# 20 Классификация онтологий. Уровни формализации и языки представления онтологий.

**КЛАССИФИКАЦИЯ ОНТОЛОГИЙ**

**метаонтология** – содержит общие понятия и отношения, не зависящие от предметной области («объект», «свойство», «значение» и т.п.);

**предметная онтология** – содержит понятия, описывающие конкретную предметную область и отношения, семантически значимые для данной предметной области;

**онтология задач** (и методов) – содержит в качестве понятий типы решаемых задач, а отношения специфицируют декомпозицию задач на подзадачи;

**прикладная онтология** (онтология приложения) – описывает концепты, зависящие как от предметной области, так и от задач;

**сетевая онтология** – используется для описания конечных результатов действий, выполняемых объектами предметной области или задачи.

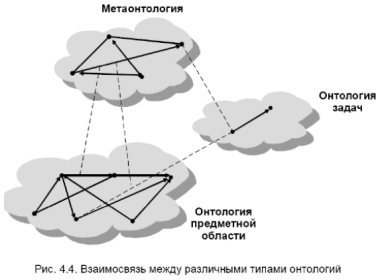
Другие классификации:

• по уровню детализации

• по “природе” предметных областей

• по степени разработки и сопровождения

**ВЗАИМОСВЯЗЬ МЕЖДУ РАЗЛИЧНЫМИ ТИПАМИ ОНТОЛОГИЙ**



**УРОВНИ ФОРМАЛИЗАЦИИ ОНТОЛОГИИ**

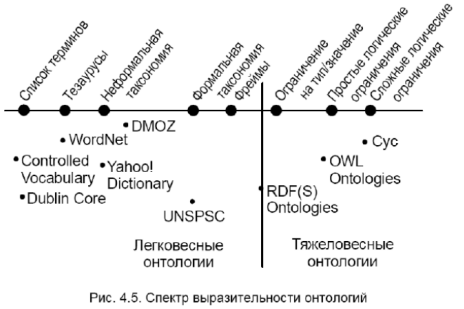
1 **Неформализованный**. Слабо описанная на естественном языке онтология.

2 **Неформально-структурированный**. Онтология описана ограниченным набором лексики естественного языка в структурированном виде.

3 **Полуформальный**. Онтология описана на специальном формально определенном языке.

4 **Строго формальный**. Для описания используются тщательно определенные термины с формальной семантикой, теоремы и доказательства для таких свойств, как надежность и полнота онтологии.

**СПЕКТР ВЫРАЗИТЕЛЬНОСТИ ОНТОЛОГИЙ**



**ЯЗЫКИ ПРЕДСТАВЛЕНИЯ ОНТОЛОГИЙ**

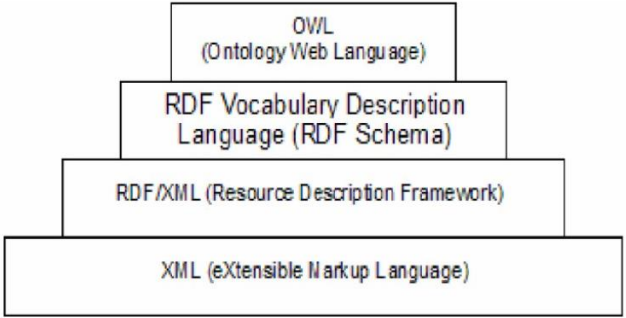
Традиционные языки спецификации онтологий (LOOM, OKBC, OCML, Flogic, LBase)

специальные языки спецификации онтологий (Ontolingua, CycL, SHOE)

языки основанные на Web-стандартах (UPML, DAML, OIL, XODL, XML, RDF, RDFS, OWL)

Выбор языка зависит от целей разработки онтологии.

**ЛОГИЧЕСКАЯ ЗАВИСИМОСТЬ ЯЗЫКОВ ПРЕДСТАВЛЕНИЯ ОНТОЛОГИЙ**



**OWL (WEB ONTOLOGY LANGUAGE) – ЯЗЫК WEBОНТОЛОГИЙ**

Основное назначение OWL – для использования приложениями, которые должны не только представлять информацию человеку, но и обрабатывать ее.

OWL спроектирован для хранения и отображения большого количества разнородной информации и знаний, а также для поддержки семантики и связей между этими знаниями.

OWL предназначен для описания классов и отношений между ними, которые присущи для Web-документов и приложений.

# 21 Жизненный цикл онтологий. Средства построения онтологий.

1. Планирование (задачи, ресурсы, сроки)
2. Разработка
   * Спецификация – определяет цели создания, предполагаемое использование и потенциальных пользователей
   * Концептуализация – обеспечивает структурирование предметных знаний
   * Формализация – трансформирует концептуальную модель в формальную или вычислительную
   * Реализация – вычислительная модель программируется на соответствующем языке представления знаний
   * Проверка – включает в себя идентификацию и редактирование неопределенных терминов; выявление конфликтов и несогласованностей; проверку онтологии на полноту; идентификацию семантических различий между двумя терминами в различных онтологиях; синтаксический анализ; анализ таксономии; семантическую оценку
3. Поддержка (одновременно с разработкой) – приобретение знаний, оценка онтологии, интеграция при построении новой онтологии с использованием существующих



Построить можно в Protégé, CMapTools, FaCT

# 22 Основоположники генетики. Основные понятия генетики.

Основоположником генетики является Иоганн Грегор Мендель, который в середине XIX в. открыл общий закон природы и вывел формулы расщепления признаков в гибридном потомстве. Известны три знаменитых закона И. Менделя:

1. закон однородности и реципрокности (взаимность, эквивалентность; механизм совместного достижения эволюционного преимущества). Первое гибридное поколение оказывается полностью однородным.
2. закон расщепления. При скрещивании гибридов первого поколения в потомстве происходит расщепление признаков по фенотипу 3:1, а по генотипу в отношении 1:2:1 0;
3. закон независимой комбинации. Закон справедлив для потомков родителей, отличающихся более чем одной парой признаков, и говорит о том, что признаки наследуются независимо друг от друга.

***Жан Батист Ламарк*** – эволюция как возникновение новых форм путем постепенного изменения старых. При этом происходит усложнение форм. Он считал, что организмы изменяются под прямым воздействием окружающей среды, что причиной эволюции являются «упражнения» органов и внутреннее стремление к прогрессу.

***Чарльз Дарвин*** – в живой природе существование общего принципа – естественного отбора. Он различал 2 стороны эволюционного учения:

* О материале для эволюции
* О факторах

Движущая сила эволюции – естественный отбор.

***Чарльз Вильсон*** – хромосомы состоят из генов. Модель хромосомы в настоящее время – то нить, на которую нанизаны гены.

***Вильгельм Иогассен*** – ввел термины Ген, Генотип, Фенотип, Аллель.

***Генетика*** *—* биологическая наука, изучающая наследственность и изменчивость живых организмов.

***Эволюция*** *— процесс постепенного и непрерывного изменения форм организмов от одного состояния к другому.*

***Адаптация***— процесс, а также результат приспособления строения и функций организмов к условиям внешней среды.

***Изменчивость*** *— разнообразие признаков и свойств у особей и групп особей любой степени родства.*

***Естественный******отбор*** *—* общий принцип живой природы. Движущая сила эволюции. Процесс, направленный к повышению (или понижению) вероятности оставления потомства одной формой организмов по сравнению с другими.

***Хромосомы*** *—* представляют собой нитевидные структуры, находящиеся в клеточном ядре. Моделью хромосомы является нить, на которую «словно бусинки» нанизаны гены.

***Ген*** *—* единица наследственной информации, неделимая в функциональном отношении, которая передается от родителей к потомкам. Она рассматривается как участок молекулы ДНК, кодирующий синтез одной макромолекулы или выполняющий какую-либо другую элементарную функцию.

***Генотип***— совокупность всех генов, находящихся в хромосомах организма. Генетическая конструкция организма.

***Фенотип*** *—* совокупность внешних признаков, характеризующих организм.

***Геном*** *—* комплекс генов, содержащихся в наборе хромосом одного организма (копия всех генов организма).

***Аллель***— различные формы гена.

***Локус***— место нахождения конкретного гена на хромосоме.

***Рекомбинация******—*** перераспределение наследственных факторов.

***Селекция —*** форма искусственного отбора, где эволюция направляется факторами внешней среды.

***Мутации******—*** наследственные изменения отдельных генов.

***Инверсии*** *— повороты участка или всей хромосомы на 180°.*

***Генетическая изменчивость***— изменения, произошедшие в структуре генотипа и передаваемые по наследству.

***Генные мутации***— принцип преобразования хромосом, в процессе реализации которого участвует, как правило, один или несколько генов. При этом один ген (или их последовательность) может превратиться в другой, может выпасть либо дублироваться, а группа генов может развернуться на 180°.

***Хромосомные мутации*** *— конструкция, в процессе построения которой происходит изменение числа, размеров и организации хромосом. Существуют внутрихромосомные и межхромосомные перестройки.*

***Транслокации* —** межхромосомные перестройки, при которых участок хромосомы перемещается (транслоцируется) на другое место.

***Транспозиции******—*** межхромосомные перестройки, при которых участок хромосомы изменяет свое положение или включается в другую хромосому без взаимного обмена.

***Кроссинговер*** (скрещивание) – обмен частями хромосом

# 23 Эволюция и модели эволюции.

Под эволюцией понимаются медленные, постепенные количественные и качественные изменения объекта. При этом каждое новое состояние объекта, как правило, имеет по сравнению с предыдущим более высокий уровень развития и организации.

Эволюция приводит к формированию адаптации (приспособлений) организмов к условиям их существования.

В науке под адаптацией понимают процесс накопления и использования информации в системе, направленный на достижение ее (системы) оптимального состояния, при первоначальной неопределенности и изменяющихся внешних условиях.

Это условная структура, реализующая процесс, посредством которого особи некоторой популяции, имеющие более высокое функциональное значение, получают большую возможность для воспроизведения потомков, чем «слабые» особи. Такой механизм часто называют методом «выживания сильнейших».

1. Модель эволюции Дарвина

Эволюция по Дарвину состоит из следующих положений:

* в природе все подвержено неопределенной наследственной изменчивости, производится потомство, отличающееся по многим признакам;
* все организмы в природе размножаются в геометрической прогрессии, но численность всех организмов в среднем остается более или менее постоянной, она колеблется около средней величины;
* основой отбора является метод «выживания сильнейших».



*Популяция.* Пусть существует популяция особей (например, табун  
лошадей) на некоторой территории обитания (степь). Табун (попу­  
ляция) насчитывает 50 взрослых лошадей (особей).

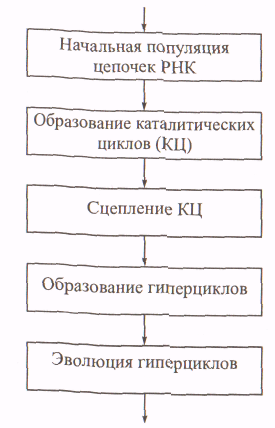
*Наследственность.* Каждый год в табуне (популяции) рождаются  
жеребята (новое поколение, потомство).

*Изменчивость.* Все родившиеся жеребята (потомки) отличаются  
друг от друга и от родителей цветом, размерами, ростом, физиче­  
скими данными (скорость, выносливость и т.д.).

*Отбор.* Наибольшие шансы выжить и закрепиться в табуне (по­  
пуляции) имеют физически более мощные жеребята (способность к самозащите, возможность победить соперников, способность к воспроизводству), обладающие большей скоростью (возможность спастись от хищников), выносливостью (способность преодолевать большие расстояния в поисках пищи). В то же время в ходе эволюции могут появляться особи с так называемыми «полезными» отклонениями от доминирующего в табуне (популяции) «стандарта» (генотипа), которые позволяют их носителям успешнее выживать в существующих условиях.

*Эволюционная смена форм.* Именно такие особи, как правило, выживают в результате схемы эволюции Дарвина, и они постепенно, поколение за поколением становятся преобладающим (доминирующим) видом в данной популяции.

1. Модель эволюции Эйгена и Шустера (модель гиперциклов)



Моделирует условную стадию эволюции. М. Эйген на основе гиперциклов описал добиологическую фазу эволюции, в ходе которой происходят процессы отбора, выражающие свойства вещества в особых системах реакций. Они известны как каталитические циклы.

М. Эйген отмечал, что в далеких от равновесия биохимических системах каталитические реакции объединяются, формируя сложные сети, в которых могут содержаться и замкнутые циклы.

М. Эйген установил, что в условиях достаточного времени и непрерывного потока энергии каталитические циклы сцепляются, образуя замкнутые циклы, в которых ферменты, созданные в одном цикле, являются катализатором в последующем цикле.

Он назвал гиперциклами те петли, в которых каждый узел представляет собой каталитический цикл.

1. Модель эволюции Ламарка

Основана на предположении, что характеристики, приобретенные особью (организмом) в течение жизни, наследуются его потомками.

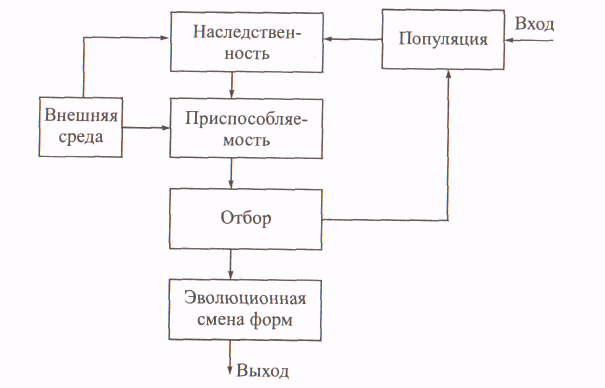
Эти изменения, как утверждал Ж. Ламарк, вызываются прямым влиянием внешней среды, упражнением органов и наследованием приобретенных при жизни признаков.

Он объясняет одну из особенностей эволюции органического мира приспособляемостью. Прогрессивную эволюцию, появление форм, более сложных и совершенных, он объяснял «законом градаций» — стремлением живых существ усложнять свою структуру.

Согласно Ж. Ламарку виды эволюционируют, приспособляясь и усложняясь, потому что у них существуют свойства — приспосабливаться и усложняться.

Причины направленных изменений объясняются различно, но их можно свести к двум:

* направленное влияние внешней среды;
* способность самого организма.



Рассмотрим схему эволюции на уже знакомом нам сообществе особей (табун лошадей) из предыдущего примера.

*1-2* *Популяция + Наследственность.* Каквидно из схемы, первые два пункта у нас остались неизменными. Так же существует табун лошадей на некоторой территории обитания. Каждый год в табуне появляется потомство.

*3. Внешняя среда.* Происходит воздействие внешних факторов в ви­де сильной засухи в степях, где обитает наш табун. Трава в степи выгорела от засухи и табун в поисках пищи вынужден мигриро­вать на соседние территории, являющиеся предгорьями. На этих  
территориях более мягкий климат и трава все еще сохранилась.

*4. Приспособляемость.* В условиях изменившейся среды обитания (предгорья) популяции (табуна) более предпочтительные шансы на добычу корма и, следовательно, на выживание имеют коротко­ногие, невысокие особи, которым гораздо удобнее передвигаться в условиях пересеченной гористой местности.

*5-6. Отбор + Эволюционная смена форм.* Соответственно особи именно такого типа будут иметь преобладающие шансы на выживание и, следовательно, на закрепление своего фенотипа в популяции. Таким образом, если условия внешней среды останутся неизменными достаточно долго и данный табун продолжит обитать в данной местности, то после нескольких поколений указанный тип станет доминирующим в данной популяции. В свою очередь, высокие длинноногие лошади, скорее всего, обречены на вымирание в изменившихся условиях обитания.

Теория не была принята, поскольку это применимо только к некоторым органам и, по факту, никакие характеристики не приобретаются, а только проявляются, уже существующие на протяжении очень долгого количества времени, черты, появившиеся случайной мутацией и которые, как раз, НЕ ПРОЯВЛЯЛИСЬ ИЗ-ЗА ОСОБЕННОСТЕЙ ОКРУЖАЮЩЕЙ СРЕДЫ.

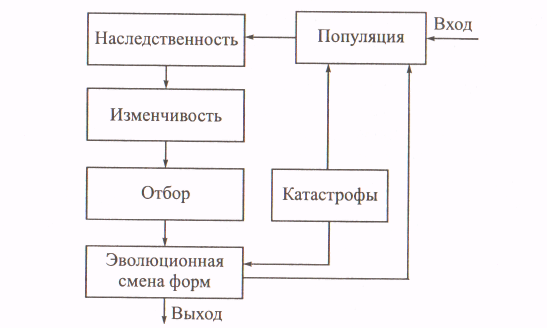
1. Модель эволюции Де Фриза

В ее основе лежит моделирование социальных и географических катастроф, приводящих к резкому изменению видов и популяций.

Эволюция, таким образом, представляет собой последовательность скачков в развитии популяции без предварительного накопления количественных изменений в эволюционных процессах.

Такой механизм эволюции иногда называют эволюцией катастроф. Он проявляется, ориентировочно, один раз в несколько тысяч поколений.

Основная идея его состоит во внесении глобальных изменений в генофонд на момент катастрофы.



Пусть существует известный нам табун лошадей из предыдущих примеров. В ходе эволюции данной популяции происходит постепенная смена поколений, причем происходящие изменения генотипа популяции носят регулярный постепенный характер.

Однако на некотором шаге эволюции данная популяция случайным образом подвергается катастрофическому воздействию внешней среды, которое приводит к значительному сокращению (или вымиранию) популяции и вызывает кардинальное изменение генотипа.

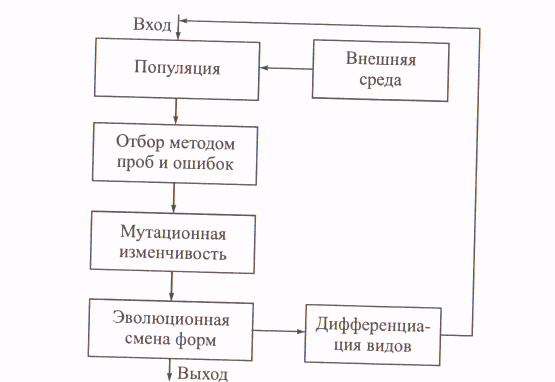
В нашем примере таким фактором может быть большая засуха, в результате которой были уничтожены кормовые угодья. Вследствие случившегося голода выживает лишь незначительное количество особей из рассматриваемого табуна.

Эти пережившие катастрофу особи составят впоследствии новую популяцию (табун) особей, с новыми качествами (рост, цвет и т. д.), которая заменит старую популяцию, и процесс эволюции будет продолжаться.

1. Модель эволюции Поппера

Это условная структура, реализующая иерархическую систему гибких механизмов управления, в которых мутация интерпретируется как метод случайных проб и ошибок, а отбор — как один из способов управления с помощью устранения ошибок при взаимодействии с внешней средой.

К. Поппер интерпретировал эволюцию Дарвина в виде триады: дедуктивизм - отбор –устранение ошибок.



Эволюция Поппера излагается в виде 12 тезисов. Основными из них являются:

* проблемы эволюции всегда решаются методом проб и ошибок;
* устранение ошибок может осуществляться либо путем полного  
  устранения неудачных форм, либо в виде эволюции механизмов управления;
* популяция использует тот механизм управления, который выработался в процессе эволюции;
* популяция является пробным решением, анализируемым в процессе эволюции, выбирающим окружающую среду и преобразующим ее;
* эволюционная последовательность событий представляется в виде последовательности *F\* —*> TS —> ЕЕ* —> *fz,*

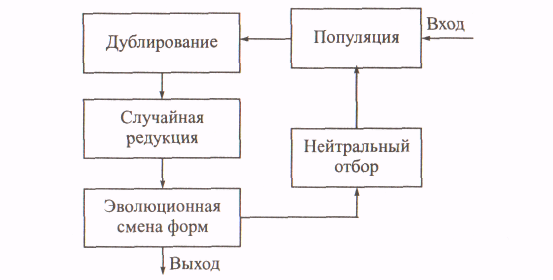
где *F\ —* исходная проблема, *TS —* пробные решения, *ЕЕ —* устранение ошибок, *fz —* новая проблема.

* В отличие от эволюции Дарвина, где существует одна проблема -«выживание сильнейших», в эволюции Поппера существуют и другие проблемы: воспроизводство, избавление от лишнего потомства и т. п.
* Согласно К. Попперу естественные системы исследуют окружающую среду и активно получают из нее информацию.
* Процесс выбора лучшей индивидуальности в данной эволюции может являться процессом отбора (селекции), а отбор из некоторого множества случайных событий не обязан быть случайным.

1. Модель нейтральной эволюции Кимуры

*М. Кимура* предложил модель нейтральной эволюции с нейтральным отбором. По его теории на генетическую изменчивость исходно влияют мутации, обуславливающие изменчивость и генетический дрейф, исключающий ее в отсутствии дифференцирующего отбора.

Теория нейтральности предполагает, что большая часть молекулярных вариантов имеет равную приспособленность друг относительно друга. Изменчивость здесь поддерживается балансирующим отбором.



Пусть задана популяция, состоящая из «больших» и «маленьких» особей. Тогда эволюция заключается в реализации последовательностей поколений.

Процесс реализации поколения состоит из двух шагов. На первом шаге дублируются все особи: большие имеют два больших потомка, маленькие — имеют два маленьких потомка.

На втором шаге из популяции случайным образом удаляется ровно половина особей с равной вероятностью для больших и маленьких.

Рассматриваемый процесс всегда сходится к одному из поглощающих состояний (все особи большие или все маленькие).

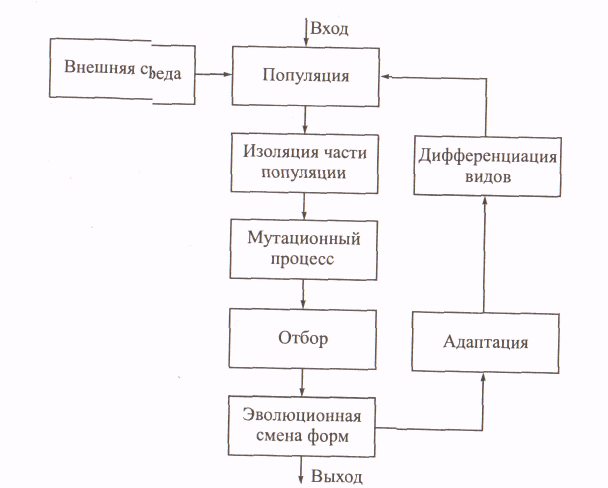
-----------------------------------

Механизм работы модели нейтральной эволюции Кимуры можно пояснить на примере знакомого нам табуна лошадей.

Пусть существует некий табун лошадей (популяция). В этом табуне имеется две большие группы особей (лошадей), отличающиеся друг от друга окраской (вороные и гнедые).

Тогда согласно данной модели в результате эволюции по прошествии некоторого определенного числа поколений в табуне останется только одна доминирующая группа особей (лошадей) одинаковой окраски.

1. Модель синтетической эволюции Дубинина



Отметим, что признание единства факторов эволюции в виде наследственности, изменчивости и естественного отбора не исключает существование разных форм эволюции.

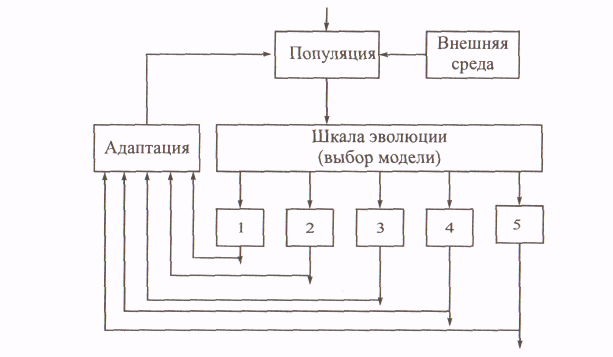
Н. Дубинин выделяет 4 основные формы осуществления внутренне единого эволюционного процесса:

* микроэволюция (процессы внутривидовой эволюции);
* эволюция на основе фазы нарастающего эволюционного усовершенствования;
* эволюция на основе переломных моментов;
* эволюция на основе интеграционных особенностей в организации естественных систем.

1. Условная упрощенная интегрированная схема эволюции

Эволюционный процесс связан с двумя типами адаптации. Один тип адаптации основан на выработке приспособлений к условиям внешней среды, в которых вид существует в настоящее время. Другой тип адаптации связан с выработкой таких особенностей в структуре, которые должны обеспечить его соревнование с другими видами во времени.

Основная задача синтетической теории эволюции — определение природы противоречий или постепенной эволюции, т.е. разных форм противоречий между наследственностью и постоянно меняющимися потребностями в приспособлениях.



Кардинальное положение синтетической теории эволюции — признание стохастичности процессов мутаций и больших резервов рекомбинационной изменчивости. Условия внешней среды — не только факторы исключения неприспособленных особей, но и особенности, формирующие синтетическую теорию эволюции.

В этой связи авторы считают важным объединение всех видов и моделей эволюции.

Блоки 1-5 соответствуют схемам моделей эволюции Дарвина, Ламарка, де Фриза, Поппера и Кимуры *соответственно.* Основным этапом в каждой модели эволюции является анализ популяции, ее преобразование тем или иным способом и эволюционная смена форм.

# 24 Генетические алгоритмы: основа, цель, отличия. Предварительные этапы ГА (4 этапа).

ОСНОВА

*Основой* для возникновения генетических алгоритмов послужили модель биологической эволюции и методы случайного поиска. Л. Растригин отмечал, что случайный поиск возник как реализация простейшей модели эволюции, когда случайные мутации моделировались случайными шагами оптимального решения, а отбор — «устранением» неудачных вариантов.

*Эволюционный поиск* с точки зрения преобразования информации — это последовательное преобразование одного конечного нечеткого множества промежуточных решений в другое.

Само преобразование можно назвать алгоритмом поиска, или генетическим алгоритмом. Генетические алгоритмы — это не просто случайный поиск. Они эффективно используют информацию, накопленную в процессе эволюции.

ЦЕЛЬ

Цель генетических алгоритмов состоит в том, чтобы:

* абстрактно и формально объяснять адаптацию процессов в естественной системе и интеллектуальной исследовательской системе;
* моделировать естественные эволюционные процессы для эффективного решения оптимизационных задач науки и техники.

В настоящее время используется новая парадигма решений оптимизационных задач на основе генетических алгоритмов и их различных модификаций. Генетические алгоритмы осуществляют поиск баланса между эффективностью и качеством решений за счет «выживания сильнейших альтернативных решений» в неопределенных и нечетких условиях.

ОТЛИЧИЯ

Генетические алгоритмы отличаются от других оптимизационных и поисковых процедур следующим:

* работают в основном не с параметрами задачи, а с закодированным множеством параметров;
* осуществляют поиск не путем улучшения одного решения, а путем использования сразу нескольких альтернатив на заданном множестве решений;
* используют *целевую функцию,* а не ее различные приращения для оценки качества принятия решений;
* применяют не детерминированные, а вероятностные правила анализа оптимизационных задач.

Для работы генетических алгоритмов выбирают множество натуральных параметров оптимизационной проблемы и кодируют их в последовательность конечной длины в некотором алфавите.

Они работают до тех пор, пока не будет выполнено заданное число генераций (итераций алгоритма) или на некоторой генерации будет получено решение определенного качества, или когда найден локальный оптимум, т. е. возникла преждевременная сходимость и алгоритм не может найти выход из этого состояния.

В отличие от других методов оптимизации эти алгоритмы, как правило, анализируют различные области пространства решений одновременно и поэтому они более приспособлены к нахождению новых областей с лучшими значениями целевой функции.

ПРЕДВАРИТЕЛЬНЫЕ ЭТАПЫ

При решении практических задач с использованием генетических алгоритмов обычно выполняют 4 предварительных этапа:

* выбор способа представления решения;
* разработка операторов случайных изменений;
* определение способов «выживания» решений;
* оздание начальной популяции альтернативных решений.

***1 этап***

*На первом этапе* для представления решения в формальном виде требуется такая структура, которая позволит кодировать любое возможное решение и производить его оценку.

Математически доказано, что не существует идеальной структуры представления, так что для создания хорошей структуры требуется анализ, перебор и эвристические подходы.

Возможный вариант представления должен позволять проведение различных перестановок в альтернативных решениях.

Для оценки решений необходимо определить способ вычисления целевой функции.

***2 этап***

*На втором этапе* достаточно сложным является выбор случайного оператора (или операторов) для генерации потомков. Существует огромное число таких операторов. Существуют два основных типа размножения: половое и бесполое. При половом размножении два родителя обмениваются генетическим материалом, который используется при создании потомка. Бесполое размножение — это фактически клонирование, при котором происходят различные мутации при передаче информации от родителя к потомку. Модели этих типов размножения играют важную роль в генетических алгоритмах.

В общем случае можно применить модели размножения, которые не существуют в природе. Например, использовать материал от трех или более родителей, проводить голосование при выборе родителей и т. п. При решении технических задач нет смысла слепо копировать законы природы и ограничиваться только ими.

Успех генетических алгоритмов во многом зависит от того, как взаимодействуют между собой схема представления, методы случайных изменений и способ определения целевой функции. Поэтому для определенного класса задач целесообразно использовать направленные методы.

В качестве примера рассмотрим два способа представления перестановок при решении оптимизационных задач. В первом случае будем использовать одного родителя (альтернативное решение) и получать одного потомка. Во втором случае используем двух родителей, случайно выберем точку перестановки и для образования потомка возьмем первый сегмент у первого родителя, а второй сегмент — у второго. Первый метод похож на бесполое размножение, а второй — на половое размножение. Стоит отметить, что если первый метод всегда генерирует реальное решение, то второй может генерировать недопустимые решения. При этом требуется «восстанавливать» допустимые решения перед их оценкой.

***3-4 этапы***

*На третьем из рассматриваемых этапов* задаются правила выживания решений для создания потомства. Существует множество способов проведения селекции альтернативных решений. Простейшее правило — это «выживание сильнейших», т. е. остаются только лучшие решения с точки зрения заданной целевой функции, а все остальные устраняются. Такое правило часто оказывается малоэффективным при решении сложных технических проблем. Иногда лучшие решения могут происходить от худших, а не только от самых лучших. Тем не менее, логично использовать принцип: «*Чем «лучше» решение, тем больше вероятность его выживания»*

*На последнем предварительном этапе* создается начальная популяция. При неполноте исходных данных о проблеме решения могут случайным образом выбираться из всего множества альтернатив. Это реализуется генерацией случайных внутрихромосомных перестановок, каждая из которых представляет собой определенное решение. При создании начальной популяции рекомендуется использовать знания о решаемой задаче. Например, эти знания могут быть получены из опыта разработчика, существующих стандартов и библиотек алгоритмов решения задач данного класса.

# 25 Эффективность ГА и формирование популяции (4 основных принципа)

***Эффективность генетического алгоритма*** *—* степень реализации запланированных действий алгоритма и достижение требуемых значений целевой функции. Эффективность во многом определяется структурой и составом начальной популяции. При создании начального множества решений происходит формирование популяции на основе четырех основных принципов:

* «одеяло» — генерируется полная популяция, включающая все возможные решения в некоторой заданной области;
* «дробовик» — подразумевает случайный выбор альтернатив из всей области решений данной задачи.
* «фокусировка» — реализует случайный выбор допустимых альтернатив из заданной области решений данной задачи.
* «комбинирование» — состоит в различных совместных реализациях первых трех принципов.

Отметим**, что популяция обязательно является конечным множеством.**

# 26 Основные генетические операторы (4).

1. Оператор репродукции (селекции)
2. Оператор кроссинговера
3. Оператор мутации
4. Оператор инверсии

**ОПЕРАТОР РЕПРОДУКЦИИ (СЕЛЕКЦИИ)**

***Оператор репродукции*** *(селекция) —* это процесс, посредством которого хромосомы (альтернативные решения), имеющие более высокое значение целевой функции (с «лучшими» признаками), получают большую возможность для воспроизводства (репродукции) потомков, чем «худшие» хромосомы. Элементы, выбранные для репродукции, обмениваются генетическим материалом, создавая аналогичных или различных потомков.

Существует большое число видов операторов репродукции. К ним относятся следующие.

***Селекция на основе рулетки***— это простой и широко используемый в простом генетическом алгоритме метод. При его реализации каждому элементу в популяции соответствует зона на колесе рулетки, пропорционально соразмерная с величиной целевой функции. Причем элемент с большим значением целевой функции имеет большую вероятность для выбора.

***Селекция на основе заданной шкалы****.* Здесь популяция предварительно сортируется от «лучшей» к «худшей» на основе заданного критерия. Каждому элементу назначается определенное число и тогда селекция выполняется согласно этому числу.

***Элитная селекция****.* В этом случае выбираются лучшие (элитные) элементы на основе сравнения значений целевой функции. Далее они вступают в различные преобразования, после которых снова выбираются элитные элементы. Процесс продолжается аналогично до тех пор, пока продолжают появляться элитные элементы.

***Турнирная селекция****.* При этом некоторое число элементов (согласно размеру «турнира») выбирается случайно или направленно из популяции, и лучшие элементы в этой группе на основе заданного турнира определяются для дальнейшего эволюционного поиска.

**ОПЕРАТОРЫ КРОССИНГОВЕРА (СКРЕЩИВАНИЯ).**

***Оператор кроссинговера*** *—* это языковая конструкция, позволяющая на основе преобразования (скрещивания) хромосом родителей (или их частей) создавать хромосомы потомков. Существует огромное число операторов кроссинговера, так как их структура в основном и определяет эффективность генетических алгоритмов.

***Простой (одноточечный) оператор кроссинговера****.* Перед началом работы одноточечного оператора кроссинговера определяется так называемая точка оператора кроссинговера, или разрезающая точка оператора кроссинговера, которая обычно определяется случайно. Эта точка определяет место в двух хромосомах, где они должны быть «разрезаны».

**ОДНОТОЧЕЧНЫЙ ОПЕРАТОР КРОССИНГОВЕРА**

Одноточечный оператор кроссинговера выполняется в три этапа:

1. Две хромосомы *А = а1,а2,...* , аL, и *В* = *а1’,а2’,*... *,aL’* выбираются  
случайно из текущей популяции.

2. Число *k* выбирается из {1,2, ...,L— 1} также случайно. Здесь L — длина хромосомы, *k —* точка оператора кроссинговера (номер, значение или код гена, после которого выполняется разрез хромосомы).

3. Две новые хромосомы формируются из А и В путем перестановок элементов согласно правилу:

**ДВУХТОЧЕЧНЫЙ ОПЕРАТОР КРОССИНГОВЕРА**

В каждой хромосоме определяются две точки оператора кроссинговера, и хромосомы обмениваются участками, расположенными между двумя точками оператора кроссинговера.

Точки оператора кроссинговера в двухточечном операторе кроссинговера также определяются случайно.

Существует большое количество модификаций двухточечного оператора кроссинговера.

Развитием двухточечного оператора кроссинговера является многоточечный или *N*-точечный оператор кроссинговера. Многоточечный оператор кроссинговера выполняется аналогично двухточечному, хотя большое число «разрезающих» точек может привести к потере «хороших» родительских свойств.

Выделяют также:

* упорядоченный оператор кроссинговера (ОК);
* частично-соответствующий ОК;
* циклический ОК;
* универсальный ОК;
* жадный ОК

**ТРЕХТОЧЕЧНЫЙ ОПЕРАТОР КРОССИНГОВЕРА**

Здесь точки оператора кроссинговера делят хромосому на ряд строительных блоков (в данном случае 4).

Потомок *Р1’* образуется из нечетных блоков родителя *Р1 (1 и 3)* и четных блоков родителя *Р2 (2 и 4).*

Потомок *Р2'* образуется соответственно из нечетных блоков родителя *Р2* и четных блоков родителя *P1.*

**ОПЕРАТОР МУТАЦИИ**

***Оператор мутации —*** языковая конструкция, позволяющая на основе преобразования родительской хромосомы (или ее части) создавать хромосому потомка.

Оператор мутации обычно состоит из двух этапов:

* 1. В хромосоме *А = (а1, а2, а3,* aL-2, aL-1, Al) определяются случайным образом две позиции (например, *а2* и aL-1).
* 2. Гены, соответствующие выбранным позициям, переставляются, и формируется новая хромосома

*А' = (а1, аL-1, a3, •••, аL-2, a2, aL)*

**ДВУХТОЧЕЧНЫЙ ОПЕРАТОР МУТАЦИИ**

* При реализации ***двухточечного оператора мутации***случайным или направленным образом выбираются две точки разреза. Затем производится перестановка генов между собой, расположенных справа от точек разреза.
* Развитием двухточечного оператора мутации является многоточечный (или n-точечный) оператор мутации. В этом случае происходит последовательный обмен генов, расположенных правее точек разреза друг с другом в порядке их расположения. Ген, расположенный правее последней точки разреза, переходит на место первого.

***ОПЕРАТОР ИНВЕРСИИ***

***Оператор инверсии –*** это языковая конструкция, позволяющая на основе инвертирования родительской хромосомы (или ее части) создавать хромосому потомка. При его реализации случайным образом определяется одна или несколько точек разреза (инверсии), внутри которых элементы инвер­тируются.

# 27 Дополнительные генетические операторы (7).

Транслокации

Транспозиции

Сегрегации

Удаления

Вставки

Редукции

Рекомбинации

***Оператор транслокации —*** это языковая конструкция, позволяющая на основе скрещивания и инвертирования из пары родительских хромосом (или их частей) создавать две хромосомы потомков. Другими словами, он представляет собой комбинацию операторов кроссинговера и инверсии.

***Оператор транспозиции* —** языковая конструкция, позволяющая на основе преобразования и инвертирования выделяемой части родительской хромосомы создавать хромосому потомка.

***Оператор сегрегации —*** *э*то языковая конструкция, позволяющая на основе выбора строительных блоков из хромосом родителей (или их частей) создавать хромосомы потомков**.**

***Оператор удаления —*** *э*то языковая конструкция,  
позволяющая на основе удаления строительных блоков из хромосом родителей (или их частей) создавать хромосомы потомков.

***Оператор вставки —*** языковая конструкция, позволяющая на основе вставки строительных блоков в хромосомы родителей создавать хромосомы потомков.

При его реализации направленным или случайным образом создается хромосома (донор), состоящая из строительных блоков, которые желательно разместить в другие хромосомы популяции.

После этого направленным или случайным образом определяется хромосома для реализации оператора вставки. В ней находится точка или точки разреза.

Затем анализируются другие хромосомы в популяции для определения альтернативных вставок.

Далее производится пробная вставка строительных блоков с вычислением изменения значения целевой функции и получением реальных решений. Новые строительные блоки вставляются в хромосому справа от точки оператора вставки или между его двумя точками.

Отметим, что оператор удаления и оператор вставки могут изменять размер хромосом. Для сохранения постоянного размера хромосом эти операторы можно применять совместно.

***Оператор редукции* —** языковая конструкция, позволяющая на основе анализа популяции после одной или нескольких поколений генетического алгоритма уменьшать ее размер до заданной величины.

Рассмотрим способы реализации оператора редукции. Он выполняется для устранения неудачных решений. В некоторых генетических алгоритмах, в частности, в простом генетическом алгоритме, этот оператор применяется для сохранения постоянного размера начальной популяции.

Основная проблема здесь — нахождение компромисса между разнообразием генетического материала и качеством решений.

Сначала формируют репродукционную группу из всех решений, образовавшихся в популяции *Nt*, затем выполняют отбор решений в следующую популяцию.

***Оператор рекомбинации* —** языковая конструкция, которая определяет, как новая генерация хромосом будет построена из родителей и потомков. Другими словами, оператор рекомбинации — это технология анализа и преобразования популяции при переходе из одной генерации в другую.

Существует много путей выполнения рекомбинации. Один из них состоит из перемещения родителей в потомки после реализации каждого генетического оператора.

Другой путь заключается в перемещении  
некоторой части популяции после каждой генерации.

# 28 Простой генетический алгоритм (ГА). Простые ГА Холланда, Гольдберга, Девиса – основные этапы.

Эволюционный процесс представляется как способность «лучших» хромосом оказывать большее влияние на состав новой популяции на основе длительного выживания из более многочисленного потомства. Основные этапы эволюционного поиска следующие.

ОСНОВНЫЕ ЭТАПЫ ЭВОЛЮЦИОННОГО ПОИСКА **(простой генетический алгоритм Холланда)**

1. Сконструировать начальную популяцию. Ввести точку отсчета по­  
колений t = 0. Вычислить приспособленность каждой хромосомы  
в популяции, а затем среднюю приспособленность всей популяции.

2. Установить t = t+l. Произвести выбор двух родителей (хромосом)  
для реализации оператора кроссинговера. Он выполняется случай­  
ным образом пропорционально приспособляемости родителей.

3. Сформировать генотип потомков. Для этого с заданной вероятно­  
стью выполнить оператор кроссинговера над генотипами выбран­  
ных хромосом. Далее с вероятностью 0,5 выбрать один из потомков Pi(t) и сохранить как член новой популяции. После этого к Pi(t) последовательно применить оператор инверсии, а затем — оператор мутации с заданными вероятностями. Полученный генотип потомка сохранить как Pk(t).

4. Определить количество хромосом для исключения их из популя­  
ции, чтобы ее размер оставался постоянным. Текущую популяцию  
обновить заменой отобранных хромосом на потомков flt(t).

5. Произвести определение приспособленности (целевой функции) и  
пересчет средней приспособленности всей полученной популяции.

6. Если t = Заданному, то перейти к 7, если нет, то перейти к 2.

7. Конец работы.

Данный алгоритм известен как упрощенный «репродуктивный план Д.Холланда». Заметим, что в практических задачах вместо понятия «приспособленность» используют понятие «целевая функция».

**ПРОСТОЙ ГЕНЕТИЧЕСКИЙ АЛГОРИТМ Гольдберга**

*Простой генетический алгоритм* был впервые описан  
Д. Гольдбергом на основе работ Д.Холланда. Его механизм несложен.

Предварительно простой генетический алгоритм случайно генерирует  
популяцию последовательностей —хромосом (альтернативных упорядоченных и неупорядоченных решений). Затем производится копирование последовательности хромосом и перестановка их частей. Далее простой генетический алгоритм реализует множество простых операций к начальной популяции и генерирует новые решения.

Простой генетический алгоритм состоит из трех операторов:

* репродукции;
* кроссинговера;
* мутации.

***Репродукция***— процесс, в котором хромосомы копируются пропорционально значению их целевой функции. Копирование хромосом с «лучшим» значением целевой функции имеет большую вероятность для попадания в следующую генерацию. Рассматривая эволюцию Дарвина, можно отметить, что оператор репродукции является искусственной версией натуральной селекции — «выживание сильнейших». Он представляется в алгоритмической форме различными способами. Самый простой — создать модель «колеса рулетки», в которой каждая хромосома имеет поле, пропорциональное значению целевой функции.

**ОПЕРАТОР КРОССИНГОВЕРА**

На основе реализации оператора репродукции выбираются хромосомы для применения оператора кроссинговера.

Оператор кроссинговера, как правило, выполняется в 3 шага, одним из операторов кроссинговера, описанным выше. Точка разрыва *k* выбирается случайно между 1 и числом, равным длине хромосомы минус единица, т. е. в интервале (1, *L* — 1). Длина хромосомы *L —* это число значащих цифр в ее коде.

Применяя к популяции, полученной после реализации оператора репродукции (столбец 2 табл. 3), оператор кроссинговера, получим новую популяцию хромосом (5-й столбец таблицы 3). В принципе оператор кроссинговера можно применять любое число раз. После проведения одной генерации простого генетического алгоритма улучшились все показатели: среднее и максимальное значение целевой функции.

**ОПЕРАТОР МУТАЦИИ**

Далее, согласно схеме выполнения простого генетического алгоритма, реализуется оператор мутации. Существует большое количество видов операторов мутации. Эти операторы соответствуют перестановкам элементов внутри заданного множества. Очевидно, что при небольшой длине хромосомы L (порядка 10-20) можно выполнить полный перебор за приемлемое время и найти наилучшие решения.

При увеличении L до 50-200 и выше полный перебор произвести затруднительно и необходимы другие механизмы поиска. Здесь как раз и приходит на помощь направленно-случайный поиск, который реализуется на основе простого генетического алгоритма.

**ДРУГОЙ СТАНДАРТНЫЙ ТИП ГЕНЕТИЧЕСКОГО АЛГОРИТМА, ОПИСАННЫЙ Л.ДЕВИСОМ**

1. Инициализировать популяции хромосом.

2. Оценить значения каждой хромосомы в популяции.

3. Создать новые хромосомы посредством скрещивания текущих хро­  
мосом; применить операторы мутации и рекомбинации.

4. Устранить хромосомы из популяции, чтобы освободить место для  
новых хромосом.

5. Оценить значения новых хромосом и вставить их в популяцию.

6. Если время, заданное на реализацию алгоритма, закончено, то  
остановиться и возвратиться к наилучшей хромосоме; если нет, то  
перейти к 3.

7. Конец работы алгоритма.

Сравнивая описания простых генетических алгоритмов Д. Голдберга, Д. Холланда и Л.Девиса, видим, что в них реализована одна основная идея моделирования эволюции с некоторыми модификациями. Однако заметим, что эти изменения могут существенно влиять на окончательное качество решения.

# 29 Простой модифицированный ГА –отличие и основные этапы.

**МОДИФИЦИРОВАННЫЙ ПРОСТОЙ ГЕНЕТИЧЕСКИЙ АЛГОРИТМ**

1. Создать начальную популяцию решений.

2. Смоделировать популяцию (определить ЦФ (целевую функцию)  
для каждой хромосомы).

3. Если еще не проведено необходимое число генераций или не закон­  
чилось время, заданное на реализацию алгоритма, или не найдено  
оптимальное значение целевой функции (если оно известно):

а) выбрать элементы для репродукции;  
Применить:

б) оператор кроссинговера для создания потомков;

в) оператор мутации;

г) оператор инверсии;

д) оператор транспозиции;

е) оператор транслокации;

ж) оператор сегрегации;

з) оператор удаления вершин;  
и) оператор вставки вершин;

к) рекомбинацию родителей и потомков для создания новой генерации;

л) оператор редукции.

4. Реализовать новую генерацию.

Генетические алгоритмы отличаются от других оптимизационных и поисковых процедур следующим:

* работают в основном не с параметрами задачи, а с закодированным множеством параметров;
* осуществляют поиск не путем улучшения одного решения, а путем использования сразу нескольких альтернатив на заданном множестве решений;
* используют *целевую функцию,* а не ее различные приращения для оценки качества принятия решений;
* применяют не детерминированные, а вероятностные правила анализа оптимизационных задач.
* Для работы генетических алгоритмов выбирают множество натуральных параметров оптимизационной проблемы и кодируют их в последовательность конечной длины в некотором алфавите.
* Они работают до тех пор, пока не будет выполнено заданное число генераций (итераций алгоритма) или на некоторой генерации будет получено решение определенного качества, или когда найден локальный оптимум, т. е. возникла преждевременная сходимость и алгоритм не может найти выход из этого состояния.
* В отличие от других методов оптимизации эти алгоритмы, как правило, анализируют различные области пространства решений одновременно и поэтому они более приспособлены к нахождению новых областей с лучшими значениями целевой функции.

# 30 Базовая структура генетического алгоритма с использованием эволюционной адаптации.



***Простой генетический алгоритм***случайно генерирует популяцию хромосом (альтернативных упорядоченных и неупорядоченных  
решений). Затем производится копирование хромосом, перестановка  
их частей и генерация новых хромосом (решений) на основе трех операторов: репродукции, кроссинговера и мутации.

***Блок эволюционной адаптации*** – специальный блок, который на основе обратных связей управляет процессом эволюционного поиска. Согласно данной схеме на первом этапе случайным, направленным или комбинированным методом получают некоторое подмножество решений рассматриваемой задачи. Эти решения образуют текущую генерацию или популяцию исследуемых решений на шаге

t (t = 0,1, ..., Т).

Далее вводится или вычисляется значение целевой функции. Вычисление целевой функции является сложной задачей, причем от точности значений целевой функции зависит качество будущих решений.

Качество — степень, с которой совокупность присущих решению задачи характеристик удовлетворяет заданным требованиям.

Отметим, что для каждой оптимизационной задачи желательно строить новую целевую функцию. При построении целевой функции необходимо использовать знания о конкретной задаче

На основе целевой функции производятся ранжирование и сортировка популяции решений.

Затем в результате различных методов селекции в популяции подбираются родительские хромосомы для применения различных генетических операторов.

После реализации всех операторов получается новое подмножество решений Р'. Оно объединяется с первоначальным подмножеством решений.

Получается новое множество РГА = PUP'.

Используя значение целевой функции, производится анализ РГА. Все элементы в РГА (решения задачи), значения целевой функции которых хуже заданного порога, являются с нашей точки зрения неперспективными решениями и удаляются из РГА.

Получается новое множество Р‘ГА, причем | Р‘ГА | = |Р|. Если данное условие не выполняется, например, |Р‘ГА | <|Р|, то в Р‘ГА включается элемент с лучшими характеристиками из отброшенных. Множество Р‘ГА объявляется новой текущей популяцией решений и далее процесс может повторяться на основе блока **эволюционной адаптации** итерационно до получения подмножества или одного оптимального решения.

# 31 Генетические алгоритмы, использующие идеи построения фракталов (множество Кантора, снежинка Коха, ковер Серпинского).

В 1980 г. Б. Мандельброт указал на фрактальную геометрию природы. Согласно Б. Мандельброту, эволюционирующие системы имеют фрактальную природу и наличие направленного нестихийного отбора, и самосогласованную эволюцию.

*Фрактальные объекты* самоподобны, т.е. их вид не претерпевает существенных изменений при изменении масштабов их деятельности. Множества, имеющие такую структуру, считаются обладающими геометрической (масштабной) универсальностью.

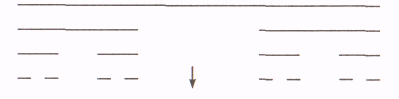
Преобразования, создающие такие структуры, — это процессы с обратной связью с большим числом итераций, когда одна и та же операция выполняется снова и снова, аналогично эволюционным процессам. Здесь результат одной итерации является начальным условием для другой и требуется нелинейная зависимость между результатом и реальным значением.

Такие множества объектов называются *фрактальными множествами.* Основными примерами таких множеств являются множество Кантора и ковер Серпинского. Эти множества обладают геометрической инвариантностью и называются «множества средних третей».

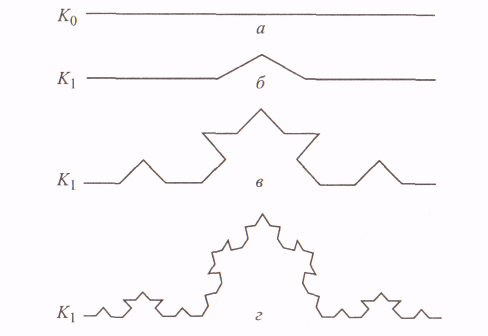
**МНОЖЕСТВО КАНТОРА**

* Рассмотрим простой вариант построения *множества Кантора.* Пусть на вещественной оси задан отрезок единичной длины [0,1]. Он делится на три равные части и средняя часть является открытым интервалом (1/3, 2/3 вырезается), как показано на рисунке.
* Аналогичные действия выполняются с каждым из оставшихся отрезков. Получаем последовательность отрезков убывающей длины. На первом этапе — это один отрезок, на втором — два, на третьем — 4 и т.д., на *k-м — 2k.* При *k —*> оо, имеем множество точек, называемое множеством Кантора. Суммарная длина всех вырезанных отрезков равна 1.
* Множество Кантора, промежуточное между точками *d = 0* и *d =* 1, является *фракталом.*

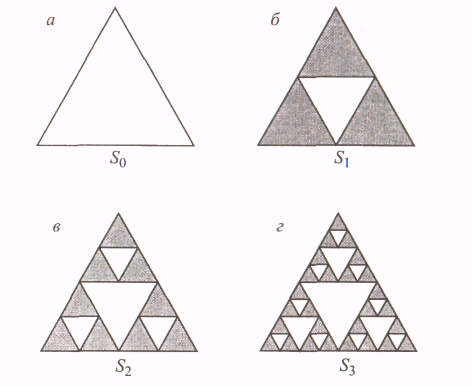
**ПРИМЕР ПОСТРОЕНИЯ МНОЖЕСТВА КАНТОРА**



**ПРОЦЕСС ПОСТРОЕНИЯ МНОЖЕСТВА КАНТОРА «СНЕЖИНКИ» (КРИВАЯ КОХА)**

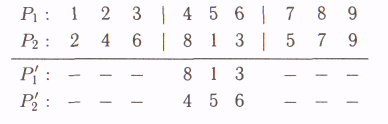


**ОБОБЩЕНИЕ МНОЖЕСТВА КАНТОРА НА СЛУЧАЙ ПЛОСКИХ ФИГУР ПРИВОДИТ К *«КОВРУ СЕРПИНСКОГО».***



**ОПЕРАТОР КРОССИНГОВЕРА НА ОСНОВЕ ИДЕЙ ПОСТРОЕНИЯ МНОЖЕСТВА КАНТОРА**

* Пусть заданы две родительские хромосомы *Р1, Р2.* Согласно идее построения множества Кантора в *Р1, Р2* определим две точки разрыва, которые делят хромосому на три равные части (1/3 + 1/3 + 1/3). Если при делении длины хромосомы на 3 ответ не является целым, то берутся ближайшие целые, например, при *L = 7* получим три части *L1=2, L2=2, Lз = 3 (L1 + L2 + Lз* *= 7).*
* Вторая часть хромосомы *Р2* помещается в хромосому- потомок *Р’1 ,* а вторая часть хромосомы *Р1* помещается в хромосому- потомок *Р’2.*
* Дальнейшее заполнение потомка *Р’1* выполняется из *Р1* слева направо, исключая повторяющиеся и вырезанные гены. Пустые позиции заполняются генами из *Р2* *.* Аналогичная процедура выполняется и для построения потомка *Р’2 .*



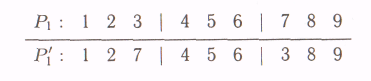
**ОПЕРАТОР МУТАЦИИ НА ОСНОВЕ ИДЕЙ ПОСТРОЕНИЯ МНОЖЕСТВА КАНТОРА**

Он заключается в перестановке генов, находящихся за точками разреза.

Например, пусть задана родительская хромосома *Р1.*

Определим две точки разрыва в *Р1,* которые делят хромосому на три равные или близкие к ним части.

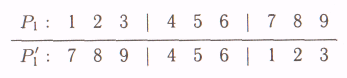
**ПРИМЕР ОПЕРАТОРА МУТАЦИИ НА ОСНОВЕ ИДЕЙ ПОСТРОЕНИЯ МНОЖЕСТВА КАНТОРА**



**МОДИФИКАЦИЯ ОПЕРАТОРА МУТАЦИИ МНОЖЕСТВА КАНТОРА**

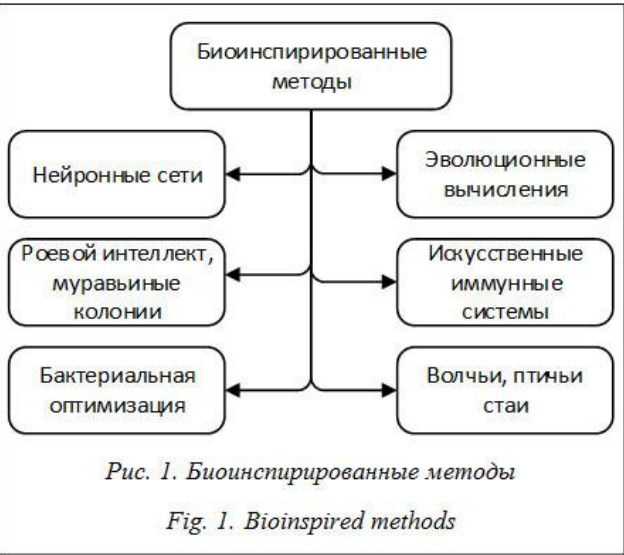
* Модификацией оператора мутации множества Кантора является процедура, когда первая часть родительской хромосомы меняется с третьей

**ПРИМЕР МОДИФИКАЦИИ ОПЕРАТОРА МУТАЦИИ МНОЖЕСТВА КАНТОРА**



* На основе множества Кантора и ковра Серпинского можно строить любой генетический оператор.
* При использовании последовательного поиска выполняется перебор точек разрыва для нахождения хромосомы с оптимальным значением целевой функции.

# 32 Классификация биоинспирированных методов. Пример роевого и стаевого интеллекта.



1. Нейронные сети – это системы, построенные по модели человеческого мозга;
2. эволюционные вычисления – алгоритмы, действующие по принципу генетических мутаций и эволюционного развития;
3. роевой интеллект, муравьиные колонии и прочие – многоагентные системы, основанные на поведении природных экосистем насекомых;
4. бактериальная оптимизация – семейство алгоритмов, отражающих поведение бактерий;
5. волчьи и птичьи стаи – алгоритмы, повторяющие поведение стайных животных.
6. Одним из актуальных классов биоинспирированных алгоритмов в современных исследованиях являются иммунные системы

Методы искусственных иммунных систем (ИИС), ориентированные на решение задачи глобальной оптимизации, основаны на некоторых аспектах поведения иммунной системы человека в процессе защиты ею организма. Защитные клетки иммунной системы (антитела) претерпевают при этом множество изменений, целью которых является создание клеток, обеспечивающих наилучшую защиту. ИИС обладает основными свойствами искусственного интеллекта: памятью, способностью к обучению и принятию решений в незнакомой

СТАЕВЫЙ АЛГОРИТМ (ПОИСК КОСЯКОМ РЫБ)  
Поиск косяком рыб или поиск агрегацией рыб – это оптимизационный алгоритм, основанный на поведении стаи рыб. Многие виды рыб проявляют так называемое стадное поведение, направленное в основном на повышение их выживаемости. С одной стороны, группировка рыб в стае нужна для защиты от преследования хищниками, а, с другой стороны ‒ как средство достижения коллективной цели, то есть поиска пищи. Областью допустимых решений является аквариум, при этом положение рыбы в нем отражает текущий вектор решений. Этапами алгоритма здесь являются операторы, применяемые ко всей популяции. Основные операторы были взяты из поведения агрегации рыб в живой природе и представляют собой кормление и плавание. Такая характеристика, как кормление, вдохновлена естественным инстинктом рыб к поиску пищи. От этого процесса зависит такой параметр алгоритма, как вес рыбы. Вес рыбы рассчитывается как разница между значением целевой функции на текущем и предыдущем шаге. Процесс кормления в алгоритме является аналогичным процессу оценки решений в оптимизации целевой функции. Этап плавания направлен на подражание коллективному движению, производимому каждой рыбой в агрегации. Процесс плавания управляется кормлением, то есть зависит от веса рыб, и в конечном счете будет направлять процесс поиска с целью получения оптимальных позиций рыб.

# **33 История возникновения направления искусственных нейронных сетей (ИНС): два периода, кибернетический и нейронный подходы**.

* **Первый период (1942-1969)**
* 1942-1943 г. – Уоррен Маккалок (нейрофизиолог) и Уолтер Питтс (математик, сотрудник Н. Винера) совместно разрабатывают теорию деятельности головного мозга.
* В 1943 году вышла их работа "Логическое исчисление идей, относящихся к нервной деятельности", в которой была построена модель нейрона, и сформулированы принципы построения искусственных нейронных сетей.
* **Второй период (1982-по н.в.)**
* [1982](https://ru.wikipedia.org/wiki/1982_%D0%B3%D0%BE%D0%B4_%D0%B2_%D0%BD%D0%B0%D1%83%D0%BA%D0%B5) — [Дж. Хопфилд](https://ru.wikipedia.org/wiki/%D0%A5%D0%BE%D0%BF%D1%84%D0%B8%D0%BB%D0%B4,_%D0%94%D0%B6%D0%BE%D0%BD) показал, что нейронная сеть с обратными связями может представлять собой систему, минимизирующую энергию ([сеть Хопфилда](https://ru.wikipedia.org/wiki/%D0%9D%D0%B5%D0%B9%D1%80%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D0%B5%D1%82%D1%8C_%D0%A5%D0%BE%D0%BF%D1%84%D0%B8%D0%BB%D0%B4%D0%B0)). Кохоненом представлены модели сети, обучающейся без учителя ([нейронная сеть Кохонена](https://ru.wikipedia.org/wiki/%D0%9D%D0%B5%D0%B9%D1%80%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D0%B5%D1%82%D1%8C_%D0%9A%D0%BE%D1%85%D0%BE%D0%BD%D0%B5%D0%BD%D0%B0)), решающей задачи [кластеризации](https://ru.wikipedia.org/wiki/%D0%9A%D0%BB%D0%B0%D1%81%D1%82%D0%B5%D1%80%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F), визуализации данных ([самоорганизующаяся карта Кохонена](https://ru.wikipedia.org/wiki/%D0%A1%D0%B0%D0%BC%D0%BE%D0%BE%D1%80%D0%B3%D0%B0%D0%BD%D0%B8%D0%B7%D1%83%D1%8E%D1%89%D0%B0%D1%8F%D1%81%D1%8F_%D0%BA%D0%B0%D1%80%D1%82%D0%B0_%D0%9A%D0%BE%D1%85%D0%BE%D0%BD%D0%B5%D0%BD%D0%B0)) и другие задачи предварительного анализа данных.
* [1986](https://ru.wikipedia.org/wiki/1986_%D0%B3%D0%BE%D0%B4_%D0%B2_%D0%BD%D0%B0%D1%83%D0%BA%D0%B5) — [Дэвидом И. Румельхартом](https://ru.wikipedia.org/wiki/%D0%A0%D1%83%D0%BC%D0%B5%D0%BB%D1%8C%D1%85%D0%B0%D1%80%D1%82,_%D0%94%D1%8D%D0%B2%D0%B8%D0%B4), [Дж. Е. Хинтоном](https://ru.wikipedia.org/wiki/%D0%A5%D0%B8%D0%BD%D1%82%D0%BE%D0%BD,_%D0%94%D0%B6%D0%B5%D1%84%D1%84%D1%80%D0%B8) и Рональдом Дж. Вильямсом[[16]](https://ru.wikipedia.org/wiki/%D0%9D%D0%B5%D0%B9%D1%80%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D0%B5%D1%82%D1%8C#cite_note-%D0%B0%D0%B2%D1%82%D0%BE%D1%81%D1%81%D1%8B%D0%BB%D0%BA%D0%B01-17), а также независимо и одновременно С. И. Барцевым и В. А. Охониным[[17]](https://ru.wikipedia.org/wiki/%D0%9D%D0%B5%D0%B9%D1%80%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D0%B5%D1%82%D1%8C" \l "cite_note-18), переоткрыт и развит [метод обратного распространения ошибки](https://ru.wikipedia.org/wiki/%D0%9C%D0%B5%D1%82%D0%BE%D0%B4_%D0%BE%D0%B1%D1%80%D0%B0%D1%82%D0%BD%D0%BE%D0%B3%D0%BE_%D1%80%D0%B0%D1%81%D0%BF%D1%80%D0%BE%D1%81%D1%82%D1%80%D0%B0%D0%BD%D0%B5%D0%BD%D0%B8%D1%8F_%D0%BE%D1%88%D0%B8%D0%B1%D0%BA%D0%B8).
* [2007](https://ru.wikipedia.org/wiki/2007_%D0%B3%D0%BE%D0%B4_%D0%B2_%D0%BD%D0%B0%D1%83%D0%BA%D0%B5) — [Джеффри Хинтоном](https://ru.wikipedia.org/wiki/%D0%A5%D0%B8%D0%BD%D1%82%D0%BE%D0%BD,_%D0%94%D0%B6%D0%B5%D1%84%D1%84%D1%80%D0%B8) в университете Торонто созданы алгоритмы глубокого обучения многослойных нейронных сетей. Хинтон при обучении нижних слоёв сети использовал [ограниченную машину Больцмана](https://ru.wikipedia.org/wiki/%D0%9C%D0%B0%D1%88%D0%B8%D0%BD%D0%B0_%D0%91%D0%BE%D0%BB%D1%8C%D1%86%D0%BC%D0%B0%D0%BD%D0%B0) (RBM — Restricted Boltzmann Machine). По Хинтону необходимо использовать много примеров распознаваемых образов (например, множество лиц людей на разных фонах). После обучения получается готовое быстро работающее приложение, способное решать конкретную задачу (например, осуществлять поиск лиц на изображении).

**КИБЕРНЕТИЧЕСКИЙ ПОДХОД**

* Попытки построить машины, способные к разумному поведению, в значительной мере вдохновлены идеями профессора МТИ Норберта Винера, одной из выдающихся личностей в интеллектуальной истории Америки.
* Помимо математики он обладал широкими познаниями в других областях, включая нейропсихологию, медицину, физику и электронику.
* Винер был убежден, что наиболее перспективны научные исследования в так называемых пограничных областях, которые нельзя конкретно отнести к той или иной конкретной дисциплине. Они лежат где-то на стыке наук, поэтому к ним обычно не подходят столь строго.
* Если затруднения в решении какой-либо проблемы психологии имеют математический характер, пояснял он, - то десять несведущих в математике психологов продвинутся не дальше одного столь же несведущего".
* Винеру и его сотруднику Джулиану Бигелоу принадлежит разработка принципа “обратной связи”, который заключается в использовании информации, поступающей из окружающего мира, для изменения поведения машины.
* В дальнейшем Винер (1958) разработал на принципе обратной связи теории как машинного, так и человеческого разума.  Он доказывал, что именно благодаря обратной связи все живое приспосабливается к окружающей среде и добивается своих целей.

**НЕЙРОННЫЙ ПОДХОД**

* Уоррен Маккалок, нейрофизиолог, изучавший философию и психологию в Йельском университете, а в то время работавший руководителем лаборатории фундаментальных исследований факультета психиатрии в Иллинойском университете, будучи на научной конференции в Нью-Йорке, услышал доклад одного из сотрудников Винера о механизмах обратной связи в биологии.
* Высказанные в докладе представления об обратной связи в чем-то перекликались с собственными идеями Маккалока относительно работы человеческого мозга.
* В течение следующего года Маккалок в соавторстве Уолтером Питтсом, который и был тем сотрудником, разработал теорию деятельности головного мозга.
* Эта теория и явилась той основой, на которой сформировалось широко распространенное мнение, что функции компьютеров и мозга в значительной мере сходны.
* Гипотеза: Нейроны можно упрощенно рассматривать как устройства, оперирующие двоичными числами.

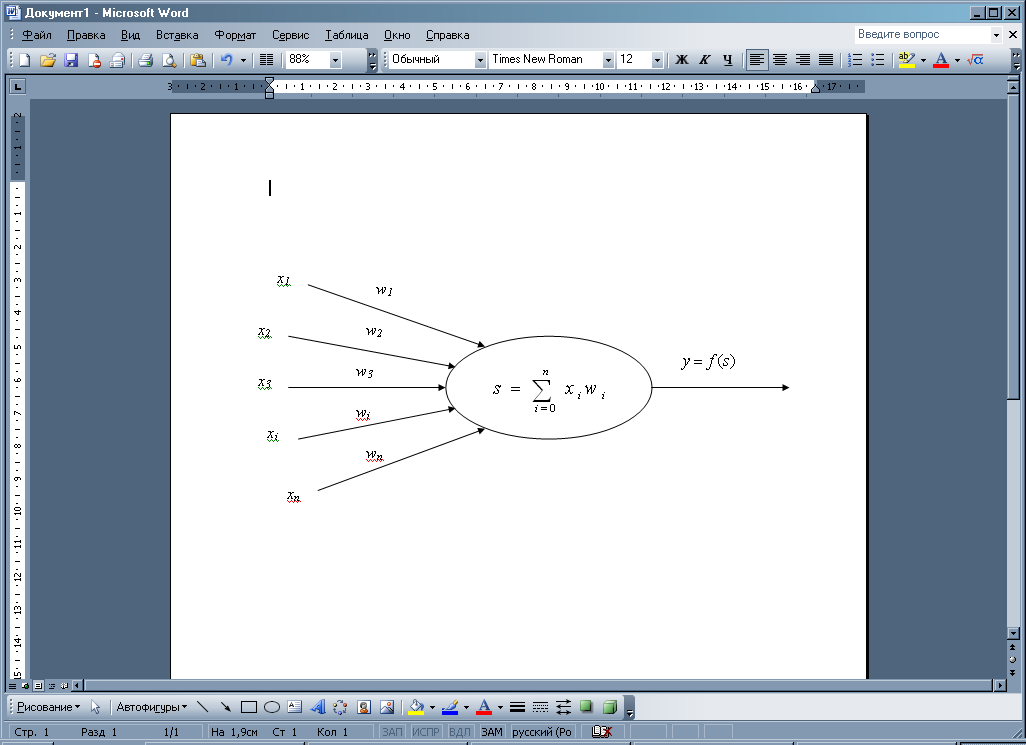
# 34 Модель искусственного нейрона (Маккалока-Питтса).

Первая вычислительная модель нейрона была предложена Уорреном Му Каллохом (нейробиологом) и Уолтером Питтсом (логиком) в 1943 году.

**МОДЕЛЬ НЕЙРОНА МАККАЛОКА-ПИТТСА**

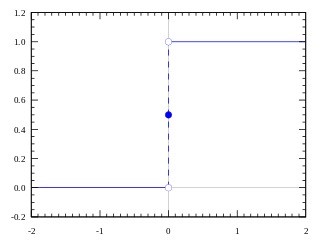


*v* - выход нейрона; *xi* – i-ый входной сигнал; *wi* - вес i-ой связи; *q* -пороговое значение; *n* - количество входных сигналов.



# 35 Активационная функция. Виды активационных функций.

**Функция активации** определяет выходное значение нейрона в зависимости от результата взвешенной суммы входов и порогового значения.

1. Функция принимает значение *1* (активирована), когда *Y > 0* (граница), и значение 0 (не активирована) в противном случае. Такая функция называется **ступенчатой** (простая пороговая функция).
2. Функция, представляющая выход регулирующего элемента - **пороговая** (threshold function) – первая активационная функция нейрона. 
3. **Логистическая**



1. **Гиперболический тангенс**



1. **Альтернативные функции**





Дополнительно можно прочитать: <https://studfile.net/preview/16436587/page:6/>, <https://studfile.net/preview/16436587/page:7/>, <https://neural.radkopeter.ru/chapter/основы-инс/#Функция-активации>

# 36 Выбор параметров ИНС. Переобученная сеть. Самоорганизующиеся карты Кохонена.

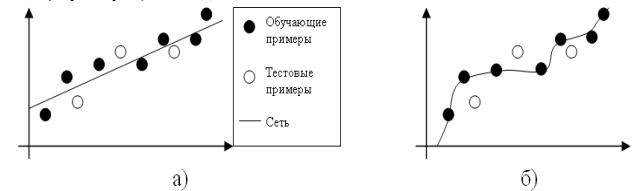
**ВЫБОР ПАРАМЕТРОВ СЕТИ**

В настоящее время нет никаких жестких правил ни для выбора количества скрытых слоев, ни для выбора количества нейронов в них. Хотя существуют ограничения, помогающие принимать решения.

1. Если функция определена на конечном множестве точек, то 3-ехслойный перцептрон способен ее апроксимировать.
2. Если функция непрерывна и определена на компактной области, то 3-ехслойный перцептрон способен ее апроксимировать.
3. Остальные функции, которым могут быть обучены нейронные сети, могут быть апроксимированы 4-ехслойным перцетроном.

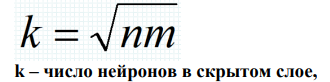
**КОЛИЧЕСТВО НЕЙРОНОВ В СКРЫТЫХ СЛОЯХ**

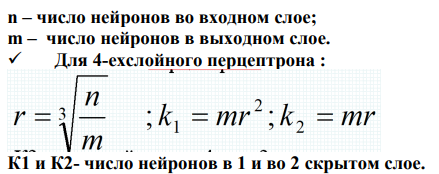
Слишком большое количество может привести к переобученности сети (overfitting), проявляющейся в том, что сеть будет прекрасно работать на обучающей выборке, но очень плохо на входных (тестовых) примерах, не входящих в нее:



**ПРАВИЛО ГЕОМЕТРИЧЕСКОЙ ПИРАМИДЫ**

Существуют эвристические правила выбора количества нейронов в скрытых слоях. Одним из таких правил является правило геометрической пирамиды (geometric pyramid rule) ⎫ число нейронов скрытого слоя в 3-ехслойном перцептроне вычисляется по следующей формуле:



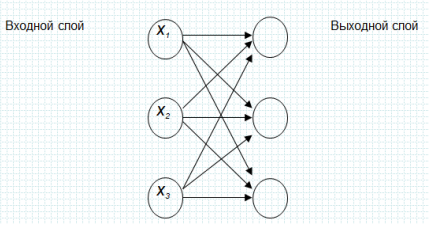


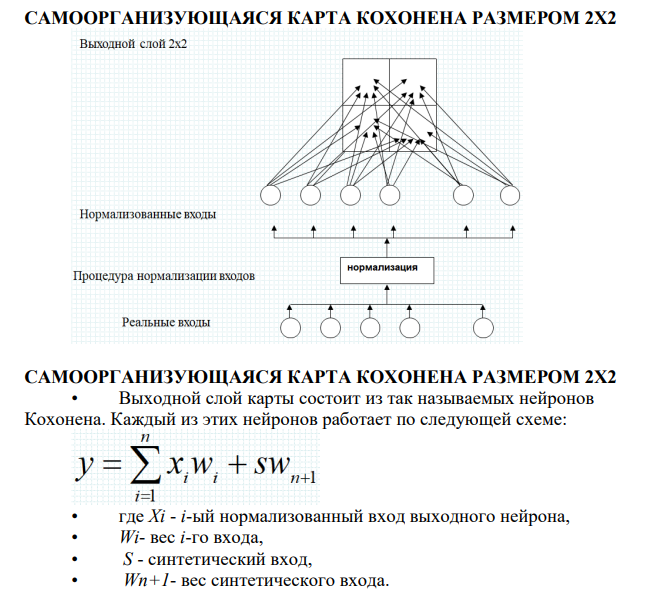
**САМООРГАНИЗУЮЩИЕСЯ КАРТЫ КОХОНЕНА**

Архитектура этой сети была создана финским исследователем Тео Кохоненом.

Далее представлена обобщенная архитектура самоорганизующейся карты Кохонена (self-organizing map - SOM) – это двухслойная сеть, хотя иногда ее называют трехслойной, из-за того, что входы подвергаются нормализации

ОБОБЩЕННАЯ АРХИТЕКТУРА САМООРГАНИЗУЮЩЕЙСЯ КАРТЫ КОХОНЕНА





# 37 Машинное обучение (МО). Место МО на основе ИНС в области ИИ.

Цель машинного обучения — предсказать результат по входным данным. Чем разнообразнее входные данные, тем проще машине найти закономерности и тем точнее результат.

Хотим определять спам — нужны примеры спам-писем, предсказывать курс акций — нужна история цен, узнать интересы пользователя — нужны его лайки или посты. Данных нужно как можно больше. Десятки тысяч примеров — это самый злой минимум для отчаянных.

Признаки:

фичи (features),

Фичи, свойства, характеристики, признаки — ими могут быть пробег автомобиля, пол пользователя, цена акций, даже счетчик частоты появления слова в тексте может быть фичей.

Машина должна знать, на что ей конкретно смотреть. Хорошо, когда данные просто лежат в табличках — названия их колонок и есть фичи. А если у нас сто гигабайт картинок с котами? Когда признаков много, модель работает медленно и неэффективно. Зачастую отбор правильных фич занимает больше времени, чем всё остальное обучение. Но бывают и обратные ситуации, когда человек сам решает отобрать только «правильные» на его взгляд признаки и вносит в модель субъективность — она начинает дико врать.

Искусственный интеллект — название всей области, как биология или химия.

• Машинное обучение — это раздел искусственного интеллекта. Важный, но не единственный. •

Нейросети — один из видов машинного обучения. Популярный, но есть и другие, не хуже. •

Глубокое обучение — архитектура нейросетей, один из подходов к их построению и обучению. На практике сегодня мало кто отличает, где глубокие нейросети, а где не очень. Говорят название конкретной сети и всё.

