УДК 519.21

Бобиль Б.В., аспірант
Терещенко В.М., д.ф-м.н., професор

Аналіз методів передтренування та ініціалізації нейронних мереж.

Київський національний університет імені Тараса Шевченка, 03680, м. Київ, пр-т. Академіка Глушкова 4д,
e-mail: bobylbohdan@gmail.com, v_ter@ukr.net

B.V. Bobyl, postgraduate
V.M. Tereschenko, Dr.Sci.(Phis.-Math.), Assoc. Prof.

Analysis of pre-training and initialization methods of neural networks.

Taras Shevchenko National University of Kyiv, 03680, Kyiv, Glushkova st., 4d,
e-mail: bobylbohdan@gmail.com, v_ter@ukr.net

У даній роботі розглянуті основні методи передтренування та ініціалізації значень параметрів нейронних мереж - передтренування мережі з використанням обмежених машин Больцмана, глибокі автокодувальники, ініціалізація параметрів методами Хе та Глоро, перенесення знань та доменна адаптація. Дані методи застосовуються для знаходження початкових значень параметрів нейронної мережі та їх попередньої ініціалізації, що є необхідною умовою для подальшого ефективного навчання глибоких моделей та дозволяє зменшити вплив негативних ефектів, які виникають під час навчання - затухання або вибуху градієнта, перенавчання, застрягання в одному з локальних мінімумів функції втрат, тощо. Дані алгоритми відносяться до групи алгоритмів навчання без учителя і не потребують розмітки для даних, на яких буде навчатися модель після ініціалізації. У статті був проведений аналіз цих методів, описані переваги та недоліки кожного алгоритму. Описано результати експериментів з використанням цих методів для вирішення задачі класифікації бази даних MNIST та запропоновані ідеї покращення алгоритмів передтренування нейронної мережі.

Ключові слова: нейронна мережа, передтренування, обмежена машина Больцмана, алгоритм порівняльної розбіжності, автокодувальник, ініціалізація Глоро, ініціалізація Хе, доменна адаптація, коефіцієнт Жаккара.

In this paper we investigate main pre-training and initialization methods of parameter values of neural networks such as pre-training using restricted Boltzmann machines, deep autoencoders, Glorot and He initialization of parameters, transfer learning and domain adaptation. Given methods are useful for finding of appropriate parameter values and initial initialization of neural network, what is necessary condition for further efficient training of deep models, because it give a possibility during training to reduce negative effects such as vanishing or explosion of gradient, overfitting, stucking in one of local minimums of loss function, etc. These methods belong to group of unsupervised training algorithms and do not need any labeling for data which will be used later for model's training after parameters initialization. Firstly, in this paper, we analyze all these methods and describe advantages and disadvantages of each of them. Secondly, we describe results of our experiments applying these methods for solving of classification task of MNIST dataset and introduce ideas for further development and improvement of these algorithms.

Key Words: neural network, pre-training, restricted Boltzmann machines, contrastive divergence algorithm, autoencoder, Glorot initialization, He initialization, domain adaptation, Jaccard index.

Статтю представив д. ф.-м. н., проф. Анісімов А.В.

1. Вступ

Машинне навчання та нейронні мережі зокрема стали потужним інструментом для вирішення низки прикладних задач сучасності - розпізнавання образів [6, 12], машинний переклад [2, 3], прогнозування поведінки покупців, тощо. Причиною цього став ріст обчислювальних потужностей, доступних на даний момент та накопичення величезних об'ємів даних (як структурованих і неструктурованих), які потрібно обробляти і діставати з них нові знання.

Коли ми чуємо про успіхи у вирішенні "інтелектуальних" задач (під інтелектуальними будем мати на увазі задачі, які не піддаються повному математичному опису і не мають детермінованих точних алгоритмів рішення, таких як класифікація, локалізація об'єктів на зображенні, переклад текстів з однієї природної мови на іншу, тощо), то саме нейронні мережі на даний момент показують найкращі результати в більшості випадків. Проте, на відміну від сьогодення, коли глибоке навчання впроваджується майже всюди, до 2005-2006 року, в цій області науки працювало відносно невелика кількість вчених і визначних результатів майже не було. Одна із причин цього - складність навчання глибоких нейронних мереж. Для навчання мережі необхідно: 1) достатньо великі об'єми даних (чим більш складна задача вирішується, тим більш складною має бути архітектура мережі і тим більше потрібно даних); 2) обчислювальні потужності (достатньо потужні нейронні мережі вимагають багато ресурсів, особливо під час фази навчання). Ці проблеми стали менш суттєвими саме на початку XXI століття - поява мережі Інтернет та багатоядерних процесорів, а потім - обчислень на графічних процесорах сприяло цьому. Це повернуло інтерес до вивчення нейронних мереж вченою спільнотою. Проте виявилося, що є одна невирішена проблема: так як тренування моделі завжди зводиться до мінімізації функції втрат, ми далеко не завжди маємо можливість використовувати складні алгоритми оптимізації для глибоких мереж по причині їх обчислювальної витратності.

Додатково, у мережі, в силу своїх архітектурних особливостей, можуть виникати небажані ефекти під час навчання: затухання градієнта, "градієнтний вибух", "перенавчання" (overfitting в англомовній літературі - процес адаптації моделі до навчальних даних, під час якого мережа добре працює на начальних даних, але дає неправильні прогнози на валідаційних даних; характерно для ситуацій коли немає можливості отримати достатню кількість даних для навчання/валідації).

У даній статті будуть розглянуті методи і алгоритми, які в тій чи іншій мірі допомагають боротися з цими проблемами і дають змогу навчати глибокі моделі - передтренування без учителя [8, 9], спеціальні методи випадкової ініціалізації параметрів моделі [6, 7], перенесення знань та доменна адаптація [11], а також описані ідеї покращення існуючих алгоритмів.

2. Передтренування без учителя

Передтренування без учителя (unsupervised pre-training в англомовній літературі) - сімейство алгоритмів, які працюють на нерозмічених даних. У даний момент практично дані алгоритми практично не використовуються для передтренування, проте вони мають значну історичну цінність і є джерелом ідей для побудови алгоритмів для вирішення інших задач.

Першим суттєвим проривом в передтренуванні нейронних мереж стала поява алгоритму без учителя, базованого на *обмежених машинах Больцмана* (restricted Boltzmann machines), розробленого групою Джефрі Хінтона [9]. Основна ідея методу - будемо вважати, що існують видимі ознаки, які представляють наші дані та невидимі(латентні) ознаки, які нам невідомі і між ними існує зв'язок, таким чином необхідно знайти такі параметри моделі, які дадуть змогу перейти із простору видимих ознак

в простір невидимих, при цьому невидимі ознаки мають давати змогу достатньо якісно відновити оригінальні дані зворотнім перетворенням.

Навчання моделі проводиться *алгоритмом порівняльної розбіжності* (contrastive divergence в англомовній літературі), розробленого також групою Хінтона [10]. Для кожного шару мережі (вихід попереднього шару є входом для поточного, виключенням є перший шар - для нього входом є оригінальні дані), алгоритмом порівняльної розбіжності проводиться максимізація функції правдоподібності вхідних даних по параметрах шару:

$$W^* = argmax_W \prod_{x \in X} P(x; W)$$

Основний недолік цього методу - складність налаштування алгоритму і необхідність вирішення додаткової задачі, яка по складності може не поступатися основній. Порівняння навчання нейронної мережі з передтренуванням обмеженою машиною Больцмана зображено на Рис.1.

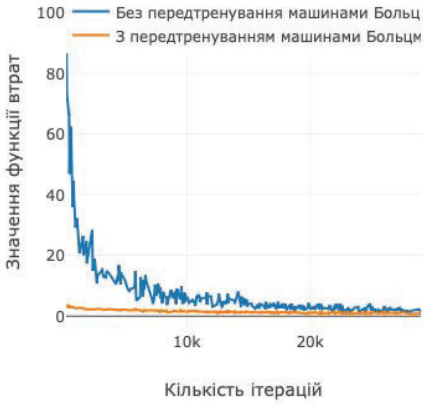

Рис.1 - Порівняння навчання нейронної мережі на базі цифр MNIST з використанням обмежених машин Больцмана

Інший спосіб передтренування мережі - використовувати автокодувальники, які можна вважати розвитком методів, базованих на обмежених машинах Больцмана.

Автокодувальник [12] - архітектура нейронної мережі, яка може використовуватися для багатьох задач - породження нових даних, зменшення зашумленості даних, стиснення даних і передтренування. Ідея полягає в наступному: аналогічно, як і в обмежених машинах Больцмана, будемо вважати, що існують видимі ознаки, які представляють наші дані, та приховані ознаки, які теж представляють наші дані та аналогічно як і в машинах Больцмана, ми повинні мати змогу по прихованих ознаках відновити оригінальні дані. Для цього використовується підхід "кодувальник-декодувальник" - мережа складатиметься із 2-х частин: 1) кодувальника, який перетворюватиме дані в точки латентного простору ознак; 2) декодувальника, який перетворюватиме точки, отримані від кодувальника в оригінальні дані. Автокодувальник із одним прихованим шаром зображено на Рис.2

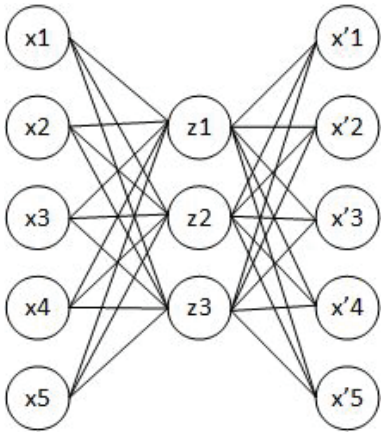

Рис.2 - Архітектура автокодувальника: *x1-x5* - вхід мережі, *z1-z3* - приховані(латентні) ознаки, *x'1-x'5* - відновлений вхід

Таким чином, натренувавши таку мережу, можна взяти ваги кодувальника та проініціалізувати ними частину параметрів іншої мережі, яка вирішуватиме основну задачу. Цей підхід належить до класу алгоритмів без учителя і є розвитком підходу із застосуванням обмежених машин Больцмана. Основна відмінність - машини Больцмана тренують за один крок один шар, який фактично виконує роль кодувальника та декодувальника одночасно. Автокодувальник же складається з 2-х підмереж,

кожна може складатися з багатьох шарів і стандартно подібна архітектура тренується в end-to-end стилі. При цьому методом градієнтного спуску зазвичай мінімізується функція середньоквадратичної похибки між входом і виходом мережі:

$$W^* = argmin_W \frac{1}{2N} \sum_{i=1}^{N} (h(x_i) - x_i)^2,$$

де $h(\cdot)$ - вихід автокодувальника.

Основна перевага цього методу - він дещо простіший, ніж обмежені машини Больцмана, тренується стандартним градієнтним спуском і розповсюдженням помилки, дає можливість проініціалізувати мережу не випадковими параметрами, а отриманими на основі апріорної інформації, закладеної в дані. Основний недолік - після навчання та ініціалізації, декодувальник не потрібен, при цьому нам потрібно у 2 рази більше даних для навчання всієї мережі (при умові, що кількість параметрів у кодувальнику та декодувальнику приблизно однакова).

3. Випадкова ініціалізація методами Глоро і Хе

Великий вклад у розвиток методів ініціалізації зробила спільна робота Ксав'є Глоро та Йошуа Бенджи [6], що вийшла у 2010 році. Ці дослідники поставили перед собою питання існування можливості навчання нейронної мережі повністю як одне ціле (в англомовній літературі - end-to-end learning). Достовірно відомо, що нейронна мережа чутлива до початкової ініціалізації параметрів. Основним результатом їх роботи став простий алгоритм ініціалізації, який покращив як швидкість, так і якість навчання та отримав назву *ініціалізація Глоро* (в англомовній літературі зустрічається назва Xavier/Glorot initialization).

Розглянемо значення одного нейрону (до застосування функції активації):

$$y = w^\top x + b = \sum_i w_i x_i + b,$$

де x - вектор вхідних значень, w - вектор параметрів. Таким чином, дисперсія $Var(y)$ не залежить від вільного члена b, а залежить тільки від вектора вхідних значень і вектора параметрів. Позначимо i-й член суми як $y_i = w_i x_i$. Припустимо, що x_i та w_i - незалежні, маємо:

$$Var(y_i) = Var(w_i x_i) = E[w_i x_i] - (M[w_i x_i])^2 =$$
$$= M[x_i]^2 Var(w_i) + M[w_i]^2 Var(x_i) + Var(x_i) Var(w_i),$$

де $M[\cdot]$ - оператор взяття математичного сподівання.

Припустимо, що ми використовуємо симетричну функцію активації та випадково ініціалізуємо параметри значеннями, середнє значення яких дорівнює нулю. Тоді дисперсія матиме наступний вигляд:

$$Var(y_i) = Var(x_i) Var(w_i)$$

Тепер, якщо припустити, що x_i та w_i ініціалізуються з одного розподілу, при цьому незалежно, маємо:

$$Var(y) = Var(\sum_{i=1}^{n_{out}} y_i) = \sum_{i=1}^{n_{out}} Var(w_i x_i) =$$
$$= n_{out} Var(x_i) Var(w_i),$$

де n_{out} - кількість нейронів останнього шару.

Бачимо, що дисперсія виходу пропорційна дисперсії входу із коефіцієнтом $n_{out} Var(w_i)$. Якщо використовується симетрична функція активації для якої для k-го шару з одиничною похідною в околі нуля, то $f'(y_i^{(l)}) \approx 1$, то отримаємо коефіцієнт пропорційності $n_{in} Var(w_i)$, де n_{in} - кількість кількість входів. Ідея Бенджи і Глоро полягає в тому, що для безперешкодного проходження значень активації та градієнтів по мережі, дисперсії в обох випадках мають бути близькими до одиниці. Оскільки для шарів різних розмірів не можна задовольнити обидві умови одночасно, було запропоновано ініціалізувати ваги симетричним розподілом з наступною дисперсією:

$$Var(w_i) = \frac{2}{n_{in} + n_{out}},$$

що для рівномірної ініціалізації приводить до розподілу:

$$w_i \sim U(-\frac{\sqrt{6}}{\sqrt{n_{in} + n_{out}}}, \frac{\sqrt{6}}{\sqrt{n_{in} + n_{out}}})$$

Для несиметричних функцій активацій (таких як ReLU), використовується *ініціалізація Хе* [7], розроблена і опублікована Кайміном Хе у

2015 році. Використовуючи аналогічні міркування, можна отримати:

$$Var(w_i) = \frac{2}{n_{in}^{(l)}},\ w_i \sim N(0,\ \sqrt{\frac{2}{n_{in}^{(l)}}})$$

Перевагою цих методів є простота реалізації і відносно висока якість ініціалізації, проте ми не можемо закласти апріорні знання в модель, так як при ініціалізації ці знання не враховуються.

Результати роботи обох методів зображено на Рис.3 та Рис.4.

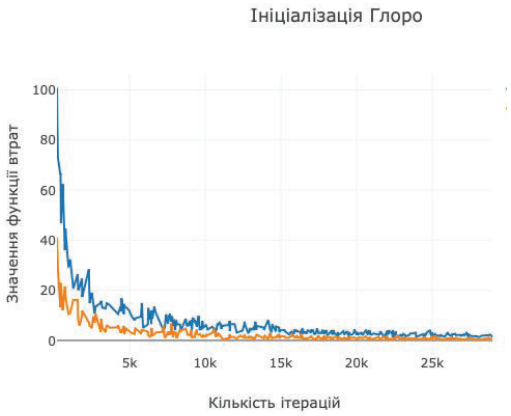

Рис.3 - Порівняння навчання нейронної мережі на базі цифр MNIST з використанням ініціалізації Глоро

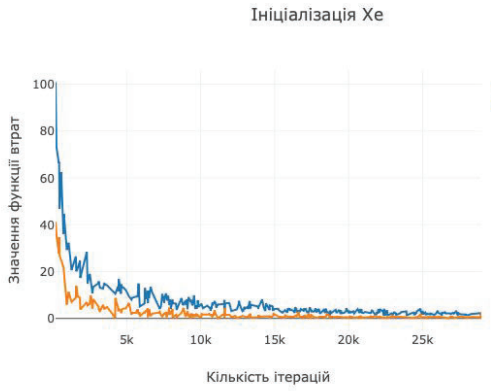

Рис.4 - Порівняння навчання нейронної мережі на базі цифр MNIST з використанням ініціалізації Хе

4. Перенесення знань та доменна адаптація

Перенесення знань (transfer learning) та *доменна адаптація моделі* [11] - потужні методики, які застосовуються в передтренуванні мережі.

На даний момент існує велика кількість якісно натренованих моделей для вирішення багатьох задач - моделі натреновані для вирішення задачі класифікації [6, 12], локалізації та детекції з багатьма класами об'єктів [5, 6] в області комп'ютерного зору, тощо. Зазвичай, дуже складно знайти модель, яка тренувалася на даних, розподіл яких подібний до розподілу даних конкретної задачі. Відповідно, моделі натреновані на одних даних можуть погано працювати на входах, які мають інший розподіл, ніж тренувальний набір даних. Наприклад класифікатор, який навчався на зображеннях, зроблених вдень, добре працюватиме на зображеннях знятих вдень, але буде помилятися на зображеннях знятих вночі. Виникає питання - чи можливо, незважаючи на це обмеження, перевикористати існуючу модель для рішення конкретної задачі? Головна ідея методу доменної адаптації або дотренування полягає в наступному: замість тренування мережі повністю з нуля, можна використати уже натреновану мережу, яка натренована для вирішення задачі дуже близької до нашої і дотренувати її на наших специфічних даних. Як результат, мережа адаптується до нового розподілу даних і дає вірні передбачення на подібних даних. Цей метод має багато переваг і добре себе зарекомендував в області комп'ютерного зору: існує багато готових моделей, наприклад для задач класифікації та детекції, які вирішують достатньо велике коло задач і їх можна використати. Ще одна перевага - для дотренування мережі зазвичай потрібно на порядок менше навчальних даних. Проте, якщо вирішувана задача достатньо специфічна або існуючі натреновані моделі через свої архітектурні особливості не дають змоги їх використовувати (приклад - мережа має занадто мало параметрів і не може виявити необхідні закономірності в даних або навпаки - має занадто

багато параметрів і є обчислювально витратною), то даний метод не застосовний в повній мірі.

Ще одна можлива проблема - необхідність змінити архітектуру моделі, наприклад, додавши додаткові виходи у модель, або додавши приховані шари. Навіть у цьому випадку ми можемо використати частину параметрів мережі. Для того, щоб зрозуміти принцип роботи методу перенесення знань, потрібно згадати принцип роботи нейронної мережі. Розглянемо мережу глибини n (де під глибиною мається на увазі кількість шарів), де під глибиною мається кількість шарів і розглянемо процес обчислень, який виконує мережа під час навчання і прогнозування. Кожен шар мережі під час навчання вивчає певні шаблони в даних. При цьому, чим глибший шар, тим більш складні шаблони він вивчає. Наприклад, припустимо, що ми навчаємо мережу для класифікації зображень - модель має давати відповідь чи є на зображені людина, чи ні. У цьому випадку, ранні шари мережі вивчають примітивні ознаки - наявність ліній різного нахилу та товщини, наявність певних кольорів, тощо. Більш глибокі шари вивчають більш складні ознаки, комбінуючи виходи попередніх - шукаючи дуги, овали та інші більш складні шаблони. Останні шари мережі на основі виходу попередніх здатні вивчити частини тіла - голову, очі, руки, тощо. Останній же шар на основі цієї інформації приймає рішення про наявність людини на зображенні. Схематично процес зображено на Рис. 5

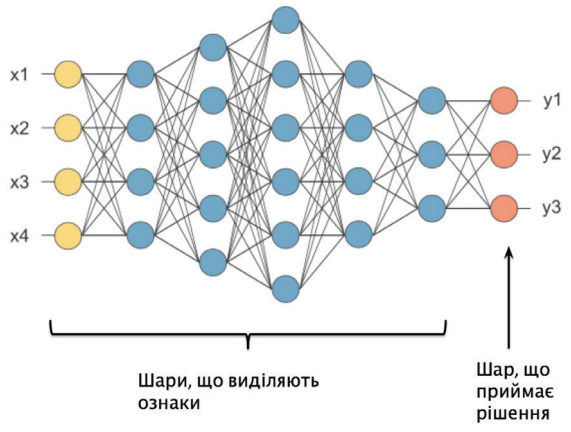

Рис. 5 - Роль шарів в нейронній мережі прямого поширення

Таким чином, ми можемо перевикористати частину параметрів перших k шарів однієї мережі. Ця техніка досить корисна, особливо у випадках коли потрібно розширити функціональні можливості нашої моделі: очевидно, маючи модель, яка класифікує зображення для виявлення присутності людини на фото, можна створити нову модель, яка додатково прогнозуватиме положення людини, при цьому достатньо взяти ваги першої моделі і перенести їх в другу та дотренувати її. Недоліки даного підходу - через специфічність задачі, можлива відсутність натренованих моделей, які вирішують подібні задачі.

Результати використання даного методу зображено на Рис.6

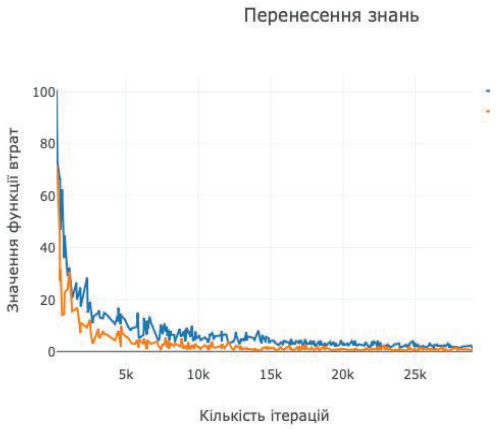

Рис.6 - Приклад навчання нейронної мережі на підмножині бази MNIST з перенесенням значень параметрів іншої модель, навченої на іншій підмножині цифр цієї ж бази

5. Ідеї покращення алгоритмів передтренування

Одним з найперспективніших напрямків є модифікація методів передтренування, що базуються на обмежених машинах Больцмана. Основна перевага цього методу - можливість закласти певні апріорні знання і розподіли, які є в даних та виконувати навчання пошарово. Але складність налаштування та навчання мережі алгоритмом порівняльної розбіжності робить даний підхід менш практичним відносно інших

підходів. Основні ідеї для покращення роботи даного підходу: 1) замінити алгоритм порівняльної розбіжності на алгоритм градієнтного спуску із зворотнім розповсюдженням помилки; 2) замінити стандартні функції втрат такі, як, наприклад, середньоквадратична помилка або бінарна кросентропія на функцію, яка буде штрафувати модель (наприклад, для задачі класифікації) за вивчення ознак, які не розділяють або слабо розділяють задані класи. Такою функцією може бути *коефіцієнт Жаккара*:

$$J(A,\ B) = \frac{|A \cap B|}{|A \cup B|},$$

який використовується в області комп'ютерного зору для оцінки якості роботи детекторів об'єктів на зображенні та відомий під назвою *"intersection over union"-метрикою*. Таким чином, наприклад, при класифікації об'єктів, чим менше перетинаються ознаки об'єктів різних класів, тим менше будуть штрафуватися параметри шару і менше буде помилка класифікації, що і є основною метою алгоритму.

Область досліджень, що розглядається, є джерелом ідей для покращення роботи існуючих алгоритмів та методів. У цій статті були проаналізовані основні та найпоширеніші методи передтренування та ініціалізації мереж, були зазначені їхні переваги та недоліки, області їх застосовності. Дані методи в тій чи іншій мірі можуть бути застосовані для широкого кола задач та архітектур мережі, проте в цьому є певний недолік - наклавши певні обмеження або використовуючи більш специфічні методи, можна отримати кращі результати в обмеженому колі задач. Наприклад, сконцентрувавшись на задачі класифікації, можна отримати значно кращі методи ініціалізації, так як для даної задачі важливо, щоб модель вивчила дискримінативні ознаки. При цьому і машини Больцмана, і автокодувальники шукають генеративні ознаки, так як вони в тій чи іншій мірі мають вміти відновлювати дані, що може корисно для семплінга нових даних, але не потрібно для класифікації.

6. Висновки

Список використаних джерел

1. *van der Maaten L. J. P.* Visualizing High-Dimensional Data Using t-SNE [Електронний ресурс] / L. J. P. van der Maaten, G. E. Hinton // Journal of Machine Learning Research. – 2008. – Режим доступу до ресурсу: http://www.jmlr.org/papers/v9/vandermaaten08a.html.
2. *Goodfellow I.* Deep Learning / I. Goodfellow, Y. Bengio, A. Courville. – Cambridge, MA: The MIT Press, 2016. – 800 с. – (Adaptive Computation and Machine Learning series).
3. *Bishop C. M.* Neural networks and machine learning / C. M. Bishop. – Berlin: Springer, 1998. – 353 с. – (Nato ASI Subseries F).
4. *Bishop C. M.* Pattern recognition and machine learning / C. M. Bishop. – New York: Springer-Verlag, 2006. – 738 с. – (Information Science and Statistics).
5. *Glorot X.* Understanding the difficulty of training deep feedforward neural networks

References

1. VAN DER MAATEN, L. J. P. (2008) Visualizing High-Dimensional Data Using t-SNE. *[Electronic resource] Journal of Machine Learning Research.* Access: http://www.jmlr.org/papers/v9/vandermaaten08a.html.
2. GOODFELLOW, I., BENGIO, Y., COURVILLE, A., (2016) Deep Learning. *Cambridge, MA: The MIT Press*. – 800 p. – (Adaptive Computation and Machine Learning series).
3. BISHOP, C. M. (1998) Neural networks and machine learning. *Berlin: Springer*. – 353 p. – (Nato ASI Subseries F).
4. BISHOP, C. M. (2006) Pattern recognition and machine learning. *New York: Springer-Verlag.* – 738 p. – (Information Science and Statistics).
5. GLOROT, X., BENGIO, Y. (2010) Understanding the difficulty of training deep feedforward neural networks. *[Electronic*

[Електронний ресурс] / X. Glorot, Y. Bengio // Journal of Machine Learning Research. – 2010. http://proceedings.mlr.press/v9/glorot10a/glorot10a.pdf.

6. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification / K. He, X. Zhang, S. Ren, S. Sun // 2015 IEEE International Conference on Computer Vision (ICCV) / K. He, X. Zhang, S. Ren, S. Sun., 2015. – С. 1026–1034.
7. *Hinton G. E.* Reducing the Dimensionality of Data with Neural Networks / G. E. Hinton, R. R. Salakhutdinov // Science, 313 / G. E. Hinton, R. R. Salakhutdinov. – New York, 2006. – С. 504–507.
8. *Salakhutdinov R. R.* Restricted Boltzmann machines for collaborative filtering / R. R. Salakhutdinov, A. Mnih, G. E. Hinton // ACM International Conference Proceeding Series, 227 / R. R. Salakhutdinov, A. Mnih, G. E. Hinton., 2007. – С. 791–798.
9. *Carreira-Perpinan M. A.* On Contrastive Divergence Learning / M. A. Carreira-Perpinan, G. E. Hinton // AISTATS 10th Int. Workshop on Artificial Intelligence and Statistics / M. A. Carreira-Perpiñán, G. E. Hinton., 2005. – С. 59–66.
10. Discriminability-Based Transfer between Neural Networks / L. Y.Pratt, S. J. Hanson, C. L. Giles, J. D. Cowan // Advances in Neural Information Processing Systems, 5 / L. Y. Pratt, S. J. Hanson, C. L. Giles, J. D. Cowan., 1993. – С. 204–211.
11. Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion [Електронний ресурс] / P.Vincent, H. Larochelle, I. Lajoie, Y. Bengio // Journal of Machine Learning Research. – 2010. – Режим доступу до ресурсу: http://www.jmlr.org/papers/volume11/vincent10a/vincent10a.pdf.
12. Deep Residual Learning for Image Recognition / K.He, X. Zhang, S. Ren, J. Sun // 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) / K.He, X. Zhang, S. Ren, J. Sun. – Las Vegas, NV: IEEE, 2016. – С. 770–778.

resource] Journal of Machine Learning Research. Access: – Режим доступу до ресурсу: http://proceedings.mlr.press/v9/glorot10a/glorot10a.pdf.

6. HE, K., ZHANG, X., REN, S., SUN, S. (2015) Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. *2015 IEEE International Conference on Computer Vision (ICCV).* – p. 1026–1034.
7. HINTON, G. E., SALAKHUTDINOV, R. R. (2016) Reducing the Dimensionality of Data with Neural Networks. *New York: Science*. – 313. – p. 504–507.
8. SALAKHUTDINOV, R. R., MNIH, A., HINTON, G. E. (2007) Restricted Boltzmann machines for collaborative filtering. *ACM International Conference Proceeding Series*. – 227. – p. 791–798.
9. CARREIRA-PERPINAN, M. A., HINTON, G. E. (2005) On Contrastive Divergence Learning. *AISTATS 10th Int. Workshop on Artificial Intelligence and Statistics.* – p. 59–66.
10. PRATT, L. Y., HANSON, S. J., GILES, C. L., COWAN, J. D., (1993) Discriminability-Based Transfer between Neural Networks. *Advances in Neural Information Processing Systems*. – 5. – p. 204–211.
11. VINCENT, P., LAROCHELE, H., LAJOIE, I., BENGIO, Y. (2010) Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion. *[Electronic resource] Journal of Machine Learning Research.* Access: http://www.jmlr.org/papers/volume11/vincent10a/vincent10a.pdf.
12. HE, K., ZHANG, X., REN, S., SUN, S. (2016) Deep Residual Learning for Image Recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Las Vegas, NV: IEEE, 2016.* – p. 770–778.

Received: 17.06.2018