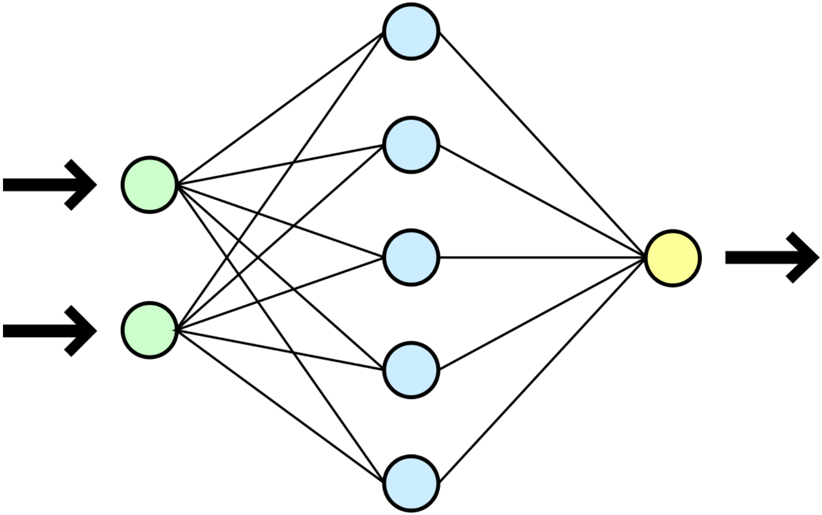
Реккурентные нейронные сети

Мадуар Дарин

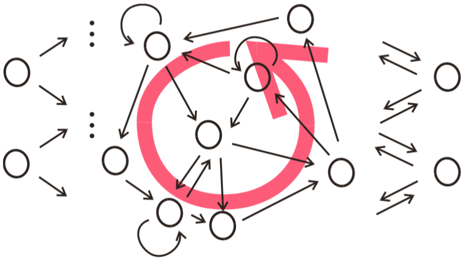
НИУ ВШЭ

27 ноября, 2018

Сети прямого распространения



Реккурентная нейронная сеть (RNN)



Реккурентная нейронная сеть – вид нейронных сетей, где связи между элементами образуют направленную последовательность.

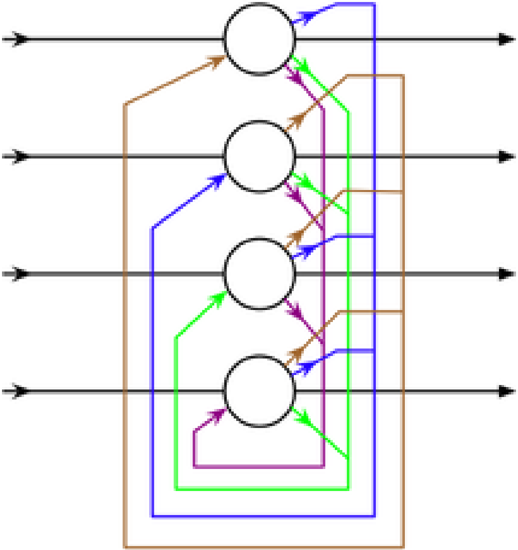
- RNN моделируютдинамическую систему
- все биологические сети – рекуррентные
- используются для анализа временных рядов и последовательностей, в которых важен порядок

Универсальная теорема аппросимации

- MLP аппроксимирует любую функцию
- RNN аппроксимирует любую динамическую систему
- все машины Тьюринга могут быть смоделированы полносвязной RNN

То есть, если сеть прямого распространения аппроксимируют функции, то рекуррентные нейросети аппроксимируют программы

Нейросеть Хопфилда



Нейросеть Хопфилда

- впервые упомянута в 1974 году, окончательно оформилась в 1982 г.
- реализовала ячейку ассоциативной памяти
- пороговая функция активации
- работает с последовательностями фиксированного размера

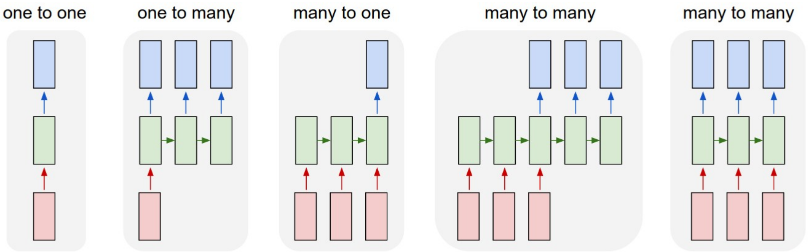
Моделирование последовательностей

- преобразование последовательностей одной природы в последовательности другой природы
  - графемы в фонемы
  - картинки в предложения
- предсказание следующего члена последовательности
  - прогнозирование следующего пикселя
  - предсказание кадра видео на основе предыдущих
  - генерация следующего слова

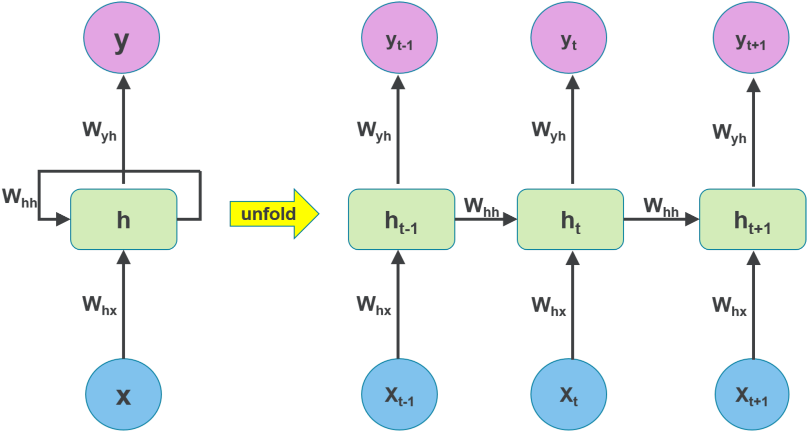
Как решать?

- авторегрессионная модель
  - модель скользящего среднего
- моделирование временных рядов нереккурентными нейронными сетями
- скрытые модели Маркова
  - есть «видимые» и «скрытые» состояния
  - в 1986 г. Джеффри Хинтон назвал слои «скрытыми»

Последовательности



Развертка сети во времени



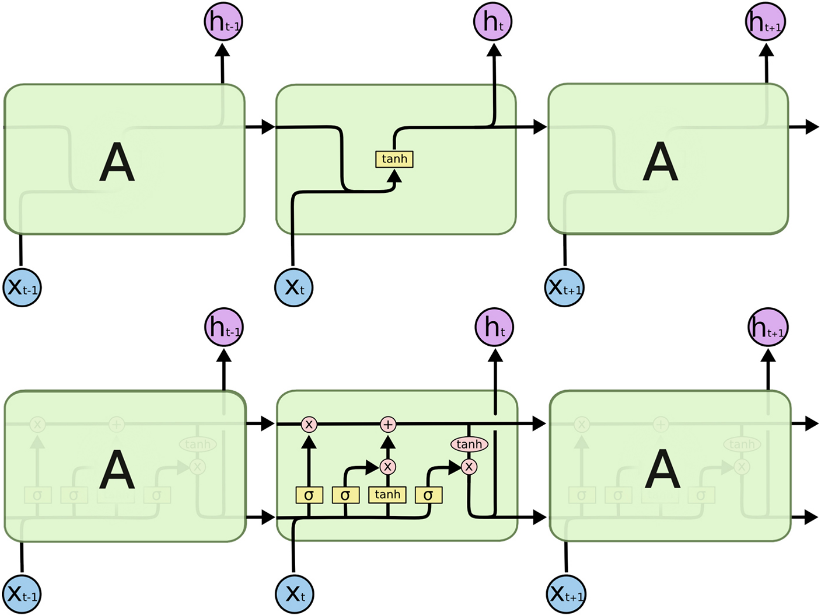
Обучение нейронных сетей

- backpropagation не работает для сетей с циклами
- существует модификация *backpropagation throught time*
- *обновление градиента на каждом шаге одинаковое*
- *алгоритм легко модифицируется так, чтобы можно было наложить любые линейные ограничения на веса.*
- *например, чтобы* $w_1 = w_2$*:*
  - $w_1 = w_2 \Rightarrow \Delta w_1 = \Delta w_2 \Rightarrow \dfrac{\partial E}{\partial w_1} = \dfrac{\partial E}{\partial w_2}$
  - $\Delta w_1^{new} = \Delta w_2^{new} = \dfrac{\partial E}{\partial w_1} + \dfrac{\partial E}{\partial w_2}$

Память в RNN

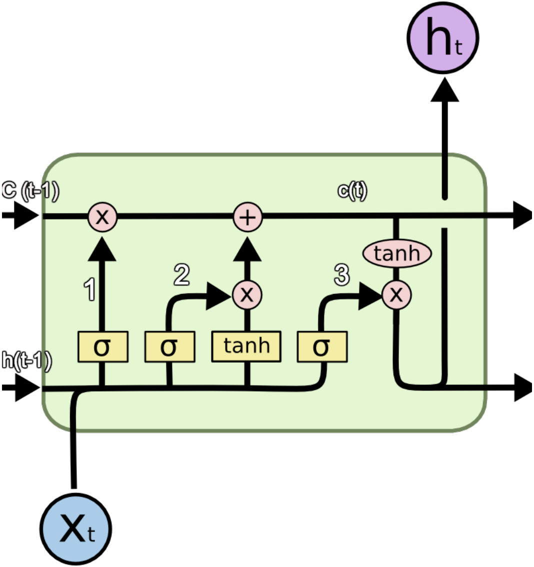
- *short-term memory: появляется в процессе прохождения сигнала по реккурентным слоям*
- *long-term memory: веса в процессе всего обучения меняются, кодируя таким образом «окружение»*
- **Долгая краткосрочная память** *(Long short-term memory; LSTM) – специальная архитектура рекуррентных нейронных сетей, предложенная в 1997 году.*
- *LSTM – промежуточный способ памяти*
- *LSTM – способ борьбы со взрывом и затуханием градиента*

LSTM

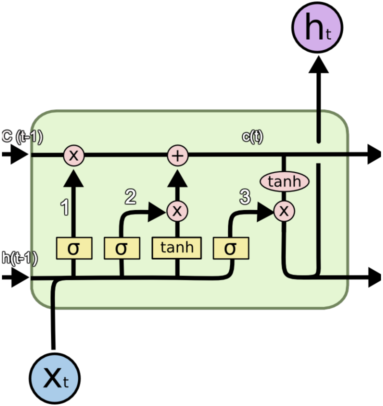


ht-1
ht
ht+1
A
tanh
A
Xt-1
Xt
Xt+1
ht-1
ht
ht+1
A
σ
σ
tanh
σ
tanh
A
Xt-1
Xt
Xt+1

LSTM

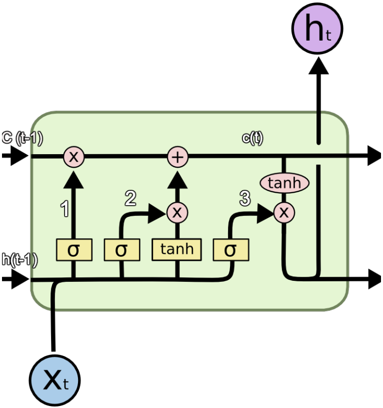


LSTM



Важная составляющая LSTM – *слой состояния сети* C_t*: сеть может как добавлять новую информацию, так и стирать старую*

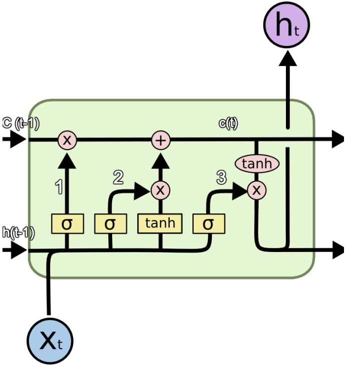
LSTM



1. *Forget gate layer* f_t – *слой, с помощью которого сигмоидальная функция смотрит на* X_t *и* h_t *и выдаёт для каждого числа в* C_{t-1} *число от 0 до 1 (вероятность забывания).*

$$f_t = \sigma(W_f[h_t, x_t] + b_f)$$

LSTM



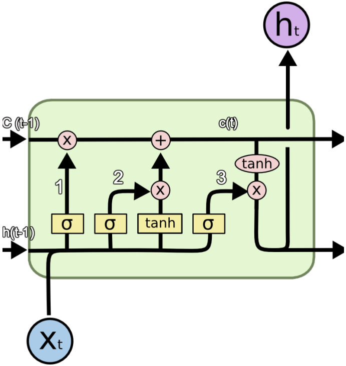
2. *Input gate layer* i_t – *входной слой, который решает какие веса обновлять.*

$$i_t = \sigma(W_i[h_t, x_t] + b_i)$$

Далее с помощью tanh вычисляются значения-кандидаты новых состояний

$$\tilde{C}_t = tanh(W_C[h_t, x_t] + b_C)$$

LSTM



3. Затем вектор состояний обновляется

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

И решаем, что выводить на данном шаге

$$o_t = sigma(W_o[h_t, x_t] + b_o)$$

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

Заключение

Примеры применения RNN:

- Paul Graham generator
- Wikipedia
- Algebraic Geometry (Latex)
- Linux Source Code
- Generating Baby Names *(Rudi Levette Berice Lussa Hany)*

Заключение

Proof. Omitted. □

Lemma 0.1. *Let $\mathcal{C}$ be a set of the construction.*

Let $\mathcal{C}$ be a gerber covering. Let $\mathcal{F}$ be a quasi-coherent sheaves of $\mathcal{O}$-modules. We have to show that

$$\mathcal{O}_{\mathcal{O}_X} = \mathcal{O}_X(\mathcal{L})$$

.

Proof. This is an algebraic space with the composition of sheaves $\mathcal{F}$ on $X_{\acute{e}tale}$ we have

$$\mathcal{O}_X(\mathcal{F}) = \{morph_1 \times_{\mathcal{O}_X} (\mathcal{G}, \mathcal{F})\}$$

where $\mathcal{G}$ defines an isomorphism $\mathcal{F} \to \mathcal{F}$ of $\mathcal{O}$-modules. □

Lemma 0.2. *This is an integer $\mathcal{Z}$ is injective.*

Proof. See Spaces, Lemma ??. □

Lemma 0.3. *Let S be a scheme. Let X be a scheme and X is an affine open covering. Let $\mathcal{U} \subset \mathcal{X}$ be a canonical and locally of finite type. Let X be a scheme. Let X be a scheme which is equal to the formal complex.*

The following to the construction of the lemma follows.

Let X be a scheme. Let X be a scheme covering. Let

$$b : X \to Y' \to Y \to Y \to Y' \times_X Y \to X.$$

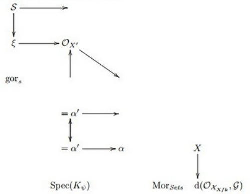
be a morphism of algebraic spaces over S and Y.

Proof. Let X be a nonzero scheme of X. Let X be an algebraic space. Let $\mathcal{F}$ be a quasi-coherent sheaf of $\mathcal{O}_X$-modules. The following are equivalent

(1) $\mathcal{F}$ is an algebraic space over S.
(2) If X is an affine open covering.

Consider a common structure on X and X the functor $\mathcal{O}_X(U)$ which is locally of finite type. □

This since $\mathcal{F} \in \mathcal{F}$ and $x \in \mathcal{G}$ the diagram



is a limit. Then $\mathcal{G}$ is a finite type and assume S is a flat and $\mathcal{F}$ and $\mathcal{G}$ is a finite type f_*. This is of finite type diagrams, and

- the composition of $\mathcal{G}$ is a regular sequence,
- $\mathcal{O}_{X'}$ is a sheaf of rings.

□

Proof. We have see that $X = \mathrm{Spec}(R)$ and $\mathcal{F}$ is a finite type representable by algebraic space. The property $\mathcal{F}$ is a finite morphism of algebraic stacks. Then the cohomology of X is an open neighbourhood of U. □

Proof. This is clear that $\mathcal{G}$ is a finite presentation, see Lemmas ??.
A *reduced above* we conclude that U is an open covering of $\mathcal{C}$. The functor $\mathcal{F}$ is a "field

$$\mathcal{O}_{X,x} \longrightarrow \mathcal{F}_{\overline{x}} \quad \text{-1}(\mathcal{O}_{X_{\acute{e}tale}}) \longrightarrow \mathcal{O}_{X_\ell}^{-1}\mathcal{O}_{X_\lambda}(\mathcal{O}_{X_\eta}^{\overline{v}})$$

is an isomorphism of covering of $\mathcal{O}_{X_i}$. If $\mathcal{F}$ is the unique element of $\mathcal{F}$ such that X is an isomorphism.
The property $\mathcal{F}$ is a disjoint union of Proposition ?? and we can filtered set of presentations of a scheme $\mathcal{O}_X$-algebra with $\mathcal{F}$ are opens of finite type over S.
If $\mathcal{F}$ is a scheme theoretic image points. □

If $\mathcal{F}$ is a finite direct sum $\mathcal{O}_{X_\lambda}$ is a closed immersion, see Lemma ??. This is a sequence of $\mathcal{F}$ is a similar morphism.

Заключение

Спасибо за внимание!

Ссылки

[1]. http://karpathy.github.io/2015/05/21/rnn-effectiveness/
[2]. https://cs224d.stanford.edu/lectures/CS224d-Lecture8.pdf
[3]. https://ru.wikipedia.org