

# Моделирование поведения СЛОЖНЫХ ДИНАМИЧЕСКИХ СИСТЕМ

Докладчик:

Юданов А.А.

Научный руководитель:

Бордаченкова Е.А

# Постановка задачи

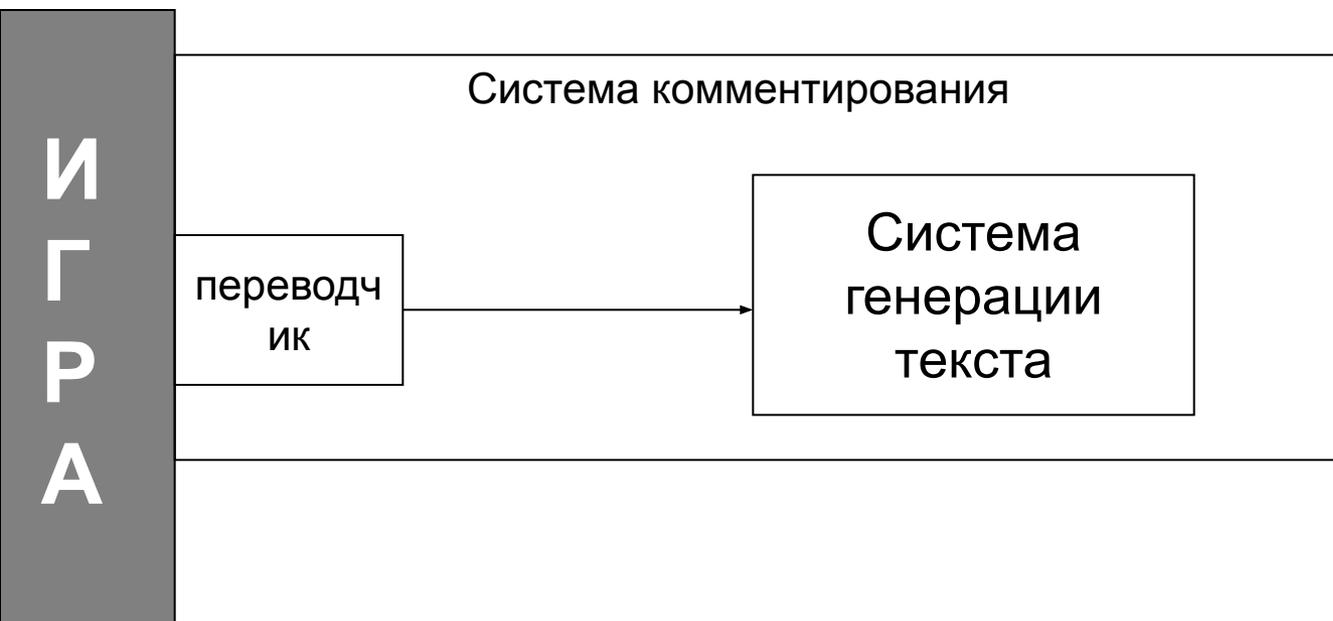
Цель – создать систему генерации связного текста на естественном языке по динамически поступающей информации из некоторой системы.

То есть, универсальный комментатор для определенного класса компьютерных игр:

- игроки управляют агентами в игровом мире.
- игровой мир – разнородный объем, содержащий агентов и другие предметы.

# Структура системы

- Игра генерирует некоторые события
- Их надо оценить и прокомментировать



# Структура системы

- Игра генерирует некоторые события
- Их надо оценить и прокомментировать

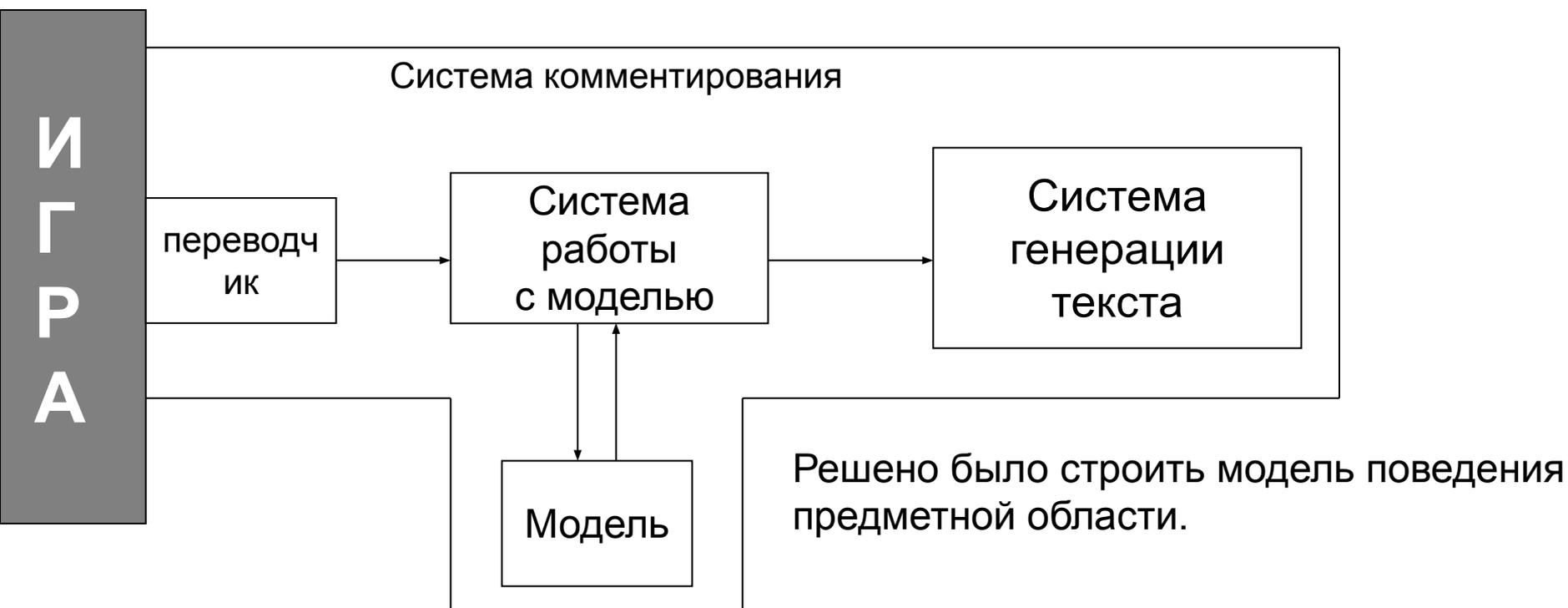
И  
Г  
Р  
А



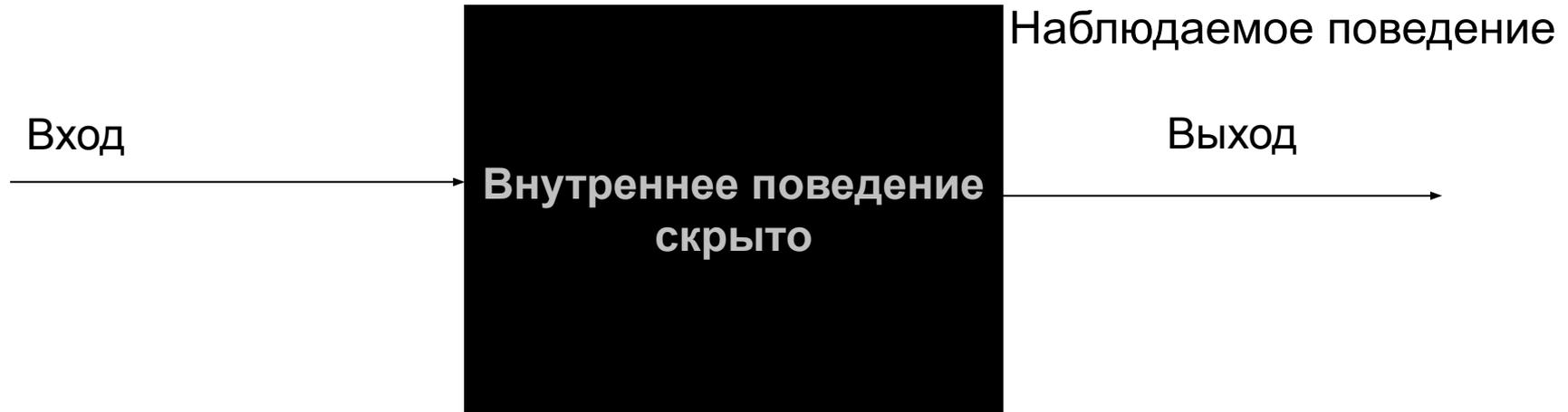
Проблема: широкий класс предметных областей вынуждает использовать адаптирующийся к конкретной задаче алгоритм.

# Структура

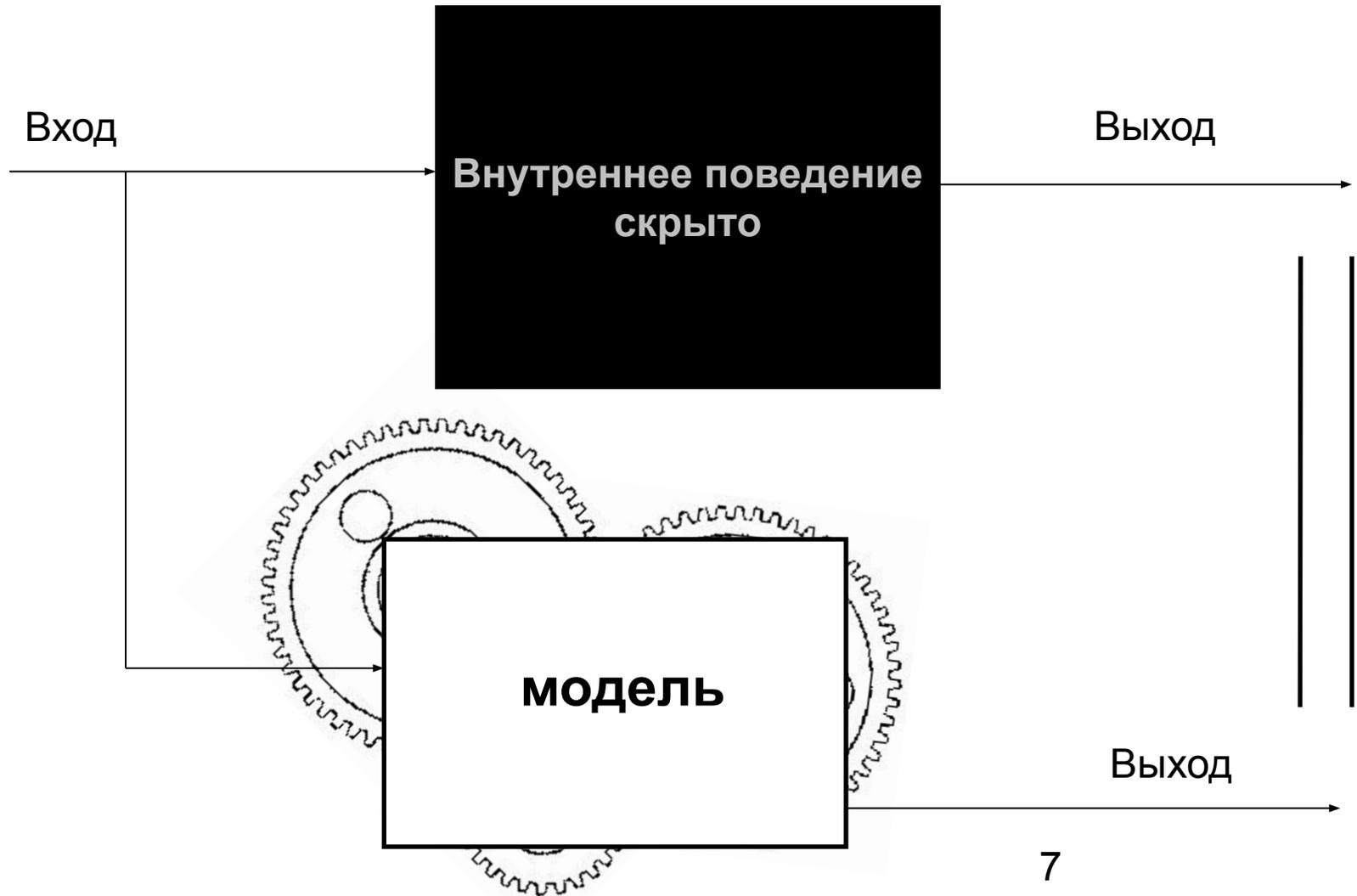
- Игра генерирует некоторые события
- Их надо оценить и прокомментировать



# Что такое модель поведения?



# Что такое модель поведения?



# Что такое модель поведения?

## Как измеряют качество модели?

Точно смоделировать систему невозможно.

### Оценки качества модели

- Ошибка обучения
  - Суммированная по обучающей выборке норма невязки модели
- Ошибка обобщения
  - Норма невязки модели на каких-либо новых данных

Основной целью при обучении является уменьшение ошибки обобщения.

# Зачем нужна модель поведения?

- Адекватно оценивать настоящее

выход моделируемой системы

=

выход модели

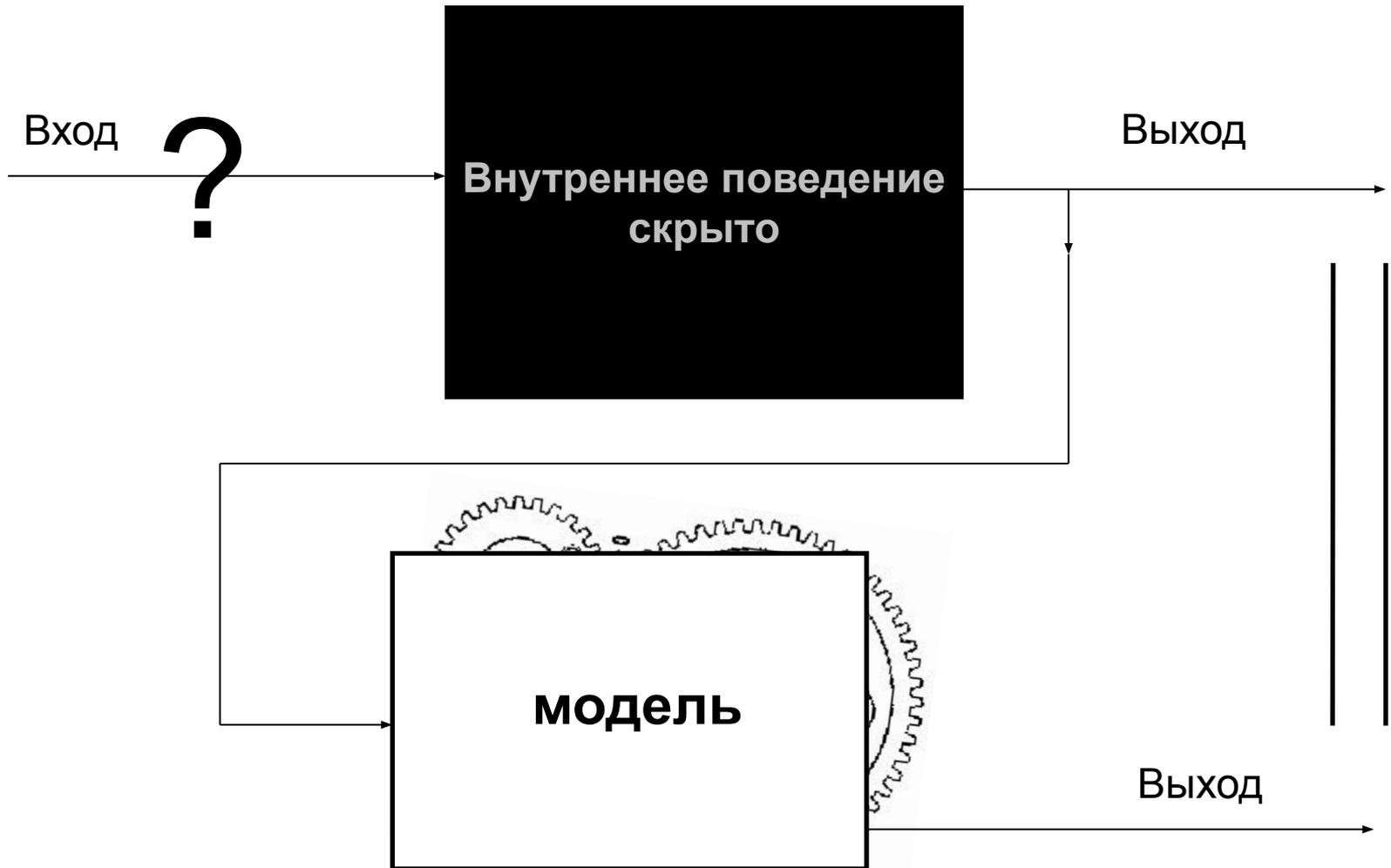
- Предсказывать будущее

предсказывать следующий выход системы на основе входа и наблюдаемого поведения системы.

# Особенности нашей задачи

- Входные данные системы не должны участвовать как в алгоритмах обучения, так и работы.
- Обязательна возможность динамического обучения без учителя.
- Желательно, получать в предсказательном алгоритме распределение вероятностей.
- Желательно иметь возможность дообучать модель по ходу работы.
- Чем проще и быстрее работа с моделью – тем лучше.

# Особенности нашей задачи



# Обзор некоторых моделей

- Нейронные сети
  - Самоорганизующиеся карты Кохонена.  
Synapse [<http://www.peltarion.com/products/synapse/>]  
WEBSOM [<http://websom.hut.fi/websom/>]
- Статистические модели
  - Скрытые Марковские модели.  
TreeAge Pro Healthcare [<http://www.treeage.com>]  
HTS [<http://hts.sp.nitech.ac.jp/>]

# Нейронные сети

- Модель представляет из себя сеть, сплетенную из одинаковых по функциям элементам с помощью синаптических связей.
- Каждый элемент (нейрон) выполняет одну функцию – обрабатывая определенным образом входные сигналы  $x_1, \dots, x_n$  генерирует выходной  $y$ :

$$y = F(x_1, \dots, x_n)$$

# Карты Кохонена

## Общие сведения

- Разновидность нейронных сетей.
- Топология – регулярная решетка.
- Каждый нейрон содержит вектор весов и свои координаты на решетке.
- Обучение итерационное, до достижения определенной погрешности

# Карты Кохонена

Отображение набора входных векторов высокой размерности в карту кластеров меньшей размерности, причем, таким образом что близким кластерам на карте отвечают близкие друг к другу входные вектора в исходном пространстве.

## Задачи

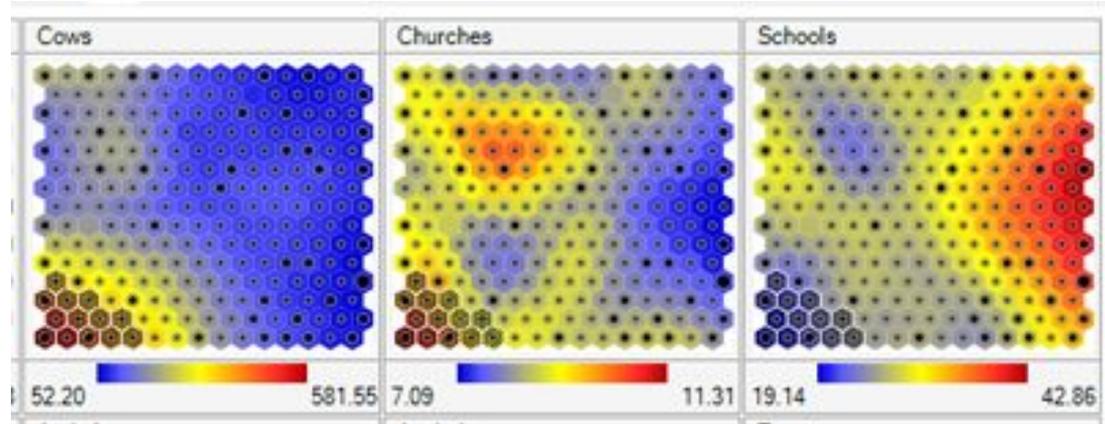
- Наглядное упорядочивание многопараметрической информации.
- Группировка близких входных сигналов для другой нейросети.

# Карты Кохонена

## Задачи

Наглядное упорядочивание многопараметрической информации.

- Выявление различий в режимах поведения системы.

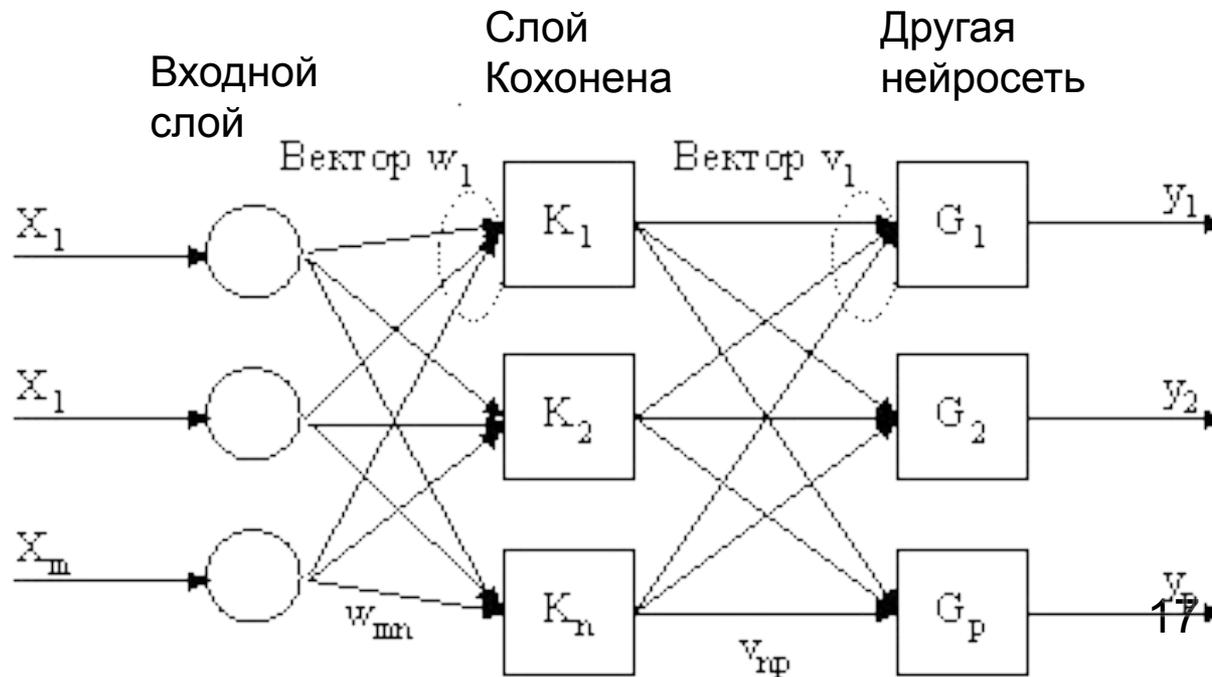


- Нахождение взаимосвязей в многопараметрических данных.

# Карты Кохонена

## Задачи

Карта Кохонена группирует близкие входные сигналы  $X$ , а требуемая функция  $Y=G(X)$  строится на основе применения обычной нейросети.

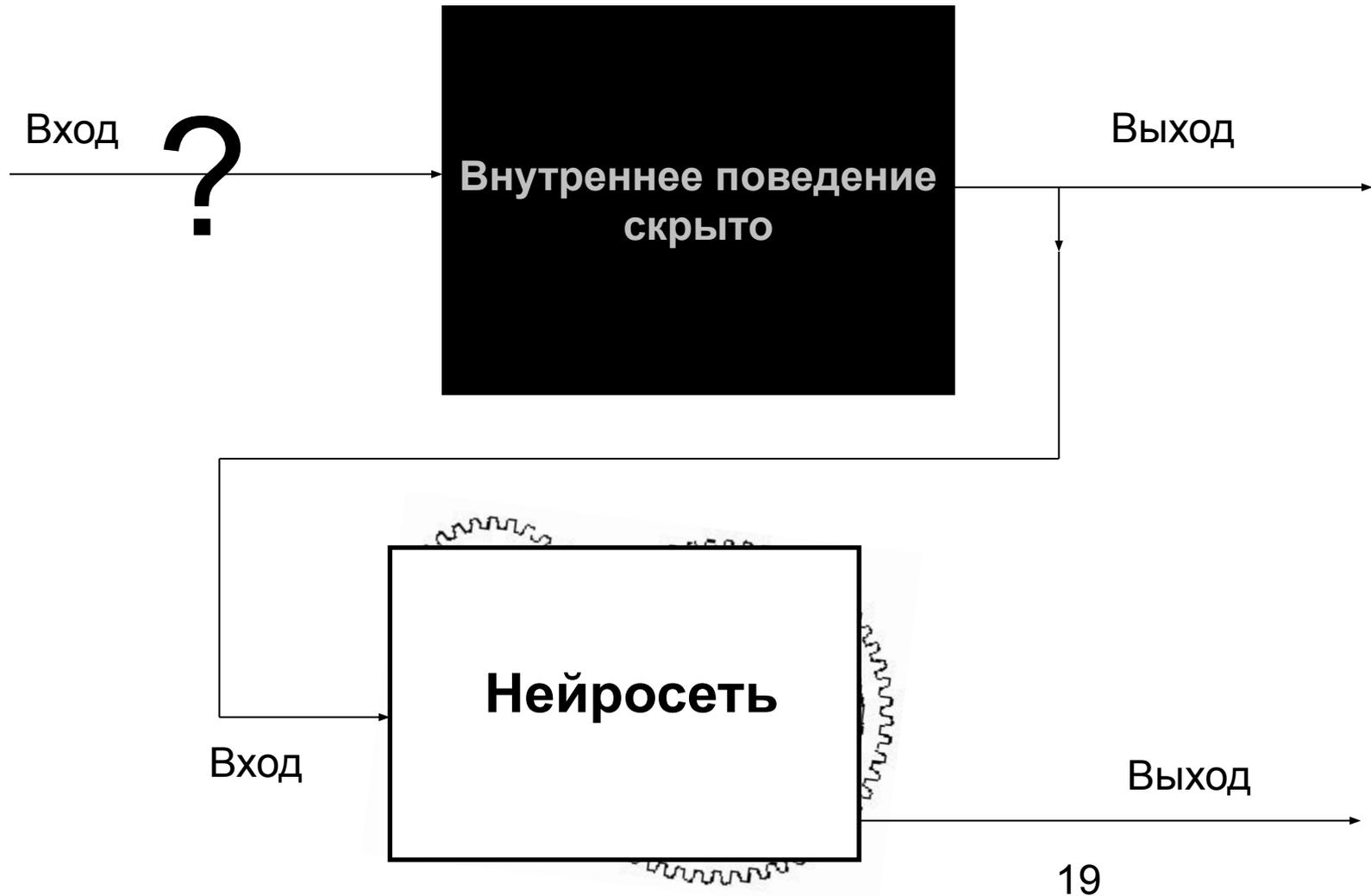


# Нейронные сети

## Приложение к нашей задаче

- Моделируемая система – игра.
- Нейронные сети должны получать вход системы, который недоступен. Значит, простой подстановки НС в задачу не получится
- Можно рассматривать выход системы как вход для НС, т.е. НС читает предыдущий выход и пытается предсказать следующий.

# Нейронные сети



# Нейронные сети

## Плюсы и минусы

- + Даже малая обучающая выборка не дает большой ошибки обобщения
- Неэффективная реализация на ПК.
- Сложный процесс проектирования топологии

# Скрытые Марковские модели

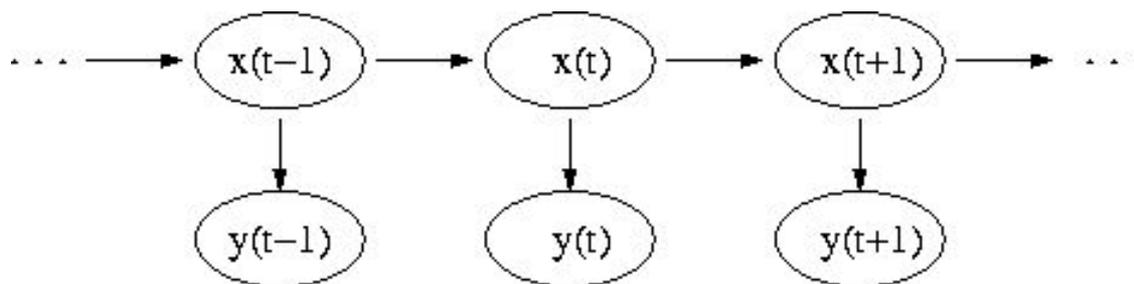
## Общие сведения

- Имеет набор скрытых внутренних состояний (скрытая переменная)  $x(t)$
- Имеет набор наблюдаемых состояний (наблюдаемая переменная)  $y(t)$
- $x(t)$  зависит только от  $x(t-1)$
- $y(t)$  зависит только от  $x(t)$

# Скрытые Марковские модели

## Общие сведения

Время – дискретное.



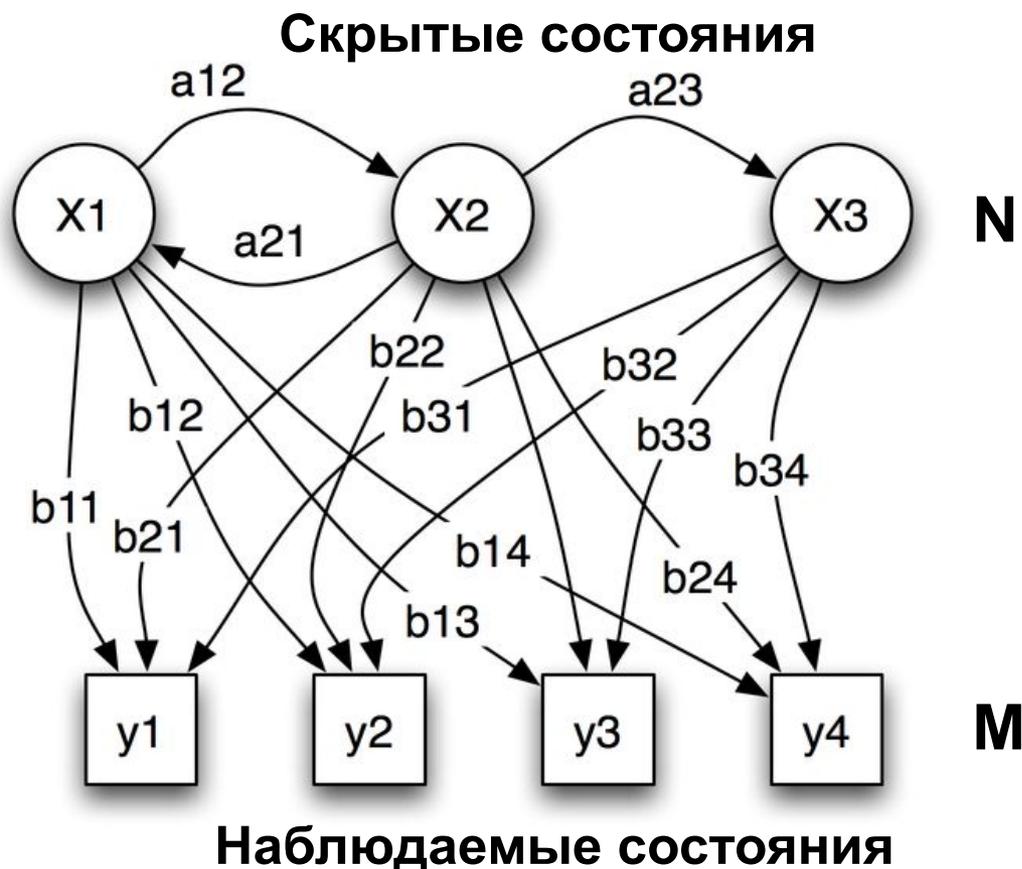
На каждом шаге система может перейти в новое скрытое состояние и как-то поменять значение наблюдаемой переменной (сменить наблюдаемое состояние)

# Скрытые Марковские модели

## Общие сведения

-Имеется матрица  $A[N \times N]$  вероятностей переходов между скрытыми состояниями.

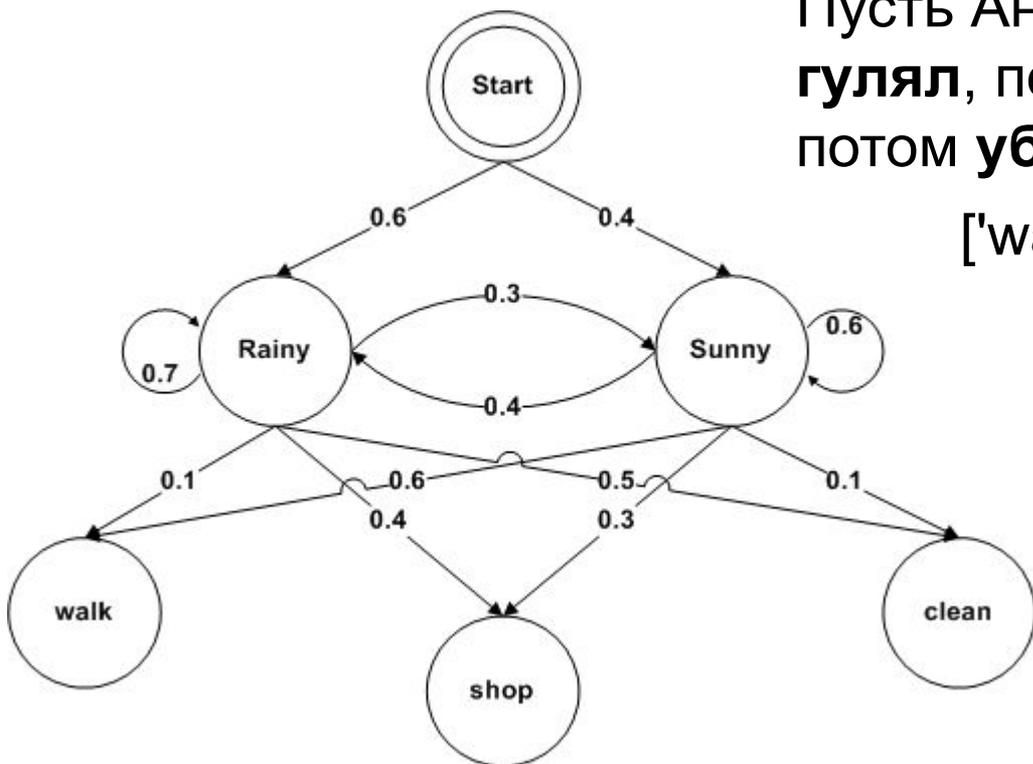
-Имеется матрица  $B[N \times M]$  вероятностей возникновения каждого из наблюдаемых событий на каждом из скрытых состояний.



# Скрытые Марковские модели пример

Пусть Аня знает, что Ваня сначала гулял, потом ходил за покупками, потом убирался.

['walk', 'shop', 'clean', 'walk', 'walk']



Наиболее вероятная скрытая последовательность погоды  
['Sunny', 'Rainy', 'Rainy', 'Sunny', 'Sunny']  
Вероятность наблюдения при этом - 4%

# Скрытые Марковские модели

## Задачи

1. Для заданной модели вычислить вероятность появления некоторой наблюдаемой последовательности.  
[алгоритм вперёд-назад]
2. Для модели и наблюдаемой последовательности определить наиболее вероятную последовательность скрытых состояний.  
[алгоритм Витерби]
3. Для набора наблюдаемых последовательностей требуется скорректировать матрицы переходов некоторой модели, чтобы эти последовательности были наиболее вероятны в ней.  
[алгоритм Баума-Уэлша]

# Скрытые Марковские модели

## Приложение к нашей задаче



# Скрытые Марковские модели

## Плюсы и минусы

- + Имитирует внутреннюю жизнь моделируемой системы.
- + Простые итерационные алгоритмы обучения.
- + Простота и быстрота работы.
- + Соответствие большинству наших требований
- + Хорошие результаты применения в схожих задачах
  
- Не учитывается время, прошедшее между сменой состояний.
- Зависимость вероятности перехода лишь от текущего состояний.

# Модификации

Что мы хотим, возможно, применить

- **Фильтр Калмана**

Алгоритм для работы СММ с непрерывными зашумленными наблюдаемыми состояниями.

- **Многоуровневые СММ**

Строится каскад СММ, где каждый следующий уровень генерирует наблюдаемые последовательности для предыдущего.

- **Иерархические СММ**

СММ, где каждое состояние может само являться ИСММ

# End of file

Спасибо за внимание!

Вопросы?