МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»**

НЕЙРОННЫЕ СЕТИ. ОБУЧЕНИЕ БЕЗ УЧИТЕЛЯ И КЛАСТЕРИЗАЦИЯ ДАННЫХ

РЕФЕРАТ

студента 5 курса 531 группы
направления 10.05.01 — Компьютерная безопасность
факультета КНиИТ
Стаина Романа Игоревича

Проверил
доцент _____ И. И. Слеповичев

Саратов 2023

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Алгоритмы обучения с учителем нейронных сетей подразумевают наличие некоего внешнего звена, предоставляющего сети, кроме входных, также и целевые выходные образы. Для их успешного функционирования необходимо наличие экспертов, создающих на предварительном этапе для каждого входного образа эталонный выходной. Обучения без учителя, наоборот, не требует разметки данных. Система старается сама найти в них общие признаки и связи.

Нейронные сети, обученные без учителя, чаще всего используются для задачи кластеризации данных, где выборка объектов разбивается на непересекающиеся подмножества, называемые кластерами, так, чтобы каждый кластер состоял из схожих объектов, а объекты разных кластеров существенно отличались.

Кластеризация обычно применяется для следующих целей:

- Визуализация данных (наглядное представление многомерных данных).
- Сегментация рынка (определение типов клиентов).
- Рекомендательные системы (на основе кластеризации пользователей можно предлагать им товары или услуги, которые могут их заинтересовать).
- Объединение близких точек на карте (может использоваться для сжатия изображений).
- Обнаружение выбросов (помогает выявить аномальные значения в наборе данных и устранить их).

1 Обучение без учителя

https://tproger.ru/articles/kak-rabotaet-obuchenie-bez-uchitelya

1.1 Сигнальный метод обучения Хебба

Сигнальный метод обучения Хебба заключается в изменении весов по следующему правилу [1]:

$$w_{ij}(t) = w_{ij}(t-1) + \alpha \cdot y_i^{(n-1)} \cdot y_j^{(n)} \quad (1)$$

где $y_i^{(n-1)}$ – выходное значение нейрона i слоя $(n-1)$, $y_j^{(n)}$ – выходное значение нейрона j слоя n; $w_{ij}(t)$ и $w_{ij}(t-1)$ – весовой коэффициент синапса, соединяющего эти нейроны, на итерациях t и $t-1$ соответственно, α – коэффициент скорости обучения. Здесь и далее, для общности, под n подразумевается произвольный слой сети. При обучении по данному методу усиливаются связи между возбужденными нейронами.

Существует также и дифференциальный метод обучения Хебба.

$$w_{ij}(t) = w_{ij}(t-1) + \alpha \cdot [y_i^{(n-1)}(t) - y_i^{(n-1)}(t-1)] \cdot [y_j^{(n)}(t) - y_j^{(n)}(t-1)] \quad (2)$$

Здесь $y_i^{(n-1)}(t)$ и $y_i^{(n-1)}(t-1)$ – выходное значение нейрона i слоя $n-1$ соответственно на итерациях t и $t-1$; $y_j^{(n)}(t)$ и $y_j^{(n)}(t-1)$ – то же самое для нейрона j слоя n. Как видно из формулы (2), сильнее всего обучаются синапсы, соединяющие те нейроны, выходы которых наиболее динамично изменились в сторону увеличения.

1.1.1 Алгоритм обучения с применением метода Хебба

Алгоритм основан на принципе ассоциативной памяти и позволяет нейронной сети устанавливать связи между входными и выходными данными [2].

1. На стадии инициализации всем весовым коэффициентам присваиваются небольшие случайные значения.
2. На входы сети подается входной образ, и сигналы возбуждения распространяются по всем слоям согласно принципам классических прямопоточных сетей, то есть для каждого нейрона рассчитывается взвешенная сумма его входов, к которой затем применяется функция активации нейрона, в результате чего получается выходное значение $y_i^{(n)}, i = 0, \ldots, M_i - 1$, где

M_i – число нейронов в слое $i; n = 0, \ldots, N-1$, а N – число слоёв в сети.

3. На основании полученных выходных значений нейронов по формуле (1) или (2) производится изменение весовых коэффициентов.
4. Цикл с шага 2, пока выходные значения сети не застабилизируются с заданной точностью.

Применение этого способа определения завершения обучения, отличного от использовавшегося для сети обратного распространения, обусловлено тем, что подстраиваемые значения синапсов фактически не ограничены.

На втором шаге цикла попеременно предъявляются все образы из входного набора. Следует отметить, что вид откликов на каждый класс входных образов не известен заранее и будет представлять собой произвольное сочетание состояний нейронов выходного слоя, обусловленное случайным распределением весов на стадии инициализации. Вместе с тем, сеть способна обобщать схожие образы, относя их к одному классу. Тестирование обученной сети позволяет определить топологию классов в выходном слое. Для приведения откликов обученной сети к удобному представлению можно дополнить сеть одним слоем, который, например, по алгоритму обучения однослойного перцептрона необходимо заставить отображать выходные реакции сети в требуемые образы.

1.2 Метод обучения Кохонена

Другой алгоритм обучения без учителя – алгоритм Кохонена – предусматривает подстройку синапсов на основании их значений от предыдущей итерации.

$$w_{ij}(t) = w_{ij}(t-1) + \alpha \cdot [y_i^{(n-1)} - w_{ij}(t-1)] \quad (3)$$

Из вышеприведенной формулы видно, что обучение сводится к минимизации разницы между входными сигналами нейрона, поступающими с выходов нейронов предыдущего слоя $y_i^{(n-1)}$, и весовыми коэффициентами его синапсов.

Полный алгоритм обучения имеет примерно такую же структуру, как в методах Хебба, но на шаге 3 из всего слоя выбирается нейрон, значения синапсов которого максимально походят на входной образ, и подстройка весов по формуле (3) проводится только для него. Эта, так называемая, аккредитация может сопровождаться затормаживанием всех остальных нейронов слоя и введением выбранного нейрона в насыщение. Выбор такого нейрона может осу-

ществляться, например, расчетом скалярного произведения вектора весовых коэффициентов с вектором входных значений. Максимальное произведение дает выигравший нейрон.

Другой вариант – расчет расстояния между этими векторами в p-мерном пространстве, где p – размер векторов.

$$D_j = \sqrt{\sum_{i=0}^{p-1}(y_i^{(n-1)} - w_{ij})^2} \quad (4)$$

где j – индекс нейрона в слое n, i – индекс суммирования по нейронам слоя $(n-1)$, w_{ij} – вес синапса, соединяющего нейроны; выхода нейронов слоя $(n-1)$ являются входными значениями для слоя n. Корень в формуле (4) не обязателен, так как важна лишь относительная оценка различных D_j.

В данном случае, «побеждает» нейрон с наименьшим растоянием. Иногда слишкомчасто получающие аккредитацию нейроны принудительно исключаются из рассмотрения, чтобы «уравнять права» всех нейронов слоя. Простейший вариант такого алгоритма заключается в торможении только что выигравшего нейрона.

При использовании обучения по алгоритму Кохонена существует практика нормализации входных образов, а так же – на стадии инициализации и нормализации начальных значений весовых коэффициентов.

$$x_i = x_i / \sqrt{\sum_{j=0}^{n-1} x_j^2},$$

где x_i – i-я компонента вектора входного образа, n – его размерность. Это позволяет сократить длительность процесса обучения.

Инициализация весовых коэффициентов случайными значениями может привести к тому, что различные классы, которым соответствуют плотно распределенные входные образы, сольются или, наоборот, раздробятся на дополнительные подклассы в случае близких образов одного и того же класса. Для избежания такой ситуации используется метод выпуклой комбинации. Суть его сводится к тому, что входные нормализованные образы подвергаются преобразованию:

$$x_i = \alpha(t) \cdot x_i + (1 - \alpha(t)) \cdot \frac{1}{\sqrt{n}},$$

где x_i – i-я компонента вектора входного образа, n – общее число его компонент, $\alpha(t)$ – коэффициент, изменяющийся в процессе обучения от нуля до единицы, в результате чего вначале на входы сети подаются практически одинаковые образы, а с течением времени они все больше сходятся к исходным. Весовые коэффициенты устанавливаются на шаге инициализации равными величине

$$w_o = \frac{1}{\sqrt{n}}$$

На основе рассмотренного выше метода строятся нейронные сети особого типа – так называемые самоорганизующиеся структуры – self-organizing feature maps. Для них после выбора из слоя n нейрона j с минимальным расстоянием D_j (4) обучается по формуле (3) не только этот нейрон, но и его соседи, расположенные в окрестности R. Величина R на первых итерациях очень большая, так что обучаются все нейроны, но с течением времени она уменьшается до нуля. Таким образом, чем ближе конец обучения, тем точнее определяется группа нейронов, отвечающих каждому классу образов.

2 Кластеризация данных

http://gorbachenko.self-organization.ru/articles/Self-organizing_map.pdf https://www

Формально задача кластеризации описывается следующим образом [3]. Дано множество объектов $I = \{i_1, i_2, \ldots, i_n\}$, каждый из которых характеризуется вектором $x_j, j = 1, \ldots, n$ атрибутов (параметров): $x_j = \{x_{j1}, \ldots, x_{jm}\}$. Требуется построить множество кластеров C и отображение F множества I на множество C, то есть $F : I \to C$. Задача кластеризации состоит в построении множества

$$C = \{c_1, c_2, \ldots, c_k, \ldots, c_g\},$$

где c_k – кластер, содержащий «похожие» объекты из I:

$$c_k = \{i_j, i_p | i_j \in I, i_p \in I \text{ и } \rho(i_j, i_p) < \sigma\}, \quad (5)$$

σ – величина, определяющая меру близости для включения объектов в один кластер, $\rho(i_j, i_p)$ – мера близости между объектами, называемая расстоянием.

Если расстояние $\rho(i_j, i_p)$ меньше некоторого значения σ, то объекты считаются близкими и помещаются в один кластер. В противном случае считается, что объекты отличны друг от друга и их помещают в разные кластеры. Условие (5) известно как гипотеза компактности.

Неотрицательное число $\rho(x, y)$ называется расстоянием (метрикой) между векторами x и y, если выполняются следующие условия:

1. $\rho(x, y) \geq 0$ для всех x и y.
2. $\rho(x, y) = 0$, тогда и только тогда, когда $x = y$.
3. $\rho(x, y) = \rho(y, x)$.
4. $\rho(x, y) \leq \rho(x, k) + \rho(k, y)$ – неравенство треугольника.

Евклидово расстояние между векторами x и y представляет собой евклидову норму разности векторов, или длину отрезка, соединяющего точки x и y.

Евклидово расстояние является частным случаем расстояния Минковского

$$\rho_p(x, y) = (\sum_{i=1}^{m} |x_i - y_i|^p)^{\frac{1}{p}} = ||x - y||_p,$$

где $||z||_p = (\sum_{i=1}^{m} |z_i|^p)^{\frac{1}{p}}$ – p-норма вектора z. Тогда 2-норма — это евклидова

норма.

Другой частный случай – 1-норма, которая называется манхэттенским расстоянием (расстоянием городских кварталов)

$$\rho_1(x, y) = \sum_{i=1}^{m} |x_i - y_i|$$

Манхеттенское расстояние – это расстояние, которое проходится, двигаясь параллельно осям координат, как в Манхеттнене или других городах с прямоугольной продольно-поперечной планировкой улиц.

2.1 Самоорганизующиеся карты Кохонена

2.2 Метод k-средних

Это наиболее популярный метод кластеризации. Цель алгоритма – минимизировать сумму квадратов внутрикластерных расстояний до центра кластера [4]. Функция потерь (или целевая функция) имеет вид:

$$J = \sum_{j=1}^{k} \sum_{x \in C_i} (x - \mu_i)^2,$$

где k – число кластеров, C_i – полученные кластеры, μ_i – центры масс всех векторов x из кластера C_i.

2.2.1 Пример работы алгоритма

Действие алгоритма в двумерном случае. Начальные точки выбраны случайно [5].

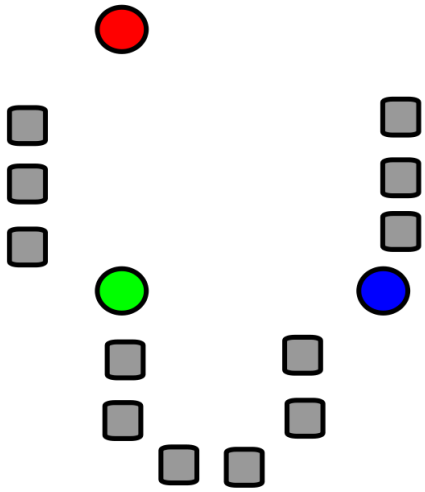

Рисунок 1 – Исходные точки и случайно выбранные начальные центры

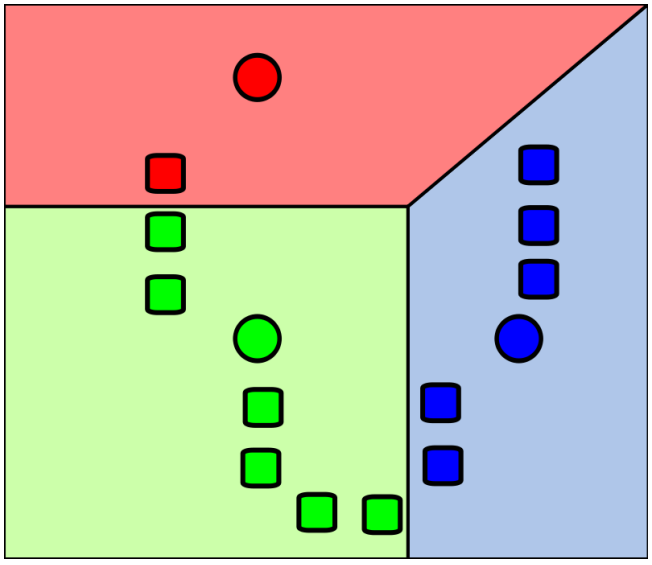

Рисунок 2 – Точки, отнесённые к начальным центрам

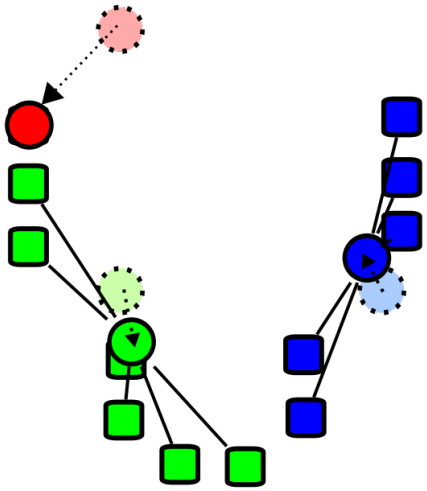

Рисунок 3 – Вычисление новых центров кластеров

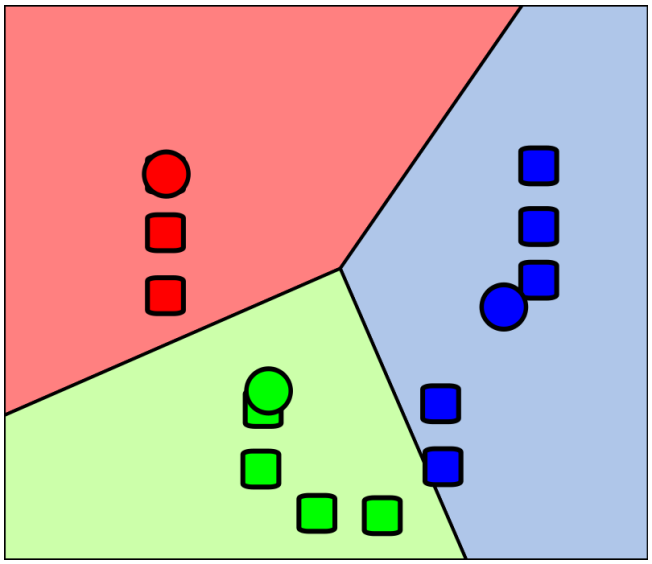

Рисунок 4 – Точки, отнесённые к новым центрам

Предыдущие шаги, за исключением первого, повторяются, пока алгоритм не сойдётся.

2.2.2 Проблемы метода

На вход алгоритма должно подаваться количество кластеров. Есть два способа выбора количества кластеров:

1. Экспертный метод (domain knowledge). Выбор количества кластеров будет зависеть от знания о предметной области.
2. Метод локтя (elbow method). Можно обучить модель используя несколько вариантов количества кластеров, измерить сумму квадратов внутрикла-

стерных расстояний и выбрать тот вариант, при котором данное расстояние перестанет существенно уменьшаться.

Так же не гарантируется достижение глобального минимума суммарного квадратичного отклонения J, а только одного из локальных минимумов. И результат зависит от выбора исходных центров кластеров, их оптимальный выбор неизвестен.

2.3 DBSCAN

Основанная на плотности пространственная кластеризация для приложений с шумами (Density-based spatial clustering of applications with noise, DBSCAN) – алгоритм кластеризации, основанной на плотности – если дан набор точек в некотором пространстве, алгоритм группирует вместе точки, которые тесно расположены, помечая как выбросы точки, которые находятся одиноко в областях с малой плотностью (ближайшие соседи которых лежат далеко) [6].

Рассмотрим набор точек в некотором пространстве, требующий кластеризации. Для выполнения кластеризации DBSCAN точки делятся на основные точки, достижимые по плотности точки и выпадающие следующим образом:

1. Точка p является основной точкой, если по меньшей мере m точек находятся на расстоянии, не превосходящем ϵ (ϵ является максимальным радиусом соседства от p), до неё (включая саму точку p). Говорят, что эти точки достижимы прямо из p.
2. Точка q прямо достижима из p, если точка q находится на расстоянии, не большем ϵ , от точки p и p должна быть основной точкой.
3. Точка q достижима из p, если имеется путь $p_1 = 1, \ldots, p_n = q$, где каждая точка p_{i+1} достижима прямо из p_i (все точки на пути должны быть основными, за исключением q).
4. Все точки, не достижимые из основных точек, считаются выбросами.

Теперь, если p является основной точкой, то она формирует кластер вместе со всеми точками (основными или неосновными), достижимые из этой точки. Каждый кластер содержит по меньшей мере одну основную точку. Неосновные точки могут быть частью кластера, но они формируют его «край», поскольку не могут быть использованы для достижения других точек.

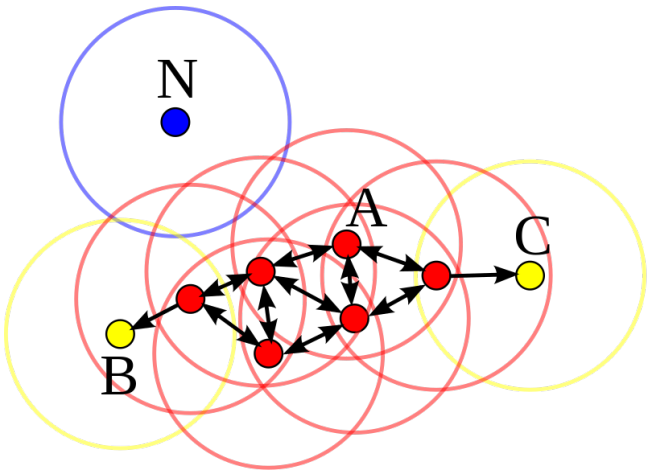

Рисунок 5 – Пример диаграммы с $m = 4$

Точка A и другие красные точки являются основными точками, поскольку область с радиусом ϵ, окружающая эти точки, содержит по меньшей мере 4 точки (включая саму точку). Поскольку все они достижимы друг из друга, точки образуют один кластер. Точки B и C основными не являются, но достижимы из A (через другие основные точки), и также принадлежат кластеру. Точка N является точкой шума, она не является ни основной точкой, ни доступной прямо.

Достижимость не является симметричным отношением, поскольку, по определению, никакая точка не может быть достигнута из неосновной точки, независимо от расстояния (так что неосновная точка может быть достижимой, но ничто не может быть достигнуто из неё). Поэтому дальнейшее понятие связности необходимо для формального определения области кластеров, найденных алгоритмом DBSCAN. Две точки p и q связаны по плотности, если имеется точка o, такая что и p, и q достижимы из o. Связность по плотности является симметричной.

Тогда кластер удовлетворяет двум свойствам:

1. Все точки в кластере попарно связны по плотности.
2. Если точка достижима по плотности из какой-то точки кластера, она также принадлежит кластеру.

2.3.1 Алгоритм

DBSCAN требует задания двух параметров: ϵ и минимального числа точек, которые должны образовывать плотную область m. Алгоритм начинается с произвольной точки, которая ещё не просматривалась. Выбирается ϵ-окрестность

точки и, если она содержит достаточно много точек, образуется кластер, в противном случае точка помечается как шум. Заметим, что эта точка может быть позже найдена в ϵ-окрестности другой точки и включена в какой-то кластер.

Если точка найдена как плотная точка кластера, её ϵ-окрестность также является частью этого кластера. Следовательно, все точки, найденные в ϵ-окрестности этой точки, добавляются к кластеру. Этот процесс продолжается, пока не будет найден связный по плотности кластер. Затем выбирается и обрабатывается новая непосещённая точка, что ведёт к обнаружению следующего кластера или шума.

DBSCAN может быть использован с любой функцией расстояния (а так же с функцией похожести или логическим условием). Функция расстояния может поэтому рассматриваться как дополнительный параметр.

2.3.2 Преимущества и недостатки

Преимущества:

1. DBSCAN не требует спецификации числа кластеров в данных априори в отличие от метода k-средних.
2. DBSCAN может найти кластеры произвольной формы. Он может найти даже кластеры полностью окружённые (но не связанные с) другими кластерами. Благодаря параметру MinPts уменьшается так называемый эффект одной связи (связь различных кластеров тонкой линией точек).
3. DBSCAN имеет понятие шума и устойчив к выбросам.
4. DBSCAN требует лишь двух параметров и большей частью нечувствителен к порядку точек в базе данных. (Однако, точки, находящиеся на границе двух различных кластеров могут оказаться в другом кластере, если изменить порядок точек, а назначение кластеров единственно с точностью до изоморфизма).
5. DBSCAN разработан для применения с базами данных, которые позволяют ускорить запросы в диапазоне значений, например, с помощью R*-дерева.
6. Параметры m и ϵ могут быть установлены экспертами в рассматриваемой области, если данные хорошо понимаются.

Недостатки:

1. DBSCAN не полностью однозначен — краевые точки, которые могут быть достигнуты из более чем одного кластера, могут принадлежать любому из

этих кластеров, что зависит от порядка просмотра точек. Для большинства наборов данных эти ситуации возникают редко и имеют малое влияние на результат кластеризации – основные точки и шум DBSCAN обрабатывает однозначно. DBSCAN является вариантом, который трактует краевые точки как шум и тем самым достигается полностью однозначный результат, а также более согласованная статистическая интерпретация связных по плотности компонент.

2. Качество DBSCAN зависит от измерения расстояния. Наиболее часто используемой метрикой расстояний является евклидова метрика. Особенно для кластеризации данных высокой размерности эта метрика может оказаться почти бесполезной ввиду так называемого «проклятия размерности», что делает трудным делом нахождение подходящего значения ϵ. Этот эффект, однако, присутствует в любом другом алгоритме, основанном на евклидовом расстоянии.
3. DBSCAN не может хорошо кластеризовать наборы данных с большой разницей в плотности, поскольку не удается выбрать приемлемую для всех кластеров комбинацию $m - \epsilon$.
4. Если данные и масштаб не вполне хорошо поняты, выбор осмысленного порога расстояния ϵ может оказаться трудным.

ЗАКЛЮЧЕНИЕ

Таким образом, обучение без учителя – это процесс обучения модели на основе неразмеченных данных, где нет известных меток или целевых переменных. В этом случае модель ищет скрытые структуры и закономерности в данных, чтобы создать представление или кластеризацию данных.

Преимуществами данного метода обучения являются:

1. Извлечение скрытых структур. Обучение без учителя позволяет модели извлекать скрытые структуры и паттерны из данных, что может быть полезно для обнаружения новых знаний и понимания данных.
2. Работа с большими объемами данных. Обучение без учителя может быть эффективным при работе с большими объемами данных, поскольку нет необходимости размечать каждый пример.
3. Автоматическое обучение. Обучение без учителя позволяет модели самостоятельно находить паттерны и структуры в данных, без необходимости вручную определять правильные ответы.

Недостатки:

1. Неопределенность результатов. Результаты обучения без учителя могут быть менее интерпретируемыми и требуют дополнительного анализа и проверки.
2. Трудность оценки. Оценка качества модели в обучении без учителя может быть сложной, поскольку нет явных правильных ответов для сравнения.
3. Необходимость предварительной обработки данных. В обучении без учителя может потребоваться предварительная обработка данных для удаления шума или выбросов, что может быть трудоемким процессом.

https://masters.donntu.ru/2006/kita/chvala/library/N3.pdf

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1 Короткий С. Нейронные сети: обучение без учителя [Электронный ресурс] – URL: https://masters.donntu.ru/2006/kita/chvala/library/N3.pdf. (Дата обращения 11.01.2024). Загл. с экр. Яз. рус.

2 Обучение Хебба: простыми словами о принципах, алгоритме и применении в нейронных сетях [Электронный ресурс]. – URL: https://skine.ru/articles/1511/ (Дата обращения 11.01.2024). Загл. с экр. Яз. рус.

3 Горбаченко В. И. Самоорганизация в нейронных сетях [Электронный ресурс] //Научно-исследовательский центр самоорганизации и развития систем.–2018. – URL: http://gorbachenko.self-organization.ru/articles/Self-organizing_map.pdf. (Дата обращения 11.01.2024). Загл. с экр. Яз. рус.

4 Алгоритм кластеризации K-средних [Электронный ресурс]. – URL: https://skine.ru/articles/1511/ (Дата обращения 11.01.2024). Загл. с экр. Яз. рус.

5 Метод k-средних [Электронный ресурс]. – URL: https://ru.wikipedia.org/wiki/Метод_k-средних (Дата обращения 11.01.2024). Загл. с экр. Яз. рус.

6 DBSCAN [Электронный ресурс]. – URL: https://ru.wikipedia.org/wiki/DBSCAN (Дата обращения 11.01.2024). Загл. с экр. Яз. рус.