УДК 519.7

Л.Г. ПЕТРОВА

МОДЕЛИРОВАНИЕ ДИНАМИЧЕСКИХ АСПЕКТОВ РАСПРЕДЕЛЕННОГО ПРЕДСТАВЛЕНИЯ ЗНАНИЙ

Рассматриваются вопросы моделирования динамических аспектов распределенного представления знаний. Распределенное представление является основой многоуровневых архитектур, используемых при решении задач глубокого обучения. Предлагается модель процессов вычисления целевой функции для распределенного представления знаний. Модель задает множество состояний, составляющих представления знаний, допустимые отношения между ними, а также наборы допустимых переходов между состояниями. Модель обеспечивает возможность верификации процессов параллельных вычислений целевой функции для распределенного представления знаний.

Введение

Одной из важных проблем, которые возникают при формировании представления знаний для высоковариативных зависимостей посредством машинного обучения, является обобщающая способность. Данная проблема связана со способностью алгоритма обучения обобщать исходные данные, не входящие в обучающую выборку. Для обучающей выборки алгоритм обучения позволяет строить правильные зависимости, однако для тех объектов, которые в эту выборку не входят, вероятность ошибки значительно увеличивается.

В работах [1 ,2] показано, что «плоские» архитектуры представления знаний имеют ряд недостатков, не позволяющих достичь требуемого уровня генерализации при решении задач, у которых число состояний значительно превышает число входных данных. В результате при применении таких архитектур возникает проблема переобучения. Поэтому в настоящее время активно развиваются так называемые глубокие архитектуры[1, 3-5], у которых закономерности распределяются по нескольким уровням, образуя иерархию абстрактных концепций или понятий. Такие архитектуры основаны на использовании распределенного представления знаний [6]. При практической реализации глубокие архитектуры требуют значительных вычислительных ресурсов, что и определяет актуальность проблемы обоснованного выбора количества элементов для представления знаний, а также распределения этих элементов по уровням представления.

Постановка задачи

Проблематика построения многоуровневых архитектур требует рассмотрения не только статических аспектов представления знаний (количество уровней, взаимосвязь элементов и т.п.), но и процессов функционирования распределенного представления при вычислении целевой функции. Последние являются процессами параллельных вычислений, что позволяет использовать соответствующие методы для формирования ограничений на архитектуру представления знаний.

Задача данного исследования заключается в разработке модели, позволяющей оценить темпоральные аспекты процесса вычисления целевой функции в рамках заданной архитектуры.

Модель процесса вычисления целевой функции

Для обоснования предлагаемого подхода к моделированию динамических аспектов распределенного представления знаний кратко уточним проблематику машинного обучения и связанные с ней особенности построения распределенного описания знаний с учетом свойств многоуровневых архитектур.

Машинное обучение представляет собой раздел искусственного интеллекта, связанный с построением обучающихся алгоритмов. Общая идея машинного обучения заключается в том, что по имеющемуся описанию прецедентов (т.е. по частным случаям) необходимо выявить такие закономерности, которые характерны не только для имеющегося множества прецедентов, но и для тех прецедентов, которые не вошли в выборку. Оценивание полученной модели выполняется с помощью заданной функции оценки качества[7, 8].

Целью машинного обучения (или обучения по прецедентам) является получение функции F, определяющей зависимости между множествами исходных данных (объектов, ситуаций, процессов) X и допустимых результатов (ответов, управляющих воздействий и т.п.) Y.

Целевая функция $Y = F(X)$ строится на основе известного подмножества пар $v_i = (x_i, y_i) \in V, x_i \in X^V, y_i \in Y^V, i = \overline{1,I}, X^V \subset X, Y^V \subset Y \mid \forall v_i \; x_i \notin (X - X^V)$, называемых прецедентами. Совокупность прецедентов составляет обучающую выборку W.

Традиционно для формализации прецедентов используется признаковое описание. Под признаком p понимают значение некоторой характеристики исходного объекта x_i. Формально признак можно представить через отображение, которое связывает обозначение j – признака i – исходного объекта $x_{i,j}^p$ с множеством его допустимых значений D^p:

$$x_{i,j}^p \to D^p \mid \forall p \exists D^p, D^p = \{d^p\}. \quad (1)$$

Тогда исходный объект, процесс, ситуация в системе ИИ отождествляется с его признаковым описанием:

$$x_i \equiv \{x_{i,j}^p\} \mid \forall x_{i,j}^p \exists d_{i,j}^p : \{d_{i,j}^p\} \subset (\underset{p}{\times} D^p). \quad (2)$$

Обучающая выборка, которая содержит набор исходных объектов в форме, доступной части признаков объектов x_i, описывается матрицей D размерности: $D = \left(d_{i,j}^p\right)_{i=1,j=1}^{I,J}$. Каждая строка матрицы содержит J значений признаков i – объекта из обучающей выборки.

Значение признака может отображать реальные характеристики объекта (как прямые, так и косвенные), а также некоторые абстрактные концепции, которые позволяют представить знания о его свойствах. В первом случае значение признака может быть измерено или вычислено (т.е. преобразовано в результате предварительной обработки результатов измерений). Основные типы значений: бинарные; конечное (упорядоченное) множество; количественные; изображения; графы, выборки из базы данных; функции. Дальнейшие рассуждения приводятся для признаков бинарного типа. Во втором случае для получения системы абстрактных концепций используются методы глубокого обучения, поскольку количество внутренних состояний S для полученного представления знаний будет значительно превышать количество элементов матрицы D. Плоские архитектуры в данном случае позволяют аппроксимировать целевую функцию, однако не решают проблемы генерализации.

Методология многоуровневого обучения связана с построением иерархии концепций для описания мира в системах искусственного интеллекта. В основе такой иерархии лежат наиболее простые концепции, знания более высокого уровня определяются в терминах их связей с концепциями нижележащих уровней. Данный подход позволяет системе искусственного интеллекта вывести более сложные знания из простых концепций, без формальной спецификации этих знаний человеком. В результате получается многоуровневая иерархия знаний, которая обеспечивает формализацию целевой функции F.

Задача обучения по прецедентам заключается в построении решающей функции α, которая бы на основе обучающей выборки W аппроксимировала функцию F на всем множестве исходных данных X. С учетом требований практической реализации эта функция должна представляться алгоритмом.

Основная проблема обучения, как уже отмечалось, связана с качеством работы алгоритма для новых исходных данных, не входивших в выборку W. Качество работы определяется через функцию потерь, которая принимает истинное значение только в том случае, если алгоритм α позволяет получить истинное значение для пары v_i. Очевидно, что для проверки истинности пары используется целевая функция F. Функция потерь принимает истинное значение в том случае, если значение y_i^*, полученное для исходного набора признаков x_i с помощью целевой функции, совпадает со значением y_i из пары v_i, либо же расхождение между ними не превышает заданного порога.

Рассмотренная проблематика машинного обучения показывает, что ограничения в применении типичных архитектур представления знаний для вариативных функций связаны в первую очередь с недостаточной глубиной архитектуры. Однако увеличение глубины архитектуры значительно повышает затраты при вычислении целевой функции.

Отметим, что возможности «плоского» представления ограничивает также принцип локальности функции оценивания[1].

Данная работа посвящена реализации подхода к моделированию динамических аспектов архитектуры, что позволяет обосновать выбор архитектуры достаточной глубины.

Рассмотрим возможности оценки глубины и сложности архитектуры в рамках предлагаемого подхода. Для определения глубины архитектуры схему вычисления функции F целесообразно представить в виде набора узлов обработки V^F и взаимосвязей между ними E^F. Тогда процесс получения функции F представляется ациклическим графом G^F:

$$G^F = (V^F = \{v_i\}, E^F = \{e_{i,j} \mid i \neq j\}). \quad (3)$$

Формализация процесса решения в виде графа G^F позволяет рассматривать его как граф потока управления и в этом качестве использовать для оценки глубины архитектуры. Потоковые графы широко применяются в области анализа программного обеспечения, в частности для оценки путей тестирования программного модуля или программной системы в целом[9, 10].

Адаптируем семантику потокового графа применительно к задаче оценивания глубины архитектуры. Граф G^F отражает управляющие потоки при вычислении функции F. Каждая вершина v_i графа соответствует одному атомарному вычислительному элементу архитектуры. Следует подчеркнуть, что в общем случае один вычислительный элемент может выполнять несколько последовательных действий, однако согласно данному подходу мы будем рассматривать это множество действий как одну процедуру и отражать его одной вершиной графа G^F. Дуги E^F отражают последовательность обработки при расчете целевой функции. Вершины графа могут быть двух типов: с одной или несколькими выходящими из вершины дугами. В первом случае в узле, моделируемом вершиной графа, выполняется линейная обработка. Во втором случае выбор подмножества последующих узлов зависит от выполнения условий, вычисляемых в текущей вершине. Для упрощения одну вершину второго типа можно разделить на несколько бинарных вершин, из которых выходят только две дуги. При таком подходе для определения глубины архитектуры выполняются следующие шаги:

1) выделение независимых путей π_k в потоковом графе в соответствии с условием:

$$\pi_k = \{(\mathrm{v}_i, \mathrm{e}_{i,j})\} \mid \forall k \ \pi_k \Delta \pi_{k+1} \neq \varnothing, \mathrm{k} = \overline{1, K-1}, \mathrm{v}_i, \mathrm{v}_j \in V^F, \mathrm{e}_{i,j} \in E^F, \quad (4)$$

2) выбор пути π_k^* с максимальным количеством вершин: $\pi_k^* : \forall l \neq \mathrm{k} \ \mid \pi_k^* \cap V^F \mid \geq \mid \pi_l \cap V^F \mid$.

На первом шаге, как видно из условия (4), выявляются пути ,которые отличаются хотя бы одной дугой. На втором шаге находится путь с максимальным числом обрабатывающих элементов.

Применение потокового графа для оценки глубины архитектуры позволяет оценить сложность распределенного представления знаний с учетом динамического аспекта. Действительно, при использовании потокового графа мы можем применять показатель O_π цикломатической сложности архитектуры, чтобы получить верхнюю оценку количества независимых путей для расчета целевой функции. Оценка цикломатической сложности представлена в работе [11]. Данная метрика имеет вид $O_\pi = \mid E^F \mid - \mid V^F \mid + 2$ для единственной архитектуры и $O_\pi = \mid E^F \mid - \mid V^F \mid + 2C$ – для набора из C архитектур. Количество независимых путей определяет и количество составляющих представления знаний. Так, при добавлении нового пути в алгоритм вычисления целевой функции мы добавляем, как минимум, дугу $\mathrm{e}_{i,j}$, отображающую новую взаимосвязь между узлами вычислений. Иными словами, способы вычисления целевой функции зависят от порядка взаимодействия элементов распределенного представления знаний. Приведенные оценки позволяют вычис-

лить сложность архитектуры представления знаний в том случае, если нам известен граф G^F. Поэтому следующий шаг моделирования динамических аспектов представления знаний заключается в моделировании последовательности взаимодействия атомарных вычислительных элементов, отображаемых вершинами графа G^F. Предлагаемый подход основан на формализации переходов между вычислительными элементами. После каждого такого перехода изменяется состояние процесса вычисления целевой функции. Обобщая, можем утверждать, что модель представления знаний должна отображать как статические зависимости между вычислительными элементами, так и динамику изменения состояний при расчете F. Зависимости первого вида определяются выбранным математическим аппаратом для представления знаний: нейронные сети, деревья решений, ансамбли правил и т.п. Зависимости второго вида отражают темпоральные взаимосвязи между состояниями процесса вычислений, обеспечивающие в том числе параллельные вычисления и потому могут быть представлены на основе структур Крипке [12].

Модель представления знаний о процессе вычисления целевой функции определяется следующим образом:

$$M = (S_0, S = \{s_i\}, E(s_i, s_j) \mid s_i, s_j \in S, I, R), \qquad (5)$$

где S - множество состояний элементов $v_i \in V^F$, реализующих отдельные процедуры по вычислению целевой функции F; S_0 - множество начальных состояний; E - отношение переходов между состояниями s_i элементов v_i; $I(s)$ - функция разметки (интерпретации), которая задает множество атомарных высказываний, истинных в состоянии s логических формул в модели M; R - ограничения реализуемости (справедливости).

Каждый элемент v_i в рамках модели M считается атомарным, хотя и может выполнять некий алгоритм вычисления функции F или ее составляющих. Набор состояний S представляет собой множество оценок в форме атомарных высказываний об элементах v_i: $s_j(v_i)$ может быть либо истинным, либо ложным. Функция интерпретации задает истинность атомарных логических выражений для состояний s_i модели M. Ограничения реализуемости задают требования к внешней среде[13], а также позволяют отсечь заведомо нереалистичные действия(вычисления) при вычислении целевой функции[14]. В соответствии с предлагаемым подходом ограничения реализуемости позволяют задать подмножества недопустимых входных объектов.

Множество E обладает следующими свойствами. Во-первых, потоковый граф G^F является направленным, поскольку задает последовательность вычисления целевой функции. Поэтому данное множество содержит только отношение следования для вычислительных элементов. Во-вторых, при формировании представления знаний следует исключить циклы (граф должен быть ацикличным). Действительно, в общем случае не следует отрицать возможность использования циклов при вычислении целевой функции. Однако эти циклы не отображаются в модели по следующим причинам:

1)Циклические вычисления возможны в рамках вычислительного элемента. Однако, как уже отмечалось ранее, такой элемент считается атомарным в модели и потому все вычисления в данном случае «прозрачны» с точки зрения представления знаний.

2)Циклическое выполнение может быть реализовано с использованием двух вычислительных элементов. Первый проверяет условие повторения цикла, а второй – выполняет действия цикла. В таком случае мы остаемся в рамках приведенной выше модели.

Модель (4) позволяет выявить допустимые потоки управления процесса вычисления F с отображением темпоральных свойств посредством учета последовательности смены состояний.

Распределенная модель процесса вычисления целевой функции представляет собой параллельную композицию M^* моделей M и определяется следующим образом: $M^* = M_1 \| M_2 .... M_i .... \| M_I \|$, при выполнении для любой пары моделей $(M_i \| M_j) \in M^*$ следующих ограничений $S^* = \{(s_i, s_j)\}$ при условии идентичности пересечений функций разметки

для состояний s_i, s_j с множеством атомарных высказываний. Множество начальных состояний композиции объединяет все комбинации множеств начальных состояний исходной пары при условии, что эти комбинации входят в состав множества состояний композиции. Функция интерпретации композиции объединяет функции интерпретации исходной пары моделей. Отношения упорядоченности для пар нескольких состояний композиции задаются только при наличии таких отношений в исходных моделях. Реализуемость пары параллельных путей возможна только в случае реализуемости каждого из пути пары.

Выводы

Предложен подход к моделированию динамических аспектов распределенного представления знаний, которое является основой многоуровневых архитектур, используемых при решении задач глубокого обучения. Подход предполагает моделирование состояний процесса вычисления целевой функции.

Предложена модель процесса вычисления целевой функции для распределенного представления знаний, которая задает множество состояний атомарных элементов представления знаний (вычисления целевой функции), допустимые отношения между элементами, а также множество атомарных логических высказываний, истинных для соответствующих состояний элементов представления знаний. Модель обеспечивает возможность верификации процессов параллельных вычислений целевой функции для распределенного представления знаний.

Список литературы: 1. *Bengio Y.* Learning deep architectures for AI. // Foundations and Trends in Machine Learning. 2009. № 2(1). P. 1–127. **2.** *Anthes G.* Deep learning comes of age// Communications of the Association for Computing Machinery (ACM). 2013. № 56(6). P. 13–15. **3.** *Bengio Y.* Deep learning of representations for unsupervised and transfer learning// Journal of Machine Learning Research Workshop and Conference Proceedings. 2012. № 27. P. 17–37. **4.** Deng L. An overview of deep-structured learning for information processing// Proceedings of Asian-Pacific Signal & Information Processing Annual Summit and Conference (APSIPA-ASC). October 2011. **5.** *Deng L.* A tutorial survey of architectures, algorithms, and applications for deep learning// Asian-Pacific Signal & Information Processing Association Transactions on Signal and Information Processing. 2013. **6.** *Handbook* of Knowledge Representation// Edited by F. van Harmelen, V. Lifschitz, B. Porter/Foundations of artificial intelligence. 2008. 1004 p. **7.** *Загоруйко Н. Г.* Прикладные методы анализа данных и знаний. Новосибирск: ИМ СО РАН, 1999. 270 с. **8.** *Hastie T.* The Elements of Statistical Learning / T. Hastie, R. Tibshirani, J. Friedman. Springer. 2001. 739 p. **9.** *Reese T.* Applications of Boolean matrices to the analysis of flow diagrams/ IRE-AIEE-ACM '59 (Eastern) Papers presented at the December 1-3, 1959. P. 133–138. **10.** *Abrahams J. R.* Chapter 1: Elements of a flow graph// J. R. Abrahams, G. P. / Coverley Signal flow analysis. 2014. P. 1-20. **11.** *McCabe T. J.* A Complexity Measure/ IEEE Transactions on Software Engineering. Apr 1976. Vol. 2. № 4. P. 308-320. **12.** *Kripke S. A.* Semantical considerations on modal logic /Acta Philosophica Fennica. № 16. 1963. P. 83 – 94. **13.** *Clarke E. M.* Model Checking// E. M. Clarke, O. Grumberg, D. Peled/ The MIT Press. 1999. 330p. **14.** *Queille J. P.* Fairness and related properties in transition systems - a temporal logic to deal with fairness// J. P. Queille, J. Sifakis /Acta Inf. 1983. № 19. P. 195–220.

Поступила в редколлегию 11.06.2015

Петрова Лариса Григорьевна, доцент кафедры информационно-коммуникационных технологий. Сумской областной институт последипломного педагогического образования. Научные интересы: методы машинного обучения. Адрес: Украина, 40007, Сумы, ул. Римського-Корсакова, 5, тел. (0542) 33-40-67.