Классификация текстов

Рысьмятова Анастасия

ВМК МГУ 417 группа

30.09.2015

Примеры использования

Классификация текстов необходима для:

1. разделения сайтов по тематическим каталогам
2. борьбы со спамом
3. распознавания эмоциональной окраски текстов
4. персонификации рекламы

Постановка задачи

$\mathfrak{C} = \{c_1, ..., c_{|\mathfrak{C}|}\}$ – множество категорий (классов, меток)
$\mathfrak{D} = \{d_1, ..., d_{|\mathfrak{D}|}\}$– множество документов
$\Phi : \mathfrak{C} \times \mathfrak{D} \to \{0, 1\}$– неизвестная целевая функция
Необходимо:
построить классификатор Φ', максимально близкий к Φ.

Этапы

Задача классификации текстов состоит из этапов

1. **Предобработка текста**
   - Удаление редких/частотных слов
   - Делать стэмминг или лемматизацию
2. **Извлечение признаков из текста**
   - TF-IDF
   - n-граммы
3. **Выбор классификатора**
   - Обычно линейные

1. **TF-IDF**
   $\text{tf-idf}(t, d, D) = \text{tf}(t, d) \times \text{idf}(t, D)$
   TF - отношение числа вхождения некоторого слова к общему количеству слов документа.
   $$\text{tf}(t, d) = \frac{n_i}{\sum_k n_k}$$
   IDF - обратная частота документа.
   $$\text{idf}(t, D) = \log \frac{|D|}{|(d_i \supset t_i)|}$$
2. **n-граммы**
   индикаторы того, что данные два слова встретились рядом; для текста «мама мыла раму» получаем биграммы «мама мыла» и «мыла раму»

Проблемы данного подхода

1. высокая размерность пространства
2. большой объем данных
3. разряженность пространства
4. непросто придумать правильные признаки
5. если изменить язык текстов, то нужно решать задачу с нуля

Нейронные сети

Активно используются в сязи появлением:

- больших объемов данных
- больших вычеслительных возможностей

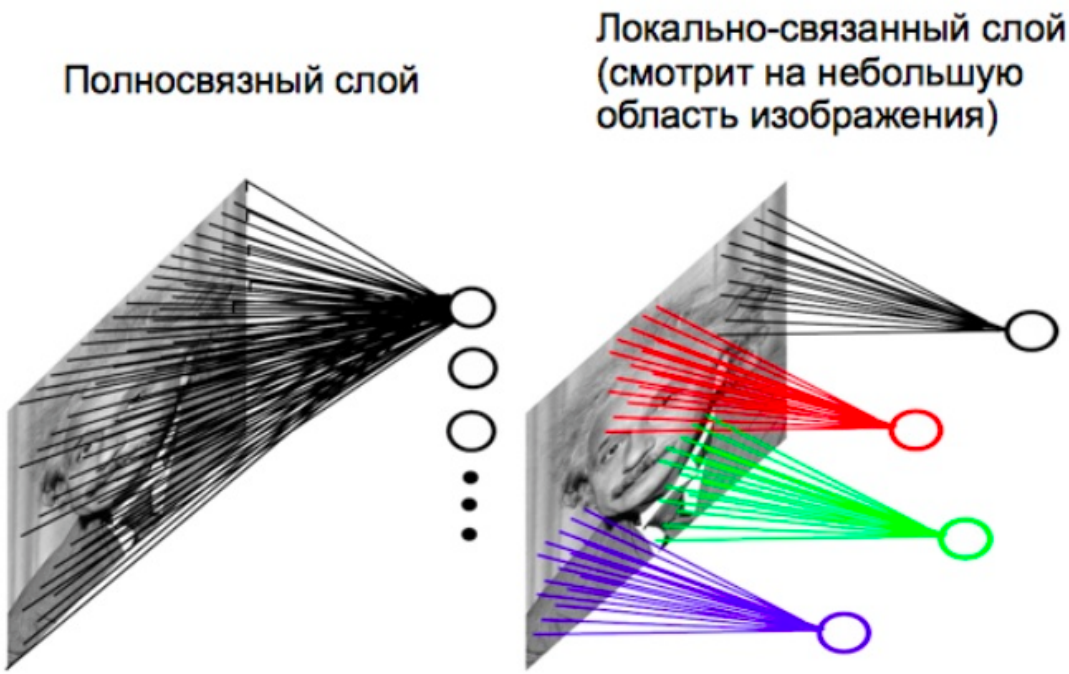
Архитектура сети выбирается таким образом, чтобы заложить априорные знания из предметной области:

- пиксель изображения сильнее связан с соседним (локальная корреляция)
- объект может встретиться в любой части изображения

Сверточные нейронные сети

Локально-связный слой

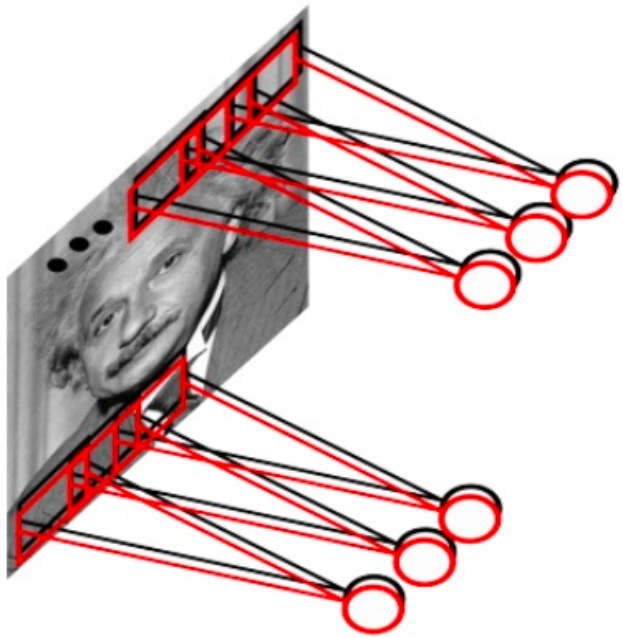
закладывает в архитектуру сети априорное знание о том, что соседние пиксели изображения сильнее связаны между собой



Сверточные нейронные сети

Сверточный слой

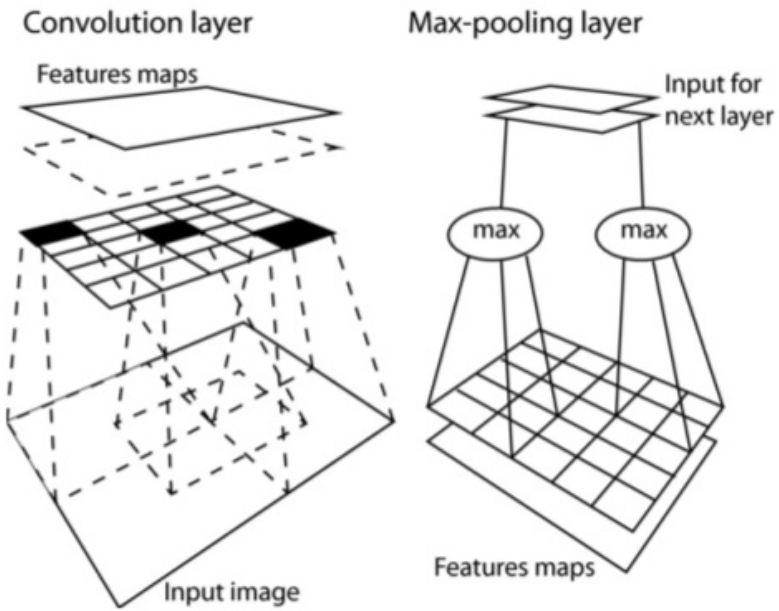
- Это локально-связный слой с одинаковыми весами в разных частях изображения
- Закладывает в сеть априорное знание о том, что объект может встретиться в любой части изображения



Сверточные нейронные сети

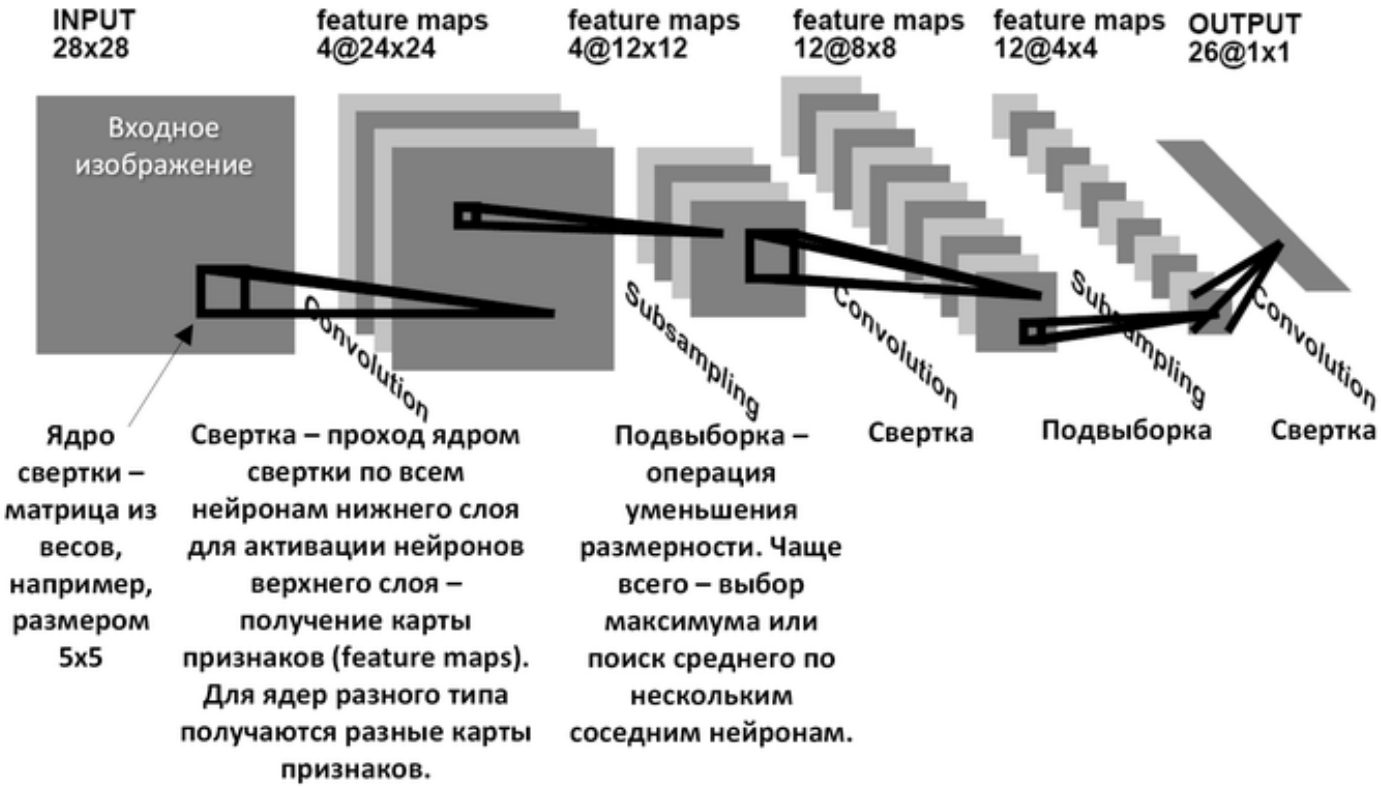
Max pooling

- Аналогичен сверточному слою, в котором операция "+"заменена на "max"
- Добавляет устойчивости к небольшим деформациям



Сверточные нейронные сети

Архитектура сверточной нейронной сети



Использование сверточных нейронных сетей для текстов

Идея

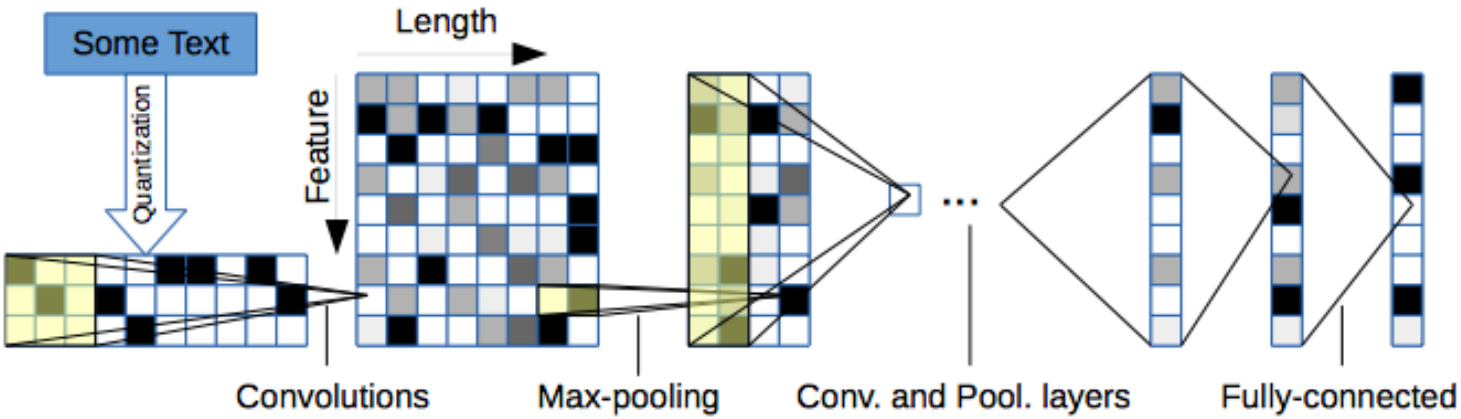
Применить сверточную нейронную сеть к текстам аналогично изображениям, при этом подавать на вход не слова а символы.

Модель нейронной сети

1. Алфавит состоит из m символов
   каждый символ кодируется с помощью 1-m кодировки
2. Из текста выбираем ℓ символов
   Считаем что в этих ℓ символах достаточно информации, чтобы определить класс текста.
3. Выбранные символы записываем в виде матрицы m x ℓ
4. Сеть состоит из сверточного, полного и max-pooling слоя

Модель нейронной сети

$m = 3$
$\ell = 10$
Features на 1 слое $= 9$
Kernel на 1 слое $= 3$



Модель нейронной сети

1. Алфавит состоит из 70 символов ($m = 70$)

```
abcdefghijklmnopqrstuvwxyz0123456789
-,;.!?:'''/\|_@#$%^&*~`+-=<>()[]{}
```

   каждый символ кодируется с помощью 1-m кодировки
2. Из текста выбираем 1014 символов ($\ell = 1014$)

Модель нейронной сети

1. Строим 2 сверточных нейронных сети: малую и большую.
2. В каждой нейронной сети 9 слоев:

   6 - сверточных
   3 - полносвязных
3. Для инициализации весов используем нормальное распределение

   - для большой модели (0, 0.02)
   - для малой модели (0, 0.05)

Модель нейронной сети

Layer	Large Feature	Small Feature	Kernel	Pool
1	1024	256	7	3
2	1024	256	7	3
3	1024	256	3	N/A
4	1024	256	3	N/A
5	1024	256	3	N/A
6	1024	256	3	3

Layer	Output Units Large	Output Units Small
7	2048	1024
8	2048	1024
9	Depends on the problem	

Синонимы

Идея

Заменить некоторые слова из текста их синонимами для устойчивости

Будем использовать словарь из LibreOffice, где для каждого слова или фразы синоним определяется семантической близостью.

- **Какие слова в тексте должны быть заменены?**
  Произвольно выбраем r слов из текста, эти слова необходимо будет заменить. Вероятность числа r определяется геометрическим распределением с параметром $p = 0.5$.
- **Какие синонимов из словаря должны быть использованы для замены?**
  Индекс s выбранного слова также определяется геометрическим распределением, с параметром $q = 0.5$

Другие модели

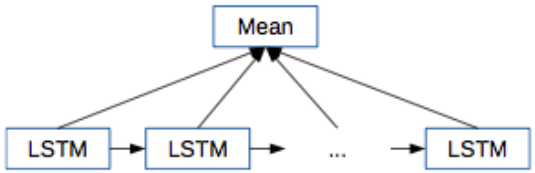
Традиционные методы

1. **Bag-of-words and its TFIDF.** Для каждого набора данных выбраем 50000 наиболее частых слов и используем их TFIDF. Затем используем мультиноминальную логистическую регрессию в качестве классификатора.
2. **Bag-of-ngrams and its TFIDF.** Для каждого набора данных выбраем 500000 наиболее частых ngrams (до 5grams) и используем их TFIDF. Затем используем мультиноминальную логистическую регрессию в качестве классификатора.
3. **Bag-of-means on word embedding.** Используем слова, которые встретились больше 5 раз в выборке. Преобразуем их в вектор с помощью word2vec. Применяем к ним k-means, k =5000. Используем лишь ценроиды, также как в "мешке слов".

Другие модели

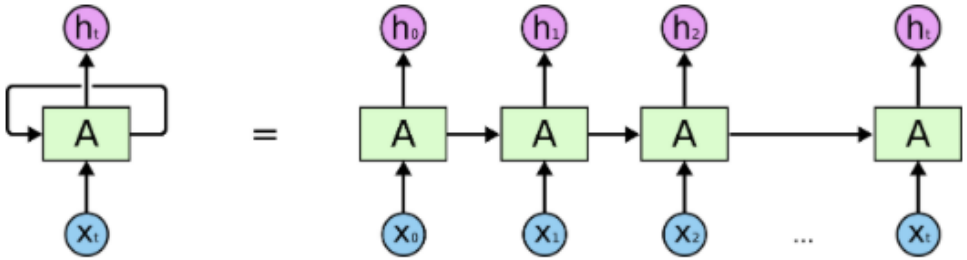
Deep learning методы

1. **Long-short term memory (LSTM).**
   Используя базу слов из word2vec, модель формируется путем взятия среднего значения выходов всех LSTM клеток для формирования вектора признаков. Затем обучаем с помощью мультиноминальной логистической регрессии на этих признаках.



Рекурентная нейронная сеть

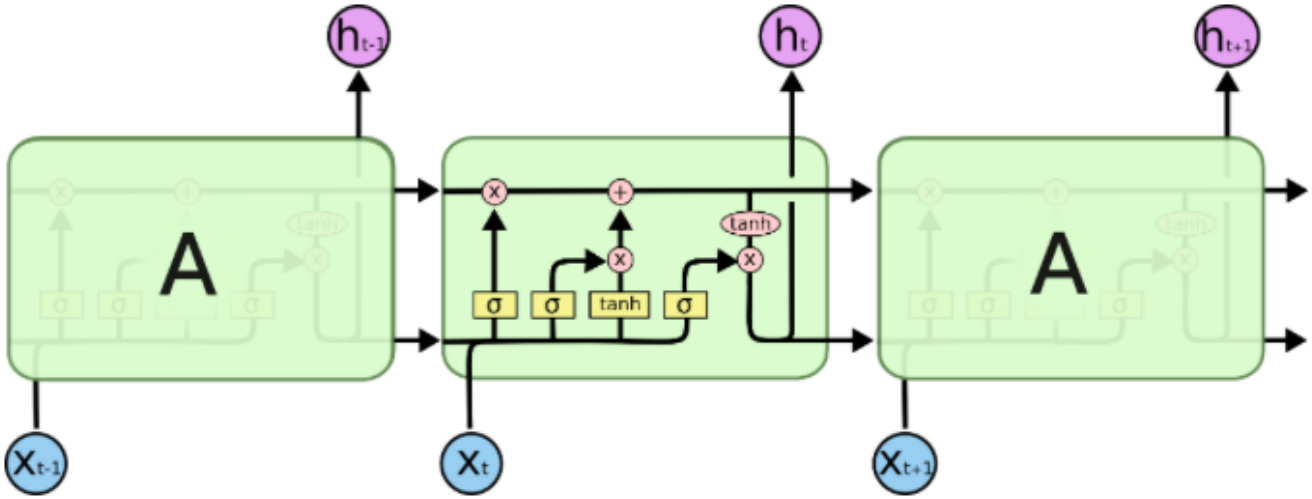
Рекурентная нейронная сеть - представляет собой сети с петлями в них, что позволяет хранить информацию о том, что было в предыдущий момент времени.



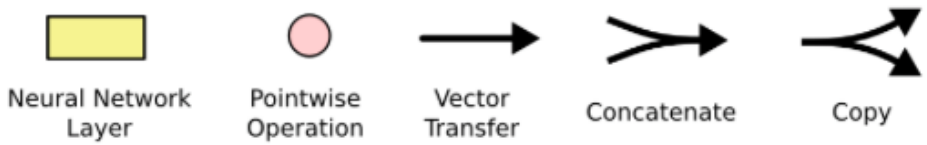
An unrolled recurrent neural network.

LSTM

LSTM - вид рекурентной сети



The repeating module in an LSTM contains four interacting layers.



Данные

Table 3: Statistics of our large-scale datasets. Epoch size is the number of minibatches in one epoch

Dataset	Classes	Train Samples	Test Samples	Epoch Size
AG's News	4	120,000	7,600	5,000
Sogou News	5	450,000	60,000	5,000
DBPedia	14	560,000	70,000	5,000
Yelp Review Polarity	2	560,000	38,000	5,000
Yelp Review Full	5	650,000	50,000	5,000
Yahoo! Answers	10	1,400,000	60,000	10,000
Amazon Review Full	5	3,000,000	650,000	30,000
Amazon Review Polarity	2	3,600,000	400,000	30,000

Результаты экспериментов

Model	AG	Sogou	DBP.	Yelp P.	Yelp F.	Yah. A.	Amz. F.	Amz. P.
BoW	11.19	7.15	3.39	7.76	42.01	31.11	45.36	9.60
BoW TFIDF	10.36	6.55	2.63	6.34	40.14	28.96	44.74	9.00
ngrams	7.96	2.92	1.37	**4.36**	43.74	31.53	45.73	7.98
ngrams TFIDF	**7.64**	**2.81**	**1.31**	4.56	45.20	31.49	47.56	8.46
Bag-of-means	**16.91**	**10.79**	**9.55**	**12.67**	**47.46**	**39.45**	**55.87**	**18.39**
LSTM	13.94	4.82	1.45	5.26	41.83	29.16	40.57	6.10
Lg. w2v Conv.	9.92	4.39	1.42	4.60	40.16	31.97	44.40	5.88
Sm. w2v Conv.	11.35	4.54	1.71	5.56	42.13	31.50	42.59	6.00
Lg. w2v Conv. Th.	9.91	-	1.37	4.63	39.58	31.23	43.75	5.80
Sm. w2v Conv. Th.	10.88	-	1.53	5.36	41.09	29.86	42.50	5.63
Lg. Full Conv.	9.85	8.80	1.66	5.25	38.40	29.90	40.89	5.78
Sm. Full Conv.	11.59	8.95	1.89	5.67	38.82	30.01	40.88	5.78
Lg. Full Conv. Th.	9.51	-	1.55	4.88	38.04	29.58	40.54	5.51
Sm. Full Conv. Th.	10.89	-	1.69	5.42	**37.95**	29.90	40.53	5.66
Lg. Conv.	12.82	4.88	1.73	5.89	39.62	29.55	41.31	5.51
Sm. Conv.	15.65	8.65	1.98	6.53	40.84	29.84	40.53	5.50
Lg. Conv. Th.	13.39	-	1.60	5.82	39.30	**28.80**	40.45	**4.93**
Sm. Conv. Th.	14.80	-	1.85	6.49	40.16	29.84	**40.43**	5.67

Выводы

- Сверточные сети на символьном уровне могут хорошо классифицировать тексты без использования слов. То есть язык можно рассматривать как сигнал.
- На небольших наборах (до нескольких 100 тысяч) данных лучше работают традиционные методы. Когда данных становится больше (более 1 миллиона текстов), лучше работают сверточные нейронные сети на символьном уровне.
- Имеет значение выбранный алфавит.
- Эксперименты еще раз подтверждают, нет ни одного алгоритма машинного обучения, который может работать на всех видах наборов данных.