# Learning Recurrent Dynamics
# Recurrent Learning Dynamics

Jochen J. Steil

Neuroinformatics Group
Faculty of Technology
Bielefeld University

Honda Research Institute Europe
Offenbach

04.04.2006 HRI Offenbach

# Vita

Abitur
1985

Diploma in Mathematics & Slavistics

1992

1995/96

The Neuroinformatics Group

Bielefeld University
Faculty of Technology

1999

Honda Research Institute

2006

Ochtrup

St. Petersburg


Electrotechnical University

2005/2006

Padua University

# Motivation



## Recurrent Networks

- are universal nonlinear systems

# Motivation





## Recurrent Networks

- are universal nonlinear systems
- provide generative models
- nonlinear but tractable

## Goals

- model dynamic processes and signals

# Motivation



## Recurrent Networks

- are universal nonlinear systems
- provide generative models
- nonlinear but tractable

## Goals

- model dynamic processes and signals
- study interaction of learning mechanisms
  - on different time scales
  - on different levels
  - in stability and learning

# Motivation





## Recurrent Networks

- are universal nonlinear systems
- provide generative models
- nonlinear but tractable

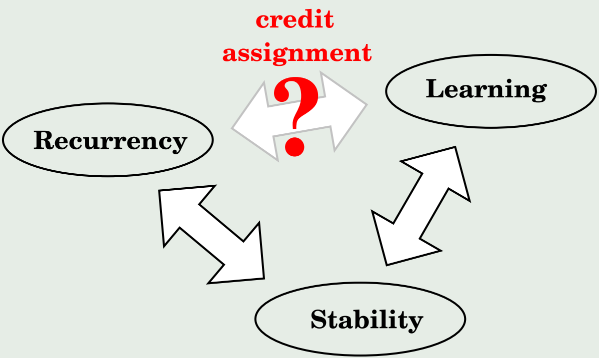
## Goals

- model dynamic processes and signals
- study interaction of learning mechanisms
  - on different time scales
  - on different levels
  - in stability and learning
- application in cognitive modeling and robots

# Interactions



## Recurrency versus Learning



# Interactions



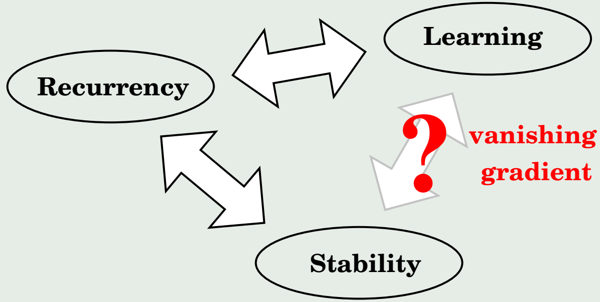
Learning Recurrent Dynamics

Motivation

BPDC Networks

Perspectives Applications

## Learning versus Stability

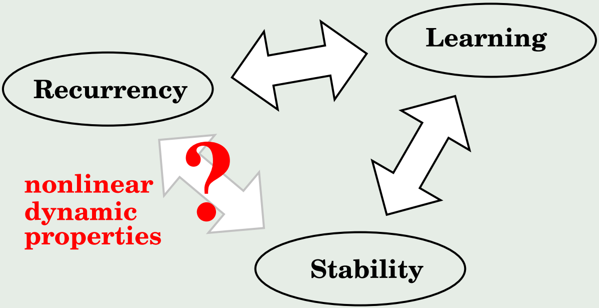


# Interactions





## Recurrency versus Stability



## Computation Based on Fixed Reservoirs

- Liquid State Machine, [Natschläger et al., Neural Computation 2002]
- Echo State Networks, [Jaeger, NIPS 2002]
- BPDC Networks, [Steil, IJCNN 2004]

## A New Approach to Recurrent Networks

- fully recurrent networks
- discrete time:

$$\vec{x}(k+1) = W \tanh(\vec{x}(k)) + \vec{u}(k)$$

- continuous time via Euler step

## Task: Learning of Time Series, Trajectories

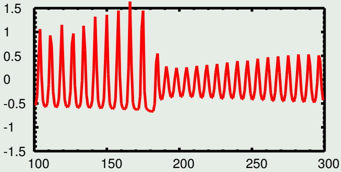
## A New Approach to Recurrent Networks

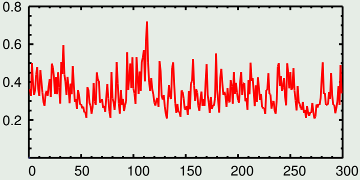
- fully recurrent networks
- discrete time:

$$\vec{x}(k+1) = W \tanh(\vec{x}(k)) + \vec{u}(k)$$

- continuous time via Euler step

## Task: Learning of Time Series, Trajectories



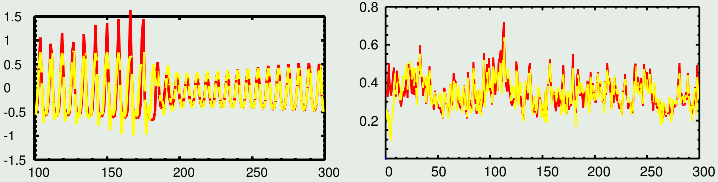
## A New Approach to Recurrent Networks

- fully recurrent networks
- discrete time:

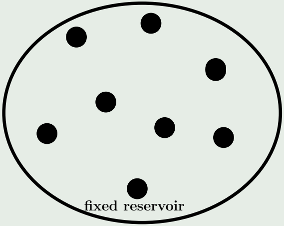
$$\vec{x}(k+1) = W \tanh(\vec{x}(k)) + \vec{u}(k)$$

- continuous time via Euler step

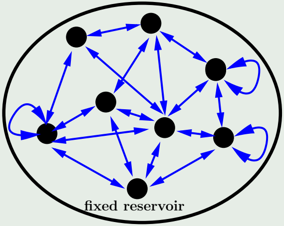
## Task: Learning of Time Series, Trajectories



# BPDC reservoir network

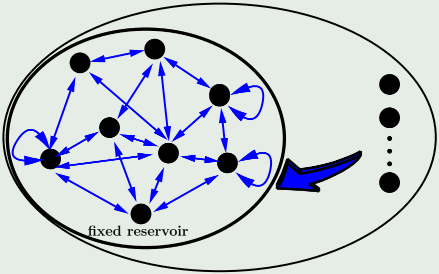


# BPDC reservoir network



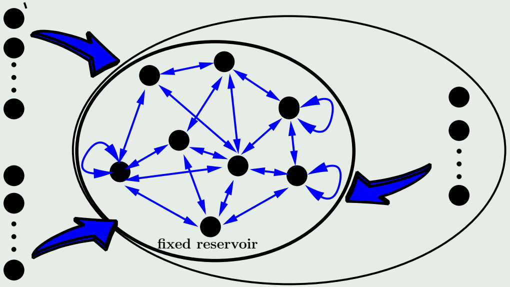

fixed connections in blue

# BPDC reservoir network



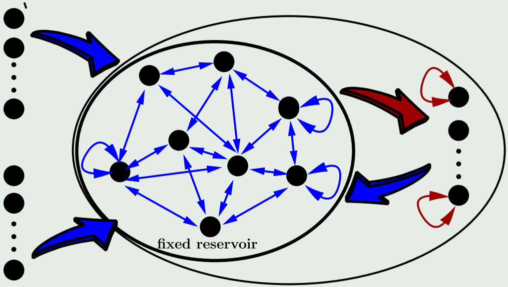

fixed connections in blue

# BPDC reservoir network




fixed connections in blue

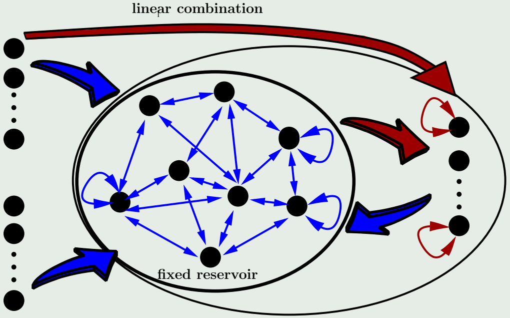
# BPDC reservoir network




fixed connections in blue

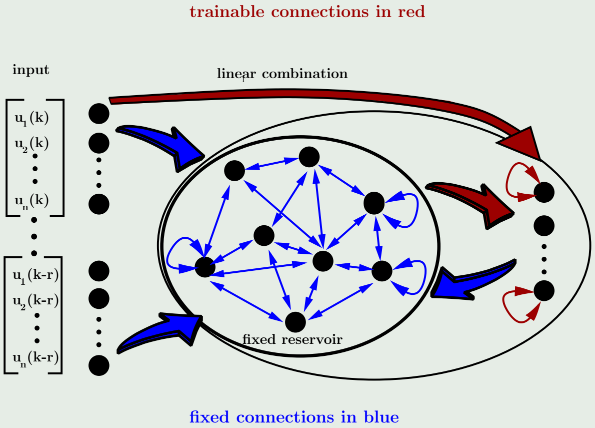
# BPDC reservoir network

trainable connections in red

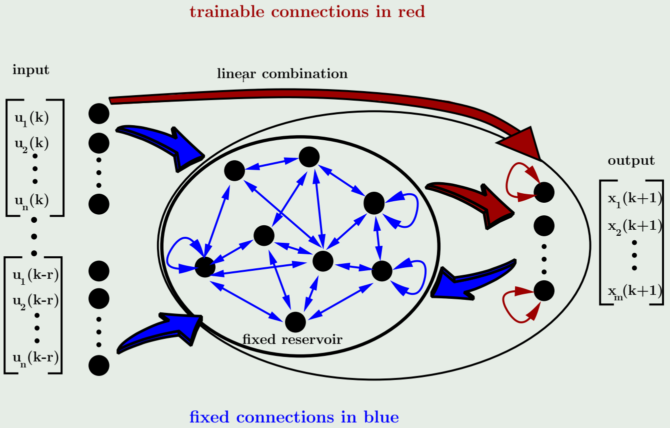



fixed connections in blue

# BPDC reservoir network



# BPDC reservoir network



# BPDC Learning





## BackPropagation-DeCorrelation Learning Rule

$$\Delta w_{1j}(k+1) = \eta \frac{\tanh(x_j(k))}{\sum_s \tanh(x_s(k))^2 + \epsilon} \gamma_1(k+1)$$

## BackPropagation-DeCorrelation Learning Rule

$$\Delta w_{1j}(k+1) = \eta \frac{\tanh(x_j(k))}{\sum_s \tanh(x_s(k))^2 + \epsilon} \gamma_1(k+1)$$

## Error Backpropagation (Error $e_1 = y_{net} - y_{target}$)

$$\gamma_1(k+1) = w_{11} \tanh'(x_1(k)) e_1(k) - e_1(k+1)$$

## BackPropagation-DeCorrelation Learning Rule

$$\Delta w_{1j}(k+1) = \eta \frac{\tanh(x_j(k))}{\sum_s \tanh(x_s(k))^2 + \epsilon} \gamma_1(k+1)$$

## Error Backpropagation (Error $e_1 = y_{net} - y_{target}$)

$$\gamma_1(k+1) = w_{11} \tanh'(x_1(k)) e_1(k) - e_1(k+1)$$

## Decorrelation Factor

$$\frac{\tanh(x_j(k))}{\sum_s \tanh(x_s(k))^2 + \epsilon} = C_k^{-1} \vec{\tanh}(\vec{x}(k))$$

$$C_k = [\tanh(\vec{x}(k))][\tanh(\vec{x}(k))]^T + \epsilon I$$

## Minimize Quadratic Error for Reference Signal $d(k)$

$$E = \sum_k (x_1(k) - d_1(k))^2$$

subject to

$$\vec{g}(k+1) = -\vec{x}(k+1) + (1-\Delta t)\vec{x}(k) + \Delta t W \tanh(\vec{x}(k)) = 0$$

## Virtual Target for States

$$\Delta \mathbf{x}_{\text{tar}} = -\left(\frac{\partial E}{\partial \mathbf{x}}\right)^T = -\left(e^T(1), \ldots, e^T(K)\right)^T,$$

$$\text{with error} \quad e_i(k) = \begin{cases} x_i(k) - d_i(k), & i = 1 \\ 0, & i \neq 1 \end{cases}$$

## Virtual Teacher Forcing

use constraint equation

$$\frac{\partial g}{\partial \mathbf{w}}\Delta \mathbf{w} + \frac{\partial g}{\partial \mathbf{x}}\Delta \mathbf{x} = 0 \quad \Rightarrow \quad \frac{\partial g}{\partial \mathbf{w}}\Delta \mathbf{w} = -\frac{\partial g}{\partial \mathbf{x}}\Delta \mathbf{x}.$$

and solve

$$\Delta \mathbf{w}_{\text{batch}} = -\eta \left(\frac{\partial g}{\partial \mathbf{w}}\right)^{\#} \frac{\partial g}{\partial \mathbf{x}}\Delta \mathbf{x}_{\text{tar}},$$

$$\Delta \mathbf{w}_{\text{batch}} = -\eta \left[\left(\frac{\partial g}{\partial \mathbf{w}}\right)^{T}\left(\frac{\partial g}{\partial \mathbf{w}}\right)\right]^{-1}\left(\frac{\partial g}{\partial \mathbf{w}}\right)^{T}\frac{\partial g}{\partial \mathbf{x}}\Delta \mathbf{x}_{\text{tar}}$$

## BPDC-Interpretation

$$\Delta \mathbf{w}_{\mathrm{batch}} = -\eta \left[ \left( \frac{\partial g}{\partial \mathbf{w}} \right)^T \left( \frac{\partial g}{\partial \mathbf{w}} \right) \right]^{-1} \left( \frac{\partial g}{\partial \mathbf{w}} \right)^T \frac{\partial g}{\partial \mathbf{x}} \Delta \mathbf{x}_{\mathrm{tar}}$$

$$= -\eta \,[\text{correlation matrix}]^{-1}\,(\text{state vector})\,(\text{error term})$$

$$= -\eta\, \text{decorrelation–backpropagation}$$

## Scaled Error Correction

$$\Delta w_{1j}(k+1) = \frac{\eta}{\sum_s \tanh(x_s(k))^2 + \epsilon} \tanh(x_j(k))\gamma_1(k+1)$$
$$= \text{scaling} \times \text{input} \times \text{error}$$

where $\gamma_1(k+1)$ is a modified error:

$$\gamma_1(k+1) = w_{11}\tanh'(x_1(k))e_1(k) - e_1(k+1)$$

Europe Honda Research Institute

## Scaled Error Correction

$$\Delta w_{1j}(k+1) = \frac{\eta}{\sum_s \tanh(x_s(k))^2 + \epsilon} \tanh(x_j(k))\gamma_1(k+1)$$
$$= \text{scaling} \times \text{input} \times \text{error}$$

where $\gamma_1(k+1)$ is a modified error:

$$\gamma_1(k+1) = w_{11} \tanh'(x_1(k))e_1(k) - e_1(k+1)$$

HRI Europe Honda Research Institute

Why such strange error ?

## Step by Step



1. $x(1), e(0) = 0$

## Step by Step

1. $x(1), e(0) = 0$
2. $e(1)$

## Step by Step

HRI Europe Honda Research Institute

1. $x(1), e(0) = 0$
2. $e(1)$
3. $\Delta x(1) \sim -\frac{\partial E}{\partial x(1)} = -e(1)$

# How does it work ?





## Step by Step

1. $x(1), e(0) = 0$
2. $e(1)$
3. $\Delta x(1) \sim -\frac{\partial E}{\partial x(1)} = -e(1)$
4. $\Delta w(1) : [x(0), w(1)] \rightarrow x(1) + \eta \Delta x(1)$

# How does it work ?



## Step by Step



1. $x(1), e(0) = 0$
2. $e(1)$
3. $\Delta x(1) \sim -\frac{\partial E}{\partial x(1)} = -e(1)$
4. $\Delta w(1) : [x(0), w(1)] \rightarrow x(1) + \eta \Delta x(1)$
5. $k = 2$ (without applying $w(1)$ step !)

## Step by Step

HRI Europe Honda Research Institute

1. $x(1), e(0) = 0$
2. $e(1)$
3. $\Delta x(1) \sim -\frac{\partial E}{\partial x(1)} = -e(1)$
4. $\Delta w(1) : [x(0), w(1)] \to x(1) + \eta \Delta x(1)$
5. $k = 2$ (without applying $w(1)$ step !)
6. $x(2) \leftarrow (w(1), x(1))$

## Step by Step

HRI Europe Honda Research Institute

1. $x(1), e(0) = 0$
2. $e(1)$
3. $\Delta x(1) \sim -\frac{\partial E}{\partial x(1)} = -e(1)$
4. $\Delta w(1) : [x(0), w(1)] \rightarrow x(1) + \eta \Delta x(1)$
5. $k = 2$ (without applying $w(1)$ step !)
6. $x(2) \leftarrow (w(1), x(1))$
7. $\gamma(2) = -e(1) +$ effect of not applied $\Delta x$ from $k = 1$
8. ...

# Intrinsic Plasticity





## Non-homogeneous Reservoir

- randomized thresholds provide richer dynamics
- randomized thresholds preserve stability

# Intrinsic Plasticity





## Non-homogeneous Reservoir

- randomized thresholds provide richer dynamics
- randomized thresholds preserve stability

## Motivates Intrinsic Plasticity

- mechanism motivated by neurobiological studies
- information maximization principle
- longer time-scale
- autonomous self-regulation
- minimize Kullback-Leiber distance to exponential distribution

## Minimize Kullback-Leibler Distance to Exponential

- use gradient rule for fermi function

$$y = \text{fermi}(x, a, b) = \frac{1}{1 + exp^{(-a*x-b)}}$$

$$\Delta b = \eta \left(1.0 - (2 + \frac{1.0}{\mu})y + \frac{1.0}{\mu}y^2\right);$$

$$\Delta a = \eta \left(\frac{1.0}{a} + x - (2 + \frac{1.0}{\mu})xy + \frac{1.0}{\mu}xy^2\right);$$

$$\Delta a = \eta \left(\frac{1.0}{a}\right) + \Delta b > 0$$

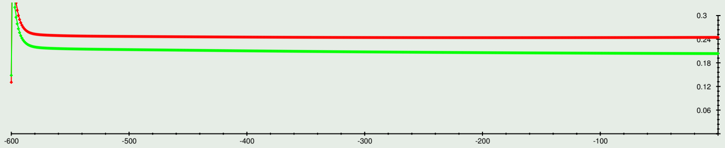
- introduced by [Triesch, ICANN 2005]

# Results

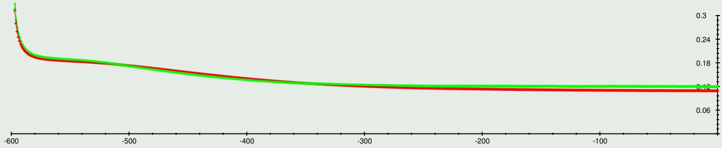




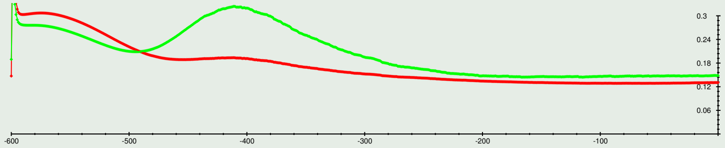
## Learning without IP



## Learning with IP



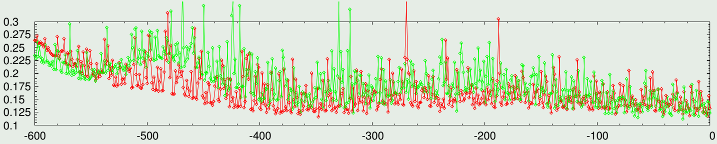
## Learning with IP/less input history

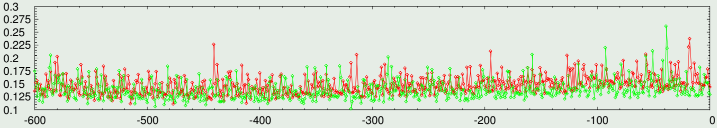


# Results



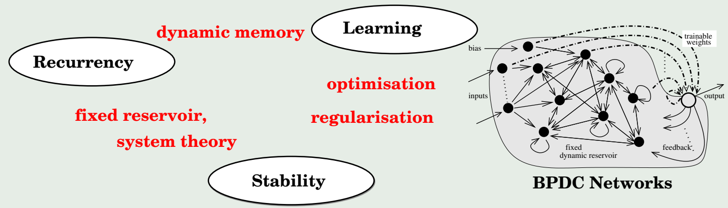
## 1/1000 random weights



# Tradeoffs

## Interactions




## Recurrent Learning Beyond Gradient Descent

- use virtual teacher → short term error propgation
- use intrinsic plasticity → longer term adaptation
- interaction of mechanisms on two scales
- stability preservation

## Three kinds of questions

- What is the nature of the encoding ?

## Three kinds of questions

- What is the nature of the encoding ?
- How do the learning mechanisms interact ?

# Perspectives at HRI



## Three kinds of questions

- What is the nature of the encoding ?
- How do the learning mechanisms interact ?
- What is represented ?

## Applications

- time series prediction

## Three kinds of questions

- What is the nature of the encoding ?
- How do the learning mechanisms interact ?
- What is represented ?

## Applications

- time series prediction
- generative modeling of data from observing humans

# Perspectives at HRI





## Three kinds of questions

- What is the nature of the encoding ?
- How do the learning mechanisms interact ?
- What is represented ?

## Applications

- time series prediction
- generative modeling of data from observing humans
- support movement perception by prediction

# Perspectives at HRI





## Three kinds of questions

- What is the nature of the encoding ?
- How do the learning mechanisms interact ?
- What is represented ?

## Applications

- time series prediction
- generative modeling of data from observing humans
- support movement perception by prediction
- data from (physics based) robot simulation HRI Europe Honda Research Institute

# Encoding





- dynamic shift between linear/nonlinear modeling

# Encoding





- dynamic shift between linear/nonlinear modeling
- multi-signal learning on one reservoir is possible

# Encoding



- dynamic shift between linear/nonlinear modeling
- multi-signal learning on one reservoir is possible
- multi-dimensional learning on one reservoir is possible HRI Europe Honda Research Institute

# Encoding





- dynamic shift between linear/nonlinear modeling
- multi-signal learning on one reservoir is possible
- multi-dimensional learning on one reservoir is possible HRI Europe Honda Research Institute
- IP tends to change neurons to threshold units HRI Europe Honda Research Institute

# Encoding



- dynamic shift between linear/nonlinear modeling
- multi-signal learning on one reservoir is possible
- multi-dimensional learning on one reservoir is possible HRI Europe Honda Research Institute
- IP tends to change neurons to threshold units HRI Europe Honda Research Institute
- measure correlation/decorrelation is difficult

# Encoding



- dynamic shift between linear/nonlinear modeling
- multi-signal learning on one reservoir is possible
- multi-dimensional learning on one reservoir is possible HRI Europe Honda Research Institute
- IP tends to change neurons to threshold units HRI Europe Honda Research Institute
- measure correlation/decorrelation is difficult

$\Rightarrow$ how generic is the feature machine ?

# Mechanisms





- IP tends to change neurons to threshold units

# Mechanisms



- IP tends to change neurons to threshold units
- local vs. global learning: how to measure contributions

# Representation



does the reservoir learn

- single trajectories ?

# Representation





does the reservoir learn

- single trajectories ?
- an operator ?

# Representation



does the reservoir learn

- single trajectories ?
- an operator ?
- the underlying system dynamics ?

# Representation





does the reservoir learn

- single trajectories ?
- an operator ?
- the underlying system dynamics ?

How to quantize, with which experiments ?

# Application





- next step: labeled multi-dimensional kinematic robot data

# Application





- next step: labeled multi-dimensional kinematic robot data
- build classification architecture based on prediction error

# Application





- next step: labeled multi-dimensional kinematic robot data
- build classification architecture based on prediction error
- what is the convincing application demonstration ?

- setup of environment NEO/NST, mysql, gcc
- implement and test of multidimensional case
- interpretation as error correction
- monitor discrete coding of IP
- clustering sequences for labeling with SOM-SD (ongoing project)