

ПЕРСЕПТРОНЫ И НЕЙРОННЫЕ СЕТИ ГЛУБОКОГО ДОВЕРИЯ: ОБУЧЕНИЕ И ПРИМЕНЕНИЕ

Введение. Нейронные сети имеют различные периоды своего развития, связанные с оптимистичным и пессимистичным взглядом на эволюцию данной технологии в контексте развития систем искусственного интеллекта. Так, в 1958 г. Ф. Розенблатт предложил однослойный перцептрон [1], который затем был подвергнут критике из-за неспособности решать линейно неразделимые задачи [2]. Хотя, как будет показано в данной лекции, однослойный перцептрон с сигнальной функцией активации способен решить задачу «исключающее или». После этого наступил период спада в области перцептронных нейронных сетей и только, начиная с 1986 г. начался подъем, связанный с разработкой алгоритма обратного распространения ошибки для многослойных перцептронных сетей. При этом в научной среде вплоть до 2006 г. господствовала парадигма, что не имеет смысла использовать более двух скрытых слоев в перцептронной сети. Это было связано с тем, что алгоритм обратного распространения ошибки не давал никакого выигрыша в решении задач при обучении перцептронной сети с более чем двумя скрытыми слоями. Также данная парадигма базировалась на теореме об универсальной аппроксимации перцептронной сетью с одним или двумя скрытыми слоями любой функции сколь угодно заданной точностью. В результате использования перцептронных сетей с малым количеством скрытых слоев сложилось мнение, что его возможности в основном исчерпаны. Тем более, что появились машины опорных векторов (SVM), которые показывали часто большую эффективность на операциях распознавания образов по сравнению с перцептронной сетью. Поэтому интерес к многослойным перцептронным сетям стал постепенно спадать. И только, начиная с 2006 г., благодаря работам [3-6] Хинтона (J. Hinton), начался обратный отсчет в развитии многослойных перцептронных сетей под новым названием - нейронные сети глубокого доверия (deep belief neural networks). Такие сети в общем случае представляют собой дальнейшее развитие многослойных перцептронных сетей и интегрируют различные парадигмы обучения нейронных сетей. Такие сети, благодаря многослойной архитектуре позволяют обрабатывать и анализировать большой объем данных, а также моделировать когнитивные процессы в различных областях. В настоящее время большинство высокотехнологичных компаний в США (Microsoft, Google и т.д.) используют нейронные сети глубокого доверия для проектирования различных интеллектуальных систем. По версии ученых Массачусетского технологического института (США), нейронные сети глубокого доверия (deep belief neural networks), входят в список 10 наиболее прорывных высоких технологий, способных в недалеком будущем в значительной степени преобразить повседневную жизнь большинства людей на нашей планете. В настоящее время нейронные сети глубокого доверия считаются революционным шагом в области интеллектуальной обработки данных. В данной статье рассматриваются и анализируются основные парадигмы обучения перцептронных нейронных сетей: от однослойного перцептрона к многослойным сетям глубокого доверия. Показана несостоятельность некоторых мифов о возможностях перцептронных нейронных сетей и обосновывается переход к сверхбольшим перцептронным сетям глубокого доверия. Рассмотрены основные модели обучения нейронных сетей глубокого доверия, базирующиеся на ограниченной машине Больцмана (restricted Boltzmann machine (RBM)) и автоассоциативном подходе. Предложен новый метод для обучения ограниченной машины Больцмана и показано, что правило обучения ограниченной машины Больцмана является частным случаем предложенного метода обучения, который базируется на минимизации суммарной квадратичной ошибки восстановления информации. Доказана эквивалентность максимизации веро-

ятности распределения данных в ограниченной машине Больцмана и минимизация суммарной квадратичной ошибки восстановления информации в слоях (RBM). Рассматривается применение нейронных сетей глубокого доверия для решения различных задач.

1 Однослойный перцептрон. В данном разделе рассматривается однослойный перцептрон, его обучение и возможности. В настоящее время существует устойчивое мнение, что однослойный перцептрон не может решить задачу «исключающее или». Однако это справедливо только для перцептрона с пороговой или монотонно возрастающей функцией активации. При использовании сигнальной или унимодальной функции активации с одним максимумом однослойный перцептрон, как будет показано далее, способен решить проблему «исключающее или». Однослойный перцептрон принято изображать в виде двухслойной нейронной сети, где первый слой нейронных элементов является распределительным, а второй – обрабатывающим. Распределительный слой передает входные сигналы на обрабатывающий слой нейронных элементов, который преобразует входную информацию в соответствии с синаптическими связями и функцией активации (рисунок 1). При этом каждый нейрон распределительного слоя имеет синаптические связи со всеми нейронами обрабатывающего слоя.

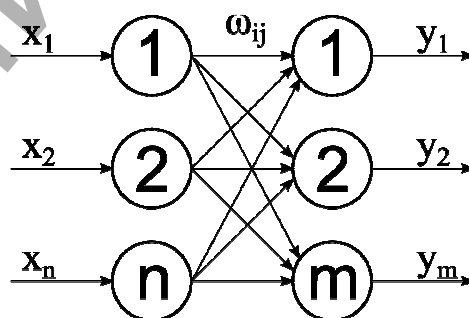


Рис. 1. Однослойный перцептрон

Выходное значение j -го нейронного элемента второго слоя можно представить как

$$y_j = F(S_j) = F\left(\sum_{i=1}^n w_{ij} x_i - T_j\right), \quad (1)$$

где T_j – порог j -го нейронного элемента выходного слоя; w_{ij} – сила синаптической связи между i -м нейроном распределительного слоя и j -м нейроном обрабатывающего слоя, F – оператор нелинейного преобразования или функция активации нейронных элементов.

Рассмотрим правило обучения однослойного перцептрона, который имеет линейную функцию активации. Такое правило обучения называется дельта правилом (delta rule) и может использоваться как для обучения линейного перцептрона, так и для перцептрона с пороговой функцией активации. Дельта правило было предложено в 60-х годах Ф. Розенблаттом [1] для перцептрона с пороговой функцией активации, а для линейной сети его предложили Видроу (Widrow) и Хофф (Hoff) [10]. Выходное значение линейного однослойного перцептрона определяется, как

Головко Владимир Адамович, д.т.н., профессор, зав. кафедрой интеллектуальных информационных технологий Брестского государственного технического университета.

Крощенко Александр Александрович, аспирант кафедры интеллектуальных информационных технологий Брестского государственного технического университета.

Беларусь, БрГТУ, 224017, г. Брест, ул. Московская, 267.

$$y_j = \sum_{i=1}^n w_{ij} x_i - T_j. \quad (2)$$

Дельта правило предполагает минимизацию суммарной среднеквадратичной ошибки нейронной сети, которая для L входных образов определяется следующим образом:

$$E_s = \sum_{k=1}^L E(k) = \frac{1}{2} \sum_{k=1}^L \sum_{j=1}^m (y_j^k - e_j^k)^2, \quad (3)$$

где $E(k)$ – среднеквадратичная ошибка сети для k -го образа; y_j^k и e_j^k – соответственно выходное и эталонное значение нейронной сети для k -го образа. Для минимизации суммарной квадратичной ошибки используется метод градиентного спуска. Существует два основных подхода к обучению однослойного персептрона: последовательное обучение (online learning) и групповое обучение (batch learning). При последовательном обучении модификация синаптических связей происходит после подачи каждого образа из обучающей выборки на нейронную сеть. В этом случае в методе градиентного спуска используется квадратичная ошибка нейронной сети для одного входного образа:

$$E = \frac{1}{2} \sum_{j=1}^m (y_j - e_j)^2. \quad (4)$$

Тогда в соответствии с методом градиентного спуска, весовые коэффициенты и пороги нейронной сети необходимо изменять с течением времени по следующим выражениям:

$$w_{ij}(t+1) = w_{ij}(t) - \alpha \frac{\partial E}{\partial w_{ij}(t)}; \quad (5)$$

$$T_j(t+1) = T_j(t) - \alpha \frac{\partial E}{\partial T_j(t)}, \quad (6)$$

где α – скорость или шаг обучения. Отсюда можно получить **дельта правило** для последовательного обучения:

$$w_{ij}(t+1) = w_{ij}(t) - \alpha (y_j - e_j) x_i; \quad (7)$$

$$T_j(t+1) = T_j(t) + \alpha (y_j - e_j). \quad (8)$$

Величина адаптивного шага при последовательном обучении [3]

$$\alpha(t) = \frac{1}{1 + \sum_i x_i^2(t)}.$$

При использовании группового обучения модификация синаптических связей будет происходить после подачи на вход сети L образов. В этом случае дельта правило можно представить следующим образом [11]:

$$w_{ij}(L) = w_{ij}(0) - \alpha(t) \sum_{k=1}^L (y_j^k - e_j^k) x_i^k; \quad (9)$$

$$T_j(L) = T_j(0) + \alpha(t) \sum_{k=1}^L (y_j^k - e_j^k).$$

Величина адаптивного шага при групповом обучении [11]

$$\alpha(t) = \frac{\sum_k \sum_j (y_j^k - e_j^k) a_j^k}{\sum_k \sum_j (a_j^k)^2}; \quad (10)$$

$$a_j^k = \sum_p (y_j^p - e_j^p) (1 + \sum_i x_i^p x_i^k).$$

Однослойный персептрон с пороговой функцией активации формирует линейную разделяющую поверхность с точки зрения классификации образов и поэтому не может решить задачу «исключающее или». Это в свое время показали американские ученые Минский и Пайперт [2], которые сделали пессимистические выводы насчет дальнейшего развития нейронных сетей. Однако последнее утверждение справедливо только для однослойного персептрона с

пороговой или монотонной непрерывной функцией активации (например, сигмоидная). При использовании функции активации Гаусса или сигмоидной функции однослойный персептрон может решить задачу «исключающее или». Покажем это для сигмоидной функции активации. В этом случае, необходимо выделить область единиц или нулей при помощи двух прямых (рисунок 2).

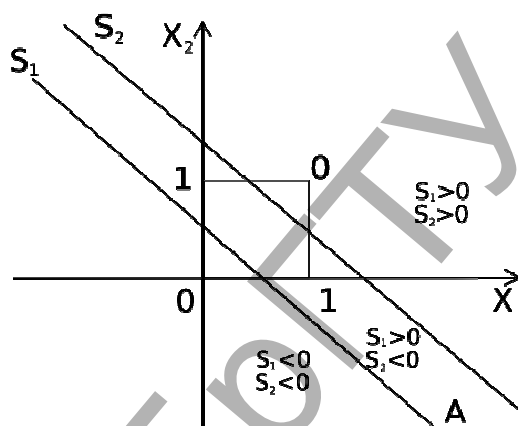


Рис. 2. Решение задачи «исключающее или»

Область А, которая характеризует класс единиц характеризуется следующим условием: $S_1 > 0$ и $S_2 < 0$. Уравнения прямых можно представить следующим образом:

$$S_1 = X_2 + X_1 - 0.5;$$

$$S_2 = X_2 + X_1 - 1.5.$$

Отсюда можно получить, что $S_2 = S_1 - 1$ (рисунок 3).

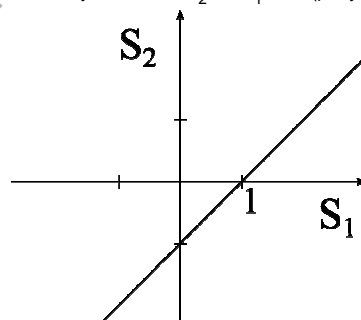


Рис. 3. Зависимость S_2 от S_1

Тогда, если $S_1 > 0$ и $S_2 < 0 \rightarrow 0 < S_1 < 1$ и можно получить следующую функцию активации (рисунок 4):

$$y = F(S) = F(S_1) = \begin{cases} 1, & 0 < S_1 < 1, \\ 0, & \text{иначе} \end{cases} \quad (11)$$

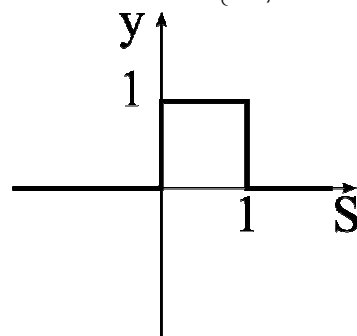


Рис. 4. Сигнальная функция активации

В результате однослойный персептрон для решения задачи «исключающее или» будет иметь следующие веса и порог (рисунок 5):

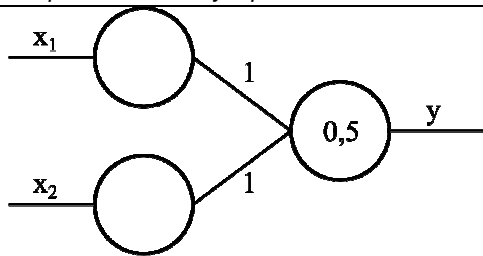


Рис. 5. Однослойный перцептрон для решения задачи «исключающее или»

Таким образом, при использовании сигнальной функции активации однослойный перцептрон может решить задачу «исключающее или» так как она разбивает входное пространство образов на классы при помощи двух прямых. При использовании однослойного перцептрона с радиально базисной функцией активации для решения данной задачи необходимо принимать выходное значение перцептрона равным единице, если оно больше определенного значения.

2 Многослойный перцептрон. В 1986 рядом авторов (Rumelhart, Hinton, Williams) независимо друг от друга был предложен алгоритм обратного распространения ошибки (backpropagation algorithm), который стал эффективным средством обучения многослойных нейронных сетей [12]. Перцептрон с одним скрытым слоем изображен на рисунке 6.

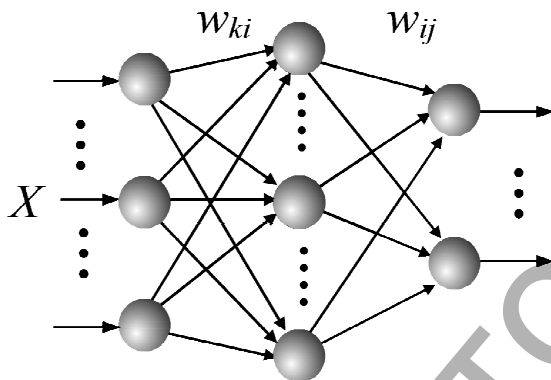


Рис. 6. Перцептрон с одним скрытым слоем

Выходное значение j -го нейрона последнего слоя определяется как

$$y_j = F(S_j),$$

$$S_j = \sum_i w_{ij} y_i - T_j.$$

Аналогичным образом для скрытого слоя

$$y_i = F(S_i),$$

$$S_i = \sum_k w_{ki} y_k - T_i,$$

где S_j – взвешенная сумма j -го нейрона, w_{ij} – весовой коэффициент от i -го к j -му нейрону, T_j – порог j -го нейрона последнего слоя.

Целью обучения является минимизация суммарной квадратичной ошибки сети:

$$E_s = \frac{1}{2} \sum_{k=1}^L \sum_j (y_j^k - e_j^k)^2 \quad (12)$$

где L – размерность обучающей выборки, y_j^k и e_j^k – соответственно выходное и эталонное значение нейронной сети для k -го образа. Для минимизации квадратичной ошибки сети весовые коэффициенты и пороги нейронных элементов должны изменяться с течением времени следующим образом:

$$w_{ij}(t+1) = w_{ij}(t) - \alpha \gamma_j F'(S_j) y_i,$$

$$T_j(t+1) = T_j(t) + \alpha \gamma_j F'(S_j),$$

где $\frac{\partial E}{\partial S_j} = F'(S_j)$, γ_j – ошибка j -го нейрона j -го слоя. Для выходного слоя

$$\gamma_j = y_j - e_j.$$

Для любого скрытого слоя i ошибка i -го нейронного элемента определяется рекурсивным образом через ошибки нейронов следующего слоя j :

$$\gamma_i = \sum_{j=1}^m \gamma_j F'(S_j) w_{ij}, \quad (13)$$

где m – количество нейронов следующего слоя по отношению к слою i ; w_{ij} – синаптическая связь между i -м и j -м нейроном различных слоев; S_j – взвешенная сумма j -го нейрона. Данное правило обучения многослойных нейронных сетей называется обобщенным дельта правилом. Для группового обучения, когда модификация синаптических связей происходит после подачи L образов на нейронную сеть:

$$w_{ij}(L) = w_{ij}(0) - \alpha(t) \sum_{k=1}^L \gamma_j^k F'(S_j^k) y_i^k,$$

$$T_j(L) = T_j(0) + \alpha(t) \sum_{k=1}^L \gamma_j^k F'(S_j^k).$$

Рассмотрим возможности многослойного перцептрона в зависимости от количества скрытых слоев. В литературе существует много ошибок по данному вопросу. Так, например, Липманн (R. Lippmann) утверждал в работе [13], что перцептрон с одним скрытым слоем может формировать только выпуклую разделяющую поверхность с точки зрения классификации образов. Позже было показано, что такой перцептрон может формировать произвольную разделяющую поверхность, однако данная ошибка попала и попадает во многие учебники. Прежде всего, следует отметить, что возможности перцептрона с одним скрытым слоем различаются в зависимости от используемой функции активации. В 1989 г. G. Cybenko [14], доказал теорему об универсальной аппроксимации:

Теорема 1. Любая функция может быть аппроксимирована перцептроном с одним скрытым слоем и сигмоидной функцией активации нейронных элементов с любой степенью точности при достаточном количестве нейронов в скрытом слое.

Данная теорема является основной для аппроксимации функций при помощи многослойных нейронных сетей. При этом точность аппроксимации зависит от количества нейронов в скрытом слое. Чем больше количество нейронных элементов в скрытом слое, тем больше точность. Однако при слишком большой размерности скрытого слоя может наступить явление, которое называется перетренировкой сети, когда сеть имеет плохую обобщающую способность. Таким образом, нейронная сеть с одним скрытым слоем позволяет осуществить любое отображение входных сигналов в выходные при использовании сигмоидной или любой ограниченной, монотонно возрастающей функции активации. Поэтому, такая сеть является также универсальным классификатором, то есть она способна формировать произвольную разделяющую поверхность в пространстве решений.

Рассмотрим теперь другой случай, когда в качестве функции активации нейронных элементов используется пороговая функция. Здесь ситуация меняется. Перцептрон с одним скрытым слоем и пороговой функцией активации нейронных элементов не способен формировать произвольную разделяющую поверхность и поэтому не является универсальным классификатором. В данном случае, только перцептрон с двумя скрытыми слоями способен формировать произвольную разделяющую поверхность. Покажем это на примере решения простой задачи бинарной классификации (рисунок 7).

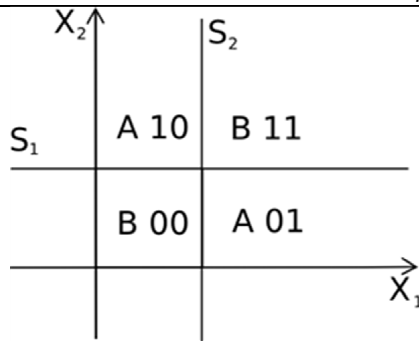


Рис. 7. Задача бинарной классификации двух классов А и В

Как видно из рисунка классы разделяются двумя прямыми S_1 и S_2 . Уравнения прямых будут следующие: $S_1 = X_2 - 1$ и $S_2 = X_1 - 1$. В результате, как следует из рисунка 7, данная задача сводится к проблеме «исключающее или». Поэтому персептрон для решения данной задачи будет состоять из двух скрытых слоев, где первый скрытый слой формирует разделяющие линии S_1 и S_2 (рисунок 8).

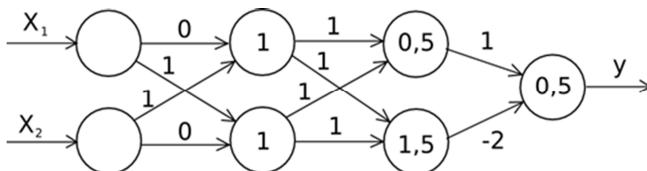


Рис. 8. Персептрон с двумя скрытыми слоями для решения задачи бинарной классификации

Рассмотрим еще один случай, когда используется сигнальная функция активации в отдельных слоях нейронных элементов. Тогда для решения приведенной выше задачи классификации достаточно персептрона с одним скрытым слоем и сигнальной функцией активации в выходном нейроне, как это показано на рисунке 9. Здесь нейроны скрытого слоя формируют дискриминантные линии S_1 и S_2 , а выходной нейронный элемент с сигнальной функцией активации формирует окончательное решение задачи.

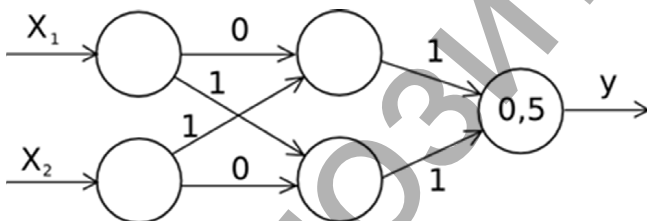


Рис. 9. Персептрон с одним скрытым слоем для решения задачи бинарной классификации

Таким образом, основная научная парадигма состояла в том, что персептрон с одним или максимум двумя скрытыми слоями является достаточным для решения различных задач. Использование персептрона с более чем двумя скрытыми слоями не имеет большого смысла.

3 Нейронные сети глубокого доверия. В общем случае нейронные сети глубокого доверия (deep belief neural networks) представляют собой персептрон с множеством слоев нейронных элементов [3–9]. До 2006 года в научной среде была приоритетной парадигма, что многослойный персептрон с одним, максимум двумя скрытыми слоями является более эффективным для нелинейного преобразования входного пространства образов в выходное по сравнению с персептроном с большим количеством скрытых слоев. Считалось, что персептрон с более чем двумя скрытыми слоями не имеет смысла применять. Данная парадигма базировалась на теореме, что персептрон с одним скрытым слоем является универсальным аппроксиматором. Другой аспект

этой проблемы заключается в том, что все попытки использовать алгоритм обратного распространения ошибки (backpropagation algorithm) для обучения персептрона с тремя и более скрытыми слоями не приводили к улучшению решения различных задач. Это связано с тем, что алгоритм обратного распространения ошибки является неэффективным для обучения персептронов с тремя и более слоями. В 2006 году Хинтон (Hinton) предложил «жадный» алгоритм послойного обучения (greedy layer-wise algorithm) [3], который стал эффективным средством обучения нейронных сетей глубокого доверия, которые как уже отмечалось, представляют собой персептрон с большим количеством слоев. Было показано, что нейронная сеть глубокого доверия имеет большую эффективность нелинейного преобразования и представления данных по сравнению с традиционным персептроном. Такая сеть осуществляет глубокое иерархическое преобразование входного пространства образов. В результате первый скрытый слой выделяет низкоуровневое пространство признаков входных данных, второй слой детектирует пространство признаков более высокого уровня абстракции и т.д. [5].

Как уже отмечалось, нейронная сеть глубокого доверия содержит множество скрытых слоев (рисунок 10) и осуществляет глубокое иерархическое преобразование входного пространства образов.

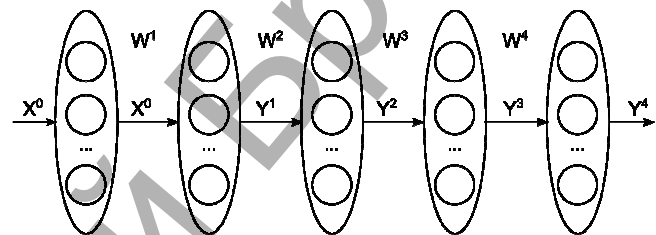


Рис. 10. Нейронная сеть глубокого доверия

Выходное значение j -го нейрона k -го слоя определяется следующим образом:

$$y_j^k = F(S_j^k), \quad (14)$$

$$S_j^k = \sum_{i=1} w_{ij}^k y_i^{k-1} + T_j^k, \quad (15)$$

где F – функция активации нейронного элемента, S_j^k – взвешенная сумма j -го нейрона k -слоя, w_{ij}^k – весовой коэффициент между i -ым нейроном ($k-1$ -го слоя) и j -м нейроном k -го слоя, T_j^k – пороговое значение j -го нейрона k -го слоя.

Для первого (распределительного) слоя

$$y_i^0 = x_i. \quad (16)$$

В матричном виде выходной вектор k -го слоя

$$Y^k = F(S^k) = F(W^k Y^{k-1} + T^k), \quad (17)$$

где W – матрица весовых коэффициентов, Y^{k-1} – выходной вектор ($k-1$ -го слоя), T^k – вектор пороговых значений нейронов k -го слоя. Если нейронная сеть глубокого доверия используется для классификации образов, то выходные значения сети часто определяются на основе функции активации **softmax**:

$$y_j^F = \text{softmax}(S_j) = \frac{e^{S_j}}{\sum_i e^{S_i}}. \quad (18)$$

Процесс обучения нейронных сетей глубокого доверия в общем случае состоит из двух этапов: предобучение нейронной сети методом послойного обучения, начиная с первого слоя (pre-training). Данное обучение осуществляется без учителя. Настройка синтаптических связей всей сети (fine-tuning) при помощи алгоритма обратного распространения ошибки или алгоритма «бодрствования и сна» (wake-sleep algorithm).

Важным этапом обучения нейронных сетей глубокого доверия является предобучение слоев нейронной сети. Существует два основных подхода к предварительному обучению слоев нейронных сетей глубокого доверия (рисунок 11).

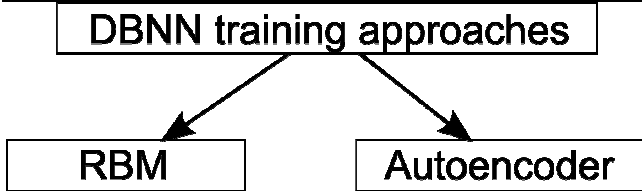


Рис. 11. Методы предварительного обучения сетей глубокого доверия

Первый подход называется автоэнкодерным и базируется на представлении каждого слоя в виде автоассоциативной нейронной сети (рециркуляционная или PCA нейронная сеть). Второй подход базируется на представлении каждого слоя нейронной сети в виде ограниченной машины Больцмана (RBM).

3.1 Автоэнкодерный метод обучения. В этом случае, вначале обучается первый слой как автоассоциативная нейронная сеть с целью минимизации суммарной ошибки реконструкции информации, затем второй и так далее. Для обучения каждого слоя можно использовать алгоритм обратного распространения ошибки. После этого осуществляется точная настройка синаптических связей всей сети (fine tuning), используя алгоритм обратного распространения ошибки. Рассмотрим персептрон с тремя скрытыми слоями (рисунок 12). Тогда в соответствии с автоэнкодерным методом, прежде всего, берутся первые два слоя нейронной сети (1 и 2) и на базе их конструируется автоассоциативная (рециркуляционная) нейронная сеть (1-2-1), то есть добавляется восстанавливающий слой (рисунок 13). Затем, происходит обучение, например, при помощи алгоритма обратного распространения ошибки такой сети с целью минимизации ошибки реконструкции информации. Продолжительность обучения обычно составляет не больше, чем 100 эпох.

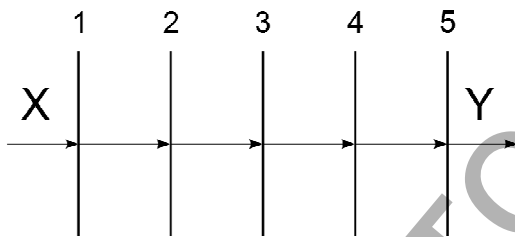


Рис. 12. Персептрон с тремя скрытыми слоями

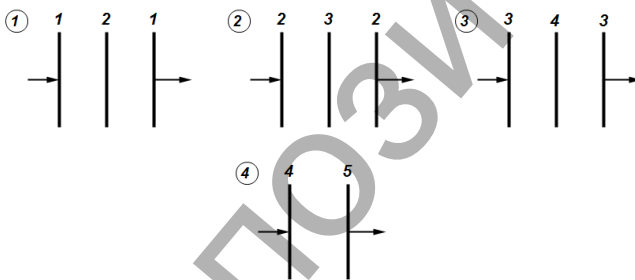


Рис. 13. Автоэнкодерный метод обучения

После этого отбрасывается восстанавливающий слой (последний слой), фиксируются веса скрытого слоя, и конструируется автоассоциативная сеть из следующих двух слоев нейронной сети (2-3-2), которая обучается на основе данных поступающих с предыдущего (2-го слоя). Процесс продолжается до последнего или предпоследнего слоя, как это схематично изображено на рисунке 13. В результате послойного обучения получается предварительно обученная нейронная сеть. Далее осуществляется точная настройка (fine tuning) посредством, например, алгоритма обратного распространения ошибки с учителем. Данный процесс можно представить в виде следующего алгоритма:

1. Конструируется автоассоциативная сеть с входным слоем X , скрытым Y и выходным слоем X .

2. Обучается автоассоциативная сеть, например при помощи алгоритма обратного распространения ошибки (как правило не более 100 эпох) и фиксируются синаптические связи первого слоя W_1 .
3. Берется следующий слой и формируется автоассоциативная сеть аналогичным образом.
4. Используя настроенные синаптические связи предыдущего слоя W_1 , подаем входные данные на вторую автоассоциативную сеть и обучаем ее аналогичным образом. В результате получаются весовые коэффициенты второго слоя W_2 .
5. Процесс продолжается до последнего слоя нейронной сети.
6. Берется последний слой нейронной сети и обучается с учителем.
7. Обучается вся сеть для точной настройки параметров при помощи алгоритма обратного распространения ошибки.

3.2 Ограниченная машина Больцмана. Как уже отмечалось, данный подход базируется на представлении каждого слоя нейронной сети в виде ограниченной машины Больцмана (RBM). Ограниченная машина Больцмана состоит из двух слоев стохастических бинарных нейронных элементов, которые соединены между собой двунаправленными симметричными связями (рисунок 14). Входной слой нейронных элементов называется видимым (слой X), а второй слой называется скрытым (слой Y). Нейронную сеть глубокого доверия можно представить как совокупность ограниченных машин Больцмана. Ограниченная машина Больцмана может генерировать (представить) любое дискретное распределение, если используется достаточное количество нейронов скрытого слоя [7].

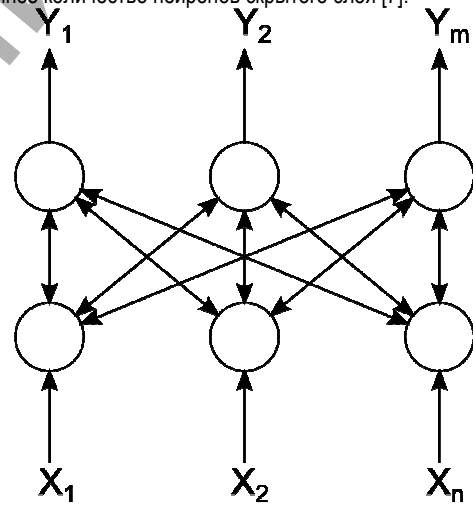


Рис. 14. Ограниченная машина Больцмана

Данная сеть является стохастической нейронной сетью, в которой состояния видимых и скрытых нейронов меняются в соответствии с вероятностной версией сигмоидной функции активации:

$$p(y_j | x) = \frac{1}{1 + e^{-S_j}}, S_j = \sum_i w_{ij} x_i + T_j, \quad (19)$$

$$p(x_i | y) = \frac{1}{1 + e^{-S_i}}, S_i = \sum_j w_{ij} y_j + T_i. \quad (20)$$

Состояния видимых и скрытых нейронных элементов принимаются независимыми:

$$P(x | y) = \prod_{i=1}^n P(x_i | y);$$

$$P(y | x) = \prod_{j=1}^m P(y_j | x).$$

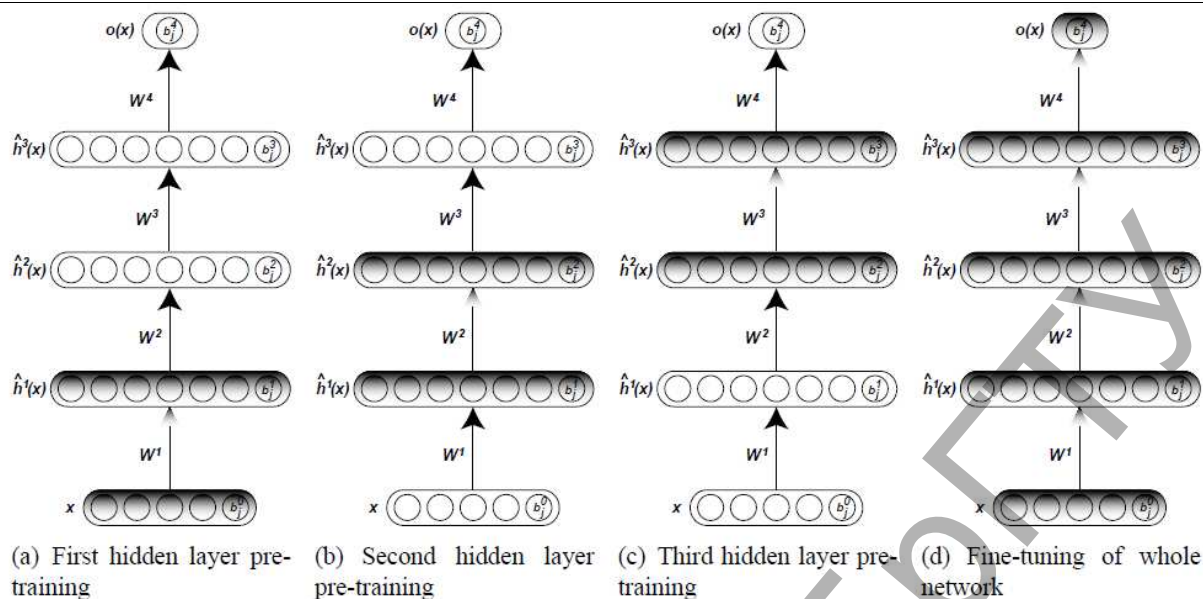


Рис. 15. Greedy layer-wise algorithm [Journal of Machine Learning Research 1 (2009) 1-40]

Таким образом, состояния всех нейронных элементов ограниченной машины Больцмана определяются через распределение вероятностей. В RBM нейроны скрытого слоя являются детекторами признаков, которые сохраняют закономерности входных данных. Основная задача обучения состоит в воспроизведении распределения входных данных на основе состояний нейронов скрытого слоя как можно точнее. Это эквивалентно максимизации функции правдоподобия путем модификации синаптических связей нейронной сети. Рассмотрим это подробнее. Вероятность нахождения видимого и скрытого нейрона в состоянии (x, y) определяется на основе распределения Гиббса:

$$P(x, y) = \frac{e^{-E(x, y)}}{Z},$$

где $E(x, y)$ – энергия системы в состоянии (x, y) , Z – параметр, который определяет условие нормализации вероятностей, то есть, чтобы сумма вероятностей равнялась единице. Данный параметр определяется следующим образом:

$$Z = \sum_{x, y} e^{-E(x, y)}.$$

Вероятность нахождения видимых нейронов в определенном состоянии равняется сумме вероятностей конфигураций $P(x, y)$ по состояниям скрытых нейронов:

$$P(x) = \sum_y P(x, y) = \sum_y \frac{e^{-E(x, y)}}{Z} = \frac{\sum_y e^{-E(x, y)}}{\sum_{x, y} e^{-E(x, y)}}.$$

Для нахождения правила модификации синаптических связей необходимо максимизировать вероятность воспроизведения состояний видимых нейронов $P(x)$ ограниченной машиной Больцмана. Для того, чтобы определить максимум функции правдоподобия распределения данных $P(x)$ будем использовать метод градиентного спуска в пространстве весовых коэффициентов и пороговых значений сети, где в качестве градиента применим функцию логарифмического правдоподобия:

$$\ln P(x) = \ln \sum_y e^{-E(x, y)} - \ln \sum_{x, y} e^{-E(x, y)}.$$

Тогда градиент равен

$$\frac{\partial \ln P(x)}{\partial w_{ij}} = \frac{\partial}{\partial w_{ij}} \ln \sum_y e^{-E(x, y)} - \frac{\partial}{\partial w_{ij}} \ln \sum_{x, y} e^{-E(x, y)}.$$

Преобразуя последнее выражение, получим

$$\frac{\partial \ln P(x)}{\partial w_{ij}} = - \frac{1}{\sum_y e^{-E(x, y)}} \sum_y e^{-E(x, y)} \frac{\partial E(x, y)}{\partial w_{ij}} + \frac{1}{\sum_{x, y} e^{-E(x, y)}} \sum_{x, y} e^{-E(x, y)} \frac{\partial E(x, y)}{\partial w_{ij}}.$$

Так как

$$P(x, y) = P(y | x)P(x)$$

то

$$P(y | x) = \frac{P(x, y)}{P(x)} = \frac{(1/Z)e^{-E(x, y)}}{(1/Z)\sum_y e^{-E(x, y)}} = \frac{e^{-E(x, y)}}{\sum_y e^{-E(x, y)}}.$$

В результате можно получить следующее выражение:

$$\frac{\partial \ln P(x)}{\partial w_{ij}} = - \sum_y P(y | x) \frac{\partial E(x, y)}{\partial w_{ij}} + \sum_{x, y} P(x, y) \frac{\partial E(x, y)}{\partial w_{ij}}.$$

В данном выражении первое слагаемое определяет позитивную фазу работы машины Больцмана, когда сеть работает на основе образов из обучающей выборки. Второе слагаемое характеризует негативную фазу функционирования, когда сеть работает в свободном режиме независимо от окружающей среды. Рассмотрим энергию сети RBM. С точки зрения энергии сети задача обучения состоит в том, чтобы на основе входных данных найти конфигурацию выходных переменных с минимальной энергией. В результате на обучающем множестве сеть будет иметь меньшую энергию по сравнению с другими состояниями. Функция энергии бинарного состояния (x, y) определяется аналогично сети Хопфилда:

$$E(x, y) = - \sum_i x_i T_i - \sum_j y_j T_j - \sum_{i, j} x_i y_j w_{ij}. \quad (21)$$

В этом случае

$$\frac{\partial E(x, y)}{\partial w_{ij}} = -x_i y_j$$

и

$$\frac{\partial \ln P(x)}{\partial w_{ij}} = \sum_y P(y | x) x_i y_j - \sum_{x, y} P(x, y) x_i y_j.$$

Так как математическое ожидание равняется:

$$E(x) = \sum_i x_i P_i$$

то

$$\frac{\partial \ln P(x)}{\partial w_{ij}} = E[x_i y_j]_{data} - E[x_i y_j]_{model}.$$

Аналогичным образом можно получить градиенты для пороговых значений:

$$\frac{\partial \ln P(x)}{\partial T_i} = E[x_i]_{data} - E[x_i]_{model};$$

$$\frac{\partial \ln P(x)}{\partial T_j} = E[y_j]_{data} - E[y_j]_{model}.$$

Как следует из последних выражений, первое слагаемое характеризует работу сети на основе данных из обучающей выборки, а второе слагаемое – на основе данных модели (данные генерируемые сетью), то есть в свободном режиме независимо от окружающей среды. Так как вычисление математического ожидания на основе RBM сети является очень сложным, Хинтон предложил использовать аппроксимацию данных слагаемых, которую он назвал контрастным расхождением (contrastive divergence (CD)) [3]. Такая аппроксимация основывается на дискретизаторе Гиббса (Gibbs sampling). В этом случае первые слагаемые в выражениях для градиента характеризуют распределение данных в момент времени $t = 0$, а вторые слагаемые характеризуют реконструированные или генерируемые моделью состояния в момент времени $t = k$. Исходя из этого, CD-k процедура может быть представлена следующим образом:

$$x(0) \rightarrow y(0) \rightarrow x(1) \rightarrow y(1) \rightarrow \dots \rightarrow x(k) \rightarrow y(k). \quad (22)$$

В результате можно получить следующие правила для обучения RBM сети. В случае применения CD-1, $k = 1$ и учитывая, что в соответствии с методом градиентного спуска

$$w_{ij}(t+1) = w_{ij}(t) + \alpha \frac{\partial \ln P(x)}{\partial w_{ij}(t)}.$$

Можно получить, что

$$\begin{aligned} w_{ij}(t+1) &= w_{ij}(t) + \alpha(x_i(0)y_j(0) - x_i(1)y_j(1)); \\ T_i(t+1) &= T_i(t) + \alpha(x_i(0) - x_i(1)); \\ T_j(t+1) &= T_j(t) + \alpha(y_j(0) - y_j(1)). \end{aligned}$$

Аналогичным образом, для алгоритма CD-k

$$\begin{aligned} w_{ij}(t+1) &= w_{ij}(t) + \alpha(x_i(0)y_j(0) - x_i(k)y_j(k)); \\ T_i(t+1) &= T_i(t) + \alpha(x_i(0) - x_i(k)); \\ T_j(t+1) &= T_j(t) + \alpha(y_j(0) - y_j(k)). \end{aligned}$$

Из последних выражений видно, что правила обучения ограниченной машины Больцмана минимизируют разницу между оригинальными данными и данными генерируемыми моделью. Генерируемые моделью данные получаются при помощи сэмплинга Гиббса.

3.3 Обучение нейронной сети глубокого доверия на основе RBM. Обучение нейронной сети глубокого доверия происходит на основе «жадного» алгоритма послойного обучения (greedy layer-wise algorithm). В соответствии с ним вначале обучается первый слой сети как RBM машина. Для этого входные данные поступают на видимый слой нейронных элементов и используя CD-k процедуру вычисляются состояния скрытых $p(y|x)$ и видимых нейронов $p(x|y)$. В процессе выполнения данной процедуры (не более 100 эпох) изменяются весовые коэффициенты и пороговые значения RBM сети, которые затем фиксируются. Затем берется второй слой нейронной сети и конструируется RBM машина. Входными данными для нее являются данные с предыдущего слоя. Происходит обучение и процесс продолжается для всех слоев нейронной сети, как показано на рисунке 15 [15]. В результате такого обучения без учителя можно получить подходящую начальную инициализацию настраиваемых параметров сети глубокого доверия. На заключительном этапе осуществляется точная настройка параметров всей

сети при помощи алгоритма обратного распространения ошибки или алгоритма «бодрствования и сна» (wake-sleep algorithm).

4 Альтернативный взгляд на ограниченную машину Больцмана. В данном разделе рассматривается альтернативный взгляд на ограниченную машину Больцмана как автоассоциативную нейронную сеть, которая может функционировать с любыми данными, как бинарными, так и числовыми. Предлагается новый метод для получения правила обучения ограниченной машины Больцмана [16]. Он базируется на минимизации ошибки реконструкции видимых и скрытых образов, которую можно получить, используя итерации сэмплинга Гиббса. По сравнению с традиционным подходом, основанным на энергии методе (energy-based method), который базируется на линейном представлении нейронных элементов, предложенный метод позволяет учитывать нелинейную природу нейронных элементов. Рассмотрим ограниченную машину Больцмана, которую будем представлять в виде трех слоев нейронных элементов [17]: видимый, скрытый и видимый (рисунок 16).

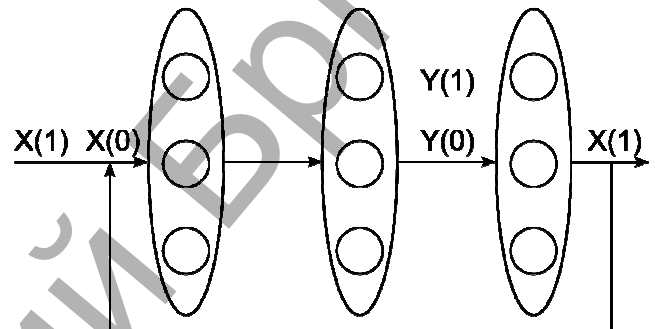


Рис. 16. Представление RBM в виде рециркуляционной сети

Сэмплирование Гиббса заключается в следующей процедуре. Пусть $x(0)$ входной вектор, который поступает на видимый слой в момент времени 0. Тогда выходные значения нейронов скрытого слоя:

$$y_j(0) = F(S_j(0)), \quad (23)$$

$$S_j(0) = \sum_i w_{ij} x_i(0) + T_j. \quad (24)$$

Инверсный (последний) слой реконструирует входной вектор на основе данных со скрытого слоя. В результате получается восстановленный вектор $x(1)$ в момент времени 1:

$$x_i(1) = F(S_i(1)), \quad (25)$$

$$S_i(1) = \sum_j w_{ij} y_j(0) + T_i. \quad (26)$$

Затем вектор $x(1)$ поступает на видимый слой, и вычисляются выходные значения нейронов скрытого слоя:

$$y_j(1) = F(S_j(1)), \quad (27)$$

$$S_j(1) = \sum_i w_{ij} x_i(1) + T_j. \quad (28)$$

Продолжая данный процесс, можно получить на шаге k

$$y_i(k) = F(S_i(k)), \quad S_i(k) = \sum_j w_{ij} x_j(k) + T_i.$$

$$x_i(k) = F(S_i(k)), \quad S_i(k) = \sum_j w_{ij} y_j(k-1) + T_i.$$

Целью обучения ограниченной машины Больцмана является минимизация суммарной квадратичной ошибки реконструкции данных на скрытом и восстанавливающем слое, которая в случае CD-k определяется следующим образом:

$$E_s = \frac{1}{2} \sum_{l=1}^L \sum_{j=1}^m \sum_{p=1}^k (y'_j(p) - y'_j(p-1))^2 +$$

$$+ \frac{1}{2} \sum_{l=1}^L \sum_{i=1}^n \sum_{p=1}^k (x'_i(p) - x'_i(p-1))^2.$$

В случае CD-1 суммарная квадратичная ошибка

$$E_s = \frac{1}{2} \sum_{l=1}^L \sum_{j=1}^m (y'_j(1) - y'_j(0))^2 + \frac{1}{2} \sum_{l=1}^L \sum_{i=1}^n (x'_i(1) - x'_i(0))^2, \quad (29)$$

где L количество входных образов. Как следует из приведенных выше выражений ошибка состоит из двух частей: ошибки восстановления информации на скрытом и выходном слое.

Теорема 2. Максимизация функции правдоподобия распределения данных $P(x)$ в пространстве синаптических связей ограниченной машины Больцмана эквивалентна минимизации суммарной квадратичной ошибки сети в том же пространстве при использовании линейных нейронов.

Доказательство: Рассмотрим последовательное обучение RBM, когда модификация синаптических связей происходит после подачи каждого входного образа на сеть. В соответствии с методом градиентного спуска для минимизации суммарной квадратичной ошибки сети, синаптические связи должны изменяться следующим образом:

$$w_{ij}(t+1) = w_{ij}(t) - \alpha \frac{\partial E}{\partial w_{ij}(t)}, \quad (30)$$

$$T_i(t+1) = T_i(t) - \alpha \frac{\partial E}{\partial T_i(t)}, \quad (31)$$

$$T_j(t+1) = T_j(t) - \alpha \frac{\partial E}{\partial T_j(t)}. \quad (32)$$

В случае CD-k квадратичная ошибка E для одного образа:

$$E = \frac{1}{2} \sum_{j=1}^m \sum_{p=1}^k (y_j(p) - y_j(p-1))^2 + \frac{1}{2} \sum_{i=1}^n \sum_{p=1}^k (x_i(p) - x_i(p-1))^2.$$

Тогда

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial y_j(p)} \frac{\partial y_j(p)}{\partial S_j(p)} \frac{\partial S_j(p)}{\partial w_{ij}} + \frac{\partial E}{\partial x_i(p)} \frac{\partial x_i(p)}{\partial S_i(p)} \frac{\partial S_i(p)}{\partial w_{ij}} = \sum_{p=1}^k (y_j(p) - y_j(p-1)) x_i(p) F'(S_j(p)) + \sum_{p=1}^k (x_i(p) - x_i(p-1)) y_j(p-1) F'(S_i(p)).$$

Если ограниченная машина Больцмана использует линейные нейроны с линейной функцией активации, то

$$F'(S_j(p)) = \frac{\partial S_j(p)}{\partial w_{ij}} = F'(S_i(p)) = \frac{\partial S_i(p)}{\partial w_{ij}} = 1,$$

Тогда

$$\frac{\partial E}{\partial w_{ij}} = \sum_{p=1}^k (y_j(p) x_i(p) - y_j(p-1) x_i(p-1)) = y_j(k) x_i(k) - y_j(0) x_i(0).$$

В результате можно получить CD-k правило обучения RBM:

$$w_{ij}(t+1) = w_{ij}(t) + \alpha (x_i(0) y_j(0) - x_i(k) y_j(k)).$$

Аналогичным образом для пороговых значений:

$$T_j(t+1) = T_j(t) + \alpha (y_j(0) - y_j(k)),$$

$$T_i(t+1) = T_i(t) + \alpha (x_i(0) - x_i(k)).$$

Как видно последние выражения совпадают с классическим правилом обучения ограниченной машины Больцмана для CD-k. Отсюда следует, что для линейной RBM максимизация функции правдоподобия распределения данных $P(x)$ эквивалентна минимизации суммарной квадратичной ошибки сети. Теорема доказана.

Следствие 1. Линейная ограниченная машина Больцмана с точки зрения обучения эквивалентна автоассоциативной нейронной сети при использовании в ней при обучении сэмплирования Гиббса.

Следствие 2. Для нелинейной ограниченной машины Больцмана правило модификации синаптических связей в случае CD-k будет следующим:

$$w_{ij}(t+1) = w_{ij}(t) - \alpha \left(\sum_{p=1}^k (y_j(p) - y_j(p-1)) x_i(p) F'(S_j(p)) + (x_i(p) - x_i(p-1)) y_j(p-1) F'(S_i(p)) \right),$$

$$T_j(t+1) = T_j(t) - \alpha \left(\sum_{p=1}^k (y_j(p) - y_j(p-1)) F'(S_j(p)) \right),$$

$$T_i(t+1) = T_i(t) - \alpha \left(\sum_{p=1}^k (x_i(p) - x_i(p-1)) F'(S_i(p)) \right).$$

Следствие 3. Для нелинейной ограниченной машины Больцмана правило модификации синаптических связей в случае CD-1 будет следующим:

$$w_{ij}(t+1) = w_{ij}(t) - \alpha \left((y_j(1) - y_j(0)) F'(S_j(1)) x_i(1) + (x_i(1) - x_i(0)) F'(S_i(1)) y_j(0) \right),$$

$$T_i(t+1) = T_i(t) - \alpha (x_i(1) - x_i(0)) F'(S_i(1)),$$

$$T_j(t+1) = T_j(t) - \alpha (y_j(1) - y_j(0)) F'(S_j(1)).$$

Если используется групповое обучение (batch learning), то в этом случае метод градиентного спуска записывается следующим образом:

$$w_{ij}(t+1) = w_{ij}(t) - \alpha \frac{\partial E_s}{\partial w_{ij}(t)}, \quad (33)$$

$$T_i(t+1) = T_i(t) - \alpha \frac{\partial E_s}{\partial T_i(t)}, \quad (34)$$

$$T_j(t+1) = T_j(t) - \alpha \frac{\partial E_s}{\partial T_j(t)}. \quad (35)$$

Теорема 3. При использовании CD-k для нелинейной ограниченной машины Больцмана в случае группового обучения правило модификации синаптических связей определяется на основе следующих выражений:

$$w_{ij}(t+1) = w_{ij}(t) - \alpha \left(\sum_{l=1}^L \sum_{p=1}^k (y'_j(p) - y'_j(p-1)) x'_i(p) F'(S'_j(p)) + (x'_i(p) - x'_i(p-1)) y'_j(p-1) F'(S'_i(p)) \right),$$

$$T_j(t+1) = T_j(t) - \alpha \left(\sum_{l=1}^L \sum_{p=1}^k (y'_j(p) - y'_j(p-1)) F'(S'_j(p)) \right),$$

$$T_i(t+1) = T_i(t) - \alpha \left(\sum_{l=1}^L \sum_{p=1}^k (x'_i(p) - x'_i(p-1)) F'(S'_i(p)) \right).$$

Процесс доказательства данной теоремы является аналогичным доказательству теоремы 2.

Следствие 4. При использовании CD-1 для нелинейной ограниченной машины Больцмана в случае группового обучения правило модификации синаптических связей определяется на основе следующих выражений:

$$w_{ij}(t+1) = w_{ij}(t) - \alpha \left(\sum_{l=1}^L (y'_j(1) - y'_j(0)) x'_i(1) F'(S'_j(1)) + (x'_i(1) - x'_i(0)) y'_j(0) F'(S'_i(1)) \right),$$

$$T_j(t+1) = T_j(t) - \alpha \left(\sum_{l=1}^L (y'_j(1) - y'_j(0)) F'(S'_j(1)) \right),$$

$$T_i(t+1) = T_i(t) - \alpha \left(\sum_{l=1}^L (x'_i(1) - x'_i(0)) F'(S'_i(1)) \right).$$

Следствие 5. При использовании CD-k для линейной ограниченной машины Больцмана в случае группового обучения правило модификации синаптических связей определяется на основе следующих выражений:

$$w_{ij}(t+1) = w_{ij}(t) + \alpha \sum_{l=1}^L (x'_i(0)y'_j(0) - x'_i(k)y'_j(k)),$$

$$T_j(t+1) = T_j(t) + \alpha \sum_{l=1}^L (y'_j(0) - y'_j(k)),$$

$$T_i(t+1) = T_i(t) + \alpha \sum_{l=1}^L (x'_i(0) - x'_i(k)).$$

Следствие 6. При использовании CD-1 для линейной ограниченной машины Больцмана в случае группового обучения правило модификации синаптических связей определяется на основе следующих выражений:

$$w_{ij}(t+1) = w_{ij}(t) + \alpha \sum_{l=1}^L (x'_i(0)y'_j(0) - x'_i(1)y'_j(1)),$$

$$T_j(t+1) = T_j(t) + \alpha \sum_{l=1}^L (y'_j(0) - y'_j(1)),$$

$$T_i(t+1) = T_i(t) + \alpha \sum_{l=1}^L (x'_i(0) - x'_i(1)).$$

В данном разделе получены правила обучения для ограниченной машины Больцмана, которые базируются на минимизации квадратичной ошибки восстановления информации в скрытом и видимом слоях. Предложенный метод позволяет учитывать нелинейную природу нейронных элементов. Показано, что классические выражения для обучения ограниченной машины являются частным случаем предложенного метода. Доказана теорема об эквивалентности максимизации функции правдоподобия распределения данных $P(x)$ в пространстве синаптических связей и минимизации суммарной квадратичной ошибки сети в том же пространстве для линейной ограниченной машины Больцмана.

5 Применение нейронных сетей глубокого доверия. Нейронные сети глубокого доверия применяются для сжатия и визуализации данных (автоэнкодерные сети), распознавания образов, обработки речи и т.д. Рассмотрим вначале применение автоэнкодерных сетей глубокого доверия для сжатия данных и сравнительного анализа различных алгоритмов.

5.1 Сжатие данных. Рассмотрим систему трех динамических уравнений [18], где параметр времени t генерируется в диапазоне [-1,1]:

$$\begin{cases} x_1 = \sin(\pi t) + \mu \\ x_2 = \cos(\pi t) + \mu \\ x_3 = t + \mu \end{cases}$$

Здесь μ – гауссовский шум с нулевым средним и квадратичным отклонением равным 0.05. Нелинейный автоэнкодер, который состоит из семи слоев нейронных элементов, изображен на рисунке 17. Для обучения использовалась выборка, состоящая из 1000 тренировочных наборов. Тестирование сети проводилось на данных, не входящих в обучающую выборку, количество которых составило 1000 образов. В качестве функции активации для всех слоев кроме сжимающего использовалась сигмоидная функция. Для сжимающего нейрона использовалась линейная функция активации. Для обучения каждого слоя нейронной сети использовалось 50 эпох, а для точной настройки параметров сети при помощи алгоритма обратного распространения

ошибки использовалось 200 эпох. Результаты экспериментов приведены в таблице 1. Здесь MSE – суммарная квадратичная ошибка на тренировочном наборе, MS – суммарная квадратичная ошибка на тестовом наборе, RBM – метод обучения на основе ограниченной машины Больцмана, REBA – предложенный метод.

Таблица 1. Сравнение RBM и REBA методов

Подход	MSE	MS
RBM	1,367	1,291
REBA	1,189	1,119

Как следует из таблицы, предложенный подход REBA показывает лучшую обобщающую способность.

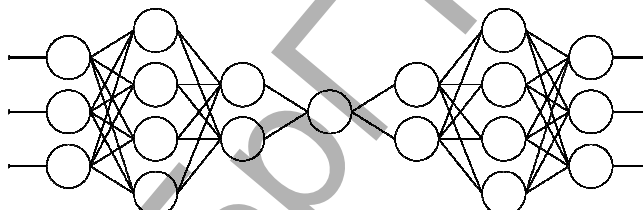


Рис. 17. Автоассоциативная нейронная сеть глубокого доверия

На рисунках 18 и 19 изображена нелинейная ось первой главной компоненты, на которую проецируется входное пространство образов.

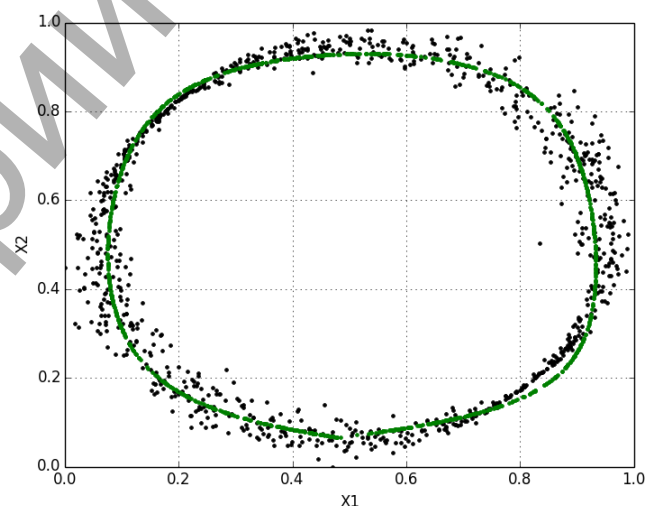


Рис. 18. Нелинейная ось первой главной компоненты в двумерном пространстве

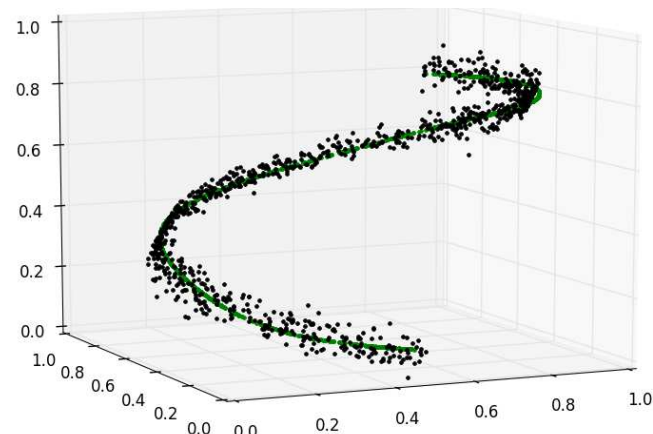


Рис. 19. Нелинейная ось первой главной компоненты в трехмерном пространстве

5.2 Визуализация данных

Autoencoder 2-D Topic Space

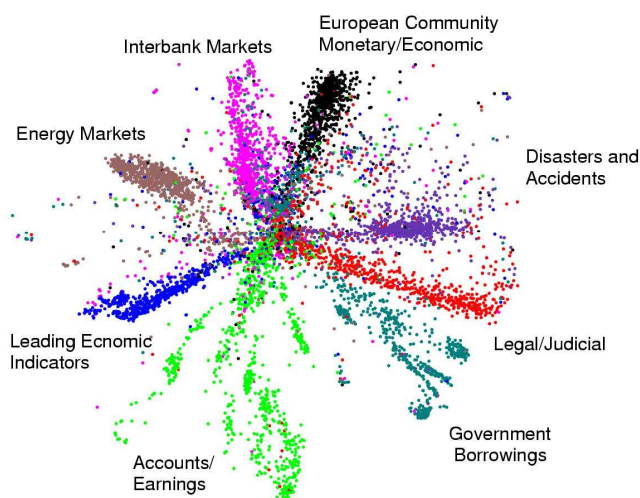


Рис. 20. Визуализация документов

На рисунке 20 изображен пример визуализации документов при помощи сети 2000-500-250-2-250-500-2000. Обучающая выборка состояла из 400000 документов [5]. Как видно из рисунка нейронная сеть глубокого доверия довольно точно осуществляет разделение документов на различные классы.

5.3 Распознавание образов. Предложенный альтернативный

подход был использован при обучении глубокой нейронной сети для распознавания рукописных цифр базы MNIST. Для этого в качестве базовой модели использовалась нейронная сеть с архитектурой 784-300-100-10. Для обучения использовалась выборка образов из 1.000 первых элементов. На этапе тестирования также использовалась выборка из 1.000 первых элементов. Основные параметры обучения: групповое обучение по 10 элементов в каждой группе, скорость обучения для этапа преднастройки (pretraining) весов – 0.05 (REBA – 0.05), для этапа финальной настройки (fine-tuning) – 0.7, ограничения на количество эпох обучения: pretraining – 6/10 эпох, fine-tuning – 50 эпох. В процессе выполнения экспериментов было установлено, что предложенный метод достигает меньшего значения ошибки обобщения при разумном ограничении количества эпох предобучения на первом слое НС. В таблице 2 отображены результаты, полученные для одинакового количества эпох предобучения для первого и второго слоев сети, а в таблице 3 – для разных.

Таблица 2. REBA с одинаковым числом эпох обучения (% ошибочно распознанных изображений)

Метод	Обучение	Тестирование
Classic	1,62	12,56
REBA	1,44	13,14

Таблица 3. REBA с разным числом эпох обучения (% ошибочно распознанных изображений)

Метод	Обучение	Тестирование
Classic	1,71	12,77
REBA	1,33	12,38

Визуализация весов первого слоя нейронной сети, полученная после этапа обучения, представлена на рис. 21.

Помимо этих экспериментов, проводились исследования предложенного метода на простейшей задаче распознавания цифр [W. Ertel - Grundkurs Künstliche Intelligenz: Eine praxisorientierte Einführung]. Оригинальный и тестовый наборы изображений представлены на рисунке 22. Задача решалась на обучающей выборке из 10 элементов и тестировалась на тестовом наборе из 100 элементов с 10% искажениями. Архитектура сети: 100-75-50-25-10 с сигмоидными функциями активации. Основные параметры: CD-1, скорость обучения: 0.05 (0.9 для REBA), ограничение количество эпох предобучения: 50.

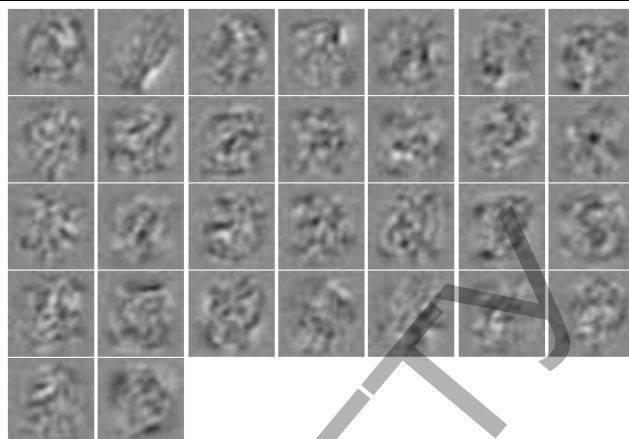


Рис. 21. Визуализация весовых коэффициентов первого слоя

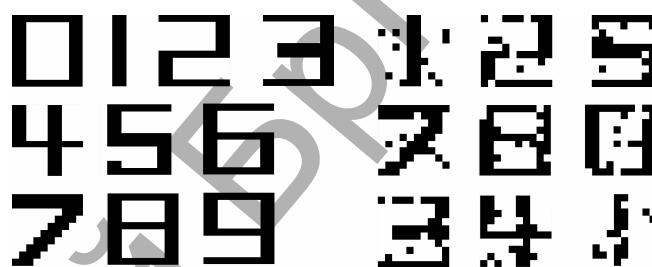


Рис. 22. Оригинальный набор и фрагмент тестового набора изображений

Сравнительные результаты, полученные для этой задачи, представлены в таблице 4.

Таблица 4. Сравнение методов (в % верно распознанных изображений)

Подход	Ограничение количества эпох обучения		
	100	200	300
RBM	74	84	88
REBA	76	85	87
BP	50	75	86

Заключение. В данной статье рассматриваются и анализируются основные парадигмы обучения перцептронных нейронных сетей: от однослойного перцептрона к многослойным сетям глубокого доверия. Показана несостоятельность некоторых мифов о возможностях перцептронных нейронных сетей и обосновывается переход к сверхбольшим перцептронным сетям глубокого доверия. Рассмотрены основные модели обучения нейронных сетей глубокого доверия, базирующиеся на ограниченной машине Больцмана (restricted Boltzmann machine (RBM)) и автоассоциативном подходе. Предложен новый метод для обучения ограниченной машины Больцмана и показано, что правило обучения ограниченной машины Больцмана является частным случаем предложенного метода обучения, который базируется на минимизации суммарной квадратичной ошибки восстановления информации. Доказана эквивалентность максимизации вероятности распределения данных в ограниченной машине Больцмана и минимизация суммарной квадратичной ошибки восстановления информации в слоях (RBM). Рассматривается применение нейронных сетей глубокого доверия для решения различных задач. Из-за ограниченности времени автор не успел рассмотреть «wake-sleep» алгоритм и рекуррентные нейронные сети глубокого доверия. Результаты, приведенные в данной лекции, получены в результате научных исследований по данной тематике и чтения автором специального курса лекций «Deep belief neural systems: theory and application» в Варшавском технологическом университете как приглашенный профессор. Автор благодарит проф. Варшавского технологического университета Янковского С. за плодотворные дискуссии.

СПИСОК ЦИТИРОВАННЫХ ИСТОЧНИКОВ

1. Розенблат, Ф. Принципы нейродинамики: Персептрон и теория механизмов мозга: Пер. с англ. – М.: Мир, 1965. – 480 с.
2. Minsky, M. Perceptrons: An introduction to Computational Geometry / M. Minsky, S. Papert. – The MIT Press. – 1969.
3. Hinton, G. A fast learning algorithm for deep belief nets / G. Hinton, S. Osindero, Y. Teh // Neural Computation. – 2006. – Vol. 18. – P. 1527–1554.
4. Hinton, G. Training products of experts by minimizing contrastive divergence // Neural Computation. – 2002. – Vol. 14. – P. 1771–1800.
5. Hinton, G. Reducing the dimensionality of data with neural networks / G. Hinton, R. Salakhutdinov // Science, 313 (5786). – 2006. – P. 504–507.
6. Hinton, G. A practical guide to training restricted Boltzmann machines // Tech. Rep. 2010-000). – Toronto: Machine Learning Group, University of Toronto, 2010.
7. Bengio, Y. Learning deep architectures for AI // Foundations and Trends in Machine Learning. – 2009. – Vol. 2(1). – P. 1–127
8. Bengio, Y. Greedy layer-wise training of deep networks / Y. Bengio, P. Lamblin, D. Popovici, H. Larochelle // In book Schölkopf, J.C. Platt, T. Hoffman (Eds.), Advances in neural information processing systems, 11. – MA: MIT Press, Cambridge, 2007. – P. 153–160.
9. Erhan, D. Why does unsupervised pre-training help deep learning? / D. Erhan, Y. Bengio, A. Courville, P.-A. Manzagol, P. Vincent, S. Bengio // Journal of Machine Learning Research. – 2010. – Vol. 11. – P. 625–660.
10. Widrow, B. Adaptive switching circuits / B. Widrow, M. Hoff // In 1960 IRE WESCON Convention Record. – DUNNO. – 1960. – P. 96–104.
11. Головки, В.А. Нейронные сети: обучение, организация и применение. – Кн. 4: учеб. пособие для вузов / Общая ред. А.И. Галушкина. – М.: ИПРЖР, 2001. – 256 с.
12. Rumelhart, D. Learning representation by backpropagation errors / D. Rumelhart, G. Hinton, R. Williams // Nature. – 1986. – N323. – P. 533–536.
13. Lippmann, R. P. An introduction to computing with neural nets // IEEEASSP Mag. – 1987. – vol. 4, no. 2. – P. 4–22.
14. Cybenko, G. Approximations by Superpositions of a Sigmoidal Function // Math. Contrl., Signals, Syst. – 1989. – Vol. 2. – P. 303–314.
15. Hinton, G. Greedy layer-wise algorithm // Journal of Machine Learning Research 1. – 2009. – P. 1–40.
16. Golovko, V. A Learning Technique for Deep Belief Neural Networks / V. Golovko, A. Kroshchanka, U. Rubanau, S. Jankowski // in book Neural Networks and Artificial Intelligence. – Springer, 2014. – Vol. 440. Communication in Computer and Information Science. – P. 136–146.
17. Golovko, V. Neural network model for transient ischemic attacks diagnostics / V. Golovko, H. Vaitsekhovich, E. Apanel, A. Mastyskin // Optical Memory and Neural Networks (Information Optics). – 2012. – 21(3). – P. 166–176.
18. Scholz, M., Fraunholz, M., Selbig, J. Nonlinear principal component analysis: neural network models and applications // In book Principal manifolds for data visualization and dimension reduction. – Springer Berlin Heidelberg, 2008. – P. 44–67.

Материал поступил в редакцию 30.12.14

GOLOVKO V.A., KROSHCHANKA A.A. Perceptrons and deep belief neural networks: training and application

This paper discusses and analyzes the basic paradigm of learning perceptron neural networks: from single layer perceptron to multilayer deep belief neural networks, which are considered now as a revolution in the field of data mining. It is shown the inconsistency of some of the myths about the possibilities of perceptron neural networks and is substantiated the transition to deep belief neural networks. The basic models of deep belief neural networks training are examined, which are based on the restricted Boltzmann machine (RBM) and auto-associative approach. A new method for training of RBM is proposed and is shown that traditional approach for restricted Boltzmann machine training is particular case of proposed technique, which is based on minimization of reconstruction square error. It is proved the equivalence of maximizing the probability distribution of the data in a restricted Boltzmann machine and the minimization of the total reconstruction squared error in layers of RBM. The application of deep belief neural network for compression, visualization and data recognition is considered.

УДК 681.4

Дудкин А.А.

ПОСТРОЕНИЕ ПОЛНОГО ИЗОБРАЖЕНИЯ ТОПОЛОГИЧЕСКОГО СЛОЯ ИНТЕГРАЛЬНОЙ СХЕМЫ ИЗ СОВОКУПНОСТИ ПЕРЕКРЫВАЮЩИХСЯ КАДРОВ

Введение. Современное производство интегральных схем (ИС) требует контроля всех критических процессов производства. Важной частью общей системы контроля являются системы технического зрения [1], в которых центральное место при обработке информации занимает идентификация объектов на изображениях.

Большие изображения, как правило, представляют совокупностью кадров с некоторым перекрытием соседних кадров. Задача формирования полного изображения по совокупности кадров получила название сшивки кадров.

Если сшиваемые кадры имеют реперные метки (специальные знаки, по которым следует выполнить сшивку соседних кадров), либо некоторые другие опорные точки (общие точки кадров), то задача сшивки сводится к пересчету координат в единую систему координат [2]. При этом, если реперами являются области, вначале находится положение, при котором сумма норм ошибок для выбранных K меток минимизируется:

$$M = \operatorname{argmin}_{m,n} \sum_{i=1}^K \|X_i - Y_i(m,n)\|, \quad (1)$$

где векторы X_i и Y_i представляют соответствующие i -е выбранные области кадров.

Сложность сшивки обусловлена наличием нежелательных эффектов, обусловленных вводом с использованием ПЗС-матриц, неодинаковыми характеристиками источника света линзы камеры, отличиями в освещенности совмещаемых изображений. Кроме того, имеют место геометрические искажения: отличия в масштабе, повороте, сдвиге, ракурсе совмещаемых изображений. Поэтому важно провести коррекцию оптических искажений и рассовмещений.

Коррекция рассовмещения включает в себя компенсацию продольных и поперечных сдвигов, поворотов при сохранении перпендикулярности; нарушения перпендикулярности: изменения масштаба и т.д. [3]. Различают аппаратную и алгоритмическую (программную) коррекцию рассовмещения. Ряд поправок определяется в процессе

Дудкин Александр Арсентьевич, доктор технических наук, доцент, зам. зав. лабораторией идентификации Объединенного института проблем информатики НАН Беларуси.