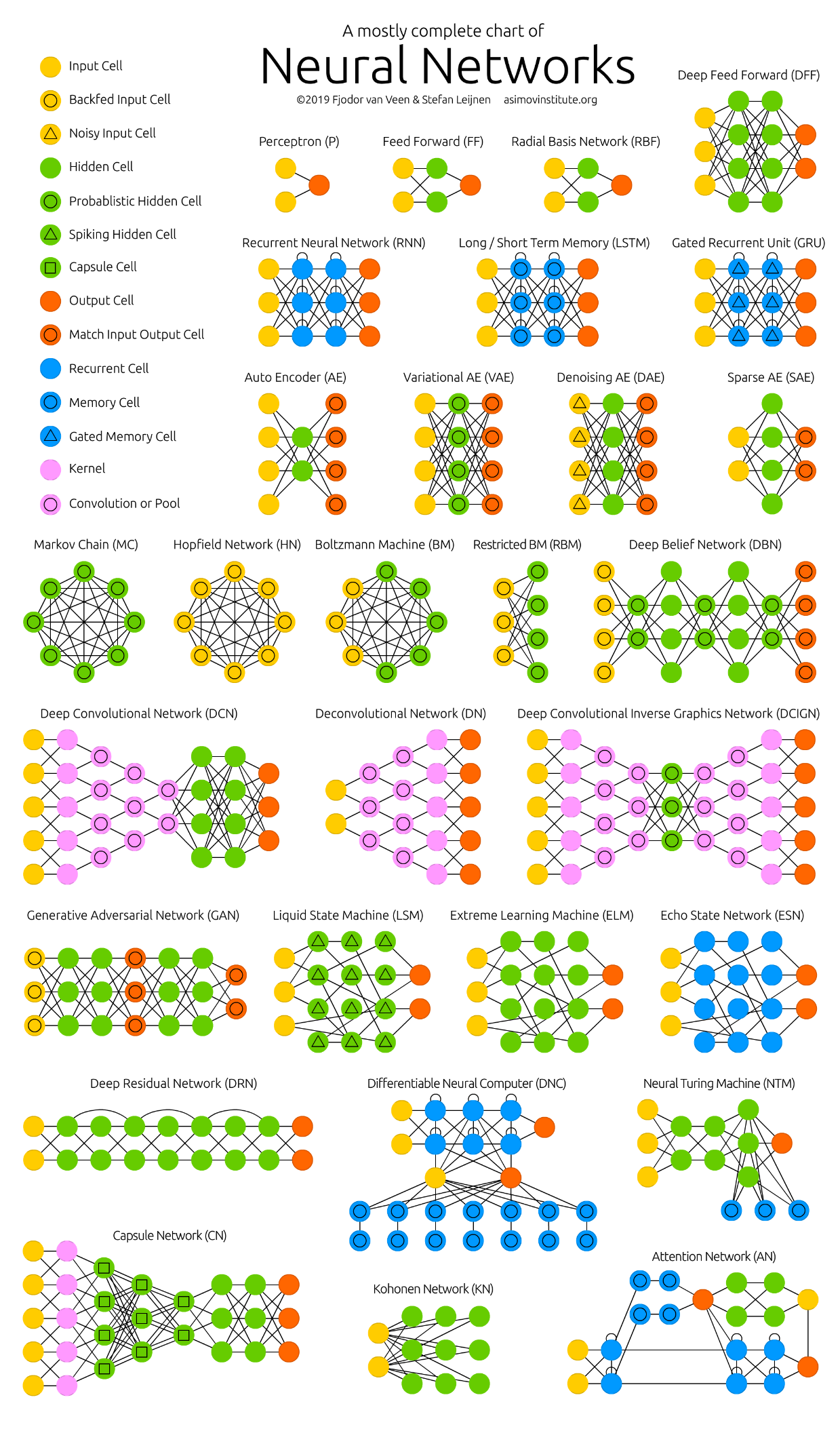
Архитектуры нейронных сетей



Искусственная нейронная сеть (ИНС) — вычислительная нелинейная модель, в основе которой лежит нейронная структура мозга, способная обучаться выполнению многих задач, таких как задач классификации, предсказания, принятия решений, визуализации и многих других. Все это достигается путем обучения на заданных примерах.

Архитектура нейронной сети определяет ее структуру и способ обработки информации, существует множество архитектур, каждая из которых имеет свои преимущества и недостатки. Выбор же между ними зависит от поставленной задачи.

**Нейронные сети прямого распространения (FF или FFNN) и перцептроны (P)** очень прямолинейны, они подают информацию с начала в конец (ввод и вывод соответственно). Нейронные сети часто описываются как имеющие слои, где каждый слой состоит из входных, скрытых или выходных ячеек, как правило, два соседних слоя полностью связаны. Самая простая и практичная сеть имеет две входные ячейки и одну выходную ячейку, которую можно использовать для моделирования логических элементов. Обычно FFNN обучаются посредством обратного распространения ошибки, предоставляя сети парные наборы данных о том, «что входит» и «что мы хотим иметь на выходе». Ошибка обратного распространения часто представляет собой некоторую вариацию разницы между входными и выходными данными. Учитывая, что в сети достаточно скрытых нейронов, она теоретически всегда может моделировать взаимосвязь между входом и выходом. На практике их использование гораздо более ограничено, но их обычно объединяют с другими сетями для формирования новых сетей.

**Рекуррентные нейронные сети (RNN)** являются FFNN с временным поворотом: у них есть связи между проходами, связи во времени. Нейроны получают информацию не только из предыдущего слоя, но и от себя из предыдущего прохода. Это означает, что порядок, в котором вы передаете входные данные и обучаете сеть, имеет значение. Одной из больших проблем с RNN является проблема исчезновения градиента, когда, в зависимости от используемых функций активации, информация быстро теряется с течением времени точно так же, как очень глубокие FFNN теряют информацию по глубине. В принципе, RNN можно использовать во многих областях, поскольку большинство форм данных не имеют временной шкалы и могут быть представлены в виде последовательности. Изображение или строка текста могут передаваться по одному пикселю или символу за раз, поэтому зависящие от времени веса используются для того, что было раньше в последовательности. В общем, рекуррентные сети являются хорошим выбором для дополнения информации, например автодополнения.

**Долговременная/краткосрочная память (LSTM)** сети пытаются бороться с проблемой исчезновения/взрыва градиента, вводя вентили и явно определенную ячейку памяти. Каждый нейрон имеет ячейку памяти и трое ворот: вход, выход и забывание. Функция этих ворот — защитить информацию, останавливая или разрешая ее поток. Входной вентиль определяет, какая часть информации из предыдущего слоя сохраняется в ячейке. Выходной определяет, какая часть следующего уровня узнает о состоянии этой ячейки. Ворота забывания на первый взгляд кажутся странным включением, но иногда полезно забыть: если он изучает книгу и начинается новая глава, сети может потребоваться забыть некоторых персонажей из предыдущей главы. Было показано, что LSTM способны изучать сложные последовательности, например, писать, как Шекспир, или сочинять примитивную музыку. Обратите внимание, что каждый из этих ворот имеет вес по отношению к ячейке предыдущего нейрона, поэтому для их работы обычно требуется больше ресурсов.

**Закрытые рекуррентные единицы (ГРУ)** являются небольшой вариацией LSTM. У них на один вентиль меньше, и они немного по-другому подключены: вместо входа, выхода и вентиля забывания у них есть вентиль обновления. Этот шлюз обновления определяет, сколько информации следует сохранить из последнего состояния и сколько информации следует впустить из предыдущего уровня. Ворота сброса функционируют так же, как ворота забывания LSTM, но расположены немного по-другому. Они всегда передают свое полное состояние, у них нет выходного вентиля. В большинстве случаев они функционируют очень похоже на LSTM, с самым большим отличием в том, что GRU немного быстрее и проще в использовании (но также немного менее выразительны).

**Автоэнкодеры (АЕ)** несколько похожи на FFNN, поскольку AE больше похожи на другое использование FFNN, чем на принципиально другую архитектуру. Основная идея автокодировщиков заключается в автоматическом кодировании информации. Вся сеть всегда напоминает форму песочных часов с меньшими скрытыми слоями, чем входной и выходной слои. АЭ также всегда симметричны относительно среднего слоя. Самый маленький уровень почти всегда находится посередине, в месте, где информация наиболее сжата (узкое место в сети). Все до середины называется частью кодирования, все после середины — декодированием, а середина — кодом. Их можно обучить с помощью обратного распространения ошибки, подавая входные данные и устанавливая ошибку как разницу между входными данными и тем, что получилось.

**Разреженные автоэнкодеры (SAE)** в некотором смысле являются противоположностью AE. Вместо того, чтобы учить сеть представлять кучу информации в меньшем «пространстве» или узлах, мы пытаемся кодировать информацию в большем пространстве. Таким образом, вместо того, чтобы сеть сходилась в середине, а затем снова расширялась до входного размера, мы расширяем середину. Эти типы сетей можно использовать для извлечения множества мелких объектов из набора данных. Если бы SAE обучался так же, как и AE, вы почти во всех случаях получили бы довольно бесполезную идентификационную сеть. Чтобы предотвратить это, вместо обратной передачи входных данных мы возвращаем входные данные плюс драйвер разреженности. Этот драйвер разреженности может принимать форму порогового фильтра, в котором обратно передается и обучается только определенная ошибка, другая ошибка будет «нерелевантной» для этого прохода и будет равна нулю.

**Цепи Маркова (MC или цепь Маркова с дискретным временем, DTMC)** являются своего рода предшественниками BM и HN. Их можно понять так: каковы шансы, что из этого узла, где я сейчас нахожусь, я попаду в любой из соседних узлов? Они не имеют памяти, что означает, что каждое состояние, в котором вы находитесь, полностью зависит от предыдущего состояния. Хотя на самом деле они не являются нейронными сетями, они напоминают нейронные сети и составляют теоретическую основу для BM и HN.

Сеть **Хопфилда (HN)** это сеть, в которой каждый нейрон связан с каждым другим нейроном. Каждый узел вводится перед обучением, затем скрывается во время обучения и выводится впоследствии. Сети обучаются путем установки значений нейронов в соответствии с желаемым шаблоном, после чего можно вычислить веса. После этого вес не меняется. После обучения одному или нескольким шаблонам сеть всегда будет сходиться к одному из изученных шаблонов, поскольку сеть стабильна только в этих состояниях. Эти сети часто называют ассоциативной памятью, поскольку они сходятся к состоянию, наиболее похожему на входные данные; если люди видят половину таблицы, мы можем отобразить другую половину, эта сеть сходится к таблице, если ей представить половину шума и половину таблицы.

**Машины Больцмана (БМ)** во многом похожи на HN, но: некоторые нейроны помечены как входные, а другие остаются «скрытыми». Входные нейроны становятся выходными нейронами в конце полного обновления сети. Он начинается со случайных весов и обучается посредством обратного распространения ошибки или, в последнее время, посредством контрастного расхождения. Процесс обучения и работы BM во многом аналогичен HN: входные нейроны настраиваются на определенные фиксированные значения, после чего сеть освобождается. Пока они свободны, клетки могут получить любое значение, и мы постоянно перемещаемся между входными и скрытыми нейронами.

**Ограниченные машины Больцмана (RBM)** удивительно похожи на БМ и, следовательно, также похожи на HN. Самая большая разница между BM и RBM заключается в том, что RBM более удобны в использовании, поскольку они более ограничены. Они не соединяют триггерно каждый нейрон с каждым другим нейроном, а лишь соединяют каждую отдельную группу нейронов с каждой другой группой. RBM можно обучать, как FFNN, с одной особенностью: вместо передачи данных вперед, а затем обратного распространения, вы передаете данные вперед, а затем передаете данные назад. После этого вы тренируетесь с прямым и обратным распространением.

**Сети глубокого доверия (DBN)** — это название многоуровневых архитектур, в основном состоящих из RBM или VAE. Было показано, что эти сети эффективно обучаются стек за стеком, где каждому AE или RBM нужно только научиться кодировать предыдущую сеть. DBN можно обучить посредством контрастного расхождения или обратного распространения ошибки и научиться представлять данные в виде вероятностной модели, как обычные RBM или VAE. После обучения или приведения к (более) стабильному состоянию посредством обучения без учителя модель можно использовать для генерации новых данных. При обучении с использованием контрастной дивергенции он может даже классифицировать существующие данные, поскольку нейроны обучены искать различные функции.

**Сверточные нейронные сети (CNN или глубокие сверточные нейронные сети, DCNN)** в основном используются для обработки изображений, но также могут использоваться для других типов ввода, например аудио. Типичный вариант использования CNN — это когда вы передаете сетевые изображения, а сеть классифицирует данные, например, она выводит «кошка», если вы даете ей изображение кошки, и «собака», когда вы даете ей изображение собаки. CNN, как правило, начинаются со входного «сканера», который не предназначен для одновременного анализа всех обучающих данных. Например, чтобы ввести изображение размером 200 x 200 пикселей, вам не понадобится слой с 40 000 узлов. Вместо этого вы создаете входной слой размером, скажем, 20 x 20, в который подаются первые 20 x 20 пикселей изображения. После того, как вы передали этот ввод, вы перемещаете сканер на один пиксель вправо. Обратите внимание, что нельзя перемещать входные данные на 20 пикселей, вы не разбиваете изображение на блоки 20 x 20, а скорее ползете по нему. Эти входные данные затем передаются через сверточные слои вместо обычных слоев, где не все узлы подключены ко всем узлам. Каждый узел касается только близких соседних ячеек. Эти сверточные слои также имеют тенденцию сжиматься по мере того, как они становятся глубже, в основном за счет легко делимых коэффициентов входных данных. Помимо этих сверточных слоев, они также часто содержат слои объединения. Объединение в пул — это способ отфильтровать детали: часто встречающийся метод объединения — это максимальное объединение, при котором мы берем, скажем, 2 x 2 пикселя и передаем пиксель с наибольшим количеством красного цвета. Реализации CNN в реальном мире часто приклеивают FFNN к концу для дальнейшей обработки данных, что позволяет создавать сильно нелинейные абстракции.

**Деконволюционные сети (DN)** представляют собой перевернутые сверточные нейронные сети. Представьте себе, что вы вводите в сеть слово «кошка» и обучаете ее создавать кошачьи изображения, сравнивая то, что она генерирует, с реальными изображениями кошек. DNN можно комбинировать с FFNN точно так же, как обычные CNN.

**Сети глубокой сверточной обратной графики (DCIGN)** имеют несколько вводящее в заблуждение название, поскольку на самом деле они представляют собой VAE, но с CNN и DNN для соответствующих кодеров и декодеров. Эти сети пытаются моделировать «особенности» кодирования как вероятности, чтобы научиться создавать изображение с кошкой и собакой вместе, когда-либо видя только одну из двух на отдельных фотографиях. Точно так же вы можете скормить ему фотографию кошки с надоедливой собакой ваших соседей и попросить убрать собаку, даже не выполняя такой операции.

**Генеративно-состязательные сети (GAN)** они из разных сетей, они близнецы: две сети, работающие вместе. GAN состоят из любых двух сетей (хотя часто это комбинация FF и CNN), одна из которых занимается созданием контента, а другая — его оценкой. Оценивающая сеть получает либо обучающие данные, либо сгенерированный контент из генеративной сети. Насколько хорошо оценивающая сеть смогла правильно предсказать источник данных, затем используется как часть ошибки для генерирующей сети.

**Машины экстремального обучения (ELM)** по сути представляют собой FFNN, но со случайными связями. Они очень похожи на LSM и ESN, но не являются повторяющимися или пиковыми. Они также не используют обратное распространение ошибки. Вместо этого они начинают со случайных весов и обучают веса за один шаг в соответствии с методом наименьших квадратов (наименьшая ошибка среди всех функций). В результате получается гораздо менее выразительная сеть, но она также намного быстрее, чем обратное распространение ошибки.

**Сети состояний эха (ESN)** — это еще один тип рекуррентной сети. Этот отличается от других тем, что имеет случайные связи между нейронами (т. е. не организован в аккуратные наборы слоев), и они обучаются по-разному. Вместо подачи входных данных и обратного распространения ошибки мы подаем входные данные, пересылаем их и некоторое время обновляем нейроны, а затем наблюдаем за выходными данными с течением времени.

**Глубокие остаточные сети (DRN)** представляют собой очень глубокие FFNN с дополнительными соединениями, передающими входные данные с одного уровня на более поздний уровень (часто от 2 до 5 уровней). Вместо того, чтобы пытаться найти решение для сопоставления некоторых входных данных с некоторыми выходными данными, скажем, на 5 уровнях, сеть вынуждена научиться сопоставлять некоторые входные данные с некоторыми выходными + некоторыми входными данными. По сути, он добавляет идентичность решению, перенося старые входные данные и передавая их на более поздний уровень.

**Нейронные машины Тьюринга (NTM)** можно понимать как абстракцию LSTM и попытку раскрыть нейронные сети из «черного ящика» (и дать нам некоторое представление о том, что там происходит). Вместо кодирования ячейки памяти непосредственно в нейрон, память отделяется. Это попытка объединить эффективность и надежность обычного цифрового хранилища с эффективностью и выразительной силой нейронных сетей. Идея состоит в том, чтобы иметь банк памяти, адресуемый по содержимому, и нейронную сеть, которая может читать и записывать из него. «Тьюринг» в нейронных машинах Тьюринга происходит от того, что они являются полными по Тьюрингу: способность читать, писать и изменять состояние на основе того, что она читает, означает, что она может представлять все, что может представить универсальная машина Тьюринга.

**Дифференцируемые нейронные компьютеры (DNC)** — это усовершенствованные нейронные машины Тьюринга с масштабируемой памятью, вдохновленные тем, как воспоминания хранятся в гиппокампе человека. Идея состоит в том, чтобы взять классическую компьютерную архитектуру фон Неймана и заменить ЦП на RNN, которая учится, когда и что читать из ОЗУ.

**Капсульные сети (CapsNet)** — это биологическая альтернатива объединению, где нейроны связаны с несколькими весами (вектором), а не только с одним весом (скаляром). Это позволяет нейронам передавать больше информации, чем просто то, какой признак был обнаружен, например, где объект находится на изображении или какой у него цвет и ориентация.

**Сети Кохонена (KN, также самоорганизующаяся (функциональная) карта, SOM, SOFM)** используют конкурентное обучение для классификации данных без присмотра. Входные данные передаются в сеть, после чего сеть оценивает, какой из ее нейронов наиболее точно соответствует этому входному сигналу. Затем эти нейроны настраиваются так, чтобы еще лучше соответствовать входным данным, увлекая за собой своих соседей.

**Сети внимания (AN)** можно рассматривать как класс сетей, включающий в себя архитектуру Трансформатор. Они используют механизм внимания для борьбы с распадом информации, отдельно сохраняя предыдущие состояния сети и переключая внимание между состояниями. Скрытые состояния каждой итерации в слоях кодирования сохраняются в ячейках памяти. Уровни декодирования связаны со слоями кодирования, но он также получает данные из ячеек памяти, отфильтрованные контекстом внимания. Этот этап фильтрации добавляет контекст для слоев декодирования, подчеркивая важность определенных функций. Сеть внимания, создающая этот контекст, обучается с использованием сигнала ошибки с выхода уровня декодирования. Более того, контекст внимания можно визуализировать, давая ценную информацию о том, какие входные функции соответствуют каким выходным функциям.

Список сточников литературы:

1. *Розенблатт, Фрэнк. «Персептрон: вероятностная модель хранения и организации информации в мозге». Психологический обзор 65.6 (1958): 386*
2. *Элман, Джеффри Л. «Обнаружение структуры во времени». Когнитивная наука 14.2 (1990): 179-211.*
3. *Хохрайтер, Зепп и Юрген Шмидхубер. «Долгая кратковременная память». Нейронные вычисления 9.8 (1997): 1735–1780.*
4. *Чунг, Джунён и др. «Эмпирическая оценка закрытых рекуррентных нейронных сетей при моделировании последовательностей». Препринт arXiv arXiv:1412.3555 (2014)*
5. *Бурлар, Эрве и Ив Камп. «Автоассоциация с помощью многослойных перцептронов и разложение по сингулярным значениям». Биологическая кибернетика 59.4-5 (1988): 291-294.*
6. *Марк'Аурелио Ранзато, Кристофер Поултни, Сумит Чопра и Янн ЛеКун. «Эффективное изучение разреженных представлений с помощью модели, основанной на энергии». Труды НИПС. 2007.*
7. *Хейс, Брайан. «Первые звенья цепи Маркова». American Scientist 101.2 (2013): 252.*
8. *Хопфилд, Джон Дж. «Нейронные сети и физические системы с появляющимися коллективными вычислительными способностями». Труды Национальной академии наук 79.8 (1982): 2554-2558.*
9. *Хинтон, Джеффри Э. и Терренс Дж. Сейновски. «Обучение и переучивание на машинах Больцмана». Параллельная распределенная обработка: Исследования микроструктуры познания 1 (1986): 282-317.*
10. *Смоленский Павел. Обработка информации в динамических системах: Основы теории гармонии. № КУ-КС-321-86. УНИВЕРСИТЕТ КОЛОРАДО НА ОТДЕЛЕНИИ КОМПЬЮТЕРНЫХ НАУК БОУЛДЕРА, 1986.*
11. *Бенджио, Йошуа и др. «Жадное послойное обучение глубоких сетей». Достижения в области нейронных систем обработки информации 19 (2007): 153.*
12. *ЛеКун, Янн и др. «Градиентное обучение, применяемое для распознавания документов». Труды IEEE 86.11 (1998): 2278-2324.*
13. *Зейлер, Мэтью Д. и др. «Деконволюционные сети». Компьютерное зрение и распознавание образов (CVPR), конференция IEEE 2010 г. IEEE, 2010.*
14. *Кулкарни, Теджас Д. и др. «Глубокая сверточная обратная графическая сеть». Достижения в области нейронных систем обработки информации. 2015.*
15. *Гудфеллоу, Ян и др. «Генераторно-состязательные сети». Достижения в области нейронных систем обработки информации (2014).*
16. *Хуанг, Гуан-Бин и др. «Машина экстремального обучения: теория и приложения». Нейрокомпьютинг 70.1-3 (2006): 489-501.*
17. *Йегер, Герберт и Харальд Хаас. «Использование нелинейности: прогнозирование хаотических систем и экономия энергии в беспроводной связи». наука 304.5667 (2004): 78-80.*
18. *Он, Кайминг и др. «Глубокое остаточное обучение для распознавания изображений». Препринт arXiv arXiv:1512.03385 (2015).*
19. *Грейвс, Алекс, Грег Уэйн и Иво Данихелька. «Нейронные машины Тьюринга». Препринт arXiv arXiv:1410.5401 (2014).*
20. *Грейвс, Алекс и др. «Гибридные вычисления с использованием нейронной сети с динамической внешней памятью». Природа 538 (2016): 471-476.*
21. *Сабур, Сара, Фрост, Николас и Хинтон, GE «Динамическая маршрутизация между капсулами». В «Достижениях в области нейронных систем обработки информации» (2017): 3856-3866.*
22. *Кохонен, Теуво. «Самоорганизованное формирование топологически правильных карт объектов». Биологическая кибернетика 43.1 (1982): 59-69.*
23. *Ядерберг, Макс и др. «Пространственные трансформаторные сети». В «Достижениях в области нейронных систем обработки информации» (2015): 2017–2025 гг.*