А.А. ПЧЕЛКИН
Рижский технический университет, Латвия
arturp@inbox.lv

НЕЙРОПОДОБНАЯ АРХИТЕКТУРА ДЛЯ ИЕРАРХИЧЕСКОГО УПРАВЛЕНИЯ АВТОНОМНЫМ АДАПТИВНЫМ АГЕНТОМ

Аннотация

Предлагается нейроподобная архитектура для задачи «обучение с подкреплением» (reinforcement learning). Предлагаемая архитектура состоит из нескольких слоев сети достижимости, в которой нейроны репрезентируют различные пересекающихся классы состояний среды, а направление связи – способность агента в результате целенаправленного поведения достигать из одного класса состояний среды другой класс.

Введение. *Обучение с подкреплением* [1-3] исследует вопрос, как построить алгоритм обучения автономного агента, который позволил бы агенту научиться достигать поставленные перед ним цели только через эксперимент с окружающей средой без априорных знаний. Для агента определено небольшое множество возможных примитивных действий и набор сенсоров для получения информации о текущем состоянии среды. В статье предполагается, что каждый из сенсоров является бинарным, а цель задается как множество сенсоров, репрезентирующее «желаемое» состояние.

В обучении с подкреплением известно много примеров алгоритмов обучения агента, которые демонстрируют реальную способность к адаптации в искусственных средах [4-7]. Тем не менее, в большинстве случаев агент решает не слишком интеллектуальные задачи (при условии, что агент не получает априорных знаний о среде до обучения). Все эти задачи, как правило, заключаются в поиске некоторой цели в лабиринте, например [4-7]. В тоже время, можно предложить очень простые, но нетривиальные для решения задачи, например, «головоломки» и им подобные задачи – такие, как манипуляция множеством определенным образом взаимосвязанных объектов. Либо случаи, когда агент должен научиться оперировать определенными «орудиями труда» или «инструментами» для достижения поставленных перед ним целей.

Любую из вышеописанных задач можно разбить на две подзадачи: (1) извлечение знаний из среды (т.е. модель, однозначно описывающая

поведение среды), и (2) алгоритм, использующий эти знания, (т.е. модель, достижения целей).

В различных областях искусственного интеллекта много усилий направляется на создание модели среды через эксперимент, например, исследуется возможность построения скрытых моделей среды в парадигме PAC (Probably Approximatly Correct) [8]. Тем не менее, возвращаясь к анализу задач типа «головоломка», основная сложность заключается в формировании умения использовать модель среды для достижения цели, т.к. даже при наличии однозначной модели среды у агента может не хватить времени для перебора всех возможных сценариев достижения цели. В то же время, пространство состояний должно быть очень симметричным (в смысле многократного повторения фрагментов), так как общее количество состояний среды очень велико, а набор правил, описывающих поведение среды, мал. Поэтому предполагается, что, если агенту удалось бы «понять» симметрию среды, то это помогло бы значительно сузить перебор сценариев достижения цели.

Ключевая идея. Вышеописанная проблема послужила мотивацией для разработки предлагаемой нейроподобной архитектуры. Ключевая идея состоит в том, чтобы агент искал «полезные» пересекающиеся классы состояний среды и исследовал возможность достигать одного класса состояний из другого в результате целенаправленного поведения. Для краткости будем называть такую сеть, состоящую из нейронов, репрезентирующих классы, и связей, репрезентирующих возможность достигать эти классы, *сетью достижимости*. Суть процесс самоорганизации сети достижимости заключаться в следующем: сформированные классы позволяют накапливать статистику о возможностях достижения, а накопленная статистика делает возможным качественный скачок – менять структуру классов с целью поиска классов более полезных для точности прогноза о возможности достижения других классов. Будем считать, что каждый класс определяется некоторой булевой функцией от текущего состояния всех сенсоров агента.

Сеть достижимости в определенной степени будет обладать свойством транзитивности (использование в данном контексте понятия транзитивности является не столько формальным, сколько интуитивным), т.е. будет иметь много транзитивных связей достижимости. Например, если из класса А будет иметься некоторый путь в класс Б, то в данной сети достижимости, скорее всего, будет также иметься и «транзитивная связь» из А в Б. Подобные транзитивные связи смогут включать в себя длинную последовательность примитивных действий в среде. Это даст

возможность оценивать полезность классов, используя в качестве критерия возможность, ассоциировать с этими классами точный многошаговый прогноз для выявления многошаговых закономерностей. Предполагается, что обладание способностью распознавать некоторое множество классов, описывающих инвариантные состояния среды, а также способностью выполнения точных многошаговых прогнозов о возможности достижения из одного класса другого, даст возможность агенту сократить перебор при планировании способа достижения поставленной перед ним цели.

Архитектура. Анализ возможности создания вышеописанной сети выявил следующую проблему. Предположим, используя сеть достижимости, агенту удалось построить план, содержащий транзитивные связи, для достижения некоторой цели. Возникает вопрос: как в такой ситуации использовать полученный план? Для возможности использования такого плана агенту придется хранить в памяти всю информации о том, из каких других связей состоит каждая из транзитивных связей достижимости. Поэтому вместо такого нерационального решения предлагается использовать многослойную нейроподобную архитектуру (рис. 1).

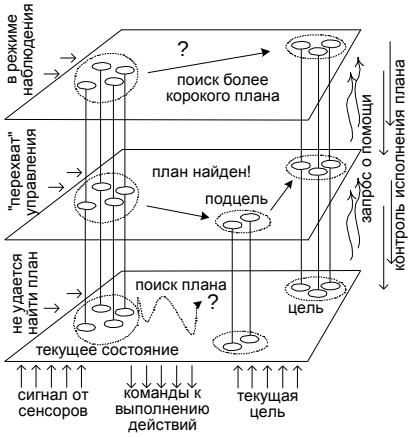

Рис. 1. Архитектура

Предлагаемая архитектура состоит из нескольких слоев нейронов. На каждом уровне, т.е. слое, расположена сеть достижимости. На рис. 1 нейроны условно обозначены кружками, каждый из которых соответствует некоторому классу состояний среды агента. Текущее состояние среды отображается через сенсоры в нижний слой как некоторое множество нейронов S, репрезентирующих эти сенсоры. Аналогично отображается цель как некоторое множество нейронов G, т.е. описание «желаемого» состояния. Цель G считается достигнутой, если $G \subset S$. Таким образом, задача внутри слоя представлена парой $<S, G>$. Для решения этой задачи агент должен найти план p, состоящий из последовательности блоков $<b_1, b_2, b_3, ..., b_k>$, где k – длина плана, а

каждый блок b_i представлен парой $<S_i, G_i>$, кроме того, $S_1 = S$, $G_k = G$, $S_i = G_{i-1}$ для $2 \le i \le k$. Каждый из блоков должен отвечать дополнительным критериям, которые будут описаны в разделе «генерация плана».

Общая схема работы сети. Каждый раз, когда среда задает агенту цель, вначале эта цель отображается в нижний уровень, и нижний уровень начинает поиск плана решения этой задачи (см. рис. 1). Если нижний уровень находит план, то сразу же начинает его выполнение, выполняя последовательность блоков. Выполнение блока $<S_i, G_i>$ состоит в ожидании исполнения условия $G_i \subset S(t)$, где $S(t)$ – текущее состояние. Если в G_i имеются сенсоры действия, то агент выполняет все соответствующие примитивные действия в среде. (Множество возможных действий, как и множество сенсоров, считается зафиксированным конструктором агента до начала обучения, и для каждого действия имеется соответствующий ему сенсор, который сообщает о выполнении данного действия.) Если в течение некоторого короткого интервала времени (1 шаг в дискретном времени) указанное условие не выполняется, то выполнение плана прекращается досрочно, при этом блок считается невыполненным, план – ошибочным, а цель – не достигнутой. После выполнения каждого блока в случае удачи или неудачи происходит перерасчет вероятностных прогнозов, хранящихся в соответствующем уровне.

Если нижнему уровню не удается найти план, то этот уровень начинает «просить» о помощи в поиске плана вышестоящий уровень. Для краткости, будем называть вышестоящий уровень «надуровнем», а нижестоящий уровень «подуровнем». При этом решаемая задача $<S, G>$ отображается в надуровень через вертикальные связи (см. рис. 1). Этот процесс продолжается рекурсивно, пока один из уровней не сумеет найти план. Как только это произойдет, соответствующий уровень «перехватит» все управление процессом на себя и начнет выполнять найденный план – блок за блоком, аналогично нижнему уровню. При этом выполнение блоков $<S_i, G_i>$ состоит из трех шагов: отобразить через вертикальные связи задачу $<S_i, G_i>$ в подуровень, передать управление подуровню, и ожидать достижения цели G_i подуровнем.

Связи между нейронами. Каждый нейрон имеет возможность образовывать связи только с нейронами в пределах уровня за исключением вертикальных связей. Нейрон в предлагаемой архитектуре имеет 3 типа связей: (1) структурные, (2) связи достижимости и (3) вспомогательные.

Структурные связи предопределяют класс состояний среды, которому внутри сети будет соответствовать данный нейрон, точнее говоря то, как этот класс логически выражен через другие классы. Как правило, это будет некоторая простая булева функция, например, конъюнкция или дизъюнкция от нескольких других нейронов. Тип булевой функции хранится в самом нейроне, а перечень переменных от каких зависит функция, – в структурных связях. Информацию о функции и конфигурации структурных связей будем называть структурой нейрона. Нейроны нижнего уровня, репрезентирующие сенсоры, ассоциированы с элементарными классами, а другие нейроны могут быть ассоциированы с производными классами. Вертикальные связи являются структурными, но при этом связывают нейроны разных уровней, где нейрон более высокого уровня является «клоном» нейрона с более низкого уровня и репрезентирует тот же класс состояний среды. Текущее (или «воображаемое» при планировании) состояние среды, представленное некоторым множеством нейронов, должно постоянно и автоматически достраиваться до логически завершенного по структурным связям в пределах уровня. Если при планировании возникает некоторое «воображаемое» состояние, которое невозможно автоматически восстановить до логически завершенного, то такое состояние и репрезентирующее его множество нейронов будем называть *противоречивым*.

Связи достижимости хранят вероятностный прогноз о возможности достижения класса одного нейрона из класса состояний другого нейрона в результате целенаправленного поведения. Каждый раз и на каждом уровне, после выполнения очередного блока $<S_i, G_i>$, происходит корректировка вероятностного прогноза, хранящегося в соответствующих связях достижимости, т.е. для каждой пары нейронов $<s, g>$, где $s \in S_i$, $g \in G_i$, происходит пересчет вероятности, хранящейся в связи $s \rightarrow g$ (в случае успеха увеличивается, а неудачи – уменьшается). Если опыт успешного достижения целей хранится в связях достижимости, то *опыт неудач* ассоциируется с самим нейроном. С каждым нейроном g ассоциируется вероятностный прогноз неудачи при попытке выполнить любой блок $<S_i, G_i>$, содержащий в правой части g (т.е. $g \in G_i$).

Вспомогательные связи используются только при генерации плана. Эти связи описаны в разделе «генерация плана».

Формирование структуры. *Для формирования структурных* связей оценивается *полезность* нейрона для всей системы в целом. Полезность нейрона определяется как вероятность дать самый точный,

ассоциированный с нейроном прогноз. С нейроном ассоциируются разные типы прогнозов, например, исходящие связи достижимости, опыт неудач. Каждый прогноз, используемый системой, как правило, делается множеством нейронов, но вознаграждение в виде увеличения его полезности для системы получает только тот нейрон, с которым был ассоциирован самый точный (имеющий наибольшее значение вероятности) и правильный прогноз (подтвержденный последующим экспериментом). Как и в предыдущих случаях, эта вероятность, т.е. полезность, оценивается только статистически, корректируется динамически после получения новых фактов.

Используя данный критерий полезности, все нейроны делятся на две группы: *полезные*, полезность которых превышает некоторую константу, и *бесполезные*. Структура полезных нейронов фиксируется, т.е. структурные связи этих нейронов не могут меняться. Бесполезные нейроны, которые не используются другими нейронами, т.е. не имеют исходящих структурных связей, могут меняться в каждый момент времени с некоторой небольшой вероятностью. В отдельных случаях эта вероятность резко возрастает, например, при ориентировочно-исследовательской деятельности, когда система пытается запомнить текущее состояние. Такие ситуации могут возникать, когда система получит опровержение некоторому очень сильному прогнозу (с большой вероятностью). Меняя свою структуру, нейрон может образовывать структурные связи только с полезными нейронами.

Бесполезные нейроны меняются для двух основных целей: (1) улучшение точности прогнозов, например, при образовании конъюнкции от нескольких других нейронов, и (2) получение более полезных, более универсальных нейронов путем увеличения частоты их использования системой, например, при образовании дизъюнкций. После изменения структуры нейрона обнуляются все ассоциированные с ним прогнозы, и нейрон теряет способность меняться на некоторый тестовый период времени, достаточный для приобретения новой статистики прогнозов и полезности.

Генерация плана. Для решения задачи $<S, G>$ на некотором уровне должен быть найден план p, состоящий из последовательности блоков $<b_1, b_2, b_3, ..., b_k>$, где каждый блок b_i является парой $<S_i, G_i>$, и $S_1 = S$, $G_k = G$, $S_i = G_{i-1}$ для $2 \le i \le k$. Кроме того, для каждого блока оценивается оптимистичность его выполнения, и ищется наиболее короткий план, состоящий только из оптимистичных блоков. Блок $<S_i, G_i>$ считается оптимистичным при выполнении нескольких условий:

(1) для каждого $g \in G_i$ либо должна существовать сильная связь достижимости $s \to g$, где $s \in S_i$, либо g должен логически следовать по структурным связям из наличия в G_i других нейронов, которые удовлетворяют первому условию;

(2) для каждого $g \in G_i$, прогноз неудачи не должен быть сильным;

(3) состояние G_i должно быть непротиворечивым, т.е. не должно противоречить структурным связям.

При прочих равных условиях, план не должен содержать нейроны в блоках, чье присутствие не улучшает прогноз успешности плана.

В предлагаемой архитектуре генерация плана рассматривается как динамическая активация сети, которая состоит во внутреннем проигрывании различных «воображаемых» сценариев достижения цели G из начального состояния S. Вначале S восстанавливается до логической завершенности по структурным связям. Потом сеть работает в два такта: формирование текущего блока, и переход к следующему за счет копирования G_i на S_{i+1}. При формировании каждого блока S_i известно, и надо случайным образом (который может быть значительно ограничен с помощью вспомогательных связей для сужения полного перебора различных вариантов сценариев) сгенерировать G_i такое, которое соответствовало бы вышеописанным критериям. Множество G_i можно получать из S_i в результате небольшой его модификации, т.е. добавлением и/или удалением нескольких нейронов. Поэтому задача сводится к генерации множеств: $Y_i = G_i - S_i$ (добавляемых нейронов) и $X_i = S_i - G_i$ (удаляемых нейронов). Обозначим через $H(S_i)$ множество всех нейронов, достижимых из S_i по сильным связям достижимости, т.е. удовлетворяющих первому условию оптимистичности блока. Множество Y_i формируется случайным образом только из множества $H(S_i) - S_i$. Если возникают проблемы с выполнением второго или третьего условия, то формируется также X_i из S_i с целью удовлетворения этих условий.

После генерации нескольких планов выбирается самый короткий план для выполнения, и обновляются вспомогательные связи. Возбуждающая вспомогательная связь $s \to y$ хранит условную вероятность, что $y \in Y_i$ при условии $s \in S_i$, $y \in H(S_i) - S_i$. Аналогично, тормозящая связь $y \to x$ – вероятность, что $x \in X_i$ при условии $y \in Y_i$, $x \in S_i$. Обе вероятности рассчитываются только на планах, принятых к выполнению. Впоследствии возбуждающие связи можно будет использовать для формирования Y_i, а тормозящие – для формирования X_i.

Образование надуровней. Как уже отмечалось, генерация плана происходит путем перебора различных вариантов достижения цели. Пока сильных связей достижимости будет немного, этот перебор не будет требовать слишком больших вычислительных затрат, но вместе с накоплением опыта, отражающегося в связях достижимости и в прогнозе о неудаче, вычислительные затраты могут значительно возрасти. Предполагается, что с помощью вспомогательных связей можно будет значительно сузить перебор, тем не менее, может возникнуть ситуация, когда этого тоже будет недостаточно. Эту проблему предлагается решать через образование более коротких планов в надуровне.

Каждый раз, когда какому-то уровню удастся сгенерировать некоторый план, затратив при этом достаточно много времени (измеряемого, например, количеством сгенерированных сценариев), сгенерированный план будет разбит на 2-3 части примерно одинаковой сложности для воспроизведения, а также будет сформирован план для надуровня, обеспечивающий последовательный вызов этих частей. Кроме того, будут добавлены вертикальные связи (если таких не окажется) и нейроны «клоны» для обеспечения связи между планом надуровня и частями первоначально сгенерированного плана. После этого управление будет передано надуровню, который начнет выполнять свой план. Таким образом, план, сформированный одним уровнем, будет выполняться с использованием его надуровня.

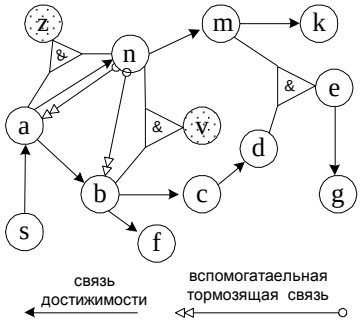

Рис. 2. Пример сети достижимости

Для иллюстрации вышеописанной идеи предлагается рассмотреть пример формирования надуровня (см. рис. 2). Требуется сгенерировать план решения задачи $<\{s\}, \{g\}>$. Нейроны *z*, *v*, *e* логически зависят от других нейронов, кроме того, нейроны *z*, *v* имеют сильный прогноз неудачи, – поэтому ранее уже сформировались вспомогательные тормозящие связи. Возможный сценарий решения задачи может выглядеть так:

{s} – {s, a} – {s, a, b} – {s, n, c} – {s, c, n, m, d, e} – {s, c, n, m, d, e, g}.

Сложность генерации такого плана состоит в том, что нейрон *n* должен появиться строго после второго появления нейрона *a*. Если нейрон *n* появиться сразу после первого появления нейрона *a*, то цель $\{g\}$ не будет

достигнута в «воображаемом» сценарии. Тем не менее, если этот план разбить на две части

$$\{s\} - \{s, a\} - \{s, a, b\} - \{s, n, c\}$$

и

$$\{s, n, c\} - \{s, c, n, m, d, e\} - \{s, c, n, m, d, e, g\},$$

то в дальнейшей будет значительно проще сгенерировать каждую часть в отдельности. При этом план для надуровня будет

$$\{s\} - \{s, n, c\} - \{s, c, n, m, d, e, g\}.$$

Заключение. В статье предложена архитектура нейроподобной многослойной сети для иерархического управления автономным адаптивным агентом. Ключевая идея состояла в использовании сети достижимости, которая моделирует способность агента достигать различные цели в среде.

Следует отметить, что идея разработки предложенной в данной статье архитектуры сети возникла на основании теории функциональных систем, изложенной в работах П.К. Анохина [9, 10], а также последующих публикациях Е.Е. Витяева [11].

Список литературы

1. Kaelbling, L.P., Littman, L.M., Moore, A.W. Reinforcement learning: a survey // Journal of Artificial Intelligence Research. Vol. 4. 1996. P.237-285.

2. Sutton, R.S., Barto, A.G. Reinforcement learning: An introduction // Cambridge, MA: MIT Press, 1998.

3. Mitchel, T.H. Machine learning // The McGraw-Hill Companies. Inc., 1999.

4. McCallum, R.A. Reinforcement learning with selective perception and hidden state (Ph.D. dissertation) // Department of Computer Science, University of Rochester, Rochester, NY, 1995.

5. Wiering, M., Schmidhuber, J. HQ-learning // Adaptive Behavior. Vol. 6.2. P.219-246. 1998.

6. Wilson S.W. Knowledge growth in an artificial animal // Proceeding of the First International Conference on Genetic Algoritms and Their Applications. Hillsdale, New Jersey: Lawrence Erlbaum Associates. 1985. P.16-23.

7. Pchelkin A. Efficient exploration in reinforcement learning based on Utile Suffix Memory // Journal «Informatica», Lithuanian Academy of Sciences. Vol.14. 2003.

8. Kearns M.J., Vazirani U.V. An Introduction to Computational Learning Theory // MIT Press, 1994.

9. Анохин П.К. Принципиальные вопросы общей теории функциональных систем // Принципы системной организации функций. М.: Наука, 1973.

10. Анохин П. К. Системный анализ интегративной деятельности нейрона // «Успехи физиол. наук». 1974. Т.5. № 2. С.5-92.

11. Витяев Е.Е. Формальная модель работы мозга, основанная на принципе предсказания // Модели Когнитивных Процессов, Труды ИМ СО РАН (Выч системы, 164), Новосибирск, 1998. С.3-62.