Localist plasticity identified by mutual information

Gabriele Scheler and Johann Schumann

Carl Correns Foundation for Mathematical Biology, Mountain View, Ca. 94040. USA
gscheler@gmail.com

Abstract. The issue of memory is difficult for standard neural network models. Ubiquitous synaptic plasticity introduces the problem of interference, which limits pattern recall and introduces conflation errors.
We present a lognormal recurrent neural network, load patterns into it (MNIST), and test the resulting neural representation for information content by an output classifier. We identify neurons, which 'compress' the pattern information into their own adjacency network, and by stimulating these achieve recall. Learning is limited to intrinsic plasticity and output synapses of these pattern neurons (localist plasticity), which prevents interference.
Our first experiments show that this form of storage and recall is possible, with the caveat of a 'lossy' recall similar to human memory. Comparing our results with a standard Gaussian network model, we notice that this effect breaks down for the Gaussian model.

1 Introduction

Storing patterns and achieving human-like recall is a problem for neural network models. Relying on synaptic plasticity introduces the problem of interference, which has been harnessed as the useful property of generalization, but does not allow for many individual events to be stored and remembered without blending their properties and losing information over time. Introducing layers allows for feature abstraction as stored memory elements but narrowly focuses a neural network on a single problem set.

Here, we employ a lognormal recurrent neural network, load patterns, and test the resulting neural representation for information content by an output classifier. This setup is reminiscent of LSN [8] and ESN [5]; but it is also a standard setup for a computational brain model [6, 11]. Mutual information (MI) analysis shows that both multi-pattern and single-pattern response neurons spontaneously form in the representations. We identify high MI neurons, which 'compress' the pattern information into their own adjacency network, and by stimulating these achieve recall that is free of interference by learning [9].

We restrict plasticity to intrinsic properties and synapses of these high MI neurons. This localist learning means that a disjunctive set of weights is adapted for each pattern. As a result, when these neurons are being activated by direct

stimulation, they activate a whole set of related neurons to recreate their original patterns. In this network model individual neurons are recruited as pattern storage elements. We thus achieve a localist memory with a distributed component. Such a model has the ability to explain a number of biological properties of memory such as low-interference storage of event memory, which needs to be activated and reconsolidated, and extinction which overlays existing memory such that it ceases to affect behavior but which does not erase the original trace. Furthermore it allows for technically useful properties of hierarchical memory by combining local and distributed memories.

Our first experiments show that this form of storage and recall is possible, with the caveat of a 'lossy' recall—which is in line with psychological evidence. The main property of psychological recall is not conflation or interference but incompleteness or loss of information. In the future, we plan to investigate the 'lossy' recall and develop it as a useful filter for information. We have also compared our encouraging results with a standard Gaussian network model and notice that the effects break down catastrophically. We do not give a mathematical explanation in this paper, only simulation results.

2 Description of Model

Network Structure: The model consists of 1000 excitatory (E) neurons, of which 400 receive direct input, and 200 inhibitory (I) neurons. Synaptic connections (NMDA, AMPA and GABA-A) are modeled as in [11]. Neurons are modeled as in [3] with an equation for the membrane model v and an equation for the gating variable u (Eq 1).

$$\begin{aligned} \dot{v} &= 0.04v^2 + 5v + 140 - u - I_{syn} \\ \dot{u} &= a(bv - u) \end{aligned} \qquad (1)$$

	E	I
a	0.00125 . . . 0.036	0.02
b	0.0125 . . . 0.36	0.2
c	-70	-70
d	3	2

When the neuron fires a spike (defined as $v(t) = 30mV$), v is set back to a low membrane potential $v := c$ and the gating variable u is increased by the amount d. All parameters for the neurons are shown in the table above. For E neurons, parameters a and b are varied [11]. This results in different intrinsic excitability (gain). Similar to attested values for cortical neurons [10], a lognormal distribution of excitability with $\mu = 4.96$ and $\sigma^2 = 0.31$ for E neurons is used for the neurons in the network. From E neurons to E and I neurons, we use full connectivity and a lognormal distribution of synaptic strength, for both AMPA and NMDA connections. From I neurons to E neurons, we use a normal distribution for GABA-A connections. There are no I-I connections. Overall synaptic strength is adjusted to achieve a realistic spontaneous spiking pattern.

Patterns: We used the MNIST database [7] of handwritten digits, with 50 variations for each digit, in the format of an integer vector of length 400.

Output Classification: We used a deep learning tool [1] with 100 and 50 nodes in two layers with ReLu and Softmax activation functions [2] to achieve high precision in classification of representations for 50 variations of 10 target patterns (rate measured for 1400 ms after 300ms stimulus presentation), plus 50 variations of no pattern, with 450 training, 100 test patterns.
MI Analysis: We run our 1200 neuron network while applying one input pattern of length 400 at a time for 300ms. All patterns s are presented with equal probability ($p(s) = 1/N_{patterns}$) and spike rates are recorded after the end of the stimulus for a length of 300ms or 1400ms. We assign a rate value r to each neuron

$$r = \begin{cases} 1 & 80\% < f^s_{rel} < 120\% \\ 2 & f^s_{rel} \leq 80\% \\ 3 & f^s_{rel} \geq 120\% \end{cases} \tag{2}$$

where f^s_{rel} is the neuron's firing rate compared to its spontaneous rate for pattern s.

For each digit pattern, we calculate mutual information (MI) between stimulus (S) and response (R), based on r and s (Eq. 3).

$$MI_R = \sum_r p(s)p(r|s) * \log_2 \frac{p(r|s)}{p(r)} \tag{3}$$

Plasticity: Learning rules for intrinsic excitability and synapses are based on a neuron's MI_R value and spike rate r_n, employing fixed learn rates $\lambda = 0.1$, $\kappa = 0.1$, $\beta = 0.05$, and threshold $\theta = 0.1$. Synapses are updated only when they receive input from a neuron with high MI and high activation, then Hebbian learning applies.

$$\begin{array}{llll} \text{if } MI_R(n) > \theta \wedge r_n = 3 & \Rightarrow & a_n := a_n(1+\lambda);\ b_n := b_n(1+\kappa); & \text{(LTP-IE)} \\ \text{if } MI_R(n) > \theta \wedge r_n = 1 & \Rightarrow & a_n := a_n(1-\lambda);\ b_n := b_n(1-\kappa); & \text{(LTD-IE)} \\ \text{if } MI_R(n) > \theta \wedge r_n = 3 & \Rightarrow & \forall i\ r_i = 3: \quad w_{ni} := w_{ni}(1+\beta) & \text{(LTP)} \\ \text{if } MI_R(n) > \theta \wedge r_n = 3 & \Rightarrow & \forall i\ r_i = 1: \quad w_{ni} := w_{ni}(1-\beta) & \text{(LTD)} \end{array}$$

3 Results

First, we established a good fit of our network with lognormal distributions of gains, spike rates, and weights in cortical networks [10].

Secondly, we presented input patterns for digits to a naive, unlearned network for 300ms, and measured responses for 300ms and 1400ms after the end of the presentation. We used the 1400ms responses throughout, which reflect the information from the input slightly better. The representations obtained with the unlearned network are sufficient to be recognized by a standard (deep-learning) classifier for all 50 variations of a digit pattern (Figure 1). The accuracy obtained could possibly be improved by adjusting the classifier or by using more patterns.

We analyzed the representations for information content, by measuring mutual information (MI) between each neuron and the 10 digit input patterns

Fig. 1. Output classification for 50 variants of 11 input conditions (10 digits and no pattern). 450 patterns were used for training, 100 patterns for testing.

(Figure 2), and by plotting the specificity for each neuron for a single digit or multiple digits (Figure 3). We found that high MI neurons ($MI > 0.1$) correspond to single input pattern specificity, while intermediate MI neurons respond to multiple input patterns, and that there are $\approx 5\%$ of high MI neurons for each pattern. (Low MI neurons are not specific).

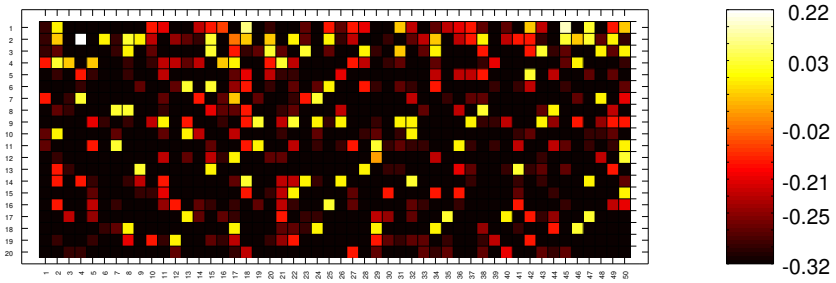

Fig. 2. MI analysis of neurons shows the MI_R for each neuron

Now we apply Hebbian learning rules for intrinsic excitability and synaptic weights to neurons and their synapses with single pattern specificity. We find that a network trained in this way (a) shows considerable stability in its response to input, since only a subset of neurons and synapses is affected and (b) that by stimulating only the pattern-specific neurons ('evoked memory') we can 'recall' the original input representation with some amount of error (Figure 4). We ran the algorithm for a number of updates and found that evoked memory improved for some time and then plateaued. It is remarkable in this model that all pattern memories are stored in parallel, and stimulation of pattern-specific neurons will reproduce the associated distributed pattern with considerable accuracy. Applying both intrinsic and synaptic learning improves the accuracy of recall (Figure 4). This is an initial result, based on a simple learning rule, which could be further developed. We compare this result with a standard Gaussian network (normal distribution of intrinsic excitability and synaptic weights). The

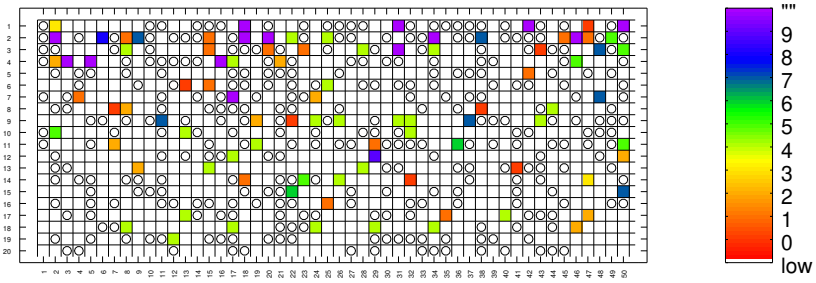

Fig. 3. Specificity of neural response. High MI ($MI > 0.1$) matches to single-pattern nodes (color gives pattern number), intermediate MI to multi-pattern nodes (circles), and low MI to no pattern response (white). The 1000 E neurons are shown in a 20x50 grid.

recall accuracy is much lower on average and not sustained for single patterns (Figure 4).

Figure 5 shows the size of response after targeted stimulation for both a lognormal and a Gaussian network. The response is much larger in the Gaussian network, which however corresponds to loss of specificity.

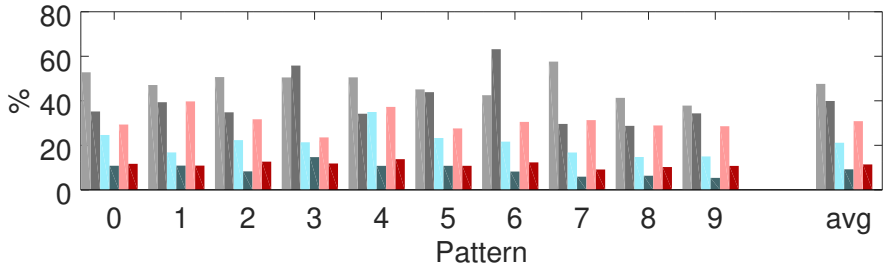

Fig. 4. Recall precision after stimulating single-pattern neurons for the corresponding pattern (light colors) and a non-corresponding pattern (darker colors). Reported is the percentage of matching neurons of all activated (> 120%) neurons. Shown are a Gaussian network (gray) and a lognormal network before (blue) and after training (red) for each of the 10 digit patterns and the average. The difference in response between corresponding and non-corresponding stimuli is universally apparent and larger for our log-normal network and has increased with training.

4 Discussion

We could show that it is possible to combine sparse (single-pattern) and distributed (multi-pattern) coding. By stimulating single-pattern neurons, the resulting representation (recall) was sufficient to recognize the corresponding pattern.

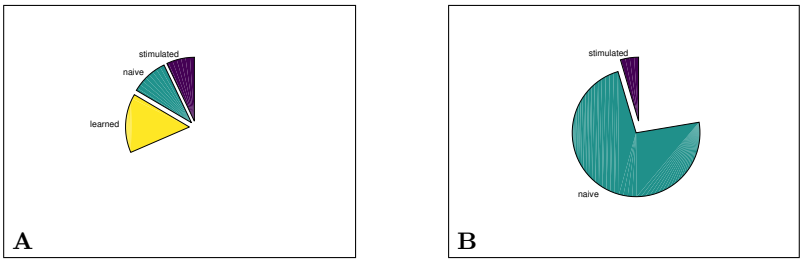

Fig. 5. Size of the recall response for lognormal network (A) and Gaussian network (B). Stimulation of the single-pattern neurons (dark blue, A: 28, B: 18) for the untrained (green, A: 38, B: 292) and trained network (yellow, A: 60). In the Gaussian network there is a much larger but undifferentiated response.

The learning rules are localized, such that only neurons (and their synapses), with high mutual information, which are most specific for a pattern, undergo plasticity. We have not discussed the biological mechanisms which could underlie such selective plasticity, but there are possibilities in the cell-internal memory that 'counts' activations over time and filters the information by a number of indicators, such as small molecules and proteins. This way of representing pattern information in a network allows for interference-free learning, and 'lossy' recall which could be considered a filter for useful information.

References

1. M. Abadi, et al.: TensorFlow: Large-scale machine learning on heterogeneous systems (2015), http://tensorflow.org/
2. F. Chollet, et al.: Keras. https://keras.io (2015)
3. E.M. Izhikevich: Which model to use for cortical spiking neurons? IEEE Trans Neural Netw. **15**(5):1063–1070 (2004)
4. E.M. Izhikevich, J. Gally, G. Edelman: Spike-timing dynamics of neuronal groups. Cereb Cortex. **14**(8): 933–944 (2004)
5. H. Jaeger: Adaptive nonlinear system identification with echo state networks. In Advances in Neural Information Processing Systems (NIPS) 15, MIT Press, pp. 593-600 (2003)
6. J. Kim, W. Leahy, E. Shlizerman. Neural Interactome: Interactive Simulation of a Neuronal System. Frontiers in Computational Neuroscience. **13**: 8 (2019)
7. Y. LeCun and C. Cortes: MNIST handwritten digit database. http://yann.lecun.com/exdb/mnist/ (2010)
8. W. Maass, T. Natschlaeger, H. Markram: Real-time computing without stable states: A new framework for neural computation based on perturbations. Neural Computation, **14**(11):2531-2560 (2002)
9. Scheler, G.: Extreme pattern compression in lognormal networks. F1000Research **6**:2177(poster) (2016)
10. Scheler, G.: Logarithmic distributions prove that intrinsic learning is Hebbian. F1000Research **6**:1222 (2017)
11. Scheler, G.: Neuromodulation influences synchronization and intrinsic readout. F1000Research, **7**:1277 (2018)