## Лабораторная работа №1

**Знакомство со средой NetMaker, создание нейронных сетей.**

Цель работы: познакомится со средой, её интерфейсом, принципом работы искусственных нейронных сетей и сконструировать собственную сеть.

**Теоретическая часть**

NetMaker – бесплатно распространяемый визуальный дизайнер ИНС. Способен моделировать многослойный персептрон, а так же каскадно-корреляционные сети с динамической подстройкой архитектуры. Включает различные учебные данные, ошибки и активационные функции.

Интерфейс программы изображён на рисунке 24.



Рисунок 24 – Интерфейс NetMaker

Первая строка состоит из выпадающих меню.

* 1. Файл
  2. Новый проект
  3. Открыть
  4. Сохранить
  5. Сохранить как
  6. Закрыть выделенное окно
  7. Выход
  8. Редактировать
  9. Добавить компонент
  10. Добавить граф
  11. Выбор окна
  12. Помощь

Несколько наиболее часто используемых действий меню могут быть доступны через панель инструментов, находящейся во второй строке.

* 1. Создать новый проект
  2. Открыть существующий проект
  3. Сохранить выделенный проект
  4. Создать блок данных
  5. Создать блок сети
  6. Создать блок классификации
  7. Создать блок трансформации
  8. Создать контейнер графа
  9. Создать рычаг
  10. Создать гистограмму
  11. Создать график координат
  12. Создать график сигнала
  13. Создать график ошибки сети

Разбор используемых блоков можно посмотреть в пользовательской документации.

**Практическая часть**

Указание к работе:

Для создания нового проекта выбираем File – New, появится окно, которое в дальнейшем будет включать все элементы, которые нам потребуются. К ним относятся:

1. Сеть. Edit – Add Component – NetWork.

Для настройки её обучения нужно нажать на кнопку Go. Здесь можно выбрать режим работы сети. Выберите алгоритм обучения BPQuickPropOptimize. Затем справа среди параметров изменим:

MaxIter =2000 (Количество итераций) InitStep = 1 (μ0)

Во вкладке Structure and Training Controls нужна галочка Save iteration info для передачи данных по итерации к графику.

Нажав на кнопку Setup можно переименовать блок и проект.

1. Два набора данных, один для обучения, второй для проверки работоспособности сети. Edit – Add Component – Data Set.

Через этот блок подключаются выборки данных, с помощью Setup – Add – путь к файлу. Функцию ошибки выберем MSE.

Нажав на кнопку VO на блоке можно соединить выборку с сетью.

1. Рычаг для начала работы сети. . Edit – Add Component – Trigger.

Для его подключения к сети, в Network нажимаем на кнопку Go. В Training setup во вкладке Structure and Training Controls выбираем Trigger Source – Internal. В качестве Sourse используем новый Trigger, перенеся его в колонку Connected. Между блоками образуется связь.

1. График точек.

Edit – Add Graph – Scatter Plot и Edit – Add Component – Graph.

Компонент граф будет роль визуализации процессов в сети.

На график добавляем нужный объект – обучающую выборку, вызвав правой кнопкой мыши всплывающее меню и выбрав в нём Add XY Data. Важно при добавлении данных к графику выбрать цвет Rainbow, иначе будет не видно процессов на графике. Получим отображение точек как на рисунке 7.1.

Нажав правой кнопкой на график и выбрав Live Update, получим визуальное отображение работы сети.

Для подключения графика к проекту нужно к компоненту присоединить сам график. Выбрав нужный график в настройках блока Graph.

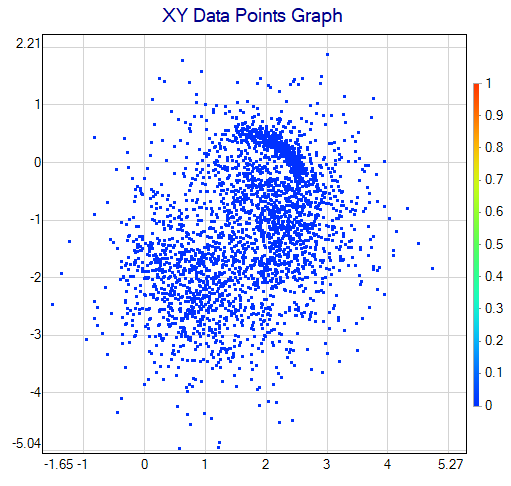


Рисунок 7.1. График обучающей выборки

1. График ошибки.

Edit – Add Graph – Network Error.

Подключаем к графику сеть, а к старому блоку визуализации сам график. Так же выбираем Live Update. Получим готовый к работе график.

1. График чистоты – эффективности.

Добавляем ещё один блок графика. В его настройках, во вкладке Setup выбираем Trigger Mode On signal, а в качестве источника будет сама сеть.

Edit – Add Graph – Signal Selection.

В качестве отслеживаемых данных выбираем выборку данных обучения.

В результате получим систему, изображённую на рисунке 7.2.

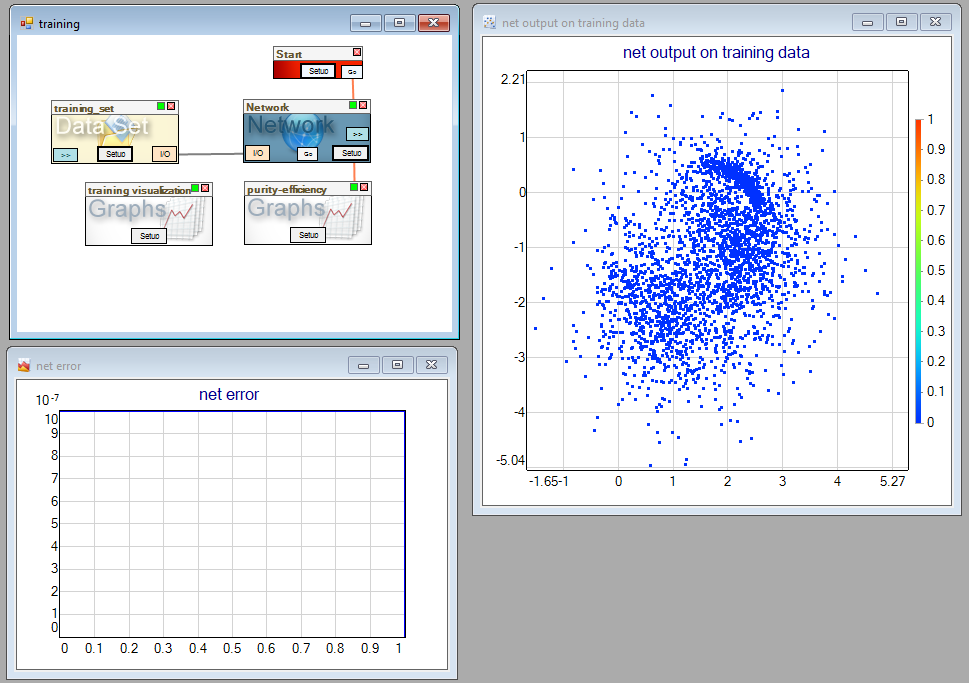


Рисунок 7.2 Система обучения сети

После добавления всех необходимых компонентов, можно начать обучение сети. Нажав на кнопку Go, начнётся обучение сети.

В прогрессе обучения можно увидеть количество пройденных итераций, скорость, количество рождённых и удалённых.

После обучения получим следующий результат, на рисунке 7.3.

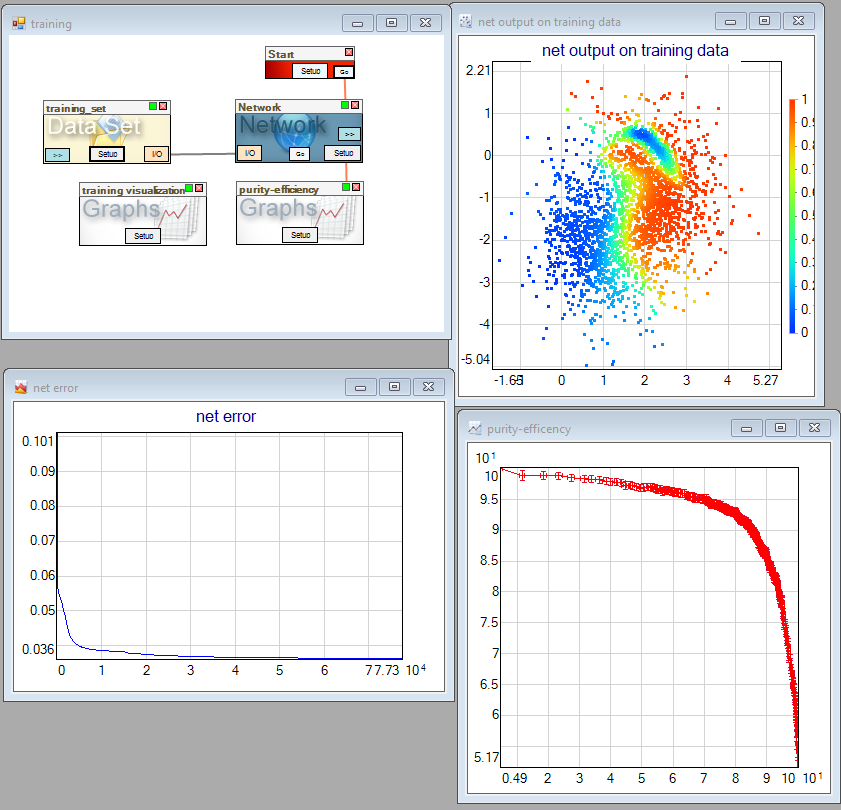


Рисунок 7.3 – Результат обучения

Полученную сеть можно сразу тестировать на работоспособность. Для этого меняем режим сети с обучения на JustRun, а вместо обучающей выборки ставим выборку другого формата и большего размера, для тестирования сети. Источник графика точек нужно поставить на выборку тестирования, цвет поставить на Topol.

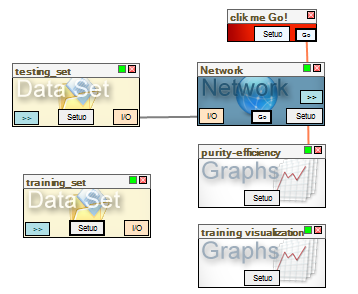


Рисунок 7.4 – Система тестирования сети

После проведения теста получим примерно следующий результат.

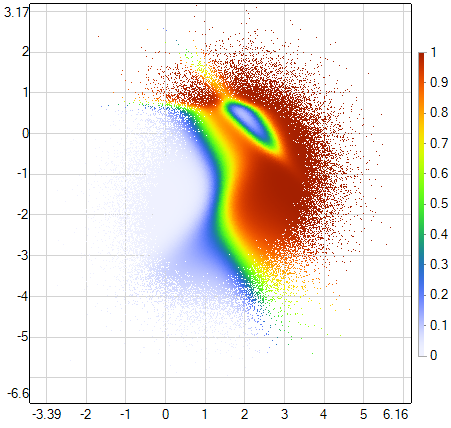


Рисунок 7.5 – Результат тестирования сети

Задание:

1. Спроектировать простую сеть с помощью редактора.
2. Обучить сеть, затем провести тесты на выборке большего размера.
3. Проанализировать полученные результаты и работоспособность сети.

Содержание отчёта:

1. Краткая теория.
2. Скриншоты результатов обучения и тестирования сети с графиками.
3. Выводы.

## Лабораторная работа №2

**ИНС и алгоритмы классификации**

Цель работы: сравнение работы ИНС и классифицирующих алгоритмов.

**Теоретическая часть:**

В качестве альтернативы нейронным сетям NetMaker включает в себя классифицирующие алгоритмы kNN, SVM, PDE. Они могут быть использованы для сравнения, если у вас нет представления о том, какой должен быть результат работы нейронной сети после её обучения.

Все функции блока классификации расписаны в пользовательской документации.

**Практическая часть:**

Для выполнения задания нужно добавить в проект блок классификации, два блок данных для хранения тестовых и обучающих данных. Для хранения результатов тестов понадобится три блока данных, для каждого классифицирующего алгоритма.

Для сравнения классифицирующих алгоритмов с искусственной нейронной сетью понадобится блок сети и всё необходимое для её обучения.

Для получения результатов работы блока классификации, нужно подключить обучающие и тесовые данные к блоку классификации. Саим выборки данных могут быть одинаковыми, но для работы блока они должны быть подключены.

На выход блока классификации подключаем блок для хранения результатов. Для этого в его настройках поставьте галочку в Read Data на Internal Source.

Для отображения результатов желательны блоки графиков, к которым подключены сами графики.

По итогу должна получиться система как на рисунке 7.6.

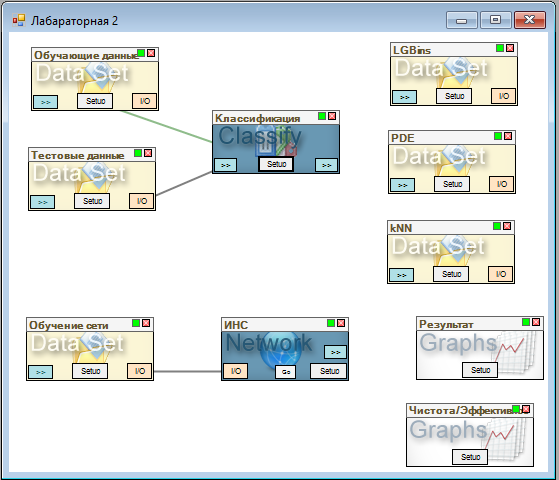


Рисунок 7.6 – Система для тестирования алгоритмов класификации

Задание:

1. Подготовить выборки задания и пропустить их через блок классификации тремя алгоритмами LGBins, PDE, kNN. Коэффициенты выбрать для используемой выборки. Результаты нужно занести в соответствующие блоки данных
2. Обучим сеть на той же выборке данных
3. Сравним результат и скорость работы различных методов.

Содержание отчёта:

1. Краткая теория.
2. Скриншоты результатов работы алгоритмов и сети с графиками.
3. Выводы.

## Лабораторная работа №3

**Статическая и динамическая архитектура**

Цель работы: освоить различные способы обучения нейронных сетей.

**Теоретическая часть:**

При создании сети в NetMaker есть возможность использовать динамическую структуру. Она позволяет наращивать нейроны в скрытых слоях, обучая новых кандидатов в процессе обучения и непосредственно добавления их, если алгоритм сочтёт его добавление необходимым. Ниже приведены параметры динамической структуры в настройках сети.

**Add new neurons:**  включает рост сетевой структуры.

**pool:**  число случайных кандидатов – каждый раз когда размер сети увеличивается пул новых рандомизированных нейронов обучен, а лучший выбирается для расширения сети (но только если кажется что он может быть полезен всей сети.  
**neuron split:** спрятанные нейроны с наивысшей вариацией ошибка выхода разделяются на два немного другие нейроны. Затем пара новых нейронов проходит предварительное обучение; эти нейроны встраиваются в сеть, если они выдают лучший результат, чем случайные кандидаты; эта настройка всегда включена.

**max neurons, max weights:** ограничивает число нейронов и коэффициенты сетевого соединения. Обучение останавливается, когда любой из этих лимитов достигнут. Значение 0 выключает ограничение.

**Remove neurons:** включает обрезку неиспользуемых или избыточных нейронов; **twins** - мера разницы между активациями двух нейронов, которые будут заменены одним эквивалентным нейроном; **dead** – мера значимости нейронов которые могут быть безопасно удалены; **const** – мера активации нейронов σ/μ , которая будет использоваться для учета нейрона как константы ; увеличивая эти значения будет удалятся больше нейронов.

**OBS (Optimal Brain Surgeon):** техника удаления единственного соединения из сети; Если OBS включён, алгоритм применяется в конце процесса обучения; если iterative тоже включён, алгоритм применяется после каждой попытки вырезания нейронов. Этот алгоритм использует большое количество времени и памяти.

**Практическая часть:**

Необходимая система для тестирования изображена на рисунке 7.7.

В ней используется блоки данных для обучения и две сети. Одна настроена на статическое обучение, другая на динамическое обучение.

Для включения динамического обучения в Go – Structure and Training Controls нужно поставить галочки напротив Add new neurons и Remove neurons. Таким образом, сеть будет обучать и добавлять новые нейроны, если она считает это нужным

Если для статической сети нужно примерно рассчитать необходимое количество нейронов в слоях, в динамической их количество можно оставить малым.

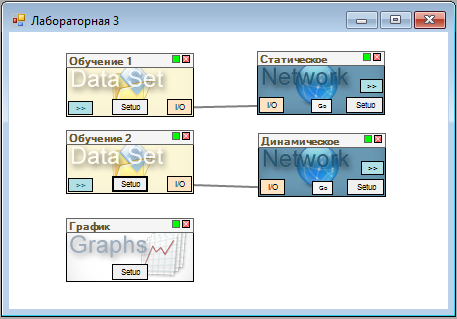


Рисунок 7.7 – Система для тестирования алгоритмов обучения

Задание:

1. Создать, обучить и протестировать на одинаковой выборке статическую и динамическую сеть.
2. Проанализировать полученные результаты обучения и тестирования, сравнить методы.

Содержание отчёта:

1. Краткая теория.
2. Время обучения, структура сети в начале и конце обучения, количество итераций работы.
3. Скриншоты результатов обучения и тестирования сети с графиками.
4. Выводы.

## Лабораторная работа №4

Сеть каскадной корреляции

Цель работы: создать сеть каскадной корреляции. Проанализировать её работу.

**Теоретическая часть:**

Сеть каскадной корреляции – это архитектура ИНС, в которой развитие структуры происходит параллельно с ее обучением. Это происходит с помощью добавления нового скрытого нейрона на каждой итерации обучения, то есть под конец обучения сеть будет иметь оптимальную архитектуру.

Архитектура сети каскадной корреляции является объединением нейронов взвешенными связями в виде развивающегося каскада. Каждый новый добавляемый нейрон присоединяется к входным узлам и ко всем уже существующим скрытым нейронам. Выходы всех скрытых нейронов и входные узлы сети напрямую подключаются также и к выходным нейронам.

**Практическая часть:**

Для создания сети каскадной корреляции нужно в Setup – Architecture нужно Network type поменять на Cascade-Correlation (original).

В системе понадобится два блока данных для обучения и тестирования сети, сама сеть и все необходимые графики. Система изображена на рисунке 7.8.

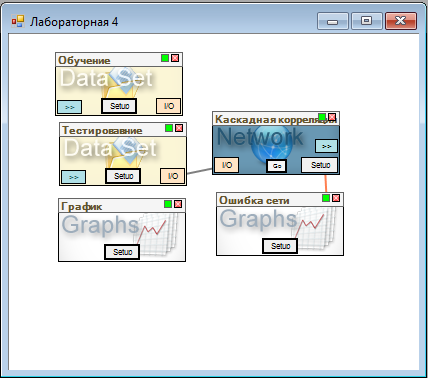


Рисунок 7.8 – Система для сети каскадной корреляции

Задание:

1. Создать сеть каскадной корреляции.
2. Обучить на одной выборке.
3. Проанализировать полученные результаты и работоспособность сети.

Содержание отчёта:

1. Краткая теория.

2. Время обучения, структура сети в начале и конце обучения, количество итераций работы.

3. Скриншоты результатов обучения и тестирования сети с графиками.

4. Выводы.