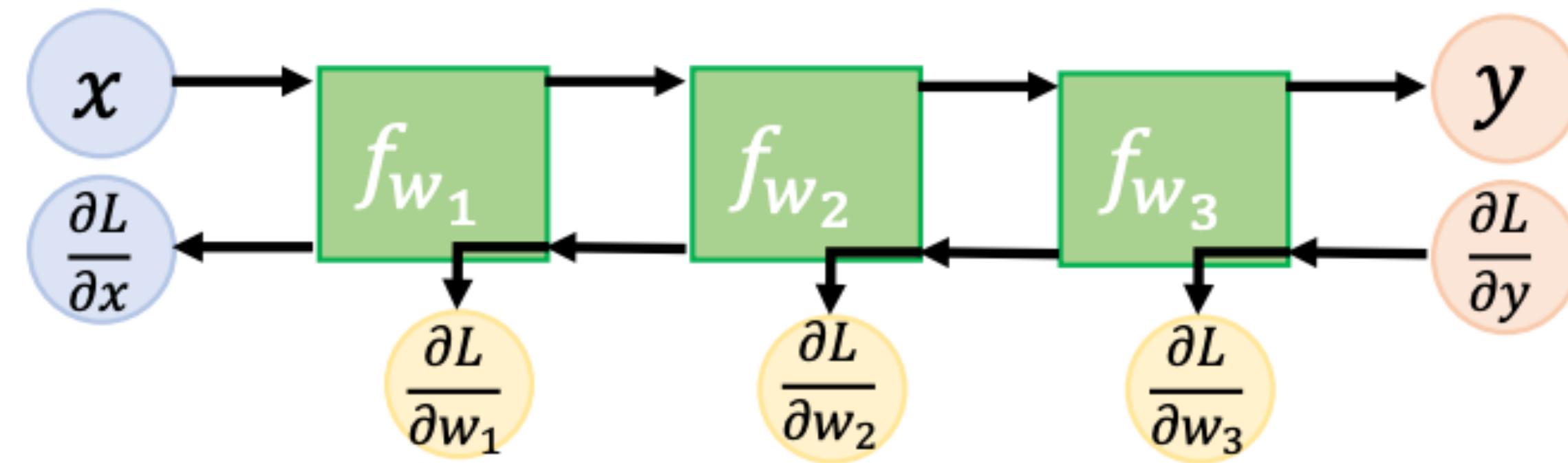


Методы оптимизации

Лекция 3

Зайдель Петр 22.09.22

Recap. Backprop



Recap: SGD

(Mini batch) SGD:

Ищем минимум $Q(w) = \frac{1}{N} \sum_1^N L(w, x_i, y_i)$

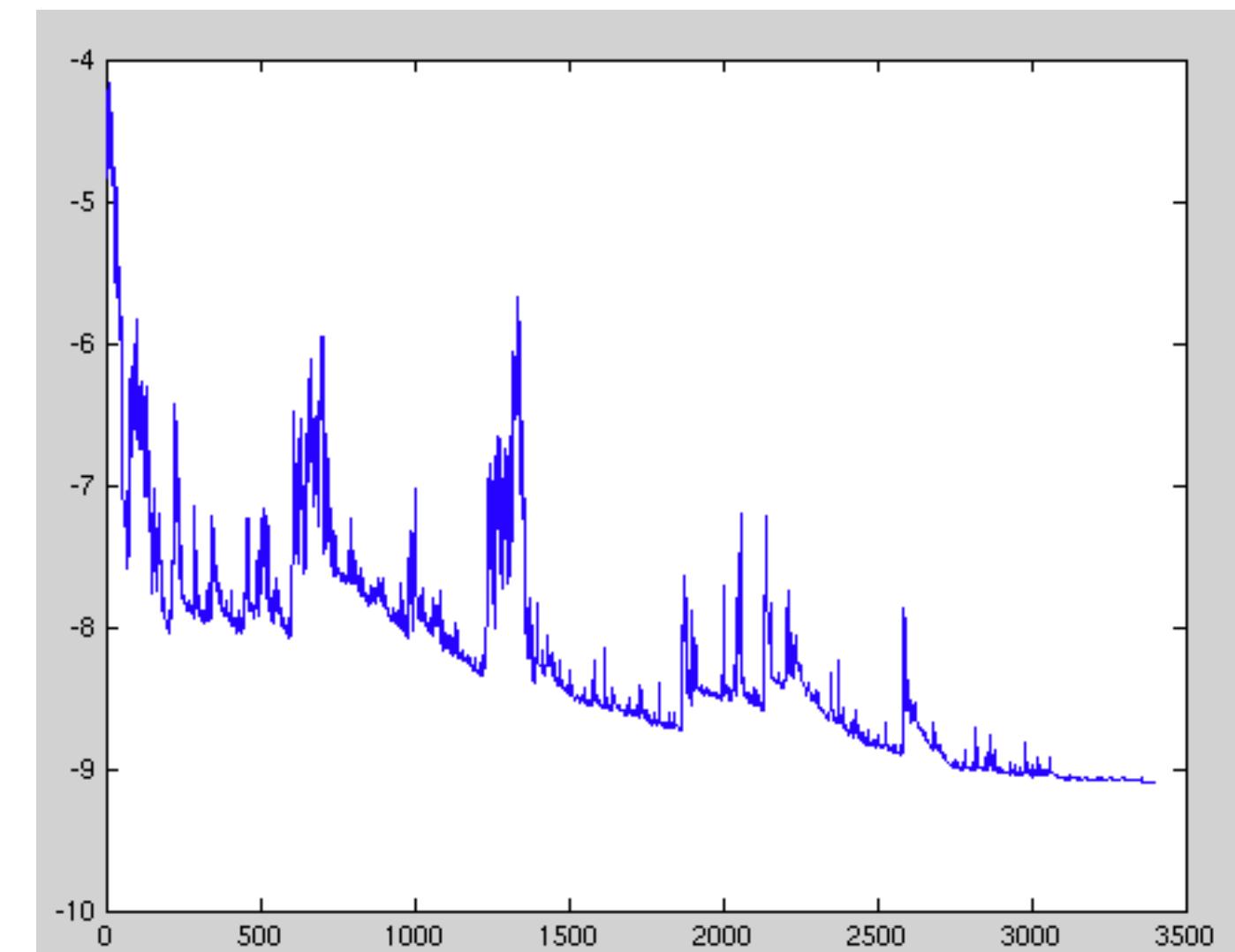
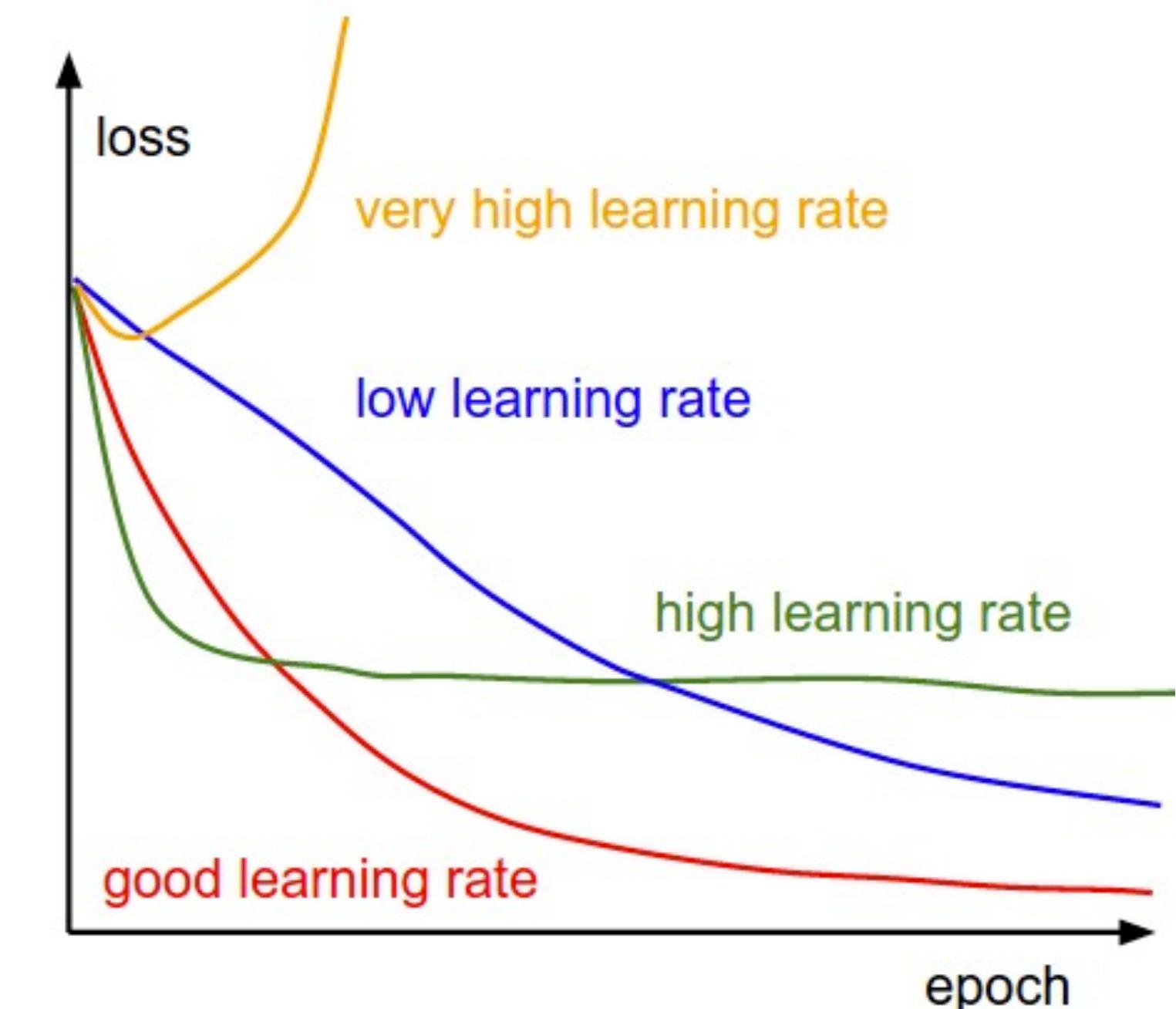
Инициализация параметров $w = w_0$, начальный шаг α_0

1. Случайно выбираем n объектов

2. Оценка градиента $\nabla_w Q(w_{old}) = \frac{1}{n} \sum_1^n \nabla_w L(w_{old}, x_j, y_j)$

3. $w_{new} = w_{old} - \alpha_k \nabla_w Q(w_{old})$

4. Повторять до сходимости



Recap: SGD

Для выпуклых функций гарантируется сходимость, если:

$$\nu_t \xrightarrow{t \rightarrow \infty} 0$$

$$\sum_{t=0}^{\infty} \nu_t = \infty$$

$$\sum_{t=0}^{\infty} \nu_t^2 < \infty$$

Ландшафт функции потерь

Почему мы вообще используем градиентный спуск?

- Большинство локальных минимумов целевой функции сконцентрированы в сравнительно небольшом подпространстве весов. Соответствующие этим минимумам сети дают примерно одинаковый loss на тестовом датасете.
- Сложность ландшафта увеличивается по приближении к глобальным минимумам. Почти во всём объёме пространства весов подавляющая часть седловых точек имеет большое количество направлений, по которым из них можно сбежать. Чем ближе к центру кластера минимумов, тем меньше «направлений побега» у встреченных на пути седловых точек.
- В глубоких нейронных сетях основным препятствием для обучения являются седловые точки, а не локальные минимумы, как считалось ранее.

Ландшафт функции потерь

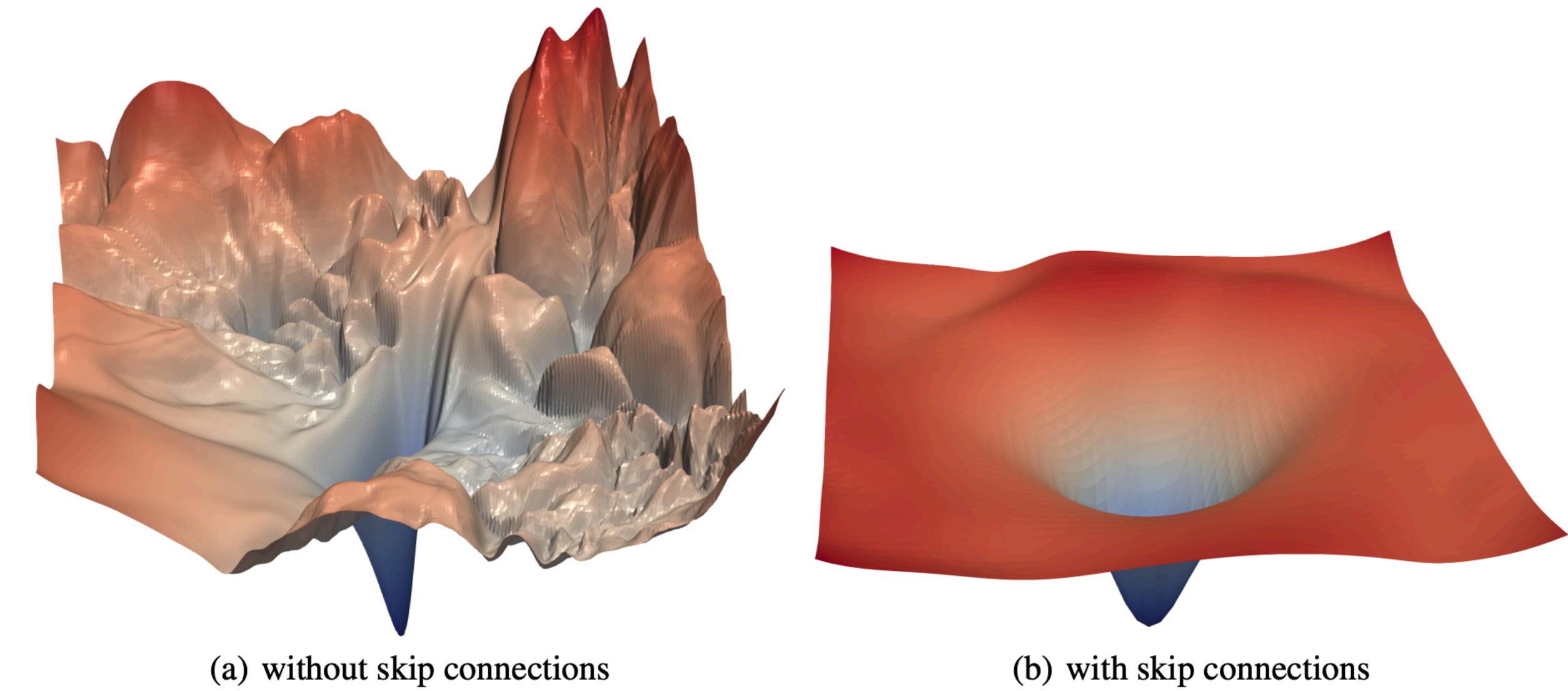
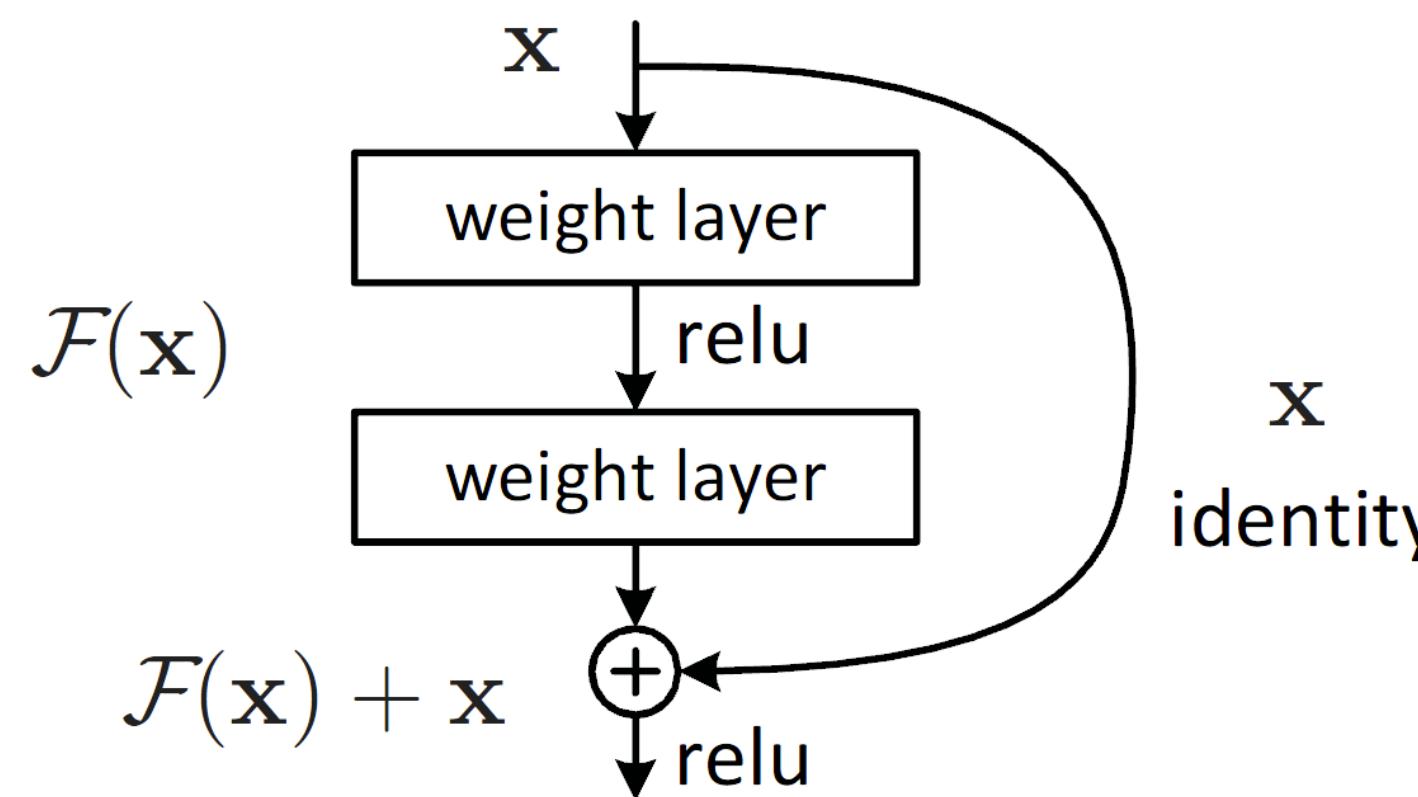
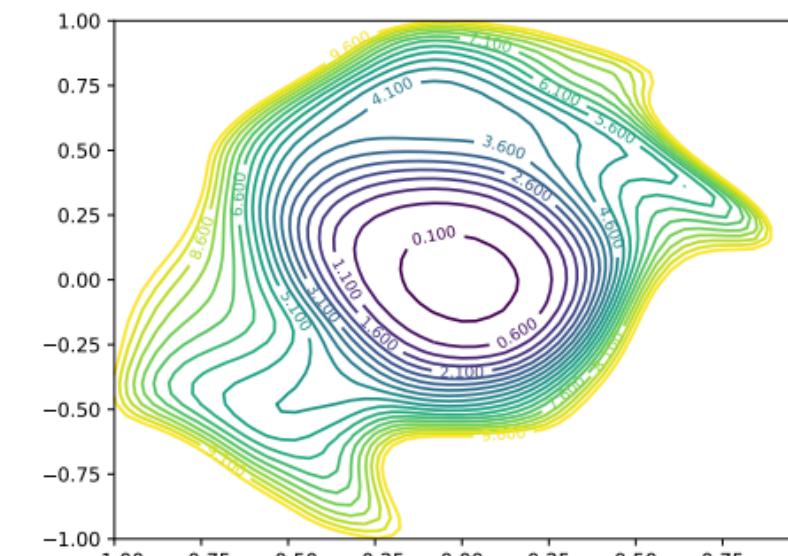
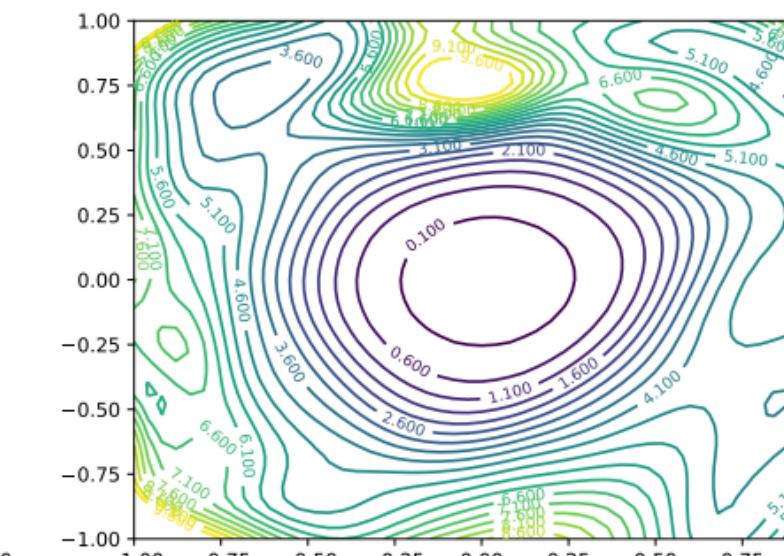


Figure 1: The loss surfaces of ResNet-56 with/without skip connections. The proposed filter normalization scheme is used to enable comparisons of sharpness/flatness between the two figures.
32nd Conference on Neural Information Processing Systems (NIPS 2018), Montréal, Canada.

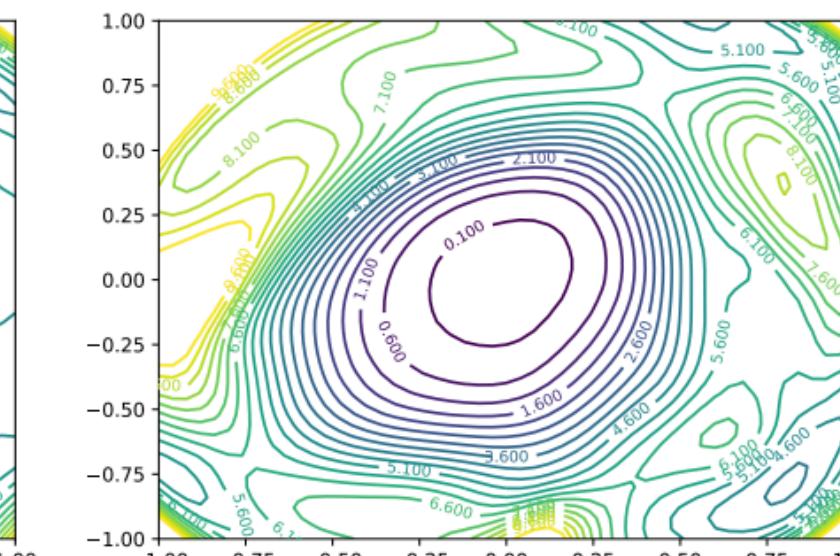
Ландшафт функции потерь



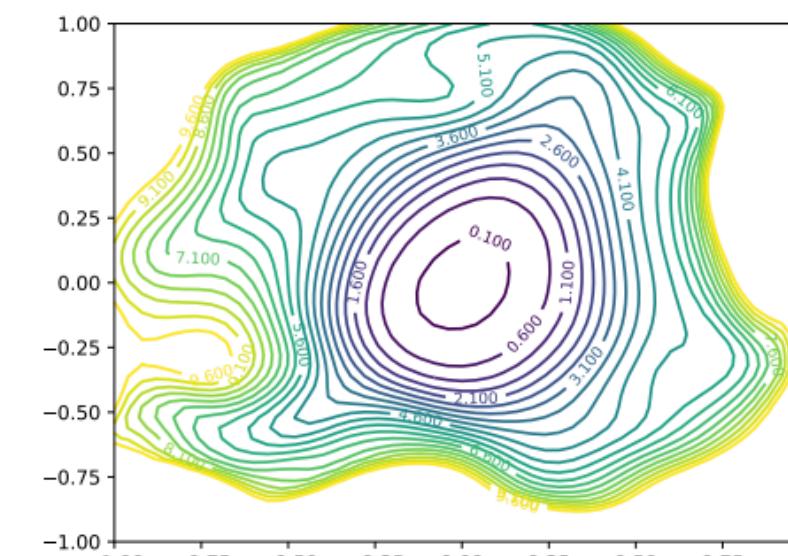
(a) ResNet-20, 7.37%



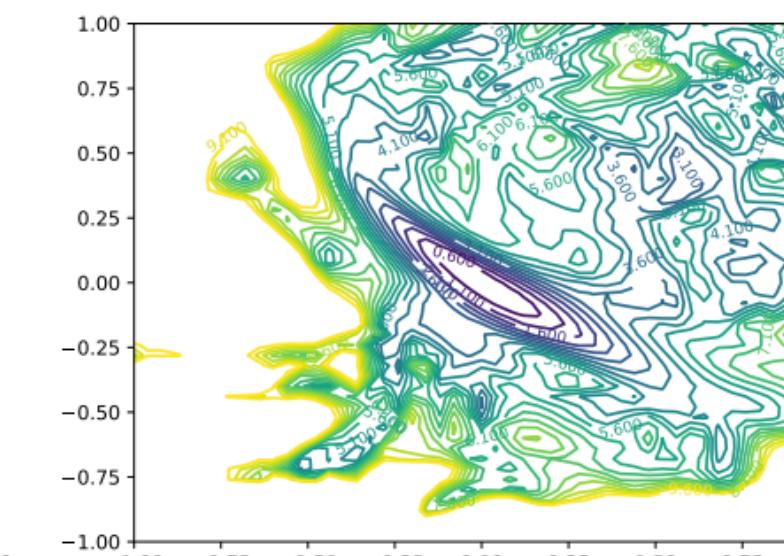
(b) ResNet-56, 5.89%



(c) ResNet-110, 5.79%



(d) ResNet-20-NS, 8.18%



(e) ResNet-56-NS, 13.31% (f) ResNet-110-NS, 16.44%

Double Descent

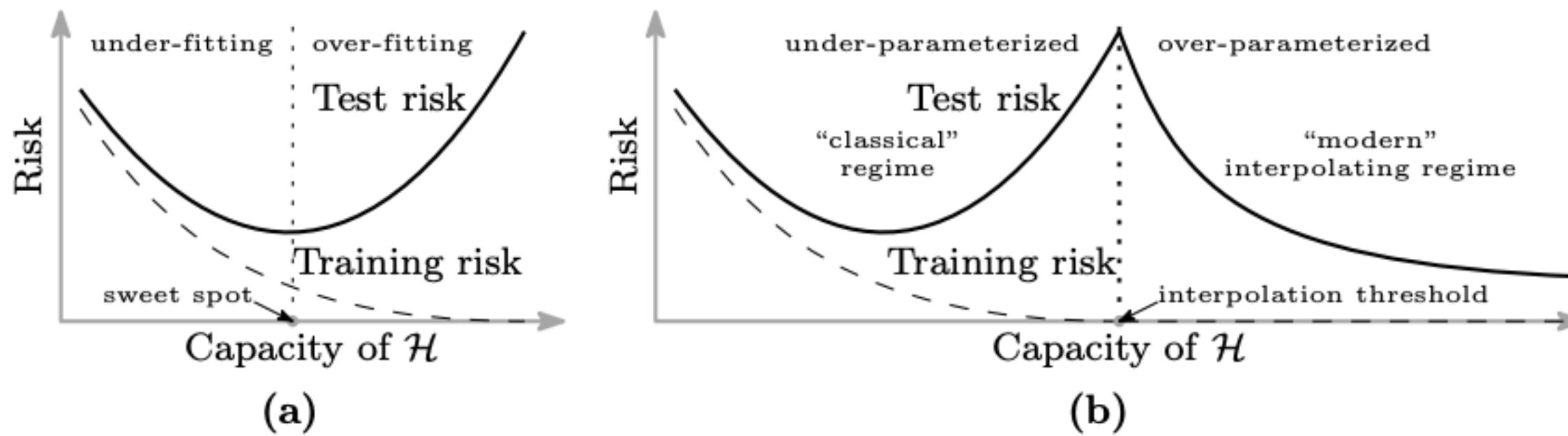


Figure 1: **Curves for training risk (dashed line) and test risk (solid line).** (a) The classical *U-shaped risk curve* arising from the bias-variance trade-off. (b) The *double descent risk curve*, which incorporates the U-shaped risk curve (i.e., the “classical” regime) together with the observed behavior from using high capacity function classes (i.e., the “modern” interpolating regime), separated by the interpolation threshold. The predictors to the right of the interpolation threshold have zero training risk.

Гипотеза - нейросеть сначала запоминает
трейн данные, находя узкий минимум, а
затем спускается в более широкие
минимумы улучшая обобщение

<https://arxiv.org/pdf/1812.11118.pdf>

<https://www.youtube.com/watch?v=BzyJCFX4dxg>

Mode connectivity

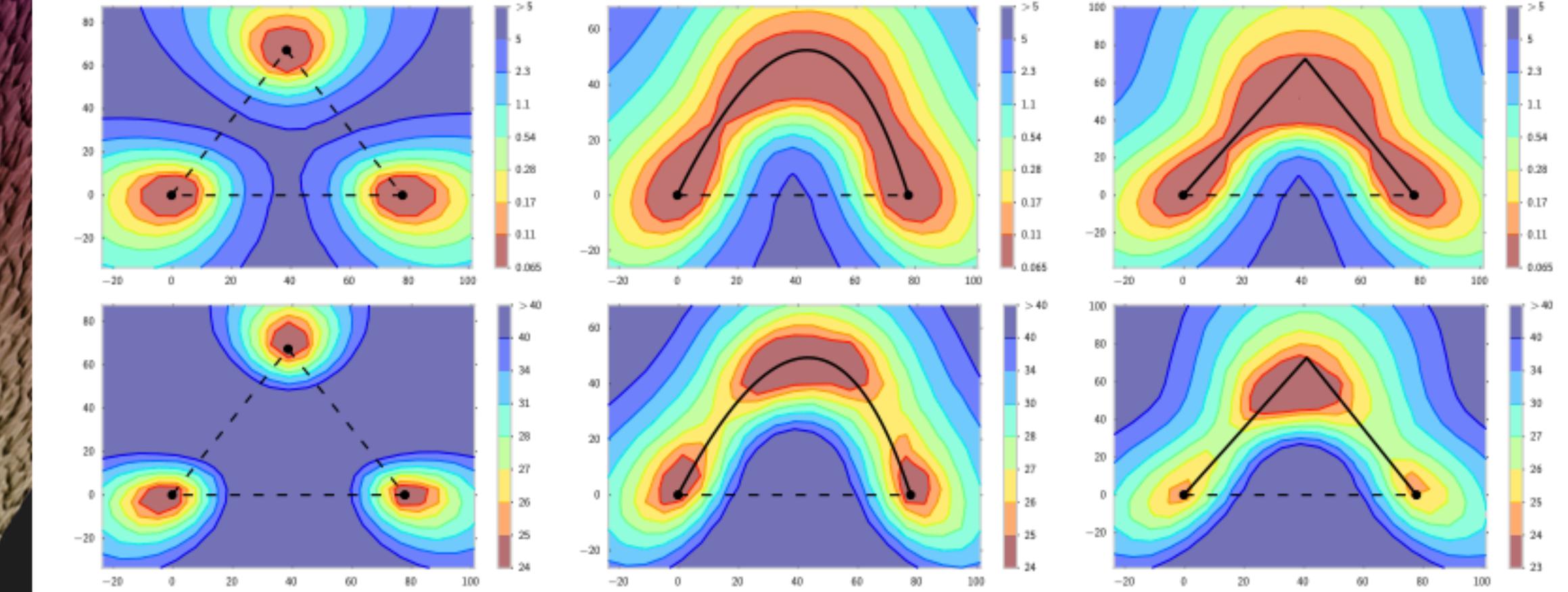
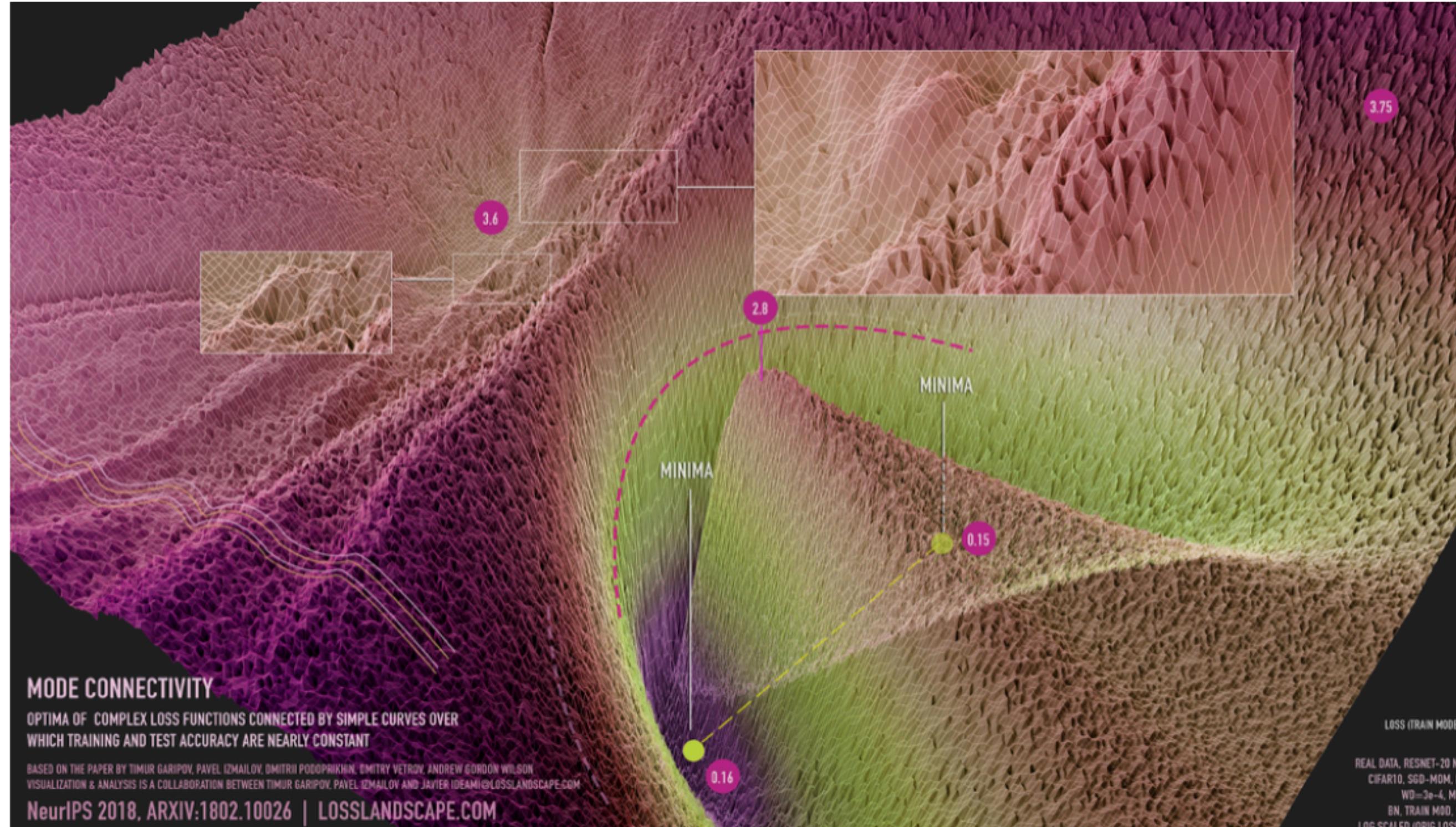


Figure 4: The ℓ_2 -regularized cross-entropy train loss (**Top**) and test error (**Bottom**) surfaces of a deep residual network (ResNet-164) on CIFAR-100. **Left:** Three optima for independently trained networks. **Middle and Right:** A quadratic Bezier curve, and a polygonal chain with one bend, connecting the lower two optima on the left panel along a path of near-constant loss. Notice that in each panel, a direct linear path between each mode would incur high loss.

<https://arxiv.org/pdf/1802.10026.pdf>

<https://www.youtube.com/watch?v=dqX2LBcp5Hs>

Проблемы SGD

1. Сложно подобрать learning rate (размер шага). Слишком маленький - очень долго сходится, большой - может вообще разойтись или застрять в локально оптимуме
2. Методы подбора learning rate не адаптируются к данным. Требуют настройки каждый раз.
4. Одинаковый шаг для всех параметров. Не учитываем, что разные параметры могут сходиться с разной скоростью (особенно заметно на разреженных данных).
5. SGD легко застrevает в седловых точках.

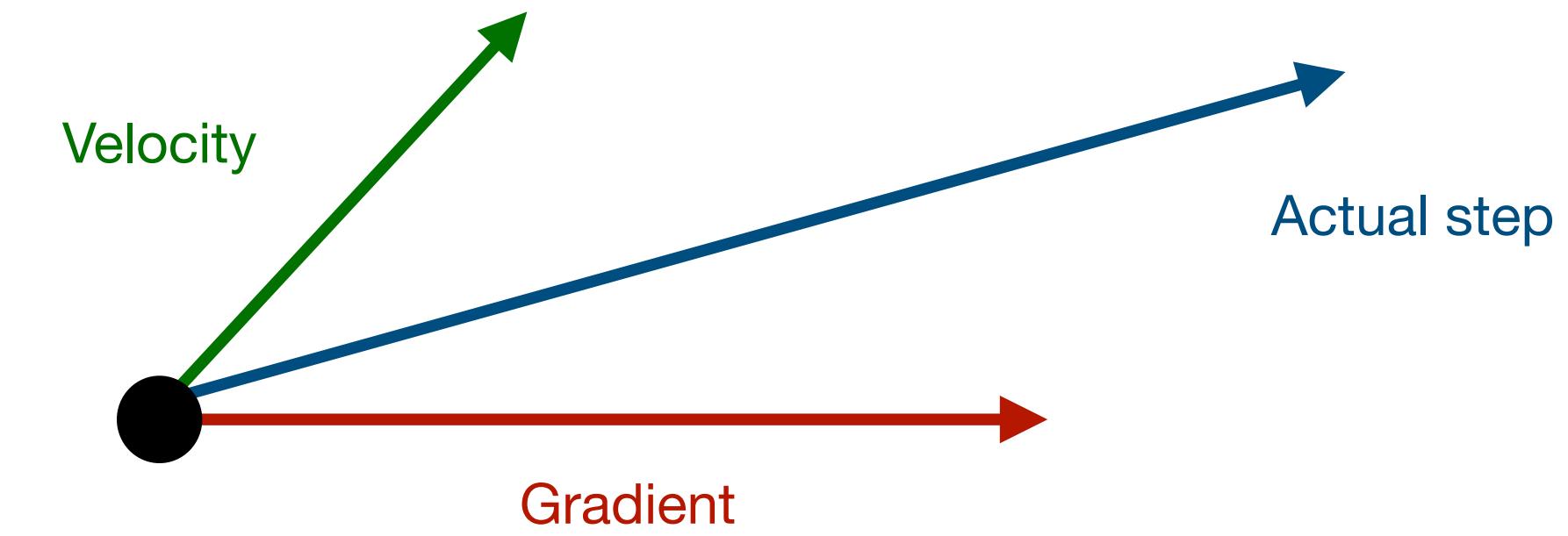
Momentum

$$v_t = \gamma v_{t-1} + \eta_t \nabla_{\theta} J(\theta_{t-1}) - \text{Инерция}$$

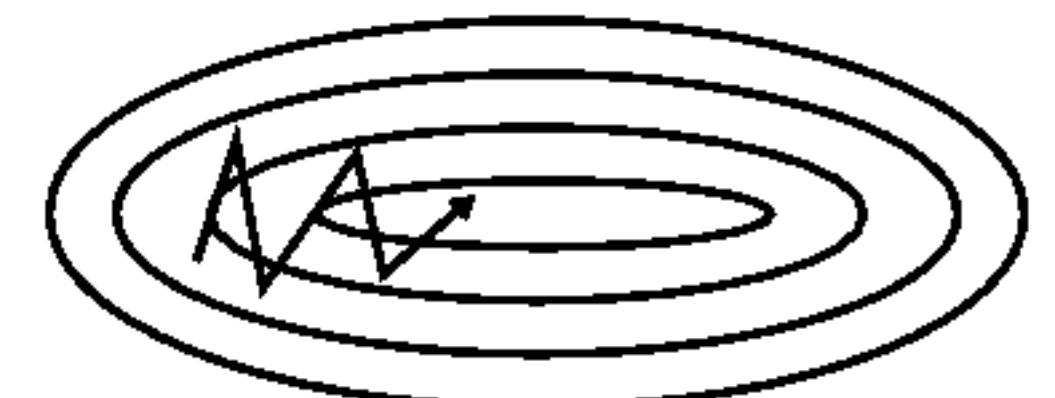
$$\theta_t = \theta_{t-1} - v_t$$

Рекомендовано брать

$$\gamma = 0.9$$



SGD



SGD with momentum

Nesterov accelerated gradient

Следующая позиция приблизительно равна

$$\theta_{t-1} - \gamma v_{t-1}$$

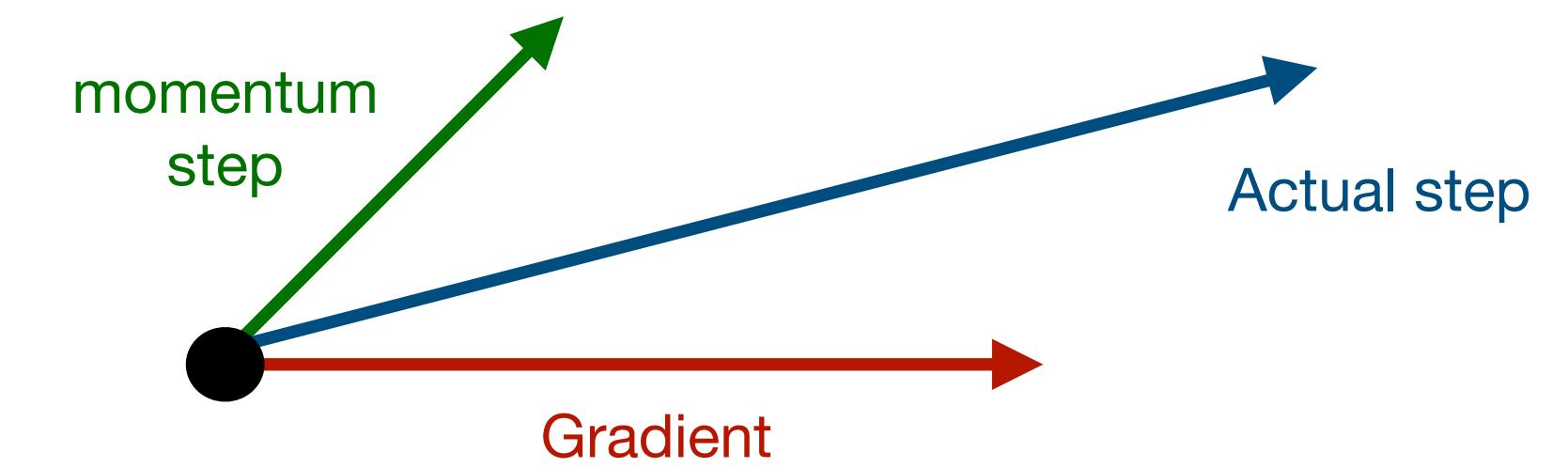
Вычисление градиента в новой точке дает возможность скорректировать направление движения

$$v_t = \gamma v_{t-1} + \eta_t \nabla_\theta J(\theta_{t-1} - \gamma v_{t-1})$$

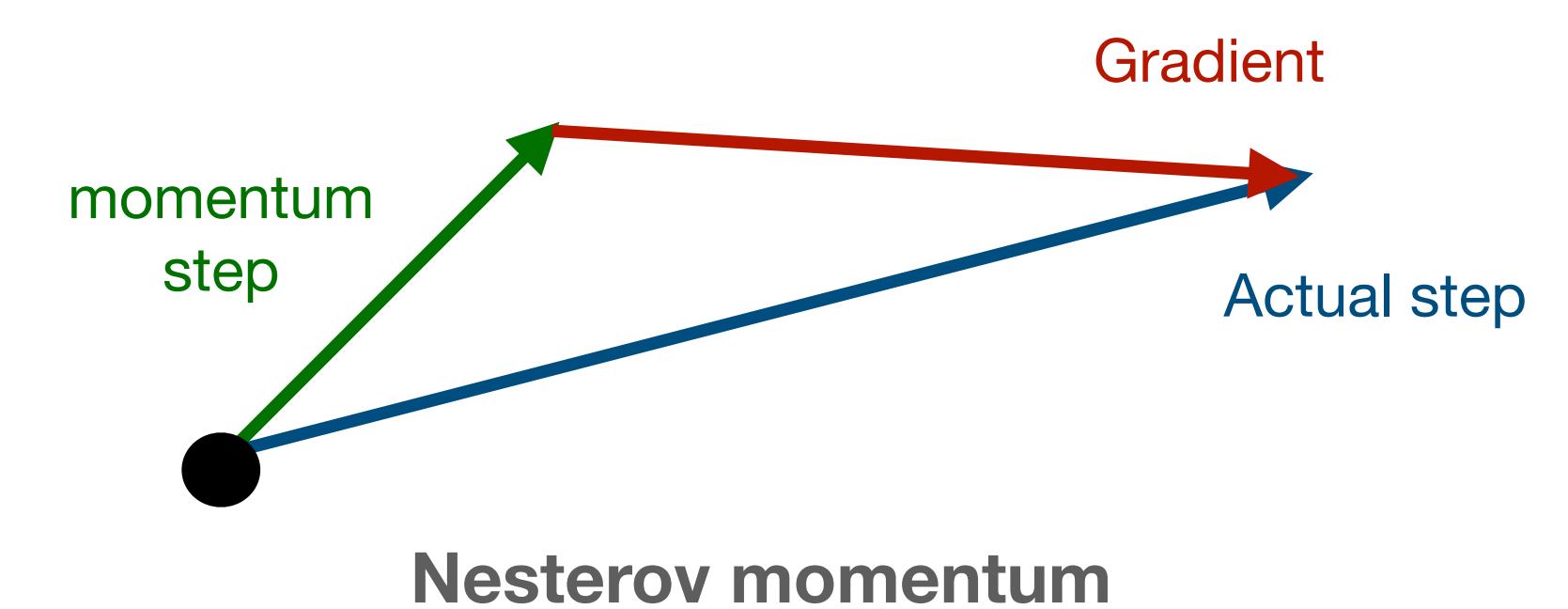
$$\theta_t = \theta_{t-1} - v_t$$

Рекомендовано брать

$$\gamma = 0.9$$



simple momentum



Nesterov momentum

SGD:

$$\theta_{t+1} = \theta_t - \eta_t \nabla_{\theta} J(\theta_{t-1})$$

SGD + Momentum:

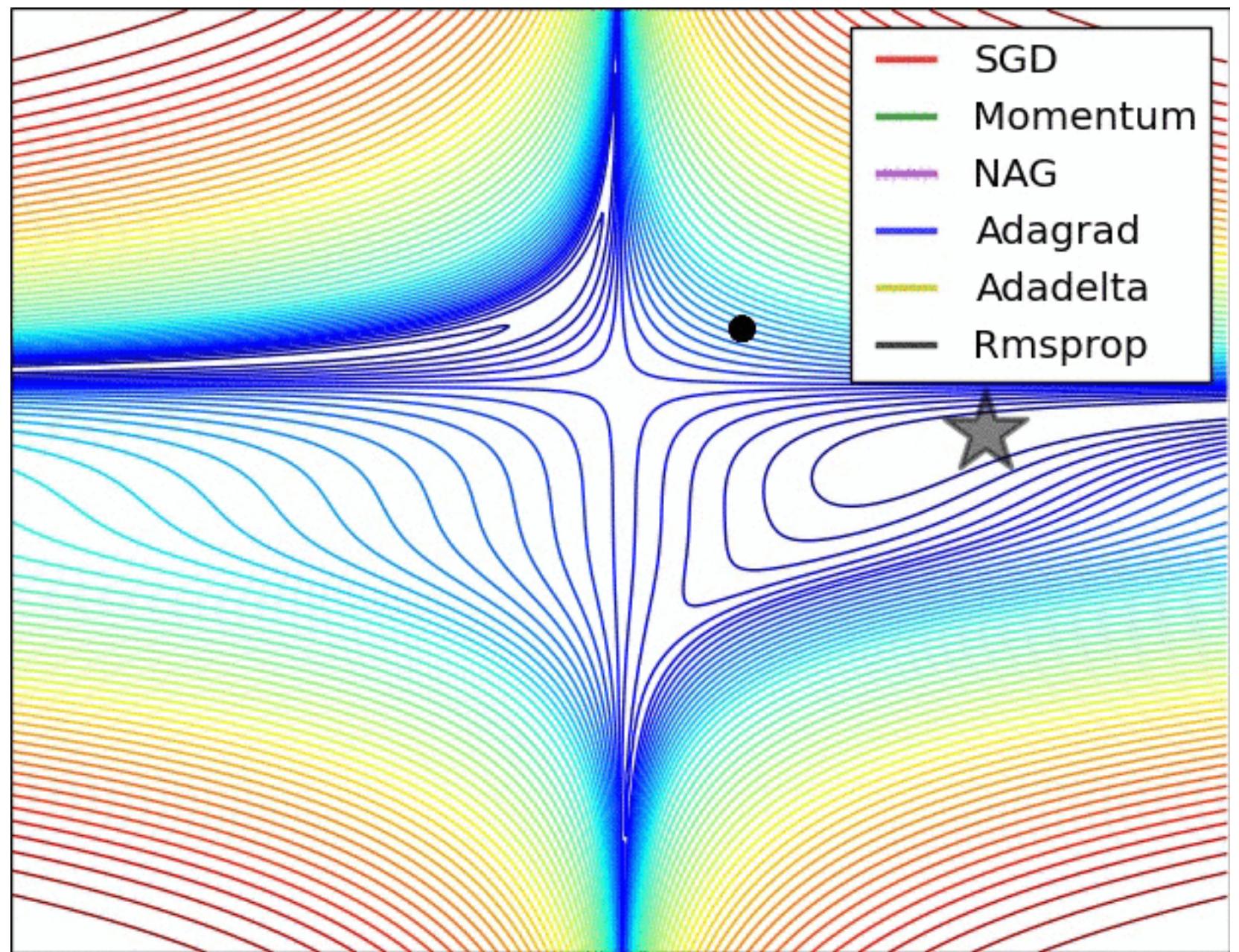
$$v_t = \gamma v_{t-1} + \eta_t \nabla_{\theta} J(\theta_{t-1})$$

$$\theta_{t+1} = \theta_t - v_t$$

Nesterov Momentum

$$v_t = \gamma v_{t-1} + \eta_t \nabla_{\theta} J(\theta_{t-1} - \gamma v_{t-1})$$

$$\theta_{t+1} = \theta_t - v_t$$



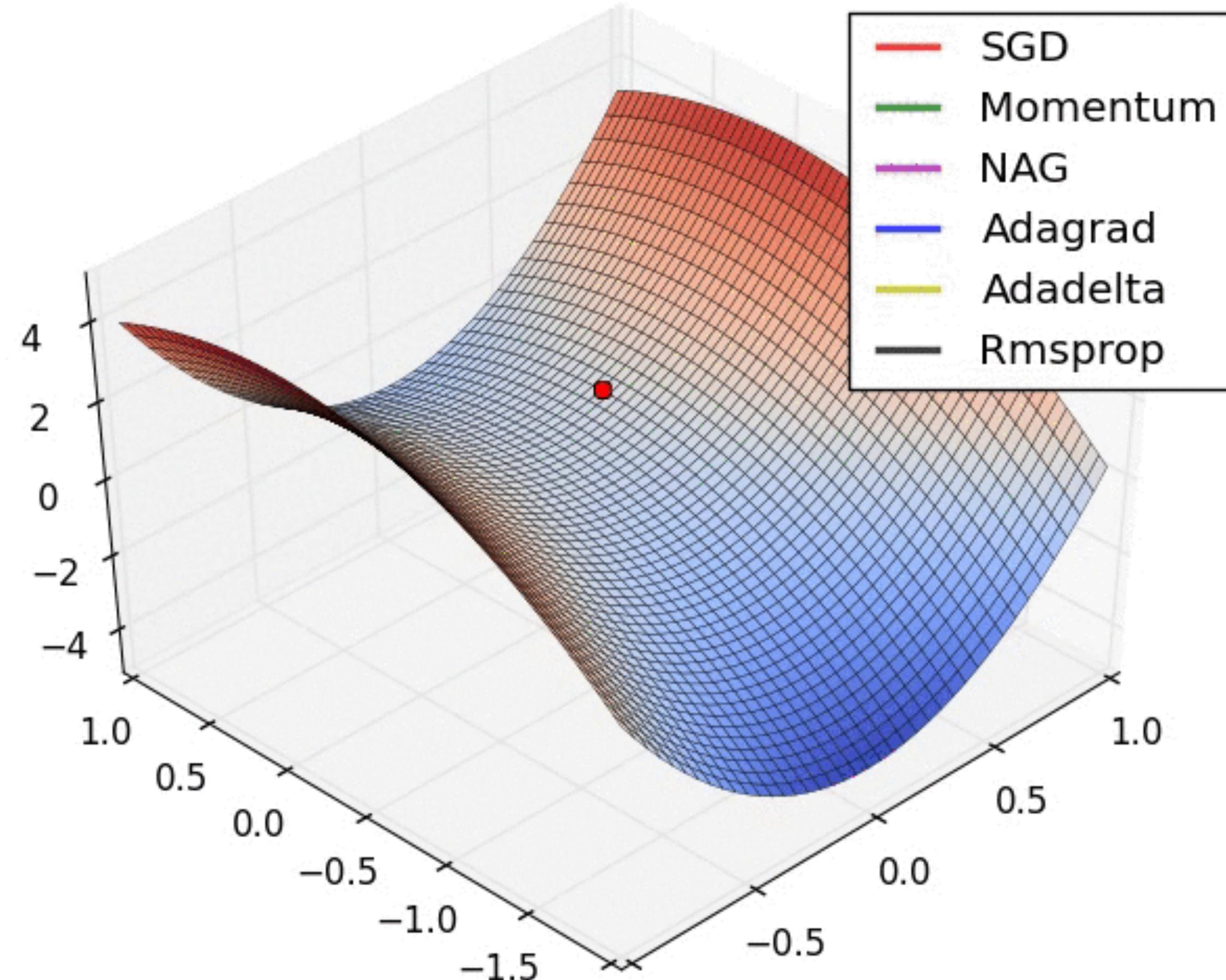
Общая проблема: одинаковый шаг для всех параметров

Трудно подобрать η_t

На практике используют разные расписания:

- $\eta_t = \gamma^t \eta_0$
- ступенчатая функция
- иногда в конце обучения немного повышают η_t , чтобы выйти из локального оптимума
- и многие другие

Седловая точка:



AdaGrad:

$g_{t,i} = \nabla_{\theta_i} J(\theta_t)$ - градиент на t шаге по параметру θ_i

$G_{t,ii}$ - сумма квадратов $g_{t,i}$ вплоть до текущего, $G_{t,ii} \in R^{d \times d}$

$$\theta_{t+1,i} = \theta_{t,i} - \frac{\eta}{\sqrt{G_{t,ii} + \epsilon}} \cdot g_{t,i}$$

Стандартные значения $\eta = 0.01$, $\epsilon = 10^{-8}$

Мотивация: маленькие обновления для часто встречающихся параметров, больше для редких

Какова проблема этого метода ?

AdaGrad:

$g_{t,i} = \nabla_{\theta_i} J(\theta_t)$ - градиент на t шаге по параметру θ_i

$G_{t,ii}$ - сумма квадратов $g_{t,i}$ вплоть до текущего, $G_{t,ii} \in R^{d \times d}$

$$\theta_{t+1,i} = \theta_{t,i} - \frac{\eta}{\sqrt{G_{t,ii} + \epsilon}} \cdot g_{t,i}$$

Стандартные значения $\eta = 0.01$, $\epsilon = 10^{-8}$

Мотивация: маленькие обновления для часто встречающихся параметров, больше для редких

Какова проблема этого метода ?

$G_{t,ii}$ не убывает \Rightarrow затухание обновлений

RMSProp / Adadelta:

Будем использовать последние несколько значений g_t^2 для подсчета G_t
Экспоненциальное среднее

$$E[g^2]_t = \gamma E[g^2]_{t-1} + (1 - \gamma)g_t^2$$
$$\gamma = 0.9$$

Обновление весов

$$\theta_t = \theta_{t-1} - \Delta\theta_t$$

$$\Delta\theta_t = \frac{\eta}{\sqrt{E[g^2]_t + \epsilon}} \cdot g_t = \frac{\eta}{RMS[g]_t} \quad - RMSprop$$

Adadelta: избавимся от η

$$\Delta\theta_t = \frac{RMS[\Delta\theta]_{t-1}}{RMS[g]_t}$$

Adadelta: интуиция

Метод Ньютона: $\Delta\theta_t = (\nabla^2 J)^{-1} \nabla J$
быстро сходится, но очень тяжелый и требует
много памяти

Диагональная аппроксимация

$$\nabla^2 J \approx \text{diag}\left(\frac{\partial J}{\partial \theta_{t,i}^2}\right)$$

$$\Delta\theta_{t,i} = \left(\frac{\partial J}{\partial \theta_{t,i}^2} \right)^{-1} \frac{\partial J}{\partial \theta_{t,i}}$$

$$d\theta = g / G^2 \rightarrow G^2 = d\theta / g$$

Adam (Adaptive Moment Estimation)

$$\begin{aligned}m_t &= \beta_1 m_{t-1} + (1 - \beta_1) g_t \\ \nu_t &= \beta_2 \nu_{t-1} + (1 - \beta_2) g_t^2\end{aligned}$$

m_t , ν_t инициализируются нулями, поэтому долгий “разгон”
=> нужно уменьшить инерцию в начале обучения

Надо обеспечить несмещенность $E[m_t] = E[g_t]$, $E[\nu_t] = E[g_t^2]$

Поправка:

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t}$$

$$\hat{\nu}_t = \frac{\nu_t}{1 - \beta_2^t}$$

$$\theta_{t+1} = \theta_t - \frac{\eta \hat{m}_t}{\sqrt{\hat{\nu}_t} + \epsilon}, \quad \epsilon = 10^{-8}$$

Критерии остановки

Когда остановить обучение?

- Превышен лимит по числу итераций или времени
- Качество на валидации начало ухудшаться
- $J(\theta) \leq \epsilon J(\theta_0)$
- $||\nabla J(\theta)|| \leq \epsilon ||\nabla J(\theta_0)||$

Сравнение методов

