Shapeless-Aware Minimization

for Efficiently Improving Generalization

Сафонов Иван (докладчик)
Рахматуллин Рамазан (рецензент)
Терехова Юлия (практик-исследователь)
Сухоросов Алексей (хакер)
БПМИ 182

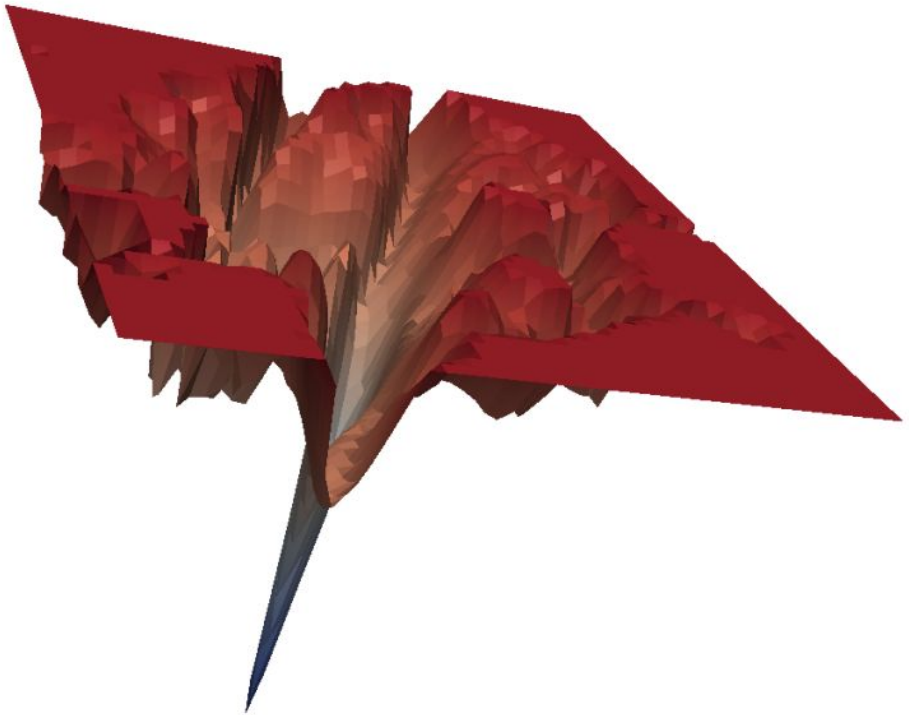
06.10.2021

План

- Мотивация метода
- Описание метода и алгоритма
- Результаты
- Исследования вокруг алгоритма

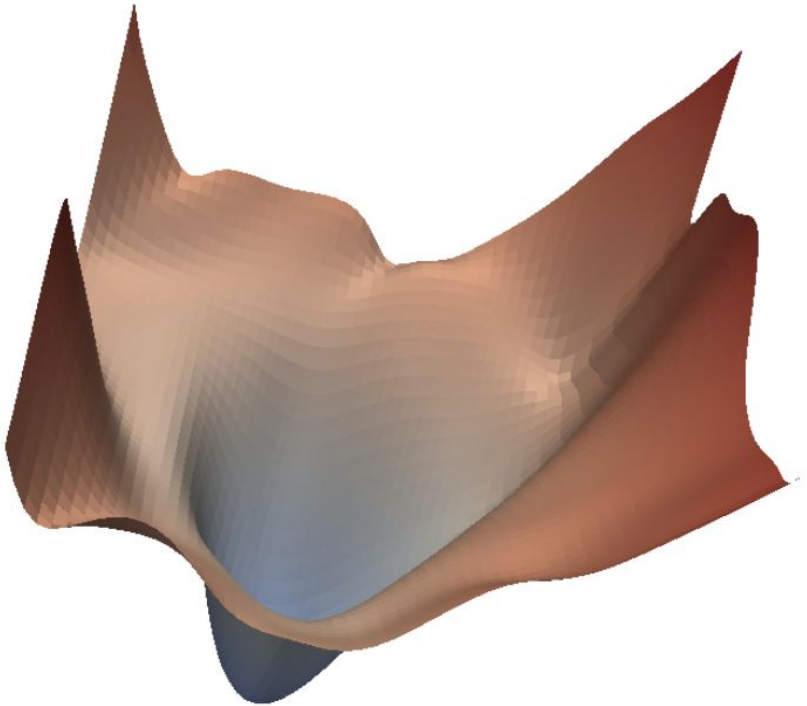
Мотивация

- loss функция крайне сложна и имеет множество точек оптимума
- В разных точках модель получает разные generalization свойства
- Выбор оптимизатора/регуляризации является очень важным шагом



Идея

- Давайте кроме самого loss минимизировать его заостренность в окрестности точки
- Тогда (как утверждается), наш алгоритм не сможет упасть в какой-то локальный минимум, приводящий к переобучению



Определения

$$\mathcal{S} \triangleq \cup_{i=1}^{n} \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}$$ - выборка

$$l : \mathcal{W} \times \mathcal{X} \times \mathcal{Y} \to \mathbb{R}_+$$ - loss function

$$L_S(\boldsymbol{w}) \triangleq \frac{1}{n} \sum_{i=1}^{n} l(\boldsymbol{w}, \boldsymbol{x}_i, \boldsymbol{y}_i)$$ - training set loss

$$L_{\mathscr{D}}(\boldsymbol{w}) \triangleq \mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})\sim D}[l(\boldsymbol{w}, \boldsymbol{x}, \boldsymbol{y})]$$ - population loss

Основная теорема

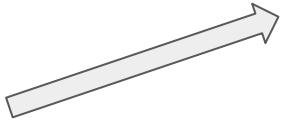
Для любого $\rho > 0$ с большой вероятностью при генерации случайной выборки:

$$L_{\mathscr{D}}(\boldsymbol{w}) \leq \max_{\|\boldsymbol{\epsilon}\|_2 \leq \rho} L_{\mathcal{S}}(\boldsymbol{w} + \boldsymbol{\epsilon}) + h(\|\boldsymbol{w}\|_2^2 / \rho^2)$$

$h : \mathbb{R}_+ \to \mathbb{R}_+$ - некоторая строго возрастающая функция

SAM

$$\underbrace{[\max_{\|\epsilon\|_2 \leq \rho} L_{\mathcal{S}}(\boldsymbol{w}+\boldsymbol{\epsilon}) - L_{\mathcal{S}}(\boldsymbol{w})]} + L_{\mathcal{S}}(\boldsymbol{w}) + h(\|\boldsymbol{w}\|_2^2/\rho^2)$$



метрика заостренности

SAM optimization problem:

$$\min_{\boldsymbol{w}} L_{\mathcal{S}}^{SAM}(\boldsymbol{w}) + \lambda||\boldsymbol{w}||_2^2 \quad \text{где} \quad L_{\mathcal{S}}^{SAM}(\boldsymbol{w}) \triangleq \max_{||\boldsymbol{\epsilon}||_p \leq \rho} L_S(\boldsymbol{w}+\boldsymbol{\epsilon})$$

Решение задачи оптимизации

$$\boldsymbol{\epsilon}^*(\boldsymbol{w}) \triangleq \underset{\|\boldsymbol{\epsilon}\|_p \leq \rho}{\arg\max}\, L_{\mathcal{S}}(\boldsymbol{w}+\boldsymbol{\epsilon}) \approx \underset{\|\boldsymbol{\epsilon}\|_p \leq \rho}{\arg\max}\, L_{\mathcal{S}}(\boldsymbol{w}) + \boldsymbol{\epsilon}^T \nabla_{\boldsymbol{w}} L_{\mathcal{S}}(\boldsymbol{w}) = \underset{\|\boldsymbol{\epsilon}\|_p \leq \rho}{\arg\max}\, \boldsymbol{\epsilon}^T \nabla_{\boldsymbol{w}} L_{\mathcal{S}}(\boldsymbol{w})$$

Решение:

$$\hat{\boldsymbol{\epsilon}}(\boldsymbol{w}) = \rho\, \mathrm{sign}\left(\nabla_{\boldsymbol{w}} L_{\mathcal{S}}(\boldsymbol{w})\right) \left|\nabla_{\boldsymbol{w}} L_{\mathcal{S}}(\boldsymbol{w})\right|^{q-1} / \left(\left\|\nabla_{\boldsymbol{w}} L_{\mathcal{S}}(\boldsymbol{w})\right\|_q^q\right)^{1/p}$$

для p = 2: градиент, нормированный на длину ρ

Решение задачи оптимизации

$$\nabla_{\boldsymbol{w}} L_{\mathcal{S}}^{SAM}(\boldsymbol{w}) \approx \nabla_{\boldsymbol{w}} L_{\mathcal{S}}(\boldsymbol{w} + \hat{\boldsymbol{\epsilon}}(\boldsymbol{w}))$$

$$\nabla_{\boldsymbol{w}} L_{\mathcal{S}}(\boldsymbol{w} + \hat{\boldsymbol{\epsilon}}(\boldsymbol{w})) = \frac{d(\boldsymbol{w} + \hat{\boldsymbol{\epsilon}}(\boldsymbol{w}))}{d\boldsymbol{w}} \nabla_{\boldsymbol{w}} L_{\mathcal{S}}(\boldsymbol{w})|_{\boldsymbol{w}+\hat{\boldsymbol{\epsilon}}(w)} =$$

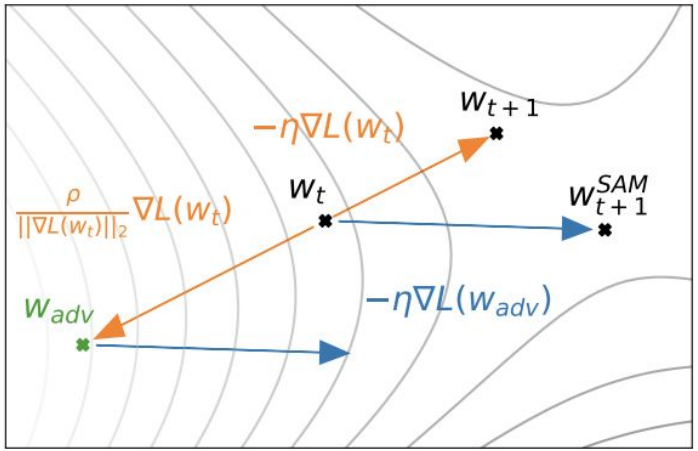
$$= \nabla_{w} L_{\mathcal{S}}(\boldsymbol{w})|_{\boldsymbol{w}+\hat{\boldsymbol{\epsilon}}(\boldsymbol{w})} + \frac{d\hat{\boldsymbol{\epsilon}}(\boldsymbol{w})}{d\boldsymbol{w}} \nabla_{\boldsymbol{w}} L_{\mathcal{S}}(\boldsymbol{w})|_{\boldsymbol{w}+\hat{\boldsymbol{\epsilon}}(\boldsymbol{w})}$$

В результате, для простоты:

$$\nabla_{\boldsymbol{w}} L_{\mathcal{S}}^{SAM}(\boldsymbol{w}) \approx \nabla_{\boldsymbol{w}} L_{\mathcal{S}}(w)|_{\boldsymbol{w}+\hat{\boldsymbol{\epsilon}}(\boldsymbol{w})}$$

Итоговый алгоритм:

Input: Training set $\mathcal{S} \triangleq \cup_{i=1}^{n}\{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}$, Loss function $l : \mathcal{W} \times \mathcal{X} \times \mathcal{Y} \to \mathbb{R}_+$, Batch size b, Step size $\eta > 0$, Neighborhood size $\rho > 0$.
Output: Model trained with SAM
Initialize weights $\boldsymbol{w}_0$, $t = 0$;
while *not converged* **do**
  Sample batch $\mathcal{B} = \{(\boldsymbol{x}_1, \boldsymbol{y}_1), ...(\boldsymbol{x}_b, \boldsymbol{y}_b)\}$;
  Compute gradient $\nabla_{\boldsymbol{w}} L_{\mathcal{B}}(\boldsymbol{w})$ of the batch's training loss;
  Compute $\hat{\boldsymbol{\epsilon}}(\boldsymbol{w})$ per equation 2;
  Compute gradient approximation for the SAM objective (equation 3): $\boldsymbol{g} = \nabla_w L_{\mathcal{B}}(\boldsymbol{w})|_{\boldsymbol{w}+\hat{\boldsymbol{\epsilon}}(\boldsymbol{w})}$;
  Update weights: $\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \eta \boldsymbol{g}$;
  $t = t + 1$;
end
return $\boldsymbol{w}_t$



Результаты, CIFAR-10, 100

		CIFAR-10		CIFAR-100	
Model	Augmentation	SAM	SGD	SAM	SGD
WRN-28-10 (200 epochs)	Basic	**2.7**$_{\pm 0.1}$	3.5$_{\pm 0.1}$	**16.5**$_{\pm 0.2}$	18.8$_{\pm 0.2}$
WRN-28-10 (200 epochs)	Cutout	**2.3**$_{\pm 0.1}$	2.6$_{\pm 0.1}$	**14.9**$_{\pm 0.2}$	16.9$_{\pm 0.1}$
WRN-28-10 (200 epochs)	AA	**2.1**$_{\pm <0.1}$	2.3$_{\pm 0.1}$	**13.6**$_{\pm 0.2}$	15.8$_{\pm 0.2}$
WRN-28-10 (1800 epochs)	Basic	**2.4**$_{\pm 0.1}$	3.5$_{\pm 0.1}$	**16.3**$_{\pm 0.2}$	19.1$_{\pm 0.1}$
WRN-28-10 (1800 epochs)	Cutout	**2.1**$_{\pm 0.1}$	2.7$_{\pm 0.1}$	**14.0**$_{\pm 0.1}$	17.4$_{\pm 0.1}$
WRN-28-10 (1800 epochs)	AA	**1.6**$_{\pm 0.1}$	2.2$_{\pm <0.1}$	**12.8**$_{\pm 0.2}$	16.1$_{\pm 0.2}$
Shake-Shake (26 2x96d)	Basic	**2.3**$_{\pm <0.1}$	2.7$_{\pm 0.1}$	**15.1**$_{\pm 0.1}$	17.0$_{\pm 0.1}$
Shake-Shake (26 2x96d)	Cutout	**2.0**$_{\pm <0.1}$	2.3$_{\pm 0.1}$	**14.2**$_{\pm 0.2}$	15.7$_{\pm 0.2}$
Shake-Shake (26 2x96d)	AA	**1.6**$_{\pm <0.1}$	1.9$_{\pm 0.1}$	**12.8**$_{\pm 0.1}$	14.1$_{\pm 0.2}$
PyramidNet	Basic	**2.7**$_{\pm 0.1}$	4.0$_{\pm 0.1}$	**14.6**$_{\pm 0.4}$	19.7$_{\pm 0.3}$
PyramidNet	Cutout	**1.9**$_{\pm 0.1}$	2.5$_{\pm 0.1}$	**12.6**$_{\pm 0.2}$	16.4$_{\pm 0.1}$
PyramidNet	AA	**1.6**$_{\pm 0.1}$	1.9$_{\pm 0.1}$	**11.6**$_{\pm 0.1}$	14.6$_{\pm 0.1}$
PyramidNet+ShakeDrop	Basic	**2.1**$_{\pm 0.1}$	2.5$_{\pm 0.1}$	**13.3**$_{\pm 0.2}$	14.5$_{\pm 0.1}$
PyramidNet+ShakeDrop	Cutout	**1.6**$_{\pm <0.1}$	1.9$_{\pm 0.1}$	**11.3**$_{\pm 0.1}$	11.8$_{\pm 0.2}$
PyramidNet+ShakeDrop	AA	**1.4**$_{\pm <0.1}$	1.6$_{\pm <0.1}$	**10.3**$_{\pm 0.1}$	10.6$_{\pm 0.1}$

Детали экспериментов

- Гиперпараметр ρ искали с помощью grid-search
- В качестве baseline запускали метод, давая ему в 2 раза больше эпох для обучения. При этом берется лучший результат среди такого же количества эпох и в 2 раза большего
- 5 независимых запусков для каждого из случаев

Результаты, ImageNet

Model	Epoch	SAM		Standard Training (No SAM)	
		Top-1	Top-5	Top-1	Top-5
ResNet-50	100	**22.5**$_{\pm 0.1}$	6.28$_{\pm 0.08}$	22.9$_{\pm 0.1}$	6.62$_{\pm 0.11}$
	200	**21.4**$_{\pm 0.1}$	5.82$_{\pm 0.03}$	22.3$_{\pm 0.1}$	6.37$_{\pm 0.04}$
	400	**20.9**$_{\pm 0.1}$	5.51$_{\pm 0.03}$	22.3$_{\pm 0.1}$	6.40$_{\pm 0.06}$
ResNet-101	100	**20.2**$_{\pm 0.1}$	5.12$_{\pm 0.03}$	21.2$_{\pm 0.1}$	5.66$_{\pm 0.05}$
	200	**19.4**$_{\pm 0.1}$	4.76$_{\pm 0.03}$	20.9$_{\pm 0.1}$	5.66$_{\pm 0.04}$
	400	**19.0**$_{\pm <0.01}$	4.65$_{\pm 0.05}$	22.3$_{\pm 0.1}$	6.41$_{\pm 0.06}$
ResNet-152	100	**19.2**$_{\pm <0.01}$	4.69$_{\pm 0.04}$	20.4$_{\pm <0.0}$	5.39$_{\pm 0.06}$
	200	**18.5**$_{\pm 0.1}$	4.37$_{\pm 0.03}$	20.3$_{\pm 0.2}$	5.39$_{\pm 0.07}$
	400	**18.4**$_{\pm <0.01}$	4.35$_{\pm 0.04}$	20.9$_{\pm <0.0}$	5.84$_{\pm 0.07}$

Результаты, finetuning, pretrained on ImageNet

Dataset	EffNet-b7 + SAM	EffNet-b7	Prev. SOTA (ImageNet only)	EffNet-L2 + SAM	EffNet-L2	Prev. SOTA
FGVC_Aircraft	$6.80_{\pm 0.06}$	$8.15_{\pm 0.08}$	**5.3** (TBMSL-Net)	$\mathbf{4.82}_{\pm 0.08}$	$5.80_{\pm 0.1}$	5.3 (TBMSL-Net)
Flowers	$\mathbf{0.63}_{\pm 0.02}$	$1.16_{\pm 0.05}$	0.7 (BiT-M)	$\mathbf{0.35}_{\pm 0.01}$	$0.40_{\pm 0.02}$	0.37 (EffNet)
Oxford_IIIT_Pets	$\mathbf{3.97}_{\pm 0.04}$	$4.24_{\pm 0.09}$	4.1 (Gpipe)	$\mathbf{2.90}_{\pm 0.04}$	$3.08_{\pm 0.04}$	4.1 (Gpipe)
Stanford_Cars	$5.18_{\pm 0.02}$	$5.94_{\pm 0.06}$	**5.0** (TBMSL-Net)	$4.04_{\pm 0.03}$	$4.93_{\pm 0.04}$	**3.8** (DAT)
CIFAR-10	$\mathbf{0.88}_{\pm 0.02}$	$0.95_{\pm 0.03}$	1 (Gpipe)	$\mathbf{0.30}_{\pm 0.01}$	$0.34_{\pm 0.02}$	0.63 (BiT-L)
CIFAR-100	$\mathbf{7.44}_{\pm 0.06}$	$7.68_{\pm 0.06}$	7.83 (BiT-M)	$\mathbf{3.92}_{\pm 0.06}$	$4.07_{\pm 0.08}$	6.49 (BiT-L)
Birdsnap	$\mathbf{13.64}_{\pm 0.15}$	$14.30_{\pm 0.18}$	15.7 (EffNet)	$\mathbf{9.93}_{\pm 0.15}$	$10.31_{\pm 0.15}$	14.5 (DAT)
Food101	$7.02_{\pm 0.02}$	$7.17_{\pm 0.03}$	7.0 (Gpipe)	$\mathbf{3.82}_{\pm 0.01}$	$3.97_{\pm 0.03}$	4.7 (DAT)
ImageNet	$15.14_{\pm 0.03}$	15.3	**14.2** (KDforAA)	$\mathbf{11.39}_{\pm 0.02}$	11.8	11.45 (ViT)

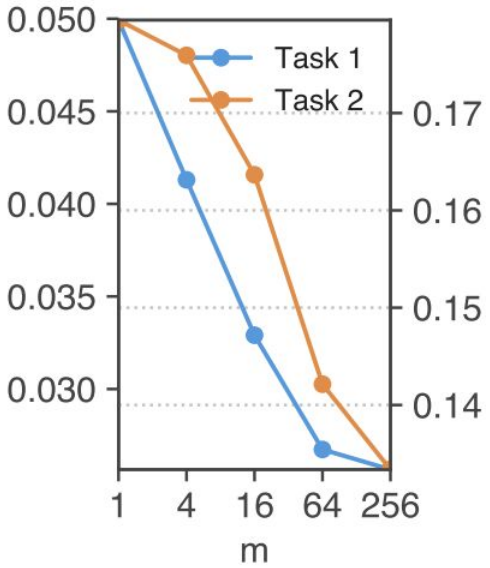
Устойчивость к label noise

- SAM более устойчив к label noise, потому что он может вызывать неверные изменения поверхности loss-а

Method	Noise rate (%)			
	20	40	60	80
Sanchez et al. (2019)	94.0	92.8	90.3	74.1
Zhang & Sabuncu (2018)	89.7	87.6	82.7	67.9
Lee et al. (2019)	87.1	81.8	75.4	-
Chen et al. (2019)	89.7	-	-	52.3
Huang et al. (2019)	92.6	90.3	43.4	-
MentorNet (2017)	92.0	91.2	74.2	60.0
Mixup (2017)	94.0	91.5	86.8	76.9
MentorMix (2019)	**95.6**	**94.2**	91.3	**81.0**
SGD	84.8	68.8	48.2	26.2
Mixup	93.0	90.0	83.8	70.2
Bootstrap + Mixup	93.3	92.0	87.6	72.0
SAM	95.1	93.4	90.5	77.9
Bootstrap + SAM	95.4	**94.2**	**91.8**	79.9

m-sharpness

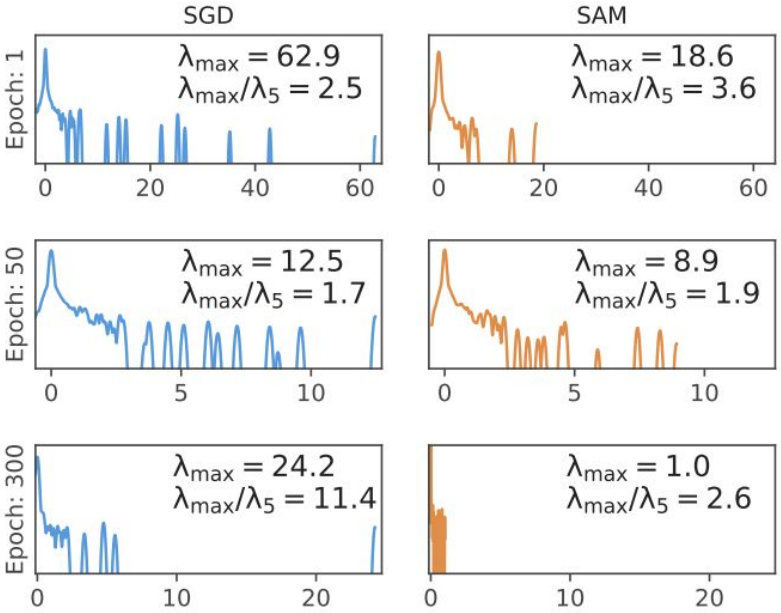
- Выберем несколько случайных подмножеств батча размера m. Для каждого из них вычислим $\hat{\epsilon}(w)$, а затем возьмем среднее SAM update-ов в качестве общего SAM update
- С меньшими значениями m получаются лучше



Эксперимент: маленький ResNet на CIFAR-10

Спектр Гессиана

- Посмотрим на спектр Гессиана в точке оптимума. Ожидаем, собственные числа близки к 1 в случае оптимизации с помощью SAM



Выводы

Плюсы:

- Новый метод регуляризации, применимый к произвольным задачам
- Есть теоретическое обоснование, геометрическая интерпретация
- Обновлены state of the art в задачах компьютерного зрения
- Есть исходный код для применения в популярных фреймворках, замедляет вычисление в 2 раза

Минусы:

- Введение m-sharpness оставляет некоторые вопросы, данная часть статьи выглядит неполной
- Улучшение оценки SAM-loss на практике не работает
- SOTA на finetune бьется бейзлайном

Информация о статье

- ICLR 2021 (можно посмотреть рассказ авторов)
- работа сделана как часть Google AI Residency program
- более 60 цитирований

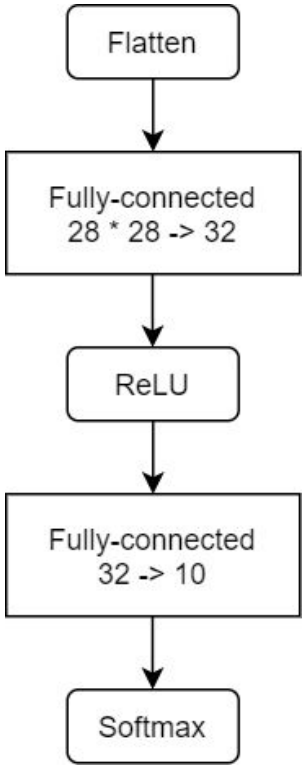
Другие статьи

- Exploring the Vulnerability of Deep Neural Networks: A Study of Parameter Corruption (ссылка) - представлен индикатор устойчивости модели
- Improved Sample Complexities for Deep Networks and Robust Classification via an All-Layer Margin (ссылка) - all-layer margin

В практической части попробовали реализовать:

- Обучение полносвязной сети на MNIST с большим значением label noise
- Обучение свёрточной сети на CIFAR-10 с большим значением label noise
- Сравнение направления спуска на избранных функционалах с двумерным доменом

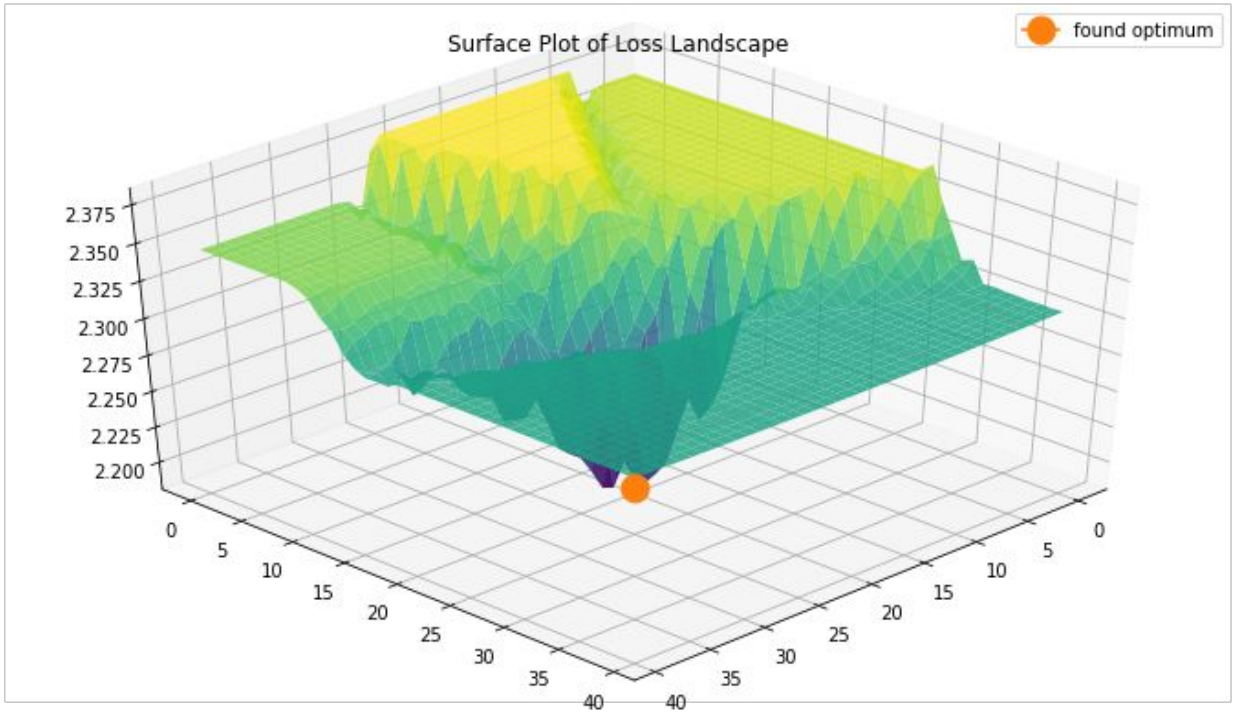
Обучение полносвязной сети на MNIST с большим значением label noise (0.8)



Эффект от SAM негативный

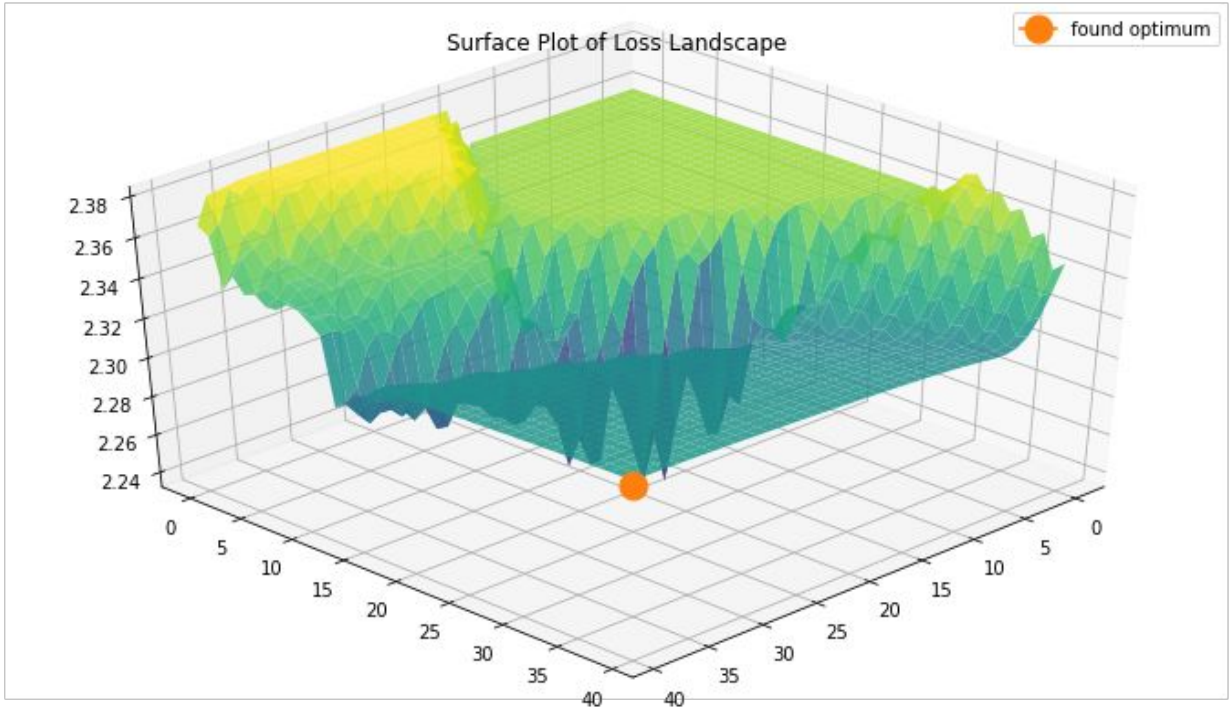
	Adam	Adam with SAM
accuracy	0.88	0.46

Эффект от SAM негативный



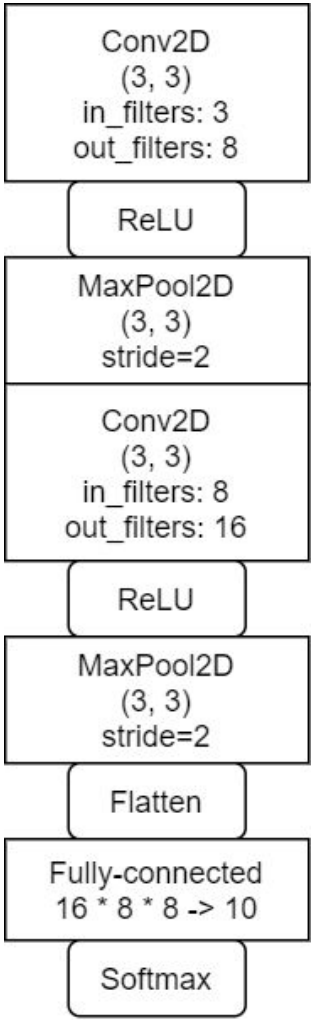
Визуализация ландшафта функции потерь модели без SAM

Эффект от SAM негативный



Визуализация ландшафта функции потерь модели с использованием SAM

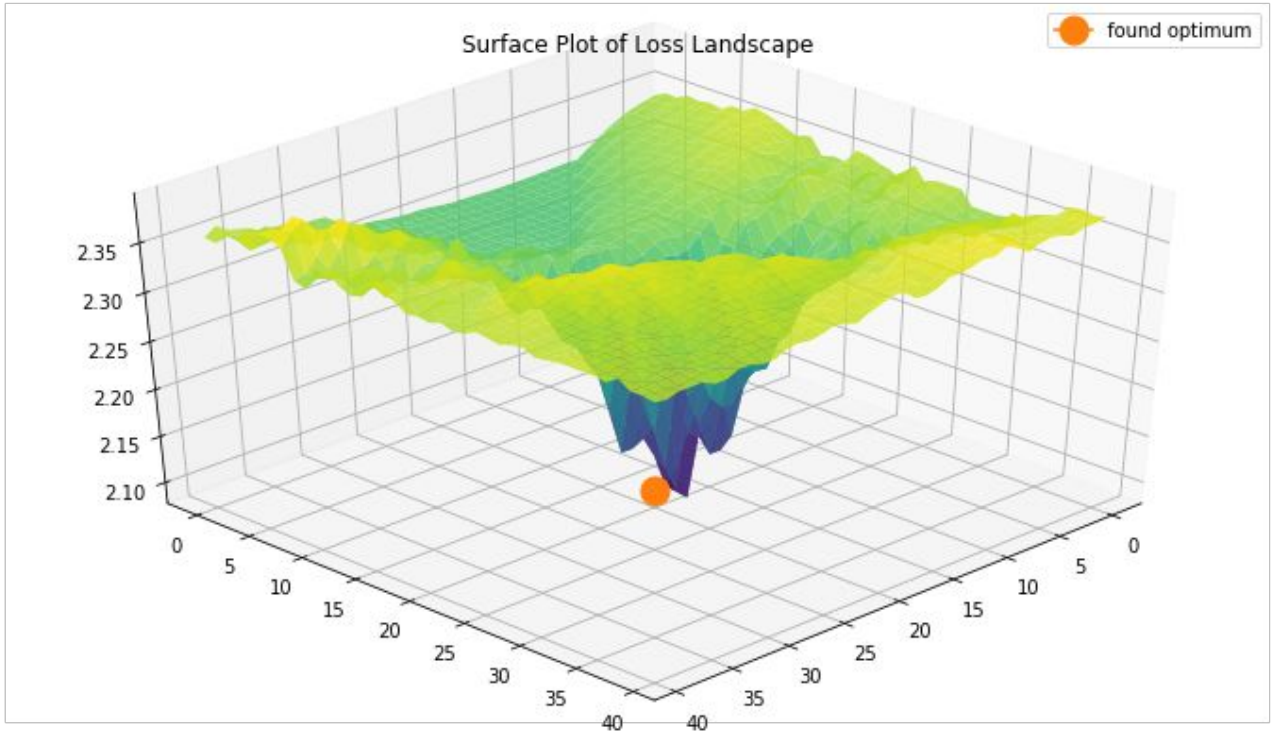
Обучение свёрточной сети на CIFAR-10 с большим значением label noise (0.5)



Эффект от SAM нейтральный

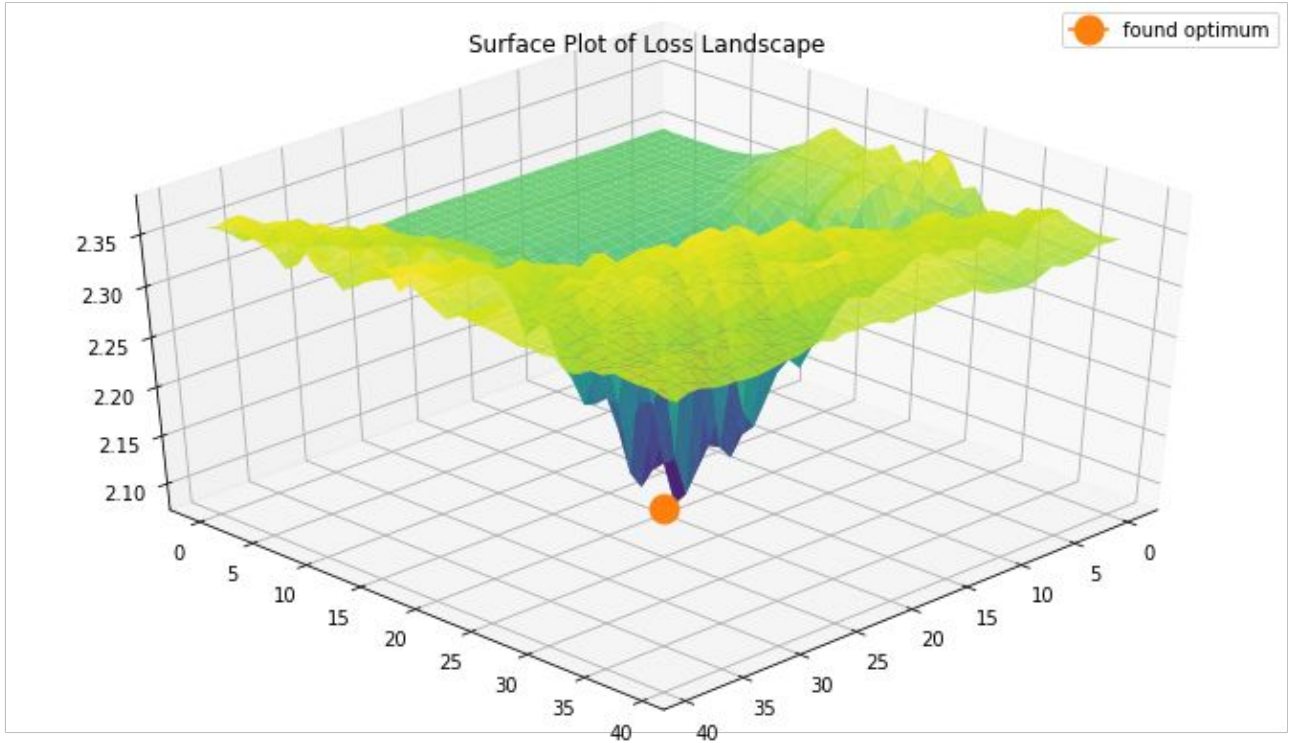
	Adam	Adam with SAM
accuracy	0.47	0.48

Эффект от SAM нейтральный



Визуализация ландшафта функции потерь модели без SAM

Эффект от SAM нейтральный



Визуализация ландшафта функции потерь модели с использованием SAM

Сравнение на функциях $\mathbb{R}^2 \to \mathbb{R}$

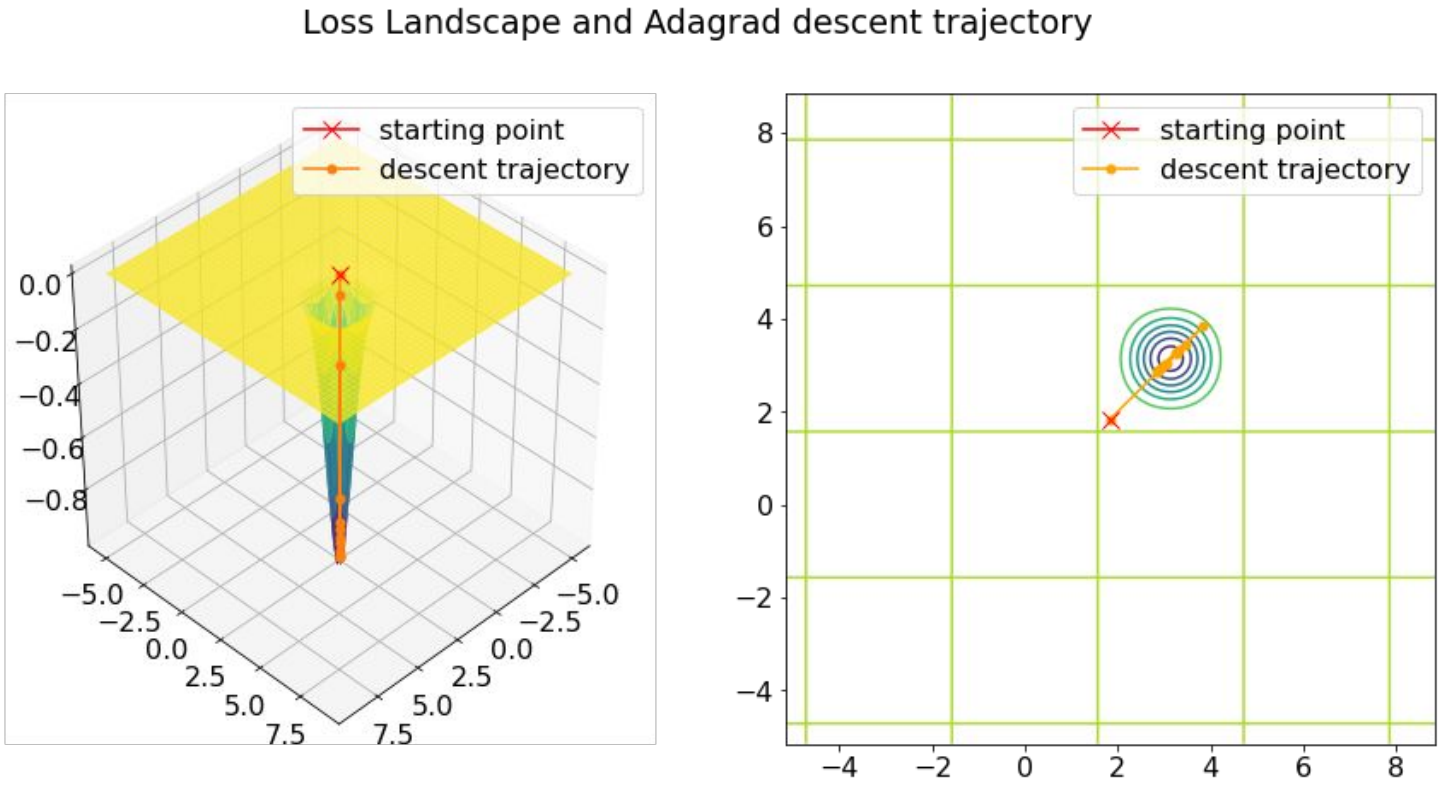
- Easom function

$$f(x, y) = -\cos(x)\cos(y)\exp\left(-\left((x-\pi)^2 + (y-\pi)^2\right)\right)$$

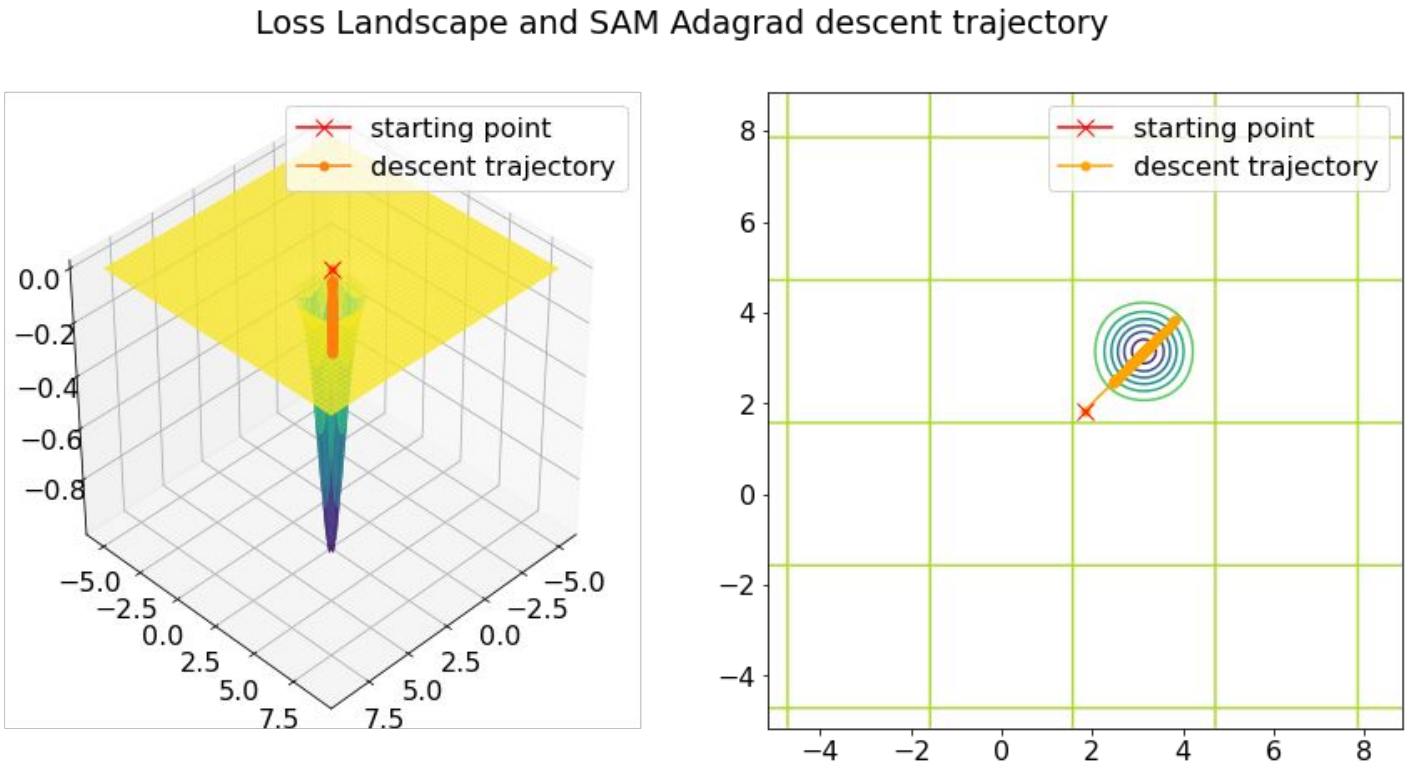
- Eggholder function

$$f(x, y) = -(y+47)\sin\sqrt{\left|\frac{x}{2} + (y+47)\right|} - x\sin\sqrt{|x-(y+47)|}$$

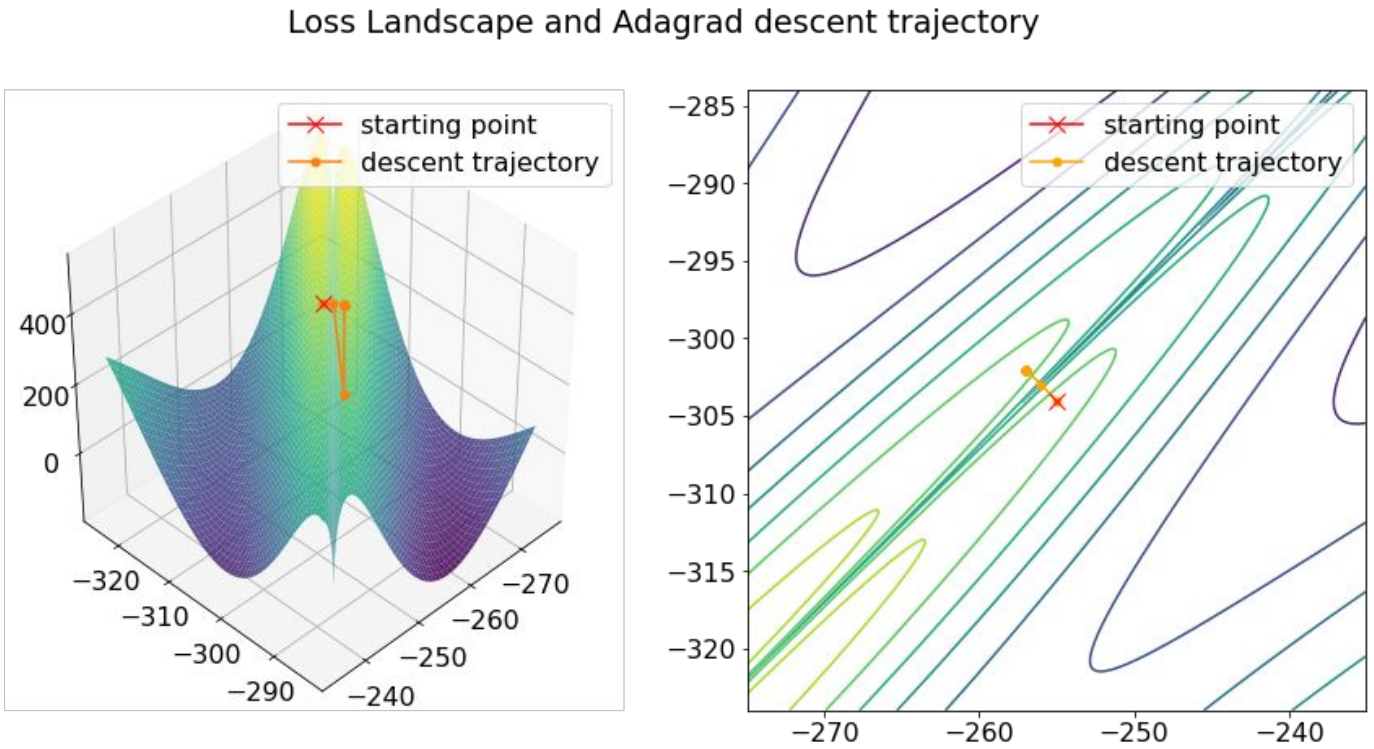
Easom function



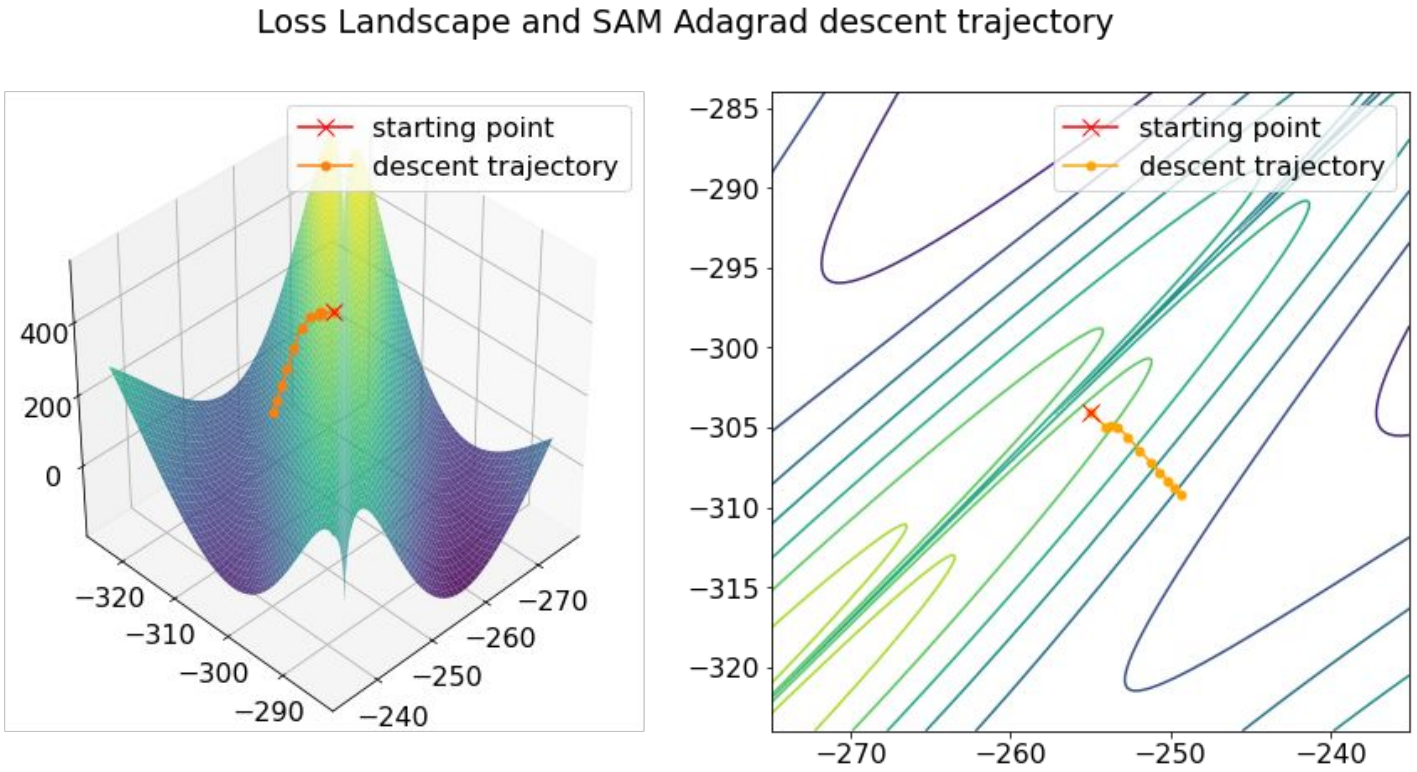
Easom function



Eggholder function



Eggholder function



Выводы практической части

- SAM действительно не позволяет упасть в “острые” минимумы.
- Применимость SAM для простых моделей под вопросом (даже с сильным label noise).

Источники

- https://arxiv.org/abs/2010.01412
- https://github.com/google-research/sam