

О.І. Чумаченко, к.т.н, доц.

Г.О. Сіпаков, студ.

Національний технічний університет України «КПІ»

### ЗАСТОСУВАННЯ МЕРЕЖІ ГЛИБОКОЇ ДОВІРИ В ЗАДАЧІ РОЗПІЗНАВАННЯ ЗОБРАЖЕНЬ

З метою дослідження концепції глибокого навчання, зокрема підміну багат шарового перцептрона на відповідну мережу глибокої довіри, було проведено комп'ютерне моделювання процесу навчання на тестовій вибірці даних. Багат шаровий перцептрон був замінений на мережу глибокої довіри, що складається із послідовних обмежених машин Больцмана. Після навчання мережі глибокої довіри алгоритмом *layer-wise training* було виявлено, що використання мереж глибокої довіри значно покращує точність навчання багат шарового перцептрона методом зворотного розповсюдження помилки.

**Ключові слова:** нейронні мережі; перцептрон; глибоке навчання.

**Постановка проблеми.** На сьогодні нейронні мережі різних топологій використовуються для вирішення різноманітних задач, наприклад прогнозування, розпізнавання образів, зокрема обробка сигналів і зображень, аналіз рукописного тексту, кластеризація образів.

Універсальною топологією, що дозволяє вирішувати всі подібні завдання, є багат шаровий перцептрон. Багат шаровий перцептрон є мережею прямого поширення і навчається методом зворотного поширення помилки, який належить до групи навчання з учителем. Незважаючи на відносну простоту алгоритму, у нього є недолік – ризик «застрягти» в локальному мінімумі поверхні помилок, тоді алгоритм не прямуватиме до бажаного мінімуму і процес навчання не може бути закінчений.

Зі збільшенням розмірності задач, ускладнюється поверхня помилок. В умовах, коли образи описуються десятками або сотнями параметрів, навчання перцептрона методом зворотного поширення помилки дозволяє досягти лише певної точності, яка буде варіюватися в межах 60–80 %. Крім того, при великій розмірності потрібен перехід до даних меншої розмірності. Це викликає необхідність нарощувати шари перцептрона, а значить з кожним новим шаром, помилка навчання буде все менше впливати на налаштування параметрів перших шарів, і тим довше буде процес навчання.

**Аналіз останніх досліджень і публікацій.** В роботах [1–3] стверджується, що концепція глибокого навчання дозволяє вирішити задачі великої розмірності, зокрема задачу розпізнавання зображень. У роботі [1] йдеться про застосування двох базових елементів: обмеженої машини Больцмана та автоенкодера.

У роботі [4] йдеться про використання декількох послідовно з'єднаних обмежених машин Больцмана для реалізації мережі глибокої довіри.

Алгоритм навчання мережі глибокої довіри запропонований у роботі [5].

**Мета дослідження.** Аналіз методів навчання обмеженої машини Больцмана та мережі глибокої довіри. Оцінка точності навчання перцептрона після попереднього навчання.

**Викладення основного матеріалу.** Для вирішення завдання великої розмірності, наприклад, розпізнавання зображень використовується багат шаровий перцептрон з кількістю шарів  $M$  і кількістю нейронів в кожному шарі  $N_m$ . Задана навчальна вибірка  $S = (x, y_0)$ , де  $x$  – безліч векторів вхідних даних,  $y_0$  – бажаний вихід, відповідний вектору  $x_i$ .

Необхідно навчити перцептрон на даному наборі даних таким чином, щоб помилка навчання стала менше заданого мінімуму, тобто:

$$E = y - y_0 < \varepsilon \quad (1)$$

Використовувати алгоритм зворотного поширення помилки для задач великої розмірності неефективно. Для того, щоб вирішити поставлене завдання, замінюємо перцептрон мережею глибокої довіри. Структура мережі глибокої довіри показана на рисунку 1.

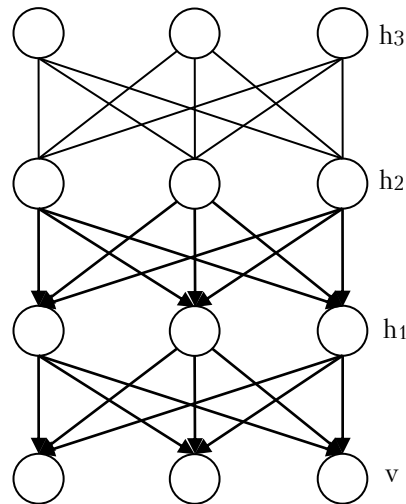


Рис. 1. Мережа глибинної довіри

Мережа глибинної довіри складається з декількох обмежених машин Больцмана (англійською Restricted Boltzmann Machine, скорочено – RBM), з'єднаних так, щоб невидимий шар однієї машини виконував роль видимого шару для наступної. Структура машини Больцмана показана на рисунку 2. Тут  $v_i$  – нейрони видимого шару,  $b_i$  – зміщення нейронів видимого шару,  $h_j$  – нейрони прихованого шару,  $c_j$  – зміщення нейронів прихованого шару і  $w_{ij}$  – вагові коефіцієнти зв'язків між шарами.

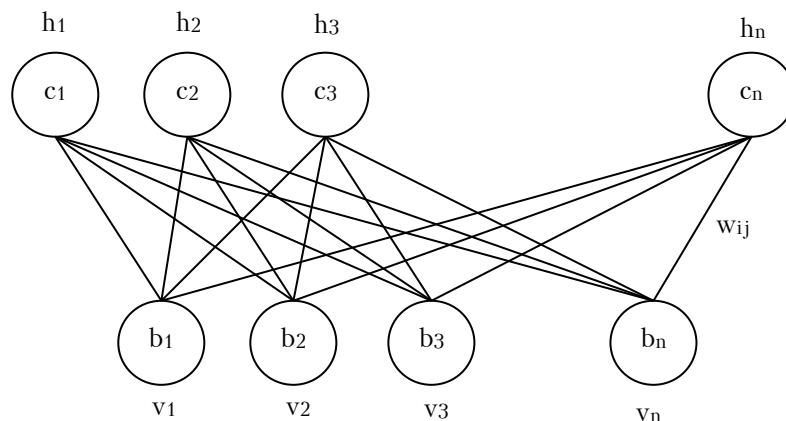


Рис. 2. Структура обмеженої машини Больцмана

Таким чином, завдання навчання перцептрона зводиться до задачі навчання обмежених машин Больцмана, об'єднаних в мережу глибинної довіри і використання вагових коефіцієнтів мережі глибинної довіри при навчанні перцептрона для більш точного навчання.

Навчання мережі глибинної довіри відбувається за алгоритмом layer-wise learning. У ході роботи даного алгоритму обмежені машини Больцмана навчаються послідовно, як показано на рисунку 3.

Алгоритм навчання мережі глибокої довіри можна представити у такому вигляді:

1. Навчити машину Больцмана, видимий шар якої збігається з початковим шаром мережі глибинної довіри (рис. 3), шляхом застосування алгоритму CD-1;
2. Прихований шар першої RBM приймається як видимий шар другий RBM і процес повторюється для другої і наступних RBM;
3. Далі вагові коефіцієнти  $W_{ij}^{(k)}$  ( $k$  – порядковий номер машини Больцмана в мережі глибинної довіри) використовуються для ініціалізації початкових ваг перцептрона.

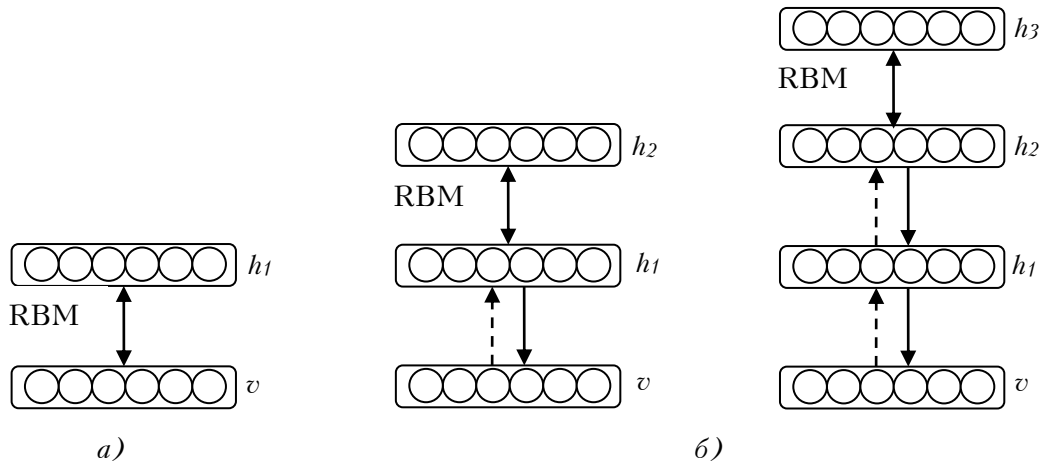


Рис. 3. Навчання мережі глибокої довіри

Як видно з алгоритму, основою навчання мережі глибокої довіри є алгоритм навчання кожної RBM окремо. В основі математичної моделі обмеженої машини Больцмана лежить енергетична функція. У загальному вигляді вона набуває вигляду:

$$E(\mathbf{v}, \mathbf{h}) = - \sum_{i \in vis} b_i v_i - \sum_{j \in hid} c_j h_j - \sum v_i h_j w_{ij} \quad (1)$$

Метою навчання обмеженої машини Больцмана є максимізація ймовірності отримати відображення вхідних даних, тобто максимізація повної ймовірності  $p(\mathbf{v})$ :

$$p(\mathbf{v}) = \sum_j p(\mathbf{v}, h_j) = \frac{1}{Z} \sum_j e^{-E(\mathbf{v}, h_j)} \quad (2)$$

$$Z = \sum_i \sum_j e^{-E(v_i, h_j)} \quad (3)$$

Якщо знайти диференціал ймовірності за параметрами (ваговим коефіцієнтам), то отримаємо формулу такого вигляду:

$$\frac{\partial P(\mathbf{v}^{(k)})}{\partial w_{ij}} = \frac{1}{Z^2} \left( Z \left( \sum_t v_i^{(k)} h_j^{(t)} e^{-E(v_i^{(k)}, h_j^{(t)})} \right) - \left( \sum_t e^{-E(v_i^{(k)}, h_j^{(t)})} \right) \left( \sum_r \sum_t v_i^{(r)} h_j^{(t)} e^{-E(v_i^{(r)}, h_j^{(t)})} \right) \right) \quad (4)$$

Якщо знайти похідну логарифма ймовірності, то отримуємо формулу

$$\frac{\partial \ln P(\mathbf{v}^{(k)})}{\partial w_{ij}} = \sum_t v_i^{(k)} h_j^{(t)} P(v_i^{(k)} | h_j^{(t)}) - \sum_r \sum_t v_i^{(r)} h_j^{(t)} P(v_i^{(r)} | h_j^{(t)}) \quad (5)$$

Оскільки  $\mathbf{v}^{(k)}$  – вхідні дані, а  $\mathbf{v}^{(r)}$  – деяке їх відображення, то за визначенням математичного очікування отримуємо формулу

$$\frac{\partial \ln P(\mathbf{v}^{(k)})}{\partial w_{ij}} = M[v_i^{(k)} h_j]_{data} - M[v_i h_j]_{model} \quad (6)$$

Таким чином, завдання навчання зводиться до мінімізації критерію. При цьому варто зауважити, що обчислення математичного очікування від даних не є важким завданням, тому що  $v_i^{(k)}$  відомо з навчальної вибірки, але обчислення другого параметра вимагає апроксимації.

Для цього Дж.Хінтон застосував алгоритм незалежного розходження (англійською Contrastive Divergence або CD-k). Сенс алгоритму полягає в тому, що математичні очікування замінюються безпосередніми значеннями, від яких беруться ці очікування, тобто:

$$M[v_i^{(k)} h_j]_{data} \approx \frac{\partial E(v_i^{(k)} h_j)}{\partial w_{ij}}, \quad (7)$$

$$M[\bar{v}_i \bar{h}_j]_{model} \approx \frac{\partial E(\bar{v}_i \bar{h}_j)}{\partial w_{ij}}$$

Опис алгоритму CD-k наведено нижче.

1) Початкові значення  $\mathbf{v}_0$  задати рівними вхідним даним.

2) Обчислити ймовірність того, що стан нейрона прихованого шару дорівнюватиме 1:

$$p(\mathbf{h}_j = 1 | \mathbf{v}) = \sigma\left(\sum_{i=1}^m w_{ij} v_i + b_j\right) \quad (8)$$

де  $c_i$  - зміщення прихованого шару, а  $\sigma$  - функція.

$$\frac{\partial \ln P(\mathbf{v}^{(k)})}{\partial w_{ij}} = \sum_t v_i^{(k)} h_j^{(t)} P(v_i^{(k)} | h_j^{(t)}) - \sum_r \sum_t v_i^{(r)} h_j^{(t)} P(v_i^{(r)} | h_j^{(t)}) \quad (9)$$

$$\sigma = \frac{1}{1 + e^{-x}} \quad (10)$$

3) Присвоїти нейрону прихованого шару стан згідно з ймовірністю (якщо ймовірність 0,9, то з ймовірністю 90 % стан нейрона буде 1, а з ймовірністю 10 % - 0).

4) Обчислити ймовірність того, що стан нейрона видимого шару дорівнюватиме 1:

$$p(v_i = 1 | \mathbf{h}) = \sigma\left(\sum_{j=1}^m w_{ij} h_j + c_j\right) \quad (11)$$

де  $c_j$  - зміщення видимого шару, а  $\sigma$  - функція.

5) Присвоїти нейрону видимого шару стан виходячи з ймовірністю

6) Змінити параметри за таким принципом:

1. для синаптичних зв'язків

$$w_{ij}(k) = w_{ij}(0) + \alpha(v_i(0)h_j(0) - v_i(k)h_j(k)) \quad (12)$$

2. для зсувів видимого шару

$$b_i(k) = b_i(0) + \alpha(v_i(0) - v_i(k)) \quad (13)$$

3. для зсувів прихованого шару

$$c_j(k) = c_j(0) + \alpha(h_j(0) - h_j(k)) \quad (14)$$

$\alpha$  - швидкість навчання.

7) Повторювати обчислення з кроку 2 до тих пір, поки номер ітерації менше  $k$ .

Як видно з таблиці 1, використання глибинного навчання значно підвищує точність роботи нейромережі. Разом із тим варто зазначити, що використання лише однієї ітерації алгоритму CD-k дозволяє досягти значних покращень точності.

Таблиця 1

Результати навчання

	Без глибинного навчання	CD-10	CD-1
Точність навчання	63,4%	97%	90,2%

**Висновки.** Застосування методів глибинного навчання для вирішення задач великої розмірності дозволяє суттєво підвищити точність навчання, що було перевірено при комп'ютерному моделюванні. Запропонований підхід дозволяє вирішувати великий спектр задач, таких як прогнозування, аналіз рукописного тексту, розпізнавання голосу, тощо.

Список використаної літератури:

1. *Bengio Y.* Learning deep architectures for AI // *Y.Bengio* // Foundations and Trends in Machine Learning. – 2009. – № 21 (6). – С. 1601–1621.
2. Greedy Layer-Wise Training of Deep Networks // *Y.Bengio, P.Lamblin, D.Popovici, H.Larochelle* // Advances in Neural Information Processing Systems. – 2007. – № 19. – С. 153–160.
3. *Deng L.* Deep Learning: Methods and Applications // *L.Deng, D.Yu* // Foundations and Trends in Signal Processing. – 2013. – № 7. – С. 197–387.
4. *Goodfellow I.* Deep learning of representations and its application to computer vision // *I.Goodfellow*. – Avril, 2014. – 139 с.
5. *Hinton G.E.* A fast learning algorithm for deep belief nets // *G.E. Hinton, S.Osindero, Y.The* // Neural Computation. – 2006. – № 18. – С. 1527–1554.

6. Hinton G.E. Training products of experts by minimizing contrastive divergence / G.E. Hinton // *Neural Computation*. – 2002. – № 14 (8). – С. 1711–1800.
7. Hinton G.E. Reducing the dimensionality of data with neural networks / G.E. Hinton, R.Salakhutdinov // *Science*. – 2006. – № 313. – С. 504–507.
8. Salakhutdinov R. Deep Boltzmann machines / R.Salakhutdinov, G.E. Hinton // *Proceedings of The Twelfth International Conference on Artificial Intelligence and Statistics (AISTATS'09)*. – 2009. – № 5. – С. 448–455.
9. Salakhutdinov R. An efficient learning procedure for deep Boltzmann machine // R.Salakhutdinov, G.E. Hinton // *Neural computation*. – 2012. – № 24 (8). – С. 1967–2006.
10. Utgoff P. Many-layered learning / P.Utgoff, D.Stracuzzi // *Neural Computation*. – 2002. – № 14. – С. 2497–2539.

#### References:

1. Bengio, Y. (2009), “Learning deep architectures for AI”, *Foundations and Trends in Machine Learning*, No. 21 (6), pp. 1601–1621.
2. Bengio, Y., Lamblin, P., Popovici, D. and Larochelle, H. (2007), “Greedy Layer-Wise Training of Deep Networks”, *Advances in Neural Information Processing Systems*, No. 19, pp. 153–160.
3. Deng, L. and Ju, D. (2013), “Deep Learning: Methods and Applications”, *Foundations and Trends in Signal Processing*, No. 7, pp. 197–387.
4. Goodfellow, I. (2014), *Deep learning of representations and its application to computer vision*, Avril, 139 p.
5. Hinton, G.E., Osindero, S. and The, Y. (2006), “A fast learning algorithm for deep belief nets”, *Neural Computation*, No. 18, pp. 1527–1554.
6. Hinton, G.E. (2002), “Training products of experts by minimizing contrastive divergence”, *Neural Computation*, No. 14 (8), pp. 1711–1800.
7. Hinton, G.E. and Salakhutdinov, R. (2006), “Reducing the dimensionality of data with neural networks”, *Science*, No. 313, pp. 504–507.
8. Salakhutdinov, R. and Hinton, G.E. (2009), “Deep Boltzmann machines”, *Proceedings of The Twelfth International Conference on Artificial Intelligence and Statistics (AISTATS'09)*, No. 5, pp. 448–455.
9. Salakhutdinov, R. and Hinton, G.E. (2012), “An efficient learning procedure for deep Boltzmann machine”, *Neural computation*, No. 24 (8), pp. 1967–2006.
10. Utgoff, P. and Stracuzzi, D. (2002), “Many-layered learning”, *Neural Computation*, No. 14, pp. 2497–2539.

ЧУМАЧЕНКО Олена Іллівна – кандидат технічних наук, доцент кафедри технічної кібернетики Національного технічного університету України «КПІ».

Наукові інтереси:

- системний аналіз;
- штучні нейронні мережі.

СПАКОВ Геннадій Олександрович – студент кафедри технічної кібернетики Національного технічного університету України «КПІ».

Наукові інтереси:

- нейронні мережі;
- глибинне навчання.

Стаття надійшла до редакції 13.09.2016.