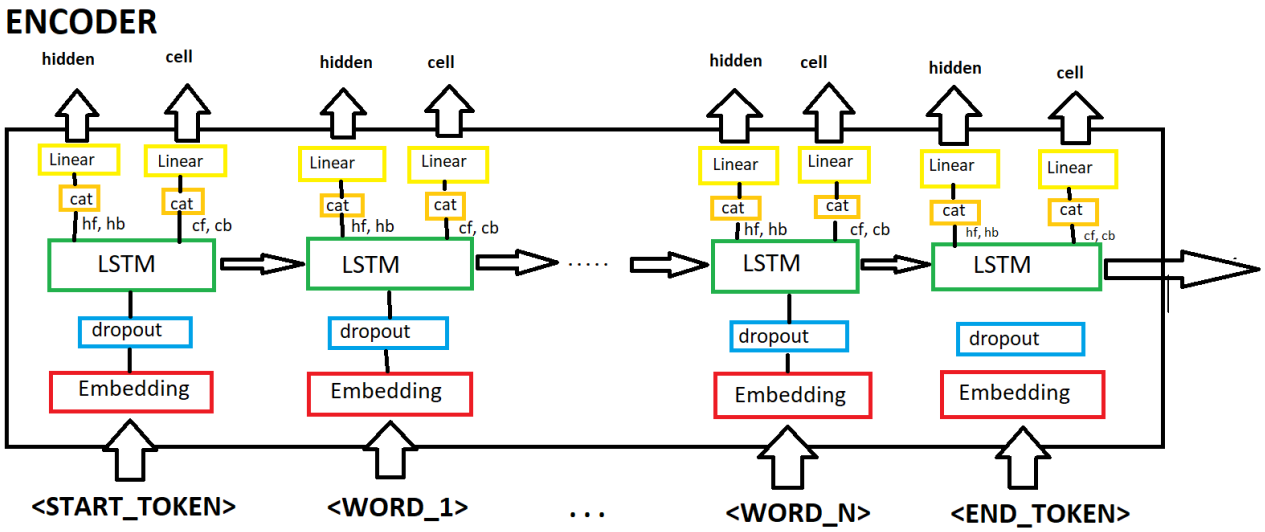
Курсов проект Търсене и извличане на информация

Боян Веселинов Дафов, ФН 82018

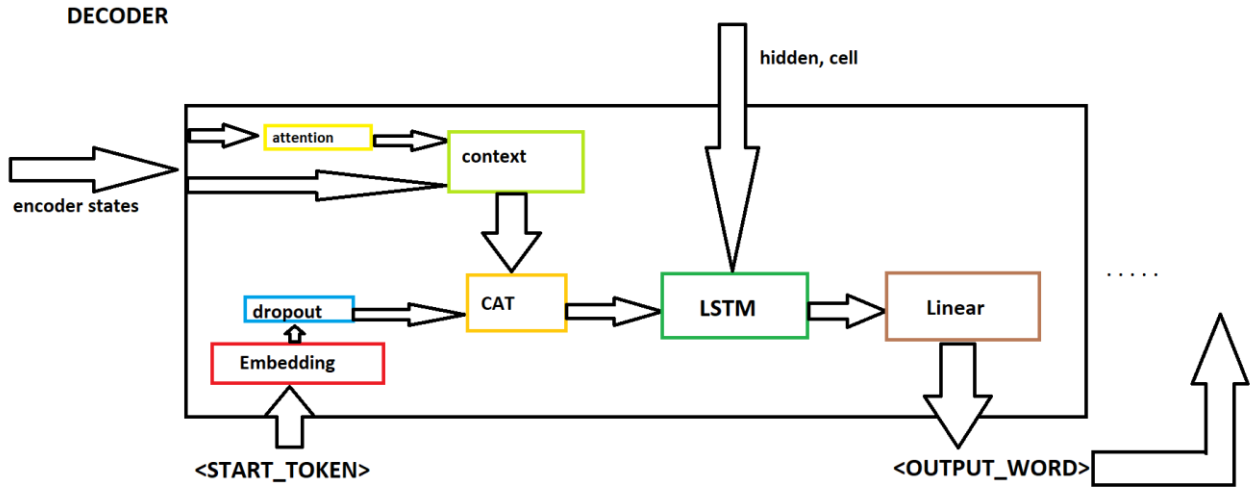
1. Архитектура

Архитектурата реализира Seq2Seq модел за превод от английски към български език. Модела се разделя на два главни модула encoder и decoder. Реализиран е двупосочен encoder, а в Decoder модула е включен и механиъм за внимание (Attention).



Encoder-а е реализиран чрез следните компоненти. Първо е използван Embedding layer чрез който се получават влагания на входната последователност. Следва dropout слой, след който получаваме входа за основния компонент в Encoder-а, а именно двупосочен LSTM слой. Изхода, който получаваме след този слой, е следния. Първо получаваме контекстния изход, който в случая на двупосочен LSTM се състои от hf, cf – получени от последователността отпред-назад и hb, cb – получени от последователността отзад-напред. След това два по два съответно hf, hb и

cf, cb биват конкатенирани в един вектор, който след това е вход към Linear слой. Идеята на този последен Linear слой чисто интуитивно е да „избере“ най-важните компоненти от двете обхождания. Накрая като изход от encoder-а получаваме hidden, cell – представящи контекста и encoder_states – представящи кодирания вход. Именно този изход на encoder-а ще бъде използван по-късно от decoder-а.



Декодирането се осъществява по следната схема. Като вход за генериране на изходна последователност се получават encoder_states – съответсващи на „кодираната“ входна последователност, hidden и cell – представляващи получения контекст при кодиране. Последния компонент от входа на decoder-а е начален token. Чрез входа генерираме следваща дума последния начин. Кодираната последователност (encoder_states) минава през механизъм за внимание (attention) след което чрез комбиниране на полученото “внимание“ с оригиналната кодирана последователност получаваме контекста на кодираната последователност. Независимо от това обработваме последно генерирания token чрез Embedding и Dropout слоеве. Входа на рекурентната мрежа, в случая LSTM, получаваме като конкатенираме получения контекст от кодираната последователност с обработения token, като заедно с това използваме и получените hidden и cell от encoder-а. Предположенията за следващия token получаваме като прекараме изхода от LSTM слоя през Linear слой, който ще ни даде разпределение за следващата дума.

Параметрите, използвани в конкретно обучения от мен модел са следните :

EMBEDDING_SIZE = 32

HIDDEN_SIZE = 512

NUMBER_OF_LSTM_LAYERS = 1

DROPOUT = 0.5

Поради ограничение на изчислителната техника, използвана за трениране на конкратния модел, това са максималните стойности, с които чисто технически е усъществено тренирането. Вероятно при възможност за трениране на „по-голям“ модел биха се получили по-добри резултати.

Основни източници на информация, използвани при реализирането на тази архитектура са :

1. Natural Machine Translation by Jointly Learning to Align and Translate

   By Dzmitry Bahdanau, Kyunghyun Cho, Yosua Bengio

2. Следните youtube видеа от Andrew Ng :

   https://www.youtube.com/watch?v=SysgYptB198

   https://www.youtube.com/watch?v=quoGRI-1l0A

2. Начин на обучение

За обучение е използвана платформата google colab, поради възможността за изполване на графичен процесор. За обучаването на конкратния модел са извършени около 10 епохи. Поради ограничението на изчислителните ресурси, експериментите са проведени с „по-малки“ модели.

3. Оценяване на модела

Перплексията на обучения модел е ~ 30.32