

Для реалізації іншого методу структурної ідентифікації на основі генетичного алгоритму потрібно виконати наступні кроки: Крок 1. Випадкове генерування хромосом. Крок 2. Селекція кращих особин популяції. Крок 3. Схрещування відібраних особин популяції, що здійснюється випадковим чином із застосуванням оператора схрещування, побудованого за принципом «розіграшу лотереї». Повернення на крок 2 поки не буде знайдено «субоптимальну» структуру IPO.

Вхідні параметри алгоритму: S (де $2S$ – розмір популяції), $[I_{\min}; I_{\max}]$ та множина F .

Із праці [2] видно, що найскладнішою процедурою в алгоритмі структурної ідентифікації є процедура обчислення показника якості структури $\delta(\lambda_s)$. Тому, ефективність алгоритмів структурної ідентифікації будемо визначити на основі кількості обчислень значень показника $\delta(\lambda_s)$, який визначає цільову функцію у задачі структурної ідентифікації у виразі (5).

Порівняльний аналіз вищезазначених методів на задачі структурної ідентифікації макромоделі розподілу вологості на поверхні листа гіпсокартону показав, що метод структурної ідентифікації на основі АБК з точки зору часової складності є більш простішим. У наслідок того, що у методі структурної ідентифікації на основі ГА, генерування нових моделей-претендентів відбувається на основі принципу «розіграшу лотереї», що передбачає генерування двох моделей-претендентів для двох відібраних випадковим чином структур із популяції не залежно від значень показника $\delta(\lambda_s)$. На противагу цьому у методі структурної ідентифікації на основі АБК генерування моделей-претендентів відбувається шляхом розподілу їх загальної кількості між уже існуючими структурами IPO прямо пропорційно до значення показника $\delta(\lambda_s)$. Що фактично означає, що на основі кожної з «кращих» структур з точки зору значення показника $\delta(\lambda_s)$ буде згенеровано більше однієї моделі-претендента, а на основі найгіршої — жодної. Крім того, метод структурної ідентифікації на основі АБК забезпечує більше покриття області розв'язків.

Висновки

Розглянуто задачу структурної ідентифікації інтервального різницевого оператора. Проведено порівняльний аналіз ефективності існуючих методів структурної ідентифікації IPO. У результаті проведеного аналізу вперше встановлено, що значення показника часової складності алгоритму структурної ідентифікації IPO на основі АБК на 20% менше, ніж на основі ГА і при цьому порівняльна ефективність алгоритму бджолиної колонії зростає при збільшенні розмірності задачі.

Список використаних джерел

1. Войтюк І. Ф., Метод та генетичний алгоритм структурної ідентифікації інтервальних різницевих операторів в задачах екологічного моніторингу / І. Ф. Войтюк, М. П. Дивак, В. М. Неміш // Збірник наукових праць Донецького національного технічного університету серії „Інформатика, кібернетика та обчислювальна техніка“. – 2011. - Вип. 14 (188). - С. 8-17.
2. Войтюк І. Ф. Особливості оптимізації структури інтервального різницевого оператора / Войтюк І. Ф., Манжула В. І., Дивак Т. М. // Прогресивні інформаційні технології в науці, освіті та економіці. Збірка наукових праць учасників міжнародної науково-практичної конференції „Трансформаційні реформи та антикризовий потенціал економіки в постсоціалістичних країнах“. – Вінниця, 2009. – С. 146-154.
3. Дивак М. П. Кількісні характеристики оцінки якості структури моделі у вигляді інтервального різницевого оператора / М. П. Дивак, Т. М. Дивак, І. Ф. Войтюк // Відбір і обробка інформації : міжвід. зб. наук. пр. – Вип. 34 (110). – 2011. – С. 86-94.

УДК 004

ОСОБЛИВОСТІ МОДЕЛЮВАННЯ ШТУЧНОЇ НЕЙРОННОЇ МЕРЕЖІ ЗАСОБАМИ ФУНКЦІОНАЛЬНОЇ МОВИ ПРОГРАМУВАННЯ ERLANG

Проць І.В.

Дніпропетровський національний університет імені Олеся Гончара, аспірант

І. Постановка проблеми

При моделюванні штучної нейронної мережі засобами мови програмування Erlang виникає багато питань щодо реалізації моделі: склад та розподіл функцій між елементами, синхронізація окремих складових моделі тощо.

II. Мета роботи

Метою дослідження є пошук оптимального складу моделі та розподілу функцій між елементами штучної нейронної мережі.

III. Основна частина

Стандартна штучна нейронна мережа являє собою направлений граф поєднаних між собою нейронів, в якому кожен нейрон може надсилати та отримувати сигнали від інших нейронів і/або сенсорів чи силових приводів. Для моделювання оберемо найпростішу мережу прямого розповсюдження сигналу (Feed Forward Neural Network), складові мережі показані на рисунку 1, відповідно до [1].

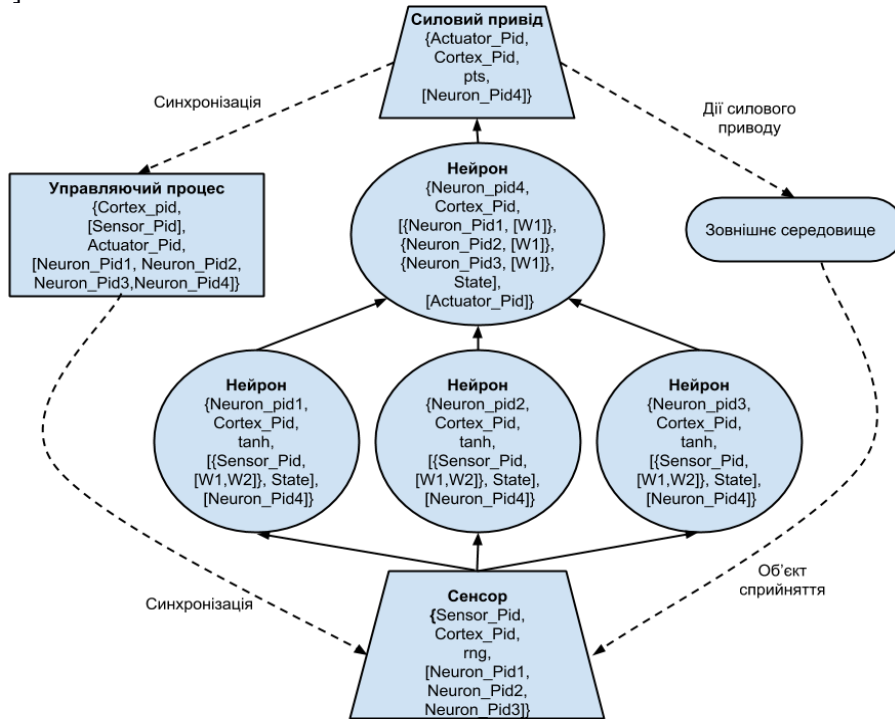


Рисунок 1 - Модель FFNN, реалізована інструментами мови програмування Erlang

В контексті функціональної мови програмування Erlang та віртуальної машини BEAM кожна складова штучної нейронної мережі являє собою окремий процес, який спілкується з іншими за допомогою системи повідомлень, у відповідності до моделі акторів. Кожен процес зберігає набір параметрів (process state), за допомогою яких здійснює обмін інформацією з іншими учасниками.

Нейрон - основна функціональна складова нейронної мережі, процес віртуальної машини, який виконує обробку вхідного сигналу, має зберігати у своєму "стейті" ідентифікатори процесів управляючого модуля, сенсора або силового приводу та всіх нейронів, що поєднані з ним (Cortex_Pid, Sensor_Pid, Actuator_Pid, [Neuron_Pid1, Neuron_Pid2, ..., Neuron_PidN]). Також модуль нейрону має описувати передаточну функцію, початковий стан та мати список вхідних ваг ([W1, W2, ..., Wn]). Основна функція даного модуля - обробка вхідного вектору сигналів і продукування вихідного сигналу [1].

Сенсор - є входом для штучної нейронної мережі, моделюється процесом віртуальної машини, який зберігає у своєму "стейті" ідентифікатори процесів управляючого модуля та всіх нейронів першого вхідного шару, що приєднаний до нього безпосередньо (Cortex_Pid, [Neuron_Pid1, Neuron_Pid2, Neuron_Pid3]). Основна задача процесу - збір даних з навколишнього середовища, та передача до всіх нейронів першого шару віртуальної машини, який зберігає у своєму "стейті" ідентифікатори процесів управляючого модуля та всіх нейронів останнього вихідного шару (Cortex_Pid, [Neuron_Pid4]). Процес приймає вхідні повідомлення від нейронів останнього шару, зберігає їх до кінця ітерації та передає у вихідну функцію, яка виконує дії з навколишнім середовищем.

Управляючий процес або супервізор - модуль синхронізації усіх складових штучної нейронної мережі. В своєму "стейті" має зберігати ідентифікатори процесів всіх нейронів, задіяних у штучній мережі, силового приводу та сенсору (Cortex_Pid, Actuator_Pid, [Neuron_Pid1, Neuron_Pid2, Neuron_Pid3, Neuron_Pid4]). В момент отримання силовим приводом всіх вхідних повідомлень від вихідного шару нейронів, генерується синхронізаційне повідомлення для супервізора, який в свою чергу сповіщає сенсор про початок нового збору даних з навколишнього середовища.

Визначившись з моделями всіх складових штучної нейронної мережі перейдемо до інструменту, який спрощує побудову нейронної мережі. Найпростіший конструктор моделі має приймати на вхід список шарів нейронної мережі, кожне значення якого відповідає кількості окремих нейронів шару, наприклад, при отриманні на вході списку [2,3,1] конструктор побудує модель, 1-й вхідний шар якої буде мати два нейрони, 2-й - три, а 3-й вихідний - один єдиний нейрон. Крім того, конструктор має визначити початкові стани нейронів та вхідні ваги. Створена конструктором модель має бути збережена у файлі у структурованому вигляді.

Модуль *exoself* необхідний для відтворення моделі з файлу, запуску у віртуальній машині та збереження з новими параметрами. Його основні функції - зчитування файлу, який створив конструктор, перетворення кожної структурованої одиниці нейронної мережі (нейрон, сенсор, силовий привід) в робочий процес віртуальної машини, а також перезапис нового стану нейронної мережі під час завершення роботи.

Висновки

Проведена декомпозиція моделі штучної нейронної мережі прямого розповсюдження сигналу, кожна складова реалізована інструментами мови програмування Erlang та протестована у віртуальній машині BEAM.

Список використаних джерел

1. Проць И.В. Моделирование нейронной сети средствами Erlang. Сборник материалов IX Международной научной конференции студентов и молодых ученых «Наука и образование - 2014». Казахстан, г.Астана, Евразийский национальный университет им. Л.Н. Гумилева 11 апреля 2014.–<http://www.enu.kz/ru/nauka/nauka-i-obrazovanie/>, 2014. – С. 3024-3028.
2. Томпсон С. Программирование в Erlang / С. Томпсон, Ф. Чезарини // М.:ДМК Пресс, 2012. - 488с.
3. Gene I. Sher Handbook of neuroevolution through Erlang / Gene I. Sher // NY.:Springer, 2013. - 835 p.
4. Armstrong J. Concurrent Programming in ERLANG / J. Armstrong, R.Virding, C. Wikstrom, M. Williams // NJ.:Englewood Cliffs, 1996. - 205p.

УДК 519.876.5

ПРОГРАМНА РЕАЛІЗАЦІЯ МОДУЛЯ КЛАСИФІКАЦІЇ НА ОСНОВІ ІНТЕРВАЛЬНИХ МОДЕЛЕЙ ІЗ РБФ ДЛЯ ЗАДАЧІ ВИЯВЛЕННЯ ГОРТАННОГО НЕРВА

Савка Н.Я.

Тернопільський національний економічний університет, здобувач

I. Вступ

Під час проведення операції на щитоподібній залозі основною проблемою є виявлення зворотного гортанного нерва (ЗГН), пошкодження якого призводить до негативних наслідків.

Інформаційний сигнал – результат подразнення області хірургічного втручання, характеризує наближеність до зворотного гортанного нерва і вказує чи точка подразнення належить ЗГН чи м'язовій тканині. У праці [1] розглянуто спосіб ідентифікації ЗГН на хірургічній рані за амплітудою результуючого інформаційного сигналу (РІС). Підхід до виявлення ЗГН, що ґрунтується на спектральних характеристиках РІС розглянуто у праці [2]. Дані підходи характеризуються низькою точністю ідентифікації ЗГН на хірургічній рані.

З огляду на це, доцільно застосувати нечіткий класифікатор на основі інтервальної моделі із радіально-базисними функціями для виявлення ЗГН на хірургічній рані в результаті класифікації РІС.

II. Особливості програмного середовища побудови інтервальної моделі із РБФ для задачі класифікації

Для реалізації програмного комплексу для побудови інтервальної моделі із РБФ для розв'язку задачі класифікації обрано пакет прикладних програм Matlab 7.1, який на даний час є ефективним інструментом для проведення прикладних розрахунків та розробки нових алгоритмів. Значною перевагою системи Matlab є відкритість коду реалізованих програм, що полегшує дослідження та розвиток реалізованих алгоритмів та наявність широкого спектру вбудованих стандартних функцій та інструментів – "тулбоксів", що базуються на матричних структурах даних.