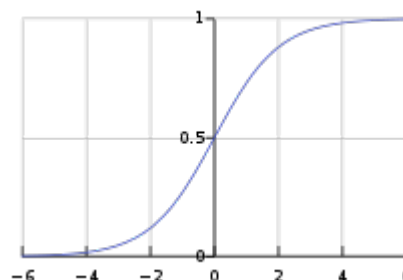


ВИКИПЕДИЯ

Функция активации

Материал из Википедии — свободной энциклопедии

В искусственных нейронных сетях **функция активации** нейрона определяет выходной сигнал, который определяется входным сигналом или набором входных сигналов. Стандартная компьютерная микросхема может рассматриваться как цифровая сеть функций активации, которые могут принимать значения «ON» (1) или «OFF» (0) в зависимости от входа. Это похоже на поведение линейного перцептрона в нейронных сетях. Однако только нелинейные функции активации позволяют таким сетям решать нетривиальные задачи с использованием малого числа узлов. В искусственных нейронных сетях эта функция также называется передаточной функцией.



Логистическая функция активации

Содержание

Функции

[Альтернативные структуры](#)

[Сравнение функций активации](#)

См. также

Примечания

Литература

Функции

В биологических нейронных сетях функция активации обычно является абстракцией, представляющей скорость возбуждения потенциала действия в клетке ^[1]. В наиболее простой форме эта функция является двоичной — то есть нейрон либо возбуждается, либо нет. Функция выглядит как $\phi(v_i) = U(v_i)$, где U — ступенчатая функция Хевисайда. В этом случае нужно использовать много нейронов для вычислений за пределами линейного разделения категорий.

Прямая с положительным угловым коэффициентом может быть использована для отражения увеличения скорости возбуждения по мере увеличения входного сигнала. Такая функция имела бы вид $\phi(v_i) = \mu v_i$, где μ — наклон прямой. Эта функция активации линейна, а потому имеет те же проблемы, что и двоичная функция. Кроме того, сети, построенные с использованием таковой модели, имеют неустойчивую сходимость, поскольку возбуждение приоритетных входов нейронов стремится к безграничному увеличению, так как эта функция не нормализуема.

Все проблемы, упомянутые выше, можно решить с помощью нормализуемой сигмоидной функции активации. Одна из реалистичных моделей остаётся в нулевом состоянии, пока не

придёт входной сигнал, в этот момент скорость возбуждения сначала быстро возрастает, но постепенно достигает асимптоты в 100 % скорости возбуждения. Математически, это выглядит как $\phi(v_i) = U(v_i)\text{th}(v_i)$, где гиперболический тангенс можно заменить любой сигмоидой. Такое поведение реально отражается в нейроне, поскольку нейроны не могут физически возбуждаться быстрее некоторой определённой скорости. Эта модель, однако, имеет несколько проблем в вычислительных сетях, поскольку функция не дифференцируема, что нужно для вычисления обратной передачи ошибки обучения.

Последняя модель, которая используется в многослойных перцептронах — сигмоидная функция активации в форме гиперболического тангенса. Обычно используются два вида этой функции: $\phi(v_i) = \text{th}(v_i)$, образ которой нормализован к интервалу $[-1, 1]$, и $\phi(v_i) = (1 + \exp(-v_i))^{-1}$, сдвинутая по вертикали для нормализации от 0 до 1. Последняя модель считается более биологически реалистичной, но имеет теоретические и экспериментальные трудности с вычислительными ошибками некоторых типов.

Альтернативные структуры

Специальный класс функций активации, известный как радиальные базисные функции (РБФ) используются в РБФ сетях, которые крайне эффективно в качестве универсальных аппроксиматоров функций. Эти функции активации могут принимать множество форм, но обычно берётся одна из следующих трёх функций:

- Гауссова: $\phi(v_i) = \exp\left(-\frac{\|v_i - c_i\|^2}{2\sigma^2}\right)$
- Мультиквадратичная (англ. *Multiquadratics*): $\phi(v_i) = \sqrt{\|v_i - c_i\|^2 + a^2}$
- Обратная мультиквадратичная (англ. *Inverse Multiquadratics*):
 $\phi(v_i) = (\|v_i - c_i\|^2 + a^2)^{-1/2}$

где c_i является вектором, представляющим *центр* функции, а a и σ являются параметрами, влияющими на расходимость радиуса.

Методы опорных векторов (SVM) могут эффективно использовать класс функций активации, который включает как сигмоиды, так и РБФ. В этом случае вход преобразуется для отражения гиперплоскости границы решений основываясь на нескольких обучающих входных данных, называемых *опорными векторами* x . О функции активации для закрытого уровня этих машин говорят как о *ядре скалярного произведения* (англ. *inner product kernel*), $K(v_i, x) = \phi(v_i)$. Опорные вектора представляются как центры в РБФ с ядром, равным функции активации, но они принимают единственный вид в перцептроне

$$\phi(v_i) = \text{th}\left(\beta_1 + \beta_0 \sum_j v_{i,j} x_j\right),$$

где для сходимости β_0 и β_1 должны удовлетворять некоторым условиям. Эти машины могут принимать полиномиальные функции активации любого порядка

$$\phi(v_i) = \left(1 + \sum_j v_{i,j} x_j\right)^p \quad [2].$$

Функции активации бывают следующих типов:

- Тожественная функция
- Двоичная ступенчатая функция
- Биполярная^[3] ступенчатая функция
- Сигмоидная функция
 - Двоичная сигмоидальная функция
 - Биполярная сигмоидная функция
- Функция подъёма^[4]

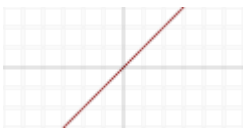
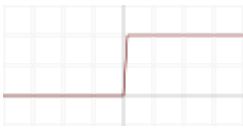
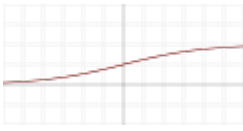
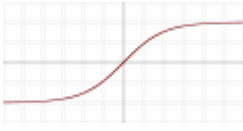
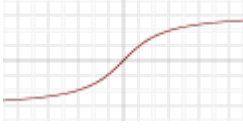
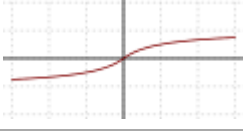
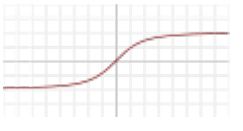

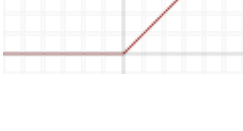

Сравнение функций активации

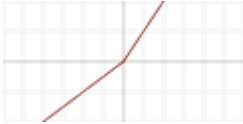
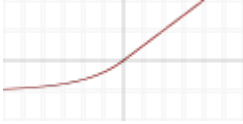
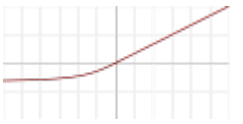
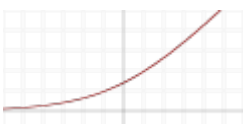

Некоторые желательные свойства функция активации:

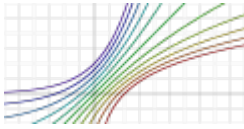
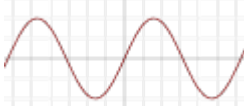
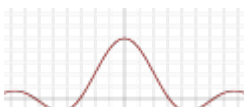

- **Нелинейность** – Если функция активации нелинейна, можно доказать, что двухуровневая нейронная сеть будет универсальным аппроксиматором функции ^[5]. Тожественная функция активации не удовлетворяет этому свойству. Если несколько уровней используют тождественную функцию активации, вся сеть эквивалентна одноуровневой модели.
- **Непрерывная дифференцируемость** – Это свойство желательно (RELU не является непрерывно дифференцируемой и имеет некоторые проблемы с оптимизацией, основанной на градиентном спуске, но остаётся допустимой возможностью) для обеспечения методов оптимизации на основе градиентного спуска. Двоичная ступенчатая функция активации не дифференцируема в точке 0 и её производная равна 0 во всех других точках, так что методы градиентного спуска не дают никакого успеха для неё^[6].
- **Область значений** – Если множество значений функции активации ограничено, методы обучения на основе градиента более стабильны, поскольку представления эталонов существенно влияют лишь на ограниченный набор весов связей. Если область значений бесконечна, обучение, как правило, более эффективно, поскольку представления эталонов существенно влияют на большинство весов. В последнем случае обычно необходим меньший темп обучения.
- **Монотонность** – Если функция активации монотонна, поверхность ошибок, ассоциированная с одноуровневой моделью, гарантированно будет выпуклой ^[7].
- **Гладкие функции с монотонной производной** – Показано, что в некоторых случаях они обеспечивают более высокую степень общности.
- **Аппроксимирует тождественную функцию около начала координат** – Если функции активации имеют это свойство, нейронная сеть будет обучаться эффективно, если её веса инициализированы малыми случайными значениями. Если функция активации не аппроксимирует тождество около начала координат, нужно быть осторожным при инициализации

весов^[8]. В таблице ниже функции активации, у которых $f(0) = 0$, $f'(0) = 1$ и f' непрерывна в точке 0, помечены как имеющие это свойство.

Следующая таблица сравнивает свойства некоторых функций активации, которые являются функциями одной свёртки x от предыдущего уровня или уровней:

Название	График	
<u>Тождественная</u>		$f(x) = x$
<u>Единичная ступенька</u>		$f(x) = \begin{cases} 0 & x < 0 \\ 1 & x \geq 0 \end{cases}$
<u>Логистическая</u> (сигмоида или Гладкая ступенька)		$f(x) = \sigma(x) = \frac{1}{1 + e^{-x}}$ [1] (https://ru.wikipedia.org/wiki/%D0%A4%D1%83%D0%BD%D)
<u>th</u>		$f(x) = \operatorname{th}(x) = \frac{(e^x - e^{-x})}{(e^x + e^{-x})}$
<u>arctg</u>		$f(x) = \operatorname{tg}^{-1}(x)$
<u>Softsign</u> ^{[9][10]}		$f(x) = \frac{x}{1 + x }$
Обратный квадратный корень (англ. <i>Inverse square root unit</i> , ISRU) ^[11]		$f(x) = \frac{x}{\sqrt{1 + \alpha x^2}}$
<u>Линейный выпрямитель</u> (или <u>Полулинейный элемент</u>) (англ. <i>Rectified linear unit</i> , ReLU) ^{[12][13]}		$f(x) = \begin{cases} 0 & x < 0 \\ x & x \geq 0 \end{cases}$
<u>Линейный выпрямитель с «утечкой»</u> (англ. <i>Leaky rectified linear unit</i> , Leaky ReLU) ^[14]		$f(x) = \begin{cases} 0,01x & x < 0 \\ x & x \geq 0 \end{cases}$
<u>Параметрический линейный выпрямитель</u> (англ. <i>Parameteric rectified linear unit</i> , PReLU) ^[15]		$f(\alpha, x) = \begin{cases} \alpha x & x < 0 \\ x & x \geq 0 \end{cases}$

Название	График	
Рандомизированный линейный выпрямитель с «утечкой» (англ. <i>Randomized leaky rectified linear unit</i> , RReLU) ^[16]		$f(\alpha, x) = \begin{cases} \alpha x & x < 0 \\ x & x \geq 0 \end{cases}$ [3] (https://ru.wikipedia.org/wiki/%D0%A4%D1%83%D0%BD%D)
Экспоненциальная линейная функция (англ. <i>Exponential linear unit</i> , ELU) ^[17]		$f(\alpha, x) = \begin{cases} \alpha(e^x - 1) & x < 0 \\ x & x \geq 0 \end{cases}$
Масштабированная экспоненциальная линейная функция (англ. <i>Scaled exponential linear unit</i> , SELU) ^[18]		$f(\alpha, x) = \lambda \begin{cases} \alpha(e^x - 1) & x < 0 \\ x & x \geq 0 \end{cases}$ с $\lambda = 1,0507$ и $\alpha = 1,67326$
Линейный S-выпрямитель (англ. <i>S-shaped rectified linear activation unit</i> , SReLU) ^[19]		$f_{t_l, a_l, t_r, a_r}(x) = \begin{cases} t_l + a_l(x - t_l) & x \leq t_l \\ x & t_l < x < t_r \\ t_r + a_r(x - t_r) & x \geq t_r \end{cases}$ t_l, a_l, t_r, a_r являются параметрами.
Обратный квадратный линейный корень (англ. <i>Inverse square root linear unit</i> , ISRLU) ^[11]		$f(x) = \begin{cases} \frac{x}{\sqrt{1+\alpha x^2}} & x < 0 \\ x & x \geq 0 \end{cases}$
Адаптивная кусочно-линейная функция (англ. <i>Adaptive piecewise linear</i> , APL) ^[20]		$f(x) = \max(0, x) + \sum_{s=1}^S a_i^s \max(0, -x + b_i^s)$
SoftPlus ^[21]		$f(x) = \ln(1 + e^x)$
Выгнутая тождественная функция (англ. <i>Bent identity</i>)		$f(x) = \frac{\sqrt{x^2 + 1} - 1}{2} + x$
Сигмоидно-взвешенная линейная функция (англ. <i>Sigmoid-weighted linear unit</i> , SiLU) ^[22]		$f(x) = x \cdot \sigma(x)$ [5] (https://ru.wikipedia.org/wiki/%D0%A4%D1%83%D0%BD%D)

Название	График	
SoftExponential ^[23]		$f(\alpha, x) = \begin{cases} -\frac{\ln(1-\alpha(x+\alpha))}{\alpha} & \alpha < 0 \\ x & \alpha = 0 \\ \frac{e^{\alpha x}-1}{\alpha} + \alpha & \alpha > 0 \end{cases}$
Синусоида ^[24]		$f(x) = \sin(x)$
Sinc		$f(x) = \begin{cases} 1 & x = 0 \\ \frac{\sin(x)}{x} & x \neq 0 \end{cases}$
Гауссова		$f(x) = e^{-x^2}$

[↑] Здесь, H является ступенчатой функцией Хевисайда.

[↑] α является стохастической переменной, берущейся из равномерного распределения в момент обучения, значение которой фиксируется равным математическому ожиданию распределения в момент тестирования.

[↑] [↑] [↑] Здесь σ является логистической функцией.

Следующая таблица перечисляет функции активации, которые не являются функциями от одной свёртки x от предыдущего уровня или уровней:

Название	Уравнение	
Softmax	$f_i(\vec{x}) = \frac{e^{x_i}}{\sum_{j=1}^J e^{x_j}}$ <p>для $i = 1, \dots, J$</p>	$\frac{\partial f_i(\vec{x})}{\partial x_j} = f_i(\vec{x})(\delta_{ij} - f_j(\vec{x}))$ <p>[7] (https://ru.wikipedia.org/wiki/%D0%A4%D1%83%D0%BD%D0%BA%D1%8)</p>
Maxout ^[25]	$f(\vec{x}) = \max_i x_i$	$\frac{\partial f}{\partial x_j} = \begin{cases} 1 & j = \operatorname{argmax}_i x_i \\ 0 & j \neq \operatorname{argmax}_i x_i \end{cases}$

[↑] Здесь δ_{ij} обозначает символ Кронекера.

См. также

- Логистическая функция
- Выпрямитель (нейронные сети)
- Устойчивость (теория обучения)
- Softmax

Примечания

- ↑ Hodgkin, Huxley, 1952, с. 500–544.
- ↑ Haykin, 1999.
- ↑ Биполярная – принимающая значение -1 до начала координат и 1 после, в отличие от двоичной ступенчатой функции, которая принимает до начала координат значения 0.)
- ↑ Функция подъёма принимает значение 0 до начала координат и линейна после.
- ↑ Cybenko, 2006, с. 303.
- ↑ Snyman, 2005.
- ↑ Wu, 2009, с. 3432–3441.
- ↑ Sussillo, David & Abbott, L. F. (2014-12-19), Random Walk Initialization for Training Very Deep Feedforward Networks, *arXiv:1412.6558* (<http://www.arxiv.org/abs/1412.6558>) [cs.NE]
- ↑ *James Bergstra, Guillaume Desjardins, Pascal Lamblin, Yoshua Bengio.* Quadratic polynomials learn better image features". Technical Report 1337 (<http://www.iro.umontreal.ca/~lisa/publications2/index.php/attachments/single/205>) (недоступная ссылка). *Département d'Informatique et de Recherche Opérationnelle, Université de Montréal* (2009). Дата обращения 30 сентября 2018. Архивировано (<https://web.archive.org/web/20180925121919/http://www.iro.umontreal.ca/~lisa/publications2/index.php/attachments/single/205>) 25 сентября 2018 года.
- ↑ Glorot, Bengio, 2010.
- ↑ Carlile, Brad; Delamarter, Guy; Kinney, Paul; Marti, Akiko & Whitney, Brian (2017-11-09), Improving Deep Learning by Inverse Square Root Linear Units (ISRLUs), *arXiv:1710.09967* (<http://www.arxiv.org/abs/1710.09967>) [cs.LG]
- ↑ По аналогии с диодом – пропускает ток (не меняя его) в одну сторону, и не пропускает в другую.
- ↑ Nair, Hinton, 2010, с. 807–814.
- ↑ Maas, Hannun, Ng, 2013.
- ↑ He, Zhang, Ren, Sun, 2015.
- ↑ Xu, Wang, Chen, Li, 2015.
- ↑ Clevert, Djork-Arné; Unterthiner, Thomas & Hochreiter, Sepp (2015-11-23), Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs), *arXiv:1511.07289* (<http://www.arxiv.org/abs/1511.07289>) [cs.LG]
- ↑ Klambauer, Unterthiner, Mayr, Hochreiter, 2017.
- ↑ Jin, Xiaojie; Xu, Chunyan; Feng, Jiashi; Wei, Yunchao; Xiong, Junjun & Yan, Shuicheng (2015-12-22), Deep Learning with S-shaped Rectified Linear Activation Units, *arXiv:1512.07030* (<http://www.arxiv.org/abs/1512.07030>) [cs.CV]
- ↑ Forest Agostinelli; Matthew Hoffman; Peter Sadowski & Pierre Baldi (21 Dec 2014), Learning Activation Functions to Improve Deep Neural Networks, *arXiv:1412.6830* (<http://www.arxiv.org/abs/1412.6830>) [cs.NE]
- ↑ *Xavier Glorot, Antoine Bordes, Yoshua Bengio.* Deep sparse rectifier neural networks (<http://proceedings.mlr.press/v15/glorot11a/glorot11a.pdf>) . *International Conference on Artificial Intelligence and Statistics* (2011).
- ↑ Elfwing, Uchibe, Doya, 2018.
- ↑ Godfrey, Gashler, 2016, с. 481–486.

- !4. Gashler, Ashmore, 2014.
- !5. Goodfellow, Warde-Farley, Mirza, Courville, Bengio, 2013, с. 1319–1327.

Литература

- *Hodgkin A. L., Huxley A. F.* A quantitative description of membrane current and its application to conduction and excitation in nerve // The Journal of Physiology. — 1952. — Т. 117, вып. 4. — С. 500–544. — PMID 12991237.
- *Simon S. Haykin.* [[8] (<http://books.google.com/books?id=bX4pAQAAMAAJ>) в «Книгах Google» Neural Networks: A Comprehensive Foundation]. — Prentice Hall, 1999. — ISBN 978-0-13-273350-2.
- *Cybenko G.V.* [[9] (<http://books.google.com/books?id=4RtVAAAAMAAJ>) в «Книгах Google» Approximation by Superpositions of a Sigmoidal function] // Mathematics of Control, Signals, and Systems / Jan H. van Schuppen. — Springer International, 2006. — С. 303.
- *Jan Snyman.* [[10] (http://books.google.com/books?id=0tFmf_UKl7oC) в «Книгах Google» Practical Mathematical Optimization: An Introduction to Basic Optimization Theory and Classical and New Gradient-Based Algorithms]. — Springer Science & Business Media, 2005. — ISBN 978-0-387-24348-1.
- *Huaiqin Wu.* Global stability analysis of a general class of discontinuous neural networks with linear growth activation functions (<http://linkinghub.elsevier.com/retrieve/pii/S0020025509002539>) // Information Sciences. — 2009. — Т. 179, вып. 19. — С. 3432–3441. — doi:10.1016/j.ins.2009.06.006 (<https://dx.doi.org/10.1016%2Fj.ins.2009.06.006>).
- *Xavier Glorot, Yoshua Bengio.* Understanding the difficulty of training deep feedforward neural networks (<http://jmlr.org/proceedings/papers/v9/glorot10a/glorot10a.pdf>) // International Conference on Artificial Intelligence and Statistics (AISTATS'10). — Society for Artificial Intelligence and Statistics, 2010.
- *Vinod Nair, Geoffrey E. Hinton.* Rectified Linear Units Improve Restricted Boltzmann Machines (<http://dl.acm.org/citation.cfm?id=3104322.3104425>) // 27th International Conference on International Conference on Machine Learning. — USA: Omnipress, 2010. — С. 807–814. — (ICML'10). — ISBN 9781605589077.
- *Andrew L. Maas, Awni Y. Hannun, Andrew Y. Ng.* Rectifier nonlinearities improve neural network acoustic models (<https://pdfs.semanticscholar.org/367f/2c63a6f6a10b3b64b8729d601e69337ee3cc.pdf>) // Proc. ICML. — 2013. — Июнь (т. 30, вып. 1).
- *Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun.* Proceeding ICCV'15 Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV). — Washington: IEEE Computer Society, 2015. — С. 1026–1034. — ISBN 978-1-4673-8391-2.
- *Bing Xu, Naiyan Wang, Tianqi Chen, Mu Li.* Empirical Evaluation of Rectified Activations in Convolutional Network // Computer Vision and Pattern Recognition. — 2015.
- *Günter Klambauer, Thomas Unterthiner, Andreas Mayr, Sepp Hochreiter.* Self-Normalizing Neural Networks // Advances in Neural Information Processing Systems. — 2017. — Июнь (т. 30, вып. 2017). — . — arXiv:1706.02515.
- *Stefan Elfving, Eiji Uchibe, Kenji Doya.* Sigmoid-Weighted Linear Units for Neural Network Function Approximation in Reinforcement Learning // Neural Networks. — 2018.

- *Luke B. Godfrey, Michael S. Gashler*. A continuum among logarithmic, linear, and exponential functions, and its potential to improve generalization in neural networks // 7th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management: KDIR. — 2016. — Февраль (т. 1602). — . — [arXiv:1602.01321](#).
- *Michael S. Gashler, Stephen C. Ashmore*. Training Deep Fourier Neural Networks To Fit Time-Series Data // International Conference on Intelligent Computing. — Springer, Cham, 2014. — С. 48-55.
- *Ian J. Goodfellow, David Warde-Farley, Mehdi Mirza, Aaron Courville, Yoshua Bengio*. Maxout Networks // JMLR Workshop and Conference Proceedings. — 2013. — Т. 28, вып. 3. — С. 1319-1327. — . — [arXiv:1302.4389](#).

Источник — https://ru.wikipedia.org/w/index.php?title=Функция_активации&oldid=106534546

Эта страница в последний раз была отредактирована 23 апреля 2020 в 10:50.

Текст доступен по лицензии Creative Commons Attribution-ShareAlike; в отдельных случаях могут действовать дополнительные условия.

Wikipedia® — зарегистрированный товарный знак некоммерческой организации Wikimedia Foundation, Inc.