Spiking Neurons for Filtering Noisy Speech

Improving speech classification performance in the presence of noise using gating neurons

Nivedya S Nambiar, 190070039
Guided by
Prof. Udayan Ganguly
Anmol Biswas

Report for Bachelor's Thesis Project - 2
Spring 2023

Department of Electrical Engineering
Indian Institute of Technology Bombay

Contents

1 Acknowledgements

I would like to thank my project guides, Anmol Biswas and Prof. Udayan Ganguly for their suggestions and insights throughout the course of the project, enriching the learning experience, and for providing constant support without which the project could not have been completed. I express my gratitude to Abhishek Kadam who provided me with the data that was crucial for this project. I would also like to thank the MeLoDe Algorithms team whose inputs at weekly interactions enabled me to revise my perspective, while giving me a platform to share my ideas, gain feedback, and clarify doubts. I would also like to thank the Department of Electrical Engineering for granting me the opportunity to work on this project and for providing me with access to resources that were imperative in completing this project.

2 Introduction

The presence of additive noise impedes the performance of all kinds of speech classifiers. For liquid state machines, the additive noise could manifest as unwanted spikes in the input layer at any instant, or even as distortions of the spiking pattern of speech. Through this project, I have attempted to develop a mechanism by which the input layer spikes are gated in time to pass the spikes only at instants where speech is detected. The overarching assumption here is that simpler kinds of noise - with uniform and predictable waveforms - do not create significant distortion in speech, making them easier to gate off.
The basic objective is filtering away the noise, which ideally involves retaining those spikes in the input that would have been present had only the original speech signal been passed, and ignoring the spikes resulting from noise. A primary step toward this is gating in time, i.e., detecting the presence of speech at a time instant, and then have the liquid "listen" only at these instants. This approach essentially is equivalent to voice activity detection in the presence of noise.
The performance of the liquid can be tested after validating that this approach "works". In this project, the gating was verified through visual inspection of the waveforms, as described in the methods section that follows.

3 Brief look at spike frequency adaptation

Spike frequency adaptation (SFA) is the control of firing rate of a neuron presented with a constant intensity stimulus. It has been found in biological systems, and can be implemented in spiking neural networks, as has been attempted in this project. SFA can thought of as a transient increase in the firing threshold of the neuron, or an increase in its refractory period. The definition of SFA makes it a promising candidate to be considered for noise elimination from speech data, as the neuron could potentially stop firing in the presence of a

constant stimulus noise it is presented with, and spike when a signal of interest that is of a higher amplitude is presented.
As described in [6], frequency adaptation is interpreted as given by the following equations:

$$\frac{dV_m}{dt} = -g * a_K * V_m - g_K * a_K * (V_m - V_K) + u$$

$$\frac{dg_K}{dt} = -\frac{g_K}{\tau_K} + dg * \Sigma_l \delta(t - t_l)$$

where V_m represents the neuron voltage, u represents the input current stimulus to the neuron, and g,a_K,V_K,dg and τ_K are constants. g_K is the parameter responsible for SFA, the value of which increases as the neuron spikes (as indicated by times t_l within each interval of updating g_K). This increase in g_K slows down the increase in voltage V_m of the neuron.

4 Methods

4.1 Initial attempts at noisy speech classification

At first, I attempted to classify noisy speech directly using a liquid state machine with spike frequency adaptation. I used samples of words from the TI-46 dataset mixed with the noise of travelling in an autorickshaw. The motivation behind using SFA here is that with frequency adaptation, neurons will cease to fire in the presence of noise that is relatively uniform and of low intensity but restart the spiking once speech signal that has higher intensity of stimulus is presented. SFA was implemented in neurons within a 10x10x10 liquid. The input waveform was converted to cochleagram using Lyon's ear model[4] and further to spikes using BSA encoding[3], as implemented in the Python package pyspikes[1]. For this section of the project, this processing into spikes had been done using custom functions in MATLAB. The spikes are fed to the liquid through an input layer with as many channels as the cochleagram produced using Lyon's ear model. The liquid was initialised with probabilistic connections between neurons in the liquid, and random connections with neurons in the input layer. The probability of two neurons in the liquid being connected depended on the distance between these in the liquid. The liquid activation pattern averaged over time was used to classify the input by a supervised linear classifier.
However, only a maximum accuracy of 23% was obtained here. On further inspecting the samples, it was found that there was considerable distortion in the signal after addition of noise due to low signal-to-noise ratio (SNR), and the implementation. Hence, the approach was revamped to build the model from ground up, and also to work with a simpler dataset.

4.2 Dataset used

The 10 spoken commands in the TI46 dataset were chosen and they were mixed with a segment of noise taken from a recording of indoor home noise. The

segment was chosen by inspecting random clips of noise files for uniform and predictable noise sounds, both visually through amplitude-time waveform and by hearing. The waveform of the segment of noise chosen is given below. This noise segment is mixed with the audio files of spoken words at an SNR=20dB, according to the method given in [5].

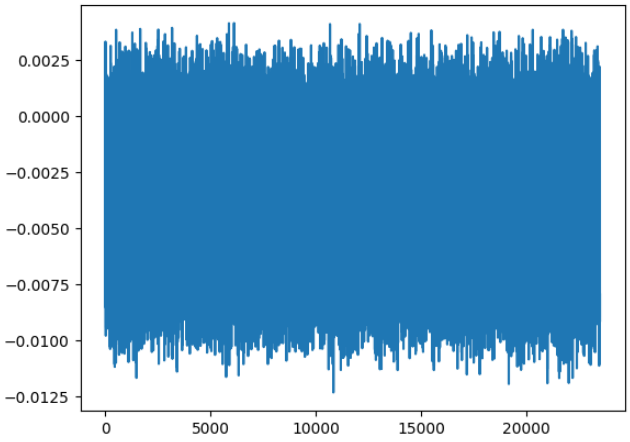

Figure 1: Amplitude vs time waveform of chosen noise segment

The word "go" spoken by speaker f1 is mixed with the noise segment, and the resulting waveform, cochleagram and spike pattern are shown in the figure below. The use of log-scale short-time Fourier transform was also explored as an alternative to the cochleagram. However, this was not be employed in this project as the grouping into channels could not be implemented.

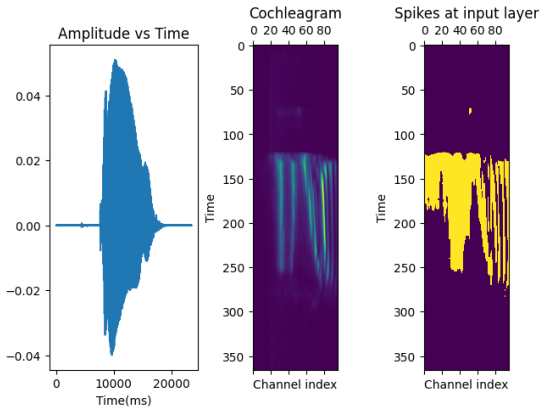

Figure 2: "go" spoken by f1, without noise

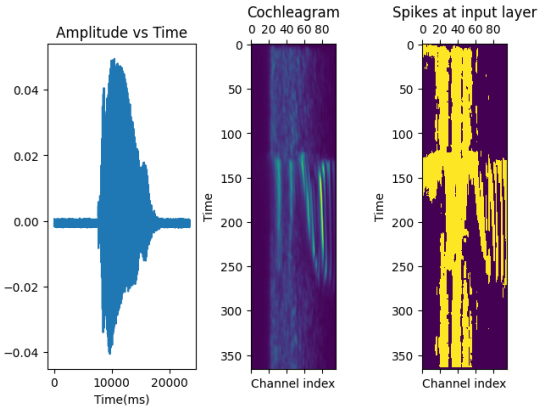

Figure 3: "go" spoken by f1, mixed with noise at SNR=20dB

4.3 Gating - Using overall energy

The initial experiments with observing the speech structure of different words with and without noise indicate that there is a certain "golden window" in time when spikes for speech signal are present. In this window, channels across the spectrum are activated for speech, showing the broadband nature of speech signals. If the reservoir were to somehow be activated exclusively during this window to let in spikes from the input neurons, we could have gating in time. This window for speech could be detected by something as simple as the collective spiking of all (nearly all) channels at once - i.e., an approximate "AND"" of all channels, or the total energy of the input layer. The following design was implemented.

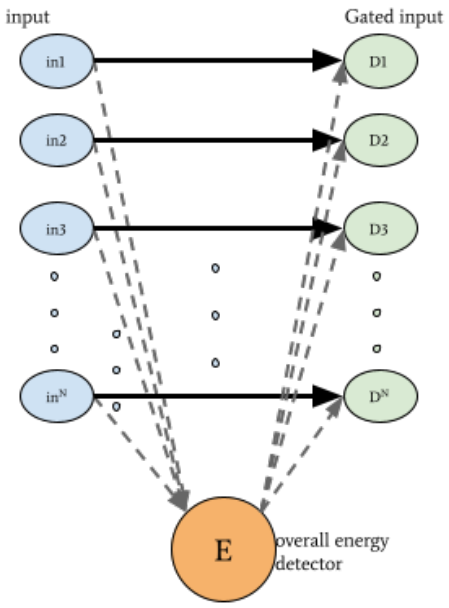

Figure 4: Initial gating scheme developed

A single neuron E is used to spike for the duration of the relevant speech information. This neuron E receives equally weighted input from all channels, and spikes only if the input stimulus is greater than a threshold $th_m = 10$(chosen after trial and error). This neuron has no 'memory', i.e., any current or voltage built up at an instant is leaked away before the next set of inputs arrives after interval dt. This was tested for pure speech, and speech with noise. The idea is that the neuron E spikes only during the time window for speech, irrespective of whether noise is present or not.

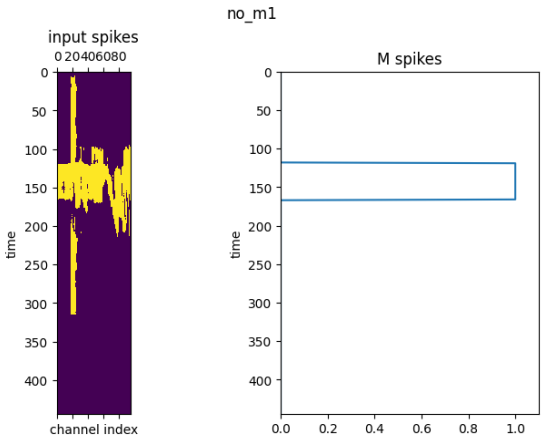

Figure 5: “no” spoken by speaker M1 (without noise): input spike pattern (left) and spike in E

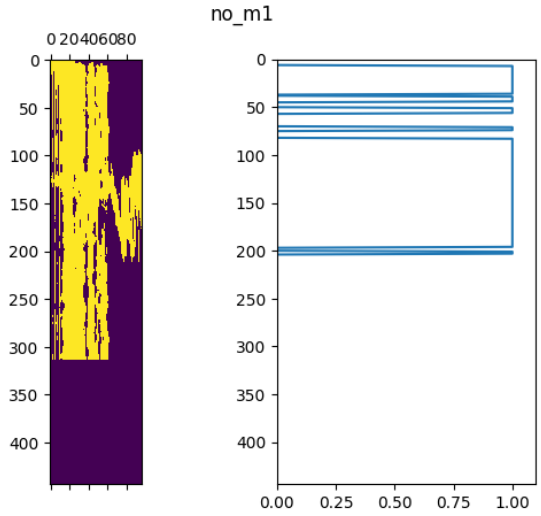

Figure 6: “no” spoken by speaker M1 (with noise): input spike pattern (left) and spike in E

Next a dummy layer (D1, D2, .. Dn) with as many neurons as the input layer is created to select the input spikes in the ”golden window” in time, by using neuron E. E activates the neurons in the dummy layer by feeding into them through synapses with weights higher than that from the input layer. The idea is that neurons in the dummy layer will now replicate the information in the input only for the duration of the useful speech signal detected by E, thus

acting as the new gated input. For this, the synapses into the dummy layer are again chosen not to have any memory, i.e., no integration of past inputs. The results are given below.

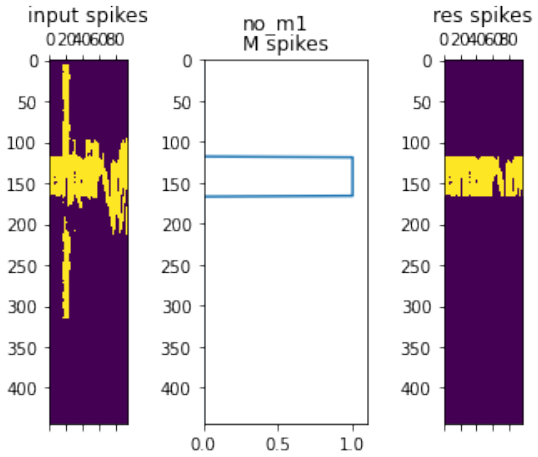

Figure 7: "no" spoken by speaker M1 (without noise): input spike pattern (extreme left) and gated input (extreme right)

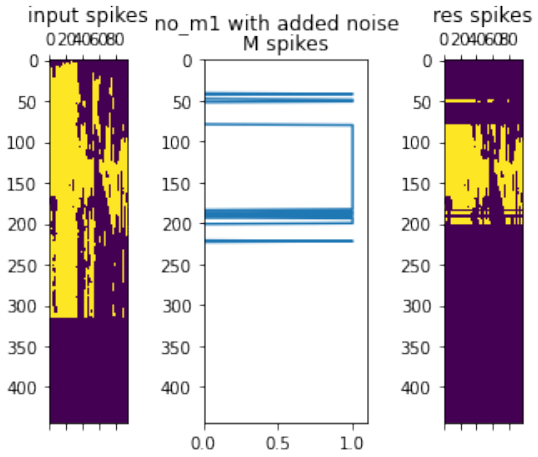

Figure 8: "no" spoken by speaker M1 (with noise): input spike pattern (extreme left) and gated input (extreme right)

On visually inspecting the gating for different words and different conditions, it was found that there was a lot of clipping of speech signal. Hence the thresholds and weights were adjusted to allow signal to pass through. However, this created

many false positives in speech detection, hence SFA was incorporated to reduce the spiking due to noise.

4.3.1 Spike Frequency Adaptation with Gating

Spike frequency adaptation was hence incorporated into neuron E. The voltage of E at any instant is this decided by a factor g_K that varies with the spiking activity in E. Spiking in E causes g_K to rise, which lowers the voltage of E, slowing down the firing rate.

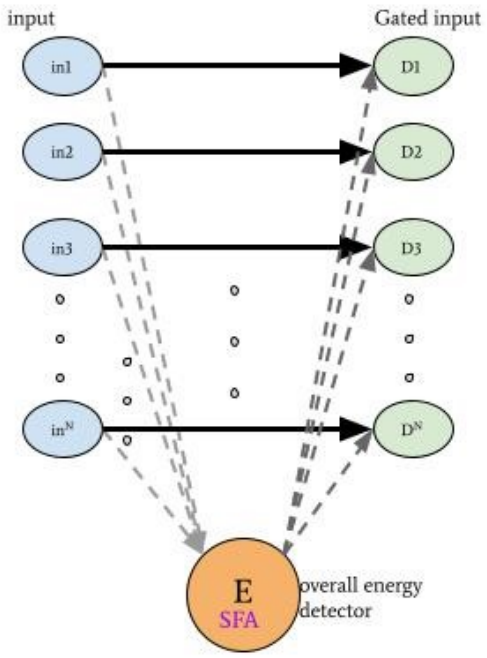

Figure 9: SFA incorporated in E

The resulting graphs of spiking patterns are shown below. SFA is shown here to be only partially effective for reducing the spikes from neuron E at low spiking thresholds.

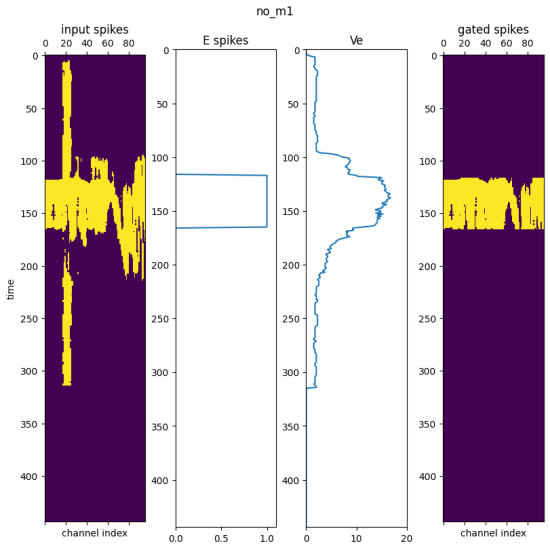

Figure 10: Gated output after adding SFA for a clean sample of “no” spoken by m1

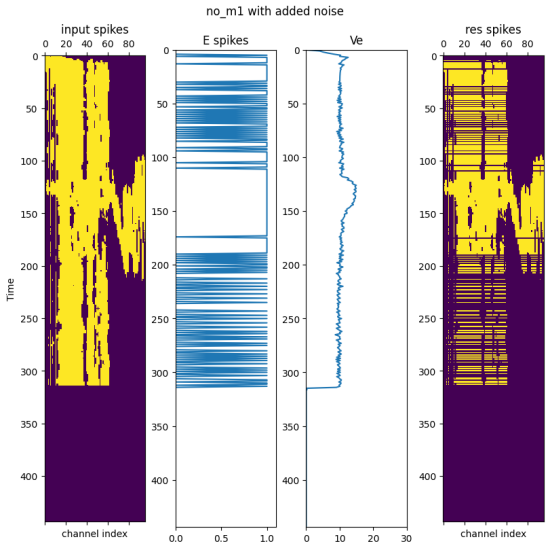

Figure 11: Gated output after adding SFA for a noisy sample of “no” spoken by m1

4.4 Gating - Adding voice bar detection

After a more in-depth inspection of the spike patterns of speech vs noise, and a study of the properties of speech, this next model was developed. Sounds of speech can be voiced or unvoiced. Voiced sounds have activation of the lower frequency bands of speech, while unvoiced sounds are characterised by turbulent airflow in the vocal tract that shows up as high energy[2]. The motivation of the voice bar detector is to capture sounds such as vowels that are voiced, i.e., have a voice bar, but without enough overall energy to be picked up the energy detector E. The energy detector is still functional in parallel to pick up sounds such as unvoiced ones where there is very high energy overall. The presence of the voice bar at an instant is assumed here to be a sufficient condition for detecting speech. The proposed design is as follows. The first 10 channels are designated as being responsible for the voice bar.

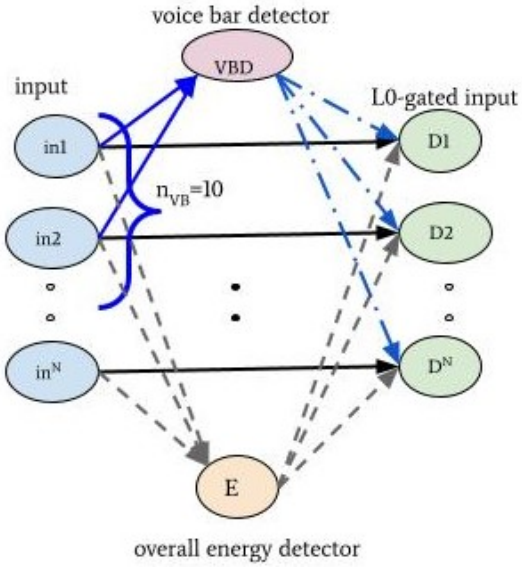

Figure 12: Gating scheme with voice bar and energy detector

The resulting spike patterns are shown in the following figures. In addition, a condition that at least 40 out of 96 channels should be activated for the neuron E to fire was placed. The thresholds and weights were adjusted here for an optimal result.

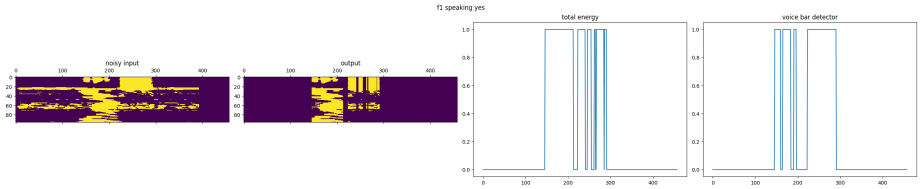

Figure 13: Result of gating applied on "yes" spoken by f1 mixed with noise. First two graphs show the spiking pattern before gating and after gating respectively, while the third figure is the firing in neuron E. The last figure depicts the firing in the voice bar detector.

4.4.1 Adding Frequency Adaptation

To restrict the firing of the energy detector at high levels of noise at an instant, SFA was incorporated into the neuron E. The design is as follows.

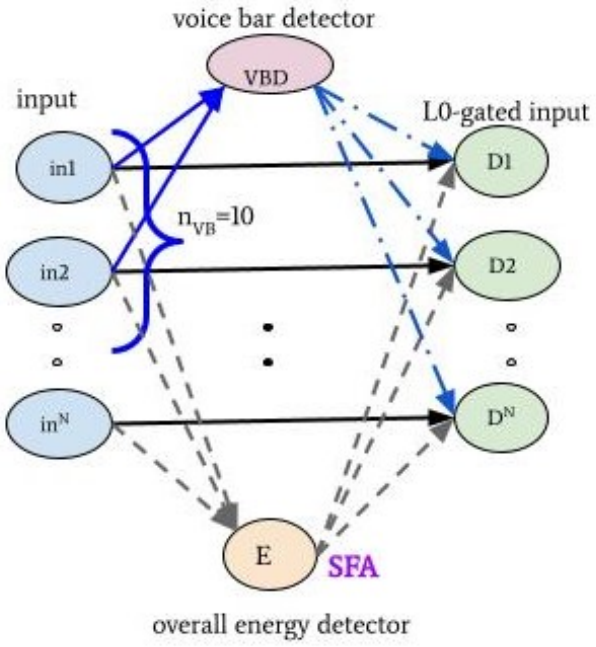

Figure 14: Gating scheme with voice bar and energy detector having SFA

The results of spiking patterns are shown below.

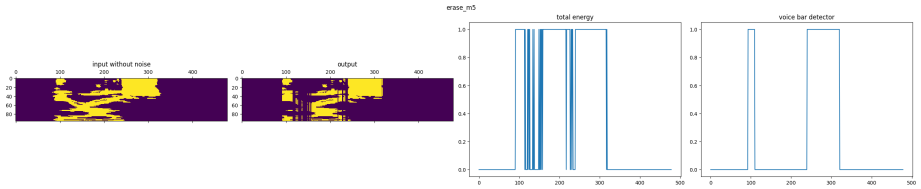

Figure 15: “erase” spoken by m5 without noise

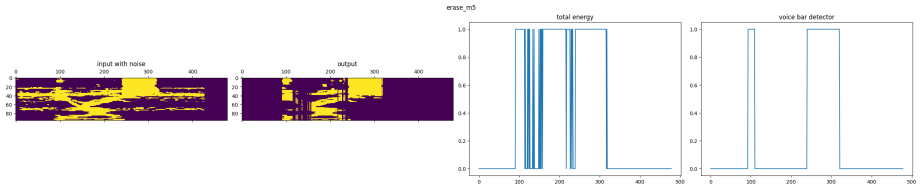

Figure 16: “erase” spoken by m5 with noise

5 Results

In order to formally evaluate the effectiveness of this gating scheme, the gated output was first directly fit and tested against the labels using a linear SGD classifier, i.e., without an LSM for preprocessing. To this end, the features across time and channel indices were flattened to pass to the classifier. Following this, the clean data was used for training and the gated output was used for testing. The same procedure was repeated for noisy data before gating to confirm the effectiveness of gating.

Table 1: Test scores for different data types - noisy, with and without gating

Data Type	trained on same type	trained on clean data
Noisy data without gating	0.924	0.574
Noisy data after gating	0.880	0.701

5.1 Adding a reservoir for further processing

Next, a 10x10x10 reservoir is added to which the gated input is fed. This reservoir propagates the gated spikes in time to produce a spike pattern which can be used to train an SGD classifier. The reservoir has random excitatory and inhibitory connections to the gated input layer “L0” and distance-based probabilistic connections between reservoir neurons. The result of training is given in the following table.

Table 2: Test scores after adding reservoir and training on clean data, tested on - noisy, with and without gating

Data Type	Train score	Test score
Noisy data without gating	0.969	0.425
Noisy data after gating	0.969	0.604

The resulting confusion matrices for both cases are shown below.

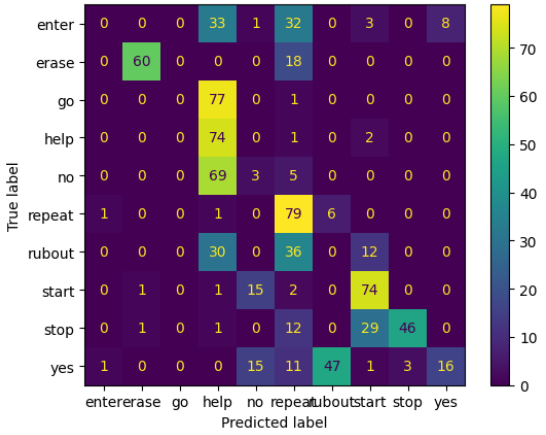

Figure 17: Confusion matrix after testing on noisy input

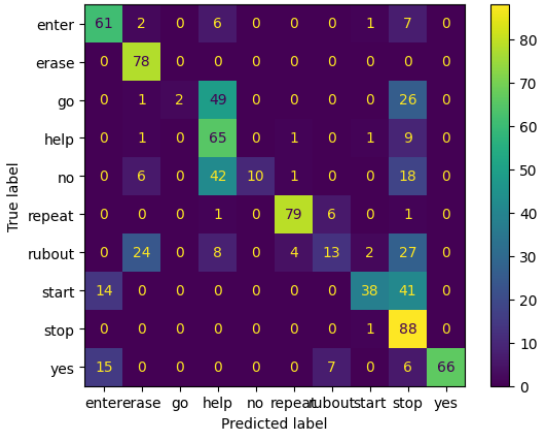

Figure 18: Confusion matrix after testing on gated input

6 Conclusion

In this project, I have attempted to develop a spiking neuron model that can gate away the noise by detecting the presence of speech. This scheme does provide improvement in terms of classification accuracy after training on clean data. The model utilises two kinds of neurons - an overall energy detector and a voice bar detector. The design is based on the known properties of speech vs noise.

7 Limitations

The major limitation is that the current scheme only performs gating in time and does not filter out the noise entirely as desired in the ideal case. There is also a lot of clipping of the signal even when there is speech, especially when the firing thresholds are higher. Hence, there is a trade-off between this type of error and the inclusion of noisy spikes in the data. In addition, to optimise the performance one requires a lot of parameter tuning. The model also needs to be improved to work for lower SNRs and different kinds of noise.

8 Future Work

To include filtering instead of only gating away the noise, the spikes could be examined and eliminated based on a more sophisticated scheme. For example the correlation between channels' activities could be taken into account. In speech signals, and voiced sounds in particular, the spikes occur in clusters. i.e., there are certain channels consecutive to one another that are activated at once. This depends on the "filter" defined by the vocal tract for that sound, what its formant frequencies are, etc. For this, the channels in the gated layer "L0" could be activated depending on whether nearby channels are activated
There could also be learning scheme, where the neurons learn to take up their roles as filters - for example a neuron that learns to compute the total energy or the energy from the voice bar for detecting speech. The weights between the gating neurons could also be learnt in this way.
Another avenue to explore is the use of log-scale short-time Fourier transform as opposed to cochleagram. This would ensure that the speech is not distorted by noise by gain controllers as in Lyon ear model.

9 Code links

Code showing the functions and graphs for final gating scheme developed - `https://colab.research.google.com/drive/1QMjeOlbZQdsyJLbVpK8Z9s5YyUCLHTJv?usp=sharing`

References

[1] Akshay Raj Gollahalli. Spike Encoders. https://github.com/akshaybabloo/Spikes, 2020.

[2] Rob Hagiwara. How do I read a spectrogram? https://home.cc.umanitoba.ca/~robh/howto.html#intro, 2009.

[3] B. Schrauwen and J. Van Campenhout. Bsa, a fast and accurate spike train encoding scheme. In *Proceedings of the International Joint Conference on Neural Networks, 2003.*, volume 4, pages 2825–2830 vol.4, 2003.

[4] Sciforce. Our Adaptation of Lyon's Auditory Model for Python. https://medium.com/sciforce/our-adaptation-of-lyons-auditory-model-for-python-4f41adf55d4e, 2019.

[5] StackOverflow. Mix second audio clip at specific SNR to original audio file in Python. https://stackoverflow.com/questions/71915018/mix-second-audio-clip-at-specific-snr-to-original-audio-file-in-python, 2022.

[6] Alessandro Treves. Mean-field analysis of neuronal spike dynamics. *Network: Computation in Neural Systems*, 4(3):259–284, 1993.