Выбор функции активации при прогнозировании нейронными сетями

Г. И. РУДОЙ

Аннотация. Черновик работы по порождению существенно нелинейных авторегрессионных моделей.

1. Система обозначений

- $\mathcal{A}$ — множество индексов информативных признаков.
- G — множество элементарных функций, используемых для составления суперпозиций. [1]
- $\overline{G}$ — пополненное множество элементарных функций: $\overline{G} = G \cup G_X = \{\overline{g}_i : X \to R \mid i = 1, \dots, N\}$, где $\overline{g}_i$ — функция, возвращающая i-ый признак из множества признаков X.
- $\mathcal{I}$ — множество индексов всех признаков.
- N — мощность множества числовых признаков (число признаков).
- $\overline{N}$ — мощность пополненного множества числовых признаков ($\overline{N} = |G| + N$).
- x_i — признак.
- X — множество числовых признаков: $X = \cup_{i=1}^{N} x_i$.

2. Описание регрессионных моделей при помощи матриц смежности

Условимся руководствоваться следующим очевидным правилом при составлении суперпозиций: в листьях соответствующего дерева выражений содержатся функции из G_X и только они.

Сопоставим суперпозиции матрицу смежности, ее описывающую, и укажем способ ее построения.

Пусть дана произвольная суперпозиция функций из G в инфиксной записи. Требуется построить соответствующую данной суперпозиции матрицу. Таким образом мы заодно и опишем, что представляет из себя матрица, характеризующая суперпозицию, и укажем некоторые ее свойства.

Первым шагом является построение дерева грамматического разбора выражения в инфиксной записи, эта тема хорошо изучена и разобрана, например, в [1], поэтому положим, что мы уже имеем дерево выражения. Заметим, что дерево выражения

Научный руководитель В. В. Стрижов.

[1]Имеет смысл четко определить, откуда и куда действуют функции из G: $G = g : R^n \to R^n \mid \forall n$ или же $G = g : R \to R \vee R \times R \to R$. В принципе, второго случая должно за глаза хватить — используемые функции либо одноместные (например, log, tan, $\neg$), либо двухместные (например, $+$, $\times$, $\wedge$).

является ориентированным графом. Пронумеруем его вершины, запустив по графу поиск в глубину. Присвоим каждой вершине номер, соответствующий ее порядку при таком обходе в глубину, и условимся считать первой вершину, из которой был начат обход.

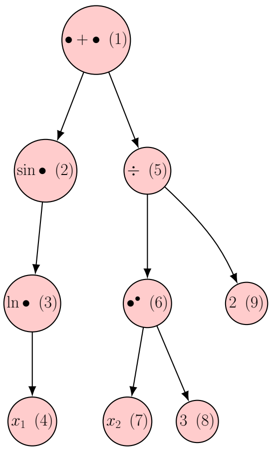
Если в суперпозиции участвует n функций, то, соответственно, получим индексы $1, \dots, n$. Построим матрицу $A = \|a_{ij}\|$ размера $n \times n$ и запишем в элемент a_{ij} единицу, если в графе есть ребро, соединяющее i-ую и j-ую вершины, иначе — ноль.

Так как в G могут существовать коммутативные сами с собой операции, то порядок следования узлов графа, вообще говоря, недетерминирован, поэтому вместе с матрицей необходимо хранить вектор размерности n, указывающий, какому индексу соответствует какая функция.

Заметим следующие очевидные свойства:

- На диагонали матрицы стоят нули (так как в графе по определению отсутствуют петли).
- Матрица верхнетреугольная (так как граф — дерево, и по построению нумерации ребра есть только от элементов с меньшим номером к элементам с большим).
- В каждом столбце может быть только одна единица, исключая первый столбец, в котором ноль единиц.
- Сумма числа единиц в строке характеризует арность функции с соответствующим индексом и ее принадлежность к G либо G_X: унарная функция из G имеет одну единицу, бинарная G — две, а имеющая только нули функция лежит в G_X.

2.1. **Пример.** Пусть дано выражение $\sin(\ln x_1) + \frac{x_2^3}{2}$. Построим соответствующий граф и пронумеруем вершины:



Тогда матрица смежности для этого графа будет выглядеть как:

$$
\begin{array}{cc}
 & \begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \end{matrix} \\
\begin{matrix} + \\ \sin \\ \ln \\ x_1 \\ \div \\ \bullet^{\bullet} \\ x_2 \\ 3 \\ 2 \end{matrix} &
\begin{pmatrix}
0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}
\end{array}
$$

2.2. **Сложность модели.** Если считать сложностью модели число используемых признаков, то число нулевых строк в матрице смежности как раз и будет характеризовать сложность модели.

3. Описание регрессионных моделей при помощи матриц инцидентности

Аналогично методу построения матриц смежности, представим данное выражение ориентированным графом и запустим поиск в глубину, нумеруя по мере обхода вершины и ребра, и условимся считать вершину, из которой начат обход, первой. Очевидно, что ребер на единицу меньше, чем функций, и что ребро перед i-ой вершиной имеет номер $i-1$.

Построим матрицу размера $n \times (n-1)$, где n — число функций, участвующих в суперпозиции, и для каждого столбца за номером i запишем единицы в клетки с номерами строк, соответствующими номерам функций, соединяемых этим ребром, а все остальные клетки запишем нули.

Заметим, что матрица инцидентности сама по себе характеризует лишь структуру соответствующего выражения, и необходимо хранить еще и вектор функций, ставящий в соответствие номеру строки конкретную функцию из выражения. Назовем такой вектор связанным с матрицей вектором (или просто связанным вектором).

- В каждом столбце матрицы ровно две единицы.
- Для первой (по порядку обхода) функции число единиц в соответствующей строке равно арности функции, для всех остальных функций это число равно арности функции, увеличенной на единицу.
- $a_{ij} = 1 \mid \forall j, i = j+1$ и $a_{ij} = 0 \mid \forall j, \forall i > j+1$.

Таким образом, соблюдение и проверка вышеупомянутых свойств матрицы при дальнейших преобразованиях, в том числе, гарантирует выполнение базовых ограничений, получаемых из соображений разумности, таких, как:

- Соответствие реального числа аргументов в выражении арности функции (достаточно сопоставить количество единиц в соответствующей строке информации с функцией в связанном векторе).

- Отсутствие функций, вызывающих сами себя (для выполнения этого условия необходимо и достаточно выполнение первого и третьего свойств).

Отметим также следующие ограничения, которые также тривиально проверяются для матриц инцидентности:

- Соответствие области значений дочерних функций области определения вызывающей функции.
- Условия на область определения результата.
- Соответствие типов аргументов.

3.1. **Структура матрицы и связь с поддеревьями.** Рассмотрим некоторое поддерево выражения. Пусть оно начинается с вершины за номером b, и последняя вершина по порядку обхода имеет номер e.

Номера вершин поддерева содержатся в интервале $[b, e]$ и, более того, для каждого натурального числа из этого интервала найдется вершина, имеющая соответствующий номер; это следует из порядка обхода графа. Из этого напрямую следует, что подматрица, соответствующая данному поддереву, лежит в диапазоне между клетками $(b, b-1)$ и $(e, e-1)$.

3.1.1. *Выделение поддерева по номеру начальной вершины.* Таким образом, с учетом указанных выше свойств, можно сформулировать следующий алгоритм нахождения подматрицы, описывающей поддерево, начинающееся с вершины за номером b (который, по факту, номер последней вершины e):

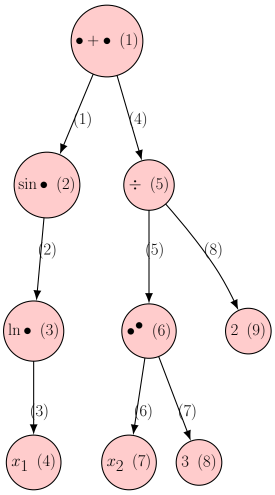
(1) $i := b$
(2) Находим $c_{max} := \sup\{j \mid a_{ij} = 1\}$ (множество не пусто по свойству матрицы 3). Если $c_{max} = i - 1$, то завершаем алгоритм с $e := i$, иначе переходим к следующему шагу.
(3) Находим $r_{max} := \sup\{r \mid a_{rc_{max}} = 1\}; i := r_{max}$ и переходим к предыдущему шагу.

Алгоритм всегда завершается, так как на каждом шаге мы сдвигаемся вправо либо вниз, и матрица ограничена.

Заметим также, что последний шаг корректен, и $r_{max} > i$.[2]

3.2. **Пример.** Пусть дано выражение $\sin(\ln x_1) + \frac{x_2^3}{2}$. Построим соответствующий граф, пронумеруем вершины и ребра:

[2] Строго доказать, почему



Тогда матрица инцидентности для этого графа будет выглядеть как:

$$\begin{array}{c} \\ + \\ \sin \\ \ln \\ x_1 \\ \div \\ \bullet^{\bullet} \\ x_2 \\ 3 \\ 2 \end{array}
\begin{array}{c} \begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \end{matrix} \\
\begin{pmatrix}
1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{pmatrix} \end{array}$$

3.3. **Применение генетических алгоритмов.** Выразим операции кроссовера и мутации через язык матриц инцидентности.

3.3.1. *Мутация.* Мутация, то есть, замена функции на другую, сводится к замене функции на соответствующей позиции в связанном векторе на другую функцию. При этом необходимо обеспечить сохранение арности, то есть, заменять функцию только на функцию с таким же числом аргументов. Необходимо также не испортить области значения: сохранить соответствие области значений дочерних функций области определения новой функции, а области значений новой функции — областям определения функций, для которых новая функция является дочерней.

3.3.2. *Кроссовер.* Рассмотрим сначала одноточечный кроссовер, при котором поддерево выражения заменяется на соответствующее поддерево другого выражения целиком, начиная с некоторой вершины и до листьев дерева.

Пусть номер вершины, начиная с которой произведется замена, в одном выражении — b_1, а в другом b_2, при этом соответствующим поддеревьям принадлежат вершины с номерами, не большими e_1 и e_2, и эти границы достигаются. Тогда необходимо:

(1) Выделить в матрице первого выражения подматрицу, начиная с клетки $(b_1, b_1 - 1)$ и до клетки $(e_1, e_1 - 1)$ включительно.
(2) Выделить в матрице второго выражения подматрицу, начиная с клетки $(b_2, b_2 - 1)$ и до клетки $(e_2, e_2 - 1)$ включительно.
(3) Обменять выделенные подматрицы местами, при необходимости записывая в недостающие клетки нули (если новая подматрица больше выделенной) либо убирая ненужные столбцы и строки (если новая подматрица меньше).
(4) Обменять элементы в связанных векторах начиная с n_1 для первого вектора и с n_2 для второго соответственно.

Однако, генетические алгоритмы работают напрямую с матрицами и не следят за сопутствующим графом (это было бы неоправданной тратой ресурсов), поэтому необходимо заметить, что соответствующая процедура должна для сохранения корректности и разумности выражения выбирать те подматрицы, для которых на строках, принадлежащих этой подматрице, вне этой подматрицы стоят только нули, и аналогично для столбцов, принадлежащих этой подматрице, кроме самого левого, который, возможно, имеет одну единицу над подматрицей.

Заметим, что, с учетом предыдущего замечания, по указанному ранее свойству локализации поддерева в матрице эти операции корректны и приводят к корректным результирующим матрицам и, таким образом, при кроссовере необходимо вручную проверять только соответствие областей определения.

Рассмотрим для примера, как в матричном представлении выражается получение выражения $\sin x_1 + x_1 \cos x_2$ из выражений $\sin x_1 + x_2$ и $x_1 + x_1 \cos x_2$, то есть, в соответствующем графе для первого родительского выражения поддерево, отвечающее подвыражению x_2, заменяется на поддерево $x_1 \cos x_2$.

Запишем матрицы инцидентности для первого и второго исходных выражений, соответственно, и отметим соответствующие поддеревьям подматрицы:

$$
\left(\begin{array}{c|ccc}
+ & 1 & 0 & 1 \\
\sin & 1 & 1 & 0 \\
x_1 & 0 & 1 & 0 \\ \cline{4-4}
x_2 & 0 & 0 & \multicolumn{1}{|c}{1}
\end{array}\right)
\qquad
\left(\begin{array}{c|ccccc}
+ & 1 & 1 & 0 & 0 & 0 \\
x_1 & 1 & 0 & 0 & 0 & 0 \\ \cline{3-6}
\times & 0 & \multicolumn{1}{|c}{1} & 1 & 1 & 0 \\
x_1 & 0 & \multicolumn{1}{|c}{0} & 1 & 0 & 0 \\
\cos & 0 & \multicolumn{1}{|c}{0} & 0 & 1 & 1 \\
x_2 & 0 & \multicolumn{1}{|c}{0} & 0 & 0 & 1
\end{array}\right)
$$

Для результирующего выражения:

$$
\left(\begin{array}{c|cccccc}
+ & 1 & 0 & \multicolumn{1}{c|}{1} & 0 & 0 & 0 \\
\sin & 1 & 1 & \multicolumn{1}{c|}{0} & 0 & 0 & 0 \\
x_1 & 0 & 1 & \multicolumn{1}{c|}{0} & 0 & 0 & 0 \\ \cline{2-7}
\times & 0 & 0 & \multicolumn{1}{|c}{1} & 1 & 1 & 0 \\
x_1 & 0 & 0 & \multicolumn{1}{|c}{0} & 1 & 0 & 0 \\
\cos & 0 & 0 & \multicolumn{1}{|c}{0} & 0 & 1 & 1 \\
x_2 & 0 & 0 & \multicolumn{1}{|c}{0} & 0 & 0 & 1
\end{array}\right)
$$

Двухточечный кроссовер можно выразить как последовательное применение двух одноточечных кроссоверов: первый обменивает меньшие поддеревья, затем второй — большие.

4. Замечания

- При использовании только функций $R \to R$ и $R \times R \to R$ очень легко считать сложность модели по ее матрице, но, в то же время, описание даже банальной однослойной нейросети становится довольно громоздким.
- Если же, наоборот, разрешить функции из пространства произвольной размерности, то (по крайней мере, для моделей типа двухсловной нейросети) придется в G иметь варианты операции суммирования, действующие из $R^n \mid \forall n \in \mathbb{N}$, что, по факту, делает G по крайней мере счетным. Либо надо делать для операции суммирования исключение, а потребность в таком частном случае, скорее всего, означает о неправильно выбранных примитивах.
- Можно попробовать обойтись без введения пополненного множества, а вместо этого договориться записывать в соответствующий вектор не $\overline{g}_i$, а сразу x_i, но это по большей части вопрос обозначений.

Список литературы

[1] Alfred V. Aho, Ravi Sethi, and Jeffrey D. Ullman. *Compilers, Principles, Techniques, and Tools.* Addison-Wesley, 1986.

Московский физико-технический институт, ФУПМ, каф. «Интеллектуальные системы»