

## **Wprowadzenie do sieci neuronowych**

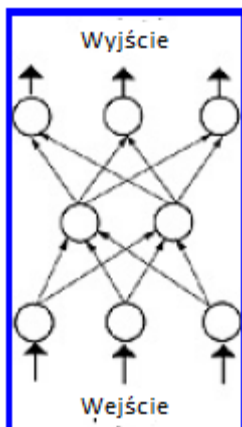
### **Przetwarzanie neuronowe**

Jak rozpoznajesz twarz w tłumie? Jak ekonomista przewiduje kierunek stóp procentowych? W obliczu takich problemów ludzki mózg wykorzystuje sieć połączonych elementów przetwarzania zwanych neuronami do przetwarzania informacji. Każdy neuron jest autonomiczny i niezależny; wykonuje swoją pracę asynchronicznie, to znaczy bez synchronizacji z innymi zachodzącymi zdarzeniami. Dwa przedstawione problemy, a mianowicie rozpoznawanie twarzy i prognozowanie stóp procentowych, mają dwie ważne cechy, które odróżniają je od innych problemów: Po pierwsze, problemy są złożone, to znaczy nie można opracować prostego algorytmu krok po kroku lub precyzyjnego wzoru, aby dać ci odpowiedź; po drugie, dane dostarczone w celu rozwiązania problemów są również złożone i mogą być głośnie lub niekompletne. Mogłeś zapomnieć o okularach, gdy próbujesz rozpoznać tę twarz. Ekonomista może dysponować tysiącami danych, które mogą lub nie mogą być istotne dla jego prognozy dotyczącej gospodarki i stóp procentowych. Ogromna moc przetwarzania tkwiąca w biologicznych strukturach neuronowych zainspirowała badania samej struktury pod kątem wskazówek dotyczących organizacji struktur komputerowych stworzonych przez człowieka. Sztuczne sieci neuronowe, , omawiają sposób organizowania syntetycznych neuronów w celu rozwiązania tego samego rodzaju trudnych, złożonych problemów w podobny sposób, jak sądzimy, że ludzki mózg może. Ta część zawiera próbki terminów i nazewnictwa używanych do mówienia o sieciach neuronowych. Terminy te zostaną omówione bardziej szczegółowo w kolejnych częściach.

### **Sieć neuronowa**

Sieć neuronowa jest strukturą obliczeniową inspirowaną badaniami biologicznego przetwarzania neuronowego. Istnieje wiele różnych typów sieci neuronowych, od stosunkowo prostych do bardzo złożonych, podobnie jak istnieje wiele teorii na temat działania biologicznego przetwarzania neuronowego. Zaczniemy od omówienia warstwowego typu sieci neuronowej z wyprzedzeniem i omówimy inne paradygmaty w dalszej części sekcji oraz w innych częściach. Warstwowa przednia sieć neuronowa ma warstwy lub podgrupy elementów przetwarzających. Warstwa elementów przetwarzających wykonuje niezależne obliczenia na danych, które otrzymuje i przekazuje wyniki do innej warstwy. Kolejna warstwa może z kolei wykonywać swoje niezależne obliczenia i przekazywać wyniki do kolejnej warstwy. Wreszcie podgrupa jednego lub większej liczby elementów przetwarzających określa wyjście z sieci. Każdy element przetwarzający dokonuje obliczeń na podstawie ważonej sumy danych wejściowych. Pierwsza warstwa to warstwa wejściowa, a ostatnia warstwa wyjściowa. Warstwy umieszczone między pierwszą a ostatnią warstwą są warstwami ukrytymi. Elementy przetwarzania są postrzegane jako jednostki podobne do neuronów w ludzkim mózgu, a zatem są nazywane komórkami, neuromimami lub sztuczne neurony. Funkcja progowa jest czasami używana do zakwalifikowania wyjścia neuronu w warstwie wyjściowej. Nawet jeśli nasza tematyka dotyczy sztucznych neuronów, będziemy po prostu nazywać je neuronami. Synapsy między neuronami określa się jako połączenia, które są reprezentowane przez krawędzie ukierunkowanego wykresu, w którym węzły są sztucznymi neuronami.

Rysunek



to wielowarstwowa sieć neuronowa. Okrągłe węzły reprezentują neurony. Są tu trzy warstwy, warstwa wejściowa, warstwa ukryta i warstwa wyjściowa. Wymieniony wykres pokazuje połączenia z węzłów z danej warstwy do innych węzłów w innych warstwach. W tej książce zobaczysz wiele odmian liczby i typów warstw.

### Wyjście neuronu

Zasadniczo, wewnętrzna aktywacja lub surowe wyjście neuronu w sieci neuronowej jest niesłyszana sumą jego wejść, ale funkcja progowa jest również używana do określenia końcowej wartości lub wyjścia. Gdy wynik wynosi 1, mówi się, że neuron wystrzeliwuje, a gdy wynosi 0, neuron uważa się za nie wystrzelony. Gdy używana jest funkcja progowa, różne wyniki aktywacji, wszystkie w tym samym przedziale wartości, mogą powodować tę samą końcową wartość wyjściową. Ta sytuacja pomaga w tym sensie, że jeśli precyzyjne wejście powoduje aktywację 9, a zaszumione wejście powoduje aktywację 10, to wyjście działa tak samo, jak gdyby szum był filtrowany. Aby opisać sieć neuronową w prostym i znanym otoczeniu, opiszmy przykład o teleturnieju telewizyjnym „The Price is Right”.

### Kasa Gra

Opisano kilka produktów, ich ceny nie są znane zawodnikowi, a uczestnik musi zadeklarować, ile jednostek każdego przedmiotu chciałby (udawać) kupić. Jeśli łączny zakup nie przekracza określonej kwoty, uczestnik wygrywa nagrodę specjalną. Po tym, jak zawodnik ogłosi, ile przedmiotów danego produktu chce, cena tego produktu zostaje ujawniona i jest umieszczana na kasie. Zawodnik musi uważać, w tym przypadku, aby suma nie przekracza pewnej wartości nominalnej, aby zdobyć związaną z nią nagrodę. Możemy teraz spojrzeć na całą operację tej gry, pod względem sieci neuronowej, zwanej Perceptronem, w następujący sposób. Rozważ każdy produkt na półce jako neuron w warstwie wejściowej, a jego wejście to cena jednostkowa tego produktu. Kasa fiskalna to pojedynczy neuron w warstwie wyjściowej. Jedyne połączenia w sieci znajdują się między każdym z neuronów (produktów wyświetlanych na półce) w warstwie wejściowej a neuronem wyjściowym (w kasie). Taki układ jest zwykle nazywany neuronem, w tym przypadku kasą fiskalną, będącą instancją w terminologii sieci neuronowej. Zawodnik faktycznie określa te połączenia, ponieważ gdy zawodnik mówi, że chce, powiedzmy, pięć jednostek, określonego produktu, zawodnik przypisuje w ten sposób wagę 5 do połączenia między tym produktem a kasą. Całkowity rachunek za zakupy przez uczestnika jest niczym innym jak ważoną sumą cen jednostkowych różnych oferowanych produktów. W przypadku przedmiotów, których uczestnik nie zdecyduje się na zakup, przypisana domyślna waga wynosi 0. Zastosowanie limitu dolara do rachunku jest po prostu zastosowaniem progu, z wyjątkiem tego, że wartość progowa nie powinna zostać przekroczona dla wyniku z tej sieci faworyzować zawodnika,

wygrywając mu dobrą nagrodę. W Perceptronie sposób działania progów polega na tym, że neuron wyjściowy ma strzelać, jeśli jego wartość aktywacji przekracza wartość progową

### **Wagi**

Wagi stosowane w połączeniach między różnymi warstwami mają duże znaczenie w działaniu sieci neuronowej i charakteryzowaniu sieci. Następujące działania są możliwe w sieci neuronowej:

1. Zaczynaj od jednego zestawu wag i uruchom sieć. (NIE PADA)
2. Zaczynaj od jednego zestawu wag, uruchom sieć i zmodyfikuj niektóre lub wszystkie wagi, a następnie uruchom sieć ponownie z nowym zestawem wag. Powtarzaj ten proces, aż zostanie osiągnięty określony cel. (TRENING)

### **Trening**

Ponieważ dane wyjściowe mogą nie być zgodne z oczekiwaniami, waga może wymagać zmiany. Niektóre reguły muszą być wtedy użyte, aby określić, jak zmienić wagi. Powinno również istnieć kryterium określające, kiedy kończy się proces kolejnych modyfikacji wag. Ten proces zmiany wag, a raczej aktualizowania wag, nazywa się treningiem. Mówi się, że sieć, w której wykorzystuje się naukę, podlega szkoleniu. Szkolenie to proces zewnętrzny lub schemat. Nauka jest pożądanym procesem, który odbywa się wewnątrz sieci.

### **Sprężenie zwrotne**

Jeśli chcesz wyszkolić sieć, aby mogła rozpoznać lub zidentyfikować niektóre z góry określone wzorce lub ocenić niektóre wartości funkcji dla danych argumentów, ważne byłoby, aby informacje były przekazywane z neuronów wyjściowych do neuronów w jakiejś warstwie, aby umożliwić dalsze przetwarzanie i regulacja odważników na połączeniach. Taka informacja zwrotna może dotyczyć warstwy wejściowej lub warstwy między warstwą wejściową a warstwą wyjściową, czasami oznaczoną warstwą ukrytą. To, co jest sprzężone zwrotnie, to zazwyczaj błąd na wyjściu, odpowiednio zmodyfikowany zgodnie z pewnym użytecznym paradygmatem. Proces sprzężenia zwrotnego trwa przez kolejne cykle działania sieci neuronowej i ustaje po zakończeniu szkolenia.

### **Nauka nadzorowana lub nienadzorowana**

Sieć może podlegać nauce nadzorowanej lub nienadzorowanej. Nauka byłaby nadzorowana, gdyby kryteria zewnętrzne były wykorzystywane i dopasowane przez wyjście sieciowe, a jeśli nie, uczenie się nie jest nadzorowane. Jest to jeden szeroki sposób na podzielenie różnych podejść sieci neuronowych. Podejścia nienadzorowane są również określane jako samoorganizujące się. Istnieje więcej interakcji między neuronami, zazwyczaj ze sprzężeniem zwrotnym i połączeniami między neuronami, promującymi samoorganizację. Sieci nadzorowane są nieco prostsze do konceptualizacji niż sieci bez nadzoru. Wprowadzasz dane wejściowe do nadzorowanej sieci wraz z oczekiwaną odpowiedzią, podobnie jak warunkowy bodziec i reakcja Pawłowa. Ty formujesz sieć z parami bodziec-odpowiedź. Forecaster giełdowy może przedstawić dane ekonomiczne (bodziec) wraz z parametrami wydajności rynku akcji (odpowiedź) do sieci neuronowej do chwili obecnej i próbować przewidzieć przyszłość po zakończeniu szkolenia. Zapewniaś sieci bez nadzoru z tylko bodźcami. Możesz na przykład chcieć, aby sieć bez nadzoru poprawnie klasyfikowała części z taśmy przenośnika na numery części, dostarczając obraz każdej części w celu dokonania klasyfikacji (bodziec). Bez nadzoru sieć w tym przypadku działałaby jak pamięć przeglądowa indeksowana przez jej zawartość lub pamięć adresowalna treści (CAM)

### **Hałas**

Hałas jest zaburzeniem lub odchyleniem od rzeczywistości. Zestaw danych używany do trenowania sieci neuronowej może mieć w sobie szum wewnętrzny lub obraz może mieć na przykład losowe plamki. Odpowiedź sieci neuronowej na szum jest ważnym czynnikiem przy określaniu jej przydatności do danego zastosowania. W procesie szkolenia możesz zastosować metrykę do sieci neuronowej, aby sprawdzić, jak dobrze sieć nauczyła się danych treningowych. W przypadkach, gdy metryka ustabilizuje się do pewnej znaczącej wartości, bez względu na to, czy wartość jest dla Ciebie akceptowalna, czy nie, mówisz, że sieć zbiega się. Możesz wprowadzić szum celowo w szkoleniu, aby dowiedzieć się, czy sieć może nauczyć się o istnieniu szumu i czy sieć może zbiegać się w hałaśliwych danych.

## **Pamięć**

Po szkoleniu sieci na zestawie danych założmy, że kontynuujesz szkolenie sieci z nowymi danymi. Czy sieć zapomni zaplanowane szkolenie na oryginalnym zestawie, czy będzie pamiętać? Jest to kolejny problem, do którego podchodzą niektórzy badacze zainteresowani zachowaniem pamięci długoterminowej sieci (LTM), jak również jej pamięci krótkoterminowej (STM). Pamięć długoterminowa to pamięć związana z uczeniem się, która utrzymuje się przez długi czas. Pamięć krótkotrwała to pamięć związana z zanikającą siecią neuronową w pewnym przedziale czasu.

## **Kapsuła historii**

Zachwycasz się możliwościami ludzkiego mózgu i odkrywasz, że jego sposoby przetwarzania informacji są w dużym stopniu nieznanne. Niesamowite jest to, że bardzo złożone sytuacje są dostrzegane z dużo większą prędkością niż komputer. Warren McCulloch i Walter Pitts sformułowali w 1943 r. Model komórki nerwowej, neuronu, podczas próby zbudowania teorii systemów samoorganizujących się. Później Frank Rosenblatt skonstruował Perceptron, układ elementów przetwarzających reprezentujących komórki nerwowe w sieć. Jego sieć potrafiła rozpoznać proste kształty. To było pojawienie się różnych modeli dla różnych zastosowań. Ci, którzy pracują w dziedzinie sztucznej inteligencji (AI), próbowali postawić hipotezę, że można modelować procesy myślowe za pomocą niektórych symboli i pewnych reguł, za pomocą których można przekształcić symbole. Ograniczenie podejścia symbolicznego jest związane z tym, jak reprezentowana jest wiedza. Część informacji jest zlokalizowana, to znaczy, być może dostępna w jednym miejscu. Nie jest rozłożony na wiele lokalizacji. Łatwo zauważyć, że wiedza rozproszona prowadzi do szybszego i większego procesu wnioskowania. Informacje są mniej podatne na uszkodzenie lub utratę, gdy są dystrybuowane, niż gdy są zlokalizowane. Rozproszone przetwarzanie informacji może być do pewnego stopnia odporne na błędy, ponieważ istnieje wiele źródeł wiedzy, które można zastosować do danego problemu. Nawet jeśli jedno źródło zostanie odcięte lub zniszczone, inne źródła mogą nadal pozwolić na rozwiązanie problemu. Co więcej, dzięki późniejszemu uczeniu się, można przekształcić rozwiązanie nowej organizacji elementów przetwarzania rozproszonego, które wykluczają wadliwy element przetwarzający. W sieciach neuronowych informacja może wpływać na aktywność więcej niż jednego neuronu. Wiedza jest rozpowszechniana i łatwo daje się jej obliczyć równolegle. Rzeczywiście istnieje wiele działań badawczych w dziedzinie projektowania sprzętu neuronowego, które wykorzystują paradygmat sieci neuronowej. Carver Mead, pionier w tej dziedzinie, zaproponował analogowe implementacje obwodów sieci neuronowych VLSI (integracja na dużą skalę).

## **Budowa sieci neuronowej**

Istnieją trzy aspekty budowy sieci neuronowej:

1. Struktura - architektura i topologia sieci neuronowej
2. Kodowanie - metoda zmiany wagi

### 3. Pobieranie - metoda i zdolność do pobierania informacji

Omówmy pierwszą - strukturę. Odnosi się to do tego, ile warstw powinna zawierać sieć i jakie są ich funkcje, takie jak dane wejściowe, wyjściowe lub ekstrakcja funkcji. Struktura obejmuje również sposób tworzenia połączeń między neuronami w sieci i ich funkcjami. Drugi aspekt to kodowanie. Kodowanie odnosi się do paradygmatu używanego do określenia i zmiany wag na połączeniach między neuronami. W przypadku wielowarstwowej przedniej sieci neuronowej można początkowo zdefiniować wagi przez randomizację. Następnie w procesie szkolenia można użyć algorytmu wstecznej propagacji, który jest sposobem aktualizowania wag począwszy od wyjścia wstecz. Po zakończeniu treningu wielowarstwowej sieci neuronowej z transmisją do przodu kończy się kodowanie, ponieważ wagi nie zmieniają się po zakończeniu treningu. Wreszcie, odwołanie jest również ważnym aspektem sieci neuronowej. Pobieranie odnosi się do uzyskania oczekiwanego wyjścia dla danego wejścia. Jeśli to samo wejście, co poprzednio, jest prezentowane w sieci, powinno wynikać to samo wyjście, co poprzednio. Rodzaj wycofania może scharakteryzować sieć jako autoasocjacyjną lub hetero-związkową. Automatyczna asocjacja to zjawisko kojarzenia wektora wejściowego ze sobą jako wyjścia, podczas gdy heteroasocjacja polega na przywoływaniu pokrewnego wektora z podaniem wektora wejściowego. Masz niewyraźną pamięć o numerze telefonu. Na szczęście zapisałeś go w autoasocjacyjnej sieci neuronowej. Kiedy zastosujesz niewyraźną pamięć, otrzymasz rzeczywisty numer telefonu. To jest użycie automatycznego powiązania. Jeśli chcesz, aby imię i nazwisko osoby było powiązane z danym numerem telefonu, wymagałoby to powiązania hetero. Przypomnijmy, że jest on ściśle związany z wprowadzonymi wcześniej koncepcjami STM i LTM. Wymienione powyżej trzy aspekty budowy sieci neuronowej zasadniczo rozróżniają różne sieci neuronowe i są częścią ich procesu projektowania

#### **Przykładowe aplikacje**

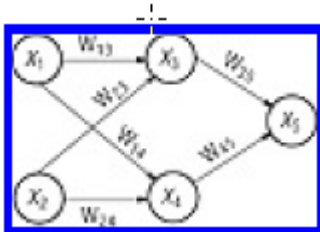
Jedną aplikacją dla sieci neuronowej jest klasyfikacja wzorców lub dopasowywanie wzorców. Wzory mogą być reprezentowane przez cyfry binarne w dyskretnych przypadkach lub liczby rzeczywiste reprezentujące sygnały analogowe w ciągłych przypadkach. Klasyfikacja wzorców jest formą ustanowienia auto-asocjacji lub heteroasocjacji. Przypomnij sobie, że kojarzenie różnych wzorców buduje rodzaj powiązania zwanego heteroasocjacją. Jeśli wprowadzisz uszkodzony lub zmodyfikowany wzorec A do sieci neuronowej i otrzymasz prawdziwy wzorec A, jest to nazywane automatycznym powiązaniem. Jak to zapewnia? Zapamiętaj przykład podany na początku tej części. Na przykładzie ludzkiego mózgu powiedz, że chcesz przypomnieć sobie twarz w tłumie i masz mgliste wspomnienie (wejście). To, czego chcesz, to rzeczywisty obraz. Automatyczna asocjacja jest zatem przydatna w rozpoznawaniu lub pobieraniu wzorców z potencjalnie niepełnymi informacjami jako danymi wejściowymi. Co z heteroasocjacją? Tutaj kojarzysz A z B. Dajesz A, dostajesz B, a czasami odwrotnie. Możesz na przykład zapisać twarz osoby i pobrać ją z imieniem i nazwiskiem osoby. W rzeczywistych okolicznościach dość często robi się coś przeciwnego, a czasem nie tak dobrze. Przypominasz sobie twarz osoby, ale nie możesz umieścić jej nazwy.

#### **Kwalifikacja do kredytu hipotecznego**

Inną przykładową aplikacją, która w rzeczywistości znajduje się w pracach agencji rządowej USA, jest opracowanie sieci neuronowej, aby uzyskać szybki rating kredytowy osoby starającej się o kredyt hipoteczny. Dotychczasowym problemem związanym z procesem składania wniosku o kredyt hipoteczny była oszałamiająca ilość dokumentów i danych wymaganych do złożenia wniosku. Po zebraniu informacji czas odpowiedzi na informację, czy kredyt hipoteczny został zatwierdzony, zwykle wynosił kilka tygodni. Wszystko to się zmieni. Proponowany system sieci neuronowych pozwoli na pełne przeprowadzenie procesu zatwierdzania i zatwierdzania przez trzy godziny, a zatwierdzenie

nastąpi w ciągu pięciu minut od wprowadzenia wszystkich wymaganych informacji. Wprowadzasz historię zatrudnienia wnioskodawcy, informacje o wynagrodzeniu, informacje o kredytach i inne czynniki i stosujesz je do wyszkolonej sieci neuronowej. Sieć neuronowa, oparta na wcześniejszym szkoleniu w tysiącach historii przypadków, szuka wzorców w profilu wnioskodawcy, a następnie tworzy ocenę tak lub nie, która jest godna przeprowadzenia określonej hipoteki. Kontynuujemy teraz dyskusję na temat czynników, które odróżniają modele sieci neuronowych od siebie.

#### Przykład - sieć przekazywania danych



Przykładowa sieć przekazywania do przodu, ma pięć neuronów rozmieszczonych w trzech warstwach: dwa neurony (oznaczone  $x_1$  i  $x_2$ ) w warstwie 1, dwa neurony (oznaczone  $x_3$  i  $x_4$ ) w warstwie 2 i jeden neuron (oznaczone etykietą  $x_5$ ) w warstwie 3. Są strzałki łączące neurony razem. To jest kierunek przepływu informacji. Sieć przekazująca informacje ma tylko informacje przepływające do przodu. Każda strzałka łącząca neurony ma przypisaną jej wagę (na przykład  $w_{31}$ ). Obliczasz stan,  $x$ , każdego neuronu, sumując ważone wartości, które wpływają do neuronu. Stan neuronu jest wartością wyjściową neuronu i pozostaje taki sam, dopóki neuron nie otrzyma nowych informacji o swoich wejściach.

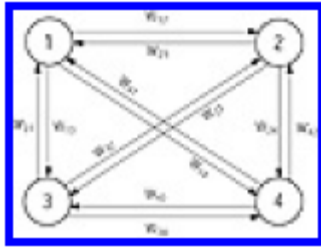
Na przykład dla  $x_3$  i  $x_5$ :

$$x_3 = w_{23} x_2 + w_{13} x_1$$

$x_5 = w_{35} x_3 + w_{45} x_4$  w części 7, w którym wyszczególniono jeden z algorytmów szkoleniowych dla sieci sprzężenia zwrotnego o nazwie Backpropagation. Zauważ, że prezentujesz informacje do tej sieci w skrajnych lewych węzłach (warstwa 1) nazywanych warstwą wejściową. Możesz pobierać informacje z dowolnej innej warstwy w sieci, ale w większości przypadków robi to z prawego węzła (węzłów), które tworzą warstwę wyjściową. Wagi są zwykle określane przez nadzorowany algorytm treningowy, w którym podajesz przykłady do sieci i odpowiednio dostosowujesz wagi, aby uzyskać pożądaną odpowiedź. Po ukończeniu szkolenia możesz korzystać z sieci bez zmiany ciężarów i zanotować odpowiedź na wprowadzone dane. Zauważ, że szczegół jeszcze nie pokazany jest nieliniową funkcją skalowania, która ogranicza zakres ważonej sumy. Ta funkcja skalowania działa przez obcinanie bardzo dużych wartości w kierunkach dodatnich i ujemnych dla każdego neuronu, tak aby sumowanie sumujące występujące w sieci pozostało w rozsądnych granicach. Typowe zakresy liczb rzeczywistych dla wejść i wyjść neuronów to  $-1$  do  $+1$  lub  $0$  do  $+1$ . Więcej informacji na temat tej sieci i aplikacji można znaleźć w części 7. Teraz porównamy tę sieć neuronową z zupełnie innym typem sieci neuronowej, siecią Hopfield i przedstawimy kilka prostych aplikacji dla sieci Hopfield.

#### Przykład - sieć Hopfield

Sieć neuronowa, którą prezentujemy, jest siecią Hopfielda z pojedynczą warstwą. Umieszczamy w tej warstwie cztery neurony, każdy połączony z resztą.



Niektóre połączenia mają masę dodatnią, a reszta ma masę ujemną. Sieć będzie prezentowana z dwoma wzorcami wejściowymi, po jednym na raz, i ma je przywoływać. Wejścia byłyby wzorami binarnymi mającymi w każdym składniku 0 lub 1. Jeśli podane są dwa wzorce o równej długości i traktowane są jako wektory, ich produkt punktowy jest uzyskiwany najpierw przez pomnożenie odpowiednich składników razem, a następnie dodanie tych produktów. Mówi się, że dwa wektory są ortogonalne, jeśli ich iloczyn punktowy wynosi 0. Matematyka zaangażowana w obliczenia wykonane dla sieci neuronowych obejmuje mnożenie macierzy, transponowanie macierzy i transponowanie wektora. Dwa wzorce, które chcemy przywołać do sieci, to  $A = (1, 0, 1, 0)$  i  $B = (0, 1, 0, 1)$ , które można zweryfikować jako ortogonalne. Przypomnij sobie, że dwa wektory  $A$  i  $B$  są ortogonalne, jeśli ich iloczyn punktowy jest równy zero. Jest tak w tym przypadku, ponieważ  $A_1B_1 + A_2B_2 + A_3B_3 + A_4B_4 = (1 \times 0 + 0 \times 1 + 1 \times 0 + 0 \times 1) = 0$

Poniższa macierz  $W$  podaje wagi połączeń w sieci.

$$W = \begin{matrix} & \begin{matrix} 0 & -3 & 3 & -3 \end{matrix} \\ \begin{matrix} -3 \\ -3 \\ 3 \\ -3 \end{matrix} & \begin{matrix} 0 & -3 & -3 & 3 \\ 3 & -3 & 0 & -3 \\ -3 & 3 & -3 & 0 \end{matrix} \end{matrix}$$

Potrzebujemy również funkcji progowej i definiujemy ją w następujący sposób. Wartość progowa 0 wynosi 0.

$$f(t) = \begin{cases} 1 & \text{if } t \geq [\text{theta}] \\ 0 & \text{if } t < [\text{theta}] \end{cases}$$

Mamy cztery neurony w jedynej warstwie w tej sieci. Musimy obliczyć aktywację każdego neuronu jako ważonej sumy jego wejść. Aktywacja w pierwszym węźle jest iloczynem kropkowym wektora wejściowego i pierwszej kolumny macierzy wagowej  $(0 \ -3 \ 3 \ -3)$ . Podobnie aktywujemy pozostałe węzły. Wyjście neuronu jest następnie obliczane przez ocenę funkcji progowej przy aktywacji neuronu. Jeśli więc przedstawimy wektor wejściowy  $A$ , iloczyn punktowy działa na 3 i  $f(3) = 1$ . Podobnie, otrzymujemy produkty kropkowe drugiego, trzeciego i czwartego węzła jako  $-6, 3$  i  $-6$ , odpowiednio. Odpowiednimi wyjściami są zatem 0, 1 i 0. Oznacza to, że wyjście sieci to wektor  $(1, 0, 1, 0)$ , taki sam jak wzorec wejściowy. Sieć przywołała wzorec jako przedstawiony lub możemy powiedzieć, że wzorec  $A$  jest stabilny, ponieważ wyjście jest równe wejściu. Gdy przedstawiono  $B$ , produkt punktowy

uzyskany w pierwszym węźle wynosi  $-6$ , a wynik wynosi  $0$ . Wyjścia dla pozostałych węzłów razem z wyjściem pierwszego węzła dają  $(0, 1, 0, 1)$ , co oznacza, że sieć ma również stabilne wycofanie dla B. Do tej pory przedstawiliśmy sieciom proste przypadki - wektory, które sieć Hopfield została specjalnie zaprojektowana (poprzez wybór matrycy wagowej) do przypomnienia. Co da sieć jako wyjście, jeśli przedstawimy wzór inny niż A i B? Niech  $C = (0, 1, 0, 0)$  zostanie przedstawione w sieci. Aktywacje będą wynosiły  $-3, 0, -3, 3$ , dzięki czemu wyjścia  $0, 1, 0, 1$ , co oznacza, że B osiąga stabilne przywołanie. To jest dość interesujące. Załóżmy, że zamierzaliśmy wprowadzić B i popełniliśmy niewielki błąd, a zamiast tego pokazaliśmy C. Sieć zrobiła to, co chcieliśmy i przypomniała sobie B. Ale dlaczego nie? Aby odpowiedzieć na to pytanie, pytajmy C bliżej A lub B? Jak się porównujemy? Używamy wzoru odległości dla dwóch punktów czterowymiarowych. Jeśli  $(a, b, c, d)$  i  $(e, f, g, h)$  są dwoma punktami czterowymiarowymi, odległość między nimi wynosi:

$$\sqrt{(a - e)^2 + (b - f)^2 + (c - g)^2 + (d - h)^2}$$

Odległość między A i C wynosi [radycznie]  $3$ , podczas gdy odległość między B i C wynosi zaledwie  $1$ . Ponieważ B jest bliżej w tym sensie, B został przywołany, a nie A. Możesz sprawdzić, czy wykonamy to samo z  $D = (0, 0, 1, 0)$ , zobaczymy, że sieć przywołuje A, czyli bliżej niż B do D.

### Odległość Hamminga

Kiedy mówimy o bliskości wzoru bitowego z innym wzorem bitowym, odległość euklidesowa nie musi być brana pod uwagę. Zamiast tego można zastosować odległość Hamminga, co jest znacznie łatwiejsze do określenia, ponieważ jest to liczba pozycji bitów, w których dwa porównywane wzorce różnią się. Wzory będące ciągami, odległość Hamminga jest bardziej odpowiednia niż odległość euklidesowa.

### Aktualizacja asynchroniczna

Sieć Hopfielda jest siecią cykliczną. Oznacza to, że wyjścia z sieci są zwracane jako wejścia. Sieć Hopfield zawsze stabilizuje się do ustalonego punktu. Aby osiągnąć tę stabilność, istnieje bardzo ważny szczegół dotyczący sieci Hopfield. W dotychczasowych przykładach nie mieliśmy problemu z uzyskaniem stabilnego wyjścia z sieci, więc nie przedstawiliśmy tego szczegółu działania sieci. Ten szczegół to potrzeba asynchronicznej aktualizacji sieci. Oznacza to, że zmiany nie zachodzą równocześnie z wyjściami, które są zwracane jako dane wejściowe, ale występują dla jednego komponentu wektorowego na raz. Prawdziwe działanie sieci Hopfield postępuje zgodnie z poniższą procedurą dla wektora wejściowego Invec i wektora wyjściowego Outvec:

1. Zastosuj wejście, Invec, do sieci i zainicjuj Outvec = Invec
2. Zaczynaj od  $i = 1$
3. Calculate  $Value_i = \text{DotProduct}(\text{Invec}_i, \text{Column}_i \text{ of Weight matrix})$
4. Oblicz  $\text{Outvec}_i = f(\text{Value}_i)$ , gdzie  $f$  jest funkcją progową omówioną wcześniej
5. Zaktualizuj dane wejściowe do sieci za pomocą komponentu Outvec
6. Zwiększyć  $i$  i powtórzyć kroki 3, 4, 5 i 6, aż  $\text{Invec} = \text{Outvec}$  (zauważ, że gdy osiągnę maksymalną wartość, to jest następnie resetowany do 1, aby cykl był kontynuowany)

Teraz zobaczmy, jak zastosować tę procedurę. Opierając się na ostatnim przykładzie, wprowadzamy teraz  $E = (1, 0, 0, 1)$ , które znajduje się w równej odległości od A i B. Bez zastosowania procedury asynchronicznej powyżej, ale zamiast tego używamy procedury skrótu, której używaliśmy jak dotąd otrzymasz wyjście  $F = (0, 1, 1, 0)$ . Ten wektor, F, jako kolejne dane wejściowe, skutkowałby E jako



wyjściem. Jest to nieprawidłowe, ponieważ sieć oscyluje między dwoma stanami. Zaktualizowaliśmy cały wektor wejściowy synchronicznie. Teraz zastosuj aktualizację asynchroniczną.

### Wejścia binarne i bipolarne

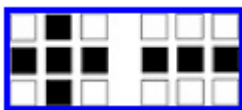
Dwa rodzaje wejść, które są używane w sieciach neuronowych, to wejścia binarne i bipolarne. Widzieliśmy już przykłady wejścia binarnego. Wejścia bipolarne mają jedną z dwóch wartości, 1 i -1. Jest wyraźne odwzorowanie jeden-na-jeden lub korespondencja między nimi, mianowicie posiadanie -1 bipolarnego odpowiada 0 binarnemu. Przy określaniu macierzy wagowej w niektórych sytuacjach, gdy łańcuchy binarne są wejściami, to mapowanie jest używane, a gdy wyjście pojawia się w wartościach bipolarnych, transformacja odwrotna jest stosowana w celu uzyskania odpowiedniego ciągu binarnego. Prostym przykładem może być to, że łańcuch binarny 1 0 0 1 jest mapowany na ciąg dwubiegunowy 1 -1 -1 1; używając transformacji odwrotnej w ciągu dwubiegunowym -1 1 -1 -1, otrzymujemy ciąg binarny 0 1 0 0.

### Stronniczość

Użycie wartości progowej może przybrać dwie formy. Jeden pokazany w przykładzie. Aktywacja jest porównywana z wartością progową, a neuron wystrzeliwuje, gdy wartość progowa zostanie osiągnięta lub przekroczona. Innym sposobem jest dodanie wartości do samej aktywacji, w którym to przypadku nazywa się to odchyleniem, a następnie określenie wyjścia neuronu. Napotkamy później stronniczość i zysk.

### Kolejny przykład dla sieci Hopfield

W części 12 zobaczysz zastosowanie mapy funkcji Kohonena do rozpoznawania wzorców. Podajemy tutaj przykład powiązania wzoru z siecią Hopfield. Wzory to niektóre postacie. Wzór reprezentujący postać staje się wejściem do sieci Hopfielda przez wektor bipolarny. Ten bipolarny wektor jest generowany z siatki pikseli (elementu obrazu) dla znaku, z przypisaniem 1 do czarnego piksela i -1 do piksela, który jest biały. W tych podejściach zazwyczaj stosuje się rozmiar siatki, taki jak 5x7 lub wyższy. Liczba zaangażowanych pikseli wyniesie wtedy 35 lub więcej, co określa wymiar wektora bipolarnego dla wzoru znaku. Dla uproszczenia użyjemy siatki 3x3 dla wzorów znaków w naszym przykładzie. Oznacza to, że sieć Hopfielda ma 9 neuronów w jedynej warstwie w sieci. Ponownie dla uproszczenia używamy dwóch wzorców lub wzorców odniesienia, które podano na rysunku



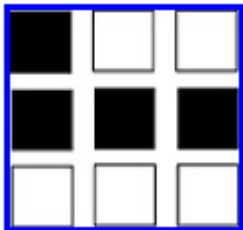
Rozważmy wzór po lewej jako reprezentację znaku „plus”, +, a po prawej znak „minus”, -. Wektory dwubiegunowe, które reprezentują znaki na rysunku, odczytujące wzorce pikseli wiersz po wierszu, od lewej do prawej i od góry do dołu, z 1 dla czerni i -1 dla białych pikseli, to  $C^+ = (-1, 1, -1, 1, 1, 1, -1, 1, -1)$  i  $C^- = (-1, -1, -1, 1, 1, 1, -1, -1, -1)$ . Macierz masy  $W$  to:

$$W = \begin{pmatrix} 0 & 0 & 2 & -2 & -2 & -2 & 2 & 0 & 2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 \\ 2 & 0 & 0 & -2 & -2 & -2 & 2 & 0 & 2 \\ 2 & 0 & -2 & 0 & 2 & 2 & -2 & 0 & -2 \\ 2 & 0 & -2 & 2 & 0 & 2 & -2 & 0 & -2 \\ 2 & 0 & -2 & 2 & 2 & 0 & -2 & 0 & -2 \\ 2 & 0 & 2 & -2 & -2 & -2 & 0 & 0 & 2 \\ 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 2 & -2 & -2 & -2 & 2 & 0 & 0 \end{pmatrix}$$

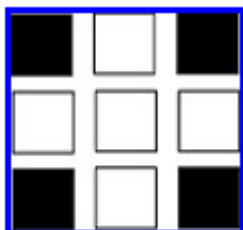
Aktywacje z wejściem C+ są podawane przez wektor (-12, 2, -12, 12, 12, 12, -12, 2, -12). W przypadku wejścia C- wektor aktywacji to (-12, -2, -12, 12, 12, 12, -12, -2, -12). Kiedy ta sieć Hopfielda korzysta z funkcji progowej

$$f(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ -1 & \text{if } x < 0 \end{cases}$$

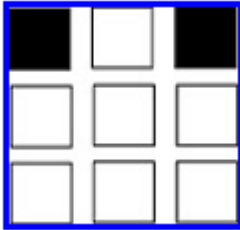
odpowiednimi wyjściami będą odpowiednio C+ i C-, pokazujące stabilne przywoływanie przykładowych wektorów, i ustanawianie dla nich automatycznego powiązania. Kiedy wektory wyjściowe są używane do konstruowania odpowiednich znaków, otrzymujesz oryginalne wzory znaków. Wprowadźmy teraz wzór znaku na rysunku



Nazwiemy odpowiedni wektor bipolarny  $A = (1, -1, -1, 1, 1, 1, -1, -1, -1)$ . Otrzymujesz wektor aktywacji (-12, -2, -8, 4, 4, 4, -8, -2, -8), dając wektor wyjściowy,  $C = (-1, -1, -1, 1, 1, 1, -1, -1, -1)$ . Innymi słowy, postać - nieco uszkodzona, jest przywoływana jako postać - przez sieć Hopfielda. Zamierzony wzór zostanie rozpoznany. Wprowadzamy teraz wektor bipolarny, który różni się od wektorów odpowiadających przykładom, i sprawdzamy, czy sieć może przechowywać odpowiedni wzorec. Wybrany wektor to  $B = (1, -1, 1, -1, -1, -1, 1, -1, 1)$ . Odpowiednie aktywacje neuronów są podawane przez wektor (12, -2, 12, -4, -4, -4, 12, -2, 12), który powoduje, że wyjście jest wektorem (1, -1, 1, -1, -1, -1, 1, -1, 1), tak samo jak B. Dodatkowy wzorec, który jest siatką 3x3 z czarnymi pikselami narożnymi, jak pokazano na rysunku



jest również przywoływany, ponieważ jest on automatycznie powiązany, przez tę sieć Hopfielda. Jeśli pominiemy część wzoru z powyższego rysunku, pozostawiając tylko czarne narożniki, jak na rysunku



otrzymamy wektor bipolarny  $D = (1, -1, 1, -1, -1, -1, -1, -1, -1)$ . Możesz to również uznać za niekompletną lub uszkodzoną wersję wzoru na rysunku poprzednim. Aktywacje sieciowe okazują się być  $(4, -2, 4, -4, -4, 8, -2, 8)$  i dają wyjście  $(1, -1, 1, -1, -1, -1, 1, -1, 1)$ , czyli B.

### Podsumowanie

W tej części wprowadziliśmy sieć neuronową jako przetwarzanie zbioru elementów rozproszonych na skończonej liczbie warstw i połączonych dodatnimi lub ujemnymi wagami, w zależności od tego, czy współpraca lub konkurencja (lub zahamowanie) jest zamierzona. Aktywacja neuronu jest zasadniczo ważoną sumą jego wejść. Funkcja progowa określa wyjście sieci. Pomiędzy warstwą wejściową a warstwą wyjściową mogą znajdować się warstwy neuronów, a niektóre takie warstwy środkowe nazywane są warstwami ukrytymi, inne nazwami, takimi jak warstwy Grossberga lub Kohonena, nazwanymi na cześć naukowców Stephena Grossberga i Teuvo Kohonena, którzy zaproponowali ich i ich funkcje. Modyfikacja wag jest procesem szkolenia sieci, a mówi się, że sieć podlegająca temu procesowi uczy się w tej fazie działania sieci. W niektórych operacjach sieciowych stosowana jest operacja sprzężenia zwrotnego, w której wyjście prądowe jest traktowane jako zmodyfikowane wejście do tej samej sieci. Widziałeś kilka przykładów sieci Hopfield, jednej z nich do rozpoznawania wzorców. Sieci neuronowe mogą być wykorzystywane do rozwiązywania problemów, których nie można rozwiązać przy użyciu znanej formuły, oraz w przypadku problemów z niepełnymi lub zaszumionymi danymi. Sieci neuronowe wydają się mieć zdolność rozpoznawania wzorców w prezentowanych danych, a zatem są przydatne w wielu typach problemów z rozpoznawaniem wzorców.