

Учреждение образования
«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ»

Кафедра интеллектуальных информационных технологий

Отчет по лабораторной работе №3
по курсу «Методы решения задач в интеллектуальных системах»
на тему: «Релаксационные нейронные сети»

Выполнил студент
группы 421702:

Бруцкий Д.С.

Проверил:

Ивашенко В.П.

Минск, 2016

Цель

Ознакомиться, проанализировать и получить навыки реализации модели релаксационной нейронной сети для задачи распознавания образов.

Реализовать модель нейронной сети Хемминга.

Используемые обозначения

n — размер запоминаемого образа

m — количество запоминаемых образов

sequence — числовая последовательность

step — коэффициент обучения

iterations — количество итераций обучения

Описание модели

Сеть состоит из двух слоев. Первый и второй слои имеют по m нейронов.

Нейроны первого слоя имеют по n синапсов, соединенных со входами сети (образующими фиктивный нулевой слой). Нейроны второго слоя связаны между собой ингибиторными (отрицательными обратными) синаптическими связями.

Единственный синапс с положительной обратной связью для каждого нейрона соединен с его же аксоном.

Реализация

Модель реализована на языке программирования java. Для проведения операций с матрицами используется библиотека jama.

Алгоритм функционирования сети

```
private Matrix startWork(Matrix matrix) {
    iterations = 0;
    Matrix oldAxons = matrix.times(inputWeights);
    oldAxons.plusEquals(new Matrix(1, layersSize, t));
    oldAxons = activationFunction(oldAxons);
    for (Matrix newAxons = null; newAxons == null || hemmingDistance(newAxons,
oldAxons) != 0; ) {
        iterations++;
        if(newAxons != null) oldAxons = newAxons;
        newAxons = oldAxons.copy();
        Matrix correction = newAxons.times(workingWeights);
        newAxons = newAxons.plusEquals(correction);
        newAxons = activationFunction(newAxons);
    }
    return oldAxons;
}
```

Для оценки стабилизации решения вычисляется расстояние Хемминга.

```
private int hemmingDistance(Matrix newAxons, Matrix oldAxons) {
    int distance = 0;
    for (int i = 0; i < newAxons.getColumnDimension(); i++) {
        if (abs(newAxons.get(0, i) - oldAxons.get(0, i)) > 0.000001) distance++;
    }
    return distance;
}
```

В сеть было загружено 7 псевдографических образов размером 6*6

,, ,#, , ,, #, #, , #, , , # , #, , , # , ##### , #, , , #	,, , , , , ,, , , , , ,, , , , , ,, , , , , ,, , , , , ,, , , , ,	#, , , , # ##, , , # #, #, , , # #, #, #, # #, , , ## #, , , , #	, #####, , #, , , # , #, , , # , #####, , #, , , , , #, , , , , #, , , ,	,, ##, , , #, , , #, ,, #, , , ,, #, , , ,, #, , , , #, , , #, ,, ##, ,	##### ,, , , , , ##### ,, , , , , ##### ,, , , , , ,, , , , ,	##### ,, , , , #, ,, , , #, , ,, #, , , , , #, , , , , , #, , , , , #####
---	--	---	--	---	---	---

Затем на вход сети подавались зашумленные образы

Входной образ	Образ распознанный сетью	Количество итераций для достижения релаксации
,, ,#, , ,, #, #, , #, , , , , #, , , # , #, , ## , #, , , #	,, ,#, , ,, #, #, , #, , , # , #, , , # , ##### , #, , , #	1021
,, , , , , ,, , , , , ,, , , , , ,, ##, , ,, #, , , ,, , , , ,	,, , , , , ,, , , , , ,, , , , , ,, , , , , ,, , , , , ,, , , , ,	1021
#, , , , # ##, , , # ,, #, , # ,, #, #, ,, , , ## ,, , , , #	#, , , , # ##, , , # #, #, , # #, #, #, #, , , ## #, , , , #	1021
, #####, , #, , , # , #, , , # , ##, #, , #, , , , , #, , , ,	, #####, , #, , , # , #, , , # , #####, , #, , , , , #, , , ,	1022
,, ##, , , #, , , #, ,, , , , , ,, , , , , , #, , , #, ,, ##, ,	,, ##, , , #, , , #, ,, #, , , ,, #, , , , #, , , #, , #, , , #, ,, ##, ,	1022

На вход сети подавались инвертированные образы

Входной образ	Образ распознанный сетью	Количество итераций, необходимых для достижения релаксации
###,## ##,## #,###, #,###, #,,,, #,###,	##### ,,,,#, ,,,##, ,,#,,, ,##### #####	1050
##### ##### ##### ##### ##### #####	#,,,,# ##,,,# #,##,## #,,##, #,,##, #,,##, #,,##,	1024
,####, ,,###, ,#,##, ,##,##, ,###,, ,####,	,,,,,, ,,,,,, ,,,,,, ,,,,,, ,,,,,, ,,,,,, ,,,,,,	1049
#,,,,# #,###, #,###, #,,,,# #,#### #,####	##### ,,,,#, ,,,##, ,,#,,, ,##### #####	1024
##,,## #,##,## ##,### ###,## #,##,## ##,,##	,,,,,, ,,,,,, ,,,,,, ,,,,,, ,,,,,, ,,,,,, ,,,,,,	1022

В данном опыте сеть обучалась на уменьшенных образах размером 5x5

Образ	,,##, ,#,##, #,,,,# ##### #,,,,#	,,,,,, ,,,,,, ,,,,,, ,,,,,, ,,,,,,	#,,,,# ##,,,# #,##,## #,,##, #,,,,#	####, #,,,,# #,,,,# ####, #,,,,#	##### ,,,,,, ##### ,,,,,, #####	##### ,,##, ,,##, ,,##, #####
Итерации до релаксации	1022	1022	1022	1022	1022	1022

Вывод

Сеть Хемминга распознает все образы, на которых она обучена. Сеть Хемминга распознает инвертированные обучающие образы, как образы, имеющее наименьшее расстояние Хемминга до распознаваемого образа.

Сеть достигает состояния релаксации при распознавании различных образов за близкое количество итераций. При этом количество итераций не зависит от размера образов, на которых обучается сеть.