

данных в кеше продолжительное время, когда вызовы внутренней системы в реальном времени невозможны.

ЛИТЕРАТУРА

1. Карпович М. Н. Проектирование микросервисных архитектур информационных систем // Информационные технологии: материалы 86-й науч.-техн. конференции профессорско-преподавательского состава, научных сотрудников и аспирантов (с международным участием), Минск, 31 января – 12 февраля 2022 года [Электронный ресурс] / отв. за издание И.В. Войтов; УО БГТУ. – Минск: БГТУ, 2022. – С. 82-85.

2. Ньюман, С. От монолита к микросервисам: эволюционные паттерны для преобразования вашего монолита / С. Ньюман – США: O'Reilly Media, 2019. – 94 с.

3. Ричардсон, С. Шаблоны микросервисов: с примерами на Java / С. Ричардсон – США: Manning Publications, 2018. – С. 65-110.

4. Кристудас, Б. Практические архитектурные шаблоны микросервисов / Б. Кристудас - США: Апрель, 2019 – С. 77-104.

5. Черноусов А. Кэширование данных в микросервисной архитектуре [Электронный ресурс] / Habrahabr. – URL: <https://habr.com/ru/post/651829> (дата обращения: 10.02.2023).

УДК 004.85

Ст. преп. И.Г. Сухорукова
(БГТУ, г. Минск)

ПРИМЕНЕНИЕ МЕТОДОВ ВЛОЖЕНИЯ В ЗАДАЧАХ МАШИННОГО ОБУЧЕНИЯ

Для работы с методами машинного обучения изображения, текст и прочие объекты обычно представляется в виде вектора или матрицы с действительными числами. Этот вектор (или матрица) в контексте машинного обучения называется вложением (embedding) входных данных.

Первоначально методы вложения получили распространение при обработке естественного языка. Наибольший интерес представляют методы вложения слов, которые позволяют учитывать семантическое значение слов. К примеру модель вложения слов *Word2Vec* обучается методами глубокого обучения на огромном количестве текста с предсказанием, какое слово возникает в схожих контекстах. После обучения *Word2Vec* генерирует вектор из 300 измерений для каждого слова в словаре, этот вектор и есть вложение слова. Имея вектор-

ное представление слов, можно вычислять сходство между словами при помощи косинусного сходства. Это дает возможность использовать вложения слов для поиска синонимов, выполнения арифметических операций со словами. Знаменитые примеры, которые показывают невероятные свойства вложений, — понятие аналогий. Например, используя библиотеку *Gensim* в *Python*, можно складывать и вычитать векторы слов, а библиотека найдёт слова, самые близкие к результирующему вектору. Реализуем известный пример – формулу «король – мужчина + женщина»:

```
model.most_similar(positive=["king", "woman"], negative=["man"])

[('queen', 0.8523603677749634),
 ('throne', 0.7664333581924438),
 ('prince', 0.7592144012451172),
 ('daughter', 0.7473883032798767),
 ('elizabeth', 0.7460219860076904),
 ('princess', 0.7424570322036743),
 ('kingdom', 0.7337411642074585),
 ('monarch', 0.721449077129364),
 ('eldest', 0.7184862494468689),
 ('widow', 0.7099430561065674)]
```

Рисунок 1 – Список наиболее похожих слов с рассчитанным коэффициентом геометрического сходства

Подход вложений слов используют в комбинации с методами машинного обучения для решения широкого спектра задач, например, в задачах классификации текстов применение вложений перед классическими методами классификации позволяет увеличивать точность модели [1]. Благодаря тому, что вложения хранят некоторое смысловое значение слова их используют для построения семантических пространств, позволяющих анализировать естественный язык и многое другое. Кроме того, возможность представлять слова в виде вложений позволяет использовать намного меньше памяти, поскольку длина вектора обычно намного короче, чем словарь языка (если выполнить горячее кодирование для представления каждого слова в словаре).

Методы вложений работают не только со словами, но и с любыми другими сущностями. Такими сущностями могут быть пользователи соцсетей или истории покупок клиентов магазина. При этом вложения используют для категориальных переменных в массивах данных. Вложения позволяют лучше учесть различные отношения, чем необработанные категории, поэтому применение вложений для массивов категориальных данных дает хорошие результаты при построении

моделей на основе методов машинного обучения. Хорошим примером служит модель рекомендаций для сети *Pinterest*, по аналогии с *Word2Vec* метод получил название *Pin2Vec*. В *Pin2Vec* для каждого пина создается 128-мерное вложение. Применение вложений позволило увеличить релевантность связанных пинов в рекомендациях пользователям [2].

Подход с использованием нейронных сетей вместе с категориальными вложениями также может быть применен к данным временных рядов.

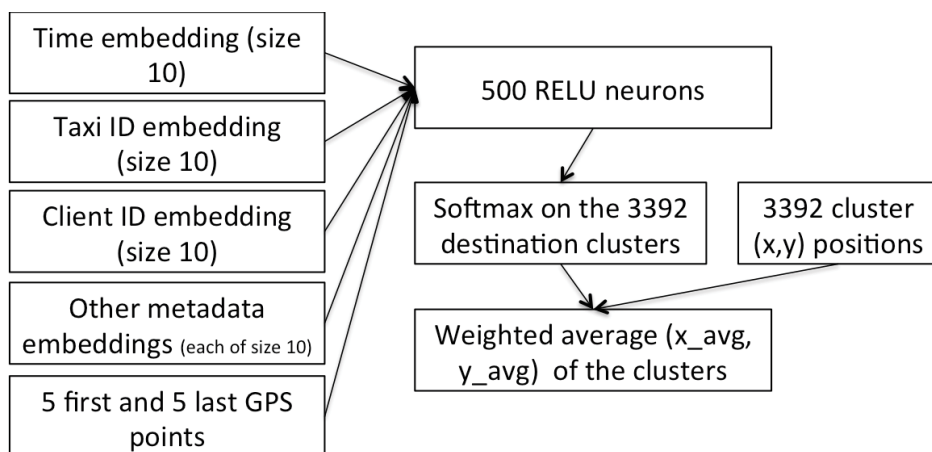


Рисунок 2 – Архитектура-победитель конкурса Kaggle Taxi Trajectory

На рис. 2 показана архитектура модели, занявшей 1-е место в конкурсе такси Kaggle, которая использует траекторию точек GPS и временные метки для прогнозирования продолжительности поездки на такси. А рис. 3 иллюстрирует сокращение размера данных в этой модели, при замене категориальных значений их вложениями [3].

Metadata	Number of possible values	Embedding size
Client ID	57106	10
Taxi ID	448	10
Stand ID	64	10
Quarter hour of the day	96	10
Day of the week	7	10
Week of the year	52	10

Рисунок 3 – Значения метаданных и связанный с ними размер внедрения

При построении системы рекомендаций платформы Instacart на основе последовательности выбора продуктов покупателями использование вложений позволило визуализировать кластеры распределения продуктов, что может быть использовано для рекомендации рас-

положения отделов магазина. [4]

Таким образом использование вложения сущностей в машинном обучении позволяет учитывать контекст, сокращать размерность, увеличивать точность модели и получать любопытные и легко интерпретируемые визуализации данных.

ЛИТЕРАТУРА

1. Ameisen E., Using Machine Learning to understand and leverage text [Электронный ресурс] – URL: <https://blog.insightdatascience.com/how-to-solve-90-of-nlp-problems-a-step-by-step-guide-fda605278e4e> (дата обращения: 03.02.2023).
2. Kevin Ma, Applying deep learning to Related Pins [Электронный ресурс] – URL: <https://medium.com/the-graph/applying-deep-learning-to-related-pins-a6fee3c92f5e> (дата обращения: 08.02.2023).
3. Brébisson A., Étienne S., Auvolat A., Pascal V. Artificial Neural Networks Applied to Taxi Destination Prediction [Электронный ресурс] – URL: <https://arxiv.org/pdf/1508.00021.pdf> (дата обращения: 08.02.2023).
4. Jeremy Stanley, Deep Learning with Emojis [Электронный ресурс] – URL: <https://tech.instacart.com/deep-learning-with-emojis-not-math-660ba1ad6cdc> (дата обращения: 05.02.2023).

УДК 004.4'2

Преп.-стаж. К.Д. Якубенко
(БГТУ, г. Минск)

UX ВИРУСНОСТЬ: ОПРЕДЕЛЕНИЕ, ВИРУСНЫЙ ЦИКЛ, ПЕРЕДАЧА, К-ФАКТОР

Бывают вещи *популярными*, но совсем не *вирусными*. Что-то может стать “вирусным”, не получив миллиардов просмотров на YouTube и не став известным за ночь, например, Pinterest, или email. Вирусность базируется на продукте, контенте или идее, которую дизайнеры, маркетологи, разработчики создают.

Если проанализировать данные, можно сделать вывод, что вирусные вещи распространяются достаточно предсказуемо, особенно в соцсетях. Вирусный (виральный) механизм роста ориентирован на привлечение новых пользователей посредством личных рекомендаций от тех, кто уже использует ваш продукт (как при непосредственном общении, так и в соцсетях). Он крайне важен для стартапов, которые предлагают приложения для знакомств или мессенджеры, т. е. тех, для эффективного использования которых требуется наличие базы активных пользователей из числа друзей.