

Skeletoni(Prepoznavanje ljudskog pokreta)

Marin Jezidžić

19. veljače 2023.

Sadržaj

1	Uvod	2
2	HAR	2
2.1	Povijest	2
3	Osnovna teoretska pozadina	2
3.1	HOSVD	3
3.2	Slučajno iteriranje potprostora	5
3.3	k-means	5
3.4	Diskriminacijska korelacijska analiza	7
3.5	Feedforward neuronska mreža	9
4	Prepoznavanje ljudskog pokreta	11
4.1	Opis skupa podataka	11
4.2	Metoda za prepoznavanje	12
5	Postignuti rezultati	13

1 Uvod

Prepoznavanje ljudskog pokreta, eng. Human Activity Recognition (HAR) jedna je od najinteresantnijih primjena umjetne inteligencije danas. Definiramo ju kao proces zaključivanja o aktivnosti čovjeka sa video sekvenci eng. action sequences koje nam u mnogo primjera opisuju radnju koju želimo predvidjeti. HAR spada u granu računalnog vida, eng. Computer Vision-a. Sa modernim poboljšanjima u softveru i hardveru računala rastu i mogućnosti implementacije HAR-a. Tu ključnu ulogu ima koncept Big data, tj. treniranje modela na velikim količinama podataka kojima, naravno dobivamo i bolje rezultate. Usko povezano s količinom podataka, raste i upotreba metoda dubokog učenja koje će nam generalno davati bolje rezultate na velikoj količini podataka u odnosu na ostale prediktivne algoritme. Sam proces ekstrakcije relevantnih podataka je vrlo kompliciran te se stoga može naići na mnogo različitih pristupa problemu, sa raznih stajališta. Npr. možemo promatrati ljudsko tijelo kroz nekoliko ključnih točaka te promatrati kretanje tih točaka kroz vrijeme. Također, interesantno je promatrati i nekakav fizički 'aspekt' u kojemu u obzir uzimamo i akceleraciju, kut nasuprot kamere, itd. Još par riječi o samome konceptu rada. Kroz rad ću dati neke teoretske rezultate za metode koje sam koristio, no pojedine ćemo 'namjerno' izostaviti, kako bi održao balans između teoretskog i praktičnog dijela samoga rada. Projekt je rađen u programskom jeziku Matlab, te će i sintaksa biti prikladna tomu.

2 HAR

2.1 Povijest

Povijest prepoznavanja ljudske aktivnosti (HAR) spada u područje računarstva i umjetne inteligencije, datira još od 1960-ih i 1970-ih. Rana istraživanja na ovom području su se prvenstveno fokusirala na korištenje tehnika računalnog vida za prepoznavanje i praćenje ljudskih pokreta.

Jedan od prvih primjetnih primjera ranog rada na području HAR-a je projekt "Human movement recognition system" koji je proveo MIT (Massachusetts Institute of Technology) u suradnji s industrijskim partnerima, gdje su koristili kameru i računalo za prepoznavanje i praćenje ljudskih pokreta u radnom okruženju.

Tijekom 1980-ih i 1990-ih, hardver je napredovao, što je omogućilo razvoj novih pristupa i metoda za prepoznavanje ljudske aktivnosti, uključujući statističke pristupe, pristupe bazirane na strojnom učenju i pristupe bazirane na umjetnoj inteligenciji.

U ranim 2000-ima, s pojavom nosivih senzora i širenjem pametnih telefona, istraživanja u području prepoznavanja ljudske aktivnosti (HAR) počela su se preusmjeravati prema korištenju podataka iz senzora ugrađenih u nosive uređaje i pametne telefone. Ova promjena omogućila je istraživačima da prikupljaju podatke o ljudskoj aktivnosti u stvarnom vremenu i na različitim lokacijama.

S razvojem tehnologije mobilnih uređaja, senzora i mreža, prepoznavanje ljudske aktivnosti postalo je sve dostupnije i široko korišteno u različitim aplikacijama, kao što su poboljšanje zdravlja, poboljšanje sigurnosti, automatizacija rada, itd.

3 Osnovna teoretska pozadina

Sada ćemo dati neke teoretske rezultate za metode koje ćemo koristiti u idućim poglavljima. Cilj rada je svakako iskoristiti neke tenzorske metode kako bismo uočili neke praktične primjene tenzorskih dekompozicija. Prvo ćemo se baviti generalizacijom SVD eng. Singular value decomposition metode na više-dimenzionalne podatke, tj. HOSVD. Nakon toga, uvesti ćemo metodu slučajnih iteracija potprostora, koja će nam iz naših 'rijetkih' podataka dati uporabljive podatke koje ćemo dalje koristiti. Također, reći ćemo nešto i o k-meansu u kontekstu u kojem ćemo ga koristiti. Na kraju, uvesti ćemo metodu Diskriminacijske korelacijske analize, vjerojatno i najapstraktnije metode koju ćemo koristiti. Teoretsku pozadinu klasifikatora nećemo obrađivati zbog njihove tehničke prirode, te kompleksnosti.

3.1 HOSVD

Intuitivno, tenzori su matrice višeg reda. Nama će u radu trebati tenzori reda 3, no primjeri za višu dimenziju se analogno poopćavaju.

Definicija 2.1.1 Tenzor $A \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ reda N sastoji se od elemenata $a_{i_1 \dots i_N}$.

Sada se standardno uvode pojmovi moda, unfolda i folda i tenzorskog množenja.

Definicija 2.1.2 Vektor moda n tenzora reda N dobit ćemo fiksiramo li sve indekse osim n -tog. Njegova duljina jednaka je I_n .

Definicija 2.1.3 S $A_{(n)} \in \mathbb{R}^{I_n(I_1 \dots I_N)}$ označavamo matricu čiji su stupci vektori moda n . Kažemo da je ona rezultat otpakiravanja eng. unfold tenzora A po n -toj dimenziji, odnosno u n -tom modu koristimo li tenzorsku terminologiju.

Fold je inverzna operacija od unfold i zahtjeva deskriptor (dimenzije tenzora kojega 'kreiramo').

Sada možemo opisati i potrebne operacije s tenzorima.

Definicija 2.1.4 Skalarni produkt dva tenzora $A, B \in \mathbb{R}^{I_1 \dots I_N}$ definira se kao $A \cdot B = \sum_{i_1} \dots \sum_{i_N} a_{i_1 \dots i_N} b_{i_1 \dots i_N}$. Tada se norma na standardan način definira.

Definicija 2.1.5 Skalarni produkt dva tenzora $A, B \in \mathbb{R}^{I_1 \dots I_N}$ definira se kao $A \cdot B = \sum_{i_1} \dots \sum_{i_N} a_{i_1 \dots i_N} b_{i_1 \dots i_N}$. Produkt moda n tenzora $A \in \mathbb{R}^{I_1 \dots I_N}$ i matrice $U \in \mathbb{R}^{I_n(I_1 \dots I_N)}$, u oznaci $A \otimes_n U$, je tenzor $B \in \mathbb{R}^{I_1 \dots I_N}$ čiji su elementi

$$(A \otimes_n U)_{i_1 \dots i_{n-1} i_{n+1} \dots i_N} = \sum_{i_n} a_{i_1 \dots i_n} u_{i_n i_{n+1} \dots i_N}.$$

Još nam treba Kroneckerov produkt.

Definicija 2.1.6 Kroneckerov produkt matrica $A \in \mathbb{R}^{i \times j}$ i $B \in \mathbb{R}^{k \times l}$, je matrica dimenzija $ik \times jl$ definirana s

$$C = A \otimes B = \begin{bmatrix} A_{1,1}B & A_{1,2}B & \dots & A_{1,j}B \\ A_{2,1}B & A_{2,2}B & \dots & A_{2,j}B \\ \vdots & \vdots & \ddots & \vdots \\ A_{i,1}B & A_{i,2}B & \dots & A_{i,j}B \end{bmatrix}$$

Bitno je naglasiti da Kroneckerov produkt nije komutativan, tj. $A \otimes B$ nije jednako $B \otimes A$.

Sada smo definirali sve potrebne alate za HOSVD.

Teorem 2.1.7 Tenzor $A \in \mathbb{R}^{I_1 \dots I_N}$ može se zapisati u obliku rastava

$$A = S \times_1 \mathbf{U}^{(1)} \times_2 \mathbf{U}^{(2)} \dots \times_N \mathbf{U}^{(N)}$$

gdje su $\mathbf{U}^{(n)} \in \mathbb{R}^{I_n \times i_n}$ ortogonalne matrice. Tenzor S ima iste dimenzije kao početni tenzor A te ga nazivamo jezgri tenzor. Također, on ima svojstvo ortogonalnosti, tj. svaka dva podtenzora koja dobijemo fiksiranjem nekog elementa su okomita u smislu skalarnog produkta ranije definiranog.

Dokaz. Prvi korak je da odredimo SVD matrica nastalih unfoldanjem zadanog tenzora A u svakom modu, tj.

$$A_i = U^{(i)} \Sigma^{(i)} V^{(i)T}, i = 1, 2, \dots, N$$

Tada jezgru S definiramo kao

$$S = A \times_1 \mathbf{U}^{(1)T} \times_2 \mathbf{U}^{(2)T} \dots \times_N \mathbf{U}^{(N)T}.$$

Preostaje nam pokazati da su kriške od S međusobno okomite te da singularne vrijednosti opadaju u svakom modu.

Jednadžba se sada može zapisati u sljedećem obliku

$$A_{(n)} = U^{(n)} S_{(n)} (U^{(n+1)} \otimes \mathbf{U}^{(n+1)T} \otimes \mathbf{U}^{(n+2)T} \dots \otimes \mathbf{U}^{(N)} \times \mathbf{U}^{(1)} \otimes \mathbf{U}^{(2)} \dots \otimes \mathbf{U}^{(n-1)})^T.$$

Označimo s r_n najveći indeks za koji je $\sigma_{r_n+1}^{(n)} > 0$. Ortogonalnost Kroneckerovog faktora sada povlači raspis

$$S_{(n)} = \Sigma^{(n)} V^{(n)T} (U^{(n+1)} \otimes \mathbf{U}^{(n+1)T} \otimes \mathbf{U}^{(n+2)T} \dots \otimes \mathbf{U}^{(N)} \times \mathbf{U}^{(1)} \otimes \mathbf{U}^{(2)} \dots \otimes \mathbf{U}^{(n-1)})^T.$$

Slijedi da za proizvoljne ortogonalne matrice $\mathbf{U}^{(i)}$ vrijedi

$$\langle S_{i_n=a}, S_{i_n=b} \rangle = 0 \text{ za } a \neq b,$$

$$\sigma_1^{(n)} \geq \sigma_2^{(n)} \geq \dots \sigma_{I_n}^{(n)} \geq 0,$$

te za $r_n < I_n$

$$\sigma_{r_n+1}^{(n)} = \dots = \sigma_{I_n}^{(n)} = 0$$

Čime je tvrdnja dokazana.

3.2 Slučajno iteriranje potprostora

Metoda iteriranja potprostora je ništa drugo no poopćenje metode potencija za aproksimiranje 'više' svojstvenih vrijednosti. Htjeli bismo l dimenzionalni invarijantni potprostor V reprezentiran kao slika matrice $V \in \mathbb{C}^{n \times l}$ takve da je $V^*V = I_l$. Kako je V A -invarijantan, možemo ga zapisati kao $AV = VM$, gdje je $M = V^*AV$ matrični Rayleighov kvocijent. Sada matricu V , čiji stupci razapinju A -invarijantan potprostor, dopunimo do unitarne, te onda koristeći invarijantnost nam transformacija sličnosti daje rezultat da su svojstvene vrijednosti od M ujedno i svojstvene vrijednosti od A . Naime, ako je $My = \lambda y$, onda $AV = VM$ povlači $A(Vy) = VM y = \lambda(Vy)$. Osnovna metoda iteriranja potprostora se dobiva trivijalnim poopćavanjem metode potencija tako da umjesto polaznog vektora $x^{(0)} \neq 0$ uzmemo l linearno nezavisnih vektora $x^{(1,0)}, x^{(2,0)}, \dots, x^{(n,0)}$ poslažemo ih u $n \times l$ matricu $X^{(0)} = (x^{(1,0)}, \dots, x^{(l,0)})$ i u k -toj iteraciji računamo $X^{(k)} = A^k X^{(0)}$.

Sada možemo i zapisati konkretan algoritam

Algorithm 1 Iteriranje potprostora

```

 $X^{(0)} = Y^{(0)} R^{(0)}$  ( $QR$  faktorizacija)
 $k = 0$ 
while uvjet nije zadovoljen do
     $X^{(k+1)} = AY^{(k)}$ 
     $X^{(k+1)} = Y^{(k+1)} R^{(k+1)}$  ( $QR$  faktorizacija)
     $k = k + 1$ 
end while

```

Primijetimo kako možemo uzeti i manje od l stupaca matrica $X^{(k)}$ i $Y^{(k)}$, te bismo na taj način samo 'smanjili' dobiveni potprostor.

Također, uzimanjem matrice Ω iz standardne normalne razdiobe, dobivamo modifikaciju opisanoga algoritma koju nazivamo Slučajno iteriranje potprostora.

Dokaz konvergencije ćemo preskočiti, no možemo primijetiti da se u cijeloj priči koristi QR faktorizacija koja nam za danu matricu X vraća matricu ortogonalnu matricu Q i gornjetrokutastu matricu R koje u umnošku daju X .

3.3 k-means

Algoritam k - sredina eng. k -means je heuristika koja rješava NP problem razvrstavanja podataka u skupine, tj. klastere s unaprijed zadanim brojem klastera. Bez obzira na svoju 'lokalnost' algoritam je vrlo popularan. Mi ćemo ga koristiti kako bi konstruirali rječnik vizualnih riječi, eng. Codebook. Koncept je potpuno analogan kao i rječnik koji imamo u u klasteriranju članaka. Doduše, može se na različite načine proširiti, interpretirati, te će nam stoga predstavljati i najveći izazov u samome problemu HAR-a.

Algoritam k - sredina spada u nenadzirano učenje, tj. on će sam morati donesti odluku o tome u koji će skup pojedinu točku/podatak ubaciti. Odmah možemo primijetiti da će nam glavni problem predstavljati određivanje tih referentnih točaka, tj. centara. Sam algoritam će se iterirati sve dok ne postignemo stabilnost,

tj. do trenutka kada nam se koordinate centara prestanu značajno mijenjati. Tu dolazimo do istaknutog problema lokalnosti, tj. kao i kod većine algoritama, doći ćemo samo do lokalnoga minimuma.

Standardno se funkcija troška definira kao suma kvadrata udaljenosti točke od pripadnog centra. Drugim riječima, algoritam generira klastere sa minimalnom totalnom varijancom.

Najopćenitije, neka je $X = \{x_1, \dots, x_n\}$ u n dimenzionalnom Euklidskom prostoru \mathbb{R}^n . Cilj algoritma je sada pronaći, za unaprijed zadani broj $k \geq 2$, optimalnu k -particiju skupa X , $\pi = \{C_1, C_2, \dots, C_k\}$, tj. naći optimalan raspored m točaka/podataka skupa X u k klastera. Svakom klasteru $C_i, i = 1, \dots, k$, pridružena je točka

$$c_i = \frac{1}{|C_i|} \sum_{x \in C_i} x$$

koju nazivamo centrom. Primijetimo kako centar ne mora nužno biti u skupu X . Također, možemo vidjeti da za centar uzimamo centroid, odnosno težište skupa.

Particiji skupa X , $\pi = \{C_1, C_2, \dots, C_k\}$ pridružena je vrijednost funkcije cilja

$$J = J(\pi) = \sum_{i=1}^k \sum_{x \in C_i} \|x - c_i\|_2^2$$

Dakle cilj nam je minimizirati vrijednost funkcije J . Kao što je rečeno i ranije, postoje razne varijacije funkcije cilja koje su pogodne za pojedini scenarij. Mi ćemo se koristiti standardnom, euklidskom udaljenosti.

Zapišimo sada i algoritam.

Algorithm 2 k-means

Ulaz: Skup točaka X , broj klastera k
Izlaz: Centri c_1, \dots, c_k , skupina pojedine točke x_1, \dots, x_m
 Inicijaliziraj centre c_1, \dots, c_k
 Inicijaliziraj početne pozicije točaka, tj. $a_i \leftarrow 0$ za sve i
while uvjet nije zadovoljen **do**
 Osvježi pozicije točaka: $a_i \leftarrow \arg \min_{j=1}^k \|x_i - c_j\|^2$ za sve i
 Osvježi centre: $c_j \leftarrow \frac{1}{\sum_{i=1}^m [c_i = j]} \sum_{i=1}^m [c_i = j] x_i$ za sve j
end while
vrati $c_1, \dots, c_k, a_1, \dots, a_m$

Slično kao kod algoritma iteriranja potprostora, broj klastera k i centri se mogu prilagoditi, tj. izabrati na temelju neke prethodne analize eng. educated guess.

Dokaz konvergencije je nešto lakši nego kod iteriranja potprostora te ćemo ga stoga i dokazati.

Teorem 2.3.1 Funkcija cilja J algoritma k-sredina je opadajuća.

Dokaz

Označimo s $J^{(l)}$ vrijednost ciljne funkcije algoritma k -sredina u l -toj iteraciji. Vrijedi:

$$\begin{aligned} J^{(l)} &= \sum_{i=1}^k \sum_{x \in C_i^{(l)}} \|x - c_i^{(l)}\|_2^2 \geq \sum_{i=1}^k \sum_{x \in C_i^{(l)}} \|x - c_{p(x)}^{(l)}\|_2^2 = \\ &= \sum_{i=1}^k \sum_{x \in C_i^{(l+1)}} \|x - c_{p(x)}^{(l)}\|_2^2 \geq \sum_{i=1}^k \sum_{x \in C_i^{(l)}} \|x - c_{p(x)}^{(l)}\|_2^2 = J^{(l+1)} \end{aligned}$$

Gdje je $p(x)$ pridruženi klaster točki x u trenutnoj iteraciji.

Time smo dokazali da zadani algoritam konvergira prema nekom lokalnom minimumu.

3.4 Diskriminacijska korelacijska analiza

Diskriminacijska korelacijska analiza jedna je od nedavno 'predloženih' metoda za tzv. feature fusion postupak kojim se kombiniranjem dva različita prediktora dobiva jedan zajednički prediktor. Cilj metode je reducirati korelaciju među različitim klasama poboljšavajući tako performanse klasifikatora.

Metoda se može koristiti u bilo kakvom problemu prepoznavanja obrasca za spajanje različitih prediktora dobivenih ih različitih uvjeta. DCA je prva metoda koja uzima u obzir strukturu prediktora pri spajanju.

Metoda ima vrlo malu kompleksnost te je stoga prikladna za 'real-time' upotrebe.

Opišimo sada kompletan postupak.

Pretpostavimo da u našem problemu imamo c različitih klasa, tj. kategorija. Nadalje, n stupaca je razdvojeno na c različitih grupa, gdje n_i stupci pripadaju i -toj klasi:

$$n = \sum_{i=1}^c n_i$$

Neka $x_{ij} \in X$ predstavlja prediktorski vektor koji se odnosi na j -ti uzorak i -te klase, tj. grupe.

Sada definiramo marginalne aritmetičke sredine:

$$\begin{aligned} \bar{x}_i &= \frac{1}{n_i} \sum_{j=1}^{n_i} x_{ij} \\ \bar{x} &= \frac{1}{n} \sum_{i=1}^c \sum_{j=1}^{n_i} x_{ij} = \frac{1}{n} \sum_{i=1}^c n_i \bar{x}_i \end{aligned}$$

Matricu gubitaka, tzv. class scatter matrix definiramo kao

$$S_{bx(p \times p)} = \sum_{i=1}^c n_i (\bar{x}_i - \bar{x})(\bar{x}_i - \bar{x})^T = \nu_{bx} \nu_{bx}^T$$

$$\Phi_{bx(p \times c)} = [\sqrt{n_1}(\bar{x}_1 - \bar{x}), \sqrt{n_2}(\bar{x}_2 - \bar{x}), \dots, \sqrt{n_c}(\bar{x}_c - \bar{x})]$$

Sada ako su klase već dobro separirane, matrica $\Phi^T \Phi$ će biti dijagonalna. Trivijalno vidimo da je ona pozitivno semidefinitna pa ju možemo dijagonalizirati:

$$P^T (\Phi_{bx}^T \Phi_{bx}) P = \hat{\Lambda}_{(r \times r)}$$

Sada standardno odabiremo r najznačajnijih svojstvenih vektora matrice S_{bx} sa preslikavanjem $Q \rightarrow \Phi_{bx} Q$:

$$(\Phi_{bx} Q)^T S_{bx} (\Phi_{bx} Q) = \hat{\Lambda}_{(r \times r)}$$

Ako je $W_{bx} := \Phi_{bx} Q \Lambda^{-1/2}$, imamo:

$$W_{bx}^T S_{bx} W_{bx} = I$$

Te iz toga $X'_{(r \times n)} = W_{bx(r \times p)}^T X_{(p \times n)}$ što je projekcija matrice X na prostor gdje je matrica gubitaka identiteta, tj. klase su separirane.

Slično, za drugi prediktor $W_{by}^T S_{by} W_{by} = I$, te $Y'_{(r \times n)} = W_{by(r \times q)}^T Y_{(q \times n)}$.

Iako vrijedi $S'_{bx} = S'_{by} = I$, matrice $\Phi'_{bx}{}^T \Phi'_{bx}$ i $\Phi'_{by}{}^T \Phi'_{by}$ su striktno dijagonalno dominirane, tj. apsolutna vrijednost svakog elementa na dijagonali je veća nego suma svih ostalih, nedijagonalnih elemenata te matrice. Zbog toga 'centri' klasa imaju minimalnu međusobnu korelaciju, te su time klase separirane.

Preostaje nam minimizirati korelaciju među pripadnim elementima prediktorima X i Y . S tim ciljem dijagonaliziramo među-prediktorsku kovarijacijsku matricu $S'_{xy} = X' Y'^T$

$$S'_{xy(r \times r)} = U \Sigma V^T \rightarrow U^T S'_{xy} V = \Sigma$$

$$\begin{pmatrix} W_{cx} \\ U \Sigma^{-1/2} \end{pmatrix}^T S'_{xy} \begin{pmatrix} W_{cy} \\ V \Sigma^{-1/2} \end{pmatrix} = I$$

Sve skupa, transformacije su

$$X^* = W_{cx}^T X' = W_{cx}^T W_{bx}^T X = W_x X$$

$$Y^* = W_{cy}^T Y' = W_{cy}^T W_{by}^T Y = W_y Y$$

Spajanje prediktora: $Z_1 = \begin{bmatrix} X^* \\ Y^* \end{bmatrix}$

Te također, $Z_2 = X^* + Y^*$.

Na taj smo način dobili dva različita načina spajanja prediktora.

Primijetimo kako u real-time problemu mi možemo separirajući prostor konstruirati na temelju trening skupa podataka, te nakon toga testne podatke samo projicirati na taj prostor, i na taj način postići 'poboljšanje' u predikciji.

Mi ćemo metodu koristiti za spajanje prostorne i vremenske reprezentacije ljudskog skeletona.

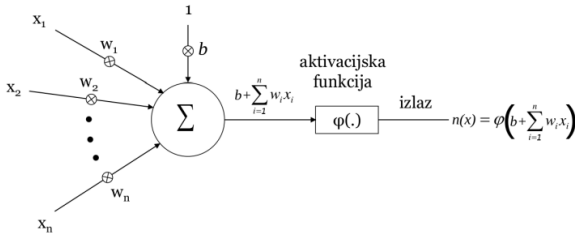
3.5 Feedforward neuronska mreža

Umjetna, tj. artificial neuronska mreža osnovni je primjer Neuronske mreže. Mogućnosti Neuronskih mreža su vrlo velike što potvrđuje i činjenica da je trenutno najpoznatiji 'proizvod' umjetne inteligencije, Chatgpt napravljen koristeći Neuronske mreže. U ovome radu ćemo se baviti samo umjetnim Neuronskim mrežama, no napomenimo kako postoji nekoliko arhitektura, svaka sa primjenom u vlastitom području. Kako ne bismo morali uvoditi previše pojmova, navesti ćemo samo osnovna svojstva i dati algoritam koji smo koristili.

Standardna početna priča oko ovoga tipa neuronske mreže je da se ona bazira na biološkim neuronskim mrežama. Dakle osnovna gradivna jedinica biološke neuronske mreže je neuron ili živčana stanica. Dendriti su osnovne veze iz tijela neurona. Oni dovode živčane impulse iz drugih živčanih stanica. Akson je 'dugačka' veza koja se proteže iz tijela neurona i služi za prijenos živčanih impulsa. Na kraju aksona nalaze se terminali koji povezuju stanicu s drugim neuronima. Najjednostavniji oblik neuronske mreže je Perceptron.

Perceptron ima strukturu sličnu živčanoj stanici. Uvedimo sada potrebne oznake:

Neka su $x = (x_1, x_2, \dots, x_n)$ izlazne informacije iz susjednih neurona, vrijednosti $b, w = (w_1, \dots, w_n)$ parametri neurona, a $n(x)$ izlazna vrijednost pojedinog neurona.



Perceptron matematički definiramo kao funkciju $n : \mathbb{R}^n \rightarrow \{0, 1\}$ oblika

$$n(x) = \varphi(\langle w, x \rangle + b), x \in \mathbb{R}^n,$$

gdje su $b \in \mathbb{R}, w \in \mathbb{R}^n$ parametri perceptrona, $\varphi : \mathbb{R} \rightarrow \mathbb{R}$ aktivacijska funkcija te $\langle \cdot, \cdot \rangle$ standardni Euklidski skalarni produkt. Izbor aktivacijske funkcije uveliko utječe na ponašanje perceptrona. Mi ćemo koristiti sigmoidnu aktivacijsku funkciju koja daje izlaznu vrijednost između 0 i 1.

$$\varphi(x) = \frac{1}{1 + e^{-x}}$$

Napomenimo kako je ovo samo jedan primjer aktivacijske funkcije. Svaka ima svoje prednosti i nedostatke.

Primijetimo kako je ovo ipak regresijski izlaz, tj. kontinuirani, te će se za klasifikaciju vršiti zaokruživanje na najbliže cijelo. Cijela priča može se izbjeći ubacivanjem SoftMax sloja koji će osigurati 'cijeli' izlaz. Bitno je naglasiti kako se SoftMax funkcija koristi i kod Naive Bayes klasifikatora.

Napišimo sada algoritam kojime optimiziramo težine na perceptronu.
Neka je zadan skup podataka S i konstanta ≥ 0 .

Algorithm 3 Optimizacija perceptrona

Ulaz: Skup točaka (x, y) , skup težina w
Izlaz: optimalne težine w
Inicijaliziraj težine na nasumične vrijednosti
while uvjet nije zadovoljen **do**
 izračunaj: $o = n(x)$
 if $o == y$ **then**
 nastavi sa sljedećim primjerkom podatka
 else
 $b = b + (y - o)$
 $w = w + (y - o)x$
 end if
end while
vрати w

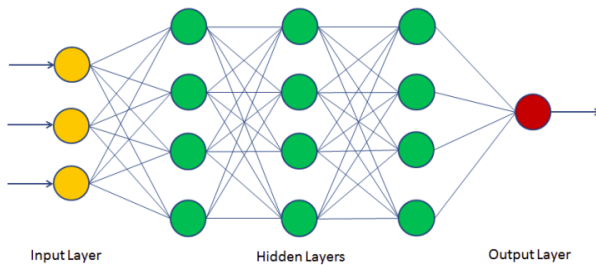
Primijetimo kako ponuđeni algoritam za optimizaciju težina perceptrona ne konvergira za podatke koji nisu linearno odvojivi. Zato uvodimo algoritam gradijentnog spusta, eng. Gradient descent.

Algorithm 4 Gradijentni spust

Ulaz: Skup točaka (x, y) , skup težina w
Izlaz: optimalne težine w
Inicijaliziraj težine na nasumične vrijednosti
while uvjet nije zadovoljen **do**
 for svaki element iz skupa podataka **do**
 izračunaj: $o = n(x)$
 nastavi sa sljedećim primjerkom podatka
 $db = (t - o)$
 $dw = (t - o)x$
 end for
end while
 $b = b + db$
 $w = w + dw$
vрати w

Gradijentni spust konvergira prema globalnom minimumu čak i kada podatci nisu linearno separabilni.

Sada kada smo uveli pojam perceptrona, priča se lagano poopći kombiniranjem više neurona na način da perceptrone shvatimo kao čvorove, i povežemo, dobijemo nekakav povezan graf koji se sastoji od slojeva. Dimenzija sloja je broj perceptrona u sloju. Neuronska mreža sastoji od ulaznog sloja, skrivenih slojeva, te izlaznog sloja. Ulazni sloj ima dimenziju jednaku dimenziji 'feature' vektora, tj. količini prediktora. Na sljedećoj slici možemo vidjeti opisani proces.



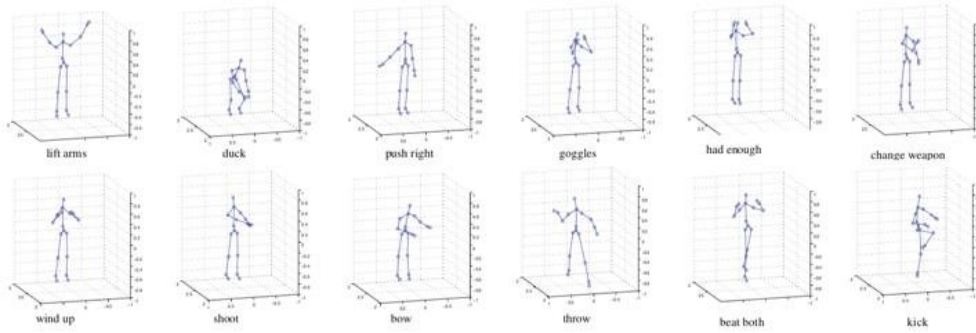
Ovo je primjer acikličke neuronske mreže. Kada koristimo sintagmu 'crna kutija' za neuronsku mrežu, referiramo se na postavljanje težina po skrivenim slojevima, što nije sasvim jasno kako se događa.

Najpoznatiji algoritam učenja acikličke neuronske mreže je eng. Backpropagation algoritam koji slijedi sličnu logiku kao i algoritam gradijentnog spusta. Kao što mu i ime sugerira, parametri perceptrona u mreži se korigiraju unazad sloj po sloj, počevši od izlaznog sloja, preko skrivenih slojeva pa sve do ulaznog sloja. Postupak se ponavlja za cijeli skup podataka i dok se ne zadovolji unaprijed postavljen uvjet preciznosti.

4 Prepoznavanje ljudskog pokreta

4.1 Opis skupa podataka

U našem primjeru koristili smo skup podataka $MSRC_{12}$, sa 12 različitih pokreta, sve skupa $6h : 40m$ video zapisa podijeljeno u 594 video zapisa. Zadani su nam videozapisi pomoću excel tablica u kojima pišu koordinate ključnog dijela ljudskog tijela u datom trenutku. Sve skupa se zapisuje 20 ključnih dijelova ljudskog tijela koje se prepoznaje pomoću kinetičkog senzora. Važno je naglasiti kako je korišteni skup podataka 'najteži' oblik skupa podataka za problem $HAR - a$ zbog varijabilnosti unutar klasa. Na slici možemo vidjeti osnovne pokrete.



4.2 Metoda za prepoznavanje

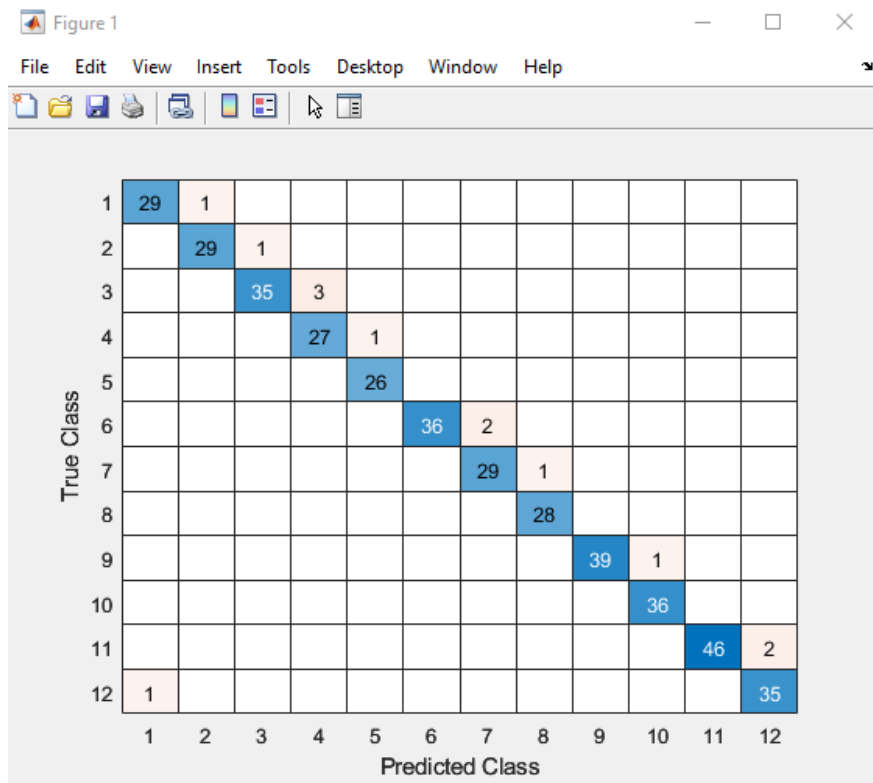
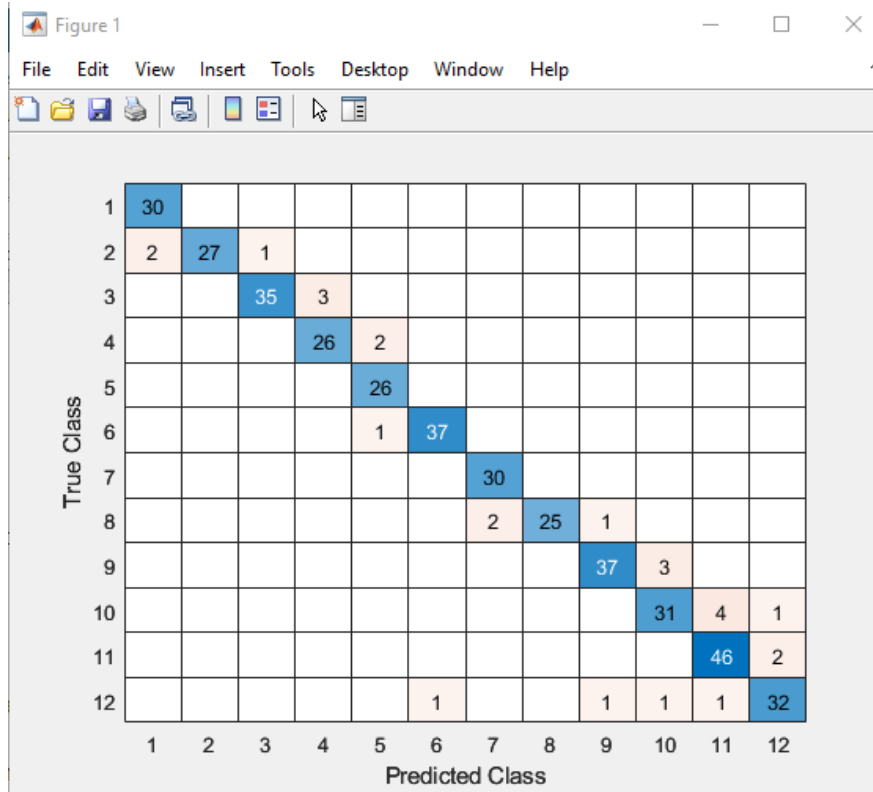
Predložena metoda od strane tima stručnjaka iz članka prema kojemu smo radili navodi kako opisani problem još uvijek nema optimalno rješenje. Različiti algoritmi bolje pašu u različitim scenarijima, to je i očekivano. Naša metoda zasniva se na reprezentaciji svakog frame-a videozapisa kao rijetkog binarnog tenzora trećeg reda. Nadalje, koristimo dvije reprezentacije, prostornu i vremensko prostornu. Prostorna reprezentacija nam predstavlja trenutnu lokaciju ključnih dijelova ljudskog tijela u datom frameu, dok vremensko prostorna reprezentacija daje sve lokacije gdje su ključni dijelovi bili do toga trenutka. To postizemo tako da u trodimenzionalni tenzor zadanih dimenzija ubacimo vrijednost 1 na mjesta gdje su ključne točke ljudskog tijela, a inače nulu. Idući korak je da te tenzore velikih dimenzija sada maksimalno reduciramo, tako da budu uporabljivi, no htjeli bismo sačuvati rijetki uzorak. U tu svrhu se koristi gore opisana metoda slučajnog iteriranja potprostora. Idući korak je poigrati se sa HOSVD metodom za dobivenim tenzorima kako bismo ekstrahirali takozvane *MSV* mean singular values. Naime, kao što znamo od gore HOSVD dekompozicija nam daje redukciju dimenzije u obliku sortiranog jezgrenog tenzora koji je istih dimenzija kao početni, no ima jako svojstvo međusobne ortogonalnosti. Vektori su unutar tog tenzora poredani u silaznom poretaku po svojoj važnosti te se uzima određeni postotak najbitnijih vrijednosti, analogno kao kod SVD-a. Kada smo to napravili, konstruiramo skup kodova. To je ništa drugo no još jedna metoda redukcije dimenzije. Na trening skupu izaberemo optimalan broj klastera i provedemo k-means algoritam, kao što je gore opisano. Dobivene centre spremimo, te za testne podatke računamo udaljenosti od tih formiranih centara i tako ih klasificiramo. Kada se opisani proces provede, preostaje nam odrediti histogramsku reprezentaciju, tj. pobrojati koliko se puta koji događaj dogodio u nekom video zapisu. Tako dobiveno rješenje se šalje na posljednju redukciju dimenzije, te kombiniranje oba prediktora koristeći Diskriminacijsku korelacijsku analizu kao gore. Ona će umanjiti korelaciju među klasama, na već opisani način te time poboljšati rezultate klasifikacije. Napomenimo kako je DCA metoda nadziranoga učenja, tj. ona će uzeti i labele pojedinog videa. Sve skupa predajemo klasiufikatoru neuronske mreže. U radu je korištena neuronska mreža sa 2 skrivena sloja.

Ovako opisanom metodom postižu se rezultati koji nadmašuju sve do 'tada' predložene metode, te je glavna prednost što je metoda primjenjiva, tj. jako se brzo izvršava, što je čini pogodno za real time upotrebu.

5 Postignuti rezultati

Sada ćemo dati rezultate koje smo mi dobili. Za usporedbu rezultata dobivenih sa ANN, koristiti ćemo metodu slučajnih šuma. Nju nećemo posebno opisivati, no ona je ništa drugo nego ansambl stabala odlučivanja. Dakle sve skupa imamo 12 kategorija u podacima, tj. 12 pokreta. Mi smo uzeli prvih 204 video zapisa iz zadanoga skupa podataka, te od uzetih 8 instanci unutar svakoga video zapisa smo ih 6 koristili za trening, a dva za testiranje. Napomenimo kako je potrebna dodatna dorada metode da bi mogla prepoznavati općenitiju okolinu, tj. još mnogo videozapisa, te npr. DTW za balansiranje broja frame-a po uzorku. Također napomenimo kako će korištena umjetna neuronska mreža davati bolje rezultate tek kada povećamo uzorak.

Odmah možemo primijetiti kako slučajne šume nadmašuju metodu umjetnih neuronskih mreža. Napomenuti ćemo da smo koristili optimalnu mrežu, za 'tjuniranim' parametrima, te također i sa slučajnim šumama.



Literatura

- [1] Panagiotis Barmpoutis , Tania Stathaki and Stephanos Camarinopoulos. *Skeleton-Based Human Action Recognition through Third-Order Tensor Representation and Spatio-Temporal Analysis*.
- [2] Dimitropoulos, K. Barmpoutis, P. Kitsikidis, A. Grammalidis, N. *Classification of multidimensional timeevolving data using histograms of Grassmannian points*