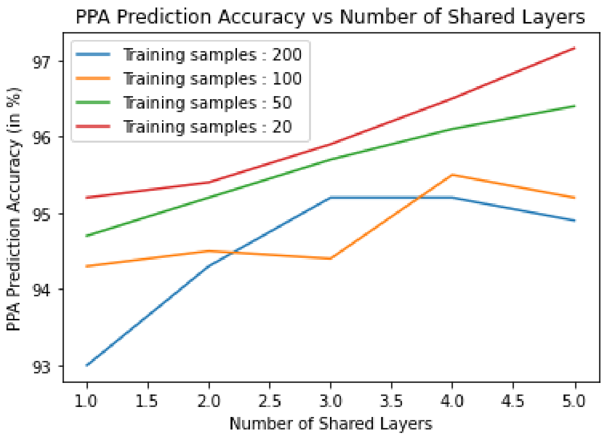

Fig. 7. PPA Prediction Accuracy vs Number of Shared Layers

We can see from the Fig. 7 that if the Number of Shared Layers is 5, the Neural Network predicts with the same prediction accuracy as it did with no shared layers, but the number of training samples is reduced to 20. Hence, the training time can be reduced without affecting the PPA prediction accuracy.

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 bnoth perform well at an accuracy of 97.16% and 97.20% respectively.

This paper also explores the possibility of using Transfer Learning in Neural Networks.So, a base model was designed with shared layers, and this model predicted with an accuracy of 97% with only 20 training samples and 5 shared layers.

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] G. Wu, J. L. Greathouse, A. Lyashevsky, N. Jayasena, and D. Chiou, “Gpgpu performance and power estimation using machine learning,” in 2015 IEEE 21st International Symposium on High Performance Computer Architecture (HPCA), 2015, pp. 564576.

[3] Y. Zhang, H. Ren, and B. Khailany, Grannite: Graph neural network inference for transferable power estimation, in Proceedings of the 57th ACM/EDAC/IEEE Design Automation Conference, ser. DAC 20. IEEE Press, 2020.

[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] D. Lee and A. Gerstlauer, Learning-based, fine-grain power modeling of system-level hardware ips, ACM Trans. Des. Autom. Electron. Syst., vol. 23, no. 3, Feb. 2018. [Online]. Available: https://doi.org/10.1145/3177865

[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] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg et al., Scikit-learn: Machine learning in python, Journal of machine learning research, vol. 12, no. Oct, pp. 28252830, 2011.

[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] Y. Nasser et al., NeuPow: Artificial neural networks for power and behavioral modeling of arithmetic components in 45 nm ASICs technology, in Proc. 16th ACM Int. Conf. Comput. Front. (CF), 2019, pp. 183189.

[12] 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.