

Naśladowanie żywego mózgu w komputerze

Ryszard Tadeusiewicz

Katedra Automatyki, Akademia Górniczo-Hutnicza im. S. Staszica
Kraków

rtad@agh.edu.pl



Streszczenie

Wykład stanowi wprowadzenie do nowego typu narzędzi informatycznych, znanych pod nazwą **sieci neuronowe**. Narzędzia te, jeszcze niedawno traktowane nieufnie przez informatyków, dziś są szeroko stosowane ze względu na liczne zalety. Naśladując w komputerze ludzki mózg staramy się bowiem połączyć zalety komputera (łatwa dostępność i szybkość działania) z zaletami mózgu (zdolność uczenia się). Wykład zawiera wyłącznie teoretyczne elementy wiedzy na temat sieci neuronowych, w dodatku przedstawione w bardzo dużym skrócie. Można jednak swoją wiedzę poszerzyć korzystając ze wskazanych w treści wykładu zasobów internetowych. Z tego samego źródła można skorzystać w celu uzupełnienia wiedzy teoretycznej elementami praktyki posługując się dostępnymi darmowymi programami, pozwalającymi na samodzielne eksperymentowanie ze sztucznymi neuronami i z sieciami neuronowymi na domowym komputerze lub na poręcznym laptopie. Tekst wykładu dostarcza wiedzy potrzebnej do tego żeby taką samodzielną zabawę z sieciami neuronowymi sensownie zacząć.

Spis treści

1. Geneza sieci neuronowych 133

2. Budowa sieci neuronowych 137

3. Inteligencja sieci neuronowych 142

4. Realizacja sieci neuronowych 144

5. Proces uczenia sieci neuronowych 146

6. Zastosowania sieci neuronowych 149

7. Zakończenie 149

Literatura 151

1 GENEZA SIECI NEURONOWYCH

Wykład stanowi wprowadzenie do nowego typu narzędzi informatycznych, znanych pod nazwą **sieci neuronowe**. Narzędzia te, jeszcze niedawno traktowane nieufnie przez informatyków, dziś są szeroko stosowane ze względu na liczne zalety.

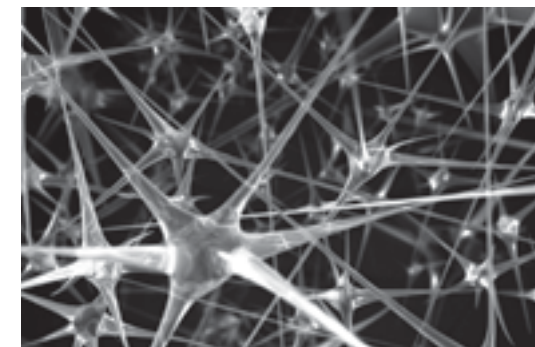
Pokazane na rysunku 1 metaforyczne połączenie elementów technicznych (układy scalone i inne elementy techniczne, z jakich zwykle składa się komputery) z kształtem ludzkiej głowy ma na celu uformowanie pierwszego skojarzenia, wskazującego na to, że omawiane dalej sieci neuronowe dziedziczą pewne właściwości po biologicznym pierwowzorze (mózgu), ale są w istocie specjalnego typu komputerami [1].



Rysunek 1.

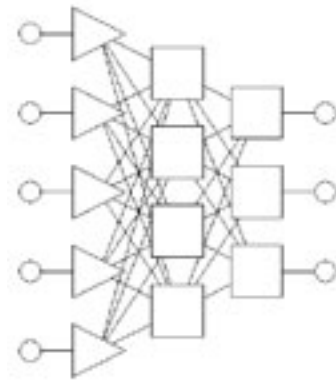
Symboliczne wyobrażenie naśladowania mózgu w komputerze [źródło: https://cs.byu.edu/files/images/neural_networking.jpg, dostęp lipiec 2011]

Badacze zbudowali wiele systemów technicznych, naśladujących w komputerze ludzki mózg. Najbardziej pożyteczne okazały się **sieci neuronowe**. Badane przez biologów wewnątrz mózgu ujawnia sieć powiązanych ze sobą elementów, tak zwanych **neuronów** (rys. 2). Mają one dosyć skomplikowaną budowę i funkcje, poznano je jednak na tyle dobrze, że potrafimy budować ich techniczne odpowiedniki – **sztuczne neurony**. Gdy się takie sztuczne neurony odpowiednio połączy, to uzyskuje się sztuczną sieć neuronową. Na rysunku 3 pokazano schemat przykładowej sieci neuronowej. Sztuczne neurony zaznaczono w postaci prostokątnych bloków, a połączenie między neuronami symbolizują kreski łączące bloki. Trójkątne symbole oznaczają tzw. neurony wejściowe, służące do tego, żeby odebrać od użytkowników dane charakteryzujące zadanie do wykonania i rozesać odpowiednie informacje w głąb sieci. Rozwiązanie zadania pojawia się w postaci sygnałów na liniach zakończonych kółkami (rys. 3).



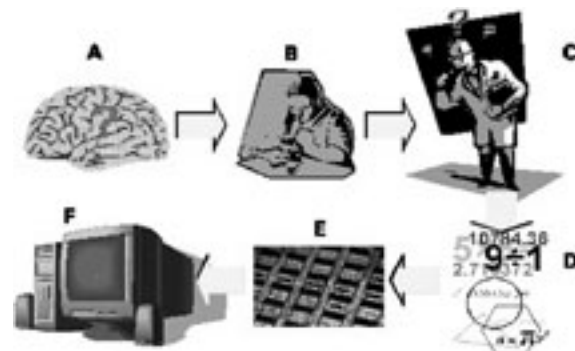
Rysunek 2.

Badanie wewnątrz mózgu ujawniło w nim sieć powiązanych ze sobą elementów [źródło: http://encefalus.com/wp-content/uploads/2010/07/neural_network.jpg, dostęp lipiec 2011]



Rysunek 3. Sztuczna sieć neuronowa powstaje, gdy się połączy ze sobą sztuczne elementy naśladowujące rzeczywiste biologiczne neurony [źródło: 3]

Jak doszło do zbudowania pierwszych sieci neuronowych? Był to długi proces, przedstawiony schematycznie na rysunku 4. Punktem początkowym był mózg człowieka (A), a dokładniej – zafascynowanie ludzi jego skomplikowaną budową i tajemniczymi funkcjami. Badania naukowe biologów (B) dostarczyły wielu szczegółowych wyników, które stały się podstawą analiz, prowadzonych przez biocybernetyków (C), którzy na tej podstawie zbudowali wiele modeli matematycznych, opisujących mózg i jego działanie (D). Opierając się na tych modelach matematycznych neurocybernetycy zbudowali z kolei specjalizowane układy elektroniczne (tzw. **neurokomputery** E), które w swoim działaniu naśladowują wybrane funkcje mózgu. Jednak w zastosowaniach praktycznych bardziej użyteczne okazały się sieci neuronowe symulowane za pomocą odpowiednich programów w strukturach zwykłych komputerów (F).



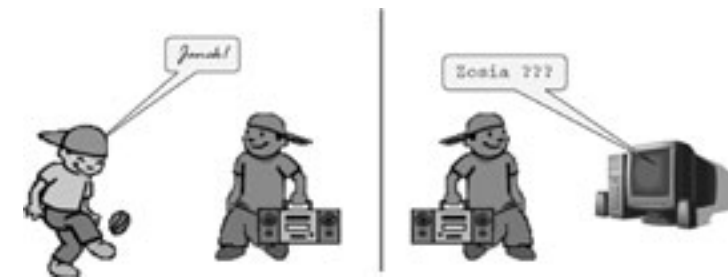
Rysunek 4. Droga powstania sieci neuronowych: od badań naukowych do technicznych zastosowań [źródło: 2]

Co chcemy osiągnąć naśladowując w komputerze ludzki mózg? Przecież komputery są dziś tak sprawne i tak dobrze zaspokajają wszelkie nasze potrzeby? Otóż naśladowanie mózgu w komputerze jest nam potrzebne między innymi do tego, żeby rozwiązać zadania, dla których nie potrafimy podać gotowych algorytmów ich rozwiązywania. Przypomnijmy sobie, że komputer potrafi zrobić tylko to, co mu narzucił programista, a programista opisuje to, co komputer powinien zrobić, tworząc odpowiedni algorytm. Jeśli więc potrafię rozwiązać jakieś zadanie i umiem podać opis postępowania prowadzącego do sukcesu (czyli właśnie algorytm), to komputer też potrafi rozwiązać zadanie posługując się tym algorytmem. Istnieją jednak liczne zadania, dla których nikt nie potrafi podać algorytmu.

Na pierwszy rzut oka wydaje się, że takie problemy, tak skomplikowane, że nikt na świecie nie potrafił podać algorytmu ich rozwiązywania, zdarzają się rzadko i dotyczą jakichś bardzo nietypowych sytuacji. Nie-

prawda! Przykładem zadania, które jest pozornie bardzo łatwe, a dla którego nikt nie umie podać algorytmu, jest automatyczne rozpoznawanie ludzi (rys. 5).

Obraz człowieka uzyskany za pomocą dowolnego cyfrowego aparatu (nawet zwykłej komórki) można łatwo wprowadzić do komputera, który może go zapamiętać, pokazać na ekranie, wydrukować, przesać, a nawet przetworzyć, na przykład zmieniając rozmiary czy proporcje. Ale nikt nie potrafi napisać programu, dzięki któremu komputer sam zamieni miliony pikseli składających się na obraz na pewną i trafną decyzję: *To jest Janek, to na pewno on i nikt inny!*



Rysunek 5. Zadanie rozpoznawania jako przykład zadania łatwego dla ludzi, dla którego jednak nie można stworzyć skutecznego algorytmu. Po lewej łatwe rozwiązywanie tego problemu przez człowieka, po prawej nieskuteczne rozwiązywanie tego problemu przez komputer [źródło: 2]

Uwzględniając to, co powiedziano wyżej, można teraz odpowiedzieć na pytanie: Kiedy sieci neuronowe są lepsze od innych metod informatycznych?

Dla dokładnego wskazania tego obszaru narysujemy prosty wykres (rys. 6), służący do scharakteryzowania i opisanego różnych problemów informatycznych. Na osi poziomej tego wykresu zaznaczamy, jak bardzo rozważany problem jest **trudny**. Łatwiejsze do rozwiązania zadania umieścimy po lewej stronie, a im trudniejsze zadanie będziemy chcieli rozwiązać, tym bardziej na prawo będziemy je umieszczać na pokazanej „mapie”. Tę miarę (że coś jest łatwiejsze albo trudniejsze do rozwiązania) nazwiemy **złożonością**. Zadania łatwe cechuje mała złożoność. Zadania trudne wiążą się z dużą złożonością. Na osi pionowej zaznaczmy z kolei stopień naszej **nieumiejętności** znalezienia algorytmu, za pomocą którego moglibyśmy problem rozwiązać. Jeśli dokładnie wiemy, jak rozwiązać zadanie, to umieścimy je nisko. Ale im więcej zagadek i trudności spodziewamy się napotkać, tym wyżej będziemy lokować nasze zadanie na rysowanej mapie.

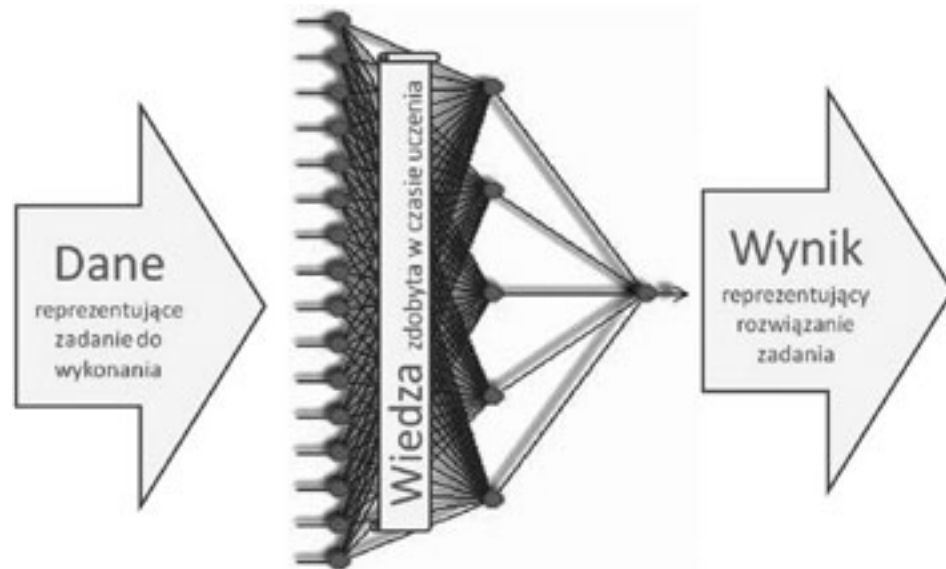


Rysunek 6. Diagram ilustrujący, kiedy sieci neuronowe są lepsze od innych metod informatycznych [źródło: 2]

Tytułem przykładu na rysunku 6 zaznaczono trzy zadania.

W lewym dolnym rogu jest zadanie łatwe i w dodatku takie, dla którego znamy sposób rozwiązywania. Naturalnym sposobem wykonania takiego zadania jest wykorzystanie algorytmu, który w tym przypadku jest możliwy do opracowania. Pośrodku zaznaczono zadanie, w którym znajomość reguł jest tylko częściowa, co utrudnia zastosowanie algorytmu. W takim przypadku zamiast na dokładnym algorytmie możemy rozwiązanie oprzeć na opinii eksperta, którego zdroworozsądkowe rady mogą być zgromadzone w specjalnej bazie wiedzy, albo możemy skorzystać z namiastki wiedzy, jaką daje statystyka. Na przykład nie znamy reguł opisujących zachowanie w sklepie każdego konkretnego klienta i nie wiemy, czym się kieruje wybierając towary, ale **statystyka** pozwala przewidzieć, że jak się pojawi w telewizji reklama jakiegoś towaru, to więcej klientów go kupi. Taka metoda rozumowania, w której ogólna reguła (*reklama zwiększa sprzedaż*) pozwala na odgadywanie konkretnych zachowań, nazywa się **dedukcją**. Właśnie znajomość ogólnych reguł opisujących zachowania ludzi (a zwłaszcza kryminalistów) umożliwiła ujawnianie przestępców najstynniejszemu detektywowi, jakim był niewątpliwie Sherlock Holmes.

Dedukcji nie da się zastosować, gdy rozważany problem jest całkowicie pozbawiony jakichkolwiek reguł dostępnych dla osoby, która próbuje go rozwiązać (jest to obszar zaznaczony na rysunku 6 w prawym górnym rogu). Brak reguł nie uniemożliwia jednak rozwiązania problemu, ponieważ ludzie potrafią w takich przypadkach stosować wnioskowanie przez analogię. Jeśli znamy kilka przykładów poprawnie rozwiązanych zadań, to potrafimy często odgadnąć prawidłowy wynik kolejnego zadania tego samego typu. Taki typ rozumowania, oparty na przykładach, a nie na regułach, nazywa się **indukcją**. Otóż sieci neuronowe są najlepszym znanym obecnie narzędziem informatycznym zdolnym do automatycznego prowadzenia rozumowania indukcyjnego. Odpowiedź na pytanie, dlaczego tak jest, stanie się jasna, gdy poznamy sposób rozwiązywania problemów przy użyciu sieci neuronowej. Wprawdzie o budowie sieci nie było jeszcze mowy, ale spojrzymy na razie, jak taka sieć działa, traktowana chwilowo jako całość (rys. 7). Na rysunku widzimy sieć jako pewną strukturę połączonych ze sobą elementów. Domyślamy się, że te elementy to sztuczne neurony, ale do budowy sieci jeszcze wrócimy i poznamy ją znacznie dokładniej. Po lewej stronie jest strzałka reprezentująca dane wejściowe. Tutaj także nie pokazano jeszcze żadnych konkretnych, ale widzimy w jaki sposób stawia się sieci zadanie do rozwiązania – trzeba wszystkie jego istotne elementy przedstawić w postaci sygnałów możliwych



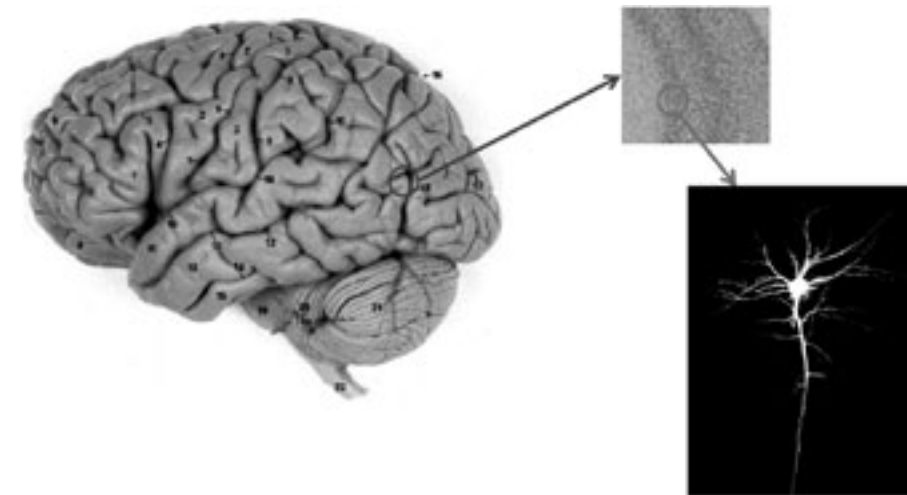
Rysunek 7. Ilustracja działania sieci neuronowej [źródło: 2]

do wprowadzenia do neuronów. Wymaga to czasem dokładnego przemyślenia tego, co w danej kwestii jest ważne i jak te ważne informacje podać, żeby mogły być przyjęte przez sieć, ale to się opłaca. Również rozwiązanie zadania ma postać sygnału pojawiającego się na wyjściu neuronu, który jest ostatnim elementem sieci. Trzeba zatem tak sformułować zadanie, żeby sygnał z wyjściowego neuronu (zwykle mający formę liczby przyjmującej wartości z przedziału od 0 do 1) można było zinterpretować (zrozumieć) jako odpowiedź na pytanie, które zostało postawione w zadaniu. Czasami jest więcej niż jedno wyjście z sieci, bo chcemy uzyskać odpowiedzi na kilka pytań równocześnie, ale tym zagadnieniem zajmiemy się nieco dalej.

Sieć potrafi wypracować taką odpowiedź na podstawie danych wejściowych, ponieważ ma pewną wiedzę. Rysunek sugeruje, gdzie się ta wiedza mieści: w połączeniach pomiędzy neuronami. Odpowiedni napis na rysunku wskazuje też, skąd się ta wiedza bierze – powstaje automatycznie w trakcie procesu uczenia, który niebawem omówimy.

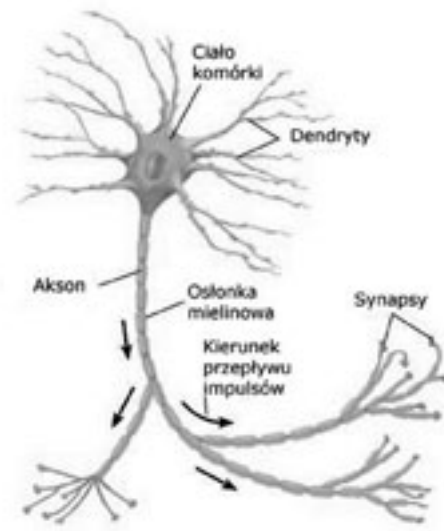
2 BUDOWA SIECI NEURONOWYCH

Naśladując w komputerze ludzki mózg staramy się wzorować na jego budowie. Analizując tę budowę coraz dokładniej najpierw widzimy cały mózg, potem wycinek jego kory, a na końcu – pojedynczą komórkę nerwową będącą głównym budulcem tej kory (patrz rysunek 8). Na rysunku należy zwrócić uwagę na zmienną skalę poszczególnych jego części: cały mózg ma rozmiar kilkunastu centymetrów, pokazany fragment kory mózgowej ma szerokość i wysokość 1 mm, a uwidoczniony pojedynczy neuron ma średnicę 20 mikrometrów.



Rysunek 8. Etapy poznawania budowy i elementów składowych mózgu [źródło: 1]

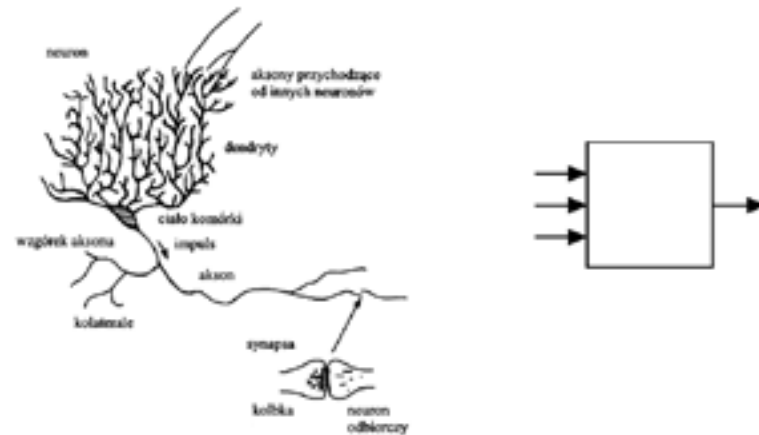
Tworząc modele mózgu nadające się do umieszczenia ich w komputerze zaczynamy od podstawowego elementu składowego, jakim jest **sztuczny neuron**. Stanowi on bardzo uproszczony model rzeczywistego biologicznego neuronu. Rzeczywista komórka nerwowa, wchodząca w skład mózgu, ma dość skomplikowaną budowę, pokazaną na rysunku 9. Elementy składowe komórki mają następujące funkcje: **dendryty** zbierają sygnały wejściowe i doprowadzają je do **ciała komórki**, które jest biologicznym procesorem analizującym te sygnały. Wynik tej analizy zbierany jest przez **akson**, który jest izolowany jak kabel elektryczny za pomocą **osłonki mielinowej**. Impulsy wyjściowe neuronu są przekazywane do następnych neuronów za pomocą **synaps**, w których lokują się procesy uczenia.



Rysunek 9. Budowa biologicznej komórki nerwowej [źródło: 3]

Przy opisie pojedynczego rzeczywistego neuronu (biologicznej komórki nerwowej będącej częścią mózgu) warto odnotować jeszcze jedną ciekawostkę, widoczną na rysunku 8. Otóż ciała komórek są pozbawione osłonki mielinowej, więc tam, gdzie są one nagromadzone w dużych ilościach (głównie w korze mózgowej, ale także i w innych strukturach anatomicznych, na przykład we wzgórzu albo w jądrach pnia mózgu), tkanka nerwowa ma charakterystyczną szarą barwę, bo taki jest naturalny kolor tych komórek. Natomiast tam, gdzie biegną aksony łączące neurony w sieci, kolor (biały z perlowym połyskiem) całej tkance nadają osłonki mielinowe przypisane do tych biologicznych „kablów połączeniowych”. Stąd już starożytni badacze mózgu wyróżniali w nim „substancję szarą” oraz „substancję białą”, zaś w wielu kontekstach mówi się o „szarych komórkach” mając na myśli siedlisko naszej inteligencji.

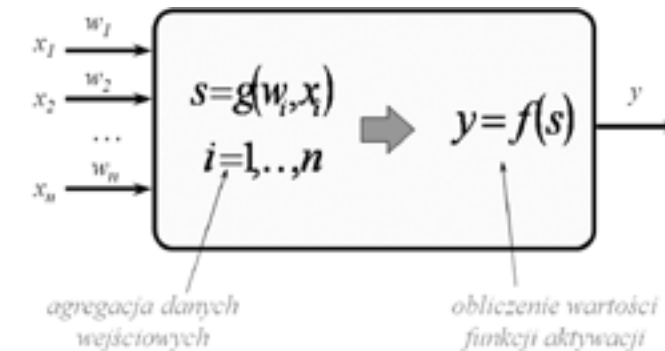
Budując sztuczne neurony, z których są tworzone sieci neuronowe, staramy się w nich odwzorować jedynie najważniejsze cechy biologicznych neuronów. Z komórki, która ma skomplikowaną budowę, pozostaje więc tylko blok przetwarzający informacje (odpowiednik ciała komórki), duża liczba kanałów wejściowych, odpowiadających dendrytom i jeden kanał wyjściowy, będący odpowiednikiem aksonu (rys. 10).



Rysunek 10. Przejście pomiędzy biologicznym neuronem i jego sztucznym modelem [źródło: 3]

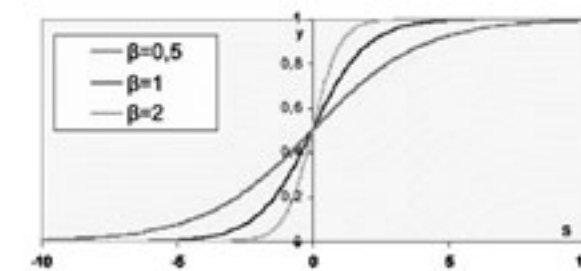
Sztuczny neuron musi się zachowywać w sposób maksymalnie podobny do zachowania biologicznego neuronu, a jednocześnie nie może być zbyt skomplikowany, bo jego realizacja techniczna będzie wtedy zbyt kosztowna. Trzeba bowiem pamiętać, że dla zbudowania sieci neuronowej, która będzie służyć do rozwiązywania jakiegoś praktycznego zadania, będziemy potrzebowali na ogół kilkudziesięciu lub nawet kilkuset neuronów. Dlatego przy tworzeniu sztucznych neuronów stosujemy wyłącznie trzy operacje, przedstawione schematycznie na rysunku 11:

- **Agregacja danych wejściowych.** Skoro neuron ma wiele wejść, a na tych wejściach wiele sygnałów x_1, x_2, \dots, x_n , zaś wyjście jest tylko jedno – to jest oczywiste, że te wejściowe sygnały trzeba zintegrować, czyli jakoś połączyć razem, aby wyprodukować ten jeden sygnał wyjściowy. W najprostszym przypadku można sygnały wejściowe zsumować, ale bywają też bardziej wyrafinowane metody agregacji. Sygnał po agregacji jest oznaczany przez s .
- Sygnały wejściowe trzeba **zróżnicować** (pod względem skutków ich oddziaływania na neuron). Do tego celu służą współczynniki nazywane **wagami**. Z każdym wejściem o numerze i , do którego dociera z zewnątrz sygnał x_i , jest związany współczynnik wagi w_i . Znaczenie tego współczynnika będzie za chwilę omówione nieco dokładniej.
- Gdy już sygnały wejściowe zostały zróżnicowane przez wagi oraz zagregowane przez specjalną funkcję $g(w_i, x_i)$ (gdzie i zmienia się od 1 do n) – trzeba ustalić, jaki sygnał y wysłanie neuron na swoim wyjściu (odpowiadającym aksonowi). Wiąże się to z zastosowaniem **funkcji aktywacji** $f(s)$.



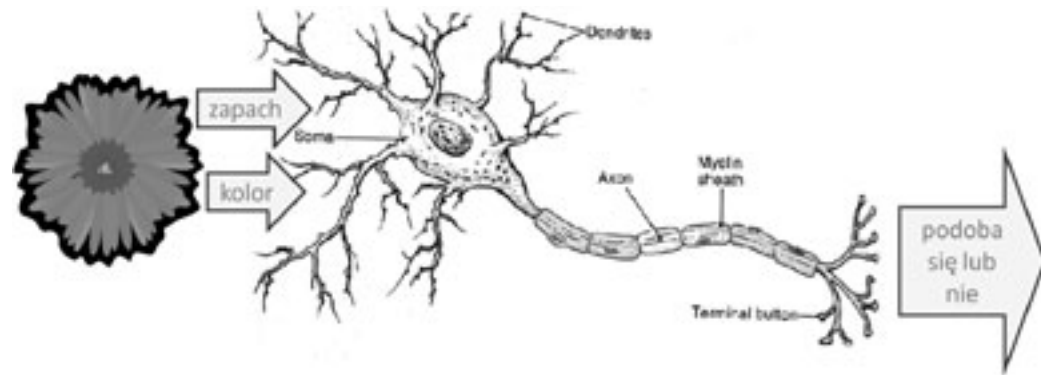
Rysunek 11. Operacje uwzględnione w sztucznym neuronie

Agregacja sygnałów wejściowych jest raczej łatwa do przeprowadzenia. Jak wspomniano wyżej, jest to zwykle po prostu sumowanie. Funkcja aktywacji też jest łatwa do zrozumienia, więc ograniczymy się do pokazania jej przykładowego kształtu (rys. 12). Na tym wykresie zmienna s (sumaryczne pobudzenie neuronu) jest odkładana wzdłuż osi poziomej, a sygnał wyjściowy y , wysyłany poprzez akson neuronu, jest odczytywany z osi pionowej. Parametr β służy do doboru kształtu funkcji zależnie od potrzeb.



Rysunek 12. Jedna z najpopularniejszych funkcji aktywacji neuronu, tak zwana **sigmoida**

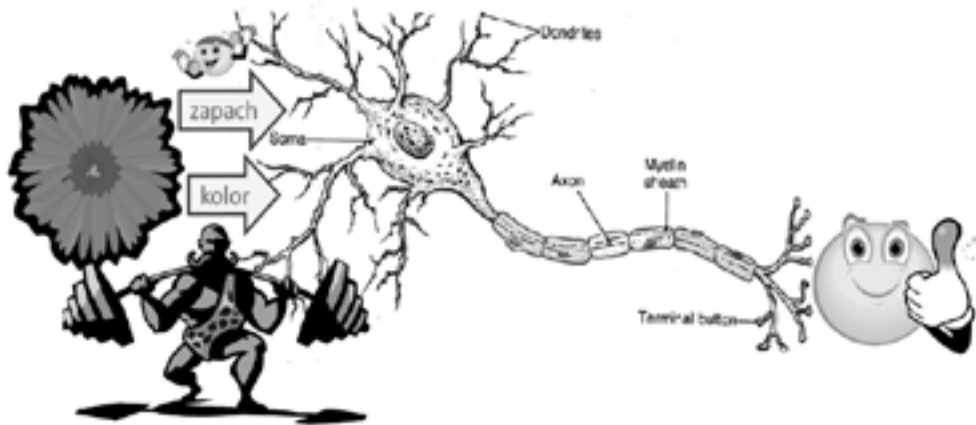
Zagadnieniem, któremu warto się przyjrzeć uważniej, jest kwestia wag różnicujących wejścia do neuronu. Wejścia te mogą być związane z różnymi cechami obserwowanego przez neuron obiektu. Na przykład na rysunku 13 pokazano neuron, do którego docierają dwa sygnały opisujące obserwowany kwiat. Jeden z tych sygnałów niesie informację o zapachu, a drugi o kolorze tego obiektu. Neuron może wysłać na wyjściu sygnał, że dany kwiat mu się podoba albo nie.



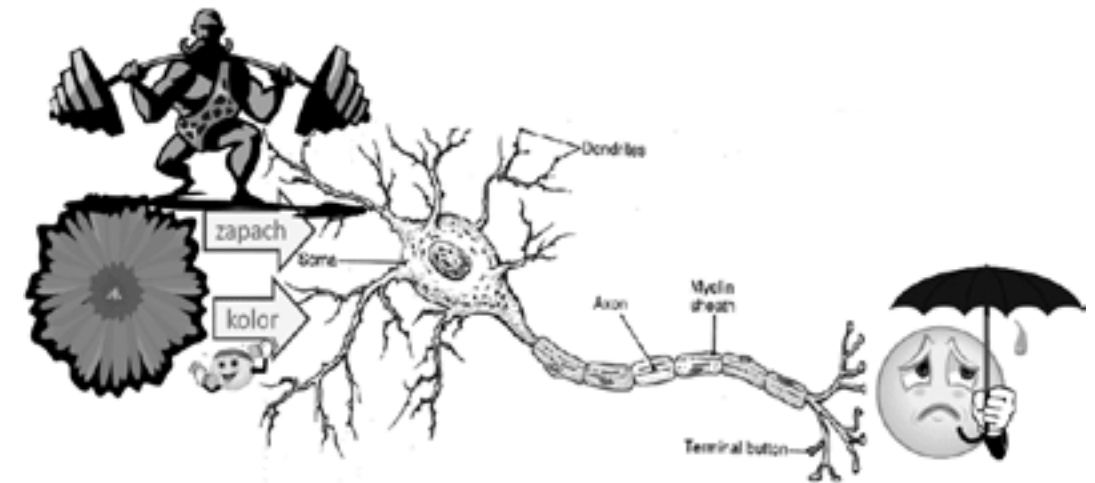
Rysunek 13. Znaczenie wag na wejściach neuronu [źródło: 2]

Załóżmy, że oceniany kwiat ma ładny kolor, ale brzydki zapach. Jeżeli do sygnału wejściowego „zapach” przypiszemy małą wagę, a do sygnału „kolor” wagę dużą, to neuron wyśle na wyjściu sygnał, że kwiat mu się podoba (rys. 14). Natomiast przy przeciwnym ustawieniu wag, ocena wystawiona przez neuron będzie negatywna (rys. 15). Przy tym samym zestawie sygnałów wejściowych mamy dwie całkiem różne reakcje neuronu.

Wagi przypisywane wejściom neuronów mogą być nie tylko duże i małe, ale także dodatnie oraz ujemne, co oznacza że pewne cechy ocenianych obiektów, chociaż obiektywnie pozytywne, będą negatywnie oceniane przez neuron. Na przykład można by było sobie wyobrazić, że neuron będzie negatywnie reagował na ładnie pachnące kwiaty (po prostu taki będzie miał gust). W takim przypadku na pierwszym wejściu będzie można ustawić ujemną wartość wagi. Albo można sobie wyobrazić, że waga związana z kolorem będzie wynosiła zero. Wówczas kolor nie będzie wcale wpływał na ocenę podawaną przez neuron, co może odpowiadać sytuacji daltonisty. Przykładów można by było mnożyć bez liku.

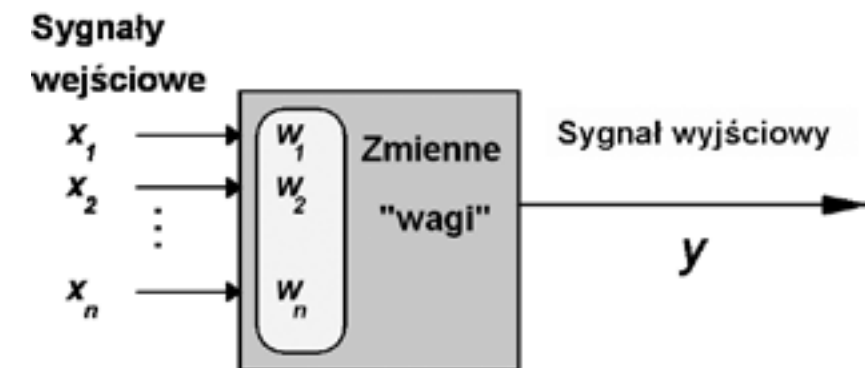


Rysunek 14. Określone ustawienie wag neuronu daje pozytywną odpowiedź neuronu [źródło: 2]



Rysunek 15. Odwrotne ustawienie wag daje odwrotny efekt [źródło: 2]

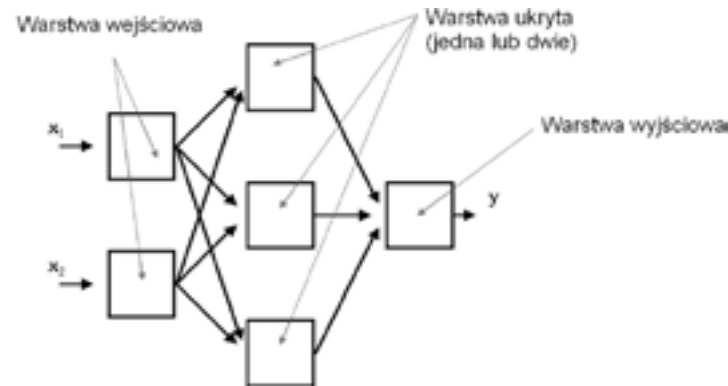
O sztucznym neuronie można by jeszcze długo opowiadać, ale poprzestaniemy na tym, że pokażemy jego zbiorczy schemat (rys. 16), uwzględniający wszystkie do tej pory ustalone fakty, bowiem najwyższa już pora zacząć budować sieci. Model z rysunku 16 będzie służył dalej jako podstawowa „cegietka” przy budowie sieci neuronowych.



Rysunek 16. Schemat właściwości sztucznego neuronu [źródło: 3]

Pytanie, które sobie teraz trzeba postawić, brzmi: Jak połączyć sztuczne neurony, żeby powstała użyteczna sieć? Współczesne sieci neuronowe buduje się z neuronów układanych w warstwy (rys. 17).

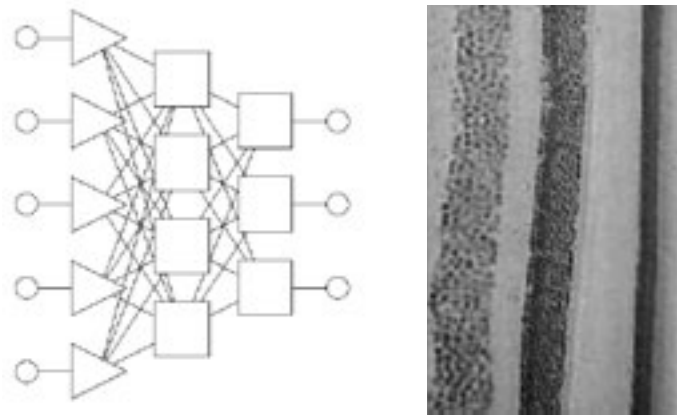
- Wyjaśnimy teraz kolejno, jaką rolę odgrywają poszczególne warstwy sieci:
- **Warstwa wejściowa** służy do wprowadzenia danych, które są potrzebne do rozwiązania postawionego zadania. Warstwa ta nie uczestniczy z reguły w procesie uczenia.
 - **Warstwa ukryta** analizuje dane i przygotowuje przesłanki do rozwiązania zadania. To tutaj mieści się głównie inteligencja sieci.
 - **Warstwa wyjściowa** wyznacza ostateczne rozwiązanie zadania i podaje je do wykorzystania.



Rysunek 17.

Przykładowa sieć z jej wszystkimi ważnymi elementami [źródło: 3]

Sieci neuronowe o budowie warstwowej najczęściej służą do rozwiązywania konkretnych zadań, ponieważ takie sieci w prosty sposób się buduje, łatwo się ich nauczyć, wygodnie się ich używa. Nie są one tak całkiem pozbawione merytorycznego uzasadnienia. Na rysunku 18 pokazano przykładową sieć neuronową i obraz mikroskopowy fragmentu kory mózgowej, odpowiedzialnego za analizę wrażeń wzrokowych. Wnioski nasuwają się same.

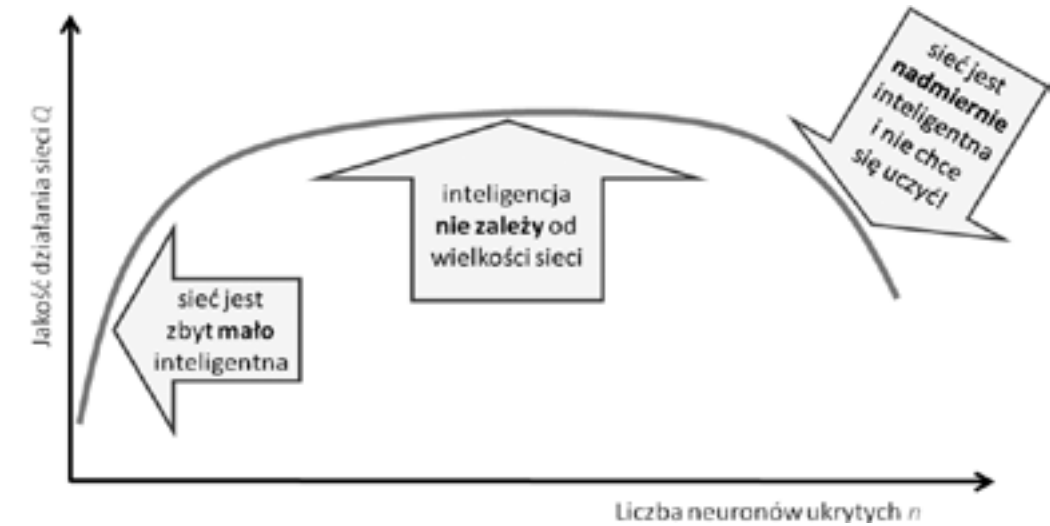


Rysunek 18.

Budowa typowej sztucznej sieci neuronowej i przekrój kory mózgowej [źródło: 2]

3 INTELIGENCJA SIECI NEURONOWYCH

Teoretycznie twórca sieci może dowolnie wybrać wszystkie jej elementy. W rzeczywistości jednak swoboda twórcy sieci jest ograniczona, bo liczba neuronów w warstwie wejściowej zależy od liczby posiadanych danych, a wielkość warstwy wyjściowej zależy od tego, jakie chcemy dostać wyniki. Twórca sieci neuronowej może więc głównie mieć wpływ na liczbę neuronów ukrytych n . Decyduje ona o jakości działania sieci Q , co pokazano na rysunku 19. Oglądając ten rysunek warto zwrócić uwagę na interpretację miary jakości Q (oznaczenie to pochodzi od angielskiego słowa *Quality* oznaczającego właśnie *jakość*). Oczywiście będzie ona różnie definiowana dla różnych zadań wykonywanych przez sieć, ale na tym etapie rozważań możemy założyć, że sieć powinna rozpoznawać jakieś obiekty (ludzkie twarze, ręcznie pisane litery, odciski palców itp.). W takim przypadku Q może po prostu pokazywać procent poprawnie rozpoznanych obiektów.



Rysunek 19.

Zależność jakości działania sieci Q od liczby neuronów ukrytych n

Kształt zależności jakości działania sieci Q od liczby neuronów ukrytych n wykazuje istnienie trzech wyraźnie odmiennych obszarów, wskazanych na rysunku 19 przez odpowiednio opisane strzałki.

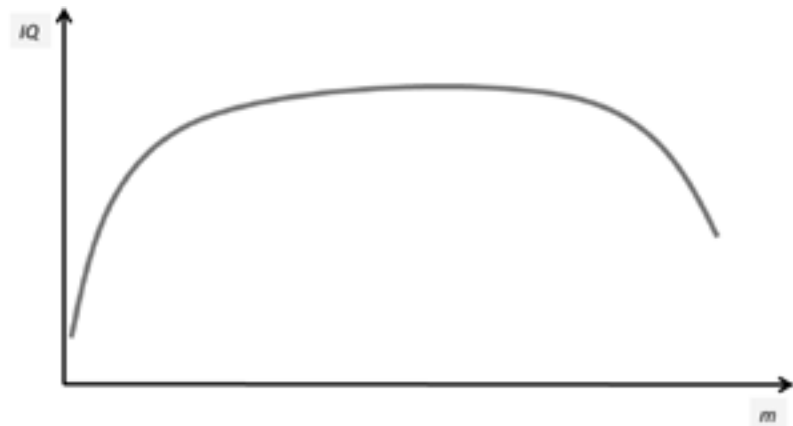
Pierwszy obszar odpowiadający zbyt małej liczbie neuronów ukrytych to obszar, w którym sieć jest zbyt mało inteligentna, żeby nauczyć się rozwiązywać postawione zadanie.

Drugi obszar cechuje się tym, że praktycznie nie ma związku między wielkością sieci, a jakością jej działania. Taką samą jakość Q mogą wykazywać (po procesie uczenia) sieci o różnej liczbie neuronów ukrytych n .

Najbardziej zagadkowy jest trzeci obszar, w którym sieć pogarsza swoje działanie w miarę jak przybywa jej elementów. W tym obszarze sieć jest już tak inteligentna, że potrafi skutecznie wykręcić się od uczenia i nie można nad nią zapanować. Fenomen ten będzie dokładniej omówiony przy dyskusji metod uczenia sieci. Zapamiętajmy jednak: zbyt duża inteligencja sieci jest szkodliwa!

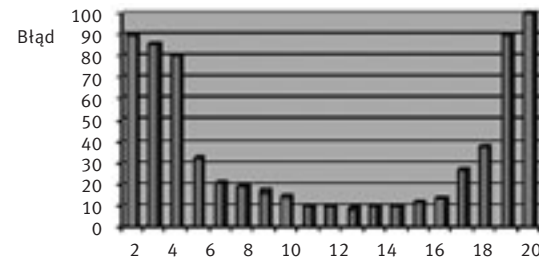
W zasadzie można by było na tym poprzestać, popatrzmy jednak, co się stanie, kiedy na wykresie z rysunku 19 zamienimy n na m oraz Q na IQ (rys. 20). Wykres ten przedstawia teraz znaną z psychologii zależność miary inteligencji człowieka (IQ to tzw. **iloraz inteligencji**) od masy jego mózgu m . Oglądając ten rysunek należy zwrócić uwagę na to, że lewa część wykresu odpowiada przypadkom tzw. niedorozwoju umysłowego (*oligocefalii*), gdzie w zależności od stopnia niedoboru tkanki mózgowej mamy do czynienia z takimi przypadkami jak debilizm, imbecylizm, kretynizm itd. Środkowa część wykresu pokazuje, że u większości ludzi poziom inteligencji nie zależy od wielkości mózgu. Na przykład znany fakt biologiczny, że mózg typowej kobiety jest mniejszy niż mózg mężczyzny w żaden sposób nie przekłada się na różnice sprawności intelektualnej. Ciekawe jest też to, że osoby mające zbyt wielki mózg z reguły nie są w stanie prawidłowo funkcjonować w społeczeństwie i ich miara inteligencji bywa ponownie dramatycznie niska.

Przedstawione wyżej (zwłaszcza na rysunku 19) ogólne reguły wskazujące na niekorzystne skutki używania zarówno za małej, jak i za dużej sieci, potwierdzają wyniki przykładowego eksperymentu, przytoczone na rysunku 21. W eksperymencie tym oceniano błąd popełniany przez sieci neuronowe mające różną liczbę neuronów ukrytych. Widać, że tam, gdzie ogólny model przewidywał niską jakość działania sieci (mierzoną wskaźnikiem Q) konkretnie badana tu sieć popełniała dużo błędów. Dotyczyło to zarówno sieci o zbyt małej liczbie neuronów ukrytych, jak i sieci o liczbie neuronów zbyt dużej.



Rysunek 20.

Wykres z rysunku 19 po zmianie znaczenia zmiennych odkładanych na osiach nabiera nowego sensu, także informującego o czymś ciekawym

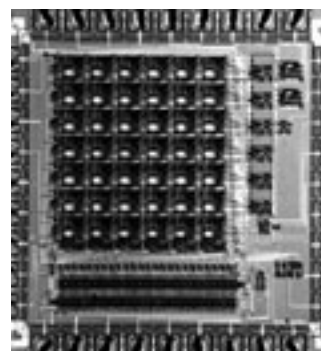


Rysunek 21.

Wyniki eksperymentu potwierdzającego zależność sprawności działania sieci neuronowej od liczby neuronów ukrytych w tej sieci [źródło: 2]

4 REALIZACJA SIECI NEURONOWYCH

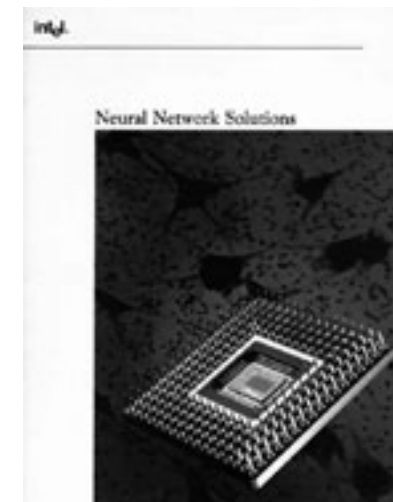
Mając zaprojektowaną sieć neuronową, to znaczy wiedząc, z jakich elementów jest ona zbudowana (sztuczne neurony), ile tych elementów trzeba zastosować i jak te elementy są łączone pomiędzy sobą – można się zastanowić, jak tę sieć zrealizować. W najwcześniejszych pracach dotyczących budowy sieci neuronowych chętnie stosowano urządzenia elektroniczne, które modelowały sieć. Przykład tego typu układu przedstawiono na rysunku 22.



Rysunek 22.

Sieć neuronowa zbudowana z indywidualnych elementów elektronicznych [źródło: <http://www.isn.ucsd.edu/papers/asic96/img50.gif>, dostęp wrzesień 2011]

Obecnie urządzenia elektroniczne służące do budowy sieci neuronowych (rzadko jednak budowanych) mają postać specjalizowanych układów scalonych (rys. 23).



Rysunek 23.

Elektroniczny układ scalony służący do modelowania sieci neuronowych wyprodukowany przez Intel [źródło: <http://www.warthman.com/images/intel%2080170%20B.jpg>]

Najczęściej jednak do budowy sieci neuronowych wykorzystuje się program komputerowy, który w zwykłym komputerze modeluje sieć (rys. 24). Komputery potrafią modelować różne obiekty (na przykład statek kosmiczny albo pogodę), więc mogą także modelować sieć neuronową, jeśli jest ona dobrze opisana.



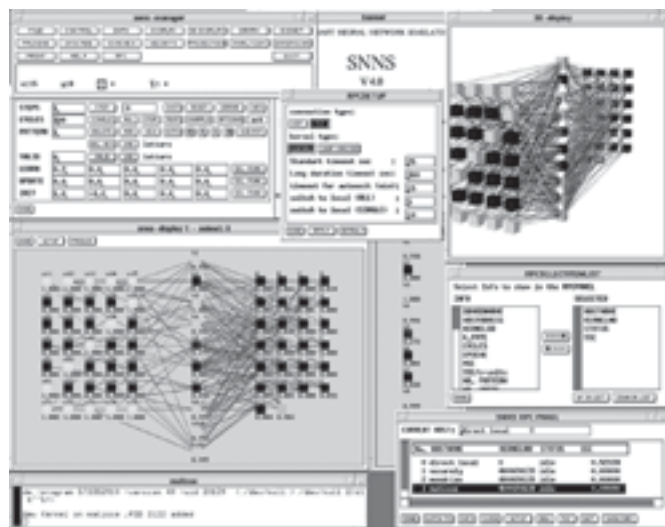
Rysunek 24.

Modelowana komputerowo sieć neuronowa jest szczególnie wygodna i łatwa w użyciu

Nazwy i loga przykładowych programów modelujących sieci neuronowe są podane na rysunku 25, a przykładowy wygląd ekranu komputera modelującego sieć neuronową pokazano na rysunku 26. Przedstawione programy bardzo dobrze modelują sieci neuronowe, mające jednak jedną poważną wadę: są bardzo drogie, więc kupują je tylko instytucje, które stosują sieci neuronowe.



Rysunek 25. Nazwy i loga przykładowych programów modelujących sieci neuronowe [źródło: 2]

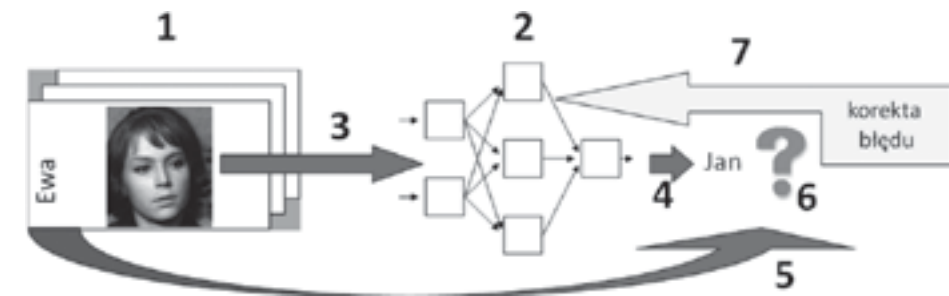


Rysunek 26. Przykładowy wygląd ekranu komputera modelującego sieć neuronową [źródło: <http://cortex.cs.nuim.ie/tools/spikeNNS/images/snns-picture.gif>, dostęp wrzesień 2011]

5 PROCES UCZENIA SIECI NEURONOWEJ

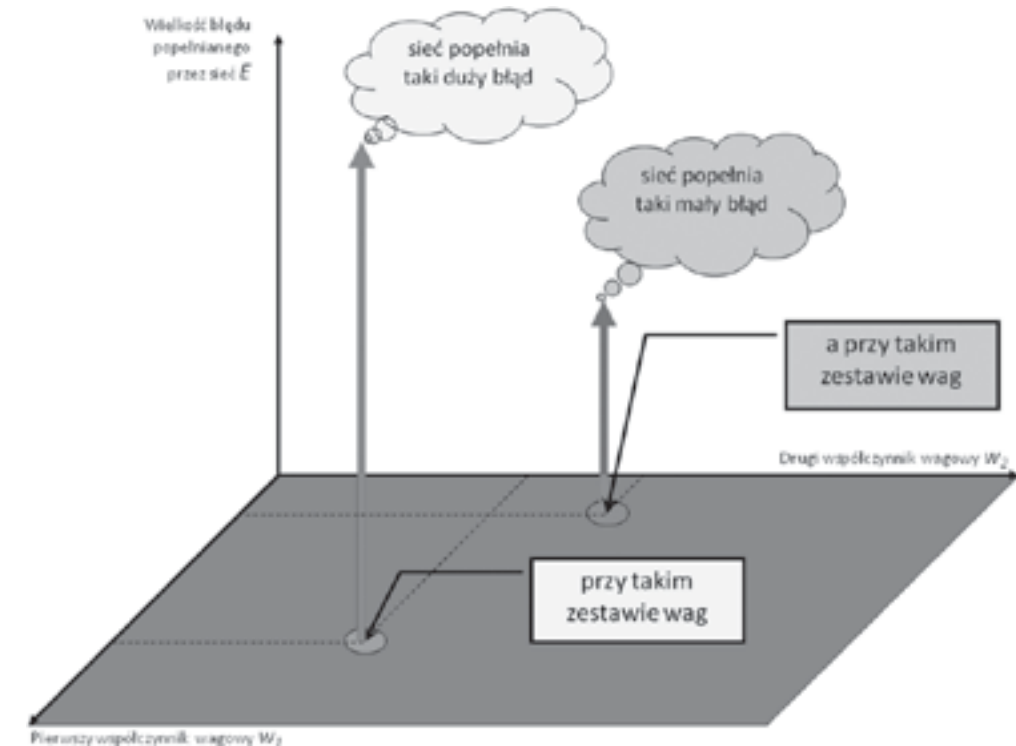
Przystąpimy teraz do omówienia procesu **uczenia** sieci neuronowych. Będziemy odwoływać się przy tym do numerów zaznaczonych na rysunku 27, który przedstawia maksymalnie uproszczony schemat tego procesu. Jego podstawą jest zbiór przykładowych danych wraz z rozwiązaniami, nazywany **zbiorem uczącym** (1). W przykładzie zbiór ten zawiera wizerunki osób, których sieć ma się nauczyć rozpoznawać oraz informacje, jakie jest poprawne rozwiązanie (tzn. kim jest osoba na zdjęciu). Uczenie polega na pokazywaniu sieci (2) kolejnych zadań – wizerunków osób (3), które sieć próbuje rozpoznać podając własne rozwiązania (4). W zbiorze uczącym są informacje o tym, jak naprawdę

nazywa się osoba na zdjęciu (5). Porównanie odpowiedzi sieci z prawidłowym rozwiązaniem umożliwia wyznaczenie błędu sieci (6). Uczenie prowadzone jest tak, żeby zminimalizować wartość błędu (7).



Rysunek 27. Uczenie polega na takim poprawianiu parametrów sieci (wartości wag we wszystkich neuronach całej sieci), żeby krok po kroku, zadanie po zadaniu, zmniejszać błąd popełniany przez sieć. Kolejne numery oznaczają kolejne czynności [źródło: 2]

Pytanie, które się nasuwa, jest następujące: Skąd wiemy, w jaki sposób zmieniać parametry sieci (wartości wag), żeby uzyskać efekt zmniejszania błędu?



Rysunek 28. Zależność błędu popełnianego przez sieć od współczynników wagowych

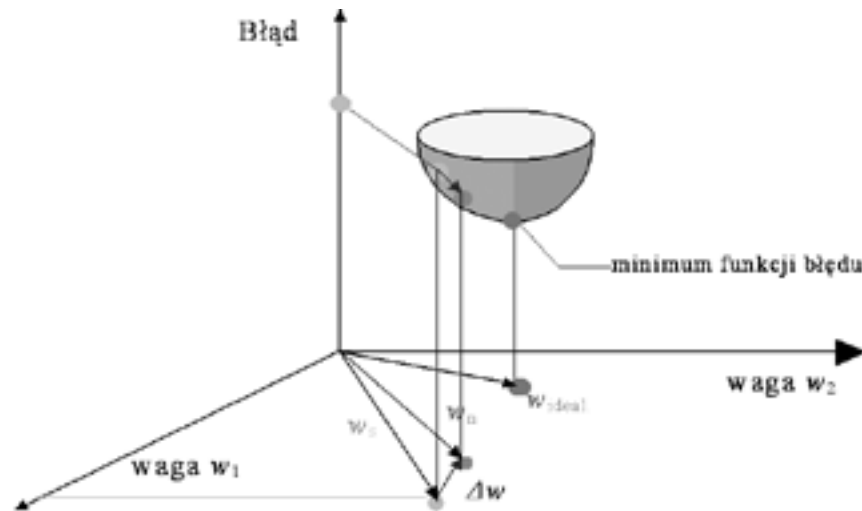
Zachowanie sieci jest wypadkową zachowania wszystkich jej neuronów, zaś zachowanie poszczególnych neuronów można uzależnić od wartości wag występujących w tych neuronach (por. rys. 14 i 15). Jeśli więc ustalimy wszystkie wagi we wszystkich neuronach całej sieci, a potem pokażemy sieci wszystkie zadania ze

zbioru uczącego, to wyznaczmy łączny błąd, popełniany przez sieć dla tych zadań. Dla różnych zestawów wag otrzymamy różne wartości łącznego błędu (rys. 28). Gdybyśmy takie strzałki wystawiali we wszystkich punktach ciemnej płaszczyzny, to powstałaby powierzchnia nazywana **powierzchnią błędu**, której za chwilę użyjemy do wyznaczenia sposobu uczenia. Najpierw jednak wyjaśnimy pewne wątpliwości, jakie się mogą nasuwać przy analizowaniu rysunku 28.

Po pierwsze w sieci zawierającej dużo neuronów (co jest regułą!) istnieje bardzo dużo współczynników wagowych, które trzeba ustalić w toku uczenia – a na rysunku pokazano tylko dwa. Czy to jest prawidłowe? Nie, to nie jest prawidłowe, ale takie uproszczenie trzeba przyjąć, aby umożliwić odzwierciedlenie tego na rysunku. Rysunek jest więc intuicyjną metaforą, a nie dokładną ilustracją rzeczywistych procesów zachodzących w prawdziwej sieci neuronowej.

Po drugie jak wyznaczyć łączny błąd sieci (dla danego zestawu wag), skoro w zbiorze uczącym jest wiele zadań, a w każdym z tych zadań sieć popełnia inny błąd? Otóż błędy ustalone dla poszczególnych zadań trzeba do siebie dodać, żeby błędy ujemne nie zmniejszyły błędów dodatnich – wszystkie błędy przed sumowaniem podnosimy do kwadratu. Dzięki temu błędy są tylko dodatnie, a w dodatku dla dużych błędów jest większa „kara” niż dla małych, bo po podniesieniu do kwadratu trzy razy większy błąd oznacza dziewięć razy większą karę.

Na rysunku 29 pokazano przykładową powierzchnię błędu (szara) oraz uczenie sieci jako poszukiwanie minimum funkcji błędu. Istota uczenia polega na szukaniu miejsca (zestawu wag określonego jako w_{ideal}), w którym błąd jest minimalny. Na rysunku zaznaczony jest stary zestaw wag w_s (przed wykonaniem jednego kroku procesu uczenia), któremu odpowiada duża wartość błędu (oznaczona kropką na powierzchni błędu). Metoda uczenia potrafi na powierzchni błędu wyznaczyć kierunek najszybszego malenia błędu (krótka strzałka na płaszczyźnie). Na tej podstawie stary zestaw wag w_s zostaje zmodyfikowany o wartość Δw i powstaje nowy zestaw wag w_n , który jest bliższy zestawowi idealnemu w_{ideal} .



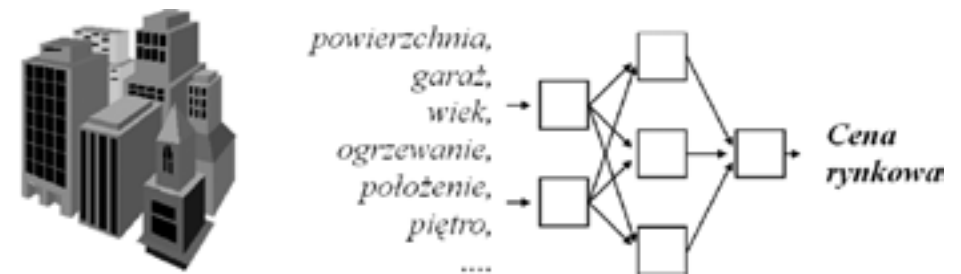
Rysunek 29. Uczenie sieci jako poszukiwanie minimum funkcji błędu

Szczegółowe algorytmy uczenia wbudowane są zwykle w programy symulujące sieci neuronowe na komputerze, dlatego użytkownik sieci nie musi się tym osobiście zajmować. I dobrze, bo przecież opisana wyżej ogólna idea sposobu uczenia jest dość prosta i może być łatwo zrozumiana, a jednak szczegóły praktycznej realizacji tej metody są dość skomplikowane i wymagają sporego wysiłku. Na szczęście nie naszego wysiłku tylko komputera.

6 ZASTOSOWANIA SIECI NEURONOWYCH

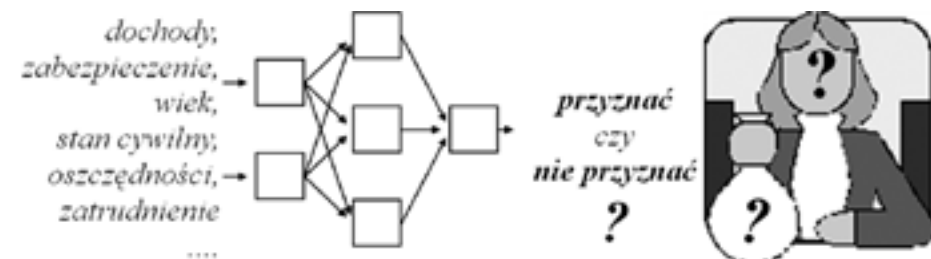
Sieci neuronowe mają wiele zastosowań. Przedstawimy krótko tylko dwa przykłady.

Tworzenie modelu procesu. Wyobraźmy sobie, że chcemy przewidzieć, jaką cenę osiągnie na wolnym rynku określone mieszkanie (rys. 30). Nie mamy gotowych reguł (bo nikt ich nie zna), ale możemy użyć jako zbioru uczącego opisu wcześniejszych transakcji kupna-sprzedaży. Na wejściu sieci są dane dotyczące mieszkania, a sieć ma podać jego cenę. Rozważając ten przykład warto zwrócić uwagę, że na wejściu sieci mogą się pojawić zarówno informacje dające się wyrazić ilościowo (na przykład powierzchnia mieszkania), jak i takie dane, które są opisowe (na przykład położenie, czyli informacja, w jakiej dzielnicy jest rozważane mieszkanie). Sieci neuronowe radzą sobie z danymi wszelkich typów.



Rysunek 30. Zastosowanie sieci neuronowej do tworzenia modelu procesu

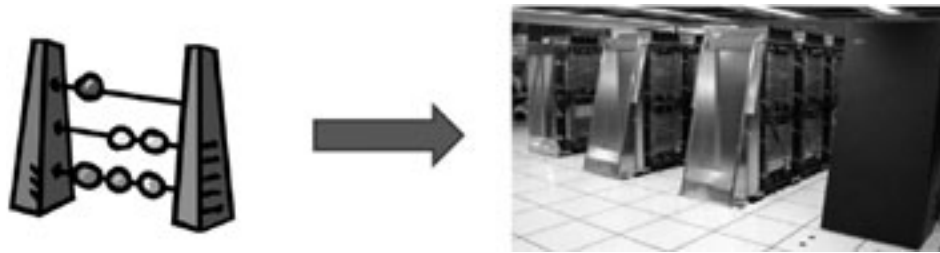
Podejmowanie decyzji. W banku trzeba zdecydować, czy przyznać konkretnemu klientowi pożyczkę, czy lepiej nie (rys. 31). Jak się nie pożyczyci pieniędzy uczciwemu klientowi, to bank nie zarobi. Ale jak się pożyczyci nieuczciwemu, to bank poniesie stratę. Nie wiadomo, po czym poznać nierzetelnego klienta, ale można dać sieci jako zbiór uczący informacje o wszystkich udzielonych pożyczkach, tych udanych i nie. Sieć się sama nauczy rozpoznawać nieuczciwych klientów i może nam radzić. Rozważając ten przykład warto podkreślić, że wielokrotnie wzmiankowane wcześniej zagadnienia rozpoznawania (na przykład ludzkich twarzy) są przykładem podejmowania decyzji przy pomocy sieci neuronowej.



Rysunek 31. Zastosowanie sieci neuronowej do wspomaganie procesu podejmowania decyzji

ZAKOŃCZENIE

Sieci neuronowe powstały w wyniku procesu twórczego, przeciwnego do tego, który doprowadził do powstania typowych komputerów. Komputery powstały bowiem w taki sposób, że stosunkowo proste (początkowo) urządzenia przeznaczone do mechanizacji obliczeń: liczydła, suwaki, kalkulatory itd. poddano procesowi intensywnego doskonalenia, dzięki czemu powstały znane nam obecnie systemy informatyczne, o ogromnych możliwościach, ale też niezwykle skomplikowane (rys. 32).



Rysunek 32.
Ewolucja informatyki – od liczydła do superkomputera

Z sieciami neuronowymi było przeciwnie: Za punkt wyjścia przyjęto niestłuchanie skomplikowany twór, jakim jest mózg, i podjęto próbę modelowania jego struktury i właściwości za pomocą opisów, które w miarę ich doskonalenia stawały się coraz prostsze. Obecnie używane sieci neuronowe są tak bardzo uproszczone, że każdy może zrozumieć ich budowę i działanie, a jednocześnie zachowały one tyle właściwości oryginalnego mózgu, że potrafią się bardzo inteligentnie zachowywać (rys. 33).



Rysunek 33.
Ewolucja neurocybernetyki – od mózgu do sieci neuronowej

Zasadniczą cechą użytkową, odróżniającą sieci neuronowe od typowych, ogólnie znanych i powszechnie stosowanych komputerów, jest ich zdolność do samodzielnego nabywania wiedzy w procesie uczenia się. Komputery mogą bardzo szybko i dokładnie wykonywać rozmaite czynności, czasem bardzo skomplikowane i ogromnie użyteczne, ale robią to tylko wtedy, gdy człowiek wcześniej dokładnie określi, co i jak mają robić. To człowiek zasila komputer wiedzą, tworząc algorytm i pisząc na jego podstawie program. Natomiast sieci neuronowe nie wymagają programowania. Wystarczy, że pokażemy sieci trochę przykładów poprawnie rozwiązanych zadań, a sieć sama zgromadzi potrzebną wiedzę i potrafi potem rozwiązywać podobne zadania. Jest to bardzo wygodne, a ponadto umożliwia rozwiązywanie także takich zadań, dla których nikt nie potrafi napisać algorytmu! Dlatego sieci neuronowe są dziś bardzo chętnie stosowane i dlatego warto je poznać jako fascynujące narzędzia nowoczesnej informatyki.

Jednak nie tylko sprawność działania i wygoda stosowania powoduje, że sieciami neuronowymi zajmuje się coraz większe grono badaczy i praktyków na całym świecie. Dodatkowy powód jest taki, że mimo ogromnych uproszczeń sieci te zachowały wiele elementów podobieństwa do naszego mózgu (od którego badania zaczęła się droga, która doprowadziła do powstania tych sieci). Dlatego używając sieci i obserwując procesy w nich zachodzące poznajemy także jedną z najbardziej fascynujących tajemnic Natury: zagadkę ludzkiego intelektu...

LITERATURA

1. Tadeusiewicz R. (red.), *Neurocybernetyka teoretyczna*, Wydawnictwo Uniwersytetu Warszawskiego, Warszawa 2009
2. Tadeusiewicz R., Gąciarz T., Borowik B., Leper B., *Odkrywanie właściwości sieci neuronowych przy użyciu programów w języku C#*, Wydawnictwo Polskiej Akademii Umiejętności, Kraków 2007; <http://home.agh.edu.pl/~tad/>
3. Tadeusiewicz R., *Elementarne wprowadzenie do sieci neuronowych z przykładowymi programami*, Akademicka Oficyna Wydawnicza, Warszawa 1998; <http://winntbg.bg.agh.edu.pl/skrypty2/0263/>
4. Tadeusiewicz R., *Sieci neuronowe*, Akademicka Oficyna Wydawnicza, Warszawa 1993; (wyd. I i II wydanie poprawione); <http://winntbg.bg.agh.edu.pl/skrypty/0001/>