SFC – Demonštrácia učenia backpropagation s aktivačnou funkciou SoftMax

Dávid Oravec (xorave06)

# Inštalácia

Pre spustenie scriptu je nutné mať nainštalovaný python 3.9. Pomocou použitia *pip3 install -r requirements.txt* sa nainštalujú potrebné balíčky.

# Spustenie

Na spustenie scriptu slúži súbor main.py. Script sa dá modifikovať prostredníctvom argumentov.

Príklad spustenia scriptu: *python main.py --layers 8*.

Argumenty:

* --help – vypíše pomoc na spustenie programu
* -i I – modifikuje počet vstupov do neurónovej siete na I vstupov (východzie - 9)
* -o O – modifikuje počet výstupov neurónovej siete na O výstupov (východzie - 4)
* --custom\_input INPUT.TXT – definuje, ktorý súbor bude použitý ako vstup. Každý riadok súboru je jeden vstup do neurónovej siete, poďľa ktorého sa bude učiť. V riadku sú jednotlivé vstupy oddelené medzerou. Počet hodnôt v riadku sa musí zhodovať s nastaveným počtom vstupov do neurónovej siete podľa parametru -i. Počet riadkov musí byť zhodný s počtom riadkov v súbore definujúci výstup. Pre príklad si pozrite input.txt. (východzia hodnota – input.txt)
* --custom\_output OUTPUT.TXT – definuje, ktorý súbor bude použitý ako výstup, podľa ktorého sa bude učiť. Každý riadok súboru je jeden výstup z neurónovej siete. V riadku sú jednotlivé výstupy oddelené medzerou. Počet hodnôt v riadku sa musí zhodovať s nastaveným počtom výstupov z neurónovej siete podľa parametru -o. Počet riadkov musí byť zhodný s počtom riadkov v súbore definujúci vstup. Pozri output.txt. (východzia hodnota – output.txt)
* -l LAYERS [LAYERS …], --layers LAYERS [LAYERS …] – definuje vrstvy neurónovej siete tak, že jednotlivé čísla určujú koľko neurónov bude v danej vrstve. Posledná vrstva sa nedefinuje a je automaticky vytvorená o počte neurónov zhodných s počtom výstupov siete. Ostatné vrstvy sa definujú tak, že každé číslo znázorňuje počet neurónov na danej vrstve podľa poradia čísiel v argumente. Napríklad: *main.py -i 9 -o 4 --layers 5 8* vytvorí neurónovú sieť s 3 vrstvami tak, že na prvej vrstve bude 5 neurónov na druhej 8 neurónov a na 3 vrstve 4 neuróny a na vstupe bude očakávať 9 hodnôt.  
  Diagram, schematic

  Description automatically generated

# Použitie

Po spustení hlavného scriptu sa zobrazí vykreslená neurónová sieť. Neurónová sieť už má vygenerované všetky kroky učenia a aj všetky hodnoty pomocou numpy knižnice. Pre zobrazenie nasledujúceho kroku výpočtu treba stlačiť tlačidlo **šípky doprava**. Ak by sme sa chceli vrátiť späť a pozrieť si predchádzajúci krok výpočtu treba stlačiť tlačidlo **šípky doľava**.

Každý krok vyznačí na neurónovej sieti čo všetko využíva pri danom kroku na výpočet. V strede hore zobrazuje textový popis čo akurát robí a na čom a pod ním znázorní číselný výpočet podľa aktuálnych hodnôt. (Hodnoty sa zobrazujú zaokruhlené buď na 3 alebo na 2 desatinné miesta!)

Diagram

Description automatically generated

Program znázorňuje výpočet a učenie pre každý definovaný vstup iba raz! Mení váhy prepojení pre každý vstup. Pri poslednom demonstračnom kroku odstráni vstupy a výstupy.

# Súbory

Projekt obsahuje súbory: main.py, Network.py, Layer.py, DemonstrationSteps.py, input.txt, output.txt, requirements.txt

Hlavným scriptom je **main.py**, ktorý je aj vstupným bodom programu. Obsahuje spracovanie argumentov, načítanie vstupného a výstupného súboru, vygenerovanie vrstiev neurónovej siete, inicializáciu vykreslovacieho okna, spustenie vykreslenia neurónovej siete a vygenerovanie demonstračných krokov učenia.

V pomocných súboroch Network.py, Layer.py, DemonstrationSteps.py sa nachádzajú definície daných tried.

Pre Network.py je definovaná trieda, ktorá zastrešuje celú logiku vykreslovania, definuje poradie vykonávania výpočtov vo vrstvách.

Pre Layer.py je definovaná trieda, ktorá má v sebe všetky prepojenia, ktoré do danej vrstvy vstupujú, hodnoty neurónov, hodnoty aktivácií neurónov podľa aktivačnej funkcie, hodnoty vstupov a funkcie výpočtu všetkých hodnôt vrátane aktivačnej funkcie softmax, výpočtu delty a výpočtu váh pri použití backpropagation.

Súbor DemonstrationSteps.py obsahuje základnú logiku vytvárania a uchovávania jednotlivých krokov výpočtu. Následne sa táto logika použije pri generovaní jednotlivých krokov v súbore Network.py.

# Implementácia

Funkcia SoftMax - kde *j* je číslo neuronu, *u* je hodnota neurónu a *m* je počet neurónov. Na výpočet novej váhy sa určí delta δ. Delta pre poslednú vrstvu sa vypočíta:  
, kde *o* je chcený výstup neurónu poslednej vrstvy. Deltu pre ostatné vrstvy vypočítame pomocou vzorca: , kde *l* je daná vrstva siete. A pre modifikáciu daných váh použijeme vzorec:, kde *x*  je vstup do neuronu *j*, *i*je číslo vstupu a *μ*  je koeficient učenia – nastavený na 0.5.

*u* sa vypočíta ako súčin vektoru vstupov a vektoru váh k danému neurónu: .