**ДОСТОИНСТВА АЛГОРИТМА**

* Достаточно высокая эффективность.
* Через каждый нейрон проходит информация только о связных с ним нейронах, поэтому этот алгоритм легко реализуется на вычислительных устройствах с параллельной архитектурой.
* Высокая степень общности. Алгоритм легко записать для произвольного числа слоёв, произвольной размерности входов и выходов, произвольной функции потерь и произвольных функций активации.
* Этот метод не накладывает никаких ограничений на используемый метод оптимизации. Его можно применять вместе с методом скорейшего спуска, сопряженных градиентов, Ньютона-Рафсона и др.

**НЕДОСТАТКИ АЛГОРИТМА**

* Несмотря на многочисленные успешные применения обратного распространения, оно не является универсальным решением.
* Больше всего неприятностей приносит неопределённо долгий процесс обучения.
* В сложных задачах для обучения сети могут потребоваться дни или даже недели, она может и вообще не обучиться.
* Причиной может быть одна из описанных ниже.

**Паралич сети**

* В процессе обучения сети значения весов могут в результате коррекции стать очень большими величинами.
* Это может привести к тому, что все или большинство нейронов будут функционировать при очень больших значениях OUT, в области, где производная сжимающей функции очень мала.
* Так как посылаемая обратно в процессе обучения ошибка пропорциональна этой производной, то процесс обучения может практически замерить.

**Локальные минимумы.**

* Обратное распространение использует разновидность градиентного спуска, то есть осуществляет спуск вниз по поверхности ошибки, непрерывно подстраивая веса в направлении к минимуму.
* Поверхность ошибки сложной сети сильно изрезана и состоит из холмов, долин, складок и оврагов в пространстве высокой размерности.

**Размер шага**

* Внимательный разбор доказательства сходимости показывает, что коррекции весов предполагаются бесконечно малыми.
* Это неосуществимо на практике, так как ведёт к бесконечному времени обучения.
* Размер шага должен браться конечным.
* Если размер шага фиксирован и очень мал, то сходимость слишком медленная, если же он фиксирован и слишком велик, то может возникнуть паралич или постоянная устойчивость.
* Эффективно увеличивать шаг до тех пор, пока не прекратится улучшение оценки в данном направлении антиградиента и уменьшать, если такого улучшения не происходит.
* **П. Д. Вассерман** описал адаптивный алгоритм выбора шага, автоматически корректирующий размер шага в процессе обучения.
* В книге **А. Н. Горбаня** предложена разветвлённая технология оптимизации обучения.
* Есть **возможность переобучения сети**, что является скорее результатом ошибочного проектирования её топологии.
* При слишком большом количестве нейронов теряется свойство сети обобщать информацию.
* Весь набор образов, предоставленных к обучению, будет выучен сетью, но любые другие образы, даже очень похожие, могут быть **классифицированы** неверно.

**НЕЙРОСЕТИ НА ПРИМЕРЕ РАСПОЗНАВАНИЯ ЦИФР**

**Постановка задачи распознавания цифр**

Представим, у нас есть **число** **3**, изображенное в чрезвычайно низком **разрешением 28х28** пикселей.

Наш мозг без труда узнает это число.

Поэтому задача распознавания цифр в таком контексте воспринимается как простая.

Нейронная сеть принимает на вход изображение любой цифры в виде массива **28х28 пикселей** и выдает на выходе саму «сущность» - **цифру от 0 до 9**

**Активация нейронов. Слой нейросети**

Так как наша сетка состоит из **28x28=784 пикселей**, пусть есть 784 нейрона, содержащие различные числа **от до 1**: чем ближе пиксель к белому цвету, тем ближе соответствующее число к единице.

Эти заполняющие сетку числа называются **активациями нейронов.**

* Описанные **784 нейрона** образуют **первый слой нейросети.**
* Последний слой содержит **10 нейронов**, каждый из которых соответствует одной из десяти цифр.
* В этих числах активация это также число от нуля до единицы, отражающее насколько система уверена, что входное изображение содержит соответствующую цифру.

**Назначение скрытых слоев**

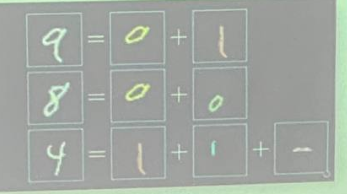
* Выбор количества скрытых слоев и содержащихся в них нейронов произволен однако обычно они выбираются из определённых представлений о задаче, решаемой нейронной сетью.
* Принцип работы нейросети состоит в том, что активация в одном слое определяет активацию в следующем.
* Возбуждаясь, некоторая группа нейронов вызывает возбуждение другой группы.

В процессе распознавания цифр мы сводим воедино различные компоненты.

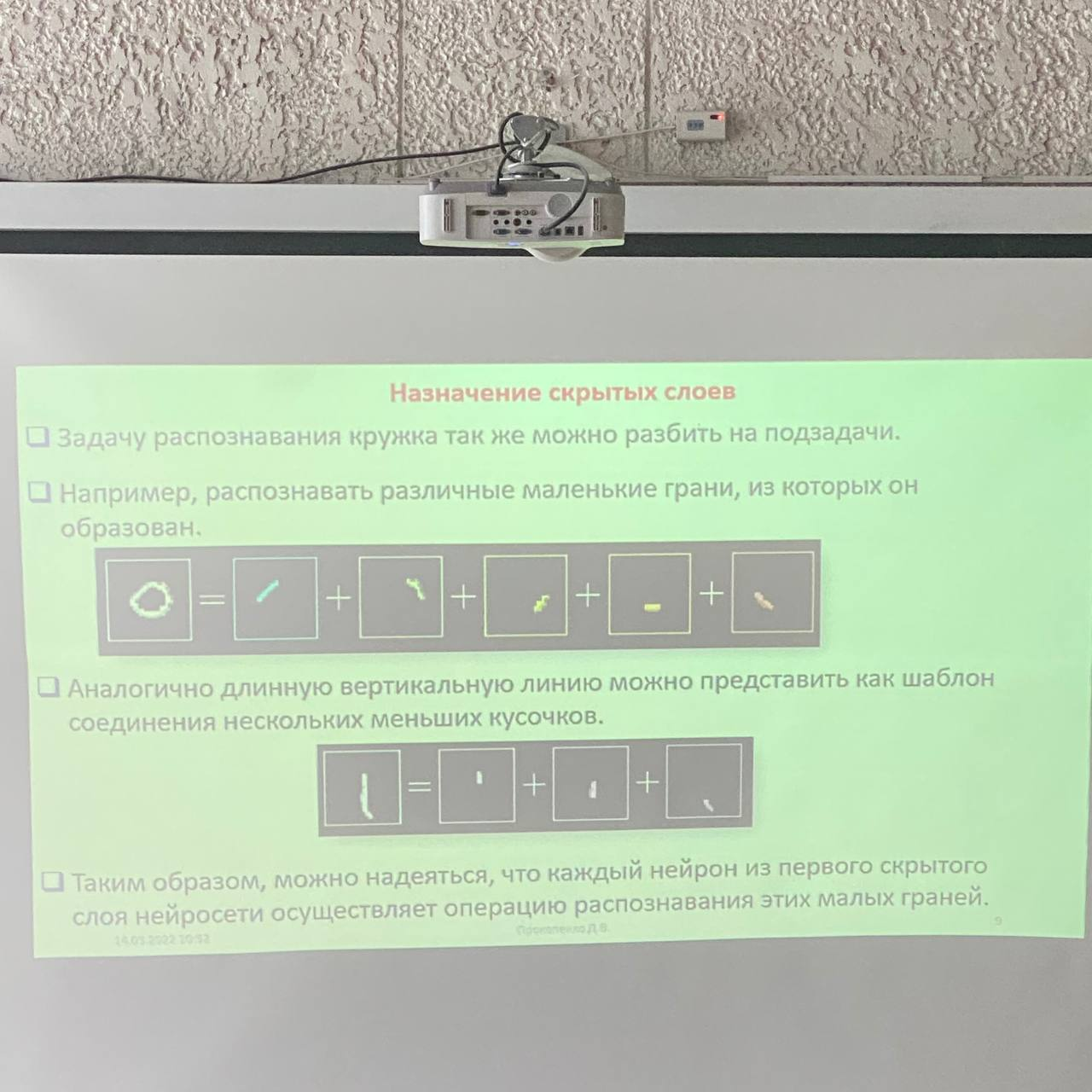
Например, девятка состоит из кружка сверху и линии справа.

Восьмерка также имеет кружок вверху, но вместо линии справа, у неё есть парный кружок снизу.

Четверку можно представить как три определённым образом соединённые линии.



* В идеализированном случае можно ожидать, что каждый нейрон из второго слоя соотносится с одним из этих компонентов.
* И когда передается нейросети изображение с кружком в верхней части, существует определённый нейрон, чья активация станет ближе единице.
* Таким образом, переход от второго скрытого слоя к выходному соответствует знаниям о том, какой набор компонентов какой цифре соответствует.



Задачу распознавания кружка так же можно разбить на подзадачи.

Например, распознавать различные маленькие грани, из которых он образован.



Аналогично длинную вертикальную линию можно представить как шаблон соединения нескольких маленьких кусочков.



Таким образом, можно надеяться, что каждый нейрон из первого скрытого слоя нейросети осуществляет операцию распознавания этих малых граней.

**Определение области распознавания**

Пусть цель отдельного нейрона в первом скрытом слое это определить,содержит ли картинка грань в отмеченной на рисунке области.

Первый вопрос: какие параметры настройки должны быть у нейросети, чтобы иметь возможность обнаружить этот шаблон или любой другой шаблон из пикселей.

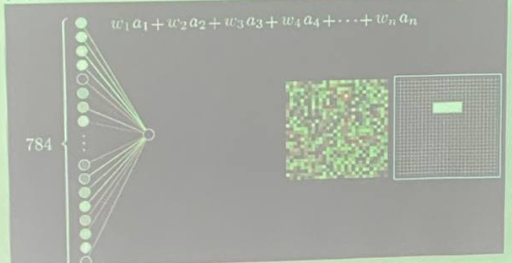
Назначим числовой вес wi каждому соединению между нашим нейроном и нейроном из входного слоя.

Затем возьмем все активации из первого слоя и посчитаем их взвешенную сумму согласно этим весам.

Так как количество весов такое же, как и число активаций, им также можно сопоставить аналогичную сетку.

Обозначим **зелеными** пикселями **положительные веса**, а **красными - отрицательные**.

Яркость пикселя будет соответствовать абсолютному значению веса.



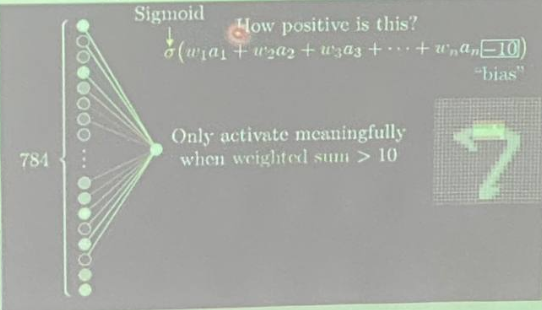
Теперь, если мы установим все веса равными нулю, кроме пикселей, соответствующих нашему шаблону, то взвешенная сумма сведется к суммированию значений активаций пикселей в интересующей нас области.

Если же мы хотим, определить есть ли там именно ребро, то можно добавить вокруг зеленого прямоугольника весов красные весовые грани, соответствующие **отрицательным весам.**

Тогда **взвешенная сумма** для этого участка будет максимальной, когда средние пиксели изображения в этой части ярче, а окружающих их пиксели темнее.

Таким образом, активация нейрона это по сути мера того, насколько **положительна соответствующая взвешенная сумма.**

Чтобы нейрон не активировался при малых положительных числах, можно добавить к взвешенной сумме некоторое **отрицательное число - сдвиг**, определяющий насколько большой должна быть взвешенная сумма, чтобы активировать нейрон.



**Обучение нейронной сети для распознавания цифр**

В результате обучения нейросеть должна правильным образом различать числа из ранее не представленных, тестовых данных.

Соответственно в качестве проверки обучения нейросети можно использовать отношение числа актов корректного распознавания цифр к количеству элементов тестовой выборки.

Откуда берутся данные для обучения?

Рассматриваемая задача очень распространена, и для ее решения **создана крупная база данных MNIST**, состоящая из 60 тыс. размеченных данных и 10 тыс. тестовых изображений.

**Пример сети**

Простая модель в который сигналы передаются с лева на право и все нейроны соединены друг с другом.

Каждый нейрон входного слоя соединен выходами со всеми нейронами скрытого слоя и т.д.

Первым делом нам надо задать число нейронов в входном, выходном и скрытом слое.

Еще нужно задать коэффициент обучения (с какой скоростью будет обучаться сеть от 0 до 1).