Fast and Accurate PPA Modelling 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) needs to be considered while developing a System-On-Chip to improve performance, durability and reliability. This paper focuses on an accurate PPA estimation using Machine Learning models like Neural Networks, Decision Trees and Gradient Boost Regressor where Transfer Learning is used to train the model. Transfer Learning achieves a high prediction accuracy with a constrained number of samples and also reduces the time complexity by using pre-trained models. A comparative study of the different models is also presented in this paper. An accuracy of up to 98% can be achieved with the proposed models.**

***Index Terms*—Power, Performance, Area, PPA, Machine Learning, Transfer Learning, Neural Network, Decision Trees, Gradient Boost Regressor, System-On-Chip.**

I. Introduction

Power, Performance and Area of a System-On-Chip should be taken into consideration through all the design phases like architectural design, Register Transfer Level (RTL) implementation, RTL Synthesis, Place and Route. 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 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 and life-cycle of the hardware. Therefore, getting an estimate of the PPA at an early design phase of the hardware is crucial. In case the PPA target is not met, the hardware can be re-designed without wasting time and resources. Hence, Machine Learning models are used to predict the PPA at an early design phase and aid in the development of an optimized circuit.

The most extensive phase in designing PPA models is gathering data and the training time of the model. In this paper, 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. Hence, the data needed for training new PPA models as well as the time complexity of training is significantly reduced. However, to implement transfer learning the base dataset and target dataset should have atleast a few common parameters. As demonstrated in [5], Transfer Learning predicts the PPA with an accuracy of more than 98%. The model can be trained on the PPA measurements that are 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, the 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 and Section IV dives into the parameters and model structure and the test framework. Section V deals with the dataset and Section VI provides an insight into the evaluations and results. The paper concludes with the Conclusion in Section VII.

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

PPA Modelling is critical in an early design phase of all of System-On-Chips to predict the PPA of the final hardware. Different Machine Learning techniques have been used over time for PPA predictions.

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, Machine Learning models are demonstrated which use Transfer Learning to achieve high accuracy with a constrained amount of data. The Machine Learning models are built using Neural Networks and Gradient Boost Regressor as explained 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.

III. Machine Learning models for PPA Estimation

Here, a few Machine Learning models like Neural Networks, Decision Trees and Gradient Boost Regressors 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.

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 exist 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 that has an input layer, hidden layers and an output layer. Each layer contains many neurons which are mathematical functions that have 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 explained in [14] which is added to help the NN learn complex patterns in the data. The activation function adds non-linearity to the data and can also restrict 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].

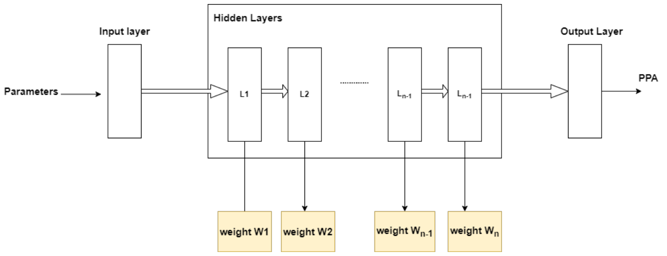

Fig. 1. Neural Network with no shared layers [5]

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 as shown in [5]. This paper demonstrates the use of Transfer Learning, where a base model with a shared set of weights, and a shared set of layers 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 number 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.

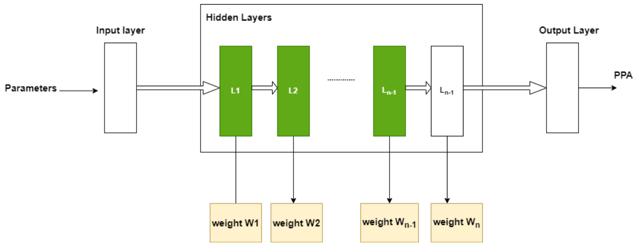

Fig. 2. Neural Network with the shared layers depicted in green [5]

B. Decision Trees

A decision Tree, as demonstrated in [2], is an algorithm that contains conditional control statements. It has a tree-like structure that branches out from a root 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.

The parent node can be defined as the node where splitting occurs and the children nodes are formed after splitting at the parent node based on conditional statements. To implement Transfer Learning, a previously built Decision Tree can be used as the base model and new branches 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. The new set of training data should have a few features which were already present in the base model for Transfer Learning to be effective.

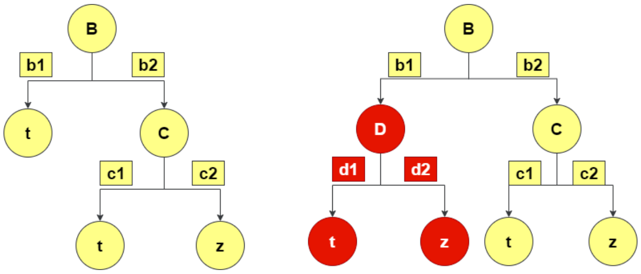

Fig. 3. Transfer Learning in Decision Trees [12]

In Fig. 3, a previously built decision tree is shown on the left. On this prior knowledge, a new Decision Tree is built depending on the different features the new task has compared to the prior task. The process of applying Transfer Learning to Decision Trees is known as the Transfer Decision Tree Algorithm(TDT) as explained in [12]. For example, applying TDT to the pre-built Decision Tree in Fig. 3, appends the new feature D, which was not present previously, to the Decision Tree and then branches from D to lead to the desired result based on decisions d1 and d2.

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 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 to minimize the loss function. Hence, the Gradient Boost(GB) Regressor iteratively combines the weak learners like Decision Trees into a strong learner and in the process implements Transfer Learning.

IV. PPA Model parameters and model structure

A. *Parameters used for PPA model Generation*

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, the PPA of the hardware can be predicted .

B. *PPA Model Generation Framework*

A Framework to generate the PPA model is shown in Fig. 4. In the first stage, design data for RTL Synthesis 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]. The RTL is synthesized by taking up random combinations of data from the sample space. The PPA is calculated for all the combinations. The parameters and the estimated PPA can be used as the training data to train the model and the final model is obtained.

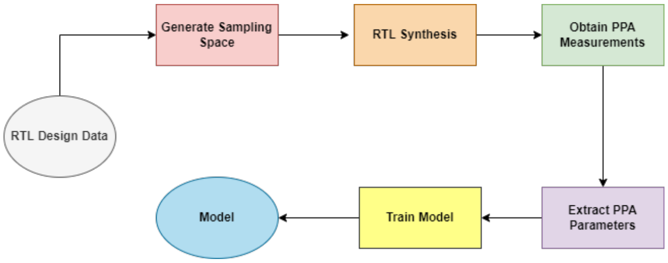

Fig. 4. PPA Model Generation Framework

C. *PPA Model Testing Framework*

A Framework to test the generated PPA model is being shown in Fig. 5. This framework determines whether the RTL design parameters of the test hardware are accurate enough for the PPA to be optimal. Firstly, the RTL design data is taken and then the model generated by the PPA Model Generation Framework as shown in Fig. 4 is tested with the RTL data. If the PPA target is met, the RTL is synthesized and the next phase 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.

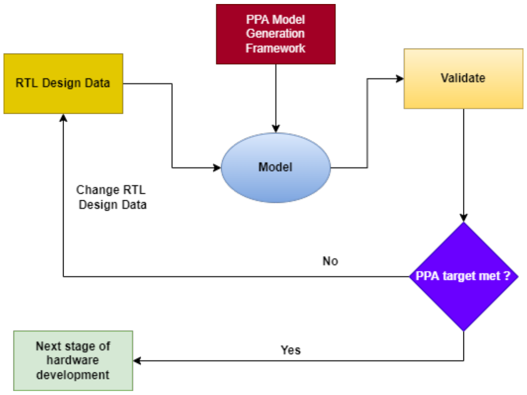

Fig. 5. PPA Model Testing Framework

V. Dataset for building and evaluating 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 RTLs. These PPA measurements serve as the training data 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. Evaluations 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 Square Error (MSE) which can be defined as the average squared difference between the estimated values and the actual values. The MSE can be calculated as:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y - y_1)^2$$

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

TABLE I
MODELS EVALUATION [5]

Accuracy Parameters	**Machine Learning Model**		
	Neural Network	***Decision Tree***	***GB***
Dynamic Power	97.59	97.18	97.58
Critical Path	99.11	98.80	99.17
Area	97.71	97.83	98.47
PPA	97.74	97.40	98.03

Table I [5] shows the accuracy parameters that is obtained after implementing different Machine Learning Models. The Accuracy Parameters, as seen in [5], are measured as:

$$AccuracyParameters = 100\% - MSE$$

The chosen parameters for the estimation are dynamic power which gives an estimate of the power consumed, critical path which estimates the time a signal takes to travel through a circuit hence giving a measure of the performance, Area which takes into account the area consumed by the circuit as a whole and PPA which measures the overall PPA of the circuit. It can be concluded that the Gradient Boost Regressor has the highest PPA accuracy while the Decision Trees have the least. However, all the three models have an overall PPA prediction of accuracy of more than 97%.

The Neural Network has the highest prediction accuracy for Dynamic Power and the least for Area and the overall PPA. While the Decision Tree has a consistent prediction accuracy for all the accuracy parameters, the Gradient Boost Regressor has a prediction accuracy with Dynamic Power at 97.58% and Area at 98.47%.

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 Machine Learning models is shown in Fig. 6. 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.

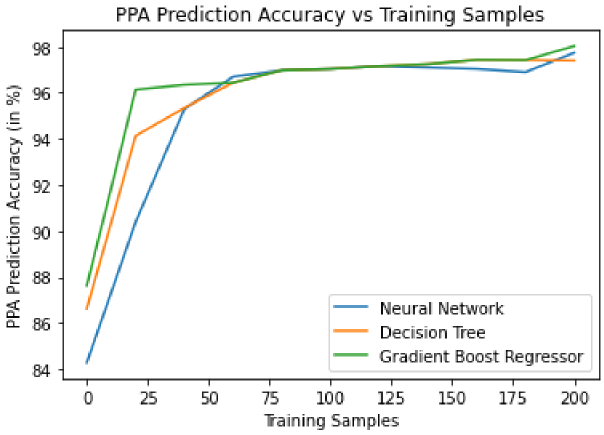

Fig. 6. PPA Prediction Accuracy vs Training Samples [5]

As it can be seen from the Fig. 6 that the Gradient Boost Algorithm has a higher Prediction Accuracy than the other two models if the number of samples is close to 200. However, if the number of training samples is between 75 and 125, all the three models predict with almost the same accuracy. 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 in which Transfer Learning is used to train the model to predict the Power, Performance and Area of System-On-Chips hardware design. The models are successful in obtaining a high PPA prediction accuracy. Gradient Boost Regressor turns out to be the best model with a PPA prediction accuracy of 98.03%. However, Neural Networks and Decision Trees both perform well at an accuracy of 97.74% and 97.40% respectively.

Machine Learning models to predict the PPA of the hardware at an early design phase which improves the durability, reliability, performance and also determines the future design stages so that it leads to the design of a hardware with an optimized PPA.

Transfer Learning reduces the amount of data that is needed to train our models and also reduces the time complexity by using pre-trained models. Also, using Transfer Learning a higher prediction accuracy can be achieved using a relatively fewer number of new training samples as shown in Fig 6.

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.