ML3 AIM 2023 HW2

Решение контеста на классификацию изображений Tiny Imagenet 200

ilya

Исходный код моделей, трейнера, лернеров, скрипты обучения и всё остальное размещено в репозитории https://github.com/voorhs/tiny-imagenet-clf.

Описание решения

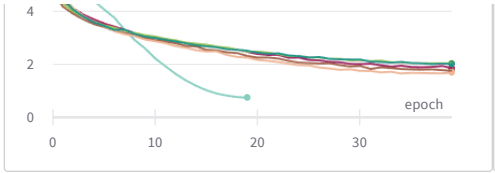
Использованные архитектуры

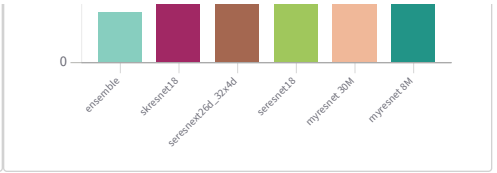
- самопальный ResNet на 8M и 30M параметров
- `seresnet18` и `34` из библиотеки `timm`
- `skresnet18` из библиотеки `timm`
- `seresnext26d_32x4d` из библиотеки `timm`
- ансамбль над этими шестью моделями: линейный слой над конкатенацией выходов сетей

В моделях из `timm` первый слой свёртки 7x7 заменялся на обычную свёртку 3x3, а число выходов менялось с 1000 на 200.

Графики обучения

и затраченное время (в секундах).



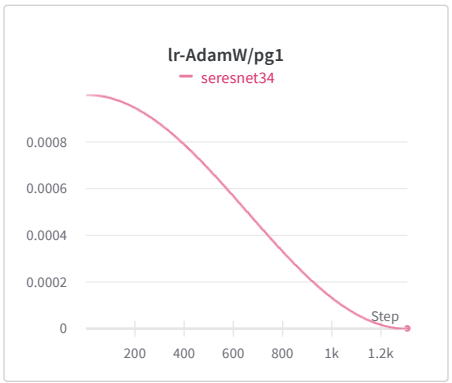


Процедура обучения

Взяты трюки из статьи https://arxiv.org/abs/1812.01187

- аугментация
- инициализация
- label smoothing

Оптимизатор AdamW со стандартными параметрами и cosine annealing от 1e-3 до 1e-6



Все параметры разделены на две группы
https://github.com/karpathy/minGPT/blob/3ed14b2cec0dfdad3f4b2831f2b4a86d11aef150/mingpt/model.py#L136

- веса `nn.Linear` и `nn.Conv2d` --- применяется weight decay
- все остальные параметры --- не применяется weight decay

Фреймворк обучения Lightning. Логгер WandB.

Прочие гиперпараметры

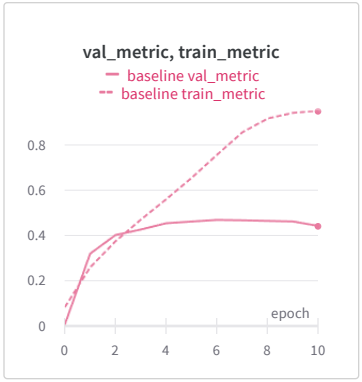
- `batch size` 64
- `max_epochs` 40

Почти все модели обучалась не больше 2.5 часов в колабе. Исключение skresnet (чуть больше 3 часов).

Путь от бейзлайна до решения

Самопальный ResNet

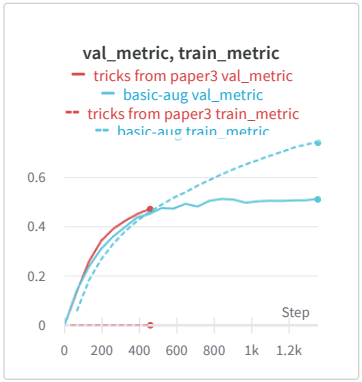
Сперва я сделал простенький resnet и обучил его без аугментаций. Такая модель сильно переобучается и не достигает майлстоуна в 0.5 аккураси на валидации.



Самопальный ResNet + базовые аугментации

Такая модель уже спокойно преодолевает майлстоун, но кривая обучения быстро сатурирует.

Отдельно я заметил, что после добавления трюков из статьи (все кроме аугментаций), обучение становится более стабильным, кривая становится "идеально" ровной и приятной глазу.



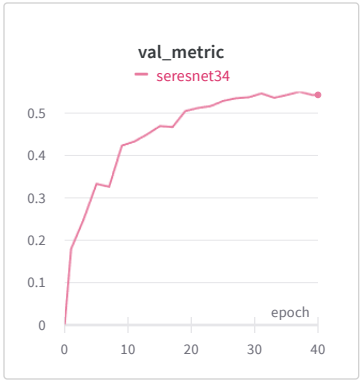
Какие модели рассмотреть

Дальше я решил поизучать, какие есть модели, похожие на ResNet, и остановился на трех

- seresnet
- skresnet

- seresnext

Все следующие эксперименты я стал делать с ними и обнаружил, что со всеми трюками из статьи можно выжать 0.55 аккураси на валидации.



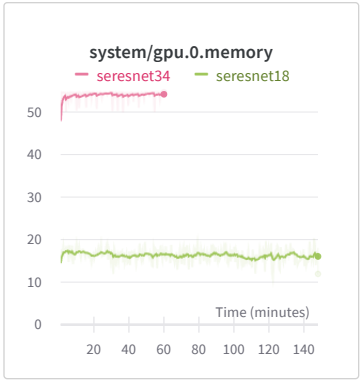
Шесть плохих моделей = ансамбль

Когда я экспериментировал с resnet-like моделями, я использовал кастомный торч датасет, в реализации которого допустил ошибку. Она заключается в том, что при запуске на разных системах классы картинок нумеруются по-разному. Т.е. датасет возвращал разные метки классов когда я запускал у себя локально и в колабе.

В теории можно разобраться, какой логит какому классу соответствует и как-то перегруппировать выходы или внести корректировку в датасет. Но моделей много (шесть штук), а кодит больно, поэтому пришла идея обучить над выходами моделей линейный слой. Ему все равно, в каком порядке компоненты у входного вектора. В итоге получилось аккураси 0.6 на валидации. Причем у использованных моделей максимальное аккураси было около 0.53, т.е. прирост +7% (лучшую модель с аккураси 0.55 я не использовал).

Что меня удивило

- Ансамблирование дало +7%.
- Увеличение числа параметров в 2-3 раза дает буст лишь +2%. Поэтому я не стал наращивать параметры и обучать долго.
- Трюки из статьи действительно сглаживают кривую обучения. Я всегда думал, что лютый тусич графиков это нормально, а оказывается, может быть так красиво.
- В разных ОС разный порядок считывания файлов командой `os.listdir(path)`! Чтобы исправить это, пришлось явно сортировать имена всех объектов в папке.
- С увеличением размера модели скорость обращения к его весам заметно увеличивается. Ниже для моделей 11M и 22M приведён график того, сколько процентов времени ГПУ простаивает в ожидании данных. Поначалу я экспериментировал только с большими моделями, поэтому долго ломал голову, почему так долго простаивает гпу. Убрал все аугментации -- не помогло. Изменил датасет так, чтобы он хранился целиком в ОЗУ -- не помогло. А потом уменьшил модель и понял, что так много времени занимает не обращение к обучающим батчам, а обращение к весам.



- Единственное, что я не понял, это влияние MixUp. Ощущение, что он только привносит нестабильность в обучение. Ибо в первых экспериментах, когда я обучал без него, но с трюками из статьи, кривая обучения была очень приятна глазу. А с ним нет. Убрать его и посмотреть, что будет, я не успел.

Created with ❤️ on Weights & Biases.

https://wandb.ai/ilya_alekseev_2016/lightning_logs/reports/ML3-AIM-2023-HW2--Vmlldzo1ODEzMTA5