Глава 4

Лекция 7. Алгоритмы ESP и H-ESP

Оглавление

В этой и последующих лекциях будут описаны некоторые известные нейроэволюционные алгоритмы. Знание об этих алгоритмах может дать представление о самом процессе разработки алгоритма, его настройке, модификации и способах заставить алгоритм работать.

В большинстве случаев новый нейроэволюционный алгоритм, даже если он позволяет получить решение той или иной задачи, работает не так хорошо как хотелось бы: слишком долго, получаются слишком «сложные и запутанные» решения (с точки зрения структуры ИНС), низкий процент успешных запусков, сложно настроить под задачу и т.д. Путь от первой версии алгоритма до конечного варианта может быть очень сложен, за бортом часто остаются мегабайты и гигабайты экспериментальных данных, полученных в ходе обкатывания разных идей. Поэтому знание о чужом опыте может оказаться очень полезным и часто содержит уже готовые (или «полуфабрикатные») рецепты.

7.1 Алгоритм ESP

Предложен Фаустино Гомесом [1]. Представляет вариант коэволюционного алгоритма эволюции весов ИНС.

Может использоваться для любой ИНС с 1 скрытым слоем:

- Прямого распространения
- Сеть Элмана
- Полносвязная рекуррентная

Используется вещественное кодирование. Хромосома содержит следующие параметры:

- Веса всех входных связей
- Веса всех выходных связей

- Веса всех рекуррентных связей

Общая схема работы ESP описана в алгоритме 7.1.

Отличительными особенностями ESP являются: использование механизма взрывной мутации (алг. 7.2) и базовая подстройка структуры ИНС (алг. 7.3). Взрывная мутация направлена на выход из локального экстремума, а подстройка структуры ИНС необходима для случаев, когда адаптация параметров существующей ИНС не приводит к улучшениям качества решения.

7.1.1 Специализация нейронов

Использование коэволюционного подхода позволило реализовать так называемую специализацию нейронов. Она заключается в том, что работа нейрона зависит от занимаемого положения в сети, поэтому разные нейроны должны иметь разные веса и решать разные подзадачи. Можно предположить, что параллельный поиск отдельных нейронов, реализованный в ESP, благодаря использованию подпопуляций, повышает эффективность поиска весов связей ИНС.

Иллюстративный пример специализации нейронов показан на рис. 7.1. Изображены проекции весов связей на плоскость первых двух собственных векторов в начале поиска, а также после 20, 50 и 100 поколений. Проекции, соответствующие различным нейронам, показаны разным цветом. Можно увидеть, что в ходе эволюции происходит локализация и «отделение» векторов весов нейронов и, следовательно, специализация реализуемых ими функций.

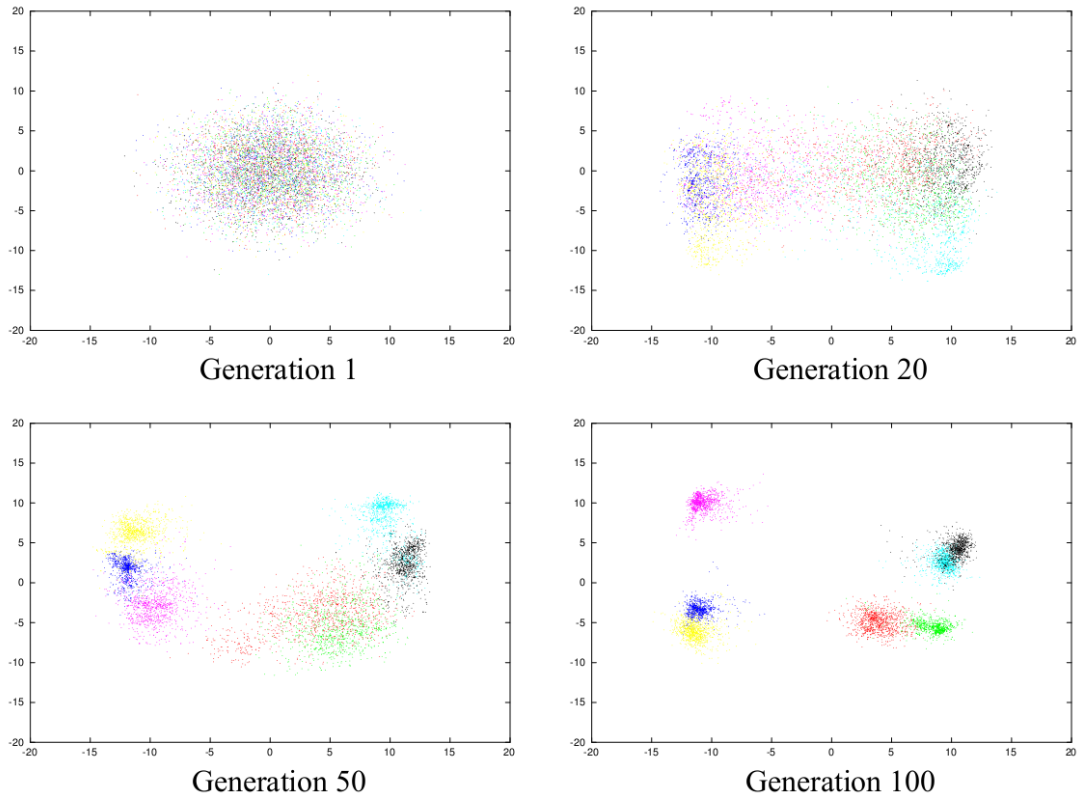

Рис. 7.1: Иллюстрация специализации нейронов, получаемой в результате работы алгоритма ESP. Использован рисунок из [1]

Алгоритм 7.1 Алгоритм Enforced Sub-Populations (ESP)

1: **Инициализация**.

- Задается число h скрытых нейронов
- Создается h подпопуляций из n особей каждая

2: **Оценка приспособленности**.

- Каждый нейрон принимает участие в попытках (*trials*), при которых он помещается в нейронную сеть случайно сформированную с использованием нейронов из других подпопуляций.
- Приспособленность для каждой особи-нейрона суммируется (*кумулятивная приспособленность*).
- Оценивание проводится, пока каждый нейрон не будет использован в, как минимум, 10 попытках.

3: **Проверка вырождения популяции**.

- Если приспособленность лучшей особи не улучшается в течение последних b поколений, то производится взрывная мутация (*burst mutation*).
- Если после двух взрывных мутаций нет улучшения приспособленности, то изменяется структура ИНС.

4: **Скрещивание**.

- Вычисление средней приспособленности для каждого нейрона: кумулятивная приспособленность делится на число попыток.
- Сортировка нейронов в каждой подпопуляции в порядке ухудшения приспособленности
- Из популяции удаляюся лишние особи, т.е. особи номер которых превосходит начальный размер популяции.
- Скрещивается 1/4 часть лучших особей подпопуляции. Используется 1-точечный кроссинговер. Потомки добавляются в конец подпопуляции.
- Для «нижней половины» популяции производится мутация с распределением Коши.

5: Повторение шагов 2–4 пока не выполнен критерий останова.

Алгоритм 7.2 Взрывная мутация (*burst mutation*)

1: Для каждой подпопуляции определяется наилучшая особь.
2: Генерируются новые подпопуляции вокруг «своих» лучших особей.
3: Используется распределение Коши.

- Распределение Коши с центром в точке $x = 0$ описывается формулой $f(x) = \frac{\alpha}{\pi(\alpha^2+x^2)}$.
- 50% СВ попадает в интервал $\pm\alpha$.
- 99.9% СВ попадает в интервал $\pm 381.3\alpha$.
- Другими словами, это распределение с очень тяжелым хвостом.

Алгоритм 7.3 Адаптация структуры в ESP

1: Уменьшение размера ИНС.

- По очереди отключается удаляется одна подпопуляция. Оставшиеся $h - 1$ популяции оцениваются стандартным способом.
- Если лучшая приспособленность новой ИНС, полученной после отключения популяции, лучше, чем лучшая приспособленность до отключения, то подпопуляция удаляется. Т.е. предполагается, что она не нужна.

2: Если ни одна популяция не была удалена (количество нейронов не уменьшилось), то к популяциям добавляется новая подпопуляция со случайно проинициализированными нейронами.

Алгоритм	Количество вычислений ЦФ	Время, с
Q-MLP	2056	33
SARSA-CMAC	540	487
SARSA-CABA	965	1713
CNE	352	5
SANE	302	5
NEAT	743	7
ESP	289	4

Таблица 4.1: Результаты решения задачи балансирования 1 шеста. Использованы данные из [1]

Алгоритм	Количество вычислений ЦФ	Время, с	Успешность, %
VAPS	(500,000)	(5days)	(0)
Q-MLP	11331	340	100
SARSA-CMAC	13562	2034	59
SARSA-CABA	15617	6754	70
CNE	724	15	100
NEAT	1523	15	100
ESP	589	11	100

Таблица 4.2: Результаты решения задачи балансирования 1 шеста без данных о скоростях. Использованы данные из [1]

7.1.2 Задача балансировки шестов

Рассматриваются варианты как с информацией о скорости движения шестов, так и без нее.

Входы и выходы ИНС – стандартные для данной задачи. Результаты экспериментов (табл. 4.1, 4.2, 4.3, 4.4) показали высокую эффективность алгоритма ESP.

Также Ф. Гомесом был рассмотрен вариант задачи для плоского двумерного поля [3], на котором располагается тележка. При этом шест мог упасть уже в любом направлении.

7.1.3 Задача поиска жертвы

Задача заключается в поиске ИНС, которая сможет управлять агентом-хищником, чтобы он поймал агента-жертву, движущуюся по детерминированному алгоритму [2].

Агент-хищник обладает сенсорным полем конечного радиуса, в котором он «чувствует» жертву. Это поле поделено на 8 равных секторов. При этом имеется датчик, определяющий находится ли жертва вблизи хищника (расстояние до жертвы меньше некоторого r), или на периферии. Также агент обладает детекторами стен (с непрерывными сигналами), которые активизируются, когда агент приближается к стене (рис. 7.2).

Выходами ИНС являются возможные направления движения хищника (Север, Юг, Запад, Восток). При этом в каждый момент времени выбирается ровно 1 выход с максимальным состоянием, который и определяет в какую сторону будет двигаться хищник.

Алгоритм	Количество вычислений ЦФ	Время, с
Q-MLP	10582	153
CNE	22100	73
EP	307200	–
SANE	12600	37
NEAT	3600	31
ESP	3800	22

Таблица 4.3: Результаты решения задачи балансирования 2 шестов. Использованы данные из [1]

Method	Evaluations	
	Standard fitness	Damping fitness
CE	–	(840,000)
CNE	76906	87623
NEAT	20918	24543
ESP	20456	26342

Таблица 4.4: Результаты решения задачи балансирования 2 шестов без данных о скоростях. Использованы данные из [1]

Изначально в рекуррентной сети используется 5 нейронов.

В исследовании рассматривалось два варианта эволюции: прямая и инкрементная. В прямой эволюции жертва изначально находится на расстоянии 4 шага и двигается с максимально возможной скоростью. В инкрементном варианте обучение начинается с условий, когда жертва изначально находится ближе к хищнику, а скорость ее движения меньше. Когда хищник научается ловить жертву, то условия усложняются: жертва ставится дальше от хищника и двигается быстрее, при этом эволюция нейронной сети продолжается. В каждом тесте хищник помещается в центр поля, а жертва в случайную позицию в пределах видимости сенсоров хищника.

Если обозначить задачу как E_m^s, где m – количество шагов, которые делает жертва, прежде чем хищник начнет движение, s – скорость движения жертвы, то инкрементный вариант предусматривал следующую последовательность задач:

$$E_0^{0.0} \to E_2^{0.0} \to E_3^{0.0} \to E_4^{0.0} \to E_4^{0.3} \to E_4^{0.6} \to E_4^{0.8} \to E_4^{1.0}$$

Т.е. $E_0^{0.0}$ – случай неподвижной жертвы, находящейся в непосредственной близости к хищнику, а $E_4^{0.6}$ – случай, когда жертва делает 4 шага и далее двигается со скоростью, равной 60% от максимально возможной. Есть вероятность, что после 4 шагов жертва выйдет из области видимости хищника и необходимо сначала будет найти жертву на поле.

При прямом обучении сразу использовалась самая сложная задача, $E_4^{1.0}$.

В результате инкрементное обучение позволило получить более качественные результаты с точки зрения поведения хищника. В частности, при использовании прямого обучения так и не удалось найти ИНС для задачи $E_4^{1.0}$.

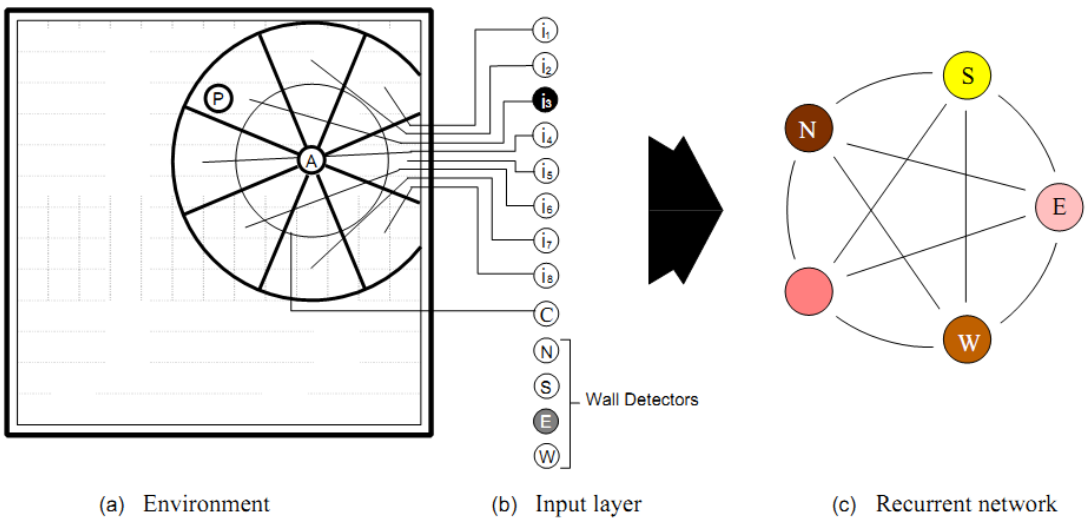

Рис. 7.2: Иллюстрация использования ИНС для управления движением хищника. Использован рисунок из [2]

7.2 Алгоритм H-ESP

Также предложен Ф Гомесом в статье [4] как вариант развития алгоритма ESP. Нововведением является использование эволюции уровня сетей (L1), а не только нейронов (L2). Общая схема работы алгоритма представлена на рис. 7.3.

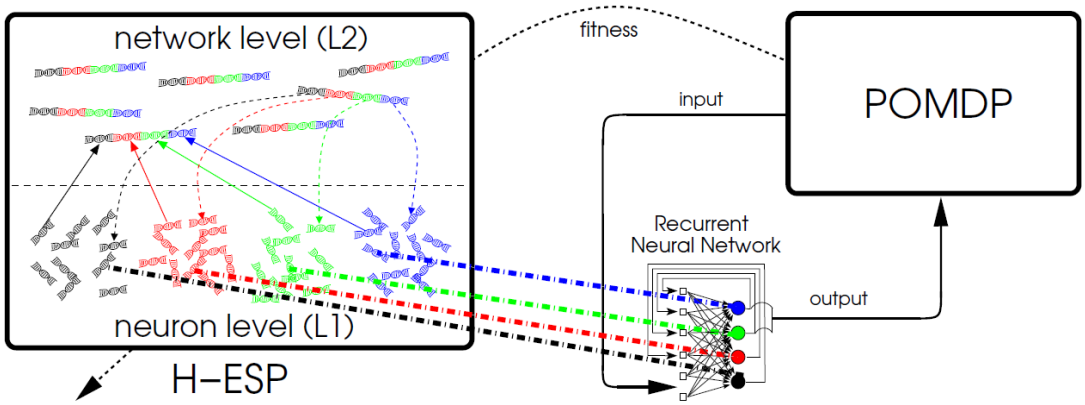

Рис. 7.3: Схематичное изображение алгоритма H-ESP. Использован рисунок из [4]

Описание укрупненного алгоритма представлено в 7.4. Описание разделено на уровень нейронов и сетей.

Как видно отличий от ESP не так уж и много, однако введение дополнительного уровня иерархии позволяет вести учет найденных хороших решений и направлять с их помощью эволюционный поиск.

Алгоритм 7.4 Алгоритм Hierarchical ESP (H-ESP)

1: Инициализация.

- Уровень нейронов. Все по-старому.
- Уровень нейронных сетей. Формируется N случайных сетей с h скрытыми нейронами.

2: Оценивание.

- Уровень нейронов. Если полученная при оценивании сеть лучше, чем худшая сеть из L2, то она добавляется в L2.
- Уровень нейронных сетей. Формируется N случайных сетей с h скрытыми нейронами.
  - Определяется приспособленность еще не оцененных сетей.
  - Если приспособленность сети лучше, чем приспособленность лучшей ИНС из L1, то нейроны из этой сети добавляются в подпопуляции.

3: Рекомбинация.

- Уровень нейронов. Как в ESP.
- Уровень нейронных сетей. Каждая ИНС скрещивается с более приспособленной с использованием h-точечного кроссинговера («понейронно»). Потомки мутируют.

4: Повторение шагов 2-3.

Литература

[1] Faustino Gomez. *Robust Non-Linear Control through Neuroevolution.* PhD thesis, Department of Computer Sciences, 2003.

[2] Faustino Gomez and Risto Miikkulainen. Incremental evolution of complex general behavior. *Adaptive Behavior*, 5:317–342, 1997.

[3] Faustino Gomez and Risto Miikkulainen. 2-d pole balancing with recurrent evolutionary networks. In *Proceedings of the International Conference on Artificial Neural Networks (ICANN-98, Skovde, Sweden)*, pages 245–250. Berlin, New York: Springer, 1998.

[4] Faustino Gomez and Juergen Schmidhuber. Co-evolving recurrent neurons learn deep memory pomdps. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-05, Washington, D.C.)*, 2005.