

Курс лекций «Введение в ИИ.». Часть III.
Нейрокомпьютеринг.
Лекция 17. Сети встречного распространения

О.Г. Чанышев

Содержание

1 Введение	1
2 Структура сети	1
3 Нормальное функционирование	2
3.1 Слои Кохоненна	2
3.2 Слой Гроссберга	2
4 Обучение слоя Кохонена	3
4.1 Выбор начальных значений весовых векторов	4
4.2 Режим интерполяции	4
4.3 Статистические свойства обученной сети	4
5 Обучение слоя Гроссберга	5
6 Сеть встречного распространения полностью	5
7 Приложение: сжатие данных	6
8 Обсуждение	6

1 Введение

Возможности сети встречного распространения, разработанной в [5-7], превосходят возможности однослойных сетей. Время же обучения по сравнению с обратным распространением может уменьшаться в сто раз. Встречное распространение не столь общо, как обратное распространение, но оно может давать решение в тех приложениях, где долгая обучающая процедура невозможна. Встречное распространение обладает собственными интересными и полезными свойствами. Во встречном распространении объединены два хорошо известных алгоритма: самоорганизующаяся карта Кохонена [8] и звезда Гроссберга [2-4] (см. Приложение). Их объединение ведет к свойствам, которых нет ни у одного из них в отдельности.

Рис. 1: Сеть с встречным распознаванием без обратных связей

Методы, которые подобно встречному распространению, объединяют различные сетевые парадигмы как строительные блоки, могут привести к сетям, более близким к мозгу по архитектуре, чем любые другие однородные структуры. Похоже, что в мозгу именно каскадные соединения модулей различной специализации позволяют выполнять требуемые вычисления. Сеть встречного распространения функционирует подобно столу справок, способному к обобщению. В процессе обучения входные векторы ассоциируются с соответствующими выходными векторами. Эти векторы могут быть двоичными, состоящими из нулей и единиц, или непрерывными. Когда сеть обучена, приложение входного вектора приводит к требуемому выходному вектору. Обобщающая способность сети позволяет получать правильный выход даже при приложении входного вектора, который является неполным или слегка неверным. Это позволяет использовать данную сеть для распознавания образов, восстановления образов и усиления сигналов.

2 Структура сети

На рис. 1 показана упрощенная версия прямого действия сети встречного распространения. На нем иллюстрируются функциональные свойства этой парадигмы. Полная двунаправленная сеть основана на тех же принципах, она обсуждается позднее.

Каждый нейрон слоя 0 соединен с каждым нейроном слоя 1 (называемого слоем Кохонена) отдельным весом w_{mn} . Эти веса в целом рассматриваются как матрица весов W . Аналогично, каждый нейрон в слое Кохонена (слое 1) соединен с каждым нейроном в слое Гроссберга (слое 2) весом v_{np} . Эти веса образуют матрицу весов V . Все это весьма напоминает другие сети, встречавшиеся в предыдущих главах, различие, однако, состоит в операциях, выполняемых нейронами Кохонена и Гроссберга. Как и многие другие сети, встречное распространение функционирует в двух режимах: в нормальном режиме, при котором принимается входной вектор \vec{X} и выдается выходной вектор \vec{Y} , и в режиме обучения, при котором подается входной вектор и веса корректируются, чтобы дать требуемый выходной вектор.

3 Нормальное функционирование

3.1 Слои Кохоненна

В своей простейшей форме слой Кохонена функционирует в духе «победитель забирает все», т. е. для данного входного вектора один и только один нейрон Кохонена выдает на выходе логическую единицу, все остальные выдают ноль.

Ассоциированное с каждым нейроном Кохонена множество весов соединяет его с каждым входом. Например, на рис. 4.1 нейрон Кохонена K_1 имеет веса $w_{11}, w_{21}, \dots, w_{m1}$, составляющие весовой вектор W_1 . Они соединяются через входной слой с входными сигналами x_1, x_2, \dots, x_m , составляющими входной вектор \vec{X} . Подобно нейронам большинства сетей выход NET каждого нейрона Кохонена является просто суммой взвешенных входов. Это может быть выражено

следующим образом:

$$NET_j = \sum_i w_{ij} x_i \quad (1)$$

где NET_j – это выход NET нейрона Кохонена j , или в векторной записи

$$\vec{N} = \vec{X}W, \quad (4.3)$$

где \vec{N} – вектор выходов NET слоя Кохонена.

Нейрон Кохонена с максимальным значением NET является «победителем». Его выход равен единице, у остальных он равен нулю.

3.2 Слой Гроссберга

Слой Гроссберга функционирует в сходной манере. Его выход NET является взвешенной суммой выходов k_1, k_2, \dots, k_n слоя Кохонена, образующих вектор \vec{K} . Вектор соединяющих весов, обозначенный через \vec{V} , состоит из весов $v_{11}, v_{21}, \dots, v_{np}$. Тогда выход NET каждого нейрона Гроссберга есть

$$NET_j = \sum_i k_i v_{ij} \quad (3)$$

где NET_j – выход j -го нейрона Гроссберга, или в векторной форме

$$\vec{Y} = \vec{K}V, \quad (4)$$

где \vec{Y} – выходной вектор слоя Гроссберга, \vec{K} – выходной вектор слоя Кохонена, V – матрица весов слоя Гроссберга.

Если слой Кохонена функционирует таким образом, что лишь у одного нейрона величина NET равна единице, а у остальных равна нулю, то лишь один элемент вектора K отличен от нуля, и вычисления очень просты. Фактически каждый нейрон слоя Гроссберга лишь выдает величину веса, который связывает этот нейрон с единственным ненулевым нейроном Кохонена.

4 Обучение слоя Кохонена

Слой Кохонена классифицирует входные векторы в группы схожих. Это достигается с помощью такой подстройки весов слоя Кохонена, что близкие входные векторы активируют один и тот же нейрон данного слоя. Затем задачей слоя Гроссберга является получение требуемых выходов. Обучение Кохонена является самообучением, протекающим без учителя. Поэтому трудно (и не нужно) предсказывать, какой именно нейрон Кохонена будет активироваться для заданного входного вектора. Необходимо лишь гарантировать, чтобы в результате обучения разделялись несхожие входные векторы. Предварительная обработка входных векторов весьма желательно (хотя и не обязательно) нормализовать входные векторы перед тем, как предъявлять их сети. Это выполняется с помощью деления каждой компоненты входного вектора на длину вектора. Эта длина находится извлечением квадратного корня из суммы квадратов компонент вектора. В алгебраической записи

$$x' = \frac{x_i}{\sqrt{\sum_i x_i^2}} \quad (5)$$

Это превращает входной вектор в единичный вектор с тем же самым направлением, т. е. в вектор единичной длины в n -мерном пространстве.

При обучении слоя Кохонена на вход подается входной вектор и вычисляются его скалярные произведения с векторами весов, связанными со всеми нейронами Кохонена. Нейрон с максимальным значением скалярного произведения объявляется «победителем» и его веса подстраиваются. Так как скалярное произведение, используемое для вычисления величин NET , является мерой сходства между входным вектором и вектором весов, то процесс обучения состоит в выборе нейрона Кохонена с весовым вектором, наиболее близким к входному вектору, и дальнейшем приближении весового вектора к входному. Снова отметим, что процесс является самообучением, выполняемым без учителя. Сеть самоорганизуется таким образом, что данный нейрон Кохонена имеет максимальный выход для данного входного вектора. Уравнение, описывающее процесс обучения имеет следующий вид:

$$w^{(new)} = w^{(old)} + \alpha(x - w^{(old)}), \quad (4.7) \quad (6)$$

где $w^{(new)}$ – новое значение веса, соединяющего входную компоненту x с выигравшим нейроном; $w^{(old)}$ – предыдущее значение этого веса; α – коэффициент скорости обучения, который может варьироваться в процессе обучения.

Каждый вес, связанный с выигравшим нейроном Кохонена, изменяется пропорционально разности между его величиной и величиной входа, к которому он присоединен. Направление изменения минимизирует разность между весом и его входом.

Переменная α является коэффициентом скорости обучения, который вначале обычно равен ~ 0.7 и может постепенно уменьшаться в процессе обучения. Это позволяет делать большие начальные шаги для быстрого грубого обучения и меньшие шаги при подходе к окончательной величине. Если бы с каждым нейроном Кохонена ассоциировался один входной вектор, то слой Кохонена мог бы быть обучен с помощью одного вычисления на вес. Веса нейрона-победителя приравнялись бы к компонентам обучающего вектора ($\alpha = 1$). Как правило, обучающее множество включает много сходных между собой входных векторов, и сеть должна быть обучена активировать один и тот же нейрон Кохонена для каждого из них. В этом случае веса этого нейрона должны получаться усреднением входных векторов, которые должны его активировать. Постепенное уменьшение величины α уменьшает воздействие каждого обучающего шага, так что окончательное значение будет средней величиной от входных векторов, на которых происходит обучение. Таким образом, веса, ассоциированные с нейроном, примут значение вблизи «центра» входных векторов, для которых данный нейрон является «победителем».

4.1 Выбор начальных значений весовых векторов

Всем весам сети перед началом обучения следует придать начальные значения. Общепринятой практикой при работе с нейронными сетями является присваивание весам небольших случайных значений. При обучении слоя Кохонена случайно выбранные весовые векторы следует нормализовать. Окончательные значения весовых векторов после обучения совпадают с нормализованными входными векторами. Поэтому нормализация перед началом обучения приближает весовые векторы к их окончательным значениям, сокращая, таким образом, обучающий процесс.

4.2 Режим интерполяции

До сих пор мы обсуждали алгоритм обучения, в котором для каждого входного вектора активировался лишь один нейрон Кохонена. Это называется методом аккредитации. Его точность ограничена, так как выход полностью является функцией лишь одного нейрона Кохонена. В методе интерполяции целая группа нейронов Кохонена, имеющих наибольшие выходы, может передавать свои выходные сигналы в слой Гроссберга. Число нейронов в такой группе должно выбираться в зависимости от задачи, и убедительных данных относительно оптимального размера группы не имеется. Как только группа определена, ее множество выходов NET рассматривается как вектор, длина которого нормализуется на единицу делением каждого значения NET на корень квадратный из суммы квадратов значений NET в группе. Все нейроны вне группы имеют нулевые выходы.

Метод интерполяции способен устанавливать более сложные соответствия и может давать более точные результаты. По-прежнему, однако, нет убедительных данных, позволяющих сравнить режимы интерполяции и аккредитации.

4.3 Статистические свойства обученной сети

Метод обучения Кохонена обладает полезной и интересной способностью извлекать статистические свойства из множества входных данных. Как показано Кохоненом [8], для полностью обученной сети вероятность того, что случайно выбранный входной вектор (в соответствии с функцией плотности вероятности входного множества) будет ближайшим к любому заданному весовому вектору, равна $1/k$, где k – число нейронов Кохонена.

5 Обучение слоя Гроссберга

Слой Гроссберга обучается относительно просто. Входной вектор, являющийся выходом слоя Кохонена, подается на слой нейронов Гроссберга, и выходы слоя Гроссберга вычисляются, как при нормальном функционировании. Далее, каждый вес корректируется лишь в том случае, если он соединен с нейроном Кохонена, имеющим ненулевой выход. Величина коррекции веса пропорциональна разности между весом и требуемым выходом нейрона Гроссберга, с которым он соединен. В символьной записи

$$v_{ij}^{(new)} = v_{ij}^{(old)} + \beta(y_j - v_{ij}^{(old)})k_i, \quad (4.8) \quad (7)$$

где k_i – выход i -го нейрона Кохонена (только для одного нейрона Кохонена он отличен от нуля); y_j – j -ая компонента вектора желаемых выходов.

Первоначально β берется равным ~ 0.1 и затем постепенно уменьшается в процессе обучения. Отсюда видно, что веса слоя Гроссберга будут сходиться к средним величинам от желаемых выходов, тогда как веса слоя Кохонена обучаются на средних значениях входов. Обучение слоя Гроссберга – это обучение с учителем, алгоритм располагает желаемым выходом, по которому он обучается. Обучающийся без учителя, самоорганизующийся слой Кохонена дает выходы в недетерминированных позициях. Они отображаются в желаемые выходы слоем Гроссберга.

Рис. 2: Полная сеть встречного распространения

6 Сеть встречного распространения полностью

На рис. 2 показана сеть встречного распространения целиком. В режиме нормального функционирования предъявляются входные векторы \vec{X} и \vec{Y} , и обученная сеть дает на выходе векторы \vec{X}' и \vec{Y}' , являющиеся аппроксимациями соответственно для \vec{X} и \vec{Y} . Векторы \vec{X} и \vec{Y} предполагаются здесь нормализованными единичными векторами, следовательно, порождаемые на выходе векторы также будут иметь тенденцию быть нормализованными.

В процессе обучения векторы \vec{X} и \vec{Y} подаются одновременно и как входные векторы сети, и как желаемые выходные сигналы. Вектор \vec{X} используется для обучения выходов \vec{X}' , а вектор \vec{Y} – для обучения выходов \vec{Y}' слоя Гроссберга. Сеть встречного распространения целиком обучается с использованием того же самого метода, который описывался для сети прямого действия. Нейроны Кохонена принимают входные сигналы как от векторов \vec{X} , так и от векторов \vec{Y} . Но это неотлично от ситуации, когда имеется один большой вектор, составленный из векторов \vec{X} и \vec{Y} , и не влияет на алгоритм обучения.

В качестве результирующего получается единичное отображение, при котором предъявление пары входных векторов порождает их копии на выходе. Это не представляется особенно интересным, если не заметить, что предъявление только вектора \vec{X} (с вектором \vec{Y} , равным нулю) порождает как выходы \vec{X}' , так и выходы \vec{Y}' . Если F – функция, отображающая \vec{X} в \vec{Y}' , то сеть аппроксимирует ее. Также, если F обратима, то предъявление только вектора \vec{Y} (приравнивая \vec{X} нулю) порождает \vec{X}' . Уникальная способность порождать функцию и обратную к ней делает сеть встречного распространения полезной в ряде приложений.

Рис. 2 в отличие от первоначальной конфигурации [5] не демонстрирует противоток в сети, по которому она получила свое название. Такая форма выбрана потому, что она также иллюстрирует сеть без обратных связей и позволяет обобщить понятия, развитые в предыдущих главах.

7 Приложение: сжатие данных

В дополнение к обычным функциям отображения векторов встречное распространение оказывается полезным и в некоторых менее очевидных прикладных областях. Одним из наиболее интересных примеров является сжатие данных.

Сеть встречного распространения может быть использована для сжатия данных перед их передачей, уменьшая тем самым число битов, которые должны быть переданы. Допустим, что требуется передать некоторое изображение. Оно может быть разбито на подизображения S , как показано на рис. 3. Каждое подизображение разбито на пиксели (мельчайшие элементы изображения). Тогда каждое подизображение является вектором, элементами которого являются пиксели, из которых состоит подизображение. Допустим для простоты, что каждый пиксель – это единица (свет) или нуль (чернота). Если в подизображении имеется p пикселей, то для его передачи потребуется p бит. Если допустимы некоторые искажения, то для передачи типичного изображения требуется существенно меньшее число битов, что позволяет передавать изображения быстрее. Это возможно из-за статистического распределения векторов подизображений. Некоторые из них встречаются часто, тогда как другие встречаются так редко, что могут быть грубо аппроксимированы. Метод, называемый векторным квантованием, находит более корот-

Рис. 3: Система сжатия изображений

кие последовательности битов, наилучшим образом представляющие эти подизображения.

Сеть встречного распространения может быть использована для выполнения векторного квантования. Множество векторов подизображений используется в качестве входа для обучения слоя Кохонена по методу аккредитации, когда лишь выход одного нейрона равен 1. Веса слоя Гроссберга обучаются выдавать бинарный код номера того нейрона Кохонена, выход которого равен 1. Например, если выходной сигнал нейрона 7 равен 1 (а все остальные равны 0), то слой Гроссберга будет обучаться выдавать 00...000111 (двоичный код числа 7). Это и будет являться более короткой битовой последовательностью передаваемых символов.

На приемном конце идентичным образом обученная сеть встречного распространения принимает двоичный код и реализует обратную функцию, аппроксимирующую первоначальное подизображение. Этот метод применялся как к речи, так и к изображениям, с коэффициентом сжатия данных от 10:1 до 100:1. Качество было приемлемым, хотя некоторые искажения данных на приемном конце неизбежны.

8 Обсуждение

Роберт Хехт-Нильсон, создатель сети встречного распространения (СВР), осознавал ее ограничения: «СВР, конечно, уступает обратному распространению в большинстве приложений, связанных с сетевыми отображениями. Ее преимущества в том, что она проста и дает хорошую статистическую модель для своей среды входных векторов» ([5], с. 27). К этому можно добавить, что сеть встречного распространения быстро обучается, и при правильном использовании она может сэкономить значительное количество машинного времени. Она полезна также для быстрого моделирования систем, где большая точность обратного распространения вынуждает отдать ему предпочтение в окончательном варианте, но важна быстрая начальная аппроксимация. Возможность породить функцию и обратную к ней также нашло применение в ряде систем.

Список литературы

- [1] DeSieno D. 1988. Adding a conscience to competitive learning Proceedings of the IEEE International Conference on Neural Networks, pp. 117-24. San Diego, CA: SOS Printing.
- [2] Grossberg S. 1969. Some networks that can learn, remember and reproduce any number of complicated space-time patterns. *Journal of Mathematics and Mechanics*, 19:53-91.
- [3] Grossberg S. 1971. Embedding fields: Underlying philosophy, mathematics, and applications of psycho-logy, physiology, and anatomy. *Journal of Cybernetics*, 1:28- 50.
- [4] Grossberg S. 1982. *Studies of mind and brain*. Boston: Reidel.
- [5] Hecht-Nielsen R. 1987a. Counterpropagation networks. In Proceedings of the IEEE First International Conference on Neural Networks, eds. M. Caudill and C. Butler, vol. 2, pp. 19-32. San Diego, CA: SOS Printing.

- [6] Hecht-Nielsen R. 1987b. Counterpropagation networks. *Applied Optics* 26(23): 4979-84.
- [7] Hecht-Nielsen R. 1988. Applications of Counterpropagation networks. *Neural Networks* 1: 131-39.
- [8] Kohonen T. 1988. Self-organization and associative memory. 2d ed. New-York, Springer-Verlag.

Следующая лекция

Лекция 18. Стохастические методы