A SYNAPTIC NEURAL NETWORK AND SYNAPSE LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

A Synaptic Neural Network (SynaNN) consists of synapses and neurons. Inspired by the research of biological synapses, we abstracted a synapse as a nonlinear function of two excitatory and inhibitory probabilities. After introducing the surprisal space, we discovered that the surprisal synapse is the sum of the surprisal of the excitatory probability and the topologically conjugate surprisal of the inhibitory probability. From the formula $\frac{\partial}{\partial \gamma} log(1 - e^{-\gamma}) = \frac{1}{e^{\gamma}-1}$, we concluded that the derivative of the surprisal synapse over the parameter is equal to the negative Bose-Einstein distribution. In addition, we could construct a synapse graph such as a fully connected synapse graph (synapse tensor), and embedded it into other neural networks to form a hybrid neural network. Furthermore, we proved the gradient expression of the cross-entropy loss function over parameters, so the synapse learning was compatible with the gradient descent and backpropagation of deep learning. To prove the concept, we performed the MNIST experiment with synaptic neural network and achieved a high accuracy in the model training and inference.

1 INTRODUCTION

Synapses play an important role in biological neural networks (Hubel & Kandel (1979)). Along with the synapses, neurons may form neural networks with complex topologies. Based on the analysis of synaptic excitatory and inhibitory channels (Hubel & Kandel (1979)), we proposed a probabilistic model (Feller (2008)) of the synapse which is a non-linear function of excitatory and inhibitory probabilities (Li (2017)(synapse function)). Equivalently, we started to define the surprisal space (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)), then we represented the synapse function as the addition of the surprisal excitatory probability and the topological conjugation of the surprisal inhibitory probability. In the surprisal space, we discovered that the gradient of the synapse function over its parameter is equal to the negative Bose-Einstein distribution (Nave (2018)).

Expressing fully connected synapse graph as a synapse tensor, we can embed synapse tensors as blocks to construct a large scale neural network similar to RestNet (He et al. (2016)). Moreover, we discussed the swap equation, a special feature of the synapse function, where the swap ratio and odds ratio are connected. The relationship between swap and odds may have potential applications on risk and reward analysis of swap currency ratio .etc in FinTech business. In order to use the gradient descent algorithm in synapse learning, we constructed the cross-entropy loss function for fully-connected synapse graph and proved the gradient equations. The parameters of learning matrix were updated by a factor of Bose-Einstein distribution with a learning rate. Finally, through MNIST (LeCun et al. (2010), Gorner (2017)) training, we verified the correctness and high accuracy of synaptic multiple layer perceptrons (Minsky et al. (2017)).

Inspired by the research of biological neurons and biological chemistry, we proposed a synapse probability model and studied the model in both the probability and surprisal spaces from statistics and information theory. Topologically conjugate function and quantum Bose-Einstein statistics are found in the synaptic neural network. In the meantime, synapse tensor allows us to apply gradient descent and backpropagation algorithms on the synaptic neural network for deep learning.

Because the topologically conjugate function can be implemented by addition and shift operations, in surprise space, we can implemented synaptic neural network without multiplication. That will bring a degree speed up and greatly reduce the use of elements. By the way, new physical and chemistry synapse components could help to implement synaptic neural network with ultra-low-power consumption.

2 SYNAPTIC NEURAL NETWORK (SYNANN)

Suppose a synaptic neural network contains neurons in which they are connected through non-linear synapses. 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 accept inputs from other neurons and output to a neuron. In advance, many synapse graphs can connect to neurons to construct a neuron graph. We are going to discuss the Synaptic Neural Network (SynaNN) from the aspects of synapse definition and features, synapse in surprisal space, synapse graph and tensor, synapse swap and odds, and synapse multiple layer perceptron.

2.1 SYNAPSE

Definition 2.1. *Given two random variables X, Y and their probability density/mass functions x, y and two parameters α and β, the joint probability distribution function $S(x, y; \alpha, \beta)$ for X, Y (the contribution 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 $\alpha \geq 0$ is the parameter of the excitatory channels; $y \in (0, 1)$ is the open probability of all inhibitory channels and $\beta \geq 0$ is the parameter of the inhibitory channels.

Considering the synaptic membrane connected to a neuron, there are many channels in the membrane so that ions can flow in and out. Have

(i) Two types of channels on the membrane: one is called the excitatory channel (α-channel), whose opening increases the activity of neurons; the other is called the inhibitory channel (β-channel), whoses opening suppresses the neuronal activity. In addition, the opening and closing of channels shows the characteristics of randomization.

(ii) Hypothetical variables of neurons (i.e. conductivity) are related to the number of open channels. The more the number of open excitatory channels, the more likely the variable is to take a larger value and the more open inhibitory channels, the more likely the variable is to take a smaller value.

The figure below shows the synapse probability model, where the largest circle represents the membrane; the small circle with a $\oplus$ symbol represents a α-channel where $ions$ inflow the membrane; the small circle with a $\ominus$ symbol represents a β-channel where $ions$ outflow the membrane. The open probability x of the excitatory α-channel is equal to the number of open excitatory channels divided by the total number of excitatory channels; the open probability y of the inhibitory β-channel is equal to the number of open inhibitory channels divided by the total number of inhibitory channels. Eq.(1) indicates the probability of a synapse's contribution to neuronal activation. Changes in neurons and synaptic membranes (i.e. potential gate control channnel and chemical gate control channel show selectivity and threshold) explain the interactions between neurons and synapses (Torre & Poggio (1978)).

The process of the chemical material affecting the control channel of the chemical gate is accomplished by a random process of mixing tokens of the small bulbs and membrane. Since the opening and closing of many channels shows random features, it does make sense for our probabilistic model as a simulated biological synapse (Hubel & Kandel (1979)), (Marr (1982)).

The biological research has demonstrated that the structure and the change of the membrane protein is the key of the neuron activity. The change of the channel protein on neuron and

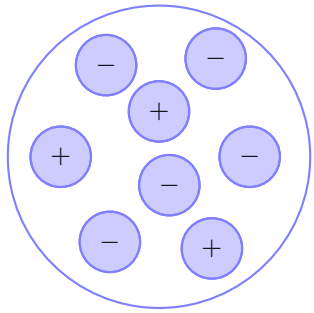

Figure 1: Synapse probability model

synaptic membrane (i.e. potential gate control channel and chemical gate control channel shows the selectivity and threshold) can explain the interaction among neurons and synapses (Hubel & Kandel (1979)).

The Na+ channel illustrates the effect of an excitatory-channel. The Na+ channel allows the Na+ ions flow in membrane and make the conductivity increase, then produce excitatory post-synapse potential. The K+ channel illustrates the effect of an inhibitory channel. 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 kind of channels (i.e. Ca channel) have more complicated effects. Biological experiments showed that there were only two types of channels in a synapse while a neuron may have been related to more types of channels on the membrane. Experiments illustrated that while a neuron is firing, it will generate a series of spiking where the spiking rate (frequency) reflects the strength of stimulation.

In statistical physics, the relation between probability distribution P_r and the energy E_r is $P_r = e^{-bE_r}$ where the e^{-bE_r} is called Boltzmann factor. Replacing x and y in Eq.[1] by the probability distribution, there is the formula $\alpha e^{-au}(1 - \beta e^{-bv})$. Therefore, a synaptic neural network can be considered as a dynamic probability network.

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 the molecular chemistry and the natural language processing.

2.2.1 Surprisal space

Definition 2.2. *Given two random variables X and Y with values x and y, the probabilities of occurrence of x and y are p(x) and q(y) respectively.*

Surprisal *$\mathcal{I}_p(x)$ is the measure of the surprise in the unit of bit (base 2), nat (base e), or hartley (base 10) when the random variable X is sampled at x. It is the negative logarithmic probability of x, that is $\mathcal{I}_p(x) = -log(p(x))$.* ***Information entropy*** *of a random variable X is defined as $\mathcal{H}(X) = -\sum_x p(x)log(p(x)) = \sum_x p(x)\mathcal{I}_p(x)$. It is the expected or average surprisal across over the probability distribution;* ***Cross entropy*** *of two random variables X and Y is defined as $\mathcal{H}(X,Y) = -\sum_{x,y} p(x)log(q(y)) = \sum_{x,y} p(x)\mathcal{I}_q(y)$;* ***Surprisal function*** *is defined as the $\mathcal{I}(x) = -log(x)$ where $x \in (0,1)$. Its inverse function is $\mathcal{I}^{-1}(u) = exp(-u)$ where $u \in \mathbb{R}^+$.*

Since $\mathcal{I}(x)$ is bijective, exists inverse and is continuous, $\mathcal{I}(x)$ is homeomorphism. Actually $\mathcal{I}(x)$ is ∞ continuously differentiable, $\mathcal{I}(x)$ is a ∞-diffeomorphism from the manifold of probability value space to the manifold of surprisal space.

Definition 2.3. ***Surprisal space*** $\mathbb{S}$ *is the mapping space of the probability value space* $\mathbb{P}$ *with the negative logarithmic function which is a bijective map from the space* $\mathbb{P}$ *in open set interval* $(0,1)$ *to the surprisal space* $\mathbb{S}$ *in open set interval* $(0,\infty) = R^+$.

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

As an open set interval is a manifold, we can apply the differential topology to the surprisal space.

2.2.2 Surprisal synapse

Definition 2.4. *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,*

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

Let $\circ$ be the composition of functions, $\mathcal{F}(x) = 1 - x$, parametric function $D(u;\theta) = u + \theta$, we have

$$\begin{aligned}
\mathcal{LS}(u, v; \theta, \gamma) &= (\theta + u) + \mathcal{I}(1 - e^{-(\gamma+v)}) && (4)\\
&= (\theta + u) + \mathcal{I}(1 - \mathcal{I}^{-1}(\gamma + v)) && (5)\\
&= (\theta + u) + (\mathcal{I} \circ \mathcal{F} \circ \mathcal{I}^{-1})(\gamma + v)) && (6)\\
&= D(u;\theta) + (\mathcal{I} \circ \mathcal{F} \circ \mathcal{I}^{-1})(D(v;\gamma)) && (7)
\end{aligned}$$

Let $\mathcal{G}(u)$ be the topological conjugation of $\mathcal{F}(x)$, it is defined as

$$\mathcal{G}(u) = \mathcal{I} \circ \mathcal{F} \circ \mathcal{I}^{-1}(u) = -log(1 - e^{-u}) \tag{8}$$

i) The iterated function $\mathcal{F}$ and its topologically conjugate $\mathcal{G}$ 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) In case $\theta = 0$ the D(u;0) is an Identity function. iv) -log(x) is a function that preserves convexity.

It is clear to express the topological conjugation of surprisal synapse as a commutative diagram in category theory and there is $\mathcal{I}(\mathcal{F}(x)) = \mathcal{G}(\mathcal{I}(x))$.

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

In summary, we have the surprisal synapse formula

$$\begin{array}{c}\textbf{Surprisal}\\ \textbf{Synapse}\end{array} = \begin{array}{c}\textbf{Surprisal}\\ \textbf{Excitatory X}\end{array} + \begin{array}{c}\textbf{Conjugate Surprisal}\\ \textbf{Inhibitory Y}\end{array}$$

This is a general expression of surprisal synapse in the surprisal space. A synapse acts as an addition function of the excitation and the conjugate inhibition in surprisal space. It is interesting to find that the synapse function is related to the pair $\mathcal{F}(x) = 1 - x$ and its topologically conjugate $\mathcal{G}(u) = -log(1-e^{-u})$ under the surprisal function $\mathcal{I}(x) = -log(x)$. Note $\mathcal{G}(u) = -ln(1-e^{-u}) = Li_1(e^{-u})$ that is a special function of the polylogarithm $Li_1(z) = \sum_{n=1}^{\infty} \frac{z^n}{n}$. In quantum statistics, the polylogarithmic function acts as the closed form of integrals of both the Fermi–Dirac and Bose–Einstein distributions (Lee (2009)).

2.2.3 Topological conjugation and Bose-Einstein distribution

It is interesting to find the connection between the surprisal synapse especial of the topologically conjugate surprisal and the Bose-Einstein distribution. The Bose-Einstein distribution ($\mathcal{BED}$) is represented as (Nave (2018)),

$$\mathcal{BED}(v;\gamma) = \frac{1}{e^{\gamma+v} - 1} \tag{9}$$

where variable $v \in \mathbb{S}$ and parameter $\gamma \in \mathbb{S}$ all are positive real numbers. So $\mathcal{BED}(v;\gamma)$ is also a positive real number. From the formula below

$$\frac{\partial}{\partial \gamma} log(1 - e^{-(\gamma+v)}) = \frac{1}{e^{\gamma+v} - 1} \tag{10}$$

the gradient of $\mathcal{LS}(u, v; \theta, \gamma)$ is

$$(\frac{\partial}{\partial u}, \frac{\partial}{\partial v}, \frac{\partial}{\partial \theta}, \frac{\partial}{\partial \gamma})\mathcal{LS}(u, v; \theta, \gamma) = (1, -\mathcal{BED}(v;\gamma), 1, -\mathcal{BED}(v, \gamma)) \tag{11}$$

From Eq.(8) we know that the derivative of the topologically conjugate $\mathcal{G}$ in surprisal space is the negative Bose-Einstein distribution. This builds the connection between surprisal synapse and $\mathcal{BED}$. In physics, $\mathcal{BED}(v;\gamma)$ can be thought of as the probability that boson particles remain in v energy level with initial value γ.

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 longer distance. A ***synapse graph*** is the set of synapses on dendritic trees of a neuron. Typically neurons receive signals via the synapses on dendrites and send out spiking plus to an axon (Hubel & Kandel (1979)).

Definition 2.5. *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*

$$y_i(\boldsymbol{x}; \boldsymbol{\beta}_i) = x_i \prod_{j=1}^{n} (1 - \beta_{ij} x_j), \ for\ all\ i \in [1, n] \tag{12}$$

where $\boldsymbol{x} = (x_1, \cdots, x_n)$, $x_i \in (0., 1.)$ and $\boldsymbol{y} = (y_1, \cdots, y_n)$ are row vectors of probability distribution; $\boldsymbol{\beta}_i = (\beta_{i1}, \cdots, \beta_{in})$, $\beta_{ij} \geq 0$. 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.

In the case of the diagonal value β_{ii} is zero, there is no self-correlated factor in the *i*th item.

Theorem 1 (Synapse tensor equation)**.** *The following **synapse tensor equation** Eq.13 is equivalent to **fully-connected synapse graph** defined in the Eq.12*

$$log(\boldsymbol{y}) = log(\boldsymbol{x}) + \boldsymbol{1}_{|x|} * log(\boldsymbol{1}_{|\beta|} - diag(\boldsymbol{x}) * \boldsymbol{\beta}^T) \tag{13}$$

or

$$\mathcal{I}(\boldsymbol{y}) = \mathcal{I}(\boldsymbol{x}) + \boldsymbol{1}_{|x|} * \mathcal{I}(\boldsymbol{1}_{|\beta|} - diag(\boldsymbol{x}) * \boldsymbol{\beta}^T) \tag{14}$$

*where **x**, **y**, and $\boldsymbol{\beta}$ are distribution vectors and parameter matrix defined in 2.5. $\boldsymbol{\beta}^T$ 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 log is the logarithm of the tensor (matrix).*

Proof. applies the log on both sides of the equation Eq.(12) and completes the matrix multiplications in ***fully-connected synapse graph*** in the equation Eq.12, we can prove the equation Eq.(13). By the definition of I(x), we can also prove the equation Eq.(14). □

Compare ***synapse tensor*** to the block of Residual Network (He et al. (2016)), we found that the **synapse tensor** is similar to the block of Residual Neural Network (ResNet) in the surprisal space,

$$log(\mathbf{y}) = log(\mathbf{x}) + \mathcal{H}(\mathbf{x}; \boldsymbol{\beta}) \tag{15}$$

where

$$\mathcal{H}(\mathbf{x}; \boldsymbol{\beta}) = \mathbf{1}_{|x|} * log(\mathbf{1}_{|\beta|} - diag(\mathbf{x}) * \boldsymbol{\beta}^T) \tag{16}$$

The great benefit of using **synapse tensor** is that we can construct ultra deep learning networks without overfitting. This is very useful feature for computer vision application in object detection and segmentation.

2.4 SynaMLP : Synaptic Multiple Layer Perceptrons

SynaMPL illustrates the application of Synaptic Neural Network to Multiple Layer Perceptrons.

Definition 2.6. *Synaptic Multiple Layer Perceptrons is the fully-connected Multiple Layer Perceptron (Minsky et al. (2017)) with two hidden layers connected by a* ***synapse tensor*** *along with an input layer and an output layer. Input and output layers act as downsize and classifiers while hidden layer of the synapse tensor plays the role of distribution transformation.*

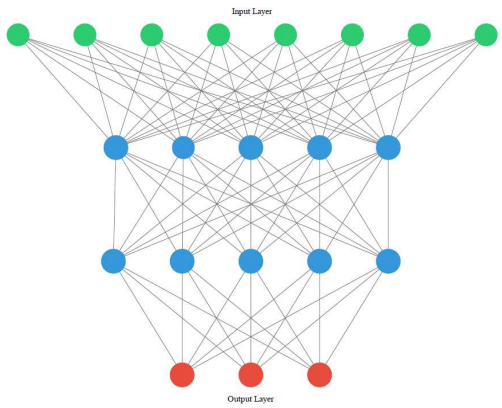

Figure 2: SynaMLP: (green, blue, red) dots are (input, hidden, output) layers.

The activation functions are required to connected ***synapse tensor*** to input and output layers.

3 Synapse Learning

We are going to prove that ***synapse learning*** is compatible to the standard back-propagation algorithm where the cross-entropy is applied for the loss function and the gradient descent is going to be used to minimize the loss function.

3.1 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 2 (Gradient Equation)**.** *Let the loss function $L(\hat{\boldsymbol{o}}, \boldsymbol{o})$ of the* ***fully-connected synapse graph*** *Eq.12 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.12, 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 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 equation Eq.17 which is Eq.18 as well. □

3.2 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

$$v_k = u_k + \sum_i (-log(1 - e^{-(\gamma_{ki}+u_i)})) \tag{24}$$

Compute the gradient over parameters

$$\frac{\partial v_k}{\partial \gamma_{pq}} = -\sum_i \frac{e^{-(\gamma_{ki}+u_i)}}{1-e^{-(\gamma_{ki}+u_i)}}\frac{\partial \gamma_{ki}}{\partial \gamma_{pq}} = -\sum_i \frac{e^{-(\gamma_{ki}+u_i)}}{1-e^{-(\gamma_{ki}+u_i)}}\delta_{kp}\delta_{iq} \tag{25}$$

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

$$\frac{\partial v_i}{\partial \gamma_{ij}} = \frac{-1}{e^{\gamma_{ij}+u_j} - 1} \tag{26}$$

The right side of Eq.(26) 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. We may not use activation functions since synapses can complete non-linear computing. Thus $t_k = v_k$. By Eq.(26),

$$\frac{\partial \mathcal{L}(\hat{\boldsymbol{t}}, \boldsymbol{t})}{\partial \gamma_{ij}} = \sum_k \frac{\partial \mathcal{L}(\hat{\boldsymbol{t}}, \boldsymbol{t})}{\partial v_k}\frac{\partial v_k}{\partial \gamma_{ij}} = \frac{-\hat{t_i}}{e^{\gamma_{ij}+u_j} - 1} \tag{27}$$

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

$$\gamma_{ij} \leftarrow \gamma_{ij} - \eta\frac{\partial \mathcal{L}(\hat{\boldsymbol{t}}, \boldsymbol{t})}{\partial \gamma_{ij}} \tag{28}$$

where η is the learning rate.

The equation Eq.(28) illustrates that the learning of synaptic neural network follows the Bose-Einstein statistics in the surprisal space. The synapse tensor is in the BE distribution. "Memory as an equilibrium Bose gas" by (Fröhlich (1968), Pascual-Leone (1970)) shows a possible advanced research on memory of synaptic neural network.

4 EXPERIMENTS

To proof of concept, we have implemented a synaptic neural network for MNIST. Hand-written digital MNIST datasets are widely used for training and testing in machine learning. It is split into three parts: 55,000 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)).

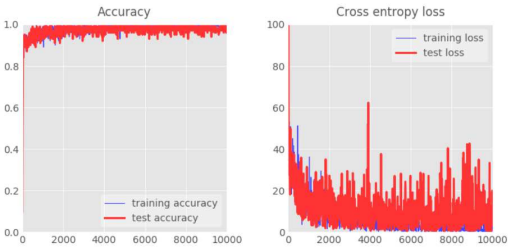

Figure 3: SynaMLP Accuracy and Loss

Iteration	10001
Train accuracy	0.99000
Test accuracy	0.97950
Train loss	2.70175
Test loss	7.62575
Learning ratio	0.00012

Table 1: SynaMLP Testing

The MNIST SynaMLP training and testing is implemented by Python and Tensorflow (Abadi et al. (2016)) from the samples in Tensorflow MNIST Tutorial (Gorner (2017)). Hyperparameters and parameters are setting to: the number of input neurons is 784; the number of output neurons is 10; the number of hidden layer neurons is 280; the batch size is 100; all 3 weight matrices are initialized with a normal distribution of $stddev = 0.1$; the learning optimizer is Adam and learning decay speed 2000.0, max learning rate 0.003, and min learning rate 0.0001.

We use classical neural network at the input and output layers, but use a synapse tensor in the hidden layer. From the input layer to hidden layer, the data is required to be normalized to meet the probability distribution requirements. After using dropout behind ***synapse tensor*** with the keep probability set to 0.75, the overfitting problem has been greatly improved (Srivastava et al. (2014)), and the test accuracy has increased to near 98 percent.

5 CONCLUSION

We built and analyzed a ***Synaptic Neural Netowk (SynaNN)*** and ***Synapse Learning*** in this article. i) SynaNN transforms data processing into probability distribution processing, converts the weights into parameters, and calculates the loss function as cross-entropy. ii) SynaNN ***synapse learning*** updates network parameters with Bose-Einstein statistics. iii) Using a large number of synapses and neurons SynaNN can solve the complex problems in the real world.

Based on the research of surprisal space and the discovery of the topological conjugation, we can implement synaptic neural network with the addition and the conjugate function:

$$\mathcal{LS}(u, v; \theta, \gamma) = (\theta + u) + (\mathcal{I} \circ \mathcal{F} \circ \mathcal{I}^{-1})(\gamma + v)) \tag{29}$$

It is possible to implement synaptic neural network with new type of optical and superconductor components such as Bose-Einstein condensate (BEC) devices (Byrnes et al. (2013)).

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.

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.

Tim Byrnes, Shinsuke Koyama, Kai Yan, and Yoshihisa Yamamoto. Neural networks using two-component bose-einstein condensates. *Scientific reports*, 3:2531, 2013.

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.

M. Gorner. Tensorflow mnist tutorial, 2017. URL `https://github.com/martin-gorner/tensorflow-mnist-tutorial`.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

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.

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

M Howard Lee. Polylogarithms and logarithmic diversion in statistical mechanics. *Acta Physica Polonica B*, 40(5), 2009.

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`.

David Marr. *Vision: A Computational Investigation Into*. WH Freeman, 1982.

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.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

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