SYNANN: A SYNAPTIC NEURAL NETWORK AND SYNAPSE LEARNING

Chang LI *
changli@neatware.com

ABSTRACT

A Synaptic Neural Network (SynaNN) consists of synapses and neurons. Inspired by the synapse research of neuroscience, we built a synapse model with a nonlinear and log-concave synapse function of excitatory and inhibitory probabilities of channels. Introduced the concept of surprisal space and constructed a commutative diagram, we proved that the function of inhibitory probability $-log(1-e^{-x})$ in surprisal space is the topologically conjugate function of the inhibitory complementary probability $1 - x$ in probability space. Furthermore, we found that the derivative of the synapse over the parameter in the surprisal space is equal to the negative Bose-Einstein distribution. The inhibitory synapse parameter in the surprisal space is the negative chemical potential of the synapse, so SynaNN is an explainable neural network. Moreover, we constructed a synapse block as a fully-connected synapse tensor and proved the gradient formula of a cross-entropy loss function, so that synapse learning can work with the gradient descent and backpropagation algorithms. In the experiment, we performed an MNIST training and testing on the MLP model with synapse network as hidden layers and achieved good accuracy similar to the popular neural network. In summary, we presented an abstract synapse model from an exact nonlinear synapse function based on computational neuroscience analysis, explored synapse network and synapse learning, and built the synapse representation in surprisal space.

1 INTRODUCTION

Synapses play an important role in biological neural networks (Hubel & Kandel (1979)). They are joint points of the connection of neurons where learning and memory happened. Based on the analysis of excitatory and inhibitory channels of synapses(Hubel & Kandel (1979)), we proposed a probability model (Feller (2008) for probability introduction) of the synapse along with a nonlinear function of excitatory and inhibitory probabilities (Li (2017)(synapse function)). Inspired by the concept of surprisal from (Jones (1979)(self-information), Levine (2009), Bernstein & Levine (1972)(surprisal analysis), Levy (2008)(surprisal theory in language)) or negative logarithmic space (Miyashita et al. (2016)), we proposed the concept of surprisal space and represented the synapse function as the addition of the excitatory function and inhibitory function in the surprisal space. By applying a commutative diagram, we figured out the fine structure of inhibitory function and proved that it was the topologically conjugate function of an inhibitory function. After the proof of the log-concavity of the synapse function, we can apply the latest mathematical concepts and tools on studying the synaptic neural network (Vinzant et al. (2018)(log-concavity). Moreover, we discovered (rediscovered) that the derivative of the inhibitory function over parameter was equal to the negative Bose-Einstein distribution (Nave (2018)). Furthermore, we constructed a fully connected synapse graph and figured out its synapse tensor expression. From synapse tensor and a cross-entropy loss function, we found and proved its gradient formula that was the basis for gradient descent learning and using the backpropagation algorithm. In surprisal space, the parameter (weight) updating for learning was the addition of the value of the negative Bose-Einstein distribution. Finally, we designed the program to implement a Multiple Layer Perceptrons (MLP) (Minsky et al. (2017)) for MNIST

*

(LeCun et al. (2010)) and tested it to achieve the near equal accuracy of standard MLP in the same setting.

1.1 Background

Hodgkin and Huxley presented a physiological neuron model that described the electronic potential of the membrane between a neuron and a synapse with a differential equation (Hodgkin & Huxley (1952)). Later, neuron scientists have found that a synapse might have a complicated channel structure with rich chemical and electronic properties (Lodish et al. (1995)(biological synapse), Destexhe et al. (1994)(computing synaptic conductances), Abbott & Nelson (2000)(synaptic plasticity)). Other synapse models based on differential equations had been proposed and been simulated by analogy circuits like Spiking Neural Network (SNN) (Kumar (2017) (differential equations), Lin et al. (2018) (Intel's SNN Loihi)). In these approaches, synapses acted as linear amplifiers with adjustable coefficients. An example was the analog circuit implementation of a Hopfield neural network (Hopfield & Tank (1986)(analog neural circuits)).

In this paper, we proposed a simple synapse model represented by the joint opening probability of excitatory and inhibitory channels in a synapse. It was described as a non-linear computable synapse function. This neuroscience-inspired model was motivated on our unpublished research to solve optimization problems by neural networks.

To do learning by gradient descent and backpropagation algorithm (Goodfellow et al. (2016)(book on deep learning)), because of the differentiable of the synapse function in the synaptic neural network, we could compute Jacobian matrix explicitly and compute the gradient of the cross-entropy loss function over parameters. Therefore, we provided detailed proof of the formula of gradients in Appendix A

In analyzing Jacobian matrix, we found that the derivative of the inhibitory function $log(1-e^{-x})$ was equal to the $1/(e^{x}-1)$ which was the formula of Bose-Einstein distribution (Einstein (1924)(quantum ideal gas)). Usually, in statistical physics and thermodynamics, Bose-Einstein distribution was concluded from the geometric series of the Bose statistics.

A dual space analysis was an efficient scientific method. After successful expressing fully-connected synapse network in a logarithmic matrix, we considered log function and log space. The concept of surprisal, the measurement of surprise from Information Theory (Shannon (1948)), gave us hints. Original surprisal was defined on the random variable, however, it was convenient to consider the probability itself as a variable. So we introduced the surprisal space with a mapping function -log(p). The motivation was to transform any points from probability space to surprisal space and in reverse.

In surprisal space, a synapse function was the addition of an excitatory identity function and an inhibitory function. Although we had figured out the inhibitory function being $-log(1-e^{-x})$, we wanted to know its structure and what class it belonged to.

This was a procedure we rediscovered the way to construct a commutative diagram for synapse inhibitory function Diagram (2.2.3). In 1903, Mathematician Bertrand Russell presented the first commutative diagram in his book (Russell (2009)) before the category theory. You can find a good introduction to applied category theory by (Bradley (2018)). In this paper, we did not require knowing the category theory.

The basic idea was to given two spaces and two points in source space which have corresponding points in target space by a continuous and inverse mapping function from source space to target space, plus, a function that maps start point to the endpoint in the same source space. Our question is to find the function that maps the corresponding start point to the corresponding endpoint in the target space (refer to diagram 2.2.3). There are two paths from the source start point to target endpoint: one is from top-left, go right and down to bottom-right; another is from top-left, go down and right to bottom-right. The solution is to solve the equation that has the same target endpoint.

We found that the synapse inhibitory function $-log(1-e^{-x})$ was also a topologically conjugate function. Therefore, the synaptic neural network has the same dynamical behavior in both probability space and surprisal space. To convince that the synaptic neural network can work for learning and using the back-propagation algorithm, we proved the gradients of loss function by applying basic calculus. In surprisal space, the negative Bose-Einstein distribution applied to the updating of parameters in the learning of synaptic neural network. Finally, we implemented an MNIST experiment of MLP to be the proof-of-concept.

1.2 CONTRIBUTIONS

1) present a neuroscience-inspired synapse model and a synapse function based on the opening probability of channels. 2) defined surprisal space to link information theory to the synaptic neural network. 3) figure out function $\mathcal{G}(x) = -log(1 - e^{-x})$ as the inhibitory part of a synapse. 4) find the derivative of $\mathcal{G}(x)$ to be the formula of negative Bose-Einstein distribution. 5) discover $\mathcal{G}(x)$ to be a topologically conjugate function of the complementary probability. 6) represent fully-connected synapse as a synapse tensor. 7) express synapse learning of gradient descent as a negative Bose-Einstein distribution. 8) prove synapse log-concavity

2 SYNAPTIC NEURAL NETWORK (SYNANN)

A Synaptic Neural Network (SynaNN) contains non-linear synapse networks that connect to neurons. A synapse consists of an input from the excitatory-channel, an input from the inhibitory-channel, and an output channel which sends a value to other synapses or neurons. Synapses may form a graph to receive inputs from neurons and send outputs to other neurons. In advance, many synapse graphs can connect to neurons to construct a neuron graph. In traditional neural network, its synapse graph is simply the wight matrix or tensor.

2.1 SYNAPSE IN PROBABILITY SPACE

2.1.1 BIOLOGICAL SYNAPSE

Changes in neurons and synaptic membranes (i.e. potential gate control channel and chemical gate control channel show selectivity and threshold) explain the interactions between neurons and synapses (Torre & Poggio (1978)). The process of the chemical tokens (neurotransmitters) affecting the control channel of the chemical gate is accomplished by a random process of mixing tokens of the small bulbs on the membrane. Because of the randomness, a probabilistic model makes sense for the computational model of the biological synapse (Hubel & Kandel (1979)).

In a synapse, the Na+ channel illustrates the effect of an excitatory-channel. The Na+ channels allow the Na+ ions flow in the membrane and make the conductivity increase, then produce excitatory post-synapse potential. The K+ channels illustrate the effects of inhibitory channels. The K+ channel that lets the K+ ions flow out of the membrane shows the inhibition. This makes the control channel of potential gate closing and generates inhibitory post-potential of the synapse. Other kinds of channels (i.e. Ca channel) have more complicated effects. Biological experiments show that there are only two types of channels in a synapse while a neuron may have more types of channels on the membrane. Experiments illustrate that while a neuron is firing, it generates a series of the spiking pulse where the spiking rate (frequency) reflects the strength of stimulation.

2.1.2 NEUROSCIENCE-INSPIRED SYNAPSE MODEL

From neuroscience, there are many types of chemical channels in the membrane of a synapse. They have the following properties: 1) the opening properties of ion channels reflect the activity of synapses. 2) the excitatory and inhibitory channels are two key types of ion channels. 3) the random properties of channels release the statistical behavior of synapses.

From the basic properties of synapses, we proposed the synapse model below:

1) The open probability x of the excitatory channel (α-channel) is equal to the number of open excitatory channels divided by the total number of excitatory channels of a synapse. 2) The open probability y of the inhibitory channel (β-channel) is equal to the number of open inhibitory channels divided by the total number of inhibitory channels of a synapse. 3) The joint probability of a synapse that affects the activation of the connected output neuron is the product of the probability of excitatory channel and the complementary probability of the inhibitory channel. 4) There are two parameters to control excitatory channel and inhibitory channel respectively.

Given two random variables (X, Y), their probabilities (x, y), and two parameters (α, β), the joint probability distribution function $S(x, y; \alpha, \beta)$ for X, Y (the joint probability of a synapse that activates the connected neuron) is defined as

$$S(x, y; \alpha, \beta) = \alpha x(1 - \beta y) \tag{1}$$

where $x \in (0, 1)$ is the open probability of all excitatory channels and $\alpha > 0$ is the parameter of the excitatory channels; $y \in (0, 1)$ is the open probability of all inhibitory channels and $\beta \in (0, 1)$ is the parameter of the inhibitory channels. The symbol semicolon ";" separates the variables and parameters in the definition of function S.

2.2 Synapse in surprisal space

Surprisal (self-information) is a measure of the surprise in the unit of bit, nat, or hartley when a random variable is sampled. Surprisal is a fundamental concept of information theory and other basic concepts such as entropy can be represented as the function of surprisal. The concept of surprisal has been successfully used in molecular chemistry and natural language research.

2.2.1 Surprisal space

Given a random variable X with value x, the probability of occurrence of x is p(x). The standard definition of **Surprisal** $\mathcal{I}_p(x)$ is the measure of the surprise in the unit of a bit (base 2), a nat (base e), or a hartley (base 10) when the random variable X is sampled at x. Surprisal is the negative logarithmic probability of x such that $\mathcal{I}_p(x) = -log(p(x))$. Ignored random variable X, we can consider p(x) as a variable in **Probability Range Space** or simply called **Probability Space** in the context of this paper which is the open interval (0,1) of real numbers.

Surprisal Function is defined as $\mathcal{I} : (0, 1) \to (0, \infty)$ and $\mathcal{I}(x) = -log(x)$ where $x \in (0, 1)$ is an open interval in $\mathbb{R}^+$. Its inverse function is $\mathcal{I}^{-1}(u) = e^{-u}$ where $u \in \mathbb{R}^+$. Since surprisal function $\mathcal{I}(x)$ is bijective, exists inverse and is continuous, $\mathcal{I}(x)$ is a **homeomorphism**.

Surprisal Space $\mathbb{S}$ is the mapping space of the **Probability Space** $\mathbb{P}$ with the negative logarithmic function which is a bijective mapping from the open interval $(0, 1)$ of real numbers to the real open interval $(0, \infty) = R^+$.

$$\mathbb{S} = \{s \in (0, \infty) : s = -log(p), for all\ p \in (0, 1)\} \tag{2}$$

The probability space $\mathbb{P}$ and the surprisal space $\mathbb{S}$ are topological spaces of real open interval (0,1) and positive real numbers $\mathbb{R}^+$ that inherit the topology of real line respectively.

2.2.2 Surprisal synapse

Given variables $u, v \in \mathbb{S}$ and parameters $\theta, \gamma \in \mathbb{S}$ which are equal to variables $-log(x), -log(y)$ and parameters $-log(\alpha), -log(\beta)$ respectively. The **Surprisal Synapse** $\mathcal{LS}(u, v; \theta, \gamma) \in \mathbb{S}$ is defined as,

$$\mathcal{LS}(u, v; \theta, \gamma) = -log(\mathcal{S}(x, y; \alpha, \beta)) \tag{3}$$

Expanding the right side, there is $\mathcal{LS}(u, v; \theta, \gamma) = (-log(\alpha x)) + (-log(1 - \beta y))$. The first part is an identity mapping plus a parameter. To understand the second part more, we need to figure out its structure and class.

2.2.3 Topological conjugacy

Theorem 1 (Topologically conjugate function)**.** *Given $y = \mathcal{F}(x)$ where $\mathcal{F}(x) = 1 - x$; $x, y \in \mathbb{P}$, $(u, v) = \mathcal{I}(x, y)$ where $u, v \in \mathbb{S}$, and the homeomorphism $\mathcal{I}(x) = -log(x)$ from $\mathbb{P}$ to $\mathbb{S}$, then function $\mathcal{G}$ such that $v = \mathcal{G}(u)$ is*

$$\mathcal{G}(u) = \mathcal{I} \circ \mathcal{F} \circ \mathcal{I}^{-1}(u) = -log(1 - e^{-u}) \tag{4}$$

Proof. Building a commutative diagram with the homeomorphism $\mathcal{I}(x)$ below,

$$\begin{tikzcd}
x \in \mathbb{P} \arrow[r, "\mathcal{F}"] \arrow[d, "\mathcal{I}"'] & y \in \mathbb{P} \arrow[d, "\mathcal{I}"] \\
u \in \mathbb{S} \arrow[r, "\mathcal{G}"'] & v \in \mathbb{S}
\end{tikzcd}$$

The proof is to figure out the equivalent of two paths from x to v. One path is from top x, go right to y and go down to bottom so $v = \mathcal{I}(\mathcal{F}(x))$. Another path is from top x, go down to u and go right to bottom so $v = \mathcal{G} \circ \mathcal{I}$, thus, $\mathcal{I}(\mathcal{F}(x)) = \mathcal{G}(\mathcal{I}(x))$. Let $\circ$ be the composition of functions, the previous equation is $\mathcal{I} \circ \mathcal{F} = \mathcal{G} \circ \mathcal{I}$. Applying $\mathcal{I}^{-1}$ on both right sides and compute $\mathcal{G}$ on given functions, we proved Eq.(4). □

Given two topological spaces $\mathbb{P}$ and $\mathbb{S}$, continuous function $\mathcal{F} : \mathbb{P} \to \mathbb{P}$ and $\mathcal{G} : \mathbb{S} \to \mathbb{S}$ as well as homeomorphism $\mathcal{I} : \mathbb{P} \to \mathbb{S}$, if $\mathcal{I} \circ \mathcal{F} = \mathcal{G} \circ \mathcal{I}$, then $\mathcal{G}$ is called the topologically conjugated function of the function $\mathcal{F}$ in the standard definition. From Theorem 1, specially $\mathcal{G}(u) = -log(1 - e^{-u})$ is the topologically conjugate function of the complementary probability function $1 - x$. Features:

i) The iterated function $\mathcal{F}$ and its topologically conjugate function $\mathcal{G}$ have the same dynamics. ii) They have the same mapped fixed point where $\mathcal{F} : x = 1/2$ and $\mathcal{G} : u = -log(1/2)$. iii) $\mathcal{I}(x) = -log(x)$ is a infinite differentiable and continuous function in real open interval (0,1).

Let parametric function be $D(u; \theta) = u + \theta$, the surprisal synapse is

$$\mathcal{LS}(u, v; \theta, \gamma) = D(u; \theta) + (\mathcal{I} \circ \mathcal{F} \circ \mathcal{I}^{-1})(D(v; \gamma)) \tag{5}$$

From Eq.(5), the universal function of a surprisal synapse is the addition of the excitatory function and the topologically conjugate inhibitory function in surprisal space. By constructed a commutative diagram, we figured out the elegant structure and topological conjugacy of the function $-log(1-e^{-u})$, which is a new example of the commutative diagram and the topological conjugate function from synaptic neural network. A bridge has been built to connect the synaptic neural network to the category theory, the topology, and the dynamical system.

2.2.4 Bose-Einstein distribution

It is interesting to find the connection between the surprisal synapse and the topologically conjugate function. Furthermore, we are going to figure out the connection between the surprisal synapse and the Bose-Einstein distribution. The Bose-Einstein distribution ($\mathcal{BED}$) is represented as the formula $f(E) = \frac{1}{Ae^{E/kT}-1}$ where f(E) is the probability that a particle has the energy E in temperature T. k is Boltzmann constant, A is the coefficient (Nave (2018)).

Theorem 2. *The $\mathcal{BED}$ function is defined as $\mathcal{BED}(v; \gamma) = \frac{1}{e^{v+\gamma}-1}$ where variable $v \in \mathbb{S}$, parameter $\gamma \in \mathbb{S}$, and $v + \gamma \geq ln(2)$, so that $0 \leq \mathcal{BED}(v; \gamma) \leq 1$, then there is $\frac{\partial}{\partial\gamma}\mathcal{G}(D(v; \gamma)) = -\mathcal{BED}(v; \gamma)$ or*

$$\frac{\partial}{\partial\gamma}(-log(1 - e^{-D(v;\gamma)})) = \frac{-1}{e^{D(v;\gamma)} - 1} \tag{6}$$

Proof. Proved by computing the derivative of the function on left side. □

Recall that $D(v; \gamma) = v+\gamma$, the derivative of the topologically conjugate function $\mathcal{G}$ over parameter γ is equal to the negative Bose-Einstein distribution. The gradient of the surprisal synapse $\mathcal{LS}(u, v; \theta, \gamma)$ is

$$(\frac{\partial}{\partial u}; \frac{\partial}{\partial \theta}, \frac{\partial}{\partial v}; \frac{\partial}{\partial \gamma})\mathcal{LS}(u, v; \theta, \gamma) = (1; 1, -\mathcal{BED}(v; \gamma); -\mathcal{BED}(v, \gamma)) \tag{7}$$

This is a connection between surprisal synapse and statistical physics.

In statistical physics, $\mathcal{BED}(v; \gamma)$ is the probability that boson particles remain in v energy level with the negative chemical potential γ. Chemical potential is a quantity in which two systems in equilibrium can exchange particles. For bosons, the chemical potential of a system must be negative or zero so so the system can attract new particles. Otherwise there is a singularity in the Bose-Einstein distribution. When all the chemical potentials are zeros, for bosons, the system is in the Bose-Einstein Condensation.

2.2.5 Synapse Log-concavity

A non-negative function $f : R^n \rightarrow R$ is log-oncave if its domain is a convex set and it satfisfies the inequality $f(\theta x + (1-\theta)y) \geq f(x)^\theta f(y)^{1-\theta}$ for all $x, y \in dom f$ and $0 < \theta < 1$. That is $log f(\theta x + (1-\theta)y) \geq \theta log(f(x) + (1-\theta)$. A theorem said that if the Hessian matrix of a function $f(\mathbf{x})$ where $x \in R^n$ is negative definite in a convex set, then f is log-concave.

Theorem 3. *(1) In the domain $(x, y) \in (0,1)^2$, Synapse function $S(x, y; \alpha, \beta)$ is not a concave function; (2) Synapse function $S(x, y; \alpha, \beta)$ is a log-concave function; (3) Multivariate inhibited synapse function $S(\mathbf{x}; \beta)$ is log-concave function in the domain $\mathbf{x} \in (0,1)^n$*

Proof. (1) Given $S(x, y; \alpha, \beta) = \alpha x(1-\beta y)$, the Hessian matrix is $H(S) = \begin{bmatrix} 0 & -\alpha\beta \\ -\alpha\beta & 0 \end{bmatrix}$. Since $z^T H(S) z = -\alpha\beta(2ab)$ where z=(a,b), H is indefinite, thus S is not concave.

(2) Since $H(log(S)) = \begin{bmatrix} \frac{-1}{x^2} & 0 \\ 0 & \frac{-\beta^2}{(1-\beta y)^2} \end{bmatrix}$, $z^T H(log(S)) z = -\left(\left(\frac{a}{x}\right)^2 + \left(\frac{b\beta}{1-\beta y}\right)^2\right) \leq 0$ where z=(a,b), H(log(S) is concave and S is log-concave.

(3) Given multivariate inhibited synapse function $S(x_1, ..., x_n; \beta_1, ..., \beta_n) = (1-\beta_1 x_1)...(1-\beta_n x_n)$ where $x_i \in (0,1)$, let $z^T = (a_1, ..., a_n)$

$$z^T H(log(S)) z = -\left(\left(\frac{\beta_1 a_1}{1-\beta_1 x_1}\right)^2 + ... + \left(\frac{\beta_n a_n}{1-\beta_n x_n}\right)^2\right) \leq 0 \tag{8}$$

S is log-concave. All the roots of $S(x_1, ..., x_n; \beta_1, ..., \beta_n) = 0$ are real for all $\beta_i \in (0,1)$. S is stable log-concave. □

Remark 1. *A log-concave function has many good features such as that convolution is going to preserves log-concavity. Many common probability distributions are log-concave such as normal, exponential and uniform. Moreover, the CDF log-concave distribution is also log-concave.*

2.3 Synapse graph and tensor

Generally, a biological neuron consists of a soma, an axon, and dendrites. Synapses are distributed on dendritic trees and the axon connects to other neurons in the longer distance. A ***synapse graph*** is the set of synapses on dendritic trees of a neuron. A synapse can connect its output to an input of a neuron or to an input of another synapse. A synapse has two inputs: one is excitatory input and another is inhibitory input. Typically neurons receive signals via the synapses on dendrites and send out spiking plus to an axon (Hubel & Kandel (1979)).

Assume that the total number of input of the **synapse graph** equals the total number of outputs, the **fully-connected synapse graph** is defined as

$$y_i(\mathbf{x}; \boldsymbol{\beta}_i) = x_i \prod_{j=1}^{n} (1 - \beta_{ij} x_j), \ for \ all \ i \in [1, n] \tag{9}$$

where $\mathbf{x} = (x_1, \cdots, x_n)$, $x_i \in (0,1)$ and $\mathbf{y} = (y_1, \cdots, y_n)$ are row vectors of probability distribution; $\boldsymbol{\beta}_i = (\beta_{i1}, \cdots, \beta_{in})$, $0 < \beta_{ij} < 1$ are row vectors of parameters; $\boldsymbol{\beta} = matrix\{\beta_{ij}\}$ is the matrix of all parameters. $\boldsymbol{\alpha} = 1$ is assigned to Eq.1 to simplify the computing.

An output y_i of the fully-connected synapse graph is constructed by linking the output of a synapse to the excitatory input of another synapse in a chain while the inhibitory input of each synapse is the output of neuron x_i in series. In the case of the diagonal value β_{ii} is zero, there is no self-correlated factor in the *i*th item.

This fully-connected synapse graph represents that only neuron itself acts as excitation all its connected synapses act as inhibition. This follows the observation of neuroscience that most synapses act as inhibition.

Theorem 4 (Synapse tensor formula)**.** *The following **synapse tensor formula** Eq.10 is equivalent to **fully-connected synapse graph** defined in the Eq.9*

$$log(\boldsymbol{y}) = log(\boldsymbol{x}) + \boldsymbol{1}_{|x|} * log(\boldsymbol{1}_{|\beta|} - diag(\boldsymbol{x}) * \boldsymbol{\beta}^T) \tag{10}$$

*or $\mathcal{I}(\boldsymbol{y}) = \mathcal{I}(\boldsymbol{x}) + \boldsymbol{1}_{|x|} * \mathcal{I}(\boldsymbol{1}_{|\beta|} - diag(\boldsymbol{x}) * \boldsymbol{\beta}^T)$ where **x**, **y**, and $\boldsymbol{\beta}$ are distribution vectors and parameter matrix. $\boldsymbol{\beta}^T$ is the transpose of the matrix $\boldsymbol{\beta}$. $\boldsymbol{1}_{|x|}$ is the row vector of all real number ones and $\boldsymbol{1}_{|\beta|}$ is the matrix of all real number ones that have the same size and dimension of **x** and $\boldsymbol{\beta}$ respectively. Moreover, the * is the matrix multiplication, diag(**x**) is the diagonal matrix of the row vector **x**, and the log is the logarithm of the tensor (matrix).*

Proof. Applying the log on both sides of the definition Eq.(9) and completes the matrix multiplications in the ***fully-connected synapse graph***, we proved the formula Eq.(10). Furthermore, by the definition of $\mathcal{I}(x)$, we have the expression of surprisal synapse. □

2.4 Synapse learning

To prove that ***synapse learning*** of synaptic neural network is compatible with the standard backpropagation algorithm, we are going to apply cross-entropy as the loss function and use gradient descent to minimize that loss function.

2.4.1 Gradient of loss function

The basic idea of deep learning is to apply gradient descent optimization algorithm to update the parameters of the deep neural network and achieve a global minimization of the loss function (Goodfellow et al. (2016)).

Theorem 5 (Gradient equation)**.** *Let the loss function $L(\hat{\boldsymbol{o}}, \boldsymbol{o})$ of the **fully-connected synapse graph** Eq.9 be equal to the sum of cross-entropy $L(\hat{\boldsymbol{o}}, \boldsymbol{o}) = -\sum_i \hat{o}_i log(o_i)$, then its parameter gradient is*

$$\frac{\partial L(\hat{\boldsymbol{o}}, \boldsymbol{o})}{\partial \beta_{ij}} = (o_i \tau - \hat{o}_i) \frac{-y_i x_j}{1 - \beta_{ij} x_j} \tag{11}$$

*or $\partial L(\hat{\boldsymbol{o}}, \boldsymbol{o})/\partial \beta_{ij} = (o_i \tau - \hat{o}_i) \partial log(S(y_i, x_j; 1, \beta_{ij}))/\partial \beta_{ij}$ where $\hat{\boldsymbol{o}}$ is the target vector and **o** is the output vector and the **fully-connected synapse graph** outputs through a softmax activation function that is $o_j = softmax(y_j)$.*

Proof. The proof is given in the Appendix A. □

2.4.2 Gradient and Bose-Einstein distribution

Considering the surprisal space, let $(u_k, v_k, \gamma_{ki}) = -log(x_k, y_k, \beta_{ki})$, the **fully-connected synapse graph** is denoted as $v_k = u_k + \sum_i (-log(1 - e^{-(\gamma_{ki} + u_i)}))$. Compute the gradient over parameters

$$\frac{\partial v_k}{\partial \gamma_{pq}} = -\sum_i \frac{e^{-(\gamma_{ki}+u_i)}}{1 - e^{-(\gamma_{ki}+u_i)}} \frac{\partial \gamma_{ki}}{\partial \gamma_{pq}} = -\sum_i \frac{e^{-(\gamma_{ki}+u_i)}}{1 - e^{-(\gamma_{ki}+u_i)}} \delta_{kp} \delta_{iq} \tag{12}$$

because only when $k = p$ and $i = q$, two δ are 1, so $\frac{\partial v_p}{\partial \gamma_{pq}} = \frac{-e^{-(\gamma_{pq}+u_q)}}{1-e^{-(\gamma_{pq}+u_q)}}$. Replacing the indexes and reformulating, we have

$$\frac{\partial v_i}{\partial \gamma_{ij}} = \frac{-1}{e^{\gamma_{ij}+u_j} - 1} \tag{13}$$

The right side of Eq.(13) is the negative Bose-Einstein Distribution in the surprisal space.

To compute the loss function in surprisal space, we convert the target vector $\hat{\boldsymbol{o}}$ and output vector $\boldsymbol{o}$ to surprisal space as $(\hat{\boldsymbol{o}}, \boldsymbol{o})$, so the new loss function is $\mathcal{L}(\hat{\boldsymbol{t}}, \boldsymbol{t}) = \sum_k \hat{t_k} * t_k$. The log function has been removed in $\mathcal{L}(\hat{\boldsymbol{t}}, \boldsymbol{t})$ because log is implied in the surprisal space. Without using an activation function, there is $t_k = v_k$. By Eq.(13),

$$\frac{\partial \mathcal{L}(\hat{\boldsymbol{t}}, \boldsymbol{t})}{\partial \gamma_{ij}} = \sum_k \frac{\partial \mathcal{L}(\hat{\boldsymbol{t}}, \boldsymbol{t})}{\partial v_k} \frac{\partial v_k}{\partial \gamma_{ij}} = \frac{-\hat{t}_i}{e^{\gamma_{ij}+u_j} - 1} \tag{14}$$

We can apply error back-propagation to implement gradient descent for ***synapse learning***.

$$\gamma_{ij} \leftarrow \gamma_{ij} - \eta \frac{\partial \mathcal{L}(\hat{\boldsymbol{t}}, \boldsymbol{t})}{\partial \gamma_{ij}} \tag{15}$$

where η is the learning rate.

Recall the physics explanation, the u_i is the energy level and γ_{ij} is the negative chemical potential, we see that the parameter matrix is the matrix of the negative chemical potential and the synapse learning is to update the chemical potentials to achieve an equilibrium state. The synaptic neural network is not a black box anymore.

After figured out the role of negative chemical potential in the synapse of the synaptic neural network, we believe that a **synapse** is a quantum processing unit (QPU) and biological neural network is a kind of quantum neural network worked in regular temperature.

The equation Eq.(15) illustrates that the learning of synaptic neural network follows the Bose-Einstein statistics in the surprisal space. This paper “Memory as an equilibrium Bose gas” by (Fröhlich (1968), Pascual-Leone (1970)) showed that memory maybe possible to be represented as the equilibrium of Bose gas.

3 Experiments

3.1 SynaMLP : Synaptic Multiple Layer Perceptrons

We are going to illustrate a Synaptic Neural Network implementation SynaMLP with the connection of Multiple Layer Perceptrons (MLP) (Minsky et al. (2017)).

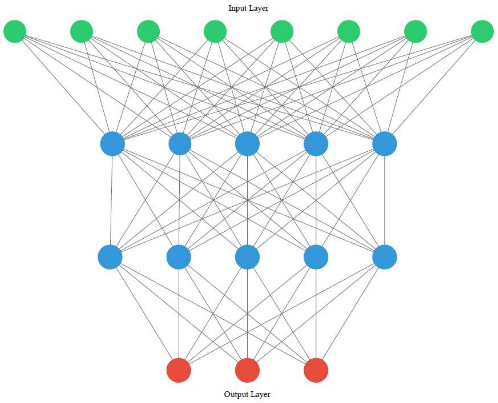

Figure 1: SynaMLP: (green, blue, red) dots are (input, hidden, output) layers.

SynaMLP has an input layer for down-sampling and an output layer for classification. The hidden layer is a block of fully-connected synapse tensor. The inputs of the block are neurons from the input layer and the outputs of the block are neurons to the output layer. The block is the implementation of synapse tensor in Eq.(10). The activation functions are connected ***synapse tensor*** to the output layer. Moreover, the input of the block is a probability distribution. The block can be thought of the replacement of the weight layer in a standard neural network.

3.2 MNIST Experiment

To proof-of-concept, we implemented the SynaMLP with MNIST. Hand-written digital MNIST data sets are used for training and testing in machine learning. It is split into three parts: 60,000 data points of training data (mnist.train), 10,000 points of test data (mnist.test), and 5,000 points of validation data (mnist.validation) (LeCun et al. (2010)).

	Synapse MLP	**Dense MLP**
Iteration	10001	10001
Test loss	0.0942904587357543	0.0906127168575471
Test accuracy	**0.9802**000087499618	**0.9830**000066757202

Table 1: SynaMLP MNIST Testing

The MNIST SynaMLP training and testing is implemented by Python, Keras and Tensorflow (Abadi et al. (2016)) from the revision of the example of mnist_mlp.py in Keras distribution. The synapse tensor is designed to be a class to replace Dense in Keras. The layer sequence is as below,

Layer Type	**Output Shape**	**Param #**
Dense[1]	(None, 300)	235500
Batch-normalization[1]	(None, 300)	1200
Activation[1]	(None, 300)	0
Synapse[1]	(None, 300)	90000
Batch-normalization[2]	(None, 300)	1200
Dense[2]	(None, 10)	3010
Optimizer	Adam	default

Table 2: SynaMLP Layers

In the comparison experiment, SynaNN MLP and traditional MLP generated the similar test accuracy of around 98%. We applied a softmax activation function in front of the input of synapse to avoid the error of NAN (computing value out of the domain). In fact, synaptic neural network handles a probability distribution (vector from neurons).

4 Conclusion

In this paper, we presented and analyzed a ***Synaptic Neural Network (SynaNN)***. We found the fine structure of synapse and the construction of synapse network as well as the BE distribution in the gradient descent learning. In surprisal space, the input of a neuron is the addition of the identity function and the sum of topologically conjugate functions of inhibitory synapses which is the sum of bits of information. The formula of surprisal synapse function is defined as

$$\mathcal{LS}(u, v; \theta, \gamma) = (\theta + u) + (\mathcal{I} \circ \mathcal{F} \circ \mathcal{I}^{-1})(\gamma + v)) \tag{16}$$

The non-linear synaptic neural network may be implemented by physical or chemical components. Instead of using a simple linear synapse function, more synapse functions maybe found in the researches and applications of neural network.

The synapse log-concavity allows us to explore wider fields with Synaptic Neural Network such as applications in FinTech which had not been found killer apps of neural network for many years. Synapse log-concavity provides a tool to understand and explain the behaviors of neural network.

REFERENCES

Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: A system for large-scale machine learning. In *OSDI*, volume 16, pp. 265–283, 2016.

Larry F Abbott and Sacha B Nelson. Synaptic plasticity: taming the beast. *Nature neuroscience*, 3 (11s):1178, 2000.

RB Bernstein and RD Levine. Entropy and chemical change. i. characterization of product (and reactant) energy distributions in reactive molecular collisions: information and entropy deficiency. *The Journal of Chemical Physics*, 57(1):434–449, 1972.

Tai-Danae Bradley. What is applied category theory? *arXiv preprint arXiv:1809.05923*, 2018.

Alain Destexhe, Zachary F Mainen, and Terrence J Sejnowski. An efficient method for computing synaptic conductances based on a kinetic model of receptor binding. *Neural computation*, 6(1): 14–18, 1994.

Albert Einstein. Quantum theory of a monoatomic ideal gas, 1924. URL `http://www.thphys.uni-heidelberg.de/~amendola/otherstuff/einstein-paper-v2.pdf`.

Willliam Feller. *An introduction to probability theory and its applications*, volume 2. John Wiley & Sons, 2008.

Herbert Fröhlich. Long-range coherence and energy storage in biological systems. *International Journal of Quantum Chemistry*, 2(5):641–649, 1968.

Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*, volume 1. MIT press Cambridge, 2016.

Alan L Hodgkin and Andrew F Huxley. A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of physiology*, 117(4):500–544, 1952.

John J Hopfield and David W Tank. Computing with neural circuits: A model. *Science*, 233(4764): 625–633, 1986.

David H. Hubel and Eric R. Kandel. The brain. *Scientific American*, 241(3):45–53, 1979.

Douglas Samuel Jones. *Elementary information theory*. Clarendon Press, 1979.

Arvind Kumar. Simple neuron and synapse models, 2017. URL `https://www.kth.se/social/files/570b469bf276542ebe37fe9f/SimpleNeuronModels_2016_full.pdf`.

Yann LeCun, Corinna Cortes, and CJ Burges. Mnist handwritten digit database. *AT&T Labs*, 2, 2010.

Raphael D Levine. *Molecular reaction dynamics*. Cambridge University Press, 2009.

Roger Levy. Expectation-based syntactic comprehension. *Cognition*, 106(3):1126–1177, 2008.

Chang Li. A nonlinear synaptic neural network based on excitation and inhibition, 2017. URL `https://www.researchgate.net/publication/320557823_A_Non-linear_Synaptic_Neural_Network_Based_on_Excitation_and_Inhibition`.

Chit-Kwan Lin, Andreas Wild, Gautham N Chinya, Yongqiang Cao, Mike Davies, Daniel M Lavery, and Hong Wang. Programming spiking neural networks on intel's loihi. *Computer*, 51(3): 52–61, 2018. URL `https://www.researchgate.net/publication/323434400_Programming_Spiking_Neural_Networks_on_Intel_Loihi`.

Harvey Lodish, Arnold Berk, S Lawrence Zipursky, Paul Matsudaira, David Baltimore, James Darnell, et al. *Molecular cell biology*, volume 3. WH Freeman New York, 1995.

Marvin Minsky, Seymour A Papert, and Léon Bottou. *Perceptrons: An introduction to computational geometry*. MIT press, 2017.

Daisuke Miyashita, Edward H Lee, and Boris Murmann. Convolutional neural networks using logarithmic data representation. *arXiv preprint arXiv:1603.01025*, 2016.

Carl R. Nave. The bose-einstein distribution, 2018. URL `http://hyperphysics.phy-astr.gsu.edu/hbase/quantum/disbe.html`.

Juan Pascual-Leone. A mathematical model for the transition rule in piaget's developmental stages. *Acta psychologica*, 32:301–345, 1970.

Bertrand Russell. *Principles of mathematics*. Routledge, 2009.

Claude Elwood Shannon. A mathematical theory of communication. *Bell system technical journal*, 27(3):379–423, 1948.

V Torre and T Poggio. A synaptic mechanism possibly underlying directional selectivity to motion. *Proc. R. Soc. Lond. B*, 202(1148):409–416, 1978.

Cynthia Vinzant, Nima Anari, KuiKui Liu, and Shayan Oveis Gharan. Completely log-concave polynomials and matroids, 2018. URL `https://clvinzan.math.ncsu.edu/slides/CLC_Matroid_JMM.pdf`.

APPENDIX A GRADIENT EQUATION THEOREM

Let the loss function $L(\hat{\boldsymbol{o}}, \boldsymbol{o})$ of the ***fully-connected synapse graph*** Eq.9 be equal to the sum of cross-entropy $L(\hat{\boldsymbol{o}}, \boldsymbol{o}) = -\sum_i \hat{o}_i log(o_i)$, then its item of parameter gradient is

$$\frac{\partial L(\hat{\boldsymbol{o}}, \boldsymbol{o})}{\partial \beta_{ij}} = (o_i \tau - \hat{o}_i) \frac{-y_i x_j}{1 - \beta_{ij} x_j} \tag{17}$$

or

$$\frac{\partial L(\hat{\boldsymbol{o}}, \boldsymbol{o})}{\partial \beta_{ij}} = (o_i \tau - \hat{o}_i) \frac{\partial log(S(y_i, x_j; 1, \beta_{ij}))}{\partial \beta_{ij}} \tag{18}$$

where $\hat{\boldsymbol{o}}$ is the target vector and $\boldsymbol{o}$ is the output vector and the ***fully-connected synapse graph*** outputs through a softmax activation function.

Proof. Given $o_j = softmax(y_j)$, then

$$\frac{\partial log(o_j)}{\partial y_k} = \delta_{jk} - o_k, \ \frac{\partial L(\hat{\boldsymbol{o}}, \boldsymbol{o})}{\partial y_k} = o_k \tau - \hat{o_k} \tag{19}$$

where δ_{jk} is the Kronecker delta and $\tau = \sum_j \hat{o_j}$ is a constant. After applying log on both sides of Eq.9, we can compute the partial derivative over parameters as,

$$\frac{\partial log(y_k)}{\partial \beta_{pq}} = \sum_i \frac{-x_i}{1 - \beta_{ki} x_i} \frac{\partial \beta_{ki}}{\partial \beta_{pq}} = \sum_i \frac{-x_i}{1 - \beta_{ki} x_i} \delta_{kp} \delta_{iq} = \frac{-x_q}{1 - \beta_{kq} x_q} \delta_{kp} \tag{20}$$

then applying log derivative on the left side and multiple y_k on both sides, we have

$$\frac{\partial y_k}{\partial \beta_{pq}} = \frac{-x_q y_k}{1 - \beta_{kq} x_q} \delta_{kp} \tag{21}$$

Since

$$\frac{\partial L(\hat{\boldsymbol{o}}, \boldsymbol{o})}{\partial \beta_{pq}} = \sum_k \frac{\partial L(\hat{\boldsymbol{o}}, \boldsymbol{o})}{\partial y_k} \frac{\partial y_k}{\partial \beta_{pq}} \tag{22}$$

from Eq.19 and Eq.21, we have

$$\frac{\partial L(\hat{\boldsymbol{o}}, \boldsymbol{o})}{\partial \beta_{pq}} = \sum_k (o_k \tau - \hat{o_k}) \frac{-x_q y_k}{1 - \beta_{kq} x_q} \delta_{kp} = (o_p \tau - \hat{o_p}) \frac{-x_q y_p}{1 - \beta_{pq} x_q} \tag{23}$$

replace index p, q by i, j in Eq.23, we proved the gradient formula Eq.11 which is Eq.5 as well. □