

Przewidywanie hałasu płatu lotniczego na
podstawie obserwacji w tunelu
aerodynamicznym (Airfoil Self-Noise)

Grzegorz Landowski i Rafał Kurek

June 5, 2022

Contents

1	Co to jest sztuczna inteligencja?	3
2	Uczenie maszynowe	5
2.1	Sieci neuronowe	6
2.2	Uczenie głębokie	6
2.3	Metody uczenia maszynowego	6
3	Regresja liniowa	7
3.1	Wstęp do regresji liniowej	7
3.2	Zadanie regresji liniowej	8
3.3	Cel regresji liniowej	9
3.4	Dopasowywanie wzoru hipotezy	9
4	Repozytorium GitHub z projektem	11
5	Bibliografia	12

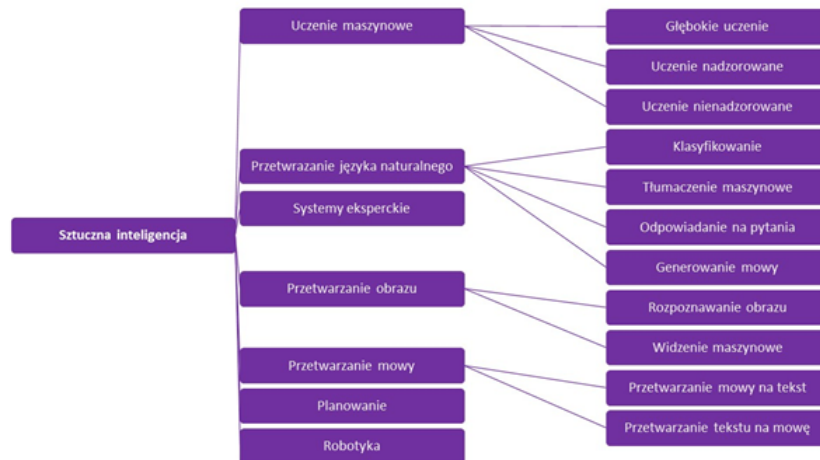
1 Co to jest sztuczna inteligencja?

Sztuczna inteligencja (ang. artificial intelligence - AI) jest to próba zasymulowania przez maszynę pewnego zachowania, które wymaga inteligencji.

Systemy AI mogą usprawniać swoje działania na podstawie zebranych informacji.

Głównym celem sztucznej inteligencji jest naśladowanie ludzkiego sposobu postrzegania rzeczywistości i reagowania na nią, a następnie wykroczenie poza właściwe im ograniczenia.

Figure 1: Schemat sztucznej inteligencji z raportu Game-changing technologies: Transforming production and employment in Europe



Definicja sztucznej inteligencji z dokumentów Komisji Europejskiej brzmi:

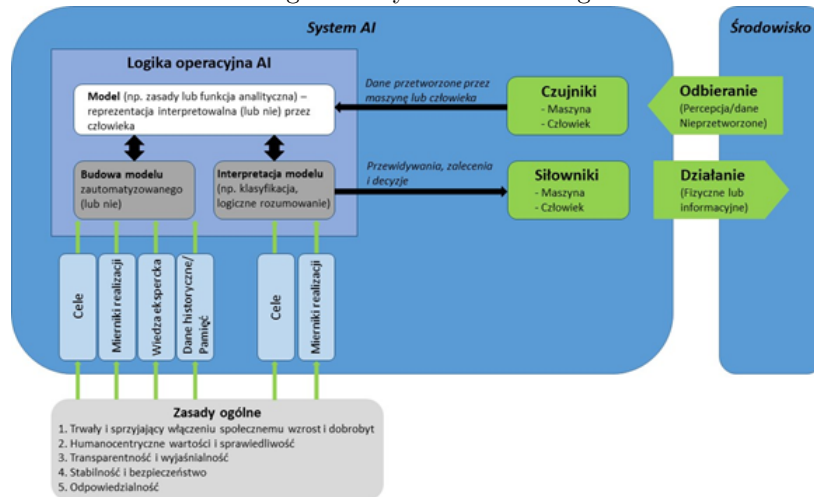
"Systemy (...) zaprojektowane przez ludzi systemy oprogramowania (i ewentualnie również sprzętu), które, biorąc pod uwagę złożony cel, działają w wymiarze fizycznym lub cyfrowym, postrzegając swoje środowisko poprzez pozyskiwanie danych, interpretując zebrane ustrukturyzowane lub nieustrukturyzowane dane, rozumując na podstawie wiedzy lub przetwarzając informacje, uzyskane z tych danych i decydując o najlepszym działaniu (działaniach), jakie należy podjąć, aby osiągnąć dany cel. ".

Organizacja Współpracy Gospodarczej i Rozwoju (OECD) także opracowała swoją definicję sztucznej inteligencji oraz jej schemat.

Według niej System AI to system oparty na koncepcji maszyny, która może wpływać na środowisko, formułując zalecenia, przewidywania lub decyzje dotyczące zadanego zestawu celów. Czyni to, wykorzystując dane wejściowe do:

- postrzegania rzeczywistych lub wirtualnych środowisk
- streszczania takiego postrzegania w modele ręcznie lub automatycznie
- wykorzystywania interpretacji modeli do formułowania opcji wyników

Figure 2: System AI według OECD



Schemat systemu AI według OECD składa się z:

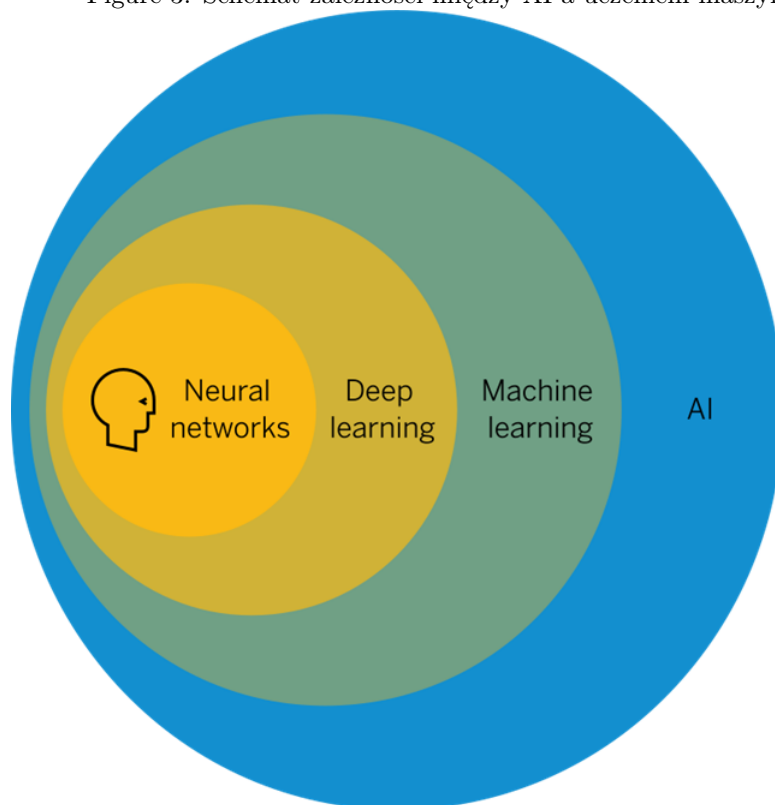
- czujników (sensorów)
- logiki operacyjnej (modeli algorytmów)
- siłowników (aparatu wykonawczego)

Czujniki zbierają nieprzetworzone dane ze środowiska, a siłowniki podejmują działania w celu zmiany stanu środowiska. Kluczowa siła systemu sztucznej inteligencji znajduje się w jego logice operacyjnej (modelach algorytmów), która dla danego zestawu celów i na podstawie danych wejściowych z czujników zapewnia ekstrakcje (wynik) dla siłowników – jako zalecenia, przewidywania lub decyzje, które mogą wpłynąć na stan środowiska.

2 Uczenie maszynowe

Uczenie maszynowe (ang. **machine learning**) to poddziedzina sztucznej inteligencji, które jest uogólnieniem algorytmów, które mają za zadanie w pewien sposób przetwarzać zbiór przykładów w celu realizacji określonego zadania. Podczas przetwarzania danych następują procesy uczenia i doskonalenia się, które nie są zaprogramowane. W uczeniu maszynowym algorytmy są trenowane pod kątem znajdowania wzorców i korelacji w dużych zbiorach danych oraz podejmowania najlepszych decyzji i formułowania prognoz na podstawie wyników takiej analizy.

Figure 3: Schemat zależności między AI a uczeniem maszynowym



Sztuczna inteligencja jest kategorią nadrzędną wobec wszystkich podzbiorów uczenia maszynowego. Pierwszy jej podzbiór stanowi uczenie maszynowe, kolejnym w jego obrębie jest głębokie uczenie, a wewnątrz niego znajdują się sieci neuronowe.

2.1 Sieci neuronowe

Sieci neuronowe to podzbiór uczenia głębokiego. Składają się z neuronów, nazywanymi „węzłami”, które są pogrupowane w liczne warstwy, które działają równolegle. Kiedy neuron otrzymuje sygnał, przetwarza go i przekazuje tę informację do pozostałych neuronów, które są z nim połączone. Wzmocnienie neuronów umożliwia lepsze rozpoznawanie wzorców i uczenie się.

W sieciach neuronowych wyróżnia się trzy typy warstw:

- **warstwa wejściowa** – jej jedyną funkcją jest pobieranie danych i przekazywanie ich do pierwszej warstwy ukrytej.
- **warstwa ukryta** – warstwa odpowiedzialna za naukę i wykonywanie obliczeń, nie można do niej w sposób bezpośredni przekazywać informacji.
- **warstwa wyjściowa** – warstwa obliczająca wartości wyjściowe całej sieci i zwracająca je na zewnątrz.

2.2 Uczenie głębokie

Uczenie głębokie (ang. deep learning) jest poddziedziną uczenia maszynowego. Podstawowym czynnikiem wyróżniającym uczenie głębokie od uczenia maszynowego jest to, że proces uczenia nie wymaga kontroli człowieka. Może bazować na nieustrukturyzowanych danych w nieprzetworzonej formie (np. teksty, obrazach) i automatycznie określać zestaw cech, które rozróżniają kategorie danych. Uczenie głębokie korzysta z sieci neuronowych uruchamiając wiele warstw sieci, uzyskując stopniowo wyniki na coraz wyższym poziomie dokładności.

2.3 Metody uczenia maszynowego

Dzieli się na:

- **uczenie nadzorowane** - do uczenia algorytmów wykorzystuje zbiory danych z etykietami, które umożliwiają dokładne klasyfikowanie danych lub przewidywanie wyników. Zbiór uczący składający się z uporządkowanej pary i -tego obiektu i jego etykiety

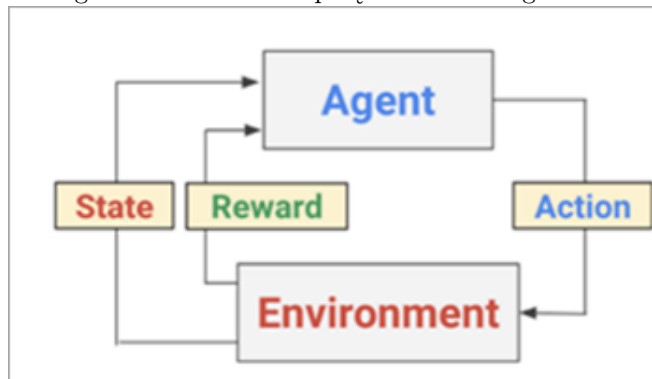
$$X = \{x^{(i)}, y^{(i)}\}, \text{ gdzie } i = 1 \dots m$$

- **uczenie nienadzorowane** - wykorzystuje algorytmy uczenia maszynowego do analizowania i grupowania zbiorów danych bez etykiet. Algorytmy te wykrywają ukryte wzorce lub zgrupowania danych bez konieczności interwencji człowieka. Zbiór uczący składa się z i różnych obiektów

$$X = \{x^{(i)}\}$$

- **uczenie ze wzmocnieniem** - przykład uczenia maszynowego, w którym agent poprzez interakcję z otoczeniem uczy się pewnej polityki postępowania. Dostaje cel, który ma osiągnąć, ale nie wskazują się mu drogi postępowania. Zawsze pożądane jest, aby każdy krok w algorytmie był podejmowany w celu osiągnięcia celu. W tego typu uczeniu algorytm uczy się poprzez mechanizm sprzężenia zwrotnego i doświadczenia z przeszłości. Za każdym razem, gdy ma zostać podjęty następny krok, otrzymuje informacje zwrotne z poprzedniego kroku, wraz z nauką z doświadczenia, aby przewidzieć, co może być następnym najlepszym krokiem. Droga prób i błędów dąży do celu.

Figure 4: mechanizm sprzężenia zwrotnego w uczeniu ze wzmocnieniem



3 Regresja liniowa

Tematyka projektu opiera się o problem regresji. Regresja jest przykładem uczenia nadzorowanego i wyróżniamy:

- regresję liniową
- regresję logistyczną

Do przewidzenia hałasu płatu lotniczego na podstawie obserwacji w tunelu aerodynamicznym korzystając ze zbioru danych "Airfoil Self-Noise" wykorzystaliśmy **regresję liniową** i to na jej opisie się skupimy.

3.1 Wstęp do regresji liniowej

Jak już zostało powiedziane regresja liniowa jest przykładem uczenia nadzorowanego.

W tym uczeniu mamy elementy uczące (cechy), które oznaczymy przez \mathbf{x} .

Z każdym elementem uczącym mamy związaną etykietę oznaczoną przez \mathbf{y} .

Przykładem nazywamy parę elementu uczącego oraz etykiety powiązanej z nią (x, y) .

Mamy dany zbiór uczący składający się z m przykładów oznaczony:

$$X = \{x^{(i)}, y^{(i)}\}, \text{ gdzie } i = 1 \dots m$$

Każdy element uczący oraz jego etykieta jest numerowana poprzez indeks górny i .

Element uczący $x^{(i)}$ możemy zakodować za pomocą wektora w n wymiarowej przestrzeni cech

$$x^{(i)} \in R^n$$

Dla przykładu możemy opisać obiekt człowieka za pomocą dwóch cech ($n=2$) wzrostu i wagi.

Dla i -tego elementu uczącego możemy mieć tylko jedną etykietę $y^{(i)}$ (wyjście) albo też kilka etykiet (wyjść). Na przykład przy rozpoznawaniu typu pojazdu oprócz rozpoznania, że dany obiekt jest samolotem chcielibyśmy także rozpoznać jego kolor. W takim wypadku moglibyśmy zapisać, że nasza etykieta $y^{(i)}$ należy do iloczynu kartezjańskiego liczby możliwych wyjść i możliwych różnych wartości tych wyjść:

$$y^{(i)} \in \{1 \dots z\} \times \{1 \dots \zeta\}$$

, gdzie z to liczba możliwych wyjść, a ζ to liczba możliwych parametrów każdego z tych wyjść.

W takim przypadku mówilibyśmy o uczeniu multiklasowym, wieloetykietowym, gdzie z oznacza klasę, a ζ etykietę.

W uczeniu nadzorowanym wyjście może być:

- **dyskretne** - tzn. może być przypisana klasa i wówczas problem nazywamy **klasyfikacją**.
- **wartością ciągłą** - wówczas problem nazywamy **regresją**, a wyjście może być opisane w Z wymiarach:

$$y^{(i)} \in R^Z$$

3.2 Zadanie regresji liniowej

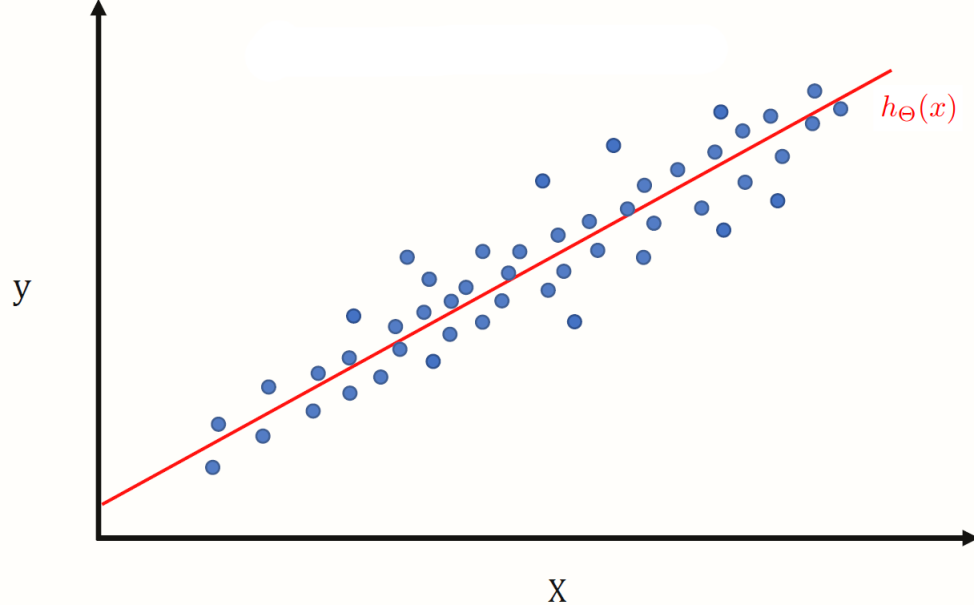
Założenia:

- $X \in \{x^{(i)}, y^{(i)}\}, i = 1 \dots m$
- $x^{(i)} \in R^n$ i -ty element uczący jest określony w n wymiarowej przestrzeni cech

- $y^{(i)} \in R$ i-te wyjście jest jednoelementową wartością, nieskalarną

Zadaniem regresji liniowej jest znalezienie takiej hipotezy (funkcji liniowej) $h_{\Theta}(x)$, która opisze nam możliwe jak najlepiej nasze dane uczące $\{x^{(i)}, y^{(i)}\}$.

Figure 5: Przykładowa hipoteza $h_{\Theta}(x)$ opisująca dane uczące



3.3 Cel regresji liniowej

Celem regresji liniowej jest takie wytrenowanie zbioru uczącego, by uzyskać hipotezę $h_{\Theta}(x)$, która jak najlepiej opisze dane uczące i za jej pomocą wstępować (przewidzieć) wartości nowych danych uczących spoza pierwotnego zbioru uczącego X .

3.4 Dopasowywanie wzoru hipotezy

Postać hipotezy, którą staramy się dopasować do danych uczących:

$$h_{\Theta}(x) = \Theta_0 + \Theta_1 X$$

Wyznaczamy losową hipotezę i sprawdzamy jej odległość od punktów uczących. Wartość punktu i-tego to wartość rzeczywista, a wartość hipotezy w $x^{(i)}$ to estymacja

Odległość i-tego punktu od wyznaczonej hipotezy nazywamy i-tym błędem i opisujemy danym wzorem:

$$e^{(i)} = (h_{\Theta}(x^{(i)}) - y^{(i)})^2$$

Chcemy określić hipotezę za pomocą funkcji błędu danej wzorem:

$$J(\Theta) = \sum_i e^{(i)}$$

Parametry θ_i (zwane także wagami) parametryzują przestrzeń funkcji liniowych $X \rightarrow Y$. To tych parametrów właśnie szukamy.

Zapiszmy je w postaci wektorowej: $\theta = \begin{bmatrix} \Theta_0 \\ \Theta_1 \end{bmatrix}$

Następnie możemy zapisać naszą hipotezę regresji liniowej w postaci wektorowej:

$$h_{\Theta}(x) = \Theta^T x$$

Teraz należy znaleźć minimum funkcji błędu względem parametru Θ :

$$\min_{\Theta} J(\Theta) = \min_{\Theta} \frac{1}{2} \sum_i e^{(i)} = \min_{\Theta} \frac{1}{2} \sum_{i=1}^m (h_{\Theta}(x^{(i)}) - y^{(i)})^2$$

, pomnożenie sumy liczbą $\frac{1}{2}$ ułatwi dalsze obliczenia. Nie zmienia to współrzędnych, zmieniają się tylko wartości bezwzględne błędów.

Chcemy znaleźć takie parametry, aby zminimalizować funkcję błędu. Zobaczmy czy zadziała następujący pomysł:

Zacznijmy od pewniej "odgadniętej" wartości początkowej. Następnie zmieniamy ją zgodnie z kierunkiem przeciwnym do gradientu funkcji błędu.

Gradient funkcji to wektor, którego kierunek pokrywa się z kierunkiem, w którym funkcja zmienia się najszybciej, a zwrot wskazuje kierunek, w którym funkcja rośnie. Formalnie jeden krok algorytmu minimalizacji gradientowej możemy zapisać:

$$\Theta_j \leftarrow \Theta_j - \eta \frac{\partial}{\partial \Theta_j} J(\Theta)$$

, gdzie parametr η to współczynnik, który wpływa na wielkość kroków algorytmu.

Obliczamy pochodną $\frac{\partial}{\partial \Theta_i} J(\Theta)$:

$$\begin{aligned} \frac{\partial}{\partial \Theta_i} J(\Theta) &= \frac{\partial}{\partial \Theta_i} \frac{1}{2} \sum_{i=1}^m (h_{\Theta}(x^{(i)}) - y^{(i)})^2 = \frac{1}{2} \sum_{i=1}^m \frac{\partial}{\partial \Theta_i} (h_{\Theta}(x^{(i)}) - y^{(i)})^2 = \\ &= \frac{1}{2} \sum_{i=1}^m 2(h_{\Theta}(x^{(i)}) - y^{(i)}) \frac{\partial}{\partial \Theta_i} h_{\Theta}(x^{(i)}) = \sum_{i=1}^m (h_{\Theta}(x^{(i)}) - y^{(i)}) \frac{\partial}{\partial \Theta_j} \sum_{j=0}^n 0_j x_j^{(i)} = \\ &= \sum_{i=1}^m (h_{\Theta}(x^{(i)}) - y^{(i)}) \sum_{j=0}^n \frac{\partial}{\partial \Theta_j} 0_j x_j^{(i)} = \sum_{i=1}^m (h_{\Theta}(x^{(i)}) - y^{(i)}) x_j^{(i)} \end{aligned}$$

Posiadając wynik pochodnej można ropisać kroki postępowania:

- Zainicjuj Θ_j
- powtarzaj, aż zbiegniesz dla każdego j : $\Theta_j := \Theta_j - \eta \sum_{i=1}^m (h_{\Theta}(x^{(i)}) - y^{(i)}) x_j^{(i)}$

4 Repozytorium GitHub z projektem

Projekt w formacie Jupyter Notebook jest dostępny pod następującym linkiem <https://github.com/grzegorzlandowski/Airfoil-Self-Noise-regresja-liniowa.git>

W repozytorium można też znaleźć to sprawozdanie w formacie PDF oraz .tex (plik .tex spakowany w archiwum .zip)

5 Bibliografia

1. <https://www.oracle.com/pl/artificial-intelligence/what-is-ai/>
2. <https://www.gov.pl/web/ai/czym-jest-sztuczna-inteligencja2>
3. <https://www.sap.com/poland/insights/what-is-machine-learning.html>
4. <https://www.ibm.com/pl-pl/cloud/learn/machine-learning>
5. <https://bulldogjob.pl/readme/czym-jest-deep-learning-i-sieci-neuronowe>
6. <https://myservername.com/types-machine-learning>