

Neurónové siete projekt 3

Hopfieldova sieť

Miloslav Torda

Úvod

Podľa zadania som naprogramoval Hopfieldovu sieť s 35 neurónmi, ktorá uchováva pamäťové vzory **A**, **B**, **X**, **O**.

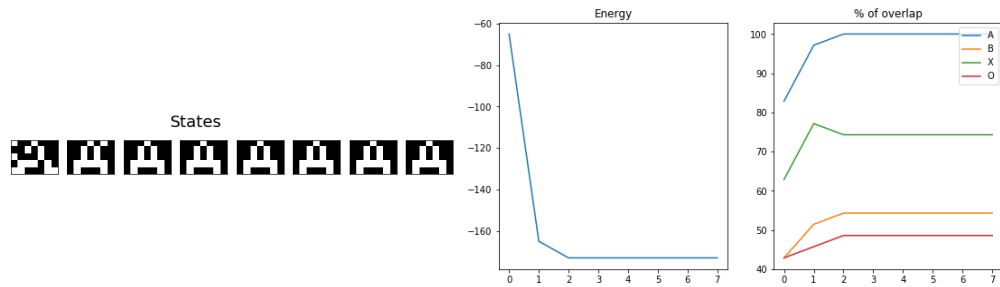
Schopnosť vyvolávať pamäťové vzory

Poškodil som jeden z pamäťových vzorov podľa rôznych úrovní šumu, následne dával ako vstup do siete a nechal ju bežať osem krokov.

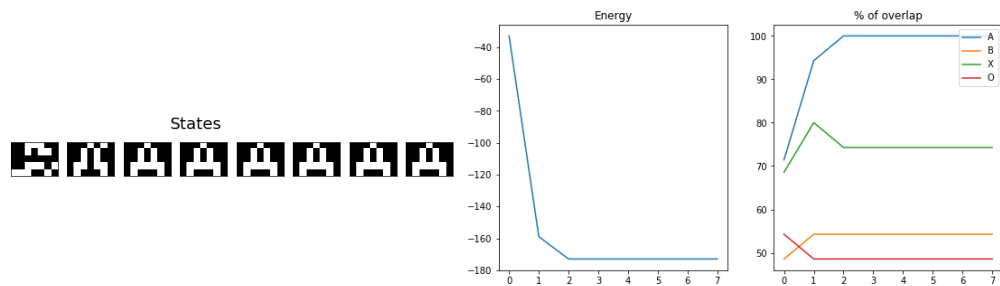
Pre každý vzor a pre každú úroveň zašumenia uvádzam vývoj stavov, energie a percenta prekryvu so všetkými pamäťovými vzormi.

A

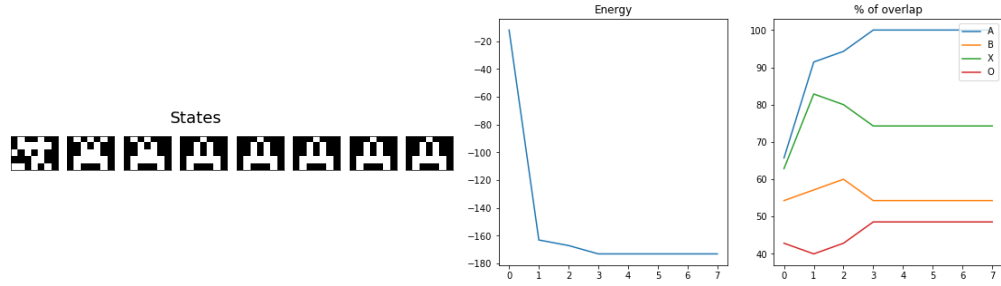
noise level 7



noise level 14

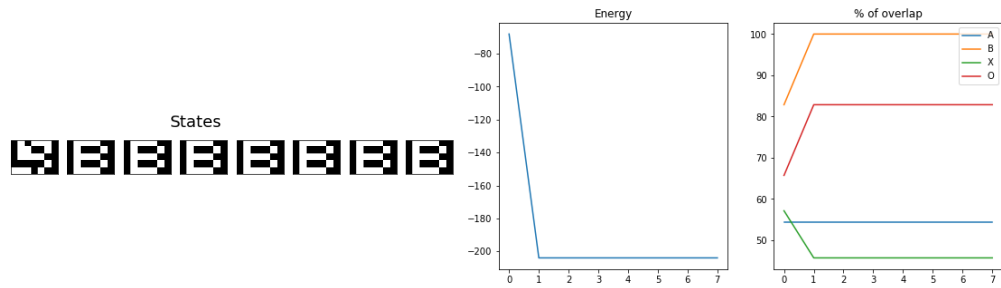


noise level 21

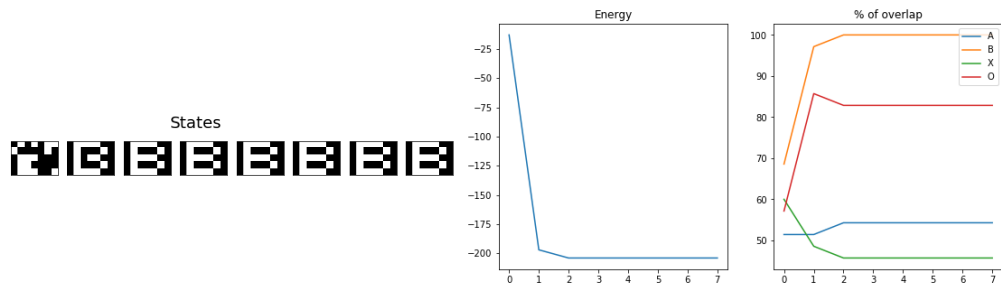


B

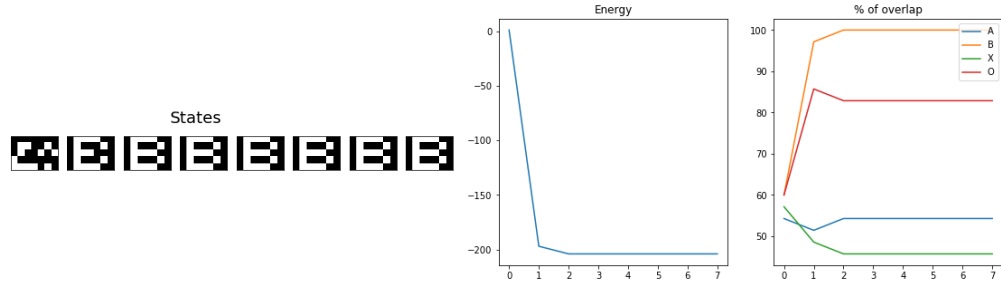
noise level 7



noise level 14

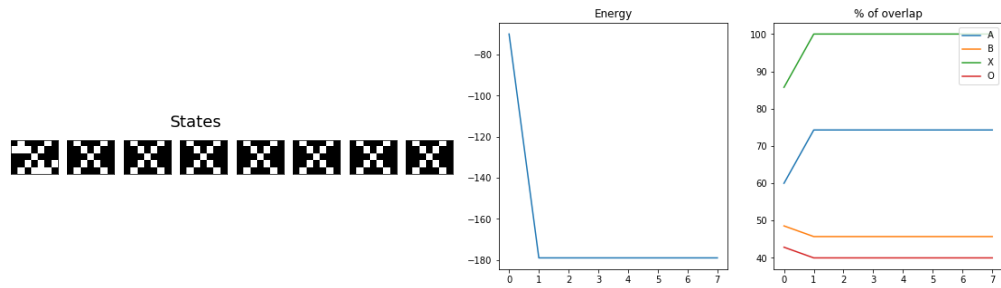


noise level 21

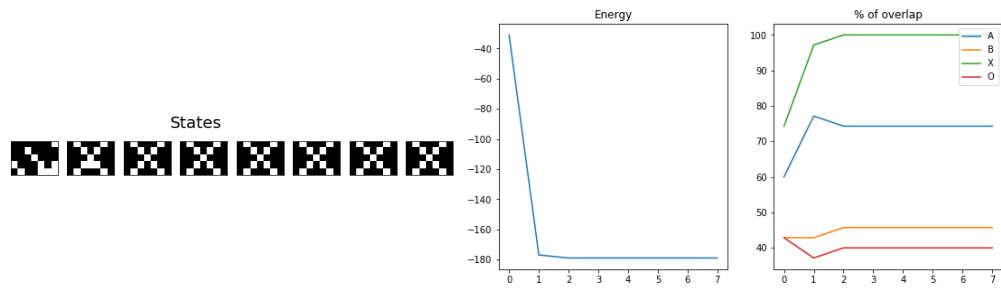


X

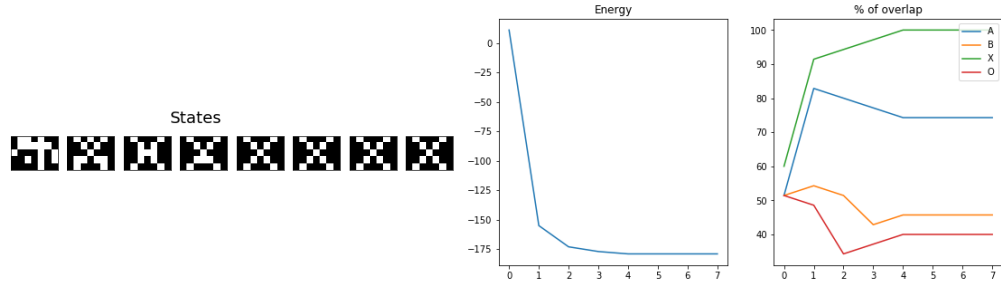
noise level 7



noise level 14

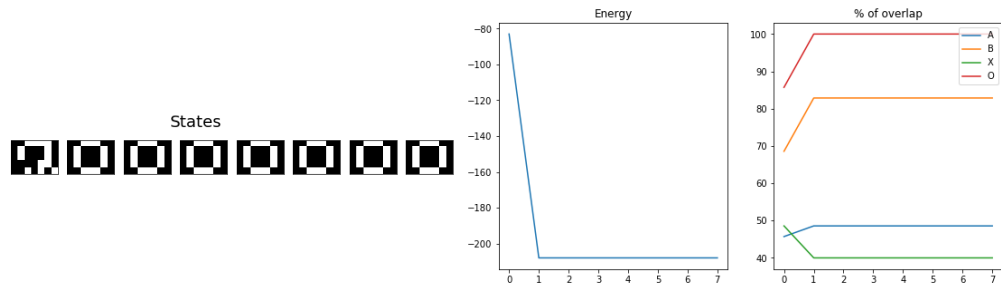


noise level 21

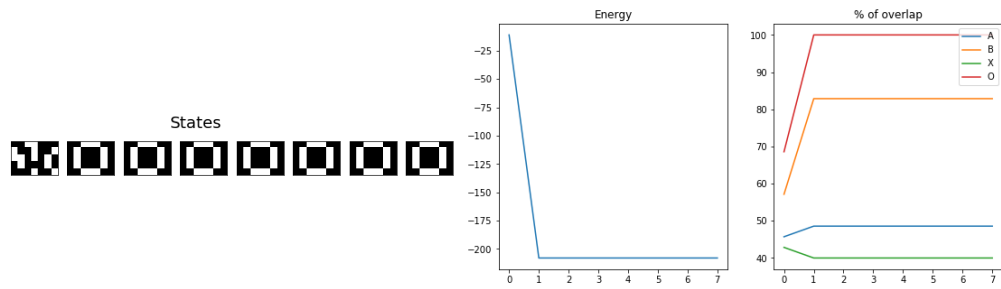


O

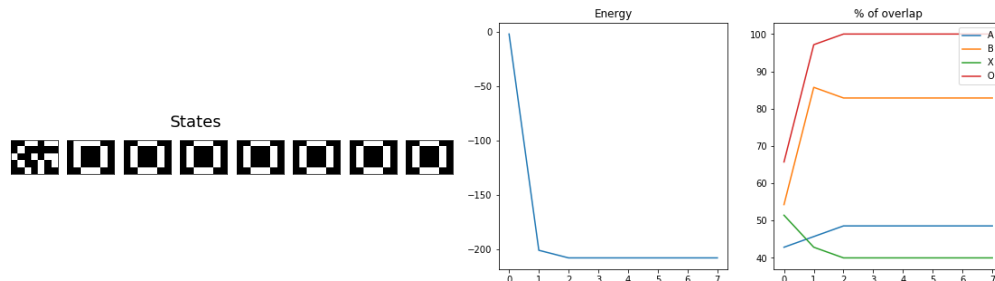
noise level 7



noise level 14



noise level 21



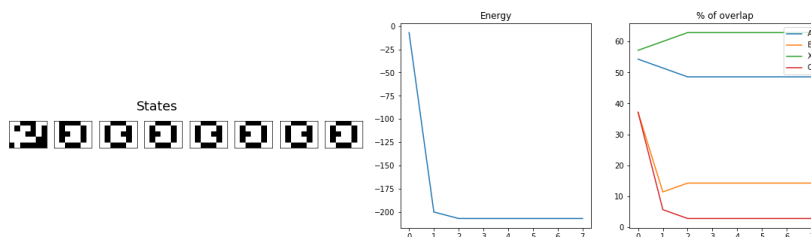
Pozorovania

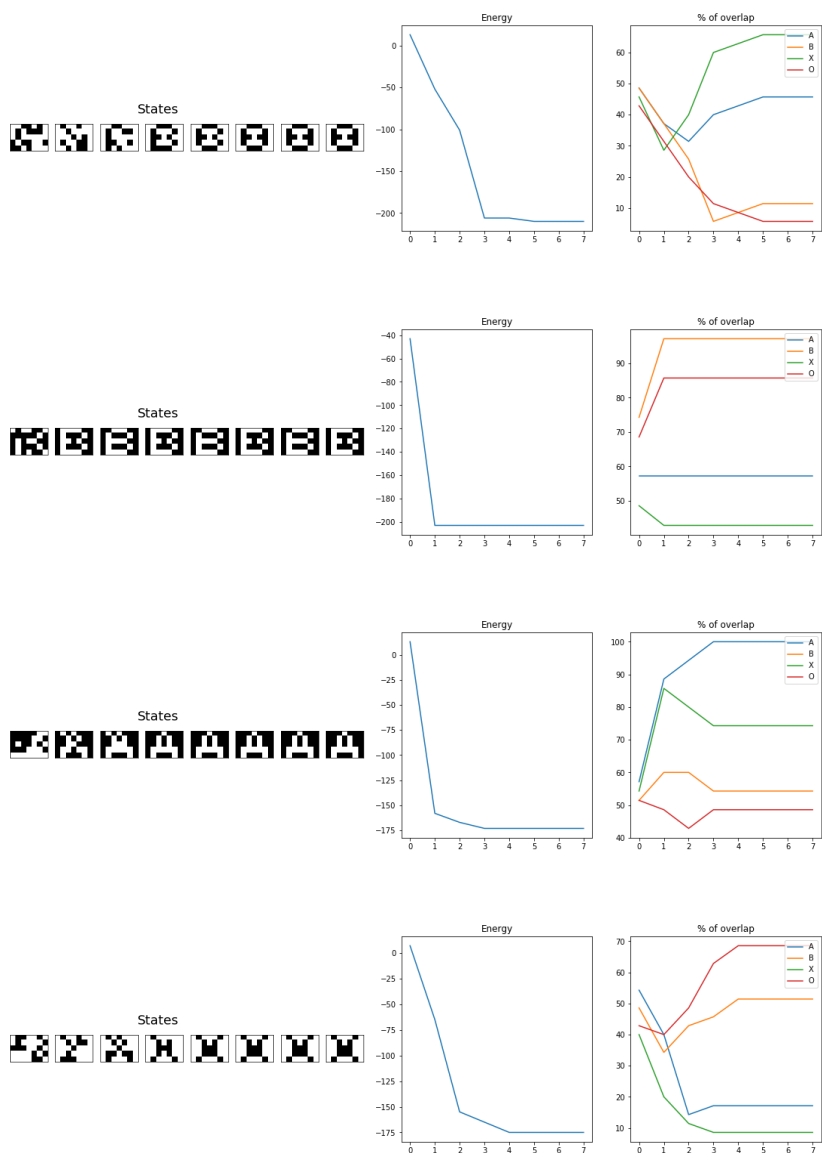
Podľa očakávania s každým ďalším krokom klesá (lepšie povedané nerastie) energia a rastie (neklesá) percento prekryvu s odpovedajúcim päťovým vzorom. Väčšinou to zkonvergovalo do daného atraktorového stavu v priebehu dvoch-troch krokov. Ďalšia vec je, že som vybral iba tie prípady kedy vstup zkonvergoval ku prisluchajúcemu pamäťovému vzoru. Teda čím väčší šum tým som viac krát musel spustiť skript.

Existuje vzťah medzi veľkosťou prekryvov v krokoch 1 a n? Percento prekryvu s prisluchajúcim päťovým vzorom v krokoch 1 a n neklesá.

Náhodný vstup

Podľa zadania som vygeneroval 11 nezávislých pozorovaní z alternatívneho rozdelenia a tie potom dával ako vstup do siete. Následne som nechal sieť bežať osem krokov. Tak ako v predchádzajúcom som vykreslil vývoj stavu, energie a percentá prekryvu s jednotlivými pamäťovými vzormi. Nižšie sú uvedené obrázky pre 5 takýchto vstupov. Snažil som sa aby som uviedol príklad pre každý typ finálnej konfigurácie. Teda skutočný atraktor, falošný atraktor a limitný cyklus.





Pozorovania

Podľa očakávania energia so zvyšujúcim sa počtom krokov nerastie.
Sformulujte všeobecné tvrdenie ako miera podobnosti medzi vstupným a pamäťovým vzorom ovplyvňuje výsledok.
 Podľa hore uvedených výsledkov ako pre poškodený vstup tak pre náhodný vstup to vyzerá tak, že systém zkonverguje do práve toho pamäťového vzoru,

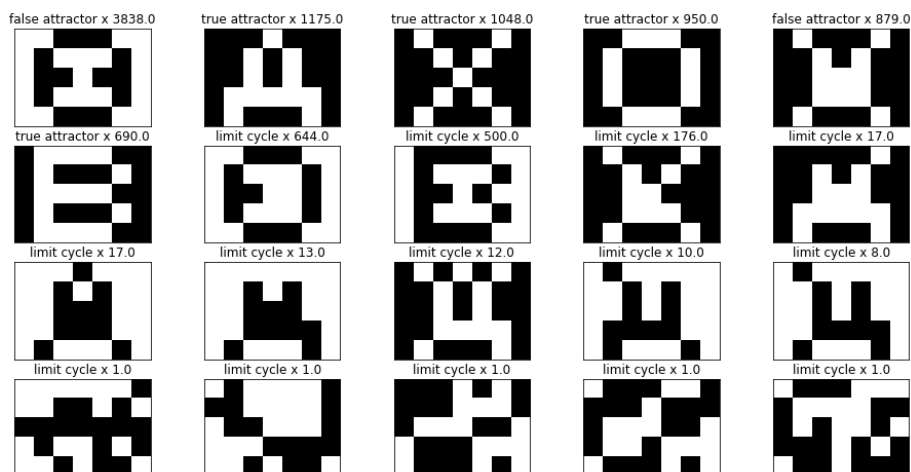
ktorý má najväčší prekryv so vstupom. Ak teda zkonverguje do skutočného atraktora a nie do falošného atraktora alebo limitného cyklu. Aj keď napríklad pri vstupe poškodeného B s úrovňou šumu 21 (čo už považujem za náhodný vstup) mal na začiatku rovnakú úroveň prekryvu s B aj O a zkonvergoval ku B. Čiže ako tvrdenie by som to neformuloval.

Finálne konfigurácie 10000 náhodných vstupov

Skúšal som do siete dávať 10000 náhodných vstupov a následne nechal sieť bežať kým nezkonvergovala do nejakej finálnej konfigurácie. Finálne konfigurácie boli iba 3. Skutočný atraktor, falošný atraktor a limitný cyklus dĺžky 2. Ani chaotické trajektórie ani limitné cykly väčšie ako 2 nevznikli.

Potom som urobil klasifikáciu jednotlivých finálnych vzorov a nechal vykresliť 20 najčastejších spolu s početnosťou a typom finálnej konfigurácie. Za rovnaký finálny vzor som považoval aj tie ktoré sa líšili iba invertovanou "farbou". Tak tiež limitné cykly neboli klasifikované podľa posledného vzoru, ale podľa oboch limitných vzorov a ich farebných inverzií. V niektorých limitných cykloch sa iba invertovali farby v dvoch po sebe idúcich krokoch a tie som považoval za limitný cyklus a nie za falošný atraktor.

V tabuľke sú uvedené celkové počty finálnych konfigurácií.



| Typ finálnej konfigurácie | Počet |
|-------------------------------|-------|
| <i>Skutočný atraktor</i> | 3863 |
| <i>Falošný atraktor</i> | 4747 |
| <i>Limitný cyklus dĺžky 2</i> | 1420 |
| Spolu | 10000 |

Bonus

Vytvoril som si ďalších 7 vzorov a tie postupne vkladal do siete a testoval vyvolávaciu schopnosť ako v prvej časti (pre poškodené vstupy vzorov A, B, X O pre úroveň šumu 7,14,21). Dole obrázok je kompletná abeceda, ktorú som používal..

Celá abeceda

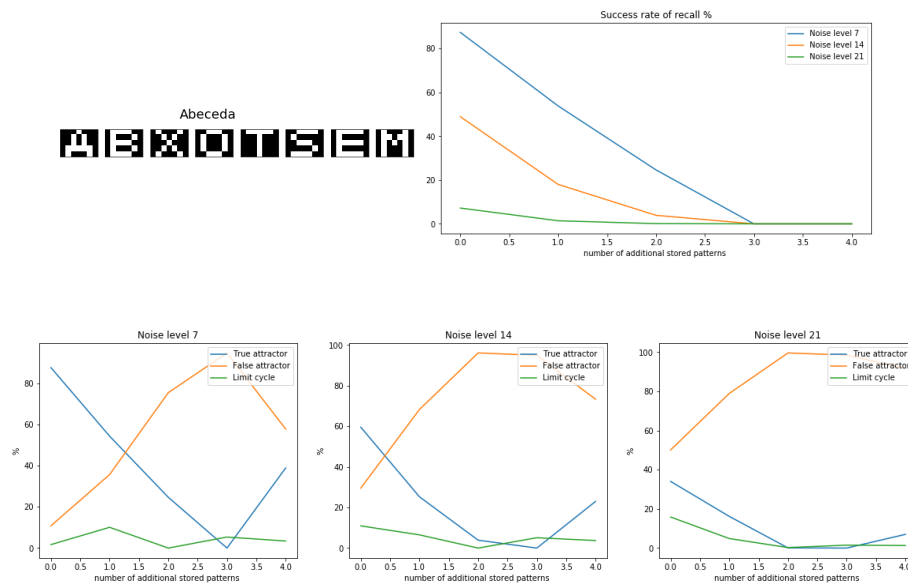


Vkladanie vzorov nebolo nenáhodné, ale vkladal som najprv tie vzory, ktoré maximalizovali vyvolávaciu schopnosť siete. Vloženie niektorého vzoru napríklad úplne zrušilo vyvolávaciu schopnosť a iné ju dokonca vedeli zvýšiť ako napríklad pri X. Všeobecne, ale pridávanie nových vzorov do siete znižovalo vyvolávanie. Vyvolávanie som testoval 10000 krát pre každú množinu zapamätaných vzorov a v grafe znázornil percento úspešného vyvolania.

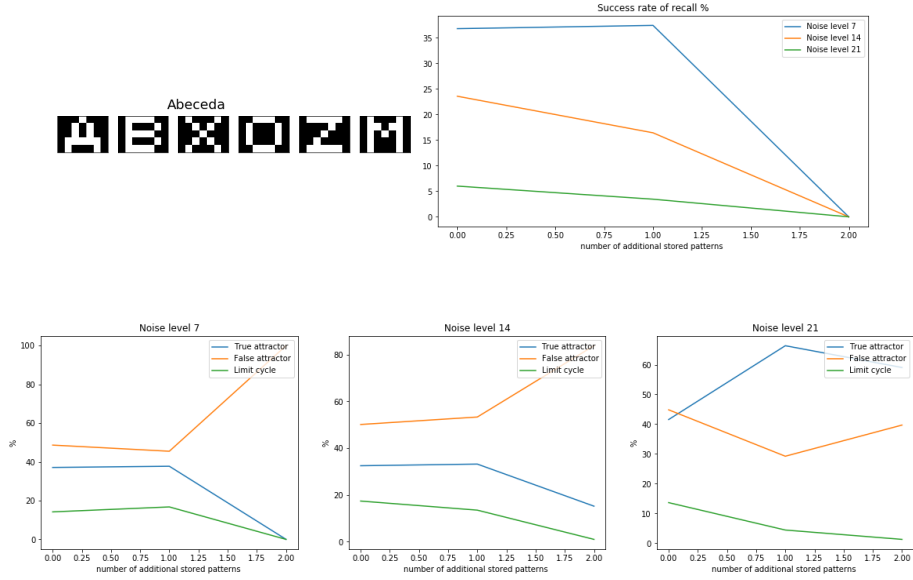
Default bola množina vzorov { A,B,X,O } (v grafoch $x=0$).

Ďalej som pre každú úroveň zašumenia vstupu znázornil percento finálnych konfigurácií zo všetkých 10000 testovaných vstupov. To, že percento skutočných atraktov nie je 0 aj keď úspešnosť vyvolávanie je nulová znamená, že vstup zkonvergoval síce ku skutočnému pamäťovému vzoru, ale nesprávne.

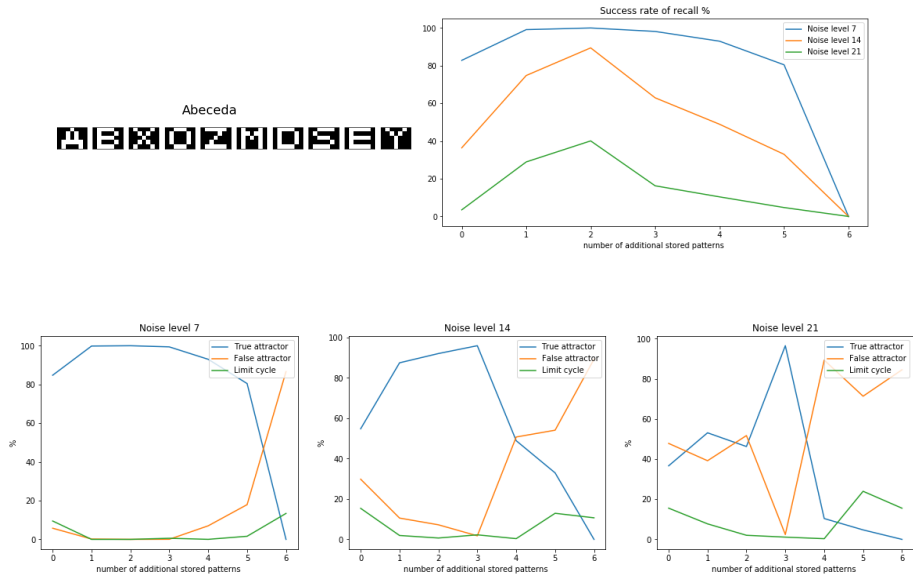
A



B

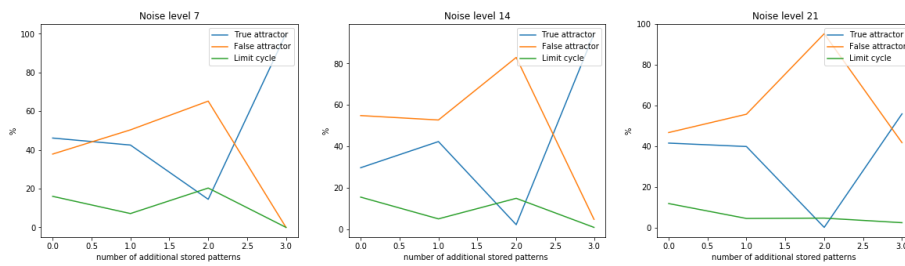
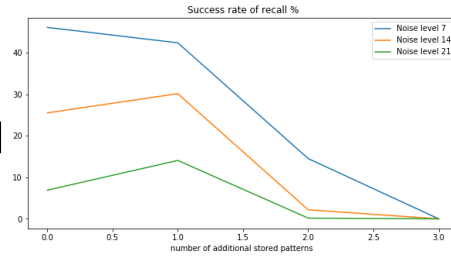


X



O

Abeceda



Poznámky

Uvedomujme si, že veľkosť obrázkov je malá a popisky sú zle čitateľné, ale je obrázkov je strašne veľa tak som sa to snažil zkompaktiniť.

Príkladám ku pdf nasledujúce zdrojáky:

hopfield.py - úlohy A a B

hopfield_C.py - úloha C

hopfield_bonus.py - bonus