

рекуррентные искусственные нейронные сети и обработка последовательностей

Евгений Борисов

рекуррентные нейронные сети

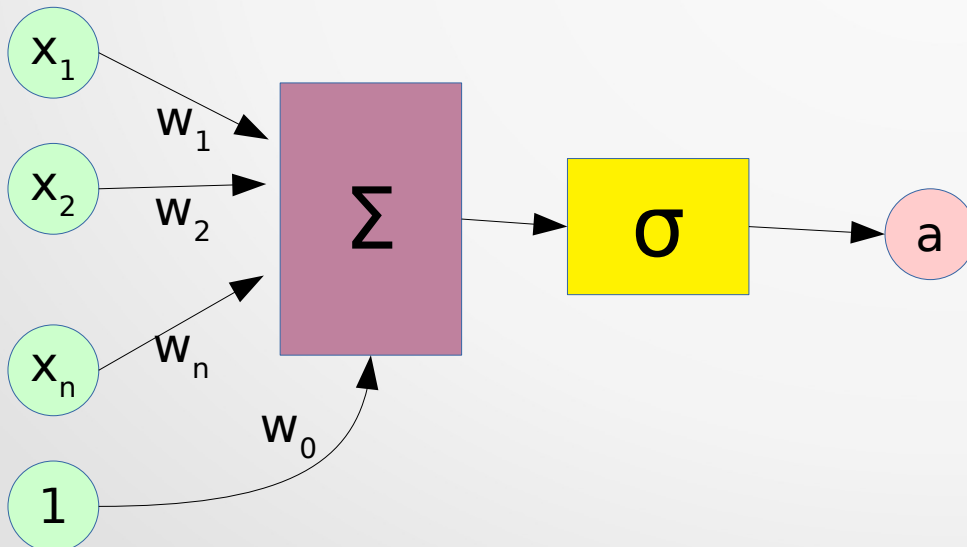
модель МакКаллока-Питтса (1943)

$$a(x, w) = \sigma \left(\sum_{i=1}^n x_i \cdot w_i - w_0 \right) = \sigma(\langle x, w \rangle)$$

x_i - ВХОД

w_i - ВЕС СВЯЗИ

σ - функция активации

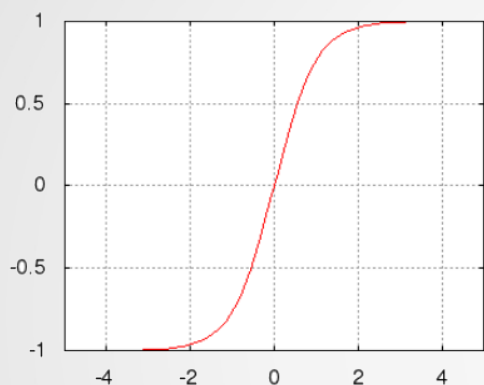


состояние нейрона

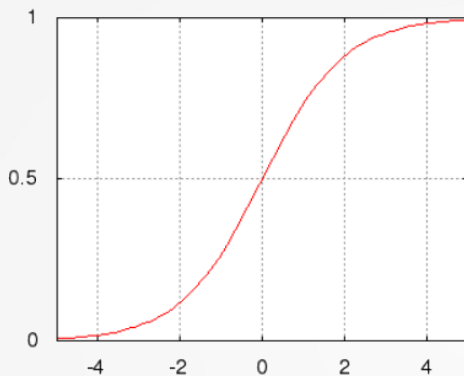
$$s(x, w) = \sum_{i=1}^n x_i \cdot w_i - w_0$$

рекуррентные нейронные сети

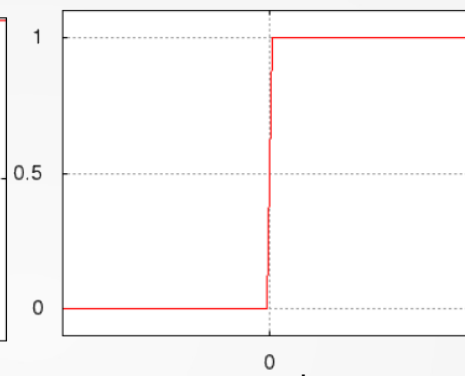
примеры функций активации



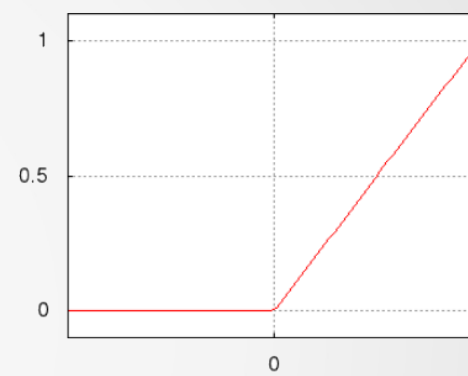
биполярный сигмоид



экспоненциальный сигмоид



пороговая функция



кусочно-линейная ReLU

softmax (экспоненциальная нормализация)
выходного слоя

$$(y_1, \dots, y_m) = \text{softmax}(s_1, \dots, s_m) = \frac{\exp(s)}{\sum_j \exp(s_j)}$$

стохастическая, выход нейрона
с вероятностью p равен 1
и $(1-p)$ равен 0

$$p = \frac{1}{1 + \exp(-s)}$$

рекуррентные нейронные сети

коннекционизм -

модель ИИ из связанных между собой простых элементов

многослойная сеть прямого распространения

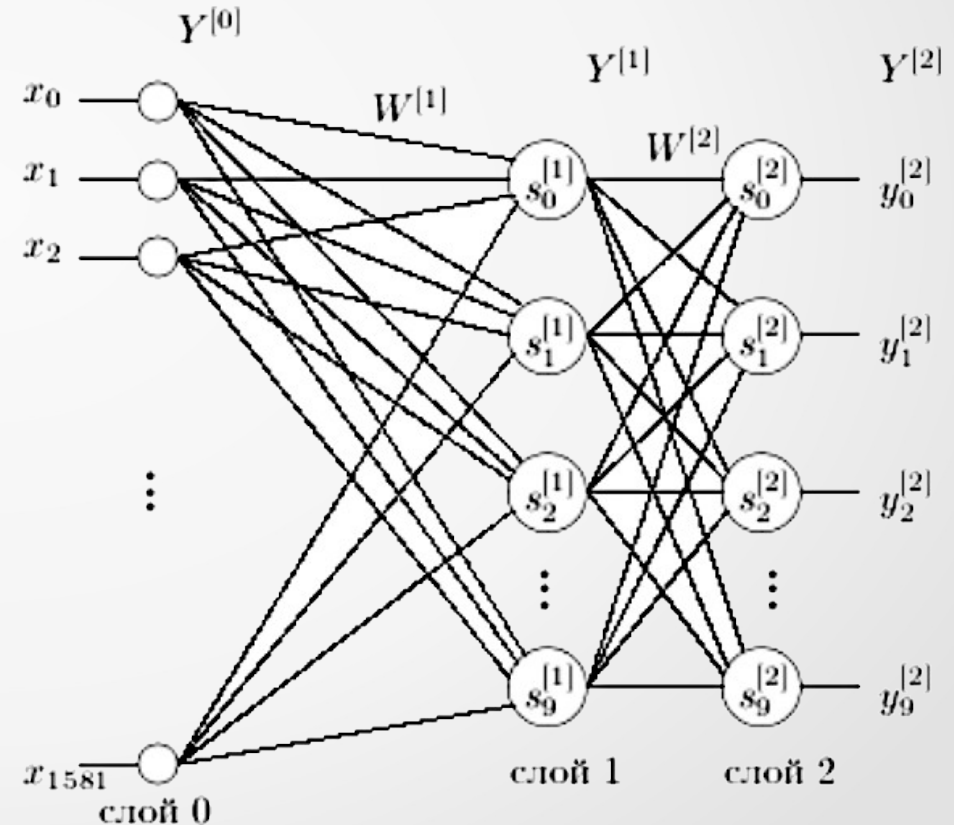
нейроны объединены в слои

сигнал распространяется послойно

входной распределительный слой

обрабатывающие скрытые слои

обрабатывающий выходной слой



рекуррентные нейронные сети

обучение многослойных сетей

$h: X \times W \rightarrow Y$ классификатор (X вход, W параметры, Y ответ)

$E: Y \times C \rightarrow \mathbb{R}$ функция потерь (Y ответ, C класс)

обучение классификатора как задача оптимизации

$$E(h(X, W), C) \rightarrow \min_W$$

примеры функций потерь

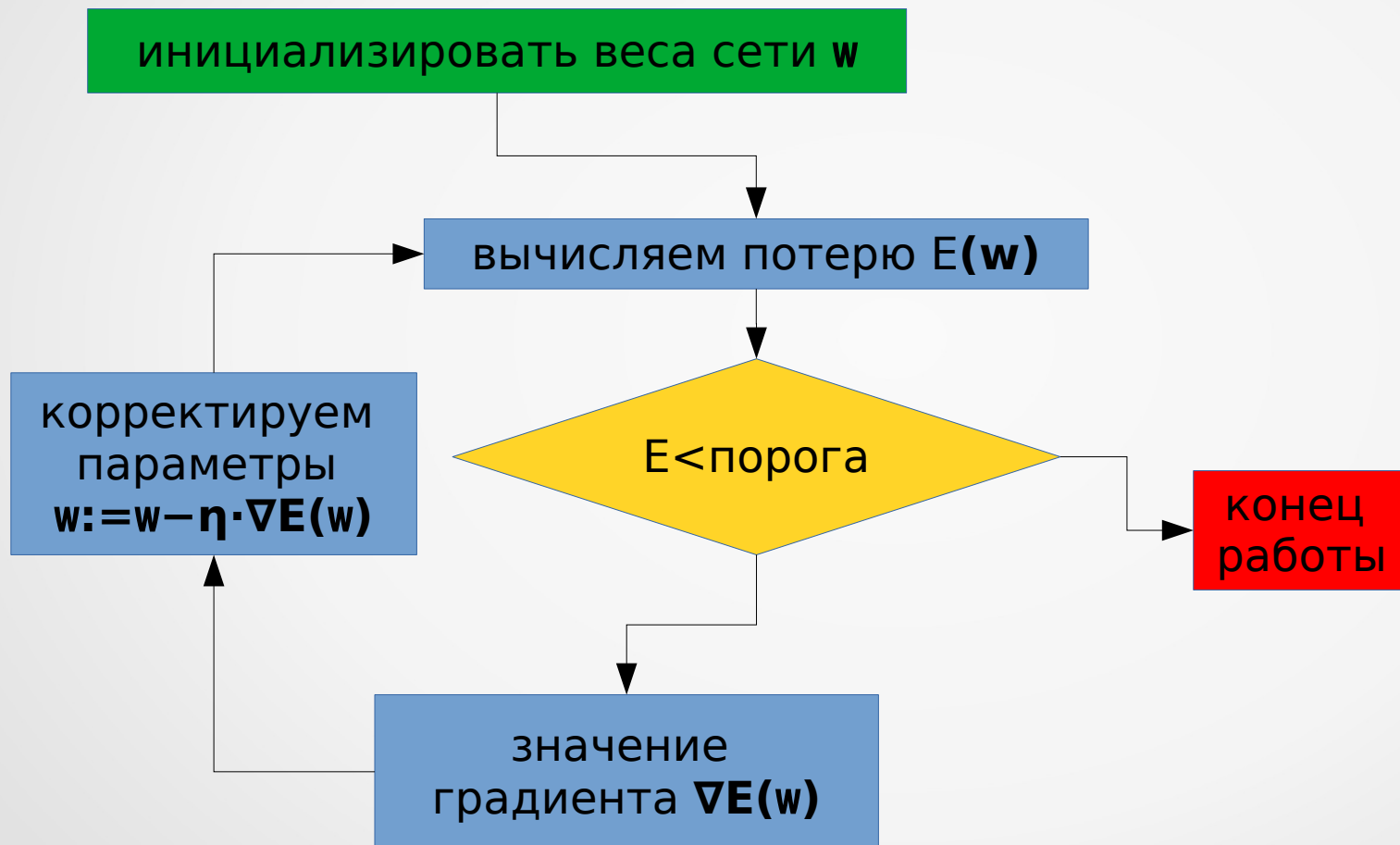
MSQE среднее квадратичное отклонение

Кросс-энтропия

Расстояние Кульбака-Лейблера

рекуррентные нейронные сети

градиентный спуск (GD)



рекуррентные нейронные сети

стратегии обучения

full batch - на каждой итерации используем все примеры

stochastic - на каждой итерации используем один случайный пример

mini batch - на каждой итерации используем случайное подмножество примеров

рекуррентные нейронные сети

метод обратного распространения ошибки

вычисление градиента функции потерь
для многослойной нейросети

$$\nabla E(W) = \left[\frac{\partial E}{\partial w_1}, \dots, \frac{\partial E}{\partial w_k} \right]$$

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial y_j} \frac{\partial y_j}{\partial s_j} \frac{\partial s_j}{\partial w_{ij}} \quad \text{градиент функции потерь для ИНС}$$

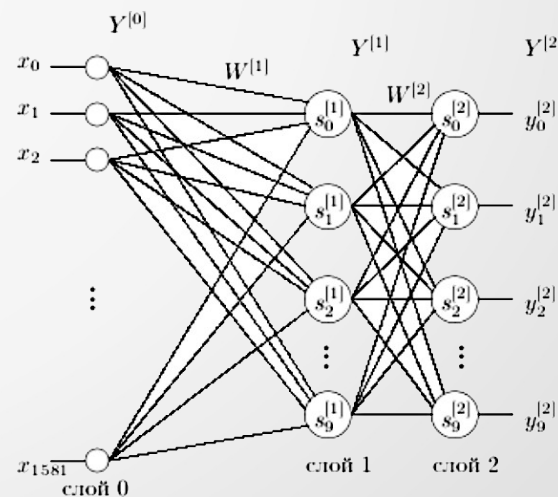
$$\frac{\partial s_j}{\partial w_{ij}} \quad \text{выход } i\text{-того нейрона предыдущего слоя (определен явно)}$$

$$\frac{\partial y_j}{\partial s_j} \quad \text{производная активационной функции (можем вычислить)}$$

$$\frac{\partial E_j}{\partial y_j} \quad \text{ошибка нейрона номер } j \text{ (определена для выходного слоя)}$$

$$\delta_i := \frac{\partial E}{\partial y_i} \quad \text{ошибка нейрона номер } j \text{ для выходного слоя}$$

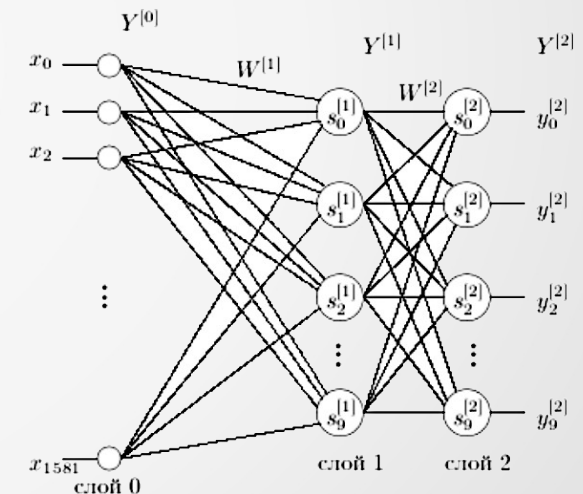
$$\delta_i := \frac{\partial y_i}{\partial s_i} \cdot \sum_j \delta_j w_{ij} \quad \text{ошибка нейрона номер } j \text{ для скрытого слоя}$$



рекуррентные нейронные сети

метод обратного распространения ошибки backProp

1. прямой проход:
вычислить состояния нейронов s для всех слоёв и выход сети y
2. вычисляем значения ошибки выходного слоя $\delta := \partial E / \partial y$
3. обратный проход:
последовательно от конца к началу
вычисляем δ для всех скрытых слоёв
4. для каждого слоя вычисляем значение градиента
 $\nabla E = \partial E / \partial w = y \cdot \delta^T$

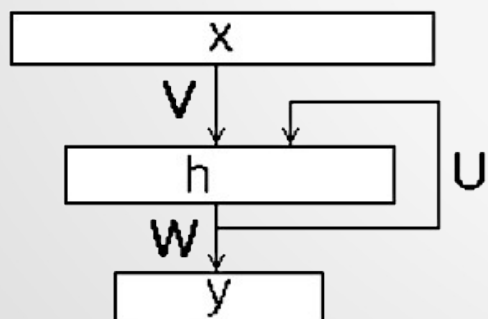
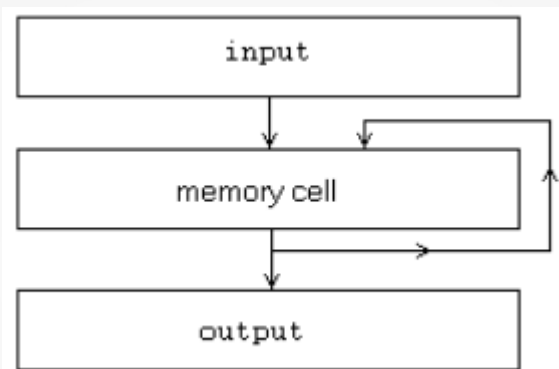


рекуррентные нейронные сети

рекуррентные нейросети

последовательности примеров

сеть Элмана



$$h(t) = f(V \cdot x(t) + U \cdot h(t-1) + b_h)$$

$$y(t) = g(W \cdot h(t) + b_y)$$

рекуррентные нейронные сети

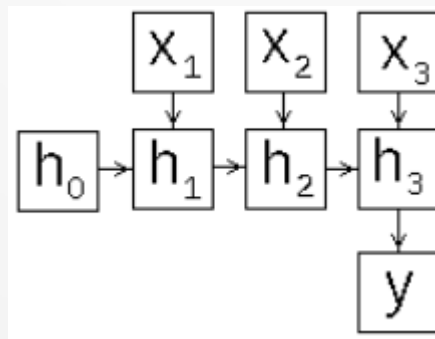
рекуррентные нейросети

последовательности примеров

способы организации работы рекуррентной сети

1. "много в один" (many-to-one) - скрытый слой последовательно изменяет своё состояние, из его конечного состояния вычисляется выход сети,

эту схему можно использовать для классификации текстов



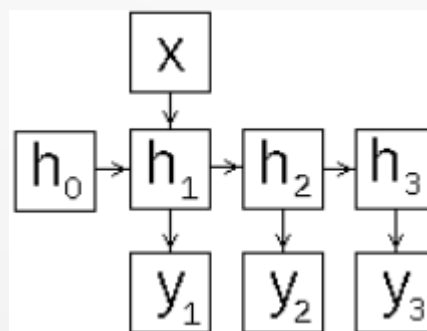
рекуррентные нейронные сети

рекуррентные нейросети

последовательности примеров

способы организации работы рекуррентной сети

2. "один во много" (one-to-many) - скрытый слой инициализируется одним входом, из цепочки его последующих состояний генерируются выходы сети, эту схему можно использовать для аннотирования изображений

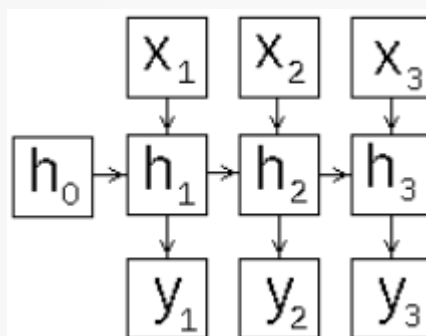


рекуррентные нейронные сети

рекуррентные нейросети

последовательности примеров

способы организации работы рекуррентной сети



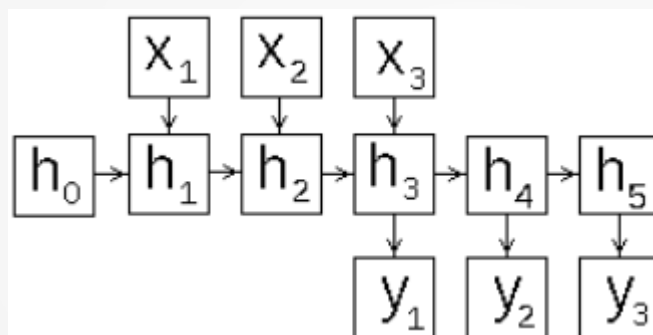
3. "много во много" (many-to-many sync.) - на каждый вход сеть выдаёт выход, который зависит от предыдущих входов, эту схему можно использовать для классификации видео

рекуррентные нейронные сети

рекуррентные нейросети

последовательности примеров

способы организации работы рекуррентной сети



4. "много во много" (many-to-many) - скрытый слой последовательно изменяет своё состояние, его конечное состояние служит инициализацией для выдачи цепочки результатов,

эту схему можно использовать для создания систем машинного перевода и чат-ботов

рекуррентные нейронные сети

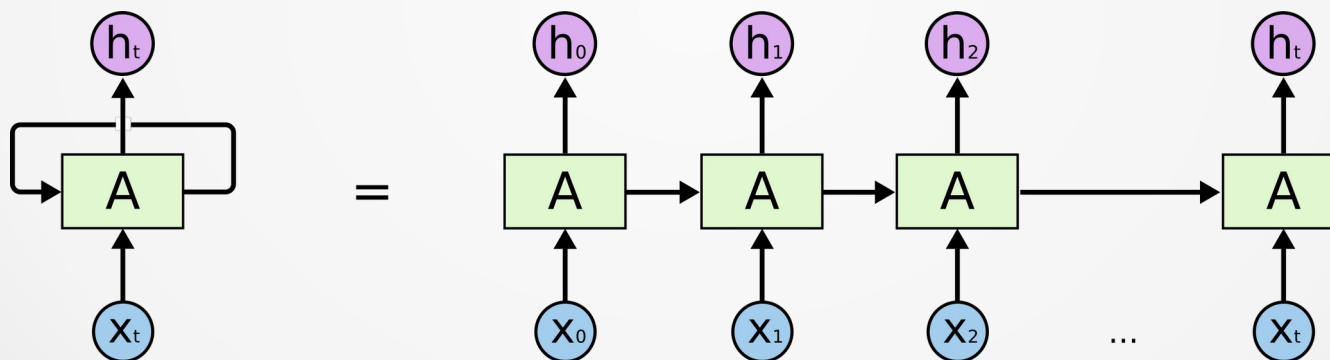
способ обучения

backpropagation through time - BPTT

(метод обратного распространения с разворачиванием сети во времени)

идея: развернуть последовательность

«превращаем» рекуррентную сеть в «обычную»



1. прямой проход - вычисляем состояния слоёв
2. обратный проход - вычисляем ошибку слоёв
3. вычисляем изменения весов

рекуррентные нейронные сети

backpropagation through time - BPTT

1. прямой проход - вычисляем состояния слоёв
для каждого вектора последовательности $\{x(1), \dots, x(n)\}$
вычисляем состояния скрытого слоя $\{s(1), \dots, s(n)\}$
выходы скрытого слоя $\{h(1), \dots, h(n)\}$
выход сети $\{y(1), \dots, y(n)\}$

$$s(t) = V \cdot x(t) + U \cdot h(t-1) + a$$

$$h(t) = f(s(t))$$

$$y(n) = g(W \cdot h(n) + b)$$

рекуррентные нейронные сети

backpropagation through time - BPTT

1. прямой проход - вычисляем состояния слоёв

для каждого вектора последовательности $\{x(1), \dots, x(n)\}$

вычисляем состояния скрытого слоя $\{s(1), \dots, s(n)\}$

выходы скрытого слоя $\{h(1), \dots, h(n)\}$

выход сети $\{y(1), \dots, y(n)\}$

$$s(t) = V \cdot x(t) + U \cdot h(t-1) + a$$

$$h(t) = f(s(t))$$

$$y(n) = g(W \cdot h(n) + b)$$

2. обратный проход - вычисляем ошибку слоёв

вычисляем ошибку выходного слоя δ_o

вычисляем ошибку скрытого слоя в конечном состоянии $\delta h(n)$

вычисляем ошибки скрытого слоя

в промежуточных состояниях $\delta h(t)$ ($t=1, \dots, n$)

$$\delta_o = y - d$$

$$\delta h(n) = W^T \cdot \delta_o \odot f'(s(n))$$

$$\delta h(t) = U^T \cdot \delta h(t+1) \odot f'(s(n))$$

рекуррентные нейронные сети

backpropagation through time - BPTT

1. прямой проход - вычисляем состояния слоёв
для каждого вектора последовательности $\{x(1), \dots, x(n)\}$
вычисляем состояния скрытого слоя $\{s(1), \dots, s(n)\}$
выходы скрытого слоя $\{h(1), \dots, h(n)\}$
выход сети $\{y(1), \dots, y(n)\}$

2. обратный проход - вычисляем ошибку слоёв
вычисляем ошибку выходного слоя δ_o
вычисляем ошибку скрытого слоя в конечном состоянии $\delta h(n)$
вычисляем ошибки скрытого слоя
в промежуточных состояниях $\delta h(t)$ ($t=1, \dots, n$)

3. вычисляем изменения весов

веса и сдвиг
выходного слоя

$$\Delta W = \delta_o \cdot (h(n))^T$$

$$\Delta b_y = \sum \delta_o$$

веса скрытого слоя

$$\Delta V = \sum_t \delta_h(t) \cdot (x(t))^T$$

$$\Delta U = \sum_t \delta_h(t) \cdot (h(t-1))^T$$

сдвиг скрытого слоя

$$\Delta b_h = \sum_t \delta_h(t)$$

$$s(t) = V \cdot x(t) + U \cdot h(t-1) + a$$

$$h(t) = f(s(t))$$

$$y(n) = g(W \cdot h(n) + b)$$

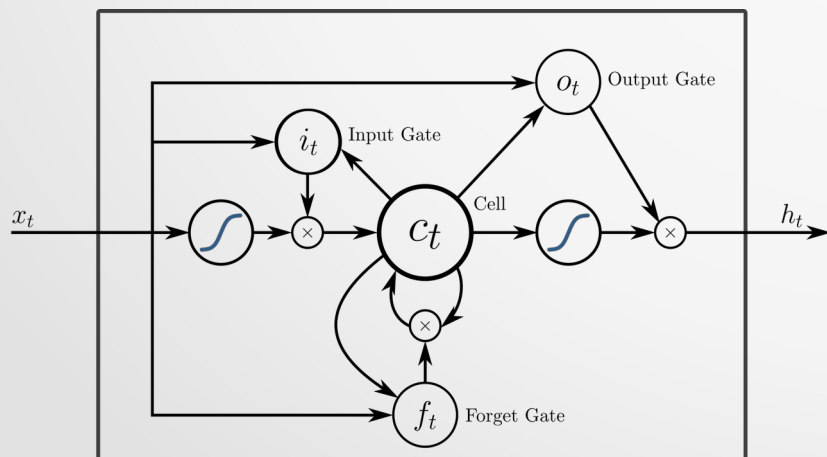
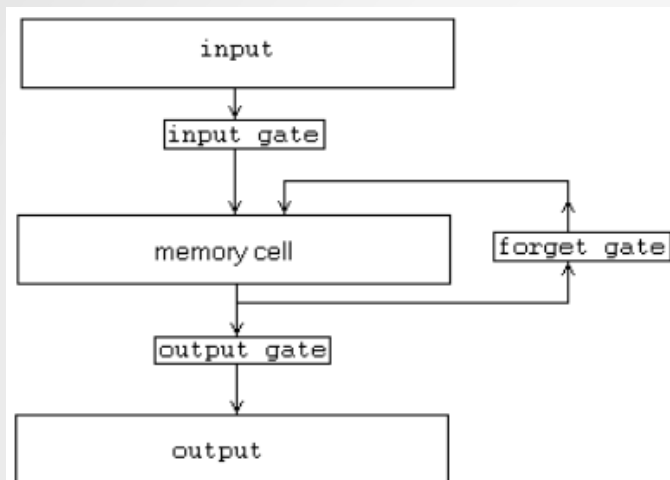
$$\delta_o = y - d$$

$$\delta_h(n) = W^T \cdot \delta_o \odot f'(s(n))$$

$$\delta_h(t) = U^T \cdot \delta_h(t+1) \odot f'(s(n))$$

рекуррентные нейронные сети

нейросеть LSTM



имеет дополнительные элементы, называемые гейтами (gate), которые должны управлять потоками данных.

В зависимости от своего состояния гейт может пропускать сигнал или не пропускать.

Вход сети (input)

Выход сети (output)

Память или состояние сети (memory cell)

Блок очистки памяти (forget gate).

Блок обновления памяти (input gate).

Блок выдачи результата (output gate).

рекуррентные нейронные сети

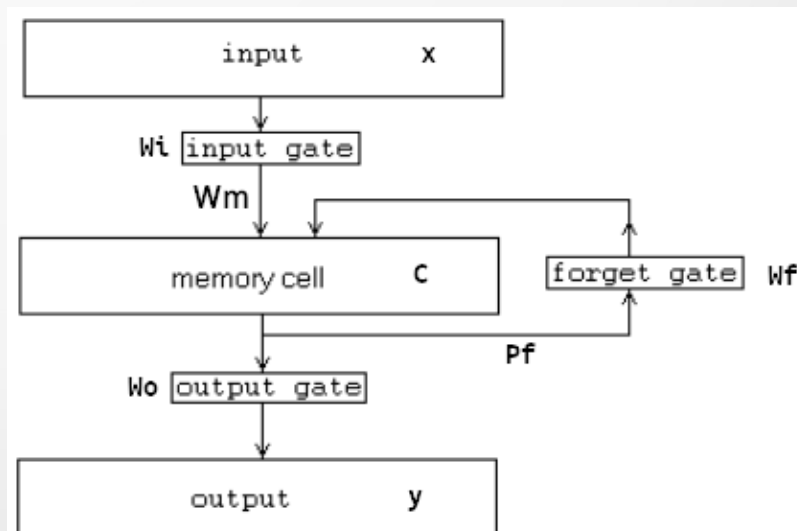
нейросеть LSTM (Long Short-Term Memory)

- 1.устанавливаем начальное состояние выхода сети y состояние памяти C
- 2.подать на вход сети очередной сигнал x из последовательности входов X
- 3.рассчитываем состояние входного гейта $g_i = \sigma(W_i \cdot x + R_i \cdot y + P_i \cdot C + b_i)$
- 4.рассчитываем состояние гейта памяти $g_f = \sigma(W_f \cdot x + R_f \cdot y + P_f \cdot C + b_f)$
- 5.рассчитываем изменение памяти $Z = \tanh(W_m \cdot x + R_m \cdot y + b_m)$
- 6.обновляем состояние памяти сети $c(t+1) = Z * g_i + C(t) * g_f$
- 7.рассчитываем состояние выходного гейта $g_o = \sigma(W_o \cdot x + R_o \cdot y + P_o \cdot C + b_o)$
- 8.рассчитываем выход сети $y = \tanh(C) * g_o$

LSTM with peephole

гейты имеют связь с памятью сети

(веса P)



рекуррентные нейронные сети

git clone https://github.com/mechanoid5/ml_lectorium.git

Jeffrey L. Elman Finding Structure in Time // COGNITIVE SCIENCE 14, 179-211 (1990)

Sepp Hochreiter, Jürgen Schmidhuber Long short-term memory // Neural Computation. 9(8) 1997

Евгений Борисов О рекуррентных нейронных сетях
<http://mechanoid.su/neural-net-rnn.html>

Евгений Борисов Рекуррентная сеть LSTM
<http://mechanoid.su/neural-net-lstm.html>

Радослав Нейчев Машинное обучение 10. Recurrent Neural Networks and Language Models // Лекторий ФПМИ
https://www.youtube.com/playlist?list=PL4_hYwCyhAvZyW6qS58x4uElZgAkMVUvj