ПОДХОД К АДАПТИВНОМУ РАСПРЕДЕЛЕНИЮ РЕСУРСОВ НЕЙРОСЕТЕЙ НА ОСНОВЕ ОШИБКИ АППРОКСИМАЦИИ

Х. Се[1], В.С. Смолин[2], **Х. Шэн**[1]

[1]*МГУ имени М.В. Ломоносова, г. Москва*
[2]*ИПМ им. М.В. Келдыша РАН, г. Москва*

hajhaose@gmail.com, smolin@keldysh.ru, haoyuishen@yandex.ru

Введение

В последние годы нейронные сети достигли значительных успехов в различных областях – от простых задач регрессии и классификации до генеративных моделей. Изучение основного метода обучения – обратного распространения ошибки (ВР) позволило не только далеко продвинуться в решении самых разных задач ИИ, но и выявило ряд проблем, встречающихся на практике. В связи с этим было проведено множество исследований, направленных на изучение причин этих проблем [1–3], но пока не все недостатки алгоритма ВР удалось полностью устранить.

В данном исследовании представлен теоретический и экспериментальный (по аппроксимации функции) анализ ключевого фактора снижения эффективности нейросетей и предлагается новый метод настройки параметров, сочетающий градиентный спуск и картирование. Первые полученные результаты показывают перспективность данного метода для разработки улучшенных алгоритмов обучения.

Эффективность использования нейронных ресурсов

Исследования показывают, что увеличение размера и глубины нейронной сети не всегда повышает её производительность [4, 5], указывая на неэффективное использование части нейронов при обучении методом обратного распространения ошибки (ВР). Эксперименты выявили две основные проблемы: а) многие нейроны не проявляют нелинейных свойств на участках с высокой ошибкой, застревая в локальных минимумах и б) часть нейронов работает только в линейном режиме.

Хотя существуют методы [6] для решения второй проблемы, вопрос оптимального распределения нейронных ресурсов изучен слабо. В настоящее время активация нейронов зависит от градиентного спуска, что недостаточно эффективно. Предлагается метода для улучшения распределения активации нейронов.

Введем следующие обозначения: $\vec{\boldsymbol{A}}^{\boldsymbol{k}} = \{\boldsymbol{a}_{\boldsymbol{i}}^{\boldsymbol{k}}\}$ и $\vec{\boldsymbol{O}}^{\boldsymbol{k}} = \{\boldsymbol{o}_{\boldsymbol{i}}^{\boldsymbol{k}}\}$ – входной и выходной вектора активации слоя $\boldsymbol{k}$, где $\boldsymbol{a}_{\boldsymbol{i}}^{\boldsymbol{k}}$ – его i-ой компонент (активация $\boldsymbol{i}$-того нейрона), $\vec{\boldsymbol{W}}_{\boldsymbol{i}}^{\boldsymbol{k}}$ – вектор входных весов связей i-того нейрона $\boldsymbol{k}$-ого слоя $\boldsymbol{b}_{\boldsymbol{i}}^{\boldsymbol{k}}$ – смещение этого нейрона, а $\boldsymbol{f}$ – нелинейная функция, тогда компоненты векторов $\vec{\boldsymbol{A}}^{\boldsymbol{k}}$ и $\vec{\boldsymbol{O}}^{\boldsymbol{k}}$ зависят от $\vec{\boldsymbol{W}}_{\boldsymbol{i}}^{\boldsymbol{k}}$ и выходной активности $\vec{\boldsymbol{O}}^{\boldsymbol{k-1}}$ предыдущего слоя следующим образом:

$$\boldsymbol{a}_{\boldsymbol{i}}^{\boldsymbol{k}} = \vec{\boldsymbol{W}}_{\boldsymbol{i}}^{\boldsymbol{k}}\vec{\boldsymbol{O}}^{\boldsymbol{k-1}} + \boldsymbol{b}_{\boldsymbol{i}}^{\boldsymbol{k}}\,;\boldsymbol{o}_{\boldsymbol{i}}^{\boldsymbol{k}} = \boldsymbol{f}\left(\boldsymbol{a}_{\boldsymbol{i}}^{\boldsymbol{k}}\right) \quad (1)$$

Развивая идею раздельного обучения весов и смещений нейронов предложенную в [3], мы обучаем веса методом BP, а для корректировки смещений используем метод картирования, который сдвигает положение областей проявления нелинейных свойств нейронов так, чтобы эти области распределялись по пространству векторов $\vec{O}^{k-1}$ равномерно по расстояниям и уровням ошибки аппроксимирования.

Картирование

Классическое картирование (SOM, по Кохонену [7]) – это разделение векторного пространства на областей, каждая из которых представлена точкой, перемещаемой в процессе картирования в центр определяемой ей области (форма и положение которой при этом тоже изменяется). В случае глубоких нейросетей в пространстве O^{k-1} (состояний векторов $\vec{O}^{k-1}$) каждому элементу i слоя k сопоставляется граница S_i^k размерности $N^{k-1}-1$, где N^{k-1} – количество элементов слоя $k-1$. Граница S_i^k делит пространство O^{k-1} на две части: в одной части $a_i^k > 0$, а в другой $a_i^k < 0$. Граница S_i^k содержит все вектора $\vec{O}^{k-1}$, такие, что $a_i^k = 0$:

$$S_i^k = \left\{\vec{O}^{k-1} \middle| a_i^k = \vec{W}_i^k \vec{O}^{k-1} + b_i^k = 0\right\}. \quad (2)$$

Любая прямая, проведённая в S_i^k будет перпендикулярна вектору $\vec{W}_i^k$ по построению. Границы S_i^k задают форму областей C_i^k в пространстве O^{k-1} (по одной области на каждый элемент слоя k). Свойством, определяющим принадлежность векторов $\vec{O}^{k-1}$ области C_i^k является минимальность длины перпендикуляра (который считается как $|d_i^k| = |a_i^k / \|\vec{W}_i^k\|\|$) опущенного из вектора $\vec{O}^{k-1}$ на границу S_i^k:

$$C_i^k = \{\vec{O}^{k-1} | \, |d_i^k| < |d_j^k|, \ \forall j \neq i\}. \quad (3)$$

Так же, как и при классическом нейрокартировании, границы S_i^k изначально могут не находится в середине определяемых ими областей C_i^k , но в процессе картирования перемещаются в данное положение. Затем, исходя из текущего отображения сети и функции потерь, двигают положение S_i^k, изменяя b_i^k по закону

$$\Delta b_i^k = -\gamma d_i^k L, \quad (4)$$

где L – положительно определённая функция потерь, в такое состояние, при котором достигается нулевое значение суммы приращений Δb_i^k по батчу для каждой области C_i^k:

$$\sum_{q=1}^{batch} \Delta b_{iq}^k \cong 0. \quad (5)$$

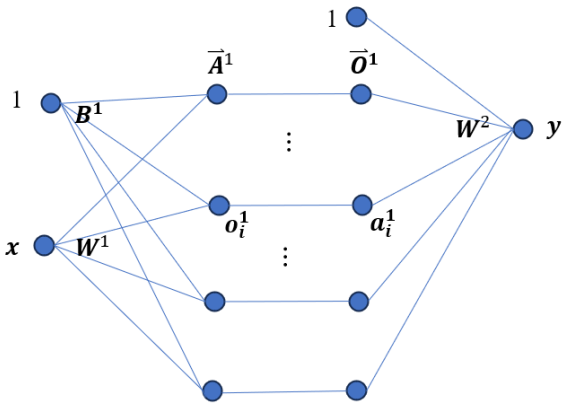

Рис. 1. Нейронная сеть с одним скрытым слоем.

Одномерная модель и эксперимент с пороговым картированием

Реализация этого метода для одномерных входных данных проста и уже достигла определенного успеха в эксперименте по аппроксимации одномерной функции. Мы построили модель с одним скрытым слоем (см. рис. 1), где в качестве нелинейной функции использовали ReLU или SoftPlus.

Инициализация: матрица входных связей скрытого слоя, состоящая из двух вектор-строк $\vec{W}^1$ и $\vec{B}^1$ инициализируется случайными значениями из интервала ±1 и вектор $\vec{W}^1$для ReLU остаётся необучаемым. Компоненты $\vec{B}^1$ (b_i^1) изменяются по закону (4), что постепенно приводит к выполнению условия (5).

Обновление $\vec{B}^1$ происходит не на каждом шаге моделирования, а после обучения сети всем эталонным преобразованием очередного батча.

Аналогично, по окончанию обучения батчу, происходит и обновление $\vec{W}^2$, но эти веса связей обучаются по методу ВР. В случае использования функции SoftPlus, компоненты вектора $\vec{W}^1$ тоже изменяются в конце батча по методике ВР.

Используя вышеописанную модель с функцией активации ReLU и SoftPlus в задаче аппроксимации одномерной функции, точность выросла на 1-2 порядка, сходимость значительно улучшилась, а также learning_rate стал более надежным. При этом нужно найти баланс между значениями learning_rate, в методе ВР и картировании и размерами батчей. Эксперименты показали, что корректировки смещений в скрытых слоях по закону (4) в процессе обучения (по сравнению с их отсутствием) дают значительный эффект – за 6000 итераций было сделано 30 корректировок, и ошибка уменьшилась более, чем вдвое.

SoftPlus и аппроксимация гладкой функции

Функция SoftPlus имеет ряд преимуществ. Она гладка во всех точках, что делает её более устойчивой в процессе градиентной оптимизации, чем ReLU [8]. При этом, SoftPlus сохраняет ключевое преимущество ReLU: отображает входы в диапазон от 0 до бесконечности. Эти характеристики обеспечивают её больший потенциал для аппроксимации гладких функций.

В вышеописанном эксперименте, градиентное обновление W^1 требуется при использовании SoftPlus для лучшей аппроксимации высокочастотных колебаний функции на заданном интервале. В отличии от ReLU, у которого изменение наклона мгновенно происходит, скорость этого процесса в SoftPlus регулируется параметром $\boldsymbol{W^1}$. Кроме того, встроенная константа в SoftPlus тоже позволяет эффективно решить эту проблему.

Направления дальнейшего развития исследования

Полученные результаты для одномерных задач и однослойных нейронных сетей не означают аналогичного улучшения в многомерном случае – это составляет основное направление дальнейших экспериментов.

В многомерных задачах используем формулу $\left|\boldsymbol{d_i^k}\right| = \left|\boldsymbol{a_i^k} / \left\|\overrightarrow{\boldsymbol{W_i^k}}\right\|\right|$ расстояния от вектора $\overrightarrow{\boldsymbol{O^{k-1}}}$ до гиперплоскости $\boldsymbol{S_i^k}$ для корректировки b_i^k для смещения её расположения к центру области $\boldsymbol{C_i^k}$. В многослойных моделях градиентное обновление применяется ко всем весовым матрицам, а картирование смещения по закону (4) – ко всем слоям, кроме последнего.

Список литературы:

1. Choromanska A. [et al.]. The Loss Surfaces of Multilayer Networks // 2014. arXiv:1412.0233. https://doi.org/10.48550/arXiv.1412.0233
2. Jakub C., Nica M. Depth Degeneracy in Neural Networks: Vanishing Angles in Fully Connected ReLU Networks on Initialization // Journal of Machine Learning Research. 2024. Vol. 25. Issue 239. Pp. 1–45.
3. Sokolov S., Smolin V., Podoprosvetov A. Separate Adjustment of Linear and Nonlinear Parameters in Neural Network Training // OpenReview. 2025.
4. Duvenaud D., Rippel O., Adams R., Ghahramani Z. Avoiding pathologies in very deep networks // Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics. 2014. Vol. 33. Pp. 202–210.
5. Otache M.Y. [et al.]. Effects of Model Structural Complexity and Data Pre-Processing on Artificial Neural Network (ANN) Forecast Performance for Hydrological Process Modelling // Open Journal of Modern Hydrology. 2021. Vol. 11. Pp. 1–18. https://doi.org/10.4236/ojmh.2021.111001
6. Awais M., Iqbal M.T.B., Bae S.H. Revisiting Internal Covariate Shift for Batch Normalization // IEEE Transactions on Neural Networks and Learning Systems. 2020. Vol. 32. Issue 11. Pp. 5082–5092. https://doi.org/10.1109/tnnls.2020.3026784
7. Kohonen, T. Self-Organized Formation of Topologically Correct Feature Maps // Biological Cybernetics. 1982. Vol 43. Issue 1. Pp. 59–69. https://doi.org/10.1007/bf00337288
8. Hammad M.M. Deep Learning Activation Functions: Fixed-Shape, Parametric, Adaptive, Stochastic, Miscellaneous, Non-Standard, Ensemble // CoRR. 2024. Vol. abs/2407.11090.