Практический анализ данных и машинное обучение: искусственные нейронные сети

Ульянкин Филипп

30 октября 2018 г.

W2V, рекурентные нейронные сетки

Agenda

- Как научить компьютер читать?
- w2v, embeddings
- Рекурентные нейронные сетки, работа с последовательностями

w2v

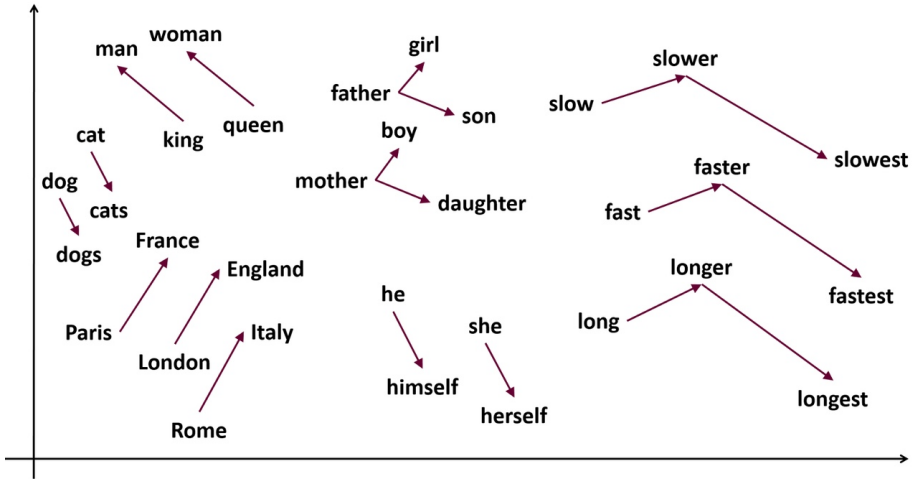
Небольшое введение в анализ текстов

- Тексты бывают очень разными
- Тексты нужно чистить от стоп-слов, мусора, нормализовывать
- Классический подход: мешок слов, предполагается, что порядок слов в тексте неважен, анализируется то насколько часто слово встречается внутри текста
- Нейросетки позволяет собрать кое-что покруче

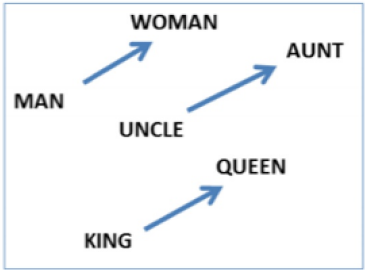
Words embeddings

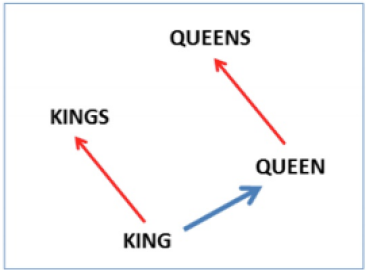
- **Наша цель:** Мы хотим понять смысл
- Давайте превратим наши слова в d-мерные вектора, эти вектора и называются embeddings

Words embeddings



Words embeddings





(Mikolov et al., NAACL HLT, 2013)

Word	Embedding
king	(0.188, 0.325, …, 0.708, 0.656)
man	(-0.471, 0.356, …, -0.806, 1.001)

Words embeddings

- Звучит как магия...
- Как будем подбирать такие вектора?!
- Начнём с рандомного embedding и будем его потихоньку менять градиентным спуском
- Как понять как правильно менять embedding?

Word2vec

- Предложена Томашем Миколовым и соавторами в 2013 году в двух разных вариантах
- CBOW (непрерывный мешок слов) --- по заданному контексту слова пытаемся предсказать слово
- Skip-gram --- по заданному слову пытаемся предсказать его контекст

Word2vec

Немного формализуем понятие контекст. Пусть вероятность встретить слово i в контексте слова j это

$$P(w_i \mid w_j) = \frac{e^{(w_i, w_j)}}{\sum_w e^{(w_i, w)}}$$

Под контекстом будем понимать k слов до рассматриваемого и k после него. Векторные представления будем настраивать так, что вероятность встретить слова из одного контекста рядом оказалась высокой. Для этого максимизируем правдоподобие:

$$\sum_{i=1}^{n} \sum_{j=-k}^{k} \log p(w_{i+j} \mid w_i) \to \max_w.$$

Word2vec Skip-gram

*To Sherlock Holmes **she** always was the woman.*

window ±2

Word2vec Skip-gram

To Sherlock Holmes ***she*** ***always*** *was the woman.*

window ±2

Word2vec Skip-gram

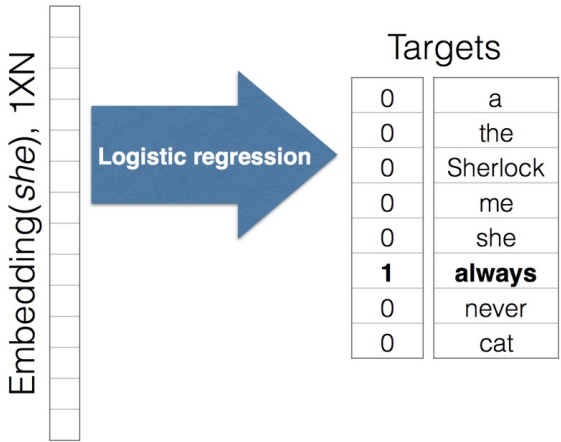
To Sherlock Holmes ***she*** ***always*** *was the woman.*

window ±2

Word2vec Skip-gram

*To Sherlock Holmes **she** **always** was the woman.*

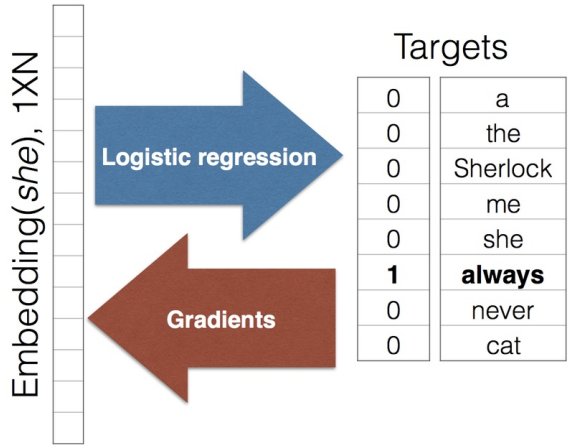
window ±2



Word2vec Skip-gram

*To Sherlock Holmes **she** **always** was the woman.*

window ±2



Word2Vec

- Трансформирует пространство текстов в d-мерное пространство векторов
- Между словами на выходе выполняются интересные арифметические свойства
- Для обучения требует много данных

Something2Vec

- Можно попробовать представить в виде embedding любую последовательность: текст, банковские транзакции, переход юзера по сайтам, граф связей и тд
- По полученным представлениям можно учить традиционные модели, например, линейные
- Более того, можно использовать embedding просто как внеочередной слой в нейросетке

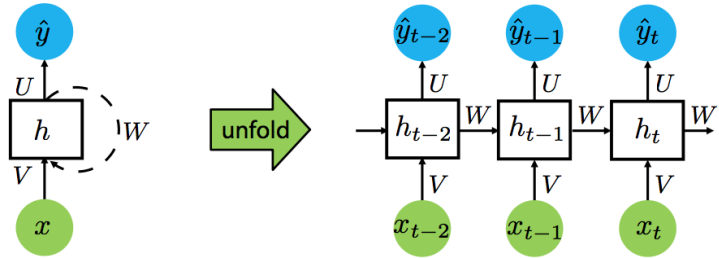
Учим свой W2V на русской Википедии

Рекурентные нейронные сети

Рекурентная сеть

- Каждый нейрон взаимодействует сам с собой
- На вход поступает последовательность (текст, видео, картинка, временной ряд), один и тот же нейрон просматривает её
- Впоследствие можно использовать эту сетку для генерации новых последовательностей (текстов, видео и тп)

Рекурентная сеть



$$h_t = f_h(Vx_t + Wh_{t-1} + b_h) \qquad \hat{y}_t = f_y(Uh_t + b_y)$$

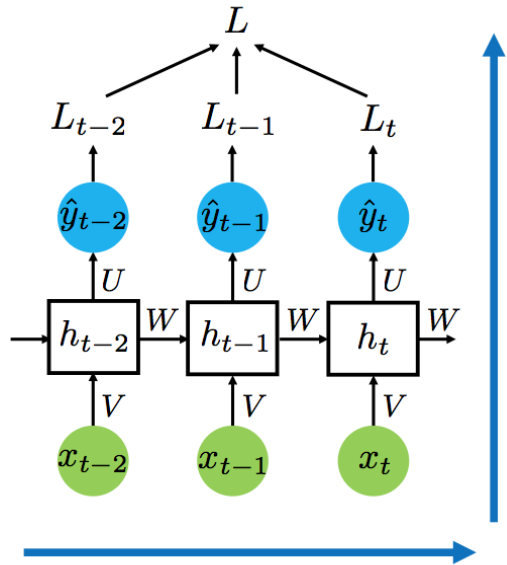
Рекурентная сеть

- **Проблема** состоит в том, что в работе сетки появилось новое измерение: время

- На каждом шаге сетка взвешивает свой предыдущий опыт и новую информацию, получается, что при обучении, мы должны брать производную назад во времени

Forward pass:

$$h_t,\ \hat{y}_t,\ L_t,\ L$$

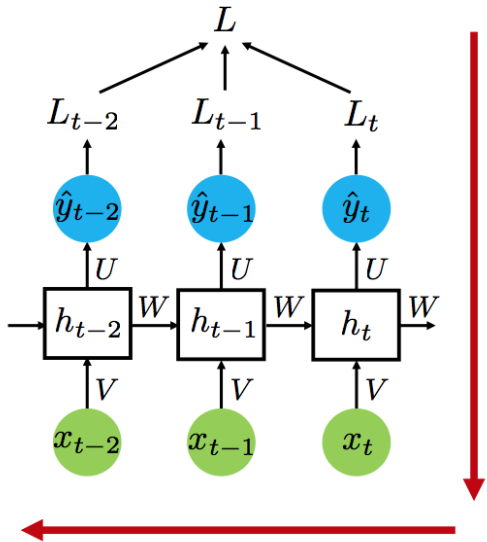


Forward pass:

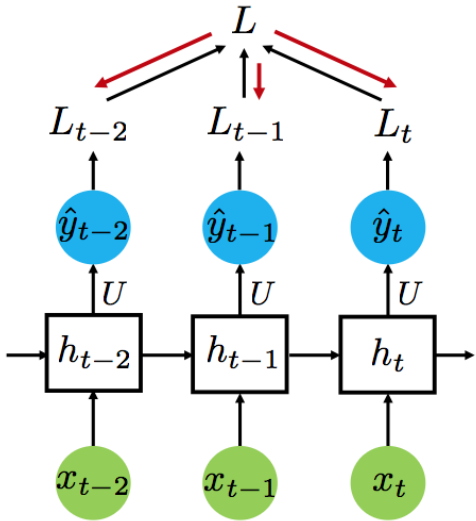
$$h_t,\ \hat{y}_t,\ L_t,\ L$$

Backward pass:

$$\frac{\partial L}{\partial U},\ \frac{\partial L}{\partial V},\ \frac{\partial L}{\partial W},$$
$$\frac{\partial L}{\partial b_x},\ \frac{\partial L}{\partial b_h}$$

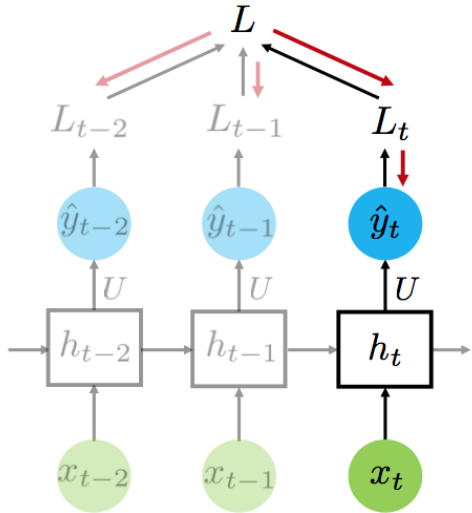


$$\frac{\partial L}{\partial U} = \sum_{i=0}^{T} \frac{\partial L_i}{\partial U}$$



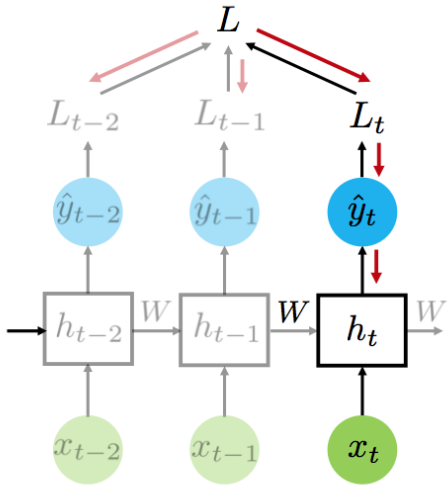
$$\frac{\partial L}{\partial U} = \sum_{i=0}^{T} \frac{\partial L_i}{\partial U}$$

$$\frac{\partial L_t}{\partial U} = \frac{\partial L_t}{\partial \hat{y}_t} \frac{\partial \hat{y}_t}{\partial U}$$



$$\frac{\partial L}{\partial W} = \sum_{i=0}^{T} \frac{\partial L_i}{\partial W}$$

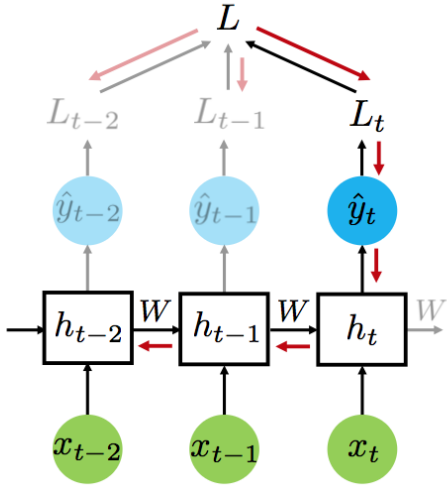
$$\frac{\partial L_t}{\partial W} = \frac{\partial L_t}{\partial \hat{y}_t} \frac{\partial \hat{y}_t}{\partial h_t} \frac{\partial h_t}{\partial W}$$



$$\frac{\partial L}{\partial W} = \sum_{i=0}^{T} \frac{\partial L_i}{\partial W}$$

$$\frac{\partial L_t}{\partial W} = \frac{\partial L_t}{\partial \hat{y}_t} \frac{\partial \hat{y}_t}{\partial h_t} \frac{\partial h_t}{\partial W}$$

$$h_t = f_h(V x_t + \boxed{W} h_{t-1} + b_h)$$



$$\frac{\partial L_t}{\partial W} = \frac{\partial L_t}{\partial \hat{y}_t} \frac{\partial \hat{y}_t}{\partial h_t} \boxed{\left(\frac{\partial h_t}{\partial W} + \frac{\partial h_t}{\partial h_{t-1}} \frac{\partial h_{t-1}}{\partial W} + \dots \right)}$$

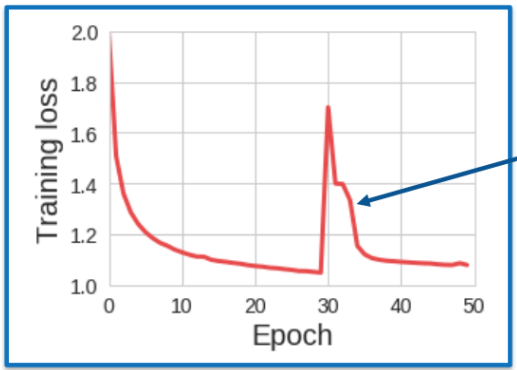
Vanishing and Exploding

$$\frac{\partial L_t}{\partial W} \propto \sum_{k=0}^{t} \left(\prod_{i=k+1}^{t} \frac{\partial h_i}{\partial h_{i-1}} \right) \frac{\partial h_k}{\partial W}$$

$$\left\| \frac{\partial h_i}{\partial h_{i-1}} \right\|_2 < 1 \quad \Rightarrow \quad \text{Vanishing gradients}$$

$$\left\| \frac{\partial h_i}{\partial h_{i-1}} \right\|_2 > 1 \quad \Rightarrow \quad \text{Exploding gradients}$$

Как понять, что градиент взорвался?

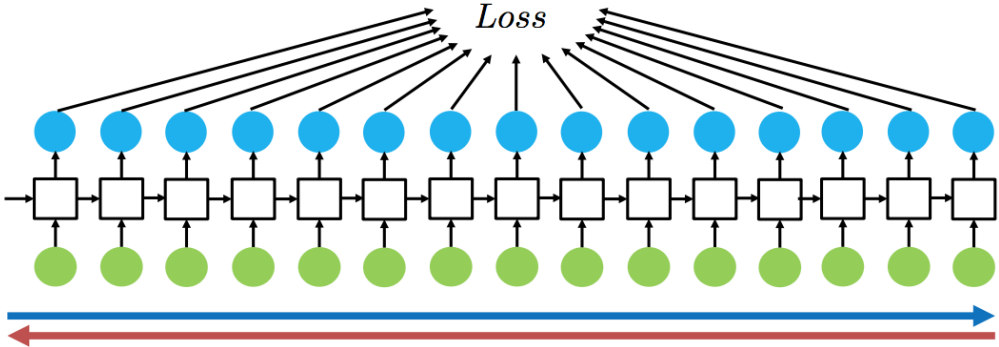


Взрыв

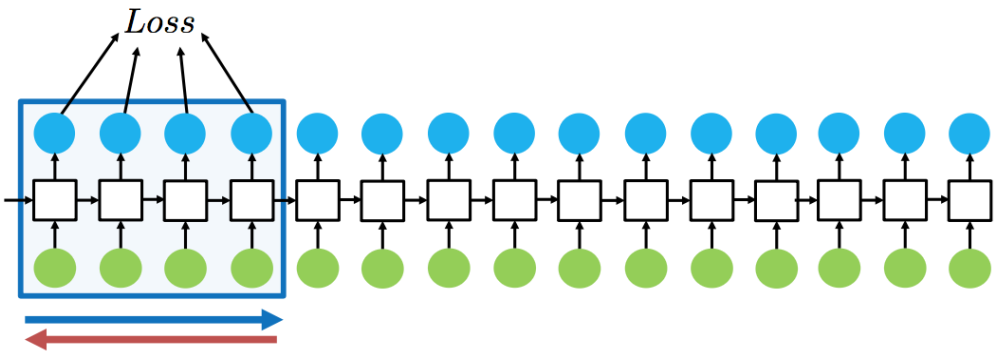
Как предотвратить взрыв?

- Поставить порог, которые не будет пробиваться градиентом
- Обучать сетку не целиком, а по кусочкам
- Аккуратно инициализировать веса
- Делать skip-connection
- Придумать специальную архитектуру

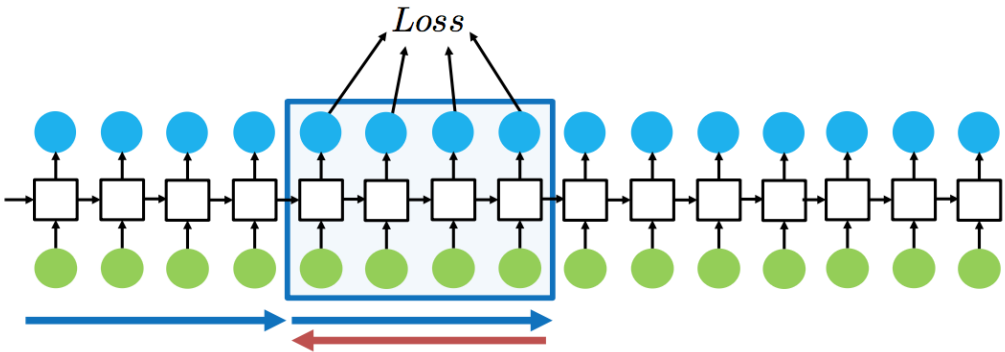
Урезанное обучение



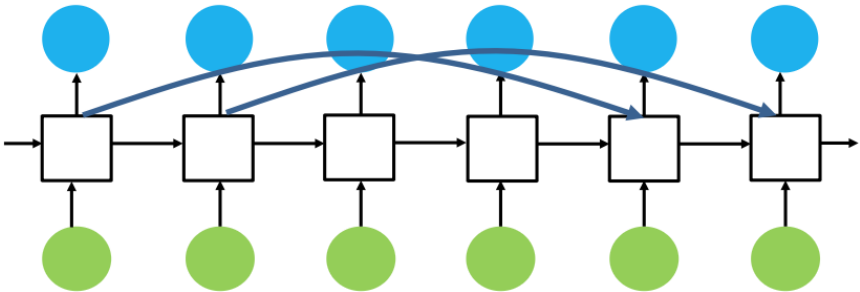
Урезанное обучение



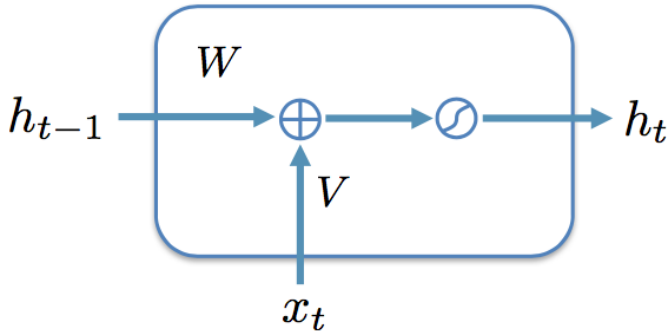
Урезанное обучение



Skip-connection

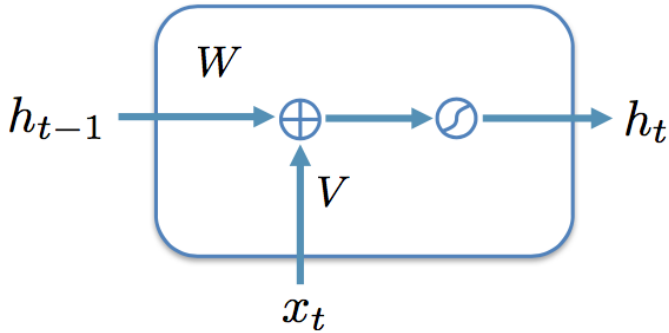


Простейшая RNN



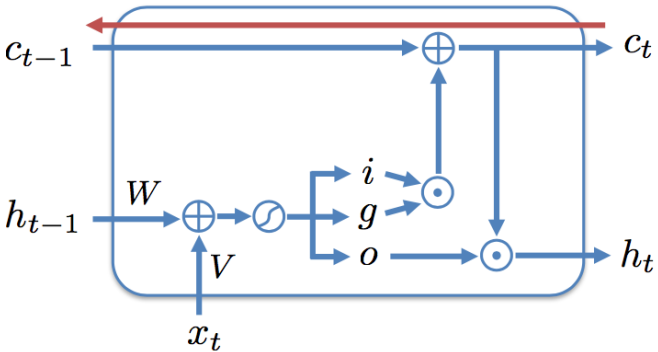
$$h_t = \tilde{f}(Vx_t + Wh_{t-1} + b_h)$$

Простейшая RNN



$$h_t = \tilde{f}(Vx_t + Wh_{t-1} + b_h)$$

LSTM



Мозг кипит! Давайте посмотрим как всё это легко запилить в Keras!