

3. Филатова, Т. В. Применение нейронных сетей для аппроксимации данных [Электронный ресурс] / Т. В. Филатова // Вестник Томского государственного университета. – 2004. – № 284. – URL: <https://cyberleninka.ru/article/n/primenenie-neuronnyh-setey-dlya-approksimatsii-dannyh> (дата обращения: 29.01.2019).

УДК 004.032.26

Соснин А. С., Сулова И. А.

**ФУНКЦИИ АКТИВАЦИИ НЕЙРОСЕТИ: СИГМОИДА, ЛИНЕЙНАЯ,
СТУПЕНЧАТАЯ, RELU, ТАHN**

Александр Сергеевич Соснин

Студент магистратуры

salexandr18@gmail.com

Ирина Александрована Сулова

кандидат педагогических наук, доцент

irina.suslova@rsyru.ru

ФГАОУ ВО «Российский государственный профессионально-педагогический университет», Россия, Екатеринбург

**FUNCTIONS OF NEURAL NET ACTIVATION: SIGMOID, LINEAR,
STEP, RELU, TANH**

Aleksandr Sergeevich Sosnin

Irina Aleksandrovana Suslova

Russian State Vocation Pedagogical University, Russia, Yekaterinburg

Аннотация. В статье раскрываются основные активационные функции, рассматриваются их отличительные особенности, присущие им достоинства и недостатки.

Abstract. The article reveals the main activation functions, highlights their distinctive features, their inherent strengths and weaknesses.

Ключевые слова: *нейронная сеть, активационная функция, линейная функция, сигмоида, параболический тангенс, ReLu.*

Keywords: *neural network, activation function, linear function, sigmoid, parabolic tangent, ReLu.*

Искусственный нейрон представляет собой узел искусственной нейронной сети, являющийся упрощенной моделью естественного нейрона. Математически нейрон представляется в виде функции. Если говорить простыми словами, нейрон производит расчёт взвешенной суммы значений на своих входах, добавляет смещение и определяет, следует исключить данное значение или использовать дальше [1].

Функция, в виде которой представляются нейроны — называется функцией активации. Функция активации определяет выходное значение нейрона в зависимости от результата взвешенной суммы входов и порогового значения.

Рассмотрим нейрон:

$$Y = \sum (\textit{weight} * \textit{input}) + \textit{bias}$$

Значение Y лежит в диапазоне $(-\infty; +\infty)$. В реальности невозможно точно указать границу, после которой произойдет активация нейрона. Для определения момента (значения) активации нейрона, применяют активационные функции. Данные функции проверяют получаемое от нейрона значения Y на предмет того, должны ли внешние связи рассматривать этот нейрон как активированный, или его можно игнорировать.

Ступенчатая функция

Первая, проблема которая возникает заключается в определении, что считать границей активации для активационной функции. Самым простым решением является установить пороговое значение. Если Y больше этого значения, считать нейрон активированным. В противном случае говорим, что

нейрон неактивен. Такая схема выглядит вполне работоспособной, но для начала требуется ее формализовать [2, 3].

Итак, функция A активирована, при условии, что $Y >$ пороговое, иначе нет.

Рассмотренная функция является ступенчатой. График функции представлен на рисунке 1.

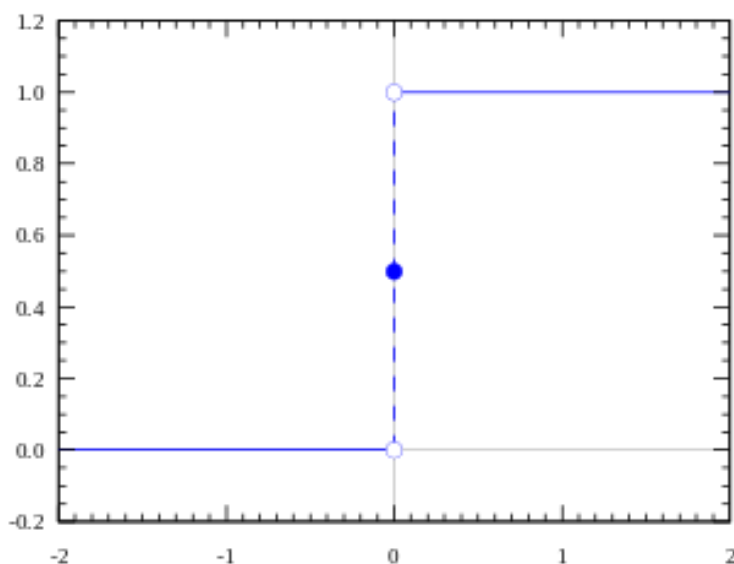


Рисунок 1 — Ступенчатая функция

Функция принимает значение 1 (активирована), когда $Y > 0$ (больше порогового значения), и значение 0 (не активирована) в противном случае.

Таким образом реализуется активационная функция нейрона. Данный способ является самым простым, однако в нём есть недостатки. Рассмотрим следующую ситуацию.

Представим, что требуется создать бинарный классификатор — модель, которая должна говорить «да» или «нет» (активирован или нет). Ступенчатая функция идеально подходит для данного случая — она в точности выводит 1 или 0 .

Однако если требуется классифицировать несколько классов: класс 1 , класс 2 , класс 3 и т. д, то активированными могут оказаться несколько нейронов. В таком случае все нейроны из функции активации выдадут значение 1 . В таком случае возникает вопрос, к какому классу отнести данный предмет.

Для точного определения класса предмета должна происходить активация только одного нейрона, а функции активации остальных нейронов должны быть равны нулю. Только в этом случае можно быть уверенным, что сеть правильно определяет класс. Такую сеть труднее обучать и добиваться сходимости. Если активационная функция не бинарная, то возможны значения «активирован на 50%», «активирован на 20%» и так далее. Если активированы несколько нейронов, можно найти нейрон с наибольшим значением активационной функции.

Для построения более универсальной сети требуется получать промежуточные значения активационной функции, а не только определять «активирован» или нет.

Линейная функция активации

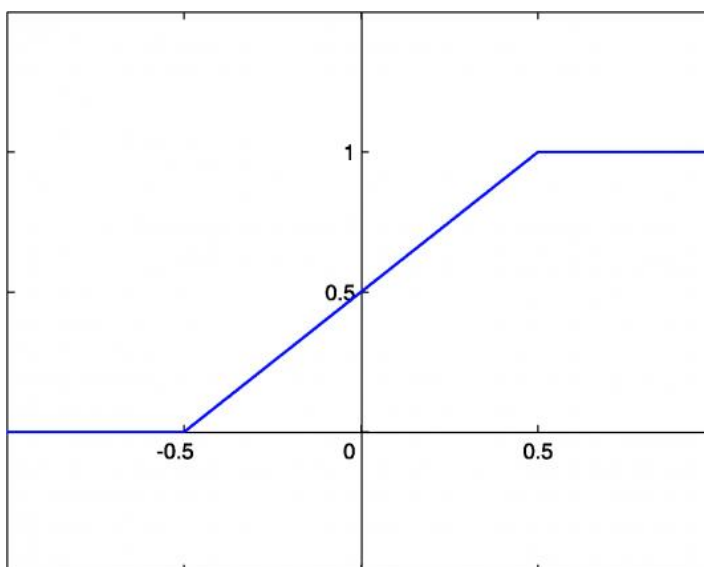


Рисунок 2 — Линейная функция

Линейная функция (рисунок 2) имеет два линейных участка, где функция активации равна минимально допустимому и максимально допустимому значению. На рисунке один также присутствует участок, на котором функция возрастает строго монотонно.

Данная активационная функция позволяет получить целый спектр значений, а не только бинарный ответ. Также существует возможность объединения нескольких нейронов вместе и, если более одного нейрона активировано,

решение принимается на основе максимального значения. Однако и данная функция имеет свои недостатки [3].

Для примера рассмотрим линейную функцию (1):

$$Y=cx \quad (1)$$

Производная от (1) по x равна c . Из этого следует, что градиент никак не связан с X . Градиент является постоянным вектором, а спуск производится по постоянному градиенту. В случае получения ошибочного ответа, дообучение, проведенное методом обратного распространения ошибки, тоже будет постоянно, и не зависимо от изменения на входе значения $\Delta(x)$. Это не есть хорошо. Но существует и другая проблема. Рассмотрим связанные слои. Каждый слой активируется линейной функцией. Значение с этой функции идет в следующий слой в качестве входа, второй слой считает взвешенную сумму на своих входах и, в свою очередь, включает нейроны в зависимости от другой линейной активационной функции [5].

Не имеет значения, сколько слоев было использовано. Если все они по своей природе линейные, то и итоговая функция активации будет иметь линейный характер. Это означает, что два слоя (или N слоев) могут быть заменены одним слоем.

Сигмоида

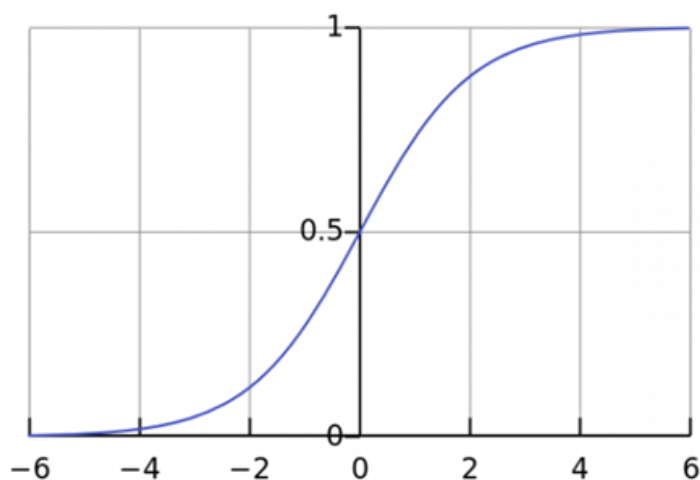


Рисунок 3 — Сигмоида

Сигмоида — это гладкая монотонная возрастающая нелинейная функция (2).

$$A = \frac{1}{1+e^{-x}} \quad (2)$$

Рассмотрим её преимущества.

Во-первых, сигмоида — не линейна по своей природе, а результатом комбинации набора таких функций имеет нелинейный характер, что позволяет строить сеть из нескольких слоев.

Главным преимуществом данной функции, является то, что она не является бинарной, в отличие от ступенчатой функции. Для сигмоиды также характерен гладкий градиент [3, 4].

В диапазоне значений X (-2;2) значения Y изменяется очень быстро. Это означает, что любое малое изменение значения X в этой области влечет существенное изменение значения Y . Такое поведение функции указывает на то, что Y имеет тенденцию стремиться к одному из краев диапазона.

Другое преимущество сигмоиды над линейной функцией заключается в следующем. В первом случае имеем фиксированный диапазон значений функции — $[0,1]$, тогда как линейная функция изменяется в пределах $(-\infty, \infty)$. Такое свойство сигмоиды очень полезно, так как не приводит к ошибкам в случае больших значений порога активации.

На данный момент сигмоида является одной из самых часто применяемых активационных функций в нейросетях. Но и у неё есть недостатки, на которые стоит обратить внимание.

При приближении к границам диапазона, значения Y начинают слабо реагировать на изменения в X . Это означает, что градиент в таких областях имеет малые значения.

В таком случае значение градиента мало и стремится к 0. Нейросеть перестает обучаться вовсе или делает это крайне медленно. Однако существует

множество вариантов решения данной проблемы. Несмотря на свои недостатки данная функция очень популярна, и часто применяется для решения задач классификации.

Гиперболический тангенс

Еще одна часто применяемая активационная функция — гиперболический тангенс (3) (рисунок 4).

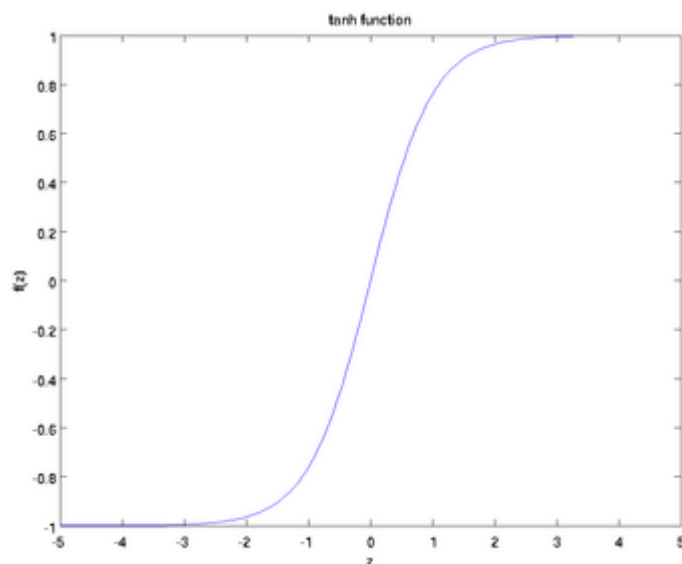


Рисунок 4 — Гиперболический тангенс

$$f(x) = \tanh(x) = \frac{2}{1+e^{-2x}} - 1 \quad (3)$$

Гиперболический тангенс является скорректированной сигмоидной функцией.

$$\tanh(x) = 2 \textit{sigmoid}(2x) - 1 \quad (4)$$

Данная функция имеет те же характеристики, что и сигмоида, рассмотренная ранее. Она так же имеет не линейный характер и хорошо подходит для комбинации слоёв, а диапазон значений функции лежит в диапазоне (-1, 1). Узкий диапазон значений предотвращает, перегрузку активационной функции. Однако стоит отметить, что градиент тангенциальной функции больше, чем у сигмоиды (производная круче). Решение выборе функции, зависит требований к амплитуде градиента. Также, как и сигмоиде, гиперболическому тангенсу свойственная проблема исчезновения градиента [6].

Тангенс также является очень популярной и часто применяемой активационной функцией.

ReLU

В последние годы большую популярность приобрела функция активации под названием «выпрямитель» (rectifier, по аналогии с однополупериодным выпрямителем в электротехнике). Нейроны с данной функцией активации называются ReLU (rectified linear unit). ReLU имеет следующую формулу (5) и реализует простой пороговый переход в нуле (Рисунок 5).

$$A(x) = \max(0, x) \quad (5)$$

Исходя из определения, становится понятно, что ReLU возвращает значение x , если x положительно, и 0 в противном случае. График функции приведен на рисунке 5.

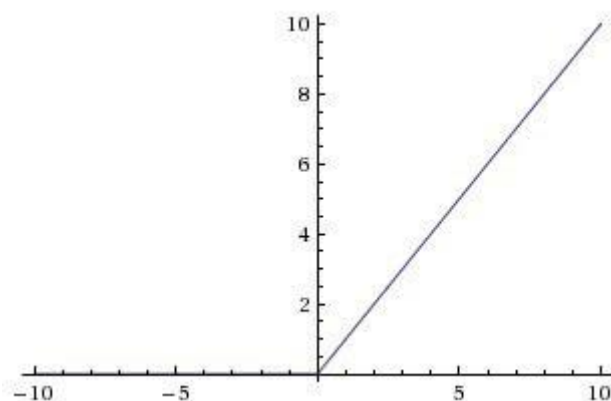


Рисунок 5 — Функция ReLU

На первый взгляд кажется, что ReLU наследует все те же проблемы, что характерны линейной функции, так как ReLU линейна в первом квадранте. Но на самом деле, ReLU не линейна по своей природе, а комбинация ReLU также не линейна.

Вычисление гиперболического тангенса и сигмоиды сопровождается выполнением ресурсоемких операций. Однако вычисление ReLU реализовано с помощью простого порогового преобразования матрицы активаций в нуле. Также к преимуществу можно отнести то, что ReLU не подвержен насыщению.

Применение ReLU существенно повышает скорость сходимости стохастического градиентного спуска по сравнению с сигмоидой и гиперболическим тангенсом. Считается, что это обусловлено линейным характером и отсутствием насыщения данной функции.

К сожалению, ReLU не всегда достаточно надежны и в процессе обучения могут выходить из строя («умирать»). Слишком большой градиент, проходящий через ReLU, приводит к такому обновлению весов, что данный нейрон никогда больше не активируется. Если это произойдет, то, начиная с данного момента, градиент, проходящий через этот нейрон, всегда будет равен нулю. Соответственно, данный нейрон будет необратимо выведен из строя. Например, при слишком большой скорости обучения (learning rate), может оказаться, что до 40% ReLU «мертвы» (то есть, никогда не активируются). Эта проблема решается посредством подбора приемлемой скорости обучения.

Заключение

Каждая из рассмотренных функций имеет свои отличительные свойства, преимущества и недостатки. Не одна из функций не является универсальной однозначно сказать в каком случае следует использовать ReLU, сигмоиду или tanh невозможно. Зная некоторые характеристики функции, которую требуется аппроксимировать, следует выбирать такую активационную функцию, которая аппроксимирует искомую функцию максимально точно и приведёт к быстрому обучению.

Например, сигмоида хорошо для использования в задачах классификации. Например, аппроксимацию классифицирующей функции, комбинацией сигмоид можно провести легче, чем применяя ReLU.

Следует использовать ту функцию, с которой процесс обучения и сходимость будут быстрее. Более того, возможно применение собственных, кustomных функций. Если природу исследуемой функции не известна, в таком случае следует начать с ReLU и потом двигаться в обратном направлении.

Список литературы

1. Горбачевская, Е. Н. Классификация нейронных сетей [Электронный ресурс] / Е. Н. Горбачевская // Вестник ВУиТ. – 2012. – № 2 (19). – URL: <https://cyberleninka.ru/article/n/klassifikatsiya-neyronnyh-setey> (дата обращения: 28.01.2019).
2. Гришин, Г. А. Использование функции Хевисайда при анализе функционирования систем массового обслуживания [Электронный ресурс] / Г. А. Гришин // Вестник МГТУ им. Н. Э. Баумана. Серия «Приборостроение». – 2006. – № 3. – URL: <https://cyberleninka.ru/article/n/ispolzovanie-funktsii-hevisayda-pri-analize-funktsionirovaniya-sistem-massovogo-obsluzhivaniya> (дата обращения: 28.01.2019).
3. Функции активации в нейронных сетях [Электронный ресурс]. – URL: <http://www.aiportal.ru/articles/neural-networks/activation-function.html> (дата обращения: 28.01.2019).
4. Сигмоида [Электронный ресурс]. – URL: <http://www.wikiwand.com/ru/%D0%A1%D0%B8%D0%B3%D0%BC%D0%BE%D0%B8%D0%B4%D0%B0> (дата обращения: 28.01.2019).
5. Метод обратного распространения ошибки: математика, примеры, код [Электронный ресурс]. – URL: <https://neurohive.io/ru/osnovy-data-science/obratnoe-rasprostranenie/>.
6. Литвинов, М. И. Применение нейронной сети с функцией активации вида $\sin(x)/x$ и $|\sin(x)/x|$ для экстраполяции стохастической функции (на примере продаж компании) [Электронный ресурс] / М. И. Литвинов // Новые информационные технологии в автоматизированных системах. – 2009. – № 12. – URL: <https://cyberleninka.ru/article/n/primenenie-neyronnoy-seti-s-funktsiey-aktivatsii-vida-sin-x-x-i-sin-x-x-dlya-ekstrapolyatsii-stohasticheskoy-funktsii-na-primere-prodazh> (дата обращения: 28.01.2019).