

THE ARTIFICIAL NEURAL NETWORKS AS A HELPING TOOL IN THE PROCESS OF NUMERICAL AGRICULTURAL ENGINEERING PROBLEMS

Summary

The discretization process of the continuous differential issue (with the initial-border conditions) leads to obtaining the linear set of algebraic equations. To resolve such a set of equations, the knowledge about the inverted form of system matrix is required. One-directional neural networks can be effectively used in matrix algebra to conduct lots of standard matrix operations, including matrix inversion. The neural models listed above during exploitation let to obtain a great functional speed (nearly real time work). The basic problem, in mentioned context, is the proper definition of an energetic function, minimalization of which lets to design, generate and learn the proper neural network topology. The aim of work was analysis of the possibilities of using modern techniques of artificial neural networks to generate the inverted matrix form.

SZTUCZNE SIECI NEURONOWE JAKO NARZĘDZIE WSPOMAGAJĄCE PROCES NUMERYCZNEGO PRZETWARZANIA W PROBLEMACH INŻYNIERII ROLNICZEJ

Streszczenie

Proces dyskretyzacji ciągłego zagadnienia różniczkowego (wraz z warunkami początkowo-brzegowymi) prowadzi do uzyskania liniowego układu równań algebraicznych. Rozwiązanie takiego układu równań wymaga znajomości postaci macierzy odwrotnej układu. Jednokierunkowe sieci neuronowe mogą być efektywnie wykorzystane w algebrze macierzowej do realizacji wielu standardowych operacji macierzowych, w tym również do odwracania macierzy. Wymienione wyżej modele neuronowe pozwalają w trakcie ich eksploatacji na uzyskanie dużej szybkości działania (praktycznie działania w czasie rzeczywistym). Problemem zasadniczym, w powyższym kontekście, jest właściwe określenie funkcji energetycznej, której minimalizacja pozwala na zaprojektowanie, wygenerowanie oraz nauczenie odpowiedniej topologii sieci neuronowej. Celem pracy była analiza możliwości wykorzystania nowoczesnych technik sztucznych sieci neuronowych do generowania postaci macierzy odwrotnej.

Wstęp

Aby rozwiązać ciągle zagadnienia różniczkowe, modelujące złożone systemy inżynierii rolniczej, zazwyczaj należy skorzystać z metod przybliżonych, takich jak metoda różnic skończonych czy metoda elementów skończonych. Proces algebraizacji ciągłego zagadnienia różniczkowego (wraz z warunkami początkowo-brzegowymi) opisującego np. zagadnienie deformacji obciążonej płyty stropowej, prowadzi do układu liniowych równań algebraicznych. Układ taki można przedstawić za pomocą równania macierzowego postaci [6]:

$$\mathbf{AX} = \mathbf{B} \quad (1)$$

gdzie:

A - pasmowa, nieosobliwa oraz rzadka macierz globalna układu równań, uzyskanych w procesie dyskretyzacji zagadnienia ciągłego (zawierająca parametry materiałowe płyty),

X - kolumnowa macierz poszukiwanych niewiadomych (zawierająca składowe wektora deformacji płyty w punktach przyjętej siatki podziału płyty),

B - kolumnowa macierz wyrazów wolnych (reprezentująca obciążenie w punktach przyjętej siatki podziału płyty).

Rozwiązanie równania (1) można przedstawić w następującej postaci:

$$\mathbf{X} = \mathbf{A}^{-1}\mathbf{B} \quad (2)$$

gdzie: \mathbf{A}^{-1} – macierz odwrotna.

W tego typu problemach istotnego znaczenia nabiera zatem procedura odwracania macierzy, której znajomość

jest niezbędna dla uzyskania rozwiązania zagadnienia (1) prezentowanego w postaci (2).

Można dowieść, że jednokierunkowe sieci neuronowe mogą być efektywnie wykorzystane w algebrze macierzowej do realizacji wielu standardowych operacji macierzowych, takich jak na przykład: odwracanie macierzy, mnożenie macierzy, obliczanie wartości oraz wektorów własnych macierzy itp. Wymienione wyżej modele neuronowe, wykorzystując jedną ze swoich ważnych cech jaką jest możliwość przetwarzania równoległego, pozwalają w trakcie ich eksploatacji na uzyskanie dużej szybkości działania (praktycznie działania w czasie rzeczywistym). Problemem zasadniczym w procesie generowania sieci jest właściwe określenie funkcji energetycznej, której minimalizacja pozwala na zaprojektowanie, wytworzenie oraz nauczenie odpowiedniej topologii sieci neuronowej.

Celem prowadzonych badań była analiza możliwości wykorzystania nowoczesnych technik przetwarzania sygnału w postaci danych numerycznych, jakie reprezentują sztuczne sieci neuronowe, w szczególności do wykonania operacji odwracania macierzy.

Podstawy teoretyczne

Dana jest nieosobliwa macierz kwadratowa **A** rzędu n . Celem jest zaprojektowanie sieci neuronowej obliczającej macierz odwrotną do macierzy **A**, czyli wykonującą w trakcie eksploatacji przekształcenie danych wejściowych do postaci $\mathbf{B} = \mathbf{A}^{-1}$. Macierz odwrotna **B** winna zatem spełniać następujące równanie macierzowe [6]:

$$\mathbf{BA} = \mathbf{1} \quad (3)$$

$$\text{gdzie: } \mathbf{A} = [A_{ij}] = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1n} \\ x_{21} & x_{22} & \dots & x_{2n} \\ \dots & \dots & \dots & \dots \\ x_{n1} & x_{n2} & \dots & x_{nn} \end{bmatrix} \quad \text{oraz} \quad \mathbf{B} = [B_{ij}] = \begin{bmatrix} y_{11} & y_{12} & \dots & y_{1n} \\ y_{21} & y_{22} & \dots & y_{2n} \\ \dots & \dots & \dots & \dots \\ y_{n1} & y_{n2} & \dots & y_{nn} \end{bmatrix}$$

$$i=1,2,3,4,\dots,n,$$

$$j=1,2,3,4,\dots,n.$$

Mnożąc obustronnie równanie (1) przez dowolny niezerowy wektor $\mathbf{x} = [x_1, x_2, \dots, x_n]$ można otrzymać następujące równanie macierzowe:

$$\mathbf{BAx} - \mathbf{x} = \mathbf{0} \quad (4)$$

Rozwiązując powyższe równanie macierzowe można wyznaczyć macierz \mathbf{B} reprezentującą poszukiwaną macierz odwrotną. Pomysł wykorzystania sieci neuronowych do wyznaczenia macierzy \mathbf{B} sprowadza się do zdefiniowania pojęcia funkcji energetycznej w postaci błędu średniokwadratowego (dla wyznaczonego wzorca uczącego):

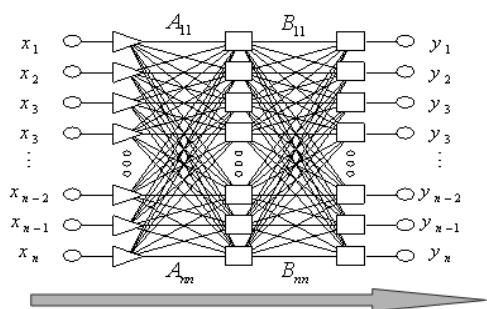
$$E = \frac{1}{2} (\|\mathbf{BAx} - \mathbf{x}\|)^2 \quad (5)$$

gdzie:

$\|\dots\|$ - norma euklidesowa.

Idea proponowanej postaci funkcji energetycznej w formie (5) sprowadza się do spełnienia postulatu, aby jej minimum odpowiadało optymalnemu rozwiązaniu problemu wyjściowego (4). Pozwala to procedurę rozwiązania równania macierzowego (2) zastąpić zagadnieniem minimalizacji funkcji E w postaci (5), co w można porównać z procesem nadzorowanego uczenia jednokierunkowej sieci neuronowej.

Proces uczenia sieci neuronowej łatwo można zinterpretować poprzez prezentację grafów przepływu sygnałów. Na rys. 1. przedstawiono strukturę sieci neuronowej modelującą przepływ sygnałów odpowiadający zdefiniowanej wyżej funkcji energetycznej E w formie (5). Wartości wag połączeń A_{ij} macierzy \mathbf{A} można interpretować jako wagi stałe (nie podlegające uczeniu) natomiast wagi B_{ij} macierzy poszukiwanej macierzy odwrotnej \mathbf{B} wagami podlegającymi procesowi „dostrajania” zgodnie z właściwym algorytmem uczącym.



Rys. 1. Jednokierunkowa sieć do odwracania macierzy \mathbf{A}
Fig. 1. One-directional network for inversion of matrix \mathbf{A}

Warto zauważyć, że po zakończeniu procesu uczenia spełniona jest następująca zależność:

$$\mathbf{y} = \mathbf{BAx} = \mathbf{1x} = \mathbf{x} \quad (6)$$

Fakt ten oznacza, że dowolny niezerowy wektor \mathbf{x} pełni w omawianym przypadku podwójną rolę: jest wektorem uczącym, stanowiącym sygnał wejściowy dla sieci neuronowej, a jednocześnie jest wektorem, jaki oczekiwany jest na wyjściu omawianej sieci (jest to więc sieć mająca charakter autoasocjacyjny).

Model neuronowy do generowania macierzy odwrotnej

Celem przeprowadzenia eksperymentu wykorzystano symulator jednokierunkowych sieci neuronowych *Statistica Neural Networks v. 4.0*. Jako zbiór danych przyjęto zbiór 1572 różnych przypadków. Stanowiły one macierze o wymiarach 4×4 w postaci 16 elementów, które stanowiły losowo wygenerowane liczby naturalne (0, 1, 2...9). Jako niezależne cechy wektorów uczących przyjęto położenie elementu w strukturze macierzy (np. cecha A44 reprezentowała element leżący w 4 wierszu oraz 4 kolumnie). Macierze te zostały odwrócone za pomocą standardowej procedury, zaimplementowanej w programie *Microsoft Office Excel 2003*. Strukturę wygenerowanego zbioru przypadków, wraz z jego podziałem na zbiory: uczący (1300 przypadków), walidacyjny (200 przypadków) oraz testowy (72 przypadki), przedstawia rys. 2.

W celu wygenerowania zbioru adekwatnych topologii sieci neuronowych posłużono się efektywną procedurą, zaimplementowaną w aplikacji symulatora *SNN v. 4.0.*, w postaci automatycznego projektanta sieci (w wersji zaawansowanej). Jest to silne narzędzie pozwalające efektywnie skrócić czas żmudnych poszukiwań adekwatnych topologii sieci neuronowych, uczonych w oparciu o posiadany zbiór danych. W efekcie uzyskano zestaw 10 wybranych topologii sieci neuronowych, który przedstawiono na rys. 3.

Najlepsze właściwości (najmniejszy błąd *RMS*) wykazała sieć typu *GRNN* (*Generalized Regression Neural Networks*) – sieć realizująca regresję uogólnioną. Struktura regresyjnej sieci *GRNN* oznaczonej w edytorze zbioru sieci numerem 10* składa się z 4 warstw i jest przedstawiona na rys. 4.

Właściwości sieci typu GRNN

Sieci regresyjne (inaczej określane mianem *bayesowskich*) należą do grupy sieci probabilistycznych, zwane również sieciami realizującymi regresję uogólnioną, lub sieciami typu *GRNN*. Mają one (z definicji) strukturę zawierającą 4 funkcjonalnie zróżnicowane warstwy: wejściową, radialną (przechowującą centra), regresyjną oraz wyjściową. W istocie stanowią one neuronową reprezentację statystycznej metody jądrowej aproksymacji funkcji. Sieci typu *GRNN*, jak sama nazwa wskazuje, służą do neuronowego modelowania problemów regresyjnych [1, 4, 5].

Edytor zbioru danych (Kopia matrix1)

Zmienna: 16 Przypadki: 1300 200 72

| | A41 | A42 | A43 | A44 | B11 | B12 | B13 | B14 |
|------|-----|-----|-----|-----|---------|---------|---------|---------|
| 1559 | 5 | 8 | 9 | 0 | 0.1435 | -0.1603 | -0.0132 | 0.0371 |
| 1560 | 8 | 0 | 7 | 2 | 0.0543 | 0.2947 | -0.2604 | 0.2077 |
| 1561 | 2 | 0 | 8 | 3 | 0.0104 | -0.1129 | 0.1603 | -0.1028 |
| 1562 | 3 | 0 | 3 | 9 | 0.3051 | -0.3559 | 0.0847 | 0.1582 |
| 1563 | 4 | 7 | 3 | 8 | -0.1429 | 0 | 0.1429 | 0 |
| 1564 | 8 | 2 | 6 | 7 | 0.2581 | 0.3548 | -0.9274 | -0.2661 |
| 1565 | 9 | 7 | 7 | 5 | -0.1463 | 0.1799 | 0.1007 | -0.0911 |
| 1566 | 2 | 1 | 7 | 2 | -0.0895 | 0.1096 | 0.1235 | -0.1019 |
| 1567 | 9 | 1 | 8 | 6 | -0.34 | 0.0267 | 0.2667 | 0.16 |
| 1568 | 6 | 9 | 5 | 2 | -0.0729 | 0.0247 | 0.2127 | -0.1022 |
| 1569 | 3 | 4 | 6 | 9 | 0.0588 | 0.0588 | 0.2647 | -0.2353 |
| 1570 | 0 | 7 | 6 | 0 | 0.1376 | -0.0458 | -0.0094 | -0.1269 |
| 1571 | 4 | 8 | 3 | 8 | 0.2357 | 0.4204 | -0.3758 | -0.2038 |
| 1572 | 8 | 3 | 4 | 9 | 0.3884 | -0.239 | -0.1952 | 0.0319 |

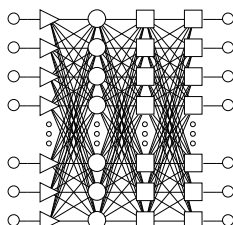
Rys. 2. Edytor zbioru danych do uczenia sieci neuronowej
Fig. 2. Editor of data set for neural network learning

Edytor zbioru sieci (Kopia matrix1)

Bieżąca sieć: 10 Pokaż: Podstawowe Opcje...

| | Typ | Błąd | Wejścia | Ukryte | Jakość |
|-----|--------|-----------|---------|--------|-----------|
| 01 | MLP | 14.23508 | 16 | 20 | 2.036053 |
| 02 | MLP | 14.12878 | 16 | 20 | 1.54935 |
| 03 | MLP | 14.00372 | 16 | 20 | 1.91245 |
| 04 | Linowa | 1.3619 | 16 | - | 0.9288065 |
| 05 | RBF | 0.6650711 | 16 | 86 | 0.6477102 |
| 06 | RBF | 0.649893 | 16 | 91 | 0.6360541 |
| 07 | RBF | 0.6354129 | 16 | 95 | 0.6269212 |
| 08 | GRNN | 0.008985 | 16 | 1300 | 0.00248 |
| 09 | GRNN | 0.008984 | 16 | 1300 | 0.002393 |
| 10* | GRNN | 0.008983 | 16 | 1300 | 0.002423 |

Rys. 3. Edytor zbioru wygenerowanych sieci neuronowych
Fig. 3. Editor of set of generated neural networks



Rys. 4. Topologia sieci GRNN
Fig. 4. Topology of the GRNN network

Warstwa wejściowa wygenerowanej sieci GRNN ma 16 (4×4) neuronów stanowiących niezależne zmienne wejściowe (reprezentatywne cechy zbioru danych) i służą do wprowadzenia informacji do sieci. Druga warstwa wygenerowanej sieci składa się z 1300 neuronów radialnych o licznie adekwatnej do liczby przypadków uczących. W powyższej interpretacji neurony warstwy radialnej reprezentują centra skupień występujące w danych uczących. Kolejna warstwa to warstwa regresyjna. Z definicji musi mieć ona dokładnie o jeden neuron więcej niż warstwa wyjściowa (w omawianym przypadku jest to liczba 17) i stanowią ją neurony liniowe. Warstwa ta zawiera dwa typy neuronów: neurony typu „A”, które obliczają warunkową regresję dla każdej zmiennej wyjściowej oraz pojedynczy neuron typu „B” obliczający gęstość prawdopodobieństwa.

Czwarta warstwa, czyli warstwa wyjściowa, ma szczególne znaczenie w sieci typu GRNN i spełnia specjalną funkcję. Każdy z jej neuronów korzysta z wartości genero-

wanych przez warstwę poprzednią i wyznacza iloraz wartości wyjściowej odpowiadającego mu neuronu klasy „A” i wartości wyjściowej neuronu klasy „B”. Do wyznaczenia ilorazu służy specjalna funkcja *PSP* (*PostSinaptic Potential*) będąca funkcją ilorazową. Warto zauważyć, że neurony wyposażone w takie funkcje *PSP* trudno uznać za wierne modele systemów biologicznych, co stanowiło podstawę koncepcji sztucznej sieci neuronowej jako komputerowego symulatora przetwarzającego sygnały podobnie jak to czyni mózg. Przeciwnie, żaden biologiczny system neuronowy nie wyznacza bezpośrednio funkcji ilorazowej. Jednak praktyczna użyteczność neuronów ilorazowych powoduje, że można użyć ich w opisaney wyżej klasie sieci neuronowych, robiąc tym samym odstępstwo od wierności biologicznym wzorcom na rzecz użyteczności dla praktyki [4].

Neurony radialne w sieciach regresyjnych mogą być uczone przy zastosowaniu dowolnej techniki samouczenia (uczenia bez nauczyciela) zaimplementowanej w strukturze symulatora *Statistica Neural Networks v.4.0.*, w szczególności przy użyciu metod dostępnych dla sieci *RBF* (*Radial Basis Function*) oraz za pomocą algorytmu *Kohonen* [4]. Natomiast warstwa regresyjna i warstwa wyjściowa uczone są w bardzo krótkim czasie za pomocą specjalizowanego algorytmu.

W trakcie eksploatacji sieć regresyjna wykorzystuje *bayesowską* technikę estymacji wartości oczekiwanej zmiennej wyjściowej przy danych wartościach wejściowych. Omawiane postępowanie odbywa się zgodnie z formułą:

$$E[y|x] = \frac{\int_{-\infty}^{+\infty} yf(x, z) dz}{\int_{-\infty}^{+\infty} f(x, z) dz} \quad (7)$$

gdzie:

y jest wartością zmiennej wyjściowej Y ,
 x jest przypadkiem wejściowym (dla którego szacowana jest wartość zmiennej wyjściowej Y),
 f jest łączną funkcją gęstości prawdopodobieństwa wartości wejściowych x i wartości wyjściowej y .

Ponieważ łączna funkcja gęstości prawdopodobieństwa zwykle nie jest znana, więc jest ona szacowana poprzez zsumowanie gaussowskich funkcji jądrowych. Jak podano wyżej, w trzeciej warstwie znajdują się dwa typy neuronów, które obliczają licznik oraz mianownik formuły (7) dla każdej zmiennej wyjściowej. Neurony typu A (ich liczba jest równa liczbie neuronów w warstwie wyjściowej) obliczają licznik a pojedynczy neuron typu B oblicza (wspólny) mianownik. Neurony wchodzące w skład czwartej warstwy realizują po prostu niezbędną operację dzielenia.

Należy podkreślić, że w sieciach typu *GRNN* neurony radialne reprezentują raczej skupienia, a nie pojedyncze przypadki uczące. Algorytm testuje każdy przypadek i przypisuje go do najbliższego neuronu radialnego i wyznacza wagi dla znajdujących się w trzeciej warstwie neuronów typu A i typu B wykonując odpowiednio operację sumowania wartości zmiennych wyjściowych oraz zliczania przypadków przypisanych do danego skupienia.

Omówienie wyników

W celu oszacowania jakości wygenerowanej sieci neuronowej wykorzystano standardowe narzędzia dostępne z poziomu aplikacji *Statistica Neural Networks v. 4.0*. Na rys. 5 przedstawiono statystyki zagadnień regresyjnych.

| | Uc. B21 | Wa. B21 | Te. B21 |
|--------------|-----------|-----------|-----------|
| Odch. std. | 1.22478 | 1.569696 | 1.584156 |
| Średni błęd. | 0.0004738 | 0.0006893 | 0.0008648 |
| Odch. błęd. | 0.005788 | 0.006937 | 0.008164 |
| Śr. bł. bez | 0.0006734 | 0.0009112 | 0.00107 |
| Iloraz odch. | 0.004726 | 0.00442 | 0.005154 |
| Korelacja | 0.9998888 | 0.999902 | 0.9999867 |

Rys. 5. Statystyki regresyjne dla przypadku nr 5
 Fig. 5. Regressive statistics for the event no. 5

Najważniejsze informacje zawierają dwa ostatnie wiersze okna prezentowanego na rys. 5 tj. iloraz odchyień standardowych oraz korelacja. Pierwszy z współczynników stanowi popularny miernik jakości modelu stosowanego do rozwiązywania problemów regresyjnych. W celu ustalenia wartości tego miernika wyznaczany jako iloraz odchylenia standardowego błędów predykcji i odchylenia standardowego zmiennej wyjściowej. Mniejsza (bliższa zeru) wartość miernika wskazuje na lepszą jakość predykcji, jest on bowiem tym mniejszy, im większa jest wariancja wyjaśniona przez model. Natomiast korelacja stanowi standardowy współczynnik korelacji *R Pearsona* (dla wartości zadanej

i wartości uzyskanej na wyjściu). Oczywiście im bliższy jedności jest ten współczynnik tym zależność stochastyczna pomiędzy wartością zadaną a uzyskaną jest silniejsza.

Uruchamiając wygenerowaną sieć dla wybranego losowo przypadku można otrzymać:

| | A11 | A12 | A13 | A14 | A21 |
|---------|-----|-----|-----|-----|-----|
| Wejście | 1 | 8 | 3 | 9 | 3 |

| | B11 | B12 | B13 | B14 | B21 |
|---------|-----------|------------|-----------|------------|-----------|
| Wyjście | -0.1297 | 0.2395927 | -0.3063 | 0.2053953 | -0.0757 |
| Zadane | -0.1297 | 0.2396 | -0.3063 | 0.2054 | -0.0757 |
| Błąd | 6.463e-06 | -7.343e-06 | 5.324e-06 | -4.665e-06 | 8.362e-07 |

Rys. 6. Odpowiedź sieci przy prezentacji przypadku nr 5
 Fig. 6. Network's response at the presentation of event no. 5

Uwagi końcowe

Uzyskane wyniki pozwalają stwierdzić, że właściwie wygenerowane i nauczone topologie sieci neuronowe potrafią realizować procedurę odwracania macierzy, uzyskując bardzo dobre rezultaty. W stosunku do tradycyjnej techniki uzyskiwania macierzy odwrotnej (w oparciu o właściwy algorytm) czas uzyskania wyniku wydaje się być znacząco krótszy. Niestety, odbywa się to kosztem czasu przeznaczonego na uczenie sieci, co determinuje praktyczne wykorzystanie przedstawionej techniki.

Należy zauważyć, że sieci liniowe, które uczone standardową techniką pseudoinwersji, w sposób naturalny są predestynowane do realizacji procedury odwracania macierzy [6], wyraźnie ustępują sieciom typu *RBF* oraz *GRNN*, realizującym nieliniowe przetwarzanie sygnałów.

Podsumowując można stwierdzić, że sieci regresyjne uczą się w stosunkowo krótkim czasie, ale mają tendencję do posiadania dużych rozmiarów. Fakt ten determinuje obszar ich wykorzystania ze względu na oczekiwane wymagania sprzętowe. Należy jednak podkreślić, że w przypadku sieci typu *GRNN* istnieje możliwość istotnej redukcji warstwy radialnej. Jednak w konsekwencji wydłuża się proces generowania sztucznej sieci neuronowej, bowiem wymaga to dodatkowego użycia (w procesie uczenia) algorytmu grupującego, np. *metody k - średnich*.

Literatura

- [1] Bishop, C. (1995). *Neural Networks for Pattern Recognition*. Oxford: University Press
- [2] Osowski S. (2000). *Sieci neuronowe do przetwarzania informacji*: Oficyna Wydawnicza Politechniki Warszawskiej, Warszawa
- [3] Hertz J., Krogh A., Palmer R. G. (1993). *Wstęp do teorii obliczeń neuronowych*: WNT, Warszawa
- [4] Tadeusiewicz R. (1993). *Sieci neuronowe*: Akademicka Oficyna Wydawnicza, Warszawa
- [5] Patterson, D. (1996). *Artificial Neural Networks*. Singapore: Prentice Hall
- [6] Zaleski R. (1987). *Metody algebry liniowej w inżynierii procesowej*: PWN, Warszawa.