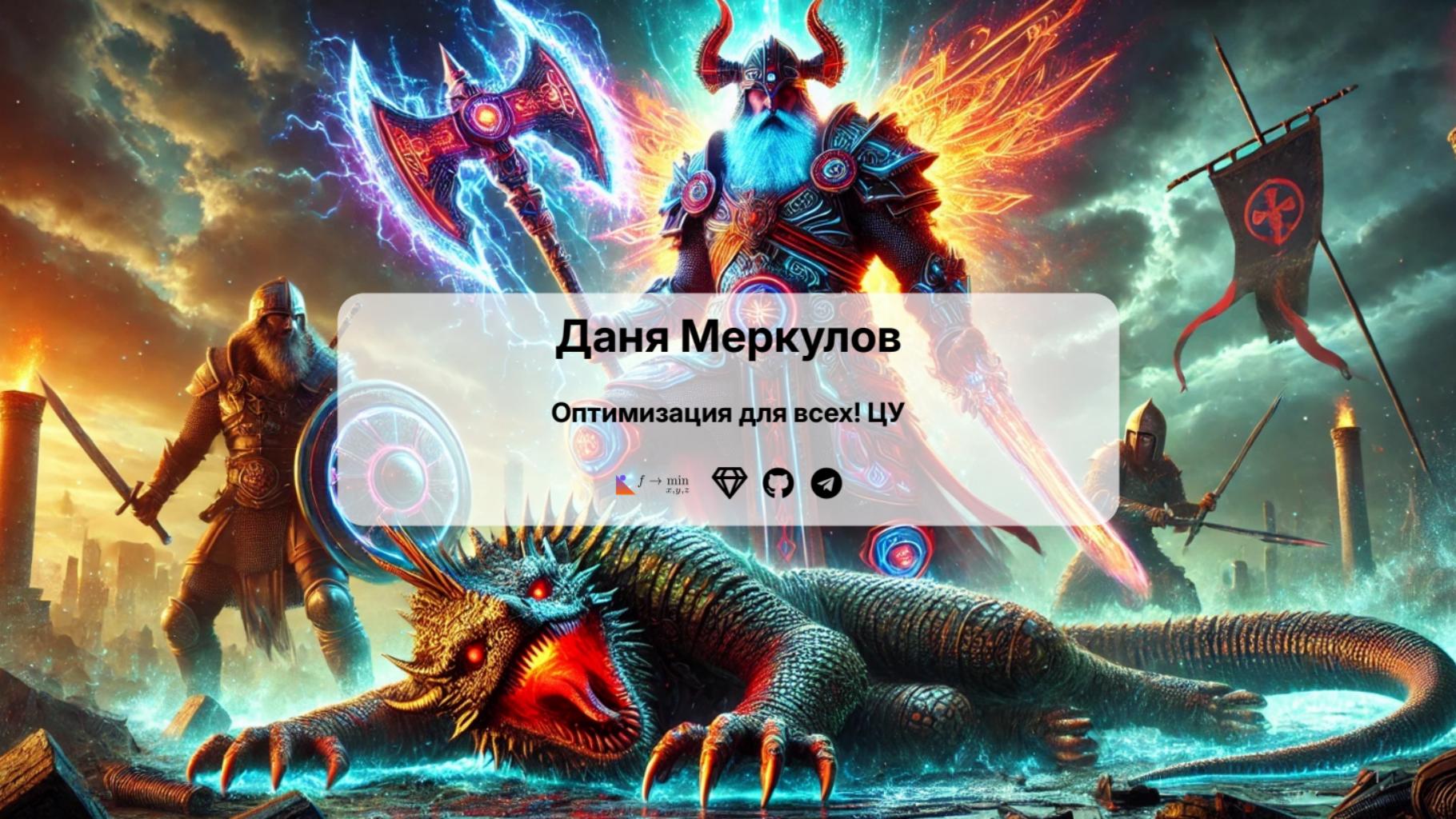


# Сюжеты из обучения нейронных сетей

МЕТОДЫ ВЫПУКЛОЙ ОПТИМИЗАЦИИ

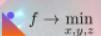
НЕДЕЛЯ 14

Даня Меркулов



# Даня Меркулов

Оптимизация для всех! ЦУ



# **Оптимизация для глубокого обучения с практической точки зрения**

# Как сравнивать методы? Бенчмарк AlgoPerf<sup>1 2</sup>



- **Бенчмарк AlgoPerf:** Сравнивает алгоритмы обучения нейросетей в двух режимах:

# Как сравнивать методы? Бенчмарк AlgoPerf<sup>1 2</sup>



- **Бенчмарк AlgoPerf:** Сравнивает алгоритмы обучения нейросетей в двух режимах:
  - Внешняя настройка (*External Tuning*): моделирует подбор гиперпараметров при ограниченных ресурсах (5 запусков, квазислучайный поиск). Оценка — медианное минимальное время достижения цели по 5 наборам задач.

# Как сравнивать методы? Бенчмарк AlgoPerf<sup>1 2</sup>



- **Бенчмарк AlgoPerf:** Сравнивает алгоритмы обучения нейросетей в двух режимах:
  - Внешняя настройка (*External Tuning*): моделирует подбор гиперпараметров при ограниченных ресурсах (5 запусков, квазислучайный поиск). Оценка — медианное минимальное время достижения цели по 5 наборам задач.
  - Самонастройка (*Self-Tuning*): моделирует автоматический подбор на одной машине (фиксированный или внутренний подбор, бюджет ×3). Оценка — медианное время выполнения по 5 наборам задач.

# Как сравнивать методы? Бенчмарк AlgoPerf<sup>1 2</sup>



- **Бенчмарк AlgoPerf:** Сравнивает алгоритмы обучения нейросетей в двух режимах:
  - **Внешняя настройка (External Tuning):** моделирует подбор гиперпараметров при ограниченных ресурсах (5 запусков, квазислучайный поиск). Оценка — медианное минимальное время достижения цели по 5 наборам задач.
  - **Самонастройка (Self-Tuning):** моделирует автоматический подбор на одной машине (фиксированный или внутренний подбор, бюджет ×3). Оценка — медианное время выполнения по 5 наборам задач.
- **Оценка:** результаты агрегируются с помощью профилей производительности. Профили показывают долю задач, решённых за время, не превышающее множитель  $\tau$  относительно самой быстрой посылки. Итоговая оценка — нормированная площадь под кривой профиля (1.0 = самая быстрая на всех задачах).

# Как сравнивать методы? Бенчмарк AlgoPerf<sup>1</sup><sup>2</sup>



- **Бенчмарк AlgoPerf:** Сравнивает алгоритмы обучения нейросетей в двух режимах:
  - **Внешняя настройка (External Tuning):** моделирует подбор гиперпараметров при ограниченных ресурсах (5 запусков, квазислучайный поиск). Оценка — медианное минимальное время достижения цели по 5 наборам задач.
  - **Самонастройка (Self-Tuning):** моделирует автоматический подбор на одной машине (фиксированный или внутренний подбор, бюджет ×3). Оценка — медианное время выполнения по 5 наборам задач.
- **Оценка:** результаты агрегируются с помощью профилей производительности. Профили показывают долю задач, решённых за время, не превышающее множитель  $\tau$  относительно самой быстрой посылки. Итоговая оценка — нормированная площадь под кривой профиля (1.0 = самая быстрая на всех задачах).
- **Затраты ресурсов:** оценка требует  $\sim 49,240$  часов суммарно на 8x NVIDIA V100 GPUs (в среднем  $\sim 3469$  ч/внешняя настройка,  $\sim 1847$  ч/самонастройка).

---

<sup>1</sup>Benchmarking Neural Network Training Algorithms

<sup>2</sup>Accelerating neural network training: An analysis of the AlgoPerf competition

# Бенчмарк AlgoPerf

**Сводка фиксированных базовых задач в бенчмарке AlgoPerf.** Функции потерь включают кросс-энтропию (CE), среднюю абсолютную ошибку (L1) и функцию потерь CTC (Connectionist Temporal Classification). Дополнительные метрики оценки: индекс структурного сходства (SSIM), коэффициент ошибок (ER), доля ошибок по словам (WER), средняя усреднённая точность (mAP) и метрика BLEU (*bilingual evaluation understudy*). Бюджет времени выполнения соответствует правилам внешней настройки; правила самонастройки допускают обучение, в 3 раза более длительное.

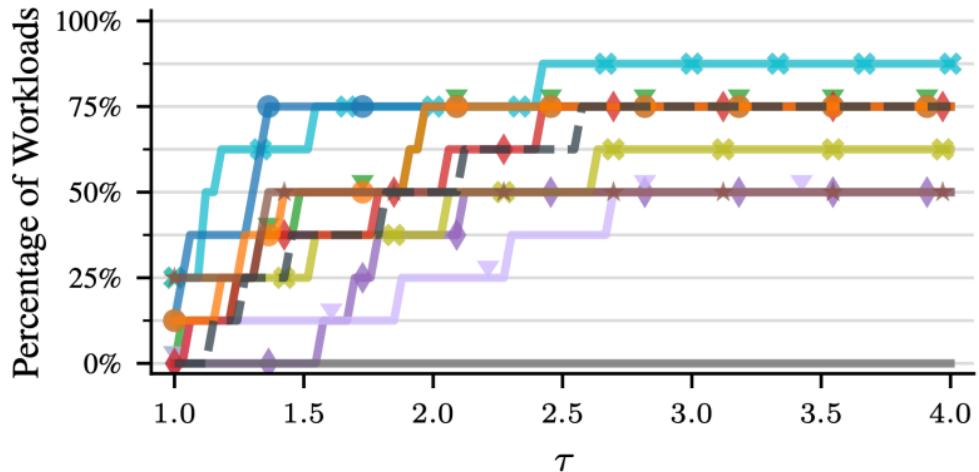
Задача	Датасет	Модель	Функция потерь		Целевое значение (валидация)	Бюджет времени
				Метрика		
Clickthrough rate prediction	CRITEO 1TB	DLRMSMALL	CE	CE	0.123735	7703
MRI reconstruction	FASTMRI	U-NET	L1	SSIM	0.7344	8859
Image classification	IMAGENET	ResNet-50	CE	ER	0.22569	63,008
		ViT	CE	ER	0.22691	77,520
Speech recognition	LIBRISPEECH	Conformer	CTC	WER	0.085884	61,068
		DeepSpeech	CTC	WER	0.119936	55,506
Molecular property prediction	OGBG	GNN	CE	mAP	0.28098	18,477
Translation	WMT	Transformer	CE	BLEU	30.8491	48,151

# Бенчмарк AlgoPerf



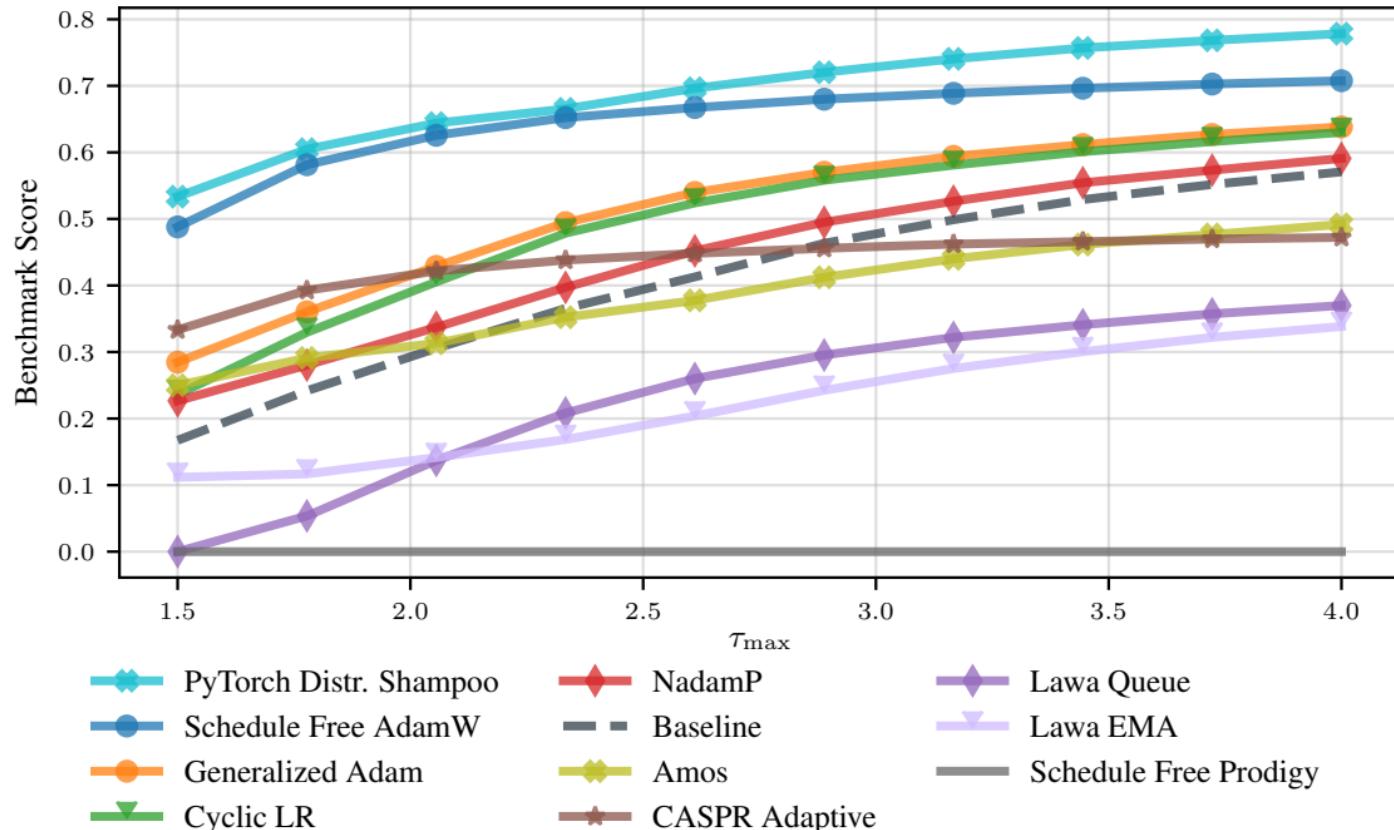
Submission	Line	Score
PYTORCH DISTRIBUTED SHAMPOO		0.7784
SCHEDULE FREE ADAMW		0.7077
GENERALIZED ADAM		0.6383
CYCLIC LR		0.6301
NADAMP		0.5909
<b>BASELINE</b>		<b>0.5707</b>
AMOS		0.4918
CASPR ADAPTIVE		0.4722
LAWA QUEUE		0.3699
LAWA EMA		0.3384
SCHEDULE FREE PRODIGY		0

(a) External tuning leaderboard

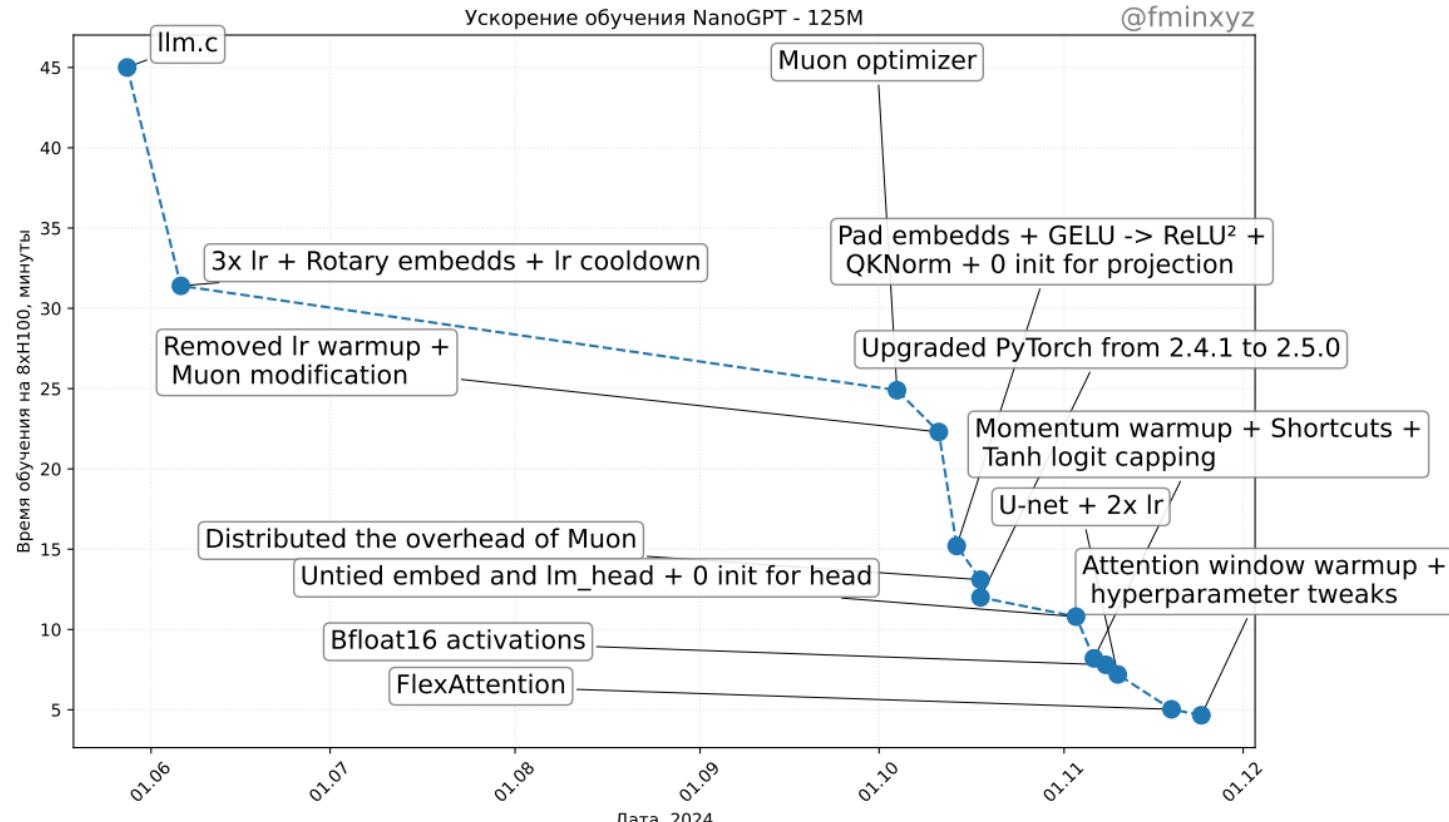


(b) External tuning performance profiles

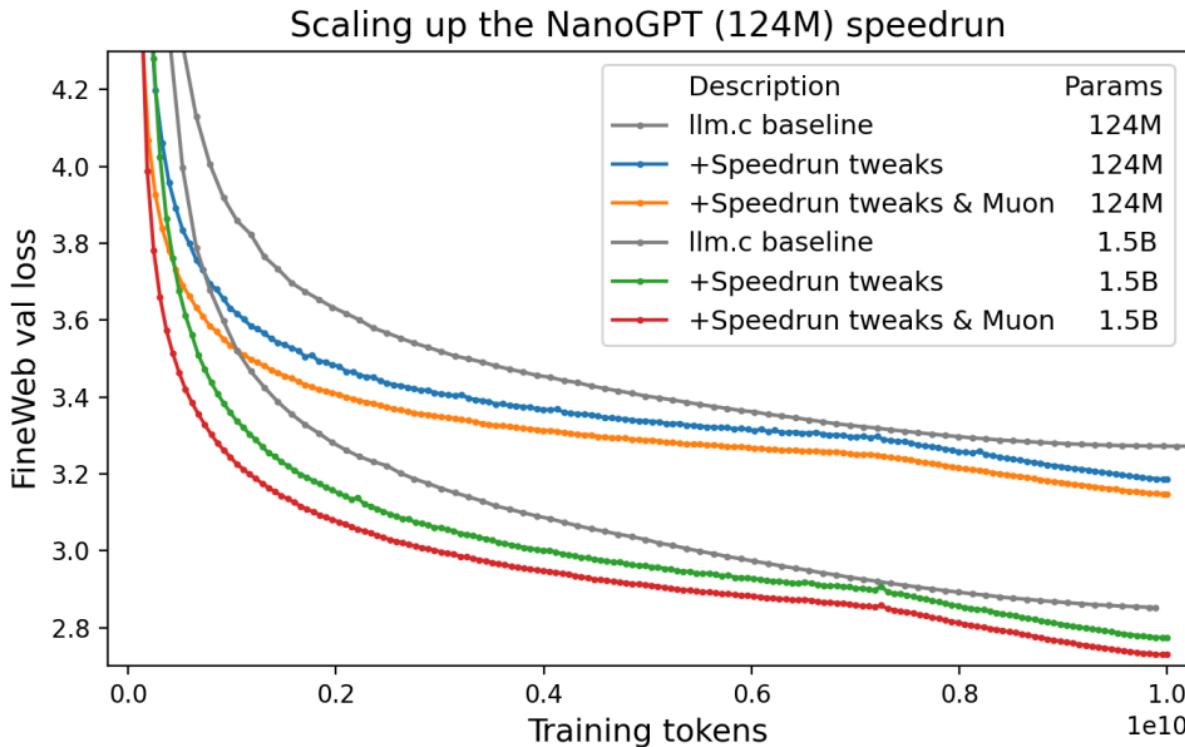
# Бенчмарк AlgoPerf



# NanoGPT speedrun



# Работают ли трюки, если увеличить размер модели?



# Работают ли трюки, если увеличить размер модели?

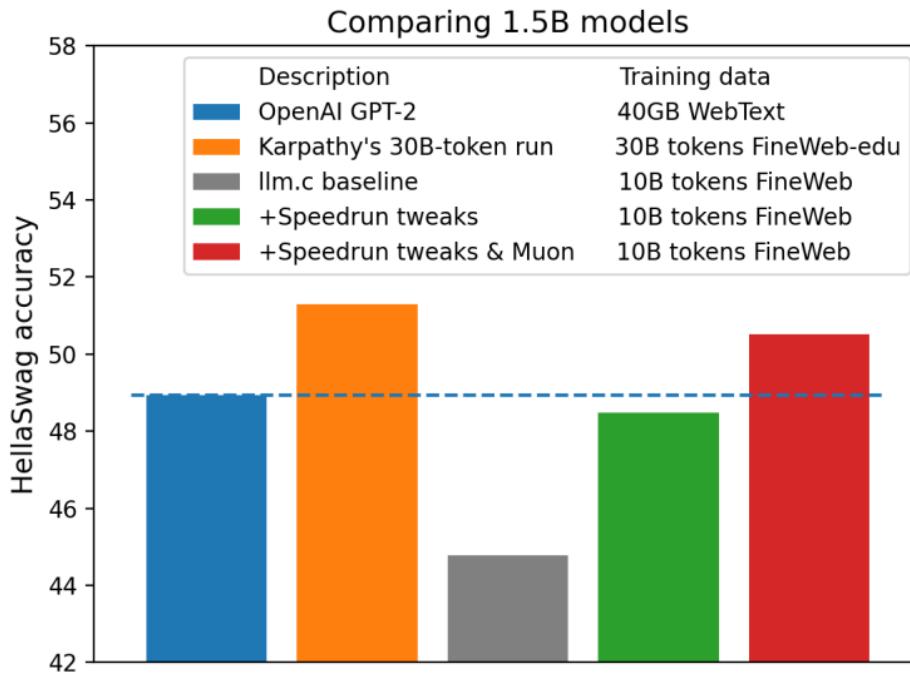


Рисунок 3. Источник



# Неожиданные истории

# Adam работает хуже для CV, чем для LLM? <sup>3</sup>

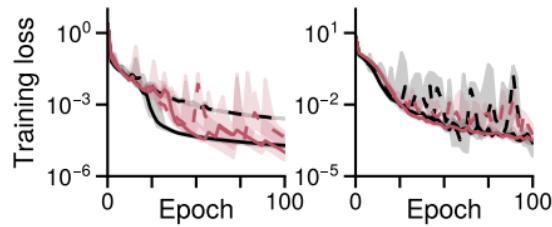


Рисунок 4. CNNs on MNIST and CIFAR10

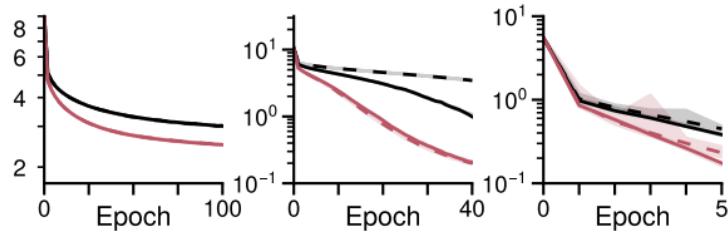


Рисунок 5. Transformers on PTB, WikiText2, and SQuAD

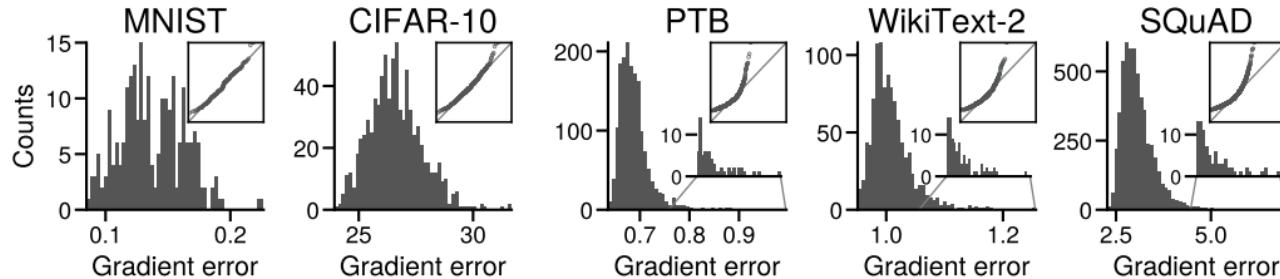
Чёрные линии — SGD, красные — Adam.

<sup>3</sup>Linear attention is (maybe) all you need (to understand transformer optimization)

# Почему Adam работает хуже для CV, чем для LLM? <sup>4</sup>



Потому что шум градиентов в языковых моделях имеет тяжелые хвосты?



<sup>4</sup>Linear attention is (maybe) all you need (to understand transformer optimization)

# Почему Adam работает хуже для CV, чем для LLM?<sup>5</sup>



Нет! Распределение меток имеет тяжёлые хвосты!

В компьютерном зрении датасеты часто сбалансированы: 1000 котиков, 1000 песелей и т.д.

В языковых датасетах почти всегда не так: слово *the* встречается часто, слово *tie* — на порядки реже.

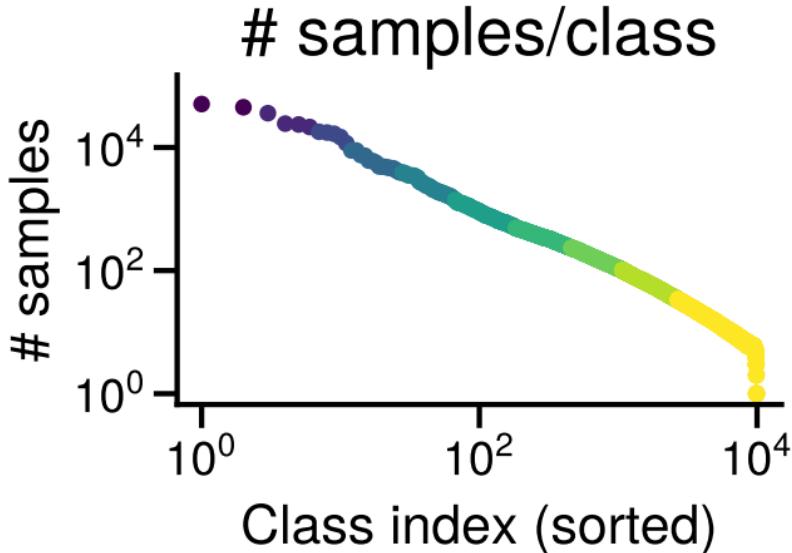


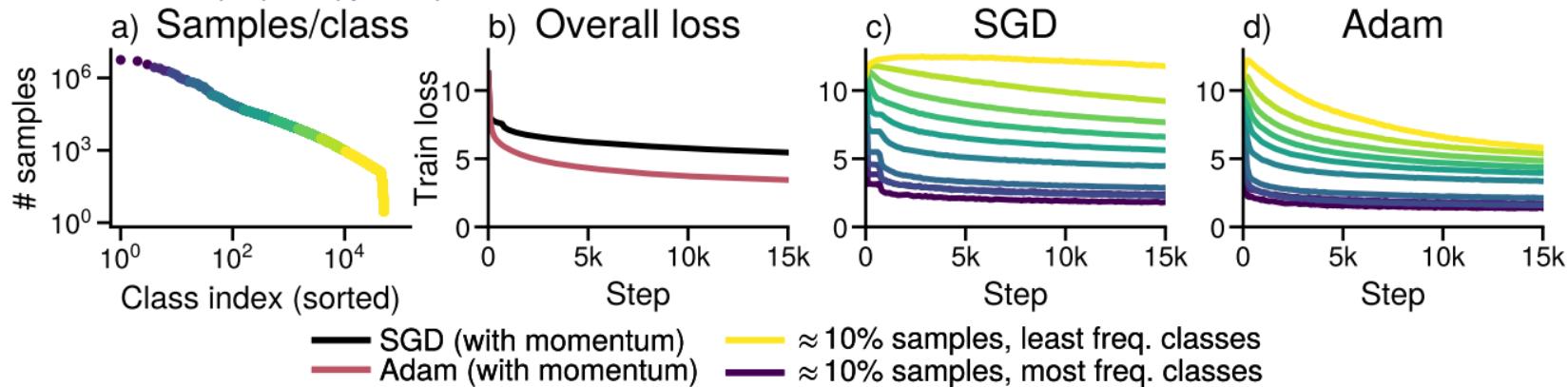
Рисунок 6. Распределение частоты токенов в PTB

<sup>5</sup>Heavy-Tailed Class Imbalance and Why Adam Outperforms Gradient Descent on Language Models

# Почему Adam работает хуже для CV, чем для LLM? <sup>6</sup>



SGD медленно прогрессирует на редких классах



SGD не добивается прогресса на низкочастотных классах, в то время как Adam добивается. Обучение GPT-2 S на WikiText-103. (a) Распределение классов, отсортированных по частоте встречаемости, разбитых на группы, соответствующие  $\approx 10\%$  данных. (b) Значение функции потерь при обучении. (c, d) Значение функции потерь при обучении для каждой группы при использовании SGD и Adam.

<sup>6</sup>Heavy-Tailed Class Imbalance and Why Adam Outperforms Gradient Descent on Language Models

# Влияние инициализации<sup>7</sup>



- 💡 Правильная инициализация нейронной сети важна. Функция потерь нейронной сети сильно невыпукла; оптимизировать её для достижения «хорошего» решения трудно, это требует тщательной настройки.

# Влияние инициализации<sup>7</sup>



 Правильная инициализация нейронной сети важна. Функция потерь нейронной сети сильно невыпукла; оптимизировать её для достижения «хорошего» решения трудно, это требует тщательной настройки.

- Не инициализируйте все веса одинаково — почему?

# Влияние инициализации



Правильная инициализация нейронной сети важна. Функция потерь нейронной сети сильно невыпукла; оптимизировать её для достижения «хорошего» решения трудно, это требует тщательной настройки.

- Не инициализируйте все веса одинаково — почему?
- Случайная инициализация: инициализируйте случайно, например, из гауссовского распределения  $N(0, \sigma^2)$ , где стандартное отклонение  $\sigma$  зависит от числа нейронов в слое. Это обеспечивает нарушение симметрии (*symmetry breaking*).

# Влияние инициализации<sup>7</sup>

💡 Правильная инициализация нейронной сети важна. Функция потерь нейронной сети сильно невыпукла; оптимизировать её для достижения «хорошего» решения трудно, это требует тщательной настройки.

- Не инициализируйте все веса одинаково — почему?
- Случайная инициализация: инициализируйте случайно, например, из гауссовского распределения  $N(0, \sigma^2)$ , где стандартное отклонение  $\sigma$  зависит от числа нейронов в слое. Это обеспечивает нарушение симметрии (*symmetry breaking*).
- Можно найти более полезные советы здесь

<sup>7</sup>On the importance of initialization and momentum in deep learning Ilya Sutskever, James Martens, George Dahl, Geoffrey Hinton

# Влияние инициализации весов нейронной сети на сходимость методов<sup>8</sup>

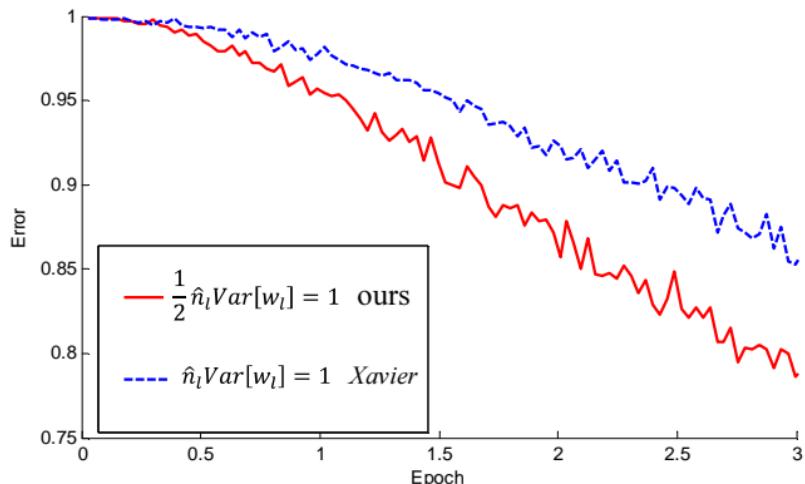


Рисунок 7. 22-layer ReLU net: good init converges faster

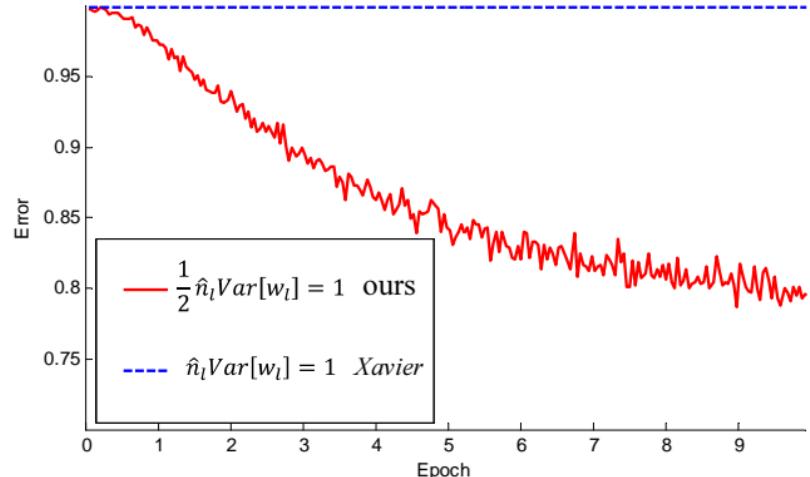


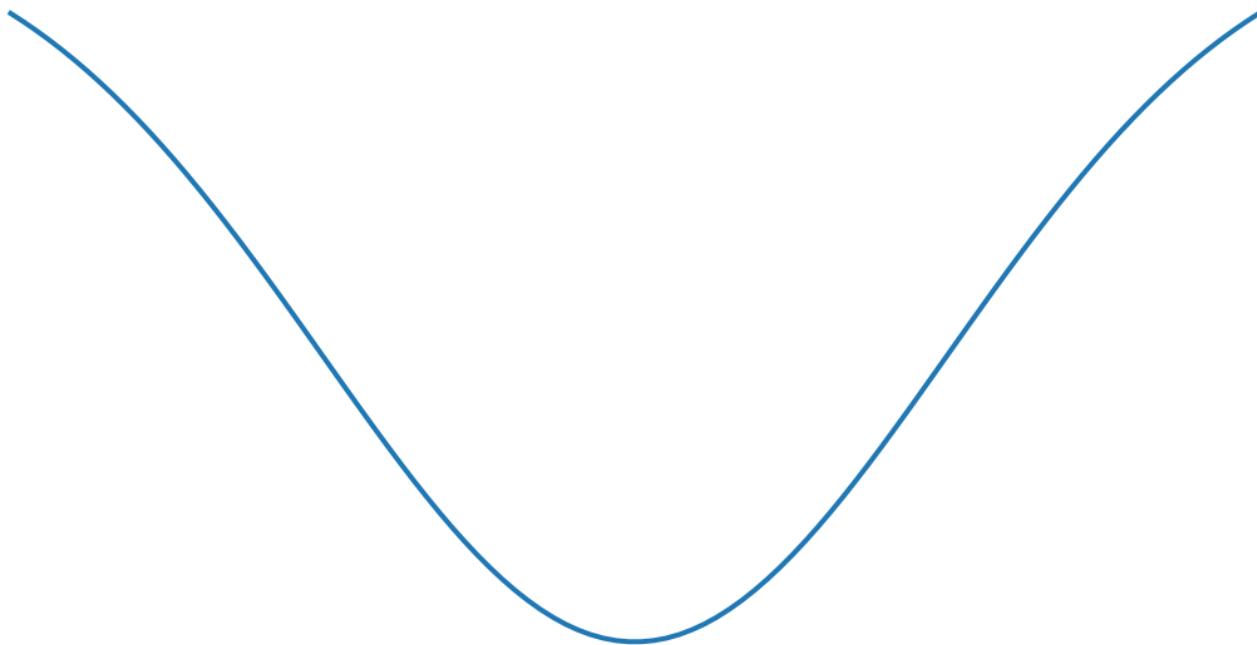
Рисунок 8. 30-layer ReLU net: good init is able to converge

<sup>8</sup>Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification, Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun

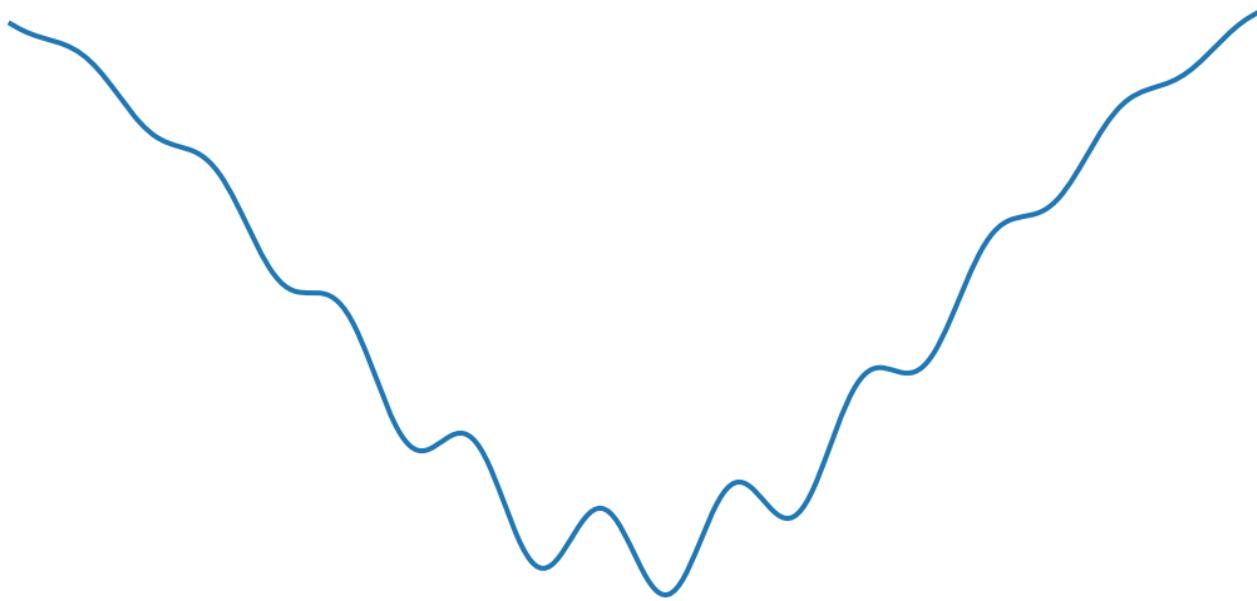


# Весёлые истории

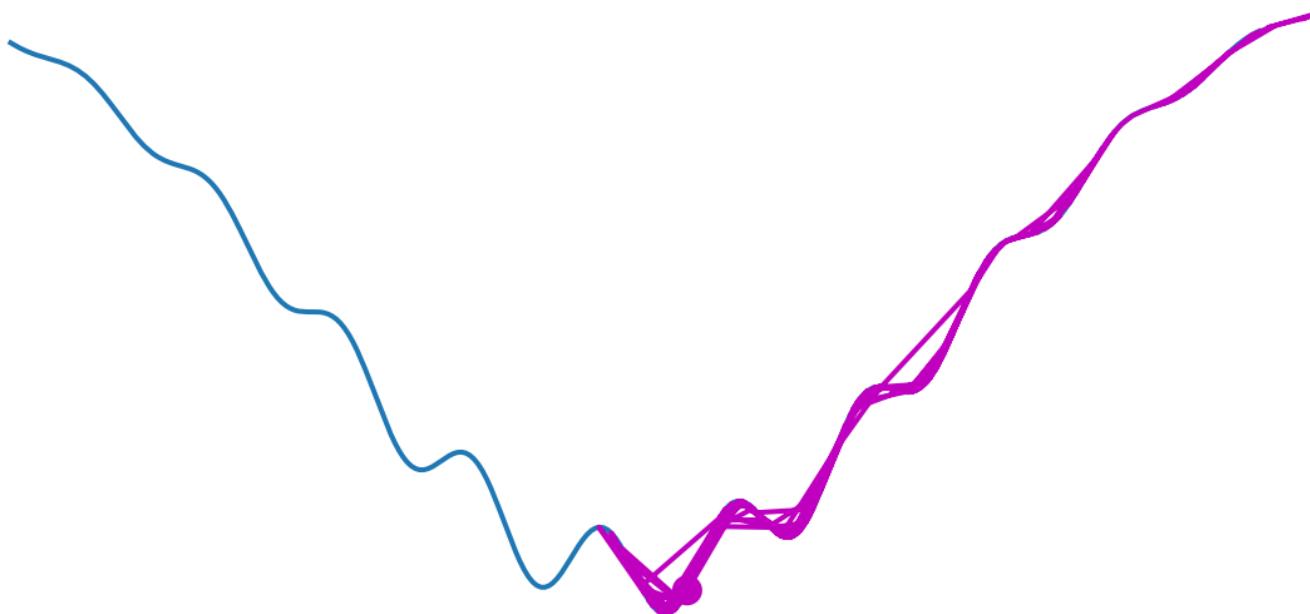
Градиентный спуск сходится к локальному минимуму



Градиентный спуск  
сходится к локальному минимуму



Стохастический градиентный спуск  
выпрыгивает из локальных минимумов



# Визуализация с помощью проекции на прямую

- Обозначим через  $w_0$  начальные веса нейронной сети. Веса, полученные после обучения, обозначим  $\hat{w}$ .

# Визуализация с помощью проекции на прямую

- Обозначим через  $w_0$  начальные веса нейронной сети. Веса, полученные после обучения, обозначим  $\hat{w}$ .

# Визуализация с помощью проекции на прямую

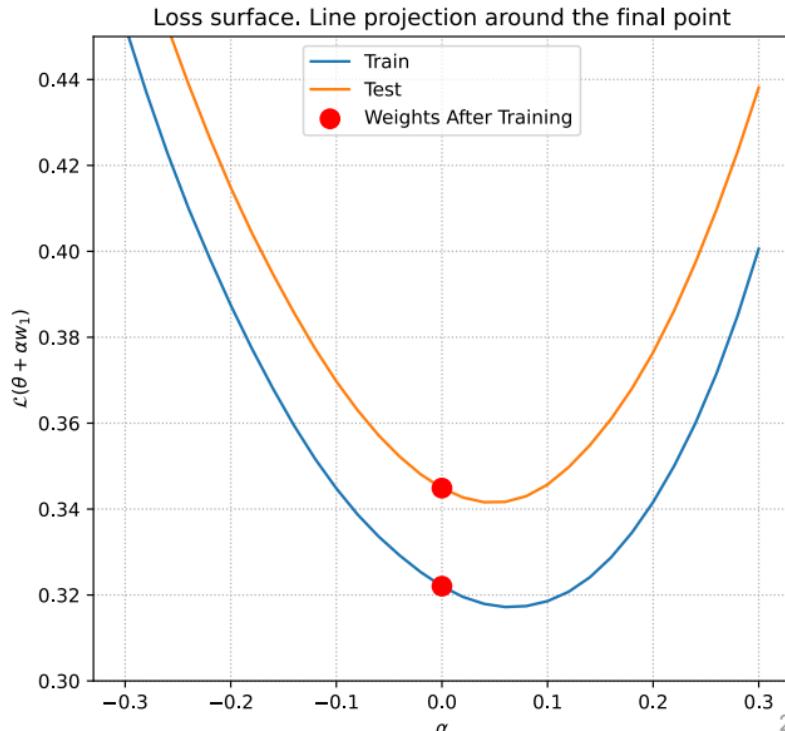
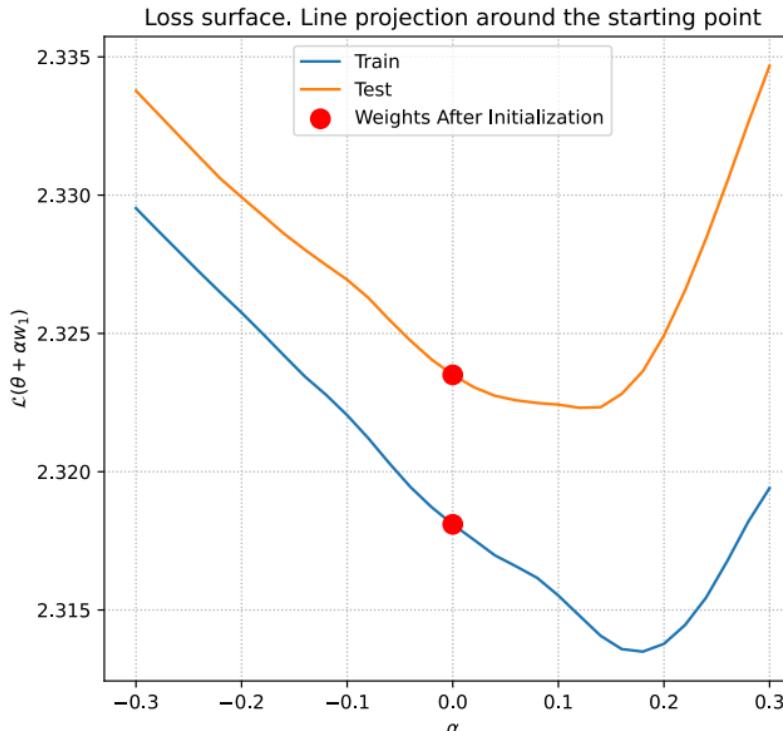
- Обозначим через  $w_0$  начальные веса нейронной сети. Веса, полученные после обучения, обозначим  $\hat{w}$ .
- Сгенерируем случайное направление  $w_1 \in \mathbb{R}^p$  той же размерности, затем вычислим значение функции потерь вдоль этого направления:

$$L(\alpha) = L(w_0 + \alpha w_1), \quad \text{где } \alpha \in [-b, b].$$

# Проекция функции потерь нейронной сети на прямую



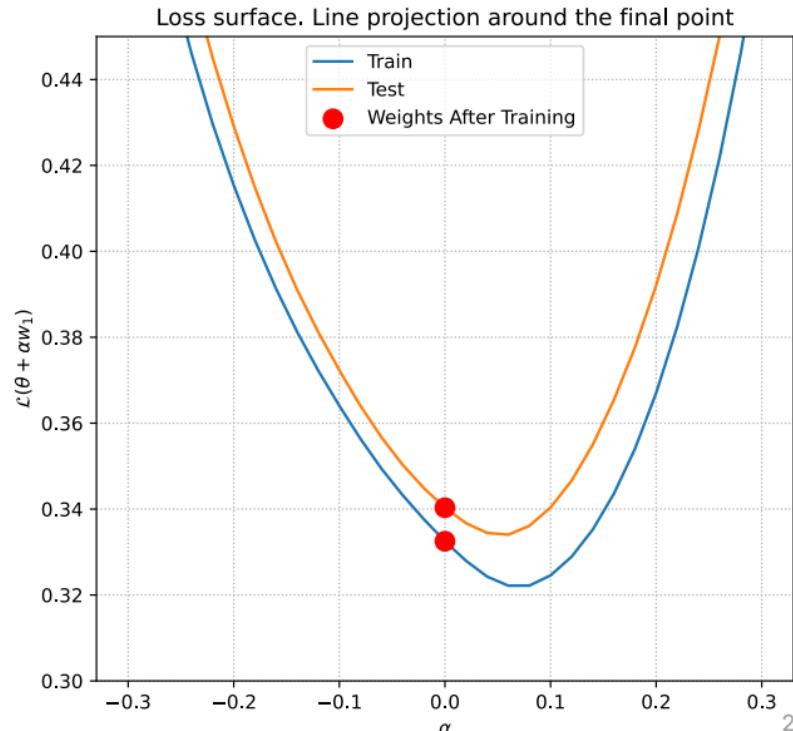
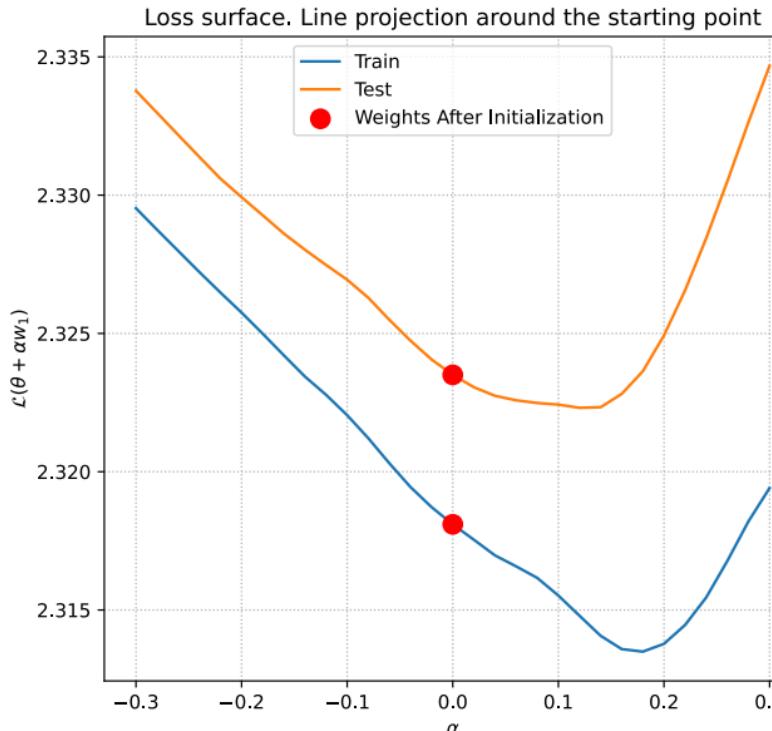
No Dropout



# Проекция функции потерь нейронной сети на прямую



Dropout 0.2



# Проекция функции потерь нейронной сети на плоскость

- Мы можем расширить эту идею и построить проекцию поверхности потерь на плоскость, которая задается 2 случайными векторами.

# Проекция функции потерь нейронной сети на плоскость

- Мы можем расширить эту идею и построить проекцию поверхности потерь на плоскость, которая задается 2 случайными векторами.

# Проекция функции потерь нейронной сети на плоскость

- Мы можем расширить эту идею и построить проекцию поверхности потерь на плоскость, которая задается 2 случайными векторами.
- Два случайных гауссовых вектора в пространстве большой размерности с высокой вероятностью ортогональны.

$$L(\alpha, \beta) = L(w_0 + \alpha w_1 + \beta w_2), \quad \text{где } \alpha, \beta \in [-b, b]^2.$$

# Проекция функции потерь нейронной сети на плоскость

- Мы можем расширить эту идею и построить проекцию поверхности потерь на плоскость, которая задается 2 случайными векторами.
- Два случайных гауссовых вектора в пространстве большой размерности с высокой вероятностью ортогональны.

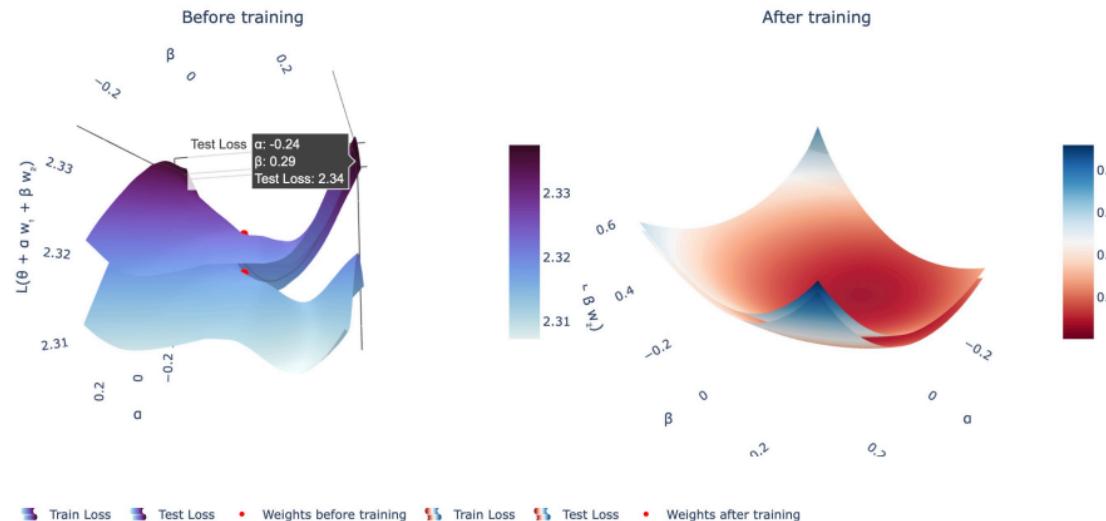
$$L(\alpha, \beta) = L(w_0 + \alpha w_1 + \beta w_2), \quad \text{где } \alpha, \beta \in [-b, b]^2.$$

# Проекция функции потерь нейронной сети на плоскость

- Мы можем расширить эту идею и построить проекцию поверхности потерь на плоскость, которая задается 2 случайными векторами.
- Два случайных гауссовых вектора в пространстве большой размерности с высокой вероятностью ортогональны.

$$L(\alpha, \beta) = L(w_0 + \alpha w_1 + \beta w_2), \quad \text{где } \alpha, \beta \in [-b, b]^2.$$

No Dropout. Plane projection of loss surface.



# Может ли быть полезно изучение таких проекций?



<sup>9</sup>

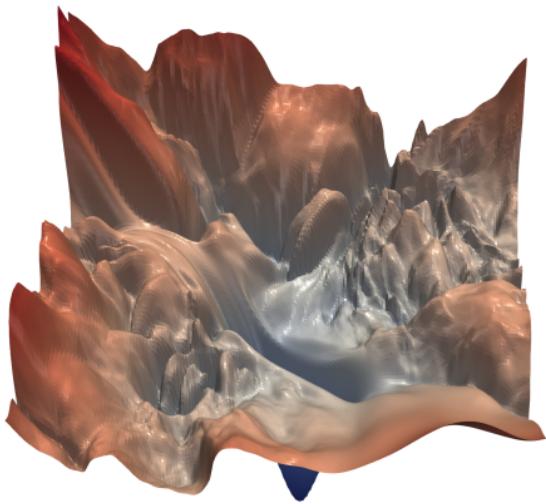


Рисунок 12. The loss surface of ResNet-56  
without skip connections

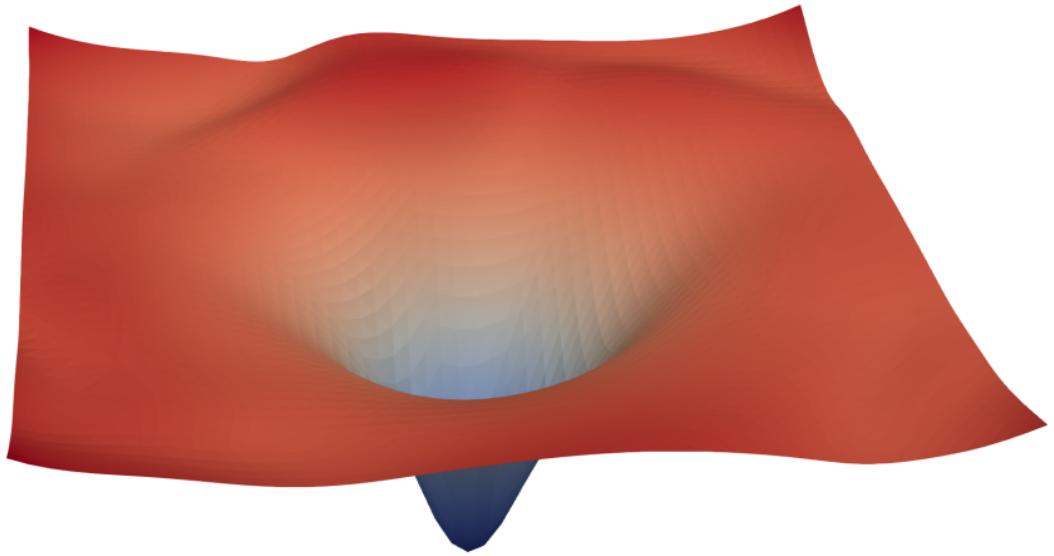


Рисунок 13. The loss surface of ResNet-56 with skip connections

<sup>9</sup>Visualizing the Loss Landscape of Neural Nets, Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, Tom Goldstein

# Может ли быть полезно изучение таких проекций, если серьезно? <sup>10</sup>

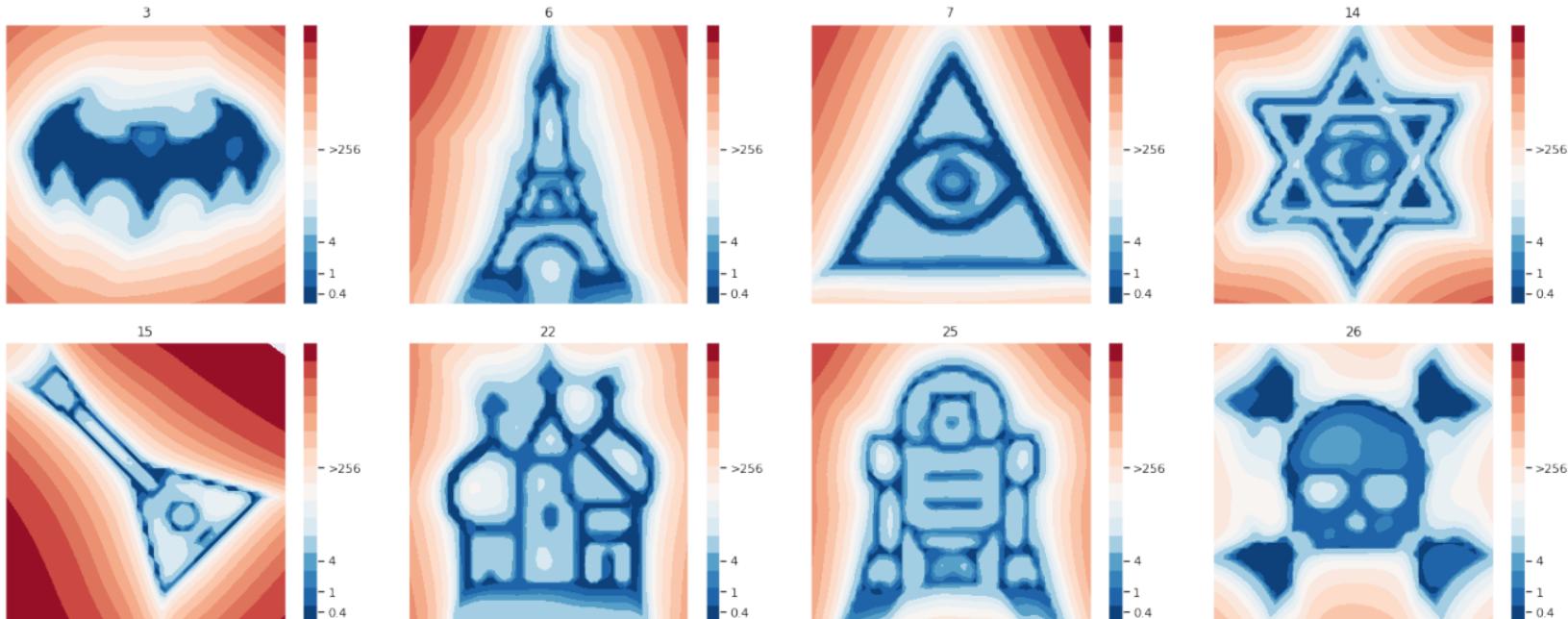
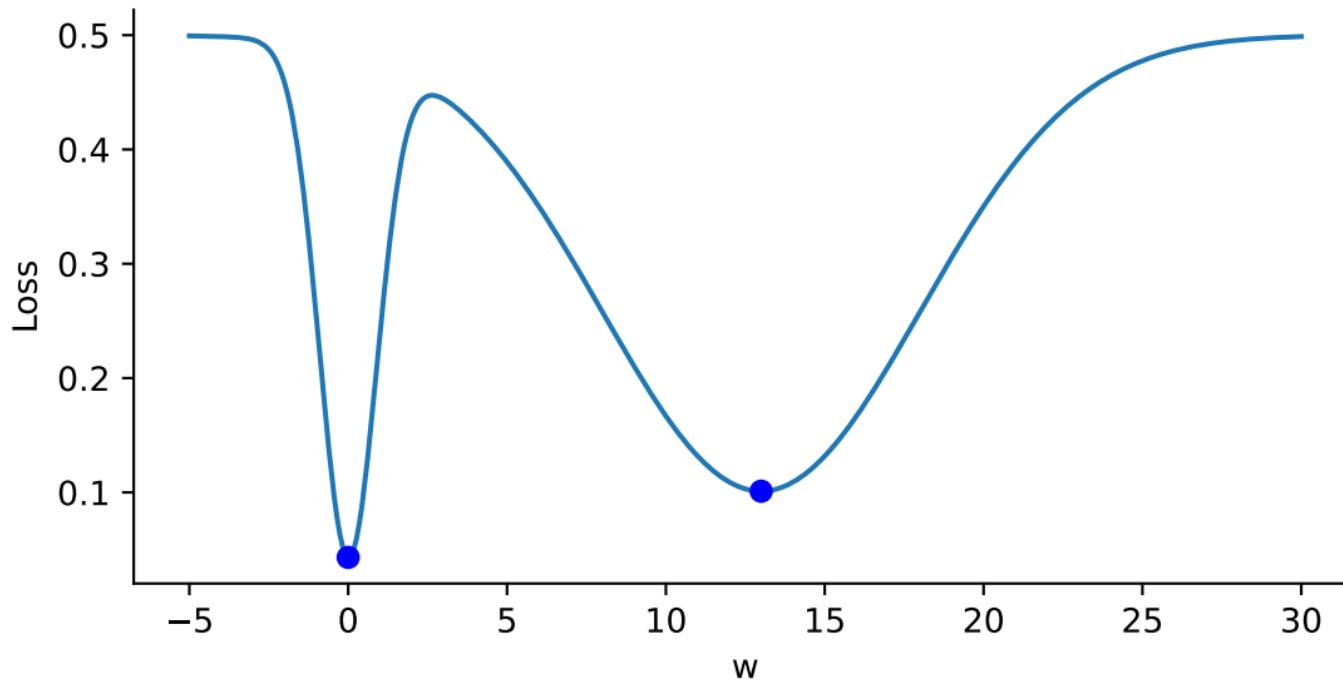


Рисунок 14. Examples of a loss landscape of a typical CNN model on FashionMNIST and CIFAR10 datasets found with MPO. Loss values are color-coded according to a logarithmic scale

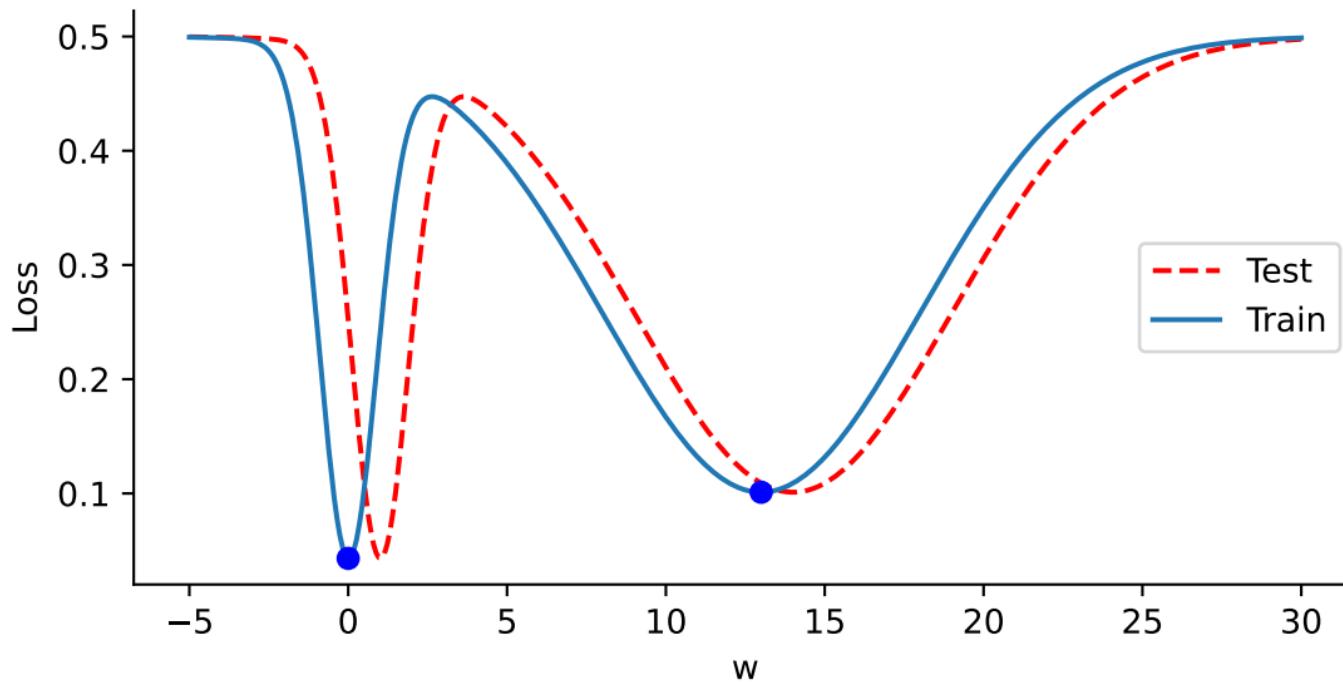
# Ширина локальных минимумов

Узкие и широкие локальные минимумы



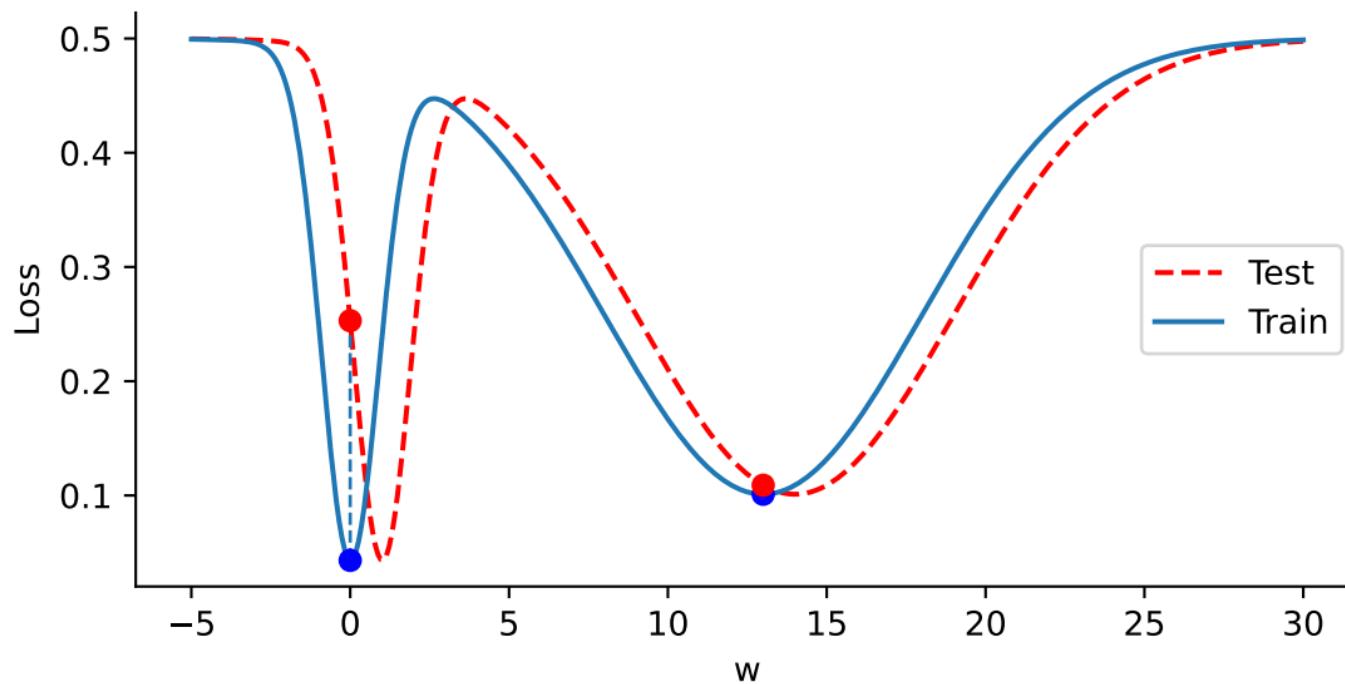
# Ширина локальных минимумов

Узкие и широкие локальные минимумы



# Ширина локальных минимумов

Узкие и широкие локальные минимумы



# Немного про LR schedulers: Экспоненциально растущий LR (ExpLR) (??!??!) <sup>11</sup>

- Вопрос авторов: действительно ли уменьшение LR является необходимым условием успешного обучения глубоких сетей?

# Немного про LR schedulers: Экспоненциально растущий LR (ExpLR) (??!??!)<sup>11</sup>

- Вопрос авторов: действительно ли уменьшение LR является необходимым условием успешного обучения глубоких сетей?
- Авторы предлагают **экспоненциально растущую** стратегию LR:

$$\eta_t = \eta_0(1 + \alpha)^t, \quad \alpha > 0$$

# Немного про LR schedulers: Экспоненциально растущий LR (ExpLR) (??!??!)<sup>11</sup>

- Вопрос авторов: действительно ли уменьшение LR является необходимым условием успешного обучения глубоких сетей?
- Авторы предлагают **экспоненциально растущую** стратегию LR:

$$\eta_t = \eta_0(1 + \alpha)^t, \quad \alpha > 0$$

- Несмотря на быстрое «взрывание» шага, обучение **всё ещё возможно**.

# Немного про LR schedulers: Экспоненциально растущий LR (ExpLR) (??!??!)<sup>11</sup>

- Вопрос авторов: действительно ли уменьшение LR является необходимым условием успешного обучения глубоких сетей?
- Авторы предлагают **экспоненциально растущую** стратегию LR:

$$\eta_t = \eta_0(1 + \alpha)^t, \quad \alpha > 0$$

- Несмотря на быстрое «взрывание» шага, обучение **всё ещё возможно**.
- Экспериментальный факт:

# Немного про LR schedulers: Экспоненциально растущий LR (ExpLR) (??!??!) <sup>11</sup>

- Вопрос авторов: действительно ли уменьшение LR является необходимым условием успешного обучения глубоких сетей?
- Авторы предлагают **экспоненциально растущую** стратегию LR:

$$\eta_t = \eta_0(1 + \alpha)^t, \quad \alpha > 0$$

- Несмотря на быстрое «взрывание» шага, обучение **всё ещё возможно**.
- Экспериментальный факт:
  - стандартные архитектуры для CIFAR-10 (например, PreResNet-32) успешно обучаются с ExpLR;

# Немного про LR schedulers: Экспоненциально растущий LR (ExpLR) (??!??!) <sup>11</sup>

- Вопрос авторов: действительно ли уменьшение LR является необходимым условием успешного обучения глубоких сетей?
- Авторы предлагают **экспоненциально растущую** стратегию LR:

$$\eta_t = \eta_0(1 + \alpha)^t, \quad \alpha > 0$$

- Несмотря на быстрое «взрывание» шага, обучение **всё ещё возможно**.
- Экспериментальный факт:
  - стандартные архитектуры для CIFAR-10 (например, PreResNet-32) успешно обучаются с ExpLR;
  - при корректном выборе  $\alpha$  траектория оптимизации оказывается близка к классической стратегии: фиксированный LR + weight decay.

# Немного про LR schedulers: Экспоненциально растущий LR (ExpLR) (??!??!) <sup>11</sup>

- Вопрос авторов: действительно ли уменьшение LR является необходимым условием успешного обучения глубоких сетей?
- Авторы предлагают **экспоненциально растущую** стратегию LR:

$$\eta_t = \eta_0(1 + \alpha)^t, \quad \alpha > 0$$

- Несмотря на быстрое «взрывание» шага, обучение **всё ещё возможно**.
- Экспериментальный факт:
  - стандартные архитектуры для CIFAR-10 (например, PreResNet-32) успешно обучаются с ExpLR;
  - при корректном выборе  $\alpha$  траектория оптимизации оказывается близка к классической стратегии: фиксированный LR + weight decay.
- Наблюдение: нормализация + weight decay создают эффект, напоминающий «эффективное увеличение» LR в процессе обучения.

# Немного про LR schedulers: Экспоненциально растущий LR (ExpLR) (??!??!) <sup>11</sup>

- Вопрос авторов: действительно ли уменьшение LR является необходимым условием успешного обучения глубоких сетей?
- Авторы предлагают **экспоненциально растущую** стратегию LR:

$$\eta_t = \eta_0(1 + \alpha)^t, \quad \alpha > 0$$

- Несмотря на быстрое «взрывание» шага, обучение **всё ещё возможно**.
- Экспериментальный факт:
  - стандартные архитектуры для CIFAR-10 (например, PreResNet-32) успешно обучаются с ExpLR;
  - при корректном выборе  $\alpha$  траектория оптимизации оказывается близка к классической стратегии: фиксированный LR + weight decay.
- Наблюдение: нормализация + weight decay создают эффект, напоминающий «эффективное увеличение» LR в процессе обучения.

<sup>11</sup>Exponential Learning Rate Schedules for Deep Learning (2020)

# Немного про LR schedulers: Экспоненциально растущий LR (ExpLR) (??!??!)<sup>11</sup>

- Вопрос авторов: действительно ли уменьшение LR является необходимым условием успешного обучения глубоких сетей?
- Авторы предлагают **экспоненциально растущую** стратегию LR:

$$\eta_t = \eta_0(1 + \alpha)^t, \quad \alpha > 0$$

- Несмотря на быстрое «взрывание» шага, обучение **всё ещё возможно**.
- Экспериментальный факт:
  - стандартные архитектуры для CIFAR-10 (например, PreResNet-32) успешно обучаются с ExpLR;
  - при корректном выборе  $\alpha$  траектория оптимизации оказывается близка к классической стратегии: фиксированный LR + weight decay.
- Наблюдение: нормализация + weight decay создают эффект, напоминающий «эффективное увеличение» LR в процессе обучения.

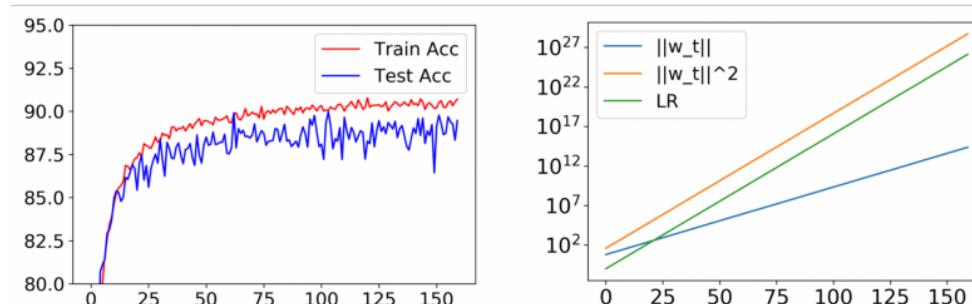
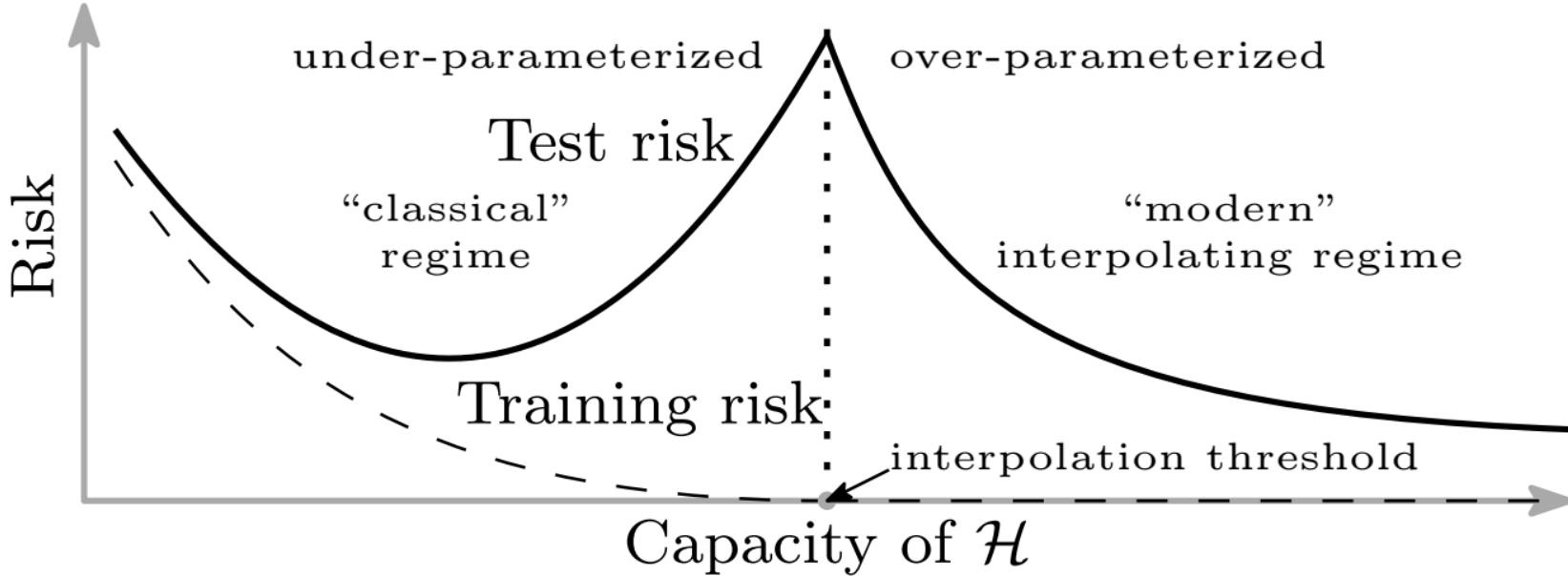


Рисунок 15. Обучение PreResNet32 на CIFAR10. Траектория с фиксированным LR и WD совпадает с ExpLR ( $\tilde{\eta}_t = 0.1 \times 1.481^t$ ) без WD. Справа: норма весов растет экспоненциально,  $|w_t|_2^2 / \tilde{\eta}_t = \text{const}$ .

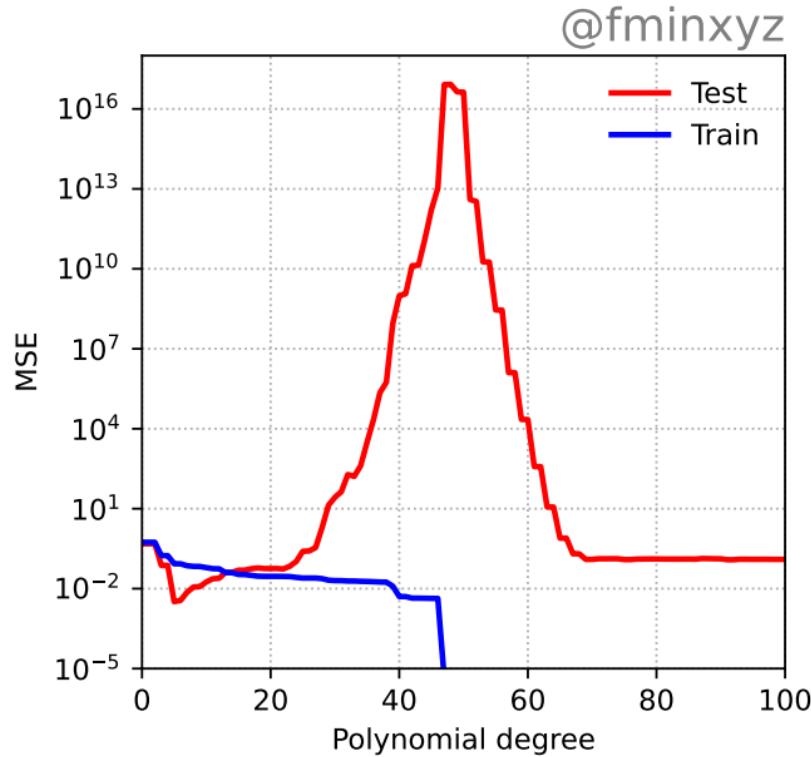
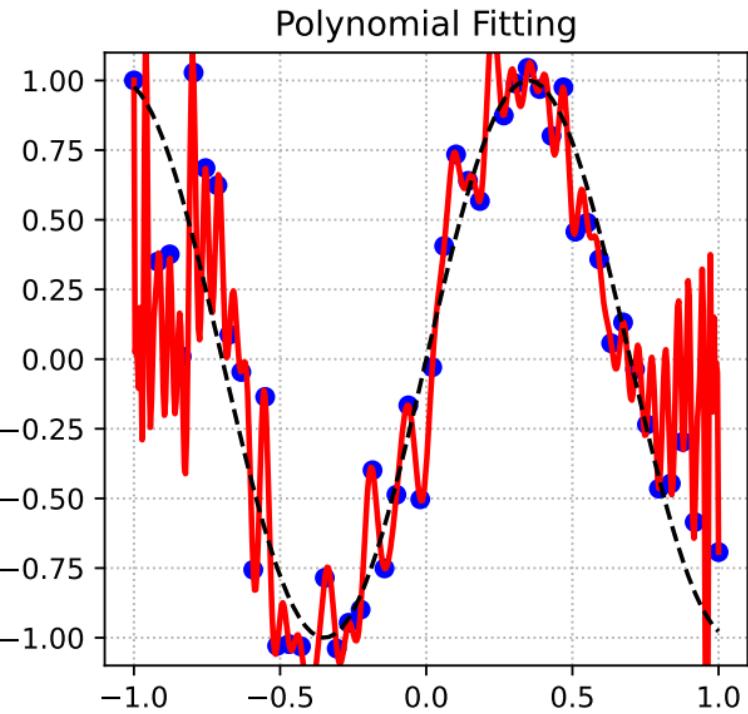
<sup>11</sup>Exponential Learning Rate Schedules for Deep Learning (2020)

# Double Descent<sup>12</sup>



<sup>12</sup>Reconciling modern machine learning practice and the bias-variance trade-off, Mikhail Belkin, Daniel Hsu, Siyuan Ma, Soumik Mandal

# Double Descent



# Grokking<sup>13</sup>

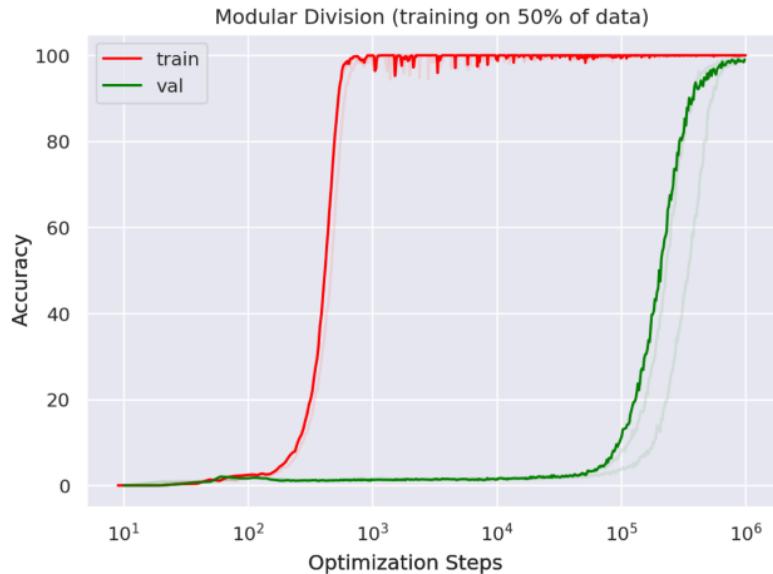


Рисунок 16. Training transformer with 2 layers, width 128, and 4 attention heads, with a total of about  $4 \cdot 10^5$  non-embedding parameters. Reproduction of experiments (~ half an hour) is available here

- Рекомендую посмотреть лекцию Дмитрия Ветрова **Удивительные свойства функции потерь в нейронной сети** (*Surprising properties of loss landscape in overparameterized models*). видео, Презентация

# Grokking<sup>13</sup>

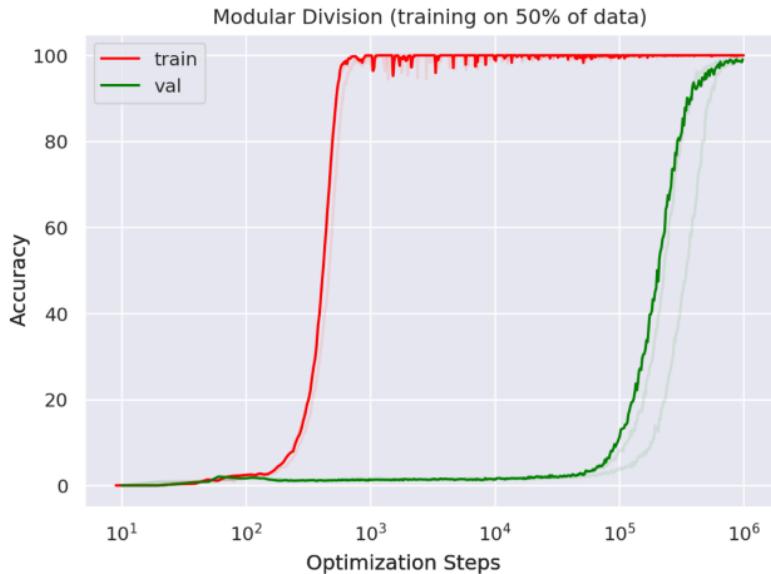


Рисунок 16. Training transformer with 2 layers, width 128, and 4 attention heads, with a total of about  $4 \cdot 10^5$  non-embedding parameters. Reproduction of experiments (~ half an hour) is available here

- Рекомендую посмотреть лекцию Дмитрия Ветрова **Удивительные свойства функции потерь в нейронной сети** (*Surprising properties of loss landscape in overparameterized models*). видео, Презентация
- Автор канала Свидетели Градиента собирает интересные наблюдения и эксперименты про гроккинг.

# Grokking<sup>13</sup>

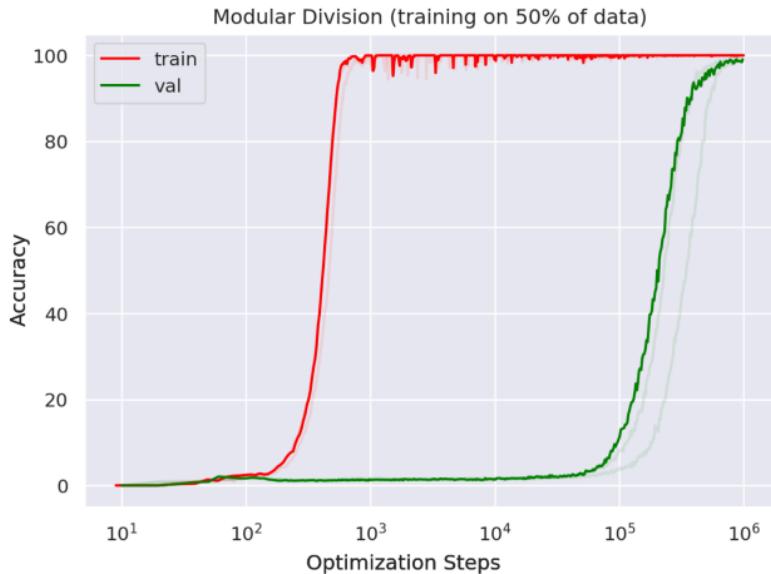


Рисунок 16. Training transformer with 2 layers, width 128, and 4 attention heads, with a total of about  $4 \cdot 10^5$  non-embedding parameters. Reproduction of experiments (~ half an hour) is available [here](#)

- Рекомендую посмотреть лекцию Дмитрия Ветрова **Удивительные свойства функции потерь в нейронной сети** (*Surprising properties of loss landscape in overparameterized models*). видео, Презентация
- Автор канала Свидетели Градиента собирает интересные наблюдения и эксперименты про гроккинг.
- Также есть видео с его докладом **Чем не является гроккинг**.

<sup>13</sup> Grokking: Generalization Beyond Overfitting on Small Algorithmic Datasets, Alethea Power, Yuri Burda, Harri Edwards, Igor Babuschkin, Vedant Misra