УДК 681.513.7

А.М. ЧЕРНОДУБ[*], Д.В. НОВИЦЬКИЙ[*]

МЕТОД СЕМПЛЮВАННЯ ДЛЯ БОРОТЬБИ З ЕФЕКТОМ ЗНИКНЕННЯ ГРАДІЄНТІВ У РЕКУРЕНТНИХ НЕЙРОМЕРЕЖАХ

[*]Інститут проблем математичних машин і систем НАН України, Київ, Україна

Анотація. *Ефект зникнення градієнтів є спільною проблемою навчання рекурентних і глибоких нейромереж. У статті розроблено метод для оцінки внеску кожного прикладу з навчальної вибірки у градієнт цільової функції навчання. Запропоновано новий універсальний метод, який дозволяє утримувати норму градієнтів у задовільних межах. Для експериментальної перевірки нашого підходу використано спеціальні синтетичні бенчмарки для тестування нейромереж на здатність виявляти довготривалі залежності. Навчена з використанням даного методу рекурентна нейромережа з одиничними затримками може знаходити залежності між подіями в часових послідовностях довжиною до 100 і більше тактів.*
Ключові слова: *ефект зникнення градієнтів, рекурентна нейронна мережа, регуряризація градієнтів.*

Аннотация. *Эффект исчезновения градиентов является общей проблемой обучения рекуррентных и глубоких нейросетей. В статье разработан метод для оценки вклада каждого обучающего примера из выборки в градиент целевой функции обучения. Предложен новый универсальный метод, который позволяет удерживать норму градиентов в приемлемых пределах. Для экспериментальной проверки нашего подхода использованы специальные синтетические бенчмарки для тестирования нейросетей на способность выявлять долговременные зависимости. Рекуррентная нейросеть с единичной линией задержек, обученная с использованием данного метода, может находить зависимости между событиями во временных последовательностях длиной до 100 и более тактов.*
Ключевые слова: *эффект исчезновения градиентов, рекуррентная нейронная сеть, регуляризация градиентов.*

Abstract. *Vanishing gradients effect is a common problem for recurrent and deep neural networks. In this paper we construct a method to estimate a contribution of each training example to the norm of the long-term components of the target functions gradient. We propose a novel universal technique that makes the norm of the gradient stay in the suitable range. To check our framework experimentally we use a special synthetic benchmarks for testing RNNs on ability to capture long-term dependencies. Our recurrent network can detect links between events in the (temporal) sequence at the range 100 and longer.*
Keywords: *vanishing gradients effect, recurrent neural network, gradient regularization.*

1. Вступ

Ефект зникнення градієнтів є найбільш серйозною перешкодою для ефективного навчання глибоких і рекурентних персептронно-подібних нейромереж. Якщо для глибоких нейромереж було винайдено методи перенавчання з використанням обмежених машин Больцмана [1] та автоенкодерів [2], для рекурентних нейромереж найбільш поширеним засобом боротьби з ефектом зникнення градієнта є модифікація архітектури нейромережі для отримання більш сприятливої динаміки для зворотного потоку похідних. Першим прикладом можна назвати нейромережі NARX, що під час виконання процедури «зворотного поширення в часі» (Backpropagation Through Time, BPTT), завдяки лінії затримок, мають прямі зв'язки між вихідною похибкою і «старими» шарами, розгорнутими у часі [3]. В цьому випадку градієнт, що поширюється назад за правилом BPTT, одразу потрапляє на віддалені у часі «старі» шари нейронів у часі, не просуваючись через велику кількість проміжних шарів.

ISSN 1028-9763. Математичні машини і системи, 2016, № 2

Іншим прикладом є резервуарні нейромережі (Echo State Networks, ESN) [4], що можуть розглядатися як великі резервуари розріджено пов'язаних нейронів з довільно ініціалізованими ваговими матрицями, які виробляють хаотичну динаміку. Для резервуарних нейромереж градієнти похибки розраховуються лише для останнього нерекурентного шару нейронів і цього може бути достатньо для навчання довгостроковій динаміці [5]. В той же самий час резервуарні нейромережі для багатьох задач мають зайву кількість вільних параметрів. Іноді їх критикують як неефективний засіб для використання для задач нейроуправління [6]. Інший підхід, який було спеціально розроблено для навчання довгостроковим залежностям, є нейромережі довго-короткотривалої пам'яті (Long-Short Term Memory, LSTM) [7]. Ці нейромережі мають складну внутрішню структуру з елементів, що називаються «клітини» ("cells") і включають в себе вхідні гейти. Гейти для забування можуть адаптивно стирати або поновлювати внутрішню пам'ять. Внутрішні рекурентні зв'язки мають константні значення градієнтів зворотного поширення. Можливо, це найбільш популярна сім'я моделей рекурентних нейромереж для багатьох практичних задач, що демонструє state-of-the-art показники якості для задач розпізнавання мови [8], генерування текстів для підпису зображень [9] та нейронного машинного перекладу [10]. Ідея використання вхідних гейтів та гейтів для забування надихнула багатьох дослідників; нейромережі GRU (Gated Recurrent Units), можливо, є найбільш успішним нащадком LSTM [11]. Нарешті, нещодавно спільна команда Google та Facebook здійснила масштабний експеримент з пошуку найбільш вдалої архітектури для рекурентних нейромереж [12]. Вони виконали масштабний чисельний експеримент і перебрали 10 000 довільних LSTM-подібних архітектур з 230 000 гіперпараметрів і отримали кілька нових моделей рекурентних нейромереж, а також рекомендації для поліпшення нейромереж LSTM. Існують і інші підходи для класифікації послідовностей [13].

Разом з тим, питання, як навчати прості рекурентні нейромережі SRN для виявлення довготривалих залежностей, все ще залишається актуальним науковим питанням. Це дуже важливо, як мінімум, для кращого розуміння процесів, що протікають під час навчання глибоких та рекурентних нейронних мереж. Також, нейромережі SRN – найбільш компактні і швидкі рекурентні нейромережі, що може бути важливим для їх реалізації на мобільних та вбудованих приладах.

Останні дослідження показують спроможність навчати нейромережі SRN виявленню багатокрокових залежностей з використанням кількох технік [14-17]. Вони включають в себе вивірену ініціалізацію початкових вагових коефіцієнтів нейромережі, перемасштабування великих градієнтів, використання більш ефективних методів навчання 1-го порядку, таких, як моменту Нестерова. Наш метод засновано на ідеї штучного підсилення або зменшення «старих» градієнтів для розгорнутих назад у часі нейромереж.

2. Динаміка зворотного поширення похибки в рекурентних нейромережах типу Simple Recurrent Network

На кожному кроці часу k нейромережа SRN отримує вектор зовнішнього входу $\mathbf{u}(k)$, вектор попереднього стану $\mathbf{z}(k-1)$ і виробляє вихідний вектор $\tilde{\mathbf{y}}(k+1)$:

$$\mathbf{a}(k) = \mathbf{u}(k)\mathbf{w}_{in} + \mathbf{z}(k-1)\mathbf{w}_{rec} + \mathbf{b}, \tag{1}$$

$$\mathbf{z}(k) = f(\mathbf{a}(k)), \tag{2}$$

$$\tilde{\mathbf{y}}(k+1) = g(\mathbf{z}(k)\mathbf{w}_{out}), \tag{3}$$

де $\mathbf{w}_{in}$ – матриця вхідних вагових коефіцієнтів, $\mathbf{w}_{rec}$ – матриця рекурентних зв'язків, $\mathbf{w}_{out}$ – матриця вихідних зв'язків, $\mathbf{a}(k)$ – вектор пресинаптичних активацій, $\mathbf{z}(k)$ – вектор пост-

синаптичних активацій (або станів нейромережі), $f(\cdot)$ і $g(\cdot)$ – нелінійні активаційні функції для прихованого та вихідного шару нейронів відповідно. Для нейромереж SRN використовуються активаційні функції гіперболічного тангенса для прихованого шару і функції *softmax* або лінійна функція в залежності від цільової задачі (класифікація або регресія).

Після обробки вхідної послідовності $\mathbf{u}(1),\mathbf{u}(2),...,\mathbf{u}(k)$ і виробки вихідного вектора $\tilde{\mathbf{y}}(k+1)$ обчислюється похибка $E(k+1)$ і розраховуються динамічні похідні функції похибки по вагових коефіцієнтах нейромережі $\frac{\partial E}{\partial \mathbf{w}}$ методом BPTT. Після застосування зворотного поширення похибки обчислюються миттєві похідні $\frac{\partial E}{\partial \mathbf{w}(n)}$ і динамічні похідні $\frac{\partial E_{BPTT}}{\partial \mathbf{w}}$ як сума миттєвих похідних:

$$\frac{\partial E_{BPTT}}{\partial \mathbf{w}} = \sum_{n=1}^{h} \frac{\partial E}{\partial \mathbf{w}(k-n)}, \tag{4}$$

де $n = 1,...,h$, а h – порядок усікання (truncation depth) BPTT.

Похідні для n-го шару у часі розгорнутої нейромережі SRN приймають вигляд $\frac{\partial E}{\partial \mathbf{w}_{in}(n)} = \mathbf{u}(n)^T \boldsymbol{\delta}(n)$, $\frac{\partial E}{\partial \mathbf{w}_{rec}(n)} = \mathbf{z}(n-1)^T \boldsymbol{\delta}(n)$, де $\boldsymbol{\delta} \equiv \frac{\partial E}{\partial \mathbf{a}}$ – локальні градієнти. Для останнього шару розгорнутої в часі нейромережі локальні градієнти $\boldsymbol{\delta}$ – це нев'язка похибки $\mathbf{e}(k) = \mathbf{y}(k+1) - \tilde{\mathbf{y}}(k+1)$, для наступних шарів відповідні локальні градієнти обчислюються згідно з відомою формулою зворотного поширення похибки:

$$\delta_j(n-1) = f'(a_j(n-1)) \sum_i w_{rec}^{ij} \delta_i(n). \tag{5}$$

Запишемо останнє рівняння в матричній формі:

$$\boldsymbol{\delta}(n-1) = \boldsymbol{\delta}(n) \mathbf{w}_{rec}^T diag(f'(\mathbf{a}(n-1))), \tag{6}$$

де $diag$ конвертує вектор у діагональну матрицю. Рівняння (6) може бути переписане з використанням матриці якобіана $\mathbf{J}(n) = \frac{\partial \mathbf{z}(n)}{\partial \mathbf{z}(n-1)}$:

$$\boldsymbol{\delta}(n-1) = \boldsymbol{\delta}(n) \mathbf{J}(n), \tag{7}$$

де

$$\mathbf{J}(n) = \mathbf{w}_{rec}^T diag(f'(\mathbf{a}(n-1))). \tag{8}$$

3. Причини ефекту зникнення градієнтів

Тепер можна пояснити ефект вибуху/зникнення градієнтів, що досліджувався у класичних [18, 19] і нових роботах [15, 17]. Як випливає з (7), норма зворотно-поширених локальних градієнтів строго залежить від норми якобіанів, тому що локальні градієнти є добутком якобіанів:

$$\boldsymbol{\delta}(n-2) = \boldsymbol{\delta}(n) \mathbf{J}(n) \mathbf{J}(n-1), \tag{9}$$

$$\dots$$

$$\boldsymbol{\delta}(n-h) = \boldsymbol{\delta}(n) \mathbf{J}(n) \mathbf{J}(n-1) ... \mathbf{J}(n-h+1). \tag{10}$$

Чим "старіші" локальні градієнти, тим більше матриць якобіанів було помножено для їх обчислення. Якщо норма якобіанів більше 1, локальні градієнти будуть експоненційно зростати, відбувається ефект "вибуху градієнтів". Це відповідає ситуації поведінки рекурентних нейромереж, коли довготривалі компоненти є більш важливими, ніж короткотривалі. І навпаки, якщо норма якобіанів менше 1, це приводить до ефекту зникнення градієнтів і "забування" довготривалих випадків.

Оскільки якобіани (8) містять множення однієї і тієї самої матриці вагових коефіцієнтів $\mathbf{w}_{rec}^T$, це в більшості випадків приводить до експоненційного збільшення або зменшення їх норм. Це подібно до ситуації, коли ми перемножуємо дійсні числа кілька раз на самих себе, тобто, $a^n \to \infty$, якщо $a > 1$, і $a^n \to 0$, якщо $a < 1$ при $n \to \infty$.

На практиці норми якобіанів при навчанні частіше бувають менше 1, тому що норми двох множників у (8) часто мають тенденцію бути менше 1. Для першого множника, звичайно, $\left\| \mathbf{w}_{rec}^T \right\| < 1$, оскільки великі норми приводять до неробастної поведінки нейромереж. Це мотивує використання ініціалізації вагових коеффіцієнтів матриць малими числами, що приводить до малих норм початкових матриць вагових коефіцієнтв.

Також можна згадати методи регуляризації для нейромереж, що запобігають збільшенню норми матриць вагових коефіцієнтів під час навчання. Що стосується другого множника в (8), то в випадку використання норми L_2 це дорівнює модулю найбільшого власного значення матриці; в нашому випадку використовується діагональна матриця, заповнена дійсними значеннями, тому дана норма є найбільшим елементом вектора похідних постсинаптичних значень $f'(\mathbf{a}(n-1))$. Максимальне значення похідних для функції гіперболічного тангенса 1, для сигмоїди $1/4$. Тому $\left\| diag(f'(\mathbf{a}(n-1))) \right\| \le 1$ для гіперболічного тангенса і $\left\| diag(f'(\mathbf{a}(n-1))) \right\| < 1$ гіперболічного тангенса. В той самий час, навіть якщо обидва множника в (8) мають норму 1, це, очевидно, не гарантує одиничну норму $\mathbf{J}(n)$.

4. Метод регуляризації градієнтів (псевдорегуляризації)

Одним із найбільш загальних методів для уникнення ефекту зникнення градієнтів, запропонований Р. Паскану, Т. Міколов і Д. Бенджио, є метод регуляризації градієнта (gradient regularization) [17, 20]. Дуже подібний метод одночасно незалежно був запропонований автором у роботах [21–23] і названий псевдорегуляризація. Але для зручності ми будемо далі використовувати більш поширений термін регуляризація градієнта.

Ідея підходу полягає в керуванні потоком зворотного поширення похибки під час навчання. В цьому випадку нейромережа вчиться не лише підлаштовуватись під дані навчальної вибірки, але й тримати норму зворотного потоку градієнтів у певних межах. Це здійснюється шляхом модифікації цільової функції навчання $L(\mathbf{w})$ для виконання багатокритеріальної оптимізації шляхом додавання додаткового члена $\Omega(\mathbf{w})$, який відповідає за величину зворотно-поширених градієнтів:

$$L(\mathbf{w}) = E(\mathbf{w}) + \lambda\Omega(\mathbf{w}), \tag{11}$$

де $E(\mathbf{w})$ – цільова функція навчання, що мінімізує похибку навчання (наприклад, регресії або класифікації), $\Omega(\mathbf{w})$ – регуляризатор, що запобігає надмірному зменшенню градієнтів, λ – коефіцієнт, що регулює вклад регуляризації градієнтів у сумарну похибку. В роботах автора [21–23] було запропоновано такий регуляризатор:

$$\Omega(\mathbf{w}) = \sum_k \left(\left\langle 1 - \left\| \boldsymbol{\delta}(\mathbf{w}, k) \right\|_{FRO} \right\rangle \right)^2. \tag{12}$$

Його метою є підтримка середньої норми локальних градієнтів близько 1. Для виконання даної мети похідні $\frac{\partial \Omega(\mathbf{w})}{\partial \mathbf{w}}$ були виведені і використані у градієнтному оптимізаційному алгоритмі. Цього було достатньо для навчання нейроконтролера для випадку нейроуправління [22], але в більш загальному випадку, для навчання рекурентних нейромереж, цей підхід показав незадовільну чутливість і взагалі дуже нестабільну поведінку під час навчання.

В [17] було використано такий регуляризатор:

$$\Omega(\mathbf{w}) = \sum_k \left(\frac{\left\| \frac{\partial E}{\partial \mathbf{z}(k+1)} \frac{\partial \mathbf{z}(k+1)}{\partial \mathbf{z}(k)} \right\|}{\frac{\partial E}{\partial \mathbf{z}(k+1)}} - 1 \right)^2. \tag{13}$$

Він примушує якобіани $\frac{\partial \mathbf{z}(k+1)}{\partial \mathbf{z}(k)}$ зберігати норму у напрямку, релевантному напрямку похибки навчання $\frac{\partial E}{\partial \mathbf{z}(k)}$, а не в довільному напрямку, як у згаданих вище роботах автора. Також такі обмеження є більш м'якими, оскільки вони спрямовані лише на «не зменшення» норми попередніх градієнтів. У той самий час аналітичні похідні $\frac{\partial \Omega(\mathbf{w})}{\partial \mathbf{w}}$ для (15) є дуже громіздкими; їх аналітичне обчислення є важкою задачею. Автори цього підходу використовували середовище Theano, що має вбудовану систему обробки символьних формул для автоматичного обчислення похідних. Але така функціональність не є загальнопоширеною, що ускладнює використання даного методу.

5. Диференціювання норми градієнта

Припустимо, мається міні-пакет навчальних даних $d = \{\mathbf{u}_1; \mathbf{t}_1; ...; \mathbf{u}_N, \mathbf{t}_{N_D}\}$, що містить N_D прикладів. Виконуються прямий і зворотний проходи нейромережі з використанням цього міні-пакета, і обчислюються похідні $d\mathbf{w} = \frac{\partial E}{\partial \mathbf{w}}$. Тепер потрібно перевірити вплив вектора $d\mathbf{w}$ на вагові коефіцієнти нейромережі $\mathbf{w}$: чи приводить застосування $d\mathbf{w}$ до мінімізації чи максимізації норми зворотно-поширених градієнтів. Ми зацікавлені в контролі норми локальних градієнтів $\left\| \boldsymbol{\delta}(\cdot) \right\|$, тому що норми похідних $\frac{\partial E}{\partial \mathbf{w}}$ і знаходяться у прямій залежності від норми локальних градієнтів.

Теорема. Нехай $\partial \mathbf{w}_{rec}$ – матриця вагових коефіцієнтів рекурентних зв'язків нейромережі SRN. Припустимо також, що вже виконано прямий і зворотний проходи і обчислено матрицю оновлення $d\mathbf{w}_{rec}$ для матриці рекурентних зв'язків, $\mathbf{w}_{rec}^{(i+1)} = \mathbf{w}_{rec}^{(i)} + d\mathbf{w}_{rec}$. Достатньою умовою для збільшення $\left\| \boldsymbol{\delta}(k-h) \right\|_{Fro}$ (норми локальних градієнтів, зворотно-поширених на h кроків назад у часі від поточного моменту k) є $dS > 0$, де

$$dS = (\mathbf{G}, d\mathbf{G}), \tag{14}$$

$$\mathbf{G} = \left(\prod_{i=h}^{1} diag(f'(\mathbf{a}(k-i+1)))\mathbf{w}_{rec} \right) \boldsymbol{\delta}(k), \tag{15}$$

$$d\mathbf{G} = \sum_{i=1}^{h} \left(\left(\prod_{j=h}^{1} diag\left[f'(\mathbf{a}(k-j+1)) \right] \mathbf{v} \right) \boldsymbol{\delta}(k) \right), \tag{16}$$
$$\mathbf{v} = d\mathbf{w}_{rec}, \ if\ i = j; \ \mathbf{v} = \mathbf{w}_{rec}, \ if\ i \neq j.$$

Аналогічно достатньою умовою для зменшення норми локальних градієнтів $\left\| \boldsymbol{\delta}(k-h) \right\|_{Fro}$ є $dS < 0$.

Proof. Розглянемо функцію $S'(\mathbf{w}_{rec})$, яка є нормою Фробеніуса (Евкліда) локальних градієнтів (8):

$$S'(\mathbf{w}_{rec}) = \left\| \boldsymbol{\delta}(k-h, \mathbf{w}_{rec}) \right\|_{Fro}. \tag{17}$$

Використовуючи (7), (8) і (10), отримаємо

$$\boldsymbol{\delta}(k-h) = \boldsymbol{\delta}(k)\mathbf{w}_{rec}^{T} diag(f'(\mathbf{a}(k-1)))...\mathbf{w}_{rec}^{T} diag(f'(\mathbf{a}(k-h+1))). \tag{18}$$

Для зручності позначимо $D_n \equiv diag(f'(\mathbf{a}(n)))$ і змінимо індекси кроків у часі таким чином: $(k-h)$-й крок – це 1-й крок, $(k-h+1)$-й крок – це 2-й крок і так далі, k-й крок – це H-й крок. Таким чином, рівняння (18) стає

$$\boldsymbol{\delta}(1) = \boldsymbol{\delta}(H)\mathbf{w}_{rec}^{T}\mathbf{D}_{H-1}\mathbf{w}_{rec}^{T}\mathbf{D}_{H-2}...\mathbf{w}_{rec}^{T}\mathbf{D}_{1}. \tag{19}$$

Оскільки диференціювання норми у квадраті більш зручне, ніж диференціювання норми $\left\| \mathbf{A} \right\|_{Fro} = \left\| \mathbf{A}^{T} \right\|_{Fro}$, $(\mathbf{A} \times \mathbf{B})^{T} = \mathbf{B}^{T}\mathbf{A}^{T}$, можна запропонувати нову службову змінну $\mathbf{G}$:

$$\mathbf{G} = \mathbf{D}_{1}\mathbf{w}_{rec}\mathbf{D}_{2}\mathbf{w}_{rec}...\mathbf{D}_{H}\mathbf{w}_{rec}\boldsymbol{\delta}(H). \tag{20}$$

Таким чином, виникає зацікавленість у дослідженні поведінки функції $S(\mathbf{w}_{rec})$ в околі поточної точки, де

$$S(\mathbf{w}_{rec}) = \left\| \mathbf{G} \right\|_{Fro}^{2}. \tag{21}$$

Для того, щоб проаналізувати поведінку S, обчислимо диференціал dS. Знак dS визначає, збільшується S або зменшується; модуль dS визначає швидкість зміни. Отже, диференціал dS становить:

$$dS = 2(\mathbf{G}, d\mathbf{G}), \tag{22}$$

де $\mathbf{G}$ визначається (15), а $d\mathbf{G}$ обчислюється як

$$d\mathbf{G} = \sum_{i=1}^{H} \boldsymbol{\delta}(k) \times \mathbf{D}_{i} \times \mathbf{w}_{rec} \times ... \times \mathbf{D}_{i} \times d\mathbf{w}_{rec} \times ... \times \mathbf{D}_{1} \times \mathbf{w}_{rec} \tag{23}$$

або в іншій формі:

$$d\mathbf{G} = \sum_{i=1}^{H} \left(\left(\prod_{j=1}^{H} \mathbf{D}_{j}\mathbf{v} \right) \boldsymbol{\delta}(k) \right), \tag{24}$$
$$\mathbf{v} = d\mathbf{w}_{rec}, \ if\ i = j; \ \mathbf{v} = \mathbf{w}_{rec}, \ if\ i \neq j.$$

Теорему доведено.

6. Метод семплювання для регуляризації градієнтів

Відомо, що навчання рекурентних нейромереж взагалі є достатньо нестабільною задачею. Модифікація цільової функції для збереження норми градієнтів призводить до додаткових труднощів навчання навіть, якщо вектори градієнтів модифікуються в релевантному напрямку $\frac{\partial E}{\partial \mathbf{z}(k)}$. Навчання нейромереж SRN з модифікованою цільовою функцією (13) на послідовностях, що містять лише короткострокову інформацію, приводить до гіршої якості навчання, ніж використання стандартної цільової функції навчання. Складність навчання рекурентних нейромереж пов'язана з більш складною поверхнею похибок порівняно з нейромережами прямого поширення; додавання додаткової мети оптимізації ускладнює поверхню похибок і погіршує процес навчання. Такі модифіковані градієнти, що не зовсім відповідають цільовій похибці навчання, спричиняють у рекурентних нейромережах так званий "ефект метелика", де малі перетурбації в даних або вагах нейромережі на початку обробки послідовності призводять до великих відхилень траєкторії в кінці послідовності. Це, зокрема, є головною причиною, чому популярні для глибоких нейромереж методи регуляризації на кшталт "dropout" погано працюють для рекурентних нейромереж [25]. З іншого боку, хоча вдала ініціалізація вагів нейромереж дуже важлива, це не гарантує збігання навчального процесу через причини, згадані вище. Можна сказати, що це є зворотною стороною різноманіття можливостей рекурентних нейромереж моделювати динаміку. Таким чином, нежорстке інтелектуальне керування нормою зворотно-поширених градієнтів все ще є інтригуючою проблемою для успішного навчання рекурентних нейромереж довготривалим залежностям.

Ідея запропонованого методу семплювання для регуляризації градієнтів полягає в використанні лише "правильних" прикладів даних для навчання. Використовуючи аналітичну умову, що була виведена в параграфі 1, тепер можливо точно оцінити вплив кожного прикладу або міні-пакета даних навчальної вибірки на норму зворотно-поширених градієнтів і використовувати цю інформацію для керування нормою зворотного потоку градієнтів.

У цій роботі вирішено використати найпростіший і найбільш очевидний спосіб: ми оцінюємо норму градієнтів; якщо норма стає дуже мала, ми ігноруємо у процесі навчання міні-пакети, які зменшують норму, тобто для яких $dS < 0$. Аналогічно, якщо норма стає дуже великою, ми ігноруємо міні-пакети, що збільшують цю норму, тобто для яких $dS > 0$. Важливим практичним зауваженням є рекомендація не використовувати для навчання ті міні-пакети, для яких абсолютні значення dS приймають великі величини, оскільки в такому випадку відбуваються великі стрибки норм градієнтів і подальша осциляція. Це викликано природою такого алгоритму оптимізації, який є подібним до найпростішої реалізації методу найскорішого спуску, який, як відомо, задовільно працює лише за умови невеликих значень компонентів векторів корекції вільних параметрів.

Для оцінки міри падіння градієнтів вводиться нова змінна, що носить назву Q-factor, яка дорівнює відношенню норми локальних градієнтів на початку процедури BPTT до відношення норми локальних градієнтів у кінці BPTT: $Q(\boldsymbol{\delta}, h) = \log_{10}\left(\frac{\|\boldsymbol{\delta}(k)\|}{\|\boldsymbol{\delta}(k-h)\|}\right)$ або

$$Q(\boldsymbol{\delta}, h) = \log_{10}(\|\boldsymbol{\delta}(k)\|) - \log_{10}(\|\boldsymbol{\delta}(k-h)\|), \tag{25}$$

де h – горизонт зворотного поширення похибки BPTT. Для ідеального навчання довготривалим залежностям Q-factor має бути близьким до 0. На практиці, для залежностей ~ 100 кроків, якщо Q-factor лежить на відрізку $[-1;1]$, що відповідає збільшенню або зменшенню

локальних градієнтів під час зворотного розповсюдження похибки максимум в 10 разів, то цього достатньо для навчання рекурентної нейромережі методом SGD.

Опис алгоритму

Припустимо, що маємо вибірку $\{\mathbf{U},\mathbf{T}\}$ для навчання нейромережі. Задамо апріорі допустимі межі падіння або зростання градієнтів, так звану «безпечну зону», визначивши діапазон $[Q_{MIN};Q_{MAX}]$.

Для кожного міні-пакета $\mathbf{d}_i=\{\mathbf{u}_i;\mathbf{t}_i\}$, що містить N_{MD} прикладів з навчальної вибірки $\{\mathbf{U},\mathbf{T}\}$:

– обчислити диференціал dS; якщо $abs(dS)>1$, ігнорувати цей міні-пакет;

– виконати прямий і зворотний проходи нейромережі, обчислити Q-factor $Q(\boldsymbol{\delta},h)$ за формулою (25);

– якщо $Q(\boldsymbol{\delta})\in[Q_{MIN};Q_{MAX}]$, вважати падіння (зростання) градієнтів задовільним; використати наявний міні-пакет $\mathbf{d}_i$ для отримання похідних $\frac{\partial E}{\partial \mathbf{w}}$ і навчання нейромережі;

– інакше, якщо 1) $Q(\boldsymbol{\delta},h)<Q_{MIN}$ і $dS>0$ або 2) $Q(\boldsymbol{\delta},h)>Q_{MAX}$ і $dS<0$ – використати наявний міні-пакет для навчання, інакше – ігнорувати цей міні-пакет

Даний алгоритм можна образно порівняти з процесом управління вітрильним судном: поки судно знаходиться в безпечній зоні, воно пливе туди, куди дме вітер. Якщо ж воно потрапляє в небезпечну зону, то вітрила піднімаються лише в тому випадку, якщо вітер дме в напрямку виходу з небезпечної зони.

7. Задачі для виявлення здатності рекурентних нейромереж виявляти довготривалі залежності

У відомій роботі [20] було запропоновано набір синтетичних бенчмарків для тестування здатності нейромереж навчанню довгостроковим залежностям. Цей набір став фактичним стандартом серед дослідників, що розробляють алгоритми навчання рекурентних нейромереж [5, 18, 27]. Точні умови формулювання задач дещо відрізняються в різних роботах, але в даній роботі використовується визначення з [17].

*Задача «Додавання» («Addition problem»). З*а умовою задачі, нейромережа послідовно отримує на вхід вектори, що мають дві компоненти (рис. 1). Перша компонента («Шум») містить випадкові значення, а друга компонента («Маркер») – нулі, за виключенням двох випадково визначених у часі моментів, коли ця компонента приймає значення “1”.

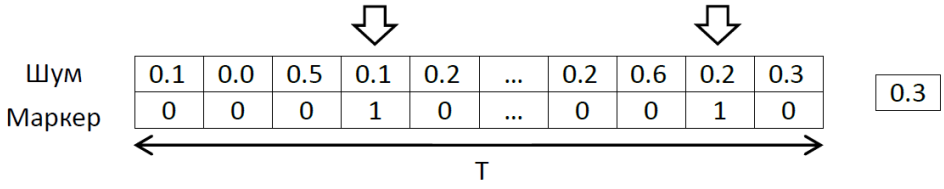


Рис. 1. Задача «Додавання»

Метою навчання нейромережі є апроксимація суми тих двох шумових сигналів, які відповідають маркерам “1” після обробки всієї послідовності. Результат додавання вважається коректним, якщо похибка менше 0,04. Чим більша довжина послідовності T, тим

важча задача, оскільки залежність між ключовими моментами стає більш довготривалою. Для кожної послідовності семплюється величина T' з відрізку $\left[T, \frac{11}{10}T\right]$. Перша позиція маркера є довільне ціле число з відрізку $\left[1, \frac{T'}{10}\right]$, а друга позиція маркера обирається з відрізку $\left[\frac{T'}{10}+1, \frac{T'}{2}\right]$. Довільні значення належать рівномірному розподіленню в межах $[0,1]$. Метою навчання нейромережі є оцінка значення суми позначених маркерами значень послідовності, поділена на 2.

Задача «Множення» («Multiplication problem»). Задача ідентична задачі «Додавання», але метою навчання є результат перемноження, а не додавання.

Задача «Порядок у часі» («Temporal order problem»). Нехай маємо цільовий словник з двох символів $\{A, B\}$ та шумовий словник з 4-х символів $\{c, d, e, f\}$. Для кодування символів у послідовності тут і далі використовується бінарне кодування, також відоме як "one-hot-encoding". Послідовність символів рівномірно семплюється зі словника шумових символів, за винятком двох випадкових позицій, коли символи випадково вибираються з цільового словника. Перша позиція обирається довільно з відрізку $\left[\frac{T}{10}, \frac{2T}{10}\right]$, а друга – з відрізку $\left[\frac{4T}{10}, \frac{5T}{10}\right]$. Метою навчання є визначення порядку, в якому цільові символи містяться в послідовності, тобто, один з 4-х варіантів: $\{AA, AB, BA, BB\}$.

Задача «Порядок у часі 3-bit» («3-bit temporal order problem»). Задача ідентична задачі «Порядок у часі», але ми маємо три випадкові позиції: перша семплюється на відрізку $\left[\frac{T}{10}, \frac{2T}{10}\right]$, друга – на відрізку $\left[\frac{3T}{10}, \frac{4T}{10}\right]$ і третя – на відрізку $\left[\frac{6T}{10}, \frac{7T}{10}\right]$.

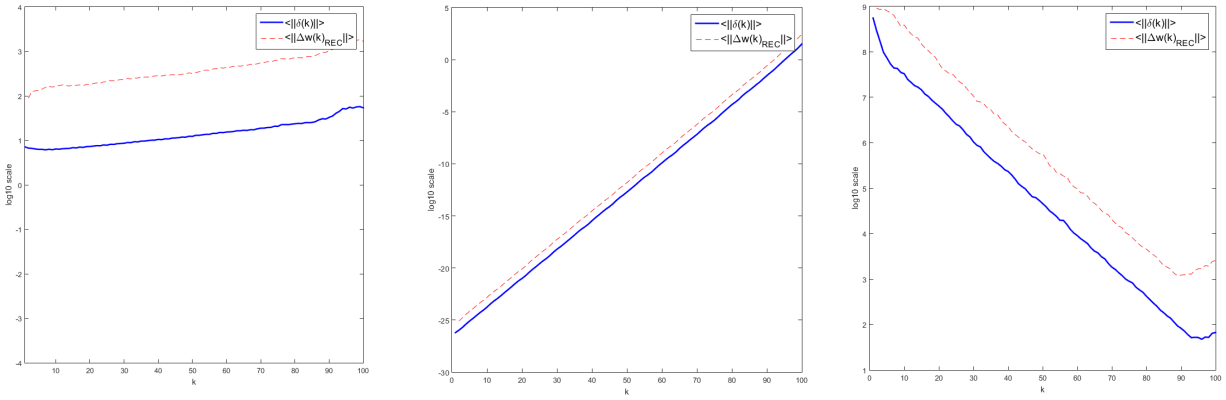
8. Результати експериментів

Для навчання використовуються рекурентні нейромережі SRN з прихованим шаром з 100 нейронів і активаційною функцією гіперболічного тангенса *tanh*, для вихідного шару нейронів була використана лінійна активаційна функція для задач регресії і функція *softmax* для задач класифікації. Як метод оптимізації було використано стохастичний градієнтний спуск (SGD) з моментом. З метою виконання коректного порівняння спочатку були згенеровані множини по 10 нейромереж і збережені. При навчанні різними методами варіювалися тільки алгоритми навчання, початкові вагові коефіцієнти нейромереж були, таким чином, одні й ті самі. В наших експериментах діапазон $[Q_{MIN}; Q_{MAX}]$ задано як $[-1;1]$. Швидкість навчання була встановлена $\alpha = 10^{-5}$, момент $\mu = 0{,}9$, розмір міні-пакетів 10 прикладів. Вибірка для тренування містить 20 000 прикладів для навчання, 1000 прикладів для валідації і 10 000 прикладів для тестування. Процес навчання містить 2000 епох, у кожній було використано по 50 ітерацій; тобто, загалом 100 000 оновлень вагових коефіцієнтів. Відбір найкращої нейромережі під час навчання здійснюється за принципом «save best»: після кожної епохи проводиться тестування на валідаційній виборці, та нейромережа, що показала найкращий результат за весь час навчання, тестується на тестовій виборці, цей результат вважається остаточним.

Правильно ініціалізувати вагові коефіцієнти нейромережі перед навчанням дуже важливо для успішного навчання в цілому. Останні досягнення в теорії глибоких нейромереж прямого поширення засновані на методах попереднього навчання вагових коефіцієнтів

нейромереж з використанням автоенкодерів або обмежених машин Больцмана [20]. У випадку рекурентних нейромереж хороша ініціалізація ще важливіша, оскільки падіння/зростання градієнтів має в більшості випадків монотонний характер через те, що градієнти пропускаються через одну і ту саму матрицю рекурентних зв'язків.

Вагові коефіцієнти нейромереж з кількістю нейронів у прихованому шарі $K=100$ були ініціалізовані малими випадковими значеннями з нормального розподілення з нульовим, середнім і середньоквадратичним відхиленням $\sigma=0,01$, як в [17]. На рис. 2 показані графіки середніх норм градієнтів нейромереж, розгорнутих назад у часі методом BPTT, що були ініціалізовані з різними параметрами середньоквадратичним відхиленням σ. Експеримент проводився на даних задачі «Порядок у часі».



а) $\sigma=0,01$ *б*) $\sigma=0,005$ *в*) $\sigma=0,02$

Рис. 2. Графіки середніх норм у часі зворотно-поширених градієнтів методом BPTT для нейромережі SRN, ініціалізованих довільними значеннями з різною дисперсією, горизонт BPTT $h=100$

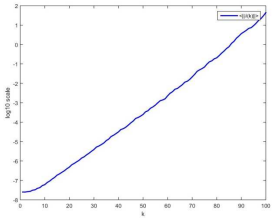
Кожний графік на рис. 2 містить дві криві: середню норму локальних градієнтів $\boldsymbol{\delta}(k)$ (суцільна лінія) та середню норму градієнтів $\Delta\mathbf{w}(k)_{rec}\equiv\dfrac{\partial E}{\partial\mathbf{w}_{rec}}$ (пунктирна лінія). З графіків на рис. 2 можна переконатися на практиці, що для контролю норм градієнтів $\dfrac{\partial E}{\partial\mathbf{w}}$, які змінюють вагові матриці нейромережі і найбільше нас цікавлять, достатньо контролювати лише норми локальних градієнтів $\boldsymbol{\delta}(k)$, оскільки норми градієнтів $\dfrac{\partial E}{\partial\mathbf{w}}$ безпосередньо залежать від них.

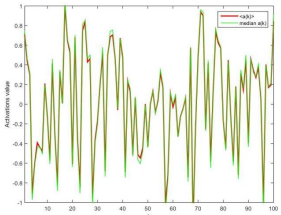
Як видно на рис. 2, течія зворотних градієнтів є дуже чутливою до параметрів розподілення випадкового процесу, з якого беруться малі значення для ініціалізації вагових коефіцієнтів. Величина дисперсії $\sigma=0,01$, що використовується в наших експериментах і була запозичена з [17], забезпечує плавну і рівномірну течію градієнтів (рис. 2 *а*). Q-factor для такої течії градієнтів $Q(\boldsymbol{\delta},h)\approx 0,7$ при $h=100$. Це добре для пошуку закономірностей як для довготривалих, так і для короткотривалих залежностей. Але зменшення або збільшення дисперсії вдвічі веде до катастрофічних наслідків для початкової течії градієнтів (рис. 2 *б* і *в*) відповідно. При дисперсії $\sigma=0,005$ середня норма градієнтів за $h=100$ кроків назад падає менше 10^{-25}, тут $Q(\boldsymbol{\delta},h)>25$, це класичний випадок ефекту зникнення градієнтів – «старі» вхідні дані практично не мають впливу на фінальну похибку $E(\mathbf{w})$ і на

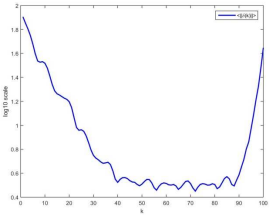
навчання нейромережі. При дисперсії $\sigma = 0,02$, навпаки, середня норма градієнтів вибухово зростає, тут $Q(\boldsymbol{\delta}, h) < -6$, і на похибку мають вплив лише «старі» вхідні дані.

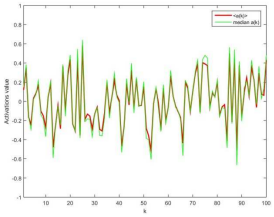
Нейромережі, що були погано ініціалізовані, мають малі шанси добре навчитися з використанням методів оптимізації 1-го порядку на кшталт SGD для задач, що містять довготривалі і короткотривалі залежності. Це пояснюється, по-перше, слабкою чутливістю методів 1-го порядку до малих коливань значень градієнтів, а по-друге, локальним характером градієнтних методів оптимізації взагалі.

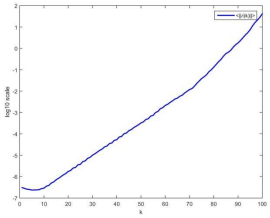
Проте хороша ініціалізація не є гарантією успішного навчання. На рис. 3 показаний окремий випадок навчання нейромережі SRN, а саме, процеси прямої і зворотної динаміки, що відбуваються в середині нейромережі. Зліва показані графіки зворотно-поширених норм градієнтів назад у часі, справа – середні і медіанні значення активацій для різних тактів у часі. Нейромережа, показана на рис. 3, була ініціалізована з дисперсією $\sigma = 0,01$, і графік падіння її градієнтів був подібний до рис. 2 *а*. Але вже після 500 ітерацій навчання норми градієнтів впали до значень менше 10^{-7}. Далі майже весь час навчання нейромережа знаходилась у зоні малих градієнтів $10^{-7}...10^{-8}$. З графіків на рис. 3, зліва, видно, що зоні малих градієнтів відповідають значення активацій нейромережі під час навчання, що часто знаходяться в зоні насичення. Такі значення активацій є ознакою поганої здатності нейромережі до навчання і узагальнення [21].

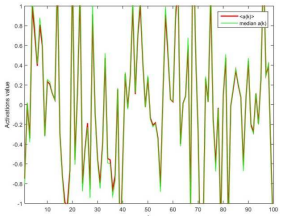


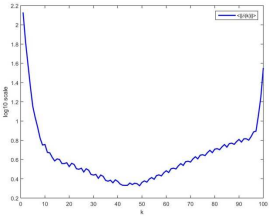












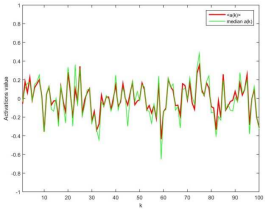


Рис. 3. Еволюція динаміки в середині нейромережі SRN під час навчання без регуляризації градієнтів. Зліва: графіки середніх норм у часі зворотно-поширених за методом BPTT градієнтів $\boldsymbol{\delta}(k)$. Справа: середні і медіанні значення активацій $\mathbf{a}(k)$ всередині

Рис. 4. Еволюція динаміки в середині нейромережі SRN під час навчання з регуляризацією градієнтів методом семплювання. Зліва: графіки середніх норм у часі зворотно-поширених за методом BPTT градієнтів $\boldsymbol{\delta}(k)$. Справа: середні і медіанні значення активацій $\mathbf{a}(k)$ всередині

Використання регуляризації градієнтів дозволяє утримувати норми зворотно-поширених градієнтів цієї нейромережі від зникнення або катастрофічного зростання. На рис. 4, зліва, показані графіки таких норм для тієї самої нейромережі, що і на рис. 3, але яка на цей раз навчалася з використанням регуляризації градієнтів методом семплювання. З графіків видно, що норми градієнтів лежать у прийнятному діапазоні, тут $Q(\boldsymbol{\delta}, h) \in [-1;1]$. Відповідні графіки значень активацій на рис. 4, справа, показують більш задовільну внутрішню динаміку, що майже знаходяться в зоні насиченості.

На рис. 5 показано якість навчання нейромереж для задачі «Порядок у часі» з використанням та без використання регуляризації градієнтів запропонованим методом семплювання для наборів ініціалізованих наборів з 10-ти нейромереж.

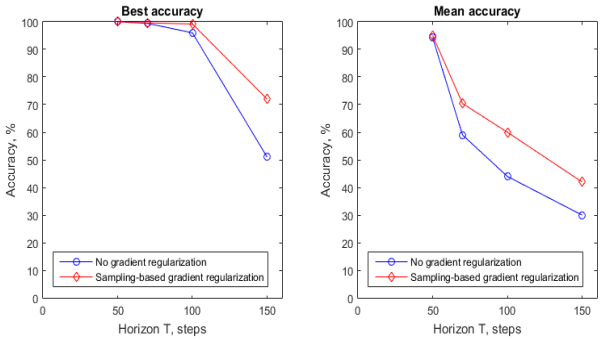


Рис. 5. Якість роботи навчених нейромереж на задачі «Порядок у часі» в залежності від довжини послідовності. Зліва: найкращі значення навчених нейромереж. Справа: середні значення точності навчених нейромереж

Як видно з графіків на рис. 5, використання розробленої регуляризації градієнтів дозволяє поліпшити якість навчання моделей. Для довжини послідовностей $T=50$ і за умови добре підібраних гіперпараметрів навчання з якістю майже 100% правильних відповідей можливо і без використання регуляризації градієнтів. Для довжин $T=100$ і $T=150$ в середньому поліпшення склало близько 10–20%. Якщо вважати успішною модель, що забезпечує кількість правильних відповідей вище 99%, то для послідовностей $T=100$ це мало вирішальне значення. Для інших задач результати експериментів наведено в табл. 1.

Таблиця 1. Кількість правильних відповідей навчених нейромереж для різних задач, що містять довготривалі залежності, без регуляризації градієнтів (звичайний спосіб навчання) та з регуляризацією градієнтів (запропонований спосіб), параметр $T=100$

	«Додавання»		«Множення»		«Порядок у часі»		«Порядок у часі 3-bit»	
	best	mean	best	mean	best	mean	best	mean
Без регуляризації	99%	68%	>99%	72%	96%	44%	>99%	50%
З регуляризацією	>99%	96%	>99%	68%	>99%	60%	>99%	62%

Для послідовностей довжини $T=100$ ми змогли натренувати принаймні одну успішну модель (з точністю >99%, як вимагається в [19]) для всіх чотирьох задач з використанням запропонованого методу. Для двох задач («Додавання» і «Порядок у часі») успішне навчання було неможливе з використанням традиційного навчання. На жаль, для послідовностей довжини $T=150$ не вдалося навчити успішну модель ні за допомогою запропонованого методу, ні за допомогою традиційного методу. Тим часом, з використанням запропонованого методу регуляризації градієнтів вдалося поліпшити кращі і середні показники майже в усіх випадках.

Таблиця 2. Кількість правильних відповідей навчених нейромереж для різних задач, що містять довготривалі залежності, без регуляризації градієнтів (звичайний спосіб навчання) та з регуляризацією градієнтів (запропонований спосіб), параметр $T=150$

	«Додавання»		«Множення»		«Порядок у часі»		«Порядок у часі 3-bit»	
	best	mean	best	mean	Best	mean	best	mean
Без регуляризації	34%	11%	-	-	51%	30%	32%	24%
З регуляризацією	47%	13%	-	-	72%	42%	37%	30%

Додатково треба відзначити декілька важливих моментів стосовно використання методу семплювання для псевдорегуляризації градієнтів. Ті елементи вибірки, що були не використані алгоритмом навчання, не обов'язково втрачені для подальшого використання.

По-перше, такі приклади можуть бути використані під час знаходження нейромережі в «безпечній» зоні. Стохастична природа алгоритму SGD сприяє такому розвитку подій. По-друге, через певний час може виникнути потреба зміни норми градієнтів у протилежному напрямку. Наприклад, буде потрібно не зменшувати, а збільшувати норму. По-третє, вагові коефіцієнти нейромережі поступово змінюються, і ті приклади, що зменшували норму градієнтів, можуть почати її збільшувати, і навпаки.

Другим зауваженням є чутливість методу семплювання для регуляризації градієнтів до вибору гіперпараметрів, яка є взагалі притаманною для навчання нейромереж, що здійснюється для методів оптимізації 1-го порядку. Діапазон «безпечних норм градієнтів» $Q(\boldsymbol{\delta}, h) \in [-1;1]$ було знайдено емпірично для тих задач, що використовувалися для експериментів. Для інших задач і довжин залежностей у даних цей діапазон може бути іншим.

10. Висновки

У роботі було запропоновано метод семплювання для регуляризації градієнтів (псевдорегуляризації) для контроля норми градієнтів під час навчання. Це підвищило здатність нейромережі підтримувати довготривалу пам'ять в середині рекурентних зв'язків без введення такої апріорної інформації в архітектуру нейромережі. В цьому випадку нейромережа вчиться не лише підлаштовуватись під дані навчальної вибірки, але й тримати норму зворотного потоку градієнтів у певних межах. Було експериментально досліджено розроблений алгоритм та показано збільшення ефективності навчання рекурентних нейромереж типу Simple Recurrent Network на довгострокових (до $T = 150$ кроків) залежностях на 10-20%. Важливою практичною перевагою запропонованого вдосконаленого алгоритму регуляризації градієнтів є можливість його реалізації в системах математичного моделювання без вбудованої системи символьної математики для автоматичного обчислення похідних.

СПИСОК ЛІТЕРАТУРИ

1. Hinton G. E. Reducing the dimensionality of data with neural networks / G.E. Hinton, R.R. Salakhutdinov // Science. – 2006. – Vol. 313, N 5786. – P. 504 – 507.
2. Greedy layer-wise training of deep networks / Y. Bengio [et al.] // Advances in neural information processing systems. – 2007. – Vol. 19. – P. 153.
3. Boné R. Advanced Methods for Time Series Prediction Using Recurrent Neural Networks / R. Boné, H. Cardot // Recurrent Neural Networks for Temporal Data Processing. – 2011. – P. 25.
4. Jaeger H. The "echo state" approach to analysing and training recurrent neural networks-with an erratum note / H. Jaeger // German National Research Center for Information Technology GMD Technical Report. – Bonn, Germany, 2001. – Vol. 148. – 34 p.
5. Jaeger H. Long short-term memory in echo state networks: Details of a simulation study / H. Jaeger // Technical report, Jacobs University Bremen. – 2012. – 29 p.
6. Prokhorov D. Echo state networks: appeal and challenges / D. Prokhorov // Neural Networks. IJCNN'05. Proc. IEEE International Joint Conference on. – 2005. – Vol. 3. – P. 1463 – 1466.
7. Hochreiter S. Untersuchungen zu dynamischen neuronalen Netzen: Master's thesis / Hochreiter S. – Institut fur Informatik, Technische Universitat, Munchen, 1991. – 74 p.
8. Graves A. Speech recognition with deep recurrent neural networks / A. Graves, A. Mohamed, G. Hinton // Acoustics, Speech and Signal Proc. (ICASSP), IEEE International Conference on. – 2013. – P. 6645 – 6649.
9. Donahue J. Long-term recurrent convolutional networks for visual recognition and description / J. Donahue // Proc. of the IEEE Conference on Computer Vision and Pattern Recognition. – 2015. – P. 2625 – 2634.
10. Graves A. Towards end-to-end speech recognition with recurrent neural networks / A. Graves, N. Jaitly // Proc. of the 31st International Conference on Machine Learning (ICML-2014). – Beijing, China, 2014. – P. 1764 – 1772.

11. Cho K. On the properties of neural machine translation: Encoder-decoder approaches / K. Cho // Proc. Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation (SSST-8). – Doha, Qatar, 2014. – P. 103 – 111.
12. Ozefowicz R. An empirical exploration of recurrent network architectures / R. Ozefowicz, W. Zaremba, I. Sutskever // Proc. of the 32nd International Conference on Machine Learning (ICML-15). – Lille, France, 2015. – P. 2342 – 2350.
13. Rachkovskij D. Approaches to sequence similarity representation / D. Rachkovskij A. Sokolov // Journal of Information Theories and Applications. – 2005. – Vol. 13. – N 3. – P. 272– 278.
14. Mikolov T. Statistical Language Models based on Neural Networks: Ph.D. thesis, manuscript / Mikolov T. – Brno University of Technology, 2012. – 129 p.
15. Bengio Y. Advances in Optimizing Recurrent Networks / Y. Bengio, N. Boulanger-Lewandowski, R. Pascanu // Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), (26–31 May 2013). – Vancouver, Canada, 2013. – P. 8624 – 8628.
16. Sutskever I. Training Recurrent Neural Networks: Ph.D. thesis, manuscript / Sutskeve I. – University of Toronto, 2013. – 93 p.
17. Pascanu R. On the difficulty of training recurrent neural networks / R. Pascanu, T. Mikolov, Y. Bengio // Proc. of the 30th International Conference on Machine Learning (ICML-2013) . – Atlanta, USA. – 2013. – P. 1310 – 1318.
18. Bengio Y. Learning long-term dependencies with gradient descent is difficult / Y. Bengio, P. Simard, P. Frasconi // Neural Networks, IEEE Transactions on. – 1994. – Vol. 5, N 2. – P. 157 – 166.
19. Hochreiter S. Long short-term memory / S. Hochreiter, J. Schmidhuber // Neural computation. – 1997. – Vol. 9, N 8. – P. 1735 – 1780.
20. Bengio Y. Deep learning MIT Press book in preparation [Електронний ресурс] / Y. Bengio, I.J. Goodfellow, A. Courville // Режим доступу: www. iro. umontreal. ca/~ bengioy/dlbook.
21. Chernodub A.N. Training Neuroemulators Using Multicriteria Extended Kalman Filter and Pseudoregularization for Model Reference Adaptive Neurocontrol / A.N. Chernodub // IEEE IV International Congress on Ultra Modern Telecommunications and Control Systems (ICUMT'2012), (St. Petersburg, Russia, October 3 – 5, 2012). – St. Petersburg, 2012. – P. 397 – 402.
22. Чернодуб А.Н. Обучение нейроэмуляторов с использованием псевдорегуляризации для метода нейроуправления с эталонной моделью / А.Н. Чернодуб // Искусственный интеллект. – 2012. – № 4. – С. 602 – 614.
23. Чернодуб А.М. Навчання рекурентних нейронних мереж методом псевдорегуляризації для багатокрокового прогнозування часових рядів / А.М. Чернодуб // Математичні машини і системи. – 2012. – № 4. – С. 41 – 51.
24. Glorot X. Understanding the difficulty of training deep feedforward neural networks / X. Glorot, Y. Bengio // International conference on artificial intelligence and statistics. – 2010. – P. 249 – 256.
25. Zaremba W. Recurrent neural network regularization / W. Zaremba, I. Sutskever, O. Vinyals // arXiv preprint arXiv:1409.2329. – San Diego, USA, 2015. – P. 1001 – 1008.
26. Martens J. Learning recurrent neural networks with hessian-free optimization / J. Martens, I. Sutskever // Proc. of the 28th International Conference on Machine Learning (ICML-11). – 2011. – P. 1033 – 1040.

Стаття надійшла до редакції 05.04.2016