DOI https://doi.org/10.15407/usim.2019.02.025
УДК 004.023
**І.В. СТЕЦЕНКО**, доктор. техн. наук, професор. кафедра автомат. систем обробки інформації та управління, Національний техн. ун-т України "Київський політехнічний інститут імені Ігоря Сікорського", (НТУУ «КПІ ім. І. Сікорського»), просп. Перемоги, 37, Київ, 03056, Україна, stiv.inna@gmail.com
**Ю.С. ТАЛЬКО**, студент кафедри автомат. систем обробки інформації та управління, Національний техн. ун-т України "Київський політехнічний інститут імені Ігоря Сікорського", (НТУУ «КПІ ім. І. Сікорського»), просп. Перемоги, 37, Київ, 03056, Україна, talko.yura@gmail.com

# МЕТОДИ СТИСНЕННЯ МОДЕЛЕЙ В ГЛИБИННОМУ НАВЧАННІ НА ОСНОВІ МЕТОДУ СТУДЕНТА-ВЧИТЕЛЯ

*Запропонований метод стиснення моделей на основі імітації навчання від декількох вчителів надає можливість зменшити кількість помилок у порівнянні зі звичайним підходом студента-вчителя.*

***Ключові слова:*** *нейромережі, модель, глибинне навчання, дистиляція знань, гаусів шум.*

## Вступ

Застосування глибоких нейромереж пов'язано з обробкою великих обсягів даних (*data set*) зовнішнього світу (зображення, відео, текстова та чисельна інформація), що при недостатній кількості обчислювального ресурсу призводить до неприйнятних витрат часу. Особливо критичною є нестача ресурсів у разі використання нейромереж в мобільних застосуваннях. Наприклад, в [1] наведено експериментальні дані для медичного мобільного застосування, що використовує глибоку нейромережу для прогнозування стану здоров'я, та підкреслюється необхідність оптимізації використання ресурсів. З появою методів стиснення інформації з'явилась можливість значно зменшувати витрати часу на обчислення глибоких мереж і, що надзвичайно важливо в сучасних умовах, застосувати нейромережі на мобільних та інших пристроях з обмеженими обчислювальними ресурсами.

Методи стиснення, що набули розвитку останнім часом, можна класифікувати так:

- методи обміну параметрами (*parameters haring methods* (*PHM*));
- методи обрізання мережі (*network pruning methods* (*NPM*);
- «темні знання»('*dark knowledge'*) (*DK*);
- методи навчання студента-вчителя (*student-teacher methods* (*STM*));
- методи декомпозиції матриць (*matrix decomposition methods* (*MDM*)).

В основному, всі методи стиснення зосереджені на зменшенні складності *глибоких* моделей. Проте після стиснення моделі потрібно виконувати обернену операцію, що потребує витрат часу та обчислювальних ресурсів. Одним з недостатньо досліджених методів, який може вирішити цю проблему, є метод навчання студента-вчителя (*student-teacher training*) для стиснення глибоких моделей [2]. Застосування цього методу передбачає, що неглибока мере-

жа (студент) навчається у глибокої мережі (вчителя). Глибока мережа досить швидко, проте з використанням значно більшої кількості ресурсів, досягає високої точності. Проблема неглибокої моделі полягає у низькій точності за рахунок економії ресурсів та зменшення складності обробки. Для досягнення точності, як у глибокої моделі-вчителя, їй потрібно набагато більше обчислень і, відповідно, часу, що є небажаним для більшості практичних задач. Методи стиснення використовують для підвищення точності мережі студента без збільшення її глибини, і, відповідно, кількості обчислювальних ресурсів. У даному дослідженні запропоновано метод, який надає можливість збільшити точність навчання мережі-студента.

## Методи стиснення моделі

Опишемо відомі методи стиснення моделей.

**Метод обміну параметрами *PHM*** передбачає використання простої хеш-функції для групування ваг (параметрів) у хеш-групи (*hash buckets*) [3]. При цьому кожна хеш-група відповідає одному параметру. У методі використовується $k$-вимірна кластеризація для повного квантування параметрів (розбивка діапазону їх значень на скінчену кількість інтервалів), пов'язаних між собою шарів моделі [4]. Метод дозволяє підвищити ступінь стиснення моделі в десятки разів (в одному з дослідів модель була стиснута в 24 рази) при втраті лише 1% точності, про що свідчать результати проведених експериментів. У даному методі використовується регуляризація замість прямого квантування параметрів зв'язаних шарів.

**Обрізання мережі *NPM*** полягає у відкиданні параметрів, вага яких нижче заданого порогового значення. Цей метод можна розширити використанням кодування Гаффмана для ще більшого скорочення кількості параметрів. Метод спрямований на скорочення кількості обчислень та ігнорування фільтрів, які мають найменший вплив на точність. Надлишкові нейрони можуть бути виявлені та відкинуті з використанням цього методу.

«**Темні знання**» *DK*. Ключовою ідеєю цієї групи методів, до якої належить метод навчання студента-вчителя (*student-teacher training*), є використання прихованих («темних») знань у складній, вже навченій нейромережі, для навчання більш простої нейромережі. Модифікація методу навчання студента-вчителя передбачає надання міток тренувальним даним без указування на мережу-вчителя, від якого вони були отримані. Ці мітки використовуються для тренування меншої моделі (студента). Однією з реалізацій такого способу є імітація логіт-змінних (*logit-values*) моделі вчителя. Проміжні результати прихованих шарів в цій моделі можуть використовуватись як цільові значення для моделі студента. Узагальнення методу шляхом введення температурної змінної $T$ в *softmax*-функцію призводить до пом'якшення значень цієї функції за умови збільшення значення змінної температури. Пом'якшені значення ймовірностей виявляють додаткову (приховану в звичайних значеннях) інформацію про ймовірності вихідних класів, ніж звичайні. Приховану інформацію називають «темні знання». Узагальнена *softmax*-функція визначає ймовірність $q^{(i)}$ за формулою [5]:

$$q^{(i)} = \frac{e^{z(i)/T}}{\sum_j e^{z(j)/T}},$$

де $T$ — температура, $z(i)$ — відповідний вивід логіт-функції (*logits*) попередньо навченого вчителя.

Зазвичай значення $T$ приймають рівним одиниці. Якщо використовується значення, більше за одиницю, то найбільше і найменше значення ймовірностей, отримані за формулою, менше відрізняються одне від одного, що називають пом'якшенням значень.

**Метод декомпозиції матриць *MDM*** [4] полягає у використанні декомпозиції для стиснення ваг в різних шарах мережі. Метод перетворює щільні матриці ваг до вигляду тензорної декомпозиції (*tensor decomposition*), що значно зменшує кількість вхідних параметрів.

Стиснення глибоких моделей (*compression of deep models*) має переваги у вирішенні трьох проблем: використання пам'яті, зменшення

витрат часу на тренування моделі, зменшення складності обчислень. Як методи обміну параметрами *PSM*, так і метод декомпозиції матриць *MDM*, зосереджені лише на зменшенні використання пам'яті глибокими моделями, але ці методи не зменшують витрати часу на тренування. Метод студента-вчителя *STM*, навпаки, зосереджений на зменшенні складності обчислень та часу на тренування.

## Навчання методом студента-вчителя

В рамках методу студента-вчителя *STM* в глибинному навчанні (*deep learning*), вчитель — це попередньо підготовлена та навчена глибинна модель, яка використовується для навчання іншої, зазвичай неглибокої, моделі, яку називають студентом. У використанні методу студента-вчителя є такі переваги:

- «темні знання», які присутні в результатах роботи моделі-вчителя, працюють як певні регуляризатори для моделі студента, оскільки вони забезпечують більш «м'який» набір знань, за яким легше відсіяти корисну інформацію;
- збіжність зазвичай швидша, ніж при використанні булевих міток, завдяки м'яким цілям що пришвидшують тренування;
- відносно невелика кількість даних для тренування моделі-студента.

Наведені переваги дозволяють використати метод шумового (*noise-based*) регуляризатора для моделі вчителя. Далі буде описано систему з одного вчителя та студента, яка є базою для наступних експериментів.

## Навчання моделі студента з використанням логістичної регресії

У [6] запропоновано метод навчання моделі студента з логарифмічною ймовірністю за змінною $z$, яка називається логіт-функцією (*logits*) і є результатом шару перед викликом функції м'якого максимуму (*softmax*). Мережа виконує навчання за допомогою регресії з використанням логіт-функції з навчальними даними, поданими у вигляді пар значень $\{(x^{(1)}, z^{(1)}),\ldots,(x^{(i)}, z^{(i)}),\ldots,(x^{(n)}, z^{(n)})\}$, де $x^{(i)}$— $i$-й навчальний рядок в піднаборі (*mini-batch*), $n$ – кількість пар значень в піднаборі даних. Вважаємо, що тут це слідує з контексту і змінювати не потрібно. $z^{(i)}$ — відповідний вивід логіт-функції попередньо навченого вчителя для $x^{(i)}$. Функція втрат (*loss function*) $L$ має вигляд:

$$L(x, z, \theta) = \frac{1}{2T} + \sum_i \left\| g(x^{(i)}, \theta) - z^{(i)} \right\|_2^2, \qquad (1)$$

де $T$ — значення, назване температурою; воно взято рівним розміру піднабору даних для однієї ітерації (*mini-batch*), $\theta$ — набір параметрів моделі студента, $g(x(i),\theta)$ — вивід учнівської моделі логіт-функції для $x^{(i)}$.

Далі дана ідея використовується для додання шумів (*noise-based*) у знання вчителя.

## Навчання моделі студента з використанням логіт-функції

Продуктивність неглибоких моделей в рамках методу студента-вчителя (*student-teacher training*) значно покращено за допомогою методів, запропонованих в [5]. Як себе поведе модель, якщо її навчає декілька вчителів? Аналогічно реальному світові, де студент може покращити швидкість і якість навчання з одного предмету, отримуючи знання про нього від декількох викладачів (з альтернативними думками, повторенням вже пройденого і т. ін.), можна припустити, що схожим чином поведе себе і модель студента в рамках методу студент-вчитель (*student-teacher method*). Але використання декількох вчителів має як переваги, так і недоліки, а саме збільшення точності та, водночас, збільшення часу навчання.

Для того, щоб зменшити вплив недоліків у підході навчання від декількох вчителів у даному дослідженні запропоновано замість використання кількох вчителів (що збільшує кількість вхідних даних і, відповідно, час навчання), використовувати симуляцію ефекту навчання у кількох вчителів шляхом введення «шумів» та «заплутувань» в початкові знання моделі вчителя. Заплутування не лише імі-

тують навчання від кількох вчителів, а також породжують шум в шарі втрат (*loss layer*), що створює ефект регуляризатора. Під регуляризатором розуміють певну зміну параметрів моделі, що має на меті не допустити стану перенавчання (*over fitting*), тобто такого стану, коли мережа зосереджується на наданих від вчителя прикладах і втрачає (або зменшує) можливість обробляти більш загальні «знання». Таким чином, новий зашумлений вчитель допомагає студентам краще навчатися та отримувати результати, більш близькі до того, якими вони є у вчителя, не втрачаючи при цьому можливості обробляти вхідні дані, відмінні від нього. Якщо припущення про використання шумів у знаннях мережі вчителя виявиться вірним, то запропонований метод імітації навчання від декількох вчителів зможе підвищити точність мережі-студента без значних затрат часу та машинних ресурсів.

Нехай число логітів у мережі вчителя задає значення вектору гаусівського шуму (*Gaussian noise*) $\varepsilon$ з нульовим середнім значенням і середнім квадратичним відхиленням $\sigma$. Якщо $z^{(i)}$ результат вихідного шару моделі вчителя для $x^{(i)}$, тоді $z^{(i)}$ змінюється так:

$$z^{(1)} = (1+\varepsilon), \qquad (2)$$

де **1** — одиничний вектор, $i \in R^n$, $n$ — кількість класів в навчальному наборі даних.

Більше значення квадратичного відхилення $\sigma$ означає більше збурення оригінальних значень логіт-функції вчителя $z^{(i)}$. Застосовувати збурення до всіх наданих наборів немає потреби. Замість цього потрібно вибрати лише деякі набори з заданою ймовірністю $\alpha$. Тоді значення логіту обраних наборів збурюються за допомогою рівняння (2). Функція втрат розраховується за формулою (1). Отримавши модель студента з початковими вагами $\theta_0$, знайдемо кінцеві параметри $\theta$, використовуючи метод стохастичного градієнта, де в $(t + 1)$ ітерації $\theta$ змінюється так:

$$\theta_{(t+1)} = \theta_t - \gamma_t \sum_{(x,y)\in D_t} \nabla_t \left[ L\left(x, z, \theta\right) \right], \qquad (3)$$

де $D_t$ — міні-вибірка, взята випадково з навчальної вибірки $D$; $\gamma_t$ — швидкість навчання, $L(x,z,\theta)$ — рівняння (3); $\nabla_t [L(x,z,\theta)]$ обчислюється з використанням методу зворотного поширення помилки (*gradient back propagation*).

Таким чином, набори відбираються з ймовірністю з міні-вибірки. Цільові значення логіт-функції збурюються за рівнянням (2). Функція втрат студентської мережі визначається рівнянням (1).

Відомо, що зашумлені дані допомагають регуляризувати модель [7]. Додавання регуляризації у функцію втрат $L$ еквівалентно додаванню гаусівського шуму у вхідні дані. Регуляризована функція втрат визначається так:

$$L(x,\theta,z) = L(x,\theta,z) + R(\theta),$$

де $x$ є гаусівським шумом, $L$ еквівалентна зашумленим вхідним даним $L(x,\theta,z)$, а $R(\theta)$ — регуляризатор $L$.

У вибраному методі збурюються цільові вихідні дані $z$ (зашумлені дані) замість вхідних даних $x$. Покажемо, що збурення цільових вихідних даних $z$ логіт-функції вчителя еквівалентно додаванню зашумленого (*noise-based*) регуляризатора до функції втрат. З рівняння (2) слідує:

$$\tilde{z}^{(1)} = (1+\varepsilon)\cdot z^{(1)} = z^{(1)} + \varepsilon \cdot z^{(1)}.$$

Тоді рівняння (1) функції втрат $L^0$ приймає вигляд:

$$L\left(x,\theta,\tilde{z}\right) = \frac{1}{2T}\left\| g\left(x^{(i)},\theta\right) - z^{(i)} - \varepsilon\cdot z^{(i)} \right\|_2^2 =$$

$$= L\left(x,\theta,z\right) + E_R,$$

де $E_R = \frac{1}{2T}\left\|\varepsilon\cdot z^{(i)}\right\|_2^2 + \frac{1}{T}\left\| z^{(i)} - g\left(x^{(i)},\theta\right)\right\|_2 \cdot \left\|\varepsilon\cdot z^{(i)}\right\|_2$ — новий регуляризатор, визначений на основі шуму $\varepsilon$.

Таким чином, збурення логітів у мережі вчителя еквівалентно додаванню зашумленого регуляризатора до функції втрат.

## Результати експериментів

Програмний продукт, якій реалізує запропонований метод навчання нейромережі, розроблено з використанням мови програмування *PHP*, фреймворку *Phalcon On PHP*, бази даних *MySql* та *Redis*.

Оцінка методу проведена на декількох наборах даних *MNIST* [8]. *MNIST* — популярний набір даних для тренування моделей розпізнавання рукописного вводу з 10 класами. Нав-

чальний набір містить 50000 зображень, з них підтверджених — 10000. Всі зображення мають розмір 28×28 та кольори, які є градаціями сірого кольору. Тренувальні дані попередньо не оброблені, що надає можливість виконувати власну попередню їх обробку.

Метод стохастичного градієнта використовується для навчання всіх мереж з розміром піднабору даних для однієї ітерації (*mini-batch*), рівному 64. Конвергенція (збіжність) була досягнута шляхом тестування на наборі підтверджених даних. Результати експериментів порівнюються з результатами і продуктивністю звичайного методу студент-вчитель, описаного в [6]. Отримані результати за всіма наборами даних показують ефективність вибраних методів. Найкращий результат було досягнуто на наборі даних *CIFAR*-10.

*Модель вчителя*. Як мережу вчителя використано модифіковану мережу *LeNet*, котра має два згорткові шари і повнозв'язаний шар з десятьма класифікаторами (конфігурація виглядатиме так [*C5* (*S1P0*) @ *20-MP2* (*S2*)]) .

*Мережа студент*. Як мережу студента використано невелику мережу з двома повнозв'язними шарами по 800 нейронів в кожному шарі. Архітектура може бути закодована як *FC*800-*FC*800-*FC*10.

*Результати*. Модель вчитель отримала 68 помилок в тестових наборах (з 10000 тестових зразків, частота помилки = 0,0068). Студентська мережа допустила 97 помилок (частота помилки = 0,0097) при базовому методі студент-вчитель (метод логіт-регресії). Так як різниця в продуктивності між мережею учителя та студента невисока, то ймовірність відбору вибірки $\alpha = 0{,}15$, тобто приблизно 15 відсотків піднаборів кожного набору параметрів відбираються для збурення. Збурення відбувалося при різних рівнях гаусівського шуму ($\mu = 0$, різними $\sigma$), як показано в таблиці. Цей шум доданий безпосередньо до ненормалізованих логітів у всіх експериментах дослідження. З наведених результатів слідує, що відбувається послідовне покращення роботи студента при застосуванні збурення до логіт. Як видно з таблиці, використання гаусівського шуму для збурення ненормалізованих логітів зменшує кількість помилок, а значить покращує роботу мережі студента.

***Таблиця*. Результати експериментів на *MNIST***

| Рівень шуму ($\sigma$) | Коефіцієнт помилок | Відсоток покращення, % |
|---|---|---|
| 0,10 | 0,0096 | 1,0 |
| 0,20 | 0,0093 | 4,1 |
| 0,30 | 0,0094 | 3,1 |
| 0,40 | 0,0087 | 10,3 |
| 0,50 | 0,0087 | 10,3 |
| 0,60 | 0,0090 | 7,2 |
| 0,70 | 0,0090 | 7,2 |
| 0,80 | 0,0086 | 11,3 |
| 0,90 | 0,0086 | 11,3 |
| 1,0 | 0,0087 | 10,3 |

## Висновки

Запропоновано метод шумового регуляризатора при навчанні моделі методами дистиляції знань та навчанні моделі студента від моделі вчителя, який імітує навчання від декількох вчителів, зменшуючи кількість помилок мережі студента. Результати експериментів свідчать, що при правильному підборі рівня шуму спостерігається зменшення кількості помилок до 11 відсотків. Наведені результати експериментів доводять, що застосування шумового регуляризатора збільшує точність мережі студента порівняно зі звичайним навчанням від вчителя.

## REFERENCES

1. B*enedetto J.I., Sanabria P., Neyem A., Navon J., Poellabauer C., Xia B.*, 2018. "Deep Neural Networks on Mobile Healthcare Applications: Practical Recommendations". Proceedings The 12th Int. Conf. on Ubiquitous Computing and Ambient Intelligence (UCAmI 2018), 2(19), pp. 1–12, https://doi.org/10.3390/proceedings2190550
2. *Wong J.H.M., Gales M.J.F.*, 2016. Sequence Student-Teacher Training of Deep Neural Networks. INTERSPEECH 2016, Sept. 8–12, San Francisco, USA, http://mi.eng.cam.ac.uk/~jhmw2/interspeech2016.paper.pdf
3. *Chen W., Wilson J.T., Tyree S., Weinberger K.Q., Chen Y.*, 2015. Compressing neural networks with the hashing trick, CoRR, https://arxiv.org/abs/1504.04788.
4. *Denil M., Shakibi B., Dinh L., N. de Freitas*, et al., 2013. "Predicting parameters in deep learning". Proc. of the 26th Int. Conf. on Neural Information Processing Systems, NIPS'13, 2, pp. 2148–2156.
5. *Hinton G., Vinyals O., Dean J.* Distilling the knowledge in a neural network. arXiv:1503.02531, 2015.
6. *Ba J., Caruana R.*, 2014. Do deep nets really need to be deep? In Advances in neural information processing systems. Part of: Advances in Neural Information Processing Systems 27 (NIPS 2014), https://papers.nips.cc/paper/5484-do-deep-nets-really-need-to-be-deep.
7. *Bishop C.M.*, 1995. "Training with noise is equivalent to tikhonov regularization". Neural computation, 7(1), pp. 108–116.
8. *Cun Y.L., Cortes, C, Burges, C.J.C.* The MNIST Database of handwritten digits, http://yann.lecun.com/exdb/mnist/

*I.V. Stetsenko*, Doctor of Technical Sciences, professor, of the Department of Computer-Aided Management and Data Processing Systems, National Technical University of Ukraine "Igor Sikorsky Kyiv Politechnic Institute", 03056, Kyiv, Peremohy Ave 37, Ukraine,
stiv.inna@gmail.com

*Yu.S. Talko*, Master of Information Systems and Technology, Student of the Department of Computer-Aided Management and Data Processing Systems, National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", 03056, Kyiv, Peremohy ave., 37, Ukraine,
talko.yura@gmail.com

## COMPRESSION METHODS OF DEEP LEARNING MODELS BASED ON STUDENT-TEACHER METHOD

**Introduction**. The use of deep neural networks is associated with the processing of large volumes of data (datasets) from the outside world (images, videos, huge data arrays like statistics), which, in case of limited computing resources leads to unacceptable time consuming. After the invent of compression methods, it has become possible to reduce significantly the time spent on calculating deep networks and, accordingly, it is possible to apply them on mobile or other devices with limited computing resources. The article presents a method of compression using a noise regulator and distillation of knowledge.

**Purpose**. The purpose of the article is to offer an effective way of compressing and learning the model through the modification of the distillation of knowledge method.

**Methods.** To provide greater accuracy and fewer errors in the model, a compression method is proposed based on the addition of a regularizer that implements the Gaussian noise to the teacher's knowledge in the teacher-student methods.

**Result.** The results of the experiments show that if the data and noise level is selected correctly, it is possible to reduce the number of errors to 11%. Consequently, the use of the proposed method leads to accelerated learning of the student model (due to the fact that the training as such has already been carried out earlier), and using the regularizer, the number of mistakes are done by the student network is reduced.

**Conclusion.** The compression method proposed is based on the simulation of training from several teachers, which allows reducing the number of errors compared to the usual approach of teacher-student (teacher-student methods).

***Keywords**: neural network, model, in-depth learning, distillation of knowledge, Gaussian noise.*

*И.В. Стеценко*, доктор. техн. наук, профессор. кафедра автомат. систем обработки информации и управления, Нац. техн. ун-т Украины «Киевский политехнический институт имени Игоря Сикорского» (НТУУ «КПИ им. И. Сикорского»), просп. Победы, 37, Киев, 03056, Украина, stiv.inna@gmail.com

*Ю.С. Талько*, студент, кафедра автомат. систем обработки информации и управления, Нац. техн. ун-т Украины «Киевский политехнический институт имени Игоря Сикорского» (НТУУ «КПИ им. И. Сикорского»), просп. Победы, 37, Киев, 03056, Украина, talko.yura@gmail.com

## МЕТОДЫ СЖАТИЯ МОДЕЛЕЙ В ГЛУБИННОМ ОБУЧЕНИИ НА ОСНОВЕ МЕТОДА СТУДЕНТА-УЧИТЕЛЯ

**Введение**. Применение глубоких нейросетей связано с обработкой больших объемов данных внешнего мира (*data set*) (изображения, видео, огромные массивы статистических данных), что при недостаточном количестве вычислительных ресурсов приводит к неприемлемым затратам времени. С появлением методов сжатия появилась возможность значительно сократить затраты времени, используя для вычислений глубокие сети, и, соответственно, появилась возможность применять их на мобильных или других устройствах с ограниченными вычислительными ресурсами. В статье приведен метод сжатия с использованием шумового регуляризатора и дистилляции знаний.

**Цель статьи** — предложить эффективный способ сжатия и обучения модели путем видоизменения способа дистилляции знаний.

**Методи.** Для обеспечения большей точности и меньшего количества ошибок в модели предложен метод сжатия на основе введения регуляризатора, который добавляет гауссовский шум к знаниям учителя в методе студента-учителя (*student-teacher training*).

**Результат.** Результаты экспериментов свидетельствуют, что при правильном подборе набора данных и уровня шума можно получить уменьшение количества ошибок до 11 процентов. Таким образом, использование предложенного метода привело к ускорению обучения модели студента (за счет того, что обучение, как таковое, уже было проведено ранее). А с помощью регуляризатора уменьшено количество ошибок, которые допускает сеть студента.

**Вывод.** Предложенный метод сжатия моделей на основе имитации обучения от нескольких учителей предоставляет возможность уменьшить количество ошибок в сравнении с обычным подходом студента-учителя (*student-teacher methods*).

***Ключевые слова***: *нейросети, модель, глубинное обучение, дистилляция знаний, гауссовский шум.*