КОНСТРУКТИВНЫЙ TOWER АЛГОРИТМ ДЛЯ ОБУЧЕНИЯ НЕЙРОННЫХ СЕТЕЙ ИЗ $\Sigma\Pi$ НЕЙРОНОВ

Шибзухов З.М.

Научно-исследовательский институт прикладной математики и автоматизации Кабардино-Балкарского научного центра РАН, ул. Шортанова 89А, г. Нальчик, 360000, КБР, РОССИЯ, e-mail: sz@zmail.ru

Аннотация

Предлагается конструктивный Tower алгоритм для построения нейронной сети (НС), представляющей собой последовательность $\Sigma\Pi$ – нейронов. Прототипом для него явился конструктивный Tower алгоритм для построения НС, представляющей собой последовательность формальных нейронов (Σ – нейронов), первоначально предложенный в [1]. В отличие от своего предшественника, он имеет более высокую скорость сходимости процесса конструктивного обучения.

1. Введение

Конструктивный Tower алгоритм для классификации (в случае 2-х классов) был первоначально предложен в [1]. С его помощью строится последовательность Σ – нейронов (Рис. 1) с функцией преобразования вида:

$$\mathsf{sgn}(\theta + \mathbf{w} \cdot \mathbf{x}),$$

где $\mathbf{x} \in \{0,1\}^n$ — вектор входов, $\theta \in \mathbb{Z}$, $\mathbf{w} \in \mathbb{Z}^n$ — вектор весов Σ – нейрона, $\mathbf{w} \cdot \mathbf{x}$ — скалярное произведение $\mathbf{w}$ и $\mathbf{x}$,

$$\mathsf{sgn}(x) = \begin{cases} 1, & \text{если} \quad x \geqslant 0, \\ 0, & \text{если} \quad x < 0. \end{cases}$$

Входы первого нейрона связаны только с входами НС, входы всех остальных нейронов связаны с выходом предыдущего нейрона и входами НС. При конструктивном обучении НС по алгоритму Tower последовательно добавляются в конец Σ – нейроны и обучаются до тех пор, пока не будет достигнута требуемая точность классификации. Обучение каждого нового Σ – нейрона осуществляется при помощи алгоритма Pocket [1, 2]. Этот алгоритм является модификацией классического алгоритма обучения перцептрона для поиска оптимального вектора весов, при котором правильно классифицируется наибольшая часть примеров. Известно, что такая задача принадлежит к классу задач экспоненциальной сложности [3]. Поэтому в алгоритме Pocket задается верхняя граница для времени поиска оптимального набора весов Σ – нейрона.

Недостатки Tower алгоритма, в основном, являются следствием ограничений, связанных с моделью Σ–нейрона, а также недостатков алгоритмов его обучения.

Переход от модели Σ–нейронов к модели $\Sigma\Pi$–нейронов, которые описываются полилинейными функциями, позволяет снять большую часть недостатков. Хотя модель $\Sigma\Pi$–нейрона является более сложной, однако для неё можно построить простые алгебраические рекуррентные алгоритмы обучения по специальным образом упорядоченным обучающим последовательностям.

2. Последовательности $\Sigma\Pi$ нейронов

Рассмотрим НС с аналогичной архитектурой, но построенные на основе $\Sigma\Pi$ – нейронов с функцией преобразования вида:

$$\mathsf{spn}(\mathbf{x}) = f\big(\mathsf{spu}(\mathbf{x})\big), \tag{1}$$

$$\mathsf{spu}(\mathbf{x}) = \sum_{k=1}^{\mathrm{R}} w_k \prod(\mathbf{x}; \mathbf{i}_k), \tag{2}$$

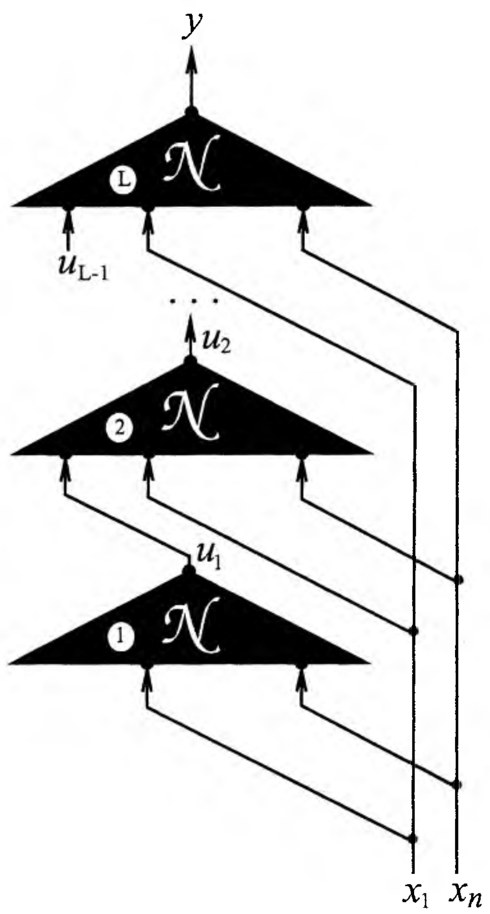

Рисунок 1: Архитектура НС.

где $f : \mathbb{Z} \to \{0,1\}$ — скалярная функция выхода, $\mathbf{x} \in \{0,1\}^n$ — вектор логических входов, $w_0, w_k \in \mathbb{Z}$ — весовые коэффициенты, $\mathbf{i}_k \subseteq \{1,\ldots,n\}$ — мультииндексы,

$$\prod(\mathbf{x};\mathbf{i}_k) = \prod_{i\in\mathbf{i}_k} x_i.$$

Далее символ spu будем использовать для обозначения ΣΠ-форм вида (2), а символ spn для обозначения функции преобразования ΣΠ-нейронов вида (1)-(2).

Введем два показателя сложности ΣΠ-форм нейронов:

$$\Sigma\text{–сложность}: \quad c_\Sigma(\mathsf{spu}) = \mathrm{R};$$
$$\Pi\text{–сложность}: \quad c_\Pi(\mathsf{spu}) = \max_{k=\overline{1,\mathrm{R}}} |\mathbf{i}_k|.$$

Σ-сложность характеризует число мультипликативных слагаемых, а Π-сложность — наибольшее число сомножителей в слагаемых ΣΠ-формы (2).

Рассматриваемые НС представляют собой последовательности ΣΠ-нейронов, в которых осуществляется цепочка преобразований:

$$u_1 = \mathsf{spn}_1(\mathbf{x}),$$
$$u_\ell = \mathsf{spn}_\ell(u_{\ell-1},\mathbf{x}), \quad \ell = 2,\ldots,\mathrm{L}-1,$$
$$y = \mathsf{spn}_\mathrm{L}(u_{\mathrm{L}-1},\mathbf{x}),$$

где $\mathsf{spn}_1,\ldots,\mathsf{spn}_\mathrm{L}$ — функции преобразования ΣΠ-нейронов, L — количество нейронов в последовательности. При этом, предполагаем, что все нейроны в последовательности имеют ограниченную Σ-сложность, т.е. $c_\Sigma(\mathsf{spu}_\ell) \leqslant \mathrm{M}$.

Рассмотрим НС с логическими входами и выходами, содержащие логические ΣΠ-нейроны, т.е. $\mathbf{x} \in \{0,1\}^n$, $u_1,\ldots,u_\mathrm{L} \in \{0,1\}$, $y \in \{0,1\}$. Такие НС реализуют логические функции от n переменных. Заметим, что если нет ограничения на $c_\Sigma(\mathsf{spu})$ и $c_\Pi(\mathsf{spu})$ (на число слагаемых и число сомножителей в слагаемых ΣΠ-формы, соответственно), то любую логическую функцию можно реализовать при помощи единственного ΣΠ-нейрона.

3. Tower алгоритм для конструктивного обучения последовательностей ΣΠ нейронов

Обучение НС осуществляем на основе последовательности входных логических векторов $\mathbf{X} = \{\mathbf{x}_k : k = \overline{1,\mathrm{N}}\}$ и, соответствующей ей последовательности ожидаемых на выходе логических значений $\mathbf{Y} = \{y_k : k = \overline{1,\mathrm{N}}\}$. Предполагаем, что $\mathbf{X}$ является $\not\geqslant$-упорядоченной последовательностью, т.е. для каждой пары $1 \leqslant j < k \leqslant \mathrm{N}$ найдется индекс $1 \leqslant t \leqslant n$ такой, что $x_{jt} < x_{kt}$.

В процессе обучения строим последовательность ΣΠ-форм:

$$\mathsf{spu}_1(\mathbf{x}),\ \mathsf{spu}_2(u_1,\mathbf{x}),\ \ldots,\ \mathsf{spu}_\ell(u_{\ell-1},\mathbf{x}),\ \ldots \quad (3)$$

где $u_\ell = \mathsf{spn}_\ell(u_{\ell-1},\mathbf{x})$, $\mathsf{spn}_\ell = f(\mathsf{spu}_\ell)$, $1 < \ell \leqslant \mathrm{L}$, L — количество нейронов. Как будет показано ниже, $\mathrm{L} \leqslant \left[\frac{\mathrm{N}}{\mathrm{M}}\right] + 1$.

Последовательность (3) строится таким образом, чтобы выполнялось условие:

A) для всех $j = \overline{1, \mathrm{N}_{\ell-1}}$ имело место равенство $\mathsf{spn}_\ell(u_{\ell-1,j}, \mathbf{x}_j) = u_{\ell-1,j}$ (при $\ell > 1$);

B) для всех $j = \overline{\mathrm{N}_{\ell-1}{+}1, \mathrm{N}_\ell}$ имело место равенство $y_j = \mathsf{spn}_\ell(\mathsf{NN}_{\ell-1}(\mathbf{x}_j), \mathbf{x}_j)$ (при всех ℓ),

где $\mathrm{N}_0 = 0$, $1 < \mathrm{N}_1 < \ldots < \mathrm{N}_{\ell-1} < \mathrm{N}_\ell < \mathrm{N}$,

$u_{\ell-1,j} = \mathsf{NN}_{\ell-,1}(\mathbf{x}_j)$,

$\mathsf{NN}_1(\mathbf{x}) = \mathsf{spn}_1(u_0, \mathbf{x})$, ...,

$\mathsf{NN}_\ell(\mathbf{x}) = \mathsf{spn}_\ell(\mathsf{NN}_{\ell-1}(\mathbf{x}), \mathbf{x})$, ...

Очередная ℓ-ая $\Sigma\Pi$–форма $\mathsf{spu}_\ell(u_{\ell-1}, \mathbf{x})$ строится по $\mathbf{X}$ и $\mathbf{U}_\ell = \{u_{\ell-1,j} : j = \overline{1, \mathrm{N}}\}$.

Легко видеть, что в этом случае для всех $j = \overline{1, \mathrm{N}_\ell}$ будет иметь место равенство $\mathsf{NN}_\ell(\mathbf{x}_j) = y_j$.

Таким образом, задача построения последовательности $\Sigma\Pi$ – нейронов (3) разделяется на L последовательных задач конструктивного обучения $\Sigma\Pi$ – нейронов.

Первый $\Sigma\Pi$ – нейрон строится по последовательности входных логических векторов $\mathbf{X}$, последовательности ожидаемых логических значений $\mathbf{Y}$, каждый ℓ-ый ($\ell > 1$) $\Sigma\Pi$ – нейрон строится по последовательности входных логических векторов $\mathbf{X}$, последовательности ожидаемых логических значений $\mathbf{Y}$ и последовательности логических величин $\mathbf{U}_\ell$ так, чтобы выполнялись условия A и B. Построение каждого $\Sigma\Pi$ – нейрона НС основано на следующем утверждении.

Теорема. Пусть задана $\not\succcurlyeq$-упорядоченная последовательность логических векторов $\mathbf{X}$, последовательность логических величин $\mathbf{Y}$, последовательность логических величин $\mathbf{U} = \{u_k \in \{0,1\} : k = \overline{1, \mathrm{N}}\}$ такая, что для всех $j = \overline{1, \mathrm{N}_0}$ ($\mathrm{N}_0 > 0$) имеет место равенство $y_j = u_j$. Тогда существует $\Sigma\Pi$ – форма $\mathsf{spu}(u, \mathbf{x})$ с $c_\Sigma(\mathsf{spu}) \leqslant \mathrm{M}$ такая, что

1) для всех $j = \overline{1, \mathrm{N}_0}$ будет иметь место равенство $\mathsf{spn}(u_j, \mathbf{x}_j) \equiv u_j$;

2) для всех $j = \overline{\mathrm{N}_0{+}1, \mathrm{N}_1}$, где $\mathrm{N}_1 \geqslant \mathrm{M}{+}\mathrm{N}_0$, будет иметь место равенство $y_j = \mathsf{spn}(u_j, \mathbf{x}_j)$.

Доказательство. Доказательство теоремы вытекает из следующей конструкции. Строим рекуррентную последовательность $\Sigma\Pi$ – форм:

$$\left.\begin{array}{l}
\mathsf{spu}^0(u, \mathbf{x}) \equiv g(u) \\
\mathsf{spu}^k(u, \mathbf{x}) \equiv \mathsf{spu}^{k-1}(u, \mathbf{x}) + w_k \prod(\mathbf{x}; \mathbf{i}_k), \\
\qquad \text{если} \quad \mathsf{spn}^{k-1}(u_k, \mathbf{x}_k) \neq y_k, \\
\mathsf{spu}^k(u, \mathbf{x}) \equiv \mathsf{spu}^{k-1}(u, \mathbf{x}), \\
\qquad \text{если} \quad \mathsf{spn}^{k-1}(u_k, \mathbf{x}_k) = y_k, \\
\mathsf{spn}^{k-1}(u, \mathbf{x}) = f\big(\mathsf{spu}^{k-1}(u, \mathbf{x})\big), \\
w_k = \widehat{z}_k - \mathsf{spu}^{k-1}(u_k, \mathbf{x}_k),
\end{array}\right\} \tag{4}$$

где

$\widehat{z}_k$ — такое число, что $f(\widehat{z}_k) = y_k$, $k = 1, 2, \ldots$,

$g(u)$ — линейная скалярная функция, такая что $f(g(u)) = u$.

Например, если $f(s)$ — пороговая функция со значениями в $\{0,1\}$, то $g(u) = 2u - 1$, или если $f(s) = s \pmod 2$, то $g(u) = u$.

Мультииндекс $\mathbf{i}_k$ строится, исходя из $\mathbf{i}_k^0 = \{i : x_{ki} = 1\}$ путём последовательного исключения несущественных индексов. Индекс $i \in \mathbf{i}_k$ является несущественным, если для каждого $j = \overline{1, k{-}1}$ найдётся $t = t(j) \in \mathbf{i}_k$ и $t \neq i$ такой, что $x_{jt} < x_{kt}$ (x_{jt} и x_{kt} — t-ая компоненты $\mathbf{x}_j$ и $\mathbf{x}_k$, соответственно). Строится последовательность мультииндексов $\mathbf{i}_k^0, \ldots, \mathbf{i}_k^p, \ldots$, в которой $\mathbf{i}_k^p$ получается из $\mathbf{i}_k^{p-1}$ путём исключения какого-нибудь одного несущественного индекса. При некотором p мультииндекс $\mathbf{i}_k^p$ не будет содержать несущественных индексов. В этом случае полагаем $\mathbf{i}_k = \mathbf{i}_k^p$. Использование процедуры исключения несущественных индексов (или, что то же самое, исключения несущественных сомножителей) позволяет в принципе строить $\Sigma\Pi$ – формы с значениями $c_\Pi(\mathsf{spu})$, близкими к минимальному.

В этом случае выполняются два важных свойства:

i) для любой пары j, k такой, что $1 \leqslant j < k \leqslant \mathrm{N}$ имеет место равенство $\prod(\mathbf{x}_j; \mathbf{i}_k) = 0$;

ii) для любого k такого, что $1 \leqslant k \leqslant \mathrm{N}$, имеет место равенство $\prod(\mathbf{x}_k; \mathbf{i}_k) = 1$.

Поэтому для каждого k будут выполняться условия:

1) при $j \leqslant \mathrm{N}_0$ выполняется равенство $\mathsf{spn}^k(u_j, \mathbf{x}_j) = u_j = y_j$ (согласно условию леммы);

2) если для всех $\mathrm{N}_0 < j \leqslant k{-}1$ выполняется равенство $\mathsf{spn}^{k-1}(u_j, \mathbf{x}_j) = y_j$, то

для всех $N_0 < j \leqslant k$ также будет выполняться равенство $\mathsf{spn}^k(u_j, \mathbf{x}_j) = y_j$.

Последовательность (4) обрывается на некотором шаге k_1 ($N_0 < k_1 < N$), если $c_\Sigma(\mathsf{spu}^{k_1-1}(u, \mathbf{x})) = M$, а $\mathsf{spn}^{k_1-1}(u_{k_1}, \mathbf{x}_{k_1}) \neq y_{k_1}$. Полагаем $\mathsf{spu}(u, \mathbf{x}) \equiv \mathsf{spu}^{k_1-1}(u, \mathbf{x})$. Если при некотором $k = N$ и количество слагаемых $\leqslant M$, то $\mathsf{spu}(u, \mathbf{x}) \equiv \mathsf{spu}^N(u, \mathbf{x})$ и $N_1 = N$. Что и доказывает настоящее утверждение. □

Рекуррентная процедура построения $\Sigma\Pi$ – нейрона по $\mathbf{X}$, $\mathbf{Y}$ и $\mathbf{U}$ является модификацией аналогичной рекуррентной процедуры построения порогово-полиномиальных функций по упорядоченным табличным данным [4–6] и развитием рекуррентного алгоритма обучения $\Sigma\Pi$ – нейрона по упорядоченной паре последовательностей $\mathbf{X}$, $\mathbf{Y}$ [7].

Теперь приступим к построению последовательности (3).

Строим сначала первый $\Sigma\Pi$ – нейрон так, чтобы выполнялось условие B ($\ell = 1$).

Затем последовательно строим $\Sigma\Pi$ – формы $\mathsf{spu}_\ell(u_{\ell-1}, \mathbf{x})$ ($\ell = 2, 3, \ldots$) так, чтобы выполнялись условия A и B.

Процесс построения сети завершаем, когда $N_\ell = N$.

Корректность и сходимость данной рекуррентной процедуры построения последовательности (3) вытекает из приведенного выше утверждения.

Обученные $\Sigma\Pi$ – формы удовлетворяют следующим требованиям:

1) для каждого $\ell = \overline{1, L}$ $c_\Sigma(\mathsf{spu}_\ell) \leqslant M$;

2) для всех $j = \overline{1, N_1}$ $\mathsf{spn}_1(\mathbf{x}_j) = y_j$;

3) при каждом $\ell = \overline{2, L}$ для всех $j = \overline{1, N_\ell}$ имеет место равенство $\mathsf{spn}_\ell(u_{\ell-1,j}, \mathbf{x}_j) = y_j$, где $u_{\ell-1,j} = \mathsf{NN}_{\ell-1}(\mathbf{x}_j)$.

4. Заключение

Таким образом, в результате получаем последовательность $\Sigma\Pi$ – нейронов, образующих НС, такую что для каждого $\mathbf{x}_j \in \mathbf{X}$ на выходе НС получается ожидаемое значение $y_j \in \mathbf{Y}$.

Каждый $\Sigma\Pi$ – нейрон имеет ограниченную Σ – сложность. Последовательность мультииндексов каждого $\Sigma\Pi$ – нейрона состоит из элементов, содержащих только существенные индексы. Это означает, что Π – сложность нейронов близка к минимально возможной.

При построении каждого $\Sigma\Pi$ – нейрона используется не менее M элементов обучающей последовательности $\mathbf{X}$, поэтому $L \leqslant \left[\frac{N}{M}\right] + 1$.

Работа выполнена при поддержке гранта РФФИ No. 01–01–00142, а также при при поддержке Комиссии РАН по работе с молодежью.

Литература

[1] Gallant S. Perceptron based learning algorithms. *IEEE Transactions on Neural Networks*. 1(2): 179–191. 1990.

[2] Gallant S. *Neural network learning algorithms*. MIT Press. Cambridge. MA. 1993.

[3] Garey M., Johnson D. *Computers and intractability*. W.H. Freeman. New York. 1979.

[4] Тимофеев А.В., Пшибихов А.В. Алгоритмы обучения и минимизации сложности полиномиальных распознающих систем. // *Изв. АН СССР. Техн. кибернетика*. 1974. №5. С.214–217.

[5] Каляев А.В., Тимофеев А.В. Методы обучения и минимизации сложности когнитивных нейромодулей супер-макронейрокомпьютера с программируемой архитектурой. // *Доклады РАН*. 1994. Т.337. №2. С.180–183.

[6] Шибзухов З.М. Рекуррентная схема построения кортежей многозначных функций и обучения нейронных сетей. // *Доклады Адыгской академии наук*. Нальчик. 1998. Т.3. №2. С.45–51.

[7] Шибзухов З.М. Конструктивные рекуррентные алгоритмы синтеза ΣΠ–нейронных сетей. // *Доклады Адыгской академии наук*. Нальчик. 2000. Т.5. №1. С.72–77