МИНОБРНАУКИ РФ

ФГБОУ ВПО Тверской государственный технический университет

Кафедра «Программное обеспечение».

Дисциплина «Системы ИИ».

Тема: «Искусственная нейронная сет. Неокогнитрон для распознавания символов».

Выполнил: студент группы

ПИН 17.05

Иванов Р.В

Проверил:

Мальков А.А

Тверь 2021

Оглавление

[Цели 3](#_Toc71544908)

[ВВЕДЕНИЕ 3](#_Toc71544909)

[ПРЕДЛАГАЕМЫЙ ПОДХОД 4](#_Toc71544910)

[Результаты 11](#_Toc71544911)

[Заключение 13](#_Toc71544912)

[Источники 13](#_Toc71544913)

# Цели

Искусственные нейронные сети уже много лет используются для классификации объектов и распознавания образов.

В данной работе исследуется использование самоорганизующейся нейронной сети, называемой неокогнитрон, для распознавания символов. Система должна успешно распознавать символы несмотря на различные варианты их написания и наличия пустого пространства. Для этого исследования будет создан неокогнитрон с использованием Java, обусловлено это объектно-ориентированной природой искусственной нейронной сети. Эта сеть будет обучена распознавать пять символы и затем будет протестирована с использование этих же символов, но с различным количеством отступов и вариантов их написания.

# ВВЕДЕНИЕ

Неокогнитрон — это самоорганизующаяся нейронная сеть, которая используется при визуальном распознавании образов. Обычно письменные символы различаются не только по размеру и отступам, но и по стилю написания, который уникален для каждого человека.

Эта проблема вызывает потребность в хорошей системе распознавания, которая может обрабатывать эти вариации. Неокогнитрон предоставляет систему, которая устойчива к изменениям внешнего вида символов и их пространственному расположению.

Неокогнитрон — это самоорганизующаяся нейронная сеть. это означает, что характерные признаки для какого-либо символа определяются в процессе обучения

Используя иерархическую систему, в которой характерные признаки извлекаются в пределах первых нескольких слоев, последующие слои могут обеспечить распознавание символов. Такая система способна научиться распознавать много разных типов символов.

Множество ранее созданных нейронной сети для распознавания символов не удавалось распознать символы, которые были искажены по форме или содержат сдвиги в позиции. Кроме того, неокогнитрон был разработан, чтобы имитировать функциональность человеческого мозга в том, как он распознает некоторые визуальные паттерны.

Для большинства констант уже есть оптимальные начальные значения, но некоторых весов они неизвестны. В этой работе будет исследована способность Неокогнитрон для распознавания символов. Математика и структура нейронной сети будет рассмотрена в следующем разделе, а затем будет предложен тестовый набор данных. Наконец, полученная нейронная сеть будет протестирована с тестовым набором данных; это показано в разделе Результаты.

# ПРЕДЛАГАЕМЫЙ ПОДХОД

Сложная структура неокогнитрона позволяет извлекать признаки из изображения независимо от сдвигов в положении. Эта структура определяется организацией весов и связей сети. Чтобы лучше визуализировать структуру неокогнитрона, сначала необходимо понять структуру сети и используемую терминологию. Входным сигналом является квадратное изображение с низким разрешением, для этой реализации это будет изображение 16 на 16 данного символа. Каждый слой состоит из двух «подслоев»; например, слой 1 состоит из s-слоя и c-слоя. Внутри каждого слоя есть несколько s-плоскостей или c-плоскостей. Блок-схема структуры высокого уровня представлена на рис. 1.

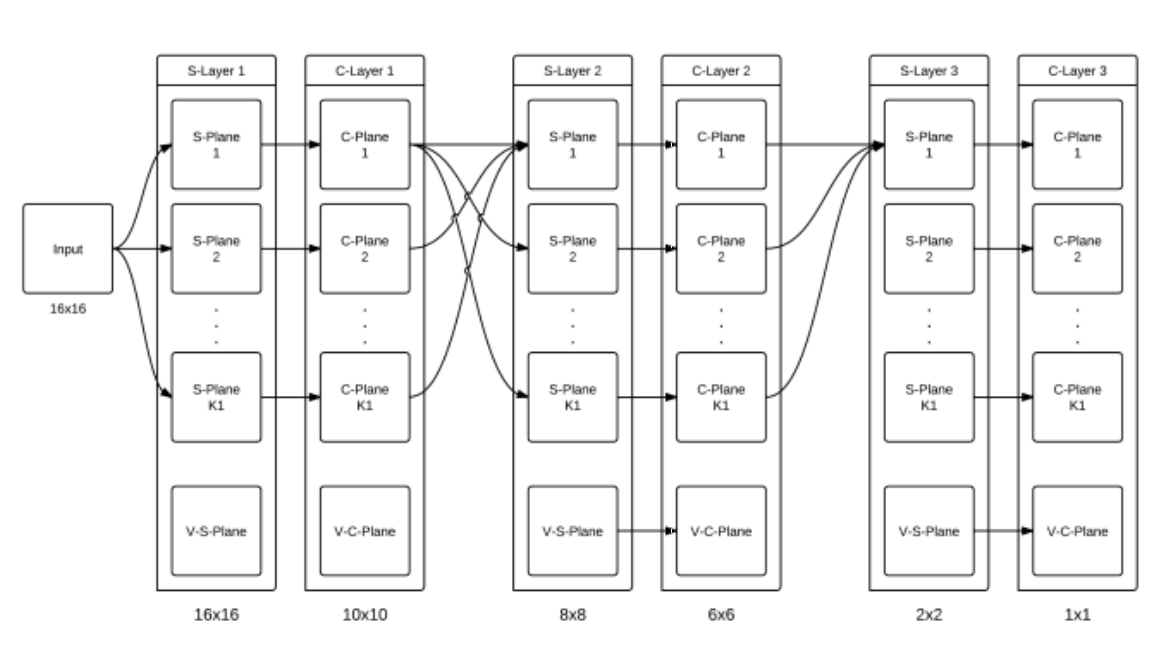


Рисунок 1: Блок-схема верхнего уровня неокогнитрона

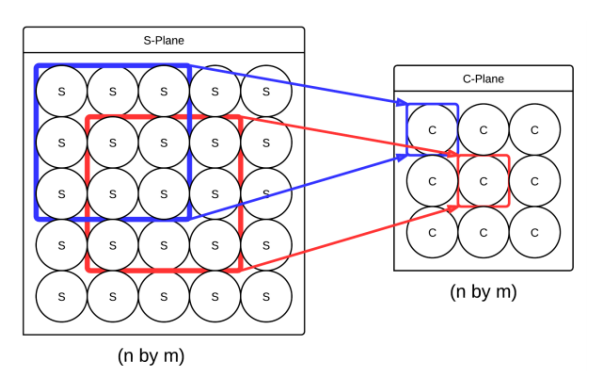


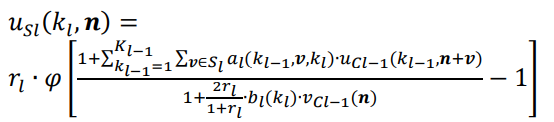
Рис. 2: Организация каждой плоскости, показывающей окно межслойного соединения.

Каждая плоскость клеток внутри каждого слоя представляет собой матрицу нейронов; тип нейрона зависит от того, в каком слое он находится. Размер плоскости обычно квадратный, как показано на рисунке. Рис. 2, где n равно m (5 на 5). Каждая плоскость в данном s-слое или c-слое имеет одинаковый размер. Некоторые соединения показанные на рис. 1, не они нарисованы. Каждая s-плоскость связана только с предшествующей s-плоскостью; каждая с-плоскость связана с каждой предшествующей с-плоскостью.

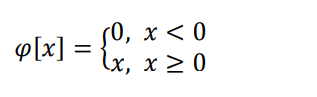
Внутри каждого слоя также существует одна v-плоскость, независимо от того, сколько существует s-плоскостей или c-плоскостей. То цель каждой v-клетки-обеспечить ингибирующую реакцию на s-клетку или c-клетку в каждой плоскости. Из-за того, что существует только одна v-плоскость на слой, каждое место в каждой s-плоскости или c-плоскости получает один и тот же ингибирующий вход. Связь каждой ячейки можно увидеть на рисунке 3.

Для этой реализации будут использоваться три слоя с размером каждого слоя, показанным на рис. 1. Важно отметить, что последний с-слой имеет размер плоскости 1 на 1. После завершения обучения сеть будет обрабатывать заданные входные данные, а результатом будет большой файл, состоящий из одной плоскости в последнем слое, который состоит из одной ячейки. Таким образом, в последнем с-столбце должно быть по крайней мере столько плоскостей, сколько символов должно быть распознано.

Прежде чем структура будет определена, будут даны уравнения, характеризующие каждый тип ячейки. Этот математическая основа обеспечит основу для того, чтобы увидеть, как каждый слой соединяется с другими слоями.



(1)



(2)

Два вышеперечисленных уравнения представляют собой выходные данные от s-клетки. Результат, , это выходные данные из l – слоя, из плоскости в точке n, где n – двумерный вектор, представляющий расположение ячейки в определенной плоскости. Это уравнение зависит от четырех значений, некоторые из которых являются векторами и матричными значениями.

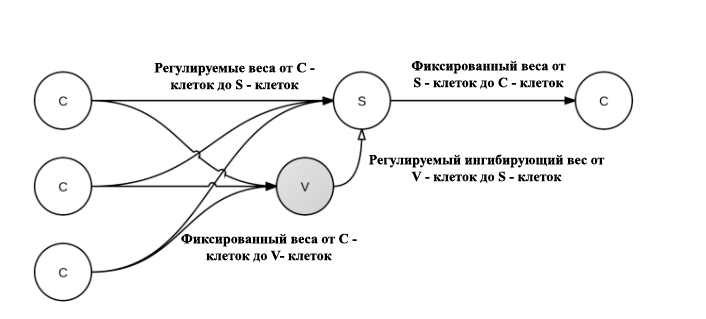


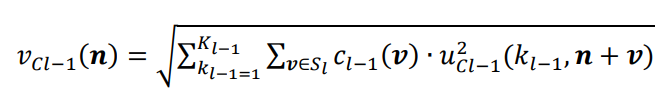
Рис. 3: Связи между каждым типом клеток; показан ингибирующий ответ v-клетки.

Входной сигнал , представляет собой сигнал, поступающий с предыдущего слоя . Конкретный вход отличается для каждой плоскости входного слоя, . Входные данные также зависят от местоположения **n** в этой конкретной плоскости. Диапазон входных данных, используемых для этой ячейки, определяется окном ячеек , которое центрируется вокруг определенной точки **n**. Это можно визуализировать на рис. 2, каждый блок входных окон, , центрирован вокруг другого.

Входные веса изменяются от плоскости к плоскости, они также изменяются в зависимости от того, из какой плоскости поступает вход в предыдущем слое. Важно отметить, что входные веса не зависят от конкретного местоположения в плоскости. Таким образом, внутри каждой плоскости веса одинаковы для любого заданного местоположения; это то, что позволяет извлекать объекты независимо от пространственного распределения.

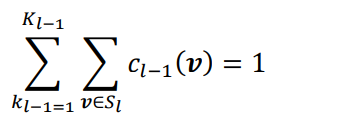
Важно также отметить, что это уравнение не допускает отрицательных значений. Уравнение 2 приводит выход к нулю для любых отрицательных выходных значений. Кроме того, используется для масштабирования выходного сигнала при положительном значении. Существует различное значение для каждого слоя. Конечное значение, на которое опирается выход s-ячейки, находится в плоскости v-ячейки. Существует только одна плоскость v-ячейки на слой s-ячейки; таким образом, выходное значение v-ячейки зависит только от местоположения в плоскости, **n**.

Выход v-ячейки затем взвешивается на одно значение это значение отличается для каждой плоскости в пределах данного слоя, но не зависит от местоположения. Оба веса и веса изменяются по мере обучения сети. Именно эти значения определяют, насколько хорошо сеть извлекает и распознает определенные паттерны.



(3)

Приведенное выше уравнение показывает, как определяется выход каждой v-ячейки в v-плоскости. Это уравнение очень похоже на числитель в уравнении 1. Весовая матрица, , умножается на выходной квадрат предыдущего c-слоя,. Весовая матрица зависит только от окна. Они одинаковы для каждой входной плоскости и каждого места в одной v-плоскости.

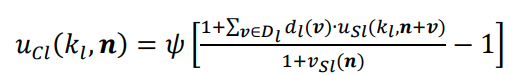


(4)

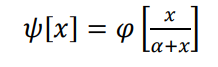
Эта весовая матрица отличается от других весов в этой сети тем, что суммирование всей матрицы для каждой плоскости в предыдущем слое должно быть меньше 1. Поскольку выход из каждой c-ячейки всегда будет меньше 1, что показано далее в этом разделе, выход каждой v-ячейки также будет меньше единицы, когда будет выполнено приведенное выше уравнение. Кроме того, выход v-ячейки, по сути, является взвешенным среднеквадратичным значением (RMS) выхода предыдущего c-слоя для конкретного местоположения n.

Начальные значения для каждого веса будут обсуждаться в разделе Результаты. Только весовые матрицы и обновляются во время обучения; веса и не обновляются, они остаются постоянными после инициализации. На этом завершается математика, необходимая для реализации s-слоя.

Для каждого s-слоя существует соответствующий c-слой, который содержит такое же количество плоскостей. Каждая плоскость в слое c получает входные данные только от одной плоскости в предыдущем слое s. Это отличается от s-плоскостей, которые получают свои входные сигналы от каждой плоскости в предыдущем слое c.



(5)

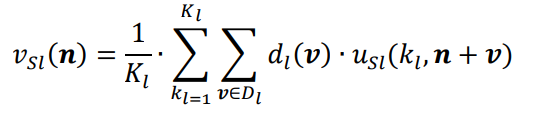


(6)

Уравнения 5 и 6 показывают, как математически определяется выход каждой с-ячейки в с-слое. Выход из предыдущего s-слоя, , умножается на весовую матрицу d. Каждая c-ячейка в данной плоскости получает вход только от s-ячеек в предыдущем слое в той же плоскости. Таким образом, нет суммирования по нескольким плоскостям для определенной позиции.

Весовая матрица d(v) зависит только от окна, содержащегося в , а не от того, в какой плоскости находится c-ячейка, ни от местоположения внутри конкретной c-плоскости. Таким образом, существует один набор весов для каждого c-слоя; по сравнению с s-слоями, которые содержат другой набор весов для плоскости входов и выходов.

Как и в случае с s-клеткой, каждая c-клетка получает ингибирующий сигнал от соответствующей v-клетки. Для каждого с-слоя существует только одна v-плоскость. Для каждой плоскости существует определенная v-ячейка для каждого местоположения; выход v-ячейки определяется с помощью уравнения 7.



(7)

Для v-ячеек в слое c используется то же весовое значение, что и для каждой c-ячейки, Как уже упоминалось ранее, начальные значения для каждого будут обсуждаться в разделе результатов. В отличие от с-плоскостей в с-слое, v-плоскость в с-слое получает входные сигналы от каждой плоскости в предыдущем s-слое. На этом заканчивается математика, необходимая для распространения сигнала по сети.

Для того чтобы сеть научилась определенным паттернам и самоорганизовалась, она должна быть обучена. Первая версия неокогнитрона, предложенная Фукусимой, использует неконтролируемое обучение. Во время тренировки изменяются только два веса: и

Для обучения сети ячейки для каждого слоя должны определяться каждый раз при загрузке изображения. Эта методика обучения является подходом “победитель получает все”. Клетки с наибольшим выходом усиливаются, в то время как другие клетки остаются в покое. Этот подход позволяет каждой клеточной плоскости самоорганизоваться так, чтобы она распознала только одну конкретную особенность из своих входных данных.

Поскольку будут обучаться только веса s-слоя, репрезентативные ячейки будут определяться только внутри каждого s-слоя. Каждая s-плоскость в обучаемом слое укладывается на одна поверх другой, как колода карт. Теперь существует стопка s-ячеек для каждой позиции. Используя окно вокруг каждой стопки s-ячеек, аналогичное показанному на рис. 2, создается столбец s. Например, если используемое окно имеет размер 3 пикселя на 3 пикселя и имеет 8 плоскостей, то каждый s-столбец “слой” будет 3 на 3 s-ячейки с общим количеством 8 “слоев”.

Этот s-столбец генерируется для каждого местоположения пикселя; таким образом, данная s-ячейка может быть частью нескольких s-столбцов. Внутри каждого s-столбца в качестве репрезентативных выбираются плоскость и расположение s-ячейки с наибольшим выходом. Поскольку существует несколько s-столбцов, возможно, что в пределах данной плоскости выбирается более одной репрезентативной ячейки; в этом случае выбирается та, которая имеет наибольший выход, так что на каждую плоскость выбирается не более одной ячейки.

Для каждой плоскости, содержащей репрезентативную ячейку, , веса a и b увеличиваются на величину, диктуемую уравнениями 8 и 9. Величина увеличения каждой ячейки определяется с помощью входного сигнала в позиции . Увеличивая эти значения, каждая плоскость распознает отдельный объект.



(8)



(9)

Для каждого символа, который распространяется сетью во время обучения, веса каждого s-слоя обновляются с помощью метода, описанного выше. Как для уравнения 8, так и для уравнения 9 константа определяет, как быстро сеть будет обучаться. Для каждого s-слоя используется разное значение . На этом завершается теоретическая основа, необходимая для реализации неокогнитрона.

# Результаты

С помощью Java был построен неокогнитрон. Этот неокогнитрон может принимать входное изображение размером 16 на 16 пикселей. Всего существует три слоя, как показано на рисунке 1; каждый слой содержит одинаковое количество плоскостей. Первый s-слой имеет разрешение ячеек 16 на 16 пикселей, а первый c-слой имеет разрешение 10 на 10 пикселей. Второй s-слой имеет разрешение 8 на 8 пикселей, а второй c-слой имеет разрешение 6 на 6 пикселей. Последний s-слой имеет разрешение 2 на 2 пикселя, а последний c-слой содержит только один пиксель на плоскость.

Весовые константы d и c определялись с помощью уравнений 10 и 11. Важно, чтобы оба веса монотонно убывали; используя значения как для δ, так и для γ, которые меньше единицы, это требование будет выполнено.



(10)

(11)

Количество плоскостей на слой и все константы, используемые для конечного неокогнитрона, были определены экспериментально путем циклического перебора ряда возможностей для определения наилучших значений. Оптимальные значения, которые были определены с использованием этого подхода, показаны на рис. 4.

Все веса b инициализируются нулем, независимо от слоя; и все веса a инициализируются случайным значением, не превышающим 0,5. Константа α была установлена равной 0,478, а число плоскостей в каждом слое-16. Теперь, когда все необходимые начальные значения определены, неокогнитрон можно обучить с помощью набора тестовых символов

Символы, показанные на рис. 5, использовались для тестирования и верификации неокогнитрона. Поскольку конечная скорость распознавания определяется начальными значениями случайно сгенерированных весов , создается, обучается и тестируется множество неокогнитрона; тот, который содержит наибольшую скорость распознавания, сохраняется для дальнейшего использования.

Четвертое число каждой последовательности, показанной на рис. 5, используется для обучения неокогнитрона; все числа используются для определения частоты ошибок сети. Неокогнитрон, созданный с использованием значений, показанных на рис. 4, смог реализовать скорость распознавания 73,3%. Из двадцати символов на рисунке 5 сеть не смогла распознать четыре. Три из четырех непризнанных символов были номером один; в то время как последний непризнанный символ был первым номером четыре, показанным на рисунке 5.

Из-за того, что число один является очень простым символом и что оно не имеет многих особенностей, объясняется, почему оно не было распознано неокогнитроном. Несмотря на многочисленные различия между верификационными и обучающими символами, неокогнитрон работал очень хорошо. Несмотря на дополнительные пиксели и изменения размера и формы каждого числа, неокогнитрон успешно распознал большой процент проверочных символов.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Слой 1 | Слой 2 | Слой 3 |
|  | 4.81 | 1.44 | 2.57 |
|  | 0.20 | 9.60 | 13.94 |
|  | 0.11 | 0.42 | 0.06 |
|  | 0.49 | 0.87 | 0.52 |
|  | 0.12 | 0.06 | 0.78 |

Рис. 4: Постоянные значения, используемые для каждого слоя

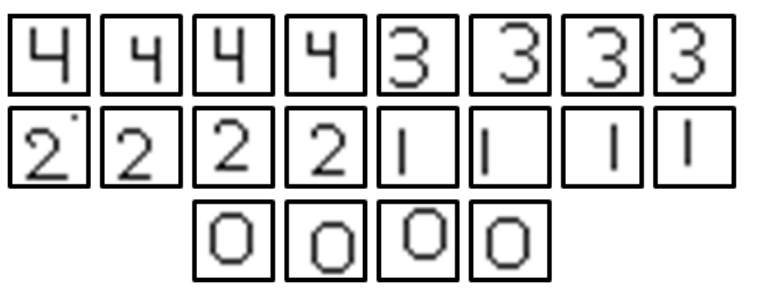


Рисунок 5: Символы, используемые для обучения и проверки

Значения, показанные на рис. 4, соответствуют многим значениям, используемым в исходной сети Фукусимы. Значение первого слоя является наибольшим, значение увеличивается в каждом слое, а константа α составляет приблизительно 0.5. Кроме того, количество плоскостей по крайней мере в два раза больше, чем количество символов, которые может распознать сеть.

Этот неокогнитрон может быть обучен распознавать до шестнадцати различных символов. Если необходимо распознать более шестнадцати символов, то количество плоскостей также должно быть увеличено. В целом сеть была очень успешной в распознавании числа от нуля до четырех несмотря на то, что символы были масштабированы и смещены пространстве.

# Заключение

Неокогнитрон идеально подходит для визуального распознавания образов. Структура и теория, лежащие в основе сети, предполагают надежный метод распознавания символов. Несмотря на это, реализация неокогнитрона, предложенного Фукусимой, не является прямолинейной. Есть много значений, которые должны быть установлены идеально, чтобы распознавание образов было успешным. Путем тестирования сети с большим количеством вариаций этих значений был найден оптимальный набор параметров. Как только такие значения были найдены, обученная сеть смогла распознать набор тестов. Символы, которые отличались от тренировочного набора, с 73,3% успеха. Дальнейшие исследования могли бы проанализировать способность сетей распознавать весь английский алфавит. Таким образом, неокогнитрон может быть успешно обучен распознавать символы, несмотря на изменения в положении.

# Источники

[1] K. Fukushima, "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position," Biological Cybernetics, vol. 36, pp. 193-202, 1980.

[2] C. N. S. Ganesh Murthy and Y. V. Venkatesh, "Modified neocognitron for improved 2-D pattern recognition," Vision, Image and Signal Processing, IEE Proceedings -, vol. 143, pp. 31-40, 1996.

[3] Zhengjun Pan; Sabisch, T.; Adams, R.; Bolouri, H.; , "Staged training of Neocognitron by evolutionary algorithms," Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on , vol.3, no., pp.3 vol. (xxxvii+2348), 1999