УДК

ФУНКЦИИ АКТИВАЦИИ

Демещенко М.В. студент группы 253503, Марковец Р.С. студент группы 253501,
Сугако Т.А. студентка группы 253503, Владымцев В. Д., ассистент каф. Информатики

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Владымцев В.Д. ассистент кафедры информатики

Аннотация. Данная статья посвящена функциям активации - важной составляющей нейронных сетей. В статье рассматриваются основные принципы работы функций активации, их роль в моделировании нелинейных зависимостей между входными и выходными данными, а также области их применения. Также в статье рассматриваются различные виды функций активации, такие как сигмоидная, гиперболический тангенс, ReLU и другие. Описываются их особенности и преимущества, а также рекомендации по выбору функции активации для конкретной задачи.

Ключевые слова. Нейронные сети, функция активации, сигмоидная функция активации, ReLU. ELU

Функция активации является фундаментальной составляющей нейронных сетей, и понимание принципов работы, а также областей применения, способствует созданию более эффективных сетей. Функция активации - это функция в нейроне, которая преобразует входные данные в выходные для их последующей передачи на следующий слой [1]. Входными данными является линейная сумма всех весов (числовая характеристика связи между двумя нейронами) и значений с предыдущего слоя и добавление смещения(bias).

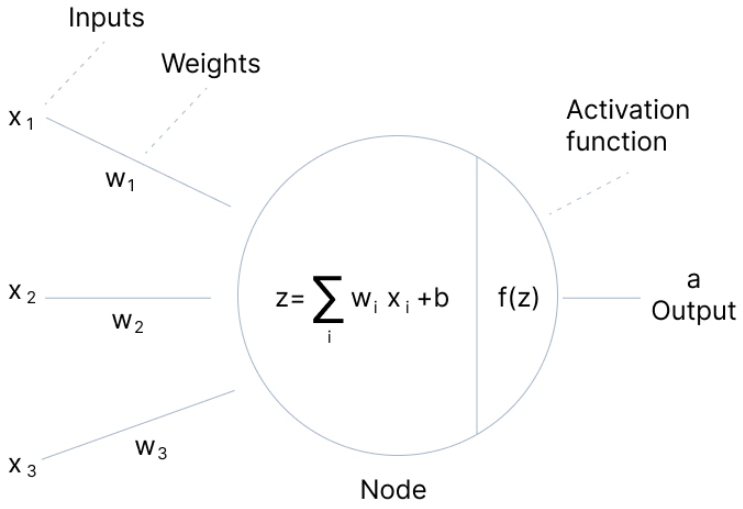

Рисунок 1 - Схема работы функции активации

Главная цель функции активации - внедрение нелинейности в модель нейронной сети. Наличие нелинейности позволяет нейронным сетям разрабатывать сложные представления и функции на основе входных данных, что было бы невозможно при использовании простых линейных функциях, ведь композиция линейных функций есть линейная функция.

Одной из самых часто используемых функций активации является сигмоида.

Функция принимает значение от 0 до 1, задается формулой

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

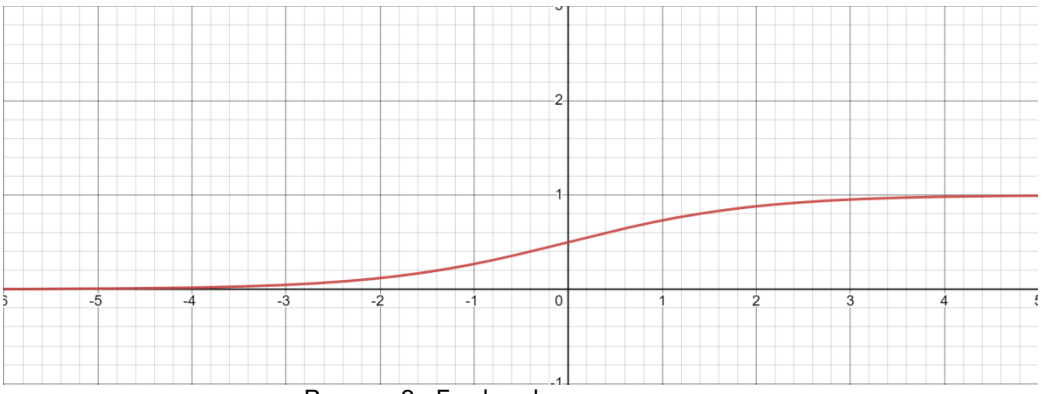

Рисунок 2 - График функции сигмоиды

Сигмоида полезна в задачах бинарной классификации, так как всегда отображает входные данные на значение между 0 и 1, что можно интерпретировать как вероятность принадлежности к одному из двух классов. Еще одним преимуществом сигмоидной функции является ее плавный градиент, что позволяет легко вычислять градиенты при обратном распространении и обновлять веса нейронной сети с помощью градиентного спуска.

Основной недостаток сигмоиды заключается в его производной.

$$\sigma'(x) = \sigma(x)\big(1 - \sigma(x)\big)$$

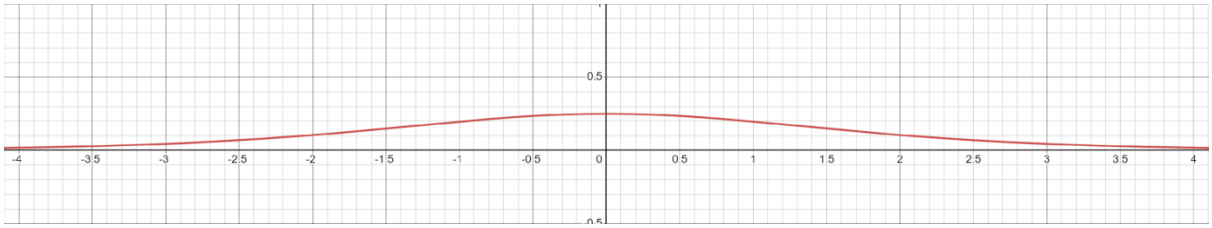

Рисунок 3 - График производной функции сигмоида

Сигмоида имеет ограниченный диапазон значимых значений градиента, который находится между -3 и 3. За пределами этого диапазона функция становится более плоской, что приводит к очень маленьким градиентам. Данное свойство приводит к проблеме исчезающего градиента (The Vanishing Gradient Problem), когда сеть с трудом обучается, поскольку градиенты приближаются к нулю. Кроме того, значения функции не симметричен вокруг нуля, что означает, что выход всех нейронов будет иметь одинаковый знак. Эта особенность делает обучение нейронной сети более сложным и нестабильным.

Гиперболический тангенс (tanh) похож на сигмоидную функцию, но его диапазон значений находится между -1 и 1, что делает его более симметричным вокруг нуля.

Это позволяет использовать его в более широком диапазоне задач, таких как классификация с несколькими классами.

$$tanh(x) = \frac{(e^x - e^{-x})}{(e^x + e^{-x})}$$

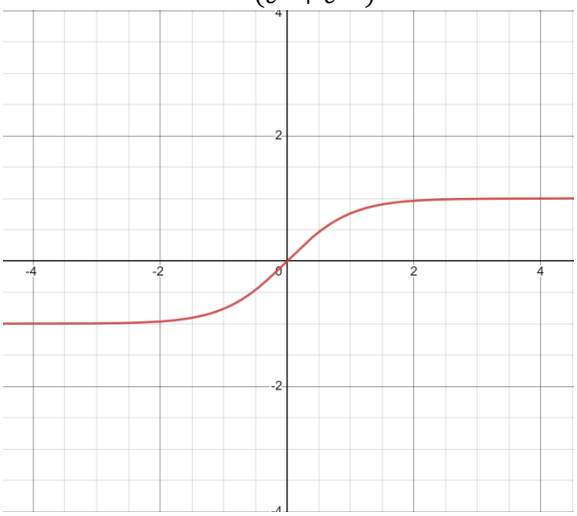

Рисунок 4 - График функции гиперболического тангенса

Как и сигмоида, градиент гиперболического тангенса также может столкнуться с проблемой исчезающего градиента за пределами диапазона значимых значений.

Производная тангенса имеет вид:

$$tanh'(x) = 1 - tanh^2(x)$$

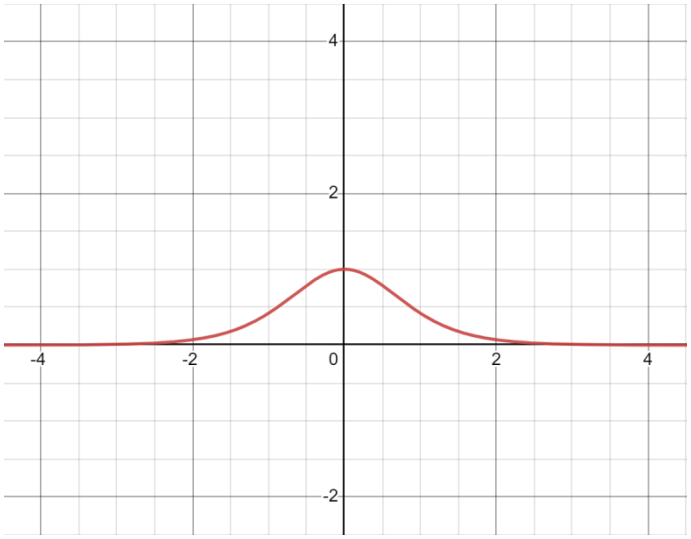

Рисунок 5 - График производной гиперболического тангенса

Решение проблемы исчезающего градиента является использование ReLU (Rectified Linear Unit).

$$ReLU(x) = max(0, x)$$

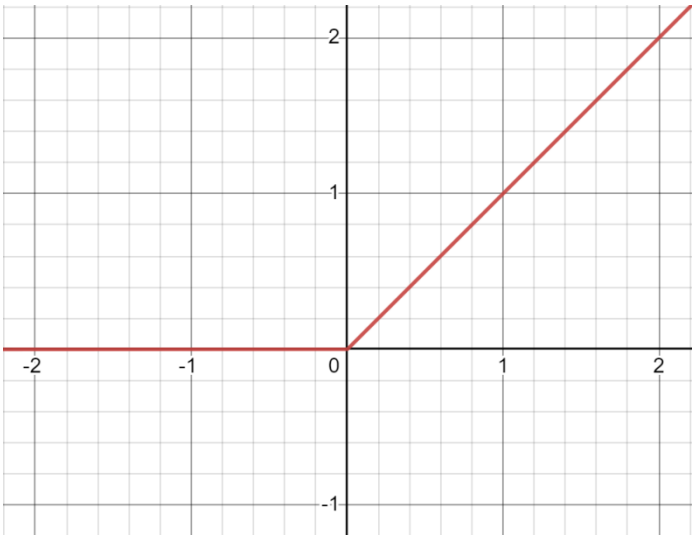

Рисунок 6 - График ReLU

Помимо решения проблемы с градиентом, ReLU также превосходит в простоте и скорости вычислений, что делает ее полезной для глубоких нейронных сетей с большим количеством параметров.

$$ReLU'(x) = \begin{cases} 1, x \geq 0 \\ 0, x < 0 \end{cases}$$

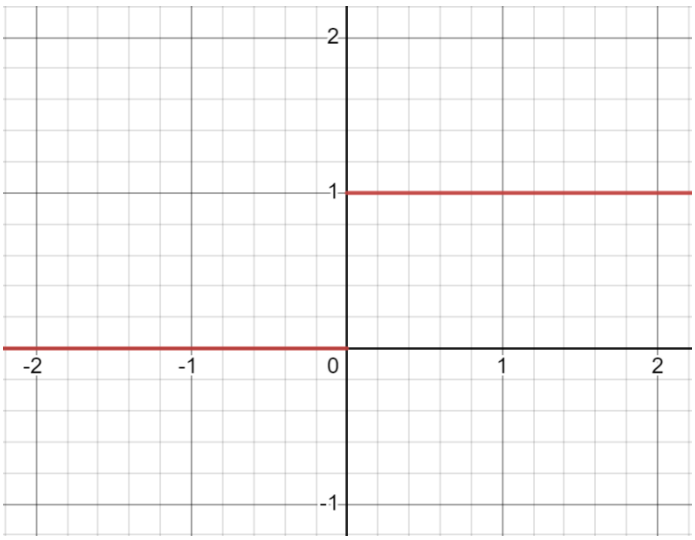

Рисунок 7 - График производной ReLU

Однако ReLU также имеет некоторые недостатки. Один из основных недостатков ReLU заключается в том, что он страдает от проблемы "умирающего ReLU"(dying ReLU [2]), когда большое количество нейронов в сети может стать неактивным и больше никогда не активироваться во время обучения. Это происходит, когда вход в нейрон отрицательный и градиент становится нулевым, в результате чего веса нейрона больше никогда не обновляются. Еще одним недостатком ReLU является то, что он не является плавной функцией, что делает его непригодным для некоторых алгоритмов оптимизации, которые полагаются на плавность, таких как метод сопряженного градиента. Наконец, ReLU не симметрична вокруг нуля, что может привести к тому, что выход всех нейронов будет иметь одинаковый знак, что приведет к замедлению сходимости и нестабильности обучения.

Проблема “умирающего ReLU” решается одной из вариаций данной функции – Leaky ReLU.

$$LeakyReLU(x) = max(0.01x, x)$$

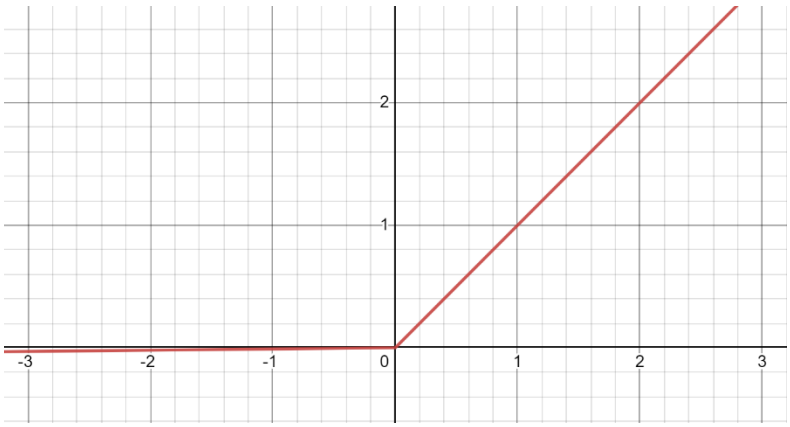

Рисунок 8 - График Leaky ReLU

Leaky ReLU добавляет небольшой наклон к отрицательной области функции, позволяя градиентам по-прежнему проходить через нее и обновлять веса нейронов. Еще одним преимуществом Leaky ReLU является то, что это гладкая функция, что делает ее подходящей для алгоритмов оптимизации, которые полагаются на гладкость, таких как метод сопряженного градиента. Более того, Leaky ReLU также не страдает от проблемы насыщения в положительной области, которая может привести к тому, что градиенты становятся очень маленькими и замедляют обучение.

$$Leaky\ ReLU'(x) = \begin{cases} 1, x \geq 0 \\ 0.01, x < 0 \end{cases}$$

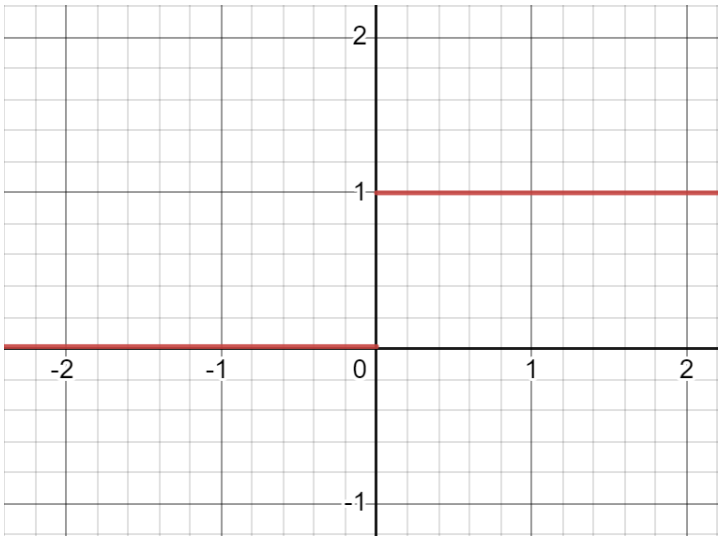

Рисунок 9 - Производная Leaky ReLU

К ограничениям Leaky ReLU относятся непоследовательные предсказания для отрицательных входных значений и небольшой градиент в отрицательной области, который может замедлить обучение параметров.

Еще одна вариация ReLU, направленная на решение проблемы мертвых нейронов - Параметрический ReLU. Эта функция предоставляет наклон отрицательной части функции в качестве аргумента a. Путем обратного распространения происходит обучение наиболее подходящему значению a.

$$ParametricReLU(x) = max(ax, x)$$

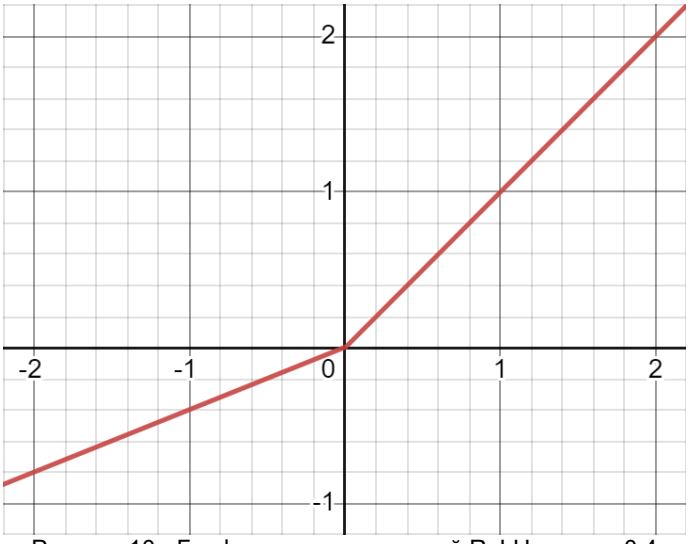

Рисунок 10 - График параметрической ReLU при a = 0.4

Где a - параметр наклона для отрицательных значений.

Параметрическая функция ReLU используется, когда функция Leaky ReLU не справляется с проблемой “мертвых” нейронов, и соответствующая информация не передается в следующий слой. Основное ограничение этой функции в том, что она может работать по-разному для разных задач в зависимости от значения параметра наклона a.

Также существует еще одна альтернатива ReLU - экспоненциальное ReLU, которое также решает проблему мертвых нейронов.

$$ELU(x) = \begin{cases} x, x \geq 0 \\ a(e^x - 1), x < 0 \end{cases}$$

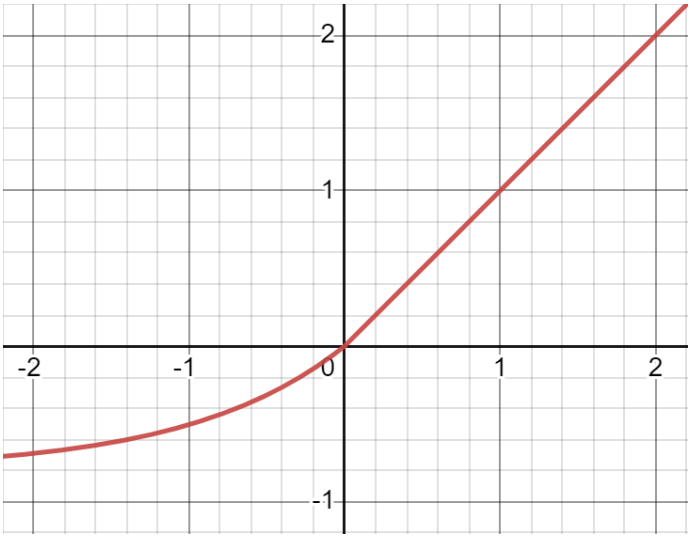

Рисунок 11 - График функции ELU при a = 0.8

Однако у функции активации ELU есть некоторые ограничения. Во-первых, она увеличивает время вычислений из-за экспоненциальной операции, включенной в функцию, что может быть неудобным при обучении больших моделей или работе с большими данными. Во-вторых, ELU имеет гиперпараметр α, который необходимо задавать вручную, так как невозможно обучить. Наконец, ELU не полностью решает проблему взрывающегося градиента (Exploding Gradient [3]), которая все еще может возникать в очень глубоких сетях, но в некоторой степени смягчает ее.

$$ELU'(x) = \begin{cases} 1, x \geq 0 \\ ELU(x) + a, x < 0 \end{cases}$$

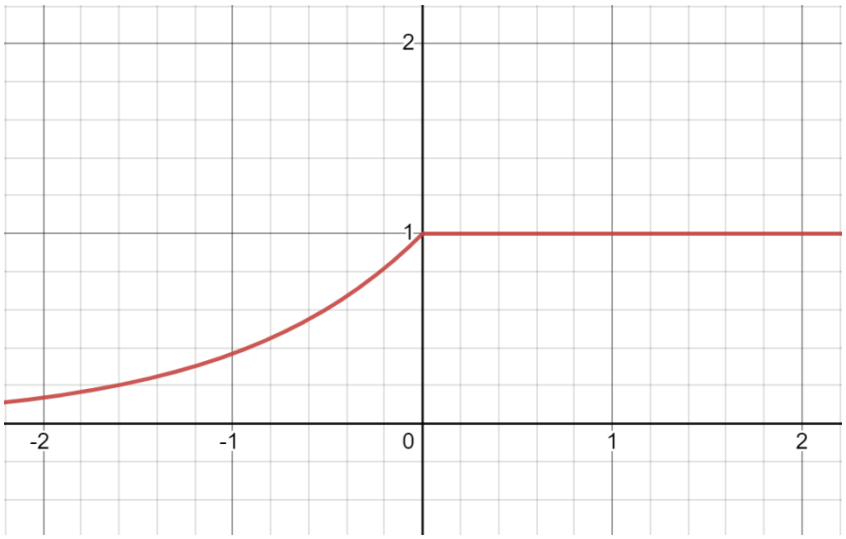

Рисунок 12 - График производной ELU при a = 1

После проведения анализа основных функций активации мы пришли к выводу о том, что некоторые функции могут быть предложены для использования в нейронных сетях. Мы выбрали данные функции с целью решить известные проблемы существующих функций активации, такие как сложные вычисления и исчезающий градиент у сигмоиды и тангенса, а также проблема "мертвых" нейронов у ReLU и его модификаций.

В сравнении с сигмоидной функцией, предложенные нами функции будут считаться намного более эффективными, так как для их вычисления необходимо выполнить лишь несколько простых арифметических операций, таких как сложение, умножение, деление и вычитание, в то время как для вычисления сигмоиды необходимо найти приближение экспоненты в некоторой степени, что требует использования ряда Маклорена и увеличивает время вычислений.

Кроме того, предложенные нами функции обладают непрерывными производными в каждой точке своей области определения, что делает возможным применение некоторых методов обучения, которые недоступны при использовании ReLU-подобных функций.

Строго говоря, предложенные нами функции не имеют параметров и гиперпараметров, что упрощает разработку нейронных сетей с использованием данных функций активации.

Первая предложенная функция активации задается формулой:

$$f(x) = \begin{cases} \frac{x}{1-x}, x < 0 \\ -\frac{2x^2}{25} + x, 0 \le x \le 5 \\ \frac{x}{5} + 2, x > 5 \end{cases}$$

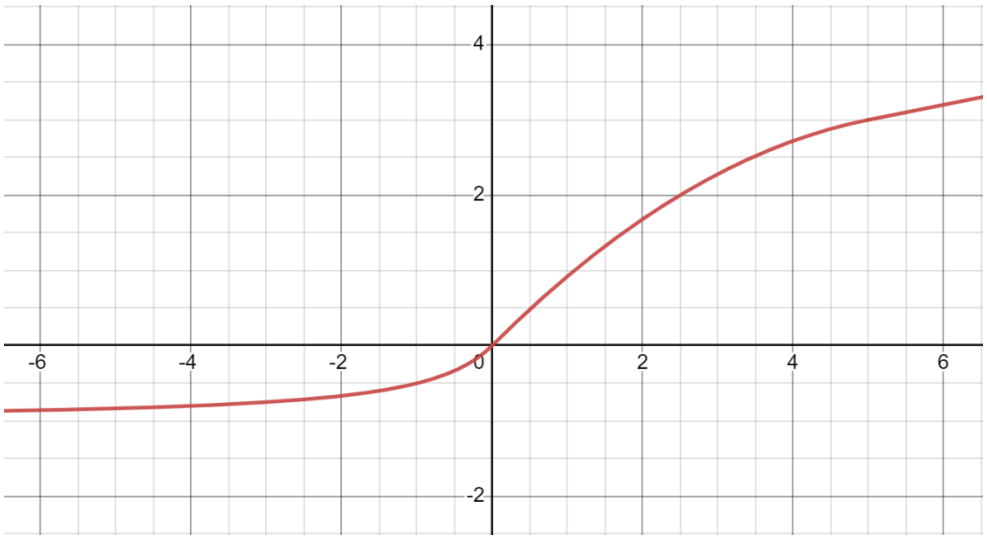

Рисунок 13 - График функции f(x)

$$f'(x) = \begin{cases} \frac{1}{(1-x)^2}, x < 0 \\ -\frac{4x}{25} + 1, 0 \le x \le 5 \\ \frac{1}{5}, x > 5 \end{cases}$$

Вторая предложенная функция активации задается следующей формулой:

$$g(x) = \begin{cases} \frac{x}{1-x}, x < 0 \\ \frac{x^2}{10} + x, 0 \le x \le 5 \\ 2x - 2.5, x > 5 \end{cases}$$

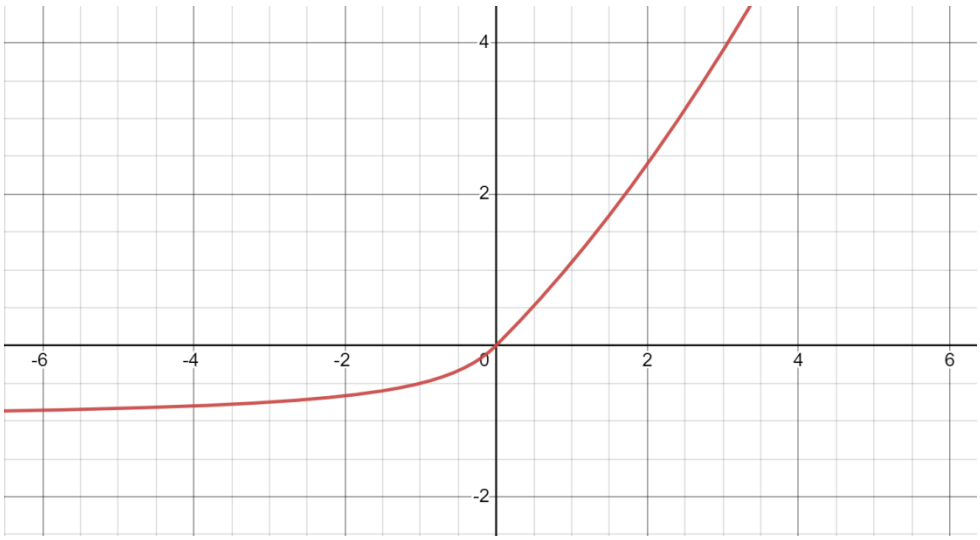

Рисунок 14 - График функции g(x)

$$g'(x) = \begin{cases} \frac{1}{(1-x)^2}, x < 0 \\ \frac{x}{5} + 1, 0 \le x \le 5 \\ 2, x > 5 \end{cases}$$

Выбор функции активации является важным этапом при создании нейронной сети, так как она определяет, какой тип нелинейности будет использоваться для преобразования входных сигналов и генерации выходных сигналов. Это может существенно влиять на способность модели к обучению, скорость сходимости и качество результатов.

При выборе функции активации необходимо учитывать как ее математические свойства, так и специфику задачи, для которой создается нейронная сеть. Например, для задач классификации изображений может быть эффективно использовать ReLU, в то время как для задачи регрессии может быть полезно применить гиперболический тангенс.

Кроме того, необходимо учитывать возможные проблемы, связанные с выбранной функцией активации, такие как исчезающий градиент или проблема "мертвых" нейронов. В таких случаях можно рассмотреть альтернативные функции активации, которые не имеют этих проблем.

В целом, выбор функции активации должен основываться на балансе между ее вычислительной эффективностью, математическими свойствами и спецификой задачи. При необходимости можно экспериментировать с различными функциями активации и выбрать ту, которая дает лучшие результаты в конкретной задаче. Общая схема выбора может быть сведена к таблице на рис. 15

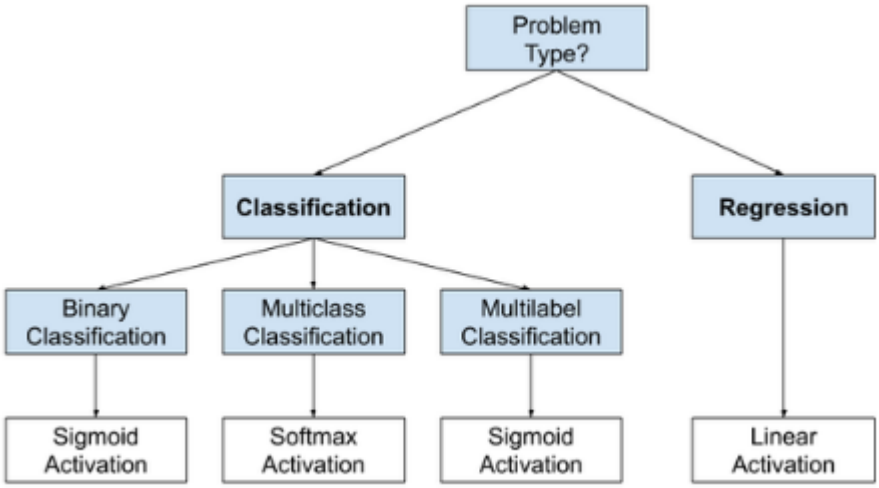

Рисунок 15 - Схема выбора функции активации

В ходе нашей работы по изучению функций активаций мы смогли выдвинуть несколько гипотез о том, какие ранее неиспользованные функции активации могут быть полезны для различных моделей нейронных сетей.

Список использованных источников:

1. Sharma, S., Sharma, S. and Athaiya, A., 2017. Activation functions in neural networks. Towards Data Sci, 6(12), pp.310-316.

2. Lu, L., Shin, Y., Su, Y. and Karniadakis, G.E., 2019. Dying relu and initialization: Theory and numerical examples. arXiv preprint arXiv:1903.06733.

3. Philipp, G., Song, D. and Carbonell, J.G., 2017. The exploding gradient problem demystified-definition, prevalence, impact, origin, tradeoffs, and solutions. arXiv preprint arXiv:1712.05577.

4. Agostinelli, F., Hoffman, M., Sadowski, P. and Baldi, P., 2014. Learning activation functions to improve deep neural networks. arXiv preprint arXiv:1412.6830.

UDC

ACTIVATION FUNCTIONS

Sugako T.A., Demeschenko M.V., Markovets R.S.

Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus

Vladymtsev V.D. – Assistant of the Department of Informatics

Annotation. This paper focuses on activation functions, an important component of neural networks. The paper discusses the basic principles of activation functions, their role in modelling non-linear dependencies between input and output data, and their application areas. Various types of activation functions, such as sigmoid, hyperbolic tangent, ReLU and others are also considered. Their features and advantages are described, as well as recommendations for choosing an activation function for a particular task.

Keywords. Neural networks, activation function, sigmoid activation function, ReLU. ELU.