Министерство образования и науки Российской Федерации
Московский физико-технический институт
Факультет управления и прикладной математики
Кафедра «Интеллектуальные системы»

Студент Жолобов Владимир Александрович

Методы типа градиентного клиппинга для задач на больших данных

03.03.01 — Прикладные математика и физика

Бакалаврская диссертация

Научный руководитель:
Гасников Александр Владимирович
доктор физико-математических наук

Москва
2021 г.

Аннотация

В данной работе рассматривается применение методов градиентного клиппинга для задач на больших данных. Существуют случаи, когда применение обычных градиентных методов в задачах с распределением стохастических градиентов с тяжелым хвостом о сходимости и качестве сходимости тяжело что-то сказать. Для проверки методов на работоспособность в таких задачах рассматриваются несколько задач. В качестве сравнения исследуются также стохастический градиентный спуск и ADAM. Получены результаты, подтверждающие работоспособность методов на больших данных. Методы исследованы на влияние на качество решения задач машинного обучения.

Ключевые слова: градиентный клиппинг, классификация изображений, семантическая сегментация, Super Resolution.

Содержание

1 Введение

Модели, основанные на нейронных сетях, используются в широком ряде задач. Существуют различные виды нейронных сетей: простые по строению сети прямого распространения (feed forward network), сверточные нейронные сети (CNN), например, для изображений [1], рекуррентные нейронные сети (RNN), например, для задач обработки естественного языка (NLP) [2]. Эти модели используются в разных задачах машинного обучения, однако вопрос выбора метода для обучения остается открытым. Проблема выбора метода обучения для каждой конкретной задачи и модели связан прежде всего с вычислительными затратами и затратами памяти. Так, для обучения нейронной сети чаще всего используются методы первого порядка типа градиентного спуска.

Обучение нейронной сети как и любой другой модели машинного обучения связано с решением оптимизационной задачи [3]. В качестве основного по популярности методом решения оптимизационной задачи выступает стохастический градиентный спуск (SGD) [4–6] Если задача достаточно хорошая, то есть распределение стохастических градиентов с узкими хвостами (light-tailed), тогда теория сходимости по математическому ожиданию хорошо коррелирует с поведением на практике.

С другой стороны, довольно много случаев, когда распределение шума в стохастических градиентах с тяжелыми хвостами (heavy-tailed) [7]. Для таких задача SGD часто менее надежен и показывает низкую производительность на практике. Также в таком случае теоретическая сходимость по математическому ожиданию гораздо хуже работает, чем в случае с узкими хвостами.

Для решения этой проблемы были предложены методы на основе градиентного клиппинга [8]. Основная идея состоит в том, что градиент по норме ограничивается сверху на каждом шаге, поэтому ожидается, что метод будет сходится лучше в случае узких хвостов. Теоретическая скорость сходимости по вероятности лучше, чем у стохастического градиентного спуска. Однако экспериментально эти методы не были проверены на задачах с большими данными, когда, например, в качестве модели используются нейронные сети. Цель работы - экспериментально проверить поведение методов на больших данных. В качестве задач машинного обучения здесь рассматриваются несколько - задача классификации изображений, задача семантической сегментации изображений и задача Super Resolution.

2 Постановка задачи

2.1 Постановка задачи оптимизации

Задача оптимизации формулируется в таком виде

$$\min_{x \in \mathbb{R}^n} f(x), \qquad f(x) = \mathbb{E}_\xi[f(x, \xi)], \tag{2.1}$$

где функция $f(x)$ гладкая выпуклая функция, а математическое ожидание (2.1) берется по случайной величине ξ, определенной на вероятностном пространстве $(\mathcal{X}, \mathcal{F}, \mathbb{P})$ с некоторой σ-алгеброй $\mathcal{F}$ и вероятностной мерой $\mathbb{P}$ Предполагается, что в любой точке $x \in \mathbb{R}^n$ для функции f доступен только стохастический градиент $\nabla f(x, \xi)$ такой, что

$$\mathbb{E}_\xi[\nabla f(x, \xi)] = \nabla f(x), \qquad \mathbb{E}_\xi[||\nabla f(x, \xi) - \nabla f(x)||_2^2] \leqslant \sigma^2$$

То есть вместо градиента функции $\nabla f(x)$ в точке x можно получить только его аппроксимацию, дисперсия которого ограничена сверху σ^2.

Определение 1. Будем говорить, что случайный вектор η имеет распределение с узкими хвостами, если существует $\mathbb{E}[\eta]$ и $\mathbb{P}\{||\eta - \mathbb{E}[\eta]||_2 > b\} \leqslant 2\exp\left(-\frac{b^2}{2\sigma^2}\right)$ для всех $b > 0$.

В другом виде выражение записывается как

$$\mathbb{E}[\exp(\frac{||\eta - \mathbb{E}[\eta]||_2^2}{\sigma^2}] \leqslant \exp(1)$$

2.2 Постановка задачи классификации изображений

Пусть X — множество описаний объектов, Y — конечное множество меток классов, где $|Y| \geqslant 2$.

Определение 2. Алгоритмом классификации $a : X \to Y$ называется функция, ставящая в соответствие описанию объекта $x \in X$ его метку класса $y \in Y$.

Определение 3. Функцией ошибки $D = (x_i, y_i)_{i=1}^m$ алгоритма a на конечной выборке называется

$$S = \frac{1}{k} \sum_{i=1}^{|D|} |a(x_i) \neq y_i|$$

Требуется построить алгоритм классификации a, который каждому объекту $x \in X$ ставит в соответствие ему метку класса $y \in Y$ и минимизирующий функцию S. В случае многоклассовой классификации модель выдает на выходе вектор вероятностей принадлежности к какому-то классу.

$$\hat{y} = a(x) = [\hat{y}_1, \ldots, \hat{y}_{|Y|}],$$

где $0 \leqslant \hat{y}_i \leqslant 1$ и $\sum_{i=0}^{|Y|} \hat{y}_i = 1$
Для функции качества используется $top1$ и $top5$ $accuracy$

$$Top1 = \sum_{i=1}^{m} |\arg\max a(x_i) = y_i|$$

$$Top5 = \sum_{i=1}^{m} [y_i \in Max5Set(a(x_i))],$$

где $Max5Set(a(x_i))$ обозначено множество из пяти значений по убыванию из множества $a(x_i)$.

2.3 Постановка задачи семантической сегментации изображений

Задан набор изображений $I \in \mathbb{R}^{M \times w \times h \times k}$, где M- число элементов выборки, w и h - размеры изображения, k - число цветовых каналов. Также задано множество классов объектов $C = \{0, 1, \ldots, N-1\}$. Здесь 0 обозначает задний фон изображения. Требуется построить отображение

$$\varphi(I_{lij}) = c$$

Здесь $c \in C, l \in \overline{1, M}$. В качестве функции ошибки для обучения используется кросс-энтропия для многоклассового случая

$$loss = -\sum_{c=1}^{M} y_{o,c} \log(p_{o,c}),$$

здесь M - число классов, $y_{o,c}$ - двоичный индикатор того, что метка класса c является правильной классификацией для наблюдения o, $p_{o,c}$ - предсказанная вероятность наблюдения o относится к классу c.
Для проверки качества модели используется функция качества степень пересечения между двумя изображениями (IoU)

$$IoU = \frac{TP}{TP + FN + FP},$$

где TP - число правильно классифицированных пикселей, FP - число пикселей, которые метод классифицировал как относящихся к классу, хотя они там не должны быть, FN - число пикселей, которые относятся к классу, но метод классифицировал их как не относящиеся к нему. Здесь в качестве двух изображений выступают

2.4 Постановка задачи super resolution

Задан набор изображений $\{I_{y_i}\}_{i=1}^{N}$ высокого разрешения. С помощью функции уменьшения размерности и ухудшения качества изображения строится набор изображений $\{I_{x_i}\}_{i=1}^{N}$ по заданному правилу

$$I_x = \mathcal{D}(I_y, \delta),$$

Задача super-resolution состоит в построении модели, которая при известном наборе строит аппроксимацию $\hat{I}_y$ изображений I_y высокого качества с помощью изображений низкого качества I_x

$$\hat{I}_y = \mathcal{F}(I_x, \theta),$$

где $\mathcal{F}$ является моделью задачи super resolution и θ обозначены ее параметры. Так как в большинстве случаев операция ухудшения изображения неизвестна, в качестве примера часто используется операция даунсэмплинга (downsampling)

$$\mathcal{D}(I_y; \delta) = (I_y) \downarrow_s, \{s\} \subset \delta,$$

где $\downarrow_s$ является операцией даунсэмплинга с параметром s.

Оптимальные параметры для модели вычисляются из задачи

$$\hat{\theta} = \arg\min_{\theta} \mathcal{L}(\hat{I}_y, I_y) + \lambda\Phi(\theta),$$

здесь $\mathcal{L}(\hat{I}_y, I_y)$ является функцией потери между сгенерированным изображением высокого разрещения $\hat{I}_y$ и истинным изображением высокого разрешения I_y, $\Phi(\theta)$ слагаемое в качестве регуляризации. Для функции ошибки используется попиксельная среднеквадратичная ошибка.

$$MSE = \frac{1}{mnk} \sum_{d=1}^{k} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I_y(i,j,d) - \hat{I}_y(i,j,k)]^2$$

В качестве критерия качества используется функция пикового отношения сигнала к шуму (PSNR)

$$\begin{aligned} PSNR &= 10 \cdot \log_{10}\left(\frac{MAX_I^2}{MSE}\right) \\ &= 20 \cdot \log_{10}\left(\frac{MAX_I}{\sqrt{MSE}}\right) \\ &= 20 \cdot \log_{10} - 10 \cdot \log_{10}(MSE), \end{aligned}$$

здесь MAX_I - это максимально возможное значение пикселя в изображении.

3 Теоретическая часть

3.1 Метод стохастического градиента

Основная идея метода в том, что вместо полного градиента функции считается его аппроксимация случайным образом путем батч-оценки из выборки.

Алгоритм 3.1 Stochastic Gradient Descent (SGD)

Input: начальная точка x^0, число итераций N, размеры батчей $\{m_k\}_{k=0}^{N-1}$, шаг $\gamma > 0$

for $k = 0, \ldots, N-1$ **do**

  Получаем $\xi_1^k, \ldots, \xi_{m_k}^k$ и вычисляем $\nabla f(x^{k+1}, \xi^k) = \frac{1}{m_k} \sum_{i=1}^{m_k} \nabla f(x^{k+1}, \xi_i^k)$

  $x^{k+1} = x^k - \gamma \nabla f(x^{k+1}, \xi^k)$

Output: $\overline{x}^N = \frac{1}{N} \sum_{k=0}^{N-1} x^k$

3.2 Методы типа градиентного клиппинга

Основное отличие методов этого класса от других является наличие операции клиппинга

$$clip(\nabla f(x,\xi), \lambda) = \min\{1, \frac{\lambda}{||\nabla f(x,\xi)||_2}\} \nabla f(x,\xi),$$

где $\nabla f(x,\xi) = \frac{1}{m} \sum_{i=1}^{m} \nabla f(x, \xi_i)$ является версией с мини-батчем функции $\nabla f(x)$. Для того, чтобы вычислить $clip(\nabla f(x,\xi), \lambda)$ нужно m раз независимо одинаково распределенно семплировать $\nabla f(x,\xi_1), \ldots, \nabla f(x,\xi_m)$, усреднить и спроектировать на шар радиуса λ по евклидовой норме с центром в начале координат.

3.2.1 Метод стохастического градиентного спуска с клиппингом

Алгоритм 3.2 Clipped Stochastic Gradient Descent (clipped-SGD)

Input: начальная точка x^0, число итераций N, размеры батчей $\{m_k\}_{k=0}^{N-1}$, шаг $\gamma > 0$, порог клиппинга $\lambda > 0$

for $k = 0, \ldots, N-1$ **do**

  Получаем $\xi_1^k, \ldots, \xi_{m_k}^k$ и вычисляем $\nabla f(x^{k+1}, \xi^k) = \frac{1}{m_k}\sum_{i=1}^{m_k} \nabla f(x^{k+1}, \xi_i^k)$

  Вычисляем $\widetilde{\nabla} f(x^{k+1}, \xi^k) = \mathrm{clip}(\nabla f(x^{k+1}, \xi^k), \lambda_{k+1})$

  $x^{k+1} = x^k - \gamma \widetilde{\nabla} f(x^k, \xi^k)$

Output: $\overline{x}^N = \frac{1}{N}\sum_{k=0}^{N-1} x^k$

Основной результат сходимости данного метода представлен в этой теореме

Теорема 1. Пусть функция f выпукла и L-гладка. Тогда для любых $\beta \in (0, 1)$ и $N \geqslant 1$ таких, что $\ln(\frac{4N}{\beta}) \geqslant 2$ верно, что после N шагов стохастического градиентного спуска с клиппингом с $\lambda = \Theta(LR_0)$ и $m_k = m = \Theta(\max\{1, \frac{N\sigma^2}{R_0^2 L^2 \ln(N/\beta)}$, где $R_0 = ||x^0 - x_*||_2$ и шаг $\gamma = \frac{1}{80L\ln(4N/\beta)}$ такой что $f(\overline{x}^N) - f(x^*) = O(\frac{LR_0^2 \ln(4N/\beta)}{N}$ с вероятностью не меньше $1 - \beta$. Здесь $\overline{x}^N = \frac{1}{N}\sum_{k=0}^{N-1} x^k$.

Другими словами, метод достигает оценки $f(\overline{x}^N) - f(x^*) \leqslant \varepsilon$ с вероятностью не менее $1 - \beta$ после $O(\frac{LR_0^2}{\varepsilon \ln(LR_0^2/\varepsilon\beta)})$ и требует $O(\max\{\frac{LR_0^2}{\varepsilon}, \frac{\sigma^2 R_0^2}{\varepsilon^2}\} \ln(\frac{LR_0^2}{\varepsilon\beta})$ вызовов оракула.

3.2.2 Метод стохастических подобных треугольников с клиппингом

Алгоритм 3.3 Clipped Stochastic Similar Triangles Method (clipped-SSTM)

Input: начальная точка x^0, число итераций N, размеры батчей $\{m_k\}_{k=1}^N$, параметр шага a, параметр клиппинга B

Обозначим $A_0 = \alpha_0 = 0$, $y^0 = z^0 = x^0$

for $k = 0, \ldots, N-1$ **do**

  Вычисляем $\alpha_{k+1} = \frac{k+2}{2aL}, A_{k+1} = A_k + \alpha_{k+1}, \lambda_{k+1} = \frac{B}{\alpha_{k+1}}$

  $\mathrm{x}^{k+1} = \frac{A_k y^k + \alpha_{k+1} z^k}{A_{k+1}}$

  $\xi_1^k, \ldots, \xi_{m_k}^k$ и вычисляем $\nabla f(x^{k+1}, \xi^k) = \frac{1}{m_k} \sum_{i=1}^{m_k} \nabla f(x^{k+1}, \xi_i^k)$

  Вычисляем $\widetilde{\nabla} f(x^{k+1}, \xi^k) = \mathrm{clip}(\nabla f(x^{k+1}, \xi^k), \lambda_{k+1})$

  $z^{k+1} = z^k - \alpha_{k+1} \widetilde{\nabla} f(x^{k+1}, \xi^k)$

  $y^{k+1} = \frac{A_k y^k + \alpha_{k+1} z^{k+1})}{A_{k+1}}$

Output: y^N

Для этого алгоритма есть оценка скорости сходимости.

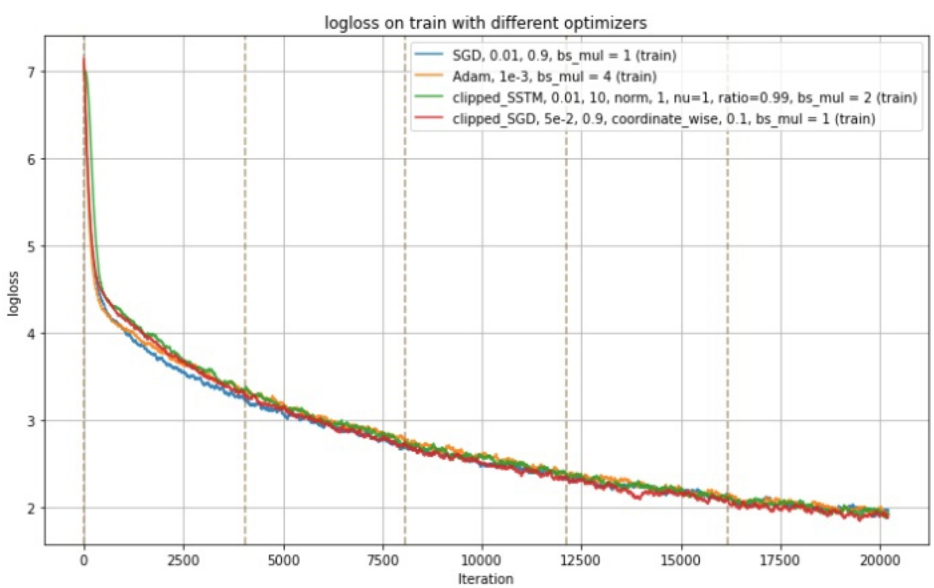
Теорема 2. Пусть функция f выпукла и L-гладка. Тогда для любого $\beta \in (0,1)$ и $N \geqslant 1$ таких, что $\ln(\frac{4N}{\beta}) \geqslant 2$ верно после N итераций метода подобных треугольников с клиппингом с параметрами $m_k = \Theta(\max\{1, \frac{2\alpha_{k+1}^2 N \ln(N/\beta)}{R_0^2}, B = \Theta(\frac{R_0}{\ln(N/\beta))}$ и $\alpha = \Theta(\ln^2(\frac{N}{\beta}$ справедливо $f(y^N) - f(x^*) = O(\frac{aLR_0^2}{N^2})$ с вероятностью не менее $1 - \beta$, где $R_0 = ||x^0 - x^*||_2||$.
Другими словами, если выбрать $a = \max\{1, \frac{16 \ln \frac{4N}{\beta}}{C}, 36(2 \ln \frac{4N}{\beta} + \sqrt{4 \ln^2 \frac{4N}{\beta} + 2 \ln \frac{4N}{\beta}}\}$, где $C = \sqrt{5}$, то метод достигнет $f(y^N) - f(x^*) \leqslant \varepsilon$ с вероятностью не менее $1 - \beta$ после $O(\sqrt{\frac{LR_0^2}{\varepsilon}} \ln \frac{LR_0^2}{\varepsilon\beta}))$ итераций и потребует вызовов оракула

4 Вычислительные эксперименты

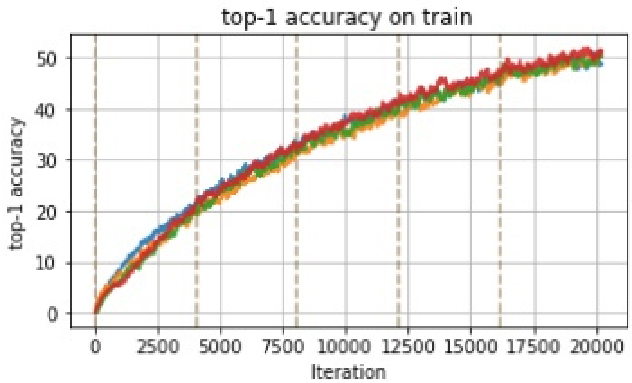
Для сравнения были использованы методы SGD, ADAM, clipped-SGD и clipped-SSTM. Эксперименты проводились на задачах классификации изображений для ImageNet-100, семантической сегментации на PascalVOC2012 и Super Resolution на DIV2K.

4.1 Классификация изображений на наборе данных ImageNet-100

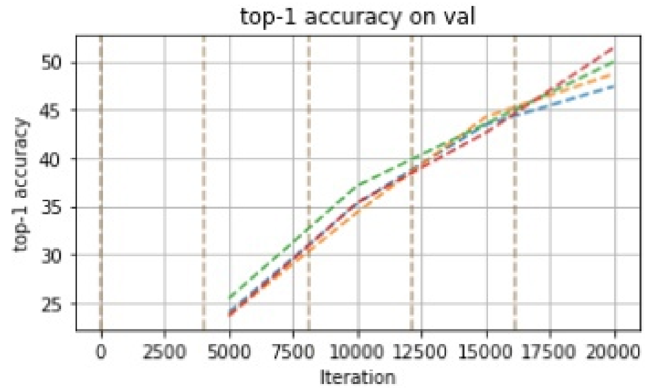
Набор данных ImageNet-100 состоит из данных первых 100 классов ImageNet [9]. В качестве модели был выбран ResNet-18. Размер батча был выбран 32. Были сравнены методы стохастического градиентного спуска (шаг 0.0001, momentum 0.99), стохастического градиентного спуска с клиппингом (шаг 0.001, momentum 0.99, покординатный спуск, уровень клиппинга 0.1), стохастических подобных треугольников с клиппингом (шаг 0.00001, $L = 10$, размер батча 2×32) и ADAM (шаг 0.0001 и размер батча 4×32).



Видим, что на этом наборе SGD применять вполне достаточно, то есть методы с градиентным клиппингом значительного улучшения здесь не дали.



(a) top-1 на Train



(b) top-1 на Validation

Рис. 1: Сравнение точностей моделей классификации на обучении и валидации

4.2 Семантическая сегментация на PascalVOC2012

Набор данных PascalVOC2012 состоит из 17125 цветных изображений в 20 различных классов. Классы представляют собой транспортные средства, домашнее хозяйство и животные, и другое: самолет, велосипед, лодка, автобус, автомобиль, мотоцикл, поезд, бутылка, стул, обеденный стол, растение в горшке, диван, телевизор/монитор, птица, кошка, корова, собака, лошадь, овца и человек.

В качестве кодировщика выбран предобученный на Imagenet VGG16, а в качестве декодировщика FCN32s. Были сравнены методы стохастического градиентного спуска (шаг 0.0001, momentum 0.99), стохастического градиентного спуска с клиппингом (шаг 0.0001, momentum 0.99, покординатный спуск, уровень клиппинга 0.1), стохастических подобных треугольников с клиппингом (шаг 0.00001, $L = 10$) и ADAM (шаг 0.0001).

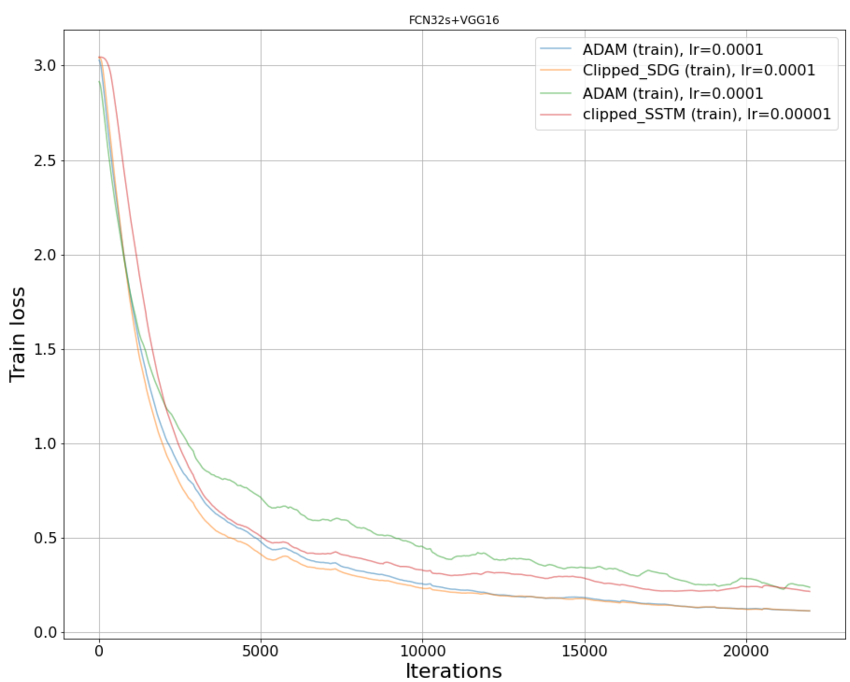
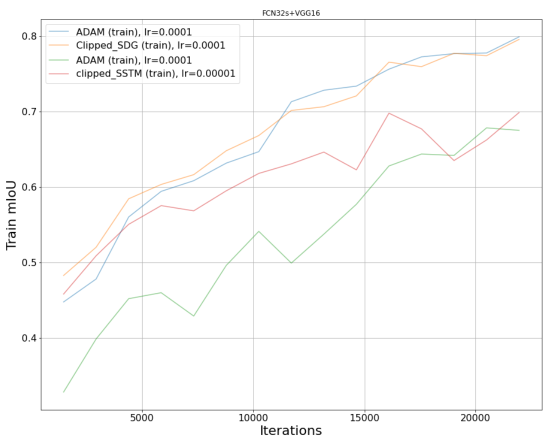
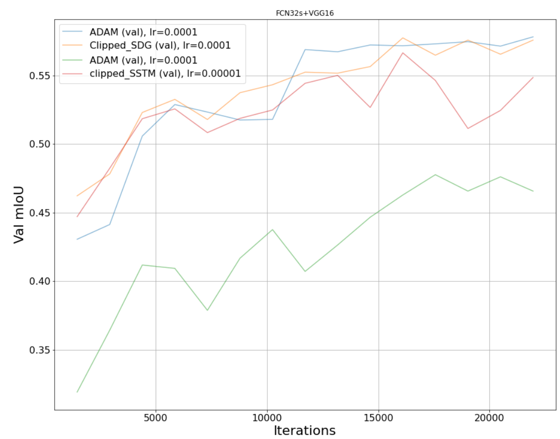


Рис. 2: Caption

Видим, что ADAM сходится чуть хуже остальных методов. Метод стохастического градиента с клиппингом сходится также как и обычный стохастический градиентный спуск. С другой стороны, метод стохастических подобных треугольников сходится медленнее в силу выбора шага.



(a) mIoU на Train



(b) mIoU на Validation

Рис. 3: Сравнение точностей моделей семантической сегментации на обучении и валидации

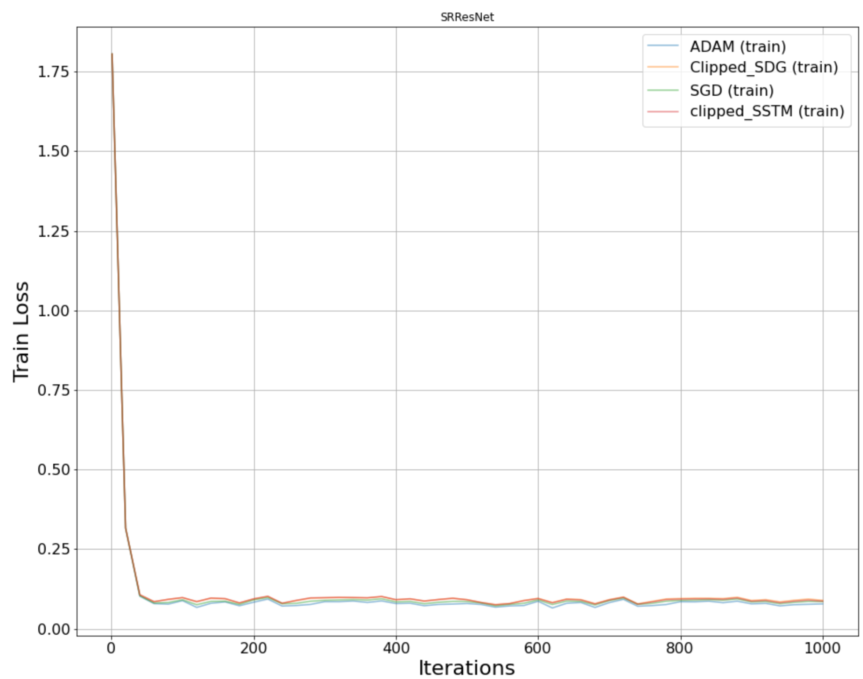
Приведем полученные значение качеств на моделях в зависимости от метода оптимизации

Метод	val mIoU
SGD	0.578
Clipped-SGD	0.576
ADAM	0.466
clipped-SSTM	0.549

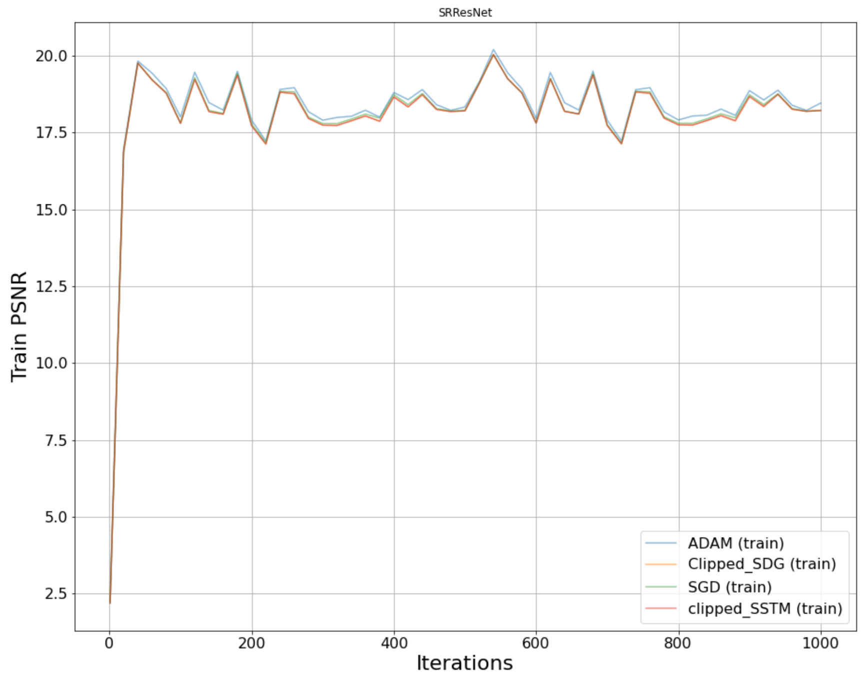
Таблица 1: Качество моделей на данных PascalVOC2012

4.3 Super Resolution на наборе данных DIV2K

Набор данных DIV2K состоит из 1000 изображений, 800 из которых используется для обучения. Для валидации и теста соответственно по 100 изображений. Элементы обучающей выборки получены путем даунсемплинга с параметром 4. Для сравнения использовались методы SGD (шаг 0.0001, momentum 0.99), clipped-SGD (шаг 0.0001, momentum 0.99, покординатный спуск, уровень клиппинга 0.1), clipped-SSTM (шаг 0.00001, $L = 10$) и ADAM (шаг 0.0001)



Как видим, на этой модели все методы работают примерно одинаково. Лучшим оказался $ADAM$ по размеру ошибки.



Приведем результаты моделей в таблице по качеству.

Метод	val mPSNR
ADAM	18.046
Clipped-SGD	17.855
SGD	17.873
clipped-SSTM	17.869

Таблица 2: Качество моделей на данных DIV2K

5 Заключение

Рассмотренные методы на задачах с большими данными продемонстрировали свою работоспособность. В рамках эксперимента не удалось обнаружить случая, когда свойства градиентного клиппинга могли бы значительно изменить качество решение. На рассмотренных задачах функция качества получилась порядка функции качества на известных методах SGD и ADAM. Также было обнаружено, что методы с градиентным клиппингом сходятся с меньшими осцилляциями.

Список литературы

[1] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012.

[2] Felix A Gers, Jürgen Schmidhuber, and Fred Cummins. Learning to forget: Continual prediction with lstm. *Neural computation*, 12(10):2451–2471, 2000.

[3] Jeff Heaton. Ian goodfellow, yoshua bengio, and aaron courville: Deep learning, 2018.

[4] Moritz Hardt, Ben Recht, and Yoram Singer. Train faster, generalize better: Stability of stochastic gradient descent. In *International Conference on Machine Learning*, pages 1225–1234. PMLR, 2016.

[5] Arkadij Semenovič Nemirovskij and David Borisovich Yudin. Problem complexity and method efficiency in optimization. 1983.

[6] Shai Shalev-Shwartz, Yoram Singer, Nathan Srebro, and Andrew Cotter. Pegasos: Primal estimated sub-gradient solver for svm. *Mathematical programming*, 127(1):3–30, 2011.

[7] Umut Simsekli, Levent Sagun, and Mert Gurbuzbalaban. A tail-index analysis of stochastic gradient noise in deep neural networks. In *International Conference on Machine Learning*, pages 5827–5837. PMLR, 2019.

[8] Eduard Gorbunov, Marina Danilova, and Alexander Gasnikov. Stochastic optimization with heavy-tailed noise via accelerated gradient clipping. *arXiv preprint arXiv:2005.10785*, 2020.

[9] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein,

et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.