

1 Введение

В современном мире тема искусственных нейронных сетей (ИНС) приобретает всё большую популярность благодаря способности нейронных сетей сравнительно легко адаптироваться к требованиям различных практических приложений и конечно же возможности обучаться. Интеллектуальные системы на основе ИНС позволяют с успехом решать проблемы распознавания образов, выполнения прогнозов, оптимизации, ассоциативной памяти и управления. Известны и иные, более традиционные подходы к решению этих проблем, однако они не обладают необходимой гибкостью за пределами ограниченных условий. ИНС дают многообещающие альтернативные решения, и многие приложения выигрывают от их использования. Но в то же время тема ИНС сравнительно мало изучена и в настоящее время стремительно развивается.

Для исследования я выбрал сеть Хопфилда. Эта сеть не требует процесса обучения и полностью задаётся входными данными. В качестве практического приложения я решал проблему распознавания образов, т.к. она наиболее проста и наглядна, и имеет важное практическое применение. Простая реализация сети (называемая синхронной) не может привести нас к стабильному результату, а точная реализация асинхронной требует либо параллельных вычислений, либо очень медленно работает. Поэтому я исследовал возможность приближенной асинхронной реализации, которая описана ниже.

2 Сеть Хопфилда

Нейронная сеть Хопфилда — полносвязная нейронная сеть с симметричной матрицей связей. В процессе работы динамика таких сетей сходится к одному из положений равновесия. Сеть может быть использована как фильтр. В отличие от многих нейронных сетей, функционирующих до получения ответа через определённое количество тактов, сети Хопфилда функционируют до достижения равновесия, когда следующее состояние сети в точности равно предыдущему, начальное состояние является входным образом, а при равновесии получают выходной образ.

Я использовал сеть в качестве фильтра для восстановления повреждённых образов.

2.1 Работа сети

Сеть Хопфильда задается матрицей W - матрицей весовых коэффициентов. Данная матрица определяется следующим образом

$$w_{ij} = \frac{1}{N} \sum_{k=0}^p x_i^{(k)} x_j^{(k)},$$

где $x^{(k)}$ - обучающие образцы. Для устойчивости главную диагональ матрицы зануляем.

При обучении получили матрицу W , которая задает сеть.

Пусть есть зашумленный образец v_0 . Последовательно применяем к нему матрицу весовых коэффициентов следующим образом

$$v_1 = F(Wv_0) \tag{1}$$

$$v_2 = F(Wv_1) \tag{2}$$

$$\dots \tag{3}$$

$$v_{i+1} = F(Wv_i), \tag{4}$$

где $F(x) = \text{sign}(x)$ — функция активации.

В результате получаем последовательность векторов $\{v_n\}$, которая сходится к вектору v за конечное число шагов. Этот вектор v и есть результат фильтрации.

Детальное описание работы сети можно найти в работах Хайкена С. и Таркова М. С. ([1], [2]) или в сети Интернет ([3]).

2.1.1 Синхронный режим работы сети

Описанный выше способ называется синхронным режимом работы сети. При этом последовательно просматриваются нейроны, однако их состояния запоминаются отдельно и не меняются до тех пор, пока не будут пройдены все нейроны сети. При синхронном режиме возможно (хотя и далеко не всегда наблюдается) бесконечное чередование двух состояний - так называемый динамический аттрактор. Поэтому синхронный режим практически для сети Хопфилда не используется, и рассматривается лишь как основа для понимания более сложного асинхронного режима.

2.1.2 Асинхронный режим работы сети

Если моделировать работу сети как последовательный алгоритм, то в асинхронном режиме работы состояния нейронов в следующий момент времени меняются последовательно: при вычислении состояния нейрона мы учитываем уже изменённое состояние других нейронов. В асинхронном режиме невозможен динамический аттрактор - вне зависимости от количества запомненных образов и начального состояния, сеть непременно придет к устойчивому состоянию (статическому аттрактору).

2.2 Реализация работы сети

При моделировании работы сети использовался математический пакет Maple. В качестве входных образцов были использованы эталонные изображения букв русского алфавита. Бинарные образцы букв и распознаваемый символ считывались из файлов. Количество запоминаемых образов варьировалось. Распознавание считалось завершенным при достижении аттрактора (статического или динамического).

Асинхронный режим обеспечивался методом, предложенным д.ф.-м.н., проф. МЭИ(ТУ) М. Н. Кирсановым. Случайным образом формируется множество индексов (меньше размерности вектора) и в векторе изменяются только элементы с индексами из этого множества. Т.к. этот режим не является чисто асинхронным режимом, есть вероятность возникновения динамического аттрактора, а также получения различных результатов фильтрации. Данная работа заключается в исследовании этого режима.

3 Эксперимент

Я исследовал бинарные образцы букв, задаваемые матрицей размера 10×12 . То есть размерность образцов 120, откуда максимальная ёмкость сети, высчитанная по формуле

$$M = \frac{N}{2 \log_2 N}$$

не превосходит 8. Основное исследование проводим с 8 образцами (максимальная ёмкость), которые приведены на рис. 1. Но для эксперимента попробуем задать и большее количество образцов.



Рис. 1: Образцы букв

Обозначим число индексов, заменяемых на каждом такте через K .

3.1 Случайное задание индексов

Первоначально мы K раз случайно определяли индексы, а не все множества. В результате множество уникальных индексов, генерируемых на каждом такте, оказалось меньше, чем число K , причем различие тем больше, чем больше K . Эта разница приведена на графике (рис. 2). По оси ординат откладывалось значение K , по оси абсцисс реальное количество уникальных индексов, а по оси аппликат вероятность получения данного количества уникальных индексов.

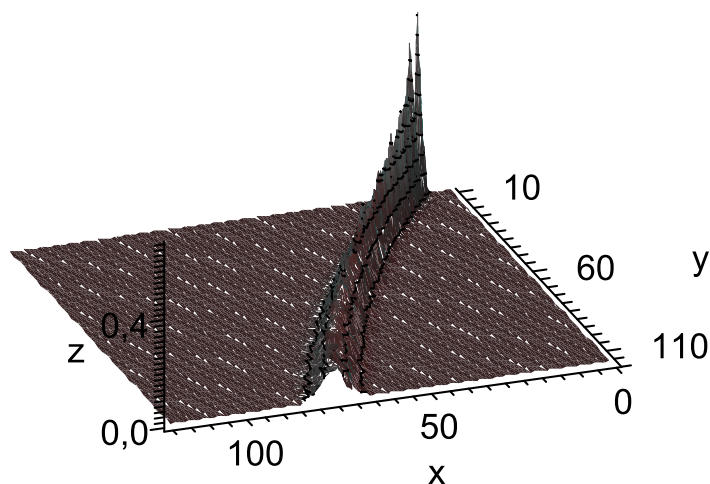


Рис. 2: График вероятности получения данного количества уникальных индексов при заданом K

Из графика видно, что получение уникального множества индексов большего 70 почти невозможное событие. Кроме того, зависимость наиболее вероятного количества индексов от K нелинейное. В общем, использование этого метода для реального исследования не представляется возможным.

3.2 Случайное задание множества индексов

При данной реализации мы строили случайно множество индексов заданного размера. Таким образом, на каждом такте у нас заменяется ровно K элементов вектора, и мы получили точную реализацию. Как мы написали ранее, случайно задание индексов позволяет исследовать поведение системы при количестве заменяемых элементов не более 70 из 120. Данным же методом мы исследуем весь диапазон. Данный способ является основным для данной работы.

Основной эксперимент заключается в варьировании числа K , и проведении большого числа опытов при каждом K для набора статистики. Для каждого эксперимента мы создавали зашумленный образец разной степени зашумленности.

Каждый опыт проводился при фиксированном K . Достижением аттрактора считался момент, когда следующий вектор равен предыдущему. Если же возникал динамический аттрактор, опыт завершался при достижении достаточно большого числа тактов (заведомо большего среднего ожидаемого числа), и считался ошибкой фильтрации (такие вылеты при необходимости можно легко фиксировать).

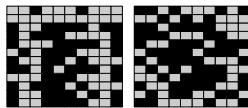


Рис. 3: Зашумленные буквы

Исследовали 2 зашумленных образца (рис. 3), с малой и большой степенью зашумленности. Число K бралось от 10 до 120 с шагом, равным 10. Для каждого K делалось 25 попыток распознать образ (из которого мы генерировали зашумленный образ). Кругочком обозначаются удачные завершения опыта (распознан предполагаемый образец), а горизонтальным крестом — ошибки.

Если рассматривать детально результаты первого эксперимента (при слабом зашумлении), то можно увидеть, что при 120 опыта заканчивался при достижении максимально допустимого числа тактов, то есть, вероятно, это возникновение ложного аттрактора. Что вполне допустима, потому что при 120 у нас получается синхронная реализация.

В случае сильного зашумления мы получили почти 100% определение неверного образа. Но при сравнении с результатами на слабозашумленном образце, можно увидеть, что время достижения конечного результата (в данном случае ошибочного) почти не отличается от времени достижения результата на слабозашумленном образце. Откуда можно сделать вывод, что время нахождения результата не зависит от степени зашумленности.

Результаты фильтрации для слабозашумленного образца приведены на рис. 4, а для сильного зашумления 5.

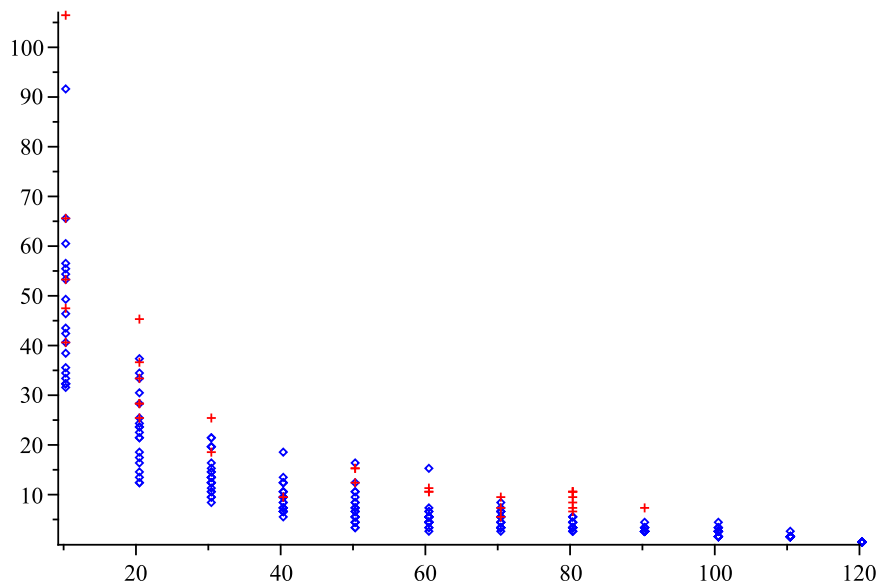


Рис. 4: Результаты для слабого зашумления

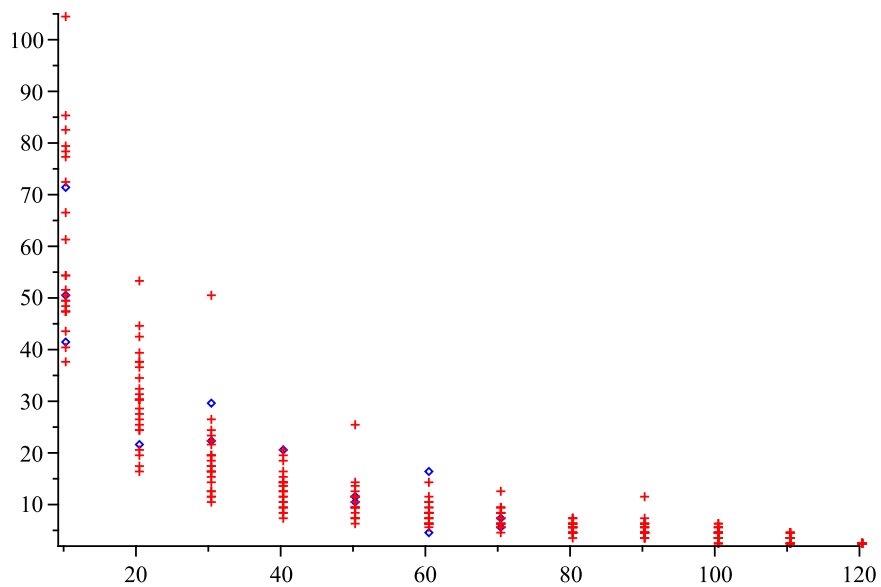


Рис. 5: Результаты для сильного зашумления

3.3 Переполнение

В данном случае мы попытались задать число образцов больше, чем вмещает сеть размером 120. В результате распознавание большинства образов стало невозможным: даже исходные образы часто искажались, в то время как они должны оставаться стационарными (то есть на каждом такте применения сети они должны остаться неизменными). А если исходные образы не являются стационарными точками сети, то и фильтрация невозможна.

Кроме того, проверка стационарности исходных образов - это хороший способ проверки правильности определения матрицы весовых коэффициентов.

4 Результаты

В работе установлены следующие закономерности:

1. Установлено, что отношение времени достижения ложного результата к времени достижения истинного постоянно для одинаковой степени зашумленности.
2. При увеличении степени синхронности сети скорость сходимости увеличивается. При этом при некоторой степени синхронности это увеличение практически незаметно.
3. Скорость сходимости не зависит от степени зашумленности.

Список литературы

- [1] Тарков М.С. *Нейрокомпьютерные системы*. Бином, М., 2006.
- [2] Хайкин С. *Нейронные сети*. Вильямс, М., 2006.
- [3] Нейронная сеть Хопфилда. 2009.