Adaptabilné aktivačné funkcie pomocou splajnovej interpolácie

Tara Stefányi

3Ib 2018-2019

Abstrakt

Výhodou použitia splajnovej aktivačnej funkcie v neurónovej sieti je možnosť upravovania a prispôsobenia aktivačnej funkcie danému problému, a teda jej rýchlejšie trénovanie. Cieľom tejto práce je implementovať neurónovú sieť využitím splajnu typu C^2 a porovnať náročnosť učenia oproti splajnom typu C^1.

Kľúčové slová: splajnová interpolácia, adaptabilné aktivačné funkcie, neurónové siete

1 Úvod

Neurónové siete sa dnes dajú využiť na riešenie mnohých problémov a ich využitie sa stále rozrastá. Podstatou každej siete je aktivačná funkcia a váhy, ktorých úpravou sa sieť učí.

Nie je jednoduché nájsť vhodnú aktivačnú funkciu pre daný problém, preto sa v štandardných sieťach používa niekoľko overených základných funkcií, ako hyperbolický tangens alebo sigmoid, ktoré však nie sú ideálne v každom prípade a nie je možné ich modifikovať počas učenia, kde štandardným postupom je iba modifikácia váh neurónov.

V minulosti bolo navrhnuté aby sa učenie týkalo aj samotnej aktivačnej funkcie, no tradičné spôsoby modifikovania funkcie ju ovplyvňujú na celom intervale, čo sa ukázalo ako nepraktické [1]. Jedným z riešení sú splajnové aktivačné funkcie, pri ktorých vieme podľa potreby upraviť iba malú časť krivky. To znamená, že pri procese učenia siete upravuje okrem váh aj istý interval aktivačnej funkcie, ktorá sa tak lepšie „prispôsobí“ riešenému problému, čím umožní rýchlejšie naučenie siete a navyše sa redukuje problém hľadania vhodnej aktivačnej funkcie.

Cieľom tejto práce je vylepšiť splajnové neurónové siete, znížiť náročnosť ich trénovania použitím splajnov typu C^2 a porovnať s existujúcou implementáciou adaptabilných aktivačných funkcií používajúcou splajny typu C^1 [1]. Splajny typu C^2 sa od splajnov typu C^1 líšia tým, že v hraničných bodoch segmentov sú derivácie druhého rádu spojité, čo zaručuje vyššiu hladkosť krivky.

2 Neurónové siete

Neurónové siete sú jeden z modelov strojového učenia s učiteľom[1] vhodné na klasifikáciu[2] alebo regresnú analýzu[3].

2.1 Neurón

Neuróny, nervové bunky, sú v anatómii označované ako základná stavebná jednotka nervovej sústavy. Zabezpečujú prenos informácii v tele organizmu, a tým riadenie ostatných orgánov.

Neuróny v neurónových sieťach v informatike chápeme ako zjednodušený model skutočnej nervovej bunky, ktorá je schopná prijímať informácie od predchádzajúcich neurónov a odovzdať ju ďalším.

2.2 Viac-vrstvová sieť

Jednotlivé neuróny v sieti sú zvyčajne organizované do viacerých vrstiev, pričom každý neurón prijíma výstupnú hodnotu od neurónov z predchádzajúcej vrstvy a odovzdáva svoju výstupnú hodnotu neurónom na nasledujúcej vrstve [4].

Rozlišujeme tri typy vrstiev:

- vstupná - prvá, distribuuje vstupné hodnoty,
- skrytá - jadro siete,
- výstupná - vráti výsledok.

Na obrázku 1 je znázornená jednoduchá neurónová sieť s tromi vrstvami.

Prepojenia medzi neurónmi v sieti majú svoju váhu, ktorá určuje jeho vplyv na výsledok siete. Výstup i-teho neurónu na k-tej vrstve bude:

$$y_k^i = f\left(\sum_{j=1}^{n_{k-1}} w_k^{j,i} \cdot y_{k-1}^j\right), \tag{1}$$

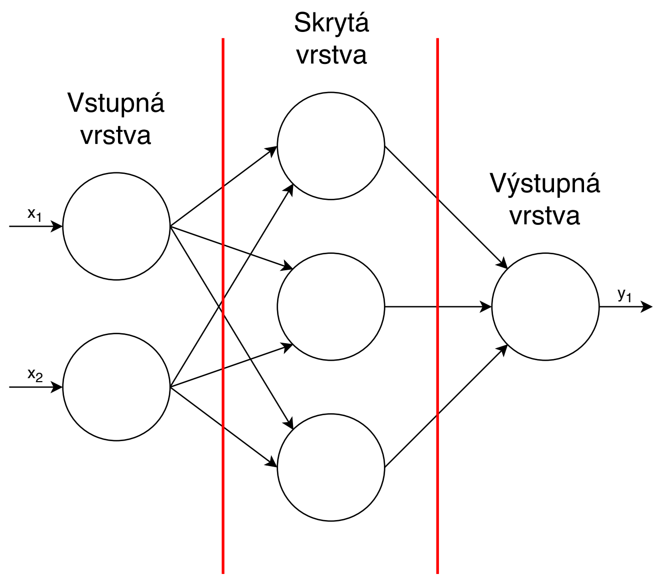
kde f je aktivačná funkcia, n_k je počet neurónov na k-tej vrstve, $w_k^{j,i}$ je váha spojenia medzi neurónom i na vrstve k a neurónom j na vrstve $k-1$.

Na obrázku 2 je znázornený detail neurónu a jeho vstupných a výstupných spojení.

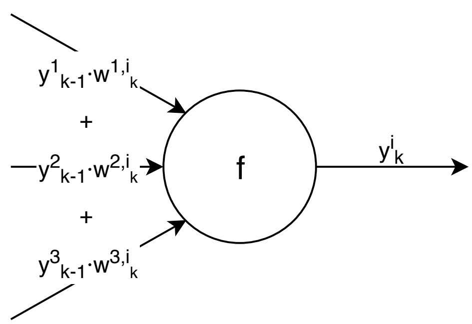
2.3 Back-propagation

Back-propagation, metóda spätného šírenia nám umožňuje upraviť zo začiatku náhodne nastavené váhy neurónovej siete tak, aby bola počítaná chyba minimálna. Algoritmus hľadá najmenšiu hodnotu chybovej funkcie metódou znižovania gradientu.

[1] požadovaný výstup je známy pre vzorky použité na trénovanie modelu
[2] priradenie vzorky do jednej z vopred známych tried, napr. detekcia spamu
[3] výpočet hodnoty na základe hodnôt atribútov, napr. odhad ceny nehnuteľnosti
[4] rekurentné neurónové siete umožňujú aj spätné prepojenia



Obr. 1: Jednoduchá nerónova sieť s jednou skrytou vrstvou



Obr. 2: Neurón

Uvažujme neurónovú sieť o M vrstvách. Hodnotu δ_i^M pre výstupnú vrstvu, pri ktorej porovnávame aktuálny výstup siete s očakávaným vyjadríme ako:

$$\delta_i^M = f'\left(\sum_{j=1}^{n_M} w_M^{j,i} \cdot y_{M-1}^j\right) \cdot \left(o_i - y_M^i\right), \tag{2}$$

kde o_i je reálny výstup siete pre daný vstup.

Potom hodnoty δ_i^m predchádzajúcich vrstiev môžeme vypočítať nasledovne:

$$\delta_i^{m-1} = f'\left(\sum_{j=1}^{n_{m-1}} w_{m-1}^{j,i} \cdot y_{m-2}^{j}\right) \cdot \left(\sum_{j=1}^{n_{m-1}} w_{m-1}^{j,i} \cdot \delta_j^{m}\right), \tag{3}$$

Takto postupne vypočítame delty pre všetky neuróny v sieti. Potom vieme vypočítať zmenu váh ako:

$$w_{nov\acute{e}}^{j,i} = w_{star\acute{e}}^{j,i} + \Delta w_{j,i}, \tag{4}$$

kde $\Delta w_m^{j,i} = \eta \dot{\delta}_i^m \dot{y}_j^{m-1}$ η určuje učiaci pomer.

2.4 Splajny v back-propagation

Definícia 2.1 Interval $\langle a, b\rangle \quad a, b \in \mathbb{R}$ rozdelíme na k pod intervalov $\langle t_i, t_{i+1}\rangle$ pre $i = 0 \dots k-1$ pričom $\langle a, b\rangle = \langle t_0, t_1\rangle \cup \dots \cup \langle t_{k-1}, t_k\rangle$ a $a = t_0 \le t_1 \le \dots \le t_{k-1} \le t_k = b$. Každý pod interval definujeme polynomickou funkciou $P_i : \langle t_i, t_{i+1}\rangle \to \mathbb{R}$. Funkciu $S : \langle a, b\rangle \to \mathbb{R}$ definovanú:

$$\begin{aligned} S(t) &= P_0(t), t_0 \le t < t_1 \\ S(t) &= P_1(t), t_1 \le t < t_2 \\ &\vdots \\ S(t) &= P_{k-1}(t), t_{k-1} \le t < t_k \end{aligned} \tag{5}$$

nazveme **splajn**.

Označenie 2.2 Vektor $[(t-t_i)^3, (t-t_i)^2, (t-t_i), 1]$ budeme označovať ako $\vec{\boldsymbol{u}}$.

Oproti štandardným neurónovým sieťam upravujeme aj aktivačnú funkciu [1], a to zmenou hodnoty polynómu $P_i : \langle t_i, t_{i+1}\rangle$ v bode t_i. Túto zmenu vypočítame ako:

$$\vec{\boldsymbol{z}} = \left(o_i - y_M^i\right) \cdot \left(\vec{\boldsymbol{u}} \times \boldsymbol{B}\right), \tag{6}$$

kde $\boldsymbol{B}$ je matica s rozmermi 4×4. Pre predchádzajúce vrstvy potom:

$$\vec{\boldsymbol{z}} = \left(\sum_{j=1}^{n_{m-1}} w_{m-1}^{j,i} \cdot \delta_j^m\right) \cdot \left(\vec{\boldsymbol{u}} \times \boldsymbol{B}\right). \tag{7}$$

Pre kubické splajny je teda $\vec{\boldsymbol{z}}$ vektor o veľkosti 4 a použijeme ho na úpravu priľahlých polynómov $P_{i-1}, \dots, P_{i+2}$ pre $i > 1$:

$$\begin{aligned} P_{i-1}(t_{i-1}) &= P_{i-1}(t_{i-1}) + \vec{\boldsymbol{z}}_1 \\ P_i(t_i) &= P_i(t_i) + \vec{\boldsymbol{z}}_2 \\ P_{i+1}(t_{i+1}) &= P_{i+1}(t_{i+1}) + \vec{\boldsymbol{z}}_3 \\ P_{i+2}(t_{i+2}) &= P_{i+2}(t_{i+2}) + \vec{\boldsymbol{z}}_4. \end{aligned} \tag{8}$$

Analogicky vieme upraviť polynómy $P_i, \dots, P_{i+3}$ pre $i = 1$.

2.5 Kubický Hermitov splajn typu C^2

Kubický Hermitov splajn S aproximujúci funkciu f je po častiach definovaná funkcia nad intervalom $[x_0, x_{n-1}]$. Teda

$$S(x) = \begin{cases} S_0(x) & \text{pre } x \in [x_0, x_1], \\ S_1(x) & \text{pre } x \in [x_1, x_2], \\ \vdots & \\ S_{n-2}(x) & \text{pre } x \in [x_{n-2}, x_{n-1}], \end{cases} \tag{9}$$

kde $S_0, \ldots, S_{n-2}$ nazveme segmenty splajnu S, pričom sa jedná o spojité polynomické funkcie stupňa 3.

Pre ľubovoľné dva susedné segmenty S_i a S_{i+1} musí platiť

- $y_{i+1} = S_i(x_{i+1}) = S_{i+1}(x_{i+1})$,
- $d_{i+1} = S_i'(x_{i+1}) = S_{i+1}'(x_{i+1})$ (definícia triedy C^1),

pričom pre prvý a posledný segment navyše platí

- $S_0'(x_0) = d_0$ a $S_{n-2}'(x_{n-1}) = d_{n-1}$.

3 Záver

Doteraz sme sa zaoberali implementáciou štandardnej a splajnovej neurónovej siete na základe článku [1] a kódu [5] . Plánujeme upraviť splajnovú neurónovú sieť tak, aby využívala Hermitové splajny typu c^2.

Poďakovanie

Týmto sa chcem poďakovať RNDr. Viliamovi Kačalovi, vedúcemu mojej bakalárskej práce, za pomoc pri vybratí témy, cenné rady a pomoc pri jej vypracovaní.

[5] https://bitbucket.org/ispamm/spline-nn

Literatúra

[1] Scardapane, S., Scarpiniti, M., Comminiello, D., Uncini, A.: *Learning activation functions from data using cubic spline interpolation*, 2016

[2] Kvasnička, V., Beňušková, Ľ., Pospíchal, J., Farkaš, I., Tiňo, P., Kráľ, A.: *Úvod do teórie neurónových sietí*

[3] Rojas, R: *Neural Networks, Springer-Verlag*, Berlin, 1996

[4] Kačala, V., Török, C.: *Speedup of Bicubic Spline Interpolation*, ICCS 2018

[5] Kačala, V.: *Zrýchlenie výpočtu interpolačných splajn povrchov*, 2016