

Content-based рекомендации

Введение



В этом уроке мы поговорим об использовании других данных для рекомендаций. Не только о «рейтингах», но и о любой иной информации об объектах и/или пользователях.

Ранее в сериале...

Обобщающий пример



U – субъекты (пользователи)

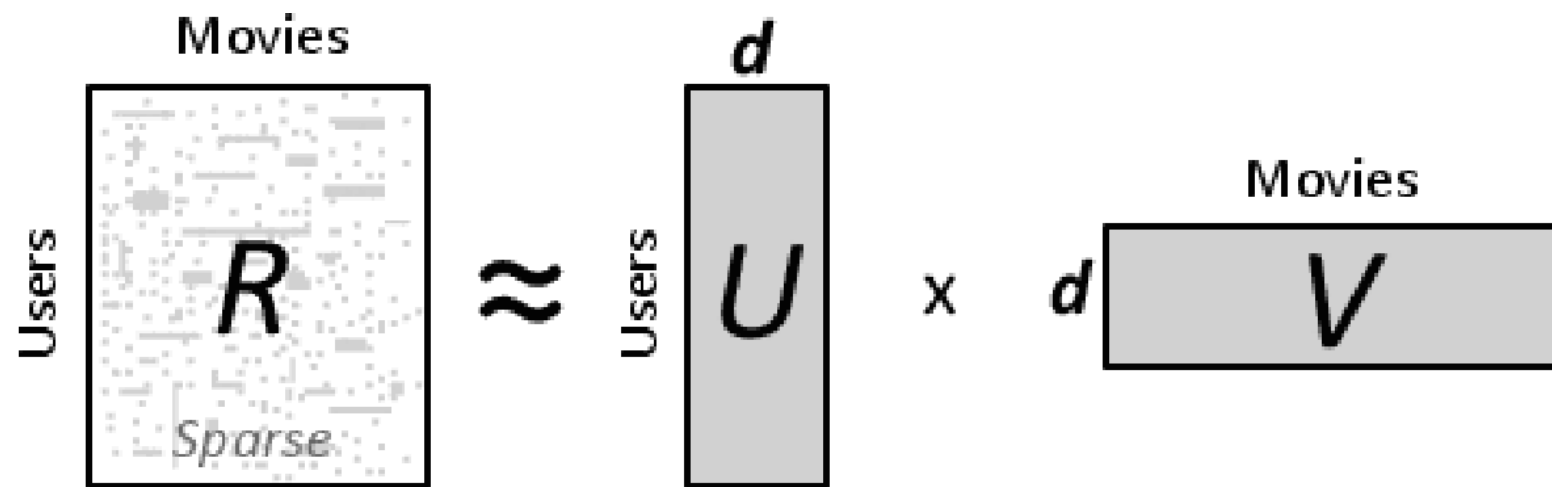
I – объекты

r_{ui} = [измеримая функция описывающая взаимодействие u и i]

Задача:

- Восстановить матрицу R
- Найти близкие («похожие») элементы по u
- Найти близкие («похожие») элементы по i

Разложение матрицы R



$$\hat{r}_{ui} = \langle \mathbf{p}_u, \mathbf{q}_i \rangle$$

Factorization Machines (FM)

Идея FM на картинке:

Feature vector x																	Target y					
$x^{(1)}$	1	0	0	...	1	0	0	0	...	0.3	0.3	0.3	0	...	13	0	0	0	0	...	5	$y^{(1)}$
$x^{(2)}$	1	0	0	...	0	1	0	0	...	0.3	0.3	0.3	0	...	14	1	0	0	0	...	3	$y^{(2)}$
$x^{(3)}$	1	0	0	...	0	0	1	0	...	0.3	0.3	0.3	0	...	16	0	1	0	0	...	1	$y^{(2)}$
$x^{(4)}$	0	1	0	...	0	0	1	0	...	0	0	0.5	0.5	...	5	0	0	0	0	...	4	$y^{(3)}$
$x^{(5)}$	0	1	0	...	0	0	0	1	...	0	0	0.5	0.5	...	8	0	0	1	0	...	5	$y^{(4)}$
$x^{(6)}$	0	0	1	...	1	0	0	0	...	0.5	0	0.5	0	...	9	0	0	0	0	...	1	$y^{(5)}$
$x^{(7)}$	0	0	1	...	0	0	1	0	...	0.5	0	0.5	0	...	12	1	0	0	0	...	5	$y^{(6)}$
	A	B	C	...	TI	NH	SW	ST	...	TI	NH	SW	ST	...	Time	TI	NH	SW	ST	...		
	User				Movie					Other Movies rated						Last Movie rated						

Первые 4 столбца (синие) - индикаторы для активного пользователя.

Следующие 5 (красных) переменных индикатора для **активного** элемента.

Следующие 5 столбцов (желтые) содержат дополнительные неявные индикаторы (т.е. другие фильмы, которые оценил пользователь). (Зеленый) представляет время до оценки. Последние 5 столбцов (коричневые) содержат индикаторы для последнего фильма, который пользователь оценил до **активного**.

Косинусное расстояние

Основная идея, на которой базируется расчет косинусного расстояния, заключается в том, что строку из символов можно преобразовать в числовой вектор. Если проделать эту процедуру с двумя сравниваемыми строками, то меру их сходства можно оценить через косинус между двумя числовыми векторами.

$$\text{similarity} = \cos(\theta) = \frac{XY}{\|X\| \|Y\|} = \frac{\sum_{i=1}^n x_i y_i}{\sqrt{\sum_{i=1}^n x_i^2 \sum_{i=1}^n y_i^2}}$$

Из курса школьной математики известно, что если угол между векторами равен 0 (то есть векторы полностью совпадают), то косинус равен 1.

План

1

Использование
информации об
объектах

2

Использование
текстовых
признаков

3

Использование
картинок

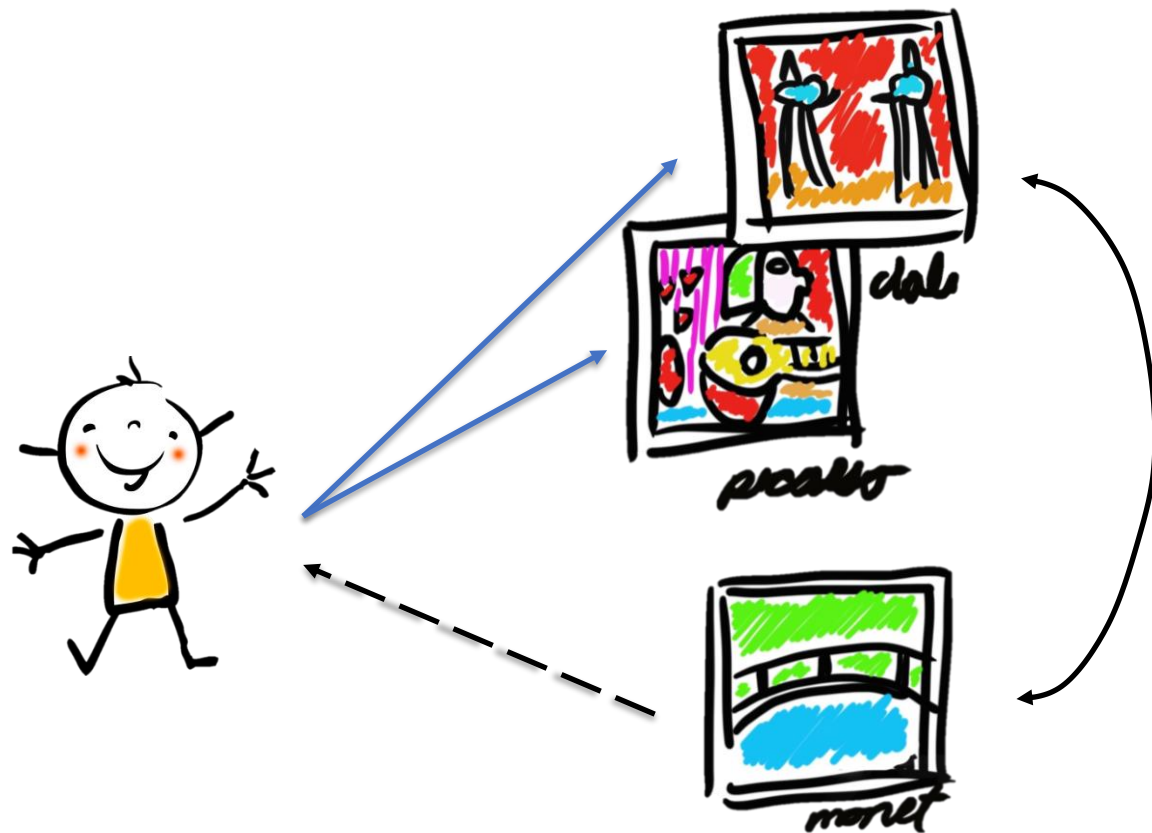
4

Постановка
задачи

Использование информации об объектах

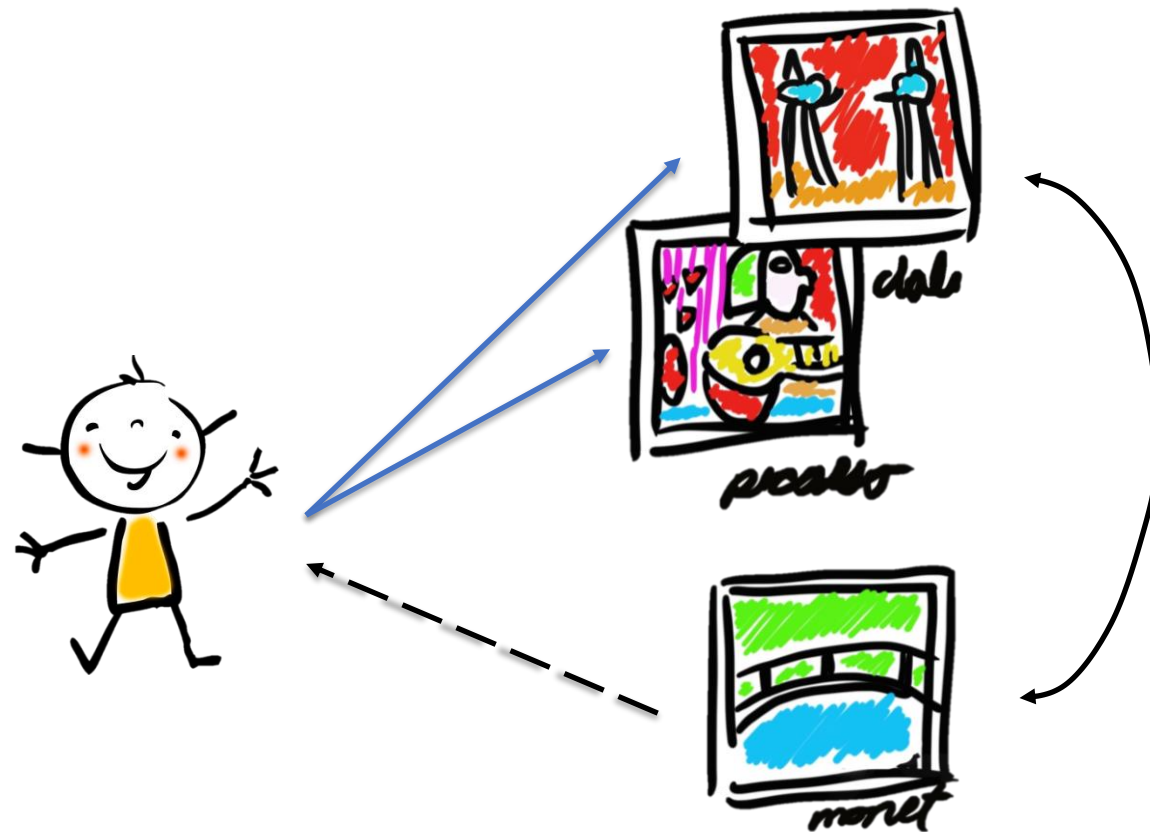
Content-based рекомендации

- Рассчитываются признаки для пользователей и объектов
- Строится модель классификации/регрессии, приближающая оценки пользователей



Content-based рекомендации

- Процесс рекомендации в состоит в сопоставлении атрибутов профиля пользователя с атрибутами объекта содержимого.
- Результатом является оценка релевантности, которая отражает уровень интереса пользователя к этому объекту.



Постановка задачи рекомендации



U – субъекты (пользователи)

I – объекты

r_{ui} = [измеримая функция описывающая взаимодействие u и i]

Задача:

- Восстановить матрицу R
- Найти близкие («похожие») элементы по u
- Найти близкие («похожие») элементы по i

Постановка задачи рекомендации

U – субъекты (пользователи)

I – объекты

r_{ui} = [измеримая функция описывающая взаимодействие u и i]

Теперь мы дополнительно знаем:

\tilde{i} - признаки, которые можем извлечь об *item*

\tilde{u} - признаки, которые можем извлечь о *user*

Задача:

- Восстановить матрицу R
- Найти близкие («похожие») элементы по u
- Найти близкие («похожие») элементы по

Какие бывают типы признаков



- Числовые (год выпуска фильма, бюджет)
- Категориальные (жанр фильма)
- Текстовые (описание фильма, тизер)
- Изображение (кадр из фильма, обложка)
- Аудио (саундтрек)
- ...

Категориальные признаки



Многие классические методы машинного обучения предполагают, что все признаки $X^j \in \mathbb{R}$. Однако в некоторых задачах признаки могут принимать значения из множеств, не совпадающих с множествами вещественных чисел. Так, например, признаки могут принимать значения из конечного неупорядоченного множества. Например, это может быть признак Город со значениями из множества {Москва, Санкт-Петербург, Новосибирск, Казань, ...}.

Такие признаки называются **категориальными, факторными или номинальными**.

One-Hot Encoding



Алгоритм:

- Выбираем все значения переменной и создаем из них колонки
- Заполняем их бинарными значениями (1/0)
 - 1 – если было такое значение было в изначальной колонке
 - 0 – если не такое
- Profit!

Аналогично со списками.

Использование текстовых признаков

Мешок слов (bag-of-words)

Мешок слов (англ. bag-of-words) — упрощенное представление текста, которое используется в обработке естественных языков и информационном поиске.


Мы можем создать мешок слов из любого текста (описания фильмов или названия и т.д. или из описания профиля)

```
«Иван», «любит», «смотреть», «фильмы», «Мария», «тоже», «любит», «фильмы»  
«Иван», «также», «любит», «смотреть», «футбольные», «матчи»
```

Из этих списков можно создать объекты, представляющие мешок слов:

```
BoW1 = { "Иван" : 1, "любит" : 2, "смотреть" : 1, "фильмы" : 2, "Мария" : 1, "тоже" : 1 };  
BoW2 = { "Иван" : 1, "также" : 1, "любит" : 1, "смотреть" : 1, "футбольные" : 1, "матчи" : 1 };
```

Мешок слов (bag-of-words)



Далее мешок слов можно закодировать **one-hot encoding**.

Можно использовать кодирование только частотные слова.

TF-IDF



TF (term frequency — частота слова) — отношение числа вхождений слова к общему числу слов документа.

IDF (inverse document frequency — обратная частота документа) — инверсия частоты, с которой некоторое слово встречается в документах коллекции.

$$\text{TF-IDF} = \text{TF} * \text{IDF}$$

TF-IDF



$$w_{x,y} = tf_{x,y} \times \log \left(\frac{N}{df_x} \right)$$

TF-IDF

вес слова x в тексте y

$tf_{x,y}$ - частота слова x в тексте y

df_x - количество текстов, содержащих слово x

N – общее количество текстов

Взвешивание признаков

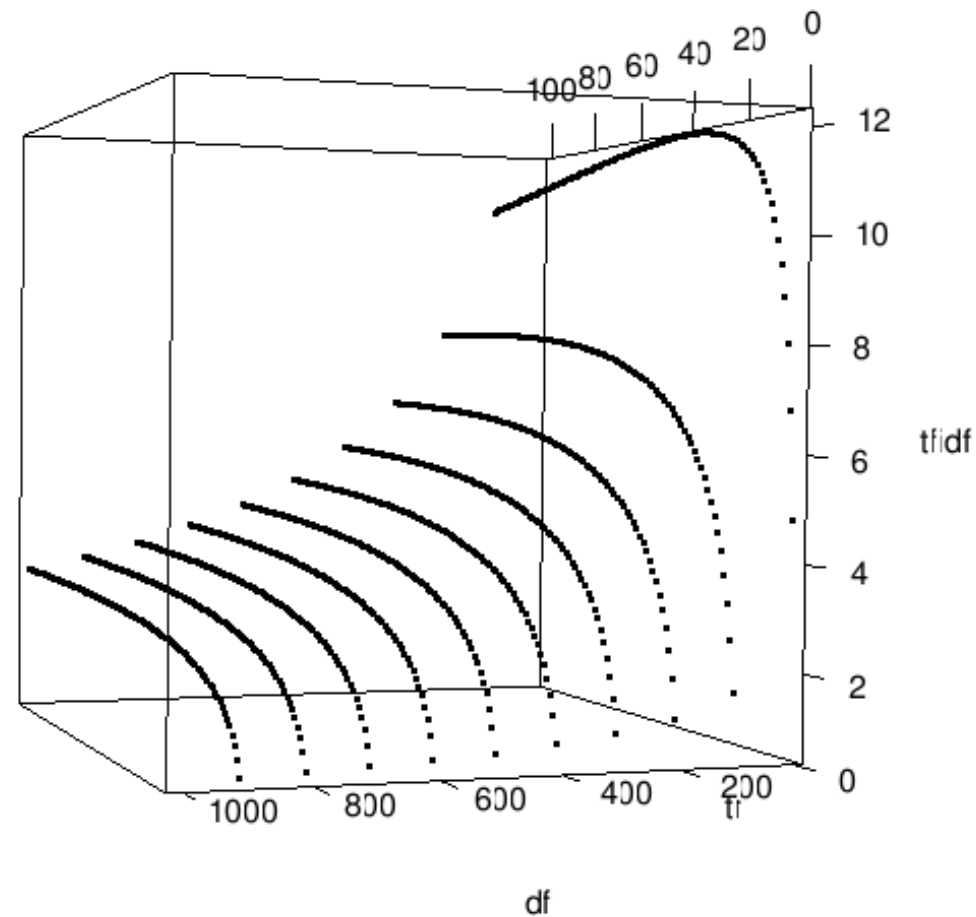


Кроме того, мы можем извлечь такие характеристики, как оценка настроений и **оценки TF-IDF** из описаний фильмов и обзоров.

Оценка TF-IDF отражает, насколько важно слово для документа в наборе документов.

Взвешивание признаков

Существует целое семейство похожих преобразований (например, BM25 и аналогичные), но содержательно все они повторяют ту же логику, что TF-IDF: редкие атрибуты должны иметь больший вес при сравнении товаров.



Извлечение признаков



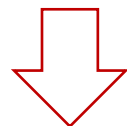
Энг Ли, комедия, поколение, еда, 1994...

(слова, извлеченные из названия/описания)



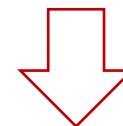
Комедия, роман, классика...

(слова, извлеченные из фильмов, которые пользователь высоко оценил, посмотрел или из покупок)



Методы ML (векторизация, TF-IDF)
Векторное представление слов

(0.3, 0.2, -0.1, ...)



(0.15, 0.4, 0.1, ...)

Векторные представления объектов



Чтобы построить и обучить модель, используют **технику embedding**, когда каждый объект превращается в вектор фиксированной длины, и близким объектам соответствуют близкие векторы. Практически всем известным моделям требуется, чтобы данные на входе были фиксированной длины, и набор векторов — простой способ привести их к такому виду.

Один из первых embedding-методов — **word2vec**.

Построение векторного описания объекта

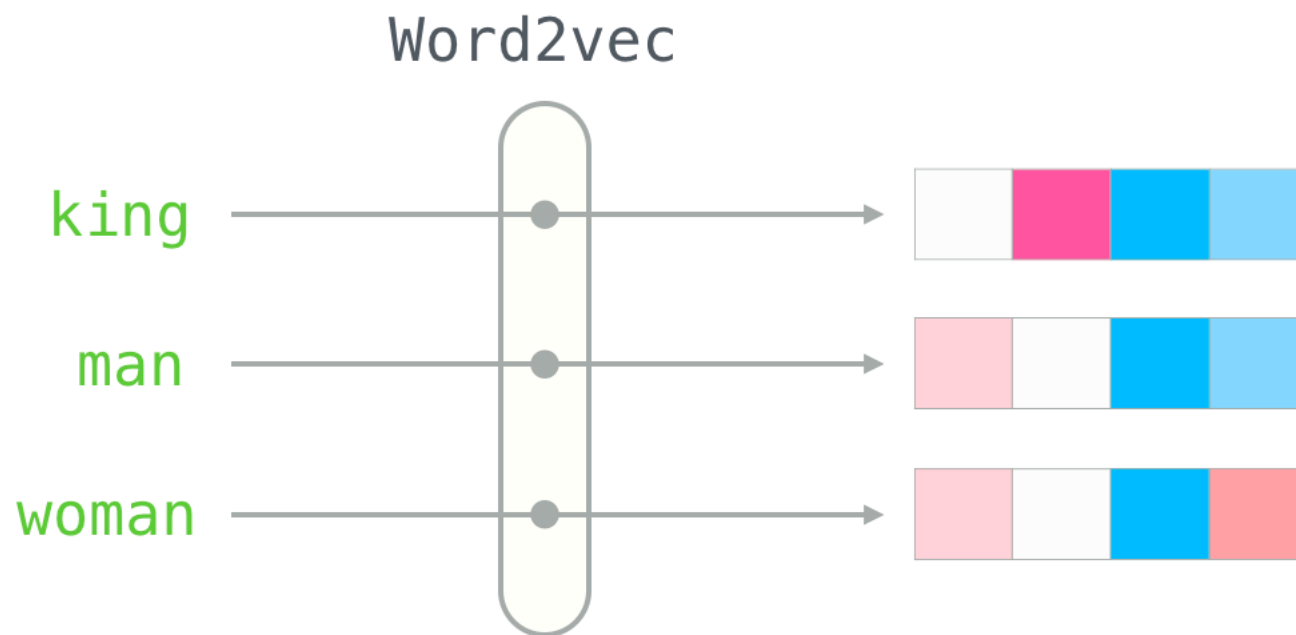


Есть два принципиально разных **способа построения векторного описания товара**:

- использовать контент — сверточные нейронные сети для извлечения признаков из фотографий, рекуррентные сети или мешок слов для анализа текстового описания;
- использование данных о взаимодействиях пользователей с товаром: какие товары и как часто смотрят/добавляют в корзину вместе с данным.

Word2vec

Word2vec — общее название для совокупности моделей на основе искусственных нейронных сетей, предназначенных для получения векторных представлений слов на естественном языке.



Word2vec



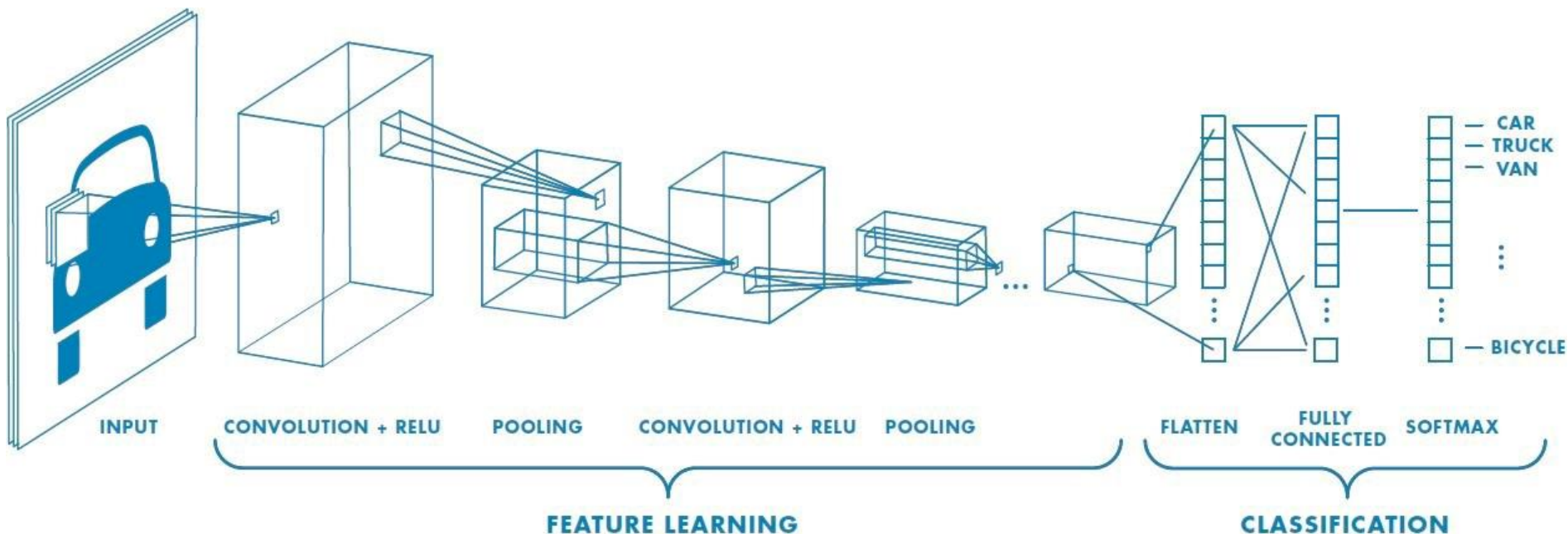
Работа алгоритма осуществляется следующим образом: word2vec принимает большой текстовый корпус в качестве входных данных и сопоставляет каждому слову вектор, выдавая координаты слов на выходе. Сначала он генерирует словарь корпуса, а затем вычисляет векторное представление слов, «обучаясь» на входных текстах.

Векторное представление основывается на контекстной близости: слова, встречающиеся в тексте рядом с одинаковыми словами (а следовательно, имеющие схожий смысл), будут иметь близкие (по косинусному расстоянию) векторы. Полученные векторные представления слов могут быть использованы для обработки естественного языка и машинного обучения.

Использование картинок

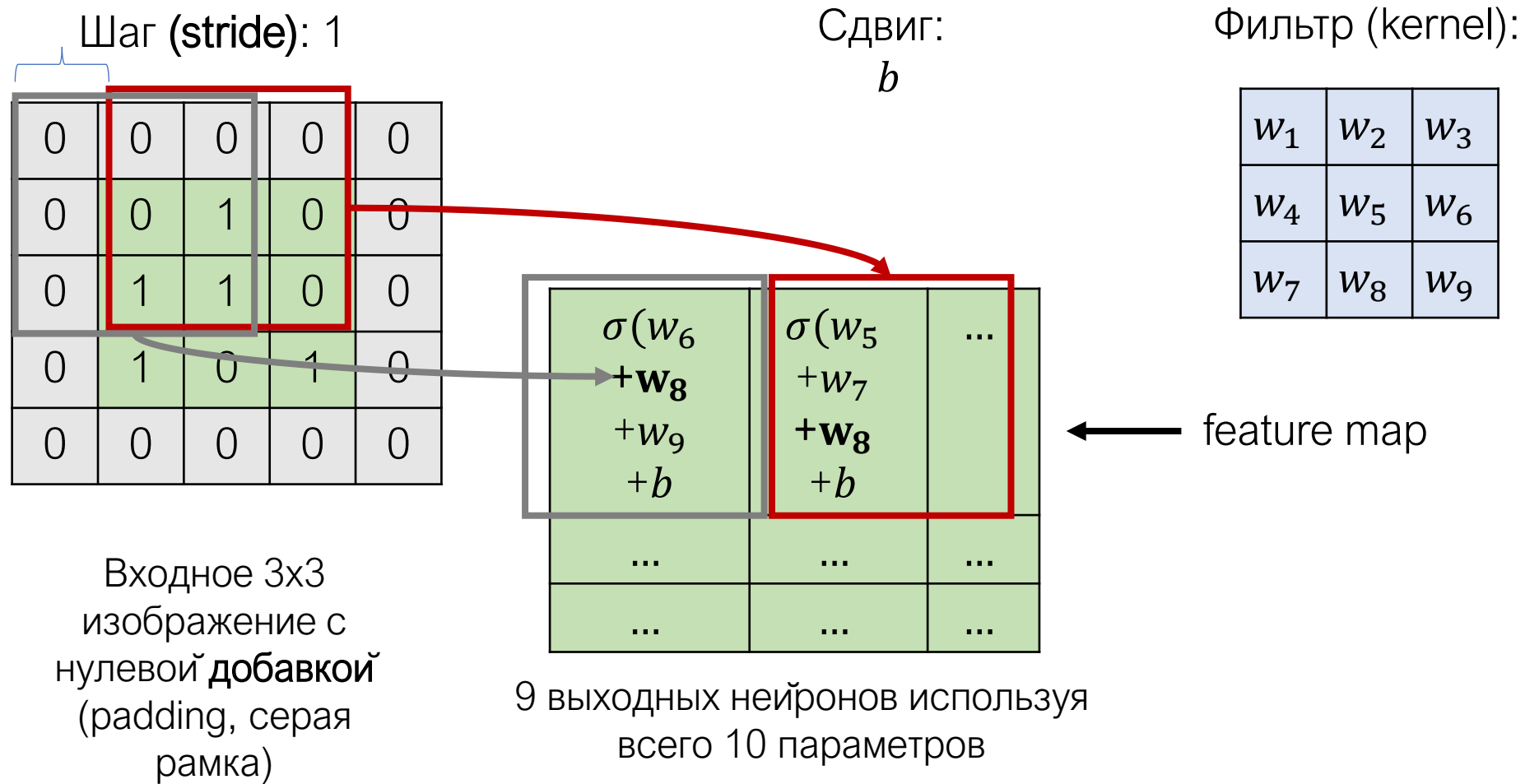
Сверточные нейронные сети

Сверточные нейронные сети - специальная архитектура искусственных нейронных сетей, предложенная Яном Лекунем в 1988 году и нацеленная на эффективное распознавание образов.



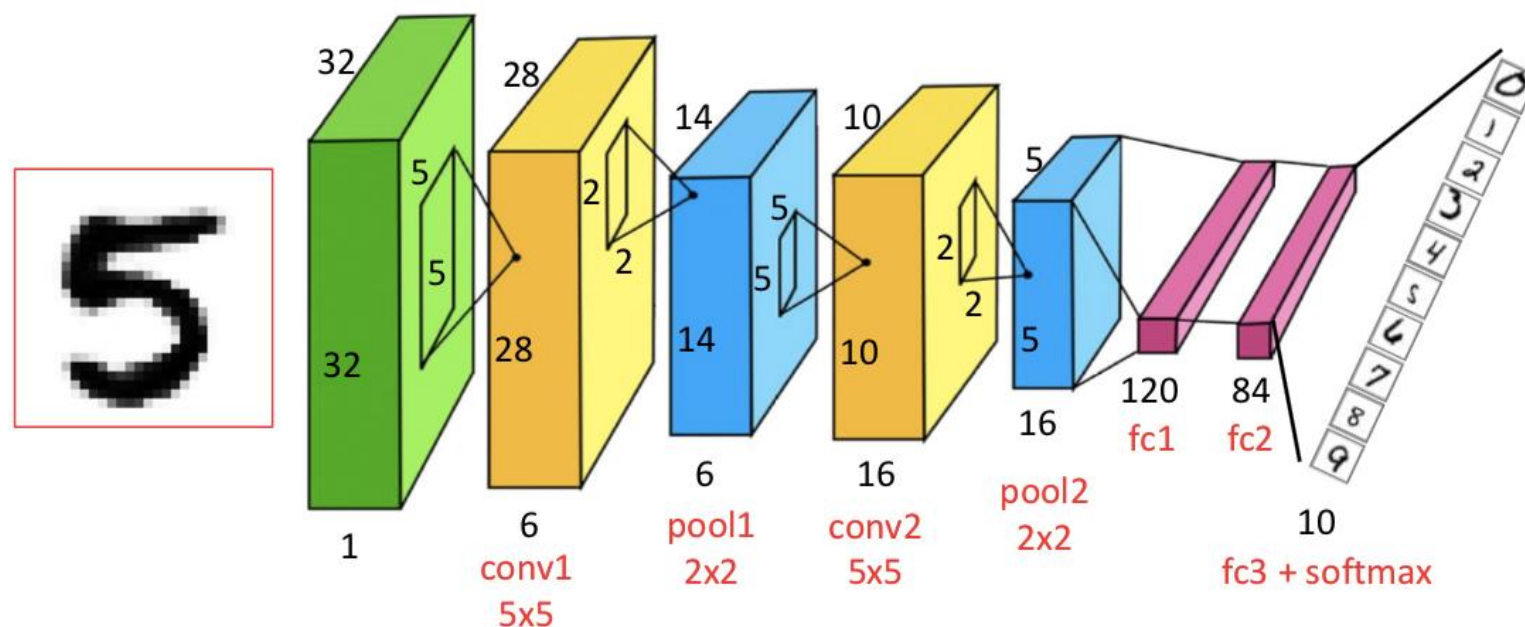
Ян Лекун

Сверточный слой в нейросети



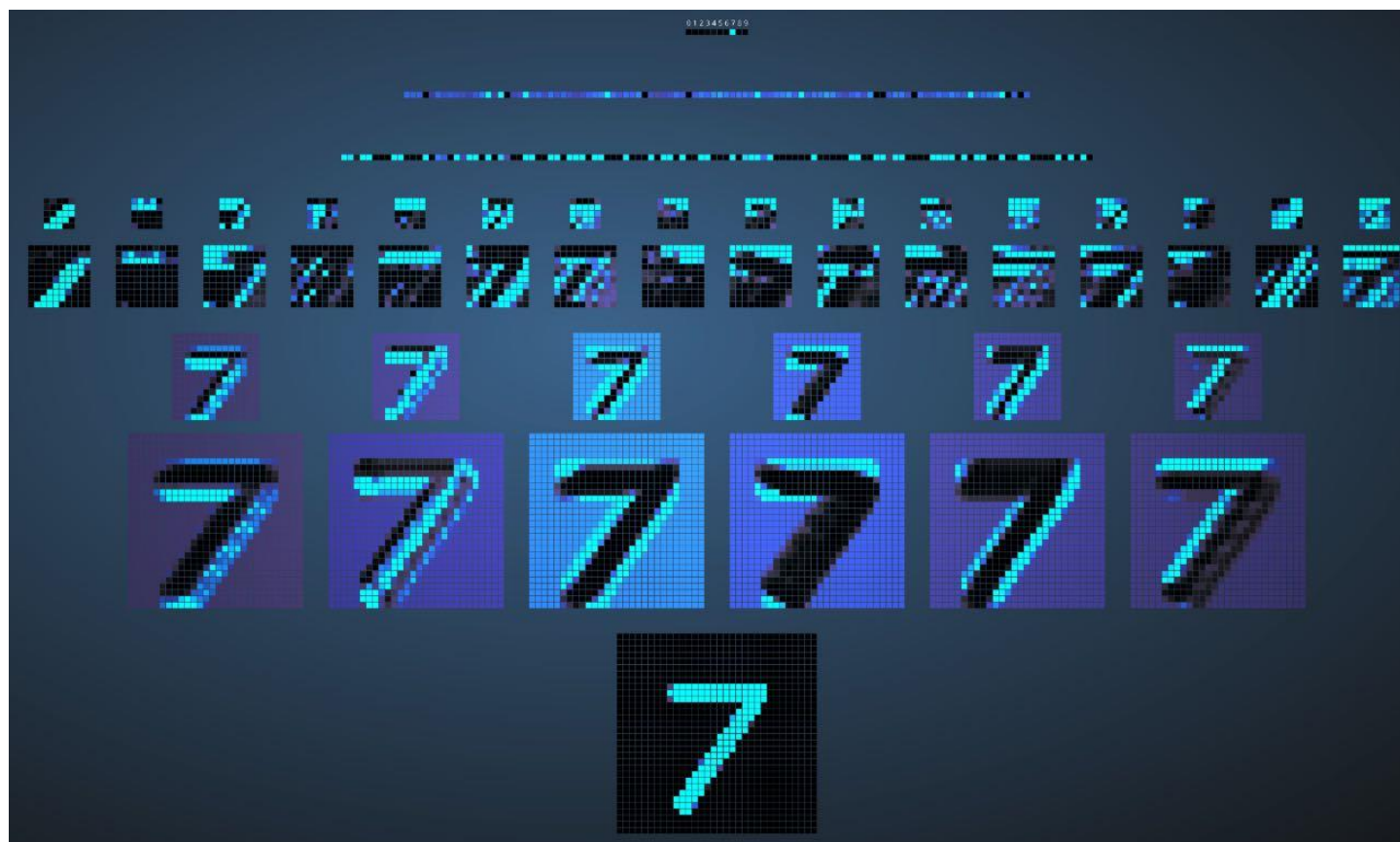
Соберем это все в сверточную сеть

LeNet-5 архитектура (1998) для распознавания рукописных цифр (датасет MNIST):



Демо: визуализация обученной сети для MNIST

<http://scs.ryerson.ca/~aharley/vis/conv/flat.html>



Content-based image retrieval (CBIR)

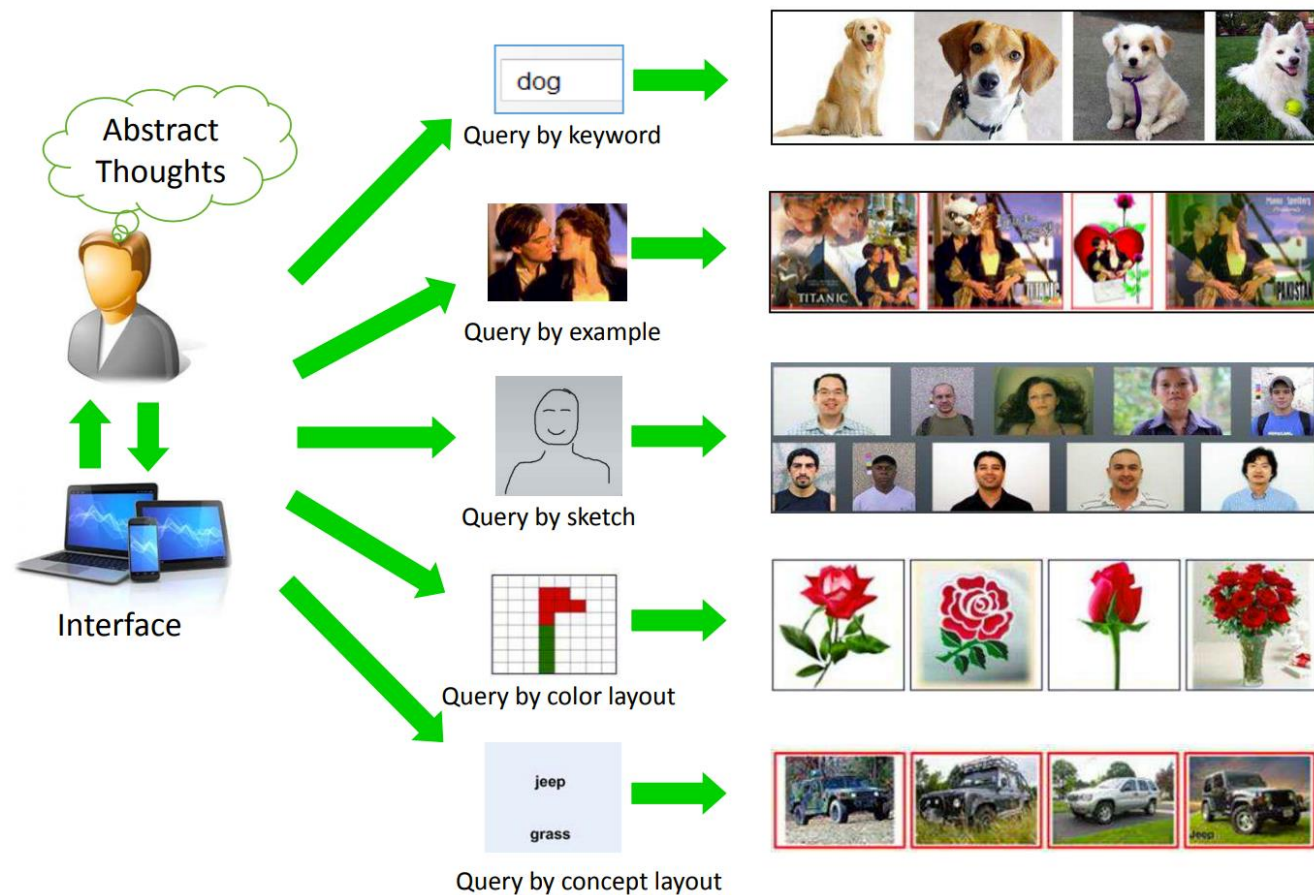


Поиск изображений по содержанию (англ. Content-based image retrieval (CBIR)) — раздел компьютерного зрения, решающий задачу поиска изображений, которые имеют требуемое содержание, в большом наборе цифровых изображений.

Алгоритм поиска должен анализировать содержание изображения, например, цвет представленных на нём объектов, их форму, текстуру, композицию сцены. При отсутствии возможности проанализировать сцену при поиске рассматриваются метаданные: ключевые слова, метки.

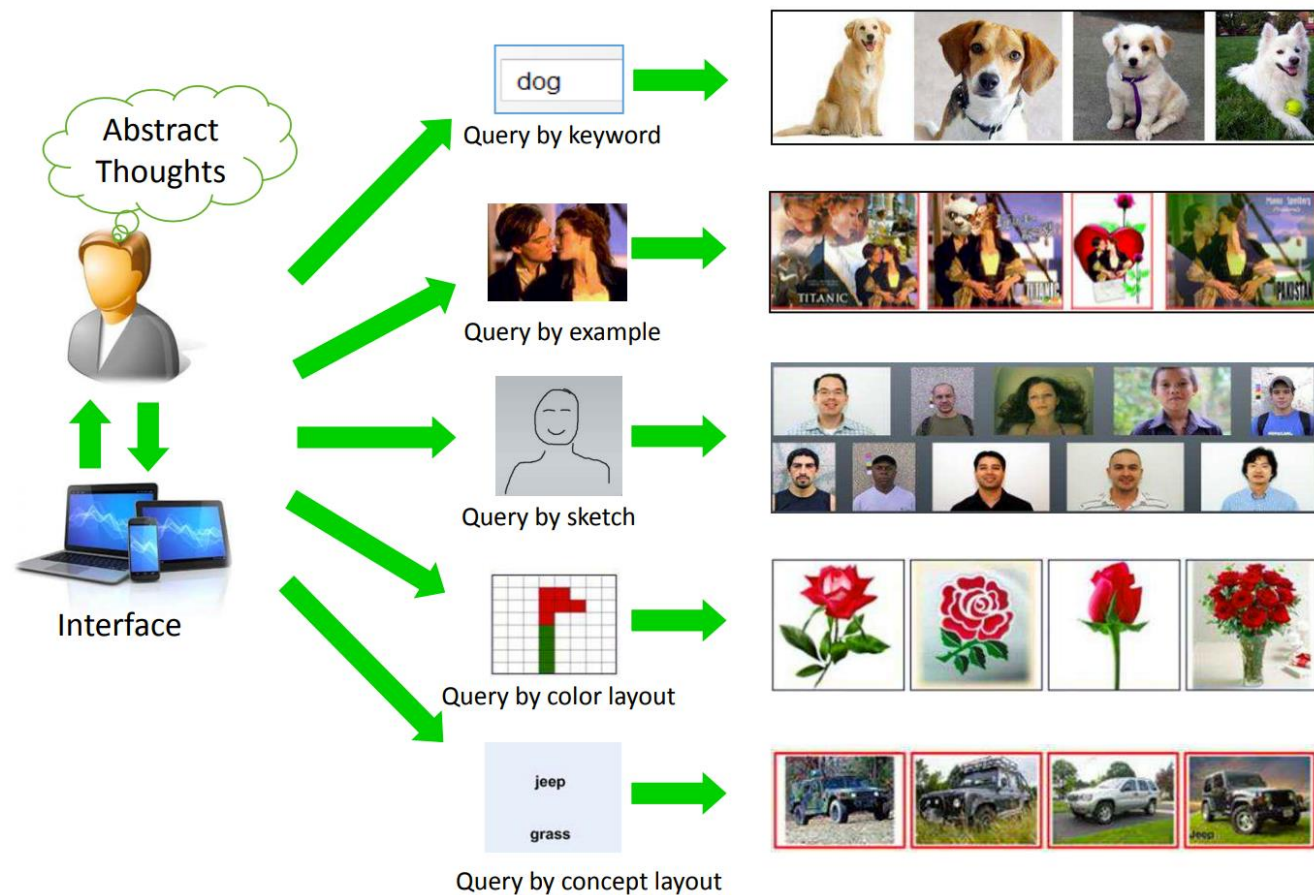
Content-based image retrieval (CBIR)

Поиск изображений по содержанию (англ. Content-based image retrieval (CBIR)) — это любой поиск, в котором участвуют изображения.

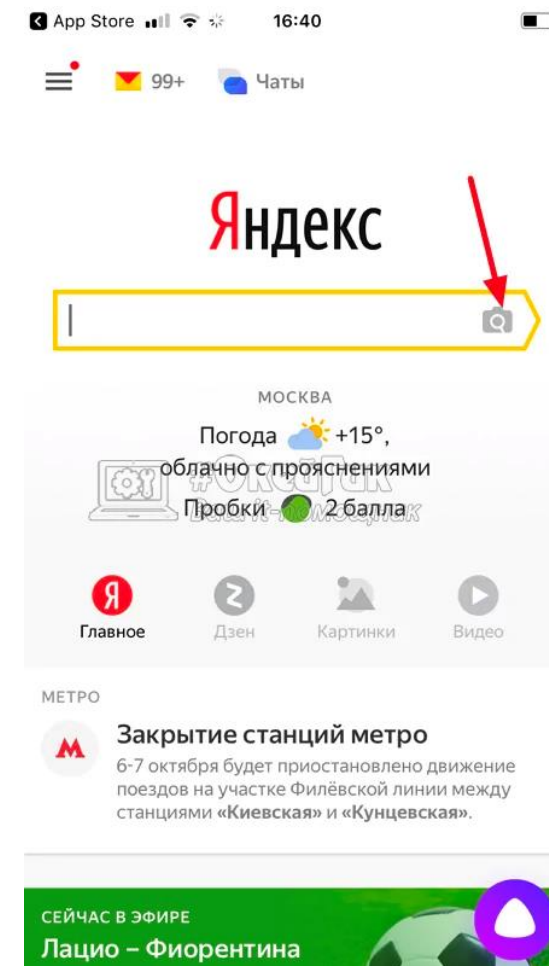
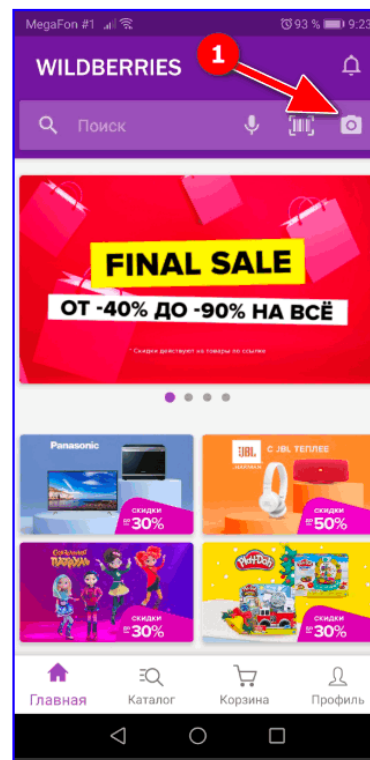
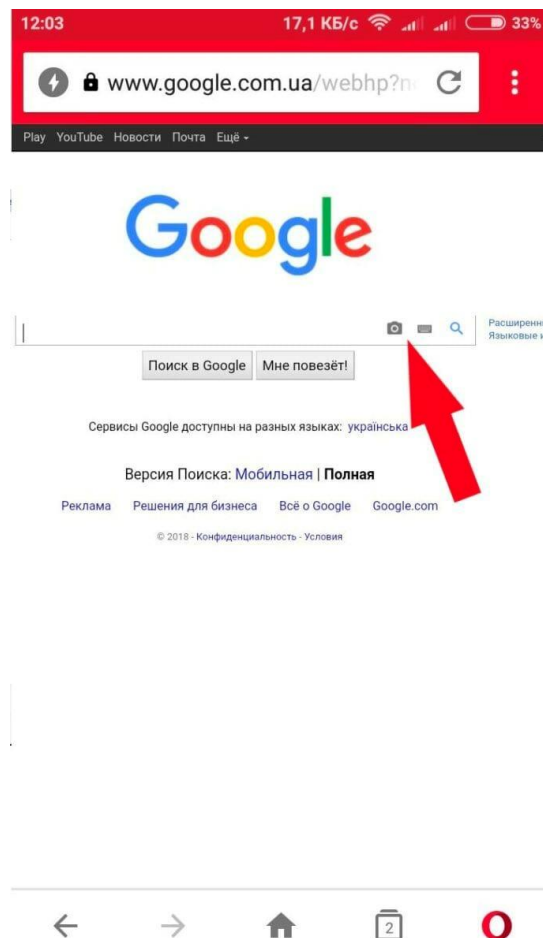


Content-based image retrieval (CBIR)

Алгоритм поиска должен анализировать содержание изображения, например, цвет представленных на нём объектов, их форму, текстуру, композицию сцены. При отсутствии возможности проанализировать сцену при поиске рассматриваются метаданные: ключевые слова, метки.



Примеры использования



Мы берем векторное представление (последний слой).

Что значит «последний слой» в нейронной сети?

Если после него можно применяется линейная функцию (+в сигмоид) – и классификация показывает хорошее качество?

Это означает, что это векторное представление хорошо описывает объект в **латентном пространстве**.

Аудио.

<https://github.com/mdeff/fma>

Постановка задачи

Постановка задачи рекомендации

U – субъекты (пользователи)

I – объекты

r_{ui} = [измеримая функция описывающая взаимодействие u и i]

Теперь мы дополнительно знаем:

\tilde{i} - признаки, которые можем извлечь об *item*

\tilde{u} - признаки, которые можем извлечь о *user*

Задача:

- Восстановить матрицу R
- Найти близкие («похожие») элементы по u
- Найти близкие («похожие») элементы по

Постановка задачи

Если мы говорим про рейтинги – то можно поставить задачи регрессии.

$$(\tilde{i}, \tilde{u}) \rightarrow r$$

Аналогично как с факторизационной машиной:

Feature vector \mathbf{x}																	Target y					
$\mathbf{x}^{(1)}$	1	0	0	...	1	0	0	0	...	0.3	0.3	0.3	0	...	13	0	0	0	0	...	5	$y^{(1)}$
$\mathbf{x}^{(2)}$	1	0	0	...	0	1	0	0	...	0.3	0.3	0.3	0	...	14	1	0	0	0	...	3	$y^{(2)}$
$\mathbf{x}^{(3)}$	1	0	0	...	0	0	1	0	...	0.3	0.3	0.3	0	...	16	0	1	0	0	...	1	$y^{(2)}$
$\mathbf{x}^{(4)}$	0	1	0	...	0	0	1	0	...	0	0	0.5	0.5	...	5	0	0	0	0	...	4	$y^{(3)}$
$\mathbf{x}^{(5)}$	0	1	0	...	0	0	0	1	...	0	0	0.5	0.5	...	8	0	0	1	0	...	5	$y^{(4)}$
$\mathbf{x}^{(6)}$	0	0	1	...	1	0	0	0	...	0.5	0	0.5	0	...	9	0	0	0	0	...	1	$y^{(5)}$
$\mathbf{x}^{(7)}$	0	0	1	...	0	0	1	0	...	0.5	0	0.5	0	...	12	1	0	0	0	...	5	$y^{(6)}$
	A	B	C	...	TI	NH	SW	ST	...	TI	NH	SW	ST	...		TI	NH	SW	ST	...		
	User				Movie					Other Movies rated					Time	Last Movie rated						

Постановка задачи



Если мы говорим про рейтинги – то можно поставить задачи регрессии.

$$(\tilde{i}, \tilde{u}) \rightarrow r$$

Если мы для пользователей не знаем ничего, то, например, можно использовать его представление из факторизационной машины.

После этого мы получаем регрессионную модель на признаках.

$$f(\tilde{i}, \tilde{u}) \rightarrow r$$

ВОПРОС: Какую метрику качества выбрать?

Какую задачу мы решаем?

Можно просто найти «соседей»

Если нам нужно выдать топ-х рекомендаций, то можно применить простой алгоритм:

- Взять элементы с большим рейтингом
- Найти топ-х ближайших соседей по какой-то метрике близости.

$$similarity = \cos(\theta) = \frac{XY}{\|X\| \|Y\|} = \frac{\sum_{i=1}^n x_i y_i}{\sqrt{\sum_{i=1}^n x_i^2 \sum_{i=1}^n y_i^2}}$$

Проблема: у разных компонент (признаков) может быть разный вес. У картинок текста, жанра, режиссера и т.д. (и их латентного представления) может быть разная значимость при выборе.

Ну и напрашивается...

А давайте соединим Content-base и Collaborative filtering.

Так и сделаем но на следующей лекции! :)

Заключение



Давайте вместе напишем заключение:

- Можно легко использовать эмбединги для пользователей и объектов. Решить задачу регрессии над ним.
- Content-based гарантированно улучшает FM.
- Эмбединги нужны не только для рекомендации!
- Можно использовать только content-based (для топ-k товаров) в стиле k-nn.

Семинар: content-based рекомендации

Dataset List

- [MQ2007](#)
- [MQ2008](#)
- [MQ2007-semi](#)
- [MQ2008-semi](#)
- [MQ2007-agg](#)
- [MQ2008-agg](#)
- [MQ2007-list](#)
- [MQ2008-list](#)