

1. Inteligentni sistemi su računarski sistemi koji koriste veštačku inteligenciju za donošenje odluka i rešavanje problema. Primeri primene uključuju personalne asistente, sisteme preporuka, autonomna vozila i sisteme za detekciju prevare.
2. Prostor stanja predstavlja sva moguća stanja sistema, a veličina prostora stanja je broj mogućih stanja sistema.
3. Slijepo pretraživanje je metoda pretraživanja prostora stanja u potrazi za rješenjem, koja se temelji na sistemu pravila za generiranje novih stanja i provjeri da li je svako novo stanje rješenje ili ne. Ova metoda ne koristi nikakvo znanje o problemu ili cilju, već samo gradi stablo mogućih stanja i pretražuje ga do pronalaska rješenja.
4. Pretraživanje u širinu je algoritam za pretraživanje grafa koji se koristi za pronalaženje najkraćeg puta od početnog do ciljnog čvora u neusmjerenom ili usmjerenom grafu. Algoritam radi tako da se prvo posjećuju svi susjedi početnog čvora, zatim susjedi njegovih susjeda i tako dalje, sve dok se ne posjeti ciljni čvor ili se ne obiđe cijeli graf. Ovaj algoritam osigurava da se najkraći put pronađe ako postoje putovi koji vode do ciljnog čvora.
5. Pretraživanje u dubinu je jedna od tehnika pretraživanja u umjetnoj inteligenciji koja se koristi za pronalaženje rješenja u prostoru stanja. Ono počinje od početnog čvora i nastavlja pretraživati sve dublje dok ne pronađe rješenje ili dok ne dođe do granice dubine koja je definirana. Ova tehnika pretraživanja često koristi stog za pohranjivanje stanja kako bi mogla vratiti prethodno stanje i nastaviti pretraživanje ako nije pronađeno rješenje.
6. Informirano pretraživanje, također poznato kao heurističko pretraživanje, je strategija pretraživanja prostora stanja koja koristi dodatne informacije, poznate kao heuristike, kako bi bolje usmjerila pretraživanje prema cilju. Heuristike su funkcije koje procjenjuju koliko je neko stanje blizu cilja. Primjerice, u pretraživanju grafa putovanja, heuristika može procijeniti udaljenost nekog grada od cilja i koristiti tu informaciju kako bi usmjerila pretraživanje u pravom smjeru. Informirano pretraživanje je učinkovitije od slijepog pretraživanja jer koristi dodatne informacije za usmjeravanje pretraživanja.
7. Nacrtamo čvorove i iskoristimo neki od gore metoda.. Najcesce moze to ici kao najkraca udaljenost ili ruta posjete neke znamenitosti i sl.
8. Naivni Bayes-ov klasifikator je algoritam za klasifikaciju koji se temelji na Bayesovoj teoremi i pretpostavlja nezavisnost značajki u podacima. U osnovi, algoritam modelira vjerojatnosti različitih klasa na temelju pojavljivanja značajki u podacima za učenje te koristi ovaj model za klasifikaciju novih podataka. Ovaj klasifikator često se koristi za klasifikaciju teksta i emailova kao spam ili ne-spam.
9. Klasteri su grupisanja sličnih objekata u skupove (klasteri) na osnovu sličnosti njihovih svojstava. Ovakva vrsta analize podataka se naziva klasterovanje. Metode: K sredina, hijerarhijska klaster analiza, mješoviti modeli, SOM (self organizing maps) i druge.
10. Decision tree je metoda strojnog učenja koja klasificira podatke u hijerarhijskoj strukturi stabla odluke, gdje se svakom čvoru dodjeljuje uvjetna grananja za donošenje odluka. To su modeli koji koriste niz logičkih pravila da bi se donijela konačna odluka.
11. Ono sto se moze primjetiti jeste da kako se X povecava, Y se smanjuje. Ovdje je moguće koristiti linearnu regresiju. Treba naci $mx+b=y$ za svaki y. Ima nacin preko srednjih vrijednosti da se sabere svaki x / broj klk ih ima, to je 11. Onda isto za y.

$$x: (-1.84 - 1.52 - 1.49 - 1.18 - 0.66 - 0.15 - 0.02 + 0.42 + 0.92 + 1.34 + 1.57) / 11 = 0.1818$$

$$y: (3.00 + 2.60 + 1.50 + 0.50 + 1.30 + 0.65 + 1.70 - 0.25 - 0.45 - 1.00 - 2.25) / 11 = 0.9727$$

$$\text{Kovarijansa } x \text{ } y: ((-1.84 - 0.1818) * (3.00 - 0.9727)) + ((-1.52 - 0.1818) * (2.60 - 0.9727)) + \dots + ((1.57 - 0.1818) * (-2.25 - 0.9727)) / 11 = -3.5027$$

$$\text{Varijansa za } x: ((-1.84 - 0.1818)^2 + (-1.52 - 0.1818)^2 + \dots + (1.57 - 0.1818)^2) / 11 = 1.4533$$

$$\text{Varijansa za } y: ((3.00 - 0.9727)^2 + (2.60 - 0.9727)^2 + \dots + (-2.25 - 0.9727)^2) / 11 = 3.3095$$

$$m = \text{kovarijanca}(x,y) / \text{varijanca}(x) = -2.4073$$

$$b = \text{SrednjaY} - (m * \text{SrednjaX}) = 0.9727 - (-2.4073 * 0.1818) = 1.3659$$

$$y = -2.4073x + 1.3659$$

12. Neuronske mreže su računalni sustavi koji su modelirani prema radu ljudskog mozga. One koriste složene matematičke modele koji omogućuju prepoznavanje obrazaca i klasifikaciju podataka. U neuronskim mrežama, podaci prolaze kroz skup slojeva neurona, gdje se obrađuju i prilagođavaju težine kako bi se postigla optimalna izlazna vrijednost.
13. Van Neumannova arhitektura je klasična arhitektura računara koja se temelji na odvojenosti procesora i memorije, gdje procesor izvršava instrukcije iz memorije i često koristi seriju instrukcija za izvršavanje složenih zadataka. S druge strane, neuronske mreže su računalni sustavi koji se temelje na arhitekturi sličnoj mozgu, gdje su neuronski čvorovi povezani u mreži koja može naučiti i izvršavati složene zadatke. Glavna razlika između ove dvije arhitekture je u načinu izvršavanja zadataka. U Van Neumannovoj arhitekturi, procesor obrađuje instrukcije u seriji, dok neuronske mreže mogu obraditi ulaze paralelno putem distribuirane obrade podataka. Također, neuronske mreže mogu učiti i prilagođavati se okolini, dok se Van Neumannova arhitektura uglavnom temelji na programiranju i izvršavanju zadataka prema predodređenom planu.
14. Prepoznavanje slika, klasifikacija teksta, predviđanje (predviđanje potrošnje električne energije, cijena dionica, prometa na cestama), medicina (dijagnostika, identifikaciju bolesti, prepoznavanje uzoraka u medicinskim slikama), robotika - koriste se za upravljanje robotima, prepoznavanje objekata u prostoru), financije (predviđanje kreditnog rizika).
15. TLU perceptron (Threshold Logic Unit) je jednostavan model neuronske mreže koji prima više ulaza i izračunava težinsku sumu ulaza. Ta težinska suma se zatim uspoređuje s nekom vrijednošću, koja odlučuje o aktivaciji perceptrona. Ako težinska suma prelazi prag, perceptron se aktivira i vraća izlaz vrijednosti 1, u suprotnom vraća izlaz vrijednost 0. Ovaj model je koristan u binarnim klasifikacijskim zadacima i moguće ga je kombinirati u višeslojne perceptronske mreže za složenije probleme.
16. Nije bas zadana funkcija I trebalo bi vise vrijednosti, ali hajde da kazemo neka je najjednostavnija funkcija $x_1 + x_2$. Vidimo da za 5,2 I 2,5 je 7 peceptron aktivan tj 1, a za 1,5 I 5,1 je -1 tj neaktivan. Nije bas skroz tacno da zakljucimo na osnovu 4 podatka, ali mozemo reci da 6 je referentna vrijednost, tako da svi ulazi cija ce funkcija imati tezinsku vrijednost vecu od 6 bi aktivirali perceptron.
17. Perceptron je jednostavna vrsta neuronske mreže koja se sastoji od jednog neurona s ulazima i jednim izlazom. Algoritam perceptrona koristi se za rješavanje problema klasifikacije, gdje se ulazni primjerci razvrstavaju u dvije ili više klasa. Algoritam perceptrona počinje sa slučajno odabranim težinama za svaki ulazni neuron. Zatim, za svaki primjer u skupu za učenje, izračunava se težinska suma ulaza koji se množe s njihovim težinama. Ako je težinska suma veća od praga (threshold), neuron generira izlaznu vrijednost 1, inače generira izlaznu vrijednost 0. Kada se generira izlaz, uspoređuje se sa željenim izlazom za taj primjer. Ako postoji razlika, težine se prilagođavaju kako bi se smanjila greška između izlaza neurona i željenog izlaza. Ovaj postupak se ponavlja za svaki primjer u skupu za učenje dok se ne postigne zadovoljavajuća točnost klasifikacije.
18. Backpropagation je algoritam za učenje vještačkih neuronskih mreža u nadziranom učenju. Cilj je minimizirati funkciju greške između predviđenih i stvarnih izlaza mreže. Algoritam radi tako što se najprije izračuna izlaz mreže za dani ulaz, zatim se izračuna greška između predviđenog izlaza i stvarnog izlaza te se ta greška propagira unatrag kroz mrežu kako bi se izračunala gradijentna vrijednost funkcije greške po svim težinama. Na kraju se koristi ta gradijentna vrijednost da bi se prilagodile težine mreže kako bi se smanjila funkcija greške.

Postupak se ponavlja za sve primjere za učenje sve dok se ne postigne željena razina performansi.

19. Mogu se podijeliti prema broju slojeva (jednoslojne, višeslojne), prema vrsti povezanosti (potpuno, rijetko, lokalno povezane), namejni (klasifikacija, mreže za obradu), prema vrsti aktivacije neurona

20. Statičke neuronske mreže:

Perceptron - jednostavna neuronska mreža s jednim slojem koji se koristi za binarnu klasifikaciju.

Kohonenova mreža - samoorganizirajuća mreža koja se koristi za klasterizaciju podataka.

Dinamičke neuronske mreže:

Hopfieldova mreža - mreža koja se koristi za memoriranje i rekonstrukciju uzoraka.

Elmanova mreža - rekurentna neuronska mreža koja se koristi za predviđanje sekvenci.

Neizrazite neuronske mreže:

Fuzzy neuronske mreže - mreže koje koriste neizrazite varijable i pravila za zaključivanje u procesu učenja i predviđanja.

Neuralne mreže s neizrazitim parametrima - mreže koje koriste neizrazite parametre za modeliranje nesigurnosti i neodređenosti u procesu učenja i predviđanja.

21. Jednoslojne neuronske mreže su mreže koje se sastoje samo od jednog sloja neurona koji prima ulazne vrijednosti, obrađuje ih i daje izlazne vrijednosti. One se često koriste za jednostavne probleme poput binarne klasifikacije. Primjer jednoslojne neuronske mreže je perceptron. On ima samo jedan sloj neurona koji obrađuje ulazne vrijednosti i daje binarnu izlaznu vrijednost (0 ili 1). Višeslojne neuronske mreže sastoje se od više slojeva neurona koji obrađuju ulazne vrijednosti u niz složenih transformacija kako bi se dobila konačna izlazna vrijednost. Ove mreže se često koriste za rješavanje složenijih problema poput prepoznavanja slika, prirodnog jezika i drugih. Primjer višeslojne neuronske mreže je konvolucijska neuronska mreža (CNN) koja se često koristi za prepoznavanje slika. Ona se sastoji od slojeva koji obrađuju prostorne značajke slike, a zatim koristi višeslojni perceptron za klasifikaciju slike na temelju obrađenih značajki.

22. Faze rada neuronskih mreža su ulazna, obradna i izlazna faza. U ulaznoj fazi podaci se unose u mrežu, u obradnoj fazi se izračunavaju težine između neurona i ažuriraju se kroz proces učenja, a u izlaznoj fazi mreža daje izlazne vrijednosti za zadane ulaze. Karakteristike ulazne faze su priprema podataka i konverzija u oblik prikladan za obradu, karakteristike obradne faze su izračunavanje težina, korištenje aktivacijske funkcije i postupak učenja, a karakteristike izlazne faze su tumačenje izlaza i eventualna korekcija parametara.

23. Podjela skupa podataka na trening, validaciju i testiranje je važan korak u procesu treniranja neuronskih mreža. Općenito se preporučuje da se oko 70-80% skupa podataka koristi za trening, 10-15% za validaciju i 10-15% za testiranje. Trening skup podataka se koristi za obuku neuronske mreže i prilagodbu parametara modela. Validacijski skup podataka se koristi za provjeru performansi modela tijekom procesa učenja. Testni skup podataka se koristi za konačnu procjenu performansi modela nakon što je obuka završena i parametri su odabrani.

Problem pretreniranosti (overfitting) nastaje kada neuronska mreža postiže vrlo dobre rezultate na trening skupu podataka, ali loše na novim podacima. To se događa kada model postane previše složen i usredotočen je na sjećanje pojedinačnih uzoraka u trening skupu podataka umjesto da nauči općenite značajke podataka.

Problem nedostatne generalizacije (underfitting) nastaje kada neuronska mreža nije dovoljno složena da nauči općenite značajke podataka, pa se performanse na trening, validacijskom i testnom skupu podataka ne poboljšavaju.

24. Ima već objašnjeno. Backpropagation npr l perceptron
25. Podržano učenje je područje strojnog učenja koje se fokusira na razvoj algoritama koji automatski poboljšavaju svoje performanse kroz interakciju s okolinom. Umjesto da se koristi prethodno pripremljen skup podataka, algoritmi podržanog učenja uče izravno iz iskustva, a modeli se prilagođavaju kako bi se poboljšala njihova sposobnost rješavanja problema u stvarnom svijetu.
26. Aktivacijske funkcije su matematičke funkcije koje se primjenjuju na izlazu svakog neurona u neuronskoj mreži kako bi se izračunala aktivacija neurona i proslijeđivala dalje u mrežu. Najčešće korištene aktivacijske funkcije u MLP (višeslojnim perceptronima) mrežama su:
- Sigmoidna funkcija: transformira bilo koji realni broj u rasponu od 0 do 1
 - ReLU (Rectified Linear Unit) funkcija: transformira sve negativne vrijednosti u 0, a sve pozitivne vrijednosti ostavlja nepromijenjenima
 - Softmax funkcija: koristi se u zadnjem sloju MLP mreže kada se radi klasifikacija, pretvara vektor izlaza u vjerojatnosti za svaku klasu.
27. Hopfieldova neuronska mreža je vrsta rekurentne neuronske mreže koja se koristi za pohranjivanje i oporavak uzoraka. Mreža koristi matricu težina kako bi pohranila uzorke, a zatim iterativno ažurira aktivaciju neurona dok se ne postigne stabilno stanje (stanje u kojem se aktivacija neurona više ne mijenja tijekom iterativnog ažuriranja, te daljnje ažuriranje neće utjecati na aktivaciju neurona). Kada mreža dostigne stabilnost, može usporediti aktivacije neurona s pohranjenim uzorcima kako bi pronašla najbliži uzorak ulaznom uzorku. Ova neuronska mreža se često koristi za rješavanje problema asociiranog pamćenja i ima široku primjenu u različitim područjima kao što su prepoznavanje uzoraka, klasifikacija i optimizacija.
28. Elmanova neuronska mreža je vrsta rekurentne neuronske mreže koja se koristi za predviđanje vremenskih serija podataka. Mreža ima skriveni sloj koji je povezan s izlaznim slojem i ima povratnu vezu na skriveni sloj kako bi se prethodna stanja mreže koristila u predviđanju budućih vrijednosti. Treniranje mreže uključuje određivanje optimalnih težina između neurona, a nakon toga se može koristiti za predviđanje budućih vrijednosti niza. Mreža se često koristi u obradi prirodnog jezika, prepoznavanju govora, predviđanju vremenskih serija i drugim sličnim područjima.
29. NARX (Nonlinear Autoregressive with eXogenous inputs) neuronska mreža je rekurentna neuronska mreža koja se koristi za modeliranje nelinearnih dinamičkih sustava. Mreža ima skriveni sloj koji je povezan s izlaznim slojem, slojem vanjskih ulaza i povratnom vezom na skriveni sloj. Mreža prima seriju vremenskih podataka kao ulaz i vanjske varijable te može predvidjeti buduću vrijednost izlaza na temelju prethodnih vrijednosti ulaza i izlaza.
30. Već je objašnjeno gore u pitanju 23. Rasporedi se skupovi gdje se za testiranje najčešće koristi 10-15%. Vazno je napomenuti samo da se za testiranje koriste podaci koji se razlikuju od onih korištenih za treniranje i validaciju, te da se skup za testiranje koristi samo jednom, nakon što je model razvijen i podešen, kako bi se procijenila njegova sposobnost generalizacije na novim podacima.
31. Heuristika se odnosi na postupak pronalaženja praktičnog rješenja za neki problem na temelju intuicije, iskustva i znanja, bez garancije da će se pronaći najbolje rješenje. Ova metoda se često koristi u situacijama u kojima je pronalaženje najboljeg rješenja preteško ili nemoguće. S druge strane, metaheuristika je opći naziv za skupinu optimizacijskih algoritama koji se mogu primijeniti na širok raspon problema, a da nisu ovisni o specifičnostima problema. Ovi algoritmi su obično inspirirani prirodnim procesima i služe za pretraživanje velikog broja mogućih rješenja kako bi se pronašlo najbolje rješenje. Razlika između heuristike i metaheuristike je u tome što heuristika pruža intuitivno rješenje problema, dok

metaheuristika koristi općenitiji algoritam za pronalaženje rješenja, pri čemu se obično ne razmatraju pojedinosti problema. Dakle, heuristika je obično usmjerena na rješavanje konkretnog problema, dok se metaheuristika može primijeniti na širi raspon problema.

32. Evolucijsko računarstvo (ER) obuhvaća tri glavne vrste algoritama: genetske algoritme (GA), strategije evolucijskih strategija (ES) i evolucijsko programiranje (EP). GA koriste selekciju, križanje i mutaciju kako bi se stvorili novi kandidati za rješenje problema optimizacije. ES koristi prilagodljive parametre za kontrolu strategije pretraživanja, a EP se temelji na evolucijskoj promjeni programskih struktura.

33. Optimizacija se odnosi na proces pronalaženja najboljeg rješenja za određeni problem, obično uzimajući u obzir određene ciljeve, ograničenja i resurse.

Problemi optimizacije se dijele na tri glavne kategorije:

Jednostavni problemi optimizacije - ovi problemi imaju samo jednu varijablu koja se treba optimizirati. Primjer takvog problema je pronalaženje najmanje vrijednosti funkcije.

Višedimenzionalni problemi optimizacije - ovi problemi imaju više varijabli koje se trebaju optimizirati. Primjer takvog problema je optimizacija parametara u neuronskoj mreži.

Problemi optimizacije s ograničenjima - ovi problemi uključuju ograničenja koja moraju biti zadovoljena pri pronalaženju najboljeg rješenja. Primjer takvog problema je pronalaženje najboljeg rasporeda s ograničenjima, gdje se moraju uzeti u obzir ograničenja poput dostupnih resursa i vremenskih ograničenja.

Problemi optimizacije se obično rješavaju primjenom različitih metoda poput linearnog programiranja, kvadratnog programiranja, genetskog programiranja i drugih tehnika optimizacije.

34. Genetički algoritam (GA) je vrsta evolucijskog algoritma koji se temelji na prirodnom procesu evolucije i genetici. GA koristi selekciju, križanje i mutaciju kako bi stvorio populaciju novih kandidata za rješenje problema optimizacije. Osnovna namjena genetičkih algoritama je rješavanje problema optimizacije, posebno onih problema koji uključuju velik broj varijabli, složene funkcije cilja ili ograničenja. GA je posebno koristan u slučajevima kada se funkcija cilja ne može analitički izraziti, a cilj je pronaći globalni optimum te funkcije. Primjeri problema koji se mogu riješiti genetičkim algoritmom su optimizacija rasporeda, pronalaženje najboljih parametara u strojnom učenju i umjetnoj inteligenciji, dizajniranje kompleksnih sustava i sl.

35. Eliminacijski genetički algoritam (Steady State Genetic Algorithm) je varijanta genetičkog algoritma koja se razlikuje od klasičnog genetičkog algoritma po tome što se ne stvara nova populacija kandidata za rješenje već se postojeća populacija nadograđuje i poboljšava.

Pseudo kod:

Inicijaliziraj populaciju P s N jedinki.

Evaluiraj sve jedinke u populaciji P koristeći funkciju cilja.

Dok nije ispunjen uvjet zaustavljanja, ponavljaj sljedeće korake:

- Odaberi dva roditelja iz populacije P koristeći selekcijski operator.

- Križaj roditelje koristeći križanje.

- Mutiraj dijete dobiveno križanjem.

- Evaluiraj dijete koristeći funkciju cilja.

- Odaberi jedinku iz populacije P koja će biti eliminirana koristeći selekcijski operator.

- Zamijeni eliminiranu jedinku s dijeteom u populaciji P.

Vrati najbolju jedinku iz populacije P kao rješenje

36. Generacijski genetički algoritam (Generational Genetic Algorithm) je algoritam evolucijske optimizacije koji koristi prirodnu selekciju, križanje i mutaciju kako bi evoluirao populaciju rješenja tijekom generacija.

Pseudokod:

Inicijalizacija populacije veličine N

Evaluacija svih jedinki u populaciji

Dok nije zadovoljen kriterij zaustavljanja:

- Selekcija roditelja za križanje
- Križanje roditelja i generiranje potomaka
- Mutacija potomaka
- Evaluacija potomaka
- Selekcija sljedeće generacije

Ispis najbolje pronađene jedinke

37. Jedinka u genetičkom algoritmu predstavlja jedno rješenje koje se sastoji od niza gena ili varijabli. Fenotip jedinke predstavlja njezina svojstva ili karakteristike koje se mogu izmjeriti i procijeniti, a proizlazi iz genotipa kroz proces genetskog izražavanja.
38. Genotip u genetičkom algoritmu predstavlja genetsku informaciju koja se prenosi s jedne generacije na drugu, a sastoji se od niza gena ili varijabli koje definiraju svojstva jedinke. Hromozom predstavlja nositelja genotipa i sastoji se od niza gena koji su poredani jedan za drugim. Svaki gen predstavlja jednu varijablu ili jedan dio informacije koja se prenosi na potomke.
39. Populacija u genetičkom algoritmu je skup jedinki koje sudjeluju u evolucijskom procesu, gdje svaka jedinka predstavlja jedno rješenje problema. Generacija je skup jedinki u populaciji u jednom trenutku. Nakon evaluacije svih jedinki, stvaraju se nove jedinke koristeći operatore genetskog križanja i mutacije, a proces se ponavlja dok se ne dostigne kriterij zaustavljanja.
40. Mravlji algoritam je metaheuristički algoritam inspiriran ponašanjem kolonije mrava pri traženju najkraćeg puta do izvora hrane. Pseudokod:
Inicijalizacija:
Generiraj početnu populaciju mrava na slučajan način
Postavi početne vrijednosti tragova na svim putevima na istu vrijednost
Ponavljaj sve dok uvjet zaustavljanja nije zadovoljen:
 - Mravi kreću iz svojih gnijezda i kreću se po putu u skladu s pravilima:
Ako je mrav na raskrižju, odaberi sljedeći put na temelju tragova i vjerojatnosti
Ako mrav dođe do kraja puta, osvježi trag na putu koji je prošao
 - Ažuriraj tragove:
Tragovi isparavaju s vremenom
Mravi ostavljaju tragove na putevima koje koriste
Jačina traga je veća ako je put uspješan (zadovoljio uvjete kvalitete rješenja)
Vrati najbolje rješenje koje su mravi pronašli