Kybernetika

Zdeněk Častorál

Некоторые общие алгоритмы обучения

Kybernetika, Vol. 7 (1971), No. 5, (377)--385

Persistent URL: http://dml.cz/dmlcz/125149

KYBERNETIKA — ТОМ 7 (1971), ВЫПУСК 5

Некоторые общие алгоритмы обучения

Зденек Часторал

Рассматриваются общие алгоритмы обучения, которые позволяют по мере поступления информации асимптотически приближатся к цели обучения.

1. ВВЕДЕНИЕ

В последнее время значительное внимание уделяется адаптивным обучающимся системам [1]. Эти системы по мере поступления информации могут улучшать свое функционирование вплоть до оптимального.

Проблема обучения может быть сформулирована как задача последовательных приближений к некоторым неизвестным параметрам. Для оценки неизвестных параметров могут быть применены вероятностные итеративные алгоритмы.

Обратим сначала внимание на аппроксимирующую функцию. Правила формирования функции строятся так, чтобы по мере увеличения числа предъявляемых в процессе обучения входных векторов аппроксимирующая функция стремилась к некоторой разделяющей функции. В основу положено построение так называемых потенциальных функций. Выбор аппроксимирующих функций зависит от ограничений первого рода, т. е. законов природы выражаемых в виде алгебраических, дифференциальных или интегральных уравнений [2].

Для различных задач обучения часто используется аппроксимирующая функция $\hat{f}_0(\mathbf{x}, \mathbf{c})$ в виде конечной суммы

$$\hat{f}_0(\mathbf{x}, \mathbf{c}) = \sum_{i=1}^{N} c_i \, \varphi_i(\mathbf{x})$$

или

$$\hat{f}_0(\mathbf{x}, \mathbf{c}) = \mathbf{c}^\mathrm{T} \boldsymbol{\varphi}(\mathbf{x}) ,$$

где $\mathbf{c} = (c_1, \ldots, c_N)$ — N-мерный вектор весовых коэффициентов, $\boldsymbol{\varphi}(\mathbf{x})$ — N-

мерный вектор линейно независимых функций от случайного вектора $\mathbf{x} = (x_1, \ldots, x_M)$.

Реализация аппроксимирующих функций осуществляется с помощью весового вектора. Для умножения на весовой коэффициент на входе каждой составляющей входного вектора включается сопротивление переменной величины. Переменное значение сопротивления достигается использованием сопротивления с двигателем, мемисторов, трансфлюкторов и т. п. В последних двух случаях не обеспечивается линейный характер изменения величины сопротивления в заивисимости от управляющего тока. Медленное изменение сопротивления весов во времени вызывает нежелательные замедления в реакциях на входные векторы.

С целью упрощения реализации обучающихся систем выгоднее использовать: аппроксимирующую функцию вида

$$\hat{f}_1(\mathbf{x}, \mathbf{c}) = \prod_{i=1}^{N} (x_i + c_i) + c_0 , \tag{1}$$

или

$$\hat{f}_1(\mathbf{x}, \mathbf{c}) = \prod_{i=1}^{N} [\varphi_i(\mathbf{x}) + c_i] + c_0 , \tag{2}$$

аппроксимирующую функцию вида

$$\begin{aligned} \hat{f}_2(\mathbf{x}, \mathbf{c}) &= \prod_{i=1}^{m_1} [x_i + c_i] + \ldots + \prod_{i=m_{l-1}+1}^{m_l} [x_i + c_i] + c_0 = \\ &= \sum_{j=1}^{l} \prod_{i=m_{j-1}+1}^{m_j} (x_i + c_i) + c_0 , \end{aligned} \tag{3}$$

или

$$\begin{aligned} \hat{f}_2(\mathbf{x}, \mathbf{c}) &= \prod_{i=1}^{m_1} [\varphi_i(\mathbf{x}) + c_i] + \ldots + \prod_{i=m_{l-1}+1}^{m_l} [\varphi_i(\mathbf{x}) + c_i] + c_0 = \\ &= \sum_{j=1}^{l} \prod_{i=m_{j-1}+1}^{m_j} (\varphi_i(\mathbf{x}) + c_i) + c_0 , \\ & m_l = N , \quad \{m_0 = 0;\ j = (1 \ldots l)\} \end{aligned} \tag{4}$$

где c_0 — компензирующая составляющая.

Эти аппроксимации были впервые предложены автором. Их можно назвать „некласическими" или „нелинейными". Детальный анализ этих аппроксимаций и условия для выбора функций $\varphi_i(\mathbf{x})$ будут приведены автором в последующих работах.

2. ВЕРОЯТНОСТНЫЕ ИТЕРАТИВНЫЕ АЛГОРИТМЫ

Как правило, в основу обучающихся систем заложены вероятностные итеративные алгоритмы с помощью которых достигается цель обучения. Цель обучения можно представить как экстремум функционала

$$J(\mathbf{c}) = \int_X Q(\mathbf{x}, \mathbf{c})\, p(\mathbf{x})\, \mathrm{d}\mathbf{x}\,, \tag{5}$$

где $\mathbf{c} = (c_1, \ldots, c_N)$ – вектор неизвестных параметров, $\mathbf{x} = (x_1, \ldots, x_M)$ – вектор входного случайного процесса, плотность распределения которого равна $p(\mathbf{x})$, $Q(\mathbf{x}, \mathbf{c})$ – заданная функция.

Условия экстремума можно записать как

$$\nabla J(\mathbf{c}) = E_x\{\nabla_{\mathbf{c}} Q(\mathbf{x}, \mathbf{c})\} = 0\,.$$

С помощью вероятностных итеративных алгоритмов (на базе стохастических аппроксимаций) можно на каждом шаге n по значениям $\mathbf{x}[n]$ и предыдущего значения $\mathbf{c}[n-1]$ и $\nabla_{\mathbf{c}} Q(\mathbf{x}[n], \mathbf{c}[n-1])$ определить вектор $\mathbf{c}[n]$ так, чтобы этот вектор по мере увеличения информации стремился к такому значению $\mathbf{c}^*$ для которого функционал (5) достигает экстремума.

Дискретный алгоритм обучения может быть представлен в виде разностного уравнения [1]

$$\mathbf{c}[n] = \mathbf{c}[n-1] - \Gamma[n]\, \nabla J(\mathbf{x}[n], \mathbf{c}[n-1])\,, \tag{6}$$

где

$$\Gamma[n] = \begin{pmatrix} \gamma_1[n], & 0 & \ldots & 0 \\ 0, & \gamma_2[n] & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & \ldots & \gamma_N[n] \end{pmatrix}.$$

Для разнообразных задач обучения прежде всего выбирается класс аппроксимирующих функций и мера уклонения, которая характеризует точность аппроксимации. Случайную меру уклонения определяем как некоторую выпуклую функцию от $y = f(\mathbf{x})$ и $\hat{f}(\mathbf{x}, \mathbf{c})$.

Обычно мера уклонения определяется как функция F разности $y - \hat{f}(\mathbf{x}, \mathbf{c})$ и в качестве меры аппроксимации выбирается функционал [1] в виде математического ожидания

$$J(\mathbf{c}) = E_x\{F(y - \hat{f}(\mathbf{x}, \mathbf{c}))\} = E_x\{Q(\mathbf{x}, \mathbf{c})\}\,, \tag{7}$$

где за $\hat{f}(\mathbf{x}, \mathbf{c})$ можно подставлять аппроксимирующую функцию $\hat{f}_1(\mathbf{x}, \mathbf{c})$ или $\hat{f}_2(\mathbf{x}, \mathbf{c})$.

Напомним, что задача об аппроксимации в стохастическом смысле возникла в связи с развитием регрессионного анализа. Эта задача и заключается в построении итерационного процесса, который предполагается сходящимся в вероятностном смысле.

Минимум функционала (7) будем искать (учитывая аппроксимацию (2)) по измеренным градиемтным реализациям

$$\frac{\partial J(\mathbf{c})}{\partial c_0} = -F'[y - (\prod_{i=1}^{N} (\varphi_i(\mathbf{x}) + c_i) + c_0)] \,.\, 1 \,,$$

$$\frac{\partial J(\mathbf{c})}{\partial c_1} = -F'[y - (\prod_{i=1}^{N} (\varphi_i(\mathbf{x}) + c_i) + c_0)] \prod_{\substack{i=1 \\ i \neq 1}}^{N} (\varphi_i(\mathbf{x}) + c_i) \,,$$

. . . .

$$\frac{\partial J(\mathbf{c})}{\partial c_j} = -F'[y - (\prod_{i=1}^{N} (\varphi_i(\mathbf{x}) + c_i) + c_0)] \prod_{\substack{i=1 \\ i \neq j}}^{N} (\varphi_i(\mathbf{x}) + c_i) \,,$$

. . . .

$$\frac{\partial J(\mathbf{c})}{\partial c_N} = -F'[y - (\prod_{i=1}^{N} (\varphi_i(\mathbf{x}) + c_i) + c_0)] \prod_{\substack{i=1 \\ i \neq N}}^{N} (\varphi_i(\mathbf{x}) + c_i) \,.$$

Обозначим

$$\nabla_{\mathbf{c}}[\prod_{i=1}^{N} (\varphi_i(\mathbf{x}) + c_i) + c_0] =$$
$$= (1, \prod_{\substack{i=1 \\ i \neq 1}}^{N} (\varphi_i(\mathbf{x}) + c_i), \ldots, \prod_{\substack{i=1 \\ i \neq j}}^{N} (\varphi_i(\mathbf{x}) + c_i), \ldots, \prod_{\substack{i=1 \\ i \neq N}}^{N} (\varphi_i(\mathbf{x}) + c_i)) \,.$$

Потом

$$\nabla_{\mathbf{c}} F[y - (\prod_{i=1}^{N} (\varphi_i(\mathbf{x}) + c_i) + c_0)] =$$
$$= -F'[y - (\prod_{i=1}^{N} (\varphi_i(\mathbf{x}) + c_i) + c_0)] \nabla_{\mathbf{c}}[\prod_{i=1}^{N} (\varphi_i(\mathbf{x}) + c_i) + c_0] \,.$$

С учетом предыдущего вероятностный итеративный алгоритм обучения в дискретной форме принимает вид

$$\mathbf{c}[n] = \mathbf{c}[n-1] + \gamma[n] \, F'\{y[n] - (\prod_{i=1}^{N} (\varphi_i(\mathbf{x}[n]) + c_i[n-1]) + c_0[n-1])\} \times$$
$$\times \nabla_{\mathbf{c}}\{\prod_{i=1}^{N} (\varphi_i(\mathbf{x}[n]) + c_i[n-1]) + c_0[n-1]\} \,,$$

или в непрерывной форме

$$\frac{\mathrm{d}\mathbf{c}(t)}{\mathrm{d}t} = \gamma(t) \, F'[y(t) - (\prod_{i=1}^{N} [\varphi_i(\mathbf{x}(t)) + c_i(t)] + c_0(t))] \times$$
$$\times \nabla_{\mathbf{c}}[\prod_{i=1}^{N} [\varphi_i(\mathbf{x}(t)) + c_i(t)] + c_0(t)] \,.$$

Упрощенная схема реализации этого алгоритма в дискретной форме показана на рис. 1. В блочную схему включены функции $\varphi_i(\mathbf{x})$ и $F'(.)$. Для трех составляющих входного вектора получим схему на рис. 2.

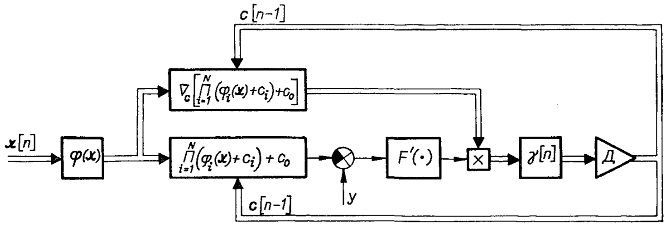

Рис. 1.

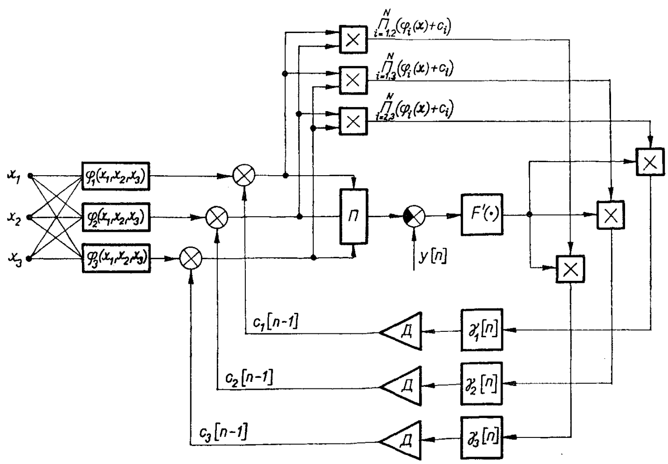

Рис. 2.

При классификации до двух групп можно аппроксимирующую функцию записать в виде

$$\hat{f}_1(\mathbf{x}[n], \mathbf{c}[n-1]) = \prod_{i=1}^{N} (\varphi_i(\mathbf{x}[n]) + c_i[n-1]) + c_0[n-1],$$

или

$$\hat{f}_2(\mathbf{x}[n], \mathbf{c}[n-1]) = \sum_{j=1}^{l} \prod_{i=m_{j-1}+1}^{m_j} (\varphi_i(\mathbf{x}[n]) + c_i[n-1]) + c_0[n-1].$$

На основании первой аппроксимации и разных критериев качества можно составить алгоритмы которые будут эти функционалы минимизировать:

Критерий

$$J(\mathbf{c}) = E_x\{\{\operatorname{sign} y - \operatorname{sign}(\prod_{i=1}^{N} (\varphi_i(\mathbf{x}) + c_i) + c_0)\} (\prod_{i=1}^{N} (\varphi_i(\mathbf{x}) + c_i) + c_0)\}.$$

Алгоритм

$$\mathbf{c}[n] = \mathbf{c}[n-1] +$$
$$+ \gamma[n] \{\operatorname{sign} y[n] - \operatorname{sign}[\prod_{i=1}^{N} (\varphi_i(\mathbf{x}[n]) + c_i[n-1]) + c_0[n-1]]\} \times$$
$$\times \nabla_{\mathbf{c}}[\prod_{i=1}^{N} (\varphi_i(\mathbf{x}[n]) + c_i[n-1]) + c_0[n-1]].$$

Критерий

$$J(\mathbf{c}) = E_x\{|y - (\prod_{i=1}^{N} (\varphi_i(\mathbf{x}) + c_i) + c_0)|\}.$$

Алгоритм

$$\mathbf{c}[n] = \mathbf{c}[n-1] +$$
$$+ \gamma[n] \operatorname{sign}\{y[n] - (\prod_{i=1}^{N} (\varphi_i(\mathbf{x}[n]) + c_i[n-1]) + c_0[n-1])\} \times$$
$$\times \nabla_{\mathbf{c}}[\prod_{i=1}^{N} (\varphi_i(\mathbf{x}[n]) + c_i[n-1]) + c_0[n-1]].$$

Критерий

$$J(\mathbf{c}) = E_x\{(y - (\prod_{i=1}^{N} (\varphi_i(\mathbf{x}) + c_i) + c_0))^2\}.$$

Алгоритм

$$\mathbf{c}[n] = \mathbf{c}[n-1] +$$
$$+ G[n] \{y[n] - (\prod_{i=1}^{N} (\varphi_i(\mathbf{x}[n]) + c_i[n-1]) + c_0[n-1])\} \times$$
$$\times \nabla_{\mathbf{c}}[\prod_{i=1}^{N} (\varphi_i(\mathbf{x}[n]) + c_i[n-1]) + c_0[n-1]].$$

4. АЛГОРИТМЫ ОСНОВАННЫЕ НА СОВПАДЕНИИ ЗНАКОВ y И $\prod_{i=1}^{N}(\varphi_i(\mathbf{x}) + c_i) + c_0$ И ОПРЕДЕЛЕНИИ $\mathbf{c}$ ИЗ СИСТЕМЫ НЕРАВЕНСТВ

Тождество знаков выполняется если

$$y[n]\,(\prod_{i=1}^{N}(\varphi_i(\mathbf{x}[n]) + c_i[n-1]) + c_0[n-1]) > 0\,.$$

$$y[n]\,(\prod_{i=1}^{N}(\varphi_i(\mathbf{x}[n]) + c_i[n-1]) + c_0[n-1]) = \xi[n-1]\,,\quad \xi > 0\,.$$

Критерий качества запишем в виде

$$J(\mathbf{c}, \xi) = E_x\{F[y(\prod_{i=1}^{N}(\varphi_i(\mathbf{x}) + c_i) + c_0) - \xi]\}\,.$$

Для решения применим алгоритмы

$$\mathbf{c}[n] = \mathbf{c}[n-1] -$$
$$-\gamma_1[n]F'\{y[n]\,[\prod_{i=1}^{N}(\varphi_i(\mathbf{x}[n]) + c_i[n-1]) + c_0[n-1]] - \xi[n-1]\}\times$$
$$\times\nabla_{\mathbf{c}}\{[\prod_{i=1}^{N}(\varphi_i(\mathbf{x}[n]) + c_i[n-1]) + c_0[n-1]]\,y[n]\}\,,$$

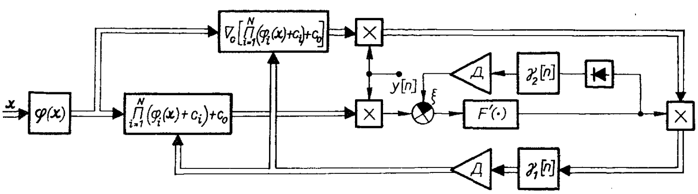

Рис. 3.

$$\xi[n] = \xi[n-1] +$$
$$+\gamma_2[n]\,\{F'[y[n]\,(\prod_{i=1}^{N}(\varphi_i(\mathbf{x}[n]) + c_i[n-1]) + c_0[n-1]) - \xi[n-1]] +$$
$$+\,|F'[y[n]\,(\prod_{i=1}^{N}(\varphi_i(\mathbf{x}[n]) + c_i[n-1]) + c_0[n-1]) - \xi[n-1]]|\}\,,$$

$$\xi(0) > 0\,.$$

Реализация этих алгоритмов показана на рис. 3.

Обучение можно ускорить выбором (в предыдущих алгоритмах) оптимальной величины $\gamma_{\text{опт}}[n]$. Это можно сделать для каждого входного вектора аналитическим способом. Ускорить процесс обучения можно также использованием

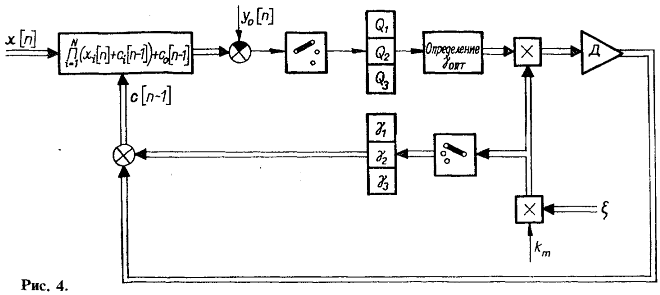

Рис. 4.

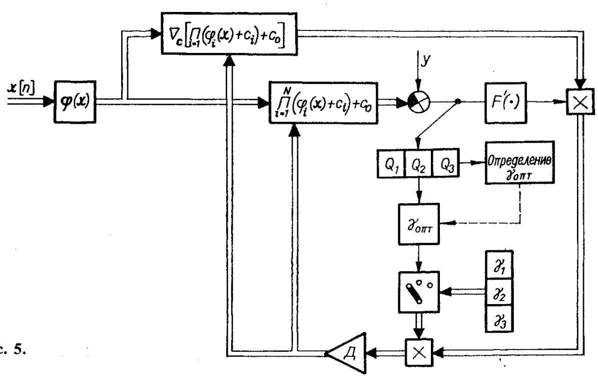

Рис. 5.

алгоритмов с параболической интерполяцией или алгоритмов со случайными векторами и параболической интерполяцией. Блочная схема реализации алгоритмов обучения с параболической интерполяцией показана на рис. 4 и схема алгоритма с вспомогательными случайными векторами и параболической интерполяцией на рис. 5.

Доказательство сходимости приведенных алгоритмов и связь „класических" и „некласических" аппроксимаций будут опубликованы в последующих номерах журнала.

Новая аппроксимация была проверена на АВМ а ЦВМ в алгоритмах классификации и распознавания образов.

(Поступила 26. июня 1970.)

ЛИТЕРАТУРА

[1] Цыпкин Я. З.: Основы теории обучающихся систем. Изд. „Наука", Москва 1970.

[2] Цыпкин Я. З.: Адаптация и обучение в автоматических системах. Изд. „Наука", Москва 1968.

VÝTAH

Některé obecné algoritmy učení

ZDENĚK ČASTORÁL

Článek je věnován obecným algoritmům učení, které je možné využít jak pro klasifikaci a rozeznávání obrazů, tak i pro identifikaci, adaptivní filtraci a vlastní řízení. Vychází se z nové aproximační funkce a ta se využívá v diskrétních i spojitých algoritmech. Posuzují se metody zrychlení konvergence.

Ing. Zdeněk Častorál, CSc., Fakulta elektrotechnická ČVUT (Политехнический институт — электротехнический факультет), Karlovo nám. 13, Praha 2.