

# Интеллектуальные системы и технологии

Лекция 11. Часть 2.

Рекуррентные многослойные  
нейронные сети

# Рекуррентные сети

- Вид нейронных сетей, в которых имеется обратная связь. При этом под обратной связью подразумевается связь от логически более удалённого элемента к менее удалённому.
- Наличие обратных связей позволяет запоминать и воспроизводить целые последовательности реакций на один стимул.
- С точки зрения программирования в таких сетях появляется аналог циклического выполнения, а с точки зрения систем — такая сеть эквивалентна конечному автомату. Такие особенности потенциально предоставляют множество возможностей для моделирования биологических нейронных сетей.

# Простая рекуррентная сеть Джордана

- Сеть Джордана имеет обратные связи от выходного слоя к входному, и некоторые входные нейроны имеют обратные связи от своего выхода к своему входу.
- Может обучаться обратным распространением ошибки.
- Имеет некоторую форму кратковременной памяти.

# Рекуррентная сеть Джордана

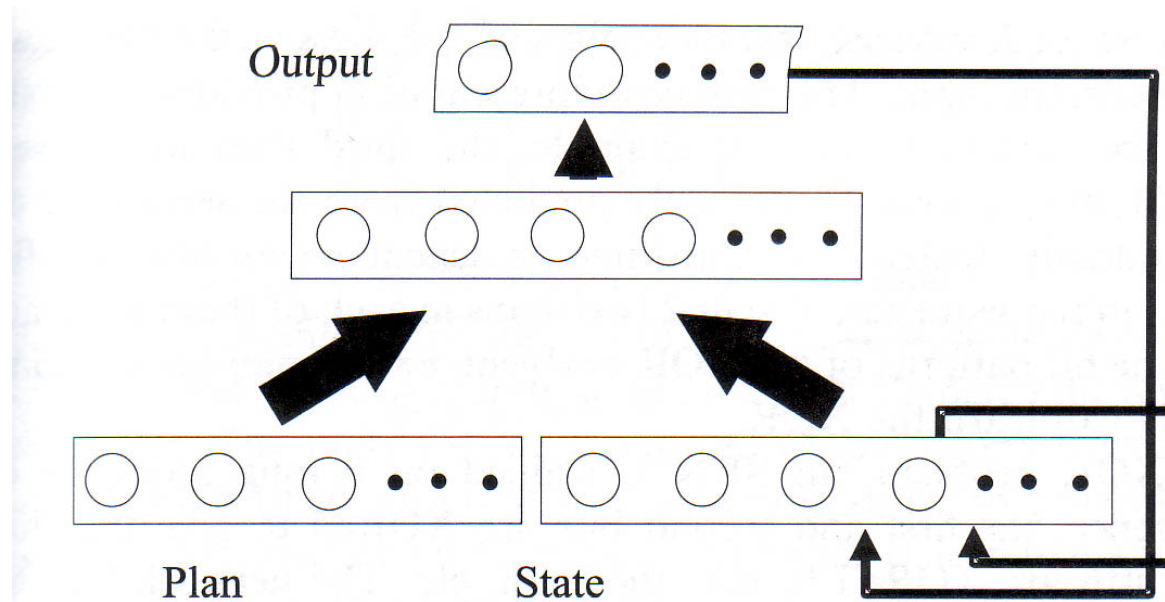
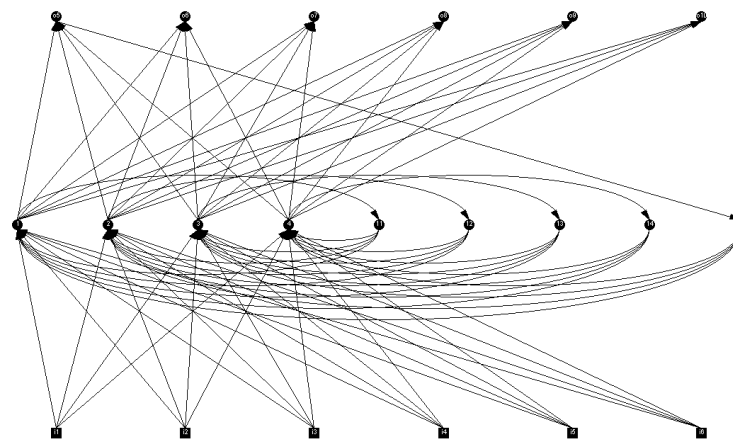
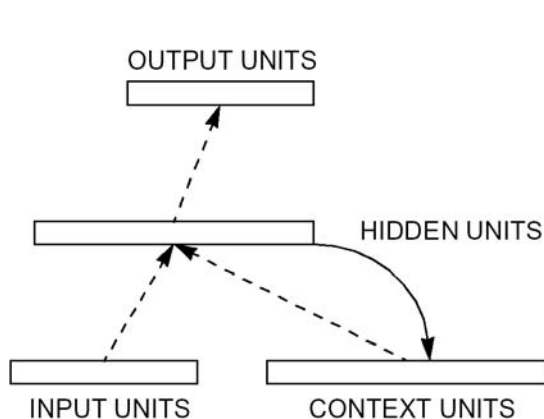


Figure 5.6 Jordan network.

# Сеть Элмана (Elman) (SRN)

- Сеть Элмана имеет является сетью прямого распространения с частичной рекуррентностью.



- В отличие от сетей прямого распространения, сеть Элмана имеет память о времени или «чувство времени».

# Кратковременная память в SRN

- Контекстный слой (context units) запоминает предыдущее внутреннее состояние сети.
- Таким образом, скрытый слой (hidden units) решает задачу отображения внешнего входного паттерна и предыдущего состояния на желательные выходные паттерны.
- SRN может предсказывать следующий образ в последовательности на основе текущего и предыдущего.

# Рекуррентная нейронная сеть с обратным распространением ошибки

- Обратное распространение ошибки не требует наличия только прямых связей.
- Нейрон может по выходу быть связан и сам с собой и с нижележащими нейронами.
- Обратные связи влияют на выходные нейроны через сеть в следующие итерации, во время обработки следующих входных векторов.
- Для каждой рекуррентной сети существует эквивалентная сеть прямого распространения с тем же поведением.

# Рекуррентная нейронная сеть (Связи выходов выходного слоя с входным)

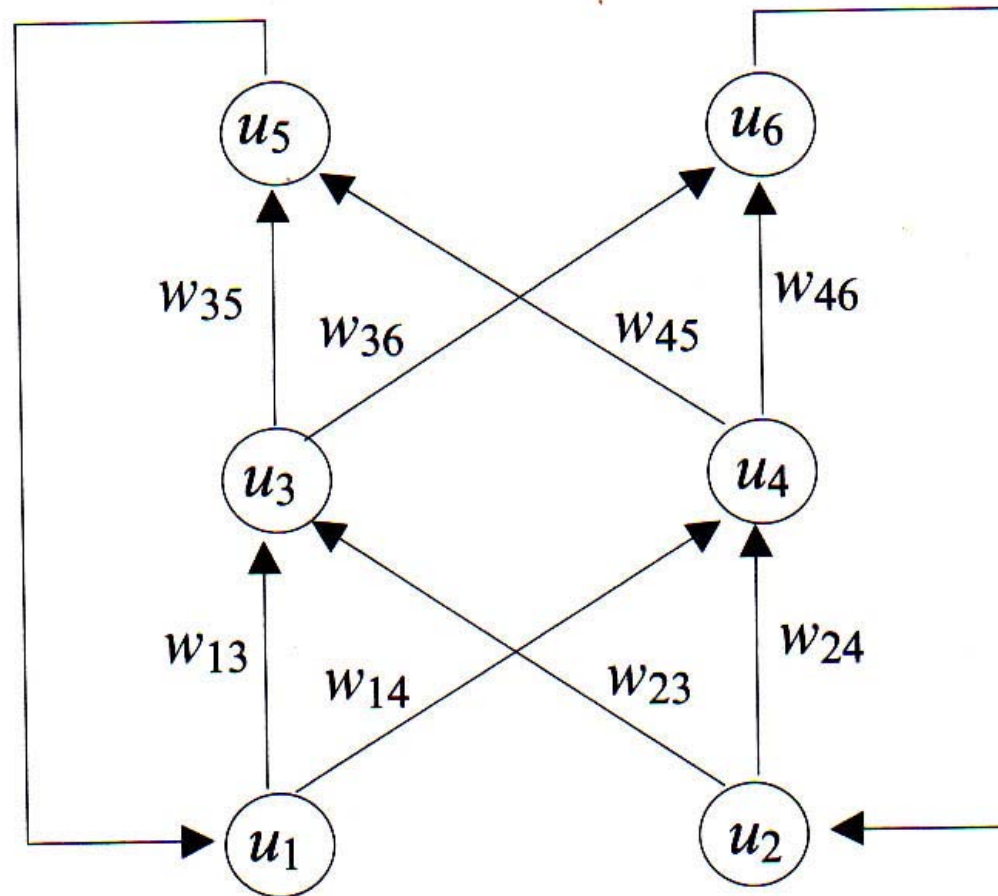
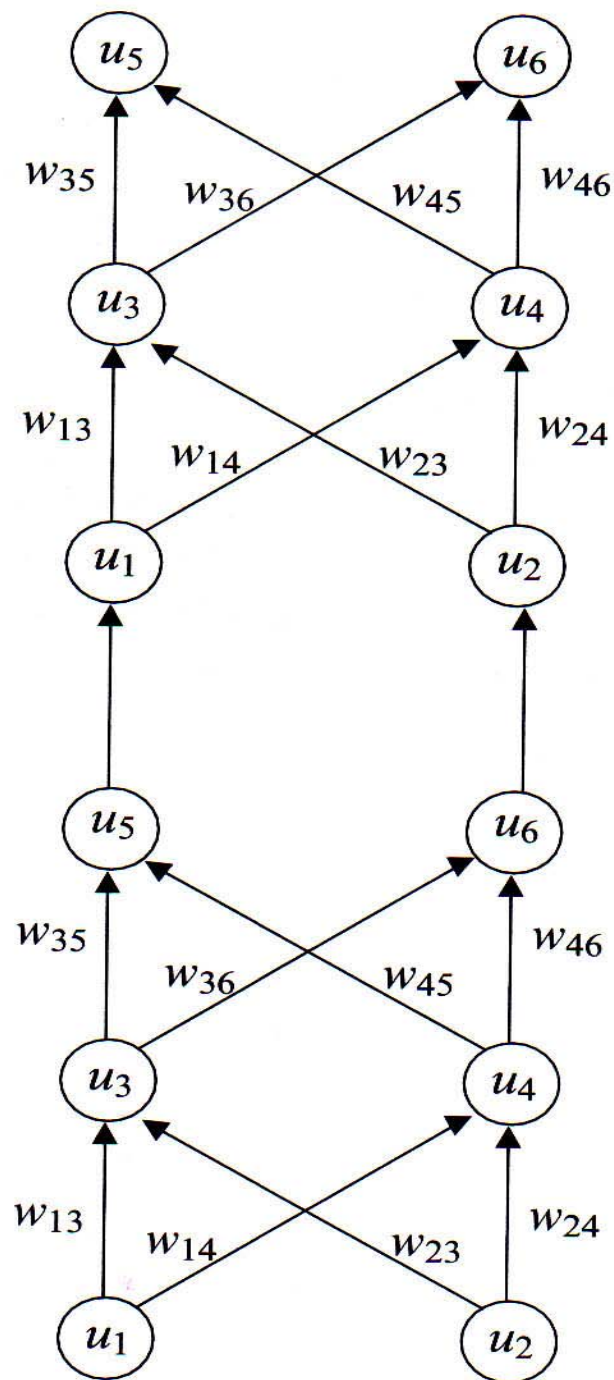


Figure 5.2 A recurrent backpropagation network.



Расширенная версия  
сети из предыдущего  
рисунка  
(эквивалентная сеть  
прямого  
распространения)



# Backprop Through Time

(Обратное распространение ошибки  
через время)

- During training a pattern is presented to the network and a feedforward pass is made.
- Each network copy corresponds to a time step.
- The weight changes calculated for each network copy are summed before individual weights are adapted.
- The set of weights for each copy(time step) always remain the same.

# Sequence Completion Example

- Rumelhart et. al. (1986) performed experiments with recurrent connections.
- Train a network to perform sequence completion.
- A sequence consisted of 6 characters: two letters from the set {A,B,C,D,E} and 4 being numbers from the integer set {1,2,3}.
- For example, A can denote “12” and C can denote “13”.
- The complete sequence prefixed by AC would be “AC1213”.
- The complete sequence prefixed by AA would be “AA1212”.

# Rumelhart's network

- The network has:
  - 5 input units, one for each letter.
  - 3 output units, one for each number.
  - 30 hidden units.
  - Each hidden unit is connected to itself and every other hidden unit.

# Training Steps

- T1: Input unit corresponding to the first letter is 1 and other inputs are off.
- T2: Input unit corresponding to the second letter is 1 and other inputs are off.
- T3: The target output is the third character in the sequence.
- T4: The target output is the fourth character in the sequence.
- T5: The target output is the fifth character in the sequence.
- T6: The target output is the sixth character in the sequence.



Output generation at different time steps.

# Rumelhart Network Results

- All hidden and output unit activations started at 0.2
- Network is trained on 20 sequences.
- Given first two characters, the rest of the characters were completed in five test cases.
- The completion could be achieved even if there were delays between the presentation of the first two letters.

Classic experiment on language acquisition and processing  
(Elman, 1990)

- **Task**

- Elman net to predict successive words in sentences.

- **Data**

- Suite of sentences, e.g.
  - “The boy catches the ball.”
  - “The girl eats an apple.”
- Words are input one at a time

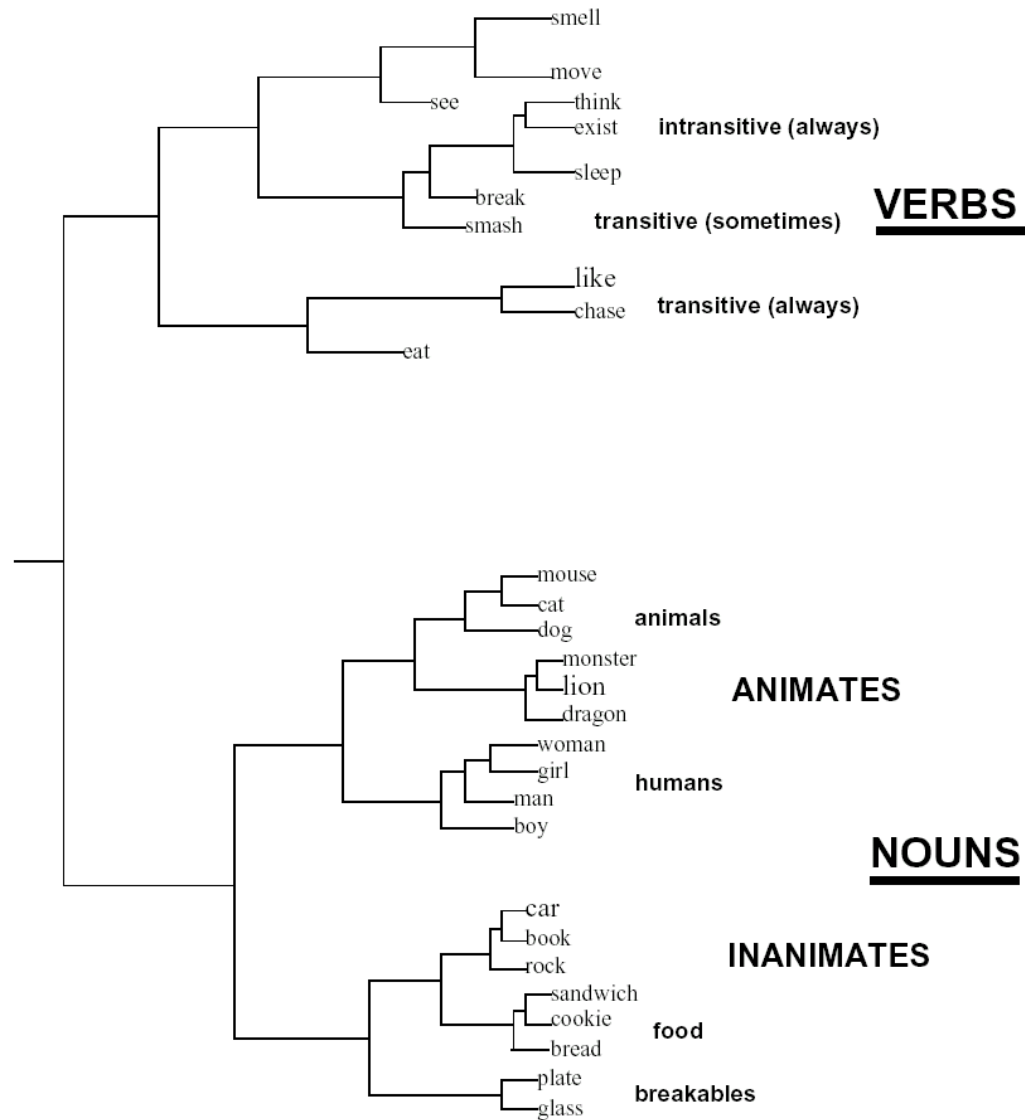
- **Representation**

- Binary representation for each word, e.g.
  - 0-1-0-0-0 for “girl”

- **Training method**

- Backpropagation

- Internal representation of words





# Grammatical inference by RNN (S.Lawrence, 1996)

*I am happy John to be here	I believe John to be here
I am happy for John to be here	*I believe for John to be here
I am happy to be here	*I believe to be here

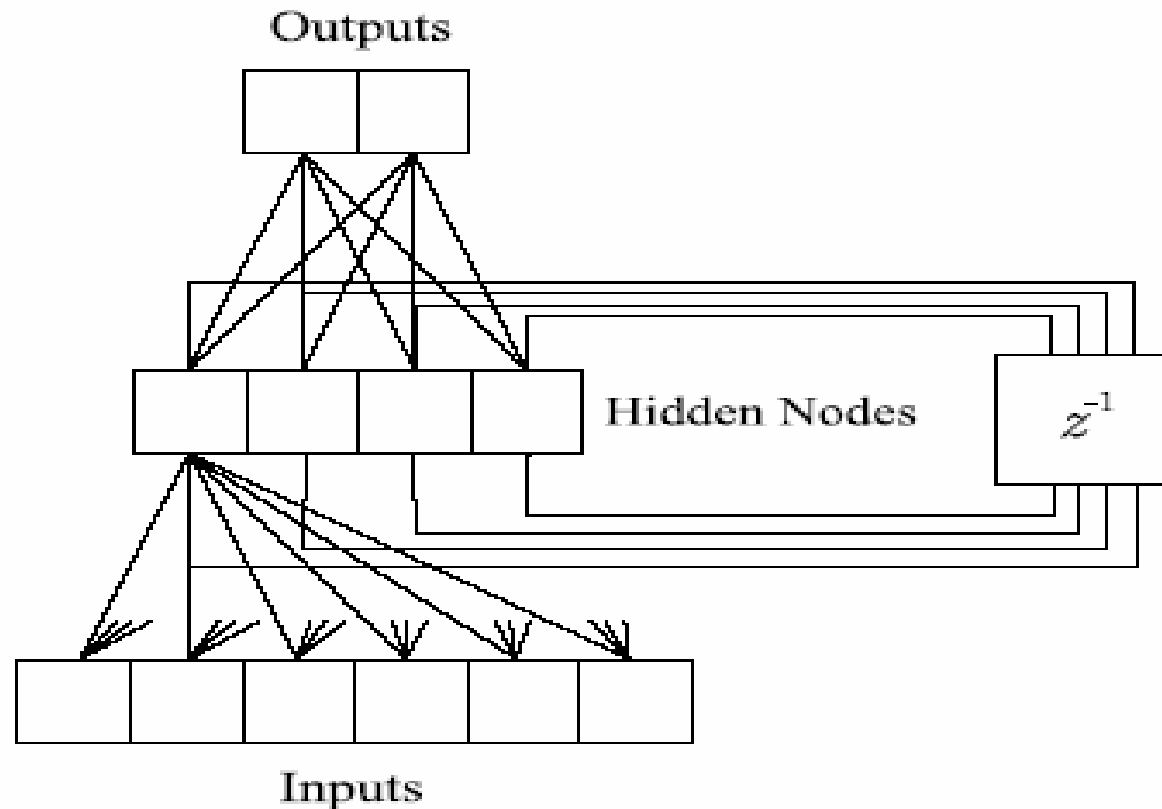
Asterisk marks the ungrammaticality.

Task is grammaticality judgment of sentence.

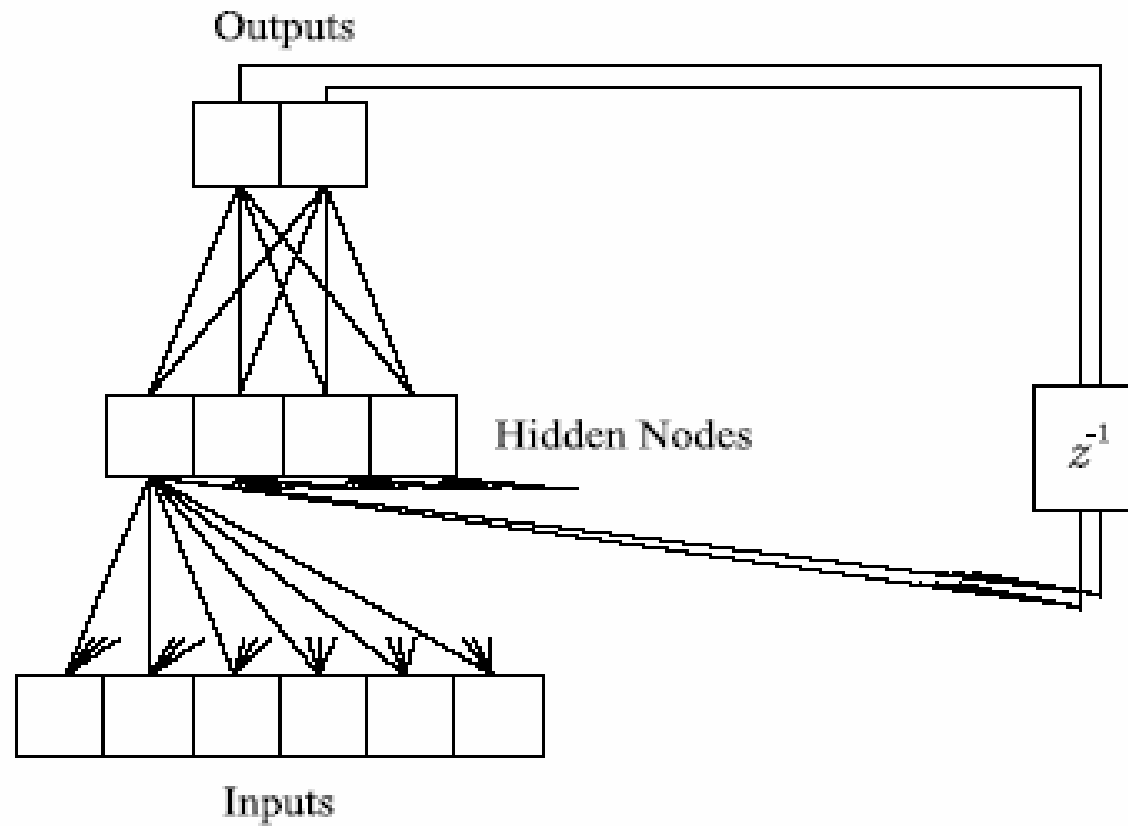
Different following RNNs was compared:

1. *Frasconi-Gori-Soda locally recurrent networks [9]*. A multi-layer perception augmented with local feedback around each hidden node. We have used the local-output version where the output of a node,  $y(t) = f(wy(t-1) + \sum_{i=0}^n w_i x_i)$ .
2. *Narendra and Parthasarathy [19]*. A recurrent network with feedback connections from each output node to all hidden nodes.
3. *Elman [8]*. A recurrent network with feedback from each hidden node to all hidden nodes.
4. *Williams and Zipser [29]*. A recurrent network where all nodes are connected to all other nodes.

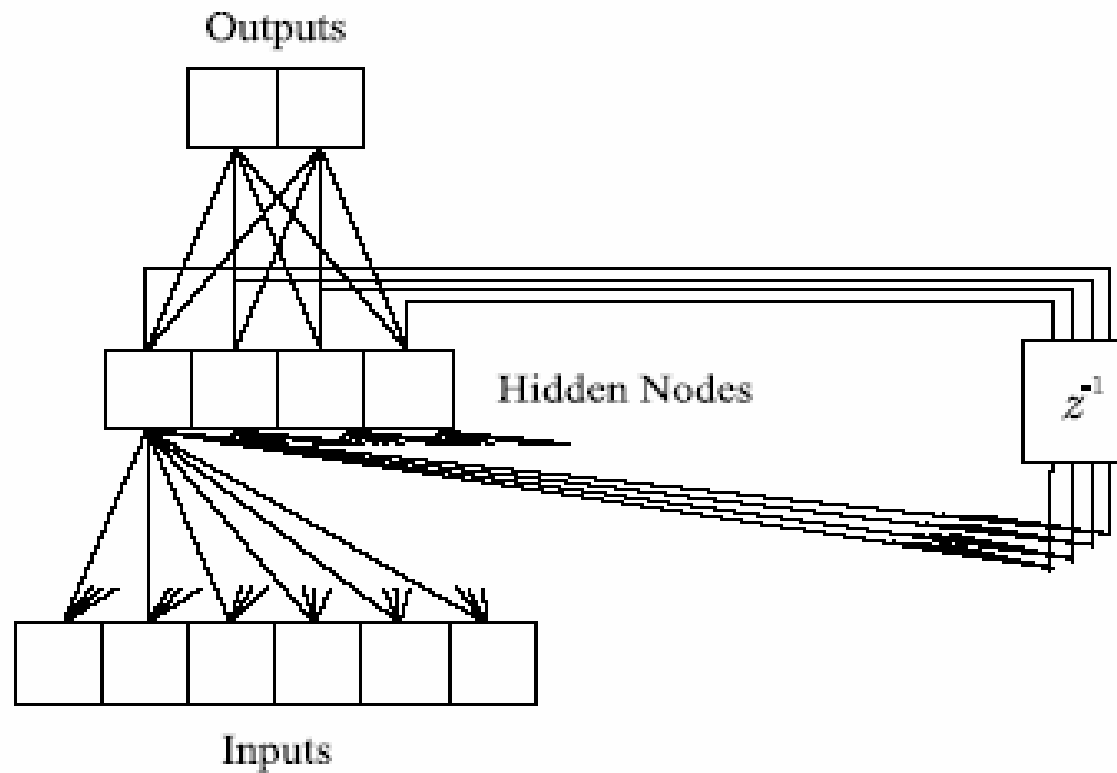
A Frasconi-Gori-Soda locally recurrent network. Not all connections are shown fully



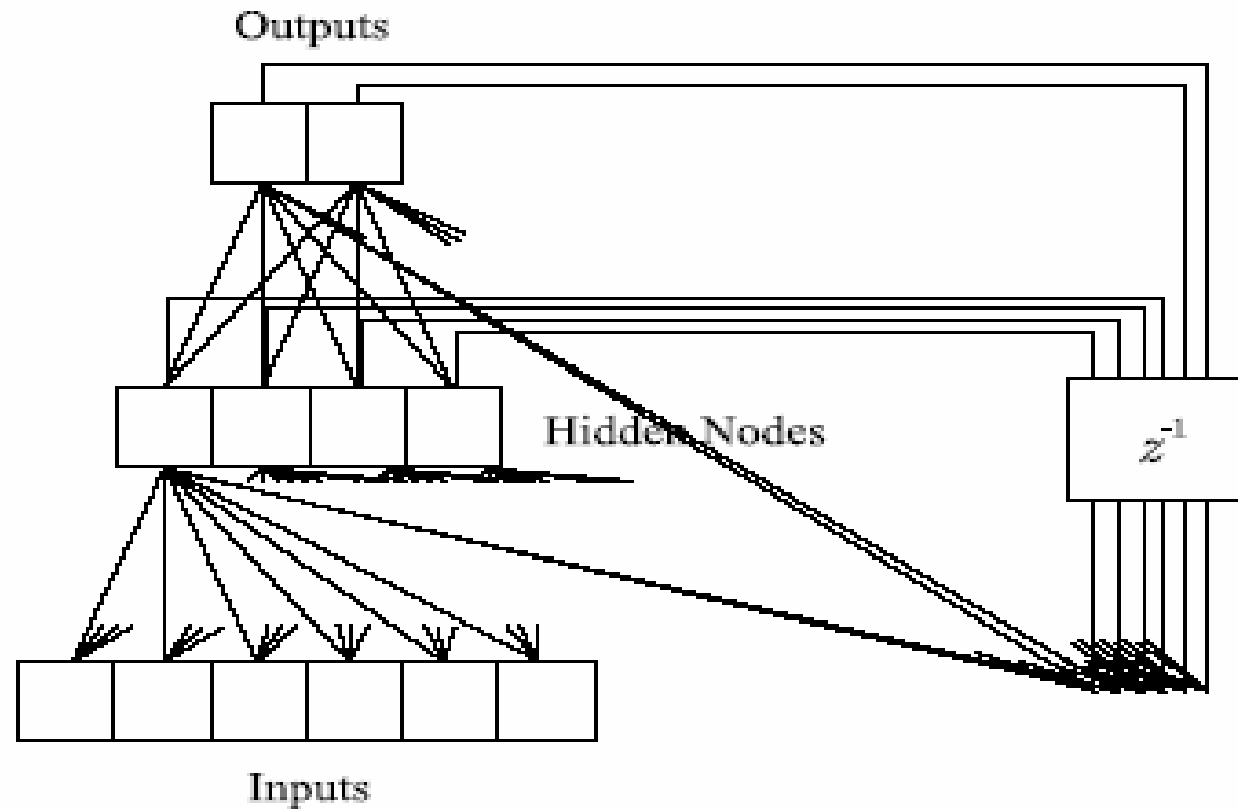
A Narendra & Parthasarathy recurrent network.  
Not all connections are shown fully



An Elman recurrent network. Not all connections are shown fully



A Williams & Zipser fully recurrent network.  
Not all connections are shown fully



# Parts of speech used in experiments

Category	Examples
Nouns (N)	<i>John, book and destruction</i>
Verbs (V)	<i>hit, be and sleep</i>
Adjectives (A)	<i>eager, old and happy</i>
Prepositions (P)	<i>without and from</i>
Complementizer (C)	<i>that or for as in I thought that ... or I am eager for ...</i>
Determiner (D)	<i>the or each as in the man or each man</i>
Adverb (Adv)	<i>sincerely or why as in I sincerely believe ... or Why did John want ...</i>
Marker (Mrkr)	<i>possessive 's, of, or to as in John's mother, the destruction of ... , or I want to help ...</i>

# A simple grammar

S → NP VP<sup>''</sup>

NP → PropN | N | N RC

VP → V (NP)

RC → who NP VP | who VP (NP)

N → boy | girl | cat...

PropN → John | Mary

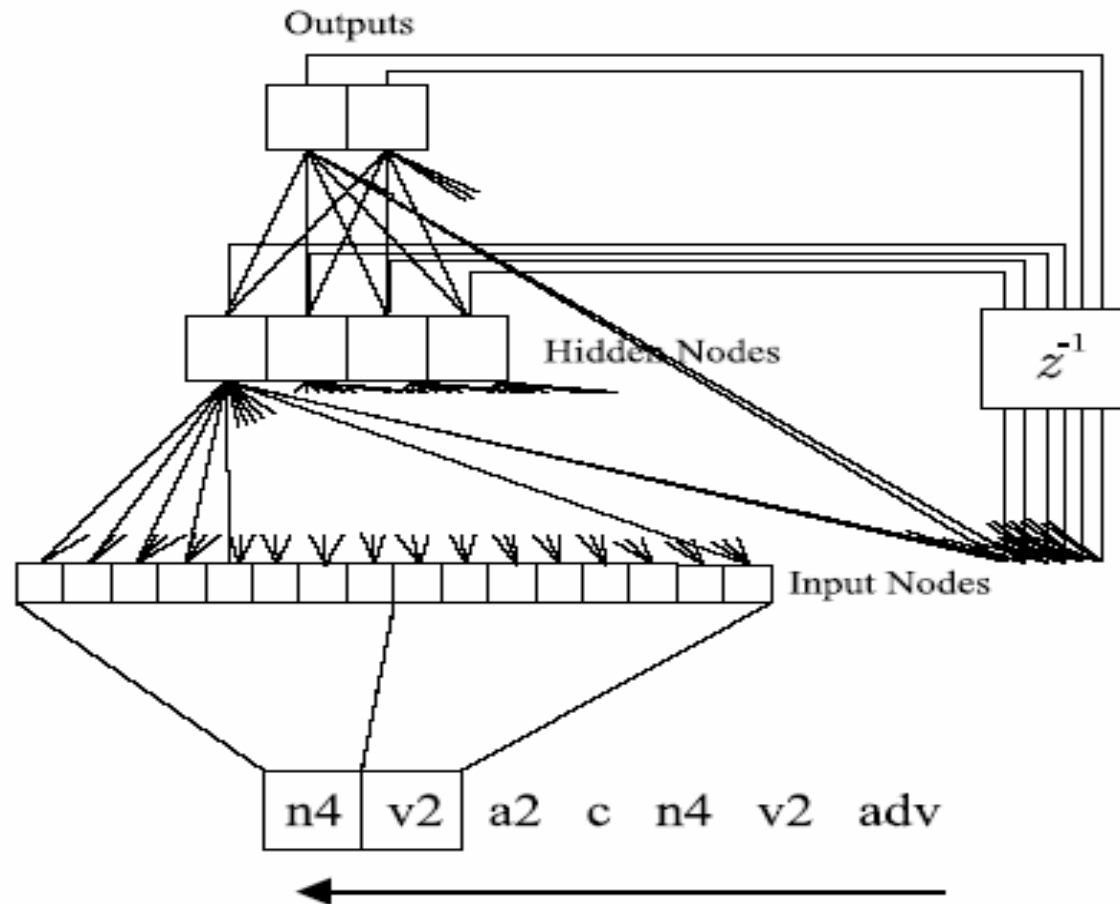
V → chase | feed | see...

# Examples of part-of-speech tagging

Sentence	Encoding	Grammatical Status
I am eager for John to be here	n4 v2 a2 c n4 v2 adv	1
	n4 v2 a2 c n4 p1 v2 adv	1
I am eager John to be here	n4 v2 a2 n4 v2 adv	0
	n4 v2 a2 n4 p1 v2 adv	0
I am eager to be here	n4 v2 a2 v2 adv	1
	n4 v2 a2 p1 v2 adv	1



Depiction of how the neural network inputs come from an input window on the sentence. The window moves from the beginning to the end of the sentence



# Results of comparison

TRAIN	Classification	Std. dev.	Confidence
Elman	<b>99.6%</b>	0.84	78.8%
FGS	67.1%	1.22	17.6%
N&P	75.2%	1.41	32.2%
W&Z	<b>91.7%</b>	2.26	63.2%

ENGLISH TEST	Classification	Std. dev.	Confidence
Elman	<b>74.2%</b>	3.82	75.4%
FGS	59.0%	1.52	18.8%
N&P	60.6%	0.97	26.9%
W&Z	<b>71.3%</b>	0.75	65.1%

Std. dev. – the standart deviation value.

The confidence is the average confidence of the networks.

# Results of comparison with other methods

Table 3. Percentage correct classification for the training data.

TRAIN	large window	small window
MLP	100	55
FGS	100	56
BT-FIR	100	56
Elman	100	<b>100</b>
W&Z	100	<b>92</b>

Table 4. Percentage correct classification for the test data.

	TEST	large window	small window
Nearest-neighbors	Edit-distance	55	N/A
	Euclidean	65	55
	Decision trees	60	N/A
	MLP	63	54
	FGS	65	59
	BT-FIR	64	54
	Elman	65	<b>74</b>
	W&Z	59	<b>71</b>

# Conclusions about applications of RNN for learning of NL

- Nearest-neighbors, decision trees, feedforward networks do not learn parsimonious representations of the grammar – they work by finding statistically close matches in the training data. They are expected to require a much larger amount of data for similar performance
- RNN may be learned an appropriate grammar for discriminating between sharply grammatical/ungrammatical pairs. 100% correct classification of the training data is not possible using only a small temporal input window without forming internal states
- Generalization is limited by the amount of data available

# Long Short-Term Memory (LSTM)

Для решения следующих задач:

- Распознавание временных паттернов в зашумленных последовательностях
- Распознавание временного порядка событий в зашумленной последовательности
- Извлечение информации из расстояния между событиями
- Стабильная генерация ритмов и периодических траекторий