

## Содержание

<b>1</b>	<b>Введение</b>	<b>1</b>
<b>2</b>	<b>Постановка задачи</b>	<b>3</b>
2.1	Описание данных и постановка задачи . . . . .	3
2.2	Постановка задачи порождения модели . . . . .	3
2.2.1	WANN №1 . . . . .	3
2.2.2	WANN №2 . . . . .	4
<b>3</b>	<b>Метрики</b>	<b>4</b>
3.1	Эталонные метрики . . . . .	4
3.2	Структурные метрики . . . . .	5
3.3	Методы сравнения метрик . . . . .	6
<b>4</b>	<b>Генетический алгоритм и его модификации</b>	<b>7</b>
4.1	Общие понятия . . . . .	7
4.1.1	WANN 1 . . . . .	8
4.1.2	WANN 2 . . . . .	8
4.2	Классический генетический алгоритм . . . . .	8
4.3	Детектирование стагнаций с помощью метрики . . . . .	9
4.4	Детектирование стагнаций по значению ошибок . . . . .	9
4.5	Генетический алгоритм с кластеризацией . . . . .	10
<b>5</b>	<b>Вычислительный эксперимент</b>	<b>10</b>
5.1	Метрики . . . . .	10
5.2	Генетические алгоритмы . . . . .	12
<b>6</b>	<b>Заключение</b>	<b>16</b>
<b>7</b>	<b>Список литературы</b>	<b>16</b>

### Аннотация

В работе решается задача аппроксимации неизвестной зависимости с помощью агностических нейронных сетей. На множестве агностических нейронных сетей вводится структурная метрика. Предлагается несколько методов сравнения метрик друг с другом. На искусственно сгенерированных данных производится сравнение метрик. Рассматриваются четыре вариации генетического алгоритма, используемые для генерации агностических нейронных сетей.

**Ключевые слова:** WANN, структурное обучение, метрика, генетический алгоритм.

## 1 Введение

Одним из методов для решения задачи восстановления функциональной зависимости по набору исходных данных является символьная регрессия

[1]. Задача символьной регрессии заключается в отыскании такой композиции функций из некоторого заданного множества, которая бы наилучшим образом аппроксимировала зависимость. Джон Коза предложил метод, являющийся аналогом эволюционного алгоритма [2], получивший дальнейшее развитие в работе Ивана Зелинки [3]. Анна Варфоломеева для решения задачи символьной регрессии использует совершенно другой подход, строящий дерево композиции на основе уже имеющихся примеров [4].

Основной интерес для данной работы представляет статья Андрея Кулунчакова и Вадима Викторовича Стрижова [5], в которой рассматривается генерация непараметрических моделей, являющихся композицией некоторых примитивных функций.

**Definition 1.1.** Функция  $\mu : X \times X \longrightarrow \mathbb{R}_{\geq 0}$  называется метрикой, если она удовлетворяет следующим условиям:

1.  $\mu(x_1, x_2) = 0 \iff x_1 = x_2$
2.  $\mu(x_1, x_2) = \mu(x_2, x_1)$
3.  $\mu(x_1, x_2) + \mu(x_2, x_3) \geq \mu(x_1, x_3)$  (неравенство треугольника)

**Definition 1.2.** Структурная метрика - метрика, зависящая только от структуры модели.

**Definition 1.3.** Эталонная метрика - метрика, которая вводится из эмпирических соображений и считается идеально подходящей для рассматриваемой задачи.

В [5] на множестве моделей задается три структурных метрики. В качестве эталонной метрики берется следующая:

$$\eta(f, f') = \frac{1}{|C|(|C| - 1)} \sum_{d_j, d_k \in C} [f(d_j) < f(d_k)][f'(d_j) > f'(d_k)]$$

Здесь  $C$  - некоторое наперед заданное множество. Утверждается, что структурные метрики близки к эталонной. Более строго, предлагается считать метрики  $\mu, \eta$  близкими на множестве деревьев  $\mathfrak{M}$ , если

$$\alpha(\mathfrak{M}) = \nu([\mu(f, f') \leq \alpha_1] \longrightarrow [\eta(f, f') \leq \alpha_2] | f, f' \in \mathfrak{M}) \geq 1 - \varepsilon$$

где  $\alpha_1, \alpha_2, \varepsilon$  - константы,  $\nu(A)$  - частота события  $A$ . В нашей работе не используется такой способ сравнения метрик, потому что он требует задания сразу трех констант, что чрезмерно.

Далее в [5] предлагается детектировать стагнации с помощью структурной метрики, считая, что алгоритм застagnировался в случае, если диаметр популяции ниже некоторого фиксированного порога, то есть, иными словами, если все деревья в популяции слишком похожи друг на друга. В этом случае половина худших деревьев в популяции меняется на заново сгенерированные.

В данной работе схожий подход был применен для агностических нейронных сетей (Weight Agnostic Neural Networks), подробно описанных в статье [6]. Суть данной модели заключается в том, что все веса на ребрах нейронной сети заменяются одним единственным весом, то есть пространство параметров становится одномерным.

## 2 Постановка задачи

### 2.1 Описание данных и постановка задачи

Задана пара  $(X, y)$ ,  $X \in \mathbb{R}^{n \times m}$ ,  $y \in \mathbb{R}^{n \times k}$ , строки матриц  $X$  и  $y$  находятся в некоторой неизвестной функциональной зависимости. Задано множество порождающих функций  $\mathfrak{G}$ .  $\mathfrak{F}_{\mathfrak{G}}$  - множество агностических нейронных сетей, у которых функции активации содержатся в  $\mathfrak{G}$ .

Требуется найти оптимальную модель  $f : \mathbb{W} \times \mathbb{X} \rightarrow \mathbb{R}^{n \times k}$ ,  $f \in \mathfrak{F}_{\mathfrak{G}}$ ,  $\mathbb{W}$  - множество параметров модели (в случае агностической нейронной сети состоит из одного элемента).

Более формально, если задана функция ошибки  $S(y_{pred}, y_{true})$ , то необходимо найти  $f \in \mathfrak{F}_{\mathfrak{G}}$  и  $w \in \mathbb{W}$ , минимизирующие значение функции  $S$ :

$$(f, w) = \arg \min_{(f, w) \in \mathfrak{F}_{\mathfrak{G}} \times \mathbb{W}} S(f^w(x), y) \quad (1)$$

Здесь и далее запись  $f^w(x)$  эквивалентна  $f(w, x)$ .

В качестве функции ошибки используется среднее квадратов регрессионных остатков:

$$S(y_{pred}, y_{true}) = \frac{1}{n} \|y_{pred} - y_{true}\|_2^2 \quad (2)$$

### 2.2 Постановка задачи порождения модели

В данной работе рассматриваются два способа представления агностических нейронных сетей. В дальнейшем нейросети, представленные первым способом, мы будем для краткости называть **WANN №1**, нейросети, представленные вторым способом - **WANN №2**. Если номер не указывается, это означает, что сказанное не зависит от способа представления агностической нейронной сети.

#### 2.2.1 WANN №1

Рассматривается неполносвязная сеть прямого распространения, у которой все параметры равны между собой.

**Definition 2.1.** Нейронную сеть можно разбить на слои, если существует такое разбиение множества нейронов на подмножества (слои), что:

- никакие 2 нейрона внутри одного слоя не соединены друг с другом

- на множестве слоев можно ввести линейный порядок, т.ч. если нейрон из слоя  $L_1$  связан с нейроном из слоя  $L_2$ , то  $L_1 < L_2$  и  $\nexists L : L_1 < L < L_2$ , или  $L_1 > L_2$  и  $\nexists L : L_1 > L > L_2$ .
- Если  $n_1 \in L_1$ ,  $L_1 < L_2$ ,  $\nexists L : L_1 < L < L_2$ , то  $\nexists n_2 \in L_2 : n_1$  и  $n_2$  соединены.
- Если  $n_1 \in L_1$ ,  $L_1 > L_2$ ,  $\nexists L : L_1 > L > L_2$ , то  $\nexists n_2 \in L_2 : n_1$  и  $n_2$  соединены.

Важной особенностью данной модели является то, что ее можно разбить на слои.

**Definition 2.2.** Подмножество нейронов называется слоем, если:

### 2.2.2 WANN №2

В данной модели нейроны могут соединяться между собой в произвольном порядке, единственное требование - в графе конфигураций не должно быть направленных циклов.

## 3 Метрики

### 3.1 Эталонные метрики

Важно отметить, что для эталонной метрики допустимо невыполнение первого свойства ( $\mu(x, y) = 0 \iff x = y$ ). В таком случае мы просто считаем, что метрика отождествляет  $x$  и  $y$ , для нее они неразличимы. Можно еще сказать, что метрика разбивает наше множество на классы эквивалентности и мы отныне работаем только с их представителями.

Ниже приведен список метрик, которые в данной статье считаются эталонными для множества агностических нейронных сетей:

$$\mu_1(f_1, f_2) = \frac{1}{n|W|} \sum_{w \in W} \sum_{i=1}^n \|f_1^w(x_i) - f_2^w(x_i)\|_2 \quad (3)$$

$$\mu_2(f_1, f_2) = \max_{w_1, w_2 \in W} \frac{1}{n} \sum_{i=1}^n \|(f_1^{w_1}(x_i) - f_2^{w_2}(x_i))\|_2 \quad (4)$$

Здесь  $x_i, x_j \in X$ ,  $X \subset \mathbb{R}^n$  - некоторый конечный набор точек,  $W$  - конечное множество параметров. Заметим, что эти метрики определены для агностических нейронных сетей, чьи входные и выходные размерности совпадают.

**Lemma 3.1.** Все предложенные функции являются метриками.

*Доказательство.* Выполнение всех свойств, кроме неравенства треугольника, очевидно. Докажем, что и последнее тоже выполнено.

1) Следует из того, что для каждого фиксированного  $w$  неравенство выполнено, так как при фиксированном  $w$  имеем не что иное как сумму евклидовых норм.

2) Пусть максимум для  $(f_1, f_3)$  достигается на  $(w'_1, w'_2)$ . Тогда  $\sum_{i=1}^n \|(f_1^{w'_1}(x_i) - f_2^{w'_2}(x_i))\|_2 \leq$  (по свойствам евклидовой метрики)  $\leq \sum_{i=1}^n \|(f_1^{w'_1}(x_i) - f_2^{w'_2}(x_i))\|_2 + \sum_{i=1}^n \|(f_2^{w'_1}(x_i) - f_3^{w'_2}(x_i))\|_2 \leq \max_{w_1, w_2 \in W} \sum_{i=1}^n \|(f_1^{w_1}(x_i) - f_2^{w_2}(x_i))\|_2 + \sum_{i=1}^n \|(f_2^{w_1}(x_i) - f_3^{w_2}(x_i))\|_2$   $\square$

Эталонные метрики обладают существенным недостатком: они трудно-вычислимы. Как следствие, хочется уметь приближать эталонную метрику некоторой функцией близости, которая, с одной стороны, была бы легко вычислима, а с другой, как можно лучше согласовывалась бы с эталонной.

Для лаконичности изложения под словом метрика мы в дальнейшем будем понимать любую функцию близости, потому что строгое соблюдение свойств метрики нам больше нигде не понадобится.

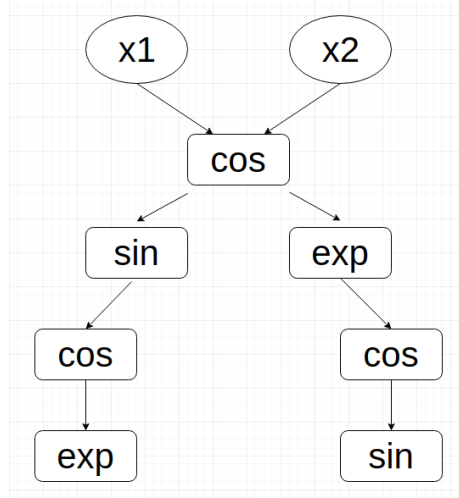
### 3.2 Структурные метрики

Важно подчеркнуть, что, вообще говоря, вводится не метрика, а функция близости, которая метрикой называется только для лаконичности изложения. Из свойств метрики можно гарантировать лишь неотрицательность и симметричность.  $\mu(x, y) = 0 \iff x$  изоморфен  $y$  - означало бы решить задачу проверки графов на изоморфизм, принадлежность которой классу  $P$  до сих пор не доказана.  $\mu(x, y) = 0 \iff x$  и  $y$  ведут себя одинаково, как функции - означало бы решить задачу SAT, которая является NP-полной.

Каждой агностической нейронной сети поставим в соответствие матрицу размера  $(|\mathfrak{G}| + ar(f)) \times |\mathfrak{G}|$ , где  $ar(f)$  - арность функции, или, что то же самое, размер входного слоя сети. Строкам соответствуют все функции из  $\mathfrak{G}$ , а так же  $x_1, \dots, x_{ar(f)}$ , столбцам - только функции из  $\mathfrak{G}$ .

В ячейке с координатами  $(i, j)$  стоит число, равную количеству направленных ребер в агностической нейронной сети, ведущих из нейрона с функцией активации соответствующей  $i$ -ой строке в нейрон с функцией активации, соответствующей  $j$ -ому столбцу (определение аналогично для  $x_1, \dots, x_{ar(f)}$ ).

Например, пусть нейросеть имеет следующий вид:



Тогда ей соответствует матрица

$$\begin{bmatrix}
 & \sin & \cos & \exp \\
 \sin & 0 & 1 & 0 \\
 \cos & 2 & 0 & 2 \\
 \exp & 0 & 1 & 0 \\
 x_1 & 0 & 1 & 0 \\
 x_2 & 0 & 1 & 0
 \end{bmatrix}$$

Первый способ измерять расстояние между двумя агностическими нейронными сетями по построенным матрицам - брать евклидову норму разности векторизованных матриц.

$$\mu_1(f_1, f_2) = \|v_1 - v_2\|_2 \quad (5)$$

Этот способ можно обобщить. Пусть  $(f_1, f_2)$  - две агностических нейронных сети,  $v_1, v_2$  - построенные по ним векторизованные матрицы. Тогда рассмотрим данную пару как объект, признаковым описанием которого является вектор  $abs(v_{f_1} - v_{f_2})$ , где  $abs$  - операция взятия модуля, применяемая покомпонентно. Требуется построить отображение  $g : \{abs(v_{f_1} - v_{f_2}) | f_1, f_2 \in \mathfrak{F}_{\mathfrak{G}}\} \rightarrow \mathbb{R}_{\geq 0}$ , так чтобы  $g$  как можно лучше согласовывалось с эталонной метрикой. До этого мы взяли  $g(v) = \|v\|_2$ . Возможно существует более подходящая функция.

Для поиска оптимальной  $g$  мы будем обучать полносвязную нейросеть прямого распространения предсказывать эталонную метрику по признаковому описанию пары WANN.

$$\mu_2(f_1, f_2) = g(v_1, v_2) \quad (6)$$

### 3.3 Методы сравнения метрик

В данном разделе вводятся четыре способа сравнения метрик.

Пусть дано фиксированное конечное множество  $T$  (в нашем конкретном случае  $T \subset \mathfrak{F}_6$ ). Необходимо ввести функцию  $\gamma(\mu_1, \mu_2)$ , по значению которой можно было бы понять, насколько согласуются между собой метрики на данном множестве.

**1 способ.** Посчитать коэффициент корреляции Пирсона.

$$\text{corr}(\mu_1, \mu_2) = \frac{\overline{\mu_1 \mu_2} - \overline{\mu_1} \cdot \overline{\mu_2}}{(\overline{\mu_1^2} - \overline{\mu_1}^2)(\overline{\mu_2^2} - \overline{\mu_2}^2)} \quad (7)$$

Здесь  $\overline{\mu_i} = \frac{1}{n(n-1)} \sum_{t_1, t_2 \in T} \mu_i(t_1, t_2)$

**2 способ.**

$$\nu(\mu(f_1, f_2) < \mu(f_1, f_3) \wedge \eta(f_1, f_2) > \eta(f_1, f_3)) \quad (8)$$

$\nu$  - частота события

**3 способ.**

$$\frac{\inf_{x,y} \frac{\mu_1(x,y)}{\mu_2(x,y)}}{\sup_{x,y} \frac{\mu_1(x,y)}{\mu_2(x,y)}} \quad (9)$$

**4 способ.** Пусть для множества  $T$  заданы матрицы попарных расстояний  $D_1, D_2$  для метрик  $\mu_1$  и  $\mu_2$  соответственно. Нормализуем эти матрицы (эквивалентно делению каждого элемента на диаметр множества  $T$ ). Векторизовав их, получим вектора  $v_1$  и  $v_2$  соответственно. Тогда в качестве функции близости метрик можно рассмотреть

$$\|v_1 - v_2\|_2 \quad (10)$$

## 4 Генетический алгоритм и его модификации

### 4.1 Общие понятия

Хотя размерность пространства параметров в агностических нейронных сетях и так снижена до 1, все равно перебор по сетке может занимать слишком много времени. Поэтому рассматриваются 3 функции ошибки, по значению которых будет производиться отбор в конце каждой эпохи генетического алгоритма.

$$\min_{w \in \mathbb{W}'} S(f^w(x), y) \quad (11)$$

$$\frac{1}{|\mathbb{W}'|} \sum_{w \in \mathbb{W}'} S(f^w(x), y) \quad (12)$$

$$\max_{w \in \mathbb{W}'} S(f^w(x), y) \quad (13)$$

Здесь  $\mathbb{W}'$  - некоторое небольшое множество возможных значений параметра (в данной работе  $\{-2, -1, -0.5, 0.5, 1, 2\}$ ). В конце работы ген. алгоритма полученный WANN оптимизируется по параметру.

#### 4.1.1 WANN 1

**Мутация.** Операция мутации производит 3 возможных действия:

- Удаляет случайно выбранный слой, соседние с ним слои соединяет между собой произвольным образом.
- Заменяет случайно выбранный слой на новый, соединяет с соседними слоями произвольным образом.
- В случайное место добавляет новый сгенерированный слой, соединяет с соседними слоями произвольным образом.

**Скрещиванье.** Операция скрещиванья:

1. Случайным образом выбирает в первой сети и второй сети по одному слою.
2. Все слои, которые шли в первой сети после выбранного в ней слоя, заменяет на слои второй сети, начиная с выбранного.

В следующем примере продемонстрирована работа алгоритма скрещиванья:

$$L_1^1 L_2^1 \dots L_k^1 \dots L_s^1, L_1^2 L_2^2 \dots L_m^2 \dots L_r^2 \longrightarrow L_1^1 \dots L_k^1 L_m^2 \dots L_r^2$$

Здесь  $L_i^j$  -  $i$ -й слой  $j$ -й нейросети.

#### 4.1.2 WANN 2

**Мутация.** Операция мутации производит 3 возможных действия:

- Добавляет новое ребро.
- Добавляет новый нейрон на уже существующее ребро.
- Изменяет функцию активации у уже существующего нейрона.

**Скрещиванье.** Отсутствует.

### 4.2 Классический генетический алгоритм

В классической вариации генетического алгоритма ведется ненаправленный поиск оптимальной структуры нейросети с помощью применения к популяции описанных ранее операциях скрещивания и мутации.

**Result:** WANN

```
generate_population();  
while error >  $\varepsilon$  or iteration < max_iterations do  
    mutate();  
    crossover(); //for WANN №1;  
    select();  
end  
return population[0];
```

**Algorithm 1:** Классический генетический алгоритм



### 4.3 Детектирование стагнаций с помощью метрики

В статье Кулунчакова [4] предложено детектировать стагнации в работе генетического алгоритма с помощью метрики: если диаметр популяции (как множества) не превосходит некоторого  $\varepsilon$ , то мы считаем, что алгоритм застагнировался и меняем половину худших деревьев в популяции на совершенно случайные.

```
Result: WANN
generate_population();
while  $error > \varepsilon$  or  $iteration < max\_iterations$  do
    mutate();
    crossover(); //for WANN №1;
    select();
    if  $diam(population) \leq \delta$  then
        for  $wann$  in  $population[\frac{n}{2} :]$  do
            regenerate( $wann$ );
        end
    end
end
return population[0];
```

**Algorithm 2:** Генетический алгоритм с детектированием стагнаций с помощью метрики

### 4.4 Детектирование стагнаций по значению ошибок

Если лучшее значение ошибки в популяции перестает существенно меняться, то мы считаем это стагнацией и действуем аналогично предыдущему пункту, т.е. заменяем половину худших деревьев в популяции на совершенно случайные.

```
Result: WANN
generate_population();
while  $error > \varepsilon$  or  $iteration < max\_iterations$  do
    mutate();
    crossover(); //for WANN №1;
    select();
    if  $\frac{error[iteration-l]}{error[iteration]} \geq \delta$  then
        for  $wann$  in  $population[\frac{n}{2} :]$  do
            regenerate( $wann$ );
        end
    end
end
return population[0];
```

**Algorithm 3:** Генетический алгоритм с детектированием стагнаций по значению ошибки

## 4.5 Генетический алгоритм с кластеризацией

В этой вариации генетического алгоритма, если ошибка перестает изменяться, то множество нейросетей в популяции кластеризуется по структурной метрике, а потом в каждом кластере все нейросети, кроме той, у которой ошибка среди них минимальна, заменяются на заново сгенерированные. Тем самым мы избавляемся от похожих нейросетей в популяции.

```
Result: WANN
generate_population();
while  $error > \varepsilon$  or  $iteration < max\_iterations$  do
    mutate();
    crossover(); //for WANN №1;
    select();
    if  $\frac{error[iteration-1]}{error[iteration]} \geq \delta$  then
        clusters := cluster(population);
        for cluster in clusters do
            for wann in cluster \ {best_in_cluster} do
                | wann.regenerate();
            end
        end
    end
end
return population[0];
```

**Algorithm 4:** Генетический алгоритм с кластеризацией

## 5 Вычислительный эксперимент

**Цель:** сравнить структурные метрики (5) и (6), сравнить скорости сходимости всех вариаций генетического алгоритма с функциями ошибки (11), (12), (13).

### 5.1 Метрики

Нейросеть с тремя скрытыми слоями была обучена предсказывать эталонные метрики по построенной матрице, как это уже описывалось ранее.

Ниже нейросеть, обученная предсказывать (3), обозначается **net1**, а нейросеть, обученная предсказывать 4 - **net2**. В качестве обучающей выборке были взяты попарные расстояния между ста сгенерированными WANN'ами, в качестве тестовой - попарные расстояния между другими ста.

Далее представлены результаты для агностических сетей первого типа.

	standart 1	standart 2
structural 1	0.31	0.37
net1	0.12	0.22
net2	0.13	0.35

Таблица 1: 1 способ

	standart 1	standart 2
structural 1	0.44	0.46
net1	0.49	0.49
net2	0.52	0.51

Таблица 2: 2 способ

	standart 1	standart 2
structural 1	0.00	0.00
net1	0.01	0.03
net2	0.00	0.20

Таблица 3: 3 способ

	standart 1	standart 2
structural 1	21.92	29.01
net1	24.92	51.92
net2	22.79	27.92

Таблица 4: 4 способ

Таким образом, получаем соответствие между эталонной метрикой и той, которая лучше всего с ней согласуется.

	1 способ	2 способ	3 способ	4 способ
standart 1	structural 1	structural 1	net 1	structural 1
standart 2	structural 1	structural 1	net 2	net 2

Таблица 5: Соответствие между эталонными и структурными метриками

Приведем аналогичные результаты для агностических нейронных сетей второго типа.

	standart 1	standart 2
structural 1	0.32	0.34
net1	0.28	0.20
net2	0.18	0.28

Таблица 6: 1 способ

	standart 1	standart 2
structural 1	0.44	0.47
net1	0.44	0.47
net2	0.47	0.47

Таблица 7: 2 способ

	standart 1	standart 2
structural 1	0.00	0.00
net1	0.00	0.00
net2	0.00	0.08

Таблица 8: 3 способ

	standart 1	standart 2
structural 1	36.94	21.82
net1	23.12	45.04
net2	27.39	31.30

Таблица 9: 4 способ

	1 способ	2 способ	3 способ	4 способ
standart 1	structural 1	structural 1 / net 1	—	net 1
standart 2	structural 1	—	net 2	structural 1

Таблица 10: Соответствие между эталонными и структурными метриками

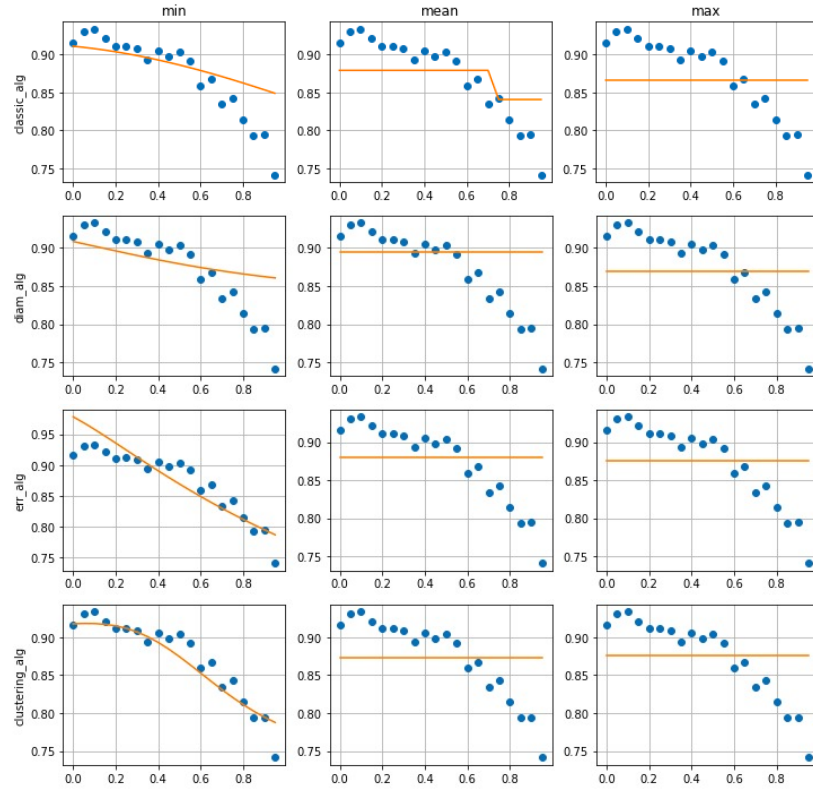
Таким образом, предсказание эталонных метрик с помощью нейросетей на тестовой выборке показывает хоть и не слишком плохие результаты, но не лучшие, чем 5, поэтому в дальнейшем мы будем использовать именно последнюю.

## 5.2 Генетические алгоритмы

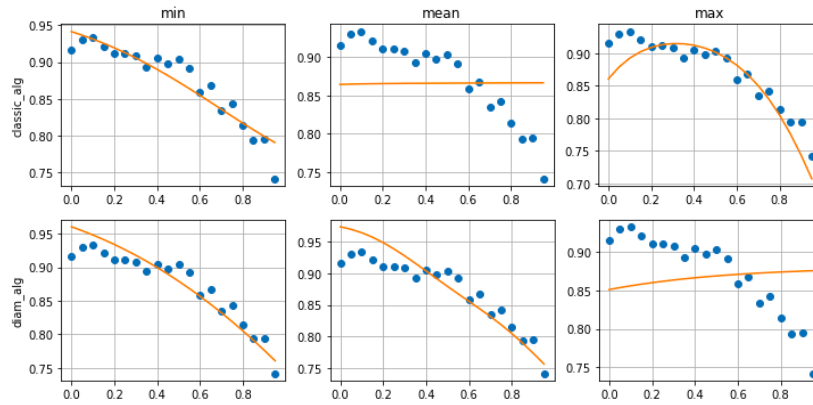
Сначала работа алгоритмов была протестирована на искусственно сгенерированных данных. Каждая из модификаций генетического алгоритма была

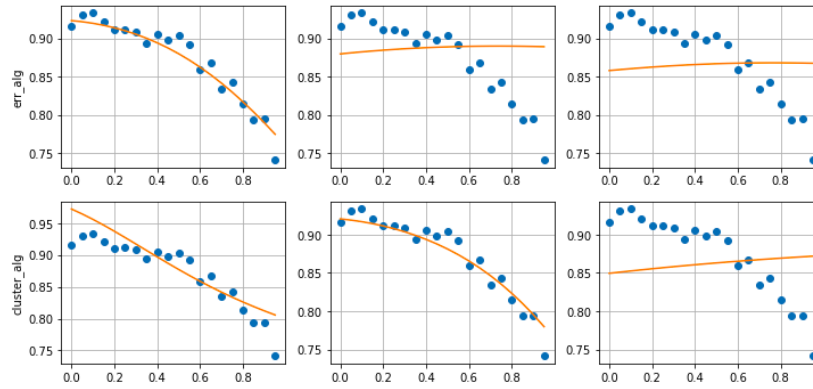
запущена на 50 эпох с каждой из трех функций ошибки (11), (12), (13). Получившаяся агностическая нейронная сеть была оптимизирована по параметру. Ниже на графиках представлены исходное множество точек и их функциональная аппроксимация.

Для агностических сетей первого типа:



Для агностических сетей второго типа:



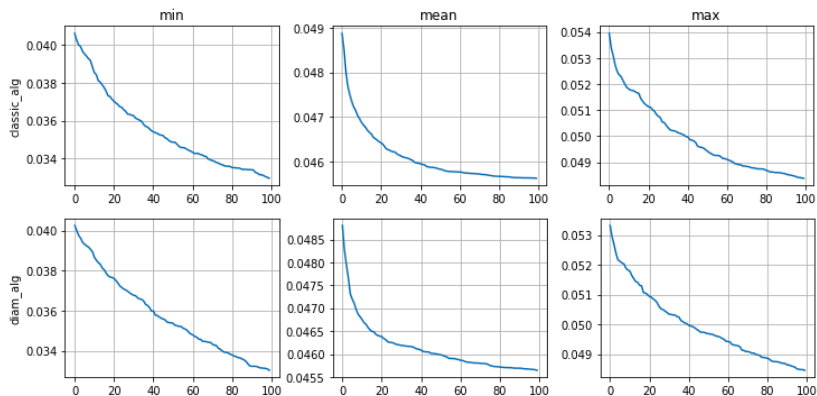


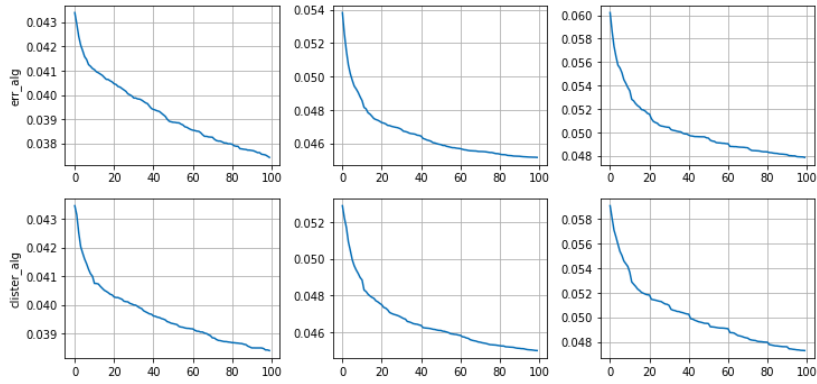
После этого был поставлен эксперимент на датасете Iris. Ниже на графиках приведена зависимость усредненного по экспериментам значения функции ошибки от числа эпох. Все 4 модификации генетического алгоритма была запущены по 100 раз с каждой из трех функций ошибки 11, (12), (13) со следующими параметрами:

- 100 эпох
- размер популяции - 10
- количество мутаций за одну эпоху - 5/10 (WANN №1/WANN №2)
- количество скрещиваний 5/0 (WANN №1/WANN №2)

Результат усреднялся по всем 100 экспериментам. Так же в таблицах представлено значение функции ассигасу на полученной агностической нейронной сети после оптимизации параметра. Оно так же было усреднено по всем экспериментам.

Для агностических сетей первого типа:

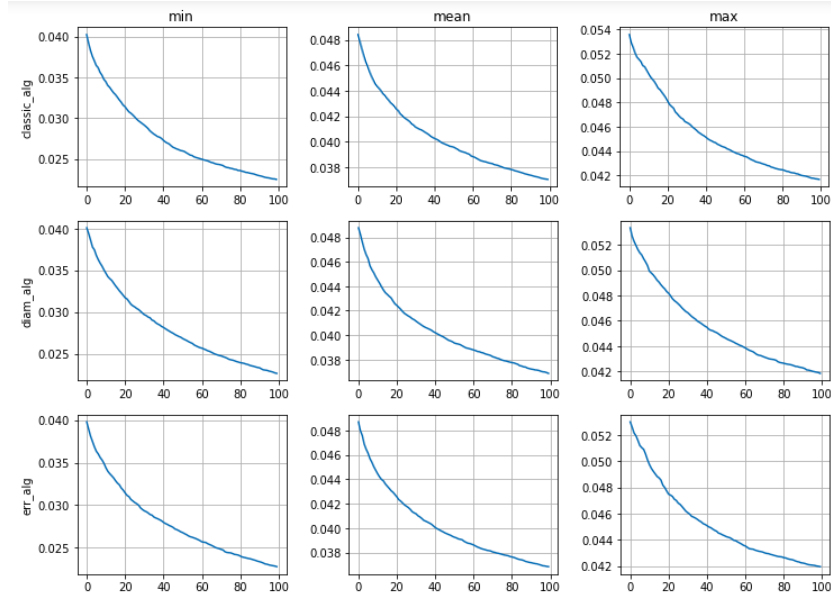


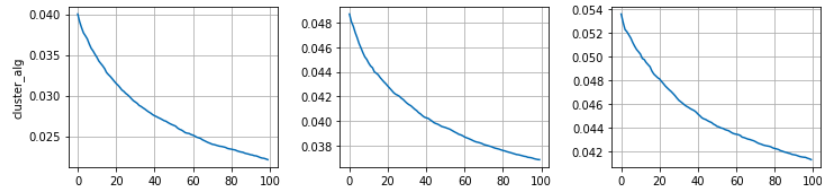


	min	mean	max
classic	0.70	0.37	0.33
diameter	0.70	0.36	0.33
error	0.57	0.39	0.32
clustering	0.61	0.37	0.33

Таблица 11:

Для агностических сетей второго типа:





	min	mean	max
classic	0.88	0.77	0.64
diameter	0.87	0.77	0.63
error	0.88	0.81	0.62
clustering	0.88	0.80	0.65

Таблица 12:

## 6 Заключение

Итак, в данной работе были предложены:

- две структурных метрики на множестве агностических нейронных сетей
- четыре метода сравнения метрик, заданных на конечном множестве
- три модификации классического генетического алгоритма для генерации агностических нейронных сетей

Улучшить скорость сходимости классического генетического алгоритма не получилось. Мы дадим одно из возможных объяснений этому. В [5] генетический алгоритм производит ненаправленный поиск подходящей модели. При построении WANN №2 все мутации приводят только к усложнению модели, поэтому в этом случае генетический алгоритм представляет собой рандомизированный жадный, в следствие чего поиск подходящей модели уже не является ненаправленным. Именно поэтому замена худших моделей в популяции на случайно сгенерированные не ведет к улучшению: мы пытаемся увеличить скорость сходимости направленного поиска, подменяя его на ненаправленный.

## 7 Список литературы

- [1] Riccardo Poli, William B. Langdon, Nicholas F. McPhee A Field Guide to Genetic Programming, 2008.  
 [2] Koza, J. R. Genetic programming. In Williams, J. G. and Kent, A. (editors) // Encyclopedia of Computer Science and Technology, 1998. Vol. 39. P.: 29-43.  
 [3] Ivan Zelinka, Zuzana Oplatkova, and Lars Nolle Analytic programming –



- symbolic regression by means of arbitrary evolutionary algorithms, August 2008.
- [4] А. А. Варфоломеева Выбор признаков при разметке библиографических списков методами структурного обучения, 2013.
  - [5] Kulunchakov A.S., Strijov V.V. Generation of simple structured Information Retrieval functions by genetic algorithm without stagnation // Expert Systems with Applications, 2017, 85 : 221—230.
  - [6] <https://weightagnostic.github.io/>