Differential plasticity - Based on the paper by Thomas Miconi et al in ICML 2018

Sriram Ganapathi Subramanian (20676799)

University of Waterloo

s2ganapa@uwaterloo.ca

November 1, 2018

Motivation - Meta learning problem

- Neural Networks are static in architecture.
- Humans and higher order animals possess **synaptic plasticity**.
- Plasticity is the capability of changing the network connections over time.
- Differential plasticity - The plastic connections' behavior is trained using gradient descent
- Example: Life long learning in NN trained to recognize alphabets.

Introduction

- Tackle meta learning.
- Special emphasis on gradient descent capability.
- Optimize base weights and the amount of plasticity using Back Propagation.
- Ease of implementation of plasticity with popular deep learning packages and computational efficiency.
- Considers three complex and different domains.

Related Work - Previous Approaches

- Using external memory banks.
- Non trainable plastic weights that grows and decays as a function of inputs and outputs.
- Parameterized learning rule and expert systems.
- Separate plasticity network trained separately.
- Meta learning base network and several derived networks.

Model

Activation Equation

$x_j(t)$ the output of neuron j is given by the equation

$$x_j(t) = \sigma\{ \sum_{i \in \text{inputs}} [(w_{i,j} x_i(t-1) + \alpha_{i,j} H_{i,j}(t)) x_i(t-1)] \} \tag{1}$$

Plastic Component

The Hebbian trace is given by:

$$H_{i,j}(t+1) = \eta x - i(t-1) x_j(t) + (1-\eta) H_{i,j}(t) \tag{2}$$

Alternate Form/Oja's Rule

To prevent hebbian trace decay to 0, it The is replaced by

$$H_{i,j}(t+1) = H_{i,j}(t) + \eta x_j(t)(x_i(t-1) - x_j(t) H_{i,j}(t)] \tag{3}$$

Experiments

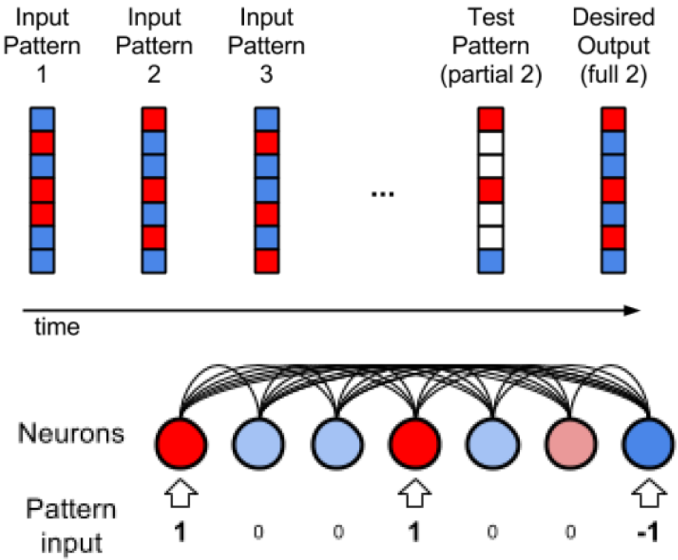
Four different experiments have been considered in this paper to demonstrate the superiority of performance.

- Experiment 1 - Binary pattern memorization
- Experiment 2 - Natural images pattern memorization
- Experiment 3 - Omniglot task
- Experiment 4 - Reinforcement learning Maze navigation task

Binary pattern memorization experimental setup

- The network is shown a set of five binary patterns with 1000 elements in succession.
- Each pattern is shown for 10 time steps, with 3 time steps of zero input between them. Whole sequence is presented 3 times in random order.
- One of the presented pattern is chosen at random and degraded by setting half of its bits to 0 and shown to network.
- The network has to reproduce the correct full pattern in its output using its memory that it developed during training.

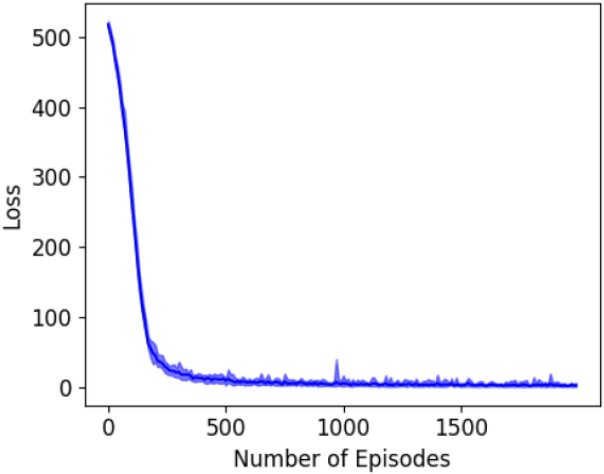
Experimental Setup



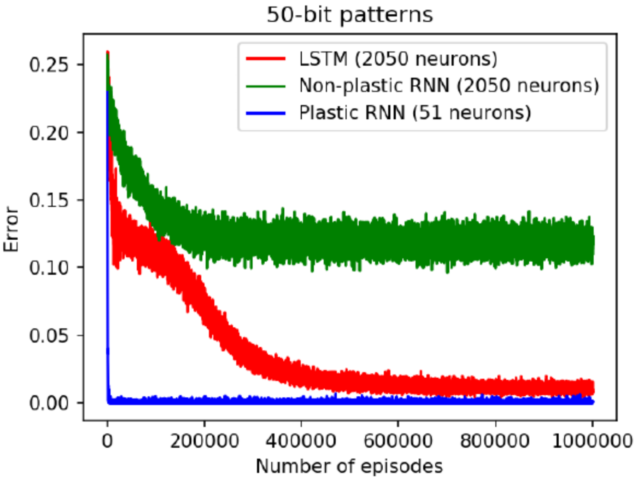
Binary pattern memorization network architecture

- Fully connected RNN with one neuron per pattern element, plus one fixed-output neuron.
- Value of each neuron is clamped to the value of the corresponding element (except for 0 input).
- Loss = Network output - correct pattern.
- The gradient computed back propagation .
- There are a total of $1001 \times 1001 \times 2 = 2004002$ trainable parameters.

Results



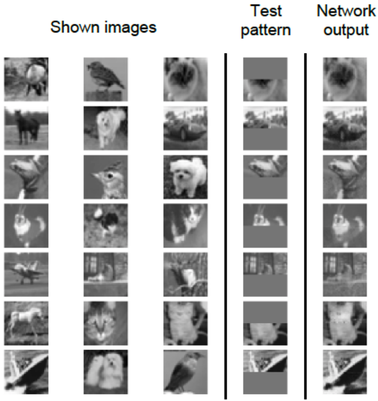
Comparison to non plastic architectures



Experiment2 - Natural images pattern memorization
Experimental Setup

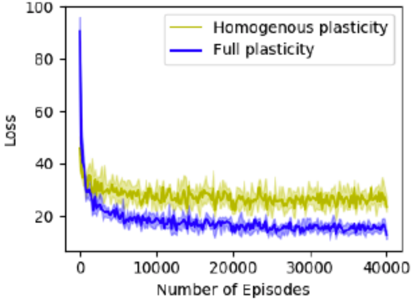
- Images are from the CIFAR-10 database where there are a total of 60000 images each of size 32×32.
- The architecture has 1025 neurons in total with a total of $2 \times 1025 \times 1025 = 2101250$ parameters.
- Each episode has 3 pictures, shown 3 times for 20 time steps each time, with 3 time steps of zero input between the presentations.
- The test images are degraded by zeroing out one full contiguous half of the image.

Results



(a) Typical image reconstruction results from a withheld test set (not seen during training). Each row is a full episode.

Comparing independent and shared α value runs.



Experiment2 - Results

- The model has learned to solve the task from the final output images.
- The full plastic network is compared against a similar architecture with shared plasticity coefficients.
- Independent plasticity has better performances.

Experiment3- Omniglot task experimental setup

- The Omniglot dataset is a collection of handwritten characters from various writing systems.
- In each episode N character classes are randomly selected and K instances from each class are sampled.
- These instances, together with the class label (from 1 to N), are shown to the model.
- Then, a new, unlabelled instance is sampled from one of the N classes and shown to the model.
- Model performance is defined as the model accuracy in classifying this unlabelled example.

Experiment3- Omniglot task Architecture

- Model architecture has 4 convolutional layers with 3×3 receptive fields and 64 channels.
- All convolutions have a stride of 2.
- The output is a single vector of 64 features, which feeds into a softmax layer.
- The label of the current character is also concurrently fed as a one-hot encoding to this softmax layer and compared.
- Plasticity is applied only to the weights from the final layer to the softmax layer.

Experiment3- Omniglot task Data Preparation and Training

- The dataset is augmented with rotations by multiples of 90 deg.
- It is divided into 1,523 classes for training and 100 classes are chosen for testing.
- The networks are trained with an Adam optimizer with a learning rate 3×10^{-5}, multiplied by 2/3 every 1M episodes over 5,000,000 episodes.
- The networks are tested on prediction of the unlabelled instances.

Omniglot task results

Algorithm	Accuracy
Memory Networks	82.8%
Matching Networks	98.1%
ProtoNets	97.4%
Memory Module	98.4%
MAML	98.7% $\pm$ 0.4
SNAIL	99.07% $\pm$ 0.16
DP	98.3% $\pm$ 0.80

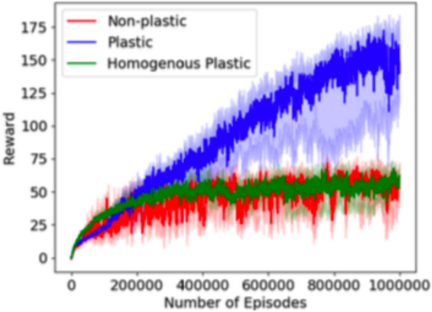
Experiment 4 - Reinforcement learning task - Experimental Setup

- Domain - A classic grid world maze in Reinforcement learning consisting of walled and non walled squares.
- At each episode, one non-wall square is randomly chosen as the reward location.
- Each episode lasts 250 time steps, during which the agent must accumulate as much reward as possible.
- Inputs to the agent consist of a binary vector describing the 3×3 neighborhood centered on the agent.
- Three conditions: full differentiable plasticity, no plasticity at all, and homogeneous plasticity.

Experiment 4 - Reinforcement learning task - Architecture and Results

- The architecture is a simple recurrent network with 200 neurons, with a softmax layer on top of it to select between the 4 possible actions (up, right, left or down).
- The plastic network shows considerably better performance as compared to the other networks.
- The non plastic and homogeneous networks get stuck on a sub optimal policy.
- Individually sculpting the plasticity of each connection is crucial.

Results


Non-plastic
Plastic
Homogenous Plastic
Reward
175
150
125
100
75
50
25
0
0
200000
400000
600000
800000
1000000
Number of Episodes

Future work

- Different application settings.
- Complement other recurrent structures.
- Plastic LSTMs can be considered.
- Neuromodulatory approaches can be explored using the framework designed in the paper.
- More complex RL settings.

Critiques

- Not a significant algorithmic contribution.
- Performances when test set don't match well with training set.
- Too many hyper parameters.
- Different plastic complements were considered without proper explanation.
- No comparisons to transfer learning or previous meta learning state of the art algorithms for most domains.
- Only performances in the pattern memorization task is better than the state of the art.

Any Questions?