

Úvod do umelej inteligencie

PROJEKT 2: VIACVRSTVOVÁ NEURÓNOVÁ SIEŤ, ERROR BACKPROPAGATION

Pavol Lucina

2. Jan 2025

1 Aktivačné funkcie

Najprv najväčšiu chybu spôsoboval zlý výber aktivačných funkcií:

- V prvej skrytej vrstve hneď za vstupmi je obzvlášť dôležité použiť tanh, nakoľko sú vstupné dáta aj záporné.
- Najprv som dlho dával všetkým skrytým vrstvám tanh, no neskôr sa mi osvedčilo použiť sigmoid pre skryté vrstvy okrem vstupnej
- Vo výstupnej vrstve je dôležité použiť lineárnu aktivačnú funkciu, pretože predikujeme funkciu s hodnotami od -1 do 3. Tanh, sigmoid aj relu majú iné rozsahy hodnôt, s čím by sa dalo po preškálovaní alebo použití leaky relu, no lineárna bola dostatočne dobrá.

2 Počet layerov a neurónov

- Prvotné naivné nastavenie 1 hidden layer a 20 neurónov sa ukázalo ako zjavný underfitting.
- Druhý väčší pokus bol 60 v prvej skrytej vrstve a 50 v druhej. Neskôr som zistil, že to bola celkom šťastná náhoda. Celkovo vyšší počet neurónov už spôsoboval overfitting, menší underfitting. V druhom hidden layeri bolo lepší nižší počet neurónov ako v prvej, keďže na výstupe je iba 1 neurón namiesto 2. Skúšal som aj iné pomery, no približne 6:5 alebo 4:3 boli najlepšie.
- Nakoniec pokus s 3 skrytými vrstvami ešte viac pomohol zvýšiť presnosť modelu. Zvolil som počty neurónov 55, 35, 20.

3 Bias

Bias sa ukázal ako nevyhnutný, najdôležitejší bol na vstupnej vrstve. Použitie biasu na každej ďalšej vrstve len vylepšilo presnosť, čiže som ho dal na všetky vrstvy.

4 Learning rate

- Pri hľadaní ideálnej rýchlosti učenia som sa pozastavil na 0.007. Okolité hodnoty ako 0.01 a 0.005 boli tiež fajn.
- Learning rate schedule je riešený naivne. Pre jednoduchosť na každej 100 epoche sa aktuálna α vynásobí lr_decay . Nevýhoda tohto postupu je, že treba upraviť lr_decay v závislosti od počtu epoch.
- Použitie schedule pomohlo relatívne málo. Pri 2000 epochách sa ako podobné javili hodnoty od 0.92 do 0.98. Medzi jednotlivými spusteniami s identickými parametrami bola istá variabilita presnosti, ktorá zťažovala hľadanie. Vyššie hodnoty (0.98) znížili learning rate iba trochu, čo by mohlo znamenať overfitting, no bol ťažko pozorovateľný. Vyskúšal som aj postupné mierne zvyšovanie lr (1.05), čo tiež nebolo zlé.

5 Inicializácia váh

- Začal aj skončil som na jednotuchom random (použitie `np.random.randn`) čiže normálneho rozdelenia
- Xavier aj Lecun vždy začali aj skončili na vysokej loss, výsledné MSE bolo horšie tiež.

6 Momentum

- Najlepšie sa mi javilo $momentum = 0.7$
- Momentum malo celkovo dosť nízky vplyv

7 Early stopping

Väčšinou sa loss každými pár epochami postupne znižovala, čiže som toto neimplementoval. Chyba počas učenia sa medzi epochami niekedy nezmenila vôbec alebo sa odošť zhoršila, inokedy bolo zlepšenie nárazové (často zrovna pri zmene learning rate na každej 100-tej epoche). To by veľmi sťažovalo určovanie vhodného momentu na zastavenie.

8 Iné loss funkcie

Na grafe prvého úspešnejšieho pokusu som si všimol, že model vytvoril iba jeden kopec, namiesto viacerých lokálnych maxím. Myslel som si, že je chybou použitie MSE pri trénovaní, že sa príliš zameriava na globálne maximum. Keď som použil Huber loss ktorá by mala zmierňovať veľkosť chyby vyšších rozdielov, presnosť sa zvýšila, aj bolo vidno vznik ostatných kopčekov - lokálnych maxím. Neskôr keď bol zvyšok modelu vylepšený som sa vrátil k MSE, lebo trénovanie ním dosahovalo vyššiu presnosť.