

## Распознавание зашумленных монохромных изображений посредством нейронной сети Хопфилда

*Е.Д. Пойманова, Т.М. Татарникова*

*Санкт-Петербургский государственный университет аэрокосмического  
приборостроения*

**Аннотация:** Обсуждается актуальность применения нейронной сети Хопфилда для распознавания простых монохромных изображений небольшого объема в условиях их зашумленности. Приводится архитектура нейронной сети Хопфилда и алгоритм обучения. Демонстрируются примеры и результаты распознавания зашумленных монохромных изображений на примере дорожных знаков.

**Ключевые слова:** распознавание образов, рекуррентная нейронная сеть, зашумленное монохромное изображение, эталонный образец, обучение.

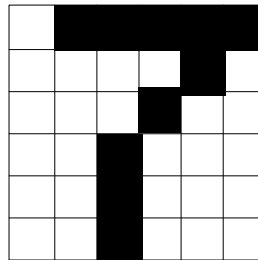
### Введение

Распознавание образов является одной из частных задач машинного обучения, для решения которой привлекаются искусственные нейронные сети [1]. В силу особенностей различных архитектур нейронных сетей можно распознавать цветные и монохромные изображения, простые и сложные объекты на изображении и многое другое [2].

В данной работе исследуется возможность применения рекуррентной нейронной сети Хопфилда для распознавания зашумленных монохромных изображений типа отдельных символов, отпечатков пальцев, отдельных несложных объектов на изображении. Зашумленность в практическом смысле означает отсутствие, размытость или искажение части изображения, но при этом неизменной части изображения достаточно для распознавания образа, а фактически его восстановления. В качестве примера зашумленного изображения может быть налипание грязи на дорожные знаки или автомобильные номера.

Для кодирования монохромного изображения используется квадратная матрица пикселей, размерность которой равна размеру изображения [3,4]. Например, в известном наборе данных MNIST, содержащем 70 000 примеров

рукописных цифр, изображение любой цифры от 0 до 9 представляется матрицей размером  $28 \times 28$  пикселей. В ячейку пиксельной матрицы записывается 1 при попадании в нее элемента изображения или 0 в противном случае. Пример кодирования изображения цифры семь показан на рис. 1.



(0,1,1,1,1,1,0,0,0,0,1,0,0,0,0,1,0,0,0,0,1,0,0,0,0,0,1,0,0,0,0,0,1,0,0,0)

Рис. 1. – Монохромное изображение цифры Семь

### Архитектура нейронной сети Хопфилда

Нейронная сеть Хопфилда относится к классу рекуррентных, поскольку в ее архитектуре присутствуют обратные связи между выходами и входами различных нейронов. Наличие обратных связей приводит к тому, что изменение состояния всей нейронной сети ставится в зависимость от состояния одного нейрона. Таким образом, функционирование нейронной сети Хопфилда представляет собой переходный процесс от одного устойчивого состояния  $S(t)$  к другому устойчивому состоянию  $S(t+1)$  в результате возбуждения одного или нескольких нейронов [5].

Нейроны сети Хопфилда могут находиться в одном из двух состояний:  $S(t) \in \{-1; 1\}$ : возбужденному состоянию соответствует (+1), остановке (-1). Таким образом, ступенчатая функция активация нейрона обусловлена дискретностью его состояний [6].

Уравнение переходного процесса для  $i$ -го нейрона:

$$S_i(t+1) = F \left( \sum_{j=1}^N w_{ij} S_j(t) \right), \quad i, j = \overline{1, N}, \quad (1)$$

где  $w_{ij}$  – значение весового коэффициента обратной связи входа  $i$ -го нейрона с выходом  $j$ -го нейрона;  $N$  – размерность нейронной сети.

Случай спокойного состояния нейрона (остановки) не рассматривается.

Общий вид архитектуры нейронной сети Хопфилда показан на рис. 2 [7].

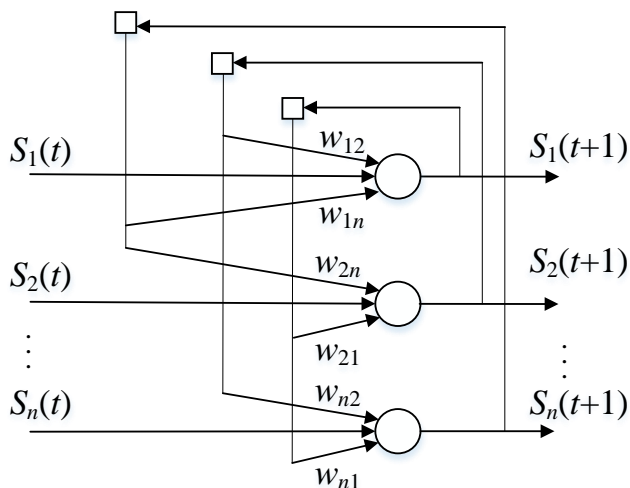


Рис. 2. – Архитектура нейронной сети Хопфилда

### Обучение нейронной сети Хопфилда

Рассмотрим процесс обучения нейронной сети Хопфилда распознавать образы. Пусть  $\mathbf{X}$  – множество распознаваемых образов,  $M$  – мощность множества  $\mathbf{X}$ , тогда распознаваемый образ  $\bar{x}_k$  представляет собой:

$$\bar{x}_k = x_k + \xi, \quad (2)$$

где  $\xi$  – шум, наложенный на исходный образ  $x_k$ .

Уравнение (2) задает цель обучения – определение нормы  $\delta$  в сопоставлении эталонного образа с распознаваемым образом, то есть  $|\bar{x}_k - x_k| < \delta$ .

Размерность нейронной сети Хопфилда зависит от количества эталонных образов. Выражение (3) позволяет оценить количество нейронов сети:

$$N = \frac{M}{\alpha}, \quad (3)$$

где  $M$  – количество хранимых эталонных образов;  $\alpha \leq 0,14$  – гиперпараметр нейронной сети Хопфилда [8].

Эталонный образ хранится в памяти нейронной сети Хопфилда в виде матрицы весовых коэффициентов  $\mathbf{W}$ . Уравнение (4) задает правило вычисления элементов матрицы  $\mathbf{W}$ , известное также, как правило Хебба:

$$w_{ij} = \frac{1}{N} \sum_{k=1}^M (x_{ik} \cdot x_{jk}), \quad (4)$$

где  $x_{jk}$  –  $j$ -й элемент изображения  $k$ -го образа.

Подача исходного образа на вход нейронной сети Хопфилда задает начальные условия и запускает переходный процесс:

$$S_i(0) = \bar{x}_k. \quad (5)$$

Таким образом, выражения (1) –(5) полностью описывают процесс функционирования нейронной сети Хопфилда по распознаванию образов.

### Эксперимент на модели нейронной сети Хопфилда

Эксперимент на модели нейронной сети Хопфилда заключается, во-первых, в обучении распознавать дорожные знаки и, во-вторых, распознавании (восстановлении) зашумленных образов с оценкой допустимого уровня зашумленности. В качестве эталонных образов рассматриваются: Въезд запрещен, Уступите дорогу, Парковка.

Каждый из знаков представим в виде матрицы размером  $13 \times 13$ , заполненной значениями (+1), либо (–1).

На рис. 3 приведена визуализация созданных дорожных знаков [9].

Создадим зашумленные образы дорожных знаков имеющих дефект в изображении. Для этого к каждому пикселю в изображении были добавлены случайные значения. На рис. 4 приведены примеры зашумленных образов.

```
number_patterns = 3      #Количество распознаваемых образов
pattern_height = 13     #Высота одного образа
pattern_width = 13      #Ширина одного образа

> patterns = np.array([...
visual(patterns, number_patterns, pattern_height, pattern_width)

✓ 0.3s
```

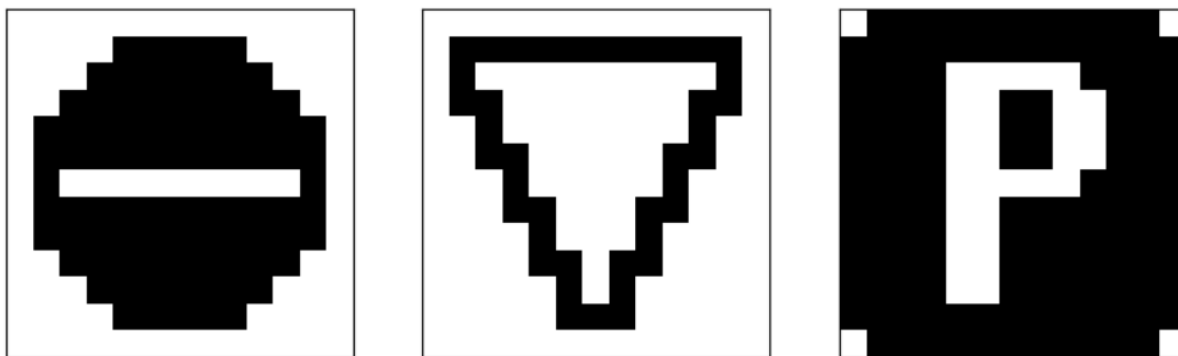


Рис. 3. – Исходные дорожные знаки

```
test = create_noise(patterns, 1.5)      #тестовый набор данных
for i in range(len(test)):
    print(f'Величина ошибки {i+1}-го рисунка: {error_counter(test[i], patterns[i])}')
visual(test, number_patterns, pattern_height, pattern_width)

✓ 0.3s
```

Величина ошибки 1-го рисунка: 0.1301775147928994  
Величина ошибки 2-го рисунка: 0.10650887573964497  
Величина ошибки 3-го рисунка: 0.21301775147928995

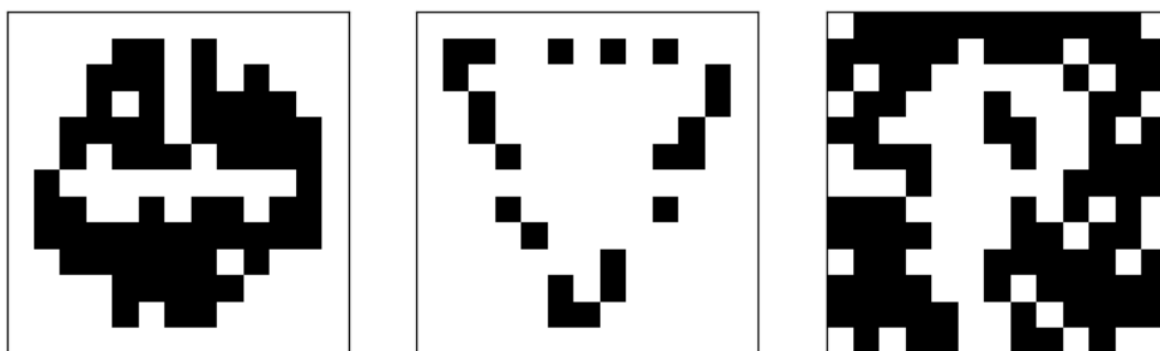


Рис. 4. – Искаженные образы

Результат восстановления зашумленных образов продемонстрирован на рис. 5. Как видно из рис. 5, восстановленные дорожные знаки ничем не отличаются от эталонов. Отсюда можно сделать вывод о том, что при величине изменений в 1.5 единицы нейронная сеть отлично восстанавливает образы.

```
errors, results = testing(weights, test, number_patterns, patterns)
for i in range(len(errors)):
    print(f'Итоговая величина ошибки {i+1}-го рисунка: {errors[i]}')

visual(results, number_patterns, pattern_height, pattern_width)
✓ 0.2s
```

Итоговая величина ошибки 1-го рисунка: 0.0  
Итоговая величина ошибки 2-го рисунка: 0.0  
Итоговая величина ошибки 3-го рисунка: 0.0

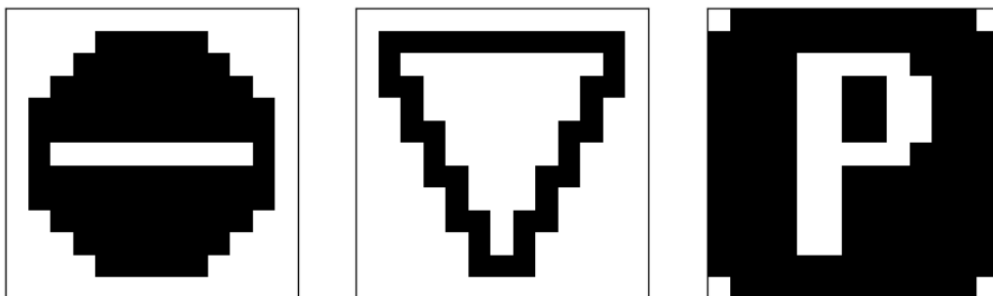
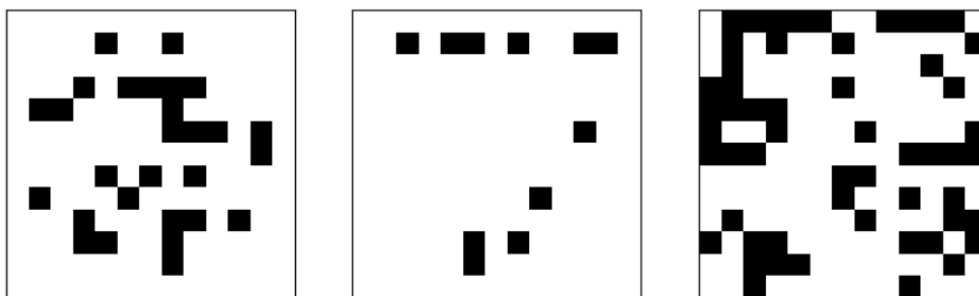


Рис. 5. – Результат распознавания образов

Далее уменьшим значение данного параметра до 0.9 и оценим результаты. На рис. 6 приведена визуализация результата работы нейронной сети Хопфилда по распознаванию (восстановлению) образов.



Итоговая величина ошибки 1-го рисунка: 0.24260355029585798  
Итоговая величина ошибки 2-го рисунка: 0.0  
Итоговая величина ошибки 3-го рисунка: 0.4260355029585799

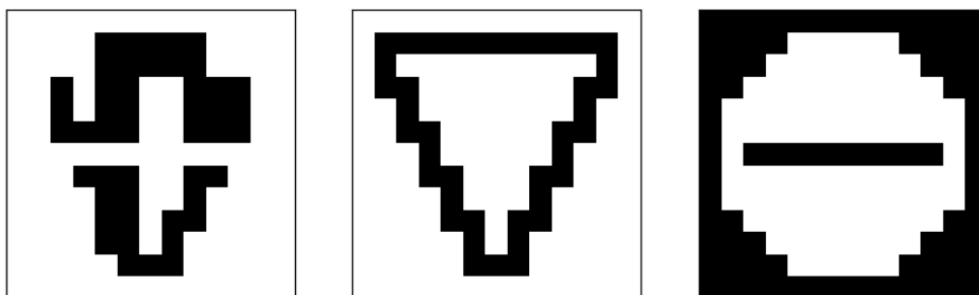


Рис. 6. – Распознавание сильного искажения

Как видно теперь (рис. 6), сеть справилась с распознаением и восстановлением одной из трех знаков картинок. Можно заметить, что третья восстановленная картинка является инвертированным первым дорожным знаком.

Далее можно добавлять и другие эталоны. На рис. 7 приведен фрагмент программного кода добавления новых эталонов, в частности знака Главная дорога, и их визуализации.

```
new = [-1, -1, -1, -1, -1, -1, 1, -1, -1, -1, -1, -1, -1,
      -1, -1, -1, -1, -1, 1, -1, 1, -1, -1, -1, -1, -1,
      -1, -1, -1, -1, 1, -1, 1, -1, 1, -1, -1, -1, -1,
      -1, -1, -1, 1, -1, 1, 1, 1, -1, 1, -1, -1, -1,
      -1, -1, 1, -1, 1, 1, 1, 1, 1, -1, 1, -1, -1,
      -1, 1, -1, 1, 1, 1, 1, 1, 1, 1, -1, 1, -1,
      1, -1, 1, 1, 1, 1, 1, 1, 1, 1, -1, 1, -1, 1,
      -1, 1, -1, 1, 1, 1, 1, 1, 1, 1, -1, 1, -1,
      -1, -1, 1, -1, 1, 1, 1, 1, -1, 1, -1, -1,
      -1, -1, -1, 1, -1, 1, 1, 1, -1, 1, -1, -1, -1,
      -1, -1, -1, -1, 1, -1, 1, -1, 1, -1, -1, -1, -1,
      -1, -1, -1, -1, -1, 1, -1, 1, -1, -1, -1, -1, -1,
      -1, -1, -1, -1, -1, -1, 1, -1, -1, -1, -1, -1, -1]

#Знак главная дорога

patterns = np.append(patterns, [new], axis=0)
number_patterns = 4
visual(patterns, number_patterns, pattern_height, pattern_width)
```

✓ 0.1s

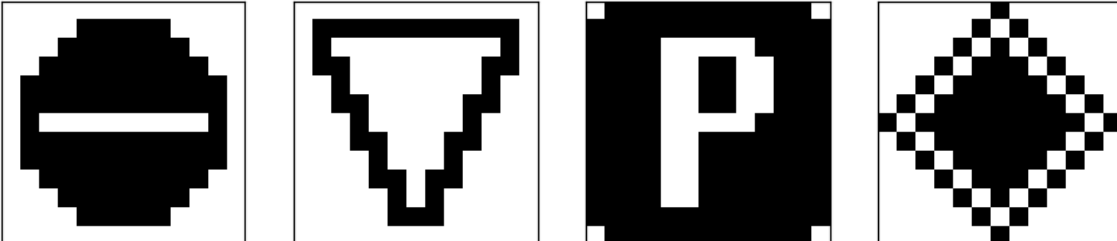


Рис. 7. – Добавление нового знака

На возможность распознавания образа с применением нейронной сети Хопфилда влияет не только зашумленность, но и смещение или поворот образа относительно эталонного образа. Очевидный выход из такой ситуации видится в создании коллекции эталонных образов с разными поворотами и смещениями. Однако, для архитектуры нейронной сети Хопфилда данная ситуация является неразрешимой – ограниченный объем памяти не позволяет

запоминать коллекции, в результате чего происходит «перекрытие» эталонных изображений образов [10].

### **Выводы**

В работе показано, что рекуррентная нейронная сеть с архитектурой Хопфилда может быть основой для встроенного программного обеспечения, например для систем автоматического контроля доступа на охраняемую территорию в случае распознавания автомобильных номеров или бортового компьютера при распознавании автомобильных номеров в режиме реального времени.

Результаты эксперимента на зашумленных изображениях демонстрируют возможность восстановления изображений при менее 40% искаженных битах.

Небольшой объем памяти нейронной сети Хопфилда обуславливает область ее применения – распознавание и восстановление небольших монохромных изображений.

### **Литература**

1. Гелиг А.Х. Введение в математическую теорию обучаемых распознающих систем и нейронных сетей. М.: Издательство СПбГУ, 2017. 224 с.
2. Каллан Р. Нейронные сети: Краткий справочник. М.: Вильямс, 2017. 288 с.
3. Самойлова И. А. Технологии обработки больших данных // Молодой ученый. 2017. № 49 (183). С. 26-28.
4. Кабаков Р. И. R в действии. Анализ и визуализация данных в программе R. – М.: ДМК Пресс, 2016. 588 с.
5. Созыкин А.В. Обзор методов обучения глубоких нейронных сетей // Вестник ЮУрГУ, Серия: Вычислительная математика и информатика. 2017.



Т.6. № 3. С. 28-59.

6. Tatarnikova T.M. Restricting data leakage through non-obvious features of Android 5 smartphone // Informationno-Upravliaiushchie Sistemy. 2019. Vol. 5. P. 25-29.

7. Ertam F., Aydin G. Data Classification with Deep Learning Using Tensorflow // International Conference on Computer Science and Engineering. 2017. P. 755-758.

8. Крысова И.В. Экспертная система для автоматизации процесса классификации деталей по ЕСКД // Инженерный вестник Дона, 2013, №4. URL: ivdon.ru/ru/magazine/archive/n4y2013/1888

9. Татарникова Т.М., Пойманова Е.Д., Богданов П.Ю., Краева Е.В., Вережкин С.А. Программные среды для изучения основ нейронных сетей // Программные продукты и системы. 2021. № 1. С. 145-150.

10. Сергеев А.М. О совмещении изображений и способах их реализации // Инженерный вестник Дона, 2022, № 8. URL: ivdon.ru/ru/magazine/archive/n8y2022/7832.

### References

1. Gelig A.KH. Vvedeniye v matematicheskuyu teoriyu obuchayemykh raspoznayushchikh sistem i neyronnykh setey [Introduction to the Mathematical Theory of Learning Recognition Systems and Neural Networks]. M.: SPbGU, 2017. 224 p.

2. Kallan R. Neyronnyye seti: Kratkiy spravochnik [Neural Networks: A Brief Handbook]. M.: Williams, 2017. 288 p.

3. Samoylov I. A. Young scientist. 2017. № 49 (183). pp. 26-28.

4. Kabakov R. I. R v deystvii. Analiz i vizualizatsiya dannykh v programme R [R in action. Data analysis and visualization in the R program]. M.: DMK Press, 2016. 588 p.

5. Sozykin A.V. Vestnik Yuzhno-Ural'skogo gosudarstvennogo universiteta.



Seriya: Vychislitel'naya matematika i programmaya inzheneriya. 2017. Vol.6. № 3. pp. 28-59.

6. Tatarnikova T.M. Informationno-Upravliaiushchie Sistemy. 2019. Vol. 5. pp. 25-29.

7. Ertam F., Aydin G. International Conference on Computer Science and Engineering, 2017, pp. 755-758.

8. Krisova I.V. Inzhenernyj vestnik Dona, 2013, №4. URL: [ivdon.ru/ru/magazine/archive/n4y2013/1888](http://ivdon.ru/ru/magazine/archive/n4y2013/1888)

9. Tatarnikova T.M., Poymanova E.D., Bogdanov P. Yu., Kraeva E.V., Verevkin C.A. Software & Systems. 2021. № 1. pp. 145-150.

10. Sergeev A.M. Inzhenernyj vestnik Dona, 2022, № 8. URL: [ivdon.ru/ru/magazine/archive/n8y2022/7832](http://ivdon.ru/ru/magazine/archive/n8y2022/7832).

**Дата поступления: 14.10.2024**

**Дата публикации: 21.11.2024**