Identifikacija b-jetova na LHC-u pomoću neuronskih mreža

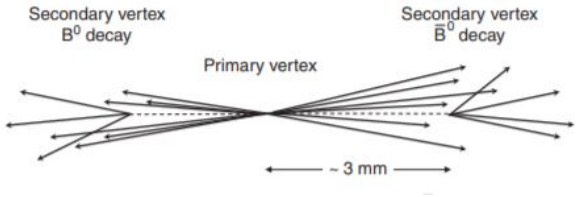
Luka Klinčić*
Fizički odsjek, Prirodoslovno-matematički fakultet, Bijenička 32, Zagreb
(Dated: 26. siječnja 2020.)

Prilikom raspršenja hadrona na visokim energijama opažamo mlazove čestica kojima možemo mjeriti energiju, impuls, distribuciju mase itd. U praksi nam je korisno znati od kojih čestica potječu ti mlazovi, a unutar ovog rada bavit ćemo se metodama prepoznavanja i razvrstavanja mlazova koji potječu od bottom kvarka.

I. UVOD

Unutar Large Hadron Collidera (LHC) uzrokuju se sudari protona pri TeV-skim energijama prilikom kojih opažamo hadronske mlazove (eng. *jets*). Mlazovi nastaju procesom hadronizacije te svaki od njih može nastati od bilo kojeg kvarka ili gluona te je njihovo podrijetlo u principu teško odrediti.

Unatoč tome, mlazovi nastali od bottom ili b-kvarkova su ponešto drugačiji i lakše ih je prepoznati nego nastale od ostalih okusa. Tome je tako jer hadroni koji sadrže b kvark imaju relativno dugo vrijeme života u odnosu na ostale, reda veličine $1.5 \cdot 10^{-12}$s.[1] Uračunavši i efekt dilatacije vremena, hadroni s b kvarkovima putuju nekoliko milimetara prije nego se dalje raspadnu. Iz tog razloga umjesto jednog, imamo dva konusa mlaza, prvi nastao raspršenjem u točki sudara, a drugi raspadom b hadrona, što je grafički prikazano na slici 1. Taj fenomen je karakterističan za mlazove nastale od b kvarkova i pri njihovom prepoznavanju oslanjamo se na mogućnost razlučivanja primarnog od sekundarnog vrha konusa.



Slika 1: Grafički prikaz primarnog i sekundarnog vrha.[1]

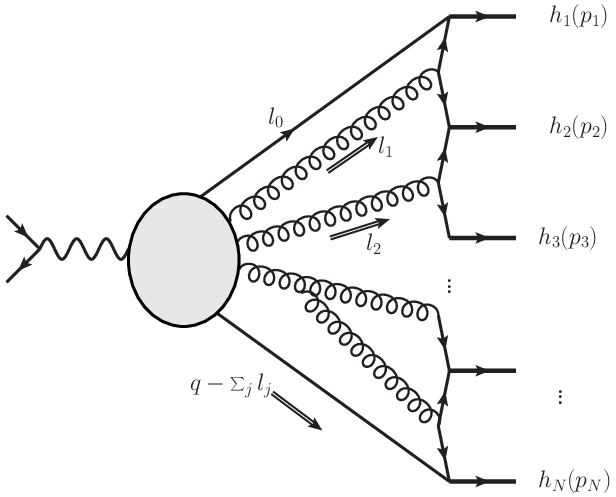
Mlazovi hadrona karakterizirani su određenim fizikalnim veličinama kao što su transverzalni impuls p_T, energija, invarijantna masa, smjer, širina itd. *B-jetove* moguće je prepoznavati nametanjem niza rezova na skup varijabli, što se pokazalo neefikasnim i računalno skupim procesom. Kako bi se taj dugotrajan proces zaobišao, zadnjih godina se u tu svrhu koriste duboke umjetne neuronske mreže, što je kulminiralo algoritmom DeepCSV iz 2018. godine.[2] Mi ovim seminarskim radom želimo sastaviti vlastitu neuronsku mrežu, pomoću nje iskušati koliku efikasnost možemo dobiti na vlastitom skupu podataka te provjeriti utječu li sve varijable mlaza jednakom mjerom na točnost klasifiricanja mlazova hadrona.

II. TEORIJA

U ovom poglavlju ukratko ćemo opisati fizikalnu pozadinu *b-tagginga* te neke teorijske koncepte koje je korisno poznavati.

II.1. Hadronizacija

Kvarkovi nikad nisu opaženi van hadrona, što se u teoriji kvantne kromodinamike (QCD) navodi kao hipoteza zatočenja kvarkova i gluona. Hipoteza zatočenja navodi kako su sva kvantna stanja bojni singleti odnosno da se slobodno propagiraju samo bojno neutralne čestice. Hipoteza još nije formalno dokazana, no zasad objašnjava neopažanje slobodnih kvarkova. To nas navodi na pitanje što se događa kada hadronu pridamo energiju veću od njegove energije vezanja.



Slika 2: Grafički prikaz hadronizacije.[3]

* lklincic.phy@pmf.hr

Proces još nije potpuno objašnjen, no heuristički, zbog privlačne sile između gluona potencijalna energija raste linearno s udaljenošću između kvarkova te ubrzo postane povoljnije stvoriti dva kvark-antikvark para nego održati dva slobodna kvarka. U tom slučaju imamo dva nova hadrona. Ako početnom hadronu pridamo jako visoku energiju (kao na ubrzivačima čestica) proces stvaranja parova ne staje te se nastavlja i na novim hadronima. Tada nastaju mlazovi hadrona koje opažamo na detektorima te se proces naziva hadronizacijom.

II.2. Parametri mlaza

Kao što smo spomenuli u prijašnjem odlomku, mlazove karakteriziramo određenim fizikalnim veličinama koje možemo mjeriti. U našem slučaju, koristili smo transformaciju tih parametara pod imenom *generalized angularities* definirane na sljedeći način

$$\lambda_\beta^\kappa = \sum_{i \epsilon jet} z_i^\kappa \theta_i^\beta \tag{1}$$

gdje su z_i i θ_i definirani kao

$$z_i = \frac{p_{Ti}}{\sum_{j \epsilon jet} p_{Tj}} \tag{2}$$

$$\theta_i = \frac{R_{i\hat{n}}}{R}. \tag{3}$$

U ovom zapisu, z_i je zapravo omjer transverzalnog impulsa pojedine čestice s ukupnim transverzalnim impulsom mlaza. θ_i je, s druge strane, omjer udaljenosti čestice od osi mlaza u $\eta - \phi$ prostoru s konstantom R koja je radijus mlaza i u našem slučaju iznosi 0.4.

Mi smo promatrali *angularitiese* cjelobrojnih eksponenata i u rasponu od 0 do 2 te su fizikalne interpretacije nekih od njih u nastavku.

$$\lambda_0^0 \Rightarrow \text{multiplicitet mlaza}, \tag{4}$$

$$\lambda_0^1 = 1 \Rightarrow \text{trivijalni parametar}, \tag{5}$$

$$\lambda_0^2 = (p_T^D)^2 \Rightarrow p_T^D = \sqrt{\frac{\sum p_T^2}{(\sum p_T)^2}}^5, \tag{6}$$

$$\lambda_1^1 \Rightarrow \text{širina distribucije mase}^6, \tag{7}$$

$$\lambda_2^1 \approx \frac{m_{jet}^2}{E_{jet}^2} \Rightarrow \text{masa odnosno potisak}. \tag{8}$$

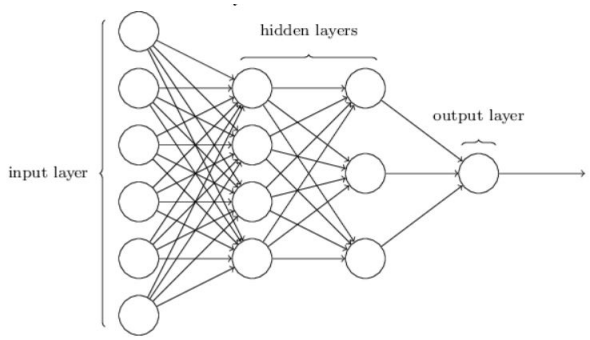
III. METODE RADA

U svrhu klasifikacije jetova, ideja nam je bila koristiti metode strojnog učenja. Strojno učenje je skup računalno-statističkih paradigmi kojima koristimo sam skup podataka tj. prilagodbu na njih kako bismo izvršili neki zadatak, umjesto seta instrukcija koje bismo morali zadati koristeći klasično programiranje.

Strojno učenje je vrlo širok pojam te obuhvaća velik broj algoritama za klasifikaciju i regresiju podataka te predikciju oznaka, no u ovom radu mi ćemo se držati jednog područja pod nazivom **umjetne neuronske mreže**.

III.1. Umjetne neuronske mreže

Umjetna neuronska mreža (kolokvijalno neuronska mreža) jest algoritam inspiriran radom neurona u mozgu, koji koristi sustav čvorova (neurona) povezanih težinskim faktorima i pristranostima kako bi aproksimirao traženu funkciju ili klasificirao podatke. Umjetna neuronska mreža shematski je prikazana na slici 3.



Slika 3: Shematski prikaz neuronske mreže.[7]

Obična *feed-forward* neuronska mreža sastoji se od slojeva s određenim brojem neurona - ulaznog sloja, skrivenih slojeva i izlaznog sloja. Neuroni predstavljaju funkcije koje kao nezavisnu varijablu primaju težinski zbroj izlaza neurona u sloju prije njih (odnosno aktivacija) te mu dodaju određenu pristranost i taj broj koriste kao ulaz za određenu aktivacijsku funkciju. Matrično to možemo zapisati kao (9).

$$\mathbf{x}^{(n+1)} = f(\mathbf{W}\mathbf{x}^{(n)} + \mathbf{b}) \tag{9}$$

Aktivacijska funkcija služi kako bismo izlaze neurona "stisnuli" u interval manji od potencijalno cijelog skupa realnih brojeva, ali i bitnije, aktivacijske funkcije koje nisu identitet uvode nelinearnost u model neuronske mreže te omogućavaju aproksimiranje nelinearnih funkcija. Dapače, teorem o univerzalnoj aproksimaciji govori kako feed-forward neuronska mreža s jednim skrivenim slojem može aproksimirati bilo koju neprekidnu funkciju definiranu na kompaktnom podskupu $\mathbb{R}^n$, uz blage pretpostavke na aktivacijsku funkciju.[8] Aktivacijske funkcije koje ćemo koristiti u ovom seminaru su ispravljena linearna i sigmoidalna.

	λ_0^0	λ_1^0	λ_2^0	λ_0^1	λ_1^1	λ_2^1	λ_0^2	λ_1^2	λ_2^2	B-jet
count	5478.000000	5478.000000	5478.000000	5.478000e+03	5478.000000	5478.000000	5478.000000	5478.000000	5478.000000	5478.000000
mean	6.061701	2.917876	1.744179	1.000000e+00	0.394265	0.203560	0.263143	0.085038	0.037589	0.500000
std	1.364884	0.949713	0.762121	6.462513e-08	0.103119	0.079772	0.079100	0.028560	0.018556	0.500046
min	5.000000	0.588933	0.093315	9.999998e-01	0.071170	0.008284	0.074152	0.009694	0.001135	0.000000
50%	6.000000	2.762933	1.636655	1.000000e+00	0.401550	0.203063	0.250241	0.083448	0.035886	0.500000
max	18.000000	9.116858	6.999252	1.000000e+00	0.662898	0.450626	0.788921	0.188065	0.106308	1.000000

Slika 4: Neka statistička obilježja skupa značajki.

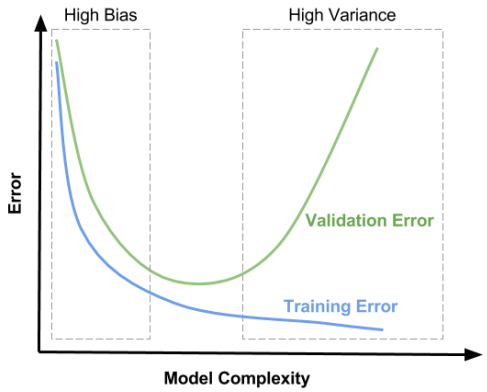
Slično kao i kod ostalih modela strojnog učenja, neuronska mreža radi na principu "treniranja". Naš skup podataka dijeli se na određen broj primjera od kojih svaki posjeduje neke značajke, odnosno karakteristike podataka te oznake pripadnosti određenoj klasi. Princip rada neuronske mreže svodi se na postavljanje vrijednosti ulaznog sloja neurona na vrijednost značajki pojedinog primjera, računanje aktivacija neurona u skrivenim slojevima na način opisan u (9) te uspoređivanjem predikcije neuronske mreže s postojećim oznakama primjera.

Treniranje mreže je zapravo optimizacija parametara (težinskih faktora i pristranosti) mreže. Cilj treniranja (kao i općenito strojnog učenja) je pronaći skup parametara modela koji radi najbolju predikciju oznaka. U svrhu mjere točnosti predikcije koristimo tzv. **funkciju gubitka**, čiji minimum želimo postići. Unutar strojnog učenja koriste se raznolike funkcije gubitka, najjednostavnija od kojih je kvadratna pogreška, no kad imamo posla s binarnim (Bernoullijevim) varijablama u izlazu, bolje rješenje je gubitak unakrsne entropije (10) budući da njime manje kažnjavamo jako točno klasificirane primjere

$$L(y, h(\mathbf{x})) = -y \ln h(\mathbf{x}) - (1-y) \ln (1 - h(\mathbf{x})) \qquad (10)$$

gdje je $h(\mathbf{x})$ hipoteza odnosno izlaz neuronske mreže, a y su oznake primjera.

Bitno svojstvo je mogućnost generalizacije neuronske mreže - želimo točnu predikciju oznake i na primjerima koje nismo koristili za učenje mreže. Međutim, ako dovedemo pogrešku na skupu za treniranje do minimuma, model će se savršeno prilagoditi svim dostupnim podacima, pa tako i sveprisutnom šumu te povećati vjerojatnost pogrešne klasifikacije neviđenih primjera. Iz tog razloga kada provjeravamo performansu neuronske mreže, jedan dio podataka rezerviramo i ne treniramo mrežu na njima, nego ih koristimo za validaciju. Ta metoda zove se unakrsna provjera (eng. *cross-validation*) i tipičan graf pogrešaka na različitim skupovima prikazan je na slici 5.



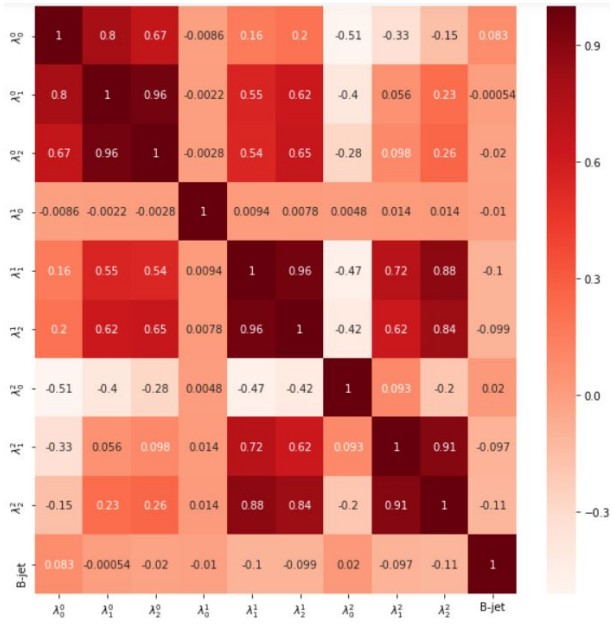
Slika 5: Primjer ovisnosti pogreške na skupu za treniranje i testiranje.[9]

Valja spomenuti da se povećanjem broja parametara mreže povećava nelinearnost modela tj. mogućnost prilagodbe šumu što rezultira većoj sklonosti prenaučenosti. Kod dubokih neuronskih mreža (više skivenih slojeva), taj problem postaje pogotovo izražen te mu je potrebno priskočiti metodama regularizacije kao što su ograničavanje norme vektora značajki i deaktivacija nasumičnih čvorova u mreži.

IV. REZULTATI

U ovom seminarskom radu koristili smo umjetno generirane podatke iz generatora događaja PYTHIA. Naš skup podataka sastojao se od 5478 primjera (događaja) od kojih je 50% bilo označeno kao poteklih od b-jetova. Skup 9 značajki sastojao se od *generalized angularitiesa* s koeficijentima κ i β imeđu 0 i 2.

Prije nego se počnemo baviti neuronskom mrežom, dobra je praksa statistički obraditi podatke i steći neko razumijevanje o njima. Na slici 4. vidimo tablični prikaz osnovnih statističkih veličina vezanih uz skup podataka, kao što su prosjek, standardna devijacija, minimum, 50. percentil i maksimum. Odavdje možemo procijeniti jesu li nam svi primjeri shodni za klasificiranje i imaju li statistički značaj. To možemo prosuditi po značajki λ_0^0, čija je najmanja vrijednost 5. Budući da je to multiplicitet



Slika 6: Grafički prikaz korelacijskih koeficijenata između pojedinih značajki i oznake.

mlaza, to nam je u redu jer ne želimo kao set za učenje koristiti mlazove s anomalnim brojem čestica.

Potom smo htjeli saznati nešto o značajkama i njihovoj potencijalnoj multikolinearnosti. U tu svrhu za početak smo računali Pearsonov koeficijent korelacije r

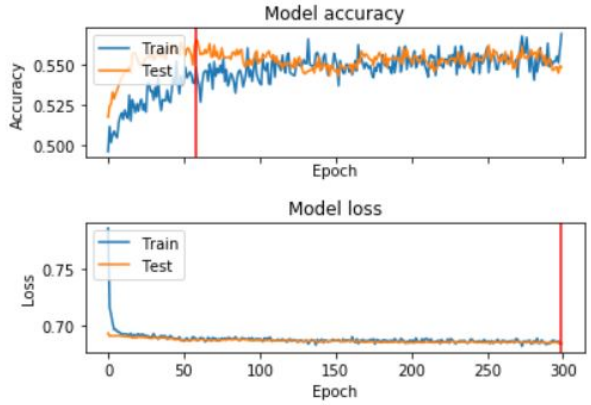
$$r = \frac{\mathrm{cov}(\lambda_j^i, \lambda_l^k)}{\sigma_{\lambda_j^i}\sigma_{\lambda_l^k}} \tag{11}$$

odnosno omjer kovarijance dviju značajki s umnoškom njihovih standardnih devijacija. Korelacijski koeficijenti između značajki prikazani su tablično na slici 5. U ovom prikazu možemo primijetiti nekoliko zanimljivih rezultata. Trivijalna značajka je nekorelirana s ostalima, što je razumljivo, no primjećujemo da je λ_0^2, značajka koja opisuje transverzalni impuls, u prosjeku negativno korelirana s koeficijentima manjeg faktora κ. Također, primjetimo da su varijable λ_β^κ i $\lambda_{\beta+1}^\kappa$ visoko korelirane, pogotovo za $\beta = 1$. To bi u nekim slučajevima moglo značiti kako su te varijable suvišne i jednu po paru je moguće izbaciti, no prilikom klasifikacije taj problem nije toliko izražen pa ih mi nećemo izbacivati. Na poslijetku, možemo primjetiti kako nijedna pojedina varijabla nije značajno korelirana s oznakom "B-jet", te ćemo odabir najbitnijih značajki morati učiniti drugačije.

Dobra praksa prilikom obrade podataka je i skalirati ih. Ako imamo jako velike razlike u rasponu magnitude pojedinih značajki, to će se odraziti i na težinske faktore te će toj značajki dati veću važnost bez opravdanja. Naše značajke skalirali smo tako da smo im aritmetičke sredine postavili na nulu, a standardne devijacije na 1.

Neuronsku mrežu gradili smo u Python biblioteci keras. Keras je intuitivan za izgradnju neuronskih mreža budući da koristeći razred Sequential možemo na jednostavan način (medodom .add) dodavati i slojeve neuronske mreže koji mogu varirati od potpuno povezanih Dense slojeva, preko Dropout regularizacijskih do konvolucijskih.

Mi smo tijekom glavnine rada na seminaru koristili plitku neuronsku mrežu, s dva skrivena sloja, redom 30 i 10 neurona. I s relativno malenim brojem neurona imali smo problema s prenaučenošću, pa smo stoga liberalno koristili regularizaciju. Koristili smo *dropout* regularizaciju s faktorom 0.2 nakon ulaznog sloja te 0.5 nakon skrivenih slojeva. Aktivacijska funkcija svih slojeva bila je "relu" odnosno ispravljena linearna, osim izlazne, koja je sigmoidalna. Kao što smo već spomenuli, funkcija gubitka bila je unakrsne entropije, a optimizacija adaptivnog momenta odnosno "adam".



Slika 7: Ovisnost točnosti klasifikacije i gubitka u ovisnosti o broju epoha.

Prilikom građenja neuronske mreže eksperimentirali smo s različitim arhitekturama te hiperparametrima kao što su broj epoha i veličina "batcha". Da bismo izbjegli prenaučenost, skup podataka podijelili smo na skup za treniranje i test u omjeru 80-20. Većinu vremena trenirali smo model na 300 epoha, no bilježili bismo epohu s maksimalnom vrijednošću preciznosti kao optimalnu, čiji primjer možemo vidjeti na slici 7.

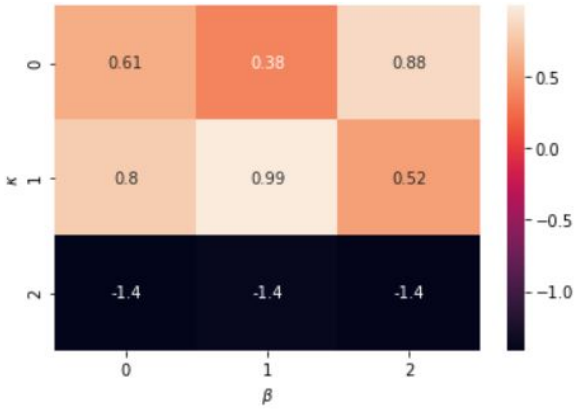
Iako postoje mnoge mjere performanse mreže kao preciznost, odziv i f1, zbog podjednakog broja b-jetova i ne b-jetova mi smo koristili mjeru točnosti, odnosno udio točno klasificiranih primjera u ukupnom broju. Kako bismo mogli izračunati ikakvu statistiku rezultata, mrežu smo testirali 10 puta te iz tog skupa izračunali prosjek i standardnu devijaciju dane jednadžbom (10).

$$\eta = 0.586 \pm 0.002 \tag{12}$$

Nakon što smo izgradili našu mrežu, napokon smo htjeli provjeriti koliko koja značajka utječe na točnost klasifikacije mreže. Kod neuronskih mreža to je nešto

teže izmjeriti nego kod ostalih modela unutar strojnog učenja, no mi smo koristili metodu koja se bazira na eliminaciji pojedine značajke te treniranja i validacije mreže na novom skupu.

Redom smo u svim primjerima iz skupa za treniranje pojedinu značajku stavili na nulu te bilježili kako se mijenja točnost klasifikacije od značajke do značajke. Ako točnost klasifikacije značajno padne, to znači da je ta značajka bitna za rad mreže. Dobivene rezultate smo standardno skalirali i prikazali grafički na slici 8.



Slika 8: Utjecaj pojedine značajke na točnost klasifikacije mreže.

V. ZAKLJUČAK

Cilj ovog seminara bio je razmotriti i pobliže se upoznati s metodama identifikacije i klasifikacije b-jetova pritom koristeći neuronske mreže. Promotrili smo fizikalnu pozadinu nastanka mlazova hadrona i mehanizam prepoznavanja mlazova.

Testirali smo koliko pojedini parametri mlaza utječu na efikasnost mreže te iz našeg eksperimenta pokazalo da su najutjecajniji parametri s koeficijentom $\kappa = 2$. To donekle ima smisla budući da je koeficijent κ upravo potencija transverzalnog impulsa. Budući da je b kvark relativno masivan, očekujemo da će produkti njegovog raspršenja imati velik p_T, odnosno da će transverzalni impuls biti važna značajka u prepoznavanju b-jetova.

Unutar opsega ovog seminara nismo uspjeli dobiti mrežu koja efikasno razdvaja b-jetove od ostalih. Unatoč tome, postotak točnosti koji smo postigli je konzistentno iznad nasumičnog razvrstavanja i nije neočekivan budući da se mreže koje su korisne u praksi treniraju na skupovima mnogo većim nego što je naš. Stoga bi se rad mreže mogao unaprijediti korištenjem većeg skupa podataka s više primjera koji jesu b-jet, te viših transverzalnih impulsa. U tom slučaju postoji mogućnost da bi naša mreža pogodovala od dodatnih skrivenih slojeva.

ZAHVALE

Htio bih zahvaliti docentu Nikoli Poljaku i asistentu Marku Jerčiću na pruženoj pomoći, savjetima i vodstvu tijekom izrade ovog seminara.

[1] M. Thomson, *Modern Particle Physics* (Cambridge University Press, Cambridge, 2013) p. 24-25
[2] The CMS Collaboration, *Identification of heavy-flavour jets with the CMS detector in pp collisions at 13 TeV*, [arXiv:1712.07158]
[3] J. Collins, T.C. Rogers, *Graphical Structure of Hadronization and Factorization in Hard Collisions*, 2018, [arXiv:1801.02704v2]
[4] P. Gras, S. Hoche, D. Kar, A. Larkoski, L. Lonnblad, S. Platzer, A. Siodmok, P. Skands, G. Soyez, and J. Thalerl, *Systematics of quark/gluon tagging*, arXiv:1704.03878v2 [hep-ph]
[5] The CMS Collaboration, *Search for the standard model Higgs Boson in the decay channel* $H \to ZZ \to l^- l^+ q\bar{q}$ *at CMS*, CMS PAS HIG-11-006, 2011, p.6
[6] S. Marzani, G. Soyez, M. Spannowsky, *Looking inside jets: an introduction to jet substructure and boosted-object phenomenology*, Lecture Notes in Physics, volume 958 (2019),p. 74-75 , [arXiv:1901.10342]
[7] http://neuralnetworksanddeeplearning.com/chap1.html
[8] G. Cybenko, *Approximation by Superpositions of a Sigmoidal Function*, Math. Control Signals Systems (1989) 2:303-314
[9] https://towardsdatascience.com/cross-validation-a-beginners-guide-5b8ca04962cd