Piotr BONIECKI, Jacek PRZYBYŁ
Akademia Rolnicza w Poznaniu
Instytut Inżynierii Rolniczej

THE ARTIFICIAL NEURAL NETWORK AS A HELPING TOOL IN THE PROCESS OF NON-LINEAR DATA COMPRESSION

Summary

An autoassociative network is one which reproduces its inputs as outputs. Autoassociative networks have at least one hidden layer with less units than the input and output layers (which obviously have the same number of layers as each other). Hence, autoassociative networks perform some sort of dimensionality reduction or compression on the cases.
Dimensionality reduction can be used to pre-process the input data to encode information in a smaller number of variables. This approach recognizes that the intrinsic dimensionality of the data may be lower than the number of variables. In other words, the data can be adequately described by a smaller number of variables, if the right transformation can be found.

AUTOASOCJACYJNA SIEĆ NEURONOWA JAKO NARZĘDZIE DO NIELINIOWEJ KOMPRESJI DANYCH

Streszczenie

Sieci autoasocjacyjne to sieci, które odtwarzają wartości wejściowe na swoich wyjściach. Działanie takie zdecydowanie ma sens, ponieważ rozważana sieć autoasocjacyjna posiada w warstwie środkowej (ukrytej) zdecydowanie mniejszą liczbę neuronów niż w warstwie wejściowej czy wyjściowej. Dzięki takiej budowie dane wejściowe muszą przecisnąć się przez swojego rodzaju zwężenie w warstwie ukrytej sieci, kierując się w do wyjścia. Dlatego też, w celu realizacji stawianego jej zadania reprodukcji informacji wejściowej na wyjściu, sieć musi się najpierw nauczyć reprezentacji obszernych danych wejściowych za pomocą mniejszej liczby sygnałów produkowanych przez neurony warstwy ukrytej, a potem musi opanować umiejętność rekonstrukcji pełnych danych wejściowych z tej "skompresowanej" informacji. Oznacza to, że sieć autoasocjacyjna w trakcie uczenia zdobywa umiejętność redukcji wymiaru wejściowych danych.

Wstęp

Sieci autoasocjacyjne, występujące na ogół w postaci perceptronów wielowarstwowych (*MLP- **M**ulti**L**ayer **P**erceptron*), mają na celu odtwarzanie na swoich wyjściach wartości podanych na wejściu. Sens użycia sieci autoasocjacyjnej polega zwykle na tym, że warstwa ukryta liczy mniej neuronów niż warstwa wejściowa (oraz wyjściowa zarazem). Fakt ten powoduje, że w trakcie pracy sieci następuje w jej strukturze redukcja liczby danych zawartych w wektorze wejściowym. Sieci tego typu mogą być wykorzystywane z powodzeniem m.in. do redukcji wymiaru wektora reprezentującego dane wejściowe (Fausett, 1994; Bishop, 1995), co w sposób istotny wspomaga proces tworzenia optymalnej topologii neuronowej. W szczególności, technika ta może stanowić efektywne narzędzie do kompresji różnego rodzaju danych.

Problem redukcji wymiaru

Znaną i popularną metodą redukcji wymiaru jest analiza składowych głównych (*PCA - **P**rincipal **C**omponents **A**nalysis*). W istocie jest to transformacja liniowa, która redukuje liczbę zmiennych do dowolnej zadanej wartości w taki jednak sposób, aby zachować maksymalną wariancję danych wejściowych. Należy podkreślić, że oczekuje się przy tym zachowania w przetworzonych danych tak dużo wartościowych informacji, jak tylko jest to możliwe. Na marginesie warto też zauważyć, że kierunki maksymalnej wariancji nie są koniecznie kierunkami maksymalnej ilości informacji.

Analiza głównych składowych wyznacza transformację liniową polegającą na rotacji danych do nowego układu współrzędnych, utworzonego przez wektory własne macierzy autokorelacji, wyznaczonej dla tych danych. Wartości własne odpowiadające poszczególnym wektorom własnym określają jak wiele „zmienności" występującej w danych reprezentują odpowiednie wektory własne. Macierz autokorelacji *M* dana jest wzorem:

$$M = \sum_n \left(x_n - \overline{x}\right)^T \left(x_n - \overline{x}\right) \qquad (1)$$

gdzie:
x_n - *n*-ty przypadek uczący,
$\left(x_n - \overline{x}\right)^T$ - wektor transponowany.

W pakiecie *Statistica v. 7.0*. wektory własne i wartości własne wyznaczane są poprzez redukcję macierzy do postaci *trójdiagonalnej* za pomocą metody *Householdera* oraz zastosowanie algorytmu *QL*. Jednym z istotnych problemów związanych z analizą głównych składowych (*PCA*) jest jej liniowy charakter. Z tego też powodu nie może ona być użyta do redukcji wymiaru danych wejściowych w dowolnym przypadku. Może jedynie służyć do identyfikacji wyłącznie liniowych transformacji, optymalizujących "kondensację" informacji zawartej w rozważanych zmiennych, opierających się na wyszukiwaniu kierunków maksymalnej wariancji. Alternatywne podejście, wolne od wskazanego wyżej ograniczenia, polega na wykorzystaniu szczególnej topologii autoasocjacyjnej sieci neuronowej, realizującej nieliniową wersję *PCA*.

Sieci autoasocjacyjne

Zgodnie z definicją, neuronowe sieci autoasocjacyjne (będące na ogół topologiami typu *MLP – **M**ulti**L**ayer **P**erceptrons*), są to sieci, które odtwarzają wartości wejściowe na swoich wyjściach. Działanie takie ma sens, ponieważ rozważana sieć autoasocjacyjna posiada w warstwie środkowej (ukrytej) znacząco mniejszą liczbę neuronów niż w warstwie wejściowej (oraz wyjściowej). W trakcie procesu prezentacji sieci neuronowej wektora danych uczących (np. danych empirycznych) przez to "wąskie gardło" muszą przecisnąć się dane przesyłane z wejścia na wyjście. W celu realizacji przez sieć neuronową stawianego jej zadania, polegającego w tym wypadku na reprodukcji informacji wejściowej na wyjściu (zgodnie z definicją sieci autoasocjacyjnej), sieć musi się najpierw nauczyć reprezentacji danych wejściowych za pomocą mniejszej liczby sygnałów generowanych przez neurony warstwy ukrytej. Dopiero podczas następnego etapu sieć neuronowa może opanować umiejętność rekonstrukcji pełnych danych wejściowych z uprzednio "skompresowanej" informacji, zakodowanej wewnątrz (w warstwie ukrytej) sieci. Oznacza to, że neuronowa sieć autoasocjacyjna w trakcie procesu uczenia zdobywa umiejętność redukcji wymiaru danych wejściowych, spakowanych do warstwy ukrytej (środkowej).

Ogólnie rzecz biorąc, sieć autoasocjacyjna przeznaczona do kompresji danych winna składać się z minimum trzech warstw:

- warstwy wejściowej (o liczbie neuronów odpowiadającej wejściowej liczbie danych),
- warstwy wyjściowej (takiej samej co do wielkości),
- warstwy ukrytej (o znacznie mniejszej liczbie neuronów).

Proponowana sieć neuronowa powinna być uczona w taki sposób, aby wiernie odtwarzała dane wejściowe na swoich wyjściach. Dlatego właśnie posiada ona dokładnie taką samą liczbę wejść co wyjść, a zmiennym używanym do jej uczenia nadaje się szczególny charakter tzw. zmiennych wejściowo/wyjściowych. Jak wspomniano, idea działania rozważanej sieci polega na tym, że liczba neuronów ukrytych jest znacznie mniejsza niż liczba wejść czy wyjść, co w rzeczywistości wymusza “przeciśnięcie" informacji przez reprezentację o mniejszym (skompresowanym) wymiarze.

Tak skonstruowana sieć autoasocjacyjna (trójwarstwowa) realizuje transformację danych wejściowych do warstwy ukrytej (o zredukowanym wymiarze), a następnie wykonuje kolejną transformację powrotną do warstwy wyjściowej. Można udowodnić, że w przypadku kiedy używane neurony w warstwie ukrytej i w warstwie wyjściowej mają charakterystyki liniowe (lub qasiliniowe), to taka sieć w rzeczywistości uczy się aproksymować standardowy algorytm analizy głównych składowych. W omawianym przypadku jest więc ona substytutem, przedstawionej wyżej, liniowej metody *PCA* (Bourland and Kamp, 1988).

Pomysł zrealizowania nieliniowej redukcji wymiaru polega w istocie na zastosowaniu analogicznej topologii neuronowej, jednak zbudowanej z neuronów nieliniowych, tzn posiadających w swojej strukturze nieliniowy potencjał membranowy *PSP- **P**ost-**S**ynaptic **P**otential* (np. neuronów radialnych). Jednak aby w pełni wykorzystać możliwości, jakie stwarza sieć nieliniowa potrzebna jest więcej niż jedna warstwa neuronów dla każdej z **2** realizowanych transformacji (zarówno dla kompresji jak i dla dekompresji). Dlatego tworząc sieć do nieliniowej kompresji danych należy wygenerować topologię sieci neuronowej o **5** warstwach (Kramer, 1991). Ukryta warstwa środkowa jest warstwą redukującą wymiar sygnału wejściowego, zaś warstwa znajdująca się pomiędzy nią i warstwą wejściową dokonują właśnie wymaganej nieliniowej kompresji wejściowych danych. Odpowiednio dwuwarstwowa struktura sieci, znajdująca się pomiędzy warstwą ukrytą a warstwą wyjściową, realizuje transformację odwrotną dekompresując, „zagęszczony uprzednio”, sygnał.

Metodyka badawcza

Jednym z obszarów naukowo-badawczych w dyscyplinie inżynieria rolnicza, w których redukcja wymiaru wektora danych wydaje się być szczególnie przydatna, jest szeroko rozumiana analiza obrazu, wykonywana z pomocą sztucznych sieci neuronowych. Coraz częściej stosowaną w praktyce metodą jest wykorzystanie modeli neuronowych w procesie identyfikacji wybranych obiektów na podstawie odpowiednio przetworzonych kolorowych obrazów, prezentowanych właściwej strukturze neuronowej w postaci plików graficznych, na ogół typu *.bmp, .jpg* lub *.gif*.

W trakcie generowania topologii sieci neuronowych dedykowanych do rozpoznawania obiektów, prezentowanych w postaci dwuwymiarowych obrazów (np. zdjęć fotograficznych), w trakcie procesu digitalizacji pojawiają się zbiory uczące o bardzo dużej liczbie zmiennych. Proces ten w sposób istotny (przyrost wielomianowy) pogłębia się w miarę wzrostu rozdzielczości narzędzia dygitalizującego prezentowane obrazy. Np. wykorzystując model koloru *RGB* oraz stosując rozdzielczość 32*32 otrzymujemy wektor uczący zawierający 3072 zmienne. Ta stosunkowo niewielka rozdzielczość generuje przypadek uczący, który znajdując się w strukturze zbioru uczącego, stwarza już poważny problem zarówno dla symulatora sieci neuronowej, jak również dla komputera realizującego przetwarzanie numeryczne. Wytworzenie reprezentatywnego zbioru uczącego o zredukowanej liczbie zmiennych wydaje się w tym przypadku szczególnie wazne.

W celu wygenerowania sieci autoasocjacyjnej posłużono się modułem popularnego pakietu *Statistica* v. 7.0. stanowiącym efektywny symulator szerokiej rodziny jednokierunkowych sieci neuronowych. Standardowa procedura zaimplementowana w tym pakiecie składała się z następujących etapów:

etap 1: przygotowanie zbioru danych do uczenia sieci autoasocjacyjnej. W tym celu najpierw wszystkim zmiennym wyjściowym występującym w pierwotnym problemie nadaje się charakter zmiennych „pominiętych” (na etapie poszukiwania metod redukcji wejściowego zbioru danych sygnały wyjściowe nie mają żadnego znaczenia). Następnie wszystkim zmiennym wejściowym nadaje się charakter danych „wejściowo/wyjściowych”,

etap 2: utworzenie pięciowarstwowej, autoasocjacyjnej sieci typu *MLP* (warstwa środkowa musi posiadać znacząco mniejszą liczbę neuronów niż warstwa wejściowa czy wyjściowa). Dwie pozostałe warstwy ukryte mogą posiadać relatywnie dużą liczbę neuronów (obie powinny posiadać taką samą ich liczbę),

etap 3: przeprowadzenie procesu uczenia sieci autoasocjacyjnej w oparciu o zbiór uczący za pomocą dowolnego algorytmu iteracyjnego (na przykład algorytmu

wstecznej propagacji błędu czy gradientów sprzężonych),

etap 4: usunięcie 2 ostatnich warstw (cięcie - rys. 4) w wytworzonej sieci autoasocjacyjnej (za pomocą przycisku „usuń”, znajdującego się w „edytorze sieci”). W ten sposób zostaje wygenerowana sieć zawierająca wyłącznie strukturę przetwarzającą liczne dane wejściowe na stosunkowo nieliczne dane w środkowej warstwie (dawniej ukrytej, a obecnie wyjściowej). Ta "okaleczona" sieć będzie teraz dokonywała przetwarzania danych z warstwy wejściowej do warstwy wyjściowej (dawniej ukrytej) w celu realizacji nieliniowej redukcji wymiaru,

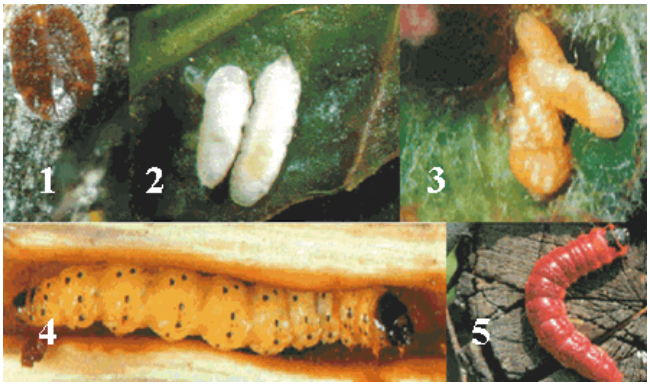
etap 5: zastosowanie wygenerowanej sieci neuronowej do utworzenia wersji danych wejściowych o zredukowanym (skompresowanym) wymiarze. W ten sposób uzyskuje się nowy zbiór uczący o zredukowanym wymiarze wejściowego wektora danych,

etap 6: utworzenie drugiej sieci (sieci rozwiązującą zasadniczy problem) i przeprowadzenie procesu jej uczenia (korzystając ze zbioru o zredukowanym wymiarze).

W celu wykorzystania sieci autoasocjacyjnej do kompresji danych należy wykonać etapy od 1 do 6. Następnym krokiem jest wykonywane dwóch kopii wygenerowanych sieci, z których za pomocą (zaimplementowanej w pakiecie *Statistica v. 7.0.*) techniki usuwania warstw, tworzone są 2 sieci: kompresująca oraz dekompresująca. Zadaniem sieci kompresującej jest przetwarzanie oryginalnych danych do postaci skompresowanej, natomiast sieć dekompresująca przywraca dane oryginalne.

Omówienie wyników

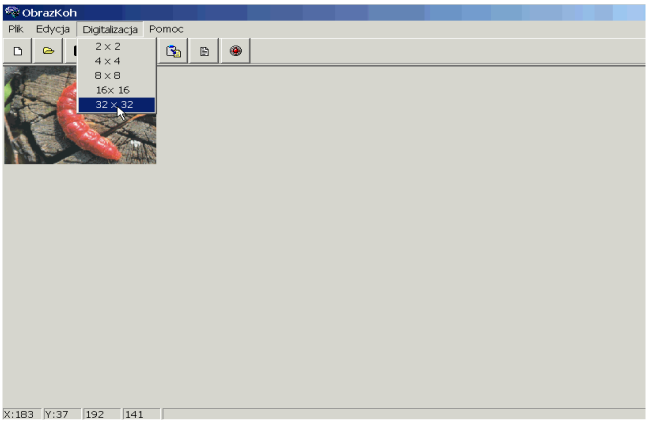
Technikę nieliniowej kompresji wektora danych za pomocą sieci neuronowych wykorzystano do redukcji wymiaru wektora uczącego sieć neuronową służącą do identyfikacji wybranych szkodników sadów.



Rys. 1. Wybrane szkodniki sadów
Fig. 1. Chosen pests of orchards

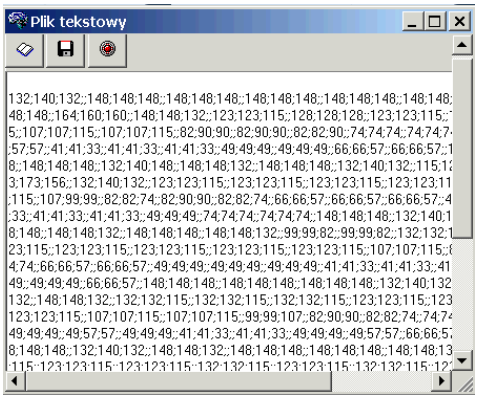
Rozpoznawaniu poddano 5 następujących gatunków:
1 - misecznik śliwowy (*Parthenolecanium Corni Bouche*),
2 - pryszczarek gruszowiec (*Dasyneura piri Bonche*),
3 - pryszczarek jabłoniak (*Dasyneura mali Kieff*),
4 - trociniarka torzyśniad (*Zeuzera pyrina L.*),
5 - trociniarka czerwica (*Cossus cossus L.*).

Za pomocą wytworzonej aplikacji *ObrazKoch* (Boniecki P., Piekarska-Boniecka H., 2004) dokonano digitalizacji zdjęć szkodników przekształcając je do postaci adekwatnej dla procesu uczenia sieci neuronowej, przyjmując rozdzielczość 32*32.



Rys. 2. Aplikacja *ObrazKoch* do digitalizacji obrazów
Fig. 2. The application ObrazKoch to picture digitalisation

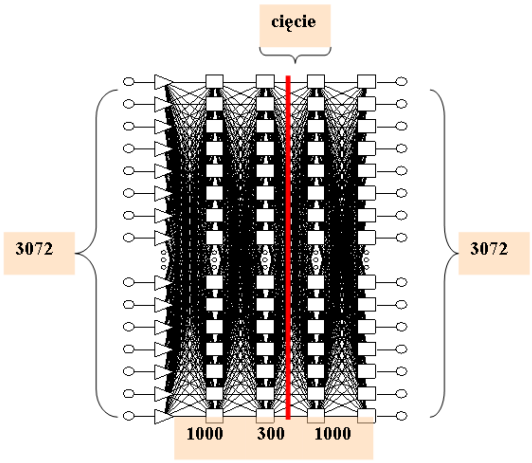
W efekcie uzyskano plik typu .csv, który jest akceptowany przez symulator sieci neuronowych, zaimplementowany w pakiecie *Statistica v. 7.0*. Przy założonej rozdzielczości (32*32) jeden wektor uczący składał się z 3072 wyrazów.



Rys. 3. Fragment wygenerowanego przez *ObrazKoch* pliku typu .csv
Fig. 3. The fragment generated through ObrazKoch of file of type .csv

Po zaimportowaniu wygenerowanych plików do edytora danych symulatora sieci neuronowych pakietu *Statistica v. 7.0.*, posługując się standardowa procedurą, wytworzono autoasocjacyjną sieć neuronowa typu *MLP* składającą się z 5-ciu warstw i posiadającą następującą strukturę:
3072 – 1000 – 300 – 1000 – 3072.

Po nauczeniu pięciowarstwowej sieci neuronowej nastąpiło usuniecie 2 ostatnich warstw wytworzonej sieci autoasocjacyjnej (cięcie-rys.4.). W ten sposób została wygenerowana sieć przetwarzająca liczne (3072) dane wejściowe na stosunkowo nieliczne (300) dane, „skompresowane” w środkowej warstwie (dawniej ukrytej a obecnie wyjściowej). Tak zbudowana sieć może dokonać przetwarzania danych z warstwy wejściowej do warstwy wyjściowej, realizując tym samym zadaną, nieliniową redukcję wymiaru.



Rys. 4. Autoasocjacyjna sieć neuronowa typu *MLP*
Fig. 4. Autoassociative neural network type MLP

W celu sprawdzenia jakości tak skompresowanych danych wykorzystano opcje automatycznego projektanta do wygenerowania zbioru topologii *SNN* mających za zadanie identyfikację wymienionych wyżej szkodników sadów. W tym celu do uczenia sieci przyjęto wytworzony, zredukowany do 300 zmiennych wejściowych, zbiór danych uczących. Najlepsze parametry wykazała sieć neuronowa typu *RBF* (***R**adial **B**asic **F**unction*) o strukturze postaci: **300 – 212 - 5,** uczona trzyetapowo metodą *k-średnich*, *k-sąsiadów* oraz (dla warstwy wyjściowej) metodą *pseudoinwersji*. Wygenerowana w oparciu zredukowane dane sieć typu *RBF*, efektywnie klasyfikowała reprezentacje wybranych owadów, co pozwalało na wykorzystanie jej do poprawnej identyfikacji wybranych szkodników sadów.

Uwagi końcowe

Właściwe przygotowanie zbioru reprezentatywnych danych uczących jest zadaniem fundamentalnym, zarówno w procesie edukacji sieci neuronowych jak również ich eksploatacji. Omówiona wyżej procedura przygotowania zbioru danych do edukacji sieci można traktować jako element *preprocessingu* (wstępnego przygotowania danych do postaci akceptowalnej przez sieć). Wykorzystana technika nieliniowej kompresji danych za pomocą autoasocjacyjnej sieci neuronowej o strukturze perceptronu wielowarstwowego to efektywna metoda, pozwalająca na istotną redukcję wymiaru wektora uczącego bez straty informacji w nim zawartej. Dzięki zastosowanemu opisanego sposobu przetwarzania danych empirycznych było możliwe skuteczne wygenerowanie topologii sieci neuronowej do efektywnej identyfikacji wybranych szkodników sadów.

Literatura

[1] Bishop C., (1995). Neural Networks for Pattern Recognition. Oxford: University Press.
[2] Fausett L., (1994). Fundamentals of Neural Networks. New York: Prentice Hall.
[3] Bouland H., Kamp Y., (1988). Auto-association by multilayer perceptrons and singular value decomposition. Biological Cybernetics 59, 291-294.
[4] Kramer M.A. (1991). Nonlinear principal components analysis using autoassociative neural networks. AIChe Journal 37 (2), 233-243.
[5] Osowski S. (2000). Sieci neuronowe do przetwarzania informacji: Oficyna Wydawnicza Politechniki Warszawskiej, Warszawa
[6] Stateczny A., Praczyk T. (2002). Sztuczne sieci neuronowe w rozpoznawaniu obiektów morskich: GTN, Gdynia
[7] Tadeusiewicz R. (1993). Sieci neuronowe: Akademicka Oficyna Wydawnicza, Warszawa
[8] Boniecki P., Piekarska-Boniecka H.(2004). Neuronowa identyfikacja wybranych szkodników drzew owocowych w oparciu o analizę obrazu: Journal of Research and Applications in Agricultural Engineering, Vol. 49(3), str.25-30.