

Список литературы

- [1] Виноградов И.М. Основы теории чисел. М.: Наука, 1965.
- [2] Diffie W., Hellmann M. New directions in cryptography // IEEE Transaction on Information Theory IT-22. 1976.
- [3] Саломая А. Криптография с открытым ключом. М.: Мир, 1996.
- [4] Stinson D.R. Cryptography: Theory and Practice. CRC Press, 1995.
- [5] Adleman L., Manders K., Miller G. On Taking Roots in Finite Fields // 20th IEEE FOCS. Vol. 20. 1977.
- [6] Rabin M.O. Digitalized Signatures and Public Key Functions as Intractable as factorization // MIT Laboratory for Computer Science. January, 1979. TR 212.
- [7] Гэри М., Джонсон Д. Вычислительные машины и труднорешаемые задачи. М.: Мир, 1982.
- [8] Berlekamp E.R. Factoring Polynomials over Large Finite Fields // Mathematics of Computation. Vol. 24. 1970.
- [9] Menezes A.J., van Oorschot P.C., Vanstone S.A. Handbook of Applied Cryptography. CRC Press, 1999.

Обзор основных нейросетевых моделей

В.В. Псиола

Введение

Людей всегда интересовало собственное мышление, устройство нервной системы. Уже в конце средних веков ученые имели общее представление о нервной системе человека. Во второй половине XIX века существовали две школы: ретикуляристы (reticularists) – считавшие, что нервная система представляет сеть из непрерывных нервных волокон, покрывающая все тело человека, и нейронисты (neuronists) – считавшие, что существует огромное число отдельных связанных клеток – нейронов.

В 1880 году К. Голди (Camilo Goldi) разработал новую технику окрашивания нервных волокон, и в 1888 доктор С.Р. Каджал (Santiago Ramon y Cajal) с помощью этой техники доказал несостоятельность модели ретикуляристов. В 1906 Голди и Каджал получили нобелевскую премию в медицине за свое открытие. Это принято считать началом современной науки о нервной системе человека [1].

Детальное изучение структуры нейрона (особенно после изобретения электронного микроскопа) позволило выделить некоторые основные его части: тело клетки, аксон, дендриты, синаптические связи. Подсчитать количество нейронов в мозге человека очень сложно, лишь приблизительно можно оценить количество нейронов в коре головного мозга как 3×10^{10} , общее количество нейронов как 10^{11} , количество синаптических связей как 10^{15} . Скорость передачи импульсов между нейронами головного мозга можно оценить как 0,5–2 м/с [1].

Искусственные нейронные сети появились как попытка моделирования деятельности мозга человека. Нейронные сети состоят из элементов, функциональные возможности которых аналогичны большинству элементарных функций биологического нейрона. Эти элементы организованы в структуры, которые некоторым образом приближают анатомию мозга.

Определения

В настоящее время существует множество нейросетевых парадигм. На данный момент нет строгого определения нейросетевой парадигмы. Одним из приемлемых определений может быть следующее:

Нейросетевой парадигмой (или просто *нейронной сетью*) называется тройка: топология, алгоритм функционирования и алгоритм обучения.

Топология нейронной сети определяется как взвешенный ориентированный граф. Вершины этого графа называются *нейронами*, а ориентированные ребра – *связями* или *синаптическими связями*. Веса, приписанные ребрам, называются *весами связей*. Некоторые нейроны этого графа выделяются как *входные*, некоторые как *выходные*, остальные нейроны называются *внутренними*. Считается, что на входные нейроны подается входной сигнал, а с выходных нейронов считывается выход нейросети. В каждый момент времени каждой вершине графа (нейрону) приписано некоторое число называемое *уровнем активации нейрона* или *состоянием нейрона*. Начальная инициализация весов связей и уровней активации полностью определяется парадигмой. В каждый фиксированный момент времени множество состояний входных нейронов называется *входной информацией*, множество состояний выходных нейронов – *выходной информацией* или *ответом нейросети*, множество состояний внутренних нейронов – *состоянием нейросети*. В некоторых моделях множество всех нейронов разбивается на группы, которые называются *слоями нейросети*. Обычно предполагается, что все слои

можно упорядочить таким образом, что от всех нейронов i -го слоя идут связи только к нейронам $(i + 1)$ -го слоя.

Алгоритм функционирования – это правила изменения уровней активации нейронов со временем. Считается, что изменения уровней активации входных нейронов определяются решаемой задачей, а не правилами функционирования. *Входным вектором* нейрона будем называть набор значений уровней активации тех нейронов, от которых ведет связь к данному нейрону. В подавляющем большинстве случаев состояние нейрона зависит только от состояний связанных с ним нейронов на предыдущем шаге и от весов этих связей. При этом уровень активации нейрона вычисляется по формуле $x_i(t+1) = f\left(\sum_j w_{ji}x_j(t)\right)$, где w_{ji} – вес связи от j -го нейрона к i -му, $x_i(t)$ – состояние i -го нейрона в момент времени t , а f – некоторая функция, которая называется *функцией активации*.

Алгоритм обучения – это правила изменения весов связей нейронной сети в зависимости от значений уровней активации нейронов.

В случае дискретной модели эти правила обычно заданы алгоритмически, в случае аналоговой модели – системой дифференциальных уравнений.

Работу нейронной сети обычно разбивают на два этапа: *этап обучения* и *этап функционирования*. Функционирование нейронной сети начинается с этапа обучения. На этом этапе происходит изменение уровней активации и весов связей одновременно по заданным правилам функционирования и обучения. Время окончания этапа обучения определяется из алгоритма обучения. Далее начинается этап функционирования, когда по правилам функционирования изменяются только уровни активации нейронов и остаются неизменными веса связей.

Выделяют два класса нейросетевых парадигм. Первый – нейронные сети, которые обучаются *без учителя* (без *учительской информации*), второй – *с учителем* (с *учительской информацией*). В первом случае алгоритм обучения (изменения весов связей) в каждый

момент времени зависит только от состояния нейронной сети и входной информации. Во втором случае необходим еще «желательный ответ нейронной сети». То есть, на этапе обучения в каждый момент времени помимо входной информации, считается заданной «желательная» или «требуемая» выходная информация. И эта информация используется в алгоритме обучения существенным образом.

На данный момент класс объектов, которые называют искусственными нейронными сетями, настолько обширен, что некоторые из этих объектов не отвечают этому определению, однако многие «классические» нейросетевые парадигмы определяются именно таким образом.

Развитие искусственных нейронных сетей

В 1943 году У. МакКаллок (Warren McCulloch) и У. Питтс (Walter Pitts) предложили общую теорию обработки информации, основанную на нейронных сетях [2]. В качестве активационной функции в своей модели они использовали пороговую функцию («все или ничего»). В своей работе они показали, что с помощью предложенной ими модели может быть решен широкий спектр задач. «Программой» для такой вычислительной модели можно считать матрицу весов синаптических связей. Основная проблема заключается в нахождении этой матрицы.

В 1949 году О. Хеббом (Donald O. Hebb) была разработана модель обучения нейронов человеческого мозга [7].

В 1958 году Ф. Розенблат (Frank Rosenblatt) с коллегами разработали примитивную модель работы мозга – *персептрон* (от слова *perception* – восприятие, ощущение, понимание). Группа Розенבלата разработала алгоритм обучения этой модели для простейшего случая (*двухслойный персептрон*) [3].

В 1961 году Э. Кайяньелло (Eduardo Caianiello) предложил алгоритм для определения матрицы весов для модели МакКаллока и Питтса. Это была примитивная разновидность правила обучения по Хеббу [1].

В 1969 году М. Минский (Marvin Minsky) и С. Пейперт (Seymour Papert) в своей книге «Персептроны» [6] детально проанализировали алгоритм обучения персептрона, предложенный Розенблатом, и показали, что предложенная модель имеет существенные ограничения. Например, двухслойный персептрон в принципе не может решить проблему *XOR* (исключающего *ИЛИ*). К возможности развития персептронной модели авторы отнеслись скептически. Блеск и строгость аргументации Минского, а также его авторитет породили огромное доверие к книге и разочарование исследователей в области нейронных сетей. Правительства перераспределили свои субсидии, и нейронные сети были практически забыты на 2 десятилетия. Однако некоторые исследователи продолжали работать в этой области.

В 1974 году В. Литл (W. Little) провел аналогию между моделью МакКаллока–Питтса и системой элементарных магнитных моментов (спинов). Подобная аналогия помогла сделать глубокий теоретический анализ проблемы. Впоследствии, развивая эту модель, существенных результатов добились Г. Шоу (Gordon Shaw) и Дж. Хопфилд (John Hopfield).

В 1982 году Хопфилд предложил модель ассоциативной памяти и опубликовал некоторые свойства запоминания нейронных сетей и некоторых физических моделей [12].

В 1982 году Т. Кохонен (Teuvo Kohonen) предложил модель *самоорганизующейся топологической карты*, которая внесла существенный вклад в развитие нейронных сетей [10].

В 1985 году несколько групп ученых почти одновременно пришли к идее обучения многослойных сетей прямого распространения [1].

С этого момента началась новая волна интереса к искусственным нейронным сетям, возникло огромное количество публикаций и периодических изданий. Например, в 1987 году было проведено 4 крупных совещания по нейронным сетям и опубликовано свыше 500 научных сообщений [1].

Персептрон

В 1958 году Розенблатом было предложено одно из первых (по его мнению) устройств, реализующих процесс распознавания визуальных образов. Исходя из устройства зрительного тракта человека, Розенблат предложил свою модель. В качестве входных данных для нейросети использовался некоторый аналог сетчатки человеческого глаза. Каждая клетка сетчатки могла принимать значения 1 (есть сигнал) или 0 (нет сигнала).

Топология

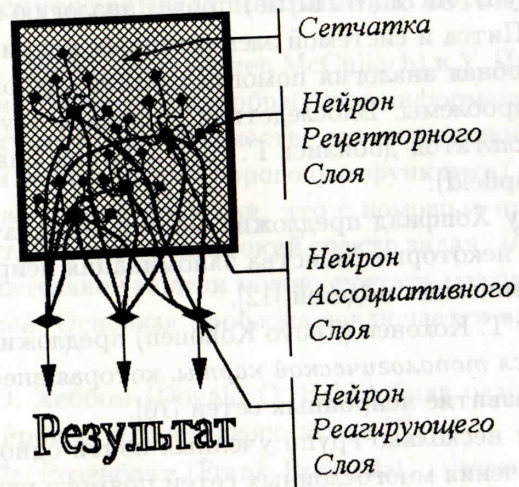


Рис. 1. Персептрон.

В модели определены три слоя. Первый, рецепторный слой, передает электрические импульсы от освещенных элементов сетчатки на следующий, ассоциативный слой. Выходы нейронов ассоциативного слоя соединены с нейронами последнего (выходного) реагирующего слоя. От каждого нейрона ассоциативного слоя к каждому реагирующему нейрону ведет связь.

В своей модели персептрона «МАРК I» Розенблат использовал 400 рецепторов, 512 нейронов ассоциативного слоя, каждый из которых имел 10 входов, случайным образом соединенных с рецепторами [1, 3, 5].

Функционирование

Изменение уровней активации нейронов происходит по формуле $x_i(t+1) = f\left(\sum_j w_{ji}x_j(t)\right)$. Функция активации в модели выбирается пороговой: $f(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases}$. Вес связи от рецепторного нейрона к ассоциативному всегда равен +1 или -1. Вес связи от ассоциативного нейрона к реагирующему равен некоторому действительному числу. Зависимость каждого нейрона ассоциативного слоя от рецепторов определяется случайным образом и не меняется впоследствии. В модели персептрона «МАРК I» из 10 входов, ведущих к каждому ассоциативному нейрону, 5 выбирались с весом -1 и 5 - с весом +1. Выбор осуществлялся случайным образом. Перед обучением веса связей от ассоциативных нейронов к реагирующим устанавливались равными 0 [1, 3, 5].

Обучение

Обучение персептрона является «обучением с учителем» и происходит за счет изменения весов реагирующего слоя. То есть в процессе обучения изменяются только веса связей от ассоциативных нейронов к реагирующим. Веса связей от рецепторных нейронов к ассоциативным остаются неизменными. Таким образом, рецепторы можно вынести за модель обучения и нейроны ассоциативного слоя рассматривать как входные.

Обозначения:

m - размерность входного вектора.

k - количество выходных нейронов.

$w_{ir}(t)$ – значение веса i -го входа r -го нейрона на шаге обучения t .
 $y_i(t)$ – значение компоненты входного вектора, поступающей на i -й вход нейрона на шаге обучения t .

Персептрон относит входной вектор к классу j , если $\sum_{i=1}^m w_{ij}y_i \geq 0$,

0, $\sum_{i=1}^m w_{in}y_i < 0$, $\forall n : n \neq j$, $1 < n < k$. Обучение персептрона

происходит итеративным образом. Пусть на шаге обучения t вектор

y относится к классу j , тогда

если $\sum_{i=1}^m w_{ij}(t)y_i(t) \geq 0$ – веса не меняются,

если $\sum_{i=1}^m w_{in}(t)y_i(t) < 0$, $n \neq j$ – веса не меняются,

если $\sum_{i=1}^m w_{ij}(t)y_i(t) < 0$, то $w_{ij}(t+1) = w_{ij}(t) + y_i(t)$,

если $\sum_{i=1}^m w_{in}(t)y_i(t) \geq 0$, $n \neq j$, то $w_{in}(t+1) = w_{in}(t) - y_i(t)$.

В 1960 году американский ученый А. Новиков доказал, что персептрон может обучиться на любую решаемую им задачу за конечное число шагов.

Теорема 1 ([5]). Пусть дана бесконечная ограниченная по модулю последовательность векторов $y_1, y_2, \dots, y_i, \dots$, принадлежащих множествам $\{y\}$ и $\{\bar{y}\}$, $\sup_{y \in \{y\} \cup \{\bar{y}\}} |y| = D < \infty$. Пусть существует гиперплоскость, проходящая через начало координат и разделяющая множества $\{y\}$ и $\{\bar{y}\}$, и существует вектор нормали к этой гиперплоскости N такой, что $(y_i, N) > \rho_0 \forall y_i \in \{y\}$ и $(y_i, N) < -\rho_0 \forall y_i \in \{\bar{y}\}$. Тогда при использовании персептронной процедуры построения разделяющей гиперплоскости с начальными весами, равными 0, число исправлений ошибок не превзойдет числа $k = \left\lceil \frac{D^2}{\rho_0^2} \right\rceil$.

Анализ Минского

Обычно, когда упоминается анализ Минского, говорят о проблеме *XOR*, но его анализ гораздо глубже. Проблема *XOR* легко решается трехслойным персептроном. Минский указал на целый ряд ограничений персептронной модели, которые обычно упускают из виду.

Было показано, что единообразие программирования и простота обучения, которые так привлекают исследователей, приводят к тому, что персептрон может обучиться распознавать только очень узкий класс свойств геометрической фигуры. Например, персептрон не может распознавать связность фигуры, четность числа точек, входящих в нее. Также Минский продемонстрировал, что наряду с возможностью распознавания персептроном простых геометрических фигур (треугольник, прямоугольник и т.п.), персептрон не может распознавать эти фигуры в контексте других.

Было показано, что обычно исследователями уделяется мало внимания информационному содержанию значений весов. Были приведены примеры (причем скорее типичные, чем исключительные), в которых отношение наибольшего значения веса к наименьшему значению бессмысленно велико.

Был серьезно исследован вопрос о времени сходимости процедуры обучения, и было показано, что в некоторых ситуациях, представляющих определенный геометрический интерес, время сходимости растет даже быстрее, чем показательная функция от количества точек распознаваемой фигуры.

Исследовался вопрос параллельных вычислений, который так привлекал многих исследователей персептронной модели, и было показано, что зачастую параллельность себя не оправдывает, так как суммарное количество операций несоизмеримо велико по сравнению с последовательными процессами, приводящими к тем же результатам.

Одним из серьезных достоинств персептронной модели считалась ее простота воплощения в аналоговых устройствах. Минский

показал, что зачастую, когда дело доходит до реальных вычислительных задач, это воплощение практически невозможно.

Также он считал, что аналогия подобных устройств с мозгом человека не совсем корректна.

В общем, он дал пессимистическую оценку развития этой области [6].

Нейросеть прямого распространения

Развитием перцептронной модели можно считать нейросети прямого распространения (feed forward network) [1, 4, 8, 20].

Топология

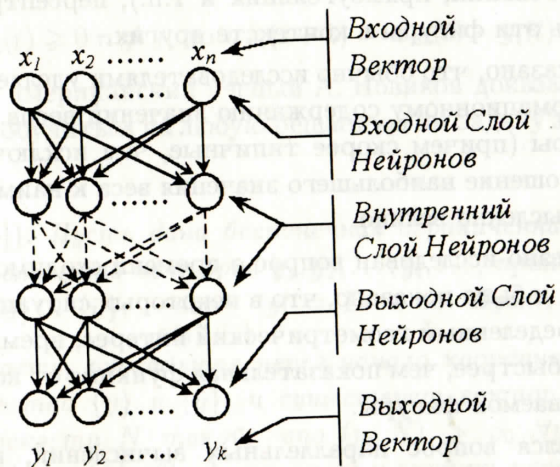


Рис. 2. Сеть прямого распространения.

В случае сети прямого распространения сигнала без обратных связей топологию можно описать следующим образом: множество всех вершин графа (нейронов) разбивается на слои. Все входные нейроны (и только они) объединяются во *входной слой*, все выходные (и

только они) объединяются в *выходной слой*. Слой нейронов, который не является ни входным, ни выходным, называется *внутренним слоем*. На множестве слоев введена нумерация, входной слой считается первым, выходной – последним. В каждую вершину n -го (кроме $n = 1$) слоя ведут ребра из всех вершин $(n - 1)$ -го слоя. Других связей нет (рис. 2).

Функционирование

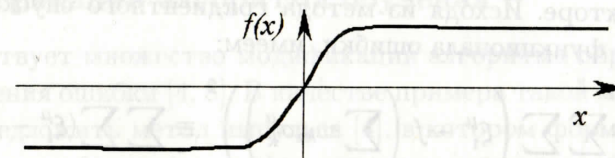


Рис. 3. Функция сигмоидного типа.

Каждой связи приписан вес – некоторое действительное число. По каждой связи в нейрон передается импульс, который вычисляется как произведение выходного импульса нейрона, из которого выходит эта связь, и веса связи. Во входные нейроны импульс передается «извне». Нейрон представляет собой сумматор входных импульсов и активационную функцию, которая вычисляет активационный уровень нейрона (выходной импульс). То есть значение уровня активации нейрона вычисляется по формуле $x_i(t + 1) = f\left(\sum_j w_{ji}x_j(t)\right)$. В качестве активационной функции берут, как правило, функцию сигмоидного типа (рис 3), например: $f(x) = \text{th}(\beta x) = \frac{1 - e^{-2\beta x}}{1 + e^{-2\beta x}}$. Могут быть применены также кусочно-линейные функции.

Обучение

Во второй половине 80-х был разработан алгоритм обучения многослойной сети прямого распространения, который основан на

уменьшении функционала квадратичной ошибки методом градиентного спуска. Этот алгоритм является алгоритмом обучения с учителем и называется «алгоритмом обратного распространения ошибки».

Пусть имеется конечное обучающее множество входных векторов $\bar{\sigma}^\mu = \{\sigma_1^\mu, \dots, \sigma_N^\mu\}$ и соответствующие им требуемые ответы каждого выходного нейрона ξ_j^μ , где j – номер выходного нейрона.

Рассмотрим двухслойную сеть, функционал ошибки равен $D = \frac{1}{2} \sum_{\mu} \sum_i (s_i^\mu - \xi_i^\mu)^2$, где s_i^μ – выход i -го выходного нейрона на μ -ом входном векторе. Исходя из метода градиентного спуска для квадратичного функционала ошибки, имеем:

$$D[w_{ki}] = \sum_{\mu} \sum_i \left(\xi_i^\mu - f \left(\sum_k w_{ki} \sigma_k^\mu \right) \right)^2 \equiv \sum_{\mu} \sum_i (\xi_i^\mu - f(h_i^\mu))^2,$$

$$\frac{\partial D}{\partial w_{ki}} = - \sum_{\mu} (\xi_i^\mu - f(h_i^\mu)) f'(h_i^\mu) \frac{\partial h_i^\mu}{\partial w_{ki}} = - \sum_{\mu} \Delta_i^\mu \sigma_k^\mu,$$

$$\text{где } \Delta_i^\mu \equiv (\xi_i^\mu - f(h_i^\mu)) f'(h_i^\mu).$$

Значит, изменение веса с использованием данного обучающего множества равно $\delta w_{ki} = -\varepsilon \frac{\partial D}{\partial w_{ki}} = \varepsilon \sum_{\mu} \Delta_i^\mu \sigma_k^\mu$, при этом изменение функционала равно $\delta D[w_{ki}] = \sum_{i,k} \frac{\partial D}{\partial w_{ki}} \delta w_{ki} = -\varepsilon \sum_{i,k} \left(\frac{\partial D}{\partial w_{ki}} \right)^2 \leq 0$.

На практике обычно используют последовательный алгоритм изменения весов.

Аналогичным образом можно определить изменение весов при обучении многослойной сети, например, в случае трехслойной сети изменение весов происходит по следующим формулам:

$$\delta w_{ji} = -\varepsilon \frac{\partial D}{\partial w_{ji}} = \varepsilon \sum_{\mu} (\xi_i^\mu - f(h_i^\mu)) f'(h_i^\mu) \frac{\partial h_i^\mu}{\partial w_{ji}} = \varepsilon \sum_{\mu} \Delta_i^\mu s_j^\mu,$$

$$\delta \bar{w}_{kj} = -\varepsilon \frac{\partial D}{\partial \bar{w}_{kj}} = \varepsilon \sum_{\mu,i} (\xi_i^\mu - f(h_i^\mu)) f'(h_i^\mu) \frac{\partial h_i^\mu}{\partial s_j} \frac{\partial s_j}{\partial \bar{w}_{kj}} =$$

$$= \varepsilon \sum_{\mu,i} \Delta_i^\mu w_{ji} f'(h_j^\mu) \frac{\partial \bar{h}_j^\mu}{\partial \bar{w}_{kj}} \equiv \varepsilon \sum_{\mu} \bar{\Delta}_j^\mu \sigma_k^\mu,$$

$\Delta_i^\mu \equiv (\xi_i^\mu - f(h_i^\mu)) f'(h_i^\mu),$
 $\bar{\Delta}_i^\mu \equiv \left(\sum_i \Delta_i^\mu w_{ij} \right) f'(\bar{h}_j^\mu).$

$k \text{ ————— } \text{входной слой}$
 $\downarrow \bar{w}_{kj} \downarrow$
 $j \text{ ————— } \text{внутренний слой}$
 $\downarrow w_{ji} \downarrow$
 $i \text{ ————— } \text{выходной слой}$

Модификации алгоритма обучения

Существует множество модификаций алгоритма обратного распространения ошибки [4, 8]. В качестве примера такой модификации можно предложить метод импульса [4], в котором формулы изменения весов при обучении модифицируются следующим образом:

$$\delta w_{ji}(t) = \varepsilon(t) \sum_{\mu} \Delta_i^\mu(t) s_j^\mu(t) + \alpha(\delta w_{ji}(t-1)),$$

$$\delta \bar{w}_{kj}(t) = \varepsilon(t) \sum_{\mu} \bar{\Delta}_j^\mu(t) \sigma_k^\mu(t) + \alpha(\delta \bar{w}_{kj}(t-1)),$$

где α – коэффициент импульса, обычно устанавливается равным около 0,9. Этот метод хорошо работает на некоторых задачах, но дает слабый или отрицательный эффект на других.

Другой метод, основанный на экспоненциальном сглаживании, может иметь преимущество в ряде приложений [4]:

$$\delta w_{ji}(t) = (1 - \alpha) \sum_{\mu} \Delta_i^\mu(t) s_j^\mu(t) + \alpha(\delta w_{ji}(t-1)),$$

$$\delta \bar{w}_{kj}(t) = (1 - \alpha) \sum_{\mu} \bar{\Delta}_j^\mu(t) \sigma_k^\mu(t) + \alpha(\delta \bar{w}_{kj}(t-1)),$$

где α – коэффициент сглаживания, варьируемый в диапазоне от 0 до 1.

Существует много других модификаций алгоритма обратного распространения ошибки с различными свойствами [1, 4, 8].

Нейронная сеть Хопфилда

Хопфилд сделал важный вклад как в теорию, так и в применение систем с обратными связями. Поэтому его именем названы некоторые модели нейронных сетей.

Топология

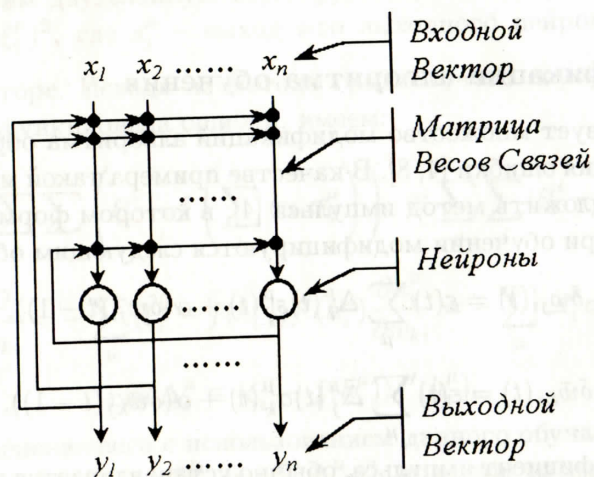


Рис. 4. Сеть Хопфилда.

В отличие от рассмотренных выше парадигм, сети Хопфилда имеют обратные связи, то есть имеют пути от выходов к входам. Отклик таких сетей является динамическим, то есть выход нейронной сети зависит не только от входа и величины связей, но и от состояния нейронной сети, которое в свою очередь зависит от всех предыдущих входных сигналов. Топология сети Хопфилда проста: задается количество нейронов в сети и полносвязный граф, вершинами которого являются эти нейроны. Все нейроны являются одновременно входными, выходными и внутренними. Связям, которые идут от одного нейрона к другому (или к самому себе), приписаны веса. Для простоты сети Хопфилда изображаются таким образом, что связи

отображаются как точки пересечения, а не линии от одного нейрона к другому (рис. 4).

Функционирование

В местах пересечения связей, отмеченных черной точкой (рис. 4), происходит суммирование входного сигнала и выходных сигналов других нейронов, с некоторыми весами. Таким образом, в каждый момент времени на вход i -го нейрона поступает следующий сигнал:

$$s_i(t+1) = x_i(t+1) + \sum_{j=1}^m w_{ji} y_j(t),$$

где $x_i(t)$ – вход i -го нейрона в момент t , $s_i(t)$ – суммарный входной сигнал i -го нейрона в момент t , $y_j(t)$ – выходной сигнал j -го нейрона в момент t , w_{ji} – вес, приписанный связи от j -го нейрона к i -тому.

Способ изображения сети Хопфилда, предложенный на рисунке 4, объясняется еще и тем, что именно таким образом сети Хопфилда реализуются на аппаратном уровне.

По аналогии с системой элементарных магнитных моментов, уровень активации нейрона принимает значение $+1$ или -1 . Выходной сигнал нейрона вычисляется в зависимости от активационной функции: $y_i = f(s_i)$, где f – функция активации.

В различных работах и моделях рассматриваются различные активационные функции [4]. В первой работе Хопфилда [12], например, функция f была просто пороговой функцией:

$$f(s) = \begin{cases} +1, & s > T_i \\ -1, & s \leq T_i \end{cases},$$

где T_i – некоторое пороговое значение, которое выбирается для каждого нейрона отдельно.

Состоянием сети Хопфилда называют набор выходных сигналов всех нейронов в данный момент времени. В работе [12] предполагалось дискретное изменение состояний сети, однако не составляет

труда в терминах дифференциальных уравнений описать алгоритм функционирования сети в случае непрерывного изменения состояний. Для сетей с обратными связями возникает проблема устойчивости, то есть возникает вопрос, наступит ли стабилизация выходных сигналов при отсутствии изменений во входных сигналах или нет. Ответ на этот вопрос был найден в 1983 году [13]. Для доказательства устойчивости выбирается функция энергии E , которая зависит от значений весов и состояния нейросети:

$$E = -\frac{1}{2} \sum_i \sum_j w_{ij} y_i y_j - \sum_j x_j y_j + \sum_j T_j y_j.$$

Нейронная сеть, любое изменение состояния которой не вызывает увеличения функционала Ляпунова E , называется устойчивой.

Утверждение 1 ([13]). Если матрица весов $W = \{w_{ij}\}$ симметрична и все диагональные элементы равны нулю, то нейронная сеть Хопфилда устойчива.

В справедливости этого утверждения легко убедиться, рассмотрим на каждом шаге изменение энергетической функции E . В силу условий, накладываемых на матрицу весов, верно следующее равенство:

$$\delta E = - \left[\sum_{i \neq j} (w_{ij} y_i) + x_j - T_j \right] \delta y_j = -[s_j - T_j] \delta y_j,$$

где δE — изменение значения функции E , вызванное изменением δy_j состояния y_j . Из формулы и правил функционирования сети видно, что множители $[s_j - T_j]$ и δy_j либо одновременно положительны, либо одновременно отрицательны, либо один из них равен нулю. Таким образом, любое изменение любого состояния сети либо не изменяет энергетическую функцию, либо уменьшает ее. Значит, при изменении состояния сети, значение функции энергии уменьшается до тех пор, пока функция не достигнет своего минимума. После этого считается, что сеть перешла в стационарное состояние.

Обучение (модель ассоциативной памяти)

Сети Хопфилда используются во многих задачах. И в зависимости от решаемой задачи выбираются различные алгоритмы обучения. Одной из классических моделей, в которых применяются сети Хопфилда, является модель ассоциативной памяти. Хопфилдом была разработана модель ассоциативной памяти с непрерывно изменяющимися входами в пределах от -1 до $+1$.

Постановка задачи: надо запомнить p N -битовых векторов σ_i^m ($i = 1, \dots, N$, $m = 1, \dots, p$) и потом, для каждого предъявленного вектора n_i надо найти ближайший по Хэммингу из запомненных векторов.

$$H_m = \sum_{i=1}^N (n_i - \sigma_i^m)^2.$$

Значения весов связей устанавливаются по правилу Хебба равными $w_{ij} = \frac{1}{N} \sum_{m=1}^p \sigma_i^m \sigma_j^m$. Результатом работы нейросети являются выходные значения нейронов после того, как сеть придет в стабильное состояние. Используя статистические методы, Хопфилд показал, что в случае, когда запоминаемые вектора некоррелируют, подобная сеть может успешно запомнить количество векторов $p = 0,14 \cdot N$ [1, 12]. Дальнейшее развитие этой модели привело к другим алгоритмам обучения, при которых количество запоминаемых векторов может достигать значения $2N$ [1].

Обучение (задача комивояджера)

Сети Хопфилда успешно используются при решении разнообразных классов задач. Например, в работе [14] предложен метод решения задачи комивояджера с использованием сети Хопфилда. Пусть поставлена задача комивояджера для n городов. Заданы расстояния между всеми городами: d_{ij} — расстояние от i -го города от j -го. Строится сеть Хопфилда с количеством нейронов, равным n^2 . Каждый нейрон снабжен двумя индексами, соответствующими городу и по-

рядковому номеру его посещения в маршруте. Функционирование нейрона задается таким образом, что каждый нейрон может принимать значение 0 или 1. Состояние нейрона, равное единице, означает то, что город, соответствующий этому нейрону, будет стоять в маршруте посещения под номером, соответствующим этому нейрону. Через y_{Xj} обозначается выход (состояние) нейрона, соответствующего городу с номером X и порядковому номеру j . Функционал, минимизация которого соответствует решению задачи коммивояжера, выглядит следующим образом:

$$E = \frac{A}{2} \sum_{X=1}^n \sum_{i=1}^n \sum_{j \neq i} y_{Xi} y_{Xj} + \frac{B}{2} \sum_{i=1}^n \sum_{X=1}^n \sum_{Y \neq X} y_{Xi} y_{Yi} +$$

$$+ \frac{C}{2} \left[\left[\sum_{i=1}^n \sum_{X=1}^n y_{Yi} \right] - n \right]^2 + \frac{D}{2} \sum_{X=1}^n \sum_{Y \neq X} \sum_{i=1}^n [d_{XY} y_{Xi} (y_{Yi-1} - y_{Yi+1})],$$

где A, B, C и D – некоторые константы.

Полная минимизация первого слагаемого обеспечивает то, что каждому городу соответствует один номер в маршруте. Полная минимизация второго слагаемого обеспечивает то, что каждому номеру в маршруте соответствует один город. Полная минимизация третьего слагаемого обеспечивает то, что маршрут состоит из n городов. Минимизация четвертого слагаемого обеспечивает минимальную длину маршрута. Состояние нейросети, которое обеспечивает выполнение первых трех условий, соответствует допустимому маршруту. Подбирая константы A, B, C и D , можно задавать степень важности того или иного условия. Ослабив условия допустимости, можно получить набор «недопустимых маршрутов» меньшей длины, среди которых методом перебора находится допустимый маршрут. Для того чтобы нейронная сеть Хопфилда минимизировала выписанный функционал, надо задать веса связей следующим образом:

$$w_{Xi, Yj} = -A \delta_{X,Y} (1 - \delta_{i,j}) - B \delta_{i,j} (1 - \delta_{X,Y}) - C - D d_{XY} (\delta_{i,j-1} + \delta_{i,j+1}),$$

$$\text{где } \delta_{i,j} = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases}.$$

Кроме того, каждый нейрон соединен с входом, по которому постоянно передается сигнал +1 и вес, приписанный этому соединению, равен Cn . В работе [14] состояния нейронов могли принимать значения от 0 до 1 и активационная функция была выбрана следующим образом: $f(s) = \frac{1}{2(1+\text{th}(s/u_0))}$, где u_0 – некоторое пороговое значение, одинаковое для всех нейронов.

Двунаправленная ассоциативная память

Топология

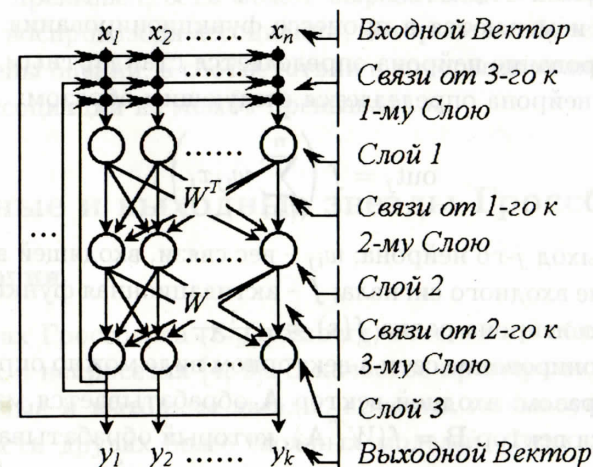


Рис. 5. Двунаправленная ассоциативная память.

Модель двунаправленной ассоциативной памяти является развитием модели ассоциативной памяти [21]. В топологии данной модели есть обратные связи. Нейросеть имеет три слоя, каждый из которых содержит n нейронов. Нейроны первого слоя являются входными, нейроны последнего слоя – выходными. От каждого нейрона первого слоя к каждому нейрону второго слоя ведут связи. От каждого нейрона второго слоя к каждому нейрону третьего слоя также ведут

связи. От каждого нейрона третьего слоя есть связь к соответствующему нейрону первого слоя.

Функционирование

Набор весов между первым и вторым слоями задается матрицей $W^T = \{w_{ji}\}$, где w_{ji} – вес связи от j -го нейрона первого слоя к i -му нейрону второго слоя. Матрица, определяющая веса связей от второго слоя к третьему, является транспонированной к матрице W^T . То есть $W = \{w_{ij}\}$ – матрица связей от второго слоя к третьему, где w_{ij} – вес связи от i -го нейрона второго слоя к j -му нейрону третьего слоя. Связи от третьего слоя к первому имеют единичные веса, которые не изменяются в процессе функционирования и обучения. Функционирование нейрона определяется стандартным образом, то есть выход нейрона определяется следующим образом:

$$\text{out}_j = f \left(\sum_{i=1}^n w_{ij} x_i \right),$$

где out_j – выход j -го нейрона; w_{ij} – вес связи, входящей в j -й нейрон; x_j – значение входного сигнала; f – активационная функция, которая для каждого нейрона равна $f(s) = \frac{1}{1+e^{-\lambda \cdot s}}$.

Функционирование сети в векторном виде можно определить следующим образом: входной вектор \mathbf{A} обрабатывается матрицей W^T и получается вектор $\mathbf{B} = f(W^T \mathbf{A})$, который обрабатывается матрицей W и вновь передается в качестве входа нейросети $\mathbf{A} = f(W\mathbf{B})$.

Обучение

В отличие от модели ассоциативной памяти, двунаправленная модель ассоциативной памяти «запоминает» два вектора: на каждый входной вектор запоминается ассоциированный с ним выходной вектор. Веса устанавливаются таким образом, что если входной вектор равен одному из запомненных или похож на него, то выходной вектор равен вектору, ассоциированному с соответствующим запомненным входным вектором. Из соображений, аналогичным тем, которые

применяются в модели ассоциативной памяти (см. выше), веса нейронной сети определяются следующим образом:

$$W = \sum_{i=1}^k A_i^t B_i,$$

где W – матрица весов, (A_i, B_i) , $i = 1, \dots, k$ – набор пар векторов для запоминания, A_i – входной вектор, B_i – желательный выходной вектор. Как и сеть ассоциативной памяти, сеть двунаправленной ассоциативной памяти имеет ограничение на максимальное количество ассоциаций, которые она может точно воспроизвести. Если этот лимит превышен, сеть может вырабатывать неверный выходной сигнал, воспроизводя ассоциации, которым не обучена. В работе [21] приведены оценки, в соответствии с которыми количество запомненных ассоциаций не может превышать n .

Входные и выходные звезды Гроссберга

Топология

В работах Гроссберга (S. Grossberg) есть много общих идей, используемых в нейросетях [4, 9]. В качестве примера можно рассмотреть входные и выходные звезды, которые используются как составные части других более сложных моделей. Функционирование и правила обучения звезд Гроссберга являются локальными, и поэтому опишем только функционирование одного нейрона, который является звездой Гроссберга, абстрагируясь от всей модели целиком. Для этого достаточно знать, что звезда имеет несколько входных и несколько выходных ребер с приписанными весами.

Функционирование

Значение уровня активации звезды Гроссберга описывается по правилу взвешенного суммирования уровней активации входных

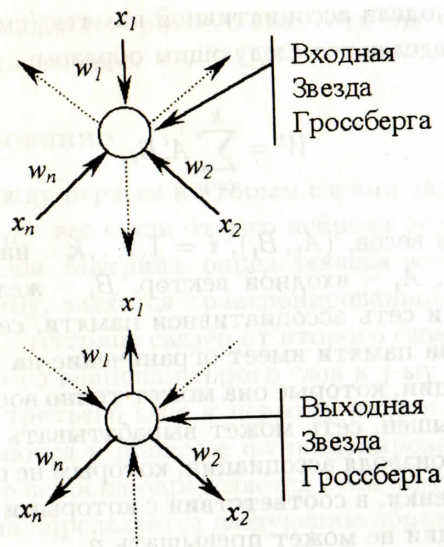


Рис. 6. Звезды Гроссберга.

нейронов:

$$x(t+1) = \sum_j w_j x_j(t).$$

В различных моделях используется еще и некоторая активационная функция, при этом правило изменения уровней активации приобретает следующий вид:

$$x(t+1) = f \left(\sum_j w_j x_j(t) \right),$$

где функция активации f определяется в каждой модели по-своему.

Обучение

При обучении входной звезды изменяются только те веса, которые приписаны входным ребрам, при обучении выходной – выход-

ным (рис. 6). Правила обучения входной звезды Гроссберга выбираются таким образом, чтобы этот нейрон обучался реагировать на определенный входной вектор X и ни на какой другой. Процесс обучения выражается следующим равенством:

$$w_i(t+1) = w_i(t) + \alpha(t)(x_i - w_i(t)),$$

x_i – уровень активации i -го входного нейрона, w_i – вес связи от i -го входного нейрона, $\alpha(t)$ – нормирующий коэффициент обучения, который имеет обычно начальное значение 0,1 и постепенно уменьшается в процессе обучения.

В то время как входная звезда возбуждается при появлении определенного входного вектора, выходная звезда вырабатывает требуемый возбуждающий сигнал для других нейронов всякий раз, когда возбуждается сама. Для реализации этого свойства, правила обучения весов выходной звезды можно взять схожими с правилами обучения входной звезды.

$$w_i(t+1) = w_i(t) + \beta(t)(y_i - w_i(t)),$$

y_i – i -я координата требуемого целевого вектора, w_i – вес i -го выхода нейрона, $\beta(t)$ – нормирующий коэффициент обучения, который имеет обычно начальное значение, приблизительно равное 1, и постепенно уменьшается в процессе обучения.

Самоорганизующаяся карта Кохонена

Топология

Самоорганизующиеся карты Кохонена (Self-Organizing Map) или просто сети Кохонена [10, 11] основаны на той же идее, что и входные звезды Гроссберга, а именно, веса нейронов настраиваются таким образом, чтобы реагировать на определенный входной вектор и на вектора близкие ему. Сети Кохонена часто используют для кластеризации множества объектов. Кластеризация – это разбиение

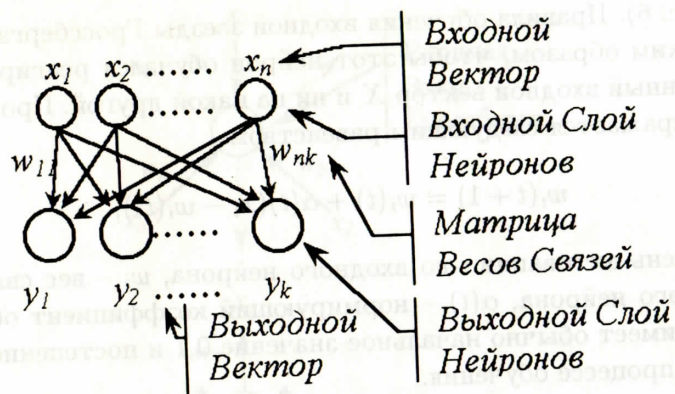


Рис. 7. Самоорганизующаяся карта Кохонена.

множества векторов на группы по некоторому критерию, в данном случае, например, по близости векторов в евклидовой метрике. Таким образом, в сети Кохонена имеется n входных нейронов, на которые подаются вектора n -мерного евклидова пространства, и k выходных нейронов. Все входные нейроны соединены с каждым из выходных (рис. 7).

Функционирование

Результатом решения задачи кластеризации является разбиение n -мерного пространства признаков на k классов, то есть автомат, который по заданному входному n -мерному вектору вырабатывает номер класса, к которому этот вектор следует отнести. Подобным автоматом служит обученная сеть Кохонена. Входной вектор относится к классу j в том случае, если уровень активации j -го выходного нейрона является минимальным среди уровней активации всех выходных нейронов. Уровень активации выходного нейрона вычисляет евклидово расстояние между входным вектором и вектором

весов этого нейрона:

$$o_i = \sum_j (w_{ij} - x_j)^2,$$

x_j – уровень активации j -го входного нейрона, o_i – уровень активации i -го выходного нейрона, w_{ij} – вес связи от i -го входного нейрона к j -му выходному нейрону.

Выходной нейрон с минимальным значением уровня активации называется нейроном-победителем. Для каждого входного вектора выходом нейронной сети Кохонена считается вектор, у которого 1 на месте нейрона-победителя и 0 на всех остальных местах. В некоторых задачах ненулевые значения оставляются для нескольких нейронов-победителей, и значения выхода для этих нейронов берутся обратно пропорциональными к уровням активации соответствующих нейронов.

Обучение

Все выходные нейроны сети обучаются по правилу, похожему на правило обучения звезд Гроссберга, и каждый из них старается научиться реагировать только на определенный вектор и близкие к нему. Однако существенным отличием обучения сети Кохонена является то, что это обучение не является локальным. Нейроны сети Кохонена имеют двойное представление: во-первых, нейрон представляет собой выходной нейрон сети, а во-вторых – точку в некотором r -мерном пространстве (обычно рассматривают \mathbb{R}^2), на так называемой карте Кохонена. На карте Кохонена нейроны расположены в узлах координатной решетки, то есть каждому выходному нейрону сети Кохонена приписана пара целых чисел, которая указывает координаты этого нейрона на карте Кохонена и используется только в процессе обучения. Далее, исходя из предположения, что требуется разбить множество векторов $\{x_1, x_2, \dots, x_M\}$ на k классов, случайным образом устанавливаются начальные веса связей. Затем каждый элемент предметного пространства (обучающий

вектор) подается на вход сети, вычисляются расстояния от входного вектора до весов нейронов (уровень активации нейрона) и находится нейрон-победитель – нейрон с наименьшим расстоянием (уровнем активации).

$$\text{Net}_j(n) = \|\vec{x}(n) - \vec{w}_j(n)\|,$$

$$c(n) = \operatorname{argmin}\{\text{Net}_j(n)\},$$

n – номер входного вектора, $\vec{x}(n)$ – входной вектор, $c(n)$ – номер нейрона победителя, $\vec{w}_c(n)$ – вектор весов нейрона-победителя.

После этого веса выигравшего нейрона изменяются по правилу Гроссберга таким образом, чтобы уменьшить расстояние между данным входным вектором и вектором весов нейрона-победителя:

$$\Delta w_c(n) = h(n)(x_i(n) - w_c(n)),$$

$h(n)$ – коэффициент обучения, который обычно берется равным $A/(B + n)$, где A и B – некоторые константы, которые выбираются в зависимости от специфики решаемой задачи.

Если помимо этих изменений веса нейрона-победителя не производится никаких изменений весов других нейронов, то подобный алгоритм называется *алгоритмом линейной кластеризации векторов* (*Linear Vector Quantization*). Существенным отличием самоорганизующейся карты Кохонена является изменение весов всех остальных нейронов по следующим формулам:

$$\Delta w_j(n) = e_j(n)(x(n) - w_j(n)),$$

$$e_j(n) = h(n)e^{-(d_j/r(n))^2},$$

d_j – расстояние от j -го нейрона до нейрона-победителя в карте Кохонена, $r(n)$ – эффективный радиус окрестности нейрона – некоторый коэффициент, который обычно берется равным $C/(D + n)$, где C и D – некоторые константы, которые выбираются в зависимости от специфики решаемой задачи.

Для упрощения вычислений часто выбирается некоторая окрестность нейрона-победителя на карте Кохонена, вне которой изменения весов нейронов считаются пренебрежительно малыми и веса этих нейронов не пересчитываются.

Образно говоря, после окончания процесса обучения веса нейронов в предметном пространстве «скапливаются» в местах «скопления» данных, при этом сохраняя внутреннюю топологию Кохоненовской карты.

Инициализация весов сети Кохонена обычно производится случайным образом, так чтобы изначально нейроны равномерно покрывали область, в которой предположительно находятся вектора обучения и кластеризации. Так как часто перед решением задачи затруднительно определить такую область, то стандартной считается процедура нормализации весов: к каждому входному вектору добавляется еще одна координата, равная 1, и после этого все координаты вектора делятся на его длину. Эта процедура обеспечивает то, что все входные вектора лежат на единичной сфере в $(n + 1)$ -мерном пространстве. Таким образом, областью, в которой лежат все входные данные, является единичная сфера, и веса сети могут быть инициализированы таким образом, чтобы нейроны равномерно «покрывали» единичную сферу.

Алгоритм встречного распространения

Топология

Сеть встречного распространения является модернизацией однослойной сети. Возможности сети встречного распространения существенно превосходят возможности однослойной сети. Сети встречного распространения решают более частные задачи, чем сети, обучаемые процедурой обратного распространения [4]. Однако время обучения сети уменьшается в сотни раз по сравнению с процедурой обратного распространения.

Сеть встречного распространения состоит из 3-х слоев нейронов.

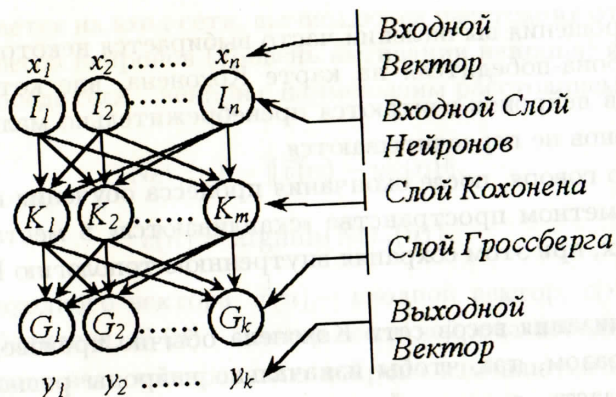


Рис. 8. Сеть встречного распространения.

Количество нейронов в каждом слое произвольно и не зависит друг от друга. Первый слой содержит входные нейроны, и они служат только для разветвления входного сигнала и не изменяют его значения. От каждого нейрона первого слоя к каждому нейрону второго есть связь, вес которой равен w_{ij} . Второй слой называется слоем Кохонена. От каждого нейрона второго слоя к каждому нейрону третьего есть связь, вес которой равен v_{ij} . Третий слой называется слоем Гроссберга. Нейроны третьего слоя являются выходными нейронами. Других связей между нейронами сети нет.

Функционирование

Функционирование нейронной сети встречного распространения описывается как комбинация функционирования слоя Кохонена и слоя Гроссберга. Функционирование слоя Кохонена происходит именно таким образом, как функционирование сети Кохонена (см. выше). Таким образом, выходом этого слоя будет вектор с одной единичной координатой и нулевыми остальными. Иногда рассматривается модель с несколькими ненулевыми выходами (см. выше). Все нейроны слоя Гроссберга функционируют как выходные звезды Гроссберга (см. выше).

Обучение

Обучение сети встречного распространения разделяется на два этапа: обучение слоя Кохонена, которое происходит без использования учительской информации по алгоритму обучения сети Кохонена (см. выше), и обучения слоя Гроссберга, которое происходит по алгоритму обучения выходных нейронов Гроссберга (см. выше). Обучение нейронов Гроссберга происходит по формуле:

$$v_{ij}(t+1) = v_{ij}(t) + (y_j(t) - v_{ij}(t))k_i(t),$$

где $k_i(t)$ – выход i -го нейрона Кохонена на шаге t ; $y_j(t)$ – j -ая компонента вектора желаемых выходов на шаге t .

Обучение и функционирование нейронной сети, определенной таким образом, обеспечивает следующие свойства сети: все входные вектора разбиваются на группы по критерию геометрической близости. Количество групп соответствует количеству нейронов в слое Кохонена. С каждой из групп ассоциируется некоторый вектор, который записывается в соответствующие веса слоя Гроссберга. Таким образом, на этапе функционирования для каждого входного вектора определяется группа принадлежности, и на выход подается вектор, ассоциированный с этой группой. Функционирование сети усложняется, если выходной вектор слоя Кохонена имеет несколько ненулевых компонент (см. описание сети Кохонена). Сети встречного распространения могут применяться в различных задачах, например, с помощью таких сетей может происходить архивация данных [4].

Стохастические методы обучения

Топология

Стохастические методы обучения могут применяться к нейронным сетям с любой топологией, однако в большинстве случаев рассматриваются сети без обратных связей. В качестве типичного при-

мера обычно рассматриваются сети прямого распространения с несколькими слоями [4].

Функционирование

Функционирование нейронной сети определяется стандартным способом, который описан в разделе «Нейросеть прямого распространения».

Обучение

Стохастические методы обучения предполагают псевдослучайные изменения величин весов, сохраняя те изменения, которые ведут к улучшениям. То есть процесс обучения можно представить следующим алгоритмом.

- 1) Случайным образом инициализируются веса.
- 2) Вектора обучающего множества подаются последовательно в качестве входов нейронной сети.
- 3) Для каждого обучающего вектора вычисляется выход нейронной сети.
- 4) Выходы нейронной сети сравниваются с желаемыми выходами, и вычисляется сумма квадратов разностей между ними. Целью обучения является минимизация этой функции, которая называется целевой функцией.
- 5) Случайным образом выбирается один вес нейронной сети и корректируется на небольшое случайное значение.
- 6) Вычисляется целевая функция с учетом коррекции.
- 7) Если коррекция привела к уменьшению целевой функции, то коррекция остается, иначе весу возвращается первоначальное значение (в некоторых моделях изменения, которые привели к увеличению целевой функции, сохраняются с маленькой вероятностью).
- 8) Пункты 3-7 повторяются до тех пор, пока сеть не обучится до требуемой степени.

Все стохастические методы обучения работают по описанному алгоритму, различия между ними заключаются в реализации пунк-

тов 5 и 7. Можно определить некоторые основные методы стохастического обучения:

1) Больцмановское обучение.

Определяется некоторая переменная T , которая описывает искусственную температуру. Изначально температура инициализируется большим начальным значением. Вес, изменение которого происходит на шаге 5, случайно равновероятно выбирается из множества всех весов. Величина случайного изменения веса может выбираться в соответствии с распределением Гаусса:

$$P(w) = \exp(-w^2/T^2),$$

$P(w)$ – вероятность изменения веса на величину w .

В пункте 7 алгоритма изменение веса, которое привело к увеличению целевой функции сохраняется с вероятностью $P(c) = \exp(-c/kT)$, где c – величина изменения целевой функции, k – константа Больцмана, которая выбирается в зависимости от задачи. В соответствии с этим величина изменения веса может быть выбрана, например, методом Монте-Карло. В работе [22] показано, что для того чтобы была достигнута сходимость к глобальному минимуму, скорость уменьшения температуры должна быть обратно пропорциональна логарифму времени.

$$T(t) = T_0 / \log(1 + t),$$

T_0 – начальная искусственная температура, t – искусственное время.

2) Обучение Коши.

Обучение методом Больцмана занимает очень много времени, в связи с этим часто применяют метод Коши, который не дает гарантии сходимости к глобальному минимуму.

$$T(t) = T_0 / (1 + t),$$

T_0 – начальная искусственная температура, t – искусственное время.

$$P(w) = T(t) / [T(t)^2 + w^2],$$

$P(w)$ – вероятность изменения веса на величину w .

3) Метод искусственной теплоемкости.

Этот метод применяется для ускорения метода Коши. Идея метода взята из термодинамики и заключается в том, что вводится искусственная теплоемкость – скорость изменения температуры со временем. При подходе к локальному минимуму теплоемкость увеличивается, обеспечивая, таким образом, сильное скачкообразное поведение сети и выход из локального минимума. В остальных случаях теплоемкость достаточно мала, обеспечивая быстрое стремление сети к минимуму.

4) Комбинированные методы.

Существует множество моделей комбинированных методов [4], когда детерминированные алгоритмы, например, алгоритм обратного распространения обучения нейронных сетей объединяются со стохастическими методами. В этом случае детерминированные методы обеспечивают высокую скорость обучения, а стохастические методы – выход из локальных минимумов целевой функции.

Список литературы

- [1] Muller B., Reinhardt J., Strickland M.T. Neural Networks. An Introduction.
- [2] McCulloch W.S., Pitts W. A logical calculus of the ideas immanent in nervous activity // Bulletin of Mathematic Biophysics. 1943. 5. P. 115–133.
- [3] Rosenblatt F. The perceptron: a probabilistic model for information storage and organization in the brain // Psychological Review. 1958. 65. P. 386–408.
- [4] Уоссермен Ф. Нейрокомпьютерная техника. Теория и практика. М.: Мир, 1992.
- [5] Кольцов П.П. Математические модели классификации больших объемов информации. М., 1984.

- [6] Minsky M., Papert S. Perceptrons. An Introduction to Computational Geometry. Massachusetts Institute of Technology: The MIT Press.
- [7] Hebb D.O. The organization of behavior. New York: Science Edition, 1961.
- [8] Hristev R.M. The ANN book.
- [9] Grossberg S. Adaptive pattern classification and universal recoding: I. Parallel development and coding of neural feature detectors // Biological Cybernetics. 1976. 23. P. 121–134.
- [10] Kohonen T. Self-organized formation of topologically correct feature maps // Biological Cybernetics. 1982. 43. P. 59–69.
- [11] Kohonen T. The Self-organized map // Proc. IEEE. 1990. V. 78. № 9. P. 1464–1480.
- [12] Hopfield J.J. Neural Network and physical systems with emergent collective computational abilities // Proc. of the National Academy of Sciences. 1982. 79. P. 2554–2558.
- [13] Cohen M.A., Grossberg S.G. Absolute stability of global pattern formation and parallel memory storage by competitive neural networks // IEEE Transaction on System, Man and Cybernetics. 1983. 13. P. 815–826.
- [14] Hopfield J.J., Tank D.W. Neural computations of decisions in optimization problems // Biological Cybernetics. 1985. 52. P. 141–152.
- [15] Grossberg S. The adaptive brain 1, 2. Amsterdam: North-Holland, 1987.
- [16] Abu-Mostafa Y.S, St. Jacques J. Information capacity of the Hopfield model // IEEE Transaction on Information Theory. 1985. 31(4). P. 461–464.
- [17] Carpenter G., Grossberg S. Adaptive resonance theory: Stable self-organization of neural recognition codes in response to arbitrary list of input patterns // Proc. 8th Annual Conf. Cognitive Sci. Soc. 1986. P. 45–62.

- [18] Carpenter G., Grossberg S. ART2: Self-organization of stable category recognition codes for analog input patterns // Appl. Optics. 1987. V. 26. №. 23. P. 4919–4930.
- [19] Carpenter G., Grossberg S. ART3: Hierarchical search using chemical transmitters in self-organization pattern recognition architectures // Neural Networks. 1990. V. 3. P. 129–152.
- [20] Block H.D. The Perceptron: a model for brain function // Review of Modern Physics. 1962. 34. P. 123–135.
- [21] Kosko B. Bi-directional associative memories // IEEE Transaction on Systems, Man and Cybernetic. 1987. 18(1). P. 49–60.
- [22] German S., German D. Stochastic relaxation, Gibbs distributions and Bayesian restoration of images // IEEE Transaction on Pattern Analysis and Machine Intelligence PAMI-6. 1984. P. 721–741.

Сферическая модель когнитивных и исполнительных механизмов

Е.Н. Соколов

1. Структура сферической модели

Основной проблемой психофизиологии является соотношение субъективных феноменов, поведенческих актов и процессов, протекающих в нейронных сетях. С тем, чтобы приблизиться к решению этой задачи и интегрировать нейрональную активность с её субъективными проявлениями, была предложена модель, представляющая собой гиперсферу в n -мерном пространстве. Сферические координаты (углы гиперсферы) соответствовали субъективным шкалам, тогда как декартовы координаты модели соотносились с возбуждениями нейронов, участвующих в возникновении субъективных явлений [1].

Сферическая модель описывает не только когнитивные процессы, но и реализацию поведенческих актов. В случае представления когнитивных процессов модель охватывает восприятие, память и семантику. Распространение модели на реализацию поведенческих актов основано на представлении о том, что множество субъективных интенций также образует гиперсферу. Концепция интенций была разработана представителями Вюрцбургской школы психологии, рассматривавшими интенции как ключевой элемент инициации произвольных актов.

Когнитивная гиперсфера образована множеством нейронно-детекторов и гностических единиц, представляющих когнитивные события активацией локальных участков гиперсферы.

- [18] Carpenter G., Grossberg S. ART2: Self-organization of stable category recognition codes for analog input patterns // Appl. Optics. 1987. V. 26. №. 23. P. 4919–4930.
- [19] Carpenter G., Grossberg S. ART3: Hierarchical search using chemical transmitters in self-organization pattern recognition architectures // Neural Networks. 1990. V. 3. P. 129–152.
- [20] Block H.D. The Perceptron: a model for brain function // Review of Modern Physics. 1962. 34. P. 123–135.
- [21] Kosko B. Bi-directional associative memories // IEEE Transaction on Systems, Man and Cybernetic. 1987. 18(1). P. 49–60.
- [22] German S., German D. Stochastic relaxation, Gibbs distributions and Bayesian restoration of images // IEEE Transaction on Pattern Analysis and Machine Intelligence PAMI-6. 1984. P. 721–741.

Сферическая модель когнитивных и исполнительных механизмов

Е.Н. Соколов

1. Структура сферической модели

Основной проблемой психофизиологии является соотношение субъективных феноменов, поведенческих актов и процессов, протекающих в нейронных сетях. С тем, чтобы приблизиться к решению этой задачи и интегрировать нейрональную активность с её субъективными проявлениями, была предложена модель, представляющая собой гиперсферу в n -мерном пространстве. Сферические координаты (углы гиперсферы) соответствовали субъективным шкалам, тогда как декартовы координаты модели соотносились с возбуждениями нейронов, участвующих в возникновении субъективных явлений [1].

Сферическая модель описывает не только когнитивные процессы, но и реализацию поведенческих актов. В случае представления когнитивных процессов модель охватывает восприятие, память и семантику. Распространение модели на реализацию поведенческих актов основано на представлении о том, что множество субъективных интенций также образует гиперсферу. Концепция интенций была развита представителями Вюрцбургской школы психологии, рассматривавшими интенции как ключевой элемент инициации произвольных актов.

Когнитивная гиперсфера образована множеством нейронно-детекторов и гностических единиц, представляющих когнитивные события активацией локальных участков гиперсферы.