Using LASSO to Optimize SNN Decoders on Mixed-signal Hardware

Science Research Program

C. Spencer Bowles, Mourad Musallam, Jeffrey S. Young
Center for Research into Novel Computing Hierarchies, Georigia Institute of Technology, Atlanta, GA

2022

Abstract

The operation of efficient methods of computation is imperative to reduce the energy consumption of computation. SNNs, LASSO, and mixed-signal architectures, of which FPAAs and Braindrop are explored, provide methods to decrease the power consumption of running neural networks. A Nengo model can be constructed to deploy a network to several architectures, where compression and flattening of input SHD elements are necesitated to meet memory requirements. On the SDH dataset, the model proposed has RMSEs between 0.11 and 0.17 for digits. Unfortunately, neither novel architecture could be tested for power efficiency. For the FPAA, issues with the Reconfigurable Analog Signal Processor (RASP) toolchain prevented development of a hardware description designed for the FPAA. For Braindrop, the server hosting the architecture went offline before development of the Nengo model could be finished.

Introduction

The operation of efficient methods of computation is imperative to reduce the energy consumption of computation. As the rate of reductions in transistor size have slowed, so has the rate of reductions in transistor power consumption. Additionally, with increases in data collection, requirements for processing power have increased likewise. As such, researchers attempt to find additional methods to reduce computer power consumption beyond reducing transistor size. Two such methods are computing using event-based neural networks and operating on mixed-signal architectures.

The efficiency of biological methods of computation, such as the mammalian brain, inspire researchers to model elements present in biological systems. Biological systems operate on continuous values in parallel rather; however, they operate slower, with higher variability, and lower reproducibility than modern, von Neumann digital systems. In this work, researchers will propose the usage of the least absolute shrinkage and selection operator operating on spiking neural network output, all operating on various mixed-signal architectures.

Spiking Neural Networks (SNN)

For the development of more biologically inspired neural networks, neuron, synapse, and network models are all reconsidered to implement in more biologically accurate manners. From a standard artificial neural network (ANN) design, SNNs exchange McCulloch-Pitts neuron for event-based neurons, such as the leaky integrate-and-fire neuron model, and feed-forward networks for recurrent network architectures. Given these modifications, SNNs access a continuous time domain not present in ANNs, which at most operate on a distinct clock. In an LIF neuron, instantaneous membrane potential accumulates with an input signal, and can fire the neuron if a threshold potential is met.

LASSO

The least absolute shrinkage and selection operator (LASSO) is a method of regression analysis which performs both variable selection and regularization. LASSO operates by performing ordinary least squares regression subject to an L1-norm constraint. This is similar to ridge regression, which has an L2-norm constraint instead; however, an L1-norm constraint allows elements to be sent to 0, hence the selection in LASSO. This allows LASSO to eliminate the impact of correlated covariates. Additionally, LASSO operates under the assumption that the output vector dimension m is much smaller than the input vector dimension n, $m \ll n$, and that the input vector X has high sparsity. Where y is the output vector, the lagrangian form of LASSO is,

$$\min_{\beta \in \mathbb{R}^m} \frac{1}{n} ||y - X\beta||_2^2 + \lambda ||\beta||_1$$

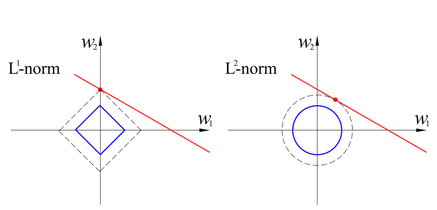


Figure 1: A comparison between LASSO and Ridge Regression constraints for a 2-dimensional vector. From [3].

Nengo

Nengo is an open-source framework for the creation, simulation, and deployment of SNNs. Nengo nodes present vectors of real-valued signals to neurons in the model. An ensemble is a collection of neurons with gains and biases, by default randomly distributed on a unit hypersphere, allowing the ensemble to act as a reservoir. A Nengo probe allows for a readout of neuron activity, where the addition of a solver can compute a weight matrix for a decoder. One builtin solver is the `LstsqL1s`, an elastic net which can be configured to act as LASSO. A Nengo network object contains the previously stated objects, and can be simulated for an amount of time using its run method. Nengo supports the automated mapping and deployment of network objects to several architectures with the import of a proper configuration script.

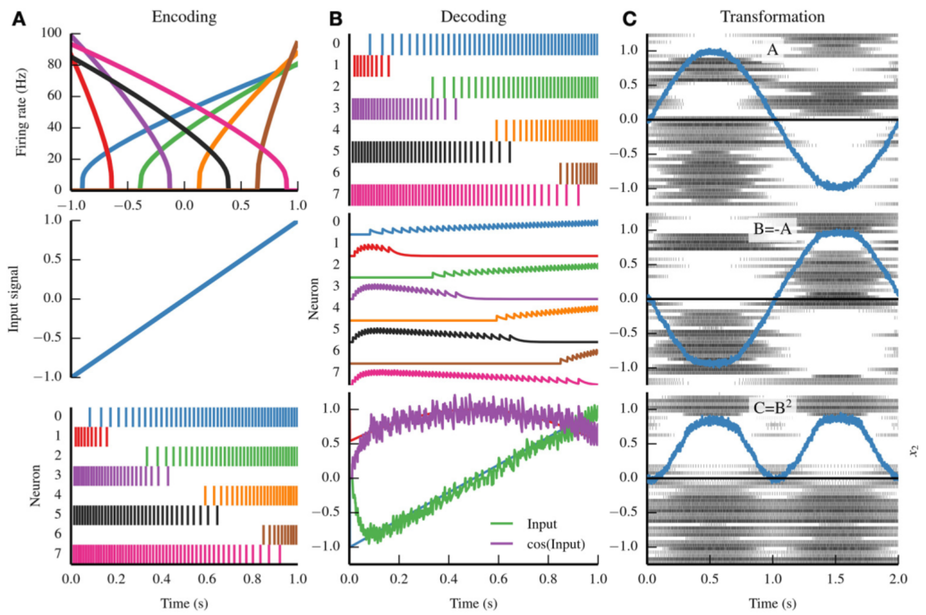


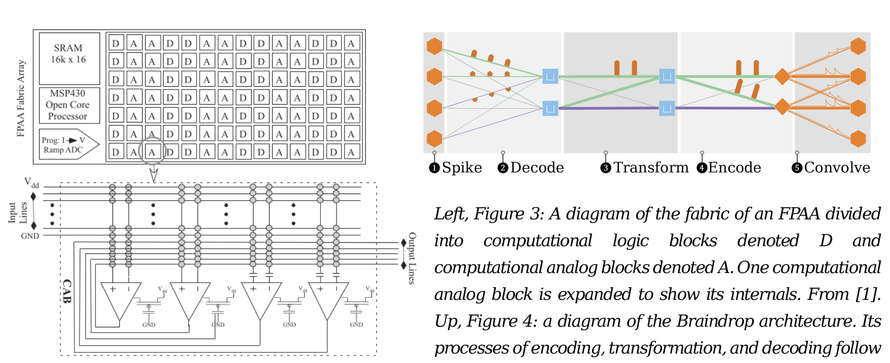
Figure 2: A) Nengo assigns tuning curves at random to neurons, allowing real-values to be encoded as spike trains upon input. B) By using decoders, Nengo can convert spike trains to real-values. C) Transformations of spike trains can occur using ensembles to fit models to non-linear systems. From [4].

Architectures

Two unconventional architectures are studied: the field programmable analog array (FPAA) and Braindrop.

FPAA: Whereas field programmable gate arrays (FPGAs) are reconfigurable hardware which configure computational logic blocks, FPAAs are reconfigurable hardware which configure a mixture of computational logic blocks and computational analog blocks. Computational analog blocks contain a C4 bandpass filter, amplitude detector/filter, and a first order low pass filter [1]. In FPAAs, floating-gate transistors are activated across routing crossbars during synthesis to perform computation during routing.

Braindrop: In contrast to FPAAs, which need to be configured on a hardware description level, the Braindrop architecture is designed for high level programming. Additionally, Braindrop decodes using accumulative thinning, a method to sparsify digital communication, hence reducing power consumption. Neurons are directly implemented in hardware and are supported by several auxiliary devices, including accumulators and buffers [2].



Left, Figure 3: A diagram of the fabric of an FPAA divided into computational logic blocks denoted D and computational analog blocks denoted A. One computational analog block is expanded to show its internals. From [1]. Up, Figure 4: a diagram of the Braindrop architecture. Its processes of encoding, transformation, and decoding follow the Nengo library. From [2].

Methods

For the testing of the model, two datasets were chosen, the Boston Housing Dataset (BHD) as a baseline and the Spiking Heidelberg Dataset (SHD) for a larger dataset, both in the size of each element and the overall number of elements, that is event-based. The BHD contains 14 real-valued attributes, one of which to be predicted, is median price. Alternatively, the SHD is composed of individual audio files from English and German speakers saying base-10 digits. By default, these audio files are pre-processed through an SNN to be represented as the times of activations for 700 different neurons and the activated neuron's id. The signals were transformed into a 2d-array A comprised of $a[i][j]$ where $a[i][j] = 1$ if the i-th neuron fired at time j / dt <= t < (j + 1) / dt else 0, and dt is a timesetp chosen to be 1e-2 seconds. Additionally, because A has a resulting size of 966000 elements, which causes a memory overflow, A is truncated and compressed to B (Fig. 5) comprised of $b[i][j]$ where

$$b[i][j] = \sum_{k=0}^{9} \sum_{l=0}^{9} a[i+k][j+l]$$

The model will begin with a node using the `PresentInput` process presenting each spike train as a flattened input vector. The model will then have an ensemble composed of 500 neurons and dimension equal to the node. The node and ensemble will be linked through a default connection. An additional node of dimension 10 will then be connected to the ensemble using a default `LstsqL1` solver, and the node will be probed. Finally, the model will be packaged in a `Simulator` object and simulated using the `run` method.

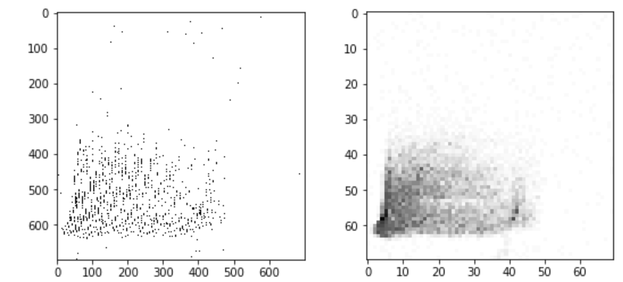


Figure 5: Left, the A array truncated. Right, the compressed B array. The vertical axis represents the y-th neuron; the x-axis is time.

Results

Unfortunately, neither novel architecture could be tested for power efficiency due to independent issues associated with both. For the FPAA, issues with the Reconfigurable Analog Signal Processor (RASP) toolchain prevented development of a hardware description designed for the FPAA. Since there is no FPAA backend for Nengo, development must occur using the RASP toolchain. The current version of the tools are built in an Ubuntu 12.04 virtualbox virtual machine, and as such the toolchain requires communication with an email server managed by the creator of the RASP toolchain to receive compiled bitstreams. Through the duration of the experiment, this email server was not online, so bitstreams could not be received, making the use of the toolchain impossible. Additionally, the built in neuron type for the architecture is a hodgkin-huxley neuron. During compilation of descriptions containing the hh_neuron component, the component that maps to a hodgkin-huxley neuron on board, the compiler throws errors, further making compilation impossible. For Braindrop, the server hosting the architecture went offline before development of the Nengo model could be finished, and it will likely not go back online until the development of Brainstorm, the second iteration of the architecture, is finished. The only data collected was the root mean squared error (RMSE) of each digit trained on a CPU (Fig. 6).

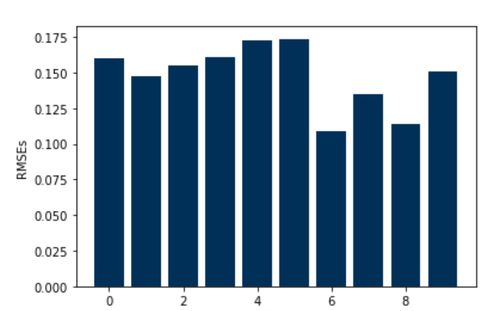


Figure 6: The root mean square error for every digit as passed through the model.

Discussion

From the data gathered, the model is shown to fit well with LASSO. This is evidenced by the root mean squared error values which were at a maximum 0.17 and at a minimum 0.11 as seen in Figure 6.

Since the experimenters did not have access to the servers associated with each novel architecture, and because the servers were down during development and testing, an evaluation of the model on these architectures was impossible. As such, no power usage data could be collected for these architectures. Because there is no data for comparison, the power usage data for the control, the CPU, has also been omitted from this study. Additionally, compilation errors on the RASP toolchain when using hh_neuron blocks prevents the usage of SNNs on the FPAA architecture.

Future Directions

To meet the goals outlined in this experiment, the experimenters require working access to mixed-signal architectures. For the FPAA, a RASP command line interface is under development; thus, no plans are currently in place to update the containerized RASP tools. And due to lack of time, communication with the administrators of the Braindrop server could not take place, nor would it be guaranteed that they would reinstate the server as they are occupied with manufacturing Brainstorm.

Since there is no FPAA backend for Nengo, two options exist to test LASSO models on FPAAs. Firstly, a Nengo FPAA backend could be created to allow automatic synthesis from Nengo models onto FPAAs. This is likely possible because a reconfigurable hardware backend for FPGAs already exists for Nengo. However, the creation of such a backend would require significant expertise with both analog hardware and hardware synthesis. Secondly, once the RASP command line interface exists, the model could be transcribed to the new method to mimic the structure of the Nengo network while synthesizing onto an FPAA. For Braindrop / Brainstorm, once a server is online, the Nengo model can be directly copied and run, only needing to reference the proper backend.

Additionally, because the usage of LASSO requires flattening the time domain of the input, the usage of a recurrent SNN might improve upon the accuracy of the network as it would maintain the time domain of its input. Legendre memory units show promise for this. Finally, additional datasets, such as NMIST, could be used with this network model as well.

Conclusion

The purpose of this experiment was to implement the LASSO into Nengo for the BHD and SHD datasets and to evaluate the power efficiency of the implementations on the FPAA and Braindrop mixed signal hardwares. SNNs, LASSO, and mixed-signal architectures all provide methods to decrease the power consumption of running neural networks, While the implementation of LASSO in Nengo was successful using a compression of the time domain of input vectors, expirementers lacked access to the resources necessary to test the power efficiency of the implementation. In the future, improvements on the FPAA RASP synthesis toolchain would be required to transfer the model onto its architecture, and Braindrop servers will need to be turned online to test the implementation on that architecture.

Works Cited

[1] S. George et al., "A Programmable and Configurable Mixed-Mode FPAA SoC," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 24, no. 6, pp. 2253–2261, Jun. 2016, doi: 10.1109/TVLSI.2015.2504119.
[2] A. Neckar et al., "Braindrop: A Mixed-Signal Neuromorphic Architecture With a Dynamical Systems-Based Programming Model," Proc. IEEE, vol. 107, no. 1, pp. 144–164, Jan. 2019, doi: 10.1109/JPROC.2018.2881432.
[3] "Lasso (statistics)," Wikipedia. Accessed: Oct. 28, 2021. [Online]. Available: https://en.wikipedia.org/wiki/Lasso_(statistics)
[4] T. Bekolay et al., "Nengo: a Python tool for building large-scale functional brain models," Front. Neuroinform., vol. 7, 2014, doi: 10.3389/fninf.2013.00048.

Aknowledgements

I would like to acknowledge Jeffrey Young at the Georgia Institute of Technology for his advisory and close tutelage in this research; Mourad Musellam for his collaboration and assistance in this research; Tom Conte at the Georgia Institute of Technology for directing me to Jeffery Young; Christopher Hogue at Choate Rosemary Hall for his teachings and advising; Harry Chen, Rhea Shea, Rohan Shivakumar, Praj Chirathivat, Anisa Murray, Dan Xiao, and Jessica Wu for their mutual support; Conor Brown for his constant support and kind words; and Claire Bowles, Dillon Bowles, Emma Catherine Bowles, and Carter Bowles for their loving support.