

ВИКОРИСТАННЯ ГЕНЕТИЧНОГО АЛГОРИТМУ ПРИ ДІАГНОСТУВАННІ РПН ТРАНСФОРМАТОРІВ

Лежнюк П.Д., Рубаненко О.Є., Пиріжок М.І.
Вінницький національний технічний ун-т
rubanenko@bk.ru

In the article the example of the use of genetic algorithm is considered in forming of database for the studies of neutron network which is used for optimization of microprocessor device software of diagnostics of TCUL (tap-changing-under-load) of power transformers.

Вступ.

Велику роль в забезпеченні високої якості електричної енергії відіграють сучасні системи діагностування як силових трансформаторів так і їх РПН. Це дозволяє виявляти та прогнозувати їх технічний стан та надійність трансформатора в цілому. В свою чергу це дозволить не тільки своєчасно виводити обладнання в ремонт, а й значно ефективніше керувати перетоками потужностей в системі, враховуючи та прогножуючи технічний стан РПН силових трансформаторів в кожному вузлі окремо. Особливо актуально це питання постає в умовах неповноти отриманої інформації про РПН трансформатора та про його стан. Саме для розв'язання подібних задач найкраще зарекомендували себе алгоритми, які базуються на нейронних мережах та генетичних алгоритмах.

Генетичний алгоритм є найвідомішим на даний момент представником еволюційних алгоритмів і за своєю суттю є алгоритмом для знаходження глобального екстремуму багатоекстремумної функції. Як відомо, автоматичні і автоматизовані пристрої складаються з апаратної та програмної частини. Традиційно автоматичні системи діагностування реалізовували математичні моделі діагностованих об'єктів та враховували специфіку зміни технічного стану цих об'єктів в конкретних умовах експлуатації та враховували особливості поведінки діагностованого об'єкту в умовах експлуатації. Останніми роками все більш широкого поширення набувають системи діагностування, чий принципи функціонування відносяться до області роботи з базою знань - до штучного інтелекту (ШІ). Такі стріми використовують експертні системи, штучні нейронні мережі, системи з підкріплюючим навчанням, системи на основі нечіткої логіки, і т.п. Також перспективним для застосування в автоматичних та автоматизованих системах діагностування є метод автономного адаптивного діагностування (ААД) [7].

Задача полягає в розробці автономної адаптивної системи діагностування РПН трансформатора, яка б дозволяла прогнозувати залишковий ресурс РПН в умовах неповноти вихідних даних.

Метод оптимізації БД системи ААД РПН.

Пропонуємо використовувати метод генетичного алгоритму з метою оптимізації БД системи ААД РПН трансформаторів. Одним з показників технічного стану РПН є залишковий ресурс. Під час управління режимом ЕЕС диспетчер енергосистеми враховує стан РПН трансформаторів, але визначає його по результатам постійного моніторингу та по результатах попередніх вимірювань, наприклад під час ремонту. За таких умов значення параметрів, які в даний момент не вимірювались фактично є прогнозованими. Тому і залишковий ресурс РПН, на момент прийняття диспетчером рішень по підтриманню параметрів режиму ЕЕС, є прогнозованим.

Похибка прогнозування ресурсу РПН трансформатора повинна бути мінімальною. Використання генетичного алгоритму направлено на мінімізацію цієї похибки.

Для цього формуємо популяція серій вимірів, які є базою початкових даних параметрів (CO , CO_2 , C_2H_4 , C_2H_6 , C_2H_2 , $tg\delta$, I , W , R , K_f), де CO , CO_2 , C_2H_4 , C_2H_6 , C_2H_2 – гази, які містяться в баку вибирача РПН, R – опір обмоток, $tg\delta$ – тангенс кута діелектричних втрат масла в баку трансформатора, I – струм споживання привода РПН, W – вміст вологи в маслі.

База початкових даних створюється за результатами попередніх періодичних вимірювань, випробовувань, контролю під час різних видів ремонтів РПН. По ним навчається початкова нейронна мережа. Результати навчання корегуються шляхом врахування думки спеціалістів – експертів, які займаються ремонтами, обслуговуванням та експлуатацією РПН. Такі моделі будуються для РПН однотипних трансформаторів, які створюють популяцію РПН. Під час експлуатації в контролер РПН будуть надходити нові дані від сенсорів, які характеризують стан РПН.

Для кожної такої серії вимірів (з популяції РПН) обчислюється функція пристосованості для використання цієї серії в новій базі даних та в перенавчанні мережі. Це необхідно для того щоб забезпечити доповнення існуючої бази даних лише такими результатами, які не погіршують похибку навчання.

В подальшому ця база даних використовується в перенавчанні нейронної мережі з метою адаптації нейромоделі до реальних умов експлуатації. За допомогою функції пристосованості проводиться відбір серій вимірювань, які мають найкращі властивості, для визначення наступного покоління подібних серій вимірювань на основі операцій схрещування і мутації, які описуються нижче.

Інформація надходить в мікропроцесорний контролер від сенсорів, які встановлені на РПН. Під «правдивістю» розуміємо узгодженість (з задалегідь прийнятою похибкою) отриманих в результаті моделювання значень коефіцієнта залишкового ресурсу трансформатора зі значеннями цього ж коефіцієнта з бази даних.

Значення функції пристосованості даної серії вимірів для її застосування в перенавчанні нейронної мережі обчислюється шляхом інтегрування функції похибки навчання мережі по кількості серій вимірів в новій базі даних. Нова БД була створена шляхом додавання до існуючої БД нової серії вимірів. Якщо значення функції пристосованості зростає, то нова БД відкидається і залишається передостання БД.

Зона інтегрування визначається інтервалом від 0 до $N+1$, де N кількість серій вимірів в передостанній БД.

Для реалізації запропонованого методу в зону інтеграції періодично надходять значення кожної наступної серії вимірів. Ці дані отримують від сенсорів в режимі ON-Line під час експлуатації РПН.

Для кожного випадку надходження значень параметрів нової серії вимірів в БД, обчислюється значення штрафної функції, яка характеризує середньоквадратичну похибку навчання мережі.

Така процедура надходження нових серій вимірів і підрахунку значень штрафної функції проводиться багато (N) раз, наприклад $N=100$ (100 серій в одній сукупності тобто для одного РПН).

Далі розраховується значення функції пристосованості нової сукупності серій вимірів до існуючої БД за виразом:

$$F = \int_0^N [1 - MSE(i)] di \approx N \times \overline{MSE}, \quad (1)$$

де N – кількість серій вимірів в новій сукупності, i – поточний номер серії вимірів в новій сукупності серій вимірів, MSE – середньоквадратична похибка навчання мережі за умови додавання до існуючої БД поточної серії вимірів із сукупності серій, яка розраховується шляхом порівняння результатів моделювання з результатами в БД.

MSE для кожної серії вимірів розраховується за формулою:

$$\sigma = \sqrt{\frac{\sum_{j=1}^N (P_j - \bar{P})^2}{N}}. \quad (2)$$

Наступні 100 вимірів розраховуються так само.

Після обробки одночасно отриманих (для різних РПН з популяції) серій вимірів вони ранжуються за зростанням значення функції пристосованості.

Використання методу генетичних алгоритмів.

Метод генетичних алгоритмів передбачає наступні етапи: формування батьківських пар, схрещування, мутація, селекція.

Розглянемо перший етап – формування батьківських пар. Відбір сукупностей серій вимірів в батьківські пари відбувається у відповідності зі значеннями функції пристосовуваності. Розраховується середнє значення функції пристосованості всієї популяції \bar{F} . Для кожної сукупності серій вимірів (ССВ) визначаємо параметр $p_{i \text{ розр}}$, який дорівнює відношенню функції пристосованості ССВ до сумарної пристосованості всієї популяції (складається з ССВ різних РПН). Для тих ССВ, в яких значення функції пристосованості $F_i < \bar{F}$ ймовірність вибору приймаємо такою, яка дорівнює $0,1p_{i \text{ розр}}$. Підраховуємо кількість M_1 таких ССВ і паралельно розраховуємо суму

$$\tilde{P}_{iM-M1} = \sum_{i=1}^{M-M1} 0,9p_{i \text{ розр}}, \quad (3)$$

де M – кількість ССВ (а значить і контрольованих РПН). Для тих ССВ в яких функція пристосованості $F_i \geq \bar{F}$ ймовірність вибору приймаємо такою, яка дорівнює

$$P_{iM-M1} = p_{i \text{ розр}} + \frac{\tilde{P}}{M - M_1}, \quad (4)$$

де через M позначена чисельність всієї популяції. Після цього відповідно до значення «вірогідності вибору» формуються батьківські пари. Кожна пара складається з двох різних ССВ з найбільшими сумарними значеннями вірогідності вибору. Допускається входження однієї і тієї ж серії в декілька батьківських пар [8].

Якщо чисельність популяції дорівнює M , то формується $M-1$ батьківська пара. Кожна батьківська пара утворює одного нащадка. Серія вимірів з якнайкращими значенням функції пристосованості додається до наступного покоління автоматично.

Розглянемо другий етап генетичного алгоритму – схрещування. Як тільки батьківські пари вибрані, до них застосовується оператор схрещування. Існує багато різних версій цього оператора, серед яких простим, є однорідний оператор. По результатах вимірів, які містяться в ССВ_i та ССВ_j (одного – i та іншого – j батьків) він будує рішення ССВ_{ij} нащадка, надаючи кожному параметру цього вектора (ССВ_{ij}) з вірогідністю 0,5 відповідне значення одного з батьків. Якщо вектора параметрів ССВ_i та ССВ_j співпадали скажемо по першій серії вимірів, то вектор параметрів ССВ_{ij} "успадкує" це значення. Геометрично, оператор схрещування вибирає в гіперкубі вершину ССВ_{ij}, яка належить мінімальній грані, що містить вершини ССВ_i та ССВ_j. Оператор схрещування вибирає нове рішення ССВ_{ij} між ССВ_i та ССВ_j. Таким рішенням ССВ_{ij} буде оптимальне рішення початкової задачі на відповідній грані гіперкуба.

Результатом схрещування ССВ_i та ССВ_j буде сукупність параметрів ССВ_{ij}, яка містить в собі параметри серій вимірів ССВ_i та ССВ_j. Де ССВ_i та ССВ_j сукупність параметрів серії вимірів РПН_i та РПН_j відповідно. Допускається входження однієї і тієї ж серії в декілька батьківських пар.

Параметри ССВ_j дописуються до існуючої БД по кожній серії вимірів окремо. Потім мережа перенавчається. Якщо похибка навчання зростає порівняно з похибкою попереднього навчання, то така серія вимірів з ССВ_{ij} відкидається.

Розглянемо третій етап генетичного алгоритму – мутацію. Кожна ССВ_{ij} з популяції з вірогідністю p_{mut} може піддатися мутації. В результаті мутації утворюється ССВ_m, якій надаються нові значення серій вимірів, змінені випадковим чином з області допустимих значень. Нові параметри серії вимірів, які піддалися мутації, вибираються, виходячи з нормального закону розподілу $N(\mu, \sigma^2)$, у якого μ рівно початковому значенню параметра в серії вимірів, а $\sigma^2 = 2p$, де p - відстань між значеннями параметрів серії вимірів, що піддалися мутації, і серії вимірів з покоління “батьків” з якнайкращими значеннями функції пристосованості.

Якщо нові значення визначальних параметрів не потрапляють в область допустимих значень, то операція випадкового вибору нових параметрів проводиться знову. Створення ССВ_m припиняється, коли кількість серій вимірів в ній дорівнюватиме кількості серій вимірів в інших ССВ. Параметри з ССВ_m дописуються в БД за раніше запропонованим алгоритмом.

Розглянемо четвертий етап генетичного алгоритму – селекцію (відбір). Одним з найважливіших еволюційних чинників є природний відбір. Він діє, перш за все, в межах кожної популяції, залишаючи (або знищуючи) ті або інші генотипи.

Оператор відбору, вживаний до популяції $K_{pop\ i}$, записуватимемо як:

$$Selection(in : K_i, MaxSize, E_{abs}^{max}, E_{rel}^{max}), \quad (5)$$

Де MaxSize – максимальний розмір популяції, E_{abs}^{max} , E_{rel}^{max} – максимальні абсолютні і відносні похибки правила.

Запропонований генетичний алгоритм пропонуємо реалізувати за наступною структурою:

1. Створення початкової бази даних для кожного РПН з популяції,
2. Вибір декількох пар індивідів, призначених для схрещування (ССВ різних РПН),
3. Відбір в поточній ССВ_i - видалення найменш пристосованих серій вимірів (СВ),
4. Схрещування вибраних пар,
5. Дописування в БД кожного РПН ССВ_{сxp,i} отриманих в результаті схрещування,
6. Мутації ССВ_{ij},
7. Дописування в БД кожного РПН ССВ_{мут,i} отриманих в результаті мутації.

При розпаралелюванні обчислень використовувалася так звана «острівна модель» генетичних алгоритмів. Острівна модель характеризується тим, що вся популяція РПН ділиться на декілька підпопуляцій, які характеризують РПН кожного трансформатора. Кожна ССВ розраховується на своєму процесорі. Процесори виконують генетичні операції, а результатами обмінюються за допомогою міграції (передачі) даних через центральний сервер. На кожному острові даних підпопуляція піддається в точності таким же генетичним операціям на такому ж генотипі, що і в послідовному алгоритмі, і, в результаті, знаходиться кращий ССВ з урахуванням особливостей всієї популяції [5].

Початкова популяція ССВ ділиться по можливості на однакові частини і в процесі розрахунків в кожній ітерації циклу проводиться обмін серією вимірів кожної ССВ для покращення властивостей ССВ кожного РПН та покращення роботи систем ААД всієї популяції РПН.

Процес розрахунків на кожній з підстанцій зводився до ранжування кожної ССВ по функціям пристосованості. Серія вимірів з найменшим значенням функції пристосованості відкидається. Далі розраховується середнє значення функції пристосованості для кожної ССВ. Під популяції, в яких сума середніх значень функцій пристосованості найбільша, формують батьківські пари. Після проведення розрахунків функції пристосованості склали:

$$F_1 = 16.15, F_2 = 8.935, F_3 = 9.408, F_4 = 10.518.$$

Середнє значення функції пристосованості всієї популяції:

$$\bar{F} = 11,252.$$

Сумарне значення функцій пристосованості всієї популяції:

$$\sum F = 45.011.$$

Далі розраховується параметр P_{iproz} для кожної популяції

$$P_{iproz} = \frac{F_i}{\sum F}, \quad (6)$$

$$P_{1proz} = 0,358; P_{2proz} = 0,198; P_{3proz} = 0,209; P_{4proz} = 0,233.$$

Далі розраховується M_1 , для $F_i < \bar{F}$.

$$M_1 = 0.1 \times P_{iproz}.$$

$$M_1 = 0.1 \times 0.198 = 0.0198; M_1 = 0.1 \times 0.209 = 0.0209; M_1 = 0.1 \times 0.233 = 0.0233;$$

Далі за формулами (2) і (3) розраховуємо ймовірність вибору:

$$\tilde{P}_{iM-M1} = \sum_{i=1}^{M-M1} 0.9 p_{i \text{ розр}} \cdot \quad (7)$$

$$\tilde{P}_2 = 0.018; \tilde{P}_3 = 0.019; \tilde{P}_4 = 0.021.$$

Для ССВ, де функція пристосованості $F_i \geq \bar{F}$ ймовірність вибору

$$P_{iM-M1} = p_{i \text{ розр}} + \frac{\tilde{P}}{M - M_i} \cdot \quad (8)$$

$$\tilde{P}_1 = 0.038.$$

Тобто батьківські пари формуються із значень, які найбільш адекватно описують ресурс контролюемого РПН. В нашому випадку батьківські пари сформували \tilde{P}_1 та \tilde{P}_4 БД з хмельницької та тернопільської підстанцій. Їх сумарні значення вірогідності вибору склали для *тернопільської* = 0,76 та *хмельницької* = 0,81.

Далі застосовується оператор схрещування. До бази ССВ_{тернопіль} (ССВ_i) додається по черзі кожна строка (серія вимірів) ССВ_{хмельницький} (ССВ_j),

$$(ССВ_{i,j}) = (ССВ_{i(k,n)}) + (ССВ_{j(1,n)}), \quad (9)$$

де k – серія вимірів, n – параметри що вимірюються.

Після чого проводиться перенавчання мережі. Якщо похибка мережі зростає, така серія вимірів відкидається, якщо ж похибка зменшується, проводиться наступна ітерація схрещування, тобто додається наступна серія вимірів.

Перевірка ефективності використання генетичного алгоритму в діагностуванні РПН трансформатора.

Перевірка алгоритму здійснювалась шляхом порівняння результатів визначення залишкового ресурсу РПН трансформаторів ПЗЕС. Порівнювались результати визначення ресурсу мікропроцесорними пристроями діагностування (які використовували та не використовували генетичний алгоритм) з оцінками залишкового ресурсу під час виведення трансформаторів в ремонт. Розбіжність в оцінках ресурсу РПН кваліфікованими спеціалістами зі значеннями наданими пристроями діагностування на 6% менше у випадку використання генетичного алгоритму і становить 8%.

Висновок.

Використання генетичного алгоритму дозволяє зменшити похибку у визначенні залишкового ресурсу РПН, формувати базу даних з урахуванням найбільш інформативних серій вимірів, більш ефективно використовувати засоби зберігання результатів вимірів, шляхом відкидання мало впливових на визначення ресурсу, результатів вимірів.

ЛІТЕРАТУРА

1. А.А. Жданов, Метод автономного адаптивного управління // Известия Академии Наук. Теория и системы управления, 1999, № 5, с. 127-134.
2. А.А. Жданов, М.В. Крыжановский, Н.Б. Преображенский, Бионическая интеллектуальная автономная адаптивная система управления мобильным роботом // Мехатроника, 2004, №1, С. 21-30 и №2, С.17-22 (часть 2).
3. <http://www.ispras.ru/groups/ctt/ispran.html>
4. Victor Ivannikov, Serguei Gaissaryan, Arutyun Avetisyan, Vartan Padaryan. Improving properties of a parallel program in ParJava Environment // The 10th EuroPVM/MPI conference. LNCS 2840. Sept. 2003, Venice. pp. 491-494.
5. Rajkumar Buyya "High Performance Cluster Computing", Prentice Hall PTR, vol.2, P. 331-349.
6. Лежнюк П.Д., Писклярова А.В. Нейрон-нечётке моделирование в задачах визначення та оптимізації втрат електроенергії в розподільних електричних мережах // Вісник Харківського національного технічного університету сільського господарства ім. Петра Василенка. – 2006. – Вип.43. т.1. –С.50-58.
7. Саймон Хайкин, Нейронные сети: полный курс // Издательский дом "Вильямс", 2006.-1104 с.
8. В. П. Дьяконов, В. В. Круглов, MATLAB 6.5 SP1/7/7 SP1/7 SP2 + Simulink 5/6. Инструменты искусственного интеллекта и биоинформатики. // СОЛОН-ПРЕСС, 2006. – 456с.:

Рекомендовано д.т.н. Курінним Е.Г.