論文の内容の要旨

論文題目 **Exploring New Neural Architectures for Adaptation to Complex Worlds**

(新しい構造に基づいて複雑性に対処する

ニューラルネットワークの研究)

氏名　シナパヤ　ラナ　Sinapayen Lana
Advisor: Ikegami Takashi

Introduction

The real world, compared to simulated models, contains massive amounts of information. The amount of information itself is not complexity, but the way this information comes together is referred as environmental complexity. After losing popularity in the 2000s, neural-network based processing has been gathering attention again thanks to the impressive results obtained by the Deep Learning community. Despite progress in classification and reinforcement learning, deep learning networks sill have issues such as the lack of robustness to noise and the difficulty to optimize hyper-parameter values. In this work, we present two learning algorithms for two different kinds of artificial neural networks that are unrelated to deep learning methods, but provide promising results learning under high environmental complexity. The first algorithm, Learning by Stimulation Avoidance (LSA), allows a spiking network to learn a desired behaviour despite intrinsically noisy inputs and outputs. The second algorithm is applied to a new kind of neural network that changes its number of neurons, which we name Epsilon Network (e-network). The e-network adapts its architecture to the complexity of a data stream in real time.

1 LSA

1.1 Introduction

Inspired by the work of Shahaf and Marom [1], we argue for the existence of a principle allowing to steer the dynamics of a biologically inspired neural network: “Learning by Stimulation Avoidance” (LSA). We show that LSA works in a minimal network, then in a 100-neuron network, and use it in a embodied application: learning of wall avoidance by a simulated robot.

We simulate excitatory neurons and inhibitory neurons and add a Spike Timing Dependent Plasticity rule. The neurons receive three kinds of input: (1) Zero-mean Gaussian noise is injected in each neuron at each time step; (2) External stimulation, which is stopped when the network exhibits the desired output. (3) Stimulation from other neurons: when a pre-synaptic neuron spikes, the value of the weight is added as an input for the post-synaptic neuron.

1.2 Results

In the first experiment we examine the weights dynamics in 3 fully connected excitatory neurons to see how LSA works when stimulation is applied to a network with an input neuron, a hidden neuron and an output neuron. The minimal network dynamics follow the principle of LSA. When spiking of the output neuron stops the stimulation in the input neuron, the weight from input to output is increased (reinforcement). When spiking of the output neuron starts external stimulation to the input neuron, the weight from input to output is pruned (pruning).

In the second experiment we find that global bursts in the network can impair learning and we suppress bursting in a 100-neuron network by reducing the number of connections and applying high internal noise. The goal is to obtain selective learning, by increasing the weights to Output Zone A and prune those to Output Zone B, therefore obtaining different firing rates. We fix two experimental conditions:
(Stop Condition) Input Zone A is stimulated. After $n >= 1$ neurons in Output Zone A spike, the external stimulation to Input Zone A is stopped. If $n >= 1$ neurons do not spike in Output Zone A after 10,000 ms of stimulation, the stimulation is also stopped. After a random delay, the stimulation starts again. (Stimulus Condition) After $n >= 1$ neurons spike in Output Zone B, the whole network (excluding inhibitory neurons and Output Zone B itself) is stimulated for 10 ms. As a result of LSA, the network moves from a state where both output zones fire at the same rate, to a state where Output

Zone B fires at lower rates and Output Zone A fires at higher rates.

In the last experiment we simulate a robot with distance sensors that stimulate the network when the robot is close to walls: the more the robot learns to avoid walls, the less stimulation it receives. As a result, the robot learns to avoid walls. Here the stimulation is not controlled by the experimenter, but by the interaction between the robot and the environment. LSA is organized by the robot in the environment, and is more robust to noise than a standard wall avoidance algorithm.

2 Epsilon-Network

2.1 Introduction

We propose the Epsilon Network (ϵ-network, related to [2]), a network that automatically adjusts its size (adding and removing neurons and weights) to the complexity of a stream of data while performing online learning. We evaluate it on simple, complex, and noisy videos and show that the final number of neurons is a good indicator of the complexity and predictability of the data stream.

The network is composed of binary valued neurons and 2 types of weights: prediction weights PW and instantaneous weights IW. The IW propagate activation through the network. The PW are only used to calculate the predicted activation of the neurons. There are no layers, and initially there are no PW in the network, only disconnected neurons. The difference between the prediction computed by the network and the actual input at t + 1 is called surprise and is used to update the weight values and add neurons and PW to minimise the value of the surprise; simultaneously, redundancy is calculated by finding equivalent output probabilities on neurons, and redundant neurons are pruned. Two neurons are fused together if they have the same output values; their input weights are reported to the fused neuron. (Fig. 1),

2.2 Results

In the 1st experiment we perform a simple modeling task with a time series consisting frames showing a ball falling. The network starts with 1500 neurons sensitive to the pixels from the images. The network is trained by looping on these images; we find that the number of neurons decreases until there are only 4 neurons left. The number of weights increases at first, then decreases proportionally with the number of neurons. At the end of the task, the prediction is perfect and the network exactly represents the automaton

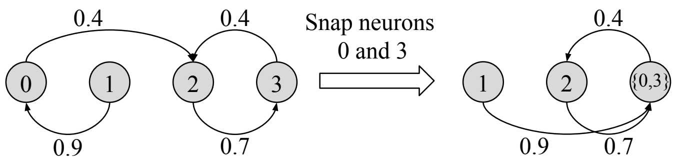

Figure 1: **Example of snapping procedure.** Two neurons are fused together if they have the same output values; their input weights are reported to the fused neuron.

describing the time series.

In the second experiment, we compare the results for a simple, repetitive video and a complex video with the same number of frames. By the end of the experiment, both networks achieve equally good prediction performance, but the number of neurons and connections is much higher for the complex video sequence. We also compare a video with and without noise, in order to see the difference in network structures. We find that the network needs more neurons to deal with the noisy video. Finally, we compare ϵ-network to a state of the art Deep learning algorithm and find that although the performance of our network is worse by a factor 10, it uses 10^7 times less neurons than the Deep Learning network, trains faster, and does not require parameter tuning.

References

[1] Shahaf, G., Marom, S., "Learning in networks of cortical neurons". In: The Journal of Neuroscience 21.22 (2001), pp. 8782–8788.
[2] Crutchfield, J. P., Young, K., "Inferring Statistical Complexity". In: Physical Review Letters 63.2 (1989), pp. 105–108.