**Практическая работа №10**

# 7 лучших программ для векторизации изображений для дизайнеров и художников

Векторные изображения состоят из основанных на курсоре операций рисования, которые состоят из фигур, таких как круги, прямоугольники, линии и кривые (или произвольная форма), созданных путем перемещения [курсора](https://gadgetshelp.com/windows/kak-skachat-bolshe-kursorov-v-windows-8-windows-10/) или рисования линии и кривой в точку с помощью какого-либо элемента управления, затем обводки или заливки их создавать произведения искусства. Чтобы использовать их или адаптировать изображения к векторам, вам необходимо использовать программное обеспечение для векторизации изображений.

Однако бывают случаи, когда вы просто хотите преобразовать определенное изображение в векторное изображение, и именно здесь вам пригодится лучшее **программное обеспечение для векторизации изображений** .

Растровое изображение уже имеет сетку пикселей и значений цвета, но вы можете нарисовать одно и то же изображение, используя различные формы и последовательности штрихов и заливки.

В зависимости от того, чего вы хотите достичь, программное обеспечение для векторизации изображений лучше, чем начинать с нуля, потому что оно более точное и чистое, если вы не профессионал в этом.

## Каковы лучшие инструменты для векторизации изображений?

1

### Adobe Illustrator

Adobe Illustrator является одним из многих премиальных решений, предлагаемых Adobe. Сказать, что Adobe Illustrator является лучшим и самым лучшим решением для векторного дизайна, было бы преуменьшением. Если вы профессиональный дизайнер, у вас будут все необходимые инструменты.

Он охватывает все необходимые инструменты для разработки обложек и художественного оформления, включая векторизацию изображений, расширенную типографику, градиенты произвольной формы и многое другое. Поначалу это изобилие функций может быть даже немного подавляющим, особенно если вы начинающий пользователь, но его легко освоить.

Наиболее отличительные особенности Adobe Illustrator CC:

* Отраслевой набор инструментов цифрового дизайна.
* Отличная поддержка с подробными уроками.
* Поддержка Creative Cloud.

Вы можете попробовать Adobe Illustrator CC с месячной бесплатной пробной версией.

[**Получить Adobe Illustrator сейчас**](https://translate.googleusercontent.com/translate_c?depth=1&rurl=translate.google.ru&sl=en&sp=nmt4&tl=ru&u=https://clk.tradedoubler.com/click%3Fp%3D284424%26a%3D3111099%26g%3D24061066%26epi%3Dvectorize_image_software&xid=17259,15700023,15700043,15700186,15700190,15700256,15700259,15700262,15700265,15700271&usg=ALkJrhg1m4E5Na3aUWAcp3aG2xeYBhd5FQ)

2

### Vextractor

Это лучшее программное обеспечение для векторизации изображений преобразует растровые изображения, такие как рисунки или карты, и другие, такие как [фото логотипы](https://gadgetshelp.com/tvorcheskii/luchshee-programmnoe-obespechenie-dlia-udaleniia-logotipov-dlia-redaktirovaniia-izobrazhenii-rukovodstvo-po-2020/) и иллюстрации, в векторный формат.

Программа сохраняет преобразованные изображения в векторные форматы DXF, WMF, EMF, EPS, AI или SVG, которые могут быть импортированы в популярные приложения векторной графики, такие как Corel Draw, AutoCAD и Illustrator, и другие.

Это идеальная замена для традиционного отслеживания и оцифровки, экономит много времени и быстрее, чем отслеживание вручную или с использованием цифрового преобразователя.

[**Оптимизируйте свои изображения, чтобы лучше соответствовать вашим проектам.**](https://gadgetshelp.com/tvorcheskii/luchshee-programmnoe-obespechenie-dlia-optimizatsii-izobrazhenii-dlia-pk-rukovodstvo-po-2020/) [**Получите лучшие инструменты для оптимизации изображений прямо здесь.**](https://gadgetshelp.com/tvorcheskii/luchshee-programmnoe-obespechenie-dlia-optimizatsii-izobrazhenii-dlia-pk-rukovodstvo-po-2020/)

Это простой и легкий способ конвертировать чертежи из бумаги в [САПР](https://gadgetshelp.com/windows/luchshee-programmnoe-obespechenie-dlia-preobrazovaniia-failov-autocad-v-pdf-dlia-windows/) для дальнейшего редактирования и использования, экономя ваше время и монотонность при работе.

Некоторые функции и инструменты включают в себя масштабирование, прокрутку, выбор цвета, поддержку TWAIN для импорта изображений, автоматическое распознавание линий и дуг или окружностей, преобразование цветных и черно-белых изображений для распознавания, встроенный векторный редактор для редактирования результаты векторизации, сплайны Безье для создания кривых и многое другое.

Такие профессионалы, как ГИС, используют его для оцифровки печатных карт, поскольку он поддерживает форматы ГИС, такие как шейп-файлы ArcInfo и MapInfo MID / MIF, а также имеет географическую привязку для растровых изображений по точкам.

[Получить Vextractor](https://translate.googleusercontent.com/translate_c?depth=1&rurl=translate.google.ru&sl=en&sp=nmt4&tl=ru&u=http://www.vextrasoft.com/download.htm&xid=17259,15700023,15700043,15700186,15700190,15700256,15700259,15700262,15700265,15700271&usg=ALkJrhj_Lu5lpsjzwgwtaJKHpTNQ7Y9cjw)

3

### Inkscape

Это бесплатный инструмент с векторным графическим редактором с открытым исходным кодом, в функции которого входят инструменты для редактирования фотографий или изображений и текста, а также поддержка большинства графических форматов, таких как SVG, PNG, WMF, [PDF](https://gadgetshelp.com/otzyvy-o-produkte/5-luchshikh-besplatnykh-programm-dlia-chteniia-pdf-dlia-windows-10/) и многих других.

Являетесь ли вы иллюстратором, дизайнером, веб-дизайнером или просто хотите создавать векторные изображения, это лучшее программное обеспечение для векторизации изображений.

[**Соедините ваш векторный графический редактор с онлайн-редактором HTML5.**](https://gadgetshelp.com/windows/5-luchshikh-onlain-redaktorov-html5-dlia-ispolzovaniia-v-2019-godu/) [**Проверьте лучшие из них здесь.**](https://gadgetshelp.com/windows/5-luchshikh-onlain-redaktorov-html5-dlia-ispolzovaniia-v-2019-godu/)

Он также имеет кривые Безье и Спиро, мощный текстовый инструмент и широкую совместимость форматов файлов.

[Получить Inkscape](https://translate.googleusercontent.com/translate_c?depth=1&rurl=translate.google.ru&sl=en&sp=nmt4&tl=ru&u=https://inkscape.org/en/release/&xid=17259,15700023,15700043,15700186,15700190,15700256,15700259,15700262,15700265,15700271&usg=ALkJrhjBnkvABE5V1s_5ntbNkFKisV-qsw)

4

### WinTopo

Это высококачественное программное обеспечение для преобразования файлов изображений в различных форматах, таких как JPG, PNG, BMP, GIF или TIF, и отсканированных изображений для использования файлов векторных изображений, подходящих для приложений САПР, ГИС и ЧПУ.

Он поставляется в двух версиях: **Freeware** и **Pro** .

Первое, WinTopo Freeware, совершенно бесплатно для всех, но все же дает лучшие результаты, чем большинство коммерческих альтернатив, которые довольно дороги, и имеет более 10 миллионов пользователей по всему миру.

С другой стороны, WinTopo Pro, профессиональная версия, имеет больше возможностей обработки и более тонкий контроль при преобразовании изображений в векторы или векторизации изображений.

[**Поднимите ваш проект на новый уровень с лучшим программным обеспечением для сжатия изображений из этого списка.**](https://gadgetshelp.com/tvorcheskii/luchshee-programmnoe-obespechenie-dlia-szhatiia-izobrazhenii-dlia-windows-10-rukovodstvo-po-2020/)

Он включает в себя опции предварительной обработки, которые помогают получать оптимальные результаты, а также более продвинутый механизм векторизации, созданный за годы программирования, проектирования и обработки изображений.

Каждая из этих версий поставляется с векторизацией в одно касание, распознаванием дуг, сглаживанием и уменьшением полилиний, предварительным просмотром в окнах параметров, автоматическим масштабированием, а также различными форматами сохранения векторов.

Тем не менее, Pro версия имеет больше функций, таких как, например, географическая привязка, пакетная обработка, параметры негатива и яркости, заполнение отверстий, удаление пятен и обрезка.

[Получить WinTopo](https://translate.googleusercontent.com/translate_c?depth=1&rurl=translate.google.ru&sl=en&sp=nmt4&tl=ru&u=http://wintopo.com/download.htm&xid=17259,15700023,15700043,15700186,15700190,15700256,15700259,15700262,15700265,15700271&usg=ALkJrhgJLPRMvtbHc_il-ETpZuq7c850Kg)

5

### Вектор магии

Это одна из самых популярных программ для векторизации изображений, которую вы можете использовать для быстрого преобразования растровых изображений в векторы и подготовки ваших рисунков к печати или вышивания, резки и других процессов.

С помощью этого инструмента вы можете автоматически конвертировать изображения в настоящие SVG, EPS и PDF-векторы онлайн, загружая их без необходимости устанавливать программное обеспечение, и вы получите мгновенные результаты.

Он также поставляется с настольным автономным приложением для векторизации изображений, с поддержкой форматов файлов и вывода AI или DXF, а также хорошо сочетается с Illustrator, Corel и другими программами.

Векторизация происходит автоматически после того, как Vector Magic обнаружит ваше изображение, проанализирует и предоставит соответствующие настройки для векторизации, а затем отследит формы в полном цвете, после чего вы сможете просмотреть результат.

[**Хотите воплотить свои образы в жизнь?**](https://gadgetshelp.com/windows/luchshee-programmnoe-obespechenie-dlia-3d-pechati-na-pk-rukovodstvo-po-2020/) [**Вот лучшие инструменты для 3d-печати.**](https://gadgetshelp.com/windows/luchshee-programmnoe-obespechenie-dlia-3d-pechati-na-pk-rukovodstvo-po-2020/)

Другие функции включают субпиксельную точность, с которой вы можете отследить каждый бит информации и точные пиксели среза, чтобы воссоздать намерение исходного изображения, правильное количество узлов, размещенных в отличных местах, облегчает работу с результатом, легко используйте параметры для редактирования результатов в Интернете и в приложении для настольных компьютеров, а также трассировку изображения для выделения фигур и создания реального векторного изображения.

В Интернете существует больше сервисов, которые утверждают, что векторизуют изображения, но в действительности они просто встраивают пиксели, не отслеживая их по векторам, оставляя размытый файл при масштабировании.

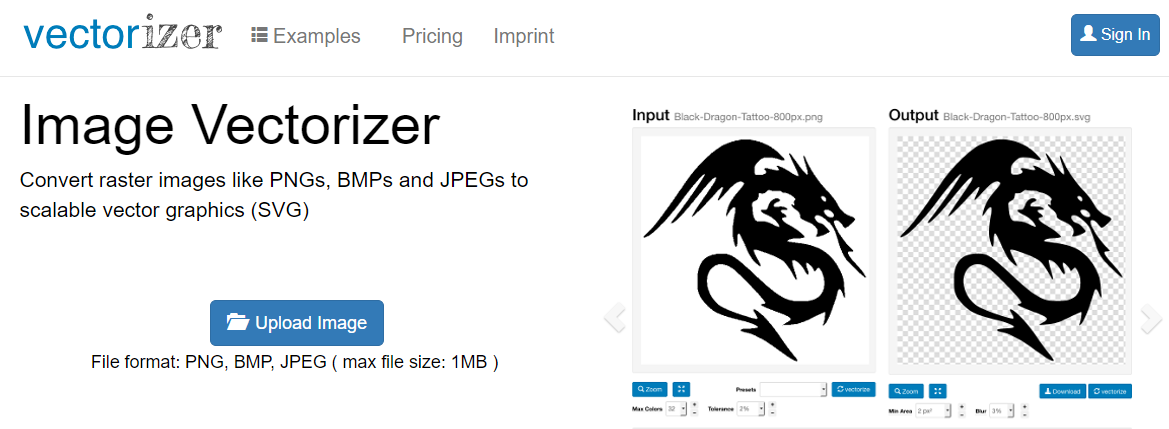
Vector Magic всегда отслеживает ваше растровое изображение, тщательно выявляя лежащие в его основе фигуры, и предоставляет вам настоящее векторное изображение со всеми его преимуществами.

В отличие от этого, в Интернете доступно множество сервисов, которые утверждают, что преобразуют растровые изображения в векторы, но в действительности они просто встраивают пиксели, фактически не отслеживая их в векторных формах.

[Получить вектор магии](https://translate.googleusercontent.com/translate_c?depth=1&rurl=translate.google.ru&sl=en&sp=nmt4&tl=ru&u=https://vectormagic.com/&xid=17259,15700023,15700043,15700186,15700190,15700256,15700259,15700262,15700265,15700271&usg=ALkJrhgKpjv5xh8aVKLYMwfsjrLQsWECBw)

6

### Векторизатор

[](https://cdn.windowsreport.com/wp-content/uploads/2018/02/2018-02-14_12-46-32.png)

Этот векторизатор изображений преобразует растровые изображения, такие как PNG, BMP и JPEG, в масштабируемую векторную графику или SVG. Для векторизации растровых изображений вы преобразуете информацию о цвете пикселей в простые геометрические объекты.

Векторная графика основана не на пикселях, а на точках, линиях и кривых, которые также известны как примитивы, представленные математическими выражениями.

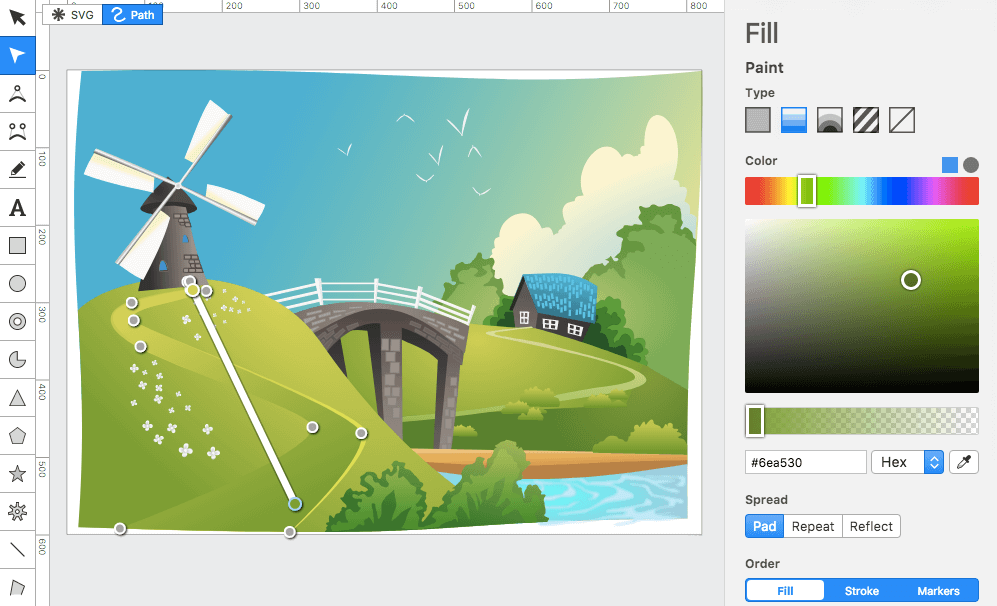
Vectorizer преобразует эти растровые изображения без потери качества графики, что делает их легко масштабируемыми и вращаемыми. Некоторые примеры включают логотипы, татуировки, клипарты, наклейки, наклейки и дизайны футболок.

[Получить Vectorizer](https://translate.googleusercontent.com/translate_c?depth=1&rurl=translate.google.ru&sl=en&sp=nmt4&tl=ru&u=https://www.vectorizer.io/pricing/&xid=17259,15700023,15700043,15700186,15700190,15700256,15700259,15700262,15700265,15700271&usg=ALkJrhimYMxakb-f8pPPwddGJ0EGsFMmIQ)

[**Нужно редактировать свои изображения как эксперт?**](https://gadgetshelp.com/windows/8-luchshikh-programm-dlia-redaktirovaniia-fotografii-na-pk-2019/) [**Проверьте лучшие графические редакторы, доступные прямо здесь.**](https://gadgetshelp.com/windows/8-luchshikh-programm-dlia-redaktirovaniia-fotografii-na-pk-2019/)

7

### BoxySVG

[](https://cdn.windowsreport.com/wp-content/uploads/2018/02/BoxySVG.png)

BoxySVG — это относительно простой в использовании инструмент. В отличие от инструментов, перечисленных выше, эта программа не предлагает расширенные функции, но она идеально подходит для вас, если вы новичок.

Как только вы освоите этот инструмент и научитесь создавать векторные изображения, вы сможете поиграть с опциями типографии и геометрии. Как только вы будете удовлетворены результатами, вы можете экспортировать SVG-файлы в другие форматы.

Вы можете использовать [BoxySVG как самостоятельный инструмент](https://translate.googleusercontent.com/translate_c?depth=1&rurl=translate.google.ru&sl=en&sp=nmt4&tl=ru&u=https://boxy-svg.com/&xid=17259,15700023,15700043,15700186,15700190,15700256,15700259,15700262,15700265,15700271&usg=ALkJrhg_5lbTV6zuTKm5Eyt3K8LrkXAWmw" \t "_blank) или загрузить расширение для браузера.

[**Нужно больше инструментов векторной графики?**](https://gadgetshelp.com/tvorcheskii/luchshee-programmnoe-obespechenie-dlia-vektornoi-grafiki-dlia-sozdaniia-krasivykh-dizainov-rukovodstvo-po-2020/) [**Проверьте этот список с лучшими инструментами для создания удивительных дизайнов.**](https://gadgetshelp.com/tvorcheskii/luchshee-programmnoe-obespechenie-dlia-vektornoi-grafiki-dlia-sozdaniia-krasivykh-dizainov-rukovodstvo-po-2020/)

Вы пробовали какое-либо из этих лучших программ для векторизации изображений?

Если это так, дайте нам знать, какой из них и как он оказался для вас, или инструмент, который вы использовали, оставив комментарий в разделе ниже.

**СВЯЗАННЫЕ ИСТОРИИ, ЧТОБЫ ПРОВЕРИТЬ:**

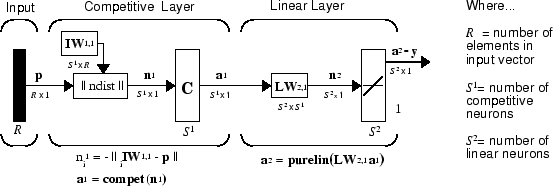
* [4 лучших поисковых системы для поиска визуально похожих изображений](https://gadgetshelp.com/internet/4-luchshikh-poiskovykh-sistemy-dlia-poiska-vizualno-pokhozhikh-izobrazhenii/)
* [6 лучших программ для рендеринга вашего проекта SolidWorks](https://gadgetshelp.com/tvorcheskii/luchshee-programmnoe-obespechenie-dlia-vizualizatsii-solidworks-rukovodstvo-po-2020/)

**Сети векторного квантования (LVQ)**

## Нейронные сети векторного квантования обучения (LVQ)

### Архитектура

Архитектура сети LVQ показана ниже.



Сеть LVQ имеет первый конкурентный уровень и второй линейный уровень. Конкурентный слой учится классифицировать входные векторы во многом так же, как и конкурентные слои [Кластера с Самоорганизующейся картографической нейронной сетью](https://translated.turbopages.org/proxy_u/en-ru.ru.a34c63d0-624452c2-8cb32aab-74722d776562/https/www.mathworks.com/help/deeplearning/ug/cluster-with-self-organizing-map-neural-network.html), описанной в этом разделе. Линейный уровень преобразует классы конкурентного уровня в целевые классификации, определенные пользователем. Классы, изучаемые конкурентным уровнем, называются подклассами, а классы линейного уровня-целевыми классами.

Как конкурентный, так и линейный слои имеют по одному нейрону на класс (суб-или целевой). Таким образом, конкурентный слой может изучить до S1 подклассов. Они, в свою очередь, объединяются линейным слоем для формирования S2 целевых классов. (S1 всегда больше S2.)

Например, предположим, что нейроны 1, 2 и 3 в конкурентном слое изучают все подклассы входного пространства, принадлежащего целевому классу 2 линейного слоя. Тогда конкурентные нейроны 1, 2 и 3 будут иметь веса **LW**2,1 1,0 для нейрона **n2** в линейном слое и веса 0 для всех других линейных нейронов. Таким образом, линейный нейрон производит 1, если любой из трех конкурентных нейронов (1, 2 или 3) выигрывает соревнование и выводит 1. Именно так подклассы конкурентного слоя объединяются в целевые классы в линейном слое.

Короче говоря, a 1 в i-й строке **a**1 (остальные элементы **a**1 будут равны нулю) эффективно выбирает i-й столбец **LW**2,1 в качестве выходного сигнала сети. Каждый такой столбец содержит один 1, соответствующий определенному классу. Таким образом, подклассы 1s из слоя 1 помещаются в различные классы путем умножения **LW**2,1**a**1 в слое 2.

Вы заранее знаете, какая доля нейронов слоя 1 должна быть классифицирована в различные выходы класса слоя 2, поэтому вы можете указать элементы **LW**2,1 в начале. Однако вы должны пройти процедуру обучения, чтобы получить первый слой для получения правильного вывода подкласса для каждого вектора обучающего набора. Этот тренинг обсуждается на [тренинге](https://translated.turbopages.org/proxy_u/en-ru.ru.a34c63d0-624452c2-8cb32aab-74722d776562/https/www.mathworks.com/help/deeplearning/ug/cluster-with-a-competitive-neural-network.html#bss4b6l-6). Во-первых, рассмотрим, как создать исходную сеть.

### Создание сети LVQ

Вы можете создать сеть LVQ с помощью функции[lvqnet](https://translated.turbopages.org/proxy_u/en-ru.ru.a34c63d0-624452c2-8cb32aab-74722d776562/https/www.mathworks.com/help/deeplearning/ref/lvqnet.html),

net = lvqnet(S1,LR,LF)

где

* S1 это количество скрытых нейронов первого слоя.
* LR это скорость обучения (по умолчанию 0,01).
* LF это функция обучения (по умолчанию [learnlv1](https://translated.turbopages.org/proxy_u/en-ru.ru.a34c63d0-624452c2-8cb32aab-74722d776562/https/www.mathworks.com/help/deeplearning/ref/learnlv1.html)).

Предположим, у вас есть 10 входных векторов. Создайте сеть, которая назначает каждый из этих входных векторов одному из четырех подклассов. Таким образом, в первом конкурентном слое имеется четыре нейрона. Затем эти подклассы назначаются одному из двух выходных классов двумя нейронами в слое 2. Входные векторы и цели задаются

П = [-3 -2 -2 0 0 0 0 2 2 3; 0 1 -1 2 1 -1 -2 1 -1 0];

и

Тк = [1 1 1 2 2 2 2 1 1 1];

Это может помочь показать детали того, что вы получаете от этих двух строк кода.

P,Tc

Р =

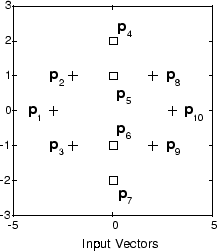
-3 -2 -2 0 0 0 0 2 2 3

0 1 -1 2 1 -1 -2 1 -1 0

Тк =

1 1 1 2 2 2 2 1 1 1

Далее следует график входных векторов.



Как видите, существует четыре подкласса входных векторов. Вам нужна сеть, которая классифицирует **p**1, **p**2, **p**3, **p**8, **p**9 и **p**10 для получения результата 1, а также которая классифицирует векторы **p**4, **p**5, **p**6 и **p**7 для получения результата 2. Обратите внимание, что эта проблема нелинейна. отделима и поэтому не может быть решена персептроном, но сеть LVQ не имеет никаких трудностей.

Затем преобразуйте Tcматрицу в целевые векторы.

T = ind2vec(Tc);

Это дает разреженную матрицуT, которая может быть отображена в полном объеме с

цели = полные(T)

что дает

цели =

1 1 1 0 0 0 0 1 1 1

0 0 0 1 1 1 1 0 0 0

Это выглядит правильно. Например, в нем говорится, что если у вас есть первый столбец Pв качестве входных данных, вы должны получить первый столбец targetsв качестве выходных данных; и этот вывод говорит, что входные данные относятся к классу 1, что правильно. Теперь вы готовы позвонить lvqnet.

Вызов lvqnetдля создания сети с четырьмя нейронами.

net = lvqnet(4);

Настройка и подтверждение начальных значений весовой матрицы первого уровня инициализируются функцией midpoint до значений в центре диапазона входных данных.

net = configure(net,P,T);

net.IW{1}

ans =

0 0

0 0

0 0

0 0

Подтвердите, что веса второго слоя имеют 60% (6 из 10 дюймов Tc) его столбцов с 1 в первой строке (что соответствует классу 1) и 40% его столбцов с 1 во второй строке (что соответствует классу 2). Имея всего четыре столбца, 60% и 40% фактически округляются до 50%, и в каждой строке есть два 1.

net.LW{2,1}

ans =

1 1 0 0

0 0 1 1

Это тоже имеет смысл. В нем говорится, что если конкурентный уровень производит 1 в качестве первого или второго элемента, то входной вектор классифицируется как класс 1; в противном случае это класс 2.

Вы можете заметить, что первые два конкурентных нейрона соединены с первым линейным нейроном (с весом 1), в то время как вторые два конкурентных нейрона соединены со вторым линейным нейроном. Все остальные веса между конкурентными нейронами и линейными нейронами имеют значения 0. Таким образом, каждый из двух целевых классов (линейных нейронов) фактически является объединением двух подклассов (конкурентных нейронов).

Вы можете смоделировать сеть с [sim](https://translated.turbopages.org/proxy_u/en-ru.ru.a34c63d0-624452c2-8cb32aab-74722d776562/https/www.mathworks.com/help/deeplearning/ref/sim.html)помощью . Используйте исходную Pматрицу в качестве входных данных, чтобы посмотреть, что вы получите.

Y = нетто(P);

Yc = vec2ind(Y)

Yc =

1 1 1 1 1 1 1 1 1 1

Сеть классифицирует все входные данные в класс 1. Поскольку это не то, что вы хотите, вы должны обучить сеть (корректируя веса только уровня 1), прежде чем вы сможете ожидать хорошего результата. В следующих двух разделах обсуждаются два правила обучения LVQ и процесс обучения.

### Правило обучения LVQ1 (learnlv1)

Обучение LVQ на конкурентном уровне основано на наборе пар вход/цель.

Каждый целевой вектор имеет один 1. Остальные его элементы равны 0. Цифра 1 указывает на правильную классификацию связанных входных данных. Например, рассмотрим следующую обучающую пару.

Здесь есть входные векторы из трех элементов, и каждый входной вектор должен быть отнесен к одному из четырех классов. Сеть должна быть обучена таким образом, чтобы она классифицировала входной вектор, показанный выше, в третий из четырех классов.

Для обучения сети задается входной вектор **p**, а расстояние от **p** до каждой строки матрицы входного веса **IW**1,1 вычисляется с помощью функции [negdist](https://translated.turbopages.org/proxy_u/en-ru.ru.a34c63d0-624452c2-8cb32aab-74722d776562/https/www.mathworks.com/help/deeplearning/ref/negdist.html). Скрытые нейроны слоя 1 конкурируют друг с другом. Предположим, что i-й элемент **n**1 наиболее положителен, и нейрон i\* выигрывает соревнование. Тогда конкурентная передаточная функция производит a 1 как i\* - й элемент **a**1. Все остальные элементы **a**1 равны 0.

Когда **a**1 умножается на веса слоя 2 **LW**2,1, единственный 1 в **a**1 выбирает класс k\*, связанный с входным сигналом. Таким образом, сеть отнесла входной вектор **p** к классу k\* и α 2k \* будет равен 1. Конечно, это назначение может быть хорошим или плохим, ибо tk\* может быть 1 или 0, в зависимости от того, принадлежал ли вход к классу k\* или нет.

Отрегулируйте i\* - ю строку **IW**1,1 таким образом, чтобы переместить эту строку ближе к входному вектору **p**, если назначение правильное, и переместить строку подальше от **p**, если назначение неправильное. Если **p** классифицируется правильно,

вычислите новое значение i\* - й строки **IW**1,1 следующим образом:

С другой стороны, если **p** классифицируется неправильно,

вычислите новое значение i\* - й строки **IW**1,1 следующим образом:

Вы можете внести эти исправления в i\*-ю строку **IW**1,1 автоматически, не затрагивая другие строки **IW**1,1, путем обратного распространения выходных ошибок на уровень 1.

Такие поправки перемещают скрытый нейрон к векторам, попадающим в класс, для которого он образует подкласс, и от векторов, попадающих в другие классы.

Функция обучения, реализующая эти изменения весов уровня 1 в сетях LVQ, такова[learnlv1](https://translated.turbopages.org/proxy_u/en-ru.ru.a34c63d0-624452c2-8cb32aab-74722d776562/https/www.mathworks.com/help/deeplearning/ref/learnlv1.html): Его можно применять во время тренировок.

### Обучение

Затем вам нужно обучить сеть, чтобы получить веса первого уровня, которые приводят к правильной классификации входных векторов. Вы делаете это с [train](https://translated.turbopages.org/proxy_u/en-ru.ru.a34c63d0-624452c2-8cb32aab-74722d776562/https/www.mathworks.com/help/deeplearning/ref/network.train.html)помощью следующих команд. Во-первых, установите тренировочные эпохи равными 150. Затем используйте[train](https://translated.turbopages.org/proxy_u/en-ru.ru.a34c63d0-624452c2-8cb32aab-74722d776562/https/www.mathworks.com/help/deeplearning/ref/network.train.html):

net.trainParam.epochs = 150;

net = поезд(net,P,T);

Теперь подтвердите веса первого слоя.

net.IW{1,1}

ans =

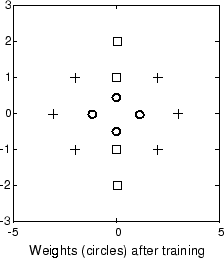
0.3283 0.0051

-0.1366 0.0001

-0.0263 0.2234

0 -0.0685

Следующий график показывает, что эти веса переместились в соответствующие классификационные группы.



Чтобы подтвердить, что эти веса действительно приводят к правильной классификации, возьмите матрицу Pв качестве входных данных и смоделируйте сеть. Затем посмотрите, какие классификации создаются сетью.

Y = нетто(P);

Yc = vec2ind(Y)

Это дает

Yc =

1 1 1 2 2 2 2 1 1 1

что вполне ожидаемо. В качестве последней проверки попробуйте ввести данные, близкие к вектору, который использовался при обучении.

pchk1 = [0; 0,5];

Y = нетто(pchk1);

Yc1 = vec2ind(Y)

Это дает

Yc1 =

2

Это выглядит правильно, потому pchk1что близко к другим векторам, классифицированным как 2. Аналогично,

pchk2 = [1; 0];

Y = нетто(pchk2);

Yc2 = vec2ind(Y)

дает

Yc2 =

1

Это тоже выглядит правильно, потому pchk2что близко к другим векторам, классифицированным как 1.

Возможно, вы захотите попробовать пример программы [Обучения векторному квантованию](https://translated.turbopages.org/proxy_u/en-ru.ru.a34c63d0-624452c2-8cb32aab-74722d776562/https/www.mathworks.com/help/deeplearning/ug/learning-vector-quantization.html). Это следует из приведенного выше обсуждения обучения.

### Дополнительное правило обучения LVQ2.1 (learnlv2)

Следующее правило обучения может быть применено после первого применения LVQ1. Это может улучшить результат первого обучения. Эта конкретная версия LVQ2 (называемая в литературе LVQ2.1 [[Koho97](https://translated.turbopages.org/proxy_u/en-ru.ru.a34c63d0-624452c2-8cb32aab-74722d776562/https/www.mathworks.com/help/deeplearning/ug/shallow-neural-networks-bibliography.html)]) воплощена в функции [learnlv2](https://translated.turbopages.org/proxy_u/en-ru.ru.a34c63d0-624452c2-8cb32aab-74722d776562/https/www.mathworks.com/help/deeplearning/ref/learnlv2.html). Еще раз обратите внимание, что LVQ2.1 следует использовать только после применения LVQ1.

Обучение здесь аналогично тому, что происходит в [learnlv2](https://translated.turbopages.org/proxy_u/en-ru.ru.a34c63d0-624452c2-8cb32aab-74722d776562/https/www.mathworks.com/help/deeplearning/ref/learnlv2.html)за исключением того, что теперь два вектора слоя 1, которые находятся ближе всего к входному вектору, могут быть обновлены, при условии, что один принадлежит правильному классу, а другой принадлежит неправильному классу, и далее при условии, что входные данные попадают в “окно” вблизи средней плоскости двух векторов.

Окно определяется

где

(где di и dj - евклидовы расстояния **p** от i\***IW**1,1 и j\***IW**1,1 соответственно). Возьмем значение w в диапазоне от 0,2 до 0,3. Если вы выберете, например, 0.25, то s = 0.6. Это означает, что если минимальное из двух соотношений расстояний больше 0,6, то два вектора корректируются. То есть, если вход находится вблизи средней плоскости, отрегулируйте два вектора, при условии также, что входные векторы **p** и j\***IW**1,1 принадлежат к одному классу, а **p** и i\***IW**1,1 не принадлежат к одному классу.

Внесенные коррективы таковы:

и

Таким образом, при наличии двух векторов, ближайших к входу, до тех пор, пока один принадлежит неправильному классу, а другой-правильному, и до тех пор, пока вход попадает в окно средней плоскости, эти два вектора корректируются. Такая процедура позволяет вектору, который едва ли правильно классифицирован с помощью LVQ1, быть перемещенным еще ближе к входу, поэтому результаты более надежны.

| **Функция** | **Описание** |
| --- | --- |
| [competlayer](https://translated.turbopages.org/proxy_u/en-ru.ru.a34c63d0-624452c2-8cb32aab-74722d776562/https/www.mathworks.com/help/deeplearning/ref/competlayer.html) | Создайте конкурентный слой. |
| [learnk](https://translated.turbopages.org/proxy_u/en-ru.ru.a34c63d0-624452c2-8cb32aab-74722d776562/https/www.mathworks.com/help/deeplearning/ref/learnk.html) | Правило обучения Кохонена. |
| [selforgmap](https://translated.turbopages.org/proxy_u/en-ru.ru.a34c63d0-624452c2-8cb32aab-74722d776562/https/www.mathworks.com/help/deeplearning/ref/selforgmap.html) | Создайте самоорганизующуюся карту. |
| [learncon](https://translated.turbopages.org/proxy_u/en-ru.ru.a34c63d0-624452c2-8cb32aab-74722d776562/https/www.mathworks.com/help/deeplearning/ref/learncon.html) | Функция обучения смещению совести. |
| [boxdist](https://translated.turbopages.org/proxy_u/en-ru.ru.a34c63d0-624452c2-8cb32aab-74722d776562/https/www.mathworks.com/help/deeplearning/ref/boxdist.html) | Расстояние между двумя векторами положения. |
| [dist](https://translated.turbopages.org/proxy_u/en-ru.ru.a34c63d0-624452c2-8cb32aab-74722d776562/https/www.mathworks.com/help/deeplearning/ref/dist.html) | Весовая функция евклидова расстояния. |
| [linkdist](https://translated.turbopages.org/proxy_u/en-ru.ru.a34c63d0-624452c2-8cb32aab-74722d776562/https/www.mathworks.com/help/deeplearning/ref/linkdist.html) | Функция расстояния связи. |
| [mandist](https://translated.turbopages.org/proxy_u/en-ru.ru.a34c63d0-624452c2-8cb32aab-74722d776562/https/www.mathworks.com/help/deeplearning/ref/mandist.html) | Функция веса манхэттенского расстояния. |
| [gridtop](https://translated.turbopages.org/proxy_u/en-ru.ru.a34c63d0-624452c2-8cb32aab-74722d776562/https/www.mathworks.com/help/deeplearning/ref/gridtop.html) | Функция топологии слоя Gridtop. |
| [hextop](https://translated.turbopages.org/proxy_u/en-ru.ru.a34c63d0-624452c2-8cb32aab-74722d776562/https/www.mathworks.com/help/deeplearning/ref/hextop.html) | Функция топологии шестиугольного слоя. |
| [randtop](https://translated.turbopages.org/proxy_u/en-ru.ru.a34c63d0-624452c2-8cb32aab-74722d776562/https/www.mathworks.com/help/deeplearning/ref/randtop.html) | Функция топологии случайного слоя. |
| [lvqnet](https://translated.turbopages.org/proxy_u/en-ru.ru.a34c63d0-624452c2-8cb32aab-74722d776562/https/www.mathworks.com/help/deeplearning/ref/lvqnet.html) | Создайте обучающую векторную квантованную сеть. |
| [learnlv1](https://translated.turbopages.org/proxy_u/en-ru.ru.a34c63d0-624452c2-8cb32aab-74722d776562/https/www.mathworks.com/help/deeplearning/ref/learnlv1.html) | Функция обучения весу LVQ1. |
| [learnlv2](https://translated.turbopages.org/proxy_u/en-ru.ru.a34c63d0-624452c2-8cb32aab-74722d776562/https/www.mathworks.com/help/deeplearning/ref/learnlv2.html) | Функция обучения весу LVQ2. |

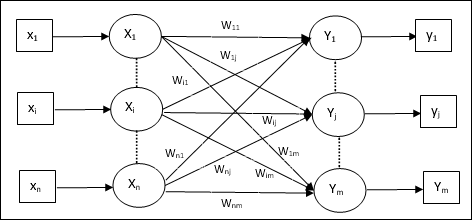
[Нейронные сети обучения векторному квантованию (LVQ) - MATLAB & Simulink (turbopages.org)](https://translated.turbopages.org/proxy_u/en-ru.ru.a34c63d0-624452c2-8cb32aab-74722d776562/https/www.mathworks.com/help/deeplearning/ug/learning-vector-quantization-lvq-neural-networks-1.html)

**Изучение вектора квантования**

Векторное квантование обучения (LVQ), отличающееся от векторного квантования (VQ) и самоорганизующихся карт Кохонена (KSOM), в основном представляет собой конкурентную сеть, в которой используется контролируемое обучение. Мы можем определить его как процесс классификации шаблонов, где каждая единица вывода представляет класс. Поскольку он использует контролируемое обучение, сети будет предоставлен набор обучающих шаблонов с известной классификацией наряду с начальным распределением выходного класса. После завершения процесса обучения LVQ классифицирует входной вектор, назначая его тому же классу, что и выходной блок.

**Архитектура**

На следующем рисунке показана архитектура LVQ, которая очень похожа на архитектуру KSOM. Как мы видим, существует **«n»** количество единиц ввода и **«m»** количество единиц вывода. Слои полностью взаимосвязаны с весом на них.



**Используемые параметры**

Ниже приведены параметры, используемые в процессе обучения LVQ, а также в блок-схеме

* **x** = вектор обучения (x 1 , …, x i , …, x n )
* **T** = класс для обучающего вектора **x**
* **w j** = весовой вектор для **j- й** единицы измерения
* **C j** = класс, связанный с **j- й** единицей вывода

**x** = вектор обучения (x 1 , …, x i , …, x n )

**T** = класс для обучающего вектора **x**

**w j** = весовой вектор для **j- й** единицы измерения

**C j** = класс, связанный с **j- й** единицей вывода

**Алгоритм обучения**

**Шаг 1** — Инициализировать опорные векторы, что можно сделать следующим образом:

* **Шаг 1 (а)** — Из данного набора тренировочных векторов взять первые « **m** » (количество кластеров) тренировочных векторов и использовать их в качестве весовых векторов. Остальные векторы могут быть использованы для обучения.
* **Шаг 1 (б)** — Присвойте начальный вес и классификацию случайным образом.
* **Шаг 1 (с)** — Применение метода кластеризации K-средних.

**Шаг 1 (а)** — Из данного набора тренировочных векторов взять первые « **m** » (количество кластеров) тренировочных векторов и использовать их в качестве весовых векторов. Остальные векторы могут быть использованы для обучения.

**Шаг 1 (б)** — Присвойте начальный вес и классификацию случайным образом.

**Шаг 1 (с)** — Применение метода кластеризации K-средних.

**Шаг 2** — Инициализировать опорный вектор  alpha

**Шаг 3** — Продолжайте с шагов 4-9, если условие остановки этого алгоритма не выполняется.

**Шаг 4** — Выполните шаги 5-6 для каждого входного вектора обучения **x** .

**Шаг 5** — Рассчитать квадрат евклидова расстояния для **j = 1 до m** и **i = от 1 до n**

D(j)= displaystyle sum limitni=1 displaystyle sum limitmj=1(xi−wij)2

**Шаг 6** — Получить выигрышную единицу **J,** где **D (j)** минимально.

**Шаг 7** — Рассчитайте новый вес выигрышной единицы по следующему соотношению —

если **T = C j,** то wj(новый)=wj(старый)+ alpha[x−wj(старый)]

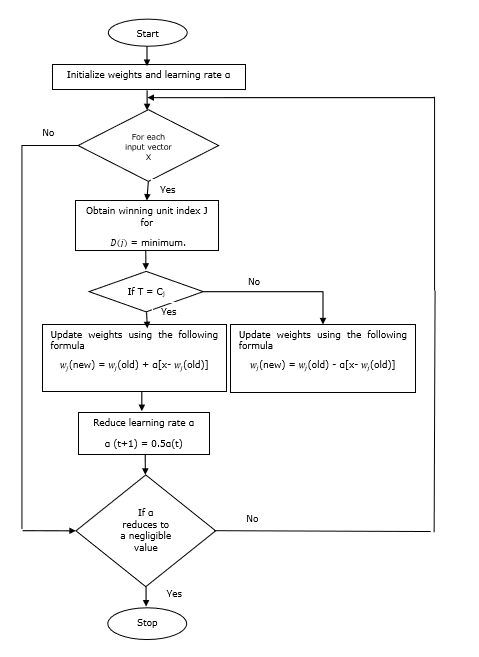
если **T ≠ C j,** то wj(новый)=wj(старый)− alpha[x−wj(старый)]

**Шаг 8** — Уменьшите скорость обучения  alpha.

**Шаг 9** — Проверка состояния остановки. Это может быть следующим:

* Максимальное количество эпох достигнуто.
* Скорость обучения снижена до незначительного значения.

**блок-схема**



**Варианты**

Три других варианта, а именно LVQ2, LVQ2.1 и LVQ3, были разработаны Кохоненом. Сложность во всех этих трех вариантах из-за концепции, которую изучат победитель и юнит, занявший второе место, больше, чем в LVQ.

**LVQ2**

Как уже обсуждалось, концепция других вариантов LVQ выше, состояние LVQ2 формируется окном. Это окно будет основано на следующих параметрах —

* **х** — текущий входной вектор
* **y c** — опорный вектор, ближайший к **x**
* **y r** — другой опорный вектор, ближайший к **x**
* **d c** — расстояние от **x** до **y c**
* **d r** — расстояние от **x** до **y r**

**х** — текущий входной вектор

**y c** — опорный вектор, ближайший к **x**

**y r** — другой опорный вектор, ближайший к **x**

**d c** — расстояние от **x** до **y c**

**d r** — расстояние от **x** до **y r**

Входной вектор **х** попадает в окно, если

 гидроразрываDCDг>1− тетаи гидроразрываDгDс>1 + Theta

Здесь  theta — количество обучающих образцов.

Обновление можно выполнить по следующей формуле:

У−с(т +1)=у−C(т) + альфа(т)[х(т)−у−C(t)] **(принадлежит к другому классу)**

У−г(т +1)=у−г(т) + альфа(т)[х(т)−у−г(t)] **(принадлежит к тому же классу)**

Здесь  alpha — скорость обучения.

**LVQ2.1**

В LVQ2.1 мы возьмем два ближайших вектора, а именно **y c1** и **y c2,** и условие для окна следующее:

Min начинаютсяbmatrix гидроразрываDc1Dc2, гидроразрываDc2Dc1 конецbmatrix> :(1 :− Theta)

Max начинаютсяbmatrix гидроразрываDc1Dc2, гидроразрываDc2Dc1 конецbmatrix < :(1 :+ тета)

Обновление можно выполнить по следующей формуле:

У−c1(т +1)=у−c1(т) + альфа(т)[х(т)−у−c1(t)] **(принадлежит к другому классу)**

У−с2(т +1)=у−с2(т) + альфа(т)[х(т)−у−с2(t)] **(принадлежит к тому же классу)**

Здесь  alpha — это скорость обучения.

**LVQ3**

В LVQ3 мы возьмем два ближайших вектора, а именно **y c1** и **y c2,** и условие для окна будет следующим:

Min начинаютсяbmatrix гидроразрываDc1Dc2, гидроразрываDc2Dc1 конецbmatrix> :(1 :− тета)(1 + тета)

Здесь  theta ок.0,2

Обновление можно выполнить по следующей формуле:

У−c1(т +1)=у−c1(т) + бета(т)[х(т)−у−c1(t)] **(принадлежит к другому классу)**

У−с2(т +1)=у−с2(т) + бета(т)[х(т)−у−с2(t)] **(принадлежит к тому же классу)**

Здесь  beta — это кратное значение скорости обучения  alpha и **beta=m alpha(t)** для каждых **0,1 <m <0,5**

**Изучение векторного квантования**

Из Википедии, бесплатной энциклопедии

[Перейти к навигации](https://www.hmong.press/wiki/Learning_vector_quantization#mw-head)[Перейти к поиску](https://www.hmong.press/wiki/Learning_vector_quantization#searchInput)

В [информатике](https://www.hmong.press/wiki/Computer_science) , **изучение векторного квантования** ( **LVQ** ), является [прототипом на основе](https://www.hmong.press/wiki/Prototype)[контролируемой](https://www.hmong.press/wiki/Supervised_learning)[классификации](https://www.hmong.press/wiki/Statistical_classification)[алгоритм](https://www.hmong.press/wiki/Algorithm) . LVQ - это контролируемый аналог систем [векторного квантования](https://www.hmong.press/wiki/Vector_quantization" \o "Векторное квантование) .

**СОДЕРЖАНИЕ**

* [1 Обзор](https://www.hmong.press/wiki/Learning_vector_quantization#Overview)
* [2 Алгоритм](https://www.hmong.press/wiki/Learning_vector_quantization#Algorithm)
* [3 ссылки](https://www.hmong.press/wiki/Learning_vector_quantization#References)
* [4 Дальнейшее чтение](https://www.hmong.press/wiki/Learning_vector_quantization#Further_reading)
* [5 Внешние ссылки](https://www.hmong.press/wiki/Learning_vector_quantization#External_links)

**Обзор [ [править](https://en.wikipedia.org/w/index.php?title=Learning_vector_quantization&action=edit&section=1" \o "Раздел редактирования: Обзор" \t "_blank)]**

LVQ можно рассматривать как частный случай [искусственной нейронной сети](https://www.hmong.press/wiki/Artificial_neural_network) , точнее говоря, он применяет подход, основанный на [изучении Hebbian по принципу «](https://www.hmong.press/wiki/Hebbian_learning)[победитель получает все»](https://www.hmong.press/wiki/Winner-take-all_(computing)) . Он является предшественником [самоорганизующихся карт](https://www.hmong.press/wiki/Self-organizing_map) (SOM) и связан с [нейронным газом](https://www.hmong.press/wiki/Neural_gas) , а также с [алгоритмом k-ближайшего соседа](https://www.hmong.press/wiki/K-nearest_neighbor_algorithm) (k-NN). LVQ был изобретен [Теуво Кохоненом](https://www.hmong.press/wiki/Teuvo_Kohonen) . [[1]](https://www.hmong.press/wiki/Learning_vector_quantization#cite_note-1)

Система LVQ представлена ​​прототипами, которые определены в [пространстве признаков](https://www.hmong.press/wiki/Feature_space) наблюдаемых данных. В алгоритмах обучения по принципу «победитель получает все» для каждой точки данных определяется прототип, ближайший к входу в соответствии с заданной мерой расстояния. Положение этого так называемого прототипа победителя затем адаптируется, т.е. победитель перемещается ближе, если он правильно классифицирует точку данных, или удаляется, если он классифицирует точку данных неправильно.

Преимущество LVQ в том, что он создает прототипы, которые легко интерпретировать специалистам в соответствующей области приложения. [[2]](https://www.hmong.press/wiki/Learning_vector_quantization#cite_note-2) LVQ-системы могут применяться к мультиклассовым задачам естественным образом. Он используется во множестве практических приложений. См. [«Библиографию по самоорганизующейся карте (SOM) и векторному квантованию обучения (LVQ)](http://liinwww.ira.uka.de/bibliography/Neural/SOM.LVQ.html) ».

Ключевой проблемой LVQ является выбор подходящей меры расстояния или сходства для обучения и классификации. Недавно были разработаны методы, которые адаптируют параметризованную меру расстояния в процессе обучения системы, см., Например, (Schneider, Biehl, and Hammer, 2009) [[3]](https://www.hmong.press/wiki/Learning_vector_quantization#cite_note-3) и ссылки в них.

LVQ может стать большим подспорьем при классификации текстовых документов. [[*необходима цитата*](https://www.hmong.press/wiki/Wikipedia:Citation_needed)]

**Алгоритм [ [править](https://en.wikipedia.org/w/index.php?title=Learning_vector_quantization&action=edit&section=2" \o "Раздел редактирования: Алгоритм" \t "_blank)]**

Ниже следует неофициальное описание.  
Алгоритм состоит из трех основных шагов. Входные данные алгоритма:

* сколько нейронов будет в системе (в простейшем случае равно количеству классов)
* какой вес каждый нейрон имеет для
* соответствующая метка каждому нейрону
* как быстро нейроны обучаются
* и входной список, содержащий все векторы, метки которых уже известны (обучающий набор).

Последовательность алгоритма:

1. Для следующего ввода (с меткой ) найдите ближайший нейрон , т.е. где используется метрика ( [евклидова](https://www.hmong.press/wiki/Euclidean_distance) и т. Д.).
2. Обновить . Лучшее объяснение - приблизиться к входу , если и принадлежат одному ярлыку, и разделить их дальше, если они этого не делают. если (ближе друг к другу) или если (дальше друг от друга).
3. Пока остались векторы, перейдите к шагу 1, иначе завершите работу.

Примечание: и - [векторы](https://www.hmong.press/wiki/Vector_space) в пространстве признаков. Более формальное описание можно найти здесь: <http://jsalatas.ictpro.gr/implementation-of-competitive-learning-networks-for-weka/>

**Ссылки [ [править](https://en.wikipedia.org/w/index.php?title=Learning_vector_quantization&action=edit&section=3" \o "Раздел редактирования: Ссылки" \t "_blank)]**

1. [**^**](https://www.hmong.press/wiki/Learning_vector_quantization#cite_ref-1) Т. Кохонен. Самоорганизующиеся карты. Спрингер, Берлин, 1997.
2. [**^**](https://www.hmong.press/wiki/Learning_vector_quantization#cite_ref-2) *Т. Кохонен (1995), "Изучение векторного квантования", в М.А. Арбибе (ред.), Справочник по теории мозга и нейронных сетей , Кембридж, Массачусетс: MIT Press, стр. 537–540*
3. [**^**](https://www.hmong.press/wiki/Learning_vector_quantization#cite_ref-3) *П. Шнайдер, Б. Хаммер и М. Биль (2009). «Адаптивные матрицы релевантности в обучающем векторном квантовании». Нейронные вычисления .****21****(10): 3532–3561. [CiteSeerX](https://www.hmong.press/wiki/CiteSeerX_(identifier)" \o "CiteSeerX (идентификатор))*[*10.1.1.216.1183*](https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.216.1183)*.*[*DOI*](https://www.hmong.press/wiki/Doi_(identifier))*:*[*10.1162 / neco.2009.10-08-892*](https://doi.org/10.1162%2Fneco.2009.10-08-892)*.*[*PMID*](https://www.hmong.press/wiki/PMID_(identifier))[*19635012*](https://pubmed.ncbi.nlm.nih.gov/19635012)*.*[*S2CID*](https://www.hmong.press/wiki/S2CID_(identifier))[*17306078*](https://api.semanticscholar.org/CorpusID:17306078)*.*