

Arkadiusz Popa

SAMO OPTYMALIZUJĄCE SIECI
NEURONOWE -TOPOLOGIA,
DZIAŁANIE, KIERUNKI ROZWOJU

Praca magisterska napisana
w Katedrze Analizy Matematycznej
i Teorii Sterowania
Wydziału Matematyki
Uniwersytetu Łódzkiego
pod kierunkiem
prof. dr hab. Andrzeja Nowakowskiego

2006

Spis treści

Wstęp	3
1 Wiadomości wstępne	4
1.1 Jak widzi ludzkie oko, co rozumie nasz mózg	4
1.2 Cel 'naszych' sieci neuronowych	5
1.3 Trywialne poszukiwania dyskryminujących cech	5
1.4 Budowa sieci poprzez dedukcję	6
2 Sieci oparte o jednokrotne wyliczanie wartości dyskryminantu	10
2.1 Wzorce uczące	10
2.2 Obliczanie dyskryminantów	11
2.2.1 Topologia sieci	14
2.2.2 Przykładowe obliczenia dyskryminantów	17
2.3 Obliczanie wag połączeń	18
2.4 Sieci dwu czy N-warstwowe	18
3 Sieci Neuronowe oparte na wielokrotnym obliczaniu dyskryminantów cech	24
3.1 Obliczanie współczynników	24
3.2 Budowa Topologii	25
4 Ograniczenia Samo Optymalizujących Sieci Neuronowych	27
5 Własne badania	29
5.1 Nowy dyskryminant I	30
5.2 Nowy dyskryminant II	31
5.3 Nowy dyskryminant III	31

5.4	Nowy dyskryminant IV	31
5.5	Nowy dyskryminant V	31
5.6	Wnioski	32
	Bibliografia	34

Wstęp

W ostatnich latach można zauważyć wyjątkowy rozwój sieci neuronowych. Ich próby informatycznego odwzorowania działania ludzkiego mózgu są coraz bardziej doskonałe. Ciągłe występuje jednak problem, który jest związany z dostosowaniem tradycyjnej sieci neuronowej do rozwiązywania konkretnych problemów. Dlatego obecnie chcąc zbudować sieć neuronową, która będzie umiała rozróżnić litery alfabetu (np: A,B,C,D,E,F...) wymagany jest czynnik ludzki, który określi ilość warstw wejściowych, ilość warstw ukrytych i wyjścia z sieci neuronowych. Sieć taka będzie miała zupełnie inną budowę niż sieć stworzona do rozpoznawania np. symboli +,-,# itd. Jeszcze inna będzie musiała być budowa sieci neuronowej, której zadaniem będzie stworzenie systemu eksperckiego pomagającego lekarzowi określić chorobę pacjenta na podstawie pewnych cech. Dodatkowo zdefiniowanie ilości warstw ukrytych często wiąże się z przeprowadzeniem dużej ilości prób, bądź eksperymentów skierowanych dokładnie na rozwiązanie danego problemu.

Dlatego celem niniejszej pracy jest omówienie 'algorytmu' budowy sieci neuronowej, która będzie sama dostosowywać swoją strukturę, topologię i wartości do zadanego jej problemu. W pracy tej będą omawiane tylko problemy, które możliwe są do przedstawienia w postaci dyskretnej, czyli wzorce będą posiadały pewien zbiór nie ciągłych cech, które będą przedstawiane za pomocą dwóch wartości. Na przykład 0 i 1, lub 1 i -1 . Wartości te określają czy cecha występuje w danym wzorcu. Rozważane będą również cechy, które nie są określone. Czy nie wiadomo jaki znak posiada dana cecha. W tym przypadku wykorzystywane będą trzy wartości cech, 1 (cecha występuje w danym wzorcu), -1 (cecha nie występuje we wzorcu) i 0 (nie ma pewności co do wartości cechy).

Rozdział 1

Wiadomości wstępne

1.1 Jak widzi ludzkie oko, co rozumie nasz mózg

Naszym celem jest zbudowanie takiego schematu sieci neuronowej, który potrafiłby z zadanego ciągu wzorców wyodrębnić te cechy, które najbardziej je dyskryminują i użyć ich do budowy sieci neuronowej. Pomijamy w tym miejscu kwestie związane na przykład ze skalowalnością, czy też obrotami wzorców. Czy tak działa nasz mózg? Weźmy od uwagę twarze ludzkie. Potrafimy rozpoznać osobę, patrząc na jej twarz. Możemy w ten sposób rozróżnić dziesiątki naszych bliskich i setki osób znanych na całym świecie. Potrafimy zapamiętać i rozróżnić szczegóły każdej twarzy, kształt oczu, ich kolor, wielkość nosa, ułożenie ust i wiele wiele innych cech. Ważne jest stwierdzenie, że u *każdego* wzorca, czyli twarzy, potrafimy zapamiętać charakterystyczne dla *niego* cechy. Można by w tym miejscu powiedzieć, że do takiego rozpoznawania twarzy można napisać 'zwykły' program algorytmiczny, który będzie porównywał wszystkie wyróżnione cechy i na ich podstawie rozpoznawał osobę. Otóż nie. Musimy wziąć pod uwagę kilka czynników. Na przykład zaburzenia. Zauważmy że nasz mózg potrafi rozpoznać twarz, na którą został nałożony makijaż, albo która nosi okulary, czy też jest częściowo zasłonięta przez kapelusz. Dzieje się tak dlatego, że zapamiętujemy pewien *układ cech*. Układ ten jednoznacznie wyznacza nam daną twarz. Układu tych cech uczymy się przez długi czas i pewne ich zaburzenie nie psują skuteczności naszego rozpoznawania. Jednak u małych dzieci można zauważyć charakterystyczną rzecz. Otóż ich mózg zapamiętuje 'niewiele' cech charakterystycznych twarzy. Skutkiem tego jest płacz dziecka, które zobaczyło twarz

matki np. w okularach. Efekt taki jest mimo małej ilości wzorców pamiętanych przez dzieci. Oczywiście wszystko ma swoje granice i nasz mózg również nie jest w stanie rozpoznać twarzy w której zmieniono zbyt dużo cech charakterystycznych. Czyli twarzy w której zbyt wiele cech było zaburzonych. Albo ściślej ujmując, zaburzonych cech, które bezpośrednio wchodziły w układ, który zapamiętał nasz mózg.

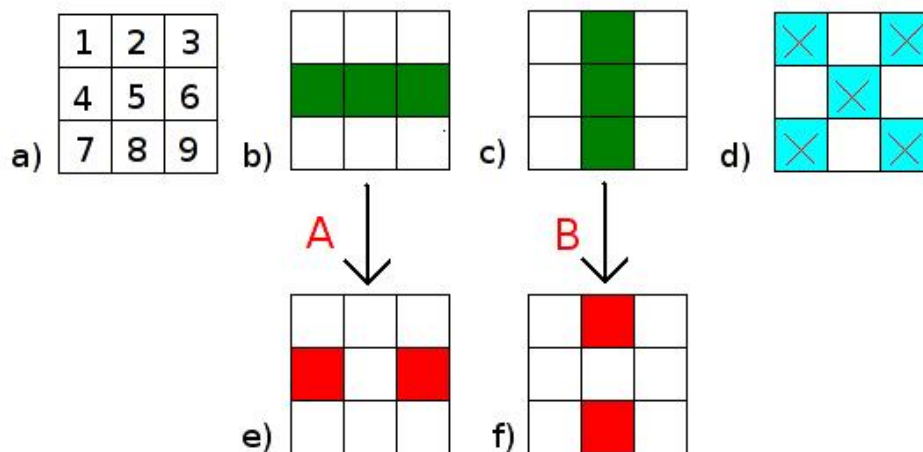
1.2 Cel 'naszych' sieci neuronowych

Chcielibyśmy zbudować taki schemat tworzenia sieci neuronowych, który będzie umiał:

- a) zdyskryminować wszystkie cechy dla każdego wzorca i każdej cieszce nadać odpowiednią wartość
- b) wybrać cechy najbardziej istotne dla danych wzorców wejściowych
- c) automatycznie zbudować topologie sieci
- d) wyliczyć wagi, funkcje aktywacji na każdym neuronie
- e) zawsze rozpoznać wszystkie wzorce uczące
- f) umieć rozpoznawać zaburzone wzorce uczące

1.3 Trywialne poszukiwania dyskryminujących cech

Rozważmy prosty przykład. Znajdźmy najbardziej dyskryminujące cechy dla podanych dwóch wzorców w następującym przykładzie: Weźmy pod uwagę wzorce wrysowane w siatkę 9 pól, o wymiarach 3x3 pola (Rys. 1 a). Pierwszy wzorec to linia pozioma przechodząca przez pola 4, 5 i 6 (Rys. 1 b), drugi to linia pionowa przechodząca przez pola 2, 5 i 8 (Rys. 1 c). Spróbujmy określić najbardziej znaczące cechy dla tych dwóch wzorców. Na pierwszy rzut oka widać, że 'narożne' cechy (nr. 1, 3, 7 i 9) nie mają żadnego znaczenia dla podanych wzorców. W obu przypadkach są 'białe', posiadają wartość 0, czyli nie wnoszą żadnej wartości do rozpoznawania wzorców. Również środkowa cecha (nr. 5) nie wnosi nic do rozpoznawania wzorców. Ponieważ w obu przypadkach przecina się tam linia wzorca i cecha ta posiada

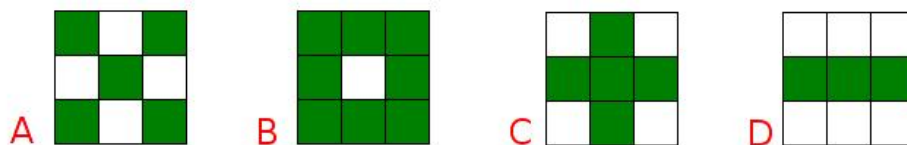


Rysunek 1.1: Przykład wyboru najbardziej dyskryminujących cech dla wzorca.

wartość 1. Wykluczone cechy zostały zaprezentowane na części d) rysunku nr. 1. Czyli możemy je całkowicie pominąć do rozpoznawania wzorców. Dlatego ostatecznie otrzymujemy, że do rozpoznania pierwszego wzorca A (linia pozioma) wystarczy sprawdzić wartość cech 4 i 6 (co najmniej jedna cecha jest wymagana do rozpoznania wzorca), co przedstawiono na rysunku 1 e). Natomiast do rozpoznania wzorca B (linia pionowa) wymagane jest użycie co najmniej jednej z dwóch cech: 2 i 8 (Rys. 1 f). Gdy nasza sieć neuronowa będzie oparta o wydedukowane tutaj cechy, będzie prawidłowo rozpoznawać przynajmniej wzorce uczące. Na razie w ramach wstępu zaburzenia nie występują, czyli jednoznacznie nie są brane pod uwagę. Czy jednak wyszukiwanie jednej najlepszej cechy jest wystarczające do prawidłowego rozpoznawania choćby wzorców? Nie. Dodatkowo wybór najbardziej istotnych cech komplikuje się wraz ze wzrostem ilości wielkości wzorców.

1.4 Budowa sieci poprzez dedukcję

Rozważmy kolejny przykład. Dla podanych na Rysunku 2) wzorców określmy najbardziej znaczące, inaczej, najbardziej dyskryminujące dany wzorec cechy. Jednak wybierając najbardziej charakterystyczne cechy dla jednego wzorca trzeba uwzględnić kształt pozostałych wzorców. Chcąc określić 'na oko' najbardziej znaczące cechy dla przedstawionych wyżej wzorców, są to graficznie przedstawione symbole X,O,+ i -, można napotkać na problem. Nasz mózg umie bez żadnych przeszkód prawidłowo

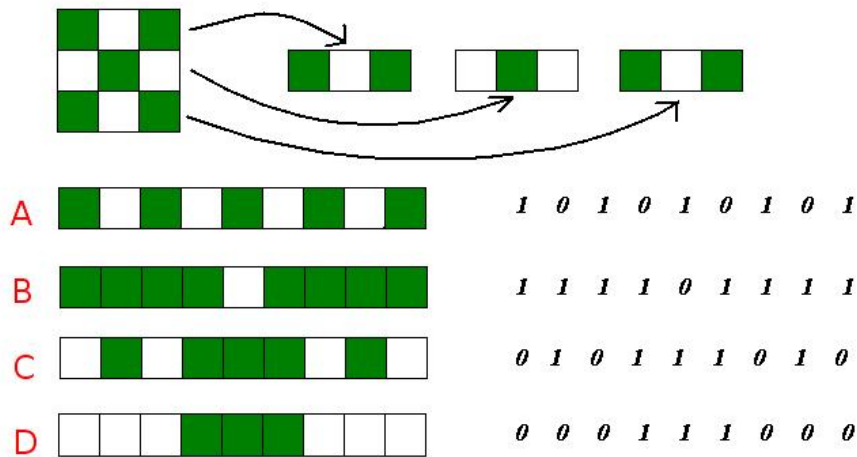


Rysunek 1.2: Przykład rozróżnienia najbardziej dyskryminujących cech we wzorcach.

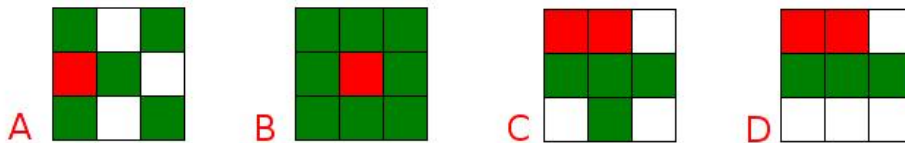
wo wskazać, który symbol przedstawia znak X,0,+ czy -, jednak znalezienie cechy, która odróżnia dany symbol od pozostałych nie jest rzeczą najprostszą. Trzeba wyszukać taką cechę, bądź układ cech, który jest jednoznaczny dla danego wzorca. Po przedstawieniu prezentowanych symboli w innej formie można łatwiej wskazać różnice między wzorcami. W miarę szybko można zauważyć, że we wzorze A) cechy 4 i 6 mają wartość 0, zaś w pozostałych wzorach 1. Natomiast wzór B) posiada jako jedyną białą cechę 5, czyli o wartości 0. Cecha ta jednoznacznie identyfikuje wzorec na tle od pozostałych, gdzie występują wartości 1. Łatwiej jest to zauważyć na Rysunku 3. Rozsunęto tu wzorce według pokazanego schematu. Wyraźnie widać, że dla wzorca A cecha 4 i/lub 6 powinna zostać użyta do budowy sieci, natomiast dla wzorca B cecha 5. Problem jest natomiast z wzorcami C i D. Nie posiadają one cech które by w sposób jednoznaczny identyfikowały dany wzorec. Jedynym rozwiązaniem jest wybranie pewnego układu cech który:

- a) będzie jednoznacznie identyfikował problematyczne wzorce C i D
- b) nie będzie przy tym identyczny z określonymi już cechami we wzorcach A i B

Można na przykład wybrać cechy 1 i 2 dla wzorców C i D. Z rysunku 3 widać wyraźnie, że taki układ cech jednoznacznie identyfikuje nam wzorce C i D. Dla wzorca C mamy wartości cech 1[0],2[1] natomiast dla wzorca D otrzymujemy 1[0], 2[0] (przyjmując zapis numer_cechy[wartość_cechy]). Również wzorce A i B są w ten sposób rozróżniane, ponieważ do ich rozróżnienia wybrano inne cechy (układy cech). Dla wzorca A otrzymujemy wartości dla cech następujące: 1[1], 2[0], natomiast dla B: 1[1], 2[1]. Łatwo zauważyć, że cechy wybrane do rozróżnienia wzorców C i D mają inne wartości niż we wzorcach A i B. Jak już wspomniano wcześniej wzorce A i B są identyfikowane jednoznacznie przez cechy, odpowiednio: 4[0] i 5[1] (Rysunek 4). W ten sposób wyznaczając najlepiej dyskryminujące cechy dla podanych wzorców można pokusić się o budowę sieci neuronowej, która zaprezentowane cztery wzorce



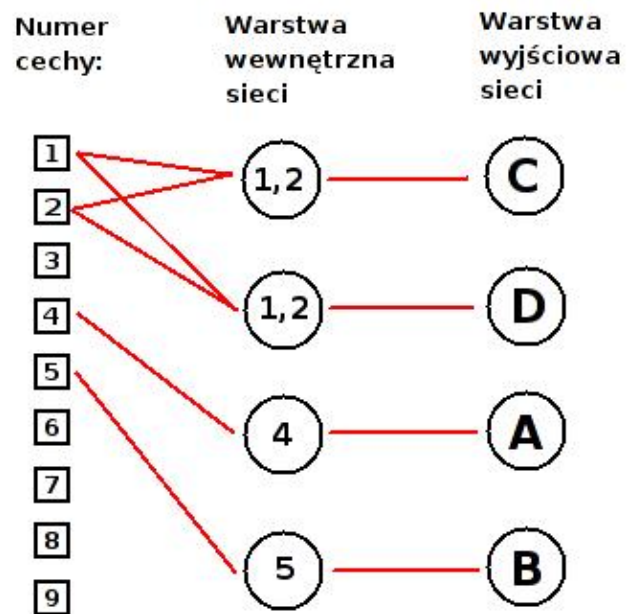
Rysunek 1.3: Inny sposób przedstawienia wzorców w celu łatwiejszego określenia cech znaczących.



Rysunek 1.4: Zaznaczenie najbardziej znaczących cech dla poszczególnych wzorców.

będzie rozpoznawać jednoznacznie. Na rysunku 5) widać jak taka topologia wygląda. Warstwa wejściowa zawiera cechy wykorzystane do rozpoznawania wzorca. Warstwa wewnętrzna składa się z neuronów niezbędnych do rozpoznawania wzorców. Natomiast warstwą wyjściową jest klasa wzorców uczących. Wyraźnie widać, że do rozpoznania wzorca A i B użyto po jednym połączeniu do neuronu, z wydedukowanej wcześniej cechy. Natomiast do prawidłowego rozpoznania wzorców C i D niezbędne są po dwa połączenia. Zadaniem prezentowanych sieci neuronowych jest automatyczne wybranie najbardziej dyskryminantnych cech, wygenerowanie topologii, obliczenie dyskryminantu i wag dla dowolnych danych uczących. Proces ten przebiega bez udziału czynnika ludzkiego. Praca ta będzie omawiać trzy wersje *Samo Optymalizujących Sieci Neuronowych*:

- sieci oparte o jednokrotne wyliczanie wartości dyskryminanta dla każdej cechy w każdym wzorcu
- sieci z minimalną ilością połączeń między neuronami, z wyliczaniem wartości dyskryminantów dla każdej warstwy wewnętrznej



Rysunek 1.5: Przedstawienie budowy topologii sieci neuronowej.

- sieci z wyliczaniem wartości dyskryminantów dla każdej warstwy wewnętrznej

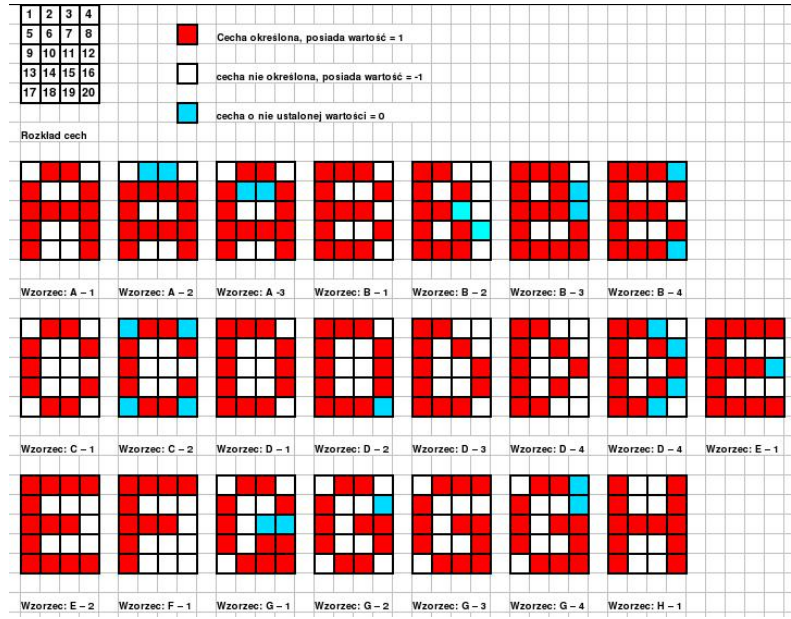
Rozdział 2

Sieci oparte o jednokrotne wylizczanie wartości dyskryminantu

2.1 Wzorce uczące

Do omówienia tego rodzaju sieci neuronowych zostaną wykorzystane wzorce uczące z pracy [1]. Przyjmuje się, że cecha, która występuje w danym wzorcu ma wartość 1, cecha która nie występuje we wzorcu ma wartość -1 . Natomiast cecha, która nie jest określona, nie ma pewności czy występuje w danym wzorcu czy też nie posiada wartość 0; Rozpoznawane będą litery alfabetu, od A do H, wpisane w siatkę 4×5 . Czyli każda litera posiadać będzie 20 cech. Wzorce te przedstawiono na rysunku 2.1. Można wyróżnić 8 klas i 22 wzorce. Niektóre wzorce są liczniejsze w klasie, np. wzorce D występują aż 5 razy w klasie. Natomiast klasa H liczy tylko jeden wzorec. Wpływ liczności klas na dyskryminowanie cech zostanie omówiony w dalszej części pracy.

Teraz można rozpisać wartości cech w podanych wzorcach. Przedstawiono to na rysunku 2.2. Wzorec A-2 powtórzono dwukrotnie. Pierwsze wystąpienie wzorca A-2 ma charakter informacyjny, gdzie kolorem zaznaczono odpowiednie wartości poszczególnych cech.



Rysunek 2.1: Przykład wzorców uczących.

2.2 Obliczanie dyskryminantów

W tej wersji sieci neuronowych dyskryminanty obliczane są jednokrotnie. To znaczy dla każdej cechy i każdego wzorca oblicza się dyskryminant tylko jeden raz. Wartość tego dyskryminantu klasyfikuje daną cechę czy jest ona odróżnialna od tej samej cechy dla pozostałych wzorców. Na podstawie tych wyliczeń wybiera się cechy, które mają najwyższą wartość dyskryminanta. Oznacza to wybór cech które najlepiej dyskryminują dany wzorec względem innych. Zapewnia nam to wybór takich cech, które pozwalają jednoznacznie zakwalifikować wzorec do odpowiedniej mu klasy. W pracy [1] zaproponowano następujące wzory na obliczanie wartości dyskryminantów:

Do obliczania ilości cech w klasie służą wzory 2.1 i 2.2. Jest to suma wartości cech dodatnich i ujemnych liczona w ramach wzorców występujących w klasach. :

$$\forall_{m \in \{1, \dots, M\}} \forall_{k \in \{1, \dots, K\}} : P_k^m = \sum_{u_k^n \in \{u \in U \vee C^m : u_k^n > 0, n \in \{1, \dots, Q\}\}} u_k^n \quad (2.1)$$

$$\forall_{m \in \{1, \dots, M\}} \forall_{k \in \{1, \dots, K\}} : N_k^m = \sum_{u_k^n \in \{u \in U \vee C^m : u_k^n < 0, n \in \{1, \dots, Q\}\}} -u_k^n \quad (2.2)$$

Gdzie:

1. K - określa ilość cech
2. M - określa ilość klas
3. u_k^n - określa wartość cechy dla k-tej klasy i n-tej cechy

Wzorzec Cecha	A-2	A-1	A-2	A-3	B-1	B-2	B-3	B-4	C-1	C-2	D-1	D-2	D-3	D-4	D-5	E-1	E-2	F-1	G-1	G-2	G-3	G-4	H-1
1	-1	-1	-1	-1	1	1	1	1	-1	0	1	1	1	1	1	1	1	1	-1	-1	-1	-1	1
2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
4	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
5	1	1	1	1	1	1	1	1	-1	0	1	1	1	1	1	1	1	1	-1	-1	-1	-1	1
6	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	-1
7	-1	-1	1	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
8	-1	1	-1	-1	1	1	1	1	-1	-1	-1	-1	-1	-1	-1	1	1	1	-1	-1	-1	-1	1
9	-1	-1	1	1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
10	-1	-1	-1	-1	1	1	1	1	1	1	1	1	1	1	1	1	1	-1	1	1	1	1	-1
11	0	1	0	1	1	-1	1	1	1	1	1	1	-1	-1	0	1	1	1	1	1	1	1	-1
12	-1	-1	1	0	-1	1	1	-1	-1	-1	-1	-1	1	1	1	-1	-1	-1	-1	-1	-1	-1	-1
13	-1	1	-1	1	1	0	1	1	-1	-1	-1	-1	-1	-1	-1	1	1	1	0	1	0	1	1
14	-1	-1	1	1	-1	1	-1	-1	-1	-1	-1	-1	-1	1	1	-1	-1	-1	1	-1	-1	-1	-1
15	-1	-1	-1	-1	1	1	1	1	1	1	1	1	-1	0	1	1	-1	1	1	1	1	1	-1
16	-1	-1	-1	-1	-1	-1	-1	0	-1	0	-1	-1	-1	-1	-1	1	1	1	-1	-1	1	0	1
17	1	1	1	1	1	-1	0	1	1	1	1	1	-1	-1	0	-1	-1	-1	1	0	-1	0	1
18	1	1	1	1	-1	-1	0	-1	-1	-1	1	1	1	1	1	0	-1	-1	0	1	1	1	1
19	1	1	1	1	1	0	1	1	1	1	1	1	-1	0	-1	-1	-1	1	1	1	1	1	1
20	1	1	1	1	-1	-1	1	0	-1	0	-1	0	-1	-1	-1	1	1	-1	1	-1	1	1	1

Rysunek 2.2: Rozkład wartości cech we wzorcach uczących.

4. P_k^m - określa ilość cech m o wartości 1 w klasie k
5. N_k^m - określa ilość cech m o wartości -1 w klasie k

Wzory te można zaimplementować następująco:

```

$P=$N=array();
foreach ($tablica[$krok] as $w => $cecha)
{
    if($w!=0)
    {
        foreach ($cecha as $q => $wart)
        {
            $m=$pozycjeWzorcow[$q];
            if($wart==1)
            {
                if(isset($P[$w] [$m]))
                {
                    $P[$w] [$m]++;
                }
                else
                {
                    $P[$w] [$m]=1;
                }
            }
        }
    }
}

```

```

    }
elseif ($wart===-1)
{
    if(isset($N[$w][$m]))
        $N[$w][$m]++;
    else
        $N[$w][$m]=1;
    }
}
}
}
}

```

Za pomocą wzoru (2.3) obliczana jest ilość wzorców występujących w klasie, czyli moc klasy.

$$\forall_{m \in \{1, \dots, M\}} Q^m = \|\{u^n \in U \vee C^m : n \in \{1, \dots, Q\}\}\| \quad (2.3)$$

W zaimplementowanym programie wzór ten jest obliczany następująco:

```

foreach ($tablica[0][0] as $k => $w)
{
    //$k numer kolumny
    //$w wzorzec (np: A)
    $pozycjeWzorcow[$k]=$w;
    if(isset($Q[$w]))
        $Q[$w]++;
    else
        $Q[$w]=1;
}
}

```

Następnie obliczana jest prawdopodobna wartość dla cech których wartość wzorców uczących wynosi 0:

$$x_k^n = \begin{cases} 1 & \text{jeżeli } u_k^n = +1 \\ \frac{P_k^m}{P_k^m + N_k^m} & \text{jeżeli } u_k^n = 0 \\ 0 & \text{jeżeli } u_k^n = -1 \end{cases} \quad (2.4)$$

$$y_k^n = \begin{cases} 1 & \text{jeżeli } u_k^n = -1 \\ \frac{N_k^m}{P_k^m + N_k^m} & \text{jeżeli } u_k^n = 0 \\ 0 & \text{jeżeli } u_k^n = +1 \end{cases} \quad (2.5)$$

Następnie wyliczane są (wzory 2.4 i 2.5) wartości gdy występują zera w danej cesze w określonym wzorcu. Jeśli występuje zero w danej cesze to nie wskazane by było otrzymać wartość dyskriminanta równą 0 dla takiej cechy. Dlatego oblicza się iloraz cech dodatnich przez sumę cech dodatnich i ujemnych. Uzyskujemy w ten sposób dwie wartości x_k^n i y_k^n . Niejako maskujemy wartość cechy i ustawiamy ją na 1 lub -1 w zależności od tego jakich wartości cech w danej klasie jest więcej. Jest to proste szacowanie prawdopodobieństwa wystąpienia cechy o określonej wartości na podstawie pozostałych wartości cech w danej klasie. Dzięki temu we wzorze (2.6) możliwe jest obliczenie ilości cech dodatnich w K -tej cesze i w M -tej klasie, również dla cech posiadających zera.

$$\forall_{m \in \{1, \dots, M\}} \forall_{k \in \{1, \dots, K\}} \hat{P}_k^m = \sum_{u^n \in C^m} x_k^n \in \hat{N}_k^m = \sum_{u^n \in C^m} y_k^n \quad (2.6)$$

$$\forall_{m \in \{1, \dots, M\}} \forall_{u^n \in C^m} \forall_{n \in \{1, \dots, Q\}} \forall_{k \in \{1, \dots, K\}} \begin{cases} \hat{u}_k^{n+} = x_k^n & \text{jeżeli } x_k^n \neq 0 \\ \hat{u}_k^{n-} = y_k^n & \text{jeżeli } y_k^n \neq 0 \end{cases} \quad (2.7)$$

Korzystając z już dokonanych wyliczeń, wzór na obliczenie wartości dyskriminanta w pracy [1] zaproponowano następująco.

$$\forall_{m \in \{1, \dots, M\}} \forall_{u^n \in C^m} \forall_{n \in \{1, \dots, Q\}} \forall_{k \in \{1, \dots, K\}} \begin{cases} d_k^{m+} = \frac{\hat{P}_k^m}{(M-1)Q^m} \sum_{\substack{h=1 \\ h \neq m}}^M \left(1 - \frac{\hat{P}_h^n}{Q^h}\right) & \text{jeżeli } x_k^n \neq 0 \\ d_k^{m-} = \frac{\hat{N}_k^m}{(M-1)Q^m} \sum_{\substack{h=1 \\ h \neq m}}^M \left(1 - \frac{\hat{N}_h^n}{Q^h}\right) & \text{jeżeli } y_k^n \neq 0 \end{cases} \quad (2.8)$$

Używając przedstawionych wzorów można obliczyć wartości dyskriminatów dla przykładu przedstawionego na wstępie tego rozdziału. Obliczone wartości dyskriminatów, według powyższych wzorów, dla każdego wzorca i każdej cechy znajdującej się w nim przedstawia rysunek 2.3.

2.2.1 Topologia sieci

Po dokonaniu obliczeń dyskriminatów można przejść do budowy topologii sieci. Procedure budowy topologii sieci neuronowej można przedstawić następująco:

	A	A	A	B	B	B	B	C	C	D	D	D	D	D	E	E	F	G	G	G	G	H
1	0.71	0.71	0.71	0.43	0.43	0.43	0.43	0.71	0.71	0.43	0.43	0.43	0.43	0.43	0.43	0.43	0.43	0.71	0.71	0.71	0.71	0.43
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0.29	0.29	0.29	0.29	0.29	0.29	0.29	0.86	0.86	0.29	0.29	0.29	0.29	0.29	0.29	0.29	0.29	0.86	0.86	0.86	0.86	0.29
6	0.14	0.14	0.14	0.14	0.14	0.14	0.14	0.14	0.14	0.14	0.14	0.14	0.14	0.14	0.14	0.14	0.14	0.14	0.14	0.14	0.14	1
7	0	0.5	0.5	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07
8	0.14	0.38	0.38	0.52	0.52	0.52	0.52	0.62	0.62	0.62	0.62	0.62	0.62	0.62	0.62	0.52	0.52	0.62	0.62	0.62	0.62	0.52
9	0	0.67	0.67	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
10	0.71	0.71	0.71	0.43	0.43	0.43	0.43	0.43	0.43	0.43	0.43	0.43	0.43	0.43	0.43	0.43	0.71	0.43	0.43	0.43	0.43	0.71
11	0.25	0.25	0.25	0.16	0.2	0.16	0.16	0.25	0.25	0.09	0.09	0.41	0.41	0.41	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.89
12	0.08	0.42	0.42	0.08	0.42	0.42	0.08	0.23	0.23	0.06	0.06	0.51	0.51	0.51	0.23	0.23	0.23	0.23	0.23	0.23	0.23	0.23
13	0.19	0.24	0.19	0.33	0.33	0.33	0.33	0.81	0.81	0.81	0.81	0.81	0.81	0.81	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33
14	0.04	0.58	0.58	0.14	0.2	0.14	0.14	0.22	0.22	0.1	0.1	0.33	0.33	0.22	0.22	0.22	0.2	0.14	0.14	0.14	0.14	0.22
15	0.68	0.68	0.68	0.46	0.46	0.46	0.46	0.46	0.46	0.32	0.32	0.32	0.14	0.32	0.46	0.46	0.68	0.46	0.46	0.46	0.46	0.68
16	0.48	0.48	0.48	0.48	0.48	0.48	0.48	0.48	0.48	0.48	0.48	0.48	0.48	0.48	0.67	0.67	0.67	0.29	0.29	0.19	0.29	0.67
17	0.48	0.48	0.48	0.29	0.19	0.29	0.29	0.48	0.48	0.2	0.2	0.3	0.3	0.3	0.67	0.67	0.67	0.2	0.3	0.3	0.3	0.48
18	0.57	0.57	0.57	0.57	0.57	0.57	0.57	0.57	0.57	0.57	0.57	0.57	0.57	0.57	0.57	0.57	0.57	0.57	0.57	0.57	0.57	0.57
19	0.32	0.32	0.32	0.32	0.32	0.32	0.32	0.32	0.32	0.21	0.21	0.21	0.18	0.21	0.82	0.82	0.82	0.32	0.32	0.32	0.32	0.32
20	0.56	0.56	0.56	0.36	0.36	0.15	0.36	0.58	0.58	0.58	0.58	0.58	0.58	0.58	0.56	0.56	0.58	0.39	0.12	0.39	0.39	0.56

Rysunek 2.3: Obliczenia wartości dyskryminantów.

1. Wyliczenie wartości dyskryminanta dla każdej cechy i dla każdego wzorca
2. Z każdego wzorca zostaje wybrana cecha która posiada najwyższą wartość dyskryminanta. Jeśli wzorzec posiada kilka dyskryminantów, które mają maksymalną wartość, wybierana jest pierwsza dowolna cecha z maksymalnym dyskryminantem. Wybrana cecha stanowi pierwszą warstwę sieci.
3. Każdy wzorzec powinien być jednoznacznie reprezentowany przez wybrane cechy. Jeśli wybrana cecha, bądź zbiór cech, jest identyczny dla kilku wzorców (z różnych klas) wybierana jest kolejna cecha z najwyższym dyskryminantem (z cech, które nie były jeszcze użyte do budowy topologii sieci). Wyboru dokonujemy dla wszystkich cech. Nie dokonuje się powtórnego wyliczania dyskryminantów.
4. Krok 3 powtarzany jest, dopuki każda z klas nie będzie jednoznacznie zidentyfikowana przez zbiór cech z kolejnymi największymi dyskryminantami. Zbiór ten musi być unikalny w ramach klas. Każda kolejna iteracja powoduje dodanie nowej warstwy do sieci neuronowej.
5. Jeśli ostatnią warstwę sieci dla dowolnej klasy stanowi kilka neuronów, wybierana jest maksymalna wartość z grupy tych neuronów.

Okazuje się jednak, że algorytm ten, przedstawiony w pracy [1] nie daje oczekiwanych wyników. Mimo, że zapewnia unikalność wybranych cech dla wzorców w ramach wszystkich klas, to nie gwarantuje, że wybrane cechy będą jednoznacznie

identyfikować tylko i wyłącznie określoną klasę. Innymi słowy algorytm ten powoduje, że cechy wybrane w kolejnych epokach (warstwach) mogą idealnie pasować do kilku klas, a z tego wynika że może wystąpić niejednoznaczność w późniejszym przyporządkowaniu wzorca do określonej klasy. W pracy tej wysunięto bardzo mylny wniosek 'The 2-nd epoch is the final one and the presented topology is sufficient to classify all training data correctly and unambiguously', czyli 'Druga epoka (warstwa) jest końcowa, a utworzona topologia jest wystarczająca do klasyfikowania danych wszystkich wzorców uczących prawidłowo i nie dwuznacznie' (tłumaczenie własne). Wadę tego algorytmu i błędne wnioski wyciągnięte w cytowanej pracy, można łatwo wykazać na prostym przykładzie. Weźmy pewien zbiór klas, który przedstawiono na rysunku 2.4. Każda klasa zawiera jeden wzorec. Przykład ten zawiera 7 klas, każda składa się z 11 cech. Obliczając wartość dyskryminantów dla poszczegól-

	A	B	C	D	E	F	G
1	-1	1	-1	1	1	1	-1
2	1	1	-1	1	1	1	-1
3	1	1	-1	-1	1	1	-1
4	-1	1	1	1	1	-1	1
5	1	1	-1	-1	1	1	0
6	-1	-1	-1	-1	-1	-1	1
7	-1	1	1	1	1	-1	1
8	-1	-1	-1	-1	1	1	-1
9	1	1	1	1	-1	-1	1
10	1	-1	-1	1	0	-1	0
11	1	1	1	1	-1	-1	1

Rysunek 2.4: Przykład cech, dla których algorytm zbuduje złą topologię.

nych cech, otrzymywany jest wynik zgodny z tabelą 2.5. Następnie dokonywany jest

	A	B	C	D	E	F	G
1	0.67	0.5	0.67	0.5	0.5	0.5	0.67
2	0.33	0.33	0.83	0.33	0.33	0.33	0.83
3	0.5	0.5	0.67	0.67	0.5	0.5	0.67
4	0.83	0.33	0.33	0.33	0.33	0.83	0.33
5	0.5	0.5	0.83	0.83	0.5	0.5	0
6	0.17	0.17	0.17	0.17	0.17	0.17	1
7	0.83	0.33	0.33	0.33	0.33	0.83	0.33
8	0.33	0.33	0.33	0.33	0.83	0.83	0.33
9	0.33	0.33	0.33	0.33	0.83	0.83	0.33
10	0.83	0.67	0.67	0.83	0	0.67	0
11	0.33	0.33	0.33	0.33	0.83	0.83	0.33

Rysunek 2.5: Obliczone wartości dyskryminantów.

wyбір cech z największymi wartościami dyskryminantów dla poszczególnych klas. Algorytm dokonuje wybory trzech warstw sieci. W pierwszej warstwie wybrana jest wspólna cecha numer 4 dla dwóch klas: A i F. Cecha ta ma najwyższą wartość dyskryminanta równą 0.83 właśnie dla cechy 4. W drugiej epoce wybierane są kolejne największe wartości dyskryminanta. Po raz kolejny największe wartości dla klas A

i F wypadają na wspólnej cesze 7. Dlatego dobierana jest kolejna warstwa, która to dopiero zapewnia unikalność wybranych cech w ramach klasy. Całość wyboru maksymalnych dyskrymiantów przedstawiono w tabeli 2.6. Jak do tej pory całość jest

	A	B	C	D	E	F	G
4	0.83	0	0	0	0	0.83	0
7	0.83	0	0	0	0	0.83	0
10	0.83	0.67	0	0.83	0	0	0
1	0	0.5	0.67	0	0	0	0.67
3	0	0.5	0	0.67	0	0	0
2	0	0	0.83	0	0	0	0.83
5	0	0	0.83	0.83	0	0	0
8	0	0	0	0	0.83	0.83	0
9	0	0	0	0	0.83	0	0
11	0	0	0	0	0.83	0	0
6	0	0	0	0	0	0	1

Rysunek 2.6: Wybór maksymalnych dyskrymiantów.

zgodna z algorytmem. Niestety po sprawdzeniu sieci za pomocą wzorców uczących okazuje się, że skonstruowana sieć neuronowa nie identyfikuje jednoznacznie wszystkich wzorców uczących. Zaprezentowano to na rysunku 2.7. Dzieje się tak dlatego, że

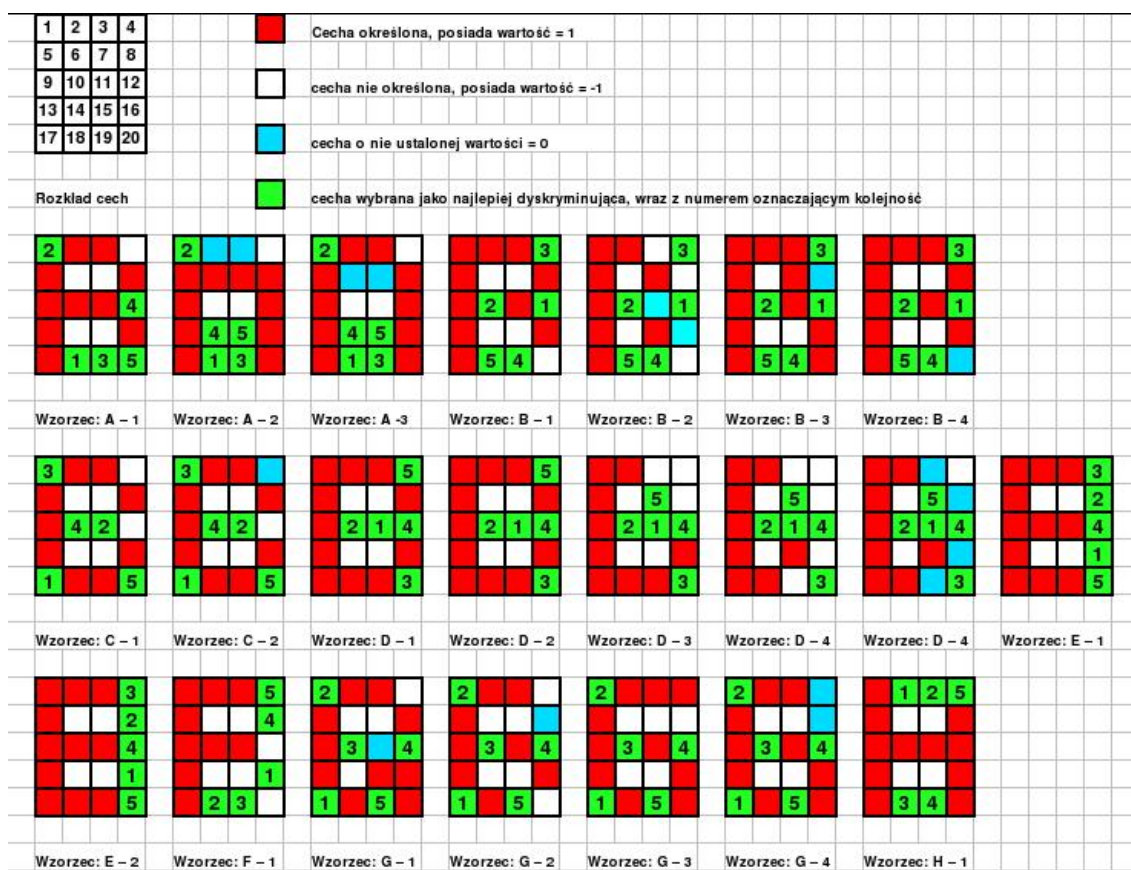
	A	B	C	D	E	F	G
1 A	1	-0.4	-0.43	-0.29	-1	0.33	-0.47
2 B	-1	1	-1	-1	-1	-1	-1
3 C	-1	-0.2	1	0.29	-1	-1	0.2
4 D	-0.33	-0.4	-0.29	1	-1	-1	-1
5 E	-0.67	0.6	-1	-0.64	1	-0.33	-1
6 F	0.33	-1	-1	-1	-1	1	-1
7 G	-0.67	-0.6	0.64	0.29	-1	-1	1

Rysunek 2.7: Niejednoznaczne rozpoznanie wzorców uczących przez cechy

2.2.2 Przykładowe obliczenia dyskrymiantów

Wykorzystując wzory opisane wcześniej można dla prezentowanego przykładu zaprezentować obliczenie dyskrymiantów.

Zbiór wybranych cech dla każdego wzorca przedstawia rysunek 2.8. Rysunek przedstawia minimalną ilość, czyli 5 cech, jaka jest wymaga by wzorce były rozpoznawane jednoznacznie przez sieć neuronową. Cechy użyte do budowy topologii sieci zostały zaznaczone na zielono. Numery znajdujące się w zielonych polach oznaczają kolenność z jaką zostały wybrane prezentowane cechy.

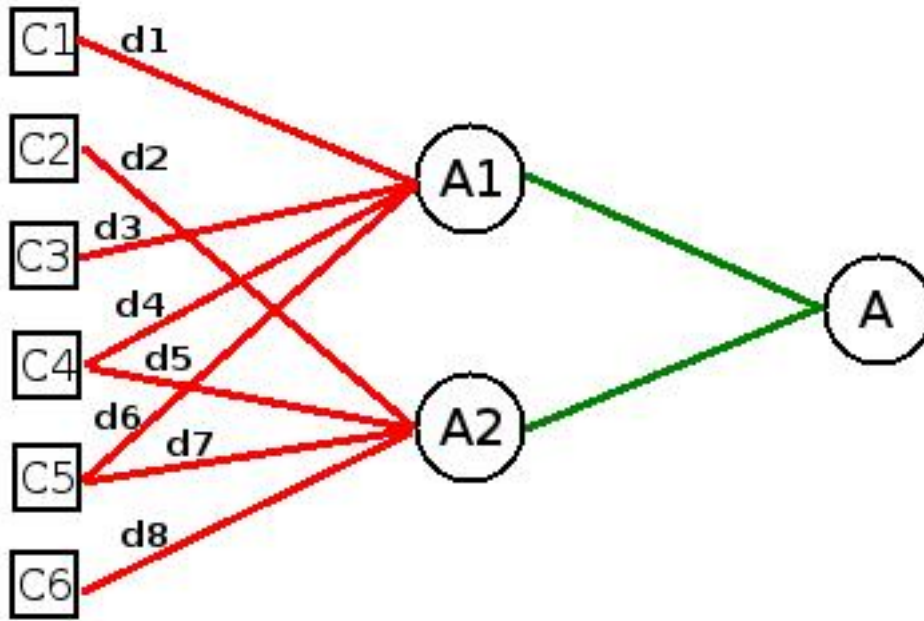


Rysunek 2.8: Wybór pięciu cech najlepiej dyskryminujących każdy wzorzec.

2.3 Obliczanie wag połączeń

2.4 Sieci dwu czy N-warstwowe

Z artykułu [1] jano wynika, że sieć posiada tyle warstw ile jest potrzebnych do rozpoznawania wzorców. W całym artykule podkreślana jest waga kolejnych epok (inaczej warstw) sieci, ich kolejność, a nawet ułożenie neuronów w sieci. Jednak zaproponowany sposób obliczania wag sieci prowadzi do interesujących wniosków. Okazuje się, licząc wagi proponowanym sposobem, że dowolną sieć N-warstwową można przedstawić jako sieć dwu warstwową (przyjmując że pierwszą warstwą jest warstwa wejściowa sieci). Mało tego, okazuje się, że sieć można zapisać jako pewien wielomian i w taki właśnie sposób przechowywać jej strukturę. Na rysunku 2.9 przedstawiono przykład prostej sieci neuronowej. Sieć ta posiada warstwę wejściową, którą stanowią cechy oznaczone jako C1, C2, C3, C4, C5 i C6. Cechy są połączone z warstwą pośrednią, czyli neuronami A1 i A2 stanowiącymi wzorce z klasy A. Dla każdej klasy w warstwie wewnętrznej musi znajdować się conajmniej jeden neuron.



Rysunek 2.9: Obniżenie znaczenia cech prowadzi do zubożenia wzorców uczących

Jeśli ilość wzorców uczących w klasie jest większa ilość takich neuronów również może być większa. Wzrost liczby neuronów w klasie wewnętrznej jest spowodowany tym, że poszczególne wzorce uczące różnią się od siebie, w ramach jednej klasy, pod względem wartości cech wybranych do budowy topologii sieci. Na rysunku 2.9 przedstawiono sytuację, w której wybrane zostały 4 cechy niezbędne do jednoznacznego rozpoznania poszczególnych wzorców. Wzorzec A1 wymaga cech C1, C3, C4, C5, a do rozpoznania wzorca A2 wybrane zostały cechy C2, C4, C5 i C6. Niech SA_1 oznacza sumę dyskryminantów dla neuronu A1, a SA_2 sumę dyskryminantów odpowiednio dla neuronu A2, co zapisać można wzorem:

$$SA_1 = d_1 + d_3 + d_4 + d_6 \quad (2.9)$$

$$SA_2 = d_2 + d_5 + d_7 + d_8 \quad (2.10)$$

Jeśli ilość wzorców uczących jest większa, to wybierane są tylko wzorce, które posiadają unikalny zestaw cech. Inaczej mówiąc, w sytuacji gdy część wzorców uczących, przynależnych do jednej klasy, jest identyfikowana przez taki sam zbiór cech, wybierany jest tylko jeden wzorzec. Wagi obliczane są w sposób opisany w poprzednim podrozdziale. Czyli na przykład waga dla połączenia między cechą C1 a neuronem A1 będzie ilorazem wartości dyskryminantu na tym połączeniu przez sumę dyskry-

minantów na wszystkich połączeniach do tego neuronu. Co można zapisać:

$$W_{C_1}^{A_1} = d_1/(d_1 + d_3 + d_4 + d_6) \quad \text{inaczej} \quad W_{C_1}^{A_1} = d_1/SA_1 \quad (2.11)$$

$$W_{C_2}^{A_2} = d_2/(d_2 + d_5 + d_7 + d_8) \quad \text{inaczej} \quad W_{C_2}^{A_2} = d_2/SA_2 \quad (2.12)$$

gdzie SA_1 i SA_2 są sumami wartości dyskryminantów dla wszystkich połączeń do neuronu A1 i A2. Analogicznie przedstawiono w następnym wzorze obliczenie wagi połączenia $W_{C_2}^{A_2}$ dla drugiego neuronu. Teraz można zapisać wagę wszystkich połączeń dla neuronów A1 i A2 jako sumę wartości wag i wartości wybranych cech. Zostało to zapisane wzorem:

$$W_{A_1} = W_{C_1}^{A_1} * P_1 + W_{C_3}^{A_1} * P_3 + W_{C_4}^{A_1} * P_4 + W_{C_6}^{A_1} * P_6 \quad (2.13)$$

$$W_{A_2} = W_{C_2}^{A_2} * P_2 + W_{C_5}^{A_2} * P_5 + W_{C_7}^{A_2} * P_7 + W_{C_8}^{A_2} * P_8 \quad (2.14)$$

gdzie wartości P_x oznaczają wartość cechy x .

Okazuje się, że by zapisać działanie podsieci A1 i A2 dla klasy A należy wybrać maksimum z dwóch kombinacji liniowych. Kombinacje to pochodzą właśnie z obliczonych powyżej wag dla wzorca A1 i A2. Ogólnie można wzór ten zapisać następująco:

$$A = \max\{W_{A_1}, W_{A_2}\}$$

gdzie po podstawieniu wcześniej wyliczonych wartości wag otrzymuje się funkcję maksimum z dwóch wartości:

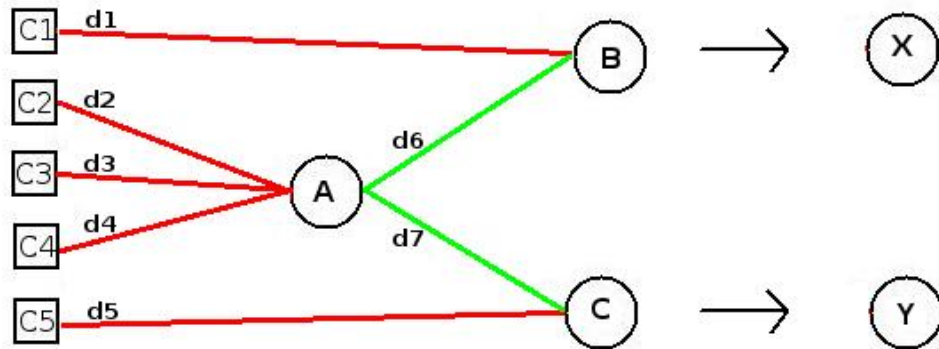
$$A = \max \left\{ W_{C_1}^{A_1} * P_1 + W_{C_3}^{A_1} * P_3 + W_{C_4}^{A_1} * P_4 + W_{C_6}^{A_1} * P_6, \right. \\ \left. W_{C_2}^{A_2} * P_2 + W_{C_5}^{A_2} * P_5 + W_{C_7}^{A_2} * P_7 + W_{C_8}^{A_2} * P_8 \right\} \quad (2.15)$$

Definiując w ten sposób funkcję dla każdej klasy, rozpoznawanie wzorca sprowadza się do obliczenia wagi dla każdego połączenia. Następnie wybierana jest wzorzec z klasy, dla której wyliczona wartość jest największa. Mogą wystąpić trzy przypadki:

- Największą wartość zwróciła funkcja, do której należał testowy wzorzec - czyli sieć prawidłowo rozpoznała wzorzec. Prawidłowa klasyfikacja wzorca nastąpiła jednoznacznie.
- Największą wartość zwróciła funkcja, do której nie należał testowy wzorzec - czyli sieć nieprawidłowo rozpoznała wzorzec. Wzorzec został jednoznacznie źle sklasyfikowany.

- Największą równą sobie wartość zwróciło więcej niż jedna funkcja - czyli wzorec nie został jednoznacznie sklasyfikowany. Został rozpoznany przez większą ilość klas. W takim przypadku zaleca się dodać koejne cechy identyfikujące wzorce.

Kolejnym interesującym wnioskiem z badań, wynikającym poniekąd z powyższego, jest fakt iż dowolną sieć neuronową (budowaną według opisanych tu reguł) składającą się z N-warstw (epok) można przedstawić w postaci sieci dwuwarstwowej. Niech wynikiem pewnych danych wejściowych będzie sieć pokazana na rysunku 2.10, gdzie przez A,B,C oznaczono trzy neurony. Pierwszą warstwę wejściową stanowią połączenia cech użytych do budowy topologii sieci z neuronem A. Neuron A znajduje się w warstwie drugiej, natomiast neurony B i C przynależą do trzeciej warstwy sieci. Wyjściem z neuronu B niech będzie klasa oznaczona jako X, natomiast wyjściem neuronu C niech będzie klasa oznaczona jako Y. Zgodnie z opisanym wcze-



Rysunek 2.10: Sieć neuronowa N-wartwowa

śniej sposobem obliczone zostają wagi. Wzór przedstawia obliczenie wag dla neuronu A:

$$W_{C2}^A = \frac{d2}{(d2+d3+d4)} \quad (2.16)$$

$$W_{C3}^A = \frac{d3}{(d2+d3+d4)} \quad (2.17)$$

$$W_{C4}^A = \frac{d2}{(d2+d3+d4)} \quad (2.18)$$

gdzie $d2$ oznacza wartość dyskryminanta. Czyli wagę dla całego neuronu można

zapisać jako:

$$W_A = \frac{d2}{(d2 + d3 + d4)} * P_2 + \frac{d3}{(d2 + d3 + d4)} * P_3 + \frac{d4}{(d2 + d3 + d4)} * P_4$$

gdzie P_x oznacza wartość cechy C_x . Postępując analogicznie należy obliczyć wartość wagi dla neuronu B. Będzie to iloraz wartości dyskryminanta przez sumę wszystkich dyskryminatów wchodzących do neuronu A i B. Można to zapisać następującym wzorem:

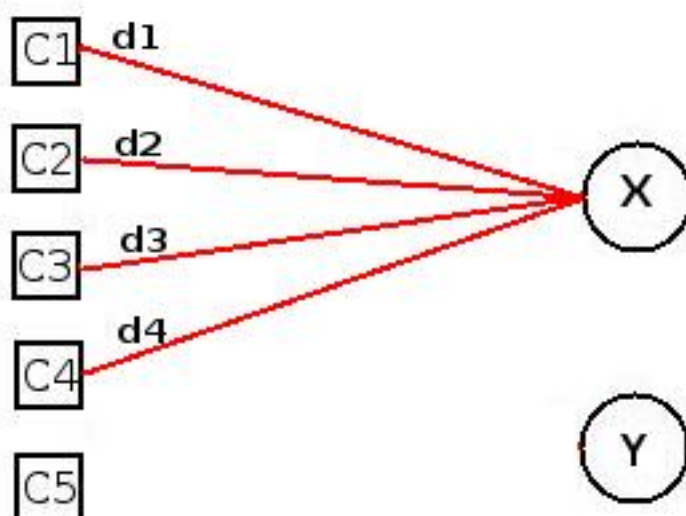
$$W_{AB} = \frac{d2 + d3 + d4}{(d1 + d2 + d3 + d4)} * \left(\frac{d2}{(d2 + d3 + d4)} * P_2 + \frac{d3}{(d2 + d3 + d4)} * P_3 + \frac{d4}{(d2 + d3 + d4)} * P_4 \right) + \frac{d1}{(d1 + d2 + d3 + d4)} * P_1$$

gdzie po skróceniu otrzymujemy

$$W_{AB} = \frac{d2}{(d1 + d2 + d3 + d4)} * P_2 + \frac{d3}{(d1 + d2 + d3 + d4)} * P_3 + \frac{d4}{(d1 + d2 + d3 + d4)} * P_4 + \frac{d1}{(d1 + d2 + d3 + d4)} * P_1 \quad (2.19)$$

Łatwo zauważyć, że powstały wzór jest prostym ilorazem wartości dyskryminanta na obliczanym połączeniu przez sumę wszystkich dyskryminatów wchodzących budowy tej części sieci. W ten sposób można dowolną sieć złożoną z N-warstw przedstawić jako prostą sieć dwuwarstwową, bez względu na ilość epok niezbędnych do prawidłowego działania sieci. Wzór 2.20 został graficznie przedstawiony na rysunku 2.11, na którym wyraźnie widać brak wcześniej występujących (rysunek: 2.9) warstw wewnętrznych.

Na zaprezentowanym przykładzie można wysunąć wnioski, że skomplikowane sposoby przekształceń topologii sieci (np. zamiana miejscami neuronów) opisy w pracy [1] nie ma zastosowania w praktyce. Budując sieć neuronową ilość warstw jest potrzebna tylko by przedstawić wizualizację graficzną powstałej sieci. Do celów informatycznych jak i obliczeniowych użyć można sieci dwuwarstwowej, w sposób tu zaprezentowany. Zamiany neuronów opisane w cytowanej prasie, których celem jest zmniejszenie liczby połączeń występujących w sieci, również są niepotrzebne przy zastosowaniu przedstawionej techniki. Dodatkowo warto zauważyć, że topologię sieci neuronowej można wtedy w prosty sposób przechowywać jako wzory złożone z funkcji wybierających maksimum. Zaprezentowana technika w dużym stopniu ułatwia liczenie wartości poszczególnych neuronów, oszczędzając w znaczny sposób czas, jak i moc obliczeniową maszyn.



Rysunek 2.11: Sieć neuronowa dwu warstwowa

Rozdział 3

Sieci Neuronowe oparte na wielokrotnym obliczaniu dyskryminantów cech

3.1 Obliczanie współczynników

Obliczanie funkcji aktywacji w prezentowanych sieciach neuronowych jest poprzedzone obliczeniem szeregu współczynników. Współczynniki te pomagają ustalić, która cecha najlepiej dyskryminuje dane wzorce. Dzięki temu podział wzorców jest najbardziej optymalny ze względu na ilość użytych neuronów oraz połączeń między nimi. Współczynniki te obliczamy dla każdej z cech oddzielnie.

1. LP – liczba cech dodatnich w danym wierszu Jest to ilość wzorców, które posiadają dodatnie wartości (1) dla obliczanej cechy
2. LM – liczba cech ujemnych w danym wierszu Jest to ilość wzorców, które posiadają ujemne wartości (-1) dla obliczanej cechy
3. S^+ - liczba cech o identycznych znakach, we wszystkich cechach danych klas wejściowych, jak znak dodatni (1) występujący w danym wierszu. Jest do suma wierszy, których wartości wzorców są zgodne co do znaku z wartościami wiersza aktualnie obliczanego. Brane są pod uwagę tylko wzorce z wartością dodatnią (1)
4. S^- - liczba cech o identycznych znakach, we wszystkich cechach danych klas

wejściowych, jak znak ujemny (-1) występujący w danym wierszu. Jest to suma wierszy, których wartości wzorców są zgodne co do znaku z wartościami wiersza aktualnie obliczanego. Brane są pod uwagę tylko wzorce z wartością ujemną (-1)

5. ZOA – współczynnik który decyduje według której cechy będzie występował podział wzorców. Obliczany jest według wzoru:

$$ZOA = S^+(LP - 1) + S^-(LM - 1) \quad (3.1)$$

Współczynnik ten określa, która cecha najlepiej dyskryminuje dane wzorce. Dzięki temu ustalamy cechę, która będzie cechą podziału tablicy wzorców. Wybór cechy, która ma najwyższy współczynnik ZOA gwarantuje nam również taki podział wzorców, a co się z tym wiąże, taką budowę sieci, że ilość neuronów potrzeba do jednoznacznego rozpoznania wzorca będzie minimalna. Dzięki temu ilość zaoszczędzonych połączeń między neuronami będzie największa.

3.2 Budowa Topologii

Po obliczeniu wszystkich współczynników można zbudować topologię sieci. Warto podkreślić, że topologia ta jest uzależniona od rodzajów wzorców wejściowych. Powstaje ona automatycznie po dokonaniu podziału wzorców według cech które posiadają najwyższy współczynnik ZOA . Czyli cecha, która posiada współczynnik ZOA największy dzieli klasy według wartości cech. Może się zdarzyć, że wzorce z pewnej klasy zostaną rozdzielone, czyli przyporządkowane do innych neuronów podziału. Klasy dzielimy według występujących cech -1 i 1 na dwa neurony. Podziału takiego dokonujemy tak długo aż podzielimy jednoznacznie wszystkie występujące wzorce. Po każdym dokonanym podziale następuje wyliczenie od nowa wszystkich współczynników. Cechy które posiadają jednolite wartości dla wszystkich wzorców są pomijane. nie wnoszą one żadnych wartości co do częstości, bądź rzadkości ich występowania w klasach.

Algorytm budowy topologii sieci można przedstawić następująco:

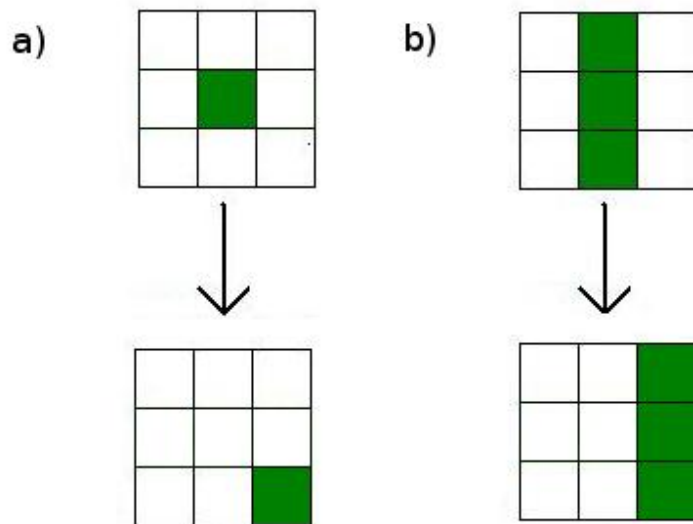
1. Wyznacz współczynniki LP, LM, S^+, S^-, ZOA dla aktualnej tablicy

2. Usuń cechy, które dla wszystkich wzorców z klas posiadają takie same wartości
3. Wybierz wiersz, który posiada najwyższy współczynnik ZOA. Jeśli kilka cech posiada równy, najwyższy współczynnik ZOA wybierz pierwszą
4. Podziel wzorce na dwie grupy. Do jednej zakwalifikuj wzorce, które dla cechy podziału miały wartość $+1$, do drugiej zakwalifikuj wzorce, które dla cechy podziału miały wartość -1 .
5. Wróć do punktu 1) i zastosuj go na każdej z wydzielonych grup.
6. Warunkiem końca jest posiadanie wzorca z określonej jednej klasy w podzielonej grupie

Rozdział 4

Ograniczenia Samo Optymalizujących Sieci Neuronowych

Dużym ograniczeniem tego typu sieci neuronowych jest brak możliwości rozpoznawania wzorców, których położenie na nałożonej siatce cech zmieniło się. Gdyby wyobrazić sobie, że wzorzec uległ przesunięciu na całej płaszczyźnie nałożonej siatki o jedną pozycję, to taki wzorzec najprawdopodobniej nie zostanie prawidłowo rozpoznany. Wynika to z faktu, że sieć neuronowa uczy się występowania określonych wartości wzorca w konkretnych cechach. Zaburzenie tej hierarchii powoduje, że sieć zaczyna traktować zaburzony symbol jako zupełnie nowy wzorzec. W prezentowanych przykładach sieci neuronowych, gdzie rozpatrywane były symbole płaskie na płaszczyźnie X, Y , sieci te dają sobie całkiem dobrze radę z zaburzeniami pojedynczych cech lub nawet zbioru cech. Jednak przesunięcie symbolu o wektor np. $[X, Y] - i \cdot [0, 1]$ spowoduje że większość cech zmieni swoje wartości. Jest to traktowane jak bardzo duże zaburzenie. Dla tego typu sieci tak duże zaburzenia prowadzą do nie rozpoznawania wzorca lub co gorsza, do klasyfikacji wzorca do zupełnie innej klasy. Problem ten przedstawiono na rysunku 4.1. Na rysunku tym przedstawiono dwie proste sytuacje. W sytuacji *a*) przesunięciu uległ punkt. Jednak dla sieci neuronowej jest to zupełnie inny układ wartości cech, i sieć taka nie będzie w stanie zidentyfikować przynależności punktu po przesunięciu do klasy punktu z przed przesunięcia. W sytuacji *b* przesunięciu uległa pionowa linia. Okazuje się, że tutaj sytuacja wygląda dużo go-



Rysunek 4.1: Przesunięcia symboli na siatce.

rzej, gdyż więcej cech zmieniło swoje wartości. Wobec czego zaburzenie wzorców jest tak duże, że sieć nieprawidłowo zakwalifikuje symbol po przesunięciu do właściwej klasy. Jest to zachowanie znacznie odbiegające od pracy ludzkiego mózgu.

Dla ludzkiej sieci neuronów w mózgu nie ma znaczenia czy na kartce papieru określony symbol zostanie narysowany w lewym rogu strony, w jej przawym rogu, czy też na środku. Nie ma również znaczenia wielkość sybolu. Jeśli tylko ludzki mózg odnajdzie wystarczającą liczbę cech, lub ich układów, to jest w stanie odróżnić i zidentyfikować dany symbol. Mało tego, często na rozpoznawalność symbolu nie ma wpływu również jego rotacja. W obecnej formie wzorce testujące, do prezentowanych sieci neuronowych trzeba, dostosowywać do 'rozmiarów' i położenia wzorców uczących. Dlatego chcąc rozpoznać punkt, z rysunku 4.1, dla ludzkiego mózgu nie ma znaczenie jego położenie na siatce. Podobnie sprawa wygląda z rozpoznaniem pionowej linii.

Rozdział 5

Własne badania

Zgodnie z oczekiwaniami szukany jest dyskryminant, który lepiej określi rzadkość występowania danej cechy w innych klasach. Znalezienie takiego dyskryminantu pozwoli określić klasę wzorca za pomocą mniejszej liczby cech. Pozwoli również wybrać najbardziej charakterystyczny układ cech. Najpierw zostaną zaprezentowane przykładowe modyfikacje dyskryminantów, a następnie zostaną wyciągnięte wnioski. Na każdym nowym dyskryminancie przeprowadzone zostały badania które wskazują:

- ile cech (warstw) jest niezbędnych do jednoznacznego rozpoznawania wzorca
- przy ilu cechach (warstwach) wzorzec jest najmniej wrażliwy na zaburzenia
- ile maksymalnie cech można zaburzyć by sieć nadal rozpoznawała wzorce uczące
- jak różne wzorce uczące wpływają na jakość dyskryminanta i wartości powyższych punktów

5.1 Nowy dyskryminant I

$$\forall_{m \in \{1, \dots, M\}} \forall_{u^n \in C^m} \forall_{n \in \{1, \dots, Q\}} \forall_{k \in \{1, \dots, K\}} \left\{ \begin{array}{l} d_k^{n+} = \frac{\hat{P}_k^m}{(M-1)Q^m} \frac{\sum_{\substack{h=1 \\ h \neq m}}^M \binom{N_k^h}{+1}}{\sum_{\substack{h=1 \\ h \neq m}}^M \binom{P_k^h}{+1}} \text{ jeśli } x_k^n > 0 \\ d_k^{n-} = \frac{\hat{N}_k^m}{(M-1)Q^m} \frac{\sum_{\substack{h=1 \\ h \neq m}}^M \binom{P_k^h}{-1}}{\sum_{\substack{h=1 \\ h \neq m}}^M \binom{N_k^h}{-1}} \text{ jeśli } y_k^n > 0 \end{array} \right. \quad (5.1)$$

W tym przypadku modyfikacji uległa druga część wzoru. Różnica zostanie opisana na przykładzie 10 i 1 cechy z Klasy A. Poszukiwana jest taka cecha, która by najlepiej dyskryminowała tę klasę, czyli jako jedyna z wszystkich klas miała wartość inną niż wartość cech w pozostałych klasach, lub aby w innych klasach cech z taką samą wartością było jak najmniej. Gdy sprawdzona zostanie ilość wystąpień cech $+1$ i -1 w pozostałych klasach można wyciągnąć wniosek, że cecha pierwsza nie najlepiej dyskryminuje klasę A. Zauważmy, że cecha 1 w klasie A jest -1 , a w pozostałych klasach cech o wartości -1 jest 5 a o wartości $+1$ jest 13. Można by napisać, że rzadkość występowania tej cechy wynosi $13/5$ czyli $23/5$ Natomiast cecha 10 w klasie A ma wartość -1 , natomiast w pozostałych klasach cech o wartości -1 jest 2 a o wartości $+1$ jest 117. Moglibyśmy zapisać, że rzadkość występowania tej cech wynosi $17/2$ czyli $81/2$ Dla porównania zobaczmy cechę 2, która dla wszystkich wzorców ma wartość 1. Tutaj rzadkość moglibyśmy zapisać jako $0/19$ czyli 0 Ponieważ chodzi tylko o wzmocnienie tych cech które mają znaczenie, sposób ten wydaje się działać prawidłowo. We wzorze proponowanym w pracy [?] przez p. Horzyka cechy 1 i 10 dla klasy A mają równe dyskryminanty, czyli równie dobrze dyskryminują klasę A. W proponowanym wzorze cecha 10 ma wyraźnie większą wartość od 1 cechy. Dyskryminant ten wydaje się poprawiać wybór cech.

5.2 Nowy dyskryminant II

$$\forall_{m \in \{1, \dots, M\}} \forall_{u^n \in C^m} \forall_{n \in \{1, \dots, Q\}} \forall_{k \in \{1, \dots, K\}} \left\{ \begin{array}{l} d_k^{n+} = \frac{P_k^h}{Q^m} \frac{\hat{P}_k^m}{(M-1)Q^m} \frac{\sum_{\substack{h=1 \\ h \neq m}}^M \binom{N_k^h}{P_k^h + 1}}{\sum_{\substack{h=1 \\ h \neq m}}^M \binom{P_k^h}{N_k^h + 1}} \text{jeśli } x_k^n > 0 \\ d_k^{n-} = \frac{N_k^h}{Q^m} \frac{\hat{N}_k^m}{(M-1)Q^m} \frac{\sum_{\substack{h=1 \\ h \neq m}}^M \binom{P_k^h}{N_k^h + 1}}{\sum_{\substack{h=1 \\ h \neq m}}^M \binom{N_k^h}{P_k^h + 1}} \text{jeśli } y_k^n > 0 \end{array} \right. \quad (5.2)$$

5.3 Nowy dyskryminant III

$$\forall_{m \in \{1, \dots, M\}} \forall_{u^n \in C^m} \forall_{n \in \{1, \dots, Q\}} \forall_{k \in \{1, \dots, K\}} \left\{ \begin{array}{l} d_k^{n+} = \frac{P_k^h}{Q^m} \frac{P_k^h + N_k^h}{Q^m} \frac{\hat{P}_k^m}{(M-1)Q^m} \frac{\sum_{\substack{h=1 \\ h \neq m}}^M \binom{N_k^h}{P_k^h + 1}}{\sum_{\substack{h=1 \\ h \neq m}}^M \binom{P_k^h}{N_k^h + 1}} \text{jeśli } x_k^n > 0 \\ d_k^{n-} = \frac{N_k^h}{Q^m} \frac{P_k^h + N_k^h}{Q^m} \frac{\hat{N}_k^m}{(M-1)Q^m} \frac{\sum_{\substack{h=1 \\ h \neq m}}^M \binom{P_k^h}{N_k^h + 1}}{\sum_{\substack{h=1 \\ h \neq m}}^M \binom{N_k^h}{P_k^h + 1}} \text{jeśli } y_k^n > 0 \end{array} \right. \quad (5.3)$$

5.4 Nowy dyskryminant IV

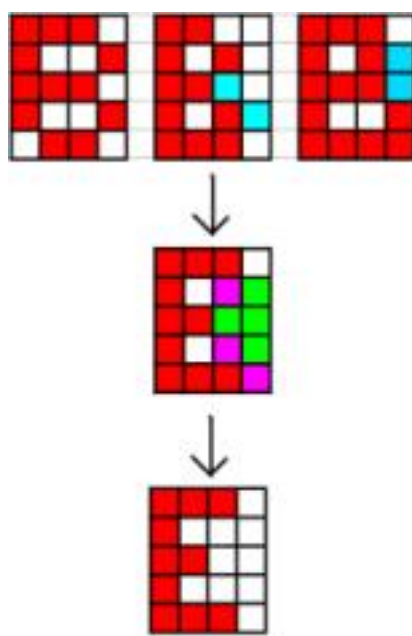
$$\forall_{m \in \{1, \dots, M\}} \forall_{u^n \in C^m} \forall_{n \in \{1, \dots, Q\}} \forall_{k \in \{1, \dots, K\}} \left\{ \begin{array}{l} d_k^{n+} = \frac{P_k^h}{Q^m} \frac{\hat{P}_k^m}{(M-1)Q^m} \sum_{\substack{h=1 \\ h \neq m}}^M \left(1 - \frac{\hat{P}_k^h}{Q^h}\right) \text{jeśli } x_k^n \neq 0 \\ d_k^{n-} = \frac{N_k^h}{Q^m} \frac{\hat{N}_k^m}{(M-1)Q^m} \sum_{\substack{h=1 \\ h \neq m}}^M \left(1 - \frac{\hat{N}_k^h}{Q^h}\right) \text{jeśli } y_k^n \neq 0 \end{array} \right. \quad (5.4)$$

5.5 Nowy dyskryminant V

$$\forall_{m \in \{1, \dots, M\}} \forall_{u^n \in C^m} \forall_{n \in \{1, \dots, Q\}} \forall_{k \in \{1, \dots, K\}} \left\{ \begin{array}{l} d_k^{n+} = \frac{P_k^h}{Q^m} \frac{P_k^h + N_k^h}{Q^m} \frac{\hat{P}_k^m}{(M-1)Q^m} \sum_{\substack{h=1 \\ h \neq m}}^M \left(1 - \frac{\hat{P}_k^h}{Q^h}\right) \text{jeśli } x_k^n \neq 0 \\ d_k^{n-} = \frac{N_k^h}{Q^m} \frac{P_k^h + N_k^h}{Q^m} \frac{\hat{N}_k^m}{(M-1)Q^m} \sum_{\substack{h=1 \\ h \neq m}}^M \left(1 - \frac{\hat{N}_k^h}{Q^h}\right) \text{jeśli } y_k^n \neq 0 \end{array} \right. \quad (5.5)$$

5.6 Wnioski

Okazuje się, że zbyt duże uogólnianie powoduje pogorszenie wyników sieci neuronowej. Za przykład niech posłuży klasa posiadająca trzy wzorce. Wzorec pierwszy posiada cechy określone jako 1 i -1 . Wzorec drugi również posiada cechy o określonych wartościach lecz wartości niektórych cech różnią się, w porównaniu z wzorcem pierwszym. Wzorec trzeci natomiast, posiada kilka cech o nie ustalonej wartości czyli 0. Dyskryminant konstruujemy w taki sposób, że będzie pomniejszał wartość cech które nie mają takiej samej wartości dla określonej cechy lub któraś z cech ma nieustaloną wartość. Niejako cechy takie od razu odrzucamy. Powoduje to zmniejszenie ilości cech po których możemy identyfikować wzorec. Przyjmijmy chwilowo skrajne obniżenie wartości dyskryminanta dla cech które różnią w kolejnych wzorcach klasy do wartości 0. Taką sytuację przedstawiono na rysunku 5.1. Cechy, które



Rysunek 5.1: Obniżenie znaczenia cech prowadzi do zubożenia wzorców uczących przestają się liczyć w konstrukcji sieci neuronowej przedstawiono kolorami zielonym i fioletowym. Na zielono zaznaczone zostały cechy, które nie są brane pod uwagę ponieważ w jednym z wzorców cecha ta ma nie ustaloną wartość równą 0. Natomiast kolor fioletowy cechy oznacza, że w jednym ze wzorców danej klasy cecha ta ma inną wartość. Jeśli w skrajnym przypadku dyskryminanty tych cech zostaną pomniejszone do wartości równej zero, lub bliskiej zero, a co za tym idzie nie zostaną uwzględnione do topologii sieci, to wzorce stają się ubogie w cechy. Na rysunku

tym pokazano, że tak drastyczne obniżenie wartości dyskryminantów, w sytuacji gdy chcemy znaleźć układ cech dla całej klasy, a nie dla wzorca, może spowodować, że 'braknie' odpowiednio dobrych cech do rozpoznawania wzorca. Czyli może się zdarzyć sytuacja, że sieć będzie zmuszona do rozpoznania wzorca o symbolu 'B' za pomocą cech idealnie pasujących do symbolu 'E', co łatwo zauważyć na rysunku 5.1.

Bibliografia

- [1] A. Horzyk, R. Tadeusiewicz, *Self-Optimizing Neural Networks. A New Perspective in Analyzing Training Data and Building Neural Models*
- [2] A. Horzyk, R. Tadeusiewicz, *Self-Optimizing Neural Networks*
- [3] A. Horzyk *Unsupervised Clustering using Self-Optimizing Neural Networks*