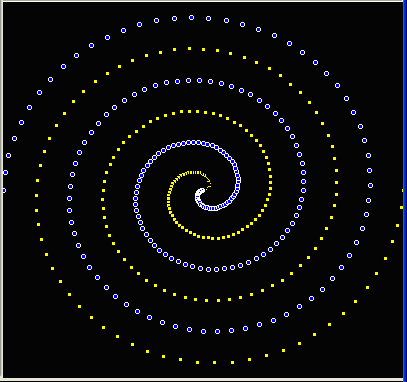
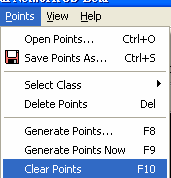
# Класифікація фігури подвійної спіралі

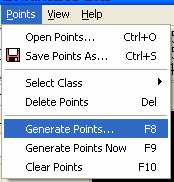


## Підготовка вихідних даних.

Для підготовки вихідних даних в меню Points вибирається пункт Clear Points, щоб очистити вікно.



У Points вибираємо Generate Points



з'являється діалогове вікно, в якому робляться налаштування (рис. 1)

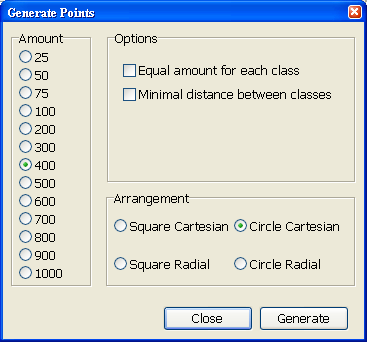


Рис. 1

Після натиску на кнопку «Generate», генеруються нові групи точок (рис.2-рис.4):

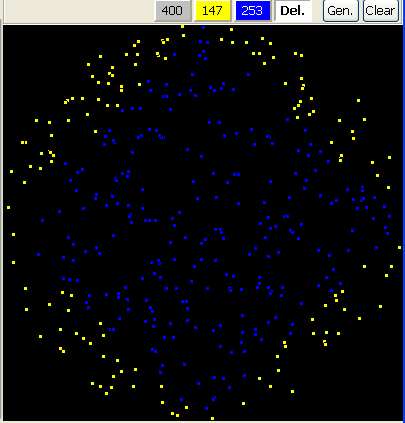


Рис.2

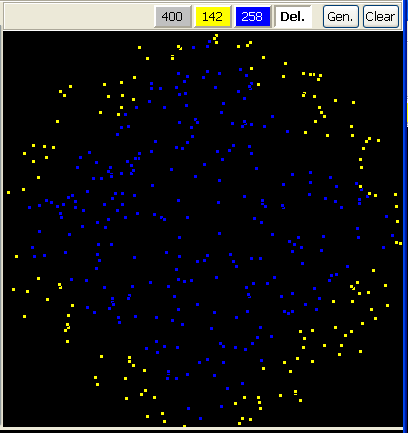


Рис. 3.

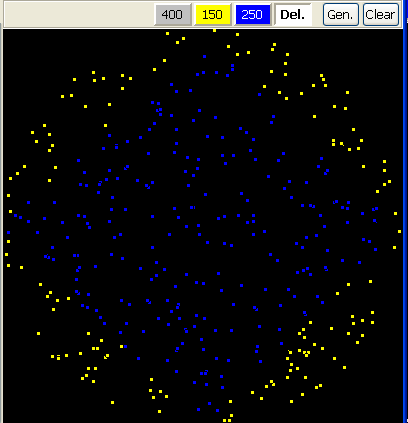


Рис. 4

Якщо змінити налаштування на Square Radial, то отримується нова конфігурація точок (Рис.5)

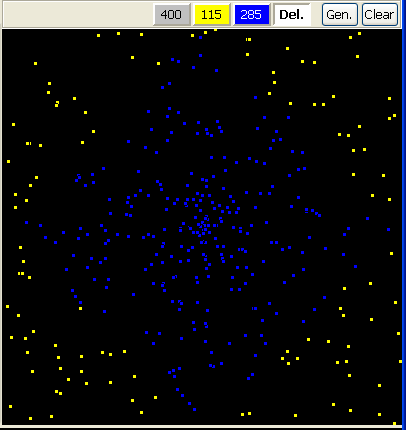


Рис. 5

Або після зміни кількості точок та розташування на Square Cartesian (рис. 6)

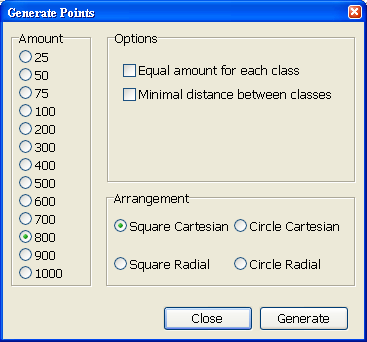


Рис. 6

Отримується наступний вигляд (рис.7)

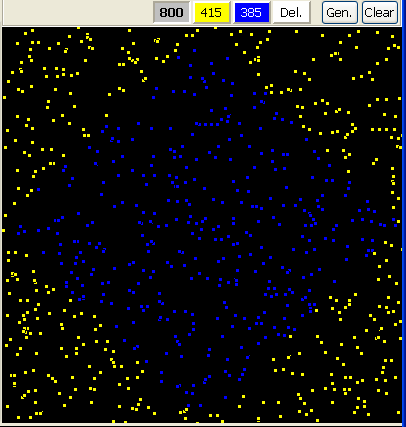


Рис. 7

Після додавання умови «Мінімальна дистанція між класами» (рис.8):

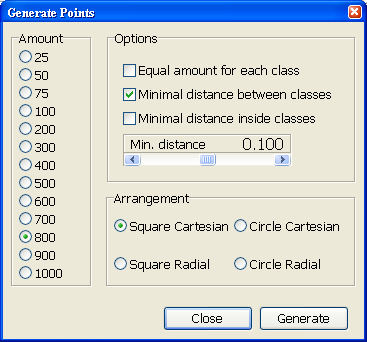


Рис. 8



Рис. 9

Встановлюється умова «Мінімальна дистанція всередині класів» (Рис. 10)

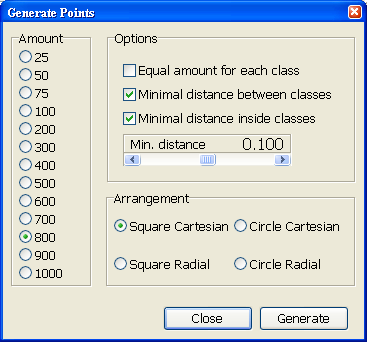


Рис. 10

Остаточний вигляд налаштованої множини точок (рис. 11).

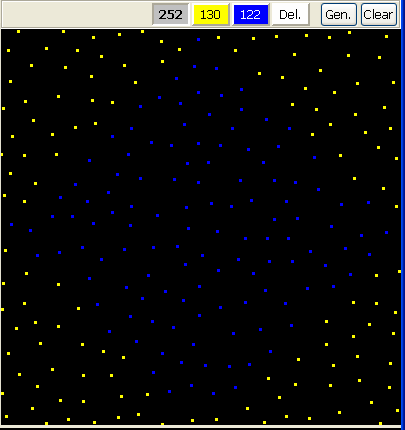
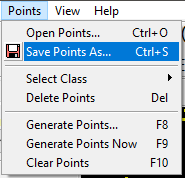


Рис. 11

Утворений набір точок зберігається командою Points – Save Points As



## Використання програми Sharky для контрастування нейронної мережі.

Контрастування нейронної мережі служить для спрощення її архітектури, зменшення числа вхідних сигналів і зниження вимог до точності і кількості вхідних сигналів.

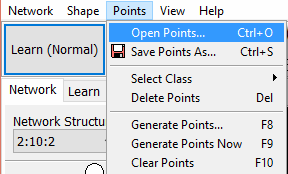
### Послідовність роботи:

1. Створити, навчити і перевірити нейромережу в Sharky, що вирішує завдання класифікації точок "Two Spirals Cartesian". Привести і розшифрувати графіки, що характеризують навчання.
2. Змінюючи різні параметри мережі, записувати характеристики, що отримані у всіх дослідах, і на їх основі виявити мінімальне число шарів і нейронів у них, коли завдання ще вирішується.
3. За рахунок зміни кількості епох визначити оптимальне їх значення для навчання розглянутого типу мережі.
4. Описати, як впливають регульовані параметри на ефективність навчання.
5. Виявити умови, при яких відбувається перенавчання мережі.

## Виконання.

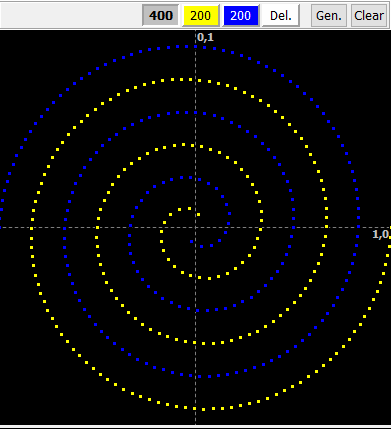
### 1. Створення і навчання нейромережі

Для початку завантажимо в Sharky власний набір точок. Для цього у верхньому меню вибирається пункт: Points - Open Points – Потрібний файл.



Вибір множини «Two Spirals Cartesian»

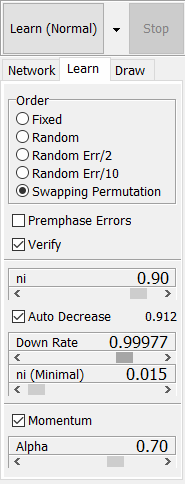
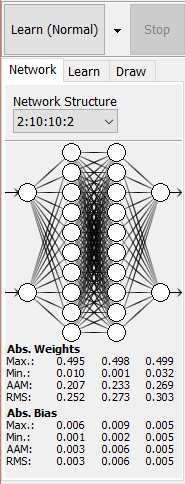
Після вибору в полі відображається наступна картина:



Тепер приступимо до навчання. Вибираємо чотиришарову мережу з 10 нейронами на кожному з двох прихованих шарів і з 2 нейронами на вході і виході. (2:10:10:2)

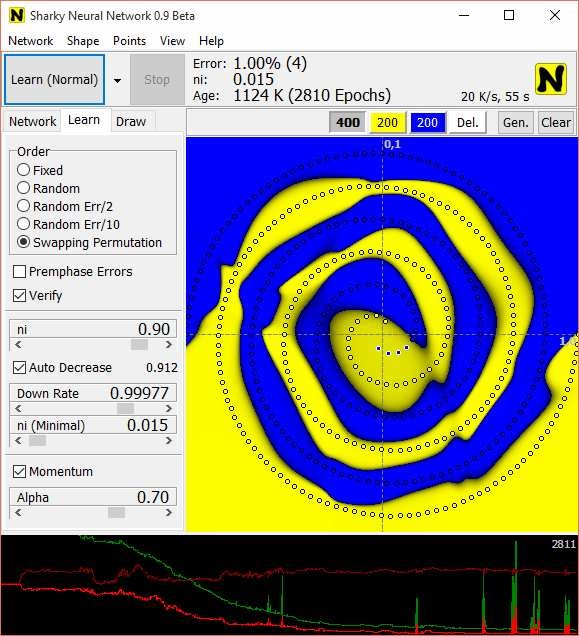
Режим навчання залишаємо Learn (Normal).

Далі, всі параметри залишаємо за замовчуванням.



Задано режим навчання зі всіма параметрами і архітектурою мережі.

#### Навчання мережі



Результат навчання мережі

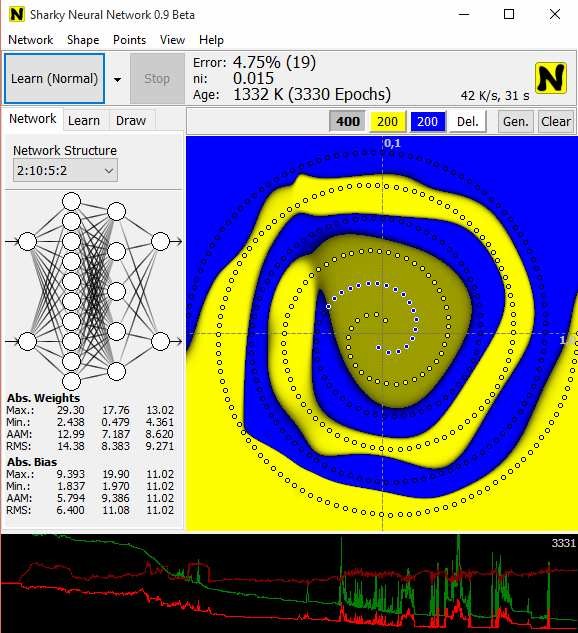
Дане завдання було вирішено за 2810 епох, при цьому відсоток помилки склав 1%. Числове значення швидкості навчання (ni) склало 0,015. Під спіраллю відображається трьохколірний графік.

* Зеленою лінією позначено значення середньоквадратичної помилки. На даному прикладі це значення зменшується, але має 7 великих стрибків. У ці моменти в мережі спостерігалося перенавчання.
* Яскраво-червоною лінією позначено просту помилку, вона теж монотонно убуває, маючи ті ж 7 стрибків у моменти перенавчання.
* Бордовою лінією позначено помилку перевірки, яка не має значних стрибків і змін.

Після навчання мережі, можна здійснити її контрастування.

### 2. Контрастування навченої мережі.

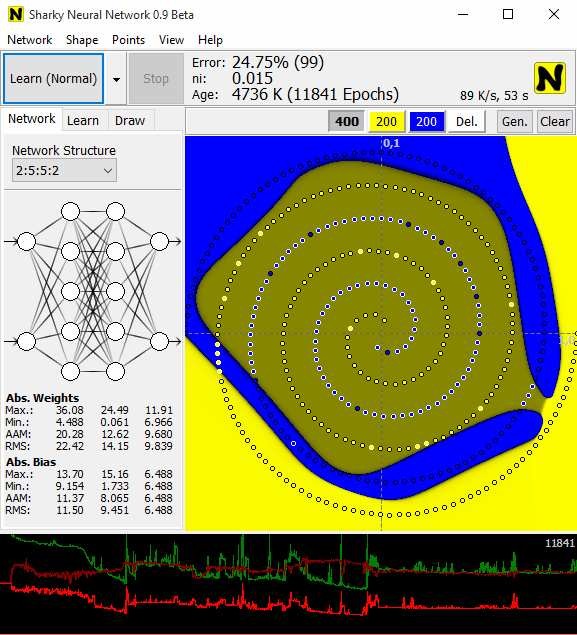
1. Змінимо архітектуру мережі та розглянемо одержувані характеристики навчання. Для початку можна змінити архітектуру мережі, а саме, третій прихований шар з 10 нейронами замінити на 5 нейронів.



Навчання мережі з новою архітектурою

Дане завдання виконане за 3330 епох (тобто часу знадобилося більше), причому відсоток помилки становить 4,75%, що в 4 рази більше за попередній результат. За графіками внизу видно, що до кінця навчання виникає явне перенавчання, тобто якби не було натиснута кнопка Stop, помилка б почала збільшуватися ще більше. Числове значення швидкості навчання не змінилося (0,015).

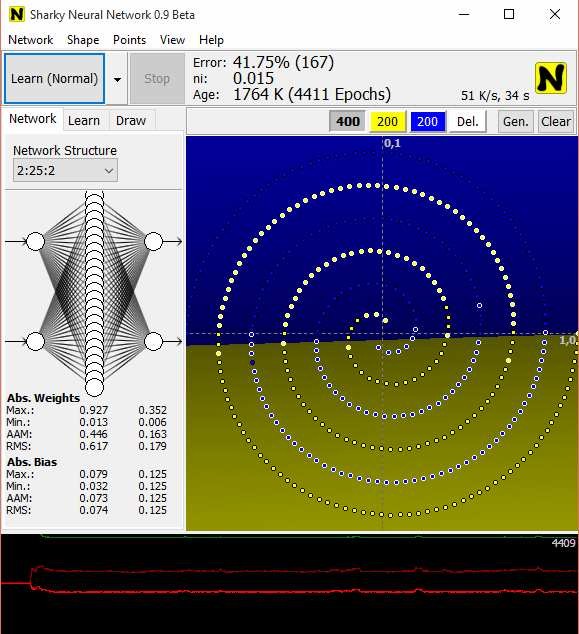
Тепер спробуємо зменшити кількість нейронів і в другому, і в третьому шарах, а саме, виберемо структуру 2:5:5:2. Інші параметри залишимо незмінними.



Навчання нової мережі

Спостерігаємо, що вже така мережа не справляється з поставленим завданням: на графіках явно видно перенавчання, яке веде до величезної помилку (ми зафіксували 25% на 11841 епохах, але вона може збільшитися ще). Таким чином, зменшувати більше число нейронів не можна.

Спробуємо змінити кількість шарів нейронів. Зменшимо наше число шарів до 3, а саме, виберемо структуру 2:25:2.



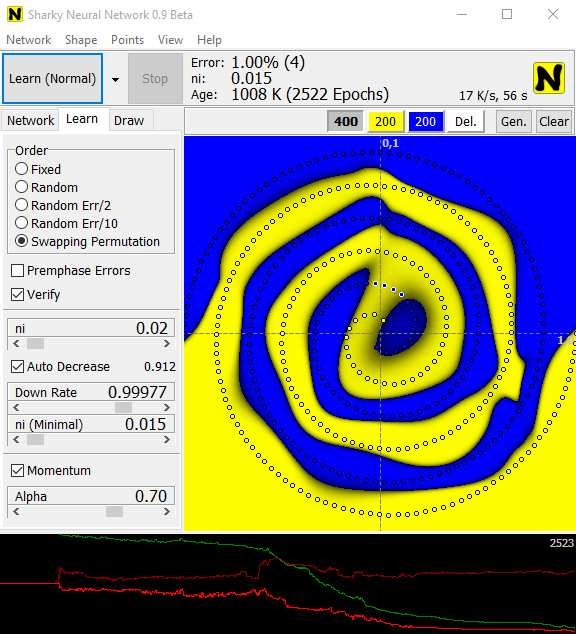
Навчання мережі з 3 шарами

Трьохшарова мережа з цим завданням також не справляється. Помилка коливається від 40 до 45% і не зменшується. При заміні 25 нейронів прихованого шару на 50 і 100 результат не змінюється, процес лише сповільнюється.

Який висновок можна зробити? Мінімальна мережа, що вирішує дану задачу складається з 4 шарів, на прихованих шарах якої не менше 15 нейронів. (Тобто архітектури 2:10:5:2, 2:10:10:2 справляються).

2. Яку мінімальну кількість епох потрібно для виконання завдання. Оскільки відкинуто всі мережі нижче чотиришарових, то потрібно визначити мінімальне число епох, за яке впораються мережі 2:10:10:2 і 2:10:5:2.

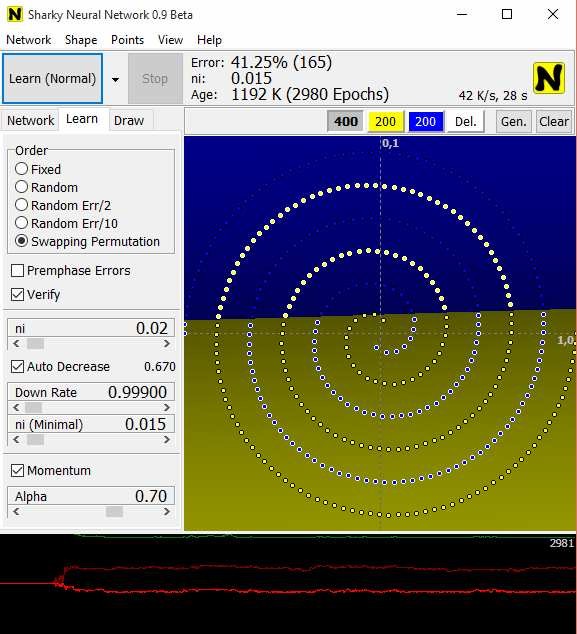
Можна змінити значення ni, а саме від 0,90 зробити 0,02. Розглянемо архітектуру 2:10:10:2.



Робота мережі 2:10:10 2 зі зміненим ni

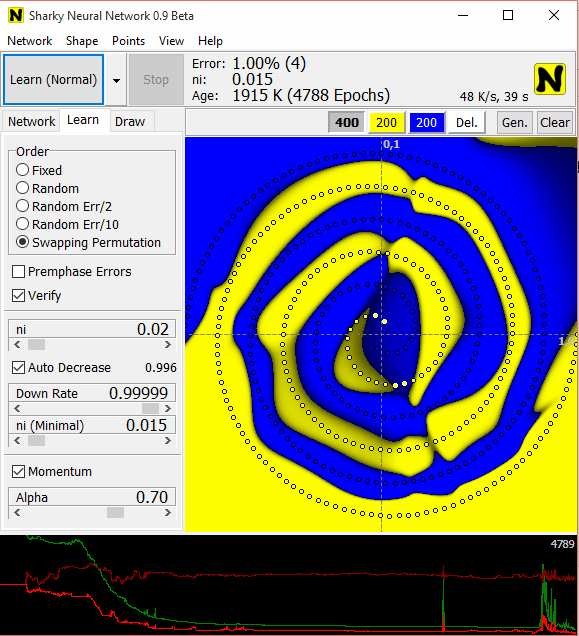
Кількість епох, за які завдання було виконано, зменшилася з 2810 до 2522.

Якщо змінити Down Rate з 0,99977 до 0,999, то, по закінченні 2810 епох, результат не досягається.



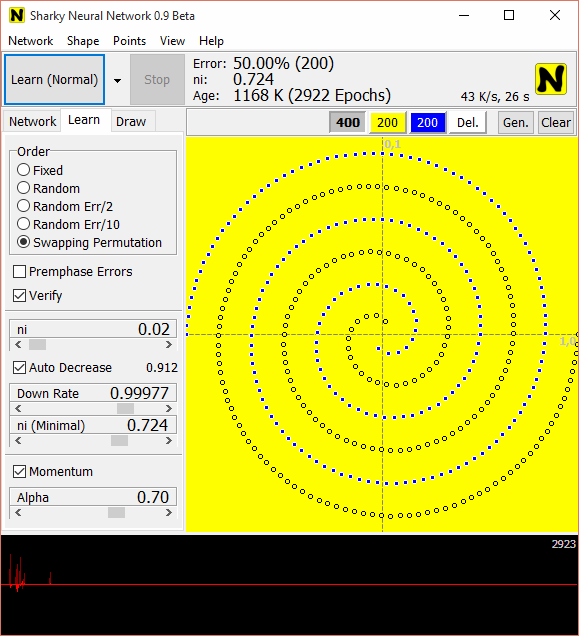
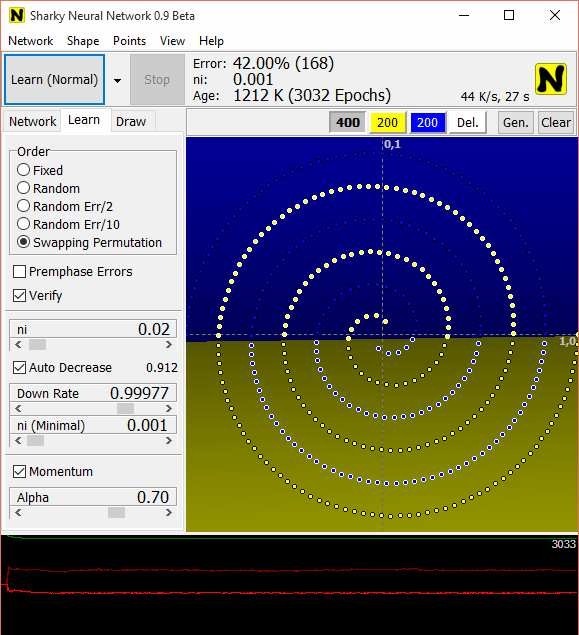
Робота мережі зі зміною Down Rate

І навіть при збільшенні цієї характеристики до 0,99999, завдання виконується за більше число епох (4788).



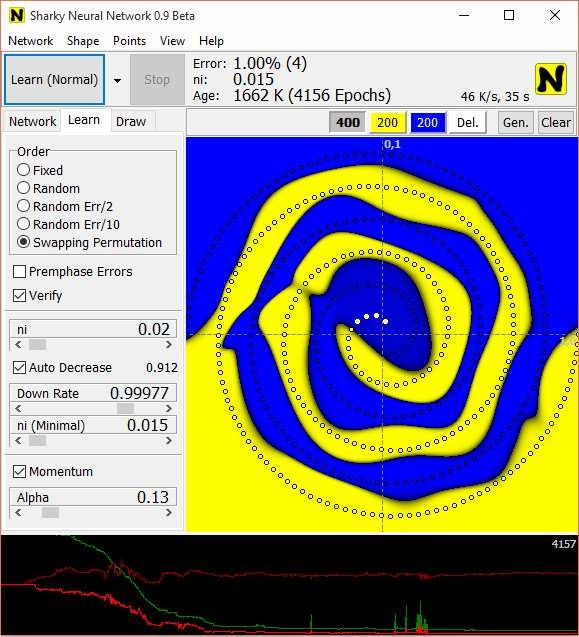
Робота мережі зі збільшеним Down Rate

При коригуванні ni (Minimal) у більший та менший бік найкращим результатом все одно залишається вихідний.



Робота мережі зі зміненим ni (Minimal)

Якщо зменшити Alpha з 0,70 до 0,13, то результат буде отримано, але теж за більше число епох (4156).



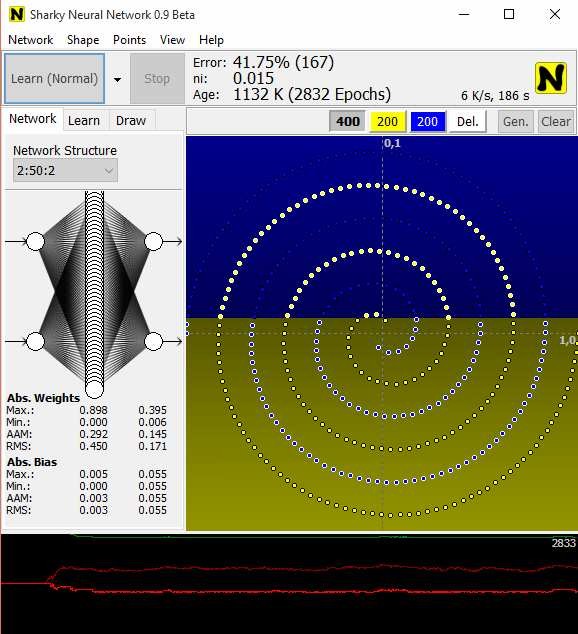
Робота мережі зі зменшеним Alpha

Змінивши Alpha в більшу сторону, вирішити дану задачу не буде можливим. Аналогічно, помінявши Order мережі, результат досягнутий не буде.

Таким чином, мінімальна кількість епох, за які мережа 2:10:10:2 вирішує поставлене завдання, так само є 2522. Це відбувається при зміні параметра ni, а саме, зменшення його з 0,90 до 0,02. При цьому зміна інших параметрів призводить лише до збільшення кількості необхідних епох або зовсім не вирішує задачу.

Аналогічна ситуація відбувається зі структурою мережі 2:10:5:2. Тільки при зменшенні ni кількість епох скорочується з 3330 епох до 3125 епох.

Можна подумати, що зменшене ni покращує роботу мережі, тому, можливо і тришарова мережа впорається з цим завданням? Але, провівши експеримент, бачимо, що це не так. Значить, дійсно, оптимально з цим завданням справляється чотиришарова мережу виду 2:10:10:2 зі зменшеним ni до 0,02 за 2 522 епох.



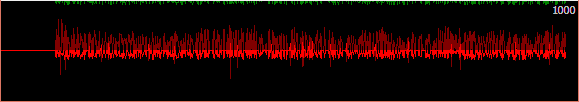
Робота мережі виду 2: 50: 2 зі зменшеним ni

### 3. Висновки щодо роботи мережі при зміні різних параметрів.

#### Перенавчання.

1. Змінюючи архітектуру мережі, із заданим завданням справляються лише дві: 2:10:10:2 і 2:10:5:2. Тришарові і нижче мережі не вирішують поставленого питання.
2. Щоб оптимізувати час виконання завдання, необхідно зменшити ni до 0,02, це призводить до раннього виконання завдання в обох мережах.
3. Встановлення даного параметра не допомагає мереж з меншим числом шарів впоратися із завданням.
4. При зменшенні Down Rate завдання не виконується, а при його збільшенні з вихідного 0,99977 процес вирішення завдання подовжується. (З обома мережами)
5. При будь-якій зміні ni (Minimal) спостерігається перенавчання мережі. Це не демонструється на наведених рисунках, оскільки графіки внизу занадто швидко змінюються, але при спостереженні за процесом роботи це видно.

При зменшенні Alpha (характеристика Momentum) завдання досягається, але за більше число епох, а при його збільшенні також видно перенавчання. Це видно з графіка, що представляє собою безперервні стрибки.



Аналогічну картину можна спостерігати при зміні характеристики Order з Swapping Permutation на будь-яку іншу. Тобто знов перенавчання.