POLITECHNIKA GDAŃSKA

AI TECH

Wprowadzenie do Sztucznej Inteligencji

podstawy technik uczenia maszynowego

Adam Kurowski
Katedra Systemów Multimedialnych,
Wydział Elektroniki, Telekomunikacji i Informatyki PG

Fundusze Europejskie Polska Cyfrowa
Rzeczpospolita Polska
Unia Europejska Europejski Fundusz Rozwoju Regionalnego

Projekt współfinansowany ze środków Unii Europejskiej w ramach Europejskiego Funduszu Rozwoju Regionalnego
Program Operacyjny Polska Cyfrowa na lata 2014-2020.
Oś priorytetowa nr 3 „Cyfrowe kompetencje społeczeństwa”, działanie nr 3.2 „Innowacyjne rozwiązania na rzecz aktywizacji cyfrowej”.
Tytuł projektu: „Akademia Innowacyjnych Zastosowań Technologii Cyfrowych (AI Tech)”.

Wprowadzenie

Termin **„sztuczna inteligencja” jest niezwykle szeroki** i związany jest z bardzo różnymi zagadnieniami z obszarów wielu różnorodnych dziedzin nauki. Ich natura jest zarówno techniczna, jak i filozoficzna.

Ogólnym celem sztucznej inteligencji jest wytworzenie algorytmu lub maszyny zdolnej do rozumowania w sposób taki sam lub lepszy niż ludzki.

Jest to bardzo niejasno zdefiniowany, ogólny i trudny do zrealizowania cel. Zwłaszcza jeżeli algorytm implementujący sztuczną inteligencję ma realizować tak zróżnicowany szereg zadań, jak człowiek.

Przykładem takiego zróżnicowania jest na przykład jednoczesna umiejętność rozwiązywania skomplikowanych równań matematycznych i umiejętność namalowania na płótnie inspirującego obrazu, który dodatkowo nawiązuje do wydarzeń historycznych.

Wprowadzenie

Ze względu na **niejasną definicję tego czym jest sztuczna inteligencja** i trudnością z otrzymaniem tak zwanej ogólnej sztucznej inteligencji (ang. *general artificial intelligence*) w praktyce konieczny jest **podział i ustandaryzowanie zadań możliwych do rozwiązania za pomocą algorytmów inteligentnych.**

Są to **np. zadania klasyfikacji, przybliżania funkcji (tzw. regresji), albo interakcji z otoczeniem** (uczenie przez wzmacnianie, ang. *reinforcement learning*).

Bardzo wielu **narzędzi** do rozwiązywania tego typu dobrze formalnie zdefiniowanych problemów dostarcza **dziedzina techniki nazywana uczeniem maszynowym** (ang. *machine learning*).

Podstawowe definicje

sztuczna inteligencja

szerokie pojęcie oznaczające wszelkie techniki mające na celu naśladowanie ludzkiej inteligencji w bardzo szerokim sensie (np. do tak abstrakcyjnych zadań, jak wytworzenie tzw. sztucznej świadomości)

uczenie maszynowe

zestaw technik bazujących między innymi na statystyce matematycznej i teorii optymalizacji w celu realizacji zadań takich jak klasyfikacja, czy regresja (np. drzewa decyzyjne, SVM, sieci neuronowe, maszyny Boltzmanna, itd.)

uczenie głębokie

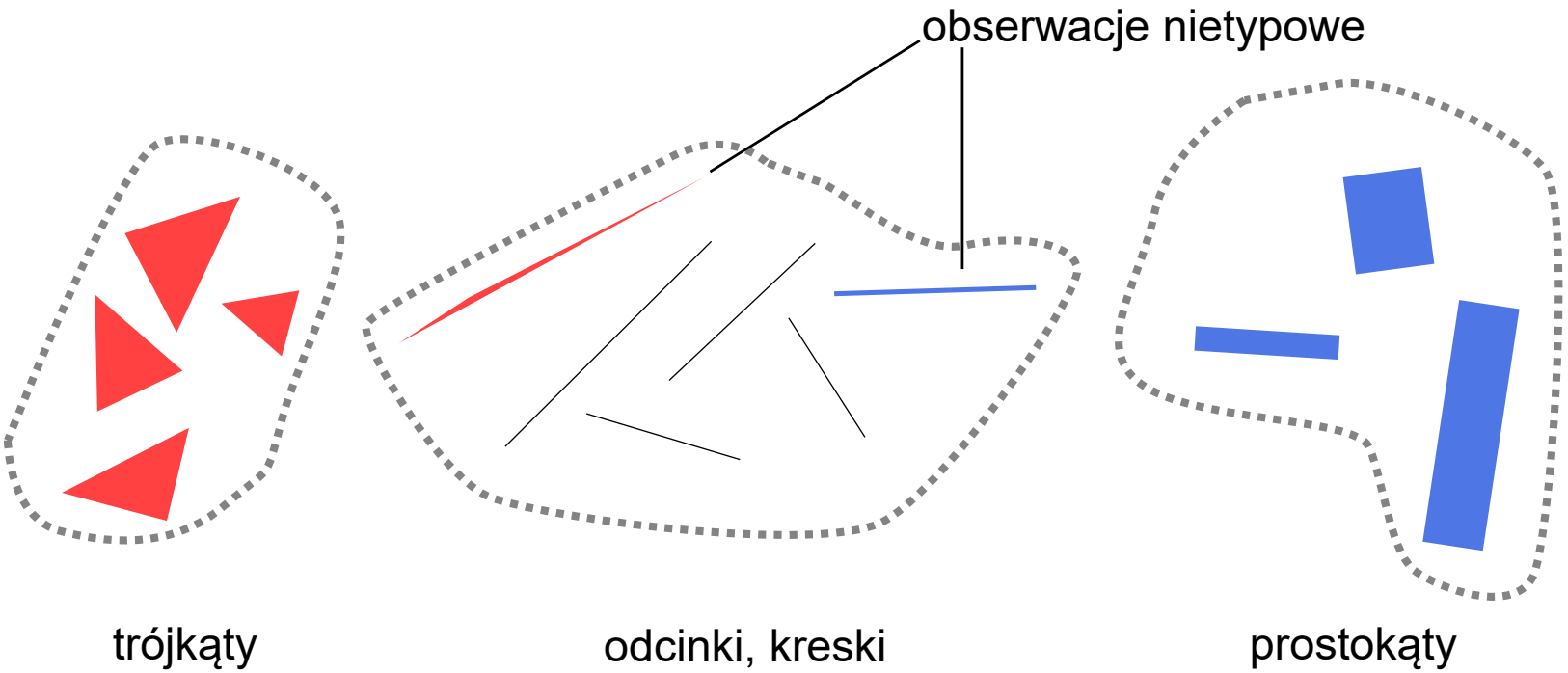
zestaw technik skupiających się na rozwiązywaniu problemów uczenia maszynowego za pomocą algorytmów bazujących na sztucznych sieciach neuronowych, składających się ze znacznej liczby warstw

Typowe problemy w uczeniu maszynowym

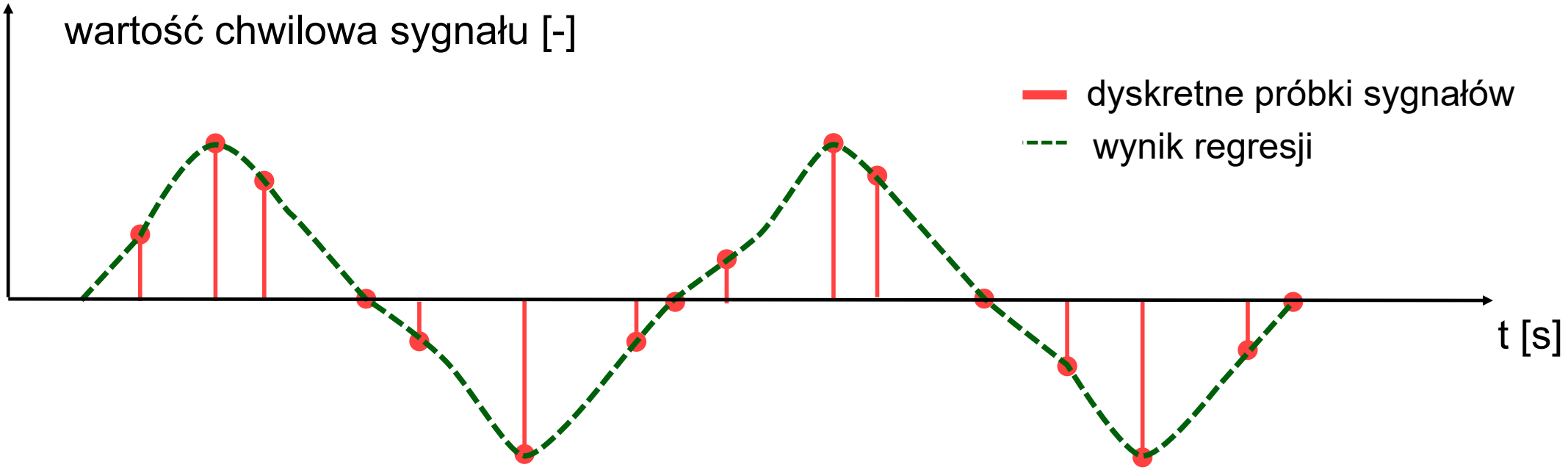
Bardzo często dokonujemy podziału problemów możliwych do rozwiązania za pomocą technik uczenia maszynowego na wąskie przypadki. Przykładami klasycznych problemów tego typu są:

- **klasyfikacja** – przydział wzorców wprowadzanych na wejście algorytmu uczenia maszynowego do jednej z kilku możliwych klas,
- **regresja** – aproksymacja przez model uczenia maszynowego wartości funkcji na przykładzie danych w postaci par argumentów i wartości funkcji, która ma być przez algorytm przybliżona,
- **uczenie przez wzmacnianie** – odnajdywanie optymalnego zachowania się inteligentnego algorytmu (zwanego agentem) poprzez ciągłą interakcję z otoczeniem (symulacją lub światem rzeczywistym np. poprzez sterowanie robotem) i utrwalanie najbardziej korzystnych zachowań algorytmu (agenta).

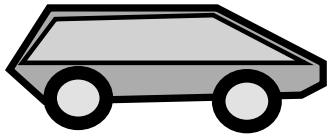
Klasyfikacja



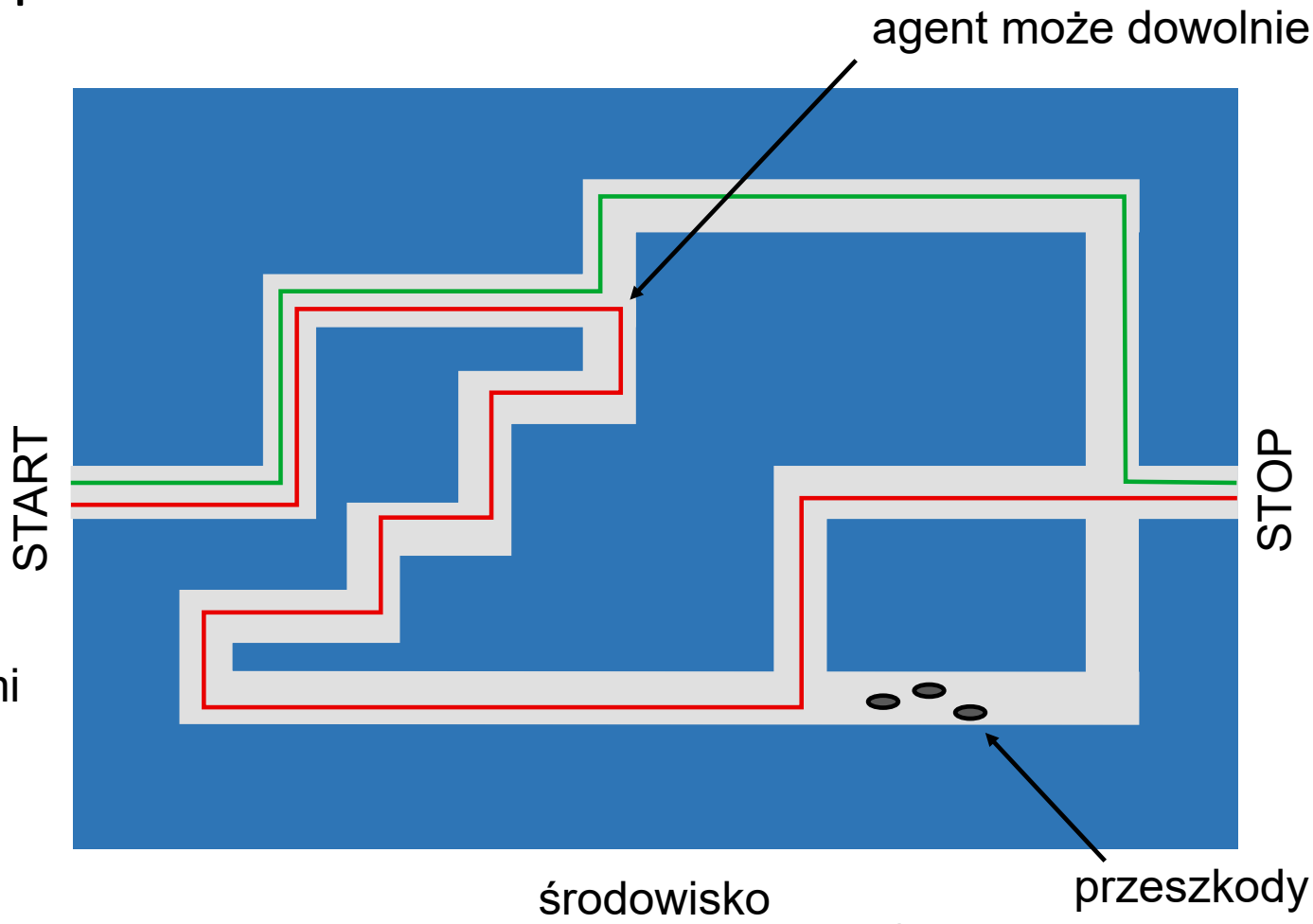
Regresja


wartość chwilowa sygnału [-]
dyskretne próbki sygnałów
wynik regresji
t [s]

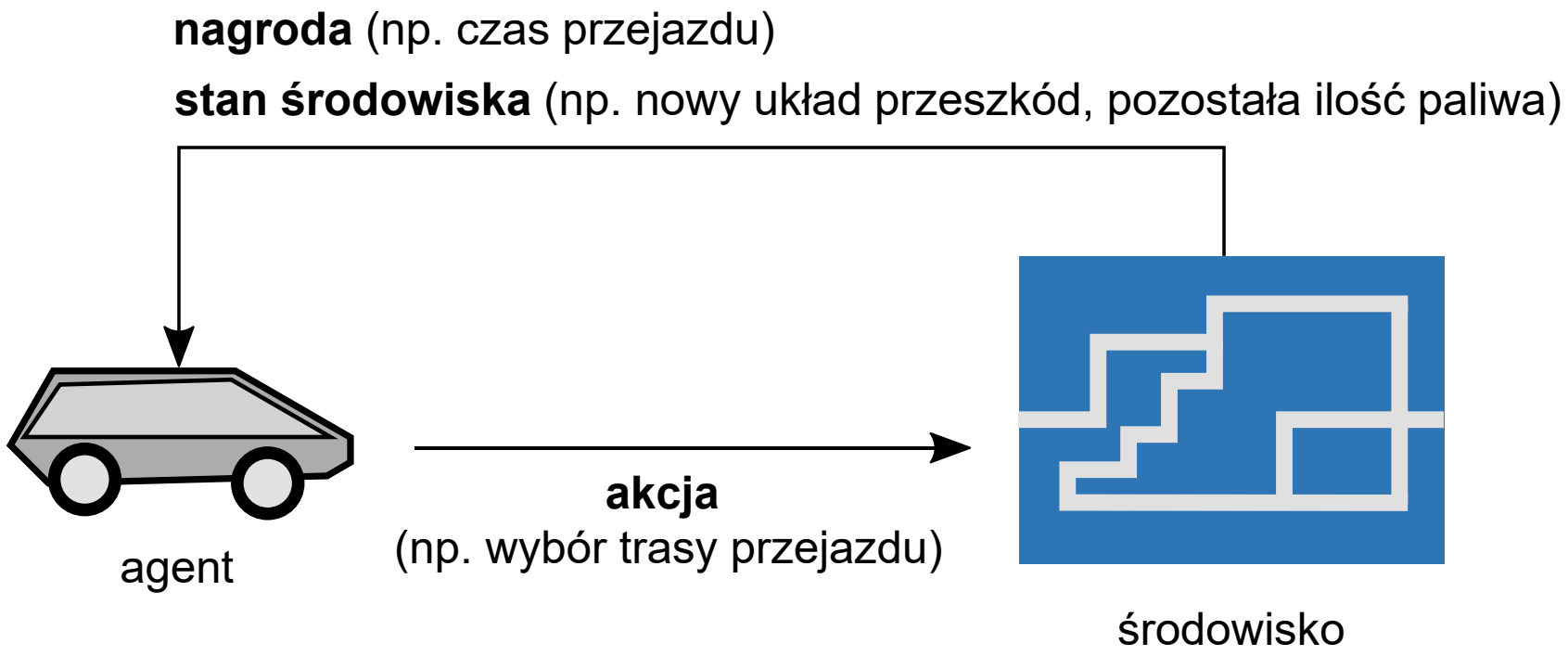
Uczenie przez wzmacnianie



agent
(może dysponować ograniczonymi zasobami np. paliwem)



Uczenie przez wzmacnianie



Typowe problemy w uczeniu maszynowym

Oczywiście tego typu ujęcie problemów rozwiązywanych przez uczenie maszynowe jest bardzo ogólne. Stąd warto zastanowić się nad tym, jak mogą być one wykorzystywane do rozwiązywania rzeczywistych problemów. Na przykład:

- **klasyfikacja** może zostać wykorzystana do analizy tego jakie typy pojazdów (samochody, ciężarówki, motocykle, itp.) poruszają się daną trasą. Na tej podstawie można np. śledzić ruch pojazdów ciężarowych i odpowiednio sterować ruchem, aby zapobiegać degradacji nawierzchni,
- **regresja** może stanowić element algorytmu szacującego możliwe do uzyskania plony wybranego rodzaju rośliny na podstawie informacji o stężeniach substancji chemicznych w glebie, na której uprawa ma się odbywać,
- **uczenie przez wzmacnianie** może posłużyć do wytrenowania (np. w symulacji) agenta kierującego pojazdem autonomicznym (po testach – w warunkach rzeczywistych).

Typowe problemy w uczeniu maszynowym

Przytoczone problemy klasyfikacji, regresji i uczenia przez wzmacniane nie są jedynymi problemami możliwymi do postawienia w uczeniu maszynowym.

Przykładem innego, nieco bardziej skomplikowanego problemu jest **reprezentacja** złożonych obiektów (opisywanych przez wiele zmiennych) za pomocą krótszych wektorów liczb (tzw. reprezentacji). Dodatkowym wymogiem w takim przypadku może być na przykład możliwość zrekonstruowania oryginalnego obiektu na podstawie jego reprezentacji. **W takim przypadku mamy do czynienia z przetwarzaniem typu koder-dekoder, jednym z prostszych algorytmów bazującym na takim przetwarzaniu jest sieć neuronowa typu autoenkoder.**

Algorytm tego typu może posłużyć do kompresji danych oraz jako część większego systemu w którym na przykład **słowa języka naturalnego opisywane są jako punkty w wielowymiarowej przestrzeni.**

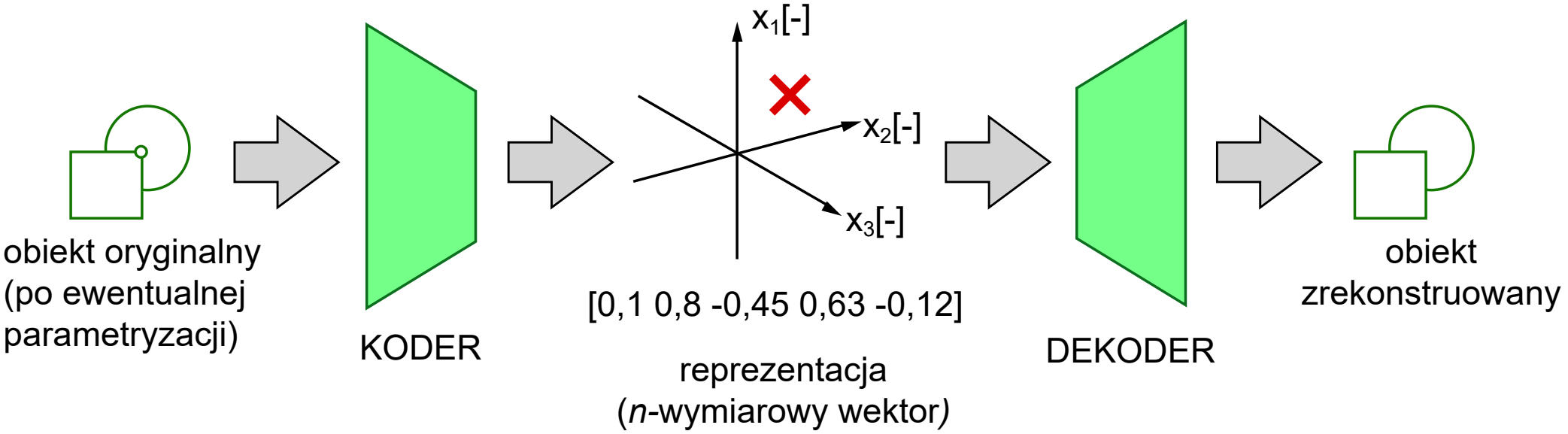
Typowe problemy w uczeniu maszynowym

Jeżeli algorytm uwzględnia dodatkowo, na przykład **grupy znaczeniowe wyrazów do układania punktów w przestrzeni decyzyjnej** według znaczenia odpowiadających im wyrazów, to jest to **przykład algorytmu uczącego się metryk odległości (znaczeniowej) między wyrazami (ang. *distance metric learning*).**

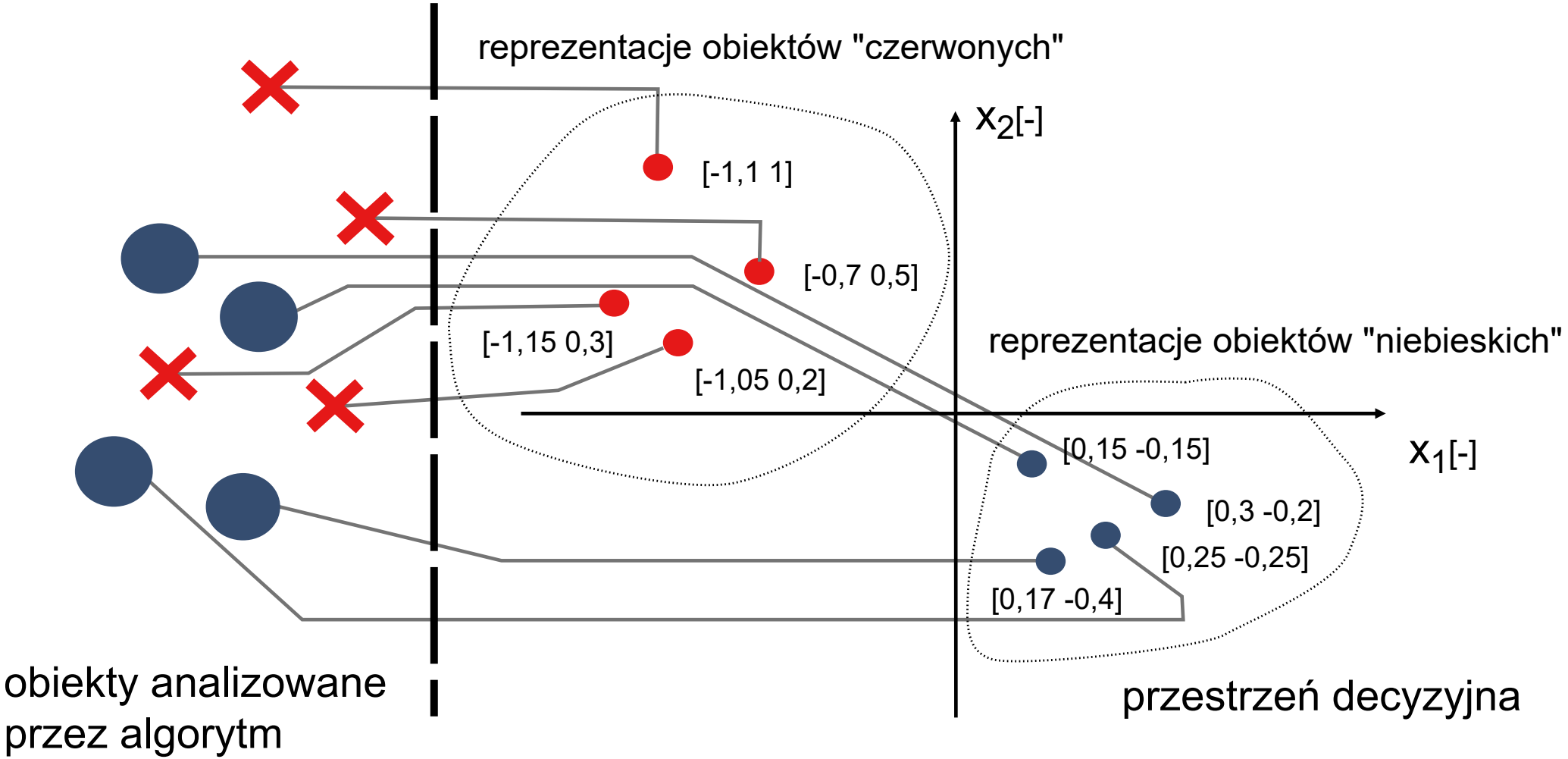
Wygenerowana przez model **metryka podobieństwa** może potem na przykład posłużyć **do wyszukiwania synonimów**.

Na podobnej zasadzie algorytm może na przykład uczyć się **wyszukiwania obrazów wizualnie podobnych** do wskazanych przykładów.

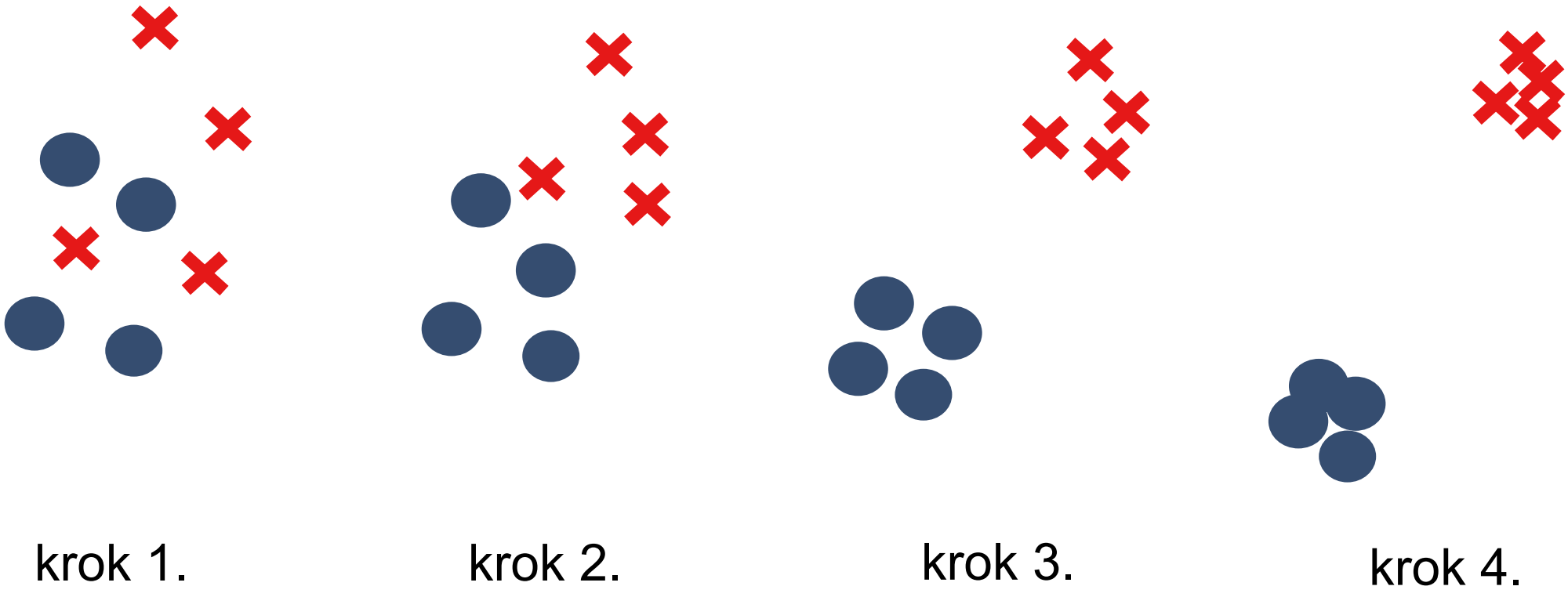
Model koder-dekoder (na przykładzie autoenkodera)



Analiza reprezentacji



Uczenie metryk odległości



krok 1. krok 2. krok 3. krok 4.

Przygotowanie danych dla algorytmów uczenia maszynowego

Pierwszym i niezwykle ważnym krokiem w przygotowywaniu rozwiązania bazującego na uczeniu maszynowym jest zebranie zbioru danych. Na tym etapie można popełnić kilka błędów, które sprawią, że zebrane dane nie będą odwzorowywać rzeczywistości.

Przy pozyskiwaniu danych (np. zbieraniu zdjęć przedstawiających obiekty różnych klas dla systemu klasyfikacyjnego) konieczne jest **zidentyfikowanie możliwie jak największej liczby możliwych źródeł zakłóceń i ich eliminacja** – na przykład należy upewnić się, że posiadamy sensor umożliwiający zarejestrowanie danych z zadowalającą jakością.

Przygotowanie danych dla algorytmów uczenia maszynowego

Ważne jest także upewnienie się, że jeżeli jakieś zakłócenia są niemożliwe do wyeliminowania, to będziemy posiadać **na tyle dużo przykładów ich występowania**, że systemu nauczy się **uwzględniać ich występowanie** w danych wejściowych.

Jeżeli zbierane są dane dla systemu klasyfikacji, należy starać się, aby liczby przykładów dla każdej klasy były możliwie zbliżone. W innym przypadku konieczne jest stosowanie specjalnych technik umożliwiających uczenie maszynowe na danych o niezrównoważonej liczebności klas (np. oversampling, undersampling, SMOTE).

Podobnie należy dbać o to, by **w przypadku regresji przykłady pokrywały równomiernie cały zakres wartości argumentów, dla których ma być możliwa regresja.**

Przygotowanie danych dla algorytmów uczenia maszynowego

Tak pozyskane dane czasami należy poddać procesowi parametryzacji, aby zmniejszyć ilość danych przekazywanych na wejście algorytmu uczenia maszynowego. Może to być na przykład obliczenie spektrogramu sygnału uzyskanego z mikrofonu i posługiwanie się nim zamiast surowych próbek sygnału akustycznego.

Dane często mogą wymagać **wstępnej eliminacji wartości odstających** (np. za pomocą reguły trzech sigm).

Często algorytm nie będzie uczyć się na podstawie danych które nie były poddane **normalizacji** albo **standaryzacji**.

Przygotowanie danych dla algorytmów uczenia maszynowego

Dane poddane procesowi parametryzacji i normalizacji lub standaryzacji należy na potrzeby procesu uczenia podzielić na **trzy zbiory**:

- **zbiór treningowy**, który będzie wykorzystany przez algorytm do nauczenia się zależności, które w nim występują,
- **zbiór walidacyjny**, na którym w trakcie treningu **kontrolowana jest jakość** działania algorytmu – są to dane, które **nie są wykorzystywane w treningu** i nie są „znane” algorytmowi,
- **zbiór testowy**, na którym testowana ostatecznie jest skuteczność wytrenowanego algorytmu. Jest on niewykorzystany aż do samego końca procesu treningu, gdyż autor algorytmu na podstawie postępów mierzonych na zbiorze walidacyjnym często dokonuje zmian w strukturze algorytmu. **Zastosowanie osobnego zbioru daje pewność, że dobre wyniki algorytmu wynikają faktycznie z jakości treningu, a nie specyficznego doboru jego struktury pod konkretny zbiór danych.**

Sztuczne sieci neuronowe

Sztuczne sieci neuronowe są jednym z algorytmów uczenia maszynowego, które w **luźny sposób zainspirowane są naukami biologicznymi**.

Zasada działania sieci neuronowych polega na wykorzystaniu **wielu podobnych do siebie elementów** (sztucznych neuronów), które połączone są w sieć. Każdy neuron może przyjmować wyniki obliczeń neuronów poprzednich i przekazywać własny wynik obliczeń do neuronów występujących w sieci po nim.

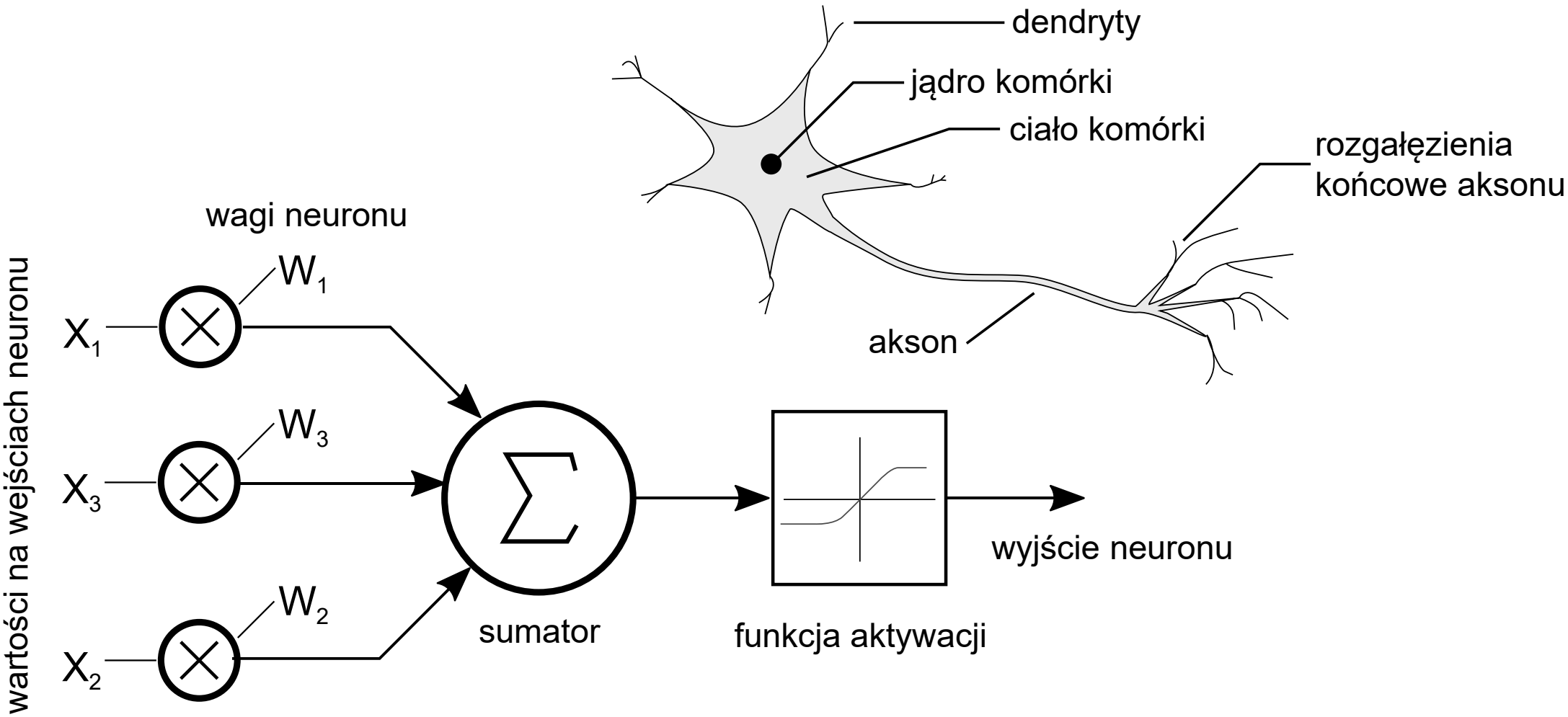
Taki algorytm stanowi **uniwersalny aproksymator**, który w połączeniu z odpowiednim algorytmem treningu pozwala na aproksymację nawet bardzo skomplikowanych funkcji matematycznych przyjmujących **wiele parametrów wejściowych i także zwracających wiele wartości wyjściowych**.

Sztuczne sieci neuronowe

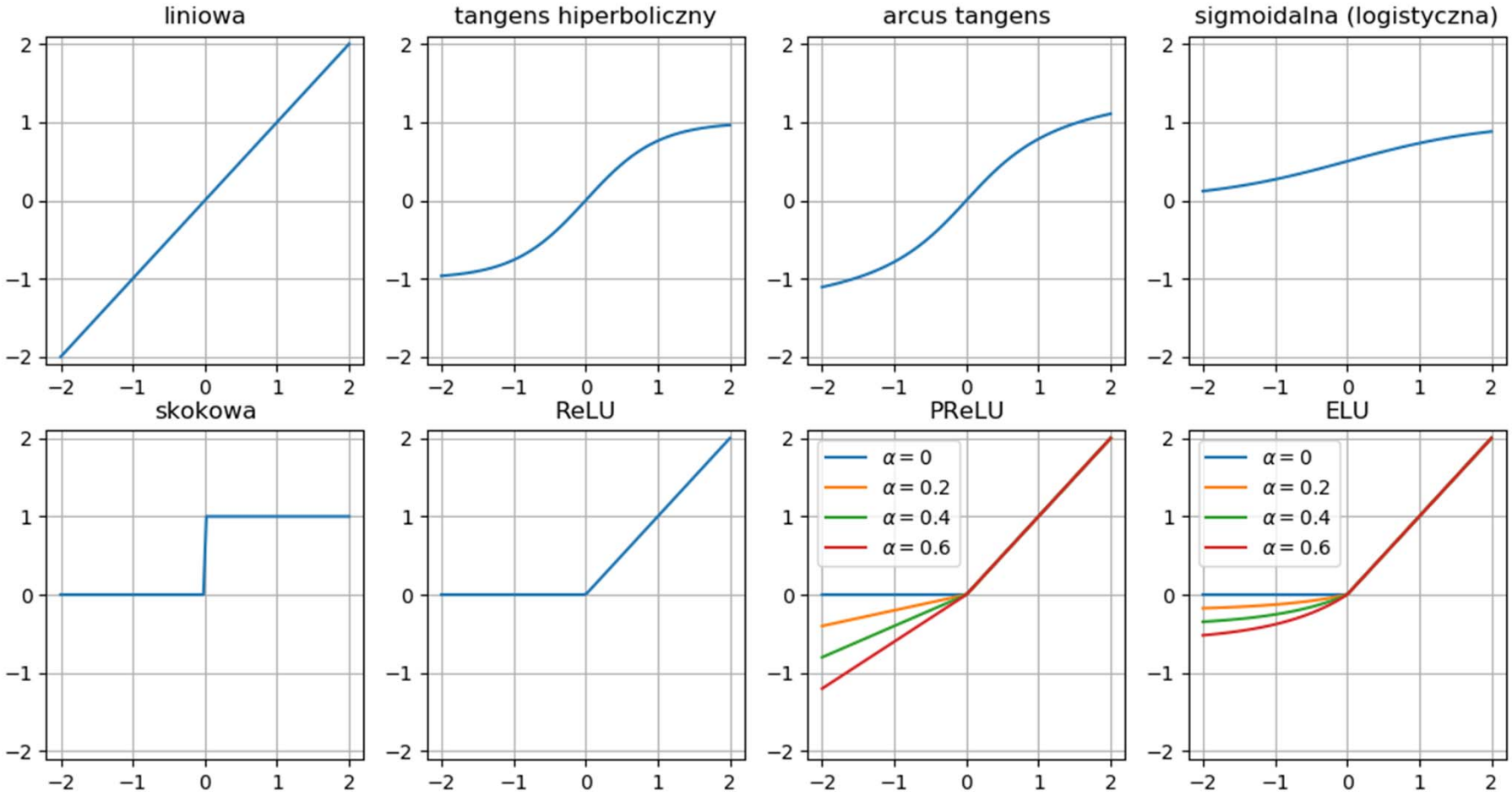
Typowym algorytmem treningu sztucznych sieci neuronowych jest **algorytm wstecznej propagacji błędów**, który bazuje na analizie funkcji błędu popełnianego przez sieć. **Wyznacza on gradient funkcji błędu**, którego analiza pozwala na **wyliczenie takich modyfikacji sieci, które zmniejszą popełniany przez nią błąd.**

Wielkość poprawek kontrolowana jest za pomocą parametru nazywanego **współczynnikiem szybkości nauki.** Ze względu na fakt, że algorytm bazuje na obliczaniu gradientu – jest on podatny na problemy takie jak **lokalne minima funkcji błędu i przeuczenie (ang. *overfitting*).** Jest to przyczyna dla której w trakcie treningu wykorzystujemy dodatkowy zbiór walidacyjny w celu kontroli zachowania się algorytmu, gdy otrzyma on dane inne niż te, które otrzymał on w trakcie treningu.

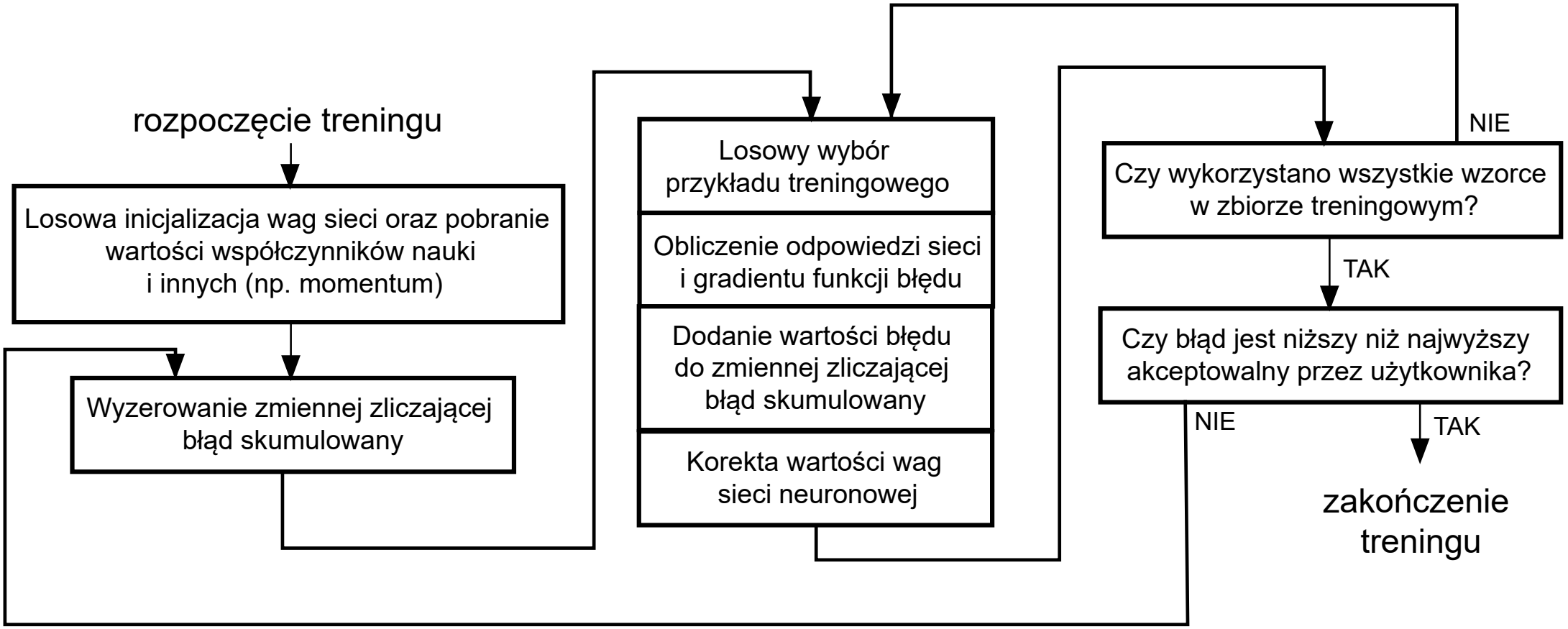
Neuron biologiczny i sztuczny



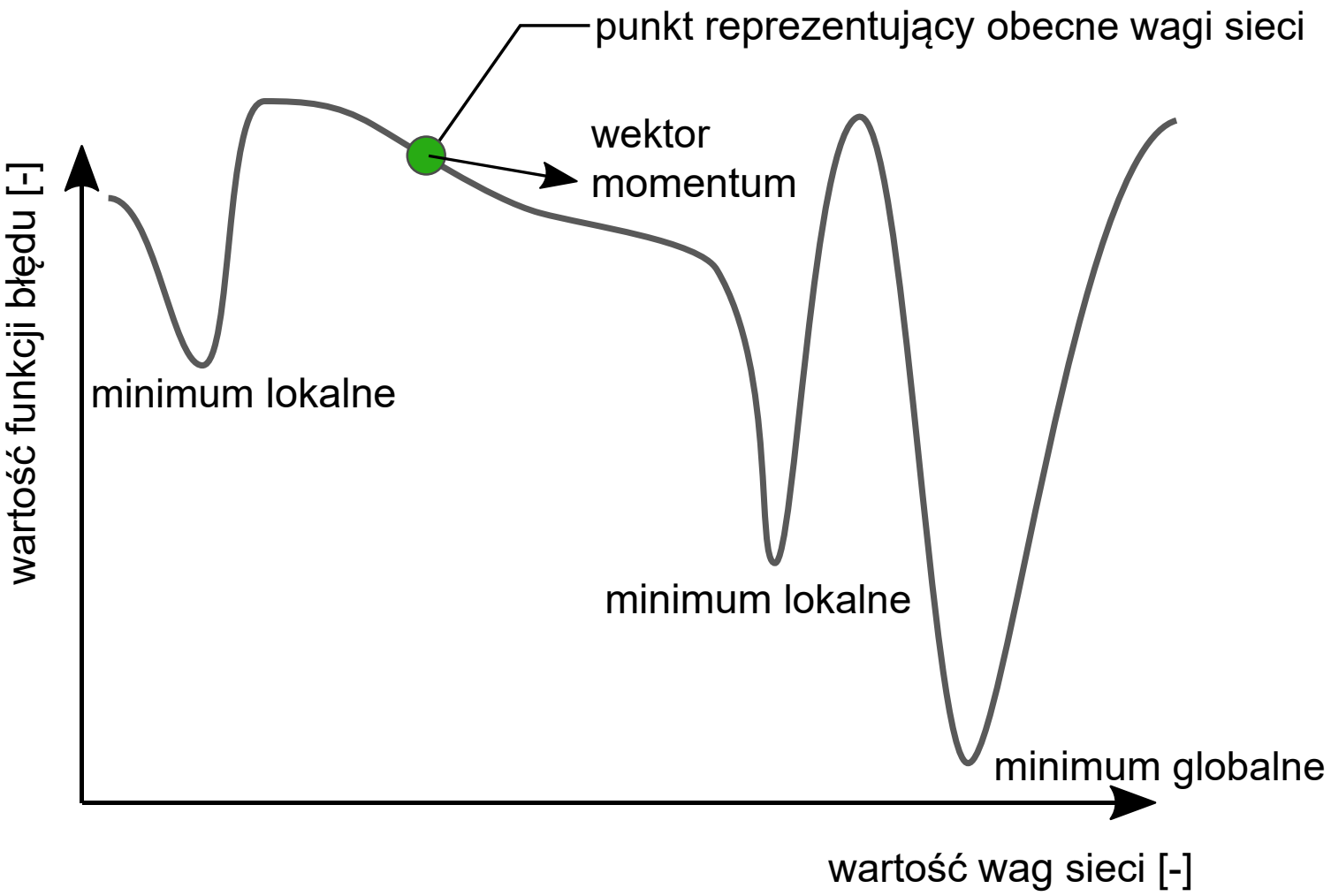
Funkcje aktywacji



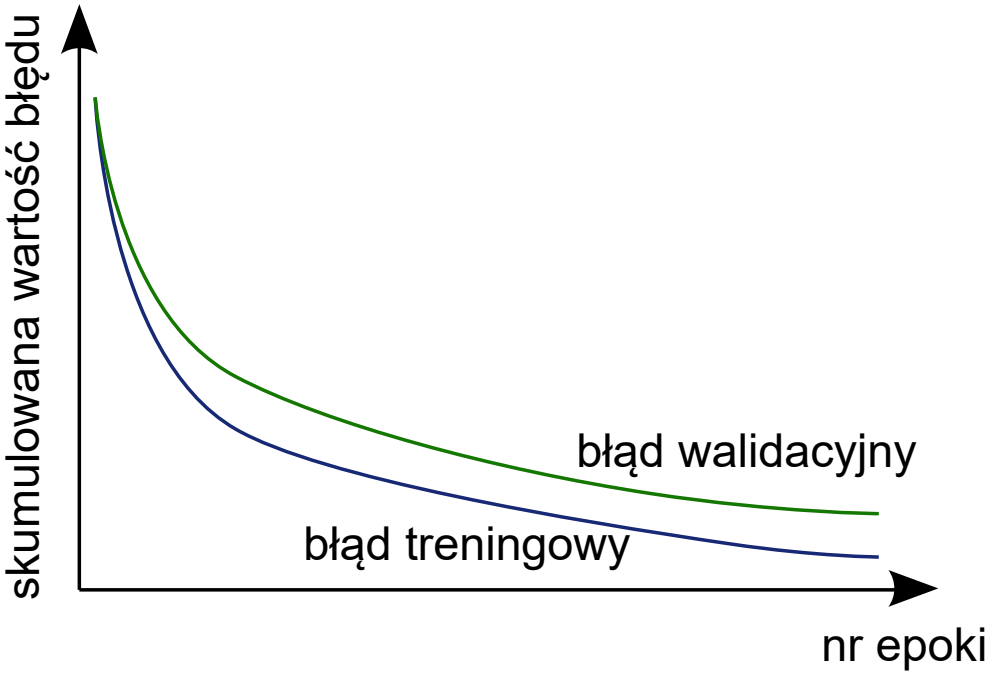
Algorytm wstecznej propagacji błędów



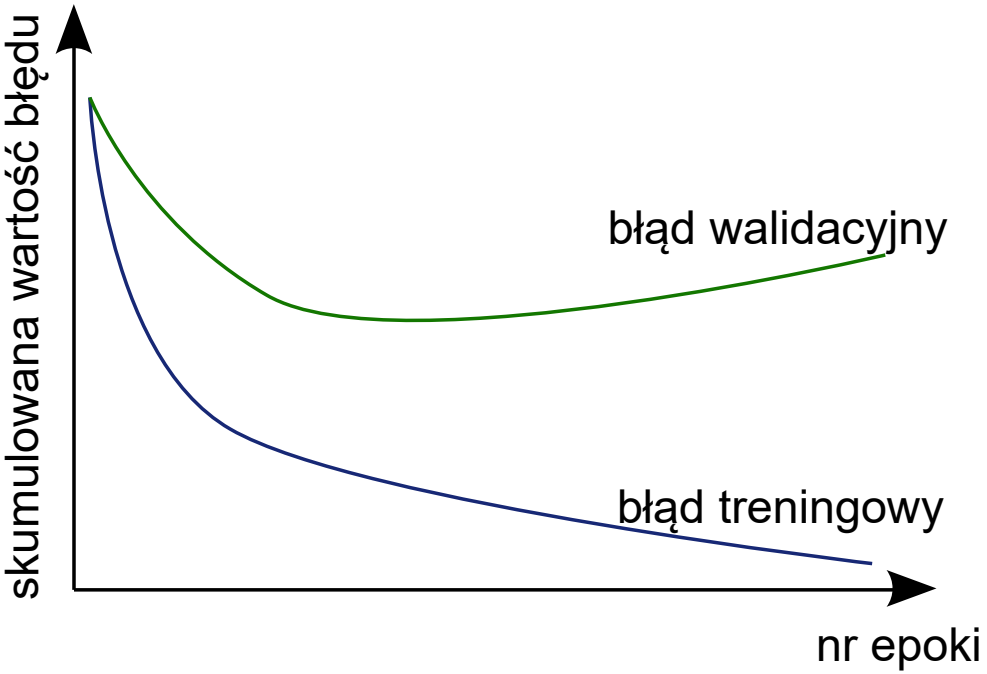
Proces treningu



Zjawisko przetrenowania

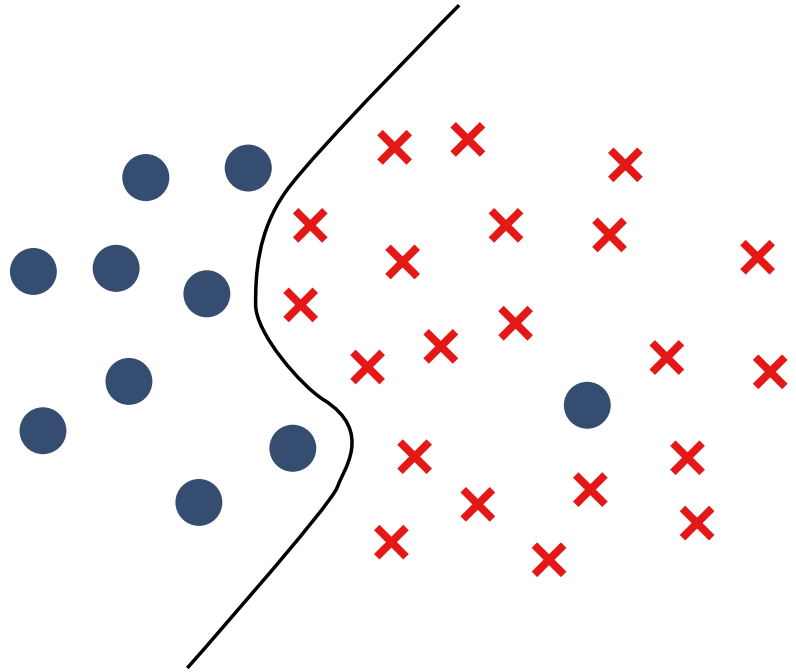


algorytm wytrenowany prawidłowo

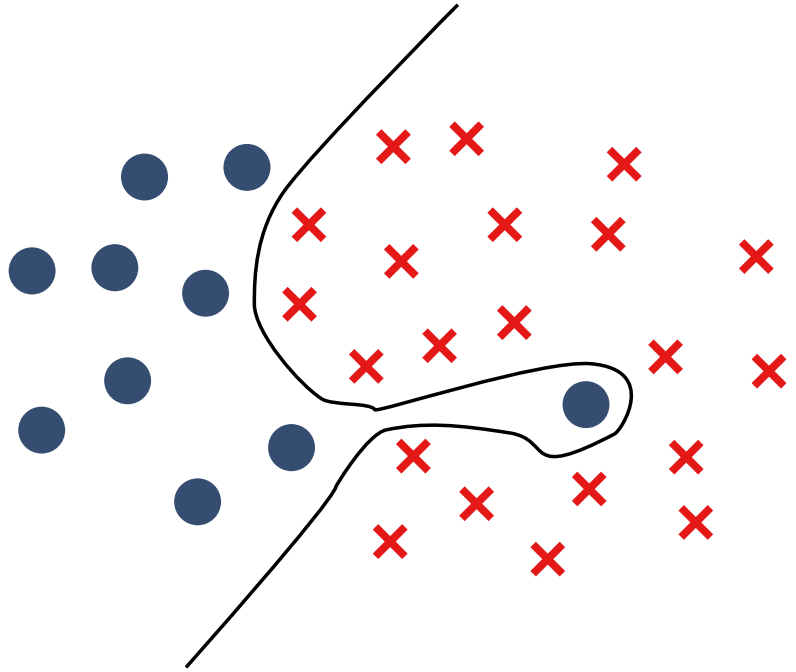


algorytm przetrenowany

Zjawisko przetrenowania



algorytm wytrenowany prawidłowo



algorytm przetrenowany

Typy sieci neuronowych

Ze względu na sposób organizacji neuronów w strukturze sieci neuronowej możemy wyróżnić kilka typów sieci:

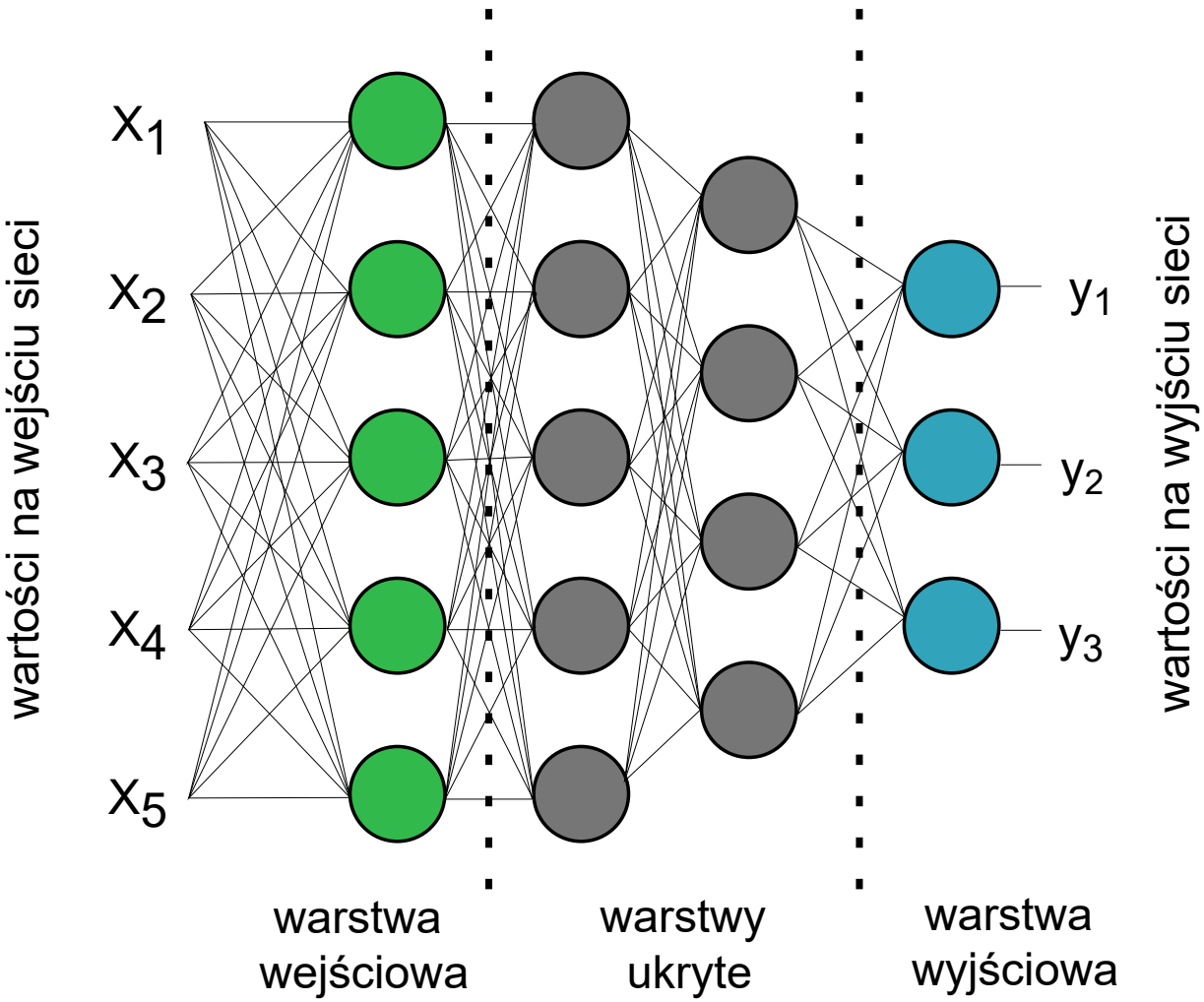
- **sieć płaska (ang. *feedforward network*)** – najbardziej typowa odmiana sieci neuronowej, w której neurony ułożone są w warstwy (przynajmniej wejściową i wyjściową, mogą zawierać warstwy ukryte). Jej struktura jest płaska, bez sprzężeń zwrotnych, warstwy sieci połączone są na zasadzie „każde wyjście z każdym wejściem”.
- **splotowe (ang. *convolutional network*)** – sieć w której współczynniki neuronu definiują operacje splotu na danych wejściowych zorganizowanych w postaci *n*-wymiarowych macierzy. Zwykle stosuje się sploty 1,2 i 3-wymiarowe. Często wykorzystywane są operacje zmniejszające rozmiar danych – tzw. pooling. Ich wielką zaletą jest fakt, że w przetwarzaniu obrazów można na ich wejście przekazywać całe, niesparametryzowane obrazy. Sieć uczy się sposobu parametryzacji obrazów dopasowanego do danych, które otrzymała.

Typy sieci neuronowych (c.d.)

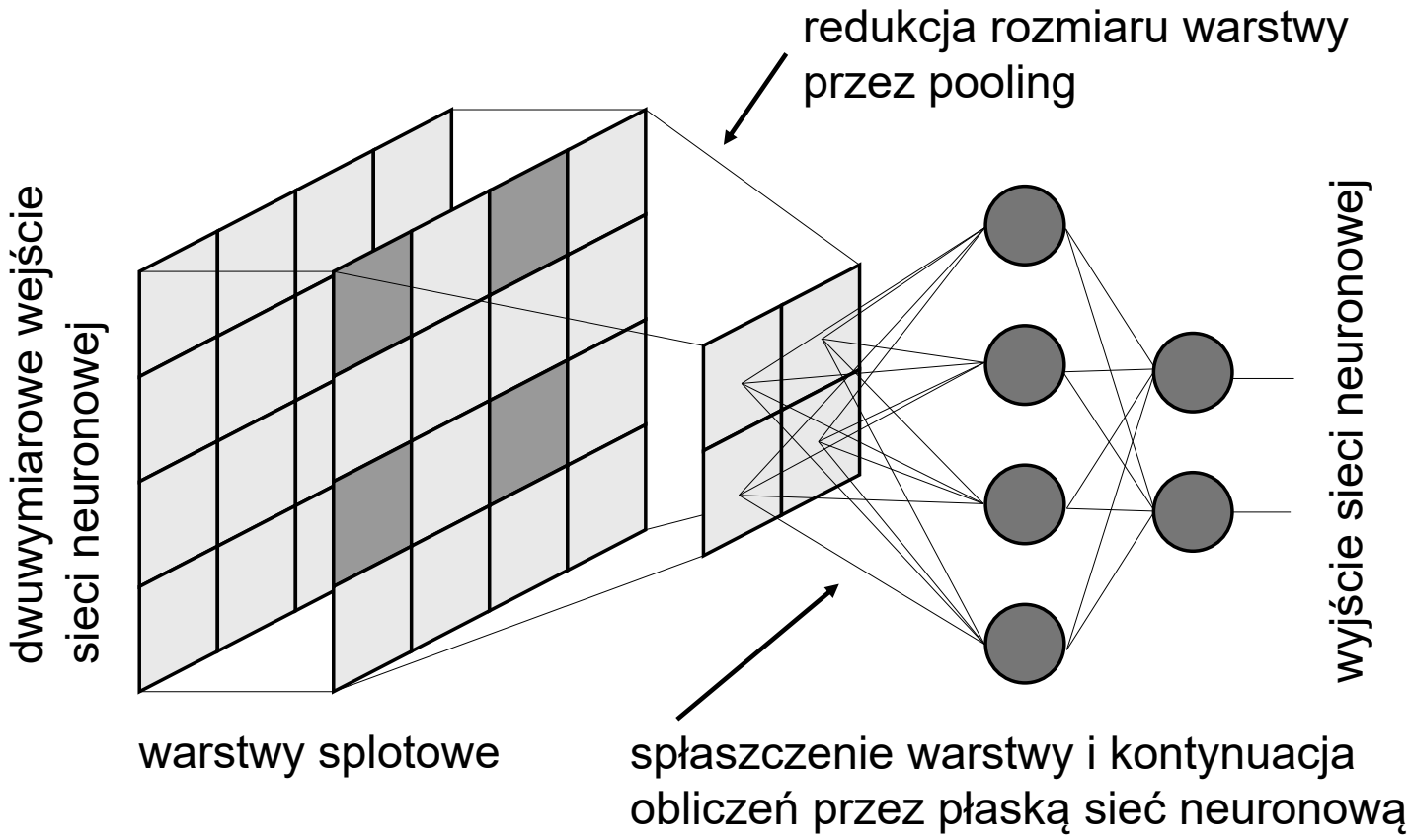
- **rekurencyjne (ang. *recurrent networks*)** – sieci neuronowe, które składają się w części z sieci płaskiej lub splotowej, na której wejście przekierowane są wszystkie lub część wyjść z płaskiego fragmentu sieci – jest to tzw. sprzężenie zwrotne. Sieci tego typu nadają się do analizowania sekwencji obiektów i szeregów czasowych.
- **LSTM (ang. *long short-term memory networks*)** – tzw. sieci z długą pamięcią krótkotrwałą, jest to specyficzny rodzaj sieci rekurencyjnej, która dodatkowo rozbudowana jest o komórki pamięci. Jej struktura jest bardziej skomplikowana niż sieci rekurencyjnych, gdyż zawiera ona specjalne podsieci sterujące procesem zapisu i zwalniania danych z komórek pamięci. Pozwala ona na analizę dłuższych sekwencji.

Istnieją także inne rodzaje sieci przystosowane do specyficznych zastosowań takie jak np. sieci Kohonena, czy Hopfielda.

Typy sieci neuronowych – sieć płaska



Typy sieci neuronowych – sieć splotowa



Przykład operacji w sieci splotowej

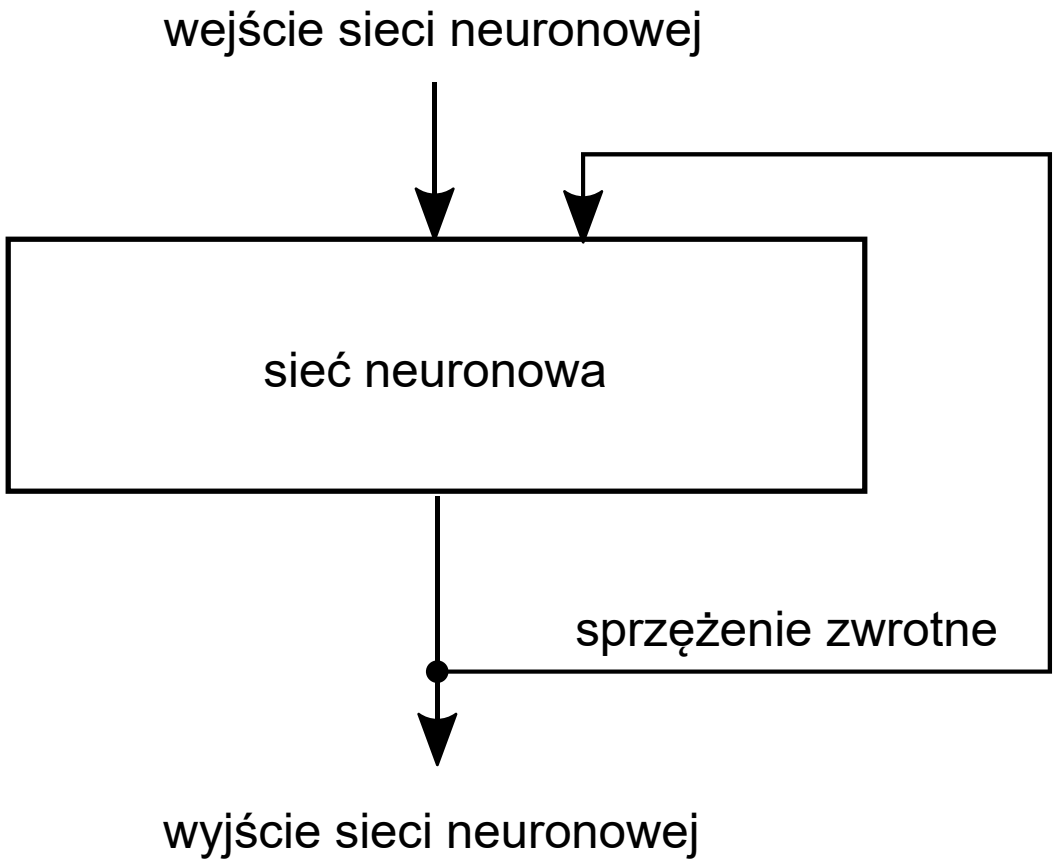
Operacja splotu dwuwymiarowego (możliwe są inne liczby wymiarów):

$$\begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{bmatrix} * \begin{bmatrix} 1 & -1 \end{bmatrix} = \begin{bmatrix} -1 & 0 & 1 & 0 \\ -1 & 0 & 1 & 0 \\ -1 & 0 & 1 & 0 \\ -1 & 0 & 1 & 0 \end{bmatrix}$$

Operacja zmniejszania rozmiaru warstwy przez wybór największej wartości (ang. *max pooling):*

$$\begin{bmatrix} -1 & -2 & 1 & 0 \\ -1 & 0 & 1 & 0 \\ -1 & 0 & 1 & 5 \\ -1 & 0 & 1 & 0 \end{bmatrix} \xrightarrow[\text{max pooling}]{} \begin{bmatrix} -1 & 1 \\ -1 & 5 \end{bmatrix}$$

Typy sieci neuronowych – sieć rekurencyjna



Uczenie nadzorowane

Uczenie nadzorowane (ang. *supervised learning*) to typ uczenia maszynowego, w którym znane są dane wejściowe (oznaczane jako macierz $\boldsymbol{X}$) i oczekiwane dane wyjściowe, które powinny być obliczone na podstawie macierzy $\boldsymbol{X}$ i które często oznaczamy jako $\boldsymbol{Y}$.

Algorytm uczenia maszynowego uczy się przekształcenia możliwego do zapisania jako:

$$\boldsymbol{Y} = f(\boldsymbol{X}).$$

Typowymi przykładami uczenia nadzorowanego są **klasyfikacja**, czy **regresja**.

Uczenie nadzorowane

Przykład zbiorów danych możliwych do wykorzystania w dwóch przykładowych zadaniach należących do kategorii uczenia nadzorowanego (klasyfikacji i regresji).

Nazwa obiektu	Oczekiwana wartość na wyjściu sieci	Nazwa klasy
kot	[0,0,1]	zwierzę
pies	[0,0,1]	zwierzę
talerz	[0,1,0]	przedmiot
sosna	[0,0,1]	roślina
samochód	[0,1,0]	przedmiot

przykład zbioru danych przygotowanych do treningu klasyfikatora

Wartość na wejściu sieci	Oczekiwana wartość na wyjściu sieci
1	0,1
3	0,3
1.2	0,12
8	0,8
2	0,2

przykład zbioru danych przygotowanych do treningu modelu dokonującego regresji

Uczenie nienadzorowane

Uczenie nienadzorowane (ang. *unsupervised learning*) charakteryzuje się tym, że **dana jest jedynie macierz danych wejściowych *X***. Algorytm w takim przypadku analizuje strukturę przekazanych na jego wejście.

Przykładem zadań należących do kategorii uczenia nienadzorowanego są

- **klasteryzacja** – **podział zbioru danych na grupy składające się z podobnych do siebie obiektów**, np. grup osób, które słuchają podobnej muzyki w serwisie streamingowym,
- **asocjacja** – **wyszukiwanie występujących w danych związków przyczynowo-skutkowych**, na przykład odkrywanie na podstawie danych, że młodsi klienci bardziej preferują dokonywanie zakupów on-line, a starsi wolą dokonywać zakupów tradycyjnie.

Uczenie nienadzorowane

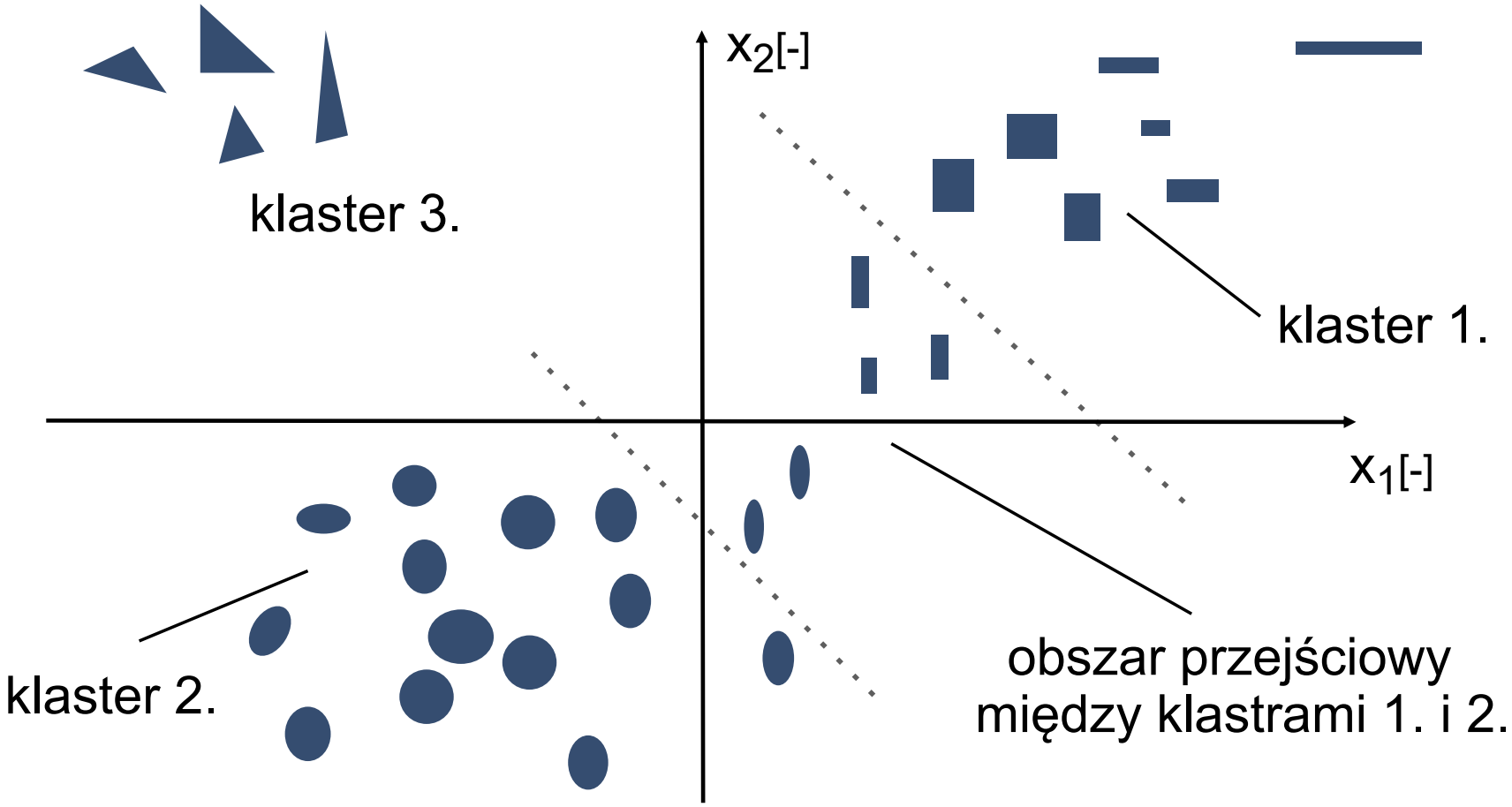
Przykładem uczenia nienadzorowanego są też wspomniane wcześniej **autenkodery**, które uczą się skompresowanych reprezentacji danych z macierzy ***X***.

Tego typu analiza **reprezentacji może być wykorzystana na przykład do wytworzenia nowych przykładów**, podobnych do tych ze zbioru uczącego. Przy specjalnej modyfikacji autoenkodera (tzw. autoenkoderze wariacyjnym) możliwe jest wygenerowanie nowego przykładu o właściwościach podobnych do obiektów z macierzy ***X***.

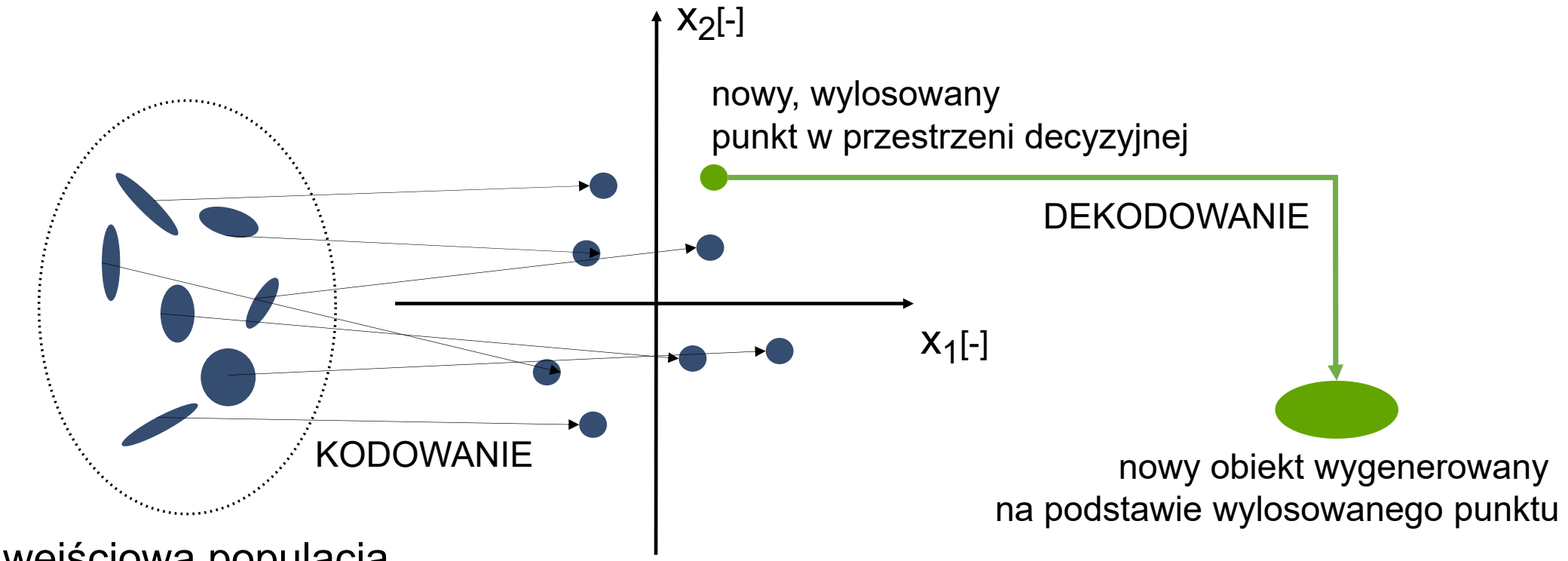
Jest to przykład tzw. modelu generatywnego, który tworzy nowe przykłady na podstawie danych, na których został on wytrenowany.

Innym przykładem popularnego modelu generatywnego są **generatywne sieci przeciwstawne** (ang. ***generative adversarial networks, GANs***).

Klasteryzacja danych



Model generatywny



Literatura

1. Buduma, N., Locasio, N., Fundamentals of Deep Learning. Designing next-generation machine intelligence algorithms, O'Reilly Media, Inc., 2017.
2. Chollet, F., Deep Learning with Python, Manning Publications, 2017.
3. Geron, A., Hands-On Machine Learning with Scikit-Learn & TensorFlow. Concepts, Tools, and Techniques to Build Intelligent Systems, O'Reilly Media, Inc., 2019.
4. Goodfellow, I., Bengio, J., Courville, Aaron, Deep Learning, The MIT Press, 2016.
5. François-Lavet, V., Henderson, P., Islam, R., Bellemare, M.G., & Pineau, J., An Introduction to Deep Reinforcement Learning, Foundations and Trends in Machine Learning 11, 219-354, 2018.

POLITECHNIKA GDAŃSKA

AI TECH

Dziękuję

Adam Kurowski

Rzeczpospolita Polska

Unia Europejska
Europejski Fundusz
Rozwoju Regionalnego

Projekt współfinansowany ze środków Unii Europejskiej w ramach Europejskiego Funduszu Rozwoju Regionalnego
Program Operacyjny Polska Cyfrowa na lata 2014-2020.
Oś priorytetowa nr 3 „Cyfrowe kompetencje społeczeństwa”, działanie nr 3.2 „Innowacyjne rozwiązania na rzecz aktywizacji cyfrowej”.
Tytuł projektu: „Akademia Innowacyjnych Zastosowań Technologii Cyfrowych (AI Tech)”.