ЗАДАЧА 1: Да се направи скрипт, който чрез НМ, да **реализира класификация**.

Опишете какво преформатиране (PreProcessing) на входните данни сте направили и защо:

Таблица : Описание на формата на подаваните данни в мрежата[[1]](#footnote-1)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Име на полето (колоната) от мрежата | Тип на данните в полето | Тип на полето | Изчисляемо | Пояснение |
| Pri\_1 | Integer | Входно | Не | Пример 1 |
| Pri\_2 | Boolean | Входно | Не | Пример 2 |
| Pri\_3 | Char | Изходно | Не | Пример 3 |
| Pri\_4 | Real | Изходно | Да | Пример 4 |
| Pri\_5 | Integer | Входно | Да | Пример 5 |
|  |  |  |  |  |
| …. |  |  |  |  |

Таблица 2: Първоначална архитектура на мрежата

|  |  |  |
| --- | --- | --- |
| Слой неврони | Брой неврони в слоя преди оптимизацията | Брой неврони в слоя след оптимизацията |
| Входни | 2 |  |
| Изходни | 1 |  |
| Скрит слой 1 |  |  |
| Скрит слой 2 |  |  |
| …. |  |  |

Таблица : Параметри на мрежата

|  |  |  |
| --- | --- | --- |
| Параметри на мрежата | преди оптимизацията | След оптимизацията |
| Активационна функция |  |  |
| Learning rate |  |  |
| Momentum |  |  |

Оптимизация чрез на ГА (ако е използван)

|  |  |  |
| --- | --- | --- |
| Параметър | Тип | Пояснение |
| Pri\_1 | Входен параметър, подлежащ на промяна |  |
| Pri\_2 | Входен параметър, подлежащ на промяна |  |
| Pri\_3 | Входен параметър, подлежащ на промяна |  |
| Res | Функция за изчисляване на резултат, който се сравнява с предишен и последващ, за да се избере определен вариант | Може например да тренира НМ при по-малък брой цикли, т.е тук се описва ролята на параметъра. |
| Тип на селекцията |  |  |
| Новото поколение е резултат от | Мутация и/или кръстосване (посочва се конкретният тип кръстосване) |  |
| Степен на промяна | Ако е въведен регулатор за степента на мутация и /или кръстосване |  |

Оптимизация чрез на НМ (ако е използван) – посочете в свободен методиката на оптимизация

Таблица : Крайни резултати за задачата

|  |  |
| --- | --- |
| Параметър за оценка | Стойност |
| Общ брой записи |  |
| Брой записи в тренинг сета |  |
| Брой записи за валидация |  |
| Брой записи за реално тестване |  |
| Брой цикли на трениране на мрежата |  |
| Избран критерий за стоп на тренинга |  |
| Средна Реално измерена грешка върху три произволни примера |  |
| Време за трениране на мрежата |  |
| Време за оптимизация на архитектурата[[2]](#footnote-2) |  |
| Време за зареждане на данните |  |
| Общо време |  |

Графика на грешките по време на обучението на мрежата

ЗАДАЧА 2: Да се направи скрипт, който чрез НМ, да **реализира прогноза**.

Опишете какво преформатиране (PreProcessing) на входните данни сте направили и защо:

Таблица : Описание на формата на подаваните данни в мрежата[[3]](#footnote-3)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Име на полето (колоната) от мрежата | Тип на данните в полето | Тип на полето | Изчисляемо | Пояснение |
| Pri\_1 | Integer | Входно | Не | Пример 1 |
| Pri\_2 | Boolean | Входно | Не | Пример 2 |
| Pri\_3 | Char | Изходно | Не | Пример 3 |
| Pri\_4 | Real | Изходно | Да | Пример 4 |
| Pri\_5 | Integer | Входно | Да | Пример 5 |
|  |  |  |  |  |
| …. |  |  |  |  |

Таблица 2: Първоначална архитектура на мрежата

|  |  |  |
| --- | --- | --- |
| Слой неврони | Брой неврони в слоя преди оптимизацията | Брой неврони в слоя след оптимизацията |
| Входни | 2 |  |
| Изходни | 1 |  |
| Скрит слой 1 |  |  |
| Скрит слой 2 |  |  |
| …. |  |  |

Таблица : Параметри на мрежата

|  |  |  |
| --- | --- | --- |
| Параметри на мрежата | преди оптимизацията | След оптимизацията |
| Активационна функция |  |  |
| Learning rate |  |  |
| Momentum |  |  |

Оптимизация чрез на ГА (ако е използван)

|  |  |  |
| --- | --- | --- |
| Параметър | Тип | Пояснение |
| Pri\_1 | Входен параметър, подлежащ на промяна |  |
| Pri\_2 | Входен параметър, подлежащ на промяна |  |
| Pri\_3 | Входен параметър, подлежащ на промяна |  |
| Res | Функция за изчисляване на резултат, който се сравнява с предишен и последващ, за да се избере определен вариант | Може например да тренира НМ при по-малък брой цикли, т.е тук се описва ролята на параметъра. |
| Тип на селекцията |  |  |
| Новото поколение е резултат от | Мутация и/или кръстосване (посочва се конкретният тип кръстосване) |  |
| Степен на промяна | Ако е въведен регулатор за степента на мутация и /или кръстосване |  |

Оптимизация чрез на НМ (ако е използван) – посочете в свободен методиката на оптимизация

Таблица : Крайни резултати за задачата

|  |  |
| --- | --- |
| Параметър за оценка | Стойност |
| Общ брой записи |  |
| Брой записи в тренинг сета |  |
| Брой записи за валидация |  |
| Брой записи за реално тестване |  |
| Брой цикли на трениране на мрежата |  |
| Избран критерий за стоп на тренинга |  |
| Средна Реално измерена грешка върху три произволни примера |  |
| Време за трениране на мрежата |  |
| Време за оптимизация на архитектурата[[4]](#footnote-4) |  |
| Време за зареждане на данните |  |
| Общо време |  |

Графика на грешките по време на обучението на мрежата

ЗАДАЧА 3: Да се направи скрипт, който чрез НМ, да **реализира клъстериазация**.

Опишете какво преформатиране (PreProcessing) на входните данни сте направили и защо:

Таблица : Описание на формата на подаваните данни в мрежата[[5]](#footnote-5)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Име на полето (колоната) от мрежата | Тип на данните в полето | Тип на полето | Изчисляемо | Пояснение |
| Pri\_1 | Integer | Входно | Не | Пример 1 |
| Pri\_2 | Boolean | Входно | Не | Пример 2 |
| Pri\_3 | Char | Изходно | Не | Пример 3 |
| Pri\_4 | Real | Изходно | Да | Пример 4 |
| Pri\_5 | Integer | Входно | Да | Пример 5 |
|  |  |  |  |  |
| …. |  |  |  |  |

Таблица 2: Първоначална архитектура на мрежата

|  |  |  |
| --- | --- | --- |
| Слой неврони | Брой неврони в слоя преди оптимизацията | Брой неврони в слоя след оптимизацията |
| Входни | 2 |  |
| Изходни | 1 |  |
| Скрит слой 1 |  |  |
| Скрит слой 2 |  |  |
| …. |  |  |

Таблица : Параметри на мрежата

|  |  |  |
| --- | --- | --- |
| Параметри на мрежата | преди оптимизацията | След оптимизацията |
| Активационна функция |  |  |
| Learning rate |  |  |
| Momentum |  |  |

Оптимизация чрез на ГА (ако е използван)

|  |  |  |
| --- | --- | --- |
| Параметър | Тип | Пояснение |
| Pri\_1 | Входен параметър, подлежащ на промяна |  |
| Pri\_2 | Входен параметър, подлежащ на промяна |  |
| Pri\_3 | Входен параметър, подлежащ на промяна |  |
| Res | Функция за изчисляване на резултат, който се сравнява с предишен и последващ, за да се избере определен вариант | Може например да тренира НМ при по-малък брой цикли, т.е тук се описва ролята на параметъра. |
| Тип на селекцията |  |  |
| Новото поколение е резултат от | Мутация и/или кръстосване (посочва се конкретният тип кръстосване) |  |
| Степен на промяна | Ако е въведен регулатор за степента на мутация и /или кръстосване |  |

Оптимизация чрез на НМ (ако е използван) – посочете в свободен методиката на оптимизация

Таблица : Крайни резултати за задачата

|  |  |
| --- | --- |
| Параметър за оценка | Стойност |
| Общ брой записи |  |
| Брой записи в тренинг сета |  |
| Брой записи за валидация |  |
| Брой записи за реално тестване |  |
| Брой цикли на трениране на мрежата |  |
| Избран критерий за стоп на тренинга |  |
| Средна Реално измерена грешка върху три произволни примера |  |
| Време за трениране на мрежата |  |
| Време за оптимизация на архитектурата[[6]](#footnote-6) |  |
| Време за зареждане на данните |  |
| Общо време |  |

Графика на грешките по време на обучението на мрежата

1. Попълнените данни са примерни [↑](#footnote-ref-1)
2. Може да бъде 0, ако не е правена оптимизация на мрежата. [↑](#footnote-ref-2)
3. Попълнените данни са примерни [↑](#footnote-ref-3)
4. Може да бъде 0, ако не е правена оптимизация на мрежата. [↑](#footnote-ref-4)
5. Попълнените данни са примерни [↑](#footnote-ref-5)
6. Може да бъде 0, ако не е правена оптимизация на мрежата. [↑](#footnote-ref-6)