Отчет по БДЗ-2
Глубинное Обучение 1

Панфилов Борис
ФКН ВШЭ

10 марта 2024 г.

1 Начало

В самом начале этого задания нужно было выбить $BLEU \geq 5$.. Поскольку дальше нужно было выбивать больший скор, было очевидно, что так или иначе придется реализовывать архитекруру трансформера. Поэтому я решил сразу сделать это для чекпоинта, несмотря на то, что по утверждениям семинаристов и лектора, рекурентные сети должны заводиться проще.

Таким образом в начале я реализовал обычный трансформер, зафиксировав следующие гиперпараметры.

Для предобработки:

- min-freq = 10 (добавляем в словарь все токены, которые встречаются в тексте больше 10 раз)

Для модели:

- dropout = 0.1
- Размерность embedding'ов = 128
- Размер скрытого слоя в feed forward'е = 256
- Количество голов в attention'е = 4
- Количество слоев в encoder'е и decoder'е = 3

Для оптимизатора:

- beta1 = 0.9
- beta2 = 0.98
- factor = 1
- lr = 0.0001

- warmup = 400

Для обучения обучения:

- Размер батча = 64
- Количество эпох = 10

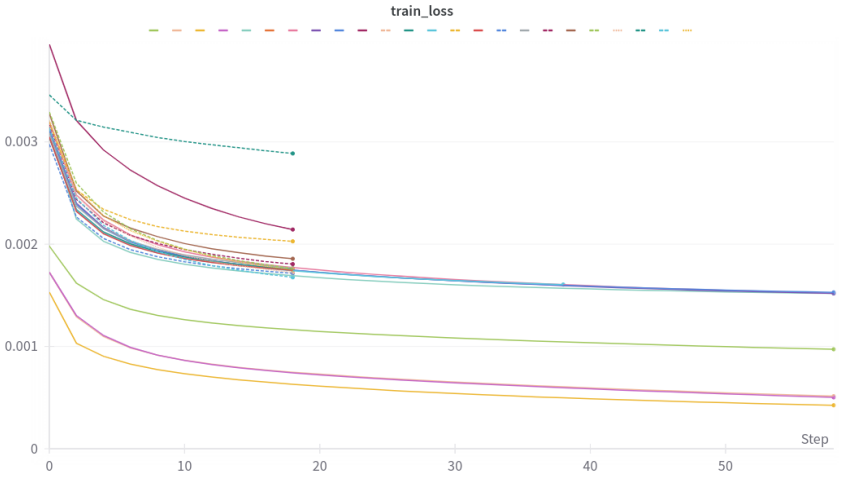
И получил результат BLEU на публичном сплите 22.36. Круто! Но конечно же для финальной сдачи этого мало.

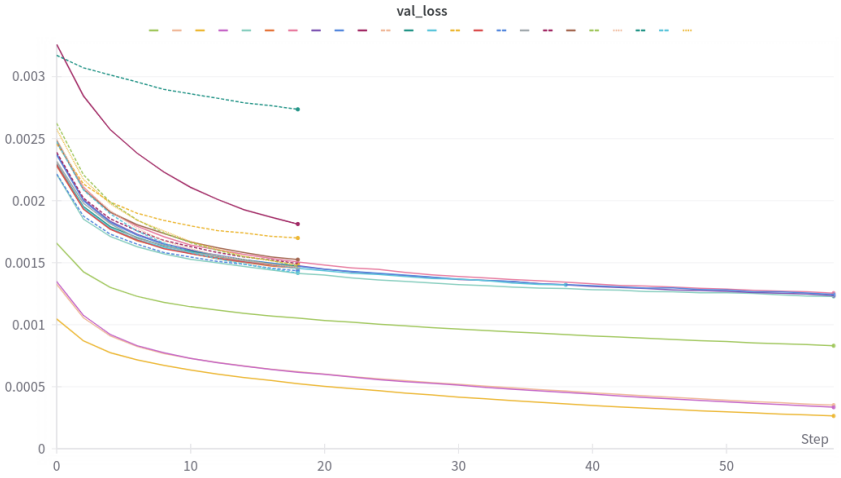
2 Улучшение модели

§2.1 Графики

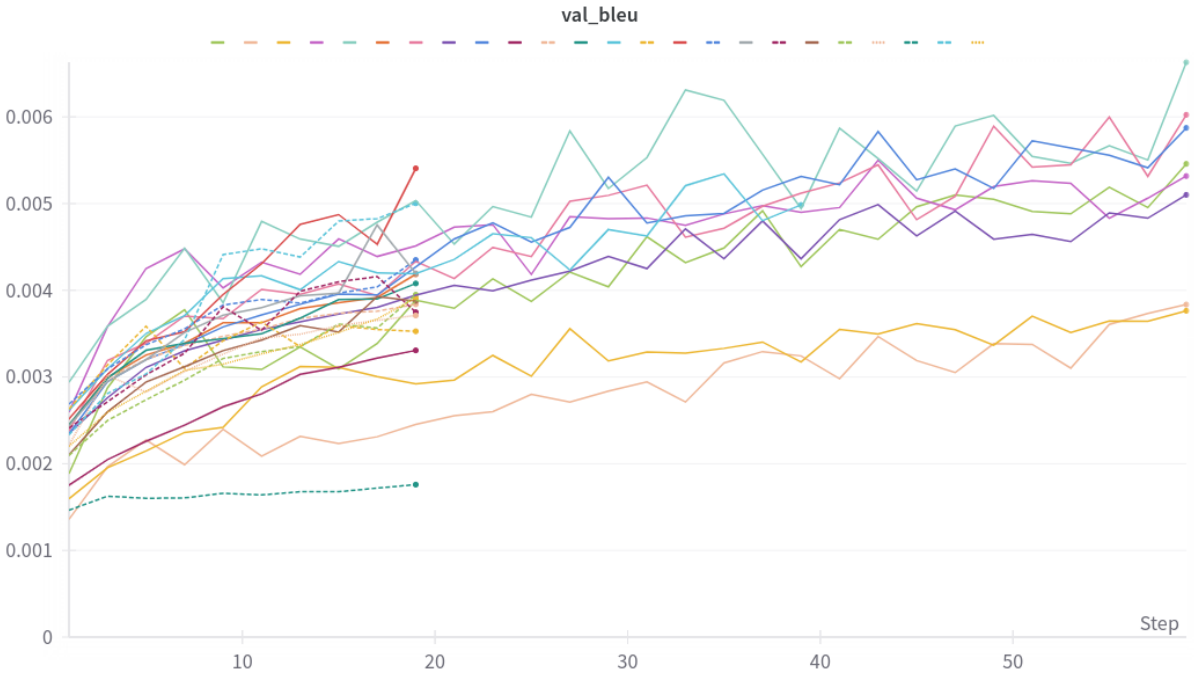
Для проведения экспериментов нужно было начать строить графики и научиться считать BLEU на валидационной выборке, чтобы выработать какую-то интуицию на счет того, как влияют те или иные изменения.

Тем не менее, построив графики лоссов я ненемного разочаровался - они все были абсолютно одинаковой формы, только начинались из различных точек, в силу разницы используемых моделей. А дальше все было абсолютно идентично, поэтому я решил, что графики я буду использовать сугубо для того, чтобы смотреть, что лосс падает, то есть моделька учиться. На рисунках изображениы графики для всех моделей, которые я обучал. Они подтверждают вышесказанное утверждение.

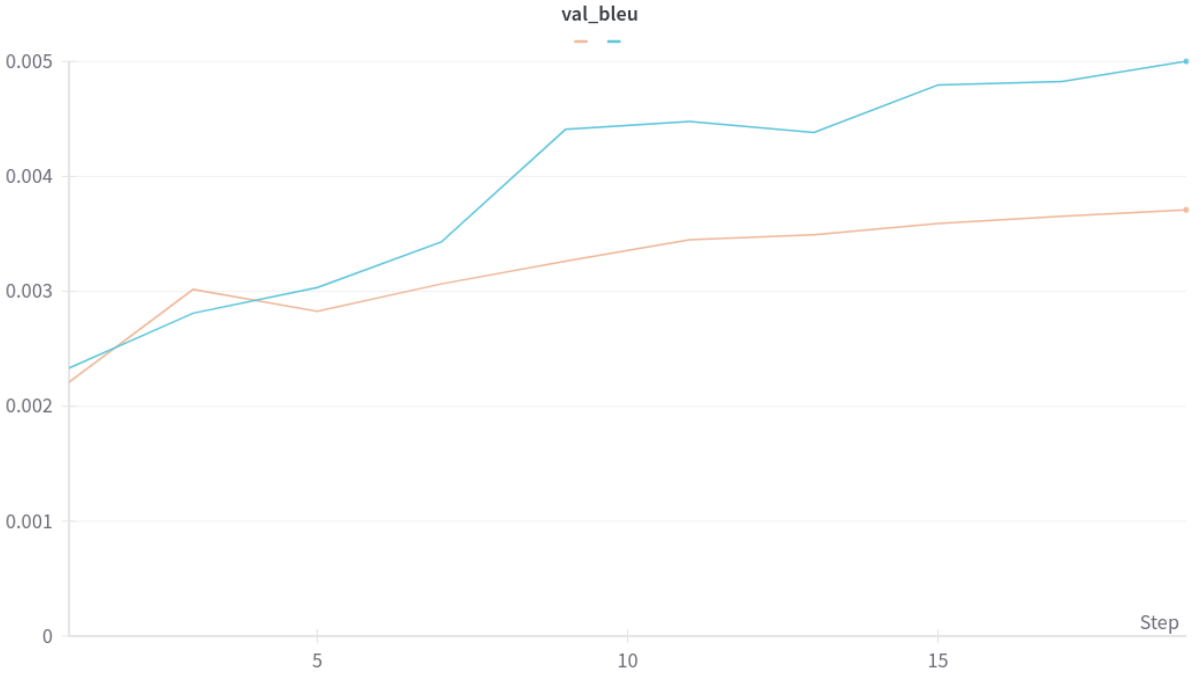




Подсчет BLEU на валидации разочаровал меня еще больше. Графики для всех моделей, которые я обучал выглядят вот так:



И вроде бы все окей - bleu растет, а значит с каждой эпохой наши ответы все больше и больше похожи на правду. Но есть одно но, покажу на примере:



По графику видно, что в конце обучения BLEU у голубой модели в полтора раза больше, чем у рыжей. Тем не менее значение BLEU на публичном сплите у голубой модели 20.54, а у рыжей - 20.87. Это говорит о том, что опираться на валидационное блеу при проведении экспериментов тоже нельзя, потому что оно невсегда понятных образом коррелирует с значением BLEU на тесте. Поэтому к этим графикам я тоже начал относиться, как к признаку того, что модель учиться.

§2.2 Эксперементы

Итого при проведении экспериментов я мог смотреть, что модель и правда учиться, а так же смотреть на ее конечный результат - значение Bleu на тесте и делать из этого какие-то выводы.

Далее буду писать об экспериментах в хронологическом порядке, чтобы сохранялась логика и нить повествования.

На размер модели (в формате Количество слоев в encoder'е и decoder'е_Размерность embedding'ов_Размер скрытого слоя в feed forward'е_Количество голов в attention'е - Значение BLEU на публичном сплите):

- 3_64_256_4 - 19.13
- 3_256_256_4 - 20.87
- 3_256_512_4 - 20.29
- 3_256_1024_8 - 20.54
- 3_512_2048_8 - 3.64
- 3_128_512_8 - 20.84
- 3_128_512_4 - 23.34.

Мне было не очень понятно как делать правильно, потому что перебирать какую-то одну размерность при прочих равных странно, так как когда я начну подбирать другой гиперпараметр, тот который я подобрал первым уже может не подходить. Поэтому я просто насемплировал сколько-то вариантов основываясь на представлении о том как работает мир и посмотрел на результаты. Соответственно после этого эксперимента я взял за основу последнее сочетание - 3_128_512_4.
На параметр токенизации - минимальное количество встречи токена, для его добавления в словарь. Тут хотелось посмотреть как зависит результат от количества неизвестных токенов в итоговом переводе.

- min_freq = 3 - 22.59
- min_freq = 5 - 22.59
- min_freq = 8 - 22.12
- min_freq = 10 - 23.34
- min_freq = 13 - 22.86

Посмотрев на результаты оставил перебираемый параметр равным 10.
Далее я заметил, что 10 эпох может быть маловато и решил увеличить это количество:

- epochs = 10 - 23.34

- epochs = 20 - 24.01
- epochs = 30 - 24.11

Видно, что скор улучшается, но хотелось для экспериментов сильно количество эпох не увеличивать, чтобы не ждать результаты слишком долго. Но уже приятно. На количество слоев в encoder'е и decoder'е:

- 3 - 24.11
- 4 - 22.56
- 5 - 20.16
- 6 = 18.44

Как видим увеличение приводит только к ухудшению качества, поэтому я оставил все как было, то есть по 3 слоя.

На этом моменте мне я устал, пошел посмотрел пару туториалов и поставил параметры, как в одном из них, а именно:
Для модели:

- dropout = 0.1
- Размерность embedding'ов = 512
- Размер скрытого слоя в feed forward'е = 512
- Количество голов в attention'е = 8
- Количество слоев в encoder'е и decoder'е = 3

Для оптимизатора:

- beta1 = 0.9
- beta2 = 0.98
- factor = 1
- lr = 0.0001
- warmup = 400

Для обучения обучения:

- Размер батча = 128
- Количество эпох = 30

Принципиальные различия только в размере модели и размере батча, но данный вариант показал себя лучше, выбив значение BLEU на публичном сплите 25.07. Так что дальше я начал эксперементировать с ним.

Попробовал сделать количество итераций warmup'а равным 1000 - качество слегка ухудшилось и стало 24.48. Решил вообще убрать warmup - стало лучше, значение BLEU 25.7. Попробовал добавить label_smoothing в лосс - не помогло, значение BLEU 22.93.

Далее я решил пописать код, в частности реализовать beam search, потому что в условии задания писали, что пост обработка довольно важна и бууууууум, взяв ширину луча равной 5 получил значительный прирост в качестве со значением BLEU 28.57!

Вот такое получилось приключение, не такое красочное, как в бдз-1 в силу нерепрезентативности графиков, но тоже довольно интересное :-)