

TUGAS AKHIR
“PRAKTIKUM ARTIFICIAL INTELLIGENCE”



Disusun Oleh :

Nama : Muhammad Razief
NIM : 21346015
Prodi : Informatika (NK)
Dosen Pengampu : Widya Darwin, S.Pd., M.Pd.T

PROGRAM STUDI INFORMATIKA
JURUSAN TEKNIK ELEKTRONIKA
FAKULTAS TEKNIK
UNIVERSITAS NEGERI PADANG
2023

KATA PENGANTAR

Rasa syukur kita hanya milik Allah SWT atas segala semua rahmatnya, sehingga saya dapat menyelesaikan tugas akhir yang saya susun ini. Meskipun banyak rintangan dan hambatan yang saya alami dalam proses pekerjaan tetapi saya berhasil mengerjakan dengan baik dan tetap pada waktunya.

Dan harapan saya di sini semoga atas tugas akhir yang saya buat ini bisa menambah pengetahuan dan pengalaman bagi para pembaca, dan untuk kedepan nya kiat juga sama-sama memperbaiki bentuk atau menambah isi dari makalah agar semua akan lebih baik dengan sebelumnya.

Saya mengucapkan terima kasih kepada Ibu Widya Darwin, S.Pd., M.Pd.T sebagai Dosen Pengampu pada mata kuliah Pratikum artificial intellegence yang telah membimbing saya dalam menyelesaikan tugas akhir yang saya buat ini.

Karena dari semua keterbatasan dari pengetahuan atau pun pengalaman saya, saya yakin masih banyak sekali dari kekurangan yang terdapat pada makalah ini. Oleh karena itu saya sangat berharap untuk saran dan kritik yang bisa membangun dari pembaca demi semua tugas akhir ini akan terselesaikan dengan benar.

Painan, Juni 2023

Muhammad Razief

DAFTAR ISI

KATA PENGANTAR	i
DAFTAR ISI.....	ii
BAB I PENDAHULUAN.....	1
A. Defenisi Kecerdasan Buatan	1
B. Teknik Penyelesaian Kasus dalam Kecerdasan Buatan	1
BAB II RUANG MASALAH & SISTEM PRODUKSI	3
A. Ruang Masalah.....	3
B. Sistem Produksi.....	4
BAB II METODE-METODE PENCARIAN DALAM KECERDASAN BUATAN.....	6
A. Pencarian Buta.....	6
B. Pencarian Heuristik	7
BAB IV REASONING	8
A. Propotional Logic	8
B. First Order Logic (Predicate Calculus)	9
C. Fuzzy Logic.....	10
BAB V PLANNING(TEKNIK DEKOMPOSISI MASALAH).....	14
A. Goal Stack Planning (GSP).....	14
B. Constraint Posting(CP).....	14
BAB VI LEARNING.....	16
A. ID3.....	16
B. C.45	16
BAB VII JARINGAN SYARAF TIRUAN (JST).....	18
A. Konsep Dasar JST	18
B. Model Syaraf Tiruan (Neuron).....	19
C. Aktivasi & Arsitektur Jaringan pada JST.....	20
D. Supervised Learning & Unsupervised Learning	21
BAB VIII ALGORITMA GENETIKA (AG).....	22
A. Definisi algoritma genetika.....	22
B. Komponen-komponen AG.....	22

BAB IX COGNITIF SCIENCE.....	24
A. Defenisi.....	24
DAFTAR PUSTAKA.....	25

BAB I

PENDAHULUAN

A. Defenisi Kecerdasan Buatan

Kecerdasan buatan adalah bidang studi dalam ilmu komputer yang bertujuan untuk mengembangkan sistem atau mesin yang mampu melakukan tugas-tugas yang biasanya memerlukan kecerdasan manusia. Tujuan utama dari kecerdasan buatan adalah menciptakan mesin yang dapat berpikir, belajar, dan mengambil keputusan seperti manusia.

Definisi kecerdasan buatan telah berkembang seiring dengan perkembangan teknologi dan pemahaman tentang bagaimana komputer dan sistem cerdas dapat bekerja. Pada dasarnya, kecerdasan buatan melibatkan pengembangan algoritma dan model matematika yang memungkinkan komputer untuk menganalisis data, memahami konteks, mengambil keputusan, dan melakukan tugas-tugas yang kompleks.

Kecerdasan buatan dapat diterapkan dalam berbagai bidang, termasuk pemrosesan bahasa alami, visi komputer, pengenalan suara, robotika, sistem ekspert, kendaraan otonom, dan banyak lagi. Metode yang digunakan dalam kecerdasan buatan termasuk logika, statistik, pengenalan pola, jaringan saraf tiruan, pemrosesan bahasa alami, dan algoritma pembelajaran mesin.

B. Teknik pencarian

Dalam kecerdasan buatan, terdapat beberapa teknik atau metode yang digunakan untuk menyelesaikan kasus atau masalah. Berikut adalah beberapa teknik penyelesaian kasus yang umum digunakan dalam kecerdasan buatan:

→Pencarian Lurus (Breadth-First Search, BFS): Pencarian ini mengeksplorasi simpul-simpul dalam graf secara berlapis, mulai dari simpul awal dan meluas ke simpul-simpul tetangga sebelum melanjutkan ke tingkat yang lebih dalam. Metode ini memastikan pencarian dilakukan secara sistematis dan menemukan solusi terpendek jika ada.

→Pencarian Dalam (Depth-First Search, DFS): Pencarian ini mengeksplorasi simpul-simpul dalam graf secara vertikal, yaitu dengan melanjutkan ke simpul anak pertama sebelum kembali ke simpul orang tua dan melanjutkan ke simpul anak berikutnya. Metode ini cenderung meluas terlebih dahulu sebelum mendalam, dan dapat digunakan ketika solusi yang dalam atau panjang lebih diutamakan daripada solusi terpendek.

→Pencarian Terinformasi (Informed Search): Pencarian ini menggunakan informasi heuristik untuk memandu pencarian dan memilih langkah-langkah yang lebih mungkin menuju solusi. Salah satu contoh metode pencarian terinformasi adalah Pencarian A* (A-star), yang menggabungkan biaya sejauh ini (cost-so-far) dan estimasi biaya sisa (heuristik) untuk mengevaluasi simpul-simpul dalam pencarian.

→ Pencarian Garis Tepi (Hill Climbing): Metode ini mencoba meningkatkan solusi saat ini secara iteratif dengan memilih langkah yang meningkatkan nilai fungsi tujuan. Namun, pencarian ini dapat terjebak pada minimum lokal jika tidak ada langkah yang dapat meningkatkan solusi saat ini.

→ Algoritma Genetika (Genetic Algorithms): Metode ini terinspirasi oleh teori evolusi biologi dan menggunakan konsep seperti seleksi alami, reproduksi, dan mutasi dalam mencari solusi. Populasi solusi kandidat dievaluasi, dan solusi-solusi yang lebih baik memiliki probabilitas reproduksi yang lebih tinggi, sehingga menciptakan generasi-generasi baru dengan perbaikan bertahap.

→ Pencarian Beam (Beam Search): Metode ini membatasi jumlah simpul yang dieksplorasi pada setiap langkah pencarian. Pada setiap tingkat, hanya beberapa simpul dengan skor terbaik yang dipertahankan, sementara yang lainnya dibuang. Pendekatan ini memungkinkan eksplorasi yang lebih cepat dan pengurangan kompleksitas pencarian.

→ Pencarian Monte Carlo (Monte Carlo Search): Pendekatan ini menggunakan acak dan simulasi untuk mencari solusi dalam ruang pencarian. Solusi dihasilkan dengan melakukan serangkaian percobaan acak dan evaluasi, dan kemudian digunakan untuk memperbaiki pencarian di iterasi selanjutnya.

BAB II

RUANG MASALAH & SISTEM PRODUKSI

A. Ruang Masalah

Dalam kecerdasan buatan (Artificial Intelligence), ruang masalah merujuk pada kumpulan semua kemungkinan solusi atau keadaan yang relevan untuk suatu masalah yang ingin dipecahkan oleh sistem AI. Ruang masalah mencakup semua variabel, konstrain, dan kemungkinan keadaan yang relevan yang harus dipertimbangkan saat mencari solusi.

Ruang masalah dapat bervariasi tergantung pada jenis masalah yang ingin diselesaikan. Berikut adalah beberapa contoh ruang masalah dalam AI:

1. Ruang Masalah Pencarian: Ruang masalah pencarian berkaitan dengan mencari solusi optimal atau pendekatan terbaik melalui ruang kemungkinan keadaan. Contohnya adalah masalah jalan terpendek dalam peta, di mana ruang masalah mencakup semua rute yang mungkin antara dua titik.
2. Ruang Masalah Klasifikasi: Ruang masalah klasifikasi melibatkan pemisahan objek atau data ke dalam kelas-kelas yang berbeda berdasarkan atribut-atribut tertentu. Ruang masalah mencakup semua kemungkinan kombinasi atribut dan label kelas yang relevan.
3. Ruang Masalah Regresi: Ruang masalah regresi terkait dengan memodelkan hubungan antara variabel input dan output kontinu. Ruang masalah mencakup semua kemungkinan kombinasi nilai-nilai input dan output yang relevan.
4. Ruang Masalah Pengenalan Pola: Ruang masalah pengenalan pola melibatkan pengenalan atau klasifikasi pola yang kompleks dalam data. Ruang masalah mencakup semua kemungkinan kombinasi pola dan atribut yang relevan dalam data.
5. Ruang Masalah Perencanaan: Ruang masalah perencanaan berkaitan dengan mencari urutan tindakan yang optimal untuk mencapai tujuan tertentu. Ruang masalah mencakup semua kemungkinan urutan tindakan yang mungkin dan keadaan yang relevan selama perencanaan.
6. Ruang Masalah Pemrosesan Bahasa Alami: Ruang masalah dalam pemrosesan bahasa alami mencakup semua kemungkinan konstruksi kalimat, pola kata, dan makna yang relevan dalam bahasa manusia.

Selain itu, ruang masalah juga dapat bergantung pada faktor-faktor seperti dimensi data, kompleksitas, dan jenis informasi yang diperlukan. Memahami dan memodelkan

ruang masalah yang tepat adalah langkah penting dalam merancang dan mengimplementasikan solusi AI yang efektif.

B. Sistem Produksi

Sistem produksi adalah salah satu paradigma yang digunakan dalam bidang kecerdasan buatan (Artificial Intelligence) untuk memodelkan pengetahuan dan perilaku berbasis aturan. Sistem produksi terdiri dari aturan-aturan produksi yang terdiri dari kondisi dan aksi, yang digunakan untuk mengambil keputusan atau menghasilkan keluaran berdasarkan kondisi yang ada.

Sebuah sistem produksi terdiri dari beberapa komponen utama:

1. **Basis Pengetahuan (Knowledge Base):** Basis pengetahuan berisi aturan-aturan produksi yang mewakili pengetahuan dan kebijakan domain yang ingin diterapkan. Aturan-aturan ini terdiri dari kondisi (juga dikenal sebagai klausul IF) yang harus terpenuhi dan tindakan (juga dikenal sebagai klausul THEN) yang harus diambil jika kondisi tersebut terpenuhi.
2. **Mesin Inferensi (Inference Engine):** Mesin inferensi bertanggung jawab untuk menerapkan aturan-aturan produksi dalam basis pengetahuan. Ini adalah komponen yang mengevaluasi kondisi-kondisi dalam aturan-aturan produksi dan mengambil tindakan yang sesuai berdasarkan fakta-fakta yang ada.
3. **Basis Fakta (Fact Base):** Basis fakta berisi informasi aktual yang ada pada suatu saat dalam sistem. Ini mencerminkan keadaan atau situasi yang sedang diamati dan dapat berubah seiring waktu. Basis fakta ini digunakan oleh mesin inferensi untuk mengevaluasi kondisi-kondisi dalam aturan-aturan produksi.
4. **Antarmuka Pengguna:** Antarmuka pengguna memungkinkan interaksi antara pengguna dan sistem produksi. Pengguna dapat memasukkan input, mengubah basis pengetahuan atau basis fakta, dan melihat output atau hasil dari sistem produksi.

Proses kerja sistem produksi adalah sebagai berikut:

1. **Evaluasi Kondisi:** Mesin inferensi mengevaluasi kondisi-kondisi dalam aturan-aturan produksi dengan memeriksa kecocokan dengan basis fakta yang ada. Aturan-aturan produksi yang memenuhi kondisi akan diaktifkan.
2. **Pemilihan Aksi:** Setelah aturan-aturan produksi yang teraktifkan ditentukan, mesin inferensi memilih tindakan atau aksi yang sesuai untuk dijalankan berdasarkan kondisi-kondisi yang terpenuhi.
3. **Eksekusi Aksi:** Tindakan yang dipilih kemudian dieksekusi, menghasilkan perubahan pada basis fakta atau menghasilkan output yang diinginkan.

Sistem produksi dapat digunakan dalam berbagai aplikasi, seperti sistem pakar, sistem diagnosis, sistem rekomendasi, atau sistem kontrol. Mereka memungkinkan representasi pengetahuan yang berbasis aturan dan kemampuan untuk membuat keputusan atau menghasilkan keluaran berdasarkan kondisi yang ada.

BAB III

METODE-METODE PENCARIAN DALAM KECERDASAN BUATAN

A. Pencarian Buta

metode pencarian buta atau tak terinformasi sering digunakan untuk mencari solusi atau jalur optimal dalam ruang pencarian yang kompleks. Berikut ini beberapa metode pencarian buta yang umum digunakan dalam AI:

- **Pencarian Breadth-First (Pencarian Lebar):** Metode ini mengeksplorasi simpul-simpul secara berurutan pada setiap level dalam grafik masalah. Ini berarti semua simpul pada level saat ini akan diperiksa sebelum pindah ke level berikutnya. Metode pencarian lebar menggunakan struktur data antrian (queue) untuk menyimpan simpul-simpul yang akan dieksplorasi selanjutnya.
- **Pencarian Depth-First (Pencarian Dalam):** Metode ini mengeksplorasi simpul-simpul secara vertikal, mengikuti cabang-cabang yang mungkin sampai tidak ada simpul lagi yang dapat dieksplorasi. Jika tidak ada lagi simpul yang tersedia, metode pencarian dalam akan kembali ke simpul sebelumnya dan melanjutkan pencarian dari simpul lain yang belum dieksplorasi. Metode pencarian dalam menggunakan struktur data tumpukan (stack) untuk melacak simpul-simpul yang akan dieksplorasi berikutnya.
- **Pencarian Iterative Deepening (Pencarian Dalam dengan Meningkatkan Kedalaman):** Metode ini menggabungkan pendekatan pencarian lebar dan dalam dengan melakukan pencarian dalam pada kedalaman terbatas pada setiap iterasi. Pencarian dilakukan pada kedalaman 1, kemudian 2, 3, dan seterusnya hingga solusi ditemukan. Metode ini memungkinkan pencarian dalam dengan overhead yang lebih rendah dibandingkan dengan pencarian dalam biasa.
- **Pencarian Bidirectional (Pencarian Dwi-Arah):** Metode ini melibatkan pencarian dari titik awal dan titik tujuan secara bersamaan, dengan mencoba untuk menemukan simpul yang sama di kedua arah. Pencarian dimulai dari kedua titik, dan ketika simpul yang sama ditemukan, solusi dianggap ditemukan. Metode ini dapat mengurangi kompleksitas pencarian pada beberapa kasus.
- **Pencarian Uniform-Cost (Pencarian dengan Biaya Seragam):** Metode ini mempertimbangkan biaya pergerakan antara simpul-simpul dalam ruang pencarian. Ini memilih jalur dengan biaya terendah pada setiap langkah pencarian. Metode pencarian uniform-cost menggunakan struktur data antrian prioritas (priority queue) untuk memprioritaskan simpul-simpul berdasarkan biaya.

B. Pencarian Heuristik

Metode pencarian heuristik menggunakan informasi tambahan yang disebut heuristik untuk membimbing proses pencarian. Heuristik adalah aturan praktis atau strategi yang dirancang untuk mempercepat proses pencarian. Beberapa metode pencarian heuristik yang umum digunakan adalah sebagai berikut:

a. Pencarian Hill Climbing (Pencarian Naik Bukit):

Metode ini mencoba untuk mencari solusi dengan melakukan gerakan yang mengarah ke peningkatan nilai fungsi heuristik. Namun, metode ini dapat terjebak pada optimum lokal jika tidak ada jalan keluar yang lebih baik.

b. Pencarian Best-First (Pencarian Terbaik Pertama):

Metode ini memilih simpul yang paling menjanjikan berdasarkan fungsi evaluasi atau heuristik tertentu. Simpul-simpul yang memiliki nilai heuristik terbaik dieksplorasi terlebih dahulu.

c. Pencarian A* (A-Star):

Metode ini menggabungkan konsep pencarian terbaik pertama (best-first search) dengan fungsi evaluasi heuristik. Fungsi evaluasi A* adalah kombinasi dari biaya sejauh ini dan perkiraan biaya yang tersisa untuk mencapai tujuan. Metode ini memastikan solusi optimal jika fungsi heuristik memenuhi sifat-sifat tertentu.

Metode pencarian heuristik cenderung lebih efisien daripada metode pencarian buta karena penggunaan informasi tambahan. Namun, metode heuristik juga dapat memberikan solusi suboptimal jika heuristiknya tidak memadai atau jika terjebak pada optimum lokal. Pilihan metode pencarian tergantung pada sifat masalah dan ketersediaan informasi. digunakan dengan masalah yang sedang diselesaikan. Heuristik yang buruk atau tidak relevan dapat menghasilkan solusi yang tidak optimal atau bahkan tidak dapat ditemukan.

BAB IV REASONING

A. Propotional Logic

Logika proposional, juga dikenal sebagai logika proposisi atau logika kalimat, adalah cabang logika matematika yang mempelajari hubungan logis antara proposisi atau pernyataan. Logika proposional berfokus pada pengolahan simbol-simbol logis, di mana proposisi dinyatakan dengan menggunakan simbol-simbol logis dan dihubungkan melalui operator logis.

- **Negasi**

Negasi (NOT) digunakan untuk membalik nilai kebenaran sebuah proposisi. Dalam logika proposional, negasi dinyatakan dengan menggunakan simbol '~' atau '¬'. Jika p adalah sebuah proposisi, maka negasi dari p ditulis sebagai $\sim p$ atau $\neg p$. Jika p bernilai True, maka $\sim p$ atau $\neg p$ akan bernilai False, dan sebaliknya.

Contoh: Jika p adalah proposisi "Saya sedang belajar", maka negasi dari p adalah $\sim p$ atau $\neg p$ yang berarti "Saya tidak sedang belajar".

- **Konjungsi**

Konjungsi (AND) digunakan untuk menghubungkan dua proposisi dan menghasilkan proposisi baru yang bernilai True hanya jika kedua proposisi yang terhubung juga bernilai True. Dalam logika proposional, konjungsi dinyatakan dengan menggunakan simbol '^' atau 'A'.

Contoh: Jika p adalah proposisi "Saya sedang belajar" dan q adalah proposisi "Saya sedang membaca", maka $p \wedge q$ adalah konjungsi dari p dan q yang berarti "Saya sedang belajar dan saya sedang membaca".

- **Disjungsi**

Disjungsi (OR): Disjungsi digunakan untuk menghubungkan dua proposisi dan menghasilkan proposisi baru yang bernilai True jika salah satu atau kedua proposisi yang terhubung bernilai True. Dalam logika proposional, disjungsi dinyatakan dengan menggunakan simbol 'v' atau 'V'.

Contoh: Jika p adalah proposisi "Saya sedang belajar" dan q adalah proposisi "Saya sedang membaca", maka $p \vee q$ adalah disjungsi dari p dan q yang berarti "Saya sedang belajar atau saya sedang membaca".

- **Implikasi**

Implikasi (Imply): Implikasi menghubungkan dua proposisi dan menghasilkan proposisi baru yang menyatakan bahwa jika proposisi pertama bernilai True, maka proposisi kedua juga bernilai True. Namun, jika proposisi pertama bernilai False, maka tidak ada konsekuensi yang dapat ditarik tentang nilai proposisi kedua. Dalam logika proposional, implikasi dinyatakan dengan menggunakan simbol ' \rightarrow '.

Contoh: Jika p adalah proposisi "Jika hujan, maka jalanan basah", dan q adalah proposisi "Jalanan basah", maka $p \rightarrow q$ berarti "Jika hujan, maka jalanan basah".

- **Biconditional**

Biconditional (If and only if): Biconditional menghubungkan dua proposisi dan menghasilkan proposisi baru yang bernilai True jika kedua proposisi yang terhubung memiliki nilai kebenaran yang sama (baik keduanya True atau keduanya False). Dalam logika proposional, biconditional dinyatakan dengan menggunakan simbol ' \leftrightarrow '.

Contoh: Jika p adalah proposisi "Saya sedang belajar" dan q adalah proposisi "Saya sedang membaca", maka $p \leftrightarrow q$ berarti "Saya sedang belajar jika dan hanya jika saya sedang membaca".

B. First Order Logic (Predicate Calculus)

First Order Logic (FOL), juga dikenal sebagai Logika Predikat atau Kalkulus Predikat, adalah sistem logika yang digunakan untuk merepresentasikan pengetahuan dalam bentuk pernyataan yang melibatkan predikat, variabel, dan kuantifikasi kuantor. FOL merupakan perluasan dari logika proposional yang memungkinkan penjelasan yang lebih rinci dan formal tentang relasi antarobjek dalam domain.

- **Variable**

Variabel digunakan untuk mewakili objek atau entitas dalam domain yang dianalisis. Variabel sering dilambangkan dengan huruf kecil seperti x, y, z.

- **Konstanta**

Konstanta adalah simbol yang merepresentasikan objek yang tetap dalam domain. Contoh konstanta adalah "John" atau "Mary".

- **Fungsi**

Fungsi adalah simbol atau ungkapan yang menggambarkan hubungan antara objek dalam domain dan menghasilkan nilai baru. Fungsi sering dilambangkan dengan huruf kecil atau kata kerja seperti "FatherOf(x)" yang berarti "x adalah ayah dari seseorang".

- **Predikat**

Predikat adalah simbol atau ungkapan yang menggambarkan sifat atau hubungan antara objek dalam domain. Predikat sering dilambangkan dengan huruf besar atau kata kerja seperti "Man(x)" yang berarti "x adalah seorang pria".

Contoh pernyataan dalam FOL:

- $\forall x \text{ Man}(x) \rightarrow \text{Mortal}(x)$ (Untuk setiap x, jika x adalah seorang pria, maka x adalah makhluk fana)
- $\exists x \text{ FatherOf}(x, \text{John})$ (Ada seseorang yang adalah ayah dari John)

FOL digunakan secara luas dalam kecerdasan buatan, pemrosesan bahasa alami, pemrograman logika, dan pemahaman pengetahuan. Ini memberikan kerangka kerja formal untuk merepresentasikan pengetahuan, melakukan penalaran, dan menghasilkan kesimpulan logis. Dengan FOL, kita dapat mengartikan dan menganalisis pernyataan dengan lebih rinci dan akurat dalam domain yang kompleks.

C. Fuzzy Logic

Logika Fuzzy adalah cabang logika matematika yang digunakan untuk memodelkan dan mengolah ketidakpastian dan keambiguan dalam penalaran. Berbeda dengan logika klasik yang hanya mengakui nilai kebenaran yang mutlak yaitu benar (True) atau salah (False), logika fuzzy mengizinkan nilai kebenaran yang dapat berada di antara dua nilai tersebut, seperti "sedikit benar" atau "cukup benar".

Dalam logika fuzzy, variabel dan pernyataan dinyatakan dalam bentuk fungsi keanggotaan yang memetakan nilai-nilai ke dalam interval antara 0 dan 1. Fungsi keanggotaan ini menunjukkan sejauh mana suatu elemen atau variabel memenuhi suatu sifat atau kondisi tertentu. Nilai keanggotaan ini mencerminkan tingkat kebenaran dari pernyataan tersebut.

Logika fuzzy juga memperkenalkan operasi-operasi logika fuzzy seperti konjungsi fuzzy, disjungsi fuzzy, negasi fuzzy, dan implikasi fuzzy. Ketika operasi-operasi logika fuzzy diterapkan pada variabel fuzzy, nilai kebenaran hasil operasi tersebut dapat menggambarkan tingkat ketidakpastian atau keambiguan dalam penalaran.

Keuntungan utama logika fuzzy adalah kemampuannya untuk mengatasi ketidakpastian dan keambiguan dalam pemodelan dan penalaran. Ini memungkinkan logika fuzzy digunakan dalam berbagai aplikasi yang melibatkan penalaran berbasis aturan, pengendalian otomatis, pengambilan keputusan, dan sistem pendukung keputusan.

Contoh penggunaan logika fuzzy adalah dalam sistem pengendalian kecepatan kendaraan berdasarkan kondisi lalu lintas. Dalam sistem logika fuzzy ini, variabel seperti "kecepatan" dan "jarak" dinyatakan dalam bentuk himpunan fuzzy, dan aturan-aturan fuzzy didefinisikan untuk mengendalikan kecepatan kendaraan berdasarkan kombinasi kecepatan dan jarak. Hal ini memungkinkan sistem pengendalian untuk mengambil

keputusan yang lebih adaptif dan responsif terhadap kondisi lalu lintas yang berubah-ubah.

Logika fuzzy memberikan kerangka kerja yang kuat untuk mengolah ketidakpastian dan keambiguan dalam penalaran dan pengambilan keputusan. Dalam banyak kasus di mana lingkungan atau data yang dihadapi tidak dapat diukur secara akurat atau tegas, logika fuzzy dapat memberikan solusi yang lebih fleksibel dan efektif.

1. Fuzzyness & Probability

Fuzziness dan probabilitas adalah dua konsep yang berbeda dalam konteks ketidakpastian. Fuzziness mengacu pada tingkat ketidakpastian atau keambiguan dalam suatu konsep atau pernyataan, sedangkan probabilitas mengukur tingkat keyakinan atau peluang suatu peristiwa terjadi. Fuzziness berkaitan dengan logika fuzzy, sementara probabilitas berkaitan dengan teori probabilitas. Dalam logika fuzzy, nilai kebenaran adalah kontinum antara benar dan salah, sedangkan dalam probabilitas, nilai peluang adalah antara 0 dan 1. Meskipun ada beberapa persamaan konseptual antara fuzziness dan probabilitas, keduanya digunakan dalam konteks yang berbeda dan memiliki pendekatan yang berbeda dalam menghadapi ketidakpastian.

2. Fuzzy Set

Fuzzy set (himpunan fuzzy) adalah konsep dasar dalam logika fuzzy. Sebuah himpunan fuzzy adalah himpunan yang elemen-elemennya memiliki derajat keanggotaan yang berada dalam interval antara 0 dan 1. Derajat keanggotaan ini mencerminkan tingkat keterkaitan elemen dengan himpunan tersebut. Dalam himpunan klasik, sebuah elemen hanya bisa anggota atau bukan anggota dari himpunan, sedangkan dalam himpunan fuzzy, elemen dapat memiliki tingkat keanggotaan yang berbeda dalam himpunan tersebut. Misalnya, dalam himpunan fuzzy "tinggi", tinggi suatu orang dapat memiliki derajat keanggotaan yang berbeda-beda, seperti 0.8 atau 0.5, yang menunjukkan sejauh mana tinggi orang tersebut memenuhi kriteria tinggi.

3. Fuzzy Logic

Fuzzy logic (logika fuzzy) adalah pendekatan logika yang menggunakan konsep nilai kebenaran yang berada dalam rentang kontinum antara benar dan salah. Dalam logika fuzzy, proposisi atau pernyataan dapat memiliki nilai kebenaran yang bukan hanya benar atau salah, tetapi juga berbagai tingkatan di antaranya, seperti "sedikit benar" atau "cukup benar". Logika fuzzy memperluas logika klasik dengan memperkenalkan fungsi keanggotaan dan operasi-operasi logika fuzzy seperti konjungsi fuzzy, disjungsi fuzzy, negasi fuzzy, dan implikasi fuzzy. Fuzzy logic digunakan dalam berbagai aplikasi yang melibatkan penalaran berbasis aturan dan pengambilan keputusan di bawah

Fuzzy logic menyediakan cara untuk memodelkan dan merumuskan informasi yang samar dan tidak pasti. Ini sangat berguna dalam situasi di mana nilai numerik yang tepat sulit didapatkan atau di mana penalaran dan pertimbangan manusia terlibat.

Beberapa aplikasi utama fuzzy logic meliputi:

1. Sistem Kontrol: Fuzzy logic umum digunakan dalam sistem kontrol untuk menangani lingkungan yang kompleks dan tidak pasti. Kontroler fuzzy dapat menangkap pengetahuan ahli dan memberikan kontrol yang kuat dalam situasi di mana metode kontrol tradisional mungkin tidak mencukupi.

2. Pengambilan Keputusan: Fuzzy logic dapat digunakan dalam proses pengambilan keputusan untuk mengakomodasi informasi yang tidak tepat atau tidak lengkap. Ini memungkinkan representasi faktor-faktor subjektif atau kualitatif yang sulit diukur secara tepat.

3. Pengenalan Pola: Teknik-teknik fuzzy logic diterapkan dalam tugas pengenalan pola, seperti pengenalan gambar dan ucapan. Sistem fuzzy dapat mengatasi variasi dan ketidakpastian dalam data, sehingga cocok untuk menangani pola yang berisik atau ambigu.

4. Sistem Pakar: Fuzzy logic digunakan dalam sistem pakar untuk meniru penalaran dan pengambilan keputusan manusia. Dengan menggabungkan aturan fuzzy dan mekanisme inferensi fuzzy, sistem-sistem ini dapat menangani pengetahuan yang tidak tepat dan tidak pasti.

5. Pemrosesan Bahasa Alami: Fuzzy logic digunakan dalam tugas pemrosesan bahasa alami, seperti pengambilan informasi dan klasifikasi teks. Ini membantu dalam menangani kesamaran dan ambiguitas inheren dari bahasa manusia.

Penting untuk dicatat bahwa fuzzy logic bukan pengganti logika klasik, tetapi alat tambahan yang melengkapi logika tradisional untuk mengatasi informasi yang tidak pasti dan samar dengan lebih efektif.

4. Fuzzy System

Sistem fuzzy (fuzzy systems) merujuk pada sistem yang menggunakan prinsip-prinsip logika fuzzy untuk memodelkan dan mengontrol sistem yang kompleks. Sistem fuzzy memanfaatkan konsep-konsep seperti himpunan fuzzy, aturan inferensi, dan metode defuzzifikasi untuk mengambil keputusan atau menghasilkan keluaran berdasarkan masukan yang tidak pasti atau ambigu.

Sistem fuzzy terdiri dari tiga komponen utama:

- **Fuzzification (Fuzzifikasi):** Proses konversi masukan numerik atau non-numerik menjadi nilai keanggotaan dalam himpunan fuzzy. Fuzzifikasi dilakukan dengan menggunakan fungsi keanggotaan yang menentukan sejauh mana suatu masukan termasuk dalam himpunan fuzzy yang relevan.
- **Rule Base (Basis Aturan):** Basis aturan merupakan himpunan aturan fuzzy yang menghubungkan kondisi (berupa himpunan fuzzy yang diperoleh dari fuzzifikasi) dengan tindakan atau keluaran (juga dalam bentuk himpunan fuzzy). Aturan-aturan ini berbentuk "JIKA kondisi A maka tindakan B". Basis aturan dikembangkan berdasarkan pengetahuan manusia atau melalui teknik pembelajaran.
- **Inference Engine (Mesin Inferensi):** Mesin inferensi melakukan proses penalaran fuzzy berdasarkan basis aturan dan masukan yang diberikan. Mesin inferensi menggabungkan aturan-aturan fuzzy dengan menggunakan operasi logika fuzzy (seperti konjungsi, disjungsi, implikasi) untuk menghasilkan keluaran fuzzy.

Sistem fuzzy digunakan dalam berbagai bidang aplikasi, termasuk pengendalian otomatis, pengenalan pola, pengambilan keputusan, analisis data, penggalian data, prediksi, dan banyak lagi. Kelebihan sistem fuzzy adalah kemampuannya untuk mengatasi ketidakpastian, ambiguitas, dan kompleksitas dalam sistem yang sulit dipahami dengan pendekatan klasik. Sistem fuzzy juga memungkinkan integrasi pengetahuan manusia dan pemodelan yang lebih intuitif.

BAB V

PLANNING(TEKNIK DEKOMPOSISI MASALAH)

A. Goal Stack Planning (GSP)

Goal Stack Planning (GSP) adalah salah satu pendekatan dalam perencanaan AI yang digunakan untuk merencanakan urutan tindakan yang dapat mencapai tujuan tertentu. GSP berfokus pada representasi tujuan sebagai tumpukan (stack) dari subtujuan yang harus dicapai untuk mencapai tujuan akhir.

Dalam GSP, tumpukan tujuan dimulai dengan tujuan akhir dan berisi subtujuan yang harus dicapai secara berturut-turut. Setiap subtujuan mewakili keadaan atau kondisi yang harus dicapai dalam proses perencanaan. Tindakan dihasilkan berdasarkan pemecahan setiap subtujuan dalam tumpukan tujuan, dan setiap tindakan dapat memunculkan subtujuan baru yang ditambahkan ke tumpukan.

Berikut adalah langkah-langkah umum dalam Goal Stack Planning:

- Inisialisasi: Tujuan akhir ditempatkan di bagian atas tumpukan tujuan.
- Seleksi Subtujuan: Subtujuan yang paling atas di tumpukan tujuan dipilih untuk dipecahkan.
- Generasi Tindakan: Untuk mencapai subtujuan yang dipilih, tindakan yang memenuhi prasyarat subtujuan tersebut dihasilkan. Prasyarat adalah keadaan atau kondisi yang harus terpenuhi sebelum tindakan dapat dilakukan.
- Pemilihan Tindakan: Tindakan yang dihasilkan dievaluasi dan dipilih berdasarkan kriteria tertentu, seperti relevansi dan efektivitas dalam mencapai subtujuan.
- Penyempurnaan Tindakan: Jika diperlukan, tindakan yang dipilih dapat diperinci lebih lanjut atau dikombinasikan dengan tindakan lain untuk mencapai subtujuan.
- Penambahan Subtujuan: Jika tindakan yang dipilih menghasilkan subtujuan baru, subtujuan tersebut ditambahkan ke tumpukan tujuan.
- Proses Berulang: Langkah-langkah 2-6 diulang sampai seluruh tumpukan tujuan habis dan tujuan akhir tercapai.

GSP memberikan pendekatan sistematis dalam merencanakan tindakan untuk mencapai tujuan kompleks. Dengan menggunakan tumpukan tujuan, GSP dapat melacak progres dalam mencapai tujuan dan memperhitungkan ketergantungan antara subtujuan. GSP telah digunakan dalam berbagai aplikasi seperti perencanaan robotik, sistem kontrol, dan permainan video.

B. Constraint Posting(CP)

Constraint Posting (CP) adalah salah satu pendekatan dalam pemecahan masalah yang digunakan untuk menyelesaikan masalah yang melibatkan sejumlah kendala atau batasan. CP berfokus pada pemodelan dan penyelesaian masalah dengan menyatakan kendala secara eksplisit dan memanfaatkan teknik-teknik khusus untuk menemukan solusi yang memenuhi semua kendala tersebut.

Dalam CP, kendala dinyatakan dalam bentuk konstrain logika, yaitu pernyataan yang menghubungkan variabel-variabel dengan relasi tertentu. Konstrain-konstrain ini mencerminkan hubungan yang harus dipenuhi oleh variabel-variabel dalam masalah yang sedang diselesaikan.

Berikut adalah langkah-langkah umum dalam Constraint Posting:

- Definisi Variabel: Variabel-variabel yang relevan untuk masalah didefinisikan, dan domain masing-masing variabel ditentukan.
- Spesifikasi Kendala: Kendala-kendala yang mengatur hubungan antara variabel-variabel dinyatakan. Kendala dapat bersifat uniterm (menghubungkan satu variabel) atau multiterm (menghubungkan beberapa variabel).
- Pemecahan Kendala: Teknik-teknik pemecahan kendala digunakan untuk menemukan solusi yang memenuhi semua kendala yang diberikan. Beberapa teknik yang umum digunakan dalam CP meliputi Constraint Satisfaction Problem (CSP), Constraint Optimization Problem (COP), dan Constraint Programming.
- Verifikasi dan Penyelesaian: Solusi yang ditemukan diperiksa dan dievaluasi untuk memastikan bahwa semua kendala terpenuhi. Jika ada kendala yang tidak terpenuhi, langkah-langkah tambahan diambil untuk memperbaikinya atau memodifikasi kendala agar dapat terpenuhi.

CP telah digunakan dalam berbagai bidang aplikasi, termasuk perencanaan jadwal, optimisasi produksi, permasalahan penjadwalan, dan pemecahan masalah logistik. Pendekatan CP sangat berguna dalam kasus-kasus di mana terdapat banyak kendala yang harus diperhatikan dan di mana pemodelan eksplisit dari kendala-kendala tersebut penting untuk mencapai solusi yang memenuhi semua kriteria..

BAB VI

DecisionTree LEARNING

A. ID3

ID3 (Iterative Dichotomiser 3) adalah algoritma pembelajaran mesin yang digunakan untuk membangun pohon keputusan (decision tree) dalam konteks klasifikasi. Algoritma ini dikembangkan oleh Ross Quinlan pada tahun 1986 dan merupakan salah satu algoritma pembelajaran mesin yang paling awal dan populer.

ID3 bekerja dengan membagi dataset berdasarkan atribut-atribut yang ada dan memilih atribut yang paling informatif untuk digunakan sebagai node pemisah dalam pembentukan pohon keputusan. Tujuan dari ID3 adalah untuk membangun pohon keputusan yang dapat membagi data dengan cara yang paling efisien sehingga menghasilkan klasifikasi yang akurat.

Berikut adalah langkah-langkah umum dalam algoritma ID3:

- **Memilih Atribut Pemisah:** ID3 memilih atribut yang paling informatif sebagai atribut pemisah pada setiap langkah dalam membangun pohon keputusan. Atribut dengan informasi yang tinggi akan memberikan pemisahan yang paling baik antara kelas target yang berbeda.
- **Membangun Node Pohon:** Setiap atribut pemisah digunakan sebagai node pada pohon keputusan. Dataset dibagi berdasarkan nilai-nilai atribut pemisah tersebut.
- **Rekursif:** Langkah-langkah 1 dan 2 diulang secara rekursif untuk setiap percabangan dari pohon keputusan yang sedang dibangun. Rekursi berlanjut hingga satu atau lebih kondisi berikut terpenuhi: (a) Semua instance dalam subset dataset terklasifikasi ke dalam kelas yang sama, atau (b) Tidak ada atribut pemisah yang tersisa untuk dijadikan node.
- **Penentuan Label Leaf Node:** Ketika rekursi berakhir, label kelas yang paling umum dari instance dalam subset dataset diberikan kepada leaf node yang sesuai.

ID3 mengandalkan metrik informasi seperti entropi dan gain informasi untuk mengukur kebaikan pemisahan atribut dan memilih atribut yang paling informatif. Semakin tinggi gain informasi atau semakin rendah entropi, atribut tersebut dianggap lebih baik untuk digunakan sebagai pemisah.

B. C4.5

C4.5 adalah algoritma pembelajaran mesin yang digunakan untuk membangun pohon keputusan (decision tree) dari data pelatihan. Tujuannya adalah untuk melakukan klasifikasi data atau memprediksi nilai target berdasarkan fitur-fitur yang ada dalam data.

Berikut adalah langkah-langkah utama dalam algoritma pembelajaran C4.5:

- **Memilih atribut terbaik:** C4.5 menggunakan metode pengukuran gain rasio untuk

memilih atribut yang paling baik dalam memisahkan data. Gain rasio menggabungkan pengukuran informasi (seperti yang digunakan dalam ID3) dengan penyeimbangan ukuran atribut.

- Membuat simpul dalam pohon keputusan: Setelah atribut terbaik dipilih, simpul baru dibuat dalam pohon keputusan berdasarkan atribut tersebut. Setiap cabang dari simpul mewakili nilai-nilai yang mungkin dari atribut.
- Membagi data berdasarkan nilai atribut: Data pelatihan dibagi menjadi subset-subset yang berbeda berdasarkan nilai-nilai atribut yang mungkin. Setiap subset kemudian diproses secara rekursif menggunakan langkah-langkah C4.5 untuk membangun cabang-cabang pohon keputusan yang lebih lanjut.
- Mengulangi langkah-langkah di atas: Langkah-langkah 1 hingga 3 diulangi secara iteratif untuk setiap cabang atau simpul baru yang dibuat hingga semua data pelatihan terbagi dengan sempurna atau sampai kondisi penghentian tertentu terpenuhi.
- Menentukan label atau kelas di daun: Setelah pohon keputusan dibangun, label atau kelas ditentukan untuk setiap daun dalam pohon berdasarkan mayoritas kelas data pelatihan yang terkait.

Selain membangun pohon keputusan, C4.5 juga dapat mengatasi atribut kontinu dengan melakukan diskritisasi atau pengelompokan nilai kontinu menjadi beberapa interval. Selain itu, C4.5 dapat menangani data yang hilang (missing data) dengan mengganti nilai yang hilang berdasarkan statistik atau menggunakan probabilitas distribusi atribut yang ada.

C4.5 adalah salah satu algoritma pembelajaran pohon keputusan yang lebih canggih dan fleksibel dibandingkan dengan ID3, dan telah digunakan secara luas dalam berbagai aplikasi klasifikasi dan prediksi.

BAB VII

JARINGAN SYARAF TIRUAN (JST)

A. Konsep Dasar JST

Konsep dasar Jaringan Syaraf Tiruan (JST) melibatkan beberapa elemen utama yang membentuk struktur dan fungsi jaringan tersebut. Berikut adalah konsep dasar yang perlu dipahami dalam JST:

- **Neuron**
Neuron adalah unit pemrosesan dasar dalam JST, yang mirip dengan neuron dalam sistem saraf manusia. Setiap neuron menerima input dari neuron sebelumnya, memproses informasi dengan menggunakan fungsi aktivasi, dan menghasilkan output. Output neuron ini kemudian menjadi input untuk neuron-neuron berikutnya.
- **Bobot**
Bobot adalah parameter numerik yang terhubung ke setiap koneksi antara neuron. Bobot mengatur kontribusi relatif dari input yang diterima oleh neuron dalam pemrosesan informasi. Bobot ini diatur selama proses pelatihan jaringan untuk mempengaruhi pembelajaran dan hasil prediksi.
- **Fungsi Aktivasi**
Fungsi aktivasi diterapkan pada setiap neuron untuk mengubah nilai input menjadi output yang berbeda. Fungsi aktivasi ini memungkinkan jaringan untuk memodelkan hubungan non-linear antara input dan output. Contoh fungsi aktivasi yang umum digunakan adalah fungsi sigmoid, fungsi ReLU (Rectified Linear Unit), dan fungsi tangen hiperbolik.
- **Layer**
JST terdiri dari beberapa layer yang terhubung satu sama lain. Layer pertama disebut input layer, menerima input data. Layer terakhir disebut output layer, menghasilkan output atau prediksi. Layer di antara input dan output layer disebut hidden layer. Setiap neuron dalam layer terhubung ke neuron dalam layer sebelumnya dan layer selanjutnya.
- **Feedforward**
Feedforward adalah proses mengirimkan input melalui jaringan dari input layer ke output layer. Informasi mengalir maju melalui koneksi-koneksi antar neuron hingga mencapai output layer. Selama feedforward, setiap neuron melakukan komputasi berdasarkan bobot dan fungsi aktivasinya sendiri.
- **Backpropagation**
Backpropagation adalah algoritma yang digunakan untuk melatih jaringan syaraf tiruan. Dalam algoritma ini, kesalahan atau perbedaan antara output yang dihasilkan oleh jaringan dan output yang diharapkan dikembalikan ke belakang melalui jaringan. Bobot-bobot yang menghubungkan neuron-neuron diperbarui berdasarkan kesalahan ini untuk meningkatkan kinerja jaringan.

Dengan konsep dasar ini, JST dapat memodelkan hubungan kompleks antara input dan output, dan dapat belajar dari data untuk membuat prediksi atau pengambilan keputusan. JST telah berhasil diterapkan dalam berbagai aplikasi AI, seperti pengenalan pola, klasifikasi, prediksi, dan pengenalan suara.

B. Model Syaraf Tiruan (Neuron)

Model syaraf tiruan, atau lebih dikenal sebagai neuron, adalah unit komputasional dasar dalam jaringan syaraf tiruan (artificial neural networks). Neuron digunakan untuk memodelkan sifat-sifat dasar dari neuron biologis dalam otak manusia, dan menjadi komponen utama dalam pemodelan dan pemahaman komputasional sistem saraf.

Sebuah neuron tiruan terdiri dari beberapa komponen penting:

- **Input:** Neuron menerima input dari neuron-neuron lain atau sumber input eksternal. Setiap input diberi bobot (weight) yang menggambarkan kekuatan dan pengaruh relatif dari input tersebut.
- **Fungsi Aktivasi:** Input-input yang diterima oleh neuron diolah menggunakan fungsi aktivasi. Fungsi ini menggabungkan input-input tersebut dan menghasilkan output yang akan diteruskan ke neuron-neuron lain dalam jaringan.
- **Bobot:** Setiap input memiliki bobot yang menggambarkan kekuatan pengaruhnya terhadap output neuron. Bobot ini dapat disesuaikan selama proses pembelajaran untuk mengoptimalkan kinerja jaringan.
- **Bias:** Neuron juga dapat memiliki bias yang merupakan nilai tetap yang ditambahkan ke input sebelum diolah oleh fungsi aktivasi. Bias membantu mengontrol keputusan yang dibuat oleh neuron.
- **Fungsi Aktivasi:** Fungsi aktivasi menentukan output neuron berdasarkan input yang diterima. Beberapa fungsi aktivasi umum termasuk sigmoid, ReLU, dan tangen hiperbolik. Fungsi ini membantu memperkenalkan sifat non-linearitas ke dalam jaringan syaraf tiruan.

Neuron-neuron dalam jaringan syaraf tiruan saling terhubung dan berinteraksi untuk melakukan komputasi secara paralel. Proses pembelajaran, seperti backpropagation, digunakan untuk mengatur bobot dan bias pada setiap neuron dalam rangka memperbaiki kinerja jaringan. Dengan demikian, jaringan syaraf tiruan dapat belajar dari data latih dan melakukan tugas seperti klasifikasi, regresi, pengenalan pola, dan lainnya.

Model syaraf tiruan telah digunakan dalam berbagai aplikasi, termasuk pengenalan wajah, prediksi harga saham, pengenalan suara, dan pemrosesan bahasa alami.

C. Aktivasi & Arsitektur Jaringan pada JST

Dalam jaringan saraf tiruan (JST), aktivasi mengacu pada fungsi matematika yang diterapkan pada output dari setiap neuron dalam jaringan. Fungsi ini menentukan apakah neuron tersebut harus aktif atau tidak aktif, atau sejauh mana kontribusinya terhadap output jaringan.

Fungsi aktivasi memperkenalkan sifat non-linearitas ke dalam jaringan saraf tiruan, yang memungkinkannya untuk memodelkan hubungan yang kompleks antara input dan output. Tanpa fungsi aktivasi non-linear, jaringan saraf tiruan akan menjadi jaringan linier yang terbatas dalam kapasitas representasinya.

Beberapa fungsi aktivasi yang umum digunakan dalam JST adalah:

- **Sigmoid:** Fungsi sigmoid (atau fungsi logistik) memiliki rentang output antara 0 dan 1. Fungsi ini digunakan untuk menghasilkan output yang terbatas pada rentang tersebut dan sering digunakan dalam jaringan saraf tiruan dengan arsitektur sederhana.
- **ReLU (Rectified Linear Unit):** Fungsi ReLU menghasilkan output yang sama dengan inputnya jika inputnya positif, dan output 0 jika inputnya negatif. Fungsi ini sering digunakan dalam jaringan saraf tiruan dengan banyak lapisan (deep neural networks) karena dapat membantu mengatasi masalah vanishing gradient.
- **Tanh (Tangen Hiperbolik):** Fungsi tangen hiperbolik memiliki rentang output antara -1 dan 1. Fungsi ini mirip dengan fungsi sigmoid tetapi simetris terhadap sumbu $y = 0$. Fungsi tangen hiperbolik sering digunakan dalam jaringan saraf tiruan dengan lapisan tersembunyi.
- **Softmax:** Fungsi softmax digunakan khusus untuk output layer dalam jaringan saraf tiruan klasifikasi multikelas. Fungsi ini menghasilkan distribusi probabilitas di antara kelas-kelas yang mungkin, memastikan bahwa total probabilitas untuk semua kelas adalah 1.

Arsitektur jaringan dalam JST mengacu pada struktur atau tata letak dari neuron-neuron dalam jaringan. Beberapa arsitektur yang umum digunakan dalam JST termasuk:

- **Jaringan Saraf Tiruan Feedforward:** Merupakan arsitektur paling dasar dalam JST, di mana informasi mengalir hanya dari lapisan input ke lapisan output. Tidak ada koneksi mundur (backward) antara neuron-neuron di dalam lapisan yang sama.
- **Jaringan Saraf Tiruan Rekurent:** Merupakan arsitektur yang memungkinkan aliran mundur informasi dalam jaringan. Neuron-neuron dalam jaringan rekurent dapat memiliki koneksi ke neuron di lapisan sebelumnya, memungkinkan pemrosesan data bersejarah dan memori jangka pendek.
- **Jaringan Saraf Tiruan Konvolusi:** Merupakan arsitektur yang dirancang khusus untuk pemrosesan data berupa grid, seperti gambar. Jaringan konvolusi menggunakan operasi konvolusi untuk mengekstrak fitur-fitur penting dari data dan memiliki lapisan-lapisan khusus seperti lapisan konvolusi, lapis
- **Jaringan Saraf Tiruan Long Short-Term Memory (LSTM):** Merupakan jenis arsitektur

rekurent yang khusus digunakan untuk memodelkan urutan data yang panjang. LSTM memiliki mekanisme memori jangka panjang dan memori jangka pendek yang membantu dalam mempertahankan informasi yang relevan dari urutan sebelumnya.

Setiap arsitektur jaringan memiliki kegunaan dan keunggulan tertentu tergantung pada tugas yang dihadapi. Pemilihan fungsi aktivasi dan arsitektur jaringan yang tepat merupakan faktor penting dalam merancang jaringan saraf tiruan yang efektif untuk tujuan yang diinginkan.

D. Supervised Learning & Unsupervised Learning

Supervised learning dan unsupervised learning adalah dua paradigma utama dalam pembelajaran mesin (machine learning). Berikut adalah penjelasan singkat untuk kedua paradigma tersebut:

1. **Supervised Learning (Pembelajaran Terbimbing):** Dalam supervised learning, algoritma mesin belajar menggunakan dataset yang telah diberi label (labelled dataset) yang menghubungkan setiap contoh data dengan label atau kelas yang sesuai. Tujuan utama dari supervised learning adalah untuk mempelajari pola atau hubungan antara fitur (input) dengan label (output) yang diberikan.

Proses pembelajaran terbimbing melibatkan dua tahap utama: tahap pelatihan dan tahap pengujian. Pada tahap pelatihan, algoritma belajar dari data yang telah diberi label untuk membangun model yang dapat memprediksi label untuk data baru. Pada tahap pengujian, model tersebut diuji dengan menggunakan data yang tidak diberi label untuk mengukur kinerja atau akurasi prediksi model.

Contoh dari supervised learning termasuk klasifikasi (classification), di mana tujuannya adalah untuk memprediksi kelas atau label yang diskret, dan regresi (regression), di mana tujuannya adalah untuk memprediksi nilai yang kontinu. Contoh algoritma supervised learning adalah Decision Trees, Naive Bayes, Support Vector Machines (SVM), dan Artificial Neural Networks (ANN).

2. **Unsupervised Learning (Pembelajaran Tanpa Pengawasan):** Dalam unsupervised learning, algoritma mesin belajar menggunakan dataset yang tidak memiliki label (unlabelled dataset). Tujuan dari unsupervised learning adalah untuk menemukan pola, struktur, atau informasi tersembunyi yang ada dalam dataset tanpa adanya informasi label yang terkait.

Proses unsupervised learning tidak memerlukan label atau kelas yang ditentukan sebelumnya. Algoritma ini menggali informasi dari data dengan mencari pola atau kesamaan di antara contoh-contoh data. Contoh tugas unsupervised learning termasuk clustering (pengelompokan) untuk mengelompokkan data berdasarkan kesamaan dan asosiasi untuk menemukan hubungan antara entitas dalam dataset.

Contoh algoritma unsupervised learning termasuk K-Means Clustering, Hierarchical Clustering, Principal Component Analysis (PCA), dan Generative Adversarial Networks (GAN).

Perbedaan utama antara supervised learning dan unsupervised learning terletak pada ketersediaan label atau kelas dalam dataset. Supervised learning memanfaatkan informasi label yang diberikan untuk mempelajari pola, sedangkan unsupervised learning mencari pola atau struktur tanpa label yang sudah ditentukan.

BAB VIII

ALGORITMA GENETIKA (AG)

A. Definisi algoritma genetika

Algoritma Genetika (Genetic Algorithms) adalah sebuah metode optimisasi dan pencarian heuristik yang terinspirasi oleh proses evolusi dalam alam. Algoritma ini menggunakan konsep seleksi alam, reproduksi, dan mutasi untuk mencari solusi optimal dalam ruang pencarian yang kompleks.

B. Komponen-komponen AG

1. Populasi

Populasi dalam algoritma genetika merujuk pada kumpulan individu-individu atau solusi-solusi potensial yang ada pada setiap generasi. Setiap individu dalam populasi mewakili suatu kemungkinan solusi untuk masalah yang sedang diselesaikan.

2. Genotype

Genotype adalah representasi genetik dari individu dalam populasi. Biasanya, genotype direpresentasikan dalam bentuk urutan gen atau bit yang mengodekan informasi tentang solusi yang diusulkan. Misalnya, dalam masalah optimisasi, genotype dapat berupa urutan nilai-nilai parameter yang mencirikan solusi.

3. Fungsi Fitness

Fungsi fitness adalah metrik yang digunakan untuk mengevaluasi sejauh mana suatu individu cocok dengan tujuan yang ditetapkan dalam masalah yang sedang diselesaikan. Fungsi fitness mengukur kualitas atau kinerja individu dalam mencapai solusi yang diinginkan. Nilai fitness biasanya ditentukan berdasarkan kriteria yang telah ditentukan sebelumnya.

4. Seleksi

Seleksi adalah proses dalam algoritma genetika di mana individu-individu terbaik dipilih dari populasi untuk menjadi orang tua dalam generasi berikutnya. Seleksi biasanya didasarkan pada nilai fitness individu, di mana individu dengan nilai fitness yang lebih tinggi memiliki peluang yang lebih besar untuk dipilih. Prinsip seleksi adalah bahwa individu yang lebih baik dalam mencapai tujuan akan memberikan kontribusi genetik yang lebih baik untuk generasi berikutnya.

5. Rekombinasi (Crossover)

Rekombinasi, juga dikenal sebagai crossover, adalah proses di mana bagian-bagian genotype antara dua individu dipertukarkan untuk menghasilkan keturunan baru. Proses ini mensimulasikan persilangan genetik dalam reproduksi seksual alami. Rekombinasi menghasilkan variasi genetik baru dalam populasi dan membantu menjaga keragaman dan eksplorasi ruang pencarian.

6. Mutasi

Mutasi adalah proses acak di mana beberapa gen dalam genotipe individu mengalami perubahan. Tujuan mutasi adalah memperkenalkan variasi genetik baru yang tidak ada dalam populasi saat ini. Mutasi membantu dalam eksplorasi ruang pencarian dan mencegah populasi terjebak dalam minimum lokal. Tingkat mutasi yang tepat penting untuk menjaga keseimbangan antara eksplorasi dan eksploitasi.

7. Penggantian Generasi

Setelah dilakukan seleksi, rekombinasi, dan mutasi, generasi baru individu dibentuk untuk generasi berikutnya. Dalam proses penggantian generasi, individu-individu baru dan beberapa individu terbaik dari generasi sebelumnya digunakan untuk membentuk populasi baru. Dengan demikian, proses evaluasi, seleksi, rekombinasi, dan mutasi diulang dalam setiap generasi hingga kondisi berhenti terpenuhi atau solusi yang memadai ditemukan.

BAB IX

COGNITIF SCIENCE

A. Defenisi

Ilmu kognitif, atau ilmu kognitif, adalah bidang studi lintas disiplin yang menyelidiki berbagai aspek kognisi, termasuk persepsi, berpikir, penalaran, ingatan, bahasa, pemecahan masalah, pengambilan keputusan, dan kesadaran. Bidang ini menggabungkan metodologi dan prinsip-prinsip dari psikologi, neurosains, linguistik, filsafat, ilmu komputer, antropologi, dan disiplin terkait lainnya.

Tujuan utama ilmu kognitif adalah untuk memahami bagaimana pikiran bekerja dan bagaimana proses kognitif berkontribusi terhadap perilaku dan aktivitas mental manusia. Para peneliti dalam ilmu kognitif berupaya mengungkapkan mekanisme dan proses dasar yang mengatur kognisi manusia, termasuk bagaimana informasi diakuisisi, diproses, disimpan, dan digunakan.

Ilmu kognitif memandang kognisi dari perspektif komputasional, menganggap pikiran sebagai sistem pemrosesan informasi. Bidang ini menggunakan metode empiris, termasuk percobaan, observasi, dan pemodelan komputasional, untuk mempelajari fenomena kognitif dan mengembangkan teori dan model yang menjelaskan dan memprediksi proses kognitif manusia.

Beberapa area kunci studi dalam ilmu kognitif meliputi:

- **Psikologi Kognitif:** Mempelajari proses mental seperti persepsi, perhatian, ingatan, pembelajaran, dan pemecahan masalah.
- **Neurosains:** Menyelidiki dasar saraf dari kognisi dan bagaimana otak mendukung fungsi kognitif.
- **Linguistik:** Berfokus pada studi bahasa, termasuk perolehan bahasa, pemahaman, produksi, dan struktur bahasa.
- **Filsafat Pikiran:** Mengeksplorasi pertanyaan-pertanyaan mendasar tentang sifat pikiran, kesadaran, dan hubungan antara pikiran dan tubuh.
- **Kecerdasan Buatan:** Menerapkan model dan algoritma komputasional untuk mensimulasikan dan memahami proses kognitif serta mengembangkan sistem cerdas.

Dengan mengintegrasikan pengetahuan dari berbagai disiplin, ilmu kognitif bertujuan untuk memberikan pemahaman komprehensif tentang kognisi manusia dan mekanisme yang mendasarinya. Temuan dan wawasan dari ilmu kognitif berkontribusi pada berbagai aplikasi praktis, termasuk pengembangan teknologi seperti kecerdasan buatan, interaksi manusia-komputer, rehabilitasi kognitif, dan intervensi pendidikan.

DAFTAR PUSTAKA

- Russell, S., & Norvig, P. (2016). Artificial Intelligence: A Modern Approach 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. Pearson.
- Poole, D., Mackworth, A., & Goebel, R. (2017). Computational Intelligence: A Logical Approach. Oxford University Press.
- Stuart Russell, P., & Peter Norvig, N. (2010). Artificial Intelligence: A Modern Approach (Vol. 1). Pearson.
- Nils J. Nilsson. (2009). Introduction to Artificial Intelligence. Cambridge University Press.
- Winston, P. H. (1992). Artificial Intelligence. Addison-Wesley.
- Rich, E., & Knight, K. (1991). Artificial Intelligence. McGraw-Hill.
- Bratko, I. (2001). Prolog Programming for Artificial Intelligence. Addison Wesley.
- Hayes-Roth, B., & Hayes-Roth, F. (1995). A Cognitive Model of Planning. Artificial Intelligence, 69(1-2), 287-349.