SOFT KOMPJUTING

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

Od 2009. godine, ImageNet održava takmičenje „Large Scale Visual Recognition Challenge“. Godinama je greška rezultata na ovom takmičenju bila oko 25%, ali je u 2012. spala za gotovo 10%. Od tada je svaki pobednik ImageNet takmičenja bila neuronska mreža. Danas su rezultati još bolji, ali je ovaj pad veoma važan, jer je pobednički algoritam te godine bila konvoluciona neuronska mreža. Iako su CNN tek 2012. imale veliki proboj, one nisu izmišljene tad, već 1998. godine. CNN su postale popularne tek od 2012. iz dva razloga: brzina računara i mogućnost paralelizacije, i količina labeliranih podataka. Po Murovom zakonu, dobijali smo sve brže i brže računare svake godine, a posebno je doprineo razvoj GPU-ova, koji se mogu maksimalno paralelizovati i koji su se pokazali kao izuzetno koristan alat za treniranje CNN-a. Zahvaljujući povećanju računarske moći, istraživačima je dozvoljeno da eksperimentišu sa sve većim modelima. Količina labeliranih podataka je znatno povećana u odnosu na 90-e pre svega zbog razvoja podrške interneta, koja je omogućila kreiranje velikih i raznovrsnih skupova podataka.

2. Opisati razliku između soft kompjutinga i hard kompjutinga. Navedite gde se primenjuje soft, a gde hard kompjuting.

Hard kompjuting podrazumeva: ekspertske sisteme, formalnu logiku, dokazivanje teorema, i slično, odnosno tu spadaju problemi koji su intelektualno zahtevni za čoveka. On zahteva eksplicitno programiranje i tačno definisan ulaz, a rezultuje preciznim rešenjima i njegovi sistemi imaju deterministično ponašanje. Primeri primene hard kompjutinga su TurboTax ekspertski sistem za računanje poreza u SAD-u, ili neki sistemi za dokazivanje formalne logike. Soft kompjuting ima za cilj modelovanje ljudskog uma koji ima toleranciju na nepreciznost. Realan svet je zašumljen i ima kontinualne vrednosti, a računarstvo je precizno i određeno, i ima binarni princip true/false. Pomoću soft kompjutinga želimo da primenimo računarsko rezonovanje na realan svet. Soft kompjuting rezultuje približnim rešenjima, njegovi programi evoluiraju sa iskustvom, tj. sa podacima, ulazi su dvosmisleni i zašumljeni, i sistemi imaju stohastičko ponašanje. Ovakve sisteme ne programiramo eksplicitno, već ih obučavamo na osnovu podataka. Primeri primene soft kompjutinga su: prepoznavanje govora, image processing (kolorizacija, uklanjanje šuma, kompresija, i slično), computer vision (klasifikacija slika, detekcija objekata, prepoznavanje aktivnosti, image captioning, i slično)...

3. Razlika deduktivnog i induktivnog zaključivanja

Deduktivno zaključivanje je top down pristup zaključivanja, gde top predstavlja generazilaciju, a down specifičnu instancu, dok je induktivno zaključivanje bottom up pristup. Ideja kod deduktivnog zaključivanja je da od generalnog idemo ka specifičnom, gde krećemo od nekih premisa i primenjujemo formalnu logiku. Ako su premise tačne i ako smo pravilno primenili pravilo formalne logike, zaključak će sigurno biti tačan. Kod induktivnog zaključivanja idemo obrnuto, tj. krećemo se od specifičnog i pokušavamo da generalizujemo. Trudimo se da izvršimo što više eksperimenata, posmatramo njihove rezultate i tražimo neke generalne šablone koji postoje u onome što smo uključili. Induktivno zaključivanje podrazumeva da na osnovu prikupljenih opservacija vršimo generalizaciju, tj. dolazimo do verovatnog zaključka. Deduktivno zaključivanje je jedan od osnovnih načina zaključivanja kod ljudi, a induktivno zaključivanje je način na koji učimo uz pomoć mašinskog učenja.

4. Šta je nadgledano obučavanje (supervised learning)? Dajte primer nadgledanog obučavanja. Objasnite zašto taj problem spada u nadgledano obučavanje.

Nadgledano obučavanje je obučavanje kod kog imamo označene, tj. anotirane podatke. Kod njega nam je poznata ciljna labela y. Primer nadgledanog obučavanja je klasifikator da li se na slici nalazi mačka ili pas. Ovo spada u nadgledano obučavanje jer je skup podataka anotiran, odnosno za svaku sliku unutar skupa podataka imamo oznaku da li se na toj slici nalazi mačka ili pas.

5. Koja su dva osnovna problema u okviru nadgledanog obučavanja? Po čemu se razlikuju? Dajte po jedan primer za svaki problem i objasnite zašto se svrstava u taj problem.

Dva osnovna problema u okviru nadgledanog obučavanja su klasifikacija i regresija. Razlika između ova dva problema je priroda y, gde kod klasifikacije imamo diskretno y, tj. imamo diskretan broj klasa, a kod regresije imamo kontinualno y.

Primer za klasifikaciju bi mogao biti klasifikator da li se na slici nalazi mačka ili pas, gde sliku svrstavamo u jednu od dve klase, ili OCR za prepoznavanje ručno napisanih cifara, gde sliku svrstavamo u jednu od deset klasa. Primeri za regresiju bi bili predviđanje cene nekretnine, predviđanje količine kiše koja će pasti u narednoj nedelji, i slični problemi koji imaju kontinualnu vrednost.

6. Šta je nenadgledano obučavanje (unsupervised learning)? Dajte primer nenadgledanog obučavanja. Objasnite zašto taj problem spada u nenadgledano obučavanje.

Nenadgledano obučavanje je obučavanje kod kog imamo neoznačene podatke. Kod njega nemamo y, već tražimo neku strukturu u podacima, odnosno iz blizine podataka određujemo koliko su slični. Pod blizinom podataka se podrazumeva neka mera sličnosti podataka i prema tome određujemo koliko su neki podaci međusobno „udaljeni". Primer nenadgledanog obučavanja je klasterovanje, što spada u ovaj način obučavanja jer nemamo anotirane podatke, tj. nedostaje nam y, a treba da vršimo obučavanje.

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

Tipične mere performanse koje se koriste za regresiju su: MSE (Mean Squared Error) i R^2 (koeficijent determinacije). MSE predstavlja zbir kvadrata greške nad svim primerima. Ideja ove mere je dati predikciju i posmatrati koliko se ona razlikuje u odnosu na tačnu vrednost. Pošto je kontinualna varijabla u pitanju, posmatramo pozitivnu razliku između te dve vrednosti, odnosno kvadriramo dobijenu razliku. Kod MSE je u nekim slučajevima teže odrediti da li je dobijena vrednost velika ili mala greška, pa tada treba koristiti koeficijent determinacije, jer se on kreće između 0 i 1, pa je lakše otkriti smisao dobijenog rezultata. Koeficijent determinacije nam govori koliko varijanse u podacima smo pokrili.

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

Tipične mere performanse koje se koriste za klasifikaciju su: tačnost (accuracy), f-mera i preciznost/odziv (precision/recall). Tačnost je procenat uspešnih predikcija u odnosu na ukupan broj predikcija. Ova mera je odlična ukoliko imamo balansirane klase (npr. ako klasifikujemo tweet-ove u pozitivne i negativne, trebali bi imati jednak broj pozitivnih i negativnih tweet-ova u test skupu). Zavisno od problema, tačnost ne mora biti dobra mera, što može biti slučaj ukoliko imamo nebalansirane klase (npr. ljudi koji boluju od raka i ljudi koji ne boluju od raka) ili višekategorijsku klasifikaciju. U ovim slučajevima se obično koriste precision/recall ili f-mera. Kod precision/recall preciznost je procenat dobro anotiranih primera pozitivnom klasom u odnosu na sve primere, a odziv je procenat koliko smo primera pozitivne klase uspešno pronašli u odnosu na sve primera pozitivne klase. Problem kod precision/recall-a je što poređenje dobijenih rezultata može biti teško, jer imamo dva broja na koja treba da obratimo pažnju. Tada je bolje koristiti f-meru, koja predstavlja kombinaciju vrednosti preciznosti i odziva. Kod nje posmatramo samo tu jednu kombinovanu vrednost i ona će biti visoka samo ako su i preciznost i odziv visoki.

9. Šta su hiper-parametri modela? Kako možemo izvršiti optimizaciju hiper-parametara i testiranje modela?

Hiper-parametri modela su parametri koji se ne uče u toku obučavanja modela, već ih unapred zadajemo. Primer ovakvog parametra je learning rate. Problem kod hiper-parametara je što oni veoma zavise od problema. Težine i bias-e neuronske mreže možemo da učimo direktno iz podataka uz gradient descent optimizaciju, dok learning rate ne možemo. Gotovo svaki model ima hiper-parametre koje treba optimizovati (npr. kod K-NN treba da odredimo optimalno K, kod SVM treba da odaberemo C i γ...). Da bi optimizovali hiper-parametre i testirali model, ideja je da podatke podelimo na trening, validacioni i test skup. Trening skup nam služi za treniranje modela i optimizaciju parametara koji se uče iz podataka (npr. težine, bias-i), validacioni skup nam služi za optimizaciju hiper-parametara, a test skup nam služi za evaluaciju performansi modela. Hiper-parametre biramo isprobavanjem (npr. za K-NN postavimo da je K=1, treniramo na trening skupu, testiramo na validacionom, pa promenimo da je K=2, treniramo na trening skupu, testiramo na validacionom, itd. sve dok ne dođemo doo K koje daje najveću tačnost. Tada testiramo taj model na test skupu).

10. Objasnite kako se slika reprezentuje na računaru. Koje vrste diskretizacije moramo da izvršimo da bismo to uradili (dajte primer diskretizacije)?

Digitalna reprezentacija slike podrazumeva da su slike diskretne, i imamo dve vrste diskretizacije: u prostoru i u intenzitetu. Oba nivoa diskretizacije su veoma važna za kvalitet slike. Prostorna diskretizacija predstavlja rezoluciju, a diskretizacija intenziteta, tj. broj nijansi, predstavlja koliko imamo nijansi boje po pikselu. Obično se za broj nijansi uzima intenzitet svetlosti za svaki piksel, odnosno jedna od 256 (8 bita) nijansi sive. Ukoliko nemamo grayscale sliku, već sliku u boji, ako svaki od tri kanala koristi 8 bita da opiše intenzitet boje, onda je ukupno 24 bita posvećeno svakom pikselu. Primer diskretizacije je da imamo sliku rezolucije 256x256, gde svaki piksel može biti reprezentovan uz pomoć 8 bita, što nam daje 256 mogućih nijansi sive, pa svaki od 256x256 piksela može imati jednu od mogućih 256 nijansi. Slika je u memoriji računara smeštena kao matrica, gde je svaki element te matrice piksel koji ima jednu od 256 nijansi, ako pričamo o grayscale slikama. U slučaju slika u boji, neće svaki piksel imati jednu vrednost, nego vektor nekih vrednosti (npr. za RGB ćemo imati tri kanala, tj. tri matrice - za intenzitet crvene, zelene i plave boje).

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

Video se na računaru reprezentuje kao sekvenca slika. Kod reprezentacije videa, za određeni vremenski rok imamo određenu sekvencu slika. Obično imamo 30 frame-ova (slika) po sekundi. Ako su u pitanju sekvence grayscale slika, imaćemo jednu matricu po frame-u, a ako su u pitanju slike u boji, onda ćemo imati tri matrice po frame-u, po jedna za intenzitet svake od boja (RGB). Primer bi bio da imamo video u boji za koji nam je za jednu sekundu videa neophodno 30 frame-ova, tj. 90 matrica (30x3), gde je npr. svaka matrica rezolucije 512x512 i svaki piksel unutar matrice je reprezentovan sa 8 bita. Iz ovog primera se vidi da nam je za samo jednu sekndu videa neophodno mnogo podataka.

12. Opisati rasterski i vektorski model slike.

Rasterski model slike reprezentuje sliku piksel po piksel, tj. deli sliku u elemente iste veličine i regularnog rasporeda. Kod vektorskog modela slike pokušavamo da dekomponujemo sliku u osnovne geometrijske oblike (npr. trouglovi, krugovi, kvadrati...). Rasaterski model je jednostavnije dobiti, jer upravo on predstavlja sliku koju uslikamo fotoaparatom, ali je kod njega problem što kod razvlačenja slike moramo vršiti interpolaciju kako bi znali koju vrednost piksela treba da postavimo između poznatih piksela koje smo uslikali. Vektorski model zahvaljujući geometrijskim oblicima ima neograničenu rezoluciju, ali je njegov najveći problem biranje osnovnih oblika i načina dekompozicije.

13. U zavisnosti od podataka koji se nalaze u prostornim koordinatama (pikselima), kakve slike postoje?

U zavisnosti od podataka koji se nalaze u pikselima, digitalne slike možemo podeliti na: binarne slike, slike u nijansama sive (grayscale images) i multispektralne slike. Binarne slike su slike koje za vrednosti piksela imaju samo vrednosti 0 ili 1. Grayscale slike kao vrednost piksela imaju jednu od nijansi sive, tj. vrednost intenziteta za svaki piksel (npr. sa 256 nijansi sive reprezentujemo nivo osvetljenja). Multispektralne slike, tj. slike u boji, kao vrednost piksela imaju vektor vrednosti intenziteta, gde obično imamo vektor od tri elementa (jedna element za svaki od kanala - RGB).

14. Kako možemo konvertovati RGB u grayscale?

RGB u grayscale se može konvertovati računanjem srednje vrednosti RGB komponenti, tj. prostim uprosečavanjem piksela, ili pomoću metoda perceptivne osvetljenosti. Srednja vrednost RGB komponenti se računa kao $Y=((R+G+B))/3$, a metod perceptivne osvetljenosti kao $Y=0.229\cdot R+0.578\cdot G+0.114\cdot B$. Ljudska percepcija boja nije direktno proporcionalna intenzitetu svetlosti određene talasne dužine, jer se broj ćelija osetljivih na određene talasne dužine razlikuje (60% zelena boja, 30% crvena, 10% plava). Iz ovoga se dobijaju koeficijenti kojima se množi intenzitet boja u formuli za metod perceptivne osvetljenosti, i oni odgovaraju udelu prijemnika u ljudskom oku za svaku od tri boje. Metod perceptivne osvetljenosti daje bolje rezultate u odnosu na prosto uprosečavanje piksela.

15. Objasnite metod perceptivne osvetljenosti.

Metod perceptivne osvetljenosti je jedan od načina za konvertovanje RGB slike u grayscale. Kod njega se vrednost svakog piksela dobija pomoću formule $Y=0.229\cdot R+0.578\cdot G+0.114\cdot B$. Ljudska percepcija boja nije direktno proporcionalna intenzitetu svetlosti određene talasne dužine, jer se broj ćelija osetljivih na određene talasne dužine razlikuje (60% zelena boja, 30% crvena, 10% plava). Iz ovoga se dobijaju koeficijenti kojima se množi intenzitet boja u formuli za metod perceptivne osvetljenosti, i oni odgovaraju udelu prijemnika u ljudskom oku za svaku od tri boje.

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

Modeli boja koji postoje su: aditivni (svetlosni) model i suptraktivni (pigmentni) model. Aditivni model je model gde boju doživljavamo kao energiju svetlosnog izvora, tj. gde boju percipiramo kao zbir intenziteta svetlosti. Aditivnim mešanjem svetla određene talasne dužine i intenziteta možemo iz osnovnih komponenti crvene, zelene i plave postići ostale nijanse spektra. Svaka boja može da varira između minimuma (potpuno tamna) i maksimuma (potpuno osvetljena), pa se zato za odsustvo intenziteta sve tri boje dobija crna, a za njihov maksimalan intenzitet se dobija bela. Suptraktivni model se bazira na činjenici da je boja objekta energija svetlosti koja se reflektovala o dati objekat, tj. on podrazumeva da kako dodajemo pigmente, tako pigment upija određene frekvencije svetlosti, a određene reflektuje. Mi vidimo samo one frekvencije koje su reflektovane o površinu objekta. Za maksimalan intenzitet svih boja ćemo kod suptraktivnog modela imati crnu boju, a za minimalan ćemo imati belu. Predstavnici aditivnog modela su RGB (red, green, blue) i HSV (hue, saturation, value), a predstavnik suptraktivnog modela je CMY (cyan, magenta, yellow), odnosno CMYK (cyan, magenta, yellow, key (black)).

17. Objasniti RGB model boja.

RGB model boja spada u aditivne (svetlosne) modele boja. Aditivni model je model gde boju doživljavamo kao energiju svetlosnog izvora, tj. gde boju percipiramo kao zbir intenziteta svetlosti. Aditivnim mešanjem svetla određene talasne dužine i intenziteta možemo iz osnovnih komponenti crvene, zelene i plave postići ostale nijanse spektra. Svaka boja može da varira između minimuma (potpuno tamna) i maksimuma (potpuno osvetljena), pa se zato za odsustvo intenziteta sve tri boje dobija crna, a za njihov maksimalan intenzitet se dobija bela. Osnovne boje u RGB modelu boja su crvena (red), zelena (green) i plava (blue). Ove tri vrednosti određuju koordinate u 3D prostoru, a ponekad se dodaje i jedan kanal za providnost.

18. Objasniti HSV model boja. Navedite primer kada bi nam bilo zgodnije da koristimo HSV model boja naspram RGB modela.

HSV model boja spada u aditivne (svetlosne) modele boja. Aditivni model je model gde boju doživljavamo kao energiju svetlosnog izvora, tj. gde boju percipiramo kao zbir intenziteta svetlosti. HSV se bazira na ljudskom mentalnom modelu. Hue je ugao u stepenima meren od vertikale, tj. označava izbor boje, saturation je količinu sive u boji u procentima, a value (brightness) je nijansa sive u procentima, tj. intenzitet. HSV model je zgodnije koristiti u odnosu na RGB kada želimo da detektujemo objekat na slici prema njegovoj boji (npr. izdvajanje šake od pozadine). Ukoliko imamo RGB model i slike uslikane pod različitim osvetljenjima, sve tri nijanse boja na slici će se menjati sa promenom osvetljenja, što rezultuje time da nećemo moći detektovati objekat po boji. Kod HSV modela, promenom osvetljenja se menja samo value komponenta, a hue i saturation ostaju isti. Ovim dobijamo da objekte sa slike možemo izdvojiti prema boji tako što ćemo koristiti hue i saturation, a ignorisati value.

19. Kako možemo ukloniti šum sa slike?

Uklanjanje šuma sa slike se može vršiti sabiranjem, gde se sabiranje vrši piksel po piksel, ili lokalnim uprosečavanjem, što se češće koristi. Za lokalno uprosečavanje je neophodno da definišimo lokalno susedstvo piksela, odnosno klizeći prozor koji se kreće po slici sa leva na desno, i od gore na dole. Za svaku poziciju prozora posmatramo prosek piksela koji se nalaze u njemu i tu vrednost susedstva postavljamo kao vrednost posmatranog piksela. Klizeći prozor može biti različitih dimenzija i oblika, odnosno susedstvo može biti definisano na različite načine, a rezultati uprosečavanja zavise upravo od izbora susedstva. Pomoću lokalnog uprosečavanja uklanjamo šum, ali dobijamo zamućeniju sliku. Intenzitet zamućenosti će takođe zavisiti od toga kakvo susedstvo koristimo.

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

Oduzimanje slika se vrši piksel po piksel, gde se za svaki piksel rezultujuće slike uzima apsolutna vrednost razlike piksela na korespodentntim pozicijama. Ova operacija je zgodna za pronalaženje razlike između dve slike. Primeri primene oduzimanja slike su: detekcija pokreta (motion tracking), praćenje pokreta kamere, video kompresija, posmatranje medicinskih snimaka... Detekcija pokreta obično podrazumeva da imamo stacionarnu kameru (npr. video nadzor) i da na

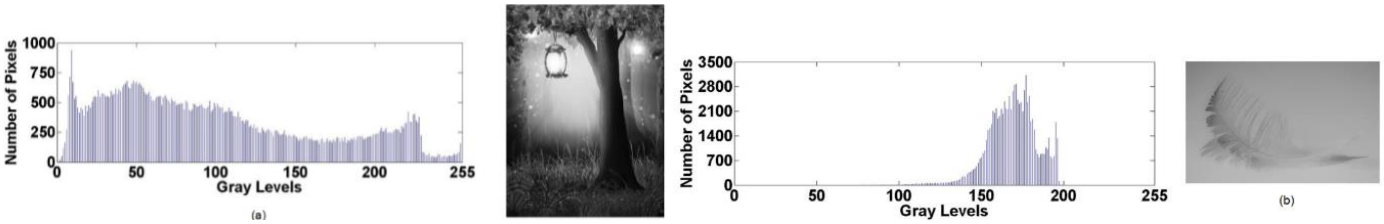
statičnoj pozadini tražimo značajne promene i pretpostavljamo da one odgovaraju pokretima. Ovo se obično koristi za nadgledanje, praćenje vozila na putu, prebrojavanje ljudi koji ulaze i izlaze iz prodavnice... Praćenje pokreta kamere se obično koristi pri snimanju filmova. Video kompresija podrazumeva prilično sličan proces kao i detekcija pokreta, gde se posmatraju frame-ovi i enkodiraju se samo razlike između njih, odnosno praktično isti frame-ovi se posmatraju kao jedan. Ovim se znatno dobija na uštedi memorije. Primer za korišćenje oduzimanja pri posmatranju medicinskih snimaka bi bilo posmatranje aktivnosti mozga, gde se posmatra slika mozga pre primene nekog sredsta i posle, kako bi se videlo koji su se krvni sudovi u mozgu aktivirali.

21. Šta su afine transformacije? Navedite primere primene.

Afine transformacije su transformacije nad slikom kao što su: skaliranje, rotacija, translacija, krivljenje, itd. One se mogu primeniti za promenu orijentacije slika (npr. iskošena slika papira na kom trebamo da detektujemo QR kod, otisak prsta ili linije teksta), ili za augmentaciju skupa podataka.

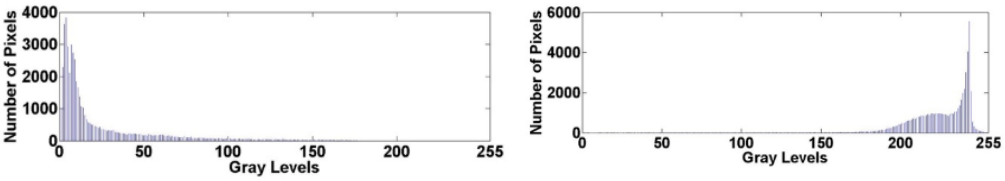
22. Kako proveriti da li slika ima dobar kontrast?

Proveravanje da li slika ima dobar kontrast se može izvršiti posmatranjem histograma. Histogram je grafički prikaz brojnosti piksela datog intenziteta. Kod njega se na x-osi nalaze intenziteti piksela, za svaki intenzitet prebrojimo koliko ima piksela tog intenziteta, i visina stupca prikazanog na histogramu je proporcionalna tom broju. Ako na čitavom rasponu intenziteta piksela imamo ravnomerno raspodeljene vrednosti, tj. imamo piksele i niskog i visokog intenziteta, onda slika ima dobar kontrast, a ako su vrednosti koncentrisane oko jednog određenog intenziteta, onda slika ima loš kontrast.



23. Objasnite kako dobijamo histogram slike. Šta možemo saznati o slici na osnovu njenog histograma?

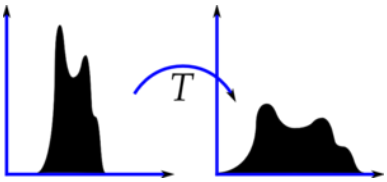
Histogram je grafički prikaz brojnosti piksela datog intenziteta. Kod njega se na x-osi nalaze intenziteti piksela, za svaki intenzitet prebrojimo koliko ima piksela tog intenziteta, i visina stupca prikazanog na histogramu je proporcionalna tom broju. Na osnovu histograma slike možemo dobiti informacije o njenom osvetljenju i kontrastu. Za osvetljenje se posmatra raspon vrednosti na histogramu. Ako su vrednosti histograma koncentrisane uglavnom na levoj strani, onda imamo nisko osvetljenje, a ako su koncentrisane uglavnom na desnoj strani, onda imamo visoko osvetljenje.



Za kontrast posmatramo da li imamo piksele svih intenziteta. Ako na čitavom rasponu intenziteta piksela imamo ravnomerno raspodeljene vrednosti, tj. imamo piksele i niskog i visokog intenziteta, onda slika ima dobar kontrast, a ako su vrednosti koncentrisane oko jednog određenog intenziteta, onda slika ima loš kontrast.

24. Kako poboljšati kontrast slike?

Za poboljštavanje kontrasta slike se može koristiti poravnavanje histograma (Histogram Equalization). Poravnavanjem histograma modifikujemo distribuciju intenziteta piksela tako da postoji otprilike jednak broj piksela za svaki od intenziteta. Kako želimo da histogram bude poravnat, ali da zadrži generalan oblik, mi ga zapravo „razvlačimo“. Ova operacija je veoma korisna za slike gde su i pozadina i objekti previše svetli ili previše tamni.

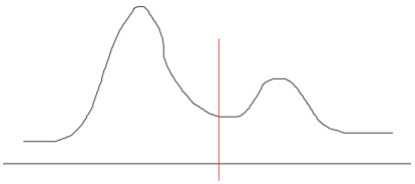


25. Šta je segmentacija slike pomoću praga (threshold)?

Segmentacija je postupak podele slike na regione sa sličnim atributima. Segmentacija slike pomoću praga (thresholding) je najjednostavniji način za segmentaciju slike. Pri segmentaciji slike pomoću praga poredimo osvetljenost piksela sa pragom ili više pragova, i svrstavamo piksele u dve ili više kategorija.

26. Na koji način možemo da odredimo prag?

Prag možemo odrediti pomoću Otsu metode. Ona automatski pronalazi prag na osnovu histograma. Ideja je da se posmatraju „doline“ histograma između jasno razdvojenih grupa intenziteta piksela. „Doline“ zapravo posmatramo kao granice između kategorija piksela.



27. Koja je intuicija iza Otsu metode za pronalaženje globalnog praga? Šta su prednosti, a šta nedostaci ovog pristupa?

Otsu metoda za pronalaženje globalnog praga automatski pronalazi prag na osnovu histograma. Ideja je da se posmatraju „doline“ histograma između jasno razdvojenih grupa intenziteta piksela. „Doline“ zapravo posmatramo kao granice između kategorija piksela. Prednosti Otsu-a su što je jednostavno za implementaciju i što je potpuno automatski. Međutim, na zašumljenim slikama, segmentacija pomoću Otsu praga nije dobra, ali se problem može ispraviti ako pre primene Otsu algoritma uklonimo šum nekom tehnikom za pretprocesiranje, kao što je lokalno uprosečavanje. Takođe, nedostatak je i što Otsu zanemaruje prostorne relacije, tj. koristi samo histogram koji nam ne govori ništa o prostornoj povezanosti, već se samo posmatra intenzitet piksela, bez obzira na to gde se regije nalaze na slici. Problem koji se još može javiti je problem nejednakog osvetljenja, što je generalni problem kod kod korišćenja globalnog praga, a rešenje ovog problema bi bilo da tražimo lokalni prag.

28. Šta je globalni, a šta lokalni prag?

Globalni prag je jedan jedini prag koji imamo pri segmentaciji slike pomoću praga. Korišćenje globalnog praga je najjednostavniji vid segmentacije pomoću praga, gde poredimo vrednosti svih piksela sa samo jednim pragom. Problem koji se ovde javlja je što se segmentacija neće dobro izvršiti za slike sa nejednakim osvetljenjem. Rešenje ovog problema je da tražimo lokalni prag. Traženje lokalnog praga podrazumeva da prag izaberemo pojedinačno za svaki piksel na osnovu lokalnih vrednosti piksela u njegovom susedstvu. Lokalni prag dobijamo na sličan način kao što radimo uprosečavanje slike, tj. imamo klizeći prozor koji se kreće po slici i za svaki od piksela posmatramo njegovo susedstvo, odnosno piksele unutar prozora, pronalazimo prag i na osnovu praga kategorišemo posmatrani piksel.

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

Nedostatak korišćenja jednog globalnog praga na celoj slici je nejednako osvetljenje. Nejednako osvetljenje na slikama će dovesti do toga da se segmentacija neće vršiti kako treba, tj. da će se slabije osvetljeni regioni svrstati u regione ispod praga, i obrnuto. Segmentacija na osnovu globalnog praga se obično koristi nad slikama gde imamo uniformno osvetlje (npr. mikroskopci snimci), ali je treba izbegavati pri segmentaciji prirodnih slika.

30. Objasnite postupak označavanja povezanih regiona na slici.

Označavanje povezanih regiona na slici se koristi kada želimo da grupišemo piksele na slici povezane u regione, tj. komponente. Kada otkrijemo sve grupe piksela koje postoje, svaki piksel anotiramo u zavisnosti od kategorije kojoj on pripada. Prva varijanta algoritma je da skeniramo sliku piksel po piksel i vršimo klasifikacije na osnovu susedstva. Obično se za susedstvo koristi 8-susedstvo, a postoji još i 4-susedstvo ili neke druge definicije susedstva. Skeniranje piksela jedan po jedan se vrši kroz klizeći prozor susedstva, i u tom procesu se dodeljuju kategorije pikselima i beleže ekvivalencije kategorija. Po završetku skeniranja piksela, dodeljuje se jedinstvena kategorija svim ekvivalentnim kategorijama. Druga varijanta algoritma je da se krene od prvog obojenog piksela na koji naiđemo, a onda pretražujemo okolne piksele koji su iste boje. Svaki piksel koji posetimo dodajemo u listu posećenih piksela, a sve njemu susedne piksele iste boje dodajemo u listu otvorenih ivica. U svakoj iteraciji algoritma isprobavamo po jedan piksel iz liste otvorenih ivica. Kako imamo liste piksela, možemo vršiti i pretrage kao što su breadth first search ili depth first search. U ovoj verziji algoritma izdvajamo regione komponentu po komponentu. Neke od primena pronalaženja povezanih regiona su: prebrojavanje objekata na realnim slikama (npr. prebrojavanje ćelija na mikroskopskom snimku), određivanje pozicije objekta na slici, segmentacija niza karaktera na pojedinačne karaktere (npr. prepoznavanje registarskih tablica).

31. Objasnite morfološku obadu slike. Šta je cilj (zašto primenjujemo morfološke operacije)? Šta su ulazi? Šta je izlaz? Navedite nazive nekih morfoloških operacija.

Pri segmentaciji slika pomoću praga, ili nekim drugim vrstama segmentacije, često kao izlaz dobijamo binarne slike koje imaju probleme u vidu crnog ili belog šuma. U binarnim slikama je pozadina predstavljena crnim pikselima, a objekat od interesa belim. Crni šum predstavlja rupe unutar objekata od interesa, a mi ga tretiramo kao pozadinu, dok beli šum predstavlja piksele pozadine, a mi ga tretiramo kao piksele objekta od interesa. Ovi šumovi predstavljaju false negative i false positive slučajeve koje trebamo rešiti, što je upravo cilj morfoloških operacija. Ulazi u morfološku obradu slike su binarna slika koja se obrađuje i strukturni element, tj. njegova veličina i oblik. Strukturni element je matrica sastavljena od nula i jedinica, i on definiše susedstvo piksela. Pomoću strukturnog elementa vršimo popunjavanje šumova. Izlaz iz morfološke obrade slike je obrađena binarna slika iste veličine kao i ulazna. U morfološkim operacijama, vrednost svakog piksela rezultujuće slike se zasniva na poređenju odgovarajućeg piksela na originalnog slici sa svojom okolinom. Morfološke operacije su skup operacija za obradu slika koje se baziraju na oblicima. Osnovne binarne morfološke operacije su dilacija i erozija, a njihovom kombinacijom se dobijaju otvaranje, zatvaranje i detekcija ivica.

32. Šta je dilacija?

Dilacija je osnovna binarna morfološka operacija. Ona predstavlja logičku OR operaciju između piksela okoline, gde je okolina definisana strukturnim elementom. Dilacija uklanja false negative slučajeve (crni šum), tj. popunjava rupe koje su se pojavile u objektu od interesa.

33. Šta je erozija?

Erozija je osnovna binarna morfološka operacija. Ona predstavlja logičku AND operaciju između piksela okoline, gde je okolina definisana strukturnim elementom. Erozija uklanja false positive slučajeve (beli šum), tj. uklanja šum koji predstavlja pozadinu, a mi ga tretiramo kao objekat od interesa.

34. 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.

35. Kako možemo rešiti probleme erozije i dilacije?

Probleme erozije i dilacije možemo rešiti primenom nekih kombinacija ovih morfoloških operacija, kao što su otvaranje i zatvaranje. Otvaranje je morfološka operacija gde se prvo primenjuje erozija, pa dilacija, a zatvaranje je morfološka operacija gde se prvo primenjuje dilacija, pa erozija. Kod obe kombinacije koristimo isti strukturni element pri primeni i

dilacije, i erozije. Efekat otvaranja i zatvaranja je izglađivanje (smoothing) objekta, pri čemu se otvaranje obično koristi za eliminasnje belog šuma, a zatvaranja za eliminisanje crnog šuma.

36. 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...).

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

Detekcija ivica pomoću morfoloških operacija se može izvšiti uz razne kombinacije dilacije, erozije i operacije oduzimanja. Neki od načina su: od slike nad kojom je primenjena dilacija oduzimanje originalne slike, od originalne slike oduzimanje slike nad kojom je primenjena erozija, i od slike nad kojom je primenjena dilacija oduzimanje slike nad kojom je primenjena erozija.

38. Objasnite operaciju konvolucije.

Konvolucija je matematička operacija definisana nad dve funkcije koja proizvodi treću funkciju. U obradi slike, konvolucija podrazumeva da imamo neki filter, tj. kernel, koji preklapamo preko piksela originalne slike i računamo šta će se nalaziti na izlaznoj slici, tj. na centralnom elementu kernela. Vrednost centralnog elementa na izlaznoj slici je razultat primene operacije konvolucije, što je zbir pomnoženih odgovarajućih vrednosti ulazne slike i kernela. Kernel dalje „prevlačimo“ preko svakog piksela na slici, tj. pomeramo ga za jadan korak kao klizeći prozor. Ukoliko želimo da održimo istu veličinu slike, možemo padovati ulaznu sliku nulama kako bi popunili sve elemente matrice. Operacija konvolucije je linearna i diferencijabilna. Pomoću konvolucije možemo vršiti različite operacije nad slikom, tj. možemo postići različite efekte u zavisnosti od izgleda filtera (npr. šiftovanje u levo, mean filter, izoštravanje...).

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

Segmentacija slike predstavlja proces u kom delimo sliku na smislene regije. Šta je smislena regija možemo definisati na različite načine (npr. grupa piksela slična po intenzitetu, teksturi ili nekom drugom atributu). Cilj segmentacije je da se pojednostavi ili promeni predstava slike u nešto što bi bilo smislenije i lakše za analizu. Segmentacija slike predstavlja jedan od ključnih koraka u analizi slike, jer rezultati segmentacije mogu da utiču na sve naredne procese analize slike. Smatramo da je segmentacija dobro izvršena kada vidimo da su regije koje su zajedno međusobno povezane po kriterijumu koji smo odredili, i da su pikseli u jednoj regiji slični po nekom atributu, a različiti od piksela iz drugih regija.

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

Izazovi koji se javljaju kod segmentacije slike su utvrđivanje ground truth-a i potreba za integracijom informacija visokog nivoa. Problem utvrđivanja ground truth-a se javlja jer je manuelna segmentacija dugotrajan i mukotrpan proces, a i rezultujuća segmentacija je subjektivna. Subjektivne segmentacije različitih eksperata će se najverovatnije razlikovati. Do svega ovoga dolazi jer su prirodne slike vrlo kompleksne, pa imamo problem da uvidimo da li su pikseli anotirani kako treba. Da bi se rešili ovi izazovi, umesto evaluacije piksel po piksel, trebala bi da se vrši evaluacija u smislu krajnjeg cilja. U ovom slučaju treba da se vrši segmentacija kao što je segmentacija bazirana na klasterovanju, gde se pikseli klasteruju u različite regije pomoću algoritama kao što je k-means. Takvu segmentaciju onda možemo evaluirati na osnovu krajnjeg cilja, tj. na osnovu toga za šta se segmentacija koristi (npr. ako u automnoj vožnji tražimo pešaka, posmatraćemo da li krajnji rezultat prepoznaje i izbegava pešaka).

41. Objasnite k-means algoritam. Šta sve moramo definisati da bismo mogli primeniti ovaj algoritam? Koji se problemi mogu javiti u kontekstu primene k-means na segmentaciju slike?

K-means algoritam podrazumeva da na početku na slučajan načini inicijalizujemo centroide, tj. centre klastera, i onda posmatramo distance za svaku tačku u odnosu na centroide. Za svaku tačku pronalazimo najbliži centar klastera i označavamo da tačka pripada tom klasteru. Nakon toga ažuriramo centroide klastera prema tačkama koje su mu dodeljene, odnosno pomeramo ih tako da budu u stvarnom centru pripojenih tačaka. Dalje vršimo ponavljanje ova dva koraka sve dok ne dođe do konvergencije. Da bismo primenili ovaj algoritam moramo definisati broj klastera i metriku distance. Problem koji se možu javiti pri primeni k-means algoritma na segmentaciju slika je ako se vektor koji opisuje jedan piksel definiše samo putem boje. Ovim dobijamo da se pri klasterovanju neće gledati prostorna povezanosst objekata, već će se pikseli grupisati isključivo po boji, tako da ćemo dobiti diskonektovane regije.

42. Objasnite ideju superpixels. Šta sve moramo definisati da bismo mogli primeniti ovaj algoritam?

Problem koji se možu javiti pri primeni k-means algoritma na segmentaciju slika je ako se vektor koji opisuje jedan piksel definiše samo putem boje. Ovim dobijamo da se pri klasterovanju neće gledati prostorna povezanosst objekata, već će se pikseli grupisati isključivo po boji, tako da ćemo dobiti diskonektovane regije. Superpixels algoritam predstavlja rešenje ovog problema. On je varijanta k-means algoritma koja se često koristi u obradi slike. Kod njega se pronalaze kontinualne regije koje su slične boje i intenziteta, tj. on predstavlja modifikaciju k-means-a u kojoj, pored boje i intenziteta, obraćamo pažnju i na prostornu povezanost. Varijanta superpixels algoritma podrazumeva da sliku delimo na ravnomernu rešetku i centroide klastera inicijalizujemo otprilike u centar svakog regiona rešetke. Dalje pri klasterovanju ne posmatramo čitavu sliku, nego samo određene susedne regione rešetke. Ovim dozvoljavamo da pikseli mogu preći iz jedne približne regija u drugu, ali ne mogu prelaziti iz nepovezanih regija. Nakon klasterovanja tačaka, vršimo ažuriranje centroida, kao u k-means-u. Korake klasterovanja i ažuriranja cetroida ponavljamo sve dok ne dođemo do konvergencije. Ideja superpixels algoritma je da obične piksele slike zamenimo superpikselima, tj. da više ne tretiramo sliku piksel po piksel, već regiju po regiju. Da bismo primenili ovaj algoritam, moramo definisati distancu i broj superpiksela. Pri definiciji distance, moramo fiksirati maksimalnu distancu u prostoru boja na neku konstantu m. Konstanta m definiše koliko želimo jaku prostornu povezanost, a koliko želimo da budemo slični po boji i intenzitetu. Za veliko m ćemo više vrednovati prostornu povezanost i superpikseli će biti kompaktniji, a za malo m ćemo više vrednovati intenzitet i boje i superpikseli će se više vezati za ivice na slici. Broj superpiksela utiče na to koliko želimo da superpikseli prate ivice. Ako imamo mnogo superpiksela, oni će bolje pratiti ivice, ali nećemo puno uraditi povodom segmentacije, a za malo superpiksela ćemo imati dobro odrađenu segmentaciju, ali će slabije pratiti ivice na slici.

43. Objasnite segmentaciju pomoću rasta regiona (region growing). Šta sve moramo definisati da bismo mogli primeniti ovaj algoritam? Koji se problemi mogu javiti?

Segmentacija pomoću rasta regiona (region growing) počinje od jednog piksela potencijalne regije. Regija se dalje proširuje dodavanjem susednih piksela, sve dok se potencijalni pikseli za dodavanje ne pokažu previše različitim. Kada dobijemo uniformnu regiju, ne proširujemo je dalje, već prelazimo na sledeću. Polazni piksel se bira ručno ili automatski (npr. klasterovanje, segmentacija praga...). Pri proširivanju regije, razmatramo sve susedne piksele, i za svaki gledamo da li je povezan sa regijom ili ne pre nego što ga dodamo. Da bismo primenili ovaj algoritam, moramo definisati: polazni piksel, uniformnost, tj. šta je karakteristika uniformne regije, i kriterijum sličnosti (npr. boja, tekstura...). Problem koji se može javiti je da regija može da „procuri“ kroz jednu „slabu tačku“ granice, tj. „slabe ivice“ mogu znatno uticati na krajnji rezultat algoritma.

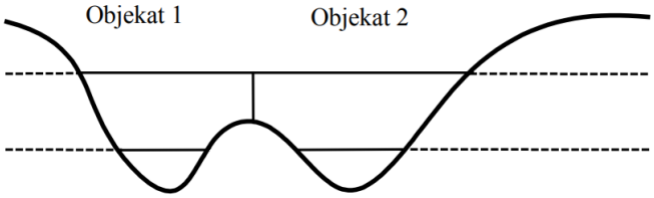
44. Objasnite segmentaciju pomoću razdvajanja i spajanja regiona (split-and-merge). Šta sve moramo definisati da bismo mogli primeniti ovaj algoritam? Koji se problemi mogu javiti?

Segmentacija pomoću razdvajanja i spajanja regiona (split-and-merge) je algoritam u kom za početnu regiju uzimamo celu sliku. Ako ta regija nije uniformna, delimo je na četiri kvadrata. U sledećem koraku spajamo sve susedne dovoljno slične regije, tj. one regije koje imaju isto uniformno obeležje. Dalje ponavljamo prethodna dva koraka sve dok ne bude ispunjen kriterijum zaustavljanja. Da bismo primenili ovaj algoritam, moramo definisati: uniformnost, prag i kriterijum zaustavljanja. Uniformnost označava šta je karakteristika uniformne regije (npr. razlika između najveće i najmanje osvetljenosti piksela u regionu). Prag označava kada je regija dovoljno uniformna da je ne delimo na regione, odnosno kada su regije dovoljno slične da možemo da ih spojimo. Kriterijum zaustavljanja označava koji uslov mora biti ispunjen

da bi se algoritam završio (npr. došli smo do najmanje veličine bloka, nema više deljenja i spajanja...). Ideja split-and-merge algoritma je da deli sliku na blokove, a upravo ta ideja dovodi do problema koji se može javiti. Podelom na blokove dobijamo blokovsku strukturu regiona u segmentiranoj slici, što znači da nemamo fine zakrivljene ivice.

45. Objasnite ideju Watershed segmentacije. Koja je prednost ovog postupka? Kako treba da pripremimo sliku da bismo primenili ovaj algoritam? Koji su nedostaci Watershed segmentacije?

Watershed segmentacija ima za ideju da interpretira sliku kao 3D površinu sa „dolinama“ i „planinama“, gde se mogu razlikovati tri tipa tačaka: tačke koje pripadaju minimumu regije, catchment basins i watershed lines. Catchment basins su tačke sa kojih bi kap vode pala u jedinstven minimum, a watershed lines su tačke sa kojih kap vode ima jednaku verovatnoću da padne u više od jednog minimuma. Watershed lines su tačke koje želimo da detektujemo, jer one razdvajaju dva objekta reprezentovana „dolinama“. Kod Watershed segmentacije pretpostavimo da u svakoj tački minimuma ima „rupa“ i, da se površ zaroni u vodu, voda bi ušla kroz „rupe“ u minimumima i poplavila površinu. Kako bi se izbeglo da se voda koja dolazi od dva različita minimuma izmešta, pravi se „brana“ kadgod se desi spajanje vodenih površina. Time bi se dobilo da bi jedine stvar vidljive na povšini bile upravo te „brane“, koje se zovu watershed lines.



U prevodu, ovo znači da se inicijalno primeni prag sa niskim nivoom svetline i na taj način se pronađu svi objekti, gde su područja objekata premala. Zatim se prag povećava za jedan nivo svetline u svakom koraku. Ovim područja oko objekata rastu sa povećanjem praga, ali se ne dozvoljava njihovo spajanje kada se dodirnu. Tačka prvog kontakta se uzima za granicu između objekata. Ovaj proces se zaustavlja pre nego što prag dostigne nivo pozadine. Postupkom Watershed algoritma se obezbeđuje dobra segmentacija slika sa objektima koji se nalaze veoma blizu jedan drugog. Prednost Wateshed segmentacije je što ume da razdvoji komponente koje su konektovane. Da bismo primenili ovaj algoritam sliku trebamo da pripremimo tako da je posmatramo kao gradijent ili da odradimo distance transform. Gradijent može biti korišćen za grayscale slike, ali bi postojao veliki šum, pa bi dobili veliki broj grupa i prekomernu segmentaciju. Distance transform može biti korišćen nad binarnim slikama. Kod njega je ideja da se kreiraju granice koje su najdalje moguće od centara objekata koji se preklapaju. Prekomerna segmentacija (over-segmentation), koja predstavlja problem Wateshed segmentacije, se javlja kada imamo veliki broj potencijalnih minimuma od kojih nisu svi relevantni, što je u stvari posledica šuma. Rešenje ovog problema je ograničiti broj dozvoljenih regija, ili koristiti predefinisan skup markera kako bismo specificirali samo dozvoljene regione minimuma.

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

Kako smo sliku predstavili kao funkciju intenziteta, tamo gde se nalazi ivica dolazi do nagle promene funkcije intenziteta. Osnovna ideja detekcije ivica je potražiti susedstvo sa značajnim znacima promene. Da bi uvideli te značajne znake promene, tj. brzinu promene funkcije, posmatramo njen gradijent. Što je gradijent po apsolutnoj vrednosti veći, to se funkcija brže menja. Pošto je slika u 2D-u, tražimo gradijent po dve komponente, tj. po x i po y. Nad tim vrednostima posebno primenjujemo neki od operatora koji predstavljaju funkcije konvolucije, iz čega dobijamo dve matrice. Kombinacijom ovih matrica dobijamo magnitudu gradijenta. Magnituda, tj. dužina, gradijenta označava jačinu ivice, a ugao gradijenta govori pod kojim uglom je ivica. Neki operatori za detekciju ivica su: Roberts-Cross operator, Sobelov operator i Canny.

47. Objasnite ideju Roberts-Cross operatora za detekciju ivica. Koje u prednosti, a koje mane ovog algoritma?

Roberts-Cross operator za detekciju ivica je zasnovan na konačnim razlikama i predstavlja funkciju konvolucije koja ima dva kernela koji su jednaki, samo zarotirani za 90°. Kerneli su dimenzije 2x2 i takve strukture da praktično traže gradijent po dijagonalama. Ovo predstavlja jednu od mana, jer najveći odziv imamo na ivice koje su dijagonalne, a trebale bi sve ivice da imaju jednaku šansu detekcije, bez obzira na to pod kojim se uglom nalaze. Pošto je slika u 2D-u, tražimo gradijent po dve komponente, tj. po x i po y. Ovo zapravo znači da kernele primenjujemo odvojeno na ulaznu sliku i kao

rezultat dobijamo dve matrice, čijom kombinacijom pronalazimo magnitudu gradijenta. Magnituda, tj. dužina, gradijenta označava jačinu ivice, a ugao gradijenta govori pod kojim uglom je ivica. Kako imamo relativno mali filter, dobijamo da je on osetljiv na šum, jer promenu gledamo na jednom susednom pikselu, koji upravo može biti piksel koji sadrži šum. Takođe, zbog malog kernela, Roberts-Cross operator ima slab odziv na ivice koje nisu oštre. Još jedna mana je da je nejasno koji izlazni piksel odgovara kom ulaznom, jer će slika gradijenta biti pomerena za pola piksela i po x i po y u odnosu na originalnu sliku. Međutim, prednost ovog operatora je da se brzo izvršava, upravo zbog malog kernela, i da nema parametara koje treba podešavati.

48. Objasnite ideju Sobelovog operatora za detekciju ivica. Koje u prednosti, a koje mane ovog algoritma?

Sobelov operator za detekciju ivica je sličan Roberts-Cross operatoru. On predstavlja funkciju konvolucije koja ima dva kernela koji su jednaki, samo je jedan zarotiran za 90°. Kerneli su dimenzije 3x3 i takve strukture da maksimalno reaguju na ivice koje su vertikalne i horizontalne. Pošto je slika u 2D-u, tražimo gradijent po dve komponente, tj. po x i po y. Ovo zapravo znači da kernele primenjujemo odvojeno na ulaznu sliku i kao rezultat dobijamo dve matrice, čijom kombinacijom pronalazimo magnitudu gradijenta. Magnituda, tj. dužina, gradijenta označava jačinu ivice, a ugao gradijenta govori pod kojim uglom je ivica. Mana Sobelovog operatora je da je sporiji od Roberts-Cross operatora, jer sa većim kernelom imamo i više množenja matrica. Takođe, veći kernel više „izglađuje“ ulaznu sliku, pa ćemo imati ivice široke par piksela, a idealno bi bilo da su one široke jedan piksel. Ovaj problem bi se mogao rešiti nekom morfološkom operacijom, kao što je erozija ili otvaranje. Prednosti Sobelovog operatora su da je manje osetljiv na šum i da reaguje i na blaže ivice, a ne samo na oštre.

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

Canny algoritam za detekciju ivica je dizajniran da bude optimalan detektor ivica. Kod njega imamo nagodbu između detekcije i lokalizacije ivica. Njegov prvi korak je uklanjanje šuma, gde se vrši izglađivanje Gausovim operatorom. Zatim se računa magnituda i orijentacija gradijenta pomoću jednostavnog operatora kao što je Roberts-Cross, čime ćemo dobiti da su ivice na ulaznoj slici grebeni na izlaznoj. Nakon toga se vrši non-maxima suppression, što je korak koji „istanjuje“ linije izlaza. Non-maxima suppression algoritam ide po grebenima i postavlja na nulu sve piksele koji nisu na vrhu grebena, jer se ivica nalazi tamo gde je gradijent najveći. I poslednji korak Canny algoritma je primena dva praga. Doble thresholding, tj. primena dva praga, se vrši kako bi se rešili problemi nekonektovanih ivica i prisustva šuma. Kao dva praga se koriste T1 i T2 (T1 > T2), gde je sve iznad T1 „jaka“ ivica, sve između T1 i T2 je „slaba“ ivica, a sve ispod T2 nije ivica. „Slaba“ ivica znači da postoji nesigurnost da li su ti pikseli ivica ili nisu. Da bi se odredilo koje slabe ivice su zapravo ivice, koristi se edge tracking. Edge tracking podrazumeva da su stvarne ivice one slabe ivice koje su povezane sa jakim ivicama, a one koje nisu povezane sa jakim ivicama se uklanjaju. Canny algoritam ima tri parametra: širina Gausovog kernela, T1 i T2. Širina Gausovog kernela utiče na izlaz tako što sa većom širinom imamo manju osetljivost na šum, ali se gube finiji detalji. Za dobre rezultate je obično preporuka da se bira visoko T1 i nisko T2. Postavka T1 na prenisku vrednost može rezultovati zadržavanjem lažnih fragmenata ivica u izlazu, a postavka T2 na previsoku vrednost može rezultovati da ivice sa šumom budu razbijene u fragmente.

50. Čemu služi Hough transformacija?

Hough transformacija služi da adresira problem grupisanja tačaka ivice u objekat, odnosno pomoću nje prelazimo sa kompleksne reprezentacije ivica pomoću piksela na detekciju oblika. Možemo je koristiti za detekciju proizboljnih oblika, kao što su linije, krugovi, ili bilo koji drugi oblici koji se mogu predstaviti u matematičkoj formi. Hough transformacija se koristi kao tehnika u analizi slike, digitalnoj obradi slike i computer vision-u.

51. Šta su prednosti, a šta mane Hough algoritma?

Prednosti su da je otporan na šum i delimično zaklanjanje oblika. Mane su da je memorijski zahtevan i spor algoritam, jer za kompleksnije oblike imamo više parametara, koje je neophodno smestiti u memoriju i isprobati sve njihove kombinacije, pa je Hough praktičan uglavnom samo za jednostavnije krive. Takođe, mana je da se u jednom prolazu može tražiti samo jedan tip objekata, tj. oblika.

52. Izložite osnovnu ideju algoritma Hough transformacije.

Prvi korak Hough transformacije, tj. pretprocesiranje je primena nekog algoritma za detekciju ivica, kao što su Robert-Cross operator, Sobelov operator, Canny... Zatim se vrši transformacija originalne slike u parametarski prostor, tj. mapiranje piksela sa ivica na Hough prostor i snimanje u akumulator. Svaka linija se može predstaviti jednačinom y = ax + b, tj. parametri a i b jedinstveno određuju pravu. Hough prostor je parametarski prostor koji predstavlja sve moguće nagibe a i preseke sa y-osom b. Svaka tačka ivice će glasati za skup mogućih parametara u Hough prostoru. Isto tako, i pikseli koji predstavljaju šum će glasati, ali će njihovi glasovi biti nekonzistentni sa većinom „dobrih" glasova. Glasove akumuliramo u diskretan broj ćelija, a parametri sa najviše glasova predstavljaju linije na slici. Hough prostor je beskonačan, pa se zato deli u konačne ćelije, što su zapravo ćelije akumulatora. Nakon smeštanja ivica u akumulator, vrši se interpretacija akumulatora, tj. pronalaženje beskonačnih linija. Za ćelije koje imaju najveći broj „glasova" se smatra da reprezentuju odgovarajuće linije u originalnom prostoru slike, tako da u ovom koraku biramo kandidate iz akumulatora. Na kraju se vrši konverzija beskonačnih linija u konačne, za šta postoji nekoliko algoritama. Jedan pristup je da čuvamo koordinate svih tačaka u akumulatoru i na osnovu njih limitiramo linije, a drugi pristup je da vršimo pretragu duž beskonačnih linija na binarnoj slici.

53. Koji su najčešći problemi kod klasifikacije slika?

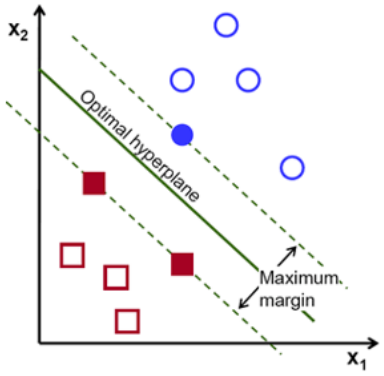
Najčešći problemi kod klasifikacije slika su: varijacija u položaju kamere, varijacija u osvetljenju, deformacija objekta, zaklonjenost objekta, velika sličnost objekta i pozadine, varijacija u klasi, loš kvalitet slike... Varijacija u položaju kamere podrazumeva da veoma mali pomeraj kamere dovodi do toga da imamo potpuno drugačiju matricu kojom se reprezentuje slika. Deformacija objekta podrazumeva da se isti objekat može naći u različitim pozama i položajima. Varijacija u klasi predstavlja problem gde imamo različite vrste objekata unutar iste klase (npr. mačke različitih izgleda, stolice različitih vrsta...).

54. Dajte konceptualni pregled sistema za klasifikaciju slika („klasičan" pristup, ne deep learning pristup). Kako se deep learning pristup razlikuje?

Konceptualni pregled sistema za klasifikaciju slika podrazumeva da se prvo vrši pretprocesiranje slika, zatim ekstrakcija obeležja (feature extraction), pa se onda primenjuje neki obučavajući (learning) algoritam, i na kraju se dodeljuju labele. Tradicionalni pristupi prate ovaj pipeline, a deep learning preskače korak ekstrakcije obeležja. Kod pretrpocesiranja slike se vrše operacije kao što su normalizacija vrednosti piksela, poboljšavanje kontrasta, i slično, ali je ključno da se svedu sve slike iz skupa podataka na neku fiksnu veličinu. Cilj ekstrakcije obeležja je da uklonimo informacije irelevante sa aspekta klasifikacije, što nas čini otpornim na sve varijacije koje se javljaju kao problemi klasifikacije (npr. ne posmatramo boju novčića pri njihovoj detekciji). Obučavajući algoritmi su algoritmi koje koristimo za klasifikaciju (npr. SVM, K-NN, ANN...), i za njih nam je potreban trening skup. Kod tradicionalnog pristupa, korak ekstrakcije obeležja je ključan za performanse klasifikacije, i to vrše ljudi tako što kažu šta je bitno za rešavanje problema. Ovakav pristup nije idealan, jer je ljudima često teško da odrede na osnovu čega se klasifikuju objekti na slikama, a i skupovi obeležja se menjaju od problema do problema. Kod deep learning pristupa nemamo ove mane, jer puštamo model da sam uči obeležja koja su bitna, tj. nemamo čoveka koji ručno dizajnira obeležja, već sam algoritam uči obeležja bitna za problem. Ovo je mnogo bolja varijanta za velike skupove podataka, jer ljudima nije ekspertski lako da izraze šta je bitno kada posmatraju sliku. Za manje skupove podataka će uglavnom tradicionalni prisup dati bolje rezultate. Takođe, može se vršiti i kombinacija tradicionalnog i deep learning prisupa.

55. 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 razdvajajuće hiper-ravni je minimum rastojanja od te hiper-ravni do neke od tačaka 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 imamo podatke koji nisu linearno separabilni, možemo koristiti parametar C koji kontroliše nagodbu između veličine margine i broja grešaka. Za malo C imamo veliku marginu, ali dozvoljavamo da dosta primera bude pogrešno klasifikovano, a za veliko C imamo malu marginu, ali mali broj primera je pogrešno klasifikovan. Takođe, možemo i transformisati podatke u prostor u kom jesu linearno separabilni i onda primeniti SVM. Ovo se može postići korišćenjem RBF kernela i podešavanjem njegovog parametra γ.

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

HOG (Histogram of Oriented Gradients) deskriptor služi za ekstrakciju obeležja, odnosno za transformaciju slike u vektor obeležja. U tom dobijenom vektoru se nalaze stvari koje su bitne za klasifikaciju objekata, što su kod HOG-a konture objekata, tj. ivice. Ideja HOG-a je da se izgled objekta može efektivno opisati pomoću histograma smera ivica, gde je smer orijentacija gradijenta. Prvo se vrši računanje gradijenta po x-osi i po y-osi, pomoću nekog od operatora za detekciju ivica. Iz ovoga dobijamo magnitudu i orijentaciju gradijenta. Gradijent uklanja informacije koje nam nisu bitne (npr. pozadinu konstantne boje) i iseca konture. Dalje se vrši podela slike na NxN ćelija, a nakon toga se vrši računanje histograma gradijenta za svaku ćeliju. Računanje histograma po ćelijama čini reprezentaciju otpornijom na šum. Nakon računanja histograma, vrši se normalizacija bloka. Konstruisan histogram je baziran na gradijentu slike, a gradijent je osetljiv na osvetljenje, pa je i čitav postupak HOG-a osetljiv na osvetljenje. Da bi ovo rešili, vršimo normalizaciju histograma, gde svaku vrednost unutar dobijenih vektora delimo sa magnitudom vektora. Normalizacija se vrši nad blokom ćelija, a ne nad jednom ćelijom, kako ne bi izgubili informaciju koji objekat je u pitanju. Nakon normalizacije blokova, vrši se konkatenacija vektora, što podrazumeva da reprezentaciju čitave slike dobijamo tako što konkateniramo dobijene vektore. Kada imamo vektorsku reprezentaciju slike, dalje se trenira model (npr. SVM) i vršimo klasifikaciju.

57. Algoritam mašinskog učenja za prepoznavanje objekata na slikama se trenira na slikama fiksne veličine. Kako se ovaj algoritam može primeniti na prirodne slike gde ne znamo unapred na kom delu slike se objekat nalazi, koje veličine će biti i kojih će biti proporcija? Navedite koje algoritme u ovu svrhu možemo primeniti.

Pošto se algoritam mašinskog učenja za prepoznavanje objekata na slikam trenira na slikama fiksne veličine, da bi se on mogao primeniti na prirodne slike, moramo vršiti selektovanje isečaka originalne slike i primenu treniranog modela na te isečke. Objekat će biti lociran tamo gde klasifikator daje najveću verovatnoću da isečak sadrži objekat. Algoritmi koje možemo koristiti u ovu svrhu su: sliding window i region proposal.

58. Objasnite kako funkcioniše i čemu služi sliding window pristup u detekciji objekta na slici. Koje su mane ovog pristupa?

Sliding window pristup je algoritam koji se koristi kako bi algoritam mašinskog učenja za prepoznavanje objekata na slikama mogli primeniti na prirodne slike. On podrazumeva da pri klasifikaciji slike pomeramo prozor preko slike da bi selektovali isečak. Za taj isečak računamo HOG deskriptor i primenjujemo trenirani model da vidimo da li se tu nalazi traženi objekat. Prozor se preko slike pomera sa levo na desno, i od gore ka dole. Mana je da se objekat može nalaziti bilo gde na slici, pa moramo da analiziramo sve moguće isečke. Time dobijamo da nepotrebno ispitujemo mnogo pozicija na slici. Analizom svih mogućih isečaka možemo doći i do overlapping bounding box problema, koji podrazumeva da smo dobili više isečaka za istu instancu objekta. Ovaj problem se može rešiti primenom non-maximum suppression algoritma. Takođe, mana je i da objekti mogu biti različite veličine, tj. mogu biti premali ili preveliki u odnosu na prozor koji

pomeramo, ili različitih oblika, orijetancija i proporcija. Zbog različitih veličina moramo da vršimo skaliranje slika i da ponavljamo isti postupak na skaliranim verzijama slike, a zbog različitih proporcija moramo ponoavljati postupak za različite proporcije isečaka. Zbog svih ovih mana, sliding window je veoma spor algoritam.

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

Region proposal pristup je algoritam koji se koristi kako bi algoritam mašinskog učenja za prepoznavanje objekata na slikama mogli primeniti na prirodne slike. On predstavlja alternativu sliding window pristupu i kod njega ne ispitujemo svaku moguću lokaciju na slici, već tražimo kandidate. Ulaz u region proposal algoritam je slika, a izlaz iz njega su svi isečci-kandidati koji verovatno sadrže objekat. Ovi regioni mogu da sadrže šum, da se preklapaju i da ne sadrže objekat u potpunosti, ali će se među njima verovatno naći kandidat veoma blizak pravom objektu na slici. Za svakog kandidata računamo HOG deskriptor i primenjujemo obučeni klasifikacioni model. Isečci-kandidati se određuju segmentacijom, gde se grupišu međusobno slični pikseli u regije. Postoje različiti načini implementacije segmentacije kod region proposal-a, ali kod svih njih je bitno da imaju veoma veliki odziv. Ovo znači da se objekti moraju naći među isečcima-kandidatima i da su false positives manji problem od false negatives. Najčešće korišćen algoritam za segmentaciju je selective search, koji je brz i ima veoma velik odziv. Prednosti region proposal pristupa u odnosu na sliding window su da nismo ograničeni na određenu veličinu, oblik i proporciju regiona, i da imamo značajno manje regiona na koje moramo primeniti klasifikator, pa je mnogo brži algoritam.

60. Objasnite Hammingovu distancu za poređenje stringova. Dajte jedan primer (dva stringa i njihova Hammingova distanca). Koju manu ima ova mera udaljenosti?

Hammingova distanca za poređenje stringova je metod poređenja gde se stringovi porede karakter po karakter i posmatra se da li su isti karakteri na korespodentnim pozicijama. Distanca koju dobijemo je broj bitova u kojima se stringovi razlikuju, tj. ona predstavlja broj različitih bitova u nizovima iste dužine. Primer bi bio da ćemo za stringove MACKA i MAČKA dobiti da je Hammingova distanca jednaka 1, jer se navedeni stringovi razlikuju samo po jednom slovu (C i Č). Mana ove mere udaljenosti je što se mogu porediti isključivo stringovi iste dužine. Alternativa Hammingove distance je Levenshtein distanca.

61. Objasnite Levenshtein distancu za poređenje stringova. Navedite jedan primer gde bi se ona mogla koristiti. Šta sve treba definisati kako bismo mogli korisiti Levenshtein distancu?

Levenshtein distanca za poređenje stringova je metod poređenja gde se posmatra broj transformacija koje trebaju biti izvšrene da bi se od jednog stringa dobio drugi. Distanca koju dobijemo je minimalan broj operacija neophodan da se jedan string pretvori u drugi. Levenshtein distanca je mera razlike između dva niza, koji mogu biti nizovi slova, brojeva ili bilo kojih drugih objekata koji se mogu definisati kao niz karaktera. Kod ovog metoda nema ograničenja da dužine poređenih nizova moraju biti iste. Primer bi bio da ćemo za stringove MACKA i MAČKA dobiti da je Levenshtein distanca jednaka ceni zamene slova C za slovo Č, a za stringove MAČ i MAČKA će distanca biti jednaka ceni dodavanja slova K i A. Da bi se ovaj algoritam mogao koristiti, moramo definisati koje sve transformacije postoje i koje su njihove cene. Cena operaije može zavisiti od raznih svojstava stringova koji se porede. Ako definišemo cene karakter po karakter za svaku transformaciju, imamo mnogo parametara modela, model sklon overfitting-u i treba nam puno podataka da bismo pravilno naučili cene. Iz tih razloga je bolje koristiti neke predefinisane cene ili jednostavnije načine definisanja (npr. različite cene za samoglasnike i suglasnike, cena u zavisnosti blizine karaktera na tastaturi...). Levenshtein distanca se može koristiti za: spelling correction, traženje duplikata u registru ulica, sekvenciranje proteina, poređenje pesama ptica...

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

Euklidska udaljenost između dva vektora podrazumeva da gledamo njihovu razliku po magnitudi, a kosinusna podrazumeva da gledano razliku po orijentaciji, tj. posmatramo ugao između njih. Kod Euklidske udaljenosti se sve dimenzije pomsatraju jednako, a mogu se i nekim dimenzijama dodeliti veće težine ukoliko su određena obeležja relevantnija za problem. Otežnjavanje obeležja je veoma teško, pa se u praksi uglavnom izbegava. Ukoliko poredimo dva dokumenta pomoću Euklidske udaljenosti, oni će biti slični ako su iste dužine, bez obzira na sadržaj. Ovo u nekim

slučajevima može biti opravdano (npr. poređenje kuća prema kvadraturi), ali kada nije, može se rešiti normalizaciojom vektora na dužinu 1. Normalizovan vektor dužine 1 se dobija kada svaku komponentu vektora podelimo sa njegovom dužinom. Kosinusna udaljenost je normalizovana, tj. poredi normalizovane vektore. Ukoliko poredimo dva dokumenta pomoću kosinusne udaljenosti, ona neće posmatrati dužinu dokumenata, već isključivo njihov sadržaj.

63. Koja je suštinska razlika između Euklidske i kosinusne udaljenosti? Kada biste koristili jednu, a kada drugu?

Euklidska udaljenost između dva vektora podrazumeva da gledamo njihovu razliku po magnitudi, a kosinusna podrazumeva da gledano razliku po orijentaciji, tj. posmatramo ugao između njih. Kod Euklidske udaljenosti se sve dimenzije pomsatraju jednako, a mogu se i nekim dimenzijama dodeliti veće težine ukoliko su određena obeležja relevantnija za problem. Otežnjavanje obeležja je veoma teško, pa se u praksi uglavnom izbegava. Ukoliko poredimo dva dokumenta pomoću Euklidske udaljenosti, oni će biti slični ako su iste dužine, bez obzira na sadržaj. Ovo u nekim slučajevima može biti opravdano (npr. poređenje kuća prema kvadraturi), ali kada nije, može se rešiti normalizaciojom vektora na dužinu 1. Normalizovan vektor dužine 1 se dobija kada svaku komponentu vektora podelimo sa njegovom dužinom. Kosinusna udaljenost je normalizovana, tj. poredi normalizovane vektore. Ukoliko poredimo dva dokumenta pomoću kosinusne udaljenosti, ona neće posmatrati dužinu dokumenata, već isključivo njihov sadržaj. Euklidska udaljenost se koristi za u K-NN algoritmu i u k-means klasterovanju.

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

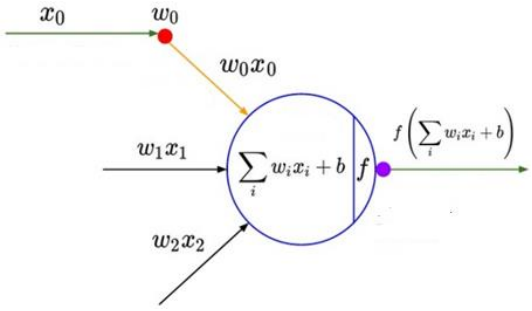
Pri odabiru mere udaljenosti objekata trebamo uzeti u obzir koju meru želimo da koristimo (Euklidksa, kosinusna, Manhattan, i druge, ili neka njihova kombinacija), koji problem želimo da rešimo i kako ćemo reprezentovati objekte (npr. sirova reprezentacija u pikselima je za detekciju cifara dobra, ali za detekciju lica i nije baš).

65. Objasniti K-NN algoritam. Za šta se koristi? Koje hiper-parametre treba da optimizujemo (objasnite uticaj) i kako? Koji je nedostatak ovog algoritma?

K-NN (K-Nearest Neighbors) je algoritam koji se koristi za nadgledano učenje, tj. za klasifikaciju i regresiju, gde se posmatraju najbliži susedi. On funkcioniše tako što se objekat koji se posmatra klasifikuje u odnosu na to kako su njegovi susedi klasifikovani. Hiper-parametar koji se treba optimizovati kod K-NN-a je K. Da bi optimizovali hiper-parametre i testirali model, ideja je da podatke podelimo na trening, validacioni i test skup. Trening skup nam služi za treniranje modela i optimizaciju parametara koji se uče iz podataka (npr. težine, bias-i), validacioni skup nam služi za optimizuju hiper-parametara, a test skup nam služi za evaluaciju performansi modela. Hiper-parametar K biramo isprobavanjem, gde prvo postavimo da je K=1, treniramo na trening skupu, testiramo na validacionom, pa promenimo da je K=2, treniramo na trening skupu, testiramo na validacionom, itd. sve dok ne dođemo do K koje daje najveću tačnost. Tada testiramo taj model na test skupu. Za veoma malo K se mogu javiti problemi outlier-a i overfitt-ovanja modela, a za veoma veliko K može doći do underfitt-ovanja modela. Što je veće K, to je veće izglađivanje (smoothing) podataka. Nedostatak K-NN algoritma je što može imati veoma loše performanse kada je broj obeležja velik, jer najbliži susedi u vidokodimenzionalnom prostoru mogu biti veoma daleko.

66. Objasniti matematički model veštačkog neurona.

U matematičkom modelu veštačkog neurona imamo ulaze (xi) koji se množe sa sebi pridodatim težinama (wi). Ideja je da se težinama kontroliše jačina i smer uticaja jednog neurona na drugi. Unutar neurona radimo linearnu kombinaciju ulaza i težina, tj. sabiramo izmnožene ulaze i težine, i na to dodajemo bias (b). Taj rezultat se provlači kroz aktivacionu funkciju (f). Aktivaciona funkcija je nelinearna funkcija koja određuje nivo pobuđenosti neurona za date ulaze. Izlaz iz aktivacione funkcije predstavlja izlaz iz neurona.



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

Ukoliko bi imali samo jedan neuron, imali bi linearni klasifikator, tj. granica odluke ovog klasifikatora bi bila prava linija. Ako nam je potrebna neka nelinearna granica odluke (npr. kružnica), to ne možemo postići sa samo jednim neuronom. Pomoću kombinacije neurona možemo dobiti nelinearnu granicu odluke.

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

Feedforward model neuronske mreže je struktura sa propagacijom unapred. On podrazumeva da su izlazi iz neurona povezani isključivo sa neuronima iz sledećeg sloja, tj. ne postoje povratne veze, veze koje preskaču slojeve ili rekurentne veze. U neuronskim mrežama se neuroni grupišu u slojeve, gde je prvi ulazni, dalje ide jedan ili više skrivenih slojeva, i na kraju je izlazni sloj. U ulaznom sloju se nalaze neuroni koji prosleđuju ulaze, a izlazni sloj interpretiramo kao rezultat neuronske mreže. Feedforward model u izlaznom sloju za klasifikaciju ima onoliko neurona koliko ima klasa, a za regresiju ima samo jedan neuron. Ni u slučaju klasifikacije, ni u slučaju regresije, se ne primenjuje aktivaciona funkcija u izlaznom sloju, da ne bi nepravilno ograničili vrednosti.

69. Čemu služi aktivaciona funkcija?

Aktivaciona funkcija je nelinearna funkcija koja određuje nivo pobuđenosti neurona za date ulaze. Ona dodaje nelinearnost modelu kako bi mogli da modelujemo kompleksne granice odluka. Ako ne koristimo aktivacionu funkciju, izlazni signal će biti linearna funkcija, pa bi se bez nje cela neuronska mreža svela na veoma komplikovanu linearnu regresiju.

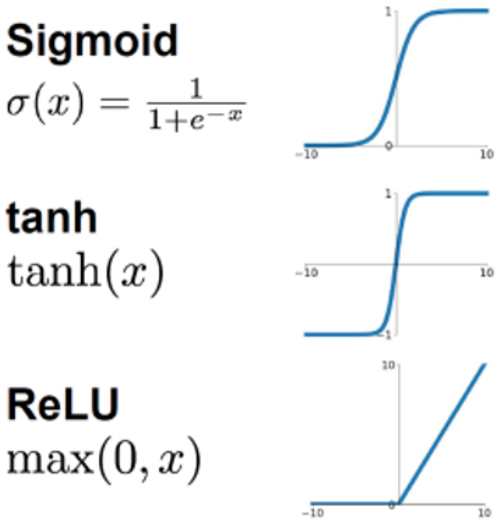
70. Konceptualno objasniti kako se trenira neuronska mreža. Šta je to što „učimo" u procesu treniranja? Koju funkciju greške koristimo za regresiju, a koju za klasifikaciju?

Neuronska mreža se trenira da bi za date ulaze i težine izračunali šta je predikcija modela. Ono što učimo u procesu treniranja su težine i bias-i, tj. obučavanje modela podrazumeva da korigujemo težine i bias-e kako bi dobili da se predikcije što bolje poklapaju sa očekivanim vrednostima. Učenje se vrši tako što se izračuna greška predikcije, koja zavisi od težine i biasa, i gleda se da se ta greška minimizuje. Greška predikcije predstavlja razliku stvarnih vrednosti i onoga što model predviđa. Za minimizovanje greške se traže težine za koje je greška najmanja, i za to se obično koristi algoritam gradijentnog spusta (gradient descent). Funkcija greške koju koristimo za regresiju je mean squared error, koja se računa kao suma kvadrata razlika predikcija i stvarnih vrednosti. Funkcija greške koju koristimo za klasifikaciju je log loss, koja se računa kao negativna vrednost logaritma verodostojnosti parametra. Verodostojnost parametra predstavlja verovatnoću da je neuronska mreža određenom xi dodelila klasu yi. Ako je yi stvarna klasa za xi, a dobijamo malu verovatnoću, log loss će biti velik, i obrnuto. Obično se pri klasifikaciji na izlazu vrši softmax funkcija, kako bi se transformisali izlazni signali tako da su nenegativni i da se sumiraju na 1. Upravo rezultati ove funkcije predstavljaju verovatnoće odgovarajućih klasa koje koristimo pri računanju log loss funkcije greške.

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

Tri poznate aktivacione funkcije su: Sigmoid, tanh i ReLU. Sigmoid mapira izlazni signal na interval [0, 1]. Prednosti Sigmoid aktivacione funkcije su što vrši lepu interpretaciju, gde 0 označava da nema signala, 1 označava punu saturaciju signala, a između 0 i 1 imamo postepen prelaz. Mane Sigmoid funkcije su da „ubija" gradijente saturisanih neurona i da nije centrirana oko nule, tj. njen izlaz nije centriran oko nule i ulaz u naredni neuron će uvek biti pozitivan. Takođe,

problem je i što je eksponencijalna funkcija skupa za računanje. Aktivaciona funkcija tanh mapira izlazni signal na interval [-1, 1]. Prednosti ove funkcije su da je centrirana oko nule, za razliku od Sigmoid funkcije, ali kao i kod nje, dolazi do saturacije aktivacija, tj. „ubijanja“ gradijenata saturisanih neuorna. ReLU je veoma popularna aktivaciona funkcija i jedna je od boljih aproksimacija onoga što se dešava kod bioloških neurona. Njene prednosti su da neće saturisati u pozitivnoj regiji i da u praksi mnogo brže konvergira od Sigmoid i tanh funkcije. Mane ReLU funkcije su da nije centrirana oko nule, da kod negativnog ulaza dolazi do saturacije i da neuron može „umreti“ tokom treninga. Do „umiranja“ neurona dolazi ako imamo lošu inicijalizaciju i veliki learning rate, pa nikada nećemo dobijti ulaz koji aktivira određene neurone.



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

Pri primeni neuronskih mreža, treba voditi računa o pretprocesiranju podataka, izboru aktivacione funkcije i inicijalizaciji težina. Pri pretprocesiranju podataka treba vršiti centralizaciju oko nule i normalizaciju. Centralizacija oko nule podrazumeva da od svakog obeležja (x) oduzimamo njegovu srednju vrednost kako bismo dobili i pozitivne, i negativne, x vrednosti. Za normalizaciju podataka, neophodno je da svaku komponentu obeležja podelimo sa njegovom dužinom, odnosno sa standardnom devijacijom tog obeležja. Izbor aktivacione funkcije može da zavisi od samog problema koji rešavamo. Pri izboru treba sagledati sve prednosti i mane aktivacionih funkcija. Najčešće se kao aktivaciona funkcija bira ReLU ili neke njene novije modifikacije, dok se tanh ređe koristi, a Sigmoid se više i ne koristi. Inicijalizacija težina je veoma važna, jer se na osnovu nje mreža može lepo trenirati ili može ništa da se ne desi. Težine nije dobro inicijalizovati na nule, jer će tada svi neuroni imati istu vrednost, pa će se svi identično ažurirati. Težine i bias-i se trebaju inicijalizovati na slučajne različite vrednosti.

73. Zašto na problemu klasifikacije slika koristimo konvolucione neuronske mreže (CNN), a ne klasične potpuno povezane mreže?

Za rešavanje problema klasifikacije slika koristimo CNN, a ne klasične potpuno povezane mreže, jer za slike imamo veliki broj ulaznih obeležja, a samim tim i velik broj parametara u klasičnim neuronskim mrežama. U ovom slučaju bi bilo gotovo nemoguće da pronađemo dovoljno podataka da ova mreža ne bi overfitt-ovala. Takođe, ne postoje računarski i memorijski resursi neophodni za treniranje ovolikih mreža.

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

Model konvolucionih neuronskih mreža se sastoji od kombinacije konvolucionih, pooling i potpuno povezanih slojeva, uz naravno ulazni i izlazni sloj. Na ulazu imamo 3D volume slike, gde su tri dimenzije uglavnom širina, visina i tri kanala boja. Naredni sloj je konvolucioni, po kom je CNN i dobila ime. U njemu se preko ulazne slike prevlače filteri kako bi se izdvojila pojedina obeležja. Filteri moraju biti iste dubine kao i ulazni volume, a prostorna dimenzija se može razlikovati, tj. ona predstavlja hiper-parametar. Prevlačenje filtera se vrši s leva na desno i od gore na dole, za odabrani korak (stride), i na svakoj poziciji slike dobijamo po jednu vrednost. Ova vrednost se dobija tako što vršimo skalarni proizvod težina filtera i ulaznih piksela. Za svaki od filtera kreiramo feature (aktivacionu) mapu, gde je svaka mapa dubine 1, pa je

dubina izlaza iz konvolucionog sloja jednaka broju primenjenih filtera. Obično se nakon konvolucionog sloja feature mapa propušta kroz aktivacionu funkciju (npr. ReLU). S vremena na vreme se posle konvolucionog sloja stavlja pooling sloj, u cilju smanjenja prostorne dimenzije. On ne menja dubinu volume-a, već samo smanjuje dimenziju, čime redukujemo parametre i filtriramo obeležja, kako bi sačuvali samo najbitnija. Dalje se uglavnom naizmenično ponavljaju konvolucioni i pooling slojevi, i na kraju uglavnom imamo potpuno povezane (fully connected) slojeve. Potpuno povezani sloj obezbeđuje povezanost između neurona kao u klasičnoj neuronskoj mreži, s tim što ćemo ovde imati znatno manji broj parametara zbog manjih dimenzija slike. Poslednji sloj se flatten-uje, tj. razvija se u vektor obeležja, i nakon toga imamo izlazni sloj kao i u klasičnoj neuronskoj mreži.

75. Kako se CNN obučavaju?

CNN mreže se obučavaju otprilike na isti način kao i klasične neuronske mreže, tj. koristi se back-propagation algoritam. Jedina razlika je što parametri koji se uče nisu težine ulaza i bias-i, već težine filtera, odnosno njih korigujemo kako bi dobili da se predikcije što bolje poklapaju sa očekivanim vrednostima. Učenje se vrši tako što se izračuna greška predikcije, koja zavisi od težina, i gleda se da se ta greška minimizuje. Greška predikcije predstavlja razliku stvarnih vrednosti i onoga što model predviđa. Za minimizovanje greške se traže težine za koje je greška najmanja, i za to se obično koristi algoritam gradijentnog spusta (gradient descent).

76. Objasniti konvolucioni sloj u CNN.

Konvolucioni sloj je sloj po kom je CNN dobila ime. U njemu se preko ulazne slike prevlače filteri kako bi se izdvojila pojedina obeležja. Filteri moraju biti iste dubine kao i ulazni volume, a prostorna dimenzija se može razlikovati, tj. ona predstavlja hiper-parametar. Prevlačenje filtera se vrši s leva na desno i od gore na dole, za odabrani korak (stride), i na svakoj poziciji slike dobijamo po jednu vrednost. Ova vrednost se dobija tako što vršimo skalarni proizvod težina filtera i ulaznih piksela. Za svaki od filtera kreiramo feature (aktivacionu) mapu, gde je svaka mapa dubine 1, pa je dubina izlaza iz konvolucionog sloja jednaka broju primenjenih filtera.

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

Broj slobodnih parametara u CNN mreži se smanjuje podešavanjem koraka (stride-a). Korak predstavlja broj piksela za koje se filter konvolucionog sloja pomera pri svakoj sledećoj iteraciji, tj. pri računanju svake sledeće pozicije feature mape. Pri izboru koraka, treba uvek voditi računa da se za odabrani korak može uklopiti filter u dimenzije slike. Takođe, parametri se smanjuju i primenom pooling sloja, kroz koje se vrši smanjivanje prostorne dimenzionalnosti ulaza.

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

Zero padding je proširivanje ivica nulama. On služi da bi se očuvala prostorna dimenzija, tj. da bi veličina izlaza bila jednaka veličini ulaza. Zero padding se obično mora vršiti jer filter konvolucionog sloja nije uspeo da pokrije sve piksele, konkretno piksele po ivicama, pa se na tim mestima unutar matrice postavljaju nule. Bez zero padding-a, veličina feature mapa bi se brzo smanjivala sa brojem slojeva i postepeno bi gubili informacije o originalnoj slici, što je veoma nezgodno ako imamo duboke mreže.

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

Slojevi sažimanja (pooling) služe za smanjivanje prostorne dimenzionalnosti, gde se dubina ne menja. Oni se koriste nezavisno nad svakom feature mapom u ulazu, a kao izlaz dobijamo reprezentaciju koja je lakša za obradu. Pomoću pooling slojeva se smanjuje broj parametara i imamo nezavisnost reprezentacije u nekoj regiji. Jedan od primera pooling sloja je Max Pooling. Filteri sažimanja nemaju parametre i često se gleda da se filteri ne preklapaju. Pooling sloj sažima reprezentaciju tako što se posmatra regija filtera i za tu regiju se bira neki broj koji je reprezentuje. Konkretno kod Max Pooling-a će se za svaku od regija odabrati najveća vrednost regije.

80. Šta je i čemu služi transfer learning?

Osnovna ideja trensfer learning-a je da u prethodnim slojevima mreže naučimo neka univerzalno dobra obeležja za klasifikaciju slika (npr. ivice su uvek korisne), a onda naknadno dotreniramo poslednje slojeve u zavisnosti od konkretnog

problema koji rešavamo. Koliko će se slojeva dotreniravati zavisi od toga na kom smo skupu učili i na kom ćemo skupu primeniti tu mrežu. Prednost transfer learning-a je da model možemo obučiti na manje podataka, a samim tim možemo izbeći i overfitting, jer je jedan od razloga da dođe do overfitting-a upravo nedostatak podataka.

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

Semantička segmentacija slike je metod čiji je ulaz slika, a izlaz je slika gde je svakom pikselu dodeljena jedna od predefinisanih kategorija. Primenom ovog procesa želimo da imamo regije slike u kojima su pikseli anotirani sa određenom semantičkom kategorijom.

82. Kakav nam je trening skup potreban da bismo vršili semantičku segmentaciju slika pomoću CNN (kako izgleda trening primer)?

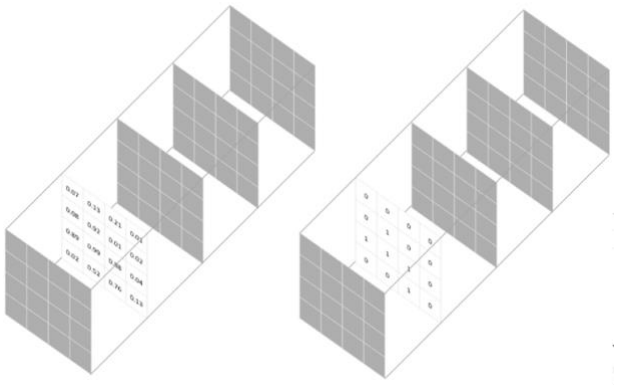
Da bismo vršili semantičku segmentaciju slika pomoću CNN-a, treba nam skup slika gde su za svaku sliku anotirani pikseli slike, tj. za svaki piksel je označeno kojoj klasi pripadaju. Pošto se anotacija vrši za sve piksele na slici, trening podaci za semantičku segmentaciju su veoma skupi.

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

Semantička segmentacija ne pravi razliku između instanci, već samo vodi računa o pikselima i njihovoj anotaciji. Metod koji bi mogao da reši ovaj problem je segmentacija instanci (instance segmentation).

84. Opisati ideju semantičke segmentacije (semantic segmentation) primenom CNN (objasnite kako se računa gubitak, kako se mreža trenira i izgled arhitekture).

Semantička segmentacija primenom CNN-a se vrši tako što se kreira konvoluciona neuronska mreža koja pravi predikcije za sve piksele odjednom. Kod ovog pristupa ne izdvajamo individualne isečke slika i klasifikujemo ih nezavisno, već posmatramo mrežu kao veliki stek konvolucionih slojeva, bez potpuno povezanih slojeva ili bilo čega drugog. Finalni sloj mreže ima dimenzije CxHxW, gde su H i W dimenzije originalne slike, a C je broj kategorija u našem sistemu. Ovaj sloj predstavlja rezultate klasifikacije za svaki piksel u ulaznoj slici. Za stvarne vrednosti takođe imamo kvadar istih dimenzija, gde su svi pikseli anotirani stvarnim vrednostima sa kojima se porede predikcije. U izlaznom sloju su po dubini raspoređene feature mape, gde jedna feature mapa predstavlja jednu kategoriju i sadrži verovatnoće svih piksela za tu kategoriju. Softmax će za svaku od kategorija da izračuna verovatnoću da piksel pripada toj kategoriji i anotiraće piksel onom klasom za koju dobije najveću vrednost verovatnoće.



Gubitak se računa pomoću cross-entropy loss funkcije. Za svak u sliku u trening skupom tažimo prosečan gubitak za sve piksele na njoj, a onda za čitav trening skup tražimo prosečan gubitak za svaku sliku unutar njega. Model se trenira kao i klasična i konvolutivna neuronska mreža, pomoću backpropagation algoritma. Kreiranje ovakve mreže dobijamo relativno jednostavan model koji bi uglavnom radio dobro, ali problem je što primenjujemo puno konvolucija, od kojih sve očuvavaju prostorne dimenzije, što čini postupak veoma skupim. Zbog ovoga u praksi retko kad imamo ovaj tip arhitekture, već koristimo downsampling, pa upsampling procese unutar mreže. Umesto da primenjujemo sve konvolucije celokupne prostorne dimenzije slike, prolazimo kroz mali broj konvolucionih slojeva koji rade na originalnoj slici. Zatim radimo downsampling dobijenih feature mapa (npr. pooling). Više puta kombinujemo konvoluciju i

downsample-ovanje, a nakon toga povećavamo prostornu rezoluciju u drugoj polovini mreže, tako da izlazna slika može biti iste veličine kao i ulazna. Za povećavanje prostorne rezolucije se koriste metode za upsampling, kao što su unpooling, transponovana konvolucija... Arhitektura ove mreže za semantičku segmentaciju je simetrična.

85. Objasnite kako se uz pomoć konvlucione neuronske mreže istovremeno rešava problem klasifikacije i lokalizacije (bounding box).

Problem klasifikacije i lokalizacije je problem u kom, pored predikcije kategorije objekta, treba da odredimo i gde se objekat nalazi, tj. da vratimo tačno jedan bounding box koji će oivičiti taj objekat. Razlika ovog problema u odnosu na detekciju objekta je što unapred pretpostavljamo da postoji jedna instanca objekta na slici koju ćemo klasifikovati. Rešavanjem problema klasifikacije i lokalizacije znamo da ćemo kao rezultat imati jednu klasifikacionu odluku i određen jedan bounding box. Arhitektura CNN-a koja bi rešavala ovaj problem je veoma slična arhitekturi klasične CNN mreže za klasifikaciju, s tim što imamo dva izlaza iz mreže. Jedan izlaz je class scores, tj. klasifikaciona odluka, a drugi je bounding box, tj. (x, y, w, h), gde su x i y koordinate gornjeg levog ugla, a w i h su širina i visina. Pošto imamo dva izlaza, tokom treniranja ćemo imati dva gubitka: classification loss i regression loss. Classification loss se koristi za proveru prediktovane klase i stvarne vrednosti (npr. softmax), a regression loss se koristi za proveru prediktovanih i stvarnih koordinata bounding box-a (npr. L2). Na kraju ove gubitke treba sabrati, što može biti nezgodno jer se uglavnom kreću u različitim opsezima, pa se može desiti da jedan gubitak izdominira. U praksi je preporuka da se prvo trenira klasifikacija, pa lokalizacija, ili obrnuto, pa se nakon pojedinačnog treniranja ukombinuju i treniraju kao celokupan sistem.

86. Objasnite problem detekcije objekata. Kako se ovom problemu pristupa primenom konvolucione neuronske mreže?

Problem detekcije objekata je problem u kom je cilj da pronađemo određene kategorije objekata na slici. Za svaku ulaznu sliku, kad god se jedna od predefinisanih kategorija pojavi na slici, želimo da iscrtamo bounding box oko nje i predvidimo kategoriju objekta unutar tog box-a. Razlika ovog problema u odnosu na klasifikaciju i lokalizaciju je što možemo imati različit broj izlaza za svaku ulaznu sliku, tj. ne znamo unapred koliko objekata očekujemo na slici. Pošto možemo imati više klasifikacionih odluka i bounding box-ova kao rezultat, ne možemo detekciju objekata posmatrati kao jednostavan regresioni problem, što je moglo kod lokalizacije. Postoje dva pristupa problemu detekcije objekata primenom CNN-a: region proposals algoritmi i pristup bez predlaganja regija. Region proposals ima za ideju da nađe listu regija kandidata, npr. pomoću selective search algoritma, i onda na svaku od pronađenih regija primenimo CNN, koja predviđa klasu i bounding box. Pristupi za detekciju objekata pomoću region proposals algoritma su: R-CNN, Fast R-CNN i Faster R-CNN. R-CNN predstavlja osnovnu implementaciju, gde se regioni pronađu, proslede CNN mreži i dobiju se rezultati. Problem kod R-CNN-a je što kandidati mogu različitih dimenzija, a trebali bi biti istih jer ih šaljemo na ulaz CNN za klasifikaciju. Rešenje ovog problema je da svaki region od interesa konvertujemo u kvadratne regije fiksne veličine. Takođe, problemi R-CNN-a su što je računarski dosta skupo, a samim tim i sporo, što možemo imati regije koje se preklapaju i što zavisimo od fiksiranog region proposals algoritma. Fast R-CNN je naprednija verzija R-CNN-a. Kod njega ne procesiramo svaki region od interesa odvojeno, već celu sliku propuštamo kroz konvolucione slojeve. Iz toga dobijamo feature mapu koja odgovara celoj slici, a onda iz nje izvlačimo regije kandidate pomoću region proposals algoritma. Ovo ubrzava proces jer ne vršimo skupe konvolucione operacije nad svakim isečkom. Kao i kod R-CNN-a, moramo da vršimo reshape-ovanje ulaze sa feature mape. Faster R-CNN je još naprednija verzija, gde više ne koristimo selection search kao region proposals algoritam, već mreža vrši predikciju koje su nam regije od značaja. Ulaznu sliku propuštamo kroz konvolucione slojeve i dobijamo feature mapu, kao kod Fast R-CNN-a, a onda pomoću odvojene region proposals mreže dobijamo classification loss da li je regija od interesa ili ne, kao i koordinate bounding box-a za tu regiju. Ostatak arhitekture izgleda kao i Fast R-CNN. U pristup detekcije objekata bez predlaganja regija spadaju YOLO i SSD algoritmi. Ideja kod njih je da tretiramo detekciju objekata kao regresioni problem i da napravimo sve predikcije odjednom, sa jednom velikom konvolucionom mrežom, umesto da se radi odvojeno procesiranje za svaku od potencijalnih regija. Kod ovih algoritama sliku delimo na mrežu ćelija i unutar svake ćelije zamislimo skup osnovnih bounding box-ova. Onda za svaku ćeliju i za svaki bounding box predviđamo offset osnovnog bounding box-a i classification scores, tj. u svakoj ćeliji rešavamo problem klasifikacije i lokalizacije. Na osnovu dobijenih rezultata, vršimo predviđanje nad celokupnom slikom. Region proposals algoritmi daju bolje rezultate detekcije objekata, ali pristup bez predlaganja regija je znatno brži.