

О.С. КОСУХІНА, кандидат технічних наук, доцент, e_kos@ukr.net
Дніпровський державний технічний університет, м. Кам'янське
С.Є. ТОНКОНОГ, магістрант, sofia.tonkonoh@gmail.com
Одеський національний політехнічний університет, м. Одеса

Алгоритмічне забезпечення методу прогнозування обсягів споживання електроенергії з використанням рекурентної нейронної мережі

В рамках дослідження розглянуто проблему погодинного прогнозування споживання електроенергії. На основі вхідних даних у вигляді часового ряду споживання електроенергії за певний період здійснено короткострокове прогнозування за допомогою рекурентної нейронної мережі.

The study examined the problem of hourly forecasting of electricity consumption. Based on the input data in the form of a time series of electricity consumption for a certain period, short-term forecasting was performed using a recurrent neural network.

Постановка проблеми

Завдання точного прогнозування споживання електроенергії набуває дуже важливого значення для вирішення проблем, пов'язаних з плануванням роботи енергосистем, розподілом потужностей, оперативним управлінням енергетичною галуззю. Похибки прогнозування неодмінно призводять до необґрунтованих витрат. Це пов'язано з тим, що переоцінка майбутнього споживання веде до невіправданої перевитрати палива всіх видів, а його недооцінка до зниження якості енергопостачання споживачів. У зв'язку з цим підвищення точності прогнозування споживання електроенергії дійсно є актуальною проблемою, яка потребує ефективних методів її вирішення.

У даній роботі розглядається задача погодинного прогнозування споживання електроенергії. Дана задача відноситься до короткострокового прогнозування. У якості вхідних даних маємо часовий ряд споживання електроенергії в Україні за 2014-2016 р. Одиниці виміру MW. Потрібно спрогнозувати споживання електроенергії на одну, п'ять та десять годин вперед за допомогою нейронної мережі. Оцінити отримані результати, обчисливши середньоквадратичну похибку прогнозу.

Аналіз останніх досліджень та публікацій

Прогнозування електроспоживання здійснюється за допомогою різних методів, які базуються на аналізі ретроспективної динаміки електроспоживання і факторів його впливу та виявленні статистичного зв'язку між ними для побудови прогнозних моделей з використанням різноманітних методів і програмних засобів.

У 2000-х роках було розроблено такі програмні реалізації методів статистичного аналізу і прогнозу часових послідовностей: CatePillar, ForExSale, NCSS97. Всі вони є універсальними і не спеціалізовані під конкретну предметну область, тому не можуть бути використані диспетчером як дорадча інтегрована автоматизована система у галузі електроенергетики.

Серед багатьох підходів, що використовуються в даній час для вирішення розглянутої задачі, можна виділити традиційні методи аналізу часових рядів і, насамперед, підхід Боксу-Дженкінса, регресійні алгоритми, і як найбільш ефективні методи обчислювального інтелекту і, перш за все, штучні нейронні мережі. Успіх штучних нейронних мереж в розглянутій проблемі пояснюється нелінійним характером прогнозованих процесів та здатністю нейронної мережі до самонавчання та узагальнення, високим рівнем невизначеності

(структурної та параметричної), стохастичністю і хаотичністю. На сьогоднішній день можна відзначити досить велике число вдалих прикладів використання нейронних мереж в задачі прогнозування споживання електроенергії як у нас в країні, так і за кордоном. В переважній більшості випадків в основі прогнозуючої системи лежить багатосаровий перцептрон з усіма своїми модифікаціями, об'єднаними загальною архітектурою з прямою передачею інформації.

Альтернативою нейронних мереж з прямою передачею інформації в задачах прогнозування можуть служити рекурентні нейронні мережі, що включають у свою архітектуру як глобальні, так і локальні (на рівні шарів) зворотні зв'язки і навчають за допомогою спеціалізованих процедур [1]. Завдяки, перш за все, своїм універсальним апроксимуючим і екстраполюючим можливостям і здатності до навчання в умовах суттєвої структурної і параметричної невизначеності характеристик прогнозованих процесів. В більшості випадків рекурентні нейронні мережі з обчислювальної точки зору набагато більш ефективні, ніж мережі з прямою передачею інформації [2]. До теперішнього часу в задачах обробки нелінійних часових рядів найбільшого поширення набули три типи рекурентних нейронних мереж: мережі Вільямса-Зіпсера [3], Елмана [4] і Джордана [5]. Для вирішення задач аналізу і передбачення часових рядів і виявлення змін їх властивостей ці мережі вимагають істотної модифікації, що стосується, перш за все, алгоритмів навчання, оскільки всі зазначені нейронні мережі навчаються в пакетному режимі і не передбачають ситуацію, коли дані спостережень часового ряду надходять на обробку послідовно одне за іншим.

У прикладній області споживання електроенергії є різні типи прогнозування навантаження, які в основному відрізняються за встановленим проміжком прогнозу. В даній роботі розглядається саме короткострокове прогнозування.

Формулювання мети дослідження

Метою даної роботи було підвищення якості прогнозування погодинного споживання електроенергії шляхом розробки алгоритмічного та програмного забезпечення методу з використанням рекурентної нейронної мережі.

Виклад основного матеріалу

Розглядаємо послідовність спостережень часового ряду y_0, y_1, \dots, y_t для прогнозування наступних

значень цього ряду $y_{t+1}, y_{t+2}, \dots, y_{t+K}$ з абсолютною похибкою меншою за певне значення ε :

$$|\tilde{y}_{t+i} - y_{t+i}| < \varepsilon, \quad i = \overline{1, K}. \quad (1)$$

Часовий ряд споживання електроенергії характеризується хаотичною динамікою. Ми припускаємо, що всі перехідні процеси в системі, були завершені, і часовий ряд відображає траєкторію руху в околі дивного атратора. При багатокроковому прогнозі, слід взяти до уваги, що для хаотичного часового ряду, прогноз може бути виконаним належним чином до певної межі (горизонт прогнозу). Існування горизонту пояснюється тим, що, для хаотичних рядів, похибка даних, яка була мала в початковій точці часу, зростає в геометричній прогресії у зв'язку з розбіжністю траєкторій, які спочатку були близькі. Якщо траєкторія системи рухається по тій же області атратора, що пов'язана з цією ділянкою, ряд вважається нормалізованим.

Рекурентні нейронні мережі (РНМ) — це клас штучних нейронних мереж, у якому з'єднання між вузлами утворюють орієнтований цикл. Це створює внутрішній стан мережі, що дозволяє їй проявляти динамічний поведінку в часі. На відміну від нейронних мереж прямого поширення, РНМ можуть використовувати свою внутрішню пам'ять для обробки довільних послідовностей входів.

В якості тестової моделі обираємо мережу Елмана тому що, вона має можливість навчання та вирішення численних практичних задач.

Також мережа Елмана може зберігати свого роду стан, що дозволяє їй виконувати такі задачі, як передбачення послідовностей, що є за межами можливостей стандартного багатосарового перцептронів. Формально описати динаміку моделі можна наступним чином:

$$p(t) = f(w_1 x(t) + w_2 p(t-1)), \quad (2)$$

$$y(t) = f(w_3 p(t)), \quad (3)$$

де $x(t)$ — вектор вхідних сигналів розмірністю m , $p(t)$ — вектор вихідних сигналів прихованого шару розмірністю q , $f(\bullet)$ — нелінійна функція, яка характеризує прихований шар, w_1 — матриця синаптичних ваг, які поєднують прихований шар та вхідний, w_2 — матриця синаптичних ваг, які поєднують прихований шар та контекст, w_3 — матриця синаптичних ваг, які поєднують вихідний шар та прихований, $y(t)$ — вихідний сигнал мережі.

Мережа має рекурентні зв'язки прихованих нейронів із шаром контексту. Ці контекстні елементи зберігають виходи прихованих нейронів на один крок в часі, після чого передають їх на вихідний шар. Таким чином нейрони запам'ятовують свої попередні дії. Приховані нейрони також передають інформацію вихідним нейронам, які формують реакцію мережі на зовнішнє збурення. Так як природа зворотного зв'язку пов'язана виключно з прихованими нейронами, вони можуть поширювати повторювані цикли інформації по мережі на протязі великої кількості кроків в часі і таким чином відкривати доступ до абстрактного представлення часу. У момент часу t вхідні нейрони отримують перший вхід послідовності. Кожен вхід може бути скаляр або вектор, в залежності від характеру задачі. Контекстні елементи

спочатку налаштовані на 0,5. Вхідні нейрони та контекст активують приховані нейрони; а вони в свою чергу активують вихідний шар. Приховані нейрони також посилюють сигнали до контексту. Під час наступного кроку $t+1$ послідовність повторюється. На цей раз контекст містить значення прихованого шару в момент t . Ці контекстні блоки таким чином, можуть забезпечити мережу пам'яттю.

Для навчання мережі використовуємо алгоритм зворотного поширення похибки. Перш за все рекурентні з'єднання фіксуються на 1,0 і не підлягають коригуванню.

Навчання методом зворотного поширення помилки передбачає два проходи по всім шарам мережі: прямий і зворотний. При прямому проході образ (вхідний вектор) подається на сенсорні вузли мережі, після чого поширюється по мережі від шару до шару. В результаті генерується набір вихідних сигналів, який і є фактичною реакцією мережі на даний вхідний образ. Під час прямого ходу усі синаптичні ваги мережі фіксовані. Під час зворотного проходу всі синаптичні ваги налаштовуються відповідно до правила корекції помилок, а саме: фактичний вихід мережі віднімається з бажаного відгуку, в результаті чого формується сигнал помилки. Цей сигнал поширюється по мережі в напрямку, зворотному напрямку синаптичних зв'язків. Звідси і назва алгоритму зворотного поширення помилки. Синаптичні ваги налаштовуються з метою максимального наближення вихідного сигналу мережі до бажаного. Процес навчання, реалізований цим алгоритмом, називається навчанням на основі зворотного розповсюдження.

Нехай навчальна вибірка представлена у вигляді:

$$(u(k), d(k)), \quad k = \overline{1, K},$$

де $d(k)$ — очікуваний результат роботи мережі. Після прямого ходу вираховуємо помилку мережі для поточного вектора:

$$E = \frac{1}{2} \sum_{j=1}^{N_t} (y_j^{(L)} - d_j)^2, \quad (4)$$

де $y_j^{(L)}$ — вихід мережі, d_j — очікуваний результат.

Задача зводиться до мінімізації функції помилки. Для цього використовуємо метод градієнтного спуску:

$$w_{ij}^{(l)}(t+1) = w_{ij}^{(l)}(t) - \gamma \Delta w_{ij}^{(l)}, \quad (5)$$

$$\Delta w_{ij}^{(l)} = \delta_i^{(l)} y_j^{(l-1)}, \quad (l = \overline{0, L}, i = \overline{0, N}, j = \overline{0, M}) \quad (6)$$

де γ — коефіцієнт навчання, $\delta_i^{(l)}$ — помилка i -го нейрона l -го шару.

Для останнього шару $\delta_i^{(l)}$ буде мати вигляд:

$$\delta_i^{(l)} = f'(s_i^{(L)}) \cdot (y_i^{(L)} - d_i), \quad (7)$$

де $f'(s_i^{(L)})$ — похідна від функції активації.

Функцію активації нейрона вибираємо у вигляді

$$f(s) = \tanh(s) = \frac{1 - e^{-s}}{1 + e^{-s}}.$$

Для всіх інших шарів функція помилки нейрона буде мати вигляд:

$$\delta_i^{(l)} = f'(s_i^{(l)}) \cdot \sum_{j=1}^{N_{l+1}} \delta_j^{(l+1)} w_{ij}^{(l+1)}, \quad (8)$$

Процедуру навчання потрібно неодноразово повторювати. Іноді навіть проходження по всім векторам навчальної вибірки може не мати достатнього ефекту для зміни ваг. Проходження по всім векторам навчальної вибірки зі зміною ваг називається епохою навчання. Число епох може досягати сотень тисяч. Функціонування мережі ділиться на 2 етапи:

- 1) етап навчання;
- 2) етап функціонування в реальних ситуаціях.

Здатність мережі функціонувати на основі даних, на яких вона не навчалася є здатністю до генералізації або узагальнення. Метод перевірки здатності до генералізації називається стратегією навчання.

Для мережі Елмана використаємо наступну стратегію. Вся вибірка ділиться на підмножини:

- 1) вибірка для навчання (70 %);
- 2) вибірка для тестування (30 %).

Розподілення векторів по вибіркам випадкове. Після завершення кожної епохи фіксуємо ваги та пропускаємо спочатку всю вибірку для навчання, потім для тестування. Для кожної вибірки рахуємо середню помилку по всім векторам (помилка навчання та помилка узагальнення відповідно). При збільшенні числа епох помилки повинні збігатися одна до іншої і мережа вважається навченою якщо на протязі тривалого часу помилки близькі за значенням. Це дає критерій зупинки навчання. Величина до якої збігаються помилки називається потужністю мережі. Вона характеризує здатність мережі працювати з даною вибіркою. Якщо ж помилки вийшли на деякі асимптотичні криві і практично не змінюються на протязі великого числа епох, то це означає, що потужність мережі недостатня для роботи з даними.

Побудуємо алгоритм вирішення задачі прогнозування за допомогою мережі Елмана. Архітектуру мережі оберемо наступним чином: кількість нейронів на вихідному шарі буде дорівнювати одиниці, а на вхідному шарі та прихованому будемо підбирати кількість нейронів експериментально, оцінюючи для кожного варіанту похибку прогнозування за формулою:

$$MSE = \sum_{i=1}^{n_L} \left(\frac{y_i - d_i}{y_i} \right)^2, \quad (9)$$

де MSE — значення відносної середньоквадратичної похибки, y_i — вихід мережі на i -му кроці, d_i — бажаний відгук мережі.

Тоді алгоритм можна представити у вигляді наступних кроків:

1. Підготовка даних до подання на нейронну мережу. Так як вибірка має бути нормалізованою, то можемо це зробити за наступною формулою:

$$\tilde{x}_i = \frac{2 \cdot \left(x_i - \frac{1}{2} \left(\min_{x \in X} x + \max_{x \in X} x \right) \right)}{\left(\min_{x \in X} x + \max_{x \in X} x \right)}, \quad (10)$$

де \tilde{x}_i — нормоване значення вхідного сигналу x_i .

2. Задаємо кількість нейронів на першому та прихованому шарі та створюємо мережу.

3. Навчаємо мережу, за наступними етапами.

3.1 Сформувавши навчальну вибірку. Для нашого випадку це 20000 значень. Розподілити їх на вектори для навчання та тестування випадковим чином.

3.2 Подати на вхід мережі вектор навчаючої вибірки.

3.3 Прямий хід за алгоритмом оберненого поширення похибки.

3.4 Оцінити похибку отриманих результатів за формулою (4).

3.5 Зворотній хід за алгоритмом оберненого поширення похибки.

3.6 Фіксуємо ваги мережі.

3.7 Пропускаємо всю навчальну вибірку через мережу, обчислюємо похибку навчання.

3.8 Пропускаємо всю тестову вибірку через мережу, обчислюємо похибку генералізації.

3.9 Кроки 3.1—3.8 повторюються на протязі достатньо великої кількості епох, доки похибки навчання та генералізації не зійдуться з потрібною точністю.

4. Після завершення етапу навчання, можемо випробувати мережу для прогнозування. Вибираємо, на скільки кроків вперед потрібно отримати прогноз: один, п'ять чи десять. Подаємо на вхід мережі вибірку для опрацювання. Після цього за допомогою прямого ходу мережі отримуємо значення прогнозу.

5. За формулою (9) оцінюємо похибку отриманого результату.

На основі побудованого алгоритму проведено проектування програмної системи прогнозування споживання електроенергії. Під час проектування системи треба чітко визначити усі варіанти використання систем, усі можливі прецеденти, розробити архітектуру системи, тощо. Тому було побудовано діаграму прецедентів (рис.1), яка є графічним відображенням різних варіантів використання системи користувачем.

Розроблена програма складається з трьох основних частин: «Інтерфейс», «Розрахунковий блок» та «Інформаційний блок». Вони, в свою чергу, поділяються на модулі.

Блок «Інтерфейс» поділяється на 6 модулів, кожен з яких виконує свою функцію. «Розрахунковий блок» складається з 5 модулів, серед яких: «Навчання мережі Елмана», в якому відбувається навчання нейронної мережі; «Розрахунок похибки», в якому виконується розрахунок відносної похибки. Наступний модуль — «Прогнозування на 1; 5 та 10 год.», в якому виконується прогнозування за допомогою навченої нейронної мережі. Останній пункт «Побудова графіків», де виконується побудова графіків за результатами прогнозування (рис. 3, 4).

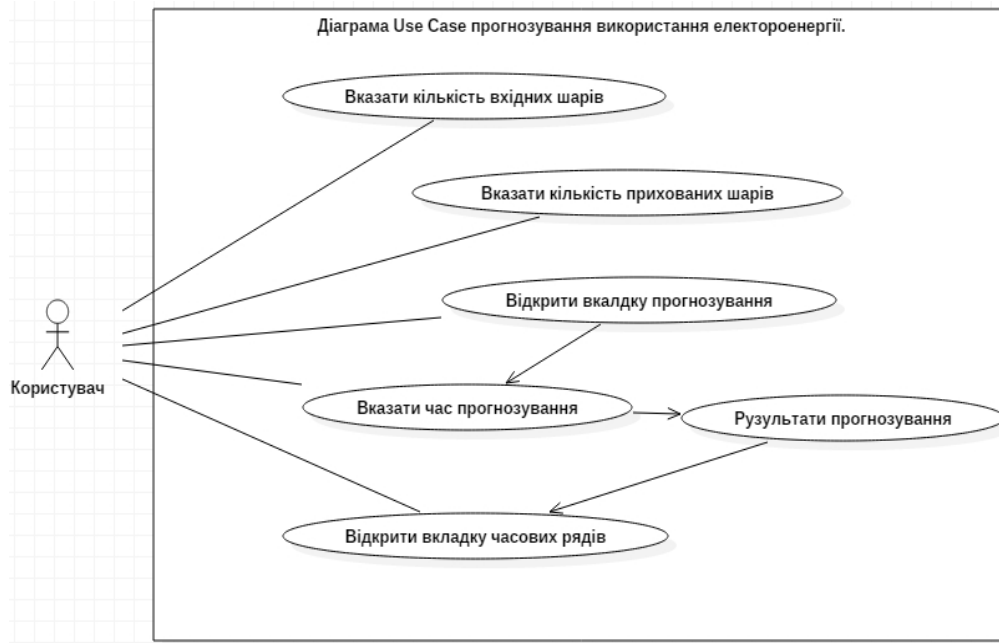


Рис. 1. Діаграма прецедентів

А також діаграму класів (рис. 2), яка має 5 класів а саме:

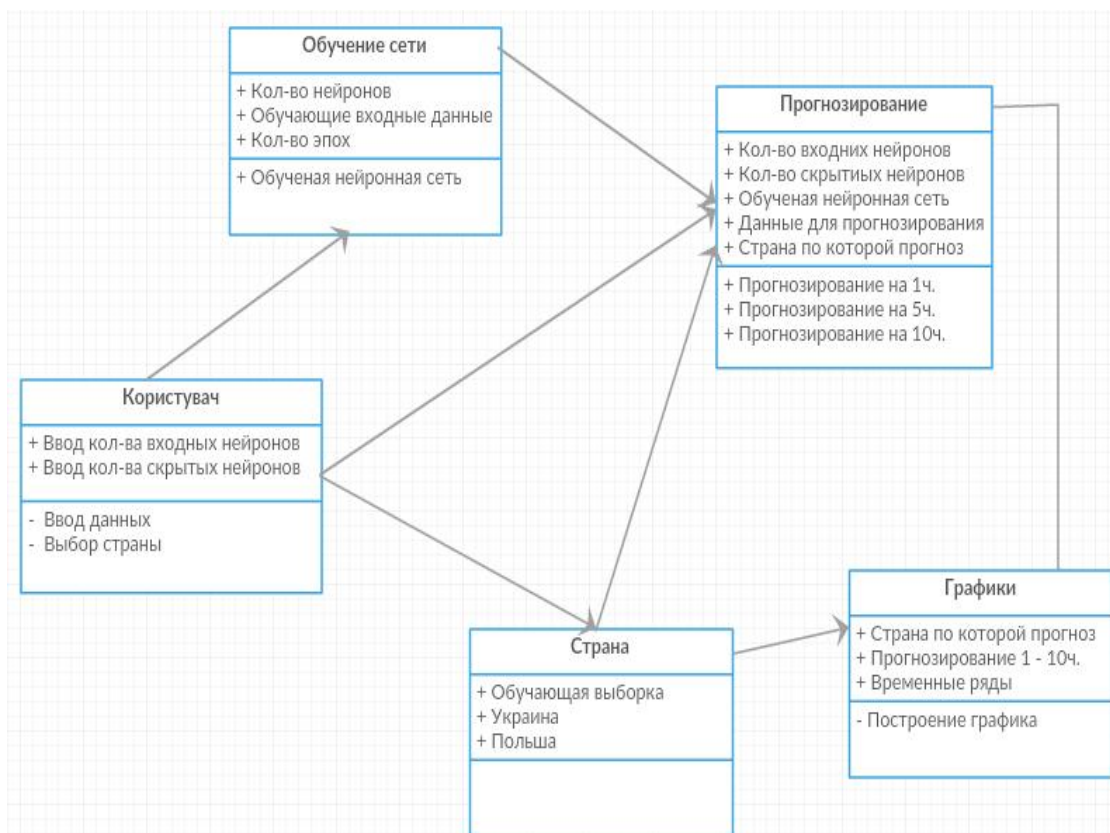


Рис. 2. Діаграма класів

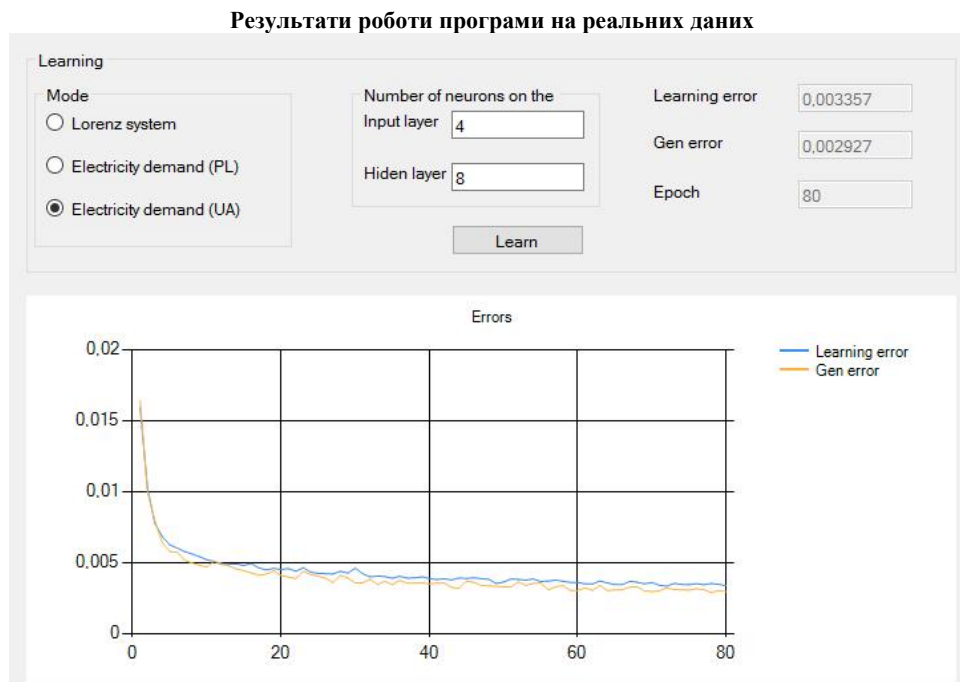


Рис. 3. Результати навчання мережі

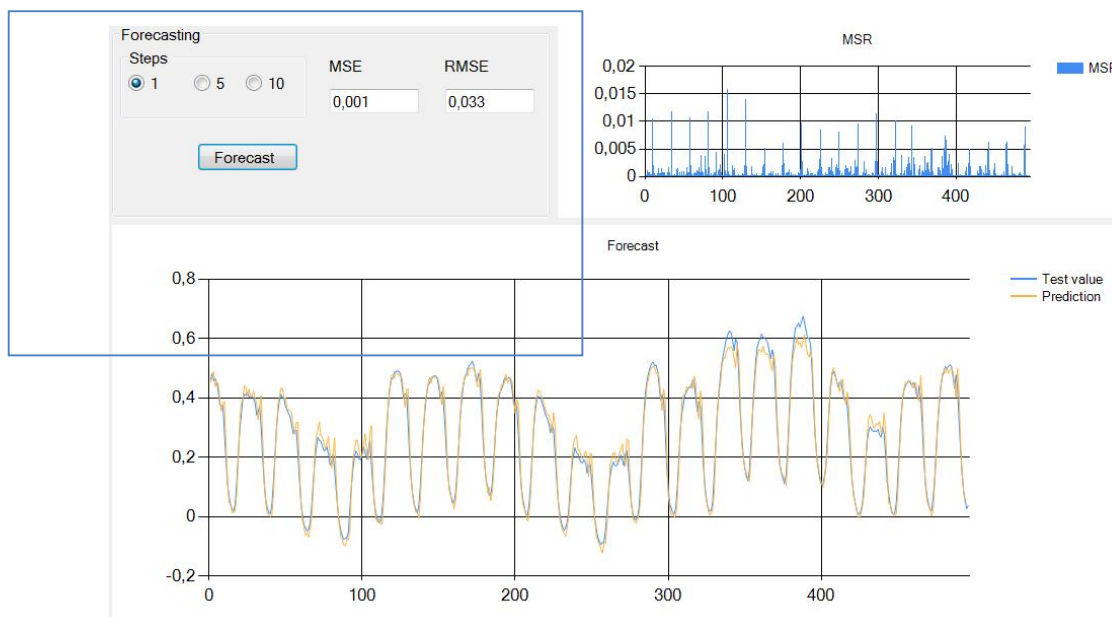


Рис. 4. Результати прогнозування електроспоживання на один крок вперед

Висновки

В результаті проведеної роботи було розроблено алгоритмічне та програмне забезпечення методу прогнозування часових рядів з використанням рекурентної нейронної мережі.

В результаті аналізу існуючих методів прогнозування з використанням нейронних мереж було застосовано мережу Елмана, тому що вона може зберігати свого роду стан, що дозволяє їй виконувати такі задачі, як передбачення послідовностей, що є за межами можливостей стандартного багатощарового перцептрон.

Розроблено алгоритмічне та програмне забезпечення методу прогнозування часових рядів з використанням рекурентної нейронної мережі.

Час виконання навчання та прогнозування даною програмою складає 60 сек. Даний час прогнозування навіть із навчанням є допустимим оскільки час прогнозування складає 1/60 частину мінімального часового відрізка, прогнозування.

Відносна похибка навчання нейронної мережі складає 0.011963, тоді коли похибка прогнозування на реальних даних складає 0.011855. Таким чином похибка прогнозування не перевищує похибки навчання.

ЛІТЕРАТУРА

1. Tzafestas S. Computational intelligence techniques for short-term electric load forecasting / S. Tzafestas, E. Tzafestas // *Journal of Intelligent and Robotic Systems*. – 2001. – 31. – P. 7–68.
2. Mandic D.P. *Recurrent Neural Networks for Prediction* / D.P. Mandic, J.A. Chambers. – Chichester: John Wiley&Sons, 2001. – 285 p.
3. Williams R.J. A Learning Algorithm for Continually Running Fully Recurrent Neural Networks / R.J. Williams, D. Zipser // *Neural Computation*. – 1989. – 1. – P. 270–280.
4. Elman J.L. Finding structure in time / J.L. Elman // *Cognitive Science*. – 1990. – 14. – P. 179–211.
5. Jordan M. Constrained supervised learning / M. Jordan // *Journal of Mathematical Psychology*. – 1992. – 36. – P. 396–452.
6. Vikko N., Lautala P. Short-term electric power production scheduling using simulated annealing algorithm: Proc. of the IASTED Inter. Sym / N. Vikko, P. Lautala // ACTA Press, Anaheim, CA, USA, 1990.
7. Чумаченко, Е.И. Алгоритм решения задачи прогнозирования / Е.И. Чумаченко, В.С. Горбатюк // *Искусственный интеллект*. – 2012. – № 2. – С. 24–30

пост. 06.10.2017