SIEĆ NEURONOWA JAKO NARZĘDZIE APROKSYMACJI I KLASYFIKACJI DANYCH

Jakub Karbowski
Gimnazjum nr 17 w Krakowie

KRAKÓW 2017

1. Spis treści

2. Wstęp

W pracy przedstawiłem możliwości zastosowania sieci neuronowych w zagadnieniach aproksymacji i klasyfikacji. Zamieściłem przykłady rozpoznawania gatunków win oraz aproksymacji wydajności kart graficznych.

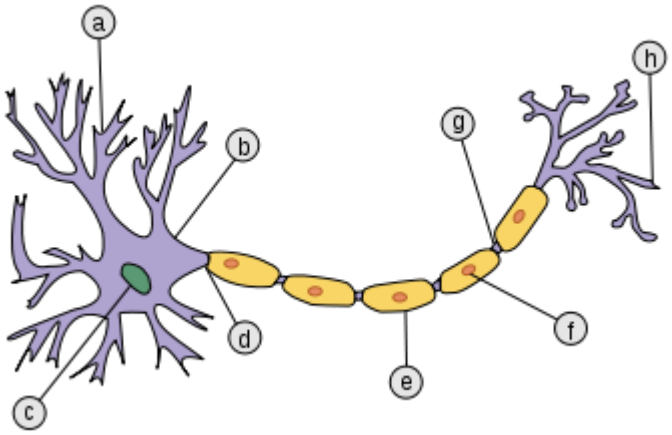
Sieci neuronowe to bardzo szeroka dziedzina nauki, dlatego skupiłem się na wykorzystaniu nadzorowanych jednokierunkowych sieci neuronowych uczonych algorytmem wstecznej propagacji błędu.

Wykorzystałem dane pochodzące z internetu oraz dane zebrane przeze mnie.

3. Sieci neuronowe

3.1. Co to są sieci neuronowe

Sieć neuronowa - ogólna nazwa struktur matematycznych i ich programowych lub sprzętowych modeli, realizujących obliczenia lub przetwarzanie sygnałów poprzez rzędy elementów, zwanych sztucznymi neuronami, wykonujących pewną podstawową operację na swoim wejściu. Oryginalną inspiracją takiej struktury była budowa naturalnych neuronów, łączących je synaps, oraz układów nerwowych, w szczególności mózgu [1].



Rys. 2.1. Schemat budowy neuronu: a – dendryty, b – ciało komórki, c – jądro komórkowe, d – akson, e – otoczka mielinowa, f – komórka Schwanna, g – przewężenie Ranviera, h – zakończenia aksonu [2].

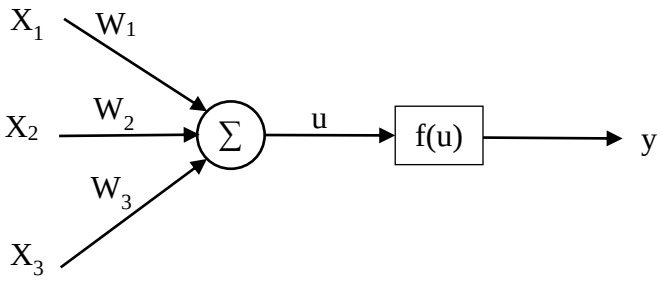
3.2. Dlaczego sieci neuronowe?

Sieci neuronowe są głównie stosowane w przypadku, kiedy musimy wyznaczyć bardzo skomplikowaną funkcję, która ma dziesiątki, setki, a nawet tysiące argumentów. Zrobienie tego ręcznie jest praktycznie niemożliwe, ze względu na poziom skomplikowania danych.

Sieci neuronowe pozwalają na automatyczne wyznaczanie takich funkcji z minimalnym udziałem człowieka.

3.3. Struktura sieci neuronowych

Wzorując się na strukturze biologicznej, pokazanej na rys.2.1 można opracować model matematyczny sztucznego neuronu (rys. 2.2).



Rys.2.2. Model matematyczny neuronu

Wartość wyjściowa neuronu obliczana jest według wzoru:

$$y = f(u) \qquad (2.1)$$

$$u = \sum_{i=1}^{N} w_i x_i \qquad (2.2)$$

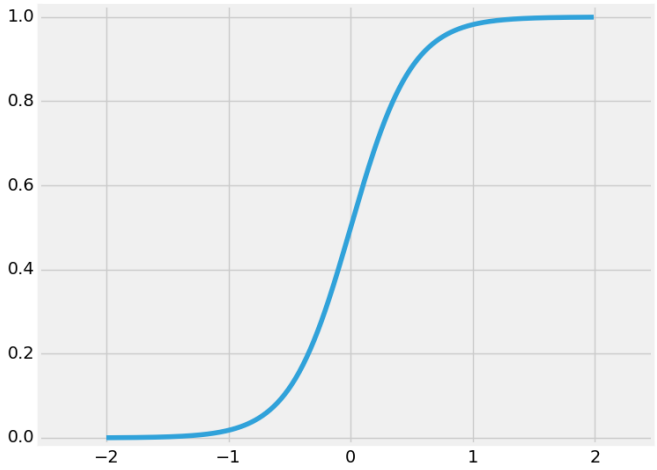
gdzie:
x_i – wartości wejściowe neuronu,
w_i – współczynniki wagowe neuronu,
f(u) – funkcja aktywacji neuronu.

Najczęściej stosowane funkcje aktywacji to:

- funkcja sigmoidalna zwana funkcją logistyczną [3]

$$f(u) = \frac{1}{1+e^{-au}} \qquad (2.3)$$

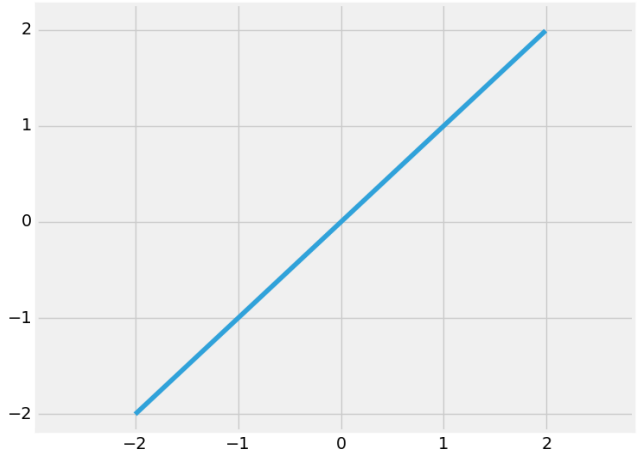
gdzie a jest parametrem funkcji



Rys. 2.3. Wykres sigmoidalnej funkcji aktywacji dla a = 4

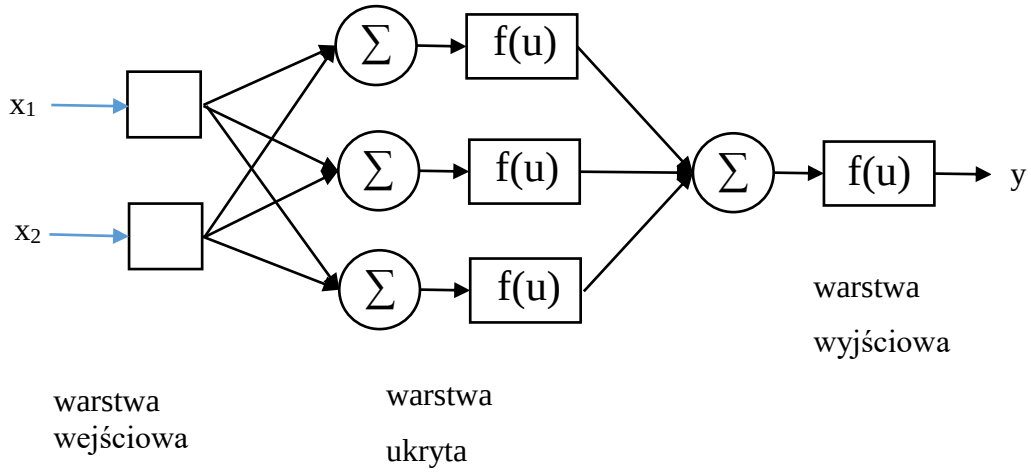
- funkcja liniowa

$$f(u) = a \cdot u \qquad (2.4)$$



Rys. 2.4. Wykres liniowej funkcji aktywacji dla a = 1

Z pojedynczych neuronów buduje się struktury nazywane sieciami neuronowymi (rys.2.5).



Rys. 2.5. Schemat przykładowej sieci neuronowej

W tym przykładzie sieć ma dwa wejścia, trzy neurony w warstwie ukrytej i jedno wyjście. Każdy neuron przekazuje swoją wartość pomnożoną przez daną wagę do wszystkich neuronów w kolejnej warstwie. Im więcej neuronów w sieci, tym bardziej skomplikowane funkcje możemy wyznaczać.

Warstwa wejściowa normalizuje wielkości wejściowe, to znaczy skaluje zakresy wejściowe najczęściej w przedziale [-1,1]. Dzieje się tak, ponieważ funkcja sigmoidalna dla wartości zbyt dużych lub zbyt małych, przyjmuje wartość bliską 0 lub 1.

Sygnał jest przesyłany do warstw ukrytych. Ostatnia warstwa nazywana jest warstwą wyjściową. Działa ona tak samo jak warstwy ukryte, ale jest ona również wyjściem sieci.

Poniżej przedstawiłem wzór na tę sieć, gdzie:

- y - wyjście sieci,
- x_i - wartości wejściowe,
- w_i - wartości wagowe.

$$y = f(f(x_1w_1 + x_2w_2) \cdot w_7 + f(x_1w_3 + x_2w_4) \cdot w_8 + f(x_1w_5 + x_2w_6) \cdot w_9) \qquad (2.5)$$

3.4. Uczenie sieci neuronowych

Zadaniem neuronu jest przetwarzanie sygnałów. Neuron jest funkcją, której współczynniki wagowe należy obliczyć. Obliczanie współczynników wagowych nazywane jest uczeniem neuronu [4]. Jest to proces iteracyjny. Wartości wejściowe (x_i) wysyłane są do neuronu, który je przetwarza i oblicza wartość wyjściową (y). Wartość ta porównywana jest z oczekiwaną wartością wyjściową (z) (tab.2.1).

Tab. 2.1. Tabela wartości wejściowych i wyjściowych

	x_1	x_2	x_3	z
1	x_{11}	x_{21}	x_{31}	z_1
2	x_{12}	x_{22}	x_{32}	z_2
…	…	…	…	…
n	x_{1n}	x_{2n}	x_{3n}	z_n

Współczynniki wagowe oblicza się według wzoru nazywanego regułą delta [4]:

$$w_i^{(j+1)} = w_i^{(j)} + \eta x_i \frac{df(u)}{du} \delta^{(j)} \qquad (2.5)$$

gdzie:

i – numer wielkości wejściowej,

j – numer kroku uczenia,

η – współczynnik uczący (learning rate); liczba z przedziału [0,1], której wartość ustalana jest przez nadzorującego proces uczenia,

$\frac{df(u)}{du}$ – pochodna funkcji aktywacji neuronu,

$\delta^{(j)} = z^{(j)} - y^{(j)}$ – błąd na wyjściu neuronu.

Do wzoru (2.5) podstawia się wartości z tabeli 2.1. Po przejściu przez wszystkie dane proces powtarza się od początku tabeli. Ponieważ jest to proces iteracyjny, wymagane jest zdefiniowanie kryterium zakończenia. Może to być liczba powtórzeń lub dopuszczalna wartość błędu na wyjściu neuronu.

W podobny sposób uczy się warstwy ukryte, ale występuje tu problem z obliczaniem błędu, ponieważ nie znamy ich oczekiwanych wartości wyjściowych. Zostało to rozwiązane przez algorytm wstecznej propagacji błędu [4]. Można zauważyć, że na wartość błędu neuronu wyjściowego składają się błędy neuronów warstwy wcześniejszej. Wsteczna propagacja błędu polega na obliczeniu wartości błędów neuronów ukrytych poprzez pomnożenie wartości błędu neuronu wyjściowego przez wartości współczynników wagowych. W rezultacie otrzymuje się wartości propagowanych wstecz błędów dla neuronów warstwy wcześniejszej, które można wykorzystać we wzorze (2.5).

Opisane wcześniej kryteria zakończenia procesu uczenia sieci mogą spowodować tak zwane przeuczenie, to znaczy sieć będzie bardzo dobrze rozpoznawała wzorce, które poznała podczas procesu uczenia, ale nie będzie prawidłowo aproksymowała danych. Z tego względu w napisanym przeze mnie programie komputerowym zastosowałem inna metodę uczenia. Zbiór danych, który ma posłużyć do uczenia sieci dzielony jest na dwa podzbiory – zbiór uczący i zbiór testowy. Sieć uczona jest na danych ze zbioru uczącego. Co pewien czas proces uczenia jest przerywany, a do sieci wysyłane są dane ze zbioru testowego. Wynik obliczony przez sieć porównywany jest z wartościami wyjściowymi zapisanymi w zbiorze testowym. Na tej podstawie obliczany jest błąd działania sieci (wskaźnik efektywności sieci) według następującego algorytmu:

1. program kolejno oblicza wartości wyjściowe sieci (y) dla zbioru testowego,
2. następnie oblicza błąd na wyjściu sieci: $e = |y - z|$, gdzie z jest wartością ze zbioru testowego,

3. wskaźnik efektywności sieci obliczany jest ze wzoru: $E = \frac{k}{N} \cdot 100\%$, gdzie k jest liczbą wystąpień sytuacji, gdy $e < t$; t jest dopuszczalną wartością błędu; N = (liczebność zbioru testowego) · (liczba neuronów wyjściowych).

Jeśli liczba powtórzeń cyklu uczenie-testowanie, które nie spowodowały zmniejszenia błędu działania sieci jest większa od ustalonej wartości proces uczenia jest przerywany.

4. Aproksymacja i klasyfikacja danych

Aproksymacja to zastąpienie wielkości matematycznych innymi, o przybliżonych własnościach, łatwiejszymi do badania i zastosowania.

Klasyfikacja danego zjawiska to systematyczny podział czegoś (np.: przedmiotów lub zjawisk) na klasy, działy, poddziały itp. wg. określonej zasady.

5. Cel i zakres pracy

Celem pracy jest pokazanie możliwości sieci neuronowych w zakresie aproksymacji i klasyfikacji danych.

Zakres pracy obejmuje:

1. opracowanie programu komputerowego w języku Python do uczenia, symulacji i wizualizacji sieci neuronowych,
2. zastosowanie sieci neuronowej do klasyfikacji danych enologicznych [6] – klasyfikacja gatunków wina na podstawie właściwości fizycznych i chemicznych,
3. opracowanie programu do pobierania danych ze stron internetowych, które będą wykorzystane do uczenia sieci przewidującej wydajności kart graficznych,
4. zastosowanie sieci neuronowej do przewidywania wydajności kart graficznych.

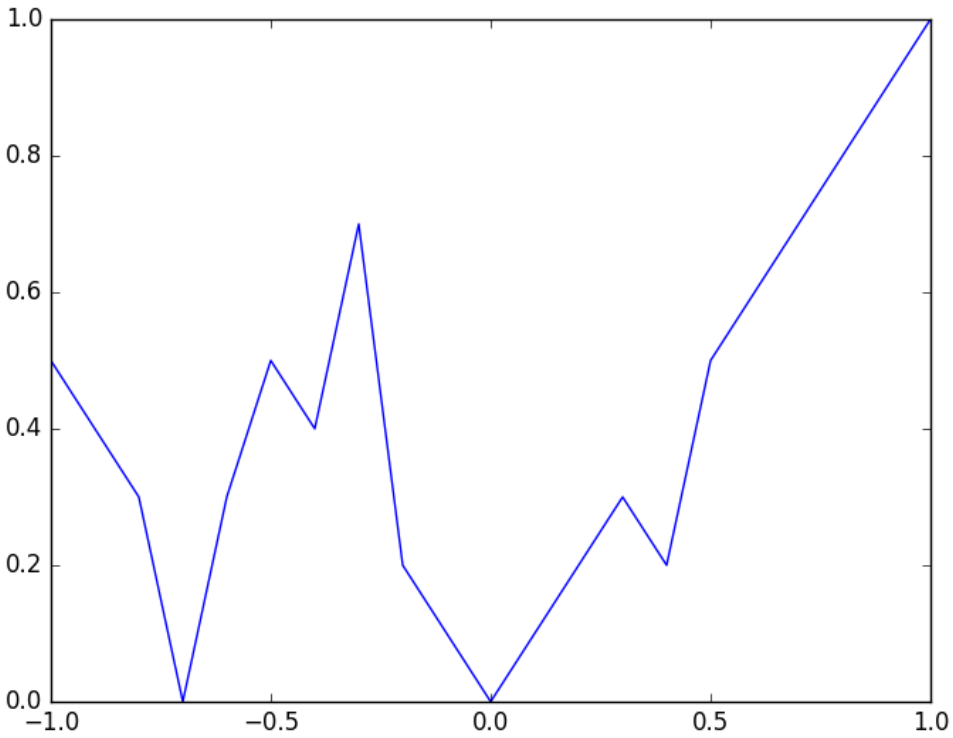
6. Badania własne

6.1. Program komputerowy do symulacji i wizualizacji sieci neuronowych

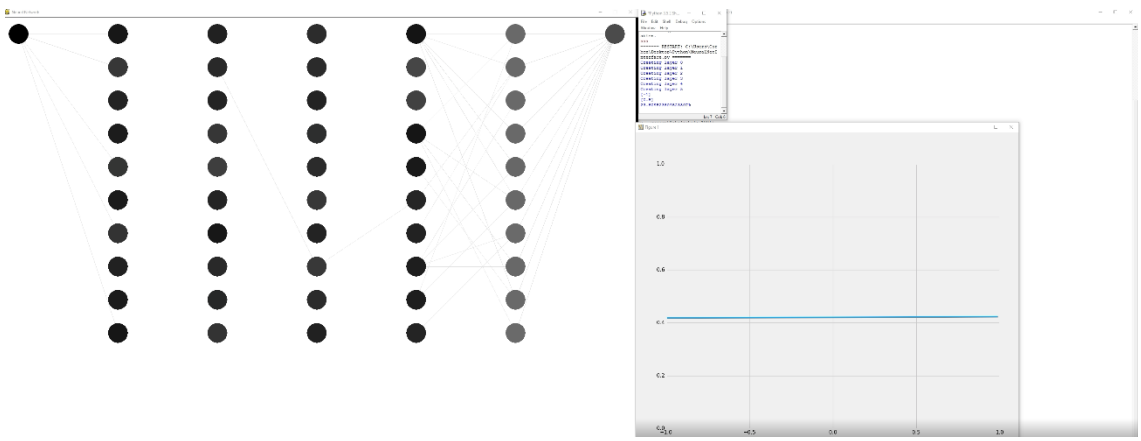
Dla potrzeb pracy przygotowałem program do symulacji i wizualizacji sieci neuronowych. Do programowania użyłem języka Python. Ponieważ jest on zrobiony przeze mnie od podstaw mam wpływ na wszystko, co dzieje się w mojej sieci. Cały program został umieszczony za dołączonej płycie CD.

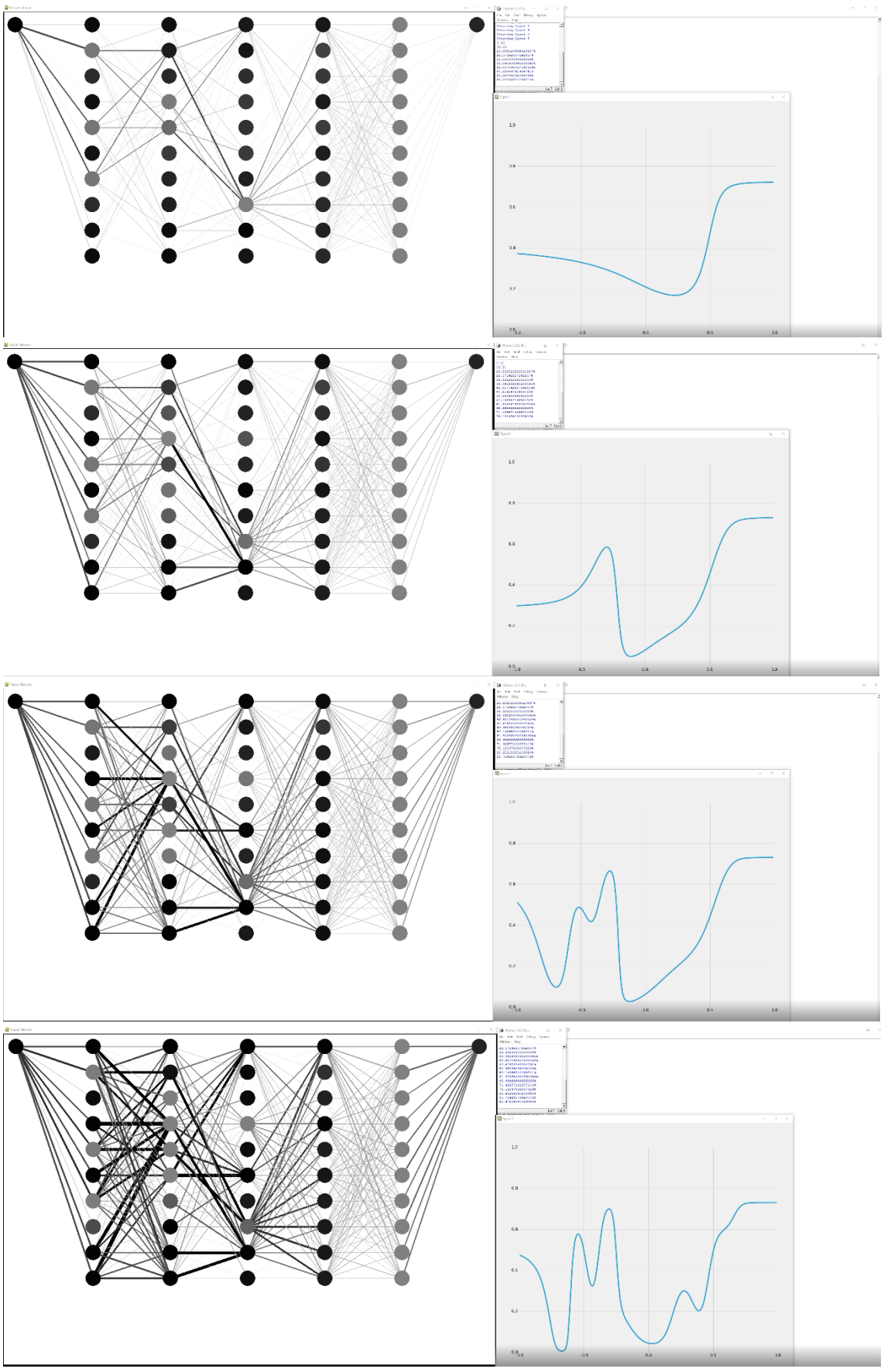
6.2. Testowanie programu na przykładowej funkcji

Aby przetestować mój program, postanowiłem zaproksymować następującą funkcję.



Poniżej przedstawiłem kolejne etapy uczenia sieci, będącej funkcją aproksymującą. Grubość linii jest proporcjonalna do wartości bezwzględnej współczynnika wagowego. Na dołączonej płycie CD zamieściłem animację procesu uczenia sieci – nazwa pliku „Animacja.mp4”.





Rys. 5.1. Wizualizacja procesu uczenia sieci

Każda kropka to jeden neuron, a każda linia to współczynnik wagowy łączący dwa neurony. Jej grubość i kolor zależy od wartości współczynnika wagowego.

Jak można zauważyć, z każdym krokiem funkcja (sieć neuronowa) przyjmuje wartość coraz bardziej zbliżoną do docelowej. Na początku funkcja jest prawie płaska, ponieważ proces wstecznej propagacji błędu dopiero kilka razy zmodyfikował wagi według wzoru 2.5. Pod koniec treningu funkcja jest bardzo podobna do funkcji docelowej. Po nauczeniu sieci można do wejścia podać dowolną wartość a funkcja zaproksymuje wartość wyjściową.

6.3. Zastosowanie sieci neuronowej do klasyfikacji danych enologicznych

W rozdziale tym pokażę przykład zastosowania sieci neuronowej do klasyfikacji gatunków win. Dane do uczenia sieci pochodzą z internetu [5].

Wielkościami wejściowymi sieci są między innymi:

- zawartość alkoholu,
- zawartość kwasu jabłkowego,
- ilość osadów,
- zasadowość osadów,
- zawartość magnezu,

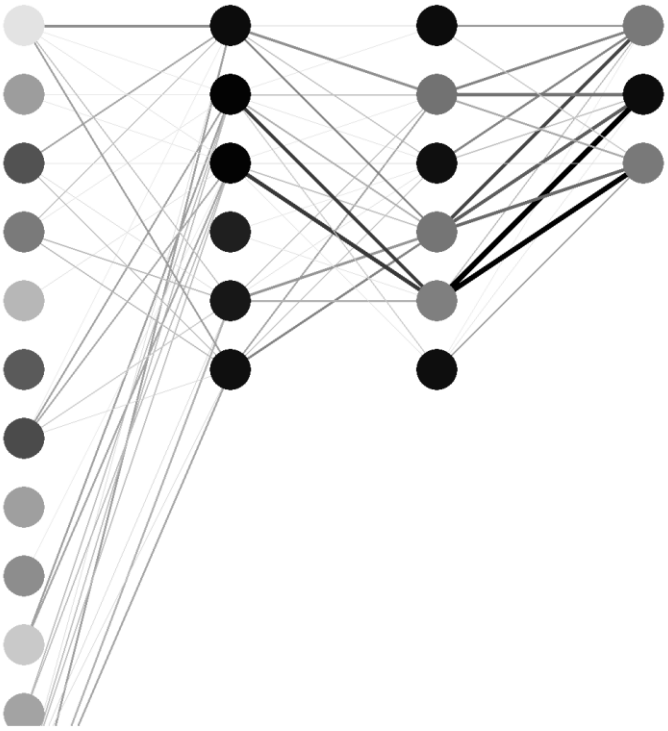
Specjaliści od win, czyli sommelierzy dokonują klasyfikacji win w trzech grupach (klasach) numerowanych od 1 do 3. W tym celu posługują się zmysłem smaku. Zaprojektowana przeze mnie sieć zastępuje specjalistę i dokonuje klasyfikacji na podstawie zmierzonych właściwości chemicznych i fizycznych wina.

Dane uczące mają 13 wejść, a sieć musi odróżniać 3 klasy. W przypadku problemu klasyfikacji, każda klasa ma swój własny neuron wyjściowy. Jeśli sieć przewidziała klasę 2, to wyścia będą mogły wyglądać następująco:

y_1	y_2	y_3
0.054343637	0.8512521	0.1564573

Jak można zauważyć, dominująca wartość wskazuje wynik klasyfikacji.

Zbiór uczący zawierał 153 zestawy danych wejściowych, a zbiór do testowania 40 zestawów danych.



Rys. 6.1. Architektura sieci

Sieć była uczona metodą opisaną we wcześniejszym rozdziale. Wartość wskaźnika efektywności sieci wyniosła 100%, czyli sieć dobrze obliczyła wszystkie dane z zestawu testowego.

6.4. Zastosowanie sieci neuronowej do przewidywania mocy obliczeniowej kart graficznych

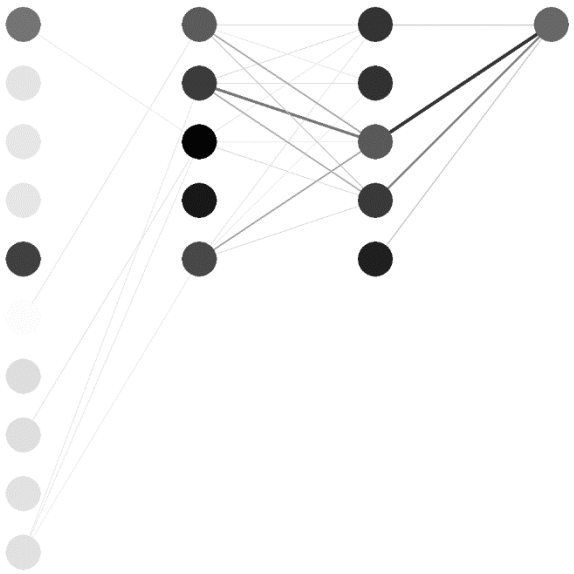
Ostatnim przykładem, który pokażę jest zastosowanie sieci neuronowych do aproksymacji osiągów kart graficznych. Wielkości wejściowe to między innymi:

- prędkość zegara,
- liczba jednostek cieniujących,
- liczba jednostek mapowania tekstur,
- liczba procesorów renderujących,
- prędkość pamięci,
- szerokość interfejsu pamięci,
- przepustowość pamięci.

Sieć oblicza wynik benchmarku 3DMark Fire Strike danej karty.

Napisałem program, który automatycznie zbiera informacje o istniejących kartach graficznych, a następnie przygotowuje zestaw uczący i testowy. Udało mi się

zebrać informacje o 193 kartach. Na zbiór testowy przeznaczyłem 30 kart graficznych. Wskaźnik efektywności sieci E wyniósł 86.6% z dopuszczalnym błędem t = 1300 punktów.



Rys. 6.2. Architektura sieci

Działanie sieci sprawdziłem na danych karty graficznej, której sieć nie znała wcześniej. Dla karty graficznej z mojego komputera (GeForce GTX 1080) sieć przewidziała wynik 23408 punków. W rzeczywistości wynik ten wynosi 22370. A zatem błąd wskazania wynosi 5%.

Opisaną wcześniej sieć wykorzystałem do przewidzenia osiągów karty graficznej, która jest dopiero zapowiadana przez producenta (GeForce GTX 1080 Ti). Sieć przewidziała, że karta ta będzie miała wynik równy 24655 punktów.

7. **Wnioski**

Na podstawie przeprowadzonych przeze mnie badań zauważyłem, że sieci neuronowe:

- są wygodnym narzędziem do aproksymacji funkcji,
- mogą pełnić rolę eksperta zastępującego człowieka (przykład z klasyfikacją win),
- można użyć do przewidywania,
- mogą służyć do aproksymacji funkcji wielu zmiennych.

W moich dalszych pracach chciałbym zastosować sieci neuronowe do analizy obrazów. Planuję użyć bibliotekę Keras, której działanie wstępnie sprawdziłem, a uzyskane rezultaty bardzo mnie zadowalają.

Literatura

[1] Sieć neuronowa. Wikipedia. Dostęp 3.01.2017.

[2] Neuron. Wikipedia. Dostęp 3.01.2017.

[3] Osowski S. Sieci neuronowe w ujęciu algorytmicznym. WNT Warszawa 1996.

[4] Tadeusiewicz R. Sieci neuronowe. Akademicka Oficyna Wydawnicza RM. Warszawa 1993.

[5] Forina, M. et al, PARVUS - An Extendible Package for Data Exploration, Classification and Correlation. Institute of Pharmaceutical and Food Analysis and Technologies, Via Brigata Salerno, 16147 Genoa, Italy. https://archive.ics.uci.edu/ml/datasets/Wine

[6] Enologia – nauka o winie, Wikipedia, Dostęp 3.01.2017