Slovak University of Technology
Faculty of Informatics and Information Technologies

Architectural Bias of Recurrent Neural Networks

Thesis Presentation

Michal Čerňanský

cernansky@fiit.stuba.sk
www.fiit.stuba.sk/~cernans

Motivation

Recurrent neural networks – processing input data with time dependant information, searching for temporal patterns

Difficult adaptation process

– common gradient-based algorithms – slow, local. min.

– approaches based on Kalman filtration

  – EKF, multistream EKF, DEKF – state of the art techniques

  – computationally expensive

Markovian architectural bias

– clusters in the RNN state space

– explain state space organization of RNN randomly initialized

– correspondence between variable length Markov models

– potentially useful – “cheap” and efficient models

Overview

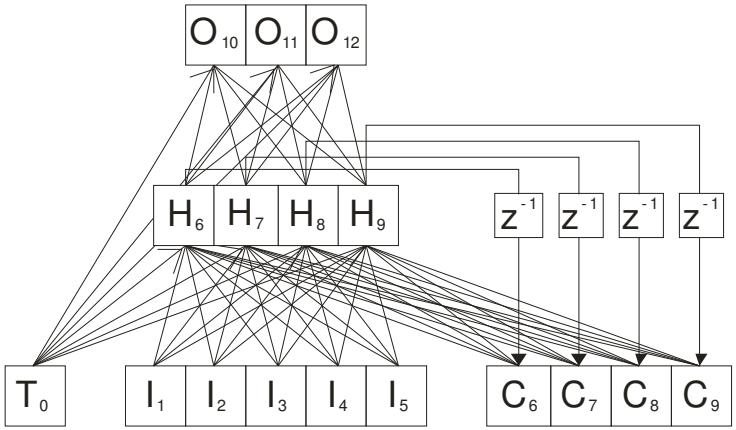
- Introduction
- Recurrent neural networks
  - Werbos notation
  - Architectures
  - Adaptation algorithms
- Architectural bias
  - IFS
  - Markov models
  - Fractal prediction machines
  - Echo state networks

- Experiments with symbolic time series
  - Datasets
  - Architectures (Elman, W&Z, Jordan, Bengio,..)
  - Adaptation algorithms (EKF,MS-EKF,BPTT, …)
  - Mark. arch. bias approaches (NPM, FPM, ESN)
  - Markov models (MM, VLMM)
- Applications (symbolic time series processing)
  - Prob. model of written text – prediction, correction
  - Biological symbolic sequences
- Conclusion
- Future work
- Appendix

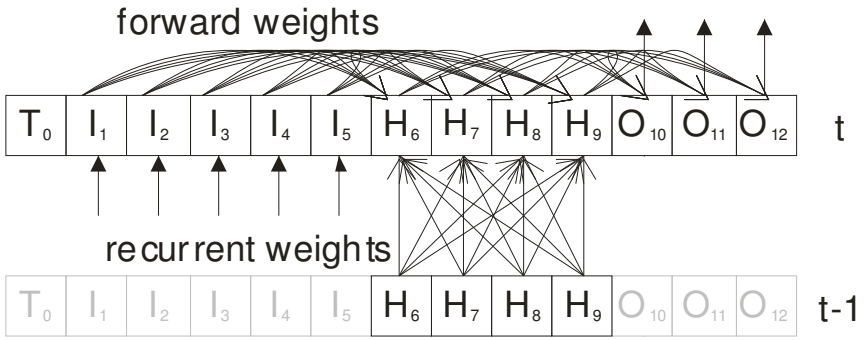
Recurrent neural networks

- Processing data with spatio-temporal structure
- RNN – recurrent multi-layer perceptron operating in discrete time
- RNN Encoding – Werbos notation
  - Elman's SRN – popular RNN architecture

a) Elman's SRN - Layered Structure



b) Elman's SRN - Werbos Representation



Elman's SRN – Werbos Representation

c) Weight Connections

Index	Source	Dest.	Delay	Value
WI	SRC	DST	TD	W
0	0	6	0	
1	1	6	0	
2	2	6	0	
3	3	6	0	
4	4	6	0	
5	5	6	0	
6	6	6	1	
7	7	6	1	
8	8	6	1	
9	9	6	1	
10	0	7	0	
11	1	7	0	
12	2	7	0	
13	3	7	0	
14	4	7	0	
15	5	7	0	
16	6	7	1	
17	7	7	1	
18	8	7	1	
19	9	7	1	

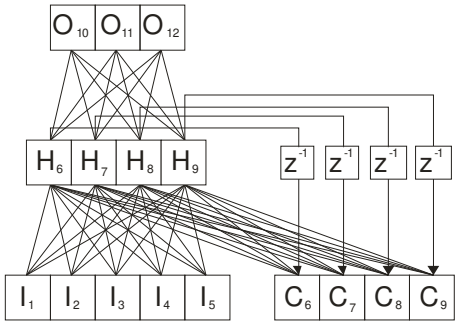
Index	Source	Dest.	Delay	Value
WI	SRC	DST	TD	W
20	0	8	0	
21	1	8	0	
22	2	8	0	
23	3	8	0	
24	4	8	0	
25	5	8	0	
26	6	8	1	
27	7	8	1	
28	8	8	1	
29	9	8	1	
30	0	9	0	
31	1	9	0	
32	2	9	0	
33	3	9	0	
34	4	9	0	
35	5	9	0	
36	6	9	1	
37	7	9	1	
38	8	9	1	
39	9	9	1	

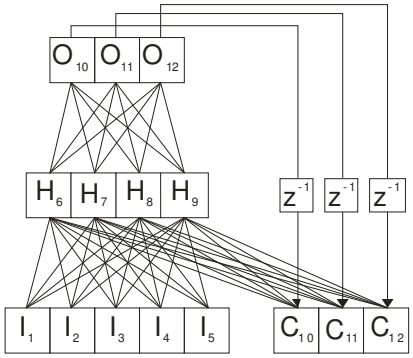
Index	Source	Dest.	Delay	Value
WI	SRC	DST	TD	W
40	0	10	0	
41	6	10	0	
42	7	10	0	
43	8	10	0	
44	9	10	0	
45	0	11	0	
46	6	11	0	
47	7	11	0	
48	8	11	0	
49	9	11	0	
50	0	12	0	
51	6	12	0	
52	7	12	0	
53	8	12	0	
54	9	12	0	

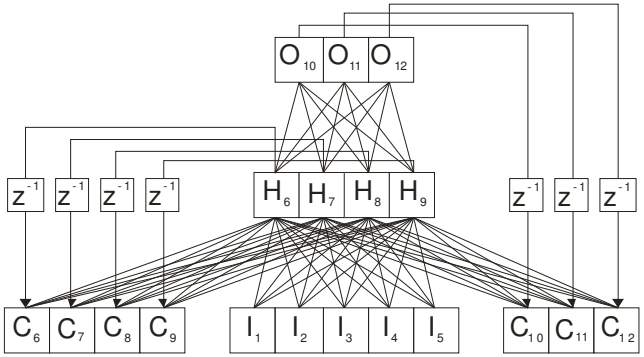
d) Units

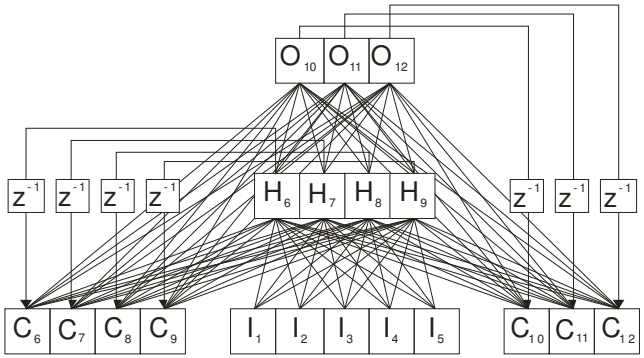
Index	First W.	Last W.	Type	Act. F.
UI	FW	LW	UT	AF
0			T	
1			I	
2			I	
3			I	
4			I	
5			I	
6	0	9	H	SGM
7	10	19	H	SGM
8	20	29	H	SGM
9	30	39	H	SGM
10	40	44	O	LIN
11	45	49	O	LIN
12	50	54	O	LIN

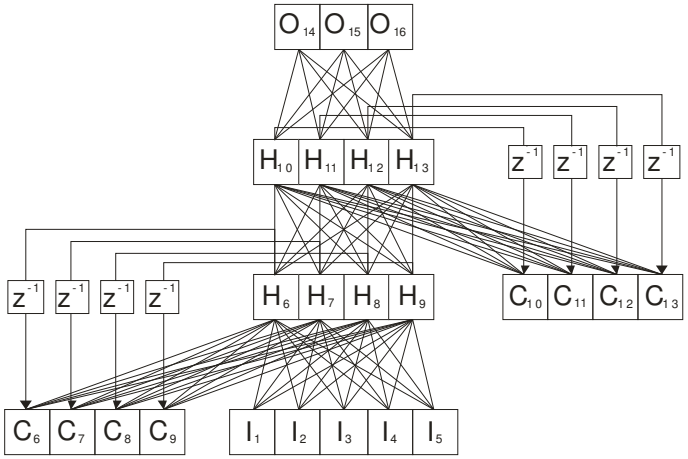
RNN Architectures









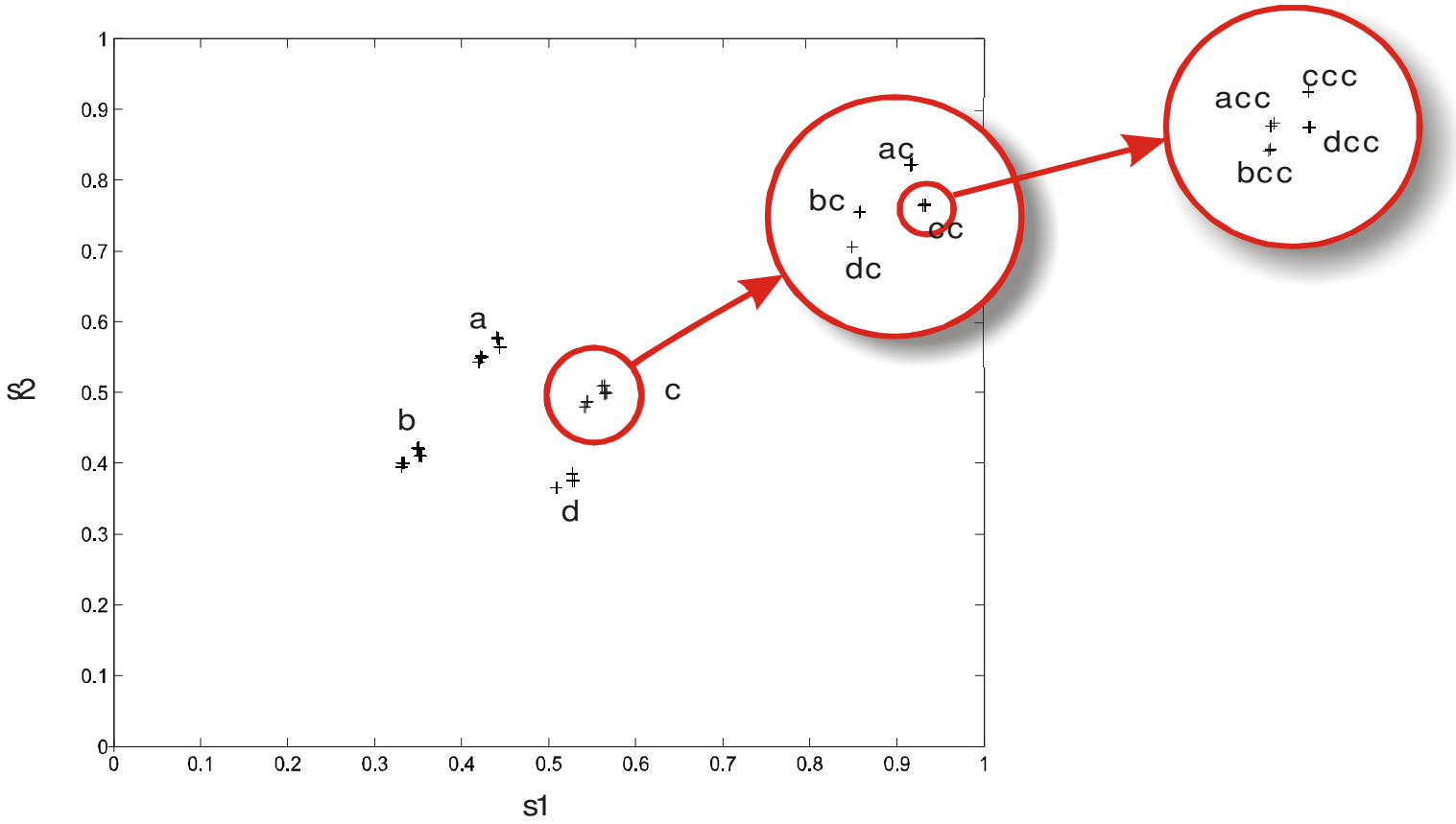


Adaptation algorithms

- Gradient based algorithms
  - Elmans's SRN – simple BP algorithm
  - BPTT – backpropagation through time
  - RTRL – real time recurrent learning
- KF – based approaches
  - KF, EKF, EKF for training (R)NNs
  - RTRL or BPTT for calculating derivatives
  - MS-EKF
  - Dual estimation Dual-EKF, decoupling DEKF, Sigma point EKF (SP-EKF – unscented, NPR)

Markovian architectural bias of RNN

- Clusters in the state space of randomly initialized network

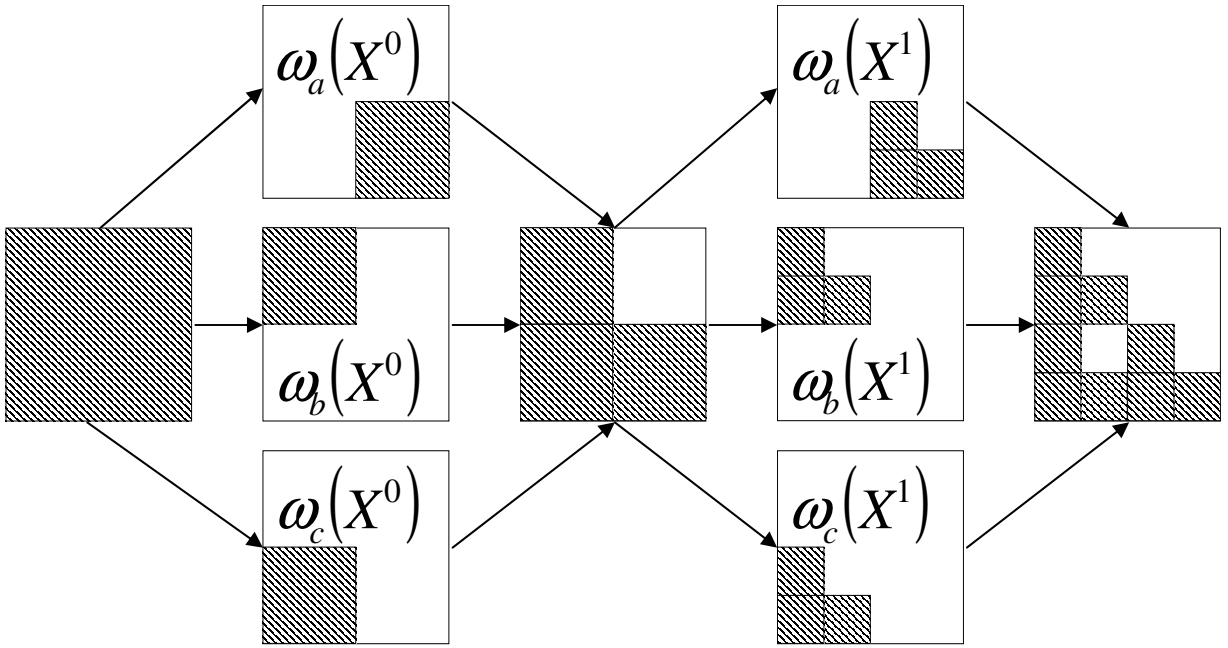


Iterated Function Systems (IFS)

- IFS - finite set of transformations
- Example of IFS: $\Omega = \{\omega_i \mid \omega_i : X \to X, i \le n\}$

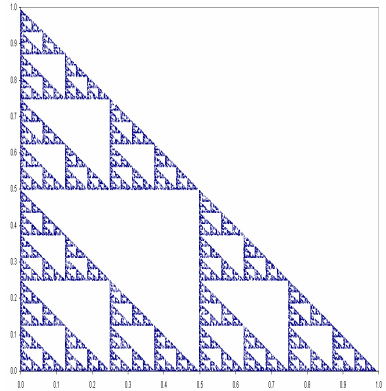
$$\omega_1(x, y) = (0.5x + 0.5, 0.5y)$$
$$\omega_2(x, y) = (0.5x, 0.5y + 0.5)$$
$$\omega_3(x, y) = (0.5x, 0.5y)$$

Sierpinski Triangle



$$X^\infty = (\omega_a \cup \omega_b \cup \omega_c)^\infty (X)$$

∞



$$X^0 = X \qquad X^1 = (\omega_a \cup \omega_b \cup \omega_c)(X^0) \qquad X^2 = (\omega_a \cup \omega_b \cup \omega_c)^2 (X^1)$$

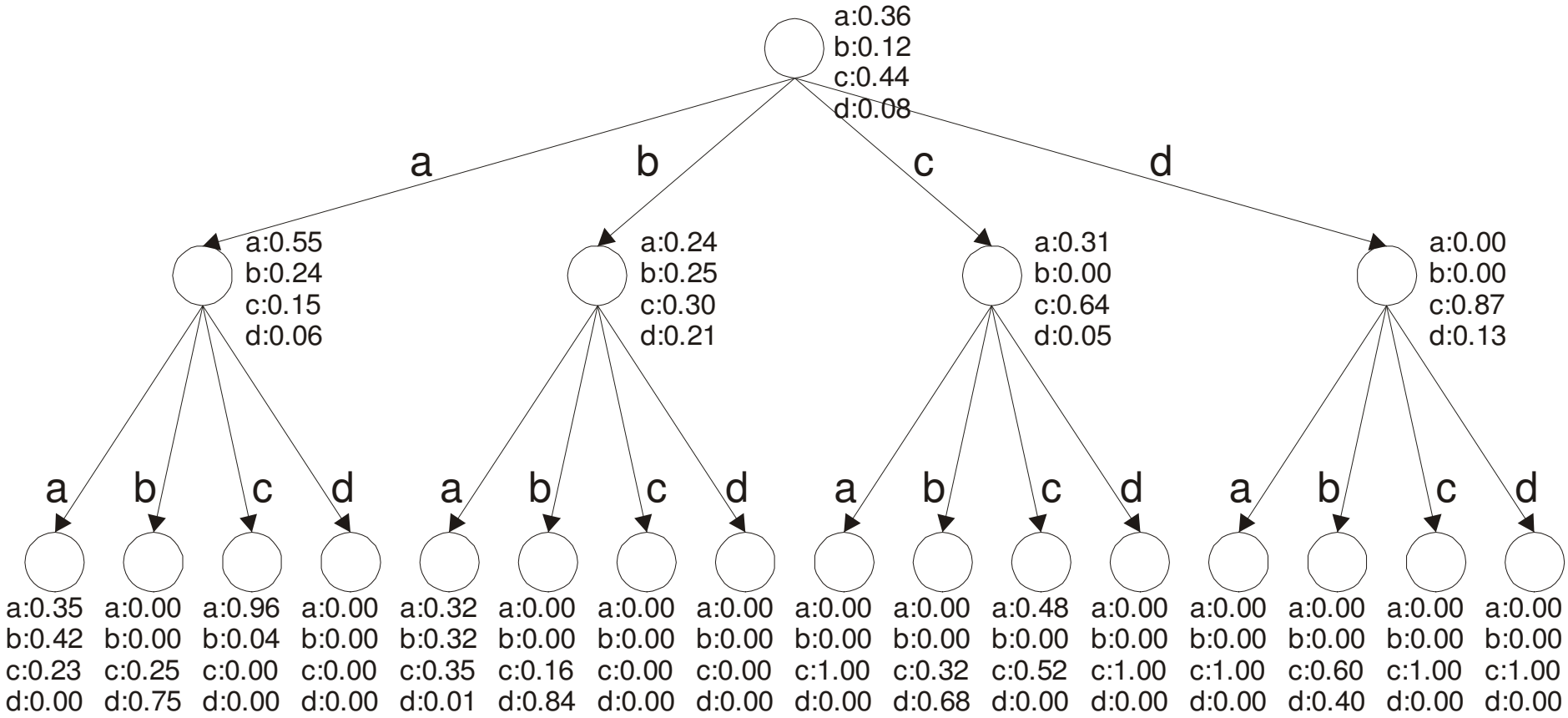
Chaos Game: $x_0 \in X$ $x_t = \omega_t(x_{t-1})$ $\omega_t = \begin{cases} \omega_a \\ \omega_b \\ \omega_c \end{cases}$ with equal probability

IFS Properties of RNNs

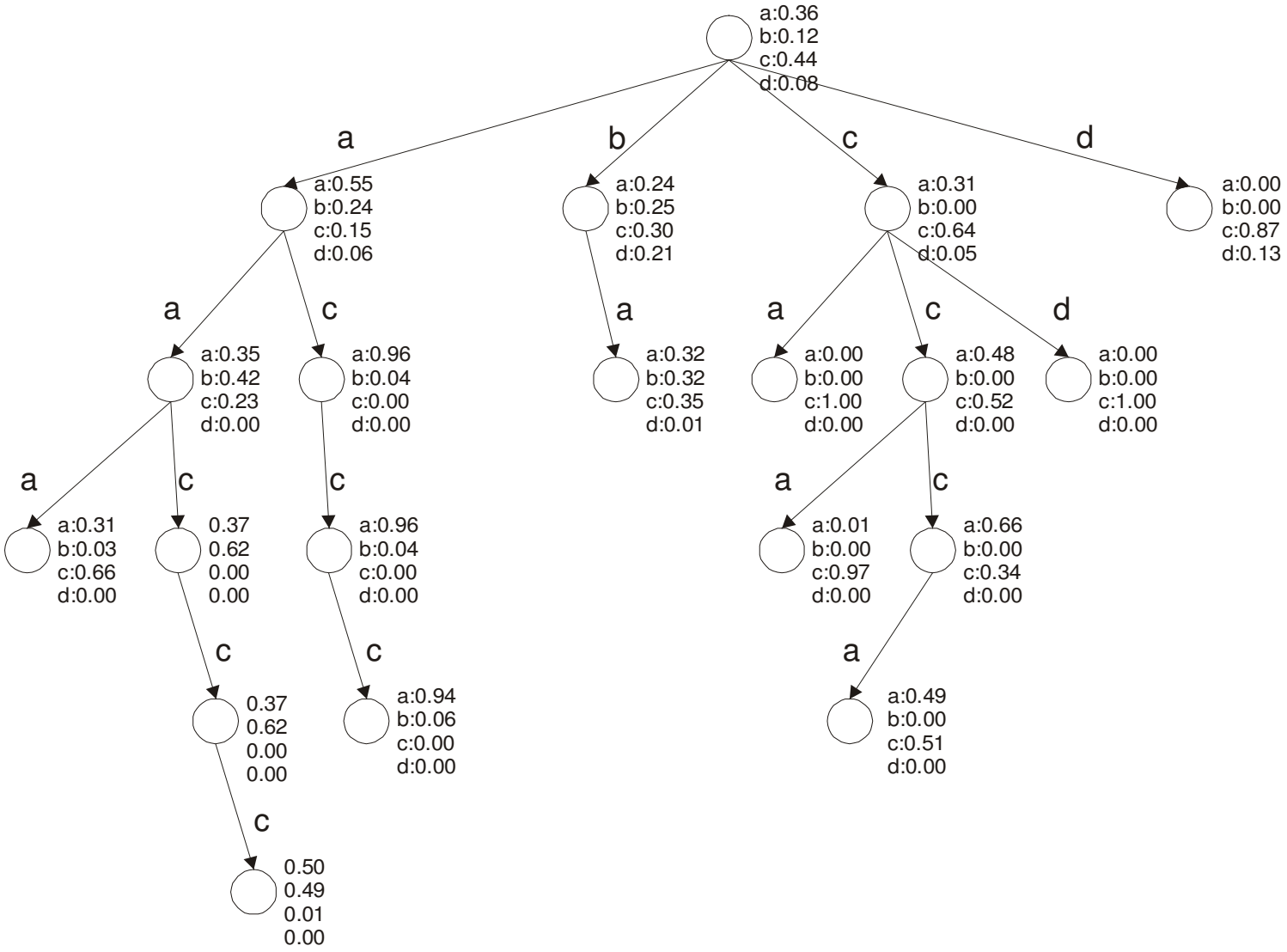
- State of RNN is determined by
  - the last input symbol
  - the second last input symbol
  - the third last input symbol
  - …
- The longer common suffix of two sequences, the smaller distance of corresponding points in the state space

Markov Models

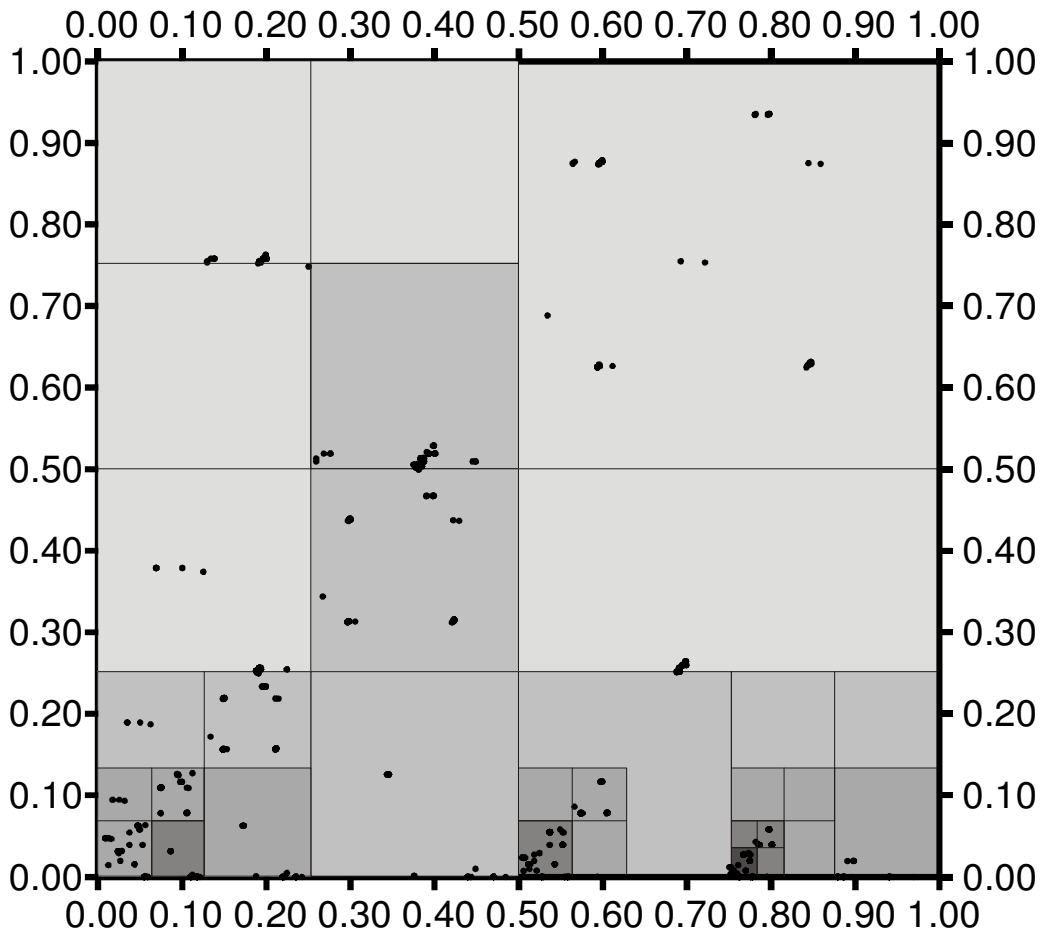
$$P\left(x_{t+1} \mid x_1..x_t\right) = P\left(x_{t+1} \mid x_{t-m+1}..x_t\right)$$



Variable Length Markov Models (Chain)



VLMM vs. IFS


0.00 0.10 0.20 0.30 0.40 0.50 0.60 0.70 0.80 0.90 1.00
1.00
0.90
0.80
0.70
0.60
0.50
0.40
0.30
0.20
0.10
0.00

- VLMM
  - PST definition and properties
  - PST creation algorithm

- FPM – fractal prediction machine
  - training algorithm
    - clusterisation – K-means
    - creation of prediction model

- ESN – Echo state networks

Experiments

- 4 Datasets: Laser, Feigenbaum, CERandRBR, DeepRec
- Architectures: Elman, W&Z, Jordan, Bengio, F&S, …
- Algorithms:

  - BPTT, EKF-BPTT, (MS-EKF, SP-EKF, DEKF)

  - NPM, FPM, ESN

  - MM, VLMM

Experiment results

- Elman – 16 units for most datasets
- Elman >= other architectures
- EKF > BPTT
- NPM, FPM = VLMM
- NPM, FPM

  for Laser, Feigenbaum = EKF, BPTT

  for CERandRBR, DeepRec <EKF, BPTT

Application

- Written English – probabil. model
  - VLMM – prediction NNL, correction (Viterbi)
  - Arch. bias appr.: FPM, NPM, ESN
  - Dataset: Bible
  - Task: text correction

- DNA processing