Online Training of Deep Neural Networks on Simulated Neuromorphic Hardware

Prasanna Date
Rensselaer Polytechnic Institute
datep@rpi.edu

Overview

- Current Work - Design Index, Predicting Node Failures
- Online training of deep neural networks on simulated neuromorphic hardware
- Online training algorithms
- Integer Programming techniques for obtaining integral weights
- Conclusion & Future work

Design Index for Deep Neural Networks (DNN)

Tuple of indices that serves as a performance metric to assess the quality of DNN

- Accuracy Index: $$\boldsymbol{I_a = log_{10}\left(\frac{e_p}{e_t}\right)}$$
- Overfitting Index: $$\boldsymbol{I_o = log_{10}\left(\frac{e_v}{e_t}\right)}$$

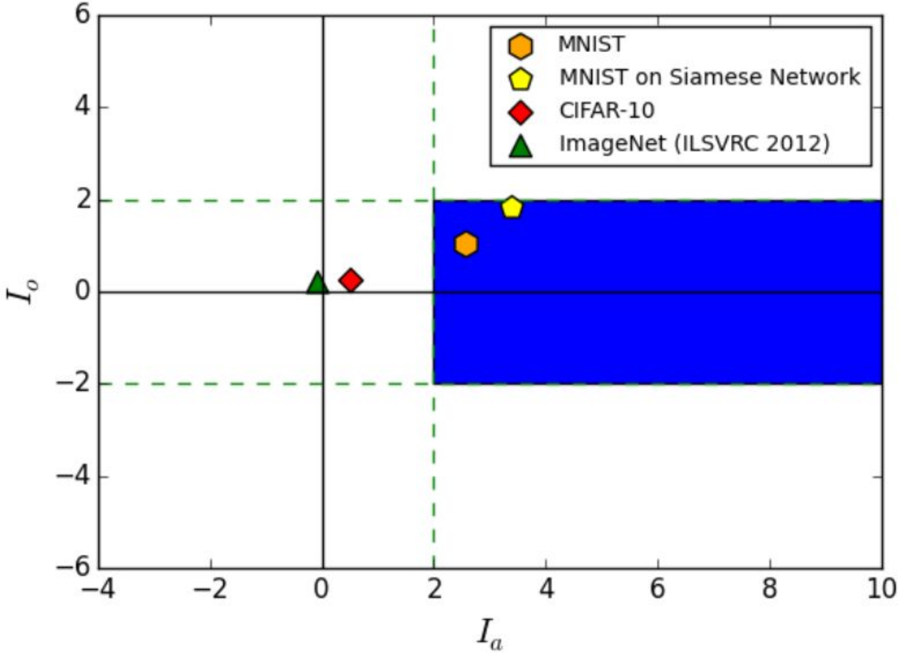
e_p: Threshold Error

e_t: Training Error

e_v: Validation Error

Design Index for Deep Neural Networks (DNN)

Benchmark Results



Node Failure Prediction on Supercomputers

- Aim: To be able to predict node failures on supercomputing platforms using machine learning techniques
- One year worth of log files from the RPI Blue Gene/L supercomputer
- Comparative analysis of several machine learning techniques:
  - Regression
  - K-Nearest Neighbors (KNN)
  - Support Vector Machines (SVM)
  - Deep Neural Networks (DNN)
  - Spiking Neural Networks (SNN) running on Neuromorphic Hardware

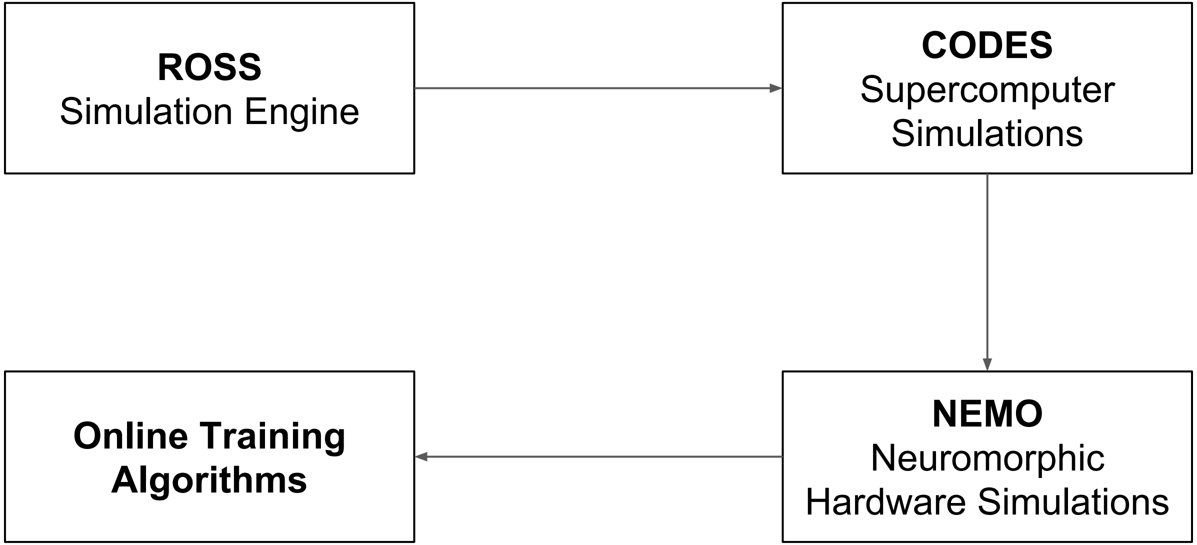
DNN for Node Failure Prediction on Supercomputers

Machine Learning Technique	Training Accuracy (%)	Validation Accuracy (%)	Test Accuracy (%)
Linear Regression	93.59	98.63	94.52
K-NN (K = 3)	97.84	98.67	96.75
K-NN (K = $\sqrt{83{,}903}$ = 289)	94.14	98.48	94.97
SVM	92.85	97.65	93.75
DNN	96.57	98.54	96.23
SNN	**95.8**	**97.98**	**95.4**

Online Training of DNN on Simulated Neuromorphic Hardware

- Online Training of DNN is the direction to go in Deep Learning research
- Want to enable online training of DNN on simulated neuromorphic hardware
- Neuromorphic hardware requires integral weights
- Use Integer Programming techniques to obtain integral weights

Online Training of DNN on Simulated Neuromorphic Hardware



Online Training Algorithms

- Online training algorithms return non-integral weights
- Need strictly integral weights for Neuromorphic hardware
- Use Branch and Bound (Integer Programming) coupled with Backpropagation to obtain integral weights

Integer Programming to obtain Integral Weights

- Traditional Backpropagation algorithm (and all its variants) return non-integral weights
- Use Backpropagation coupled with Branch and Bound Technique to obtain integral weights
- Downside: can result in exponential time complexity
- Can be improved with heuristics
- Getting integral weights for Logistic Regression model for linearly separable and linearly inseparable data

Integer Programming to obtain Integral Weights

Integral Weights for Logistic Regression Model on MNIST dataset

- State of the art accuracy: 92.4 %
- Accuracy with non-integral weights: 92.3 %
- Accuracy with integral weights: **91.6 %**

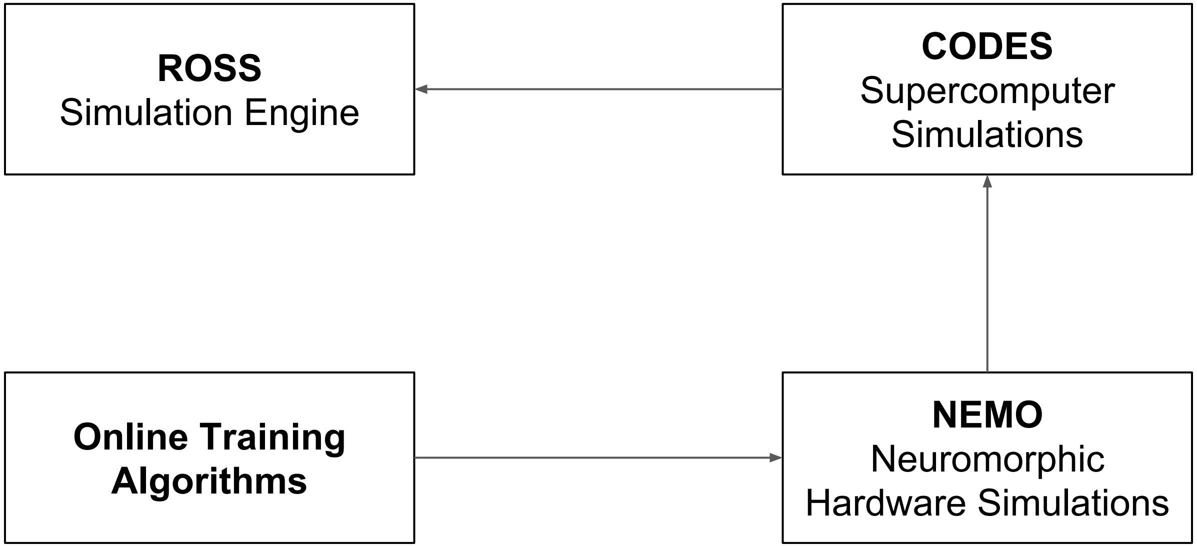
Conclusion

- Design Index for Deep Neural Networks
- Node failure prediction on supercomputers: A comparative analysis of Machine Learning techniques
- Branch and Bound coupled with Backpropagation to obtain integral weights for spiking neural networks

Future Work

- Integral weights for Convolutional Neural Networks (CNN) on MNIST using Branch and Bound - Summer 2017
- Implement the algorithm (Branch and Bound coupled with Backpropagation) on simulated neuromorphic hardware (NeMo) - Summer 2018

Future Work



ROSS
Simulation Engine
CODES
Supercomputer
Simulations
Online Training
Algorithms
NEMO
Neuromorphic
Hardware Simulations

Comments & Questions

Thank You!

Prasanna Date
Rensselaer Polytechnic Institute
datep@rpi.edu