Лабораторная работа №2
Моделирование ассоциативная памяти при помощи нейронных сетей.

Цель работы: изучить обучение и функционирование релаксационных ИНС в качестве ассоциативной памяти при решении задач распознавания образов.

Нейронная сеть Хопфилда

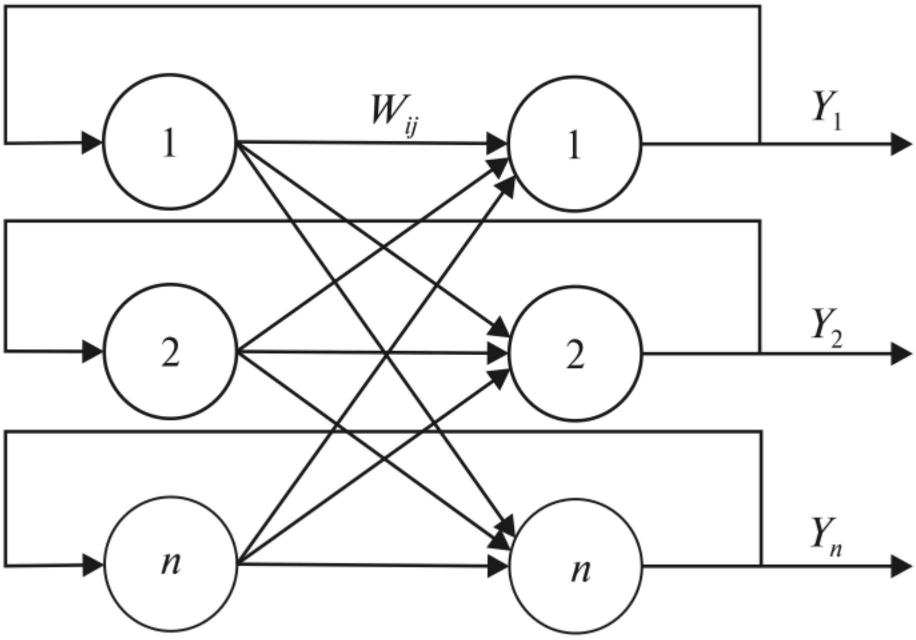

Рисунок 1 – Архитектура нейронной сети Хопфилда

Режимы работы сети:
1. Асинхронный
2. Синхронный

Для асинхронного режима работы в каждый такт времени вычисляется значение только k-го нейронного элемента. Нейронный элемент выбирается из тех, которые ещё не изменили своё состояние. Формулы:

$$y_j(t+1) = F(\sum_i W_{ij} y_i(t)), \text{ если j = k}$$
$$y_j(t+1) = y_j(t), \text{ если j != k} \tag{1.1}$$

Для синхронного режима работы в каждый такт времени вычисляются все нейронные элементы (формула 1.1 для всех i или же запись в матричном виде):

$$y_j(t+1) = F(WY(t)) \tag{1.2}$$

Процесс настройки выполняется до тех пор, пока нейронные элементы на выходе не перестанут изменяться. Данный процесс называется ***релаксацией***.

Пример использования НС Хопфилда для решения задачи корректировки числовой последовательности:

1. Задаем исходный числовой вектор или вектор образов:

$$Y = [1 \quad 1 \quad 1 \quad 0]$$

2. Настраиваем весовые коэффициенты на данный входной вектор:

$$W = (2Y-1)^T(2Y-1) - I = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \end{bmatrix} [1 \quad 1 \quad 1 \quad -1] - \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$W = \begin{bmatrix} 1 & 1 & 1 & -1 \\ 1 & 1 & 1 & -1 \\ 1 & 1 & 1 & -1 \\ -1 & -1 & -1 & 1 \end{bmatrix} - \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 1 & -1 \\ 1 & 0 & 1 & -1 \\ 1 & 1 & 0 & -1 \\ -1 & -1 & -1 & 0 \end{bmatrix}$$

3. Вычисляем значения выходного нейрона для асинхронного режима

Для примера вычислим значением 1-го выходного нейрона. Для этого будет использоваться 1-й столбец матрицы весовых коэффициентов. Используем зашумленный образ Y' вместо оригинального:

$$Y' = [0 \quad 0 \quad 1 \quad 0]$$

$$y_i(t+1) = sign(S_i(t))$$

$$S_1(0) = Y'W_1 = [0 \quad 0 \quad 1 \quad 0] \begin{bmatrix} 0 \\ 1 \\ 1 \\ -1 \end{bmatrix} = 0 + 0 + 1 + 0 = 1$$

$$y_1(1) = sign(1) = 1$$

В результате, для 1-го выходного нейрона получим:

$$Y'(1) = [1 \quad 0 \quad 1 \quad 0]$$

Проделаем аналогичную операцию для 2-4 нейронов:

$$S_2(1) = Y^{'}W_2 = [1 \quad 0 \quad 1 \quad 0] \begin{bmatrix} 1 \\ 0 \\ 1 \\ -1 \end{bmatrix} = 2 * (1 + 0 + 1 - 1) = 2$$

$$y_2(2) = sign(2) = 1$$

$$Y^{'}(2) = [1 \quad 1 \quad 1 \quad 0]$$

$$S_3(2) = Y^{'}W_3 = [1 \quad 1 \quad 1 \quad 0] \begin{bmatrix} 1 \\ 1 \\ 0 \\ -1 \end{bmatrix} = 3 * (1 + 1 + 0 - 1) = 3$$

$$y_3(3) = sign(3) = 1$$

$$Y^{'}(3) = [1 \quad 1 \quad 1 \quad 0]$$

$$S_4(3) = Y^{'}W_4 = [1 \quad 1 \quad 1 \quad 0] \begin{bmatrix} -1 \\ -1 \\ -1 \\ 0 \end{bmatrix} = 3 * (-1 - 1 - 1 + 0) = -9$$

$$y_4(4) = sign(-9) = 0$$

$$Y^{'}(4) = [1 \quad 1 \quad 1 \quad 0]$$

Если продолжить процесс дальше, то можно заметить, что состояние нейронов на выходе перестанет изменяться. Это свидетельствует о переходе сети в стабильное состояние - релаксацию. Для этого понадобилось 4 такта времени (т.к. метод за 1 такт обрабатывает только 1 нейрон) или 1 раз полностью обработать зашумленный вектор признаков.

4. Вычисляем значения выходного нейрона для синхронного режима:

Для этого задаем зашумленный вектор-признаков, вычисляем взвешенную сумму и находим выходное значение используя пороговую функцию активации:

$$Y^{'} = [0 \quad 0 \quad 1 \quad 0]$$

$$y(t+1) = sign(S(t))$$

$$S(t) = Y(t)W$$

$$S(0) = Y^{'}(0)W = [0 \quad 0 \quad 1 \quad 0] \begin{bmatrix} 0 & 1 & 1 & -1 \\ 1 & 0 & 1 & -1 \\ 1 & 1 & 0 & -1 \\ -1 & -1 & -1 & 0 \end{bmatrix} = [1 \quad 1 \quad 0 \quad -1]$$

$$y(1) = sign(S(0)) = [1 \quad 1 \quad 0 \quad 0]$$

Повторим операцию для полученного выходного образа:

$$S(1) = Y'(1)W = [1 \quad 1 \quad 0 \quad 0] \begin{bmatrix} 0 & 1 & 1 & -1 \\ 1 & 0 & 1 & -1 \\ 1 & 1 & 0 & -1 \\ -1 & -1 & -1 & 0 \end{bmatrix} = [1 \quad 1 \quad 2 \quad -2]$$

$$y(2) = sign(S(1)) = [1 \quad 1 \quad 1 \quad 0]$$

Как и в случае с асинхронным методом, сеть перестанет изменять свои значения после продолжения данных операций. Всего понадобилось 2 такта времени или 2 раза полностью обработать зашумленный вектор признаков.

Нейронная сеть Хэмминга

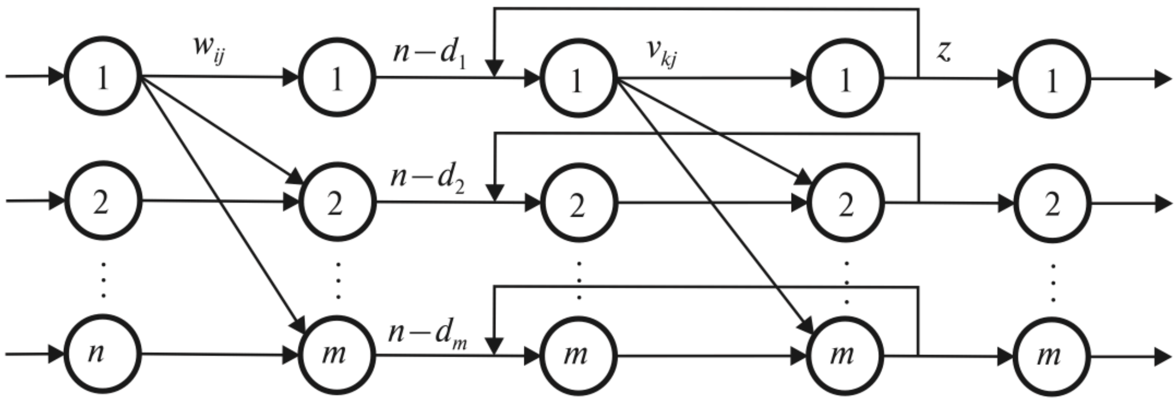

Рисунок 1 – Архитектура нейронной сети Хэмминга

Как можно заметить, архитектура данной сети основана на сети Хопфилда. Отличие заключается в наличии однойслойного персептрона выполняющего предварительную обработку данных перед отправкой на релаксационные слои.

$$w_{ij} = \frac{x_i^j}{2}, T_j = \frac{n}{2}, \text{где } i = [1,n], j = [1,m] \tag{2.1}$$

В таком случае, известная с лабораторной работы №1 формула вычисления выходного нейрона изменится следующим образом:

$$y_j = \sum_{i=1}^{n} w_{ij}x_i + T_j = \sum_{i=1}^{n} \frac{x_i^j}{2}x_i + \frac{n}{2} = \frac{1}{2}(\sum_{i=1}^{n} x_i^j x_i + n) \tag{2.2}$$

Весовые коэффициенты для сети Хопфилда:

$$z_j(t+1) = F(S_j(t))$$
$$S_j(t) = \sum_{k=1}^{m} v_{kj}z_k(t)$$

$$v_{kj} = \begin{cases} 1, & \text{если } k = j \\ -e, & \text{если } k \neq j \end{cases}$$

e = *const*, и, как правило, находится в следующем диапазоне:

$$0 < e < \frac{1}{m}$$

Составим итоговую формулу для вычисления взвешенной суммы для сети Хопфилда:

$$z_j(0) = y_j, j = [1,m]$$

$$S_j(t) = \sum_{k=1}^{m} v_{kj} z_k(t) = z_j(t) - e \sum_{k=1}^{m} z_k(t), \text{ где k} \neq \text{j} \qquad (2.3)$$

$$z_j(t+1) = F(S_j(t)) \begin{cases} S_j(t), & \text{если } S_j(t) > 0 \\ 0, & \text{если } S_j(t) \leq 0 \end{cases}$$

Преобразуем данное выражение с учетом того, что при k = j, $v_{ji} = 1$:

$$z_j(t+1) = F(S_j(t)) = \begin{cases} S_j(t), & \text{если } S_j(t) > 0 \\ 0, & \text{если } S_j(t) \leq 0 \end{cases}$$

Для сети Хэмминга релаксационный процесс происходит до тех пор, пока только один нейронный элемент сети Хопфилда не останется с положительной активностью. Номер данного нейрона-победителя соответствует номеру распознанного образа. Число образов, хранимых в сети, соответствует числу входных и выходных элементов сети Хопфилда.

Пример использования НС Хэмминга для решения задачи корректировки числовой последовательности:

1. Задаем исходные числовые вектора или векторы образов:

$$y_1 = [1 \quad 0 \quad 0 \quad 0]$$

$$y_2 = [0 \quad 1 \quad 1 \quad 0]$$

2. Настраиваем весовые коэффициенты на данный входной вектор:

$$w_{ij} = \frac{x_i^j}{2}, T_j = \frac{n}{2}$$

$$w_{ij} = \begin{bmatrix} 0.5 & 0 & 0 & 0 \\ 0 & 0.5 & 0.5 & 0 \end{bmatrix}$$

$$T_j = 2$$

3. Для зашумленного образа y' вычисляем выходное значение 1-го слоя:

$$y' = [1 \quad 0 \quad 0 \quad 1]$$

$$y_1^1 = \sum [0.5 \quad 0 \quad 0 \quad 0] * [1 \quad 0 \quad 0 \quad 1] + 2 = 0.5 + 2 = 2.5$$

$$y_2^1 = (\sum [0 \quad 0.5 \quad 0.5 \quad 0] * [1 \quad 0 \quad 0 \quad 1] + 2) = 0 + 2 = 2$$

4. Находим выходные значения для сети Хопфилда:

$$z_j(0) = y_j$$

$$0 < e < \frac{1}{2}$$

$$e = 0.3$$

$$S_j(t) = z_j(t) - e\sum_{k=1}^{m} z_k(t), k \neq j$$

Поскольку есть всего 2 выходных нейрона получаем следующее:

$S_1(0) = z_1(0) - ez_2(0) = 2.5 - 0.3 * 2 = 2.5 - 0.6 = 1.9$
$S_2(0) = z_2(0) - ez_1(0) = 2 - 0.3 * 2.5 = 1.25$
$z_1(1) = F(S_1(t)) = 1.9$, т.к. $1.9 > 0$
$z_2(1) = F(S_2(t)) = 1.25$, т.к. $1.25 > 0$

5. Повторяем пока не останется 1 нейрон с положительным значением:

$S_1(1) = z_1(1) - ez_2(1) = 1.9 - 0.3 * 1.25 = 1.525$
$S_2(1) = z_2(1) - ez_1(1) = 1.25 - 0.3 * 1.9 = 0.68$
$S_1(2) = z_1(2) - ez_2(2) = 1.525 - 0.3 * 0.68 = 1.32$
$S_2(2) = z_2(2) - ez_1(2) = 0.68 - 0.3 * 1.525 = 0.22$
$S_1(3) = z_1(3) - ez_2(3) = 1.32 - 0.3 * 0.22 = 1.254$
$S_2(3) = z_2(3) - ez_1(3) = 0.22 - 0.3 * 1.32 = -0.176$
$z_1(5) = F(S_1(4)) = 1$, т.к. $1.254 > 0$
$z_2(5) = F(S_2(4)) = 0$, т.к. $-0.176 < 0$

Следовательно, для образа $y' = [1 \quad 0 \quad 0 \quad 1]$ нейронная сеть Хэмминга предложила 1-й нейрон победитель, который соответствует образу $y_1 = [1 \quad 0 \quad 0 \quad 0]$ что и является ожидаемым результатом.

Двунаправленная ассоциативная память

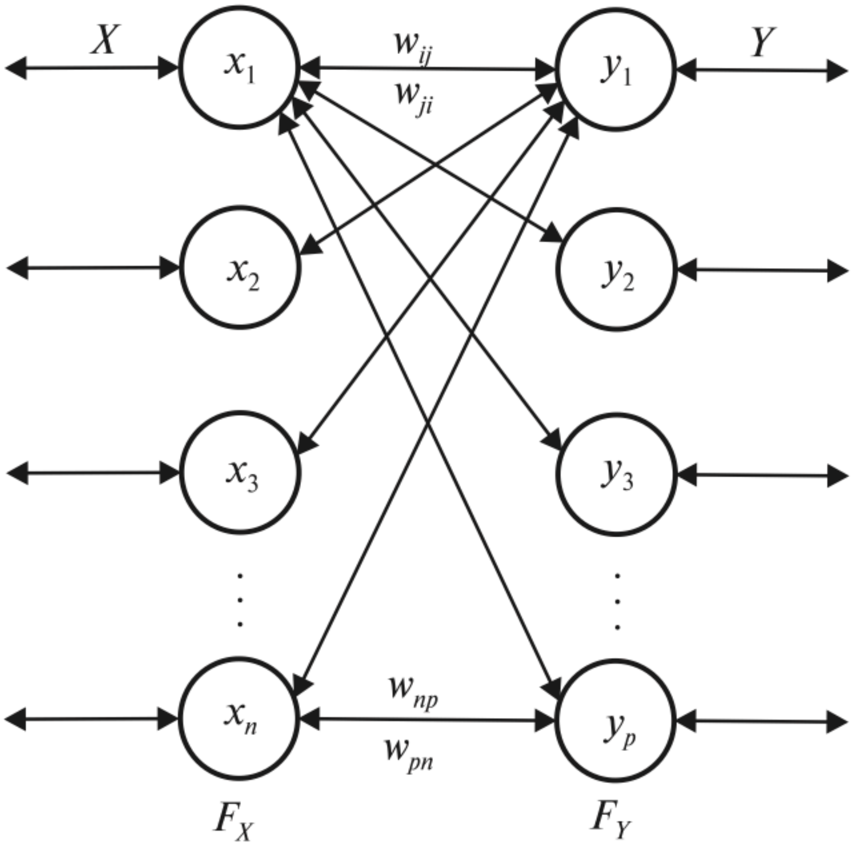

Рисунок 3 - Архитектура двунаправленной ассоциативной памяти

Работа данной сети напоминает работу сети Хопфилда. Её отличительной особенностью является возможность работать одновременно с обоих сторон - при подаче слева направо x преобразуется в y и наоборот, при подаче y преобразуется в x. Весовые коэффициенты такой сети определяются следующим образом:

$$W_{ij} = \sum_{k=1}^{L} x_i^k y_j^k \tag{3.1}$$

В матричной форме это будет выглядеть следующим образом:

$$W = X^T Y$$

Формула для вычисления взвешенной суммы:

$$S_j = \sum_{i=1}^{n} x_i w_{ij} \tag{3.2}$$

Функция активации взвешенной суммы:

$$y_j(t+1) = F(S_j) = y_j(t+1) = F(S_j) = \begin{cases} 1, & \text{если } S_j > 0 \\ y_j(t), & \text{если } S_j = 0 \\ -1, & \text{если } S_j < 0 \end{cases}$$

$$S_i = \sum_{j=1}^{n} y_j w'_{ji} \tag{3.3}$$

$$x_i(t+1) = F(S_i) = y_j(t+1) = F(S_i) = \begin{cases} 1, & \text{если } S_i > 0 \\ x_i(t), & \text{если } S_i = 0 \\ -1, & \text{если } S_i < 0 \end{cases}$$

В матричной форме выходные значения определяются следующим образом:

$$Y = F(XW)$$

$$X = F(YW^T)$$

Пример использования двунаправленной ассоциативной памяти для решения задачи корректировки числовой последовательности:

1. Задаем исходные числовые вектора или векторы образов:

$$X_1 = [-1 \quad 1 \quad -1], Y_1 = [-1 \quad 1 \quad -1 \quad -1]$$

$$X_2 = [1 \quad -1 \quad 1], Y_2 = [1 \quad -1 \quad 1 \quad 1]$$

2. Вычисляем весовые коэффициенты:

$$W = X^T Y = \begin{bmatrix} -1 & 1 \\ 1 & -1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} -1 & 1 & -1 & -1 \\ 1 & -1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 2 & -2 & 2 & 2 \\ -2 & 2 & -2 & -2 \\ 2 & -2 & 2 & 2 \end{bmatrix}$$

3. Вычислим выходные значения для оригинальных образов образов Y', X':

$$S^Y = X_1 W = \begin{bmatrix} -1 \\ 1 \\ -1 \end{bmatrix} \begin{bmatrix} 2 & -2 & 2 & 2 \\ -2 & 2 & -2 & -2 \\ 2 & -2 & 2 & 2 \end{bmatrix} = \begin{bmatrix} -2 & 2 & -2 & -2 \\ -2 & 2 & -2 & -2 \\ -2 & 2 & -2 & -2 \end{bmatrix} = [-6 \quad 6 \quad -6 \quad -6]$$

$$Y = F(S^Y) = [-1 \quad 1 \quad -1 \quad -1]$$

$$S^X = Y_1 W^T = [-1 \quad 1 \quad -1 \quad -1] \begin{bmatrix} 2 & -2 & 2 & 2 \\ -2 & 2 & -2 & -2 \\ 2 & -2 & 2 & 2 \end{bmatrix} = \begin{bmatrix} -2 & -2 & -2 & -2 \\ 2 & 2 & 2 & 2 \\ -2 & -2 & -2 & -2 \end{bmatrix} = [-8 \quad 8 \quad -8]$$

$$X = F(S^X) = [-1 \quad 1 \quad -1]$$

Для входного значения X_1 получилось выходное значение Y_1, для X_2 получился Y_2. Следовательно оригинальные вектора были распознаны верно.

4. Проверка распознавания зашумленного образа:

$$X_1 = [-1 \quad -1 \quad -1]$$

$$S^Y = X_1 W = \begin{bmatrix} -1 \\ -1 \\ -1 \end{bmatrix} \begin{bmatrix} 2 & -2 & 2 & 2 \\ -2 & 2 & -2 & -2 \\ 2 & -2 & 2 & 2 \end{bmatrix} = \begin{bmatrix} -2 & 2 & -2 & -2 \\ 2 & -2 & 2 & 2 \\ -2 & 2 & -2 & -2 \end{bmatrix} = [-2 \quad 2 \quad -2 \quad -2]$$

$$Y = F(S^Y) = [-1 \quad 1 \quad -1 \quad -1]$$

Полученное значение Y соответствует значению Y_1, что является парным значением для X_1. В пункте №3 приведен пример вычисления оригинального значения X_1 при подаче на него вектора Y_1. Следовательно зашумленный вектор был распознан верно.

Порядок выполнения работы.

Задача лабораторной работы – провести исследование полученной модели. При этом на вход сети необходимо подавать искаженные образы, в которых инвертированы некоторые биты. Критерий эффективности процесса распознавания - максимальное кодовое расстояние (количество искаженных битов) между исходным и поданным образом.

Оформить отчет, содержащий:
1. Титульный лист
2. Цель работы
3. Задание
4. Текст программы
5. Результаты работы программы
6. Выводы по лабораторной работе

Таблица 1. Варианты заданий

Вариант	n	m	№ векторов
1	10	10	6,8,1,10,3
2	11	4	1,8,2,11
3	12	8	2,8,3
4	13	13	3,8,4,9
5	14	4	4,8,5,9
6	15	5	5,9,6
7	16	16	6,8,7,5
8	17	3	3,8,1
9	18	4	4,10,2
10	19	19	5,8,3
11	20	3	6,8,4
12	10	10	7,8,5,12

Примечание:
1. Для сети Хопфилда реализовать синхронный и асинхронный режимы функционирования.
2. Для двунаправленной ассоциативной памяти первые n элементов берем за X, последние m элементов заданного вектора берем за Y.

Таблица 2. Набор векторов

№	Данные вектора																			
1	0	1	0	0	1	1	0	1	0	0	0	0	1	0	1	0	1	0	0	0
2	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0
3	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
4	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0
5	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
6	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1
8	1	1	1	0	0	0	1	1	1	0	0	0	1	1	1	0	0	0	1	1
9	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0
10	0	0	0	1	1	1	0	0	0	1	1	1	0	0	0	1	1	1	0	0
11	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0
12	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1

Ожидаемые результаты:

Сеть Хопфилда:

1. Source vectors:

   y1 = [1 0 0 1]
   y2 = …
   …
   yk = …

2. Async example for y1 # need for every input vector

   y_original = [1 0 0 1]

   Stage 1:
   y_model (1) = [(1) 0 0 1]
   y_model (2) = [1 (0) 0 1]
   y_model (3) = [1 0 (0) 1]
   y_model (4) = [1 0 0 (1)]

   y_stage_1 == y_original -> relaxation, correct # may be incorrect if not, or no relaxation if model can’t find relaxation (for example 10 is max tries)

3. Sync example for y1 # need for every input vector

y_original = [1 0 0 1]

Stage 1:
y_model (1) = [1 0 0 1] # at first time moment

y_stage_1 == y_original -> relaxation, correct # may be incorrect if not, or no relaxation if model can’t find relaxation (for example 10 is max tries)

4. Maximum number of recognised noisy bits:
   Async:
      y_1 = 4 # example number
      y_2 = ..
      y_k = ..
   Sync:
      y_1 = 4 # example number
      y_2 = ..
      y_k = ..

   **Проверяется максимально возможное число искаженных бит для каждого образа, которое может распознать модель (для обоих режимов).*

Сеть Хэмминга

1. Source vectors:

   y1 = [1 0 0 1]
   y2 = …
   yk = …

2. Example for y1 # need for every input vector

   y_original = [1 0 0 1]
   winner (1) = [1.5 0.5 0.5 ..] # after activate
   winner (2) = [1 0 0 ..] # after activate, only 1 neuron > 0, all other will be < 0
   y_model (2) = [1 0 0 1]
   y_model (2) == y_original -> correct

3. Maximum number of recognised noisy bits:

   y_1 = 4 # example number
   y_2 = ..
   y_k = ..

   **Продолжать до тех пор, пока не будет найдено максимальное число бит, которое можно восстановить для каждого образа.*

Двунаправленная ассоциативная память:

1. Source vectors:

    x1 = [0 1 0]; y1 = [1 0 0 1]
    x2 = …; y2 = …
    …
    xk = … ; yk = …

2. Example for x1 # need for every x-input vector

    x_original = [0 1 0]

    Stage 1:
    y_model (1) = [1 0 0 1] # at first time moment
    x_model (1) = [0 1 0]
    x_model (1) == x_original -> relaxation, correct

3. Example for y1 # need for every y-input vector

    y_original = [1 0 0 1]

    Stage 1:
    x_model (1) = [0 1 0] # at first time moment
    y_model (1) = [1 0 0 1]
    y_model (1) = y_original -> relaxation, correct

4. Maximum number of recognised noisy bits:

    y_1 = 4 # example number
    y_2 = ..
    y_k = ..

    x_1 = 5 # example number
    x_2 = ..
    x_k = ..