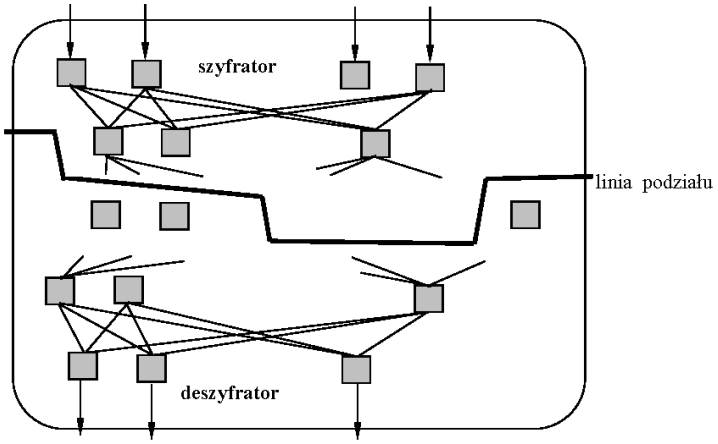
Please cite this article as:
Henryk Piech, Aleksandra Ptak, Wojciech Dobrzański, Dariusz Leks, Koncepcja neuronowej kryptografii, Scientific Research of the Institute of Mathematics and Computer Science, 2002, Volume 1, Issue 1, pages 177-185.
The website: http://www.amcm.pcz.pl/

KONCEPCJA NEURONOWEJ KRYPTOGRAFII

Henryk Piech, Aleksandra Ptak, Wojciech Dobrzański, Dariusz Leks

Instytut Matematyki i Informatyki, Politechnika Częstochowska

Streszczenie: Istotą kryptografii jest użycie narzędzi szyfrujących przez wysyłającego wiadomość oraz deszyfrujących przez odbiorcę. Zastosowanie neuronów do szyfrowania lub deszyfrowania może być zrealizowane na wiele różnych sposobów. Koncepcja prezentowana przez autorów zakłada rozdzielenie struktury neuronowej na dwie lub więcej części (rys. 1). Części te zostaną zgrupowane w dwa zbiory: zbiór szyfrujący oraz zbiór deszyfrujący. Zbiory te są całkowicie niezależne.



Rys. 1. Idea rozdziału struktury neuronowej na układ szyfrujący i deszyfrujący

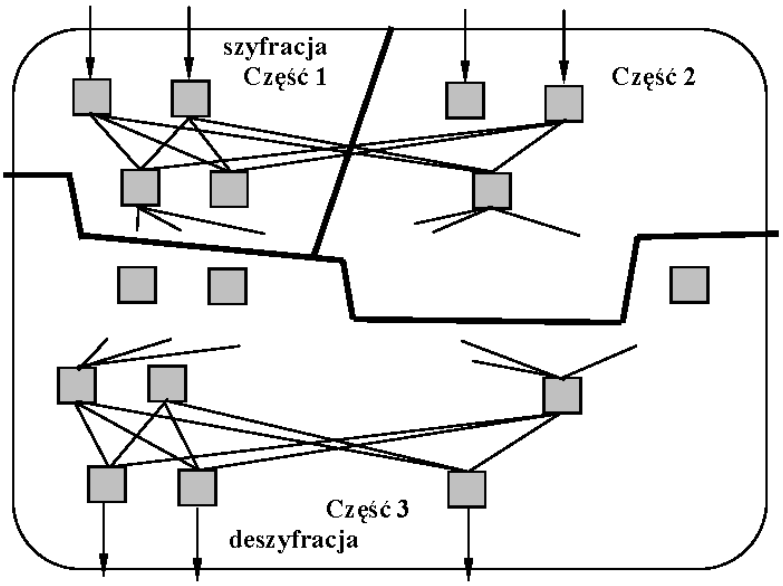
Pozostaje kwestia analizy jakości szyfrowania i analizy, czy tak zaszyfrowane dane nie będą mogły zostać z łatwością rozszyfrowane bez pomocy deszyfratora. Porównanie z klasycznymi wariantami szyfrowania powinno dać podobne rezultaty. Uczenie sieci neuronowej w pełnym strukturalnym ujęciu może zakładać: tożsamość danych wejściowych, przejście w inny układ liczbowy, zmianę skali przedstawienia danych, przesunięcie oraz inne modyfikacje danych wejściowych.

Wprowadzenie

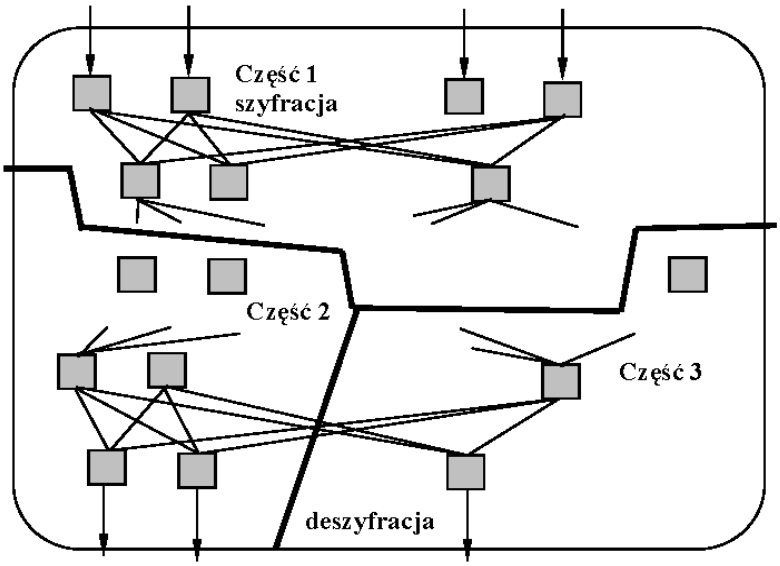
Podzielmy przykładową strukturę neuronową (rys. 1) na dwie części:

$$\left\{\begin{array}{l}1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9\\ 1\ 2\ 3\ 4\ 5\\ 1\ 2\ 3\ 4\ 5\ 6\ 7\\ 1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\end{array}\right\} = \left\{\begin{array}{l}1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9\\ 1\ 2\ 3\ \ldots\ldots\ldots\\ \ldots\ldots\ldots\ldots\ldots 5\ 6\ 7\\ \ldots\ldots\ldots\ldots\ldots\end{array}\right\} + \left\{\begin{array}{l}\ldots\ldots\ldots\ldots\ldots\ldots\ldots\\ \ldots\ldots\ldots\ 4\ 5\\ 1\ 2\ 3\ 4\ \ldots\ldots\ldots\\ 1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\end{array}\right\}$$

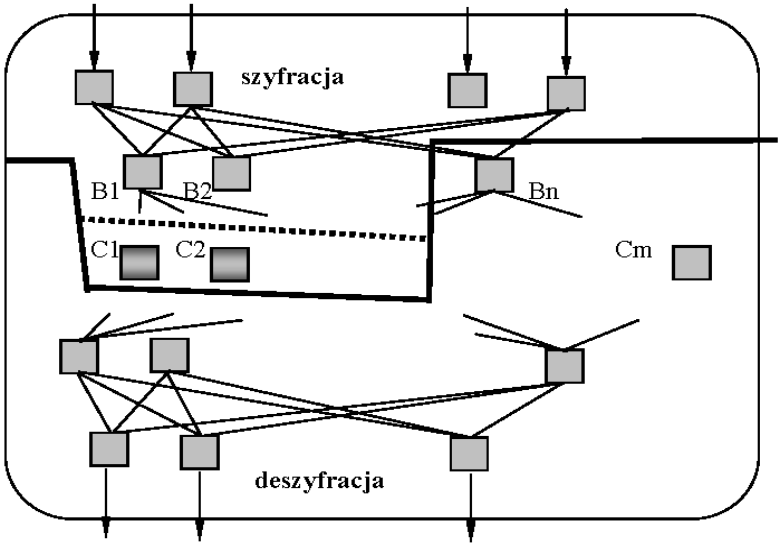
Skrótowo można to zapisać: $\boxed{A} = \boxed{B} + \boxed{C}$. Strukturę A można rozbić na kilka części. Przy takim rozbiciu może dojść do sytuacji, kiedy będziemy mieli dostarczane dane z kilku źródeł (rys. 2) lub też taką, kiedy złączenie kilku kluczy deszyfrujących daje finalny rezultat (rys. 3). Może też zaistnieć sytuacja, kiedy niepełne dane stworzą błędne pośrednie rezultaty (rys. 4).



Rys. 2. Rozdzielenie danych wejściowych; brak części danych fałszuje rezultat końcowy



Rys. 3. Podział części deszyfrującej; brak jednej z części uniemożliwia deszyfrację



Rys. 4. Brak kompletnych danych wejściowych dla neuronów C1, C2 ... (brak na przykład danych z wyjścia Bn) powoduje pojawienie się dezinformacji na ich wyjściu

Obecność w części szyfrującej elementów informacji i dezinformacji wymusza wyselekcjonowanie właściwej części danych i skierowanie ich do deszyfratora. Wywołuje to również konieczność wykorzystania wag neuronów C1, C2 ... do procesu deszyfracji (pod linią). Potrzebne będą również dane wyjściowe z neuronów B1, B2,..., Bn, czyli z całej poprzedniej warstwy.

1. Opis funkcjonowania kryptograficznych narzędzi i realizowanych przez nie procedur

Narzędzie do szyfracji można opisać jako zestaw neuronowej struktury uzupełnionej o wagi w poszczególnych węzłach oraz o funkcje realizowane przez węzły S = < Ws, Fs >, gdzie

$$Ws = \left\{ \begin{array}{l} w11\ ,\ w12,\ \ ...\ \ ,\ w1n1 \\ w21\ ,\ w22\ ,\ \ \ ...\ \ \ ,\ w2n2 \\ 0\ \ \ ,\ w32\ ,\ w33\ ,\ ...\ ,\ 0\ ,\ ...\ ,\ w3n3 \\ ..... \end{array} \right\}$$

macierz wag z wyłączeniem wierszy o wszystkich wagach zerowych (część szyfrująca)-wariant przykładowy

$$Fs = \left\{ \begin{array}{l} f11\ ,\ f12,\ \ ...\ \ ,\ f1n1 \\ f21\ ,\ f22\ ,\ \ \ ...\ \ \ ,\ f2n2 \\ *\ \ \ ,\ f32\ ,\ f33\ ,\ ...\ ,\ *\ ,\ ...\ ,\ f3n3 \\ ..... \end{array} \right\}$$

macierz funkcji (część szyfrująca) wariant przykładowy

Podobnie deszyfrator może być zdefiniowany jako D = < Wd, Fd >, gdzie

$$Wd = \left\{ \begin{array}{l} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ w31\ ,\ 0\ ,\ 0\ ,\ \ldots\ ,\ w3i\ ,\ \ldots\ ,\ 0 \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ wm\text{-}11\ ,\ wm\text{-}12\ ,\ \ldots\ ,\ wm\text{-}1nm\text{-}1 \\ wm1\ ,\ wm2\ ,\ \ \ldots\ \ ,\ wmn1 \end{array} \right\}$$ **macierz wag z wyłączeniem wierszy o wszystkich wagach zerowych (część deszyfrująca)- wariant przykładowy**

$$Fd = \left\{ \begin{array}{l} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ f31\ ,\ 0\ ,\ 0\ ,\ \ldots\ ,\ w3i\ ,\ \ldots\ ,\ f3n3 \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ fm\text{-}11\ ,\ fm\text{-}12\ ,\ \ldots\ ,\ fm\text{-}1nm\text{-}1 \\ fm1\ ,\ fm2\ ,\ \ \ldots\ \ ,\ fmnm \end{array} \right\}$$ **macierz funkcji (część deszyfrująca) - wariant przykładowy**

tutaj **wij** to wektor wag określonych dla par neuronów sąsiadujących warstw [3], **ni** - liczba neuronów i-tej warstwy.

Załóżmy, że chcemy zaszyfrować informację postaci (x1, x2,..., xn1) i użyjemy szyfratora S. W rezultacie otrzymujemy dwa wektory rezultatów zaszyfrowanych: wektor informacji Z = {z1, z2,..., zni} oraz wektor dezinformacji DZ = {d1, d2,..., dnj}(rys. 4)

$$S(x1, x2,..., xn1) = Z + DZ = \{z1, z2,..., zni\} \cup \{d1, d2,..., dnj\} \quad (1)$$

gdzie **zk -** wartość na wyjściu k-tego neuronu i-tej warstwy.

Dane, które stanowią elementy wektora informacji Z, poddajemy procedurze deszyfracji D (pomijamy w tej chwili problem odszukania tych elementów ze zbioru połączonego):

$$D(z1, z2,..., zni) = \{y1, y2,..., ynm\} \quad (2a)$$

$$D(z1, z2,..., zni, d1, d2,..., dnj) = \{y1, y2,..., ynm\} \quad (2b)$$

Załóżmy, iż liczba neuronów pierwszej warstwy szyfrującej **n1** oraz ostatniej warstwy deszyfrującej **nm** są sobie równe (**n1 = nm**) i mamy do czynienia z transformacją typu „jeden do jednego", tzn. dane do szyfracji pokrywają się z rezultatami deszyfracji). Ponadto ustalmy, iż struktura neuronowa działa w obu kierunkach (atrybut wstecznej propagacji). Wówczas **{x1, x2,..., xn1} = {y1, y2,..., ynm}**:

$$S'(y1, y2,..., ynm) = \{z1, z2,..., znj\} \cup \{d1, d2,..., dni\} \quad (3a)$$

$$D'(z1, z2, ..., znj, d1, d2, ..., dni) = D'(z1, z2, ..., znj) = \{x1, x2, ..., xn1\} \quad (3b)$$

$$D(S(x1, x2,..., xn1)) = \{y1, y2,..., ynm\} \quad (4a)$$

gdzie **S'**, **D'** - szyfracja i deszyfracja wsteczna.

Wprowadźmy pojęcie funkcji kryptograficznej **KF(*) = D(S(*))** (lub **KF'(*) = = D'(S'(*))** dla propagacji wstecznej). Oczywiście **KF(KF'(*)) = X** oraz **KF'(KF (*)) = Y**. Ponadto zdefiniujmy grupę neurokryptograficzną jako szóstkę {X, Y, S, D, +, *}. Tym razem X oznacza zbiór danych wejściowych, Y - zbiór danych wyjściowych, S = <Ws,Fs> (szyfrator), D = <Wd,Fd> (deszyfrator), +, * - podstawowe działania w sieci neuronowej [7]. Aby przedstawić etap szyfracji, użyjemy zapisu z pomocą zdefiniowanej grupy {X, Z, S, 0, + ,*}. Fragmenty szyfracji i deszyfracji (części 1, 2, 3 na rys. 3) opiszemy następująco: {X1, Z1, S1, 0 ,+, *}, {Z2, Y2, 0, D2, +, *} oraz {Z3, Y3, 0, D3, 0, +}. Jeśli fragmenty te traktować odrębnie, to składowe Z1, Y2, Y3 są typową dezinformacją, jeśli zaś łącznie, to będą one częściami „właściwej" informacji. Brak któregokolwiek z fragmentów czyni układ szyfrator-deszyfrator bezużytecznym. Również „brak" choćby jednego neuronu dyskwalifikuje układ całkowicie:

$$S = <Ws1,Fs1> + <Ws2,Fs2> + ...+ <Wsp,Fsp> \quad (5)$$

$$D = <Wd1,Fd1> + <Wd2,Fd2> + ...+ <Wdr,Fdr>$$

gdzie:
p - liczba części szyfratora,
r - liczba części deszyfratora.

Podobne zależności można określić dla propagacji wstecznej. Oczywiście części struktury neuronowej dotyczące szyfracji i deszyfracji podlegają zamianie, zmienia się także kierunek formowania sum warstwowych [3]. W zależności od sposobu przygotowania sieci podczas etapu uczenia wagi prostej i wstecznej propagacji mogą być albo równe sobie w obu kierunkach, albo też mogą się różnić od siebie. W pierwszej wersji sygnały wejściowe rozdzielane są proporcjonalnie do wag określonych dla propagacji prostej [4]. W drugiej wersji uczenie w obu kierunkach traktowane jest niezależnie, zamianie podlegają jedynie dane wejściowe i wyjściowe (ze zbiorów uczących).

2. Ocena jakości zabezpieczenia kryptograficznego

Wprowadźmy pojęcie wrażliwości szyfrowania neuronowego sw. Byłaby to minimalna wartość zmiany wagi, która wywołuje jakąkolwiek zmianę kodu wynikowego

$$sw = \min \{wij - w'ij \; ; \; fij - f'ij \neq 0\} \quad (6)$$

$1 \leq i \leq m$
$1 \leq j \leq nm$

gdzie:
j - numer neuronu w i-tej warstwie,

w', f' - wartość skorygowanej wagi oraz wartość zmiany kodu na wyjściu neuronu wywołana korektą wybranej wagi,

m - ilość warstw, ni - ilość neuronów w i-tej warstwie.

Definicję wrażliwości oparto na założeniu, iż wartości funkcji kodowych f i f' zmieniają się dyskretnie dla określonego układu liczbowego.

Miarą jakości zabezpieczenia kryptograficznego może być liczba wariacji stanów wagowych dla wszystkich neuronów oraz dla pełnego zakresu zmian wag (z krokiem równym wrażliwości szyfrowania)

$$CQS = [1/sw]^{nw,}\ nw = n0*n1 + n1*n2 +. .. + nm\text{-}1*\ nm \qquad (7)$$

gdzie: CQS - współczynnik jakości zabezpieczenia, nw - ilość wag.

Przykładowo jeśli sw = 0.01 oraz liczba neuronów w układzie szyfrująco-deszyfrującym wynosi 50 (10 + 10 + 10 + 10 + 10, tzn. po 10 neuronów w pięciu warstwach), to współczynnik jakości zabezpieczenia kryptograficznego wyniesie 100 do potęgi 400, a więc będzie to liczba 801-cyfrowa.

W oszacowaniu nie uwzględniono jeszcze roli dezinformacji, która zwiększa jakość zabezpieczenia. Ocena chaosu wprowadzonego przez wymieszanie informacji prawdziwej i dezinformacji mogłaby być przeprowadzona na różne sposoby. Wyraźmy to na przykład stosunkiem liczby wariacji stanów wag wykorzystywanych do tworzenia dezinformacji do liczby wariacji stanów wag zastosowanych do generowania pełnej informacji. Załóżmy, iż ostatnia pełna warstwa szyfratora ma numer j oraz że ilość warstw niepełnych wynosi k. Liczba neuronów w warstwach niepełnych wynosi odpowiednio: ne1, ne2,..., nek. Stąd ilość połączeń wagowych wykorzystywanych do tworzenia dezinformacji wynosi:

$$md = n0*n1 + n1*n2 + ... + nj\text{-}1*nj + nj*ne1 + ne1*ne2 + ...+ nek\text{-}1*nek \qquad (8)$$

a do tworzenia pełnej informacji

$$mi = n0*n1 + n1*n2 + ... + nj\text{-}1*nj \qquad (9)$$

Współczynnik oddziaływania chaosu będzie większy od jedności i można go wyrazić następująco:

$$CCH = (ws*[1/sw]^{md-mi} + [1/sw]^{mi})/\ [1/sw]^{mi} = 1 + ws*[1/sw]^{md-2mi} \qquad (10)$$

gdzie ws - współczynnik skali prezentacji dobierany dla zakresu (md – 2mi). W tabeli 1 „uwypuklono" zakres md = [20 – 16], mi = 10, stosując współczynnik ws = 1000.

Współczynnik **CCH** wskazuje na nikłość zabezpieczenia, które wynika z wprowadzenia dezinformacji (**CCH** ≅ 1).

Znacznie bardziej wyraziste oddziaływanie chaosu informacyjnego uzyskamy dzięki zastosowaniu współczynnika zdefiniowanego jako stosunek sumy liczby

neuronów w warstwach wyjściowych informacyjnej i dezinformacyjnej do liczby neuronów w wyjściowej warstwie informacyjnej

$$CCN = (ni + nk)/ni = 1 + nk/ni \tag{11}$$

Tabela 1

mi = 10 sw / md	20	18	16	14	12	10	8
	Współczynnik oddziaływania chaosu informatycznego CCH						
0,002	1001	1,004	1	1	1	1	1
0,004	1001	1,016	1,0000003	1	1	1	1
0,006	1001	1,036	1,0000013	1	1	1	1
0,008	1001	1,064	1,0000041	1	1	1	1
0,01	1001	1,1	1,00001	1	1	1	1
0,02	1001	1,4	1,00016	1	1	1	1
0,03	1001	1,9	1,00081	1,000001	1	1	1

Aby uzyskać ostateczny poziom zabezpieczenia, przemnażamy współczynnik jakości (7) przez współczynnik oddziaływania chaosu (10) lub (11)

$$CS = CQS * CCH \text{ (lub CCN)} \tag{12}$$

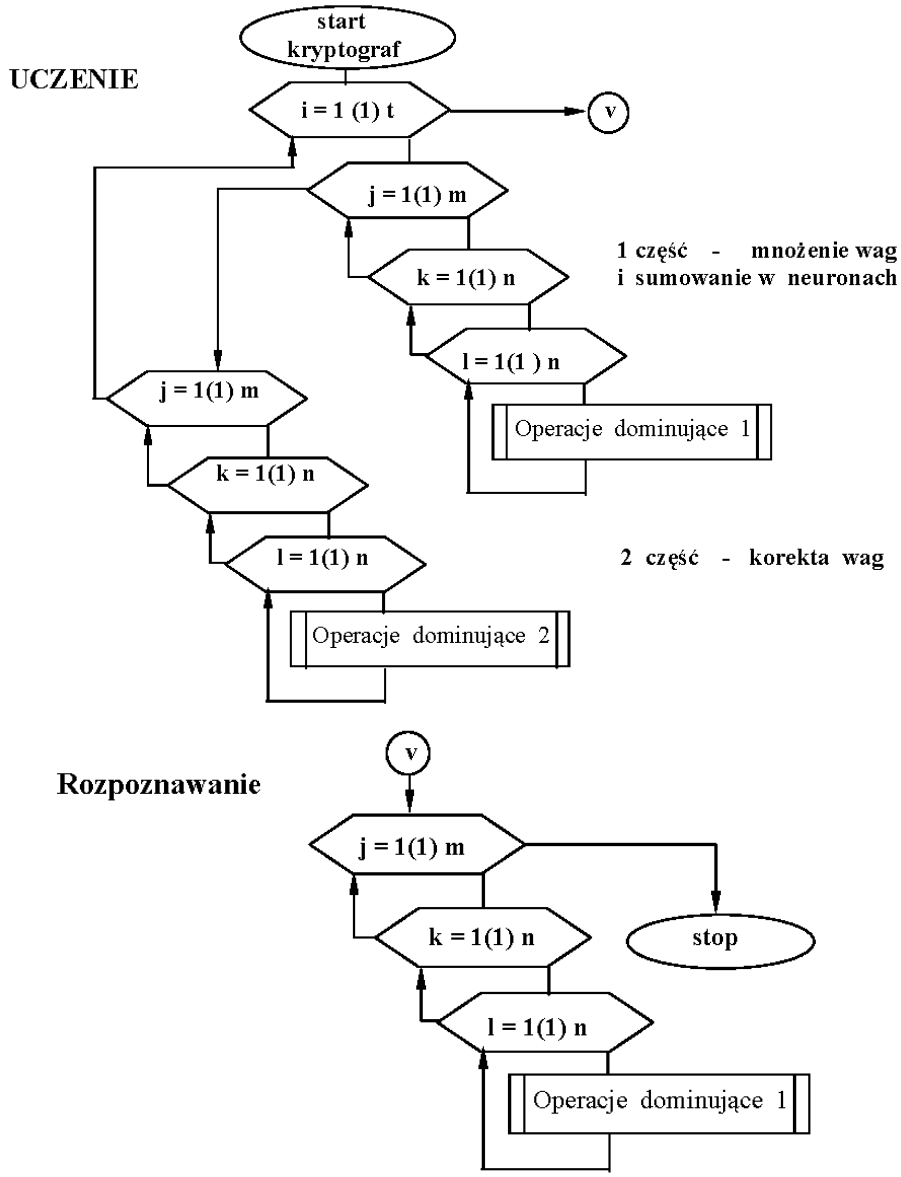
3. Złożoność (czasowa) algorytmu neuronowej szyfracji

Działanie struktury neuronowej zwykle dzielimy na dwa etapy: etap uczenia i etap rozpoznawania. Charakteryzują się one różnymi stopniami złożoności. Patrząc na zagadnienie od strony kryptograficznej należałoby obie fazy zadania kryptograficznego, tzn. szyfrowanie i deszyfrowanie, potraktować integralnie jako jeden spójny algorytm. W algorytmie neuronowej kryptografii w etapie uczenia wyróżnimy następujące cyklicznie powtarzające się działania zależne od następujących parametrów:

- czas uczenia t realizowany w cyklach uczenia
  - w każdym cyklu uczenia przechodzimy przez wszystkie warstwy, których jest m,
  - w każdej warstwie znajduje się średnio n neuronów,
  - do każdego neuronu dostarczana jest informacja średnio z n neuronów poprzedniej warstwy.

Tak więc złożoność pierwszej części uczenia można oszacować następująco: $\mathbf{O(tmn^2)}$. W drugiej części korygujemy według wybranej metody każdą z wag. Złożoność tej części szacujemy w podobny sposób: $\mathbf{O(tmn^2)}$. Złożoność procesu uczenia jest więc równa: $\mathbf{O(tmn^2) + O(tmn^2) = O(tmn^2)}$.

Etap rozpoznawania różni się od etapu uczenia tym, że realizowany jest w jednym cyklu czasowym. Jego złożoność jest więc równa: $\mathbf{O(mn^2)}$ Złożoność całego procesu uczenia i rozpoznawania wynosi: $\mathbf{O(tmn^2) + O(mn^2) = O(tmn^2)}$. Schemat blokowy przedstawiony na rysunku 5 ilustruje problem złożoności neuronowego algorytmu kryptograficznego.



Rys. 5. Schemat blokowy ilustrujący złożoność algorytmu szyfrowania i deszyfrowania neuronowego (i - numer cyklu czasowego, j - numer warstwy, k, l - numery neuronów w sąsiadujących warstwach)

Wnioski

1. Neuronowa kryptografia to elastyczne narzędzie do zabezpieczenia danych, pozwalające na wielokrotne i wielorakie rozdzielenie elementów szyfracji i de-

szyfracji, uwzględniające kryterium veta zarówno od strony odbiorcy, jak i nadawcy zaszyfrowanej informacji.

2. Jakość zabezpieczenia (7) poprzez zastosowanie klucza neuronowego jest porównywalna z jakością zabezpieczenia za pomocą znanych jawnych i prywatnych kluczy [6].
3. Złożoność algorytmu szyfracji (deszyfracji) bazującego na strukturze neuronowej jest o rząd wyższa $\{\mathbf{O(mn^2) > O(n^2)}\}$ niż złożoność klasycznych algorytmów [7], co oczywiście wpływa na wydłużenie czasu operacji kryptograficznych.
4. Elastyczność modyfikowania kluczy neuronowych jest o dwa rzędy większa niż w klasycznych algorytmach, co wynika z modyfikowalnej liczby wag, którą oszacować można następująco: $\mathbf{O(mn^2)}$ $\{\mathbf{O(mn^2) >> O(m)}\}$.
5. Wpływ oddziaływania dezinformacji na stopień zabezpieczenia danych jest niewielki (10) i (11) i możemy go pominąć przy analizie jakości procedur kryptograficznych.

Literatura

[1] Korbicz J., Obuchowicz A., Uciński D., Sztuczne sieci neuronowe - podstawy i zastosowania, Akademicka Oficyna Wydawnicza PLJ, Warszawa 1994.

[2] Osowski S., Sieci neuronowe w ujęciu algorytmicznym, WNT, Warszawa 1996.

[3] Rutkowska D., Piliński M., Rutkowski L., Sieci neuronowe, algorytmy genetyczne i systemy rozmyte, PWN, Warszawa 1997.

[4] Rutkowska D., Inteligentne systemy obliczeniowe, Akademicka Oficyna Wydawnicza, Warszawa 1998.

[5] Tadeusiewicz R., Sieci neuronowe, Akademicka Oficyna Wydawnicza, Warszawa 1993.

[6] Koblitz N., Algebraiczne aspekty kryptografii, WNT, Warszawa 2000.

[7] Schneier B., Kryptografia dla praktyków, WILEY, WNT, Warszawa 2002.