One-to-one Mapping between Stimulus and Neural State: 
Memory and Classification

Sizhong Lan\textsuperscript{1}
Fuyun Research, Beijing, 100000, China\textsuperscript{2}

(Dated: 4 December 2018)

Synaptic strength can be seen as probability to propagate impulse, and function could exist from propagation activity to synaptic strength. If the function satisfies constraints such as continuity and monotonicity, neural network under external stimulus will always go to fixed point, and there could be one-to-one mapping between external stimulus and synaptic strength at fixed point. In other words, neural network "memorizes" external stimulus in its synapses. A biological classifier is proposed to utilize this mapping.

I. INTRODUCTION

It is still unanswered how memory works. Specifically, exposed to external stimulus, how can neural network memorize the information of stimulus for a long time if not permanently? Exposed to different stimulus, how can neural network attain different memory? Ideally, memory should be a one-to-one mapping between external stimulus and long-lasting state of neural network. Known experiment results show that "neurons that fire together wire together\textsuperscript{1,2}" synaptic connection strengthens or weakens over time in response to increases or decreases in impulse propagation\textsuperscript{3}. This biochemical mechanism, called synaptic plasticity\textsuperscript{4,5}, establishes a relation between synaptic strength and impulse propagation frequency. We find out that, with some "lightweight" constraints on this relation, neural network under constant stimulus will always evolve to a state and stay there, and there can be one-to-one mapping between stimulus and the fixated state.

The remainder of paper goes as follows. Section II identifies the constraints, under which synaptic plasticity of one synaptic connection leads to fixed state and one-to-one stimulus-state mapping (memory). Section III extends the concepts of fixed state and one-to-one mapping for neural network consisting of many synaptic connections. Section IV proposes a biological classifier utilizing this memory.

II. SYNAPTIC CONNECTION AND ITS FIXED POINT

Let us start with one synaptic connection as shown in FIG 1. In nature, synapses are known to be plastic, low-precision and unreliable\textsuperscript{6}. This stochasticity allows us to assume synaptic strength \( s \) to be the probability (reliability) of propagating a nerve impulse through\textsuperscript{7,8} instead of being weight scalar as in artificial neural network\textsuperscript{9,10} (ANN). Easily we have \( y = xs \) where \( x, s, y \in [0,1] \). Now we treat synaptic plasticity, i.e. the relation between synaptic strength \( s \) and impulse propagation probability \( y \), as a function

\[
s^* = \lambda(y).
\]

Here \( s^* \in [0,1] \) represents the target strength a connection will be strengthened or weakened to over time if the connection is under constant propagation probability \( y \), while \( s \) in \( y = xs \) represents current strength. By \( y = xs \) and Eq. 1, we have \( s^* = \lambda(xs) \) stating that, under constant stimulus probability \( x \), the connection initialized with strength \( s \) will evolve towards \( s^* \). Our following reasoning hinges on this target strength function \( \lambda \), and we will put constrains on this uncharted function to see how they affect the dynamics of connection strength, and most importantly how stimulus is one-to-one mapped to strength at fixated state.

Here is our first constraint: \( \lambda \) is continuous on \( y \). This constraint is neurobiologically justifiable regarding synaptic plasticity, since sufficiently small change in impulse probability would most probably result in arbitrarily small change in synaptic strength. In that case, given any \( x \), \( \lambda(xs) \) is a continuous function on \( s \) from unit interval \([0,1]\) to unit interval \([0,1]\), and according to Brouwer’s fixed-point theorem\textsuperscript{11} (which states that, for any continuous function \( f \) mapping a compact convex set to itself, there is a point \( t \) such that \( f(t) = t \)) there must exist a fixed point \( s^* \in [0,1] \) such that \( s^* = \lambda(xs^*) \): connection strength at \( s^* \) will evolve to \( s^* \) and thus fixate, no longer strengthened or weakened. Moreover, as illustrated in FIG 2 given any initial value the strength always goes to fixed point. Therefore, a gentle constraint of continuity on \( \lambda \) function can preferably drives synaptic connection to fixed state.

To verify connection strength’s tendency towards fixed points, we design Algorithm 1\textsuperscript{12} to simulate our connection model. In this simulation\textsuperscript{13}, recent successful propagation is recorded and the success rate is supposed to approximate propagation probability \( y \); connection strength changes by a small step \( \Delta s \) each iteration to the direction of target strength. As shown in FIG 3, we run the simulation for four typical target strength functions, and the strength trajectories resulted

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{fig1.png}
\caption{Synaptic connection with strength \( s \) is directed from neuron 1 to neuron 2. Neuron 1 receives stimulus with probability \( x \) from environment or upstream neurons. Synaptic connection propagates nerve impulse (action potential) to neuron 2. As a result, neuron 2 receives impulse with probability \( y \); that is, neuron 1 and 2 "fire together" with probability \( y \).}
\end{figure}

\textsuperscript{1}Electronic mail: ls2@fuyunresearch.org
show that constraint of continuity ensures the tendency towards fixed points given any initial strength.

**Algorithm 1**

**Input:** stimulus probability $x$, initial strength $s_0$, function $\lambda$.

**Output:** trajectory of strength $s$.

1. initialize propagation recorder (10^4-entries array): $\text{recorder} \leftarrow 0$.
2. initialize propagation recorder pointer: $p \leftarrow 0$.
3. initialize current strength: $s \leftarrow s_0$.
4. initialize strength step: $\Delta_s \leftarrow 10^{-4}$.
5. for $i=0$ to $10^5$ iterations do
6. preset current pointed entry of recorder: $\text{recorder}[p] \leftarrow 0$.
7. pick random $r1$ and $r2$ from uniform distribution $\text{Unif}(0, 1)$.
8. if $x > r1$ and $s > r2$ then
9.   stimulus received, impulse propagated: $\text{recorder}[p] \leftarrow 1$.
10. end if
11. if recorder has been traversed once ($i \geq 10^4$) then
12.   set $y$ with the proportion of 1-entries in recorder.
13.   set target strength: $s^* \leftarrow \lambda(y)$.
14. if $s > s^*$ then
15.   step-increase current strength: $s \leftarrow \min(s+\Delta_s, 1)$.
16. end if
17. if $s < s^*$ then
18.   step-decrease current strength: $s \leftarrow \max(0, s-\Delta_s)$.
19. end if
20. end if
21. forward recorder pointer: $p \leftarrow (p+1) \mod 10^4$.
22. end for

Our goal is to establish one-to-one mapping between stimulus and connection strength at fixed point. Specifically, we could (1) given any stimulus $x \in [0, 1]$, identify the fixed point $s^*$ of connection strength without ambiguity; (2) given any strength $s^* \in [0, 1]$ at fixed point, identify stimulus $x$ without ambiguity. Among the four target strength functions in $\lambda(x)$, pin-point function $\theta$ as well as the conditions for it to be one-to-one mapping. In addition to constraint of continuity, let $\lambda(y)$ be strictly monotonic on $[0, 1]$ and hence one-to-one; let $\lambda(0) \neq 0$ to rule out fixed point $s^* = 0$. In that case, $\lambda$ has inverse function $\lambda^{-1}(s)$ which is strictly monotonic be-
Simulation is parameterized with \( \lambda \) and incremental \( x \). Ten trails are run for each incremental \( x \); in each trial initial strength is randomized; red line connects mean of ten fixed point values. The blue line represents true \( s^* = \theta(x) \).

(a) Simulation is parameterized with \( \lambda(y) = 0.9y + 0.05 \) and results match \( \theta(x) = 0.05/(1 - 0.9x) \) which is monotonically increasing.

(b) Simulation is parameterized with \( \lambda(y) = -y + 1 \) and results match \( \theta(x) = 1/(1 + x) \) which is monotonically decreasing.

FIG. 5. Simulation results of target strength function \( \lambda \) and fixed point strength \( s^* \). Simulation is parameterized with \( \lambda \) and incremental \( x \). Ten trails are run for each incremental \( x \); in each trial initial strength is randomized; red line connects mean of ten fixed point values. The blue line represents true \( s^* = \theta(x) \).

(a) Simulation is parameterized with \( \lambda(y) = 0.9y + 0.05 \) and results match \( \theta(x) = 0.05/(1 - 0.9x) \) which is monotonically increasing.

(b) Simulation is parameterized with \( \lambda(y) = -y + 1 \) and results match \( \theta(x) = 1/(1 + x) \) which is monotonically decreasing.

FIG. 7. (a) Our choices of \( \lambda \) functions are \( \lambda_L(y) = 0.99\sqrt{y} + 0.01 \) in blue and \( \lambda_T(y) = \frac{2}{1 + e^{-y+1}} - 1 \) in green. They both obey the \( \lambda \) constraints as previously mentioned. (b) Simulation results show that, \( \lambda_L \) leads to linear-like \( \theta_L \) in blue such that \( \theta_L(x) \approx x \), and \( \lambda_T \) leads to threshold-like \( \theta_T \) in green.

Now we have one-to-one (continuous and strictly monotonic) function \( \lambda, \lambda^{-1}, \theta \) and \( \theta^{-1} \). Given \( s^+ \) we can identify \( x \) and \( y \) without ambiguity, and vice versa. Our interpretation of these mappings is, synaptic connection at fixed point precisely "memorizes" the information of what (stimulus) it senses and how it responds (with impulse propagation).

III. NEURAL NETWORK AND ITS FIXED POINT

Now let us turn to neural network shown in FIG 8. Neural network could be treated as an "aggregate connection" as it turns out. We shall see that, definitions and reasoning for neural network align well with neural connection in last section.

As with synaptic connection, we can describe neural network by defining (1) external stimulus as an \( n \)-dimensional vector \( X \in [0,1]^n \) in which each \( x_i \) is the probability of neuron \( i \) receiving stimulus; (2) connections' strength as a \( c \)-dimensional vector \( S \in [0,1]^c \) in which each \( s_{ij} \) is the strength of connection from neuron \( i \) to neuron \( j \) (denoted as \( i \rightarrow j \)); (3) impulse propagation as a \( c \)-dimensional vector \( Y \in [0,1]^c \) in

between \( \lambda(0) \) and \( \lambda(1) \), and given any fixed point strength \( s^+ \) between we can identify stimulus \( x = \lambda^{-1}(s^+)/s^+ \). That is, function \( \theta^{-1}(s^+) = \lambda^{-1}(s^+)/s^+ \) exists. Let \( \lambda^{-1}(s)/s \) be strictly monotonic between \( \lambda(0) \) and \( \lambda(1) \). Then given any stimulus \( x \in [0,1] \) there is one single fixed point \( s^+ \) such that \( x = \lambda^{-1}(s^+)/s^+ \). That is, function \( s^+ = \theta(x) \) exists. Both of \( \lambda(y) = 0.9y + 0.05 \) and \( \lambda(y) = -y + 1 \) obey all those constraints and their one-to-one \( \theta \) functions can be verified by simulation results in FIG 4 whereas \( \lambda(y) = 0.5\sin(4\pi y) + 0.5 \) is not even strictly monotonic. However, neither \( \lambda(y) = 0.9y + 0.05 \) nor \( \lambda(y) = -y + 1 \) is ideal for our purpose. Guided by these constraints, we choose \( \lambda \) function carefully such that its derived \( \theta(x) \) function is monotonically increasing and range of which spans nearly the entire \([0,1]\) interval, as shown in FIG 7. Of all the \( \lambda \) constraints, continuity and strong monotonicity are reasonable requirements of consistency on the neurobiological process of synaptic plasticity, whereas \( \lambda(0) \neq 0 \) and strong monotonicity of \( \lambda^{-1}(s)/s \) are rather specific and peculiar claims. Admittedly, those \( \lambda \) constraints need to be supported by neurobiological evidences.
which each $y_{ij}$ is the impulse propagation probability through $i\rightarrow j$. In fact, one single neural connection is a special case of neural network with $c=1$ and $n=2$.

Stimulus and strength determine impulse propagation within neural network, so there exists a mapping $\Psi:(X,S)\rightarrow Y$. Presumably, the mapping $\Psi$ is continuous on $S$. By Eq. (1), there exists a mapping $\Lambda:Y\rightarrow S^\ast$, where $S^\ast=\{0,1\}^c$ is $c$-dimensional vector of connections’ target strength; each $i\rightarrow j$ has its own target strength function $\lambda_{ij}^\ast: y_{ij}\rightarrow \lambda_{ij}(y_{ij})$ for each $\lambda_{ij}^\ast$ in $S^\ast$. (Mapping $\Lambda$ could be visualized as a vector of such functions). With mapping $\Psi$ and $\Lambda$, we have a composite mapping $\Lambda \circ \Psi:(X,S)\rightarrow S^\ast$. If each $\lambda_{ij}$ function is continuous on its $y_{ij}$, mapping $\Lambda \circ \Psi$ must be continuous on $S$ and according to Brouwer’s fixed-point theorem given $X$ there exists one fixed point $S^\ast \in [0,1]^c$ such that $\Lambda \circ \Psi(X,S^\ast)=S^\ast$. And under constant stimulus $X$, neural network should go to fixed point $S^\ast$ as each connection $i\rightarrow j$ goes to its fixed point $\lambda_{ij}^\ast$. Our simulation verifies that tendency as shown in FIG 3.

In this simulation, impulses traverse neural network stochastically such that each neuron is fired at most once per iteration; synaptic connections update their strength as in Algorithm 1.

As with synaptic connection, our goal is to establish one-to-one mapping between stimulus $X$ and fixed point $S^\ast$ for neural network. Generally, the number of stable fixed points for a neural network is $\prod f_{ij}$ where $f_{ij}$ is the number of stable fixed points for $i\rightarrow j$. As in FIG 2(b), $\prod f_{ij}$ can be enormous when $f_{ij} \geq 2$. Therefore, continuity of $\lambda$ function makes neural network go to fixed point — not necessarily the unique one yet. With all $\lambda$ constraints, we have: (1) $\Lambda$ is one-to-one mapping and thus has inverse mapping $\Lambda^{-1}: S^\ast \rightarrow Y$; (2) there exists a mapping $\Theta:X\rightarrow S^\ast$, because under stimulus $X$ neural network will go to the unique fixed point $S^\ast$ no matter what initial strength $S_0$ it starts with; (3) if $\Theta$ is a one-to-one mapping, $\Theta$ has inverse mapping $\Theta^{-1}: S^\ast \rightarrow X$. With mapping $\Lambda$, $\Lambda^{-1}$, $\Theta$ and $\Theta^{-1}$ being one-to-one, given $S^\ast$ we can identify $X$ and $Y$ without ambiguity, and vice versa. Therefore, the same interpretation with respect to synaptic connection could apply: neural network precisely “memorizes” the information about stimulus on many neurons and propagation across many connections.

Nevertheless, even all of $\lambda$ constraints are not sufficient to secure one-to-one $\Theta:X\rightarrow S^\ast$ for neural network, as opposed to single neural connection. Here is a case. For $\Theta$ to be one-to-one, all neurons must have outbound connection. Otherwise, e.g., for a neural network with three neurons (say 0, 1 and 2) and two connections (say $0\rightarrow 1$ and $1\rightarrow 2$), stimulus $X_1=(1,1,0)$ and $X_2=(1,1,1)$ will lead to the same fixed point because stimulus on neuron 3, no matter what it is, affects no connection. Or equivalently, for $\Theta$ to be one-to-one, the definition of $X$ should consider only the neurons with outbound connections such that $X$’s dimension $\dim(X) \leq n$. In the perspective of information theory, many-to-one $\Theta$ introduces equivocation to neural network at fixed point, as if information loss occurred due to noisy channel. If $\dim(X) > \dim(S)=c$, mapping $\Theta$ conducts “dimension reduction” on stimulus $X$, and information loss is bound to occur. Here is another case. Consider a neural network with $0\rightarrow 2$, $1\rightarrow 2$ and $2\rightarrow 3$, and stimulus $X=(x_0,x_1)$. When the neural network is at fixed point, $x_3=x_0s_{10}^\ast + x_1s_{12}^\ast - x_0s_{12}^\ast x_0\Pr(1|0)$ where $\Pr(1|0)$ is the probability of neuron 1 being stimulated conditional on neuron 0 being stimulated. Then $\Pr(1|0)$ affects $s_{12}^\ast$ and hence $S^\ast$, or in other words neural network at fixed point gains the hidden information of $\Pr(1|0)$. However, if $\Pr(1|0)$ varies, given mere $X$ there will be uncertainty about $S^\ast$ such that mapping $\Theta$ doesn’t exist unless stimulus $X$ is “augmented” to $X=(x_0,x_1,\Pr(1|0))$.

IV. AN APPLICATION FOR CLASSIFICATION

Memory can manifest only as impulses propagation. Ideally, a neural network with memory of stimulus $X$ — formally,
mapping $\Theta$ casts memory of stimulus $X$ as fixed point $S^+$—should response to stimulus $X$ more "intensely" than the neural network with different memory responses to $X$. It is natural to differentiate response by counting the neurons fired or synaptic connections propagated by impulses. In this section, we adopt the count of synaptic connections propagated as a macroscopic measure of how intensely memory responses to stimulus or stimulus "recalls" memory. And we will propose a classifier consisting of $g$ neural networks, which classifies stimulus into one of $g$ classes by the decision criteria of which neural network gets the most synaptic connections propagated. Reminiscent of supervised learning, each neural network of our classifier is trained to its fixed point by impulses propagate across neural network stochastically, the count of synaptic connections propagated in one test should be random variable. Let it be $Z_{XX}$. Then for the neural network in FIG 8 $Z_{XX} = \sum z_{ij}$ where each r.v. $z_{ij} \sim \text{Bernoulli}(x_i, s_{ij}^+)$, i.e., synaptic connection $i \rightarrow j$ is propagated with probability $x_i s_{ij}^+$ in the test such that $Pr(z_{ij} = 1) = x_i s_{ij}^+$, $z_{ij}$'s expected value $\mathbb{E}[z_{ij}] = x_i s_{ij}^+$ and its variance $\text{Var}(z_{ij}) = x_i s_{ij}^+ (1 - x_i s_{ij}^+)$. By central limit theorem, $Z$’s distribution could tend towards Gaussian-like (bell curve) as $c$ increases, even if all $z_{ij}$ are not independent and identically distributed. We have

$$\mathbb{E}[Z_{XX}] = \sum_{i \rightarrow j} \mathbb{E}[z_{ij}] = \sum_{i \rightarrow j} x_i s_{ij}^+.$$

And when $c$ is large,

$$\text{Var}(Z_{XX}) \approx \sum_{i \rightarrow j} \text{Var}(z_{ij}) = \sum_{i \rightarrow j} x_i s_{ij}^+ (1 - x_i s_{ij}^+).$$

For any $i \rightarrow j$, in training stage because $S^+ = \Theta(X)$ we have $s_{ij}^+ = \Theta_j(X)$, and in testing stage $x_i$ is uniquely determined by $S^+$ and $X$ such that $x_i$ is a function of $X$ and $X$.

We experiment with this classifier to classify handwritten digit image. Ten identical neural networks (hence $g = 10$) of FIG 10 each designated for a digit from 0 to 9, are trained to their fixed points by their training images in FIG 11 as stimulus, and then testing images, also as stimulus, are classified into the digit whose designated neural network gets the biggest $Z$ value. We run many tests to evaluate classification accuracy, and collect $Z$ values to approximate r.v. $Z$’s distribution. With all synaptic connections equipped with $\lambda_i$ in FIG 7, the classifier has accuracy $\sim 44\%$, and $\sim 51\%$ with $\lambda_T$. Note that, equipped with $\lambda_L$ or $\lambda_T$, neural network of FIG 10 will have one-to-one $\Theta_L$ or $\Theta_T$ according to last section. FIG 12 and FIG 13 show that, in positive testing (e.g. digit-6 image is tested in neural network trained by digit-6 images), $Z$’s expected value (sample mean) could be considerably bigger than that in negative testing (e.g. digit-6 image is tested in neural network trained by digit-1 images), so as to discriminate digit-6 images from the others. Given the same testing image classification target can be different test by test since ten $Z$ outcomes are randomized. To improve classification accuracy, we shall distance the distribution of positive testing $Z$ from those of negative testing $Z$ as far as possible. We present another two special neural networks in FIG 14 to show how our classifier utilizes memory to classify images and how to improve its accuracy.
FIG. 12. The histogram (in probability density form) of \( Z \) values, a digit-6 image is tested many times on each of the ten trained neural networks. All connections are equipped with \( \lambda_T \). \( Z_{66} \) of positive testing is in red, and the other nine \( Z_{66} \) of negative testing, where \( k=0, 1, 2, 3, 4, 5, 7, 8, 9 \), is in gray. \( Z \)'s sample mean for each digit is depicted as vertical dotted line.

FIG. 13. The histograms of \( Z \) for all ten digits. For each digit, testing image randomly drawn, instead of the same one, is used in each test. From digit 0 to 9, classification accuracy is 70%, 41%, 56%, 42%, 53%, 33%, 77%, 51%, 57% and 32%.

value, is essentially a measure of similarity between the vectors. The bigger \( \mathbb{E}[Z_{XX}] \), is, the more intensely neural network with memory of training \( X \) responses to testing \( X \), and the more similar \( X \) and \( X \) are to each other. Therefore, Eq. (4) simply links otherwise unrelated neural response intensity and stimulus similarity. Comparing ten \( \mathbb{E}[Z_{XX}] \) values, we can tell which \( X \) is the most similar to \( X \) and hence which digit is classification target. Only, \( Z_{XX} \) value from test actually deviates around true \( \mathbb{E}[Z_{XX}] \) randomly, which makes it useable and yet unreliable classification criteria.

When the classifier equips all connections with threshold-like \( \lambda_T \) in FIG 7 classification accuracy raises to \( \sim 44\% \). By comparing FIG 15(b) with FIG 15(a), the distance between \( Z_{66} \)'s distribution and the other nine \( Z_{66,k \neq 6} \)'s distribution is bigger with threshold-like \( \lambda_T \) than with linear-like \( \lambda_L \). This accuracy improvement can be explained conveniently with a}

true threshold function

\[
\theta_i(x) = \begin{cases} 
0, & 0 \leq x < x_i \\
1, & x_i \leq x \leq 1 
\end{cases}
\]

By Eq. (4), of the sum terms in \( \sum_{i}^{64} x_i \bar{x}_i \), \( \theta_i \) basically diminishes small \( \bar{x}_i \in [0, x_i] \) to 0 and enhances big \( \bar{x}_i \in [x_i, 1] \) to 1, such that most probably \( \mathbb{E}[Z_{66}] \) would increase by having many \( \bar{x}_i=1 \) in its sum terms while the other nine \( \mathbb{E}[Z_{66,k \neq 6}] \) would decrease by having many \( \bar{x}_i=0 \) in their sum terms, so as to increase \( \mathbb{E}[Z_{66}] \sim \mathbb{E}[Z_{66,k \neq 6}] \). And meanwhile likewise \( \mathbb{V}[Z] \) would most probably decrease.

FIG 14(b) provides another type of neural network to improve classification accuracy without adopting threshold-like \( \lambda \) function for all synaptic connections. Let linear-like \( \lambda_L \) be equipped back and take \( \omega_0(\bar{x}_i)=100\bar{x}_i^3 \) for example. Then classification accuracy is \( \sim 47\% \). Now we have \( \mathbb{E}[Z_{XX}] \approx \sum_{i}^{64} x_i(100\bar{x}_i^3) \) where, likewise, \( 100\bar{x}_i^3 \) actually transforms \( \bar{x}_i \in [0, \sqrt[3]{0.1}] \) (here \( 0.1=0.56 \) to within \( 0,1 \)) and transforms \( \bar{x}_i \in [\sqrt{0.1},1] \) to across \( [0,100] \) — again, strong training pixel-stimulus are weighted while weak ones are relatively suppressed. As shown in FIG 15(c) the distance between the distribution of \( Z_{66} \) and \( Z_{66,k \neq 6} \) is increased compared to FIG 15(a). Here our neurobiological interpretation regarding \( \omega_0(\bar{x}_i) \) is that, training stimulus affects not only synaptic strength, but also the growth of neuron cluster in terms of replication of neuron cells and formation of new synaptic connections.

And TABLE I shows the performance of our classifiers with the four typical \( \lambda \) functions in FIG 3 to demonstrate how "pathological" target strength functions affect classification.

As mentioned, handwritten digit images classification could be simplified to a task of generic linear classification: given ten classes each with its discriminative function \( \delta_i(X)=\bar{x}_i X \), image \( X \) is classified to the class with the largest
FIG. 15. Z histogram of testing digit-6 images for three different classifier settings. For each classifier setting, values of $Z_{66}$ and $Z_{66,k=6}$ are transferred to z-scores (i.e. the number of standard deviations from the mean a data point is) with respect to the distribution of all $Z_{66,k=6}$'s values combined. The distance between the distribution of $Z_{66}$ and $Z_{66,k=6}$ is approximately evaluated by $E[Z_{66}]$ and standard deviation $\sigma(Z_{66})$. (a) Classifier with neural networks of FIG 14(a) and $\lambda_T$. $E[Z_{66}] \approx 0.93$ and $\sigma(Z_{66}) \approx 0.98$. (b) Classifier with neural networks of FIG 14(a) and $\lambda_T$. $E[Z_{66}] \approx 1.33$ and $\sigma(Z_{66}) \approx 0.9$. (c) Classifier with neural networks of FIG 14(b), $\lambda_T$ and $\omega_i(\bar{x}_i)=100\bar{x}_i^5$. $E[Z_{66}] \approx 1.39$ and $\sigma(Z_{66}) \approx 0.87$.

TABLE I. Classification accuracy on different classifier settings. For a classifier, we adopt neural networks of FIG 10, FIG 13(a) or 13(b), and equip synaptic connections with target strength functions of FIG 3. The accuracy listed is the average of many results taken from the same trained classifier, and thus could fluctuate from one training to another.

| $\lambda$ functions | FIG 10 | FIG 14(a) | FIG 14(b) |
|---------------------|--------|-----------|-----------|
| $\lambda(y) = 0.9y + 0.05$ | 14% | 16% | 19% |
| $\lambda(y) = 0.5 \sin(4\pi y) + 0.5$ | 56% | 6% | 2% |
| $\lambda(y) = -y + 1$ | 4% | 5% | 19% |
| Discontinuous $\lambda$ | 23% | 20% | 28% |

a Accuracy under 10% is actually worse than wild guessing.
b When classification criteria is changed to "which neural network gets the least synaptic connections propagated", accuracy is $\approx 40%$.

d value. Our neural classifier simply takes over the computation of vectors' dot product $\bar{x}_i X$ and adds randomness to the ten results. To parameterize $\bar{x}_i$ in ten $\delta_i$, "supervisors" could train neural networks in the classifier with the images they deem the best — "average images" in our case or digits learning cards in teachers' case. Our neural classifier is rather unreliable and primitive compared to ANN which is also capable of linear classification. On one hand, given the same image ANN always outputs the same prediction result. On the other hand, ANN is not only a classifier but also more importantly a "learner", which learns from all kinds of handwritten digits to find the optimal $\bar{x}_i$ for ten $\delta_i$; Optimal $\bar{x}$ minimize error of misclassification, making ANN more tolerant with poor handwriting and thus have better prediction accuracy. Only, ANN's learning optimal $\bar{x}_i$, an optimization process of many iterations, requires massive computational power to carry out, which is unlikely to be provided by real-life nervous system — there is no evidence that individual neuron can even conduct basic arithmetic operation. Despite of its weakness, our neural classifier has merit in its biological nature: it reduces the computation of vectors’ dot product to simple counting of synaptic connections propagated; its training and testing could be purely neurobiological development and activities where no arithmetic operation is involved; its classification criteria, i.e. "deciding" or "feeling" which neural (sub)network has the most connections propagated, could be an intrinsic capability of intelligent agents.

V. CONCLUSION

This paper proposes a mathematical theory to explain how memory forms and works. It all begins with synaptic plasticity. We find out that, synaptic plasticity is more than stimulus affecting synapses; it actually plays as a force that can drive neural network over time to a long-lasting state. We also find out that, under certain conditions there would be one-to-one mapping between the neural state and the stimulus that neural network is exposed to. With the mapping, given stimulus we know exactly what neural state will be; given neural state we know precisely what stimulus has been. The mapping truly is a link between past event and neural present; between the short-lived and the enduring. In that sense, the mapping itself is memory, or the mapping casts memory in neural network. Next, we study how memory affects neural network’s response to stimulus. We find out that, neural network with memory of stimulus can respond to similar stimulus more intensely than to stimulus of less similarity, if response intensity is evaluated by the number of synaptic connections propagated by impulses. That is, neural network with memory is able to classify stimulus. To verify this ability, we experiment with classifiers consisting of ten neural networks, and they turn out to have considerable accuracy in classifying handwritten digit images. Those classifiers prove that neurons could collectively provide fully neurobiological computation for classification.

Our reasoning takes root in the mathematical treatment of synaptic plasticity as target strength function $\lambda$ from impulse frequency to synaptic strength. We put hypothetical constraints on this $\lambda$ function to ensure that the ideal one-to-one mapping exists. Although these constraints are necessary to keep our theory mathematically sound, they raise concerns. Firstly, they could be overly restrictive. Take continuity constraint for example. Even the discontinuous function of FIG 3(d), whose nonexistent $\theta$ function would map certain stimulus to any point within "fixed interval" instead of a specific fixed point as shown in FIG 3, can be a useable $\lambda$ in our classifier according to TABLE II. In this case, fixed point per se doesn’t have to exist, and mere tendency to seek out for it could serve the purpose. Secondly, as discussed in Section II, those $\lambda$ constraints have yet to be supported by neurobiological evidences. Above all, evidences that reveal true $\lambda$ is vital to clarify uncertainty.
1. D. Hebb. *The organization of behavior: a neuropsychological theory*. John Wiley & Sons, 1949.

2. S. Lowel and W. Singer. Selection of intrinsic horizontal connections in the visual cortex by correlated neuronal activity. *Science*, 255(5041):209–212, 1992.

3. J. Hughes. Post-tetanic potentiation. *Physiol Rev.*, 38(1):91–113, 1958.

4. G. Berlucchi and H. Buchtel. Neuronal plasticity: historical roots and evolution of meaning. *Experimental Brain Research*, 192(3):307–19, 2009.

5. A. Citri and R. Malenka. Synaptic plasticity: multiple forms, functions, and mechanisms. *Neuropsychopharmacology*, 33:18–41, 2008.

6. C. Baldassi, F. Gerace, H. Kappen, C. Lucibello, L. Saglietti, E. Tartaglione, and R. Zecchina. Role of synaptic stochasticity in training low-precision neural networks. *Phys. Rev. Lett.*, 120(26), 2018.

7. C. Laing and G. Lord. *Stochastic methods in neuroscience*. Oxford University Press, 2010.

8. H. Seung. Learning in spiking neural networks by reinforcement of stochastic synaptic transmission. *Neuron*, 40(6):1063–1073, 2003.

9. T. Branco and T. Staras. The probability of neurotransmitter release: variability and feedback control at single synapses. *Nat Rev Neurosci.*, 10(5):373–83, 2009.

10. J. Schmidhuber. Deep learning in neural networks: an overview. *Neural Networks*, 61:85–117, 2015.

11. I. Istratescu. *Fixed point theory: an introduction*. Springer, 1981.

12. Source code can be found at https://github.com/lansiz/neuron.

13. C. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27(3):379–423, 1948.

14. T. Hastie, R. Tibshirani, and J. Friedman. *The elements of statistical learning*. Springer, 2009.

15. The dataset of 1797 handwritten digit images can be obtained with Python code "from sklearn import datasets; datasets.load_digits()".