OTUS

ОНЛАЙН-ОБРАЗОВАНИЕ

Методы оптимизации

...а чтобы попасть в другое место,
нужно бежать вдвое быстрее.

Артур Кадурин
Преподаватель

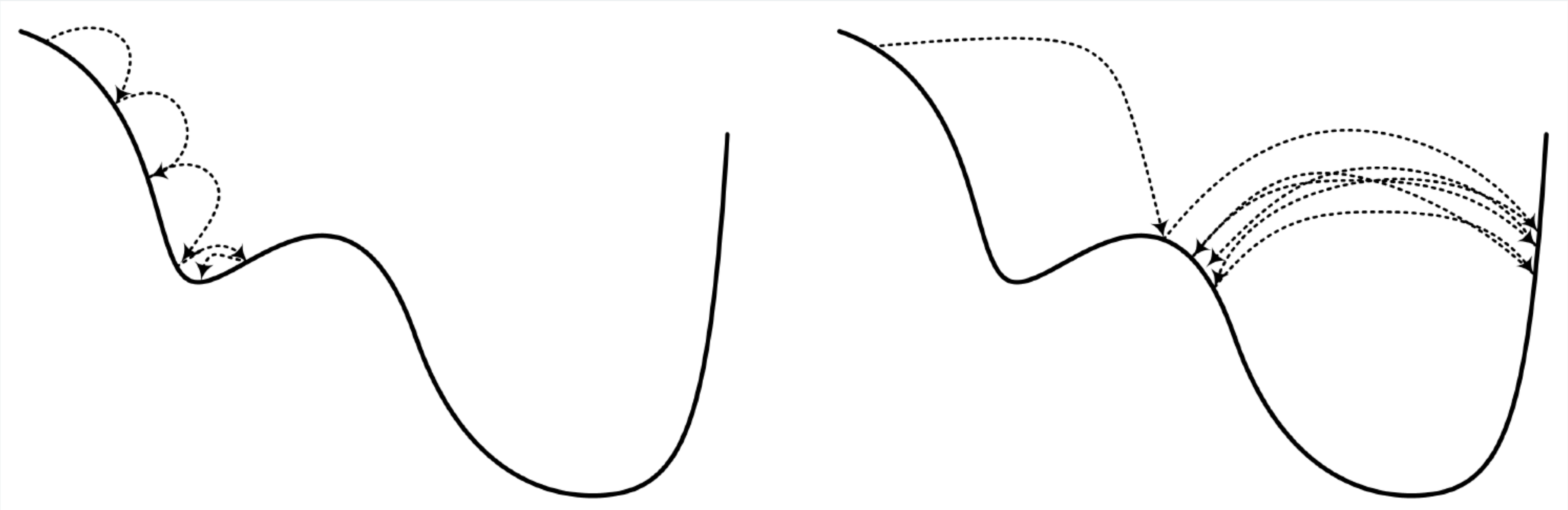
План на сегодня

1. **Скорость**
2. Инерция
3. Адаптация
4. Практика

Градиентный спуск

$$u_t = \eta \nabla_\theta \mathcal{L}(\theta_t)$$
$$\theta_{t+1} = \theta_t - u_t$$

Каждый шаг мы вычисляем изменение весов как градиент от функции ошибки и делаем небольшой шаг в нужную сторону.



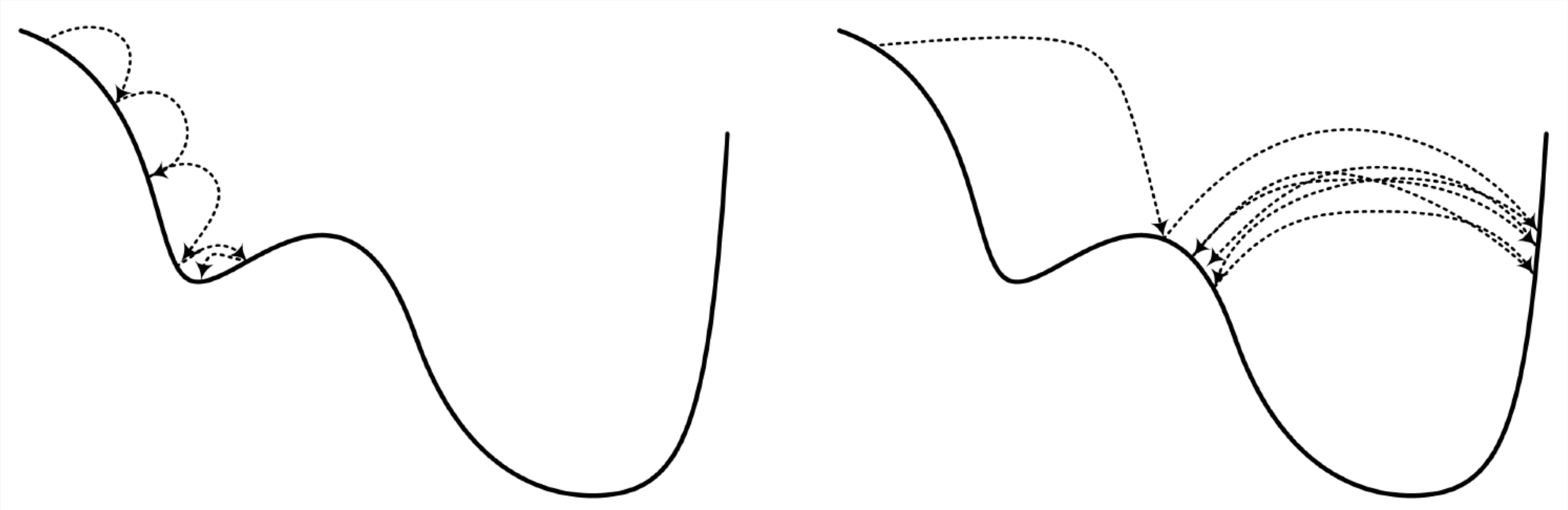
Градиентный спуск

$$u_t = \eta \nabla_\theta \mathcal{L}(\theta_t)$$

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

$$\eta = \eta_0 e^{-\frac{t}{T}}$$

Каждый шаг мы вычисляем изменение весов как градиент от функции ошибки и делаем небольшой шаг в нужную сторону. От размера шага может зависеть результат. Типичный способ: уменьшать шаг со временем.



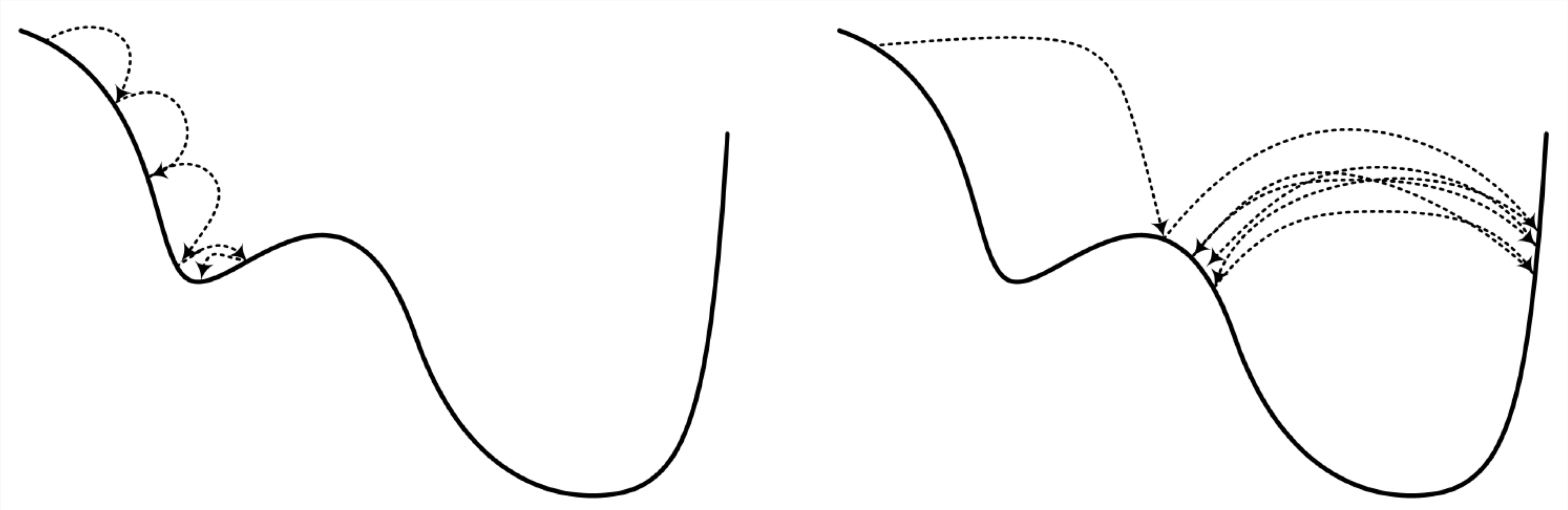
Градиентный спуск

$$u_t = \eta \nabla_\theta \mathcal{L}(\theta_t)$$

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

$$\eta = \eta_0 e^{-\frac{t}{T}}$$

Каждый шаг мы вычисляем изменение весов как градиент от функции ошибки и делаем небольшой шаг в нужную сторону. От размера шага может зависеть результат. Типичный способ: уменьшать шаг со временем.
Но что если у нас разный наклон в разных измерениях?



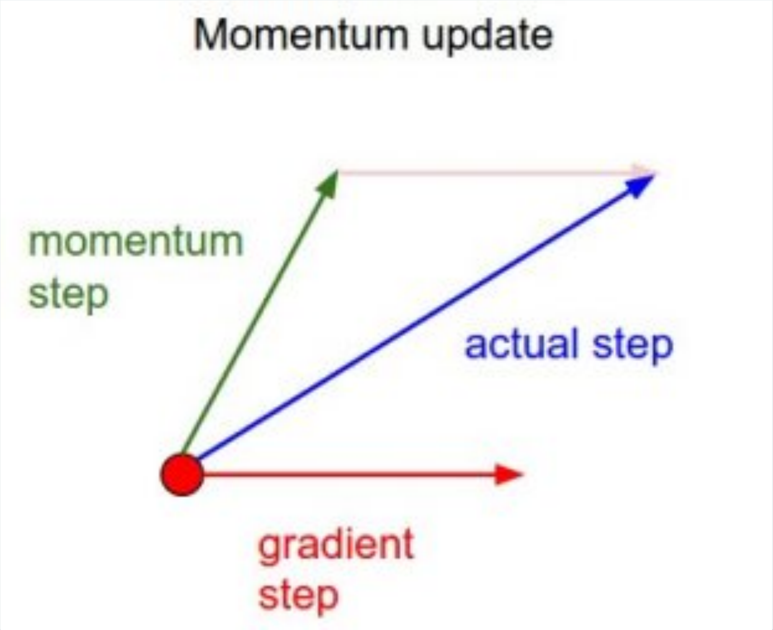
План на сегодня

1. Скорость
2. **Инерция**
3. Адаптация
4. Практика

Метод моментов

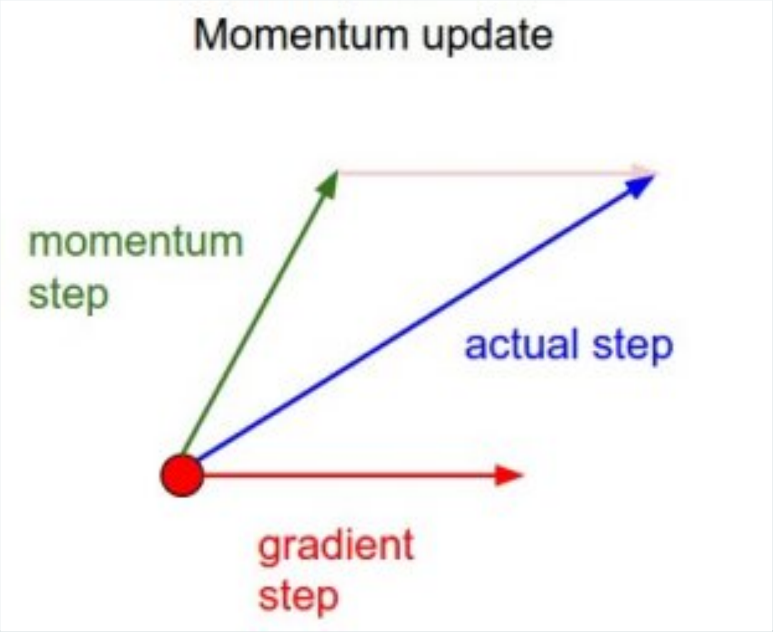
$$u_t = \gamma u_{t-1} + \eta \nabla_\theta \mathcal{L}(\theta_t)$$
$$\theta_{t+1} = \theta_t - u_t$$

Давайте накопим «инерцию» движения по ландшафту нашей функции ошибки. Тогда, если градиент какого-то параметра «скачет», мы будем менять его медленно, а если мы долго двигались в одном направлении, то накопим скорость.



Метод моментов

$$u_t = \gamma u_{t-1} + \eta \nabla_\theta \mathcal{L}(\theta_t)$$
$$\theta_{t+1} = \theta_t - u_t$$



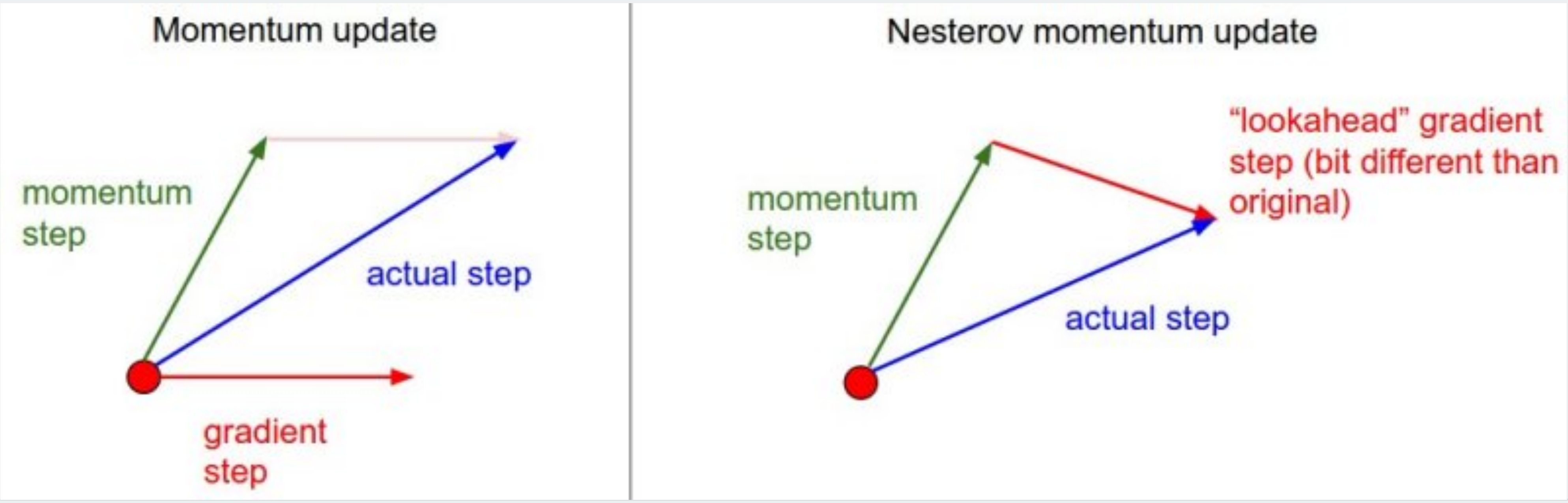
Давайте накопим «инерцию» движения по ландшафту нашей функции ошибки. Тогда, если градиент какого-то параметра «скачет», мы будем менять его медленно, а если мы долго двигались в одном направлении, то накопим скорость.

Какое улучшение можно сделать?

Метод моментов Нестерова

$$u_t = \gamma u_{t-1} + \eta \nabla_\theta \mathcal{L}(\theta_t - \gamma u_{t-1})$$
$$\theta_{t+1} = \theta_t - u_t$$

Давайте накопим «инерцию» движения по ландшафту нашей функции ошибки. Тогда, если градиент какого-то параметра «скачет», мы будем менять его медленно, а если мы долго двигались в одном направлении, то накопим скорость.



План на сегодня

1. Скорость
2. Инерция
3. **Адаптация**
4. Практика

Adagrad

$$u_t = \frac{\eta}{\sqrt{G_t + \epsilon}} \nabla_\theta \mathcal{L}(\theta_t)$$

$$G_t = G_{t-1} + g_t^2$$

Adagrad основан на той же идее: большая вариативность градиента должна уменьшать скорость обучения и наоборот.
Теперь, если градиенты становятся очень большими, то скорость обучения в соответствующем направлении быстро затухает.
Минусы?

Adagrad

$$u_t = \frac{\eta}{\sqrt{G_t + \epsilon}} \nabla_\theta \mathcal{L}(\theta_t)$$

$$G_t = G_{t-1} + g_t^2$$

Adagrad основан на той же идее: большая вариативность градиента должна уменьшать скорость обучения и наоборот.
Теперь, если градиенты становятся очень большими, то скорость обучения в соответствующем направлении быстро затухает.

g_t^2 всегда больше 0!
Как улучшить?

RMSProp

$$u_t = \frac{\eta}{\sqrt{G_t + \epsilon}} \nabla_\theta \mathcal{L}(\theta_t)$$

$$G_t = \rho G_{t-1} + (1 - \rho) g_t^2$$

RMSProp добавляет очевидное улучшение. Теперь G_t это не просто сумма квадратов, а экспоненциальное среднее.
Несмотря на существенную популярность этого алгоритма, он никогда не был опубликован. Хинтон просто рассказал о нем в одной из своих лекций на курсере☺

Adadelta

$$u_t = \frac{\eta}{\sqrt{G_t + \epsilon}} \nabla_\theta \mathcal{L}(\theta_t)$$

$$G_t = \rho G_{t-1} + (1 - \rho) g_t^2$$

Adadelta добавляет еще одно небольшое «улучшение» к **Adagrad**. Раз уж мы уже рассуждаем в физических терминах, то можно обратить внимание на «размерности». **Какова размерность u_t?**

Adadelta

$$u_t = \frac{\sqrt{U_{t-1} + \epsilon}}{\sqrt{G_t + \epsilon}} \nabla_\theta \mathcal{L}(\theta_t)$$

$$U_t = \pi U_{t-1} + (1 - \pi) u_t^2$$

$$G_t = \rho G_{t-1} + (1 - \rho) g_t^2$$

Adadelta добавляет еще одно небольшое «улучшение» к **Adagrad**. Раз уж мы уже рассуждаем в физических терминах, то можно обратить внимание на «размерности». **Какова размерность u_t?** Если наши параметры имеют размерность, то шаг который мы делаем — нет, а это странно.

Adam

$$u_t = \frac{\eta}{\sqrt{G_t + \epsilon}} M_t$$

$$M_t = \beta_1 M_{t-1} + (1 - \beta_1) g_t$$
$$G_t = \beta_2 G_{t-1} + (1 - \beta_2) g_t^2$$

И, наконец, самый популярный на текущий момент метод оптимизации — **Adam**. В нем шаг градиентного спуска, так же как и в RMSPropделится на экспоненциальное среднее квадратов, но сам шаг вычисляется как экспоненциальное среднее градиентов.

План на сегодня

1. Скорость
2. Инерция
3. Адаптация
4. **Практика**

Спасибо
за внимание!