

IAut Tema 3: Retina Artificială

Andrei Olaru

Descrierea problemei:

Extraterestrul Tuk se află pe o planetă unde există multe teritorii cu limbi diferite, și unde fiecare limbă are un alfabet relativ redus (maxim 10 simboluri) iar toate cuvintele din limbă au același număr de simboluri. De asemenea, orice simbol de pe planeta lui Tuk este reprezentabil prin 14 segmente¹. Un alt lucru interesant pe planeta lui Tuk este că toate informațiile se găsesc pe afișaje mari care arată un singur cuvânt la un moment dat, în alb și negru.

Extraterestrul Tuk și-a pierdut vederea cu ceva timp în urmă și are niște ochelari speciali pentru recunoașterea scrisului, care pot fi programați cu caracteristicile limbii teritoriului unde se află (simboluri, număr de simboluri din cuvinte) și care sunt conectați la creierul lui Tuk, producând pentru un cuvânt de n simboluri $n \times 14$ semnale electrice, care indică reprezentarea cuvântului prin caractere de 14 segmente, astfel încât Tuk este capabil să citească informațiile de pe afișaje.

Realizarea temei constă în realizarea unui program pentru ochelarii lui Tuk. Acesta este configurat la fiecare execuție cu parametrii unei limbi – numărul de simboluri pentru un cuvânt și alfabetul – și apoi produce, pentru fiecare intrare (o matrice de $20n \times 30$ pixeli cu valori 0 sau 1 pentru alb sau negru), ieșirea corespunzătoare.

Se cere:

Programul trebuie să folosească o rețea neurală de tip Hopfield, împreună cu un strat de ieșire. Modul de determinare al ponderilor este la alegerea voastră. În README trebuie să prezentați modul de determinare al ponderilor și pragurilor de activare ale neuronilor. Aceste alegeri trebuie motivate.

Intrare / ieșire:

Intrarea programului este formată din două fișiere: `intrare.txt` și `config.txt`. Programul trebuie să suporte 0, 1 sau 2 argumente, reprezentând numele celor două fișiere, în ordinea dată (întâi intrare, apoi config). În cazul în care se dau mai puțin de două argumente, se folosesc pentru argumentele lipsă numele implicite "intrare.txt" și "config.txt". Este bine să suportați fișiere text scrise atât pe Windows cât și pe Linux.

Fișierul `config.txt` conține pe prima linie numărul de simboluri dintr-un cuvânt și numărul de simboluri din alfabet. Apoi o linie goală. Apoi, pentru fiecare simbol din alfabet o serie de 30 de linii, fiecare cu 20 de caractere de 0 și 1 (nedespărțite prin spațiu), care specifică simbolurile alfabetului.

¹http://en.wikipedia.org/wiki/Fourteen-segment_display

Fișierul `intrare.txt` conține o matrice de $w \times 30$, cu $w = 20n$, formată din 0 și 1 (nedespărțite prin spațiu), care constituie imaginea unui afișaj prezentând n simboluri.

Ieșirea programului trebuie afișată la consolă și pusă într-un fișier numit `ieșire.txt` și trebuie să conțină $n \times 14$ valori 0 sau 1, despărțite prin spațiu, reprezentând ieșirile obținute de rețeaua neurală. La consolă se va afișa și interpretarea prin afișaj de 14 segmente al ieșirii, ca în exemplul de mai jos (unde toate segmentele sunt activate):

```

---
| \ | / |
- -
| / | \ |
---
```

Ordinea celor 14 valori pentru fiecare simbol (reprezentând segmentele de afișaj activate) este următoarea (numărată mai jos în hexa):

```

1
23456
7 8
9ABCD
E
```

Pentru a vă ajuta la testare cu generarea intrării, se recomandă utilizarea unui program (se găsesc atât executabile cât și cod în Internet) pentru transformarea din fișiere bitmap în fișiere text (bitmap to ascii).

Bonus:

În anumite zone de pe planeta lui Tuk afișajele au panourile pentru simboluri puse la diverse distanțe orizontale unele de altele, și nu este foarte clar unde începe afișajul și unde se termină. În acest caz, intrarea va avea o dimensiune de $w \times 30$, cu $w = 40n$, în care simbolurile au tot 20×30 fiecare și nu se suprapun, dar nu se știe exact ce distanță este între ele (distanța este 0 în versiunea fără bonus).

Exemplu pe pagina următoare.

imaginea corespunzătoare intrării (cu titlu informativ):

image mărită + grid:

intrare.txt:

[illegible]

...exemplul integral în fisierul `ex-intrare.txt`.

```
config.txt:
```

3 5

```
0000000000000000000000000000
0000000000000000000000000000
0000000000000000000000000000
0000000000000000000000000000
0000000000000000000000000000
0000000000000000000000000000
0000000000000000000010000
00000000000000000000111000
00000000000000000000111000
⋮
```

...un exemplu de simbol (cifra 1) este prezent în fișierul `ex-simbol.txt`.

Pentru fișierul de intrare de mai sus, considerând că simbolurile 9, 6 și jumătatea din dreapta a lui X există în alfabet, iesirea este:

iesire.txt:

1 1 0 0 0 1 1 1 0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 0 0 0 1 1 0 0 0 0 1 0 0 0 0 0 1 0 0

Iar la consolă:

1 1 0 0 0 1 1 1 0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 0 0 0 1 1 0 0 0 0 1 0 0 0 0 0 1 0 0

```

---
|   |
- -
   |
---
```

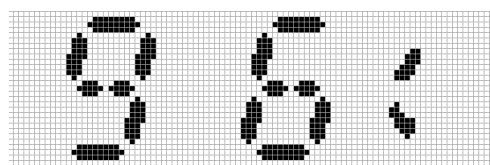
```

---
|
- -
|   |
---
```

/

\

Exemplu pentru bonus (doar imaginea):



SUCCES!