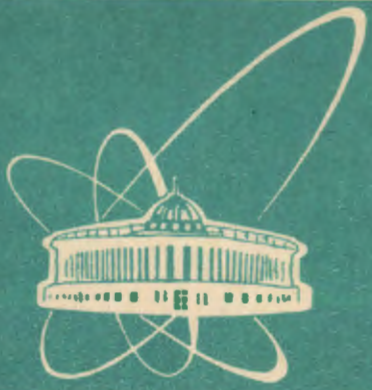

Объединенный
институт
ядерных
исследований
дубна

Д11-93-317

В.В.Иванов, И.В.Пузынин, Б.Пурэвдорж

АЛГОРИТМ ОБУЧЕНИЯ НЕЙРОННОЙ СЕТИ НА ОСНОВЕ МЕТОДА НЬЮТОНА

Направлено в журнал «Computer Physics Communications»

1993

Введение

В настоящее время искусственные нейронные сети находят широкое применение в методике физического эксперимента [1, 2]. Они используются как для построения интеллектуальных триггеров высокого уровня, позволяющих проводить эффективную дискриминацию фоновых событий в реальном времени эксперимента, так и в задачах обработки экспериментальных данных, таких как распознавание траекторий частиц, восстановление масс, идентификация первичных и/или вторичных вершин и т.д.

Наибольшее распространение получили многослойные **feed forward** сети [4, 5]. Настройка таких сетей на конкретную задачу реализуется путем их обучения, для чего, как правило, используется алгоритм **back-propagation** [3, 6], представляющий собой градиентный метод. Исследования этого алгоритма (см. ниже) показали, что с его помощью не удается добиться хорошего качества обучения сети, которое особенно важно для таких задач экспериментальной физики, где маловероятные события выделяются на подавляющем фоне от конкурирующих процессов.

С математической точки зрения процедура обучения **feed forward** сети сводится к задаче минимизации квадратичного функционала [3]. В настоящей работе исследован алгоритм обучения на основе метода Ньютона, проведено его сравнение с методами первого порядка и даются рекомендации по их использованию.

1 Постановка задачи

1.1 Архитектура нейронной сети

Feed forward сеть состоит из нескольких слоев: входного, на который подаются анализируемые данные, одного или нескольких скрытых и выходного, с которого снимается результат.

Выходному сигналу отвечает функция $\vec{y}$:

$$y_i = f(a_i), \qquad a_i = \sum_j \omega_{ij} h_j + \theta_i,$$

$$h_j = \begin{cases} f(a_j) & \text{для скрытых слоев,} \\ x_j & \text{для входного слоя,} \end{cases}$$

$$f(x) = g(x/T).$$

Здесь x_j – данные, подаваемые на вход сети, ω_{ij} – веса связей нейронов, θ_i – пороги, T – "температура", g – переходная функция. Нами использовалась переходная функция вида

$$g(x) = \frac{1}{2}(1 + tanh(x)).$$

Задача обучения сети состоит в нахождении весов ω, минимизирующих функционал энергии

$$E = \frac{1}{2}\sum_p \left(\vec{y}^{(p)} - \vec{t}^{(p)}\right)^2,$$

где $p = 1, \ldots, N_{train}$, N_{train} – число обучающих образцов, $\vec{t}$ – целевое значение выходного сигнала сети.

Сеть программировалась на IBM PC AT386/486 в среде NDP [7] с использованием пакета JETNET 2.0, разработанного в Университете Лунда [6].

1.2 Задача классификации

Нейронная сеть используется для классификации событий, представляющих собой случайные выборки из двух перекрывающихся распределений с плотностями f_1 и f_2. Возьмем сеть с числом входных нейронов, равным размерности события, и с одним выходным нейроном. В процессе обучения на вход сети подается выборка, содержащая равные количества событий обоих сортов. Зададим величину целевого сигнала равной 0 для событий из распределения f_1 и 1 для событий из распределения f_2.

Вводя обозначение $y = F(\omega, x)$ и переходя к непрерывному пределу, получим

$$E = \int \left((F(\omega,x))^2 f_1(x) + (1 - F(\omega,x))^2 f_2(x)\right) dx.$$

Обозначим через

$$opt(x) = \frac{f_2(x)}{f_1(x) + f_2(x)}, \qquad D = \{x | f(x) = f_1(x) + f_2(x) \neq 0\}.$$

Тогда

$$E(opt + \epsilon) = E(opt) + \int \epsilon(x)^2 f(x) dx.$$

Вводя в D норму $y = (\int y^2(x) f(x) dx)^{\frac{1}{2}}$, сводим задачу минимизации E к задаче вычисления функции $y(x)$, ближайшей к opt. Под качеством обучения здесь понимается точность минимизации функционала E. Очевидно, что, повышая качество обучения нейронной сети,можно получить больше информации об анализируемых распределениях.

В настоящей работе в качестве распределений f_1 и f_2 используются многомерные гауссианы.

2 Методы первого порядка

В алгоритме **Back-propagation (BP)**, описанном в [3], процедура вычисления весов состоит в следующем:

1. начальные веса выбираются случайным образом из равномерного распределения на интервале $[-\omega_0, \omega_0]$,
2. поправки к весам определяются из соотношения

$$\Delta\vec{\omega}_n = -\eta \nabla E_n + \alpha \Delta\vec{\omega}_{n-1},$$

$$\frac{\partial E}{\partial \omega_{ij}} = d_i h_j, \qquad d_i = \begin{cases} (y_i - t_i) f'(a_i) & \text{– для выходного слоя,} \\ \sum_k d_k \omega_{ki} f'(a_i) & \text{для скрытых слоев.} \end{cases}$$

Отличие этого алгоритма от градиентного спуска состоит в том, что ∇E_n вычисляется на подмножестве всей выборки, содержащем N образцов, а также в наличии момента α, играющем важную роль при минимизации овражных функций [3].

Эффективность алгоритма BP сильно зависит от параметров сети, для выбора которых сложно сформулировать чёткие критерии. Можно только предложить ряд качественных соображений, приведённых ниже.

Оптимальный размер сети можно подобрать исходя из геометрии задачи. Чем сложнее область, которую должна ограничить сеть [4], тем больший нужен размер. Для решения многих задач [1] требуются небольшие сети, а дополнительное увеличение числа нейронов не улучшает качества обучения.

Параметр N должен удовлетворять условию $N \ll N_{train}$. Эффективность сети повышается, если N образцов для очередного шага выбираются случайным образом. Параметр ω_0 не следует брать близким к нулю, из-за того, что $E(\vec{\omega})$ более овражна в окрестности нуля, и сильно отличным от нуля, так как норма градиента может быть очень мала. Параметр η обычно подбирается эмпирически. Разумно уменьшать η при увеличении размера сети и числа обучающих образцов. По мере приближения к минимуму целесообразно уменьшать η и увеличивать T, что ограничивает колебания выходного сигнала во время обучения. Процедура изменения η не играет при этом существенной роли.

Следует отметить, что функция E обычно не имеет чётко выраженных локальных минимумов, что обеспечивает сходимость метода к решению.

Алгоритм BP переходит в метод градиентного спуска при $N = N_{train}$ и $\alpha = 0$. В этом случае процесс обучения протекает очень медленно. Параметр η приходится выбирать малым, а попытка определять η путём минимизации $E(\eta)$ на интервале $0 < \theta \leq \eta \leq 1$ с помощью подпрограмм E04JAF или E04ABF из пакета NAG [8] не влияет заметно на качество результата. Градиентный спуск может быть целесообразен только в окрестности минимума, так как он характеризуется меньшими колебаниями.

[1] В частности, это относится к большинству задач многомерного анализа данных в области физики высоких энергий.

Хотя алгоритм **BP** достаточно эффективен, следует отметить его недостатки:

- метод не обеспечивает высокой точности минимизации;
- параметры сети часто приходится подбирать эмпирическим путём;
- достаточно сложно установить момент попадания в окрестность минимума;
- норма градиента убывает медленно и, как правило, оказывается сильно отличной от нуля, что приводит к большим колебаниям.

3 Методы второго порядка

3.1 Метод Ньютона

В методе Ньютона поправки к весам вычисляются из выражения

$$\Delta\omega = -\eta(\nabla^2 E)^{-1}\nabla E,$$

где гессиан $\nabla^2 E$ определяется следующим образом

$$\frac{\partial^2 E}{\partial\omega_{ij}\partial\omega_{mn}} = d_i\frac{\partial h_j}{\partial\omega_{mn}} + \frac{\partial d_i}{\partial\omega_{mn}}h_j.$$

Метод оказывается неприменимым для рассматриваемой задачи, так как гессиан знакопеременен, а в окрестности минимума почти вырожден.

Укажем на одну из возможностей вырождения в случае, когда сеть используется для классификации двух распределений, обладающих радиальной симметрией с общим центром.

Заметим, что $F(\vec{\omega},\vec{x}) = F(\vec{\omega}_1\cdot\vec{x},\ldots,\vec{\omega}_n\cdot\vec{x},\vec{\omega}_0)$, где $\vec{\omega}_i$ – подвектор, состоящий из весов, ведущих от входных нейронов к i-ой скрытому нейрону, а $\vec{\omega}_0$ – все остальные веса. Действуя на $\vec{\omega}$ матрицей вращения системы координат U, получим в непрерывном пределе

$$E(\vec{\omega}') = \int F(\vec{\omega}',\vec{x}')d\vec{x}' = \int F(U\vec{\omega}_1,\ldots,U\vec{\omega}_n,\vec{\omega}_0,U\vec{x})d\vec{x} = \int F(\vec{\omega},\vec{x})d\vec{x} = E(\vec{\omega}).$$

Когда угол вращения стремится к нулю, имеем $\vec{\omega}_n \to \vec{\omega}$, $E(\vec{\omega}_n) = E(\vec{\omega})$.

Строго говоря, минимум не является полностью вырожденным, однако матрица $\nabla^2 E$ при приближении к минимуму становится плохо обусловленной. Типичные значения числа обусловленности $|\lambda|_{max}/|\lambda|_{min}$ (λ-собственные значения) составляли 10^3 в точке начального приближения и доходили до $10^5 - 10^6$ (иногда даже до 10^{12}) в окрестности минимума.

Использование регуляризации [10]

$$\Delta\omega = -\eta H^{-1}A^*\nabla E,$$

$$H = A^*A + \alpha\|\nabla E\|^2, \qquad A = \nabla^2 E,$$

обеспечивает сходимость только при больших α ($\alpha \sim 1$) и не даёт лучшего качества, чем градиентный спуск. При этом матрица H оказывается весьма плохо обусловленной, когда регуляризующая добавка приближается к нулю. Это не позволяет обратить матрицу с приемлемой точностью.

3.2 Метод Ньютона с коррекцией гессиана

В тех случаях, когда гессиан не обладает достаточно хорошими свойствами, используется метод

$$\Delta\omega = -\eta H^{-1}\nabla E,$$

где H является положительно определенной и хорошо обусловленной матрицей. При этом в случае, когда $\|(\nabla^2 E)^{-1}\| > A$ в малой окрестности минимума, где A – некоторое максимально приемлемое значение, $H \not\to \nabla^2 E$ при приближении к минимуму.

Матрица H выбиралась в виде [9]

$$H = \nabla^2 E + \mu I,$$

где μ таково, что собственные значения $H \geq \delta > 0$. Хорошие результаты дает выбор $\delta = 1$. При этом H является сжимающим оператором.

Шаг η определялся с помощью одномерной минимизации $E(\eta)$ (см. раздел 2), что не оказывает заметного влияния на скорость обучения, так как основное время затрачивается на вычисление гессиана. Как правило, стабилизация шага в районе 1 свидетельствует о попадании в окрестность минимума.

Для вычисления η удобно использовать алгоритм [11]

$$\eta_{n+1} = \eta_n\frac{\|\nabla E_{n-1}\|}{\|\nabla E_n\|}, \quad \theta \leq \eta_{n+1} \leq 1.$$

где η_0 достаточно мало. Однако это несколько замедляет скорость обучения.

К достоинствам метода можно отнести следующее:

- этот метод обеспечивает качество обучения, практически недостижимое с помощью **BP**; это проявляется в величине ошибки E и в норме градиента;
- как правило, скорость сходимости не сильно зависит от размера сети, количества обучающих образцов и сложности задачи распознавания;
- метод позволяет решать более сложные задачи, в частности, разделять сильно перекрывающиеся распределения, и делать надёжные предсказания по малой выборке;
- параметры обучения подбираются автоматически.

К недостаткам метода следует отнести большие вычислительные затраты. Если для **BP** требуется память порядка n и время порядка $n \times N_{train}$ на каждой эпохе, где n – число весов, то для нового алгоритма необходима память порядка n^2 и время $n^2 \times N_{train}$ (при $n \ll N_{train}$). Отметим, что скорость обучения не играет решающей роли, если сеть используется многократно.

4 Сравнение алгоритмов обучения

Напомним, что **feed-forward** сеть используется для классификации двух распределений. В таблице 1 представлены характеристики различных методов обучения для конфигурации сети 2–6–1 со следующими параметрами: $\omega_0 = 0.5,\ T = 1;$ и для **BP**: $\eta_0 = 0.1,\ \alpha = 0.5,\ N = 10.$

Таблица 1

Характеристики различных алгоритмов обучения **feed forward** сети

Метод	Пример	N_{train}	*Limit*	*Epoch*	E_{min}	$\|\nabla E\|$
BP	1/0.3	1000	76,1	7	0.160*	1.0 – 20
NW				7	0.155*	0.047
GRAD				107	0.161*	0.48
REG				–	0.159*	0.0014
BP	1/0.7	1000	58,5	52	0.235	2.0 – 40
NW				4	0.229	0.13
BP	1/0.8	1000	55,3	–	0.242	4.0 – 40
NW				6	0.238	0.092
BP	1/0.3	100	76,1	19	0.144	0.4 – 4.0
NW				17	0.138	0.065
BP	1/0.5	100	66,1	–	0.182	0.2 – 5.0
NW				25	0.142	0.26
BP	1/0.3	5000	76,1	4	0.164	3.0 – 60
NW				7	0.163	0.10

Здесь введены следующие обозначения: **BP** – **back-propagation**, **GRAD** – чистый градиентный спуск, **NW** – метод Ньютона с коррекцией гессиана, **REG** – метод Ньютона с регуляризацией (применяемый после достижения окрестности минимума с помощью **BP**), Пример – стандартные отклонения разделяемых двумерных гауссианов (математические ожидания взяты равными нулю), *Limit* – байесовский предел распознавания, *Epoch* – число эпох, за которое сеть достигает байесовского предела распознавания и стабилизируется, E_{min} – наименьшее

достигаемое за 100 (* – за 1000) эпох значение функции ошибки, деленное на N_{train}, $\|\nabla E\|$ – норма градиента в конце обучения.

На рис. 1 показаны кривые обучения для **BP** и метода Ньютона.

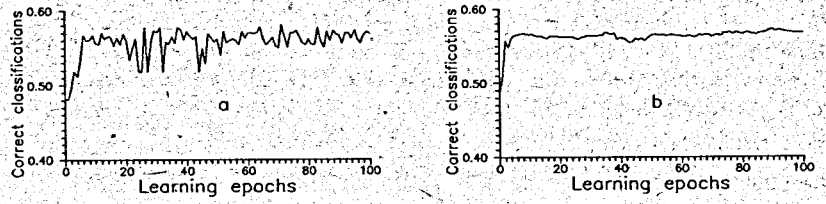

Рис.1. Зависимость доли правильно распознанных событий от номера эпохи обучения для примера 1/0.7 при $N_{train} = 1000$: а) **BP**, б) метод Ньютона.

Заключение

Результаты сравнения различных алгоритмов обучения многослойных **feed forward** сетей можно просуммировать следующим образом:

1. При оптимальном выборе параметров сети алгоритм **BP** позволяет достичь окрестности минимума довольно быстро и с минимальными затратами оперативной памяти.
2. Метод градиентного спуска целесообразно применять только после достижения окрестности минимума.
3. Чистый метод Ньютона неприменим для рассматриваемого класса задач.
4. Метод Ньютона с коррекцией гессиана требует бо́льших затрат вычислительных ресурсов, чем **BP**, однако он обеспечивает лучшее качество обучения и проще в использовании.

Таким образом, можно сделать вывод, что алгоритм **BP** хорош для больших сетей, используемых при распознавания сложных образов, а метод Ньютона с коррекцией гессиана – для небольших сетей, от которых требуется максимальная точность распознавания.

Литература

[1] B.Denby. *"Tutorial on Neural Networks Applications in High Energy Physics: 1992 Perspective"*. In Proc. of the Second International Workshop on *"Software Engineering, Artificial Intelligence and Expert Systems in High Energy Physics"*. January 13-18, 1992 L'Agelaude France-Telecom La Londe-les-Maures (France). New Computing Techniques in Physics Research II, edited by D.Perret-Gallix, World Scientific, 1992, p.287.

[2] C.Peterson. *"Neural Networks in High Energy Physics — Algorithms and Results"*, ibidem, p.327.

[3] D.E.Rumelhart, G.E.Hinton, R.J.Williams. *"Learning Internal Representations by Error Propagation"* in D.E.Rumelhart & J.L.McClelland (Eds.), Parrallel Distributed Processing: Explorations in the Microstructure of Cognition. Vol. 1: Foundations. MIT Press (1986).

[4] R.P.Lippmann. *"An Introduction to Computing with Neural Nets"*, IEEE ASSP Mag.4, 1987, p.4.

[5] B.Humpert. *"A Comparative Study of Neural Network Architectures"*, Comp. Phys. Comm. 58, 1990, p.223.

[6] L.Lönnblad, C.Petterson, T.Rögnvaldsson. *"Pattern Recognition in High Energy Physics with Artificial Neural Networks – JETNET 2.0"*, Comp. Phys. Comm. 70, 1992, p.167.

[7] NDP – FORTRAN *Reference Manual.* Microway Inc., Kingston, Massachusetts.

[8] NAG *Fortran Library Manual.* NAG Ltd., Oxford, 1990.

[9] М.Мину. *Математическое Программирование.* /Пер. с фр. – М.: Наука, 1990.

[10] В.В.Ермаков, Н.Н.Калиткин. *"Оптимальный Шаг и Регуляризация Метода Ньютона"*, ЖВМиМФ, т.21, 2, 1981, с.491.

[11] Т.Жанлав, И.В.Пузынин. *"О Сходимости Итераций на Основе Непрерывного Аналога Метода Ньютона"*, ЖВМиМФ, т.32, 6, 1992, с.846.

Рукопись поступила в издательский отдел
20 августа 1993 года.