

Московский государственный университет им. М. В. Ломоносова

Физический факультет

Кафедра математики

**Оптимизация архитектуры линейного персептрона с помощью
генетического алгоритма.**

Курсовая работа по физике
студента 206 группы
Клиентова Григория Алексеевича

Научный руководитель:
д. ф.-м. н., профессор П. В. Голубцов

Москва

2022

Оглавление

1. Введение.....	3
2. Цель работы.....	3
3. Рассмотренные виды задач для линейного персептрона.....	3
4. Описание работы генетического алгоритма.....	4
4.1 Описание среды.....	4
4.2 Описание особей.....	4
4.3 Описание способа мутации.....	5
5. Описание реализованного алгоритма.....	5
6. Полученные результаты.....	6
6.1 Задача аппроксимации.....	6
6.2 Задача классификации «Исключающее ИЛИ».....	11
7. Основные итоги работы.....	13
8. Литература.....	14

1. Введение

Искусственные нейронные сети (ИНС) всё чаще начинают использоваться для решения различных прикладных задач широкой разновидности. Они позволяют находить субоптимальные решения за адекватное время и с использованием разумных ресурсов в условиях неполной или недостаточной априорной информации для построения точного решения.

Однако выбор в архитектуре модели, использующейся для нахождения такого решения может приводить к довольно различающимся результатам. В подавляющем числе случаев вид ИНС подбирают эмпирическим методом — путём перебора некоторых предварительно выбранных моделей нейронных сетей.

В данной работе рассмотрен один из методов для выбора наилучшей архитектуры линейной нейронной сети — персептрона — с помощью применения генетического алгоритма.

2. Цель работы

- Реализовать модель генетического алгоритма для оптимизации архитектуры линейного персептрона
- Изучение работы модели ГА

3. Рассмотренные виды задач для линейного персептрона

Для проверки работоспособности нейронных сетей и оценивания в первом приближении были использованы следующие «де факто» эталонные

задачи:

- Аппроксимация некоторых функций одного переменного
- Классификация точек на плоскости по схеме «исключающее ИЛИ».

4. Описание работы генетического алгоритма

В рамках генетического алгоритма существуют особи — каждой из которой сопоставляется генотип, в котором зашифрована информация об архитектуре НС. На основе генотипа строится модель НС. Далее эта модель обучается на тренировочных данных, идентичных для каждой особи. После того, как особи были обучены на определённом количестве эпох происходит валидация каждой особи на специальном валидационном наборе данных. Каждой особи присваивается значение ошибки, которое получилось на этом наборе. Далее происходит эволюционный шаг — среда, в которой содержится популяция, генерируют новый набор особей на основе предыдущей популяции путём случайной мутации и с учетом стратегии элитизма — особи, показавшие наилучший результат, переходят в новую популяцию без мутаций генов.

4.1 Описание среды

Была выбрана модель среды, которая содержит в тренировочный и валидационный наборы, а также популяцию особей, число которых на каждом эволюционном шаге неизменно. Изначально генотип каждой особи генерируется случайным образом.

4.2 Описание особей

Особь содержит в себе полносвязную нейронную сеть. Информация о её структуре содержится в генотипе в виде последовательного списка слоев. Слой может быть одним из списка:

- Линейный слой, содержащий нейроны
- Слой активации с функцией ReLU
- Слой активации с функцией Tanh
- Слой активации с функцией Sigmoid

4.3 Описание способа мутации

Мутация происходит на основе генотипа одной особи. 3 особи, показавшие наименьшую ошибку на валидационном наборе дублируются в новую популяцию без изменений, однако у них заново пересоздается модель нейронной сети, чтобы обновить веса для обучения на новой эволюционной эпохе. В новую популяцию также входят особи, мутировавшие из этих трёх. Такая стратегия называется стратегией элитизма. Она призвана улучшать сходимость генетического алгоритма.

Три особи, показавшие самые наихудшие ошибки на валидационном наборе, не переходят в новую популяцию, как и их потомки.

У особей, которые занимают промежуточные положения в ранжированном списке ошибок на валидационном наборе, переходят в новую популяцию только потомки, прошедшие мутацию. Каждая мутация может происходить с заранее определенной вероятностью.

Возможны следующие виды мутации:

1. Изменение количества параметров слоя.
2. Добавления промежуточного скрытого слоя.
3. Изменения типа слоя.
4. Удаление слоя

5. Описание реализованного алгоритма

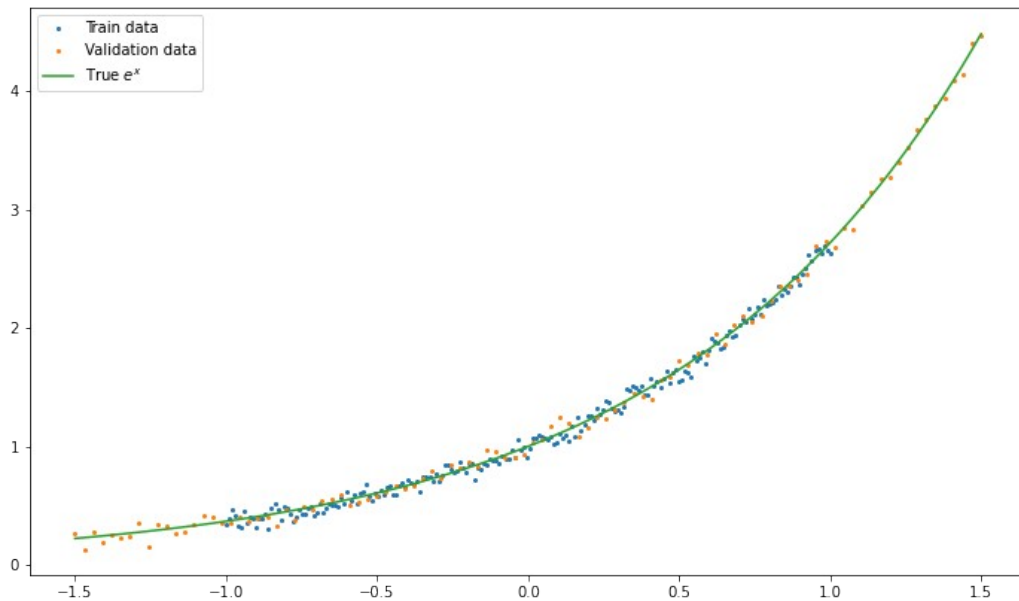
Изначально создается популяция, генотипы особей которых генерируются случайным образом. Далее эта популяция обучается на тренировочном наборе. Все наборы данных одинаковы для любых особей любых популяций. После обучения на заданном количестве обучающих эпох происходит валидация каждой особи в популяции и определение их ошибок на этом наборе. На основе этих данных происходит эволюционный шаг — в данном случае мутация особей по ранее описанному алгоритму.

6. Полученные результаты

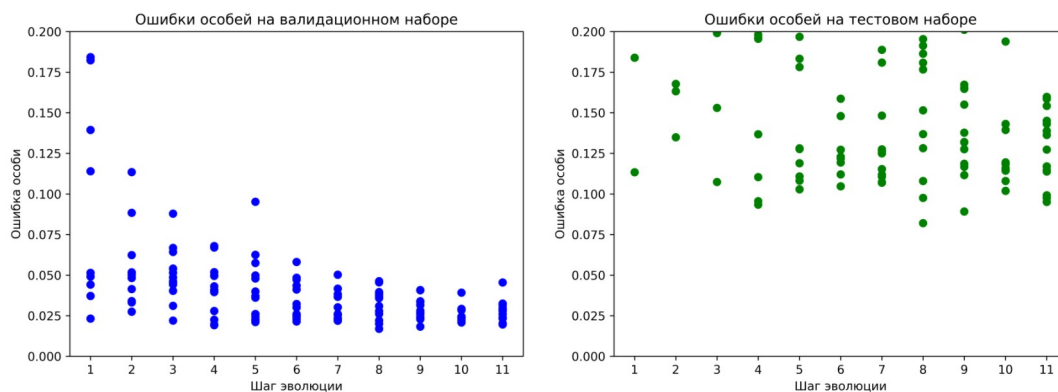
6.1 Задача аппроксимации

Поставим задачу аппроксимации функции.

Возьмем в качестве исходной функции график $y=e^x$ с шумом.



Создадим 15 особей со случайным набором генов. Установим количество обучающих эпох равным 25 на каждую эволюционную эпоху. Спустя 11 эволюционных эпох посмотрим на результаты обучения особей.

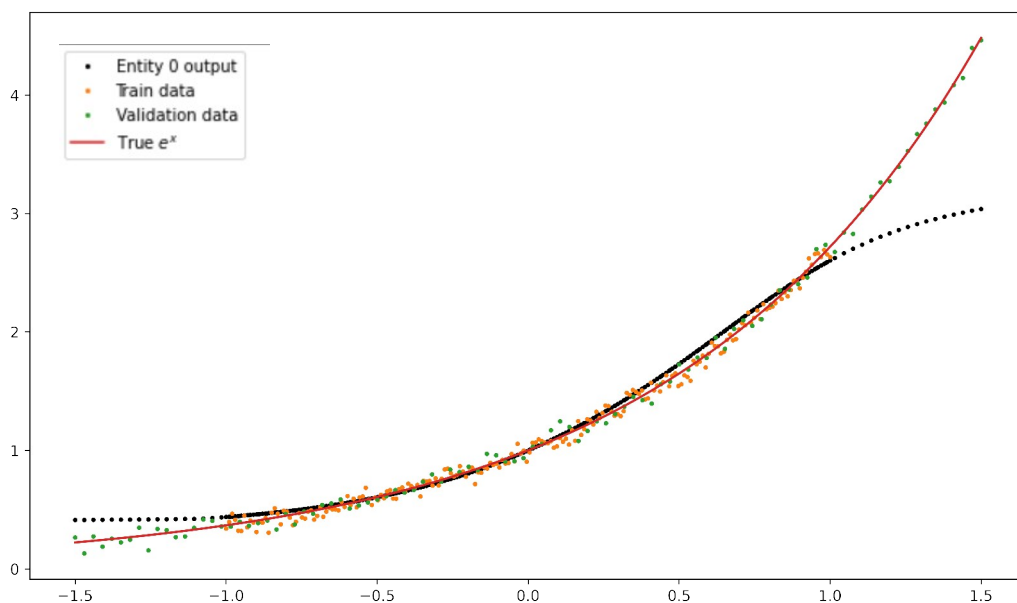


Как можно видеть из графика ошибки на валидационном наборе, размах ошибок среди всех особей на одной и той же эволюционной эпохе уменьшается, однако минимальная ошибка среди всех особей не сильно

отличается от минимума ошибки среди всех эпох и особей. Согласно стратегии элитизма минимальная ошибка среди особей должна неувеличиваться с течением эволюционных эпох, так как особи, показавшие наилучший результат, переходят в новую популяцию без изменения архитектуры. Однако из-за того, что при переходе в новую популяцию все особи заново инициализируют, причем случайным образом, обучение идентичных особей различается, как и получавшаяся ошибка.

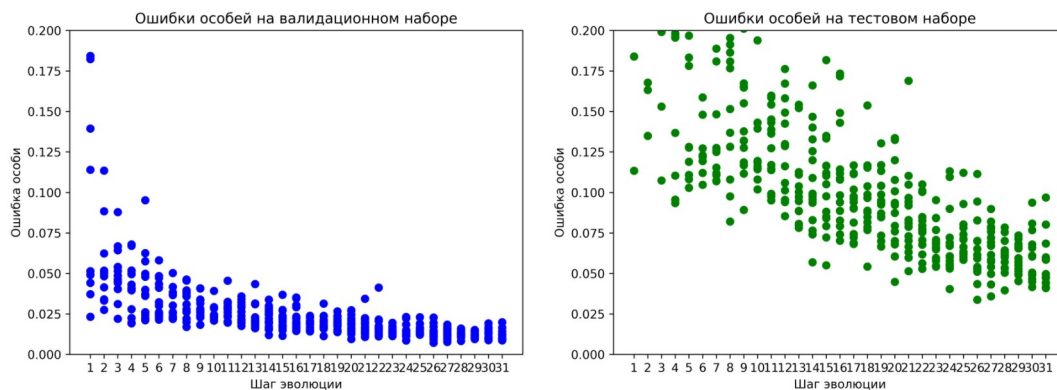
Из графика ошибки на тестовом наборе видно, что её размах внутри одной популяции также уменьшается и с количеством эволюционных эпох всё больше особей ближе к минимальной ошибке.

Посмотрим на результат одной из особей в последней популяции.



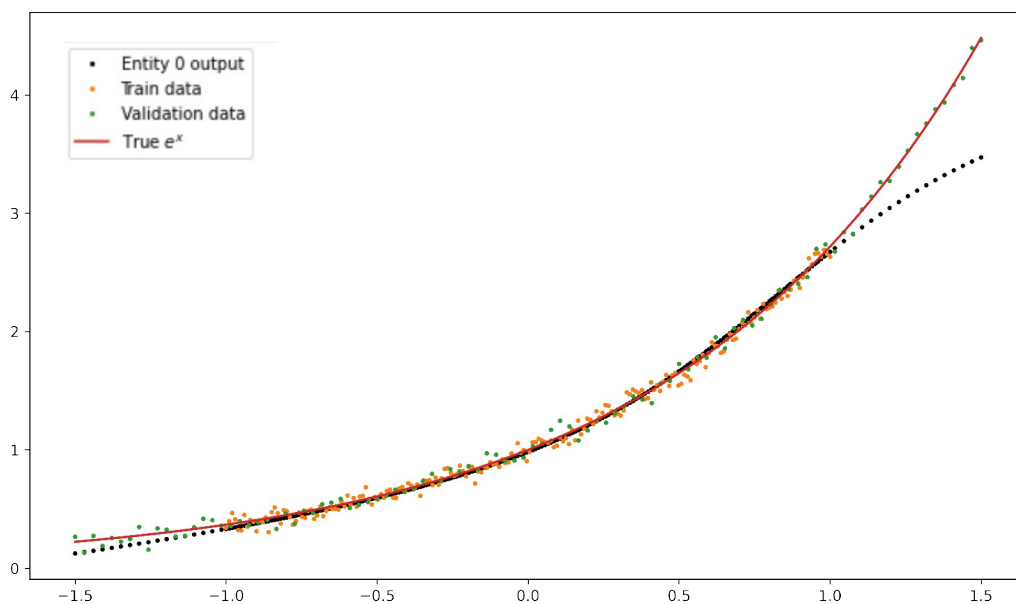
Как видно, сеть особи уже хорошо аппроксимирует исходную функцию, однако интерполяция получается неудовлетворительная.

Попробуем обучить особей на ещё 20 эволюционных эпохах.



Как видно из графиков, ошибки на тестовом и валидационном наборах падают, что говорит о работоспособности генетического алгоритма на данной задаче.

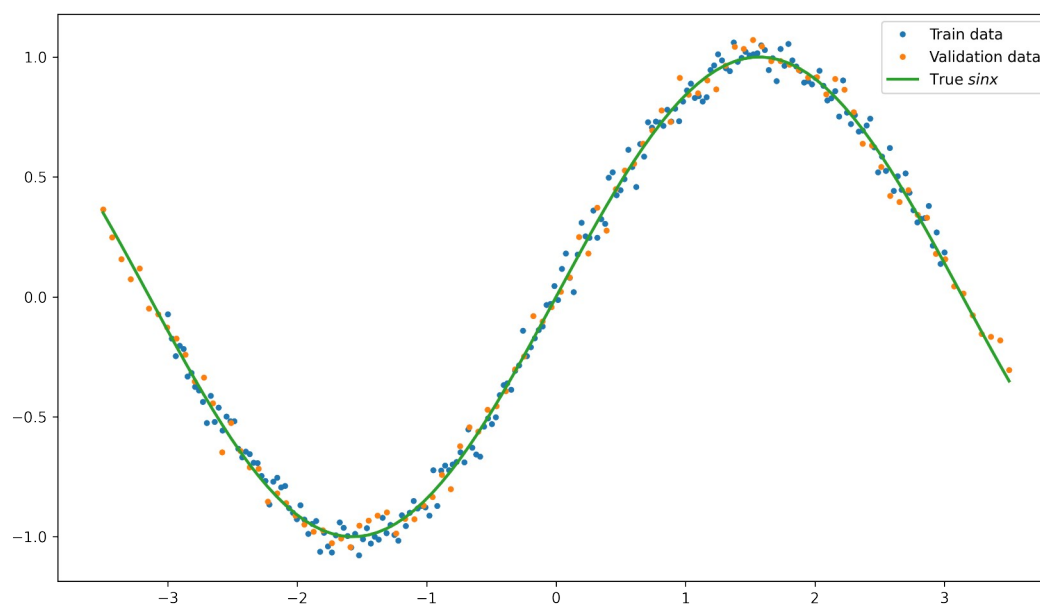
Посмотрим на предсказание одной из особей на последней популяции.



Можно увидеть, что особь научилась очень хорошо аппроксимировать исходную функцию, даже с учетом шума, однако интерполяция все также неудовлетворительная.

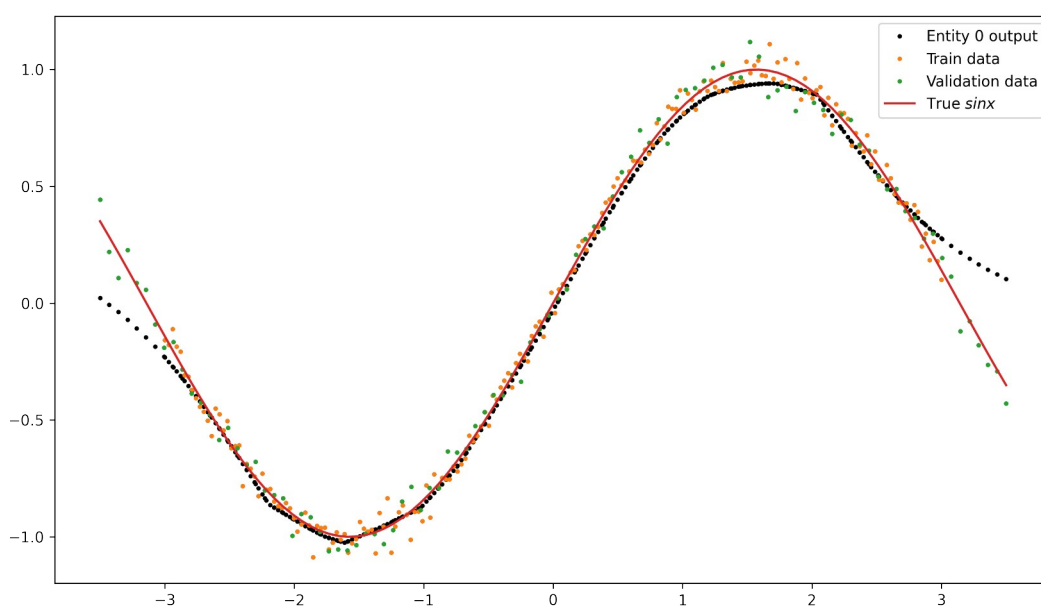
Попробуем выбрать в качестве исходной функции график вида $y = \sin x$.

Настройки генетического алгоритма идентичны.

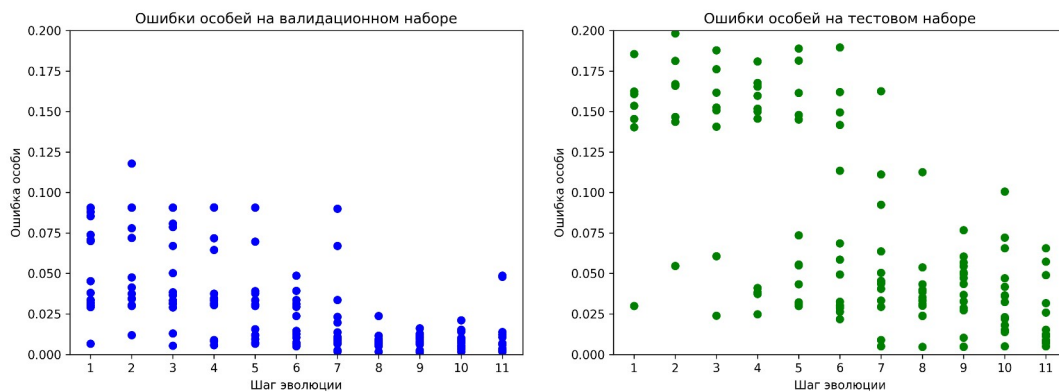


После обучения на 11 эволюционных эпохах были получены следующие результаты.

Предсказание одной из особей из последней популяции:
Как можно увидеть, даже после 10 эволюционных эпох

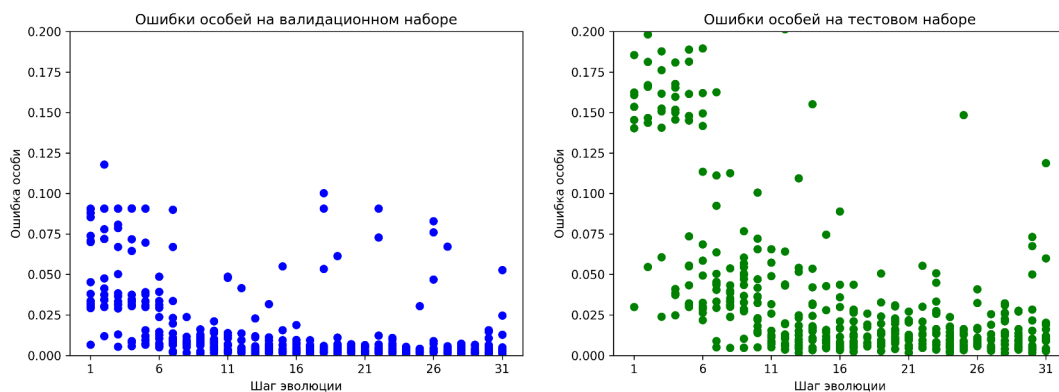


аппроксимация исходной функции достаточно хорошая.



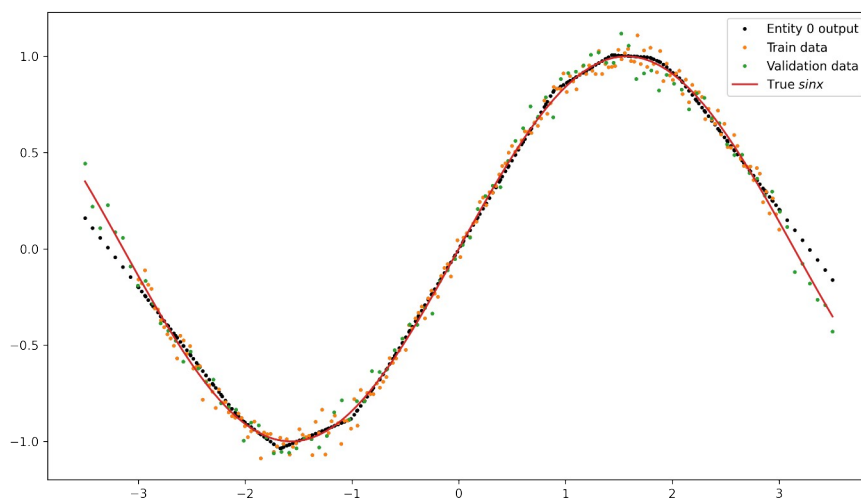
При близком рассмотрении графиков ошибок можно заметить, что размах ошибок особей внутри одной популяции уменьшается с шагом эволюции, как и при выборе экспоненты в качестве исходной функции.

Посмотрим на результаты спустя ещё 20 эволюционных эпох.



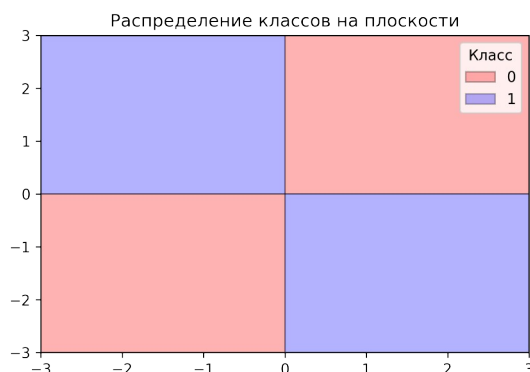
Как можно увидеть, ошибки сети уже после 10 эволюционных эпох сильно близки к 0.

Предсказание одной из особей из последней популяции:



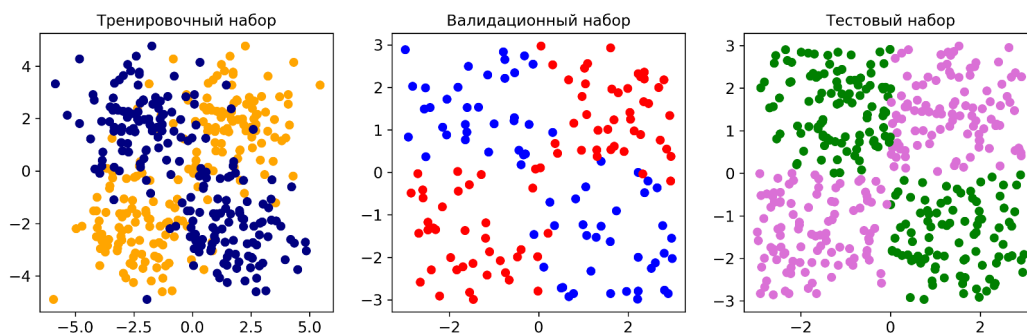
6.2 Задача классификации «Исключающее ИЛИ»

Поставим задачу классификации точек на плоскости. Распределение классов показано следующим образом:



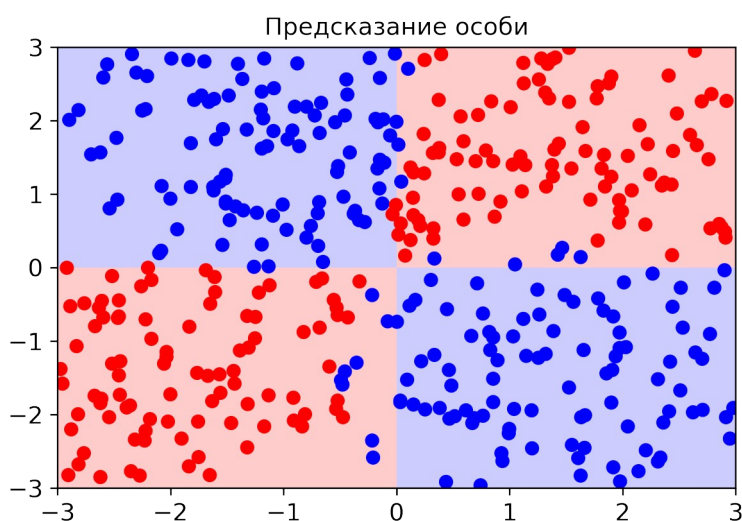
Настройки среды остаются без изменений относительно задачи аппроксимации. Исходя из этих областей генерируем данные для тренировочного, валидационного и тестового наборов.

Сами наборы выглядят следующим образом:



Тренировочный набор сгенерирован с помощью нормального распределения с достаточно большим шумом по сравнению с валидационным набором. Тестовый набор сгенерирован без шума.

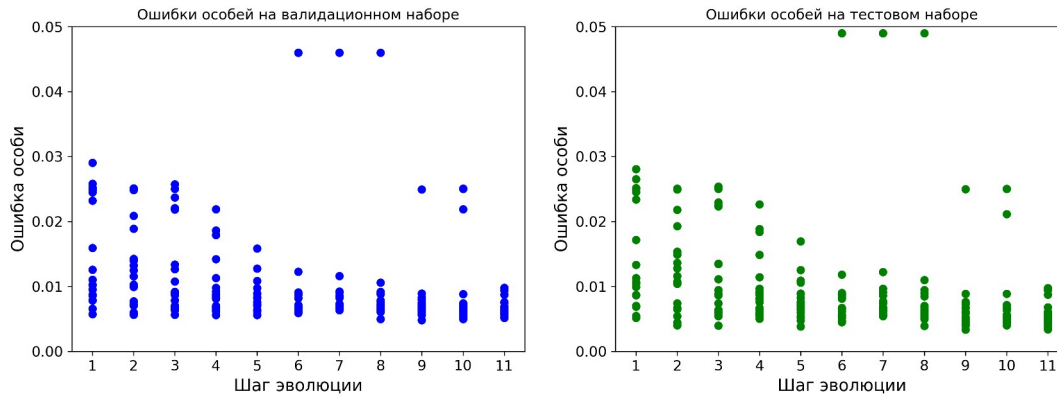
После 11 эволюционных эпох получены следующие результаты на тестовом наборе:



Цвет каждой точки обозначает принадлежность к классу.

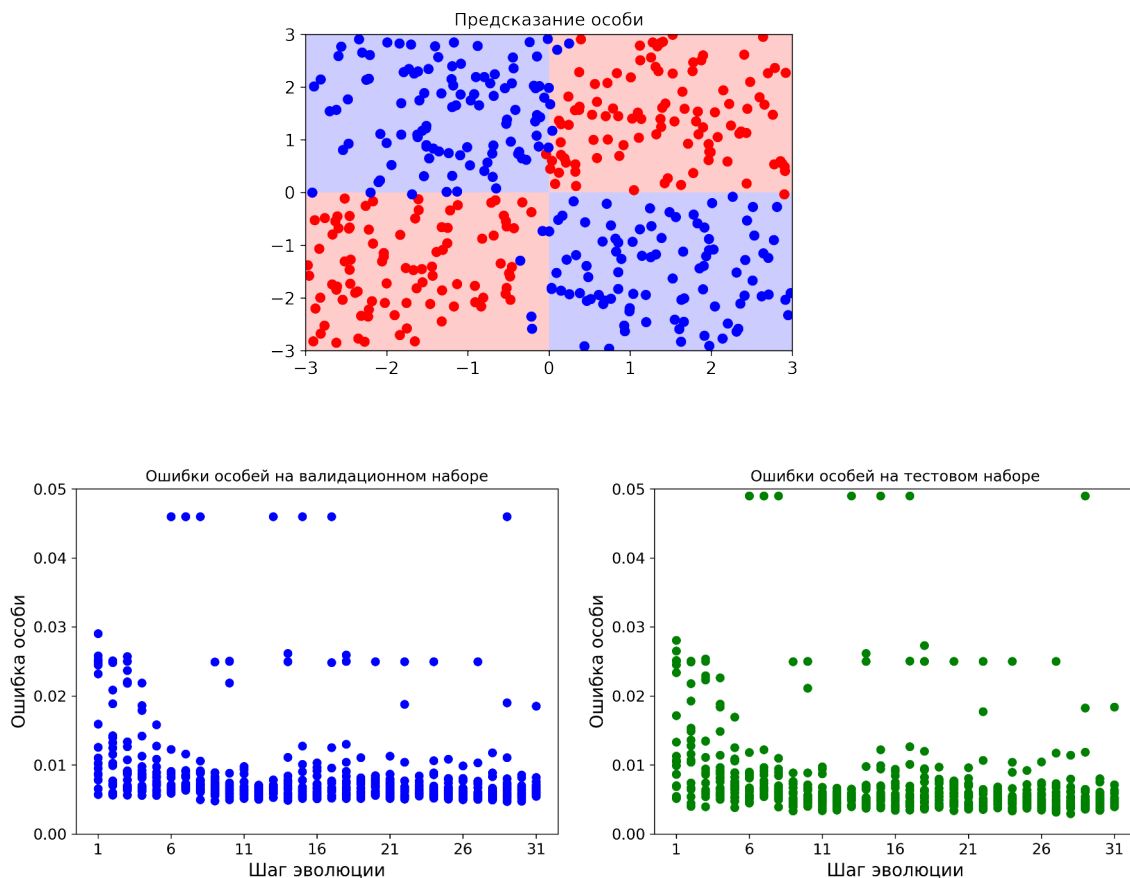
Как можно видеть из выхода особи из последней популяции, результаты получаются достаточно адекватными.

История ошибок особей выглядит следующим образом:



Здесь видно, что особь, показавшая наилучший результат за все 11 эпох, появилась ещё в начале эволюции. Это говорит либо о том, что нам повезло увидеть нейронную сеть, близкую к оптимальной, уже в начале эволюции, либо о том, что методы мутации недостаточно быстро изменяют архитектуру нейронной сети.

Результаты спустя 31 эволюционную эпоху:



Как видно из графика ошибок особей, видимых различий в результатах не

наблюдается, что говорит о довольно устойчивой архитектуре получившихся особей.

7. Основные итоги работы

В результате проделанной работы была реализована модель генетического алгоритма, оптимизирующая архитектуру линейного персептрона. Задачей алгоритма было нахождение такой архитектуры НС, которая выдавала наиболее удовлетворяющее решение для поставленной задачи. Алгоритм был изучен на задачах аппроксимации функций экспоненты и синуса, а также на классификации точек на плоскости.

В условиях задачи аппроксимации функций алгоритм выделил наиболее оптимальные архитектуры НС, максимально удовлетворяющих условиям задачи за конечное время. Из графиков ошибок популяции на всех наборах можно сделать вывод, что эволюционный подход действительно находит наилучшие решения для решения задачи аппроксимации, а стратегия элитизма гарантирует сходимость данного алгоритма.

В условиях задачи классификации точек на плоскости по схеме «исключающее ИЛИ» наиболее оптимальное модель нейронной сети была получена сразу, на этапе генерации первой популяции особей. Это связано со спецификой задачи — для её решения удовлетворяет широкий класс архитектур нейронных сетей.

В заключение стоит отметить, что реализация данного алгоритма универсальна относительно задач, которые поставлены для нейронных сетей. Благодаря эволюционному подходу в переборе различных архитектур НС можно найти наиболее подходящую модель для решения поставленной задачи за конечное время. Однако стоит принять во внимание, что время генетический алгоритм основан на переборе получающихся вариантов, что в совокупности со случайной генерацией популяций даёт право утверждать, что работа алгоритма может занять продолжительное время, прежде чем будет найден наиболее оптимальное решение задачи.

8. Литература

1. Цой, Ю. Р. Разработка генетического алгоритма настройки искусственной нейронной сети — ВКР, Томск 2004.
2. И.Б. Бондаренко, Ю.А. Гатчин, В.Н. Гераничев, Синтез оптимальных искусственных нейронных сетей с помощью модифицированного генетического алгоритма, - УДК 519.816
3. Towardsdatascience. Genetic Algorithms + Neural Networks = Best of Both Worlds. <https://towardsdatascience.com/gas-and-nns-6a41f1e8146d>