

Глава 4

Лекция 9. Алгоритм NEAT

Оглавление

4	Лекция 9. Алгоритм NEAT	1
9.1	Алгоритм NEAT	2
9.1.1	Кодирование информации	2
9.1.2	Скращивание	3
9.1.3	Мутация	4
9.1.4	Специализация особей	4
9.1.5	Козволюционное усложнение (The Robot Duel Domain)	6
9.2	Алгоритм HyperNEAT	8
9.2.1	О вычислительной сложности HyperNEAT	9

9.1 Алгоритм NEAT

Алгоритм NEAT (Neuroevolution of Augmenting Topologies) предложен Кеннетом Стенли¹ [7]. Представляет вариант эволюционного алгоритма для эволюционного поиска весов и структуры ИНС.

Используется как для ИНС прямого распространения, так и для рекуррентных ИНС. Ограничений на структуру сети (количество нейронов, связей, слоев и др.) нет.

Другие особенности алгоритма:

- Адаптивные операторы скрещивания и мутации.
- «Исторические метки» («Historical markings»).
- Эволюция начинается с ИНС без скрытых нейронов и идет в направлении усложнения структуры.

9.1.1 Кодирование информации

1 особь = 1 ИНС, представленная списком связей и списком нейронов (рис. 9.1). Информация об одной связи включает:

- Индекс входного нейрона.
- Индекс выходного нейрона.
- Вес связи (вещественное число).
- Метку, обозначающую, используется ли данная связь.

¹ <http://www.cs.ucf.edu/~kstanley/>

- Историческую метку («Номер инновации»).

Список нейронов содержит информацию об уникальном индексе каждого нейрона и его типе (входной, выходной или скрытый).

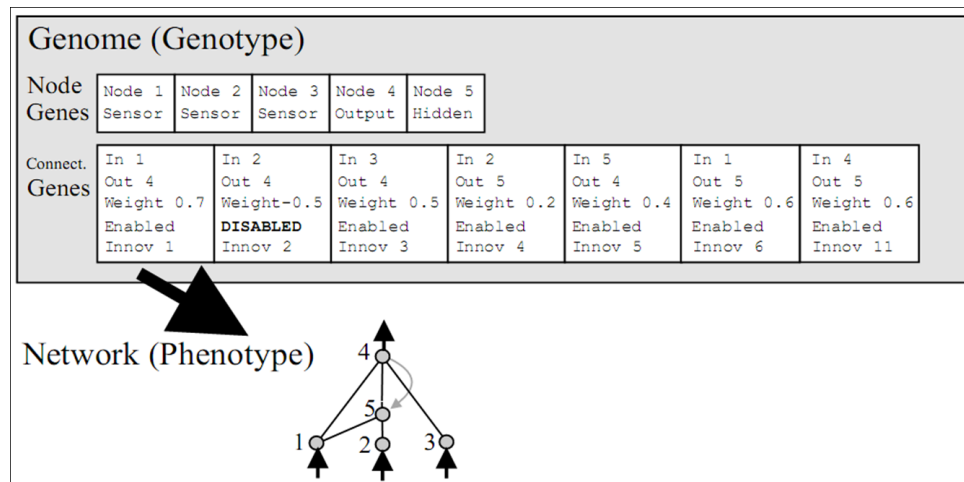


Рис. 9.1: Кодирование информации об ИНС в алгоритме NEAT. Использован рисунок из [7].

Историческая метка является глобальным идентификатором, задающим порядковый номер гена. Она определяется следующим образом: каждый раз, когда в хромосоме в результате мутации появляется новый ген, этому гену присваивается текущее значение метки, после чего метка увеличивается на единицу (см. пример из раздела 9.1.3). Метка гена не меняется, таким образом, по историческим меткам можно отследить порядок появления связей в нейронных сетях особей.

Существует вероятность, что одинаковые изменения структур разных ИНС получат разные исторические метки. Чтобы этого избежать в каждом поколении создается список, в котором регистрируются новые межнейронные связи, и с его помощью происходит отслеживание одинаковых изменений и коррекция меток. Стенли пишет, что списка на одно поколение вполне достаточно, чтобы избежать коллизий, по сравнению с общим списком, который накапливается с самого начала эволюционного поиска [4].

9.1.2 Скрещивание

Одна из особенностей алгоритма NEAT – это возможность скрещивания ИНС с различной структурой, при этом из 2 родительских ИНС получается 1 ИНС-потомок (рис. 9.2). Особи для скрещивания выбираются случайным образом.

Чтобы обойти сложности, связанные с операцией скрещивания, рекомбинировать могут только те связи двух ИНС, которые имеют одинаковую метку, т.к. предполагается, что такие связи обозначают одни и те же изменения. Значение веса связи потомка при этом может определяться одним из двух способов: случайный выбор от одного из родительских значений, либо усреднение.

Все несовпадающие связи копируются в хромосому потомка от более приспособленного родителя. Если приспособленность обоих родителей одинакова, то тогда такие связи попадают к потомку случайно.

В результате скрещивания ранее отключенные гены могут снова активированы.

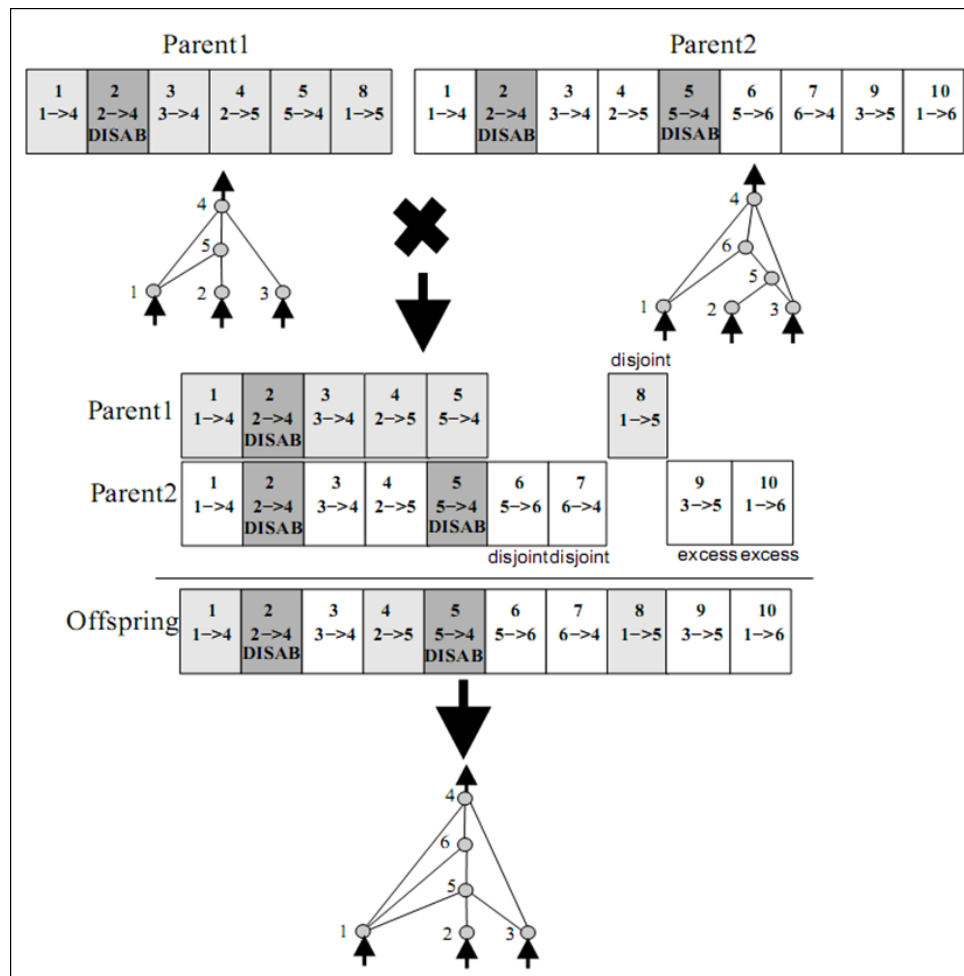


Рис. 9.2: Оператор скрещивания в алгоритме NEAT. Использован рисунок из [7].

9.1.3 Мутация

К данной ИНС возможно применить два различных оператора мутации (рис. 9.3):

1. Добавление веса связи (рис. 9.3,верху). Приводит к появлению нового гена, которому ставится в соответствие уникальная историческая метка.
2. Добавление нейрона (рис. 9.3,внизу). Нейрон «вставляется» в существующую связь, в результате чего соответствующий ген помечается как отключенный, а в хромосому добавляются два новых гена для связей к и от нового нейрона.

9.1.4 Специализация особей

Для предотвращения преждевременной сходимости и увеличения разнообразия особей в популяции в алгоритме NEAT применяется прием, известный как «нишинг»² (*niching*) [2]. Основная идея заключается в том, что конкуренция в популяции имеет место только среди ИНС одного «вида», но не между видами. Чтобы определить принадлежность ИНС к одному виду, необходимо вычислить, насколько эта ИНС

²«Разделение на ниши», на мой взгляд, слишком громоздко.

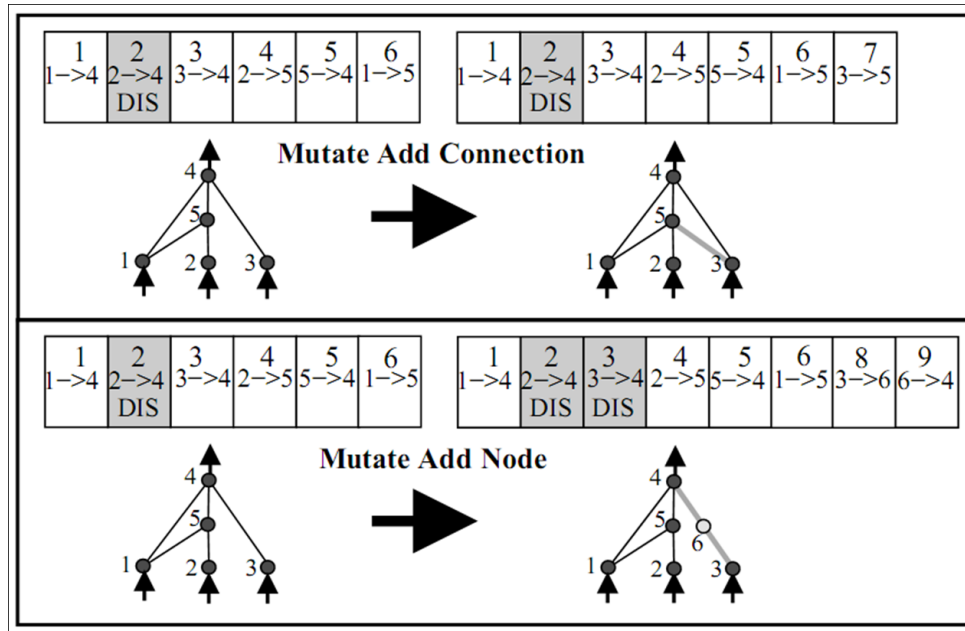


Рис. 9.3: Операторы добавления связи и нейрона в алгоритме NEAT. Использован рисунок из [7].

близка к ИНС этого вида. Для этого используется следующая формула:

$$\delta = \frac{c_1 E}{N} + \frac{c_2 D}{N} + c_3 \bar{W},$$

где E – количество избыточных (excess) генов, D – количество непересекающихся (disjoint) генов, \bar{W} – средняя разница весов совмещенных (matching) генов, c_i – числовые коэффициенты, определяющие важность этих величин, N – количество генов в большей хромосоме.

Для определения, близки ли две ИНС используется порог δ_t , если расстояние между ИНС и любой ИНС из рассматриваемого вида меньше порога, то считается что ИНС совместима с этим видом.

В алгоритме содержится множество S видов особей. В начальный момент времени это множество пустое. Формирование видов описано в алг. 9.1

Алгоритм 9.1 Алгоритм разбиения популяции на виды

- 1: **for** каждая особь g из популяции **do**
 - 2: **for** каждый вид s из текущего множества видов S **do**
 - 3: Если g совместима с s , то поместить g в s и выйти из цикла.
 - 4: **end for**
 - 5: Если g несовместима ни с одним видом из S , то создать новый вид s_{new} , добавить туда особь g и расширить множество S новым видом.
 - 6: **end for**
-

Чтобы какой-нибудь вид не вытеснил остальные, применяется явное разделение приспособленности (*explicit fitness sharing* [1]), в котором приспособленности особей пересчитываются согласно выражению:

$$f'_i = \frac{f_i}{\sum_{j=1}^n \text{sh}(\delta(i, j))},$$

где $\text{sh}(\delta(i, j))$ вычисляется в соответствии с [3]:

$$\text{sh}(\delta(i, j)) = \begin{cases} 0, & \delta(i, j) > \delta_t \\ 1, & \delta(i, j) \leq \delta_t \end{cases}$$

Таким образом, чем больше особей какого-либо вида в популяции, тем меньше скорректированная приспособленность этих особей. Лучшая особь каждого вида автоматически попадает в следующее поколение.

Использование специализации особей позволяет защитить от «каннибализма» ИНС с оригинальной структурой и способствует исследованию пространства поиска.

9.1.5 Козволюционное усложнение (The Robot Duel Domain)

Для анализа возможностей алгоритма NEAT к поиску решений в открытой среде, в которой «предел эволюции» заранее неизвестен, использовалась задача Дуэль роботов (*The Robot Duel*).

Вопросы, на которые необходимо было найти ответы в результате исследования, таковы:

- Способен ли алгоритм NEAT искать все более сложные с точки зрения топологии решения, если в этом есть необходимость?
- Действительно ли в новых решениях усложнение структуры ведет к улучшению качества?
- Насколько лучше или хуже будут решения с более сложной структурой по сравнению с решениями, полученными с явными ограничениями на топологию ИНС.

В рассматриваемой задаче два робота располагаются на плоском поле (рис. 9.4). Каждый робот управляется своей ИНС, которая выдает сигналы на «приводы» колес. При движении роботы тратят энергию, причем количество энергии пропорционально величине управляющих сигналов. Если роботы сталкиваются, то побеждает (остается в «живых») тот, у которого остается больше энергии. На поле присутствует еда³, съев которую робот пополняет запас энергии.

Описание входов и выходов ИНС, а также общее сенсоров робота приведено на рис. 9.5. Каждый робот получает информацию о наличии поблизости еды, другого робота, стен, а также данные о разности своей энергии и энергии робота-соперника. За движение отвечают два колеса, управляемые независимо⁴.

Выходные сигналы ИНС определяет величину сигнала, подаваемого на левое и правое колеса, а также общее направление вращения колес (вперед или назад).

Каждый запуск нейроэволюционного поиска длился 500 поколений. Рассматривалась **конкурентная козволюция**. За эволюцию ИНС для каждого робота отвечает отдельная популяция. Каждая особь оценивается в левой и в правой начальных позициях, т.е. ИНС должна выработать достаточно универсальные правила поведения для роботов. Для оценки особи из конкурирующей популяции выбираются по лучшей особи из 4 видов⁵ + еще 8 особей из коллекции чемпионов прошедших поколений.

³Да-да те самые гамбургеры :).

⁴Т.е. робот напоминает Khepera, который уже рассматривался ранее

⁵В NEAT есть специализация, помните?

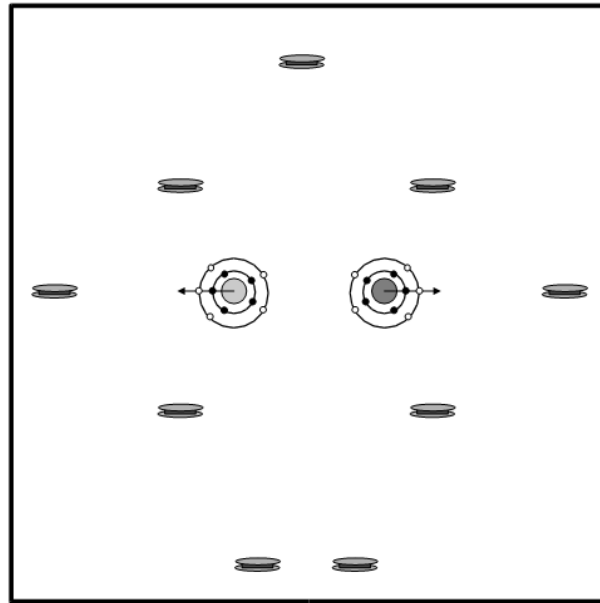


Рис. 9.4: Среда для задачи с дуэлью роботов. Показаны начальные положения роботов, направление зрения, расположение еды на поле. Использовано изображение из [4]

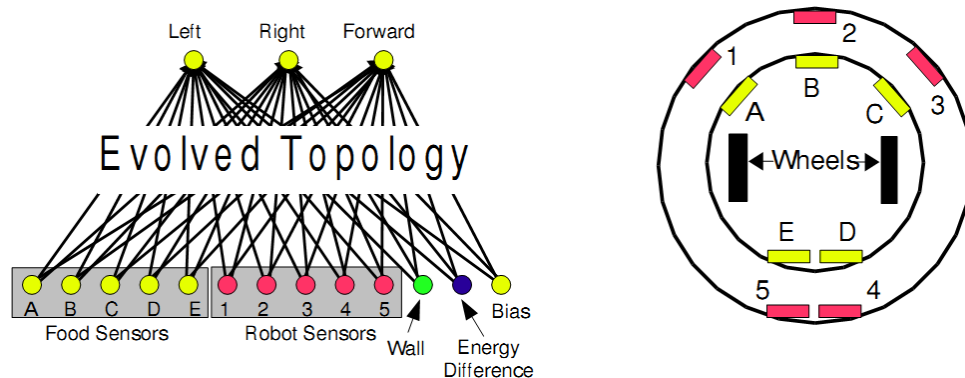


Рис. 9.5: Описание входов и выходов нейронной, управляющей роботом (а), а также сенсоров робота (б). Использовано изображение из [4]

Таким образом, каждая ИНС оценивается 24 раза: 12 противников и для каждого рассматриваются две стартовые позиции.

Пример результата для конкурентной коэволюции показан на рис. 9.6.

Ниже приведены особенности ИНС из разных поколений, часть из которых показана на рис. 9.6:

- Поколение 100. Постоянно преследует противника, даже если у того больше энергии, или если появляется возможность подкрепиться. Характер нордический.
- Поколение 200. Появилась стратегия *отдыхать*: когда энергии мало, робот практически переставал ездить по полю, а ждал, пока противник бегал вокруг. В этом случае появлялся шанс, что робот-соперник изматывается и его можно атаковать.

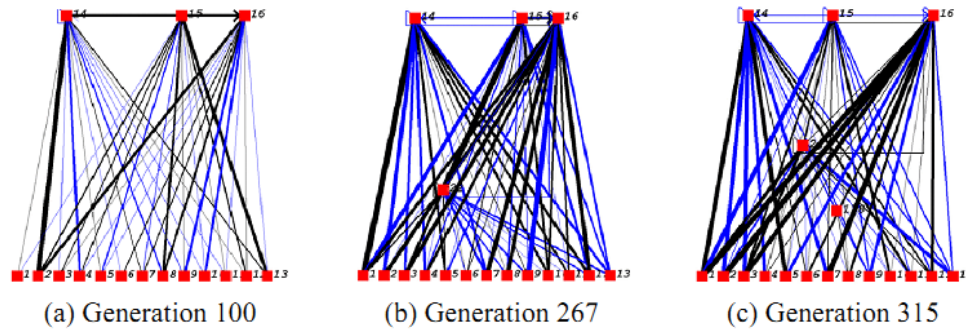


Рис. 9.6: Результаты усложнения структуры ИНС в ходе эволюции. Показаны ИНС для разных поколений, появление скрытых нейронов приводило к изменению особенностей поведения робота. Использовано изображение из [4]

- Поколение 267. Несколькими поколениями ранее появился скрытый нейрон, связи которого успели улучшиться. Благодаря этому появилась возможность переключать поведение: робот атаковал, когда у него было преимущество в энергии.
- Поколение 267. Еще один скрытый нейрон позволил улучшить общую стратегию. Теперь робот не бегал по пятам за противником, старался собрать как можно больше еды, и атаковал, только при благоприятных обстоятельствах. Т.е. поведение гораздо более аккуратным.

Таким образом, было показано, что усложнение структуры ИНС может привести к появлению более сложных и полезных стратегий поведения.

9.2 Алгоритм HyperNEAT

Данный алгоритм уже упоминался ранее, поэтому здесь мы рассмотрим его кратко, чтобы напомнить основные особенности.

Прежде всего, это алгоритм с косвенным кодированием решений. В нем используется специальная сеть Compositional Pattern Producing Network (CPPN) [6] для определения весов связей в нейронной сети. При этом большое влияние оказывает расположение нейронов в так называемом субстрате. У каждого нейрона есть свои координаты. Чтобы определить вес связи между нейронами, на вход CPPN подаются координаты этих нейронов, а выходной сигнал CPPN и является искомым весом связи (рис. 9.7).

Если при неизменной CPPN поменять положение нейронов в субстрате, то, скорее всего, веса связей изменятся. Важным и интересным свойством CPPN является возможность масштабирования размера ИНС.

В алгоритме HyperNEAT предусмотрена возможность изменения функции активации нейронов, что не часто встречается в НЭ алгоритмах. Список функций активации, используемых по умолчанию, приведен в табл. 4.1.

Пример CPPN показан на рис. 9.8.

При добавлении нового узла в алгоритме HyperNEAT этому узлу по умолчанию присваивается случайная функция активации из имеющегося множества функций. Кроме этого в силу того, что две ИНС могут отличаться не только структурой, но и функциями активации нейронов, в формулу для вычисления расстояния между

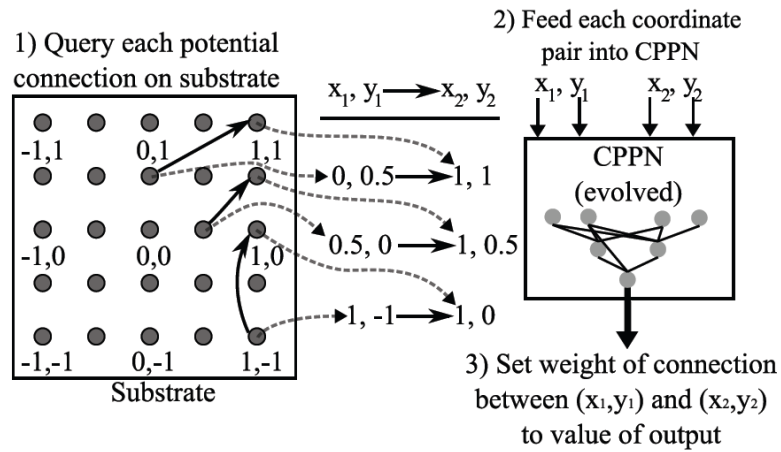


Рис. 9.7: Кодирование весов с использованием алгоритма HyperNEAT. Изображение с сайта <http://eplex.cs.ucf.edu/hyperNEATpage/HyperNEAT.html>

Название	Формула
Линейная с ограничениями	$\max(-1, \min(1, x))$
Биполярная сигмоида	$\frac{2}{1 + \exp^{-4.9x}} - 1$
Гауссовская	$\exp^{-(2.5x)^2}$
Синусоидальная	$\sin(2x)$

Таблица 4.1: Функции активации для алгоритма HyperNEAT. $S = \mathbf{w}^T \mathbf{x}$.

ИНС добавлено слагаемое, учитывающее различия в функциях активации. Других существенных отличий алгоритмов NEAT и HyperNEAT нет [5].

9.2.1 О вычислительной сложности HyperNEAT

Сравнивая вычислительную сложность HyperNEAT с генными регуляторными сетями (ГРН) (*gene regulatory networks, GRN*), необходимую для формирования ИНС, Стенли отмечает, что последние слабее по эффективности, требуя порядка $O(c^2 t)$ операций для вычисления выходного сигнала, где c – корень из количества активных генов (узлов), t – количество тактов для вычисления выхода ГРН. При этом для HyperNEAT декларируется сложность порядка $O(n^2)$, где n – размер измерения в двумерном субстрате. Т.е. тоже квадратичная сложность, но уже без дополнительного множителя t .

Однако, Стенли опять-таки отмечает (справедливо!), что сравнивать напрямую оценки вычислительной сложности для CPPN и ГРН нельзя, т.к. рассматриваются разные параметры. В принципе, все правильно, только нужно добавить, что вычисление выходного сигнала в CPPN не бесплатно и зависит от количества узлов в сети. Т.е. если обозначить это количество узлов через a , то количество операций для CPPN, как минимум, будет $O(n^2 a)$, таким образом, дополнительный множитель появляется и здесь. В целом, что эффективнее для задания структуры ИНС, сказать однозначно сложно, хотя было бы любопытно определить и формализовать условия, когда CPPN эффективнее, чем ГРН, и наоборот.

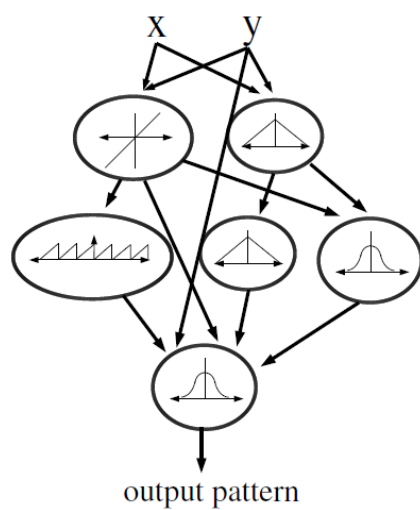


Рис. 9.8: Пример CRRN, представленной в виде графа. Изображение из статьи [5]

Литература

- [1] D. E. Goldberg and J. Richardson. Genetic algorithms with sharing for multimodal function optimization. In J. J. Grefenstette, editor, *Proceedings of the Second International Conference on Genetic Algorithms*, pages 148–154, San Francisco, CA, 1987. Morgan Kaufmann.
- [2] J. Horn. *The Nature of Niching: Genetic Algorithms and the Evolution of Optimal Cooperative Populations*. PhD thesis, University of Illinois at Urbana-Champaign, 1997.
- [3] W. Spears. Speciation using tag bits. In *Handbook of Evolutionary Computation*. IOP Publishing Ltd. and Oxford University Press, 1995.
- [4] Kenneth O. Stanley. *Efficient Evolution of Neural Networks Through Complexification*. Phd thesis, Department of Computer Sciences, The University of Texas at Austin, 2004.
- [5] Kenneth O. Stanley. Compositional pattern producing networks: A novel abstraction of development. *Genetic Programming and Evolvable Machines Special Issue on Developmental Systems*, 8(2):131–162, 2007.
- [6] K.O. Stanley. Patterns without development, 2006. Technical report CS-TR-06-01, University of Central Florida, 2006.
- [7] K.O. Stanley and R. Miikkulainen. Evolving neural networks through augmenting topologies. *Evolutionary Computation*, 10(2):99–127, 2002.