

Metody Sztucznej inteligencji

30h laboratorium dla kierunku Informatyka

Marcin Korzeń

*Politechnika Szczecińska, Wydział Informatyki,
Katedra Sztucznej Inteligencji i Metod Matematycznych
mkorzen@wi.ps.pl*

Spis treści

1	Wprowadzenie do programu Matlab i pakietów: Netlab i Neural Network Toolbox	2
2	Perceptron	2
3	Sieci neuronowe jednokierunkowe warstwowe	3
4	Klasyfikacja i regresja	4
5	Sieci neuronowe jednokierunkowe (ang. <i>feed-forward</i>, <i>MLP</i>)	4
6	Sieci neuronowe typu RBF	5
7	Sieci rekurencyjne	6
8	Wprowadzenie do logiki rozmytej	7
9	Identyfikacja funkcji przynależności	7
10	Lingwistyczne bazy wiedzy	7
11	Sterownik rozmyty Mamdaniego	7
12	Regulacja rozmyta	7
13	Robocode	7

Rozliczenie zajęć

Rozliczenie zajęć odbywa się na podstawie:

1. Obecności i czynnej pracy na zajęciach
2. Ocen z wejściówek
3. Ocen z zadań praktycznych i sprawozdań

Część materiału i zadań ma charakter dodatkowy co na leży rozumieć: trzeba czytać i robić więcej jeżeli chce się mieć ocenę powyżej 4.0.

1 Wprowadzenie do programu Matlab i pakietów: Netlab i Neural Network Toolbox

Wymagania Podstawowa znajomość programowania w Matlabie oraz znajomość podstawowych funkcji do obliczeń numerycznych.

Zadania do wykonania

1. Nauczyć się wczytywać dane różnych formatów do Matlab'a, dane można znaleźć tu:
 - <http://mlr.cs.umass.edu/ml/datasets.html>
 - <http://weka.sourceforge.net/wiki/index.php/Datasets>
 - <http://lib.stat.cmu.edu/>
 - <http://www.inf.ed.ac.uk/teaching/courses/dme/html/datasets0405.html>
2. Zapoznać się z pakietem Neural Network Toolbox (zaczynając od: » `help nnet`)
3. Zapoznać się z pakietem Netlab (autorzy: Ian Nabney, Christopher Bishop) dostępny pod adresem: <http://www.ncrg.aston.ac.uk/netlab/book.php>
4. Nauczyć się wykonywać w Matlab'ie znane ze statystyki i metod numerycznych metody jak: wyznaczanie podstawowych statystyk, obliczanie macierzy korelacji, PCA, regresja liniowa, rozwiązywanie zadania optymalizacji, itp.
5. Nauczyć się rysowania wykresów funkcji jednej i dwóch zmiennych, rysowania histogramów, itp.

literatura i uwagi Wszystkie potrzebne informacje można znaleźć na stronie <http://www.mathworks.com/access/helpdesk/help/techdoc/>.

2 Perceptron

Wymagania Zbiór liniowo separowalny, margines separacji. Perceptron, struktura sieci, model neuronu, reguła perceptronu. Zbieżność perceptronu, twierdzenie Novikoff'a. Interpretacja geometryczna wag.

Zadania do wykonania

1. Napisać algorytm uczenia perceptronu, funkcja powinna brać na wejście dane uczące, oraz zwracać wagi neuronu/ów i wartość/ści progowe. Prototyp funkcji może wyglądać następująco: `[w,b]=regulaPerceptronu(X,d)`. Uwaga funkcja powinna w jakiś sposób zabezpieczać się przed danymi nieseparowalnymi np. ustawiając maksymalną liczbę kroków algorytmu na 1e6 lub uzależniając stop od czasu działania.
2. Wygenerować dowolne dane liniowo seprowalne¹ np:
`X=rand(100,2);`
`d=2*(X(:,1)-X(:,2)>0)-1;`

¹Uwaga: Kolejne dane mogą być zapisane wierszami jak w przykładzie lub kolumnami – ten sposób preferowany jest przez pakiet Neural Network Toolbox. Oba sposoby zapisu są również często spotykane w literaturze. Należy mieć to na uwadze transponując odpowiednio wzory w sytuacjach, gdy jest to potrzebne.

3. Uruchomić algorytm
 $[w, b] = \text{regulaPerceptronu}(X, d)$
 4. Zwizualizować wynik działania algorytmu.
 5. Poeksperymentować z różnymi danymi wejściowymi np. zwiększając rozmiar zbioru uczącego zachowując margines separacji, zmieniając rozkład danych.
 6. Wyznaczyć maksymalny margines w danych oraz minimalny promień danych, a następnie sprawdzić, że zachodzi teza twierdzenia Novikoff'a.
- * Zmodyfikować algorytm tak, aby w sensowny sposób zachowywał się dla danych nieseparowalnych. Poeksperymentować z różnymi zbiorami danych.

literatura i uwagi Większość informacji można znaleźć w pracach: [4, rozdział 1.], [2, Rozdział 5], przeczytać ze zrozumieniem [2, §5.3]. Książkę [4] można traktować jako podstawowy podręcznik do sieci neuronowych na naszych zajęciach.

3 Sieci neuronowe jednokierunkowe warstwowe

Wymagania Struktura sieci, model neuronu, uczenie metodą wstecznej propagacji błędu w postaci prostej (on-line) i wsadowej (off-line), przegląd zastosowań.

Zadania do wykonania

1. Napisać algorytm² uczenia sieci neuronowej metodą wstecznej propagacji błędu w postaci on-line, dla pojedynczej warstwy neuronów. Prototyp funkcji może mieć postać: $[w, b] = \text{backprop}(X, d)$, gdzie (X, d) to pary uczące.
 2. Zmodyfikować algorytm do tak, aby pracował w trybie wsadowym. Powinna wystarczyć tu zmiana pojedynczej linii, podmienioną linię wziąć w komentarz.
- * Zmodyfikować algorytm do tak, aby pracował dla sieci dwuwarstwowej, proponuje tryb wsadowy oraz w miejsce zwykłej wstecznej propagacji błędu wykorzystać jakiś rodzaj gradientów sprzężonych. Wywołanie może mieć postać: $[w1, b1, w2, b2] = \text{backprop}(X, d, K)$, gdzie K określa liczbę neuronów warstwy ukrytej, do obsługi funkcji ze zmienną liczbą argumentów można użyć stałej `margin`.
3. Napisać funkcję, która dla zadanych wag, oraz próbek wejściowych wyznaczy wartościowe sieci neuronowej. Funkcja może obsługiwać zmienną liczbę argumentów:
 $y = \text{mlp}(X, w, b)$
 $y = \text{mlp}(X, w1, b1, w2, b2)$
- * Napisać funkcje do wizualizacji, wywołania: `sepline(X, d, w, b)` lub `sepline(X, d, w1, b1, w2, b2)`. Funkcja powinna ona rysować prostą separacji na tle próbek uczących w przypadku sieci jednowarstwowej, oraz linię separacji w przypadku sieci dwuwarstwowej. Zastanowić się nad przypadkiem gdy danych wejściowe są więcej niż dwuwymiarowe, wtedy można posłużyć się metodą PCA. (wskazówka: przypadek danych dwuwymiarowych jest prosty należy posłużyć się funkcjami: `meshgrid` i `contour`.)
4. Poeksperymentować z różnymi zbiorami danych, porównać działanie z algorytmem reguła perceptronu.

²Proszę wziąć pod uwagę, że podstawowym typem Matlab'a są macierze, dla których przeciężono odpowiednie działania i funkcje, czyli należy korzystać z wygodnych notacji wektorowych i nie powinno się nadużywać pętli `for`.

literatura i uwagi Większość informacji można znaleźć w pracach: [4, rozdział 2.], parz również [2].

4 Klasyfikacja i regresja

Wymagania Ocena dokładności maszyny uczącej, testowanie, krosvalidacja, MAE, MSE, dokładność klasyfikacji, błąd klasyfikacji, czułość, specyficzność, krzywa ROC, AUC.

Zadania do wykonania

1. Wczytać do Matlab'a dowolne dane dotyczące problemu klasyfikacji dla dwóch klas.
 2. Podzielić dane na trzy części uczącą L , walidującą V i testującą T
 3. Nauczyć na danych uczących następujące liniowe maszyny uczące: perceptron, MLP (sieć jednowarstwowa różne algorytmy uczenia dostępne w matlabie), (ewentualnie SVM, regresja logistyczna), powinniśmy dostać różne zestawy wag prostej separującej w , b , oraz różne błędy klasyfikacji³.
 4. Wybrać najlepszy z modeli używając próby walidującej, a następnie ocenić jego dokładność używając próby testowej
- * Napisać funkcję która dla zadanych wag i danego zbioru testowego narysuje krzywą ROC.
- * Obliczyć pole pod ta krzywą.

literatura i uwagi Większość informacji na ten temat można znaleźć w książce: [3, §2.3.2], zachęcam jednak do szerszego zapoznania się z tą pozycją.

5 Sieci neuronowe jednokierunkowe (ang. *feed-forward*, *MLP*)

Wymagania MLP, struktura sieci model neuronu. Uczenie sieci metodą wstecznej propagacji błędu i jej uogólnienia: dodanie czynnika momentu, minimalizacja kierunkowa, gradienty sprzężone, metody drugiego rzędu. Optymalna prosta separująca dla danych linowo separowalnych i nieseparowalnych, metoda SVM.

Zadania do wykonania

1. Pobrać dane dowolne dane z repozytorium UCI:
<http://archive.ics.uci.edu/ml/datasets.html>.
Powinny dotyczyć one interesującego zagadnienia o którym mamy pewną wiedzę (np. wine).
2. Wczytać dane do Matlab'a
3. Podzielić dane na dwie części: część uczącą (P, T) i część testującą (PT, TT)

³Należy rozróżniać błąd sieci (w przypadku MLP jest to zwykle MSE lub SSE, w przypadku regresji logistycznej to DEV lub log-likelihood) oraz błąd klasyfikacji

4. Podzielić zbiór uczący (P, T) na część uczącą (PU, TU) i walidującą (PV, TV)
5. Nauczyć sieć neuronową na całym zbiorze uczącym (bez podziału na części uczącą i walidującą) i dokonać oceny błędu na zbiorze testowym
6. Nauczyć sieć neuronową na całym zbiorze uczącym z wykorzystaniem zbioru walidującego i dokonać oceny błędu na zbiorze testowym
 - * Dokonać wizualizacji danych (w rzutach na wybrane podprzestrzenie lub lepiej używając PCA) oraz wizualizacji modelu.
 - * Zastosować uczenie sieci metodą SVM
7. Dokonać oceny modelu: oszacowanie błędu klasyfikacji (odsetek błędnych klasyfikacji na zbiorze testowym), krosswalidacja, AUC.
8. Powtórzyć uczenie dla sieci jednowarstwowych i dwuwarstwowych (z przyjętą arbitralnie liczbą neuronów ukrytych), porównać różne rodzaje sieci neuronowych oraz porównać różne algorytmy uczenia, do porównania można wykorzystać również program WEKA, wyniki zebrać w tabeli.
 - * Dobrać liczbę neuronów warstwy ukrytej z wykorzystaniem zbioru walidującego.
 - * Spróbować dokonać oceny istotności i współzależności zmiennych. (do oceny współzależności można użyć współczynnika korelacji liniowej miary przyrostu informacji lub innych, istotność można ocenić biorąc pod uwagę jaki wpływ na jakość modelu ma usunięcie jednej lub kilku zmiennych.)
9. Analiza i wnioski dotyczące eksperymentu (należy brać pod uwagę wielkość błędu, złożoność modelu, praktyczną przydatność takiego klasyfikatora w konkretnym przypadku, dokona jakościowej analizy).

literatura i uwagi Większość informacji można znaleźć w pracach: [4, rozdział 2.], o metodzie maszynach wektorowych (SVM) można przeczytać w [1] lub [3, §6.2].

6 Sieci neuronowe typu RBF

Wymagania Sieci RBF, struktura sieci, model neuronu, uczenie sieci (samorganizacja + regresja), algorytm K-środków, algorytm EM, mieszanka modeli gaussowskich (GMM), sieci probabilistyczne, optymalny klasyfikator Bayesowski

Zadania do wykonania

1. Pobrać dane `autoPrices.txt` ze strony `ksir.wi.ps.pl` lub inne dowolne dane z repozytorium UCI:
<http://archive.ics.uci.edu/ml/datasets.html> dotyczące zadania regresji.
2. Wczytać dane do Matlab'a
3. Dokonać normalizacji zmiennych: zmienne przyjmujące wartości różnych znaków normalizujemy do przedziału $[-1, 1]$ z zachowaniem zera, zmienne przyjmujące wartości dodatnie normalizujemy do zakresu $[0, 1]$.

4. Podzielić dane na dwie części: część uczącą (PU, TU) i część testującą (PT, TT)
5. Nauczyć sieć neuronową typu RBF na całym zbiorze uczącym (bez podziału zbioru uczącego na części uczącą i walidującą) i dokonać oceny błędu na zbiorze testowym.
6. Dokonać oszacowania błędu sieci RBF (mse lub mae)
7. Powtórzyć uczenie dla sieci z różną liczbą neuronów RBF.
8. Analiza istotności poszczególnych atrybutów (zmiennych) warunkowych.
9. Stworzyć dla porównania standardowy model regresji liniowej, porównać błędy (mse lub mae), spojrzeć na współczynniki i stad wnioskować nt. istotności zmiennych.
10. Analiza i wnioski dotyczące eksperymentów (należy brać pod uwagę wielkość błędu, praktyczną przydatność prognozy (np. szacowanie ceny samochodu dla potrzeb komisju samochodowego), dyskusję czym spowodowany jest błąd, dyskusję nt. brakujących czynników mających wpływ na błąd prognozy, itp.).

Zadania dodatkowe:

1. Dokonać wizualizacji danych (w rzutach na wybrane podprzestrzenie lub używając PCA)
2. Spróbować dokonać oceny istotności i współzależności zmiennych. (do oceny współzależności można użyć współczynnika korelacji liniowej, istotność można ocenić biorąc pod uwagę jaki wpływ na jakość modelu ma usunięcie jednej lub kilku zmiennych)
3. Dobrać liczbę neuronów warstwy ukrytej lub szerokość neuronów RBF z wykorzystaniem zbioru walidującego.

7 Sieci rekurencyjne

Wymagania Sieci Hopfielda, struktura sieci model neuronu, tryb pracy, tryb odtwarzania, reguła Hebba, pamięci skojarzeniowa, funkcja Lapunowa

- 8 Wprowadzenie do logiki rozmytej
- 9 Identyfikacja funkcji przynależności
- 10 Lingwistyczne bazy wiedzy
- 11 Sterownik rozmyty Mamdaniego
- 12 Regulacja rozmyta
- 13 Robocode

Literatura

- [1] Ch J. C. Burges. A Tutorial on Support Vector Machines for Pattern Recognition, in *Data Mining and Knowledge Discovery*, **2**, 121–167 (1998), Kluwer Academic Publishers, Boston.
- [2] J. Hertz, A. Krogh, R. G. Palmer *Wstęp do teorii obliczeń neuronowych*. WNT, Warszawa, 1993.
- [3] J. Koronacki, J. Ćwik, *Statystyczne systemy uczące*. WNT, Warszawa 2005.
- [4] S. Osowski. *Sieci neuronowe w ujęciu algorytmicznym*. WNT, Warszawa, 1998.
- [5] A. Piegat, *Modelowanie i sterowanie rozmyte*. Akademicka oficyna wydawnicza EXIT, Warszawa 2000.
- [6] I. H. Witten, E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, 2005