

Marcin Kolibabka^{*}, Andrzej Cader^{*}

Intuicyjne metody wstępnego kształtowania wag w jednokierunkowych sieciach perceptronowych

1. Wstęp

Jednokierunkowe, wielowarstwowe nieliniowe sieci neuronowe zwane MLP (*Multi Layered Perceptron*), można też spotkać polską nazwę – wielowarstwowe sieci perceptronowe [1], dzięki prostej implementacji są najczęściej stosowanymi rozwiązaniami z zakresu struktur neuronowych. Zasady przedstawione pod koniec lat 80. XX w., opisujące możliwości sieci neuronowych (każda ograniczona funkcja ciągła może być aproksymowana z dowolnie małym błędem przez sieć z jedną warstwą ukrytą [2, 5] ponadto dowolna funkcja może być aproksymowana z dowolną dokładnością przez sieć z dwoma warstwami ukrytymi [2, 4]), oraz opracowanie algorytmu wstecznej propagacji błędu EBP (*Error Back Propagation*) [4] bezpośrednio przyczyniły się do ich rozpowszechnienia, po wcześniejszym wieloletnim porzuceniu badań nad nimi. Do wielu zastosowań predysponuje je również stosunkowo prosta struktura nauczanej sieci, która w połączeniu z odpowiednim zrównolegleniem przetwarzania sygnałów pozwala na szybkie uzyskiwanie reakcji sieci na zmianę parametrów wejściowych. Jednak, aby ta zaleta była w pełni wykorzystana, konieczne jest wcześniejsze nauczanie sieci.

Uczenie sieci podlega wielu czynnikom począwszy od stawianego przed nią zadania, poprzez zbiór danych uczących, strukturę sieci aż po sposób uczenia i parametry przyjęte przy jego rozpoczęciu. Zmiana każdego z tych czynników może znacząco wpłynąć na przebieg i wynik uczenia. Z drugiej strony do efektywnego rozwiązania można dojść na wiele sposobów. Dokonujemy wyboru nie tylko algorytmu trenowania sieci, ale również inicjowania wag oraz precyzowania pewnych fragmentów algorytmu (np. wyłączenie spod uczenia niektórych neuronów lub całych warstw).

W niniejszej pracy zaproponowano metodę wstępnego doboru wag dla metody z rodziny wstecznej propagacji błędu. Metoda pozwala w sposób bardziej efektywny realizować proces uczenia sieci.

^{*} Wyższa Szkoła Humanistyczno-Ekonomiczna, Łódź

2. Momentowa metoda wstecznej propagacji błędu

Dzięki swojej efektywności w połączeniu z względnie prostą strukturą algorytmu metody z grupy wstecznej propagacji błędu cieszą się dużą popularnością w wielu spektrach zastosowań. Podczas uczenia sieci dąży się do minimalizacji nieliniowej funkcji celu. W wypadku metody wstecznej propagacji błędu dąży się do minimalizacji błędu E będącego sumą błędów E_i obliczonych dla każdego wektora danych uczących. Oblicza się go ze wzoru dla i -tego wektora danych

$$E_i = \frac{1}{2} \sum_{k=1}^m (out_{ki} - wzor_{ki})^2 \quad (1)$$

gdzie out_{ki} – wartość na k -tym wyjściu sieci, a $wzor_{ki}$ – wartość oczekiwana na tym wyjściu.

Po przetworzeniu przez sieć n -tego wektora ze zbioru uczącego następuje modyfikacja neuronów w sieci w oparciu o metodę największego spadku. Oznacza to, że wagi neuronów są modyfikowane na podstawie wzoru

$$w^{(n+1)} = w^{(n)} - \eta \nabla E_i \left(w^{(n)} \right) \quad (2)$$

Oznaczenia we wzorze to:

- $w^{(n+1)}$ – wektor wag w $n+1$ kroku algorytmu,
- $w^{(n)}$ – wektor wag w n -tym kroku algorytmu,
- η – współczynnik prędkości uczenia,
- ∇E_i – gradient funkcji E_i w punkcie $w^{(n)}$.

Modyfikacja tej metody poprzez wprowadzenie parametru zwanego momentum (pędem), pozwala na uczynienie jej bardziej równomierną i mniej wrażliwą na lokalne zmiany przebiegu błędu. Osiąga się to poprzez wprowadzenie do każdej zmiany czynnika powiązanego ze zmianą z poprzedniego kroku. Opisując to wzorem, otrzymujemy

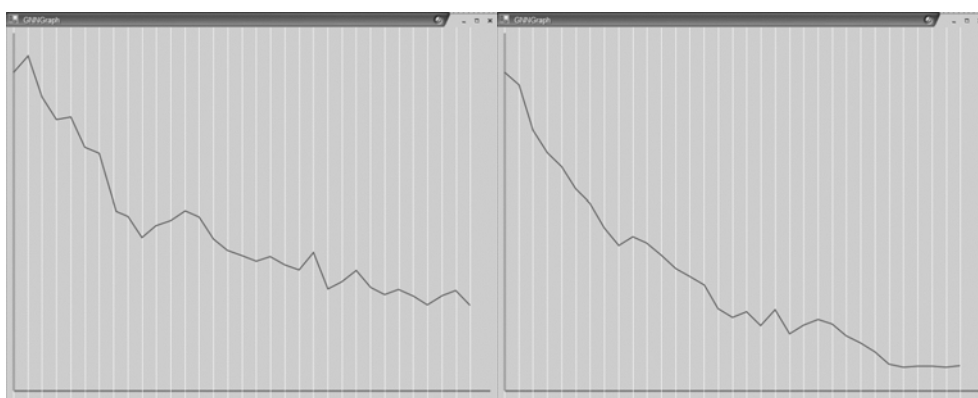
$$\Delta w^n = -\eta \nabla E_i \left(w^n \right) + \mu \Delta w^{n-1} \quad (3)$$

gdzie μ to współczynnik pędu (momentum).

Tak zmodyfikowaną metodę nazywa się momentową metodą wstecznej propagacji błędu MEBP (*Momentum Error Back Propagation*).

W obu tych metodach, z tak przyjętych założeń wynika, że w kolejnym $n+1$ kroku nie następuje wyliczenie nowych wag, a jedynie istniejące wagi są modyfikowane o wyliczoną wartość. W wyniku tego należy już w pierwszym kroku nadać wagom określone wartości. Najczęstszym sposobem jest losowe przydzielenie wartości z określonego przedziału, np. $(-1, 1)$. Dla efektywności uczenia takie losowe przydzielanie wag jest korzystniejsze od nadania wszystkim wagom tej samej wartości początkowej, ale wcale nie musi być rozwiązaniem najkorzystniejszym. Najprostszym dowodem jest porównanie wykresów średniego błędu sieci od kroku procesu uczenia (rys. 1). Błąd średni jest liczony z błędów E_i

(równanie 1), a liczba kroków, co ile jest obliczany, jest dobierana indywidualnie dla danego uczenia (w przedstawionych wykresach błąd średni jest liczony z 300 kolejnych wyników iteracji procesu uczenia). Oba wykresy (rys. 1) przedstawiają przebieg błędu w procesie uczenia dla problemu irysów, gdzie wartości wag zostały wybrane losowo, a samo uczenie przebiegało po przy tych samych parametrach i zbiorze danych uczących. Wykresy pokazują różnicę w prędkości i dokładności procesu treningu sieci dla różnych losowych zbiorów wag wejściowych. W skrajnym przypadku wylosowane wagi mogą uniemożliwić zakończenie treningu sukcesem. Idealnym rozwiązaniem byłoby wylosowanie od razu dobrych wag lub przynajmniej wag zbliżonych do rozwiązania.



Rys 1. Przebieg błędu w procesie uczenia dla dwóch zestawów wag początkowych wybranych losowo. Problem klasyfikacji irysów, 10 000 iteracji. Na osi pionowej zaznaczona jest wartość średniego błędu liczonego co 300 kroków uczenia (przedziały liczenia oznaczone białymi pionowymi liniami). Błąd ten jest liczony z błędów ze wzoru (1). Na osi poziomej oznaczone są kolejne kroki iteracji

3. Graficzny interfejs regulacji wag

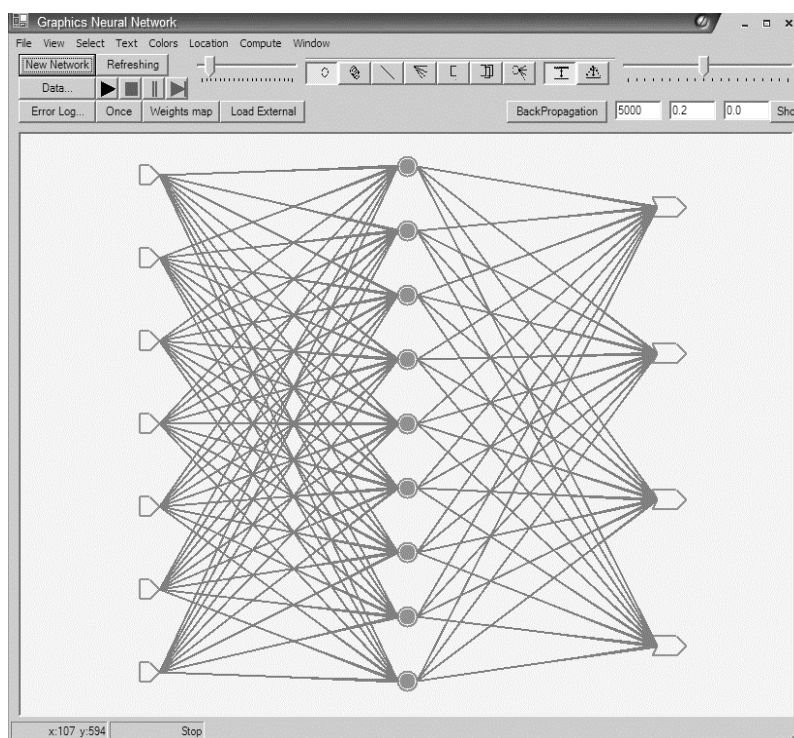
W praktycznych zastosowaniach sieci neuronowych okazuje się, że duże znaczenie ma samo doświadczenie i intuicja twórcy sieci neuronowej. W losowym dobieraniu wag początkowych, udział twórcy sieci był ograniczony do wyboru przedziału, z jakiego wartości były losowane. Nawet taka kontrola dawała poprawienie skuteczności uczenia. Celem proponowanego interfejsu jest uproszczenie procesu przygotowania wag dla treningu metodą wstecznej propagacji błędu oraz znalezienie zasad, które pomagają w takich przygotowaniach.

Pierwszym napotkanym problemem okazał się sposób wizualizacji zależności pomiędzy wartościami wag a ich reprezentacją. Wagi reprezentowane są poprzez połączenia pomiędzy głównymi elementami neuronów, czyli sumatorami i funkcjami aktywacji.

W mniej intuicyjnym podejściu, ale za to dającym większe możliwości konfiguracji, wartość wagi jest oznaczona odcieniem barwy połączenia. Wraz ze zwiększaniem wartości wagi, które odbywa się poprzez przeciąganie wskazanego elementu lub elementów interfejsu, barwa zwiększa swoją jasność. W skrajnym przypadku, gdy waga osiąga wartość ujem-

ną reprezentacją barwną wagi zmienia się na negatyw koloru reprezentującego wagę. Zaletą tej reprezentacji jest to, że zachowana zostaje swoboda niezależności zmian wag, i to przy manipulacji zarówno indywidualnymi wagami, jak i całymi ich grupami. Jednak obsługa interfejsu za pomocą reprezentacji kolorystycznej wymaga przyzwyczajenia operatora.

Bardziej naturalnym podejściem jest reprezentacja wartości wagi za pomocą długości połączenia. W tym wypadku zmiana długości połączenia zmienia proporcjonalnie wartość wagi. Aby wprowadzić większy zakres zmian wartości, możliwe jest ustawienie „czułości”, czyli współczynnika proporcjonalności zmian, za pomocą suwaka w interfejsie programu (rys. 2).



Rys 2. Wygląd proponowanego interfejsu programu

Możliwe jest manipulowanie wartościami wag poprzez kilka różnych reprezentacji elementów sieci. Oczywiście podstawową reprezentacją jest pojedyncza waga, jednak posługiwanie tylko tą możliwością niewiele upraszcza wstępne kształtowanie wag. Sposób ten jest szczególnie przydatny w końcowym etapie procesu, dając możliwość szczegółowego nadania wartości wadze. Dlatego w interfejsie zastosowano inne sposoby regulacji wag. Zaznaczenie wielu wag pozwala zmieniać równocześnie ich wartości. W praktyce okazało się, że najbardziej przydatnym sposobem jest zaznaczanie grup wag i ich łączna zmiana. Powodowało to zmianę wartości wszystkich wag połączonych z zaznaczonymi elementami,

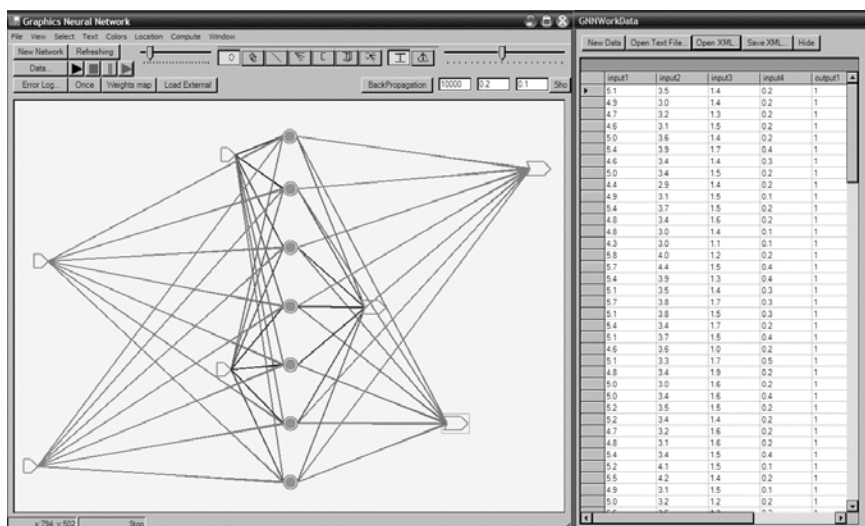
gdyż przemieszczenie wybranego neuronu lub ich grupy zmieniało jego odległość od sąsiednich neuronów, co wpływało na zmianę długości połączeń. Ponieważ odwzorowanie wartości wag jest powiązane z długością ich reprezentacji graficznej, więc w wyniku przesuwania grupy elementów w jej wnętrzu nie zachodzą zmiany wag.

4. Wstępne ustawianie wag

Regulacja wartości wag, za pomocą proponowanego interfejsu, przed przystąpieniem do uczenia sieci neuronowej pozwala na przyspieszenie przebiegu procesu uczenia, jednak wymaga doświadczenia operatora lub posłużenia się intuicją w podejściu do rozwiązania problemu.

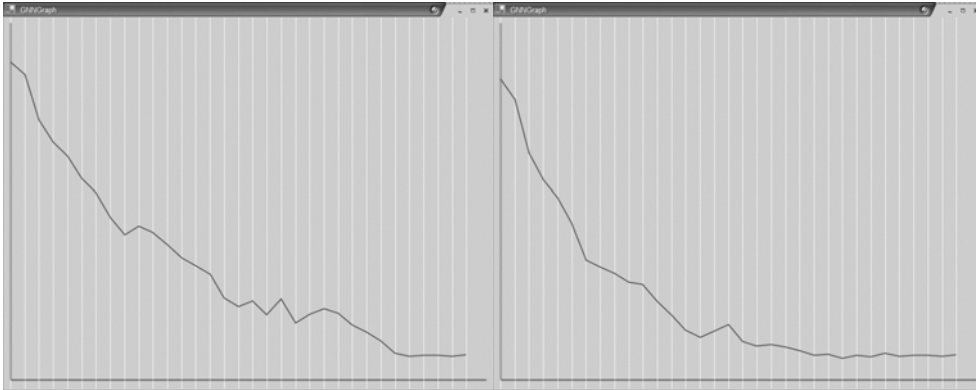
W dalszym ciągu zaprezentowane zostaną dwa odmienne podejścia do zasad ustalania wag początkowych przy uczeniu sieci metodą momentowej wstecznej propagacji błędów. Przykłady będą dotyczyły problemów klasyfikacji irysów i klasyfikacji ryżu jako problemów dobrze opisanych i dających możliwość porównania z innymi opracowaniami.

W zagadnieniu klasyfikacji irysów uzyskany zysk prędkości uczenia osiągnięty został poprzez operacje zależne od zakresu przedziałów danych wprowadzanych na wejścia sieci oraz długości przedziałów klasyfikacji dla wyjść. Wartości ze zbioru uczącego odpowiadające pierwszemu i trzeciemu wejściu miały maksymalną wartość 2-krotnie większą od analogicznej wartości dla drugiego wejścia i 4-krotnie większą od czwartego. Wagi określające propagację sygnału od tych neuronów wejściowych do neuronów warstwy ukrytej znacząco zmniejszono, podczas gdy pozostałe zwiększono. W reprezentacji graficznej znajdowało to swoje odzwierciedlenie w przemieszczaniu neuronów (rys. 3), co przełożyło się na zmianę wartości związanych z nimi wag.

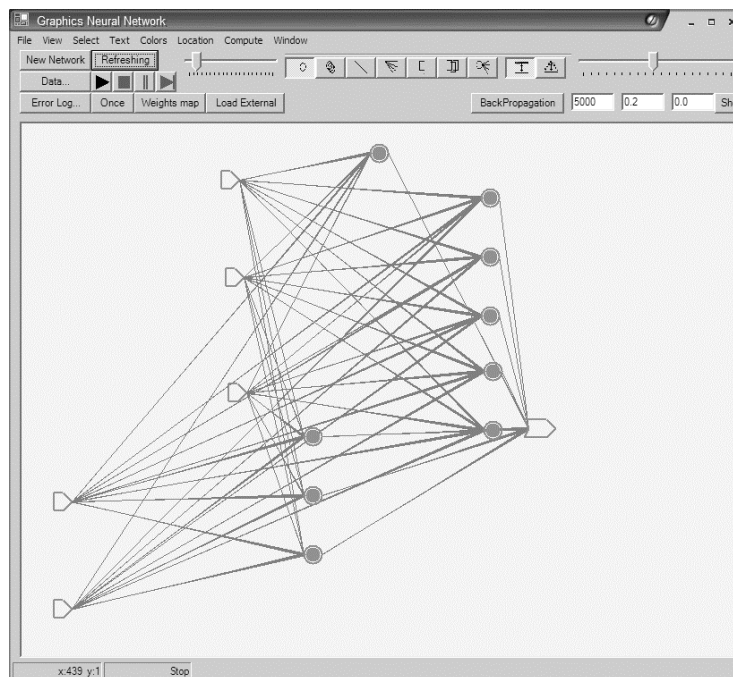


Rys. 3. Wygląd reprezentacji graficznej odzwierciedlający wstępne ustawienie wag w problemie klasyfikacji irysów

Zysk w procesie uczenia został przedstawiony na rysunku 4.



Rys. 4. Po lewej najlepszy wykres z 10 losowań wag początkowych, po prawej wykres z uczenia po intuicyjnym ustawieniu wag. Liczba iteracji w procesie treningu sieci to 10 000. Na osi pionowej zaznaczona jest wartość średniego błędu liczonego co 300 kroków uczenia (przedziały liczenia oznaczone białymi pionowymi liniami). Błąd ten jest liczony z błędów z wzoru (1). Na osi poziomej oznaczone są kolejne kroki iteracji

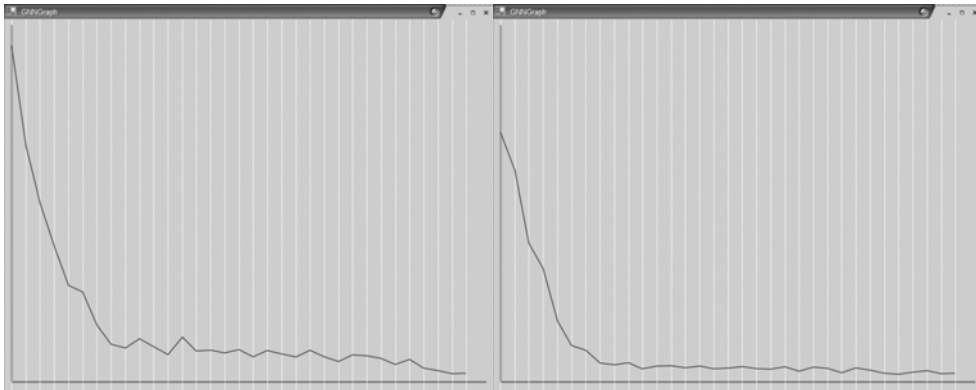


Rys. 5. Wygląd sieci po wstępnym ustawieniu wag dla zagadnienia klasyfikacji ryżu

Jak widać z przedstawionych rysunków, stabilny błąd uzyskany został 3000 iteracji wcześniej. Proces uczenia w obu wypadkach przebiegał przy tych samych parametrach: prędkość uczenia równa 0,2, a pęd uczenia 0,3. Błąd średni kwadratowy na wykresie liczony jest co 300 iteracji.

Całkiem odmienne podejście zostało wykorzystane przy rozwiązywaniu problemu oceny ryżu. W rozwiązaniu tego zagadnienia wykorzystuje się sieć o jednym wyjściu. Ułożenie wejść sieci jest w tym wypadku powiązane nie z wartościami wektora wejściowego, ale z semantycznym znaczeniem wejść. Wejścia związane z wyglądem ryżu otrzymały wagi o mniejszej wartości, a pozostałe wejścia otrzymały zwiększone wartości wag. Proponowane modyfikacje wag początkowych przedstawia interfejs graficzny na rysunku 5. Okazało się, iż zmniejszenie wag na wyjściu w znaczący sposób zmniejszyło błąd wyjściowy już przed uczeniem sieci.

Zysk ze wstępnego kształtowania wag powoduje uzyskanie stabilnego rozwiązania już po około 3000 iteracji. Natomiast początkowy błąd był o jedną trzecią mniejszy niż przy najlepszym z losowych doborów wag (rys. 6).



Rys. 6. Zagadnienie oceny ryżu. Prędkość uczenia 0,25, pęd 0,5, liczba iteracji 10 000. Po lewej najlepszy wykres z 10 losowań wag początkowych, po prawej wykres z uczenia po intuicyjnym ustawieniu wag. Na osi pionowej zaznaczona jest wartość średniego błędu liczonego co 300 kroków uczenia (przedziały liczenia oznaczone pionowymi liniami). Błąd ten jest liczony z błędów ze wzoru (1).

Na osi poziomej oznaczone są kolejne kroki iteracji

5. Wnioski

Intuicyjne, jak również wynikające z nabytej praktyki zasady inicjowania wartości wag mogą przynieść znaczącą poprawę prędkości uczenia, przy oczywistym świadomym udziale osoby nadzorującej. Nie zawsze udaje się uzyskać przyspieszenie prędkości uczenia, jednak w większości przypadków metoda jest skuteczniejsza niż przy całkowicie losowym doborze wag.

Innym zastosowaniem proponowanego interfejsu może być zmiana wag w sieci już częściowo nauczonej, gdy proces uczenia nie przynosi dalszego efektu, np. z powodu utknięcia w minimum lokalnym.

Literatura

- [1] Tadeusiewicz R.: *Sieci neuronowe*. Warszawa, Akademicka Oficyna Wydaw. RM 1993
- [2] Cybenko G.: *Approximation by Superpositions of a Sigmoidal Function*. Mathematics of Control, Signals, and Systems, vol. 2, 1989, 303–314
- [3] Rutkowska D., Piliński M., Rutkowski L.: *Sieci neuronowe, algorytmy genetyczne i systemy rozmyte*. Warszawa, Wydawnictwo Naukowe PWN 1997
- [4] Rumelhart D., Hinton G., Williams R.: *Learning Internal Representations by Error Propagation*. Parallel Distributed Processing, vol. 1, 1986, 318–362
- [5] Hornik K., Stinchcombe M., White H.: *Multilayer feedforward networks are universal approximators*. Neural Networks, 2, 1989, 359–366