Цель работы – использование самоорганизующихся карт Кохонена для восстановления изображений.

Самоорганизующиеся карты – это одна из разновидностей нейросетевых алгоритмов. Основным отличием данной технологии от рассмотренных нами ранее нейросетей, обучаемых по алгоритму обратного распространения, является то, что при обучении используется метод обучения без учителя, то есть результат обучения зависит только от структуры входных данных.

SOM состоит из набора нейронов, количество которых задается аналитиком. Нейрон описывается двумя векторами – вектор веса, размерность которого обычно совпадает с размерностью вектора входных данных, и вектора координат, который определяет местоположение нейрона на карте.

Обычно нейроны располагаются в узлах двумерной сетки с прямоугольными или шестиугольными ячейками. При этом, как было сказано выше, нейроны также взаимодействуют друг с другом. Величина этого взаимодействия определяется расстоянием между нейронами на карте. На рисунке 1 дан пример расстояния для шестиугольной и четырехугольной сеток.

В основном карты Кохонена используют для решения таких задач как моделирование, прогнозирование, поиск закономерностей в больших массивах данных, выявление наборов независимых признаков, сжатие информации, классификация входных данных.

Мерой соседства между нейронами на карте называют функцию от номера итерации и индексов нейронов. Обычно в этих целях используют функцию Гаусса вида:

(1.1),

где *k1* и *k2* – индексы нейронов, *i* – номер итерации, имеет вид (1.2), а имеет вид (1.3), имеет вид (1.4).

(1.2),

(1.3),

(1.4).

Функция (2) приведена к такому виду для равномерной инициализации карты.

Мера сходства – это функция, которая показывает насколько один нейрон «похож» на другой. Другая мера сходства – евклидово расстояние между векторами *x* и *y*, определяется следующим выражением:

На первом этапе строится вектор входных данных на основе исходного изображения. Ранее в работе был выбран внутренний формат этого вектора, который будет использоваться в модели.

Следующий этап – это поиск лучшего нейрона. Для этого находятся расстояния до всех векторов весов нейронов карты, после чего выбирается наиболее похожий нейрон. Возможна ситуация, особенно на первых итерациях алгоритма, когда вышеуказанному условию соответствует несколько нейронов. В этом случае BMU выбирается случайным образом.

Далее определяются все нейроны, которые находятся рядом с BMU, в какой-либо определенной заранее окрестности. После этого происходит изменение векторов весов нейронов по формуле:

где – функция вида (1).

В качестве критерия остановки алгоритма обучения используется количество итераций цикла, которое задается перед началом обучения. Однако на этапе увеличения счетчика итераций так же проверяется количество нейронов, веса которых были изменены за одну итерацию. Если изменения карты незначительно малы, это означает что процесс обучения сети можно остановить.

Другой возможной альтернативой критерия остановки являлась ошибка карты, которая может вычисляться различными способами, к примеру, как среднее арифметическое расстояние между наблюдениями и векторами веса соответствующих ему BMU, но использование такого рода критерия необходимо при использовании более сложных мер соседства и сходства, либо при более сложном формате входных данных. В данном случае вычисления довольно простые, поэтому было решено использовать первый способ.

**2.2 Алгоритм восстановления изображения**

Этот алгоритм также является итерационным. Количество итераций определяется количеством поврежденных пикселей на изображении. При восстановлении из набора векторов входных данных выбирается такой вектор, чтобы в нем содержался всего один поврежденный элемент.

На следующем этапе ищется BMU среди нейронов обученной ранее карты, однако метрика поиска в этом случае меняется:

Другими словами, при поиске наиболее подходящего нейрона не учитывается поврежденная компонента в векторе входных данных, а в векторе веса нейрона не учитывается компонента, которая находится на той же позиции в векторе веса, что и поврежденная в векторе входных данных.

После выбора BMU из его вектора веса берется необходимая компонента и вставляется в изображение вместо поврежденной.

Следует отметить, что после восстановления пикселя изображения, он учитывается при восстановлении других пикселей, которые находятся рядом с ним.

Процесс восстановления заканчивается после восстановления всех пикселей.

**3.1 Основные возможности**

В разрабатываемом приложении должны быть реализованы следующие основные функции:

1. приведение изображения к внутреннему формату программы;
2. обучение нейронной сети;
3. возможность сохранения состояния сети;
4. загрузка обученной сети;
5. обозначение поврежденной области;
6. восстановление изображения;
7. сохранение восстановленного изображения;
8. сбор различных параметров работы для проведения анализа результата.