Московский Государственный Университет им. М.В.Ломоносова
Механико-математический факультет
Кафедра общих проблем управления

# Дипломная работа

## Численное моделирование в задаче определения оптимальных архитектур нейронных сетей

Выполнил:
Студент 6 курса 632 группы
отделения математики
механико-математического факультета
Карасев Алексей Александрович

Научный руководитель:
доктор физико-математических наук,
профессор
Локуциевский Лев Вячеславович

Москва, 2018

# Введение

В наш информационный век все большее применение в промышленности находит технология искусственного интеллекта, а именно, такой ее раздел как машинное обучение. С практической точки зрения машинное обучение заключается в создании алгоритма для решения поставленной задачи, используя опыт от решения задач, схожих по своей природе поставленной.

В данной работе рассматриваются алгоритмы машинного обучения, в основе которых лежат модели искусственных нейронных сетей и обучающие методы обеспечения их корректного функционирования. Определяется сущность искусственных нейронных сетей и процесс их обучения с помощью метода обратного распространения ошибки, основанного на методе градиентного спуска. Изучается вопрос о нахождении оптимальной архитектуры нейронных сетей на множестве архитектур, состоящих из одинакового числа нейронов, с некоторыми ограничивающими условиями.

В качестве приложения исследуется вопрос о численном моделировании в двух задачах нахождения оптимальной архитектуры нейронных сетей, на множестве архитектур, состоящих из двух и из трех промежуточных нейронов, с ограничивающими условиями, аппроксимирующих функции из пространства многочленов степени не больших 5, с коэффициентами, принадлежащими отрезку $[-1; 1]$, в метрике $L_2$.

Цель данной работы - выяснить насколько критично с точки зрения точности выдаваемых нейронной сетью результатов предсказаний использовать архитектуры нейронных сетей с неполным набором связей, для уменьшения вычислительной сложности алгоритма обучения нейронной сети.

# Модель искусственной нейронной сети

Искусственные нейронные сети (нейросети) являются математическими моделями сетей нервных клеток живых организмов. Нейросеть представляет собой систему соединенных и взаимодействующих друг с другом искусственных нейронов: узлов, которые, по сути, являются простыми процессорами.

На рис.1 рассмотрен общий вид искусственного нейрона, где:

1) Нейроны, выходные сигналы которых поступают на вход данному;
2) Веса входных сигналов;
3) Коэффициент смещения;
4) Сумматор входных сигналов и коэффициента смещения;
5) Вычислитель функции активации;
6) Выходной сигнал нейрона.

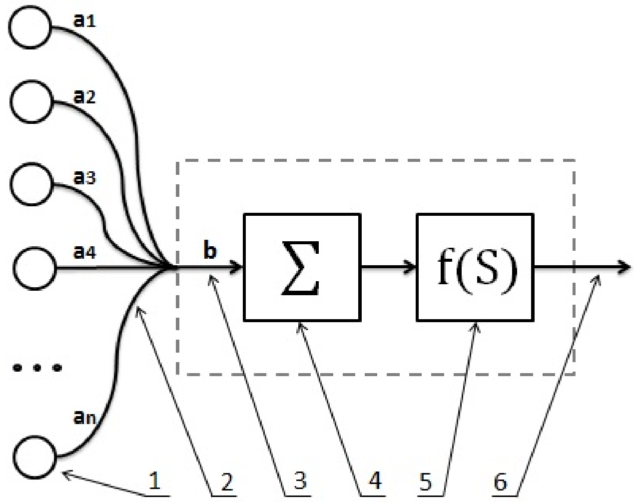



Рис. 1: Вид искусственного нейрона.

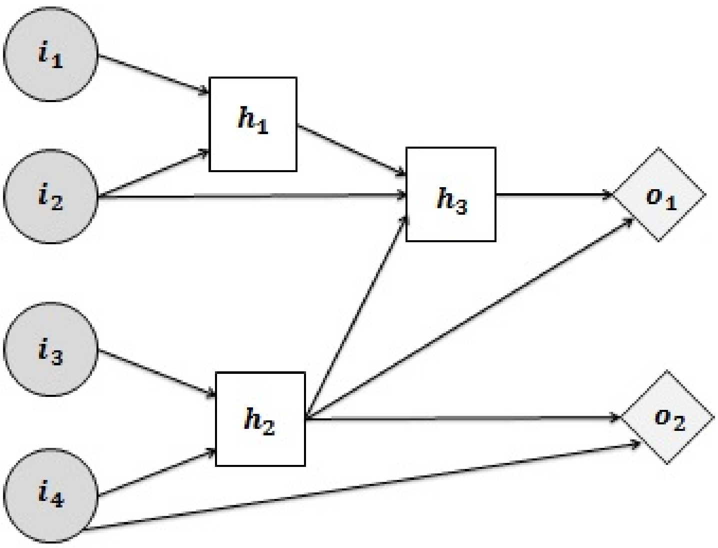
В данной работе в качестве топологической структуры (архитектуры) нейросети используется модель в виде ориентированного графа без циклов и петель, в котором вершинами являются искусственные нейроны, ребрами – синапсы, связывающие их в единую систему.

Нейроны бывают трех типов:

- Нейроны входа – принимают исходные данные (круг, $i$);
- Промежуточные нейроны – выполняют вычислительные операции: сумматор, функция активации (прямоугольник, $h$);
- Нейроны выхода – выполняют вычислительные операции: сумматор; представляют собой выходы сети (ромб, $o$).

В скобках указано, каким образом в данной работе обозначается нейрон, принадлежащий тому или иному типу, в виде вершины графа.

Рассмотрим пример топологической структуры нейросети:



Введем обозначения, используемые в данной работе:

$N$ - граф (нейросеть);

$A = \{a_1, ..., a_q\}$ - множество ребер (синапсов);

$I = \{i_1, ..., i_k\}$ - множество нейронов входа;

$H = \{h_1, ..., h_p\}$ - множество промежуточных нейронов;

$O = \{o_1, ..., o_m\}$ - множество нейронов выхода;

$V = I \sqcup H \sqcup O = \{v_1, ..., v_s\}$ - множество вершин (нейронов), где $s = k + p + m$;

$B = \{b_1, ..., b_{p+m}\}$ - множество коэффициентов смещения;

Вершина $v \in V$ - вычислима, если в нее входят ребра точно из вычислимых ребер. Считаем, что $v \in I$ вычислима.

Граф $N$ - вычислим, если все его вершины вычислимы.

Если граф $N$ - вычислим, то множество его вершин $V$ разбивается на множество слоев $L_t \subset V$, $t = 0, ..., n + 1$:

1) $L_0 = I$;

2) слои $L_t$, $t = 1, ..., n$ итеративно образуются промежуточными нейронами множества $H$, таким образом, что слой $L_t$, $t = 1, ..., n$ состоит из вершин, в которые ведут ребра только из слоев с номерами строго меньше $t$;

3) $L_{n+1} = O$.

**Утверждение:**

$$V = \bigsqcup_{t=0}^{n+1} L_t$$

Считаем, что:

1) $\forall$ вершине $v \in H$ соответствует ее функция активации $F_v : R \to R$;

2) $\forall$ ребру $a \in A$ соответствует его вес $c_a \in R$;

3) $\forall$ коэффициенту смещения $b \in B$ соответствует его вес $c_b \in R$;

4) $\forall$ ребра $a \in A$ $\exists!$ вершины $l(a), r(a) \in V$ - левый и правый конец ребра;

Таким образом, $\forall$ ребру $a \in A$ ставится в соответствие пара вершин $(l(a), r(a))$, обозначим как $l(a) \to r(a)$.

$\forall$ вершины $v \in V$ $N_v$ - минимальная вычислимая подсеть графа $N$, содержащая $v$.

$\forall$ вычислимому графу $N$ ставим в соответствие вектор-функцию $N(x, c) : R^k \times R^{q+p+m} \to R^m$, порожденную его топологической структурой.

$$N(x, c) = \begin{pmatrix} N_{o_1}(x, c) \\ \cdots \\ N_{o_m}(x, c) \end{pmatrix}, \quad N_{o_t}(x, c) = \sum_{v \to o_t} c_{v \to o_t} N_v(x, c) + c_{b_{o_t}}$$

Как видим, функция $N(x, c)$ зависит от вектора переменных $x = (x_1, ..., x_k)^T$

и вектора параметров $c = (c_{a_1}, ..., c_{a_q}, c_{b_1}, ..., c_{b_{p+m}})^T$.

Рассмотрим явное вычисление частных производных функций $N_{o_t}(x, c)$ по координатам вектора параметров $c$:

$$\frac{\partial N_{o_t}(x,c)}{\partial c_{a_j}} = \begin{cases} \sum c_{v \to o_t} \frac{\partial N_v(x,c)}{\partial c_{a_j}} + \begin{cases} N_{l(a)}(x,c), & \text{если } r(a) = o_t \\ 0, & \text{если } r(a) \neq o_t \end{cases} \\ 0, \text{если } a_j \notin N_{o_t} \end{cases}$$

$$\frac{\partial N_{o_t}(x,c)}{\partial c_{b_j}} = \begin{cases} \sum c_{v \to o_t} \frac{\partial N_v(x,c)}{\partial c_{b_j}} + \begin{cases} 1, & \text{если } b_j = b_{o_t} \\ 0, & \text{если } b_j \neq b_{o_t} \end{cases} \\ 0, \text{если } b_j \notin N_{o_t} \end{cases}$$

# Обучение нейронных сетей

С математической точки зрения, постановка задачи обучения нейронных сетей выглядит как построение функции, задаваемой фиксированной архитектурой нейронной сети, подобранными весами синапсов и коэффициентами смещения нейронов, которая аппроксимирует функцию/класс функций из заданного пространства. Таким образом, поставленная задача сводится к решению многопараметрической задачи нелинейной оптимизации по весам синапсов и коэффициентами смещения нейронов.

$$\max_{\varphi \in \mathcal{F}} \min_{c \in \mathcal{C}} ||N(x,c) - \varphi(x)||$$

# Численные методы для обучения нейронных сетей

В данной работе в качестве численного обучения нейронных сетей используется метод обратного распространения ошибки, основанный на градиентном спуске. Градиентный спуск опирается на факт математического анализа о том, что антиградиент функции в точке указывает направление наиболее быстрого убывания функции в данной точке.

В качестве исходных данных для обучения выступает обучающая выборка – $d$-мерный вектор пар $[(X, Y = f(X))]^T = [(X_1^T, Y_1^T), ..., (X_d^T, Y_d^T)]^T$, где $X_j = (x_1, ..., x_k)^T$ – $k$-мерный вектор переменных (признаков),
$Y_j = (y_1, ..., y_k)^T$ – $k$-мерный вектор значений фюункции $\varphi(X)$, которая аппроксимируется искомой нейросетью.

Обучение нейросети происходит по следующему алгоритму:

0) Фиксируется архитектура сети;

1) Случайным образом задаются начальные значения весов синапсов и коэффициентов смещения нейронов. Таким образом, указывается начальная нейросеть $N(X, c_0)$;

2) Выполняется эпоха - цикл по $i$ от 1 до $d$ (длина выборки):

Прямое распространение:

- на вход сети подается вектор признаков $X_i$;
- вычисляется значение нейросети $N(X_i, c_j)$, где $c_j$ - вектор значений весов синапсов и коэффициентов смещения нейронов в данный момент;

Обратное распространение:

- вычисляется антиградиент функции ошибки в точке $(X_i, c_j)$:
$$g(X_i, c_j) = -grad(||N(X, c) - \varphi(X)||)(X_i, c_j);$$
- вычисляются новые значения весов синапсов и коэффициентов смещения нейронов: $c_{j+1} = c_j + \alpha * g$, где $\alpha$ – параметр шага градиентного спуска (чаще всего наперед заданное значение);

3) Находится ошибка нейросети с полученными после выполнения эпохи значениями вектора $c = c_{j+1}: \quad \gamma(X, c_{j+1}) = ||N(X, c_{j+1}) - \varphi(X)||$;

4) Последовательно выполняется несколько эпох, пока значение ошибки $\gamma(X, c_{j+1})$ не станет меньше наперед заданного допустимого значения $\varepsilon$.

# Оптимальной архитектура нейронных сетей

В данной работе также рассматривается задача нахождения оптимальной архитектуры нейронной сети на некотором множестве архитектур. Таким образом, рассматриваемая задача имеет вид:

$$\min_{N \in \mathcal{N}} \max_{\varphi \in \mathcal{F}} \min_{c \in \mathcal{C}} ||N(x, c) - \varphi(x)||$$

# Приложение к практике

В качестве приложения рассматривается вопрос о численном моделировании в задаче нахождения оптимальной архитектуры нейронной сети на множестве архитектур, состоящих из одного нейрона входа и одного нейрона выхода, а также a) из двух; b) из трех промежуточных нейронов, с ограничивающими условиями, аппроксимирующих функции из пространства многочленов степени не больших 5, с коэффициентами, принадлежащими отрезку $[-1; 1]$, в метрике $L_2$. В данной работе в качестве ограничений на множество архитектур $\mathcal{N}$ используется запрет на наличие синапсов, которые связывают нейроны входа с нейронами всех слоев, кроме первого; а так же нейроны выхода с нейронами всех слоев, кроме $n$-го.

Таким образом получаем задачи вида:

$$\min_{N\in\mathcal{N}} \max_{\varphi\in M_5^{[-1;1]}} \min_{c\in R} \int_0^1 |N(x,c)-\varphi(x)|^2\,dx \qquad (1)$$

Для численного моделирования данной задачи нахождения оптимальной архитектуры нейронной сети и их численного обучения используем машинный код, написанный на языке программирования **Python**, и его компиляцию на ЭВМ.

Для написания кода задачи численного моделирования задач вида (1) на языке программирования **Python** используем следующий алгоритм:

1) Генерируем набор значений $X_{test}$, каждый элемент которого принадлежащит $[0,1]$, для валидации всех моделей;

2) Генерируем случайную функцию $\varphi(X)$, принадлежащую множеству $M_5^{[-1;1]}$;

3) Инициализируем случайную обучающую выборку $(X_{train}, Y_{train} = \varphi(X_{train}))$;

4) Используем цикл по множеству $\mathcal{N}$, перебирая, таким образом, все архитекруры, принадлежащие данному множеству:

Обучаем нейросеть, соответствующую каждой архитектуре;

5) Среди обученных нейросетей, соответствующих каждой архитектуре, находим нейросеть, имеющую наименьшее значение метрики ошибки на валидирующей выборке $X_{test}$.

Считаем архитектуру, соответствующую данной нейросети, оптимальной на множестве архитектур $\mathcal{N}$.

Далее рассмотрим все архитектуры нейросетей, соответствующие множествам $\mathcal{N}$, с ограничениями, заявленными выше, для случаев a) с двумя; b) с тремя промежуточными нейронами. А также, архитектуры нейросетей, содержащие полный набор связей.

В данной работе индексы архитектур нейросетей $N_{ij}$, для удобства использования, пронумерованы следующим образом:

$i$ - количество промежуточных нейронов в архитектуре;

$j$ - номер архитектуры на множестве архитектур, содержащих одинаковое число промежуточных нейроов.

Индексы весов синапсов $a_{ijk}$ и коэффициентов смещения $b_{ijk}$, для удобства использования, пронумерованы следующим образом:

$i$ - номер соответствующей архитекруры нейросети;

$j$ - номер слоя, которому принадлежит $r(a_{ijk})$;

$k$ - номер нейрона соответствующего слоя.

а) Два промежуточных нейрона:

1) $N_{21}(X,c) = a_{121}f(a_{111}X + b_{111}) + a_{122}f(a_{112}X + b_{112}) + b_{121}$

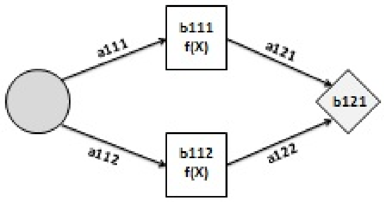



Рис. 2: 2 промежуточных нейрона (Архитектура 1).

2) $N_{22}(X,c) = a_{231}f(a_{221}f(a_{211}X + b_{211}) + b_{221}) + b_{231}$


Рис. 3: 2 промежуточных нейрона (Архитектура 2).

3) $N_{23}(X,c) = a_{333}f(a_{322}f(a_{311}X + b_{311}) + a_{321}X + b_{321}) +$
$+ a_{332}f(a_{311}X + b_{311}) + a_{331}X + b_{331}$

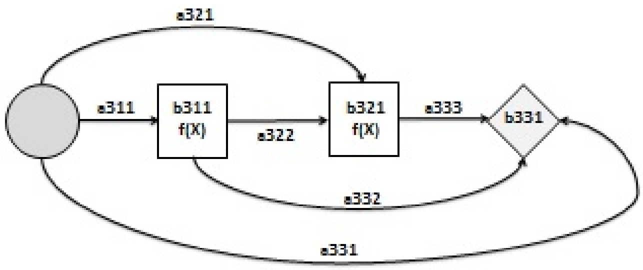



Рис. 4: 2 промежуточных нейрона (Архитектура 3 (Полный набор связей)).

b) Три промежуточных нейрона:

1) $N_{31}(X, c) = a_{131} f(a_{121} f(a_{111} X + b_{111}) + b_{121}) +$
$+ a_{132} f(a_{122} f(a_{111} X + b_{111}) + b_{122}) + b_{131}$

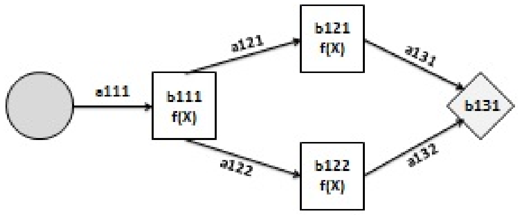



Рис. 5: 3 промежуточных нейрона (Архитектура 1).

2) $N_{32}(X, c) = a_{241} f(a_{231} f(a_{221} f(a_{211} X + b_{211}) + b_{221}) + b_{231}) + b_{241}$

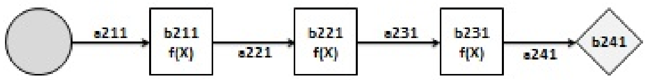



Рис. 6: 3 промежуточных нейрона (Архитектура 2).

3) $N_{33}(X, c) = a_{321} f(a_{311} X + b_{311}) + a_{322} f(a_{312} X + b_{312}) +$
$+ a_{323} f(a_{313} X + b_{313}) + b_{331}$

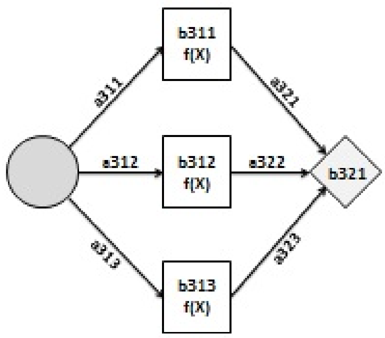



Рис. 7: 3 промежуточных нейрона (Архитектура 3).

4) $N_{34}(X,c) = a_{431}f(a_{421}f(a_{411}X + b_{411}) + a_{422}f(a_{412}X + b_{412}) + b_{421}) + b_{431}$

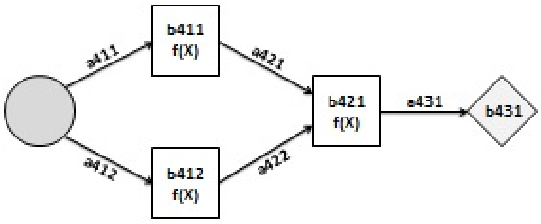



Рис. 8: 3 промежуточных нейрона (Архитектура 4).

5) $N_{35}(X,c) = a_{541}f(a_{531}f(a_{521}f(a_{511}X+b_{511})+b_{521})+a_{532}f(a_{511}X+b_{511})+$
$+ b_{531}) + b_{541}$

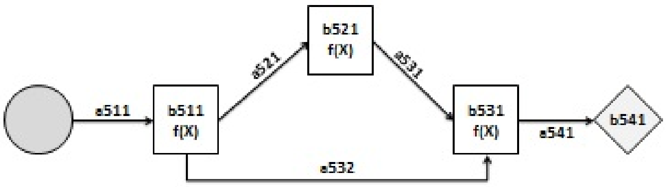



Рис. 9: 3 промежуточных нейрона (Архитектура 5).

6) $N_{36}(X,c) = a_{644}f(a_{633}f(a_{622}f(a_{611}X+b_{611})+a_{621}X+b_{621})+a_{632}f(a_{611}X+$
$+ b_{611}) + a_{631}X + b_{631}) + a_{643}f(a_{622}f(a_{611}X + b_{611}) + a_{621}X + b_{621}) +$
$+ a_{642}f(a_{611}X + b_{611}) + a_{641}X + b_{641}$

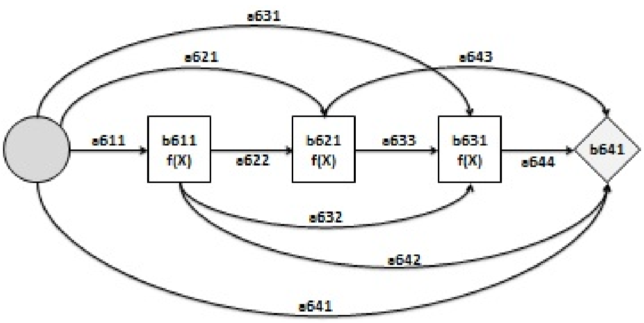



Рис. 10: 3 промежуточных нейрона (Архитектура 6 (Полный набор связей)).

# Вывод.

После компиляции программного кода языка **Python** на ЭВМ получаем следующие значение метрики ошибки на валидирующей выборке:

a) Два промежуточных нейрона:

$N_{21}$ : 0.024 (оптимальная, не считая, полносвязной архитектуры)
$N_{22}$ : 0.038
$N_{23}$ : 0.022

b) Три промежуточных нейрона:

$N_{31}$ : 0.019
$N_{32}$ : 0.183
$N_{33}$ : 0.018
$N_{34}$ : 0.011 (оптимальная, не считая, полносвязной архитектуры)
$N_{35}$ : 0.032
$N_{36}$ : 0.01

Таким образом, из результатов исследования делаем вывод, что **не критично** с точки зрения точности результатов предсказаний, выдаваемых нейронной сетью с двумя или тремя промежуточными нейронами, использовать оптимальные архитектуры нейронных сетей из множества архитектур с ограничивающими условиями, описанными в данной работе, для уменьшения вычислительной сложности алгоритма обучения нейронной сети.

# Список литературы

|1| М.Б. Беркинблит. Нейронные сети. // М.: МИРОС и ВЗМШ РАО, 1993. - 96 с.
|2| А.Н. Горбань. Обучение нейронных сетей. // М. СССР-США СП "Параграф 1990. - 160 с.