

УДК 004.89(045)

**В.П. Харченко**, д.т.н., проф.  
**О.М. Алексєєв**, асист.

## ПРОГРАМНА РЕАЛІЗАЦІЯ ШТУЧНОЇ НЕЙРОННОЇ МЕРЕЖІ В АВТОМАТИЗОВАНИХ ІНТЕЛЕКТУАЛЬНИХ СИСТЕМАХ

*Розглянуто диференційну діагностику факторів ризику виникнення авіаційних подій. Запропоновано застосування нейронно-мережових технологій.*

*Application of neural networks technologies effectively decides the task of synthesis of origin of accident risk and gives out the vector of managing signals of network on incomplete and distorted information about the phenomena, events and processes which influence on safety flights.*

**автоматизовані системи, безпека польотів, діагностика, нейронні мережі, подвійний сигнал, синапс, фактори ризику, функціонування**

### Постановка проблеми

Авіаційна галузь є складною поліергатиною системою особливої відповідальності і характеризується різноманіттям вирішуваних завдань і чисельністю вимог, які вони мають задовольняти. Прийняття рішення про вибір кращого варіанта засобу керування безпекою польотів (КБП) у таких умовах зводиться до задоволення цілого ряду показників і незважаючи на їх можливу суперечність, є невирішеною проблемою [1; 2].

Диференціальна діагностика факторів ризику (ФР) виникнення авіаційних подій (АП) пов'язана зі значними труднощами ймовірнісного характеру. Недоліки апріорної інформації призводять до того, що рішення на всіх етапах приймаються в умовах невизначеності, а характеристики кожного експлуатанта є індивідуальними [2].

Застосування нейронних мереж (НМ) дозволять ефективно розв'язати задачу синтезу виникнення ризику АП та видати вектор керувальних сигналів мережі за неповною та спотвореною інформацією про явища, події та процеси, що впливають на безпеку польотів [3].

### Аналіз останніх досліджень і публікацій

Значну увагу в наукових працях провідних учених у галузі авіації [1; 4–7] приділено локальним діям з прогнозування, ідентифікації та керування ФР, а також висвітленню основних методологічних питань щодо створення і застосування математичних моделей КБП, проте немає матеріалів, що містять комплексне інтегроване розв'язання проблеми КБП та її автоматизації.

Одним із методів підвищення ефективності КБП є автоматизація обробки та інтерпретації даних з використанням сучасних комп'ютерних технологій як методу, що дозволяє взяти до уваги велику

кількість ФР з урахуванням їх індивідуального коефіцієнта значущості для кожного експлуатанта та водночас виключити помилки, спричинені суб'єктивними факторами, такими, як недооцінювання значущості окремих факторів, упереженість думки того чи іншого експерта та ін.

### Нейромережовий підхід до керування безпекою польотів

На основі програмного середовища Delphi розроблено програмну реалізацію штучної НМ автоматизованої системи КБП (АСКБП) [8–10]. Базовим типом елементів, що використовуються для опису НМ в АСКБП, є загальний елемент НМ – клас, що інкапсулює основні властивості і методи, характерні для всіх компонентів НМ. Назва цього класу в програмній реалізації – TNetPiece.

Об'єкти цього класу містять такі поля:

NextPiece – покажчик на наступний елемент НМ;  
PriorPiece – покажчик на попередній елемент НМ;

ForwOut – значення сигналу, що передається елементом вперед за прямого функціонування;

BackOut – значення сигналу, що передається елементом назад за зворотного функціонування.

Набір методів включає:

Create – опис створення об'єкта;

Destroy – дії при руйнуванні (видаленні) об'єкта;

ForwardTact – дії елемента під час такту прямого функціонування;

BackwardTact – дії елемента під час такту зворотного функціонування.

При описі методів ForwardTact і BackwardTact вони були залишені порожніми, оскільки функціонування окремих елементів НМ істотно різниться. Проте введення цих методів має досить глибокий сенс, оскільки клас TNetPiece є предком усіх інших класів, що описують елементи НМ,

і наявність типових процедур прямого і зворотного функціонування дозволяє використовувати такі властивості моделі об'єктно-орієнтованого програмування, як спадкоємність властивостей та методів і поліморфізм.

Для зв'язку НМ із задачником і передачі використовуються об'єкти класу TNetInput – вхідний елемент НМ. Цей клас є нащадком TNetPiece і тому успадковує його набір полів та методів цього класу, а крім того, додано поле SourceSignal, яке містить номер поля задачника, з якого даний вхід НМ збирає значення.

Методи ForwardTact і BackwardTact перекриті, тобто їх код замінено на той, який відповідає призначенню вхідного елемента.

Метод ForwardTact виконує передачу значення з відповідного цьому елемента поля задачника на вихідний сигнал елемента поля ForwOut. Метод BackwardTact передає подвійний сигнал наступного елемента на свій подвійний сигнал (поле BackOut).

Вихідний елемент мережі, що описує клас TNetOutput, також є нащадком TNetPiece.

У методах ForwardTact і BackwardTact закладено дії елемента при прямому і зворотному тактах функціонування.

Метод ForwardTact виконує передачу сигналу від виходу попереднього на вихід даного елемента. Крім того, в поле  $H$  заноситься значення помилки мережі при обчисленні функції  $Y$ .

Метод BackwardTact передає на зворотний вихід елемента (поле BackOut) значення подвійного сигналу. Подвійний сигнал  $X$  є похідною функції помилки за вихідним сигналом НМ:

$$\mu_h = \frac{\partial H}{Y_{\text{NM}}} = 2(Y_{\text{NM}} - F_{\text{пр}}),$$

де

$Y_{\text{NM}}$  – апроксимоване значення функції, видаване мережею;

$F_{\text{пр}}$  – значення функції, що апроксимується в цьому прикладі.

Для опису синапсів мережі використовуються об'єкти класу TNetSynapse. Як спадкоємець класу TNetPiece він успадковує всі його поля і методи. Крім цього до списку полів включено Alpha – параметр, вагу синапсу, що є, і MuAlpha – сигнал, подвійний вазі синапсу.

На такті прямого функціонування метод ForwardTact знімає вихідний сигнал попереднього елемента, множить його на вагу синапсу і передає на вихідний сигнал даного об'єкта (поле ForwOut).

На такті зворотного функціонування метод BackwardTact передає в полі BackOut подвійний сигнал синапсу, який може бути обчислений за формулою:

$$\mu_w = Y_{\text{back}} \frac{\partial W}{dx_w} = Y'_{\text{back}} a_w,$$

де

$Y_{\text{back}}$  – подвійний сигнал, передаваний синапсом;

$W$  – функція перетворення в синапсі;

$x_w$  – сигнал, що надходить у синапс від попереднього елемента на такті прямого функціонування;

$Y'_{\text{back}}$  – вхідний подвійний сигнал, що надходить у синапс від наступного елемента на такті зворотного функціонування;

$a_w$  – вага синапсу.

Крім того, на зворотному такті обчислюється сигнал, подвійний  $a_w$  (значення частотної похідної функції помилки). За цим параметром

$$\mu_w = Y_{\text{back}} \frac{\partial W}{da_w} = Y'_{\text{back}} x_w,$$

де

$\mu_w$  – сигнал, подвійний  $a_w$ .

Для значень  $a_w$ ,  $\mu_w$  в класі TNetSynapse передбачено поля Alpha і MuAlpha.

Програмою моделлю тривіального суматора є клас TSummator.

Крім полів, успадкованих від класу-предка TNetPiece, TSummator має в своїй структурі PriorPieces, що на відміну від стандартного поля PriorPiece є не покажчиком на попередній елемент, а списком покажчиків на набір таких елементів.

Метод ForwardTact здійснює підсумовування вихідних сигналів елементів із списку PriorPieces і поміщає отриманий результат в полі ForwOut.

На такті зворотного функціонування відбувається передача подвійного сигналу наступного елемента на подвійний сигнал суматора BackOut.

Під терміном «нейрон» мають на увазі нелінійний перетворювач вигляду

$$y = f(x_{\text{вх}} a_{\text{H}} a_{\text{c}}),$$

де

$y$  – вихідний сигнал перетворювача;

$x_{\text{вх}}$  – вхідний сигнал;

$a_{\text{H}}$  – параметр перетворювача;

$a_{\text{c}}$  – параметр спонтанної активності.

Нейрон описується в програмній моделі класом TNeuron. Вихідний сигнал на такті прямого функціонування заноситься в поле ForwOut.

Навчаними в нейроні є обидва параметри  $a_H$  і  $a_c$ , тому в класі TNeuron, крім полів Alpha і AlphaS, в яких зберігаються значення відповідних параметрів, передбачені MuAlpha і MuAlphaS, в яких поміщаються значення подвійних сигналів.

Крім цього, в поле BackOut заноситься сигнал, подвійний вхідному.

Об'єкт класу TNeuron характеризується й полем FunctionType, що являє собою номер функції перетворювача, що використовується в списку функцій, які використовуються при оптимізації.

Подвійні сигнали в нейронах обчислюються в загальному випадку за формулою

$$\mu_w = Y_{back} \frac{\partial f}{\partial a},$$

де

$a$  – параметр, для якого обчислюється подвійний сигнал.

Фрагмент мережі, який складається із шару вхідних синапсів, суматора, нейрона та вихідного синапсу, є «поток», поданим у програмній моделі класом TNetStream.

Крім звичайних для нащадка класу TNetPiece полів NextPiece, PriorPiece, ForwOut і BackOut, цей клас включає:

FirstLayer – список синапсів першого шару;  
Summator – об'єкт класу;

TSummator, що реалізує тривіальний суматор;

Neuron – об'єкт класу TNeuron, що реалізує нелінійний перетворювач – нейрон;

SecondLayer – об'єкт класу TSynapse, вихідний синапс потоку.

Прямий такт потоку, описаний методом ForwardTact, відбувається таким чином:

– перебираються елементи списку FirstLayer, для кожного з яких викликається його власний метод ForwardTact;

– відбувається «спрацювання» (виклик цього ж методу) для об'єктів Summator, Neuron і SecondLayer;

– вихідний сигнал об'єкта заноситься в поле ForwOut – вихідний сигнал всього потоку.

Такт зворотного функціонування потоку, описаний в методі BackwardTact, включає такі дії:

– послідовний виклик власного методу BackwardTact для об'єктів SecondLayerNeuron і Summator;

– перебір елементів списку FirstLayer;

– виклик методу BackwardTact для кожного синапсу, що входить в нього.

Структуру зв'язків між елементами потоку показано на рис. 1.

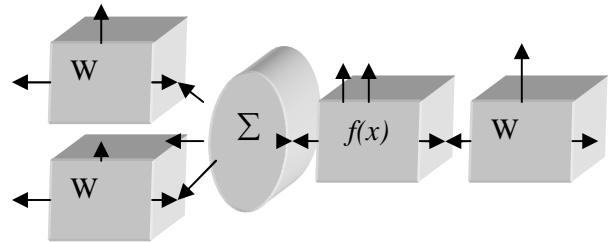


Рис. 1. Структура зв'язків типових елементів у потоці мережі

Цілу мережу подано класом TNet. Цей клас також є нащадком класу TNetPiece – загального елемента мережі, що дозволяє, використовуючи властивість поліморфізму об'єктів, досить легко компонувати з окремих фрагментів НМ більш складної структури.

Окрім полів, успадкованих від TNetPiece, мережа характеризується полями:

Inputs – список вхідних елементів;

Streams – список потоків мережі;

SecondLayer – суматор вхідного шару;

Answer – вихідний елемент мережі.

Для опису такту прямого функціонування мережі використовується, як і в інших елементах мережі, метод ForwardTact.

Методом ForwardTact виконуються такі дії:

– перебір списку вхідних елементів;

– виклик методу ForwardTact для кожного з них;

– перебір і пряме функціонування кожного з елементів списку потоків, потім для об'єктів SecondLayer і Answer.

Метод BackwardTact, що описує зворотне функціонування мережі, задається такою послідовністю дій:

– зворотне функціонування елементів Answer, SecondLayer;

– перебір і виконання методів BackwardTact для елементів списків Streams і Inputs.

Структуру зв'язків між стандартними елементами повністю скомп'юваної потокової НМ показано на рис. 2. Ця схема добре ілюструє переваги застосування об'єктно-орієнтовного підходу до моделювання НМ.

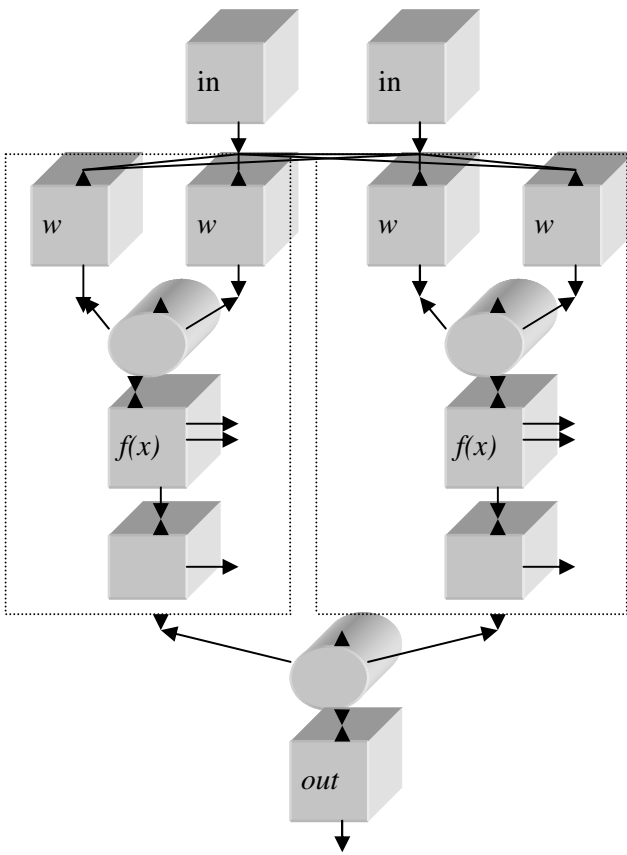


Рис. 2. Структура зв'язку типових елементів у поточній мережі

Верифікація НМ передбачала дослідження відомих попередніх перетворювань на відповідність вимогам стаціонарності та несуперечливості і співвідношення отриманих результатів з фактичною помилкою прогнозу контрольної вибірки.

Початковими даними є тимчасовий ряд статистичних даних з безпеки польотів за 10-річний період з інтервалом 15 діб. Параметри прогнозованої системи вибирались згідно з реальними статистичними даними виникнення АП:

- час опису ситуації: 24 год, що відповідає глибині занурення  $d = 96$ ;
- прогнозована величина: збільшення подій за 4000 год;
- вибірки: було сформовано 1464 наборів, 75 % з них (1098 наборів) становить навчаюча вибірка, 25 % (366 наборів) – контрольна вибірка;
- структура НМ: тришаровий перцептон з 80 нейронами в прихованому шарі;
- алгоритм навчання: класичний алгоритм зворотного розповсюдження помилки з моментом 0,5 та параметром швидкості навчання 0,1, навчання зупинялося з досягнення помилки НМ на навчальній вибірці 0,05 від первинної.

Досліджувались такі типи попередніх перетворювань:

- неперетворені значення (опис ситуацій являв собою 96 останніх значень часового ряду);
- значення прирощень (використовувались як опис ситуації не значення часових рядів, а різниця двох послідовних значень, названих як приростом);
- стрибки стабільності.

Ситуація описувалась у вигляді переміжків стабільності, де рівень безпеки польотів зростає, зменшувався, або залишався незмінним.

Кожний із таких переміжків характеризується парою значень прирощення – довготривалістю. Опис ситуації становив 14 таких переміжків.

В основі вейвлет-перетворення є вигляд

$$c_i = \int_{-\infty}^{\infty} x(t)\varphi_i(t)dt,$$

де

$C_i$  – отриманий коефіцієнт перетворення;

$\varphi_i$  – базис перетворення.

Отримані результати дослідження показано на рис. 3–5 (А – неперетворені значення, В – значення прирощень, С – переміжки стабільності, D – вейвлет-перетворення).

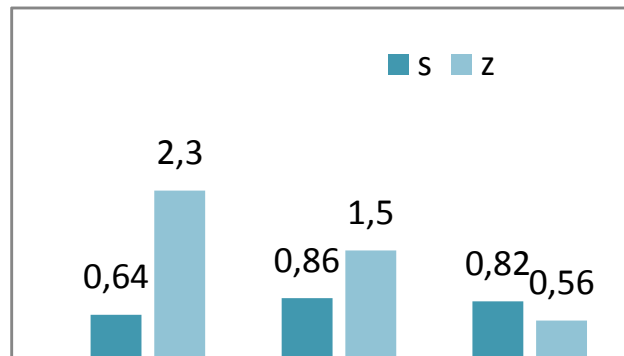


Рис. 3. Критерії виконання вимог:

s – стаціонарність;  
z – суперечливість

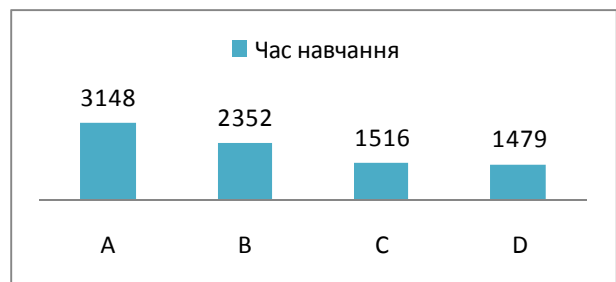


Рис. 4. Час ітерацій



Рис. 5. Середня помилка контрольної вибірки

Розроблена програмна реалізація штучної НМ АСКБП має важливу нову властивість – генерацію структури від простого до складного. Визначено ітераційний процес нарощування обсягу НМ. Визначення адекватності НМ передбачало дослідження відомих попередніх перетворювань на відповідність вимогам стаціонарності та несуперечності і співвідношення отриманих результатів з фактичною помилкою прогнозу контрольної вибірки.

На стаціонарних та несуперечних даних були отримані менші значення часу навчання НМ та помилки прогнозу щодо контрольної вибірки. На нестаціонарних та суперечливих даних після більшої кількості циклів навчання, помилка контрольної вибірки була велика.

Отримані дані підтвердили відповідність сформульованих критеріїв на практиці.

### Висновки

Виконано експеримент з розв'язання задач ідентифікації багатовимірних об'єктів і керування за допомогою ієрархічної НМ і з'ясовано, що застосування ієрархічної НМ дозволяє значно скоротити об'єм необхідної пам'яті для реалізації НМ, що, в свою чергу, дозволяє розширити галузь застосування НМ та робить можливим її використання для розв'язання задач ідентифікації і керування багатомірними об'єктами.

Нейронно-мережева модель автоматизованого КБП дозволить ефективно розв'язати задачу синтезу виникнення ризику АП та видати вектор керувальних сигналів мережі за неповної і спотвореної інформації про явища, події та процеси, що впливають на БП.

Необхідно лише визначити перелік факторів, які впливають на прогнозну величину, та підібрати достатню кількість прикладів які описують поведінку цих величин в минулому.

Нейронна мережа сама настроїться на задану сукупність прикладів, зводячи до мінімуму сумарну помилку прогнозування.

Аналіз настроєної НМ дозволяє виявляти скриті залежності між вхідними і вихідними даними, що не можливо зробити традиційними методами. Передбачаючи, що характер взаємозв'язку між заданими параметрами ще деякий час не зміниться, експерт може використовувати настроєну та навчену НМ для короткочасного/довгочасного прогнозування та напрацювання керувальних рішень.

### Література

1. *Гузій А.Г.* Методология активного управления уровнем безопасности предстоящих полетов в авиакомпании / А.Г. Гузій, В.В. Онуфриенко // Тр. Общества независимых расследователей авиационных происшествий. – М., 2005. – Вып. 17. – С. 52–62.
2. *Харченко В.П.* Інтеграція засобів діагностики, прогнозування і управління рівнем безпеки польотів / В.П. Харченко, А.Є. Луппо, О.М. Алексєєв // Вісник НАУ. – 2007. – № 2 (32). – С. 43–49.
3. *Kharchenko V.P.* Some aspect automation safety flight system in civil aviation of Ukraine / V.P. Kharchenko, O.N. Alexeiev // Proc. of the third World Congress "Aviation in the XXI – st century". – 2008. – P. 31.22–31.25.
4. *Осташкевич В.А.* Общий подход к количественной оценке безопасности полетов / В.А. Осташкевич // Проблемы безопасности полетов: информ. сб. – М.: ВИНТИ, 2006. – № 9. – С. 45.
5. *Прозоров С.Е.* Безопасность полетов. Ч. III. Оценка риска и управление безопасностью полетов / С.Е. Прозоров. – М.: МГТУ ГА, 2008. – 88 с.
6. *Харченко В.П.* Некоторые аспекты установления системы менеджмента безопасности в гражданской авиации Украины / В.П. Харченко, А.Е. Луппо, О.Н. Алексєєв // Проблеми інформатизації та управління: зб. наук. пр. – К., 2007. – Вип 1(19). – С. 148–153.
7. *Энциклопедия безопасности авиации* / [Н.С. Кулик, В.П. Харченко, М.Г. Луцкий и др.], под ред. Н.С. Кулика. – К.: Техніка, 2008. – 1000 с.
8. *Орлик С.* Секреты Delphi на примерах / С. Орлик. – М.: Бином, 1996. – 316 с.
9. *Lane S.H.* Theory and development of higher-order CMAC neural networks / S.H. Lane, D.A. Handelman, J.J. Gelfand // IEEE Control System. – 1992. – Vol. 12, № 2. – P. 23–30.
10. *Artificial Neural Networks: Concept & Theory* // IEEE Computer Society Press. – 1992. – P. 249.