Contents

1 Theoretical background

1.1 General notion of artificial neural networks

Artificial neural networks are getting more and more attention over the last decades as advances in raw computational power make it possible to implement these computationally heavy algorithms as well as recent achievements in better then human performance in certain tasks. In the next paragraphs we provide basic theoretical concepts behind artificial neural networks.

1.1.1 Neural network model

Haykin [1] defines a neural network in the following way.

Definition 1 *A neural network is a massively parallel distributed processor made up of simple processing units, which has a natural property for storing experiential knowledge and making it available for use. It resembles the brain in two respects:*

1. *Knowledge is acquired by the network from its environment through a learning process.*

2. *Interneuron connection strengths, known as synaptic weight, are used to store acquired knowledge.*

The basic information-processing unit is called *neuron*. In mathematical terms the neuron k of a system consisting of N neurons, can be written as a pair of equations:

$$u_k = \sum_{j=1}^{m} w_{kj} x_j \tag{1}$$

$$y_k = \varphi(u_k + b_k) \tag{2}$$

where $x_1, x_2, \ldots, x_m$ are the *input signals*, $w_{k1}, w_{k2}, \ldots, w_{km}$ are *synaptic weights*, u_k is called *linear combiner output* due to the input signals, b_k is called *bias*, $\varphi(\cdot)$ is the *activation function*, and y_k is the output signal of the neuron.

There are two basic activation functions:

1. *Threshold function*

$$\varphi(v) = \begin{cases} 1 & \text{if } v \geq 0 \\ 0 & \text{if } v < 0 \end{cases} \tag{3}$$

2. *Sigmoid function*

   - logistic function

$$\varphi(v) = \frac{1}{1 + e^{-av}} \tag{4}$$

   - hyperbolic tangent function

$$\varphi(v) = \tanh(v). \tag{5}$$

1.1.2 Neural network as an approximation

One way to look at artificial neural networks is as a mapping

$$f : X \to Y$$

where X is the set of inputs and Y is the set of outputs. In the language of statistics we are dealing with nonparametric statistical inference. This view is strongly supported by an important theoretical result proved by Hornik [4], namely the universal approximation theorem.

Theorem 1 *Let*

$$\mathfrak{N}_k^{(n)}(\varphi) = \left\{ y : \mathbb{R}^k \to \mathbb{R} \mid y(x) = \sum_{i=1}^{n} \beta_i \varphi(\sum_{j=1}^{m} w_{ij} x_j - b_i) \right\}$$

denote a set of all functions implemented by a network with n hidden units and one output unit. If φ is continuous, bounded an nonconstant, then $\mathfrak{N}_k^{(n)}$ is dense in $C(X)$ for all compact subsets X of $\mathbb{R}^k$ ($C(X)$ is the space of all continuous functions on X).

Here we can observe some similarities to the Stone-Weierstrass theorem. The process of estimation of the synaptical weights w_{ij} for $i = 1, \ldots, n$ and $j = 1, \ldots, m$ is called *learning*.

1.1.3 Classes of neural networks

The neurons of a neural network can be organized into many different structures. Many have been already developed and successfully applied to different tasks. The structure is called *network architecture* and in general there are two fundamentally different classes of network architectures:

1. **Feedforward networks**

   The structure of neurons is organized in layers. There is an *input layer*, *hidden layer* and *output layer*. The synaptical weights are only between neurons of two concurrent layers and only in one direction. In other words the output signals from one layer serve serve as input signals to the neurons in the second layer. We say that the network is *fully connected* when every unit in each layer of the network is connected to every unit in the next layer. Otherwise we say that the network

is *partially connected*. In general there can be multiple hidden layers. Theorem 1 deals with a feedforward network with a single hidden layer. The standard learning algorithm for feedforward networks is called *backpropagation of error*. Basically it is a form of gradient based optimisation method.

2. **Recurrent networks**

   A recurrent neural network has at least one *feedback loop*. Feedback as is used is used in dynamical systems, that is when the output influences the input of an element of the system. In the setting of neural network let us consider a linear operators A and B, and an input-output relationship between two neurons

   $$y_k(n) = A[x_j^{'}(n)]$$

   and

   $$x_j^{'}(n) = x_j(n) + B[y_k(n)].$$

   Modifying these equations we get

   $$y_k(n) = \frac{A}{1 - AB}[x_j(n)].$$

   We call the term $\frac{A}{1-AB}$ an *closed-loop operator* of the system, and AB the *open-loop operator*. Generally the existence of feedback loops in the neural network make the process of learning very difficult.

1.2 Echo state network

Echo state networks belong to the class of recurrent neural networks. This type of architecture was first proposed by Jaeger [2]. In the next paragraphs we will provide the structure and some properties of an echo state network.

1.2.1 Architecture

The architecture consists of input layer, reservoir and output layer. The reservoir is a fully connected recurrent hidden layer. The update equations are given by

$$\bar{\mathbf{x}}(t) = \tanh(\mathbf{W}^{in}[1; \mathbf{u}(t)] + \mathbf{W}\mathbf{x}(t-1)) \tag{6}$$

$$\mathbf{x}(t) = (1 - \alpha)\mathbf{x}(t-1) + \alpha\bar{\mathbf{x}}(t) \tag{7}$$

where $\mathbf{x}(t) \in \mathbb{R}^{N_x}$ is vector of reservoir neuron activations and $\bar{\mathbf{x}}(t) \in \mathbb{R}^{N_x}$ is its update at time step t, $\tanh(\cdot)$ is the activation function, $\mathbf{W}^{in} \in \mathbb{R}^{N_x \times (1+N_u)}$ is the input weight matrix, $\mathbf{W} \in \mathbb{R}^{N_x \times N_x}$ is the recurrent weight matrix, and $\alpha \in (0, 1]$ is the leaking rate. The output is defined as

$$\mathbf{y}(t) = \mathbf{W}^{out}[1; \mathbf{u}(t); \mathbf{x}(t)] \tag{8}$$

where $\mathbf{y}(t) \in \mathbb{R}^{N_y}$ is network output and $\mathbf{W}^{out} \in \mathbb{R}^{N_y \times (1+N_u+N_x)}$ is the output weight matrix.

Lukoševičius [5] writes that the reservoir serves two functions:

1. as a high-dimensional nonlinear expansion, similarly to kernel methods, where input in input space is not linearly separable and by projecting in a higher dimensional space it may become linearly separable.

2. as a dynamical short-term memory.

1.2.2 Echo state property

Jaeger in [2] defined the term *echo state property*, which is a necessary condition for the echo state network to work. The definition for a network with no output feedback is as follows:

Definition 2 *Assume that input is drawn from a compact input space U and network states lie in a compact set A. Assume that the network has no output feedback connections. Then, the network has echo states, if the network state $\mathbf{x}(n)$ is uniquely determined by any left-infinite input sequence $\bar{\mathbf{u}}^{-\infty}$. More precisely, this means that for every input sequence $\ldots, \mathbf{u}(n-1), \mathbf{u}(n) \in \mathrm{U}^{-\mathbb{N}}$, for all state sequences $\ldots, \mathbf{x}(n-1), \mathbf{x}(n)$ and $\mathbf{x}'(n-1), \mathbf{x}'(n) \in \mathrm{A}^{-\mathbb{N}}$, where $\mathbf{x}(i) = \mathrm{T}(\mathbf{x}(i-1), \mathbf{u}(i))$ and $\mathbf{x}'(i) = \mathrm{T}(\mathbf{x}'(i-1), \mathbf{u}(i))$, it holds that $\mathbf{x}(n) = \mathbf{x}'(n)$.*

1.2.3 Hyperparameters

The specific instance of an echo state network is defined by a set of parameters, namely $(\mathbf{W}^{in}, \mathbf{W}, \alpha)$. In the next paragraphs we will outline the principles for choosing these parameters. We are going to follow the recommendations according to Lukoševičius [5].

1. **Reservoir matrix W**

- *Size of reservoir* - the consensus is that the larger number of neurons in the reservoir the better, but on the other there is danger of over-parametrization and over-fitting. In general

$$T >= 1 + N_x + N_u$$

where T is the size of the dataset, N_x is the number of neurons in reservoir, and N_u is the dimension of the input.

- *Distribution of reservoir weights* - at the initialisation step of the algorithm the elements of the matrix $\mathbf{W}$ are randomly generated and not much manipulated afterwards. The weights are drawn independently from uniform or normal distribution symmetrical around zero.
- *Spectral radius* - as was stated in section 1.2.2 the existence of the echo state property in an echo state network is essential for the network to work. Jaeger [2] states the network has no echo state when the spectral radius of the reservoir matrix is $|\lambda_{max}(\mathbf{W})| > 1$, the admissible state set is $[-1, 1]^N$ and if the input set contains 0. Thus is standard practice to scale the reservoir matrix $\mathbf{W}$ so that $|\lambda_{max}(\mathbf{W})| < 1$. The specific value $|\lambda_{max}(\mathbf{W})|$ has to determined experimentally in way that maximizes performance.

2. **Input matrix $\mathbf{W}^{in}$**

   Similarly to the initialization of the reservoir matrix $\mathbf{W}$, the elements of input matrix $\mathbf{W}^{in}$ randomly selected either from uniform or normal symmetrical distribution.

3. **Leaking rate α**

   Assume a dynamical system

$$\dot{\mathbf{x}} = -x + \tanh(\mathbf{W}^{in}[1; \mathbf{u}] + \mathbf{W}\mathbf{x})$$

   and using a discretization of the system

$$\dot{\mathbf{x}} \approx \frac{\mathbf{x}(t) - \mathbf{x}(t-1)}{\triangle t}$$

   we get

$$\mathbf{x}(t) = (1 - \triangle t)\mathbf{x}(t-1) + \triangle t \tanh(\mathbf{W}^{in}[1; \mathbf{u}(t)] + \mathbf{W}\mathbf{x}(t-1)). \quad (9)$$

By comparing equations (7) and (9) we can regard the leaking rate parameter as the size of the discrete time step and scale accordingly. In practice this parameter has to be determined experimentally.

1.2.4 Learning process

As can be seen from previous sections there hasn't been any training involved, yet. The the only training is in in finding the output layer weight matrix $\mathbf{W}^{out}$. Precisely solving

$$\mathbf{Y}^{target} = \mathbf{W}^{out}\mathbf{X} \tag{10}$$

where $\mathbf{Y}^{target} \in \mathbb{R}^{N_y \times T}$, $\mathbf{X} \in \mathbb{R}^{(1+N_u+N_x)\times T}$ and T the size of the training set. Lukoševičius [5] proposes to use regression with Tikhonov regularization

$$\mathbf{W}^{out} = \mathbf{Y}^{target}\mathbf{X}^{\mathrm{T}}(\mathbf{X}\mathbf{X}^{\mathrm{T}} + \beta\mathbf{I})^{-1} \tag{11}$$

where $\beta \in \mathbb{R}$ is the regularization parameter and has to be determined experimentally. When $\beta = 0$ we get

$$\mathbf{W}^{out} = \mathbf{Y}^{target}\mathbf{X}^{+} \tag{12}$$

where $\mathbf{X}^{+} = \mathbf{X}^{\mathrm{T}}(\mathbf{X}\mathbf{X}^{\mathrm{T}})^{-1}$ is the right Moore-Penrose inverse.

1.2.5 Short term memory capacity

As has been already stated, due to the feedback loops, echo state networks can be regarded as a dynamic short-term memory. Jaeger [3] defined a measure to quantify the reservoirs ability to store and recall past input. It is called the short-term memory capacity:

Definition 3 *Let $\nu(n) \in \mathrm{U}$ (where $-\infty < n < +\infty$ and $\mathrm{U} \in \mathbb{R}$ is a compact interval) be a single-channel stationary input signal. Assume that we have a recurrent neural network, specified by its internal weight matrix $\mathbf{W}$, its input weight (column) vector $\mathbf{w}^{in}$ and the unit output functions $\mathbf{f}, \mathbf{f}^{out}$. The network receives $\nu(n)$ at its input unit. For a given delay k and an output unit y_k with connection weight (row) vector $\mathbf{w}_k^{out}$ we consider the determination coefficient*

$$\begin{aligned} & d[\mathbf{w}_k^{out}](\nu(n-k), y_k(n)) = \\ & = d\left(\nu(n-k), \mathbf{w}_k^{out}\begin{pmatrix}\nu(n)\\ \mathbf{x}(n)\end{pmatrix}\right) \\ & = \frac{cov^2(\nu(n-k), y_k(n))}{\sigma^2(\nu(n))\sigma^2(y_k(n))} \end{aligned} \tag{13}$$

where cov denotes covariance and σ^2 variance.

1. *The k-delay short term memory capacity of the network is defined by*

$$MC_k = \max_{\mathbf{w}_k^{out}} d[\mathbf{w}_k^{out}](\nu(n-k), y_k(n)). \tag{14}$$

2. *The short term memory capacity of the network is*

$$MC = \sum_{k=1}^{\infty} MC_k. \tag{15}$$

Jaeger in [3] also proved a theoretical limit for the short-term memory capacity:

Proposition 1 *The memory capacity for recalling i.i.d input by a N-unit recurrent neural network with linear output units is bounded by N.*

1.2.6 Lyapunov characteristic exponent

The reservoir of an echo state network can be looked at as a discrete dynamical system and the performance of the the whole network depends on its stability. A popular measure of the degree of the instability of a dynamical system is called *the maximum Lyapunov characteristic exponent.*

Definition 4 *Let*

$$\mathbf{x}(t+1) = \mathbf{f}(\mathbf{x}(t)) \tag{16}$$

be a discrete time dynamical system and

$$\mathbf{x}^{'}(0) = \mathbf{x}(0) + \delta\mathbf{x}(0) \tag{17}$$

be the initial conditions a trajectory $\mathbf{x}^{'}(t)$ obtained by an infinitesimal displacement from $\mathbf{x}(0)$ such that $\gamma_0 = \|\delta\mathbf{x}(0)\| \ll 1$. Then the maximum Lyapunov characteristic exponent is

$$\lambda = \lim_{t\to\infty} \frac{1}{t} \ln\left(\frac{\gamma_t}{\gamma_0}\right) < \infty \tag{18}$$

where $\gamma_t = \|\mathbf{x}^{'}(t) - \mathbf{x}(t)\|$.

From the approximation

$$\gamma_t \sim \gamma_0 e^{\lambda t} \tag{19}$$

it it can be seen that when $\lambda > 0$ the system is sensitive to initial conditions and therefore is chaotic. For $\lambda < 0$ the system is sub-critical. The value $\lambda \approx 0$ is when phase transition occurs and is often referred to as the edge of criticality or critical point.

For numerical computation of λ we implement a popular method according to Cencini et.al [6].

1. We start with the infinitesimal perturbation γ_0 and evolve to system one time step ahead, thus getting a normalized tangent vector $\mathbf{w}(1) = \frac{\mathbf{x}'(1)-\mathbf{x}(1)}{\gamma_0}$ and setting $\alpha(1) = \|\mathbf{w}(1)\|$.

2. Next we rescale $\mathbf{w}(1)$ to $\frac{\mathbf{w}(1)}{\|\mathbf{w}(1)\|}$ and evolve system one step ahead again. We repeat this process and store the amplitudes $\alpha(t) = \|\mathbf{w}(t)\|$.

3. The maximum Lyapunov exponent is obtained as:

$$\lambda = \lim_{t\to\infty} \frac{1}{t} \sum_{i=1}^{t} \ln(\alpha(i)). \tag{20}$$

In the echo state network setting we set the infinitesimal perturbation to every neuron unit individually and compute λ_i of the $i-th$ perturbed for every $i = 1, \ldots, N$ of the N unit reservoir. The final estimated is computed as average $\lambda =< \lambda_i >$.

1.2.7 Orthogonalization and orthonormalization of reservoir

We have already mentioned reservoir matrix scaling according to the desired spectral radius. Another ways to scale the reservoir matrix is in relation to orthogonality. More precisely to satisfy the condition $\mathbf{W}^\top\mathbf{W} = \mathbf{I}$. Farkaš et. al. [7] proposed a gradient based orthogonalization and orthonormalization methods and have shown that the theoretical limits in the memory capacity task, proved by Jaeger [3], can be achieved via this methods.

The update formula in case of the orthogonalization method is

$$\triangle\mathbf{w}_i = -\eta\frac{4}{\|\mathbf{w}_i\|}(\mathbf{I} - \tilde{\mathbf{w}}_i\tilde{\mathbf{w}}_i^\top)(\tilde{\mathbf{W}}\tilde{\mathbf{W}}^\top)\tilde{\mathbf{w}}_i \tag{21}$$

where η is the learning rate and $\tilde{\mathbf{W}}$ denotes matrix $\mathbf{W}$ with normalized columns.

The update formula for the orthonormalization method is

$$\triangle \mathbf{w}_i = -\eta \; 4 \; (\mathbf{W}\mathbf{W}^\top \mathbf{W} - \mathbf{W}). \tag{22}$$

1.3 Information measures

1.3.1 Introduction to information theory

In order to get an insight into transfer entropy and active information storage we have to define some key information theoretic concepts. We will follow the definitions according to MacKay [8]. At first we define the Shannon information content and entropy of a discrete random variable X.

Definition 5 *Shannon information content of an outcome x is defined to be*

$$h(x) = \log_2 \frac{1}{P(x)}. \tag{23}$$

The units are called bits.

Definition 6 *The entropy of an ensemble X is defined to be the average Shannon information content of an outcome:*

$$\begin{aligned} H(X) &= \sum_{x \in \mathcal{A}_X} P(x) \log \frac{1}{P(x)} \quad \textit{for } P(x) \neq 0 \\ &= 0 \qquad\qquad\qquad\qquad\quad \textit{for } P(x) = 0 \end{aligned} \tag{24}$$

since $\lim_{\theta \to 0^+} \theta \log \frac{1}{\theta} = 0$.

Next we need to define conditional entropy.

Definition 7 *The conditional entropy of X given $y = b_k$ is the entropy of the probability distribution $P(x \mid y = b_k)$.*

$$H(X \mid y = b_k) = \sum_{x \in \mathcal{A}_X} P(x \mid y = b_k) \log \frac{1}{P(x \mid y = b_k)} \tag{25}$$

Definition 8 *The conditional entropy of X given Y is average, over y, of the conditional entropy of X given y.*

$$H(X \mid Y) = \sum_{y \in \mathcal{A}_y} P(y) \left[\sum_{x \in \mathcal{A}_X} P(x \mid y) \log \frac{1}{P(x \mid y)} \right] \tag{26}$$
$$= \sum_{x \in \mathcal{A}_X, y \in \mathcal{A}_Y} P(x, y) \log \frac{1}{P(x \mid y)}.$$

The interpretation of conditional transfer entropy is that it measures the average uncertainty that remains about X when Y is known.

Now we can define mutual information and conditional mutual information.

Definition 9 *The mutual information between X and Y is*

$$I(X;Y) = H(X) - H(X \mid Y). \tag{27}$$

Mutual information measures the reduction of uncertainty about X that results from learning the value of y.

Definition 10 *The conditional mutual information between X and Y given $z = c_k$ is the mutual information between the random variables X and Y in the joint ensemble $P(x, y \mid z = c_k)$,*

$$I(X;Y \mid z = c_k) = H(X \mid z = c_k) - H(X \mid Y, z = c_k). \tag{28}$$

Definition 11 *The conditional mutual information between X and Y given $z = c_k$ is the mutual information between the random variables X and Y in the joint ensemble $P(x, y \mid z = c_k)$,*

$$I(X;Y \mid z = c_k) = H(X \mid z = c_k) - H(X \mid Y, z = c_k). \tag{29}$$

Definition 12 *The conditional mutual information between X and Y given Z is the average over z of the conditional mutual information from Definition 11.*

$$I(X;Y \mid Z) = H(X \mid Z) - H(X \mid Y, Z). \tag{30}$$

Entropy and all other measures based on entropy we have just introduced are defined for a discrete set of probabilities. Shannon in [9] analogously defined the entropy of a continuous distribution with the density distribution function $p(x)$ as

$$H(X) = \int_{-\infty}^{\infty} p(x) \log \frac{1}{p(x)} dx. \tag{31}$$

This quantity is sometimes called *differential entropy* although this is not a measure of uncertainty of the random variable X as Shannon intended it to be.

Similarly, from the identity

$$I(X;Y) = \sum_{x \in \mathcal{A}_X, y \in \mathcal{A}_Y} P(x,y) \log \frac{P(x,y)}{P(x)P(y)} \tag{32}$$

we can define mutual information of a continuous variable as:

$$I(X;Y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} p(x,y) \log \frac{p(x,y)}{p(x)p(y)} dxdy. \tag{33}$$

1.3.2 Limiting density of discrete points

As it has been already noted in the previous section, Shannon's formulation of entropy for continuous variables is not an information measure. First it lacks many properties of discrete entropy and second it is not a result of any proper derivation. Nevertheless differential entropy has many theoretical applications.

Jaynes in [19] proposed a different approach to defining entropy of a continuous variable. Let $x_i, i = 1, \ldots, n$ be discrete points such that

$$\lim_{n \to \infty} \frac{1}{n} (\text{number of points in a} < x < \text{b}) = \int_a^b m(x) dx \tag{34}$$

exists. Then the differences $(x_{i+1} - x_i)$ in the neighbourhood of any particular value of x will tend to zero so that

$$\lim_{n \to \infty} n(x_{i+1} - x_i) = m(x_i)^{-1}. \tag{35}$$

The discrete probability $P(x_i)$ from Definition 6 will transform to

$$P(x_i) = p(x_i)(x_{i+1} - x_i). \tag{36}$$

Equivalently from equation (35)

$$P(x_i) \to p(x_i)\frac{1}{nm(x_i)}. \tag{37}$$

Plugging (36) and (37) into the definition of Shannon entropy Definition 6 we get

$$H(X) \to -\int p(x)log\left[\frac{p(x)}{nm(x)}\right]dx. \tag{38}$$

Following this reasoning, Jaynes defined the continuous information measure as:

$$\overline{H}(X) = \lim_{n\to\infty}[H(X) - \log(n)] = -\int p(x)log\left[\frac{p(x)}{m(x)}\right]dx. \tag{39}$$

1.3.3 Transfer entropy and active information storage

Let $\{x(t)\}$ and $\{y(t)\}$ be the realizations of two processes $\{X(t)\}$ and $\{Y(t)\}$. Paluš in [10] proposed, as an approach to measure the directional information rate, to measure conditional mutual information $I(y(t); x(t+\tau) \mid x(t))$. It is the average amount of information contained in the process $\{Y(t)\}$ about the process $\{X(t)\}$ in its future τ time units ahead conditioned on $X(t)$, as opposed to the mutual information $I(y(t); x(t+\tau)$ which can contain information about $X(t+\tau)$ in $X(t)$. Similarly Shreiber in [11] defines mutual information rate of two Markov processes $\{I\}$ and $\{J\}$ of order k and l by measuring the deviation from independence given by the Markov property $p(i_{t+1} \mid i_t^{(k)}) = p(i_{t+1} \mid i_t^{(k)}, j_t^{(l)})$ using conditional Kullback-Leibler divergence in the following form

$$D_{KL}(P(I_{t+1} \mid I_t^{(k)}, J_t^{(l)}) \| P(I_{t+1} \mid I_t^{(k)}, J_t^{(l)})) = \\ = \sum p(i_{t+1}, i_t^{(k)}, j_t^{(l)}) \log \frac{p(i_{t+1} \mid i_t^{(k)}, j_t^{(l)})}{p(i_{t+1} \mid i_t^{(k)})}.$$

These ideas inspired the current definition of transfer entropy.

Definition 13 *Let X_t and Y_t be two processes. The transfer entropy from the source Y with the history length k to the target X with history length l is*

$$T_{Y\to X}^{(k,l)} = I(X_t\ ; \mathbf{Y}_{t-1}^{(l)} \mid \mathbf{X}_{t-1}^{(k)}) \tag{40}$$

where

$$\mathbf{X}_{t-1}^{(k)} = (X_{t-1}, X_{t-1-\tau_k}, X_{t-1-2\tau_k}, ..., , X_{t-1-(k-1)\tau_k})$$
$$\mathbf{Y}_{t-1}^{(l)} = (Y_{t-1}, Y_{t-1-\tau_l}, Y_{t-1-2\tau_l}, ..., , Y_{t-1-(l-1)\tau_l}).$$

From the identity for conditional mutual information

$$I(X;Y \mid Z) = H(X) - I(X;Z) - H(X \mid Y,Z)$$

it follow that transfer entropy can be expressed as

$$T_{Y\to X}^{(k,l)} = H(X_t) - I(X_t \mid \mathbf{X}_{t-1}^{(k)}) + H(X_t \mid \mathbf{Y}_{t-1}^{(l)}, \mathbf{X}_{t-1}^{(k)}).$$

The second term on the right in the expression above has a special significance for us.

Definition 14 *The active information storage of the process X with history length k is*

$$A_X^{(k)} = I(\mathbf{X}_{t-1}^{(k)}; X_t) = H(X_t) - H(X_t \mid \mathbf{X}_{t-1}^{(k)}). \tag{41}$$

The active information storage measures the amount of information in the past state of $\mathbf{X}_t^{(k)}$ of X about its next value X_t.

1.3.4 Transfer entropy and Granger causality

Transfer entropy is closely related to another concept of dependency, namely Granger causality. This relationship provides a different interpretation of the concept of transfer entropy. Lets look take a closer look at it.

We take the definition of Granger causality from [12]

Definition 15 *Let us use the notation from Definition 13. Let $F(x_t|\mathbf{x}_{t-1}^{(k)}, \mathbf{y}_{t-1}^{(l)})$ be the distribution function of the target variable X_t conditional on $\mathbf{X}_{t-1}^{(k)}$, $\mathbf{Y}_{t-1}^{(l)}$ and $F(x_t|\mathbf{x}_{t-1}^{(k)})$ be the distribution function of X_t conditional on its own past, then variable Y is said to Granger-cause variable X if*

$$F(x_t|\mathbf{x}_{t-1}^{(k)}, \mathbf{y}_{y-1}^{(l)}) \neq F(x_t|\mathbf{x}_{t-1}^{(k)}). \tag{42}$$

Now, consider tow linear regression models

$$X_t = X_{t-1}A_1 + \ldots + X_{t-k}A_k + Y_{t-1}B_1 + \ldots + Y_{t-l}B_l + \epsilon_t \tag{43}$$

$$X_t = X_{t-1}A_1^{'} + \ldots + X_{t-k}A_k^{'} + \epsilon_t^{'} \tag{44}$$

with parameters of the models $A_i, A_i^{'}, B_j^{'}$. Geweke in [17] defined the measure of Granger causality from Y to X the following way,

$$F_{Y\to X}^{(k,l)} = \log(|\mathrm{var}(\epsilon_t^{'})|/|\mathrm{var}(\epsilon_t)|) \tag{45}$$

where $|\cdot|$ denotes the determinant. We can observe that this is actually log-likelihood ratio test under the null hypothesis

$$H_0 : B_1 = \cdots = B_l = 0 \tag{46}$$

when the residuals of the both model is gaussian.

From what has already been written it can be seen that both transfer entropy and Granger causality are measures of predictive causality. The similarity is even closer when we consider the joint process X_t, Y_t as multivariate gaussian. Barnett et. al. in [18] proved that under these conditions that Granger causality and transfer entropy are equivalent up to factor 2 i.e.

$$F_{Y\to X}^{(k,l)} = 2T_{Y\to X}^{(k,l)}. \tag{47}$$

1.3.5 Estimation of transfer entropy and active information storage

Estimation of entropy measures is an open problem. Currently there are few available classes of estimators that one can choose from, based on the properties of the observed data. Transfer entropy is no exception and the best estimator given some specific criteria is yet to be determined. An overview of transfer entropy estimators can be found in [12] or [13].

We are going to focus on one specific estimator of the class of estimators based on k-nearest neighbour search.

Kozachenko-Leonenko Shannon entropy estimator We are going to put the basic idea of behind Kozachenko-Leonenko differential entropy estimator as was presented in [14].

Let X be a continuous random variable, $f(x)$ be its density and its differential entropy as defined in (9). Specifically

$$H(X) = \int_{-\infty}^{\infty} f(x) \log \frac{1}{f(x)} dx. \tag{48}$$

Then the Monte-Carlo estimate of $H(X)$ is

$$\widehat{H}(X) = \frac{1}{N} \sum_{i=1}^{N} \log \frac{1}{f(x_i)}. \tag{49}$$

Since we don't know $f(x_i)$ it has to be substituted by an estimate $\widehat{f}(x_i)$ which we are going to find using k-nearest neighbours of x_i.

Let $P_k(\epsilon)$ be the probability distribution of the distance between x_i and its kth nearest neighbour. Then $P_k(\epsilon)d\epsilon$ is the probability that there is one point in the r distance from x_i where $r \in < \frac{\epsilon}{2}; \frac{\epsilon}{2} + \frac{d\epsilon}{2} >$, $k-1$ points are at distances less than r and $N-k-1$ points are at distances greater than kth nearest neighbour. Using multinomial distribution formula we get

$$P_k(\epsilon)d\epsilon = \frac{(N-1)!}{1!(k-1!)(N-k-1)!} \left(\frac{dp_i(\epsilon)}{d\epsilon} d\epsilon \right) (p_i(\epsilon))^{k-1} (1-p_i(\epsilon))^{N-k-1} \tag{50}$$

where $p_i(\epsilon)$ is the probability mass of ϵ ball centered at x_i, that is

$$p_i(\epsilon) = \int_{\|\xi - x_i\| < \frac{\epsilon}{2}} f(\xi) d\xi. \tag{51}$$

It follows from (50) and (51) that the expectation value $\log p_i(\epsilon)$ is given by

$$\begin{aligned} E(\log p_i(\epsilon)) &= \int_0^{\infty} \log p_i(\epsilon) P_k(\epsilon) d\epsilon \\ &= \psi(k) - \psi(N) \end{aligned} \tag{52}$$

where $\psi(x)$ is the digamma function.

If we assume that $f(x)$ is constant in the entire ϵ ball we can approximate $p_i(\epsilon)$ by

$$p_i(\epsilon) \approx c_d \epsilon^d f(x_i), \tag{53}$$

where d is the dimension of x and c_d is the volume of the d-dimensional unit ball. For the maximum norm $c_d = 1$.

Finally taking the logarithm and expectation of (53), and combining it with (52) and (49) we get Kozachenko-Leonenko entropy estimator

$$\widehat{H}(X) = -\psi(k) + \psi(N) + \log c_d + \frac{d}{N}\sum_{i=1}^{N} \log \epsilon(i), \tag{54}$$

where $\epsilon(i)$ is twice the distance from x_i to its kth nearest neighbour.

Kraskov-Stögbauer-Grassberger mutual information estimator Kraskov, Stögbauer and Grassberger in [14] came with a method of using Kozachenko-Leoneko entropy estimator to estimate mutual information. Using the identity

$$I(X,Y) = H(X) + H(Y) - H(X,Y) \tag{55}$$

we are going to obtain an estimator $\widehat{I}(X,Y)$.

First we directly apply Kozachenko-Leonenko entropy estimator to joint random variable $Z = (X,Y)$ with maximum norm and we get

$$\widehat{H}(X,Y) = -\psi(k) + \psi(N) + \frac{d_X + d_Y}{N}\sum_{i=1}^{N} \log \epsilon(i) \tag{56}$$

where $\epsilon(i)$ is the $\frac{\epsilon}{2}$ from z_i to its kth nearest neighbour and $d_Z = d_X + d_Y$.

For the estimate of $H(X)$ we take the distance $\epsilon(i)$ from (56) as an approximation of $[n_x(i)+1]$st nearest neighbour of x_i, where $n_x(i)$ is the number of points in within $\|x_j - x_i\| < \frac{\epsilon}{2}$, and we get

$$\widehat{H}(X) = -\frac{1}{N}\sum_{i=1}^{N} \psi(n_x(i)+1) + \psi(N) + \frac{d_X}{N}\sum \log \epsilon(i). \tag{57}$$

Analogously for the marginal space Y we get

$$\widehat{H}(Y) = -\frac{1}{N}\sum_{i=1}^{N} \psi(n_y(i)+1) + \psi(N) + \frac{d_Y}{N}\sum_{i=1}^{N} \log \epsilon(i). \tag{58}$$

Combining (55), (56), (57) and (58) we get the Kraskov, Stögbauer and Grassberger estimator

$$I^{(1)}(X,Y) = \psi(k) - \langle \psi(n_x+1) + \psi(n_y+1) \rangle + \psi(N) \tag{59}$$

where $\langle \cdots \rangle = \frac{1}{N}\sum_{i=1}^{N}(\cdots)$.

Estimating transfer entropy and active information storage estimator Since active information storage of a process is defined in Definition 14 as mutual information between its current and past state, it follows from (59) that the active information estimator is written as

$$\widehat{A}_X^{(k)} = \psi(k) + \psi(N) - \left\langle \psi(n_{\mathbf{x}_{t-1}^{(k)}} + 1) + \psi(n_{x_t} + 1) \right\rangle. \tag{60}$$

As for an estimate of transfer entropy Gomez-Herrero et. al. [15] generalized the idea of Kraskov et. al. [14]. Let $V = (V_1, \ldots, V_m)$ be a random m-dimensional vector. Then the entropy combination is defined as

$$C(V_{\mathcal{L}_1}, \ldots, V_{\mathcal{L}_p}) = \sum_{i=1}^{p} s_i H(V_{\mathcal{L}_1}) - H(V) \tag{61}$$

where $\forall i \in < i, p >$: $\mathcal{L}_i \subset < 1, m >$ and $s_i \in \{-1, 1\}$ such that $\sum_{i=1}^{p} s_i \chi_{\mathcal{L}_i} = \chi_{<1,m>}$ where χ_S is characteristic function of S and the entropy combination estimator is given by

$$\widehat{C}(V_{\mathcal{L}_1}, \ldots, V_{\mathcal{L}_p}) = F(k) - \sum_{i=1}^{p} s_i \left\langle F(k_i(j)) \right\rangle \tag{62}$$

where $F(k) = \psi(k) - \psi(N)$ and $k_i(j) = n_{V_{\mathcal{L}_i}}(j) + 1$ for $j = 1, \ldots, N$ as was formulated in section 1.3.5.

For example, from equation (55) it can be seen that mutual information is an entropy combination. Similarly from the one of the expressions for conditional mutual information

$$I(X; Y \mid Z) = H(X, Z) + H(Y, Z) - H(Z) - H(X, Y, Z) \tag{63}$$

it follows that $I(X; Y \mid Z)$ is also an entropy combination and combining (62) and (63) we get the Kraskov, Stögbauer and Grassberger estimator for conditional mutual information

$$\widehat{I}(X; Y \mid Z) = \psi(k) - \left\langle \psi(n_{xz} + 1) + \psi(n_{yz} + 1) - \psi(n_z + 1) \right\rangle. \tag{64}$$

Finally directly from the definition 13 we get the Kraskov, Stögbauer and Grassberger transfer entropy estimator

$$\widehat{T}_{Y \to X}^{(k,l)} = \psi(k) - \left\langle \psi(n_{x_t, \mathbf{x}_{t-1}^k} + 1) + \psi(n_{\mathbf{y}_{t-1}^l, \mathbf{x}_{t-1}^k} + 1) - \psi(n_{\mathbf{x}_{t-1}^k} + 1) \right\rangle. \tag{65}$$

1.3.6 Estimators parameter determination

The Kraskov, Stögbauer and Grassberger belongs to a class of nonparametric estimation methods. Nevertheless, since its based on k-nearest neighbour searches, k in the nearest neighbour algorithm is one of the parameters we have to chose.

This parameter has to be empirically determined. Kraskov et. al. [14] suggest $k > 1$ but not too large because of the increase of systematic errors. Generally they propose to use $k = 2 - 4$. Bossomaier et. al. in [12] writes that for $k \geq 4$ the estimator is robust and Wibral in [13] writes that $k = 4$ has been determined as a good choice for ECoG data.

Another problem is determining the embedding vector as was defined in Definition 13. For example, in case of transfer entropy we would like as much of the information that is contained in the target process to condition out. If we won't do this, the measured information transfer might be due to self prediction of the target process and not the dependance of the source and target processes. The method for finding the optimal values for the embedding vector is also called state space reconstruction.

There are multiple methods to determine the state space vectors. One way is to set (m, τ) such that it maximizes the active information storage. Wibral et. al. [13] propose to optimize the embedding parameters using the Ragwitz-Kantz criterion.

Ragwitz and Kantz in [16] proposed a method to extract a Markov process of order $m > 1$ from observed scalar time series using locally constant predictors. In the following paragraphs we will briefly describe this method.

Let $\vec{s}_n = (s_n, s_{n-\tau}, \ldots, s_{n-(m-1)\tau})$ be time delay embedding vector such that $s_{n+1} = g(\vec{s}_n)$ exists. Then the locally constant predictor for the unobserved s_{n+1} is

$$\widehat{s_{n+1}} = \frac{1}{|\mathcal{U}_n|} \sum_{\vec{s}_k \in \mathcal{U}_n} s_{k+1} \tag{66}$$

where $\mathcal{U}_n = \{\vec{s}_k : \|\vec{s}_k - \vec{s}_n\| \leq \epsilon\}$. In other words, the mean of the immediate futures of the ϵ-neighbours of $\vec{s}_n$. Ragwitz and Kantz in [16] argue that locally constant predictor is a predictor based on Markov transition probability assumption $p(s_{n+1}|s_n, s_{n-\tau}, \ldots, s_\tau) = p(s_{n+1}|s_n, s_{n-\tau}, \ldots, s_{n-(m-1)\tau})$. To get the transition the probabilities $p(s_{n+1}|\vec{s}_n)$, a locally constant approximation is used in the form of $p(s_{k+1}|\vec{s}_k) \approx \widehat{p}(s_{n+1}|\vec{s}_n)\ \forall\ \vec{s}_k \in \mathcal{U}_n$. The justification of locally constant predictor follows then from the idea that if we use the mean

square error of predictions $e^2 = \sum_{i=1}^{N}(s_{i+1} - \widehat{s}_{i+1})^2$ then the best estimator of s_{n+1} is

$$\widehat{s}_{n+1} = \int_{\vec{s}_k \in \mathcal{U}_n} s_{k+1} p(s_{k+1}|\vec{s}_n) ds_{k+1} \tag{67}$$

Thus we take those time delay embedding vector parameters as optimal which minimize the locally constant prediction error i.e.

$$(m, \tau) = \underset{m \in \mathbb{Z}, \tau \in \mathbb{Z}}{\arg\min} \left\| \vec{s} - \widehat{\vec{s}} \right\|. \tag{68}$$

1.3.7 Transfer entropy significance testing

One of the problems that plague all transfer entropy estimators is bias. Either systematic errors or statistical errors are both properties of estimators that has to be dealt with. If we observe non-zero value of transfer entropy it can be due to bias or variance and the transfer entropy estimator (65) is no exception. Since Kraskov, Stögbauer, Grassberger method is non-parametric one of the suggested statistical testing options is to use a permutation test [13] [12].

We are going to test the null hypothesis that there is no relationship between the source and target variables. First we have to come with surrogate source variable under to assumption that the null hypothesis is true. One way to do it is to by permuting $\mathbf{y}_{t-1}^{(l)}$ in the joint probability space $\{x_t, \mathbf{x}_{t-1}^{(k)}, \mathbf{y}_{t-1}^{(l)}\}$ thus destroying any predictive dependence of Y to X, and computing $T_{Y_{sur} \to X}$ of the surrogate source variable Y_{sur}. Repeating this procedure multiple times we can compute one-sided p-value of the test by the following equation:

$$P(T_{Y_{sur} \to X} \geq T_{Y \to X}) = \sum_{Y_{sur}: T_{Y_{sur} \to X} \geq T_{Y \to X}} P(Y_{sur}). \tag{69}$$

If the p-value is less 0.05 we reject the null hypothesis at the 5% significance level.

References

[1] S. Haykin, Neural Networks and Learning Machines (3rd Edition), Prentice Hall, 2009.

[2] H. Jaeger, "The "echo state" approach to analysing and training recurrent neural networks", German National Research Institute for Computer Science, Tech. Rep. GMD Report 148, 2001.

[3] H. Jaeger, "Short term memory in echo state networks", German National Research Institute for Computer Science, Tech. Rep. GMD Report 152, 2001.

[4] K. Hornik, "Approximation capabilities of multilayer feedforward networks", Neural networks, vol. 4, pp. 251-257, 1991.

[5] M. Lukoševičius, "A practical guide to applying echo state networks", In: G. Montavon, G. B. Orr, and K.-R. Mller (eds.) Neural Networks: Tricks of the Trade, 2nd ed. Springer,pp. 659-686, 2012.

[6] M. Cencini, F. Cecconi, A. Vulpiani, Chaos: From Simple models to complex systems (Series on Advances in Statistical Mechanics) (Volume 17), World Scientific Publishing Co, 2009.

[7] I. Farkaš and R. Bosák amd P. Gergeľ,"Computational analysis of memory capacity in echo state networks", Neural Networks, vol. 83, pp. 109–120, 2016.

[8] D. MacKay, Information Theory, Inference, and Learning Algorithms (fourth printing), Cambridge University Press, 2005.

[9] C. E. Shannon, "A mathematical theory of communication", Bell Systems Technical Journal, vol. 27, pp. 379–423, 1948.

[10] M. Paluš, V. Komárek, Z. Hrnčíř, K. Šteřbová, "Synchronization as adjustment of information rates: Detection from bivariate time series", Physical Review E, vol. 63, p. 046211, 2001.

[11] T. Schreiber, "Measuring information transfer", Physical Review Letters, vol. 85, no. 2, pp. 461–464, 2000.

[12] T.Bossomaier, L.Barnett, J. T.Lizier, An Introduction to Transfer Entropy, Springer, 2016.

[13] M. Wibral, R. Vicente, J. T.Lizier, Eds., Directed Information Measures in Neuroscience. Springer, 2014.

[14] A. Kraskov, H. Stögbauer, P. Grassberger, "Estimating mutual information,: Physics Reviews E, vol. 69, p. 066138, 2004.

[15] G. Gómez-Herrero, W. Wu, K. Rutanen, M. C. Soriano, G. Pipa, R. Vicente, "Assessing coupling dynamics from an ensemble of time series", Entropy, vol. 17, no.4, pp. 1958–1970.

[16] M. Ragwitz, H. Kantz, "Markov models from data by simple nonlinear time series predictors in delay embedding spaces",Physics Reviews E, vol. 65, no. 5, p. 056201, 2002.

[17] J. Geweke, "Measurement of linear dependence and feedback between multiple time series", Journal of American Statistical Association, vol. 77, no. 378, pp. 304–313, 1982.

[18] L. Barnett, A. B. Barrett, A. K. Seth, "Granger causality and transfer entropy are equivalent for Gaussian variables", Physical Review Letters, vol. 103, no. 23, p. 238701, 2009.

[19] E. T. Jaynes, Probability Theory: The Logic of Science, Cambridge University Press, 2003.