Quadratic Optimization with Stochastic Recurrent Neural Networks

Gábor Jeney, János Levendovszky,
Sándor Imre, László Pap

Department of Telecommunications
Budapest University of Technology and Economics

Outline

- Quadratic Optimization (QO) in general
- Recurrent Neural Networks (RNN)
- The Stochastic extension (SRNN)
  - Markovian description, the state transition probability matrix
  - Boltzmann Machine theory and its application for SRNN
  - Maximum probability at the global optimum if using logistic distribution

  - Asymptotically one probability at the global optimum with logistic distribution

- Numerical results in MUD

- Conclusions and future works

Quadratic Optimization

- Some kind of quadratic energy function

$$E[\boldsymbol{x}] = \boldsymbol{x}^{\mathsf{H}}\boldsymbol{A}\,\boldsymbol{x} - \boldsymbol{b}^{\mathsf{H}}\boldsymbol{x} - \boldsymbol{x}^{\mathsf{H}}\boldsymbol{b},$$

- One wants to optimize the energy function over a finite set $\mathcal{A}$ $(\forall i, x_i \in \mathcal{A})$,

$$\boldsymbol{x}^{\mathsf{opt}} = \underset{\boldsymbol{x}:\forall i, x_i \in \mathcal{A}}{\arg\min}\, \boldsymbol{x}^{\mathsf{H}}\boldsymbol{A}\,\boldsymbol{x} - \boldsymbol{b}^{\mathsf{H}}\boldsymbol{x} - \boldsymbol{x}^{\mathsf{H}}\boldsymbol{b}.$$

- Typical examples: Travelling Salesman Problem (TSP), N-queen problem, Multi-User Detection (MUD).

Conventional Recurrent Neural Networks

- The iteration equation is given as

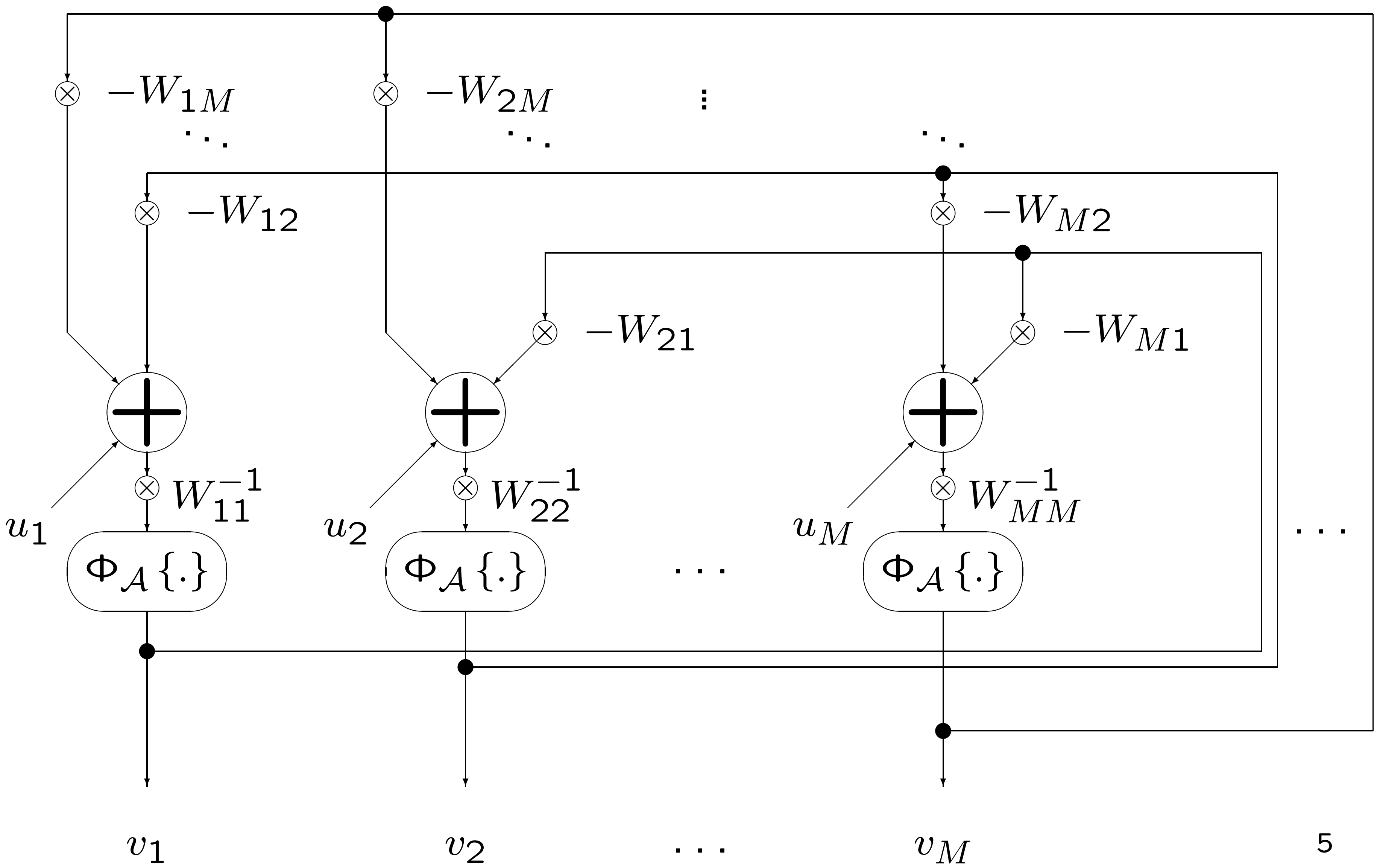
$$v_l[\ell+1] = \Phi_{\mathcal{A}}\left\{\frac{1}{W_{ll}}\left(u_l - \sum_{i=1}^{l-1} W_{li}v_i[\ell+1] - \sum_{i=l+1}^{M} W_{li}v_i[\ell]\right)\right\}.$$

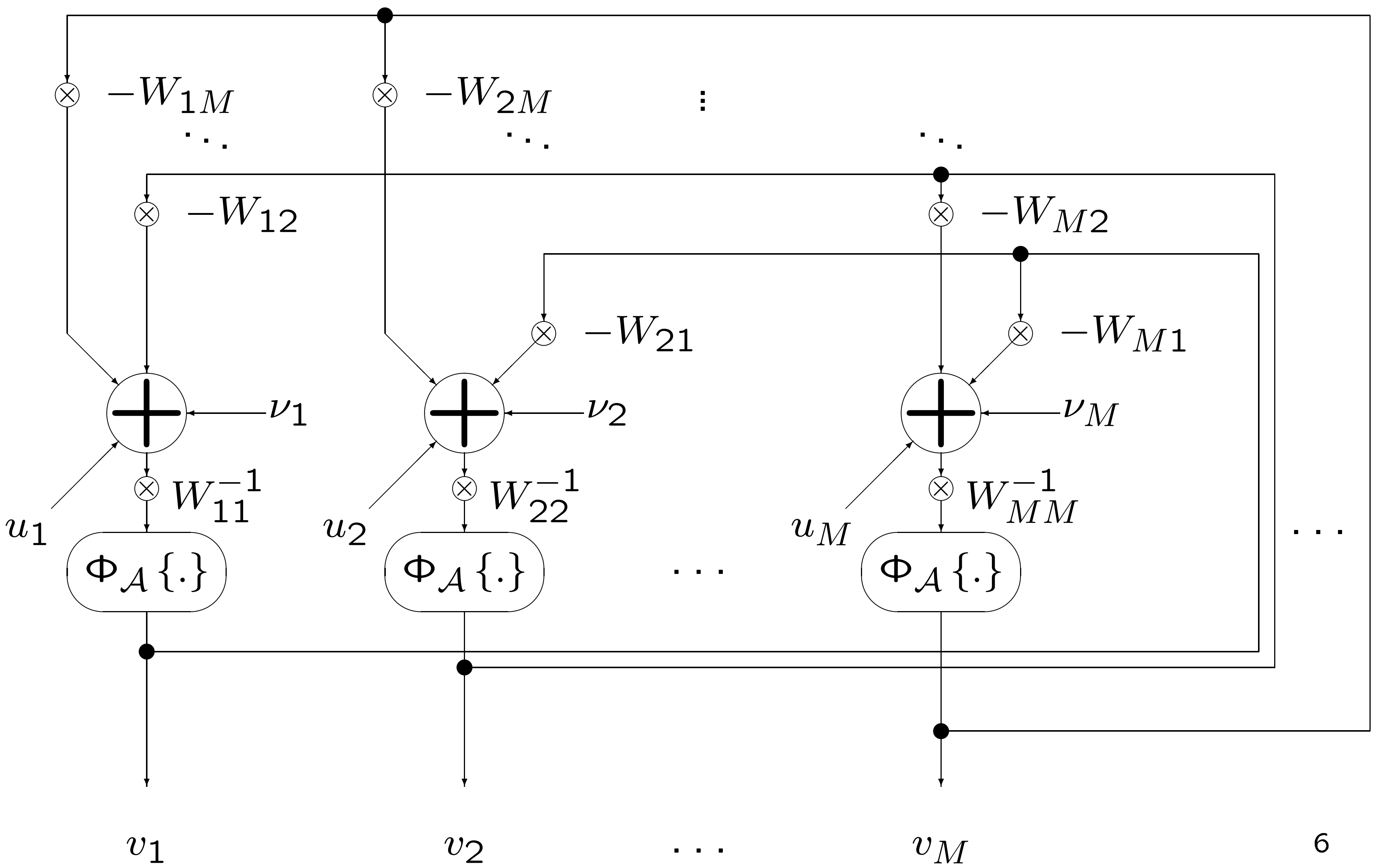
where

$$\Phi_{\mathcal{A}}\{x\} = \arg\min_{y\in\mathcal{A}} |x-y|.$$

- RNN locally minimizes its energy function

$$E[\boldsymbol{v}[\ell]] = (\boldsymbol{v}[\ell])^{\mathsf{H}}\boldsymbol{W}(\boldsymbol{v}[\ell]) - \boldsymbol{u}^{\mathsf{H}}(\boldsymbol{v}[\ell]) - (\boldsymbol{v}[\ell])^{\mathsf{H}}\boldsymbol{u},$$


$-W_{1M}$
$-W_{2M}$
$-W_{12}$
$-W_{M2}$
$-W_{21}$
$-W_{M1}$
W_{11}^{-1}
W_{22}^{-1}
W_{MM}^{-1}
u_1
u_2
u_M
$\Phi_{\mathcal{A}}\{.\}$
$\Phi_{\mathcal{A}}\{.\}$
$\Phi_{\mathcal{A}}\{.\}$
v_1
v_2
$\cdots$
v_M


$-W_{1M}$
$-W_{2M}$
$-W_{12}$
$-W_{M2}$
$-W_{21}$
$-W_{M1}$
ν_1
ν_2
ν_M
W_{11}^{-1}
W_{22}^{-1}
W_{MM}^{-1}
u_1
u_2
u_M
$\Phi_{\mathcal{A}}\{\cdot\}$
$\Phi_{\mathcal{A}}\{\cdot\}$
$\Phi_{\mathcal{A}}\{\cdot\}$
v_1
v_2
$\cdots$
v_M

Stochastic Recurrent Neural Network

- Only an additional noise inside

$$v_l[\ell+1] = \Phi_{\mathcal{A}}\left\{\frac{1}{W_{ll}}\left(u_l - \sum_{i=1}^{l-1} W_{li}v_i[\ell+1] - \sum_{i=l+1}^{M} W_{li}v_i[\ell]\right) + \nu_l[\ell]\right\}. \quad (1)$$

- The noise is characterized by a distribution function $F(z,\ell)$, i. e.

$$\text{Pr}\left\{\text{Re}\left\{\nu_l[\ell]\right\} \le \text{Re}\left\{z\right\} \cap \text{Im}\left\{\nu_l[\ell]\right\} \le \text{Im}\left\{z\right\}\right\} = F(z,\ell),$$

which should be unbounded.

- Some simmetricity assumptions are also needed in the proofs.

Markovian Description of SRNN

- State transition probabilities are defined as

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

  where vector $\boldsymbol{v}^i$ refers to a constant vector. In the same manner, the stationary probabilities ($\boldsymbol{p}[\ell+1] = \boldsymbol{P}[\ell]\,\boldsymbol{p}[\ell]$)

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

- Since the network represents an aperiodic and irreducible Markov chain, the asymptotical state transition yields

$$\boldsymbol{p} = \boldsymbol{P}\,\boldsymbol{p},$$

Derivation of the State Transition Matrix

- By brute force

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

where

$$G[i,j,k,\ell] = \mathsf{Pr}\{v_k[\ell+1] = v_k^i \mid \boldsymbol{v}[\ell] = \boldsymbol{v}^j \cap v_1[\ell+1] = v_1^i \\ \cap\, v_2[\ell+1] = v_2^i \cap \ldots \cap v_{k-1}[\ell+1] = v_{k-1}^i\}.$$

- Following the state transition rule in (1) this might be computed.

State Transition Matrices

- The binary case: with symmetric distribution ($F(x,\ell) = 1 - F(-x,\ell)$)

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

- The QPSK case: with circle symmetric distribution $\mathrm{d}F(z,\ell) = \mathrm{d}F(z \cdot e^{j\pi/2}, \ell)$

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

Global Optimization with Logistic Distribution

- The real logistic distribution function (for binary alphabet):

$$F^{\text{log}}(x,\gamma) = \frac{1}{1+e^{-\gamma x}}, \tag{2}$$

where γ is a parameter inversely proportional to the deviation.

- The complex version (for QPSK alphabet):

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

- The stationary distribution of Boltzmann Machines:

$$\Pr\left\{\lim_{\ell\to\infty} \boldsymbol{v}[\ell] = \boldsymbol{v}^i\right\} = \frac{1}{Z} e^{-\frac{E[\boldsymbol{v}^i]}{T}}, \tag{3}$$

where

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

and T is to model the effect of temperature.

- If one sets $\gamma = 2^{\frac{10-|\mathcal{A}|}{4}} W_{ll}/T$, and applies logistic distribution (2) the binary and the QPSK case yield (3).

The Inhomogenous Case

- Thanks to the open parameter T the inhomogenous behaviour can be easily examined:

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

Dividing with the numerator

$$\Pr\left\{\boldsymbol{v}=\boldsymbol{v}^i\right\}=\frac{1}{1+\sum\limits_{\boldsymbol{x}\in\mathcal{A}^M,\boldsymbol{x}\neq\boldsymbol{v}^i} e^{-\frac{1}{T}\left(E[\boldsymbol{x}]-E[\boldsymbol{v}^i]\right)}}. \tag{4}$$

- if T tends to zero, it results in one probability at the optimum, since

$$\text{if } E[\boldsymbol{x}] < E[\boldsymbol{v}^i], \text{ then } \lim_{T \to 0} e^{-\frac{1}{T}\left(E[\boldsymbol{x}] - E[\boldsymbol{v}^i]\right)} \longrightarrow \infty$$

$$\text{if } E[\boldsymbol{x}] > E[\boldsymbol{v}^i], \text{ then } \lim_{T \to 0} e^{-\frac{1}{T}\left(E[\boldsymbol{x}] - E[\boldsymbol{v}^i]\right)} \longrightarrow 0$$

Simulation Results

- The vector of received signal

$$\boldsymbol{y} = \boldsymbol{R}\boldsymbol{d} + \boldsymbol{n}, \tag{5}$$

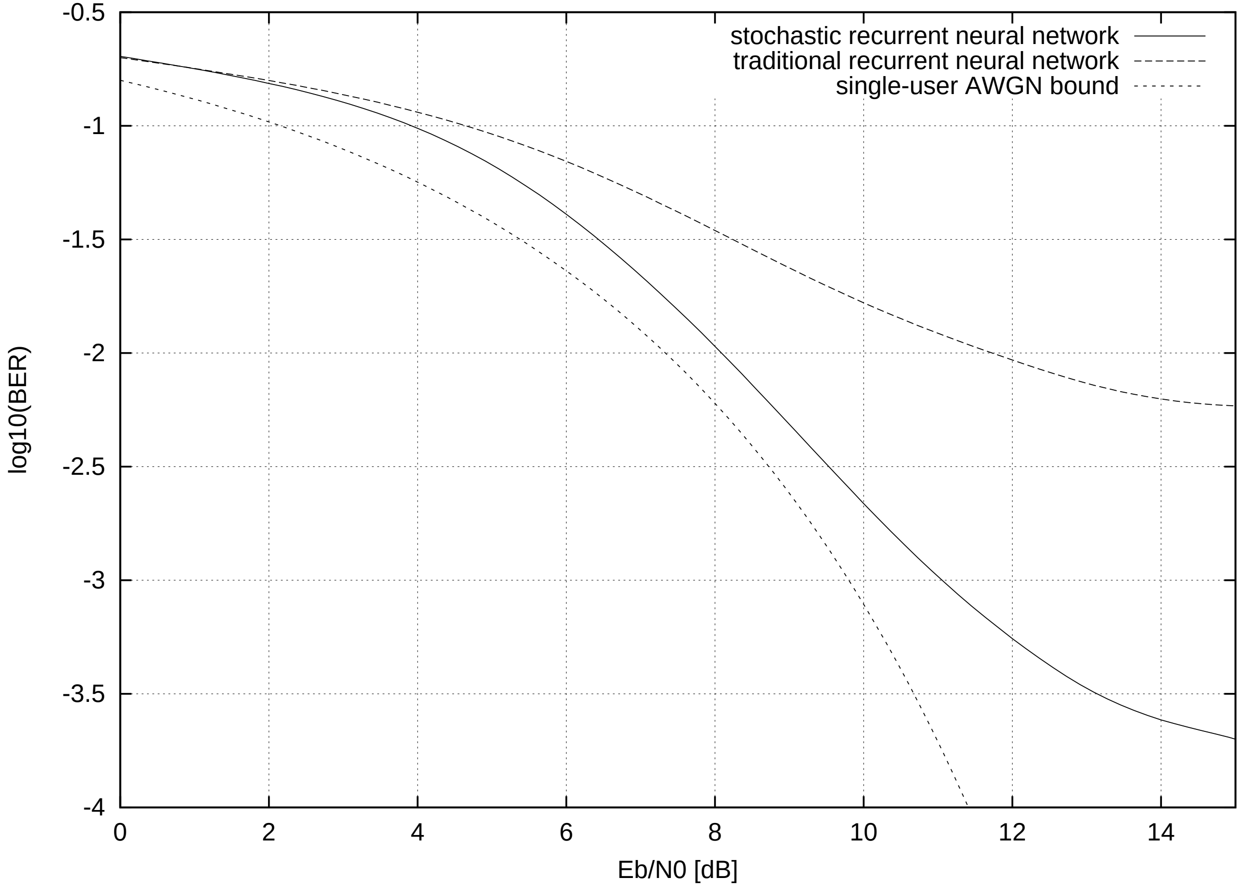
where $\boldsymbol{R} = \boldsymbol{D}^{\mathsf{T}}\boldsymbol{D}$, and $\boldsymbol{n} = \boldsymbol{D}^{\mathsf{T}}\boldsymbol{n}^{\text{white}}$.

- matrix $\boldsymbol{D}$ is randomly filled with binary values (± 1 random codes in AWGN channel with channel matched filtering)

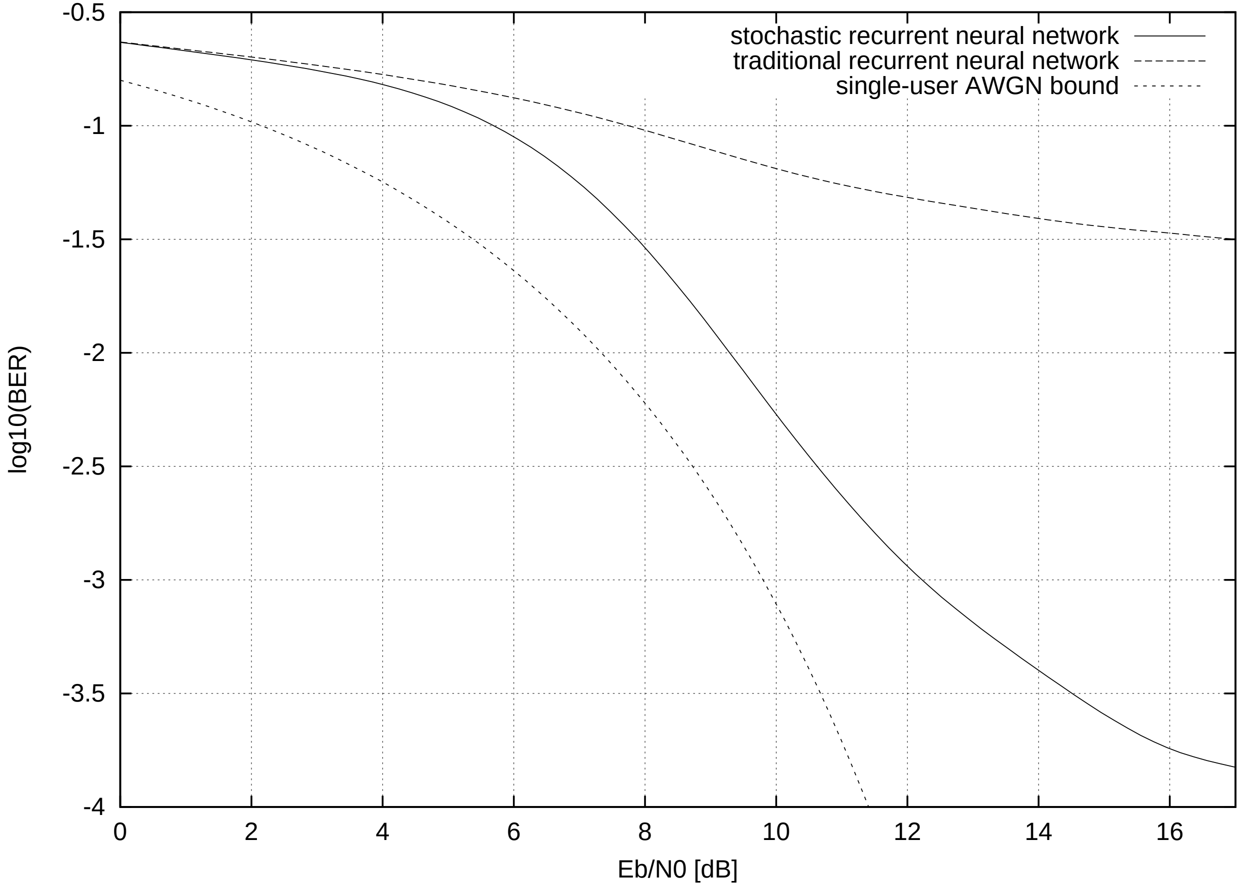
- The optimal cooling schedule is not derived yet $\ell = 1, 2, \ldots, 1000$:

$$\gamma[\ell] = 0.01 \cdot \ell.$$

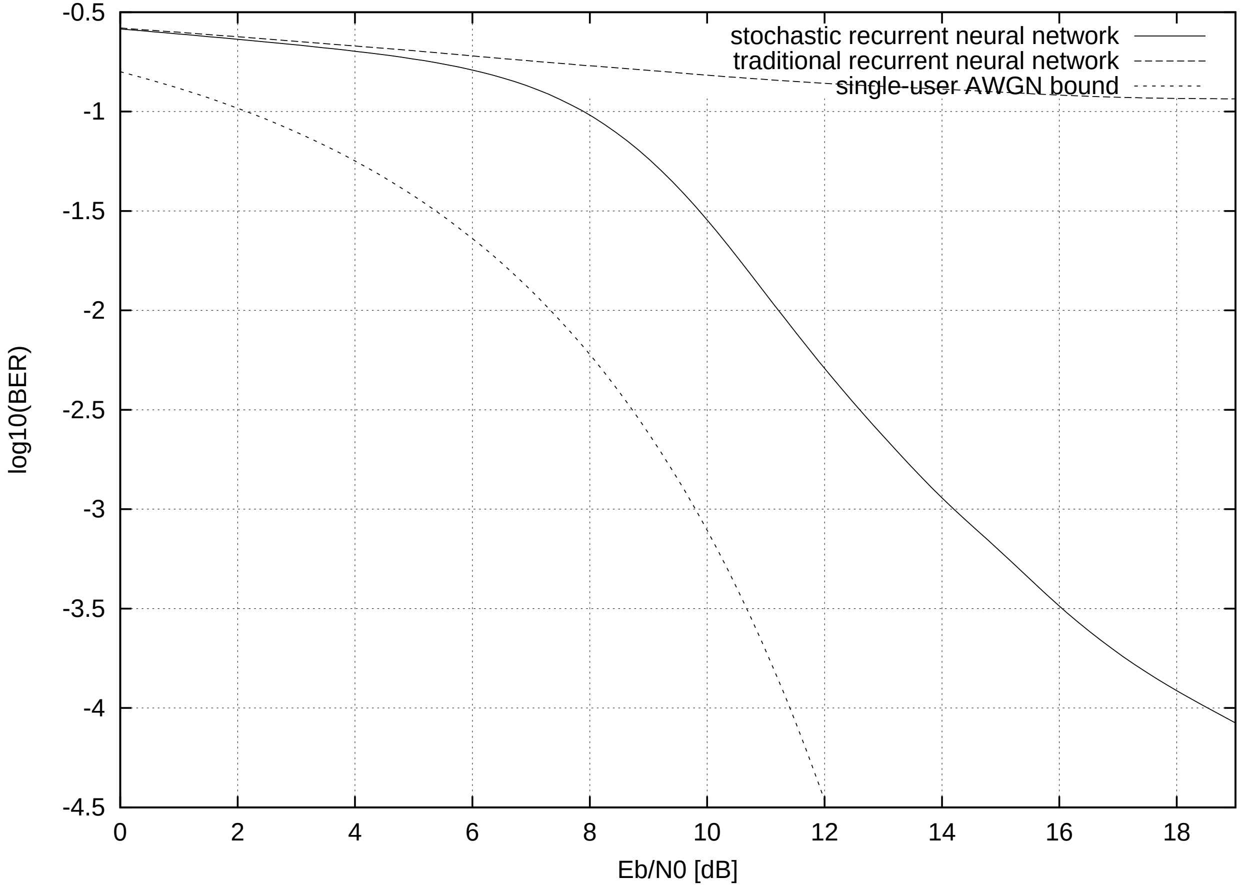
Underloaded Communication System: 100 bit codes with 80 users



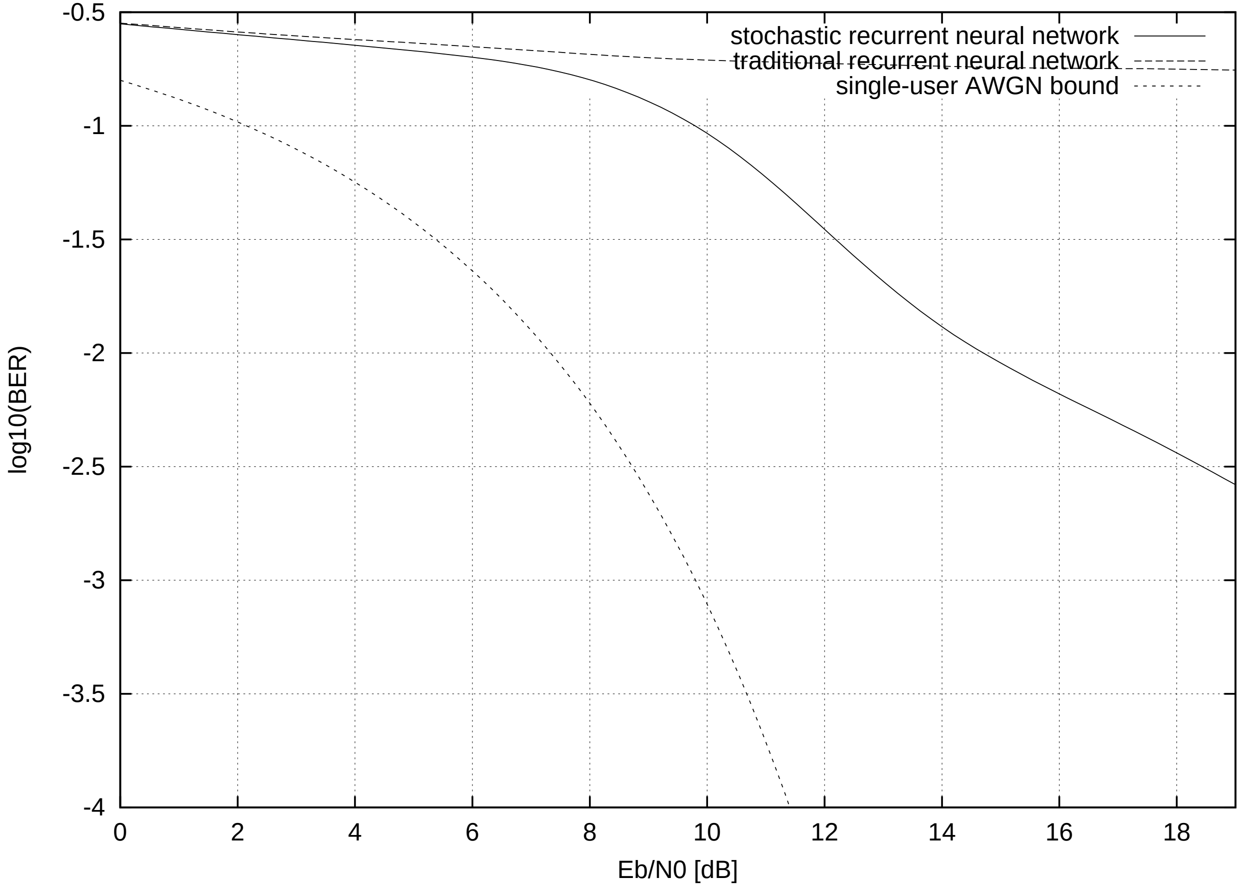
Fully Loaded Communication System: 100 bit codes with 100 users



Overloaded Communication System: 80 bit codes with 100 users



Overloaded Communication System: 70 bit codes with 100 users



Conclusions

- Discrete-time stochastic recurrent neural networks are shown to be modelled by Markov chains; the state transition probability matrix can be computed in the binary and QPSK case.

- When applying logistic distribution, the stationary probability yields maximum at the global optimum.

- In the inhomogenous case the optimum results in one probability asymptotically.

Future Works

- The optimal cooling schedule (initial value, function, number of iterations)

- Extension of the proof to the continous case

- Any comments are welcome (help me to finish my Ph. D!)