Machine Learning Applications of Signal Integrity and Power Integrity

Dr. Zhiping Yang and Dr. Tianjian Lu

14th Central PA Symposium on Signal Integrity

April 16, 2021

Agenda

- Motivation
- Machine learning and hardware design
- Case studies
  - Placement optimization of decoupling capacitors
  - PDN impedance prediction
  - Eye-diagram-metrics prediction with DNN
  - Eye-diagram generation with RNN
- Conclusion

Acknowledgement

Professor Antonio Orlandi and Francesco de Paulis, DIIIE, University of L'Aquila, L'Aquila, Italy

Professor Chulson Hwang, ECE, Missouri S&T

Professor Ju Sun, CSE, University of Minnesota at Twin Cities

Thong Nhu Nguyen and Professor Jose Schutt-Aine, ECE, UIUC

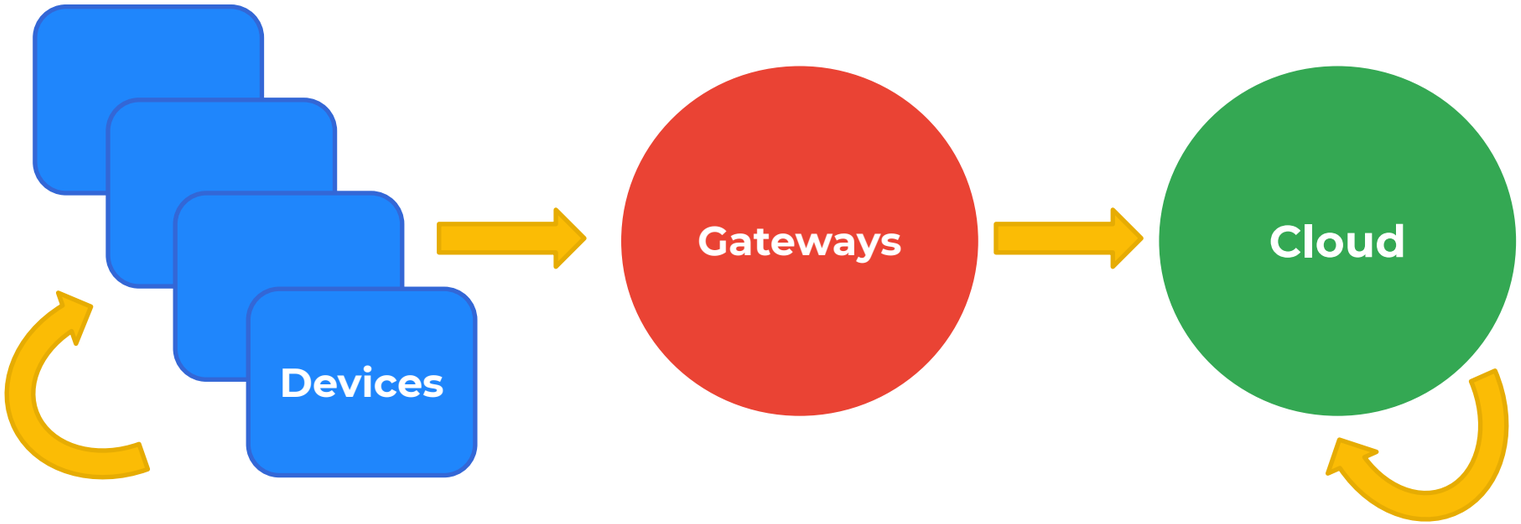
Dr. Ken Wu, Google LLC

Motivation: Global IoT and Consumer Electronics

- Around **29 billion** connected devices are forecast by 2022, of which around 18 billion will be related to IoT [1].
- **70%** of wide-area IoT devices will use cellular technology in 2022 [1].
- The IoT in consumer electronics industry globally is expected to expand up to **USD $124 billion** while growing at **CAGR 24.16 percent** through the forecast period from 2017 to 2023 [2].

- [1] The connected future, https://www.ericsson.com/en/mobility-report/internet-of-things-forecast
- [2] Market research future, link https://www.marketresearchfuture.com/reports/iot-consumer-electronics-market-997
- Figure from https://towardsdatascience.com/iot-machine-learning-is-going-to-change-the-world-7c4e0cd7ac32

Motivation: Global IoT and Consumer Electronics



- The IoT has three components at the top level, namely, devices, gateways, and cloud.
- The devices such as smartphones, wearable devices, smart TVs, kitchen appliances directly interact with the physical world, convert valuable information from the users into digital data, and generate **massive amount of data**.
- The massive amount of data are filtered and analyzed, providing visibility on users, products, services, and applications, etc.

Motivation: Incorporate the Power of ML into IoT

- Machine learning has become a robust analytical tool for vast amount of data generated by IoT devices.
- The incorporation of ML into IoT enables an efficient way of wring visionary insights from data, leading to the new areas including **self-driving cars, speech/facial image recognition, natural language processing, active web search**, etc.
- Google Cloud IoT Edge which extends Google's powerful data processing and machine learning to billions of edge devices, such as robotic arms, wind turbines, and oil rigs, so they can act on the data from their sensors in real time and predict outcomes locally.

Machine Learning and Hardware Design

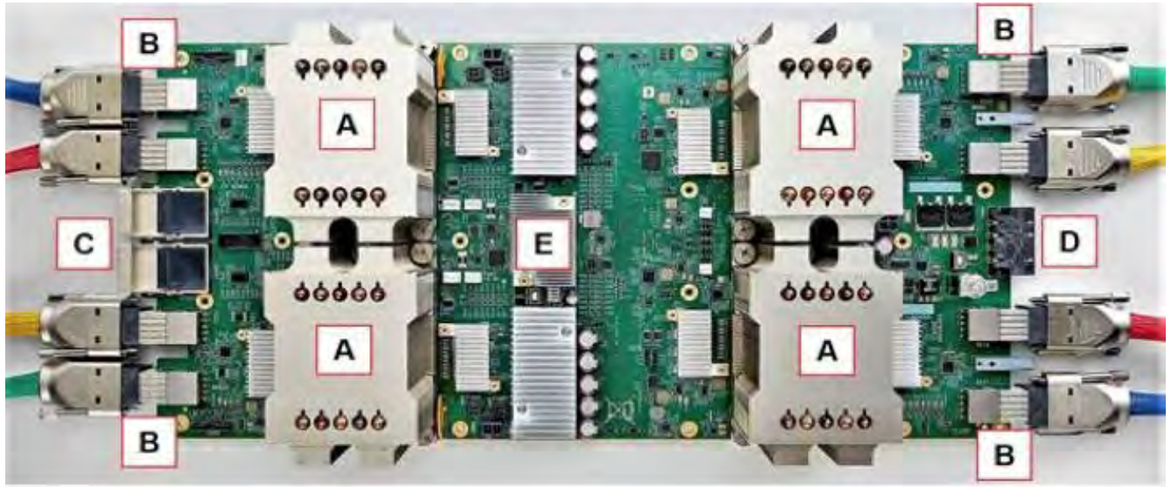
Data centers

- To store the massive amounts of data, massive-scale data centers are built with petabytes or even exabytes of storage, hosting ~100,000 servers.

Machine Learning and Hardware Design

To process the massive amounts of data,

- CPU and GPU clusters,
- Transistor scaling (Moore's Law), power limitation (constant power density, Dennard scaling), and parallelism limitation (number of processors per chip, Amdahl's law)
- Domain-specific hardware including Google's Tensor Processing Unit (TPU) deployed in 2015.



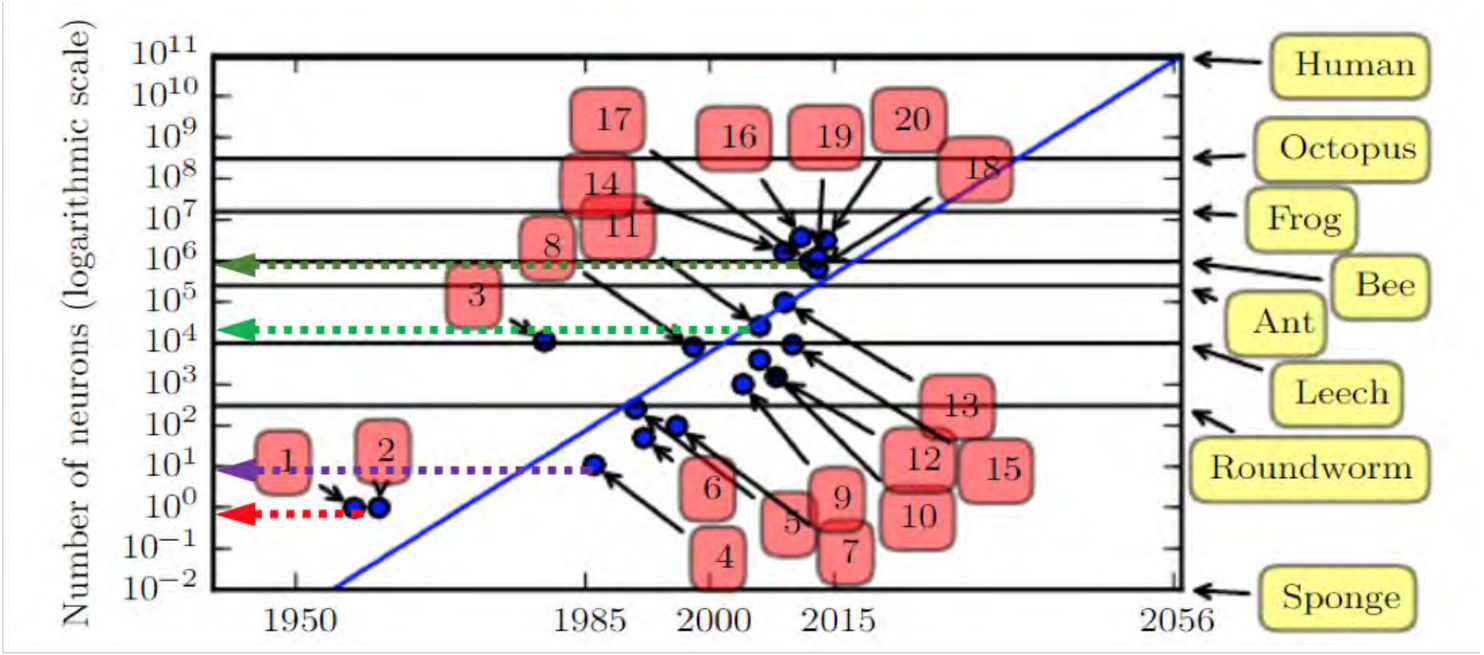
Machine Learning and Hardware Design

Multi-GPU convolutional neural network (2012)

GPU accelerated convolutional neural network (2006)

Early back-propagation network (1986)

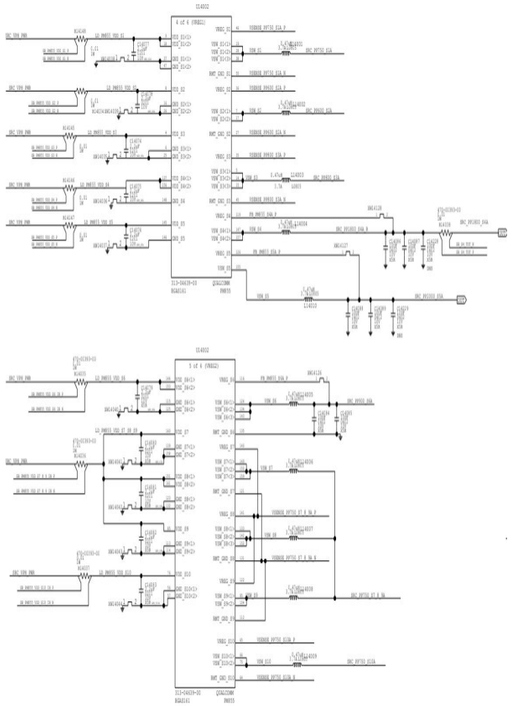
Perception (1958)

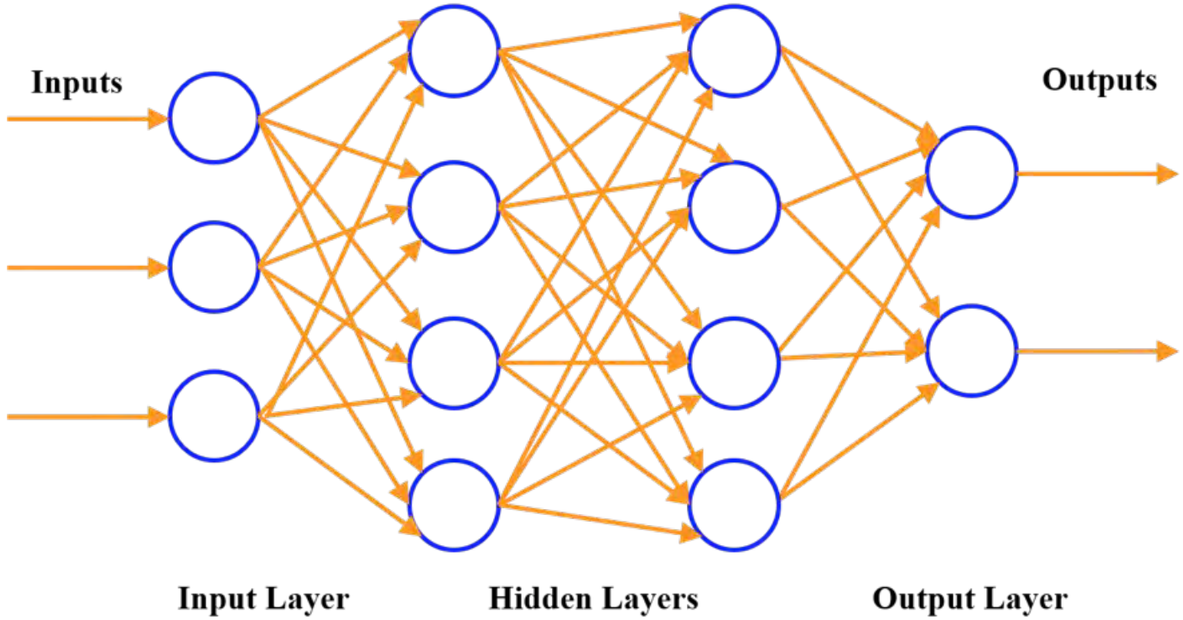


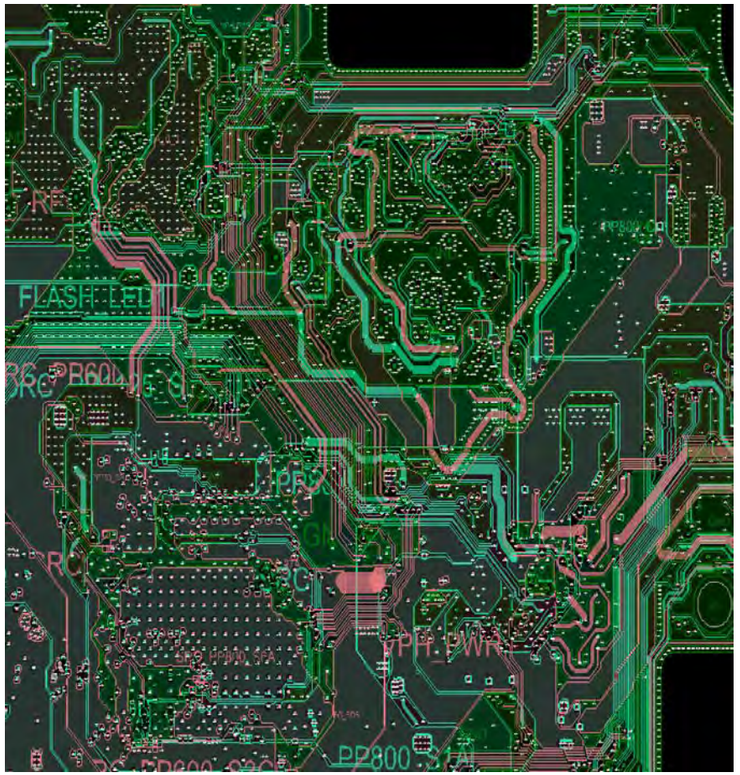
Ivan Goodfellow et al.

Neural Network Size Grows with Hardware Advancement

Can Neural Networks be Applied in Hardware Design?







Schematic

Layout

Mark Hayter, Plenary Talk, 2018 IEEE EMC Symposium, Singapore

Case Study: Optimization for Decoupling Capacitor Placement

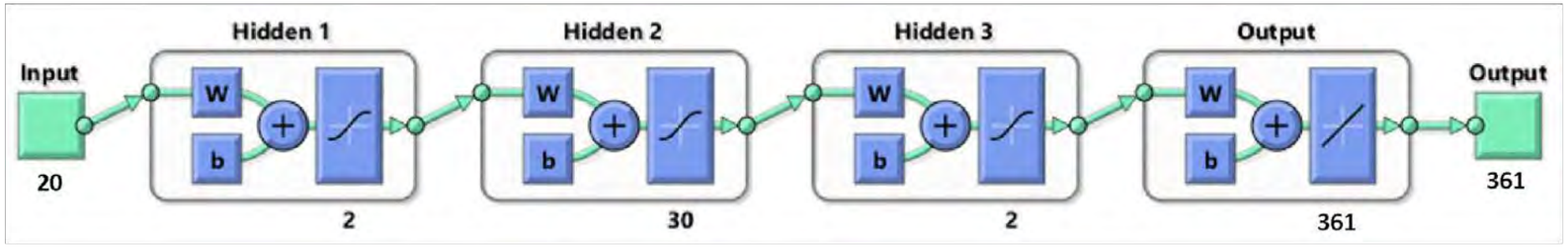
- An iterative optimization for decoupling capacitor placement on a power delivery network (PDN) is developed based on Genetic Algorithm (GA) and Artificial Neural Network (ANN).
- The ANN is first trained by an appropriate set of results obtained by a commercial simulator.
- Once the ANN is ready, it is used within an iterative GA process to place a minimum number of decoupling capacitors for minimizing the differences between the input impedance at one or more location, and the required target impedance.

Case Study: Optimization for Decoupling Capacitor Placement

Collaboration with UAq EMC Laboratory through Google Faculty Research Award (2018):

- Proposed architecture of the ANN for placement optimization
- Basic structure of the GA
- Structure and the electrical properties of the PDN test vehicle
- Results of the optimization processes

Case Study: Optimization for Decoupling Capacitor Placement



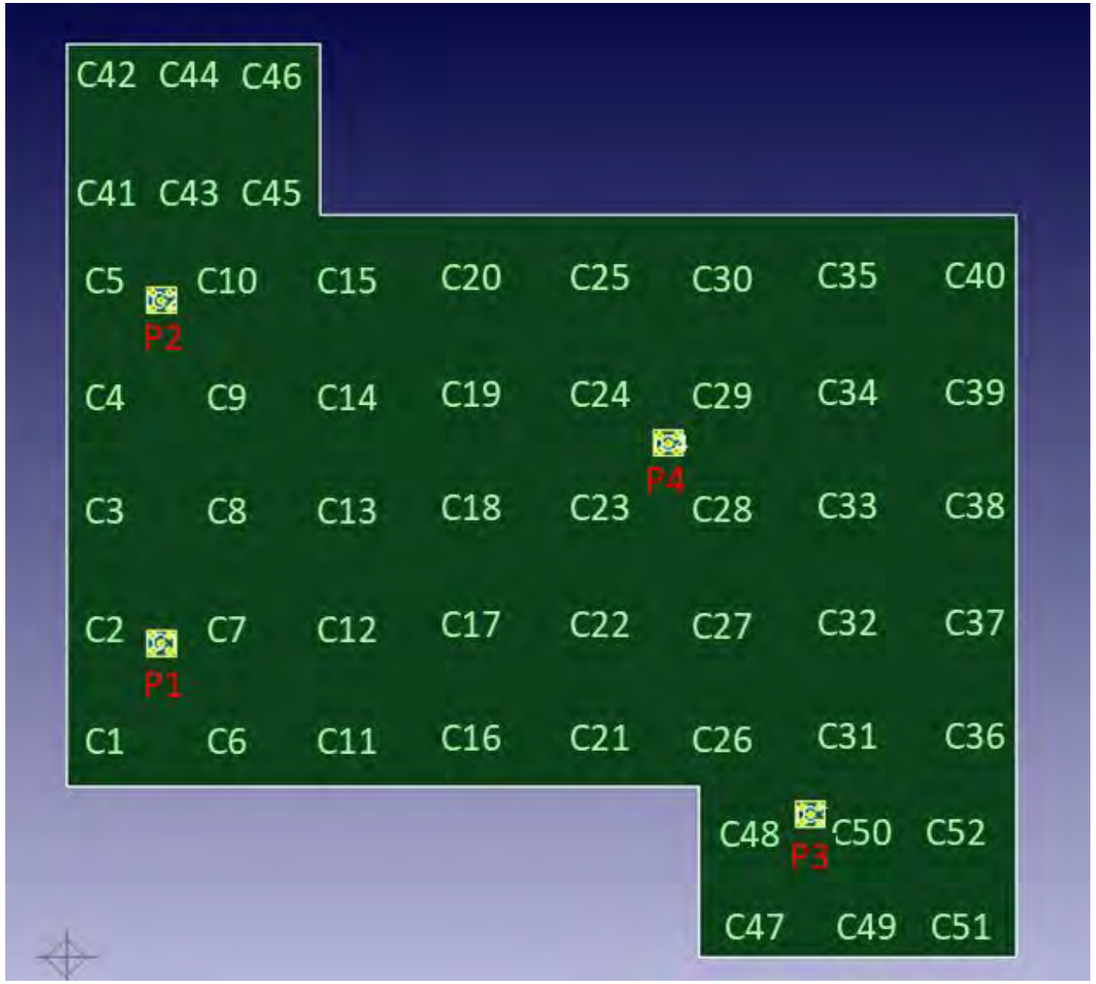
Input: decaps value and location

Output: input impedance

- The final architecture consists on 3 hidden layer (width of 2, 30 and 2, respectively)
- The output layer has size of 361 : the values of **Zin(f)** at the 361 frequency points of the spectrum

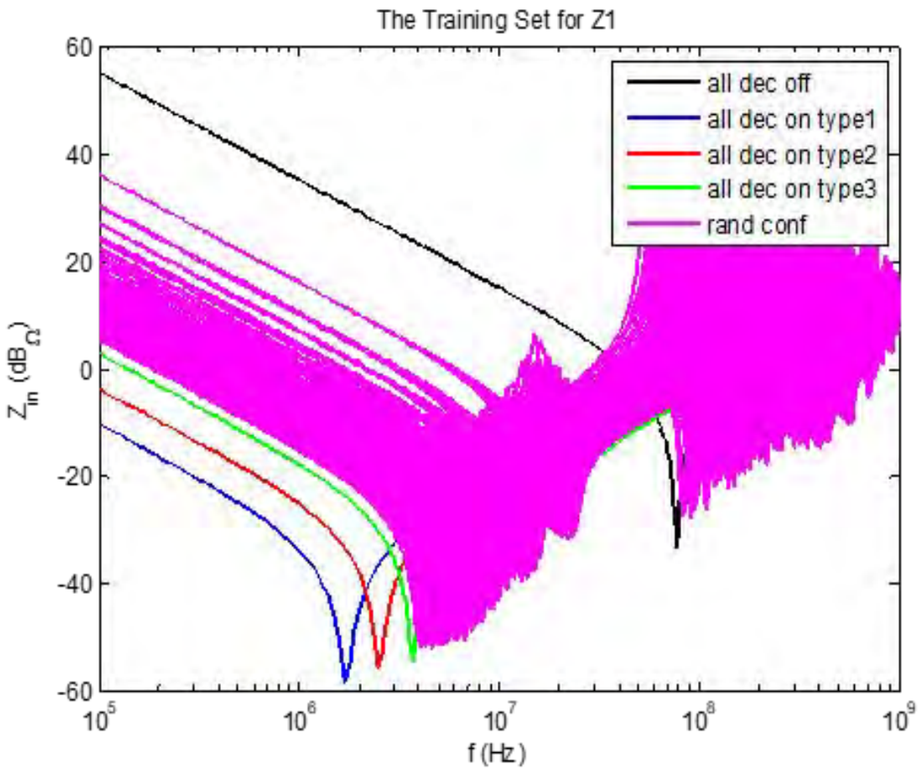
Training Set

- In order to speed up the generation of the TS, a configuration matrix is built. It will be used as input to compute the frequency spectrum of the **Zin(f)** at the four ports **Pi**
- The TS should be not only **representative** of the most significant configurations of the actual working space, but should also be matched with the **iterative nature** of the GA.
- In the TS there are configurations with 1 to 10 decaps at the time.



Training Set

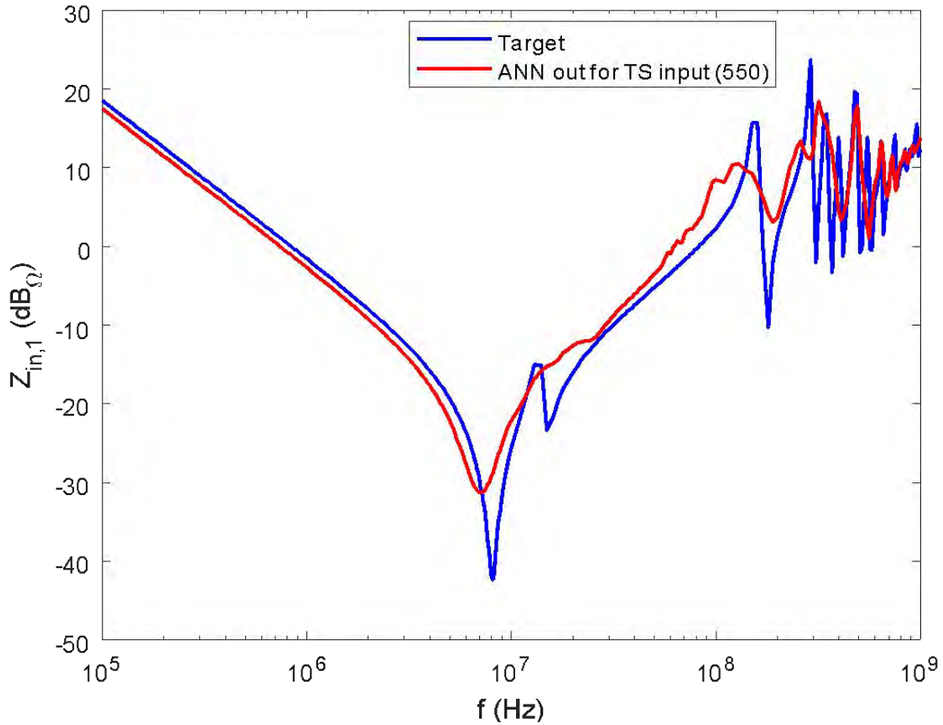
- Generalization:
  - A limit of the ANN is its reduced capability to generalize the outputs for inputs that are outside the boundaries of the input configurations used in the TS.
- To overcome this limitation, the TS also includes some “extreme” input configurations as:
  - Bare board (no decoupling capacitors at all)
  - All decaps of each of the type considered



TS for one of the four ports (P1)

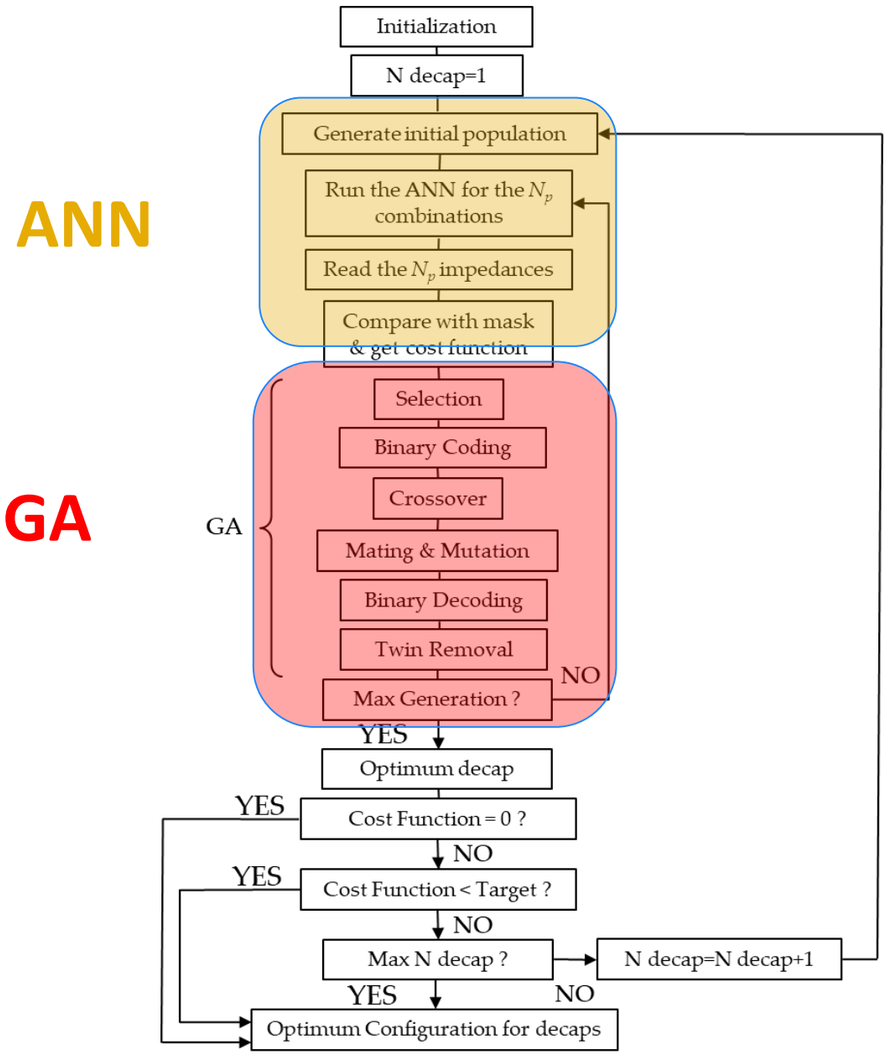
Validation

- Reserve part of the data set for validation purpose.
- Compare the NN prediction to the results obtained using high-fidelity simulation.



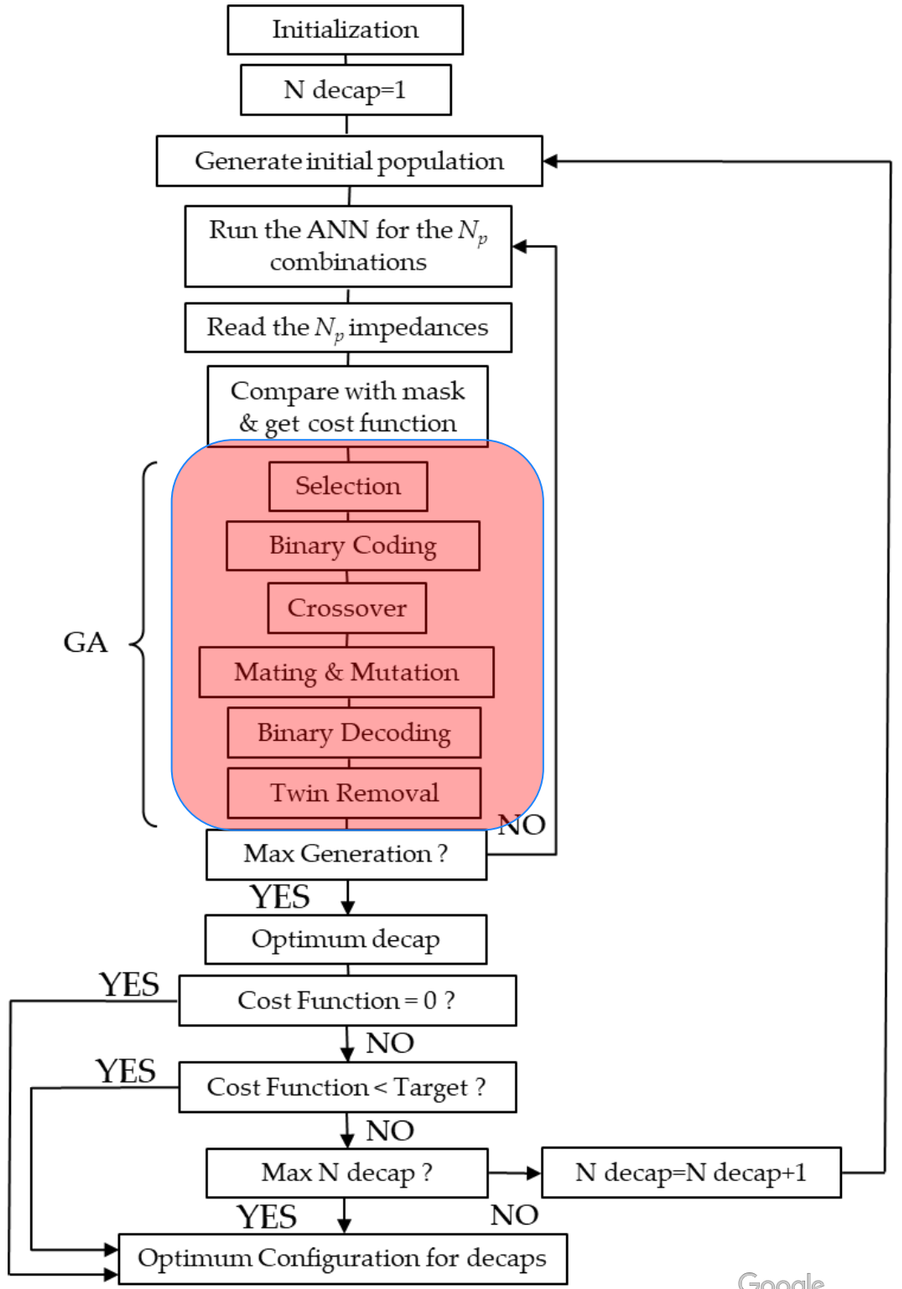
Example of validation

Global ANN-GA Architecture



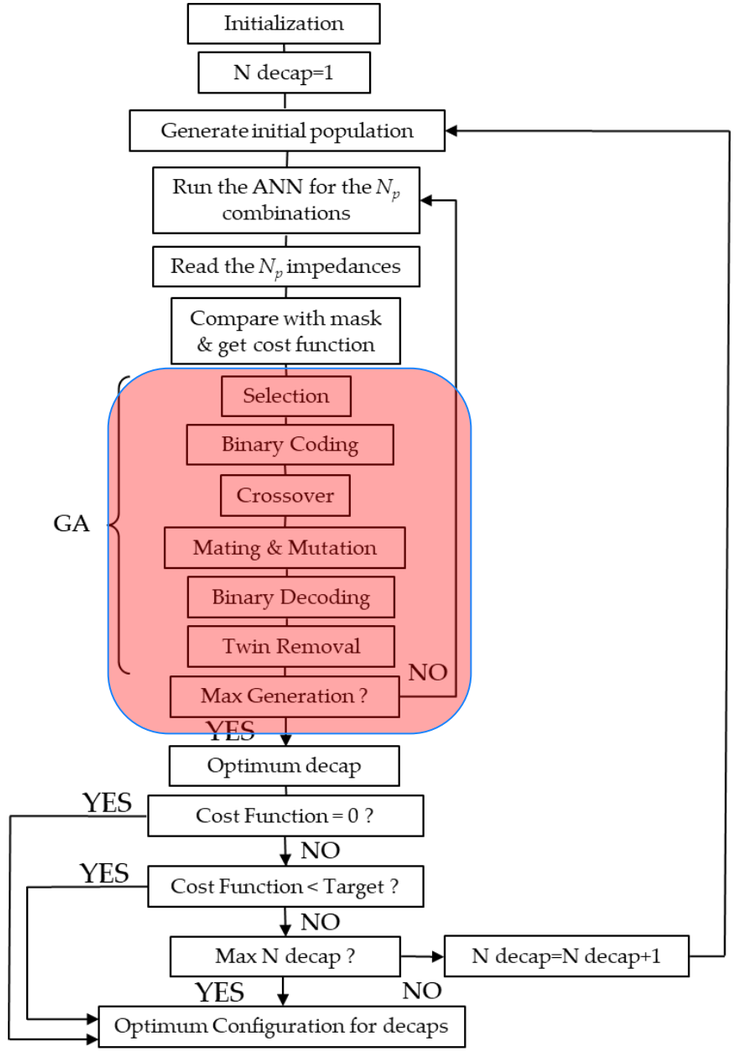
Basic Structure of the GA

- The GA iteratively places one decap at each optimization run, such that when the exit condition from the algorithm is met, a minimum number of decap is obtained.
- The GA is based on a binary encoding of the chromosome composed by two variables: (1) decap position and (2) decap value



Basic Structure of the GA

- Main features of the implemented GA:
  - The binary coding is able to better randomize the generation of new chromosomes
  - The twin removal avoids any redundancy in the creation of new chromosomes within a given population
- The global cost function is the sum of the cost functions associated to the Nports (= 4) for which the PDN impedance should be optimized.

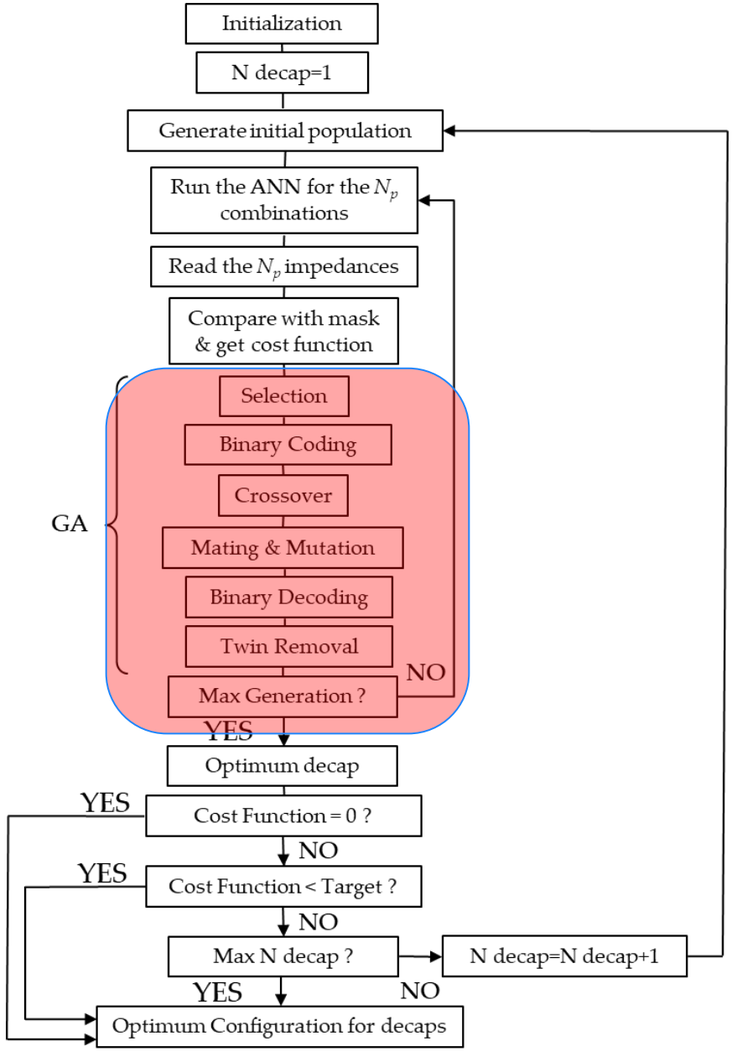


Cost Function of GA

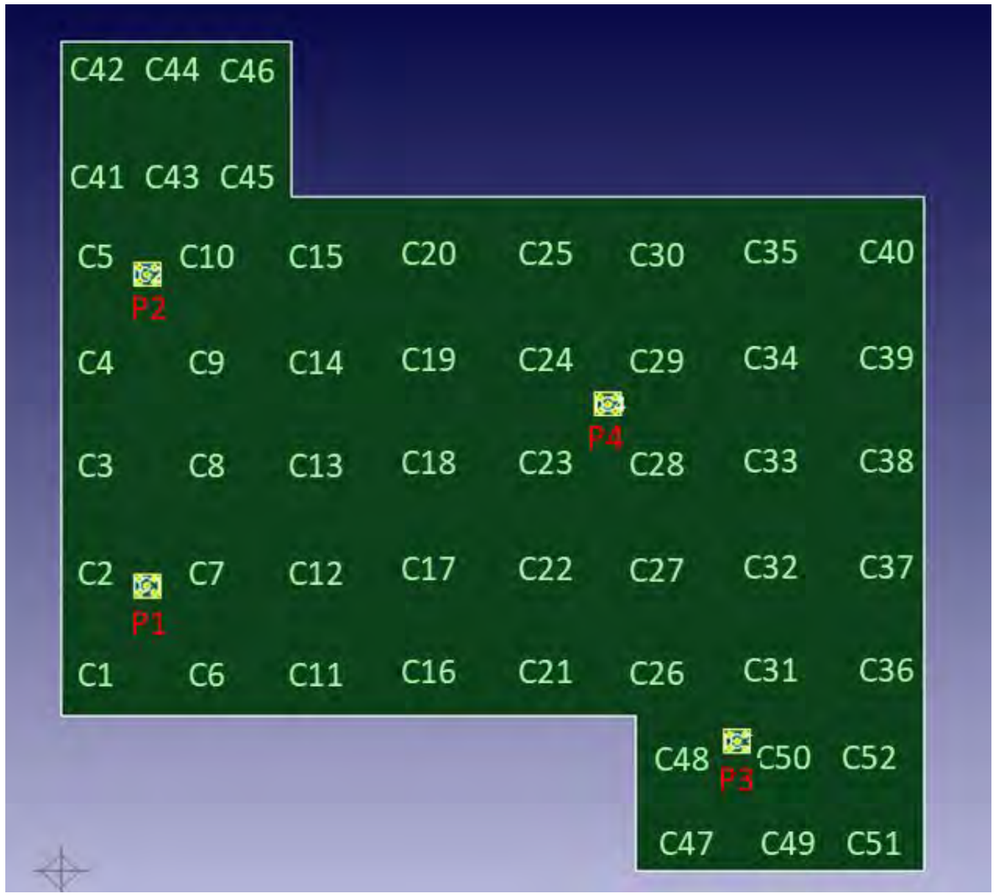
$$\Delta Z_{in\_i}(f_j) = \begin{cases} \left| Z_{in\_i}(f_j) - Z_{mask}(f_j) \right| & \text{when} \quad \left| Z_{in\_i}(f_j) \right| \geq \left| Z_{mask}(f_j) \right| \\ 0 \quad \text{when} \quad \left| Z_{in\_i}(f_j) \right| < \left| Z_{mask}(f_j) \right| \end{cases}$$

$$f_{cost\_i} = \frac{\sum_{j=1}^{N_{freq}} \Delta Z_{in\_i}(f_j)}{N_1}$$

$$f_{cost} = \sum_{i=1}^{N_{ports}} f_{cost\_i}$$



Results of the Optimization

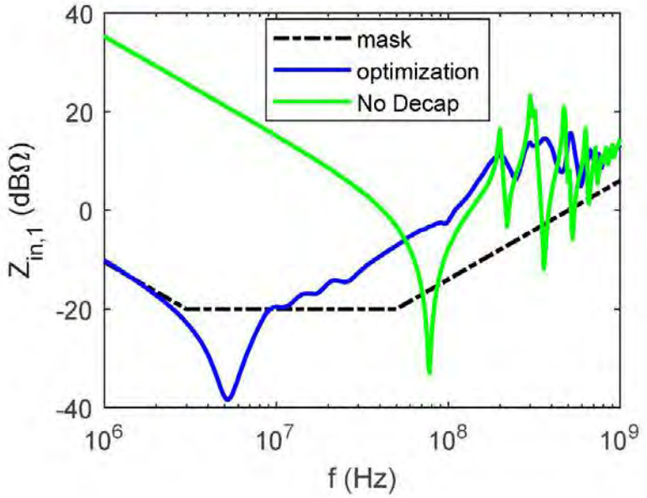


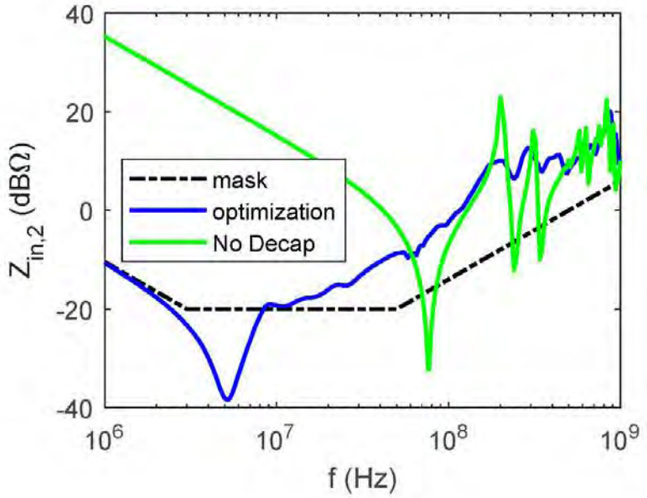
Tests run on the virtual model of the PDN

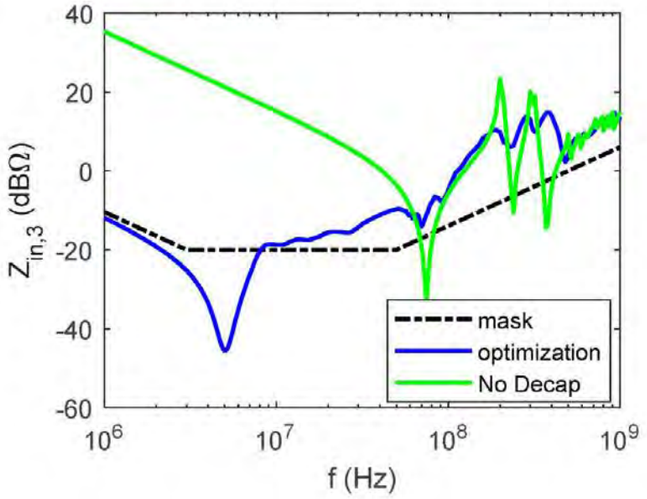
Elementary capacitors			
	C (nF)	ESL (pH)	ESR (mΩ)
C_1	100	222	8.9
C_2	47	154	21.4
C_3	22	142	25.2

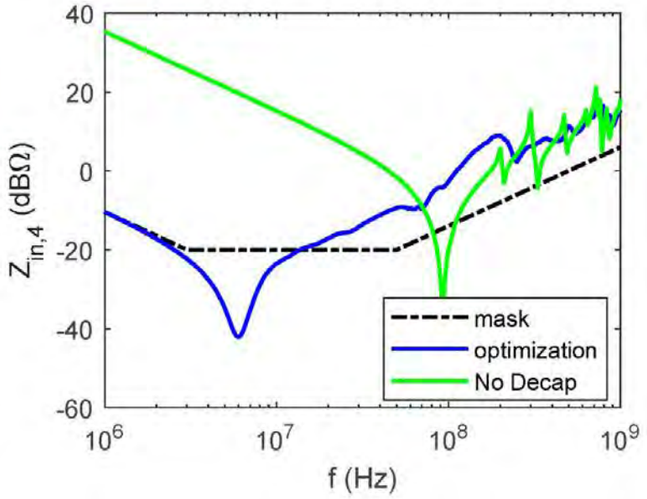
Results of the Optimization

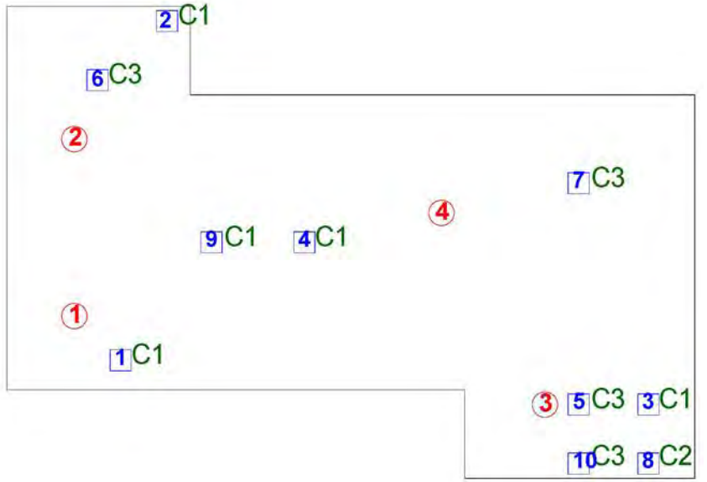
Pop: 10 chrom
gen: 10
f_{max}: 1 GHz
iter: 10

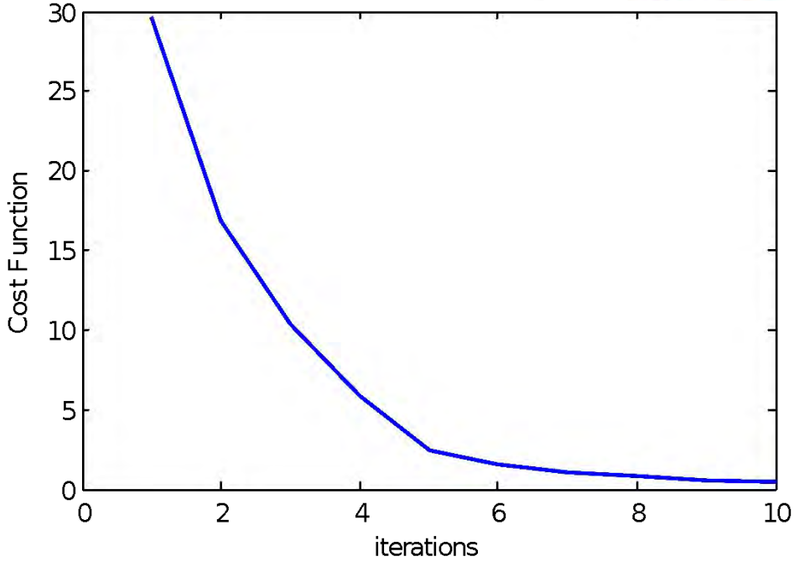










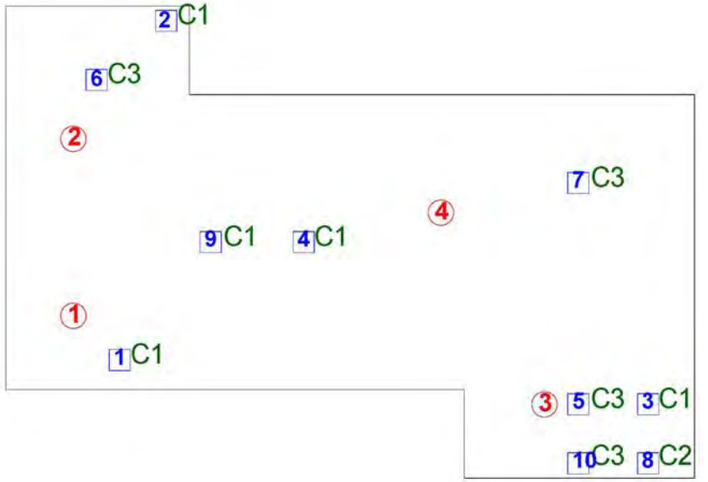


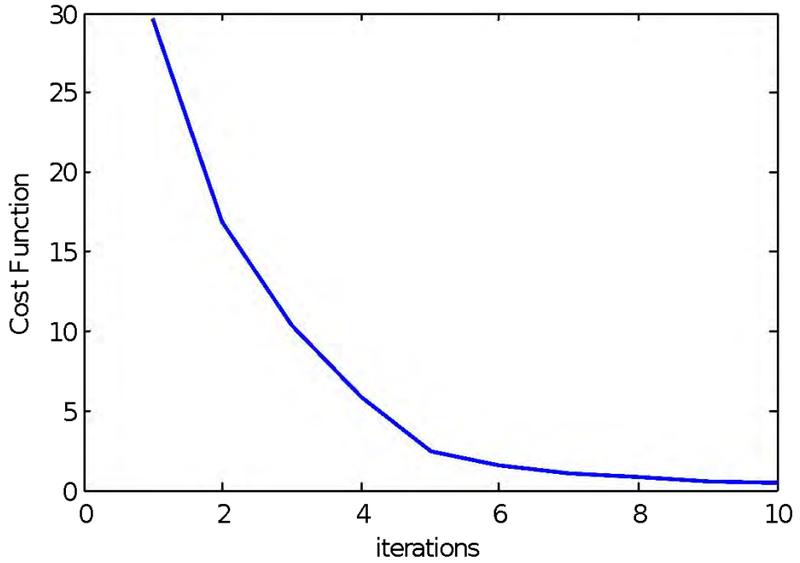
Elementary capacitors			
	C (nF)	ESL (pH)	ESR (mΩ)
C_1	100	222	8.9
C_2	47	154	21.4
C_3	22	142	25.2

Results of the Optimization

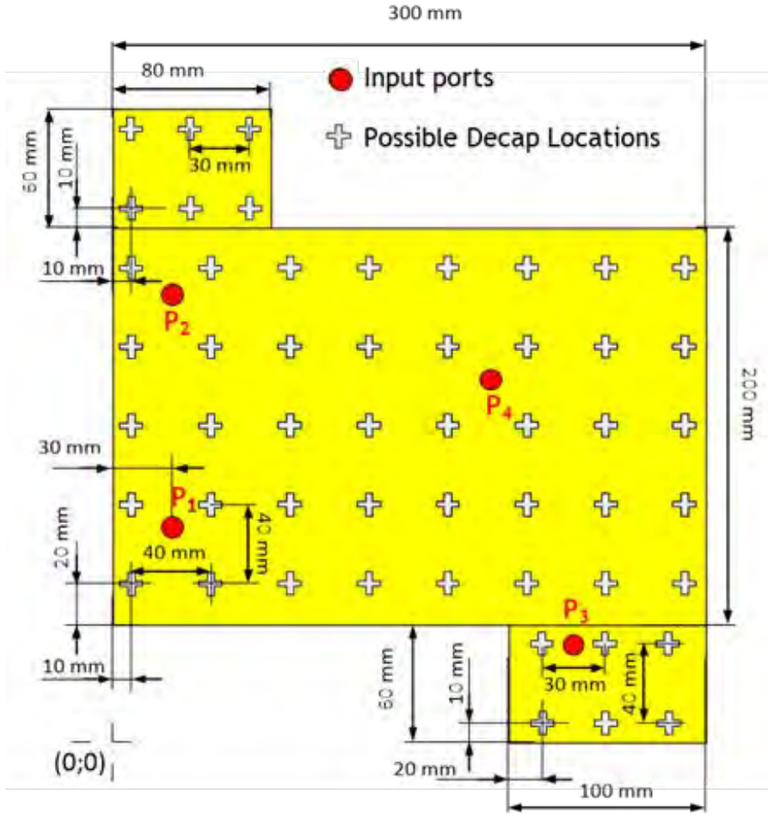
Pop: 10 chrom
gen: 10
f_{max}: 1 GHz
iter: 10

- The decaps are not clustered around any specific port
- The cost function drops quickly, evenly and fast up to a very small value

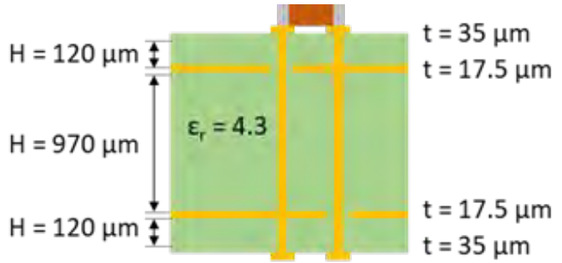




The Structure and the Electrical Properties of the PDN Test Vehicle

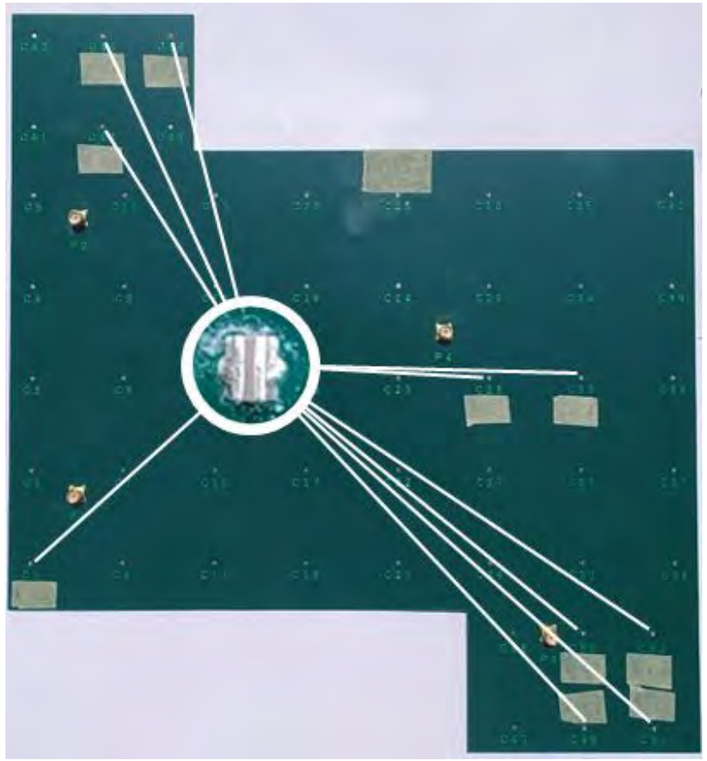


Top view,
decaps positions grid
and ports

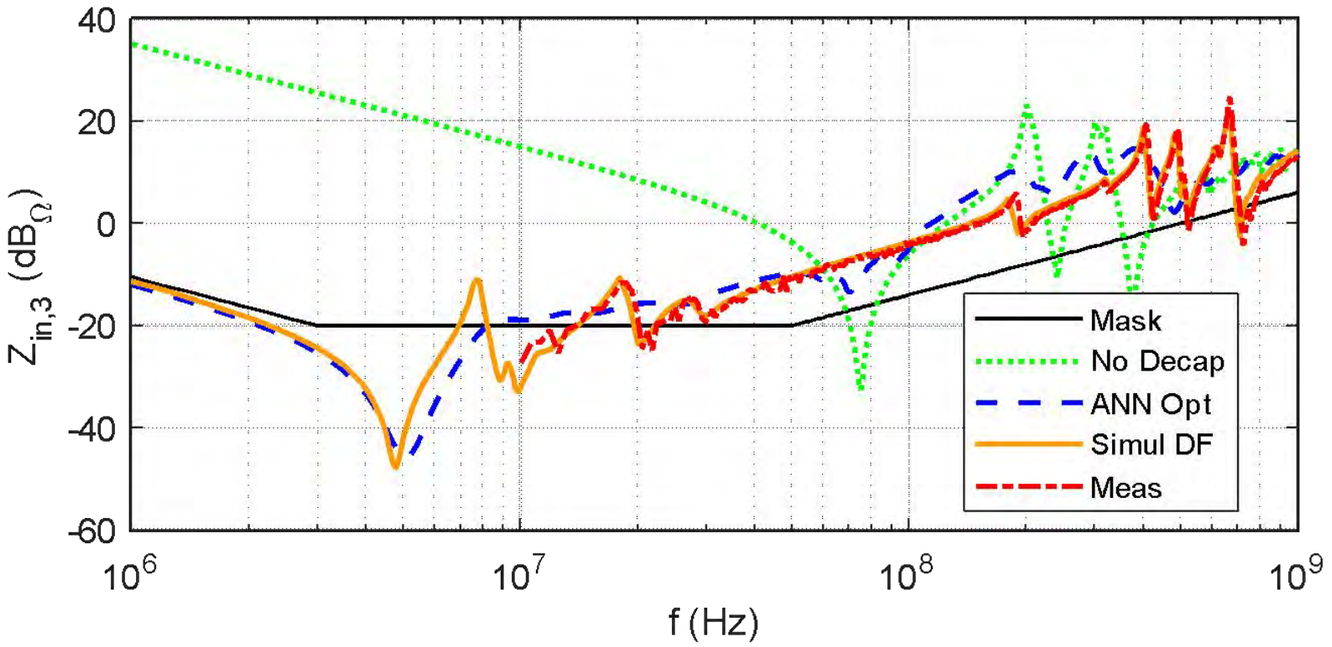


Stack-up

The Structure and the Electrical Properties of the PDN Test Vehicle



Decap locations in the real board



Z_{in} at Port 3:

- from **ANN-GA**
- from **measurement**
- from **simulation**

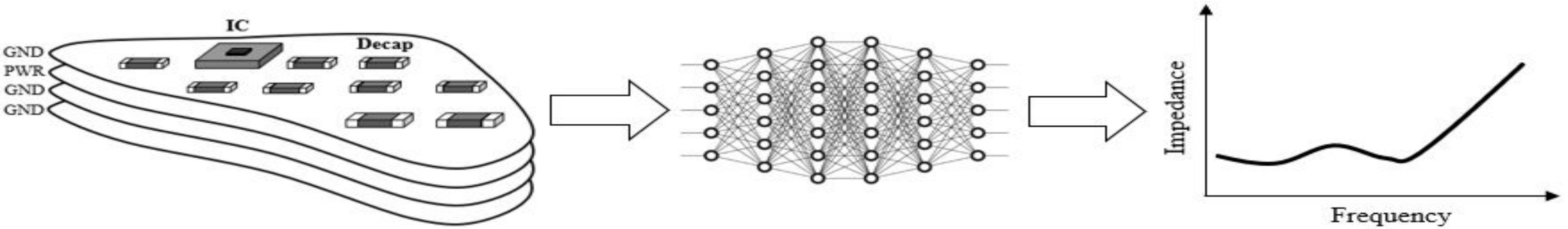
Case Study: PDN Impedance Prediction

Collaboration with EMC Laboratory Missouri S&T through Google Faculty Research Award (2020):

- Deep learning to predict the PDN impedance

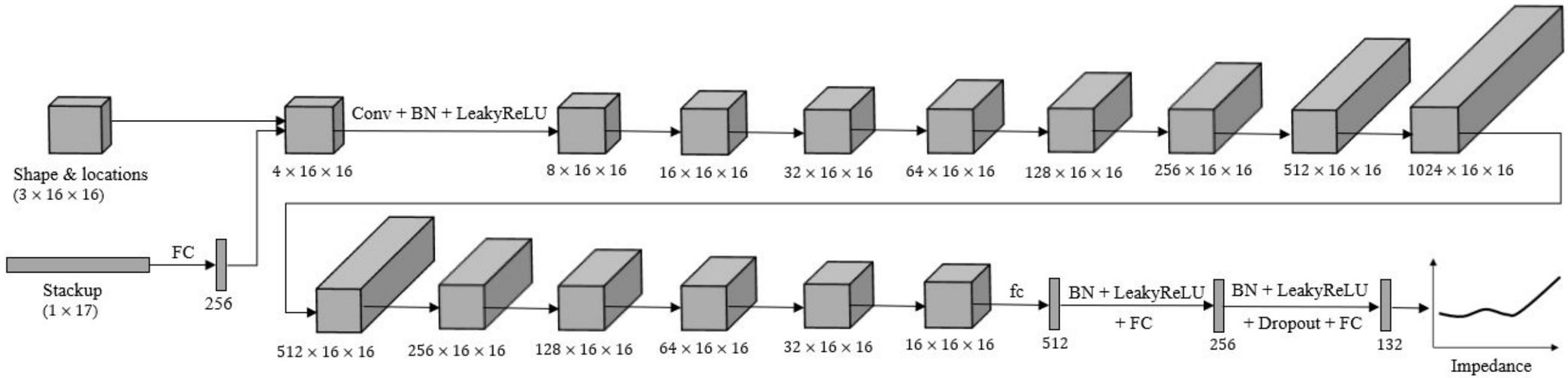
Case Study: PDN Impedance Prediction

Deep learning to predict the PDN impedance given any: board shape, stack-up, IC location, and decap placement



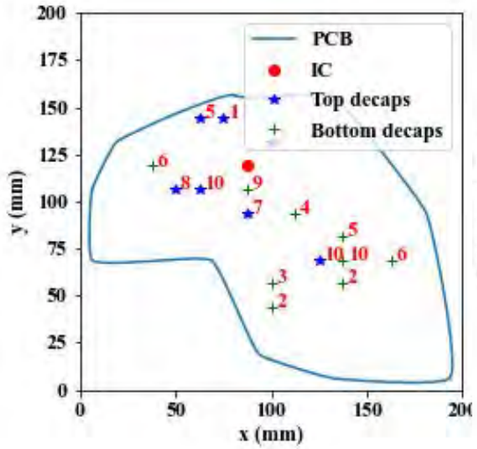
Case Study: PDN Impedance Prediction

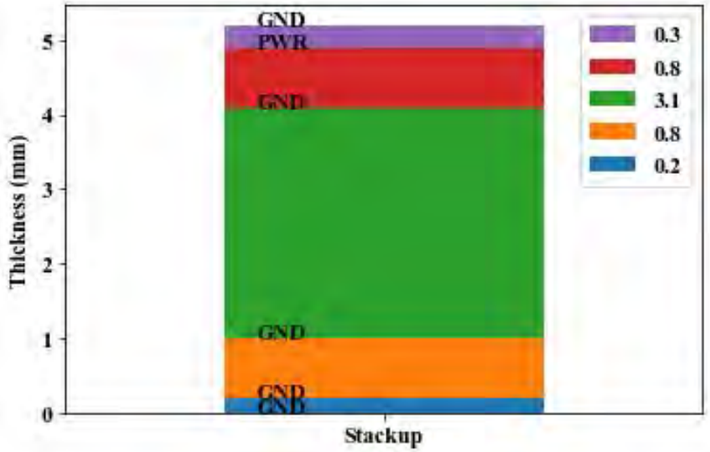
Convolutional neural network (CNN) structure:

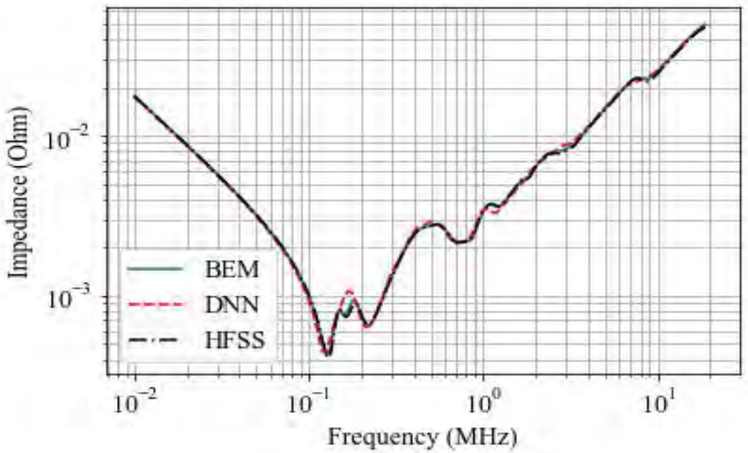


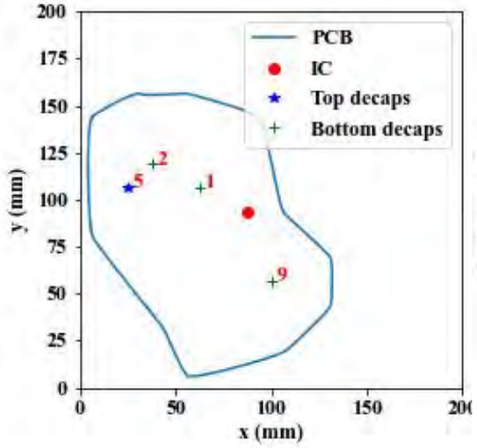
- Training: 1.3M board
- Testing: 10K board
- Training time: 80 hours (1 NVIDIA Tesla K80 GPU)

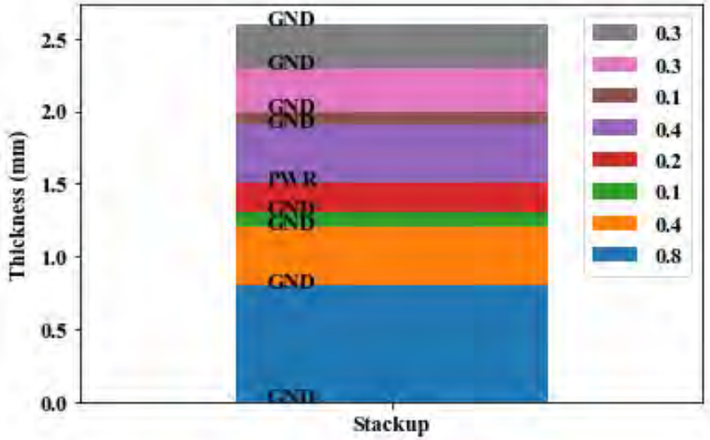
Case Study: PDN Impedance Prediction

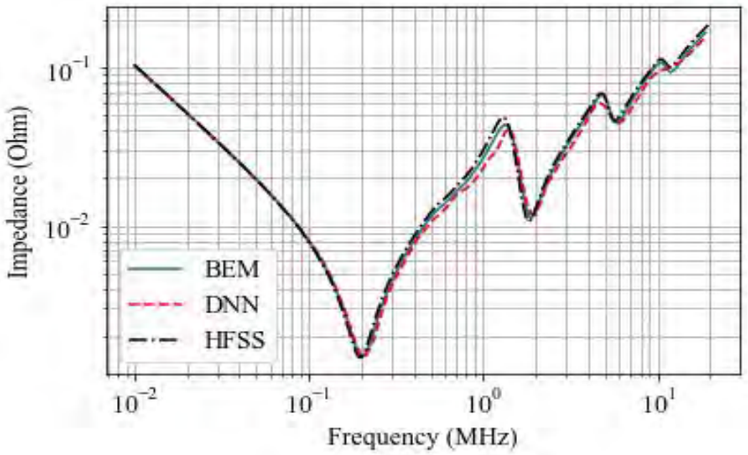












Methods	Case #1	Case #2
Full-wave	35 min	40 min
BEM	10 s	30 s
DNN	0.1 s	0.1 s

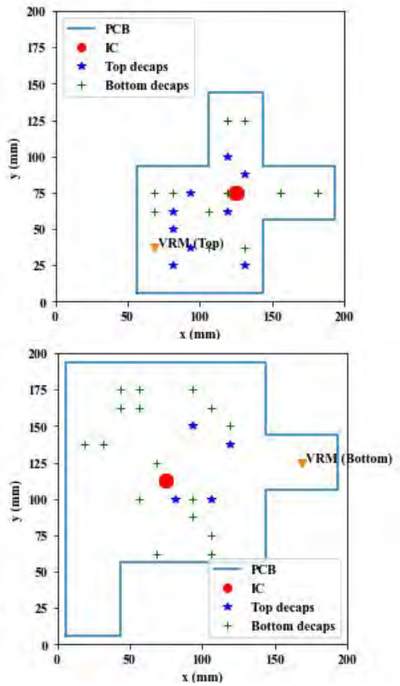
System

Processor:	Intel(R) Xeon(R) Gold 6140 CPU @ 2.30GHz 2.30 GHz
Installed memory (RAM):	192 GB (191 GB usable)

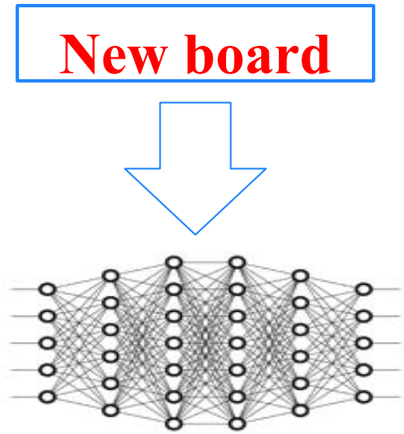
Case Study: Application of PDN Impedance Prediction

Deep learning to optimize Decap placement given any: board shape, stack-up, IC location, and # of decaps

Many boards



Train



New board

Decap solution

Finetune

GA

Two step approach: trained network + fine tune (GA)

Use the predicted solution by the DRL as a seeded solution of the GA

Case Study: ML in High-Speed Channel Modeling

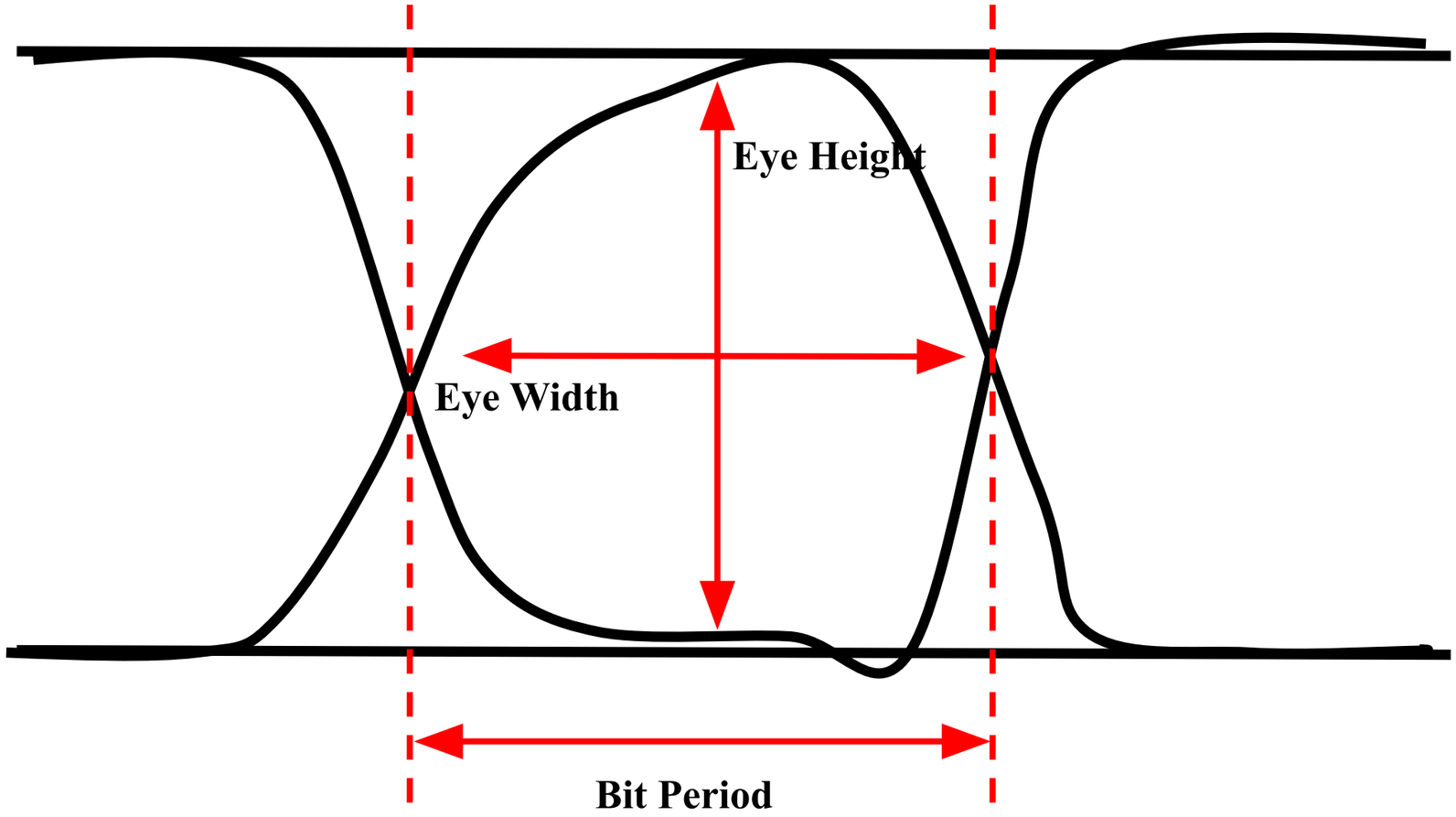
Eye-Diagram Metrics Prediction

- Linear Regression
- Support Vector Regression
- Neural Network

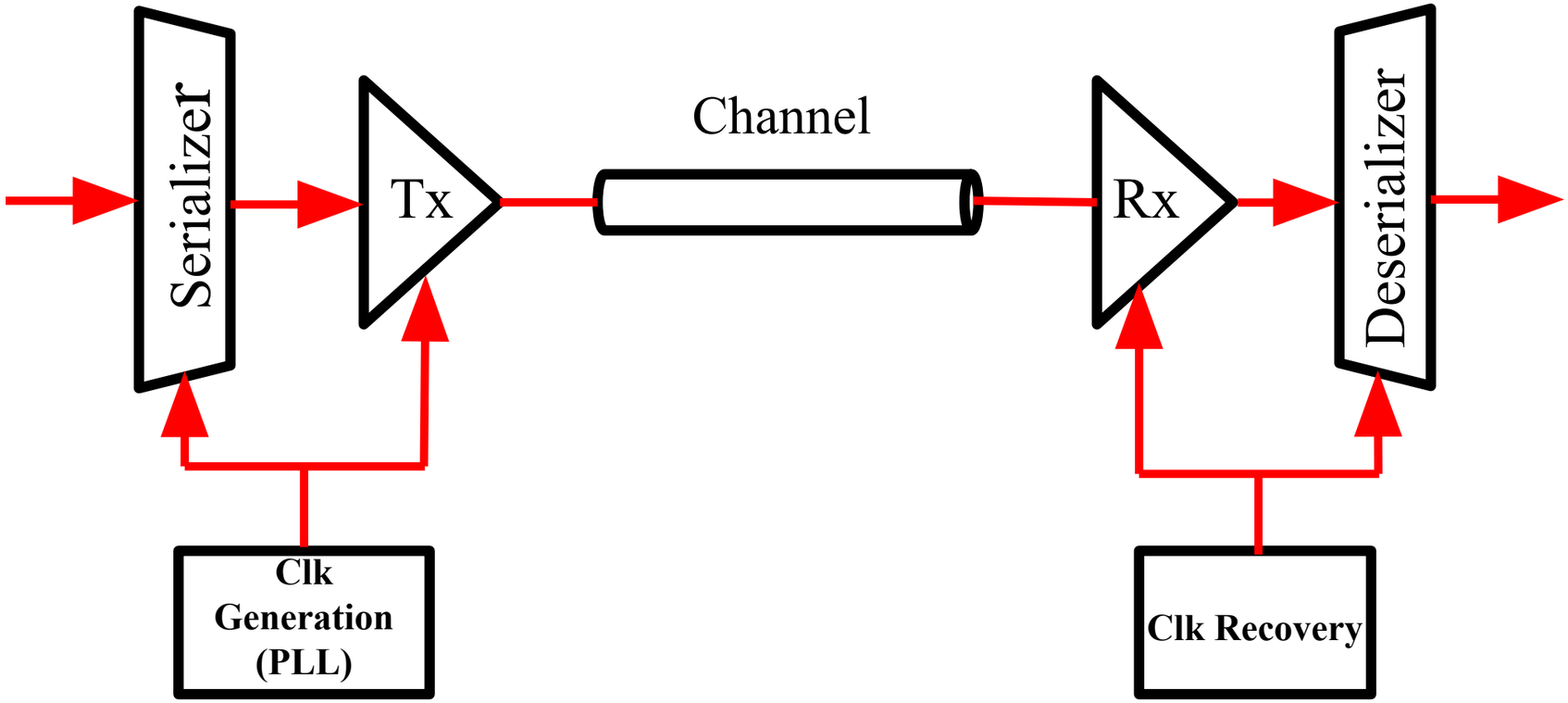
Eye-Diagram Generation (transient waveform)

- Recurrent Neural Network
- With LSTM unit and GRU

Case Study: ML in High-Speed Channel Modeling



Case Study: ML in High-Speed Channel Modeling



ML in Eye-Diagram Metrics Prediction

Simulation techniques in characterizing high-speed channels

- Electromagnetics solvers
- Eye-diagram generations

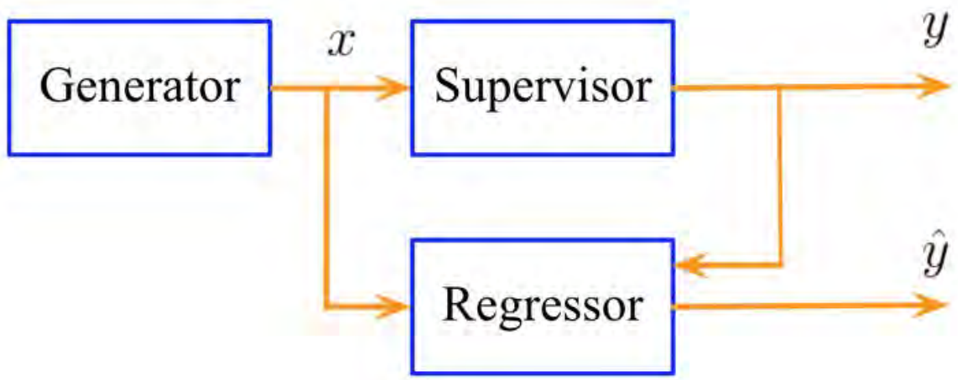
Computationally intensive

- Domain decomposition, parallel computing, fast frequency sweep, statistical eye

Utilize the large amount of data made available in a previous design or at an earlier design stage to improve efficiency

	Transmitter	Interconnect	Receiver
Input Parameters	• Pre-emphasis	• Trace With • Spacing • Substrate Thickness • Loss Tangent • Dielectric Constant	• Equalization

ML-Based Channel Model



- Generator creates a set of input parameters for a high-speed channel, stored in $\{x\}$.
- Supervisor returns eye height or width y based on $\{x\}$.
- The learning process is essentially the selection of the right regression function $f(\{x\},\{\theta\})$ where $\{\theta\}$ contains the parameters to be learned, such that the prediction made by $f(\{x\},\{\theta\})$ approximates the value returned by the supervisor uniformly over all possible input $\{x\}$.
- Regression method in this work includes linear, support vector, and DNN regressions.

Linear Regression

- The linear regression predicts eye height or width through

$$y = \{\beta\}\{x\}^{\mathrm{T}} + \beta_0$$

- where $\{x\}$ represents input parameters and $\{\beta\}$ contains weights.
- The weights are chosen by solving the least-squares problem

$$\mathrm{minimize}_{\{\beta\},\beta_0} \quad \sum_{i=1}^{N} (y_i - \hat{y}_i)^2 .$$

Support Vector Regression

- In ε-SVR, instead of minimizing the total squared errors, one hopes to make the prediction errors uniformly bounded by a parameter $\varepsilon > 0$.
- Let us start with the case when the prediction model is still linear

$$y = \{w\}\{x\}^{\mathrm{T}} + b.$$

- To promote the model simplicity, one can still minimize the quadratic function

$$E = \frac{1}{2}\{w\}\{w\}^{\mathrm{T}}.$$

- To ensure the bounded prediction error, we have

$$\left| y_i - \left(\{w\}\{x_i\}^{\mathrm{T}} + b \right) \right| \leq \varepsilon, \ \forall\, i.$$

Support Vector Regression

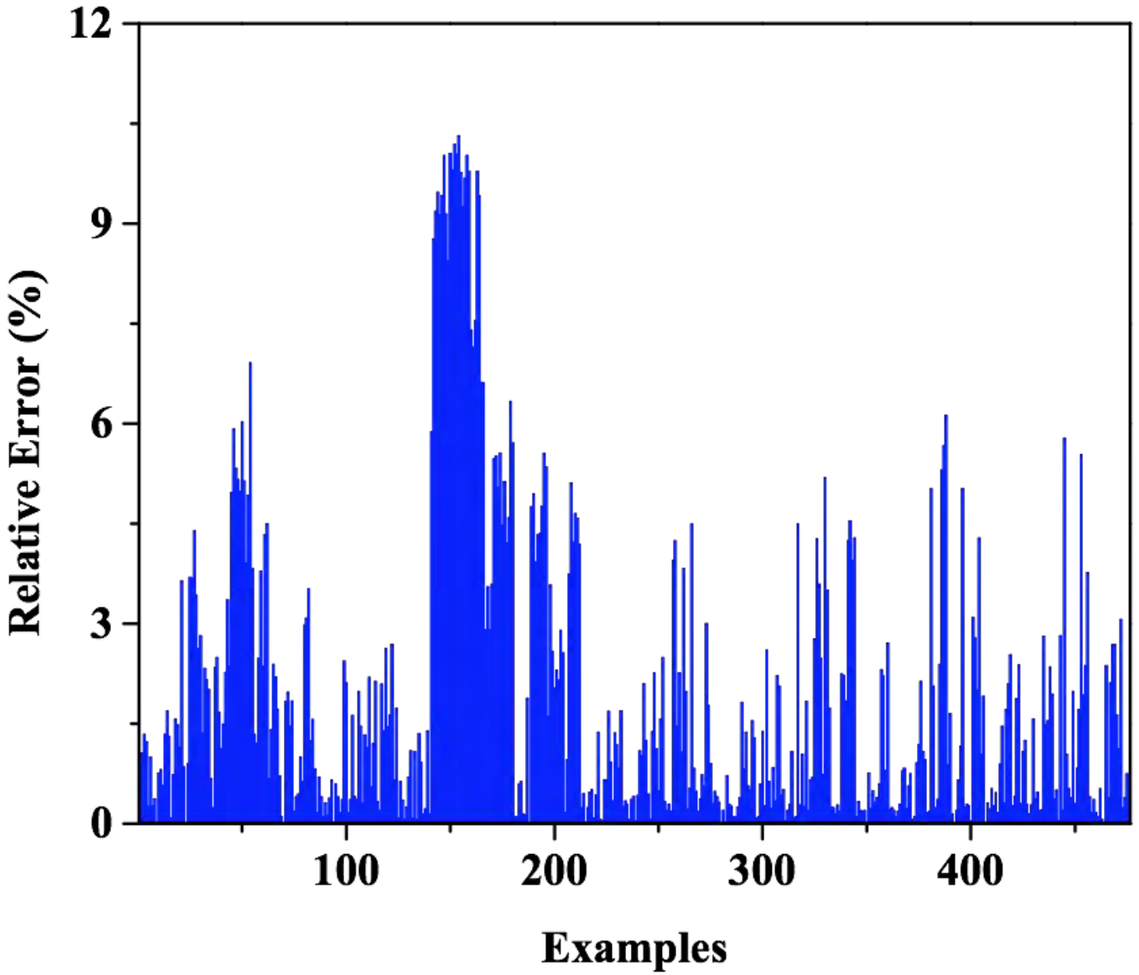
- By introducing Lagrange multipliers to handle the inequality constraints and after simplification, we arrive at the dual problem:

$$\text{maximize } g\left(\{\alpha\},\{\beta\}\right)$$
$$\text{subject to } \sum_{i=1}^{N}\alpha_i = \sum_{i=1}^{N}\beta_i,\ \alpha_i,\beta_i \in [0,C]\ \forall\, i,$$

- And the dual objective function is

$$g\left(\{\alpha\},\{\beta\}\right) = -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\left(\alpha_i-\beta_i\right)\left(\alpha_j-\beta_j\right)\{x_i\}\{x_j\}^{\mathrm{T}}$$
$$-\ \varepsilon\sum_{i=1}^{N}\left(\alpha_i+\beta_i\right)+\sum_{i=1}^{N}y_i\left(\alpha_i-\beta_i\right).$$

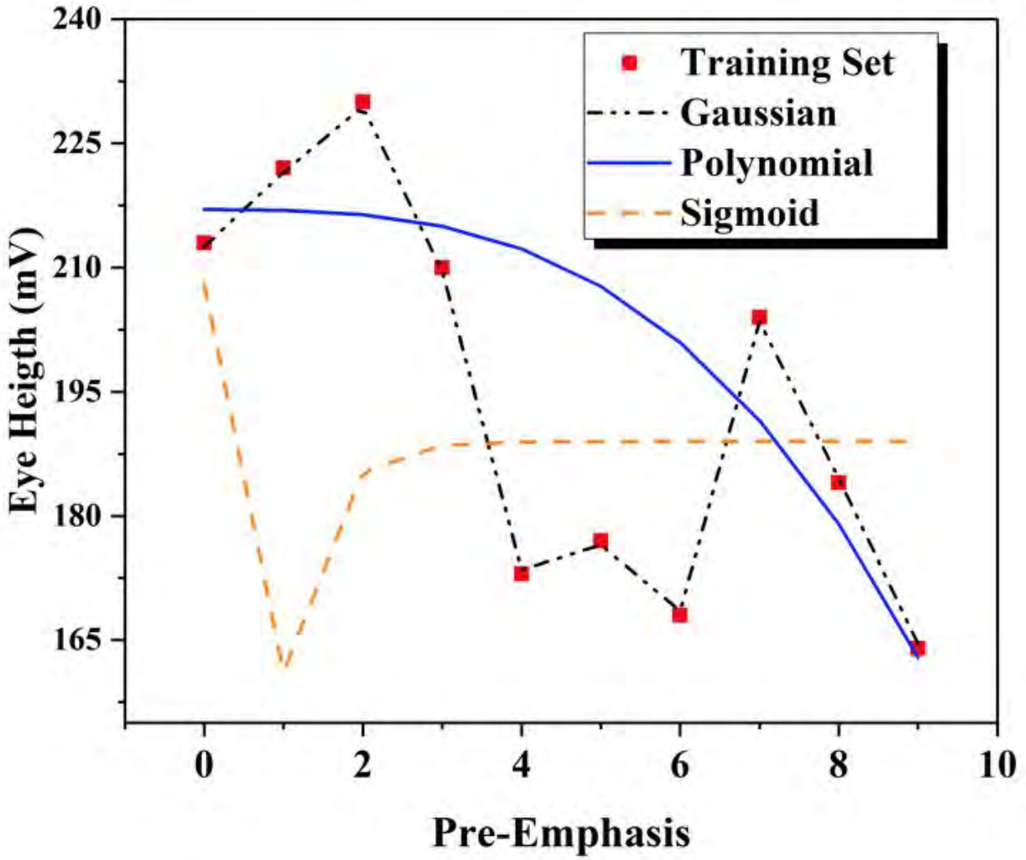
Prediction Accuracy: SVR



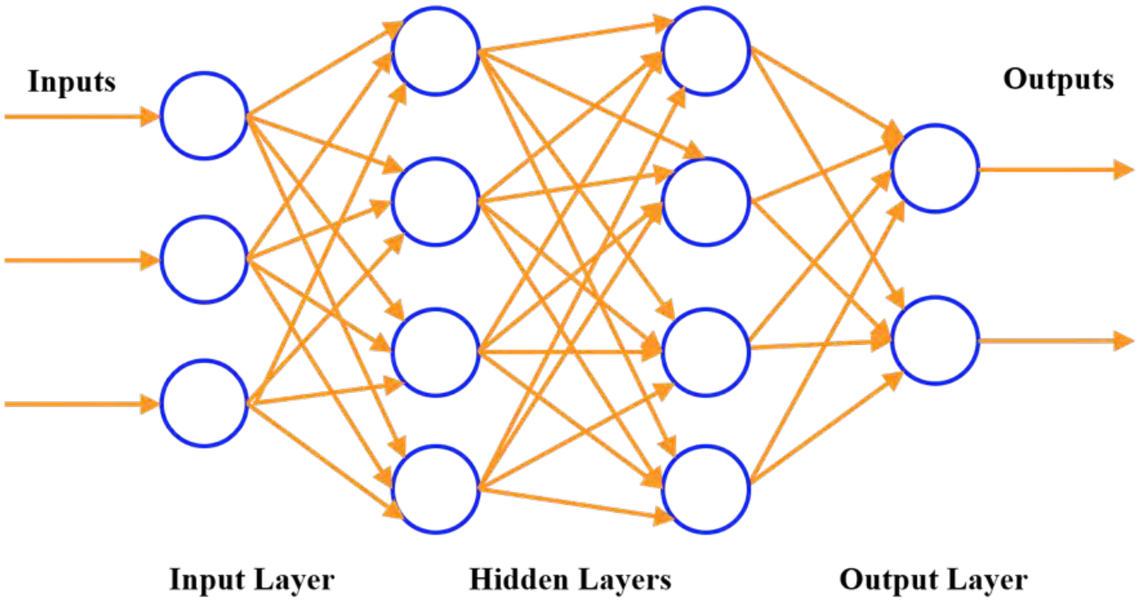
- SVR with Gaussian Kernel is trained and used to predict the eye height.
- The training, validation, and test sets contain 717, 48, and 476 examples.
- The maximum number of iterations is set to 4000.

Impact from Kernels

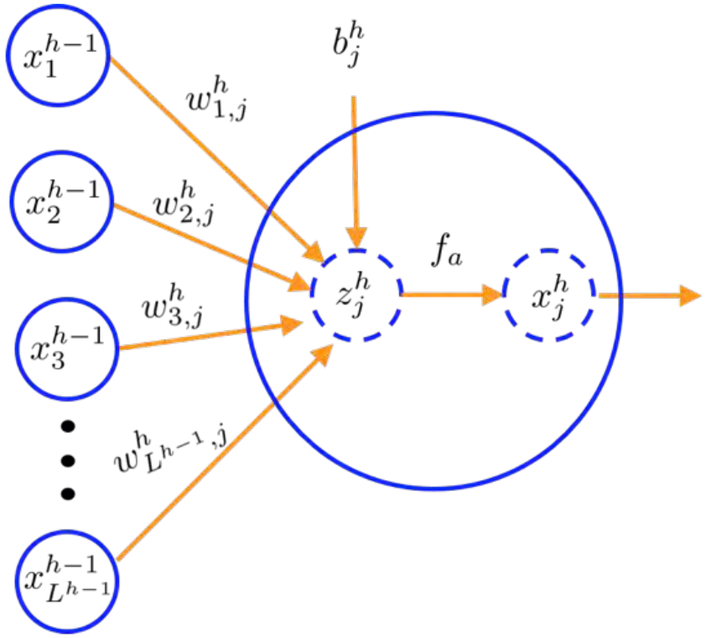
- SVR with Gaussian kernel well addresses the nonlinearity.
- Poly and linear SVR cannot handle the nonlinearity in the eye-diagram-metrics prediction.
- Polynomial kernel has degrees of three.
- Therefore, it is critical to select an appropriate kernel in order for SVR to accurately model the high-speed channel.



DNN Regression



Feedforward NN



Single Neuron

DNN Regression

- The input to the h^{th} hidden layer can be found through

$$\{z^h\} = \{x^{h-1}\}\,[W^h]\,.$$

- The output vector $\{x^h\}$ of the h^{th} hidden layer is obtained as

$$\{x^h\} = f_a\left(\{z^h\} + \{b^h\}\right).$$

- Based on the input vector, the described feedforward mechanism predicts the eye height or width, which differs from the true value by

$$\{e_i\} = \{y_i\} - \{\widehat{y_i}\}.$$

DNN Regression

- To make a good approximation, one minimizes the cost function

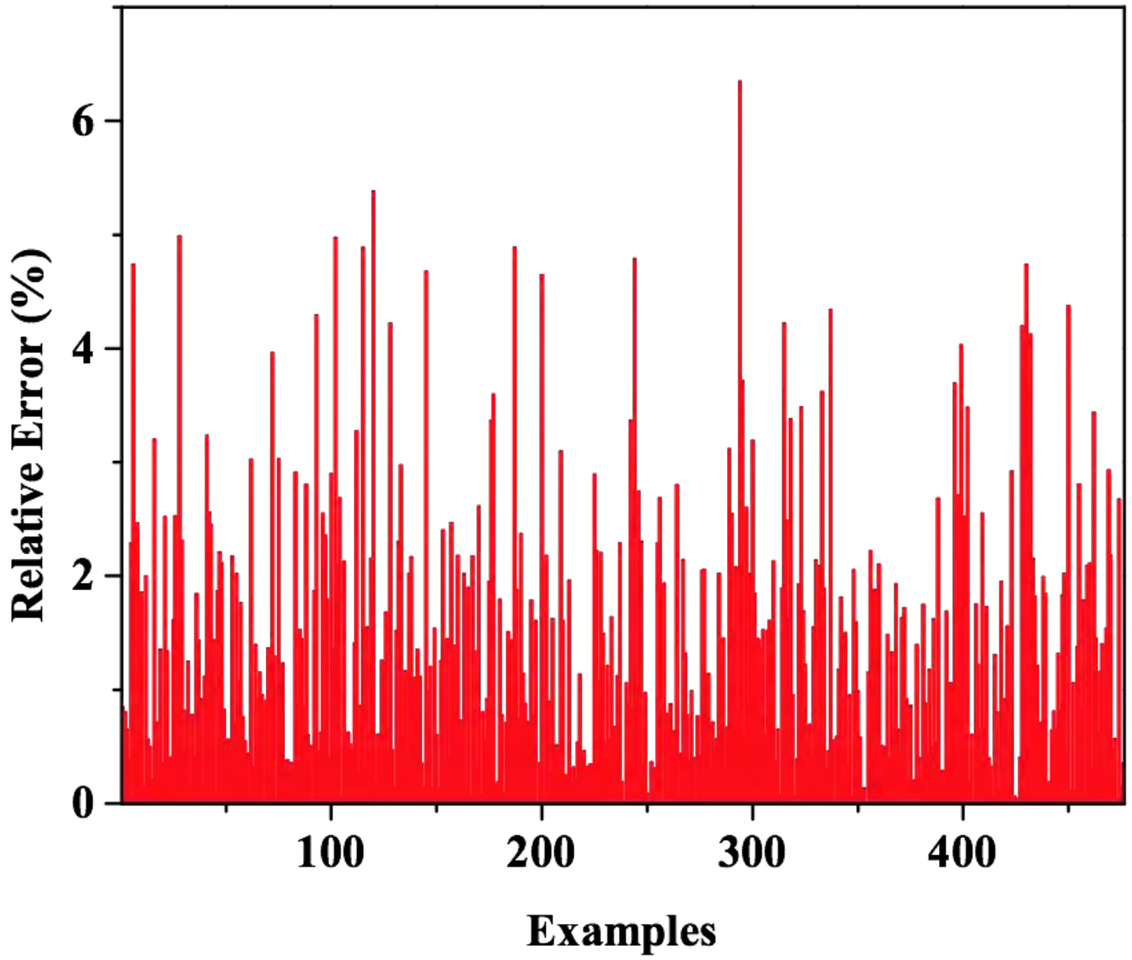
$$E = \frac{1}{2}\sum_{i=1}^{N}\{e_i\}\{e_i\}^{\mathrm{T}}.$$

- To find a local minimum of the cost function, the stochastic gradient descent (SGD) method is used, i.e.,

$$\left[W^h\right] = \left[W^h\right] - \gamma\widehat{\nabla}_{[W^h],\{b^h\}}E,\ \forall\ h = 1,\ldots,n$$

- where $\widehat{\nabla}_{[W^h],\{b^h\}}E$ is a stochastic approximation to the true $\nabla_{[W^h],\{b^h\}}E$.
- The stochastic gradients are computed efficiently by the back-propagation method.

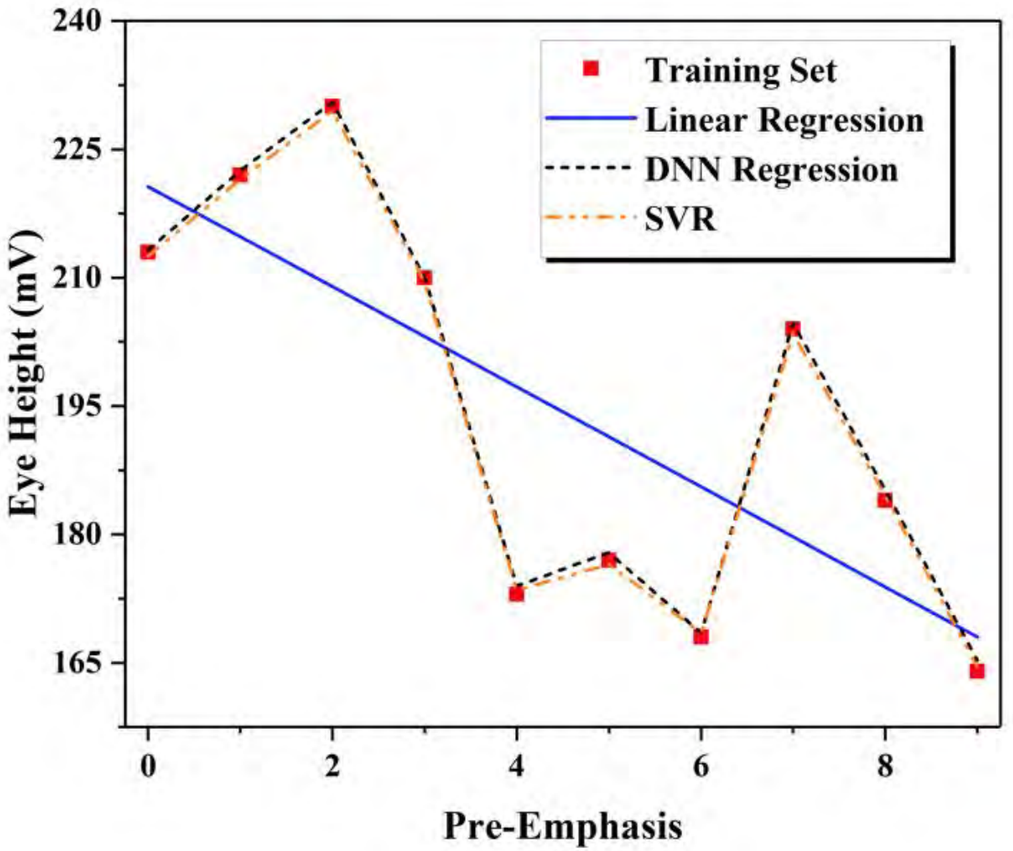
Prediction Accuracy: DNN Regression



- A DNN trained to predict the eye height.
- The training, validation, and test sets contain 717, 48, and 476 examples.
- The DNN has three hidden layers of 100, 300, and 200 nodes, respectively.
- The learning rate is chosen as 0.01 and the batch size is 25.
- The maximum number of iterations is set to 4000.

Nonlinearity

- The predicted eye heights by linear regression deviate significantly from the truth.
- Both SVR and DNN regression achieve excellent accuracy in eye-height prediction in the presence of nonlinearities.



Prediction Accuracy: DNN Regression

TABLE I: Accuracy of predicted eye heights from the DNN regression.

		Gradient Descent	Momentum	RMSProp
On Training Set	RMSE (mV)	3.1	1.9	2.6
	Maximum Relative Error (%)	6.2	4.1	5.2
On Test Set	RMSE (mV)	3.4	2.7	3.1
	Maximum Relative Error (%)	5.9	6.1	6.3

TABLE II: Accuracy of predicted eye widths from the DNN regression. A unit interval (UI) is defined as one data bit-width, regardless of data rate. Eye width is measured in the UI.

		Gradient Descent	Momentum	RMSProp
On Training Set	RMSE (UI)	0.006	0.006	0.008
	Maximum Relative Error (%)	7.9	8.1	8.7
On Test Set	RMSE (UI)	0.008	0.008	0.01
	Maximum Relative Error (%)	10.6	9.3	9.5

Prediction Accuracy: SVR

TABLE III: Accuracy of predicted eye heights from the SVR. Three different kernels are compared and the polynomial kernel has degree three. The results shown here are after standardization is applied to the data set.

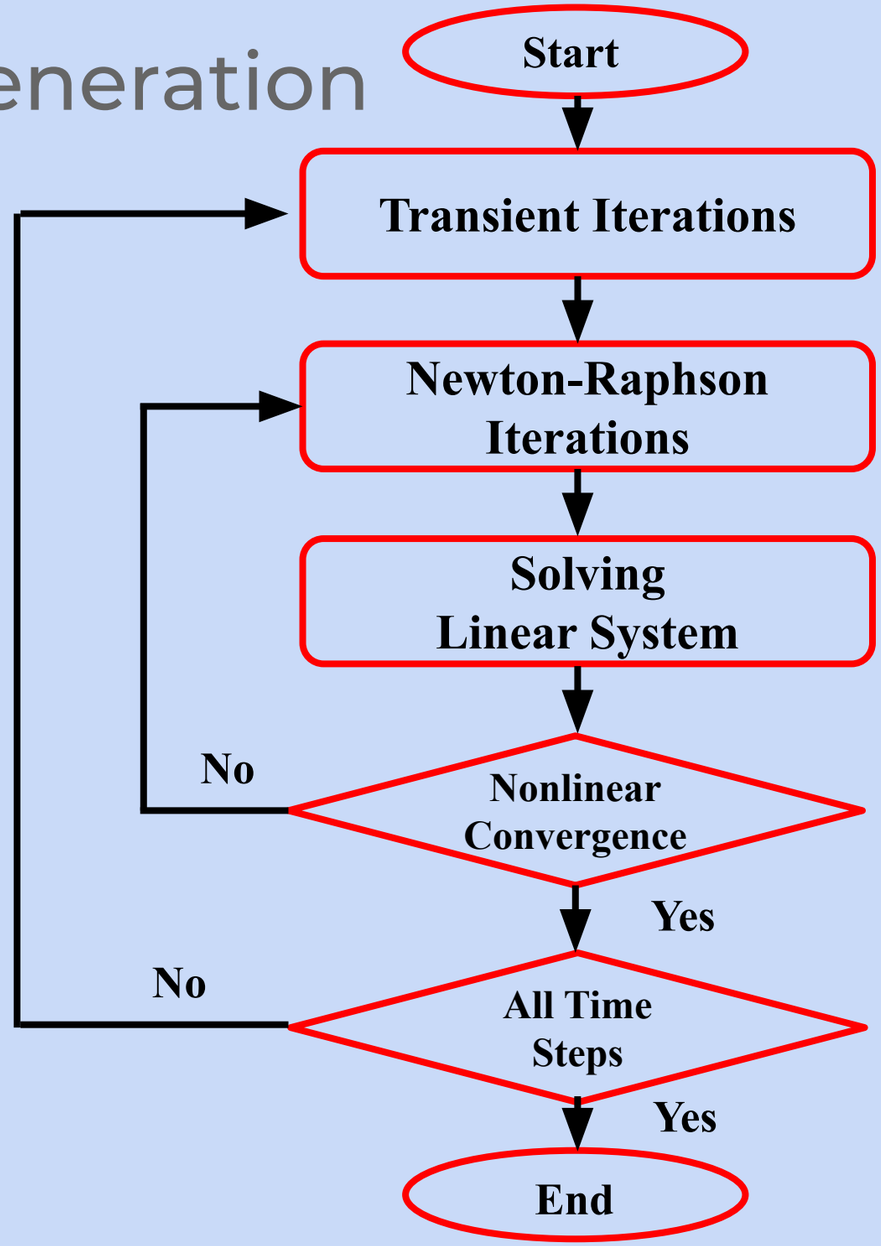
		Gaussian	Polynomial	Linear
On Training Set	RMSE (mV)	2.5	17.2	17.9
	Maximum Relative Error (%)	6.2	20.7	19.3
On Test Set	RMSE (mV)	5.7	16.7	19.3
	Maximum Relative Error (%)	10.2	19.5	17.9

TABLE IV: Accuracy of predicted eye widths from the SVR with three different kernels. A unit interval (UI) is defined as one data bit-width, regardless of data rate. Eye width is measured in the UI. The results shown here are after standardization is applied to the data set.

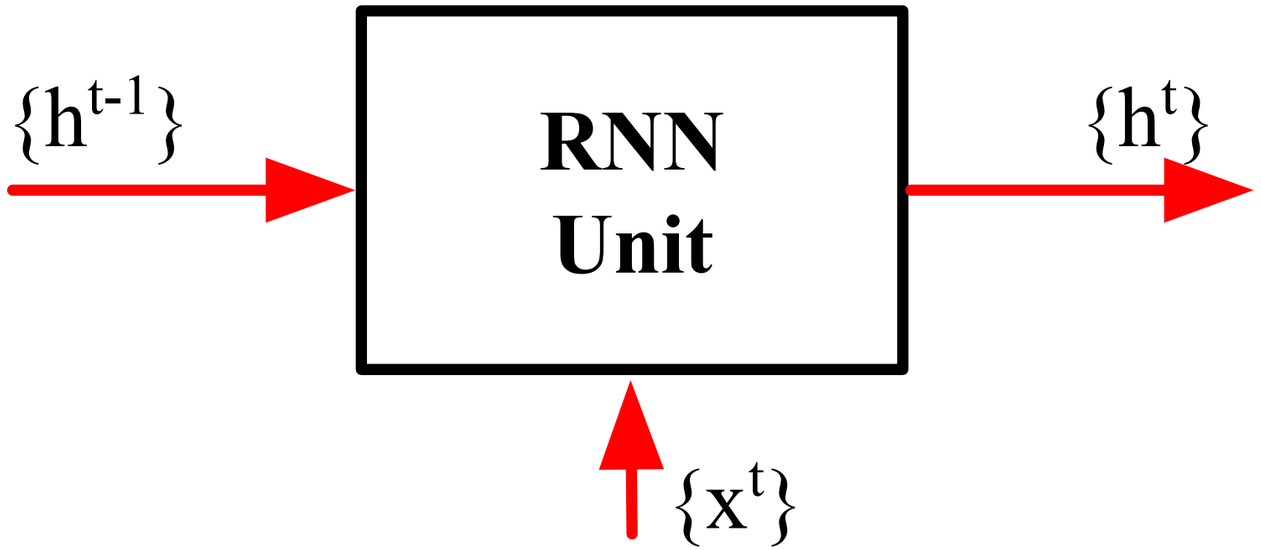
		Gaussian	Polynomial	Linear
On Training Set	RMSE (UI)	0.008	0.02	0.029
	Maximum Relative Error (%)	12.8	41.3	32.9
On Test Set	RMSE (UI)	0.04	0.038	0.081
	Maximum Relative Error (%)	23.4	22.8	33.3

Case Study: Eye Diagram Generation

In order to generate eye diagrams, transient waveforms are first obtained from a circuit simulation and then overlaid.
Generating eye diagrams by using circuit simulator can be very computationally expensive, especially with nonlinear components.
There are multiple Newton-like iterations involved at every time step when a SPICE simulator handles nonlinearities.



Recurrent Neural Network



$$\{h^t\} = \phi\left([W]\{x^t\} + [U]\{h^{t-1}\}\right)$$

Unfortunately, as the span of the temporal dependencies increases, the gradients tend to vanish or explode.

RNN with Long Short-Term Memory (LSTM) Unit

LSTM utilizes a more sophisticated activation achieved by gating signals to resolve this issue

- Input gate

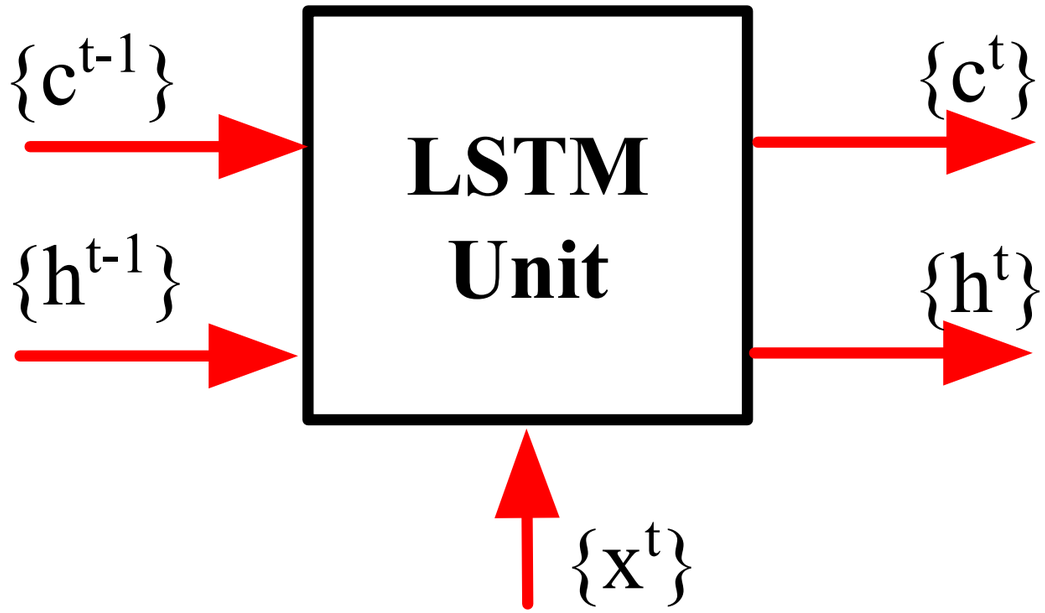
$$\{i^t\} = \sigma\left([W_i]\{x^t\} + [U_i]\{h^{t-1}\}\right)$$

- Forget gate

$$\{f^t\} = \sigma\left([W_f]\{x^t\} + [U_f]\{h^{t-1}\}\right)$$

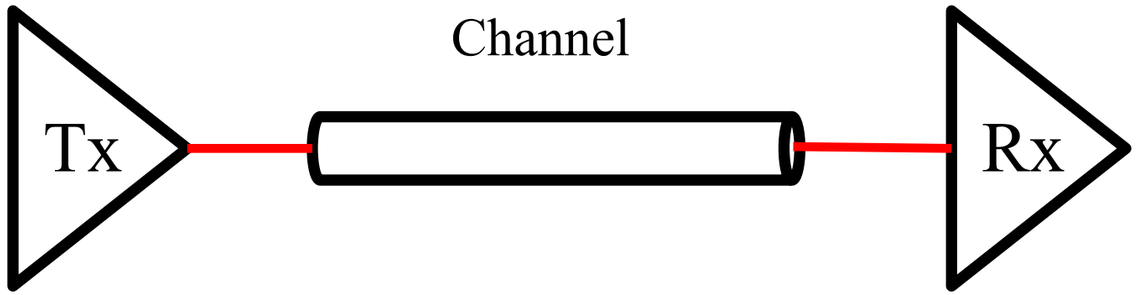
- Output gate

$$\{o^t\} = \sigma\left([W_o]\{x^t\} + [U_o]\{h^{t-1}\}\right)$$

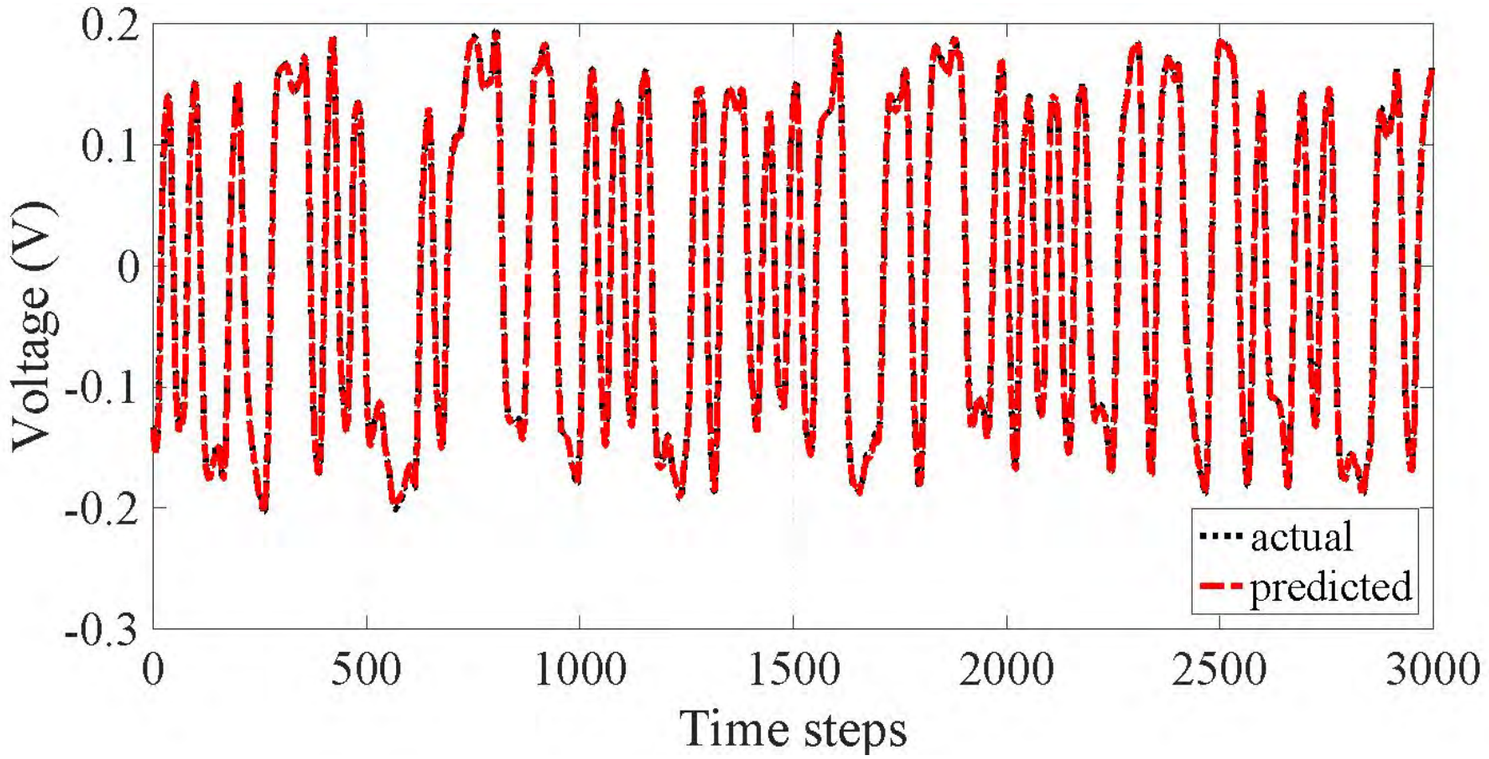


Eye Diagram Generation

- A high-speed channel consists of transmitter, receiver, and interconnect model.
- Simulate the voltage at transmitter and receiver with a circuit simulator at a total number of *N* time steps.
- Divide the voltage at N time steps into three subsets, namely, the training set, the validation set, and the test set.
- Train a RNN model with the training set that can predict the transient voltage at both the transmitter and receiver ends on the unseen test set.

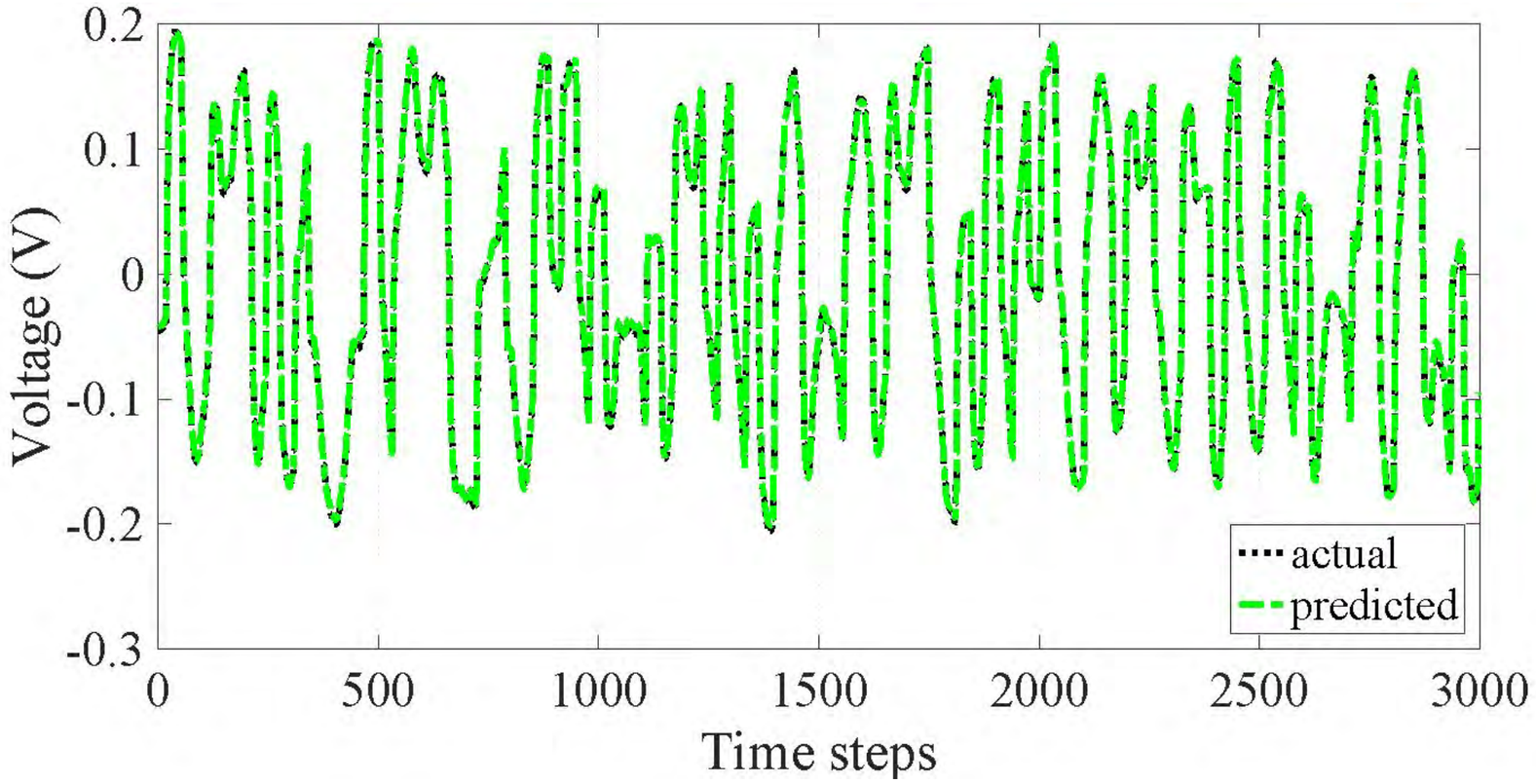


Predicted Voltage on Receiver End



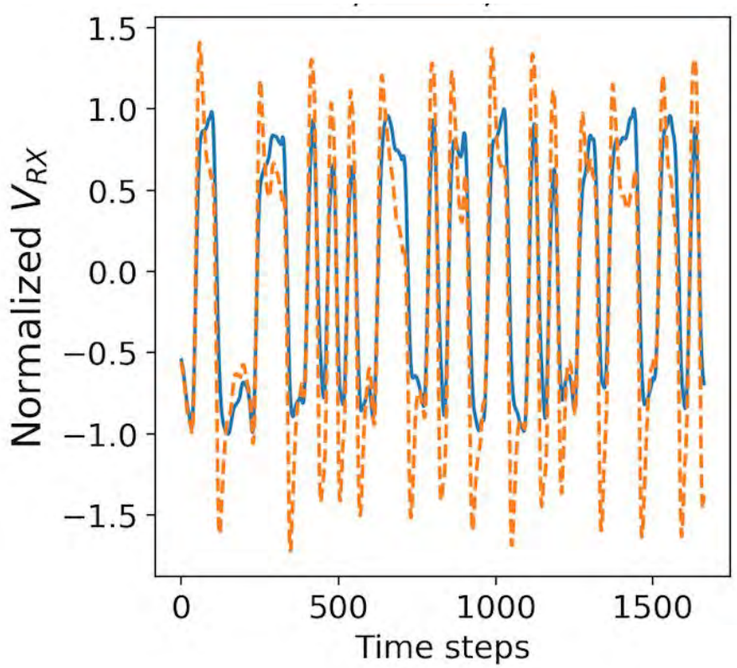
- In this example, a stack of four LSTM cells of 20 hidden units is used.
- The optimization method used is Adam with 0.3 dropout regularization.
- The time steps for training is 11,000 and the model converges in 48 epochs.
- Accurate predictions are achieved on unseen sequence as shown in the Figure.

Predicted Voltage on Transmitter End

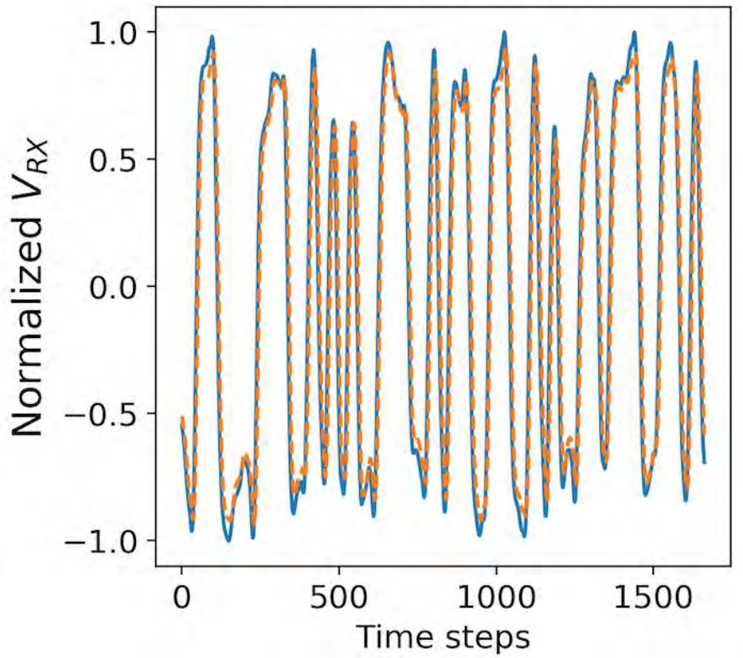


- In this example, a stack of four LSTM cells of 20 hidden units is used.
- The optimization method used is Adam with 0.3 dropout regularization.
- The time steps for training is 11,000 and the model converges in 48 epochs.
- Accurate predictions are achieved on unseen sequence as shown in the Figure.

Prediction Accuracy

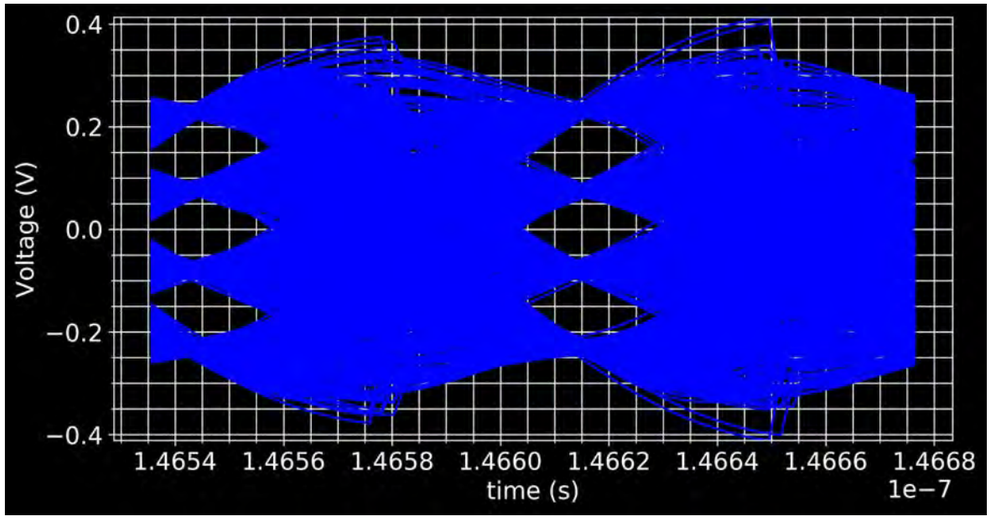


RNN

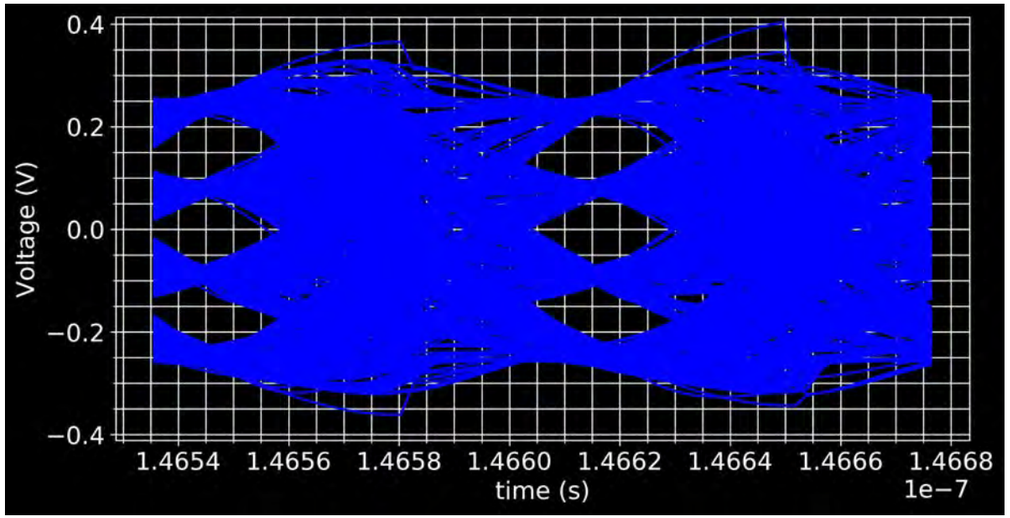


LSTM network

Eye Diagram Generation with a PAM4 Example



Circuit Simulator



LSTM Network

T. Nguyen, T. Lu, K. Wu, J. Schutt-Aine, "Transient simulations of high-speed channels with recurrent neural network," *in IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems.*

Conclusion

- Propose using machine learning methods to SI and PI including Decap placement, PDN impedance prediction, eye-diagram metrics predictions, and eye-diagram generation.
- With neural network models, the computation efficiency is significantly improved as no complex simulation is required.
- The neural network models make predictions through inference and it requires no substantial domain knowledge.

PUBLICATIONS

[A] Stefano Piersanti, Riccardo Cecchetti, Carlo Olivieri; Francesco de Paulis; Antonio Orlandi; Markus Bueker, **Decoupling Capacitors Placement at Board Level Adopting a Nature-Inspired Algorithm**, in Electronics 2019, Volume 8, Issue 7, October 2019, available online: https://www.mdpi.com/2079-9292/8/7/737/pdf

[B] Francesco de Paulis; Riccardo Cecchetti; Carlo Olivieri; Stefano Piersanti; Antonio Orlandi; Markus Bueker, **Efficient Iterative Process based on an Improved Genetic Algorithm for Decoupling Capacitor Placement at Board Level**, in Electronics 2019, Volume 8, Issue 11, available online: https://www.mdpi.com/2079-9292/8/11/1219/pdf

[C] R. Cecchetti, F. de Paulis, C. Olivieri, A. Orlandi, M. Buecker, "**Effective PCB Decoupling Optimization by Combining an Iterative Genetic Algorithm and Machine Learning**" in Electronics 2020, Volume. 9, Issue 8, August 2020, available online: https://www.mdpi.com/2079-9292/9/8/1243

[D] F. de Paulis, R. Cecchetti, C. Olivieri and M. Buecker, "**Genetic Algorithm PDN Optimization based on Minimum Number of Decoupling Capacitors Applied to Arbitrary Target Impedance**," 2020 IEEE International Symposium on Electromagnetic Compatibility & Signal/Power Integrity (EMCSI), Reno, NV, USA, 2020, pp. 428-433, doi: 10.1109/EMCSI38923.2020.9191458. "**Best SIPI Symposium Paper Award**"

[E] Lu, Tianjian, Ju Sun, Ken Wu, and Zhiping Yang. "**High-speed channel modeling with machine learning methods for signal integrity analysis**." IEEE Transactions on Electromagnetic Compatibility 60, no. 6 (2018): 1957-1964.

[F] Nguyen, Thong, Tianjian Lu, Ju Sun, Quang Le, Ken We, and Jose Schut-Aine. "T**ransient simulation for high-speed channels with recurrent neural network**." In 2018 IEEE 27th Conference on Electrical Performance of Electronic Packaging and Systems (EPEPS), pp. 303-305. IEEE, 2018.

[G] L. Zhang, J. Juang, Z. Kiguradze, B. Pu, S. Jin, S. Wu, Z. Yang, and C. Hwang, "**Fast PDN Impedance Prediction using Deep Learning**", submitted to International Journal of Numerical Modeling: Electronic Networks, Devices and Fields.

[H] J. Juang, L. Zhang, Z. Kiguradze, B. Pu, S. Jin, S. Wu, Z. Yang, and C. Hwang, "**A Modified Genetic Algorithm for the Selection of Decoupling Capacitors in PDN Design**", accepted to IEEE EMC+ SIPI 2021.

[I] L. Zhang, J. Juang, Z. Kiguradze, B. Pu, S. Jin, S. Wu, Z. Yang, and C. Hwang, "**Efficient DC and AC Impedance Calculation for Arbitrary-shape and Multi-layer PDN Using Boundary Integration**," IEEE Trans. Electromagn. Compat., to be submitted.

[J] L. Zhang, J. Juang, Z. Kiguradze, S. Jin, S. Wu, Z. Yang, J. Fan, C. Hwang, **"PCB-Level Decap Placement Using Deep Reinforcement Learning**", IEEE Trans Microw Theory Tech., to be submitted.

Thank you