

1. Что такое проблема долговременной зависимости в рекуррентных сетях?

RNN сети умеют связывать предыдущую информацию с текущей задачей, так, например, знания о предыдущем кадре видео могут помочь в понимании текущего кадра, или если мы хотим предсказать последнее слово в предложении “облака плывут по небу”, нам не нужен более широкий контекст; в этом случае довольно очевидно, что последним словом будет “небу”.

Однако, сеть теряет способность связывать информацию при наличии долговременных зависимостей. Например, допустим, мы хотим предсказать последнее слово в тексте “Я вырос во Франции... Я бегло говорю по-французски”. Ближайший контекст предполагает, что последним словом будет название языка, но, чтобы установить, какого именно языка, нам нужен контекст Франции из более отдаленного прошлого. Таким образом, разрыв между актуальной информацией и точкой ее применения может стать очень большим. К сожалению, по мере роста этого расстояния, RNN теряют способность связывать информацию.

2. В чем преимущество построения модели в функциональном виде в Keras?

При создании модели в функциональном виде мы можем создать модель с несколькими входными или выходными слоями.

Также с функциональным API легко использовать обученные модели: можно обрабатывать любую модель, как если бы она была слоем, вызывая ее для тензора. Функциональный API позволяет легко манипулировать большим количеством переплетенных потоков данных.

3. Какая взаимосвязь между задачей классификации и семантической сегментации?

Семантическая сегментация изображения означает присвоение каждому пикселю определенной метки, изображение разбивается на сегменты, которые являются объектами (человек, машина и т.д.). А в задаче классификации всему изображению ставится в соответствие только одна метка.

То есть задача классификации определяет есть ли объект на изображении, а семантическая сегментация определяет не только есть ли объект на изображении, но и его расположение на нем.

4. Какая в итоге была полученная максимальная точность? Можно ли ее повысить?

В итоге была получена точность $\approx 60\%$. Можно повысить вероятность исключения нейронов на слоях Dropout, а также изменить число нейронов на слоях свертки. Также можно добавить слои субдескретизации и прореживающие слои.

5. Какие параметры принимает MaxPooling2D?

`keras.layers.MaxPooling1D(pool_size=2, strides=None, padding='valid', data_format='channels_last')`

- Целое число или кортеж из 2 целых чисел, факторы, по которым уменьшается (вертикальный, горизонтальный). (2, 2) уменьшит вдвое входные данные в обоих пространственных измерениях. Если указано только одно целое число, одинаковая длина окна будет использоваться для обоих измерений.

- `strides`: Integer или кортеж из 2 чисел, или None. Фактор, по которому снижается. Если None, по умолчанию `pool_size`.

- `padding`: "valid" или "same"

Например, у нас входное изображение формы [2, 3], 1 канал. При "valid" нет отступов на выходе будет форма [1, 1]. При "same" тут здесь мы дополняем изображение до формы [2, 4], а затем применяем максимальный пул, поэтому выходная форма равна [1, 2]

- `data_format`: `channels_last` (по умолчанию) или `channels_first`. `channels_last` соответствует входам с формой, (batch, steps, features) а `channels_first` соответствует входам с формой (batch, features, steps).

6. Активируется ли слой Dropout, если у сети были вызван метод predict? Ответ обоснуйте

Dropout используется только при обучении сети. Он используется только для того, чтобы избежать переобучения. Какой смысл его использовать в predict, если веса уже обучены.

Цитата из документов keras(<https://keras.io/layers/core/#dropout>):

Dropout consists in randomly setting a fraction `rate` of input units to 0 at each update during training time, which helps prevent overfitting.