МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ОДЕСЬКИЙ ДЕРЖАВНИЙ ЕКОЛОГІЧНИЙ УНІВЕРСИТЕТ

Факультет Магістерської та аспірантської підготовки
Кафедра інформаційних технологій

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему: «Створення нейронної мережі на основі багатошарового перцептрона для короткострокового прогнозування метеопараметрів атмосфери»

Виконав студент 2 курсу групи КН- 2
спеціальності 122 Комп'ютерні науки

Ільєва Ірина Петрівна

Керівник к.геог.н., доц.
Кузніченко Світлана Дмитрівна

Консультант

Рецензент к.т.н., доц.
Гнатовська Ганна Арнольдівна

Одеса 2019

АНОТАЦІЯ

на магістерську роботу «Створення нейронної мережі на основі багатошарового перцептрона для короткострокового прогнозування метеопараметрів атмосфери», студентки Ільєвої Ірини Петрівни

Актуальність обраної теми обумовлюється тим, що передбачення погоди з наукової точки зору – одне з найскладніших завдань фізики атмосфери.

Метою дипломної роботи є створення нейронної мережі на основі багатошарового перцептрона для короткострокового прогнозування метеопараметрів атмосфери.

Для реалізації поставленої мети були вирішені наступні задачі: проведений огляд огляд сучасних методів інтелектуальних обчислень, які можуть бути використані для короткострокового прогнозування метеопараметрів атмосфери; надано опис методів прогнозування метеопараметрів атмосфери за часовими рядами; проведено збір та аналіз метеоданих; розроблена система короткострокового прогнозування метеопараметрів атмосфери на основі штучної нейронної мережі та виконана оцінка її адекватності.

Об’єкт дослідження – метеопараметри атмосфери, представлені часовими рядами.

Предмет дослідження – модель прогнозування за часовими рядами, заснована на штучній нейронній мережі.

Методами дослідження: математичне і комп’ютерне моделювання та методи штучного інтелекту. Результатами прогнозування є часові ряди прогнозованої температури повітря в м. Одеса, похибка результату становить менше 4%, отже система може використовуватися в системах прогнозування погоди.

Кваліфікаційна робота містить 77 сторінок, 4 розділи, 3 таблиці, 54 рисунки, 12 літературних посилань і 2 додатка.

Ключові слова: нейронні мережі, синоптичний прогноз, KNIME.

SUMMARY

….

The relevance of the topic chosen due to the fact that the weather prediction from a scientific point of view – one of the most difficult tasks of atmospheric physics.

The goal of the thesis is to study the possibility of forecasting weather parameters by its time series and forming and processing on the basis of method of calculation of short-term weather forecasting using artificial neural network model to create weather reports and customized Odessa region.

To achieve this goal have been resolved following tasks: a review of current models and technologies of intelligent computing techniques; examined the literature on methods for time series forecasting; collection and analysis of weather information to establish statistical patterns; develop predictive mathematical models; develop a system of forecasting, assessment of adequacy.

Object of research - time series.

Subject of research - time series model based on artificial neural networks.

As research methods mathematical and computer modeling and artificial intelligence were selected. The results forecasting is time series forecast temperatures in Odesa., The result of error is less than 4%, so the system can be used in weather forecasting systems.

Qualifying work includes 78 pages, 4 sections, 3 tables, 54 images, 11 literature references and 2 additions.

Keywords: neural networks, synoptic forecast, KNIME.

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

ВКФ – взаємна кореляційна функція

ШНМ – штучна нейронна мережа

CRM (скорочено від англійського CustomerRelationshipManagement) – управління відносинами з клієнтами, поняття, що охоплює концепції, котрі використовуються компаніями для управління їхніми взаємовідносинами зі споживачами, включаючи збір, зберігання й аналіз інформації про споживачів, постачальників, партнерів та інформації про взаємовідносини з ними.

DataMining – виявлення прихованих закономірностей або взаємозв'язків між змінними у великих масивах необроблених даних.

Java – об'єктно-орієнтована мова програмування, випущена компанією SunMicrosystems у 1995 році як основний компонент платформи Java.

JDBC (скорочено від англійського JavaDataBaseConnectivity) – прикладний програмний інтерфейс Java, який визначає методи, з допомогою яких програмне забезпечення на Java здійснює доступ до бази даних.

ODBC (скорочено від англійського OpenDataBaseConnectivity) – це відкритий інтерфейс доступу до баз даних, розроблений консорціумом X/Open.

Плагін (похідне від англлійськогоplug-in – підключати) – додаток, незалежно скомпільований програмний модуль, що динамічно підключається до основної програми, призначений для розширення або використання її можливостей.

R – мова програмування і програмне середовище для статистичних обчислень, аналізу та представлення даних в графічному вигляді.

Weka (WaikatoEnvironmentforKnowledgeAnalysis) – вільне програмне забезпечення для аналізу даних та машинного навчання, написане на Java в університеті Уайкато (Нова Зеландія), розповсюджується за ліцензією GNU GPL.

Apache POI – проект, що розповсюджує бібліотеки на Java для роботи з файлами Microsoft Office.

PMML (PredictiveModelMarkupLanguage) – мова розмітки для прогнозного моделювання на основі XML.

URL – стандартизована адреса певного ресурсу (такого як документ, чи зображення) в інтернеті.

XML – запропонований консорціумом WorldWideWeb (W3C) стандарт побудови мов розмітки ієрархічно структурованих даних для обміну між різними застосунками, зокрема, через Інтернет.

SDK – набір із засобів розробки, утиліт і документації, який дозволяє програмістам створювати прикладні програми за визначеною технологією або для певної платформи (програмної або програмно-апаратної).

WMO – Всесвітня метеорологічна організація, спеціалізований міжурядовий заклад ООН в області метеорології, заснований в 1950 році.

NOAA – Національне управління океанічних і атмосферних досліджень, федеральне відомство в структурі Міністерства торгівлі США, займається різними видами метеорологічних і геодезичних досліджень і прогнозів для США і їх володінь, вивченням світового океану і атмосфери.

SYNOP (КН-02) – цифровий код передачі синоптичних даних з автоматичних станцій спостереження.

METAR – авіаційний метеорологічний код для передачі інформації про фактичну погоду на аеродромі.

ВСТУП

В наш час прогнозування погоди є важливим завданням, вирішення якого надає можливість знизити економічні збитки. Так кожного року у світі стихійні нещастя забирають близько 250 000 людських життів та наносять збитків майну на 50-100 млрд. доларів. Але світова статистика показує: якщо довіряти гідрометеорологічній інформації та адекватно на неї реагувати, то можна відвернути від 30 до 40% втрат і повністю уникнути людських жертв [1][1)].

Актуальність обраної теми обумовлюється тим, що передбачення погоди з наукової точки зору – одне з найскладніших завдань фізики атмо-сфери. Існують різні методи для прогнозування метеорологічних явищ і їх величин, наприклад, синоптичні, кількісні, статистичні методи, але в повному об'ємі жоден метод не забезпечує поки що точного прогнозу. Традиційні підходи до вирішення перерахованих вище завдань не завжди дають необхідну гнучкість і деякі з них можуть бути ефективно вирішені за допомогою штучних нейронних мереж. Саме цьому дослідження у сфері прогнозування погодних явищ є важливими і корисними.

Метою магістерської роботи є створення нейронної мережі на основі багатошарового перцептрона для короткострокового прогнозування метеопараметрів атмосфери.

Для досягнення поставленої мети, в роботі необхідно вирішити наступні завдання:

- провести порівняльній огляд сучасних методів інтелектуальних обчислень, які можуть бути використані для короткострокового прогнозування метеопараметрів атмосфери;
- надати опис методів прогнозування метеопараметрів атмосфери за часовими рядами;
- виконати збір та аналіз метеоданих для встановлення статистичних закономірностей;
- розробити систему короткострокового прогнозування метеопараметрів

[1)] [1] Кому и зачем нужен прогноз? URL: http://www.primpogoda.ru/articles/prosto_o_pogode/komu_i_zachem_nuzhen_prognoz/ (дата звернення 15.11.2019)

атмосфери на основі штучної нейронної мережі, виконати оцінку її адекватності.

Магістерська робота містить 72 аркушів, 4 розділи, 3 таблиці, 54 рисунки і 2 додатка.

1 ПРИНЦИПИ МОДЕЛЮВАННЯ ПО ЧАСОВИХ РЯДАХ

1.1 Термінологія

У багатьох ситуаціях дані про досліджуваний об'єкт (процес) являють собою кінцеву послідовність значень спостерігаємої величини в різні моменти часу, тобто часовий ряд: $\eta(t_1),\eta(t_2),...,\eta(t_N)$, де $t_1,t_2,...,t_N$ – моменти спостережень, і їх число N називається довжиною ряду. Якщо в кожен момент часу t_i вимірюється значення тільки однією скалярною величини, то часовий ряд називають скалярним. Якщо ж одночасно в момент t_iвимірюються значення k величин $\eta_1,\dots\ \eta_k$, то ряд називають векторним, тому ці величини можна розглядати як компоненти k-мірного вектора η.

Елементи часового ряду (числа або вектори) називають також точками. Порядковий номер точки i називають дискретним часом. Якщо тимчасові інтервали між послідовними моментами спостережень t_i однакові– $t_i-t_{i-1}=\Delta t,\ i=2,...,N$, то ряд називають еквідистантним, в іншому випадку – нееквідістантним. Кажуть також, що вибірка зроблена рівномірно або нерівномірно, відповідно. Інтервал Δt між послідовними вимірами називають вибірковим інтервалом або інтервалом дискретизації. У разі нееквідістантного ряду вибірковий інтервал є величина змінна: $\Delta ti= t_i- t_{i-1}$. На практиці частіше зустрічаються і використовуються еквідистантні ряди [2][2)].

1.2 Загальна схема формування моделі

Незважаючи на безмежне число ситуацій, об'єктів і цілей, що вносять в процес своє специфічне, можна виділити основні етапи моделювання і представити їх у вигляді схеми (рис. 1.1).

Робота починається з розгляду наявної інформації про об'єкт (експериментальних даних про нього самого або подібних об'єктів; теорій, розроблених для опису досліджуваного класу об'єктів; інтуїтивних уявлень і т.д.) з позицій мети дослідження, з отримання та попереднього аналізу рядів

[2)] [2] Безручко Б.П., Смирнов Д.А. Математическое моделирование и хаотические временные ряды. Саратов: «Колледж», 2005. 320 с.

спостережуваних величин, а закінчується використанням отриманої моделі для вирішення конкретного завдання. Але цей процес зазвичай є ітераційним, тобто супроводжується неодноразовими повтореннями, поверненнями у вихідну і проміжні точки схеми, послідовними наближеннями до «гарної» моделі.

Рисунок 1.1 – Типова схема процесу емпіричного моделювання

Розглянемо більш детально кожен етап. На першому етапі здійснюється отримання даних та їх систематизація. Далі застосовується один або кілька методів аналізу наявних часових рядів спостережуваних метеовеличин. Це, наприклад, візуальний аналіз у вигляді графіків залежності змінної від часу, відновлення фазової траєкторії, спектральний і статистичний аналіз та інші.

Одним з ключових в процесі моделювання є етап 2, на якому формується структура моделі. Це проводиться наступним чином: спочатку вибирається тип рівнянь, далі задається вид входять до них функцій, після чого встановлюється зв'язок динамічних змінних (компонент вектораx) зі спостережуваними величинами η. В якості змінних можуть виступати і самі спостережувані, але в більш загальному

випадку цей зв'язок задають у вигляді $\eta = h(x)$, де h називають вимірювальної функцією. Часто вводять ще випадкову добавку ζ: $\eta = h(x) + \zeta$, щоб врахувати вимірювальний шум. Щоб зробити модель більш реалістичною, випадкову добавку вводять нерідко і в самі рівняння – так званий динамічний шум [3][3)].

Цей етап – найбільш складний і творчий. На ньому вибирається тип рівнянь, вид вхідних до них функцій та їх аргументів.

Задача визначення аргументів функції полягає в тому, щоб визначити найменшу розмірність моделі, що забезпечує однозначність прогнозу. Існують різні підходи до її оцінки, – це метод найближчих помилкових сусідів [4][4)], метод головних компонент [5][5)], метод Грассбергера-Прокаччі, метод добре пристосованого базису.

Далі йде етап визначення структури модельних рівнянь. Для цього використовуються різні методи апроксимації функцій багатьох змінних: метод узагальнених многочленів, використання радіальних базисних функцій, штучні нейронні мережі [6][6)].

Після вибору структури виконують «підгонку моделі». Для цього, як правило, проводиться пошук екстремального значення деякої цільової функції, наприклад, мінімізується сума квадратів відхилень рішення модельних рівнянь від спостережуваних даних. При необхідності на даному етапі проводяться попередні перетворення спостережуваного ряду: фільтрація від шумів, чисельне диференціювання або інтегрування і т.п. Це, в основному, технічний етап чисельних розрахунків, але і тут потрібно зробити вибір принципу розрахунку параметрів та методики для його реалізації.

На останній стадії моделювання проводиться перевірка «якості» розробленої моделі. Зазвичай це здійснюється з використанням збереженої для цієї мети тестової

[3)] [3] Дымников В.П., Филатов А.Н. Основы математической теории климата, М.: ВИНИТИ, 1994. 254 с.

[4)] [4] Использование метода ближайших ложных соседей для предобработки временных рядов. URL: http://moyuniver.net/ispolzovanie-metoda-blizhajshix-lozhnyx-sosedej-dlya-predobrabotki-vremennyx-ryadov/ (дата звернення 15.11.2019)

[5)] [5] Метод головних компонент URL: http://bko.com/book_346_glava_69_%C2%A7_2.6._%D0%9C%D0%B5%D1%82%D0%BE%D0%B4_%D0%B3%D0%BE%D0%BB%D0%BE%D0%B2%EF%BF%BD.html (дата звернення 15.11.2019)

[6)] [6] Решение задачи прогнозирования с помощью нейронных сетей. URL: http://www.rusnauka.com/1-NIO_2011/Informatica/78176.doc.htm (дата звернення 15.11.2019)

частини часового ряду. Проводиться перевірка ефективності моделі в плані досягнення необхідної точності прогнозу. Якщо модель визнана задовільною (ефективною), отримана конструкція береться у справу, інакше – повертається на доопрацювання на будь-який з перерахованих етапів.

2 ЗАСТОСУВАННЯ ШТУЧНИХ НЕЙРОННИХ МЕРЕЖ В ЗАДАЧАХ ПРОГНОЗУВАННЯ

2.1 Огляд класичних методів та засобів прогнозування

У різних областях людської діяльності часто виникають ситуації, коли за наявною інформацією (даними), позначимо її X, потрібно передбачити (спрогнозувати, оцінити) деяку величину Y, яка стохастично зв'язана з X, але яку безпосередньо зміряти неможливо (наприклад Y може відноситися до майбутнього). Прогнозування, знаходження схованих періодичностей у даних, аналіз залежностей, оцінка ризиків при прийнятті рішень і інших задач вирішуються в рамках статистичних моделей.

За допомогою статистичних моделей описуються явища, у яких присутні статистичні фактори, що не дозволяють пояснити явище в чисто детерміністських термінах. Типові приклади такого роду моделей становлять часові ряди в економіці, фінансовій сфері й природних явищах, що мають тренд-циклічний компонент і випадкову складову. Хоче того чи ні, дослідник не може виключити випадкову складову і повинен будувати свої висновки, з огляду на її наявність.

Статистика оперує наступними поняттями.

Генеральна сукупність – множина всіх об'єктів у дослідженнях.

Вибірка – підмножина генеральної сукупності, що безпосередньо бере участь у статистичній обробці. Вибірка повинна бути об'єктивною, інакше результати будуть перекручені. Способи відбору вибірок: випадковий, систематичний (наприклад, кожний сьомий), експертний, районований. Вибірку задають у вигляді статистичного ряду – послідовності чисел.

По вибірці будують гістограму – графічне зображення статистичного ряду, статистичний аналог функції щільності в теорії ймовірностей. Властивість функції щільності збігається з аналогічною властивістю гістограми: площа гістограми дорівнює одиниці. По вигляду гістограми роблять припущення про характер закону розподілу досліджуваної величини і уточнюють параметри цього розподілу одним з методів: метод моментів, метод максимальної правдоподібності, метод найменших квадратів, також оцінюють математичне очікування, дисперсію, знаходять довірчий

інтервал для цих оцінок. Для перевірки несуперечності даних припущеному закону з уточненими параметрами використовують критерій Пірсона або Колмогорова.

Теорія стохастичного прогнозування вивчає методи побудови предикторів. У загальному випадку X позначає деяку сукупність {X1, X2 ., Xn} спостережуваних випадкових величин, які в даному контексті називаються прогнозними змінними. Задача полягає в побудові такої функції Ф(X), яку можна було б використовувати як оцінку для прогнозованої величини Y: Ф(X)=Y (тобто щоб вона була в якомусь сенсі "близька" до Y); такі функції Ф(X) називають предикторами величини Y по X. Розробка методів побудови оптимальних (у тому або іншому сенсі) предикторів і складає головну задачу прогнозування. Якщо Ф(X) використовується для передбачення величини Y, то однієї з розумних мір розбіжності між ними є (Ф(X)-Y)2, або квадратична помилка, але тому що величина Y невідома, то для виміру точності предиктора Ф використовується середньоквадратична помилка ΔФ=M(Ф(X)-Y)2, де M – знак математичного очікування. Середньоквадратична помилка – міра, що традиційно використовується в теорії стохастичного прогнозування, хоча в принципі можна було б використовувати й інші міри точності, наприклад середню абсолютну помилку. Предиктор, який мінімізує середньоквадратичну помилку в заданому класі предикторів, називають оптимальним предиктором або прогнозом [6][7)].

Якщо сукупність величин {X1, X2 ., Xn} є значеннями якого-небудь параметра, який змінюється в часі, то таку сукупність називають часовим рядом. При цьому кожне значення відповідає значенню параметра в конкретний час t1, t2 .,tn. Задача прогнозування в цьому випадку полягає у визначенні значення вимірюваної величини X у моменти часу tn+1, tn+2, tn+3, тобто для виконання прогнозування необхідно виявити закономірність цього часового ряду.

2.2 Класифікація задач прогнозування часових рядів

Особливе значення мають задачі передбачення та прогнозування часових рядів, серед яких виділяються завдання з набором певних специфічних ознак, тому варто провести їх класифікацію. Задачі дослідження явищ, розвиток яких пов'язаний із

[7)] [6] Решение задачи прогнозирования с помощью нейронных сетей. URL: http://www.rusnauka.com/1-NIO_2011/Informatica/78176.doc.htm (дата звернення 15.11.2019)

часом, можна поділити на декілька класів:

За характером основних ознак об'єкту:

- прогнозування явищ, реалізації яких представлені у вигляді детермінованих часових рядів. Такі задачі, зокрема, можна вирішити шляхом застосування методів математичного аналізу;
- прогнозування явищ, реалізації яких представлені у вигляді індетермінованих часових рядів. Вирішення цих задач традиційно здійснюється шляхом застосування методів теорії ймовірностей та математичної статистики.

Зокрема, реалізації таких явищ, можуть мати вигляд:

а) стаціонарного часового ряду, який характеризується однорідністю в часі, без суттєвих змін характеру коливань та їх середньої амплітуди, вибір проміжку для формування навчальної множини довільний;

б) нестаціонарного часового ряду, який характеризується певною тенденцією розвитку в часі, при дослідженні нестаціонарних процесів можна виділити ділянки, на яких процес можна вважати стаціонарним, до того ж вибір проміжку для формування навчальної множини в такому випадку обирається згідно задачі прогнозування.

За числом ознак об'єкту досліджень:

- одновимірна задача; явище представлене лише однією ознакою, зміни якої відбуваються в часі;
- багатовимірна задача; об'єкт або явище представлені кількома ознаками; така задача прогнозування може бути розширена завдяки представленню даних в просторі.

Враховуючи специфічний характер прогнозування часових рядів і певний різнобій в термінології, дотримуватимемося ряду визначень.

Передісторією ряду назвемо набір елементів часового ряду, який враховується для одного кроку прогнозування наступних елементів часового ряду. Однокрокове прогнозування зводиться до задач відображення у випадку, коли значення елементів передісторії можуть визначати лише один дискретний відлік вихідних величин. Багатокрокове прогнозування характеризується збільшенням дискретних відліків вихідної величини і, відповідно, збільшенням часу, на який здійснюється прогноз (час випередження Твип). При багатокроковому прогнозуванні Твип=a*r, де R – кількість

кроків обчислення прогнозування; a – крок дискретизації вихідного параметра (наприклад, рік, місяць, день, і тому подібне) [7][8)].

За часом випередження розрізняють види прогнозів:

- згладжування, R= 0;
- короткостроковий прогноз, R= 1 – 2;
- середньостроковий прогноз, R= 3 – 7;
- довгостроковий прогноз, R= 10 – 15.

Очевидно, що вид прогнозу суттєво впливає на вибір засобів і методику його реалізації.

2.3 Адаптація нейромереж до даних навчальних множин

Дані про поведінку об'єкту, ознаки якого пов'язані з часом, представлені як результати спостережень в рівномірні відліки часу. Для моментів часу t=1, 2 ..., n дані спостережень набувають вигляду часового ряду x(t1), x(t2) ..., x(tn). Інформація про значення часового ряду до моменту n дозволяє давати оцінки параметрів x(tn+1), x(tn+2) ..., x(tn+m). Для прогнозування елементів часових рядів широко використовують так званий метод "часових вікон". Ідея його аналогічна для однопараметричних і багатопараметричних задач прогнозування, тому розглянемо його особливості стосовно однопараметричної задачі прогнозування.

Нехай часовий ряд x(t) заданий відліками процесу x(t1), x(t2)..., x(ti) в дискретні моменти часу t. Задамо ширину (кількість дискретних відліків) вхідного часового вікна m, ширину вихідного вікна p. Вхідне і вихідне вікна накладаються на дані ряду, починаючи з першого елементу (рис.2.1).

[8)] [7] Искусственные нейронные сети. URL: http://victoria.lviv.ua/html/oio/html/ theme5_rus.htm (дата звернення 15.11.2019)

Рисунок 2.1 – Формування множин даних для однопараметричної задачі прогнозування за методом "часових вікон"

Вхідне вікно формує дані для входів нейронної мережі, а вихідне, відповідно, для виходів. Подібна пара вхідного і вихідного векторів приймається за одну реалізацію часового ряду. При зсуві часових вікон за часовим рядом з кроком s, отримуємо другу і наступні реалізації.

Значення ширини вікон та кроку зсуву повинні узгоджуватися з особливостями часового ряду, що забезпечується шляхом проведення експериментів. Нехай вхідне вікно має ширину m, вихідне вікно p=1, крок зсуву s=1. Тоді сформована множина значень для однопараметричної задачі матиме вигляд, приведений нижче [6][9].

Таблиця 2.1 – Множина даних для однопараметричної задачі

Входи				Виходи
$x(t_1)$	$x(t_2)$	...	$x(t_m)$	$x(t_{m+1})$
$x(t_2)$	$x(t_3)$	...	$x(t_{m+1})$	$x(t_{m+2})$
$x(t$	$x(t$	.	$x(t_{m+2})$	$x(t_{m+3})$

[9] [6] Решение задачи прогнозирования с помощью нейронных сетей. URL: http://www.rusnauka.com/1-NIO_2011/Informatica/78176.doc.htm (дата звернення 15.11.2019)

$_3)$	$_4)$	..		
...	...	. ..	...	...
$x(t_i)$	$x(t_{i+1})$	. ..	$x(t_{i+m-1})$	$x(t_{i+m})$

2.4 Однокрокове прогнозування (передбачення)

Розрізняють багатокроковий і однокроковий прогноз. Однокроковим прогнозуванням називають короткостроковий прогноз (на один крок), при цьому для отримання прогнозованої величини використовують тільки фактичні дані. Ясно, що однокрокове прогнозування точніше, але воно не дозволяє виконувати довгострокові прогнози. Таким чином, завдання однокрокового прогнозування зводиться до завдання відображення, коли один вхідний вектор відображається у вихідний.

В режимі навчання встановлюються коефіцієнти ваг зв'язків, після чого стає можливим перехід до режиму функціонування. Для передбачення на входи нейромережі поступають значення останньої реалізації навчальної множини x(tn-2), x(tn-1), x(tn). На виході формується прогнозована величина x*(tn+1) [6][10]. Передбачення застосовують для моделювання дискретних послідовностей, які не зв'язані з часом. Зважаючи на специфіку часових рядів, такий тип прогнозу не завжди доцільний, але в певних випадках короткострокових прогнозів їм можна скористатися.

2.5 Багатокрокове прогнозування

Багатокрокове прогнозування застосовують лише для явищ, ознаки яких представлені у вигляді часових рядів. Багатокрокове прогнозування часового ряду здійснюється таким чином. На входи нейромережі подається вектор відомих значень x(tn-2), x(tn-1), x(tn). На виході формується прогнозована величина x*(tn+1), яка визначає вектор прогнозованих виходів і одночасно долучається до значень

[10] [6] Решение задачи прогнозирования с помощью нейронных сетей. URL: http://www.rusnauka.com/1-NIO_2011/Informatica/78176.doc.htm (дата звернення 15.11.2019)

навчальної множини, тобто, приймається як достовірна. Далі на входи подається вектор x(tn-1), x(tn), x*(tn+1), а на виході отримується x*(tn+2) і наступні прогнозовані значення [7][11)]. Багатокрокове прогнозування дозволяє робити короткострокові і середньострокові прогнози, оскільки суттєвий вплив на точність має накопичення похибки на кожному кроці прогнозування. При застосуванні довгострокового багатокрокового прогнозування спостерігається характерне для багатьох прогнозуючих систем поступове затухання процесу, фазові зсуви і інші спотворення картини прогнозу. Такий тип прогнозування підходить для стаціонарних часових рядів з невеликою випадковою складовою.

2.6 Оцінювання точності прогнозів

Як правило, після навчання нейромережі здійснюють контрольне відтворення даних, які складали навчальну множину. Якщо точність відтворення задовільна і відхилення знаходяться в допустимих межах, вважають, що побудовано задовільну модель і слід очікувати достатню якість відображення. Якщо при відтворенні мережею даних навчальної множини спостерігаються великі розбіжності, можна припустити що це викликано:

- наявністю неточних даних з великою випадковою складовою, для усунення цього явища підвищують вимоги до точності вимірювань, а у випадку часового ряду, можливе зменшення кроку дискретизації, наприклад використання щомісячних значень замість річних;
- неврахуванням суттєвих ознак, які в значній мірі визначають закономірність – ця проблема може бути вирішена розширенням набору ознак, які приймаються до уваги;

Після отримання передбачених значень при наявності правильних можливо отримати абсолютні та відносні відхилення на всій контрольній множині, для кожного кроку прогнозування. При наявності задовільних результатів прогнозування на контрольній множині, можна вважати, що налаштована мережа для даної задачі має

[11)] [7] Искусственные нейронные сети. URL: http://victoria.lviv.ua/html/oio/html/ theme5_rus.htm (дата звернення 15.11.2019)

оптимальну складність і готова до відтворення даних, для яких немає відповідних відомих відгуків.

3 ПОСТАНОВКА ЗАДАЧІ

Метою цієї роботи є розроблення системи, що моделює роботу багатошарової нейронної мережі типу перцептрон, яка навчена методом зворотного поширення похибки, і яка дозволить виконувати однокрокове і багатокрокове прогнозування часових рядівметеовеличин.

Завдання прогнозування коротко можна викласти таким чином: за наявною інформацією (даним), позначимо її X, потрібно спрогнозувати (оцінити) деяку величину Y, що стохастично пов'язана з X (тобто X і Y мають деякий розподіл L(X,Y)), але яку безпосередньо виміряти неможливо (наприклад, Y може відноситися до майбутнього, а X – до сьогодення). В даному випадку, інтерес становить прогноз метеовеличин – температури повітря, наприклад. Тут X –температура повітря в місті Одеса за попередні роки, взята з періодичністю у 8 разів на добу, а Y –прогнозована температура повітряв місті Одеса, яку можна визначити (оцінити) за аналогічним даними за минулі роки.

У загальному випадку X означає деяку сукупність $\{X_1, X_2,\ldots\}$ спостережуваних випадкових величин, які в розглянутому контексті називаються прогнозними змінними, і завдання полягає в побудові такої функції F(X), яку можна було б використовувати як оцінку для прогнозованої величини Y:F(X)=Y. Більш докладно можливості застосування нейронних мереж для рішення задач прогнозування розглянуті в розділі 2.

Після закінчення розробки системи необхідно провести аналіз результатів однокрокового прогнозування температури повітря в місті Одеса.

4 РОЗРОБКА ПРОГРАМНОЇ РЕАЛІЗАЦІЇ СИСТЕМИ КОРОТКОСТРОКОВОГО ПРОГНОЗУВАННЯ

4.1 Платформа KNIME

Платформою для розробки системи прогнозування була вибрана KNIME.

KNIME (KonstanzInformationMiner) – це інтеграційна платформа з відкритим сирцевим кодом для аналізу даних, звітності тощо. KNIME інтегрує різні компоненти для машинного навчання та інтелектуального аналізу даних за рахунок своєї модульної концепції конвеєризації даних. Графічний користувальницький інтерфейс дозволяє складання вузлів для підготовки даних, для моделювання та аналізу даних і візуалізації [8][12)].

KNIME –середовище для аналізу даних, що становить найбільший інтерес в плані використання в сфері освіти. Суть проведення аналізу даних полягає в наступному: аналітик використовує різноманітні методи аналізу, які представлені програмою у вигляді "вузлів", виходячи з завдання, яке необхідно вирішити, шляхом перетягування на робочу область цих елементів-"вузлів" послідовно, формуючи в кінцевому рахунку потік даних, приклад якого представлений на рис. 4.1.

Набір методів аналізу, запропонованих програмою, дуже широкий: визначення стандартних статистичних величин, кореляційний аналіз, кластеризація, класифікація, нейронні мережі, асоціативні правила і т.д. Тобто по суті, KNIME дозволяє реалізувати всі методи DataMining.

Крім іншого, програма дозволяє налагодити зв'язок з базою даних за допомогою JDBC і ODBC, що є очевидною перевагою, адже зчитування даних з бази даних дозволить своєчасно обробляти дані, а також уникнути можливості неврахованих даних або їх втрати.

KNIME має і інструментарій в плані візуалізації даних, який представлений різними таблицями, графіками і діаграмами, приклад яких можна побачити на рис. 4.2.

---

12) [8] KNIME. Wikipedia. URL: https://en.wikipedia.org/wiki/KNIME (дата звернення 15.11.2019)

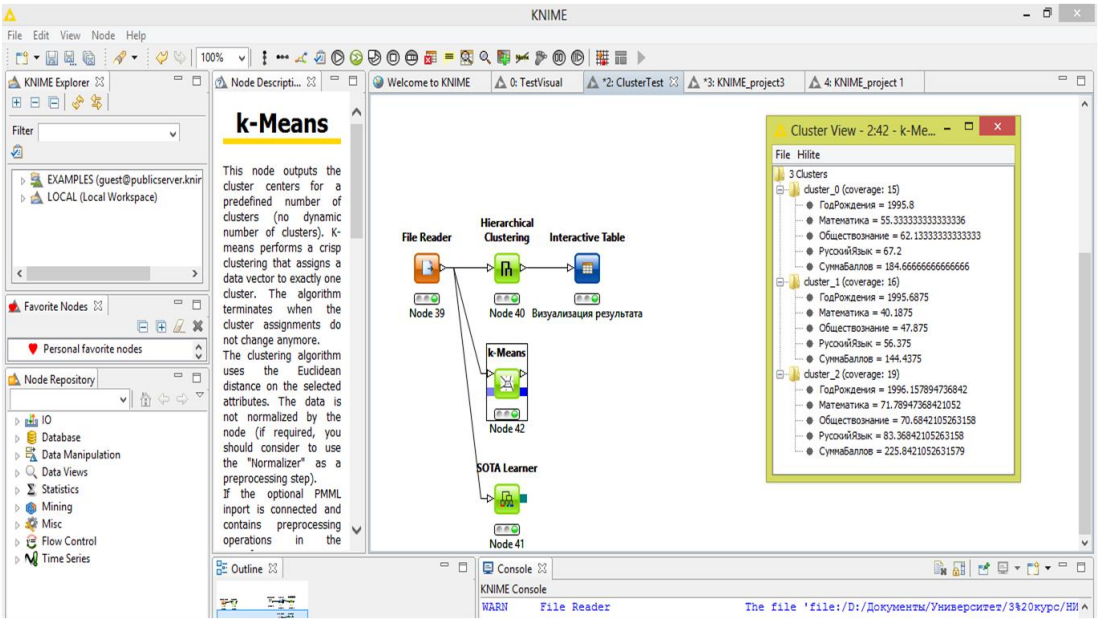


Рисунок 4.1 – Потік даних в KNIME

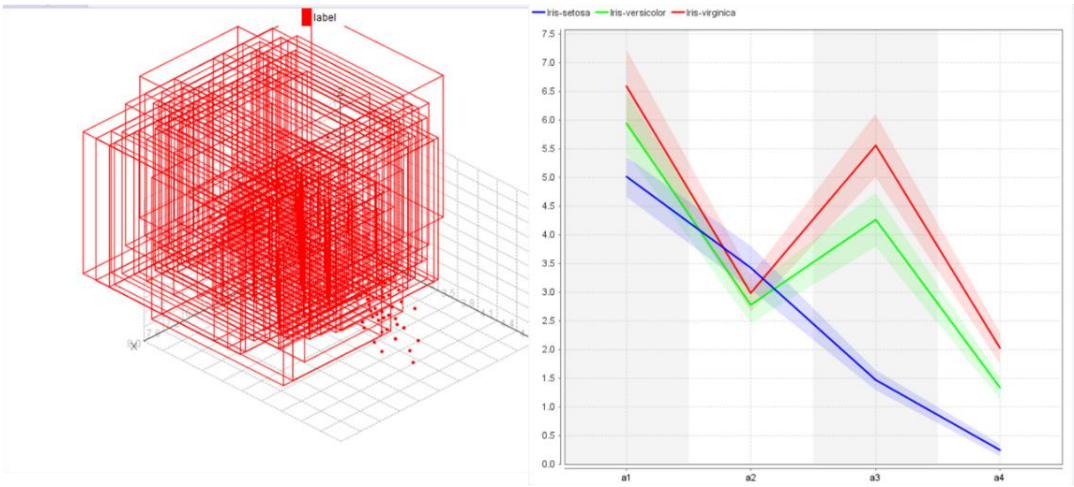


Рисунок 4.2 – Візуалізація даних в KNIME

Ще одна можливість KNIME – це створення метавузлів на робочій області в числі потоків (рис. 4.3). Метавузел може містити в собі кілька вузлів, також розміщених за потоковим принципом. Підсумком роботи метавузла може бути один або кілька виходів в залежності від виду метавузла.

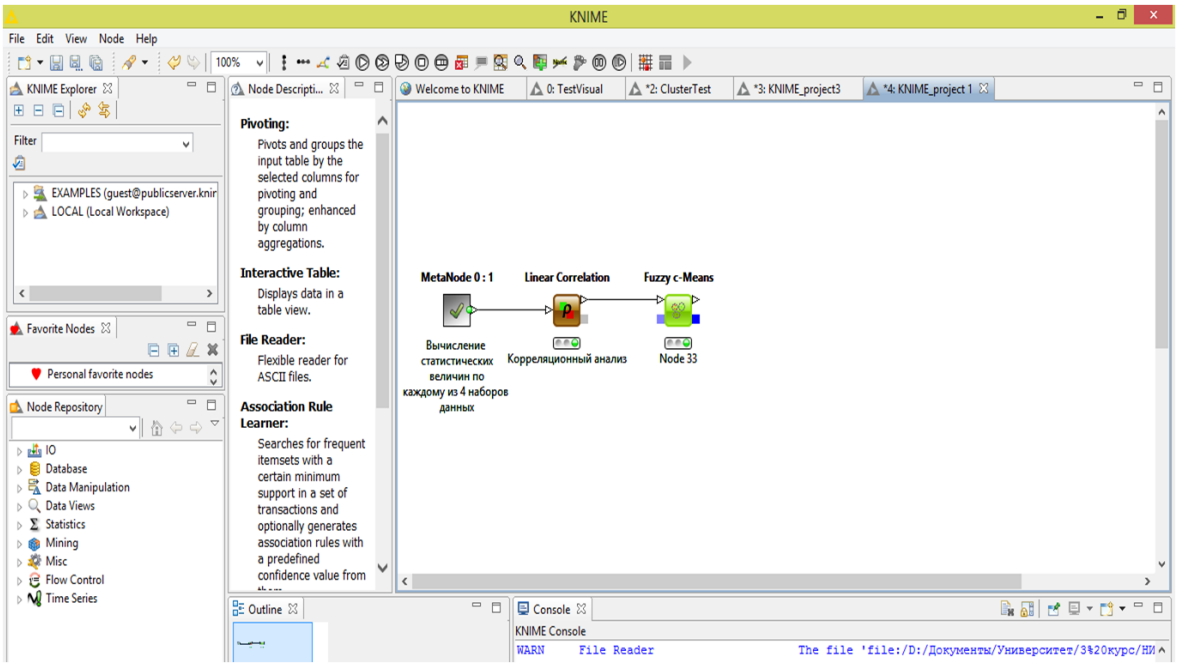


Рисунок 4.3 – Вигляд метавузла в потоці даних

Зазвичай метаузли потрібні, щоб продемонструвати складну логіку протікання операцій, в ході чого вихідні дані на вході метавузла багаторазово піддаються будь-якій обробці (рис. 4.4).

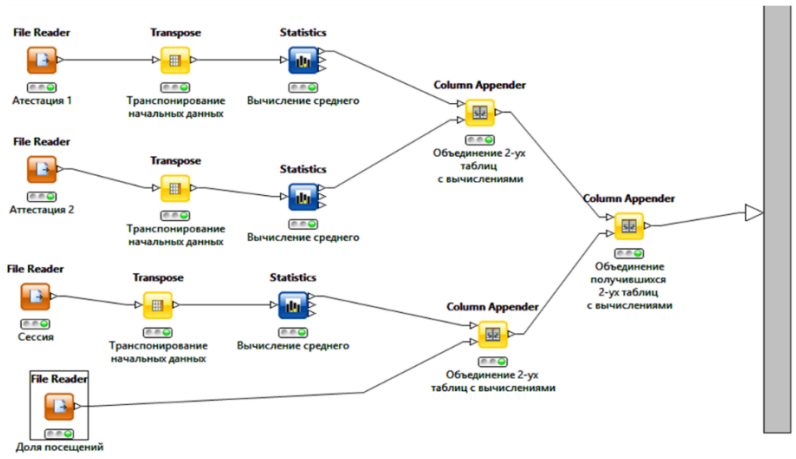


Рисунок 4.4 – Обробка даних всередині метавузла

Програма вимагає мінімум витрат на установку (по суті, установка не потрібно, необхідний лише запуск, який спричинить створення робочої області на вибраному диску), тому робота з KNIME не вимагає яких-небудь особливих знань – інтерфейс, хоча і англомовний, але тим не менш достатній для розуміння (рис. 4.5).

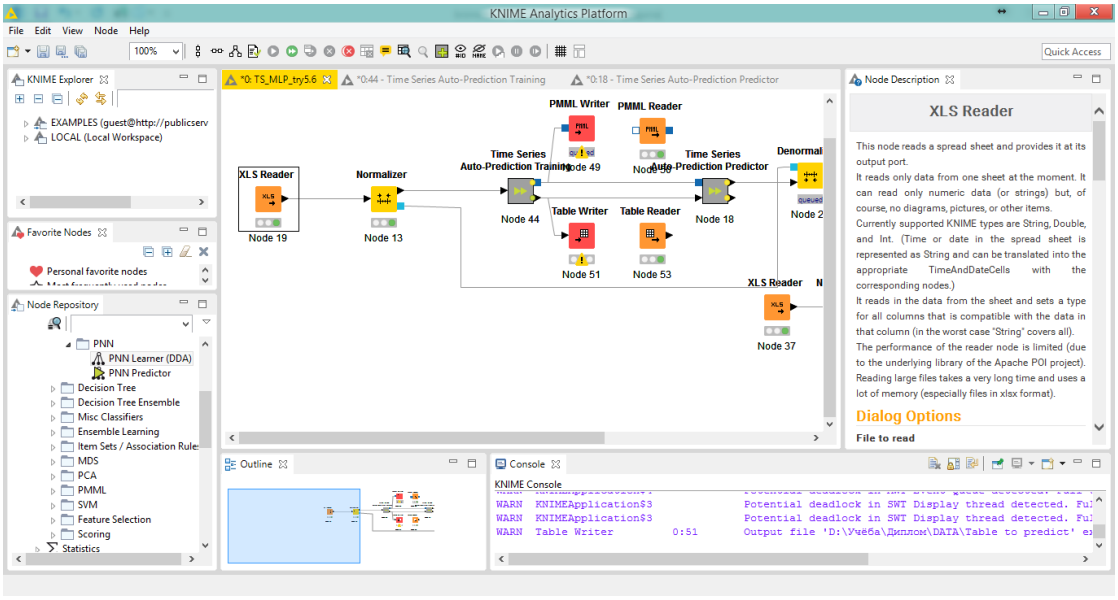


Рисунок 4.5 – Користувацький інтерфейс KNIME

KNIME, як і личить ВПЗ, має відкритий вихідний код, а це означає, що програма може бути доопрацьована з урахуванням деяких вимог. Тут важливо зазначити, що програма написана на мові Java, а середовище розробки KNIME –Eclipse.

Варто відзначити, що хоча ВПЗ дещо поступаються лідерам ринку, проте представляється, що можливостей ВПЗ досить, щоб проводити інтелектуальний аналіз даних. Причому лідери ринку зазвичай включають різні аналітичні доповнення DataMining до основного продукту, тоді як KNIME – це самостійний продукт, націлений саме на DataMining.

4.2 Робота зплатформою KNIME

Коли KNIME запускається вперше з'являється екран вітання (рис. 4.6).

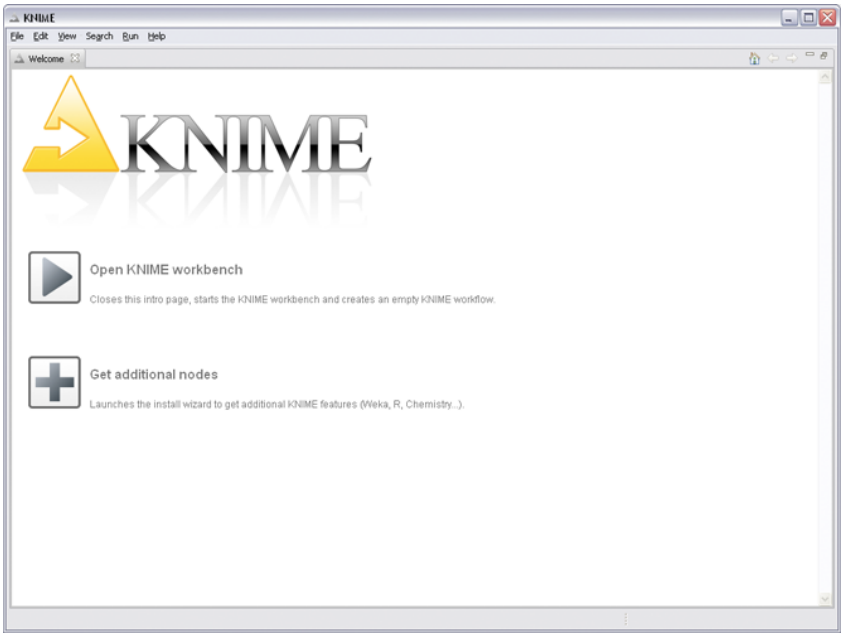


Рисунок 4.6 – Екран вітання KNIME

Робоча область KNIME містить такі елементи:

- Проекти потоків даних. Кожен потік даних відноситься до якогось проекту. Всі проекти відображаються в цьому вікні. Підтримується імпорт та експорт потоків. Стан (зачинений, в покої, виконується або виконаний) відображається іконкою.
- Улюблені вузли. Керуйте своїми улюбленими вузлами, вузлами, що найчастіше використовуються та вузлами, що використовувалися останнім часом. Вузли додаються перетягуванням їх з репозиторію вузлів до категорії улюблених вузлів.
- Репозиторій вузлів. Тут знаходяться всі вузли, впорядковані за категоріями. Детальну інформацію про вибраний вузол можна знайти в області Опис вузла. Щоб скористатися вузлом, треба перетягнути його в область Редактора потоків.
- Редактор потоків. Тут збираються потоки даних перетягуванням вузлів в цю область, з'єднуванням їх один з одним, налаштуванням та виконанням вузлів.
- Опис вузла. Надає детальний опис вибраного вузла, його діалогові опції, очіковані вхідні дані та результуючі вихідні.

– Ескіз. Зменшений вигляд всієї області редактору потоків для спрощення навігації по великих потоках даних.
– Консоль. Інформація про стан, попередження та репорти про помилки логуються сюди.

Потік даних побудований шляхом перетягуванням вузлів від репозиторію вузлів в редактор потоків і поєднанням їх. Вузли є базовими одиницями обробки в потоках даних. Кожен вузол має ряд вхідних і/або вихідних портів. Дані (або модель) передається через з'єднання з вихідного порту одного вузла до вхідного порту іншого вузла.

Коли вузол перетягли до редактору потоків даних індикатор стану вузла спалахує червоним, що означає, що вузол повинен бути налаштований для того, щоб мати можливість бути виконаним. Вузол налаштовується так - правою кнопкою миші клацніть на вузол, виберіть пункт «Налаштувати» і коригуйте необхідні налаштування в діалоговому вікні вузла (рис. 4.7).

Коли діалогове вікно закривається за допомогою кнопки "OK", то вузол налаштований і індикатор стану змінюється на жовтий: вузол готовий до виконання. Клацніть правою кнопкою миші - вузол знову показує на включену опцію "Виконати" (рис. 4.8), натиснувши яку буде виконуватися вузол і результати роботи цього вузла будуть доступні на вихідних портах вузла. Після успішного виконання індикатор стану вузла стане зеленим. Результат(и) можуть бути перевірені шляхом вивчення перегляду(ів) з порту: останні записи в контекстному меню.

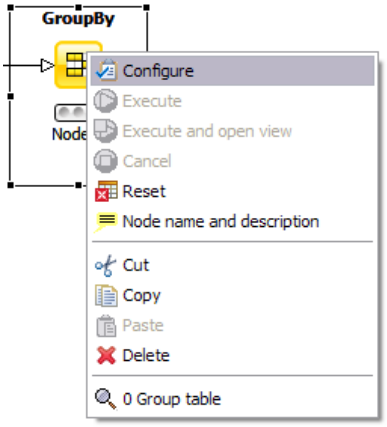


Рисунок 4.7 – Контекстне меню вузла до налаштування

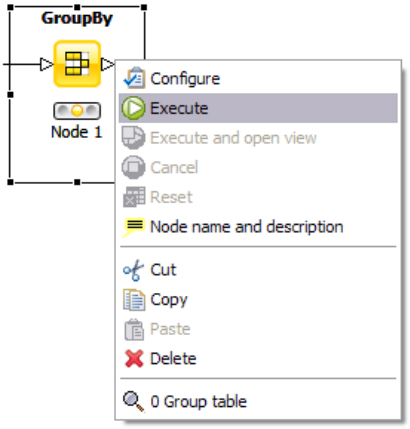


Рисунок 4.8 – Контекстне меню вузла після налаштування

Порти зліва є входи, куди передаються дані з вихідного порту вузла попередника. Порти справа є вихідними портами. Результат обробки цим вузлом даних передається на вихідний порт вузлу-наступнику. У підказці міститься інформація про вихід вузла, додаткову інформацію можна знайти в описі вузла. Вузли задумані таким чином, що тільки порти одного і того ж типу можуть бути з'єднані.

Найбільш поширеним типом є порт даних (білий трикутник, а після третьої версії – чорний трикутник), який передає пласкі таблиці даних від вузла до вузла. Приклад таких портів показаний на рис. 4.9.

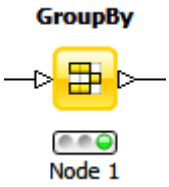


Рисунок 4.9 – Вузол з портами даних

Вузли виконання команд всередині бази даних визнаються за їхніми портами баз даних (коричневий квадрат), що зображено на рис. 4.10.

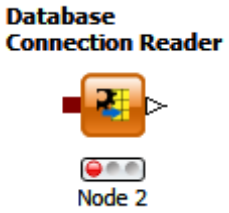


Рисунок 4.10 – Вузол з портом бази даних

Вузли інтелектуального аналізу даних навчають модель, яка передається в відповідний вузол-провісник через синій квадратний PMML-порт, що зображено на рис. 4.11.

Рисунок 4.11 – Вузол з PMML-портом

Всякий раз, коли вузол надає дані, що не вписується плоску структуру таблиці даних, використовується порт загального призначення для структурованих даних (темно блакитний квадрат), приклад якого наведений на рис. 4.12.

Всі порти, не перераховані вище, відомі як «невідомі» типи (сірий квадрат на рис. 4.13).

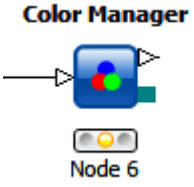


Рисунок 4.12 – Вузол з портом загального призначення
для структурованих даних

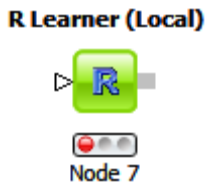


Рисунок 4.13 – Вузол з портом«невідомого» типу

4.3 Вузли, що використовуються в системі

4.3.1 XLS Reader

Цей вузол зчитує електронну таблицю і передає її на свій вихідний порт.Вигляд вузла представлений на рис. 4.14.

Він зчитує дані тільки з одного аркуша. Він може читати тільки числові і текстові дані. В діалоговому меню, представленому на рис. 4.15 представлено багато опцій.

Рисунок 4.14 – Вигляд вузла XLS Reader

В даний час підтримуються наступні типи даних KNIME – String, Double і Int. Продуктивність вузла зчитування обмежена (через обмеження бібліотеки проекту Apache POI). Читання великих файлів займає дуже багато часу і використовує багато пам'яті (особливо файли в XLSX форматі).

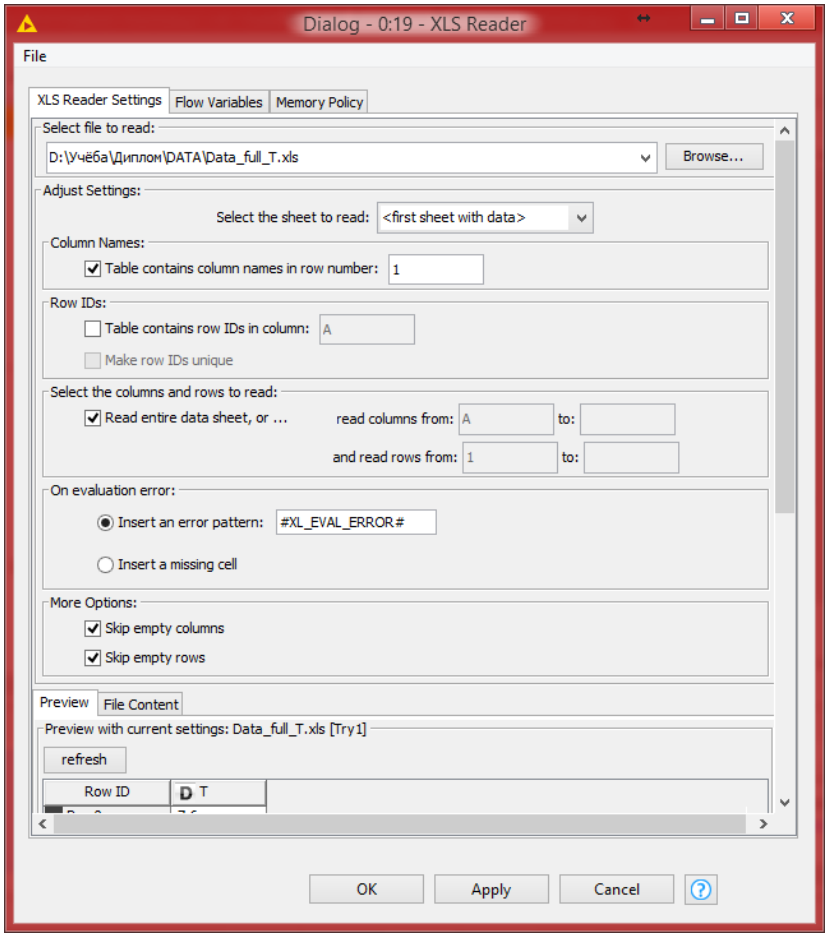


Рисунок 4.15 – Діалогове вікно вузла XLS Reader

На вихідний порт подається зчитана таблиця даних. Доступні такі діалогові опції:

- «Filetoread». Введіть дійсне ім'я файлу. Ви також можете вибрати раніше вибраний файл зі списку, або виберіть файл з діалогу "Browse". Підтримуються XLS і XLSX формати. (Примітка: читання великих XLSX файлів дуже повільне і споживає багато пам'яті.)
- «SheettoRead». Після вибору файлу, ви можете вибрати лист з доступних листів в файлі.
- «ColumnNames». Якщо ви хочете використовувати імена стовпців з електронної таблиці, гляньте "Табличні імена стовпців" і вкажіть номер рядка, який містить імена стовпців (ввести номер (починаючи з одного), не індекс).
- «RowIds». Якщо ви хочете використовувати ідентифікатори рядків з електронної таблиці, встановіть прапорець "Табличні ідентифікатори рядків"

і виберіть стовпець, що містить рядок ідентифікаторів. Введіть мітку шпальти ("A", "B" і т.д.), або номер (починаючи з одного) колонки. Ідентифікатори рядків у листі повинні бути унікальними, в іншому випадку виконання зазнає невдачі. Якщо ви встановите прапорець "Зробити ідентифікатори рядків унікальними ", вузол буде додавати суфікс до дублікатів, забезпечуючи ідентифікаторам рядків унікальність. Для дуже великих наборів даних це може викликати проблеми з пам'яттю.

- «AreaofInterest». Вкажіть область даних на листі, який повинен бути прочитаний. Якщо ви встановите прапорець "Читати весь лист даних", то будуть зчитані всі дані з листа. Це включає в себе діаграми, кордони, колоризацію і т.п. що може створити кілька порожніх рядків або стовпців (дивіться опцію "SkipEmptyRowsorColumns " нижче). Якщо ви хочете зчитати фіксовану область, приберіть галочку і введіть перший і останній рядок, і перший і останній рядок, щоб прочитати. (Останній рядок і стовпець не є обов'язковим, змушуючи його читати до останнього рядка або стовпчика, наданими листом). Для стовпців необхідно ввести мітку ("A", "B" і т.д.), для рядків введення номера.
- «SkipEmptyRowsorColumns». Якщо порожні рядки або стовпці повинні бути видалені з таблиці даних результатів, відзначте відповідну опцію.
- «EvaluationErrorHandling». Вкажіть дані, які вставляються з помилкою. Не всі формули підтримуються всіма додатками електронних таблиць підтримуються XLS ReaderNode. При виникненні помилки під час оцінки формули, вибрані дані вставляються. Ви можете вибрати, щоб вставити осередок, що представляє відсутнє значення, або вставити певний шаблонний рядок. Шаблон викликає перетворення стовпця у тип даних String в разі виникнення помилки.
- «Preview». Вкладка "Попередній перегляд" показує вихід таблицю з поточними настройками в діалоговому вікні. Якщо параметри є недійсними повідомлення про помилку буде відображатися на цій вкладці. Таблиця оновлюється тільки коли кнопка "оновити" натискається. Попередження показано, якщо вміст таблиці не синхронізовані з поточними параметрами.
- «FileContent». Показує вміст обраного листа. Це показує весь вміст

(налаштування не застосовуються там). Імена стовпців і номера рядків тут є ті, які повинні бути вказані в відповідних налаштуваннях полів.

4.3.2 Normalizer

Цей вузол нормалізує значення всіх (числових) стовпців. У діалоговому вікні можна вибрати стовпці, які ви хочете опрацювати. Вигляд вузла представлений на рис. 4.16, його діалогові опції – на рис. 4.17.

Рисунок 4.16 – Вигляд вузла Normalizer

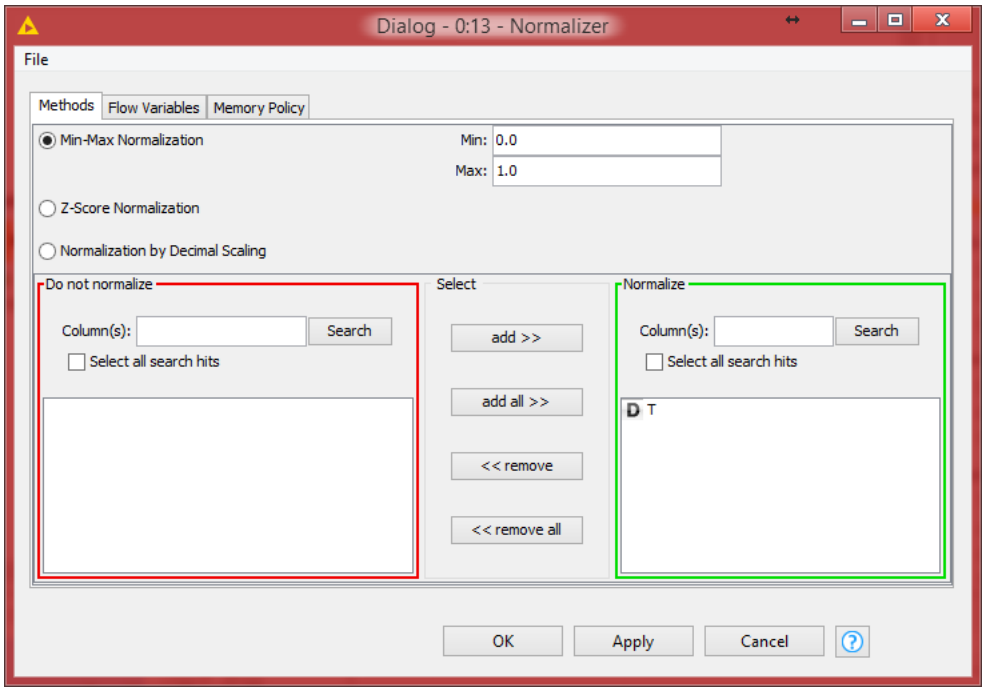


Рисунок 4.17 – Діалогове вікно вузла Normalizer

Наступні методи нормалізації доступні в діалогових опціях:

– «Min-maxnormalization». Лінійне перетворення всіх значень таких, що

мінімальний і максимальний в кожному стовпці, як зазначено.

– «Z-scorenormalization». Лінійне перетворення, при якому значення в кожному стовпці розподілені по Гауссіані (0,1), тобто значення становить 0,0 і стандартне відхилення становить 1,0.
– «Normalizationbydecimalscaling». Максимальні значення в стовпці (як позитивні, так і негативні) ділиться J-раз по 10, поки його абсолютна величина не менше або дорівнює 1. Всі значення в стовпці потім діляться на 10 в ступені у.

На вхід цього вузла подається таблиця з даними для нормалізація, а на виходи – таблиця з нормалізованими даними та модель, що містить параметри нормалізації, які можуть бути використані в " застосуванні нормалізації" вузла для нормалізації тестових даних так само, як були нормалізовані дані для навчання.

4.3.3 Denormalizer

Цей вузол нормалізації вхідних даних відповідно до параметрів нормалізації, як зазначено на вході моделі PMML (зазвичай надходять з вузла Normalizer). Афінне перетворення інвертується і відновлюються вихідні значення. Цей вузол зазвичай використовується після нормалізації тестових даних щоб можливі інші навчальні або прогностичні дані бути перетворені назад у вихідний діапазон.

Вузол має два входи – для параметрів нормалізації та для таблиці, що потребує денормалізації. Вихід вузла один – вхідні дані, перетворені назад у вихідний діапазон.Вигляд вузла представлений на рис. 4.18.

Рисунок 4.18 – Вигляд вузла Denormalizer

4.3.4 ColumnSplitter

Цей вузол розбиває стовпці вхідної таблиці на дві вихідні таблиці. Для цього треба вказати в діалоговому вікні, які стовпці повинні відобразитися у верхній таблиці (лівий список) і нижній таблиці (правий список). Вигляд вузла представлений на рис. 4.19, його діалогове вікно – на рис. 4.20.

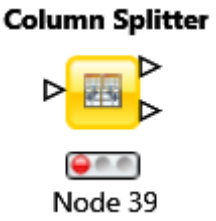


Рисунок 4.19 – Вигляд вузла ColumnSplitter

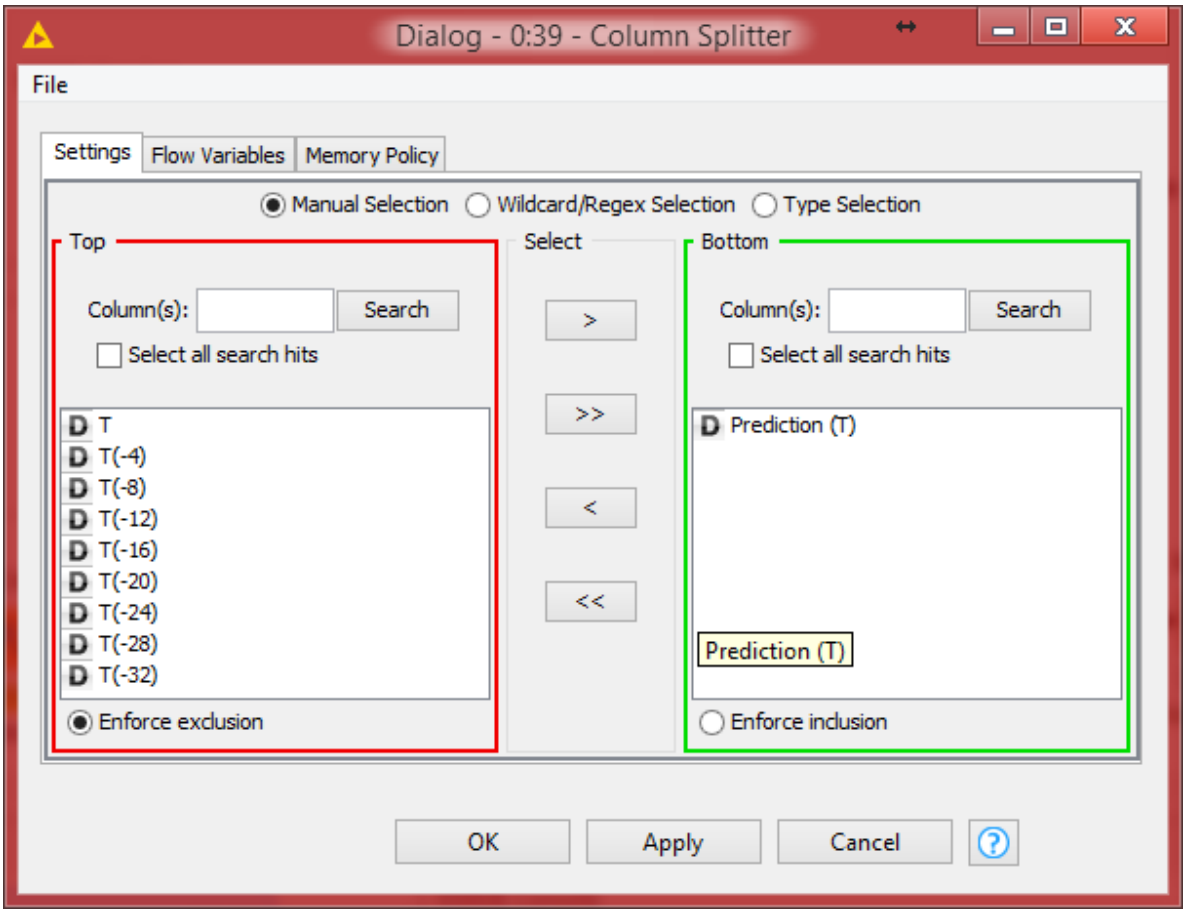


Рисунок 4.20 – Діалогове вікно вузла ColumnSplitter

Список діалогових опцій вузла:

- «Top». Список імен стовпців, які складають першу вихідну таблицю (верхній вихідний порт).
- «EnforceExclusion (Top)». Виберіть цю опцію, щоб виконуваний поточний верхній список залишився незмінним і нижній список змінився щодо специфікацій вхідної таблиці. Якщо деякі вибрані колонки не доступні більше, відображається попередження (Нові колонки будуть автоматично додані в нижній список).
- «Select». Використовуйте ці кнопки для переміщення колонки між верхнім і нижнім списком.
- «Search». Скористайтеся одним із цих полів для пошуку або у верхній або нижній частині списку для деяких імен стовпців або назва підрядків. Повторне натискання кнопки пошуку відзначає наступну колонку, яка відповідає шуканому тексту. Прапорець "Відзначити всі хіти пошуку" викликає всі відповідні стовпці, які будуть відібрані, роблячи їх рухомими між двома списками.
- «Bottom». Список імен стовпців, які складають другу вихідну таблицю (нижній вихідний порт). Виберіть цю опцію, щоб виконуваний поточний верхній список залишився незмінним і нижній список змінився щодо специфікацій вхідної таблиці. Якщо деякі колонки з нижньої таблиці не доступні більше, відображається попередження (Нові колонки будуть автоматично додані в нижній список).

В цьому вузлі один вхідний порт, на який подається таблиця для розділу і два вихідних – вхідна таблиця з колонами, які визначено в лівому списку діалогу і вхідна таблиця з колонами, які визначено в правому списку діалогу.

4.3.5 ColumnAppender

Цей вузол бере дві таблиці і швидко об'єднує їх шляхом додавання стовпців другої таблиці до першої таблиці. Якщо у вхідних таблицях використовуються однакові імена стовпців, до імен стовпців з нижньої таблиці буде додано суфікс "(# 1)".

Діалогові опції цього вузла:

– «Identicalrowkeysandtablelengths». Якщо ключі рядків в обох вхідних таблиць ідентичні (тобто ключі імен стовпців, їх порядок, і їх число повинні відповідати) ця опція може бути відмічена, щоб дозволити більш швидке виконання з меншим споживанням пам'яті. Якщо ключі рядків (імена, порядок, число) не збігаються, то виконання вузла не вдасться. Якщо прапорець не встановлений, то створюється нова результуюча таблиця. Це може привести до тривалого часу обробки. Проте, в цьому випадку кількість рядків у вхідних таблицях може відрізнятися і пропущені значення відповідно доповнюються. Ключі рядків або взяті з першої, другої таблиці, або генеруються абсолютно нові (дивіться варіанти нижче).
– «Userowkeysfrom FIRST table». Доступно тільки якщо опція "Identicalrowkeysandtablelengths" не вибрана. Використовуються ключі рядків першої вхідної таблиці. Якщо перша таблиця довше другої таблиці, пропущені значення вставляються. Якщо перша таблиця коротше, друга таблиця буде скорочуватися.
– «Userowkeysfrom SECOND tab». Доступно тільки якщо опція "Identicalrowkeysandtablelengths" не вибрана. Використовуються ключі рядків другої вхідної таблиці. Якщо друга таблиця довше першої таблиці, пропущені значення вставляються. Якщо друга таблиця коротше, перша таблиця буде скорочуватися.
– «Generatenewrowkeys». Доступно тільки якщо опція "Identicalrowkeysandtablelengths" не вибрана. Ключі рядків генеруються. Якщо одна з вхідних таблиць більше, ніж інша, пропущені значення вставляються відповідно.

Вигляд вузла представлений на рис. 4.21, його діалогове вікно – на рис. 4.22.

Вузол має два вхідних порти, на які відповідно подаються дві таблиці для зведення і один вихідний порт зі зведеною таблицею.

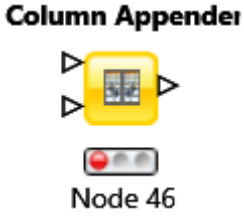


Рисунок 4.21 – Вигляд вузла ColumnAppender

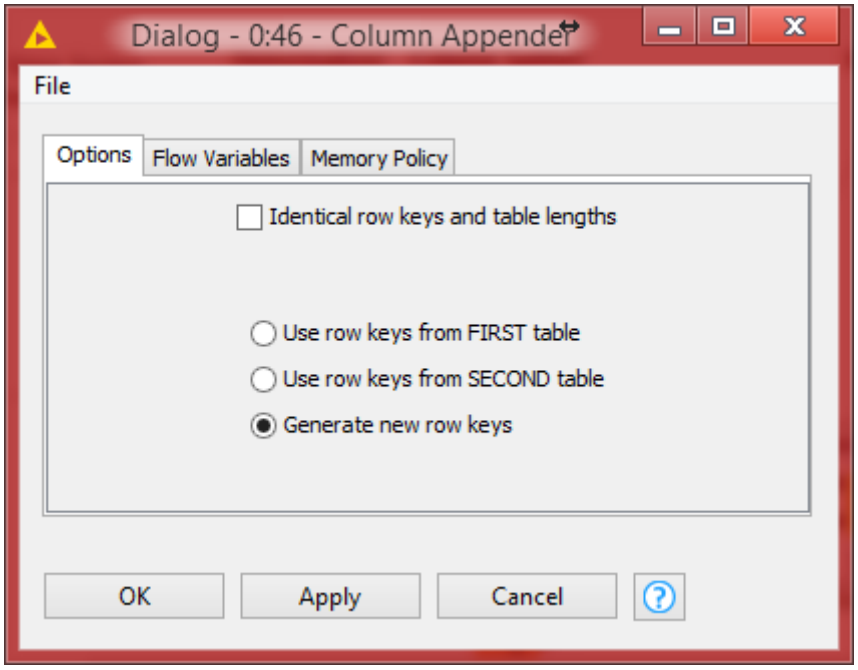


Рисунок 4.22 – Діалогове вікно вузла ColumnAppender

4.3.6 LagColumn

Копіює значення стовпців з попередніх рядків в таблиці. Вузол може бути використаний для того, щоб:

1) зробити копію таблиці і перенести клітини на I кроків вгору (I = інтервал затримки);
2) зробити L копій вибраного стовпця і перенести клітини кожної копії на 1, 2, 3, ... L-1 кроків вгору (L = затримка).

Варіант затримка L в цьому вузлі є корисним для прогнозування часових рядів. Якщо рядки упорядковано відповідно до часу порядку зростання, застосування затримки L до колонки означає вставку L-1 минулих значень стовпця і поточного значення стовпця в один рядок таблиці. Потім таблиця даних може бути використана для передбачення часових рядів.

L і I можуть бути об'єднані для отримання L-1 копій вибраного стовпця, кожен з яких зміщений на I, 2 * я, 3 * я, ... (L-1)*I кроків у зворотному напрямку.Вигляд вузла

представлений на рис. 4.23, його діалогове вікно – на рис. 4.24.

Рисунок 4.23 – Вигляд вузла LagColumn

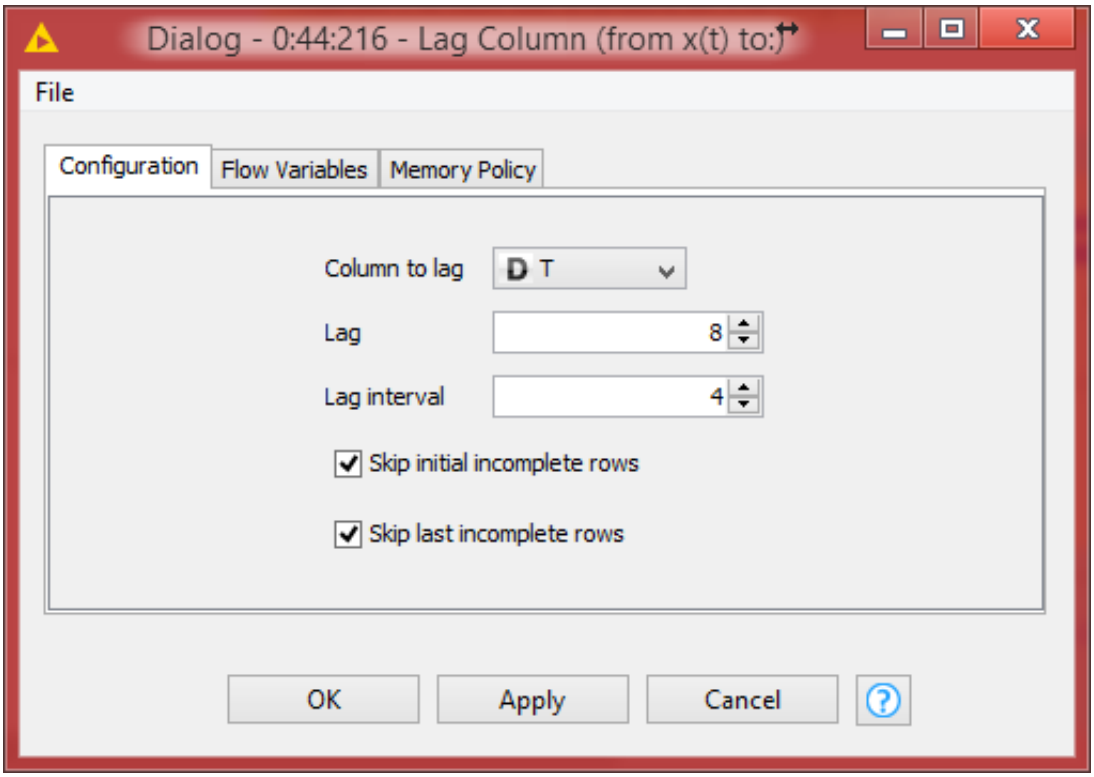


Рисунок 4.24 – Вигляд вузла LagColumn

Діалогові опції цього вузла:

– «Lag». L = затримка визначає, скільки копій стовпців і рядків змістити.
– «LagInterval». I = інтервал затримки (іноді також називають періодичність або сезонність) визначає, скільки копій стовпців і рядків, змістити.
– «Skipinitialincompleterows». Якщо вибрано, перші рядки з таблиці введення опущені на виході, так що стовпці в таблиці відставання повні (якщо опорні дані повні).
– «Skiplastincompleterows». Якщо вибрано реальні дані, що містять відстрочені значення останніх реальних даних у повному рядку не опущені (ніяких

штучних нових рядків). В іншому випадку нові рядки додаються, які містять пропущені значення у всіх колонках, окрім реальних даних.

Вузол має один вхідний порт, на який подається таблиця з даними для застосування до них затримки і один вихідний порт з таблицею даних вже з затримкою.

4.3.7 RProp MLP Learner

Реалізація алгоритму RProp для багатошарових Feedforward мереж типу перцептрон. RProp виконує локальну адаптацію оновлення вагвідповідно до поведінки функції помилки.Вигляд вузла представлений на рис. 4.25, його діалогове вікно – на рис. 4.26.

Цей вузол має два вхідних порти, на які подаються дані для навчання та додатковий (опціональний) порт для PMML-об'єкту, що містить оперції дляпередобробки даних.

Вузол також надає можливість перегляду графік залежності помилки від кількості епох навчання, приклад якого можна побачити на рис. 4.27.

Рисунок 4.25 – Вигляд вузла RProp MLP Learner

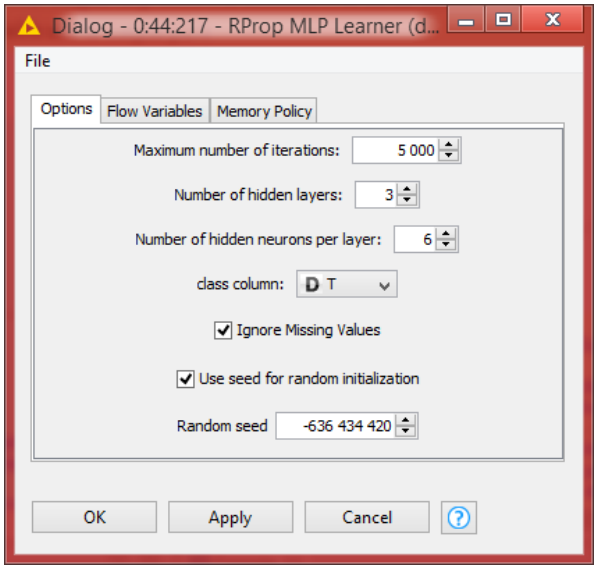


Рисунок 4.26 – Діалогове вікно вузла RProp MLP Learner

Вузол має такі діалогові опції:

- «Maximumnumberofiterations». Кількість навчальних ітерацій.
- «Numberofhiddenlayers». Визначає кількість прихованих шарів в архітектурі нейронної мережі.
- «Numberofhiddenneuronsperlayer». Визначає кількість нейронів, що містяться в кожному прихованому шарі.
- «Classcolumn». Виберіть стовпець, який містить цільову змінну: вона може або бути номінальною або чисельною. Всі значення номінального класу витягуються і призначаються вихідним нейронам. При використанні чисельної цільової змінної (регресії), будь ласка, переконайтеся, що вона нормалізована!
- «Ignoremissingvalues».
- «Useseedforrandominitialization». Seed – база для генератора псевдовипадкових чисел. Цим параметром ініціалізується генератор псевдо-випадкових чисел. Таким чином, при кожному запуску мережа буде мати один і той же початковий стан. Це позбавляє від внесення небажаних флуктуацій в експеримент і дозволяє аналізувати якість навчання при зміні різних параметрів. Якщо цей прапорець встановлений, seed може бути встановлений для ініціалізації помилки і порогів.

– «Randomseed». Seed для генератора випадкових чисел.

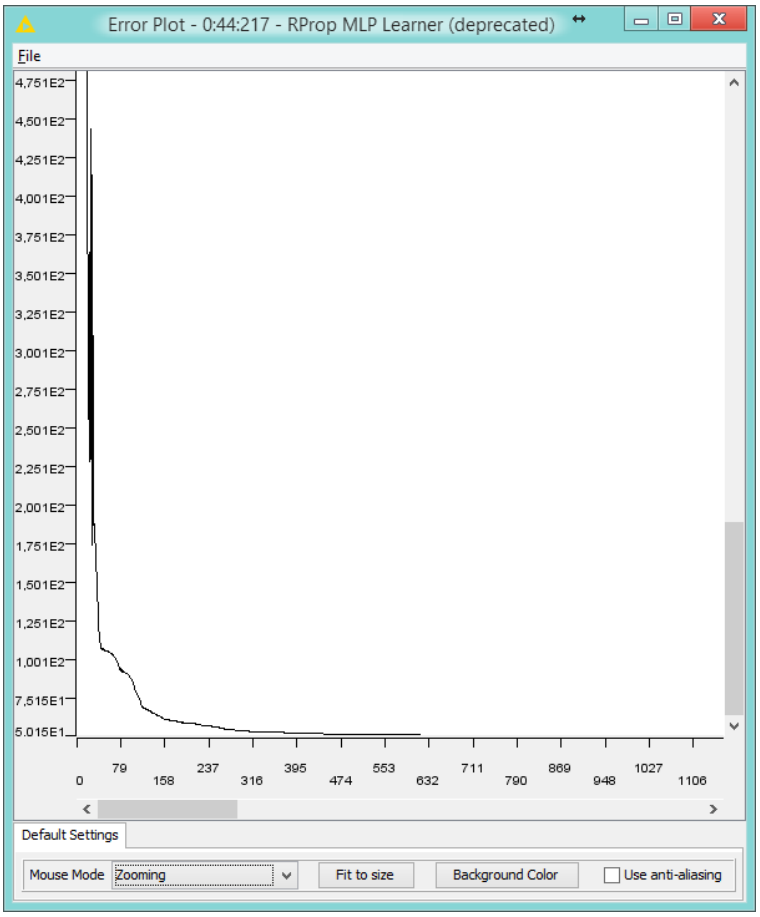


Рисунок 4.27 – Приклад графіку залежності помилки
від кількості епох навчання вузла RProp MLP Learner

4.3.8 MultiLayerPerceptronPredictor

На основі підготовлених моделі багатошарового перцептрону, що подається на вхідний порт даного вузла, обчислюються очікувані вихідні значення. Якщо вихідна змінна є номінальною, виконуються вихід кожного нейрона і клас нейрона переможця. В іншому випадку обчислюється значення регресії. Треба відфільтрувати відсутні значення, перш ніж використо-вувати цей вузол.Вигляд вузла представлений на рис. 4.28, його діалогове вікно – на рис. 4.29.

Рисунок 4.28 – Вигляд вузла MultiLayerPerceptronPredictor

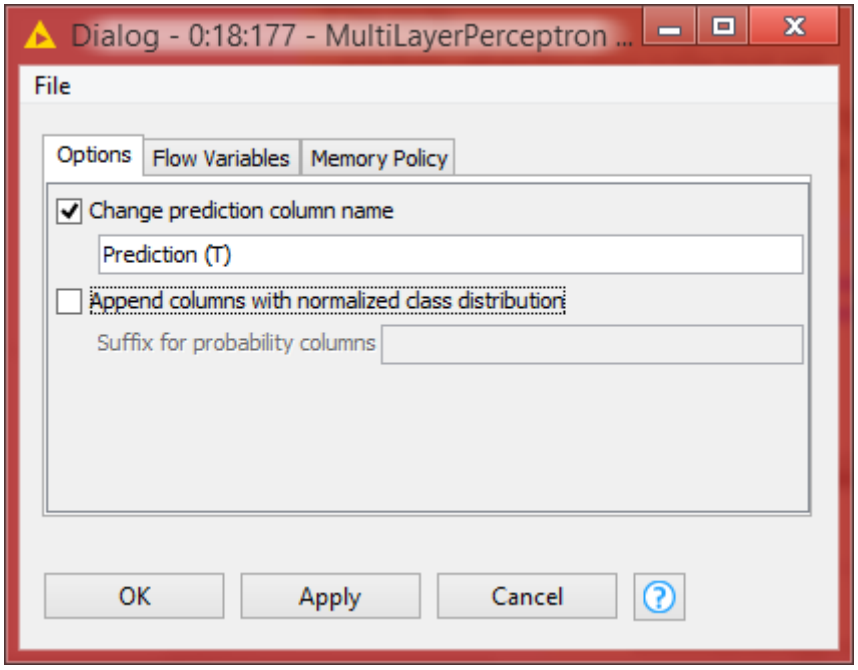


Рисунок 4.29 – Діалогове вікно вузла MultiLayerPerceptronPredictor

Для цього вузла доступні такі діалогові опції:

– «Changepredictioncolumnname». При установці, ви можете змінити ім'я стовпця прогнозування.
– «PredictionColumn». Ім'я стовпця можливість дублювання пророкованої колонці. (За замовчуванням: Прогнозування (колонкаНавчання).)
– «Appendprobabilityvaluecolumnperclassinstance».Коли класифікація закінчена і ця опція встановлена, вірогідність класу додається.
– «Suffixforprobabilitycolumns». Суфікс для нормованих стовпців розподілу. Їх імена типу: P (колонкаНавчання = значення).

Цей вузол має два вхідні порти, на які подається навчена багатошарова нейронна мережа типу перцептрон та таблиця даних для прогнозування відповідно. На вихідний порт подається оброблена таблиця даних із прогнозованими даними. Ці дані необхідно

потім денормалізувати.

4.3.9 NumericScorer

Цей вузол обчислює певні статистики між числовим значенням стовпців (r_i) і спрогнозованими (p_i) значеннями.

Він вираховує:

- коефіцієнт детермінації $R^2 = 1 - SS_{res}/SS_{tot}$ (може бути негативним),
- середню абсолютну похибку ($1/n*\Sigma|p_i-r_i|$),
- середньоквадратичну похибку ($1/n*\Sigma(p_i-r_i)^2$),
- корінь середньоквадратичної похибки ($\sqrt{(1/n * \Sigma(p_i - r_i)^2)}$),
- середню похибок ($1/n*\Sigma(p_i-r_i)$).

Розраховані значення можуть бути перевірені з допомогою цього вузла та/або додатково оброблені за допомогою таблиці виведення.

Вигляд цього вузла представлений на рис. 4.30, його діалогове вікно – на рис. 4.31.

Рисунок 4.30 – Вигляд вузла NumericScorer

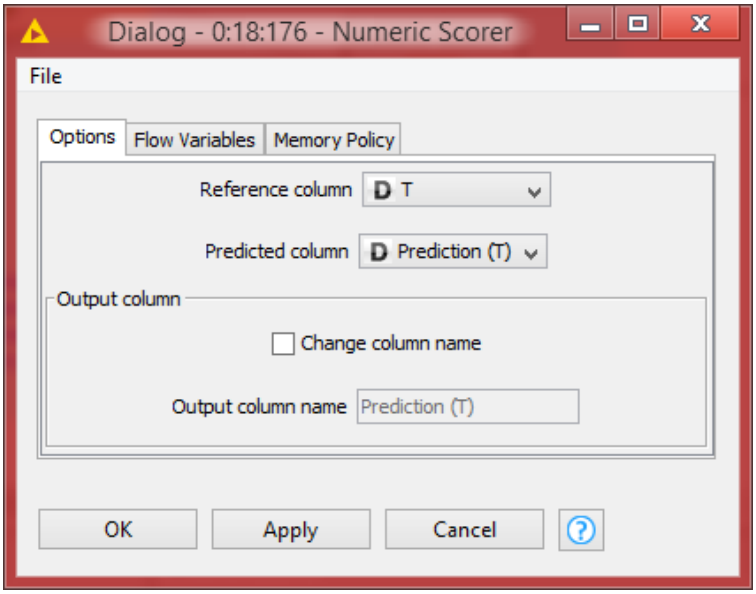


Рисунок 4.31 – Діалогове вікно вузла NumericScorer

Для цього вузла є такі діалогові опції:

– «Referencecolumn». Колонка з правильними, спостережуваними, значен-нями навчальних даних.
– «Predictedcolumn». Колонка зі змодельованими, передбаченими значен-нями даних.
– «Changecolumnname». Змінити ім'я стовпця виведення за замовчуванням.
– «Outputcolumnname». Ім'я стовпця на вихідному порті.

4.3.10 PMMLWriter

Цей вузел пише модель PMML з модельного PMML-порту в сумісний з PMML v4.0 файл або в віддалене сховище, адрес якого зазначається як URL. Якщо PMML-файл іншої версії зчитується вузлом PMML Reader і безпосередньо записується цим вузлом, він перетвориться в PMML v4.0. Якщо модель не дійсна (невідомі типи даних і т.д.) виникає виняток під час виконання.

Якщо місце призначення є віддаленим URL не всі варіанти доступні, тому що в цілому це не представляється можливим визначити, чи існує віддалене розташування. У цьому випадку він завжди буде перезаписан.

Вигляд цього вузла представлений на рис. 4.32, його діалогове вікно – на рис. 4.33.

Рисунок 4.32 – Вигляд вузла PMMLWriter

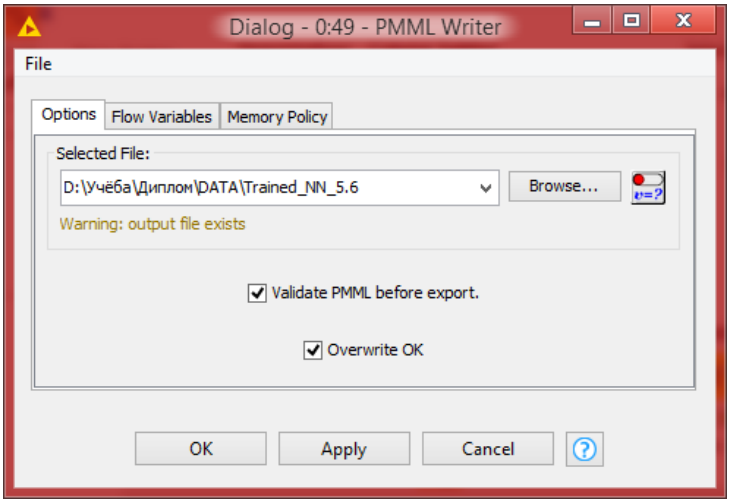


Рисунок 4.33 – Діалогове вікно вузла PMMLWriter

Для цього вузла доступні такі діалогові опції:

– «Outputlocation». Введіть правильне ім'я файлу або URL. Ви також можете вибрати раніше обране місце розташування зі списку, або вибрати локальний файл в діалогу "Browse...".
– «Overwrite OK». Якщо прапорець не встановлено, вузол відмовляється виконуватися, коли вихідний файл існує (запобігання ненавмисному перезапису).

Вузол має тільки один вхідний порт, на який подається модель до запису у PMML.

4.3.11 PMMLReader

Цей вузол зчитує будь-яку модель з файлу. Підключіть вихідний порт до вхідного порту моделі будь-якої моделі, яка вимагає вузол. Введіть розташування

джерела в діалозі конфігурації вузла. При виконанні даного вузол зчитує модель з опису XML в зазначений файл і передає його на своїй вихідний порт. Якщо необов'язковий імпорт PMML підключений і містить операції попередньої обробки KNIME в словник трансформації, то ці дані додаються в модель для читання.

Вигляд цього вузла представлений на рис. 4.34, його діалогове вікно – на рис. 4.35.

Рисунок 4.34 – Вигляд вузла PMMLReader

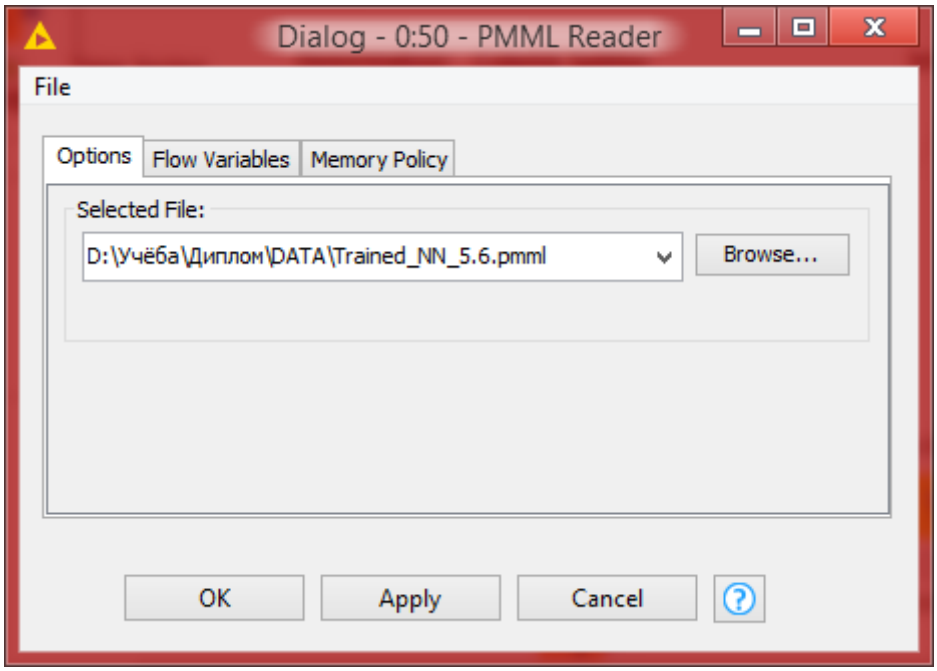


Рисунок 4.35 – Діалогове вікно вузла PMMLReader

Для цього вузла є тільки одна опція – «Selectedfile», що потребує введення шляху до файлу, що представляє модель.

На вихід цього вузла подається оброблена і підготовлена PMML-модель.

4.3.12 TableCreator

Цей вузол дозволяє вручну створити таблицю даних.

Вигляд цього вузла представлений на рис. 4.36, його діалогове вікно – на рис. 4.34.

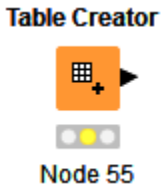


Рисунок 4.36 – Вигляд вузла TableCreator

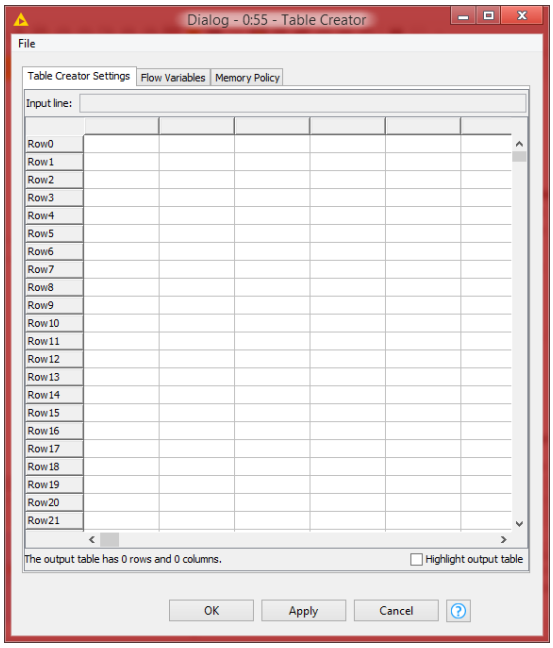


Рисунок 4.37 – Діалогове вікно вузла TableCreator

В діалоговому вікні користувач самостійно вводить дані у клітини таблиці. Доступні такі опції:

– «ColumnProperties». Контекстне меню заголовка стовпчика. Назва та тип можуть бути змінені. Шаблон може бути введений, що викличе відсутню клітку. Крім того, можливі значення домену колонці можуть бути оновлені, вибравши домен. І, ви можете пропустити цю колонку цілком, тобто він не буде включений у вихідну таблицю.

– «Row ID Properties». Контекстне меню заголовка рядка.Використовуйте це, щоб змінити іменування всіх ідентифікаторів рядків. Ідентифікатори рядків побудовані з послідовного ряду з префіксом і суфіксом. Крім префікса і суфікса можна вказати індекс першого ряду.

На вихідний порт цього вузла подається створена користувачем таблиця.

4.3.13 InteractiveTable

Відображає дані у вигляді таблиці. Якщо число рядків невідомо, вид підраховує кількість рядків, коли відкритий. Крім того, рядки можуть бути обрані і виділені.

Вигляд цього вузла представлений на рис. 4.38, вікно перегляду результуючої таблиці – на рис. 4.39.

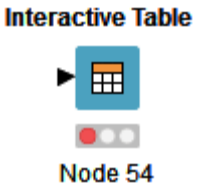
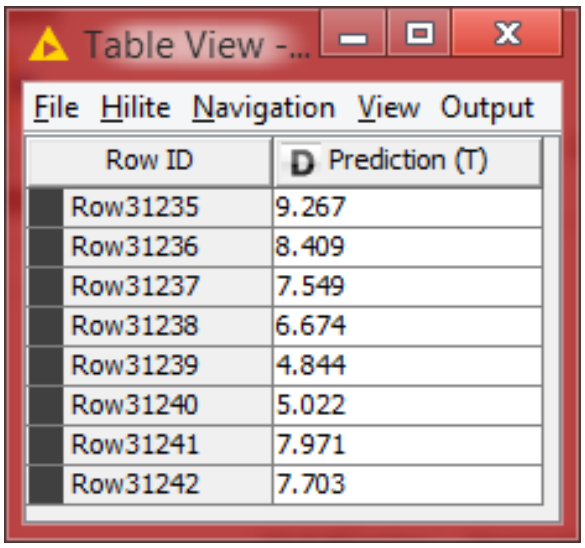


Рисунок 4.38 – Вигляд вузла InteractiveTable



Row ID	Prediction (T)
Row31235	9.267
Row31236	8.409
Row31237	7.549
Row31238	6.674
Row31239	4.844
Row31240	5.022
Row31241	7.971
Row31242	7.703

Рисунок 4.39 – Вікно перегляду результуючої таблиці вузла InteractiveTable

Вузол має тільки один вхідний порт, на який подаються табличні дані для

відображення.

4.4 Модернізація вузла RProp MLP Learner

Одним із серйозних недоліків алгоритму зі зворотним поширенням помилки, використовуваного для навчання багатошарових нейронних мереж, є занадто довгий процес навчання, що робить непридатним використання даного алгоритму для широкого кола завдань, які вимагають швидкого вирішення.

В даний час відомо достатню кількість алгоритмів, що прискорюють процес навчання, таких як: QuickProp, метод сполучених градієнтів, метод Левенберга-Маркара та ін. У даній ситуації розглядається один з таких алгоритмів названий ResilientPropagation (Rprop) який був запропонований М. Рідміллером (M. Riedmiller) і Г. Брауном (H.Braun).

Під стандартним алгоритмом зворотного поширення (backprop) домовимося розуміти алгоритм зворотного поширення помилки. Він використовує метод градієнтного спуску, і, кажучи про недоліки цього алгоритму, будемо мати на увазі недоліки методу градієнтного спуску.

На відміну від стандартного алгоритму Backprop, RProp використовує тільки знаки приватних похідних для підстроювання вагових коефіцієнтів. Алгоритм використовує так зване «навчання за епохами», коли корекція ваг відбувається після пред'явлення мережі всіх прикладів з навчальної вибірки.

Для визначення величини корекції використовується наступне правило:

$$\Delta_{ij}^{(t)} = \begin{cases} \eta + \Delta_{ij}^{(t)}, \dfrac{\partial \mathrm{E}^{(t)}}{\partial \mathrm{w}_{ij}} \dfrac{\partial \mathrm{E}^{(t-1)}}{\partial \mathrm{w}_{ij}} > 0 \\ \eta - \Delta_{ij}^{(t)}, \dfrac{\partial \mathrm{E}^{(t)}}{\partial \mathrm{w}_{ij}} \dfrac{\partial \mathrm{E}^{(t-1)}}{\partial \mathrm{w}_{ij}} < 0 \end{cases}, \qquad (4.1)$$

$$0 < \eta^{-} < 1 < \eta^{+}$$

Якщо на поточному кроці приватна похідна за відповідною вагою w_{ij} поміняла свій знак, то це говорить про те, що остання зміна була великою, і алгоритм проскочив локальний мінімум (відповідну теорему можна знайти в будь-якому підручнику з

математичного аналізу), і, отже, величину зміни необхідно зменшити на η і повернути попереднє значення вагового коефіцієнта: іншими словами необхідно зробити "відкат":

$$\Delta w_{ij}(t) = \Delta w_{ij}(t) - \Delta_{ij}^{(t-1)}, \tag{4.2}$$

Якщо знак похідної не змінився, то потрібно збільшити величину корекції на η^+для досягнення більш швидкої збіжності. Зафіксувавши множники η^- та η^+ можна відмовитися від глобальних параметрів налаштування нейронної мережі, що також можна розглядати як перевагу розглянутого алгоритму перед стандартним алгоритмом Backprop.

Рекомендовані значення для η^-= 0.5, η^+= 1.2, але немає ніяких обмежень на використання інших значень для цих параметрів.

Для того, щоб не допустити занадто великих або малих значень ваг, величину корекції обмежують зверху максимальним Δmax і знизу мінімальним Δmin значеннями величини корекції, які за умовчанням, відповідно, встановлюються рівними 50 і 1.0E-6.

Початкові значення для всіх Δ_{ij} встановлюються рівними 0.1. Знову ж таки, це слід розглядати лише як рекомендацію, і в практичній реалізації можна задати інше значення для ініціалізації.

Для обчислення значення корекції ваг використовується наступне правило:

$$\Delta w_{ij} = \begin{cases} -\Delta_{ij}^{(t)}, & \frac{\partial E^{(t)}}{\partial w_{ij}} > 0 \\ \Delta_{ij}^{(t)}, & \frac{\partial E^{(t)}}{\partial w_{ij}} < 0 \\ 0, & \frac{\partial E^{(t)}}{\partial w_{ij}} = 0 \end{cases}, \tag{4.3}$$

Якщо похідна позитивна, тобто помилка зростає, то ваговий коефіцієнт зменшується на величину корекції, в іншому випадку - збільшується.

Потім підлаштовуються ваги:

(4.4)

$$w_{ij}(t+1) = w_{ij}(t) + \Delta w_{ij}(t),$$

Алгоритм:

1) Проініціалізувати величину корекції Δij
2) Пред'явити все приклади з вибірки і обчислити приватні похідні.
3) Підрахувати нове значення Δij за формулами (4.1) і (4.3).
4) Скорегувати ваги за формулою (4.4).
5) Якщо умова зупинки не виконана, то перейти до 2.

Даний алгоритм сходиться в 4-5 разів швидше, ніж стандартний алгоритм Backprop [9][13].

Виходячи з явних переваг цього методу над старим, було вирішено змінити сирцевий код вузла RProp MLP Learner, підлаштовуючи його під новий алгоритм. Як було згадано раніше, платформа KNIME має відкритий сирцевий код та власну безкоштовну SDK [10][14], яка бере основу в EclipseMars [11][15].

Знайшовши потрібний файл з сирцевим кодом вузла RProp MLP Learner– org.knime.base.data.neural.MultiLayerPerceptron, ми додали зміни у метод run.

```
@Override
publicdouble run(Outline outline, Options options,
ErrorHandlererrorHandler){
Model model = outline.getModel();
JCodeModelcodeModel = model.codeModel;
     JClasshasIdInterface = codeModel.ref("org.dmg.pmml.HasId");
     JClass
hasExtensionsInterface=codeModel.ref("org.dmg.pmml.HasExtensions");
     JClassiterableInterface = codeModel.ref("java.lang.Iterable");
     JClassiteratorInterface = codeModel.ref("java.util.Iterator");
publicintgetDropsCount(intfloorsCount) {
double D = floorsCount;
 D = (Math.sqrt(1 + 8 * D) - 1)/2;
```

[13] [9] Алгоритм обучения RProp – математический аппарат. URL: https://basegroup.ru/community/articles/rprop (дата звернення 15.11.2019)

[14] [10] Герберт Шилдт Java 8. Руководство для начинающих / Шилдт Герберт. М.: Диалектика Вильямс, 2015. 899 с.

[15] [11] Барнет Э. Eclipse IDE Карманный справочник: Пер. с англ. М.:КУДИЦ-ОБРАЗ, 2006. 160 с.

```
return (int)Math.ceil(D);
  }
Collection<? extendsClassOutline>clazzes = outline.getClasses();
for(ClassOutlineclazz : clazzes){
     JDefinedClassdefinedClazz = clazz.implClass;
     FieldOutlineidField = getIdField(clazz);
if(idField != null){
     definedClazz._implements(hasIdInterface);
}
FieldOutlineextensionsField = getExtensionsField(clazz);
     if(extensionsField != null){
     definedClazz._implements(hasExtensionsInterface);
     }
intfl = 200;
int calcFloors4Drop = BalsCalculator.getDropsCount(fl);
for (inti=calcFloors4Drop, s = 0; i>0 && (s=s+i)<=fl; i--)
System.out.print(" " + s);
FieldOutlinecontentField = getContentField(clazz);
     if(contentField != null){
     CPropertyInfopropertyInfo = contentField.getPropertyInfo();
     JFieldVarfieldVar = CodeModelUtil.getFieldVar(contentField);
     JTypeelementType = CodeModelUtil.getElementType(fieldVar.type());
     definedClazz._implements(iterableInterface.narrow(elementType));
     JMethoditeratorMethod = definedClazz.method(JMod.PUBLIC,
iteratorInterface.narrow(elementType), "iterator");
iteratorMethod.body()._return(JExpr.invoke("get" +
propertyInfo.getName(true)).invoke("iterator"));
     }}
returntrue; }
```

Після компіляції модифікований вузол RProp MLP Learner очікувано зменив затрати часу на навчання штучної нейронної мережі.

4.5 Налаштування і використання системи короткострокового синоптичного прогнозування з використанням нейронних мереж

4.5.1 Обробка даних попередніх спостережень

Для навчання нейронної мережі використовувались дані спостережень за температурою повітря на метеостанції міста Одеси (WMOID33837).Дані знаходяться у вільному доступі для перегляду і завантаження на веб-сайті http://rp5.ua/. Період спостережень – з 02:0001.02.2005 до 14:00 05.11.2018.

Сайт розроблений та супроводжується компанією (ТОВ) "Розклад Погоди", Санкт-Петербург, Росія, з 2004 року. Компанія має ліцензію Федеральної служби Росії по гідрометеорології та моніторингу навколишнього середовища (Росгідромет) на формування і ведення банків даних в області гідрометеорології та суміжних з нею областях.

Інформація про фактичну погоду надходить з двох джерел: з сервера даних міжнародного обміну, NOAA, США, і автоматичної системи передачі даних (АСПД) Росгідромету. Дані з другого джерела надаються згідно з договором між Федеральним державною бюджетною установою "Головний центр інформаційних технологій та метеорологічного обслуговування авіації Федеральної служби по гідрометеорології та моніторингу навколишнього середовища" та ТОВ "Розклад Погоди". Сервер NOAA надає дані спостережень в форматах SYNOP і METAR, АСПД Росгідромету – в форматах SYNOP і КН-02 [12][16)].

Дані представлені як стовбець чисел електронної таблиціExcel формату XLS. Тож перший крок – зчитати дані в програмі аналізу даних KNIME. Для цього використовується вузол XLSReader з назвою «1. Зчитування файлу з відомими даними для навчання». Налаштування вузла показані на рис. 4.41.

[16)] [12] Сайт погоди. URL: http://rp5.ua/docs/about/ru (дата звернення 15.11.2019)

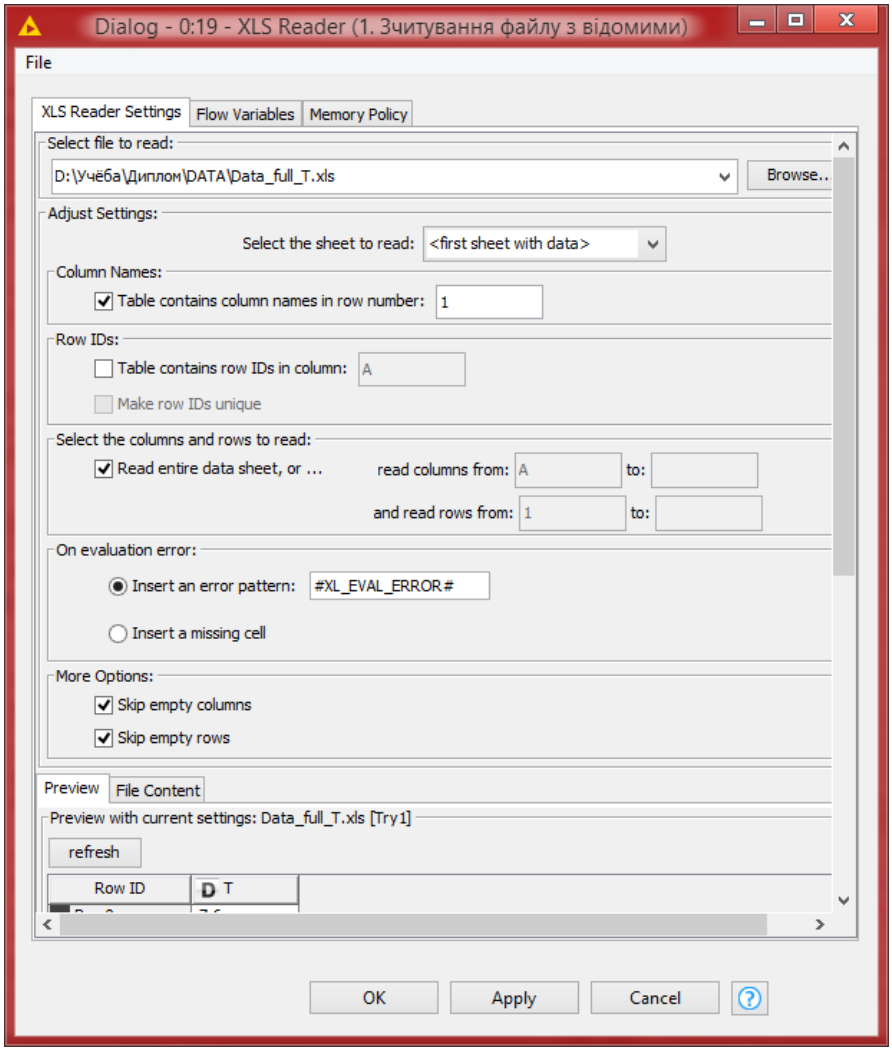


Рисунок 4.41 – Налаштування вузла «1. Зчитування файлу з відомими даними для навчання»

Вибирається файл з назвою Data_full_T.xls, що містить дані попередніх спостережень. Позначається, що таблиця містить назви стовпців у першому рядку. Також позначається, що треба прочитати весь лист даних, пропускаючи пусті стовпці і рядки.

В даному випадку таблиця містить один стовпець T, який і передається на вихідний порт в якості таблиці з одним стовпцем.

Далі ця таблиця потрапляє на вхідний порт вузла Normalizer з назвою «2. Нормалізація даних для навчання». Задача цього вузла – нормалізувати дані для потреб вузла-вчителя нейронної мережі RProp MLP Learner. Налаштування даного вузла Normalizer показані на рис. 4.42.

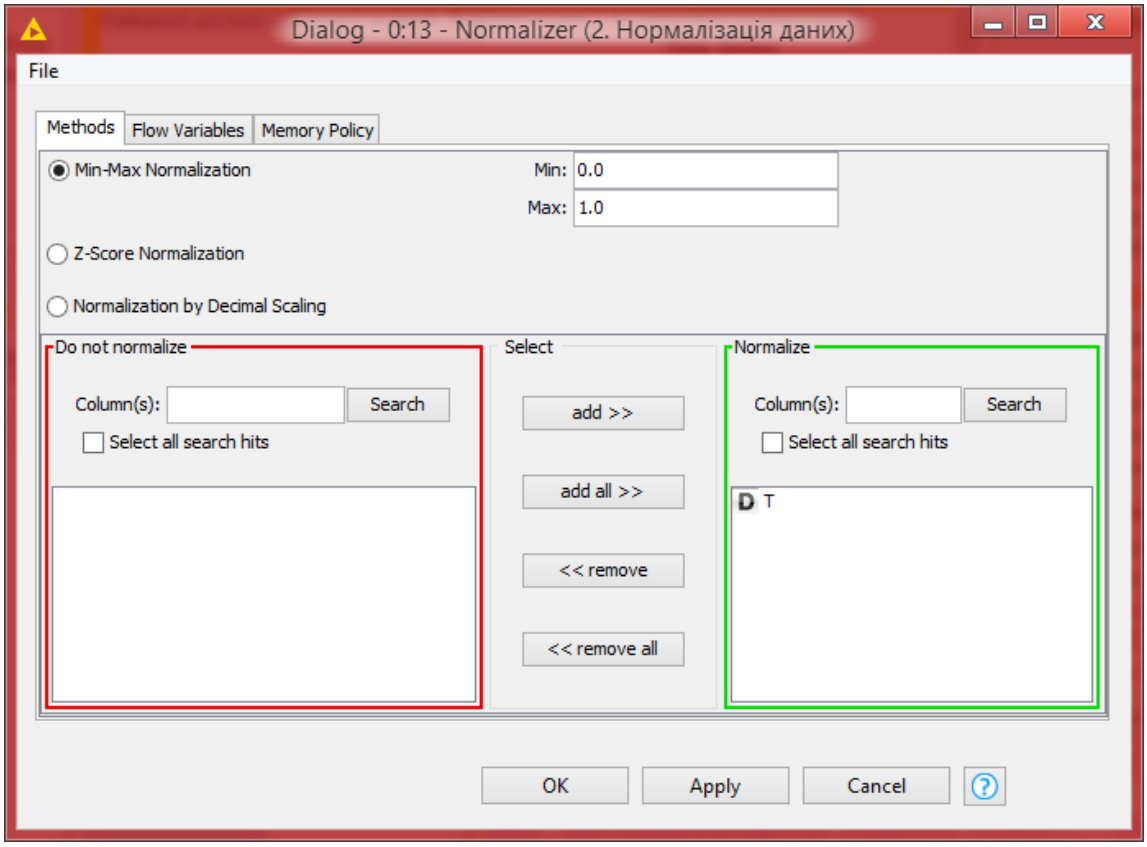


Рисунок 4.42 – Налаштування вузла «2. Нормалізація данихдля навчання»

В даному випадку використовується проста min-maxнормалізація для стовпця T. Нормалізований стовпець передається з виходу вузла «2. Нормалізація данихдля навчання» на вхід даних мета-вузла «3. Навчання нейронної мережі».

4.5.2 Навчання моделі нейронної мережі

Навчання моделі нейронної мережі забезпечує метавузол TimeSeriesAuto-PredictionTraining під назвою «3. Навчання нейронноїмережі». Вміст цього метавузла показаний на рис. 4.43. Цей метавузол навчає модель нейронної мережі передбачити значення часового ряду на основі його N минулих значень. Цей вузол:

– Створює N минулих значень часового ряду, який ми хочемо прогнозувати.
– Навчає модель нейронної мережі з цифровими даними прогнозувати вибраний часовий ряд на основі його N минулих значень.

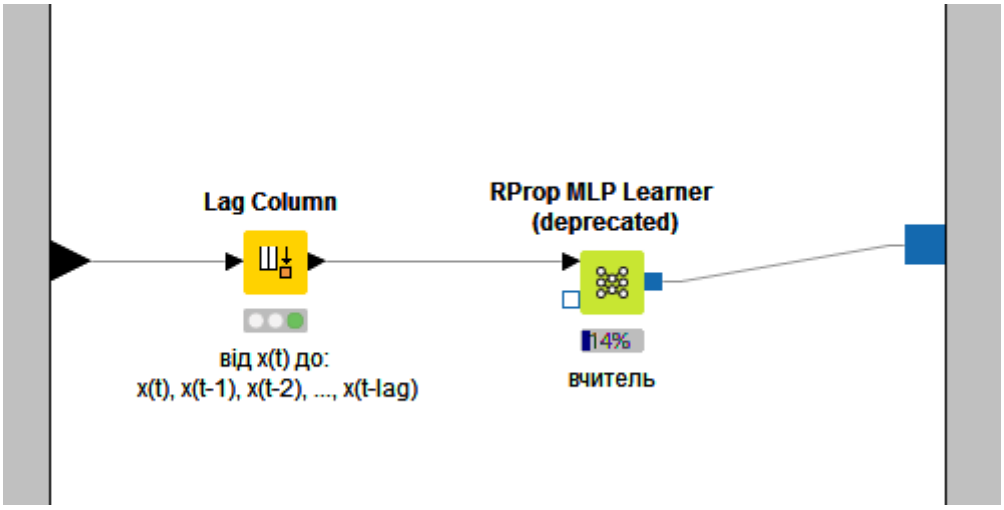


Рисунок 4.43 – Вміст метавузла «3. Навчання нейронної мережі»

Першим в метавузлі отримує дані вузол LagColumn, який створює деяку кількість попередніх зразків для навчання моделі для прогнозування, тобто він генерує: x(t-запізнювання), x(t-запізнювання+1), ... x(t-2), x(t-1), щоб передбачити x(t). Налаштування цього вузла показані на рис. 4.44.

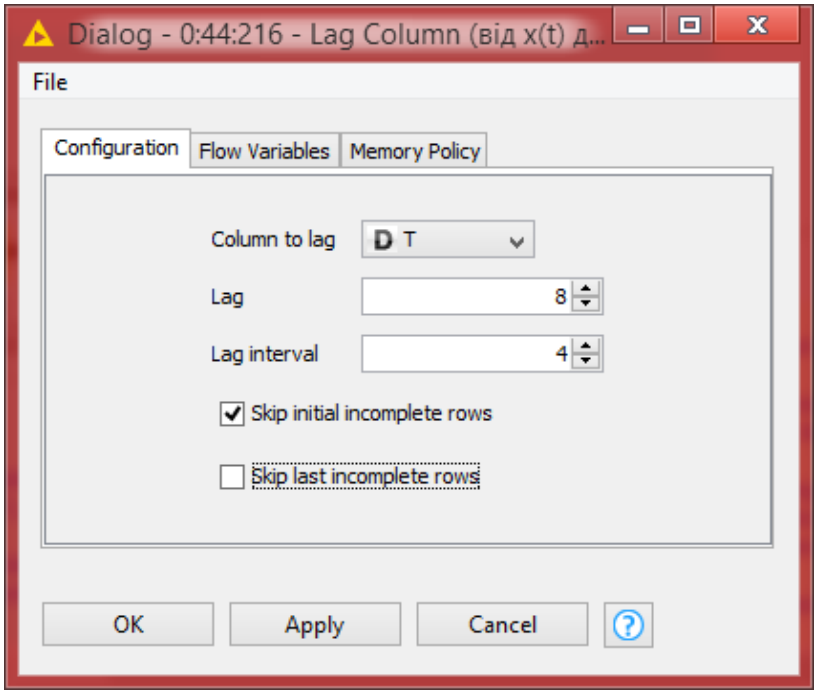


Рисунок 4.44 – Налаштування вузла LagColumn

Налаштування вузла показують, що робиться вісім копій попередніх значень в кожному рядку таблиці, але з інтервалом в чотири клітинки – це якраз і дає нам

цільовий інтервал в 12 годин. Перші неповні рядки пропускаються. Після виконання цього вузла на вихідний порт передається таблиця з 31243 рядків і 9 стовпців.

Далі ця таблиця з вихідного порту вузла LagColumn передається на вхідний порт вузла RProp MLP Learner, який моделює роботу багатошарового перцептрону. Налаштування цього вузла показані на рис. 4.45.

Для налаштування модифікованого мною вузла я обрала 40 000 як максимальне число ітерацій навчання моделі нейронної мережі. Архітектура мережі складається з трьох прихованих шарів нейронів по 6 нейронів в кожному. Звичайно, стовпцем для прогнозування вибираю T. Помічаю прапорець «Ігнорувати пропущені значення» про всяк випадок і обираю використання «сім'я» для ініціалізації «випадкових» чисел для корегування вагів.

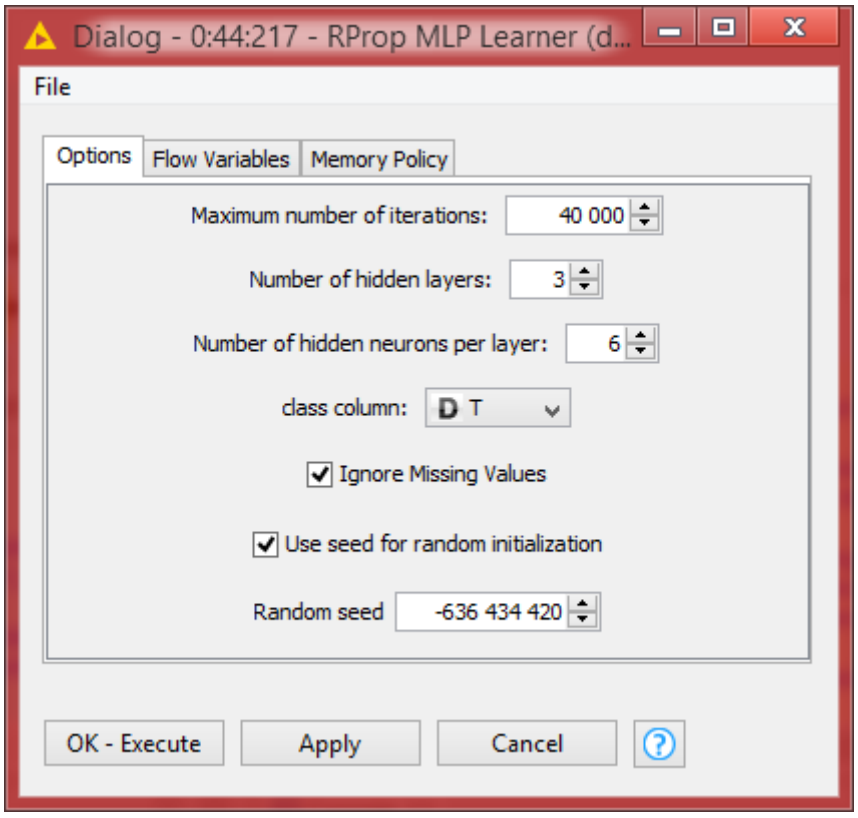


Рисунок 4.45 – Налаштування вузла RProp MLP Learner

На вихідному порті цього вузла після виконання з'явиться навчена модель нейронної мережі, яка передається на вихідний порт метавузла «3. Навчання нейронноїмережі».

4.5.3 Збереження моделі нейронної мережі

Для збереження моделі нейронної мережі використовується вузол PMMLWriter «4. Запис моделі навченої нейронної мережі у PMML-файл», який отримує на свій вхідний порт модель навченої нейронної мережі і записує її у файл формату PMML, шлях до якого вказується в налаштуваннях цього вузла (див. рисунок 4.46)

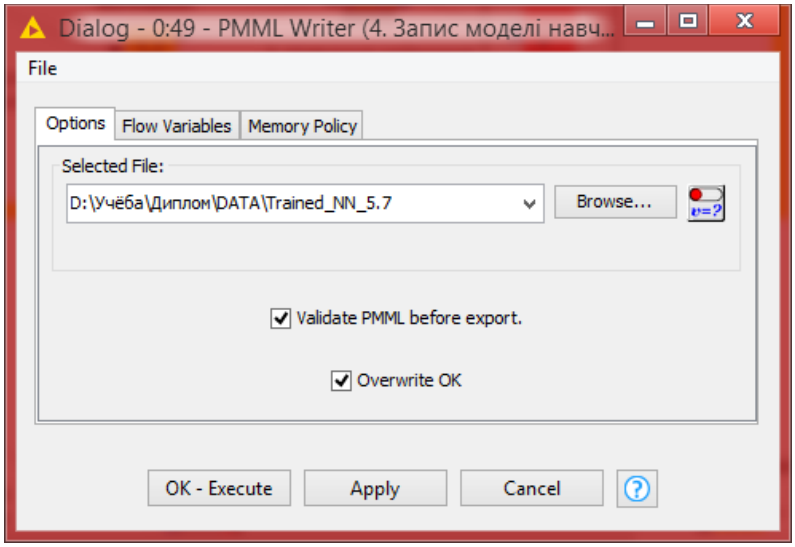


Рисунок 4.46 – Налаштування вузла «4. Запис моделі навченої нейронної мережі у PMML-файл»

В налаштуваннях помічені прапорцем опції «Перевірити на правильність PMML-модель перед експортом» та «Дозволяється перезапис файла».

Коли модель навченої нейронної мережі збережена, її можна неодноразово використовувати для прогнозу на місцевості, де велися спостереження (в даному випадку, це місто Одеса, але якщо навчити нейронну мережу на інших даних, то й інші міста) – це значно швидше і ефективніше того, щоб кожного разу вчити модель нейронної мережі наново.

4.5.4 Короткострокове синоптичне прогнозування з використанням нейронної мережі

Для отримання короткострокового синоптичного прогнозу температури повітря

використовуючи розроблену систему треба спочатку прочитати PMML-файл з моделлю навченої нейронної мережі. Для цього є вузол PMMLReaderіз назвою «5. Зчитування PMML-моделінавченої нейронної мережі». Він зчитує PMML-файл, шлях до якого заданий у налаштуваннях вузла (рис. 4.47).

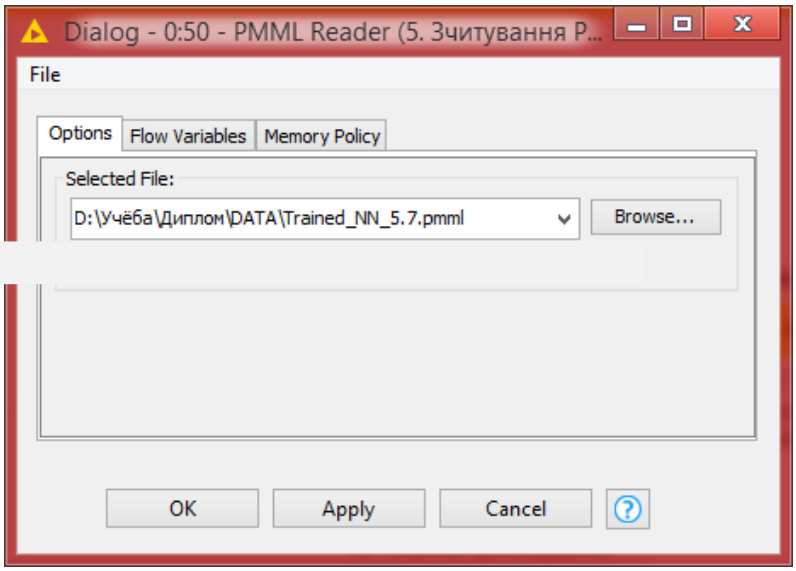


Рисунок 4.47 – Налаштування вузла «5. Зчитування PMML-моделінавченої нейронної мережі»

Потім треба занести у вузол TableCreator «6. Таблиця з добовими показниками температури» показники температуриповітря минулої (минаючої) доби, щоб отримати прогноз на поточну (наступну) добу. Важливо пам’ятати, що показники мають бути взяті з інтервалом у 3 години, тобто на добу повинно бути 8 показників температури повітря. Приклад наведено на рис. 4.48.

Після виконання вузлів «5. Зчитування PMML-моделінавченої нейронної мережі» і «6. Таблиця з добовими показниками температури», їх результати відповідно передаються на вхідні порти метавузлаTimeSeriesAuto-PredictionPredictor «4. Прогнозування за допомогою нейронної мережі та обчислення похибки». Там на ці дані вже чекає вузол MultiLayerPerceptronPredictor, який використовує модель навченої нейронної мережі, отриманої від вузла «5. Зчитування PMML-моделінавченої нейронної мережі» для прогнозування температури повітря на наступну (поточну) добу по даних, що передає на інший вхідний порт вузол «6. Таблиця з добовими показниками температури».

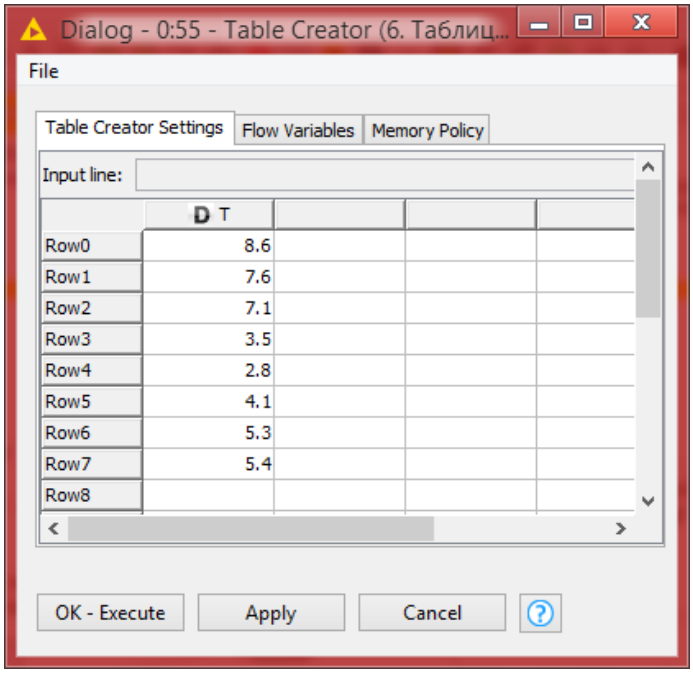


Рисунок 4.48 – Вручну занесені показники в таблицю вузла «6. Таблиця з добовими показниками температури»

Налаштувань вузла MultiLayerPerceptronPredictor (рис.4.49) не багато – всього лиш вибирається назва для ствопця зі спрогнозованими значеннями, в даному випадку – Prediction (T).

Спрогнозовані значення передаються через вихідний порт вузла Multi-LayerPerceptronPredictor на вихідний порт метавузлаTimeSeriesAuto-PredictionPredictor «4. Прогнозування за допомогою нейронної мережі та обчислення похибки» (вміст цього вузла можна побачити на рис. 4.50) та на вхідний порт вузла NumericScorer, який обчислює коефіцієнт детермінації R^2, середню абсолютну похибку, середньоквадратичну похибку, корінь середньоквадратичної похибки і середню похибок – ці дані передаються на вихідний порт вузла NumericScorer та метавузлаTimeSeriesAuto-PredictionPredictor «7. Прогнозування за допомогою нейронної мережі та обчислення похибки».

Дані про похибки передаються на вхідний порт вузла InteractiveTable «10. Похибка прогнозування», який відображає дані у вигляді таблиці (рис. 4.51).

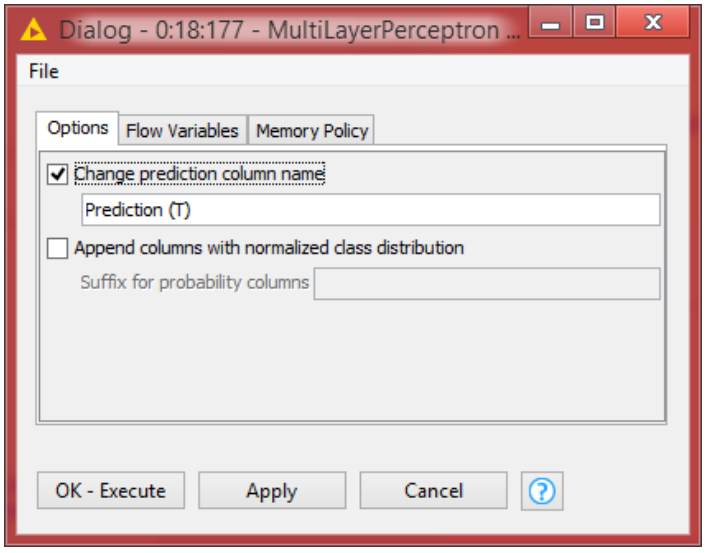


Рисунок 4.49 – Налаштування вузла MultiLayerPerceptronPredictor

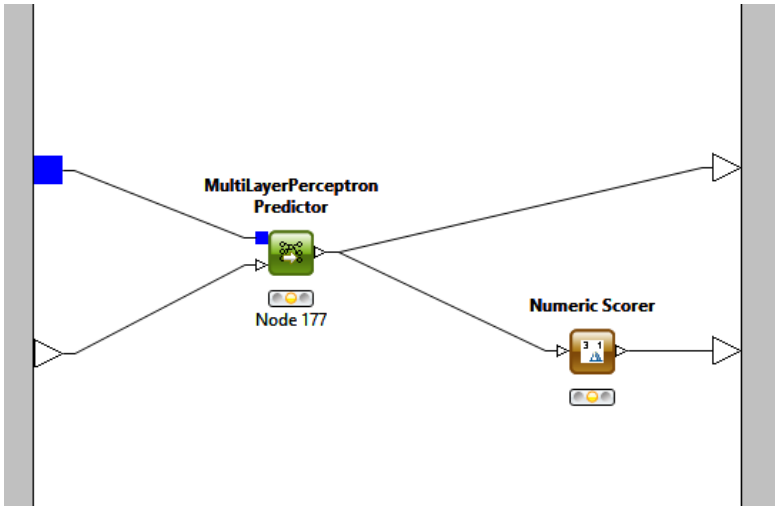


Рисунок 4.50 – Вміст метавузла«7. Прогнозування за допомогою нейронної мережі та обчислення похибки»

При заданих параметрах, похибки складають:

– коефіцієнт детермінаціїR^2= 0,012
– середня абсолютна похибка = 0,034,
– середньоквадратична похибка= 0,001
– корінь середньоквадратичної похибки = 0,027,
– середня похибок = 0,024.

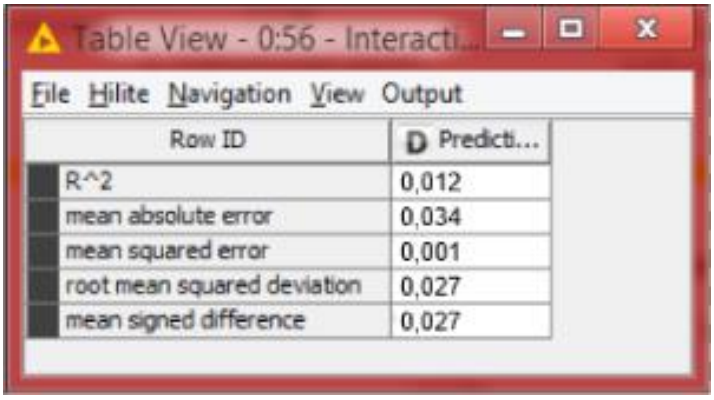


Рисунок 4.51 – Дані про похибку прогноза у режиму перегляду таблиці вузла «10. Похибка прогнозування»

Як можна побачити з приведених даних, похибка прогнозу складає менше 4%, що досить прийнятно.

Далі вузол «11. Денормалізаціяспрогнозованогостовпця даних» денормалізує дані прогнозу, а вузол «11. Результат прогнозування» відображає денормалізовані результати короткострокового прогнозу температури повітря в табличному вигляді, що визивається з контекстного меню вузла.

Система прогнозування розділена на дві частини – навчання, що представлена в додатку А і прогнозування, представлена в додатку Б.

4.5.5 Результати прогнозування

Шляхом проведення ряду експериментів була побудована система короткострокового прогнозування температури повітря, заснована на багатошаровій нейронній мережі зворотного поширення похибки, яка дозволила отримати задовільні відхилення прогнозованих значень на виході (за відомими фактичними значеннями), а також контрольного відтворення даних, які є складовими навчальної множини. У табл. 4.1 зведені результати прогнозів.

Таблиця 4.1 – Результати прогнозу

Параметри нейромережі	Прогноз
Кількість шарів і структура нейромережі	3 (6-6-6)

Крок прогнозування	12 годин
Вид прогнозування	однокроковий
Епохи	40000
Середньоквадратична помилка прогнозу	1,56%

Результати короткострокового прогнозу температури повітря відображає вузол «11. Результат прогнозування» в табличному вигляді (рис. 4.52), що визивається з контекстного меню вузла.

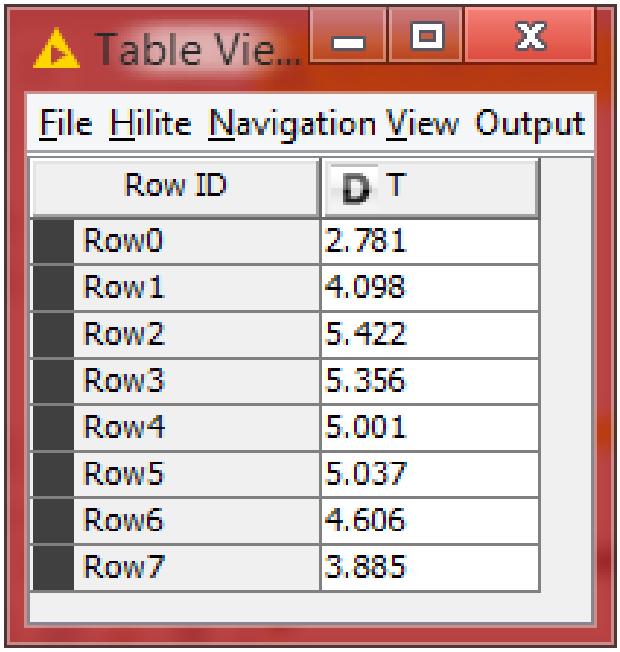


Рисунок 4.52 – Таблиця з результатами прогнозу

Відмінності результату прогнозування від реальних даних спостережень приведені в табл. 4.2.

Таблиця 4.2 – Порівняння спрогнозованих і реальних даних

Дата та час спостереження	Спостережувані дані	Спрогнозовані дані
04.11.2018 02:00	3,9°C	3,885
06.11.2018 23:00	4,6°C	4,606

06.11.2018 20:00	5,1°C	5,037
06.11.2018 17:00	5,1°C	5,001
06.11.2018 14:00	5,4°C	5,356
06.11.2018 11:00	5,3°C	5,422
06.11.2018 08:00	4,1°C	4,098
06.11.2018 05:00	2,8°C	2,781

ВИСНОВКИ

У магістерській роботі було розглянуто рішення задачі прогнозування часових рядів за допомогою штучних нейронних мереж зворотного поширення похибки. Була розроблена система короткострокового синоптичного прогнозування температури повітря в місті Одеса на основі платформи інтелектуального аналізу даних KNIME. Для навчання штучної нейронної мережі був спеціально модифікований метод зворотнього поширення помилки на основі алгоритму методу Rprop, який дозволяє значно прискорити навчання нейронної мережі. Виконано однокроковепрогнозування температури повітря в місті Одеса на підставі даних за 2005-2018 рр.

Відзначимо, що рішення подібної задачі традиційно здійснюється шляхом застосування методів теорії ймовірностейі математичної статистики, проте штучні нейронні мережі є хорошою альтернативою і дають задовільний результат прогнозування.

В роботі був наданий огляд сучасних моделей та методів технологій інтелектуальних обчислень, на підставі якого обґрунтоване застосування багатошарових штучний нейронних мереж саме для вирішення задач прогнозування часових рядів. Розглянути основні компоненти штучного нейрона та архітектури побудови штучних нейронних мереж. Були розглянути класичні методи та засоби прогнозування, до яких у першу чергу відносяться статистичні моделі та методи.

Аналіз результатів однокрокового прогнозуваннятемператури повітря з використання штучної нейронної мережі показав, що отримані задовільні відхилення прогнозованих значень на виході (за відомими фактичними значеннями). Середня абсолютна похибка прогнозу не перевищила 4% .

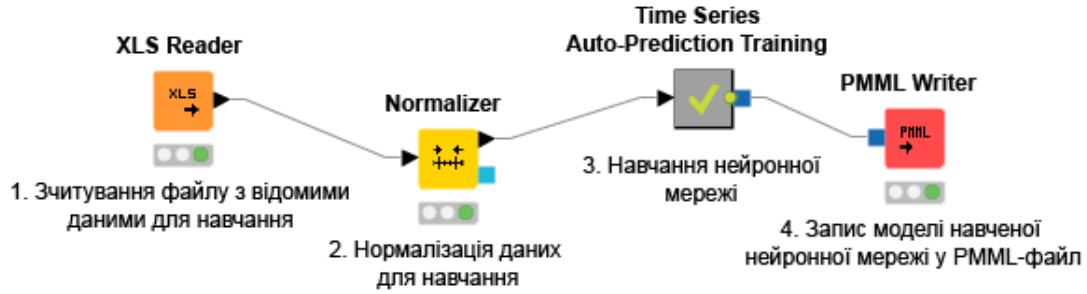
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Кому и зачеп нужен прогноз? URL: http://www.primpogoda.ru/articles/prosto_o_pogode/komu_i_zachem_nuzhen_prognoz/ (дата звернення 15.11.2019)
2. Безручко Б.П., Смирнов Д.А. Математическое моделирование и хаотические временные ряды. Саратов: «Колледж», 2005. 320 с.
3. Дымников В.П., Филатов А.Н. Основы математической теории климата, М.: ВИНИТИ, 1994. 254 с.
4. Использование метода ближайших ложных соседей для предобработки временных рядов. URL: http://moyuniver.net/ispolzovanie-metoda-blizhajshix-lozhnyx-sosedej-dlya-predobrabotki-vremennyx-ryadov/ (дата звернення 15.11.2019)
5. Метод головних компонент. URL: http://bko.com/book_346_glava_69_%C2%A7_2.6._%D0%9C%D0%B5%D1%82%D0%BE%D0%B4_%D0%B3%D0%BE%D0%BB%D0%BE%D0%B2%EF%BF%BD.html (дата звернення 15.11.2019)
6. Решение задачи прогнозирования с помощью нейронных сетей. URL: http://www.rusnauka.com/1-NIO_2011/Informatica/78176.doc.htm (дата звернення 15.11.2019)
7. Искусственные нейронные сети. URL: http://victoria.lviv.ua/html/oio/html/theme5_rus.htm (дата звернення 15.11.2019)
8. KNIME. Wikipedia. URL: https://en.wikipedia.org/wiki/KNIME (дата звернення 15.11.2019)
9. Алгоритм обучения RProp – математический аппарат. URL: https://basegroup.ru/community/articles/rprop (дата звернення 15.11.2019)
10. Герберт Шилдт Java 8. Руководство для начинающих / Шилдт Герберт. М.: Диалектика Вильямс, 2015. 899 с.
11. Барнет Э. Eclipse IDE Карманный справочник: Пер. с англ. М.:КУДИЦ-ОБРАЗ, 2006. 160 с.
12. Сайт погоди. URL: http://rp5.ua/docs/about/ru (дата звернення 15.11.2019)

Д О Д А Т К И

ДОДАТОК А
ПІДСИСТЕМА НАВЧАННЯ

I. Навчання штучної нейронної мережі. Виділити вузли 1-4 і запустити на виконання.



ДОДАТОК Б

ПІДСИСТЕМА ПРОГНОЗУВАННЯ ТА ВІДОБРАЖЕННЯ РЕЗУЛЬТАТУ