Stochastic Training is Not Necessary for Generalization

Докладчик: Михаил Малафеев

Почему все используют SGD вместо GD?

- “зашумленность” градиента
- избегание седловых точек
- способность избегать неоптимальных локальных оптимумов
- дешевле

Как обучать GD в отличие от SGD?

- больше итераций для обучения
- агрессивное ограничение градиента

Поведение в окрестности оптимума

SGD

$$\theta^{k+1} = \theta^k - \tau_k \underbrace{\nabla L(\theta^k)}_{\text{full loss gradient}} + \tau_k \left(\underbrace{\frac{1}{|X|} \sum_{x \in X} \nabla \mathcal{L}(x, \theta^k) - \frac{1}{|B|} \sum_{x \in B} \nabla \mathcal{L}(x, \theta^k)}_{\text{gradient noise } g_k} \right).$$

Шум градиента - гауссовская случайная величина

Σt - ковариация в момент времени t, обратно пропорциональна размеру батча. Отвечает за плоскостность в оптимуме

Wt - Броуновское движение, моделирующее шум градиента

$$\mathrm{d}\theta_t = -\nabla \left(L(\theta_t) + \frac{\tau}{4} ||\nabla L(\theta)||^2 \right) \mathrm{d}t + \sqrt{\tau \Sigma_t}\, \mathrm{d}W_t,$$

Проблема обучения с большим батчем

- Даже при дорогостоящем подборе гиперпараметров и скорости обучения обычный GD сходится за большее количество шагов в сравнении с SGD
- Обобщающая способность SGD теряется с увеличением размера батча и достигаемые оптимумы обычно “острее”.

$$L(\theta) + \frac{\tau}{4|\mathcal{B}|} \sum_{B \in \mathcal{B}} \left\| \frac{1}{|B|} \sum_{x \in B} \nabla \mathcal{L}(x, \theta) \right\|^2$$

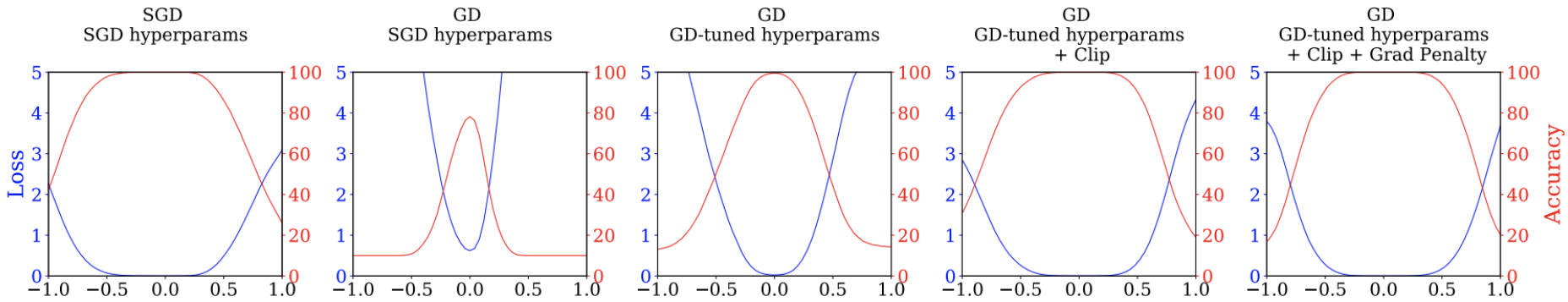
Функция потерь с регуляризацией для SGD

Сравнение SGD и GD. Оптимизировано под SGD

Source of Gradient Noise	Batch size	Val. Accuracy %
Sampling without replacement	128	$95.70(\pm 0.11)$
Sampling with replacement	128	$95.70(\pm 0.05)$
Sampling without replacement (fixed across epochs)	128	$95.25(\pm 0.07)$
Additive $n = 0.01$	50'000	$61.41(\pm 0.09)$
Multiplicative $m = 0.01$	50'000	$79.25(\pm 0.14)$
-	50'000	$75.42(\pm 0.13)$

Сравнение SGD и GD. ResNet-18

Experiment	Mini-batching	Epochs	Steps	Modifications	Val. Accuracy %
Baseline SGD	✓	300	117,000	-	95.70(±0.11)
Baseline FB	✗	300	300	-	75.42(±0.13)
FB train longer	✗	3000	3000	-	87.36(±1.23)
FB clipped	✗	3000	3000	clip	93.85(±0.10)
FB regularized	✗	3000	3000	clip+reg	95.36(±0.07)
FB strong reg.	✗	3000	3000	clip+reg+bs32	95.67(±0.08)
FB in practice	✗	3000	3000	clip+reg+bs32+shuffle	95.91(±0.14)



Сравнение SGD и GD. ResNet-18. Без аугментаций

Обучение с GD при таком эксперименте является стабильней, чем SGD:

- GD не требует подбора новых гиперпараметров модели при ограничении градиента и регуляризации (89.17% acc)
- SGD без подбора гиперпараметров модели показывает себя хуже (84.32%). С подбором – 90.07%

Experiment	Fixed Dataset	Mini-batching	Steps	Modifications	Val. Accuracy
Baseline SGD	CIFAR-10	✓	$117,000$	-	$84.32(\pm 1.12)$
Baseline SGD*	CIFAR-10	✓	$117,000$	-	$90.07(\pm 0.48)$
FB strong reg.	CIFAR-10	✗	3000	clip+reg+bs32	$89.17(\pm 0.24)$
Baseline SGD	$10\times$ CIFAR-10	✓	$117,000$	-	$95.20(\pm 0.09)$
FB	$10\times$ CIFAR-10	✗	3000	-	$88.44(-)$
FB strong reg.	$10\times$ CIFAR-10	✗	3000	clip+reg+bs32	$95.11(-)$

Сравнение SGD и GD. ResNet-18.
Фиксированные аугментации.

Experiment	Fixed Dataset	Mini-batching	Steps	Modifications	Val. Accuracy
Baseline SGD	CIFAR-10	✓	$117,000$	-	$84.32(\pm 1.12)$
Baseline SGD*	CIFAR-10	✓	$117,000$	-	$90.07(\pm 0.48)$
FB strong reg.	CIFAR-10	✗	3000	clip+reg+bs32	$89.17(\pm 0.24)$
Baseline SGD	$10\times$ CIFAR-10	✓	$117,000$	-	$95.20(\pm 0.09)$
FB	$10\times$ CIFAR-10	✗	3000	-	$88.44(-)$
FB strong reg.	$10\times$ CIFAR-10	✗	3000	clip+reg+bs32	$95.11(-)$

Выводы

- GD может достигать таких же значений качества, что и SGD, но при этом GD требуется больше итераций до сходимости в “хорошие” оптимумы
- SGD не обязателен для обобщения модели
- SGD не может сам по себе объяснять хорошее обобщение модели

Ревьюер (Артем Стрельцов)

В статье описана роль SGD и GD в аспекте обобщающей способности модели. В частности, авторы сравнивают SGD с full-batch GD вкупе с применением явной регуляризации к последнему. На примере нескольких архитектур и CIFAR-10 авторы показывают, что full-batch подход может достичь результатов SGD, если применить к нему ряд улучшений (регуляризация, клиппинг, увеличенный LR и так далее).

В результате авторы подкрепляют свои предположения экспериментами, где можно наблюдать, что GD с правильной регуляризацией действительно доходит до качества SGD.

Ревьюер (Артем Стрельцов)

Положительные стороны:

- Статья хорошо написана, в аппендиксе много подробных экспериментов.
- В целом, авторы учли все замечания после публикации и опубликовали вторую версию статьи – сделали более понятный абстракт, добавили еще экспериментов, а также обновили заключение, с оговоркой, что обновление на полном батче все еще сильно хуже по производительности: “Nonetheless, our training routine is highly inefficient compared to SGD (taking far longer run time), and stochastic optimization remains a great practical choice for practitioners in most settings.”
- Большая ценность, хоть и больше эвристик, чем теоретических обоснований: авторы показывают, что в целом обобщающая способность модели с обычным GD и явной НЕстохастической регуляризацией примерно сравнима с SGD, где эта регуляризация неявная

Ревьюер (Артем Стрельцов)

Отрицательные стороны:

- В первой версии казалось, будто авторы считают, что GD в данном случае предпочтительнее SGD, в новой версии это поправили, с оговоркой про скорость при использовании GD, но не указали, насколько это все же дольше.
- Не очень понятно, как сложится картина на других (например, больших) наборах данных – в статье это не описано.
- Также из статьи не очень ясно, можно ли добавить какую-нибудь явную регуляризацию поверх SGD, чтобы улучшить модель.
- Не понятно, почему увеличение learning rate в два раза дает результат лучше, нет какого-то явного обоснования

Ревьюер (Артем Стрельцов)

Воспроизводимость:

- Просто с точки зрения кода, в целом авторы статьи достаточно хорошо расписали подробности экспериментов, и выглядит, как будто их нетрудно реализовать, но
- Это все тяжело с точки зрения необходимости иметь достаточные вычислительные мощности

Ревьюер (Артем Стрельцов)

Итог:

Оценка – 7

Уверенность – 4

Хакер

Гольдман Артур

Идея

В статье исследуются модели, применимые в задаче компьютерного зрения. Стало интересно попробовать обучить задачу из другой области с помощью Full GD.

Хотелось выбрать несложные задачи: задача легко формулируется, обучаемая сеть содержит небольшое количество параметров.

Задача 1. Классификация имен

Первый эксперимент построен по мотивам официального туториала: https://pytorch.org/tutorials/intermediate/char_rnn_classification_tutorial.html

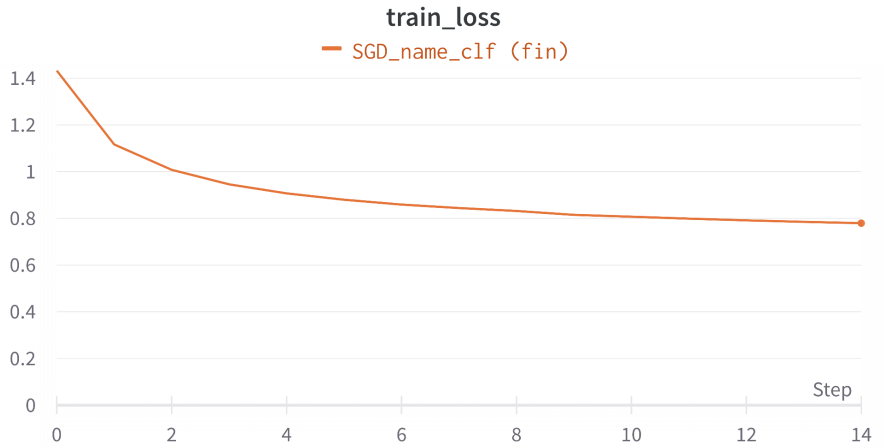
Задача: классифицировать имя в один из 18-ти языков

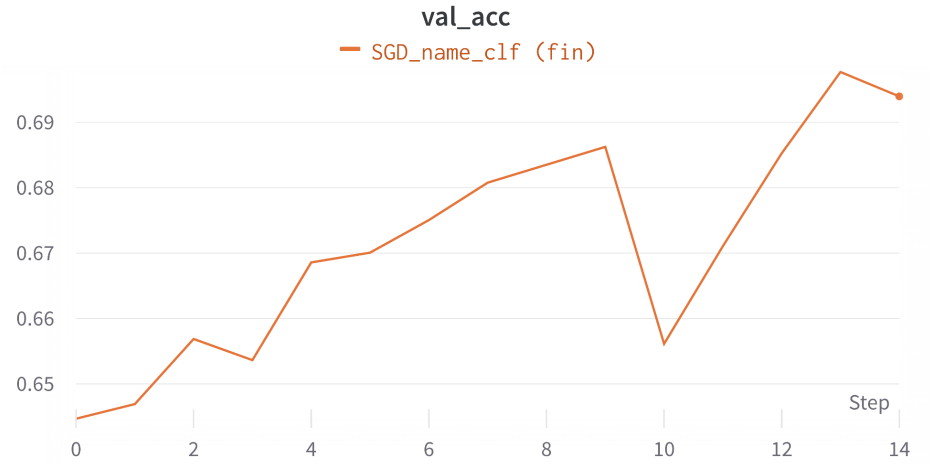
Данные: ~20 тыс. имен (16 тыс на обучение, 4 на валидацию, разбиваем в соответствии с балансом классов)

Сеть: RNN, ~ 27 тыс параметров

Классификация имен. SGD Baseline

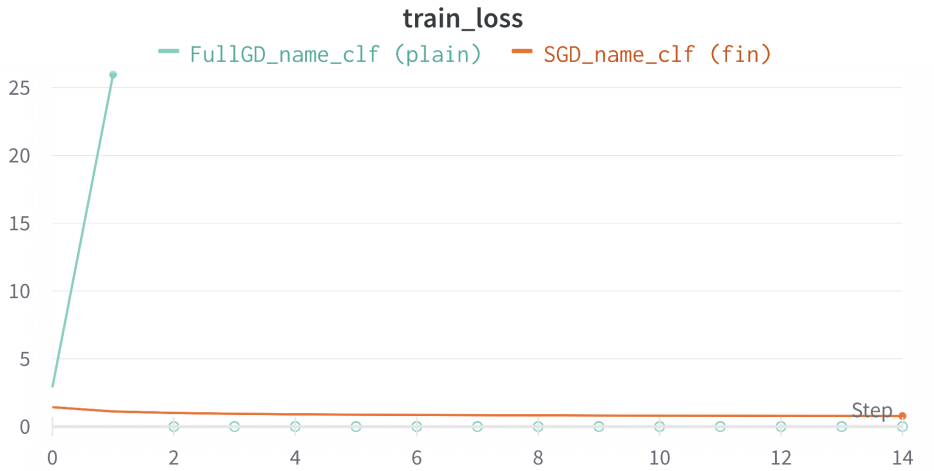
Запуск немного модифицированного туториала на 15 эпох: качество на валидации ~ 0.69

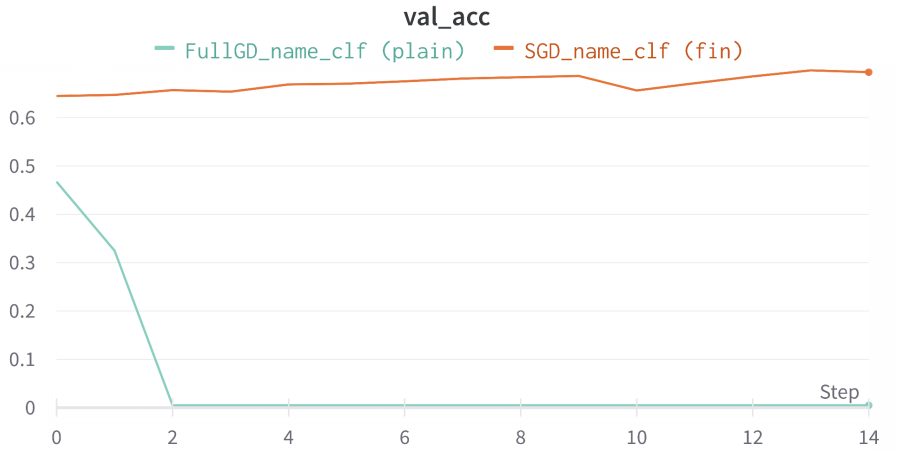




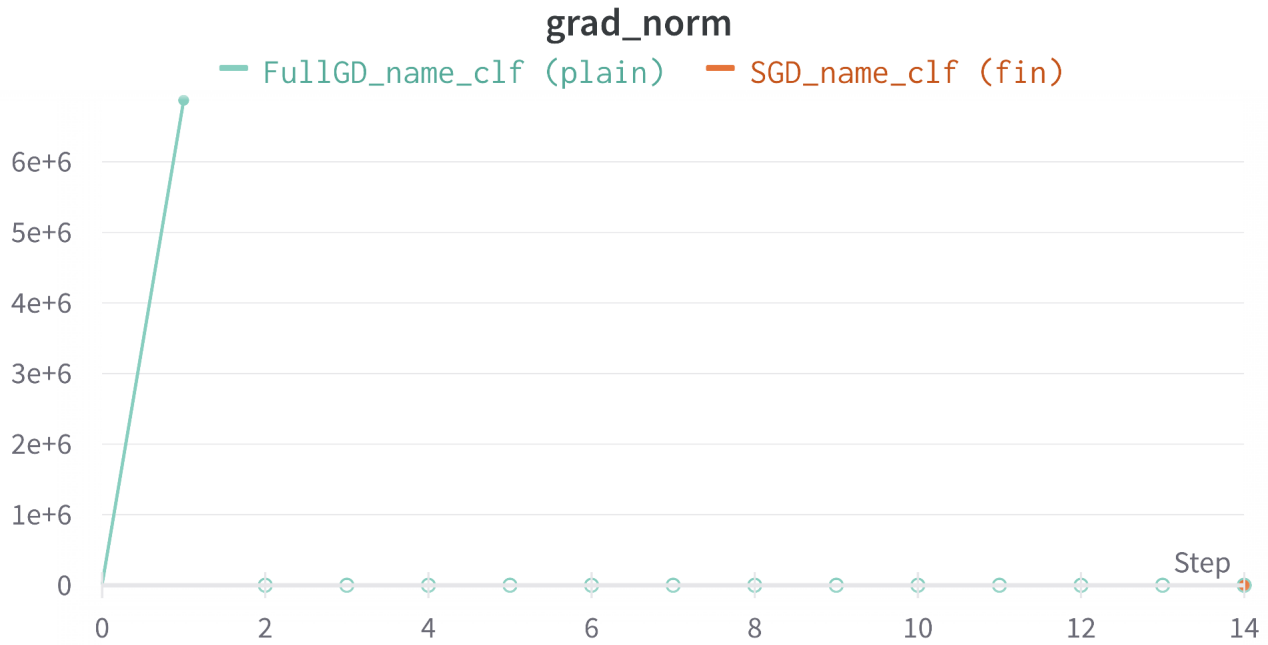
Классификация имен. Full GD, naive launch

Что будет, если в предыдущем запуске просто поменять SGD на Full GD





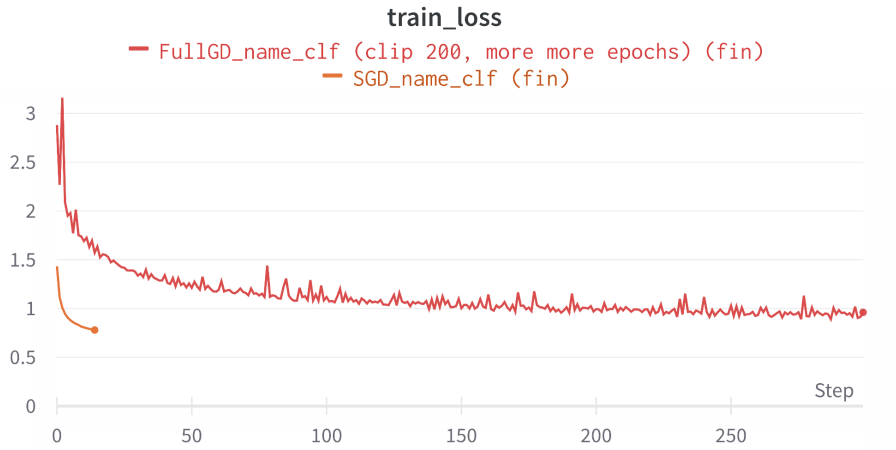
Причина?

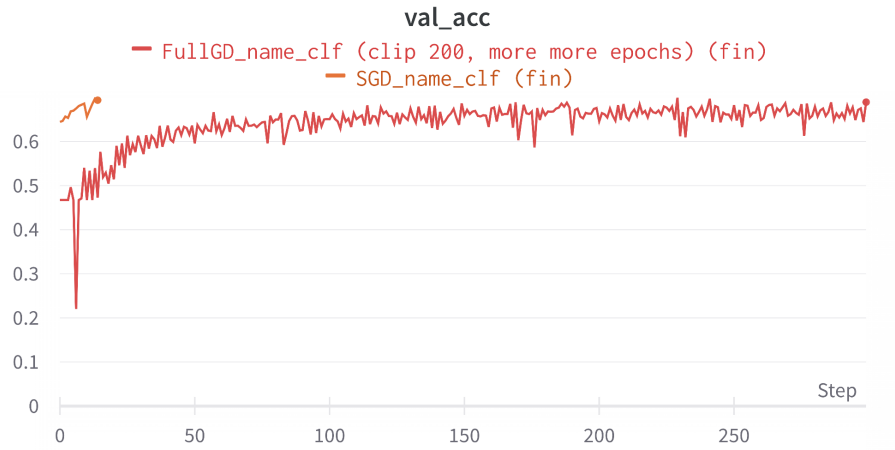


Градиента SGD даже не видно, но он там есть

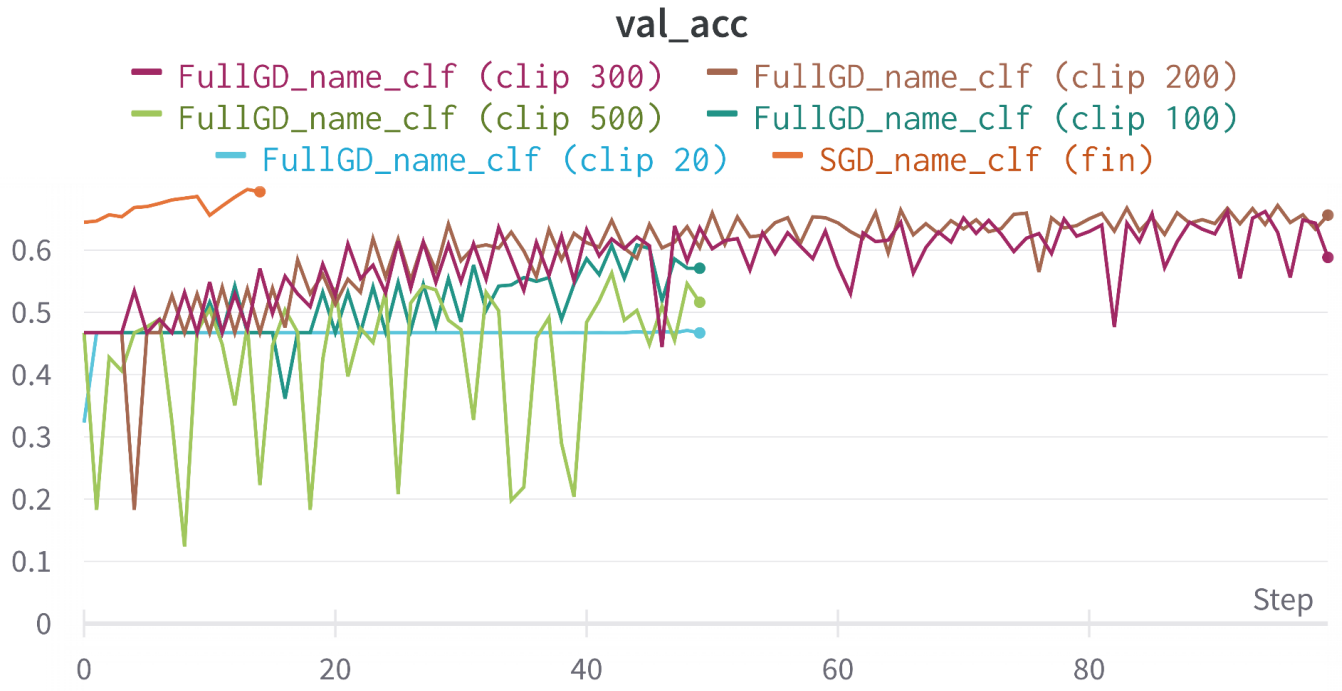
Классификация имен. Full GD, gradient clip 200 + more epochs (300)

Качество на валидации: ~ 0.68





(After extensive hyperparameter search)

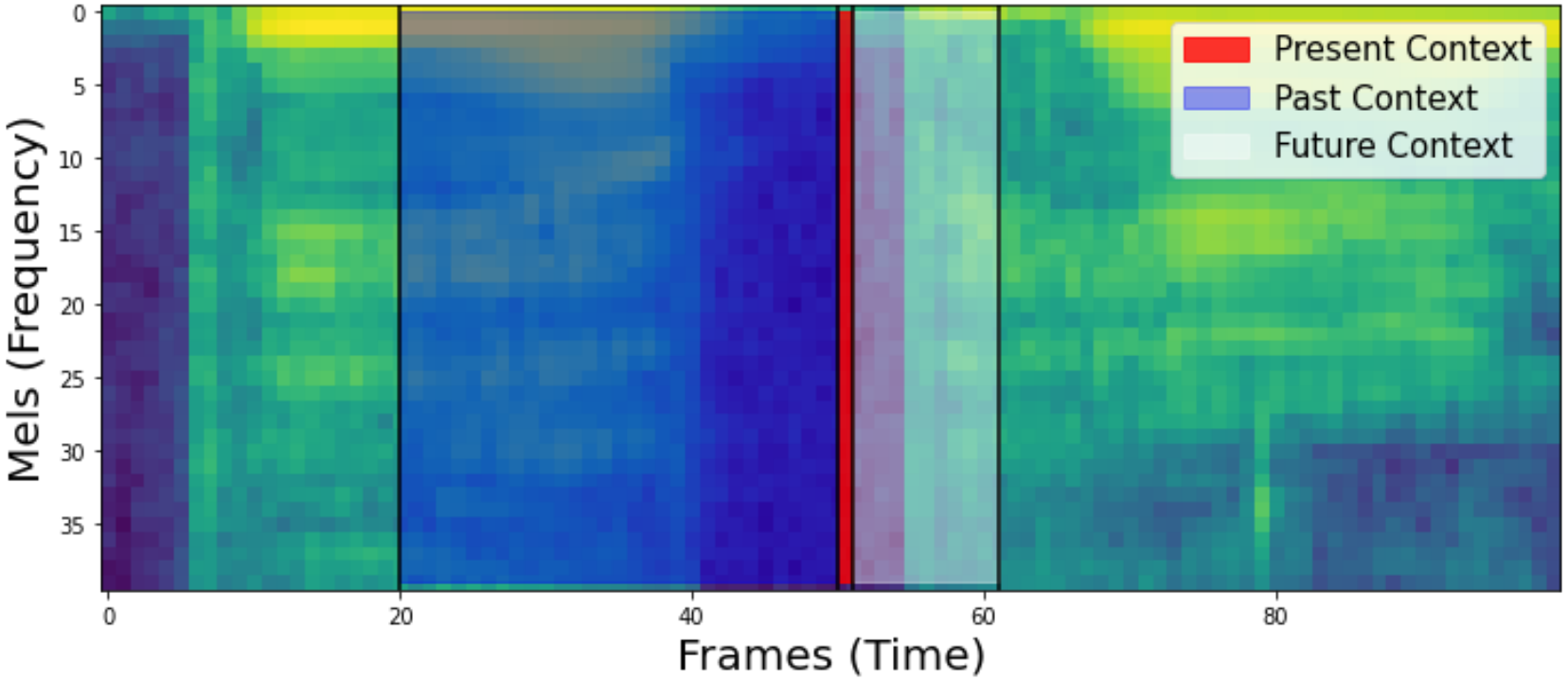


Задача 2. KeyWord Spotting (KWS)

Задача: научиться распознавать ключевое слово (‘Ok Google’, ‘Эй, Алиса’, ‘отправь (СОП до дедлайна)’)

(Картинки и информация взяты из курса DLA ВШЭ: https://github.com/markovka17/dla)

KWS. Как выглядит обучение.



KWS

Специфика задачи: модель нужна максимально легкой, что как раз нам подходит

Эксперимент основан на семинаре курса DLA ВШЭ: https://github.com/markovka17/dla/blob/2021/week06/seminar.ipynb

Будем отделять 3 выбранных слова от всех остальных.

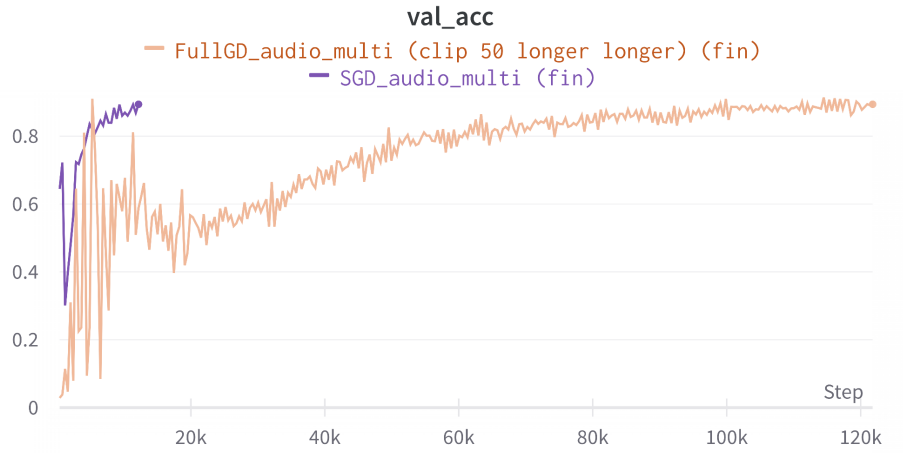
Данные: https://www.tensorflow.org/datasets/catalog/speech_commands

Модель: Conv2d+GRU+Attention+Linear, ~ 70 тыс параметров

KWS. Такие же эксперименты как с именами

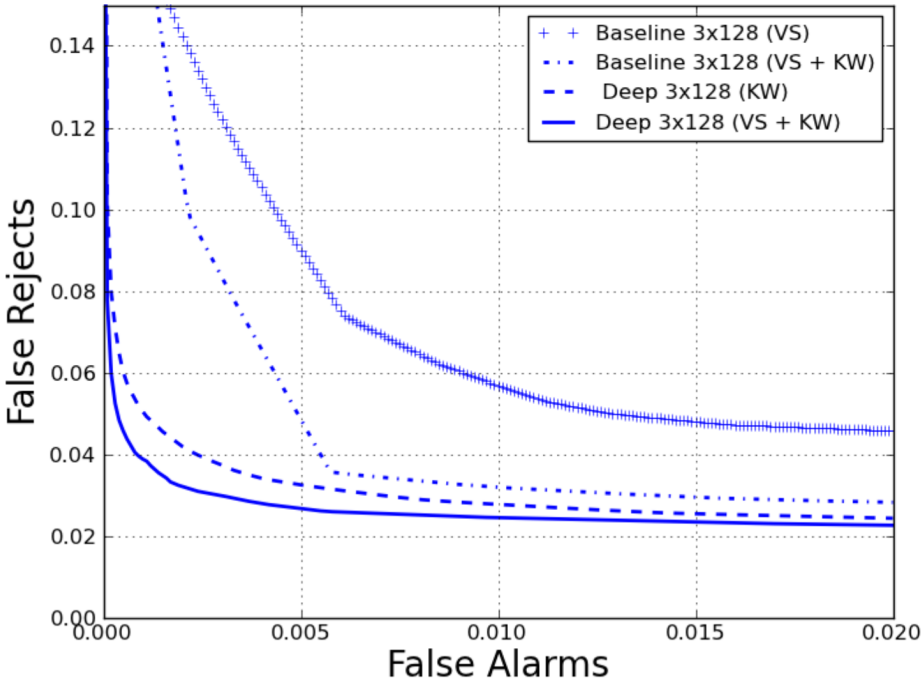
SGD, 30 эпох, качество на валидации: ~0.89

Full GD, grad clip 50, 300 эпох, качество на валидации: ~0.89



KWS. Качество на валидации - это всё?

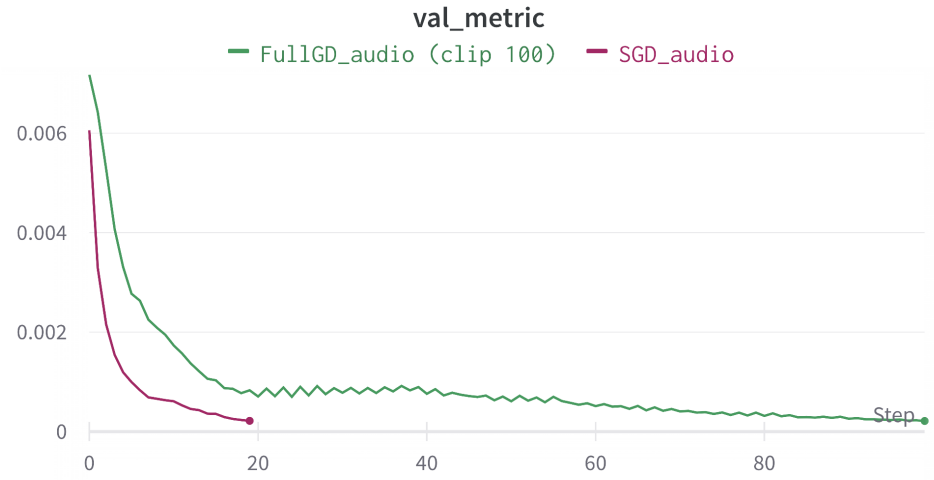
Вспоминая курс МО известно, что уверенность предсказания можно измерить с помощью метрики AUC-ROC. В KWS есть похожая метрика: AUC-FA-FR. Чем она ближе к 0, тем лучше.

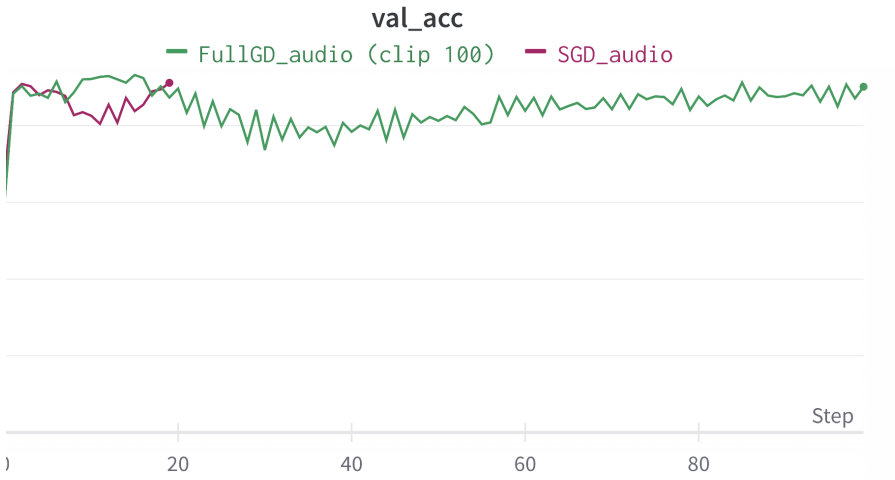


KWS. Binary classification

Метрика у SGD: ~0.000217, у FullGD: ~0.000214

(А если учить с помощью Adam, то вообще можно получить ~3e-5)





Вывод

Данные эксперименты скорее показывают, что возможно достичь качество, полученное с помощью SGD, используя Full GD и различные регуляризации. В экспериментах запуски SGD не доводились до наилучшего возможного качества, но даже полученный результат подтверждает намерение статьи рассматривать Full GD процедуры в теории. Однако даже на легких задачах видно что это неэффективно с практической точки зрения.

Наблюдения

- Gradient clipping сильно регуляризует обучение. Очень полезно его использовать в своих задачах (даже при Stochastic обучении)
- Важно смотреть не только на точность предсказания, но и на их уверенность
- В некоторых экспериментах можно заметить ситуацию с преодолением границы bias-variance tradeoff, описанную в статье deep double descent https://arxiv.org/pdf/1912.02292.pdf