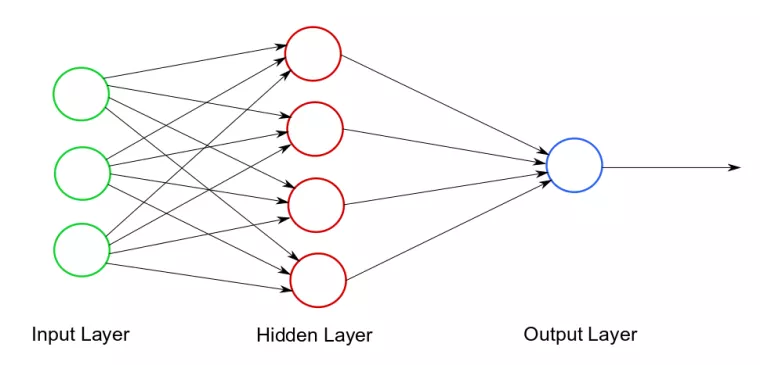
Sieć neuronowa to statystyczny model obliczeniowy stosowany w uczeniu maszynowym. Można o nim myśleć jak o [systemie](https://impicode.pl/oprogramowanie-dedykowane/) połączonych synapsami neuronów, które przesyłają między sobą impulsy (dane). Sieć neuronowa składa się z trzech warstw: **warstwy wejścia** (input layer), **warstwy ukrytej** (hidden layer), oraz **warstwy wyjścia** (output layer).



Warstwa wejścia przyjmuje dane wejściowe do obliczeń, w warstwie ukrytej odbywają się wszystkie obliczenia. Wynik tych obliczeń jest przesyłany do warstwy wyjścia.

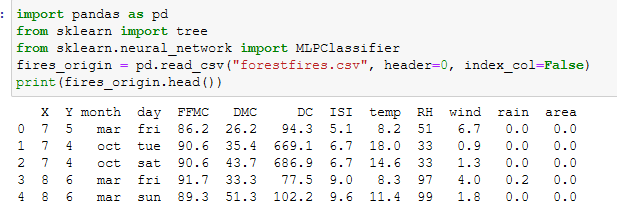
Na powyższym diagramie okręgi reprezentują neurony, zaś strzałki - synapsy. Każda synapsa ma przypisaną pewną wagę, tzn. liczbę, która (nieco upraszczając) określa, jak silnie przesyłana wartość wpływa na ostateczny wynik obliczeń. Żeby przesłać wartość, synapsa najpierw czyta wartość z neuronu wejściowego, następnie wartość tę mnoży przez wagę, by w końcu przesłać wynik do neuronu wyjściowego. Następnie neuron wyjściowy dokonuje obliczeń na dostarczonych mu przez synapsy wartościach i otrzymany wynik przekazuje do wychodzącej z niego synapsy.

Zad1. Proszę podzielić zbiór na dane trenujące i testujące, wytrenować i przetestować 5 sieci neuronowych o różnych architekturach. Proszę o sporządzenie sprawozdania z wnioskami.

Dane wykorzystane w zadaniach pochodzą z pliku forestfires.csv, który został pobrany ze

strony https://archive.ics.uci.edu/ml/index.php. Jest to zbiór informacji na temat pożarów

w parku Montensinho.



Docelowa baza danych:





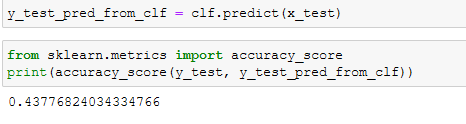
Powyższy fragment kodu przedstawia ostateczny układ zbioru danych który został wykorzystany do ćwiczeń.

Poniżej przedstawiamy wyniki testów wykonanych na 5 sieciach neuronowych o różnych architekturach.

Wyniki dla ilości neuronów = 2.

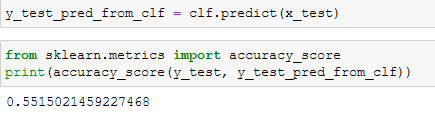


hidden\_layer\_sizes- parametr ten reprezentuje liczbę neuronów w warstwie ukrytej.  
random\_state- określa generowanie liczb losowych na potrzeby inicjalizacji wag i odchyleń.  
max\_iter- parametr ten określa liczbę epok, ile razy każdy punkt danych zostanie użyty.



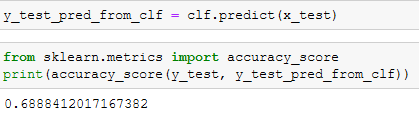
Wyniki dla ilości neuronów = 3.





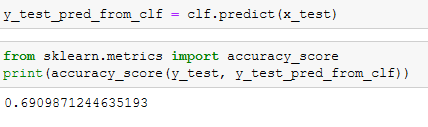
Wyniki dla ilości neuronów = 5.





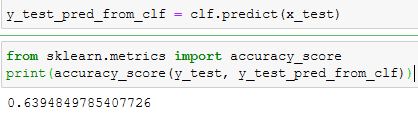
Wyniki dla ilości neuronów = 8.





Wyniki dla ilości neuronów = 12.





Wnioski:  
Z przedstawionych badań wynika, że zwiększenie ilości neuronów działa pozytywnie na końcowy wynik programu. Należy jednak pamiętać, by nie przesadzić z ilością neuronów w badaniu, ponieważ jakość programu ulegnie pogorszeniu.