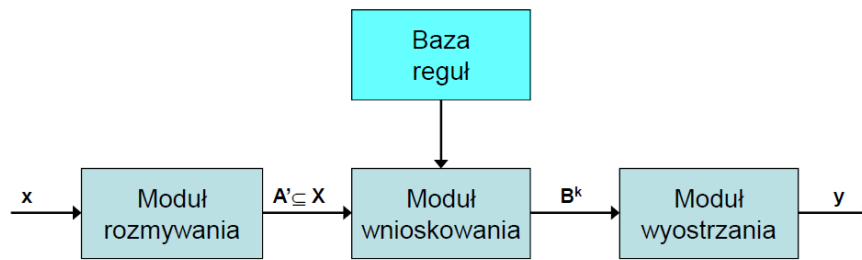


Podaj schemat blokowy sterownika rozmytego. Omów funkcje podstawowych elementów sterownika.



x – sygnały wejściowe (liczbowe)

y – sygnały wyjściowe (liczbowe)

A' , B^k – zbiory rozmyte

Moduł rozmywania: Moduł rozmywania służy do zastąpienia każdej „ostrej” wartości wejściowej zbiorem rozmytym. Jest to konieczne, ponieważ moduł wnioskujący działa na zbiorach rozmytych. Konkretnie wartości liczbowe sygnałów wejściowych zastępowane są odpowiednimi wartościami zmiennych lingwistycznych np. „mały”, „średni”, „duży”. Często stosowanym rozwiązaniem jest zastąpienie każdej wartości ostrej odpowiednim singletonem rozmytym. Singleton rozmyty jest to zbiór rozmyty, który stanowi jeden punkt x_0 z wartością $m(x_0) \in [0,1]$.

Baza reguł: Baza reguł stanowi reprezentację wiedzy eksperta o typowych i możliwych wartościach zmiennych stanu i zmiennych sterowania, o zachowaniu się urządzenia pod wpływem zmian sterowania, o pożądanym stanie urządzenia itd..

Baza ta składa się z reguł rozmytych postaci:

IF (x_1 jest A_1) AND (x_2 jest A_2) AND ... AND (x_n jest A_n)

THEN (y_1 jest B_1) AND (y_2 jest B_2) AND ... AND (y_m jest B_m)

Baza reguł rozmytych nazywana jest modelem lingwistycznym procesu sterowania.

Moduł wnioskowania

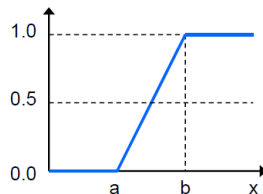
Głównym elementem sterownika rozmytego jest moduł wnioskowania. Działa on w oparciu o jeden schemat wnioskowania przybliżonego np. modus ponens. Sposób wyliczenia zbioru wynikowego zależy w tym schemacie od wyboru T-normy. W sterowaniu rozmytym najczęściej przyjmuje się operację minimum lub produktu (iloczynu algebraicznego). W czasie wnioskowania, dla każdej reguły z bazy określa się stopień prawdziwości części przesłankowej (na podstawie stopnia prawdziwości każdego ze stwierdzeń). Na tej podstawie określa się stopień prawdziwości konkluzji (zakłada się, są one sobie równe).

Moduł wyostrzania: Moduł wyostrzania zamienia zbiory rozmyte na ostre wartości liczbowe (rzeczywiste wartości sterujące). Jeżeli uzyskany na wyjściu modułu wnioskowania zbiór rozmyty ma pojedynczą wartość szczytową (nie występują maksima lokalne), to należy ją wybrać jako poszukiwaną wartość ostrą. Na ogół wybór wartości ostrej nie jest zadaniem prostym. Stosowane są wówczas różne metody wyostrzania np. metoda środka ciężkości lub metoda średniej z maksimów.

Podaj definicję zbioru rozmytego. Omów 4 wybrane klasy funkcji przynależności dla zbiorów rozmytych.

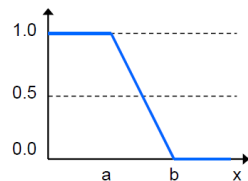
Zbiór rozmyty to zbiór, w którym każdemu elementowi z przestrzeni odniesienia przyporządkowana jest funkcja przynależności, przyjmująca wartości w przedziale $[0,1]$, określająca stopień przynależności elementu do tego zbioru.

➤ funkcja klasy Γ (gamma)



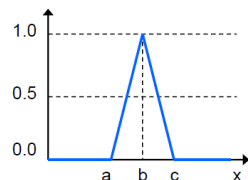
$$\Gamma_{a,b} = \begin{cases} 0 & \text{dla } x \leq a \\ (x-a)/(b-a) & \text{dla } a < x \leq b \\ 1 & \text{dla } x > b \end{cases}$$

➤ funkcja klasy L



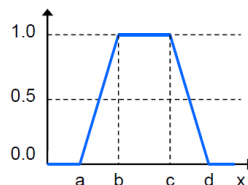
$$L_{a,b} = \begin{cases} 1 & \text{dla } x \leq a \\ (b-x)/(b-a) & \text{dla } a < x \leq b \\ 0 & \text{dla } x > b \end{cases}$$

➤ funkcja klasy Λ (lambda)



$$\Lambda_{a,b,c} = \begin{cases} 0 & \text{dla } x \leq a \vee x \geq c \\ (x-a)/(b-a) & \text{dla } a < x \leq b \\ (c-x)/(c-b) & \text{dla } b < x < c \end{cases}$$

➤ funkcja klasy Π (Pi)



$$\Pi_{a,b,c,d} = \begin{cases} 0 & \text{dla } x \leq a \vee x \geq d \\ (x-a)/(b-a) & \text{dla } a < x \leq b \\ 1 & \text{dla } b < x \leq c \\ (d-x)/(d-c) & \text{dla } c < x < d \end{cases}$$

Gamma- modelowanie wyrażeń typu „duży”, „wysoki”

L- modelowanie wyrażeń typu „mały”, „niski”

Lambda- modelowanie wyrażeń typu „średni”, „umiarkowany”

Omów architekturę i zastosowanie konwolucyjnych sieci neuronowych (CNN)

Architektura:

Konwolucyjne sieci neuronowe (CNN – *Convolutional Neural Networks*) to specjalny typ sieci neuronowych, które zostały zaprojektowane z myślą o danych przestrzennych, takich jak obrazy. Kluczowe cechy architektury CNN to:

1. **Lokalne połączenia** – jednostki w pierwszej warstwie ukrytej otrzymują dane z lokalnych regionów obrazu (tzw. receptywne pola), co pozwala uchwycić lokalne cechy (np. krawędzie, tekstury).
2. **Jądro konwolucyjne (kernel)** – to zestaw wag, który jest przesuwany po obrazie w operacji zwanej konwolucją (splotem), tworząc tzw. mapę cech (feature map). Schemat ten jest wspólny dla wielu lokalnych regionów, co drastycznie zmniejsza liczbę parametrów.
3. **Warstwy splotowe i zagęszczające (np. pooling)** – po warstwach konwolucyjnych często stosuje się warstwy, które redukują wymiar danych i zwiększają odporność na przesunięcia, np. przez wybieranie maksymalnej wartości (max pooling) z danego obszaru.
4. **Warstwy w pełni połączone** – na końcu sieci często występują warstwy gęste (fully connected), które służą do klasyfikacji na podstawie wyodrębnionych wcześniej cech.

Zastosowania:

CNN znajdują zastosowanie wszędzie tam, gdzie istotna jest analiza danych przestrzennych i lokalnych wzorców, w szczególności:

- rozpoznawanie obrazów i obiektów (np. cyfry w zbiorze MNIST, twarze, znaki drogowe),
- analiza i klasyfikacja obrazów medycznych,
- rozpoznawanie mowy i dźwięku (po odpowiednim przekształceniu sygnału np. do postaci spektrogramu),
- przetwarzanie języka naturalnego (w ograniczonym zakresie),
- wykrywanie anomalii w danych czasowych (jeśli traktowane są jako „obraz” danych).

Omów metody selekcji osobników stosowane w tradycyjnym algorytmie genetycznym

1. Metoda koła ruletki (roulette wheel selection)

Każdemu chromosomowi przypisuje się wycinek koła proporcjonalny do jego przystosowania. Im wyższa wartość funkcji przystosowania, tym większe prawdopodobieństwo, że dany osobnik zostanie wybrany do nowej populacji. Wadą tej metody jest spadek presji selekcyjnej, gdy osobniki są do siebie podobne.

2. Selekcja rankingowa

Osobniki są sortowane według wartości funkcji przystosowania, a następnie przypisywane są im rangi. Prawdopodobieństwo wyboru zależy od rangi, a nie od bezpośredniej wartości przystosowania. Metoda ta jest mniej wrażliwa na różnice wartości funkcji przystosowania i zapewnia bardziej równomierne traktowanie osobników.

3. Selekcja turniejowa

Polega na losowym wyborze kilku osobników (np. 2–5), którzy rywalizują ze sobą w tzw. „turnieju”. Zwycięzca zostaje wybrany do nowej populacji. Proces powtarzany jest do utworzenia całej nowej populacji. Ta metoda jest prosta i efektywna.

4. Selekcja ($\mu + \lambda$)

Choć szczegóły tej metody nie są rozwinięte w pliku, jej nazwa sugeruje podejście znane z ewolucyjnych strategii, w którym do kolejnej generacji wybierani są najlepsi osobnicy spośród populacji rodziców (μ) i potomków (λ).

Podaj algorytm programowania genetycznego. Omów sposób reprezentacji programów w programowaniu genetycznym.

Algorytm programowania genetycznego:

1. **Wygenerowanie początkowej populacji programów** – losowo tworzy się zestaw programów złożonych z funkcji i terminali, właściwych dla rozważanego problemu.
2. **Ocena programów** – każdy program jest uruchamiany i oceniany przy pomocy funkcji przystosowania (np. liczba poprawnych wyników, czas działania, złożoność).
3. **Tworzenie nowej populacji:**
 - o kopiowanie istniejących programów,
 - o tworzenie nowych programów przez **rekombinację (krzyżowanie)** fragmentów (np. poddrzew) wybranych osobników.
4. **Zakończenie** – po osiągnięciu kryterium (np. liczba pokoleń, osiągnięcie dopasowania), jako wynik przyjmowany jest **najlepszy program z całej historii ewolucji**.

Reprezentacja programów w programowaniu genetycznym:

- Programy są reprezentowane jako **drzewa**:
 - o **węzły** zawierają funkcje (np. arytmetyczne, logiczne),
 - o **liście (terminalne)** zawierają zmienne, stałe, ewentualnie funkcje bez argumentów.
- Struktury te są **dynamiczne** – rozmiar i kształt programu może się zmieniać w trakcie ewolucji.
- Każdy program to złożenie elementów ze zbioru funkcji $F = \{f_1, f_2, \dots, f_N\}$ oraz terminali $T = \{a_1, a_2, \dots, a_N\}$ – program powstaje przez rekurencyjne łączenie funkcji i terminali.
- Przykład drzewa dla wyrażenia $a + 2 * b * c$ ilustruje ten sposób reprezentacji.

Podaj algorytm programowania genetycznego. Omów operatory ewolucyjne stosowane w programowaniu genetycznym.

Algorytm programowania genetycznego:

1. Wygenerowanie początkowej populacji programów.
2. Wykonanie iteracyjne następujących akcji:
 - a. uruchomienie każdego programu z populacji i określenie dla niego wartości funkcji przystosowania
 - b. utworzenie nowej populacji programów przez zastosowanie dwóch podstawowych operacji:
 - i. kopiowanie istniejącego programu do nowej populacji,
 - ii. tworzenie nowego programu przez krzyżowanie (rekombinację losowo wybranych części dwóch istniejących programów)
3. Najlepszy program który pojawił się w dowolnym pokoleniu jest rozwiązaniem optymalnym (lub bliskim optymalnego).

Podstawowe operatory do tworzenia nowej populacji to:

- reprodukcja: Reprodukacja polega na wyborze osobnika (programu) z danej populacji i przekopiowaniu go (bez zmian) do nowej populacji. Najbardziej popularna jest selekcja proporcjonalna do dopasowania (np. dopasowanie znormalizowane). Stosowane są również inne metody np. metoda rang lub metoda turniejowa.
- krzyżowanie: Dwa osobniki, które będą pełnić funkcję rodziców, wybierane są na podstawie ich funkcji dopasowania. Następnie w każdym z rodziców wybierany jest (niezależnie) w sposób losowy punkt krzyżowania. Fragment używany przy krzyżowaniu jest poddrzewem o korzeniu w punkcie krzyżowania i zawierającym cały fragment poniżej punktu krzyżowania. Poddrewo może zredukować się do jednego tylko terminala lub stanowić całe drzewo dla danego rodzica.

Algorytm Genetyczny:

Mechanizm działania klasycznego algorytmu genetycznego:

1. Losowana jest pewna populacja początkowa.
2. Populacja poddawana jest ocenie (selekcja). Najlepiej przystosowane osobniki biorą udział w reprodukcji.
3. Genotypy wybranych osobników poddawane są operatorom ewolucyjnym:
 - a. krzyżowania (odpowiednie połączenie genotypów rodziców),
 - b. mutacji (wprowadzenia do genotypów drobnych losowych zmian).
4. Utworzenie kolejnego pokolenia (nowej populacji). Jeżeli nie znaleziono dostatecznie dobrego rozwiązania, to powrót do punktu 2. W przeciwnym wypadku uzyskujemy wynik.

Omów algorytm minimaksu. Podaj przykład zastosowania tego algorytmu.

Twierdzenie minimaksowe: Każda skończona gra dwuosobowa o sumie zerowej ma co najmniej jedno rozwiązanie, które określa wartość gry i optymalne strategie graczy.

Strategia minimaksu:

- utwórz drzewo gry o maksymalnej głębokości
- przypisz wierzchołkom końcowym drzewa (liściom) wartości zgodnie z przyjętą funkcją oceny np. +1, -1, 0 (w przypadku dużych drzew użyj funkcji heurystycznej)
- wierzchołkom na kolejnych poziomach (od liści do korzenia) nadaj wartości zgodnie z zasadą: maksymalna wartość potomka dla gracza oraz minimalna wartość potomka dla przeciwnika
- po osiągnięciu korzenia wybierz strategię, która zapewnia największe zyski

Najczęściej spotykanym zastosowaniem algorytmu minimaksu są gry planszowe:

- Szachy,
- Warcaby,
- Kółko i krzyżyk (Tic-Tac-Toe).

Algorytm cięć α - β :

Algorytm cięć α - β stanowi modyfikację algorytmu minimaksowego. W algorytmie tym węzły nie wpływające na wartość przypisywaną ich przodkom są eliminowane z dalszej analizy (znaczące ograniczenie drzewa gry). W algorytmie występują dwa rodzaje cięć:

- **cięcie α** - dla wierzchołków na poziomie gracza ocenę wierzchołków potomnych można zakończyć, gdy wartość będzie niższa niż dotychczasowe maksimum α
- **cięcie β** - dla wierzchołków na poziomie przeciwnika ocenę wierzchołków potomnych można zakończyć, gdy wartość będzie wyższa niż dotychczasowe minimum β