

Stanisław PŁACZEK*

PORÓWNANIE CHARAKTERYSTYK WYDAJNOŚCIOWYCH ALGORYTMU UCZENIA DLA STANDARDOWEJ SIECI NEURONOWEJ ORAZ SIECI ZDEKOMPOWANEJ I KOORDYNOWANEJ METODĄ PROGNOZOWANIA WARTOŚCI INTERFEJSÓW

Standardową Sieć Neuronową zdefiniujemy jako zintegrowany moduł składający się z wielu warstw, bez wewnętrznych sprzężeń zwrotnych i pełnym pokryciem wag w poszczególnych warstwach. Warstwa składa się z macierzy wag łączących wektor wejściowy X z wewnętrznym wektorem U , który z kolei przetwarzany jest przez funkcję aktywacji tworząc wektor wyjściowy Y . Algorytmy uczenia tak zdefiniowanej sieci są znane. Napotykają one jednak na określone problemy numeryczne związane z szybkością zbieżności do wartości minimum. W artykule proponuje się zastąpienie jedno poziomowej, wielowarstwowej sieci, siecią zdekomponowaną. Konfiguracja składa się z niezależnych modułów zawierających warstwy sieci pierwotnej oraz koordynatora. Koordynator koncepcyjnie znajduje się na drugim poziomie. Jego celem jest skoordynowanie lokalnych funkcji celu warstw w taki sposób, aby realizując swoje lokalne funkcje celu, zostało jednocześnie osiągnięte minimum globalnej funkcji celu. Sieć jako całość będzie przygotowana do realizacji zadań klasyfikacji. Porównuje się charakterystyki uczenia dwóch przedstawionych konfiguracji sieci.

SŁOWA KLUCZOWE: sieć dwupoziomowa, dekompozycja, koordynacja, algorytm uczenia.

1. STRUKTURY SIECI STANDARDOWEJ I ZDEKOMPOWANEJ

1.1. Sieć standardowa wielowarstwowa

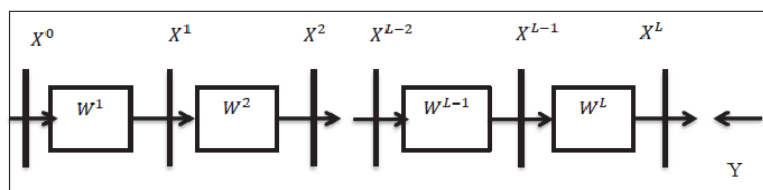
W zadaniach praktycznych, najpopularniejsza konfiguracja sieci neuronowej składa się z warstwy wejściowej, jednej warstwy ukrytej oraz wyjściowej. W tego typu sieciach, struktura warstwy ukrytej zawiera określoną ilość neuronów w warstwie oraz odpowiednią funkcję aktywacji. Parametry te decydują o skuteczności algorytmu uczenia. W celu poprawnej ekstrakcji elementów cech w wektorach wejściowych, sieć musi posiadać stosunkowo dużą ilość neuronów

* Wyższa Szkoła Handlowa Wrocław

w warstwie, w stosunku do wymiarów wektora wejściowego i wyjściowego. Często w literaturze spotyka się określenie sieci szerokiej w odróżnieniu od sieci głębokiej. Pojęcie sieci głębokiej pojawiło się w literaturze stosunkowo niedawno wraz z Głębokim Uczeniem, co związane jest z dwoma bardzo ważnymi czynnikami:

- Dużymi bazami danych zarówno zaetykietowanych jak i bez etykiet. Pierwsze są używane do uczenia sieci z nauczycielem (nadzorowane), drugie natomiast są używane do klasyfikacji (ang. clustering) dużych zbiorów danych o nieznanym liczbie klas. W strategii uczenia zbiory danych pierwotnych dzielone są na trzy rozłączne podzbiory: uczenia, testowania oraz weryfikacji.
- Pojawienie się szybkich wieloprocesorowych (wielordzeniowych) komputerów, co pozwala na praktyczną realizację złożonych struktur sieci neuronowych oraz przetwarzanie dużych wielowymiarowych zbiorów danych w oparciu o zasadę równoległego przetwarzania.

Konfigurację wielowarstwowej, głębokiej sieci neuronowej przedstawiono na rysunku 1.



Rys. 1. Sieć standardowa wielowarstwowa jednopoziomowa

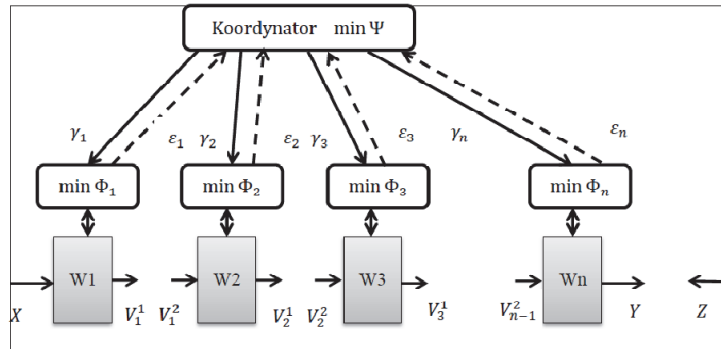
Sieć neuronowa składa się z jednolitych topologicznie i logicznie modułów, zwanych warstwami. W celu uniknięcia nieporozumienia, warstwy sieci liczymy od pierwszej warstwy ukrytej, kończąc na ostatniej warstwie wyjściowej. Tak więc schemat sieci na rysunku 1, zawiera „L” warstw, w tym „L-1” ukrytych i jedną wyjściową. Wymiary macierzy wag wszystkich warstw sieci uzależnione są od rozmiarów wektorów wyjścia poszczególnych warstw. Wzdłuż sieci następuje kompresja danych wraz z ekstrakcją elementów cech. W warstwie wyjściowej następuje końcowe sumowanie składowych częściowych cech. Proces ten jest bardzo skomplikowany i jak dotychczas nie istnieje teoria, która w oparciu o wartości współczynników wag macierzy poszczególnych warstw może określić wyselekcjonowane cechy. Należy zwrócić uwagę, że wektor wyjściowy X warstwy „l” jest jednocześnie wektorem wejściowym warstwy „l+1”.

1.2. Hierarchiczna struktura sieci neuronowej

Zgodnie z [1] złożony system techniczny, przyrodniczy, a także złożoną sztuczną sieć neuronową, trudno jest opisać dokładnie i precyzyjnie używając pojęć i terminologii tylko z jednej dziedziny. Problemem jest konflikt pomiędzy prostotą opisu, a dokładnością. W [2] przedstawiono dwa różne sposoby opisy hierarchicznej struktury SSN.

- Wprowadzając pojęcie straty lub stratyfikowanego opisu przyjmujemy zbiór modeli z różnych dziedzin nauki i techniki. Każdy model używa swoich zmiennych i terminologii o różnym poziomie abstrakcji. Tak więc dla holistycznego zrozumienia SSN należy zapoznać się z koncepcjami i pojęciami stosowanymi w każdej stracie. Przejrzyste zhierarchizowanie pojęć ułatwia zrozumienie procesu analizy lub syntezy SSN.
- Przeprowadzając identyfikacje nieliniowych stacjonarnych systemów z użyciem SSN wystarczy tylko raz przeprowadzić proces uczenia sieci w oparciu o dostępne dane wejściowe i wyjściowe. Strojenie parametrów sieci odbywa się w słoju optymalizacji minimalizującym błąd średniokwadratowy. Sytuacja ulega skompilowaniu w przypadku procesów niestacjonarnych, w których wartości parametrów mogą zmieniać się w czasie. W miarę upływu czasu, SSN z coraz mniejszą dokładnością odwzorowuje pierwotny niestacjonarny proces. Należy dokonywać ciągłej korekty parametrów poprzez wprowadzenie kolejnego poziomu podejmowania decyzji – słoju adaptacji. Dla bardziej złożonych zagadnień, zastosowanie słoju samoorganizacji będzie celowe a wręcz konieczne. Wielo-słojowa struktura podejmowania decyzji będzie optymalną konfiguracją w przypadku nieliniowych, niestacjonarnych procesów.

Dotychczas wprowadzone pojęcia realizują pionową dekompozycję pojęć i algorytmów uczenia. Nie pokazują w sposób przejrzysty konkretnej struktury lub koncepcji algorytmu uczenia. W tym też celu wprowadzamy opis dwuwarstwowej koncepcji algorytmu uczenia SSN (rys. 2). Na pierwszym poziomie, znajdują się autonomiczne podsystemy z lokalnymi funkcjami celu $\Phi_1, \Phi_2, \dots, \Phi_L$. Celem każdego podsystemu jest minimalizacja lokalnej funkcji celu poprzez iteracyjną modyfikację współczynników macierzy w SSN podzielonej na warstwy. Natomiast na drugim poziomie umieszczamy koordynator z własną funkcją celu „ Ψ ”. Głównym, chociaż nie jedynym celem koordynatora, jest takie skoordynowanie podsystemów pierwszego poziomu, żeby osiągnąć końcowy cel algorytmu uczenia, czyli zminimalizować globalną funkcję celu Φ .



Rys. 2. Hierarchiczna struktura algorytmu uczenia sieci

Tak więc, poprzez jawną dekompozycję SSN na poziomy, można zaproponować nową efektywną strukturę algorytmu uczenia opartą na koordynacji podsystemów, czyli koordynacji podzadań pierwszego poziomu.

1.3. Zasady koordynacji

Zadanie koordynacji nie jest prostym zadaniem. Powyższe wynika z kilku przesłanek, a mianowicie:

- SSN oraz algorytmy uczenia w swojej podstawowej strukturze są zadaniami nieliniowymi, które rozwiązuje się metodami iteracyjnymi.
- Są to zadania wielowymiarowe, gdzie wymiary wektorów wejściowych, ukrytych jak i wyjściowych mogą być naprawdę duże, np. zawierające tysiące, a nawet miliony pikseli dla obrazów 3D.
- Dekomponując podstawową strukturę SSN w sposób jawny na warstwy i przypisując lokalnym podsystemom swoje funkcje celu, wprowadza się podwójną sytuację konfliktową: pomiędzy podzadaniami pierwszego poziomu oraz konflikt pomiędzy poziomami – pierwszego poziomu i koordynatorem. W pierwszym przypadku mówi się o konflikcie wewnętrznym poziomu pierwszego, natomiast drugi to konflikt między poziomami w wewnętrznej strukturze.

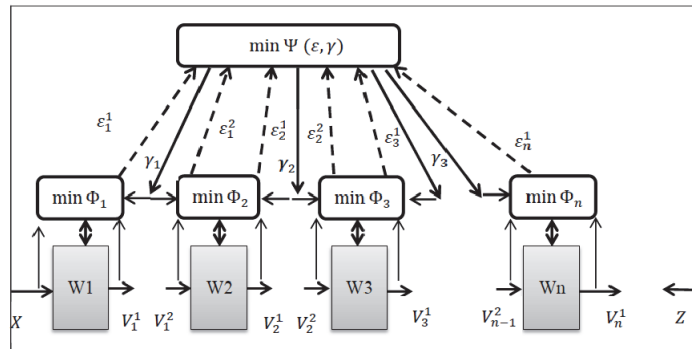
Głównym zadaniem koordynatora jest więc niedopuszczenie do powstania konfliktów, a w przypadku zaistnienia, koordynator musi podjąć decyzje (rozwiązania) usuwające przyczynę konfliktu. W celu znalezienia przyczyn konfliktu, definiujemy:

- Globalną funkcję celu uczenia SSN, która jest zależna nie tylko od wektorów wejściowego „X” i uczenia „Z” lecz również od całej struktury sieci wyrażonej zbiorem macierzy $W = (W_1, W_2, \dots, W_L)$, czyli $\Phi(X, Z, W)$,

- Funkcją celu koordynatora Ψ , która zależy od sygnałów sprzężenia zwrotnego $\varepsilon = (\varepsilon_1, \varepsilon_2, \dots, \varepsilon_L)$ jak również od wypracowanych przez koordynator sygnałów koordynujących $\gamma = (\gamma_1, \gamma_2, \dots, \gamma_L)$, czyli $\Psi(\gamma, \varepsilon)$. Sygnały sprzężenia zwrotnego wypracowane są w każdej iteracji przez podsystemy pierwszego poziomu i przesyłane do koordynatora (rys. 3).
- Zbiór funkcji celu podsystemów pierwszego poziomu $\Phi_1, \Phi_2, \dots, \Phi_L$. Każda funkcja $\Phi_i(\gamma_i)$, dla $i = 1, 2, \dots, L$, zależy od swojego wektora wejściowego i wyjściowego, macierzy podsystemu oraz koordynującego parametru γ_i , (rys. 3). Koordynator, w każdej iteracji, na podstawie swojej własnej funkcji celu Ψ oraz sygnału sprzężenia zwrotnego ε , oblicza nowe, lepsze wartości sygnału koordynującego γ .

Otwarte pozostaje pytanie, jaką strategię powinien zastosować koordynator, wypracowując w iteracyjnym procesie wymiany informacji pomiędzy podsystemami pierwszego poziomu, a koordynatorem, nowe wartości wektora koordynującego $\gamma(n+1)$ w oparciu o wektor błędu $\varepsilon(n)$. W teorii systemów hierarchicznych [1], zaaprobowano trzy zasady koordynacji oraz zdefiniowano warunki jakie muszą spełniać wszystkie podsystemy w celu rozwiązania konfliktów. Dla wielowarstwowych, hierarchicznych systemów [1], definiuje się trzy prawa koordynacji:

- Predykcja (prognoza) wektorów powiązań (interfejsów) pomiędzy warstwami SSN. Tak więc, jednym z zadań koordynatora jest takie określenie wartości wektorów koordynacji $\gamma = (\gamma_1, \gamma_2, \dots, \gamma_L)$, aby rzeczywiste wartości sygnałów międzywarstwach $(V_1^1, V_1^2, V_2^1, V_2^2, \dots, V_{L-2}^2, V_{L-1}^2)$ były równe wartościom prognozowanym (rys. 3.). Koordynator prognozując wartości powiązań, oddziałuje na podzadania pierwszego poziomu w małej skali.



Rys. 3. Struktura algorytmu koordynacji metodą prognozowania wartości interfejsów

- Rozwiązywanie (uwolnienie) wektorów powiązań pomiędzy warstwami. Przyjmuje się, że podzadania pierwszego poziomu są maksymalnie niezależne poprzez pełne uwolnienie interfejsów. Podzadania pierwszego poziomu muszą optymalizować swoje funkcje celu poprzez dobór nie tylko współczynników macierzy lecz również wartości interfejsów. W tym miejscu warto podkreślić, że koordynator może oddziaływać na podzadania pierwszego poziomu tylko poprzez wartości lokalnych funkcji celu. Ten sposób koordynacji nazywany jest również koordynacją w dużej skali.
- Estymacja powiązań wartości wektorów pomiędzy warstwami. To prawo koordynacji jest rozszerzeniem prawa pierwszego, w którym koordynator zadaje dokładne wartości prognozowanych powiązań. Tym razem koordynator zwiększa swobodę wyboru wartości interfejsów, poprzez określenie przedziałów, w których podzadania pierwszego poziomu wybierają wartości powiązań.

2. RÓWNANIA

2.1. Algorytm wstecznej propagacji błędu dla sieci standardowej

W oparciu o rysunek 1 możemy zapisać przetwarzanie w przód w formie łańcucha.

$$X_0 \rightarrow W^1 \rightarrow U^1 \rightarrow X^1 \rightarrow W^2 \rightarrow X^2 \rightarrow \dots \rightarrow X^{L-1} \rightarrow W^L \rightarrow X^L \rightarrow \Phi \quad (1)$$

Pochodną funkcji celu względem wektora X^{L-1} obliczamy jako pochodną funkcji złożonej:

$$\frac{\partial \Phi}{\partial X^{L-1}} = \frac{\partial \Phi}{\partial X^L} \cdot \frac{\partial X^L}{\partial U^L} \cdot \frac{\partial U^L}{\partial X^{L-1}} \quad (2)$$

gdzie: wektor U – wektor pomocniczy w danej warstwie, wektor X- wektor danych wejściowych lub wyjściowych w zależności od indeksu górnego

$$\frac{\partial \Phi}{\partial X^L} = X^L - Y \quad (3)$$

$$\frac{\partial X^L}{\partial U^L} = \sigma'(U^L) \quad (4)$$

Wzór (2) określa sposób przekazywania części błędu funkcji celu do następnej warstw „L-1”. W podobny sposób określa się pochodną funkcji celu względem wag warstwy „L”

$$\frac{\partial \Phi}{\partial W^L} = \frac{\partial \Phi}{\partial X^L} \cdot \frac{\partial X^L}{\partial U^L} \cdot \frac{\partial U^L}{\partial W^L} \quad (5)$$

W celu poprawnego przekazania błędu funkcji celu z warstwy „l+1” do warstwy „l”, utwórzmy odwrotny ciąg przekształceń podobny do wzoru (1).

$$W^l \leftarrow \delta^l \leftarrow \sigma'(U^l) \leftarrow X^l \leftarrow W^{l+1} \leftarrow \sigma^{l+1} \leftarrow \dots \quad (6)$$

Uniwersalny wzór przekazywania błędu z warstwy „l+1” do warstwy „l”

$$\delta_i^l = \sigma'(u_i^l) \cdot \sum_{k=1}^{k^{l+1}} w_{ki}^{l+1} \cdot \delta_k^{l+1} \quad (7)$$

Ostatecznie dla danej warstwy „l” pochodną funkcji celu względem danej wagi oblicza się wg wzoru:

$$\frac{\partial \Phi}{\partial w_{ij}^l} = x_i^{l-1} * \delta_j^l \quad (8)$$

Ostatecznie, stosując metodę gradientowa obliczenia wartości wag w poszczególnych iteracjach, określa się wzór obowiązujący dla każdej warstwy (dla uproszczenia zapisu opuszczono górny indeks „l”).

$$w_{ij}^{l+1} = w_{ij}^l - \alpha \cdot \frac{\partial \Phi}{\partial w_{ij}^l} \quad (9)$$

2.2. Algorytm koordynacji dwupoziomowej sieci neuronowej

Rysunek 3 przedstawia algorytm uczenia zdekomponowanej sieci neuronowej. Sygnał koordynacji zadawany przez koordynator drugiego poziomu, prognozuje wartości powiązań interfejsów pomiędzy warstwami, spełnia jednocześnie dwie funkcje. Dla lewej podsieci jak i wewnętrznych podsieci przekazuje do funkcji celu wartość zadaną wektora wejściowego. Tak więc sygnał koordynujący precyzuje jakie wartości powinny osiągnąć wektory wyjściowe danej podsieci.

Dla podsieci wejściowej, definiuje się lokalną funkcję celu:

$$(W1, X, \gamma) = \frac{1}{2} \sum_{i=1}^{N_1} \sum_{p=1}^{N_p} (v1_i^p - \gamma_i^p)^2 = \frac{1}{2} \sum_{i=1}^{N_1} \sum_{p=1}^{N_p} \left(f \left(\sum_{j=0}^{N_0} W1_{ij} \cdot x_j \right) - \gamma_i^p \right)^2 \quad (10)$$

gdzie: $\gamma [1:N_1, 1:N_p]$ – Sygnał koordynatora dla całej paczki, N_1 - liczba neuronów wyjściowych pierwszej podsieci, N_0 - liczba neuronów wejściowych pierwszej podsieci.

Tak więc podsieć przyjmuje tylko jedną wartość prognozy dla sygnału wyjściowego.

Dla podsieci ukrytej, lokalna funkcja celu przyjmuje postać

$$(W_u, \gamma_{u-1}, \gamma_u) = \frac{1}{2} \sum_{i=1}^{N_1} \sum_{p=1}^{N_p} (v1_i^p - \gamma_i^p)^2 = \frac{1}{2} \sum_{i=1}^{N_1} \sum_{p=1}^{N_p} \left(f \left(\sum_{j=0}^{N_0} W u_{ij} \cdot \gamma_j^p \right) - \gamma_i^p \right)^2 \quad (11)$$

Koordinator nie ma bezpośredniego kontaktu z SSN. Może oddziaływać na wartości wewnętrznych parametrów SSN poprzez podsieci pierwszego poziomu. Do tego celu, wykorzystując prawo prognozowania powiązań oraz prawo priorytetu oddziaływania, narzuca w sposób arbitralny podsiociom pierwszego poziomu wartości wejść i wyjść. Niestety jego prognozy nie zawsze są trafne, podsieci wykorzystując sygnały sprzężenia zwrotnego informują koordynator o rozbieżnościach pomiędzy prognozą i realnymi wartościami powiązań.

Jako jedną z funkcji celu koordynatora, wybiera się sumaryczny błąd średniokwadratowy dla wszystkich podsieci i całej epoki.

$$\Psi = \frac{1}{2} \sum_{p=1}^{N_p} \sum_{i=1}^{N_1} (\varepsilon 1_i^p - \gamma_i^p)^2 + \frac{1}{2} \sum_{p=1}^{N_p} \sum_{i=1}^{N_1} (\gamma_i^p - \varepsilon 2_i^p)^2 \quad (12)$$

Wartość sygnału koordynatora w następnej iteracji oblicza się ze wzoru (13):

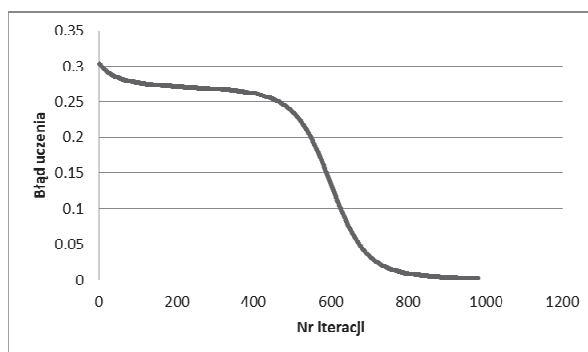
$$\gamma_i^p(n+1) = \gamma_i^p(n) - \lambda_1 \cdot \frac{\partial \Psi}{\partial \gamma_i^p} \quad (13)$$

gdzie: λ_1 - współczynnik uczenia dla koordynatora.

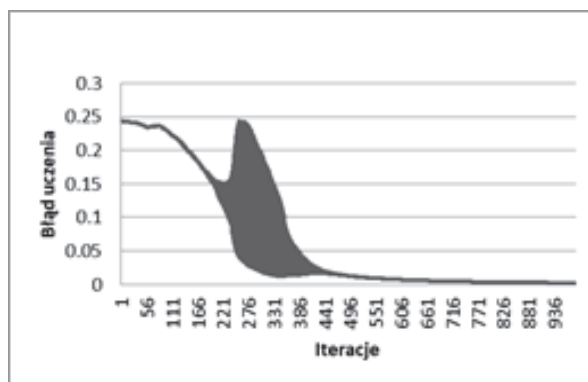
3. PRZYKŁAD OBLICZENIOWY

W charakterze przykładu przedstawiono uczenie trzech różnych sieci neuronowych. Pierwsza sieć składa się z 4 warstw z identycznymi funkcjami aktywacji typu sigmoid. Konfigurację sieci opisuje się jako 4-4-4-1, czyli sieć zawiera trzy warstwy ukryte oraz warstwę wyjściową z jednym neuronem. Epoka wektorów uczących zawiera 160 danych. Na rys. 4. przedstawiono kształt funkcji celu w zależności od numeru iteracji.

Uczenie sieci odbywa się metodą off-line (modyfikacja współczynników wagowych odbywa się po podaniu na wejście sieci całej paczki). Błąd uczenia obliczony dla warstwy wyjściowej, przekazywany jest przez algorytm wstecznej propagacji błędów do kolejnych warstw ukrytych. W charakterystyce uczenia można zauważyć stosunkowo duże opóźnienie. Powyższe można wytłumaczyć właśnie czasem potrzebnym na przekazanie błędów do wszystkich warstw.



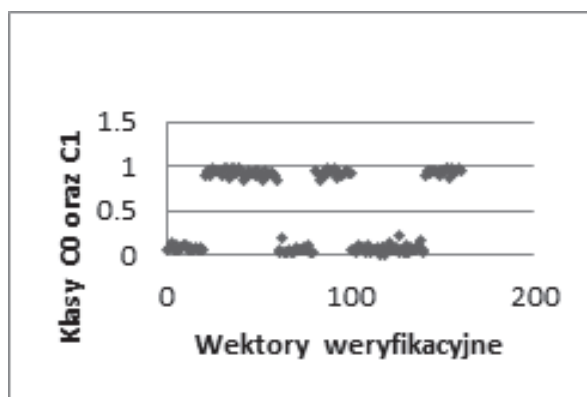
Rys. 4. Dynamika procesu uczenia standardowej sieci neuronowej



Rys. 5. Dynamika uczenia sieci z funkcjami aktywacji typu ReLU

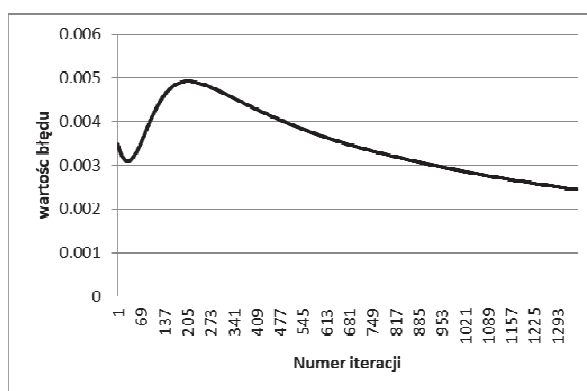
Na rysunku 5 przedstawiono charakterystykę uczenia sieci o tej samej konfiguracji lecz warstwy ukryte wyposażono w funkcje aktywacji typu ReLU.

Jak wynika z analizy teoretycznej, błąd uczenia (sprzężenie zwrotne) przekazywany w głąb sieci jest większy, a tym samym sieć jest bardziej narażona na występowanie wewnętrznych oscylacji. Opóźnienie jest dużo mniejsze lecz w środkowej części występują gwałtowne oscylacje. Powyższe zjawisko tłumaczy się występowaniem zbyt wysokiej wartości współczynnika uczenia „ α ” w warstwach ukrytych. Dobór optymalnej lub zadawalającej wartości współczynnika jest zagadnieniem bardzo trudnym i wymaga zastosowania adaptacyjnych metod doboru wartości.



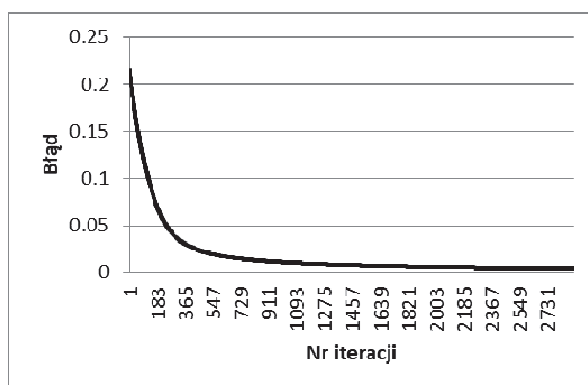
Rys. 6. Wynik uczenia sieci w zadaniu klasyfikacji dwóch klas

Powyższe przebiegi uczenia sieci porównywane są z nową strukturą algorytmu uczenia dla sieci zdekomponowanej i skoordynowanej oddzielnym procesem. Zgodnie z rysunkiem 3, koordynator koordynuje rozwiązania cząstkowe poszczególnych sieci. Spełnia więc rolę propagatora błędów od warstwy wyjściowej do wszystkich warstw ukrytych.

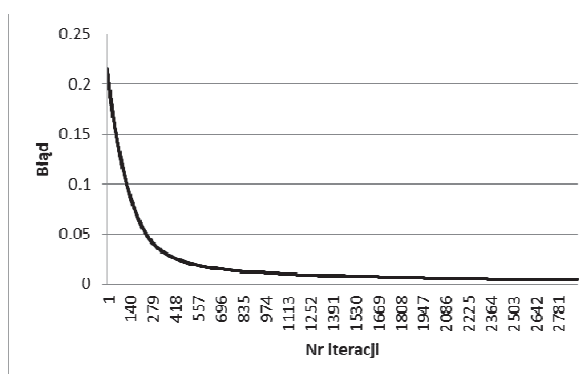


Rys. 7. Charakterystyka uczenia pierwszej podsieci ukrytej

Na rysunku 7 i 8 przedstawiono charakterystyki uczenia pierwszej warstwy ukrytej (czyli ostatniej licząc od wyjścia sieci) oraz trzeciej. Przebiegi są bardzo różne. W warstwie pierwszej występują przeregulowania, co można tłumaczyć działaniem koordynatora, który dostarczał informacji o błędzie z pewnym opóźnieniem. Na rysunku 9 podsumowano działanie koordynatora, który asymptotycznie sprowadza błąd uczenia do zera.



Rys. 8. Charakterystyka uczenia trzeciej podsieci



Rys. 9. Optymalna strategia koordynatora w uczeniu sieci

4. PODSUMOWANIE

Podsumowując, można powiedzieć, że dynamika uczenia standardowej sieci i sieci zdekomponowanej są podobne co do jakości i kształtów charakterystyk uczenia. Gwałtowne jakościowe zmiany występują po zastosowaniu funkcji aktywacji typu ReLu. Uczenie sieci wielowarstwowych (warstw więcej niż 5) wymaga wnikliwej analizy wpływu poszczególnych składników sieci na jakość procesów uczenia w każdej warstwie. Zagadnienie to będzie analizowane w kontekście optymalnego i adaptacyjnego doboru współczynników uczenia „ α ”.

LITERATURA

- [1] Mesarovic M. D., Macko D., Takahara Y., Theory of hierarchical, multilevel systems., Academic Press, New York and London 1970.

- [2] Płaczek S., Hierarchiczne struktury w sztucznych sieciach neuronowych, Zeszyty Naukowe Uczelni Vistula 38/2014.
- [3] Płaczek S., Adhicari B., Analysis of Multilayer Neural Network with Direct and Cross – Forward Connection, CS7P Conference in the University of Warsaw, Warsaw 2013.
- [4] Quoc V. Le, Part 1: Nonlinear Classifiers and The Backpropagation Algorithm, Google Brain, Google Inc, CA 94043 2015.
- [5] Goodfellow I., Bengio Y., Courville A., Deep Learning, 2016.
- [6] Raschka S., Python Machine Learning.

**A COMPARISON OF THE PERFORMANCE CHARACTERISTICS
OF A TEACHING ALGORITHM FOR A STANDARD NEURAL NETWORK
AND A DECOMPOSED NETWORK COORDINATED BY THE INTERFACE
FORECASTING METHOD**

A Standard Neural Network is defined as an integrated module of a set of layers with both forward and full weight coefficient connections in all layers. Every layer is built by the matrix of the weight coefficients connecting an input vector X with an internal vector U , which, in the next step, is the input of the activation function, and the output vector Y is calculated. For these kinds of neural networks, the teaching algorithms are well known. Unfortunately, in an algorithm practice realization, a lot of numeric problems appear to achieve fast convergence. A lot of components have negative impacts on the entire calculation process. In the article, a decomposed network replaces a level in a multilayer network. A network is built by independent layers in the first level and the coordinator in the second. Layers have to solve their local optimization task using their own algorithms. Local solutions are coordinated by the coordinator. The coordinator, working together with the first level, is responsible for solving the global optimization task, which is laid outside the network. Finally, a network is ready to classify new input data. In the article, quality and quantity characteristics for these two networks are compared.

(Received: 07.02.2018, revised: 12.03.2018)