

Нейронные сети и их практическое применение.

Лекция 8. Рекуррентные сети.

Дмитрий Буряк
к.ф.-м.н
dyb04@yandex.ru

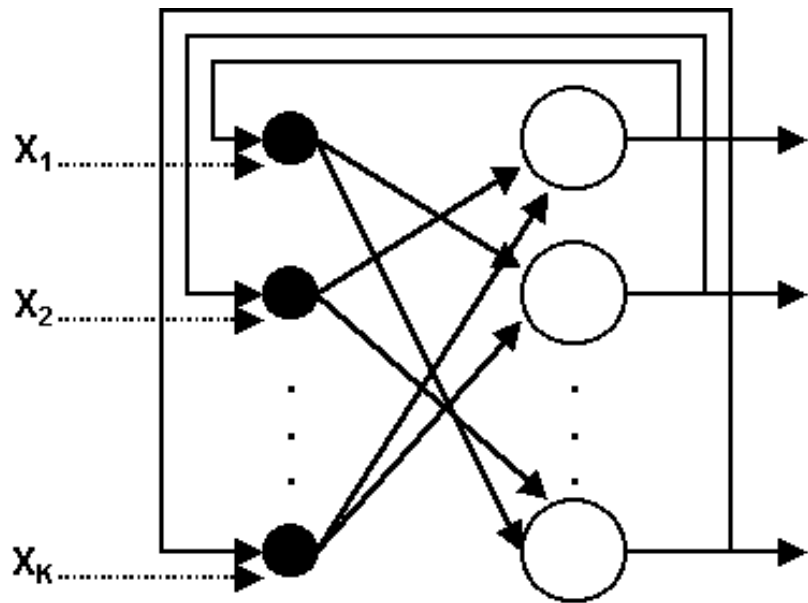
Содержание

- Ассоциативная память
 - Ассоциативная память: сеть Хопфилда.
 - Гетероассоциативная память: сеть Хемминга.

- Рекуррентные сети на базе персептрона
 - Recurrent MultiLayer Perceptron
 - Рекуррентная сеть Эльмана

- Рекуррентные сети LSTM, GRU.

Сеть Хопфилда



$$y_i(n) = f\left(\sum_{j=1, j \neq i}^K w_{ji} y_j(n-1)\right)$$
$$y_i(0) = x_i$$

Условие окончания: $y_i(n) = y_i(n-1)$

Достаточное условие
устойчивости:

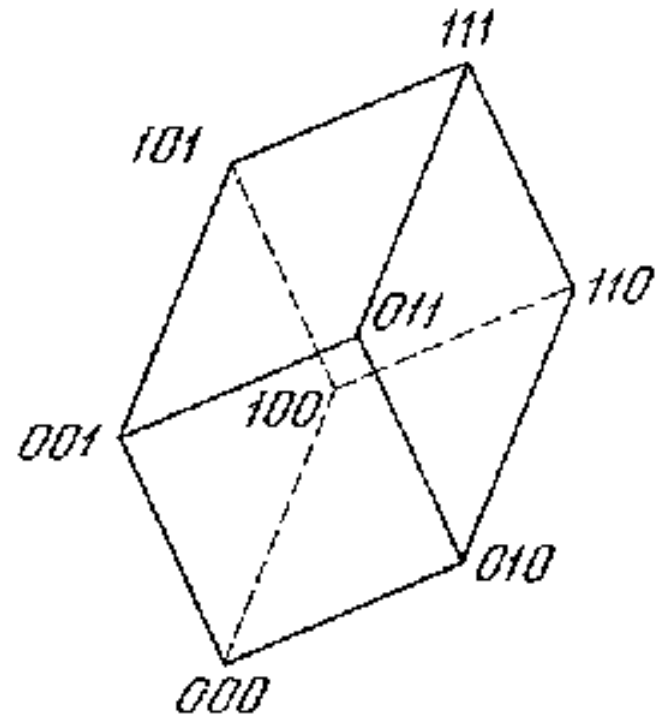
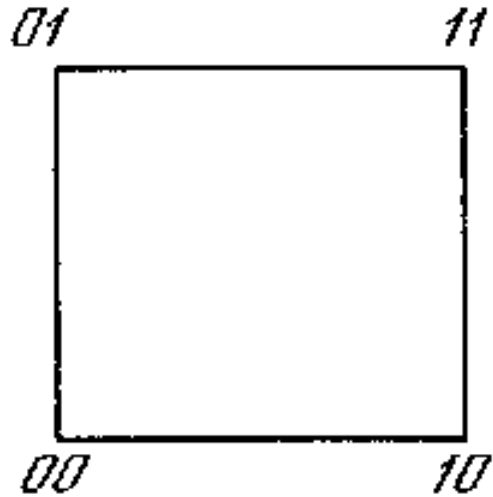
$$w_{ij} = w_{ji}$$

$$w_{ii} = 0$$

Бинарные системы

$$y_i = f(u_i) = \begin{cases} 1, u_i > 0 \\ -1, u_i \leq 0 \end{cases}$$

$$u_i = \sum_{j=1}^K w_{ji} z_j$$



Обучение методом проекций

$$WX = X$$

W - матрица весов $K \times K$;

X - матрица $K \times P$, составленная из обучающих векторов.

$$W = XX^+$$

$$W = X(X^T X)^{-1} X^T$$

$$W^{(i)} = W^{(i-1)} + \frac{1}{[x^{(i)}]^T x^{(i)} - [x^{(i)}]^T W^{(i-1)} x^{(i)}} \times [W^{(i-1)} x^{(i)} - x^{(i)}] \times [W^{(i-1)} x^{(i)} - x^{(i)}]^T$$

$x^{(i)}$ - обучающий вектор;

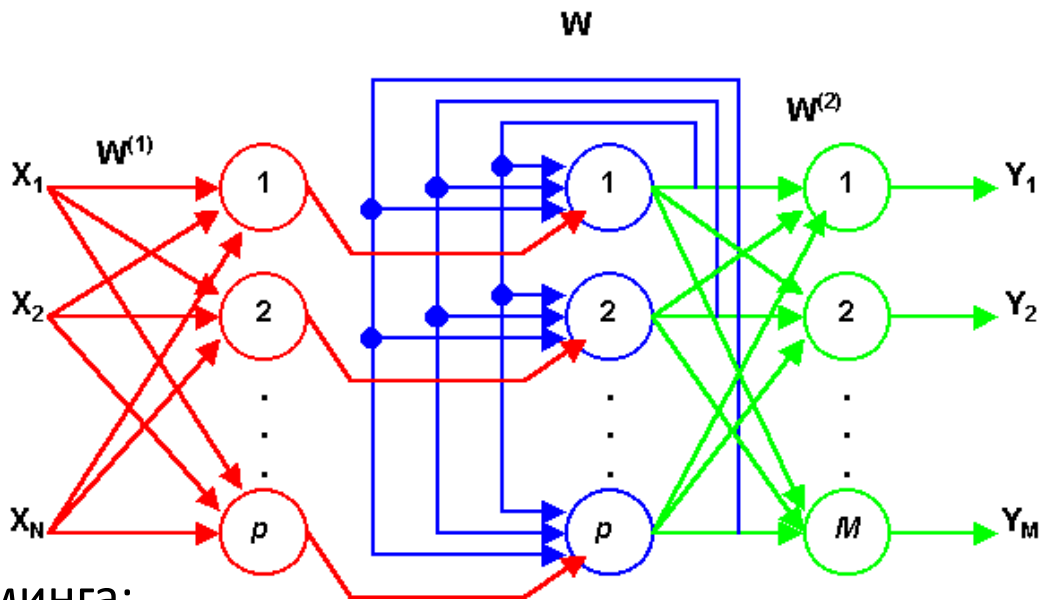
$W^{(0)} = 0$;

Емкость сети Хопфилда: $K-1$

Сеть Хопфилда. Области применения

1. Ассоциативная память.
2. Распознавание образов.
3. Задачи оптимизации
 - Задача коммивояжера
 - Составление расписания
 - Аналого-цифровое преобразование
 - другие NP-полные проблемы
4. Математика
 - инвертирование матрицы
 - решение систем линейных и нелинейных уравнений

Сеть Хемминга



Расстояние Хемминга:

Двоичные вектора:
$$d_H(y, d) = \sum_{i=1}^n [d_i(1 - y_i) + (1 - d_i)y_i]$$

Биполярные вектора:
$$d_H(y, d) = \frac{1}{2} \left[n - \sum_{i=1}^n y_i d_i \right]$$

Сеть Хемминга. Вычисление.

Обрабатываемые данные: биполярные вектора.

Входной вектор: x

1. Вычисление расстояния Хемминга между входным вектором и образцами, закодированными в весах первого слоя. $y'_i = 1 - \frac{d_H(x^{(i)}, x)}{N}$

2. Выбор образца с наименьшим расстоянием (сеть MAXNET).

$$y_j(k) = f\left(\sum_i w_{ij} y_i(k-1)\right) = f\left(y_j(k-1) + \sum_{i \neq j} w_{ij} y_i(k-1)\right)$$

$$y_j(0) = y'_j$$

3. Формирование выходного вектора, соответствующего входному вектору.

$$f(y) = \begin{cases} y, & y \geq 0 \\ 0, & y < 0 \end{cases}$$

Сеть Хемминга. Обучение

Обучающая выборка: $\{(x^{(j)}, y^{(j)})\} j=1, 2, \dots, p$.

Слой 1: $w_{ij}^{(1)} = x_i^{(j)}$

Слой MAXNET:

$$w_{ii} = 1$$

$$-\frac{1}{p-1} < w_{ij} < 0$$

$$w_{ij} = -\frac{1}{p-1} + \xi$$

Выходной слой: $w_{ij}^{(2)} = y_i^{(j)}$

Сеть Хемминга. Особенности

□ Емкость сети: P (число нейронов первого слоя).

□ Достоинства:

1. Простой алгоритм работы.
2. Простой алгоритм обучения.
3. Емкость не зависит от размерности входного сигнала.

□ Недостатки:

1. Неопределенность результата, при одинаковом расстоянии до двух и более векторов.
2. Способность распознавать только слабозашумленные образы.
3. Бинарные (биполярные) входные вектора.

Сравнение сетей Хопфилда и Хемминга

Задача кластеризации.

Вход: 100.

Количество кластеров: 10.

Количество связей:

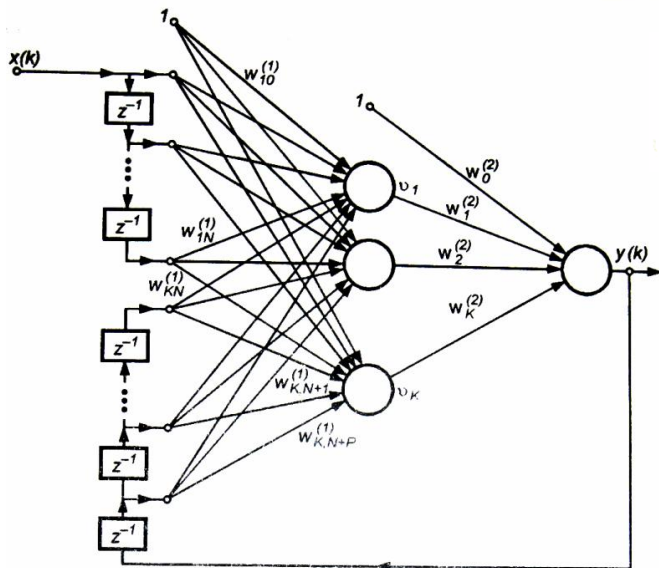
Хопфилд: $100 * 100 = 10000$;

Хемминг: $1000 + 100 = 1100$.

Персептронная сеть RMLP

RMLP – Recurrent MultiLayer Perceptron

NARX - Non-linear Auto-Regressive with eXogeneous inputs



$$u_i = \sum_{j=0}^{N+P} w_{ij}^{(1)} x_j \quad g = \sum_{i=0}^K w_i^{(2)} f(u_i)$$

$$v_i = f(u_i) \quad y = f(g)$$

Один входной нейрон, K скрытых и 1 выходной нейрон

$X(k) = (x(k), x(k-1), \dots, x(k-(N-1)), y(k-1), \dots, y(k-P))$

$N-1$ – количество задержек входного сигнала

P – количество задержек выходного сигнала

Применение. Моделирование динамических систем.

Прогнозирование

RMLP: алгоритм обучения

BPTT – Backpropagation Through Time

Метод наискорейшего спуска, online режим

Функция ошибки (один выход):

$$E(k) = \frac{1}{2}[y(k) - d(k)]^2$$

1. Случайная инициализация весов

2. Для момента времени t и входного вектора $x(t)$ вычислить состояния всех нейронов сети

$$u_i = \sum_{j=0}^{N+P} w_{ij}^{(1)} x_j \quad g = \sum_{i=0}^K w_i^{(2)} f(u_i)$$

$$v_i = f(u_i) \quad y = f(g)$$

3. Вычислить значения производных

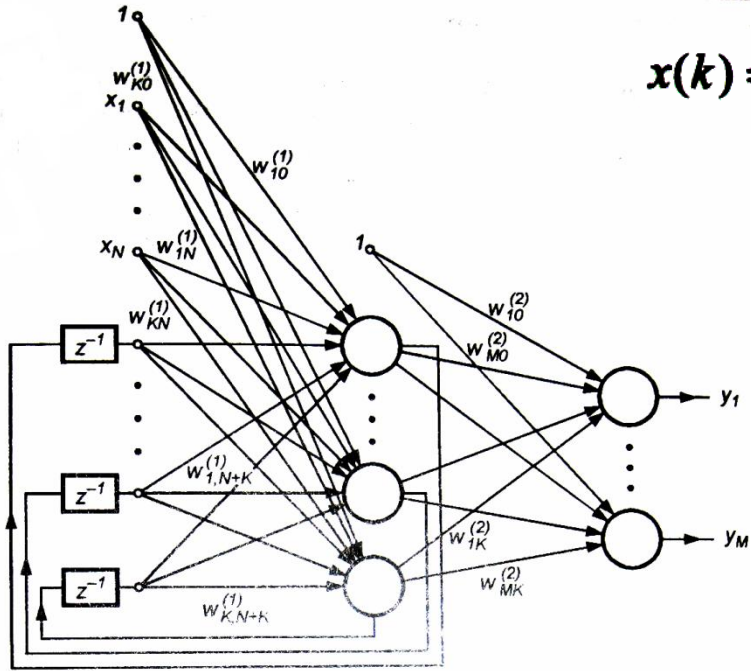
$$\frac{dy(k)}{dw_{\alpha,\beta}^{(1)}} = \frac{df(g(k))}{dg(k)} \sum_{i=1}^K w_i^{(2)} \frac{df(u_i(k))}{du_i^{(k)}} \left[\sum_{j=1}^P w_{i,j+N}^{(1)} \frac{dy(k-P-1+j)}{dw_{\alpha}^{(2)}} + \delta_{i\alpha} x_{\beta} \right]$$

$$\frac{dy(k)}{dw_{\alpha}^{(2)}} = \frac{df(g(k))}{dg(k)} \left[v_{\alpha}(k) + \sum_{i=0}^K w_i^{(2)} \frac{df(u_i(k))}{du_i^{(k)}} \sum_{j=1}^P w_{i,j+N}^{(1)} \frac{dy(k-P-1+j)}{dw_{\alpha}^{(2)}} \right]$$

4. Изменить веса $\Delta w_{\alpha}^{(2)} = -\eta[y(k) - d(k)] \frac{dy(k)}{dw_{\alpha}^{(2)}}$ $\Delta w_{\alpha\beta}^{(1)} = -\eta[y(k) - d(k)] \frac{dy(k)}{dw_{\alpha\beta}^{(1)}}$

5. Перейти на шаг 2.

Рекуррентная сеть Эльмана



$$\mathbf{x}(k) = [x_0(k), x_1(k), \dots, x_N(k), v_1(k-1), v_2(k-1), \dots, v_K(k-1)]$$

$$u_i(k) = \sum_{j=0}^{N+K} w_{ij}^{(1)} x_j(k) \quad v_i(k) = f_1(u_i(k))$$

$$g_i(k) = \sum_{j=0}^K w_{ij}^{(2)} v_j(k) \quad y_i(k) = f_2(g_i(k))$$

Применение.

- Анализ естественного языка: классификация слов в тексте с учетом контекста.
- Прогнозирование: предсказание трафика

RTRN – Real Time Recurrent Network – частный случай сети Эльмана

Сеть Эльмана. Обучение

Метод наискорейшего спуска, online режим

Функция ошибки:
$$E(k) = \frac{1}{2} \sum_{i=1}^M [y_i(k) - d_i(k)]^2 = \frac{1}{2} \sum_{i=1}^M e_i(k)^2$$

1. Случайная инициализация весов, равномерное из $[-1,1]$
2. Для момента времени t сформировать входной вектор $x(t)$
3. Вычислить вектор градиента

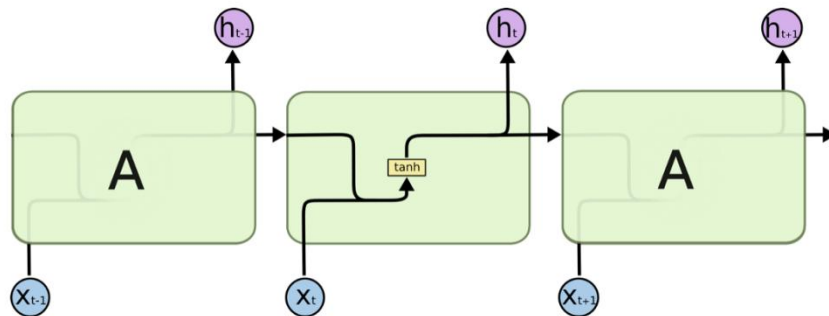
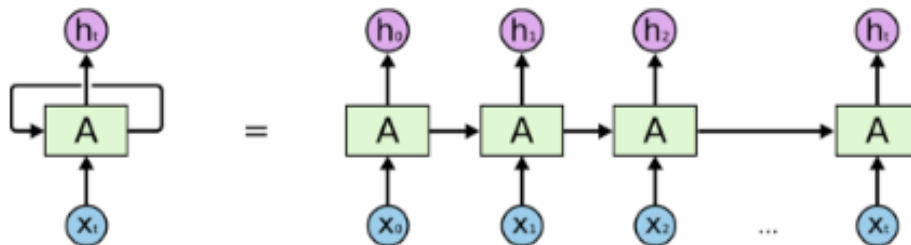
4. Изменить веса
$$w_{\alpha,\beta}^{(2)}(k) = w_{\alpha,\beta}^{(2)}(k-1) - \eta \nabla_{\alpha,\beta}^{(2)} E(k)$$

$$w_{\alpha,\beta}^{(1)}(k) = w_{\alpha,\beta}^{(1)}(k-1) - \eta \nabla_{\alpha,\beta}^{(1)} E(k)$$

5. Перейти на шаг 2.

Современные рекуррентные НС

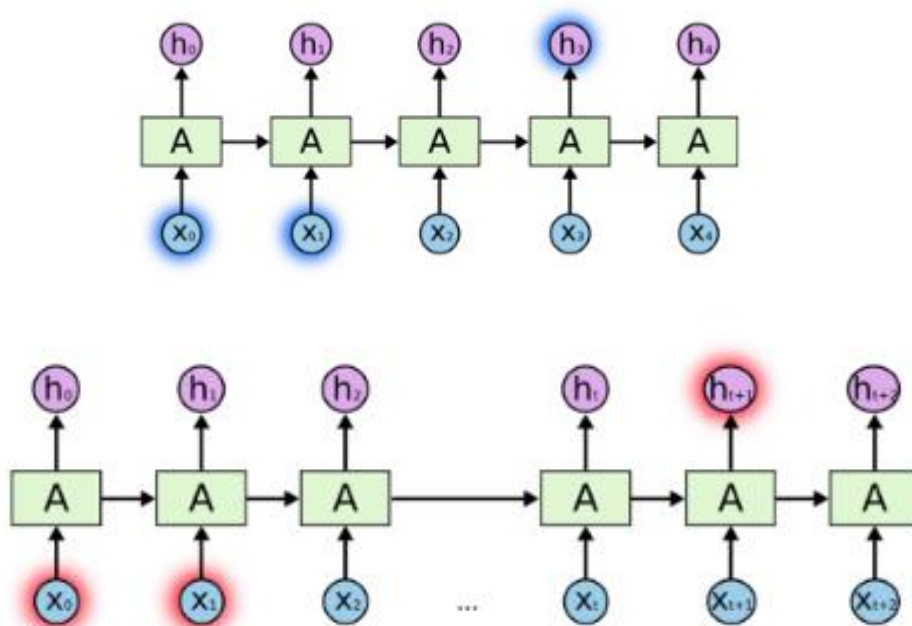
- Обратная связь → «запоминание» информации



$$h_t = \tanh(W_{hh}h_{t-1} + W_{xt}x_t).$$

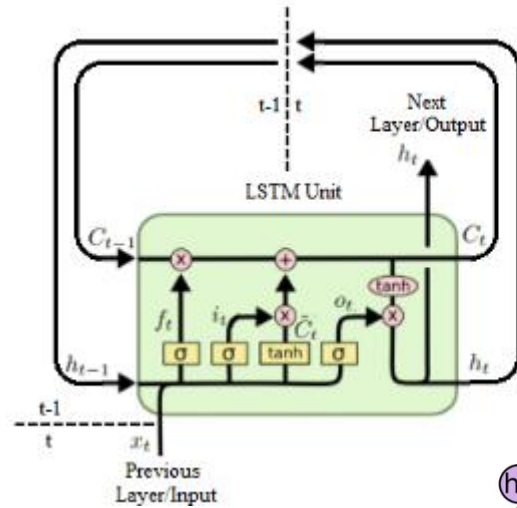
Краткосрочная память

- Короткий интервал «запоминания» информации



LSTM

- Long Short Term Memory – «запоминание» долгосрочных зависимостей.



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

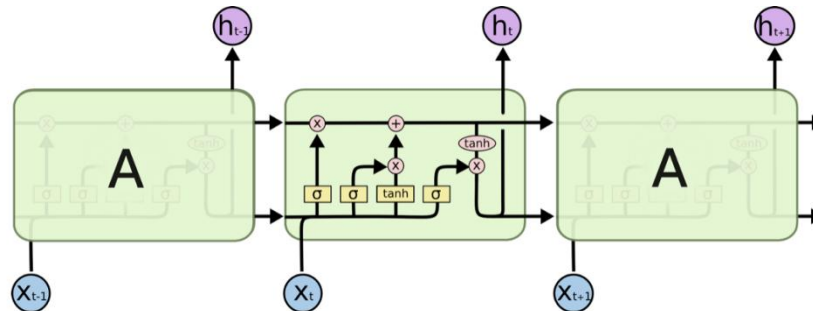
$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

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

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

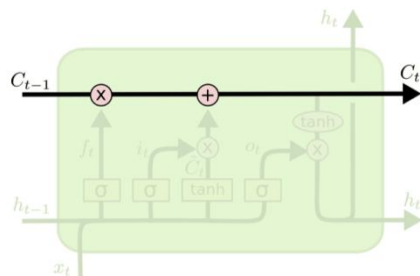
$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$$

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

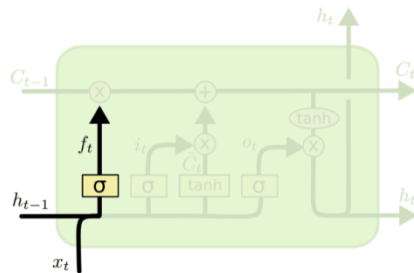


LSTM. Компоненты (1)

- ❑ Cell state - Состояние сети



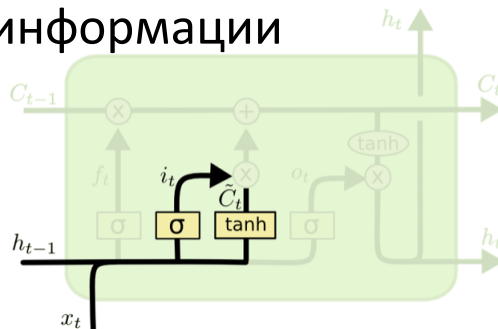
- ❑ Forget gate – контроль «забывания»



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

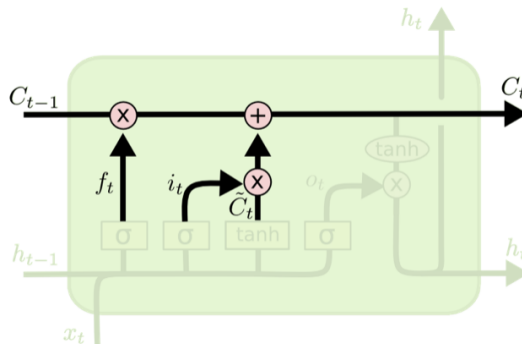
LSTM. Компоненты (2)

- ❑ Новая информация
- ❑ Input gate – добавление информации



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

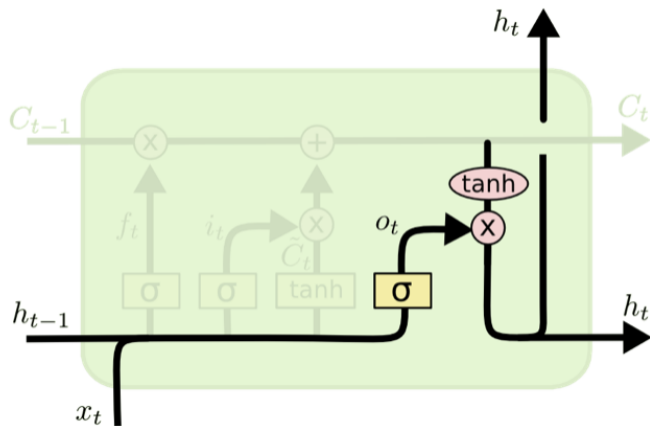
- ❑ Изменение состояния



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

LSTM. Компоненты (3)

- Выход сети

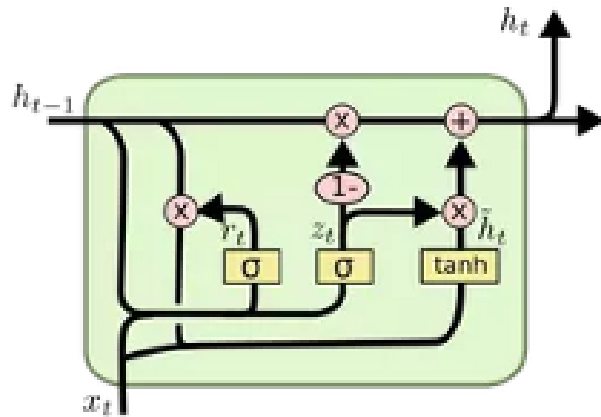


$$o_t = \sigma (W_o [h_{t-1}, x_t] + b_o)$$

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

GRU

- Gated Recurrent Unit.



$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t])$$

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t])$$

$$\tilde{h}_t = \tanh(W \cdot [r_t * h_{t-1}, x_t])$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$