

Магістерська дипломна робота на тему

Метод та засіб ранжування операторів CRM-систем засобами нейромереж

Виконав студент групи 2КІ-16м
Комаров В. Л.

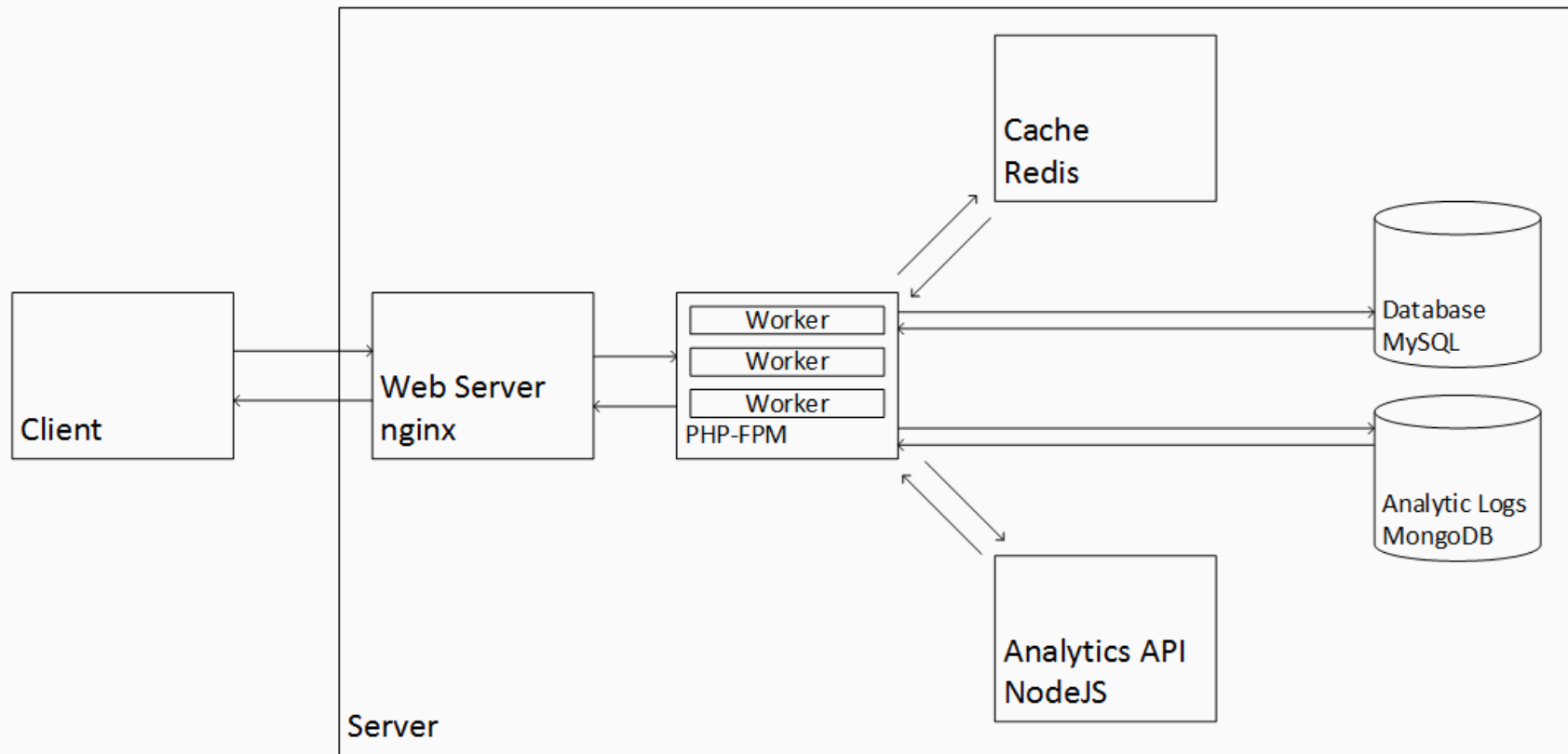
Керівник
к.т.н., доц. Трояновська Т. І.

Мета розробки

- Автоматизувати процес оцінювання
- Оптимізувати процес ранжування
- Покращити процес оцінювання та ранжування

Використані технології

- Docker та Docker-Compose
- PHP 7.2, Laravel
- NodeJS, ExpressJS
- Synaptic



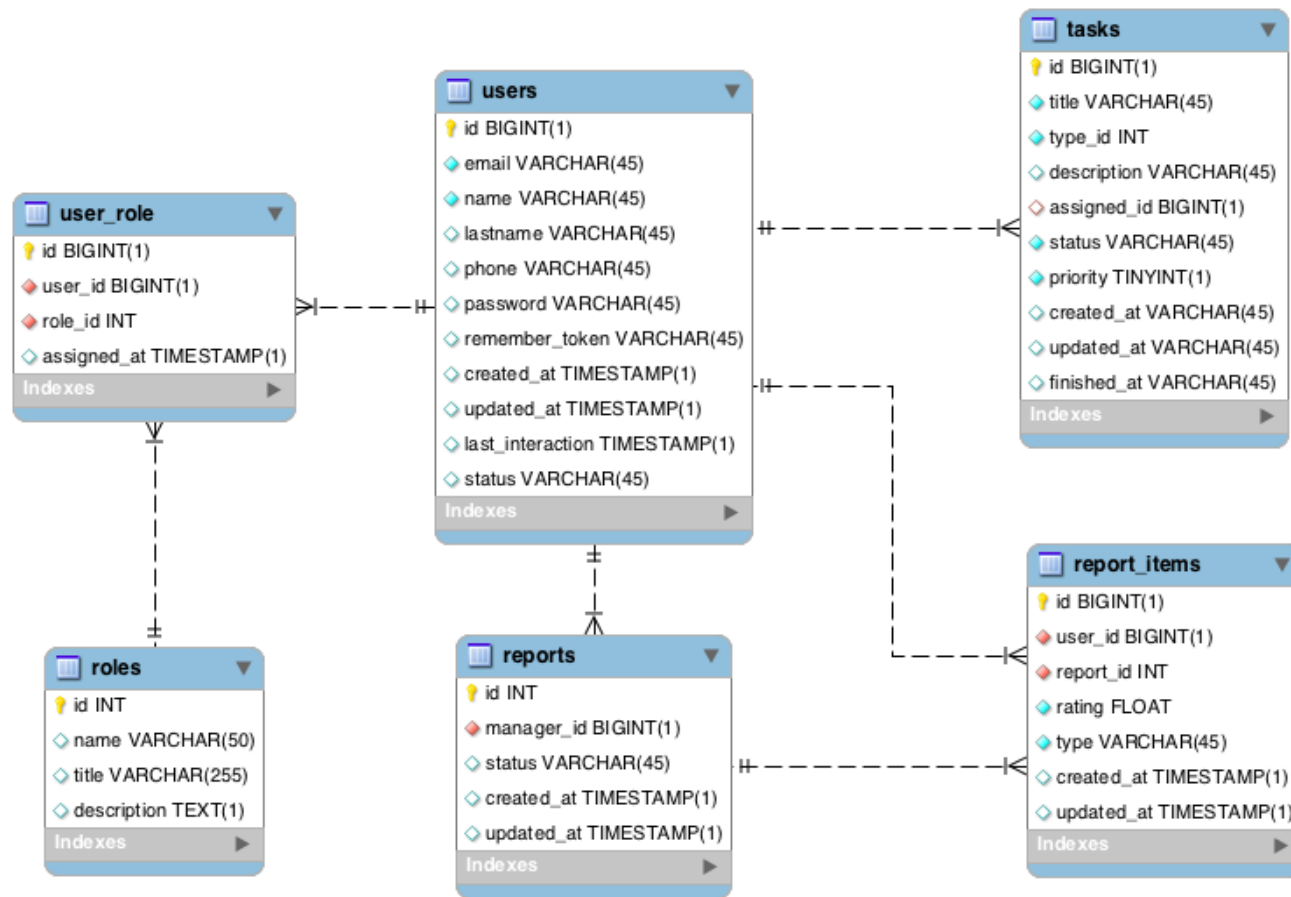


Схема бази даних прототипу CRM-системи

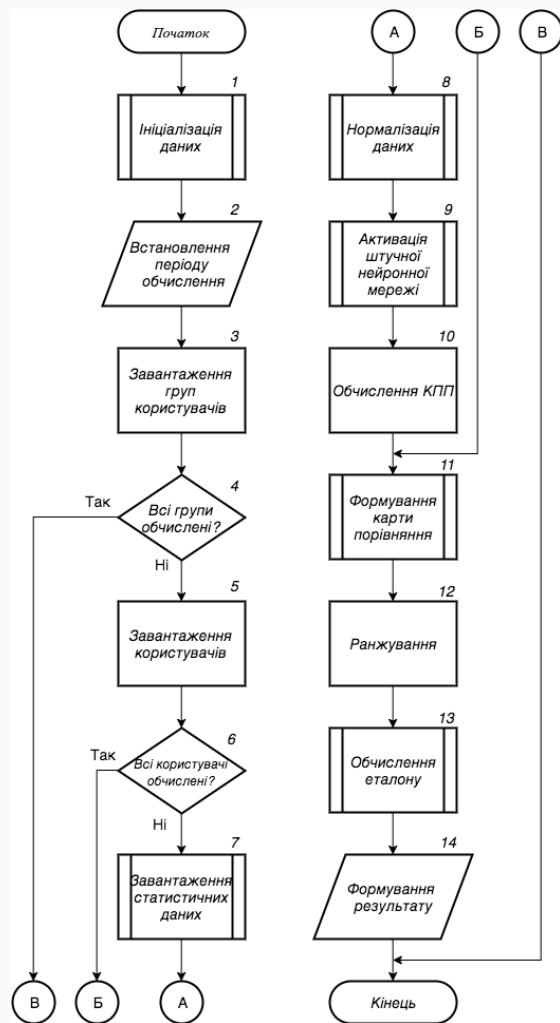
Метод оцінювання

Необхідні початкові дані:

- Групи співробітників
- Групи критеріїв
- Питома вага критеріїв
- Шкала оцінювання

Алгоритм ранжування

1. Ініціалізація даних
2. Введення часового діапазону
3. Завантаження груп співробітників
4. Завантаження співробітників
5. Завантаження статистичних даних
6. Нормалізація даних
7. Активація нейромережі
8. Обчислення КПП
9. Формування карти порівняння
10. Ранжування
11. Обчислення еталону
12. Формування звіту

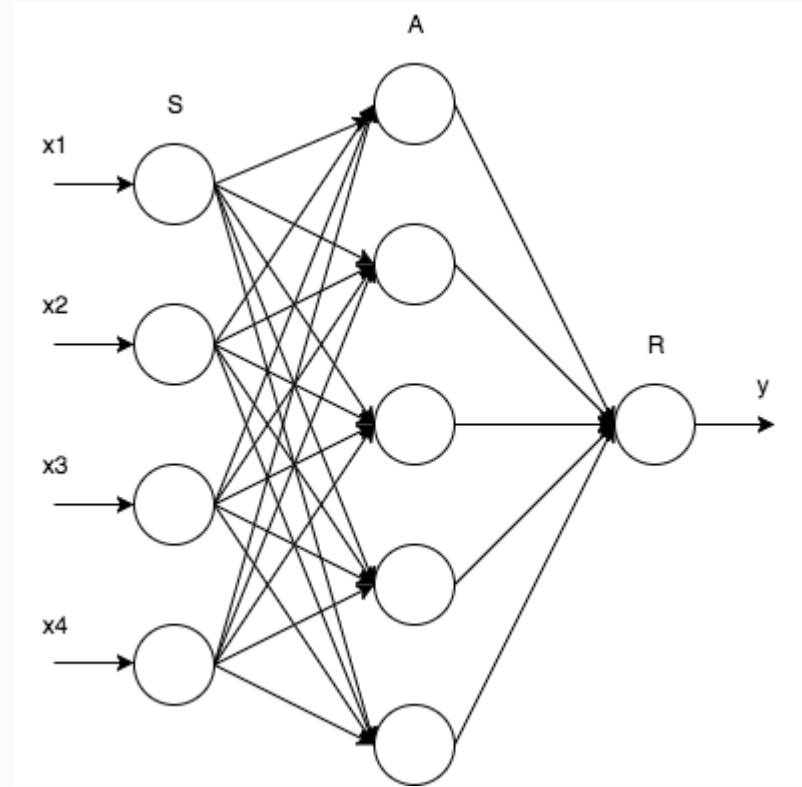


Особливості штучної нейронної мережі

Перцептрон

Складається з:

- S-елементів - сенсори
- A-елементів - асоціативні
- R-елементів - реагуючі



```
let network = new Architect.Perceptron(4, 5, 1);
```

Нормалізація

Особливості нормалізації
вхідних та вихідних даних

- вхід та вихід - числові значення $\{0, 1\}$
- числові параметри - у відсотковому вигляді
- не числові параметри - бінаризація

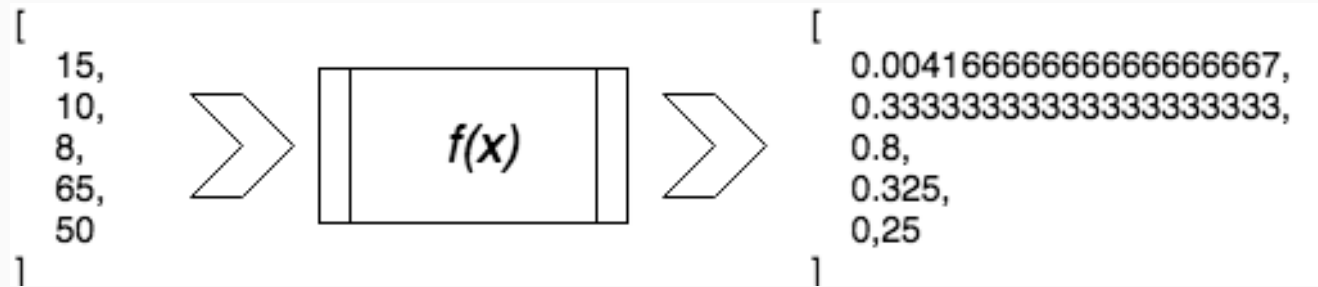
Вхідні дані

Група користувачів - “Оператор першого рівня”

Критерії оцінювання:

- відвідуваність (макс. затримка 3600 секунд)
- оперативність (макс. тривалість 30 секунд)
- якість (оцінка клієнтів за десятибальною шкалою)
- продуктивність (макс. кількість дзвінків за день - 200)
- кваліфікація (макс. кількість задоволених клієнтів за день - 200)

Нормалізація вхідних даних



Навчання

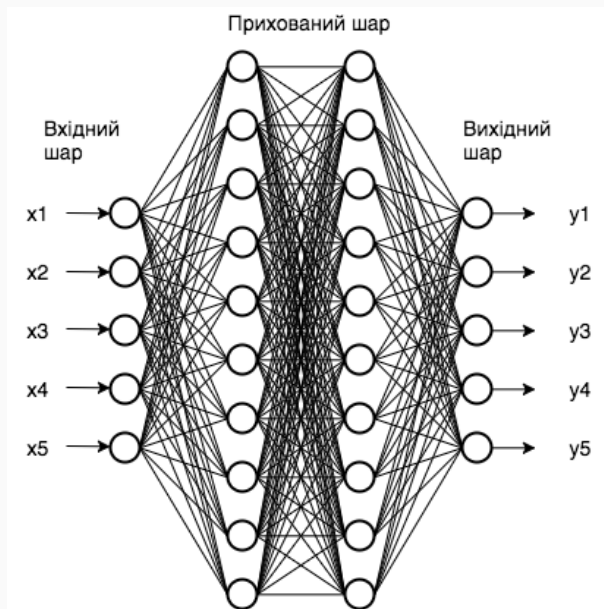
Тип: навчання з учителем

Параметри навчання:

- швидкість навчання
- макс. кількість ітерацій
- цільова помилка
- перемішування
- логування

```
let trainer = new Trainer(network);  
  
trainer.train(trainingSet,{  
    rate: .1,  
    iterations: 80000,  
    error: .005,  
    shuffle: true,  
    log: 1000  
});
```

Оптимальна топологія нейронної мережі



```
root@b14bcs1352e7:usr/src/helper# node testing.js
Start training.
iterations 1000 error 0.008724958791468008 rate 0.1
iterations 2000 error 0.008633705005226344 rate 0.1
iterations 3000 error 0.008291826820021615 rate 0.1
iterations 4000 error 0.0081000425748211 rate 0.1
iterations 5000 error 0.008281672077260847 rate 0.1
iterations 6000 error 0.008046262789103992 rate 0.1
iterations 7000 error 0.008001508308342105 rate 0.1
iterations 8000 error 0.008099110602107702 rate 0.1
iterations 9000 error 0.008094244955739282 rate 0.1
iterations 10000 error 0.007952943734027046 rate 0.1
iterations 11000 error 0.007910472189201287 rate 0.1
iterations 12000 error 0.006803015222655505 rate 0.1
iterations 13000 error 0.006123744982698667 rate 0.1
iterations 14000 error 0.005920752925121288 rate 0.1
iterations 15000 error 0.005860347915910524 rate 0.1
iterations 16000 error 0.005730916144480057 rate 0.1
iterations 17000 error 0.005716218369630016 rate 0.1
iterations 18000 error 0.005596482424453766 rate 0.1
iterations 19000 error 0.005474719649800349 rate 0.1
iterations 20000 error 0.005497666644303288 rate 0.1
iterations 21000 error 0.005410126430219308 rate 0.1
iterations 22000 error 0.0053062937911207315 rate 0.1
iterations 23000 error 0.005322299432405755 rate 0.1
iterations 24000 error 0.005318309733023312 rate 0.1
Result of training: { error: 0.004983483968424854, iterations: 24621, time: 15266 }
Testing
Input [ 0.8, 0.7, 0.9, 1, 0.4 ]
Output [ 0.6694707534006066,
0.6671959933747598,
0.6095077888945609,
0.4763301794518062,
0.457930035294432 ]
```

Практичне застосування

- Автоматичне ранжування
- Інтерактивне ранжування
- Адаптивність

Дякую за увагу