Глубокое обучение и вообще

Соловей Влад

15 декабря 2020 г.

Посиделка 6: Итого

Agenda

- Пердобработка данных
- И советы про дебаг

Agenda

- Пердобработка данных
- И советы про дебаг

Предобработка данных

Непрерывные характеристики

Нормируем

Дискретизация характеристик

Биним на группы. Например, хорошо подходит для характеристики возраст.

Категориальные характеристики

Либо создаем **one-hot encoding** вектор, состоящий из нулей и единицы: $(Dog, Cat) => [(0,1),(1,0),....]$
Либо каждой категории ставим в соотвествие вектор из R^n, который называется обучаемым и называется - **embedding**. Размерность N сами должны выбрать и задать

Циклические характеристики

Такие характеристики, как день недели, месяц, час и т.п. неверно ставить в соответствие вектор и делать их категориальными. Лучше даже оставить непрерывными.

Циклические характеристики

Но у таких характеристик есть особенность - они циклические. И чтобы отобразить эту закономерность и показать, что воскресенье ближе к понедельнику, а среда к четвергу, то кодировать их нужно □□□ и □□□:

$$X_{sin} = \sin(\frac{2 * \pi * x}{max(x)}) \; X_{cos} = \cos(\frac{2 * \pi * x}{max(x)})$$

Циклические характеристики

Например, время суток - 4 часа:

$$\cos(\frac{2\pi}{24} * 4); \sin(\frac{2\pi}{24} * 4);$$

Убеждаемся, что все работает!

Начнем с данных

Смотрим на данные:

1. Они нормализованы
2. Категориальным переменным соответствует верное число класса, которыми они кодируются
3. Лейблы корректны для каждого наблюдения
4. Если есть предобработка данных, либо какаято специфичная функция формирования батча, проконтролируйте, что лейблы попрежнему правильные для каждого наблюдения
5. Выводите ошибки по каждому батчу, например, если это картинка, то каким картинкам присваиваются неверные классы
6. Продебажьте полный pipeline, что на каждом шаге получается с данными: считывание, предобработка, подача в сеть.

Зафиксируйте random seed (везде!)

```
tf.random.set seed(42)
np.random.seed(0)
```

Начинайте с простого

Начинайте с простой предобработки данных. Только после получения логичных результатов начинайте усложнение.

Выход сетки

Проверьте, что в самом начале loss выдает значение $\log(\frac{1}{\_})$ для корректного класса. Если не так, то скорее всего надо исправить инициализацию весов, в основном, править надо самый первый или последний слой.

И ещё

Посмотрите, что обученная модель лучше случайно проинициализированной модели. Либо лучше модели, которая на вход принимает одни нули.

И ещё

Заоверфитьте на одном батче! Проверьте, что значение целевой функции сильно близко к нулю, а точность под 100 %

Don’t be a hero!

Выбирайте такую архитектуру, которая уже дала хороший результат. Берите то, что пишут в статьях, но не сложное. Усложняйте модель постепенно.

И ещё

Используйте советы, которые дают другие. Берите к сведению любые эвристики. Например, что начинайте с Adam и ReLU.

Нет понятия маленького или большого Learning Rate

Есть маленький и большой Learning Rate для Вашей (!) задачи. Обращайте на его начальное значение. Вначале делайте его константным, только после получения логичных результатов вводите политику его изменения.

И ещё

Не переусердствуйте с Dropout, BatchNorm и регуляризации весов.

И ещё

Просто дайте нейронной сети **время**. Не всегда сеть может обучиться быстро.