АКТУАЛНИ ТЕНДЕНЦИИ В МОДЕЛИРАНЕТО НА НЕВРОННА АКТИВАЦИЯ

ас. д-р Александър Иванов
Бургаски свободен университет

RECENT TRENDS IN NEURAL ACTIVATION MODELLING

assistant prof. Alexander Ivanov, PhD
Burgas Free University

Abstract: *In this paper a list of recent proposals for neural activation functions is presented. The activation functions are discussed in terms of classification accuracy, convenience, performance and biological plausibility.*

Keywords: *artificial neural network, activation functions, sigmoid functions.*

I. Увод

Изкуствените невронни мрежи са водеща технология в изкуствения интелект през последните десетилетия. Един от ключовите аспекти при моделирането на невронни процеси чрез средствата на информатиката е моделирането на невронна активация. В миналото е търсено максимално сходство с биологичните процеси, но предвид съвременната употреба на тези технологии в информатиката днес се разработват и функции, фокусирани върху изчислителното ускорение на използваните модели на невронни мрежи. В настоящата статия е представен кратък обзор на някои актуални (към 2022г.) развития в моделирането на активация в изкуствените невронни мрежи.

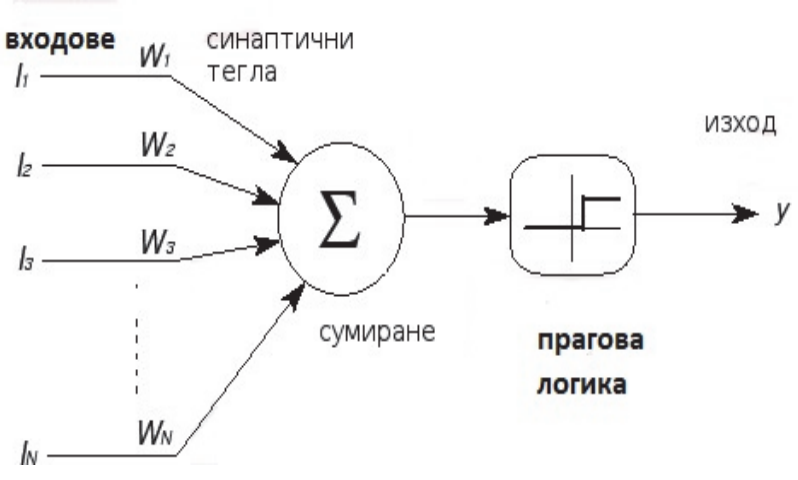
II. Дефиниции

Изкуствен неврон се представя като изчислителна единица с входове (дендрити) и изход (аксон), която извършва сумиране на претеглени входни сигнали и активация за изчисляване на изходната стойност. По същността си активацията е прагова логика – ако сумата от претеглените входове надвишава даден праг, невронът се активира. В оригиналния модел на МакКълък и Питс праговата логика се моделира чрез стъпковата функция. Тази функция има няколко недостатъка:

1) позволява използването само на двоични стойности на изхода, което ограничава възможната употреба на технологията и не съответства на биологичните процеси

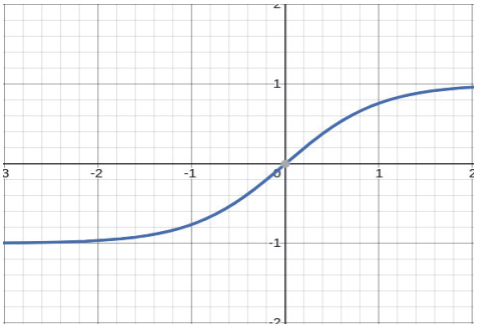
2) преходът от 0 към 1 е импулсен (рязък)

3) функцията не е непрекъснато диференцируема, което ограничава възможностите за обучение на невроните. За преодоляване на тези проблеми стъпковата функция е заменена с логистичната сигмоидална функция, която моделира плавен преход между асимптотичните граници 0 и 1 и е непрекъснато диференцируема. Като алтернативи на стъпковата и съответстващата и' логистична функция също се използват знаковата функция и съответно хиперболичния тангенс, чиито обхвати са от -1 до 1. На фиг. 1 е представена схема на изкуствен неврон.

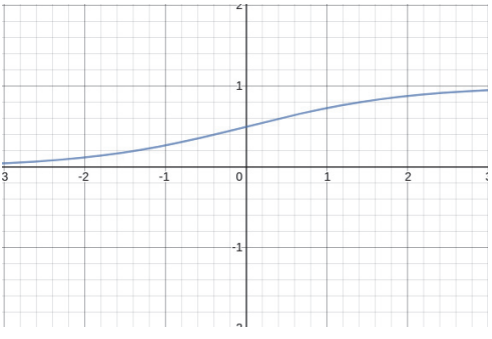


Фиг. 1. *Схема на изкуствен неврон*

На фиг. 2 а) б) са представени графики на класическите активационни функции.



Фиг. 2 а) *Хиперболичен тангенс*



Фиг. 2 б) *Логистична функция*

III. Актуални развития в моделирането на невронна активация

След известен период на застой в развитието на активационните функции в последните няколко години се наблюдава възраждане на интереса към темата. В периода от 2015 г. до 2022 г. са предл ожени десетки нови активационни функции. В Таблица 1 е систематизиран неизчерпателен списък на активационни функции, подредени по година на формулиране.

Функция	Година	Автор	Формула	Производна
Tanh	1760 [1]	Ricatti, Lambert	$\frac{e^x - e^{-x}}{e^x + e^{-x}}$	$\frac{cosh^2 x - sinh^2 x}{cosh^2 x}$
Logistic	1840 [2]	Verhulst	$\frac{1}{1+e^{-x}}$	$\sigma(x)\big(1-\sigma(x)\big)$
Heaviside step	1892 [3]	Heaviside	$\begin{cases}1, x>0\\0, x\le 0\end{cases}$	$\delta(x)$ [4]
Softmax	1989	Bridle	$\frac{e^{x_i}}{\sum_{j=1}^{K} e^{x_j}}, i=1,2,\ldots,K$ [5]	$x_i\big(1\{i=j\}-x_j\big)$
Softsign	2009	Turian	$\frac{1}{1+\lvert x\rvert}$	$\frac{1}{(1+\lvert x\rvert)^2}$
ReLU	2010	Hinton, Nair	$max(0,x)$	$step(x)$
ReLU6	2010	Hinton et al.	$min(max(0,x),6)$	$ln(1+e^x)$
Leaky ReLU	2013	Maas et al.	$\begin{cases}x, x>0\\ax, x\le 0\end{cases}$	$\begin{cases}1, x>0\\a, x\le 0\end{cases}$
Parametric ReLU	2015	He et al.	$\begin{cases}x, x>0\\ax, x\le 0\end{cases}$ [6]	$\begin{cases}1, x>0\\a, x\le 0\end{cases}$
Softplus	2001		$ln(1+e^x)$	$logistic(x)$
ELU	2016	Clevert et al.	$\begin{cases}x, x>0\\a(e^x-1), x\le 0\end{cases}$	$\begin{cases}1, x>0\\ae^x, x\le 0\end{cases}$
Swish	2017	Ramachandran et al.	$\frac{x}{1+e^{-\beta x}}$	$swish(x)+\sigma(x)\big(1-swish(x)\big)$ [7]
SiLU	2017	Elwing et al.	$\frac{x}{1+e^{-x}}$	$SiLU(x)+\sigma(x)\big(1-SiLU(x)\big)$
Fractional logistic	2018	Ivanov	$\frac{1}{1+E_{\alpha,\beta}(-x)}$ [8]	$\frac{E_{\alpha,\beta}(x)}{\big(1+E_{\alpha,\beta}(x)\big)^2}$
GELU	2020	Hendrycks et al.	$xP(x\le X)$ [9]	$\phi(x)+xP(X=x)$
PAU	2020	Molina et al.	$\frac{P(X)}{Q(X)}$	$\frac{\partial P(x)}{\partial xQ(x)}$
Mish	2020	Misra	$xtanh\big(softmax(x)\big)$	$e^x\,\omega/\delta^2$ [10]
ACON	2021	Ma et al.	$\frac{\sum_{i=1}^{n} x_i\, e^{\beta x_i}}{\sum_{i=1}^{n_i} e^{\beta x_i}}$	-
Serf	2021	Nag et al.	$\left(\frac{2}{\sqrt{\pi}}\right)e^{-ln\left(\left(1+e^{x}\right)\right)^2}x\sigma(x)+\left(\frac{f(x)}{x}\right)$	-
FALU	2022	Zamora et al.	$D^{\alpha}x\sigma(\beta x)$ [11]	-
ErfAct	2022	Biswas et al.	$erf\big(ae^{\beta x}\big)$ [12]	-
Pserf	2022	Biswas et al.	$erf\left(\gamma ln\left(1+e^{\delta x}\right)\right)$ [13]	- [14]

Таблица 1. *Списък на активационни функции*

[1] [2] [3] Години на постулиране

[4] Дирак делта функция

[5] K е брой входове от входния вектор от стойности на невронната мрежа (x_1, x_2, …, x_n),

[6] Параметричната ReLU има същата формула като leaky ReLU, разликата е че при първата параметърът α подлежи на обучение.

[7] σ(x) обозначава логистичната функция

[8] $E_{\alpha,\beta}(x)$ обозначава функцията на Миттаг-Лефлер

[9] $X \sim N(\mu, \sigma^2)$, където $N(\mu, \sigma^2)$е нормално разпределение със средна стойност μ и дисперсия σ ^{2} , а $\phi(x) = \left(1 + erf\left(\frac{x}{\sqrt{2}}\right)\right)/2$

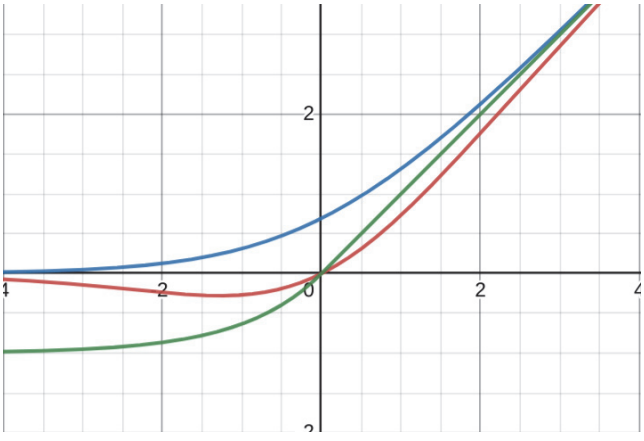
[10] $\omega = e^{3x} + e^{2x}(4) + e^{x}(6 + 4x) + 4(1 + x), \delta = (ex + 1)^2 + 1$

[11] D^{α} обозначава производна от дробен ред, α е редът на производната

[12, 13] $erf(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2}\, dt$, α, β, γ и δ са параметри

[14] Производните на последните функции не са посочени за прегледност на таблицата

В изброените формули аргументът е отбелязан с *x*, но в контекста на невронна активация входната стойност обикновено е сумата на претеглените входове *XW*; *W* е тегловата матрица, моделираща силата на синаптичните връзки между невроните; *X* е векторът от стойности на входните променливи. На Фиг. 3 са представени графики на 3 от изброените по-горе функции.



Фиг. 3. *Графики на функциите Softplus (синьо), Swish (червено) , ELU (зелено)*

През 2010 Хинтън и Наир [1] предлагат активационната функция ReLU, която е частично непрекъсната. Функцията занулява отрицателните стойности на аргумента, а върху положителните стойности изпълнява функцията на идентичност. В последното десетилетие тази активационна функция е водеща за много различни архитектури, макар че оригинално е предложена за машини на Болцман. Предимство на тази функция е че няма горна граница, което увеличава многократно обхвата на възможните стойности, но все пак функцията не е лишена от проблеми. Проблем е например, че за отрицателни стойности във входните данни е възможно определени неврони на практика да бъдат изключени от информационната обработка (т. нар. проблем за „умиращата ReLU“ –”dying ReLU”). Проблемът се адресира чрез последващите модификации Leaky ReLU, Parametric ReLU, ELU, GELU и др. През 2017 Prajit Ramachandran et al. предлагат функцията Swish [2], която е подобрение на SiLU. [3] Според представените в статията експерименти функцията подобрява точността на класификация при разпознаване на образи. През 2021 Sayan Nag и Mayukh

Bhattacharyya предлагат функцията SERF (log-Softplus ERror Function) [4]. Функцията е ограничена отдолу, гладка, немонотонна и диференцируема. Бидейки неограничена отгоре, функцията избягва т.нар. проблем на насищането. Производната и‘ е

$$f'(x) = \left(\frac{2}{\sqrt{\pi}}\right) e^{-ln\left((1+e^x)\right)^2} x\sigma(x) + \left(\frac{f(x)}{x}\right) = p(x)swish(x) + \frac{f(x)}{x}, \quad (1)$$

където *σ(x)* е логистичната функция, а *p(x)* е т.нар. предварителна функция (precondition function), чиято цел е да ускори конвергенцията в алгоритмите за спускане по градиента. В статията [4] Serf е сравнена с функциите Swish и Mish при задача за разпознаване на образи чрез конволюционна невронна мрежа. Резултатите индикират подобрение на точността при използването на Serf. През 2021 Biswas предлага подобрен вариант с 2 параметъра [5], наречен Pserf:

$$f(x;\gamma,\delta) = erf\left(\gamma ln\left(1 + e^{\delta x}\right)\right), \quad (2)$$

където $erf(x) = \frac{2}{\sqrt{\pi}}\int_0^x e^{-t^2}\,dt$.

През 2020 е предложена функцията GELU [6], която използва кумулативна функция на нормално разпределение:

$$f(x) = xP(x \le X),\ X \sim N(\mu,\sigma^2), \quad (3)$$

където $N(\mu,\sigma^2)$ е нормално разпределение със средна стойност μ и дисперсия σ^2. Стойностите на функцията могат да се изчисляват по формулата

$$f(x) = \frac{1}{2}x\left(1 + erf\left(\frac{x}{\sqrt{2}}\right)\right). \quad (4)$$

Производната на GELU се дава с формулата

$$f'(x) = \phi(x) + xP(X = x), \phi(x) = \frac{1+erf\left(\frac{x}{\sqrt{2}}\right)}{2}. \quad (5)$$

При изброените разработки целта е да се предложат алтернативи на ReLU, които са непрекъснато диференцируеми, не занулява отрицателни стойности, изчисляват се без излишно натоварване на хардуера и подобряват точността на класификацията.

Друга тенденция в разработването на активационни функции е търсенето на биологична правдоподобност. През 2016 Лиу и др. [7] предлагат вариант на функцията Softplus с наличие на шум – Noisy Softplus, което моделира поведението на LIF (Leaky Integrate-and-Fire) невроните. Предложената активационна функция е тествана в конволюционна невронна мрежа за класификация. В [8] е предложена функцията Rand Softplus, която подобно на Noisy Softplus моделира шума, наличен в биологичните неврони. Според авторите наличието на стохастика в активационната функция прави изкуствената невронна мрежа по-адаптивна към шум във входните данни. Хипотезата е тествана с ResNet, която е вид конволюционна мрежа за класификация на изображения, и резултатите са в подкрепа на хипотезата. Двете модификации на функцията Softplus са опит да се замени функцията на Siegert, която моделира активацията на неврони в пробивните невронни мрежи (Spiking neural networks). Изброените в този параграф функции са биологично правдоподобни.

Една от тенденциите в разработването на нови активационни функции е формулирането на дробни еквиваленти на класическите функции като логистичната и хи-

перболичния тангенс. Дробното диференциране и интегриране са дял от анализа, който в последните 50 години намира широко приложение в практиката, особено във физиката. Съществуват различни дефиниции за дробни производни и интеграли и няма консенсус за употребата им. Въпреки това обаче различни дефиниции се оказват полезни в практиката. Една функция, която намира универсално приложение в този дял на математиката, е функцията на Миттаг-Лефлер, която обобщава експоненциалната функция. Тази функция се появява като решение на диференциални уравнения с производна от дробен ред. През 2018 Иванов предлага дробната логистична функция [9] с формула

$$flogistic(x) = \frac{1}{1+E_{\alpha,\beta}(-x)}, \quad (6)$$

където$E_{\alpha,\beta}(x)$ е гореспоменатата функция на Миттаг-Леффлер. Функцията е тествана практически в проблеми за класификация и предварителните тестове показват, че би могло да се намерят настройки на параметрите на активационна функция, които подобряват точността на класификацията. В [10] функцията е тествана и с конволюционни невронни мрежи. Три предимства на тази функция в машинното обучение са:

1) притежава 2 параметъра, които подлежат на настройка и дори обучение
2) по-подходяща е за моделиране на определен тип проблеми
3) може да улесни обучението при използване на алгоритми с дробни производни. Недостатък на функцията е изчислителната цена, но проблемът може да се смекчи като се използват приближения.

В [11] авторът предлага адаптивно изчисляване на конкретна активационна функция за даден проблем чрез използване на дробен интеграл от класическите активационни функции – логистична, ReLU и хиперболичен тангенс. Редът на интеграла е параметър, който се адаптира към спецификата на проблема. Подходът е тестван в конволюционни невронни мрежи за разпознаване на изображения. [5]

През 2022 в [12] Zamora предлага Fractional Adaptive Linear Unit (FALU). Дефиницията и' се дава с формулата$f(x) = D^{\alpha}x\sigma(\beta x)$, където D^{α} е оператор за диференциране, α е редът на производната, β е мащабиращ параметър, σ е активационната функция swish. FALU е еквивлентна на swish при стойности на параметрите $\alpha=0$, $\beta=1$. Използваният оператор за диференциране е със следната дефиниция:

$$f(x) = x^k, D^{\alpha}f(x) = \frac{\Gamma(k+1)}{\Gamma(k+1-\alpha)}x^{k-\alpha}. \quad (7)$$

В статията [12] са дадени приближения на FALU, които намаляват изчислителната цена на използването на функцията. Функцията е тествана практически в конволюционна мрежа за разпознаване на изображения, като показва известно малко подобрение в точността спрямо класическия еквивалент. В [13] е разгледан обучителният алгоритъм „обратно разпространение на грешката" (backpropagation of error) във вариант с дробно диференциране, като е използвана конформната производна на класическата логистична функция вместо класическата целочислена производна. В [14] се разглеждат дробните варианти на ReLU, leaky ReLU, Parametric ReLU.

Тенденция, която не бива да се пренебрегва, е формулирането на активационни функции, подлежащи на обучение. Използването на мета-обучителни алгоритми за търсене на оптимални невронни архитектури (NAS) и хиперпараметри насочва учените към разработването на функции, чиято форма зависи от параметри, подлежащи на обучение. Разработки в тази посока са функциите PAU и ACON. И двете функции използват апроксимиращи полиноми.

Pade Aproximation Unit (PAU) е функция, която използва апроксимиращ рационален полином F(x) , частно на две суми [15].

Pade aproximant:

$$F(x) = \frac{P(x)}{Q(x)} = \frac{\sum_{j=1}^{m} a_j x^j}{1+\sum_{j=1}^{n} b_k x^k} = \frac{a_0+a_1x+a_2x^2+\ldots+a_mx^m}{1+b_1x+b_2x^2+\ldots+b_nx^n} \quad (8)$$

Авторите на статията [15] предлагат схема на обучение, при която параметрите *a* и *b* могат да бъдат научени автоматично и да бъдат различни за всеки слой в невронната мрежа.

В статията [16] е предложена активационната функция Activate-Or-Not (ACON), която се дефинира с апроксимиращ параметричен полином.

Общият вид на функцията е:

$$S_\beta(x_1, x_2, \ldots, x_n) = \frac{\sum_{i=1}^{n} x_i e^{\beta x_i}}{\sum_{i=1}^{n_i} e^{\beta x_i}} \quad (9)$$

Този полином е алтернатива на функциите от фамилията MaxOut, които изчисляват максимума между 2 стойности (ReLU е представител на тази фамилия функции). При ACON двете стойности, изчислявани от MaxOut, заместват x_i във формулата (9). Използвайки стойностите от класическата ReLU, PreLU и max(p_1 x, p_2 x), получаваме трите функции ACON-A, ACON-B и ACON-C :

ACON-A:

Swish

ACON-B:

$$(1-p)x\sigma(\beta(1-p)x) + px \quad (10)$$

ACON-C:

$$(p_1-p_2)x\sigma(\beta(p_1-p_2)x) + p_2x \quad (11)$$

Функциите PAU, ACON и дробна логистична са примери за активационни функции с адаптивни параметри.

В статията [17] авторите правят задълбочен анализ и сравнение на много широк набор от активационни функции. Функциите се анализират по групи и се сравняват в практически в задачи за разпознаване на образи и обработка на естествени езици.

IV. Заключение

Забелязват се няколко тенденции при предложенията за нови активационни функции:

1) Адресират се проблеми при класическите активационни функции, като изчезващия/експлодиращия градиент и др.
2) Търсят се биологично правдоподобни функции, не само за нуждите на информатиката, но и в изчислителната невронаука
3) Има интерес към формулирането на функции, свързани с дробното диференциране и интегриране

4) Основен акцент е използването на функции с параметри, подлежащи на обучение, което прави невронните мрежи по-адаптивни към различен контекст.

Активационните функции са активна област на научни изследвания, която подобрява съществуващите модели на изкуствени невронни мрежи.

Литература:

[1] Nair, Vinod & Hinton, Geoffrey. Rectified Linear Units Improve Restricted Boltzmann Machines Vinod Nair. *Proceedings of ICML.* 27. 807-814, 2010

[2] Prajit Ramachandran, Barret Zoph, Quoc V. Le, Searching for Activation Functions, https://doi.org/10.48550/arXiv.1710.05941 , 2017

[3] Elfwing, S., et al., Sigmoid-Weighted Linear Units for Neural Network Function Approximation in Reinforcement Learning, doi:1702.03118v3, 2017

[4] Nag, S. et al., SERF: Towards better training of deep neural networks using log-Softplus ERror activation Function, arXiv:2108.09598, 2021

[5]. Biswas, K., et al., ErfAct and Pserf: Non-monotonic Smooth Trainable Activation Functions. *Proceedings of the AAAI Conference on Artificial Intelligence,* 36(6), 6097-6105. https://doi.org/10.1609/aaai.v36i6.20557 , 2021

[6] Hendrycks, D., Gimpel, K., Gaussian error linear units (gelus), arXiv preprint arXiv:1606.08415, 2016

[7] Liu, Q., Furber, S. Noisy Softplus: A Biology Inspired Activation Function. In: Hirose, A., Ozawa, S., Doya, K., Ikeda, K., Lee, M., Liu, D. (eds) *Neural Information Processing. ICONIP 2016*. Lecture Notes in Computer Science(), vol 9950. Springer, Cham. https://doi.org/10.1007/978-3-319-46681-1_49, 2016

[8] Chen, Y., et al., Improving the Antinoise Ability of DNNs via a Bio-Inspired Noise Adaptive Activation Function Rand Softplus. Neural Computation 2019; 31 (6): 1215–1233. doi: https://doi.org/10.1162/neco_a_01192, 2019

[9] Ivanov, A., „Fractional activation functions in feedforward artificial neural networks," 20th International Symposium on Electrical Apparatus and Technologies (SIELA), 2018, pp. 1-4, doi: 10.1109/SIELA.2018.8447139., 2018

[10] Ivanov, A., Georgieva, P. Класификация с конволюционни невронни мрежи. *Компютърни науки и комуникации,* 7(1), 46-52., 2018

[11] Zamora-Esquivel, J., A. Cruz Vargas and P. Lopez-Meyer, „Fractional Adaptation of Activation Functions In Neural Networks," *2020 25th International Conference on Pattern Recognition (ICPR)*, pp. 7544-7550, doi: 10.1109/ ICPR48806.2021.9413338., 2020

[12] Zamora-Esquivel, J. et al., Fractional adaptive linear units, T*he Thirty-Sixth AAAI Conference on Artificial Intelligence,* 2022

[13] Alta, G., Fractional-Order Activation Function for Feed Forward Neural Networks using Conformable Derivative, Journal of Artificial Intelligence with applications, 11-14, 2021

[14] Job, M. et al., Fractional rectifying linear unit activation function and its variant, *Mathematical problems of engineering,* Volume 2022, Article ID 1860779, doi:10.1155/2022/1860779, 2022

[15] Molina, A. et al., PADÉ activation units: end-to-end learning of flexible activation in deep networks , *Conference paper ICLR*, 2020

[16] Ma, N.. et al., Zhang, X., et al., Activate or Not: Learning Customized Activation, arXiv:2009.04759v9, 2021

[17] Dubey, s. R., Singh, S., Chaudhuri, B. B., Activation Functions in Deep Learning: A Comprehensive Survey and Benchmark, 2022