

Sieć CounterPropagation i LVQ

Uwaga: Pliki z ciągami uczącymi i testowymi znajdują się w katalogu ~/cp

Proszę skopiować zawartość tego katalogu do Państwa podkatalogu w katalogu inf.

Po uruchomieniu programu nw2 w Państwa podkatalogu proszę wykonać poniższe punkty:

A. Sieć CounterPropagation z warstwą perceptronową

1. Uruchomić opcję menu **InstaNet**, wybrać **Self Organizing Map**. Ustawić **Inputs** 9, **# row** 3, **# col** 3, **Hidden** 0, **Output** 3. Odznaczyć **Coor. Layer**, **MinMax Table**, **Interpolate** (zmienić z białych na niebieskie). Opcja **Output Network** MA POZOSTAĆ ZAZNACZONA! W polu **Mapping Layer** ustawić **Delta Rule** i **Sigmoid**. Jako ciąg uczący ustawić plik cp_ucz2.nna, a jako testowy cp2.nna (zawierają one odpowiednio 9 ikon i 3 ikonki o rozmiarze 3x3). Ustawić **Epoch** na 0. Po zatwierdzeniu (OK) pojawi się dodatkowe okno, w którym należy wybrać **SOM Weights**.
2. Wyedytować instrument **SOM Weights**. Zmienić wygląd instrumentu : # x 3, rozmiar 400x50 (**height** i **width**).
3. Wyedytować warstwę wejściową (należy wybrać ikonę Layer z LEWEGO paska ikon, z rozwiniętego menu wybrać Edit i zmienionym kursorem kliknąć na warstwę wejściową) i zmienić ułożenie neuronów przez zmianę **# rows** z 1 na 3.
4. Przeprowadzić próbę nauczania sieci dla 5000 kroków.
5. Doprowadzić do nauczania sieci poprzez zmniejszenie rozmiaru sąsiedztwa w warstwie Kohonena (Wejść do opcji menu **Utilities/Global L/R Schedule**. Wyedytować parametry dla zestawu **som**. Zmniejszyć **Neighbourhood width** w kolumnach 1-4)
6. Przeprowadzić testowanie plikiem uczącym (**Run/Test...** z opcją **One/Pass All**). Sprawdzić i zinterpretować wagi warstwy wyjściowej w połączeniu z wynikami umieszczonymi w pliku o nazwie pliku uczącego i rozszerzeniu **nnr**. Następnie ponownie przetestować sieć (**Recall** z opcją **One**) i zaobserwować, które neurony „reagują” na kolejne ikonki wejściowe.
7. Przetestować sieć plikiem testowym zawierającym ikonki niewiele różniące się od tych ze zbioru uczącego. Sprawdzić czy „reagują” właściwe neurony.

B. Zmiana typu warstwy wyjściowej – klasyczna sieć CounterPropagation

Proszę: Wyedytować warstwę wyjściową i zmienić regułę uczenia i funkcję transferu warstwy wyjściowej na odpowiednio: **Widrow-Hoff** i **Linear** (z **Delta Rule** i **Sigmoid**).

Powtórnie nauczyć i przetestować sieć.

Proszę porównać wyniki działania sieci z części A z siecią „klasyczną”.

Następnie proszę powtórzyć części A i B dla zbioru uczącego cp_ucz3.nna i testowego cp3.nna . W celu nauczania sieci konieczna będzie dalsza modyfikacja parametrów uczenia. Tym razem w celu nauczania sieci oprócz zmniejszenia sąsiedztwa należy zwiększyć parametr **Alfa**.

Proszę wyjaśnić dlaczego drugi zbiór uczący (cp_ucz3) sprawia sieci więcej problemów, oraz wyjaśnić jaka jest rola pierwszej „zerowej” kolumny w zestawie parametrów uczenia **out** (która warstwa korzysta z tego zestawu?).

W projekcie symulatora należy zaimplementować uczenie metodami Widrowa-Hoffa i jej uogólnienie Delta Rule. Proszę porównać działanie Państwa symulatora na danych ćwiczenia z wynikami uzyskanymi w punktach A i B.

C. Problem parzystości

Proszę zbudować sieć typu CounterPropagation będącą w stanie rozwiązać problem parzystości trzech bitów (na trzech wejściach mogą pojawiać się zera i jedynki - na wyjściu powinno pojawić się zero gdy ilość jedynek na wejściu jest nieparzysta, a jeden gdy jest parzysta). W tym celu proszę:

1. Stworzyć odpowiedni ciąg uczący
2. Zbudować sieć posługując się opcją InstaNet: Uruchomić opcję menu InstaNet, wybrać Self Organizing Map. Ustawić Inputs 3, # row 1, # col 9, Hidden 0, Output 1. Pozostałe ustawienia takie same jak w punkcie A 1.
3. Doprowadzić do nauczania sieci i przetestować sieć.
4. Zmienić sieć na „klasyczną (Zmodyfikować regułę uczenia i funkcję transferu warstwy wyjściowej na odpowiednio: **Widrow-Hoff** i **Linear**).
5. Przeprowadzić ponowne uczenie. Przetestować sieć.

Proszę wyjaśnić w jaki sposób sieć CounterPropagation rozwiązuje problem parzystości. Przy jakiej najmniejszej ilości neuronów w warstwie Kohonena sieć ta jest w stanie rozwiązać problem parzystości?

Proszę zaprezentować działanie Państwa symulatora w problemie parzystości.

D. Sieć LVQ

Proszę:

1. Uruchomić opcję menu InstaNet, wybrać Learning Vector Quantization. Ustawić Inputs 9, Kohonen 9, Output 3. Ustawić nazwę ciągu uczącego i testowego tak jak w punkcie A 1. Po zatwierdzeniu (OK) pojawi się dodatkowe okno w którym należy wybrać Kohonen Weights.
2. Wyedytować instrument Prototype Weights. Zmienić wygląd instrumentu : # x 3, rozmiar 400x50 (height x width).

Następne punkty (3-7) proszę wykonać wg opisu z części A.

Proszę wyjaśnić rolę połączeń pomiędzy warstwą Kohonena a warstwą wyjściową.

Sieci LVQ nie trzeba implementować w Państwa symulatorze.