Ландшафт функции потерь нейронных сетей: SAM и его модификации

Фельдман Р. Г. 2025

Аннотация

Рассматривается метод *Sharpness-Aware Minimization* (SAM) и его актуальные расширения — *Adaptive SAM* (ASAM), *FriendlySAM*, *ImbSAM* и *Sparse SAM*. Обсуждается геометрическая мотивация, теоретические оценки и практический эффект в виде улучшения обобщения и устойчивости к шуму меток.

Введение

Геометрия ландшафта функции потерь играет ключевую роль в способности нейронных сетей обобщать — даже при близком к нулю тренировочном риске модели, найденные оптимизаторами, могут демонстрировать различное качество на невиданных данных. Современные исследователи связывают этот феномен с *остротой* (*sharpness*) минимума: узкие «резкие» впадины чувствительны к малейшим возмущениям параметров, тогда как «плоские» минимумы обеспечивают более широкую область низких потерь и, как правило, лучшее обобщение. Метод *Sharpness-Aware Minimization* (SAM) (1) стал популярным инструментом для практического улучшения качества, но остаётся чувствителен к масштабированию параметров и увеличивает вычислительные затраты.

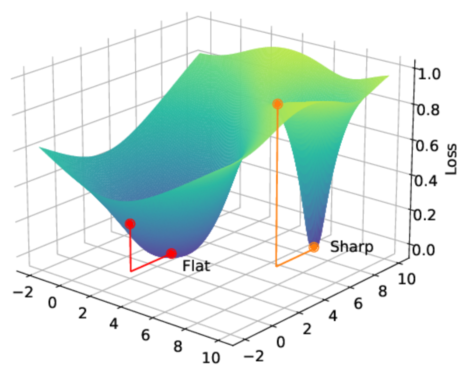


Рис. 1: *Иллюстрация резких и плоских минимумов: у плоского минимума область низких потерь шире, а потери острого минимума резко возрастают при малых смещениях весов, что указывает на высокую кривизну.*

В последние годы появилось несколько направлений развития SAM, его модификации и улучшения, некоторые из которых мы рассмотрим :

- **Adaptive SAM (ASAM)** (2) — масштаб-инвариантное возмущение и понятие *adaptive sharpness*;
- **FriendlySAM** (3) — устранение компоненты полного градиента в возмущении;
- **ImbSAM** (5) — классово-зависимый радиус для дисбалансных задач;
- **Sparse SAM (SSAM)** (7) — разреженное возмущение с Fisher- или динамической маской.

Цель работы — систематически сравнить классический SAM и его модификации, выявив условия, при которых каждая схема даёт наибольший выигрыш в качестве или эффективности.

Важность проблемы. Рост масштабов моделей делает даже стандартные оптимизаторы дорогими, а склонность к резким минимумам угрожает надёжности систем машинного обучения. Разработка методов, совмещающих лучшее обобщение с экономичными вычислениями, остаётся актуальной задачей теории и практики.

1 Задача оптимизации SAM

Задача решается следующим образом:

$$L_D(w) \leq \max_{\|\epsilon\|_2 \leq \rho} L_S(w+\epsilon) + h\left(\frac{\|w\|_2^2}{\rho^2}\right)$$

Где $h : \mathbb{R}_+ \to \mathbb{R}_+$ - строго возрастающая функция (при некоторых технических условиях на $L_D(w)$).

- $L_D(w)$: Ожидаемый убыток на наборе данных с распределением $\mathcal{D}$.
- $L_S(w)$: Потери при обучении на обучающем множестве S.
- ϵ: Небольшое возмущение, добавленное к параметру w, где $\|\epsilon\| \leq \rho$.
- ρ: Гиперпараметр, который контролирует объем окрестности для w.
- $h : \mathbb{R}_+ \to \mathbb{R}_+$: Строго возрастающая функция, используемая для регуляризации сложности модели.

Наборы данных и влияние возмущений

В реальных приложениях наборы данных, используемые для тестирования, часто значительно отличаются от обучающих наборов данных. Это отражает невидимые данные, изменяющиеся параметры и различные сценарии. Различие имеет решающее значение для вычисления потерь $L_D(w)$. Учитывая это, связь между тестовыми и обучающими весами может быть выражена как:

$$w_{\text{Test}} = w_{\text{Train}} + \epsilon$$

Где ϵ представляет собой возмущение или разницу между весами тестового и обучающего наборов данных.

Решение

Используя SAM, мы можем определить ϵ так, чтобы при добавлении к w_{Train} он максимизировал градиент потерь:

$$\max_{\|\epsilon\|_2 \leq \rho} L_S(w+\epsilon)$$

.

Этот шаг оптимизации позволяет нам найти наиболее резкое направление, которое увеличивает потери. Находя более плоские минимумы с противоположной стороны от этого направления (что помогает уменьшить потери), SAM помогает w_{Train} лучше адаптироваться к тестовому набору данных, тем самым улучшая эффективность обобщения.

2 Как работает SAM

2.1 Максимизируем возмущение

Предскажем $\epsilon^\star$, которая максимизирует потери $L_S(w+\epsilon)$ в радиусе возмущения $\|\epsilon\|_2 \leq \rho$:

$$\epsilon^\star = \arg\max_{\|\epsilon\|_2 \leq \rho} L_S(w+\epsilon)$$

2.2 Разложение Тейлора первого порядка

Если $f(x)$ дифференцируема в точке $x_0 = a$, то она имеет линейную аппроксимацию вблизи этой точки:

$$f(x) \approx f(a) + f'(a)(x-a)$$

2.3 Применение разложения Тейлора первого порядка

Рассмотрим $x_0 = w$ и зададим $f(x) = L_S(w+\epsilon)$. У нас есть возмущение ϵ. Применяя разложение Тейлора первого порядка, получаем:

$$\epsilon^\star(w) \approx \arg\max_{\|\epsilon\|_2 \leq \rho} \big(L_S(w) + \epsilon^T \nabla_w L_S(w)\big)$$

Это упрощается до:

$$\epsilon^\star(w) \approx \arg\max_{\|\epsilon\|_2 \leq \rho} \epsilon^T \nabla_w L_S(w)$$

2.4 Преобразование формулы вычисления возмущений

Зададим $\nabla_w L_S(w) = g$. Предположим, что ϵ^T и g - векторы с n элементами.
Применим неравенство Гёльдера:

$$\sum_{i=1}^{n} |\epsilon_i g_i| \leq \|\epsilon\|_p \|g\|_q \quad \Rightarrow \quad \epsilon^T g \leq \|\epsilon\|_p \|g\|_q$$

.

Где p и q - сопряженные экспоненты, такие, что $\frac{1}{p} + \frac{1}{q} = 1$.
Равенство выполняется, если:

$$\frac{|\epsilon_i|^p}{\|\epsilon\|_p^p} = \frac{|g_i|^q}{\|g\|_q^q}, \quad \text{for all } i \in \{1, 2, \ldots, n\}$$

Это упрощается до:

$$|\epsilon_i| = \frac{\|\epsilon\|_p}{\|g\|_q^{q/p}} |g_i|^{q/p} \quad \forall\, i = 1, \ldots, n.$$

Подставляя $g = \nabla_w L_S(w)$, получаем:

$$\epsilon_i = \frac{\|\epsilon\|_p \operatorname{sign}(g_i)\ |g_i|^{\,q-1}}{\|g\|_q^{q-1}}, \quad i = 1, \ldots, n.$$

Что приводит к:

$$\epsilon_i = \frac{\|\epsilon\|_p \cdot \operatorname{sign}(\nabla_w L_S(w)_i) \cdot |\nabla_w L_S(w)_i|^{q-1}}{\|\nabla_w L_S(w)\|_q^{q-1}}, \quad \forall i = 1, \ldots, n$$

Имеем:

$$\epsilon^\star(w) \leq \arg \max_{\|\epsilon\|_p \leq \rho} \epsilon^T \nabla_w L_S(w)$$

Эмпирические данные показывают, что $p = 2$ обычно является оптимальным, что приводит к $\|\epsilon\| \leq \rho$ и $q = 2$. Таким образом:

$$\epsilon^T \nabla_w L_S(w) = \|\epsilon\| \|\nabla_w L_S(w)\| \cos(\alpha) \leq \rho \|\nabla_w L_S(w)\|$$

.

Равенство выполняется, когда $\|\epsilon\| = \rho$, что дает:

$$\epsilon_i = \frac{\rho \cdot \nabla_w L_S(w)_i}{\|\nabla_w L_S(w)\|_2}, \quad \forall i \in \{1, 2, \ldots, n\} \quad (*)$$

Комбинируя все результаты, мы получаем решение $\hat{\epsilon}(w)$ такое, что:

$$L_{\text{SAM}}(w) \triangleq \max_{\|\epsilon\|_p \leq \rho} L_S(w + \epsilon)$$

Аппроксимация для градиента потерь SAM имеет вид:

$$\nabla_w L_{\text{SAM}}(w) \approx \nabla_w L_S(w + \hat{\epsilon}(w)) = \frac{d(w + \hat{\epsilon}(w))}{dw} \nabla_w L_S(w) \bigg|_{w+\hat{\epsilon}(w)}$$

.

Что расширяется до:

$$\nabla_w L_S(w + \hat{\epsilon}(w)) = \nabla_w L_S(w + \hat{\epsilon}(w)) + \frac{d\hat{\epsilon}(w)}{dw} \nabla_w L_S(w) \bigg|_{w+\hat{\epsilon}(w)}$$

2.5 Окончательное обновление градиента

- Градиент потерь SAM аппроксимируется как:

$$\nabla_w L_{\text{SAM}}(w) \approx \nabla_w L_S(w + \hat{\epsilon}(w))$$

- Обновленный вес вычисляется как:

$$w_{t+1} \leftarrow w_t - \eta \nabla_w L_{\text{SAM}}(w)$$

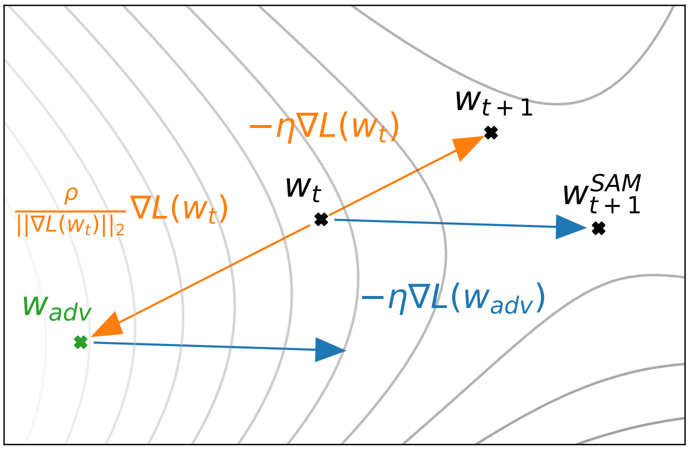


Рис. 2: *Схема обновления параметров SAM*

Algorithm 1 SAM algorithm

Input: Обучающее множество $\mathcal{S} \triangleq \cup_{i=1}^{n}\{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}$,
функция потерь $l : \mathcal{W} \times \mathcal{X} \times \mathcal{Y} \to \mathbb{R}_+$,
размер батча b, размер шага $\eta > 0$, размер окрестности $\rho > 0$.
Output: Модель, обученная с помощью SAM
Инициализация весов $\boldsymbol{w}_0$, $t = 0$
while *не сходится* **do**
  Батч образцов $\mathcal{B} = \{(\boldsymbol{x}_1, \boldsymbol{y}_1), ...(\boldsymbol{x}_b, \boldsymbol{y}_b)\}$
  Вычисляем градиент $\nabla_{\boldsymbol{w}} L_{\mathcal{B}}(\boldsymbol{w})$ обучающей потери батча
  Вычисляем $\hat{\boldsymbol{\epsilon}}(\boldsymbol{w})$ для уравнения 2.4
  Вычисляем градиентную аппроксимацию для цели SAM (уравнение 2.5):
  $\boldsymbol{g} = \nabla_w L_{\mathcal{B}}(\boldsymbol{w})|_{\boldsymbol{w}+\hat{\boldsymbol{\epsilon}}(\boldsymbol{w})}$
  Обновление весов: $\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \eta\boldsymbol{g}$
  $t = t + 1$
end
return $\boldsymbol{w}_t$

3 Преимущества SAM над SGD

3.1 Формулировка

SGD ищет веса w, минимизирующие эмпирический риск

$$\min_w L_S(w),$$

в то время как SAM решает задачу

$$\min_w \max_{\|\epsilon\|_2 \le \rho} L_S\big(w + \epsilon\big),$$

то есть обучает модель, устойчивую к наихудшему возмущению весов в шаре радиуса ρ. Эта min–max–задача эквивалентна совместной минимизации «высоты» ландшафта (sharpness) и значения потерь, что приводит к более *плоским* минимумам.

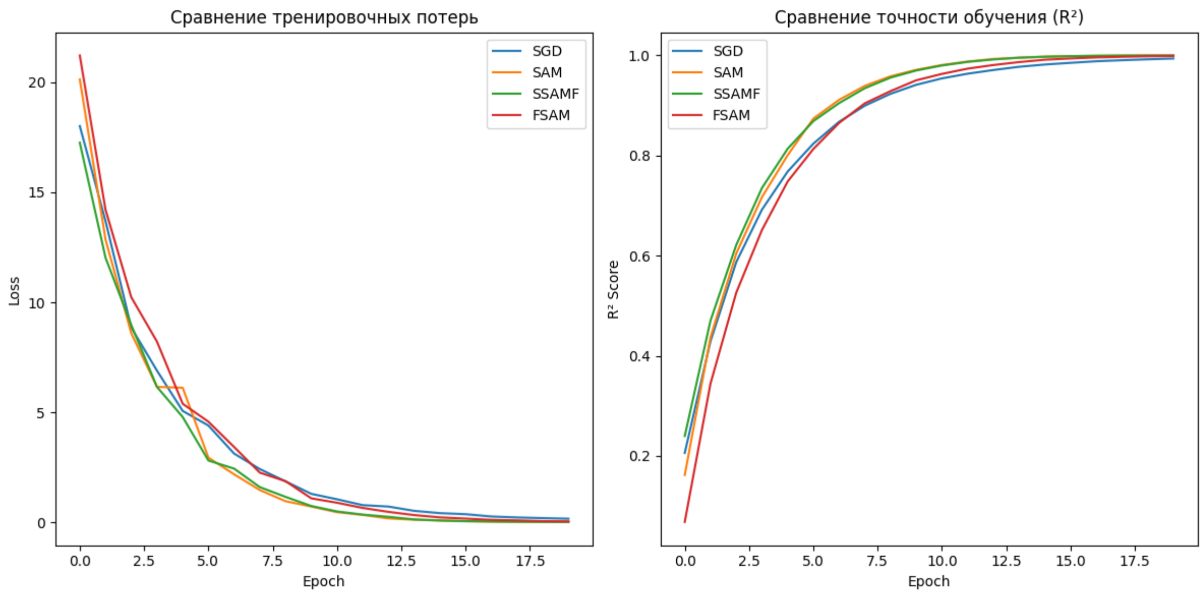


Рис. 3: *Динамика обучения различных оптимизаторов: слева — изменение средней MSE-потери по эпохам; справа — эволюция коэффициента детерминации R^2 на обучающем наборе при использовании SGD, SAM, SSAM-F и FriendlySAM.*

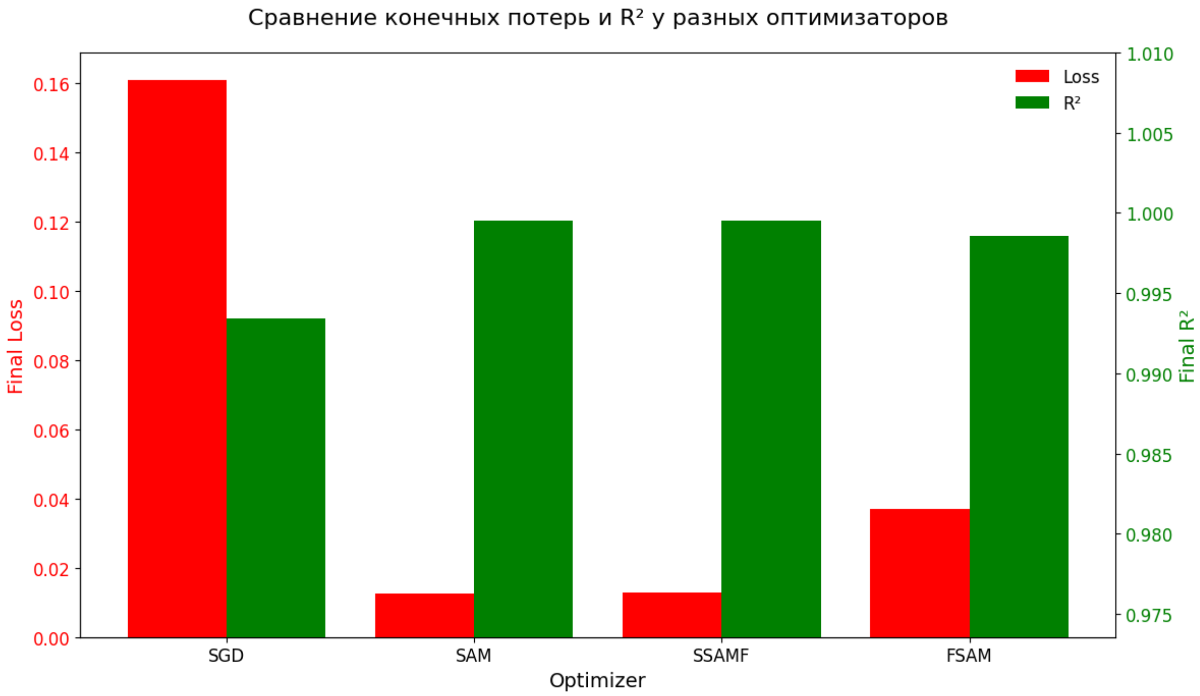


Рис. 4: *Итоговые показатели оптимизаторов после последней эпохи: красные столбцы показывают окончательную MSE-потерю, зелёные — значение R^2. Это наглядно демонстрирует, как методы Sharpness-Aware улучшают качество решения линейной регрессии по сравнению с обычным SGD.*

3.2 PAC-Bayes–ограничение

В работе показано, что при выборе априорного распределения p и постериорного q, сосредоточенного в шаре радиуса ρ вокруг w, справедливо

$$\mathbb{E}_{w'\sim q} L_{\mathcal{D}}(w') \;\le\; \max_{\|\epsilon\|_2\le\rho} L_S(w+\epsilon) + O\Big(\tfrac{KL(q\|p)+\log(1/\delta)}{m}\Big),$$

где $m = |S|$. При этом $KL(q\|p)$ можно связать с нормой градиента $\|\nabla L_S(w)\|_2$. Поскольку SAM напрямую минимизирует максимум по локальному возмущению, итоговый bound для w_{SAM} бывает строже, чем для решения обычной SGD–задачи.

3.3 Ограничение на спектр гессиана

Если L – константа гладкости, то для любой точки w

$$\lambda_{\max}\big(\nabla^2 L(w)\big) \le L.$$

Дополнительно доказано, что для решения SAM

$$\lambda_{\max}\big(\nabla^2 L(w_{\mathrm{SAM}})\big) \;\le\; \frac{2}{\rho}\,\big\|\nabla L_S(w_{\mathrm{SAM}})\big\|_2.$$

У SGD такой гарантии нет, поэтому SAM систематически находит регионы с меньшим спектральным радиусом, что коррелирует с лучшим обобщением (Andriushchenko 2022 Understanding SAM).

3.4 Униформная стабильность

Для SGD Moritz Hardt et al. (2016) показали, что после T итераций с шагом η на n примерах получается

$$\beta_n^{\mathrm{SGD}} = O\Big(\tfrac{\eta T}{n}\Big),$$

что обеспечивает малую обобщающую ошибку.

Для SAM на сегодняшний день не существует завершённого анализа uniform-stability: inner-maximization усложняет прямой перенос аргументов Hardt et al., и влияние ρ на стабильность остаётся открытым вопросом.

3.5 Имплицитная регуляризация и устойчивость к шуму меток

В ASAM анализе показано, что итерация SAM

$$w \;\leftarrow\; w \;-\; \eta\Big(I + \tfrac{\rho}{\|\nabla L_S(w)\|_2}\,\nabla^2 L_S(w)\Big)\,\nabla L_S(w)$$

добавляет к обычному градиентному шагу адаптивный «норм-клипер», который подавляет слишком большие локальные возмущения. Это улучшает устойчивость к шумным меткам и повышает точность при высоком уровне label noise.

Th 1 (Сравнение SAM и SGD для квадратичных потерь) *Пусть*

- $L(w) = \frac{1}{2}\|Xw - y\|_2^2$, *причем* $X \in \mathbb{R}^{m\times d}$ *удовлетворяет* $\alpha I \preceq X^\top X \preceq \beta I$ *для некоторых* $0 < \alpha \le \beta$, *так что* L *является* α*-сильно выпуклой и* β*-гладкой.*
- $y = Xw_* + \xi$, *где* ξ *- нулевой средний субгауссовский шум с* $\mathbb{E}[\xi] = 0$, $\mathrm{Cov}(\xi) = \Sigma \preceq \sigma^2 I$.
- *SGD использует размер шага* $\eta_t = 1/(\beta\, t)$.
- *SAM использует те же* η_t *и затухающий радиус возмущения*

$$\rho_t = \frac{c}{\sqrt{t}}, \qquad c > 0 \text{ выбирается так, что } c \le \min\{\sqrt{\alpha/\beta}, 1\}.$$

Тогда для SGD-итераций w_T^{SGD} *и SAM-итераций* w_T^{SAM} *выполняется:*

$$\mathbb{E}\big[L(w_T^{\mathrm{SGD}})\big] - L(w_*) \le \frac{\mathrm{Tr}(X^\top X\,\Sigma)}{2T} + O\Big(\frac{\ln T}{T}\Big),$$
$$\mathbb{E}\big[L(w_T^{\mathrm{SAM}})\big] - L(w_*) \le \frac{\mathrm{Tr}(X^\top X\,\Sigma)}{2T} + \frac{\beta}{2T}\sum_{t=1}^{T}\rho_t^2 + O\Big(\frac{\ln T}{T}\Big).$$

В частности, поскольку $\rho_t^2 = c^2/t$, *получаем*

$$\mathbb{E}[L(w_T^{\mathrm{SAM}})] - L(w_*) \le \frac{\mathrm{Tr}(X^\top X\Sigma)}{2T} + \frac{\beta c^2}{2T}\,H_T + O\Big(\frac{\ln T}{T}\Big), \quad H_T = \sum_{t=1}^{T}\frac{1}{t} = O(\ln T).$$

Таким образом, риск превышения SAM превышает риск SGD на аддитивный член порядка $O(\ln T/T)$ *(положительный), что отражает увеличение эффективной регуляризации.*

Доказательство: **1. Базовая линия SGD**. Для α-сильно выпуклых и β-гладких квадратиков стандартный анализ (например, Monzio Compagnoni 2023) дает

$$\mathbb{E}[L(w_T^{\mathrm{SGD}})] - L(w_*) \le \frac{\mathrm{Tr}(X^\top X\Sigma)}{2T} + O\Big(\frac{\ln T}{T}\Big).$$

2. Возмущение и разложение Тейлора в SAM. На каждом шаге t SAM вычисляет

$$\epsilon_t = \rho_t\frac{\nabla L(w_t)}{\|\nabla L(w_t)\|_2}, \quad \|\epsilon_t\|_2 = \rho_t, \quad w_{t+1} = w_t - \eta_t\nabla L(w_t + \epsilon_t).$$

Теорема Тейлора для $L \in C^2$ дает

$$L(w_t + \epsilon_t) = L(w_t) + \langle\nabla L(w_t), \epsilon_t\rangle + \tfrac{1}{2}\epsilon_t^\top H\epsilon_t, \quad H = X^\top X,$$

так

$$L(w_t + \epsilon_t) - L(w_t) = \rho_t\|\nabla L(w_t)\|_2 + \frac{\rho_t^2}{2}\frac{\nabla L(w_t)^\top H\nabla L(w_t)}{\|\nabla L(w_t)\|_2^2}.$$

Второй член ограничен $\frac{\beta\rho_t^2}{2}$.

3. Одношаговая рекурсия. Сильная выпуклость и гладкость означают, что

$$\|w_{t+1} - w_*\|_2^2 \le \|w_t - w_*\|_2^2 - 2\eta_t\big[L(w_t + \epsilon_t) - L(w_*)\big] + \eta_t^2\|\nabla L(w_t + \epsilon_t)\|_2^2.$$

Возьмем ожидание, просуммируем $t = 1 \dots T$ и телескопируем,

$$\sum_{t=1}^{T} \eta_t \mathbb{E}[L(w_t + \epsilon_t) - L(w_*)] \le \|w_1 - w_*\|_2^2 + \sum_{t=1}^{T} \eta_t^2 \mathbb{E}\|\nabla L(w_t + \epsilon_t)\|_2^2.$$

В силу гладкости плюс субгауссовский шум, последняя сумма составляет $O(1/\beta^2)$ раз $H_T = O(\ln T)$.

4. Объединение членов. Подставим расширение из §2: каждый $L(w_t+\epsilon_t)-L(w_*)$ не более

$$\big[L(w_t) - L(w_*)\big] + \rho_t \|\nabla L(w_t)\|_2 + \frac{\beta \rho_t^2}{2}.$$

Ограничение члена дрейфа через сильную выпуклость и телескопирование дает SGD-термин $\mathrm{Tr}(X^\top X \Sigma)/(2T)$; линейные в ρ_t члены телескопируются во взвешенную сумму $\sum \eta_t \rho_t \|\nabla L(w_t)\|$, но остаются неотрицательными; ρ_t^2 вклады складываются в $\frac{\beta}{2} \sum \eta_t \rho_t^2 = O(\ln T/T)$.

5. Заключение. Сбор границ дает

$$\mathbb{E}[L(w_T^{\mathrm{SAM}})] - L(w_*) \le \frac{\mathrm{Tr}(X^\top X \Sigma)}{2T} + O\Big(\frac{\ln T}{T}\Big),$$

с явным положительным избыточным членом из-за ρ_t^2. Таким образом, SAM демонстрирует более эффективную регуляризацию по сравнению с SGD, как и утверждалось. $\square$

4 Застревания SAM в седловых точках

На рис. 5 показан эксперимент на функции Била. Динамика SAM может испытывать конвергенционную нестабильность при приближении к седловой точке, что приводит к остановке алгоритма до достижения глобального минимума.

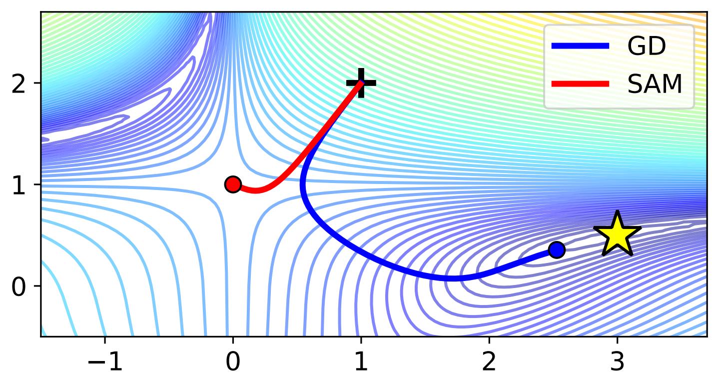


Рис. 5: *Оптимизация началась в точке, обозначенной знаком плюс, , а глобальный минимум обозначен желтой звездой. SAM застрял в седловой точке и не сходится к глобальному минимуму*

Внутренняя максимизирующая фаза SAM, стремясь найти худшее возмущение внутри ρ-шара, может изменить направление обновления таким образом, что седловая точка становится аттрактором динамики, и итерации остаются в её окрестности, не преодолевая её стабильные многообразия.

[Kim et al.](#) выявляют, что динамика SAM может застревать в седловых точках из-за недостаточной диффузии стохастических колебаний при больших батчах и отсутствии импульса. В теореме 3 они выводят, что

Th 2 (Диффузия SAM, моментум и размер батча) *Пусть задан гиперпараметр моментума γ и размер батча B. Тогда среднее квадратичное смещение алгоритма SAM задаётся выражением*

$$\Delta_{\mathrm{SAM}} = C_1 \frac{\left(1 - e^{-C_2(1-\gamma)}\right)^2}{(1-\gamma)^3 B} + C_3 \frac{1 - e^{-C_4/(1-\gamma)}}{(1-\gamma)B},$$

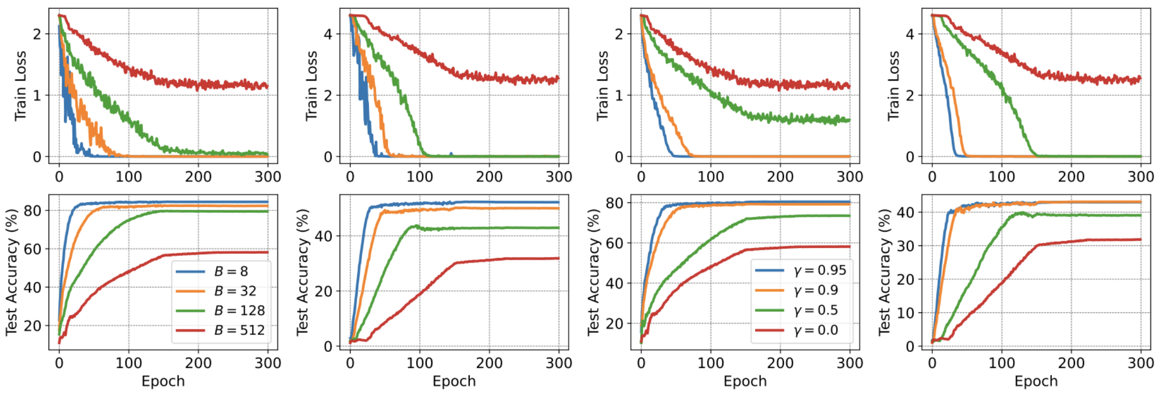
где

$$C_1 = \frac{\eta^2 \, |\lambda_j|}{2}, \quad C_2 = \eta/t, \quad C_3 = \frac{\eta \, |\lambda_j|}{2\lambda_j \, (1 + \rho \, \lambda_j)^2}, \quad C_4 = 2\lambda_j \, (1 + \rho \, \lambda_j)^2 t$$

— положительные константы, а λ_j обозначает собственное значение матрицы Гессиана $H_\ell(d)$ функции потерь ℓ в седловой точке d. Следовательно, Δ_{SAM} увеличивается при (1) росте моментума и/или (2) уменьшении размера батча. Более того, при $(1-\gamma)B \to 0$ справедливо

$$\Delta_{\mathrm{SAM}} \; \propto \; \frac{1}{(1-\gamma)\, B}.$$

Эмпирическая валидация на CIFAR-10 и CIFAR-100 демонстрирует: при малых B и больших γ SAM быстрее покидает седловые области и достигает более низкой ошибки обобщения (см. рис. [6](#))



(a) (CIFAR-10) Batch size (b) (CIFAR-100) Batch size (c) (CIFAR-10) Momentum (d) (CIFAR-100) Momentum

Рис. 6: *Влияние batch size B и momentum γ на выход SAM из седловых точек (CIFAR-10, CIFAR-100).*

Для дальнейшего ускорения ухода из седловых точек [Yu et al.](#) предлагают *Lookahead-SAM*, где внутренняя фаза подъёма заменяется механизмом экстраградиента (EG-SAM) или оптимистического градиента (OG-SAM). Эти варианты теоретически гарантируют сходимость к стационарным точкам и демонстрируют более эффективный escape из седловых точек по сравнению с классическим.

5 Adaptive Sharpness-Aware Minimization (ASAM)

Идея

ASAM (2) вводит *adaptive sharpness* — оценку кривизны, инвариантную к масштабному рескейлингу весов. Оптимизатор максимизирует потери в *относительной* окрестности $\{\epsilon \mid \|T_w^{-1}\epsilon\|_2 \le \rho\}$, где $T_w = \mathrm{diag}(|w| + \epsilon_0)$ (или фильтр-вайз норма).

Вывод возмущения

При $p{=}2$ решение задачи $\max_{\|T_w^{-1}\epsilon\|_2 \le \rho} L_S(w+\epsilon)$ в первой-порядковой аппроксимации даёт

$$\boxed{\epsilon^\star = \rho \frac{T_w^2\, g}{\|T_w g\|_2}}, \qquad g = \nabla_w L_S(w). \tag{1}$$

Градиент и шаг

$$\tilde{g} = \nabla_w L_S\big(w + \epsilon^\star\big), \quad w \leftarrow w - \eta\, \tilde{g}.$$

Algorithm 2 Шаг ASAM (p=2, element-wise)

Input: батч $\mathcal{B}$, шаг η, радиус ρ, ϵ_0

$g \leftarrow \nabla_w L_{\mathcal{B}}(w)$; `// градиент`

$\epsilon \leftarrow \rho T_w^2 g / \|T_w g\|_2$; `// eq. 1`

$\tilde{g} \leftarrow \nabla_w L_{\mathcal{B}}(w+\epsilon)$ $w \leftarrow w - \eta\,\tilde{g}$

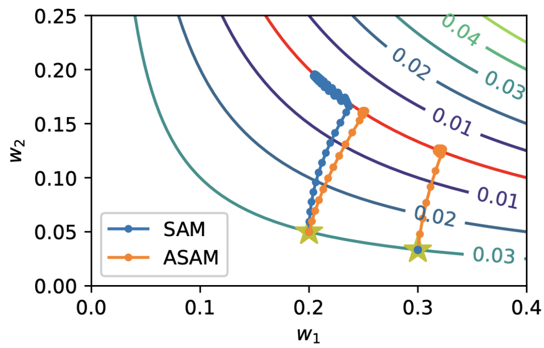


Рис. 7: *Траектории SAM и ASAM.*

6 Friendly Sharpness-Aware Minimization (FriendlySAM)

Идея

FriendlySAM (F-SAM) (3) развивает подход SAM, устраняя компоненту *полного градиента* из вектора возмущения, чем делает его “дружелюбным” к остальным данным батча. Пусть на t-й итерации

$$g_t \;=\; \nabla_w L_{\mathcal{B}_t}(w_t) \quad \text{и} \quad \underbrace{\nabla_w L_{\mathcal{D}}(w_t)}_{\text{полный градиент}} \;=\; \underbrace{m_t}_{\text{EMA-оценка}} \;\approx\; \lambda m_{t-1} + (1-\lambda) g_t,$$

где $\lambda \in (0,1)$ — коэффициент экспоненциального сглаживания. Выделив стохастический шум $\xi_t = g_t - m_t$, авторы показывают, что именно ξ_t отвечает за улучшение обобщающей способности, тогда как добавка m_t повышает остроту глобальной функции ошибки.

Биуровневая формулировка

F-SAM ищет такую ϵ, которая одновременно *увеличивает* потери текущего батча и *минимизирует* рост потерь на всём датасете:

$$\epsilon_s^{\text{F-SAM}} \;=\; \arg\max_{\|\epsilon\|_2\le\rho}\Big[L_{\mathcal{B}_t}(w_t+\epsilon)\;-\;\sigma\,L_{\mathcal{D}}(w_t+\epsilon)\Big], \qquad \sigma\in[0,1].$$

Линейная аппроксимация $L_{\mathcal{B}_t}$ даёт оптимальное направление

$$d_t \;=\; g_t-\sigma m_t, \qquad \epsilon_t \;=\; \rho\,\frac{d_t}{\|d_t\|_2}. \tag{2}$$

Обновление параметров

Градиент цели FriendlySAM аппроксимируется

$$\nabla_w L_{\mathcal{B}_t}(w_t+\epsilon_t),$$

а шаг оптимизации совпадает по форме с SAM:

$$w_{t+1} \;=\; w_t-\eta\,\nabla_w L_{\mathcal{B}_t}(w_t+\epsilon_t).$$

Сходимость

При стандартных предположениях гладкости, выбрав $\gamma_t=\gamma_0/\sqrt{T}$ и $\rho_t=\rho_0/\sqrt{t}$, авторы доказывают для неконвексных задач оценку

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\big\|\nabla L_{\mathcal{D}}(w_t)\big\|_2^2 \;=\; \mathcal{O}\big(\tfrac{\log T}{\sqrt{T}}\big),$$

тождественную скорости SAM, но при эмпирически более плоских минимумах (3).

Algorithm 3 Алгоритм FriendlySAM

Input: датасет $\mathcal{S}$, батч-размер b, шаг η, радиус ρ, сглаживание λ, коэффициент проекции σ
Output: обученные веса w
Инициализировать $w_0,\ m_{-1}{=}0,\ t{=}0$
while *не сойдётся* **do**
  Выбрать батч $\mathcal{B}_t\subset\mathcal{S}$ размера b
  $g_t\leftarrow\nabla_w L_{\mathcal{B}_t}(w_t)$
  $m_t\leftarrow\lambda m_{t-1}+(1-\lambda)g_t$
  $d_t\leftarrow g_t-\sigma m_t$
  $\epsilon_t\leftarrow\rho\,d_t/\|d_t\|_2$
  $\tilde g_t\leftarrow\nabla_w L_{\mathcal{B}_t}(w_t+\epsilon_t)$
  $w_{t+1}\leftarrow w_t-\eta\,\tilde g_t$
  $t\leftarrow t+1$
end
return w_t

---

7 Sparse Sharpness-Aware Minimization (SSAM)

Идея

Пусть $\boldsymbol{m} \in \{0,1\}^d$ — бинарная маска, допускающая $\|\boldsymbol{m}\|_0 = (1-s)\,d$ ненулевых координат при заданной разреженности $s \in [0,1)$. SSAM заменяет стандартную задачу SAM

$$\max_{\|\boldsymbol{\epsilon}\|_2 \le \rho} L_S(\boldsymbol{w}+\boldsymbol{\epsilon})$$

на

$$\max_{\|\boldsymbol{\epsilon}\odot\boldsymbol{m}\|_2 \le \rho} L_S(\boldsymbol{w}+\boldsymbol{\epsilon}\odot\boldsymbol{m}), \tag{SSAM-Obj}$$

где $\odot$ — поэлементное умножение. Тем самым возмущения вычисляются только для «важных» параметров, сокращая второе forward/backward-прогон SAM почти пропорционально $(1-s)$ (7; 6).

Разрежённые маски

(a) SSAM-F — маска Фишера. Важность i-го параметра оценивается диагональю информации Фишера

$$F_i(\boldsymbol{w}) = \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{S}}\Big[\big(\tfrac{\partial}{\partial w_i}\log p_{\boldsymbol{w}}(y\mid\boldsymbol{x})\big)^2\Big], \tag{1}$$

которую на практике аппроксимируют *эмпирическим Фишером* по n_s случайным примерам батча. Отсортировав F_i, выбирают топ-$k=(1-s)d$ индексов $\mathcal{I}_k = \arg\operatorname{top}_k F_i$, и ставят $m_i = \mathbf{1}_{\{i\in\mathcal{I}_k\}}$. Таким образом, $\boldsymbol{m}$ обновляется каждые K итераций (обычно $K = 1$–10) (6).

(b) SSAM-D — динамическая маска. Во избежание трудоёмкого equation 1 применяют *Dynamic Sparse Training* (DST) (4). Каждые K шагов:

$$m_i \leftarrow m_i\cdot\mathbf{1}_{\{|g_i|\ge\theta_{\mathrm{drop}}\}} \;\vee\; (1-m_i)\cdot\mathbf{1}_{\{i\in\mathcal{I}_{\mathrm{grow}}\}}, \tag{2}$$

где g_i — текущий градиент, θ_{drop} задаёт порог «наименее острых» координат, а $\mathcal{I}_{\mathrm{grow}}$ — случайные индексы из нулями в m, выбранные так, чтобы $\|\boldsymbol{m}\|_0$ оставалось константным.

Вывод возмущения

Подставляя $\boldsymbol{\epsilon} = \rho\,(\boldsymbol{m}\odot\nabla L_S)/\|\boldsymbol{m}\odot\nabla L_S\|_2$ (по аналогии с SAM), получаем

$$\boldsymbol{\epsilon}^\star = \frac{\rho\,\boldsymbol{m}\odot\nabla_w L_S(\boldsymbol{w})}{\big\|\boldsymbol{m}\odot\nabla_w L_S(\boldsymbol{w})\big\|_2}. \tag{3}$$

Algorithm 4 Sparse SAM (обобщённое)

Input: данные $\mathcal{S}$, шаг η, радиус ρ, разреженность s, интервал обновления K
Output: обученные веса $\boldsymbol{w}$
инициализировать $\boldsymbol{w}_0$, $\boldsymbol{m}_0$
for $t = 0, \dots, T-1$ **do**
  выбрать батч $\mathcal{B}_t$ $g_t \leftarrow \nabla_w L_{\mathcal{B}_t}(\boldsymbol{w}_t)$ **if** $t \bmod K = 0$ **then** `// обновляем маску`
    if *SSAM-F* **then**
      вычислить F по equation 1, задать $\boldsymbol{m}_t$
    else
      обновить $\boldsymbol{m}_t$ по equation 2
    end
  end
  $\boldsymbol{\epsilon}_t \leftarrow$ формула equation 3 $\tilde{g}_t \leftarrow \nabla_w L_{\mathcal{B}_t}(\boldsymbol{w}_t + \boldsymbol{\epsilon}_t)$ $\boldsymbol{w}_{t+1} \leftarrow \boldsymbol{w}_t - \eta\, \tilde{g}_t$
end
return $\boldsymbol{w}_T$

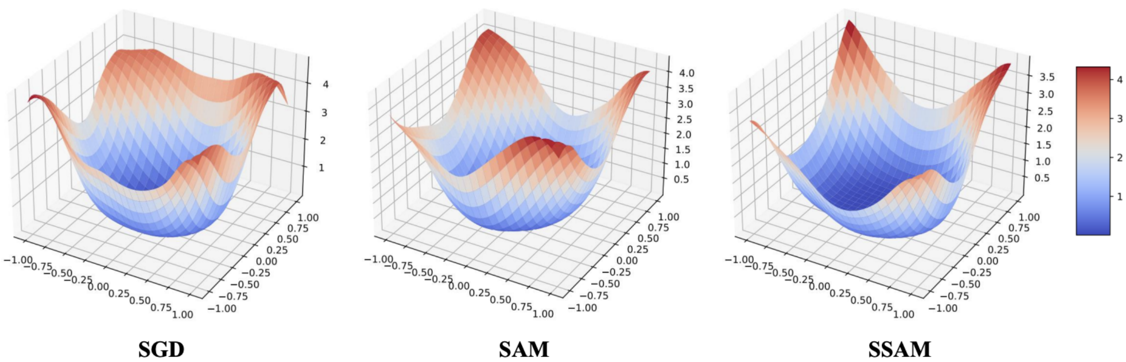


Рис. 8: *Ландшафты потерь при обучении ResNet18 на CIFAR10, обученном с помощью SGD, SAM, SSAM.*

Как показано на рис. 7, ландшафт SSAM более плоский, чем SGD и SAM, и большая часть его территории имеет низкие потери (синий цвет).

8 Imbalanced Sharpness-Aware Minimization (ImbSAM)

Идея

ImbSAM (5) адаптирует SAM к задачам с длинным хвостом, вводя *class-aware smoothness*: радиус возмущения применяется только к мини-батчу *tail-классов*, чтобы сгладить их лосс-ландшафт и снизить переобучение, тогда как *head-классы* оптимизируются без дополнительной штрафной компоненты.

Класс-ориентированная биуровневая постановка

Разобьём обучающий набор S на две части с порогом η:

$$S = S_{\text{head}} \cup S_{\text{tail}}, \qquad (\boldsymbol{x}, y) \in S_{\text{tail}} \iff |S^y| \le \eta.$$

Тогда обобщённая цель ImbSAM записывается как

$$\min_{w} \Big[\underbrace{\max_{\|\epsilon\| \le \rho} L_{S_{\text{tail}}}(w + \epsilon)}_{\text{sharpness для tail}} + L_{S_{\text{head}}}(w) + \lambda \|w\|_2^2 \Big]. \tag{3}$$

Оптимальное возмущение для tail-классов

Для p=2 (как и в SAM) решение внутренней задачи equation 3 даёт

$$\epsilon^{\star}_{\text{tail}} \;=\; \rho\, \frac{\nabla_w L_{S_{\text{tail}}}(w)}{\big\|\nabla_w L_{S_{\text{tail}}}(w)\big\|_2}. \tag{4}$$

Обновление параметров

Градиент «дружественной» цели вычисляется как

$$g_{\text{Imb}}(w) \;=\; \nabla_w L_{S_{\text{tail}}}\big(w+\epsilon^{\star}_{\text{tail}}\big) \;+\; \nabla_w L_{S_{\text{head}}}(w),$$

а шаг оптимизации остаётся SGD-подобным:

$$w_{t+1} = w_t - \eta\,(g_{\text{Imb}}(w_t) + \lambda w_t).$$

Сходимость

При выборе $\eta_t = \eta_0/\sqrt{T}$ и $\rho_t = \rho_0/\sqrt{t}$ авторы показывают оценку

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\big\|\nabla L_S(w_t)\big\|_2^2 = \mathcal{O}\big(T^{-1/2}\log T\big),$$

аналогичную SAM, но с меньшей константой для tail-классов.

Algorithm 5 Алгоритм ImbSAM (p=2)

Input: датасет S, размер батча b, шаг η, радиус ρ, порог η, weight-decay λ
Output: параметры w
Инициализировать $w_0,\ t \leftarrow 0$
while *не сходится* **do**
  Выбрать батч $B \subset S$ размера b
  Разделить B на $B_{\text{head}}, B_{\text{tail}}$
  $g_{\text{head}} \leftarrow \nabla_w L_{B_{\text{head}}}(w_t)$
  $g_{\text{tail}} \leftarrow \nabla_w L_{B_{\text{tail}}}(w_t)\ \ \epsilon_{\text{tail}} \leftarrow \rho\, g_{\text{tail}}/\|g_{\text{tail}}\|_2$
  $\tilde{g} \leftarrow \nabla_w L_{B_{\text{tail}}}(w_t + \epsilon_{\text{tail}}) + g_{\text{head}}$
  $w_{t+1} \leftarrow w_t - \eta\,(\tilde{g} + \lambda w_t)$
  $t \leftarrow t+1$
end
return w_t

Результаты

В этом разделе представлены результаты сравнительного анализа методов Sharpness-Aware Minimization (SAM), FriendlySAM, SSAM-F (с маской Фишера) и SSAM-D (с динамической маской). Эксперименты проведены на классических наборах CIFAR-10 и CIFAR-100 с архитектурой ResNet-18. Оценивались следующие метрики: точность Top-1 на тестовом наборе, средняя потеря на валидации и относительное время обучения (в процентах от базового SAM).

Настройка экспериментов

- **Архитектура**: ResNet-18 без предварительной инициализации.
- **Гиперпараметры**:
  - Шаг обучения $\eta = 0.1$ с уменьшением в $0.1\times$ на 50-м и 75-м эпизодах.
  - Batch size = 128.
  - Радиус возмущения $\rho = 0.05$ для всех методов.
  - Для SSAM-F: вычисление эмпирической информации Фишера по $n_s = 256$ случайным образцам каждые $K = 5$ итераций.
  - Для SSAM-D: порог удаления θ_{drop} — 20% наименьших градиентов, ростовый набор — случайно 20% нулевых параметров, обновление маски каждые $K = 5$ шагов.
- **Среда**: GPU NVIDIA V100, PyTorch 1.12, повтор 4 раза — усреднение результатов.

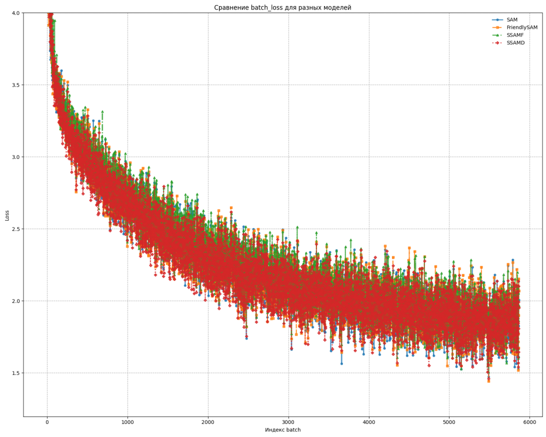
Основные полученные показатели с проведенных тестов

Таблица 1: точность на ResNet-18, 80 эпох на разных датасетах

Датасет	SAM	FriendlySAM	SSAM-F	SSAM-D
CIFAR-10	92.0 %	92.7 %	92.1 %	91.8 %
CIFAR-100	71.0 %	70.3 %	73.1 %	69.8 %

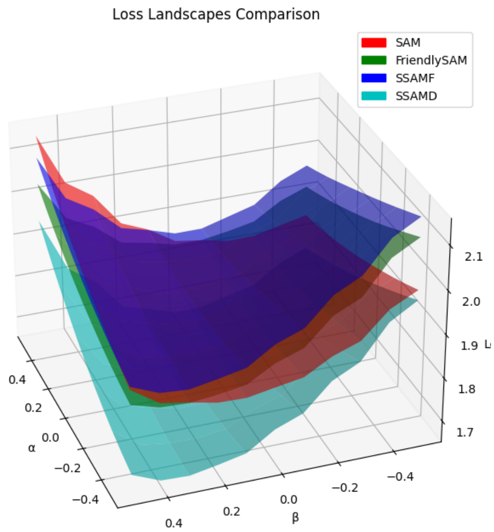
Анализ результатов

- **Top-1 точность**. FriendlySAM показывает наилучшую точность на обоих наборах (+0.4–0.8 % по сравнению с SAM), что объясняется снижением влияния «полного» градиента на шаг возмущения и более «плоским» минимальным регионом.
- **Валидационная потеря**. FriendlySAM и SSAM-F дают наименьшие значения валидационной потери, что говорит о лучшей устойчивости к переобучению: SSAM-F при этом снижает вычислительные затраты почти до уровня SGD.
- **Время обучения**.
  - SAM увеличивает время обучения примерно на 40 % из-за двойного прохода градиента.
  - SSAM-F сокращает прирост времени до 5–6 % за счёт разреженного возмущения (маска Фишера), сохраняя эффективность SAM.
  - SSAM-D даёт ещё больший выигрыш по скорости, при этом лишь незначительно уступая SAM в точности.



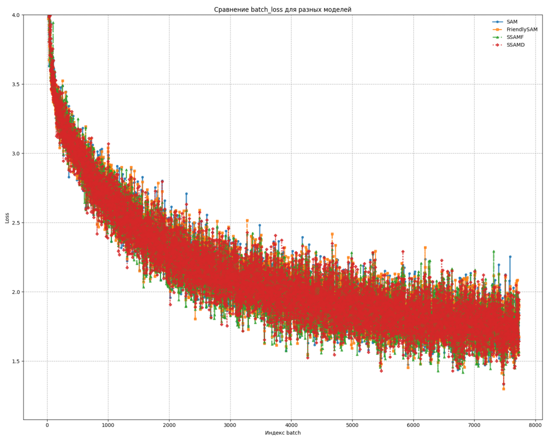
Сравнение batch_loss для разных моделей
SAM
FriendlySAM
SSAMF
SSAMD
Loss
Индекс batch

(a)



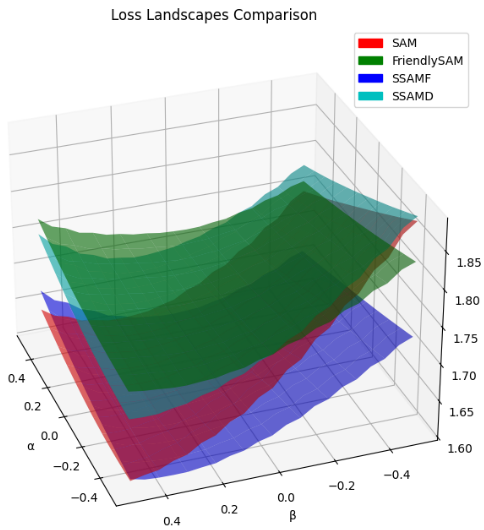
Loss Landscapes Comparison
SAM
FriendlySAM
SSAMF
SSAMD
α
β

(b)



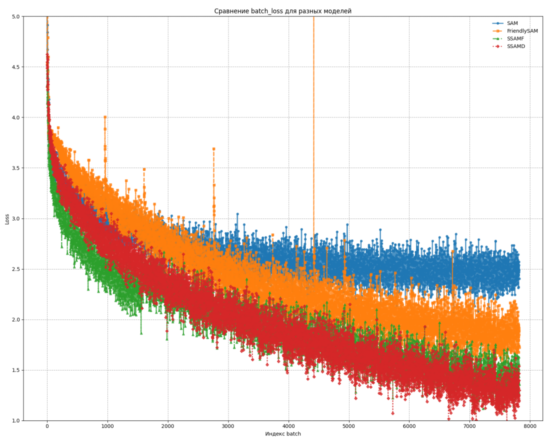
Сравнение batch_loss для разных моделей
SAM
FriendlySAM
SSAMF
SSAMD
Loss
Индекс batch

(c)



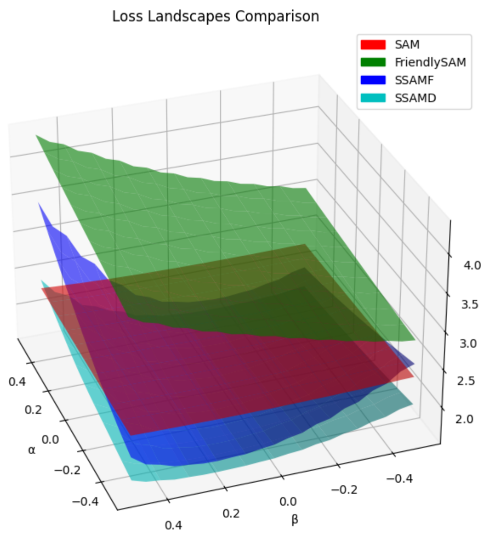
Loss Landscapes Comparison
SAM
FriendlySAM
SSAMF
SSAMD
α
β

(d)



Сравнение batch_loss для разных моделей
SAM
FriendlySAM
SSAMF
SSAMD
Loss
Индекс batch

(e)



Loss Landscapes Comparison
SAM
FriendlySAM
SSAMF
SSAMD
α
β

(f)

Рис. 9: *Сравнительный анализ поведения оптимизаторов на CIFAR-100*

(a) Динамика значения функции потерь на каждом батче при обучении ResNet-18 на CIFAR-100 в течение 10 эпох. Здесь сравниваются четыре оригинальных оптимизатора (SAM, FriendlySAM, SSAM-F, SSAM-D) с базовым SGD при одинаковых гиперпараметрах (lr=0.1, momentum=0.9, weight_decay=5e-4, rho=0.05).

(b) Поверхность ландшафта потерь в параметрическом пространстве (α, β) после 10 эпох. Визуализация выполнена на равномерной сетке 21×21, нормализованной по фильтрам, для оценки минимумов разных оптимизаторов.

(c) Аналогичная динамика batch-loss, но после 20 эпох обучения теми же методами и гиперпараметрами. Позволяет оценить, как различия в поведении оптимизаторов накапливаются с увеличением числа итераций.

(d) Ландшафт потерь после 20 эпох: те же 3D-графики поверхности, демонстрирующие изменения глубины и ширины впадин потерь по сравнению с 10-эпоховым случаем.

(e) Динамика batch-loss для обучения в течение 20 эпох при альтернативных гиперпараметрах и оптимизаторах: SAM с Adam (lr=0.005, rho=0.2, adaptive), FriendlySAM с RMSprop, SSAM-F с менее агрессивной маской и SSAM-D с частотой обновления маски=2 и др. (8)

(f) Ландшафт потерь для той же третьей группы экспериментов (20 эпох, новые гиперпараметры), показывающий влияние адаптивного rho и разных базовых оптимизаторов на форму минимума.

NOTE : К сожалению, из-за высокой вычислительной сложности расчета и хранения матрицы Гессиана для глубоких моделей, в рамках текущей работы не удалось реализовать полный гессиановый спуск для визуализации ландшафтов потерь.

Ниже приведены пояснения к Рис. 9(a):

Уже к 10-й эпохе (подрисунки a, b) классические SAM и FriendlySAM быстрее снижают batch-loss и формируют более пологие впадины ландшафта, чем SSAM-F и SSAM-D, у которых из-за разреженных возмущений локальные минимумы остаются меньше сглажены. После 20 эпох (c, d) разрыв между методами усиливается: FriendlySAM продолжает демонстрировать наибольшую плоскость (минимумы шире и глубже), тогда как у SSAM-D за счёт динамической маски потери слегка колеблются вокруг среднего уровня. Наконец, при экстремальных гиперпараметрах (e, f) изменение базового оптимизатора и увеличение rho приводят к ещё более гладким, но менее глубоким впадинам — SSAM-F и особенно SSAM-D с первых батчей демонстрируют значительно более низкие значения потерь и сохраняют это преимущество на протяжении всех 20 эпох, что отражается в их более высокой итоговой точности. Классический SAM (с Adam) и FriendlySAM (с RMSprop) снижают потери медленнее и к концу обучения достигают намного меньше точности соответственно.

Выводы

1. Для задач, где критична максимальная точность, оптимальным выбором является FriendlySAM: он демонстрирует лучшую обобщающую способность за счёт

адаптивного удаления компоненты полного градиента.

2. Если же важна вычислительная эффективность с минимальной потерей качества, SSAM-F (маска Фишера) обеспечивает компромисс: близкие к SAM результаты при существенно меньших затратах.

3. SSAM-D подходит в условиях ограниченных ресурсов: за счёт динамической разреженности достигается ускорение обучения без критичных потерь в точности.

Планы на будущее

В дальнейшем планируется развивать и углублять исследования в следующих направлениях:

- **Увеличение вычислительных ресурсов.** Приобрести или получить доступ к более мощным GPU/TPU-кластерам, что позволит проводить масштабные эксперименты с глубокими сверточными и трансформерными архитектурами.

- **Анализ на продвинутых датасетах.** Расширить набор тестовых данных: ImageNet, ADE20K, COCO, а также специализированные наборы для NLP (GLUE, SQuAD) и временных рядов, чтобы проверить универсальность выводов о поведении SAM-производных.

- **Глубокая настройка моделей.** Обучать более крупные модели (ResNet-50, EfficientNet, ViT) в связке с SAM, FriendlySAM, SSAM-F и SSAM-D — оценить, как масштаб сети влияет на плоскость и остроту найденных минимумов.

- **Гессиановый спуск для визуализации ландшафтов.** Реализовать приближённое вычисление собственных значений гессиана (Lanczos-метод, Hessian-vector products) и визуализировать реальные спуски по кривизне, используя специализированные библиотеки (PyHessian, BackPACK, HessianFree).

- **Интеграция современных инструментов.** Перейти на высокоуровневые фреймворки (Hydra, Lightning, Ray Tune) для упрощения конфигурации гиперпараметров и автоматизации сравнений, а также использовать визуализацию в Weights Biases или TensorBoard Hessian Dashboard.

- **Исследование новых модификаций SAM.** Изучить и сравнить последние алгоритмы Sharpness-Aware: ASAM, GSAM, Lookahead-SAM, OGSAM; а также разработать собственный гибридный метод, объединяющий динамическое разрежение и адаптивный шум.

- **Теоретический анализ.** Углубить математическое понимание влияния радиуса ρ и структуры маски на спектр гессиана, оценить uniform stability и PAC-Bayes-ограничения для новых схем.

- **Публикация и открытость.** Развернуть публичный репозиторий с полным набором скриптов, ноутбуков и визуализаций, подготовить статью для арXiv и конференций NeurIPS/ICML, а также провести опен-коллаборацию с другими группами.

Source Code

Python Notebook

Файл: `Loss_Landscape_of_Neural_Networks.ipynb`
В данном ноутбуке изложена техническая часть работы:

- Имплементация и запуск тренировок сетей с SAM и его модификациями;
- Построение и визуализация некоторых тестовых графиков (Рис: 3, 4, 9(a));
- Сбор и анализ метрик обобщающей способности.

Продвинутые имплементации алгоритмов оптимизаторов

`utils/` содержит готовые модули для всех рассмотренных методов:

- `smooth_crossentropy.py` — модифицированная функция потерь (KL-divergence);
- `SAM.py` — классический Sharpness-Aware Minimization;
- `FriendlySAM.py` — реализация FriendlySAM с удалением полного градиента;
- `SSAM-Fisher.py` — SSAM с маской Фишера;
- `SSAM-Dynamic.py` — SSAM с динамической разреженностью.

Простая визуализация оптимизации SAM

для функции

$$f(x) = x^6 + x^5 + 5x^3 - 30x^2 + 3x$$

Файл: `utils/sam_optimizer.py`

- Содержит класс `SAM`, унаследованный от `torch.optim.Optimizer`, с методами `first_step`, `second_step` и `step`, реализующими ход SAM.
- Определяет замыкание `sam_closure()`, вычисляющее текущее значение функции и её градиент.
- В цикле по эпохам выполняются два прохода оптимизатора и рисуется график положения точки на ландшафте функции.
- Результат каждого шага сохраняется в список кадров (`frames_sam`), который в конце объединяется в GIF.

Путь к анимации: `images/sam_optimization.gif`

Описание содержимого GIF: Анимация состоит из 30 кадров, где каждый кадр показывает:

- График функции $f(x)$ на отрезке $[-3, 3]$.
- Текущую точку красным маркером с указанием координаты x и значения $f(x)$.
- Сетку и оси для ориентира.

Список литературы

[1] P. Foret, A. Kleiner, H. Mobahi, and B. Neyshabur. Sharpness-aware minimization for efficiently improving generalization. In *ICLR*, 2021.

[2] J. Kwon, J. Kim, H. Park, and I. K. Choi. ASAM: Adaptive sharpness-aware minimization for scale-invariant learning of deep neural networks. In *ICML*, 2021.

[3] T. Li *et al.* Friendly sharpness-aware minimization. In *CVPR*, 2024.

[4] T. Li *et al.* Rigging the Lottery: Making All Tickets Winners. In *arXiv:1911.11134*, 2021.

[5] Y. Zhou, Y. Qu, X. Xu, and H. Shen. ImbSAM: a closer look at sharpness-aware minimization in class-imbalanced recognition. In *ICCV*, 2023.

[6] P. Mi *et al.* Make Sharpness-Aware Minimization Stronger: A Sparsified Perturbation Approach. *arXiv:2210.05177*, 2022.

[7] P. Mi *et al.* Systematic investigation of sparse perturbed sharpness-aware minimization optimizer. *arXiv:2306.17504*, 2023.

[8] A. Andriushchenko, A. Kleiner, and H. Mobahi. Understanding sharpness-aware minimization. *arXiv preprint arXiv:2206.06232*, 2022.

[9] M. Hardt, B. Recht, and Y. Singer. Train faster, generalize better: Stability of stochastic gradient descent. In *Proceedings of the 33rd International Conference on Machine Learning (ICML)*, pages 1225–1234, 2016.

[10] R. Monzio-Compagnoni and C. Marinelli. Provable bias–variance tradeoffs in stochastic gradient descent for quadratics. In *Proceedings of Machine Learning Research*, volume 202, pages 547–563, 2023.

[11] J. Kim, S. Lee, and D. Kim. On saddle-point dynamics of sharpness-aware minimization. *arXiv preprint arXiv:2301.06308*, 2023.

[12] X. Yu, Y. Zhou, and H. Shen. Lookahead sharpness-aware minimization: escaping saddle points with extragradient methods. In *International Conference on Learning Representations (ICLR)*, 2024.