Machine Learning: An enabling technology for electronics modeling and design optimization

ELYSE ROSENBAUM
Electrical & Computer Engineering
University of Illinois at Urbana-Champaign
February 1, 2017
What is Machine Learning?
The View from Popular Culture

• Not distinguished from AI
• Intelligent machines
  – With beyond-human capabilities
• Hollywood’s vision:
The View from Popular Culture

- Not distinguished from AI
- Intelligent machines
  - With beyond-human capabilities
- Overwhelmingly identified with neural nets
  - As far back as 1968

"The relays are not unlike the synapses of the human brain."
Some Studies in Machine Learning
Using the Game of Checkers

Abstract: Two machine-learning procedures have been investigated in some detail using the game of checkers. Enough work has been done to verify the fact that a computer can be programmed so that it will learn to play a better game of checkers than can be played by the person who wrote the program. Furthermore, it can learn to do this in a remarkably short period of time (8 or 10 hours of machine-playing time) when given only the rules of the game, a sense of direction, and a redundant and incomplete list of parameters which are thought to have something to do with the game, but whose correct signs and relative weights are unknown and unspecified. The principles of machine learning verified by these experiments are, of course, applicable to many other situations.
• The application of *statistical learning theory* to construct accurate predictors \((f: \text{inputs} \rightarrow \text{outputs})\) from data

• Today, this is feasible even with large number of inputs ("features") due to the availability of powerful computing machines
  – Optimization solvers
  – Parallel programming
How does statistical learning differ from “classical” statistics?
Vanilla Statistics

• Model the MOSFET threshold voltage shift following the removal of bias-temperature stress
• Physics-based model adopted with confidence
• Parameter values are extracted from measurement data
  – Regression (“least squares estimation”)
• Domain knowledge used to select the model class
  – Rational function: \( Z(s) = \frac{a(s)}{b(s)} \)
• A priori, I do not know what order of \( b(s) \) is needed to represent the data (# of poles)
  – If I set the order of \( b(s) \) high, I risk overfitting (e.g., fitting to measurement noise)
• Goal of SLT is to predict input-output mapping (not merely parameter extraction)
  – Cost functions penalize both under and over fitting
Cost Functions used in Machine Learning

• Example:

\[ C(\theta) = \frac{1}{n} \sum_{i=1}^{n} \left( h_{\theta}(X^{(i)}) - Y^{(i)} \right)^2 + \lambda \sum_{j=1}^{p} \theta_j^2 \]

– Least squares fitting + L2 regularization

• Regularization + sufficiently high \( p \) used to find the Goldilocks solution

Scikit-learn.org
Machine Learning Analyses and Algorithms

- Linear regression
- Logistic regression
- Neural networks
- Kernel methods
- Supervised learning
- Unsupervised learning
- Principal components analysis
- Stochastic gradient descent
- Back propagation
- And more …

- First step: Identify the right set of tools for your application
• Design respins have not been eliminated
• Many of the observed failures during qualification testing are the direct result of an insufficient modeling capability
  – Sources of such failures include mistuned analog circuits, signal timing errors, reliability problems, and crosstalk \(^1\)
  – Variability cannot be modeled in a manner that is both accurate and computationally efficient
• Simulation-based design optimization has had only limited success
  – Simulation “in-the-design-loop” often too slow and leads to impractical designs
• Proposal: use machine learning algorithms to overcome those hurdles!

\(^1\) Harry Foster, “2012 Wilson Research Group Functional Verification Study,”
• Machine learning can be applied in ways that do not make our lives more pleasant

• However, using ML to extract models needed for electronic design automation will make our professional lives better
  – Design optimization
  – Shorter time-to-market

I want to talk to a real person!!!
Thermal Design Optimization for 3D-IC

- Several parameters (e.g., $k_{\text{TIM}}$, #TSV, fan-speed) must be optimized to limit the clock skew (< 110 ps)
- Used ML-based Bayesian Optimization
- More generally, seek to find $\mathbf{X}_{\text{opt}} = \arg\min_{\mathbf{X}} (f(\mathbf{X}))$. Accurate modeling of $f(\mathbf{X})$ needed only near min
1-D Demo of Bayesian Optimization

True function: \( f(x) = (6x - 2)^2 \sin(12x - 4) \) where \( x \in [0,1] \)

Plot generated by P. Franzon (CAEML/NCSU) using python GPyOpt package
1-D Demo of Bayesian Optimization

True function: $f(x) = (6x - 2)^2 \sin(12x - 4)$ where $x \in [0,1]$
True function: \( f(x) = (6x - 2)^2 \sin(12x - 4) \) where \( x \in [0,1] \)
1-D Demo of Bayesian Optimization

True function: \( f(x) = (6x - 2)^2 \sin(12x - 4) \) where \( x \in [0,1] \)

Plot generated by P. Franzon (CAEML/NCSU) using python GPyOpt package
1-D Demo of Bayesian Optimization

True function: \( f(x) = (6x - 2)^2 \sin(12x - 4) \) where \( x \in [0,1] \)

Plot generated by P. Franzon (CAEML/NCSU) using python GPyOpt package
True function: $f(x) = (6x - 2)^2 \sin(12x - 4)$ where $x \in [0, 1]$
1-D Demo of Bayesian Optimization

True function: \( f(x) = (6x - 2)^2 \sin(12x - 4) \) where \( x \in [0,1] \)
True function: \( f(x) = (6x - 2)^2 \sin(12x - 4) \) where \( x \in [0,1] \)
1-D Demo of Bayesian Optimization

True function: \( f(x) = (6x - 2)^2 \sin(12x - 4) \) where \( x \in [0,1] \)

Plot generated by P. Franzon (CAEML/NCSU) using python GPyOpt package
1-D Demo of Bayesian Optimization

True function: \( f(x) = (6x - 2)^2 \sin(12x - 4) \) where \( x \in [0,1] \)

Plot generated by P. Franzon (CAEML/NCSU) using python GPyOpt package
1-D Demo of Bayesian Optimization

True function: $f(x) = (6x - 2)^2 \sin(12x - 4)$ where $x \in [0,1]$
1-D Demo of Bayesian Optimization

True function: \( f(x) = (6x - 2)^2 \sin(12x - 4) \) where \( x \in [0,1] \)
1-D Demo of Bayesian Optimization

True function: \( f(x) = (6x - 2)^2 \sin(12x - 4) \) where \( x \in [0,1] \)

Plot generated by P. Franzon (CAEML/NCSU) using python GPyOpt package
1-D Demo of Bayesian Optimization

True function: \( f(x) = (6x - 2)^2 \sin(12x - 4) \) where \( x \in [0,1] \)

Plot generated by P. Franzon (CAEML/NCSU) using python GPyOpt package
1-D Demo of Bayesian Optimization

True function: $f(x) = (6x - 2)^2 \sin(12x - 4)$ where $x \in [0,1]$
1-D Demo of Bayesian Optimization

True function: $f(x) = (6x - 2)^2 \sin(12x - 4)$ where $x \in [0,1]$
Black-box Component Models for System Design

- Fast-to-simulate models of components needed for system design
- Black-box or *behavioral* models replace the physical models
- Surrogate modeling methodology developed to support this

Surrogate Modeling Procedure

• Generally uses **non-parametric regression** in which the data determine the model structure
  – In essence, the model is constructed by interpolation using splines, kernels, kriging …
  – No requirement that variance be constant or that the data follow a normal distribution

• Neural network may be used
Goal: Identify the optimal set of calibration knobs for this design

Surrogate Model Based Circuit Design

Multi-dimensional response surfaces

Perform:
Sensitivity Analysis
Simulation-based sampling
Model fitting

7D : 3 responses, 4 “knobs”

Fourth knob (Vc) too sensitive to use

Ref: P. Franzon (CAEML / NCSU)
Time-Delay Neural Network Model of Power Amp

NN used to learn I-O mapping for baseband in-phase and quadrature signals


Model validation: Measurement vs. Model for a Class AB PA (IS95 input signal)
• RNN are proven to approximate any system which can be represented by a smooth nonlinear state space model, i.e.,

\[
\begin{align*}
\dot{x} &= f(x, u) \\
y &= g(x, u)
\end{align*}
\]

\(u\) is input, \(y\) is output, \(x\) is state

• Many circuits and devices can be represented by state space models
RNN for Buffer Chip Modeling

Y. Cao et al. in *Trans-MTT*, June 2006.

RNN model simulates 12x faster than transistor-level model (HSPICE sims)
Mechanistic vs. Empirical Models

• “Mechanistic models can (1) contribute to scientific understanding, (2) provide a basis for extrapolation, (3) provide a representation of the response function that is more parsimonious than the one attainable empirically.” Box, Hunter and Hunter, in *Statistics for Experimenters*, Wiley, 1978.

• Empirical model is an approximation of the true I-O function
  – Do not use when a physical I-O relation can be formulated
  – Use empirical model when complexity and dimensionality of problem make the physics-based approach infeasible
  – Example: Behavioral model for 16 high-speed input receivers that are coupled together through the common PDN
• Illustration: a poor use of behavioral / surrogate / empirical modeling

• Learn an RNN model of an RC circuit
  – Model is simulated in Spectre

• The RNN approximates the true state-space model with less than 100% accuracy
  – And is less computationally efficient
• Illustration: a potentially good use of behavioral / surrogate / empirical modeling

• Model of a LIN driver pin is extracted from pulsed I-V (+ package parasitics from IBIS)

• Transient sims (at right) have poor fidelity
  – Next, authors manually augment model with a non-linear time-variant capacitor and a parallel RL
  – Fit improved but not validated for other waveforms

• ML provides a more rigorous approach

• Machine learning can be used to derive generative models as well as discriminative models
• Generative model represents the joint pdf $p(X, Y)$ and can be used to generate new samples (i.e. synthetic data)
• May be applied to represent process/manufacturing variations
• S-parameter model of coupled microstrip lines (S41 shown)
• 50 training samples (red)
  – Line widths and dielectric permittivity vary, $\sigma = 0.1 \cdot \bar{x}$
• 500 generated samples (blue)
• 475 validation samples (green)

S. De Ritter et al., in EDAPS 2016.
Complete, machine learning pipeline:
1. Data acquisition and storage
2. Data selection and filtering
3. Cost function specification and selection of model classes
4. Model fitting to training data
5. Model evaluation and validation

End-to-end performance is key!
• Our vision:
  – To enable fast, accurate design and verification of microelectronic circuits and systems by creating machine-learning algorithms to derive models used for electronic design automation

• An NSF I/UCRC
  – Industry/University Cooperative Research Center
  – Three sites
    – University of Illinois at Urbana-Champaign
    – Georgia Tech
    – North Carolina State University
CAEML Expected Outcomes

• Model-based design becomes faster and more accurate
  – Result: designs that are more reliable, lower power, smaller area

• Simulation-based design verification is more accurate
  – Products pass qualification testing the first time; shorter time-to-market

• Computationally efficient system-level analysis of manufacturing variations
  – Eliminate excessive guard-banding
Projects Supported by CAEML (current)

- Modular machine learning (theory)
- High-speed links
- Power delivery
- System-level ESD
- IP reuse
- Design rule checking

- Future: capacitor placement optimization, compact modeling, counterfeit/tamper detection ....