Методы машинного обучения

Лекция 14

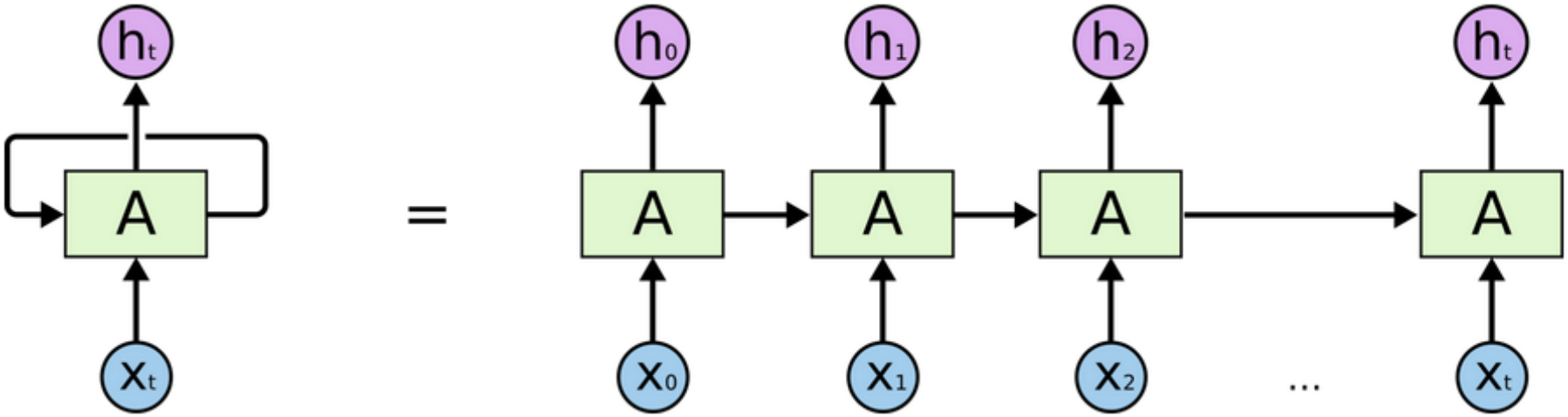
Рекуррентные нейронные сети

или как правильно кусать себя за хвост

Рекуррентные нейронные сети

Рекуррентые нейронные сети (Recurrent Neural Networks, RNN) - это сети, содержащие обратные связи, позволяющие сохранять информацию (скрытое состояние) от предыдущих входных данных. Хорошо подходят для обработки последовательностей, например, временные ряды (изменения цен акций, показания датчиков), последовательности с зависимыми элементами (предложения естественного языка), т.е. любые данные, где соседние экземпляры (точки выборки) зависят друг от друга и эту зависимость нельзя игнорировать.

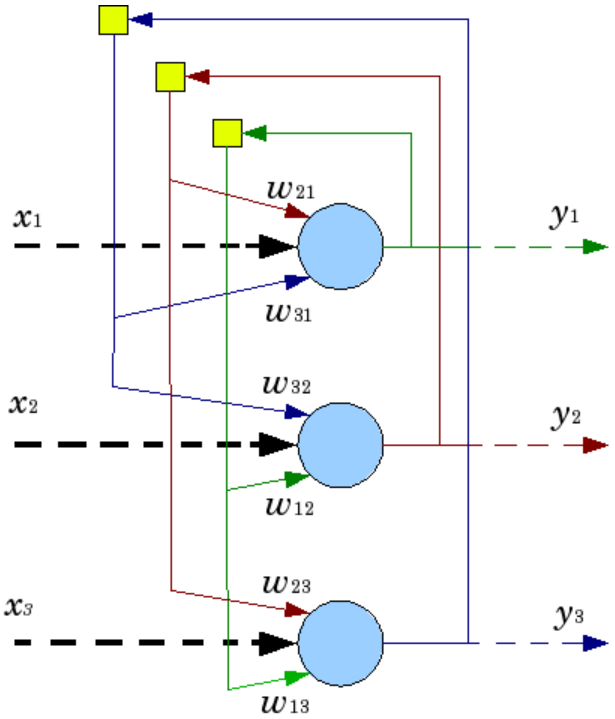
Фрагмент нейронной сети A принимает входное значение x_t и возвращает значение h_t. Наличие обратной связи позволяет передавать информацию от одного шага сети к другому. Рекуррентную сеть можно рассматривать, как несколько копий одной и той же сети, каждая из которых передает информацию последующей копии.



Нейронная сеть Хопфилда

Нейронная сеть Хопфилда - полносвязная однослойная нейронная сеть с симметричной матрицей связей. Функционирование до достижения равновесия, т.е. когда следующее состояние сети в точности равно предыдущему: начальное состояние является входным образом, а при равновесии получают выходной образ. N искусственных нейронов. Каждый нейрон системы может принимать на входе и на выходе одно из двух состояний (что аналогично выходу нейрона с пороговой функцией активации):

$$y_i(t) \in \{-1; +1\}$$



Каждый нейрон связан со всеми остальными нейронами.

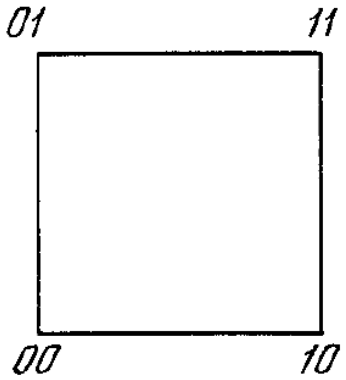
Взаимодействие нейронов сети описывается выражением:

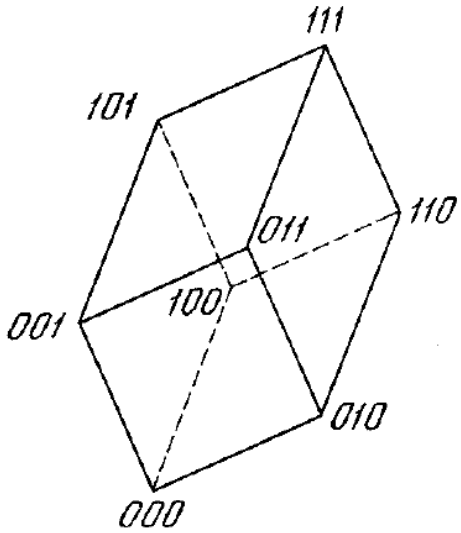
$$E = \frac{1}{2} \sum_{i,j=1}^{N} w_{ij} x_i x_j$$

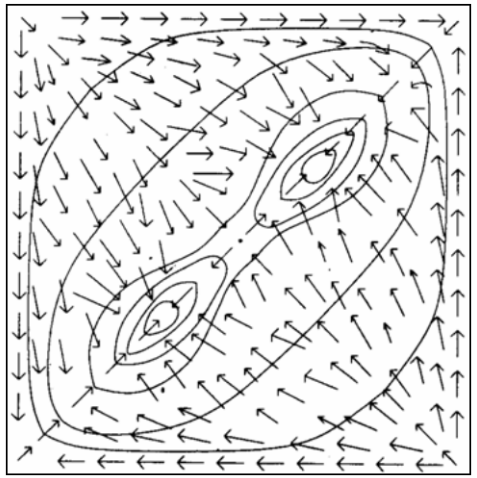
где w_{ij} — элемент матрицы взаимодействий W, которая состоит из весовых коэффициентов связей между нейронами.

Нейронная сеть Хопфилда - Состояние сети

Состояние сети – это просто множество текущих значений сигналов y_i от всех нейронов. В первоначальной сети Хопфилда состояние каждого нейрона менялось в дискретные случайные моменты времени, в последующей работе состояния нейронов могли меняться одновременно. Так как выходом бинарного нейрона может быть только два значения (промежуточных уровней нет), то текущее состояние сети является двоичным числом, каждый бит которого является сигналом y_i некоторого нейрона.







Нейронная сеть Хопфилда - Обучение

В процессе обучения формируется выходная матрица W, которая запоминает m эталонных «образов» — *N*-мерных бинарных векторов: $X_m = (x_{m1}, x_{m2}, \dots, x_{mN})$, эти образы во время эксплуатации сети будут выражать отклик системы на входные сигналы, или иначе - окончательные значения выходов y_i после серии итераций.

Вычисление коэффициентов основано на следующем правиле: для всех запомненных образов X_i матрица связи должна удовлетворять уравнению :

$$X_i = WX_i$$

Расчёт весовых коэффициентов проводится по следующей формуле:

$$w_{ij} = \frac{1}{N} \sum_{d=1..m} X_{id} X_{jd}$$

где N— размерность векторов, m— число запоминаемых выходных векторов, d — номер запоминаемого выходного вектора, X_{ij} — i-я компонента запоминаемого выходного j-го вектора.

Может быть записано в векторном виде :

$$W = \frac{1}{N} \sum_i X_i X_i^T$$

где X_i— i-й запоминаемый вектор-столбец.

В работе Кохонена и Гроссберга (доказана теорема) показано, что сеть с обратными связями является устойчивой, если её матрица симметрична и имеет нули на главной диагонали.

Нейронная сеть Хопфилда – Применение обученной сети

Обученная сеть способна распознавать входные сигналы - то есть, определять, к какому из запомненных образцов они относятся. Сеть последовательно меняет свои состояния согласно формуле:

$$X(t+1) = F(W \cdot X(t))$$

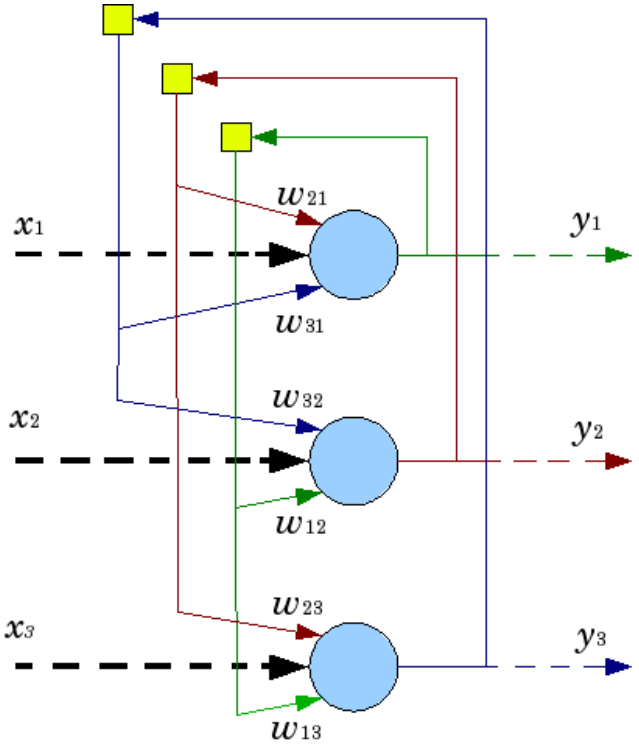
где F — активационная функция, $X(t)$ и $X(t+1)$— текущее и следующее состояния сети.

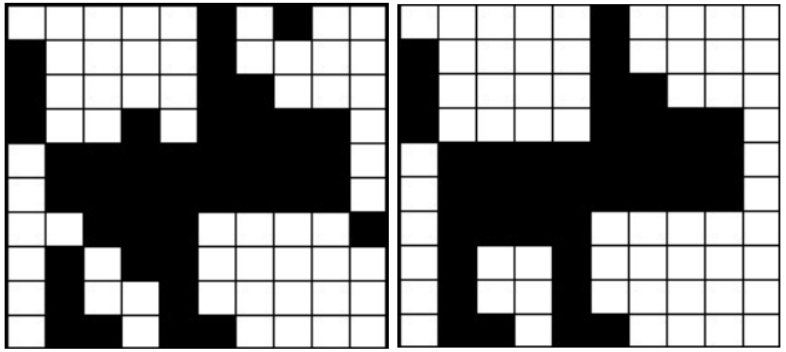
Локальным полем a_i, действующим на нейрон x_i со стороны всех остальных нейронов сети, является значение:

$$a_i(t) = \sum_{j=1, j\neq i}^{N} w_{ji} x_j(t-1)$$

Значение выхода нейрона i в текущий момент времени $x_i(t)$ рассчитывается по формуле:

$$x_i(t) = sign\left(\sum_{j=1, j\neq i}^{N} w_{ji} x_j(t-1)\right)$$





Искажённый образ Эталон

Нейронная сеть Хопфилда – Режимы работы

Две модификации (режима работы), отличающиеся по времени передачи сигнала: **Синхронный** и **Асинхронный**.

Синхронный режим работы сети - последовательно просматриваются нейроны, их состояния запоминаются отдельно и не меняются до тех пор, пока не будут пройдены все нейроны сети. Когда все нейроны просмотрены, их состояния одновременно (то есть синхронно) меняются на новые.

Асинхронный режим работы сети – состояния нейронов в следующий момент времени меняются последовательно: вычисляется локальное поле для первого нейрона в момент t, определяется его реакция, и нейрон устанавливается в новое состояние (которое соответствует его выходу в момент $t+1$), и так далее — состояние каждого следующего нейрона вычисляется с учетом всех изменений состояний рассмотренных ранее нейронов.

В асинхронном режиме невозможен динамический аттрактор: вне зависимости от количества запомненных образов и начального состояния сеть непременно придёт к устойчивому состоянию (статическому аттрактору).

Нейронная сеть Хопфилда – Ограничения сети

1. Относительно небольшой объём памяти, величину которого можно оценить выражением:

$$M \approx \frac{N}{2 \cdot log_2 N}$$

Попытка записи большего числа образов приводит к тому, что нейронная сеть перестаёт их распознавать.

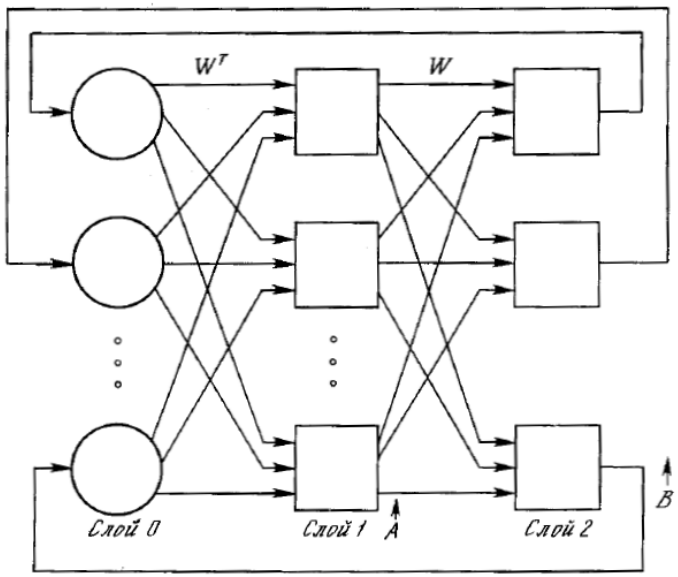
2. В синхронном режиме сеть может прийти к динамическому аттрактору.

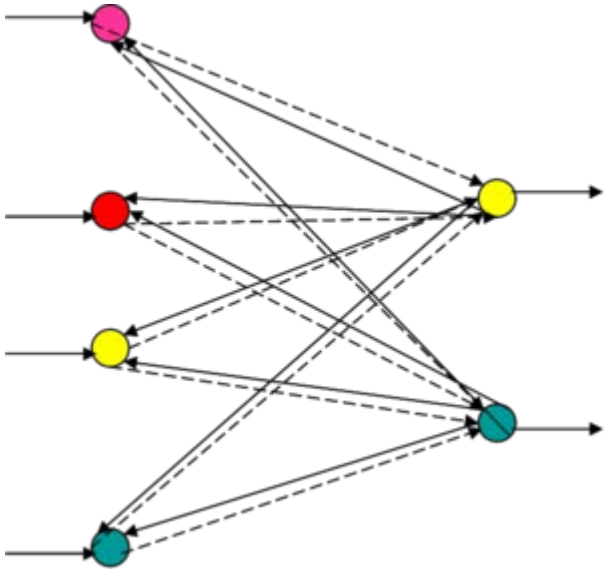
3. Достижение устойчивого состояния не гарантирует правильный ответ сети. Это происходит из-за того, что сеть может сойтись к так называемым ложным аттракторам, иногда называемым «химерами» (как правило, химеры склеены из фрагментов различных образов возникают при слишком большом количестве запомненных образов).

Двунаправленная ассоциативная память

Двунаправленная ассоциативная память (ДАП) является гетероассоциативной; входной вектор поступает на один набор нейронов, а соответствующий выходной вектор вырабатывается на другом наборе нейронов.

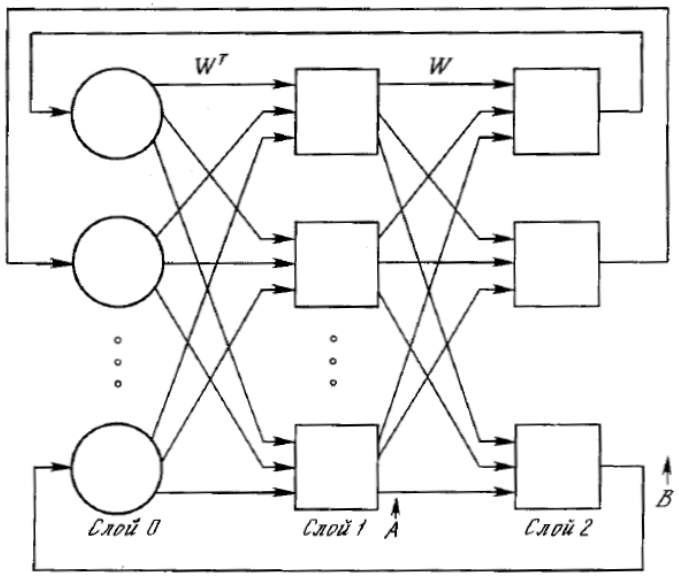
На рисунке приведены две базовые конфигурации ДАП. Одна из них выбрана таким образом, чтобы подчеркнуть сходство с сетями Хопфилда и предусмотреть увеличения количества слоев.





ДАП – работа сети

Входной вектор *A* обрабатывается матрицей весов *W* сети, в результате чего вырабатывается вектор выходных сигналов нейронов *B*. Вектор B затем обрабатывается транспонированной матрицей W^T весов сети, которая вырабатывает новые выходные сигналы, представляющие собой новый входной вектор *A*. Этот процесс повторяется до тех пор, пока сеть не достигнет стабильного состояния, в котором ни вектор *A*, ни вектор *B* не изменяются.



В векторной форме:

$$B = F(AW)$$

где B – вектор выходного сигнала нейронов слоя 2, A – вектор выходных сигналов нейрона слоя 1, W – матрица весов связей между слоями 1 и 2, F- функция активации.

$$A = F(BW^T)$$

где W^T – является транспонированной матрицей W.

Сеть функционирует в направлении минимизации функции энергии Ляпунова в основном таким же образом, как и сети Хопфилда в процессе сходимости. Таким образом, каждый цикл модифицирует систему в направлении энергетического минимума, расположение которого определяется значениями весов.

ДАП – Кодирование ассоциаций и Емкость памяти

Обучение производится с использованием обучающего набора, состоящего из пар векторов A и B. Процесс обучения реализуется в форме вычислений; это означает, что весовая матрица вычисляется как сумма произведении всех векторных пар обучающего набора. В символьной форме:

$$W = \sum_i A_i^T B_i$$

Существует взаимосвязь между ДАП и рассмотренными сетями Хопфилда. Если весовая матрица W является квадратной и симметричной, то $W = W^T$**.** В этом случае, если слои 1 и 2 являются одним и тем же набором нейронов, ДАП превращается в автоассоциативную сеть Хопфилда.

Емкость сети. Если M векторов выбраны случайно и представлены в указанной выше форме, и если M меньше чем:

$$M_1 \approx \frac{N}{2 \cdot log_2 N}$$

где N – количество нейронов в наименьшем слое, тогда все запомненные образы, за исключением «малой части», могут быть восстановлены.

Если все образы должны восстанавливаться, *M* должно быть меньше

$$M_2 \approx \frac{N}{4 \cdot log_2 N}$$

Например, если *N* = 1024, тогда M_1 должно быть меньше 51, а M_2 меньше 25.

Обработка последовательностей

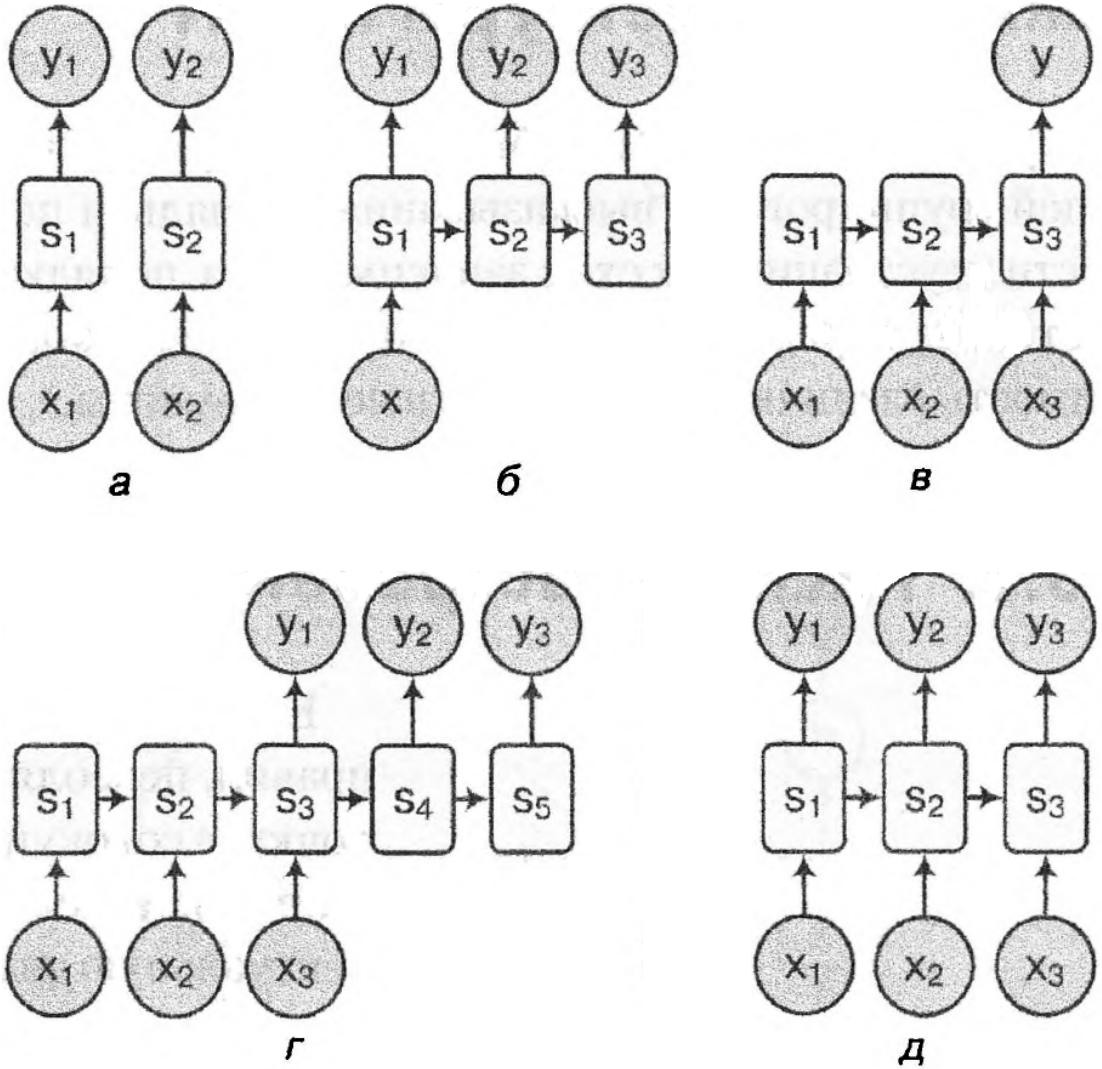
а) Один вход, один выход (**one-to-one**);

б) Один вход, последовательность выходов (**one-to-many***);*

в) *П*оследовательность входов, один выход (**many-to-one***);*

г) Последовательность входов, затем последовательность выходов (**many-tomany**);

д) Синхронизированные последовательности входов и выходов (**synchronized many-to-many**).



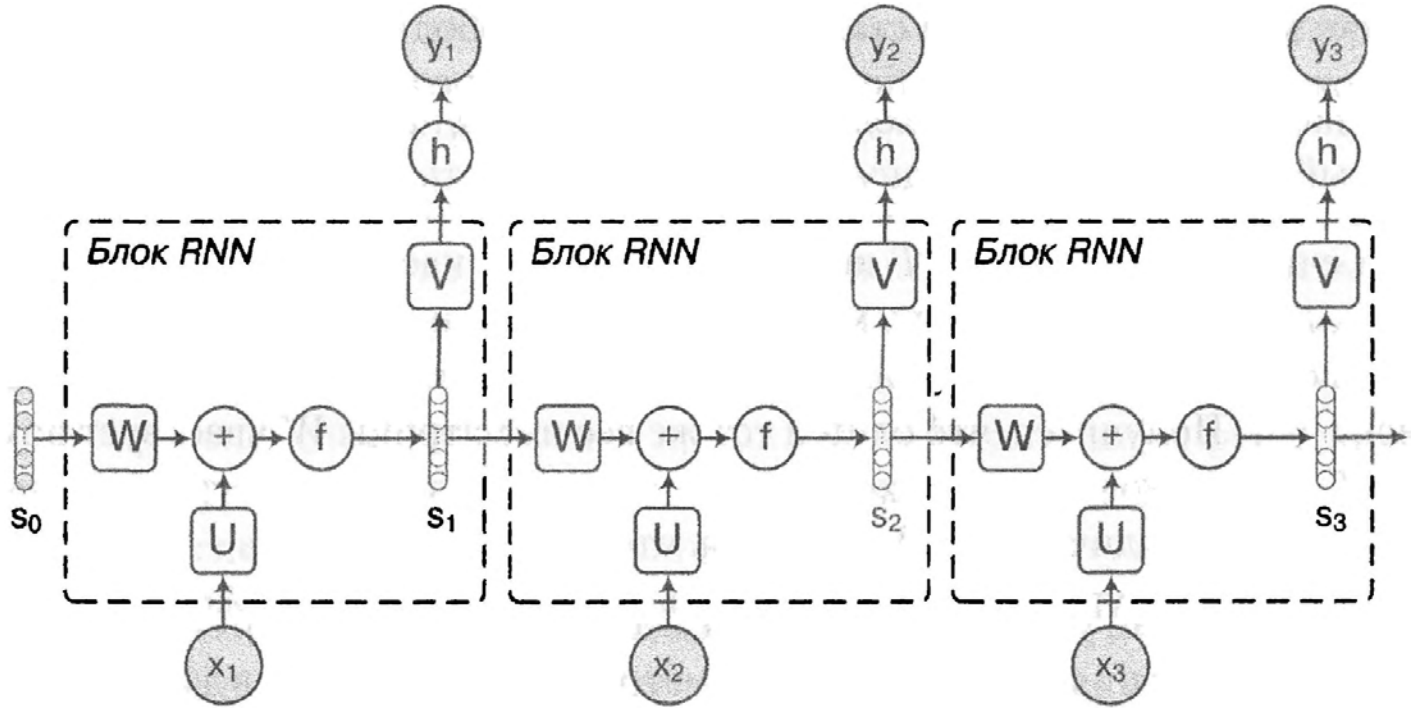
Архитектура «простой» рекуррентной нейронной сети

На каждом шаге сеть создает копию самой себя. Каждая из этих копий в определенный момент времени принимает на вход часть последовательности (текущее окно) и значение, полученное из предыдущей копии, затем их комбинирует и передает получившийся результат в следующий элемент.

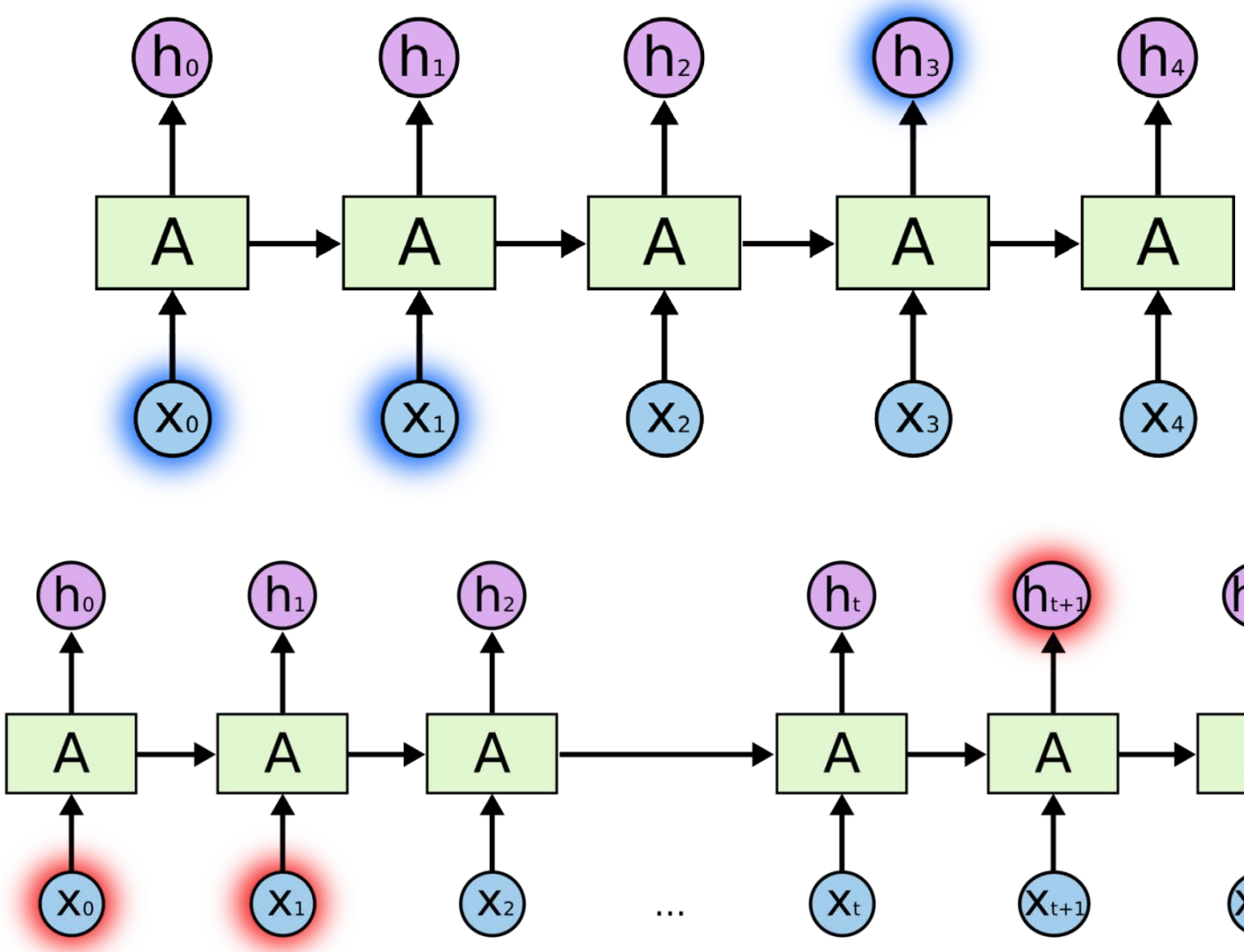
В итоге в момент времени t получаем следующее описание сети:

$$a_t = b + Ws_{t-1} + Ux_t \quad s_t = f(a_t)$$
$$o_t = c + Vs_t \qquad y_t = h(o_t)$$

где f — это нелинейность собственно рекуррентной сети (обычно σ - сигмоида, tanh или ReLU), а h — функция, с помощью которой получается ответ (например, softmax).



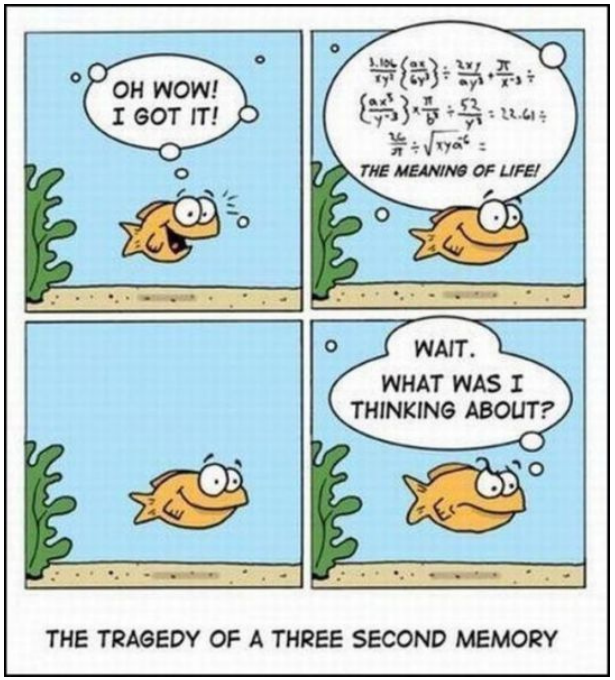
Проблема долговременных зависимостей

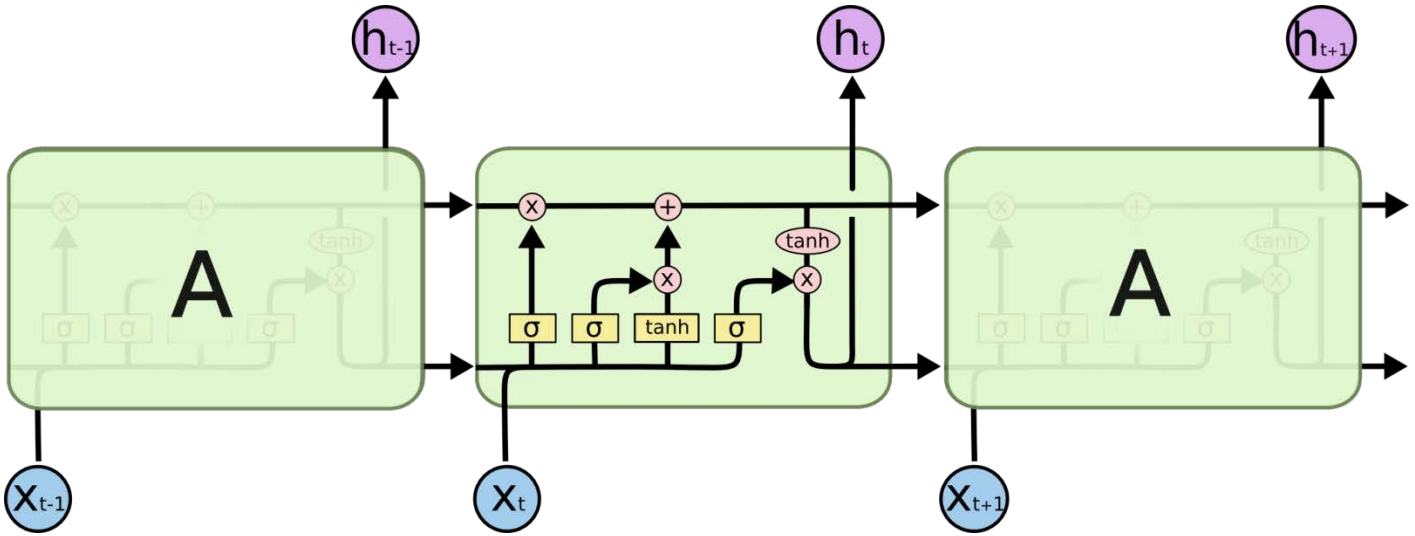


Долгая краткосрочная память – Long short-term memory (LSTM)

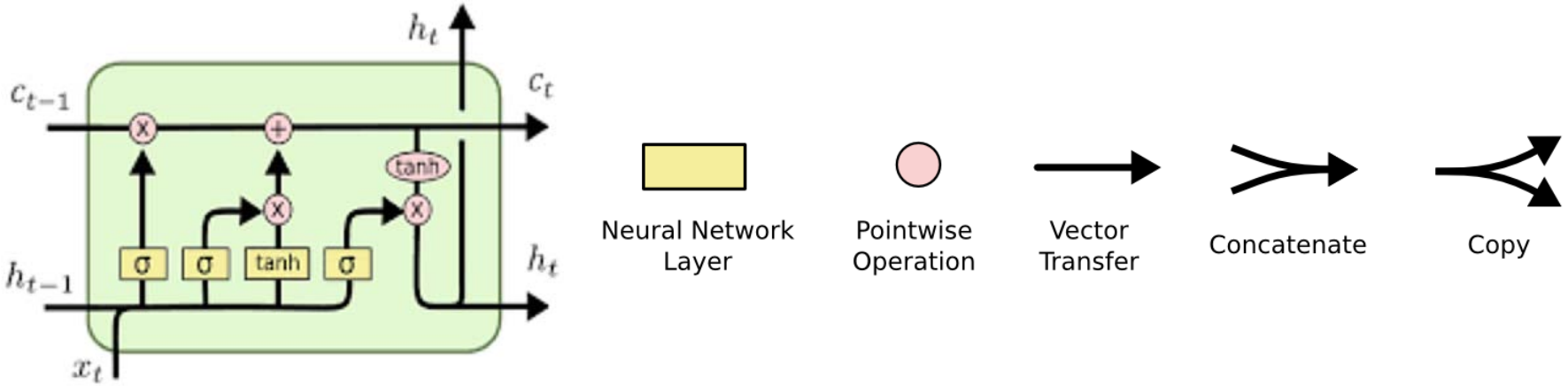
Особая разновидность архитектуры рекуррентных нейронных сетей, способная к обучению долговременным зависимостям, предложенная в 1997 году Сеппом Хохрайтером и Юргеном Шмидхубером.

LSTM-модули разработаны специально, чтобы избежать проблемы долговременной зависимости, запоминая значения как на короткие, так и на длинные промежутки времени. Хранимое значение не размывается во времени и градиент не исчезает при обучении с использованием метода обратного распространения ошибки во времени (Backpropagation Through Time - BPTT).





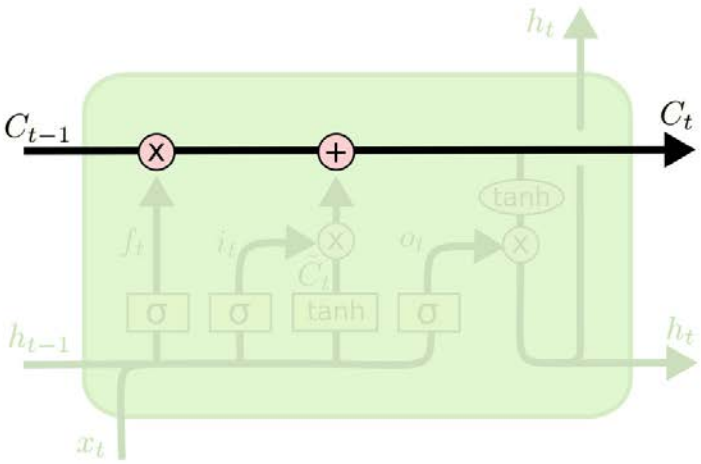
LSTM - специальные обозначения

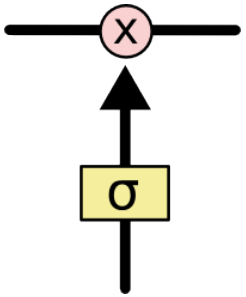


- Слой нейронной сети (Желтый прямоугольник);
- Поточечная операция (Розовая окружность) - например, сложение векторов;
- Векторный перенос - каждая линия переносит целый вектор от выхода одного узла ко входу другого;
- Объединение (Сливающиеся линии);
- Копирование (разветвляющиеся стрелки) - данные копируются и копии уходят в разные компоненты сети.

Ключевые компоненты LSTM-модуля

- **Состояние ячейки** (cell state) - память сети, которая передает соответствующую информацию по всей цепочке модулей.
- **Фильтры**, контролирующие состояние ячейки - контролируют поток информации на входах и на выходах модуля на основании некоторых условий. Состоят из слоя сигмоидальной нейронной сети и операции поточечного умножения.
  - Забывания - контролирует меру сохранения значения в памяти;
  - Входной - контролирует меру вхождения нового значения в память.
  - Выходной - контролирует меру того, в какой степени значение, находящееся в памяти, используется при расчёте выходной функции активации.

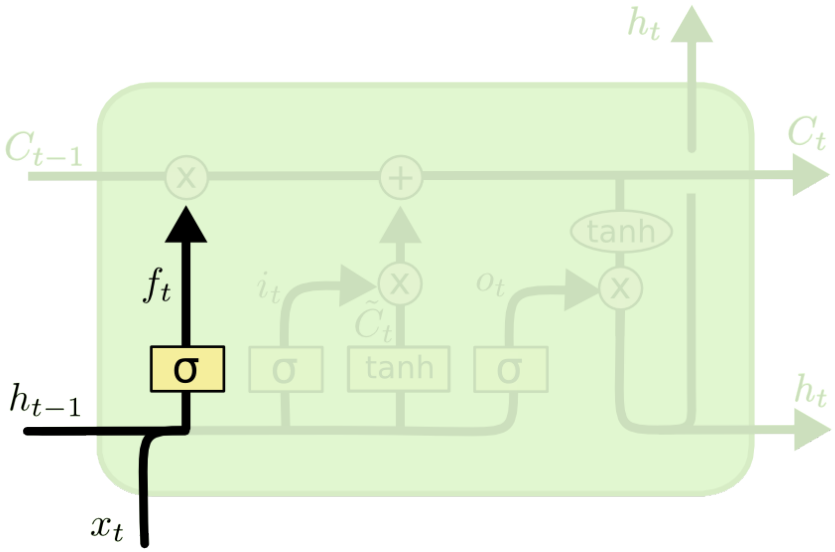




Принцип работы LSTM-модуля – Шаг 1

Определить, какую информацию можно выбросить из состояния ячейки. Для этого используется «**слой фильтра забывания**» (англ. ***forget gate layer***).

Значения предыдущего выхода h_{t-1} и текущего входа x_t пропускаются через сигмоидальный слой. Полученные значения находятся в диапазоне [0; 1]. Значения, которые ближе к 0 будут забыты, а к 1 оставлены.

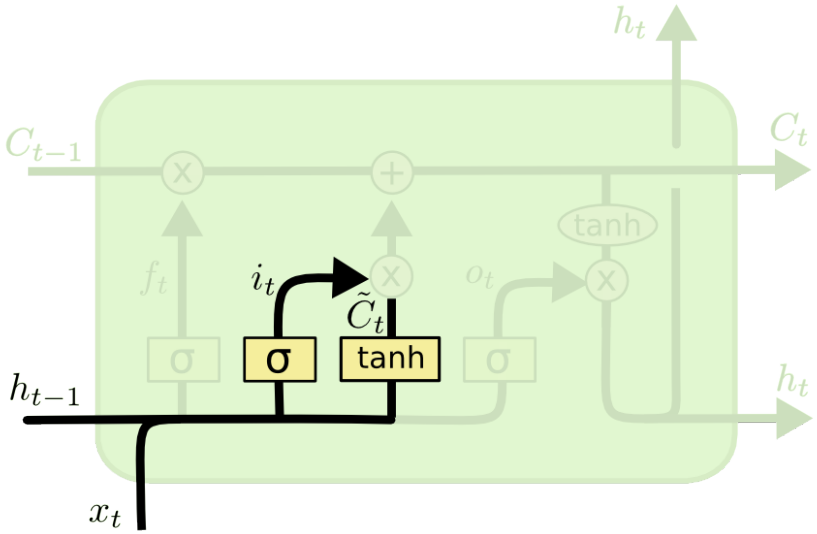


$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \; + \; b_f\right)$$

Принцип работы LSTM-модуля – Шаг 2

Далее решается, какая новая информация будет храниться в состоянии ячейки. Этот этап состоит из двух частей.

- Сначала сигмоидальный слой под названием “**слой входного фильтра**” (англ. ***input layer gate***) определяет, какие значения следует обновить.
- Затем tanh-слой строит вектор новых значений-кандидатов $\tilde{C}_t$, которые можно добавить в состояние ячейки.

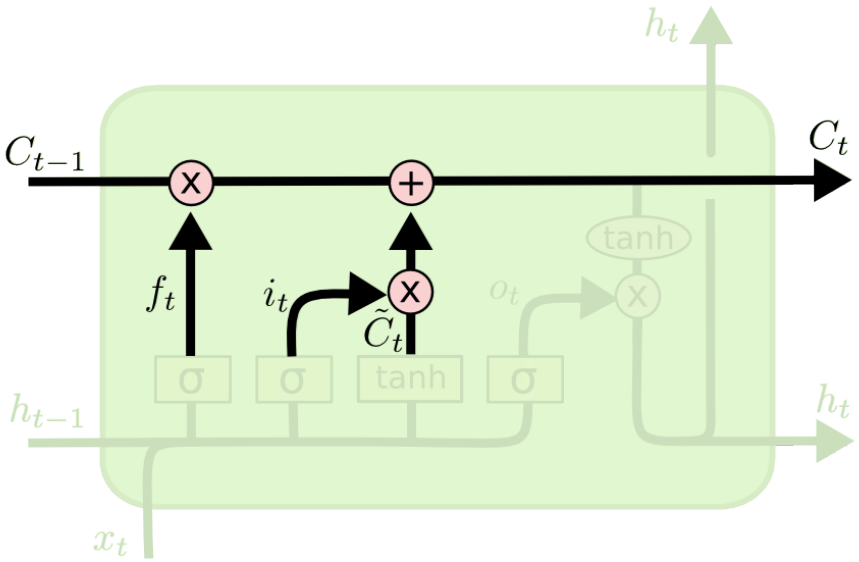


$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \; + \; b_C)$$

Принцип работы LSTM-модуля – Шаг 3

Для замены старого состояния ячейки C_{t-1} на новое состояние C_t. Необходимо умножить старое состояние на f_t, забывая то, что решили забыть ранее. Затем прибавляем $i_t * \tilde{C}_t$. Это новые значения-кандидаты, умноженные на i_t – на сколько обновить каждое из значений состояния.

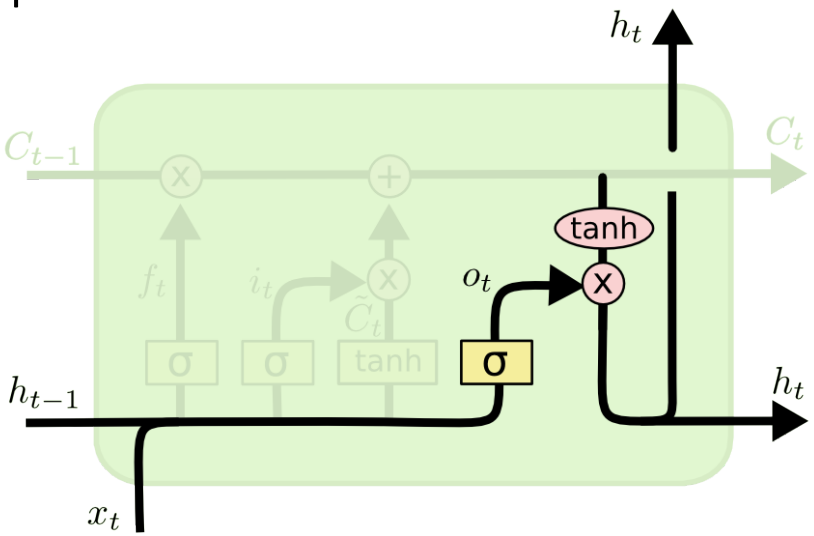


$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

Принцип работы LSTM-модуля – Шаг 4

На последнем этапе определяется то, какая информация будет получена на выходе. Выходные данные будут основаны на нашем состоянии ячейки, к ним будут применены некоторые фильтры. Сначала значения предыдущего выхода h_{t-1} и текущего входа x_t пропускаются через сигмоидальный слой, который решает, какая информация из состояния ячейки будет выведена. Затем значения состояния ячейки проходят через tanh-слой, чтобы получить на выходе значения из диапазона от -1 до 1, и перемножаются с выходными значениями сигмоидального слоя, что позволяет выводить только требуемую информацию.

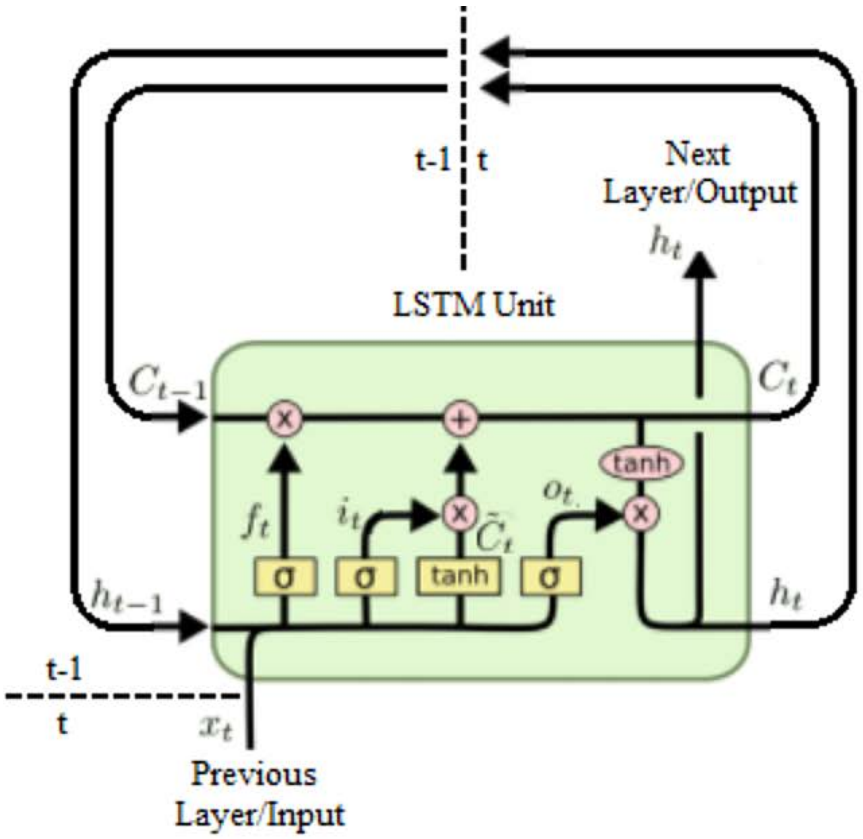
Полученные таким образом h_t и C_t передаются далее по цепочке.



$$o_t = \sigma\left(W_o\left[h_{t-1}, x_t\right] + b_o\right)$$

$$h_t = o_t * \tanh\left(C_t\right)$$

Работа LSTM-модуля



$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \; + \; b_f\right)$$
$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \; + \; b_C)$$
$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$
$$o_t = \sigma\left(W_o \; [h_{t-1}, x_t] \; + \; b_o\right)$$
$$h_t = o_t * \tanh\left(C_t\right)$$

Области и примеры применения

Используются, когда важно соблюдать последовательность, когда важен порядок поступающих объектов.

- Обработка текста на естественном языке:
  - Анализ текста;
  - Автоматический перевод;
- Обработка аудио:
  - Автоматическое распознавание речи;
- Обработка видео:
  - Прогнозирование следующего кадра на основе предыдущих;
  - Распознавание эмоций;
- Обработка изображений:
  - Прогнозирование следующего пикселя на основе окружения;
  - Генерация описания изображений.

Спасибо за внимание