Оптимизатор	Формулы
Gradient Descent (GD)	$\theta_{t+1} = \theta_t - \alpha \cdot \nabla_\theta J(\theta_t)$
Stochastic Gradient Descent (SGD)	$\theta_{t+1} = \theta_t - \alpha \cdot \nabla_\theta J(\theta_t, sample)$
Mini-Batch GD	$\theta_{t+1} = \theta_t - \alpha \cdot \nabla_\theta J(\theta_t, Nsamples)$
SGD + Momentum	$v_{t+1} = \gamma \cdot v_t + \eta \cdot \nabla_\theta J(\theta_t)$ $\theta_{t+1} = \theta_t - v_{t+1}$
NAG (Nesterov Accelerated Gradient)	$v_{t+1} = \gamma \cdot v_t + \eta \cdot \nabla_\theta J(\theta_t - v_t)$ $\theta_{t+1} = \theta_t - v_{t+1}$
Adagrad	$g_t = \nabla_\theta J(\theta_t)$ $G_t = G_{t-1} + g_t^2$ $\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{G_t + \varepsilon}} g_t$ $\varepsilon \ll 1$
RMSprop	$g_t = \nabla_\theta J(\theta_t)$ $G_t = \gamma G_{t-1} + (1-\gamma) g_t^2$ $\theta_{t+1} = \theta_t - \frac{\eta}{G_t + \varepsilon_t} g_t$
Adadelta	$g_t = \nabla_\theta J(\theta_t)$ $E\left[g^2\right]_t = \gamma E\left[g^2\right]_{t-1} + (1-\gamma) g_t^2$ $RMS\left[g^2\right]_t = \sqrt{E\left[g^2\right]_t + \varepsilon}$ $\theta_{t+1} = \theta_t - \frac{RMS[\Delta\theta]_{t-1}}{RMS[g^2]_t} g_t$
Adam	$g_t = \nabla_\theta J(\theta_t)$ $m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t$ $\widehat{m}_t = \frac{m_t}{1-\beta_1^t}$ $v_t = \beta_2 v_{t-1} + (1-\beta_2) g_t^2$ $\widehat{v}_t = \frac{v_t}{1-\beta_2^t}$ $\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\widehat{v}_t + \varepsilon}} \widehat{m}_t$

Оптимизатор	Преимущества	Недостатки
Gradient Descent (GD)	Поскольку все данные берутся за один раз, они достигают глобальных минимумов без какого-либо шума, но подходят только для небольших наборов данных.	- Вычисление происходит очень медленно, т.к. используются все данные; - Может сходится к глобальному минимуму для выпуклых функций и перейти к локальному минимуму для невыпуклых функций.
Stochastic Gradient Descent (SGD)	- Быстрее GD; - Новые samples могут быть добавлены по мере поступления новых данных.	- Большие колебания функции потерь из-за частого изменения данных.
Mini-Batch GD	Может в полной мере использовать матричные операции, которые высоко оптимизированы в библиотеке глубокого обучения для более эффективных вычислений градиента.	- Уменьшение скорость обучения может привести к замедлению схождения; - Большая скорость обучения может привести к флуктуации скорости потерь.
SGD + Momentum	- Повышенная стабильность - Обучение происходит быстрее; - Возможность избавиться от локальной оптимизации.	- Долгое обучение при малом угле наклона поверхности функции потерь.
NAG (Nesterov Accelerated Gradient)	Более надежный (лучше сходимость, чем SGD + Momentum)	- Высокая вычислительная сложность
Adagrad	Устраняет необходимость ручной настройки скорости обучения	- Продолжительное снижение скорости обучения, что приводит к минимальной скороски обучения в конце обучения.
RMSprop	Замедляет затухание скорости обучения	- Подбор начальной скорости обучения.
Adadelta	Не требуется задавать скорость обучения	- Риск возникновения колебаний в районе локальных минимумов.
Adam	- Коррекция смещения - Границы размера шага ограничены	- Колебание скорости обучения => Сложность схождения в конце обучения.