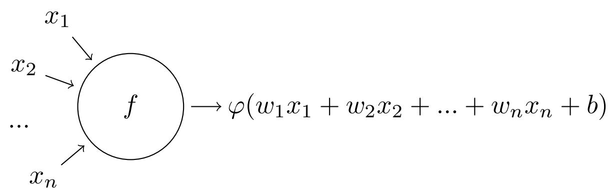
Ակտիվացիոն ֆունկցիաներ մաս 1

Հայկ Կարապետյան

1 Ներածություն

Ինչպես գիտենք հասարակ նեյրոնը մուտքային տվյալների վրա կատարում է գծային ձևափոխություն։ Հաջորդ գործողությունը, որը կարող ենք ավելացնել նեյրոնին, դա ֆունկցիայի կիրառումն է (Գծագիր 1)։ Այդ ֆունկցիան կոչվում է ակտիվացիոն ֆունկցիա (activation function) և կարող է լինել կամայական ֆունկցիա, սկսած քառակուսայինից մինչև սիգնում[1]։



$$f : R^n \to R$$

Գծագիր 1: Նեյրոն, որը կատարում է գծային ձևափոխություն և կիրառում φ ֆունկցիա

2 Անհրաժեշտություն

Դիտարկենք նեյրոնային ցանց բաղկացած երկու թաքնված շերտից։ Շերտերի նեյրոնները ակտիվացիոն ֆունկցիա չունեն։ Առաջին շերտում ունենք մի նեյրոն, երկրորդ շերտում նույնպես ունենք մի նեյրոն։ Մուտքային տվյալի չափողականությունը 1 է ($k = 1$)։ Առաջին շերտից դուրս եկող արդյունքը կլինի $w_1x_1 + b_1$, երկրորդ շերտից դուրս եկող արդյունքը՝ $w_2(w_1x_1 + b_1) + b_2$ (Գծագիր 2)։ Փակագծերը բացելով և նշանակում կատարելով, կարող ենք տեսնել, որ երկրորդ շերտի առկայությունը անիմաստ է՝

$$(w_2w_1)x_1 + (w_2b_1 + b_2) = w_1'x_1 + b_1'$$
$$w_1' = w_2w_1, \quad b_1' = w_2b_1 + b_2$$

Այսինքն առանց ակտիվացիոն ֆունկցիայի կարող ենք երկու շերտի փոխարեն օգտագործել մեկ շերտ և մոդելը ուսուցման ընթացքում կսովորի w_1' և b_1' պարամետրերի արժեքները։ Ակտիվացիոն ֆունկցիան նեյրոնի գծային ձևափոխությունը դարձնում է ոչ գծային։ Ակտիվացիոն ֆունկցիան կարող է կիրառվել ինչպես թաքնված շերտերում, այնպես էլ ելքային շերտում։

$$x_1 \longrightarrow w_1, b_1 \longrightarrow w_1x_1 + b_1 \longrightarrow w_2, b_2 \longrightarrow w_2(w_1x_1 + b_1) + b_2$$

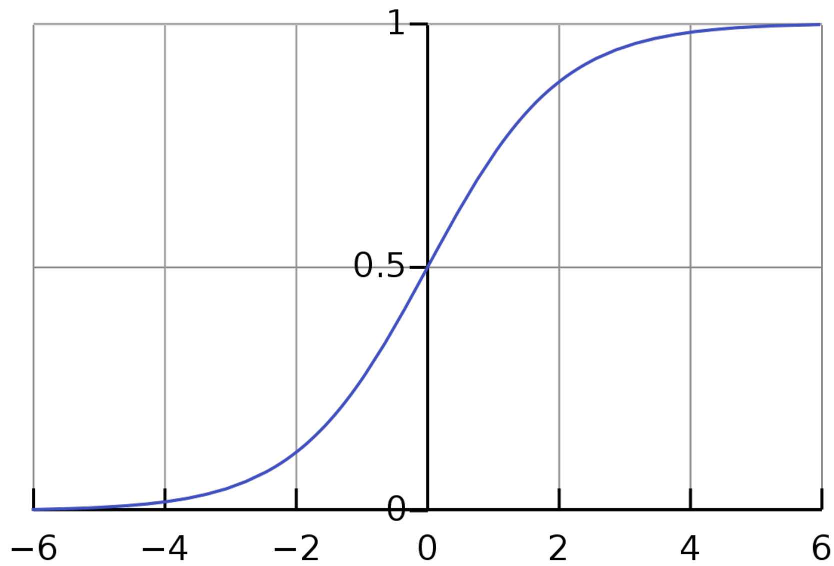
Գծագիր 2: Նեյրոնային ցանց, առանց ակտիվացիոն ֆունկցիայի

[1] $sig(x) = \begin{cases} 1, \text{երբ } x > 0 \\ 0, \text{երբ } x = 0 \\ -1, \text{երբ } x < 0 \end{cases}$

3 Օրինակներ

Ահա հաճախ օգտագործվող ակտիվացիոն ֆունկցիաների օրինակներ:

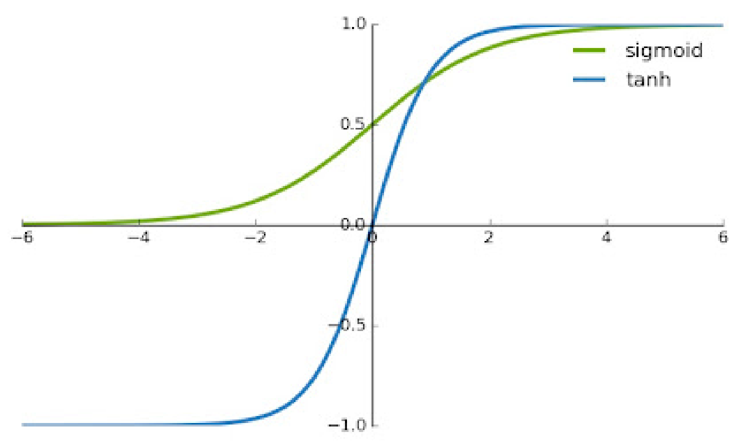
1. Sigmoid: $\sigma(x) = \dfrac{1}{1+e^{-x}}$



Նկար 1: Սիգմոիդ ակտիվացիոն ֆունկցիա

Վերջին շերտում սիգմոիդ ակտիվացիոն ֆունկցիան կարող է օգտագործվել հավանականություն վերադարձնելու համար: Օրինակ՝ պետք է գուշակենք նկարում պատկերված է շուն, թե կատու: Այս դեպքում վերջին շերտում կդնենք մեկ նեյրոն և սիգմոիդ ակտիվացիոն ֆունկցիա: Արդյունքը կասի, թե ինչքան հավանականությամբ է շուն և մեկից հանած այդ արդյունքը կլինի ինչքան հավանականությամբ է կատու ($sig(x) = 0.8$ (շուն), $1 - sig(x) = 0.2$ (կատու))):

2. Tangent hyperbolic: $tanh(x) = \dfrac{e^x - e^{-x}}{e^x + e^{-x}}$



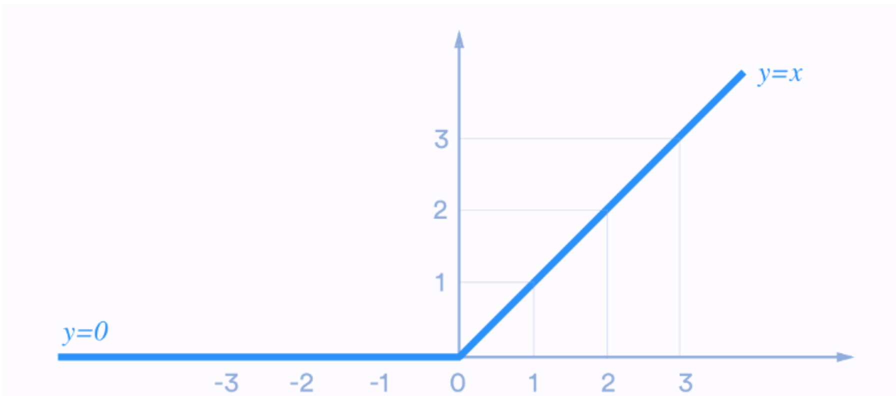
Նկար 2: Տանգես հիպերբոլական և սիգմոիդ ակտիվացիոն ֆունկցիաների համեմատություն

Տանգես հիպերբոլական ակտիվացիոն ֆունկցիան հնարավոր է արտահայտել սիգմոիդի միջոցով:

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

Տանգես հիպերբոլականը վերջին շերտում կարող է օգտագործվել, մինուս մեկից, մեկ միջակայքի թիվ վերադարձնելու համար: Օրինակ՝ ունենք ամրապնդվող ուսուցման գործակալ (RL agent) և որպես արդյունք պետք է վերադարձնել ինչքան պետք է շարժվել x առանցքով: $tanh(x)$-ի նշանը ցույց կտա շարժման ուղղությունը, իսկ արժեքը ցույց կտա ինչքան շարժվել այդ առանցքով:

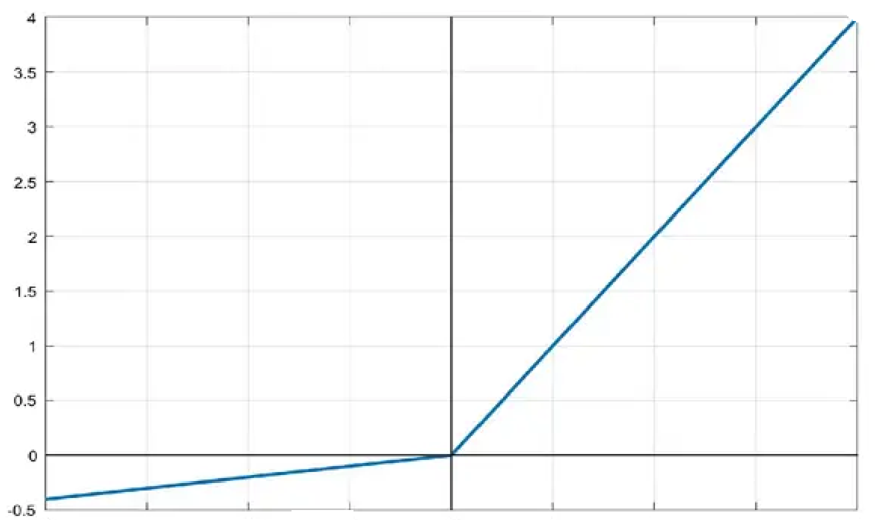
3. Rectified linear unit: $ReLU(x) = max(0, x)$



Նկար 3: ReLU ակտիվացիոն ֆունկցիա

ReLU ակտիվացիոն ֆունկցիան վերջին շերտում կարող է օգտագործվել, միայն դրական արժեքներ կամ զրո վերադարձնելու համար: Օրինակ՝ պետք է գուշակել տան գինը և գիտենք, որ այն չի կարող բացասական լինել: Թաքնված շերտերում օգտագործելիս, այս ակտիվացիոն ֆունկցիան զրոյացնում է բացասական ինֆորմացիան:

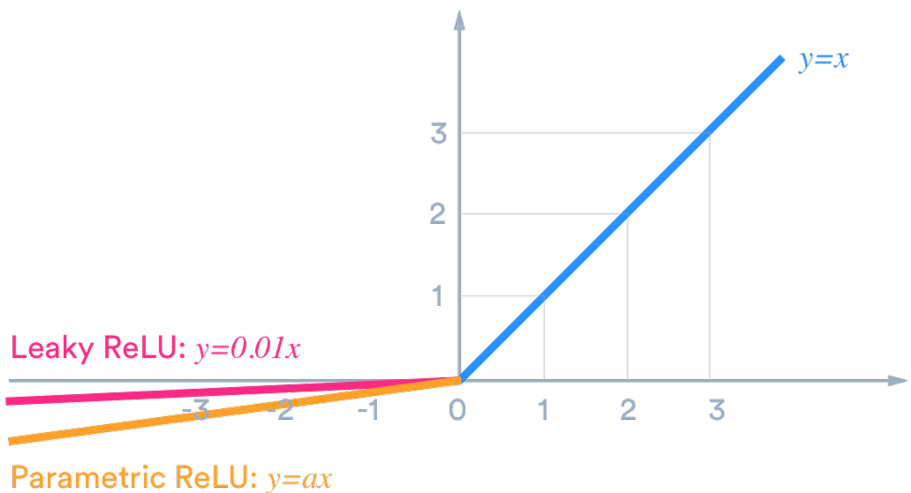
4. Leaky ReLU: $LR(x) = \begin{cases} 0.01x, \text{երբ } x < 0 \\ x, \text{երբ } x \geq 0 \end{cases}$



Նկար 4: Leaky ReLU ակտիվացիոն ֆունկցիա

Ի տարբերություն ReLU-ի, Leaky ReLU-ն բացասական ինֆորմացիան զրոյացնելու փոխարեն փոքրացնում է:

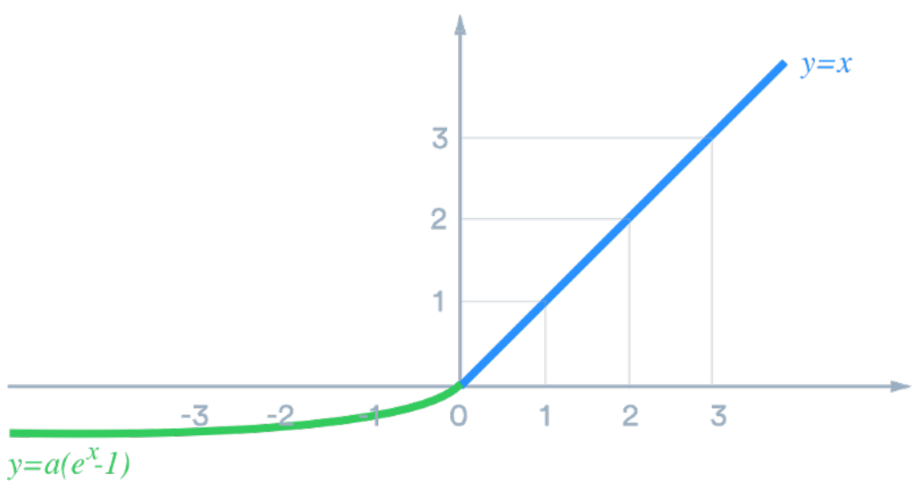
5. Parametric ReLU: $PR(x) = \begin{cases} ax, \text{երբ } x < 0 \\ x, \text{երբ } x \geq 0 \end{cases}$



Նկար 5: Parametric ReLU ակտիվացիոն ֆունկցիա

Երկու տարբերակ կա ընտրելու a արժեքը: 1-ին՝ տարբեր արժեքներով փորձարկել և տեսնել, որի դեպքում է մոդելի ճշգրտությունը ավելի բարձր լինում: 2-րդ՝ դարձնել ուսուցանվող պարամետր և մոդելի ուսուցման ընթացքում սովորել լավագույն արժեք:

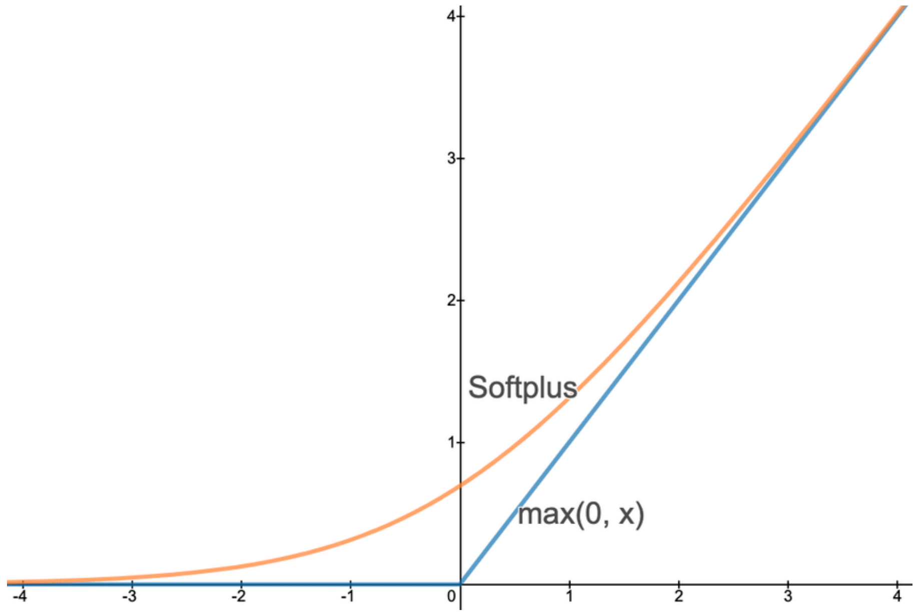
6. Exponential linear unit: $ELU(x) = \begin{cases} \alpha(e^x - 1), \text{երբ } x < 0 \\ x, \text{երբ } x \geq 0 \end{cases}$



Նկար 6: ELU ակտիվացիոն ֆունկցիա

ELU ակտիվացիոն ֆունկցիան բացասական արժեքները փոխարինում է էքսպոնենտով և այստեղ նույնպես α կարող է լինել կամ ուսուցանվող պարամետր, կամ ընտրվել տարբեր արժեքներ փորձելու եղանակով:

7. SoftPlus: $SP(x) = ln(1 + e^x)$



Նկար 7: SoftPlus և ReLU ակտիվացիոն ֆունկցիաներ

SoftPlus ակտիվացիոն ֆունկցիան մոտարկում է ReLU-ն օգտագործելով լոգարիթմական ֆունկցիա:

8. Softmax: $S(x_1, x_2, ..., x_n) = \left(\frac{e^{x_1}}{\sum\limits_{i=1}^{n} e^{x_i}}, \frac{e^{x_2}}{\sum\limits_{i=1}^{n} e^{x_i}}, ..., \frac{e^{x_n}}{\sum\limits_{i=1}^{n} e^{x_i}} \right)$

Softmax ակտիվացիոն ֆունկցիան ցանկացած մուտքային վեկտոր դարձնում է հավանականային վեկտոր: Վեկտորը, որի բոլոր արժեքները մեծ կամ հավասար են զրոյից և նրանց գումարը հավասար է մեկի, կոչվում է հավանականային վեկտոր:

$$P = (P_1, P_2, ...P_n), \quad P_i \geq 0, \quad \sum_{i=1}^{n} P_i = 1$$

P վեկտորը հավանականային վեկտոր է:

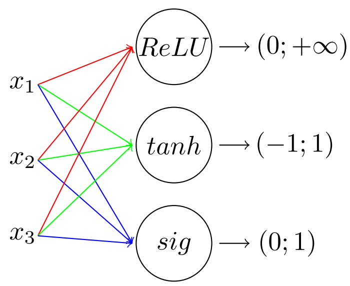
Հավանականային վեկտորի օրինակ է արդար զառի, ամեն կողմի դուրս գալու հավանականությունների վեկտորը:

$$P = (\frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6})$$

Softmax ակտիվացիոն ֆունկցիան չի օգտագործվում թաքնված շերտերում (բացառությամբ մի քանի դեպքերի), այլ օգտագործվում է վերջին շերտում և վերադարձնում է հավանականություններ: Օրինակ՝ պետք է վերադարձնել նկարում պատկերված է շուն, կատու, թե փիղ: Վերջին շերտում ունենալու ենք 3 նեյրոն և կիրառելու ենք softmax ակտիվացիոն ֆունկցիա, որտեղ ամեն հավանականություն ցույց է տալու, թե որ կենդանին ինչ հավանականությամբ է պատկերված: P = (0.1, 0.3, 0.6) նշանակում է նկարում 0.6 հավանականությամբ պատկերված է փիղ, 0.3 հավանականությամբ կատու և 0.1 հավանականությամբ շուն:

4 Ակտիվացիոն ֆունկցիան նեյրոնի համար, թե շերտի

Հարց է առաջանում ակտիվացիոն ֆունկցիան կիրառվում է նեյրոնի վրա, թե՞ շերտի: Softmax ակտիվացիոն ֆունկցիան հասկանալի է, որ կիրառվում է ամբողջ շերտի վրա, որպեսզի շերտի ելքերի գումարը հավասար լինի մեկի: Իսկ ի՞նչ է կատարվում մյուս ակտիվացիոն ֆունկցիաների դեպքում: Հնարավո՞ր է կիրառել տարբեր ակտիվացիոն ֆունկցիաներ նույն շերտի տարբեր նեյրոնների վրա:
Տեսականորեն հնարավոր է կիրառել տարբեր ակտիվացիոն ֆունկցիաներ նույն շերտի տարբեր նեյրոնների վրա և այդպիսի փորձարկումներ արվել են, բայց ստացված արդյունքը շատ չի տարբերվել կամ ավելի վատ է եղել, շերտի բոլոր նեյրոնների վրա նույն ակտիվացիոն ֆունկցիան կիրառելու հետ համեմատած: Ավելի վատ լինելը կապված է նրանով, որ տարբեր նեյրոններից դուրս եկող ելքերը, ունեն տարբեր միջակայքեր: Օրինակ` ունենք 3 նեյրոն, առաջին նեյրոնի ակտիվացիոն ֆունկցիան ReLU է, երկրորդինը` tanh, երրորդինը` sigmoid: Առաջին նեյրոնի ելքի միջակայքն է` $(0; +\infty)$, երկրորդինը` (-1; 1), երրորդինը` (0; 1) (Գծագիր 3):



Գծագիր 3: Տարբեր ակտիվացիոն ֆունկցիաներով շերտ:
Նեյրոնների ելքերը տարբեր միջակայքերի են:

Սա առաջին թերությունն է, իսկ երկրորդ թերությունն այն է, որ ամեն շերտի նեյրոնների համար պետք է ընտրել տարբեր ակտիվացիոն ֆունկցիաներ, և մոդելի մեջ ձեռքով ընտրվող փոփոխականների քանակը շատա-նում է: Օրինակ` ունենք 3 շերտից բաղկացած նեյրոնային ցանց, ամեն շերտում 10 նեյրոն: Այն դեպքում, երբ ամեն շերտի համար ընտրում ենք մեկ ակտիվացիոն ֆունկցիա, արդյունքում հնարավոր տարբերակների քանակը կլինի $3 * 7 = 21$ (7 հաճախ օգտագործվող ակտիվացիոն ֆունկցիաներից ինչ որ մեկը): Իսկ այն դեպքում, երբ ամեն նեյրոնի համար առանձին ենք ընտրում ակտիվացիոն ֆունկցիա, հնարավոր դեպքերի քանակը կլինի $3 * 7 * 10 = 210$: