



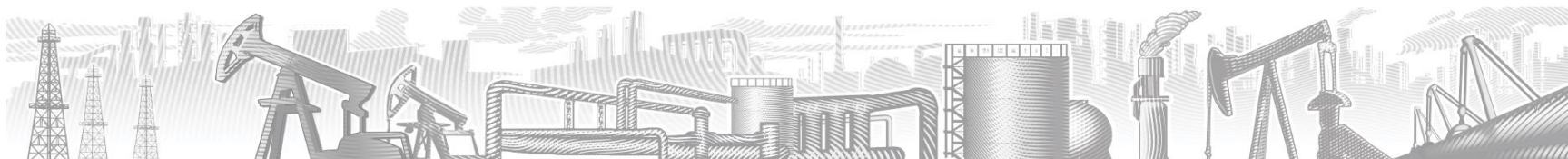
**УФИМСКИЙ ГОСУДАРСТВЕННЫЙ
НЕФТЯНОЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**
ИНСТИТУТ НЕФТИ И ГАЗА (филиал в г. Октябрьском)



Кафедра «Информационные технологии, математика и естественные науки»

Курс ДОП
ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ И НЕЙРОННЫЕ СЕТИ

Разработал д.т.н. Тынчеров Камиль Талятович



Учебные вопросы лекции

1. Классическая парадигма.
2. Логика программного мышления.

Рассматривается высокая эффективность применения логических нейронных сетей при логическом выводе на основе языка ПРОЛОГ. Цели экспериментального дедуктивного вывода дополняют фактографическую базу знаний. Однако формализация исследований этой базы знаний с помощью логической нейронной сети позволяет реализовать зачатки индуктивного мышления, т.е. дополнения понятийной базы знаний новыми правилами вывода.



Изучается эффективность применения логических нейронных сетей при логическом выводе на основе языка ПРОЛОГ. Цели экспериментального дедуктивного вывода дополняют фактографическую базу знаний. Однако формализация исследований этой базы знаний с помощью логической нейронной сети позволяет реализовать зачатки индуктивного мышления, т.е. дополнения понятийной базы знаний новыми правилами вывода.



1. Классическая парадигма.

Парадигма эксперта и парадигма ученика

В основе стратегий *искусственного интеллекта* (ИИ) лежит понятие *парадигмы* - взгляда, концептуального представления сути проблемы или задачи и подхода к ее решению.

Ранее, на первой лекции, мы рассматривали две основные парадигмы: нисходящую и восходящую. Допустили существование третьей – гибридной. Сейчас рассмотрим две основные парадигмы, но немного под другим углом:

1. Парадигма эксперта. Эта *парадигма* предполагает следующие объекты, а также этапы разработки и функционирования системы ИИ:

1. *Формализация знаний* - преобразование экспертом проблемного знания в форму, предписанную выбранной *моделью представления знаний*.
2. *Формирование базы знаний (БЗ)* - вложение формализованных знаний в программную систему.
3. *Дедукция* - решение задачи *логического вывода* на основе *БЗ*.

Эта *парадигма* лежит в основе применения экспертных систем, систем *логического вывода*, в том числе на языке *логического программирования ПРОЛОГ*. Считается, что системы на основе этой *парадигмы* изучены в большей степени.

1. Классическая парадигма.

Парадигма эксперта и парадигма ученика

2. Парадигма ученика.

1. Обработка наблюдений, изучение опыта частных примеров - формирование базы данных (БД) системы ИИ.

2. Индуктивное обучение - превращение БД в *базу знаний (БЗ)* на основе обобщения знаний, накопленных в БД, и обоснование процедуры *извлечения знаний* из БЗ.

Это означает, что на основе данных мы делаем вывод об общности той зависимости между объектами, которую мы наблюдаем. Основное внимание здесь уделяется изучению аппроксимирующих, вероятностных и логических механизмов получения общих выводов из частных утверждений. Затем мы можем обосновать, например, достаточность процедуры обобщенной *интерполяции* (экстраполяции), или процедуры ассоциативного поиска, с помощью которой будем удовлетворять запросы к БЗ.

3. *Дедукция* - по обоснованной или предполагаемой процедуре мы выбираем информацию из БЗ по запросу (например, - *оптимальную стратегию* управления по запросу, характеризующему сложившуюся ситуацию).

Считается, что исследования и разработка в рамках этой *парадигмы* проведены пока слабо, хотя она лежит в основе построения самообучающихся систем управления. Чем *база знаний*, общий и обязательный элемент системы ИИ, отличается от *базы данных*? Возможностью *логического вывода*! Мозг является и носителем *базы знаний*, и средством *логического вывода* на ее основе. И это независимо от того, по какой парадигме мы организовали свое мышление, то есть, каким способом мы заполняем *базу знаний*, - учимся!

2. Логика программного мышления.

Следуя различными путями дедуктивного и индуктивного *мышления*, осуществляя различные *парадигмы* обучения, человек стремился автоматизировать логику *мышления*, немыслимую без формализации. Продуктом этой деятельности явились такие языки *логического вывода*, как ЛИСП и ПРОЛОГ. Язык ПРОЛОГ [\[13\]](#) следует считать венцом усилий по автоматизации *логического вывода*, эффективно описывающего, в частности, *экспертные системы*.

ПРОЛОГ представляет *базу знаний* как совокупность фактов и правил (вывода). Процедурная структура позволяет включать конструкции любых других *алгоритмических языков*. То есть, он является логической надстройкой, сублимирующей лишь *операции логического вывода*.

Формулируется цель *логического вывода*, и если она не противоречива, выявляются факты, из которых эта цель следует.

И сознавая, что Природа создала единственное средство *мышления* — мозг, мы снова и снова пристально раздумываем, как же реализовать то, что гениально воплощено в языке *логического программирования ПРОЛОГ*?

Пример

... Представьте себе село, затерянное в далекой таежной глуши. Навечно изолированные от Большой Жизни, его обитатели долгими зимними вечерами, в перерывах благотворного интеллектуального напряжения игры "Тигра идет", в сумраке демократических потугов вооружившись мозолистыми кулаками, выясняют степень взаимного родства. И тут являетесь Вы!словно светоч озарения, сосланный за непримиримость свободолюбивых устремлений, Вы, наконец, находите для себя непаханое *поле* действительно яркой деятельности, полной гуманизма и самопожертвования. Вы решаетесь положить конец сомнениям, и подобно искусному укротителю, внедряете важные элементы государственного акта переписи населения...

Правила глупейшей игры ТИГРА ИДЕТ!

Все игроки садятся за стол и наливают себе алкоголь на выбор. Затем быстро выпивают все до дна и по команде «Тигр идет» прячутся под стол. Затем процедура повторяется. Проигрывает тот, кто уже не может вылезти из-под стола.

2. Логика программного мышления.

Рассмотрим упрощенную задачу в виде *ПРОЛОГ*-программы, содержащую все характерные элементы решения проблемы удовлетворения (сложной) цели на основе лишь фрагмента *базы знаний (БЗ)*, содержащего факты и правила.

Факты — *клозы* (отдельные *предикаты-высказывания* принято называть *клозами*), которые не содержат правых частей, правила — *клозы*, которые содержат правые части; одноименные факты и правила объединяются в процедуры.

База знаний

Процедура "мужчина":

мужчина (иван) мужчина (василий) мужчина (петр) мужчина (федор) мужчина (юрий)

Процедура "женщина":

женщина (марья) женщина (ирина) женщина (ольга) женщина (елена)

Процедура "родитель":

родитель (марья, иван) (Читать: "Марья — родитель Ивана") родитель (иван, елена) родитель (марья, василий) родитель (федор, марья) родитель (петр, ирина) родитель (петр, иван) родитель (федор, юрий)

Процедура "мать":

мать (X, Y): — женщина (X), родитель (X, Y)

Процедура "отец":

отец (X, Y): — мужчина(X), родитель (X, Y)

Процедура "брат":

брат (X, Y): — мужчина (X), родитель (P,X), родитель (P, Y), $X \neq Y$

Процедура "сестра":

сестра (X, Y): — женщина(X), родитель(P, X), родитель (P, Y), $X \neq Y$

Процедура "дядя":

дядя (X,Y): — брат (X, P), родитель (P, Y)

)

Пусть задана некоторая сложная (т.е. опирающаяся не на факт, а требующая вывода) цель, с которой мы обратились в эту БЗ, например:
дядя (X, Y) (запись цели образует фрейм),
и ее решение (*вывод*) заключается в нахождении всех пар переменных (имен объектов) X и Y, для которых справедливо утверждение "X является дядей Y".

Для решения такой задачи используется *прием трансформации цели*, который заключается в рекурсивном переборе различных вариантов подстановки вместо предикатов, составляющих сложную цель, фактов или правых частей клозов соответствующих процедур. Производится фиксация варианта связывания переменных и *унификация*, при которой отбрасываются несовместимые варианты связывания, противоречащие фактам. Варианты связывания всех переменных, прошедшие все этапы унификации, являются решением.

То есть, для решения данной задачи необходимо действовать следующим образом.
Находим первый (а он и единственный) *предикат* цели дядя (X, Y). Находим в БЗ процедуру с этим именем и заменяем найденный *предикат* правой частью этой процедуры. Получим трансформированную цель — *фрейм*
брат (X, P), родитель (P, Y).

К первому предикату этого фрейма применяем аналогичные действия, получаем *фрейм*
мужчина(X), родитель(Q, X), родитель(Q, P), $X \neq P$, родитель(P, Y)

(Во избежание *коллизии*, развивая *фрейм* цели, вводим новые переменные, отличные от тех, которые до того уже были использованы.)

Вновь *входим в процедуру* с именем первого предиката цели. Начинаем первый уровень *ветвления*. А именно, первый клон процедуры — факт мужчина (иван). С его помощью производим первое *связывание* переменных, т.е. подстановку конкретного значения. *Предикат* цели, породивший факт, т.е. это *связывание*, может быть исключен из трансформируемой цели, т.е. заменен своей "пустой" правой частью:

родитель (Q, иван), родитель (Q, P), иван $\leftrightarrow$ P, родитель(P,Y).

Обращаемся к процедуре "родитель", начиная второй уровень *ветвления*. Первый клон этой процедуры родитель (марья, иван) определяет дальнейшее *связывание* переменных:

родитель (марья, P), иван $\leftrightarrow$ P, родитель (P, Y).

Вновь *входим в процедуру* "родитель" (третий уровень *ветвления*), находим клон родитель (марья, иван). Трансформируем цель — получаем новый *фрейм*:

иван $\leftrightarrow$ иван, родитель (иван, Y).

Получаем противоречие, т.е. не проходит *унификация*.

Ищем на данном шаге *ветвления* другой вариант связывания, находим следующий клон родитель (марья, василий).

Трансформируем цель:

иван $\leftrightarrow$ василий, родитель (василий, Y) $\rightarrow$ родитель (василий, Y).

Вновь *входим в процедуру* "родитель", но не находим там клоза, в котором василий указан как чей-либо родитель. Т.е. вновь не проходит *унификация* — установление совместимости варианта связывания переменных.

Возвращаемся на шаг *ветвления* назад. (Реализуем стратегию поиска с *ветвлением* и возвращением назад — "*backtraking*".) На втором уровне *ветвления* пробуем клон, в котором

иван указан как сын: родитель (петр, иван). Цель трансформируется в следующий *фрейм*

Вновь (на третьем уровне *ветвления*) обращаемся к процедуре "родитель" и выбираем первый клов, в котором петр указан как отец — родитель (петр, ирина).

Цель трансформируется:

иван <> ирина, родитель (ирина, Y) → родитель (ирина, Y).

Входим в процедуру "родитель", но не находим там клова, в котором ирина указана как родитель. (Не проходит унификация.)

Возвращаемся на второй уровень *ветвления* и не находим там больше клозов, где иван указан как сын. Возвращаемся на первый уровень *ветвления* и в процедуре "мужчина" выбираем для последующего испытания следующий клов мужчина (василий).

Цель принимает вид фрейма

родитель (Q, василий), родитель (Q, P), василий <> P, родитель (P, Y).

Теперь на втором уровне *ветвления* находим первый (и единственный) клов, в котором василий указан как сын. Цель трансформируется в соответствии с новым связыванием переменных, обусловленным найденным кловом родитель (марья, василий):

родитель (марья, P), василий <> P, родитель (P, Y).

На третьем уровне *ветвления* находим первый клов, где марья — родитель: родитель (марья, иван). Связываем тем самым переменные, цель трансформируется:

василий <> иван, родитель (иван, Y) → родитель (иван, Y).

Находим в процедуре "родитель" первый клов, в котором иван указан как родитель — родитель (иван, елена). Цель выродилась, значит

дядя (X, Y) = дядя (василий, елена) — одно из решений задачи.

Продолжив перебор так, словно на данном шаге *унификация* не прошла, можно найти остальные решения: дядя (юрий, иван), дядя (юрий, василий).

В основе *распараллеливания* решения этой задачи лежит способ размножения вариантов на основе трансформации цели. Способ обеспечивает отсутствие " *backtracking 'а'* " (*ветвление* есть, а возврата назад нет), простоту самой процедуры вывода, возможность неограниченного использования *ИЛИ-параллелизма* (одновременной независимой обработки многих вариантов связывания переменных), конвейерную реализацию *И-параллелизма* (*распараллеливания* обработки одного варианта связывания переменных на конвейере, т.к. каждый раз обрабатывается лишь первый *предикат* каждого фрейма). Однако представляется, что нейросетевая технология, основанная на естественном параллелизме, может оказаться эффективной.

Нейронная сеть для решения задачи логического вывода

Рассмотрим предварительно один аспект обучения нейросети методом трассировки.

Поясняя "схемотехнический" подход к построению нейросети, мы предполагали, что никаких связей ("проводочков") не было, а мы их проложили, исходя из реализуемых *логических функций*. Обобщив подход на *нечеткую логику* (предполагающую вхождение элементов в состав *множества* с некоторой вероятностью), а также логику *передаточных функций*, мы заявили, что нашли способ построения уже обученных нейросетей.

Впоследствии мы заявили, что нас не смущает отсутствие некоторых связей в "готовых" сетях. Мы их введем как "проводочки", если сочтем это целесообразным.

Таким образом, несовершенство нейросети не стало для нас преградой. А если предположить, как при "схемотехническом" подходе, что первоначально в сети вообще никаких связей нет, ее *матрица следования* пуста, а нам предстоит создать эту *сеть* полностью? Логично, что в ней будут присутствовать только те связи, которые обеспечивают ее обучение. То есть, мы сразу можем строить обученную *нейросеть*, — *нейросеть*, *синаптические связи* в которой имеют единичный (максимальный) *вес*.

Составим ([рис.10.1](#)) *нейросеть*, соответствующую фрагменту беспредельной *базы знаний* о жителях далекого таежного села.

Нейронная сеть для решения задачи логического вывода

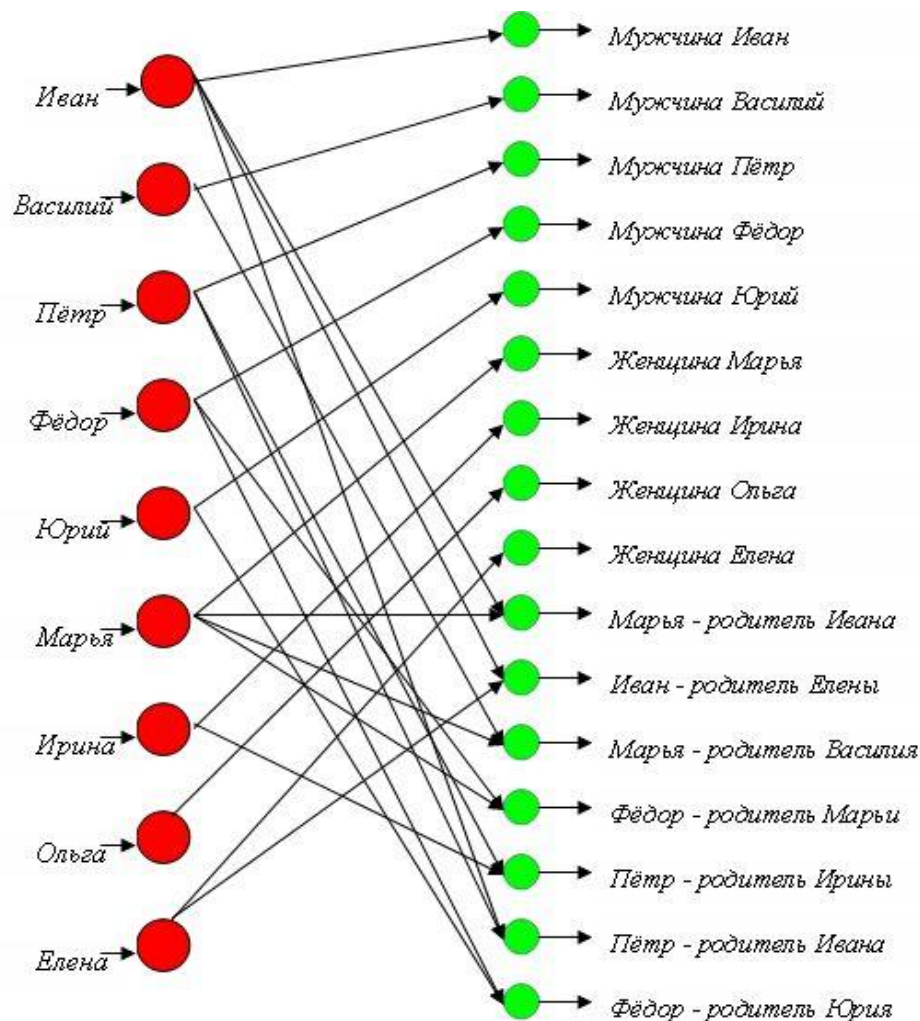


Рис. 10.1. Начальный вид фактографической нейросети

Нейронная сеть для решения задачи логического вывода

Это – начальный вид, т.к. мы надеемся дополнить ее до состояния насыщения в результате фиксирования новых фактов, получаемых в длительном рабочем режиме эксплуатации, в процессе решения задач *логического вывода*. Такое развитие фактографической нейросети позволит исключить трудоемкий повторный *вывод*, воспользоваться однажды установленным фактом, словно записью на щите орудия о пораженных целях.

Для реализации правил построим понятийную *нейросеть*, фрагмент которой показан на [рис.10.2](#). *Рецепторный слой*, несомненно, содержит элементы избыточности. Однако различие в обозначениях повторяющихся переменных (и закрепленных за ними *рецепторов*) позволяет избежать *коллизии* при формировании выводов на основе формального описания правил.

Пусть задана та же сложная цель дядя (X, Y).

Для нее необходимо сформировать все пары переменных X и Y, для которых справедливо утверждение "X является дядей Y".

Выберем следующую *функцию активации*:

$$V = \sum_j V_j \quad V = \sum_j V_j$$
$$V_j = \begin{cases} V - h, & \text{если } h \leq V < 1, \\ 1 = h, & \text{если } V \geq 1 \\ 0, & \text{если } V < h \end{cases} \quad V_j = \begin{cases} V - h, & \text{если } h \leq V < 1, \\ 1 = h, & \text{если } V \geq 1 \\ 0, & \text{если } V < h \end{cases}$$

Порог h подбирается экспериментально, в данном случае, кажется, целесообразно положить $h = 0,4$.

Нейронная сеть для решения задачи логического вывода

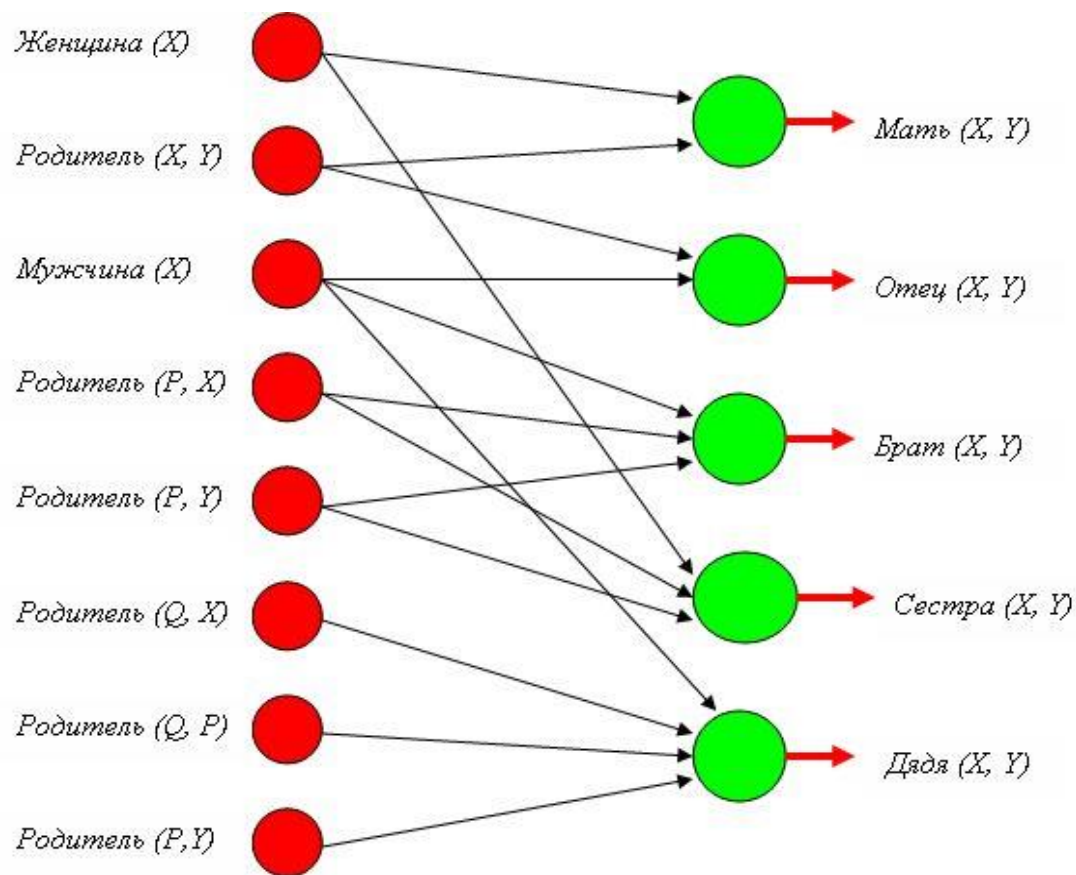


Рис. 10.2. Понятийная нейросеть

Нейронная сеть для решения задачи логического вывода

Чтобы максимально возбудить *нейрон выходного слоя*, ответственный за дядю, необходимо подать высокий сигнал возбуждения на рецепторы, передающие возбуждение этому нейрону, выполнив все варианты *непротиворечивого* связывания переменных X, Q, P, Y.

Таким образом, надо реализовать ту же схему перебора с возвратом, что была рассмотрена ранее, — схему *backtracking* 'а.

Проследим *по* шагам необходимые действия.

Полагаем X = иван.

В процедуре Родитель находим Q = мария.

В той же процедуре находим P = василий.

Однако, в той же процедуре не находим указаний на то, что Василий является чьим-то родителем. Возвращаемся на шаг назад, пытаюсь изменить связывание переменной P, но не находим нового варианта связывания.

Возвращаемся еще на шаг и т.д. — повторяем весь ход рассуждений, рассмотренных в подразделе 10.1.

Нейронная сеть для решения задачи логического вывода

После нахождения пары (василий, елена) следует развить фактографическую *сеть*, дополнив ее *выходной слой* нейроном в соответствии с понятием дядя и установив связи с ним от василия и елены.

Это может быть сделано двумя способами.

Первый способ повторяет весь ход логических рассуждений, связывая *высказывания*, приведшие к данному заключению. Тогда в результате решения всевозможных задач вывода установится окончательный вид фактографической нейросети, представленный на [рис.10.3](#). (Для упрощения рисунка некоторые входные связи недозволенно объединяются в "*кабель*".)

Как работать с этой сетью?

Можно себе представить два режима этой работы: для фискального органа — налоговой инспекции, требующей общей картины социального состояния, и для информационно-аналитической службы безопасности, интересующейся гражданами индивидуально, *по* выбору и привлечению.

А именно, если мы "засветим" (сообщив единичное возбуждение) все нейроны-рецепторы, то мы сможем прочесть все, что *нейросеть* "знает" о всех своих клиентах: кто из них мужчина, кто — женщина, в каком родстве они состоят.

2. Логика программного мышления.

Нейронная сеть для решения задачи логического вывода

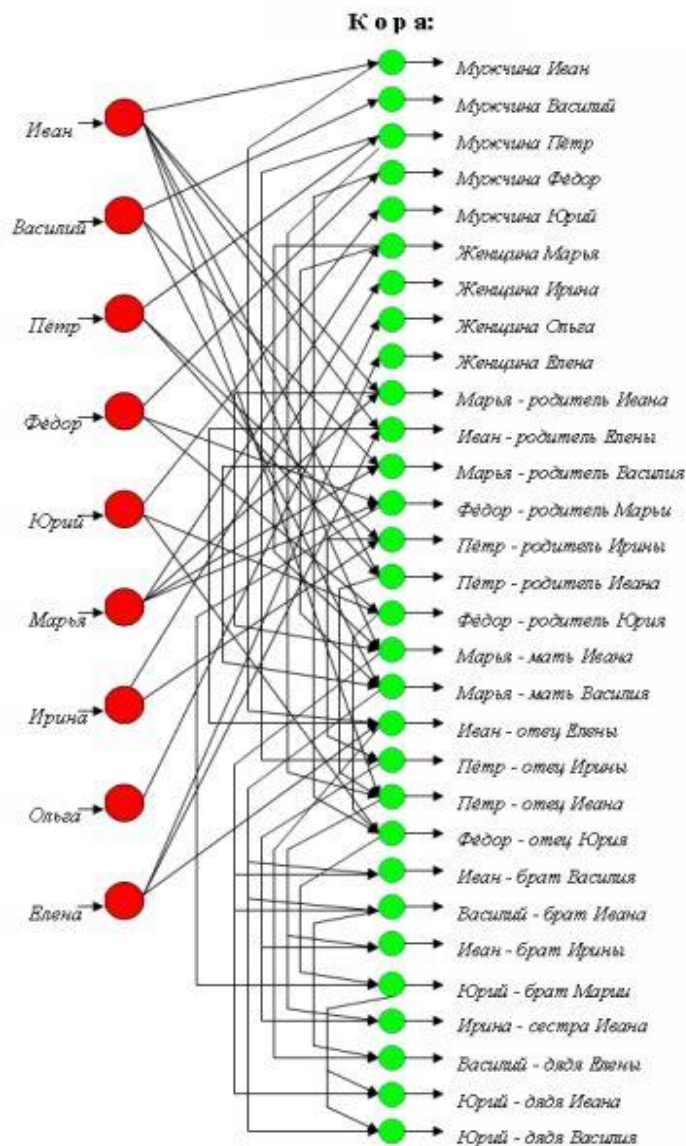


Рис. 10.3. База Знаний Антрополога – Исследователь1

Нейронная сеть для решения задачи логического вывода

Но мы можем спросить *сеть* лишь о некоторых персонажах. Например, мы можем возбудить на входе нейроны-рецепторы, соответствующие Ивану и Василию. Мы получим информацию о том, что они, несомненно, мужчины.

Но подозрительно высокого возбуждения достигнут нейроны, отвечающие за факт, что Марья — мать Ивана и она же — мать Василия.

Подозрительно возбудятся и нейроны, отвечающие за факты Иван — брат Василия и Василий — брат Ивана, Юрий — дядя Ивана и др. Тогда мы решаем: "А ну-ка, возбудим *нейрон-рецептор*, соответствующий Марье, и посмотрим, что интересного нам *сеть* сообщит!"

И *сеть* возбуждением своих нейронов нам сообщает: "Да, действительно, Марья — мать Ивана и Василия".

Заметим, что мы можем подавать на входе и не единичные значения возбуждений. Например, установив, что в "деле" участвовал не то Иван, не то Василий (уж очень они похожи!), мы можем *по* принципу "фифти-фифти" задать соответствующие значения возбуждений равными 0,5 и установить высокую величину возбуждения *нейрона*, соответствующего дяде и того и другого — Юрию. И тогда логично возложить на него ответственность за нерадивое воспитание племянников.

Нейронная сеть для решения задачи логического вывода

Таким образом, составляя эту *нейросеть*, мы руководствовались желанием сохранить причинно-следственные логические связи, обусловленные правилами. Это отражает ориентацию нейросети на пытливый ум, методом индуктивного *мышления* старающийся выяснить, какие факты достаточны для установления, например, того, что Юрий — дядя Ивана. Мы даже назвали эту *нейросеть* Базой Знаний Антрополога-Исследователя. Здесь же мы пошли на организацию коры. Мы справедливо решили, что все выводы данной сети должны быть равноправными и служить как формированию выводов "внутри себя", так и применению "вне" — для решения некоторой глобальной задачи, сверхзадачи. Равноправность и внешнее использование, в том числе для логической надстройки сети, для получения новых знаний *по* новым вводимым фактам и правилам, требует ревизии и особого установления величин возбуждения нейронов, что мы и отразили в *передаточной функции*.

Но мы ловим себя на лукавом мудрствовании... "Зачем так сложно? — спросит рядовой гражданин, к каковому следует отнести жителя далекого села. — Какое мне дело до намеков твоих? Ты давай напрямик, факты давай!.."

Действительно, зачем нам знать и делать *вывод* о том, что Юрий — дядя Ивана, потому что он — брат Марьи? Мы можем эти два факта задать независимо, предварительно рассчитав *по* понятийной нейросети.

Нейронная сеть для решения задачи логического вывода

Тогда фрагмент БЗ, которую мы назвали Базой Знаний Участкового Уполномоченного, представлен на [рис.10.4](#), и мы можем вздохнуть облегченно!

И не надо никакой коры! Строго, *по-военному* и как в анкете: посылка — следствие! Не надо сомнений, размышлений, ревизий и прочего вольнодумствия!

Таким образом, реализовав все причинно-следственные связи, мы построили *нейросеть*, отображающую только факты. Размножив ее для работы многих пользователей, можно получить достаточную информацию и исключить из рассмотрения правила вывода.

Рассмотренная схема работы с нейросетью отображает модель дедуктивного *мышления*. Но можно ли на основе полученной нейросети произвести обратные действия, построить обобщения, получить правила, достойные индуктивного *мышления*?

По-видимому, необходим *анализ* нейросети, сохранившей причинно-следственные связи, для выделения идентичных подструктур, объединяющих факты из общих процедур – носителей понятий.

2. Логика программного мышления.

Нейронная сеть для решения задачи логического вывода

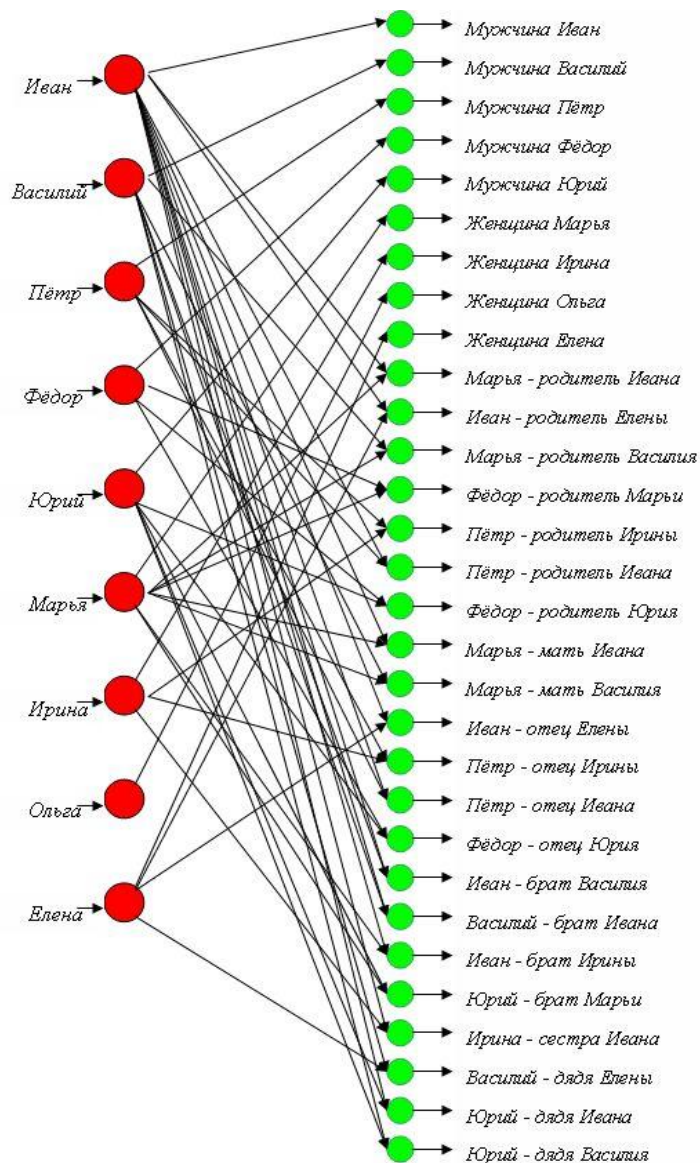


Рис. 10.4. База Знаний Участкового Уполномоченного

Нейронная сеть для решения задачи логического вывода

Например, пути (*статические цепочки*), приводящие к выводу дядя (василий, елена), дядя (юрий, иван), дядя (юрий, василий) в одинаковом порядке связывают "представителей" одних и тех же процедур. Тогда, выбрав вместо конкретных переменных их абстрактное обозначение, можно восстановить *правило вывода*. А это уже *гипотеза* — важный элемент индуктивного мышления. На основе этой гипотезы может быть развита *понятийная нейросеть*. Но *гипотеза* будет оставаться гипотезой, пока в достаточной мере не подтвердится практикой.

Тогда *нейросеть* обретает возможность широкого развития. Ведь если население села пополнилось родившимся младенцем, то правильность построенного правила (гипотезы) может быть успешно проверена!

... Да, мы вправе выбирать способ *мышления* как принцип жизни, — для ее облегчения...

Ключевые термины

ПРОЛОГ — язык *логического вывода* на основе процедур-фактов и процедур-правил. По заданной цели, объединяющей ряд признаков или отношений, обеспечивает выдачу всех объектов, обладающих этими признаками, или удовлетворяющих указанным отношениям.

Фактографическая база знаний — представленная *логической нейронной сетью база знаний*, содержащая только факты.

Понятийная база знаний — представленная *логической нейронной сетью база знаний*, объединяющая правила (вывода).

Backtracking — выполнение очередного шага (попытки) связывания переменных на пути удовлетворения цели с проверкой на совместимость (*унификация*) возникшей комбинации связывания. Совершается отход назад для анализа другой комбинации при неудачной унификации. Реализует известную схему перебора, называемую методом "ветвей и границ".

Выводы.

1. Формализация *базы знаний* на основе *логической нейронной сети*, отображающей факты и правила вывода, позволяет производить формальные алгоритмические действия (удовлетворения целей) по получению новых фактов.
2. *База знаний* мыслящего существа должна наряду с фактами хранить хотя бы начальные абстрагированные правила вывода, обобщения, сообщенные в процессе обучения (от рождения – до смерти). Только факты и правила вывода лежат в основе дедуктивного *мышления*.
3. Следует отметить главенствующую роль перебора (поиска) в алгоритмах *логического вывода*. Перебор сопровождается анализом на противоречие промежуточных результатов фактам и правилам вывода, лежащим в основе *базы знаний*. (Здесь уже кроется возможность критического отношения к "современному" состоянию *БЗ*, противодействие сложившимся догмам!)
4. В согласии с главенствующим перебором при поиске решения или всего лишь гипотезы, реализуются "*backtracking*" другие многочисленные методы "ветвей и границ".
5. Понятно, что такого рода перебор, требующий высокого уровня ясности *мышления*, усидчивости и трудолюбия, не доступен широким массам трудящихся и нуждается в автоматизации.
6. Понятийная *база знаний*, представленная *логической нейронной сетью*, является важным формальным механизмом не только для получения новых фактов, но и для построения алгоритма новых обобщений на основе накопленного опыта и результатов вывода по текущему состоянию *БЗ*. Так могут формироваться гипотезы о новых правилах вывода. Это – путь к автоматизированной реализации индуктивного *мышления*, на основе теории познания, использующей понятия анализа и синтеза (знаний).

Выводы.

7. Проверка гипотез с помощью фактов, получаемых опытным путем, позволяет расширить понятийную нейронную сеть (*базу знаний*) новыми правилами вывода.
8. Целью построения формальных алгоритмов индуктивного *мышления* и, следовательно, его автоматизации является ускоренное получение новых обобщенных знаний – прогресс человечества.
9. ПРОЛОГ является гениальным языком, формализующим логический вывод. Формализация действий над ним с помощью механизмов логических *нейронных сетей* убеждает в продуктивности попыток автоматизации как дедуктивного, так и индуктивного *мышления*.
10. На уровне оперирования только лишь фактами ПРОЛОГ обладает большими возможностями целевого выбора информации, чем реляционные базы данных. Жаль, что этот язык *логического вывода*, возникший в период появления серьезных задач и продемонстрировавший свою эффективность, предан забвению, по-видимому, в связи с кризисным исчезновением таких задач.

1. Что собой представляет ПРОЛОГ-программа и как реализуется *backtracking* ?
2. Что представляет собой фактографическая *нейронная сеть*?
3. Что представляет собой понятийная *нейронная сеть*?
4. Достаточно ли в обыденной жизни пользоваться развитой фактографической *нейронной сетью*, осуществляя только дедуктивное мышление и пополняя ее новыми фактами?
5. Осуществимо ли индуктивное мышление на основе анализа и обобщения связей, сложившихся в фактографической *нейронной сети*?
6. Вопрос к размышлению: Достаточен ли аппарат логических *нейронных сетей* для анализа связей, сложившихся в фактографической *нейронной сети* – для их обобщения и формирования гипотез, или требуются алгоритмические механизмы *мышления* более высокого порядка?

1. Галушкин А.И **Нейронные сети: основы теории** М.: «Горячая линия – Телеком», 2010
2. Rosenblatt F. (Ф. Розенблат) **Perceptual generalization over transformation groups (Обобщение восприятий по группам преобразований)** Self-Organizing Systems, Pros. Interdisciplinary Conf., 5 – 6 May, 1959, 63 – 100 (Кибернетический сборник, 1962)
3. Поспелов Д.А **Моделирование рассуждений. Опыт анализа мыслительных актов.** М.; Радио и связь. 1989
4. Поспелов Д.А **Ситуационное управление. Теория и практика** М.: Наука, 1986
5. Барский А.Б. **Математическая логика событий и логические нейронные сети.** Приложение к журналу «Информационные технологии» №7, 2007
6. Барский А.Б. **Логические нейронные сети. Учебное пособие** М.: ИНТУИТ; БИНОМ. Лаборатория знаний. 2007
7. Барский А.Б. **Нейронные сети: распознавание, управление, принятие решений.** М.: Финансы и статистика, 2004, 2007
8. Барский А.Б. **Трёхмерная экранизация компьютерных объектов «живого» моделирования** Информационные технологии. 2010, №10
9. Барский А.Б. **Параллельные информационные технологии** М.: ИНТУИТ; БИНОМ. Лаборатория знаний. 2007
10. Ярушкина Н.Г. **Основы теории нечётких и гибридных систем.** М.: Финансы и статистика, 2004
11. Понс Ж, Форсайт Д. **Компьютерное зрение. Современный подход** М.: Издательский дом «Вильямс» 2004
12. Вахромцев Е.М. **Применение методов «живого» моделирования для производственного мониторинга** Информационные технологии. 2010
13. Ivan Bratko (И. Братко) **Programming for artificial intelligence (Программирование на языке Пролог для искусственного интеллекта)**
14. E. Kardelj – J. Stefan Institute. Yugoslavia. 1986 (М.: Мир, 1990)
15. Поспелов Д.А **Десять «горячих точек» в исследованиях по искусственному интеллекту** Интеллектуальные системы (МГУ). Т. 1. Вып. 1...4. 1996
16. [НОУ ИНТУИТ | Лекция | Логическое программирование нейронной сети на базе языка ПРОЛОГ \(intuit.ru\)](http://intuit.ru)

Спасибо за внимание!