ИЗУЧЕНИЕ АЛГОРИТМА BACKPROPAGATION НА ОСНОВЕ АНАЛИЗА ГРАДИЕНТНОГО ВЕКТОРА

Шен Хаоюй, Се Хайхао

Студент, студент

Факультет ФКИ МГУ имени М. В. Ломоносова, Москва, Россия

E-mail: haoyuishen@yandex.ru, hajhaose@gmail.com

***Научный руководитель** — Смолин Владимир Сергеевич*

В последние годы нейронные сети достигли значительных успехов, но их развитие сталкивается с рядом проблем. Основной метод обучения — Backpropagation (BP) — без дополнительных алгоритмов (регуляризация, SGD, BatchNorm, Adam и др.) не позволяет добиться приемлемых результатов. Ряд исследований направлены на изучение свойств BP и методы повышения его эффективности [2,3]

В данном исследовании уделяется внимание к ключевому вектору ошибок и предлагается подход к вводу дополнительного алгоритма, формулируется перспективное направление оптимизации, и продемонстрируется его эффективность в экспериментах аппроксимации функций.

Формализуем описание (1) для полностью связанного слоя k-ого (если он имеет смещение, то добавим нулевый элемент $o_0^k = 1$ к $\mathbf{O}^k$). В теоретическом анализе есть ряд примеров, в которых сверточный слой приравнивают к виду полностью связанного слоя [1,3].

$$\mathbf{A}^k = \mathbf{W}^k \mathbf{O}^{k-1}; \quad \mathbf{O}^k = \varphi(\mathbf{A}^k) \tag{1}$$

Обозначим градиент $\boldsymbol{\delta}^k = \left\{ \delta_i^k = \frac{\partial L}{\partial a_i^k} \right\}$ – вектор ошибок k-ого слоя (где L – функция потери). На основе теоретического анализа, в котором один и тот же вход подается дважды, см. [4], мы обнаружили, что $\boldsymbol{\delta}^k$ непосредственно и независимо управляет перемещением $\mathbf{A}^k$ что позволяет открыть широкие возможности по применению алгоритмов к методу BP для оптимизации свойств нейросети

По формуле (2), в скрытом слое, $\Delta\mathbf{A}^k$ состоит не только из $\boldsymbol{\Delta}_{\mathbf{W}}^k$, строго соответствующий градиенту $\boldsymbol{\delta}^k$, но и влияния (обозначим $\boldsymbol{\Delta}_{\mathbf{O}}^k$) обучения ранее расположенных параметров. Такое влияние может быть компенсировано модификацией $\boldsymbol{\delta}^k$ без значительного увеличения сложности алгоритма.

$$\Delta\mathbf{A}^k \approx \mathbf{W}^k \Delta\mathbf{O}^{k-1} + \Delta\mathbf{W}^k \mathbf{O}^{k-1} = \boldsymbol{\Delta}_{\mathbf{O}}^k + \boldsymbol{\Delta}_{\mathbf{W}}^k \tag{2}$$

Задача аппроксимации функций важна в таких областях, как управление и робототехника. В многомерных сценариях традиционные методы теряют эффективность, а нейросетевый подход предлагает новые решения (напр. [5]). С другой стороны, хотя большинство приложений, по сути, представляют собой аппроксимацию некоторой функции, сама эта функция априори неизвестна. Поэтому моделирование проведено в рамках задач аппроксимации для лучшей интерпретации принципов работы нейронных сетей и выявления их фундаментальных проблем.

Эксперименты показали, что при коэффициенте компенсации около 0.2, модели с функцией активации ReLU и Tanh демонстрируют улучшение, особенно модели с Tanh — средняя ошибка при сходимости составляет примерно 1/10 от обычных моделей. Такая компенсация также снижает риск стагнации обучения. Мы полагаем, что полученные результаты могут быть обусловлены тем, что компенсация способствует реализации взвешенного коллективного принятия решений по глубине в процессе градиентного спуска.

Развиваемый нами подход с компенсацией влияния дополнительных к BP алгоритмов изменением $\boldsymbol{\delta}^k$ позволяет применять другие идеи оптимизации, например, метод ортогонализации векторов активации (в отличие от ортогонализации весов [3]), уменьшающий взаимодействие между учебными примерами. Этот метод теоретически позволяет повысить скорость и точность обучения, а также способствовать развитию разделения знаний и метаобучения.

Литература

1. Bjorck J., et al. Understanding Batch Normalization // arXiv:1806.02375 [cs.LG], 2018. URL: `https://doi.org/10.48550/arXiv.1806.02375`
2. Choromanska A., et al. The Loss Surfaces of Multilayer Networks // arXiv:1412.0233 [cs.LG], 2014. URL: `https://doi.org/10.48550/arXiv.1412.0233`
3. Li S., et al. Orthogonal Deep Neural Networks // IEEE Trans. Pattern Anal. Mach. Intell., 43(4), 1352–1368, 2021. DOI: `10.1109/TPAMI.2019.2948352`
4. Podoprosvetov A., et al. Vector Analysis of Deep Neural Network Training Process // DeLTA 2024 (CCIS, Vol. 2171). Springer, Cham, 2024. P. 219–237. DOI: `10.1007/978-3-031-66694-0_14`
5. Wang Y., et al. Multi-stage neural networks: Function approximator of machine precision // J. Comput. Phys., 504, 112865, 2024. DOI: `10.1016/j.jcp.2024.112865`