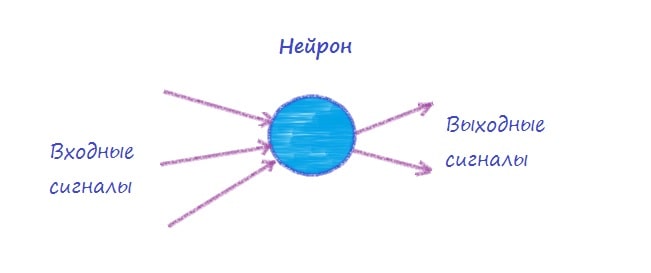
# Знакомство с нейронной сетью.

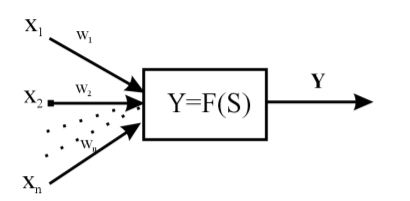
## Нейронная сеть - это компьютерная программа, работающая по принципу работы нервной системы человека.

Нейронная сеть представляет из себя совокупность нейронов, которые соединены между собой определённым образом(каждый нейрон, как отдельная программа получает на вход данные и возвращает результат)

Нейрон представляет из себя элемент, который вычисляет выходной сигнал (по определённому правилу) из совокупности входных сигналов.



У нейрона есть несколько входных каналов и только один выходной канал. По входным каналам на нейрон поступают данные задачи, а на выходе формируется результат работы. Нейрон вычисляет взвешенную сумму входных сигналов, а затем преобразует полученную сумму с помощью заданной нелинейной функции. Множество, состоящее из порогового уровня и всех весов, называют параметрами нейрона.



Здесь введены следующие обозначения: X1,X2, …, Xn - входной сигнал (паттерн), w1, w2,…,wn – весовые коэффициенты, b- порог нейрона Сначала нейрон вычисляет взвешенную сумму S = sum(Wi Xi) - b, далее применяя функцию активации F(S) вычисляет выходной сигнал Y. Функция активации нейрона - это функция, которая вычисляет выходной сигнал нейрона. На вход этой функции подается сумма всех произведений сигналов и весов этих сигналов.

По совокупности поступающих на вход сети сигналов на выходе формируется выходной сигнал (или несколько выходных сигналов). То есть нейронную сеть упрощенно можно представить в виде чёрного ящика, у которого есть входы и выходы. А внутри этого ящика сидит огромное количество нейронов.

## Такая программа способна обучаться, используя подготовленные для обучения данные. Решение задач с использованием нейронной сети проходит в следующие 9 этапов:

* Сбор данных для обучения;
* Подготовка и нормализация данных;
* Выбор топологии сети;
* Экспериментальный подбор характеристик сети;
* Экспериментальный подбор параметров обучения;
* Собственно обучение;
* Проверка адекватности обучения;
* Корректировка параметров, окончательное обучение;
* Вербализация сети с целью дальнейшего использования.

# Работа нейронной сети.

## Рассмотрим подробнее некоторые этапы.

Сбор данных для обучения:

Набор данных для обучения должен удовлетворять нескольким критериям:

* Репрезентативность — данные должны иллюстрировать истинное положение вещей в предметной области;
* Непротиворечивость — противоречивые данные в обучающей выборке приведут к плохому качеству обучения сети.

Подготовка и нормализация данных:

Исходные данные преобразуются к виду, в котором их можно подать на входы сети. Каждая запись в файле данных называется *обучающей парой* или *обучающим вектором*. Обучающий вектор содержит по одному значению на каждый вход сети и, в зависимости от типа обучения (с учителем или без), по одному значению для каждого выхода сети. Обучение сети на «сыром» наборе, как правило, не даёт качественных результатов. Существует ряд способов улучшить «восприятие» сети.

* *Нормировка* выполняется, когда на различные входы подаются данные разной размерности. Например, на первый вход сети подаются величины со значениями от нуля до единицы, а на второй — от ста до тысячи. При отсутствии нормировки значения на втором входе будут всегда оказывать существенно большее влияние на выход сети, чем значения на первом входе. При нормировке размерности всех входных и выходных данных сводятся воедино;
* Квантование выполняется над непрерывными величинами, для которых выделяется конечный набор дискретных значений. Например, квантование используют для задания частот звуковых сигналов при распознавании речи;
* Фильтрация выполняется для «зашумленных» данных.

Выбор топологии сети:

Выбирать тип сети следует, исходя из постановки задачи и имеющихся данных для обучения. Для обучения с учителем требуется наличие для каждого элемента выборки «экспертной» оценки. Иногда получение такой оценки для большого массива данных просто невозможно. В этих случаях естественным выбором является сеть, обучающаяся без учителя.

Экспериментальный подбор характеристик сети:

После выбора топологии сети нужно задать параметры обучения сети. Это может быть число слоёв, число блоков в скрытых слоях, наличие или отсутствие обходных соединений, а так же передаточные функции нейронов.

Обучение:

В процессе обучения сеть в определённом порядке просматривает обучающую выборку. Порядок просмотра может быть последовательным, случайным и т. д.Обучающие данные подаются сети для обучения, а проверочные используются для расчёта ошибки сети (проверочные данные никогда для обучения сети не применяются). Таким образом, если на проверочных данных ошибка уменьшается, то сеть действительно выполняет обобщение. Если ошибка на обучающих данных продолжает уменьшаться, а ошибка на тестовых данных увеличивается, значит, сеть перестала выполнять обобщение и просто «запоминает» обучающие данные. Это явление называется переобучением сети или оверфиттингом. В таких случаях обучение обычно прекращают.

Проверка адекватности обучения:

Даже в случае успешного, на первый взгляд, обучения сеть не всегда обучается именно тому, чего от неё хотел создатель. Известен случай, когда сеть обучалась распознаванию изображений танков по фотографиям, однако позднее выяснилось, что все танки были сфотографированы на одном и том же фоне. В результате сеть «научилась» распознавать этот тип ландшафта, вместо того, чтобы «научиться» распознавать танки. Таким образом, сеть «понимает» не то, что от неё требовалось, а то, что проще всего обобщить.

Тестирование качества обучения нейронной сети необходимо проводить на примерах, которые не участвовали в её обучении. При этом число тестовых примеров должно быть тем больше, чем выше качество обучения. Если ошибки нейронной сети имеют вероятность близкую к одной миллиардной, то и для подтверждения этой вероятности нужен миллиард тестовых примеров. Получается, что тестирование хорошо обученных нейронных сетей становится очень трудной задачей.

# Типы нейронных сетей.

## Нейронные сети разделяются на несколько типов. Разделение может зависеть от типа входных данных, от характера обучения, а так же по времени передачи сигнала.

По типу входных данных нейронные сети разделяются на:

* Аналоговые нейронные сети (используют информацию в форме действительных чисел);
* Двоичные нейронные сети (оперируют с информацией, представленной в двоичном виде);
* Образные нейронные сети (оперируют с информацией, представленной в виде образов: знаков, иероглифов, символов).

По характеру обучения нейронные сети разделяются на:

* Обучение с учителем — выходное пространство решений нейронной сети известно;
* Обучение без учителя— нейронная сеть формирует выходное пространство решений только на основе входных воздействий. Такие сети называют самоорганизующимися;
* Обучение с подкреплением — система назначения штрафов и поощрений от среды.

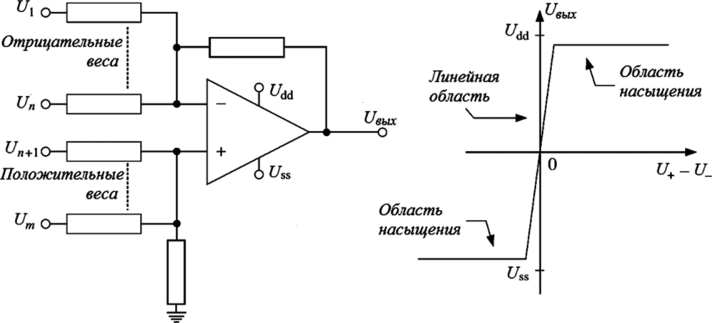
По характеру настройки синапсов нейронные сети разделяются на:

* Сети с фиксированными связями.
* Сети с динамическими связями.

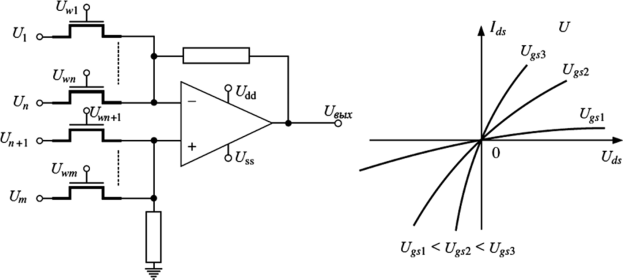
По времени передачи сигнала сети разделятся на синхронные(с временем передачи каждой связи равным 0 или константе) и асинхронные(время такой передачи может быть разное).

# Рассмотрим подробнее виды нейронных сетей, чтобы понять какую нейронную сеть лучше использовать для конкретной задачи.

Аналоговая нейронная сеть. Такие сети используют физические свойства полупроводниковых устройств, что даёт гораздо большую скорость выполнения поставленной задачи. Однако при большом количестве переменных конструкция такой нейронной сети будет очень сложна, например: параметры температуры, произведственные условия и т. д. Отсюда следует что аналоговую нейронную сеть можно использовать для быстрого выполнения задач с малым количеством переменных. Рассмотрим как же работает такая сеть.  
 Самый распространённый подход это использовать в качестве нейронов обычные операционные усилители и резисторы. Усилитель используется как функция активации и резистор определяет весовые значения.



Способ реализации с использованием резисторов обеспечивает очень хорошую линейность, но не гибкость, поскольку значения веса задаются в процессе производства и не могут быть впоследствии изменены. Существует более простой способ реализации: замена постоянных резисторов на МОП-транзисторы:



Каждый транзистор управляется напряжением на конденсаторе *Ugs,* которое должно периодически обновляться.Таким образом, обеспечивается гибкость реализации за счёт нелинейной зависимости. Данный способ позволяет изменять начальные веса, в отличие от первого способа.

Количество операционных усилителей на один чип ограниченно, из этого следует, что использование усилителей возможно лишь в небольших нейронных сетях.

Двоичные нейронные сети. Такие сети по своей сути гораздо проще чем аналоговые и другие. По сути они присваивают каждому событию вероятность либо 1 либо 0, в отличие от других, сетей которые могут присвоить другую вероятность. Из-за чего происходит потеря точности, но это можно компенсировать увеличением сети.

Огромный плюс такого подхода заключается в экономии места и энергии. Сравним с аналоговой нейронной сетью. Для хранения одной вероятности требуется всего 1 бит, в то время как в аналоговой целых 32.

Конечно даже в двоичных сетях встречаются иногда не бинарные слои. Обычно такое происходит на выходном или входном слоях, например изображение(полнота цветов записывается в не бинарном виде, как следствие нейронной сети придётся численно интерпретировать изображение).   
 Двоичные нейронные сети благодаря своей простоте, а так же малых затрат памяти и энергии широко применяются в коммерческой сфере, а так же с большой вероятности ещё большое применение в будущем.

### Образные нейронные схемы.

Такие нейронные схемы способны принимать информацию в виде образов, что позволяет тратить меньше времени на интерпретацию в числовой вид, как в аналоговых и двоичных сетях. Такие нейронные схемы не имеют широкого применения, так как обычно легче интерпретировать образ в число а дальше использовать двоичную или аналоговую нейронную сеть, но при работе с малым количеством информации в виде образа, иногда легче использовать данный тип.

### Обучение с учителем.

Такое обучение предполагает наличие полного набора размеченных данных для обучения на всех этапах построения. Нейронная сеть заранее знает что должна получить на выходе, но не пользуется этим в процессе образования ответа, таким образом способна создать для себя более точный алгоритм. В основном такое обучение используется для классификации(например из массива фотографий с животными нейронная сеть сможет узнать и распределить каждого животного в свою группу) и регрессии(например обучаюсь по наборам значений y=f(x), сеть будет способна назвать значение y получив на вход x).

### Обучение без учителя.

При таком обучении у модели есть набор данных на вход и нет конкретных указаний, что должно получиться на выходе. Нейронная сеть извлекает из входных данных полезные признаки анализирует их. Таким образом сеть может делать кластеризацию(распределять данные на классы не знаю их количество и признаки), находить аномалии в данных(полезно для банков, при поиске мошенников), создавать ассоциации, создавать автоэнкодеры (это создание более «чистых» данных из входных, например такая сеть сможет убрать шум с видеоданных)

### Обучение с подкреплением.

Такое обучение похоже на компьютерные игры и игроков, игрок проходя уровень получает поощрение, в зависимости от ценности уровня, а заваливая уровень теряет очки. В данной ситуации в качестве игрока выступает нейронная сеть, возвращая правильные выходные данные в сложном примеры нейронная сеть получает большее награждение, и наоборот. Такое обучение показывает как похоже нейронные сети и человеческий мозг.

### Сети с фиксированными связями.

В таких нейронных сетях весовые коэффициенты выбираются сразу, исходя из условий задачи, это ускорит работу сетей, но при этом сделает её более узконаправленной.

### Сети с динамическими связями.

В таких нейронных сетях настройка синаптических весов происходит в процессе обучения. Это замедлит обучение, но при этом сеть сможет работать более широко.

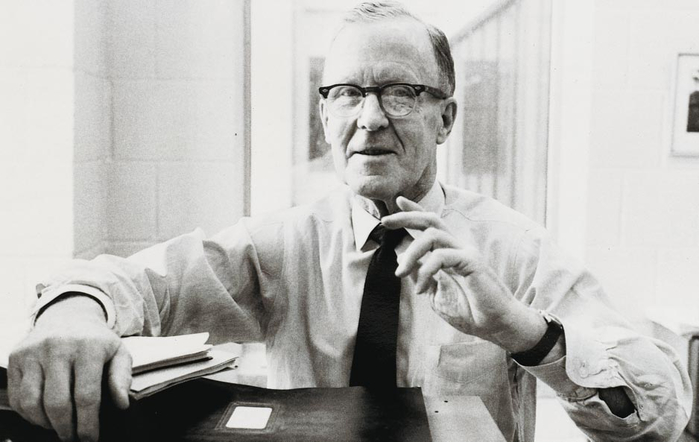
# История нейронных сетей.

Основные этапы в истории исследования и применения искусственных нейронных сетей:

• 1943 — У. Маккалок и У. Питтс формализуют понятие нейронной сети в фундаментальной статье о логическом исчислении идей и нервной активности.

• 1948 — Н. Винер вместе с соратниками публикует работу о кибернетике. Основной идеей является представление сложных биологических процессов математическими моделями.

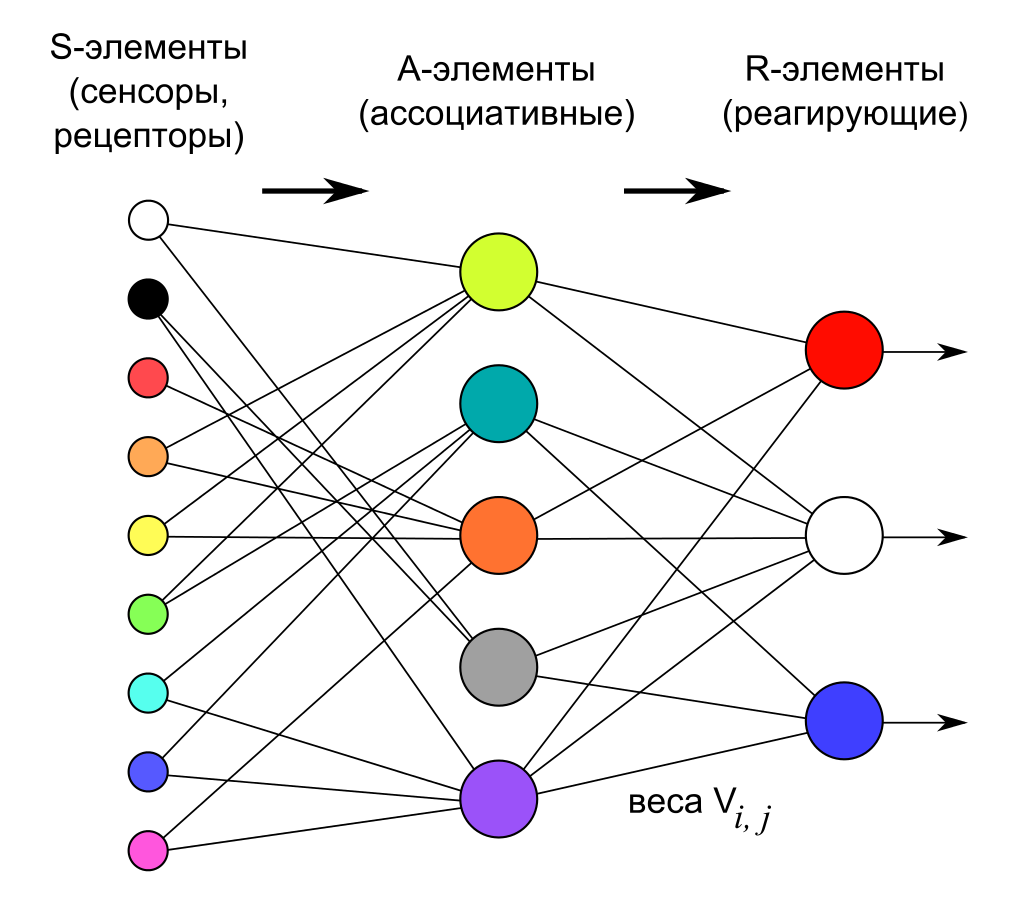
• 1949 — Д. Хебб предлагает первый алгоритм обучения.



• В 1958 Ф. Розенблатт изобретает однослойный перцептрон и демонстрирует его способность решать задачи классификации. Перцептрон обрёл популярность — его используют для распознавания образов, прогнозирования погоды и т. д.

Френк Розенблатт

Наглядный пример работы перцептрона:



• В 1960 году Уидроу совместно со своим студентом Хоффом на основе дельта-правила (формулы Уидроу) разработали Адалин, который сразу начал использоваться для задач предсказания и адаптивного управления. 11 Сейчас Адалин (адаптивный сумматор) является стандартным элементом многих систем обработки сигналов.

• В 1963 году в Институте проблем передачи информации АН СССР. А. П. Петровым проводится подробное исследование задач «трудных» для перцептрона.

• В 1969 году М. Минский публикует формальное доказательство ограниченности перцептрона и показывает, что он неспособен решать некоторые задачи (проблема «чётности» и «один в блоке»), связанные с инвариантностью представлений. Интерес к нейронным сетям резко спадает.

• В 1972 году Т. Кохонен и Дж. Андерсон независимо предлагают новый тип нейронных сетей, способных функционировать в качестве памяти.

• В 1973 году Б. В. Хакимов предлагает нелинейную модель с синапсами на основе сплайнов и внедряет её для решения задач в медицине, геологии, экологии.



• 1974 — Пол Дж. Вербос и А. И. Галушкин одновременно изобретают алгоритм обратного распространения ошибки для обучения многослойных перцептронов

• 1975 — Фукусима представляет когнитрон — самоорганизующуюся сеть, предназначенную для инвариантного распознавания образов, но это достигается только при помощи запоминания практически всех состояний образа.

• 1982 — после периода забвения, интерес к нейросетям вновь возрастает. Дж. Хопфилд показал, что нейронная сеть с обратными связями может представлять собой систему, минимизирующую энергию (так называемая сеть Хопфилда). Кохоненом представлены модели сети, обучающейся без учителя (нейронная сеть Кохонена), решающей задачи кластеризации, визуализации данных (самоорганизующаяся карта Кохонена) и другие задачи предварительного анализа данных. 12

• 1986 — Дэвидом И. Румельхартом, Дж. Е. Хинтоном и Рональдом Дж. Вильямсом и одновременно c С. И. Барцевым и В. А. Охониным (Красноярская группа) переоткрыт и существенно развит метод обратного распространения ошибки. Начался взрыв интереса к обучаемым нейронным сетям.

• 2007 Джеффри Хинтоном в университете Торонто созданы алгоритмы глубокого обучения многослойных нейронных сетей. Успех обусловлен тем, что Хинтон при обучении нижних слоев сети использовал ограниченную машину Больцмана (RBM — Restricted Boltzmann Machine).

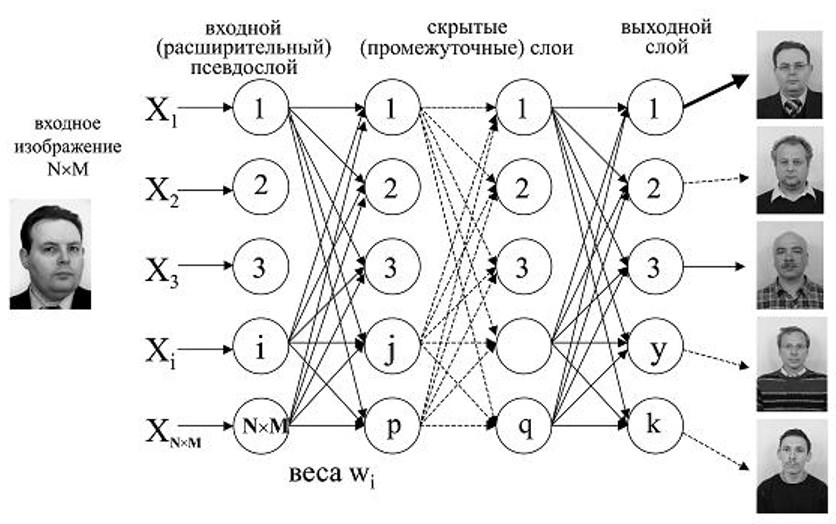
# Применение нейронных сетей в настоящее время.

## Область применения нейронных сетей весьма широка, они могут следующее:

* Распознавание образов и классификация
* Принятие решений и управление
* Кластеризация(разбиение на классы при отсутствии признаков классов)
* Прогнозирование
* Аппроксимация(замена элементов на более простые)
* Сжатие данных и ассоциативная память
* Анализ данных
* Оптимизация

Рассмотрим как применяются нейронные сети конкретнее.

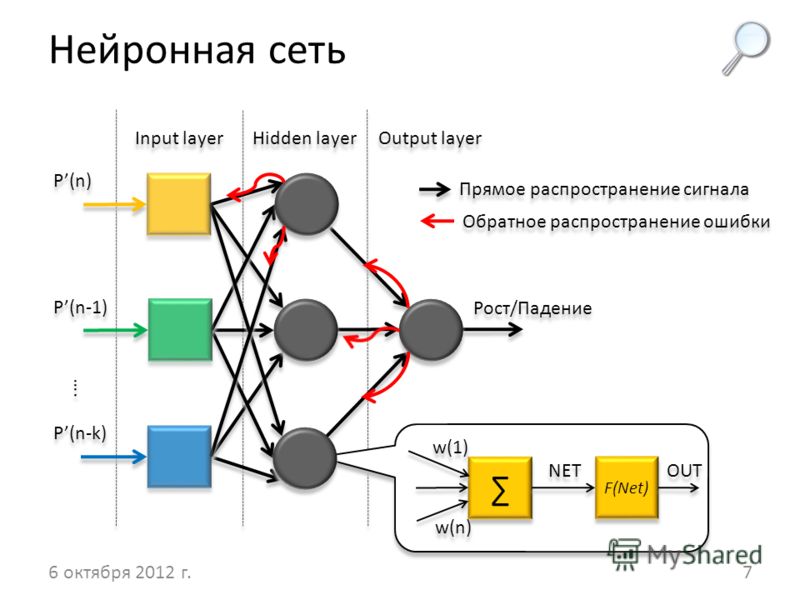
Распознавание образов и классификация происходят обычно с помощью нейронных сетей обученных с учителем и имеющие фиксированные связи. Это позволяет сети правильно обучиться разделять информацию, а так же не даст использовать новые классы и образы придуманные самой сетью. Так же следует выбрать тип сети в зависимости от входных данных, скорее всего лучше всего подойдёт образная нейронная сеть.



Принятие решений и управление. Для такой задачи лучше всего использовать нейронные сети обученные без учители и имеющие динамические связи. Благодаря чему сеть сможет рассмотреть большое количество вариантов для действия. Также лучшей для таких задач скорее всего будет двоичная нейронная сеть.

Кластеризация. Для этой задачи отлично подойдёт обучение без учителя, так как это даст возможность создание собственных классов, не ограничивая их количество, в то время как обучение с учителем ограничит выходные данные. Связи лучше всего использовать динамические, и от входных данных определить тип сети по этому параметру.

Прогнозирование. Эта ещё одна задача где человеку потребовалось бы широкое мышление, но в данном случае лучше ограничить выходные данные, готовая модель нейронной сети не должна предсказывать события, которых до этого никогда не было. Из этого следует что лучше всего подойдёт обучение с учителем и динамические связи.



Аппроксимация(замена элементов на более простые). На выход для решения данной задачи должен подаваться ограниченное количество данные(данные должны меняться в одном порядке). Из этого следует, что необходимо обучение с учителем, а так как алгоритмы схожи для переменных требуются статические связи.

Сжатие данных и ассоциативная память. На первый взгляд кажется, что очевидно в данном случае использовать обучение с учителем, но редко сжатие данных и ассоциативная память требуется при маленьких объёмах данных, а для больших объёмов получить «экспертную оценку» для огромного массива данных практически невозможно. Поэтому здесь выбор обучения зависит от объёма входных данных и возможности заранее показать сети, как нужно сжать данные.

Анализ данных. Данная задача может рассматриваться для уже известного вида данных а также способа их анализа, или же наоборот. Для первого случае очевидно использования учителя, а для второго случая без него. Связи лучше всего использовать динамические.

Оптимизация. Нейронной сети требуется строгое ограничение на выходные данные, так как после оптимизации продукт должен продолжать работать. Из этого следует, что обучение должно быть с учителем. В зависимости от того как вы хотите оптимизировать продукт, следует выбрать вид связи.

# Известные сервисы которые используют технологию нейронных сетей.

* FaceApp
* Deep Dream
* FaceHero
* http://What-Dog.net
* Algorithmia
* Murphy
* FastText
* http://How-Old.net
* Prisma
* The Emojini 3000
* Deep Photo Style Transfer
* Face Recognition
* Magneta
* Sonnet
* deeplearn.js
* Fast Style Transfer in TensorFlow
* Pysc2
* http://aiportraits.com
* http://thispersondoesnotexist.com
* #isitnude

## Давайте рассмотрим подробнее некоторые из них.

FaceApp. Мобильное приложение использующее нейронные сети для изменения лиц на фотографиях. Данная приложение является интересным примером обучения нейронной сети, тут использовали обучение с подкреплением. Разработчики специально опрашивали пользователей, как они воспринимают изменённое фото, как ещё одну фотографию или лишь картинку, и ответы загружали в сеть для того чтобы либо поощрить за хорошую работу, либо сообщить о том, что нужно было лучше. Нейронная сеть в FaceApp специльно анализирует фотографии и находит на них отличительные признаки, например улыбку или морщины, затем данные признаки изменяются опираясь на похожие примеры которые были до этого, чтобы после изменения фотографии, были заметны перемены в выражении лица и можно было с точностью сказать что это тот же человек.



Deep Dream. Данная нейронная сеть также изменяет фотографии, но изменяет не лица, а картину в целом, придавая ей психоделический вид, что обычно похоже на человеческий сон. Работа такой нейронной сети очень похожа на беспорядочную работу человеческого мозга. Она также улучшает паттерны изображения, может совмещать сразу несколько изображений. Вот наглядный пример её работы:



What-Dog. Эта нейронная сеть несёт уже менее развлекательный характер, хотя принцип её работы проще чем у выше стоящих нейронных сетей. Она позволяет определить к какой породе относится ваша собака, это обычная классификация. Скорее всего обучение такой нейронной сети происходило с учителем.

Algorithmia. Гораздо более серьёзное ПО по сравнению с выше стоящими. Данная программа основана на работе нейронных сетей и имеет огромное комерческое применение. Algorithmia, основанная в 2014 году Диего Оппенгеймером и Кенни Дэниелом, нацелена на внедрение моделей в производство с обеспечением безопасности и управления корпоративного уровня. Algorithmia сочетает в себе лучшие методы ИИ, MLOP и DevOps, создавая рабочие процессы машинного обучения и процессы для ИТ-специалистов, специалистов по обработке данных и других команд, которые организуют модели машинного обучения в производстве.



# Список литературы для более детального изучения нейронных сетей:

* Нейрокомпьютерная техника: Теория и практика — Ф. Уоссермен
* Нейронные сети — Саймон Хайкин
* Самоорганизующиеся карты — Тойво Кохонен
* Нейросетевые модели и алгоритмы. Справочник  — Дмитрий Тархов
* Make Your Own Neural Network — Tariq Rashid
* A Brief Introduction to Neural Networks — David Kriesel
* An Introduction to Statistical Learning — Gareth James
* Programming Collective Intelligence — Toby Segaran
* Neural Networks: A Systematic Introduction — Raul Rojas
* Deep Learning: Methods and Applications — Li Deng Dong Yu