Рекуррентные сети

Самарин Игорь, группа 23.М03-мм

Санкт-Петербургский государственный университет
Кафедра статистического моделирования

Санкт-Петербург
2024 г.

Нейронная сеть. Определение

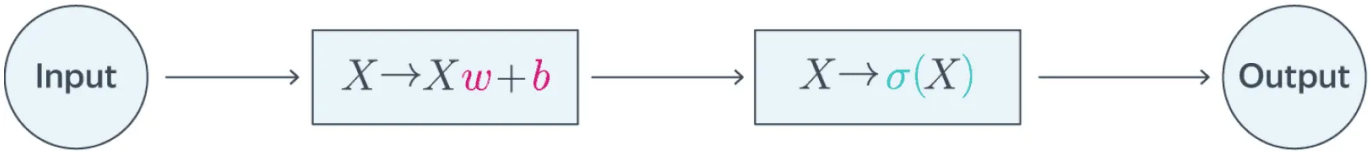
Искусственная нейронная сеть — это сложная дифференцируемая функция, задающая отображение из исходного пространства в пространство ответов, все параметры которой могут настраиваться одновременно и взаимосвязанно.

Сложную функцию удобно представлять в виде суперпозиции простых. Простейшие разновидности:

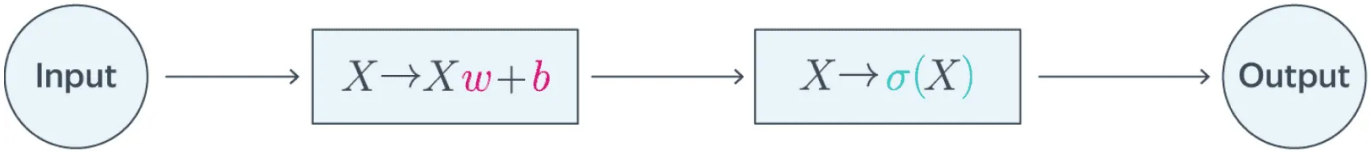
- **Линейный слой**: линейное преобразование над входящими данными. Обучаемые параметры — матрица W и вектор b такие, что $x \mapsto xW + b$, где $(W \in \mathbb{R}^{d \times k}, x \in \mathbb{R}^d, b \in \mathbb{R}^k)$;
- **Функция активации**: нелинейное преобразование, поэлементно применяющееся к пришедшим на вход данным.

Таким образом, нейронную сеть можно представить в виде вычислительного графа, где промежуточным вершинам соотствуют преобразования.

Нейронная сеть



Применение



$X^{in}=X^0$ → $X^1=X^0w+b$ → $X^2=\sigma(X^1)$ → $X^2=X^{Out}$

Приходит на вход

То что возвращает граф

Рекуррентная нейронная сеть. Мотивация

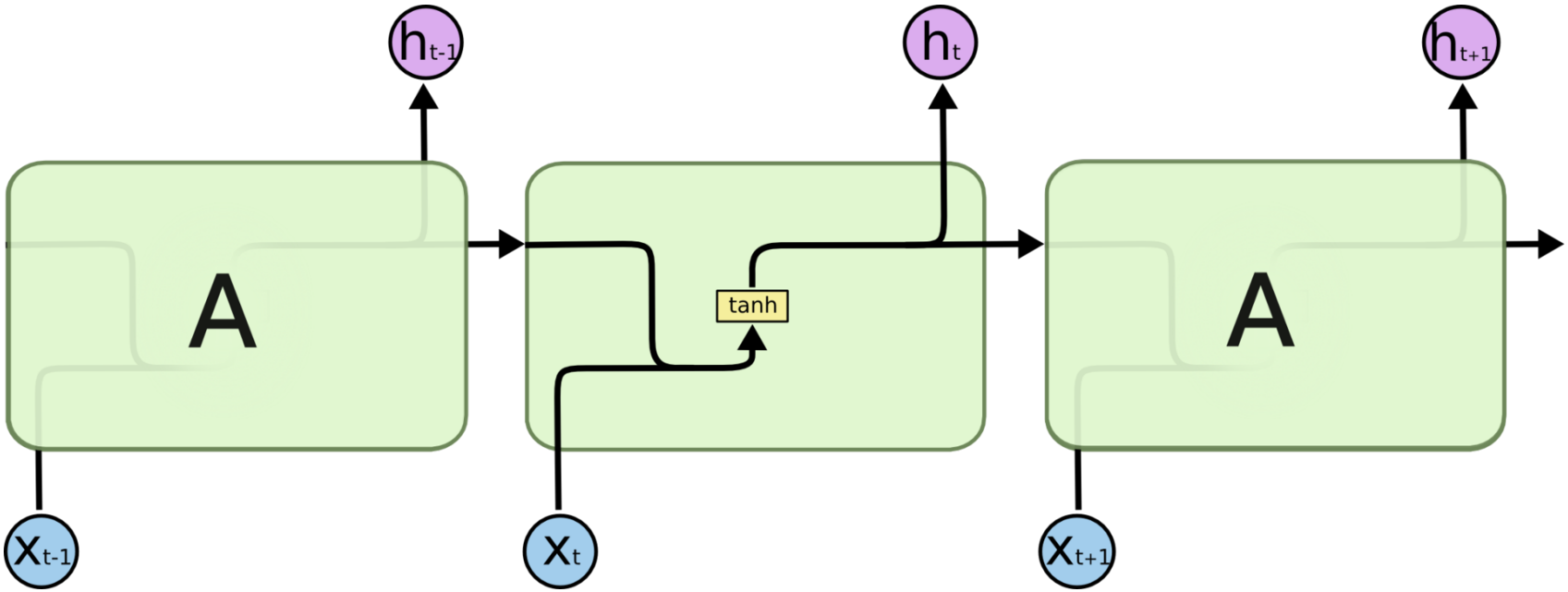
Традиционные нейронные сети не позволяют сохранять информацию и в этом их главный недостаток.

В отличие от других данных временные ряды и тексты содержат временную компоненту. Поэтому хотелось бы иметь возможность обрабатывать их последовательно, в строго определенном порядке.

Решить эту проблему помогают рекуррентные нейронные сети. Это сети, содержащие обратные связи и позволяющие сохранять информацию.

Рекуррентная нейронная сеть. Определение

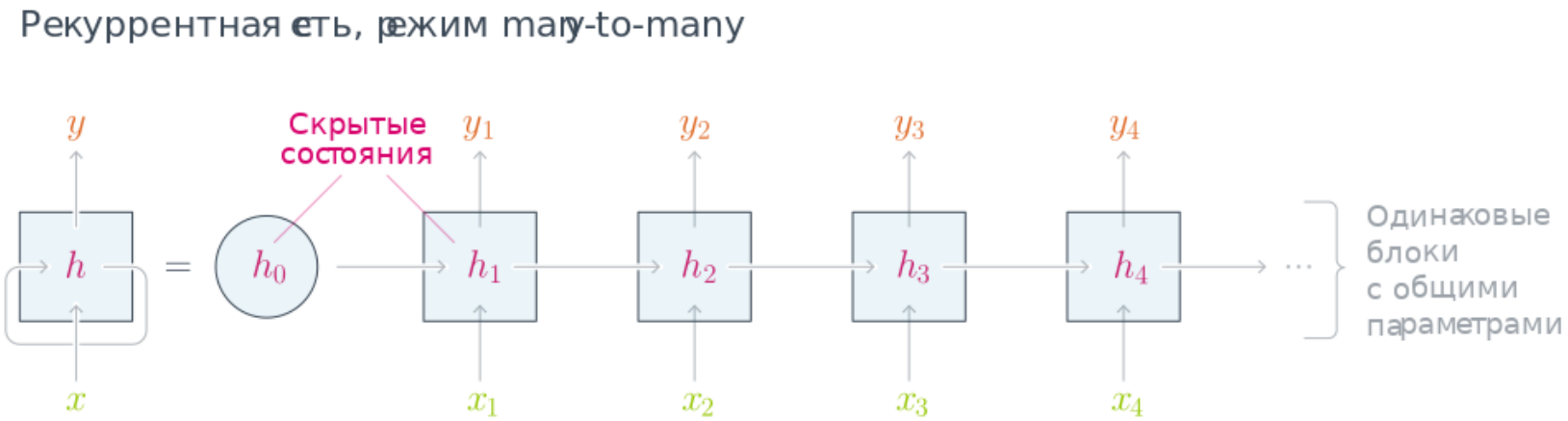
Рекуррентная нейронная сеть — это вид нейронной сети, где связи между элементами образуют направленную последовательность.



Рекуррентные сети представимы в виде цепочки блоков. В нашем случае, повторяющимся блоком является линейный слой с гиперболическим тангенсом в качестве активации.

Рекуррентная нейронная сеть. Скрытое состояние

Рассмотрим идею рекуррентных нейронных сетей:



На схеме, скрытое состояние h принимает входное значение x и возвращает значение y. Наличие обратной связи позволяет передавать информацию от одного шага сети к другому.

Рекуррентная нейронная сеть. Скрытое состояние

На каждом шаге в сеть подаются данные, при этом происходит обновление скрытого состояния:

$$h_n = \tanh(h_{n-1}W_1 + x_n W_2),$$

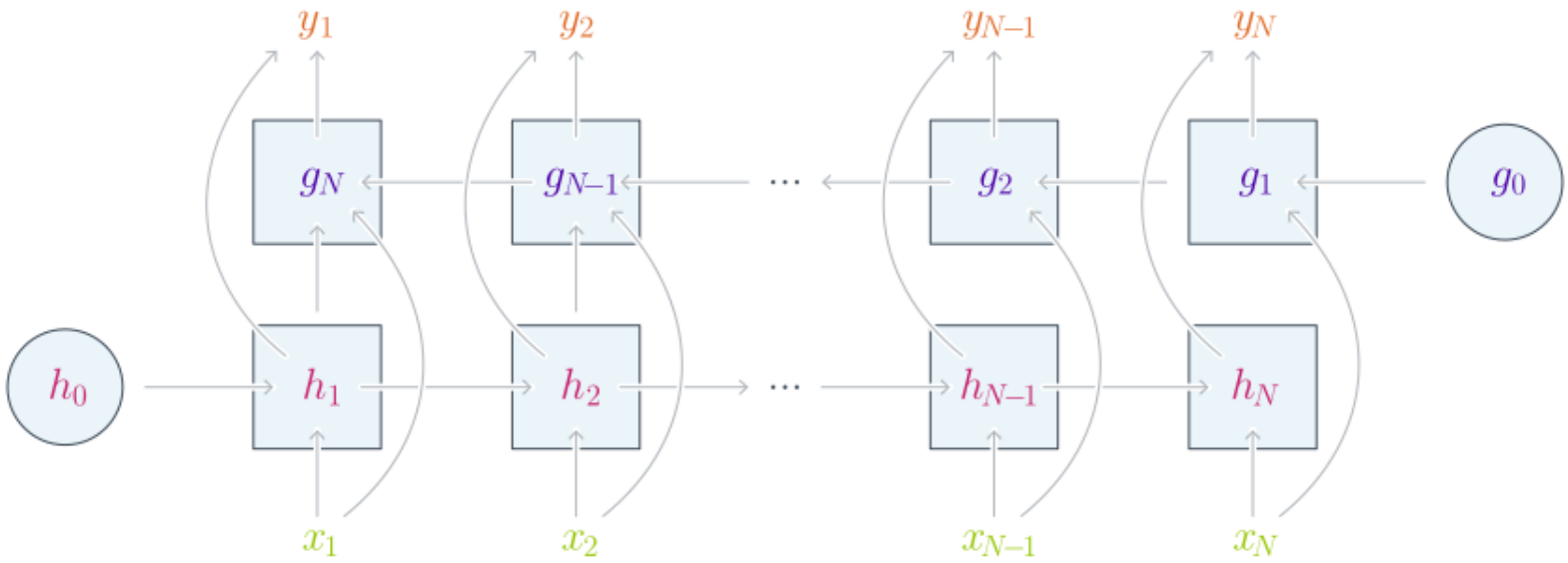
после чего, по скрытому состоянию, предсказывается выходной сигнал:

$$y_n = h_n W_3,$$

где W_i одинаковы на всех итерациях.

Двунаправленная рекуррентная нейронная сеть

Двунаправленная рекуррентная нейронная сеть позволяет учитывать не только предыдущие значения, но и последующие.



$$y_n = [h_n, g_n]\, w + b$$

Конкатенация

Формула для y_n может быть другой. Например, выходы сетей могут агрегироваться путем усреднения, или любым другим способом.

Взрыв и затухание градиента

Рассмотрим функцию потерь $\mathfrak{L}_n = L(y_n, \hat{y}_n)$, измеряющую отклонение n-го выхода от истинного. Выход y_n зависит от скрытого состояния h_n, а тот, в свою очередь, от всех h_i, $i < n$.

Обновление градиента при $h_i = \tanh(h_{i-1}W_1 + x_i W_2)$ будет иметь вид:

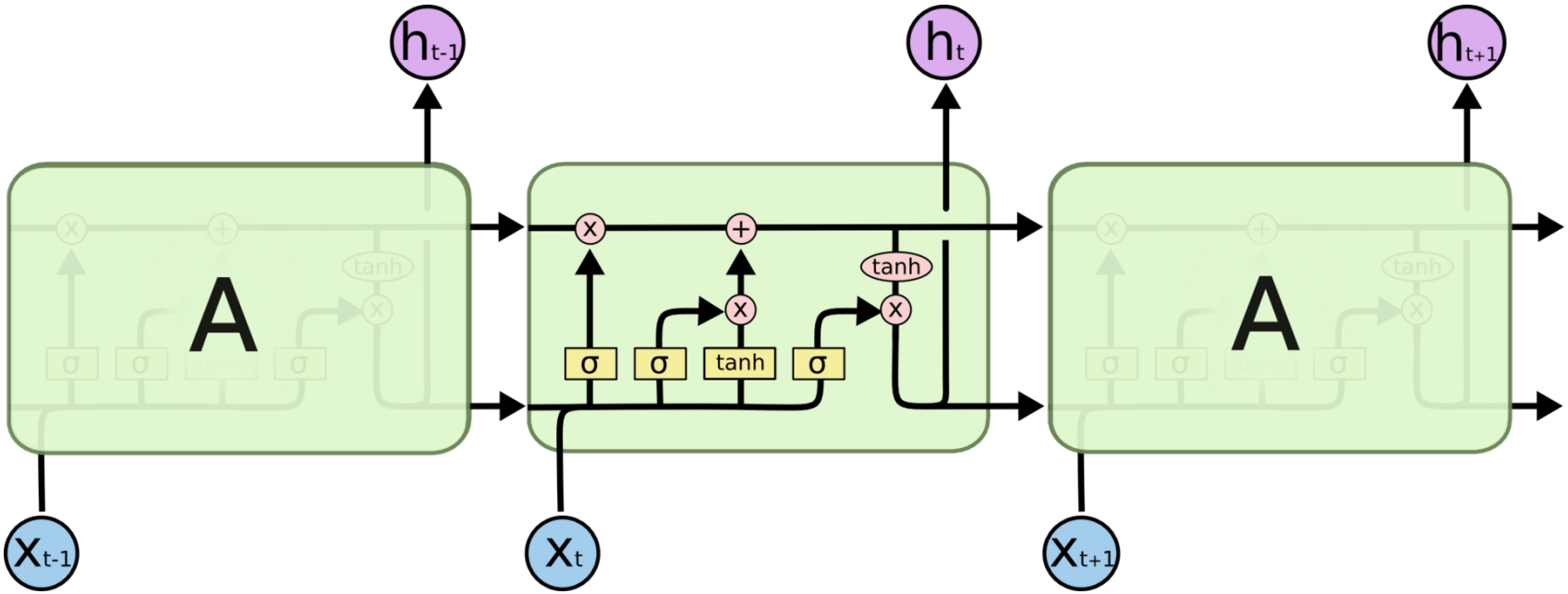
$$\nabla_{h_{i-1}} L = (\nabla_{h_{i-1}} L W_1^T \cdot \tanh'(h_{i-1}W_1 + x_i W_2)),$$

т.е. в ходе вычисления $\nabla_{W_1} \mathfrak{L}_n$ мы $(n-1)$ раз будем умножать на W_1^T. Если у нее есть с.з., по модулю большие 1, градиент будет стремиться к бесконечности (взрываться).

Вариант решения: gradient clipping — заменять значения градиента выше некоторого порога на некоторую константу.

LSTM

Сеть с долговременной и кратковременной памятью частично решает проблему взрыва и затухания градиента.



Рассмотрим механизм подробнее.

LSTM. Состояние блока

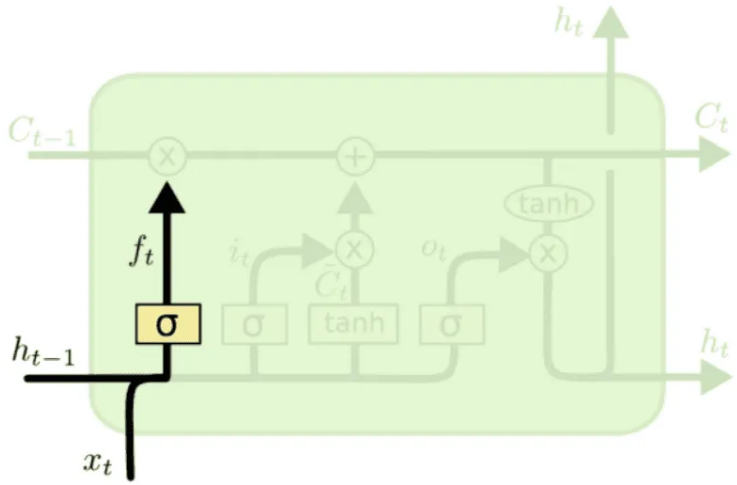
Помимо скрытого состояния h_n, появляется понятие cell state c_n.

Cell state играет роль внутренней, закрытой информации LSTM блока, тогда как скрытое состояние становится значением передаваемым наружу.

LSTM может добавлять или удалять информацию из cell state с помощью вентилей.

LSTM. Забывание информации

Вентиль забывания, по предыдущему скрытому состоянию h_{t-1} и входу x_t, позволяет определить, какую долю информации из c_{n-1} стоит пропустить, а какую забыть.



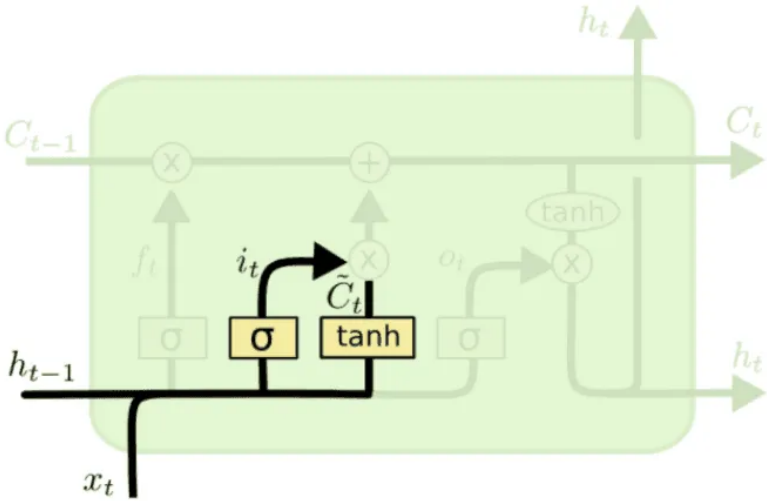
Доля f_t сохраняемой информации из c_{t-1} вычисляется:

$$f_t = \sigma(h_{t-1}W_1^f + x_tW_2^f + b_f).$$

LSTM. Сохранение информации

Необходимо определить, чего нового мы внесем в cell state. Для этого вычисляем:

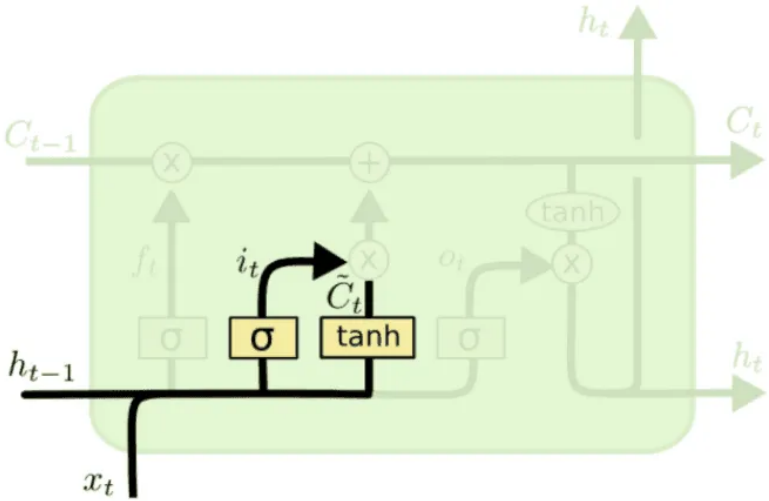
$$\tilde{c}_t = \tanh(h_{t-1}W_1^c + x_t W_2^c + b_c).$$



Однако, мы не уверены, что вся информация достойна переноса в cell state, поэтому хотим взять лишь долю.

LSTM. Сохранение информации

Вентиль входного состояния позволяет определить, какую долю релевантной информации стоит сохранить.



Доля i_t сохраняемой информации вычисляется:

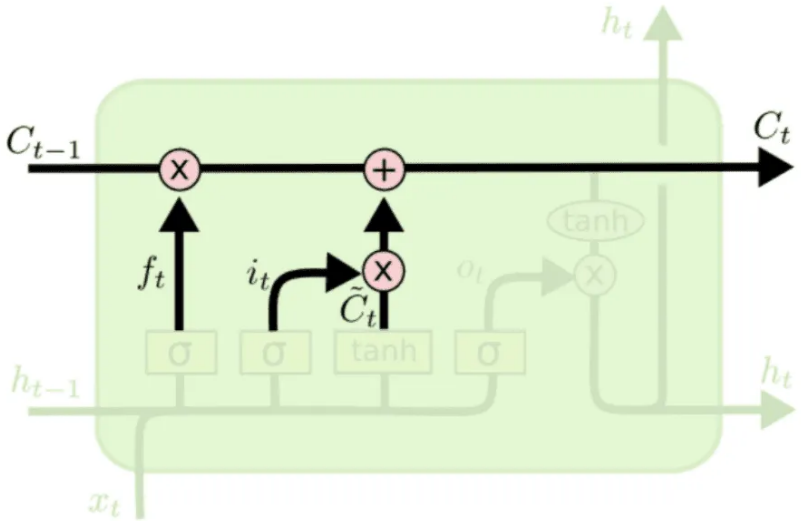
$$i_t = \sigma(h_{t-1}W_1^i + x_tW_2^i + b_i).$$

LSTM. Новое состояние блока

Новое состояние блока вычисляется:

$$c_t = f_t \cdot c_{t-1} + i_t \cdot \tilde{c}_t,$$

где $\cdot$ — это поэлементное умножение.

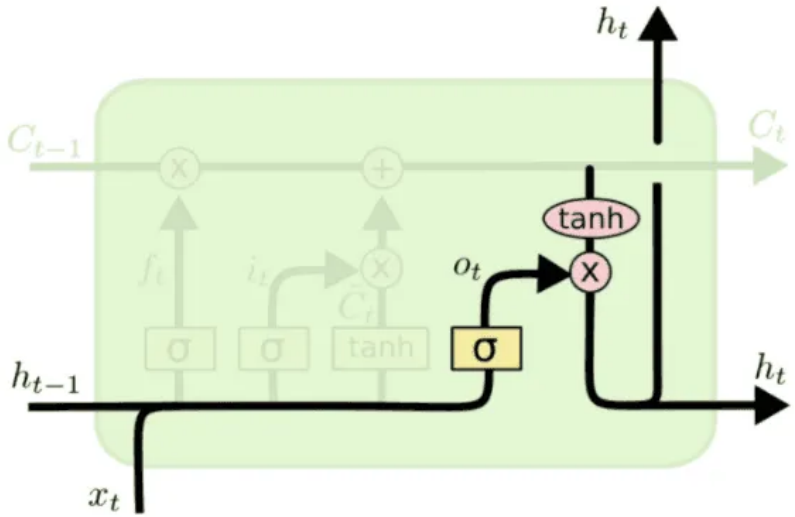


Первое слагаемое отвечает за забывание информации из c_{t-1}, а второе — за запоминание новой.

LSTM. Выходное значение

Вентиль выходного состояния позволяет определить, сколько информации следует отдать на выход. Доля вычисляется:

$$o_t = \sigma(h_{t-1}W_1^o + x_tW_2^o + b_o).$$

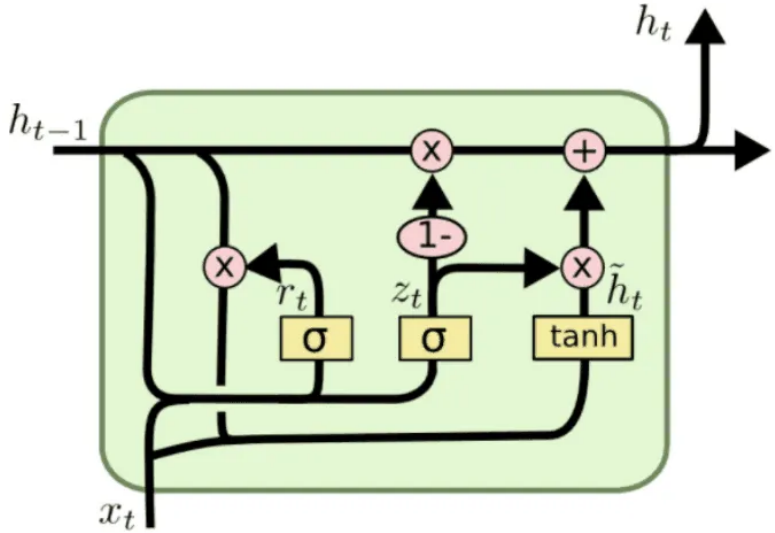


Чтобы отфильтровать информацию из cell state, вычисляем:

$$h_t = o_t \cdot \tanh(c_t).$$

GRU

Вычисление четырех гейтов может быть вычислительно трудным, поэтому иногда используют GRU.



GRU объединяет input gate и forget gate в один update gate, а также устраняет разделение на hidden и cell state.

GRU

В итоге GRU имеет меньше параметров и быстрее учится.

Более того, GRU и LSTM показывают сопостовимое качество на многих задачах обработки естественного языка и генерации музыки.