Classification of a Two-Class ECG Dataset Based on Perceptron Learning in a Cortical Pyramidal Neuron Model

İlknur Kayıkçıoğlu Bozkır (ilknurkayikcioglu@ktu.edu.tr)
Karadeniz Technical University

Zübeyir Özcan
Karadeniz Technical University

Cemal Köse
Karadeniz Technical University

Temel Kayıkçıoğlu
Karadeniz Technical University

Research Article

Keywords: ECG Data, Pyramidal Neurons, Machine learning, Synaptic Inputs, Neural Dynamics, Biological Neural Networks

Posted Date: October 14th, 2022

DOI: https://doi.org/10.21203/rs.3.rs-2149891/v1

Classification of a Two-Class ECG Dataset Based on Perceptron Learning in a Cortical Pyramidal Neuron Model

İlknur Kayıkçıoğlu Bozkır[1,3*], Zübeyir Özcan[2], Cemal Köse[3] and Temel Kayıkçıoğlu[2]

[1*]Department of Computer Engineering, Karadeniz Technical University, Trabzon, Turkiye.
[2]Department of Electrical and Electronics Engineering, Karadeniz Technical University, Trabzon, Turkiye.
[3]Department of Computer Engineering, Bülent Ecevit University, Zonguldak, Turkiye.

*Corresponding author(s). E-mail(s): ilknurkayikcioglu@ktu.edu.tr;
Contributing authors: zubeyirozcan@ktu.edu.tr; ckose@ktu.edu.tr; tkayikci@ktu.edu.tr;

Abstract

Pyramidal cells are the most prevalent neuronal type in the cortex, receiving thousands of synaptic inputs from all over the brain, and sending the largest axon outputs. They have a variety of active conductivities and complex morphologies that support highly nonlinear dendritic calculations. There has been a growing interest in understanding the classification abilities of pyramidal neurons. The perceptron learning algorithm, one of the foundations of machine learning, uses the highly simplified mathematical abstraction of a neuron, and it is unclear to what extent real biophysical neurons can perform perceptron like learning. In this article, we investigated the performance of a pyramidal neuron model in the classification problem of a two-class ECG dataset for different synaptic regions by using the perceptron learning method. The main purpose of this study is to reveal what role the soma, basilar and apical dendrites play in a classification problem. We concluded that when the synaptic receptor locations are selected close to the soma, classification performance close to

the single layer perceptron can be obtained. The results indicated that the pyramidal neuron can successfully classify real-world data.

Keywords: ECG Data, Pyramidal Neurons, Machine learning, Synaptic Inputs, Neural Dynamics, Biological Neural Networks

1 Introduction

Machine learning is the study of computer algorithms that focuses on analyzing and interpreting patterns and structures in data. Though traditional algorithms have solved numerous problems in various areas such as health (Doupe et al, 2019), security (Nassif et al, 2021), image processing (Zerouaoui and Idri, 2021) and signal processing (Hosseini et al, 2020), with the increase in the amount of data received from automated systems today, it has become more difficult for them to process huge datasets (Janiesch et al, 2021). Their compact structures have made them easy-to-implement and practical but as the time passes their incapability of processing huge datasets emerged. It is not fair to say that the era of the traditional machine learning algorithms has passed but their usage in such areas have been bleeding day by day.

The structure of the brain has been a great inspiration to the scientists studying on machine learning (Alzubaidi et al, 2021). Afterall, the concept of learning has emerged from the capabilities of the brain. Therefore, the brain, especially the human brain, has been a role model of machine learning algorithms. Though the traditional algorithms have very similarities with brain regarding the structure, they are not a typical replica of it. Therefore, the structure of the brain inspired many to research itself for the sake of machine learning (Mahesh, 2020).

In the first quarter of twentieth century, the structure of brain was well-known in anatomical aspect. But the molecular and electrical structure was incomparable (DeFelipe, 2011). When the first neuron model had been proposed, maybe nobody would know that a first step in artificial intelligence was taken. Accordingly, the first neuron model, which we call it Integrate & Fire (IF) neuron model now, suggests that a neuron receives various stimulus, integrates them and fires if the desired conditions were met (Lapique, 1907). That was the first ever research modeling the neuron for understanding the brain (Burkitt, 2006). Further research was made on models connecting neurons to each other to build networks but the concept of learning in brain was still a mystery.

Biological neuron processes information via spike trains in which the information was encoded in the spike rate and the precise timings of the spikes, but the real-world data was generally analogous (Gerstner and Kistler, 2002). McCulloch and Pitts (1943) (MP) decided to focus on a new biological neuron abstraction which assumes signals as spike rates. Their neuron model assumes that neurons are basically simple units that take weighted sum of their inputs

and spikes if the weighted sum is greater than a predefined value. This was the first ever neuron model to build networks. But the weight values had to be defined for learning purposes. Rosenblatt (1957) suggested the first ever learning algorithm to define the appropriate weights. The algorithm put accounts the difference between the output and the desired values to reach the solution step by step which is known the first generational neural networks today. But their incapability of solving XOR problem limited their usage in the area. Then, the second generational neural networks emerged. First generational networks were consisting of only two layers which are input and output layers. But the XOR problem required more layers, which are known as hidden layers, to be solved. With slight changes on the learning algorithm and on the neuron model the networks were able to be trained for hidden layers. Thus, the first sparkle for the deep learning era has flashed (Rumelhart et al, 1986).

Second generational neural networks made it possible to analyze large and complex data sets. The first ease of these networks was that they can be built according to any design. This let them to be designed for any given problem specifically. Contrary to popular belief back then, more number of neuron caused more problem since the networks tend to overfit as the number of neuron increased (Prieto et al, 2016). Therefore, the architectures inspired by the brain has been shown up. This unique approach was named as deep learning. Today, it has conquered many fields in data analysis and machine learning (LeCun et al, 2015). However, deep learning has some limitations. Even though, it is inspired by the human brain, it is still far from the capabilities of the human brain. Furthermore, deep learning architectures include millions of parameters in complex applications. Therefore, learning algorithms spend hours on training and also encounter important issues; such as appropriate architecture, learning parameters, stop criteria and local minimums (Shrestha and Mahmood, 2019).

For last decades, MP neuron model was under the spotlight for building neural networks for machine learning purposes. On the other hand, neural modelling was still an active field of research. More and more neuron models have been shown up mostly for computational neuroscience field to simulate and understand the nature of biological neural networks. To date, many studies have been carried out in the field of neural modeling using neuron models such as Hodgkin–Huxley (HH) (Hodgkin and Huxley, 1952), IF (Lapique, 1907), and Izhikevich (Izhikevich, 2007). The IF model is a valuable abstraction of a biological neuron for using in networks but lacks representing the intracellular dynamics. On the other hand, HH model successfully represents the ionic channels and molecular structure of the biological neurons. Considering their value in computational neuroscience, these models are commonly used models in the literature (Hay et al, 2011). In addition, the usage of neurons in machine learning and the disadvantages of the algorithms took attention of researchers from neuroscience. The idea of that the brain is the ultimate intelligence organ, lead researchers to give more effort to the networks which is build up from

biologically realistic neuron model. This effort resulted in Spiking Neural Networks (SNN) which are known as third generational neural network (Bose et al, 2016; Wang et al, 2020). SNN has opened new horizons in the field of machine learning. These structures receive action potentials from sensors, neurons, or external networks as postsynaptic potentials via synapses (Ponulak and Kasinski, 2011). The best part of them is that they require less neurons for the given tasks comparing to the networks with MP models, since they constitute from non-linear dynamics. Also, MP models represent data as spike rates, but biological model represent data as pure spikes. This makes SNNs energy efficient comparing to Artificial Neural Networks (ANN) (Farsa et al, 2019). But they are still not fully realistic comparing to the networks of the brain. The biologically realistic neurons mostly represented as simple units called point neurons. The biological neuron is more complex and detailed than the models used in SNNs. At first look, It can be easily seen that brain neurons have a central processing unit called soma and branches which are both responsible for processing and propagating spikes. Neglecting the branches both makes the model not realistic enough and causes inability to take advantage of the benefits they would bring (Gerstner et al, 2014). Therefore, networks with biologically realistic neurons with branches open new horizons to the field (London and Häusser, 2005).

Pyramidal neurons have a complex dendritic tree structure that receives and processes dense synaptic input. They receive synaptic input in the soma, axon, and dendrites. The dendritic structure consists of three different tree topologies: tuft, basilar, and apical. This indicates that pyramidal neurons carry different functions depending on the locations where the synaptic inputs are located (Spruston, 2008). The integration of excitatory and inhibitory inputs into pyramidal neurons is a complex process that depends on the locations of the inputs, the weights and timings of synaptic inputs, the diversity of ion channels, as well as the spatial relationship between activated synapses and the final integration site in the axon. The functional impact of input locations has been extensively studied area but not fully understood due to this complex process (Polsky et al, 2004). In this area, Poirazi et al (2003) revealed that a hippocampal CA1 pyramidal neuron can be modeled by a two-layer neural network. In this network, the nonlinear properties of the dendrites are represented by choosing the hidden layer sigmoidal. The sum of the dendritic currents in the soma is modeled using the output neuron. However in this study, the computational advantage of nonlinear branches of the pyramidal neuron is unclear. Legenstein and Maass (2011) revealed that the pyramidal neuron can solve the linearly non-separable feature binding problem. An integrated model is proposed for nonlinear dendritic computation using synaptic plasticity and branch-strength potentiation. Limbacher and Legenstein (2020) extended this idea with synaptic rewiring which shows that the locations of the synaptic receptor are also important for learning. The synaptic plasticity rule utilized in (Legenstein and Maass, 2011) and (Limbacher and Legenstein, 2020) is spike time dependent plasticity rule. In a recent study, Rao et al

(2022) employ Dendritic Logistic Regression to define connection weights. Furthermore Moldwin and Segev (2020) used perceptron rule to define synaptic connection weights. These studies show that the learning in pyramidal neurons is possible and requires to be improved. Their studies investigated the learning rules for pyramidal neurons, but they did not use real-world data. The data used by these studies were dummy data which is chosen for showing the models could perform well. It worth to say that the real capabilities of methods can only be seen if they are forced to solve a real world problem.

In this study, we investigated the performance of a pyramidal neuron model in the classification problem of a real-world dataset. Using ECG signals from MIT-BIH Dataset, we investigated whether the pyramidal neuron could classify healthy and arrythmic beats, successfully. Since the pyramidal neurons have complex structure and their dendritic morphology and biophysics vary in numerous aspects, we first designed our study to ensure that the regional specific features of the neuron can easily be demonstrated. The output of pyramidal neurons is usually affected by the locations of the synapses since the locations are related to various layers of the brain. Therefore, we elaborated our experiment such that the effect of the synaptic locations can be observed. For this purpose, we performed our experiment for various synaptic locations while the rest of the components of the experiment were fixed. Since the study omits the synaptic inhibution and considers only the synaptic excitation in neurons, we discussed whether we could compensate the inexistence of inhibitory synapses. We suggested that extending spike patterns with their mirror versions can improve the classification performance of the pyramidal neurons. Considering all above, our study's value can be stated as follows:

- The pyramidal neuron's ability to classify real world data was investigated for the first time using ECG dataset.
- Gray coding was chosen as the spike pattern generator for ECG Data, which as far as we know no study has done that before.
- The effect of neuronal branches such as somatic, apical and basilar dendrites was investigated to observe whether their morphology or biophysics could make a difference in the classification performance.
- A new approach which extends the spike pattern to compensate inexistence of inhibitory synapse was suggested. Also, the approach was investigated and evaluated for usage in classification problems, which as far as we know no study has done that before.

The rest of the paper is organized as follows. In Sections 2.1, 2.2, 2.3 and 2.4,respectively, dataset, spike pattern generation method, the neuron model and perceptron learning rule briefly introduced. In Section 2.5, our experimental setup is detailed. In Section 3.1 and Section 3.2, we have shown our results by illustration with a figure and tables. In Section 3.3 the effect of neuronal region on classification, in Section 3.4 the effect of synaptic receptor locations, in Section 3.5 the robustness of the method and in Section 3.6 the benefits of extending inputs were examined. In Section 3, we also discuss the

results, after which we conclude the study with suggestions for future studies in Section 4.

2 Material and Methods

2.1 ECG Dataset

The ECG dataset used in this study were obtained from the MIT-BIH Arrhythmia Database on the Physionet site (Moody and Mark, 2001). This database contains 48 half-hour excerpts of two-channel ambulatory ECG recordings, obtained from 47 subjects. The record 228m contains R-R intervals which consists 362 premature ventricular contraction (PVC) and 1688 healthy beats. PVC is a too-early heartbeat that is initiated by Purkinje fibers in the ventricles rather than by the sinoatrial node and disrupts the heart's normal rhythm. The ECG signals of database were digitized at 11-bit resolution at 10 mV intervals with 360 samples per second and recorded from Lead II and one of the modified leads (V1, V2, V4, or V5). For this study, recordings were selected from Lead V1.

2.2 Conversion of Analog Signal to Spike Pattern using Gray Coding

While neurons communicate with each other through the spike sequence in their internal dynamics, external environmental stimuli are analog signals that change over time. External environmental stimuli are converted into action potential patterns, in other words, spike patterns, by sensory receptors.

An analog signal is converted by spike generation algorithms into a pattern of all-or-none spikes. Binary coding techniques used in digital computers are also included in these algorithms. The algorithms are better suited than others depending on the type of data to be dealt with (Auge et al, 2021). Using the Gray Coding (GC), the conversion of a ECG beat signal into a spike pattern consists of three steps: sampling, quantizing, and coding. Sampling is the conversion of analog signals to discrete time signals by recording observations in a predetermined periodical time which is called sampling time. Most of times the signals recorded for computers are already sampled for a given sampling time. Quantizing is basically sampling in the amplitude axis but slightly different. It is for expressing infinite value range as finite number of levels. Since the quantized data generally would be converted to binary codes, the number of levels is chosen as exponents of 2. In this study, we utilized GC for converting a ECG beat signal to a spike pattern. The technique is as simple as binary encoding, besides it also preserves the continuity of the signal. In this coding technique, there is only one bit position difference in code words corresponding to adjacent samples (Monteiro et al, 2022).

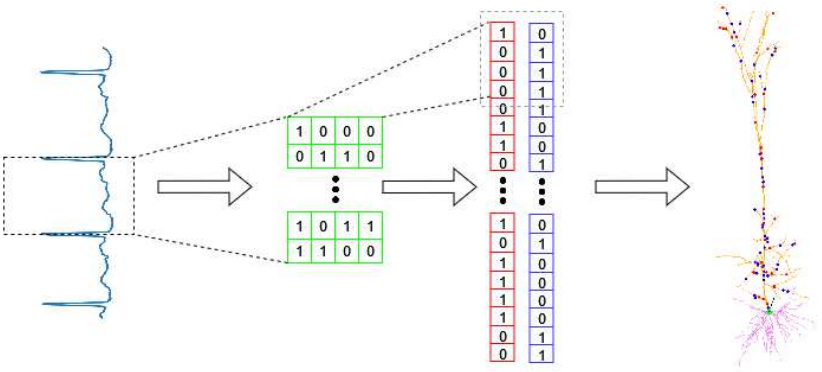

Fig. 1 The experimental setup for the study. The ECG data is partitions as R-R intervals and interpolated to 60 sample. The R-R interval were converted to gray codes (shown in green.). The gray codes were flattened (shown in red.). The mirror versions of the input bits were obtained (shown in blue) for extended inputs. The inputs were fed to pyramidal neuron (shown in the right side of the figure).

2.3 Neuron Model

Pyramidal neurons are one of the most common cell morphology types in mammals, birds, reptiles, and fish. Their inexistence in amphibians is discussed as pyramidal neurons may be correlated with advanced cortical functions (Spruston, 2008). Their dendritic diversity provides flexibility and versatility, and each pyramidal neuron has a unique dendritic structure (Galloni et al, 2020). Although their dendritic properties are diverse, they can usually be characterized as two main dendritic trees, basal and apical dendrites. Basal dendrites are located on the base of the neuron and apical dendrites are located on the apex. Mostly, the soma is connected to tuft via a relatively long apical dendrite. Studies suggest that more than one apical dendrite may connect the soma to the tuft. In our study, we have employed modelled cortical pyramidal cell given in (Hay et al, 2011). Their study includes three distinct neuron morphologies, but we included only one of them given in Figure 1.

Another dendritic property of a pyramidal neuron is the biophysics. Hay et al (2011) performed their study for two distinct experimental procedures, backpropagation-activated calcium spikes (BAC firing) and perisomatic step current firing. Consequently, they proposed four biophysical model sets. More recently, Shai et al (2015) proposed their study about the physiology of the same neurons. They re-fitted the biophysical parameters according to their own experimental data. Since our study has similar purposes, we used their parameter set as given in Figure S2 of (Shai et al, 2015).

Pyramidal neurons receive synaptic inputs from various cortical domains. Mostly, the synaptic wirings are related to the distance of dendrites to the soma. This leads researchers to cluster dendritic trees into domain specific regions. In our study we basically divided the neurons into four main clusters

as basilar dendrites, apical dendrites, soma, and axon as suggested in (Moldwin and Segev, 2020). Apical and basilar dendrites were clustered as they were given in the computational model in (Hay et al, 2011). Given that each cluster receives synaptic inputs from different domains, we performed our experiments in these domains separately.

2.4 Perceptron Learning

In this study, we used the perceptron learning algorithm to train the model. It is an algorithm for supervised learning of binary classifiers that uses adjustable weights to assign an input vector to a class. In the following, we will explain the basic notation and how to implement it in classification of two-class ECG data set using pyramidal neuron model.

We label the positive and negative class in our binary classification setting as "1" and "-1", respectively. The training set is denoted by:

$$D_T = \{(X_1, d_1), ..., (X_M, d_M)\} \tag{1}$$

where X_j is jth input vector of the training set of M vectors, and denoted by:

$$X_j = \begin{bmatrix} X_{1,j} \\ . \\ . \\ X_{i,j} \\ . \\ . \\ X_{N,j} \end{bmatrix} \qquad i = 1, .., N \tag{2}$$

where N is the dimension of input vectors. d_j is the label of the jth training input vector. It takes the value "+1" for normal ECG vectors and "-1" for arrhythmia ECG vectors. The synaptic weight vector is denoted by:

$$W = \begin{bmatrix} w_1 \\ . \\ . \\ w_i \\ . \\ . \\ w_N \end{bmatrix} \qquad i = 1, .., N \tag{3}$$

where w_i is the weight value corresponding to the ith synaptic input. The perceptron algorithm attemps to adjust the synaptic weights so as to minimize the sum-squared error E over all input vectors. It is given by:

$$E = \frac{1}{M} \sum_{j=1}^{M} E_j \tag{4}$$

where E_j is the error between the desired output d_j and the actual output o_j of a neuron on the jth training input vector, and is given by:

$$E_j = \frac{1}{2}\left(d_j - o_j\right)^2 \tag{5}$$

The value for updating the weights for jth input vector is calculated by the learning rule.

$$\Delta w_i = \eta\left(d_j - o_j\right) x_{i,j} \tag{6}$$

where η is the learning rate. The weights are updated by following iteration algorithm.

$$w_i\left(new\right) = w_i\left(old\right) + \Delta w_i \tag{7}$$

So far, the fundamentals of perceptron learning were given. But implementing perceptron learning has some limitations for usage in biological neurons since the weights of the synapses cannot be negative. This is because weights manage the conductivity of the synapses and negative weights corresponds to negative conductivity and even though this is possible in theory it is impossible for conductivity to be negative in real life. Therefore, the equation should be re-arranged for the task as follows:

$$w_i\left(new\right) = max\left(0, w_i\left(old\right) + \Delta w_i\right) \tag{8}$$

This equation states that the new weights should be set to zero in case they are in negative values after the update of the weights step. To sum up, the equations given above can be re-written as follows:

$$w_i(new) = \begin{cases} w_i(old) & if \quad d_j = o_j \\ max\left(0, w_i\left(old\right) + \Delta w_i\right) & if \quad d_j \neq o_j \end{cases} \tag{9}$$

It is wort to mention that all weights are updated simultaneously. Let us make a simple thought experiment to illustrate how the learning rule works. In case of $x_{i,j} = 0$, the weight w_i will not change. However, in case of $x_{i,j} = 1$ and the prediction is wrong, the weight w_i will be pushed towards the direction of the positive or negative target class, respectively:

$$\Delta w_i = \eta\left(d_j - o_j\right) x_{i,j} = \eta\left(1 - (-1)\right) 1 = 2\eta \tag{10}$$

$$\Delta w_i = \eta\left(d_j - o_j\right) x_{i,j} = \eta\left(-1 - 1\right) 1 = -2\eta \tag{11}$$

The strict rule that the weight values cannot be negative raises a vital issue that the input vectors which need to be weighted with negative values becomes useless as they will not transfer their information to the neuron even though they are important. In real, there are two basic types of synapses which are inhibitory and excitatory. Excitatory synapses motivate the neuron to spike, which we assume all the synapses in the study are excitatory, and inhibitory synapses does the opposite, which we did not include in our study. Excitation and inhibition are controlled over the synapses' reversal potentials and the perceptron rule only regards the weight changes. Therefore, it is impossible for perceptron learning rule to switch one to another.

In order to compensate the negative weight effect, we extended the input vectors by adding the flipped form of the vector beside it. In the flipped form, 1s are made 0 and 0s are 1. It is worth to mention that the size of the extended input vector becomes twice the size of the input vector. For the extended input vectors, the weights should increase if the mirror version of weight tend to decrease to zero value. Also, In case of $x_{i,j} = 0$, the weight w_i would not change. By this approach, the flipped version of $x_{i,j}$ will be 1. Thus, $x_{i,j}$ will also contribute to the learning algorithm.

2.5 Experimental Setup

Biological neurons require spike patterns to process. But the ECG data in this study was recorded in range variables. The first step for this study was to convert data from ECG to binary inputs. For this purpose, the data was arranged as R-R intervals as given in the database. Since the heart rate may differ for various conditions, the R-R intervals have various number of samples. Therefore we interpolated the R-R intervals to 60 samples so that they had the same number of inputs. The data consists of one dimensional 60 samples in this stage. To convert the R-R intervals to binary code, we employed GC technique. Based on our experimental studies, the code length was selected as 4 for GC, consequently the number of quantization levels was set to 16. Therefore, the data was converted to two-dimensional code samples with size of 60x4. Since biological neuron cannot deal with two-dimensional code input, we flattened the code so that they can be one dimensional with code size of 240. For the further experiments, we obtained the mirror versions of the code set by flipping each of the bit. After all, we extended the input vector with their flipped versions of themselves. So, we obtained two sets of dataset with the lengths of 240 and 480. The process was illustrated in Figure 1.

Considering the data balance, 600 beats were randomly selected from the record 228m, such a way that 300 of which would be healthy beats, remaing 300 PVC beats. Since the machine learning has two main stages as training and testing, we shuffled the samples and split into two subdatasets (training and testing data). This process was repeated for two times more and we obtained three inputs datasets (namely as Set 1, Set 2, and Set 3) to be used in the experiments. Also, we repeated the process for both the 240-sized and 480-sized datasets. The process was illustrated in Figure 1.

So far, the data processing stage was explained. But the pyramidal neurons cannot process spikes by their own. They require synaptic receptors to gather the spikes. The pyramidal neuron should have more synaptic receptors than the inputs to obtain data properly. So, we modified the pyramidal neuron by adding synaptic receptors according to number of inputs. As given in Section 2.3, the pyramidal neurons consist four main clusters of branches as somatic, apical, basilar and axonal. We grouped these regional clusters as somatic, apical, basilar and apical + basilar. So, we can seperately observe the regional effect of synaptic receptors. The groups are distinct in two reasons which are their distance to the soma and their biophysics. Somatic group which only

contains soma has both active and passive ion channels. Apical group is distant the soma and has both active and passive ion channels but the active ion channels have no ability to emit an action potential by their own. Basilar group is fairly near to soma but have only passive ion channels. The apical + basilar group was selected to observe if any advantage or disadvantage may occur if apical and basilar clusters were combined.

The second phase of the experimental process was the preparation of the pyramidal neuron for the synaptic inputs. First, one of the regions given above is selected. Then, according to the number of inputs, the synaptic receptor locations are randomly distributed across the neuron. Synaptic receptor distribution was done homogenously according to surface area. After the synaptic receptor locations are determined, synapses are assigned to given locations, randomly. Synaptic connections is modelled with a exponetially decaying function with reversel potential of $0mV$ and time constant of $1.7ms$. Since the locations of the synapses may affect the performance of the neuron, we repeated the distribution step two more times and obtained three synaptic location sets (namely as Syn Set 1, Syn Set 2 and Syn Set 3) in total for each regional group.

The ECG data was converted from analogous signal to binary input vector (spike pattern) and the pyramidal neuron was modified by adding synaptic receptors according to number of input vector. The last step of the experimental process is to combine these as a whole to simulate. First, the inputs are randomly assigned to the synapses. An ECG sample is selected to simulate. Each input bit of the sample is matched with their own synaptic receptors. If the input bit for the corresponding synapse is 1, the synaptic receptor is activated, else it is de-activated. So, each bit was matched with their own synapse and the synapses were activated/de-activated according to their bits. After all, the neuron was simulated for 50 ms. In the 10th ms of the simulation, the active synapses were stimulated with corresponding spikes of bits. In the end of the simulation, if the neuron emits at least one spike was assigned as "1" and if it never emits a spike it was assigned as "-1".

Simulations were performed with the NEURON simulation environment, interfaced with the Phython. The addition of Python allowed the use of a very comprehensive analysis tools in the simulation. We conducted simulation experiments on the computer with an AMD RyzenX 2700 8-core CPU and 16GB RAM.

3 Results and Discussion

A brief introduction to experimental setup was done in Section 2.5. In that section, all the process was explained for a single sample of ECG data. In this section, we extend the experimental setup in a holistic manner and present the results obtained.

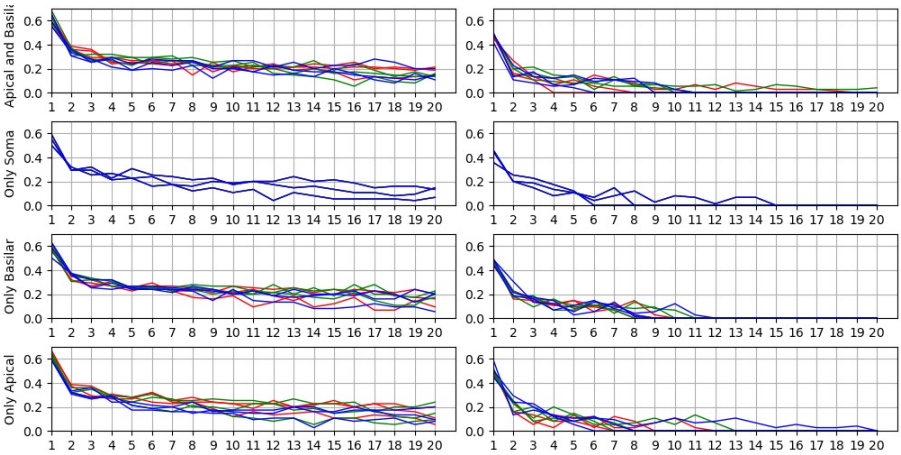

Fig. 2 MSE metrics throughout the training. The figure shows the results of the trials which are fed by pure (graphs which are in left side of the figure) and extended inputs(graphs which are in right side of the figure). The graphs in rows shows the results for the regions accross the pyramidal neuron. Each dataset was represented by a color in the graphs.

3.1 Training

Training is the step which is done for finding the appropriate parameters of the models. The pyramidal neuron requires the synaptic weights to be determined to successfully classify the samples. Since our study examines the effects of various aspects of the pyramidal neuron, the training procedure was as given below:

First of all, the input dataset (i.e., Set 1) to be used is selected. The synaptic receptor locations on the pyramidal neuron's selected regional group (i.e. Apical) were determined. Note that the study aims to investigate the effect of the synaptic locations. Therefore, we determined three separate synaptic locations, namely syn sets. So, three identical pyramidal neurons were created. For each neuron, three syn sets are determined to be located. The synaptic locations for each neuron were matched and the input dataset was applied to neurons simultaneously. So, there is three pyramidal neuron which are morphologically and biophysically identical but different for their synaptic receptor locations. The training data was fed to pyramidal neurons beat by beat and the weights were updated individually according to the outputs of the pyramidal neurons. In the beginning of the training, all the synaptic weights were set to zero since we wanted to simulate the fact that no connections were made between neurons in the beginning before learning. This training procedure was done for 20 epochs. The learning rate was chosen as $\eta = 5e - 5$. For each epoch, the MSE metric was calculated and shown in Figure 2. Also, we equivalent trained single layer perceptron (SLP) models for each dataset so that we could compare the results with ours.

Table 1 Training results using pure inputs.

Region	Syn Set	Accuracy			Precision			Recall		
		Set 1	Set 2	Set 3	Set 1	Set 2	Set 3	Set 1	Set 2	Set 3
Only Soma	1	97.33	97.67	99.33	96.64	99.28	100.00	97.96	95.80	98.58
	2	97.33	97.67	99.33	96.64	99.28	100.00	97.96	95.80	98.58
	3	97.33	97.67	99.33	96.64	99.28	100.00	97.96	95.80	98.58
	Mean±Std	**98.11 ± 0.93**			**98.64 ± 1.53**			**97.45 ± 1.26**		
Only Basilar	1	96.67	97.33	96.33	96.60	97.87	92.76	96.60	96.50	100.00
	2	96.67	97.00	97.67	95.97	99.26	97.86	97.28	94.41	97.16
	3	96.33	97.67	99.00	94.74	100.00	100.00	97.96	95.10	97.87
	Mean±Std	**97.19 ± 0.85**			**97.23 ± 2.46**			**96.99 ± 1.64**		
Only Apical	1	95.67	96.67	99.33	95.89	97.16	99.29	95.24	95.80	99.29
	2	96.33	92.33	99.33	94.16	87.50	100.00	98.64	97.90	98.58
	3	97.00	97.33	99.67	96.62	99.27	100.00	97.28	95.10	99.29
	Mean±Std	**97.07 ± 2.30**			**96.65 ± 3.99**			**97.46 ± 1.69**		
Apical and Basilar	1	95.33	95.67	97.33	92.90	100.00	99.26	97.96	90.91	95.04
	2	96.67	98.67	99.00	95.97	100.00	99.29	97.28	97.20	98.58
	3	94.67	94.67	98.33	92.26	99.22	97.89	97.28	89.51	98.58
	Mean±Std	**96.70 ± 1.71**			**97.42 ± 3.02**			**95.82 ± 3.36**		
Single Layer Perceptron		**99.83**	**99.83**	**99.67**	**99.66**	**99.65**	**99.30**	100.00	100.00	100.00

After the training was done, we wanted to see if the training data was successfully classified. For each trial, we re-classified the training data. Classification performance for all these cases was evaluated using accuracy, precision and recall metrics.

As can be seen from Table 1, the average training accuracies in pure input synaptic stimulations to the apical, basilar, somatic and apical+basilar regional groups are %98.11, %97.19, %97.07 and %96.70, respectively. In the training with pure input, the average precision values are in the range of %97-%99, and the standard deviations are in the range of 1.5-4.0, as can be seen in Table 1. The mean and standard deviations of the recall values in the same experiment are in the ranges of %96.00-%98.00 and 1.25-3.40, respectively. Accordingly, the pure inputs datasets were classified the data with accuracies of minimum %96.70. When the extended inputs were used almost all the training accuracies were %100. The training results of pure inputs datasets were mapped on Table 1. Since almost all the training accuracies of the extended inputs were %100, we did not include the table of them in the study.

3.2 Testing

As explained in Section 2.5, we shuffled and split data as training and testing data before training phase. We trained the pyramidal neuron only by using the training data. The goal of the classification is assigning the testing data labels accurately. The testing phase is the most important step since the model could be declared as successful if the testing results are accurate. As can be seen above, the testing data was never used in any process of the study before

training. Therefore, the real-world problem can be solved for a given success rate if the testing phase is successful enough.

The study can be divided into two main experiments as investigating pure inputs and extended inputs. In pure inputs, our testing results were mapped in Table 2. Depending on the neuronal regions, the accuracies were varying between %88.33 and %93.33 and standard deviations are below 1.74 as can be seen from Table 2. The mean and standard deviation of the precision values in the same experiment are in the ranges of %90.78-%92.32 and 2.58-2.77 respectively. In the testing with pure input, the average recall values are in the range of %90.79-%93.21, and the standard deviations are in the range of 2.35-4.02, as can be seen in Table 2.

Comparing with SLP, the testing results with pure inputs are not very successful. It seems pyramidal neuron is not an appropriate choice compared to SLP. But when we go in deeper by examining extended inputs result, the pyramidal neuron catches up the SLP and even overtakes it. The results of extended inputs are given in Table 3. The accuracies were varying between %94.67 and %95.33 and standard deviations are below 0.87 as can be seen from the Table 3. Similarly, the mean and standard deviation of the precision values in the same experiment are in the ranges of %93.35-%95.26 and 1.66-3.10 respectively. In the testing with extended inputs, the average recall values are in the range of %95.82-%96.84, and the standard deviations are in the range of 1.54-2.87, as can be seen in Table 3. The accuracy of the pyramidal neuron increases dramatically compared to pure inputs. In some trials, pyramidal neuron which is fed by the extended inputs outperforms SLP model. This shows that the capabilities of the pyramidal neuron increases if the extended inputs were fed.

3.3 Investigating Neuronal Regions on Classification

In Section 2.3, the features of pyramidal neuron were briefly introduced. Accordingly, the pyramidal neuron branches can be clustered into some regions considering their topologies, morphologies, biophysics, and their connectivity. Studies have shown that the synaptic connectivity of the branches may correspond various layers of the brain. Also, the biophysical features of the neuron may differ in terms of the locations of the branches. Biophysics are also differed one neuron from another. The neuron shape can be distinctive among other biological neurons (Stuart and Spruston, 2015). In the study, we have chosen pyramidal neuron which is widely seen in the brain. They receive inputs from various layers of the brain. This makes them a perfect candidate for a classification study. Also, pyramidal neurons can be clustered into four main clusters of the branches. Soma which is the main processing unit of the neuron has ion channels that leads neuron to emit spikes if the conditions are appropriate. Their capability of emitting spikes makes them the most important section of the neuron. Axon is the main section responsible for propagating the emitted spike. It is important that the spikes emitted in soma must propagated (Spruston, 2008). But in the study, we only investigated the emitted spikes but

Table 2 Testing results using pure inputs.

Region	Syn Set	Accuracy			Precision			Recall		
		Set 1	Set 2	Set 3	Set 1	Set 2	Set 3	Set 1	Set 2	Set 3
Only Soma	1	92.00	92.67	92.33	88.62	95.30	93.04	96.73	90.45	92.45
	2	92.00	92.67	92.33	88.62	95.30	93.04	96.73	90.45	92.45
	3	92.00	92.67	92.33	88.62	95.30	93.04	96.73	90.45	92.45
	Mean±Std	**92.33 ± 0.29**			**92.32 ± 2.94**			**93.21 ± 2.78**		
Only Basilar	1	91.00	91.67	92.00	90.91	92.31	90.42	91.50	91.72	94.97
	2	90.00	91.00	90.33	88.68	93.33	91.67	92.16	89.17	89.94
	3	89.67	91.33	91.33	86.31	95.17	91.82	94.77	87.90	91.82
	Mean±Std	**90.93 ± 0.78**			**91.18 ± 2.58**			**91.55 ± 2.35**		
Only Apical	1	89.00	90.67	92.33	87.04	92.72	91.98	92.16	89.17	93.71
	2	91.67	88.33	92.00	88.10	87.20	92.99	96.73	91.08	91.82
	3	93.33	92.33	93.00	90.30	94.08	92.59	97.39	91.08	94.34
	Mean±Std	**91.41 ± 1.74**			**90.78 ± 2.71**			**93.05 ± 2.73**		
Apical and Basilar	1	89.67	88.67	89.67	87.20	93.62	91.56	93.46	84.08	88.68
	2	90.67	92.00	91.67	88.34	92.90	91.36	94.12	91.72	93.08
	3	90.00	89.33	91.67	86.83	94.33	91.88	94.77	84.71	92.45
	Mean±Std	**90.37 ± 1.18**			**90.89 ± 2.77**			**90.79 ± 4.02**		
Single Layer Perceptron		**94.50**	**94.79**	**95.29**	**92.00**	**92.43**	**95.74**	**97.71**	**98.09**	**95.36**

not the propagation of spikes across the neuron. Therefore, we did not include the axon in our study for synaptic input placement. Basilar dendrites are the branches which are located at the base of the neuron and has only passive ion channels. This means that they cannot emit any spikes by themselves but only they can transmit the input signals to the soma for further decisive processing. Apical branches which are located in the apex of the neuron receives inputs just like basilar branches, but they require active ion channels since their distance to soma are relatively high. Though they contain active ion channels, the nature of their ion channels has no ability to spike. The purpose of the active ion channels in the apical branches are transmitting the inputs without any attenuation. So, there is four main clusters of branches in pyramidal neuron which are somatic, axonal, apical and basilar (Magee, 2000). In the study we investigated the effects of the regions by grouping them as somatic, apical, basilar and apical + basilar. The results showed that the regional effect of the classification is genuine.

It can be easily seen by looking at Tables 1, 2 and 3 that the regions where the synaptic receptors were located affects the performance directly. A superficial examination on tables suggests that when the distance of the synaptic location from the soma increases the performance tends to decrease. In addition to this, when looking deeper, it is a fact that the standard deviation of the accuracies increases when the distance from soma increases. This can be explained by the increase in distance range across the regions and the increase of the possibility of a single synaptic receptor to be placed in a distant location, which increases the possibility of a relative important synaptic connection to be placed far from the soma, the processor.

Table 3 Testing results using extended inputs.

Region	Syn Set	Accuracy			Precision			Recall		
		Set 1	Set 2	Set 3	Set 1	Set 2	Set 3	Set 1	Set 2	Set 3
Only Soma	1	94.33	94.67	95.00	90.48	92.22	97.37	99.35	98.09	93.08
	2	94.33	94.67	95.00	90.48	92.22	97.37	99.35	98.09	93.08
	3	94.33	94.67	95.00	90.48	92.22	97.37	99.35	98.09	93.08
	Mean±Std	**94.67 ± 0.29**			**93.35 ± 3.10**			**96.84 ± 2.87**		
Only Basilar	1	94.67	96.00	95.33	91.52	95.03	96.77	98.69	97.45	94.34
	2	95.00	96.00	95.00	92.59	95.03	96.15	98.04	97.45	94.34
	3	95.33	93.67	94.67	93.71	94.81	93.87	97.39	92.99	96.23
	Mean±Std	**95.07 ± 0.72**			**94.39 ± 1.66**			**96.32 ± 1.97**		
Only Apical	1	95.67	96.67	95.67	94.30	96.82	97.40	97.39	96.82	94.34
	2	95.67	94.33	95.33	93.75	94.87	96.18	98.04	94.27	94.97
	3	93.67	95.67	95.33	91.36	94.44	96.77	96.73	97.45	94.34
	Mean±Std	**95.33 ± 0.87**			**95.10 ± 1.91**			**96.04 ± 1.54**		
Apical and Basilar	1	96.00	94.67	95.33	97.32	92.73	97.39	94.77	97.45	93.71
	2	96.67	94.33	95.67	96.13	93.21	96.20	97.39	96.18	95.60
	3	95.67	95.00	94.33	94.87	92.77	96.71	96.73	98.09	92.45
	Mean±Std	**95.30 ± 0.79**			**95.26 ± 1.92**			**95.82 ± 1.88**		
Single Layer Perceptron		**94.50**	**94.79**	**95.29**	**92.00**	**92.43**	**95.74**	**97.71**	**98.09**	**95.36**

3.4 Investigating the Effect of Synaptic Input Locations in Regions

Previous section emphasizes that the regions where the synaptic receptors were located affects the performance of the pyramidal neuron. When looking deeper at the synaptic locations within the regions, the performance also fluctuates. Examining Tables 1, 2 and 3, the accuracy of the pyramidal neuron varies according to synaptic locations sets except for soma. Soma does not get affected by the synaptic location set since it is the processing center of the neuron and only constitutes from a single section. Therefore, defining synaptic receptor locations are not a critical step for soma since each location is the same location theoretically. But this is not applied to rest of the regions when looking at the results. The apical, basilar and apical + basilar groups seems to be sensitive where the synapses were located. Obviously, they react differently for the same data when the synaptic locations change.

If the reasons why the pyramidal neuron's classification performance got affected by the synaptic locations were sorted, the winner and the runner-up would be distance and biophysical inconsistency across the pyramidal neuron. Assuming an input connection whose corresponding inputs are highly correlated with the labels, as it spikes when the label is "1" and does not spike when the labels is "-1". In theory, only this input is enough for the model to classify data accurately. But when it is located on the pyramidal neuron, the distance is important for this synaptic input to excite the soma. If it was located far from the soma, the input would be so distant that it would not be enough for the branches to propagate the intracellular signal to soma since the signal

was attenuated throughout the branches. Therefore, the neuron cannot benefit enough from this hypothetical input because of the distance. Another reason is the biophysical variability of the dendrites. As can be seen from the model, the ion channels' distribution, their parameters, the topology, and the morphology are not consistent across the neuron. The geometrical features such as length and radius are nearly unique for each dendrite. Also, their biophysical parameters are distributed throughout the dendrites rather than being consistent. Regarding our hypothetical synaptic input exampled above, locating the synaptic receptor in the appropriate place is matter of chance if the synaptic receptor distribution was done randomly. The reasons explained above suggests that synaptic receptor locations play a key role in the performance of the pyramidal neuron such that the tables show that the pyramidal neuron can be more successful than the SLP or vice versa if the locations were determined appropriately. Another suggestion of these reasons is that learning just by the weight of the synapses is not enough, also learning by the synaptic locations was required. This issue was not raised for the SNN's because their neuron content constitutes only somas which are shown that they are not affected by the synaptic locations.

3.5 Generalization Task

The performance of the classifier on a given dataset may be illusory depending on how the dataset exemplifies the universal cluster for the task. The classifiers require training data to define their interior parameters to perform accurately. In most cases, more data means more accurate classifiers. Therefore, the ultimate classifier which yields the most accurate results is built up from all the data recorded. However, this is impossible for real life applications since it is not possible to gather all the data and most of times, the classifiers must deal with the problem with less data as possible. Therefore, the dataset chosen for classifiers to be trained should be as representative as possible.

In Section 2.5, we have mentioned that we shuffled and split the data into two as training and testing data, randomly. The results obtained may be illusory since the training data may be too separable or vice versa because of the randomizing. This would cause misleading judgements such as the classifier could perform well or poorly. Such a judgement would mean nothing scientifically. Therefore, we obtained three different datasets by shuffling and splitting data repeatedly. By considering them as distinct datasets, we wanted to see how the pyramidal neuron generalizes the data.

Examining the Tables 1, 2 and 3, it is fair to say that the pyramidal neuron can deal with generalization task. Even if the training datasets chanced, it represented the information encoded in data satisfyingly. The tables indicates that the pyramidal neuron acts similar as the SLP model for the generalization task. In this perspective of view, pyramidal neurons are satisfactory in term of data generalization.

3.6 Extending the Inputs

Perceptron learning was first proposed for the MP neurons which assume that the neuron gathers the inputs, obtains weighted average of the inputs and fires if the weighted average is above a predefined value. This approach makes the weight values important for neurons to perform properly. In MP neurons, the weight values range from negative infinity to positive infinity. In biological neurons, weights corresponds to efficacy which indicates how strongly the pre-synaptic neuron was connected to the post-synaptic neuron. When the pre-synaptic neuron fires, neurotransmitters are released from the presynaptic neuron and causes increase in synaptic current (for excitatory synapse) in the post-synaptic neuron. The increase in the synaptic current is correlated with the synaptic efficacy between the neurons, which the larger efficacy value causes larger synaptic current increase. The synaptic efficacy must be a positive value by nature since it is related to synaptic conductivity and the conductivity is a value which must be positive.

In MP neurons, the sign of the weights are not important, on the contrary in biological neurons synaptic weights must be positive. Since the perceptron learning was designed for the MP neuron, no regulations which limits the weights are required. Moldwin and Segev (2020) also mentioned about this issue stating that they limit their algorithm such that they set the values to zero when it is required to be negative. Their study performed considerably well but when we applied the algorithm on real world data, this weight limitation also limited our study's performance. The first reason why the performance of the pyramidal neuron was limited was the weight limitation. Secondly, when the data was examined deeper, we observed that the weight change equals zero if the input bit is zero in Equation 6. This means that the input does not affect the synapse for learning. The inputs do not have to affect the synapse, but it means lack of information if the input has information. Third, when the weights from a perceptron learning were examined, it can be easily seen that the inputs which are negatively correlated with the output yields negative weight and vice versa. This shows that the sign of the weight is the same with the sign of the correlation value between the input and output. In our study, since we set the weights to zero if they required to be negative, the information from inputs which are negatively correlated with output is wasted.

The issues raised above constitute very big limitations on the performance of the pyramidal neuron since it cannot perform as well as SLP model when used as suggested in (Moldwin and Segev, 2020). Table 2 advertises this issue as the SLP model outperforms the pyramidal neuron when we used only the pure inputs. Fortunately, the solution was the same for all three issues mentioned above: extending the features with their mirror versions. The extending procedure was detailed in Section 2.4. Table 3 shows the results of the extended inputs which clearly advertises that the pyramidal neuron can perform as well as the SLP model when the extended features were used.

The benefit of extending features is that it overwhelms the "weight positivity barrier". If a weight requires to be negative, this mean that the input and

output are negatively corelated, the weight is set to zero, but the mirror version requires a positive weight value since the correlation between the mirror input and the output is positive. So, the information in the input is not wasted. This approach increased the performance of the pyramidal neuron dramatically.

4 Conclusion

One of the most important questions in neuroscience is understanding how networks in the brain process input information appropriately and perform learning and memory mechanisms. So far, many experimental and modeling studies have been done to understand the learning mechanism of pyramidal neurons. In recent years, successful results have been obtained in classification problems based on pyramidal neuron models. In this study, we investigated the performance of a pyramidal neuron model in the classification problem of a two-class ECG dataset for different density inputs to different synaptic regions. In order to apply the continuous ECG signal to the selected model, the signal was converted into a spike sequence with the GC technique.

The classification performance of the proposed method was extensively analyzed with different scenarios, including the use of three different datasets, stimulation of four different region group, use of input vectors of two different lengths, and selection of different synaptic input locations. According to the results obtained, it was observed that the classification performance did not change significantly in the different datasets, increased in dendritic regions close to the soma, increased in the case of using extended input vectors, and was significantly affected by selecting different synaptic input locations. The classification performance was tested with an SLP algorithm, which was trained and tested with the same datasets. It has been observed that the proposed method has yielded similar performance with the SLP algorithm, especially in extended inputs. These results encouraged the classification of multi-class datasets with different architectures in our future studies.

Competing interests

The authors have no conflicts of interest to declare that are relevant to the content of this article.

References

Alzubaidi L, Zhang J, Humaidi AJ, et al (2021) Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions. Journal of big Data 8(1):1–74

Auge D, Hille J, Mueller E, et al (2021) A survey of encoding techniques for signal processing in spiking neural networks. Neural Processing Letters 53(6):4693–4710

Bose P, Kasabov NK, Bruzzone L, et al (2016) Spiking neural networks for crop yield estimation based on spatiotemporal analysis of image time series. IEEE Transactions on Geoscience and Remote Sensing 54(11):6563–6573

Burkitt AN (2006) A review of the integrate-and-fire neuron model: II. Inhomogeneous synaptic input and network properties. Biological cybernetics 95(2):97–112

DeFelipe J (2011) The evolution of the brain, the human nature of cortical circuits, and intellectual creativity. Frontiers in neuroanatomy p 29

Doupe P, Faghmous J, Basu S (2019) Machine learning for health services researchers. Value in Health 22(7):808–815

Farsa EZ, Ahmadi A, Maleki MA, et al (2019) A low-cost high-speed neuromorphic hardware based on spiking neural network. IEEE Transactions on Circuits and Systems II: Express Briefs 66(9):1582–1586

Galloni AR, Laffere A, Rancz E (2020) Apical length governs computational diversity of layer 5 pyramidal neurons. Elife 9:e55,761

Gerstner W, Kistler WM (2002) Spiking neuron models: Single neurons, populations, plasticity. Cambridge university press

Gerstner W, Kistler WM, Naud R, et al (2014) Neuronal dynamics: From single neurons to networks and models of cognition. Cambridge University Press

Hay E, Hill S, Schürmann F, et al (2011) Models of neocortical layer 5b pyramidal cells capturing a wide range of dendritic and perisomatic active properties. PLoS computational biology 7(7):e1002,107

Hodgkin AL, Huxley AF (1952) A quantitative description of membrane current and its application to conduction and excitation in nerve. The Journal of physiology 117(4):500

Hosseini MP, Hosseini A, Ahi K (2020) A review on machine learning for EEG signal processing in bioengineering. IEEE reviews in biomedical engineering 14:204–218

Izhikevich EM (2007) Dynamical systems in neuroscience. MIT press

Janiesch C, Zschech P, Heinrich K (2021) Machine learning and deep learning. Electronic Markets 31(3):685–695

Lapique L (1907) Recherches quantitatives sur l'excitation electrique des nerfs traitee comme une polarization. Journal of Physiology and Pathololgy 9:620–635

LeCun Y, Bengio Y, Hinton G (2015) Deep learning. nature 521(7553):436–444

Legenstein R, Maass W (2011) Branch-specific plasticity enables self-organization of nonlinear computation in single neurons. Journal of Neuroscience 31(30):10,787–10,802

Limbacher T, Legenstein R (2020) Emergence of stable synaptic clusters on dendrites through synaptic rewiring. Frontiers in computational neuroscience 14:57

London M, Häusser M (2005) Dendritic computation. Annu Rev Neurosci 28:503–532

Magee JC (2000) Dendritic integration of excitatory synaptic input. Nature Reviews Neuroscience 1(3):181–190

Mahesh B (2020) Machine learning algorithms-a review. International Journal of Science and Research (IJSR)[Internet] 9:381–386

McCulloch WS, Pitts W (1943) A logical calculus of the ideas immanent in nervous activity. The bulletin of mathematical biophysics 5(4):115–133

Moldwin T, Segev I (2020) Perceptron learning and classification in a modeled cortical pyramidal cell. Frontiers in computational neuroscience 14:33

Monteiro J, Pedro A, Silva AJ (2022) A Gray Code model for the encoding of grid cells in the Entorhinal Cortex. Neural Computing and Applications 34(3):2287–2306

Moody GB, Mark RG (2001) The impact of the MIT-BIH arrhythmia database. IEEE Engineering in Medicine and Biology Magazine 20(3):45–50

Nassif AB, Talib MA, Nasir Q, et al (2021) Machine learning for cloud security: a systematic review. IEEE Access 9:20,717–20,735

Poirazi P, Brannon T, Mel BW (2003) Pyramidal neuron as two-layer neural network. Neuron 37(6):989–999

Polsky A, Mel BW, Schiller J (2004) Computational subunits in thin dendrites of pyramidal cells. Nature neuroscience 7(6):621–627

Ponulak F, Kasinski A (2011) Introduction to spiking neural networks: Information processing, learning and applications. Acta neurobiologiae experimentalis 71(4):409–433

Prieto A, Prieto B, Ortigosa EM, et al (2016) Neural networks: An overview of early research, current frameworks and new challenges. Neurocomputing 214:242–268

Rao A, Legenstein R, Subramoney A, et al (2022) Self-supervised learning of probabilistic prediction through synaptic plasticity in apical dendrites: A normative model. bioRxiv pp 2021–03

Rosenblatt F (1957) The perceptron, a perceiving and recognizing automaton Project Para. Cornell Aeronautical Laboratory

Rumelhart DE, Hinton GE, Williams RJ (1986) Learning representations by back-propagating errors. nature 323(6088):533–536

Shai AS, Anastassiou CA, Larkum ME, et al (2015) Physiology of layer 5 pyramidal neurons in mouse primary visual cortex: coincidence detection through bursting. PLoS computational biology 11(3):e1004,090

Shrestha A, Mahmood A (2019) Review of deep learning algorithms and architectures. IEEE access 7:53,040–53,065

Spruston N (2008) Pyramidal neurons: dendritic structure and synaptic integration. Nature Reviews Neuroscience 9(3):206–221

Stuart GJ, Spruston N (2015) Dendritic integration: 60 years of progress. Nature neuroscience 18(12):1713–1721

Wang X, Lin X, Dang X (2020) Supervised learning in spiking neural networks: A review of algorithms and evaluations. Neural Networks 125:258–280

Zerouaoui H, Idri A (2021) Reviewing machine learning and image processing based decision-making systems for breast cancer imaging. Journal of Medical Systems 45(1):1–20