Sztuczne Sieci Neuronowe

Wykład 5

Sieci *“Counter-Propagation”* (CP)
Specificzne architektury

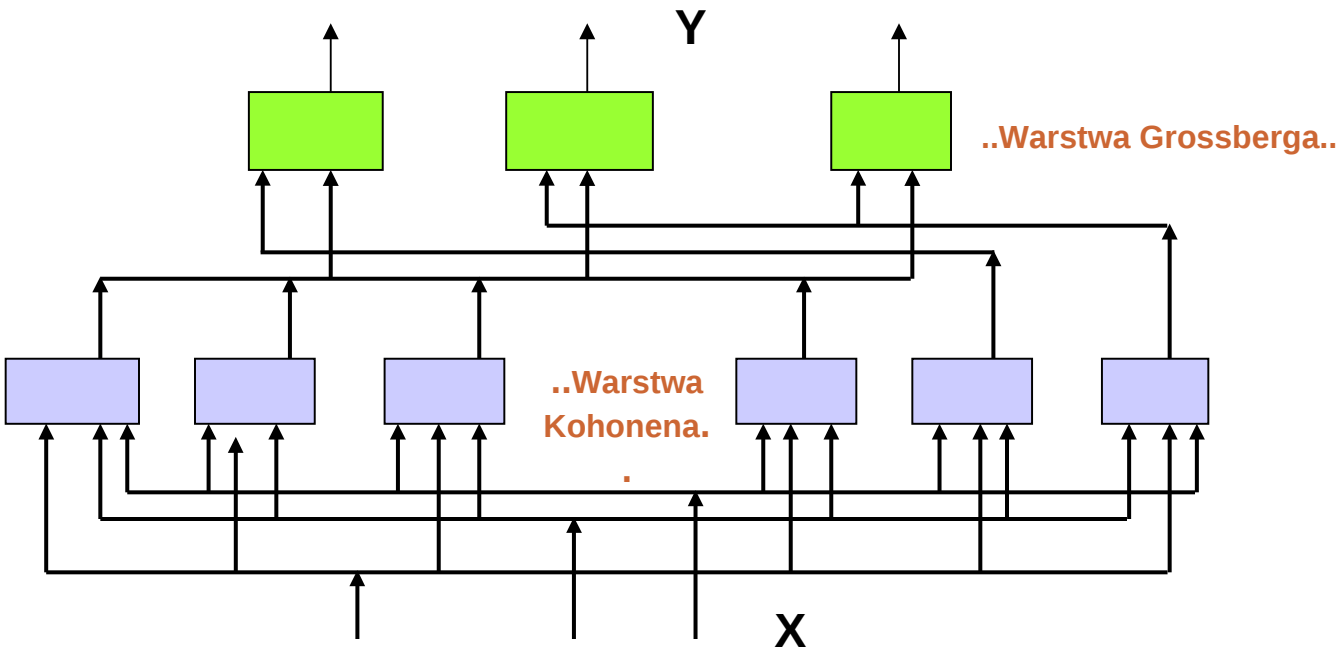
Wybrane zastosowania

wykład przygotowany na podstawie.

S. Osowski, „Sieci neuronowe w ujęciu algorytmicznym”, WNT, 1996
J. Hertz, A. Krogh, R.G.Palmer, „Wstęp do teorii obliczeń neuronowych”, WNT, 1993
R. Tadeusiewicz, “Sieci Neuronowe”, Rozdz. 5. Akademicka Oficyna Wydawnicza RM, Warszawa 1993.

Sieci CP (Counter Propagation)

Sieć **CP** właściwie nie jest oryginalną propozycją, lecz stanowi **kompilację sieci Kohonena i sieci Grossberga**. Zestawienie tych sieci w strukturze sieci **CP** wprowadziło istotnie nową jakość – sieć stosunkowo szybko się uczącą i mającą (potencjalnie) nieograniczony zakres możliwych odwzorowań pomiędzy sygnałem wejściowym **X** i wyjściowym **Y**.



Przypomnienie: metoda gwiazdy wyjsc

W koncepcji tej rozważa się wagi wszystkich neuronów całej warstwy, jednak wybiera się wyłącznie wagi łączące te neurony z ***pewnym ustalonym wejściem***.
W sieciach wielowarstwowych wejście to pochodzi od pewnego ustalonego neuronu wcześniejszej warstwy i to właśnie ten neuron staje się ***"gwiazdą wyjść"*** (outstar).

$$\omega_i^{(m)(j+1)} = \omega_i^{(m)(j)} + \eta^{(j)} \left(y_m^{(j)} - \omega_i^{(m)(j)} \right)$$

W powyższym wzorze **i** jest ustalone, natomiast **m** (wyjścia) jest zmienne i przebiega wszelkie możliwe wartości (m = 1,2,.....,k).
Reguła zmieniania $\eta^{(j)}$ jest dana wzorem

$$\eta^{(j)} = 1 - \lambda\ j$$

Przypomnienie: uczenie Kohonena

Uczenie z rywalizacja (competitive learning) wprowadził Kohonen przy tworzeniu sieci neuronowych uczących się realizacji ***dowolnych odwzorowań*** $X \Rightarrow Y$.

Zasada uczenia z rywalizacją jest formalnie identyczna z regułą "instar"

$$\omega_i^{(m^*)(j+1)} = \omega_i^{(m^*)(j)} + \eta^{(j)} \left(x_i^{(j)} - \omega_i^{(m^*)(j)} \right)$$

z dwoma dość istotnymi uzupełnieniami.

$\Rightarrow$ Wektor wejściowy X jest przed procesem uczenia normalizowany tak, aby $\|X\| = 1$.

$\Rightarrow$ Numer **podawanego treningowi neuronu m*** nie jest przypadkowy czy arbitralnie wybierany, jest to bowiem ten (i tylko ten) neuron którego **sygnał wyjściowy $y_{m^*}^{(j)}$ jest największy**. Przy każdorazowym podaniu sygnału wejściowego $X^{(j)}$ neurony rywalizują ze sobą i wygrywa ten, który uzyskał największy sygnał wyjściowy $y_{m^*}^{(j)}$. Tylko ten **zwycięski neuron podlega uczeniu**, którego efektem jest jeszcze lepsze dopasowanie wag $W^{(m^*)(j+1)}$ do rozpoznawania obiektów podobnych do $X^{(j)}$.

Przypomnienie: uczenie Kohonena

Reguła uczenia Kohonena bywa często wzbogacana o dodatkowy element związany z topologią uczącej się sieci. Neurony w sieci są ***uporządkowane***, można więc wprowadzić pojęcie ***sąsiedztwa***.
Uogólniona metoda samoorganizującej się sieci Kohonena polega na tym, że uczeniu podlega nie tylko neuron **m*** wygrywający w konkurencji z innymi neuronami sieci, ale także neurony które z nim sąsiadują.

Formalnie regułę można zapisać wzorem:

$$\omega_i^{(m)(j+1)} = \omega_i^{(m)(j)} + \eta^{(j)}\, h(m,m^*)\, (x_i^{(j)} - \omega_i^{(m^*)(j)})$$

formuła uczenia może być zapisana w formie:

$$\omega_i^{(m)(j+1)} = \omega_i^{(m)(j)} + \eta^{(j)}\, h(m,m^*)\, x_i^{(j)} (2\, y_{m^*}^{(j)} - 1)$$

Przypomnienie: uczenie Kohonena

Funkcjonowanie powyższego wzoru w istotny sposób oparto na fakcie, że $y_m^{(j)} \in \{0,1\}$.

Wzór ten nazywamy ***regułą Hebb/Anti-Hebb***.

Funkcje **h(m,m*)** można definiować na wiele różnych sposobów, na przykład:

$$h(m,m^*) = \begin{cases} 1 & \text{dla } m=m^* \\ 0.5 & \text{dla } |m-m^*| =1 \\ 0 & \text{dla } |m-m^*| > 1 \end{cases}$$

$$h(m,m^*) = 1/\rho(m,m*)$$

$$h(m,m^*) = \exp\left(- [\rho(m,m*)]^2 \right)$$

Sieci CP

Pierwsza warstwa sieci CP jest warstwą realizującą algorytm Kohonena. Dokonuje klasyfikacji wektora wejsciowego. Druga warstwa zwraca wektor skojarzony z dana klasa wejsciowa. Jest to dzialanie tablicy „look-up”

Podstawowym założeniem przy stosowaniu sieci **CP** jest normalizacja sygnałów wejściowych. Każdy wektor **X** wprowadzany do systemu musi spełniać warunek **||X|| = 1** (czyli $X^TX = 1$). Normalizacja **X** jest dokonywana poza siecią neuronową.

Normalizacja wejść jest potrzebna ze względu na element konkurencji (rywalizacji) występujący w pierwszej warstwie sieci **CP**. Do dalszego przetwarzania w kolejnej warstwie sieci przesyłany jest zaledwie jeden sygnał, pochodzący od tego elementu warstwy pierwszej który był najbardziej optymalnie dopasowany do przedstawionego sygnału wejściowego **X**.

Zasada pierwszej warstwy sieci CP

Pierwsza warstwa sieci CP jest warstwą realizującą algorytm Kohonena.

Wektory wejściowe **X** mnożone są przez wektory wag W_j poszczególnych neuronów sieci dostarczając wartości e_j będących sumarycznym (ważonym) pobudzeniem każdego neuronu.

$$e_j = W_j^T X$$

a następnie wybierany jest element o największej wartości pobudzenia e_j ***("zwycięzca")*** i tylko jego sygnał wyjściowy przyjmuje wartość **1**. To jest właśnie ten tytułowy ***counter*** zastępujący i symbolizujący wszystkie sygnały wejściowe.

Zasada drugiej warstwy sieci CP

Druga warstwa sieci realizuje algorytm *Outstar* Grossberga.

Jeżeli oznaczymy, że sygnały wejściowe do tej warstwy tworzą wektor K, a sygnał wyjściowy Y obliczany jest wg. klasycznej reguły Y = V K, gdzie macierz współczynników wagowych V składa się z transponowanych wektorów V_i odpowiadających zestawom wag kolejnych neuronów warstwy wyjściowej.
Z formalnego punkty widzenia, sygnał z neuronów warstwy wyjściowej ma postać

$$y_i = \sum_{j=1}^{m} v_{ij}\, k_j$$

gdzie m ma z reguły dużą wartość. W praktyce tylko jeden element wektora **K** ma wartość **1**, pozostałe maja wartość **0** i wystarcza utożsamić wyjścia y_j z pewnym współczynnikiem v_{ij}. Na wszystkich wyjściach pojawiają się tylko te wartości v_{ij} które odpowiadają numerowi j dla których $k_j=1$.
Zauważmy, że działanie to przypomina odczyt gotowej tabeli.

Uczenie sieci CP

Uczenie sieci CP przebiega równocześnie w obu warstwach sieci. Jest to proces ***uczenia z nauczycielem***, wraz z każdym wektorem wejściowym X podany jest wektor wyjściowy, jaki użytkownik chce uzyskać z sieci.

⇒ **przy uczeniu pierwszej warstwy stosuje się technikę Kohonena, która jest formą uczenia bez nauczyciela**

⇒ **przy uczeniu drugiej warstwy stosuje się algorytm Grossberga do bezpośredniego wymuszania pożądanych odpowiedzi sieci.**

Uczenie pierwszej warstwy sieci CP

Na **k-tym kroku** pokazuje się wektor wejściowy $X^{(k)}$, a dysponując (z wcześniejszych kroków procesu uczenia) wartościami wszystkich wektorów $W_j^{(k)}$ można obliczyć wszystkie wartości $e_j^{(k)}$

$$e_j^{(k)} = W_j^{(k)T} X^{(k)} \quad , j=1,2,...,m$$

oraz wyznaczyć **numer "zwycięskiego" neuronu, z,** (tzn. tego, dla którego zachodzi)

$$\forall\ (j \neq z) \qquad e_z^{(k)} > e_j^{(k)}$$

Korekcie podlegają wyłącznie wagi "zwycięskiego"neuronu według reguły

$$W_z^{(k+1)} = W_z^{(k)} + \eta_1 (X^{(k)} - W_z^{(k)})$$

współczynnik uczenia η_1 jest przyjmowany początkowo jako **0.7**, potem stopniowy zmniejszany.

Uczenie pierwszej warstwy sieci CP

Przy realizacji metody Kohonena najważniejsze są pierwsze kroki.

Najpierw należy nadać współczynnikom wagowym w_{ij} wartości początkowe.

Należy zapewnić unormowanie wszystkich wag $||W_j^{(1)}|| = 1$ oraz wskazane jest takie dobranie kierunków, by w sposób równomierny rozkładały się na powierzchni jednostkowej w przestrzeni n-wymiarowej.

To nie jest takie proste w realizacji.

Uczenie pierwszej warstwy sieci CP

Technika "*convex combination method*".

Początkowo, wszystkim składowym wszystkich wektorów wag nadaje się te sama wartość początkowa

$$\omega_{ij}^{(1)} = \text{sqrt}(1/n)$$

W procesie uczenia jako wektory wejściowe podajemy

$$x_i^{(k)\prime} = \eta_2(k)\, x_i^{(k)} + [1 - \eta_2(k)]\, \text{sqrt}(1/n)$$

gdzie

$\eta_2(k)$ – funkcja adaptująca, która dla małych k przyjmuje małe wartości a potem stopniowo rośnie do wartości 1 i tą wartość zachowuje podczas całego procesu uczenia.

Uczenie drugiej warstwy sieci CP

Uczenie drugiej warstwy sieci jest wykonywane wg. następującej reguły:

$$v_{ij}^{(k+1)} = v_{ij}^{(k)} + \eta_3 (y_i - z_i) k_j$$

Ponieważ **tylko jedna wartość k_j jest różna od zera** i w każdym kroku procesu uczenia korygowane są tylko te wagi, które łączą poszczególne neurony wyjściowej warstwy z jednym tylko – "zwycięskim" elementem poprzedniej warstwy.

Ta zasada (zwana regułą "outstar") znacznie zmniejsza pracochłonność procesu uczenia.

Parametr η_3 wybiera się **"ostrożnie"** tak, aby proces uczenia nie spowodował wpisania do **"look-up"** tablicy błędnych wartości.

Autoasocjacyjna siec CP

Wersja autoasocjacyjna oznacza, że sieć nauczona realizacji odwzorowania

$$X \Rightarrow Y$$

może również realizować odwzorowanie

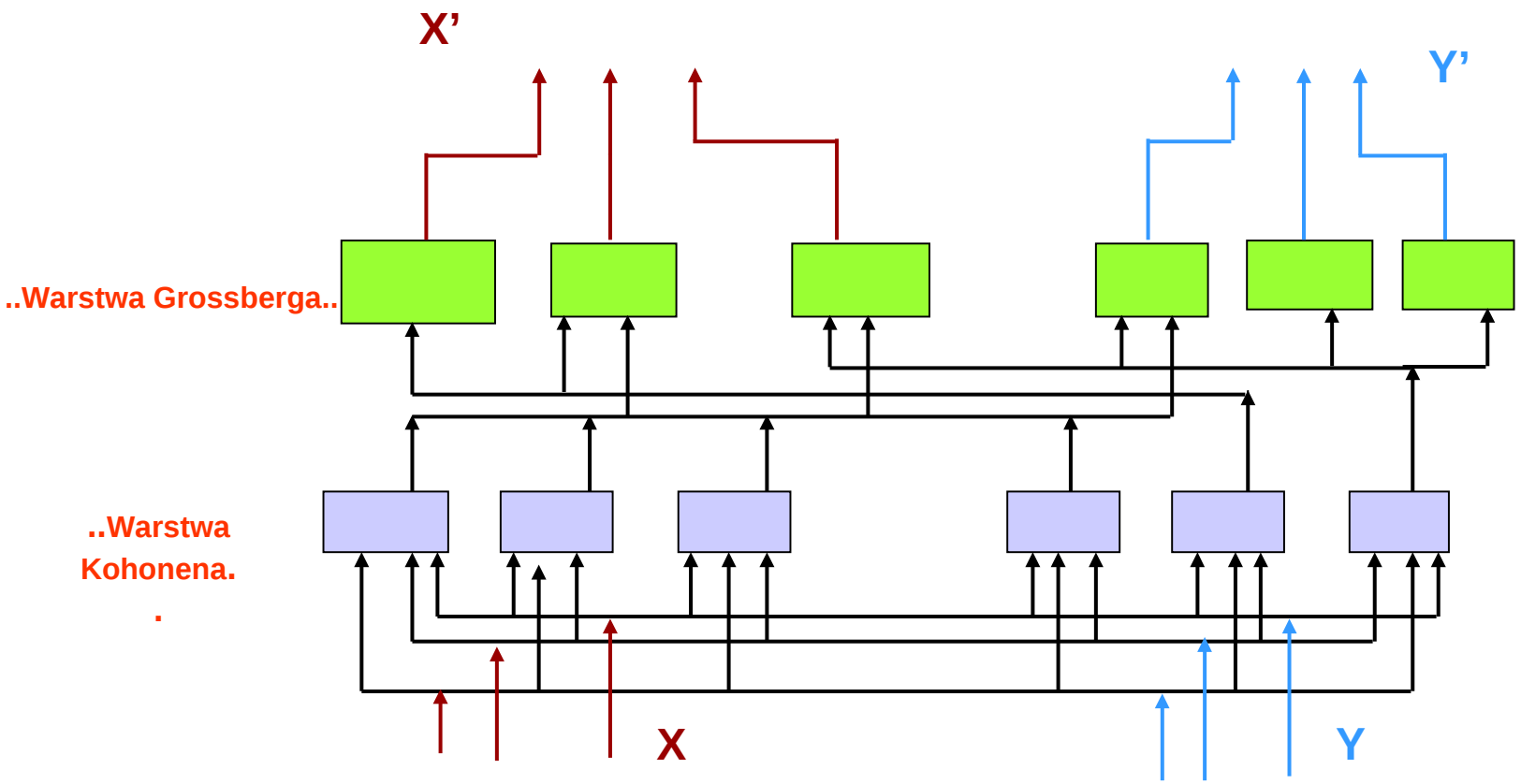
$$Y \Rightarrow X$$

Uczenie sieci polega na tym że na wejście podaje się **X, Y** (jako sygnały wejściowe) na wyjściu oczekuje się również **X,Y**.

Sieć uczy się realizacji odwzorowania tożsamościowego.

Eksploatuje się sieć podając tylko sygnał $X^{(k)}$ (wejścia **Y** bez sygnału), na wyjściu otrzymuje się odtworzony $X^{(k)}$ oraz również $Y^{(k)}$ który na etapie uczenia był kojarzony z $X^{(k)}$.

Autoasocjacyjna siec CP



Autoasocjacyjna sieć CP

Ponieważ na etapie uczenia sygnały **X, Y** są całkowicie równoprawne, sieć potrafi także odtwarzać odwzorowanie odwrotne. Wystarczy na wejściu podać sygnał $\mathbf{Y^{(k)}}$, pozostawiając wejścia $\mathbf{X^{(k)}}$ bez sygnału, na wyjściu otrzymamy $\mathbf{X^{(k)}}$ oraz oczywiście odtworzony przez sieć $\mathbf{Y^{(k)}}$.

Sieci CP

Sieć CP ***“potrafi uogólniać i kojarzyć dostarczone jej informacje”.***

W rozbudowanej wersji jest ona dość chętnie i z powodzeniem stosowana.

Doskonale zdają egzamin jako:

- systemy **klasyfikacji i rozpoznawania obrazów**
- są wykorzystywane w **automatyce i robotyce**
- są cenione jako **systemy do redukcji ilości przesyłanych informacji** (transmisji obrazów)
- rozpoznawanie mowy
- aproksymacja funkcji

Rozpoznawanie wzorców

Przez pojęcie ***rozpoznawania wzorców*** rozumiemy identyfikację lub interpretację wzorca traktowanego jako obraz. Zadaniem sieci jest **wyłowienie jego najważniejszych cech** i **zakwalifikowanie do odpowiedniej kategorii** (klasy).

Można wyróżnić dwa rodzaje podejść:

→ Najpierw następuje wydobycie najważniejszych cech obrazu, a następnie sieć dokonuje na ich podstawie klasyfikacji. W wydobyciu cech obrazu są stosowane różne metody (np. momentów statystycznych)

→ Wydobywanie cech obrazu i klasyfikacja są połączone w jedno zadanie rozwiązywane przez tą samą sieć neuronową. Np. przekształcenia obrazów typu statystycznego, stanowiące fragment działania sieci neuronowej.

Rozpoznawanie wzorców

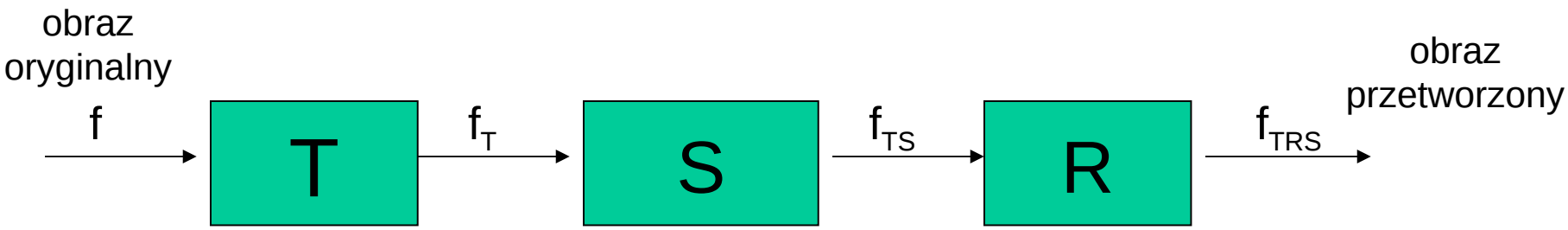
Proste podejście łączące cechy obu metod.

→ **Dane dotyczące obrazu są przetwarzane przez procesor uniezależniający obraz od przesunięcia, rotacji i skalowania.**

→ **Wynik jest podawany na sieć neuronową dokonującą właściwego rozpoznania.**

Główną cecha preprocesora musi być stabilność przekształcenia obrazu niezależna od poziomu szumów w obrazie oryginalnym oraz prosty i szybki w działaniu algorytm przekształcenia umożliwiający jej przeprowadzenie w czasie porównywalnym z czasem działania samego klasyfikatora neuronowego.
Jednym z takich rozwiązań jest ***preprocesor o strukturze kaskadowej***.

Preprocesor o strukturze kaskadowej



Preprocesor składa się z trzech bloków:

- **typu T :** uniezależniający od przesunięcia wzdłuż osi x i y
- **typu S :** skalowanie
- **typu R :** rotacja

Obraz oryginalny oraz przetworzony zakodowane są w postaci pixeli.

Blok przesunięcia

Blok przesunięcia T zapewnia niezmienność względem przesunięcia na osi x i y przez określenie położenia środka ciężkości wzorca i takie jego przesunięcie, że znajdzie się ono zawsze w początku układu współrzędnych, umieszczanym zwykle w punkcie centralnym ramy obrazu. Środek ciężkości jest obliczany metodą uśredniania współrzędnych x i y wzorca, P – liczba pixeli o przypisanej wartości binarnej 1

$$P = \Sigma \Sigma \; f(x_i, y_j)$$

przy czym N oznacza wymiar pixelowy ramy obrazu (przyjmuje się ramę kwadratowa), a $f(x_i, y_j)$ ma wartość binarna 0 lub 1, określającą jasność przypisana pikselowi o współrzędnych (x_i, y_j). Środek ciężkości oblicza się z zależności:

$$x_m = 1/P \; \Sigma \Sigma \; x_i \, f(x_i, y_j) \; ; \qquad y_m = 1/P \; \Sigma \Sigma \; y_i \, f(x_i, y_j)$$

Wzorzec wyjściowy z bloku przesunięcia określa funkcja

$$f_T(x_i, y_j) = f_T(x_i + x_m \, , \, y_j + y_m \,)$$

która zmienia położenie wzorca oryginalnego na płaszczyźnie umieszczając go w miejscu środka ciężkości.

Blok skalujący

Blok skalujący S to taka zmiana wymiarów wzorca, aby **średnia odległość miedzy początkiem układu współrzędnych a pixelami znajdującymi się w stanie wysokim była określonym ułamkiem wymiaru ramy**.

Średnia odległość określa wzór:

$$r_m = 1/(\Sigma \Sigma f_T(x_i, y_j)) \; \Sigma \Sigma f_T(x_i, y_j) \, sqrt(x_i^2 + y_j^2)$$

a współczynnik skali

$$S = r_m/R$$

przy czym R jest określonym ułamkiem wymiaru ramy.

Wzorzec wyjściowy z bloku skalowania określa funkcja

$$f_{TS}(x_i, y_j) = f_T(Sx_i, Sy_j)$$

Tego typu skalowanie zapewnia ciągłość cech charakterystycznych wzorca (przy ciągłym wzorcu wejściowym f_T wzorzec wyjściowy f_{TS} jest również ciągły.

Blok rotacji

Blok rotacji R dokonuje obrotu wzorca w taki sposób, aby **kierunek maksymalnej wariancji pokrywał się z osią x**.
Przekształcenie to wykorzystuje własność systemu, że dla danego zbioru wektorów wejściowych wektor własny stowarzyszony z największą wartością własną macierzy kowariancji wektorów wejściowych jest skierowany w kierunku maksymalnej wariancji.
Biorąc pod uwagę jedynie obrazy dwuwymiarowe, macierz kowariancji ma wymiar 2x2, dla którego wektor własny stowarzyszony z największą wartością własną, może być określony w sposób analityczny. Można doprowadzić do funkcji rzutujących postaci:

$$f_{TSR}(x_i, y_j) = f_{TS}\,(x_i \cos(\Theta) - y_j \sin(\Theta)\,,\; x_i \sin(\Theta) + y_j(\cos(\Theta))$$

gdzie **sin(Θ), cos(Θ)** odpowiadają nachyleniu wektora własnego.

Układ klasyfikatora neuronowego

→ **Sygnały wyjściowe f_{TSR} preprocesora** uporządkowane w postaci wektorowej składającej się z kolejnych wierszy tabeli pikselowej, stanowią **sygnały wejściowe sieci neuronowej** wielowarstwowej, pełniącej funkcje klasyfikatora.

→ Liczba węzłów wejściowych sieci jest równa liczbie pikseli.

→ Każdy neuron wyjściowy reprezentuje klasę, ich liczba jest również stała i równa liczbie klas.

→ Liczba warstw ukrytych i neuronów w warstwie podlega doborowi.

Klasyfikator jest trenowany metodą propagacji wstecznej przy użyciu jednego z algorytmów uczących na zbiorze danych uczących reprezentujących kolejne klasy wzorców podlegających rozpoznaniu. **Biorąc pod uwagę istnienie preprocesora, wystarczy użycie jednego wzorca wejściowego dla każdej klasy.**

Uklad interpretera

Na etapie rozpoznawania wzorców, biorąc pod uwagę **ich zaszumienie**, sygnały wyjściowe neuronów mogą przyjmować wartości ciągle z przedziału [0,1] zamiast spodziewanych wartości binarnych zero-jedynkowych z jedynką odpowiadającą rozpoznanej klasie.
→ Jednym z rozwiązań jest przyjęcie **neuronu najbardziej aktywnego** jako reprezentanta danej klasy.
→ Najlepszym rozwiązaniem wydaje się **interpretacja dwupoziomowa:**
→ sprawdza się o ile sygnał maksymalny przewyższa następny
→ jeżeli różnica jest duża za zwycięski uważa się neuron o największe aktywności
→ gdy poziomy aktywacji wszystkich neuronów są poniżej pewnego progu, interpreter ostrzega że klasyfikacja jest niepewna.

Dane literaturowe wskazują że przy bardzo prostym algorytmie przetwarzania wstępnego, tą metodą można **uzyskać 90% skuteczność**.

Kompresja danych

Zadaniem kompresji danych jest zmniejszenie informacji przechowywanej lub przesyłanej na odległość przy zachowaniu możliwości jej pełnego odtworzenia (dekompresji).

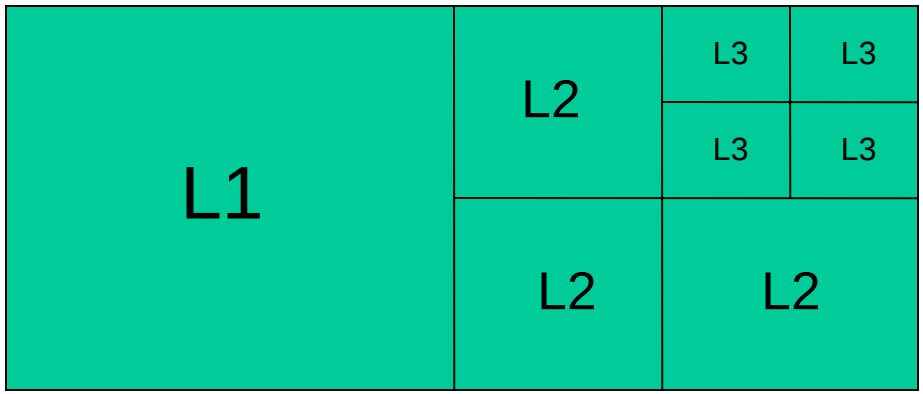
Zastosowanie sieci neuronowej umożliwia **uzyskanie nowych rozwiązań kompresji typu stratnego** (z pewną utratą informacji) o dobrych właściwościach uogólniających i stosunkowo dużym współczynniku kompresji.

Hierarchiczny podział danych

Przed przystąpieniem do kompresji dane należy podzielić na ramki odpowiednich rozmiarach:

→ podział równomierny, nie uwzględnia żadnego zróżnicowania danych w poszczególnych ramkach.

→ uwzględnienie zróżnicowania, ***podział hierarchiczny***. Obraz dzielony na segmenty o podobnym stopniu szarości. Segmentacja dokonywana przez regularną dekompozycję obrazu, prowadząca do struktury drzewiastej. Podział obrazu na bloki o różnych wymiarach, decyzja o kolejnym podziale jest podejmowana na podstawie pomiaru kontrastu rozumianego jako różnica miedzy najwyższym i najniższym stopniem szarości.



hierarchiczny podział obrazu

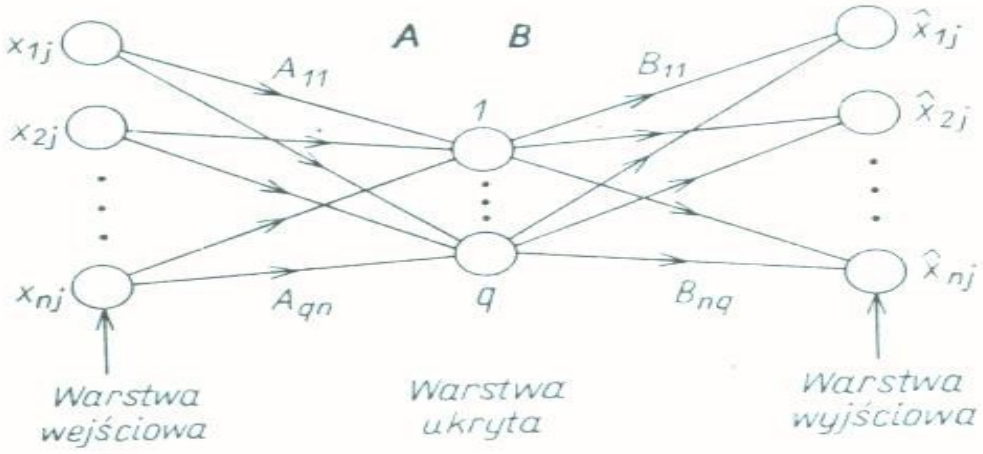
Hierarchiczny podział danych

Zastosowanie podejścia hierarchicznego w kompresji obrazów umożliwia zmniejszenie liczby wektorów uczących sieci przy zachowaniu najistotniejszych informacji zawartych w obrazie.

Zapewnienie zbliżonego do siebie kontrastu wewnątrz bloku umożliwia wydatne zmniejszenie błędu kompresji, dzięki czemu przy zadanym poziomie PSNR możliwe jest uzyskanie większych współczynników kompresji K_r .

Siec neuronowa wielowarstwowa

Przykładowa struktura sieci neuronowej do kompresji danych



Rys. 4.2
Sieć neuronowa jednokierunkowa do kompresji danych

Jest to sieć dwuwarstwowa, w której liczba elementów w warstwie wyjściowej jest równa liczbie węzłów w warstwie wejściowej. Warstwa ukryta zawiera q neuronów, przy czym q<<n. Warstwa wejściowa i ukryta stanowią właściwą kompresje danych, natomiast warstwa ukryta i wyjściowa realizują dekompresje. Sieć jest typu autoasosocjacyjnego, co oznacza, że wektor uczący d jest równy wektorowi wejściowemu x, a sygnały wyjściowe sieci x_i odpowiadają sygnałom wejściowym x_i.

Siec neuronowa wielowarstwowa

→ Kompresja dotyczy danych podzielonych na ramki (slide 11), będące ciągiem wektorów n-elementowych (n – liczba węzłów wejściowych).

→ Wobec q<<n warstwa ukryta zawiera mniejszą ilość informacji niż warstwa wejściowa, ale informacja ta prezentuje wiedzę reprezentatywną dla zbioru danych, wystarczającą do rekonstrukcji oryginalnych danych wejściowych z określoną dokładnością.

→ Warstwa ukryta reprezentuje więc **składniki główne rozkładu** (Principal Component Analysis – PCA), stanowiące jądro informacji.

→ Liczba tych składników jest równa liczbie neuronów q w warstwie ukrytej. Większa liczba q odpowiada zwiększonej informacji zawartej w neuronach warstwy ukrytej co z kolei umożliwia wierniejsze odtworzenie informacji wejściowej otrzymanej w wyniku dekompresji.

Siec neuronowa wielowarstwowa

Przy zastosowaniu sieci liniowej wektor h utworzony przez odpowiedzi neuronów w warstwie ukrytej oraz zdekompresowany wektor x odpowiadający sygnałom wyjściowym sieci są opisane następującymi równaniami macierzowymi

$$h = A\,x \quad \rightarrow \quad x' = B\,h = B\,A\,x$$

gdzie A i B są utworzone przez wagi neuronów odpowiednio warstwy ukrytej i wyjściowej sieci.
Uczenie sieci czyli optymalny dobór wag tworzących macierz A i B wymaga aby różnica miedzy x_{ij} i x'_{ij} była dla wszystkich składowych jak najmniejsza, co prowadzi do definicji funkcji celu w postaci

$$E = \tfrac{1}{2} \Sigma\Sigma \, (x_{ij} - x'_{ij})^2$$

Nie istnieje rozwiązanie analityczne problemu (prostokątności A i B).
Uczenie neuronów: najlepsze wyniki uzyskiwano stosując liniową funkcje aktywacji.

Miary kompresji

Dane odtworzone w wyniku dekompresji są obarczone zawsze pewnym błędem. Miara tego błędu może być przyjmowana w różny sposób:

→ **MSE = d(x,x') = 1/M Σ $(x_i - x_j)^2$**

gdzie M oznacza wymiar wektora danych x. W przypadku danych dwuwymiarowych wektor x tworzą kolejne dane dotyczące podobrazów.

Istotnym parametrem, określającym stosunek ilości informacji przypisanej obrazowi sprzed kompresji do ilości informacji odpowiadającej obrazowi skompresowanemu, jest współczynnik kompresji, K_r .

Im większy współczynnik K_r, tym większy zysk przy przechowywaniu i przesyłaniu informacji i zwykle większe zniekształcenia powstające w zdekompresowanym obrazie.

Zniekształcenie dekompresji

Zniekształcenie dekompresji mierzy się najczęściej za pośrednictwem współczynnika **PSNR (Peak Signal-to-Noise-Ratio)**, mierzonego w decybelach i definiowanego w postaci:

$$PSNR = 10 \log((2^k-1)^2/MSE)$$

gdzie k jest liczba bitów użytych do kodowania stopni szarości obrazu. Przy 8-bitowej reprezentacji współczynnik PSNR określa wzór

$$PSNR = 10 \log (255^2/MSE)$$

Im większa wartość współczynnika **PSNR**, tym lepsza jest jakość obrazu.

Sieć neuronowa interpolująca

Interpolacja jest procesem polegającym na określeniu wartości funkcji w punktach pośrednich w stosunku do wartości zmierzonych.
Jej celem jest przywrócenie rzeczywistej, pełnej postaci niepełnego zbioru danych na podstawie jego fragmentów.
Przy formułowaniu matematycznych założeń przyjmuje się ciągłość funkcji oraz jej pierwszej pochodnej.

Sieć neuronowa jednokierunkowa o sigmoidalnej funkcji aktywacji może z powodzeniem spełniać funkcje układu interpolującego.
Warstwa wejściowa reprezentuje dane niepełne dotyczące sygnałów zmierzonych. Warstwa wyjściowa odpowiada danym interpolowanym.

Liczba danych wyjściowych jest większa niż wejściowych, układ jest więc źle uwarunkowany i trudno jest uzyskać dobre zdolności uogólniania.
Zastosowanie **sieci z rozszerzeniem funkcyjnym Pao polepsza** uwarunkowanie problemu interpolacyjnego i powiększa zdolności uogólniania sieci.

Modelowanie obiektów dynamicznych

W odróżnieniu od procesów statycznych, takich jak rozpoznawanie wzorca niezmiennego w czasie, w **systemach dynamicznych obiekt** podlegający rozpoznaniu zależy od chwilowych wartości par uczących, będących funkcją czasu.

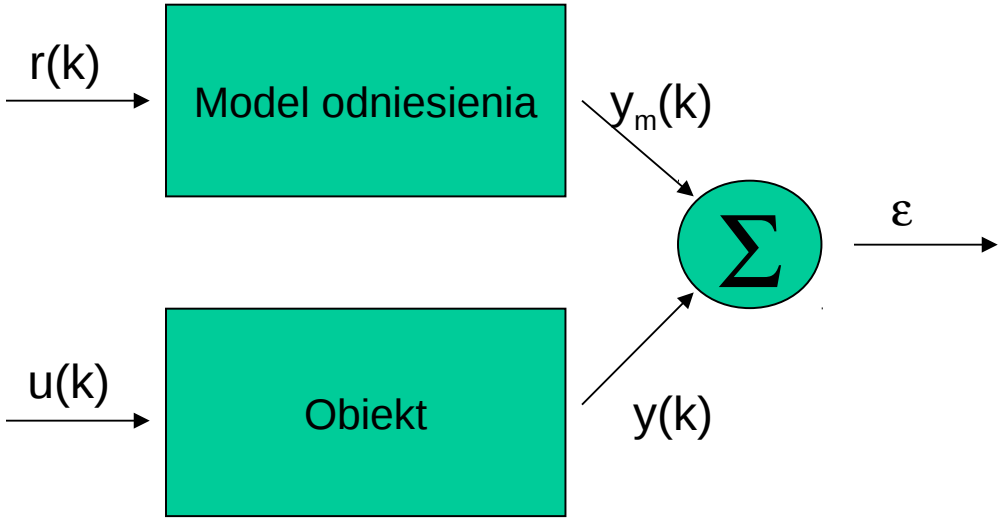
Problem identyfikacji obiektu sprowadza się do zbudowania jego modelu i określenia parametrów tego modelu w taki sposób, aby odpowiedzi obiektu y(k) i modelu y'(k) na to samo wymuszenie u(k) były sobie równe z określoną tolerancją, to znaczy

$$\| y' - y \| \leq \varepsilon$$

Sterowanie adaptacyjne znanego obiektu nieliniowego polega na doborze takiego sterowania u(k) , stanowiącego wymuszenie dla obiektu, aby odpowiedź tego obiektu y(k) śledziła i nadążała za odpowiedzią modelu odniesienia $y_m(k)$ pobudzonego sygnałem r(k).

Sterowanie adaptacyjne

Schemat układu sterowania adaptacyjnego



Wielkość $y_m(k)$ reprezentuje wielkość zadaną obiektu odniesienia przy zadanym dla niego wymuszeniu r(k). Jeżeli w układzie istnieje tylko jedno wymuszenie, to **zadaniem procesu adaptacyjnego jest dobór struktury i parametrów sterownika,** który sygnał wejściowy r(k) przetworzy w pożądaną postać sygnału sterującego u(k), zapewniającą spełnienie warunku sterowania. **Model obiektu jest zbudowany przy wykorzystaniu sieci neuronowych**.

Zestaw pytań do testu

1. Co to jest sieć CP?
2. Na czym polega uczenie Kohonena.
3. Czy sieć CP może być autoasocjacyjna?
4. Na czym polega uczenie autoasocjacyjnej sieci CP?