Spiking Neural Network on Gender Classification from Speech Data

Tai Duc Nguyen
Dept. Electrical and Computer Engineering
Drexel University
Philadelphia, PA, USA
tdn47@drexel.edu

December 9, 2019

Abstract

Spiking Neural Network, or SNN, provides a good trade-off between power consumption and accuracy, in comparison to a Convolutional Neural Network (CNN) of the same size. In a SNN, only a portion of the connected neurons remain active to fire spikes in a particular fashion, when the network is presented with a simulus at the input layer. This behavior allows the network to not only be very energy efficient, but also be capable of employing biologically plausible plasticity rules. In this paper, an unsupervised Spiking Neural Network model is simulated using C++ and applied to classify gender information in speech data. Our model is able to achieve an accuracy of 96.5%, while consuming 55 times less power and running at 3 times the speed, in comparison to a CNN of the same size.

A Introduction

It is no myths that Convolutional Neural Network and its derivates are computationally intensive [1]. Hence, no meaningful machine learning applications using CNNs can be run on small electronic devices in a reasonable amount of time. Often, data from customers' sensors are collected and sent to a powerful super-computer for CNN processing. However, this is not how human brains work – we do not amass gigabytes of information from every sensor in our body every millisecond and send it somewhere for processing. We would drive into each other before we can react to the things in front of our eyes. Therefore, in order to advance the state of art in the field of human-machine integration, it is necessary to introduce machine learning capabilities with good classification accuracy and security into ultra-low energy microelectronics. If this is achievable, it is possible for human and machine to integrate using Brain-Machine Interfaces (BMI), currently developed by researchers at Neuralink [2]. In this paper, Spiking Neural Networks will be shown that they can be the solution to the predicaments of information processing in machine-human integration.

B Background on Spiking Neural Network

B.1 Spiking Neural Network

Spiking Neural Networks are fundamentally different from any second generation's neural nets. Their neuronal models operate closely to those inside the human brain. In short, two neurons, the pre-synaptic neuron and the post-synaptic neuron, are connected by a synapse with a certain *weight number*. As the pre-synaptic neuron fires spikes towards the post-synaptic neuron, electrical charges accumulate at the latter until its potential threshold is breached. Once breached, the neuron sends spikes towards the next neuron it is connected to, and its membrane potential is reset [3]. Using this neuronal behavior, the network ”learns” using Spike-Timing Dependent Plasticity (STDP), which is a principle that partially explains the biological process of synapses' strength adjustments. STDP states that [4]:

- Long-term potentiation (LTP) occurs when the pre-synaptic neuron spikes immediately *before* the post-synaptic neuron spikes.
- Long-term depression (LTD) occures when the pre-synaptic neuron spikes immediately *after* the post-synaptic neuron spikes.

Hence, in our SNN neuron model, the *weight number*, which determines the likelihood of the pre-synaptic neuron to spike the next time around, increases with LTP and decreases with LTD. In a network trained with STDP, a certain group of neurons will fire in a certain pattern when a particular stimulus is introduced at the input layer.

B.2 Leaky Integrate-and-Fire Neurons

There are 3 common neuronal models (listed with increasing complexity): Leaky Integrate-and-Fire (LIF), Izhikevic 4-parameters, and Izhikevic 9-parameters. This paper will focus only on the LIF model, whose neurons' behavior is governed by the following equation [5]:

$$C\frac{dV}{dt} = -g_L(V(t) - V_R) + I(t) \tag{1}$$

A LIF neuron is a parallel combination of a ”leaky” resistor with conductance g_L and a capacitor with capacitance C. When $V(t)$ reaches a threshold level V_{th}, the capacitor discharges to a resting potential V_R. Since the biological neuron also ”undershoots” V_R at the discharging phase (so called Afterhyperpolarization - AHP), there exists a relative refractory period where the neuron cannot fire any new spikes if it is depolarized by smaller or equal potential. This behavior can be modeled with a dynamic potential threshold V_{th}: following each spike, increase V_{th} by a small amount [6]:

$$V_{th\_next} = V_{th\_prev}(1 + \tau t) \tag{2}$$

where t is the amount of time passed from last spike.

B.3 Poisson Spike Model

One problem with the LIF neuron model described above is that it is too deterministic in comparison to a neurons in the cortex. The spikes generated by our brains' neurons are highly randomized, which insinuates two possible theories:

- Theory 1: The irregular interspike interval (ISI) represents a random process, where the exact moments of the spikes do not hold meaningful information, but rather facilitate noise introduction into the system.
- Theory 2: The spike timing holds relevant information about a pre-synaptic event. Hence the ISI in conjunction with firing rate can represents a very large amount of information.

From these two theories, there are two main information encoding techniques: rate encoding and time encoding. Rate encoding is done by translating information into the number of spikes per unit time; and time encoding is accomplished by converting information into ISI. The SNN described in this paper leverages rate encoding together with a Poisson Spike model so as to preserve the property of ”randomness”.

The equation describing Poisson distribution is known as:

$$f(k;\lambda) = \frac{\lambda^k e^{-k}}{k!} \tag{3}$$

Hence, if the spike rate r is defined as the number of spikes s over an interval T, $r = s/T$, then the probability that n spikes ($n < s$) happen in a sub-interval Δt is:

$$P(n \in \Delta t) = C_s^n (\Delta t/T)^n (1 - \Delta t/T)^{s-n} \tag{4}$$

if $k \to \infty$, $T \to \infty$ and r stays constant:

$$P(n \in \Delta t) = \frac{(r\Delta t)^n e^{-r\Delta t}}{n!} \tag{5}$$

When Δt is small and $r\Delta t << 1$, $P(n = 1 \in \Delta t)$ can be approximated with $r\Delta t$. Hence, for each interval i, a number between 0 and 1 is generated from the Uniform distribution, $x[i]$, such that: if $x[i] \leq r\Delta t$, then a spike is initiated. The SNN described here uses this method to encode audio information into spike trains.

C Background on CARLsim4

As a collaborative effort between Drexel University and University of California, Irvine to advance in the domain of Spiking Neural Networks, the SNN simulator chosen for this paper is CARLsim4.

CARLsim4 is a GPU-accelerated SNN simulation tool written in C++ by researchers at the Cognitive Anteater Robotics Laboratory in University of California, Irvine. This software provides the basic underlying framework to create an application using a Spiking Neural Network. Its discovered and understood capabilities [1] include:

- Definitions of the Izhikevich 4-Parameters, Izhikevich 9-Parameters and LIF neuronal model
- Mechanisms to support building connections between groups of neurons of the same/different model
- Mechanisms to support simulating information encoding and spike train generation
- Mechanisms to support STDP and Homeostatis
- Two monitoring mechanisms for developing and post-processing
- GPU accelerated for very large networks

CARLsim4's workflow composes of 3 states (shown in figure below): Config, Setup, and Run. The Config state is where the topologies of the network, or, groups of neurons along with their connections to one another are defined. The Setup state allows the user to choose the back-end (CPU or GPU) of the network, and bring up monitors to look at a particular neuron/layer. The Run state is responsible for spike train generation and the executions of all neurons in the network. The state/output of the network can also be stored for post-processing or demoing at the end of the Run state.

In addition, two monitoring tools, available in CALRsim4, can write outputs to binary files, which can be used in the software's MATLAB Offline Analysis Toolbox (OAT). The outputs of the SNN in this paper will be piped into MATLAB for K-Means clustering.

D Background on DARPA's TIMIT Acoustic-Phonetic Continuous Speech Dataset

This dataset (will be refered later on as TIMIT) is created as a joint effort between: Massachusetts Institute of Technology (MIT), Stanford Research Institute (SRI), and Texas Instruments (TI), under the funding of the Defense Advanced Research Projects Agency - Information Science and Technology Office (DARPA-ISTO) [7].

TIMIT contains the transcripts and CD quality audio data of 6300 sentences, in which the same 10 sentences is recorded by each of the 630 speakers from 8 major dialects in the United States of America. The dialects' regions are as follows:

- DR1: New England
- DR2: Northern
- DR3: North Midland
- DR4: South Midland
- DR5: Southern

[1] Other capabilities are available in the software but will not be focused.

- DR6: New York City
- DR7: Western
- DR8: Army Brat (moved around)

The gender distribution for each region is listed in Table 1 below:

Dialect Region (DR)	# Male	# Female	Total
1	31 (63%)	18 (27%)	49 (8%)
2	71 (70%)	31 (30%)	102 (16%)
3	79 (67%)	23 (23%)	102 (16%)
4	69 (69%)	31 (31%)	100 (16%)
5	62 (63%)	36 (37%)	98 (16%)
6	30 (65%)	16 (35%)	46 (7%)
7	74 (74%)	26 (26%)	100 (16%)
8	22 (67%)	11 (33%)	33 (5%)
ALL	438 (70%)	192 (30%)	630 (100%)

Table 1: Table detailing the gender distribution for each dialect region in TIMIT

From the table above, it is clear that TIMIT is skewed because the male population is 2.3 times the female population. Hence, for the purpose of balancing, speech data from 192 randomly selected males (out of 438) will be chosen to form the database along with those from 192 females. Therefore, the total amount of audio clips to be processed is 3840.

E Experimental Setup

E.1 Data Pre-processing Using MFCCs

The audio data available is in the ”wav” file format – a list of millions of 32-bit floating point numbers. There are no correlations between these numbers and the speaker’s gender. Hence, it is necessary to transform the data into a different, more concise and more meaningful representation. The most famous transformation is the Short-Term-Fourier-Transform (STFT), which produces a spectrogram – a 2D matrix that entails the energy at a certain frequency band in a particular time period.

Due to the fact that the dataset is composed of people with diverse dialects speaking distinct sentences, the STFT spectrogram will look tremendously different from person to person and from sentences to sentences. Therefore, in order to further concise the STFT spectrogram to eliminate these differences, a spectrogram-of-a-spectrogram, or, a cepstral is constructed for each audio signal. In addition, the filter banks (frequency bands) used in the first spectrogram is derived from the nonlinear Mel scale [8] so that the resulting spectra reflect human hearing better. This combination is called the Mel-frequency cepstrum (MFC). And the coefficients of all MFCs are called the Mel-frequency cepstral coefficients (MFCC).

As audio signals in the database have different lengths, in order to make all Mel spectrograms congruent in shape for the neural network, different window lengths [2] (in number of samples) are calculated for each signal so that the number of windows for the entire signal is always 99 $(50 \times 2 - 1)$.

In addition to performing MFCC analysis on each audio signal, silent and noise removal is also performed before doing MFCC to further increase the ratio of relevant to irrelevant information in the output signal.

E.2 SNN Architecture

The pre-processed audio signals are fed into the SNN one by one. Since the size of the MFCC is 13 by 99, the input layer will 1287 neurons – one neuron for every number. In the second layer, every neuron is connected to 3 neurons in the previous layer in a 1 by 3 kernel with a stride of 1 in the x direction and 3 in the y direction. These non-overlapping connections are meant to discover patterns in each of the small region of the MFCC. The third layer is connected to the second layer the same way the second layer is connected to the input layer. However, the type of neurons in this layer is ***max-pooling*** instead of LIF.

A max-pooling neuron

F Simulation Results

G Future Works

References

[1]

[2]

[3]

[4]

[5]

[6] D C Somers, S B Nelson, and M Sur. An emergent model of orientation selectivity in cat visualcortical simple cells. Journal of Neuroscience, 15:5448–5465, 1995.

[7]

[8]

[9]

[2] with overlap length equals to half of window length

[10]