

# Содержание

<b>1</b>	<b>Введение</b>	<b>1</b>
<b>2</b>	<b>Постановка задачи</b>	<b>2</b>
2.1	Описание данных и постановка задачи . . . . .	2
2.2	Способ представления агностической неронной сети . . . . .	2
2.2.1	WANN №1 . . . . .	3
2.2.2	WANN №2 . . . . .	3
<b>3</b>	<b>Метрики</b>	<b>3</b>
3.1	Эталонные метрики . . . . .	3
3.2	Структурные метрики . . . . .	4
3.3	Методы сравнения метрик . . . . .	5
<b>4</b>	<b>Генетический алгоритм и его модификации</b>	<b>6</b>
4.1	Общие понятия . . . . .	6
4.1.1	WANN 1 . . . . .	7
4.1.2	WANN 2 . . . . .	7
4.2	Классический генетический алгоритм . . . . .	7
4.3	Детектирование стогнаций с помощью метрики . . . . .	8
4.4	Детектирование стогнаций по значению ошибок . . . . .	8
4.5	Генетический алгоритм с кластеризацией . . . . .	9

## Аннотация

В работе исследуется задача аппроксимации неизвестной зависимости с помощью агностических нейронных сетей. На множестве агностических нейронных сетей вводится структурная метрика. Предлагается ряд эталонных метрик, а так же методы сравнения метрик друг с другом. На искусственно сгенерированных данных производится сравнение метрик указанными способами. Рассматриваются три вариации генетического алгоритма, используемые для генерации агностических нейронных сетей.

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

## 1 Введение

**Актуальность темы.** Во многих прикладных задачах требуется восстановить неизвестную зависимость, содержащуюся в данных. Одну из основных сложностей составляет именно отыскание наилучшей модели.

**Цель работы.** Ввести структурную метрику на множестве агностических нейронных сетей, предложить модификацию генетического алгоритма для генерации агностических нейронных сетей.

**Методы исследования.** Весь необходимый для вычислительного эксперимента код был написан на языке Python, в качестве примера были взяты

данные из датасета Boston Housing.

**Научная новизна.**

- Предложены методы сравнения метрик, заданных на конечных множествах.
- Рассмотрена структурная метрика на множестве агностических нейронных сетей.
- Изучены две модификации классического генетического алгоритма для генерации агностических нейронных сетей.

**Практическая ценность.** Написан класс агностических нейронных сетей, реализованы модификации генетических алгоритмов.

**Обзор литературы.** Одним из методов восстановления функциональной зависимости по имеющимся данным является символьная регрессия [1].

## 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  $\mathfrak{F}$ .  $\mathfrak{F}_{\mathfrak{G}}$  - 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  $S$ :

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

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

В качестве функции ошибки используется среднее квадратов регрессионных остатков:

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

### 2.2 Способ представления агностической нейронной сети

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

Агностическая нейронная сеть в данном подходе представляется  $n$  слоями, причем могут быть соединены между собой только нейроны соседних слоев. В каждом слое обязательно присутствует bias. Иными словами, рассматривается неполносвязная сеть прямого распространения, у которой все параметры равны между собой.

### 2.2.2 WANN №2

У агностической нейронной сети в данном подходе присутствуют только входной и выходной слои. Остальные нейроны могут соединяться между собой в произвольном порядке, единственное требование - в графе конфигураций не должно быть направленных циклов.

## 3 Метрики

### 3.1 Эталонные метрики

Напомним основополагающее определение.

**Definition 3.1.** Метрикой на множестве  $X$  называется функция, удовлетворяющая следующим свойствам:

В общем случае эталонной метрикой на множестве  $X$  будем называть метрику, которая постулируется исходя из каких-то эмпирических соображений. Такие метрики нужны для того, чтобы сравнивая с ними другие метрики понимать, насколько последние близки к нашим ожиданиям.

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:

$$\begin{aligned} 1) \mu_1(f_1, f_2) &= \frac{1}{n|W|} \sum_{w \in W} \sum_{i=1}^n \|f_1^w(x_i) - f_2^w(x_i)\|_2 \\ 2) \mu_2(f_1, f_2) &= \max_{w_1, w_2 \in W} \frac{1}{n} \sum_{i=1}^n \|(f_1^{w_1}(x_i) - f_2^{w_2}(x_i))\|_2 \end{aligned}$$

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)  $\leq \sum_{i=1}^n \|(f_1^{w'_1}(x_i) - f_2^{w'_2}(x_i))\|_2 +$   
 $\sum_{i=1}^n \|(f_2^{w'_1}(x_i) - f_3^{w'_2}(x_i))\|_2 \leq \max_{w_1, w_2 \in W} \sum_{i=1}^n \|(f_1^{w_1}(x_i) - f_2^{w_2}(x_i))\|_2 + \sum_{i=1}^n \|(f_2^{w_1}(x_i) - f_3^{w_2}(x_i))\|_2$   $\square$

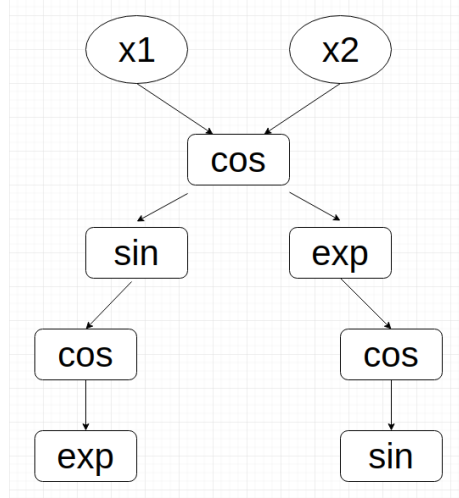
### 3.2 Структурные метрики

Структурной метрикой мы будем называть метрику, которая определена на множестве агностических нейронных сетей и зависит только от их структуры. Важно подчеркнуть, что, вообще говоря, вводится не метрика, а мера близости, метрикой мы ее будем называть лишь 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.

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

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 - вытягивать матрицы в вектор и брать евклидову норму разности этих двух получившихся векторов.

Второй способ - подавать модуль разности векторов на вход нейронной сети, предварительно обученной предсказывать одну из эталонных метрик.

### 3.3 Методы сравнения метрик

В данном разделе вводятся три способа сравнения метрик. Неформально говоря, мы хотим, чтобы структурная метрика была в некотором роде близка к эталонной.

Более строго, пусть дано фиксированное конечное множество  $T$  (в нашем конкретном случае  $T \subset \mathfrak{F}_{\mathfrak{G}}$ ). 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 обычную корреляцию.

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

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

**2 способ.**  $\nu(\mu(f_1, f_2) < \mu(f_1, f_3) \wedge \eta(f_1, f_2) > \eta(f_1, f_3))$ ,  $\nu$  - частота события  
**3 способ.** Напомним определение эквивалентных метрик.

**Definition 3.2.** Метрики  $\mu$  и  $\nu$  называются эквивалентными, если

$$\exists \alpha, \beta \quad \forall x, y \quad \alpha \nu(x, y) \leq \mu(x, y) \leq \beta \nu(x, y).$$

В дальнейшем, говоря про  $\alpha$  и  $\beta$ , мы имеем в виду наибольшее среди всех  $\alpha$ , удовлетворяющих данному неравенству, и наименьшее среди всех таких  $\beta$ .

На конечных множествах все метрики эквивалентны. Хотим рассмотреть  $g(\alpha, \beta)$ , которая, с одной стороны, показывала бы, насколько  $\alpha$  и  $\beta$  близки друг к другу (это и определяет меру близости наших метрик), с другой стороны, была бы устойчива к масштабированию метрик (т.е. не менялась бы при умножении метрик на константу). Одной из таких функций является  $\frac{\alpha}{\beta}$ , ее и будем рассматривать. Две метрики тем ближе, чем ближе значение этой функции к единице.

**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  $T$ ), вытянем каждую матрицу в вектор и будем смотреть на евклидово расстояние между этими векторами.

## 4 Генетический алгоритм и его модификации

### 4.1 Общие понятия

Напомним, что нашей исходной задачей являлось отыскание пары  $(f, w)$ , т.ч.

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

Хотя размерность пространства параметров в агностических нейронных сетях и так снижена до 1, все равно перебор по сетке может занимать слишком много времени. Поэтому мы рассмотрим три подхода к данной проблеме.

**1 подход.** На каждой итерации генетического алгоритма рассматривается усредненное значение ошибки по разреженной сетке (в данной работе -2, -1, -0.5, 0.5, 1, 2). Таким образом ищется нейросеть, именно структура которой хорошо подходит под решение задачи, а параметр слабо влияет на результат.

**2 подход.** На каждой итерации генетического алгоритма рассматривается максимальное значение ошибки по разреженной сетке. Цель та же, что и у предыдущего подхода. **3 подход.** На каждой итерации генетического алгоритма рассматривается минимальное значение ошибки по разреженной сетке. Такой подход ближе в первоначальной постановке задачи.

#### 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.

**Скращивание.** Операция скращивания:

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 скращивания:

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

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.

**Скращивание.** 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 скращивания 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 стогнаций с помощью метрики

В статье Кулунчакова предложено детектировать стогнации в работе генетического алгоритма с помощью метрики: если диаметр популяции (как множества) не превосходит некоторого  $\varepsilon$ , то мы считаем, что алгоритм застогнировался 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 стогнаций с помощью метрики

### 4.4 Детектирование стогнаций по значению ошибок

Если лучшее значение ошибки в популяции перестает существенно меняться, то мы считаем это стогнацией 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 стогнаций 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  $\frac{error[iteration-l]}{error[iteration]} \geq \delta$  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