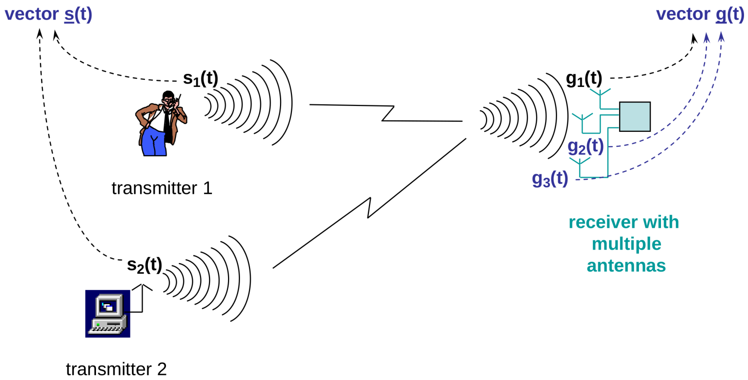
Stability Analysis of Recurrent Neural Networks with Time-Varying Activation Functions

Mohamad Mostafa, Werner G. Teich, Jürgen Lindner

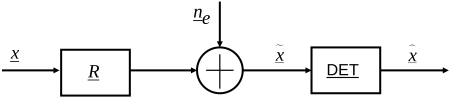
INDS'11 & ISTET'11
Klagenfurt, July 25-27, 2011

Outline

Transmission Model



Transmission Model



discrete-time channel matrix on symbol basis

vector equalizer

$$\tilde{\underline{x}} = \underline{\underline{R}} \cdot \underline{x} + \underline{n}_e$$

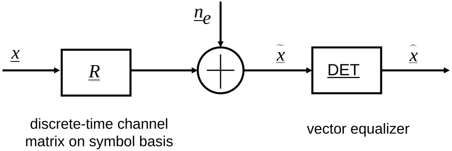
$$x_i \in \mathbb{A}_x = \{a_1, a_2, \cdots, a_M\}$$

$$a_i \in \mathbb{C},\ \tilde{\underline{x}} \in \mathbb{C}^n$$

There are M^n possible $\underline{x}$

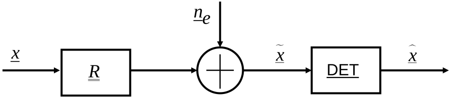
Vector Equalizer

Why a vector equalizer is needed?



Vector Equalizer

Why a vector equalizer is needed?



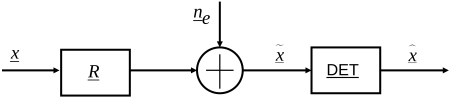
$$\underline{\tilde{x}} = \underline{\underline{R}} \cdot \underline{x} + \underline{n}_e$$

$$\begin{bmatrix} \tilde{x}_1 \\ \vdots \\ \tilde{x}_j \\ \vdots \\ \tilde{x}_n \end{bmatrix} = \begin{bmatrix} r_{11} & \dots & r_{1j} & \dots & r_{1n} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ r_{j1} & \dots & r_{jj} & \dots & r_{jn} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ r_{n1} & \dots & r_{nj} & \dots & r_{nn} \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ \vdots \\ x_j \\ \vdots \\ x_n \end{bmatrix} + \begin{bmatrix} n_{e,1} \\ \vdots \\ n_{e,j} \\ \vdots \\ n_{e,n} \end{bmatrix}$$

Vector Equalizer

Why a vector equalizer is needed?

$$\underline{\tilde{x}} = \underline{\underline{R}} \cdot \underline{x} + \underline{n}_e$$



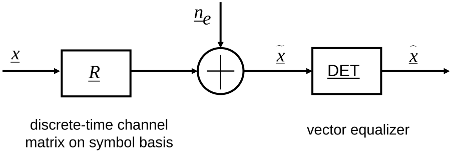
discrete-time channel matrix on symbol basis

vector equalizer

$$\begin{bmatrix} \tilde{x}_1 \\ \vdots \\ \tilde{x}_j \\ \vdots \\ \tilde{x}_n \end{bmatrix} = \begin{bmatrix} r_{11} & \dots & r_{1j} & \dots & r_{1n} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ r_{j1} & \dots & r_{jj} & \dots & r_{jn} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ r_{n1} & \dots & r_{nj} & \dots & r_{nn} \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ \vdots \\ x_j \\ \vdots \\ x_n \end{bmatrix} + \begin{bmatrix} n_{e,1} \\ \vdots \\ n_{e,j} \\ \vdots \\ n_{e,n} \end{bmatrix}$$

$$\tilde{x}_j = \underbrace{r_{jj} \cdot x_j}_{Signal} + \underbrace{\sum_{i \neq j}^{n} r_{ji} \cdot x_i}_{Interference} + \underbrace{n_{e,j}}_{Noise}$$

Vector Equalizer



Given $\underline{\tilde{x}}$ and $\underline{\underline{R}}$ what is the most likely transmitted vector?

Vector Equalizer

vector equalization

Given $\underline{\tilde{x}}$ and $\underline{\underline{R}}$ what is the most likely transmitted vector?

- Maximum Likelyhood Vector Equalizer
  - Optimal performance
  - High Complexity

There are M^n possible $\underline{x}$

Vector Equalizer

vector equalization

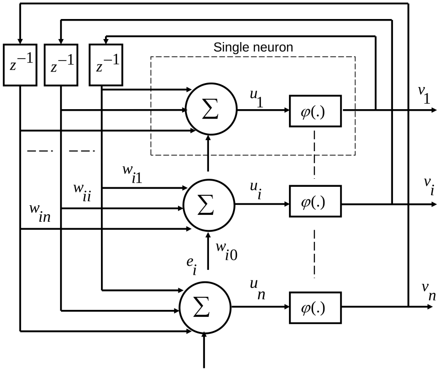
Given $\underline{\tilde{x}}$ and $\underline{\underline{R}}$ what is the most likely transmitted vector?

- Maximum Likelyhood Vector Equalizer
  - Optimal performance
  - High Complexity

There are M^n possible $\underline{x}$

- Recurrent Neural Networks
  - Good Performance
  - Less Complexity

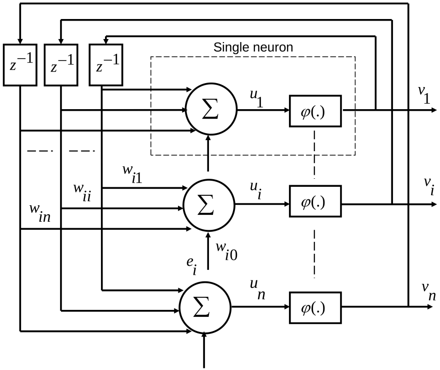
Recurrent Neural Network



$$\underline{u}(k+1) = \underline{\underline{w}} \cdot \underline{v}(k) + \underline{\underline{w}}_0 \cdot \underline{e}$$

$$\underline{v}(k) = \varphi\left[\underline{u}(k)\right]$$

Recurrent Neural Network



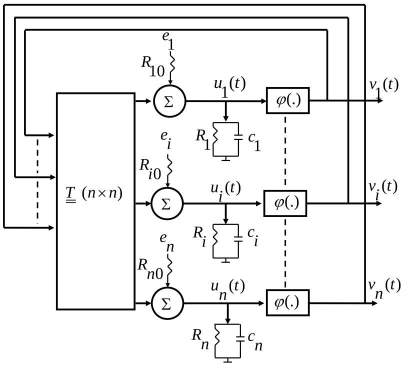
$$\underline{u}(k+1) = \underline{\underline{w}} \cdot \underline{v}(k) + \underline{\underline{w}}_0 \cdot \underline{e}$$

$$\underline{v}(k) = \varphi\left[\underline{u}(k)\right]$$

$$u_i(\rho+1) = \sum_{j=1}^{n} w_{ij} \cdot v_j(\rho) + w_{i0} \cdot e_i$$

$$v_i(\rho) = \varphi\left[u_i(\rho)\right]$$

Recurrent Neural Network



$$\underline{\underline{\tau}} \cdot \frac{\underline{u}(t)}{dt} = -\underline{u}(t) + \underline{\underline{w}} \cdot \underline{v}(t) + \underline{\underline{w}}_0 \cdot \underline{e}$$

$$\underline{v}(t) = \varphi\left[\underline{u}(t)\right]$$

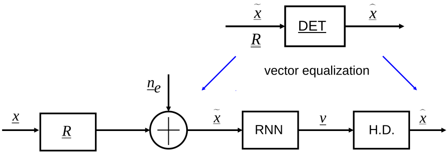
Stability of the Recurrent Neural Network

Time-invariant activation function:

- Weight matrix: $\underline{\underline{D}} \cdot \underline{\underline{w}} = \left\{ \underline{\underline{D}} \cdot \underline{\underline{w}} \right\}^T$
- $w_{ii} \geq 0$
- invertible activation function

- The RNN is stable
- Corresponding Lyapunov functions have been found

Recurrent Neural Network as Vector Equalizer



- $\underline{\underline{w}}_0 = \underline{\underline{R}}_{diag}^{-1}$
- $\underline{\underline{w}} = \underline{\underline{I}} - \underline{\underline{R}}_{diag}^{-1} \cdot \underline{\underline{R}}$
- $\underline{e} = \underline{\tilde{x}}$
- $\varphi(.)$?

$$\underline{\underline{D}} = \underline{\underline{R}}_{diag} \Rightarrow \underline{\underline{D}} \cdot \underline{\underline{w}} = \underline{\underline{R}}_{diag} - \underline{\underline{R}} \Rightarrow \underline{\underline{D}} \cdot \underline{\underline{w}} = \left\{\underline{\underline{D}} \cdot \underline{\underline{w}}\right\}^T$$

Optimal Activation Function

For real-valued modulation scheme:

- $\mathbb{A}_x = \{a_1, a_2, \cdots, a_M\}$
- $a_i \in \mathbb{R}$
- $M = 2^p, \; p \in \mathbb{N}/\{0\}$

$$v = \varphi(u) = \frac{\sum_{k=1}^{M} a_k \cdot \exp\{\beta \cdot a_k (2u - a_k)\}}{\sum_{k=1}^{M} \exp\{\beta \cdot a_k (2u - a_k)\}}$$

$0 < \varphi'(u) \leq \beta \Rightarrow \varphi(\cdot)$ is invertible

Optimal Activation Function

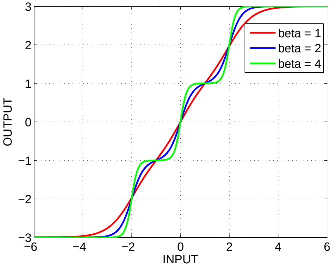
For real-valued modulation scheme:

- $\mathbb{A}_x = \{a_1, a_2, \cdots, a_M\}$
- $a_i \in \mathbb{R}$
- $M = 2^p,\ p \in \mathbb{N}/\{0\}$

$$v = \varphi(u) = \frac{\sum_{k=1}^{M} a_k \cdot \exp\{\beta \cdot a_k (2u - a_k)\}}{\sum_{k=1}^{M} \exp\{\beta \cdot a_k (2u - a_k)\}}$$

$0 < \varphi'(u) \leq \beta \Rightarrow \varphi(\cdot)$ is invertible

Example: $\mathbb{A}_x = \{-3, -1, 1, 3\}$



Slope's Adaptation

Adapting the slope β:

1. Constant slope: By simulations
2. Time-dependent slope $\beta(t)$:
   - Increasing slope: By simulations
   - Noise and Interference dependent slope $\beta(t) \propto \frac{1}{\sigma^2(t)}$

Slope's Adaptation

Adapting the slope β:

1. Constant slope: By simulations
2. Time-dependent slope $\beta(t)$:
   - Increasing slope: By simulations
   - Noise and Interference dependent slope $\beta(t) \propto \frac{1}{\sigma^2(t)}$

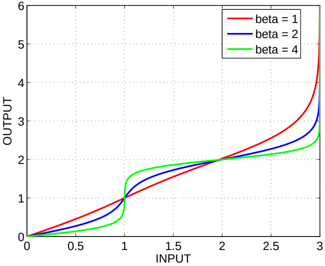
Time dependent activation functions

The stability is not proven!!

Stability with Time varying Activation Functions

$$G_\beta(x) = \int_0^x \varphi_\beta^{-1}(\psi) \cdot d\psi$$

- $G_\beta(x_2) - G_\beta(x_1) \leq G'_\beta(x_2) \cdot (x_2 - x_1)$
- $G_{\beta_1}(x) \geq G_{\beta_2}(x) \Leftrightarrow \beta_2 > \beta_1$



Stability with Time varying Activation Functions

- Take the Lyapunov function in case of time-invariant slope

$$E\left[\underline{v}(t)\right] = -\frac{1}{2}\cdot\underline{v}^T(t)\cdot\left\{\underline{\underline{D}}\cdot\underline{\underline{w}}\right\}\cdot\underline{v}(t) - \underline{e}^T\cdot\underline{\underline{D}}\cdot\underline{v}(t) + \sum_{l=1}^{n} d_{ll}\cdot G_{\beta}\left[v_l(t)\right]$$

- Modify them to time-variant slopes

$$E\left[\underline{v}(t)\right] = -\frac{1}{2}\cdot\underline{v}^T(t)\cdot\left\{\underline{\underline{D}}\cdot\underline{\underline{w}}\right\}\cdot\underline{v}(t) - \underline{e}^T\cdot\underline{\underline{D}}\cdot\underline{v}(t) + \sum_{l=1}^{n} d_{ll}\cdot G_{\beta(t)}\left[v_l(t)\right]$$

Stability with Time varying Activation Functions

- Find conditions on the slope's time variations, which keeps the stability

$$\frac{E\left[\underline{v}(t)\right]}{dt} = \sum_{l=1}^{n} \frac{\partial E\left[\underline{v}\right]}{\partial v_l} \cdot \frac{dv_l(t)}{dt}$$

$$= -\sum_{l=1}^{n} \tau_l \cdot d_{ll} \cdot \frac{du_l(t)}{dt} \cdot \frac{dv_l(t)}{dt}$$

$$\frac{E\left[\underline{v}(t)\right]}{dt} \leq 0 \Rightarrow \frac{d\beta(t)}{dt} \geq 0$$

Conclusion

- RNN as vector equalizer
- RNNs with time-variant activation function are locally stable:
  $\Rightarrow$ The slope of the activation function must be nondecreasing
- Adapting the activation function during the time:
  $\Rightarrow$ performance improvement
  $\Rightarrow$ Better local minima

Thank you for the attention, Questions?