Дискретная математика

Что есть истина ?

- **To them, I said, the truth would be literally**

noting but the shadows of the images.

-

О КУРСЕ Д. Д.

- Предмет курса. Приложения.
- Предмет – все дискретное, то есть счетные множества, графы, булевские функции, алгоритмы и т. д.
- Приложения – все компьютерное

ПЛАН.

- Часть 1 Введение Дискретное и непрерывное

- Часть 2 ГРАФЫ Рост графов Мир как сеть
- Часть 3 Булевские функции
- Часть 4 **Краткое введение в теорию алгоритмов. Жадные алгоритмы. Градиентный спуск. Динамическое программирование.**

ЧАСТЬ 1 Содержание части 1

- Дискретные и непрерывные объекты в математике.
- Целые рациональные и вещественные (действительные) числа. Открытие иррациональных чисел в Др. Греции.
- Счетные и несчетные множества.
- Теорема Кантора – множество иррациональных чисел несчетно.
- Трудности непрерывной математики и аксиоматическая теория. **Конструктивная математика и дискретная математика.**

Литература

- *Гаврилов Г. П., Сапоженко А. А.* Сборник задач по дискретной математике. — М.: Наука, 1969.
- *Кузнецов О. П., Адельсон-Вельский Г. М.* Дискретная математика для инженера. — М.: «Энергия», 1980. — 344 с.
- *Марченков С. С.* Замкнутые классы булевых функций. — М.: Физматлит, 2000.
- *Яблонский С. В.* Введение в дискретную математику. — М.: Наука, 1986.
- Алексеев В. Б. Дискретная математика (курс лекций, II семестр). Сост. А. Д. Поспелов

Литература

- **Основная книга по нейронным сетям-**
- ***Саймон Хайкин* Нейронные сети: полный курс = Neural Networks: A Comprehensive Foundation.** — 2-е. — М.: «Вильямс», 2006. — С. 1104. — ISBN 0-13-273350-1

Математика не наука

- Математика - это некий формализованный способ описания, который применим или не применим к реальным объектам

Структура математической теории

Аксиомы

Правила вывода (логика)

Теоремы

(которые применимы или не применимы к реальным объектам)

Дискретность

- **Дискре́тность** (от лат. *discretus* — разделённый, прерывистый) — свойство, противопоставляемое непрерывности, прерывность. Дискретность — всеобщее свойство материи, под дискретностью понимают:
- Нечто, изменяющееся между несколькими различными стабильными состояниями, например механические часы, которые передвигают минутную стрелку дискретно (скачкообразно) на 1/60 часть окружности
- Нечто, состоящее из отдельных частей, прерывистость, дробность. Например, дискретный спектр, дискретные структуры, дискретные сообщения.

Дискретное множество

- Дискретное множество – это счетное множество
- (то есть множество, элементы которого можно занумеровать натуральными числами). Любое конечное множество-счетное.

Первая неочевидная теорема

- Множество рациональных чисел – счетное

- Но сначала напомним некоторые факты о развитии чисел
- Натуральные числа Множество этих чисел обозначается **N**={1,2,…, n, … }
- Целые числа. Это все натуральные, 0 и
- {-1,-2,…, -n, … }. Обозначается **Z**

- Рациональные числа это несократимые дроби m/n, где числитель и знаменатель целые. Обозначается **Q .**
- Эти числа могут быть рассмотрены как точки плоскости с целыми координатами
- Можно показать, что множество таких точек - счетное. Следовательно, **Q – счетное.**

Иррациональные числа

- Долгое время в Древней Греции думали, что все числа – рациональны. Однако потом (около 550 г до н э) они доказали, что

- $\sqrt{2}$ - не рациональное число. Такие числа назвали иррациональными. Это был прорыв из дискретности в непрерывность.

Теорема Кантора

- Множество вещественных чисел **R** - несчетно.
- Доказательство этой теоремы очень простое, красивое и дальнейшее применение этого метода привело к фантастическим результатам в теории множеств и алгоритмов.

Диагональный процесс и его следствия

- Теорема об остановке Тьюринга
- Теорема Геделя

Трудности

- Трудности связаны с обработкой больших
- массивов информации. Жизнь в
- многомерном пространстве совсем другая

- Возникает так называемое проклятие размерности Беллмана.

Формальное определение Графа

- ***Ориентированный граф*** определяется как пара (V, E), где V – конечное множество, E - бинарное отношение на $V \times V$.
- Множество V называется множеством вершин, E – множество ребер.
-
-
- ***Неориентированный*** граф возникает, если отношение E симметрично.

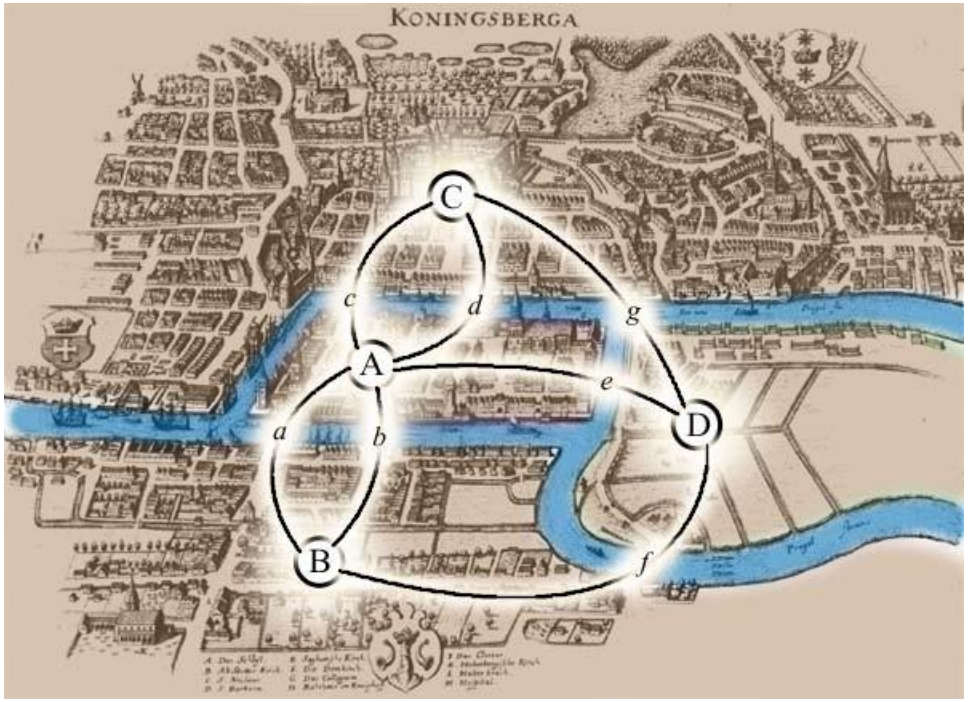
Граф неформально

- На самом деле граф очень простая вещь, которая может быть изображена картинкой. Построим граф, соответствующий отношению влюбленности.
- Пусть V – множество студентов данной группы. Каждый студент (студентка) изображается точкой. Если две точки a, b таковы, что
- (a, b) ∈ E, (то есть они влюблены), мы соединяем их ребром.
-
- ***Неориентированный*** граф возникает, если отношение E симметрично.
- (если все влюбленные отвечают друг другу взаимностью).

Картинка

- Предметом первых задач теории графов были конфигурации, состоящие из точек и линий, связывающих некоторые точки. Примеры таких конфигураций представляет следующий слайд

Первая задача о графах



Степени вершин

- Если в графе G имеется ребро e=(u,v), то говорят, что вершина v смежна с вершиной u.
- Ребро e инцидентно вершинам u,v.
- В неориентированном графе степенью вершины называется число ребер, в которые она входит.
-
- В ориентированном графе различают исходящую степень (out -degree) и входящую степень (in –degree), то есть сколько ребер выходит из вершины
- и сколько входит.

Маршруты

- Пусть G - неориентированный граф
- Маршрут – это последовательность ребер e_1, e_2,…, e_n, такая, что каждые два соседних ребра имеют общую инцидентную вершину. У маршрута есть начало и конец. Начало – это вершина, инцидентная только первому ребру.
- Если начало и конец совпадают, то маршрут называют циклическим.

Цепи

- Маршрут называется цепью, если каждое ребро встречается в нем не более одного раза, и простой цепью, если каждая вершина инцидентна не более чем 2 ребрам.
- Цепь называют циклом, если она- циклический маршрут, и простым циклом, если она – простая цепь.

Расстояние и диаметр

- Граф связен, если для каждых двух вершин u, v есть маршрут, который их соединяет.
- (то есть маршрут, где u– начало и v - конец)
- Этот маршрут можно считать простой цепью. Длина этой цепи – число ребер в ней. Расстояние d(u, v) между вершинами есть минимум длин всех возможных простых цепей, соединяющих u, v.
- Диаметр графа – это максимум всех возможных расстояний между вершинами графа.

ПУТИ

- ***Рассмотрим ориентированный граф G.***
- ***Путь*** длины k из вершины u в вершину v – это последовательность вершин, соединенных ребрами (стрелками), где первая вершина это u
- и последняя - это v. Вершина u – это начало пути, вершина v – конец пути.
-
- Если для данных двух вершин u,v существует путь, идущий из первой во вторую, то говорят что v достижима из u.
-
- Путь называется простым, если все вершины в нем различны.
- Путь называется циклом, если в нем начало и конец совпадают и в нем более чем одна вершина.

Циклы и связность

- ***Циклом*** в графе называется путь, где начальная и конечная вершина совпадают и содержащий не менее одного ребра. Цикл простой, если в
- нем все вершины различны.
-
-
- Граф, в котором нет циклов, называется ациклическим.
-
- Неориентированный граф называется связным, если для любой пары вершин имеется путь из одной в другую .
-
- Ориентированный граф называется сильно связным, если
- Из любой его вершины достижима любая другая.
-
- Отношение быть достижимым - это отношение эквивалентности
- Неориентированный граф распадается на компоненты связности, а ориентированный – на сильно связанные компоненты.

Матрица смежности графа G

- Это квадратная матрица (двумерный массив) δ(I, j), I=1,…N, j=1…N, где N - число
- Вершин в G. Для неориентированных графов δ(I, j) равно числу ребер, соединяющих вершины с номерами I, j.
- Для ориентированных графов δ(I, j) равно числу ребер, идущих из вершины с номером I в вершину с номером j.

Матрица инцидентности

- Пусть имеется неориентированный граф **G** с вершинами $v_1,\ldots, v_m$ и ребрами $e_1,\ldots e_n$. Тогда в матрице инцидентности с элементами $\varepsilon(l,j)$ будет n строк и m столбцов.
- При этом $\varepsilon(l,j)=1$, если вершина j инцидентна ребру l, и $\varepsilon(l,j)=0$ в противном случае.

Матрица инцидентности для орграфов

- Пусть имеется ориентированный граф **G** с вершинами v_1,…, v_m и ребрами e_1,… e_n. Тогда в матрице инцидентности с элементами ε(l,j) будет n строк и m столбцов.
- При этом ε(l,j)=-1 если вершина v_j – начало ребра e_l, ε(l,j)=1 если вершина v_j – конец ребра e_l,

и ε(l,j)=0 в противном случае. Если ребро -это петля, то ε(l,j)=2.

Типы графов

- Лес – ациклический неориентированный граф.
-
- Связный ациклический неориентированный граф называется деревом.
- (без выделенного корня)
-
-
- Деревья используются, в частности, для хранения информации. Типичный вид дерева – см картинку фрактального дерева ниже.

Теория роста графов

- Теория Эрдеша -Реньи –графы растут случайно. Реальные графы, соответствующие социальным сетям и т д растут не так.

Рост реальных графов

-
- Оказывается, реальные графы имеют ряд общих свойств, и несмотря на их грандиозный размер, благодаря этому мы можем их изучать.
- **Свойство 1. Малый мир**
-
- Диаметр графа Diam (V, E) мал – Diam < 10. Если граф связен, то для любых двух вершин мы можем пройти от одной вершины к другой по некоторому пути. Расстояние между вершинами – это длина кратчайшего пути, соединяющего две вершины.
-
- Диаметр – это наибольшее расстояние между вершинами. Например, если граф описывает знакомства между людьми, диаметр не превышает 6. Это свойство важно для эффективного взаимодействия.
-
-

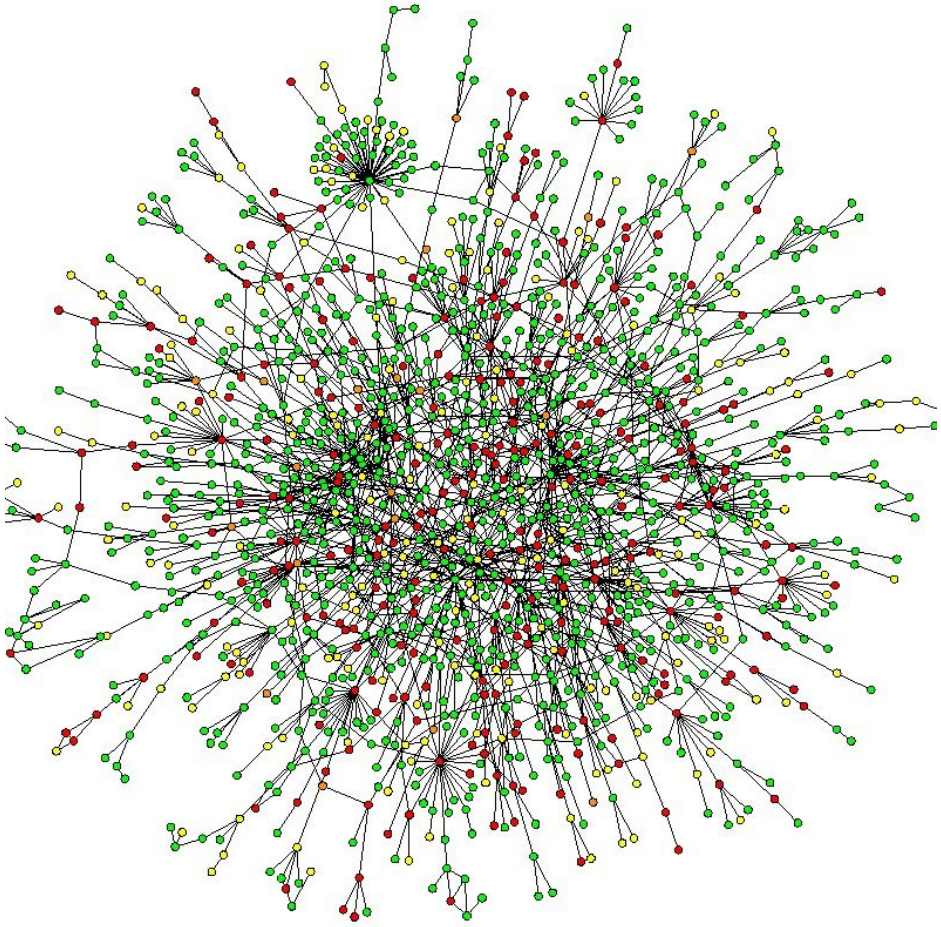
Свойства реальных графов

- **Свойство 2. Scale free structure**
-
- Граф устроен, грубо говоря, как страна с городами на карте. Имеется небольшое число больших городов, некоторое число средних городов, но большинство городов маленькие.
-
- Математически это означает что вероятность P(k), что наугад выбранная вершина соединена с k другими вершинами, примерно равна
-
- $P(k) = const\ k^{-b}$,
-
- где b – число, обычно оно между 2 и 3.
-
- **Свойство 3. В среднем вершины мало связаны, связанность (средняя степень вершин) от 2 до 10.**

Безмасштабные Графы

- **Свойство 2. Scale free structure**
-
- Граф устроен, грубо говоря, как страна с городами на карте. Имеется небольшое число больших городов, некоторое число средних городов, но большинство городов маленькие.
-
- Математически это означает что вероятность P(k), что наугад выбранная вершина соединена с k другими вершинами, примерно равна
-
- $$P(k) = \text{const}\ k^{-b},$$
-
- где b – число, обычно оно между 2 и 3.

Граф биохимических реакций



Булевские функции

Б. Ф. от N переменных отображает вектор булевских переменных длины N в булевскую переменную которая принимает значения 0 или 1.

•

Способы задания б. ф.

- Таблица
- Логические формулы в том числе КНФ и ДНФ
- Многочлены Жегалкина (АНФ)
- Функциональные схемы (нейронные сети)

ДНФ

- **Дизъюнкти́вная норма́льная фо́рма (ДНФ)** в булевой логике — нормальная форма, в которой булева формула имеет вид дизъюнкции конъюнкций литералов. Любая булева формула может быть приведена к ДНФ.[1]
- Пример- **A or B or (C Not D)**

КНФ

- **Конъюнкти́вная норма́льная фо́рма (ДНФ)** в булевой логике — нормальная форма, в которой булева формула имеет вид конъюнкции дизъюнкций литералов. Любая булева формула может быть приведена к КНФ.[1]
- Пример- **A B (C or (NOT D))**
- K-Sat – famous problem associated with conjuctive forms

АНФ

- Многочлен Жегалкина от булевских
- Переменных x, z, y, …– это многочлен
- С коэффициентами 0 или 1 от произведений этих переменных, где в качестве суммы используется исключающее или

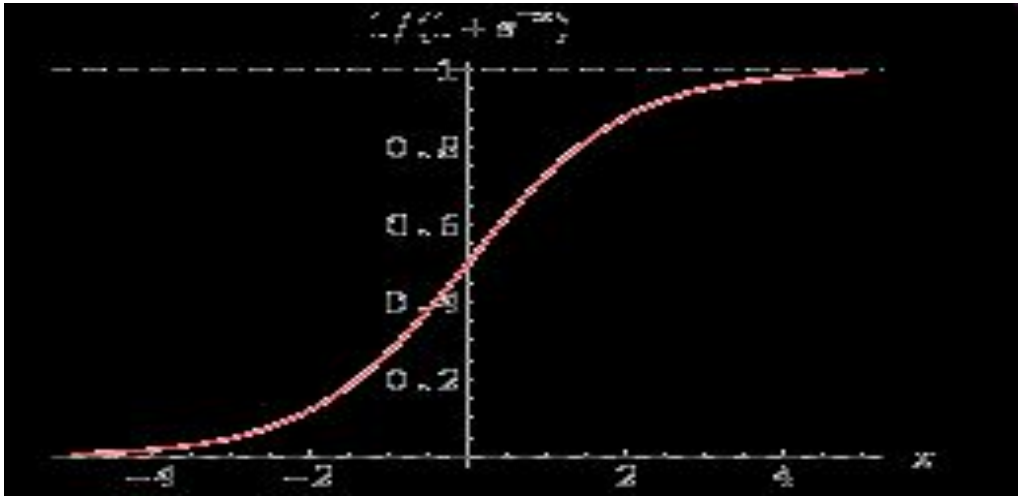
Основные булевские функции двух переменных (N=2)

- Конъюнкция
- Дизъюнкция
- Сложение по модулю 2
- Импликация
- Эквиваленция
- Штрих Шеффера
- Стрелка Пирса
- Отрицание

Таблицы для N=2

- Конъюнкция (OR) 0001
- Дизъюнкция (AND) 0111
- Импликация $\rightarrow$ 1011
- Штрих Шеффера $|$ 1110
- Стрелка Пирса $\downarrow$ 1000
- Исключающие или (XOR) 0110
- Таблицы неудобны для большого числа переменных ввиду их большого размера

Сигмоидальная функция



Суперупрощенная модель

$$u_i(t+1) = \sigma\left(\sum_k K_{ik}\, u_k(t) - h_i\right)$$

Индексы i, k в [1,… N]

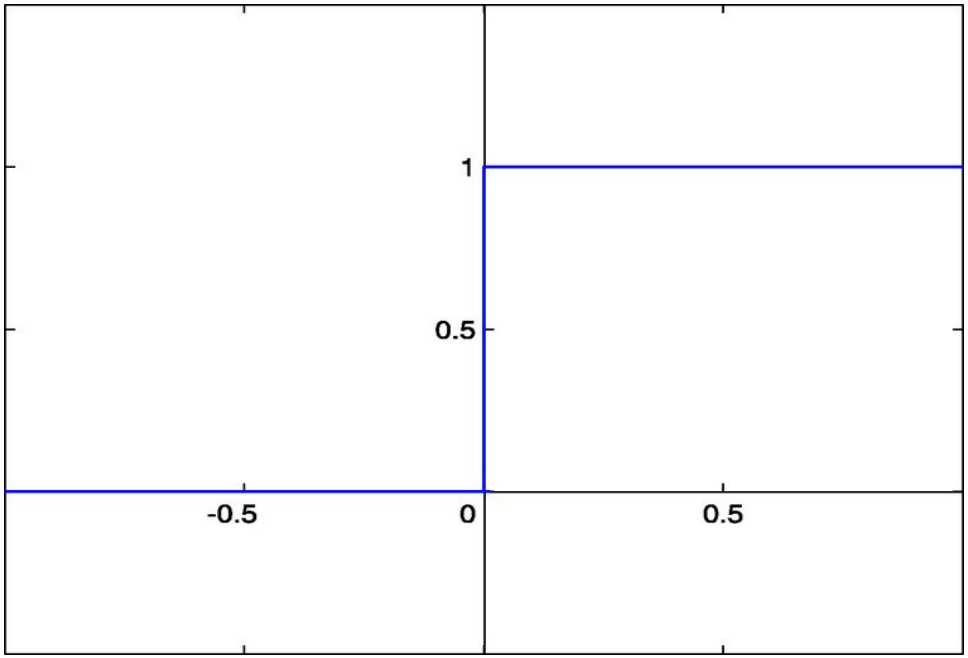
- K - синаптическая матрица, h_i - *пороги*
- σ - пороговая функция
- Модель простая на вид, но она может все !!! (моделирует любую машину Тьюринга и любые аттракторы, генерирует любые траектории). И иногда даже двух нейронов достаточно (N=2) (П. Койран и др.)

Примеры функций сигма

-

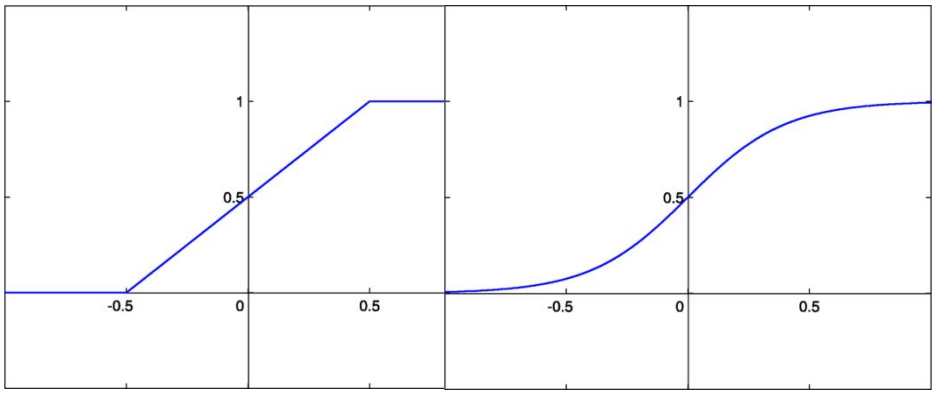
- $\sigma = 1/(1+ \exp(-a\, x))$
- $\sigma = H(x)$ (скачок)
- Кусочно-линейная (функция $\sigma = \max(x,0)$)
- Кусочно-полиномиальная

Функция скачка (безграмотный график)



Кусочно-линейная функция и функция Ферми

$$\sigma = 1/(1 + \exp(-a\,x)), \qquad a > 0.$$



Задача классификации

- Предположим, мы хотим создать автоматическую систему, которая различает два типа объектов, **A** и **B**. Системы технического зрения позволяют записать данные об объектах в цифровом виде. Предположим, что мы характеризуем объект с помощью набора признаков $x_1, x_2, \ldots x_k$. Признаки могут быть выражены с помощью целых чисел или даже булевских переменных (есть признак –нет признака)

Задача классификации

- Мы можем рассматривать признак как вектор с **k** компонентами, или элемент (точку) **k** -мерного Эвклидова пространства.

- Тогда задача классификации сводится к следующей математической задаче- имеется два множества , **A** и **B** точек в **k** -мерного Эвклидова пространства. Надо разделить их некоторой гиперповерхностью размерности
- **k-1.**

Обучение (learning)

- В задачах классификации нейронные сети обучаются таким образом.

- Имеются некие обучающие примеры, где ответ (принадлежность к классу A или B) известен. По ним строим сеть, которая проводит разделяющую поверхность.

Математическая формализация схемы обучения моделей ИИ (на примере задачи классификации)

- Формируется набор признаков $(x_1, x_2 \ldots x_n) = X$
- На основе опыта строятся обучающие примеры X(1), X(2), …., X(P), про которые известна принадлежность к классу A или B
- Индикатор: D(X)=1, если X из A, D(X)=0 при X из B.
- Берем некую систему y= F(X, W) где W набор параметров.

Общая схема обучения с учителем

- Дано - реальный выход устройства d(t) при входе X(t), где X(t) - вектор с компонентами (X_1 , X_2 , …, X_n), где t – номер опыта и число опытов равно T.
- Найти
- параметры модели w=(w_1 , …, w_m) такие, что выход модели F(**X, w**) и реальный были бы как можно ближе в среднеквадратичном смысле
- Введем эмпирический риск
- $R = T^{-1} \sum_{t=1,T} (d(t) - F(X(t), w(t)))^2$

Минимизация риска

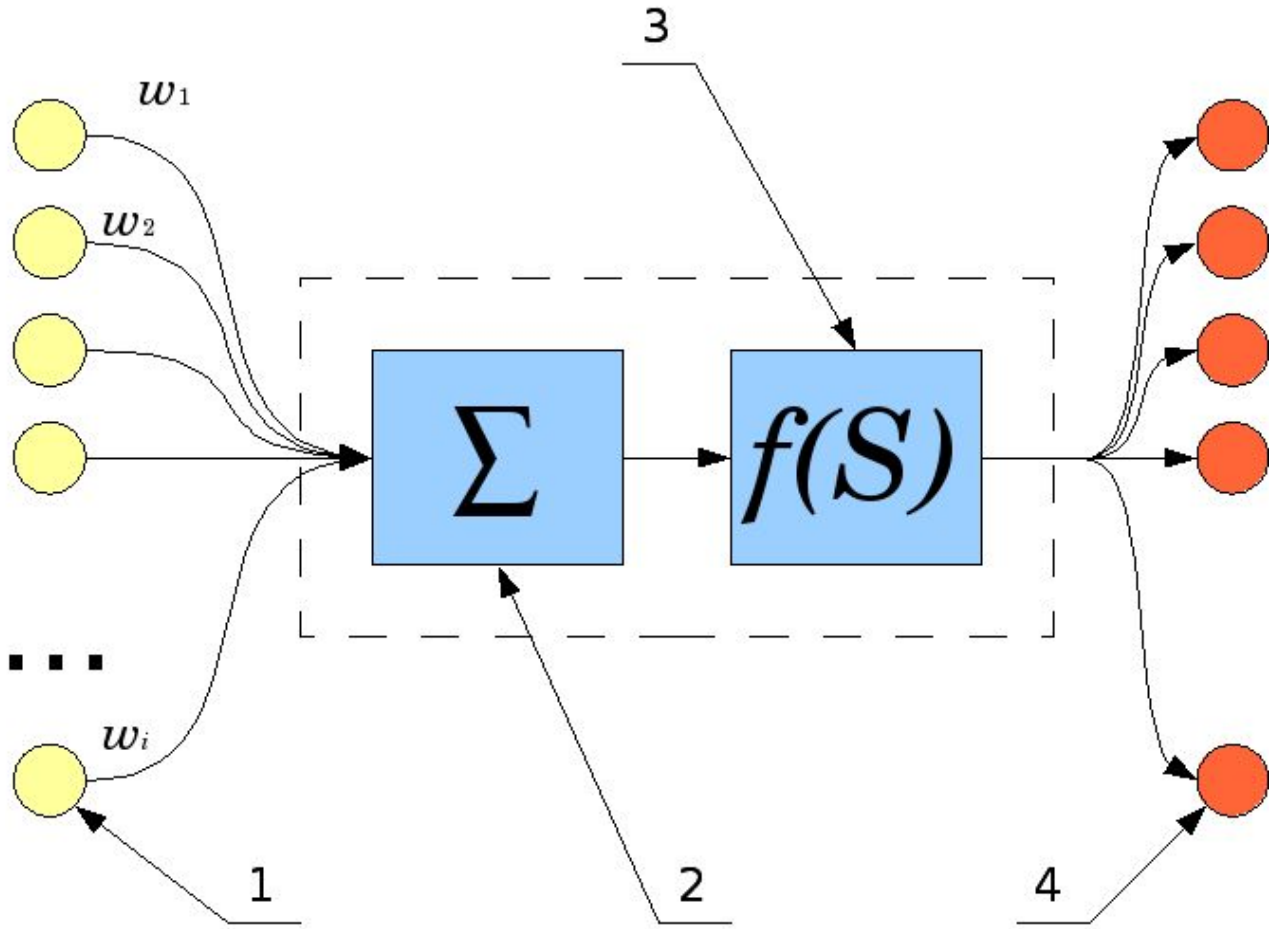
- Введем эмпирический риск
- $R = T^{-1} \sum_{t=1,T} (d(t) - F(X(t), w(t)))^2$

- Это среднее число ошибок классификации, которая делает сеть. Мы подбираем параметры w так, чтобы риск был минимален. Для этого есть специальные алгоритмы.

Однослойный персептрон

- Однослойный персептрон получает входной сигнал, заданный вектором x и выдает число
- $y = \sigma(\Sigma_k \; w_k \; x_k - h)$
- Или, если
- $S = w_1 \; x_1 + w_2 \; x_2 + \ldots \; w_n \; x_n - h,$
- $y = \sigma(S)$
- Предложен Ф. Розенблаттом. Здесь w_i , h – *параметры (синаптические веса и порог)*

Структурная форма



Моделирование логических функций персептроном

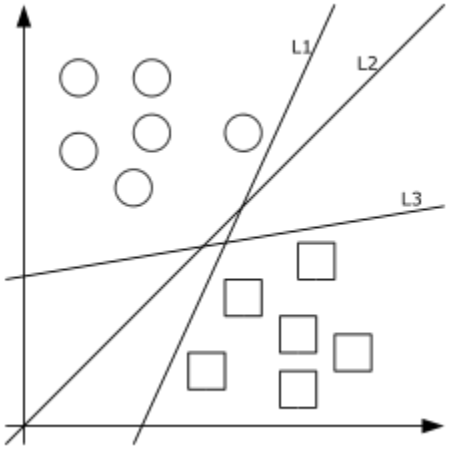
- Конъюнкция
- Дизъюнкция
- Отрицание
- Демократическое голосование

Решает ли персептрон все ??

- **БЫТЬ ИЛИ НЕ БЫТЬ ?**
- Марвин Минский показал, что возможности персептрона ограничены.

- Потому что он разделяет с помощью гиперплоскости (прямая для $k=2$ и плоскость для $k=3$)

Персептрон как игра в квадратики и шарики (n=2)



Геометрическая интерпретация

- $S=0 \leftrightarrow \quad w_1 x_1 + w_2 x_2 + \ldots w_n x_n = h$

- *Это - уравнение гиперплоскости в n-мерном линейном пространстве.*
- *Персептрон – это линейный разделитель*

Пример Минского

- Иногда квадратики и шарики неразделимы
- «Исключающее или» (XOR) нереализуемо однослойным персептроном.
- $XOR(x,y)=x+y \pmod 2$

Часть 3 Алгоритмы

- **Алгоритм** - формально описанная процедура, получающая исходные данные, называемые также *входом* алгоритма (input) и выдающая *результат* вычислений (output).
-
- Данное определение не формальное. Формальное определение может быть сделано несколькими различными, но эквивалентными способами
- (Тьюринг, Черч, Марков) и известно в литературе.
- Для практических приложений оно не очень важно.
-
- (Фундаментальные понятия, введенные упомянутыми авторами,
- это машина Тьюринга, рекурсивная функция и тезис Черча, нормальный алгоритм Маркова. В практической жизни без них можно обойтись . Однако, они нужны для доказательства фундаментальных теорем об алгоритмах

Пример Алгоритма

- Простейший пример алгоритма автор обнаружил в ирландском баре
- В городе Ренн (Франция), где он в течение 2004 -2009 работал визитинг профессором в местном университете (данный регион известен математикам тем, что дал миру гениального математика Анри Лебега,
- А любителям автогонок тем, что он дал миру Себастьяна Леэба, многократного чемпиона мира по ралли).
-
- Алгоритм.
-
- Step 1. Order at the bar
-
- Step 2 Drink anywhere
- Step 3. Return to step 1.
-
- Известные из школы алгоритмы, это алгоритм Евклида, упомянутый выше. Простым является также алгоритм Гаусса решения
- линейной системы уравнений. Возвращаясь к этому примеру с баром, заметим, что, при бесконечном количестве денег, данная процедура не останавливается (зацикливается). Таким образом, мы первый раз сталкиваемся с ***проблемой остановки***. Как мы увидим вскоре, это проблема далеко не проста.

Численные характеристики алгоритмов

- Время работы и требуемая память
- (running time T and memory M)

Время работы Алгоритма

- Скорость алгоритма определяется зависимостью времени работы (Running time) от размера входа (input size).
-
- Определение размера входа зависит от задачи - это может быть, например, число чисел на входе или общее число битов, необходимых, чтобы задать входные данные.
-
- Время работы - это число элементарных шагов. Время работы зависит от выбора элементарного шага. Можно предполагать, что одна строка псевдокода требует фиксированного числа операций

Пример: *Оценка времени работы алгоритма Гаусса.*

- Пусть мы решаем систему N уравнений с N неизвестными. Матрица системы содержит N^2 чисел. Тогда размер входа (число битов необходимых, чтобы задать вход) будет примерно $C\ N^2$
-
- Постоянная C – это число битов, необходимое для того, чтобы представить число (коэффициент в уравнении) в памяти компьютера.
-
- Элементарный шаг алгоритма – это умножение и сложение двух чисел.
-
- Можно доказать, путем анализа алгоритма Гаусса, что мы используем
- порядка N^3 таких операций. Итак, время выполнения алгоритма пропорционально кубу числа неизвестных. Получаем формулу
-
- **$T(|X|)$ (running time) = константа * $|X|^{3/2}$**
-
- **где $|X|$ - input size.**

Практическое применение

- Каковы практические следствия этой формулы (как ее практически применить)? Этот вопрос тем более актуален, поскольку мы не знаем постоянной в правой части формул, определяющих скорость работы алгоритма.
-
- Берем простой тестовый пример, где мы уверены, что наш компьютер быстро справится с задачей и замеряем время. Например, систему 10 уравнений он решил за 10 секунд. Допустим, теперь мы хотим решить систему 100 уравнений. Поскольку время пропорционально кубу числа уравнений, получаем 1000 секунд, то есть примерно 17 минут.

Простые, сложные и неразрешимые задачи.

- Простые, когда время решения растет как **степень** размера входа
-
- T(|X|) (running time) = константа * $|X|^{d}$
-
- где |X| - input size, d - некоторое число. (1)

Сложные (hard)

- Сложные, когда время решения растет, в общем случае, намного быстрее, чем степень размера входа. Например, время есть показательная функция от размера входа
-
-
- T(|X|) (running time) = const exp(c |X|)
- где |X| - input size, c > 0 – некоторое число. (2)
-
- Это не исключает того, что в каких -то частных случаях задача
- Решается быстро (для некоторых входов X верна формула (1)).

Неразрешимые (non decidable)

- Вообще алгоритмически неразрешимые, то есть нет алгоритма,
- который для любого входа выдает решение. Соответственно, программу тоже не написать.
-
- В свое время великий философ и математик Готфрид Лейбниц думал,
- что есть алгоритм решения любой задачи. В настоящее время стало ясно,
- что это не так.

Зацикливание программ

- Гениальный математик Ален Тьюринг доказал Важную Теорему
-
- *В общем случае проблема остановки алгоритмически не разрешима - нет алгоритма, который про любой другой алгоритм скажет, заканчивает он работу когда нибудь, или нет.*
-
- Это означает, что нельзя написать программу, которая про любую другую программу скажет, остановится ли ее выполнение когда - нибудь, или нет.

Примеры

- Пример простой задачи - решение системы линейных уравнений.
- Она решается в полиномиальное время методом Гаусса.
-
- Примером сложной задачи является задача коммивояжера. (См раздел о графах).

Поиск кратчайшего пути в графе

- *Пример простой задачи.*
-
- Введем так называемый взвешенный (weighted) граф (то есть припишем каждому ребру число, называемое весом).
-
- Например, если граф описывает авиалинии в стране, то весом ребра (рейса из одного города в другой) может быть расстояние между городами или
- минимальная цена авиабилета.
-
-
- *Задача* – Найти кратчайший путь в взвешенном графе (V, E) из одной вершины в другую (в том смысле, что сумма весов ребер должна быть минимальной).
-
- Эта задача решается множеством разных алгоритмов. Обозначим число вершин $|V|$, число ребер - $|E|$. Размер входа тогда $|E| + |V|$.
-
- Тогда, например, алгоритм Дейкстры требует $T = |V|^2 + |E|$ шагов. Для неплотных графов время его работы может быть меньше.
-

Алгоритм Флойда-Уоршалла

- Пусть вершины графа $G=(V,E), |V|=n$ пронумерованы от 1 до n и введено обозначение d_{ij}^k для длины кратчайшего пути от i до j, который кроме самих вершин i,j проходит только через вершины 1… k. Очевидно, что d_{ij}^{0}— длина (вес) ребра (i,j), если таковое существует (в противном случае его длина может быть обозначена как ∞).
- Существует два варианта значения d_{ij}^{k}, где k=1,…,n:
- Кратчайший путь между i,j не проходит через вершину k, тогда $d_{ij}^{k}=d_{ij}^{k-1}$
- Существует более короткий путь между i,j, проходящий через k, тогда он сначала идёт от i до k, а потом от k до j. В этом случае, очевидно, $d_{ij}^{k}=d_{ik}^{k-1}+d_{kj}^{k-1}$
- Таким образом, для нахождения значения функции достаточно выбрать минимум из двух обозначенных значений.

Алгоритм ФУ

- Тогда рекуррентная формула для d_{ij}^{k}} имеет вид:
- d_{ij}^{0} — длина ребра (i,j);
- d_{ij}^{k}=\min(d_{ij}^{k-1}, d_{ik}^{k-1}+d_{kj}^{k-1}).}
- Алгоритм Флойда-Уоршелла последовательно вычисляет все значения d_{ij}^{k} для всех i,j для k от 1 до n. Полученные значения d_{ij}^{n} являются длинами кратчайших путей между вершинами i,j.

Псевдокод ФУ

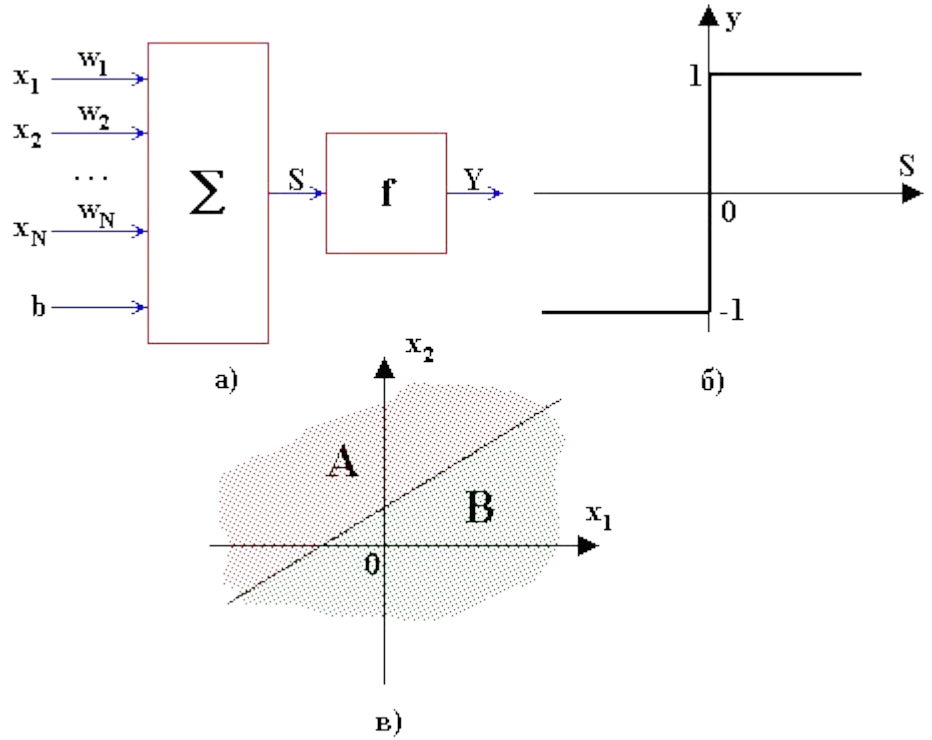
- **for** k = 1 **to** n
- **for** i = 1 **to** n
- **for** j = 1 **to** n
- W[i][j] = min(W[i][j], W[i][k] + W[k][j])

- end
- End
- end

Задача коммивояжера - трудная

- *Пример сложной задачи – задача коммивояжера.*
-
- Предположим, некий человек собирается посетить (используя авиарейсы) по одному разу все города некоей страны и вернуться в точку вылета, побывав в каждом городе и не более 1 раза.
-
- При этом он хочет заплатить минимальную цену за билеты.
-
- Математически задача формулируется так - найти в графе простой цикл, обходящий все вершины, и имеющий минимальную длину (сумма весов ребер должна быть минимальна).
-
- Для графа общего вида задача коммивояжера относится к классу NP полных задач и является сложной.
-
- Даже задача о существовании простого цикла в графе, обходящего все вершины (гамильтонов цикл) является сложной при большом размере графа. Размер входа здесь – число вершин $|V|$.

Алгоритм обучения однослойного персептрона



Первые шаги

- ***Шаг 1***. Инициализация синаптических весов и смещения:
- Значения синаптических весов $\mathbf{w_i(0)}$ (**0** <= **i** <= (**N**)) и смещение (порог) нейрона **h** устанавливаются равными некоторым малым случайным числам.
- Обозначение: $\mathbf{w_i(t)}$ - вес связи от **i**-го элемента входного сигнала к нейрону в момент времени **t**.
- ***Шаг 2***. Предъявление сети нового входного и желаемого выходного сигналов:
- Входной сигнал $\mathbf{x = (x_0, x_1 \ldots x_N)}$ предъявляется нейрону вместе с желаемым выходным сигналом **d**.
- ***Шаг 3***. Вычисление выходного сигнала нейрона

Формула перенастройки (адаптации весов)

$$w_i(t+1) = w_i(t) + r[d(t) - y(t)]x_i(t);$$

$$0 \le i \le N-1,$$

$$d(t) = \begin{cases} +1, & \text{если это класс A} \\ -1, & \text{если это класс B}. \end{cases}$$

(3.2)

Шаг 4. адаптация (настройка) значений весов:

где **r** – параметр обучения (меньше 1),

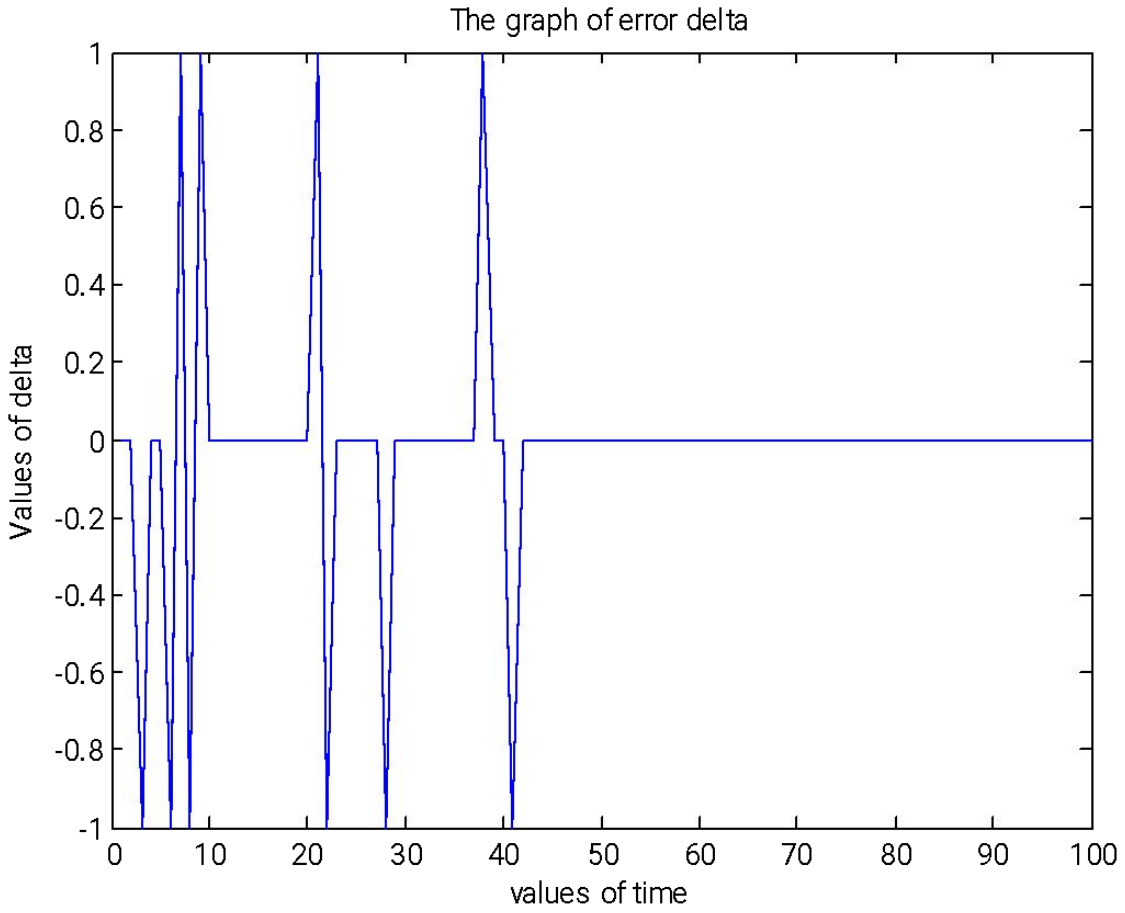
Пример программы на Матлабе

- Напишите программу по этому алгоритму
- for m=1:N
- out=out+ wout(m)*x(m);
- end;
- if(ssout > 0) yout(k)=1; else yout(k)=0; end
-

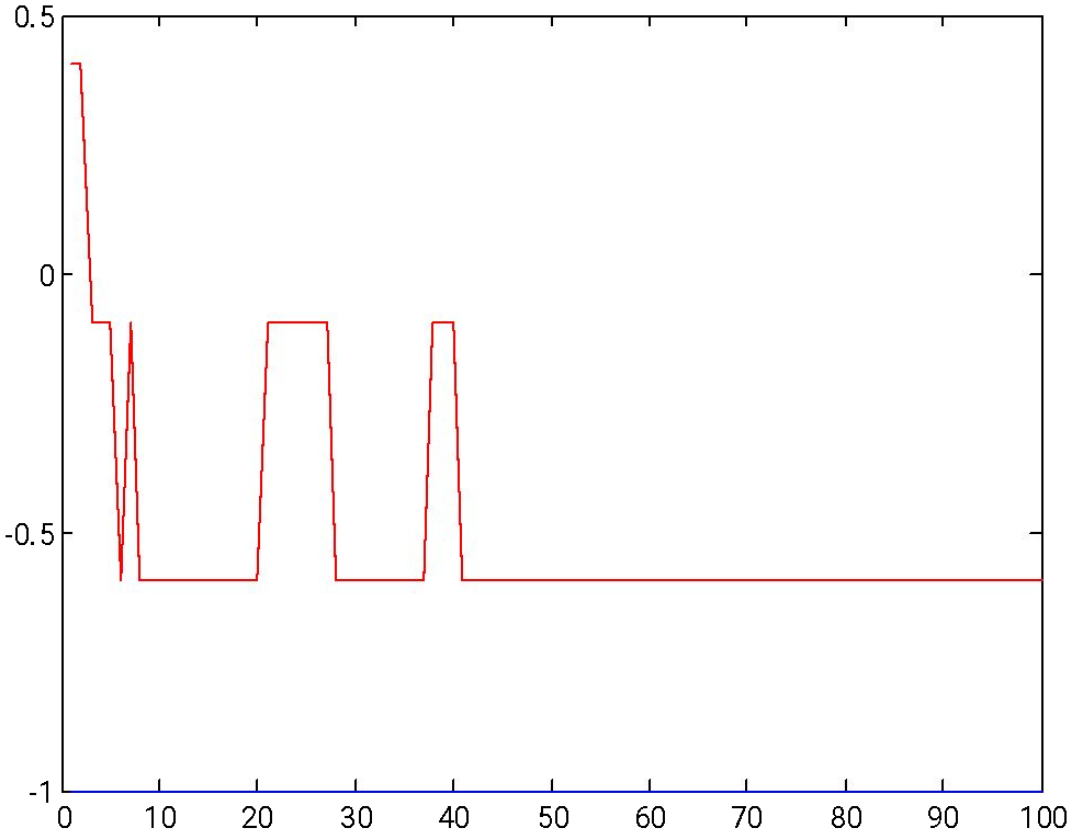
Пример программы

- delta(k)=y(k)-yout(k);
- for i=1:N
- wout(i)=wout(i) + r*delta(k)*x(i);
-
- end;

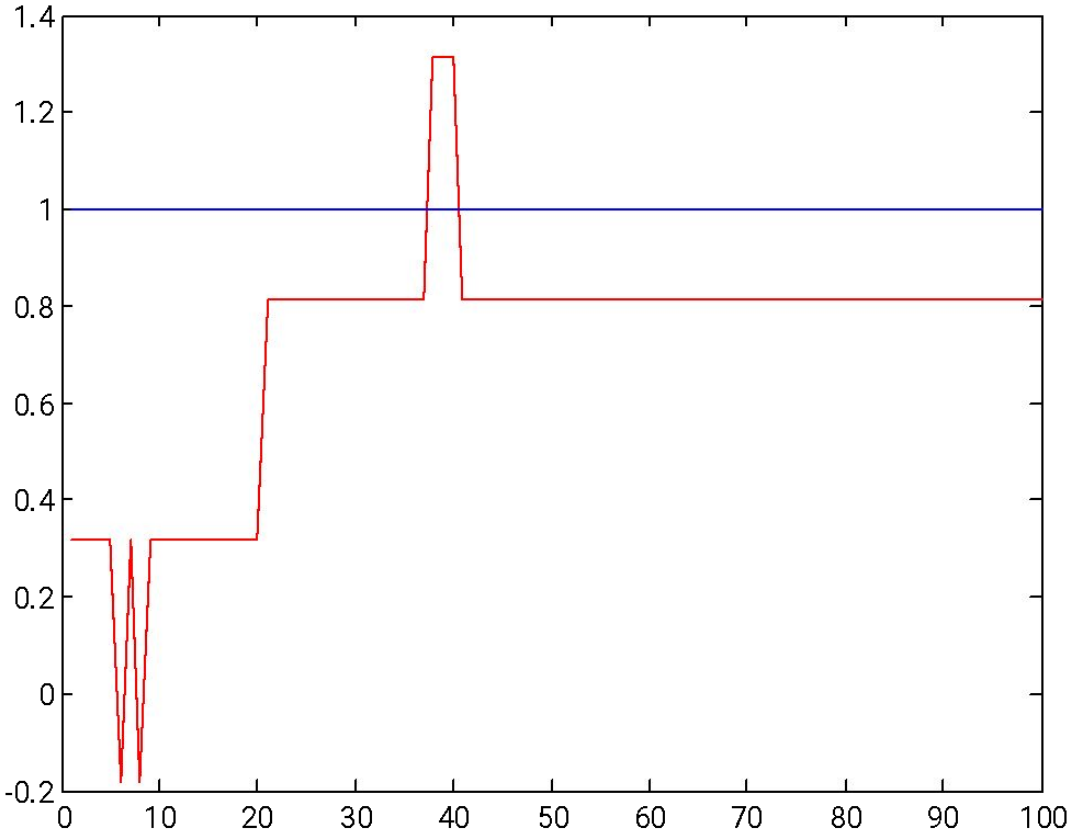
Иллюстрация работы алгоритма



Эволюция синаптических весов w_1 … w_m



Эволюция весов



Откуда появился этот алгоритм

- Это алгоритм градиентного спуска, который ищет параметры, чтобы минимизировать ошибку. Алгоритм итеративный
- Формула итераций выводится так. Берем риск
- $R= \sum_{t=1,T} (d(t) - F(\mathbf{X}(t), \mathbf{w}(t)))^2$
- Подставляем вместо F формулу для персептрона и вычисляем градиент по **w**. Получаем формулу перенастройки весов.

Градиентный спуск

- Минимизация функции $R(w)$ итеративно, движением по градиенту $\text{grad } R(w)$
- $w(t+1)=w(t) + r \text{ grad } R(w)$

Всегда работает для выпуклых вниз функций

- Не работает или плохо работает для функций с большим числом локальных минимумов

Кое что об алгоритмах

- Алгоритм получает вход X и выдает ответ Y
- Размер входа обозначаем |X|

- Пример – решить систему линейных уравнений с n неизвестными AX=B
- Размер входа пропорционален n^2
- Время работы n в кубе

Оценка скорости работы алгоритма

- Если время работы T не больше чем постоянная умножить на f(|X|)
- пишем
- T= O(f(|X|)
- Для хороших задач и алгоритмов типично

- $f = |X|^2 \ln |X|$ или $|X|^2$

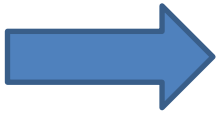
Решаемое и не решаемое

- Задача считается трудной для алгоритма, если максимальное время работы алгоритма $T(|X|)$ растет как показательная функция размера входа $|X|$
- $T(X) = O(\exp(a\,|X|)), \quad a > 0$

- Пример - задача коммивояжера
- Задача алгоритмически неразрешима, если алгоритма вообще нет
- Пример - задача об остановке, диофантовы уравнения

Трудное и легкое в среднем и некоторые замечания

- Задача линейного программирования для
- вещественных переменных теоретически трудная для симплекс метода (в особых случаях), но на практике легкая (в среднем легкая)
- Легкая $T(X)= O(|X|^{b}), \quad b > 0$
- Для целых и булевских переменных задача линейного программирования, вообще говоря, трудная



- Это значит , что для булевских (дискретных) синаптических весов задача обучения нейронной сети - трудная

Жадные алгоритмы

Задача о деньгах. Получить нужную сумму с помощью наименьшего числа купюр

Алгоритм – делаем жадный выбор. На каждом шаге берем самую большую деньгу.

Замечание – для экзотических денежных систем работать не будет.

Градиентный спуск

- Минимизация функции F(X) итеративно, движением по градиенту grad F(X)

- Работает для выпуклых вниз функций
- Не работает для функций с большим числом локальных минимумов

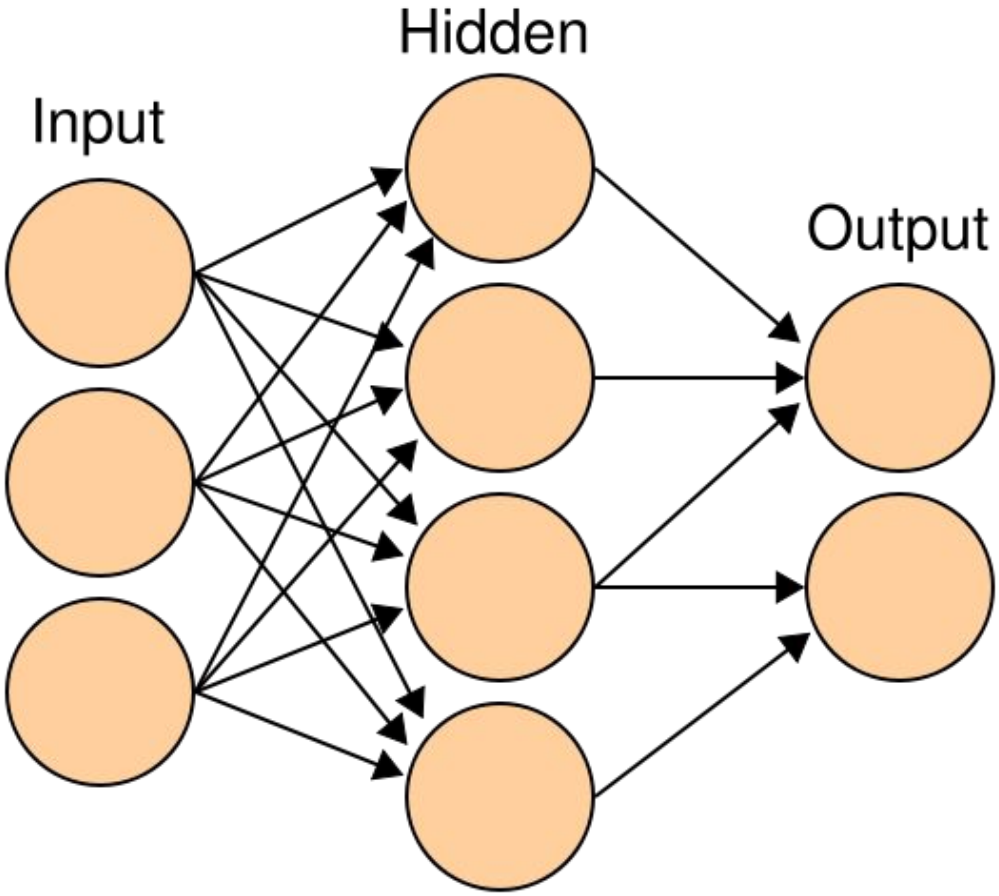
Динамическое программирование

- Метод предложен Р. Беллманом для задач оптимального управления. Основная идея- это работает, если задача можно разбить на подзадачи и решение оптимальное для подзадачи, оптимально для задачи.
- В теории ИИ применяется для НММ
- Пример – принцип Ферма и распространение света.

Принцип оптимальности подзадач

- Пример. Принцип Ферма.
- Допустим, мы должны найти оптимальную траекторию из A в B. Пусть E – точка между
- A и B на оптимальной траектории.
- Тогда участок траектории между E и B тоже оптимален.

Архитектура многослойного персептрона



Абсолютное разделение

- ФУНДАМЕНТАЛЬНАЯ ТЕОРЕМА. *Пусть имеется произвольное множество синих точек и непересекающееся с ним множество красных точек в многомерном линейном пространстве. Тогда с помощью многослойного персептрона всегда можно* ***разделить*** *эти два множества, то есть провести гиперповерхность, разделяющую их.*

Некоторые задачи для практики

- 1)Сколько существует булевских функций
- От двух переменных ? От трех ?
- 2) Найти булевские функции от 2 переменных, представимые однослойным персептроном
- 3) Представим ли штрих Шеффера однослойным персептроном ?
- 4) Представима ли стрелка Пирса однослойным персептроном

Некоторые задачи для практики

- Найти многослойный персептрон минимальной архитектуры, моделирующий XOR
- Построить многослойный персептрон, моделирующий выборы в США
-

Нечеткая логика (fuzzy logic)

- задание
- Сила ++
- Моисеенко + +
- Иванов Сергей +
- Траков +
- Большаков +
- Иванов Михаил +
- Пакин +

Некоторые задачи для практики

- Типовые задания по алгоритмам
-
- Как представить сумму в 1 миллион 644 тысяч 500 рублей с помощью наименьшего числа купюр ? , Используйте жадный алгоритм
- Вася и Женя решают систему **n** линейных уравнений с **n** неизвестными методом Гаусса на одном и том же компьютере. Вася решил систему 100 уравнений за секунду. Сколько потребуется времени Жене для системы из 150 уравнений?
-
-
-
- Решите задачу решения системы линейных уравнений в системе Матлаб.
-
- Минимизируйте функцию $2x^2 + xy + (y-2)^2$ с помощью генетического алгоритма в системе Матлаб.
-
- Минимизируйте ту же функцию с помощью gradient descent в системе Матлаб
-
- Вася и Женя решают задачу сортировки n чисел с помощью двух разных алгоритмов , которые они сами придумали. Васин алгоритм решает задачу за 20 n^3 элементарных шагов А Женин за 100 n^2 ln n шагов. Для каких n Женин алгоритм будет быстрее ?.
- Составьте задачу линейного программирования с N переменными и d ограничениями со случайными параметрами Решите ее в среде МАтлаб и найдите как скорость расчета зависит от N при фиксированном d.
- Составьте задачу линейного программирования с N переменными и d ограничениями со случайными параметрами Решите ее на Матлабе и найдите как скорость расчета зависит от d при фиксированном N.