FRI-ONLINE-1-CCT1-08

COMPARATIVE ANALYSIS OF ACTIVATION FUNCTIONS USED IN DEEP NEURAL NETWORKS TRAINING[8]

Martin Kaloev, PhD Student
Department of EEA
University of Ruse
E-mail: kaloev_92@mail.ru

Abstract: *The article discusses the training of a neural network for curve recognition. Three popular activation functions are analyzed - logistic* (*sigmoid*) *function* (*Sigmoid*), *hyperbolic tangent* (*tanh*) *and corrective linear function* (*ReLU, Rectified linear unit*)

The activation function is selected so that the input signal can accept arbitrary values and the output values are in a strictly limited range. However, although the input values can be arbitrary, a saturation effect occurs when the element is sensitive only to input values lying in a limited range. This condition is fulfilled by functions similar to the letter S (*eg Sigmoid*). *for them, the output value is in the range* (*0,1*), *and the sensitivity range for the input is slightly wider than the range* (*-1, + 1*). *This function is smooth and its derivative is easy to calculate. This is important when training the network*

The second activation function studied is the hyperbolic tangent, which is a traditional activation function, but which suffers from the problem of disappearing gradients.

The linear correction function (*ReLU*) *has been used relatively recently. It is especially suitable for working with deep neural networks, as it does not suffer from the problem of disappearing gradients. However, ReLU also has drawbacks. The function is differentiable at 0 and in rare situations can lead to the fact that part of the network will "crash" and will not participate in the next iterations of training.*

The results of this study will be useful in the design of Deep Neural Networks and in the selection of activation functions.

Keywords: *gradient decent, tensorflow, keras, Relu, Sigmoid, Tanh, activation function analysis, activation function comparison, curve graphics in artificial neural network, model layers, architecture in deep artificial neural network, starved batch methods for neural network*

JEL Codes: *L10, L11*

ВЪВЕДЕНИЕ

Изкуствените невронни мрежи мога се представят като система от неравенства. Като всеки слой е уравнение със своя крива в декартово пространство. Тази статия изследва използването на три популярни функции като активационни за скрит слой. Функциите са Сигноидна, Тангесова хиперболична, Relu. Сигноидна функция е логистична функция с константа "е" ,която се използва за функция на последен слой. Релу е линейна функция ,която е най-популярната функция за скрити слоеве. Хиперболичната тангесова функция се използва като функция в дълбоки неврони мрежи, и дълбоки филтърни мрежи.

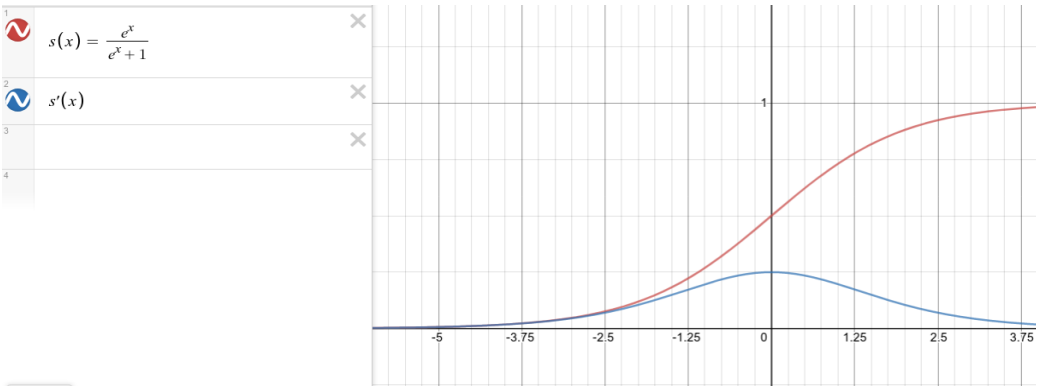
ИЗЛОЖЕНИЕ

Ролята на градиентно спускане в невронните мрежи

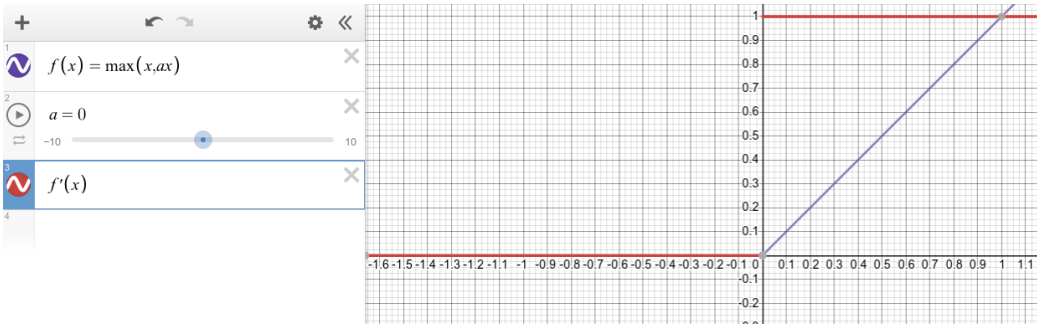
Градиентно спускане е метод които се използва за изчисляване на разминаването между криви в координатна система. При невроните мрежи този метод се прилага за кривите описани от изходните стойностите на всеки скрит слой и изходния слой. Чрез градиентно спускане се преизчисляват стойностите на всички тежести и отклонения. Като общо правило: за невронна мрежа винаги е положително да има вход с известна случайност (Haykin, S., 2008).

[8] Докладът е представен на сесия FRI-ONLINE-1-CCT1 на 13 ноември 2020 г. с оригинално заглавие: COMPARATIVE ANALYSIS OF ACTIVATION FUNCTIONS USED IN DEEP NEURAL NETWORKS TRAINING

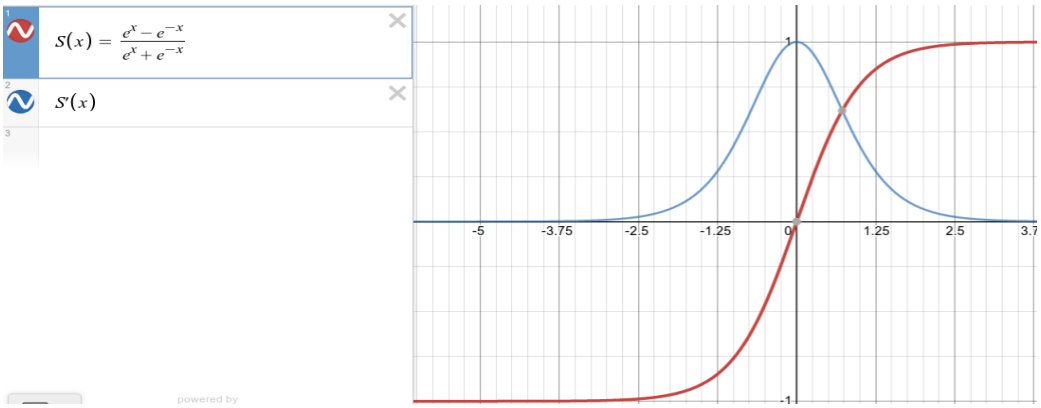
Активационни функции



Фиг. 1: Сигмоидна активационна функция



Фиг. 2: ReLu активационна функция

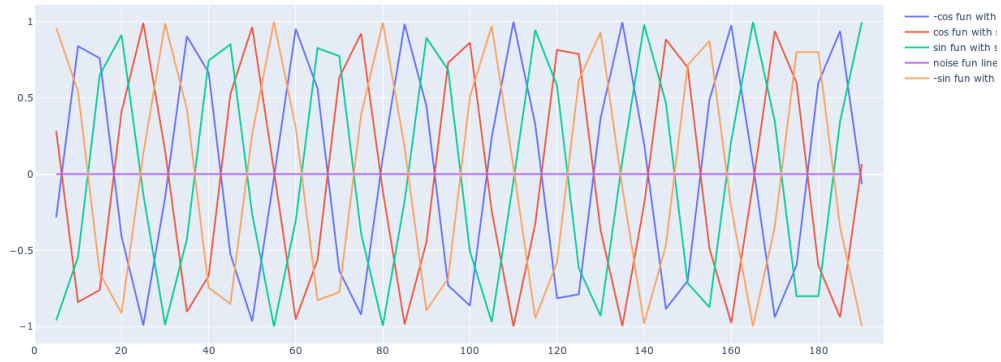


Фиг. 3: Хиперболична тангесова активационна функция

Сигмоидалната функция е математическа функция, имаща характерна "S" -образна крива или сигмоидна крива. Често срещан пример за сигмоидна функция е логистичната функция (Han, J. and Moraga, C., 1995) ReLu е тип линейна функция, в съвременните невронни мрежи от дълбок вид. Основния недостатък на този тип функция са появата на мъртви неврони, които връщат ‘0’, независимо от стойността на преходните слоеве. Може да видим този ефект илюстриран в следващите тестове. Функция на хиперболичния тангенс е тип логистична функция която намира приложение в неврони мрежи и дълбоки филтърни мрежи.

Проектиране на невронна мрежа

Проектира се невронна мрежа способна да разпознае 5 отделни криви по принадлежност описани във фигура 4.



Фиг. 4: Криви описани от sin, -sin, cos, -cos функци за стойностите от 0 до 200 със стъпка 5

Стойностите по дължина са от 0 до 200, а стойностите по височина са стойностите на функцията за всяка стъпка.

Проектиране на матица с тренировачни данни

В таблица 1 са описани тренировъчните данни използвани за обучение чрез формула. Таблицата е условно разделена на 3 колони. Колона 1 съдържаща входни данни за 1ви до 20ти входен възел на входния слой. колона 2 съдържаща входни данни за 21 до 40ти хводен възел на хводния слой. Колона 3 която съдържа класификация на изходни данни. колона 1 съдържа стойности от 0 до 200, а колона 2 съдържа функцията от тези стойности.

Таблица 2: Таблица с тренировъчни данни

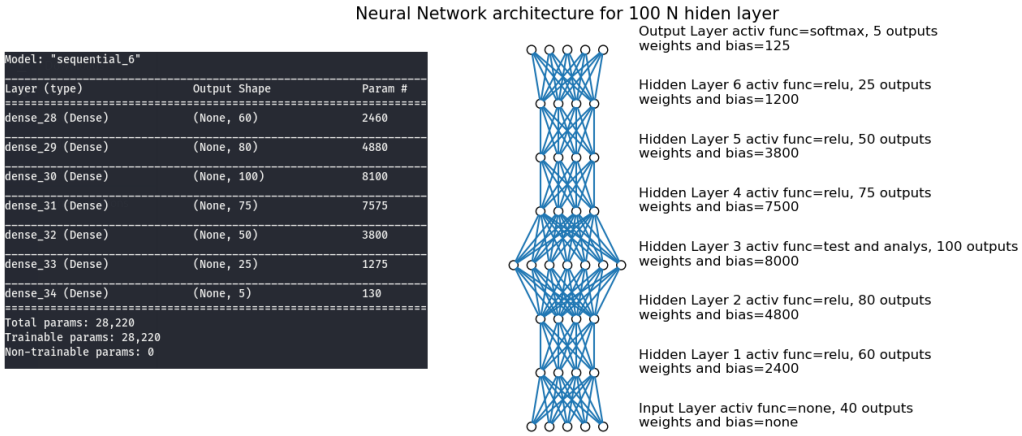
Input Layer N1..N20	Input Layer N21...N40	Classification
x1=0...x20=95	fsin(x1)...fsin(x20)	1,0,0,0,0
...	..	...
x80...x100	fsin(x80)..fsin(x100)	1,0,0,0,0
x1...x20	f-sin(x1)...f-sin(x20)	0,2,0,0,0
...	..	..
xN	fcos(xN)	0,0,3,0,0
xN	f-cos(xN)	0,0,0,4,0
xN	0...0...0	0,0,0,0,5

Проектиране и избор на брой на слоеве

Избора на брой на слоеве на невронната мрежа трябва да отговаря на няколко изисквания. Скрит слой с 100 възела които да се изследва, за да илюстрира ефекта на избора на активационни функции. Невронна мрежа способна да обучи от данни от фигура 4 и таблица 1, да отговаря на уравнение от фигура 5, (Reed, R.D. and J, R. , 1999).

$$Nh = \frac{Ns}{(a * (Nii + No))}$$

Където Nh е броя на скритите слоеве. Ns е броя на тренировъчните данни. Ni е броя на входните възли. No е броя на изходните възли. А а е коефициента на мащабиране на тренировъчните данни.



Фиг. 6: Архитектура на невронна мрежа

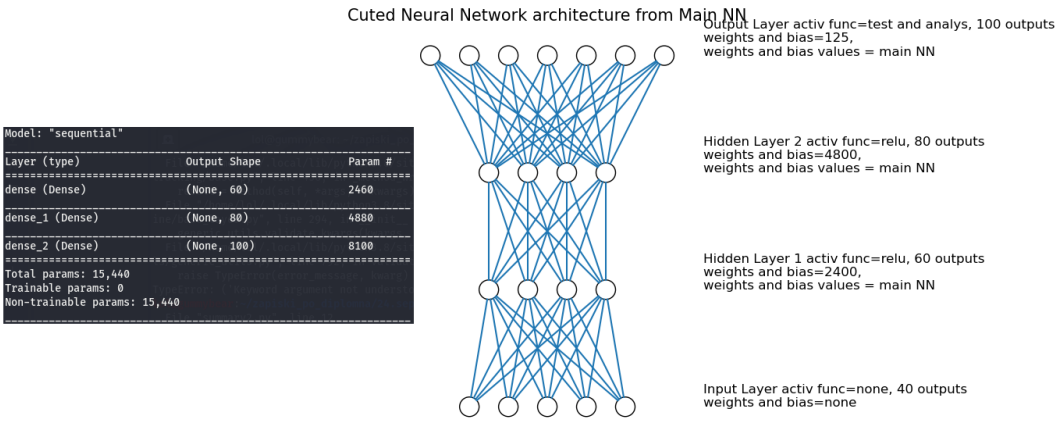
Невронната мрежа има 8 слоя, 435 неврона, 27765 тежести и отклонения. Входен слой с 40 неврона и изходен слой 5 неврона. Стойностите на възлите в скритите неврони се изчисляват по време на развиване на функцията напред в невронната мрежа. Стойностите на тежестите и отклоненията за всеки слой се променят след преизчисление на градиентното спускане в края на всеки обучителен цикъл.

Първи метод за извличане на стойност вектор на скрит слой

Програмира се функция която да връща стойности на скрития слой, след като мрежата е обучена.

Втори метод за извличане на стойност вектор на скрит слой

Създава се невронна мрежа която няма изходен слой, не е способна да се обучава. Тази мрежа приема стойностите за тежести и отклонения на оригиналната невронна мрежа. Предимствата на този метод са драстично намалено време за смятане на стойности на слоевете защото се извършва единството един цикъл от операции само в една посока.

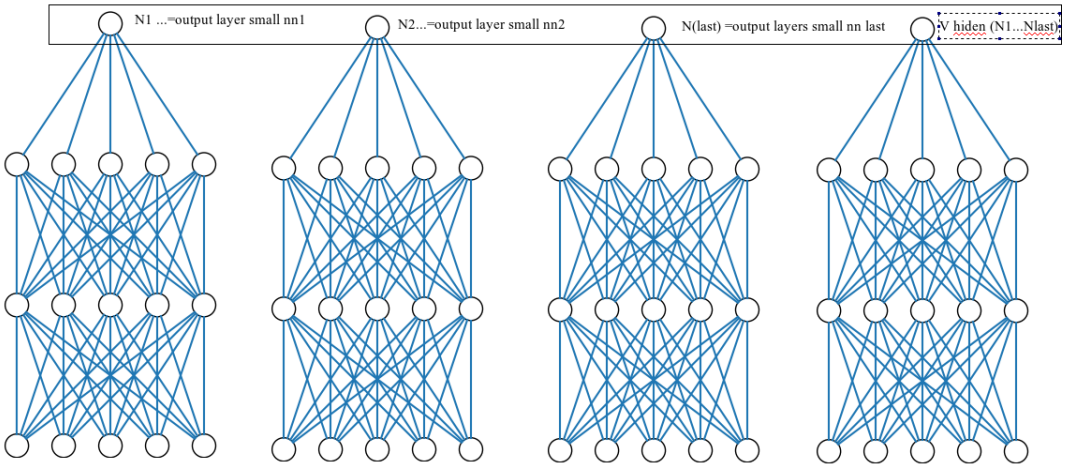


Фиг. 7: Архитектура на частична невронна мрежа

Частичната невронна мрежа има по-малко слоеве и вместо изходен слой завърша със слой които има същият размер като скрития слой от оригиналната мрежа. Стойностите получени от този слой са еднакви със стойностите в скрития слой на оригиналната изследвана мрежа.

Трети метод за извличане на стойност вектор на скрит слой

Представяне на изходен вектор на невронна мрежа като сбор от невронни мрежи с единствен изходен слой.



Фиг. 8: Архитектура на множество мрежи обединени в един вектор

Този метод се обяснява чрез уравнението от фигура 9. Сумата от числата във вектор е равна на средната стойност на от сумата от числата умножена по броя на числата. От това следва. Сумата на числата във Вектор V1 които има стойностите на неврона мрежа с много изходи е равна сумата на числата във вектор V2 които има стойност на невронна мрежа с един изход, умножена по броя на изходите във вектор V2. Средната стойност от вектор в слой на невронна мрежа е винаги функция на преходния слой от вектор със стойности на слоя преди него.

$$\sum(V1) = dim(V1) * \overline{V1}(1)$$

$$\sum(V2) = dimV2 * \overline{V2} * It(2)$$

$$\overline{V1} = funL2(\forall(L2))(3)$$

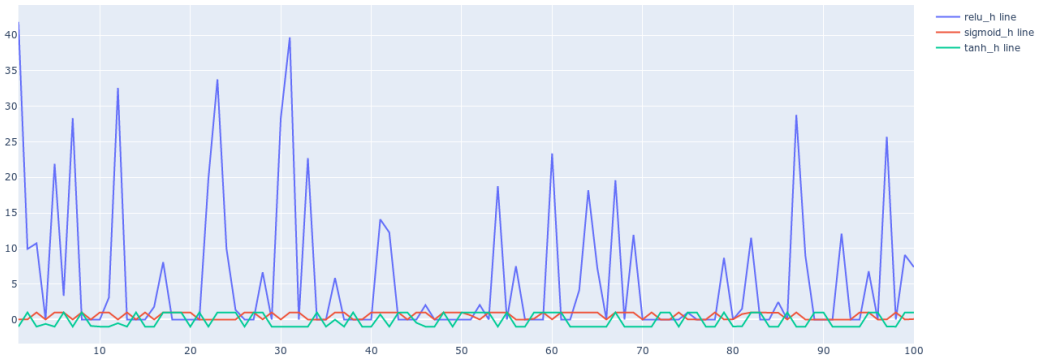
$$\overline{V2} = funL2(\forall(L2))(4)$$

$$\frac{fnn(ffull(V1))}{fnn(fcuted(V2)) * It} \simeq \frac{\sum(V1)}{\sum(v2)}(5)$$

Практическото тестване на този метод показва , че профилите на кривите от получени чрез метод 1) и метод 3) са сходни, но стойностите са различни. Този метод не намира приложение в практиката извън теоретични тестове.

Анализ на RELU активационна функция

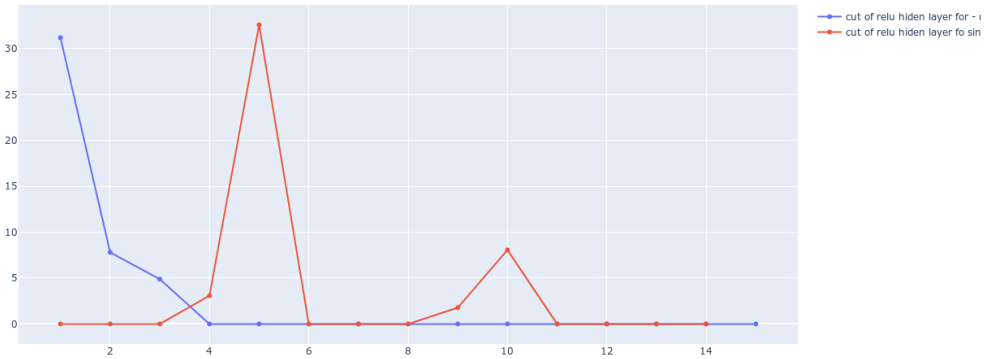
На фигура 10 се вижда стойностите които има всеки възел от изследвания слой, когато невронната мрежа има сигмоидна, релейна и тангенса функция. Стойностите получени при подаване на входни данни за sin функция.



Фиг. 10: Крива на стойностите на скрит слой при различни активационни функции

Пример за мъртви невронни при ReLu

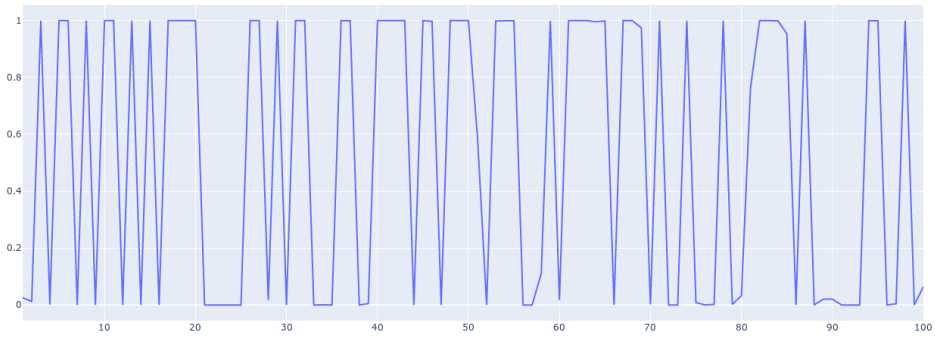
На фигура 11 се вижда пример за мъртви неврони или неврони които връщат стойност “0” без значение от входните данни. Възлите в скрития слой на невронната мрежа от 11 до 16 връщат стойност “0”, независимо каква стойност на входните данни бъде подадена, това е валидно за цялата обучителна матрица.



Фиг. 6: Пример за мъртви неврони

Анализ на сигмоидна функция

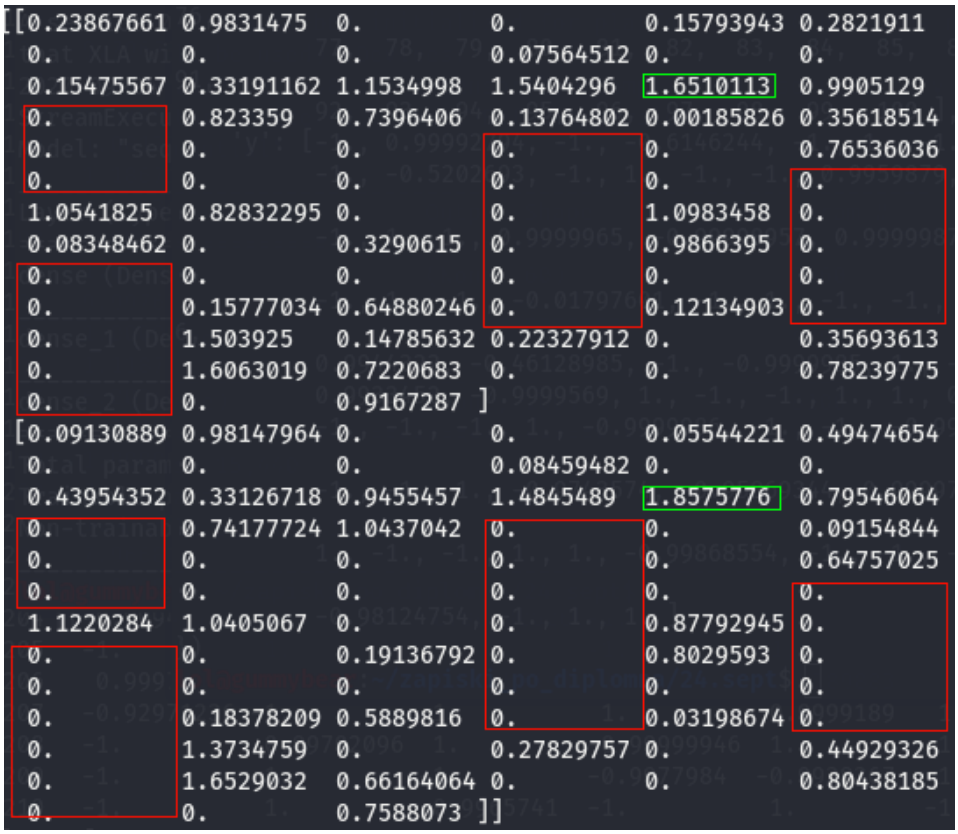
На фигура 12 се вижда вектора на скрития слой които използва сигмоидната функция.



Фиг. 7: Сигмоидна функция в скрит слой с 100 възела

Пример за затихване на мрежата когато се използва сигмоидна функция за активиране на скрит слой на фигура 13. На фигура 14 се виждат стойностите извлечени от скрития слой намиращ след слоя при които се използва сигмоидна функция. Наблюдават се клъстери от мъртви неврони. Както и максимални стойностите които намаляват при всеки следващ слой. Когато максималните стойности от слой станат твърде малки мрежата не са активира и спира

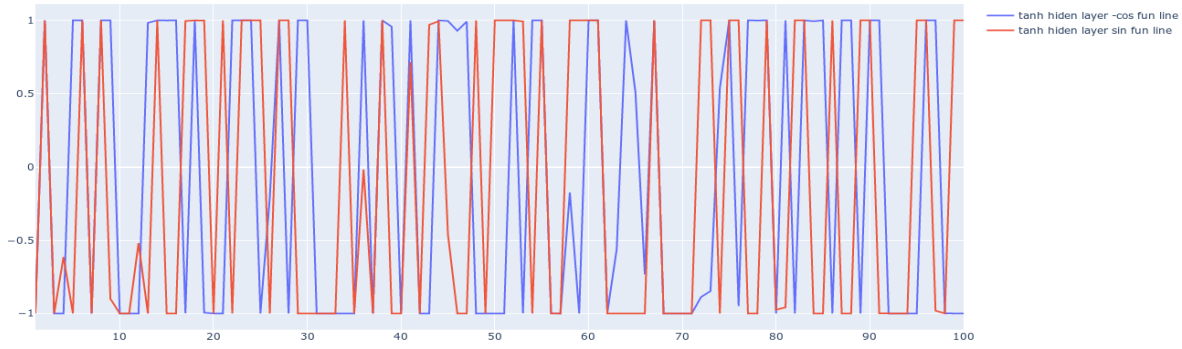
да функционира.



Фиг. 8: Пример за затихване в следствие от използване на сигмоидна функция

Анализ на хиперболична тангесова активационна функция

При изследване на tanh функцията може да се наблюдава аномалии при обучението на невронната мрежа. На фигура 15 се вижда пример, в които невронната мрежа може да разпознава само един клас. И връща един и същ резултат въпреки тренировъчните данни. Мрежата се специализира и разпознава само първия клас чийто данни заемат първо място в обучителната матрица. Този ефект е описан в статии изследващи обучение чрез непълни данни (Haibo He and Garcia, E.A. , 2009).



Фиг. 9: Крива на стойностите скрит слой с tanh активационна функция при sin и -cos тестови данни

Фиг. 10: Пример за аномалия при използване на tanh активационна функция

ИЗВОДИ

Статията се стреми да разшири каталога със съществуващи примери за използвани активационни функции и отражението им в невроните мрежи. Разглеждат се често срещани проблеми представени чрез практически изследвания и данни.

REFERENCES

Vollum, R.L. (2003). Investigation Into backprop forces and deflections at St George Wharf. *Magazine of Concrete Research*, 55(5),449–460.

Mladenov, V. and Jordanova, S. (2006). Fuzzy control and neural networks., (*Оригинално заглавие : Младенов, В , Йорданова, С,* (*2006*) . *Размито управление и невронни мрежи*), 22

Haykin, S. (2008). *Neural Networks and Learning Machines*. 3rd Edition ed. University Hamilton, Ontario, Canada.,128

Haykin, S. (2008). *Neural Networks and Learning Machines*. 3rd Edition ed. University Hamilton, Ontario, Canada. ,161

Bishop, C.M. (2016). *Pattern Recognition and Machine Learning.*, 240

Han, J. and Moraga, C. (1995). the Influence of the sigmoid function parameters on the speed of backpropagation learning, 991

Reed, R.D. and J, R. (1999). *Neural smithing : supervised learning in feedforward artificial neural networks*. Cambridge, Mass.: the Mit Press,15-31

Stathakis, D. (2009). How many hidden layers and nodes? In*ternational Journal of Remote Sensing*, 30(8), pp.2133–2147.

Haibo He and Garcia, E.A. (2009). Learning from Imbalanced Data. *IEEE Transactions on Knowledge and Data Engineering, 21*(*9*), *pp.1263–1284.*