

„Utazó ügynök”-probléma megoldása Hopfield hálózat segítségével

A projekt célja: Hopfield hálózat mint kvadratus minimalizáló szemléltetése az „utazó ügynök”-probléma megoldásán keresztül.

Implementáció nyelve: Python 2.7.6
Felhasznált modulok: numpy, matplotlib, random

Csatolt állományok:

- TSP.py: a feladat implementációja
- result, result2, result3 mappák: teszteredményeket tartalmazó mappák

Készítette: Hakkel Tamás
Tárgy: Neurális hálózatok
Dátum: 2016. december 02.

Matematikai interpretáció:

- Adott egy 2D ponthalmaz, melyek közt a távolságot Euklideszi távolsággal adom meg.
- A ponthalmaz elemeit indexszel látom el.
- Keresem az indexek azon permutációját, mely a városokat a permutációnak megfelelően bejárva a legrövidebb utat kapom.
- Ennek érdekében az állapototteret leképezem egy $V \in \{0; 1\}^{n \times n}$ -es mátrixra, ahol n a városok száma, és ha a mátrix (x, i) eleme 1 az azt jelenti, hogy az „ügynök” az útvonalában az x indexű város az i -dik.
- Az így kapott V mátrixnak a következő kritériumoknak kell megfelelnie:
 - ◆ A sorvektorai ortogonálisak legyenek egymásra
 - ◆ Az oszlopvektorai szintén legyenek ortogonálisak
 - ◆ Minden oszlopban és sorban legyen elem
- Definiálok egy energia-függvényt, ami megadja, hogy a V mátrix mennyire tér el a kritériumoktól, és az általa meghatározott útvonal mennyire hosszú. A függvényt a következőképp definiálok:

$$E = \frac{A}{2} \sum_{x=1}^n \sum_{i=1}^n \sum_{\substack{j=1 \\ j \neq i}}^n v_{xi} v_{xj} + \frac{B}{2} \sum_{i=1}^n \sum_{x=1}^n \sum_{\substack{y=1 \\ y \neq x}}^n v_{xi} v_{yi} + \frac{C}{2} \left(\sum_{x=1}^n \sum_{i=1}^n v_{xi} - (n + \sigma) \right)^2 + \frac{D}{2} \sum_{x=1}^n \sum_{\substack{y=1 \\ y \neq x}}^n \sum_{i=1}^n d_{xy} v_{xi} (v_{y,i+1} + v_{y,i-1})$$

- Ezen energiafüggvény (lokális) minimumát keresem Hopfield hálózat és gradient descent módszer segítségével.

Program működése:

- 1) 10 db térkép generálása: 10 város véletlenszerű elhelyezése egy 10x10-es (valós értékű koordinátás) térképen, majd a távolság-mátrix számítása
- 2) Minden egyes térképen 10x futtatom a hopfield hálózatot a következőképp:
 - a) Az állapotot leíró V mátrixot véletlenszerű 0 és 1 közti számokkal feltöltöm.
 - b) Futtatok egy belső ciklust, mely $5 \cdot n^2$ -szer kiválaszt egy véletlen elemet a tömbből és értékét a következő két képlet szerint frissíti (ahol $v(u)$ lesz a V mátrix elemének új értéke):

$$u_{xi}(t+1) = -A \sum_{\substack{j=1 \\ j \neq i}}^n v_{xj}(t) - B \sum_{\substack{y=1 \\ y \neq x}}^n v_{yi}(t) - C \left(\sum_{x=1}^n \sum_{j=1}^n v_{xj}(t) - (n + \sigma) \right) - D \sum_{y=1}^n d_{xy} (v_{y,i+1}(t) + v_{y,i-1}(t))$$

$$v(u) = 1/2 [1 + \tanh(\alpha u)]$$

- c) A belső ciklus futásának befejezése után a korábban említett energiafüggvényt kiértékelem, és ellenőrzöm, hogy ugyanazokat az értékeket kaptam-e az előző 5 kiértékelés alkalmával. Ha igen, akkor leáll a program, egyéb esetben tovább megy egészen addig, míg 5-ször azonos értéket kap az energiafüggvény kiértékelésével vagy eléri a 200 lépéses limitet.
- 3) A futtatás során a sikeres optimalizálás után kapott útvonalakról képet készítek, melyet megjelenítek, és el is mentek. Továbbá figyelem a 10 próbálkozás közül a legjobbat, ennek elmentem az energiafüggvényének végső értékét és a szükséges ciklusok számát. Illetve szintén mentem a 10 próbálkozás alatti energiafüggvény-értékek és ciklusszámok átlagát.

Eredmények:

- A program jelen állípotában képes arra, hogy az esetek kicsit több mint felében (a legutóbbi 300 próbálkozásból 162-szer) helyes (kritériumoknak megfelelő, stacionárium állípotba elért) megoldást kap 200 lépéses limit mellett.
- Bár szemmel láthatólag legtöbbször nem a legrövidebb útvonalat találja meg, de 10-szer futtatva még viszonylag gyorsan (a külső ciklus magját átlagosan 21.8-szor hajtja végre) szuboptimális megoldást kapok.
- A hálózat paramétereinek finomhangolásával még lehet javítani a működésen, de általában a paraméterek egymás ellen hatnak; nem érdemes egyszerre növelni mindegyik paraméter értékét, és a paraméterek változtatásával drasztikusan nőhet a sikertelen útvonalkeresési kísérletek száma.

Irodalom:

http://www.mini.pw.edu.pl/~mandziuk/PRACE/TSP_DM.pdf
https://www.ece.uic.edu/~rgandhi/resume/Hope_tsp.PDF

Példa (result/result_map4_trial*.jpg): mosolygós fej jelöli a legrövidebb útvonalat

