Abstract

xxxx

Kivonat

xxxx

Contents

Chapter 2

Recurrent Neural Networks

This thesis is focusing on recurrent neural network based multi-user detection. Before getting closer to the topic, the basic properties of recurrent neural nets are covered. Only discrete-time neural networks are considered, since they are easily implemented in DSP devices. However continous-time neural networks can also be realized in analogue devices, they are more difficult to tune. Tracking the changes of system parameters requires continous tuning, thus DSP devices—and discrete-time neural networks—are preferred.

Neural networks are artificial structures, built up by simple computational elements, called neurons. Generally, a very simple computation is taken care by each neuron. The equation describing the computation inside the neuron is termed as *iteration equation*. The properties and the functionality of neural networks mainly depend on the iteration equation. The typical structure of one neuron is given in Figure 2.1. The output of the neuron is denoted by $v_k[\ell+1]$. In general, $v_k[\ell]$ denotes the output of the kth neuron at iteration ℓ. At the beginning $\ell = 0$ and the iteration number increases if all the neurons are updated. The output is generated based on the output of other neurons multiplied by a factor of W_{ki} which denotes the weighting factor of the ith neuron's output at the kth neuron's input. The independent input u_k is also added here. The generated sum is fed into the *decision function* ($\Phi_{\mathcal{A}}\{.\}$) which develope the output. In our case the decision function chooses from a limited set of possible values (denoted by $\mathcal{A}$): $\forall k, \ell : v_k[\ell] \in \mathcal{A}$.

Based on Figure 2.1, the iteration equation is read as:

$$v_k[\ell+1] = \Phi_{\mathcal{A}}\left\{\frac{1}{W_{kk}}\left(u_k - \sum_{i=1}^{k-1} W_{ki}v_i[\ell+1] - \sum_{i=k+1}^{M} W_{ki}v_i[\ell]\right)\right\}. \tag{2.1}$$

In other words the weighted sum of the outputs of neurons is compared against the independent input u_k. The difference between the sum and u_k is meassured by the decision function $\Phi_{\mathcal{A}}\{.\}$. Equation (2.1) describes the iteration equation in the case of serial update, since parallel updating often results in cycles which prevents stability in terms of constant output. The outputs of the neurons should be set to zero before the iterations start ($\forall k : v_k[-1] = 0$), so at the beginning the independent input is the only term which should be taken into consideration ($\forall k : v_k[0] = \Phi_{\mathcal{A}}\{u_k\}$).

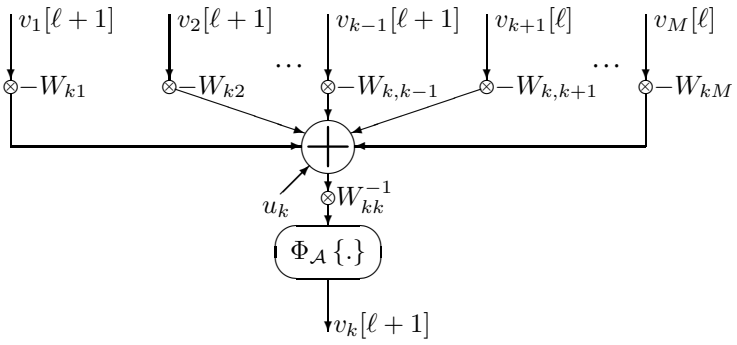

Figure 2.1: The structure of the kth neuron

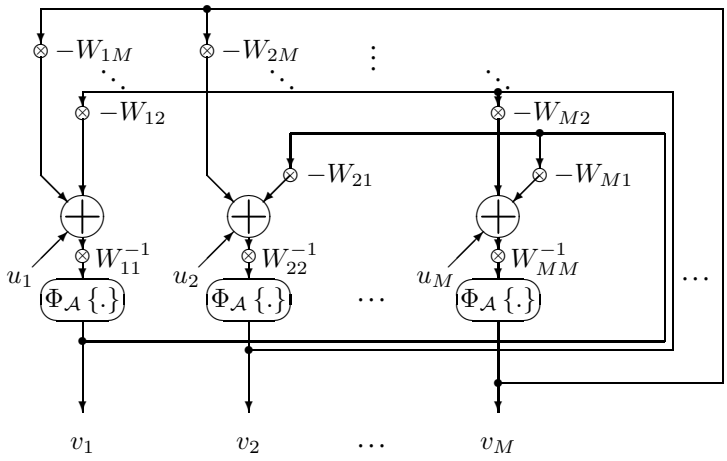

Figure 2.2: The structure of the recurrent neural network

The structure of the recurrent neural network is depicted in Figure 2.2. Each neuron in the figure represents the same functional element as Figure 2.1. The neurons form a network.

Note that in (2.1) the role of the diagonal elements of $\boldsymbol{W}$ (W_{nn}) is to normalize the expression. In the sequel, we assume that the diagonal elements in matrix $\boldsymbol{W}$ are real positive numbers and the matrix itself is Hermitian (both required for the stability analysis). For the sake of clarity the requirements are summarized here:

- Matrix $\boldsymbol{W}$ is Hermitian: $\boldsymbol{W} = \boldsymbol{W}^{\mathsf{H}}$.
- As a consequence, the main diagonal entries of matrix $\boldsymbol{W}$ are real numbers, moreover they are assumed to be positive ($\forall n$: $W_{nn} \in \mathbb{R}^+$).
- The neural network is updated serially, i. e. (2.1) is applied as the iteration equation.

Every recurrent neural network that fulfills these requirements is stable. The statement is proven in the following section.

2.1 Stability

There are several ways to prove the stability of a system. The well-known Lyapunov technique provides a complex mechanism which on the one hand proves the stability, on the other hand an upper bound can be computed for the iterations required to reach the steady state. Unfortunately our general model does not fit to Lyapunov analysis, thus an energy function is computed. Based on the parameters of the system, the energy function is explicitly given. If the energy function has a lower bound, and decreases monotonously in every iteration, the energy function converges and must have a steady state. If the energy function has a steady state, the corresponding system is stable.

2.1.1 The Energy Function

Recurrent neural networks applying the iteration equation (2.1) have an energy function in the following form:

$$\begin{aligned} E[\boldsymbol{v} = \boldsymbol{v}[\ell]] &= (\boldsymbol{v}[\ell])^{\mathsf{H}} \boldsymbol{W} (\boldsymbol{v}[\ell]) - 2\,\mathrm{Re}\left\{\boldsymbol{u}^{\mathsf{H}} (\boldsymbol{v}[\ell])\right\} \\ &= (\boldsymbol{v}[\ell])^{\mathsf{H}} \boldsymbol{W} (\boldsymbol{v}[\ell]) - \boldsymbol{u}^{\mathsf{H}} (\boldsymbol{v}[\ell]) - (\boldsymbol{v}[\ell])^{\mathsf{H}} \boldsymbol{u}, \end{aligned} \tag{2.2}$$

or in scalar form

$$E[\boldsymbol{v} = \boldsymbol{v}[\ell]] = \sum_{i=1}^{M} \sum_{j=1}^{M} v_i^*[\ell] W_{ij} v_j[\ell] - \sum_{i=1}^{M} u_i^* v_i[\ell] - \sum_{i=1}^{M} u_i v_i^*[\ell]. \tag{2.3}$$

The energy function is chosen to be upper and lower limited and to decrease in each iteration. In the sequel the upper and lower bounds are shown, and the monotonous decrease is also proven.

If not every components of vector $\boldsymbol{v}$ is updated, the energy function is still computable. During the iterations the energy function can be calculated as

$$E[\ell, l] = E[\boldsymbol{v} = [v_1[\ell+1], v_2[\ell+1], \ldots, v_{l-1}[\ell+1], v_l[\ell+1], v_{l+1}[\ell], \ldots, v_M[\ell]]^{\mathsf{T}}],$$

by substituting into (2.3):

$$\begin{aligned} E[\ell, l] = &\sum_{i=1}^{l}\sum_{j=1}^{l} v_i^*[\ell+1] W_{ij} v_j[\ell+1] + \sum_{i=1}^{l}\sum_{j=l+1}^{M} v_i^*[\ell+1] W_{ij} v_j[\ell] \\ &+ \sum_{i=l+1}^{M}\sum_{j=1}^{l} v_i^*[\ell] W_{ij} v_j[\ell+1] + \sum_{i=l+1}^{M}\sum_{j=l+1}^{M} v_i^*[\ell] W_{ij} v_j[\ell] \\ &- \sum_{i=1}^{l} u_i^* v_i[\ell+1] - \sum_{i=l+1}^{M} u_i^* v_i[\ell] - \sum_{i=1}^{l} u_i v_i^*[\ell+1] - \sum_{i=l+1}^{M} u_i v_i^*[\ell]. \end{aligned}$$

2.1.1.1 The Lower Bound

Due to the quadratic nature of the energy function in (2.2), a lower bound must exist. In the derivation the minimum energy is computed by substituting the appropriate vector $\boldsymbol{v}$ which results in extreme energy. Rewriting (2.2) results in

$$E[\boldsymbol{v}] = \left(\boldsymbol{v} - \boldsymbol{W}^{-1}\boldsymbol{u}\right)^{\mathsf{H}} \boldsymbol{W} \left(\boldsymbol{v} - \boldsymbol{W}^{-1}\boldsymbol{u}\right) - \boldsymbol{u}^{\mathsf{H}}\boldsymbol{W}^{-1}\boldsymbol{u}.$$

Since matrix $\boldsymbol{W}$ is Hermitian, it is positive semidefinite as well. Consequently, substituting $\boldsymbol{W}^{-1}\boldsymbol{u}$ into $\boldsymbol{v}$ results in minimal energy

$$E[\boldsymbol{v}] \geq \boldsymbol{v}^{\mathsf{H}}\boldsymbol{W}\boldsymbol{v} - \boldsymbol{u}^{\mathsf{H}}\boldsymbol{v} - \boldsymbol{v}^{\mathsf{H}}\boldsymbol{u}\Big|_{\boldsymbol{v}=\boldsymbol{W}^{-1}\boldsymbol{u}} = -\boldsymbol{u}^{\mathsf{H}}\boldsymbol{W}^{-1}\boldsymbol{u} \geq -\|\boldsymbol{u}\|\,\|\boldsymbol{W}\|^{-1}\|\boldsymbol{u}\|,$$

where $\|\boldsymbol{W}\|$ is the largest eigenvalue of matrix $\boldsymbol{W}$, $\|\boldsymbol{u}\|$ denotes the length of vector $\boldsymbol{u}$. Based on the length of vector $\boldsymbol{u}$ and the largest eigenvalue of matrix $\boldsymbol{W}$, the lower bound of the energy function (2.2) can be computed

$$E[\boldsymbol{v}] \geq -\frac{\|\boldsymbol{u}\|^2}{\|\boldsymbol{W}\|}. \tag{2.4}$$

2.1.1.2 The Upper Bound

The energy function has an upper bound if and only if the set $\mathcal{A}$ contains numbers with finite amplitude. Note that the existence of an upper bound is not required in the stability analysis, thus infinite values in the set $\mathcal{A}$ have no effect on the stability of the neural network. The upper bound might only be needed to approximate the maximum number of iterations to reach the steady state.

The energy function is surely less than or equal to the sum of the absolute values:

$$E[\boldsymbol{v}] \leq \|\boldsymbol{v}\|^2\|\boldsymbol{W}\| + 2\,\|\boldsymbol{u}\|\,\|\boldsymbol{v}\|.$$

Let $v^{\max}$ denote the maximum absolute value number (the number with the largest absolute value) in the set $\mathcal{A}$:

$$v^{\max} = \max_{x\in\mathcal{A}} |x|,$$

then the upper bound is given as

$$E[\boldsymbol{v}] \leq M\,(v^{\max})^2\|\boldsymbol{W}\| + 2\,\sqrt{M}\,v^{\max}\|\boldsymbol{u}\|. \tag{2.5}$$

The result shows that the upper bound can be computed using the maximum absolute value of the set $\mathcal{A}$, the largest eigenvlaue of matrix $\boldsymbol{W}$ and the length and dimension of vector $\boldsymbol{u}$.

2.1.1.3 The Alteration of the Energy Function

The change of the energy function during the update of the lth neuron in the $[\ell+1]$st iteration is denoted by $\Delta E[\ell, l]$:

$$\begin{aligned}\Delta E[\ell,l] = {} & E[\boldsymbol{v} = [v_1[\ell+1], v_2[\ell+1], \ldots, v_l[\ell+1], v_{l+1}[\ell], \ldots, v_M[\ell]]] \\ & - E[\boldsymbol{v} = [v_1[\ell+1], v_2[\ell+1], \ldots, v_{l-1}[\ell+1], v_l[\ell], \ldots, v_M[\ell]]].\end{aligned}$$

substituting (2.3) yields

$$\begin{aligned}\Delta E[\ell,l] = {} & \sum_{j=1}^{l-1} (v_l[\ell+1] - v_l[\ell])^* W_{lj} v_j[\ell+1] \\ & + \sum_{j=l+1}^{M} (v_l[\ell+1] - v_l[\ell])^* W_{lj} v_j[\ell] \\ & + \sum_{i=1}^{l-1} v_i^*[\ell+1] W_{il} (v_l[\ell+1] - v_l[\ell]) \\ & + \sum_{i=l+1}^{M} v_i^*[\ell] W_{il} (v_l[\ell+1] - v_l[\ell]) \\ & + \left(|v_l[\ell+1]|^2 - |v_l[\ell]|^2\right) W_{ll} \\ & - u_l^*(v_l[\ell+1] - v_l[\ell]) - u_l (v_l[\ell+1] - v_l[\ell])^*.\end{aligned}$$

Ordering similar terms into one group and taking into accout the Hermitian property of matrix $\boldsymbol{W}$ ($W_{ij} = W_{ji}^*$) results in

$$\begin{aligned}\Delta E[\ell,l] = {} & (v_l[\ell+1] - v_l[\ell])^* \left(\sum_{j=1}^{l-1} W_{lj} v_j[\ell+1] + \sum_{j=l+1}^{M} W_{lj} v_j[\ell] - u_l \right) \\ & + (v_l[\ell+1] - v_l[\ell]) \left(\sum_{j=1}^{l-1} W_{lj}^* v_j^*[\ell+1] + \sum_{j=l+1}^{M} W_{lj}^* v_j^*[\ell] - u_l^* \right) \\ & + \left(|v_l[\ell+1]|^2 - |v_l[\ell]|^2\right) W_{ll},\end{aligned}$$

which equals

$$\begin{aligned}\Delta E[\ell,l] = {} & \left(|v_l[\ell+1]|^2 - |v_l[\ell]|^2\right) W_{ll} \\ & + 2\,\mathrm{Re}\left\{ (v_l[\ell+1] - v_l[\ell])^* \left(\sum_{j=1}^{l-1} W_{lj} v_j[\ell+1] + \sum_{j=l+1}^{M} W_{lj} v_j[\ell] - u_l \right) \right\}.\end{aligned} \tag{2.6}$$

Note that the term inside the big round brackets in (2.6) is similar to the argument of function $\Phi_{\mathcal{A}}\{.\}$ in (2.1)—however, here the term is multiplied by the negative real number $-W_{ll}$.

Before proving the non-positive nature of (2.6), a lemma needs to be stated.

Lemma 2.1 *If $x \in \mathbb{C}$ is an arbitrary complex number, $y = \Phi_{\mathcal{A}}\{x\}$, and $z \in \mathcal{A}$ denotes an arbitrary number in the set $\mathcal{A}$, then*

$$|y|^2 - |z|^2 + 2\,\mathrm{Re}\{(y-z)^*(-x)\} \leq 0 \tag{2.7}$$

holds.

Proof: According to (1.3)

$$|x - y| \leq |x - z|.$$

Obviously the squares also satisfy the inequality (since square keeps relation):

$$|x-y|^2 \leq |x-z|^2,$$

in other form

$$(x-y)^*(x-y) \leq (x-z)^*(x-z).$$

Multiplying and ordering the terms the expression yields (2.7). □

Theorem 2.2 *For any alphabets $\mathcal{A}$, the energy function of the recurrent neural network (2.2) monotonously decreases in every iteration.*

Proof: Taking into consideration that for every l, W_{ll} is a positive real number, these numbers can be moved into and from the argument of the real part operator. For the sake of clarity the following definitions are formulated: $y = v_l[\ell+1]$, $z = v_l[\ell]$ and

$$x = \frac{1}{W_{ll}} \left(-\sum_{j=1}^{l-1} W_{lj} v_j[\ell+1] - \sum_{j=l+1}^{M} W_{lj} v_j[\ell] + u_l \right).$$

Applying the above defined variables (2.6) can be rewritten in the following form:

$$\Delta E[\ell, l] = W_{ll} \left(|y|^2 - |z|^2 + 2\,\mathrm{Re}\left\{ (y-z)^*(-x) \right\} \right).$$

Since W_{ll} is always positive and (2.7) holds (look the definitions above):

$$\forall \ell, l: \qquad \Delta E[\ell, l] \leq 0.$$

That is, (2.6) is always non-positive and thus the energy function decreases or does not change in each iteration. □

Theorem 2.3 *For any alphabets $\mathcal{A}$, the alteration of the energy function of the recurrent neural network (2.2) is the largest possible in every iteration.*

Proof: The theorem is proven indirectly. As earlier, the following definitions are formulated: $y = v_l[\ell+1]$, $z = v_l[\ell]$ and

$$x = \frac{1}{W_{ll}} \left(-\sum_{j=1}^{l-1} W_{lj} v_j[\ell+1] - \sum_{j=l+1}^{M} W_{lj} v_j[\ell] + u_l \right).$$

Assume that there exist an $y' \in \mathcal{A}$, such that $y' \neq v_l[\ell+1]$ and

$$\begin{aligned} & W_{ll} \left(|y'|^2 - |z|^2 + 2\,\mathrm{Re}\left\{ (y'-z)^*(-x) \right\} \right) \\ & < \Delta E[\ell, l] = W_{ll} \left(|y|^2 - |z|^2 + 2\,\mathrm{Re}\left\{ (y-z)^*(-x) \right\} \right). \end{aligned}$$

Following the argument of Lemma 2.1 in the opposite direction one gets

$$|y' - x| < |y - x|.$$

Since $y = \Phi_{\mathcal{A}}\{x\}$, this contradicts (1.3). That is, the assumption was wrong; there does not exist such $y' \in \mathcal{A}$ for which the energy function of the recurrent neural network would decrease more. □

Since the energy function defined in (2.3) has a lower bound and decreases (or remains the same) in each iteration, the energy function tends to a local minimum. The energy function belongs to the neural network, thus the recurrent neural network defined by the iteration equation (2.1) is stable. In other words to locally optimize the energy function (2.3) one can apply a recurrent neural network. The latter statement helps the application of recurrent neural networks for the purpose of multi-user detection.

For a two dimensional example see Figure 2.3. In the figure the energy function of a two dimensional nrecurrent neural network is depicted as a discrete function of the neuron outputs. Let us assume that the network runs from the initial state $\boldsymbol{v}[0] = [-1, -1]^{\mathsf{T}}$. In each step the net decreases the energy function, thus the first iteration produces $\boldsymbol{v}[1] = [-1, +1]^{\mathsf{T}}$, and in the second iteration the network arrives at the steady state $\boldsymbol{v}[2] = [+1, +1]^{\mathsf{T}}$.

However, local optimization lacks the assurance of finding the global optimum. Local optima may be far from the global one, thus the performance of the multi-user detector may be poor. To avoid local optima several recurrent neural network extensions have been proposed. Three of which are discussed in the following sections.

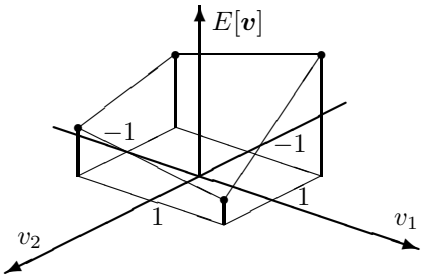

Figure 2.3: Two Dimensional Energy Function

2.2 Stochastic Recurrent Neural Networks

The idea behind stochastic recurrent neural network lies in the addition of some noise to each state transition in order to escape from local minima. Therefore, the original iteration equation of the recurrent neural network (2.1) is modified as follows

$$v_k[\ell+1] = \Phi_{\mathcal{A}}\left\{\frac{1}{W_{kk}}\left(u_k - \sum_{i=1}^{k-1} W_{ki}v_i[\ell+1] - \sum_{i=k+1}^{M} W_{ki}v_i[\ell]\right) + \nu_k[\ell]\right\}. \tag{2.8}$$

Here, the extra parameter $\nu_k[\ell]$ represents the noise which is independent identically distributed random variable with zero mean and $F(z,\ell)$ distribution function

$$\Pr\left\{\mathrm{Re}\left\{\nu_k[\ell]\right\} \leq \mathrm{Re}\left\{z\right\} \cap \mathrm{Im}\left\{\nu_l[\ell]\right\} \leq \mathrm{Im}\left\{z\right\}\right\} = F(z,\ell). \tag{2.9}$$

Furthermore, we assume that

$$\mathsf{d}F(z,\ell) = \mathsf{d}F(-z,\ell) \tag{2.10}$$

i. e. the distribution function is symmetric. Thirdly, the random variable can get values from the whole signal space, i. e.

$$\forall \mathcal{B} \in \mathbb{C} : \int_{\mathcal{B}} \mathsf{d}z \neq 0 \rightarrow \int_{\mathcal{B}} \mathsf{d}F(z,\ell) \neq 0. \tag{2.11}$$

The variance of $\nu_k[\ell]$ is assumed to be a monotonous decreasing function tending to zero ($\lim_{\ell\to\infty} \mathbb{E}\left\{\nu_k^2[\ell]\right\} = 0$). One should note that the symmetry assumption given in (2.10) embodies zero mean for $\nu_k[\ell]$ (for a detailed proof see Theorem A.1 on page 65). The stochastic recurrent neural network has a stochastic state transition rule because of the noise term. The stochastic recurrent neural network scheme is depicted in Figure 2.4.

Now we set out to prove that the stationary state of the random state transition rule yields the global optimum. To accomplish this goal we need a Markovian description of the stochastic recurrent neural network, which is introduced in Section 2.2.1. Section 2.2.4 is the main part of this document; it proves that applying logistic distribution yields maximal probability for the optimal solution. Finally in Section 2.2.5 we extend the results to the inhomogeneous case when the variance of noise tends to zero, according to a certain cooling schedule. However this extension is investigated only asymptotically, it shows that the optimal solution is reached with probability one.

2.2.1 The Stationary Distribution

In order to analyze the stochastic convergence of the network, we need to define the state space in which the algorithm operates. Since the states of the stochastic recurrent neural network are vectors with components taken from the set $\mathcal{A}$, the state space over which the convergence is to be investigated grows exponentially $|\mathcal{A}|^M$. From the state transition rule (2.8) it is clear that the vector $\boldsymbol{v}[\ell+1]$ depends on the stochastic vector $\boldsymbol{v}[\ell]$ and the noise vector $\boldsymbol{\nu}[\ell]$, all other terms are regarded constant during the iterations. It implies that vector $\boldsymbol{v}[\ell]$ is a Markovian random process.

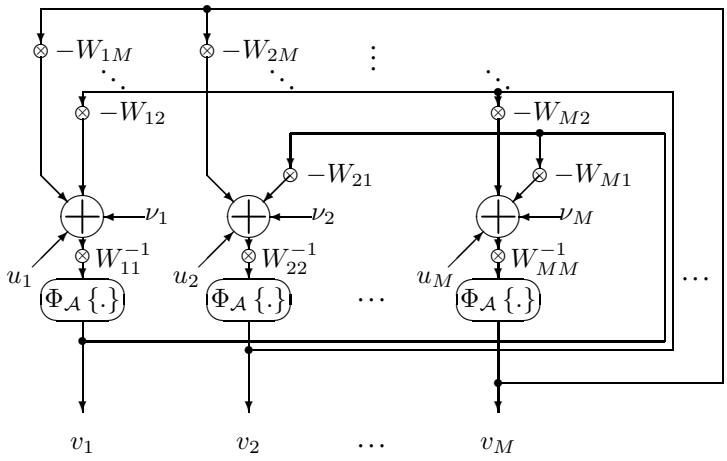

Figure 2.4: The structure of the stochastic recurrent neural network

First the possible vectors of neuron outputs must be labelled. We will use the superscript as the label in the following, thus $\boldsymbol{v}^i$ denotes a pre-defined constant vector with components in $\mathcal{A}$ ($\boldsymbol{v}^i \in \mathcal{A}^M$). Based on the system of labels the state of the stochastic recurrent neural network is referred to by the same label (state i is defined by the neuron output $\boldsymbol{v}^i$). Let $\boldsymbol{P}[\ell]$ denote the stochastic state transition matrix associated to this process at the time instant ℓ, so $\boldsymbol{P}[\ell] = [P_{ij}[\ell]]$, where

$$P_{ij}[\ell] = \Pr\left\{\boldsymbol{v}[\ell+1] = \boldsymbol{v}^i \mid \boldsymbol{v}[\ell] = \boldsymbol{v}^j\right\}, \tag{2.12}$$

Let $\boldsymbol{\pi}[\ell]$ denote the probability vector of different states at the time instant ℓ, so $\boldsymbol{\pi}[\ell] = [\pi_i[\ell]]$, where

$$\pi_i[\ell] = \Pr\left\{\boldsymbol{v}[\ell] = \boldsymbol{v}^i\right\}.$$

It is obvious that using the above defined matrix and vector, the state transition rule can be written in the following form:

$$\boldsymbol{\pi}[\ell+1] = \boldsymbol{P}[\ell]\,\boldsymbol{\pi}[\ell].$$

Obviously the dimension of both vector $\boldsymbol{\pi}[\ell]$ and matrix $\boldsymbol{P}[\ell]$ equals $|\mathcal{A}|^M$.

Lemma 2.4 *The stochastic recurrent neural network defines an aperiodic and irreducible Markovian process.*

Proof: Since the noise $\boldsymbol{\nu}[\ell]$ can get any arbitrary large values (see (2.11)), every state can be reached with a finitely small probability from each state. As a consequence, the Markovian process is neither periodic nor reducible. □

Since the Markovian process lying in the background is aperiodic and irreducible, the stationary distribution denoted by $\boldsymbol{\pi}$ is given by the following equation:

$$\boldsymbol{\pi} = \boldsymbol{P}\,\boldsymbol{\pi}, \tag{2.13}$$

where $\boldsymbol{P} = \lim_{\ell\to\infty} \boldsymbol{P}[\ell]$. Our primary aim is to reveal the stationary distribution $\boldsymbol{\pi}$, to prove that the global optimum has the highest probability in equilibrium, i. e. the state

$$\text{opt} = \arg\max_i p_i = \arg\min_i \left\{ {\boldsymbol{v}^i}^{\mathsf{H}} \boldsymbol{W}\, \boldsymbol{v}^i - 2\,\mathrm{Re}\left\{\boldsymbol{u}^{\mathsf{H}}\boldsymbol{v}^i\right\}\right\},$$

or equivalently, vector

$$\boldsymbol{v}^{\text{opt}} = \arg\min_{\boldsymbol{v}\in\mathcal{A}^M} \left\{\boldsymbol{v}^{\mathsf{H}}\boldsymbol{W}\,\boldsymbol{v} - 2\,\mathrm{Re}\left\{\boldsymbol{u}^{\mathsf{H}}\boldsymbol{v}\right\}\right\} \tag{2.14}$$

has the maximal stationary probability.

In Section 2.2.2 the state transition probability matrix $\boldsymbol{P}[\ell]$ is directly computed, forthcoming sections build on the equations derived here. In Section 2.2.3 the stationary probabilities are compared based on the

components of the state transition probability matrix. In Section 2.2.4 the so-called logistic distribution function is applied to generate the noise values. The application of logistic distribution yields tractable equations, and finally it turns out that the global optimum has the highest probability in equilibrium. Furthermore, in Section 2.2.5 the inhomogeneous behaviour of the stochastic recurrent neural network results asymptotically in one probability at the optimal solution.

2.2.2 Brute Force Derivation of the State Transition Matrix

In this section we derive (2.12) by brute force. As will be shown, the brute force algorithm results in rather complex form except the binary case. Equation (2.12) can be rewritten as

$$P_{ij}[\ell] = \Pr\left\{\left\{v_1[\ell+1] = v_1^i \mid \boldsymbol{v}[\ell] = \boldsymbol{v}^j\right\} \cap \left\{v_2[\ell+1] = v_2^i \mid \boldsymbol{v}[\ell] = \boldsymbol{v}^j \cap v_1[\ell+1] = v_1^i\right\} \cap \ldots\right\}$$

Due to the independence of noise values, the probability can be decomposed into a product

$$\begin{aligned} P_{ij}[\ell] = &\Pr\left\{v_1[\ell+1] = v_1^i \mid \boldsymbol{v}[\ell] = \boldsymbol{v}^j\right\} \\ &\cdot\Pr\left\{v_2[\ell+1] = v_2^i \mid \boldsymbol{v}[\ell] = \boldsymbol{v}^j \cap v_1[\ell+1] = v_1^i\right\} \cdot \ldots \end{aligned}$$

or equivalently

$$P_{ij}[\ell] = \prod_{k=1}^{M} G[i,j,k,\ell], \tag{2.15}$$

where $G[i,j,k,\ell]$ is given as

$$G[i,j,k,\ell] = \Pr\left\{v_k[\ell+1] = v_k^i \mid \boldsymbol{v}[\ell] = \boldsymbol{v}^j \cap v_1[\ell+1] = v_1^i \cap \ldots \cap v_{k-1}[\ell+1] = v_{k-1}^i\right\}.$$

Note that the probability defined in (2.15) tells the value of different terms in (2.8). The probability can be translated into the probability of having appropriate noise value in $\nu_k[\ell]$, since the stochastic nature is caused by the noise term only. Lemma 2.1 holds here too, i. e. $G[i,j,k,\ell]$ can be rewritten into an integral

$$G[i,j,k,\ell] = \int_{S[i,j,k]} \mathsf{d}F(z,\ell), \tag{2.16}$$

where $S[i,j,k]$ denotes the surface, where the integral should be computed. Based on Lemma 2.1 $S[i,j,k]$ is given as

$$\begin{aligned} &S[i,j,k] = z \in \mathbb{C}, \forall y \in \mathcal{A}: \\ &|v_k^i|^2 - |y|^2 + 2\,\mathrm{Re}\left\{(v_k^i - y)^*\left(\frac{1}{W_{kk}}\left(\sum_{l=1}^{k-1} W_{kl}v_l^i + \sum_{l=k+1}^{M} W_{kl}v_l^j - u_k\right) - z\right)\right\} \leq 0 \end{aligned}$$

which can be simplified into

$$\begin{aligned} &S[i,j,k] = z \in \mathbb{C}, \forall y \in \mathcal{A}: \\ &\mathrm{Re}\left\{(v_k^i - y)^*\left(\frac{1}{2}\left(v_k^i + y\right) + \frac{1}{W_{kk}}\left(\sum_{l=1}^{k-1} W_{kl}v_l^i + \sum_{l=k+1}^{M} W_{kl}v_l^j - u_k\right) - z\right)\right\} \leq 0. \end{aligned} \tag{2.17}$$

2.2.2.1 The Binary Case

In this section we derive (2.15) for the case of binary output neurons ($\mathcal{A} = \{\pm 1\}$). Note that in the case of binary neurons, the decision function turns into a simple signum $\Phi_{\mathcal{A}}\{.\} = \mathrm{sgn}\{.\}$, thus imaginary values have no role. For the same reason the real part operator disappears in (2.17):

$$\begin{aligned} &S[i,j,k] = x \in (-\infty,\infty), y = \pm 1: \\ &(v_k^i - y)\left(\frac{1}{2}\left(v_k^i + y\right) + \frac{1}{W_{kk}}\left(\sum_{l=1}^{k-1} W_{kl}v_l^i + \sum_{l=k+1}^{M} W_{kl}v_l^j - u_k\right) - x\right) \leq 0. \end{aligned} \tag{2.18}$$

Theorem 2.5 *If one uses symmetric distribution $F(x,\ell)$, which satisfies (2.10) to generate values of the noise source, then*

$$P_{ij}[\ell] = \prod_{k=1}^{M} F\left(\frac{v_k^i}{W_{kk}}\left(u_k - \sum_{l=1}^{k-1} W_{kl} v_l^i - \sum_{l=k+1}^{M} W_{kl} v_l^j\right), \ell\right) \tag{2.19}$$

holds.

Proof: In the proof both $v_k^i = +1$ and $v_k^i = -1$ are considered separately. It is highlighted that both cases yield the same equation in (2.16). Finally, $G[i,j,k,\ell]$ is substituted into (2.15) thus the statement is got.

Let us assume that $v_k[\ell+1] = v_k^i = -1$, substituting into (2.18) it turns out that $y = +1$ is the only case we have to consider (if y is equal to minus one, the expression becomes zero). Now we get

$$S[i,j,k] = x \in (-\infty, \infty):$$
$$2\left(\frac{1}{W_{kk}}\left(\sum_{l=1}^{k-1} W_{kl} v_l^i + \sum_{l=k+1}^{M} W_{kl} v_l^j - u_k\right) - x\right) \geq 0.$$

which shows that

$$S[i,j,k] = x \in \left(-\infty, \frac{1}{W_{kk}}\left(\sum_{l=1}^{k-1} W_{kl} v_l^i + \sum_{l=k+1}^{M} W_{kl} v_l^j - u_k\right)\right].$$

Mutliplying the upper bound by v_k^i (it was minus one) and substituting into (2.16) the expression results in

$$G[i,j,k,\ell] = F\left(\frac{v_k^i}{W_{kk}}\left(u_k - \sum_{l=1}^{k-1} W_{kl} v_l^i - \sum_{l=k+1}^{M} W_{kl} v_l^j\right), \ell\right). \tag{2.20}$$

Equivalently, if $v_k[\ell+1] = v_k^i = +1$, substituting into (2.18) results that $y = -1$ is the only case we have to consider (if y is equal to plus one, the expression becomes zero). Now we get

$$S[i,j,k] = x \in (-\infty, \infty):$$
$$2\left(\frac{1}{W_{kk}}\left(\sum_{l=1}^{k-1} W_{kl} v_l^i + \sum_{l=k+1}^{M} W_{kl} v_l^j - u_k\right) - x\right) \leq 0$$

yielding

$$S[i,j,k] = x \in \left[\frac{1}{W_{kk}}\left(\sum_{l=1}^{k-1} W_{kl} v_l^i + \sum_{l=k+1}^{M} W_{kl} v_l^j - u_k\right), \infty\right).$$

Substituting the result into (2.16)

$$G[i,j,k,\ell] = \int_{x=\left(\frac{1}{W_{kk}}\left(\sum_{l=1}^{k-1} W_{kl} v_l^i + \sum_{l=k+1}^{M} W_{kl} v_l^j - u_k\right)\right)}^{\infty} \mathrm{d}F(x,\ell),$$

applying the symmetry condition of (2.10)

$$G[i,j,k,\ell] = \int_{x=-\infty}^{-\left(\frac{1}{W_{kk}}\left(\sum_{l=1}^{k-1} W_{kl} v_l^i + \sum_{l=k+1}^{M} W_{kl} v_l^j - u_k\right)\right)} \mathrm{d}F(x,\ell),$$

in other words (2.20) holds. What we have got tells that independently of the value of v_k^i, $G[i,j,k,\ell]$ has the same form. After substituting $G[i,j,k,\ell]$ into (2.15), the statement (2.19) is got. □

A simple example Let us investigate the following binary system with two neurons ($M = 2$): $W_{11} = W_{22} = 1$, $W_{12} = W_{21} = 0.6$ and $u_1 = 0.4$, $u_2 = 0.2$, respectively. Let the distribution function of the noise be $F(x) = \frac{1}{1+e^{-x}}$ to obtain a stationer Markov process (independent of ℓ). Let the binary state vectors be ordered in the following manner: $\boldsymbol{v}^1 = [-1,-1]^\mathsf{T}$, $\boldsymbol{v}^2 = [+1,-1]^\mathsf{T}$, $\boldsymbol{v}^3 = [-1,+1]^\mathsf{T}$ and $\boldsymbol{v}^4 = [+1,+1]^\mathsf{T}$. Based on (2.19) the following state transition probabilities are got:

$$\begin{aligned}
P_{11} = P_{12} &= F(-1)F(-0.8) = 0.269 \cdot 0.31 = 0.083\\
P_{21} = P_{22} &= F(+1)F(+0.4) = 0.731 \cdot 0.599 = 0.438\\
P_{31} = P_{32} &= F(-1)F(+0.8) = 0.269 \cdot 0.69 = 0.186\\
P_{41} = P_{42} &= F(+1)F(-0.4) = 0.731 \cdot 0.401 = 0.293\\
P_{13} = P_{14} &= F(+0.2)F(-0.8) = 0.55 \cdot 0.31 = 0.171\\
P_{23} = P_{24} &= F(-0.2)F(+0.4) = 0.45 \cdot 0.599 = 0.270\\
P_{33} = P_{34} &= F(+0.2)F(+0.8) = 0.55 \cdot 0.69 = 0.380\\
P_{43} = P_{44} &= F(-0.2)F(-0.4) = 0.45 \cdot 0.401 = 0.180.
\end{aligned}$$

Solving (2.13), the stationary distribution equals $\boldsymbol{\pi} = [0.129, 0.351, 0.287, 0.234]^\mathsf{T}$.

2.2.2.2 The QPSK Case

In this section (2.15) is derived for the case of QPSK symbol set, i. e. $\mathcal{A} = \frac{1}{\sqrt{2}}\{\pm 1 \pm j\}$. The decision function becomes signum in both real and imaginary axis that is

$$\Phi_{\mathcal{A}}\{z\} = \frac{1}{\sqrt{2}}\Big(\operatorname{sgn}\{\operatorname{Re}\{z\}\} + j\operatorname{sgn}\{\operatorname{Im}\{z\}\}\Big).$$

Theorem 2.6 *If one uses a complex number generator with distribution $F(z,\ell)$ which satisfies*

$$\mathsf{d}F(z,\ell) = \mathsf{d}F(z \cdot e^{j\pi/2},\ell) \tag{2.21}$$

(a stronger condition than (2.10)), then

$$P_{ij}[\ell] = \prod_{k=1}^{M} F\left(\frac{\left(v_k^i\right)^* e^{j\pi/4}}{W_{kk}}\left(u_k - \sum_{l=1}^{k-1} W_{kl}v_l^i - \sum_{l=k+1}^{M} W_{kl}v_l^j\right), \ell\right) \tag{2.22}$$

holds.

Proof: As in the binary case, all four possible v_k^i values must be considered separately. Here, only one of them is detailed, the rest (without any interesting technical modification) can be found in Theorem A.2 on page 65.

If $v_k[\ell+1] = v_k^i = (1+j)/\sqrt{2}$, then $(v_k^i - y)^*$ might take a value from the set $\{\sqrt{2}, -\sqrt{2}j, \sqrt{2}(1-j)\}$. In the first case, (2.17) yields

$$\operatorname{Re}\{z\} \geq \operatorname{Re}\left\{\frac{1}{W_{kk}}\left(\sum_{l=1}^{k-1} W_{kl}v_l^i + \sum_{l=k+1}^{M} W_{kl}v_l^j - u_k\right)\right\},$$

because for any pair of complex numbers z, v, $\operatorname{Re}\{z+v\} = \operatorname{Re}\{z\} + \operatorname{Re}\{v\}$. Since for all z complex numbers $\operatorname{Re}\{j \cdot z\} = -\operatorname{Im}\{z\}$, in the second case ($(v_k^i - y)^* = -\sqrt{2}j$), (2.17) becomes

$$\operatorname{Im}\{z\} \geq \operatorname{Im}\left\{\frac{1}{W_{kk}}\left(\sum_{l=1}^{k-1} W_{kl}v_l^i + \sum_{l=k+1}^{M} W_{kl}v_l^j - u_k\right)\right\}.$$

At last, if $(v_k^i - y)^* = \sqrt{2}(1-j)$, the following constraint is got

$$\begin{aligned}
\operatorname{Re}\{z\} + \operatorname{Im}\{z\} \geq \operatorname{Re}&\left\{\frac{1}{W_{kk}}\left(\sum_{l=1}^{k-1} W_{kl}v_l^i + \sum_{l=k+1}^{M} W_{kl}v_l^j - u_k\right)\right\}\\
&+\operatorname{Im}\left\{\frac{1}{W_{kk}}\left(\sum_{l=1}^{k-1} W_{kl}v_l^i + \sum_{l=k+1}^{M} W_{kl}v_l^j - u_k\right)\right\},
\end{aligned}$$

which is weaker than the previous two (if they are satisfied, the third inequality must hold too), thus it can be neglected. There are two constraints, one for the real part of z and one for the imaginary part of z. Substituting the result into (2.16)

$$G[i,j,k,\ell] = \int_{z=\frac{1}{W_{kk}}\left(\sum_{l=1}^{k-1} W_{kl}v_l^i + \sum_{l=k+1}^{M} W_{kl}v_l^j - u_k\right)}^{\infty^{++}} \mathsf{d}F(z,\ell),$$

where ∞^{++} refers to the point, where $F(\infty^{++}) = 1$ (the right upper corner of the complex plane). Using the symmetric property (2.21),

$$G[i,j,k,\ell] = F\left(\frac{\left(v_k^i\right)^* e^{j\pi/4}}{W_{kk}}\left(u_k - \sum_{l=1}^{k-1} W_{kl}v_l^i - \sum_{l=k+1}^{M} W_{kl}v_l^j\right), \ell\right). \tag{2.23}$$

For all other cases (2.23) is got (for detailed proof see Theorem A.2 on page 65). Finally substituting into (2.15), the statement (2.22) is got. □

Comparing (2.19) and (2.22) one should perceive the striking similarity. However, for general complex alphabets the brute force derivation results in a rather challenging task. Thus other methods are needed to provide a general description of the stochastic recurrent neural network.

2.2.3 Markovian Analysis of the State Transition Matrix

In this section (2.13) and matrix $\boldsymbol{P}$ are analysed. Basically, states must be compared to obtain an expression that the optimal state has the highest probability. Thus the two states examined will be denoted by A and B.

Matrix $\boldsymbol{P}$ is decomposed with the following definitions:

- P_{AA} gives the probability of remaining in state A during the iteration,
- P_{BB} represents the probability of remaining in state B,
- P_{BA} equals the probability of moving from state A into state B,
- P_{AB} equals the probability of the reverse move (from B to A),
- $\boldsymbol{p}_{*A}$ denotes the probability vector of moving from state A to any other states excluding A and B,
- vector $\boldsymbol{p}_{*B}$ denotes the same, but starting at state B,
- $\boldsymbol{p}_{A*}$ is equal to the probability vector of moving from any other states excluding A and B to state A,
- vector $\boldsymbol{p}_{B*}$ denotes the same, but ending at state B,
- $\boldsymbol{P}_*$ equals the state transition probability matrix between the rest (all except A and B) states.

Since for all j state $\sum_i P_{ij} = 1$ (with one probability the state transition occurs), the following equations hold

$$P_{AA} + P_{BA} + \mathbf{1}^\mathsf{T}\boldsymbol{p}_{*A} = 1 \tag{2.24}$$
$$P_{BB} + P_{AB} + \mathbf{1}^\mathsf{T}\boldsymbol{p}_{*B} = 1 \tag{2.25}$$
$$\boldsymbol{p}_{A*}^\mathsf{T} + \boldsymbol{p}_{B*}^\mathsf{T} + \mathbf{1}^\mathsf{T}\boldsymbol{P}_* = \mathbf{1}^\mathsf{T}. \tag{2.26}$$

To help understanding the notations, one example is drawn. If A and B refers to the first two states (the first two rows and columns in matrix $\boldsymbol{P}$), the structure of the state transition matrix is given as

$$\boldsymbol{P} = \left[\begin{array}{c|c|c} P_{AA} & P_{AB} & \boldsymbol{p}_{A*}^\mathsf{T} \\ \hline P_{BA} & P_{BB} & \boldsymbol{p}_{B*}^\mathsf{T} \\ \hline \boldsymbol{p}_{*A} & \boldsymbol{p}_{*B} & \boldsymbol{P}_* \end{array}\right].$$

Note that (2.24)–(2.26) always hold, the special assumption is only needed to visualize the notations.

For the stationary distribution vector $\boldsymbol{\pi}$ some more notations are introduced:

- π_A denotes the stationary probability of state A,
- π_B equals the stationary probability of state B,
- vector $\boldsymbol{\pi}_*$ consists of the stationary probabilities of other states (all except A and B).

Due to whole probability theorem, $\pi_A+\pi_B+\boldsymbol{\pi}_*^\mathsf{T}\mathbf{1}=1$ holds. With the introduced decomposition the following theorem can be formulated.

Theorem 2.7 *Without the knowledge of the stationary distribution, the relation of two probabilities can be computed. The ratio of probabilities of states A and B is given as*

$$\frac{\pi_A}{\pi_B}=\frac{P_{BB}+\boldsymbol{p}_{B*}^\mathsf{T}(\boldsymbol{I}-\boldsymbol{P}_*)^{-1}\boldsymbol{p}_{*B}-1}{P_{AA}+\boldsymbol{p}_{A*}^\mathsf{T}(\boldsymbol{I}-\boldsymbol{P}_*)^{-1}\boldsymbol{p}_{*A}-1}. \tag{2.27}$$

Proof: Using (2.13), and substituting the above definitions, one arrives at

$$\pi_A=\pi_A\cdot(1-P_{BA}-\mathbf{1}^\mathsf{T}\boldsymbol{p}_{*A})+\pi_B\cdot P_{AB}+\boldsymbol{p}_{A*}^\mathsf{T}\cdot\boldsymbol{\pi}_* \tag{2.28}$$
$$\pi_B=\pi_A\cdot P_{BA}+\pi_B\cdot(1-P_{AB}-\mathbf{1}^\mathsf{T}\boldsymbol{p}_{*B})+\boldsymbol{p}_{B*}^\mathsf{T}\cdot\boldsymbol{\pi}_* \tag{2.29}$$
$$\boldsymbol{\pi}_*=\pi_A\cdot\boldsymbol{p}_{*A}+\pi_B\cdot\boldsymbol{p}_{*B}+\boldsymbol{P}_*\cdot\boldsymbol{\pi}_*, \tag{2.30}$$

where (2.24) and (2.25) was also substituted. Based on (2.30) $\boldsymbol{\pi}_*$ is given as

$$\boldsymbol{\pi}_*=(\boldsymbol{I}-\boldsymbol{P}_*)^{-1}(\pi_A\cdot\boldsymbol{p}_{*A}+\pi_B\cdot\boldsymbol{p}_{*B}).$$

which is substituted into the difference of (2.28) and (2.29):

$$\begin{aligned}\frac{\pi_A}{\pi_B}-1=\frac{\pi_A}{\pi_B}-1-\frac{\pi_A}{\pi_B}P_{BA}-\frac{\pi_A}{\pi_B}\cdot\mathbf{1}^\mathsf{T}\boldsymbol{p}_{*A}+P_{AB}-\frac{\pi_A}{\pi_B}P_{BA}+P_{AB}+\mathbf{1}^\mathsf{T}\boldsymbol{p}_{*B}\\ +\frac{\pi_A}{\pi_B}\cdot\boldsymbol{p}_{A*}^\mathsf{T}(\boldsymbol{I}-\boldsymbol{P}_*)^{-1}\boldsymbol{p}_{*A}+\boldsymbol{p}_{A*}^\mathsf{T}(\boldsymbol{I}-\boldsymbol{P}_*)^{-1}\boldsymbol{p}_{*B}\\ -\frac{\pi_A}{\pi_B}\cdot\boldsymbol{p}_{B*}^\mathsf{T}(\boldsymbol{I}-\boldsymbol{P}_*)^{-1}\boldsymbol{p}_{*A}-\boldsymbol{p}_{B*}^\mathsf{T}(\boldsymbol{I}-\boldsymbol{P}_*)^{-1}\boldsymbol{p}_{*B}.\end{aligned}$$

Using (2.26),

$$\begin{aligned}0=-2\frac{\pi_A}{\pi_B}P_{BA}+2P_{AB}-\frac{\pi_A}{\pi_B}\cdot\mathbf{1}^\mathsf{T}\boldsymbol{p}_{*A}+\mathbf{1}^\mathsf{T}\boldsymbol{p}_{*B}+\frac{\pi_A}{\pi_B}\cdot\boldsymbol{p}_{A*}^\mathsf{T}(\boldsymbol{I}-\boldsymbol{P}_*)^{-1}\boldsymbol{p}_{*A}\\ -\boldsymbol{p}_{B*}^\mathsf{T}(\boldsymbol{I}-\boldsymbol{P}_*)^{-1}\boldsymbol{p}_{*B}+\left(\mathbf{1}^\mathsf{T}(\boldsymbol{I}-\boldsymbol{P}_*)-\boldsymbol{p}_{B*}^\mathsf{T}\right)(\boldsymbol{I}-\boldsymbol{P}_*)^{-1}\boldsymbol{p}_{*B}\\ -\frac{\pi_A}{\pi_B}\cdot\left(\mathbf{1}^\mathsf{T}(\boldsymbol{I}-\boldsymbol{P}_*)-\boldsymbol{p}_{A*}^\mathsf{T}\right)(\boldsymbol{I}-\boldsymbol{P}_*)^{-1}\boldsymbol{p}_{*A}.\end{aligned}$$

Simplifying the expression $\frac{\pi_A}{\pi_B}$ yields

$$\frac{\pi_A}{\pi_B}=\frac{P_{AB}+\mathbf{1}^\mathsf{T}\boldsymbol{p}_{*B}-\boldsymbol{p}_{B*}^\mathsf{T}(\boldsymbol{I}-\boldsymbol{P}_*)^{-1}\boldsymbol{p}_{*B}}{P_{BA}+\mathbf{1}^\mathsf{T}\boldsymbol{p}_{*A}-\boldsymbol{p}_{A*}^\mathsf{T}(\boldsymbol{I}-\boldsymbol{P}_*)^{-1}\boldsymbol{p}_{*A}},$$

in other words (because of (2.24) and (2.25)) equation (2.27) also holds. □

Our result seems promising, since the matrix $(\boldsymbol{I}-\boldsymbol{P}_*)^{-1}$ contains only positive numbers [**?**]. In Markov Theory, $[(\boldsymbol{I}-\boldsymbol{P}_*)^{-1}]_{ij}$ tells the mean value of moves needed to arrive at state i, starting from state j walking through both states A and B.

There might be a condition list based on Theorem 2.7, which always holds and is adaptable to the topic of this thesis (it should be easy to check). However, it has not been found by the author yet.

The Simple Example (continued) Taking a look at the example introduced on page 16, one can check that Theorem 2.7 holds, i. e. $\pi_2>\pi_i$, where $i=1,3,4$. (the calculations are not included here to save place).

2.2.4 The Global Optimizer with Logistic Distribution

Further investigation requires more strict conditions. In this section only the binary and the QPSK cases are considered, because general alphabets prevents easy calculations. The two cases are treated in separate subsections. First, the binary case is considered, thus the QPSK subsection highlights only the differences caused by the complex domain. General complex alphabets are not treated in this section.

First, we introduce a new distribution, termed as the *logistic distribution*. This distribution function is denoted by $F^{\log}(x,\gamma)$ in the case of real values (for binary neurons):

$$F^{\log}(x,\gamma) = \frac{1}{1+e^{-\gamma x}}, \tag{2.31}$$

where γ is a positive real parameter, which changes the slope of the function at the point $x = 0$. The complex distribution is derived from the real one (for QPSK neurons):

$$F^{\log}(z,\gamma) = \frac{1}{1+e^{-\gamma \mathrm{Re}\{z\}}}\frac{1}{1+e^{-\gamma \mathrm{Im}\{z\}}}, \tag{2.32}$$

which equals (2.31) multiplied by itself. Thus the logistic distribution in the complex domain represents independent logistic distribution on both the real and imaginary axes.

Since $F^{\log}(x,\gamma) = 1 - F^{\log}(-x,\gamma)$, the real logistic distribution function (2.31) satisfies the previously defined (2.10) symmetry condition ($\mathsf{d}F^{\log}(x,\gamma) = \mathsf{d}F^{\log}(-x,\gamma)$). Moreover, the complex version, (2.32) also satisfies (2.21).

We would like to show that if one uses (2.32) distribution for QPSK neurons (or (2.31) for binary neurons) to generate random values of ν, then the optimal state has asymptotically the greatest probability in equilibrium. The first part of the proof reiterates some part of the Boltzmann machine theory, whereas the second part points out the equivalence between Boltzmann machine and stochastic recurrent neural network.

Lemma 2.8 *For binary or QPSK alphabets, for each $\boldsymbol{v} \in \mathcal{A}^M$ the energy function $E[\boldsymbol{v}] = \boldsymbol{v}^{\mathsf{H}}\boldsymbol{W}\ \boldsymbol{v} - \boldsymbol{u}^{\mathsf{H}}\boldsymbol{v} - \boldsymbol{v}^{\mathsf{H}}\boldsymbol{u}$ defined in (2.2) can be rewritten as follows*

$$E[\boldsymbol{v}] = 2\,\mathrm{Re}\left\{ v_l^* \left(\sum_{j,j\neq l} W_{lj}v_j - u_l \right) \right\} + \sum_{i,i\neq l}\sum_{j,j\neq l} v_i^* W_{ij} v_j + W_{ll} - 2\,\mathrm{Re}\left\{ \sum_{i,i\neq l} u_i^* v_i \right\}. \tag{2.33}$$

Proof: Let us use the notation $E[\boldsymbol{v}]$ defined in (2.3). It is easy to see that from any component l,

$$\begin{aligned}
E[\boldsymbol{v}] &= \sum_i\sum_j v_i^* W_{ij} v_j - \sum_i u_i^* v_i - \sum_i v_i^* u_i \\
&= \sum_{i,i\neq l}\sum_j W_{ij}v_i^* v_j + \sum_j W_{lj} v_l^* v_j - \sum_{i,i\neq l} u_i^* v_i - u_l^* v_l - \sum_{i,i\neq l} v_i^* u_i - v_l^* u_l \\
&= \sum_{i,i\neq l}\sum_{j,j\neq l} W_{ij}v_i^* v_j + \sum_{j,j\neq l} W_{lj} v_l^* v_j + \sum_{i,i\neq l} W_{il} v_l v_i^* + W_{ll}|v_l|^2 \\
&\quad - \sum_{i,i\neq l} u_i^* v_i - u_l^* v_l - \sum_{i,i\neq l} v_i^* u_i - v_l^* u_l.
\end{aligned}$$

Because matrix $\boldsymbol{W}$ is Hermitian ($W_{il} = W_{li}^*$), the second and third term are complex conjugate pairs in the above equation. Due to QPSK (or binary) states $|v_l|^2$ always equals plus one. After reordering the terms, the equation yields (2.33). □

The following events are introduced

$$\begin{aligned}
A &: v_1[\ell+1] = v_1^i, v_2[\ell+1] = v_2^i, \ldots, v_l[\ell+1] = v_l^i, v_{l+1}[\ell] = v_{l+1}^j, \ldots, \\
B &: v_1[\ell+1] = v_1^i, v_2[\ell+1] = v_2^i, \ldots, v_{l-1}[\ell+1] = v_{l-1}^i, v_{l+1}[\ell] = v_{l+1}^j, \ldots,
\end{aligned}$$

where $\boldsymbol{v}^i$ and $\boldsymbol{v}^j$ are fixed vectors. Event A and B differs in the knowledge of the lth component. Furthermore, we define two new vectors ($\boldsymbol{v}^A$ and $\boldsymbol{v}^B[x]$), as follows:

$$\begin{aligned}
\boldsymbol{v}^A &= [v_1^i, \ldots, v_{l-1}^i, v_l^i, v_{l+1}^j, \ldots, v_M^j]^{\mathsf{T}}, \\
\boldsymbol{v}^B[x] &= [v_1^i, \ldots, v_{l-1}^i, x, v_{l+1}^j, \ldots, v_M^j]^{\mathsf{T}}.
\end{aligned}$$

Note that the latter vector has an argument which determines the lth component in the vector. Based on the definitions and expression (2.33), one can easily check that

$$E[\boldsymbol{v}^A] = 2\,\mathrm{Re}\left\{(v_l^i)^*\left(\sum_{k=1}^{l-1} W_{lk}v_k^i + \sum_{k=l+1}^{M} W_{lk}v_k^j - u_l\right)\right\} + E_0,$$

and

$$E[\boldsymbol{v}^B[x]] = 2\,\mathrm{Re}\left\{x^*\left(\sum_{k=1}^{l-1} W_{lk}v_k^i + \sum_{k=l+1}^{M} W_{lk}v_k^j - u_l\right)\right\} + E_0,$$

where E_0 is the part of the energy function which remains constant during the update of the lth neuron:

$$\begin{aligned} E_0 = &\sum_{m=1}^{l-1}\sum_{k=1}^{l-1} W_{mk}(v_m^i)^* v_k^i + \sum_{m=1}^{l-1}\sum_{k=l+1}^{M} W_{mk}(v_m^i)^* v_k^j + \sum_{m=l+1}^{M}\sum_{k=1}^{l-1} W_{mk}(v_m^j)^* v_k^i \\ &+ \sum_{m=l+1}^{M}\sum_{k=l+1}^{M} W_{mk}(v_m^j)^* v_k^j + W_{ll} - 2\,\mathrm{Re}\left\{\sum_{k=1}^{l-1} u_k^* v_k^i + \sum_{k=l+1}^{M} u_k^* v_k^j\right\}. \end{aligned} \tag{2.34}$$

In the following theorem we assume that the stationary distribution is given as an exponential function of the energy function. Later we show that the stationary distribution of the stochastic recurrent neural network yields the same form.

Theorem 2.9 *Assuming that in thermal equilibrium the probability of a given vector $\boldsymbol{v}^i$ is given as*

$$\Pr\left\{\boldsymbol{v}[\ell] = \boldsymbol{v}^i\right\} = \frac{1}{Z} e^{-\frac{E[\boldsymbol{v}^i]}{T[\ell]}}, \tag{2.35}$$

where

$$Z = \sum_{\boldsymbol{x}\in\mathcal{A}^M} e^{-\frac{E(\boldsymbol{x})}{T[\ell]}},$$

and $T[\ell]$ is an arbitrary real number (it is used to model the effect of the temperature in the Boltzmann Machine), then the state transition probability $P_{ij}[\ell]$ equals the product of functions $F^{\log}(x,\gamma)$ (defined in (2.31) for binary alphabet, and in (2.32) for QPSK alphabet), each with an appropriate parameter.

Proof: It is easy to see that the probability of event B using (2.35)

$$\Pr\{B\} = \sum_{x\in\mathcal{A}} \Pr\left\{\boldsymbol{v}[\ell] = \boldsymbol{v}^B[x]\right\} = \frac{1}{Z}\sum_{x\in\mathcal{A}} e^{-\frac{E[\boldsymbol{v}^B[x]]}{T[\ell]}},$$

because the neurons' output can take only one value. Substituting (2.33) it finally becomes

$$\Pr\{B\} = \frac{1}{Z}\cdot e^{-\frac{E_0}{T[\ell]}} \sum_{x\in\mathcal{A}} e^{-\frac{2\,\mathrm{Re}\left\{x^*\left(\sum_{k=1}^{l-1} W_{lk}v_k^i + \sum_{k=l+1}^{M} W_{lk}v_k^j - u_l\right)\right\}}{T[\ell]}} \tag{2.36}$$

where E_0 is defined in (2.34). The conditional probability $\Pr\{A \mid B\}$ can be expressed as

$$\Pr\{A \mid B\} = \frac{\Pr\{A\cap B\}}{\Pr\{B\}} = \frac{\Pr\{A\}}{\Pr\{B\}},$$

since event A involves event B. Using (2.33) and (2.36) and simplifying with the common terms, the equation can be rewritten into

$$\Pr\{A \mid B\} = \frac{\exp\left[-\frac{2\,\mathrm{Re}\left\{(v_l^i)^*\left(\sum_{k=1}^{l-1} W_{lk}v_k^i + \sum_{k=l+1}^{M} W_{lk}v_k^j - u_l\right)\right\}}{T[\ell]}\right]}{\sum_{x\in\mathcal{A}} \exp\left[-\frac{2\,\mathrm{Re}\left\{x^*\left(\sum_{k=1}^{l-1} W_{lk}v_k^i + \sum_{k=l+1}^{M} W_{lk}v_k^j - u_l\right)\right\}}{T[\ell]}\right]}.$$

Now, dividing by the numerator one can obtain

$$\Pr\{A \mid B\} = \frac{1}{1 + \sum\limits_{x \in \mathcal{A}, x \neq v_l^i} \exp\left[-\frac{2\,\mathrm{Re}\left\{(v_l^i - x)^*\left(u_l - \sum_{k=1}^{l-1} W_{lk} v_k^i - \sum_{k=l+1}^{M} W_{lk} v_k^j\right)\right\}}{T[\ell]}\right]}.$$

In the binary case, the sum represents only one term, where $v_l^i - x = 2v_l^i$. Substituting the definition of (2.31) with parameter $\gamma = 4W_{ll}/T[\ell]$, one arrives at

$$\Pr\{A \mid B\} = F^{\log}\left(\frac{v_l^i}{W_{ll}}\left(u_l - \sum_{k=1}^{l-1} W_{lk} v_k^i - \sum_{k=l+1}^{M} W_{lk} v_k^j\right), \frac{4W_{ll}}{T[\ell]}\right). \tag{2.37}$$

In the QPSK case, some more calculation is needed to derive similar expression (see Lemma A.3 with $f(x) = e^x$ on page 67 for details). However, applying the definition of (2.32) with parameter $\gamma = \sqrt{2^3}W_{ll}/T[\ell]$, one gets

$$\Pr\{A \mid B\} = F^{\log}\left(\frac{(v_l^i)^* e^{j\pi/4}}{W_{ll}}\left(u_l - \sum_{k=1}^{l-1} W_{lk} v_k^i - \sum_{k=l+1}^{M} W_{lk} v_k^j\right), \frac{\sqrt{2}^3 W_{ll}}{T[\ell]}\right). \tag{2.38}$$

Note that the $\Pr\{A \mid B\}$ probability equals $G[i, j, k, \ell]$ in (2.15) with the above selection of $\boldsymbol{v}^i$ and $\boldsymbol{v}^j$ parameters. Following the argument of (2.15), the state transition probability becomes

$$P_{ij}[\ell] = \prod_{l=1}^{M} F^{\log}\left(\frac{v_l^i}{W_{ll}}\left(u_l - \sum_{k=1}^{l-1} W_{lk} v_k^i - \sum_{k=l+1}^{M} W_{lk} v_k^j\right), \frac{4W_{ll}}{T[\ell]}\right) \tag{2.39}$$

in the binary case and

$$P_{ij}[\ell] = \prod_{l=1}^{M} F^{\log}\left(\frac{(v_l^i)^* e^{j\pi/4}}{W_{ll}}\left(u_l - \sum_{k=1}^{l-1} W_{lk} v_k^i - \sum_{k=l+1}^{M} W_{lk} v_k^j\right), \frac{\sqrt{2}^3 W_{ll}}{T[\ell]}\right) \tag{2.40}$$

in the QPSK case.

Thus the state transition probability follows the multiplication of distribution functions $F^{\log}(z, \gamma)$ with parameters $\gamma^{\mathrm{bin}} = 4W_{ll}/T[\ell]$ and $\gamma^{\mathrm{QPSK}} = \sqrt{2^3}W_{ll}/T[\ell]$, respectively. □

Starting from the assumption that the probability of a vector state is given as an exponential function of the corresponding energy, we arrived at a state transition probability similar to the one obtained by brute force. Thus, our result leads to conclude that the assumption is right.

Theorem 2.10 *If one uses the distribution function (2.31) or (2.32) with proper parameters to generate values of the noise (ν) in the stochastic recurrent neural network, then the global optimum of the energy function (2.2) can be reached with maximal probability.*

Proof: If the state transition matrices of two systems are equal to each other, then the stationary distribution vectors must equal as well. Comparing (2.39) with (2.19) and (2.40) with (2.22) one can see that the two state transition matrices are completely the same with proper parameter selection. Thus the stochastic recurrent neural network has the stationary distribution defined in (2.35), which is maximized by the minimal energy function—the optimal solution of the quadratic form (2.3). □

The simple example (continued) Now consider the example on page 16, and check the statement of the theorem. The energy function results in the following values:

$$E[\boldsymbol{v}^1] = 4.4$$
$$E[\boldsymbol{v}^2] = 0.4$$
$$E[\boldsymbol{v}^3] = 1.2$$
$$E[\boldsymbol{v}^4] = 2.0,$$

The free parameter of (2.35) is T. While γ was chosen to equal one, and $\boldsymbol{W}$ contains ones in the diagonal, T must be set to 4. Based on (2.35), the following distribution is got: $\boldsymbol{\pi} = [0.129, 0.35, 0.287, 0.235]^{\mathsf{T}}$, which is equal to the one obtained on page 16.

2.2.5 The Inhomogeneous Case

Up till now, we have shown that applying (2.31) or the noise elements inside the neurons results in a maximal stationary probability for the global optimum for binary and QPSK alphabets. Now we extend our investigation to the inhomogeneous case, where we decrease the variance of the noise added to the iteration equation. This obviously defines an inhomogeneous Markov chain.

Now we show that applying the stochastic recurrent neural network algorithm with the proposed distribution function (2.31), the method asymptotically arrive at the optimal value with one probability.

Theorem 2.11 *If the distribution function of ν noise values is equal to (2.31) for binary neurons and (2.32) for QPSK neurons, where γ tends to infinity, then the global optimum ($\boldsymbol{v}^{\mathrm{opt}}$) of the associated quadratic form (2.2) (see (2.14)) is reached asymptotically with one probability.*

Proof: Let us evoke (2.35) which is proven to be the stationary distribution of the stochastic recurrent neural network in Theorem 2.10

$$\Pr\left\{\boldsymbol{v}[\ell]=\boldsymbol{v}^i\right\}=\frac{e^{-\frac{E[\boldsymbol{v}^i]}{T[\ell]}}}{\sum\limits_{\boldsymbol{x}\in\mathcal{A}^M} e^{-\frac{E[\boldsymbol{x}]}{T[\ell]}}}.$$

Now we divide with the numerator

$$\Pr\left\{\boldsymbol{v}[\ell]=\boldsymbol{v}^i\right\}=\frac{1}{1+\sum\limits_{\boldsymbol{x}\in\mathcal{A}^M,\boldsymbol{x}\neq\boldsymbol{v}^i} e^{\frac{1}{T[\ell]}(E[\boldsymbol{v}^i]-E[\boldsymbol{x}])}}. \tag{2.41}$$

We examine the exponential term. Since γ is inversely proportional to $T[\ell]$, if γ tends to infinity, $T[\ell]$ must tend to zero (as is the case of Boltzmann Machine). It is obvious that

$$\begin{aligned}&\text{if } E[\boldsymbol{x}]<E[\boldsymbol{v}^i], \text{ then } \lim\nolimits_{T[\ell]\to 0} e^{\frac{1}{T[\ell]}\left(E[\boldsymbol{v}^i]-E[\boldsymbol{x}]\right)}\to\infty\\ &\text{if } E[\boldsymbol{x}]>E[\boldsymbol{v}^i], \text{ then } \lim\nolimits_{T[\ell]\to 0} e^{\frac{1}{T[\ell]}\left(E[\boldsymbol{v}^i]-E[\boldsymbol{x}]\right)}\to 0\end{aligned}$$

which entails that if there is an $\boldsymbol{x}$ vector which satisfies the first condition, the probability defined in (2.41) becomes zero. Similarly if there does not exist any $\boldsymbol{x}$ vector which satisfies the second condition (i. e. minimum energy belongs to the vector $\boldsymbol{v}^i=\boldsymbol{v}^{\mathrm{opt}}$, i. e. it is the optimal solution), the probability defined in (2.41) becomes one. Consequently, the optimal solution has one probability after infinite iterations. □

As a result, the stochastic recurrent neural network can *asymptotically* find the optimal solution of the corresponding quadratic form (2.2) independently of the system parameters and the applied cooling schedule.

The simple example (continued) Let us increase parameter γ to 16 in the example on page 16. Based on (2.35) ($T=1/4$), the following stationary distribution is got: $\boldsymbol{\pi}=[0,0.96,0.04,0]^{\mathsf{T}}$. Further increment of γ results in almost one probability at $\boldsymbol{v}^2$.

Two remarks should be added here. First of all, note that our result valid only asymptotically. In real cases the time limit causes imperfect operation. Quick decrease of the temparature $T[\ell]$ yields stucking in local minima. Decreasing the temperature $T[\ell]$ too slowly results in more iterations. A trade-off must be found to balance the number of iterations and the performance of the stochastic recurrent neural network.

Secondly, zero probability does not prevent the existence of the corresponding states. Due to the unbounded nature of the noise, for an arbitrary large γ value, every state can occur during the iterations—however, zero probability states happen rarely (finitely many times).

An unlucky example Let us investigate the following two neuron ($M=2$) stochastic recurrent neural network, with parameters:

$$\boldsymbol{W}=\begin{bmatrix}1.0 & 0.6\\ 0.6 & 1.0\end{bmatrix}, \boldsymbol{u}=\begin{bmatrix}0.148\\ 0.149\end{bmatrix}.$$

The energies of vectors $\boldsymbol{v}^1 = [+1, -1]^\mathsf{T}$ and $\boldsymbol{v}^2 = [-1, +1]^\mathsf{T}$ are very close to each other, yielding almost similar probabilities. Even for $\gamma = 300$ ($T = 1/75$), $\Pr\left\{\boldsymbol{v}^1\right\} = 0.426$ and $\Pr\left\{\boldsymbol{v}^2\right\} = 0.574$ are almost equal. Since these two vectors are the farthest in Hamming distance from each other, this situation is quite unlucky. In this environment, the γ should be increased to a very high value to obtain one probability at $\boldsymbol{v}^2$.

In general, the closer two energies of two vectors are, the less effective the stochastic recurrent neural network becomes. This is the reason why the cooling schedule plays an important role in the operation of the stochastic recurrent neural network.

2.2.6 Possible Applications

The stochastic recurrent neural network can be applied for any exercise, where a quadratic form must be minimized, i. e. the problem is formulated as (2.14). It is impossible to list all of them, instead, a few is named:

- Multi-user detection (see Section 4.5.1)
- Pattern recognition
- Picture enhancement, sharpening of blured images
- Travelling salesman problem (TSP)
- Resource allocation in dependent systems (e. g. logistics)
- many more...

Due to the topic of this paper, only the first one is simulated in Section 5. Hopefully the introduced technique will also be used in other applications.

2.3 Alternative Decision Functions

Instead of the minimal distance decision function introduced in (1.3) one can apply more sophisticated functions to obtain better performance and/or faster convergence. The most general functions used for binary alphabet are detailed in this section. For the sake of simplicity only the binary case is considered in this section.

2.3.1 Soft Decision Function

The application of soft decision function in the iteration equation dates back to the introduction of recurrent neural networks. Soft decision function means continous output values. Generally the tangent hyperbolic function is applied:

$$\Phi_{\mathbb{R}}\{x\} = \frac{e^{\beta x} - e^{-\beta x}}{e^{\beta x} + e^{-\beta x}}, \tag{2.42}$$

where β defines the slope of the function at $x = 0$. The update of the neurons follows (2.1) (or (2.8)), with (2.42) as the decision function. Due to continous output values, usually constant output does not occur. Thus the number of iterations must be pre-defined by the user. After the end of the iterations, (1.3) is applied on the output values to obtain the estimated sequence.

The stability analyisis of the soft decision function directed recurrent neural networks is well investigated (they are stable in the sense of converging to a point). However, due to the continous nature of the output values, the optimality of the solution given by soft feedback recurrent neural networks seems mathematically untractable. Even the number of iterations cannot be determined based on the system parameters.

Usually the soft decision function provides an improved performance compared to the hard decision function directed recurrent neural network. Due to their tractability problems, the theory of recurrent neural networks with soft feedback is not covered by this thesis, though they are applied in the simulations.

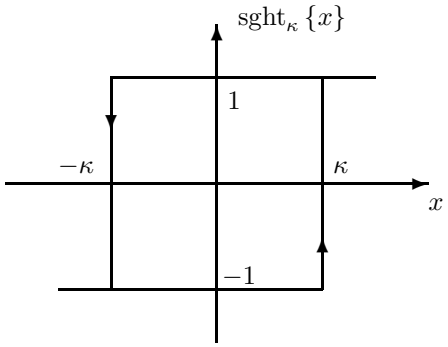

Figure 2.5: The Hysteretic Decision Function

2.3.2 Hysteresis Decision Function

Recurrent neural networks with hysteresis type nonlinearities can provide fast convergence, since hysteresis prevents output changes in the case of small input values. Hysteretic type recurrent neural networks was introduced by Levendovszky et al. [**?**]. The hysteretic iteration equation is given in the form of (2.1), or (2.8), but the decision function comprise hysteresis, which also purports some kind of memory. For (2.1) the following equation holds:

$$v_l[\ell+1] = \mathrm{sght}_\kappa\{x\} = \begin{cases} +1, \text{ if } x \geq \kappa \\ -1, \text{ if } x < -\kappa \\ +1, \text{ if } -\kappa \leq x < \kappa \text{ and } v_l[\ell] = +1 \\ -1, \text{ if } -\kappa \leq x < \kappa \text{ and } v_l[\ell] = -1, \end{cases} \tag{2.43}$$

where $\mathrm{sght}_\kappa\{.\}$ is a hysteretic decision function, with decision boundary κ (which must be non-negative) and x is given as

$$x = \frac{1}{W_{ll}}\left(u_l - \sum_{j=1}^{l-1} W_{lj}v_j[\ell+1] - \sum_{j=l+1}^{M} W_{lj}v_j[\ell]\right). \tag{2.44}$$

See Figure 2.5. If one sets $\kappa = 0$ the operation of the original recurrent neural network is obtained. Note that histeresis decision function (2.43) can be rewritten in the form

$$v_l[\ell+1] = \mathrm{sght}_\kappa\{x\} = \begin{cases} +1, \text{ if } x \geq -\kappa \cdot v_l[\ell] \\ -1, \text{ if } x < -\kappa \cdot v_l[\ell]. \end{cases},$$

thus it is equivalent to

$$v_l[\ell+1] = \mathrm{sgn}\{x + \kappa \cdot v_l[\ell]\}. \tag{2.45}$$

For special $\boldsymbol{W}$ matrices the uniqueness of the steady state, which is at the global optimum of the corresponding energy function, can be proven [**?**]. First let us define the so-called "eye-openness" parameter (D) in the following form

$$D = \min_i \left[\frac{W_{ii}}{\sum_{j;j\neq i}|W_{ij}|}\right]. \tag{2.46}$$

Secondly we define the surrounding hyper cubes (SHC) around verteces with parameter η ($0 < \eta < 1$), which is denoted by SHC_η:

$$\mathrm{SHC}_\eta \equiv \{\boldsymbol{x} = [x_i] : \forall i,\ \eta \leq |x_i| < 2 - \eta\}.$$

Thirdly we introduce vector $\boldsymbol{m}$ as the continuous minimal solution of the energy function (2.2), i. e. $\boldsymbol{m} = \boldsymbol{W}^{-1}\boldsymbol{u}$.

In the following the stationary behaviour is investigated, where state changes do not occur. Thus the parameter (ℓ) referring to the iteration instance will be neglected (there is no difference between $v_k[\ell]$ and $v_k[\ell+1]$).

Lemma 2.12 *In every iteration the following inequality must hold*

$$\frac{\max_i |m_i - v_i|}{D} + (m_l + \kappa) \geq 0. \tag{2.47}$$

Proof: First note that according to (2.45) and (2.44) the following equation is got:

$$v_l = \operatorname{sgn}\left\{\frac{1}{W_{ll}}\left(u_l - \sum_{j=1}^{l-1} W_{lj}v_j + \kappa W_{ll}v_l - \sum_{j=l+1}^{M} W_{lj}v_j\right)\right\}.$$

Secondly, note that the continous solution $\boldsymbol{m}$ can be used to substitute the value of u_l with $\sum_j W_{lj}m_j$, which implies

$$v_l = \operatorname{sgn}\left\{\frac{1}{W_{ll}}\left(\sum_{j=1}^{l-1} W_{lj}(m_j - v_j) + W_{ll}(m_l + \kappa \cdot v_l) + \sum_{j=l+1}^{M} W_{lj}(m_j - v_j)\right)\right\}.$$

Without the loss of generality the value of v_l is assumed to be $+1$ in the following ($v_l = -1$ yields the same inequality). It follows that

$$\frac{\sum_{j;j\neq l} W_{lj}(m_j - v_j)}{W_{ll}} + (m_l + \kappa) \geq 0,$$

where the left hand side is upper bounded by

$$\frac{\sum_{j;j\neq l} |W_{lj}| \cdot \max_i |m_i - v_i|}{W_{ll}} + (m_l + \kappa) \leq \frac{\max_i |m_i - v_i|}{\min_i \left[\frac{W_{ii}}{\sum_{j;j\neq i} |W_{ij}|}\right]} + (m_l + \kappa),$$

which from (2.46) yields (2.47). □

The result of the lemma plays an important role in the proof of optimality of the hysteresis recurrent neural network in the following theorem.

Theorem 2.13 *If there exists a* $0 < \eta < 1$ *parameter that* $\boldsymbol{m} \in \mathrm{SHC}_\eta$ *holds, and if*

$$\kappa = \eta\left(1 + \frac{1}{D}\right) - \frac{3}{D} \tag{2.48}$$

is positive, then the corresponding recurrent neural network converges to the global optimum. Moreover, reaching the optimal state can be accelerated by using hysteresis decision function with parameter κ*.*

Proof: We are going to prove that the only one steady state of the hysteresis recurrent neural network corresponds to $\operatorname{sgn}\{\boldsymbol{m}\}$, every other states are transient states. Thus, the network is a global optimizer finding the sole optimum of the energy function.

Note that from (2.48) ($\kappa > 0$ and $\eta < 1$) one gets

$$0 < \eta\left(1 + \frac{1}{D}\right) - \frac{3}{D} < 1 + \frac{1}{D} - \frac{3}{D},$$

which yields $D > 2$, i. e. the eye-openness parameter must be greater than two.

Assuming that $\boldsymbol{v} = \operatorname{sgn}\{\boldsymbol{m}\}$ is a steady state, validating (2.47) proves the assumption. To upper bound (2.47), $m_l < 2 - \eta$ and $\max_i |m_i - v_i| \leq 1 - \eta$, due to the SHC_η constraint. Thus (2.47) yields

$$\frac{1 - \eta}{D} + (2 - \eta + \kappa) > 0.$$

Substituting (2.48) it yields

$$\frac{1}{D} - \eta\left(\frac{1}{D} + 1\right) + 2 + \eta\left(1 + \frac{1}{D}\right) - \frac{3}{D} > 0,$$

which results in $D > 1$. The result is fulfilled since the eye-openness parameter is greater than two. So vector $\boldsymbol{v} = \operatorname{sgn}\{\boldsymbol{m}\}$ is a steady state.

Assuming that $\boldsymbol{v} \neq \operatorname{sgn}\{\boldsymbol{m}\}$ is also a steady state we will arrive in contradiction. Without loss of generality, assume that the lth component of vector $\boldsymbol{m}$ differs $\operatorname{sgn}\{m_l\} \neq v_l = 1$. The parameters of (2.47) are upper bounded as $m_l \leq -\eta$ and $\max_i |m_i - v_i| < 3 - \eta$, due to the SHC_η constraint. Thus (2.47) yields

$$\frac{3-\eta}{D} + (\kappa - \eta) > 0.$$

Substituting (2.48) it finally yields

$$-\eta\left(1 + \frac{1}{D}\right) + \frac{3}{D} + \eta\left(1 + \frac{1}{D}\right) - \frac{3}{D} > 0,$$

which obviously contradicts basic mathematics. Thus the assumption was wrong, there cannot be any other steady states than $\boldsymbol{v} = \operatorname{sgn}\{\boldsymbol{m}\}$.

One steady state means no local optima, so the steady state of the network corresponds to the global optimum of the energy function, thus the network is a global optimizer. □

Note that introducing hysteresis nonlinearity results in less state changes thus the hysteresis recurrent neural network requires less iterations (it is "more" stable) than the original one. In this way there is no need to prove the stability of the hysteresis recurrent neural network.