

SYSTEM KLASYFIKACJI GRZYBÓW POD WZGLĘDEM JADALNOŚCI

Biologically Inspired Artificial Intelligence - projekt

KACPER KOWALIK GR. GKIO3 SEKCJA 2.

GŁÓWNE ZAŁOŻENIA (KARTA PROJEKTU)

- System, który dla danych wejściowych opisujących wybrany grzyb ma stwierdzić, czy dany grzyb jest grzybem jadalnym lub trującym.
- Baza danych opisująca grzyby zawiera ponad 8000 rekordów, a każdy rekord składa się z 22 atrybutów.
- Projekt będzie obejmował implementację własnej sieci neuronowej oraz przeprowadzenie szeregu testów służących analizie napisanego klasyfikatora.
- Aplikacja zostanie napisana w języku JAVA.

REPREZENTACJA CECH GRZYBA

Attribute Information: (classes: edible=e, poisonous=p)

- cap-shape: bell=b, conical=c, convex=x, flat=f, knobbed=k, sunken=s
- cap-surface: fibrous=f, grooves=g, scaly=y, smooth=s
- cap-color: brown=n, buff=b, cinnamon=c, gray=g, green=r, pink=p, purple=u, red=e, white=w, yellow=y
- bruises: bruises=t, no=f
- odor: almond=a, anise=l, creosote=c, fishy=y, foul=f, musty=m, none=n, pungent=p, spicy=s
- gill-attachment: attached=a, descending=d, free=f, notched=n
- gill-spacing: close=c, crowded=w, distant=d
- gill-size: broad=b, narrow=n
- gill-color: black=k, brown=n, buff=b, chocolate=h, gray=g, green=r, orange=o, pink=p, purple=u, red=e, white=w, yellow=y
- stalk-shape: enlarging=e, tapering=t
- stalk-root: bulbous=b, club=c, cup=u, equal=e, rhizomorphs=z, rooted=r, missing=?
- stalk-surface-above-ring: fibrous=f, scaly=y, silky=k, smooth=s
- stalk-surface-below-ring: fibrous=f, scaly=y, silky=k, smooth=s
- stalk-color-above-ring: brown=n, buff=b, cinnamon=c, gray=g, orange=o, pink=p, red=e, white=w, yellow=y
- stalk-color-below-ring: brown=n, buff=b, cinnamon=c, gray=g, orange=o, pink=p, red=e, white=w, yellow=y
- veil-type: partial=p, universal=u
- veil-color: brown=n, orange=o, white=w, yellow=y
- ring-number: none=n, one=o, two=t
- ring-type: cobwebby=c, evanescent=e, flaring=f, large=l, none=n, pendant=p, sheathing=s, zone=z
- spore-print-color: black=k, brown=n, buff=b, chocolate=h, green=r, orange=o, purple=u, white=w, yellow=y
- population: abundant=a, clustered=c, numerous=n, scattered=s, several=v, solitary=y
- habitat: grasses=g, leaves=l, meadows=m, paths=p, urban=u, waste=w, woods=d

STRUKTURA BAZY DANYCH

Baza danych opisująca cechy grzybów zapisana jest w formacie .csv

	A	B	C	D	E	F	G
1	class, cap-shape, cap-surface, cap-color, bruises, odor, gill-attachment, gill-sp						
2	p,x,s,n,t,p,f,c,n,k,e,e,s,s,w,w,p,w,o,p,k,s,u						
3	e,x,s,y,t,a,f,c,b,k,e,c,s,s,w,w,p,w,o,p,n,n,g						
4	e,b,s,w,t,l,f,c,b,n,e,c,s,s,w,w,p,w,o,p,n,n,m						
5	p,x,y,w,t,p,f,c,n,n,e,e,s,s,w,w,p,w,o,p,k,s,u						
6	e,x,s,g,f,n,f,w,b,k,t,e,s,s,w,w,p,w,o,e,n,a,g						
7	e,x,y,y,t,a,f,c,b,n,e,c,s,s,w,w,p,w,o,p,k,n,g						
8	e,b,s,w,t,a,f,c,b,g,e,c,s,s,w,w,p,w,o,p,k,n,m						
9	e,b,y,w,t,l,f,c,b,n,e,c,s,s,w,w,p,w,o,p,n,s,m						
10	p,x,y,w,t,p,f,c,n,p,e,e,s,s,w,w,p,w,o,p,k,v,g						
11	e,b,s,y,t,a,f,c,b,g,e,c,s,s,w,w,p,w,o,p,k,s,m						
12	e,x,y,y,t,l,f,c,b,g,e,c,s,s,w,w,p,w,o,p,n,n,g						
13	e,x,y,y,t,a,f,c,b,n,e,c,s,s,w,w,p,w,o,p,k,s,m						
14	e,b,s,y,t,a,f,c,b,w,e,c,s,s,w,w,p,w,o,p,n,s,g						
15	p,x,y,w,t,p,f,c,n,k,e,e,s,s,w,w,p,w,o,p,n,v,u						
16	e,x,f,n,f,n,f,w,b,n,t,e,s,f,w,w,p,w,o,e,k,a,g						

MAPOWANIE DANYCH WEKTORA WEJŚCIOWEGO

```
enum CapShape{b, c, x, f, k, s};  
private CapShape capShape;
```

```
public Integer getCapShapeOrdinalValue() { return capShape.ordinal(); }
```

ordinal

	1	2	3	4	5	6

```
enum CapShape{b, c, x, f, k, s};
```

Dla przykładu powyższa funkcja zwróci numer porządkowy dla ustawionego parametru Cap Shape. Przykładowo, jeżeli parametr capShape będzie elementem c typu wyliczeniowego CapShape to funkcja zwróci wartość: 2. Następnie zanim wartość dla neurona wejściowego zostanie ustawiona zostanie poddana normalizacji.

FUNKCJA NORMALIZUJĄCA

Jako funkcja normalizująca użyta będzie normalizacja min-max.

1. Normalizacja min-max

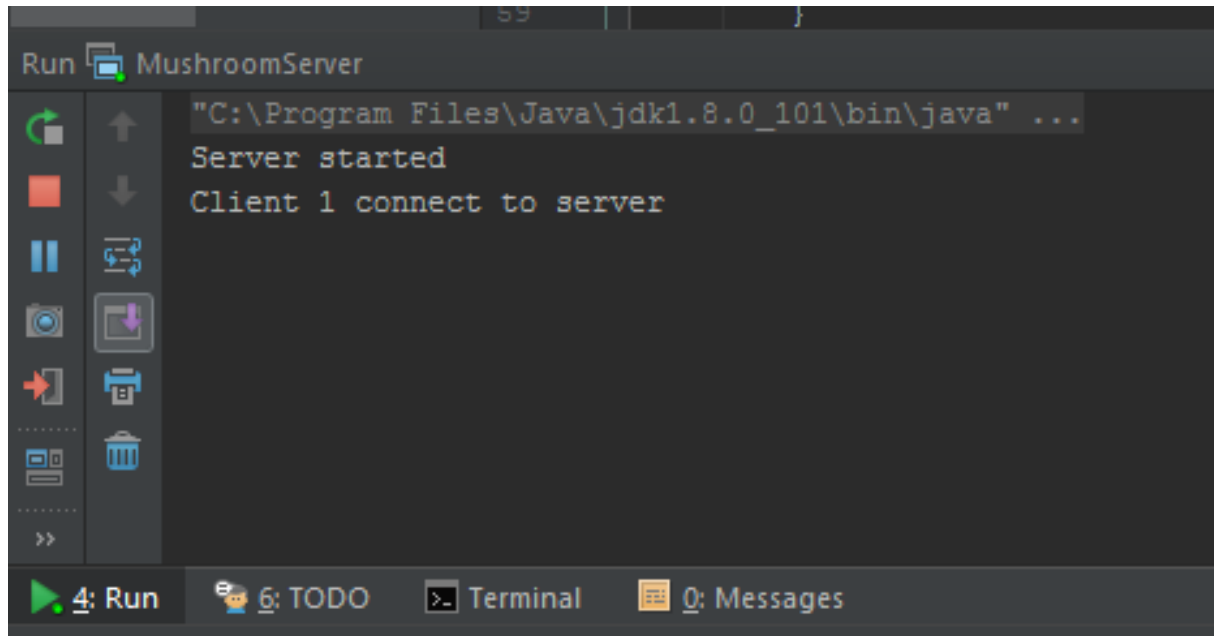
Ta metoda przeprowadza liniową transformację pierwotnych danych najczęściej do przedziału [0,1] według wzoru:

$$V' = \frac{(V - \min)}{\max - \min} * (new_max - new_min) + new_min$$

ARCHITEKTURA SYSTEMU

System składa się z 2 programów w architekturze klient serwer.

Konsolowa aplikacja serwera:



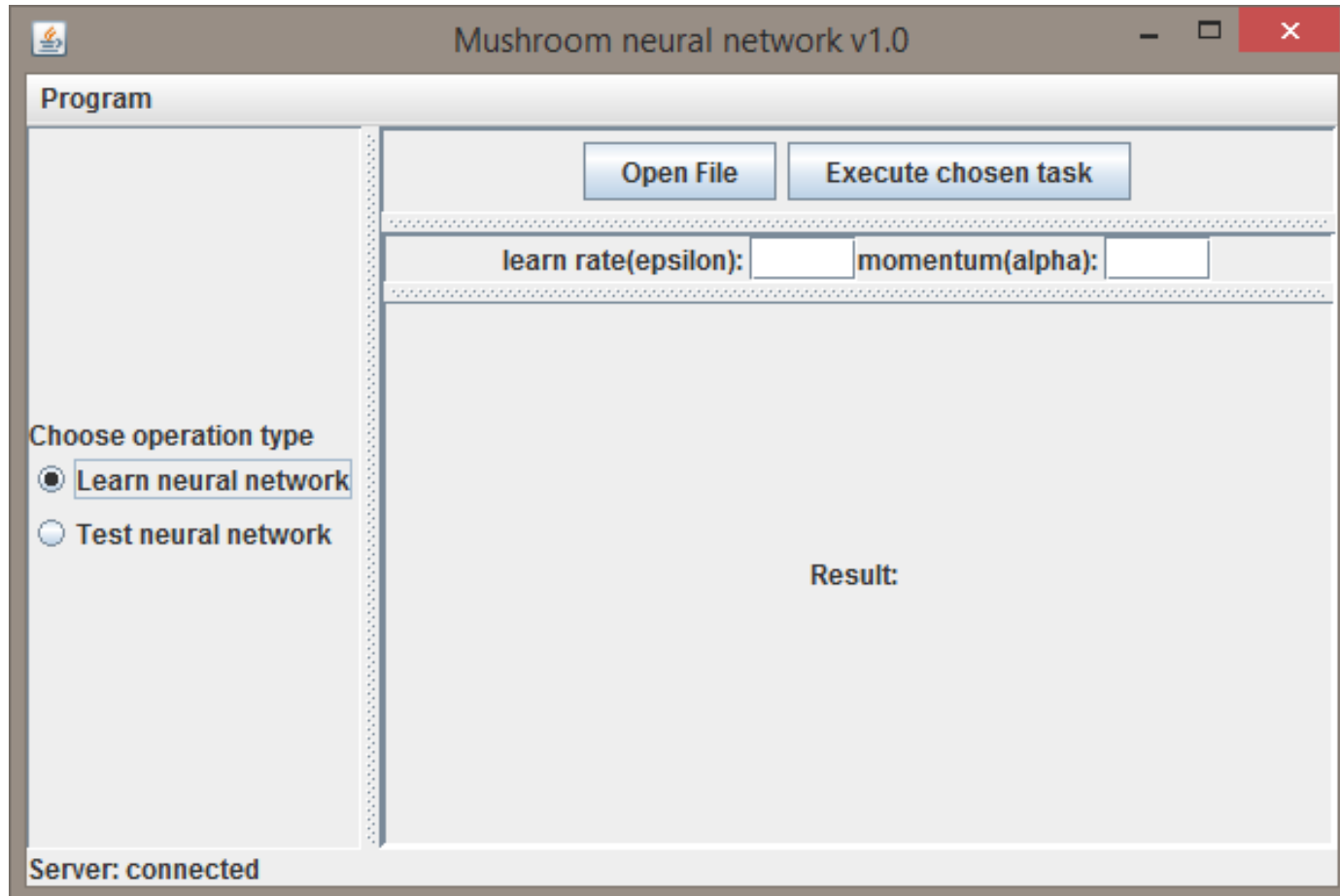
The screenshot shows a console window titled "Run" with a sub-header "MushroomServer". The console output displays the following text:

```
"C:\Program Files\Java\jdk1.8.0_101\bin\java" ...  
Server started  
Client 1 connect to server
```

The console window includes a toolbar on the left with icons for running, stopping, debugging, and other actions. At the bottom, there is a status bar with tabs for "4: Run", "6: TODO", "Terminal", and "0: Messages".

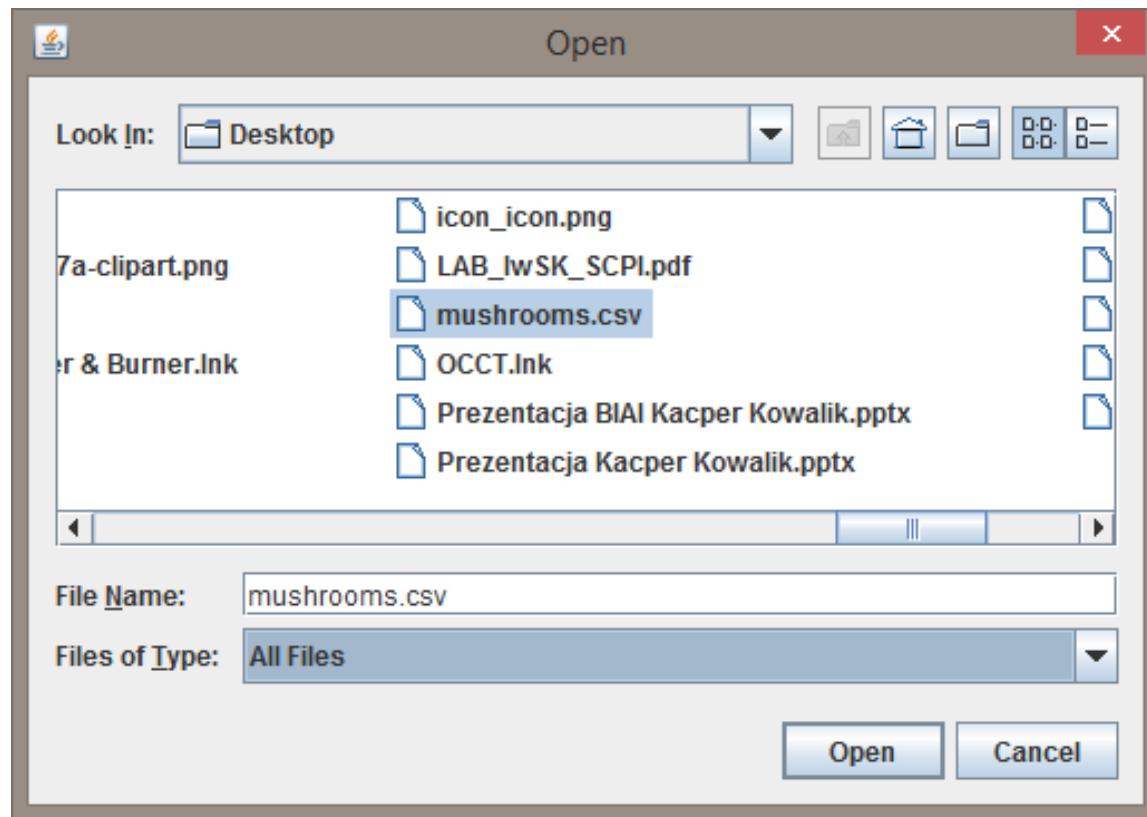
ARCHITEKTURA SYSTEMU

Okienkowa aplikacja klienta:



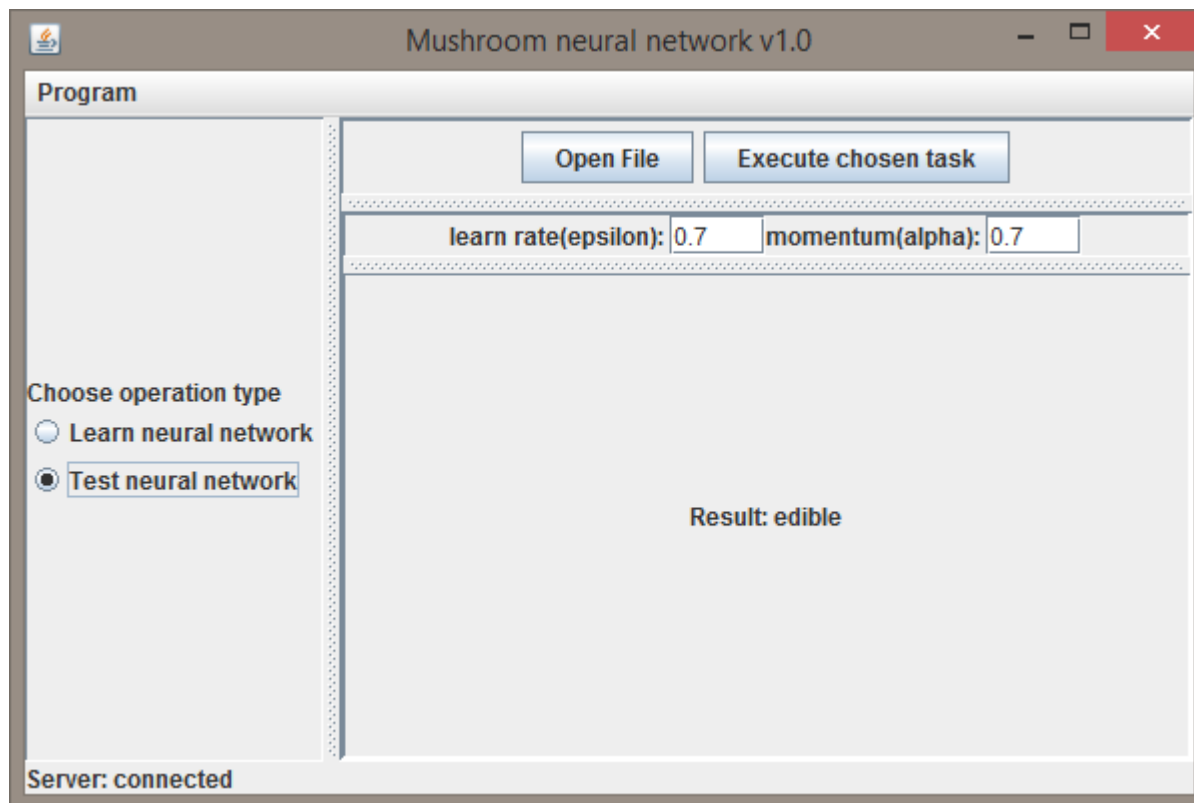
APLIKACJA KLIENTA

Po kliknięciu w przycisk open file należy wskazać plik z przygotowanym zbiorem rekordów do nauczania sieci neuronowej lub z tylko jednym rekordem do przetestowania sieci.



REZULTAT DZIAŁANIA

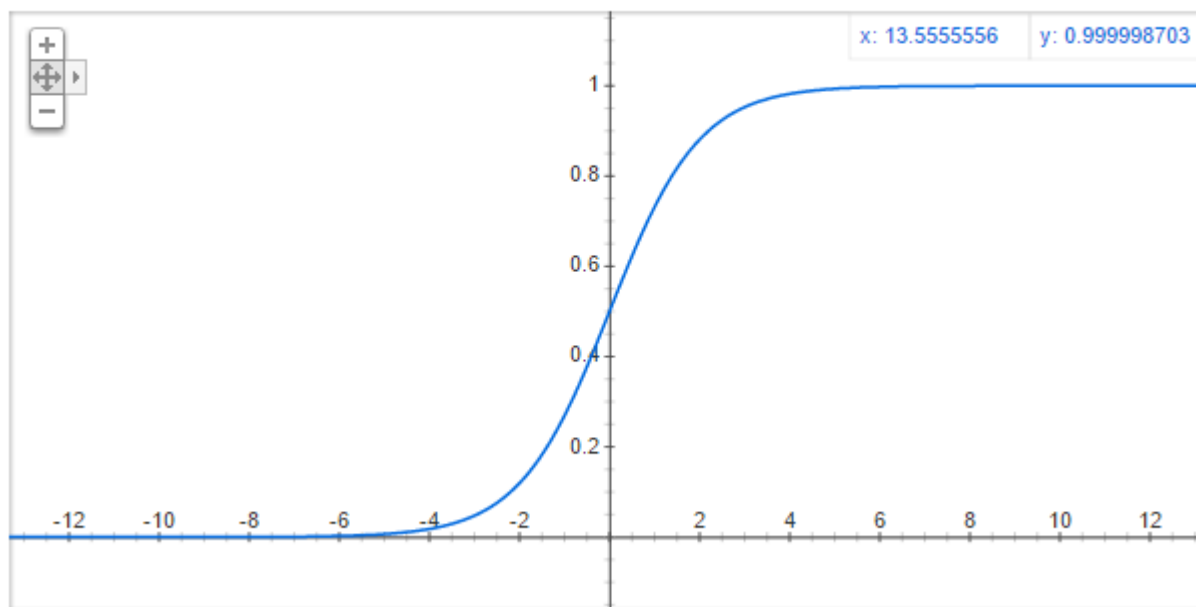
Jeśli sieć neuronowa zwróci informację o tym że dany grzyb jest jadalny lub nie, zostanie to odpowiednio zasygnalizowane poprzez ustawienie pola tekstowego przy etykiecie Result na „edible” lub „poisonous”.



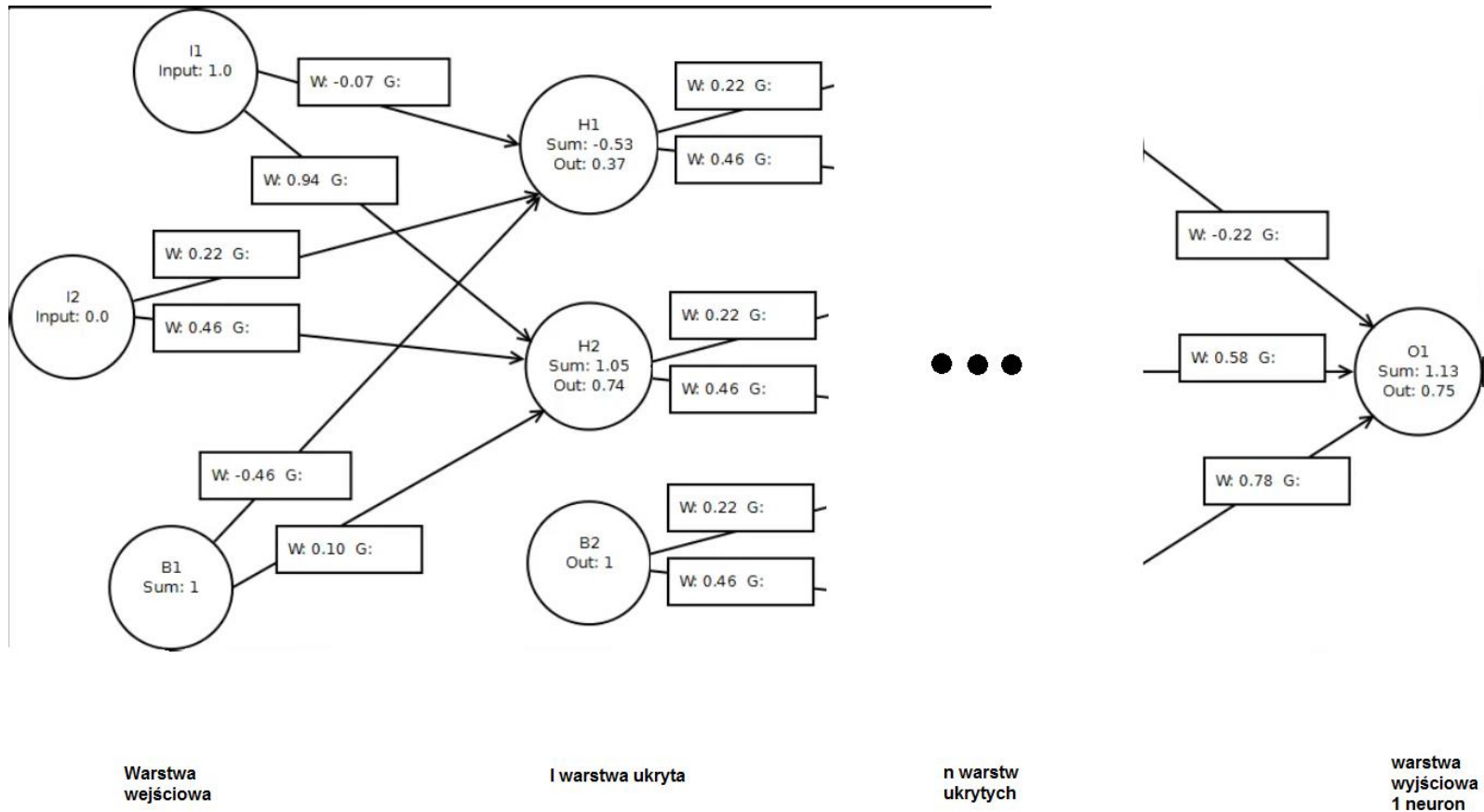
FUNKCJA AKTYWACJI

W programie jako funkcja aktywacji dla wyjść neuronów zostanie użyta funkcja sigmoidalna $f(x) = \frac{1}{1+e^{-x}}$

Wykres funkcji $1/(1+e^{-x})$



STRUKTURA SIECI NEURONOWEJ



PRZEBIEG BADAŃ NAD STRUKTURĄ SIECI NEURONOWEJ

W celu uzyskania jak najlepszych rezultatów sieć zostanie poddana szeregowi eksperymentów polegających na zmianach struktury sieci neuronowej. Zmiany te będą polegały na m. in.:

- Zmianach liczby warstw ukrytych.**
- Zmianach liczby neuronów w poszczególnych warstwach ukrytych.**
- Eksperymentowaniu z normalizacją wektora danych wejściowych.**
- Zmianie parametrów funkcji aktywacji.**
- Znalezieniu odpowiednich wartości dla parametrów learn rate oraz dla momentum.**

ŹRÓDŁA

<https://www.kaggle.com/uciml/mushroom-classification>

<http://zsi.ii.us.edu.pl/~nowak/ed/cw4.pdf>

<http://www.heatonresearch.com/jeff/>

KONIEC

Dziękuję za uwagę!