

RESUME

Pratikum Artificial Intelligence



DOSEN PENGAMPU:

Widya Darwin S.Pd,MPdt

OLEH:

Elsa Rahma Hidayani

21346006

Informatika

PROGRAM STUDI INFORMATIKA

JURUSAN ELEKTRONIKA

FAKULTAS TEKNIK

UNIVERSITAS NEGERI PADANG

2023

DAFTAR ISI

DAFTAR ISI	2
BAB I DEFINISI KECERDASAN BUATAN	3
A. Definisi Kecerdasan Buatan	3
B. Analisis Ruang Masalah Kecerdasan Buatan dan Aturan Produksi	3
BAB II Metode Pencarian dalam Kecerdasan Buatan	5
A. Metode Pencarian Buta:	5
B. Metode Pencarian Heuristik:	5
C. Logika Proposisi	6
BAB III First Order Logic/Predicate Calculus	7
BAB IV FUZZY LOGIC	8
A. Pengenalan Konsep Fuzzy Logic	8
B. Fuzziness dan Probabilitas	8
C. Fuzzy Set	9
D. Fuzzy Logic	10
E. Fuzzy Systems	11
BAB V PLANNING (TEKNIK DEKOMPOSISI MASALAH) - GOAL STACK PLANNING (GSP) - CONSTRAINT POSTING (CP)	13
BAB VI DECISION TREE LEARNING	14
A. Konsep Dasar Tree Learning	14
B. ID3 (Iterative Dichotomiser 3)	15
C. C4.5	17
BAB VII JARINGAN SYARAF TIRUAN (JST)	20
A. Konsep Jaringan Syaraf Tiruan (Jst)	20
B. Model Syaraf Toruan	20
C. Aktivasi Dan Arsitektur Jaringan Pada Jst	21
D. Supervised Learning Danunsupervised Learning	22
BAB VIII ALGORITMA GENETIKA (AG)	24
Daftar Pustaka:	25

BAB I

DEFINISI KECERDASAN BUATAN

A. Definisi Kecerdasan Buatan

Kecerdasan Buatan (Artificial Intelligence/AI) adalah bidang ilmu komputer yang berfokus pada pengembangan sistem komputer yang dapat meniru atau menampilkan kecerdasan manusia. AI melibatkan penggunaan algoritma dan teknik komputasi untuk mengumpulkan, menganalisis, dan menginterpretasi data dengan tujuan mengambil keputusan atau melakukan tugas yang kompleks.

Terdapat beberapa teknik yang digunakan dalam AI untuk menyelesaikan permasalahan:

- a. Machine Learning (ML): Teknik yang memungkinkan komputer belajar dari data dan pengalaman untuk mengidentifikasi pola atau membuat prediksi. Dalam ML, terdapat dua pendekatan utama: supervised learning (pembelajaran terarah) dan unsupervised learning (pembelajaran tanpa pengawasan).
- b. Deep Learning: Teknik yang menggunakan neural networks dengan banyak lapisan (layers) untuk mempelajari representasi yang lebih kompleks dari data. Deep learning telah memberikan kemajuan signifikan dalam bidang pengenalan wajah, pengenalan suara, dan pemrosesan bahasa alami.
- c. Natural Language Processing (NLP): Bidang AI yang berfokus pada pemahaman dan penghasilan bahasa manusia oleh komputer. NLP digunakan dalam aplikasi seperti chatbot, penerjemahan mesin, dan analisis sentimen.
- d. Computer Vision: Teknik yang memungkinkan komputer untuk melihat, memahami, dan menganalisis gambar atau video. Computer vision digunakan dalam pengenalan objek, deteksi wajah, dan pengolahan citra medis.
- e. Reinforcement Learning: Pendekatan AI di mana sebuah agen belajar berinteraksi dengan lingkungannya dan mendapatkan umpan balik dalam bentuk reward atau hukuman. Tujuan utama dalam reinforcement learning adalah untuk memaksimalkan reward yang diterima oleh agen.
- f. Genetic Algorithms: Teknik yang terinspirasi dari proses seleksi alam dalam evolusi biologis. Genetic algorithms digunakan untuk mencari solusi optimal dalam masalah kompleks dengan menghasilkan dan memodifikasi populasi solusi yang ada.

B. Analisis Ruang Masalah Kecerdasan Buatan dan Aturan Produksi

Pada tingkat intelektual, kecerdasan buatan (Artificial Intelligence/AI) dapat dibagi menjadi ruang masalah. Ruang masalah mengacu pada lingkungan di mana agen AI beroperasi dan harus menemukan solusi yang optimal untuk mencapai tujuan yang ditetapkan. Kondisi awal (initial state) dan kondisi tujuan (goal state) adalah elemen penting dalam ruang masalah kecerdasan buatan. Kondisi awal adalah keadaan di mana agen AI memulai tugasnya, sementara kondisi tujuan adalah keadaan yang ingin dicapai oleh agen AI.

Sebagai contoh, dalam masalah pencarian jalan terpendek antara dua titik di peta, kondisi awal dapat berupa posisi awal agen dan kondisi tujuan dapat berupa posisi akhir yang ingin dicapai. Agen harus menggunakan teknik dan algoritma AI yang tepat untuk mencari rute terpendek antara kedua kondisi tersebut. Selain itu, dalam ruang masalah kecerdasan buatan,

aturan produksi digunakan untuk menggambarkan pemahaman dan tindakan yang dilakukan oleh agen AI. Aturan produksi terdiri dari kondisi (if) dan aksi (then), yang menghubungkan keadaan di ruang masalah dengan tindakan yang harus diambil oleh agen AI.

Contohnya, dalam sistem pakar medis, aturan produksi dapat berbunyi "Jika pasien memiliki demam (kondisi), maka berikan obat penurun panas (aksi)". Aturan-aturan ini membantu agen AI untuk mengambil keputusan berdasarkan informasi yang diberikan.

BAB II

METODE PENCARIAN DALAM KECERDASAN BUATAN

Dalam kecerdasan buatan (Artificial Intelligence/AI), metode pencarian digunakan untuk mencari solusi yang optimal atau memenuhi kriteria tertentu dalam ruang masalah yang kompleks. Terdapat dua jenis metode pencarian yang umum digunakan, yaitu metode pencarian buta (blind search) dan metode pencarian heuristik.

A. Metode Pencarian Buta:

Metode pencarian buta adalah pendekatan yang tidak mempertimbangkan informasi tambahan tentang ruang masalah selain yang ada pada kondisi awal dan aturan transisi. Beberapa metode pencarian buta yang umum digunakan meliputi:

- a. Breadth-First Search (BFS): Menjelajahi semua simpul dalam setiap level secara sistematis sebelum berpindah ke level berikutnya.
- b. Depth-First Search (DFS): Melanjutkan ke simpul tetangga secara rekursif hingga tidak ada lagi simpul baru yang dapat dijangkau sebelum melakukan mundur.
- c. Uniform-Cost Search: Mengutamakan pencarian dengan biaya terendah dan secara bertahap meningkatkan biaya pencarian seiring dengan kedalaman pencarian.

B. Metode Pencarian Heuristik:

Metode pencarian heuristik menggunakan informasi tambahan yang disebut heuristik untuk membantu mencari solusi secara lebih efisien. Beberapa metode pencarian heuristik yang umum digunakan meliputi:

- a. Best-First Search: Memilih simpul berdasarkan nilai heuristik terbaik, yang merupakan perkiraan jarak atau biaya hingga mencapai tujuan.
- b. A* Search: Menggabungkan heuristik dengan biaya sejauh ini ($g + h$) untuk memilih simpul dengan nilai f terendah, di mana g adalah biaya sejauh ini dan h adalah heuristik yang mengestimasi biaya tersisa.
- c. Hill Climbing: Melakukan langkah-langkah lokal yang memaksimalkan nilai fungsi evaluasi dengan harapan mencapai puncak (solusi optimal). Namun, metode ini dapat terjebak pada maksima lokal.

Metode pencarian yang digunakan akan tergantung pada kompleksitas ruang masalah dan ketersediaan informasi tambahan. Pencarian buta efektif untuk ruang masalah kecil tanpa informasi tambahan, sementara pencarian heuristik lebih cocok untuk ruang masalah yang lebih kompleks dengan heuristik yang dapat membantu mengarahkan pencarian ke solusi yang lebih baik.

C. Logika Proposisi

Logika proposisi adalah cabang logika matematika yang berfokus pada pemodelan dan penalaran tentang pernyataan atau proposisi. Dalam logika proposisi, aturan inferensi digunakan untuk menarik kesimpulan yang valid dari pernyataan-pernyataan yang ada. Beberapa aturan inferensi yang umum digunakan dalam logika proposisi meliputi:

Modus Ponens: Jika pernyataan A mengimplikasikan pernyataan B ($A \rightarrow B$) dan pernyataan A benar, maka dapat disimpulkan bahwa pernyataan B juga benar.

Contoh: Jika hujan ($A \rightarrow B$) dan memang sedang hujan (A), maka dapat disimpulkan bahwa jalanan basah (B).

Modus Tollens: Jika pernyataan A mengimplikasikan pernyataan B ($A \rightarrow B$) dan pernyataan B tidak benar, maka dapat disimpulkan bahwa pernyataan A juga tidak benar.

Contoh: Jika cuaca cerah ($A \rightarrow B$) dan memang tidak hujan ($\neg B$), maka dapat disimpulkan bahwa tidak ada awan ($\neg A$).

Chain Rule: Jika pernyataan A mengimplikasikan pernyataan B ($A \rightarrow B$) dan pernyataan B mengimplikasikan pernyataan C ($B \rightarrow C$), maka dapat disimpulkan bahwa pernyataan A mengimplikasikan pernyataan C ($A \rightarrow C$).

Contoh: Jika matahari bersinar ($A \rightarrow B$) dan jika matahari bersinar, maka taman ramai dikunjungi ($B \rightarrow C$), maka dapat disimpulkan bahwa jika matahari bersinar, taman ramai dikunjungi ($A \rightarrow C$).

BAB III

FIRST ORDER LOGIC/PREDICATE CALCULUS

Logika Predikat, juga dikenal sebagai Logika Orde Pertama atau Predicate Calculus, merupakan cabang penting dalam logika matematika yang memungkinkan analisis dan penalaran tentang hubungan kompleks antara objek, konsep, dan properti dalam dunia yang didefinisikan. Mahasiswa memperoleh kompetensi untuk menganalisa masalah dengan menggunakan aturan inferensi pada Logika Predikat, yang memungkinkan mereka untuk memahami secara mendalam pernyataan logika yang kompleks dan menarik kesimpulan yang valid.

Dalam Logika Predikat, pernyataan dibentuk menggunakan variabel, konstanta, fungsi, dan predikat. Variabel digunakan untuk merepresentasikan objek umum, sedangkan konstanta adalah objek khusus dalam domain yang didefinisikan. Fungsi adalah hubungan yang menghubungkan satu atau lebih argumen dengan nilai tertentu, sedangkan predikat adalah pernyataan yang mengaitkan argumen dengan nilai kebenaran.

Aturan inferensi dalam Logika Predikat memungkinkan mahasiswa untuk melakukan penalaran yang lebih kuat dan kompleks. Beberapa aturan inferensi penting yang digunakan dalam Logika Predikat antara lain:

- a. Modus Ponens: Jika pernyataan A mengimplikasikan pernyataan B ($A \rightarrow B$) dan pernyataan A benar, maka dapat disimpulkan bahwa pernyataan B juga benar.

Contoh: Jika semua manusia adalah makhluk hidup ($\forall x: \text{Manusia}(x) \rightarrow \text{Hidup}(x)$) dan individu A adalah manusia ($\text{Manusia}(A)$), maka dapat disimpulkan bahwa individu A adalah makhluk hidup ($\text{Hidup}(A)$).

- b. Universal Instantiation (UI): Jika suatu pernyataan benar untuk semua objek dalam domain, maka pernyataan itu benar untuk objek tertentu dalam domain.

Contoh: Jika semua manusia adalah makhluk hidup ($\forall x: \text{Manusia}(x) \rightarrow \text{Hidup}(x)$), maka individu A adalah makhluk hidup ($\text{Hidup}(A)$).

- c. Existential Generalization (EG): Jika suatu pernyataan benar untuk objek tertentu dalam domain, maka pernyataan itu benar untuk setidaknya satu objek dalam domain.

Contoh: Jika ada seorang mahasiswa yang pandai dalam studinya ($\exists x: \text{Mahasiswa}(x) \wedge \text{Pandai}(x)$), maka dapat disimpulkan bahwa terdapat seorang mahasiswa yang pandai dalam studinya.

Dengan menggunakan aturan inferensi ini, mahasiswa dapat menganalisa masalah dan membuat kesimpulan yang valid dalam konteks Logika Predikat. Mereka dapat mengidentifikasi dan mengaplikasikan aturan inferensi yang relevan untuk mengevaluasi kebenaran pernyataan, memahami hubungan antarpernyataan, dan menyimpulkan informasi yang baru.

BAB IV

FUZZY LOGIC

A. Pengenalan Konsep Fuzzy Logic

Fuzzy Logic adalah salah satu cabang logika matematika yang menggambarkan dan memodelkan ketidakpastian dan ambiguitas dalam pemrosesan informasi. Berbeda dengan logika tradisional yang menggunakan nilai biner (0 atau 1) untuk menyatakan kebenaran suatu pernyataan, Fuzzy Logic memperkenalkan konsep tingkat keanggotaan dalam rentang nilai antara 0 dan 1. Konsep ini memungkinkan representasi yang lebih fleksibel dan lebih dekat dengan cara berpikir manusia.

Istilah "fuzzy" sendiri berarti "samar" atau "kabur" dalam bahasa Inggris. Konsep ini mencerminkan sifat ketidakpastian dan kekurangan informasi yang sering dihadapi dalam dunia nyata. Misalnya, ketika kita menggambarkan warna biru dan hijau, ada daerah di antara keduanya yang sulit untuk ditentukan apakah termasuk biru atau hijau dengan tegas. Dalam logika tradisional, warna tersebut harus diklasifikasikan sebagai biru atau hijau, tetapi dalam Fuzzy Logic, kita dapat memberikan tingkat keanggotaan biru dan hijau untuk mencerminkan tingkat kebiruan dan kehijauan tersebut.

Fuzzy Logic memiliki aplikasi yang luas dalam berbagai bidang, termasuk sistem kendali, kecerdasan buatan, pengambilan keputusan, pengenalan pola, pemrosesan bahasa alami, dan banyak lagi. Dengan menggunakan konsep Fuzzy Logic, kita dapat mengatasi ketidakpastian dan ambiguitas yang terkait dengan data yang tidak lengkap atau subyektif.

Penerapan Fuzzy Logic memungkinkan sistem untuk menghasilkan solusi yang lebih fleksibel dan lebih adaptif, terutama ketika terdapat banyak variabel yang saling berinteraksi atau ketika informasi yang tersedia tidak lengkap atau tidak presisi. Dalam Fuzzy Logic, penalaran dilakukan berdasarkan aturan-aturan yang bersifat heuristik atau berbasis pengalaman, mirip dengan cara berpikir manusia.

Dengan adanya Fuzzy Logic, kita dapat mengambil keputusan yang lebih baik dalam situasi yang kompleks dan tidak pasti, serta memodelkan pengetahuan yang tidak tepat atau kurang presisi. Konsep Fuzzy Logic merupakan salah satu alat yang berguna dalam menganalisis dan memahami dunia nyata yang penuh dengan ketidakpastian dan ketidakjelasan.

B. Fuzziness dan Probabilitas

Fuzziness dan Probabilitas merupakan dua konsep yang berhubungan dengan ketidakpastian dan keambiguan dalam pemrosesan informasi.

a. Fuzziness:

Fuzziness merujuk pada sifat dunia nyata yang seringkali tidak tegas dan ambigu. Dalam konteks logika tradisional, suatu pernyataan hanya memiliki nilai benar (1) atau salah (0) secara biner. Namun, dalam Fuzzy Logic, tingkat kebenaran diperkenalkan dengan menggunakan angka pada rentang $[0, 1]$. Fuzziness memungkinkan representasi tingkat keanggotaan atau tingkat kebenaran yang lebih fleksibel di antara nilai biner tersebut.

Misalnya, dalam Fuzzy Logic, warna biru tidak hanya benar atau salah, tetapi dapat memiliki tingkat keanggotaan biru yang beragam.

b. Probabilitas:

Probabilitas adalah konsep yang digunakan untuk menggambarkan tingkat kemungkinan suatu peristiwa terjadi. Probabilitas juga menggunakan angka pada rentang $[0, 1]$, di mana 0 menunjukkan ketidakmungkinan dan 1 menunjukkan kepastian. Probabilitas digunakan dalam teori probabilitas untuk menganalisis peristiwa acak atau tak terduga. Probabilitas menyediakan cara untuk mengukur dan mengestimasi tingkat ketidakpastian dalam suatu peristiwa. Berbeda dengan Fuzzy Logic, probabilitas tidak menggambarkan tingkat keanggotaan dalam arti yang sama, tetapi lebih fokus pada pengukuran kemungkinan suatu peristiwa berdasarkan data atau informasi yang tersedia.

Fuzziness dan probabilitas adalah dua konsep yang berguna dalam mengatasi ketidakpastian dan ambiguitas dalam pemrosesan informasi. Fuzzy Logic menggunakan tingkat keanggotaan untuk menggambarkan tingkat kebenaran atau keanggotaan dalam suatu konsep, sedangkan probabilitas digunakan untuk menggambarkan tingkat kemungkinan suatu peristiwa terjadi berdasarkan data yang ada. Keduanya memiliki peran penting dalam analisis dan pengambilan keputusan dalam situasi yang kompleks dan tidak pasti.

C. Fuzzy Set

a. Definisi Fuzzy Set

Fuzzy set merupakan suatu konsep dalam teori himpunan fuzzy yang memungkinkan suatu elemen memiliki tingkat keanggotaan yang tidak hanya berupa 0 atau 1, tetapi juga dapat berupa bilangan di antara keduanya. Dengan kata lain, fuzzy set memperkenalkan derajat keanggotaan suatu elemen dalam himpunan tersebut dinyatakan dalam bentuk bilangan yang memenuhi kondisi $0 \leq \mu(x) \leq 1$, di mana $\mu(x)$ adalah fungsi keanggotaan yang menggambarkan tingkat keanggotaan elemen x dalam himpunan fuzzy.

b. Operasi pada Fuzzy Set:

1. Gabungan (Union): Operasi gabungan pada fuzzy set dilakukan dengan mempertimbangkan tingkat keanggotaan elemen dari kedua fuzzy set yang terlibat. Gabungan dari dua fuzzy set A dan B dinyatakan dengan $A \cup B$ dan dihitung dengan menggunakan fungsi maksimum dari tingkat keanggotaan masing-masing elemen dalam fuzzy set A dan B.
2. Irisan (Intersection): Operasi irisan pada fuzzy set juga mempertimbangkan tingkat keanggotaan elemen dari kedua fuzzy set yang terlibat. Irisan dari dua fuzzy set A dan B dinyatakan dengan $A \cap B$ dan dihitung dengan menggunakan fungsi minimum dari tingkat keanggotaan masing-masing elemen dalam fuzzy set A dan B.
3. Komplemen (Complement): Operasi komplemen pada fuzzy set dilakukan dengan membalik tingkat keanggotaan elemen dari suatu fuzzy set. Komplemen dari fuzzy set A

dinyatakan dengan A' atau $\neg A$, dan dihitung dengan mengurangi tingkat keanggotaan elemen dalam fuzzy set A dari 1.

c. Fungsi Keanggotaan (Membership Function): Fungsi keanggotaan adalah fungsi matematis yang menggambarkan tingkat keanggotaan suatu elemen dalam himpunan fuzzy. Fungsi keanggotaan ini dapat berbagai macam bentuk, tergantung pada karakteristik dan interpretasi dari himpunan fuzzy yang bersangkutan. Berikut adalah beberapa jenis fungsi keanggotaan yang umum digunakan:

1. Fungsi keanggotaan segitiga (Triangular membership function):

- Bentuk fungsi: $\mu(x) = 0$, jika $x \leq a$; $(x - a) / (b - a)$, jika $a \leq x \leq b$; $(c - x) / (c - b)$, jika $b \leq x \leq c$; 0 , jika $x \geq c$.
- Contoh penggunaan: Fungsi keanggotaan ini sering digunakan untuk menggambarkan variabel dengan nilai-nilai yang terkonsentrasi pada suatu rentang tertentu.

2. Fungsi keanggotaan trapesium (Trapezoidal membership function):

- Bentuk fungsi: $\mu(x) = 0$, jika $x \leq a$; $(x - a) / (b - a)$, jika $a \leq x \leq b$; 1 , jika $b \leq x \leq c$; $(d - x) / (d - c)$, jika $c \leq x \leq d$; 0 , jika $x \geq d$.
- Contoh penggunaan: Fungsi keanggotaan ini cocok digunakan untuk variabel dengan rentang yang terdefinisi secara jelas.

3. Fungsi keanggotaan Gaussian (Gaussian membership function):

- Bentuk fungsi: $\mu(x) = \exp(-0.5 * ((x - c) / \sigma)^2)$.
- Contoh penggunaan: Fungsi keanggotaan ini umum digunakan untuk menggambarkan variabel yang memiliki distribusi normal.

Contoh penggunaan fungsi keanggotaan bisa ditemukan dalam berbagai aplikasi seperti sistem pengontrol logika fuzzy, analisis risiko, pengambilan keputusan, dan lain sebagainya. Misalnya, dalam sistem pengontrol logika fuzzy untuk kendaraan, fungsi keanggotaan digunakan untuk menggambarkan tingkat kecepatan rendah, sedang, atau tinggi, serta tingkat jarak dekat, sedang, atau jauh. Fungsi keanggotaan ini membantu dalam proses pengambilan keputusan berdasarkan input dan aturan yang telah ditentukan.

D. Fuzzy Logic

a. Prinsip Dasar Fuzzy Logic: Fuzzy Logic adalah sebuah logika yang digunakan untuk mengatasi ketidakpastian dan ambiguitas dalam pemodelan dan pengambilan keputusan. Prinsip dasar Fuzzy Logic adalah penggunaan variabel linguistik dan aturan fuzzy untuk menangani pemrosesan informasi yang tidak hanya berdasarkan pada nilai benar atau salah (0 atau 1), tetapi juga memperhitungkan tingkat kebenaran atau ketidakbenaran dalam bentuk derajat keanggotaan antara 0 dan 1.

b. Variabel Linguistik: Variabel linguistik adalah variabel yang nilainya dijelaskan dalam bentuk kata-kata atau frasa yang berhubungan dengan bahasa manusia. Variabel ini menggantikan variabel numerik dalam logika tradisional. Contoh variabel linguistik adalah "kecepatan rendah," "suhu tinggi," atau "pelayanan buruk." Variabel ini memiliki fungsi keanggotaan yang menggambarkan tingkat keanggotaan elemen dalam himpunan fuzzy.

c. Aturan Fuzzy: Aturan fuzzy adalah pernyataan yang menghubungkan kondisi (premises) dengan tindakan (conclusions) dalam bentuk "IF-THEN". Aturan ini digunakan untuk menggambarkan pengetahuan yang tidak presisi atau ambiguitas. Ada dua bentuk umum aturan fuzzy:

1. Bentuk aturan IF-THEN: IF <kondisi> THEN <tindakan>. Contoh: IF suhu tinggi AND kelembaban tinggi THEN nyalakan AC.
2. Contoh aturan fuzzy: IF suhu rendah THEN kecepatan kipas rendah. IF suhu sedang THEN kecepatan kipas sedang. IF suhu tinggi THEN kecepatan kipas tinggi.

d. Proses Inferensi Fuzzy: Proses inferensi fuzzy terdiri dari tiga tahap utama:

1. Fuzzyfikasi: Fuzzyfikasi adalah proses mengubah input numerik ke dalam bentuk keanggotaan dalam himpunan fuzzy. Hal ini dilakukan dengan mengaplikasikan fungsi keanggotaan pada variabel input numerik untuk mendapatkan derajat keanggotaan masing-masing dalam himpunan fuzzy yang sesuai.
2. Evaluasi Aturan: Evaluasi aturan melibatkan penerapan aturan fuzzy pada variabel linguistik input. Setiap aturan yang sesuai dievaluasi dengan menggabungkan derajat keanggotaan variabel input yang terkait dengan aturan tersebut menggunakan operasi logika fuzzy (seperti minimum atau perkalian) untuk mendapatkan derajat keanggotaan keseluruhan dari setiap variabel output.
3. Defuzzyfikasi: Defuzzyfikasi adalah proses mengubah hasil inferensi fuzzy (variabel linguistik output dengan derajat keanggotaan) menjadi nilai numerik. Hal ini dilakukan dengan menggunakan metode seperti rata-rata tertimbang, metode centroid, atau metode lainnya untuk menentukan nilai numerik yang mewakili solusi dari sistem fuzzy.

Dengan demikian, fuzzy logic memungkinkan pengambilan keputusan yang lebih fleksibel dan dapat menangani ketidakpastian dalam lingkungan yang kompleks.

E. Fuzzy Systems

a. Struktur Fuzzy Systems: Fuzzy Systems adalah sistem yang menggunakan logika fuzzy untuk memodelkan dan mengatasi ketidakpastian dan keambiguan dalam suatu domain. Struktur dasar dari Fuzzy Systems terdiri dari tiga komponen utama: input, fuzzy logic, dan output. Input adalah variabel-variabel yang digunakan sebagai masukan ke dalam sistem, fuzzy logic adalah proses yang mengubah input menjadi output menggunakan aturan fuzzy, dan output adalah hasil yang dihasilkan oleh sistem.

b. Proses Pembentukan Fuzzy Systems:

1. Pengumpulan Data: Langkah pertama dalam pembentukan Fuzzy Systems adalah mengumpulkan data yang relevan dengan domain yang ingin dimodelkan. Data ini bisa berupa pengamatan, pengukuran, atau informasi yang diperoleh dari pakar domain.
2. Pembentukan Fuzzy Set: Setelah data terkumpul, langkah berikutnya adalah membentuk himpunan fuzzy (fuzzy set) untuk setiap variabel input dan output dalam sistem. Fuzzy set adalah himpunan nilai yang mewakili tingkat keanggotaan suatu elemen dalam himpunan tersebut. Fungsi keanggotaan digunakan untuk menggambarkan tingkat keanggotaan suatu elemen dalam himpunan fuzzy.
3. Penentuan Aturan Fuzzy: Setelah fuzzy set terbentuk, aturan fuzzy harus ditentukan. Aturan fuzzy menghubungkan kondisi-kondisi pada variabel input dengan aksi-aksi pada variabel output. Aturan ini dinyatakan dalam bentuk "jika kondisi A, maka aksi B". Aturan-aturan ini menggambarkan pemikiran dan pengetahuan yang tidak pasti secara linguistik.

c. Aplikasi Fuzzy Systems:

1. Kendali Fuzzy (Fuzzy Control): Fuzzy Systems dapat digunakan untuk mengendalikan sistem yang kompleks dengan menggabungkan pengetahuan manusia dan pengolahan data numerik. Contohnya adalah pengendalian suhu ruangan, kendaraan otomatis, atau sistem penjadwalan produksi.
2. Sistem Pendukung Keputusan Fuzzy (Fuzzy Decision Support System): Fuzzy Systems dapat membantu dalam pengambilan keputusan yang kompleks dan ambigu dengan mempertimbangkan berbagai faktor. Sistem ini dapat membantu dalam penilaian risiko, evaluasi kinerja, atau pengaturan inventaris.
3. Prediksi dan Pengenalan Pola dengan Fuzzy Logic: Fuzzy Systems dapat digunakan untuk prediksi dan pengenalan pola dalam data yang tidak pasti. Contohnya adalah prediksi cuaca, diagnosa medis, atau pengenalan pola dalam citra atau suara.

Dengan menggunakan Fuzzy Systems, kita dapat mengatasi ketidakpastian dan keambiguan dalam pemodelan dan pengambilan keputusan, serta memperoleh solusi yang lebih adaptif dan fleksibel dalam berbagai domain.

BAB V

PLANNING (TEKNIK DEKOMPOSISI MASALAH) - GOAL STACK PLANNING (GSP) - CONSTRAINT POSTING (CP)

Planning adalah proses sistematis untuk merencanakan langkah-langkah yang diperlukan untuk mencapai tujuan tertentu. Dalam domain kecerdasan buatan dan ilmu komputer, planning merujuk pada pemecahan masalah dengan menggunakan teknik dekomposisi masalah.

Teknik dekomposisi masalah adalah pendekatan untuk memecah masalah yang kompleks menjadi masalah yang lebih kecil dan lebih terkelola. Dua teknik dekomposisi masalah yang umum digunakan adalah Goal Stack Planning (GSP) dan Constraint Posting (CP).

1. Goal Stack Planning (GSP): Goal Stack Planning adalah teknik perencanaan yang berfokus pada hierarki tujuan. Dalam GSP, tujuan utama (goal) dipecah menjadi subtujuan yang lebih kecil dan lebih terkelola. Subtujuan ini ditempatkan dalam tumpukan tujuan (goal stack), yang mewakili urutan tindakan yang perlu dilakukan. Perencanaan dilakukan dengan mengatasi satu subtujuan pada satu waktu, dimulai dari tujuan teratas dalam tumpukan. Ketika subtujuan tercapai, mereka dikeluarkan dari tumpukan dan perencanaan dilanjutkan dengan subtujuan berikutnya. Dengan menggunakan GSP, perencana dapat mengatur urutan tindakan yang diperlukan untuk mencapai tujuan akhir.
2. Constraint Posting (CP): Constraint Posting adalah teknik perencanaan yang berfokus pada konstrain dan ketergantungan antara tindakan. Dalam CP, konstrain-konstrain yang ada dalam masalah diposting dan diperhatikan saat merencanakan tindakan. Konstrain-konstrain ini dapat berupa keterbatasan fisik, sumber daya yang terbatas, atau ketergantungan logis antara tindakan. Dengan mempertimbangkan konstrain-konstrain ini, perencana dapat memastikan bahwa tindakan yang direncanakan memenuhi semua ketergantungan dan keterbatasan yang ada. Constraint Posting membantu dalam mengurangi ruang solusi yang memungkinkan dan memfokuskan perencanaan pada solusi yang memenuhi semua konstrain.

Kedua teknik ini, yaitu GSP dan CP, dapat digunakan secara terpisah atau dikombinasikan dalam perencanaan. Pemilihan teknik tergantung pada karakteristik masalah yang dihadapi dan lingkungan yang ada. Tujuan utama dari teknik dekomposisi masalah adalah mempermudah proses perencanaan, mengurangi kompleksitas, dan memastikan pencapaian tujuan yang diinginkan. Dengan menggunakan teknik ini, perencanaan dapat dilakukan secara lebih efisien dan efektif.

BAB VI

DECISION TREE LEARNING

A. Konsep Dasar Tree Learning

a. Definisi pohon keputusan dan peran mereka dalam pengambilan keputusan

Pohon keputusan adalah model representasi grafis yang digunakan dalam Decision Tree Learning untuk pengambilan keputusan. Model ini terdiri dari simpul (node), cabang (branch), dan daun (leaf). Pohon keputusan secara visual membagi data menjadi bagian-bagian yang lebih kecil berdasarkan aturan keputusan yang ada pada setiap simpul. Pada akhirnya, pohon keputusan menghasilkan prediksi atau keputusan berdasarkan fitur atau atribut yang ada pada data.

b. Penjelasan tentang atribut, simpul, cabang, dan daun dalam pohon keputusan:

- **Atribut:** Atribut adalah karakteristik atau fitur yang digunakan untuk membagi data di dalam pohon keputusan. Contoh atribut bisa berupa usia, jenis kelamin, atau pendapatan.
- **Simpul:** Simpul adalah titik dalam pohon keputusan yang mewakili suatu atribut dan keputusan yang harus diambil berdasarkan atribut tersebut. Setiap simpul memiliki cabang-cabang yang menghubungkannya dengan simpul atau daun lain.
- **Cabang:** Cabang menghubungkan simpul-simpul dalam pohon keputusan. Setiap cabang menunjukkan nilai atau kategori tertentu dari atribut yang digunakan untuk membagi data.
- **Daun:** Daun adalah simpul akhir dalam pohon keputusan yang mewakili hasil keputusan atau prediksi. Misalnya, dalam pohon keputusan yang memprediksi apakah seseorang akan membeli sebuah produk, daun dapat mewakili kelas "Ya" atau "Tidak".

c. Gambaran tentang konsep entropi dan Information Gain sebagai metrik dalam pemilihan atribut:

- **Entropi:** Entropi adalah ukuran ketidakaturan atau keacakan dalam kumpulan data. Dalam konteks Decision Tree Learning, entropi digunakan untuk mengukur sejauh mana data di dalam simpul tersebar dalam kelas yang berbeda. Entropi memiliki rentang nilai antara 0 hingga 1, di mana 0 menunjukkan kejelasan maksimum (semua instance dalam simpul termasuk dalam satu kelas), sedangkan 1 menunjukkan kejelasan minimum (distribusi yang sama untuk setiap kelas).
- **Information Gain:** Information Gain adalah metrik yang digunakan untuk memilih atribut terbaik yang akan digunakan untuk membagi data dalam pembentukan pohon keputusan. Information Gain mengukur pengurangan entropi yang dihasilkan oleh pemilihan atribut tertentu. Atribut dengan Information Gain yang lebih tinggi dianggap lebih berguna dalam membagi data dan mempengaruhi keputusan.

B. ID3 (Iterative Dichotomiser 3)

- a. Pengenalan ID3 sebagai algoritma pembelajaran mesin untuk membangun pohon keputusan

ID3 (Iterative Dichotomiser 3) adalah salah satu algoritma pembelajaran mesin yang digunakan untuk membangun pohon keputusan. Algoritma ini menggunakan pendekatan top-down, dimulai dari simpul akar dan secara iteratif memilih atribut terbaik untuk membagi data berdasarkan metrik Information Gain. ID3 berfokus pada pembelajaran dari data yang dilabeli, di mana setiap instance memiliki atribut dan kelas yang sesuai. Tujuan ID3 adalah meminimalkan entropi atau ketidakpastian dalam setiap simpul pohon keputusan.

- b. Penjelasan langkah-langkah ID3, termasuk pemilihan atribut dengan Information Gain tertinggi dan pembuatan simpul serta cabang: Langkah-langkah dalam algoritma ID3 adalah sebagai berikut:

1. Memilih atribut terbaik:

Menghitung entropi awal dari data di simpul saat ini.

Menghitung Information Gain untuk setiap atribut yang belum digunakan.

Memilih atribut dengan Information Gain tertinggi sebagai atribut terbaik untuk membagi data.

Atribut terbaik harus memiliki Information Gain yang tinggi, menunjukkan bahwa atribut tersebut memberikan penurunan entropi yang signifikan.

2. Membuat simpul dan cabang:

Membuat simpul dengan atribut terbaik yang dipilih sebelumnya.

Membuat cabang dari simpul tersebut berdasarkan nilai-nilai unik atribut terbaik.

Setiap cabang mewakili nilai kategorikal dari atribut terbaik.

3. Memperoleh subset data:

Membagi data pada simpul saat ini menjadi subset berdasarkan nilai-nilai kategorikal dari atribut terbaik.

Setiap subset berisi data dengan nilai tertentu dari atribut terbaik.

4. Rekursif:

Mengulangi langkah-langkah di atas untuk setiap subset data.

Melakukan rekursi hingga mencapai kondisi berhenti, misalnya jika semua instance pada subset memiliki kelas yang sama atau tidak ada atribut lagi yang tersedia.

5. Menentukan daun:

Jika kondisi berhenti tercapai, maka simpul tersebut menjadi daun.

Daun mewakili hasil prediksi atau keputusan pada akhir pohon keputusan.

- c. Contoh implementasi ID3 dengan dataset sederhana: Misalkan kita memiliki dataset yang berisi informasi tentang cuaca (atribut) dan apakah seseorang akan bermain tenis (kelas). Berikut adalah contoh implementasi ID3 dengan dataset tersebut:

No.	Cuaca	Temperatur	Kelembaban	Angin	Bermain Tenis
1	Cerah	Panas	Tinggi	Tidak	Tidak
2	Cerah	Panas	Normal	Ya	Tidak
3	Berawan	Panas	Tinggi	Ya	Ya
4	Hujan	Sedang	Tinggi	Tidak	Ya
5	Hujan	Dingin	Normal	Tidak	Ya
6	Hujan	Dingin	Normal	Ya	Tidak
7	Berawan	Dingin	Normal	Tidak	Ya
8	Cerah	Sedang	Tinggi	Tidak	Tidak
9	Cerah	Dingin	Tinggi	Ya	Ya
10	Hujan	Sedang	Normal	Tidak	Ya

Dengan menggunakan ID3, kita dapat membangun pohon keputusan dengan langkah-langkah yang telah dijelaskan sebelumnya. Pada contoh ini, kita akan menggunakan atribut "Cuaca" sebagai atribut terbaik pertama.

Hasil pohon keputusan yang dihasilkan dengan implementasi ID3 pada dataset tersebut adalah sebagai berikut:

yamlCopy code

Cuaca

|— Berawan : Ya

|— Cerah

| |— Angin : Ya

- | └─ Angin : Tidak
- └─ Hujan
- |─ Kelembaban : Tidak
- └─ Kelembaban : Ya

Pohon keputusan tersebut memberikan aturan keputusan berdasarkan atribut "Cuaca" untuk memprediksi apakah seseorang akan bermain tenis.

C. C4.5

a. Penjelasan rinci tentang langkah-langkah C4.5, termasuk perhitungan Gain Ratio dan pruning: C4.5 adalah salah satu algoritma pembelajaran mesin yang digunakan untuk membangun pohon keputusan, serupa dengan ID3. Namun, C4.5 memiliki beberapa perbedaan dalam langkah-langkahnya. Berikut adalah langkah-langkah C4.5:

a) Memilih atribut terbaik:

- Menghitung Gain Ratio untuk setiap atribut yang belum digunakan.
- Gain Ratio mengukur peningkatan informasi relatif yang diberikan oleh atribut dalam membagi data.
- Atribut dengan Gain Ratio tertinggi dipilih sebagai atribut terbaik untuk membagi data.

b) Membuat simpul dan cabang:

- Membuat simpul dengan atribut terbaik yang dipilih sebelumnya.
- Membuat cabang dari simpul tersebut berdasarkan nilai-nilai unik atribut terbaik.
- Jika atribut terbaik adalah atribut kontinu, langkah tambahan diperlukan untuk menentukan titik pemisahan yang optimal.

c) Memperoleh subset data:

- Membagi data pada simpul saat ini menjadi subset berdasarkan nilai-nilai kategorikal dari atribut terbaik.
- Setiap subset berisi data dengan nilai tertentu dari atribut terbaik.

d) Rekursif:

- Mengulangi langkah-langkah di atas untuk setiap subset data.
- Melakukan rekursi hingga mencapai kondisi berhenti, misalnya jika semua instance pada subset memiliki kelas yang sama atau tidak ada atribut lagi yang tersedia.

e) Pruning (pemangkasan):

- Setelah membangun pohon keputusan, pruning dilakukan untuk mengurangi overfitting atau kelebihan penyesuaian pada data pelatihan.
- Pemangkasan melibatkan penghapusan simpul atau cabang yang tidak signifikan berdasarkan pengukuran statistik seperti uji chi-square atau validasi silang.

b. Perbandingan C4.5 dengan ID3 dalam hal kemampuan menangani data yang lebih kompleks: C4.5 memiliki beberapa keunggulan dibandingkan dengan ID3 dalam menangani data yang lebih kompleks:

- C4.5 dapat mengatasi atribut kontinu. ID3 hanya dapat handle atribut diskrit, sedangkan C4.5 dapat melakukan pemilihan atribut dengan menggunakan nilai-nilai ambang pada atribut kontinu.
- C4.5 menggunakan Gain Ratio sebagai metrik pemilihan atribut, sedangkan ID3 menggunakan Information Gain. Gain Ratio memperhitungkan faktor penyebaran yang lebih merata pada atribut dengan banyak nilai atau kategori.
- C4.5 menggunakan pruning untuk mengurangi overfitting pada pohon keputusan, sementara ID3 tidak melibatkan langkah ini.
- Dalam hal efisiensi, ID3 cenderung lebih cepat karena C4.5 melibatkan perhitungan yang lebih kompleks seperti Gain Ratio.

c. Contoh implementasi C4.5 dengan dataset yang mencakup atribut kontinu dan diskrit: Misalkan kita memiliki dataset yang mencakup atribut seperti usia (kontinu), pendapatan (kontinu), jenis kelamin (diskrit), dan apakah seseorang memiliki mobil (diskrit). Berikut adalah contoh implementasi C4.5 dengan dataset tersebut:

No.	Usia	Pendapatan	Jenis Kelamin	Mobil
1	23	2500	Pria	Tidak
2	45	3500	Wanita	Ya
3	30	5000	Pria	Ya
4	18	1800	Wanita	Tidak
5	35	4000	Pria	Ya
6	50	6000	Wanita	Ya
7	28	3200	Wanita	Tidak
8	42	4800	Pria	Ya
9	38	5500	Pria	Ya

No.	Usia	Pendapatan	Jenis Kelamin	Mobil
10	29	3200	Wanita	Tidak

Dengan menggunakan C4.5, kita dapat membangun pohon keputusan dengan langkah-langkah yang telah dijelaskan sebelumnya. Pada contoh ini, kita akan menggunakan atribut "Mobil" sebagai atribut target.

Hasil pohon keputusan yang dihasilkan dengan implementasi C4.5 pada dataset tersebut adalah sebagai berikut:

Usia



Pohon keputusan tersebut memberikan aturan keputusan berdasarkan atribut "Usia", "Pendapatan", dan "Jenis Kelamin" untuk memprediksi apakah seseorang memiliki mobil.

BAB VII

JARINGAN SYARAF TIRUAN (JST)

A. Konsep Jaringan Syaraf Tiruan (Jst)

Konsep dasar Jaringan Syaraf Tiruan (JST) adalah model matematika yang terinspirasi oleh struktur dan fungsi jaringan syaraf biologis dalam otak manusia. JST terdiri dari kumpulan unit pemrosesan sederhana yang disebut neuron, yang bekerja secara kolektif untuk memproses informasi.

JST beroperasi dengan mengirimkan sinyal melalui jaringan neuron yang terhubung. Setiap neuron menerima input dari neuron sebelumnya, melakukan komputasi pada input tersebut, dan menghasilkan output yang kemudian dikirimkan ke neuron-neuron berikutnya. Proses ini mirip dengan aliran informasi yang terjadi dalam jaringan syaraf biologis.

Neuron dalam JST memiliki bobot dan bias yang dapat diubah selama proses pembelajaran. Bobot mengatur kekuatan koneksi sinaptik antara neuron, sedangkan bias mempengaruhi ambang aktivasi neuron. Selama proses pembelajaran, JST memperbarui bobot dan biasnya berdasarkan informasi yang diberikan oleh data latih.

Konsep dasar JST melibatkan pembentukan hubungan antara input dan output melalui serangkaian neuron yang terhubung. Proses pembelajaran JST melibatkan pengoptimalan bobot dan bias untuk mencapai kemampuan prediksi atau klasifikasi yang diinginkan. JST telah digunakan dalam berbagai aplikasi seperti pengenalan pola, pemrosesan bahasa alami, pengenalan suara, dan pengenalan citra.

B. Model Syaraf Toruan

Model Syaraf Tiruan (JST) atau Neuron dalam JST adalah unit pemrosesan dasar yang mensimulasikan fungsi neuron biologis dalam otak manusia. Model ini memiliki struktur yang terdiri dari beberapa komponen penting, yaitu:

1. Input: Neuron menerima input dari neuron-neuron sebelumnya atau dari input eksternal. Input ini bisa berupa nilai numerik atau fitur dari data yang sedang diproses.
2. Bobot (Weights): Setiap input yang diterima oleh neuron memiliki bobot yang menggambarkan kekuatan koneksi sinaptik antara neuron tersebut dengan neuron-neuron sebelumnya. Bobot ini mengontrol sejauh mana input akan mempengaruhi aktivitas neuron.
3. Fungsi Agregasi: Neuron menggabungkan semua input yang diterima dengan bobot yang sesuai. Fungsi agregasi dapat berupa penjumlahan sederhana dari hasil perkalian input dengan bobot atau operasi matematika lain yang lebih kompleks.
4. Fungsi Aktivasi: Setelah agregasi input, neuron menerapkan fungsi aktivasi pada hasil agregasi tersebut. Fungsi aktivasi mengubah hasil agregasi menjadi nilai output yang memiliki arti interpretasi. Beberapa contoh fungsi aktivasi yang umum digunakan adalah sigmoid, ReLU (Rectified Linear Unit), dan tangen hiperbolik.

5. Bias: Selain bobot, neuron juga memiliki bias yang merupakan parameter tambahan yang mempengaruhi keputusan aktivasi. Bias memungkinkan neuron untuk memiliki tingkat ambang tertentu sebelum menghasilkan output.
6. Output: Neuron menghasilkan output berdasarkan hasil dari fungsi aktivasi. Output ini kemudian akan diteruskan sebagai input kepada neuron-neuron berikutnya dalam jaringan.

Model syaraf tiruan ini dirancang untuk bekerja secara paralel dan terdistribusi, mirip dengan cara kerja jaringan syaraf biologis. Dengan menggabungkan banyak neuron dalam jaringan, JST dapat mempelajari pola-pola yang kompleks dan melakukan tugas seperti klasifikasi, prediksi, dan pemrosesan data lainnya.

C. Aktivasi Dan Arsitektur Jaringan Pada Jst

1. Aktivasi: Aktivasi dalam JST merujuk pada proses transformasi input menjadi output menggunakan fungsi aktivasi pada setiap neuron. Fungsi aktivasi mengontrol tingkat aktivasi neuron dan memberikan respons terhadap input yang diterima.

Beberapa fungsi aktivasi yang umum digunakan dalam JST adalah:

- Sigmoid: Fungsi sigmoid menghasilkan output dalam rentang antara 0 dan 1. Fungsi ini sering digunakan di lapisan tersembunyi dan lapisan output pada JST.
- ReLU (Rectified Linear Unit): Fungsi ReLU menghasilkan nilai 0 untuk input negatif dan nilainya sendiri untuk input positif. Fungsi ini efektif dalam mengatasi masalah vanish gradient.
- Tangen Hiperbolik: Fungsi tangen hiperbolik menghasilkan output dalam rentang antara -1 dan 1. Fungsi ini sering digunakan di lapisan tersembunyi dalam JST.
- Softmax: Fungsi softmax digunakan khusus untuk lapisan output dalam kasus klasifikasi multikelas. Fungsi ini menghasilkan probabilitas untuk setiap kelas sehingga total probabilitas dari semua kelas adalah 1.

Fungsi aktivasi yang tepat harus dipilih tergantung pada jenis masalah dan karakteristik data yang sedang diproses.

2. Arsitektur Jaringan: Arsitektur JST merujuk pada struktur atau tata letak jaringan neuron dalam JST. Arsitektur ini dapat bervariasi tergantung pada masalah yang ingin diselesaikan dan karakteristik data yang dihadapi. Beberapa arsitektur JST yang umum digunakan adalah:
 - Feedforward Neural Network (FNN): Merupakan arsitektur JST paling dasar di mana informasi mengalir maju melalui jaringan dari lapisan input ke lapisan output. Tidak ada siklus atau koneksi mundur dalam FNN.

- Convolutional Neural Network (CNN): Arsitektur ini dirancang khusus untuk memproses data berupa gambar atau data grid 2D. CNN memiliki lapisan konvolusi yang efektif dalam mengekstraksi fitur-fitur spasial dari data gambar.
- Recurrent Neural Network (RNN): Arsitektur ini cocok untuk memproses data sekuensial seperti teks atau waktu. RNN memiliki koneksi mundur yang memungkinkan informasi dari masa lalu dipertahankan dalam memori jangka pendek dan digunakan dalam pemrosesan saat ini.
- Long Short-Term Memory (LSTM): LSTM adalah varian dari RNN yang mampu mengatasi masalah gradien yang menghilang atau meledak dengan menggunakan sel memori khusus untuk menyimpan informasi jangka panjang.

Pemilihan arsitektur yang tepat tergantung pada jenis masalah dan sifat data yang sedang diproses. Arsitektur yang lebih kompleks dapat menangani masalah yang lebih rumit, tetapi juga memerlukan lebih banyak komputasi dan data pelatihan yang lebih besar.

D. Supervised Learning Danunsupervised Learning

1. Supervised Learning: Supervised learning adalah paradigma pembelajaran di mana JST menggunakan data latih yang sudah dilabeli (memiliki pasangan input-output yang benar) untuk melakukan pembelajaran. Tujuan dari supervised learning adalah untuk mempelajari hubungan antara input dan output yang diberikan sehingga JST dapat menghasilkan output yang akurat untuk input baru yang belum pernah dilihat sebelumnya.

Dalam supervised learning, JST akan mengubah bobot dan biasnya dengan menggunakan algoritma pembelajaran seperti backpropagation. Algoritma ini melakukan propagasi mundur dari output ke lapisan tersembunyi untuk menghitung gradien kesalahan, dan kemudian menggunakan gradien tersebut untuk memperbarui bobot dan bias sehingga kesalahan dapat diminimalkan.

Contoh dari aplikasi supervised learning dalam JST adalah klasifikasi gambar, di mana JST diajarkan untuk mengenali objek dalam gambar berdasarkan label yang ada.

2. Unsupervised Learning: Unsupervised learning adalah paradigma pembelajaran di mana JST tidak memiliki data latih yang dilabeli. Sebaliknya, JST diberikan data yang tidak memiliki label dan ditugaskan untuk menemukan pola atau struktur yang tersembunyi dalam data tersebut.

Dalam unsupervised learning, JST dapat menggunakan algoritma seperti k-means clustering atau self-organizing maps (SOM) untuk mengelompokkan data ke dalam kelompok atau menghasilkan representasi data yang lebih ringkas.

Contoh dari aplikasi unsupervised learning dalam JST adalah analisis klastering data pelanggan, di mana JST mencoba mengelompokkan pelanggan berdasarkan perilaku pembelian mereka tanpa adanya label kategori tertentu.

Baik supervised learning maupun unsupervised learning memiliki peran penting dalam pembelajaran JST. Supervised learning berguna dalam mempelajari hubungan input-output yang ada, sementara unsupervised learning membantu mengungkap pola atau struktur dalam data yang belum diketahui.

BAB VIII

ALGORITMA GENETIKA (AG)

Algoritma Genetika (AG) adalah metode optimisasi yang terinspirasi oleh teori evolusi dan mekanisme genetik dalam alam. AG digunakan untuk mencari solusi terbaik atau mendekati solusi terbaik dalam ruang pencarian yang kompleks.

Komponen-komponen utama dalam Algoritma Genetika adalah sebagai berikut:

1. **Populasi:** Populasi dalam AG terdiri dari sekelompok individu atau kromosom yang merepresentasikan solusi dalam ruang pencarian. Setiap individu dianggap sebagai kandidat solusi yang potensial.
2. **Individu atau Kromosom:** Individu atau kromosom dalam AG merepresentasikan solusi dalam bentuk struktur data yang sesuai dengan masalah yang sedang diselesaikan. Misalnya, dalam kasus masalah optimisasi numerik, individu bisa berupa vektor angka, sedangkan dalam kasus masalah penggantian jadwal, individu bisa berupa urutan tugas atau aktivitas.
3. **Gen:** Gen adalah bagian terkecil dari kromosom yang membawa informasi genetik. Gen dapat berupa nilai numerik, huruf, atau simbol lainnya yang merepresentasikan sifat atau karakteristik individu dalam solusi.
4. **Fungsi Fitness:** Fungsi fitness adalah metrik yang digunakan untuk mengevaluasi sejauh mana sebuah individu cocok atau baik dalam menghadapi masalah yang sedang diselesaikan. Fungsi fitness memberikan penilaian numerik berdasarkan kriteria yang ditentukan, dan biasanya bertujuan untuk memaksimalkan atau meminimalkan nilai tersebut.
5. **Seleksi:** Seleksi adalah proses pemilihan individu yang akan menjadi orangtua untuk reproduksi berikutnya. Individu yang memiliki nilai fitness yang lebih tinggi cenderung memiliki peluang yang lebih besar untuk dipilih.
6. **Crossover (Persilangan):** Crossover adalah proses di mana dua individu orangtua dipadukan untuk menghasilkan individu keturunan baru. Proses ini dilakukan dengan menukar bagian-bagian gen antara kedua individu dengan harapan menghasilkan kombinasi gen yang menguntungkan.
7. **Mutasi:** Mutasi adalah proses di mana terjadi perubahan acak pada gen dalam individu keturunan. Proses mutasi memungkinkan variasi genetik baru yang tidak ada dalam populasi awal. Meskipun perubahan ini bersifat acak, mutasi dapat memperkenalkan solusi baru yang berpotensi lebih baik.

Komponen-komponen tersebut bekerja bersama-sama dalam AG untuk melakukan pencarian solusi secara iteratif dengan mengaplikasikan konsep seleksi, crossover, dan mutasi. Proses ini berulang hingga kriteria penghentian terpenuhi atau solusi yang memadai ditemukan.

DAFTAR PUSTAKA:

- Russell, S., Norvig, P. (2016). Artificial Intelligence: A Modern Approach (3rd ed.). Pearson.
- Goodfellow, I., Bengio, Y., Courville, A. (2016). Deep Learning. MIT Press.
- Jurafsky, D., Martin, J.H. (2019). Speech and Language Processing (3rd ed.). Pearson.
- Szeliski, R. (2010). Computer Vision: Algorithms and Applications. Springer.
- Sutton, R. S., Barto, A. G. (2018). Reinforcement Learning: An Introduction (2nd ed.). MIT Press.
- Goldberg, D. E. (1989). Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley.
- Russell, S., Norvig, P. (2016). Artificial Intelligence: A Modern Approach (3rd ed.). Pearson.
- Nilsson, N. J. (1998). Artificial Intelligence: A New Synthesis. Morgan Kaufmann.
- Hayes-Roth, B., Waterman, D. A., & Lenat, D. B. (1983). Building Expert Systems. Addison-Wesley.
- Luger, G. F. (2016). Artificial Intelligence: Structures and Strategies for Complex Problem Solving (6th ed.). Pearson.
- Winston, P. H. (1992). Artificial Intelligence (3rd ed.). Addison-Wesley.
- Russell, S., Norvig, P. (2016). Artificial Intelligence: A Modern Approach (3rd ed.). Pearson.
- Nilsson, N. J. (1998). Artificial Intelligence: A New Synthesis. Morgan Kaufmann.
- Luger, G. F. (2016). Artificial Intelligence: Structures and Strategies for Complex Problem Solving (6th ed.). Pearson.
- Poole, D., Mackworth, A., & Goebel, R. (2017). Computational Intelligence: A Logical Approach. Oxford University Press.
- Hayes-Roth, B., Waterman, D. A., & Lenat, D. B. (1983). Building Expert