

Адаптивные стохастические методы обучения нейронных сетей

Балинова Виктория Максимовна, гр.20.Б04-мм

Санкт-Петербургский государственный университет
Прикладная математика и информатика
Вычислительная стохастика и статистическое моделирование

Научный руководитель: к.ф.-м.н., доцент Шпилев П.В.
Рецензент: лектор Кардиффского университета Пепелышев А. Н.

Санкт-Петербург
2024г.

Введение: постановка задачи

- Актуальная задача - разработка широкоприменимых и эффективных методов обучения нейронных сетей.
- Адаптивные стохастические градиентные методы сильно зависят от выпуклости оптимизируемой функции.
- Альтернативой являются метаэвристические методы, не требующие информации о структуре оптимизируемой функции.

Задача: Разработать быстрый и эффективный адаптивный стохастический алгоритм на основе метаэвристического алгоритма.

Постановка задачи обучения нейронной сети

Определение

Искусственная нейронная сеть (ИНС) — это суперпозиция функций $A(\mathbf{X}, \mathbf{W})$, отображающая исходное пространство данных X^n в множество ответов Y^n . Состоит из входного и выходного слоя и скрытых слоев с нейронами.

$$A(\mathbf{X}, \mathbf{W}) = \varphi_K(\mathbf{W}_K \cdot \dots \cdot \varphi_2(\mathbf{W}_2(\varphi_1(\mathbf{W}_1\mathbf{X}) \dots)),$$

где \mathbf{X} — матрица признаков $m \times n$, \mathbf{W} — матрица весов $n \times d_k$, K — количество скрытых слоев, d_k — количество нейронов в слое K , φ_i — некоторые преобразования, $i = 1, \dots, K$.

Обучение ИНС сводится к оптимизации функции потерь:

$$L(A(\mathbf{X}, \mathbf{W}), \mathbf{y}) \rightarrow \min_{\mathbf{W}}.$$

Адаптивные стохастические градиентные методы

- **Adaptive Subgradient (AdaGrad)** (J. Duchi, 2011)
Использует суммированные квадраты прошлых градиентов.
- **Root Mean Square Propagation (RMSProp)** (G. Hinton, 2012)
Использует скользящее среднее квадратов градиентов вместо их суммирования.
- **Adaptive Moment Estimation (Adam)** (Diederik P. Kingma, Jimmy Lei Ba, 2015)
Используем скользящее среднее как градиентов, так и квадратов градиентов, также включает механизмы для коррекции смещений.

Алгоритм Particle Swarm Optimization (PSO)

PSO (Kennedy, J., Eberhart, R. C., 1995):

- 1 Инициализация позиций x_i^0 и скоростей частиц v_i^0 , лучших решение $p_{0\omega_i}$ и g^0 .
- 2 Обновление позиций и скоростей:

$$v_i^{t+1} = \alpha v_i^t + \beta r_1 (p_i^t - x_i^t) + \gamma r_2 (g^t - x_i^t),$$
$$x_i^{t+1} = x_i^t + v_i^{t+1},$$

где α - инерционный вес, β — когнитивный вес, γ - социальный вес, r_1, r_2 - случайные числа в диапазоне $[0, 1]$.

- 3 Обновление глобальных и личных решений:

$$p_i^t = \min_{k=0,1,\dots,t} \{f(x_i^k), \}$$
$$g^t = \min_{k=1,\dots,N} \{f(p_i^t)\},$$

где f — оптимизируемая функция.

Проблема: PSO очень медленный и не изменяет шаг обучения.

Решение: Адаптивное изменение APSO

Методы улучшения сходимости

- Синусоидальное преобразование для инерции (S. Shao, 2020)

$$\alpha(t) = \alpha_{\max} - t \cdot \frac{\alpha_{\max} - \alpha_{\min}}{T} \cdot \sin \frac{t \cdot \pi}{2T}.$$

Позволяет расширить пространство поиска решений в начале и сужает его в конце.

- Адаптивное изменение когнитивных и социальных весов скорости (H. Feng, 2021)

$$\beta(t) = \beta_{\min} + (\beta_{\max} - \beta_{\min}) \cdot \left(1 - \frac{t}{T}\right), \quad (1)$$

$$\gamma(t) = \gamma_{\min} + (\gamma_{\max} - \gamma_{\min}) \cdot \frac{t}{T}, \quad (2)$$

где β — когнитивный вес, γ — социальный вес, t — текущая итерация, T — максимальное количество итераций.

Методы улучшения сходимости: продолжение

- Добавление гауссовского шума в расчет лучшей глобальной позиции (A. Sarangi, 2019)

$$G^t = g^t + \mathcal{N}(\alpha, \sigma^2), \quad (3)$$

где G^t — мутированная глобально лучшая позиция на итерации t , g^t — текущая глобально лучшая позиция на итерации t , $\mathcal{N}(\alpha, \sigma^2)$ — случайное значение из нормального распределения со средним $\alpha \in \text{rand}(0, 1)$ и дисперсией σ^2 .

- Отказ от расчета скорости (K. Ang, 2020)

$$x_i(t+1) = x_i(t) + \beta r_1(p_{best,i} - x_i(t)) + \gamma r_2(g_{best} - x_i(t)), \quad (4)$$

где $x_i(t)$ — позиция частицы i на итерации t , $p_{best,i}$ — лучшая позиция, найденная частицей i , g_{best} — глобально лучшая позиция, найденная всеми частицами, β и γ — коэффициенты обучения, r_1 и r_2 — случайные числа в интервале $[0, 1]$.

Задача многоклассовой классификации изображений MNIST

Задачу реализуем на пакетном набор данных MNIST, который содержит 70 тысяч изображений размером 28x28 пикселей. Производим классификацию цифр свёрточной нейронной сетью на 10 типов.

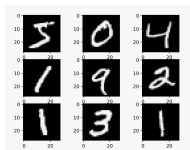


Рис.: Данные MNIST

Нейронная сеть содержит 2 сверточных слоя, функция активации ReLu пулинг слой и 2 полносвязных линейных слоя, функции потерь Cross Entropy. Общее число параметров данной сети — 47658.

Результаты для MNIST

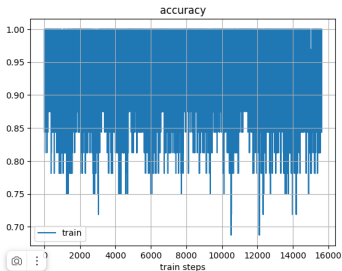


Рис.: Изменение точности предсказаний на данных MNIST, обученной с помощью Adam.

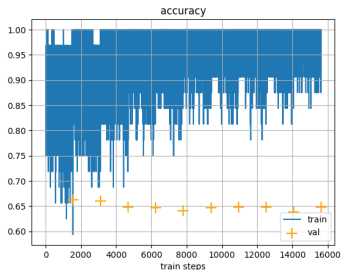


Рис.: Изменене точности предсказаний на данных MNIST, обученной с помощью предложенного алгоритма.

Задача многоклассовой классификации изображений CIFAR-10

Задачу реализуем на пакетном набор данных CIFAR-10, который содержит 60 тысяч цветных изображений размером 32×32 пикселей. Производим классификацию цифр свёрточной нейронной сетью архитектуры VGG на 10 типов.



Рис.: Данные CIFAR-10

Общее число оптимизируемых параметров — 528 474.

Результаты для CIFAR-10

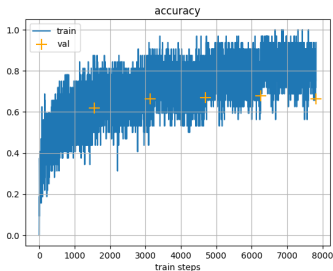


Рис.: Изменение точности предсказаний на данных CIFAR-10, обученной с помощью Adam.

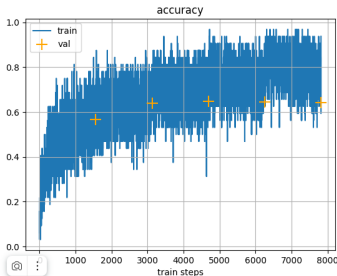


Рис.: Изменене точности предсказаний на данных CIFAR-10, обученной с помощью предложенного алгоритма.

Приведем результаты в виде таблицы, где указана точность каждого метода при установленных значениях параметра:

Таблица: Точность предсказаний нейронной сетей для двум наборов данных.

	Точность предсказания для 5 эпох	
Метод оптимизации	MNIST	CIFAR-10
Adam	0,978	0,803
Предложенный метод	0,986	0,815

Таким образом, были выполнены следующие задачи:

- Предложен собственный параллельный адаптивный стохастический алгоритм оптимизации.
- Эффективность созданного алгоритма была изучена на задачах многоклассовой классификации изображений с помощью сверточных нейронных сетей на классических наборах данных MNIST и CIFAR-10.
- Программная реализация алгоритма написана на Python и выложена в открытый доступ на DOI: [10.5281/zenodo.11369999](https://doi.org/10.5281/zenodo.11369999).