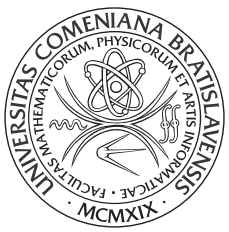
Comenius University in Bratislava
Faculty of Mathematics, Physics and Informatics



Mgr. Peter Gergeľ

Dissertation thesis summary

(Autoreferát dizertačnej práce)

Biologically Inspired Modeling of Artificial Neuro-Glial Networks

for the academic title philosophiae doctor
in the field of doctoral study: Computer Science, Informatics

Bratislava, 2019

The dissertation was conducted within the internal doctoral program at the Department of Applied Informatics at Faculty of Mathematics, Physics and Informatics of Comenius University in Bratislava.

Applicant: Mgr. Peter Gergeľ
Department of Applied Informatics
Faculty of Mathematics, Physics and Informatics
Comenius University in Bratislava
Mlynská dolina, 842 48 Bratislava

Tutor: prof. Ing. Igor Farkaš, Dr.
Department of Applied Informatics
Faculty of Mathematics, Physics and Informatics
Comenius University in Bratislava
Mlynská dolina, 842 48 Bratislava

Study programme: 9.2.1. - Computer Science, Informatics

Field committee chair:

prof. RNDr. Rastislav Kráľovič, PhD.
Department of Computer Science
Faculty of Mathematics, Physics and Informatics
Comenius University in Bratislava
Mlynská dolina, 842 48 Bratislava

1 Introduction

According to current knowledge of neuroscience, brain tissue consists of two cell populations: neurons and glia. The population of neurons is characterized by the ability to generate action potentials, whereas glia have been regarded as non-functional and supporting cells for several decades. Neurophysiological findings in the 1990s began to shift this view dramatically in a new direction, conveying evidence that glial cells are actively involved in modulation of neuronal excitability and synaptic plasticity, making them no longer merely passive cells.

Firstly identified in the 19th century, glial cells significantly contribute to the total brain mass with around 50% and the glia:neuron ratio in mammalian brains about 1:1 (Azevedo et al., 2009). The population of glia is commonly subdivided into four major groups: oligodendrocytes, microglia, ependymal cells and astrocytes. According to recent evidence, the first three types are closely specialized and account for myelination, immunity and cerebrospinal fluid production, respectively. Astrocytes, the most abundant and probably the most complex group, play a significant role in cognitive functions, traditionally attributed solely to neurons, such as learning and memory, information transfer and processing. Although not being able to be excited electrically and to generate action potentials as neurons do, they are incorporated in network called glial syncytium, where upon being excited chemically, they propagate Ca^{2+} signals through the gap junctions.

In order to better understand these low-level mechanisms, computational modelling is often employed which recently has become an essential part of neuroscience. Such models may provide testable predictions for processes that are built upon these mechanisms such as neuronal regulation, or synaptic plasticity. Better knowledge about astrocyte–neuron cooperation may also provide building blocks for studying the regulatory capability of glial syncytium on a larger scale. Computational models of artificial neural networks (ANNs) extended with astrocytes may not only be used as an interesting novel concept, but can mainly provide space for hypotheses to explain the potential roles of glia in biological neuronal circuits and networks.

The primary goal of the dissertation thesis is to study neuron–astrocyte coupling in connectionist systems. Inspired by contemporary evidence from neuroscientific research of astrocyte physiology and their interactions with surrounding neurons, and work by Ikuta et al. (2010), we propose several models of feedforward and recurrent ANNs and evaluate their performance in classification tasks and memory capacity.

2 Astrocytes in feedforward neural networks

Two major roles of astrocytes in ANNs are considered in general, either as neuronal regulators or as synaptic plasticity modulators. In this work we study solely the former function: the neuronal regulation. We start with a simplest model of a feedforward neural network (FFNN) and astrocyte per se, and we gradually move toward adding more complex mechanisms. But before we delve into networks with astrocyte units, we first provide brief

overview of multi-layer perceptron (MLP).

2.1 Brief introduction to MLP

MLP is type of ANN consisting of input, hidden and output layers of neurons, which are fully connected by weight matrices. The input vector $\boldsymbol{x}(t) = [x_1(t), ..., x_N(t)]$ is presented into input layer and the hidden layer activation is evaluated according to formula

$$h_i(t+1) = f(\sum_{j=0}^{N} w_{ij}^{hid} x_j(t)), \tag{1}$$

where w_{ij}^{hid} is the weight between j-th input neuron and i-th hidden neuron, and $\mathrm{f}^{\mathrm{res}}(\mathrm{net})$ is the suitable activation function. In our case we use logistic sigmoid defined as $\mathrm{f}^{\mathrm{res}}(\mathrm{net}) = 1/(1+\exp(-net))$. The computation of the output layer activation is given by very same formula, although the hidden layer and $\boldsymbol{W}^{\mathrm{out}}$ are considered instead.

It has been proven that MLP with a single hidden layer is capable of approximating any continuous function to any desired degree of accuracy (Hornik et al., 1989). The tradeoff for having superior representational power and efficiency is more complex and a time consuming training of the model, which includes, but is not limited to finding suitable input data encoding, selection of model parameters such as number of neurons, or type of activation functions. The most common approach for the training of MLPs are first-order iterative gradient based methods that require the definition of loss function to measure model's error in approximating the desired output. Usually the mean squared error (MSE) is considered with the following definition

$$E(t) = \frac{1}{2} \sum_{p=1}^{P} (\boldsymbol{d}^{(p)} - \boldsymbol{y}^{(p)}(t))^2 \tag{2}$$

Training per se consists of two phases. In forward pass the signal computation flows from the input layer through the hidden layers to the output layer and the decision of the output layer $\boldsymbol{y}$ is measured against the desired label $\boldsymbol{d}$. Upon computing the error for all P patterns, the backward pass is performed in which the error signal is propagated backwards from the output layer towards the input layer and successive adjustments are made to the synaptic weights of the network. Regarding the learning equations, both weight matrices (input–hidden $\boldsymbol{W}^{\mathrm{hid}}$ and hidden–output $\boldsymbol{W}^{\mathrm{out}}$) are updated by moving in the direction of the negative gradient (hence the name gradient descent):

$$\boldsymbol{W}(t+1) = \boldsymbol{W}(t) - \eta \frac{\partial E(t)}{\partial \boldsymbol{W}(t)} \tag{3}$$

2.2 Proposed models

2.2.1 Fixed-weights astrocytes in A-MLP

Since the human cortex contains on average 1.4 astrocytes for each neuron, we simplify this notion and present a model with the ratio of astrocyte to neuron being 1:1. We combine the hidden layer of an MLP with impulse astrocytes that listen to and modulate neuronal activity of hidden neurons. Each neuron is paired with a single astrocyte and each astrocyte regulates only one neuron. The output of i-th hidden neuron is given by the following formula

$$h_i(t+1) = f(\sum_{j=0}^{M} w_{ij} x_j(t) + \alpha \psi_i(t)) \tag{4}$$

and the astrocyte activity is modified according to

$$\psi_i(t) = \begin{cases} 1, & \text{if } \theta < h_i(t-1) \\ \gamma \psi_i(t-1), & \text{otherwise} \end{cases} \tag{5}$$

Each astrocyte contributes, with a weight α, to the activity of the hidden neuron. When the neuron output exceeds the given threshold θ, the astrocyte activation is set to 1 and then it starts to decay by a factor γ, where $0 < \gamma < 1$. Note that the model consists of three free hyperparameters (α, γ, θ) whose optimal values have to be found experimentally. Since each problem requires a different set of optimal parameters, finding them requires time-intensive computations. As we explain later, we try to solve these issues by replacing constant parameters with modifiable versions explained in following sections.

2.2.2 Dynamic weights in A-MLP(α)

The astrocytes regulate neurons in the hidden layer by the factor α being shared for all astrocytes, which is, nevertheless, not biologically plausible. Plethora of regulatory mechanisms are well described from the current research in biochemistry that includes neuronal excitation or inhibition by astrocytes (Fellin et al., 2006). For that reason we consider an individual weight α_i for each astrocyte which may be either excitatory or inhibitory.

Traditionally in supervised model learning, the neuron weights are updated using a gradient descent method, better known as error backpropagation algorithm. Since the astrocytic weight can be treated as any other weight, we can apply the same optimization method for its update yielding the formula

$$\Delta \alpha_i = -\delta_i \psi_i \tag{6}$$

where δ_i defines error of i-th hidden neuron. Moreover, instead of using a single mutual weight for all astrocytes, we equip each astrocyte unit with an individual weight α_i. Note that this formula is valid for the model with a single output neuron which is the case for our experiments. However, it is trivial to derive the rule for the model with multiple output neurons.

2.2.3 Dynamic threshold in A-MLP(θ)

In the case of other two free hyperparameters, γ and θ, since it is not straightforward to compute the derivation for the activation function written in eq. 5 with respect to γ and θ, we consider an alternative unsupervised rule.

Generally during training of ANNs it happens quite often that some neurons get trapped in one of the two extremes, by becoming either silent or permanently active. The gradient update of such neuronal weights is then problematic, because either the gradient is close to zero and therefore no errors would propagate through a silent neuron leading to no update of neuronal weight. On the other hand in permanently active neuron the weights might grow into large values, even infinite, leading to numerical problems, thus making the model unstable.

The same issue may happen in our model with artificial astrocytes when the threshold θ is set too low, making the astrocytes fire all the time. On the contrary, too high a value may prevent the neurons from exceeding the required threshold. This would however not advance into numerical problems, but the regulatory function of astrocytes would be lacking. Moreover, since each neuron in the neural network develops its own role in the classification task and for the same reason we explained in the previous section, we employ each astrocyte with a custom weight θ_i.

To incorporate unsupervised dynamic change of θs during training to accommodate for the change in neuronal behaviour, we propose an update rule with two variations. In order to stabilize the astrocytic regime, we can set the threshold θ either directly to the mean value $\langle . \rangle_t$ of an astrocyte unit (eq. 7) or only shift the threshold slightly closer to the mean value (eq. 8) using the learning speed η_θ. This forces the astrocyte to move only within its mean values avoiding the critical values of 0 and 1. With a higher θ it becomes harder for the neuron to overpass, thus the activity decays and vice versa. Hence, the update rules are

$$\theta_i(t+1) = \langle \psi_i(t) \rangle_t \tag{7}$$

and

$$\theta_i(t+1) = \theta_i(t) + \eta_\theta(\langle \psi_i(t) \rangle_t - \theta_i(t)) \tag{8}$$

introducing another free parameter, namely the length of an averaging window.

2.2.4 Dynamic activity decay in A-MLP(γ)

Hyperparameter γ can be updated based on the same principle as explained before. Now instead we update γ to achieve inverse correlation with the mean value of the astrocytic activity as (also two variations)

$$\gamma_i(t+1) = 1 - \langle \psi_i(t) \rangle_t \tag{9}$$

$$\gamma_i(t+1) = \gamma_i(t) + \eta_\gamma(1 - \langle \psi_i(t) \rangle_t - \gamma_i(t)) \tag{10}$$

Higher values of γ are achieved during a lower activity, thus a hypo-excited astrocyte holds its activation value for a longer period. Contrarily, the lower γ triggers faster activity decay forcing the astrocyte to avoid excessive simulation.

2.2.5 Combination of previous models: A-MLP(γ, θ), A-MLP(α, γ, θ)

The last two models are simple combinations of key ideas described in previous parts. Model A-MLP (γ, θ) combines dynamic thresholds, θs, and activity decays, γs. Model A-MLP(α, γ, θ) includes dynamic regulatory weights, αs, as well.

2.3 Experiments

To assess the performance of proposed networks with artificial astrocytes, we have chosen four classification tasks in our thesis: 1) two spirals, 2) nested circles, 3) chessboard, and 4) N-parity, although in this summary we consider only the two of them. First three datasets consist of two sets of two-dimensional points interleaved together with a high level of inseparability. This is considered difficult for a standard ANN due to a high number of potential local minima, which are generally rather problematic for gradient-based models.

We compare all results with the traditional MLP without astrocyte units which is used as a baseline. To eliminate the possibility that astrocyte units act as a simple random noise generator which is well described as a mean of regularization to avoid overfitting (Holmstrom and Koistinen, 1992; Bishop, 1995), we include in our comparison the model N-MLP, which simply extends the hidden layer activation with noise from the uniform distribution $Uni(-1, 1)$ – the very same interval that astrocytes produce.

In the first place for each dataset we found optimal hyperparameters using a grid search for MLP (number of hidden neurons, learning rate, weight initialization and number of epochs) that yielded the lowest MSE. Then we transferred the very same set of hyperparameters to all A-MLP models and searched for remaining hyperparameter values (individually for each model). Each dataset was randomly split to train/test set in the ratio 80:20. Results for the two spirals and the chessboard datasets can be seen in Fig. 1 and Fig. 2. In both cases the networks with astrocyte units outperform traditional MLP and MLP with random noise in terms of MSE and rate of convergence.

3 Astrocytes in recurrent neural networks

Whereas in a FFNN the information flows only in one direction, from the input layer directly to the output layer, in a recurrent neural network (RNN), the feedback connections are present allowing to pass and process the information typically in a loop. These feedback connections can take a variety of forms including feedback from the hidden layer to the input layer, feedback from the outer layer to the hidden layer, or alternatively feedback from the output layer directly to the input layer. A combination of multiple feedback types is also feasible.

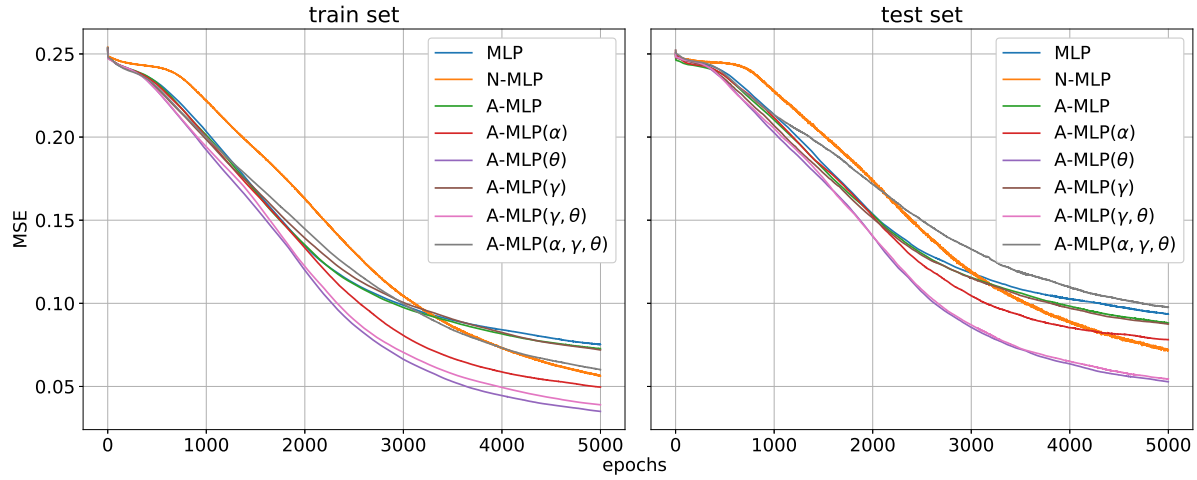


Figure 1: Learning curves for all models on the two spirals problem. Although the rate of convergence matches all other models, the final MSE is lower for each model with astrocyte units.

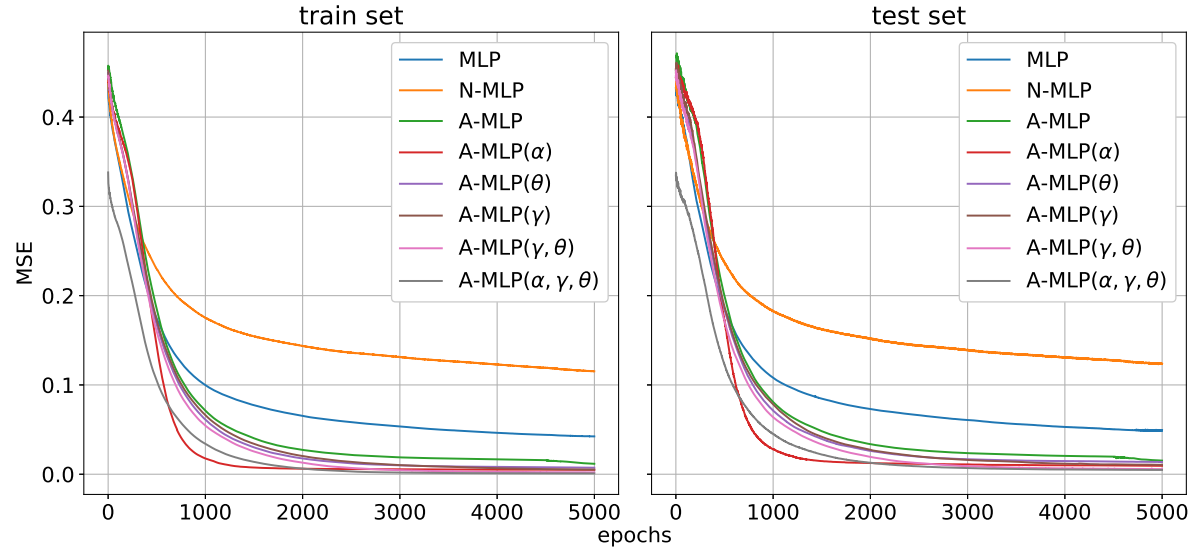


Figure 2: Learning curves for all models on the chessboard dataset. The difference in the performance of models with astrocyte units is notable compared to MLP and N-MLP.

Training traditional RNNs is considered to be difficult because of limitations of gradient descent methods which tend to be computationally expensive, to have slow convergence and to generally lead to poor local minima. Hence, the full adaption of all network weights is often omitted, yet still yielding excellent performance. This approach serves as a foundation for ESNs which were introduced by Jaeger (2001) for nonlinear system identification and time series modeling. ESNs are characterized by having randomly generated input weights

and reservoirs with the training only the readout weights.

However, in order to work well, ESNs require delicate tuning of several hyperparameters including the reservoir, the spectral radius ρ, and input weight scaling τ. ESN must have the *echo state property* which says that regardless of initial conditions, the hidden layer must converge to the same state given the same input signal. If this is met, only readout weights adaption is sufficient to obtain the ESN with high performance.

Reservoir activation vectors $\boldsymbol{x}(t) = [x_1(t), ..., x_N(t)]$ and output activations $\boldsymbol{y} = [y_1, ..., y_O]$ for given input pattern $\boldsymbol{u} = [u(1), ..., u(T)]$ are updated according to ESN dynamics given by the formulas

$$\boldsymbol{x}(t) = \mathrm{f}^{\mathrm{res}}(\boldsymbol{w}^{\mathrm{in}} u(t) + \boldsymbol{W}^{\mathrm{res}} \boldsymbol{x}(t-1)) \tag{11}$$

$$\boldsymbol{y}(t) = \mathrm{f}^{\mathrm{out}}(\boldsymbol{W}^{\mathrm{out}} \boldsymbol{x}(T)) \tag{12}$$

where $\mathrm{f}^{\mathrm{res}}$, $\mathrm{f}^{\mathrm{out}}$ are suitable activation functions, $\boldsymbol{w}^{\mathrm{in}}$ is the input weight vector, $\boldsymbol{W}^{\mathrm{res}}$ and $\boldsymbol{W}^{\mathrm{out}}$ are recurrent and output weight matrices, respectively. In our study we use $\mathrm{f}^{\mathrm{res}}(\mathrm{net}) = 1/(1 + \exp(-net))$ and $\mathrm{f}^{\mathrm{out}} = \boldsymbol{id}$.

3.1 Proposed models

3.1.1 Fixed-weights astrocytes in A-ESN

Here we propose a model of ESN augmented with the same model of astrocytes as described in Section 2.2.1. Similarly, we omit the concept of glial syncytium in which astrocytes are connected using gap junctions and communicate sharing slow Ca^{2+} signals (as opposed to neuronal firing), but we start studying the simplest possible model instead. We consider merely the role of neuronal regulation by astrocytes themselves and equip each reservoir neuron with one astrocyte.

Reservoir activation $x'_i(t)$ takes into account input pattern $u(t)$, previous time step activation vector $\boldsymbol{x}'(t-1)$ and astrocyte activation $\psi_i(t)$ weighted by a single shared weight w^α, which is expressed in the vector form as

$$\boldsymbol{x}'(t) = f(\boldsymbol{w}^{\mathrm{in}} u(t) + \boldsymbol{W}^{\mathrm{res}} \boldsymbol{x}'(t-1) + w^\alpha \boldsymbol{\psi}(t)) \tag{13}$$

Astrocytes $\psi_i(t)$ listen to their associated neurons and when some of the neurons exceed the threshold θ, astrocytes produce the activation value of 1. The rest of them decay by factor γ.

3.1.2 Hebbian-weights astrocytes in A-HL-ESN

Since using a single shared weight w^α for all astrocytes may be too constraining, we consider an individual weight for each astrocyte. Although astrocytes are not considered to be able to trigger neuronal action potential, they still modulate their membrane potential by the release of gliotransmitters including glutamate (exciting the neuron) or ATP (inhibiting

the neuron) as already stated in Section 2.2.2. For that reason we consider randomly generated weights from a uniform distribution $Uni(-1, 1)$.

The exact relationship of neuronal regulation by astrocytes is still not well understood and we can only guess to which extent is this process plastic and what are the specific mechanisms of plasticity. For that matter we speculate using Hebbian learning which is in great detail described in Hebb (1949). The basic principle is that the change of a synaptic weight w_{ji} between neurons x_i and y_j, with the learning rate η, is expressed as

$$\Delta w_{ji}(t) = \eta x_i(t) y_j(t) \tag{14}$$

In our case we apply this rule for the change of the weight w^α between a neuron x'_i and an astrocyte ψ_i. Repeated application, however, may lead to an exponential change of the weight which is not biologically plausible and this is solved by incorporating some form of stabilization. This is in many cases the normalization of the final weights. We consider Oja (1982)'s rule which introduces a nonlinear, forgetting factor for the weight change

$$\Delta w_i^\alpha(t+1) = \eta x'_i(t)[\psi_i(t) - x'_i(t) w_i^\alpha(t)] \tag{15}$$

To take into account this new dynamics, we split our training algorithm into two phases: 1) once the unsupervised learning of the weights $\boldsymbol{w}^\alpha$ (eq. 15) in the reservoir is complete, 2) a supervised learning algorithm is applied to the readout weights. Instead of using eq. 13 for the reservoir update, we consider

$$\boldsymbol{x}'(t) = f(\boldsymbol{w}^{\text{in}} u(t) + \boldsymbol{W}^{\text{res}} \boldsymbol{x}'(t-1) + \boldsymbol{w}^\alpha * \boldsymbol{\psi}(t)) \tag{16}$$

with operator '$*$' denoting the element-wise product of vectors.

3.2 Experiments

For both, classification tasks and memory capacity, we consider the following training procedure:

1. Generate random input weights $\boldsymbol{w}^{\text{in}}$ and reservoir weights $\boldsymbol{W}^{\text{res}}$ scaled by $\rho/|\lambda_{\max}|$, where $\lambda_{\max}$ denotes the largest absolute eigenvalue of $\boldsymbol{W}^{\text{res}}$ and ρ is manually selected.

2. Run ESN using the training inputs $\boldsymbol{u}_{train}$ and collect the required reservoir activation state $\boldsymbol{x}(t)$ (more precisely explained in the specific section for each task).

3. Compute the linear readout weights using formula

$$\boldsymbol{W}^{\text{out}} = \boldsymbol{Y}^{\text{tgt}} \boldsymbol{X}^{+} \tag{17}$$

   where $\boldsymbol{Y}^{\text{tgt}}$ is a matrix of concatenated target vectors (in columns) and $\boldsymbol{X}^{+}$ is the pseudoinverse matrix of concatenated reservoir activation states from step 2.

4. Use the trained network on new input data $\boldsymbol{u}_{test}$ and evaluate the performance.

3.2.1 Classification experiments

For the classification tasks we have decided for the UCR Time Series Classification Archive (Chen et al., 2015) which consists of 85 real world problems and is often used for benchmarking of Machine Learning models. In our study we selected 8 random datasets upon which we assess the performance of the proposed methods. We use a standard ESN (without astrocytes) as a baseline and compare it with models A-ESN and A-HL-ESN.

For the training of readout weights we use only the last activation state vector $\boldsymbol{x}(T)$ obtained by processing each training pattern $\boldsymbol{u}_{train} = [u(1), ..., u(T)]$. Target vectors are represented using one-hot encoding and have shape $\boldsymbol{y}^{\text{tgt}} = [y(1), ..., y(C)]$ where C is the number of output classes. After training the model, the class of a new input $\boldsymbol{u}_{test}$ is decided by selecting output neuron with maximum activation

$$\text{class}(\boldsymbol{u}_{test}) = \arg\max_k y_k \tag{18}$$

Allowing for possibility of imbalanced datasets in which one class is over-represented with the respect to the others, we use *Matthews correlation coefficient* (MCC) (Matthews, 1975) as a metrics for performance evaluation score rather than the mean-squared error, accuracy or F1-score which does not work well on imbalanced datasets. The value MCC $= 1$ corresponds to a perfect match between model predictions and observations, whereas -1 indicates total disagreement between the two.

Results in terms of MCC averaged over 100 simulations are presented in Table 1. It is clear that model with Hebbian connections, A-HL-ESN, significantly outperforms models ESN and A-ESN. Despite having more complex training procedure and thus higher time complexity, gain in terms of performance is clearly notable. Model with fixed connections, A-ESN, have yielded results equivalent to standard ESN (assuming correct settings of hyperparameters), although it is speculative why on the last dataset (ToeSegmentation1), the error rate is significantly better (MCC of 0.5 ± 0.1 vs 0.32 ± 0.11).

Dataset	ESN	A-ESN	A-HL-ESN
Earthquakes	0.20 ± 0.12	0.21 ± 0.12	0.24 ± 0.11
FaceFour	0.44 ± 0.12	0.43 ± 0.13	0.56 ± 0.14
MoteStrain	0.65 ± 0.04	0.67 ± 0.06	0.85 ± 0.03
OSULeaf	0.41 ± 0.06	0.42 ± 0.06	0.57 ± 0.06
PhalOutlCorr	0.37 ± 0.04	0.38 ± 0.04	0.43 ± 0.03
ProxPhalOutlCorr	0.48 ± 0.06	0.52 ± 0.07	0.53 ± 0.06
SwedishLeaf	0.64 ± 0.03	0.63 ± 0.03	0.84 ± 0.03
ToeSegmentation1	0.32 ± 0.11	0.50 ± 0.10	0.59 ± 0.11

Table 1: MCC (mean±standard deviation) averaged over 100 simulations on each dataset. In each case, the model A-HL-ESN is superior regarding the performance.

3.2.2 Memory capacity experiments

Memory capacity (MC) is defined by Jaeger (2002) as a measure of network's ability to reconstruct the past information from the reservoir on the network output by computing correlations. In our work (Farkaš et al., 2016; Farkaš and Gergel', 2017) we systematically investigated the effect of proper reservoir initialization on MC and proposed two gradient descent iterative methods that approach a maximum of theoretical limit of ESN's MC that drive the reservoir dynamics towards the critical regime (the transition zone between a stable and an unstable dynamics regime).

Jaeger defined the MC as

$$\mathrm{MC} = \sum_{k=1}^{k_{\max}} \mathrm{MC}_k = \sum_{k=1}^{k_{\max}} \frac{cov^2(u(t-k), y_k(t))}{var(u(t)) \cdot var(y_k(t))} \tag{19}$$

where functions *cov* and *var* denote covariance and variance, respectively. $u(t-k)$ is the input presented k-steps before the current input, $k_{max} = \infty$, and $y_k(t)$ is the reconstruction at the network output. The computation of MC is approximated using $k_{max} = O$ output neurons. The computation of MC takes into account the network ability to retrieve the past input signal (for various delays k) from the reservoir using the linear combinations of reservoir unit activations observed at the output (quantified by MC_k). Jaeger proved that the MC for recalling an independent, identically distributed (i.i.d) input by an ESN of N-units with identity activation function is bounded by N.

For assessing the performance of all models, we measure the total MC using randomly generated sequences $\boldsymbol{u} \in \mathbb{R}^T$ of length $T = 1100$, drawn from a uniform distribution, hence $u_i \in Uni(-1, 1)$. Such sequence has no underlying structure and is random: u_p and u_r are independent for $p \neq r$. We fed the first 100 inputs to the network to remove the initial transient which is normally not present once the network has "warmed up" to the task. Next 500 inputs to the network are utilized for the training, while collecting the reservoir activations $\boldsymbol{x}$ and target output vectors $\boldsymbol{y}^{\mathrm{tgt}}$ and storing them into the matrices $\boldsymbol{X}$ and $\boldsymbol{Y}^{\mathrm{tgt}}$, respectively. The optimal output weight matrix $\boldsymbol{W}^{out}$ is computed analytically according to eq. 17 and final 500 input patterns are used for assessing the MC.

Regarding the initialization, we use $N = 100$ neurons within a reservoir, initialize the input weights $\boldsymbol{w}^{in}$ from uniform distribution $Uni(-\tau, \tau)$ (τ = 1e−6), and the reservoir weight matrix $\boldsymbol{W}^{res}$ from normal distribution $\mathcal{N}(0, 1)$ scaled by $\rho/|\lambda_{\max}|$ ($\rho = 5.5$) as stated in step 1 of Section 3.2.

Similarly as in case of the classification experiments, we used a grid search for obtaining the best hyperparameters for models with astrocyte units. The values $\gamma = 1.0$ and $\theta = 0.6$ turned up to be optimal in both models. Regarding the A-ESN, we used $\alpha = -1.9$, implying extensive inhibitory role of *some* astrocytes as a result of relatively high threshold θ, which activates only few astrocytes. Upon being activated, their activity remains constant for the rest of input sequence.

A-HL-ESN, on the other hand, did not perform very well. By taking into account the fact that the weight vector $\boldsymbol{w}^{\alpha}$ remains within positive range, which contradicts weight

value for the previous model, out of curiosity we simply swapped the term $+\boldsymbol{w}^{\alpha} * \boldsymbol{\psi}(t)$ with the $-\boldsymbol{w}^{\alpha} * \boldsymbol{\psi}(t)$ in eq. 16, which turned out to perform better, however, it did not exceed MC of the simpler model, A-ESN. We call these models with excitatory and inhibitory contribution as A-HL-ESN^{+} and A-HL-ESN^{-}. The results are summarized in Table 2.

	ESN	A-ESN	A-HL-ESN^{+}	A-HL-ESN^{-}
Memory capacity	32.14 ± 3.13	40.03 ± 2.24	25.59 ± 2.32	39.94 ± 4.86

Table 2: Measure of MC on randomly generated sequences from uniform distribution on three models. The higher value signifies better performance, hence the best performing model is A-ESN with the static astrocytic weights.

Bibliography

Azevedo, F. A., Carvalho, L. R., Grinberg, L. T., Farfel, J. M., Ferretti, R. E., Leite, R. E., Filho, W. J., Lent, R., and Herculano-Houzel, S. (2009). Equal numbers of neuronal and nonneuronal cells make the human brain an isometrically scaled-up primate brain. *Journal of Comparative Neurology*, 513(5):532–541.

Bishop, C. M. (1995). Training with noise is equivalent to Tikhonov regularization. *Neural Computation*, 7(1):108–116.

Chen, Y., Keogh, E., Hu, B., Begum, N., Bagnall, A., Mueen, A., and Batista, G. (2015). The UCR time series classification archive. www.cs.ucr.edu/~eamonn/time_series_data/.

Farkaš, I., Bosák, R., and Gergel', P. (2016). Computational analysis of memory capacity in echo state networks. *Neural Networks*, 83:109–120.

Farkaš, I. and Gergel', P. (2017). Maximizing memory capacity of echo state networks with orthogonalized reservoirs. In *International Joint Conference on Neural Networks*, pages 2437–2442. IEEE.

Fellin, T., Pascual, O., and Haydon, P. G. (2006). Astrocytes coordinate synaptic networks: balanced excitation and inhibition. *Physiology*, 21(3):208–215.

Hebb, D. O. (1949). *The Organization of Behavior: A Neuropsychological Theory*. Wiley, New York.

Holmstrom, L. and Koistinen, P. (1992). Using additive noise in back-propagation training. *IEEE Transactions on Neural Networks*, 3(1):24–38.

Hornik, K., Stinchcombe, M., and White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366.

Ikuta, C., Uwate, Y., and Nishio, Y. (2010). Multi-layer perceptron with impulse glial network. *IEEE Workshop on Nonlinear Circuit Networks*, pages 9–11.

Jaeger, H. (2001). The "echo state" approach to analysing and training recurrent neural networks-with an erratum note. *Bonn, Germany: German National Research Center for Information Technology GMD Technical Report*, 148(34):13.

Jaeger, H. (2002). Short term memory in echo state networks. In *GMD-German National Research Institute for Computer Science)*. Citeseer. http://www.faculty.jacobs-university.de/hjaeger/pubs/STMEchoStatesTechRep.pdf.

Matthews, B. W. (1975). Comparison of the predicted and observed secondary structure of T4 phage lysozyme. *Biochimica et Biophysica Acta (BBA)-Protein Structure*, 405(2):442–451.

Oja, E. (1982). Simplified neuron model as a principal component analyzer. *Journal of Mathematical Biology*, 15(3):267–273.

Publications related to the dissertation thesis

Farkaš I., Bosák R., Gergeľ P. (2016). Computational analysis of memory capacity in echo state networks. *Neural Networks*, 83:109–120. (**13 citations**).

Farkaš I., Gergeľ P. (2017). Maximizing Memory Capacity of Echo State Networks with Orthogonalized Reservoirs. In *International Joint Conference on Neural Networks*, pages 2437–2442. IEEE. (**2 citations**).

Gergeľ, P. (2017). Improving the performance of impulse neuro-glial network. In *Kognícia a umelý život 2017*, pages 60–63. Vydavateľstvo Univerzity Komenského, Bratislava.

Gergeľ, P., Farkaš I. (2018). Investigating the Role of Astrocyte Units in a Feedforward Neural Network. In *International Conference on Artificial Neural Networks*, pages 73–83. Springer.

Gergeľ, P., Farkaš I. (2019). Echo State Networks with Artificial Astrocytes and Hebbian Connections. In *International Work-Conference on Artificial Neural Networks*. Springer. (accepted and to be published on 12.06.2019).

Unrelated publications

Gergeľ, P. (2014). Konekcionistické modelovanie učenia sa analógií. In *Proceedings of the Student Science Conference*, FMFI UK, pages 60–68.

Gergeľ P., Farkaš I. (2015). Connectionist modeling of part–whole analogy learning. In *Proceedings of the EuroAsianPacific Joint Conference on Cognitive Science*, pages 587–592. (**1 citation**).

Summary

The neuroscientific research for the last decades has highlighted the importance of glial cells in information processing context. Astrocytes regulate neuronal functionality in a variety of ways, particularly by maintaining the concentration of ions and neurotransmitters, by releasing gliotransmitters, and modulating both neuronal excitability and synaptic plasticity. However, limited amount of research has been done in the field of ANNs equipped with artificial astrocytes.

Inspired by Ikuta et al. (2010) and the subsequent work, as well as by recent findings from biological research of astrocyte physiology and their interactions with surrounding neurons, in our thesis we have proposed artificial astrocyte units to be integrated in feed-forward and recurrent neural networks. The role of astrocytes in both models is reduced to the regulation of neuronal excitability. The interaction is bidirectional and by listening to neural activity, astrocytes provide positive or negative feedback helping the neurons to stabilize.

In case of the FFNNs, the original model with astrocyte units consists of several hyperparameters including glial weight, threshold, attenuation factor, propagating range of astrocyte activation, refractory period and an activation decay that needs to be selected manually for each problem, which is a time-consuming and error-prone process. Since we found the model to be too complex to start with and it turned out to be challenging to obtain the very same results as Ikuta et al. did in their paper, we have simplified the model by omitting the concept of syncytium and kept astrocytes as individual units, not connected with each other within a single network, which in the end turned out to perform significantly better.

Instead of using a single constant glial weight for all astrocytes, we have proposed a gradient-descent method that updates the parameters along the negative gradient of the loss function for each astrocyte individually. For the threshold and the activation decay, we have introduced two unsupervised rules (eq. 7 and 8 for the threshold and eq. 9 and 10 for the decay) which sets the specific value according to the history of astrocytic activity. Since both rules turned out to perform practically the same, we used the first variation that updates the value directly to the averaging window of the astrocytic activity.

We have evaluated the performance of the proposed modifications on four classification problems: 1) two-spirals, 2) nested circles, 3) chessboard and 4) N-parity. For all problems we first selected an MLP with optimal hyperparameters found using an exhaustive grid search (the number of hidden neurons, the learning rate, initial weight distribution) and then used them in models with astrocyte units. The results obtained for N-parity did not outperform MLP, however all models already converged to the global minimum with zero classification error. In case of the first three tasks, all our models performed better in terms of the lower errors with statistical significance ($p < 0.001$).

Inspired by positive results from the FFNN, we transferred the same model of astrocytes to RNNs and explored their influence. Such models are closer to biological realism than FFNNs, because recurrent connections are critical and ubiquitous in the CNS. Since training recurrent neural networks is difficult for various problems, we considered ESNs

instead. In addition, we incorporated Hebbian learning for weights between astrocytes and their associated neurons. By systematic analysis of this new dynamics on eight classification tasks we found very little contribution of astrocytes with fixed weights, but in case of Hebbian learning the performance yielded significantly positive outcome. In case of the memory capacity, it is actually the fixed-weights model that outperforms traditional ESN and the model with Hebbian-weights.

Future research in this area may follow several directions. The activation function for the astrocyte, as formulated in eq. 5, is definitely not the only one and there are several varieties to be considered. Since Ca^{2+} signalling within glial syncytium operates on a much slower pace as opposed to neuronal firing, it may be beneficiary to incorporate this slow, temporal dynamics into astrocytic behaviour. Although our model of an artificial astrocyte includes slow decay, "firing", however, remained still instant. Despite focusing on the astrocytes as single separate units, it is possible to model glial syncytium and design an astrocytic network of astrocytes connected together, hence fulfilling the biologically plausible spatiotemporal dynamics. Last but not least, instead of modeling the regulation of neuronal excitability, it is possible to design models that also incorporate the rules for synaptic plasticity.