МИСИС
УНИВЕРСИТЕТ

Языковое моделирование

Корнет Мария Евгеньевна
старший преподаватель кафедры
Инженерная кибернетика

План

- Задачи языкового моделирования
- N граммные модели
- RNN моделирование

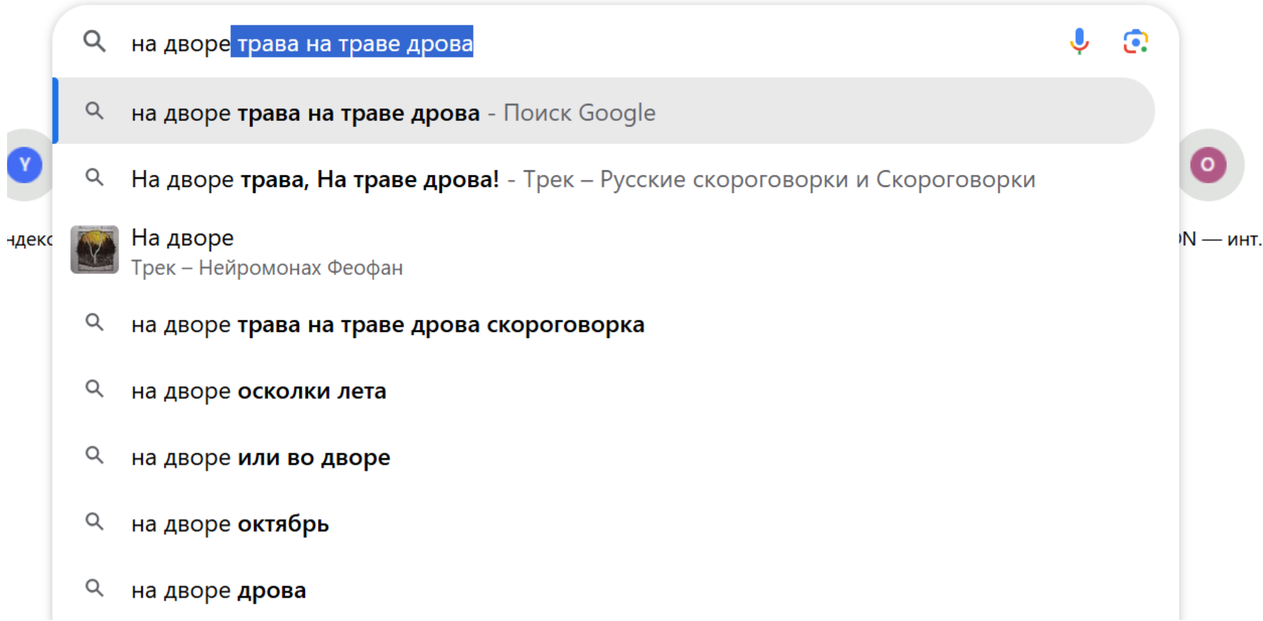
Языковое моделирование

Языковое моделирование (language modeling) — задача обработки естественного языка NLP, заключающаяся в предсказании вероятности появления последовательностей слов.

Google

В лесу родилась _____

ёлочка
белочка
обезьянка
дедушка
.....



Задачи

1. Предсказание следующего слова
Модель обучается по последовательностям и должна предсказать следующее слово, учитывая предыдущие.
Пример: «Я пошёл в...» → «магазин»
2. Маскирование и восстановление слов
Некоторые слова в предложении скрываются, и модель должна угадать их.
Пример: «Я пошёл в [MASK]» → «кино»
3. Генерация текста
Создание связных и осмысленных текстов на основе начальной подсказки.
4. Оценка правдоподобия текста
Модель оценивает, насколько вероятен данный текст.
5. Обобщённое языковое понимание
Включает задачи классификации текста, извлечения сущностей, ответа на вопросы, парафразирования и т. д.

Постановка задачи

Дан словарь: $\mathbb{V} = \ : |\mathbb{V}| = N$

Текст как конечная последовательность токенов: $w_0, w_1, \ldots, w_{t-1} : \forall i = 1, \ldots, t-1 \Rightarrow w_i \in \mathbb{V}$

Вычислить вероятность следующего токена, то есть вычислить распределение вероятностей:

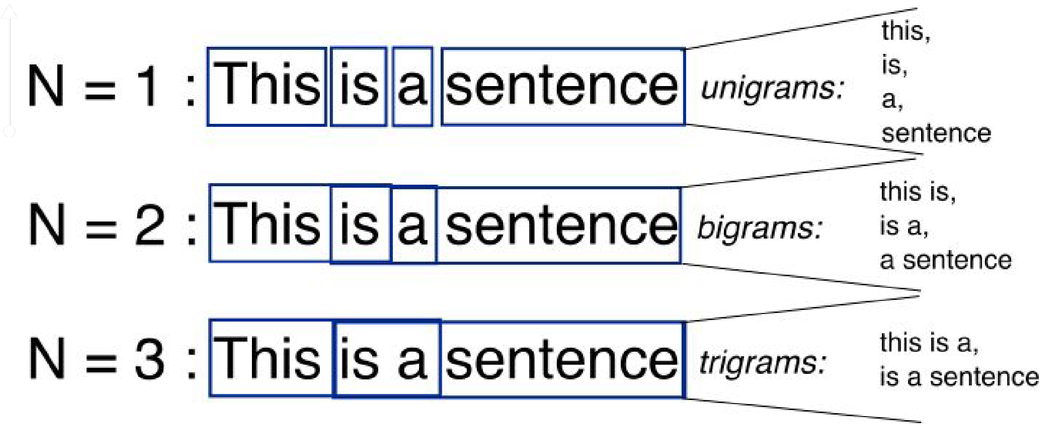
$$\forall w_t^i \in \mathbb{V} \Rightarrow P_i = P_\Theta(w_t^i | w_{t-1}, \ldots, w_0)$$

Тогда вероятность текста: $$P(w_0, w_1, \ldots, w_t) = \prod_{i=0}^{t} P(w_i | w_{i-1}, \ldots, w_0)$$

$$log[P(w_0, w_1, \ldots, w_t)] = log[\prod_{i=0}^{t} P(w_i | w_{i-1}, \ldots, w_0)] = \sum_{i=0}^{\tau} log[P(w_i | w_{i-1}, \ldots, w_0)]$$

n-gram Language Models

N-граммы — это статистические языковые модели, которые предсказывают следующее слово после N–1 слов на основе вероятности их сочетания.



Марковская цепь:

$$P(x_{t+1} \mid x_1, x_2, \ldots, x_t) = P(x_{t+1} \mid x_{t-n+2}, \ldots, x_t) = \frac{P(x_{t+1}, x_{t-n+2}, \ldots, x_t)}{P(x_{t-n+2}, \ldots, x_t)}$$

вероятность n-gram

(n-1) слов

вероятность (n-1)-gram

n-gram Language Models

Google N-grams — это коллекция n-грамм на английском языке, которая содержит частоты для более одного триллиона токенов.
Используется как инструмент анализа для культурологии, лингвистики, истории и социологии.

Проблемы с n-grams:

1. Нулевая вероятность новых фраз
Решение: применение **сглаживания** — добавление небольшого значения δ к счётчику каждого слова в словаре, что позволяет избежать нулевых вероятностей.

2. Проблема отсутствия данных
Решение: использование **бэктрекинга** (откат) — переход к модели с меньшим n.

3. Увеличение значения n или размера корпуса текстов приводит к резкому росту размера модели.

Нейросетевой подход

1. Получаем векторное представление слов (эмбеддинги):

$$s_j = W_{d\times|V|}x_j$$

2. Состояние на скрытом слое:

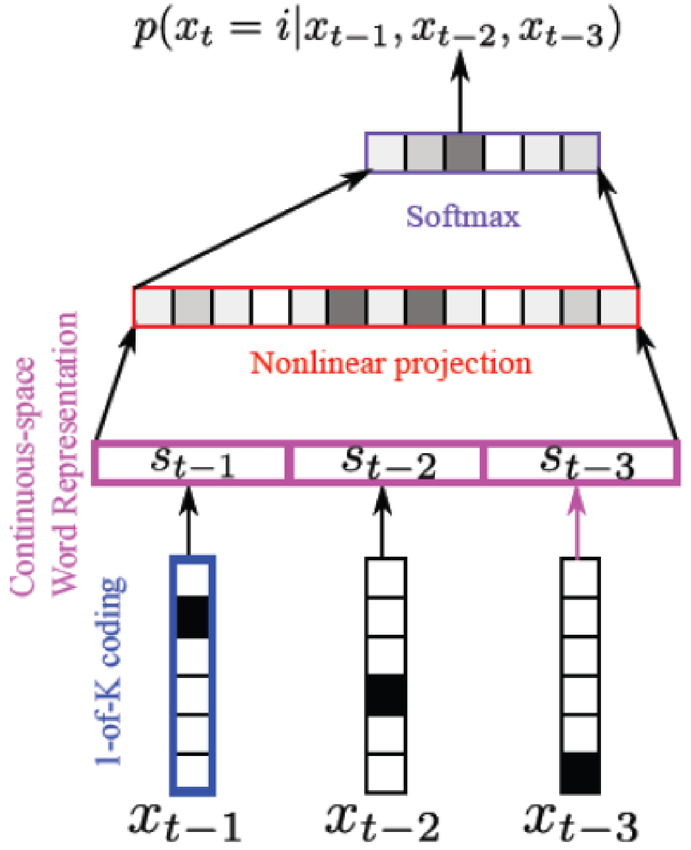
$$h = \tanh(U_{d'\times nd}[s_{t-1}, \ldots, s_{t-n}] + b)$$

3. Выходной слой:

$$y = V_{|V|\times d'}h + c$$

4. Вероятностное распределение:

$$p(x_t = i|x_{t-n}, \ldots, x_{t-1}) = \mathrm{softmax}(y)$$



«+» ушли от проблемы размерности, не нужно хранить n-gramms

«-»

ограниченный контекст (например, 3-5 слов) не позволяет уловить долгосрочные зависимости.

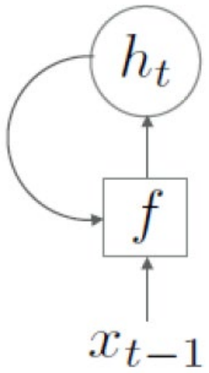
RNN моделирование

RNN не марковская цепь, способна работать с контекстом **переменной и практически неограниченной длины**.

Мы считаем зависимость от всех слов предложения:

$$p(x_t,\ldots,x_T)=\prod_{t=1}^{T} p(x_t \mid x_1,\ldots,x_{t-1})$$

Рекуррентная сеть может обрабатывать последовательность любой длины. Вместо того чтобы хранить и обрабатывать все предыдущие слова сразу, RNN использует **рекурсию:**



$$h_0 = 0$$

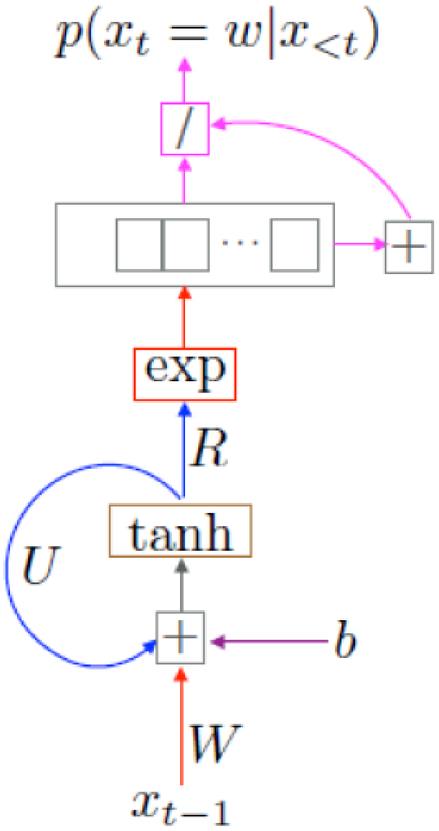
$$h_t = f(x_{t-1}, h_{t-1})$$

$$p(x_t \mid x_1,\ldots,x_{t-1}) = g(h_t)$$

внутреннее состояние - память

RNN моделирование

Предсказать вероятность следующего слова = вероятность того что следующее слово будет x_t будет словом w при условии всех предыдущих слов $x_{<t}$.



1. Функция обновления внутреннего состояния (transition):

$$h_t = \tanh(W_{d\times|V|}x_{t-1} + U_{d\times d}h_{t-1} + b)$$

2. Функция получения выхода (readout):

$$(p(x_t = w|x_{<t}))_{w=1}^{|V|} = g(h_t) = \text{softmax}(R_{|V|\times d}h_t + c)$$

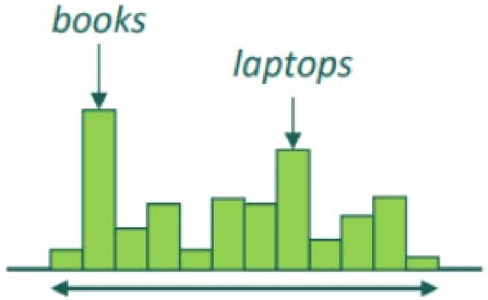
3. Функция потерь для обучения:

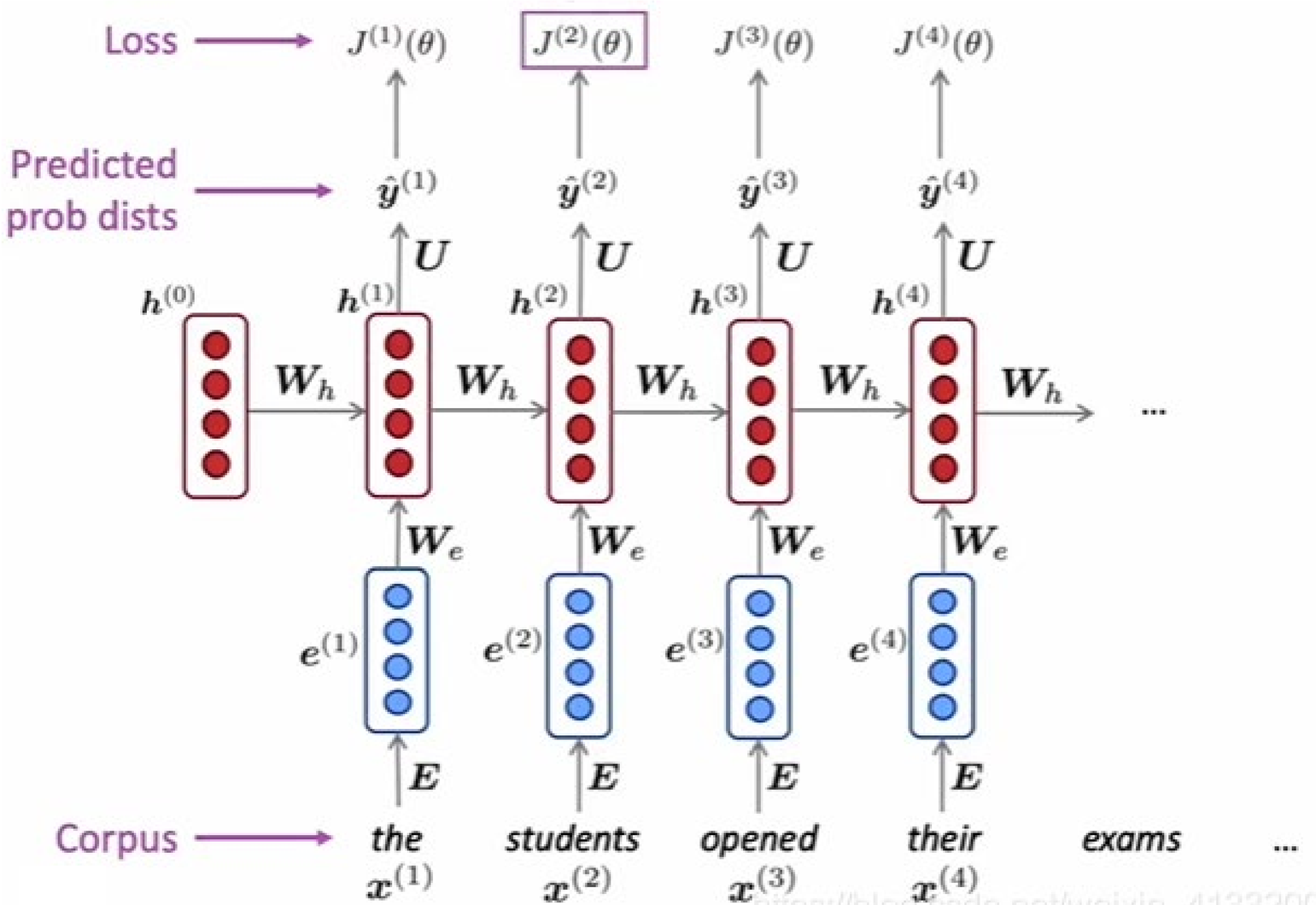
$$-\frac{1}{m}\sum_{i=1}^{m}\sum_{t=1}^{\text{len}(i)}\log p(x_t^{(i)}|x_1^{(i)},\ldots,x_{t-1}^{(i)}) \to \min$$

RNN моделирование обучение

Students open their _____

$$\hat{y}^{(4)} = P(x^{(5)}|\text{the students opened their})$$





https://web.stanford.edu/class/cs224n/

RNN для генерации текста

1. **Инициализация.** Процесс начинается с начального слова или фразы (например, «Однажды»), а также нулевого начального скрытого состояния.

2. **Предсказание.** Сеть принимает входное слово, преобразует его в векторное представление и на основе своего текущего состояния вычисляет распределение вероятностей для следующего слова.

3. **Выбор.** Следующее слово выбирается на основе этого распределения.

4. **Рекурсия.** Выбранное слово подается обратно на вход сети на *следующем* шаге. Скрытое состояние обновляется, включая в себя новую информацию, и процесс повторяется: предсказание → выбор → подача обратно.

Перплексия

Перплексия (perplexity) — это числовая метрика, которая измеряет **степень неопределённости языковой модели** относительно предсказания следующего слова в последовательности.

$$PP(W) = P(x_1, \ldots, x_T)^{-1/T} = \prod_{t=1}^{T} \left(\frac{1}{p(x_t | x_1, \ldots, x_{t-1})} \right)^{1/T}$$

Если модель уверена в своих предсказаниях (высокие вероятности правильных слов), то **перплексия будет низкой.**

Если модель неуверенна (низкие вероятности правильных слов), то **перплексия будет высокой.**

МИСИС
УНИВЕРСИТЕТ

Спасибо
за внимание!