**SOFT KOMPJUTING**

1. Kada su konvolucione neuronske mreže stekle svoju popularnost? Koji su im uslovi to omogućili?

2012. godine CNN je prvi put pobedio na takmičenju ImageNet-a. Smanjio je grešku sa 25% na skoro 15%. Od tada uvek pobeđuju CNN. Napredku je doprioneo napredak GPU-ova koji se koriste za treniranje CNN i razvoj interneta za kreiranje velikih skupova podataka.

1. Opisati razliku između soft kompjutinga i hard kompjutinga.

Hard kompjuting: podrazumeva ekspertske sisteme, formalnu logiku, dokazivanje teorema,... Zahteva eksplicitno programiranje i tačno definisan ulaz, rezultuju preciznim rešenjima. Ponašanje je determinističko. Primer: ekspertski sistem za računanje poreza.

Soft kompjuting: predstavlja modelovanje ljudskog uma. Rezultuje približnim rešenjima, programi evoluiraju na iskustvu (podacima), ulazi su zašumljeni i dvosmisleni. Sistem ima stohastičko ponašanje. Primeri: prepoznavanje govora, procesiranje slika, computer vision, ...

1. Razlika deduktivnog i induktivnog zaključivanja

Deduktivno zaključivanje: top down pristup (od generalizacije ka specifičnoj instanci). Krećemo od permisa i primenom formalne logike dobijamo rezultat. Ukoliko su permise tačne i primena logike dobra, dobijamo zaključak koji je sigurno tačan. Kod ljudi se ovo koristi.

Induktivno zaključivanje: buttom up (od specifične instance ka generalizaciji). Vršeći što više eksperimenata pokušavamo da nađemo neku generalizaciju. Rezultat je verovatan, ne siguran. Ovo koristi mašinsko učenje.

1. Šta je nadgledano obučavanje (supervised learning)?

Nadgledano obučavanje je obučavanje kod kog imamo označene, tj. anotirane podatke. Kod njega nam je poznata rezultujuća labela y. Primer: obučavanje da li se na slici nalazi pas ili mačka ukoliko imamo za svaku sliku labelizovano šta se na njoj nalazi.

1. Koja su dva osnovna problema u okviru nadgledanog obučavanja?

Klasifikacija: imamo diskretno y, tj tačno određene klase. Primer da li se na slici nalazi pas ili mačka.

Regresija: imamo kontinualno y, tj bilo koju vrednost za y. Primer koliko će pasti kiše naredne nedelje.

1. Šta je nenadgledano obučavanje (unsupervised learning)?

Nenadgledano obučavanje je kada nemamo rešenje y, nego predviđamo koliko bi moglo biti u odnosu na sličnost sa podacima koje već imamo. Primer je klasterovanje gde posmatramo blizinu podataka (sličnost) i na osnovu toga odredimo kom klasateru pripada taj novi podatak.

1. Koje su tipične mere performanse koje se koriste za regresiju?

MSE (mean squared error): zbir kvadrata greške nad svim podacima. Poredimo koliko se naša predikcija razlikuje od stvarne vrednosti.

R2 (koeficijent determinacije): koliko smo varijanse u podacima pokrili, vrednost [0, 1].

1. Koje su tipične mere performanse koje se koriste za klasifikaciju?

Tačnost: koristi se kada imamo 2 klase i u svakoj jednak broj elemenata (pozitivni i negativni tvitovi).

Preciznost/odziv (precision/recall): procenat dobro anotiranih primera u odnosu na sve primere / procenat koliko smo primera pozitivne klase uspešno našli u odnosu na sve primere pozitivne klase. Teško je porediti dobijene rezultate.

F-mera: kombinacija preciznosti i odziva. Velika je samo ako su obe vrednosti velike.

1. Šta su hiper-parametri modela?

Hiper-parametri modela su parametri koji se ne uče u toku obučavanja modela, već ih unapred zadajemo. Primeri: learning rate, kod K-NN broj klasa K,... Otimizacija se vršti tako što imamo trening skup na kom obučavamo model, validacioni na kom utvrdimo tačnost koju dobijamo. Ukoliko nismo zadovoljni menjamo hiper-parametre pa tek nakon dobijanja određenih rezultata pređemo na test skup.

1. Objasnite kako se slika reprezentuje na računaru.

Digitalna reprezentacija slike podrazumeva da su slike diskretne u prostoru (rezolucija) i intenzitetu (broj nijansi). Intenzitet predstavljamo kao 1 (siva) ili 3 (rgb) matrice gde svaki piksel ima vrenost od 0-256 (8 bita).

1. Kako se video reprezentuje na računaru (dajte primer)?

Video se reprezentuje kao sekvenca slika (frame-ova, 1 ili 3 matrice) za određeni vremenski period (npr 30 frame-ova u boji za 1s -> 30\*3 matrice za 1s).

1. Opisati rasterski i vektorski model slike.

Rasterski model: deli sliku na elemente iste veličine i regularnog rasporeda (matrica piksela). Lakše dobiti, ali ako razvlačimo sliku moramo interpolirati vrednosti između poznatih pitksela.

Vektorski model: reprezentujemo sliku kao osnovne geometrijske oblike (∆, ●, ▪). Ima neograničenu rezoluciju, ali je problem biranje osnovnih oblika i načina dekompozicije.

1. Kakve slike postoje?

Binarne slike: vrednosti piskela su 0 ili 1.

Slike u nijansama sive (grayscale): vrednost piksela 0-256 (8 bita).

Multispektralna slika: vrednost piksela je vektor vrednosti 0-256 (za svaku RGB boji po jedna vrednost).

1. Kako možemo konvertovati RGB u grayscale?

Uprosečavanje piksela: srednja vrednost intenziteta boja, y = ( r + g + b ) / 3.

Metoda perceptivne osetljivosti: proporcionalno broju ćelija oka, y = 0.6 \* g + 0.3 \* r + 0.1 \* b.

1. Objasnite metod perceptivne osvetljenosti.

Metod se koristi pri konverziji slike iz RGB formata u nijanse sive. Formula za računanje nove vrednosti piksela je: y = 0.6 \* g + 0.3 \* r + 0.1 \* b. Ljudsko oko nema jednak broj ćelija osetljivih na sve boje (zelenu prepoznaje 60%, crvenu 30% i plavu 10% ćelija oka), zbog toga dobijamo one koeficijente za određivanje vrednosti piksela.

1. Koji modeli boja postoje? Objasnite ih i za svaki navedite predstavnike.

Aditivni (svetlosni) model: boju predstavljamo kao energiju svetlosnog izvora iliti intenzitet svetlosti. Koristimo intenzitete (određene talasne dužine) crvene, zelene i plave svetlosti. Minimum vrednosti svetlosti je potpuno tamna boja, crna - dobija se stavljanjem svih boji na minimum. Maksimum vrednosti je potpuno svetla boja, bela - sve boje na maksimum. Primeri: RGB (red, green, blue) i HVS (hue, saturation, value).

Suptraktivni (pigmentni) model: boja se posmatra kao reflektovane frekvencije svetlosti o posmatrani predmet. Predmet upije deo spektra, a reflektuje drugi. Tako reflekcijom svih frekvencija dobijamo crnu, a upijanjem svih frekvencija dobijamo belu. Primeri: CMY (cyan, magenta, yellow) i CMYK (cyan, magenta, yellow, key (crna)).

1. Objasnite RBG model boja.

RGB spada u adaptivni model boja. On je zasnovan na predstavi boje pomoću intenzinteta svetlosti. Koristimo intenzitete (određene talasne dužine) crvene, zelene i plave svetlosti. Minimum vrednosti svetlosti je potpuno tamna boja, crna - dobija se stavljanjem svih boji na minimum. Maksimum vrednosti je potpuno svetla boja, bela - sve boje na maksimum. Ove tri vrednosti određuju koordinate u 3D prostoru, a ponekad se dodaje i jedan kanal za providnost.

1. Objasnite HVS model boja o navedite primer kad je on bolji od RGB.

HVS spada u adaptivni model boja. On je zasnovan na predstavi boje pomoću intenzinteta svetlosti. Hue - ugao u stepenima, meren od vertikale, tj izbor boje. Saturation - količina sive u boji u procentima. Value - nijansa (intenzitet) sive u procentima. Primer: HVS je bolje koristiti ukoliko na primer treba da detektujemo šaku na slici pod različitim osvetlenjima, hue i saturation ostaju uvek isti, a value se menja (kod RGB-a se menja kompletna boja).

1. Kako možemo ukloniti šum sa slike?

Sabiranje: vršimo ga piskel po piksel.

Lokalno uprosečavanje: izaberemo klizajući prozor (lokalno okruženje oko posmatranog piksela) i računanjem proseka piksela dobijemo vrednost posmatranog piksela. Prozor pomeramo levo-desno i gore-dole, a dimenzije i oblik (krst, kvadrat,...) mogu biti različiti. Proces izaziva zamućenost, čija mera zavisi od klizajućeg prozora.

1. Navedite primere primene gde bismo mogli primeniti oduzimanje slika.

Oduzimanje slika se vrši na slikama iste dimenzije, vrednost rezultujućeg piksela jednaka je apsolutnoj vrednosti razlike piksela na odgovarajućim mestima originalnih slika. Ovo koristimo kada želimo da nađemo razliku među slikama.

Detekcija pokreta: ukoliko imamo stacionarnu kameru (video nadzor) i na statičnoj pozadini posmatramo da li ima promena. To koristimo pri detekciji pokreta, na primer: praćenje parkinga, praćenje broja ljudi koji ulaze ili izlaze iz radnji...

Praćenje pokreta kamere: pri snimanju filma.

Video kompresija: enkodiramo razlike frame-ova (isto frame posmatramo kao jedan, štedimo memoriju).

Posmatranje medicinskih snimaka: aktivnost mozga pre i posle primene nekog sredstva (proverava da li su krvni sudovi u mozgu aktivni).

1. Šta su Afine transformacije? Navedite primere primene.

Afine tranformacije su: skalirnaje, rotacije, translacija, krivljenje slike,... One se mogu primenjivati za promenu orjentacije slike (krivo uslikan dokument) ili augmentaciju skupa podataka.

1. Kako proveriti da li slika ima dobar kontrast?

Histogram je grafički prikaz brojnosti piksela određenih intenziteta. Na x-osi se nalaze intenziteti piksela, a na y-osi broj piksela. Ako na čitavom rasponu imamo jednako raspoređene vrednosti - slika ima dobar kontrast. Ukoliko su vrednosti koncentrisane oko određenog intenzitet - slika ima loš kontrast.

1. Objasnite kako dobijamo histogram slike.

Histogram je grafički prikaz brojnosti piksela određenih intenziteta. Na x-osi se nalaze intenziteti piksela, a na y-osi broj piksela. Za svako x imamo stubić koji određuje brojnost piksela tog intenziteta x.

Ako na čitavom rasponu imamo jednako raspoređene vrednosti - slika ima dobar kontrast. Ukoliko su vrednosti koncentrisane oko određenog intenzitet - slika ima loš kontrast.

Ako su vrednosti koncentrisane na levoj strani - slika ima malu osvetljenost. Ukoliko su vrednosti koncentrisane na desnoj strani - slika ima veliku osvetljenost.

1. Kako poboljšati kontrast slike?

Poravnanje histograma: modifikujemo broj piksela za svaki intenzitet tako da histogram ima veći kontrast. Drugi naziv je “razvlačenje” jer očuvamo oblik. Ovo je dobro koristiti kod slika gde su i pozadina i objekti previse svetli/tamni.

1. Šta je segmentacija slike pomoću praga (threshold)?

Segmentacija: postpak podele slike na regione sa sličnim atributima.

Segmentacija praga: najjednostavnije, prag (1+) poredimo sa osvetljenošću piksela pa piksel svrstamo u neku od kategorija (2+).

1. Koja je intuicija iza Otsu metode za pronalaženje globalnog praga?

OTS-u metoda: automatski pronalazi prag na osnovu “udolina” (jasno razdvojenih celina) histograma, min vrednosti udoline uzme za prag. Dobro je što je automatski. Problemi su: ukoliko tražimo globalni minimum na slici sa nejednakim osvetljenjem možemo naći lokalni minimum, kao i zanemarivanje prostorne raspoređenosti piksela.

1. Šta je globalni, a šta lokalni prag?

Globalni prag: imamo jedan jedini prag pomoću kog vršimo segmentaciju. Najlakši način jer poredimo sa samo jednim pragom vrednost svakog piksela. Ovo koristimo kod uniformno osvetljenih slika (mikroskopski snimci).

Lokalni prag: klizajući prozor prevlačimo preko slike i za svaki piksel posmatramo njegovu okolinu, odredimo vrednost praga i klasifikujemo u odnosu na njega. Ovo koristimo kod slika nejednake osvetljenosti.

1. Koji je nedostatak korišćenja jednog globalnog praga na celoj slici?

Nedostatak je što slike sa nejednakim osvetljenjem sve manje osvetljene regione svrsta ispod praga, a one više osvetljene iznad praga i iz tog razloga globalni prag koristimo kod uniformno osvetljenih slika, a ne kod prirodnih slika.

1. Objasnite postupak označavanja povezanih regiona na slici.

Primena: prebojavanje objekta na realnim slikama (ćelije na mikroskopskoj slici), određivanje pozicije objekta na slici, segmentacija niza karaktera na pojedinačne karaktere (prepoznavanje registarske tablice).

Varijanta 1: klasifikujemo piksel po piksel koristeći njegove susede (8 ili 4), klizajući prozor, usput beležimo i ekvivalentne kategorije. Na kraju, spajamo ekvivalentne kategorije.

Varijanta 2: krenemo od prvog obojenog piksela, stavimo ga u listu posećenih, pretražimo okolinu da bismo našli još isto obojenih i sve njih stavimo u listu otvorenih ivica. U svakoj iteraciji algoritma, probamo novi piksel iz liste otvorenih ivica. Za pretrage koristimo algoritme BFS i DFS.

1. Objasnite morfološku obadu slike.

Cilj morfološke obrade je da kod crno-bele (binarne) slike otklonimo lažno pozitivne i lažno negativne piksele. Lažno pozitivni su beli pikseli na pozadini (koja je inače crna, beli šum), a lažno negativni su crni pikseli na objektu od interesa (koji je inače beo, crni šum).

Morfološke operacije su skup operacija za obradu slike baziranim na oblicima. Osnovne su: dilacija i erozija. Kombinacijom ove dve dobiju se: otvaranje, zatvaranje i detekcija ivica.

1. Šta je dilacija?

Dilacija: osnovna binarna morfološka operacija. Predstavlja OR operaciju između piksela unutar objekta od interesa. Ona uklanja lažno negativne slučajeve (crni šum, popuni rupe u objektu).

1. Šta je erozija?

Erozija: osnovna binarna morfološka operacija. Predstavlja AND operaciju između piksela pozadine. Ona uklanja lažno positivne slučajeve (beli šum, popuni rupe u pozadini).

1. Šta je problem kod primene erozije i dilacije?

Pomoću dilacije uspešno uklanjamo neželjene procepe u objektima (crni šum), ali pri tome uvećavamo objekte, a pomoću erozije uklanjamo neželjene tanke linije i mostove (beli šum), ali pri tome smanjujemo objekte. Da bi dobili uspešno uklanjanje šumova i približno iste veličine objekata na slici, trebamo vršiti neke kombinacije dilacija i erozija, kao što su otvaranje i zatvaranje.

1. Kako možemo rešiti probleme erozije i dilacije?

Probleme erozije i dilacije možemo rešiti primenom nekih kombinacija ovih morfoloških operacija, kao što su otvaranje (E-D, eliminiši beli šum) i zatvaranje (D-E, eliminiši crni šum). Kod obe kombinacije koristimo isti strukturni element pri primeni i dilacije i erozije.

1. Koja je uloga i uticaj strukturnog elementa kod morfoloških operacija?

Strukturni element definiše veličinu i oblik problema na slici koji rešavamo. Veličinu treba odrediti u skladu sa veličinom objekta od interesa. Za velike objekte treba koristiti veće strukturne elemente, a za manje treba koristiti manje strukturne elemente, kako bi se očuvali strukturni detalji objekta. I kod otvaranja, i kod zatvaranja, jačina izglađivanja zavisi od veličine strukturnog elementa. Takođe, treba prilagoditi i oblik strukturnog elementa, gde poželjan oblik zavisi od geometrijskih oblika objekata na slici (npr. na medicinskim slikama treba koristiti oblik diska, jer uglavnom imamo krivudave objekte, a na satelitskim slikama treba koristiti oblik pravougaonika, jer imamo zgrade, puteve...).

1. Na koji način možemo izvršiti detekciju ivica pomoću morfoloških operacija?

Mogućnosti: (slika + dilacija) - slika, slika - (slika + erozija) ili (slika + dilacija) - (slika + erozija).

1. Objasnite operaciju konvolucije.

Kod konvolucije prevlačimo kernel (filter) preko slike, koristeći kombinaciju vrednosti odgovarajućih piksela i vrednosti u polju kernela dobijamo vrednost za piksel nove slike. Ovaj proces smanjuje veličinu slike pa možemo dodati redove nula ako bismo da sačuvamo originalnu veličinu. Rezultat ove operacije može biti: šiftovanje u levo, mean filter, izoštravanje,….

1. Šta je segmentacija slike? Šta je njen cilj? Kada smatramo da je segmentacija dobro izvršena?

Segmentacija je proces podele slike na smislene celine (npr po intenzitetu piksela, teksturi,…). Cilj je da pojednostavimo ili promenimo sliku u nešto smislenije i lakše za analizu. Dobro je izvršena ako u jednoj regiji imamo slične piksele (po nekom atributu), a pikseli drugih regija se razlikuju od njih.

1. Šta su izazovi kod segmentacije slike? Kako se segmentacija može evaluirati?

Kada ekspert odradi segmentaciju, to bi se verovatno razlikovalo od rezultata drugog eksperta jer su slike veoma kompleksne. Nećemo ići piksel po piksel nego ćemo vršiti segmentaciju baziranu na klasterima (k-means na primer). Tu klasifikaciju kasnije primenjujemo na krajnji cilj - da li npr softver detektuje kako treba pešake.

1. Objasnite k-means algoritam.

K-means algoritam radi tako da na slučajan način odredimo k centoida (centara klastera), njima prostorno najbliže piksele dodelimo, pomerimo centroid na centar njegovog klastera i ponovimo korake. Zaustavljamo se ako smo došli do konvergencije. Problem: ako vrednost piksela predstavlja samo boju, tada će se klasterovati po boji, ne i prostornoj povezanosti. Moramo definisati: metriku distance i broj klastera.

1. Objasnite ideju superpixels.

Algoritam superpikseli rešava problem klasterovanja piksela samo po boji. Sliku podelimo na polja rešetke. Na svakom polju, u centru odredimo da je centroid. Primenjujemo isti postupak kao kod k-means-a, ali moramo paziti na to da su prostorno ograničeni pikseli susedstva koje proučavamo (za to nam služi parameter m). Ukoliko je m malo više vrednujemo boju i intenzitet, a za veliko m vrednujemo prostornu povezanost. Takođe, ukoliko imamo puno superpiksela bolje ćemo pratiti ivice, a ukoliko je malo onda ćemo slabije. Potrebno je odrediti: metriku distance, m, broj superpiksela.

1. Objasnite segmentaciju pomoću rasta regiona (region growing).

Segmentacija pomoću rasta regiona: krenemo od jednog piksela, njegovoj regiji dodajemo sve okolne piksele koji su dovoljno slični sve dok ne ponestane sličnih piksela. Dalje biramo novu tačku i ponavljamo postupak. Pri gledanju suseda, posmatramo da li je već povezan sa regijom. Problem: može se desiti da jedna regija ima slabu tačku (ivicu) i da zbog toga “procuri” u drugu regiju.

1. Objasnite segmentaciju pomoću razdvajanja i spajanja regiona (split-and-merge).

Segmentacija pomoću razdvajanja i spajanja regiona podrazumeva da celokupnu sliku podelimo na 4 regiona, posmatramo da li su među njima neki slični (imaju isto uniformno obeležje), ako ih ima spajamo ih, te dalje dobijene regione svaki ponaosob delimo na 4 i ponavljamo postupak. Znamo da je kraj kada dođemo do toga da je uslov zaustavljanja ispunjen. Problem: blokovska raspodela (nema krivih ivica).

Potrebno odrediti: uniformnost (kriterijum da li je regija celovita), prag (kada da stanemo sa deljenjem regije) i kriterijum zaustavljanja (šta je kraj algoritma).

1. Objasnite ideju Watershed segmentacije.

Watershed segmentacija podrazumeva da sliku posmatramo kao 3D površinu sa dolinama i planinama, u dolinama se nalaze minimalne tačke, tačke na usponima se zovu Catchment basins i tačke na vrhu planine su watershed lines. Da su rupe u minimumina i da puštamo vodu, posmatramo šta se dešava, tražeći tačke na kojima se prvi put vode iz različitih dolina spajaju (watershed lines) i to interpretiramo kao branu - granice između objekata. Problem: ako imamo previše minumuma, ali možemo ih ograničiti ili specificirati.

1. Objasniti osnovnu ideju detekcije ivica na slici. Navedite neke operatore za detekciju ivica.

Detekcija ivica se radi na delovima slike gde se naglo menja intenzitet piksela. Posmatramo gradijent, što je on veći to se f-ja brže menja. Magnituda gradijenta, tj. dužina, označava jačinu ivice, a ugao gradijenta govori pod kojim je uglom ivica. Neki operatori za detekciju ivica su: Roberts-Cross operator, Sobelov operator i Canny.

1. Objasnite ideju Roberts-Cross operatora za detekciju ivica.

Koristimo dva kernel 2x2 zarotirana za 90°. Najbolje detektujemo dijagonalne ivice zbog toga, a na kraju ivice nisu toliko oštre. Problem je što je ovo jako osetljivo na šum jer posmatramo 2 susedna piksela, a ako je jedan šum to utiče mnogo. Dobro je jer je brzo i nema dodatne parameter za podešavanje.

1. Objasnite ideju Sobelovog operatora za detekciju ivica.

Koristimo dva kernel 3x3 zarotirana za 90° i strukture tako da najbolje detektuju horizontalne i vertikalne ivice. Kernele primenjujemo odvojeno, dobijamo 2 matrice čijom kombinacijom dobijamo magnitude gradijenta (jačina ivice). Problem: duže traje (više množenja) i šire su ivice (više izglađuje) zbog većeg kernel. Dobro: manje osetljivo na šum.

1. Objasnite ideju Canny detekcije ivica. Koje parametre ima ovaj algoritam i kako njih biramo?

Prvi korak je uklanjanje šuma Gausovim operatorom. Zatim se magnituda računa jednostavnim algoritmom, npr Roberts-Cross, dobijamo ivice. Zatim radimo non-maximum suppression, da bismo istanjili linije. Idemo po grebenovima, izbacujemo one piksele koji nisu na vrhu. Koristeći pragove T1 (svi preko su jake ivice) i T2 (svi ispod nisu ivice) da odredimo piksele jake, slabe ivice ili one koji nisu ivični. Za piksele slabih ivica uzimamo samo one koje su povezane sa jakim ivicama. Parametri: širina Gausovog kernel (veći manje osetljivo na šum, ali manje detalja), T1 (dovoljno veliko) i T2 (dovoljno malo).

1. Čemu služi Hough transformacija?

Hough transformacija nam od kompleksne ivice predstavljene pikselima omogućava da detektujemo neki matematički oblik (npr linija, krug, kvadrat,..). Hough transformacija se koristi u analizi slike, digitalnoj obradi slike i computer vision-u.

Prednosti: otporan je na šum i delimično zaklanja oblik.

Mane: memorijski zahtevan, spor (za složene oblike puno parametara i kombinacija) i možeš tražiti samo jednu vrstu oblika u jedom prolazu.

1. Izložite osnovnu ideju algoritma Hough transformacije.

Primenjujemo algoritam za traženje ivica (RobertCross, Sobel, Canny..). Transformišemo originalnu sliku u parametarski prostor (mapiramo piksele na ivice i snimamo u akumulator). Parametarski prostor predstavlja sve moguće varijante parametara a i b prave y=ax+b. Svaki piksel ivice glasa, a parametri sa najviše glasova predstavljaju ivicu. Beskonačan prostor delimo na konačan broj akumulatora, u njima predstavljamo ivice kao beskonačne linije. Na kraju konvertujemo beskonačne linije u konačne (čuvamo koordinate svih tačaka u akumulatoru ili pretražimo beskonačnu ivicu na binarnoj slici).

1. Koji su najčešći problemi kod klasifikacije slika?

Najčešći problem: varijacija u položaju kamere, varijacija u osvetljenju, deformacija objekta (različit položaj), zaklonjenost objekta, velika sličnost objekta i pozadine, varijacija u klasi (različit izgled objekata u istoj klasi), loš kvalitet slike...

1. Dajte konceptualni pregled sistema za klasifikaciju slika. Kako se deep learning pristup razlikuje?

Tradicionalni pristupi podrazumevaju sva 4 koraka, a deep learning pristupi preskaču 2. korak.

Prvi korak je pretprocesiranje slika (npr normalizacija vrednosti piksela, poboljšanje kontrasta,…). Bitno je u ovom koraku svesti sve slike na istu veličinu. Drugi korak je ekstrakcija obeležja (izbacivanje obeležja koja nam nisu bitna za klasifikaciju). Ovaj korak je u tradicionalnim pristupima jako bitan za performanse, ljudi određuju koja su bitna obeležja (ali ovo je sklono greškama ili ljudi ne znaju da se izraze kako treba). Takođe za svaki skup podataka su nam bitne druge osobine, te je bolje da algoritam sam zaključi koje su to osobine (deep learning tako radi). Treći korak je primena obučavajućeg algoritma (npr k-NN, SVM, CNN,...), a za to nam je potreban trening skup. Četvrti je dodela labela.

1. Objasnite/skicirajte osnovnu ideju SVM algoritma.

Ideja SVM algoritma je da nađemo hiper-ravan koja razdvaja dve grupe tačaka tako da bude maksimalno udaljena od svih primera iz trening skupa. Margina hiper-ravni je minimum rastojanja od te hiper-ravni do neke tačake skupa podataka. SVM bira upravo optimalnu hiper-ravan, tj. onu sa najvećom marginom. Pronađena hiper-ravan je najrobusnija od svih mogućih linija koje idealno razdvajaju tačke. Ukoliko nemamo linearno separabilne podatke onda možemo transformisati podatke u prostor gde će biti linearno separabilni.

1. Objasnite čemu služi i kako se konstruiše HOG (Histogram of Oriented Gradients) deskriptor.

HOG deskriptor sliži za ekstrakciju obeležja (na osnovu posmatranja histograma smera ivica, tj smer orjentacije gradijenta). Računamo gradijent (orjentaciju i magnitudu) po x i y osi (ukloni obeležja). Delimo sliku na NxN ćelija i za svaku računamo histogram (da bi bili otporni na šum). Normalizujemo podatke (jer je histogram osetljiv na osvetljenje) nad blokom ćelija (da ne izgubimo oblik). Zatim radimo konkatenaciju dobijenih vektora, a sa tom vektorskom reprezentacijom slike treniramo model i vršimo klasifikaciju.

1. Algoritam mašinskog učenja za prepoznavanje objekata na slikama.

Algoritam mašinskog učenja, da bi radio na slikama različite veličine, moramo primeniti selektovanje isečka na kom je najverovatnije da se nalazi objekat (klasifikator nam to izračuna). Algoritmi su: sliding window i region proposal.

1. Objasnite kako funkcioniše i čemu služi sliding window pristup u detekciji objekta na slici.

Sliding window je algoritam koji koristi prozor koji se pomera na slici levo-desno i gore-dole. To podrazumeva mnogo pomeranja, takođe objekti mogu biti preveliki ili premali pa treba skalirati sliku ili u slučaju kada je objekat različitih proporcija moramo menjati i proporcije prozora. Mana je sporost jer ima mnogo mogućnosti za pomeranja, skaliranje i proporcije.

1. Objasnite kako funkcioniše i čemu služi region proposal pristup u detekciji objekta na slici.

Region proposal radi tako što koristimo neki od algoritama za grupisanje međusobno sličnih piksela (segmentaciju, npr selective search). Nije problem ako imamo false-positive (problem su false-negative) slučajeve, bitno je da je brzo i da neki od pronađenih regiona zaista sadrži objekat. Tada primenimo HOG deskriptor da od tih regiona odaberemo odgovarajući. I na kraju primenimo klasifikacioni model.

1. Objasnite Hammingovu distancu za poređenje stringova.

Hammingova distanca je metod poređenja dva string ISTE dužine (to je mana) po tome koliko se karaktera razlikuje. Porede se bitovi na svakoj odgovarajućoj poziciji i rezultat je broj bitova koji se razlikuju (npr za MACKA i MAČKA dobije broj 1). Alternativa Hammingove distance je Levenshtein distanca.

1. Objasnite Levenshtein distancu za poređenje stringova.

Levenshtein distance se koristi za poređenje 2 stringa (ili bilo šta predstavljeno kao niz slova), ali ne obavezno iste dužine. Distanca je jednaka ceni da se jedan string pretvori u drugi (npr za MACKA u MAČKA je cena(C-Č) ili za MAČ u MAČKA je cena(+K)+cena(+A)). Cene se mogu određivati na različit način, ali najbolje je da to bude neka predefinisana ili vrednost laka za računanje (npr blizina na tastaturi, cena za samoglasnike ili suglasnike,...). Levenshtein distanca se može koristiti za: spelling correction, traženje duplikata u registru ulica, sekvenciranje proteina, poređenje pesama ptica...

1. Definišite Euklidsku i kosinusnu udaljenost (dovoljno je intuitivno objašnjenje, ne mora formula).

Euklidska udaljenost: razlika vektora po magnitudi [dužina dokumenata, u KNN ili k-means algoritmu]. Sve dimenzije su podjednako važne (eventualno im se mogu dodeliti težine, ali retko se to praktikuje).

Kosinusna udaljenost: ugao između normalizovanih vektora (razlika u orjentaciji) [sadržaj dokumenata].

1. Šta sve treba uzeti u obzir prilikom odabira mere sličnosti/udaljenosti objekata?

Za sličnost ili udaljenost objekata moramo definisati: meru (euklidsku, kosinusnu, manhattan,..), koji problem rešavamo i kako ćemo predstaviti objekte (sirova reprezentacija piksela [ok detekcija brojeva, ali ne i lica] ili nešto treće).

1. Objasniti K-NN algoritam.

K-NN algoritam radi tako što imamo l labeliranih podataka (ili pripadaju klasi A ili klasi B). Za podatak c treba da odredimo da li pripada klasi A ili B, to radimo na osnovu K suseda (to je K podataka najbližih po euklidskoj meri). Dodeljujemo klasu kojoj pripada većina suseda podatka c. Hiper-parametar je broj suseda K, njega odredimo ispobavanje (K=1 pa K=2 itd na validacionom skupu). Ukoliko je preveliko K može doći do underfittinga, a za malo K do overfittinga.

1. Objasniti matematički model veštačkog neurona.

Veštački neuron ima ulaze xi koje množimo težinama wi (kontrolišu koliko jačina i smer jednog neurona utiču na drugi). Zatim sabiramo sve sum(xi\*wi) i dodajemo b (bias). Propustimo rezultat kroz aktivacionu funkciju f (odredi nivo pobuđenosti neurona za date ulaze).

1. Zašto kombinujemo neurone (šta bismo mogli postići samo jednim)?

Kombinacijom neurona dobija nelinearne granice odluke (npr kružnica), a samo jednim linearnu granicu (prava linija).

1. Objasnite feedforward model neuronske mreže (za klasifikaciju i regresiju).

Feedforward model neuronske mreže podrazumeva da su neuroni jednog sloja jedino povezani sa narednim (nema veza unazad, veza koje preskaču slojeve ili rekurentnih veza). Ne koristi se aktivaciona funkcija u izlaznom sloju. Kod klasifikacije imamo k izlaza (k = broj klasa), a kod regresije imamo 1 izalaz.

1. Čemu služi aktivaciona funkcija?

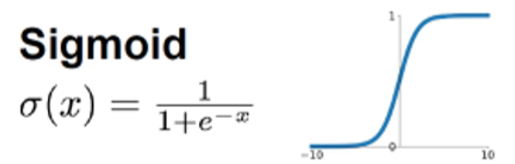
Aktivaciona funkcije doprinosi da izlaz iz neuronske mreže dobije nelinearnost (u suprotnom bi bio linearna funkcija - tj imali bismo veoma kompleksnu linearnu regresiju).

1. Konceptualno objasniti kako se trenira neuronska mreža.

Neuronska mreža se trenira tako što imamo test podatke i određene labele za svaki. U procesu treninga pronalazimo vrednosti za težine i bias-e. Ove vrednosti korigujemo dok ne dobijemo najmanju grešku modela. Za minimizovanje greške sistema koristimo algoritam gradijentnog spusta (gradient descent). Za regresiju koristimo mean sqared error (suma kvadrata razlika dobijene i očekivane vrednosti). Za klasifikaciju koristimo log loss (za malu verovatnoću stvarne klase dobijemo veliki log loss) - obično nam te vrednosti daje softmax algoritam na kraju (transformiše nenegativne izlazne signale i pobrine se da im suma bude 1).

1. Navedite barem tri poznate aktivacione funkcije (i skicirajte ih), navedite njihove prednosti i mane.

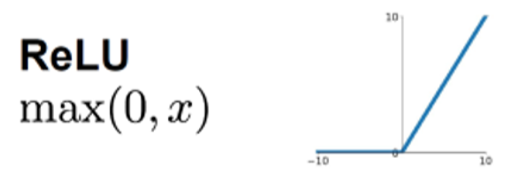
Sigmoid mapira vrednosti na interval [0, 1]. 0 - nema sigmala, 1 - puna saturacija signala. Mana: nije centrirana oko 0, „ubija“ gradijente saturisanih neurona i skupa je za računanje.



Tanh mapira vrednosti na interval [-1, 1]. Vrednosti su centrirane oko 0. Mana: „ubija“ gradijente saturisanih neurona.



ReLU najbolje simulira rad bioloških neurona. Jako brzo konvergira. Mana: nije centrirana oko 0, tokom treninga neki neuroni mogu da „umru“ (ako imamo veliki learning rate i lošu inicijalizaciju pa nikad ne aktiviramo neuron).



1. Objasniti o čemu treba voditi računa prilikom primene neuronskih mreža.

Treba voditi računa o: pretprocesiranju podataka (centriranje oko nule [obeležje - sr\_vr\_obeležja] i normalizacija [obeležje / standardna\_devijacija\_obeležja]), izboru aktivacione funckije (ugl ReLU ili neka njena modifikacija) i o inicijalizaciji težina (težine odrede koliko je koji neuron bitan pa ako su 0 svi su podjednaki, najbolje je da se težine i bias-i budu slučajne vrednosti).

1. Zašto na problemu klasifikacije slika koristimo konvolucione neuronske mreže (CNN)?

Slike imaju veliki broj parametara pa bi obična neuronska mreža pri treningu brzo overfittovala, a i nema računarskih i memorijskih resursa koji bi to treniranje izdržali.

1. Objasniti osnovni model konvolucionih neuronskih mreža (CNN).

Konvolucione neuronske mreže:

Ulaz: 3D slika (visina, širina i tri sloja za boju).

Više konvolucionih slojeva: na sliku primenimo f filtera. Svaki filter ima dubinu kao ulazna slika, a ostale dimenzije su hiper-parametri. Za svaku poziciju računamo 1 vrednost na osnovu skalarnog proizvoda težina filtera i vrednosti piksela. Za svaki filter dobijamo aktivacionu mapu pa na kraju imamo „sliku“ sa f dubinom. Možemo zatim upotrebiti aktivacionu funkciju (npr ReLU). Takođe, opciono je da li će se zatim iskoristiti pooling sloj (da bi se izdvojila najbitnija obeležja, smanji se dimenzija, ali ne i dubina slike).

Više potpuno povezanih slojeva: kao i kod obične nm, ali imamo jako malo obeležja. Poslednji sloj se razdvaja na vektor obeležja.

Izlaz: vektor obeležja.

1. Kako se CNN obučavaju?

CNN se obučavaju algoritmom back-propagation, tj pokušavamo da odredimo težine filtera koje daju najmanju grešku. Za proračun greške poredimo dobijenu sa tačnom vrednošću. Za minimizaciju greške koristimo algoritam gradijentnog spusta.

1. Objasniti konvolucioni sloj u CNN.

Konvolucioni slojevi: na sliku primenimo f filtera. Svaki filter ima dubinu kao ulazna slika, a ostale dimenzije su hiper-parametri. Prevlačimo ih levo-desno i gore-dole, imamo podesiv korak. Za svaku poziciju računamo 1 vrednost na osnovu skalarnog proizvoda težina filtera i vrednosti piksela. Za svaki filter dobijamo aktivacionu mapu pa na kraju imamo „sliku“ sa f dubinom kao izlaz.

1. Na koji način smanjujemo broj slobodnih parametara u CNN mreži?

Broj parametara CNN-a se smanjuje: promenom koraka za koji pomeramo filtere (moramo paziti da se uklopi korak, veličina filtera i veličina slike) ili primenom pooling sloja (izdvajamo bitna obeležja i smanjujemo širinu i visinu).

1. Šta je i čemu služi zero padding u CNN?

Zero padding znači da ulaznu sliku u konvolucioni sloj proširujemo sa nulama. Time postižemo da se primemom konvolucionih filtera ne smanjuju visina i širina slike.

1. Čemu služe slojevi sažimanja (pooling)? Dajte primer pooling sloja.

Sloj sažimanja (pooling) služi da se izdvoje najbitnija obeležja, smanji se visina i širina, a ostaje ista dubina slike. Primer je Max Pooling (posmatra regije i odabira najveću vrednost koja će reprezentovati tu regiju).

1. Šta je i čemu služi transfer learning?

Transfer learning: naučimo važna obeležja (npr ivice) i poslednji slojevi se dotreniraju na osnovu njih. Prednost je da nam za trening treba manje podataka, a neće doći do overfittinga.

1. Šta predstavlja semantička segmentacija slike (semantic segmentation)?

Semantička segmentacija: svakom pikselu slike dodelimo neku predefinisanu kategoriju (cilj je da imamo regije sa isto anotiranim pikselima).

1. Kakav nam je trening skup potreban da bismo vršili semantičku segmentaciju slika pomoću CNN?

Semantička segmentacija pomoću CNN zahteva da imamo trening skup u kojem je svaki piksel svake slike anotiran, te su ovakvi trening skupovi jako skupi.

1. Da li se u semantičkoj segmentaciji pravi razlika između različitih instanci objekata iste kategorije?

Semantička segmentacija vodi računa o anotaciji piksela, a segmentacija instanci vodi računa o instancama iste klase.

1. Opisati ideju semantičke segmentacije (semantic segmentation) primenom CNN?

Semantička segmentacija pomoću CNN radi tako što predviđa klase svih piksela odjednom. Izlaz CNN je slika iste širine i visine, ali ima dubinu jednaku broju kategorija. Softmax algoritam za svaki piksel odredi kojoj od kategorija pripada (na osnovu vrednosti iz izlaza CNN-a dobije verovatnoće za sve kategorije i odabere najveću). Gubitak računamo cross-entropy loss funkcijom: za svaku sliku ponaosob i za ceo trening skup prosečan gubitak. Model se trenira kao i CNN i nm back-propagation algoritmom (ali to je jako sporo jer svaki konvolucioni sloj održava veličinu slike zero padding-om) ili koristeći down i upsampling (prvo koristimo downsampling [pooling] da smanjimo sliku i zatim upsampling [unpooling] da povećamo sliku pred izazak iz CNN).

1. Objasnite problem klasifikacije i lokalizacije (bounding box).

Klasifikacija i lokalizacija: pronalaženje tačno jednog objekta na slici i njegova klasifikacija. CNN ima istu arhitekturu samo što vraća dva izlaza: rezultat za klasifikaciju i vrednosti za bounding box (x, y, w, h - x i y gornji levi ugao i w širina i h visina). Najbolje je trenirati prvo klasifikaciju pa lokalizaciju (ili obrnuto) pa posle spojiti kao jedan sistem. Imamo dve greške: classification loss (provera dobijene i tačne klase, npr softmax) i regression loss (provera dobijenih (x,y,w,h) i tačnih, npr L2). Na kraju se ovi gubici sabiraju (loše jer jedan može da izdominira).

1. Objasnite problem detekcije objekata.

Detekcija objekata: podrazumeva da nađemo sve moguće (0+) bounding box-ove unutar kojih ćemo prepoznati kategoriju objekta. Imamo dva algoritma: region proposals i pristup bez predlaganja regija.

Region proposals algoritmi: R-CNN (klasična imeplementacija RP-a), Fast R-CNN (cela slika kroz CNN pa zatim RP) i Faster R-CNN (prvo slika kroz CNN zatim koristimo mrežu za RP). RP traži regione kandidate selective search-om, zatim to propusti kroz CNN i dobije klasu i bounding box. Ovo je tačniji algoritam. R-CNN problemi: bounding box-ovi su različite veličine i potencijalno se preklapaju, sporo i skupo. Fast R-CNN problemi: veličina bounding box-ova varira [rešenje: reshape].

Pristupi bez predlaganja regija: YOLO i SSD. Sliku delimo na mrežu ćelija, u svakoj zamislimo osnovne bounding box-ove i za njega odredimo klasu i lokalizaciju objekta. Na osnovu datih rezultata vršimo predviđanje za celu sliku. Ovo je brži algoritam.