

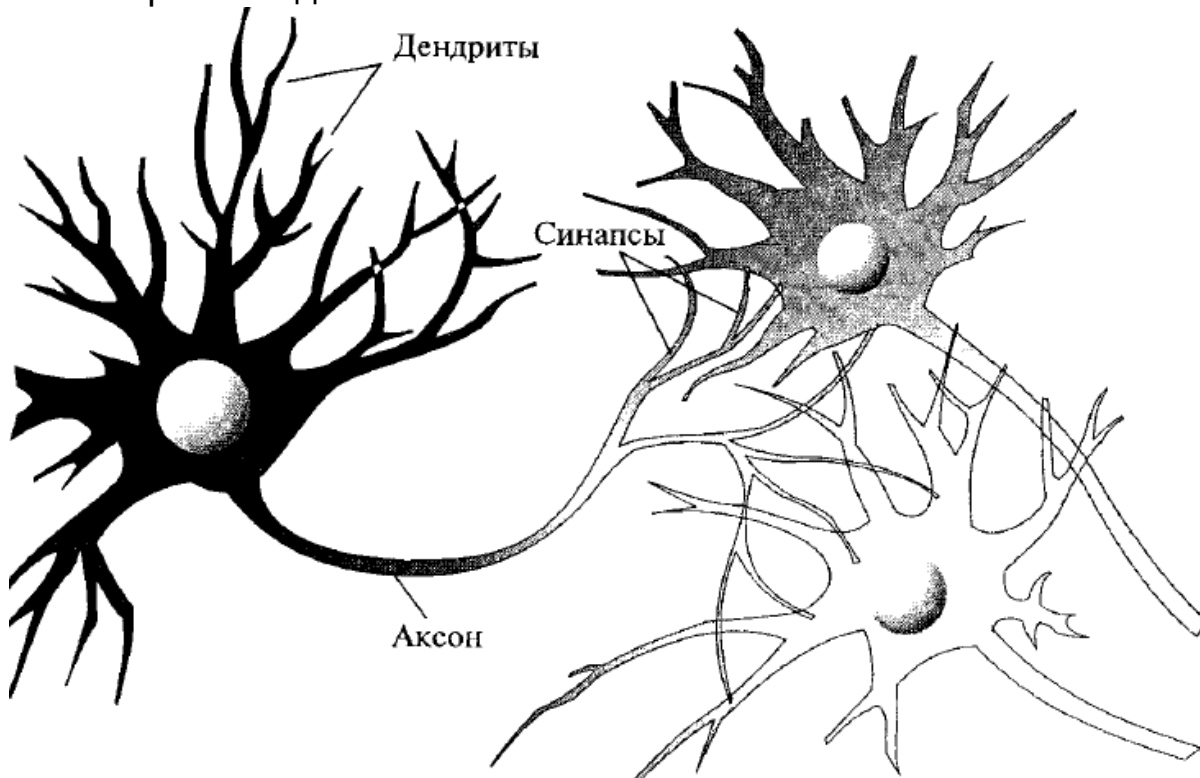
СМАДЗУ Лекция 6

Искусственные нейронные сети (ИНС)

Биологические предпосылки

Привлекательные свойства мозга - способность

- ♦ выполнять в параллельном режиме логические, распознающие и оптимизирующие функции;
- ♦ давать приемлемый ответ даже при неполных или ошибочных входных данных;
- ♦ устойчиво работать при сбоях или выходе из строя одного или даже группы нейронов (пример с Луи Пастёром);
- ♦ обучаться, т. е. улучшать свои характеристики в процессе обработки данных.



Биологический нейрон – клетка, состоящая из **сомы** - тела нейрона, к ней через отростки **дендриты** подходят **аксоны** - линии передач от других нейронов. **Синапс** преобразует входную информацию, переводя нейрон в состояние возбуждения или торможения.

Мозг состоит из 10^{11} параллельно работающих нейронов, каждый связан с 10^4 других нейронов, так что всего в биологической нейросети 10^{15} нейросвязей!

Лекция проф. Е.Черниговской
<http://snob.ru/selected/entry/99460>

Архитектура искусственных нейронных сетей (ИНС) имеет некоторое сходство с естественными нейронными сетями. Она состоит из элементов, называемых **нейронами**, – это простые логические устройства, характеризуемые:

1. уровнем активации;
2. топологией связи друг с другом;
3. мерой взаимодействия с другими нейронами, называемой синаптической силой связи или **весом**. Веса этих связей различны и могут определяться в зависимости от решаемой задачи. Синапс – точка соединения нейронов;
4. выходным уровнем, который связан с уровнем активации посредством некоторой функции.

Вся система состоит из большого числа нейронов, причем результат работы ИНС малочувствителен к характеристикам конкретного нейрона. Она допускает возможность параллельной обработки информации.

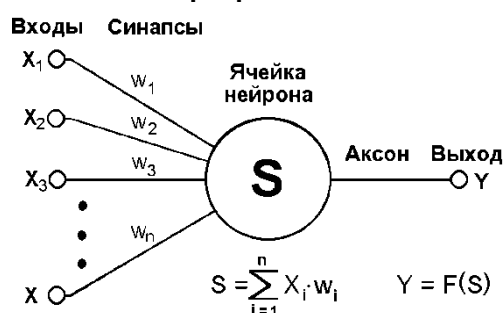
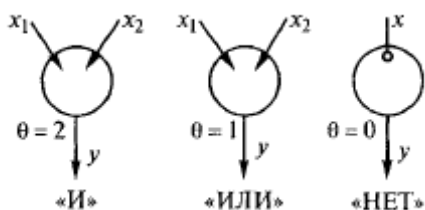


Рис. 1 Функциональная схема искусственного нейрона

Нейрон состоит из взвешенного сумматора и функции активации $f(s)$ (рис. 1). Множество входных сигналов, обозначенных x_1, x_2, \dots, x_n , поступает на искусственный нейрон. Каждый сигнал умножается на соответствующий вес w_1, w_2, \dots, w_n , и поступает на суммирующий блок, обозначенный Σ . Каждый вес соответствует «силе» одной биологической синаптической связи. Суммирующий блок складывает взвешенные входы, на выходе получается сигнал $S = \sum_{i=1}^n w_i x_i$, который затем преобразуется активационной функцией f и дает выходной нейронный сигнал $y = f(s)$.

Если активационная функция $f(s)$ является пороговой, то при превышении порога нейрон трактуется, как **возбужденный** ($=1$), иначе – **покоящийся**.



Заметим, что даже один такой нейрон позволяет решать различные логические задачи.

Нелинейная активационная функция $f(x)$ может иметь различный вид, как показано на рис. 2. Одной из наиболее распространенных является нелинейная функция с насыщением, так называемая логистическая функция или сигмоид (т.е. функция S-образного вида):

$$f(x) = \frac{1}{1 + e^{-\alpha x}}$$

При уменьшении

параметра α сигмоид становится более пологим, в пределе при $\alpha=0$ вырождаясь в горизонтальную линию на уровне 0.5, при увеличении α сигмоид приближается по внешнему виду к функции единичного скачка с порогом T в точке $x=0$. Из выражения для сигмоида очевидно, что выходное значение нейрона лежит в диапазоне $[0, 1]$.

Существенная особенность нейронных сетей состоит в том, что зависимость между входом и выходом находится в процессе **обучения сети**. Для обучения нейронных сетей применяются алгоритмы двух типов (разные типы сетей используют разные типы обучения): управляемое ("**обучение с учителем**") и не управляемое ("**без учителя**"). Чаще всего применяется обучение с учителем.

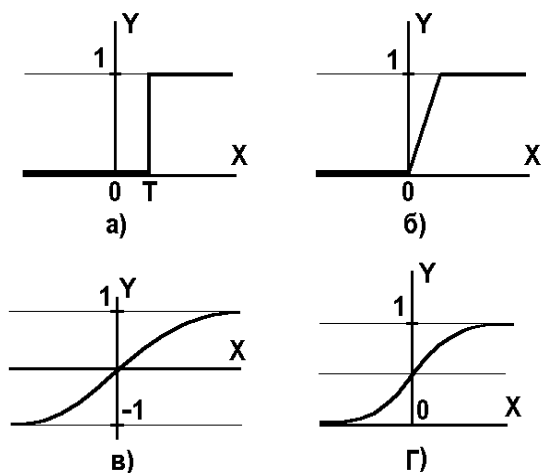


Рис.2 а) функция единичного скачка; б) линейный порог (гистерезис); в) сигмоид – гиперболический тангенс; г) сигмоид – формула (1)

ИНС, наиболее распространенные в физике, определяются двумя типами связей: **прямоточные сети без обратных связей**, например, **многослойные перцептроны** (МСП) или радиально-базисные (РБФ) сети (см. рис.3) или **полносвязные сети**, в которых все нейроны связаны друг с другом (см. рис.4), как в нейронной **сети Хопфилда** (ХНС).

Мы можем также рассматривать и **клеточные автоматы** как специальный тип нейронных сетей с локальными связями.

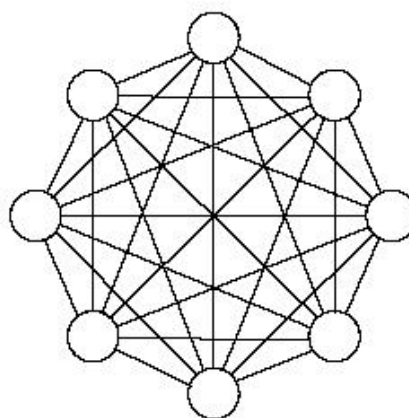
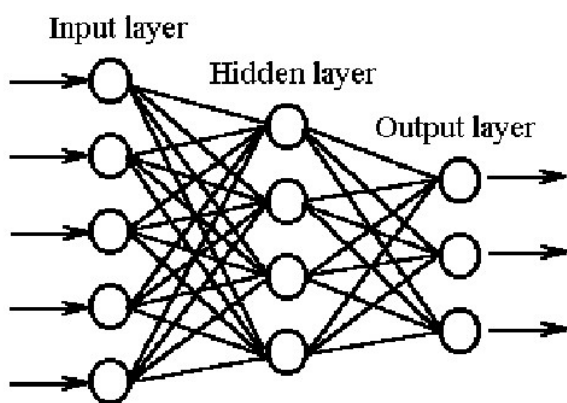


Рис.3. Прямоточная ИНС (МСП или RBF-сети)

Рис.4. Полносвязная ИНС (сеть Хопфилда)

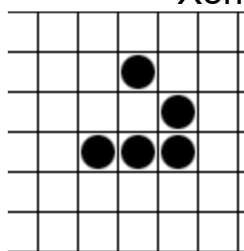


Рис.5. Фрагмент клеточного автомата

Нейронные сети успешно применяются везде, где нужно решать задачи классификации, прогнозирования или распознавания.

Наиболее многочисленны применения многослойных перцептронов, основанные, как правило, на традиционной схеме: сначала подбор структуры и обучение ИНС на большой обучающей выборке из смоделированных событий методом обратного распространения ошибок; после чего найденные

структура и веса воплощаются в ИНС в виде программы или аппаратно.

Многослойный персептрон (МСП)

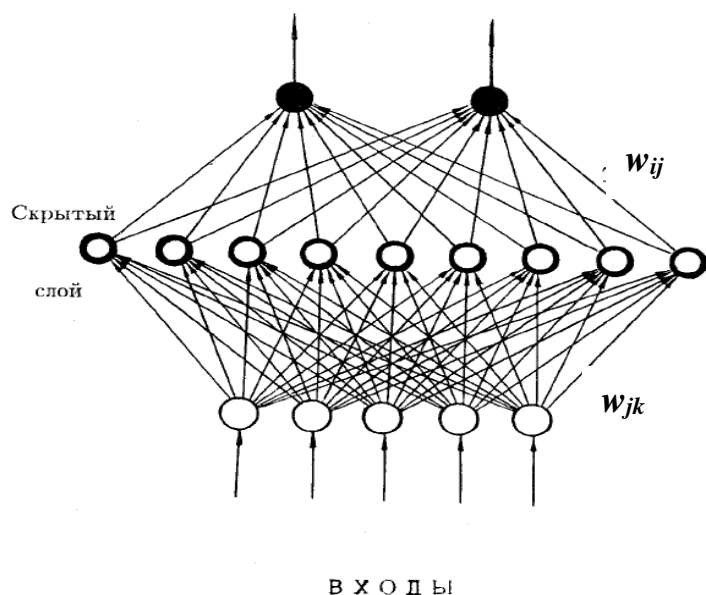


Рис.3. Схема трехслойного персептрона

является прямооточной сетью без обратных связей, архитектура которого состоит из последовательно w_{ij} соединённых слоёв, где нейрон каждого слоя своими входами связан со всеми нейронами предыдущего слоя, а выходами - следующего. (Рис.3) .

Входному вектору $X=(x_1, x_2, \dots, x_n)$ МСП ставит в соответствие выходной вектор $Y=\{y_j\}$. Для трехслойного персептрона входные сигналы сначала преобразуются в нейронах скрытого слоя как

$$h_k = f\left(\sum_i w_{ik} x_i\right) \quad \text{где } h_k \text{ – выход } k\text{-го нейрона, } f(x)$$

–активационная функция,

w_{ik} – вес синаптической связи i -го и k -го нейронов,

x_i – сигнал, исходящий от i -го входного нейрона сети.

Затем сигналы от нейронов скрытого слоя преобразуются нейронами выходного слоя как

$$y_j = f\left(\sum_k w_{kj} h_k\right)$$

Это преобразование полностью описывается с помощью синаптических весов w_{ik}, w_{kj} , которые и должны быть найдены, чтобы использовать МСП для решения задачи.

Алгоритм обучения персептрона может быть реализован на компьютере, причем сеть может выполнять свои функции классификатора только после обучения, т.е. соответствующего изменения весов. По этой причине процедуру подстройки весов обычно называют «обучением» и говорят, что сеть «обучается». МСП обучают, подавая множество образов по одному на его вход и подстраивая веса до тех пор, пока для всех образов не будет достигнут требуемый выход. При обучении предполагается, что для каждого входного вектора существует парный ему целевой вектор, задающий требуемый выход. Вместе они называются **обучающей парой**. Сеть обучается на многих парах. Группа обучающих пар называется **обучающей выборкой**. Наиболее распространенный алгоритм обучения МСП является **алгоритм обратного распространения ошибки** состоящий в минимизации суммарной ошибки сети.

$$E = \frac{1}{2} \sum_{k=1}^m \sum_j (y_j^{(k)} - z_j)^2 \quad (2.3)$$

где y_j – значение j -го выхода сети, z_j – идеальное (желаемое) значение j -го выхода сети, $k = 1, m$ – индекс по числу обучающих выборок (эпох обучения) .

Наиболее известным способом поиска минимума E является **метод наискорейшего спуска** в пространстве весов. При этом ошибки (точнее величины коррекции весов) распространяется в обратном направлении от выходов к входам, сквозь веса, соединяющие нейроны (Рис.3). Первый слой нейронов (соединенный с входами) служит лишь в качестве распределительных точек, суммирования входов здесь не производится. Входной

сигнал просто проходит через них к весам на их выходах.

Дифференцируя (2,3) по всем весам w_{ik}, w_{kj} , входящим в y_j и, соответственно, в h_j мы получим систему уравнений для определения весов. Размерность этой системы равна числу межнейронных связей в сети, т.е. уже для простой сети, изображенной на рис.3, будет равна 32. Для сетей с большим числом входных нейронов число уравнений может достигать сотен и даже тысяч, что образно называют «проклятием размерности» многослойных нейросетей. Рекурсивные правила метода наискорейшего спуска приводят к итерационным формулам изменения весов на каждой эпохе обучения t :

для весов выходного слоя $\Delta w_{kj}(t+1) = -\eta(y_j^t - z_j^t)f'(y_j^t)h_k^t$

для весов скрытого слоя

$$\Delta w_{ik}(t+1) = -\eta \sum_j w_{kj} f'(y_j^t) f'(h_k^t) x_k^t$$

где η - параметр скорости обучения, $f(\mathbf{x})$ – активационная функция, а \mathbf{z} – известный результат классификации.

Каждая модификация всех весов в сети называется **эпохой обучения**. В общем случае количество эпох неограниченно. Сеть считается обученной, если в какой-либо эпохе она максимальная ошибка обучения уменьшится до заданной точности.

Чаще всего МСП используется для непосредственной классификации изображений – на вход подаётся или само изображение в каком-либо виде, или набор ранее извлечённых ключевых характеристик изображения, на выходе нейрон с максимальной активностью указывает принадлежность к распознанному классу. Если эта активность ниже некоторого порога, то считается, что поданный образ не относится ни к одному из известных классов.

Процесс обучения устанавливает соответствие подаваемых на вход образов с принадлежностью к определённому классу. Это называется **обучением с учителем**. Такой подход обеспечивает непосредственное сравнение сетью самих образов, но с увеличением числа классов время обучения и работы сети возрастает экспоненциально. Поэтому проблема обучения МСП является весьма важной в практических приложениях. Среди многих известных способов ее решения отметим оптимизацию структуры МСП **методом главных компонент** для сокращения числа нейронов в скрытом слое и, соответственно, числа связей между ними, т.е. числа весов, по которым ведется минимизация. Чем компактнее сеть по числу нейронов в ней, тем легче ее обучить. Чтобы оценить весовые параметры с достаточной точностью **длина обучающей последовательности должна быть минимум на порядок больше числа этих параметров**. Существуют рекомендации по поводу выбора числа нейронов сети или оптимального числа межнейронных связей, т.е. весов N_w :

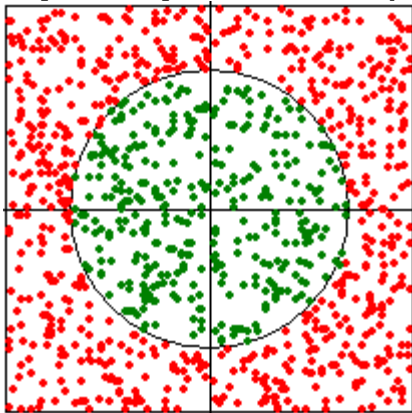
$$\frac{N_y Q}{1 + \log_2 Q} \leq N_w \leq N_y \left(\frac{Q}{N_x} + 1 \right) (N_x + N_y + 1) + N_y$$

Здесь: N_x размерность входа, N_y — размерность выхода, Q — длина обучающей выборки

В некоторых задачах, например медицины, где достоверная модель изучаемых зависимостей может и не существовать, приходится вести обучение и проверять качество работы обученной сети на часто небольшом экспериментальном материале. В этих

условиях выбор оптимальной структуры сети может играть ключевую роль.

Пример. Классификация с помощью МСП



Главное – модель обучающей выборки: на вход сети подают по 3 числа: координаты точки (X,Y) и признак (1 - внутри круга или 0 – вне его).

Обучение нейросети Neural Network Wizard: два входных нейрона $N_x=2$, один – выходной $N_y=1$. Длина обучающей выборки Q связана с числом нейронов скрытого слоя N_h . Найдем эту связь по формуле (левая

часть неравенства: $\frac{N_y Q}{1 + \log_2 Q} \leq N_w$).

Заметим, в нашем случае $N_w = 2 * N_h + 1 * N_h = 3 N_h$

Для $Q=100$ ($1 + \log_2 Q \approx 7,6$) получим $100/7,6 \leq 3N_h$

т.е. $N_h \geq 4$; для $Q=500$ получаем $N_h \geq 16$.

Контрольное тестирование обученной ИНС обычно проводят на $M=20-30$ примерах, не входивших в обучение, и определение эффективности нейросетевого

классификатора $\varepsilon \approx \frac{R}{M} \cdot 100\%$ Здесь R - количество верных ответов персептрона.

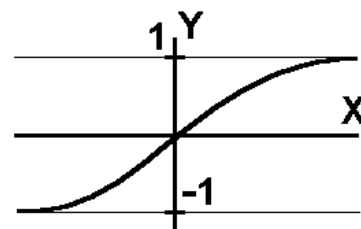
Нейронная сеть Хопфилда (ХНС)

Это **полносвязная** сеть из **бинарных** нейронов s_i с **симметричной** **весовой матрицей** $w_{ij} = w_{ji}$, $w_{ii} = 0$. Эволюция ХНС приводит ее в некоторое состояние устойчивого равновесия. Функционал энергии сети – это билинейная функция Ляпунова

$$E(s) = - \frac{1}{2} \sum_{ij} s_i w_{ij} s_j.$$

Теорема Хопфилда: в результате эволюции $E(s)$ убывает в локальные минимумы, соответствующие точкам стабильности сети.

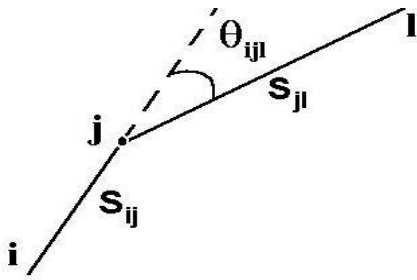
Для нахождения глобального минимума E сеть термализуется. В соответствии с теорией среднего поля состояния нейронов $v_i = \langle s_i \rangle_T$ усредняются по температуре T . Эволюция сети определяется уравнением динамики среднего поля:



$v_i = 1/2(1 + \tanh(-\partial E / \partial v_i \cdot 1/T)) = 1/2(1 + \tanh(H_i / T))$,
где $H_i = \langle \sum_j w_{ij} s_j \rangle_T$ – локальное среднее поле нейрона. Значения v_i переставшие быть целочисленными и определяют уровень активности нейрона, т.е. в случае $v_i > v_{min}$ нейрон считается активным. Температура убывает по схеме «ИМИТАЦИОННОГО ОТЖИГА» (simulated annealing).

Пример применения

Распознавание треков. Метод сегментов.



Имеется множество N

экспериментальных точек на плоскости. Требуется выбрать (распознать) среди них те, по которым проходит некоторое число непрерывных гладких кривых (треков).

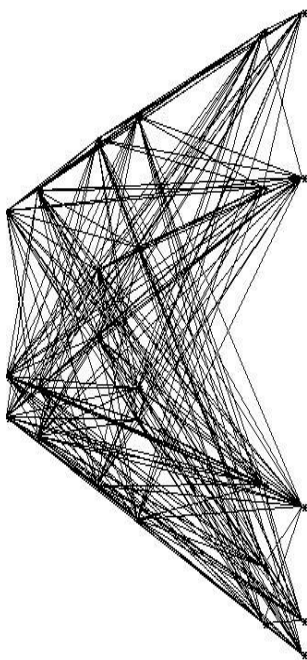
Вводится нейрон s_{ij} как направленный сегмент, соединяющий точки i, j .

Энергетический функционал (Денби и Петерсон, 1988) состоит из двух частей: $E = E_{cost} + E_{constraint}$,

где
$$E_{cost} = -\frac{1}{2} \sum_{ijkl} \delta_{jk} \frac{\cos^m \theta_{ijl}}{r_{ij} r_{jl}} v_{ij} v_{kl},$$

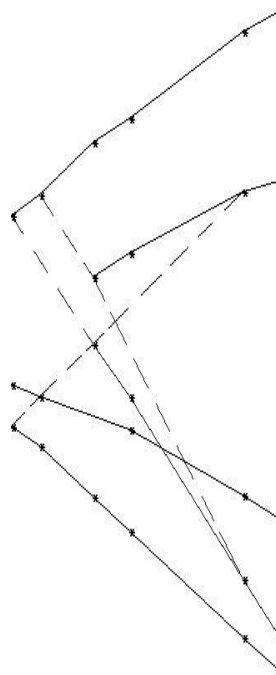
поощряет связи нейронов принадлежащих одному и тому же треку, т.е. короткие смежные сегменты с малым углом между ними.

$E_{constraint}$ запрещает как межтрековые связи (бифуркации), так и чрезмерный рост числа самих треков.



на нулевой итерации

всего 244 нейрона



на 30-ой итерации

$V_{ij} > 0.5$
у 26 нейронов

Эффективность 99% за счет гибридного алгоритма, комбинирующего применение быстрого традиционного алгоритма для 92% событий и нейропрограмму для тех 8%, которые не прошли сшивку треков в пространстве.

Клеточный автомат

Клеточный автомат — дискретная динамическая система, представляющая собой совокупность одинаковых клеток, одинаково соединенных между собой. Все клетки образуют так называемую решетку клеточного автомата. Эти правила предписывают изменения состояния каждой клетки в следующем такте времени в ответ на текущее состояние соседних клеток, а также, возможно и ее самой. В общем случае клеточные автоматы обладают следующими свойствами.

- **Изменения значений всех клеток происходят одновременно** после вычисления нового состояния каждой клетки решетки.
- **Решетка однородна** - невозможно различить какие-либо две области решетки по ландшафту.
- **Взаимодействия локальны.** Лишь клетки окрестности (как правило, соседние) способны повлиять на данную клетку.
- **Множество состояний клетки конечно.**

Алгоритм клеточных автоматов

1. Вводятся **два массива** для хранения состояний клеток: первый из них содержит текущее состояние каждой клетки, второй предназначен для хранения нового ее состояния.
2. Определяется функция переходов клетки решетки. Для выявления следующего ее состояния в качестве параметров в функцию переходов передаются текущие значения состояний клеток окрестности, возможно, включая ее саму. На нулевом шаге решетка (первый массив) заполняется начальными данными, что полностью определяет поведение системы для выбранных решетки и функции переходов клетки..

3. Для вычисления новых состояний вводится цикл. На каждой итерации для любой клетки, используя в качестве переменных элементы первого массива, определяется ее новое состояние, помещаемое во второй массив. Значения аргументов функции переходов берутся из первого массива.
4. Поскольку у клеток, обрамляющих решетку не хватает соседей, решетка свертывается в тор, т.е. краевые строки и столбцы обрамляются дубликатами строк (столбцов) с противоположной стороны, состояния клеток в которых обновляются в каждом цикле.

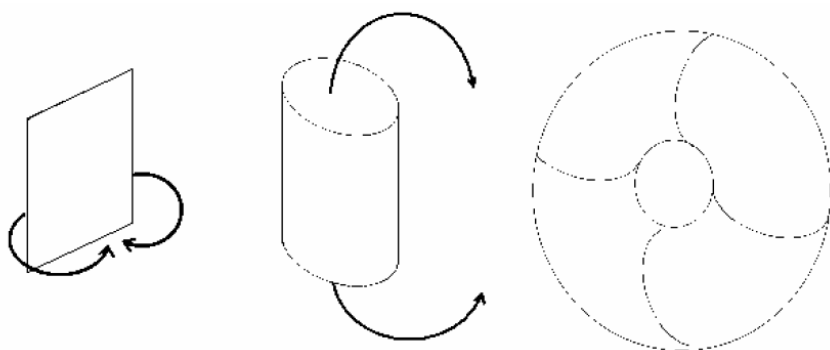


Рис. 7. Схема образования тора

5. По завершении итерации значения из всех элементов второго массива переносятся в первый, что обеспечивает синхронное изменение значений состояний всех клеток решетки.
6. **Визуализируется содержимое решетки.** В зависимости от ее размерности (одномерная или двумерная) отображение решетки производится по-разному.
 - 5.1. Для одномерной (линейной) решетки после каждой итерации выводится соответствующая ей строка. Их расположение одна под другой позволяет наблюдать динамику системы во времени (ось времени направлена вертикально вниз).
 - 5.2. Для двумерной (плоскостной) решетки в каждый момент времени отражается результат лишь последней итерации. Последовательный переход от одной итерации к другой позволяет наблюдать динамику системы.

Игра «Жизнь»

Впервые, идея таких автоматов отмечена в работах Дж.Фон Неймана в 1940-х годах, когда он работал над идеей

саморепродуцирующихся машин. Вплоть до конца 60-х идея клеточных автоматов была забыта и лишь в 1970 Джон Конвей, математик Кембриджского университета, описал ныне широко известный двумерный клеточный автомат, названный Игра "Жизнь" .

Игра разыгрывается на двумерном массиве во избежание краевого эффекта, свернутом в тор. Каждая клетка может быть в одном из двух состояний: клетка может быть "живой" (на экране - черной) или "мертвой" (на экране - белой).

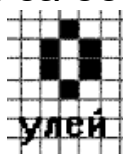
Правила эволюции автомата в игре «Жизнь»:

1. Выживание: если клетка в текущем моменте времени жива, то в следующем такте времени она будет жива в лишь в том случае, если две или три из восьми соседних клеток живы в текущем такте времени.

2. Гибель: В противном случае (одиночество или перенаселение), клетка погибает.

3. Рождение: Если клетка мертва, то в следующем такте времени она оживает тогда и только тогда, если ровно 3 соседние клетки живы в текущем такте времени. В противном случае клетка остается мертвой.

Если в качестве начального состояния установить случайное распределение живых и мертвых клеток, запустить модель и проследить за ее эволюцией, то можно увидеть следующее:



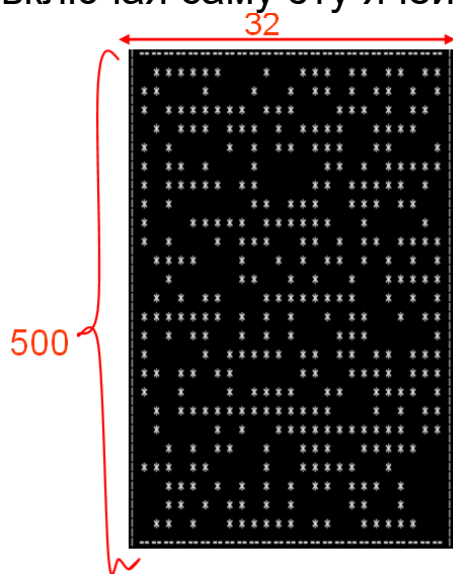
Часть структур стабилизируются и не изменяются во времени (как «улей» на рис. слева), часть претерпевают циклические изменения («мигалка-блинкер» на рис. в центре), и, наконец, некоторые развиваются, не повторяясь, практически неограниченное время («планер» на рис. справа).

2D клеточный автомат для генерации случайных векторов

На решетке 32*500 строится 2D клеточный автомат, каждая ячейка которого определяется по своим соседям по формуле:

$$\phi_1(i, j) = \left(\sum_{i'=i-1}^{i+1} \sum_{j'=j-1}^{j+1} e_{i'j'} \right) \bmod 2 \quad (1)$$

т.е. в ячейку засылается ноль при четном числе соседей (включая саму эту ячейку) или 1 при – нечетном.



Решетка сворачивается в тор для выполнения граничных условий. Ячейки вначале случайно засеваются нулями и единицами. Потом на каждую эволюцию КА по формуле (1) получаем 500 32-значных случайных числа, образующих 500-мерный случайный вектор с равномерным распределением.

Использовалось для вычислений по квантовой хромодинамике (КХД)

Одномерные клеточные автоматы

В одномерном (линейном) клеточном автомате решетка представляет собой цепочку клеток (одномерный массив), в которой для любой из них, кроме крайних, имеется по два соседа.

Для устранения краевых эффектов решетка «заворачивается» в кольцо, что позволяет для всех клеток автомата использовать следующее соотношение:

$$y'[i] = f(y[i-1], y[i], y[i+1]),$$

где f — функция переходов клетки;

$y'[i]$ — состояние i -й клетки в следующий момент времени;

$y[i-1]$ — состояние $(i-1)$ -й клетки в данный момент времени;

$y[i]$ — состояние i -й клетки в данный момент времени;

$y[i+1]$ — состояние $(i+1)$ -й клетки в данный момент времени.

Пример применения. Простейший **генератор случайных чисел** на базе одномерного клеточного автомата из 17 ячеек. Пусть функция переходов клетки имеет такой вид:

$y'[i] = y[i-1] \circ y[i] \circ y[i+1]$, где \circ — символ логической операции «исключающее или» (сложение по mod 2).

Засеем любым путем строку из 17 клеток случайными битами 0 или 1 и свернув ее в тор-кольцо, т.е. добавив $y(0)=y(17)$ и $y(18)=y(1)$, запустим эволюцию клеточного автомата. В каждом цикле в качестве псевдослучайного числа x из отрезка $(0,1)$ выдаем

$$x = \sum_{i=1}^{17} 2^{-i} y(i) .$$

Полученную последовательность следует

проверить на соответствие гипотезе о равномерном распределении в отрезке $(0,1)$ по критерию χ^2 .

Заметим, что при гистограммировании на m ячеек в случае равномерного распределения, выражение критерия упрощается:

$$\chi^2 = \frac{m}{n} \sum_{k=1}^m \nu_k^2 - n$$

Проверка работы ГСЧ на равномерность распределения в многомерном кубе весьма усложняется из-за экспоненциального роста числа ячеек гистограммы с ростом размерности.

Однако есть возможность сохранить число ячеек гистограммы при любой размерности d случайного вектора, проверяемого на равномерность. Такое разбиение пространства называется

метод вложенных гистограмм

Пояснить сущность метода вложенных гистограмм удобно для случая $d = 2, m = 4$.

Разделим квадрат $1 * 1$ (Рис.1) на 4 фигуры с равной площадью ($1/m = 1/4$), первая из которых представляет квадрат $a_1 * a_1$ (соответственно, $a_1 = \sqrt{1/m}$), а остальные расположены между 4-мя вложенными квадратами со сторонами $a_k = \sqrt{k/m}$. Таким образом, в общем случае:

$$a_k = \sqrt[k]{k/m}, k = 1, \dots, m$$

Соответственно, попадание в k -ю фигуру определяется неравенством вида:

$$a_k < \max[x_1, x_2] < a_{k+1}$$

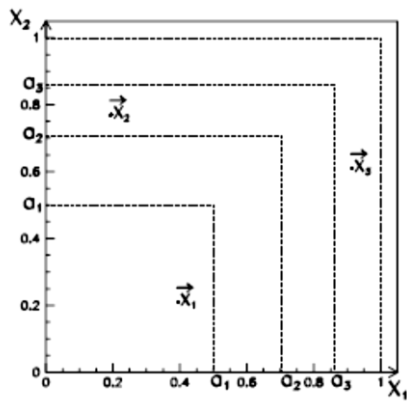


Рис.1

что в общем случае размерности d
записывается как

$$\sqrt[d]{k/m} < \max[x_1, \dots, x_d] < \sqrt[d]{(k+1)/m}$$

или

$$k < m \cdot (\max[x_1, \dots, x_d])^d < k + 1,$$

откуда следует, что ячейка гистограммы с номером k
должна быть увеличена на 1.