МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»
Кафедра математической кибернетики и компьютерных наук

ИССЛЕДОВАНИЕ ПРИМЕНИМОСТИ ФУНКЦИИ SOFTPLUS ДЛЯ РЕШЕНИЯ ЗАДАЧИ ПРИБЛИЖЕНИЯ МНОГОЧЛЕНОВ ИСКУССТВЕННЫМИ НЕЙРОННЫМИ СЕТЯМИ ПРЯМОГО РАСПРОСТРАНЕНИЯ

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студента 4 курса 411 группы
направления 02.03.02 — Фундаментальная информатика и информационные технологии
факультета КНиИТ
Стоколесова Максима Сергеевича

Научный руководитель доцент, к. ф.-м. н.	______________	С. В. Миронов
Заведующий кафедрой к. ф.-м. н.	______________	С. В. Миронов

Саратов 2019

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Актуальность: в настоящее время искусственные нейронные сети приобрели большую популярность благодаря впечатляющим результатам их применения при решении самых разных задач — от приближения простейших зависимостей и классификации объектов до обнаружения и сегментации сложных образов. ИНС активно используются в поисковых системах, обработке видео и изображений, распознавании акустических и визуальных сигналов, анализе различных показаний, симуляторах, автопилотируемых системах и т.д. Архитектуры сетей и подходы к вычислению их параметров стремительно развиваются, а вместе с ними неуклонно растет и область применения ИНС, вытесняя собой стандартные подходы, которые часто оказываются менее эффективными.

Однако в настоящее время процесс определения оптимальной для поставленной задачи архитектуры ИНС нередко носит эмпирический характер, при котором возможности полученной сети могут быть только оценены на основе результатов некоторого набора испытаний, что не всегда удовлетворяет требованиям. Более того, существенное укрепление теоретической основы нейронных сетей в конечном итоге может стать необходимым условием для их дальнейшего развития.

Цель бакалаврской работы — исследование применимости активационной функции `softplus` [1] для решения задачи приближения с заданной точностью многочленов произвольной степени с помощью классической полносвязной трехслойной искусственной нейронной сети прямого распространения. Данная цель определяет **следующие задачи**:

— изучение актуальных источников исследуемого направления;

— формирование необходимой теоретической основы;

— практическое подтверждение доказанных утверждений.

Методологические основы: в качестве основного источника в настоящей работе используется статья *B. Malakooti и Y. Zhou «Approximating polynomial functions by Feedforward Artificial Neural Networks: Capacity analysis and design»* [6], в которой приводятся доказательство теоремы о взаимосвязи между количеством нейронов скрытого слоя ИНС и степенью приближаемого ей многочлена, а также алгоритм построения такой нейронной сети.

Теоретическая значимость бакалаврской работы: в то время как в [6] рассматривалась ИНС с сигмоидальной функцией активации, в настоящей работе представлены результаты исследования применимости для решения той же задачи функции `softplus`, для которой в разделах 1.3 и 1.4 соответственно приводятся аналогичные доказательство справедливости вышеуказанной теоремы и алгоритм построения нейронной сети. Затем в разделе 2 подробно описывается способ реализации самой ИНС и двух алгоритмов вычисления ее весов — итерационного и однопроходного, результаты выполнения которых используются для подтверждения доказанных утверждений на практике.

Структура и объём работы: бакалаврская работа состоит из введения, двух разделов, заключения, списка использованных источников и шести приложений. Общий объем работы — 52 страницы, из них 33 страницы — основное содержание, включая 11 рисунков, бумажный носитель в качестве приложения, список использованных источников информации — 23 наименования.

1 Формирование теоретической основы

1.1 Основные свойства функции softplus

Функция `softplus` имеет следующий вид [1]:

$$\varphi(x) = \ln(1 + e^x).$$

Данная функция является *гладкой*, в отличие от `ReLU` [7], которую она аппроксимирует, что означает существование непрерывной производной `softplus` на всем ее множестве определения, а это, в свою очередь, является очень важной особенностью, учитываемой при проектировании любой ИНС [8, 9]. Сравнение поведения двух функций вблизи начала координат представлено на рис. 1.

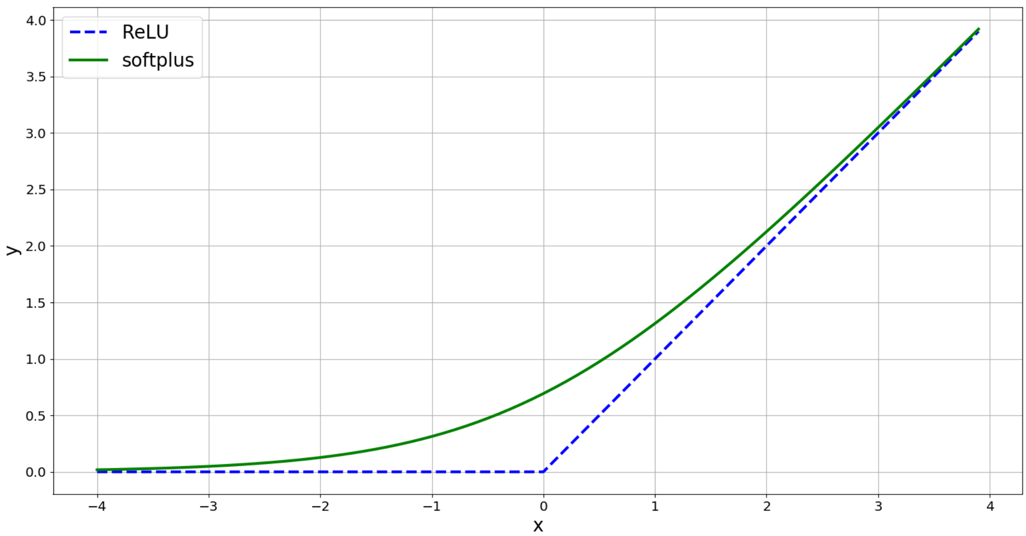

Рисунок 1 – Сравнение графиков функций `ReLU` и `softplus`

1.2 Архитектура рассматриваемой ИНС

Рассмотрим ИНС, представленную на рис. 2. Данная ИНС обладает свойством *полносвязности* [8] и состоит из трех слоев — входного, скрытого и выходного соответственно.

Рассмотрим каждый из слоев более подробно:

— входной слой представлен одним сенсорным [8] нейроном, предназначенным для передачи без изменения получаемого сигнала x во все исходящие связи;

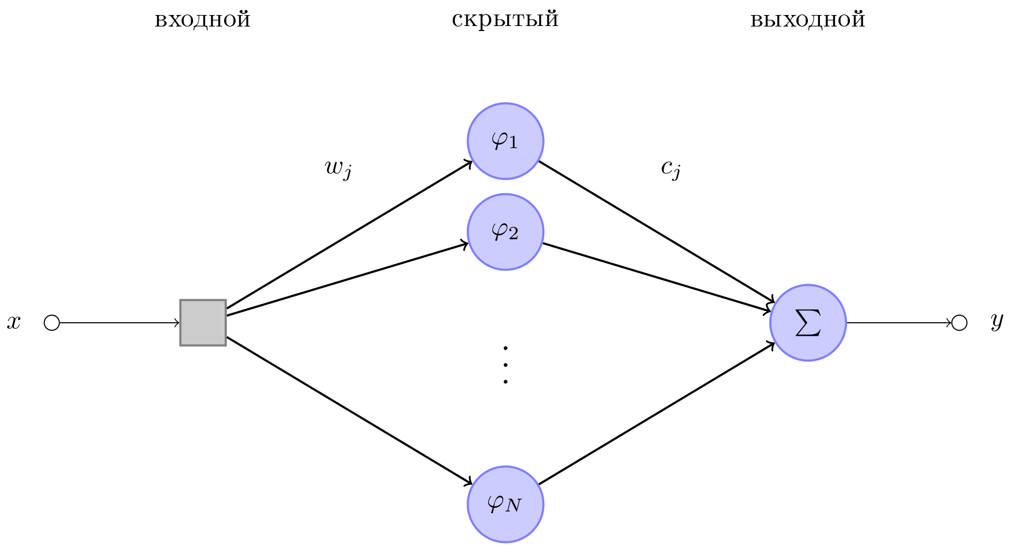

Рисунок 2 – Общий вид рассматриваемой ИНС

— скрытый слой содержит N вычислительных нейронов, комбинированный вход n_j каждого из которых выражается следующим равенством:

$$n_j = n_j(x) = xw_j + \theta_j,$$

где j — номер нейрона, θ_j — его пороговое значение, а w_j — вес связи со входным узлом. Также внутри каждого скрытого нейрона используется активационная функция `softplus` (далее φ_j), формирующая его выходное значение:

$$\varphi_j = \varphi_j(x) = \ln(1 + e^{n_j(x)});$$

— выходной слой состоит из единственного нейрона, не имеющего активационной функции, который производит на выход значение y, представляющее собой ответ ИНС на входной сигнал:

$$y = F(x) = \sum_{j=1}^{N} c_j \varphi_j(x).$$

В настоящей работе ИНС с вышеописанной архитектурой будет использоваться для приближения многочленов, определяемых следующим выражением:

$$f(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_r x^r,$$

где a_i, $i = \overline{0, r}$ — коэффициенты многочлена, а r — его степень.

1.3 Исследование применимости функции softplus

На основе идей из статьи [6], а также некоторых разъяснений из [10], в ходе настоящей работы были доказаны следующие набор лемм и теорема для случая использования функции `softplus` в качестве активационной.

Лемма 1. *Производная порядка s функции $F(x)$ может быть представлена следующим образом:*

$$\frac{d^s F(x)}{dx^s} = \sum_{j=1}^{N} c_j w_j^s Q_s(\psi_{kj}),$$

где

$$k = k(s) = \begin{cases} 1, & s > 0, \\ 0, & s = 0, \end{cases}$$

функция ψ_{kj} имеет вид

$$\psi_{kj} = \psi_{kj}(x) = \begin{cases} \sigma_j(x), & k = 1, \\ \varphi_j(x), & k = 0, \end{cases}$$

где $\sigma_j(x) = \frac{\varphi_j'(x)}{w_j}$, а $Q_s(\psi_{kj})$ выражается равенством

$$Q_s(\psi_{kj}) = \begin{cases} \psi_{kj}(1 - \psi_{kj})\frac{dQ_{s-1}(\psi_{kj})}{d\psi_{kj}}, & \text{для } s = 2, 3, \ldots, \\ \psi_{kj}, & \text{для } s = 0, 1. \end{cases}$$

Примечание. Пусть $\frac{d^s F(x)}{dx^s} = \frac{d^s f(x)}{dx^s}$ при $x = 0$.

Поскольку в точке $x = 0$ справедливо равенство $\frac{d^s f(x)}{dx^s} = s! a_s$, $s = \overline{0, r}$, то мы получаем следующее матричное выражение:

$$\underbrace{\begin{bmatrix} Q_0(\psi_{01}^0) & Q_0(\psi_{02}^0) & \ldots & Q_0(\psi_{0N}^0) \\ Q_1(\psi_{11}^0)w_1 & Q_1(\psi_{12}^0)w_2 & \ldots & Q_1(\psi_{1N}^0)w_N \\ \vdots & \vdots & \ddots & \vdots \\ Q_r(\psi_{11}^0)w_1^r & Q_r(\psi_{12}^0)w_2^r & \ldots & Q_r(\psi_{1N}^0)w_N^r \end{bmatrix}}_{\Omega} \cdot \underbrace{\begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_N \end{bmatrix}}_{c} = \underbrace{\begin{bmatrix} a_0 \\ 1! a_1 \\ \vdots \\ r! a_r \end{bmatrix}}_{a}, \qquad (1)$$

где $\psi^0_{kj} = \psi_{kj}(0)$, $k = \overline{0,1}$, $j = \overline{1,N}$, а само выражение (1) можно представить в следующем виде:

$$\Omega c = a,$$

где Ω — крайняя левая матрица, $c = (c_1, c_2, \ldots, c_N)^T$, $a = (a_0, 1!a_1, \ldots, r!a_r)$.

Лемма 2. *Если $N = r + 1$, то матрица Ω является невырожденной по крайней мере для одного набора ψ^0_{kj}, $k = \overline{0,1}$, $j = \overline{1,N}$, если $w_j \neq w_l$ при $j \neq l$ для $j, l = \overline{1,N}$.*

Лемма 3. *Для каждого параметра c_j из выражения* (1) *справедливо следующее неравенство:*

$$|c_j| \leqslant v_{\max} q_{\max} \sum_{i=0}^{r} |i!a_i| \delta^{-i}, \quad j = \overline{1,N},$$

где $v_{\max}$ — положительное число, зависящее от N и r, $q_{\max}$ — положительное число, зависящее от пороговых значений скрытых узлов, а δ выбрана таким образом, что $w_j = (j-1)\delta$ при $\delta > 0$.

На основе утверждений, полученных в леммах 1–3, была сформулирована и доказана теорема, позволяющая оценить *емкость ИНС*, под которой понимается наибольшая степень многочленов, приближаемых данной нейронной сетью с наперед заданной точностью.

Теорема 1 (О емкости ИНС)**.** *Однослойная ИНС с $N > 0$ скрытыми нейронами может приближать любой многочлен $f(x)$ степени $N-1$ с любой точностью при $|x| < x_{\max}$. Т.е. ИНС может принять такую описывающую ее функцию $F(x)$, что $|F(x) - f(x)| < \varepsilon$, где ε — требуемая верхняя граница ошибки приближения.*

1.4 Алгоритм вычисления весов ИНС

Результаты из приведенных выше утверждений могут быть использованы для построения *алгоритма вычисления весов ИНС*, приближающей *заданный* многочлен.

Алгоритм вычисления весов ИНС принимает на вход следующие параметры:

— список коэффициентов a_i, $i = \overline{0, r}$ приближаемого многочлена, где r — его степень;
— значение требуемой верхней границы ε ошибки приближения;
— максимальная по модулю граница $x_{\max}$ отрезка, на котором осуществляется приближение;
— шаг d поиска подходящего порогового значения нейронов скрытого слоя.

Алгоритм вычисления весов ИНС

1. Согласно теореме 1 задать количество N нейронов скрытого слоя равным $r + 1$.
2. Выбрать пороговое значение θ скрытых нейронов и вычислить значения функции ψ_k^0, $k = \overline{0, 1}$, определенной в лемме 3.
3. Вычислить значения $Q_s(\psi_k^0)$, $s = \overline{0, r}$ из леммы 3. Если хотя бы одно из получившихся значений равно нулю, то взять новое пороговое значение $\theta = \theta + d$ и повторить данный шаг.
4. Вычислить значения $v_{\max}$ и $q_{\max}$, определенные в лемме 3.
5. Вычислить значение M, определенное в теореме 1.
6. Задать значение $\delta = \frac{\varepsilon}{M+1}$ и вычислить веса $w_j = (j - 1)\delta$, $j = \overline{1, N}$.
7. Вычислить веса c_j, $j = \overline{1, N}$, используя уравнение 1, таким образом, $c = \Omega^{-1}a$.

1.5 Итерационный алгоритм обучения ИНС

В настоящее время одним из самых популярных методов обучения ИНС является *алгоритм обратного распространения ошибки* [8], основная идея которого заключается в применении итерационного процесса, на каждом шаге которого производится вычисление *градиента*, используемого для коррекции весов нейронной сети с целью минимизации ошибки ее работы.

В настоящей работе данный метод используется в режиме реализации *стохастического* градиентного спуска, при котором изменение весов производится для *каждого* экземпляра обучающей выборки. Выбор обусловлен тем, что такая реализация, по сравнению со своим *пакетным* вариантом, в общем случае позволяет добиться большей точности. Более подробно с алгоритмом обратного распространения ошибки можно ознакомиться в [8].

2 Реализация ИНС

2.1 Используемые инструменты

Для реализации описанной выше ИНС был выбран язык программирования `Python` версии 3 [15], поскольку он является высокоуровневым, простым и минималистичным, что позволяет сосредоточиться на решении задачи, а не на самом языке и особенностях используемой платформы.

К тому же, для языка `Python` существует множество готовых инструментов и библиотек, позволяющих быстро производить математические операции, а также осуществлять визуализацию имеющихся данных.

Таким образом, в настоящей работе вместе с собственными реализованными классами используются следующие модули:

— `sys` [16], обеспечивающий доступ к некоторым системным переменным и функциям;

— `random` [17], предоставляющий функции для генерации случайных чисел;

— `math` [18], предназначенный для работы с числами и содержащий множество констант;

— библиотека `NumPy` [19], позволяющая эффективно работать с многомерными матрицами;

— библиотека `Matplotlib` [20], с помощью которой можно визуализировать данные в виде графиков.

2.2 Реализованные классы

Вся совокупность операций с ИНС, рассмотренной в разделе 1.2, была реализована в виде системы следующих классов:

— класс `NeuralNetwork` – описывает ИНС, предназначенную для приближения многочленов;

— класс `PolyDataManager` – предназначен для работы с тестовыми и тренировочными наборами данных, а также для визуализации результатов;

— класс `NNTester` – используется для тестирования ИНС;

— класс `NNPolyTeacher` – инкапсулирует итерационный алгоритм из раздела 1.5 и предназначен для управления процессом обучения ИНС, приближающей *некоторый* многочлен, заданный набором дискретных значений;

— класс `NNPolyBuilder` — инкапсулирует однопроходный алгоритм из раздела 1.4 и предназначен для формирования сети, приближающей с указанной точностью *известный* многочлен, заданный своими коэффициентами.

2.3 Запуск обучения и тестирования

Перейдем к обзору результатов работы описанных классов, начнем с `NNPolyTeacher`. Для примера в качестве приближаемого многочлена возьмем функцию $f(x) = 1 - 15x + x^2 + x^3$, график которой представляет собой кубическую параболу. Будем рассматривать эту функцию на отрезке $[-5, 5]$, используя 30 случайно выбранных точек для обучения ИНС и 100 последовательных точек для проверки. Согласно доказанной в разделе 1.3 теореме 1, оптимальным количеством n скрытых нейронов для такой сети будет $n = r + 1$, где r — степень рассматриваемого многочлена.

В результате работы класса `NNPolyTeacher` должен построиться график, схожий с тем, что представлен на рис. 3.

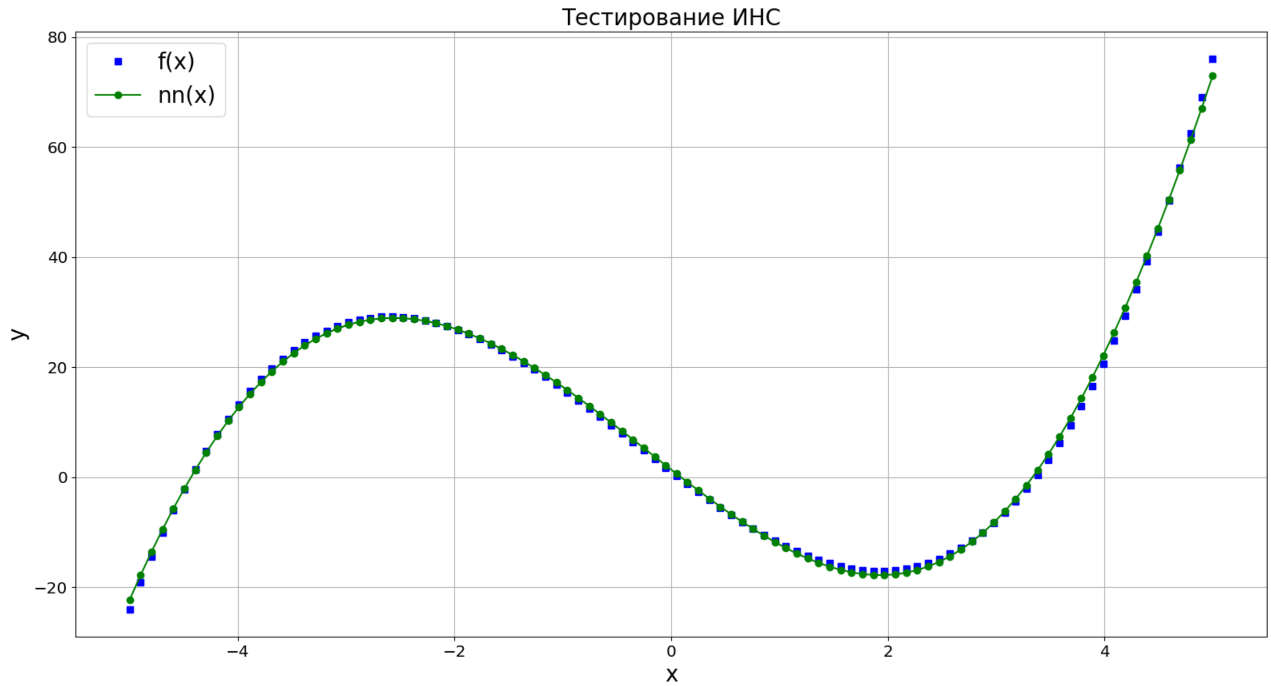

Рисунок 3 – Результат работы ИНС на тестовой выборке после процесса обучения, где $f(x) = 1 - 15x + x^2 + x^3$ — приближаемый многочлен, $nn(x)$ — функция, построенная ИНС. Результаты ошибок:
min ≈ 0.008, max ≈ 3.06, avg ≈ 0.55

Скорость и качество процесса обучения, в зависимости от начальных значений весовых коэффициентов, распределения тренировочных данных, результатов осуществления градиентного спуска, погрешности округлений и прочих влияющих на него факторов, при разных запусках всегда отличаются, однако в общем случае, прежде чем выдать результат, представленный на рис. 3, ИНС от эпохи к эпохе последовательно проходит состояния, отображенные на рис. 4 и 5 соответственно.

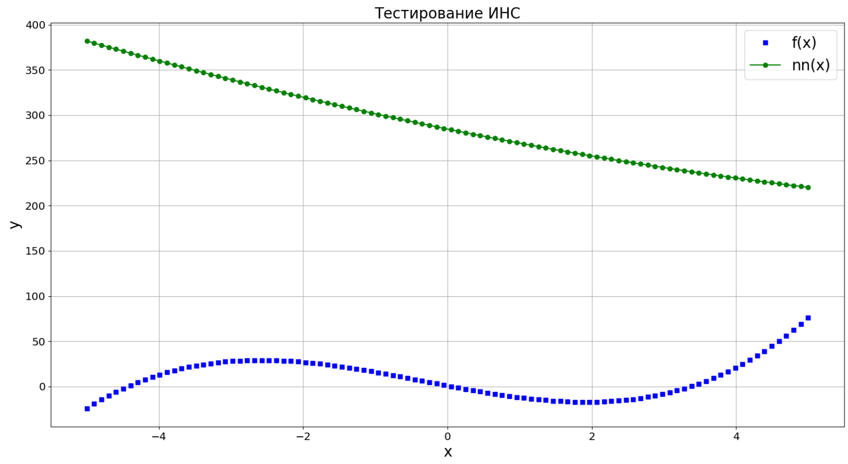

Рисунок 4 – Производительность ИНС на начальных эпохах

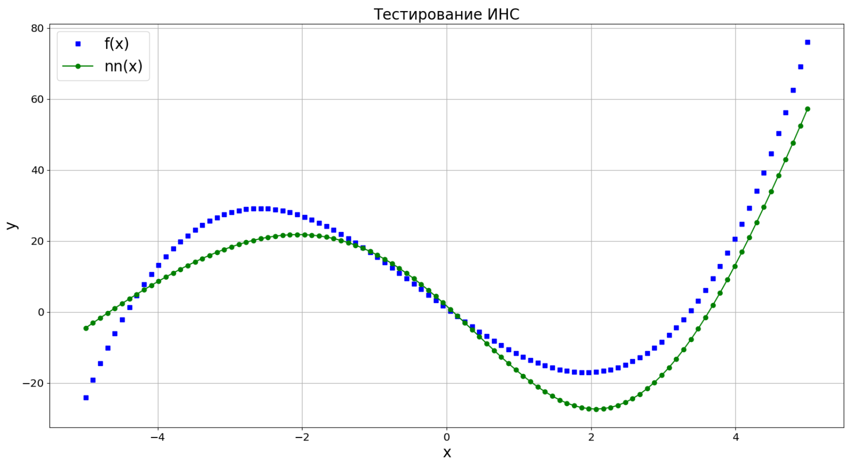

Рисунок 5 – Производительность ИНС на средних эпохах

Таким образом, результаты, представленные на рис. 3–5 демонстрируют жизнеспособность такого подхода к построению ИНС для приближения многочленов.

2.4 Запуск вычисления весов и тестирования

Теперь перейдем к рассмотрению результатов работы класса `NNPolyBuilder`. Как и в разделе 2.3, в качестве приближаемого многочлена будем использовать функцию $f(x) = 1 - 15x + x^2 + x^3$ на отрезке $[-5, 5]$, рассматривая 100 последовательных точек для проверки. В качестве верхней границы ошибки возьмем значение $\varepsilon = 0.05$.

В результате работы класса `NNPolyBuilder` должен построиться график, представленный на рис. 6. В данном случае максимальная ошибка построенной ИНС составляет ≈ 0.01, что удовлетворяет заданному требованию.

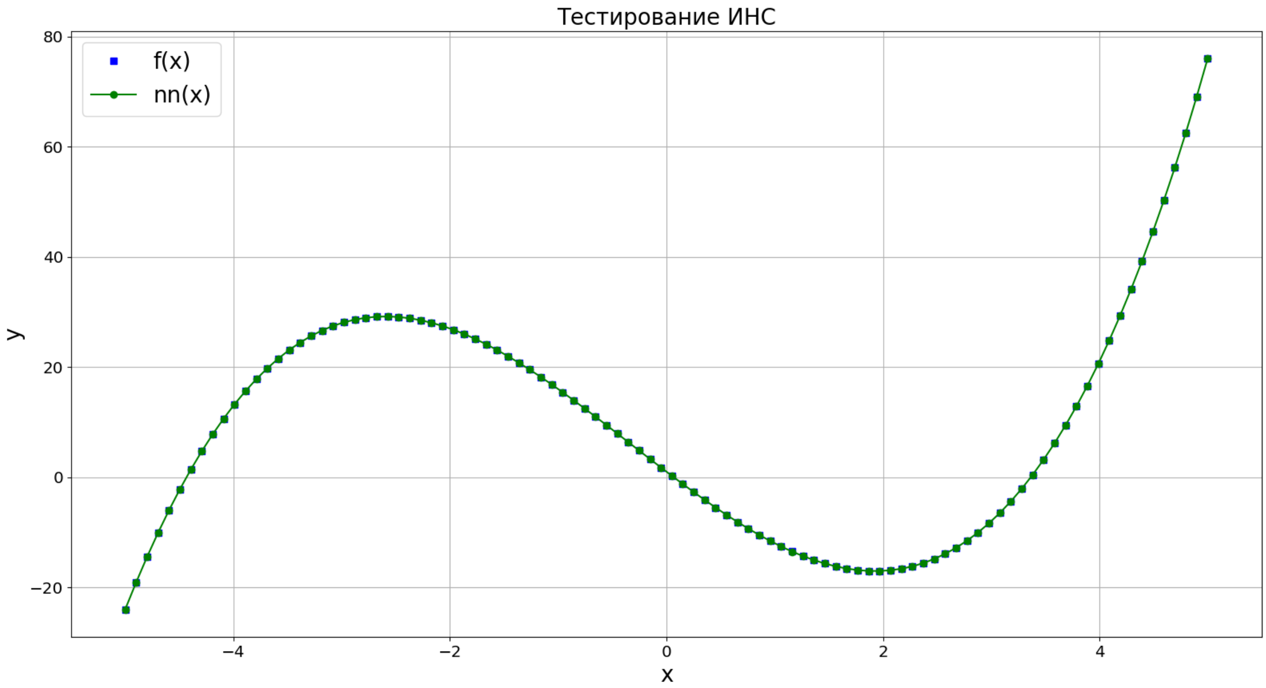

Рисунок 6 – Результат работы ИНС на тестовой выборке после процесса вычисления весов, где $f(x) = 1 - 15x + x^2 + x^3$ — приближаемый многочлен, $nn(x)$ — функция, построенная ИНС. Результаты ошибок:
$\min \approx 8.17e^{-6}$, $\max \approx 0.01$, $\text{avg} \approx 0.003$

Как следует из результатов, представленных на рис. 3 и 6, в случае известных степени и коэффициентов приближаемого многочлена, способ прямого вычисления весов ИНС в общем случае справляется с задачей значительно лучше, чем его итерационный аналог, если в качестве критерия оценки рассматривать затраченное на построение готовой сети время и полученную точность. Однако на практике далеко не всегда известны параметры приближаемой функции, поэтому способ последовательного обучения ИНС своей актуальности ничуть не теряет.

Таким образом, вышеприведенные результаты подтверждают справедливость алгоритма, полученного в разделе 1.4, а также демонстрируют его практическую применимость.

ЗАКЛЮЧЕНИЕ

В ходе настоящей работы была доказана справедливость теоремы о взаимосвязи между количеством нейронов скрытого слоя трехслойной ИНС прямого распространения и степенью приближаемого ей многочлена в случае применения активационной функции `softplus`, а также были продемонстрированы результаты работы реализованных ИНС и двух алгоритмов вычисления ее весов, подтверждающие полученные теоретические выводы на практике.

Таким образом, представленные в данной работе результаты могут послужить основой для дальнейшего исследования активационной функции `softplus`, например, в контексте популярного в последнее время *«глубокого обучения»* [22] многослойных нейронных сетей, при котором существует проблема т.н. *«затухающих градиентов»* (vanishing gradients), наиболее свойственная *сигмоиде*, в связи с чем часто отдается предпочтение `softplus` [23].

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1 *Dugas, C.* Incorporating Second-order Functional Knowledge for Better Option Pricing / C. Dugas, Y. Bengio, F. Belisle, C. Nadeau, R. Garcia. — MIT Press, 2000.

2 *Hahnloser, R. H. R.* Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit / R. H. R. Hahnloser, R. Sarpeshkar, M. A. Mahowald, R. J. Douglas, H. S. Seung. — Nature, 2000.

3 *Hahnloser, R. H. R.* Permitted and Forbidden Sets in Symmetric Threshold-Linear Networks / R. H. R. Hahnloser, H. S. Seung. — Neural Computation, 2003.

4 *Cybenko, G.* Approximation by superpositions of a sigmoidal function / G. Cybenko. — Springer London, 1989.

5 *Namig, G.* On the approximation by single hidden layer feedforward neural networks with fixed weights / G. Namig, V. Ismailov. — Elsevier, 2018.

6 *Malakooti, B.* Approximating polynomial functions by Feedforward Artificial Neural Networks: Capacity analysis and design / B. Malakooti, Y. Zhou. — Applied Mathematics and Computation, 1998.

7 *Glorot, X.* Deep Sparse Rectifier Neural Networks / X. Glorot, A. Bordes, Y. Bengio. — PMLR, 2011.

8 *Haykin, S.* Neural Networks: A Comprehensive Foundation / S. Haykin. — Prentice Hall, 1999.

9 *Галушкин, А. И.* Нейронные сети: основы теории / А. И. Галушкин. — Москва: Горячая Линия - Телеком, 2012.

10 *Узенцова, Н. С.* Об одновременном приближении алгебраических многочленов и их производных нейронными сетями прямого распространения сигнала с одним скрытым слоем / Н. С. Узенцова, С. П. Сидоров. — Известия Саратовского университета. Новая серия. Серия Математика. Механика. Информатика, 2013.

11 *Han, J.* The Influence of the Sigmoid Function Parameters on the Speed of Backpropagation Learning / J. Han, C. Moraga. — From Natural to Artificial Neural Computation, 1995.

12 *Курош, А. Г.* Курс высшей алгебры / А. Г. Курош. — Москва: Наука, 1968.

13 *Фихтенгольц, Г. М.* Основы математического анализа / Г. М. Фихтенгольц. — Москва: Наука, 1968.

14 *Хайкин, С.* Нейронные сети: Полный курс / С. Хайкин. — Вильямс, 2006.

15 Python 3.7.3 documentation [Электронный ресурс]. — URL: `https://docs.python.org/3/` (Дата обращения 15.05.2019). Загл. с экр. Яз. англ.

16 System-specific parameters and functions [Электронный ресурс]. — URL: `https://docs.python.org/3/library/sys.html` (Дата обращения 15.05.2019). Загл. с экр. Яз. англ.

17 Generate pseudo-random numbers [Электронный ресурс]. — URL: `https://docs.python.org/3/library/random.html` (Дата обращения 15.05.2019). Загл. с экр. Яз. англ.

18 Mathematical functions [Электронный ресурс]. — URL: `https://docs.python.org/3/library/math.html` (Дата обращения 15.05.2019). Загл. с экр. Яз. англ.

19 Numpy and Scipy Documentation [Электронный ресурс]. — URL: `https://docs.scipy.org/doc/` (Дата обращения 15.05.2019). Загл. с экр. Яз. англ.

20 Matplotlib documentation [Электронный ресурс]. — URL: `https://matplotlib.org/` (Дата обращения 15.05.2019). Загл. с экр. Яз. англ.

21 *Rashid, T.* Make Your Own Neural Network / T. Rashid. — 2016.

22 *Goodfellow, I.* Deep Learning / I. Goodfellow, Y. Bengio, A. Courville. — The MIT Press, 2016.

23 *Zheng, H.* Improving deep neural networks using softplus units / H. Zheng, Z. Yang, W. Liu, J. Liang, Y. Li. — 2015 International Joint Conference on Neural Networks (IJCNN), 2015.