

6. *McEvoy K.* Jumps of quasiminimal enumeration degrees // *J. Symbolic Logic.* 1985. Vol. 50, № 3. P. 839–848.
7. *Rogers H., Jr.* Theory of recursive functions and effective computability. New York: McGraw-Hill, 1967. 482 p.
8. *Solon B. Ya.* Co-total enumeration degrees // *Beckmann A., Berger U., Löwe B., Tucker J. V.* (eds). Second conference on computability in Europe, CiE 2006. Proceedings. Berlin, Heidelberg : Springer, 2006. P. 538–545. (Lecture Notes in Computer Science, Vol. 3988).

УДК 519.67

С. И. Хашин

СРАВНЕНИЕ АКТИВАТОРНЫХ ФУНКЦИЙ НЕЙРОСЕТИ

Сравнивается эффективность различных функций активации нейронной сети. Рассматривается 3-слойная нейросеть с двумя входами для задачи регрессии: по 10 нейронов в первом и втором слоях. В 3-м, выходном, слое всегда 1 нейрон. Для всех нейронов в каждом внутреннем слое выбирается одна и та же функция активации. Мы рассматриваем по 9 различных активаторных функций в каждом слое, всего 81 вариант. Для примера рассматриваются три различные обучающие матрицы (4096 обучающих векторов в одной, 255 025 в другой и 70 000 в третьей). Сравниваются результаты обучения для разных пар активаторных функций в каждом слое.

Ключевые слова: нейрон, нейросеть, передаточная функция, активаторная функция.

S. I. Khashin

A COMPARISON OF THE ACTIVATION FUNCTIONS OF THE NEURAL NETWORK

We consider a 3-layer neural network with two inputs for the regression problem: 10 neurons in the first and the second layers. The 3rd, output, layer always contains 1 neuron. For all neurons in each inner layer, the same activation function is selected. We consider 9 different activator functions in each layer, 81 options in total. As an example, three different training matrices are considered (4096 training vectors in the first, 255 025 in the second, and 70 000 in the third). The paper compares the effectiveness of various selections of activator functions.

Key words: neuron, neural network, activation function

1. Введение

Нейронные сети в последнее время активно примеряются во многих разделах компьютерной графики [1–4]. К сожалению, большинство результатов в этой области носят экспериментальный характер, выбор той или иной архитектуры нейросети, выбор активаторных функций основывается на интуиции разработчика. Обучение нейронной сети требует боль-

ших вычислительных ресурсов, и разработчики обычно работают с одной-двумя wybranными архитектурами нейронных сетей, не пытаясь исследовать все возможности.

В литературе не так часто встречаются прямые сравнения эффективности различных функций активации. Обычно (см., например, [6, 7]) анализируется не эффективность функций, а их популярность среди разработчиков нейросетей. В настоящей работе мы пытаемся на трех простейших примерах из компьютерной графики сравнить эффективность одной и той же архитектуры нейронной сети, изменяя лишь активаторную функцию. Все рассмотренные алгоритмы были реализованы на C++ (Visual Studio, [5]).

2. Функции активации и архитектура

Обычно в теории нейронных сетей ограничиваются сравнительно небольшим набором активаторных функций. Многие из них являются взаимозаменяемыми.

Теорема. Пусть функции $f(x)$ и $g(x)$ связаны соотношением

$$f(x) = C_1 + C_2 g(C_3 x + C_4)$$

для некоторых констант C_i ($1 \leq i \leq 4$). Тогда функции активации $f(x)$ и $g(x)$ эквивалентны. Более точно, если в обученной нейросети некоторый нейрон имеет функцию активации $f(x)$, то можно заменить ее на $g(x)$ и скорректировать некоторые коэффициенты нейронов так, что нейросеть будет давать в точности те же результаты, что и исходная.

Доказательство. Для перехода от функции $g(x)$ к функции $h(x) = g(C_3 x + C_4)$ достаточно поделить все коэффициенты нейрона на C_3 и вычесть из свободного члена C_4 .

Для перехода от функции $h(x)$ к функции $g(x) = C_1 + C_2 h(x)$ надо коэффициенты всех нейронов, зависящих от текущего, поделить на C_2 и вычесть из свободного члена C_1 .

Таким образом, функции активации можно считать принимающими значения от -1 до 1 и равными 0 в точке 0 . В нашем исследовании мы приняли 9 различных активаторных функций $f_i(x)$ ($1 \leq i \leq 9$). В основном, они являются антисимметричными, кроме первых трех:

- $f_0(x) = x$
- $f_1(x) = (x < 0 ? 0 : x)$ (ReLU)
- $f_2(x) = 1/(1 + \exp(-x))$ (сигмоид)
- $f_3(x) = (x < 0 ? -1 : 1)$ (sign)
- $f_4(x) = 2 \arctg(x)/\pi$
- $f_5(x) = x/(1 + |x|)$ (softSign)
- $f_6(x) = \text{sign}(x) * x^2/(1 + x^2)$ (softSign2)
- $f_7(x) = \text{th}(x)$ (тангенс гиперболический)

Функция f_8 — кусочно-линейная, задана правилом:

$$f_8(x) = \begin{cases} x, & |x| \leq 1, \\ \text{sign}(x), & |x| > 1. \end{cases}$$

И, наконец, f_9 — нечетная функция, для положительных x определена следующим образом:

$$f_9(x) = \begin{cases} x/2, & x \leq 1, \\ (x+1)/4, & 1 < x < 3, \\ 1, & 3 \geq x. \end{cases}$$

Для каждого испытания была взята трехслойная нейронная сеть с двумя входными параметрами, содержащая по 10 нейронов в первом и втором слоях и один нейрон в выходном слое. Все нейроны одного слоя имеют одну и ту же функцию активации от f_1 до f_9 , а выходной нейрон всегда имеет тривиальную функцию активации (f_0). Такая архитектура нейронной сети будет обозначаться $LLkm.arch$, если в первом слое функция активации f_k , а во втором — f_m . Например, если в первом слое функция активации f_4 , а во втором — f_7 , архитектура обозначается $LL47.arch$. Таким образом, всего у нас получается 81 вариант архитектуры, различающийся лишь активаторными функциями: $LL11.arch, \dots, LL99.arch$.

3. Обучающие матрицы

Для примера рассмотрим три типичные задачи из компьютерной графики:

- 1) аппроксимацию яркости изображения в точке через яркости изображения в предыдущих точках,
- 2) функцию, задающую яркость текущей точки изображения в зависимости от ее координат (x, y) .
- 3) стандартную задачу «MNIST» (см., например, [4]).

В первой задаче мы аппроксимируем яркость точки $R(x, y)$ через яркости трех предыдущих точек $R(x-1, y)$, $R(x, y-1)$, $R(x-1, y-1)$:

$$R(x, y) \approx R(x-1, y) + F(R(x-1, y-1), R(x, y-1)).$$

Всего в этой задаче было взято 255 026 точек (исходное изображение имеет размер 512×512).

Во второй задаче мы строим нейросеть, аппроксимирующую яркость точки изображения в зависимости от координат (x, y) точки. Так как яркость сравнительно плавно зависит от координат, размер изображения был уменьшен до $64 \times 64 = 4096$ точек.

Третья задача — это стандартная задача MNIST, содержащая 70 000 обучающих элементов. В этой задаче для каждого изображения размера 28×28 мы берем по 54 коэффициента Фурье.

При фиксированных обучающих множествах была рассмотрена 81 различная архитектура, различающаяся функцией активации в первом и втором слоях (от 1 до 9 в каждом). Функция активации выходного слоя всегда предполагалась тривиальной: $f(x) = x$.

4. Результаты

Для каждой из 81 взятой архитектуры было проведено по несколько сотен испытаний, то есть минимизаций целевой функции, начиная с различных начальных случайных параметров. В следующих таблицах при-

ведено по 12 наилучших результатов, полученных в каждой из трех рассмотренных задач.

Например, файл *LL61_33.for* означает активаторную функцию номер 6 в первом слое (т. е. $f_6(x) = \text{sign}(x) * x^2 / (1 + x^2)$) и функцию номер 1 (ReLU) во втором. Индекс 33 означает просто номер попытки. Аналогичные обозначения используются и в двух других таблицах.

Таблица 1

Результаты испытаний для задачи 1

<i>outFile</i>	<i>S</i>	<i>outFile</i>	<i>S</i>
<i>LL11_39.for</i>	0.00447444	<i>LL61_39.for</i>	0.004482715
<i>LL61_31.for</i>	0.00447937	<i>LL61_33.for</i>	0.004483161
<i>LL61_35.for</i>	0.00448081	<i>LL16_39.for</i>	0.004483752
<i>LL61_36.for</i>	0.00448136	<i>LL61_32.for</i>	0.004485782
<i>LL61_38.for</i>	0.00448144	<i>LL11_31.for</i>	0.004486463
<i>LL11_36.for</i>	0.00448216	<i>LL16_32.for</i>	0.004487027

Таблица 2

Результаты испытаний для задачи 2

<i>outFile</i>	<i>S</i>	<i>outFile</i>	<i>S</i>
<i>LL15_25.for</i>	0.00913053	<i>LL15_26.for</i>	0.00962957
<i>LL12_25.for</i>	0.00916242	<i>LL15_21.for</i>	0.00967867
<i>LL15_09.for</i>	0.00919930	<i>LL12_26.for</i>	0.00984016
<i>LL15_05.for</i>	0.00920503	<i>LL16_27.for</i>	0.00987700
<i>LL65_23.for</i>	0.00949088	<i>LL16_23.for</i>	0.01004778
<i>LL15_24.for</i>	0.00954694	<i>LL22_25.for</i>	0.01014372

Таблица 3

Результаты испытаний для задачи 3

<i>outFile</i>	<i>S</i>	<i>outFile</i>	<i>S</i>
<i>mn11_13.for</i>	0.26778785	<i>mn28_34.for</i>	0.2796177
<i>mn16_34.for</i>	0.27004601	<i>mn99_22.for</i>	0.2813517
<i>mn61_33.for</i>	0.27215121	<i>mn16_44.for</i>	0.2831303
<i>mn26_21.for</i>	0.27637881	<i>mn16_24.for</i>	0.2836603
<i>mn61_12.for</i>	0.27723661	<i>mn21_44.for</i>	0.2837765
<i>mn16_11.for</i>	0.27770696	<i>mn61_32.for</i>	0.2848535

5. Основные выводы

Во всех рассмотренных случаях в одном из слоев оказывается более эффективна функция активации ReLU. При этом функция активации в другом слое может быть различна: softSign или softSign2 ($\text{sign}(x) * x^2 / (1 + x^2)$). Для первой задачи самый лучший результат получен в случае, когда у обоих внутренних (скрытых) слоев функция активации — ReLU. Все остальные пары функций активации показали худшие результаты.

Библиографический список

1. *Галушкин А. И.* Нейрокомпьютеры : учеб. пособие. М.: Альянс, 2014. 528 с.
2. *Гафаров Ф. М., Галимянов А. Ф.* Искусственные нейронные сети и их приложения. Казань: Изд-во Казан. ун-та, 2018. 121 с.
3. *Головко В. А., Краснопрошин В. В.* Нейросетевые технологии обработки данных : учеб. пособие. Минск: БГУ, 2017. 260 с.
4. *Николенко С., Кадурын А., Архангельская Е.* Глубокое обучение. СПб.: Питер, 2018. 480 с.
5. Центр загрузки Microsoft. URL: <https://www.microsoft.com/ru-ru/download> (дата обращения: 20.02.2020).
6. *Nwankpa C., Ijomah W., Gachagan A., Marshall S.* Activation functions: comparison of trends in practice and research for deep learning // arXiv: 1811.03378 [cs.LG]. URL: <https://arxiv.org/abs/1811.03378> (дата обращения: 20.02.2020).
7. *Ramachandran P., Zoph B., Le Q. V.* Searching for activation functions // arXiv: 1710.05941 [cs.NE]. URL: <https://arxiv.org/abs/1710.05941> (дата обращения: 20.02.2020).