

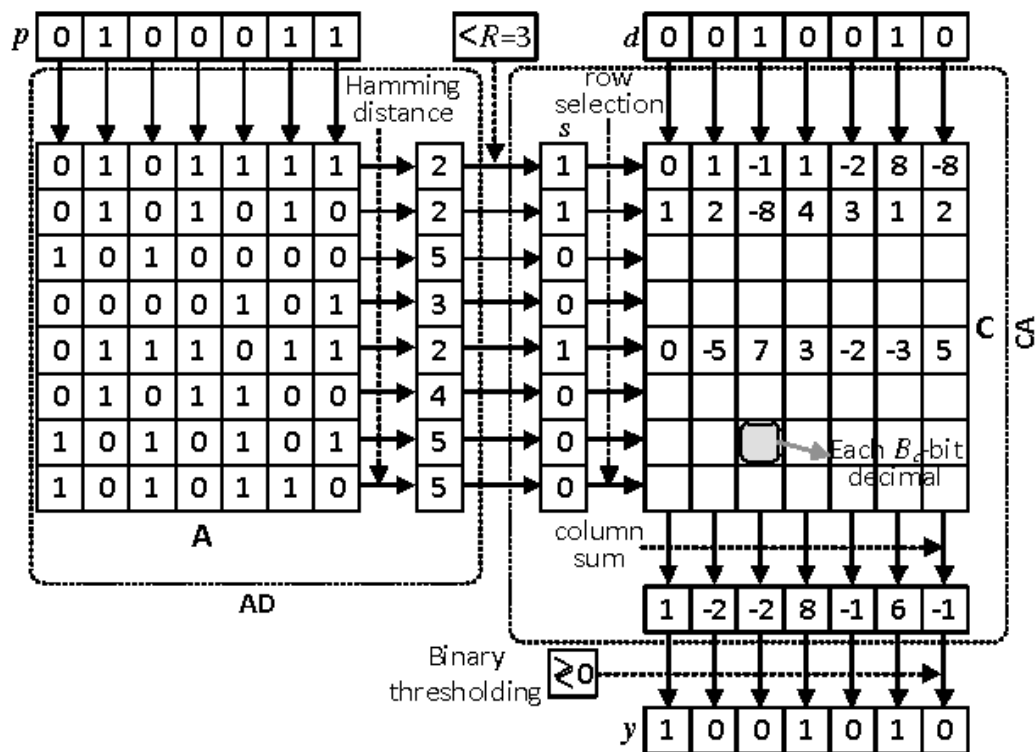
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

Demonstrace učení sítě SDM

1 Sparse distributed memory

SDM lze považovat buď za obsahově adresovatelné rozšíření klasické paměti s náhodným přístupem (RAM) nebo za speciální typ třívrstvé dopředné neuronové sítě. Hlavní změny SDM oproti RAM jsou:

- SDM vypočítává Hammingovu vzdálenost mezi referenční adresou vstupního vektoru a každou „řídkou“ adresou, které reprezentují váhy neuronů skryté vrstvy. Pro každou vzdálenost, které je menší nebo rovna zadanému poloměru, je vybrán odpovídající paměť.
- Paměť je reprezentována čítači, namísto jednobitových paměťových prvků.
- Při zápisu do paměti nedochází k jejímu přepsání, ale je reprezentováno následovně:
 - pokud je i -tý bit vstupních dat 1, odpovídající čítače (čítače odpovídající adresám jejichž vzdálenost od vstupní adresy je menší nebo rovna poloměru) se inkrementují v i -tých sloupcích.
 - naopak pokud je i -tý bit vstupních dat 0, odpovídající čítače se dekrementují.
- Čtení z paměti probíhá následovně:
 - obsah všech odpovídajících čítačů je sečten po sloupcích a představuje vnitřní potenciál neuronu
 - obsah všech čítačů je sečten po sloupcích a odpovídá a představuje práh neuronu
 - každý vnitřní potenciál je poté porovnán s prahem neuronu, pokud je vnitřní potenciál větší nebo roven hodnotě prahu, odpovídající výstupní bit je nastaven na 1, v opačném případě je nastaven na 0 [1]



Obrázek 1: Sparse distributed memory [2]

2 Implementace

K implementaci grafického uživatelského rozhraní byl použit jazyk Python a knihovna PySide6, která nabízí přístup k celému frameworku Qt 6. Pro implementaci samotné SDM byla poté použita knihovna numpy, která umožňuje snadnou práci pro operace nad maticemi.

3 Implementace SDM

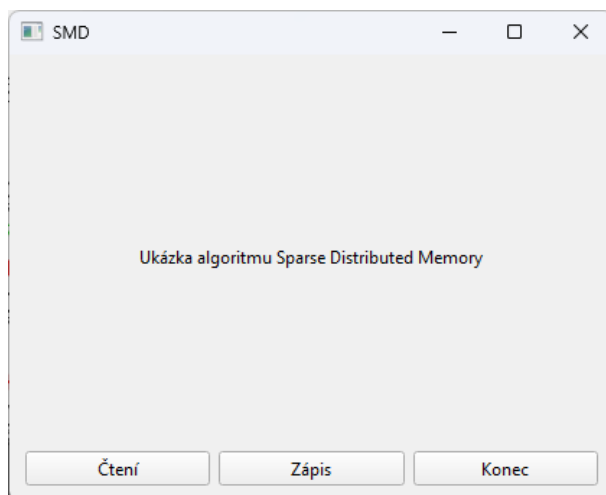
Pro implementaci SDM byla použita výše zmíněná knihovna numpy. Samotná paměť se skládá ze dvou matic. První matice obsahuje binární hodnoty 0 a 1 a reprezentuje váhy neuronů skryté vrstvy. Druhá matice poté reprezentuje pole čítačů.

Pro zápis do paměti tedy nejprve vypočítána Hammingova vzdálenost. Ta je implementována pomocí logické operace XOR. Jelikož výsledkem této operace je však pole, je potřeba prvky tohoto pole sečíst, z čehož vzniká výsledná Hammingova vzdálenost. Následně pak dochází v inkrementaci a dekrementaci čítačů, jak bylo popsáno v sekci 1.

Čtení z paměti pracuje na podobném principu. Nejprve je spočítána Hammingova vzdálenost, poté jsou po sloupcích sečteny příslušné čítače, poté se po sloupcích sečtou všechny čítače a pro jednotlivé pozice výstupu se porovná hodnoty součtu aktivních čítačů a součty všech čítačů.

4 Implementace GUI

Samotná aplikace pak může běžet ve dvou módech. Uživatel si při spuštění zvolí, zda chce aplikaci zapnout v módu pro zápis do paměti, nebo v módu pro čtení z paměti.



Obrázek 2: Úvodní okno

Při kliknutí na tlačítko Čtení se aplikace otevře v módu pro čtení z paměti.

SDM

Odezva: 1 1 0 0 1 1 0 0

Práh: -24 -8 0 3 -4 -12 25 19

Součet: 3 -7 -5 -1 5 5 -3 -1

Vstupní vektor: 1 0 0 0 1 1 0 1

Poloměr: 2

Pole adres	Vzdálenost	Výběr	Pole čítačů
0 1 1 1 1 0 1 0	7 0	-5 9 11 9 5 -13 11 -3	
1 0 1 1 0 0 1 1	5 0	5 -13 11 17 -13 -9 15 19	
1 0 0 1 1 1 0 0	2 1	3 -7 -5 -1 5 5 -3 -1	
0 1 1 1 1 0 0 0	6 0	-10 8 8 6 4 -6 -6 -4	
0 1 0 0 0 0 1 0	6 0	-9 9 -3 -5 -7 -3 5 -7	
1 0 1 1 0 0 0 0	5 0	5 -7 5 3 -7 -3 -5 -1	
0 1 0 0 1 1 1 1	3 0	-14 14 -14 -6 2 18 12 12	
0 0 0 0 0 0 1 1	4 0	-8 -16 -10 -4 -12 -14 8 18	
0 0 0 0 1 0 1 1	3 0	-11 -11 -7 -5 7 -7 11 11	
0 0 1 0 1 1 1 0	4 0	-5 -3 5 -7 7 7 3 -5	

Init Krok Konec

Obrázek 3: Mód pro čtení z paměti

Celou paměť je nejprve nutné nainicializovat kliknutím na tlačítko init. Poté už je možné klikáním na tlačítko Krok možné sledovat jednotlivé kroky algoritmu. Prvním krokem je výpočet vzdálenosti adres od vstupního vektoru.

Následuje vybrání příslušného čítače. To je indikováno hodnotou 1 v poli Výběr. Následně dojde k barevnému vyznačení příslušné adresy a čítače.

Poté následuje výpočet součtu aktivních neuronů, prahu všech neuronů a celkové odezvy sítě.

Druhou možností je otevření aplikace v režimu pro zápis do paměti při kliknutí na tlačítko Zápis.

SDM

Vstupní vektor: 1 0 0 1 1 0 1 1

Poloměr: 2

Požadovaná odezva: 1 0 0 1 1 0 1 1

Pole adres	Vzdálenost	Výběr	Pole čítačů
1 1 0 0 0 0 0 0	0 0	0 0 0 0 0 0 0 0	
0 0 1 0 1 1 0 1	0 0	0 0 0 0 0 0 0 0	
0 1 0 0 0 0 1 1	0 0	0 0 0 0 0 0 0 0	
1 1 1 1 0 0 1 0	0 0	0 0 0 0 0 0 0 0	
0 1 1 0 1 1 0 1	0 0	0 0 0 0 0 0 0 0	
0 0 1 1 1 0 0 0	0 0	0 0 0 0 0 0 0 0	
0 1 0 1 0 1 1 1	0 0	0 0 0 0 0 0 0 0	
1 0 1 1 1 0 0 1	0 0	1 -1 -1 1 1 -1 1 1	
1 0 0 0 1 1 1 0	0 0	0 0 0 0 0 0 0 0	
1 0 0 1 1 0 1 1	0 0	1 -1 -1 1 1 -1 1 1	

Init Krok Konec

Obrázek 4: Mód pro zápis do paměti

Opět je potřeba celou paměť nainicializovat kliknutím na tlačítko Init. Následně po kliknutí na tlačítko Krok dojde k výpočtu vzdáleností adresových vektorů od vstupního vektoru.

V dalším kroku dochází k výběru příslušných čítačů. Vybrání je pak následně opět barevně zvýrazněno.

Po dalším kliknutí na tlačítko Krok je v poli čítačů barevně vyznačeno upravení jednotlivých čítačů. Modrou barvou jsou zvýrazněny čítače, které se budou dekrementovat, zatímco červenou barvou jsou zvýrazněny čítače, které se budou inkrementovat.

Poté jsou zvýraznění smazána, je vygenerovaný nový vektor a je možné pokračovat se zápisem pro nový vstup.

5 Závěr

Vývoj aplikace byl proveden na operačním systému Windows, ale byla testována i na Linux, konkrétně na distribuci Fedora. V případech jakýchkoliv problémů s externími knihovnami je možné stáhnout sestavenou aplikaci.

Reference

- [1] Wikimedia Foundation. (2022, November 10). Sparse distributed memory. Wikipedia. Retrieved November 26, 2022, from https://en.wikipedia.org/wiki/Sparse_distributed_memory
- [2] Naresh R. Shanbhag. (2016, June 10). Retrieved November 26, 2022, from <http://shanbhag.ece.illinois.edu/publications/mingu-TBIOCAS-2016.pdf>