

## Sztuczne sieci neuronowe ANN. Sieci Kohonena

*Artificial neural networks (ANN). Kohonen networks*

Paweł Iljaszewicz<sup>1</sup>

**STRESZCZENIE:** Artykuł omawia sztuczne sieci neuronowe (*ang. ANN- Artificial neural networks*). Jedną z odmian są sieci Kohonena zwane Mapą Samoorganizującą (*ang. SOM – Self Organizing Map*) realizują one proces uczenia się sieci neuronowych samodzielnie tzn. rozpoznają relacje występujące w skupieniach poprzez wykrycie wewnętrznej struktury i kategoryzują je w procesie samouczenia. SOM służy do uformowania odwzorowania z przestrzeni wielowymiarowej do przestrzeni jednowymiarowej lub dwuwymiarowej. Główną cechą SOM jest to, że tworzy on nieliniową projekcję wielowymiarową kolektora danych na regularnej, niskowymiarowej (zwykle 2D) sieci. Na wyświetlaczu klastrów przestrzeni danych, jak również relacje metryczno-topologiczne elementów danych, są wyraźnie widoczne. Jeśli elementy danych są wektorami, składnikami, których są zmiennymi z określone znaczenie, takie jak deskryptory danych statystycznych lub pomiary, które opisują proces, siatka SOM może być wykorzystana, jako podstawa, na której może znajdować się każda zmienna wyświetlane osobno przy użyciu kodowania na poziomie szarości lub pseudo koloru. Ten rodzaj projekcji został uznany za bardzo przydatny do zrozumienia wzajemnych zależności między zmiennymi, a także strukturami zbioru danych.

**SŁOWA KLUCZOWE:** Sieci Kohonena, sieci neuronowe, Mapa samoorganizująca (SOM), WEBSOM

**SUMMARY:** The article discusses artificial neural networks (ANN). One of the varieties is the Kohonen network, called the Self Organizing Map (SOM), that perform the learning process of neural networks independently, i.e. they recognize relationships occurring in clusters by detecting an internal structure and categorizing them in the process of self-learning. SOM is used to form mapping from a multidimensional space to a one-dimensional or two-dimensional space. The main feature of SOM is that it creates a non-linear multi-dimensional projection of a data collector on a regular, low-dimensional (usually 2D) network. On the display, data space clustering as well as metric-topological relations of data elements are clearly visible. If the data elements are vectors, the components of which are variables with defined meanings, such as statistical data descriptors or measurements that describe the process, the SOM grid can be used as a basis on which each variable can be displayed separately using gray or pseudo-color coding. This type of projection has been found to be very useful for understanding the interrelationships between variables as well as data set structures.

**KEYWORDS:** Kohonen networks, artificial neural networks (ANN), Self Organizing Map SOM, WEBSOM.

### 1. Wprowadzenie

Architektury sieci i procesy sygnałowe używane do modelowych układów nerwowych można podzielić na trzy kategorie, każda jest oparta na innej filozofii. Sieci wyprzedzające [1] (*ang. Feedforward networks*) przekształcają zbiory sygnałów wejściowych w zestawy sygnałów wyjściowych. Pożądana transformacja wejścia-wyjścia jest zwykle określany przez zewnętrzną, nadzorowaną regulację parametrów systemu. W sieciach sprzężeń zwrotnych [2] (*ang. feedback network*) informacja wejściowa określa początkowy stan działania układu sprzężenia zwrotnego, a po przejściu przez stan asymptotyczny finalny stan jest identyfikowany, jako wynik obliczeń.

W trzeciej kategorii sąsiadnie komórki w sieci neuronowej

współdziałają poprzez wzajemne oddziaływania boczne i rozwijają się adaptacyjnie w określone detektory o różnych wzorach sygnału. W tej kategorii nazywa się uczenie się konkurencyjne, bez nadzoru ani samoorganizujące się. Sieć Kohonena inaczej samoorganizująca się mapa SOM, należy do kategorii uczenia się konkurencyjnego.

Jest to sztuczna sieć neuronowa przypominająca arkusz, której komórki są specjalnie dostrojone do różnych wzorów sygnałów wejściowych lub klas wzorców przez nie nadzorowany proces uczenia się. Tylko w wersji podstawowej jedna cela lub lokalna grupa komórek na raz daje aktywną odpowiedź na bieżące dane wejściowe. Taki sposób działania nazywamy uczeniem się konkurencyjne, bez

nadzoru, albo samoorganizującym się. Model ten został przedstawiony przez fińskiego naukowca T. Kohonena. Algorytm samouczenia odpowiednio modyfikuje wagowe współczynniki w neuronach składających się na mapę topologiczną, aby zbliżyć je do skupień na podstawie danych uczących [3]. Mapa samoorganizująca się miała być realną alternatywą dla bardziej tradycyjnych architektur sieci neuronowych. Można zapytać, jak "neuron" może być mapą. Jego analityczny opis został już opracowany, bardziej w technicznym niż biologicznym kierunku. Ale osiągnięte wyniki uczenia się wydają się bardzo naturalne, co najmniej wskazując, że same procesy adaptacyjne w pracy na mapie mogą być podobne do spotykanych w mózgu. Może więc istnieć wystarczające uzasadnienie dla kalibrowania tych map "neuronowych sieci" w takim samym znaczeniu, jak ich tradycyjni odpowiedniki.

Mapy samoorganizujące się lub systemy składające się z kilku modułów map, zostały użyte do zadań podobnych do tych, jakie są stosowane przez tradycyjne sieci neuronowe: rozpoznawanie wzorów, robotyka, sterowanie procesem i nawet przetwarzanie informacji semantycznych. Przestrzenna segregacja różnych odpowiedzi i ich organizacja w topologicznie powiązane podzbiory powodują wysoki stopień wydajności w typowych operacjach sieci neuronowych.

Kluczowym wynalazkiem Kohonena było wprowadzenie modelu systemu, który składa się z co najmniej dwóch oddziałujących ze sobą podsystemów o różnym charakterze. Jednym z tych podsystemów jest konkurencyjna sieć neuronowa, która implementuje funkcję zwycięzcy-bierzewszystko (*ang. winner-take-all function*), ale jest też inny podsystem, który jest kontrolowany przez sieć neuronową i który modyfikuje lokalną synaptyczną plastyczność neuronów w uczeniu się. Nauka jest ograniczona przestrzenią do lokalnego sąsiedztwa najbardziej aktywnych neuronów. Podsystem kontroli plastyczności mógłby opierać się na niespecyficznych interakcjach neuronalnych, ale prawdopodobnie jest to efekt kontroli chemicznej. Tylko dzięki rozdzielaniu przekazywania sygnału neuronowego i kontroli plastyczności stało się możliwe wdrożenie skutecznego i niezawodnego systemu samoorganizującego się.

SOM jest algorytmem służącym do wizualizacji i interpretacji dużych zestawów danych wysokomiarowych. Typowe aplikacje to wizualizacja stanów procesu lub wyników finansowych poprzez przedstawienie centralnych zależności w obrębie danych na mapie.

Mapa składa się z regularnej siatki jednostek przetwarzania "neuronów". Model jakiejś wielowymiarowej obserwacji to wynikowy wektor składający się z funkcji, który jest powiązany z każdą jednostką. Mapa stara się reprezentować wszystkie dostępne obserwacje z optymalną dokładnością przy użyciu ograniczonego zestawu modeli. W tym samym czasie modele stają się uporządkowane na siatce, tak że podobne modele są blisko siebie, a modele odmienne od siebie oddalone.

Niemniej jednak, zasada SOM może być również wyrażona matematycznie w czystej abstrakcyjnej formie, bez odniesienia do jakiegokolwiek leżącego u podstaw neuro-

nowego lub innych składników.

## 2. Matematyczne określenie algorytmu Kohena

Rozważmy pierwsze elementy danych, które są wektorami w  $n$ -wymiarowej przestrzeni euklidesowej.

$$x(t) = [\xi_1(t), \xi_2(t), \dots, \xi_n(t)]$$

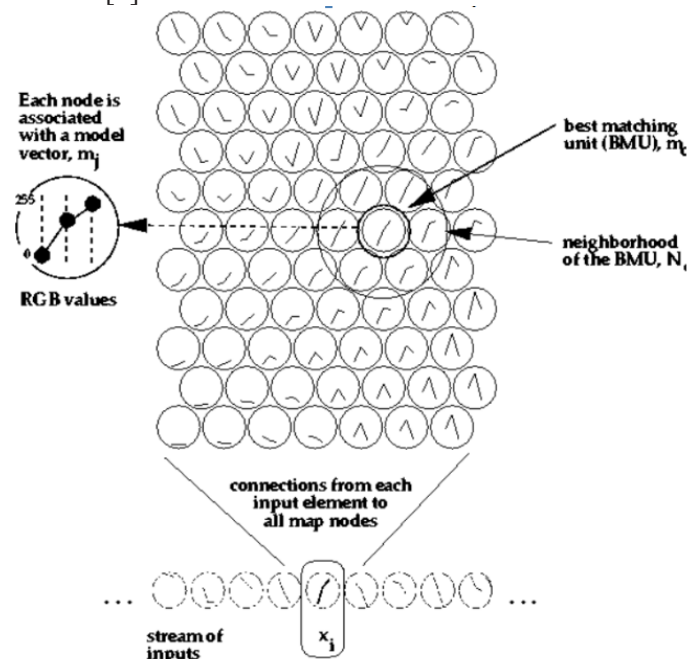
Gdzie  $t$  jest indeksem pozycji danych w danej sekwencji. Niech tym modelem będzie  $i$  :

$$m_i(t) = [\mu_{i1}(t), \mu_{i2}(t), \dots, \mu_{in}(t)]$$

gdzie teraz  $t$  oznacza indeks w sekwencji, w której generowane są modele. Ta sekwencja jest zdefiniowana, jako proces typu wygładzania, w którym nowa wartość  $m_i(t+1)$  jest obliczana iteracyjnie ze starej wartości  $m_i(t)$  i nowej pozycji danych  $x(t)$ , jako:

$$m_i(t+1) = m_i(t) + \alpha(t)h_{ci}(t)[x(t) - m_i(t)]$$

Gdzie  $\alpha(t)$  jest czynnikiem skalarnym, który definiuje wielkość korekty; jego wartość maleje wraz ze wskaźnikiem kroku  $t$ . Indeks  $i$  odnosi się do modelu w trakcie przetwarzania, a  $c$  jest indeksem modelu, który ma najmniejszą odległość od  $x(t)$  w przestrzeni euklidesowej sygnałowej. Czynniki  $h_{ci}(t)$  to rodzaj wygładzającego jądra, zwany także funkcją sąsiedztwa. Funkcja ta jest równa 1, gdy  $i=c$ , a jego wartość maleje, gdy wzrasta odległość między wzorami  $m_i$  i  $m_c$  na siatce. Ponadto, szerokość przestrzenna jądra na siatce zmniejsza się wraz ze wskaźnikiem kroku  $t$ . Te funkcje wskaźnika kroku, które określają zbieżność, należy wybrać [2] bardzo delikatnie. Również inicjalizacja  $m_i(1)$  jest problemem omówionym dokładnie przez Kohonena w [4]



Ryc. 1 Podstawowa architektura mapy samoorganizującej się (SOM). Wejście  $x$  jest w pełni połączone z tablicą węzłów map, która najczę-

ściej jest dwuwymiarowa. Każdy węzeł mapy, zwizualizowany, jako krąg na siatce służy jako model mi lub do użycia innego terminu, prototypu klasy podobnych wejść. Diagramy liniowe wewnątrz kółek oznaczają trzy wartości RGB. Na przykład węzły w lewym dolnym rogu odpowiadają kolorom, które mają wysokie wartości wszystkich komponentów, tj. czerwony, zielony i niebieski (RGB), a zatem ten narożnik zawiera bardzo ciemne kolory[3]

Na rysunku 1 przedstawiono podstawową ideę w procesie uczenia algorytmu SOM. Dla każdego wektora wejściowego próbki  $x(t)$  zwycięzca i węzły w jego sąsiedztwie zmieniają się bliższe  $x(t)$  w wejściowej przestrzeni danych. Podczas procesu uczenia się poszczególne zmiany mogą być sprzeczne, ale końcowym wynikiem sieciowym w procesie jest to, że uporządkowane wartości dla  $m_i(t)$  pojawią się na tablicy macierzy.

Jeśli liczba dostępnych próbek wejściowych jest ograniczona, próbki muszą być przedstawione w sposób powtarzalny algorytmowi SOM [5].

Chociaż powyższy algorytm został zastosowany z powodzeniem w wielu aplikacjach, okazało się, że schemat nazywany mapą wsadową (*ang. Batch Map*) daje zasadniczo podobne wyniki, ale o rząd wielkości szybsze. Podstawową ideą jest to, że dla każdego węzła  $j$  w siatce średnia  $x_j$  wszystkich elementów wejściowych  $x(t)$  jest utworzona, a  $m_i$  jest najbliższym modelem. Następnie nowe modele są obliczane jako:

$$m_i = \frac{\sum_j n_j h_{ji} \bar{x}_j}{\sum_j n_j h_{ji}}$$

Gdzie  $n_j$  to liczba elementów wejściowych odwzorowanych w węzle  $j$ , a indeks  $j$  działa nad węzłami w sąsiedztwie węzła  $i$ . W przypadku zaktualizowanego  $m_i$ , ten schemat jest powtarzany kilka razy, zawsze przy użyciu tej samej partii elementów danych wejściowych w celu ustalenia zaktualizowanego  $x_j$ .

Matematyczna teoria SOM jest bardzo skomplikowana i tylko przypadek jednowymiarowy został całkowicie przeanalizowany [6]. W przypadku zmodyfikowanej reguły adaptacji, w której dopasowanie jest uśredniane na  $h_{ci}$ , SOM może pochodzić z konkretnej potencjalnej funkcji [7]. Najwyraźniej SOM należy do problemów „źle postawionych” w matematyce.

### 3. Praktyczne porady mające zastosowanie podczas realizacji SOM

Wskazane jest, aby zwracać uwagę przynajmniej na następujące zalecenia, aby uzyskane mapowania były stabilne,

dobrze zorientowane i poprawne topologicznie. [3]

**Forma tablicy:** Sześciokątna siatka węzłów ma być preferowana do kontroli wzrokowej. Krawędzie macierzy powinny być raczej prostokątne niż kwadratowe, ponieważ "elastyczna sieć" utworzona z wektorów modelowych  $m_i$  musi być zorientowana wraz z rozmieszczeniem wejścia  $x$  w przestrzeni sygnałowej. Z drugiej strony, ponieważ  $m_i$  musi być w przybliżeniu równe rozkładowi  $x$ , pożądane jest ponadto, aby szerokość i wysokość tablicy odpowiadały co najmniej głównym wymiarom (na przykład główne elementy składowe) rozkładu  $x$ .

**Skalowanie komponentów wektorowych** jest bardzo subtelnym problemem. Można łatwo zauważyć, że orientacja lub uporządkowana "regresja" wektorów modelu w przestrzeni wejściowej zależy od skalowania komponentów (lub wymiarów) wektorów danych wejściowych. Zwłaszcza, jeśli elementy danych reprezentują zmienne różnego rodzaju i dlatego są podawane w różnych skalach, nie istnieje żadna prosta reguła określająca, jakiego rodzaju zmiana skali jest najlepsza przed wprowadzeniem danych treningowych do algorytmu uczenia. Można próbować heurystycznie uzasadnionych skalowań i sprawdzać, jakość otrzymanych map np. za pomocą mapowania Sammona [8] lub średniego błędu kwantyzacji. Często użyteczną strategią jest normalizacja wszystkich zmiennych wejściowych takich, że np. wszystkie ich wariancje stają się równe.

**Nauka z niewielką liczbą dostępnych próbek treningowych:** W niektórych problemach, np. podczas tworzenia map dla przedmiotów zdefiniowanych przez skończony zestaw atrybutów, nie ma dużego zestawu próbek treningowych. Jednak dla dobrej dokładności statystycznej proces uczenia może wymagać dość dużej liczby kroków szkoleniowych, a jeśli liczba dostępnych próbek jest znacznie mniejsza, próbki mogą być użyte ponownie w treningu. Probki mogą być stosowane cyklicznie lub w losowo permutowanej kolejności lub pobierane losowo z podstawowego zestawu przy użyciu minimalnych zasobów (nazywanego uczeniem się bootstrapu). W praktyce okazało się, że uporządkowane cykliczne nakładanie dostępnych próbek nie jest zauważalnie gorsze od innych, statystycznie rygorystycznych metod pobierania próbek.

**Jakość uczenia się:** różne procesy uczenia się będą używane dla różnych wartości początkowych  $m_i(1)$  i przy stosowaniu różnych sekwencji wektorów treningowych  $x(t)$  i różnych parametrów uczenia się. Oczywiście byłoby pożądane, aby powstała mapa zawsze była najmniej dwuznaczna.

Oczywiste jest, że musi istnieć jakaś optymalna mapa dla tych samych danych wejściowych. Porównując mapy o tej samej strukturze macierzy i definicji  $h_{ci}$ , średnia z  $\|x - m_c\|$  (błąd kwantyzacji) w danych treningowych jest przydatnym wskaźnikiem wydajności. Dlatego można próbować z większą liczbą (powiedzmy kilkadziesiąt) losowych ini-

cializacji  $m_i(l)$  tak, by powstała mapa z najmniejszym błędem kwantyzacji. Zauważmy jednak, że nie miałyby sensu porównywać błędów kwantyzacji dla map o różnych strukturach lub  $h_{ci}$ , gdyż błąd jest minimalny, jeśli  $h_{ci} = \delta_{ci}$  (delta Kroneckera). Dla pojedynczego jądra nie występuje już samoorganizacja, ponieważ algorytm zostanie zredukowany do klasycznej kwantyzacji wektora.

#### 4. Implementacja algorytmu SOM

Analizę danych, tworzenie klastrów i wizualizację przez SOM najlepiej przeprowadzić przy użyciu komercyjnego oprogramowania z kodem pobranym z domeny publicznej. Używanie oprogramowania z własnym kodem nie jest zalecane, ponieważ istnieje wiele subtelnych aspektów, które należy wziąć pod uwagę i które wpływają na zbieżność i dokładność algorytmu. SOM\_PAK to zestaw narzędzi do prawidłowego zastosowania SOM. Ten oryginalny pakiet programowy został stworzony przez zespół programistów SOM z Uniwersytetu Technicznego w Helsinkach, szczególnie w przypadku bardzo dużych problemów<sup>3</sup>. Więcej informacji o pakietach można znaleźć na stronie SOM\_PAK Jussi Hynninen<sup>4</sup>.

Poniżej wyszczególniono cztery szczegółowe zastosowania SOM, które stanowią podstawę do dalszej kontroli następujących działów:

##### 4.1. SOM jest modelem określonych aspektów biologicznych sieci neuronowych

Być może najbardziej typowym pojęciem SOM jest uznanie go za sztuczny model sieci neuronowej mózgu, szczególnie eksperymentalnie odnalezionych "map" w korze mózgowej. Istnieje wiele dowodów neurofizjologicznych na poparcie idei, że SOM przechwytuje niektóre z podstawowych zasad przetwarzania mózgu. Niektóre wcześniejsze sztuczne modele sieci samoorganizujących się oparte na sieciach neuronowych zostały wcześniej zaprezentowane, np. przez Shun-ichi Amari [9] czy Gail Carpenter i Stephen Grossberg [10]; jednak zasada SOM okazała się najbardziej efektywna, by produkować mapy podobnie jak mózg. Związek pomiędzy SOM i jego odpowiednikami w neurodynamice jest opisany szczegółowo, np. w pracach Kohonena [11, 12].

##### 4.2. SOM stanowi reprezentację nowego paradygmatu sztucznej inteligencji i modelowania kognitywnego,

SOM można postrzegać jako maszynowy model uczenia się bez nadzoru oraz jako adaptacyjny schemat reprezentacji wiedzy. W rozdziale 5 rozważany jest związek między SOM (zwłaszcza mapą kategorii słów) a sieciami semantycznymi. Tradycyjne formalizacje reprezentacji wiedzy - sieci semantyczne, systemy ramowe, logika orzecznicza stosowane jako pewne przykłady, są statyczne, a relacje referencyjne elementów są określane przez człowieka. Po-

nadto formalizmy te opierają się na milczącym założeniu, że relacja między językiem naturalnym a światem jest jeden do jednego: świat składa się z obiektów i relacji między obiektami, a te obiekty i związki mają bezpośrednią korespondencję z elementami języka. W odniesieniu do reprezentacji wiedzy i uczenia się, aspekty poznawcze i filozoficzne są bardzo istotne. Zagadnienia te zostaną omówione bardziej szczegółowo w rozdziale

##### 4.3. SOM jest narzędziem do analizy statystycznej i wizualizacji

SOM jest obecnie często używane jako narzędzie statystyczne do analizy wielowymiarowej. SOM jest zarówno metodą rzutowania, która odwzorowuje przestrzenną przestrzeń danych o dużej przestrzeni na przestrzeń niskiego wymiaru, jak i metodą grupowania, aby podobne próbki danych były odwzorowywane na pobliskie neurony. Z metodologicznego i obliczeniowego punktu widzenia można rozważyć matematyczne i statystyczne właściwości algorytmu (na przykład złożoność czasu i przestrzeni, właściwości konwergencji), a także charakter danych wejściowych (sygnały, stałe wskaźniki statystyczne, ciągi symboli) i ich wstępne przetwarzanie. Istnieje wiele wariantów SOM, w których, na przykład, reguły adaptacji są różne, można zastosować różne miary odległości, a struktura połączeń międzysystemowych jest zmienna. SOM jest narzędziem do opracowywania złożonych aplikacji [3].

##### 4.4. SOM jako narzędzie do opracowywania złożonych aplikacji

SOM jest szeroko stosowana jako metoda eksploracji danych i wizualizacji złożonych zbiorów danych. Obszary zastosowania obejmują na przykład przetwarzanie obrazu i rozpoznawanie mowy, sterowanie procesem, analizę ekonomiczną i diagnostykę w przemyśle i medycynie. Podsumowanie zastosowań inżynierii SOM podano w [5].

Niektóre aplikacje wymagają wydajnej budowy dużych map. Wyszukiwanie najlepiej pasującej jednostki jest zwykle najtrudniejszą operacją obliczeniową w SOM. Korzystając z SOM o strukturze drzewa, możliwe jest użycie wyszukiwania hierarchicznego dla najlepszego dopasowania [13]. W tej metodzie chodzi o zbudowanie hierarchii SOM i uczenie SOM na każdym poziomie przed przejściem do następnej warstwy. Kolejna metoda przyspieszenia, dzięki której zwycięzca może szybciej wyszukiwać, w oparciu o ideę Koikkalainena, jest przedstawiona w [4].

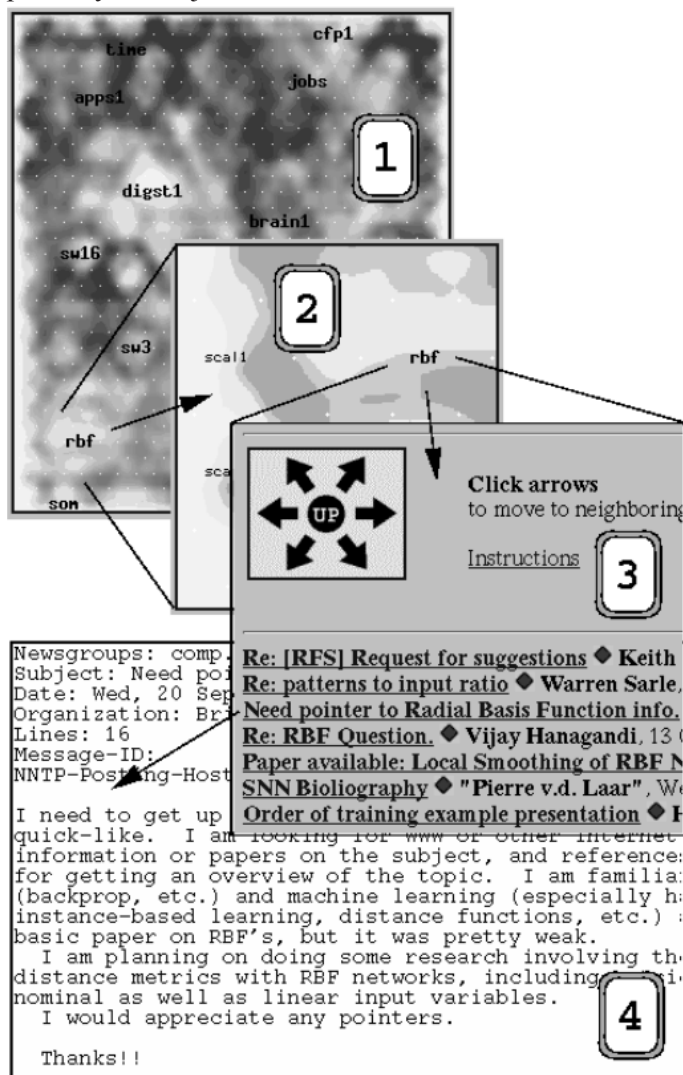
Większość aplikacji SOM używa danych liczbowych. Można również wykazać, że cechy statystyczne tekstu naturalnego także da się uznać za cechy numeryczne, które ułatwiają zastosowanie SOM w zadaniach NLP (programowanie neurolingwistyczne). Algorytm SOM znajduje zastosowanie w pojazdach autonomicznych [14].

3. Pakiet jest dostępny do pobrania pod adresem [http://www.cis.hut.fi/research/som\\_lvq\\_pak.shtml](http://www.cis.hut.fi/research/som_lvq_pak.shtml)

4. Strona dostępna pod adresem <http://www.cis.hut.fi/~hynde/lvq/>

## 5. Obliczenia i analiza wyników przy zastosowaniu metody WEBSOM

Za pomocą metody WEBSOM można organizować tekstowy zbiór dokumentów na graficznym wyświetlaczu mapy, który zapewnia przegląd kolekcji i ułatwia interaktywne przeglądanie. Interesujące dokumenty można znaleźć na mapie za pomocą wyszukiwania ukierunkowanego na treść. Każdy dokument jest zakodowany jako histogram kategorii słów, które są tworzone przez organizowanie algorytmu mapy (SOM) w oparciu o podobieństwa w kontekście słów. Zakodowane dokumenty są zorganizowane na innej samoorganizującej się mapie, mapie dokumentów, na której pobliskie lokalizacje zawierają podobne dokumenty. Szczególną uwagę poświęca się obliczaniu bardzo dużych map dokumentów, które są możliwe do obliczeń przez komputery ogólnego przeznaczenia, jeśli wymiarowość histogramów kategorii słów jest najpierw zredukowana za pomocą metody losowego mapowania i jeśli w obliczeniach SOM stosuje się wydajne obliczeniowo algorytmy [15]. Na rysunku 2 przedstawiono podstawowe poziomy interfejsu WEBSOM.



Ryc. 2. Podstawowy poziomy interfejsu WEBSOM

Przedstawiono na nim: (1) całą mapę, (2) powiększoną mapę, (3) węzeł mapy i (4) widok dokumentu, pokazany w porządku rosnącej szczegółowości. Przenoszenie międ-

dzy poziomami lub do sąsiednich obszarów na tym samym poziomie odbywa się za pomocą kliknięć myszką na obrazkach lub na łączach dokumentów. Po znalezieniu interesującego obszaru na mapie eksploracja powiązanych dokumentów w sąsiednich obszarach jest prosta.

W WEBSOM każdy dokument jest reprezentowany na mapie dokumentu jako punkt w taki sposób, że wzajemna odległość między dwoma dowolnymi punktami reprezentacji odzwierciedla podobieństwo odpowiednich dwóch histogramów. Dlatego podobne dokumenty są mapowane blisko siebie na mapie dokumentu, podobnie jak książki na półkach dobrze zorganizowanej biblioteki. Metoda WEBSOM jest z łatwością stosowana do wszelkiego rodzaju gromadzenia dokumentów tekstowych, nawet jeśli nie dostarczono im słów kluczowych. Zorganizowano kolekcje aż 100 000 dokumentów na mapach o liczbie rzędu 10 000 węzłów.

Metoda jest szczególnie odpowiednia do zadań związanych z eksploracją, w których użytkownicy albo nie znają domeny bardzo dobrze, albo mają tylko ograniczone wyobrażenie o zawartości analizowanej bazy pełno-tekstowej. Dzięki WEBSOM dokumenty są porządkowane zgodnie z ich treścią. Mapy pomagają w eksploracji, dając ogólną wizualizację tego, jak wygląda przestrzeń informacyjna. [15]

## 6. Podsumowanie

Liczba potencjalnych zastosowań algorytmu Kohonena w przyszłych kierunkach jest duża, począwszy od badań nad implikacjami w lingwistyce i filozofii języka. Z drugiej strony, rozwój aplikacji uzasadnia dalsze rozwijanie metody SOM. Interesujące byłoby zbadanie związku między modelowaniem interpretacji języka naturalnego za pomocą SOM a niektórymi pokrewnymi teoriami. Interesujące punkty widzenia można znaleźć w językoznawstwie ogólnym, językoznawstwie kognitywnym, kognitywistyce i filozofii języka. Specyficzne powiązane teorie obejmują np. teorię prototypów, gramatykę kognitywną, teoria relewancji i konstruktywizm. Wykorzystanie SOM w przetwarzaniu w języku naturalnym można porównać z szerokim spektrum innych podejść statystycznych i opartych na teorii ciał. Istnieje kilka alternatyw kodowania informacji kontekstowych w tekstach dla mapy kategorii słów. Można oczywiście opracować inne sposoby gromadzenia informacji kontekstowych. Mogą one być oparte na przykład na kojarzeniu słów z wprowadzaniem innych modalności. Można opracować nowe aplikacje w takich obszarach jak tłumaczenie maszynowe i systemy współpracy.

W tłumaczeniu maszynowym można zastosować mapy kategorii słów, np. w selekcji leksykalnej, tj. znalezienie odpowiedniego słowa lub frazy do zastosowania w kontekście. Materiały tekstowe w językach innych niż angielski mogą być również przetwarzane. W wymienionych wyżej pracach różne formy słowne leksemów były kodowane oddzielnie. Takie podejście wydawało się działać dobrze, gdy rozważany jest angielski.

Jeśli podstawowa forma słowa może wystąpić w ponad 10 000 różnych fleksyjnych form wyrazów (takich jak czasowniki w języku fińskim), nowe problemy muszą zostać rozwiązane w inny sposób.

W metodzie WEBSOM związek pomiędzy charakterystyką danych wejściowych, wyborami wartości parametrów i możliwymi narzędziami do przetwarzania wstępnego wymaga dogłębnych badań. Jako konkretny przykład można rozważyć wpływ długości dokumentu na proces WEBSOM. W niektórych przypadkach długi dokument może być podzielony na krótsze części, które prawdopodobnie mają jaśniejszą tożsamość. W tradycyjnym wyszukiwaniu słów kluczowych, które wykorzystuje algebrę Boole'a do łączenia słów kluczowych, zaletą jest możliwość łączenia terminów na różne sposoby tak, aby użytkownik kontrolował proces. Podobną funkcjonalność można osiągnąć za pomocą metody WEBSOM, umożliwiając użytkownikowi łączenie wyników kolejnych wyszukiwań na mapie dokumentu. Przydatne byłoby również zbadanie zadowolenia użytkowników z oceny metody WEBSOM.

### Literatura:

- [1] M. M. Nassand L. N. Cooper, "A theory for the development of feature detecting cells in visual cortex," *Biol. Cybern.*, vol. 19, pp. 1-18, 1975
- [2] S. Athuraliya, S.H. Low, V.H. Li and Q. Yin, "REM: Active queue management," *IEEE Network Mag.*, vol. 15, no. 3, pp. 48-53, 2001
- [3] T. Kohonen, *Self-Organizing Maps*, Springer-Verlag, Berlin, Heidelberg, New York, 1995, XVI+362pp.
- [4] Kohonen, T. (1982). Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 43:59-69.
- [5] Kohonen, T., Kaski, S. and Lappalainen, H. (1997). Self-organized formation of various invariant-feature filters in the adaptive-subspace SOM. *Neural Computation*, 9: 1321-1344.
- [6] Fort, J.C. (2006). SOM's mathematics. *Neural Networks*, 19: 812-816.
- [7] Heskes, T. (2001). Self-organizing maps, vector quantization, and mixture modeling. *IEEE Transactions on Neural Networks*, 12:1299-1305.
- [8] Sammon, J. (1969). A nonlinear mapping for data structure analysis. *IEEE Trans. Comp.*, 18(5):401-409.
- [9] Amari, S.-I. (1967). A theory of adaptive pattern classifiers. *IEEE Trans. Comp.*, 16:299-307
- [10] Carpenter, G. and Grossberg, S. (1987). A massively parallel architecture for a self-organizing neural pattern recognition machine. *Computer Vision, Graphics, and Image Processing*, 37:54-115.
- [11]. Kohonen, T. (1992). An attempt to interpret the Self-Organizing Mapping physiologically. Report A16, Helsinki University of Technology, Laboratory of Computer and Information Science.
- [12] Kohonen, T. (1993). Physiological interpretation of the self-organizing map algorithm. *Neural Networks*,

6(7):895-905.

[13] Koikkalainen, P. (1994). Progress with the tree-structured self-organizing map. In Cohn, A. G., editor, *Proceedings of ECAI'94, 11th European Conference on Artificial Intelligence*, pages 211-215, New York. John Wiley & Sons.

[14] Autonomiczne pojazdy P Kardasz, O Lyubov, E Kardasz, *Biuletyn Naukowy Wrocławskiej Wyższej Szkoły Informatyki Stosowanej* 2017

[15] Krista Lagus, Timo Honkela, Samuel Kaski, and Teuvo Kohonen *WEBSOM - A Status Report*, Publications of the Finnish Artificial Intelligence Society, pp. 73-78. 1998