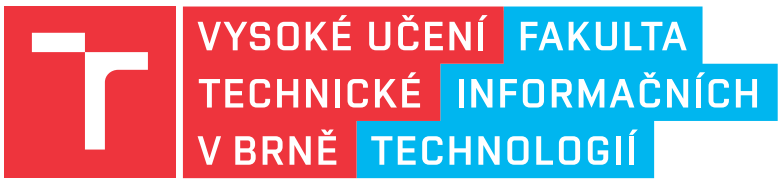


Technická zpráva

Praktická úloha řešená sítí Hopfield - autoasociace

27. listopadu 2023 Martin Zmitko (xzmitk01)

1 Úvod

Tato práce do předmětu SFC se věnuje použití Hopfieldovy sítě při autoasociaci. Cílem bylo naimplementovat aplikaci pro rekonstrukci obrázků, která taktéž demonstruje princip a průběh autoasociace s využitím Hopfieldovy sítě.

Hopfieldova síť [1, 2] je plně propojená jednovrstvá rekurentní neuronová síť, ve které jsou vzájemné vazby symetrické, přímé zpětné vazby se zde nevyskytují. Činnost sítě mé dvě fáze – fázi učení (bez učitele) a fázi vybavování (při použití sítě k autoasociaci, dále se Hopfieldovy sítě využívají k optimalizaci, tedy řešení výpočetně složitých úloh jako např. úlohu obchodního cestujícího).

Ve fázi učení se tvoří matice vah w a prahů θ na základě vstupní množiny dat $T = \{\vec{l_1}, \vec{l_2}, \cdots, \vec{l_P}\}$, kde $\vec{l_p} = \{\vec{i}_{p1}, \vec{i}_{p2}, \cdots, \vec{i}_{pn}\}$, dle vzorců:

$$w_{ij} = \frac{1}{P} \sum_{p=1}^{P} i_{pi} i_{pj}$$

$$w_{ii} = 0$$

$$\theta_i = \frac{1}{2} \sum_{j=1}^{n} w_{pi}$$

Ve fázi vybavování se k sekvenční aktualizaci hodnot neuronů používá skoková bipolární funkce

$$y_j(k+1) = \begin{cases} 1 & \text{pro } u_j(k) > \theta_j \\ -1 & \text{pro } u_j(k) < \theta_j \\ y_j(k) & \text{pro } u_j(k) = \theta_j \end{cases}$$

kde

$$u_j(k) = \sum_{i=1}^{n} w_{ji} y_i(k)$$

Celková energie sítě se počítá dle vzorce

$$E = -\frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} w_{ij} y_i y_j + \sum_{i=1}^{n} \theta_i y_i$$

Při změně polarity neuronu se síť přiblíží k jednomu z uložených vzorů – energie tedy vždy klesne a nakonec síť konverguje v lokálním či globálním minimu energetické funkce. Toto se typicky detekuje a iterace se zastaví v případě, že se prošly všechny neurony a energie se nezměnila.

2 Implementace

Aplikace je koncipována jako program v jazyce Python s uživatelským rozhraním využívajícím modul `Tkinter`. Program je tvořen jediným souborem `hopfield.py` obsahujícím veškerou potřebnou funkcionalitu.

Trénink neuronové sítě proběhne při spuštění programu a využijí se všechny obrázky s příponou `.png` ve složce `data`. Obrázky mají očekávanou velikost 64x64 pixelů a jsou černobílé. Program je schopný načíst i obrázky v odstínech šedi, v tom případě však budou všechny odstíny šedi považovány za bílou. Ke čtení a ukládání obrázků je využita knihovna `Pillow`.

Pro demonstraci činnosti je aktuální načtený obrázek zobrazován v grafickém uživatelském rozhraní. K tomu je využit prvek `Canvas`, na kterém je vykreslen pomocí funkce `create_image` obrázek `PhotoImage`, který již přímo obsahuje zobrazovaná data, které jsou aktualizována metodou `put`, kterou odhaluje `PhotoImage`.

Samotnou Hopfieldovu síť implementuje třída `HopfieldNetwork`. Ta obsahuje všechny potřebné parametry (jako váhovou matici, prahový vektor, vnitřní stav nebo vnitřní energii), stejně tak metody nutné pro trénink a autoasociaci. K ukládání a výpočtům s maticemi a vektory je použit modul `NumPy`. Proces rekonstrukce obrázků je zahájen voláním metody `run`, která inicializuje všechny potřebné hodnoty.

Metoda `update_neuron` obsahuje logiku pro aktualizaci hodnoty jednoho neuronu. Při změně hodnoty v reálném čase proběhne aktualizace odpovídajícího pixelu. Dle vzorce:

$$\Delta E = -(y_r(k+1) - y_r(k))(u_r(k) - \theta_r)$$

se vypočte změna energie pro neuron r, přičte se k aktuální energii a uloží se na konec seznamu `energy`, obsahující hodnoty energie pro každou iteraci. Tyto hodnoty se používají k vykreslování grafu, který v reálném čase ukazuje postupné klesání energie při iteraci sítě. Graf se vykresluje s využitím modulu `matplotlib`.

Další volání `update_neuron` probíhá rekurzivně s využitím funkce `after` poskytovanou modulem `Tkinter`. Ta naplánuje volání až za nějaký čas (zde byla zvolena nejnižší možná hodnota – jedna milisekunda) a zajistí korektní předávání režie modulu `Tkinter` nutné pro správné zobrazování a fungování uživatelského rozhraní při iteraci sítě. Pro okamžitou iteraci sítě bez vizuální zpětné vazby slouží metoda `update_neuron_instant`, která aktualizuje vykreslovaný obrázek a graf až po konci iterace, ta probíhá ve smyčce.

Iterace se zastaví v případě, když se jednou prošly všechny neurony bez změny energie.

3 Použití

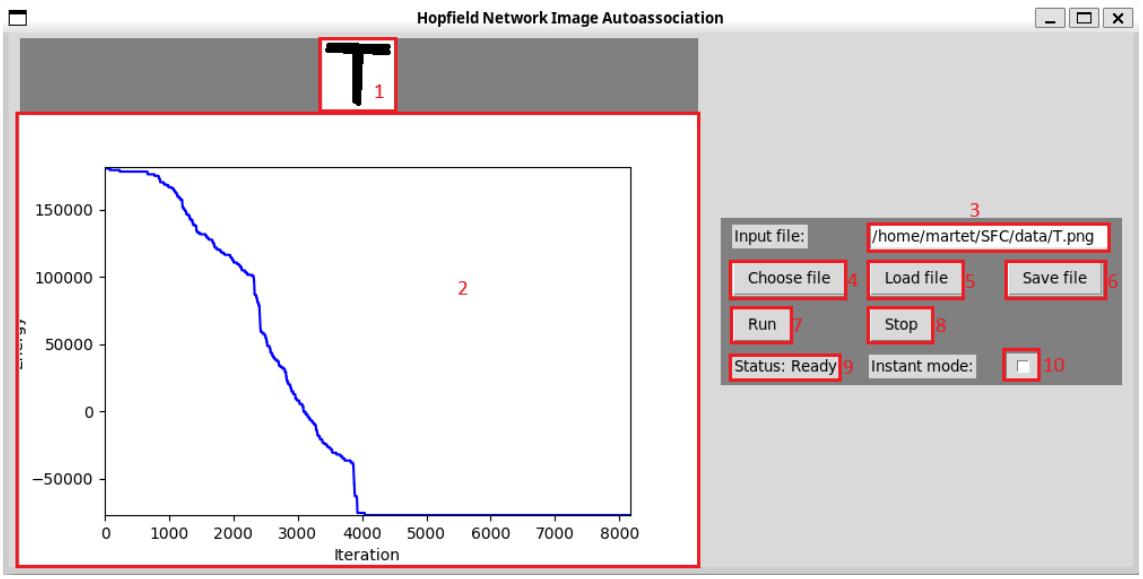
Aplikace byla testována v linuxové distribuci Ubuntu. Prerekvizity ke spuštění programu jsou následující:

- Python 3.8 a vyšší
- Matplotlib
- Pillow
- NumPy

Mezi odevzdanými soubory je skript `setup.sh`, který nainstaluje potřebné Python moduly. Aplikace se spouští příkazem `python3 hopfield.py` bez parametrů, veškeré nastavení a akce se provádí v grafickém uživatelském rozhraní, které je spolu s popisem jednotlivých ovládacích prvků na obrázku 1.

Pro jednoduché testování lze na načtený obrázek kreslit ručně – držení levého tlačítka myši vybarví pixely černě, pravé tlačítko myši bíle. Takto upravený obrázek lze pomocí taktéž pomocí tlačítka „Save file“ uložit do souboru.

Aplikace očekává obrázky pro zapamatování ve složce `data`. Použité obrázky (celkem 6) jsou součástí odevzdaných souborů a na obrázku 2. Příklady vstupů a výstupů na těchto trénovacích datech jsou na obrázku 3. Na tomto příkladě se vyskytují i případy, kdy vstup nebyl úspěšné transformován na jeden ze zapamatovaných vstupů – funkce energie klesla do lokálního minima, které ale nekoresponduje žádnému uloženému, Hammingova vzdálenost vstupu od nějakého ze zapamatovaných je moc velká a nepodařilo se ho obnovit.



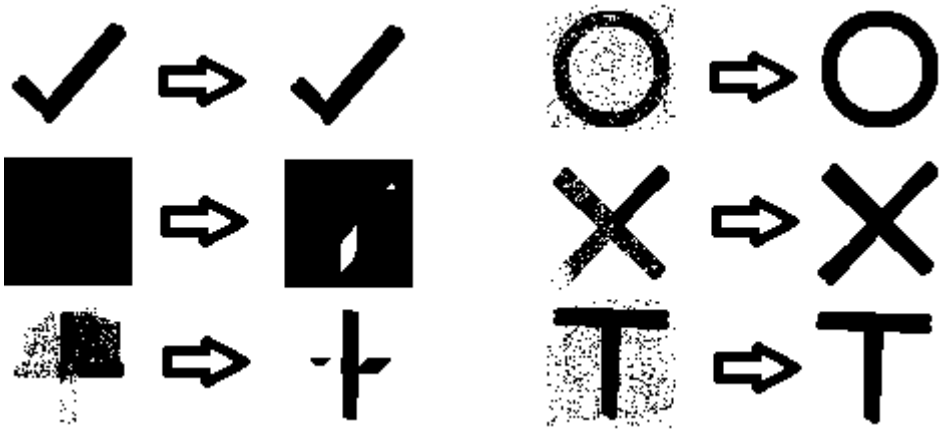
Obrázek 1: Snímek obrazovky aplikace, obsahující následující prvky:
1. současný vnitřní stav sítě (stav obrázku)
2. změna energie sítě
3. cesta k aktuálně načtenému zdrojovému obrázku k autoasociaci
4. změna zdrojového obrázku (otevře dialog pro výběr)
5. načtení zdrojového obrázku do sítě
6. uložení stavu sítě jako obrázek (otevře se dialog pro uložení)
7. zahájení autoasociace
8. zastavení probíhající autoasociace
9. momentální stav aplikace
10. nastavení okamžitého režimu (bez aktualizace v průběhu)

Reference

[1] Hopfield, J. J.: Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, ročník 79, č. 8, 1982: s. 2554–2558, ISSN 0027-8424, doi:10.1073.

[2] Ramya, C.; Kavitha, G.; Shreedhara, K. S.: Recalling of Images using Hopfield Neural Network Model. *CoRR*, ročník abs/1105.0332, 2011, `1105.0332`. Dostupné z: `http://arxiv.org/abs/1105.0332`

Obrázek 2: Obrázky použité k testování sítě



Obrázek 3: Ukázka vstupů a výstupů natrénované Hopfieldovy sítě