

**В. В. Костенко**, старший викладач кафедри інформаційних систем та технологій Академії митної служби України  
**І. В. Лавренюк**, старший викладач кафедри інформаційних систем та технологій Академії митної служби України  
**В. М. Пономарьов**, старший викладач кафедри інформаційних систем та технологій Академії митної служби України

### АНАЛІЗ АСПЕКТІВ ПРОЕКТУВАННЯ САМОНАВЧАЛЬНОЇ ПРОГРАМИ НА ОСНОВІ НЕЙРОННИХ МЕРЕЖ

*Проведено дослідження існуючих нейронних мереж з метою побудови самонавчальної програми на прикладі гри в шашки.*

*Проведено исследование существующих нейронных сетей с целью построения самообучающей программы на примере игры в шашки.*

*Research of existent neuron networks is conducted with the purpose of construction of the independently student program on the example of playing checkers.*

**Ключові слова.** Самонавчальні інтелектуальні системи, бази знань, нейрон, штучний нейрон, нейронні мережі, прецеденти, штучний інтелект, активаційні функції, UML-діаграми, клас.

**Вступ.** Неможливо виконувати дуже великий обсяг обчислень, дотримуючись нинішньої ідеології програмування. Переважно це стосується завдань розпізнавання образів у загальній постановці аналізу, прогнозування, оптимізації тощо. У природі існують принципово інші обчислювальні системи та алгоритми вирішення таких проблем, адже всі живі організми дуже швидко й ефективно адаптуються до таких ситуацій. Проте сучасні комп'ютери вимагають великого обсягу обчислень, щоб упоратися з такого роду завданнями.

Самонавчальні інтелектуальні системи базуються на методах автоматичної класифікації ситуацій з реальної практики (навчання на прикладах). Саме приклади реальних ситуацій становлять так звану навчальну вибірку, яка формується протягом певного історичного періоду. Її елементи характеризуються множиною класифікаційних ознак. Стратегія “навчання з учителем” передбачає, що для кожного прикладу фахівець задає значення ознак, які показують його належність до певного класу ситуацій. У разі навчання “без учителя” система має самостійно виділяти класи ситуацій за ступенем близькості значень класифікаційних ознак.

Під час навчання автоматично створюються загальні правила чи функції, що описують належність ситуацій до класів, які система згодом використовуватиме для інтерпретації незнайомих ситуацій. У свою чергу, загальні правила автоматично формують базу знань, яка періодично коригується в міру накопичення інформації про аналізовані ситуації.

© В. В. Костенко, І. В. Лавренюк, В. М. Пономарьов, 2013

---

Побудовані відповідно до цих принципів самонавчальні системи мають такі недоліки:

- відносно низьку адекватність баз знань реальним проблемам через неповноту та/або зашумленість навчальної вибірки;
- низький ступінь з'ясування отриманих результатів;
- поверховий опис проблемної ділянки, вузьку спрямованість застосування через обмеження в розмірності простору ознак.

Індуктивні системи дозволяють узагальнювати приклади на основі принципу індукції “від окремого до загального”. Так здійснюється класифікація за значущими ознаками. Алгоритм класифікації прикладів”.

1. Вибір класифікаційної ознаки з множини заданих.
2. Розбиття множини прикладів на підмножини за значенням вибраної ознаки.
3. Перевірка належності до кожної підмножини прикладів одного з класів.
4. Перевірка закінчення процесу класифікації. Якщо якась підмножина прикладів належить одному підкласу, тобто у всіх прикладів цієї підмножини збігається значення класифікаційної ознаки, то процес класифікації закінчується [1].

Для підмножин прикладів, що мають незбіжні значення класифікаційних ознак, процес розпізнавання триває, починаючи з першого кроку. При цьому кожна підмножина прикладів стає класифікованою множиною.

Нейронні мережі – класичний приклад технології, що ґрунтується на прикладах. Це узагальнена назва групи математичних алгоритмів, що мають здатність навчатися на прикладах, “упізнаючи” згодом риси зразків і ситуацій, які були раніше. Завдяки цій здатності нейронні мережі використовуються для обробки сигналів та зображень, розпізнавання образів, а також для прогнозування.

Нейронна мережа – це кібернетична модель нервової системи, що являє собою сукупність великої кількості порівняно простих елементів – нейронів, топологія з'єднання яких залежить від типу мережі. Щоб створити нейронну мережу для якогось конкретного завдання, слід обрати спосіб з'єднання нейронів один з одним і підібрати значення параметрів міжнейронних сполук.

У системах, що базуються на прецедентах, база знань містить описи конкретних ситуацій (прецеденти). Пошук розв'язку здійснюється на основі аналогій і включає такі етапи:

- отримання інформації про поточну проблему;
- зіставлення отриманої інформації зі значеннями ознак прецедентів з бази знань;
- вибір прецеденту з бази знань, найбільш близького до розглянутої проблеми;
- адаптація обраного прецеденту до поточної проблеми;
- перевірка коректності кожного отриманого рішення;
- занесення детальної інформації про отримане рішення в базу знань.

Прецеденти описуються множиною ознак, за якими будуються індекси швидкого пошуку. Однак у системах, що базуються на прецедентах, на відміну від індуктивних систем допускається нечіткий пошук з отриманням множини допустимих альтернатив, кожна з яких оцінюється певним коефіцієнтом упевненості. Найефективніші рішення адаптуються до реальних ситуацій за допомогою спеціальних алгоритмів.

Системи, що ґрунтуються на прецедентах, застосовуються для поширення знань і в системах контекстної допомоги.

**Постановка завдання.** Розробка інтелектуальної самонавчальної системи обробки даних на основі нейронних мереж, на прикладі гри в шашки.

Для виконання поставленого завдання потрібно визначитися з типом нейронної мережі, що буде використовуватись. Кількість нейронів визначають самостійно. Навчання мережі проводиться за допомогою генетичного алгоритму.

---

Штучний інтелект детермінуємо як наукову дисципліну, що моделює розумну поведінку. Це визначення має один істотний недолік – поняття інтелекту важко пояснити. Більшість людей упевнені, що зможуть відрізнити “розумну поведінку”, коли з нею зустрінуться. Однак навряд чи хтось зможе дати визначення інтелекту, досить конкретне для оцінки імовірно розумної комп’ютерної програми, що одночасно відображає життєздатність і складність людського розуму.

Проблеми визначення штучного інтелекту допомогли сформулювати завдання і методологію, які становлять його основу. Частково привабливість штучного інтелекту в тому й полягає, що він є оригінальною і потужною зброєю для дослідження цих проблем. Штучний інтелект дає засіб і дослідну модель для теорій інтелекту: ці теорії можна сформулювати мовою комп’ютерних програм, а потім випробувати.

Визначення штучного інтелекту, наведене на початку, не дає однозначної характеристики для цієї галузі науки. Воно лише ставить нові питання і відкриває парадокси в царині, одним з головних завдань якої є пошук самовизначення. Однак проблема пошуку точного визначення штучного інтелекту цілком з’ясовна. Штучний інтелект слугує для розширення можливостей комп’ютерних наук, а не визначення їх меж. Одне з важливих завдань дослідників – підтримка цих зусиль ясними теоретичними принципами.

Будь-яка наука, включаючи штучний інтелект, розглядає певне коло проблем і розробляє підходи до їх розв’язання. Нині штучний інтелект застосовується в таких галузях:

- автономне планування та складання розкладів;
- ведення ігор;
- автономне управління;
- діагностика;
- планування постачання;
- робототехніка;
- розуміння природної мови і виконання завдань.

Методи штучного інтелекту необхідні для створення самонавчальних систем [1]. Самонавчальні інтелектуальні інформаційні системи (ІС) ґрунтуються на методиках автоматичної класифікації ситуацій з реальної практики, або на методах навчання на прикладах. Приклади реальних ситуацій становлять так звану навчальну вибірку. Її елементи мають множину класифікаційних ознак.

Об’єкт проектування – інтелектуальна самонавчальна система на основі нейронних мереж, яка розв’язує задачу гри в шашки.

Штучний нейрон імітує в першому наближенні властивості біологічного нейрона. На вхід штучного нейрона надходить деяка множина сигналів, кожен з яких є виходом іншого нейрона. Кожен вхід множиться на відповідну вагу, аналогічну синаптичній силі, і всі добутки додаються, визначаючи рівень активації нейрона. На рис. 1 подано модель, що реалізує цю ідею. Хоча мережні парадигми досить різноманітні, в основі майже всіх їх лежить ця конфігурація. Тут множина вхідних сигналів, позначених  $x_1, x_2, \dots, x_n$ , надходить на штучний нейрон. Ці вхідні сигнали, які в сукупності позначаються вектором  $X$ , відповідають сигналам, що приходять у синапси біологічного нейрона. Кожен сигнал множиться на відповідну вагу  $w_1, w_2, \dots, w_n$  і надходить на підсумовувальний блок, позначений  $\Sigma$ . Кожна вага відповідає “силі” одного біологічного синаптичного зв’язку. (Сукупність вагових коефіцієнтів позначається вектором  $W$ ). Підсумовувальний блок, що відповідає тілу біологічного елемента, додає зважені входи алгебраїчно, створюючи вихід, який ми будемо називати  $NET$ . У векторних позначеннях це може бути компактно записано таким чином:

$$NET = XW \quad (1)$$

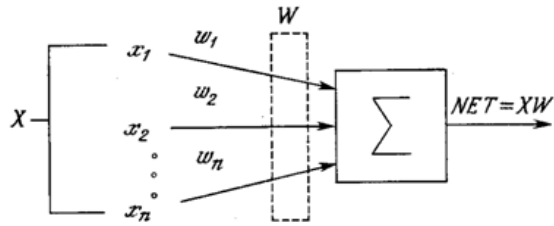


Рис. 1. Штучний нейрон

### Активаци́йні функції

Сигнал  $NET$  далі, як правило, перетворюється активаційною функцією  $F$  і дає вихідний нейронний сигнал  $OUT$ . Активаційна функція може бути звичайною лінійною функцією

$$OUT = K (NET), \quad (2)$$

де  $K$  – постійна порогової функції;

$$OUT = 1, \text{ якщо } NET > T, \quad (3)$$

$$OUT = 0 \text{ в інших випадках,} \quad (4)$$

де  $T$  – деяка постійна порогова величина, або ж функція, що точніше моделює нелінійну передавальну характеристику біологічного нейрона і надає нейронній мережі великі можливості.

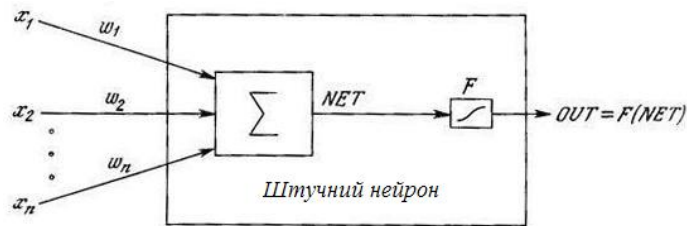


Рис. 2. Штучний нейрон з активаційною функцією

На рис. 2 блок, позначений  $F$ , приймає сигнал  $NET$  і видає сигнал  $OUT$ . Якщо блок  $F$  звужує діапазон зміни величини  $NET$  так, що за будь-яких значень  $NET$  значення  $OUT$  належать деякому кінцевому інтервалу, то  $F$  називається “стискною” функцією. Як “стискна” функція часто використовується логістична або “сигмоїдальна” ( $S$ -подібна) функція, що показана на рис. 3. Ця функція математично виражається як  $F(x) = 1/(1+e^{-x})$ . Таким чином:

$$OUT = \frac{1}{1+e^{-NET}}. \quad (5)$$

За аналогією з електронними системами активаційну функцію можна вважати нелінійною підсилювальною характеристикою штучного нейрона. Коефіцієнт посилення обчислюється як відношення приросту величини  $OUT$  до невеликого збільшення величини  $NET$ , яке зумовило його. Він виражається нахилом кривої за певного рівня збудження і

змінюється від малих значень, якщо негативні збудження великі (крива майже горизонтальна), до максимальних та при нульовому збудженні та знову зменшується, коли збудження стає великим позитивним. Гросберг (1973) виявив, що така нелінійна характеристика вирішує дилему шумового насичення. Яким чином одна і та ж мережа може обробляти як слабкі, так і сильні сигнали? Слабкі сигнали потребують великого мережного посилення, щоб дати придатний до використання вихідний сигнал. Однак підсилювальні каскади з великими коефіцієнтами посилення можуть призвести до насичення виходу шумами підсилювачів (випадковими флуктуаціями), наявними в будь-якій фізично реалізованій мережі. Сильні вхідні сигнали, у свою чергу, також зумовляють насичення підсилювальних каскадів, виключаючи можливість корисного використання виходу. Центральна область логістичної функції має великий коефіцієнт посилення і сприяє подоланню проблеми обробки слабких сигналів, водночас області зі спадним посиленням на позитивному і негативному кінцях підходять для великих збуджень. Таким чином, нейрон функціонує з великим посиленням у широкому діапазоні рівня вхідного сигналу.

$$OUT = \frac{1}{1 + e^{-NET}} = F(NET). \quad (6)$$

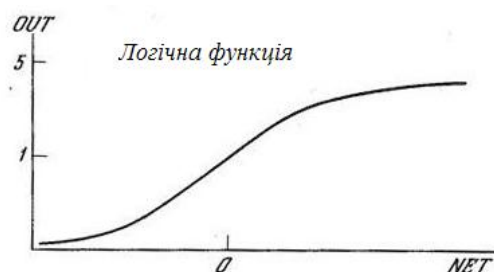


Рис. 3. Сигмоїдальна логістична функція

Широко використовується й інша активаційна функція – гіперболічний тангенс. За формою вона схожа на логістичну, її часто використовують біологи як математичну модель активації нервової клітини. Як активаційна функція штучної нейронної мережі вона записується таким чином:

$$OUT = th(x). \quad (7)$$

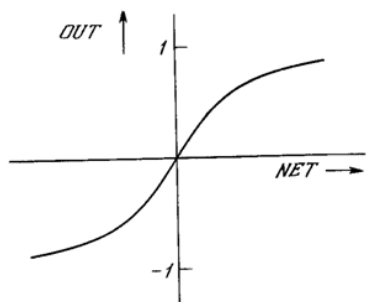


Рис. 4. Функція гіперболічного тангенса

---

Подібно до логістичної функції гіперболічний тангенс – це S-подібна функція, проте він симетричний відносно початку координат, і в точці  $NET = 0$  значення вихідного сигналу  $OUT$  дорівнює нулю (рис. 4). На відміну від логістичної функції, гіперболічний тангенс набирає значення різних знаків, що вигідно для багатьох мереж.

Розглянута проста модель штучного нейрона ігнорує багато властивостей свого біологічного двійника. Наприклад, вона не бере до уваги затримки в часі, які впливають на динаміку системи. Вхідні сигнали відразу ж породжують вихідний сигнал. І, що важливіше, вона не враховує вплив функції частотної модуляції або синхронізаційної функції біологічного нейрона, які низка дослідників вважає вирішальними.

Незважаючи на ці обмеження, мережі, побудовані з цих нейронів, виявляють властивості, які дуже нагадують біологічну систему. Тільки час і дослідження зможуть відповісти на питання, чи подібні збіги випадкові, чи це наслідок того, що в моделі правильно схоплені найважливіші риси біологічного нейрона [2].

#### **Переваги нейронних мереж**

Цілком очевидно, що свою силу нейронні мережі черпають, по-перше, з розпаралелювання обробки інформації і, по-друге, їхньої властивості самонавчатися, тобто створювати узагальнення. Термін “узагальнення” розуміємо як здатність отримувати обґрунтований результат із даних, які не зустрічалися в процесі навчання. Ці особливості дозволяють їм виконувати складні (масштабні) завдання. Однак на практиці при автономній роботі нейронні мережі не можуть забезпечити готові рішення. Їх потрібно інтегрувати в складні системи. Зокрема, комплексну задачу можна розбити на послідовність простих, частина з яких може розв’язуватися нейронними мережами.

Отже, наведемо деякі переваги нейронних мереж перед традиційними обчислювальними системами.

#### **1. Розв’язування задач із невідомими закономірностями.**

Використовуючи здатність навчання на множині прикладів, нейронна мережа спроможна виконувати завдання, в яких невідомі закономірності розвитку ситуації та залежності між вхідними й вихідними даними. Традиційні математичні методи та експертні системи в таких випадках непридатні.

#### **2. Стійкість до шумів у вхідних даних.**

Можливість роботи за наявності великої кількості неінформативних шумових вхідних сигналів. Немає необхідності робити їх попередній відсів, нейронна мережа сама визначить їх малоприслужними для виконання завдання і відкине їх.

#### **3. Адаптування до змін навколишнього середовища.**

Нейронні мережі мають здатність адаптуватися до змін навколишнього середовища. Зокрема, нейронні мережі, навчені діяти в певному середовищі, можна легко перевчити для роботи в умовах незначних коливань параметрів середовища. Більш того, для роботи в нестационарному середовищі (де статистика змінюється з часом) можуть створюватися нейронні мережі, здатні перевчатися в реальному часі. Чим вища адаптивна здатність системи, тим стійкіша її робота в нестационарному середовищі. При цьому, слід зауважити, що адаптивність не завжди веде до стійкості, іноді вона призводить до зовсім протилежного результату. Наприклад, адаптивна система з параметрами, які швидко змінюються в часі, може також швидко реагувати і на сторонні збудження, що викличе втрату продуктивності. Для використання всіх переваг адаптивності слід мати стабільні та досить гнучкі основні параметри системи. Це дозволить не враховувати зовнішні перешкоди, а також забезпечить реакцію на суттєві зміни середовища.

#### **4. Потенційна надвисока швидкодія.**

Нейронні мережі мають потенційну надвисоку швидкодію завдяки масовому паралелізму обробки інформації [3].

---

Нині існує багато програм, що використовують самонавчальні системи. Наприклад, Remy, Alchemy Spam Filter for Outlook Express тощо.

Більша частина інформації в Інтернеті передається за допомогою протоколу TCP-IP, який, хоча й розроблений досить давно, постійно вдосконалюється й успішно працює і зараз. Однак протокол, створений в “доісторичні” для Інтернету часи, що працює за рахунок обмеженого набору правил, не завжди може впоратися з покладеним на нього завданням. У деяких випадках затримка в отриманні інформації може становити досить значний час.

Дослідники Массачусетського технологічного інституту повідомляють, що їхня самонавчальна програма Remy дозволяє прокласти маршрути проходження пакетів таким чином, щоб уникнути “вузькі місця”, збільшивши швидкість доставки пакетів у 2–3 рази. На жаль, програма має низку недоліків. Найбільший з них – швидкість обробки інформації. Підбір оптимальних параметрів і створення правил може тривати від 4 до 12 годин.

Alchemy Spam Filter є додатковим програмним модулем для програми Outlook Express та самонавчальним захисним засобом. Ґрунтуючись на самонавчальній технології, він ефективніший, ніж традиційні антиспам-засоби, які базуються на перевірці статичних “чорних списків” і баз даних з ключовими словами. Але при використанні даної програми виникає ймовірність неправильного навчання нейронної мережі. У такому випадку вона буде переміщувати в категорію “спам” звичайні листи.

Нейронні мережі Кохонена – клас нейронних мереж, основним елементом яких є прошарок Кохонена. Прошарок Кохонена складається з адаптивних лінійних суматорів (“лінійних формальних нейронів”). Як правило, вихідні сигнали прошарку Кохонена обробляються за правилом “переможець забирає все”: найбільший сигнал перетворюється на одиничний, інші – на нуль. Саме тому для виконання поставленого завдання ми обрали даний тип нейронної мережі.

Шар Кохонена складається з деякої кількості  $n$  паралельно діючих лінійних елементів. Усі вони мають однакову кількість входів, отримують один і той же вектор вхідних сигналів  $x = (x_1, \dots, x_m)$ . На виході  $j$ -го лінійного елемента отримуємо сигнал:

$$y_j = w_{j0} + \sum_{i=1}^m w_{ji} x_i, \quad (8)$$

де  $w_{ji}$  – ваговий коефіцієнт  $i$ -го входу  $j$ -го нейрона;  
 $w_{j0}$  – пороговий коефіцієнт.

Після проходження шару лінійних елементів сигнали посилаються на обробку за правилом “переможець забирає все”: серед вихідних сигналів шукається максимальний, його номер  $y_{\max} = \arg \max_j y_j$ . Остаточо, на виході, сигнал з номером  $j_{\max}$  дорівнює одиниці, інші – нулю. Якщо максимум одночасно досягається для декількох  $j_{\max}$ , то або беруть усі відповідні сигнали рівними одиниці, або тільки перший у списку (за згодою).

Зберігання інформації про місце розташування шашок відбувається у вигляді двох квадратних матриць  $8 \times 8$ . На одній з них позначається розташування білих шашок, на іншій – чорних. Під час проведення дамки створюються ще дві квадратні матриці для позначання місця розташування білих та чорних дамк. Після того як гравець зробив свій хід, змінені матриці місця розташування шашок і дамк передаються на входи 24 або більшої кількості нейронів (залежно від кількості дамк на дошці). Ваги входів у кожного нейрона зберігаються в окремому каталозі у файлах даних. Далі відбувається підсумування добутку вхідного сигналу та його ваги, і ця сума подається на вихід нейрона. За правилом “переможець забирає все” на вихід нейрона з найбільшою сумою подається одиниця, на інші – нулі. Кожен нейрон відповідає за окремий хід окремої шашки. Після того як хід обрано, змінюються матриці місця розташування шашок і

дамок; зміни відображаються на екрані. Під час гри в окрему змінну записуються нейрони, які спрацювали впродовж гри, і кількість разів, коли вони спрацювали. Якщо результатом гри є поразка штучного інтелекту, то ваги усіх входів на нейронах, які спрацювали, зменшуються на певну величину, що пропорційна до кількості спрацювань даного нейрона. Якщо штучний інтелект виграв, то ваги збільшуються.

Початкове навчання нейронної мережі проводиться за генетичним алгоритмом. Створюються два екземпляри всіх класів, необхідних для роботи нейронної мережі. У базі знань, де зберігаються ваги входів нейронів цих двох екземплярів, випадковим чином створюються відмінності. Після перемоги однієї з двох нейронних мереж на основі бази знань цієї мережі знову створюються дві різні бази з випадковими відмінностями.

Структура проектування даної системи включає 4 етапи.

1) Створення ТЗ.

2) Синтез принципу дії. На цьому етапі знаходимо принципові положення, які становлять основу функціонування майбутньої системи.

На рис. 5 зображено UML-діаграму класу, що являє собою структуру нейрона. Розглянемо його детальніше.

Length – довжина ходу, за який відповідає саме цей нейрон. Може коливатись від 1 до 7.

Place\_white – бінарний масив, в якому міститься інформація про розташування білих шашок.

Place\_white\_mother – бінарний масив, в якому міститься інформація про розташування білих дамк.

Place\_black – бінарний масив з інформацією про розташування чорних шашок.

Place\_black\_mother – бінарний масив з інформацією про розташування чорних дамк.

Link\_white, Link\_white\_mother, Link\_black, Link\_black\_mother – масиви з інформацією про вагові коефіцієнти.

Weight() – метод, який дозволяє обчислити вихідний сигнал нейрона.

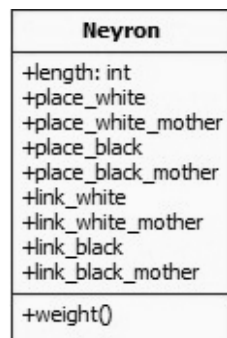


Рис. 5. Клас структури нейрона

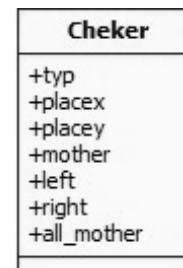


Рис. 6. Клас структури шашки

На рис. 6 зображено UML-діаграму класу, який являє собою структуру шашки. Він складається з таких полів:

Typ – поле, яке несе інформацію про колір шашки;

Placex – розташування шашки на шаховій дошці по осі X;

Placey – розташування шашки на шаховій дошці по осі Y;

Сума змінних placex і placey може бути лише парною, оскільки шашки ходять тільки по чорних клітинках.

Mother – відображає, чи є шашка дамкою.

Left – клас нейронів, який відповідає за хід ліворуч.

Right – клас нейронів, який відповідає за хід праворуч.



---

All\_mother – масив нейронів, який створюється, коли шашка стає дамкою та відповідає за всі можливі ходи.

3) Структурний синтез. На цьому етапі створюємо об'єкти, алгоритми, різні варіанти реалізації, а саме варіанти реалізації з різною кількістю нейронів.

4) Параметричний синтез. Знаходимо параметри системи, створюємо оптимальну реалізацію системи відповідно до поставленої задачі. Визначаємо оптимальне число нейронів.

Нейронна мережа, яка використовується в проєктованій системі, складається з 24 нейронів. Цей вибір обумовлений кількістю ходів, доступних кожному гравцеві на початку. За необхідності будуть підключатися нейрони для забезпечення оцінки ходів дамоч.

Відображення графіки та об'ємної ігрової дошки здійснюватимемо за допомогою відкритої бібліотеки OpenGL.

#### Математична модель задачі

Кожен нейрон на всіх своїх входах отримує чотири бінарні матриці  $8 \times 8$ . Отже, кожен нейрон має 256 входів. Для зручності подамо їх як одну матрицю  $A$  з розмірністю  $16 \times 16$ . У базі знань для кожного входу нейрона зберігається ваговий коефіцієнт. Далі вираховується вихідний сигнал нейрона.

Він дорівнює сумі добутків вхідного сигналу на відповідний ваговий коефіцієнт.

$$Y = \sum_{i=0}^{16} \sum_{j=0}^{16} x_{ij} \times w_{ij} . \quad (9)$$

Далі вихідні сигнали з усіх нейронів порівнюються, і максимальний сигнал перетворюється в одиницю, а всі інші – в нулі (рис. 7).

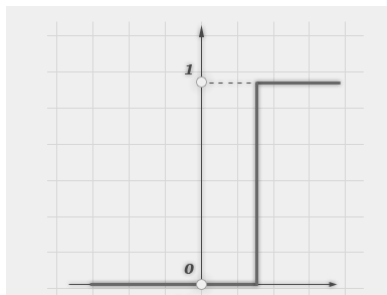


Рис. 7. Жорстка порогова функція

Після цього залежно від активованого нейрона відбувається хід відповідної шашки і очікується наступний хід користувача.

**Висновки.** Проведений аналіз нейронних мереж дає можливість визначити їх переваги під час створення самонавчальних програм:

1) доцільність та необхідність застосування самонавчальних систем у системах управління, планування та діагностики;

2) при грамотному навчанні та використанні таких систем можливе значне збільшення продуктивності, швидкодії та відмовостійкості програм, що розробляються.

#### Література

1. <http://www.zubolom.ru/lectures/iis/index.shtml>.
2. <http://www.neuronets.chat.ru/foundations.html>.
3. <http://www.aiportal.ru>.