Deep Learning. Анализ текстов

Урок 9

Егор Конягин

25 июля 2019 г.

МФТИ & АО ”ЦОСиВТ”

Содержание

Повторение

RNN

В модели RNN ответ, полученный на t-ом элементе x, передается далее

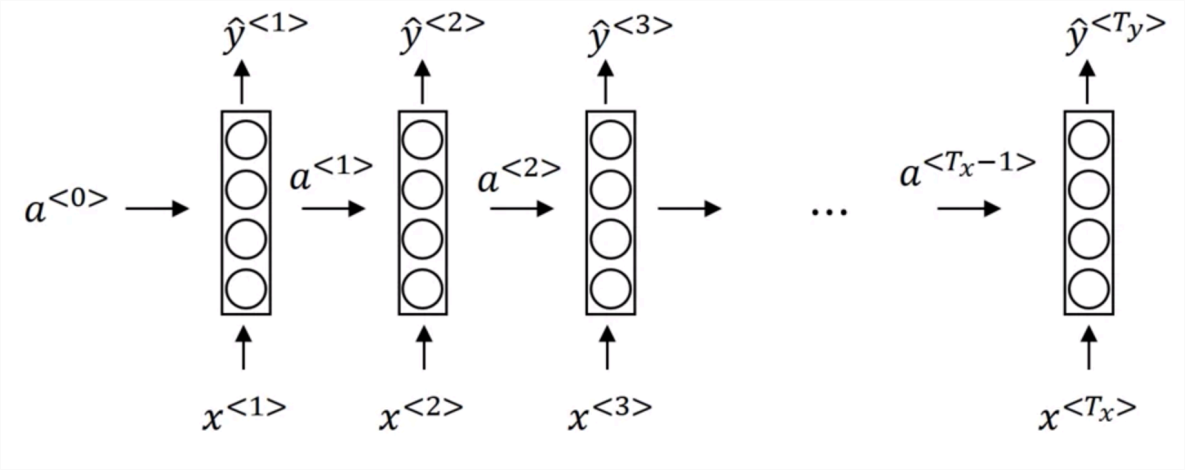

Рис. 1: Простейшая модель RNN. Источник: Andrew Ng

$$a^{<t>} = \sigma_1(w_{aa} \cdot a^{<t-1>} + w_{ax} \cdot x^{<t>} + b_a) \quad a^{<0>} = \overrightarrow{0} \tag{1}$$

$$y^{<t>} = \sigma_1(w_{ya} \cdot a^{<t-1>} + b_a) \tag{2}$$

GRU

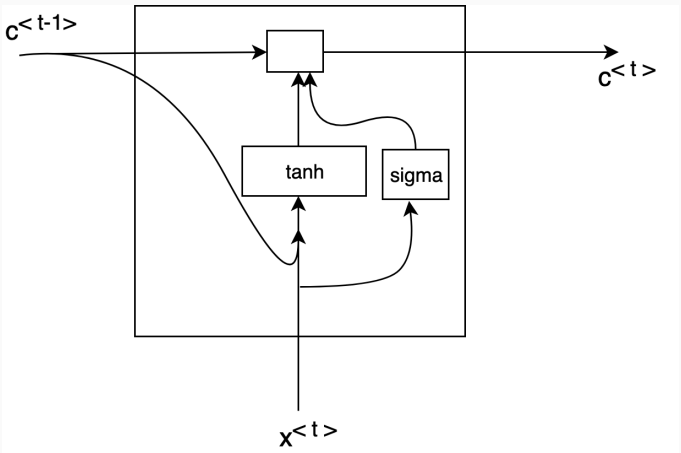

Рис. 2: GRU-ячейка

Обновление происходит только, если Γ примет соответствующее значение (затвор).

LSTM

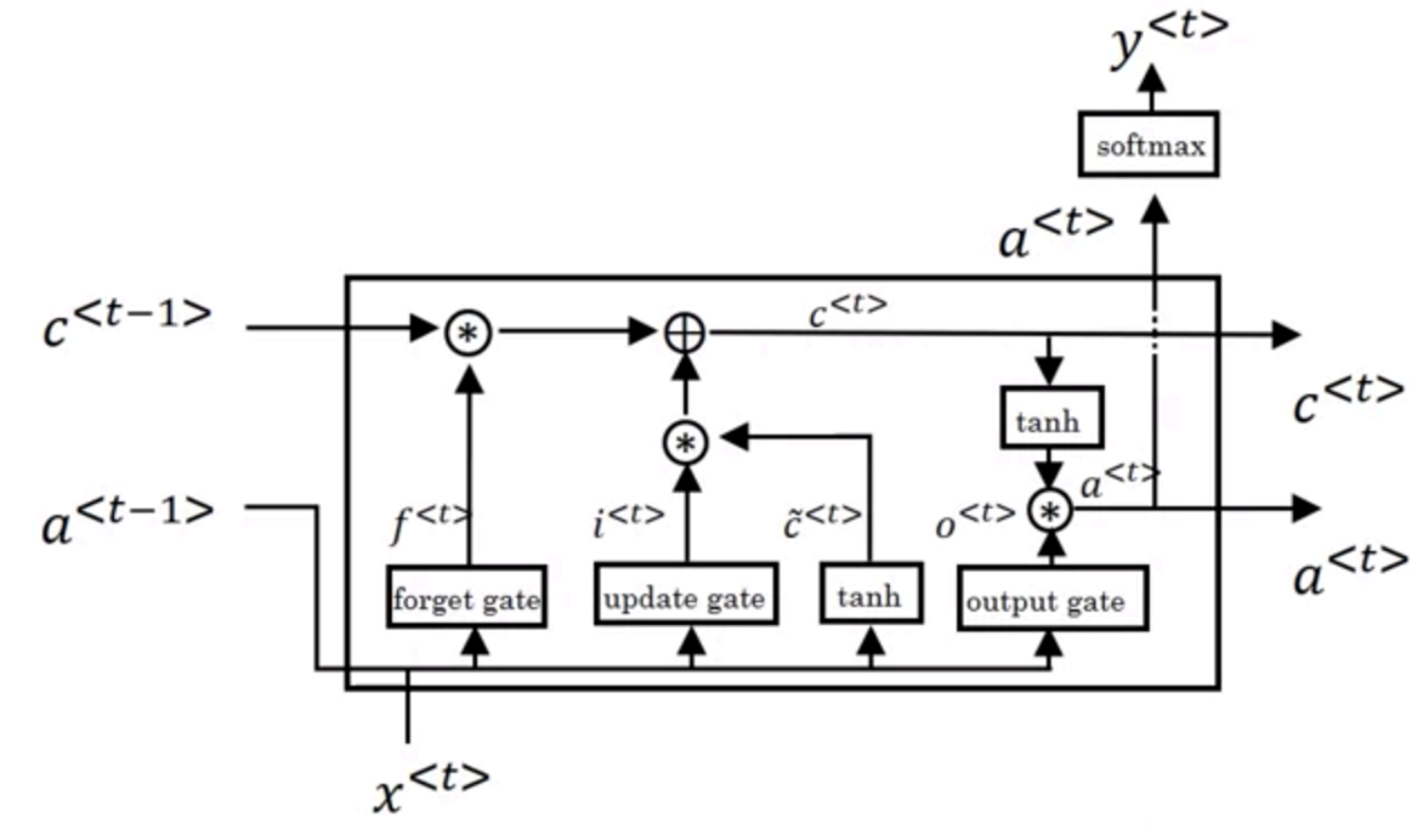

Рис. 3: LSTM-ячейка. Источник: Andrew Ng's classes

Векторизация текста (word embeddings)

Векторизация текста

Мы уже рассматривали one-hot векторизацию текста и отметили, что такая модель не является оптимальной. **Почему?**

1. векторы слов, по смыслу похожих, ничего общего не имеют;
2. разреженное представление увеличивает затратность ресурсов;

Решение: использовать координаты в векторном представлении как значения неких заданных признаков.

Построив такую модель языка, можно использовать обычное скалярное произведение для оценки похожести слов/текста...

t - SNE

Алгоритм t - SNE - это алгоритм понижения размерности для визуализации данных (как и PCA). Однако он, в отличие от PCA, является нелинейным алгоритмом и лучше позволяет визуализировать данные.

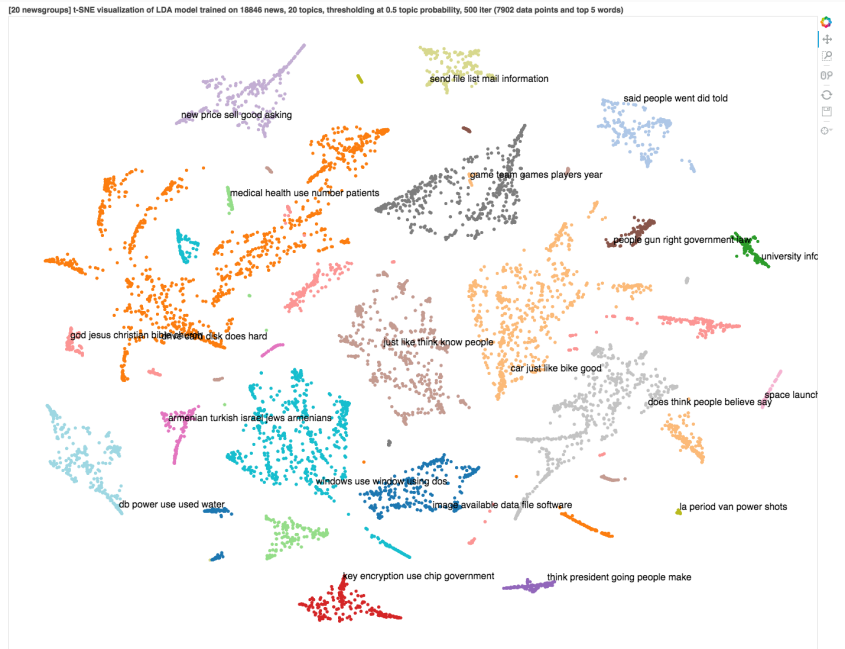

Рис. 4: Использование t-sne для отображения тем новостей. Источник: SHUAI'S AI & DATA BLOG

Word embeddings. Продолжение

Как получить векторное представление и дальше работать с ним?

1. Получить представления, обучаясь на больших объемах текста (миллиард слов).
2. Использовать эти представления для своей задачи с помощью дообучения на маленьком датасете (тысячи слов).

Word embeddings. Пример

Векторные представления способны понимать признаки, присущие слову. Рассмотрим пример:

$$\text{Мужчина} \to \text{женщина} = \text{король} \to \text{кто?}$$

Решение:

$$\overrightarrow{e}_{man} - \overrightarrow{e}_{woman} = \overrightarrow{r}; \tag{3}$$

$$\overrightarrow{e}_{king} - \overrightarrow{e}_{i} = \overrightarrow{r}_{i}; \tag{4}$$

Наша задача найти такой i', чтобы $\mathcal{D}(\overrightarrow{r}, \overrightarrow{r}_{i'}) \to \min$. **Как выбрать функцию $\mathcal{D}$?**

Обычно в анализе текста используют cosine similarity:

$$\mathcal{D}(a, b) = \frac{a^T b}{|a| \cdot |b|}. \tag{5}$$

Как получить адекватный word embedding?

Положим, мы составляем представления для 10 000 различных слов, а представления - это 300-мерные векторы. Тогда можно составить матрицу для перехода из one-hot представления к более адекватному (низкой размерности).

Эта матрица носит название embedding matrix.

Как получить адекватный word embedding?

Вероятностная модель

Построение вероятностной модели языка:

1. задать матрицу E (10000,300) случайным образом;
2. подать выход этой матрицы на вход полносвязной нейросети с softmax-выходом;
3. на вход алгоритму подавать m слов предложения. Задача алгоритма - предсказать следующее слово (то есть решать задачу многоклассовой классификации);
4. запустить обучение;
5. ЖДЕМ;
6. Готово!

Skip gram. Word2Vec

Как и в предыдущем случае, ставится задача обучения с учителем: x - это т.н. контекст, y - цель (одно слово).

В данном подходе контекст строится существенно проще:

1. выбирается случайное слово в предложении - это и будет контекстом;
2. в пределах окна фиксированной ширины (± 5 слов) выбирается случайное слово - это и будет целевое слово.

Недостатки:

- при использовании словаря большего размера softmax-слой становится огромным;
- случайно выбирая слово для контекста, мы наткнемся на наиболее используемые слова в языке (артикли, предлоги и тд).

Решение: использование иерархического классификатора, а затем softmax-слоя.

Negative sampling

Идея: сформулировать задачу как бинарную классификацию:

$$(e_i, e_j) \rightarrow classifier \rightarrow \mathcal{P}(e_i, e_j) \tag{6}$$

Используя данную модель, нам надо также в обучающую выборку добавить заведомо несочетающиеся слова (отсюда название negative sampling).

Как выбирать слова для сэмплинга? Ответ: используя частотный анализ (с небольшими модификациями).

$$\mathcal{P}(w_i) = \frac{f(w_i)}{\sum_j f(w_j)} \tag{7}$$

$$\mathcal{P}^*(w_i) = \frac{f(w_i)^{3/4}}{\sum_j f(w_j)^{3/4}} \tag{8}$$

Важно! На одно положительное словосочетание должно приходиться k негативных.

GloVe

Снова рассматриваем контекст C и целевое слово T.

В явном виде запишем величину X_{ij} - кол-во появлений i-ого слова в контексте j-ого слова.

В общем случае $X_{ij} \neq X_{ji}$.

Постановка задачи:

$$\sum_i \sum_j f(x_{ij})(\theta_i^T \cdot e_j - \log x_{ij})^2 \to \min_{\theta, e} \tag{9}$$

Summary