Embeddings

Самарин Игорь, группа 23.M03-мм

Санкт-Петербургский государственный университет
Кафедра статистического моделирования

Санкт-Петербург
2024 г.

Введение

Обработка естественного языка — это область, посвященная анализу текстов, написанных на естественных (человеческих) языках.

Различают четыре режима работы с текстовыми данными:

- **Many-to-one**: на входе последовательность объектов, на выходе один объект;
- **One-to-many**: на входе один объект, на выходе последовательность объектов;
- **Many-to-many**: на входе и выходе последовательности нефиксированной длины;
- **Синхронизированный many-to-many**: на входе и выходе последовательности фиксированной длины, токены входной последовательности явно сопоставлены токенам выходной.

Перед рассмотрением архитектур, необходимо разобраться с базовыми понятиями о нейронных сетях.

Нейронная сеть. Определение

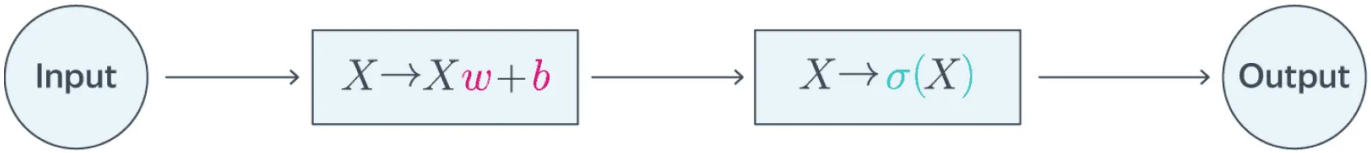
Искусственная нейронная сеть — это сложная дифференцируемая функция, задающая отображение из исходного пространства в пространство ответов, все параметры которой могут настраиваться одновременно и взаимосвязанно.

Сложную функцию удобно представлять в виде суперпозиции простых. Простейшие разновидности:

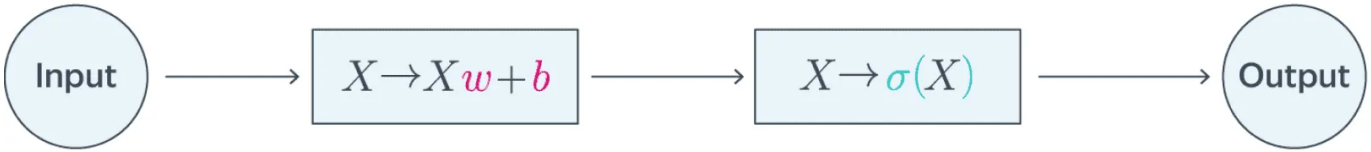
- **Линейный слой**: линейное преобразование над входящими данными. Обучаемые параметры — матрица W и вектор b такие, что $x \mapsto xW + b$, где $(W \in \mathbb{R}^{d \times k}, x \in \mathbb{R}^d, b \in \mathbb{R}^k)$;
- **Функция активации**: нелинейное преобразование, поэлементно применяющееся к пришедшим на вход данным.

Таким образом, нейронную сеть можно представить в виде вычислительного графа, где промежуточным вершинам соотствуют преобразования.

Нейронная сеть



Применение



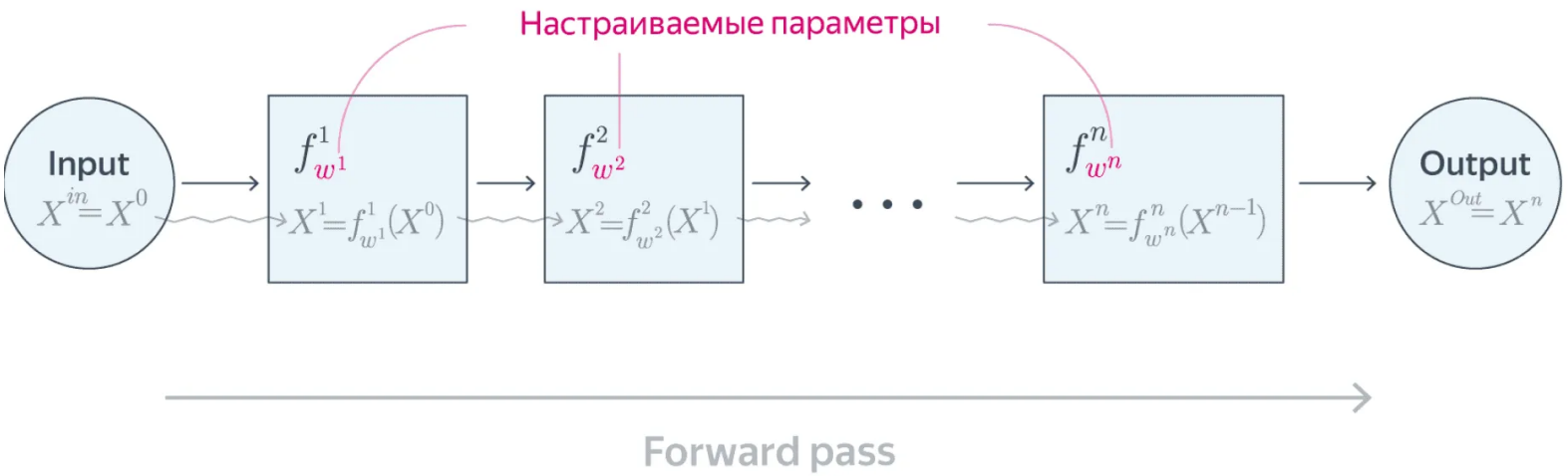
$X^{in}=X^0$ → $X^1=X^0w+b$ → $X^2=\sigma(X^1)$ → $X^2=X^{Out}$

Приходит на вход

То что возвращает граф

Нейронная сеть. Forward propagation

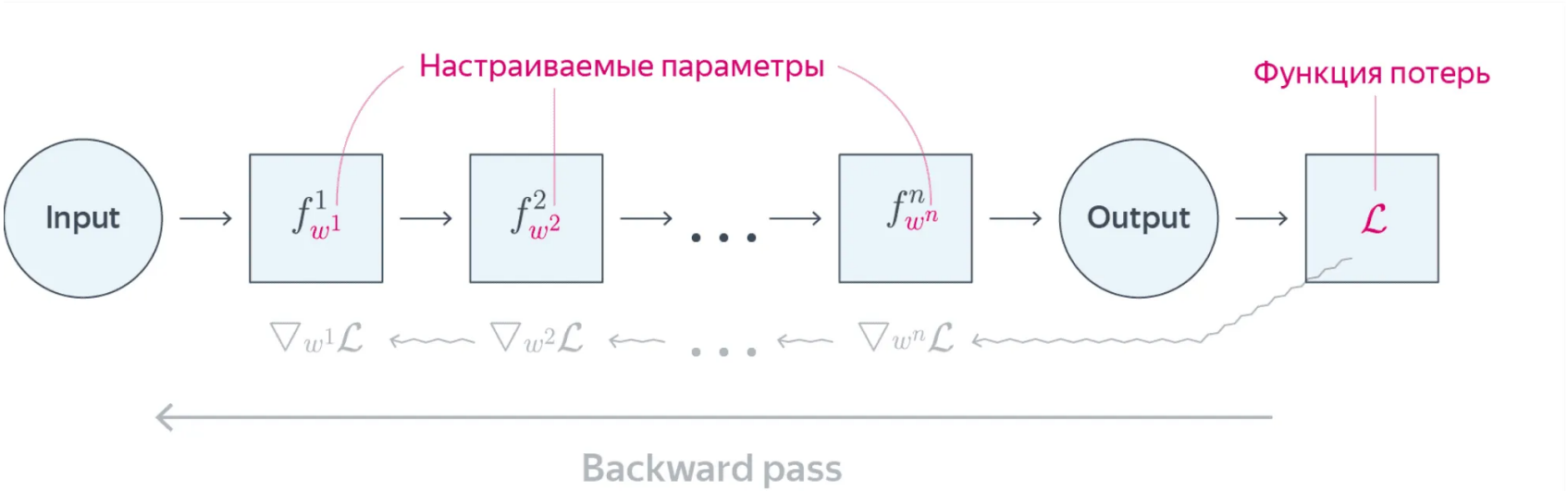
Применение нейронной сети к данным (вычисление выхода по заданному входу) назвается *прямым проходом* (forward propagation).



На этом этапе происходит преобразование исходного представления данных в целевое.

Нейронная сеть. Backward propagation

При *обратном проходе* (backward propagation), информация (обычно об ошибке предсказания) движется от финального представления к исходному через все преобразования.



Нейронная сеть. Backward propagation в одномерном случае

Пусть ω_0 — переменная, по которой хотим продифференцировать сложную функцию вида:

$$f(\omega_0) = g_m(g_{m-1}(\dots g_1(\omega_0)\dots)),$$

где все g_i скалярные.

Тогда производная имеет вид:

$$f'(\omega_0) = g'_m(g_{m-1}(\dots g_1(\omega_0)\dots)) \cdot g'_{m-1}(g_{m-2}(\dots g_1(\omega_0)\dots)) \cdot \dots \cdot g'_1(\omega_0),$$

где $g_1(\omega_0), g_2(g_1(\omega_0)), \dots, g_{m-1}(\dots g_1(\omega_0)\dots)$ известны.

Как правило, на практике пользуются матричными вычислениями.

Word Embeddings. Статистические подходы

Прежде чем подавать текстовые данные на вход нейросети, их необходимо векторизировать.

К векторизации текстов есть два базовых подхода:

- Векторизовать текст целиком, превращая его в один вектор;
- Векторизовать отдельные структурные единицы, превращая текст в последовательность векторов.

Самый простой вариант — Bag-of-Words. Текст представляется в виде вектора частот встречаемости каждого токена.

Чуть более сложный — TF-IDF. Представление текста d состоит из произведений $TF(t,d) \cdot IDF(t,D)$ по всем токенам t из коллекции D, где $TF(t,d) = \frac{n_t}{\sum_k n_k}$, а $IDF(t,D) = \log \frac{|D|}{|\{d_i \in D | t \in d_i\}|}$.

Word Embeddings. Word2vec

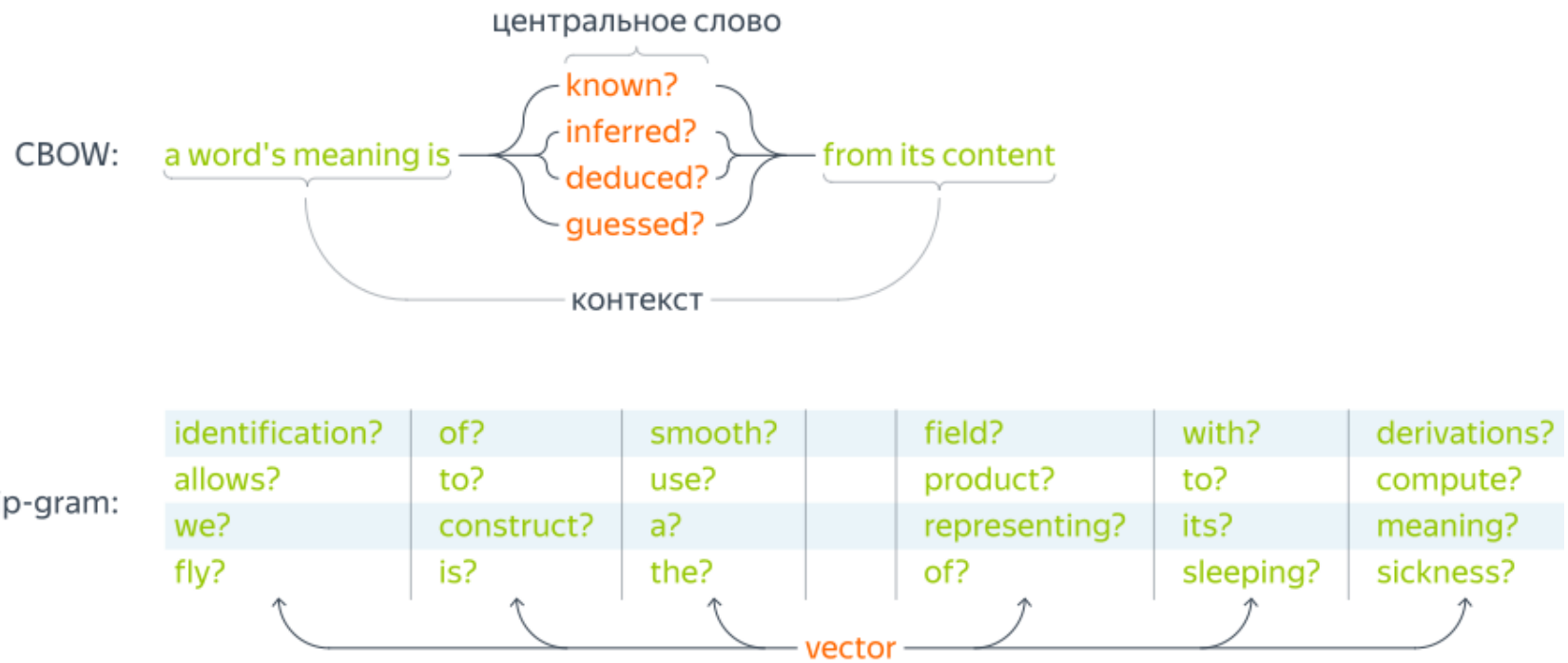
Word2vec — способ построения сжатого пространства векторов слов, использующий нейронные сети.

Для обучения авторы предлагают две стратегии:

- **CBOW**: модель учится предсказывать данное (центральное) слово по контексту;
- **Skip-gram**: модель учится по данному слову предсказывать контекст.

Нейросеть учит векторы слов так, чтобы скалярное произведение между векторами двух слов, которые часто используются в контексте, были как можно больше.

Word Embeddings. Word2vec



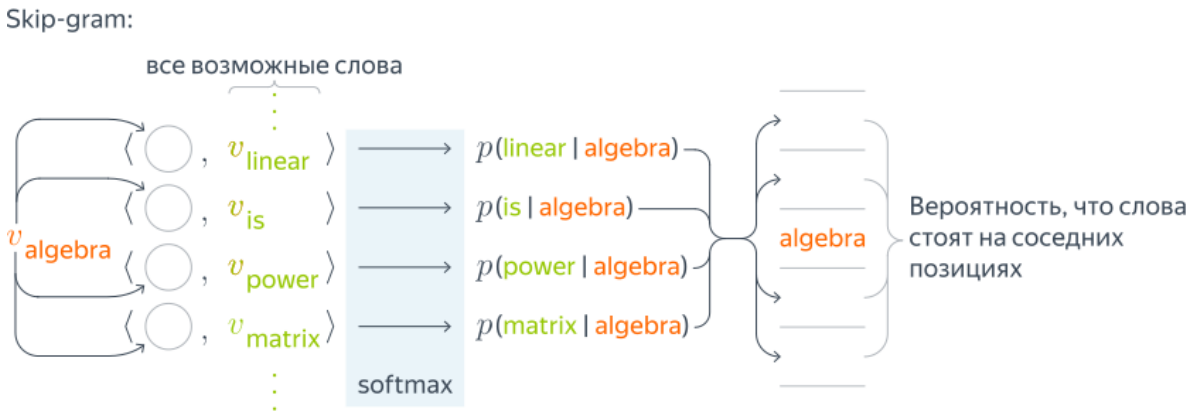
Word Embeddings. Word2vec. CBOW

В модели CBOW вычисляются $logits_u =< \sum_{w \in context} v_w, v_u >$, после чего — вероятности всевозможных слов u быть центральными для контекста как $softmax(logits)$.

Модель учится на кросс-энтропию полученного распределения с истинным распределением центральных слов.

Word Embeddings. Word2vec. SkipGram

В модели Skip-gram по центральному слову u для каждой позиции контекста предсказывается распределение вероятностей.

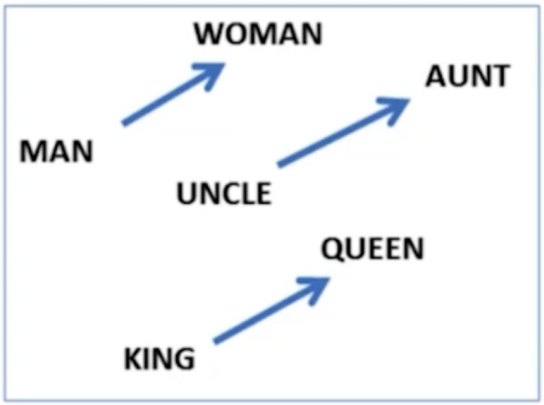


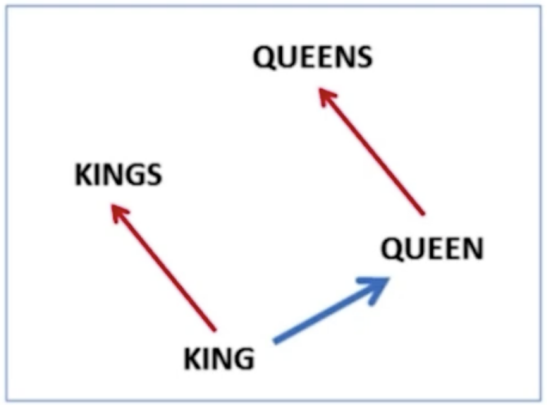
Функция потерь: сумма кросс-энтропий распределений слов контекста с их истинными распределениями.

Word Embeddings. Word2vec. Особенности

Векторы содержат "смысл"слов. Два вектора можно сравнивать при помощи косинусного расстояния.

$$\mathbf{v}(\text{king}) - \mathbf{v}(\text{man}) + \mathbf{v}(\text{woman}) \approx \mathbf{v}(\text{queen})$$





Word Embeddings. Word2vec. Преимущества и недостатки

Преимущества:

- Векторы отражают смысл слов;
- Размерность векторов не зависит от размера словаря;
- При добавлении документов векторы можно дообучить.

Недостатки:

- Фиксированный размер словаря;
- Для редких слов эмбеддинги получаются неоптимальными;
- Слова имеющие один корень, обрабатываются по-разному.

Word Embeddings. FastText

Библиотека FastText является технической модификацией Word2Vec.

Модель sub-word: разделим слова на буквенные n-граммы, тогда вектор слова будет сумма векторов его n-грамм.

Преимущества:

- Можно получать более адекватные эмбеддинги для редких и неизвестных слов;

Недостатки:

- n-грамм может быть очень много;
- Требуется больше вычислительных ресурсов.