Создание кастомизированного эмбеддинга слов с помощью нейронных сетей

Рустам Гилаздинов
hsec@ro.ru

29 августа 2017 г.

Обоснование подхода

1. Обработка исходных данных
2. Описание архитектуры
3. Выбор фреймворка
4. Результаты обучения
5. Параметры и их влияние
6. Use cases

Обработка исходных данных

Данные получены с ресурса Kaggle. Необходимо построить свой кастомизированный эмбеддинг слов. Тут сразу же возникает два варианта:

- использовать готовую модель, такую как word2vec или GloVe, и дообучить ее на своих данных. В дальнейшем выяснилось, что такой подход хуже сказывается на метрике качества
- натренировать свои эмбедденги, что в конечном итоге дало лучшее значение метрики, поэтому этот подход остался в финальном решении.

Будем решать задачу регрессии, так как оценки - непрерывные величины. В целом, распределение оценок равномерное, поэтому можно смело использовать mse, т.к. L2-loss должен сработать. Также существует подход - разбить таргет на бины, но мы обойдемся простым случаем. Есть интересная *статья* на эту тему. В описании к данным указано, что некоторые эссе имеют оценки от двух разных групп рецензентов, в связи с этим я усреднил оценки по всем группам.

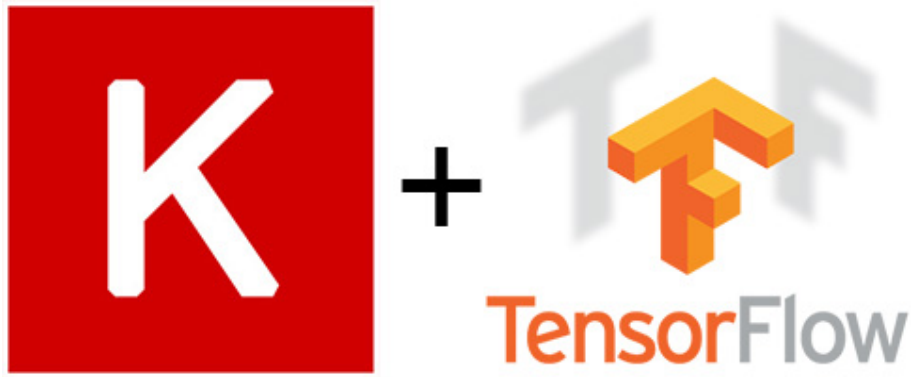
Описание архитектуры

Из специфики задачи было ясно, что нужно в первую очередь попробовать RNN, так как данные текстовые. В совокупности с LSTM-слоем это послужило baseline-решением. Следующей идеей стало следующее - да, у нас есть текстовые данные, и RNN очень хорошо работают с последовательностями, но природа данных такова, что, несомненно, длина эссе повлияла на конечную оценку, но помимо этого, существуют также внутренние характеристики эссе, такие как связность, информативность и целостность текста и т.п. Поэтому было решено использовать архитектуру, скомбинированную из RNN и CNN, что показывает себя как state-of-the-art на многих датасетах. Таким образом, представление текста перед LSTM-слоем является конкатенацией выходов нескольких сверток с разной шириной фильтра и ядра. Вдохновением послужила статья *"A Sensitivity Analysis of Convolutional Neural Networks for Sentence Classification"*.

Описание архитектуры

Рис.: Архитектура сети

Выбор фреймворка



Я использовал стандартный набор - keras и tensorflow

- functional api в керасе
- tensorboard у tensorflow
- относительно готов к продакшену в силу присутствия tf serving

Но еще есть PyTorch, который позволяет описывать граф вычислений не статично, а императивно, что удобно в отладке и экспериментировании. Пока сыроват, но стоит понаблюдать.

Результаты обучения

Так как решается задача линейной регрессии, я использую две самые простые и репрезентивные метрики - это **r2-score** и **коэффициент объясненной дисперсии**. Также, я дополнил их метрикой **quadratic weighted kappa**, которая использовалась в исходной задаче на платформе kaggle.

Метрика	**Значение**
r2-score	0.942
explained variance score	0.944
kappa	0.9688

Таблица: Результаты обучения

Результаты обучения

Пример 1

Some experts are concerned that people are spending to much time on computers, and I believe that they are correct. Most kinds would perfer to be on the computer chatting with friends on line or spending hours on facebook but what is the point of this when there is a whole to explore, ...

Оценка преподователя 11 **Оценка моделью** 10.61144638

Пример 2

Almost everyone is effected by computers in some sort of way. Some people think computers are bad because some people spend too much time on their computers. The thing is computers help us with our everyday life. If we want to...

Оценка преподователя 7 **Оценка моделью** 7.3123

Параметры и их влияние

Здесь хотелось бы отметить следующие моменты

- функция оптимизации - rmsprop лучше всего подходит для RNN
- функция потерь - регрессия, поэтому стандартный mse
- функция активации - использовал relu, лучший вариант
- дропаут - важнейший параметр для регуляризации, но у нас данных не так много, поэтому небольшой. В изображениях, например, жалеть не нужно
- размер батча - меньший батч не дает прироста метрик, а сеть учиться медленнее
- размер фильтра и ядра в свертках - в целом, при добавлении сверточных слоев точность выросла, и добавление разных значений дало ощутимый прирост

Use cases

- кастомизированный эмбеддинг - отличное подспорье в domain-specific задачах обработки естественного языка
- эмбеддингами мы можем описать не только слова, а например, посещения пользователем набора сайтов, и создавать сегменты
- эмбеддинги используются в биоинформатике - для векторизации геномов
- используются в анализе музыки - классические произведения представлены нотами
- используются в генерации последовательностей, причем, допустим, есть вектор с изображениями и их описанием, так вот, по интуиции, эмбеддинги будут способны помогать в классификации изображений, которых не было в обучающей выборке
- в целом, эмбеддинги - универсальный 'репрезентатор' данных, то есть данные разной природы могут использовать одни и те же эмбеддинги