Eesti Infotehnoloogia Kolledž

Andmekaeve osakestefüüsikas

Diplomitöö

Tudeng: Margus Pärt
Juhendaja: Mario Kadastik

Tallinn 2017

AUTORIDEKLARATSIOON

Deklareerin, et käesolev diplomitöö, mis on minu iseseisva töö tulemus, on esitatud Eesti Infotehnoloogia Kolledžile lõpudiplomi taotlemiseks Infosüsteemide arendamise erialal. Diplomitöö alusel ei ole varem eriala lõpudiplomit taotletud.

Autor Margus Pärt …….....

Töö vastab kehtivatele nõuetele

Juhendaja Mario Kadastik…….

Sisukord

1. Sissejuhatus

Uue avastuse tegemise võib jaotada 3 sammu:

1. Hüpotees (mis võib tekkida mõne nähtuse jälgimisest või matemaatilisest kirjeldusest).
2. Katsed ja saadud tulemuste võrdlemine hüpoteesi ennustusega.
3. Tulemuste tõlgendamine, teooria sõnastamine.

Hüpoteeside kinnitamine algosakeste füüsikas käib tänapäeval põhiliselt osakestekiirendist saadud andmete või kosmose vaatlusel saadud andmete võrdlusel olemasolevate teadmistega ja simulatsioonidega. Kõrge energiaga kosmilise kiirguse mõõtmine ja mõõtmistest järelduste tegemine on keerukas, sest huvitavad sündmused toimuvad harva. Osekestekiirendid võimaldavad teha palju kontrollitud katseid, aga energia (13 TeV), millel interkatsioonid toimuvad on umbes 80 000 korda väiksem.

Diplomitöö autor töötas praktika ajal ja diplomitöö valmimise ajal KBFI-s (Keemilise ja Bioloogilise Füüsika Instituut) osakonnas, mille teadlased ja töötajad tegelevad füüsikaliste protsesside jälgmise ja teoreetiliste teadmiste edendamisega koostöös CERNiga (Euroopa Tuumauuringute Organisatsioon) keskendudes algosakeste füüsikale. Käesolev diplomitöö on autori praktikatöö täienenud versioon, mis kirjeldab osa protsessist, kuidas rakendatakse infotehnoloogiat mikromaailma loodusseaduste paremini mõistmise eesmärgil.

Diplomitöö käigus õppis autor füüsikat, masinõpet ja andmekaevet. Diplomitöö käigus kirjutas autor Bashi, Pythoni, Ruby ja C++ süntaksit:

1. Tehisliku närvivõrgu topoloogia implementeerimine Pythonis kasutades TensorFlow liidestust. Graafiline haldusliides TensorFlow-le, mis kombineeris närvivõrkude struktuure leidmaks sobivaim klassifikaator, salvestas tulemused ja võimaldas hilisemat mugavamat analüüsi. Töö käigus loodud kood ilma sisendandmete ja tulemusteta on kättesaadav koodi versioonihaldusest. [1]
2. Panustas CERN-i ja KBFI teadlaste koostöös valminud raamistiku arendusse, mis toetas hajusarvutusi (andmete jaotamine, sündmuste filtreerimine, tulemuste valideerimine). [2]

[1] https://bitbucket.org/mxrguspxrt/kbfi_tensorflow
[2] https://github.com/HEP-KBFI/tth-htt/

3. Osales konkreetse lagunemiskanali uurimises (2 müüonit, 1 jet, 1 b-jet). [3]

Värskeim versioon dokumendist on kättesaadav autori kodulehelt.[4]

Käesolev diplomitöö koosneb kaheksast teemast:

1. Sissejuhatus.
2. Domeenispetsiifilised teadmised (sissejuhatus algosakeste füüsikasse).
3. Andmete saamise kirjeldus.
4. Masinõppe rakendamine ja tulemused (närvivõrgu ja otsustuspuude võrdlus).
5. Hajusarvutuste jaotamine (ja tulemuste valideerimine).
6. Konkreetse lagunemiskanali uurimise projekt.
7. Kokkuvõte
8. Lisad.

Tehtud praktika ja diplomitöö tulemusena:

1. Leiti närvivõrgu mudel ja selle konfiguratsiooniparameetrid, mis suudab pakkuda kvaliteetsemat signaali eristust mürast võrreldes seni kasutusel olnud otsustuspuu mudeli konfiguratsiooniga.
2. Aidatati muuta hajusarvutusi toetavat raamistikku efektiivsemaks. Infrastruktuuri (andmekandjad, võrk) piirangutega toimetulek, parem veakindlus (täiendatud logi, automaat-testid).
3. Korrati analüüsi dokument “CMS AN-15-289” järgi, mis kirjeldas 2012 aasta andmest leitud liiasust 28.4 GeV juures. Diplomitöö käigus kasutati võrdluseks 2015 ja 2016 aasta andmeid. [5]

Diplomitöö eesmärk on kirjeldada osa tarkvara arenduse protsessist, kuidas toimub uue algosakese avastamine kasutades masinõpet ja andmete filtreerimist.

Diplomitöö autor tänab kõiki KBFI-st, CERN-ist ja Eesti Infotehnoloogia Kolledžist, kes leidsid aega, et õpetada, täpsustada ja suunata, et selle tulemuseni jõuda. [6]

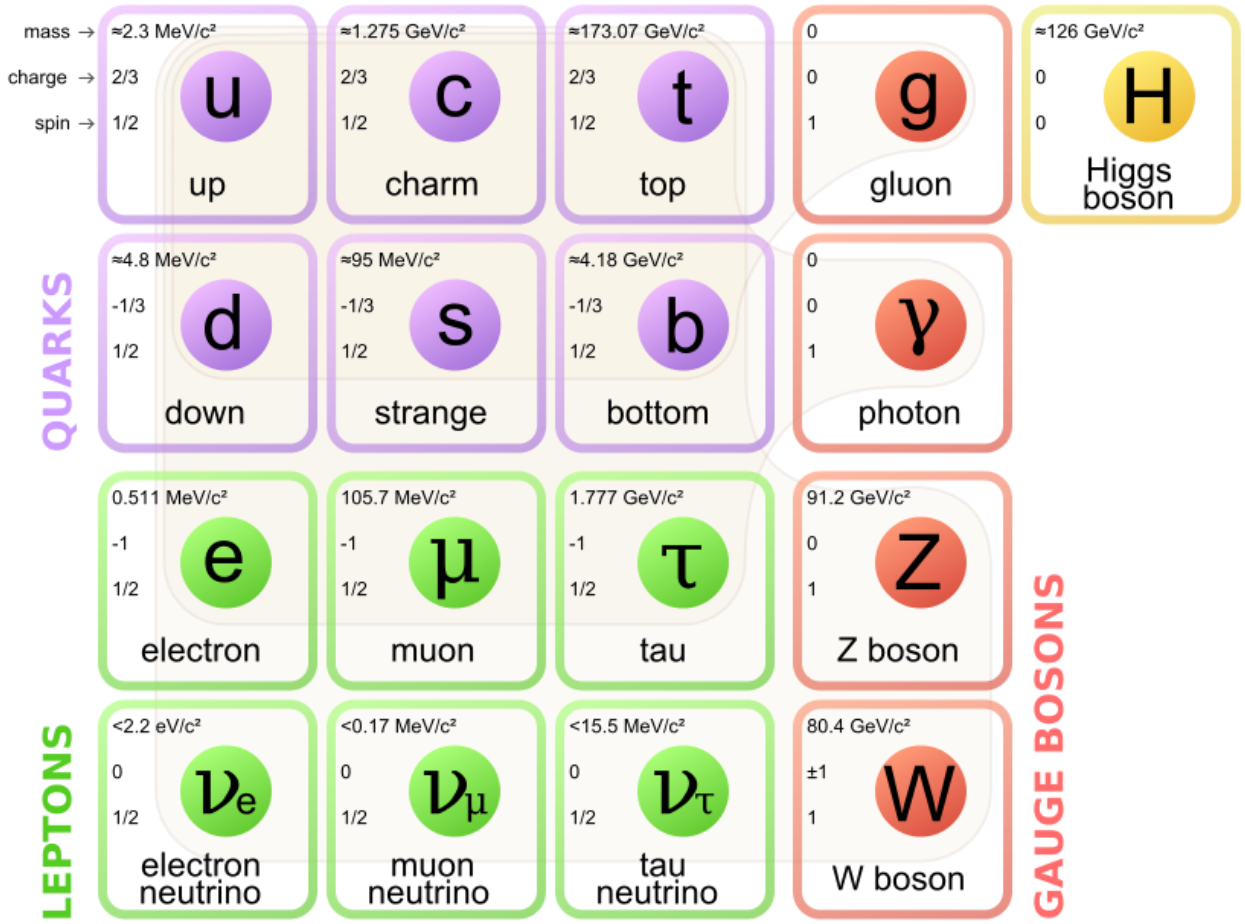
[3] https://github.com/HEP-KBFI/2mu1b1j
[4] https://drive.google.com/drive/folders/0BwepVMFFQ-MVVUI5NHVYdk1tRDg
[5] Dokumendile pole viidatud, sest see on mõeldud ainult CERN-i sisemiseks kasutuseks.
[6] Mario, Christian ja Andres, te olete minu superkangelased!

2. Sissejuhatus algosakeste füüsikasse

Algosakeste füüsika on pidevalt arenev teadusharu, mis kirjeldab ja üritab leida vastuseid küsimustele:

- Millistest algkomponentidest moodustub universum meie ümber?
- Millised on nende omadused - mass, laeng ja spinn?
- Millised on nendevahelised jõud (gravitatsioon, elektromagnet, nõrk ja tugevjõud) ning kuidas algosakesed looduses toimuvates protsessides interakteeruvad?

Standardmudel kirjeldab teadaolevad algosakesed:



Tabel "Standardmudel". Tabelist on puudu osakeste antipartnerid, kvarkide värvuste info ja looduses jälgitud tumeaine, tume energia. [20]

Kvantfüüsika kirjeldab protsesse, kus ühikud on diskreetsed ja väiksem vahemik on Plancki konstant. Ei teata, kas ruum ja aeg on pidevad või diskreetsed ja mis juhtub väiksematel suurustel.

$E = mc^2$ tähistab seost, et energiast võib saada ainet ja ainest võib saada energiat. Seda teadmist rakendatakse ka osakestekiirendites. LHC CMS prootonikiirendis kandub kineetiline energia (liikumise) mitteelastsel kokkupõrkel (laupkokkupõrge) kahe vastassuunast tuleva prootoni kvarkide interaktsiooni tulemusel tekkinud uutele osakestele, mille mass võib olla suurem kui prootoni masside summa[7] ja mis lagunevad uuesti enne kui neid on võimalik mõõta. Lagunemise tulemusel tekkinud algosade omaduste põhjal (mass, laeng, spinn, liikumise kiirus ja nurk) on võimalik kirjeldada interaktsioone, mis toimusid vahetult peale kokkupõrget ja enne mõõtmist.

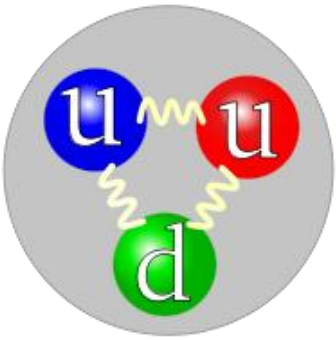
Mida suurem on energia, seda väiksematel distantsidel toimuvaid protsesse on võimalik uurida. $E= hf = hc/ƛ$ (Energia = Plancki konstant * sagedus = Plancki konstant * valguse kiirus / lainepikkus). Mida väiksem on algosakese lainepikkus, seda rohkem energiat ta sisaldab. Mida lühiajalisemalt vaakumist energiat laenata, seda suurema massi võib kokkupõrkel uus algosake omandada.

Ainult algosakesed, millel pole massi, saavad liikuda valguse kiirusel. LHC kiirendis liiguvad prootonid umbes 3 meetrit sekundis aeglasemalt kui valguse kiirus (13 TeV). [19]

Prooton koosneb

- kahest Up-kvargist ja ühest Down-kvargist
- gluuonitest (algosakesed, mis vahendavad jõude)
- ja virtuaalsetest osakestest (määramatus kvantfüüsikas väljendub ka energia fluktuatsioonides, mille käigus energia jäävuse seadust ajutiselt rikutakse - vaakumist (tühjusest) tekkivad algosakesed, mis uuesti kaovad)

[7] Näiteks kahe prootoni kokkupõrkke iteraktsioonis tekkinud Higgsi bosoni mass on umbes 130 suurem kui prootonil.



Joonis: Prooton kahest u-kvargist ja ühest d-kvargist, mis peavad kokku moodustama neutraalse värvi. Algosakestel pole värvi - sinine, punane ja roheline on kokkuleppelised nimetused. [21]

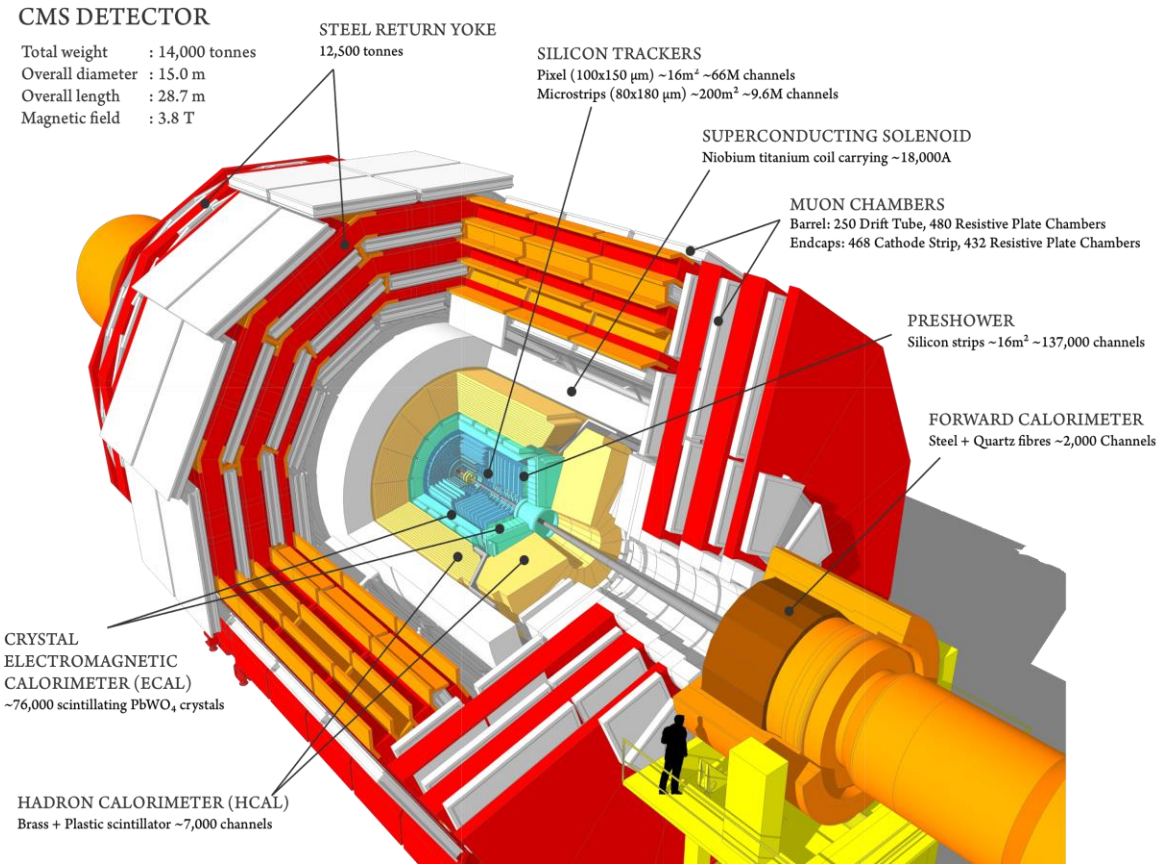
Neutron koosneb kahest d-kvargist ja ühest u-kvargist.

3. Andmed osakestekiirendist

LHC (*Large Hadron Collider*) prootonite kiirendi CMS detektoris (*Compact Muon Solenoid*) toimub sekundis umbes miljard interaktsiooni eri suunas tulnud prootonite kimpude vahel.

Enne salvestamist peavad andmed läbima filtrid (inglise keeles *triggers*). Salvestatud andmed tehakse kättesaadavaks teadlastele üle maailma. [18]

1. Esimese taseme riistvara filter filtreerib välja sündmused, mille energia oli väike. Peale esimese taseme riistvara filtri läbimist jääb alles umbes 100 000 kokkupõrke kohta käivad andmed.
2. Teise taseme filter (umbes 1000 tavaarvutit) konstrueerib 100 000 sündmust detektori sensoritest saadud info põhjal, vaatab osakeste liikumise kiirust ja trajektoore (energia ja laeng) ja jätab salvestamiseks alles umbes 300 sündmust (1 sündmus on 1 MB).



Joonis: CMS (Allikas: CERN CMS projekt)

4. Masinõppe rakendamine

CERN-is kasutatakse tehisintellekti omavahel sarnaste algosakeste eristamiseks.

Esmalt treenitakse tehisintellekti simulatsioonist saadud näidisandmete põhjal (*supervised learning*). Veendutakse tehisintellekti otsustusvõime adekvaatuses (*cross validation*).

Viimaks rakendatakse tehisintellekti katsest saadud andmete, mis on juba esimese (L1) ja teise taseme (L2) filtreid läbinud, uuesti filtreerimiseks konkreetse analüüsi tarbeks (võttes otsustamisel arvesse, kui puhast signaali soovitakse). [22]

4.1. Ülesande täpne kirjeldus

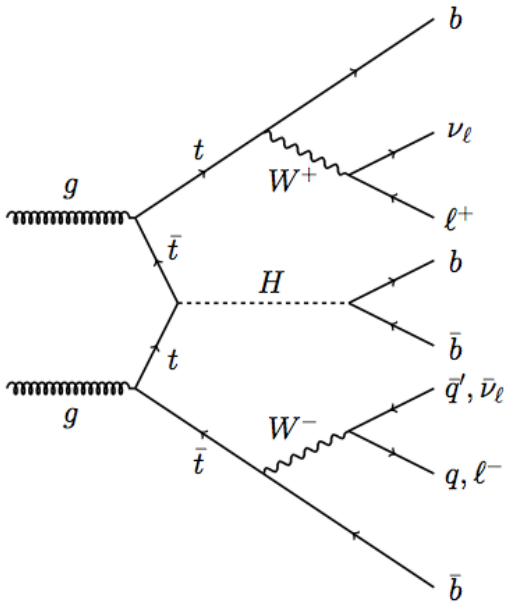
Käesoleva ülesande raames on huvitavaks interaktsiooniks gluuonite ühinemine, sest LHC energia juures interakteeruvad 90% meid huvitavaid protsesse gluuoneid kasutades.

Sellist tüüpi sündmuse, kus toimus gluuonite ühinemine, kahe top ja anti-top paari loomine ja nende lagunemine W-bosoniks ja Higgs-bosoniks mõõtmine toimub CMS sensorites lõppoleku ajal, kus W-bosonid ja Higgs on lagunenud lõppolekuks:

- 4 b-kvarki jetti
- Anti-lepton ja neutriino
- Lepton ja anti-neutriino või kvark ja anti-kvark

Ülesandeks on eristada b-kvark muud tüüpi kvarkidest võimalikult täpselt, sest meid huvitavas interaktsioonis, t-kvarki lagunemisel peab esinema b-kvark.

Sündmused, mida on treeninguks, testimiseks ja võrdluseks kasutatud, on loodud Monte Carlo simulatsiooniga.

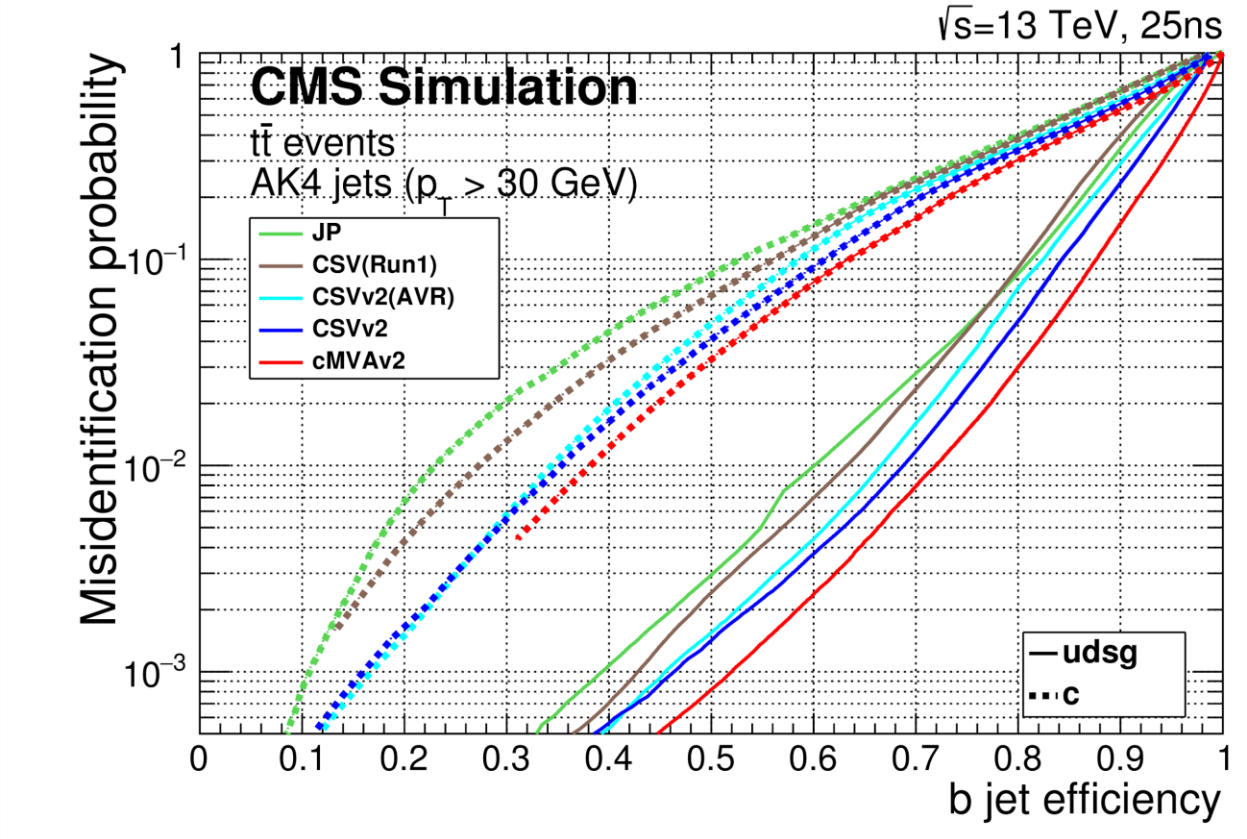


Joonis: Feynman diagramm gluuonite ühinemisest. (Allikas: CERN CMS projekt)

4.1.1. Varasemalt kasutusel olnud klassifikaatorid

Varasemalt on CERNi parima tulemuse saavutanud BDT (boosted decision tree) algoritm (nimega cMVA2).

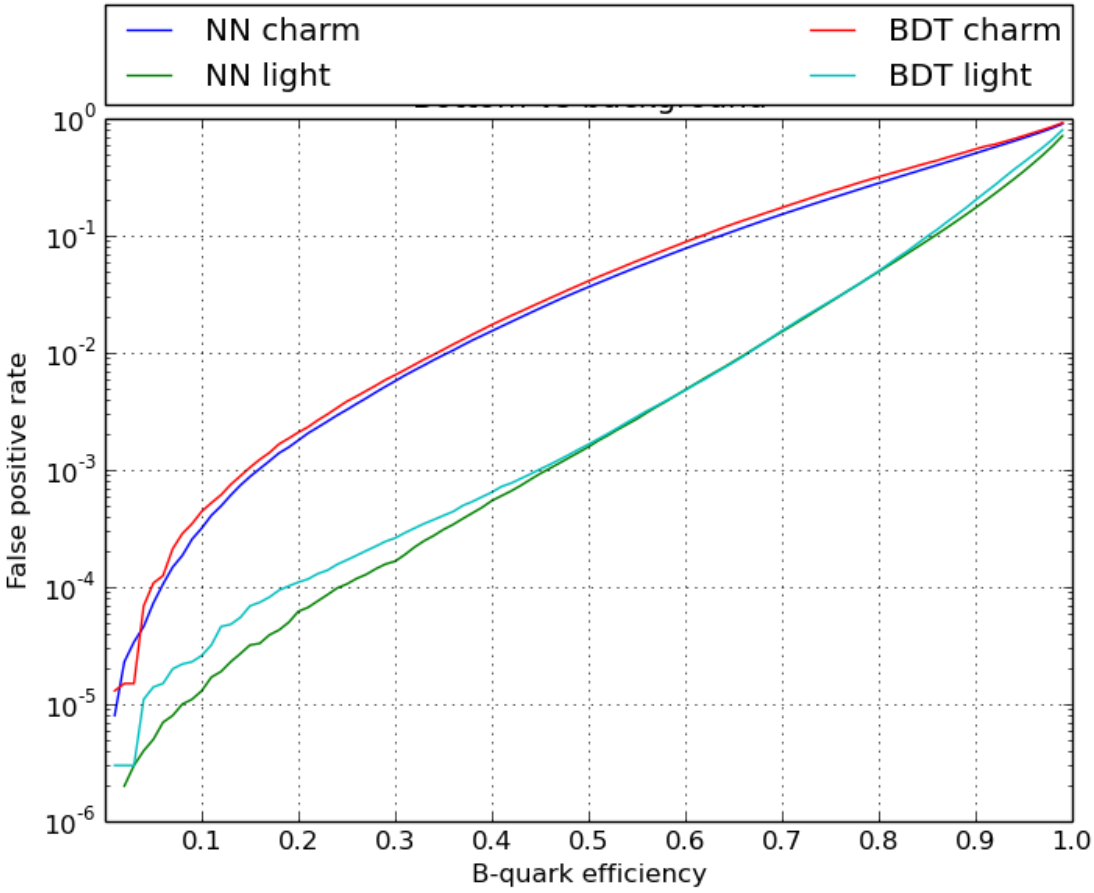
JP, CSV (*Combined Secondary Vertices*), CSVv2, cMVAv2 (*combined MultiVariate Analysis*) on varasemad CERNi algoritmid b-kvarki tuvastuseks, millest cMVAv2 on täpsem iga efektiivsuse juures.



Joonis: B-kvargi tuvastamise tõenäosus (signaal X-teljel) ja tõenäosus, et UDSG- või C-kvark loetakse B-kvargiks (müra Y-teljel). (Allikas: CERN CMS projekt)

4.1.2. Diplomitöö raames valminud närvivõrgu klassifikaatori võrdlus cMVA2-ga igas b-kvarki tuvastamise efektiivsuse piirkonnas

NN on tehisliku närvivõrgu klassifikaator, BDT on cMVA2.



Joonis: CERNi senise parima algoritmi võrdlus närvivõrguga. Võrdlus on koostatud 3 miljoni sündmuse põhjal jaotudes võrdselt B-kvarkide, C-kvarkide ja light-kvarkide vahel. B-kvark (signaal) X-teljel ja et UDSG- või C-kvark loetakse B-kvargiks (müra Y-teljel).

Näide joonise lugemise kohta: Kui lagunemiskanali uurimisel on oluline kõigist b-kvarkidest 70% tuvastada, siis on oluline teada, kui palju c-kvarke loetakse b-kvargiks. Jooniselt tuleb vaadata x-teljel väärtuse 0.7 juures lõikumisi. Jooniselt on näha, et sel juhul on tehisliku närvivõrgu kasutamisel c-kvargi müra 15.15%, otsustuspuu kasutamisel on c-kvargi müra 17.28%.

Tabel kirjeldab, kuidas sõltuvalt sellest, kui suure tõenäosusega b-kvark peab olema tuvastatud ja võrdleb seda, kui palju C-kvarke või kergeid kvarke seetõttu b-kvarkina tuvastatakse.

B-kvargi tuvastamise efektiivsus	NN C-kvargi müra	BDT C-kvargi müra	NN light-kvarkide müra	BDT light-kvarkide müra
0.1	0.00031	0.000419	0.000014	0.000034
0.2	0.001788	0.002039	0.000060	0.00012
0.3	0.005868	0.006496	0.00018	0.000289
0.4	0.015992	0.017776	0.000528	0.000658
0.5	0.03702	0.041414	0.001563	0.001657
0.6	0.077637	0.087768	0.004746	0.004863
0.7	0.151539	0.172835	0.014906	0.015401
0.8	0.280967	0.316567	0.049397	0.049774
0.9	0.507186	0.549237	0.175866	0.202589
0.99	0.899689	0.916134	0.712426	0.792432

Väiksem müra signaali hulgas võimaldab efektiivsemat andmete töötlemist ja täpsemate tulemustega analüüsi.

4.1.3. Erinevate efektiivsuspiirkondade fluksuatsioonid

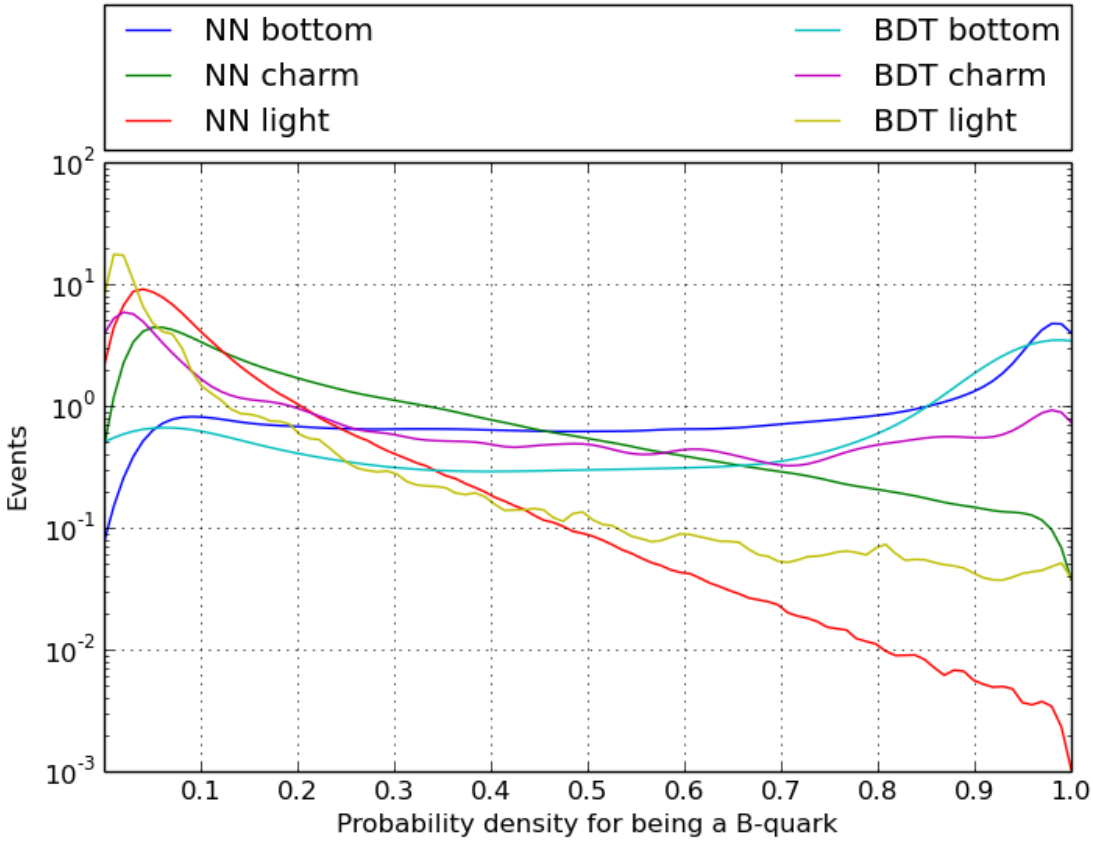
Fluksuatsioonid arvutati võttes testandmetest 10 korda juhuslikult ¼ sündmustest, arvutati müra C-kvargi ja kergete kvarkide kanalitest erinevate B-kvargi tuvastamise efektiivsuste puhul väljavalitud sündmustele ja leiti kõikumine.

Tabel kinnitab, et käsitletud testandmete puhul on tehislik närvivõrk täpsem enamus piirkondades. Võttes näiteks b-kvargi tuvastamise efektiivsuse 0.7, näeme, et 0.151539 ± 0.0013 < 0.172835 ± 0.001762 ja 0.014906 ± 0.000374 < 0.015401 ± 0.00035.

B-kvargi tuvastamise efektiivsus	**NN C-kvargi müra (+-)**	**BDT C-kvargi müra (+-)**	**NN light-kvarkide müra (+-)**	**BDT light-kvarkide müra (+-)**
0.1	0.000048	0.000036	0.000018	0.000014
0.2	0.000116	0.000126	0.000024	0.000028
0.3	0.000222	0.000188	0.000030	0.000038
0.4	0.000186	0.000236	0.000078	0.000064
0.5	0.00055	0.000468	0.000056	0.000108
0.6	0.0008	0.001052	0.00012	0.000172
0.7	0.00136	0.001762	0.000374	0.00035
0.8	0.001672	0.002362	0.000886	0.000734
0.9	0.002166	0.00252	0.002336	0.001744
0.99	0.003194	0.00226	0.006414	0.005446

4.1.4. Tõenäosusjaotus olemaks b-kvark (võrdlus diplomitöö raames valminud närvivõrgu klassifikaatori ja cMVA2 klassifikaatori vahel)

NN on närvivõrgu klassifikaator, BDT on cMVA2.



Joonis: Tõenäosusjaotus olemaks b-kvark.

Näide joonise lugemise kohta: Kui meid huvitab, kui suur hulk c-kvarke on tehisliku närvivõrgu või otsustuspuu hinnangul 0.9 tõenäosusega b-kvargid, siis tuleb vaadata x-teljel väärtuse 0.9 lõikumiskohti. Tehisliku närvivõrgu puhul on 0.2% c-kvarkidest 0.9 tõenäosusega b-kvargid, otsustuspuu puhul on 0.8% c-kvarkidest 0.9 tõenäosusega b-kvargid.

4.1.5. Sisendparameetrid

Visuaalne ülevaade:

- Normaliseeritud sisendparameetrid [Lisa 10.1]
- Sisendparameetrite vahelised seosed [Lisa 10.2]

Tehislikku närvivõrkku treeniti 9 miljoni näitega. Tehislikku närvivõrkku testiti 3 miljoni näitega, mida närvivõrk polnud treenimise ajal näinud. Näited olid jaotatud võrdselt eri tüüpi sündmuste gruppi (33.3% b-kvarke, 33.3% c-kvarke, 33.3% kergeid kvarke). Suur andmestik testimiseks võimaldab väiksema fluktuatsiooniga võrrelda otsustuspuud närvivõrguga.

Et kiirema languse meetod (*gardient descent)* leiaks võimalikult efektiivselt lokaalse miinimumi hüperruumis peab iga näidisandmete sisendparameeter eelnevalt olema:

1. Määratud naturaalselt esinevate väärtuste vahemikku. (Näiteks kui looduses saab mõne parameetri väärtus olla vahemikus [0, 1], siis Monte Carlo andmete genereerimisel ja CERNi poolt kasutatud tarkvara ROOT vormingusse salvestades võivad tekkinud väärtused olla vahemikus [-1000, 1000], kusjuures 910.9% neist jäävad ikka vahemikku [0;1]). Selline parameetri väärtuste vahemike laiendamine tähendab, et algoritm ei ole suuteline konkreetsest parameetrist enam efektiivselt õppima ja tulemused muutuvad ebatäpsemaks.)
2. Normaliseeritud vahemikku (-0.5; 0.5).

Sisendparameetrite täielik nimekiri. Jet_CombMVAETH sisaldab varasemaid klassifitseerimise tulemusi.

- Jet_JP
- Jet_JBP
- Jet_CSV
- Jet_CSVIVF
- Jet_SoftMu
- Jet_SoftEl
- Jet_pt
- Jet_eta
- Jet_phi
- Jet_mass

- TagVarCSV_vertexCategory

Sisendparameetrid iga sündmuse kohta on teineteisest ja ajast sõltumatud.

4.2. Sobiva masinõppealgoritmi valik

Tegemist on mitme sisendmuutuja abil sündmuse tüübi klassifitseerimise probleemiga.

Seda tüüpi probleemi on võimalik lahendada:

- Näidisandmega õppimisega (*supervised learning*) - andmete (sisendparameetrite) põhjal teadaolevatest sündmustest, mille kategooria on teada, luuakse mudel, mis võimaldab klassifitseerida sündmusi, mille kategooria polnud varasemalt teada (eeldusel, et andmetest leiti seos, mis aitaks seda teha).
- Andmete klasterdamine hüperruumis (*unsupervised learning*) - teadmata sündmuse kategooriat või nende hulka paigutatakse andmed (sisendparameetrid) hüperruumi ja eraldatakse kategooriateks võttes arvesse sündmuste paigutust ja tihedust hüperruumis.

Käesolevas töös keskendutakse ainult osale masinõppe koolkondadest, sest testitud tarkvararaamistikud evisid nende tuge:

- Tehislike närvivõrkude kasutamisele (*artificial neural networks*)
- Otsustuspuude kasutamisele (*decision trees*)

4.3. Sobiva tarkvara valik

Kuna nullist masinõppetarkvara kirjutamine on suur arendus- ja halduskulu, on mõistlik uurida, millist avatud lähtekoodiga tarvara on võimalik kasutusele võtta. Sobiva tarkvara valikul on olulisteks teguriteks:

- skaleeritavus, ressursikasutus, kiirus ja sobivus olemasoleva riistvaraga (CPU vs GPU)
- tarkvara küpsus ja elujõulisus
- algoritmide tugi
- elegantsus (arhitektuur, süntaks, silumine)

4.4. Näidisandmed

Andmed (treenimisele, valideerimisele ja testimisele) on loodud Monte Carlo meetodil baseeruval simulatsiooniga.

4.5. TensorFlow

4.5.1. Ülevaade

TensorFlow on avatud lähtekoodiga Google sisemises kasutuses olnud masinõppe platvormi teine versioon, mille põhifunktsionaalsus on kirjeldatud C++-s (efektiivseks ressursikasutuseks) ja mida on võimalik kasutada läbi Pythoni liidestuse.

Käesoleva töö raames on kasutatud Pythoni liidestust, sama töö taaskasutamine C++ versioonis on lihtne.

Algoritmid kirjedatakse TensorFlow poolt määratud süntaksiga, mille põhjal ehitatakse graaf, kus mööda suunatud servi liiguvad tensorid (vabalt valitud dimensioonide arvuga maatriksid, andmed) ja sõlmedes toimuvad operatsioonid (*kernels*) andmetega. Graaf ja selle algosad arvutatakse ühe või mitme serveri piires sobivatel seadmetel (protsessor/graafikaart). [10]

TensorFlow poolt loodud abstraheerimise kiht graafi ehitamiseks võib tunduda natukene kohmakas ja loob järgnevad väärtused:

1. ülevaatlikus TensorBoardis
2. parem ressursikasutus (tegevused ja andmed kirjeldatakse Pythonis, aga arvutused tehakse madalamal tasemel - C++ API)

4.5.2. TensorBoard

Logides muutujate väärtusi programmi koodi seest on võimalik TensorBoardi abil mugavalt saada visuaalne ülevaade algoritmi sees toimuvast, kui rakenduse koodis logitakse neid väärtuseid. Diplomitöö autor lõi praktika käigus mugavama liidese KBFIBoard, mis võimaldab läbi graafilise liidese eri närvivõrgukonfiguratsioone luua, treenida, testida ja tulemusi analüüsida.

4.5.3. Logistiline regressioon

TensorFlow toetab logistilist regressiooni, mille näide asub failis. Kuna logilistiline regressioon ei võimalda mittelineaarsete korrelatsioonide avastamist oli parim saavutatud tulemus 3% kehvem kui parim närvivõrgu tulemus.

4.5.4. Tehislik närvivõrk

Sobiva võrgutopoloogia valik

Närvivõrk ühe sisend- ja väljundkihiga suudab tuvastada lineaarsed seosed. Kui sellisele närvivõrgule lisada juurde üks peidetud kiht, milles on mittelineaarne aktivatsioonifunktsioon, mis peab närvivõrgu korrigeerimise tarbeks (*back propagation*) olema differentseeruv, on võimalik avastada ka mittelineaarseid seoseid. Võrgu kihtide arvu suurendamine võimaldab eri kihtides erinevate seoste leidmise, näiteks näotuvastusel leitakse esimeses kihis leitakse korduvad sarnased kujundid (nina, kõrvad, silmad) ja järgmises kihis seostatakse nad näo kui tervikuga (positsioon ruumis suhteliselt teineteisega).

Et täpselt teada, milline hulk peidetud kihte ja närve igas kihis sobib antud probleemi kõige paremaks lahendamiseks kirjutas diplomitöö autor algoritmi, mis testis läbi eri konfiguratsioonid peidetud kihtide ja närvide arvuga. Närvivõrgu tulemuste tabelis on võetud kõige täpsemad tulemused. Parameetriruumi vähendamine võimaldaks kiiremini ja vähema arvutusvõimsusega klassifitseerida sündmusi, aga vähendaks täpsust - samuti ei annaks see suurt võitu, sest sisendist edasiminevate esimese tasandi servade kaaludega toimub automaatselt olulisuste seoste leidmine (st sisendparameetrid, mis ei ole olulised, neid ei võeta arvesse).

Võis teha tähelepaneku, et kihtide ja närvide arvu suurendada täpsus suurenes mingi piirini ja siis hakkas uuesti langema. Seda võib põhjendada sellega, et närvivõrgu treenimiseks vajaminevate näidete arv suureneb, kui servade arv suureneb, mille kaale treenimise käigus peab sättima ja õppimise sammu suurus jääb samaks.

Vältimaks olukorda, kus võrk on üle treenitud või liiga sõltuv konkreetsetest seostest prooviti treenimise käigus pooli juhuslikult valitud servi eri kihtide vahel (*drop out*).

Võrgu klassifitseerimise täpsust testiti andmetega, mida treenimisel või valideerimisel ei kasutatud. Saamaks ülevaadet, kas võrk on ületreenitud võrreldi klassifitseerimise tulemusi test andmete tulemusega ja treeningul kasutud andmete tulemusega.

Parameetrid, mida närvivõrgu testimisel kombineeriti:

- Peidetud kihtide arv (1-10)
- Närvide arv kihis (10-1200)

- Aktivatsioonifunktsioon (relu, softmax)
- Esialgne õpikiirus ja õpikiiruse vähenemise samm
- Näidete arv: 9 000 000
- Närvide vaheliste servade mittekasutamine (*dropout*)

Üldine tendents on, et näidete arvu suurendamisel, peidetud kihtide ja närvide arvu suurendamisel tulemus paranes. 100 000 näite puhul oli täpsus 61.2%, 3 miljoni näite puhul oli täpsus 64.2%, 9 miljoni näite puhul oli täpsus 64.6%.

4.5.6. Hinnang TensorFlow kasutatuvusele

Autori hinnangul on Google poolt loodud TensorFlow sobiv tarkvara lahendamaks probleeme, mille lahendamiseks sobib tehislik närvivõrk.

Jõudlus

Hetkel näib pudelikaelaks olevat see, kuidas andmeid TensorFlow Pythoni ja C++ vahel liigutatakse, aga jätkuarenduses on lubatud seda optimeerida. Kui ressursikasutus peaks osutuma probleemiks, siis on võimalik leida optimaalne närvivõrgu konfiguratsioon kasutades TensorFlow raamistiku Pythoni liidestust ja hiljem see C++ liidestuses kasutusele võtta.

Jätkusuutlikus

Töö käigus esines mitmeid probleeme ja küsimusi seoses TensorFlow tarkvaraga, millele aitas kiiresti lahendus leida Google otsingumootor või Githubi keskkonnas otse arendajate poole pöördumine. Githubi andmetel on masinõpperaamistikest TensorFlow-l kõige rohkem jälgijaid (*stars*) ja arendajaid (*forks*).

Algoritmide tugi

TensorFlow-l on palju valmis komponente ja näiteid, mis võimaldavad kiiresti tehisliku närvivõrgu mudeli tööle panna.

4.6. KBFIBoard

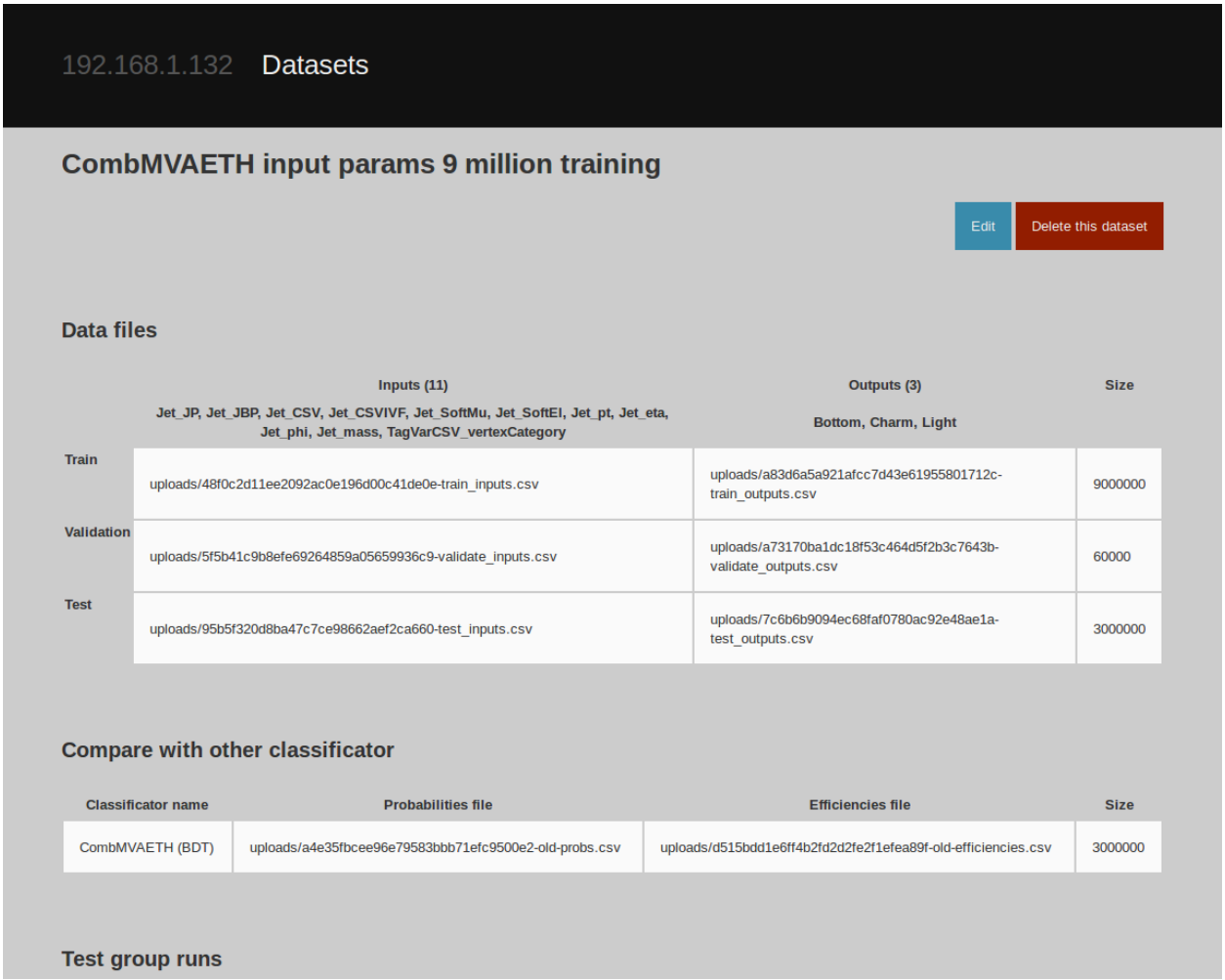
Kuna TensorBoardi ja TensorFlow poolt pakutud funktsionaalsus ei võimaldanud soovitud mugavust treeningute algatamiseks, jälgimiseks ja analüüsiks, siis valmis KBFIBoard, mis on täienduseks nendele tööriistadele. Diplomitöö autor loodab, et pikemas perspektiivis kasvab KBFIBoardist välja erinevaid masinõppe raamistike ja koolkondi võrdlev standariseeritud lahendus.

Tabel toob välja olulised erinevused TensorBoardi ja KBFIBoardi vahel.

	TensorBoard	**KBFIBoard**
Funktsionaalsus	TensorBoard võimaldab retrospektiivis toimunust ülevaadet saada (nt kuidas entroopia väheneb kui biaseid ja kaale korrigeeritakse).	KBFIBoard võimaldab algatada uute konfiguratsioonidega treeninguid, testida olemasolevaid ja näitab reaalajas treeningus oleva võrgu olekut.
Andmete säilitamine	TensorFlow vorming, mis eeldab, et iga treeningu logid salvestatakse eraldi kausta.	Andmed ladustatakse MongoDB-s, mis võimaldab erinevaid konfiguratsioone ja nende omadusi (kulunud aeg, täpsus klassifitseerimisel) lihtsamalt võrrelda.
Treenimine ja testimine	-	On võimalik automaatselt ja paralleelselt mitut erinevat konfiguratsiooni luua, testida, salvestada ja taaskasutada.
Logi analüüs	-	Logiteated on seotud süsteemsete objektidega (TrainRequest, TrainResult, TestRequest, TestResult) ja konkreetsete tegevustega, mis võimaldab saada mugavamalt ülevaadet.

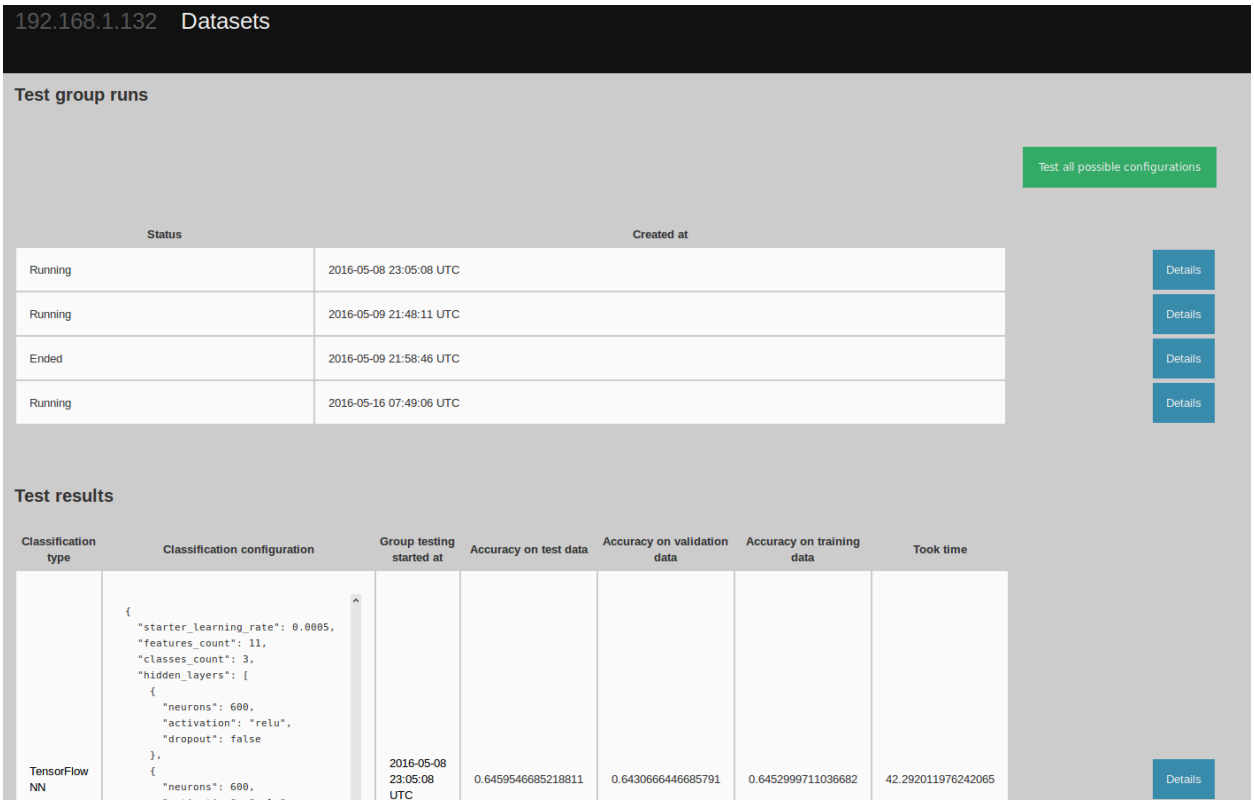
4.6.1. KBFIBoardi ekraanitõmmised

4.6.1.1. Ülevaade konkreetsest andmestikust



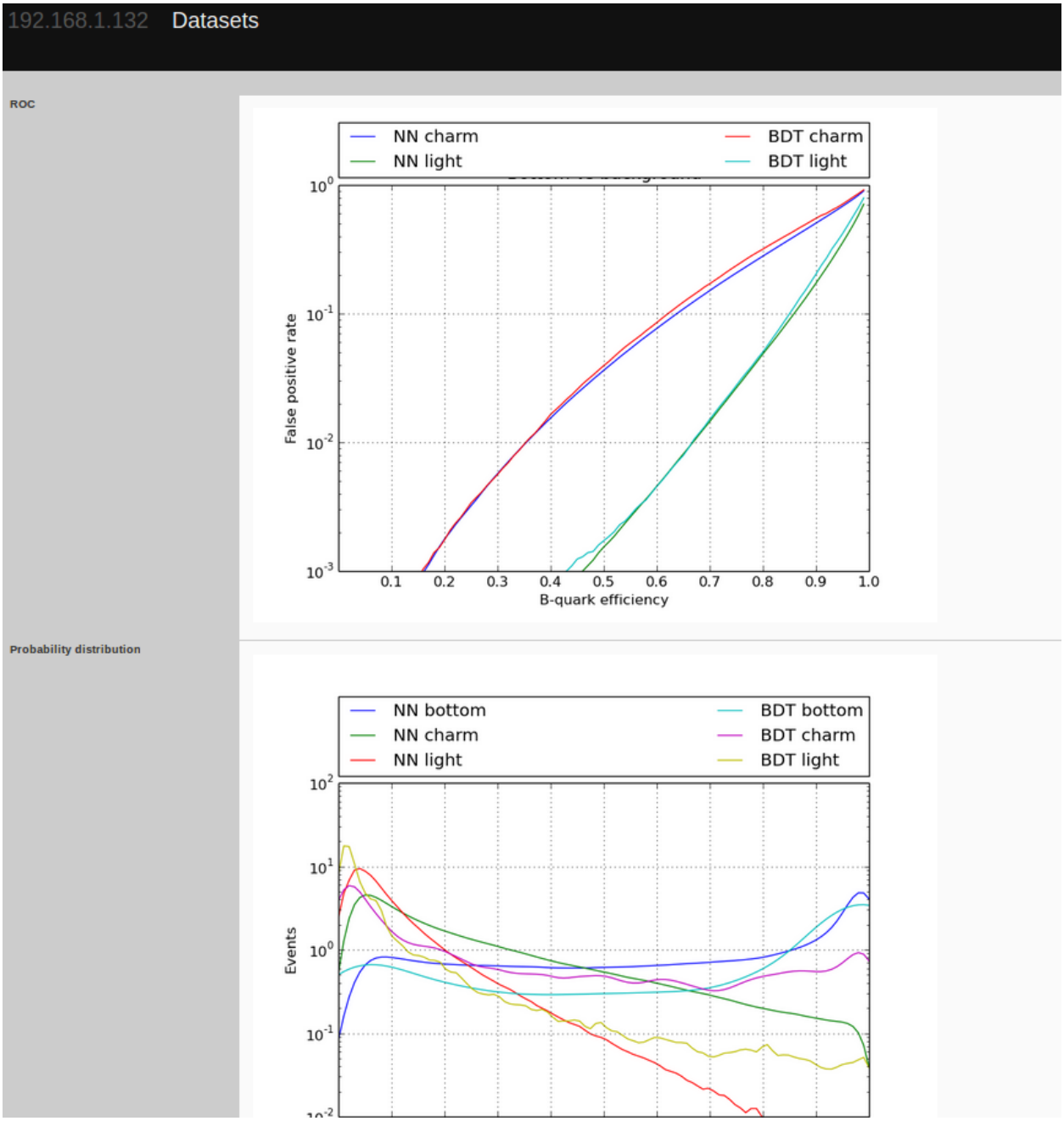
Joonis: Ekraanitõmmisel on näha rakenduse vaade, kus saab ülevaate konkreetsest andmestikust - sisendparameetrid, oodatud väljunud, metaandmed (nimed, hulk); lehe allosas "Compare with other classificator" sektsioonis on vana klassifikaatoriga seotud teave (nimi, tõenäosused ja müra-efektiivsuse suhe samadele testandmetele).

4.6.1.2. Ülevaade andmestikuga tehtud treeningutest ja nende tulemustest



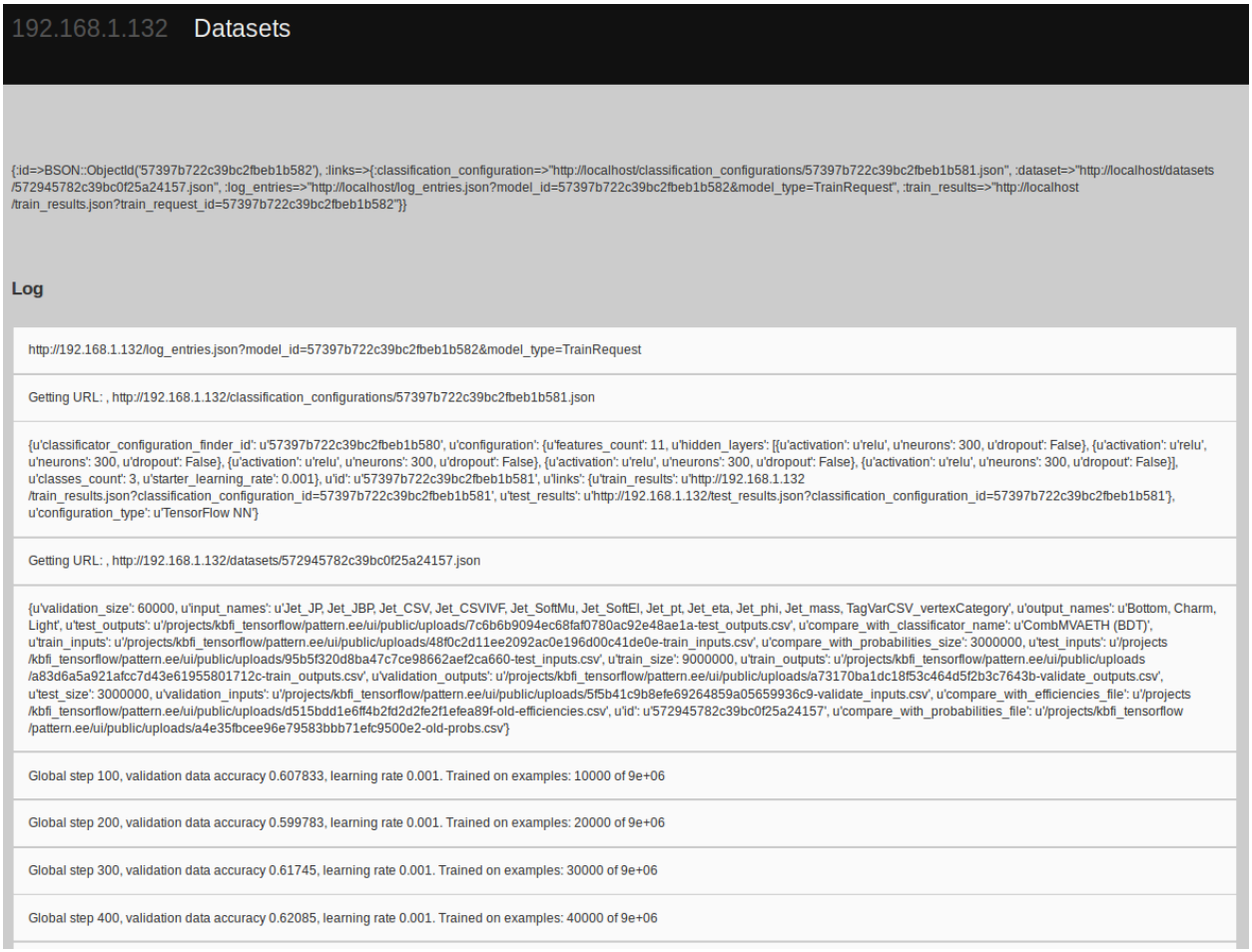
Joonis: “Test all possible configurations” nuppu vajutades testitakse järjest kõik võimalikud võrgukonfiguratsioonid, mille parameetrid on loetletud punktis 4.5.4. Sektsioonis “Test Results” on näha loetelu parimatest testitulemustest.

4.6.2. Osa testitulemuse detailse ülevaate kuvast



Joonis: Ekraanitõmmisel on näha, kuidas üks väljavalitud tehistliku närvivõrgu konfiguratsioon klassifitseerib võrreldes CombMVAETH klassifikaatoriga.

4.6.3. Näide treeningu algatamise päringu logidest



Joonis: Ekraanitõmmis vaatest, kus on võimalik näha treeningualgatamise päringu logi (milliseid andmeid kasutatakse, treeningu hetkeolekut - progressi, õpisammu, täpsust).

4.7. Abstraktsioonikiht TensorFlow-le

Kuna TensorFlow kasutab *placeholder*-eid, mis teevad süntaksi kasutamise võrdlemisi ebamugavaks, valmis ka abstraktsioonikiht, mis võimaldab arendajal puhtamat ja arusaadavamat süntaksit. (Analoogne Keras.io-le, aga paremini hallatav käesolevas kontekstis.)

KBFIBoard on visuaalne haldusvahend sellele abstraktsioonikihile.

Eelis puhta TensorFlow süntaksi ees:

- Puhtam süntaks (umbes 50x vähem ridu koodi kirjutada)
- Võimekus lihtsalt salvestada ja taaslaadida tehisliku närvivõrgu konfiguratsioone

4.8. Ideed tulevikuks

1. **Andmete mahu kasvatamine, mida treeningul kasutada.** Olenemata tehisliku närvivõrgu parameetrite muutmisest (õpikiirus ja -samm, topoloogia, aktivatsioonifunktsioonid, *drop out*), oli näha, et rohkem andmeid treenimisel annab testimisel parema tulemuse. Mis siis, kui kasvatada treeningandmeid 9-lt miljonilt 90-le miljonile?

2. **Sisendparameetrite arvu kasvatamine.** Käesolev töö raames treeniti tehislik närvivõrk näidisandmetega, mis olid loodud simulatsiooniga ja omasid piiratud hulka sisendparameetreid. Mis siis, kui sisendparameetreid on rohkem (sensori tasemel)?

3. **Osakestekiirendi andmete klasterdamine hüperruumis ja nende andmete võrdlus simuleeritud andmete klastritega hüperruumis.** Tänasel päeval pole võimalik Monte Carlo meetodil simuleerida sellises mahus ja kvaliteediga andmeid, et seda moodust oleks võimalik rakendada, sest puudub vajalik arvutusvõimsus.

5. Hajusarvutused lagunemiskanali uurimiseks (kolmanda taseme filter)

Mitme kuu andmete salvestamise tulemusena on kokkupõrgete kohta käiv info on jaotatud umbes 70 megabaidi suurustesse failidesse. Kokku on analüüsitavate failide maht ~10TB olenevalt uuritavast lagunemiskanalist.

Diplomitöö raames optimeeriti, kuidas olemasolevas raamistikus teostati arvutuste jaotamist ja logimist.

Tarkvara, mida kasutatakse andmekaeves, olulised omadused:

1. Arvutuste jaotamist Scientific Linux serverite klastrile toetab SLURM[8].
2. Kasutades Hadoopi on igal kobarserveril ligipääs andmetele.
3. Olenevalt uuritavast lagunemiskanalist rakendatakse igas kobarserveris kolmanda taseme filter mingile andmete alamosale.
4. Filtri läbinud andmed agregeeritakse, veendutakse nende õiguses ja analüüsitakse.
5. Analüüsi tulemusena osatakse hinnata, kas on avastatud uus alamosake ja mis on tema omadused (mass, laen, spinn jne).

Diplomitöö raames tehtud täiendused:

1. Andmefailidega säilitatakse metaandmed, mis võimaldavad kontrollida andmefailide korrektsust.
2. Andmefailide agregeerimine klastris.

[8] https://slurm.schedmd.com/

6. Analüüsi kordamine dokument “CMS AN-15-289” järgi

Dokument “CMS AN-15-289”[9] kirjeldas 2012 aasta andmest leitud liiasust 28.4 GeV piirkonnas. Ülesande eesmärk on korrata analüüsi ja kasutada 2015 ja 2016 aasta andmeid.

6.1. Filtrid

Tabel loetleb, mis peavad olema parameetrite väärtused, et sündmus läbiks filtri.

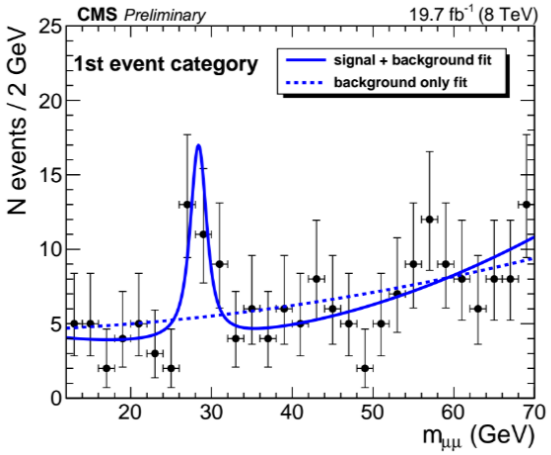
Parameeter	Filter 1	Filter 2
Müüonid	Vastand laenguga, p_T > 25 GeV, |η| < 2.1	Vastand laenguga, p_T > 25 GeV, |η| < 2.1
b-jet	p_T > 30 GeV, |η| < 2.4	p_T > 30 GeV, |η| < 2.4
kvargi-jet	p_T > 30 GeV, |η| > 2.4	p_T > 30 GeV, |η| < 2.4
veto	Ükski jet ei tohi olla p_T > 30 GeV, |η| < 2.4	Ükski jet ei tohi olla p_T > 30 GeV, 2.4 < |η| < 4.7
Puuduolevat energiat		< 40 GeV
Δφ(μμ, jj)		> 2.5
mμμ	12 GeV < mμμ < 70 GeV	12 GeV < mμμ < 70 GeV

p_T – liikumise energia risti prootonkiirendi kiirega.
η – nurk prootonkiirendi kiirest.
Δφ(μμ, jj) – nurk müüonite ja kvargi jettide vahel.

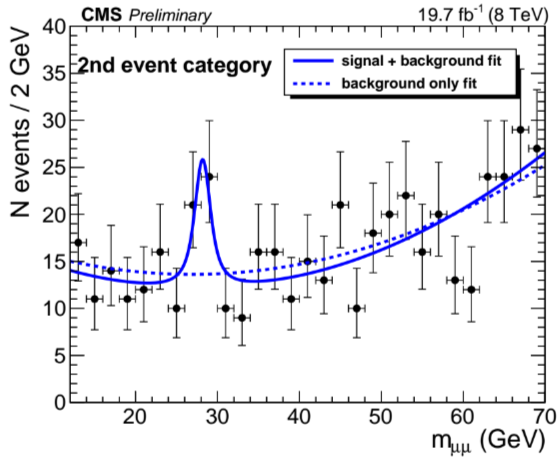
[9] Dokumendile pole viidatud, sest seda on lubatud kasutada ainult CERN-i sisemiselt.

6.2. Filtrid läbinud andmete tulemuste visuaalne esitlus

6.2.1. Aasta 2012 (8 TeV) andmete tulemused



Joonis: Filter 1 läbinud sündmused. Y-teljel on sündmuste hulk, X teljel on kahe müüoni massi resonants.

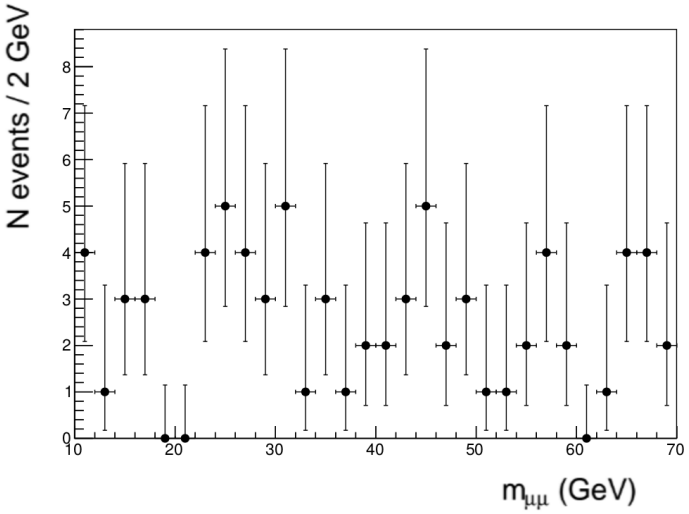


Joonis: Filter 2 läbinud sündmused. Y-teljel on sündmuste hulk, X teljel on kahe müüoni massi resonants.

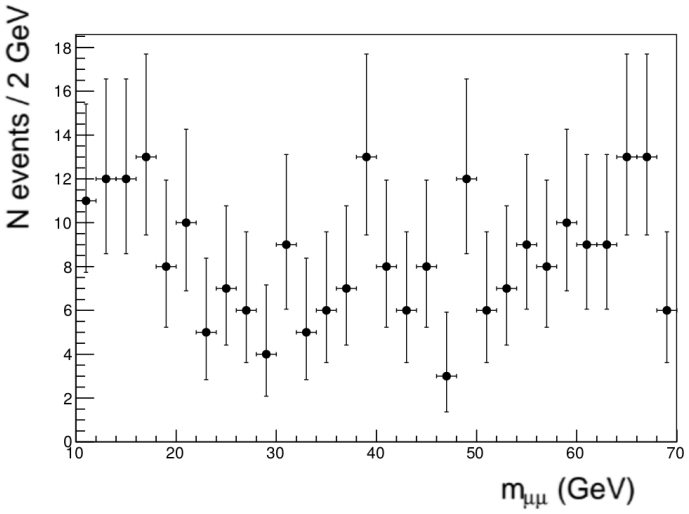
Joonistel on näha, et 28.4 GeV piirkonnas on kahe müüoni massi resonantsiga sündmuste hulk umbes 40% kõrgem kui paremal või vasakul piirkonnas. Kuna joonisel kujutatud sündmuste hulk on väike, siis on fluktuatsioonid suured ja ei saa väga kindlalt väita, et avastatud on uus komposiit- või algosake, mille mass on umbes 28.4 GeV-i.[10]

[10] Katkendlik joon näitab müra ja ei võtta arvesse signaali, katkematu joon näitab signaali ja müra summat; katkematu joone ja katkendliku joone vahe näitab signaali.

6.2.2. Aasta 2015 (13 TeV) andmete tulemused



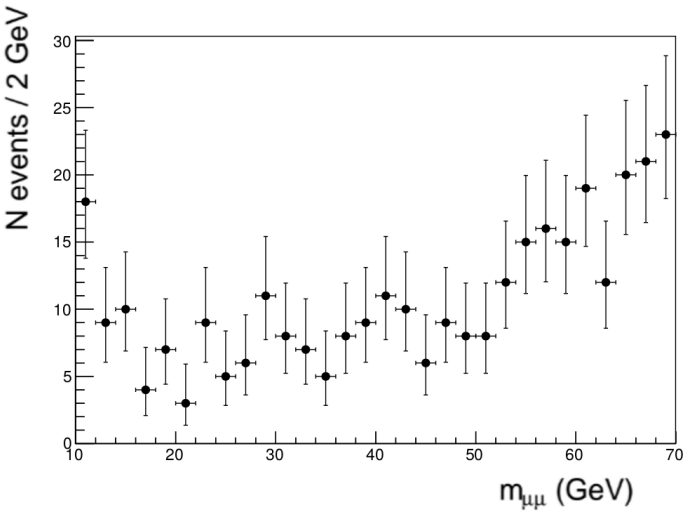
Joonis: Filter 1 läbinud sündmused. Y-teljel on sündmuste hulk, X teljel on kahe müüoni massi resonants (GeV-ides).



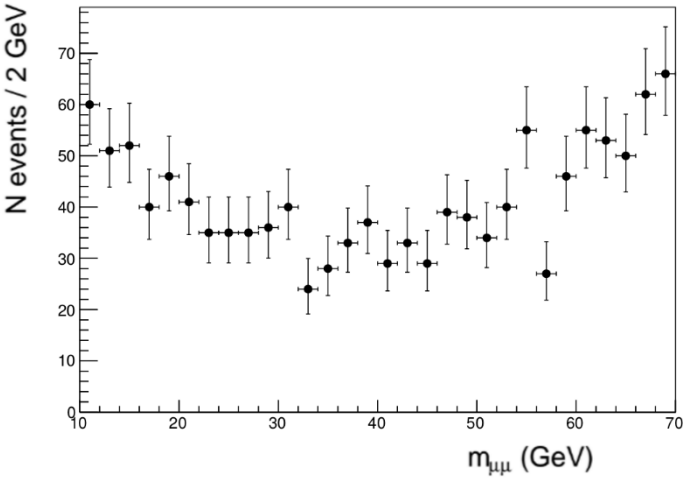
Joonis: Filter 2 läbinud sündmused. Y-teljel on sündmuste hulk, X teljel on kahe müüoni massi resonants (GeV-ides).

Joonistelt on näha, et sündmuste hulk on väiksem kui 2012 andmete puhul ja fluktuatsioonid on suuremad.

6.2.3. Aasta 2016 (14 TeV) andmete tulemused



Joonis: Filter 1 läbinud sündmused. Y-teljel on sündmuste hulk, X teljel on kahe müüoni massi resonants (GeV-ides).



Joonis: Filter 1 läbinud sündmused. Y-teljel on sündmuste hulk, X teljel on kahe müüoni massi resonants (GeV-ides).

Joonistelt on näha, et sündmuste hulk on suurem ja fluktuatsioonid on väiksemad võrreldes 2012 aasta andmete tulemustega ja liiasus puudub 28.4 GeV piirkonnas.

2016 aasta andmete juures on näha, et 28.4 GeV juures pole avastatud uut algosakest või komposiitosakest, mille resonants kuuluks sellesse piirkonda.

6.3. Kasutatud tarkvara

1. CERNi tarkvara TMVA (kvarkide klassifitseerimiseks).
2. CERNi tarkvara ROOT 6 (sündmuste andmete ladustamine ja taasesitamine).
3. KBFI tarkvarakomplekt HEP-KBFI (abstraktsioonikiht, mis automatiseerib andmete töötlemist, vahelüli punkt 1 ja punkt 4 vahel).
4. SLURM tarkvara (sündmuste filtreerimise jaotamine klastrisse).

6.4. Dimplomitöö autori poolt sooritatud tegevused

1. Loodi rakenduse kood olemasolevale raamistikule, mis filtreeris välja meid huvitavad sündmused ja salvestas nende kohta käiva informatsiooni. (https://github.com/HEP-KBFI/2mu1b1j)
2. Väljafiltreeritud andmete visualiseerimine kasutades RooFit-i.

6.5. Olulised viited

2015 ja 2016 andmetele rakendatud filtrite kood: https://github.com/HEP-KBFI/2mu1b1j/blob/master/bin/analyze_2mu1b1j.cc

Graafikute loomise kood:
https://github.com/HEP-KBFI/2mu1b1j/tree/master/macros

Ingliskeelne dokumentatsioon:
https://github.com/HEP-KBFI/2mu1b1j/blob/master/docs/1%20-%20Running%20the%20project.md

7. Kokkuvõte

Tehtud praktika ja diplomitöö tulemusena:

1. Leiti närvivõrgu mudel ja selle konfiguratsiooniparameetrid, mis suudab pakkuda kvaliteetsemat signaali eristust mürast võrreldes seni kasutusel olnud otsustuspuu mudeli konfiguratsiooniga.
2. Aidatati muuta hajusarvutusi toetavat raamistikku efektiivsemaks. Infrastruktuuri (andmekandjad, võrk) piirangutega toimetulek, parem veakindlus (täiendatud logi, automaat-testid).
3. Korrati analüüsi dokument “CMS AN-15-289” järgi, mis kirjeldas 2012 aasta andmest leitud liiasust 28.4 GeV juures. Diplomitöö käigus kasutati võrdluseks 2015 ja 2016 aasta andmeid.

Diplomitöö autor tänab Mariot, Andrest, Christianit, Joosepit, kõiki KBFI-st, CERN-ist ja Eesti Infotehnoloogia Kolledžist, kes leidsid aega, et õpetada, täpsustada ja suunata, et selle tulemuseni jõuda.

8. Conclusion in English

In result of diploma work:

1. Author found a artificial neural network configuration with more accurate result for predicting type of a particle.
2. Author helped to improve framework, what supports sharing computation between cluster nodes.
3. Author repeated “CMS AN-15-289” analysis, that was originally done with CMS 2012 data, with 2015 and 2016 data.

Author is grateful to everybody in National Institute of Chemical Physics and Biophysics, CERN and Estonian Information Technology College who supported this opportunity to gain a better understanding of the process of data mining in particle physics.

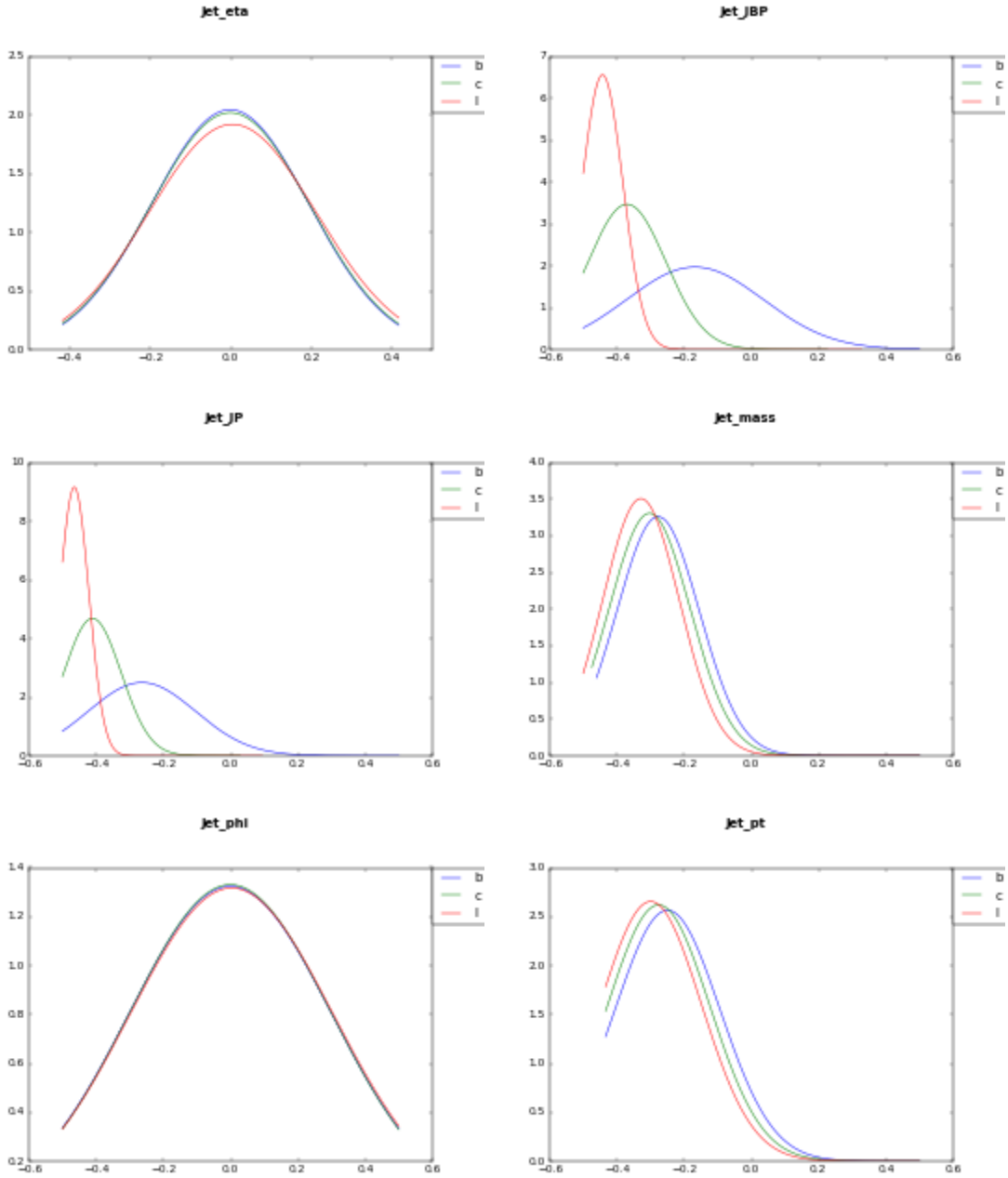
9. Kasutatud materjalid

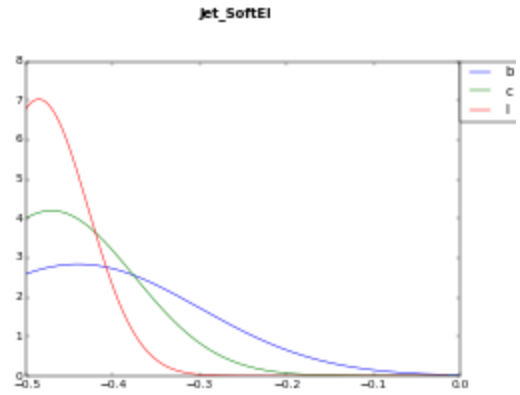
1. The Master Algorithm, Pedro Domingas http://www.amazon.com/The-Master-Algorithm-Ultimate-Learning/dp/0465065708
2. Machine Learning by Stanford University, Andrew Ng https://www.coursera.org/learn/machine-learning/home/welcome
3. Deep learning, Vincet Van Houce https://www.udacity.com/course/progress#!/c-ud730
4. Kuidas liigitada kvargi maitset https://twiki.cern.ch/twiki/bin/view/CMSPublic/SWGuideBTagMCTools
5. https://github.com/tensorflow/skflow
6. https://homes.cs.washington.edu/~pedrod/papers/cacm12.pdf
7. Identification of b-quark jets with the CMS experiment http://arxiv.org/pdf/1211.4462.pdf
8. Neural Networks. R. Rojas. 1996 https://page.mi.fu-berlin.de/rojas/neural/neuron.pdf
9. https://en.wikipedia.org/wiki/Statistical_learning_theory
10. http://download.tensorflow.org/paper/whitepaper2015.pdf
11. http://www.dcs.gla.ac.uk/~vincia/textbook.pdf
12. http://cds.cern.ch/record/2138504?ln=en
13. Identification of b quark jets at the CMS Experiment in the LHC Run 2 https://twiki.cern.ch/twiki/bin/view/CMSPublic/PhysicsResultsBTV
14. http://home.cern/about/computing
15. Efficient BackProp, Yann LeCun http://yann.lecun.com/exdb/publis/pdf/lecun-98b.pdf
16. http://kbfi.ee
17. A Comparison of Prediction Accuracy, Complexity, and Training Time of Thirty-three Old and New Classification Algorithmsm, Tjen Siem Lim http://www.stat.wisc.edu/~loh/treeprogs/guide/mach1317.pdf
18. http://cms.web.cern.ch/news/triggering-and-data-acquisition
19. https://en.wikipedia.org/wiki/Large_Hadron_Collider
20. https://commons.wikimedia.org/wiki/File:Standard_Model_of_Elementary_Particles.svg
21. https://commons.wikimedia.org/wiki/File:Quark_structure_proton.svg
22. https://home.cern/about/experiments/cms

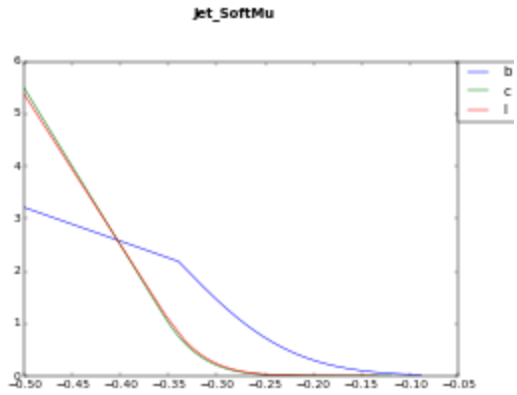
10. Lisad

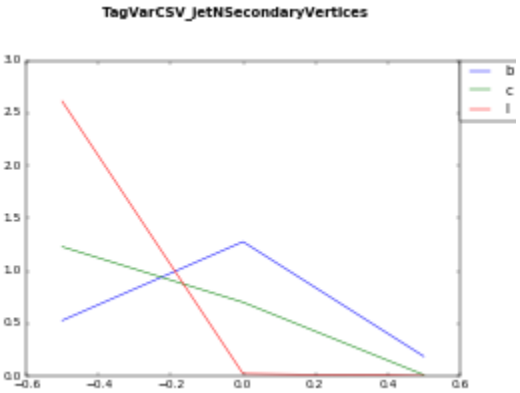
10.1. Lisa 1: Normaliseeritud sisendparameetrite jaotused

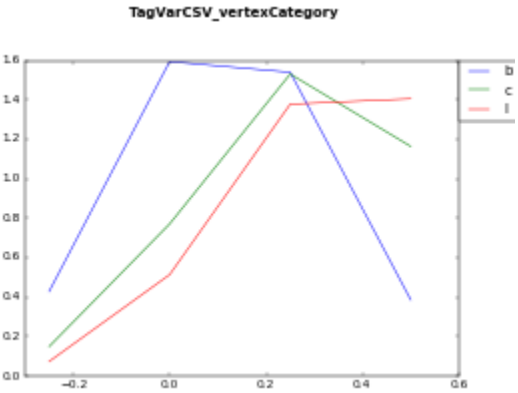
Joonstel on b-kvarkide, c-kvarkide ja kergete kvarkide erinevate parameetrite väärtuste jaotused.




Jet_SoftEl
b
c
l

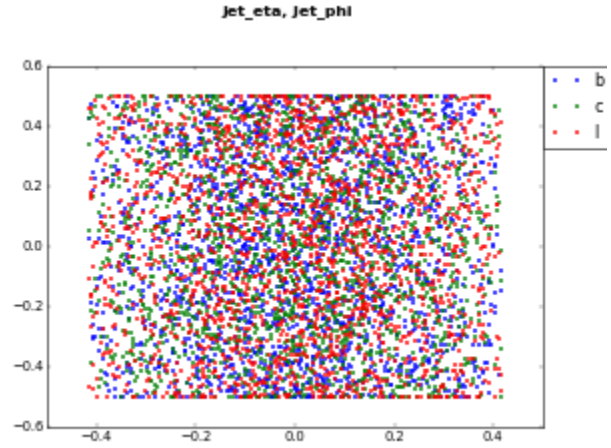

Jet_SoftMu
b
c
l

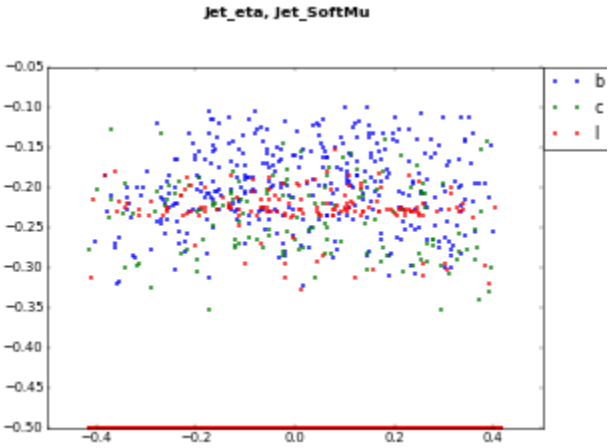

TagVarCSV_jetNSecondaryVertices
b
c
l

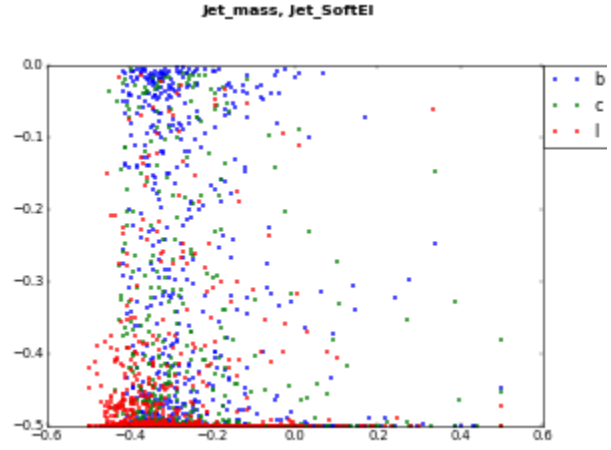

TagVarCSV_vertexCategory
b
c
l

10.2. Lisa 2: Sisendparameetrite vaheliste soeste 2d esitlus

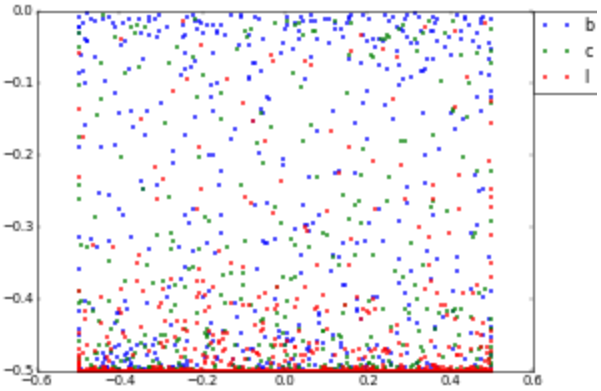
X teljel on esimene pealkirja parameeter, Y teljel teine. Selline 2D esitlus võib peale vaadates aimu anda, kuidas parameetrid on omavahel sõltuvad. Sarnaselt jaotab tehislik närvivõrk parameetrid hüperruumis (ruumis, kus on rohkem kui 3 dimensiooni).



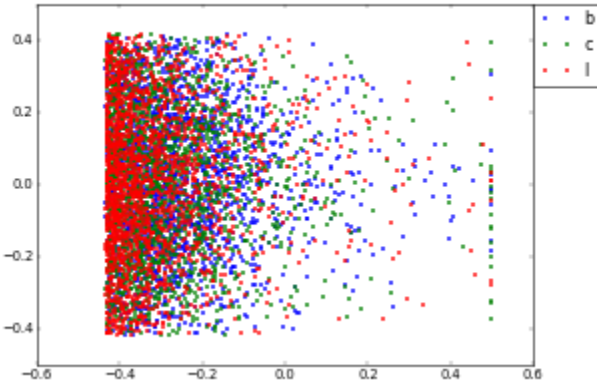




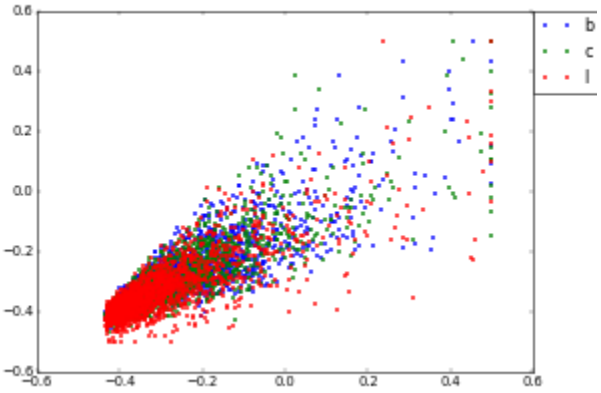
Jet_phi, Jet_SoftEl

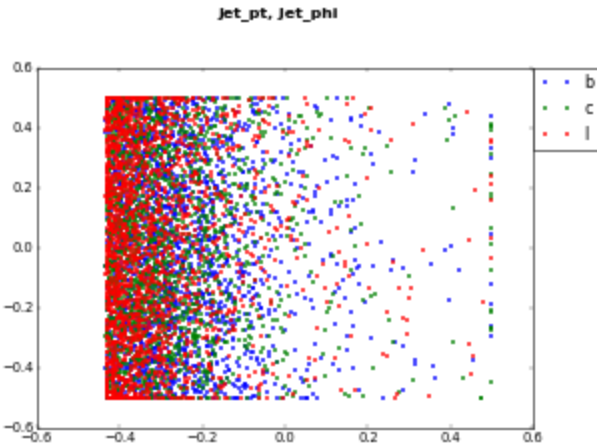


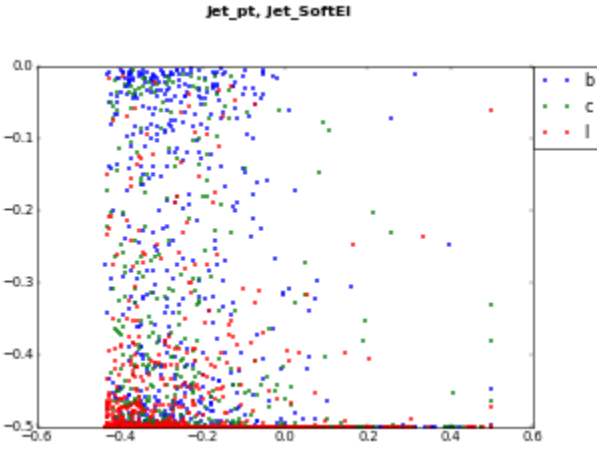
Jet_pt, Jet_eta



Jet_pt, Jet_mass




Jet_pt, Jet_phi
b
c
l


Jet_pt, Jet_SoftEl
b
c
l

10.3. Lisa 3: Masinõppe algoritmide täpsem kirjeldus

10.3.1. Otsustuspuu

Kasvanud välja 1960-ndates, kui psühholoogid üritasid modelleerida, kuidas inimene õpib uusi teadmisi. Sisuliselt puustruktuuriga graaf, kus iga tee juurtipust lõpptipuni moodustab reegli. Üks-ühele seos sümboli ja selle poolt esitatud mõiste vahel. Õppimine on järjestikune, igal sammul luuakse uusi oletusi, et jõuda oodatud tulemuseni. Lihtne aru saada, miks mingi tulemus esines ja parandada. Kasutusel Microsoft Kineticu poolt keha positsiooni tuvastamiseks.

10.3.2. Tehislik närvivõrk

10.3.2.1. Matemaatiline tõestus

1957 aastal tõestas vene matemaatik Kolmogorov, et n-argumendiga pidevad funktsioone **f(x1, x2, .. xn)** on alati võimalik esitleda lõpliku hulga ühe argumendiliste funktsioonidega ja liitmis tehtega **f(x1)**. Näiteks: **xy = exp(ln x + ln y).** [8]

10.3.2.2. Närvivõrgu üldine kirjeldus

Lihtsustatud mudel inimaju töötamise põhimõtetest. Mitu-mitmele seos närvide ja selle poolt esitatud mõiste vahel. Õppimine on paralleelne, kõik närvid õpivad samaaegselt. Võrreldes oodatud väljundit reaalse väljundiga, parandatakse sünapsite kaale, et jõuda lokaalse optimumini hüperruumis.

Sisuliselt on tegemist graafiga, kus on erinevad sõlmede grupid jaotatud kihtideks, mida tavaliselt kujutatakse vasakult paremale või ülevalt alla. Kõige esimene kiht on sisendandmed, viimane väljund, vahepealseid nimetatakse peidetud kihtideks. Kaaludega servad ühendavad sõlmi eri kihtide vahel. Sõlm peidetud kihis on (mittelineaarne) aktiveeringu funktsioon, mis sisendkaalude väärtuste põhjal (nt nende summa on suurem kui mingi väärtus, *threshold*) otsustab, kas anda signaal edasi mööda neid servi, mis ühendab teda järgmise kihiga. Sisendkaal moodustub signal * weight + bias.

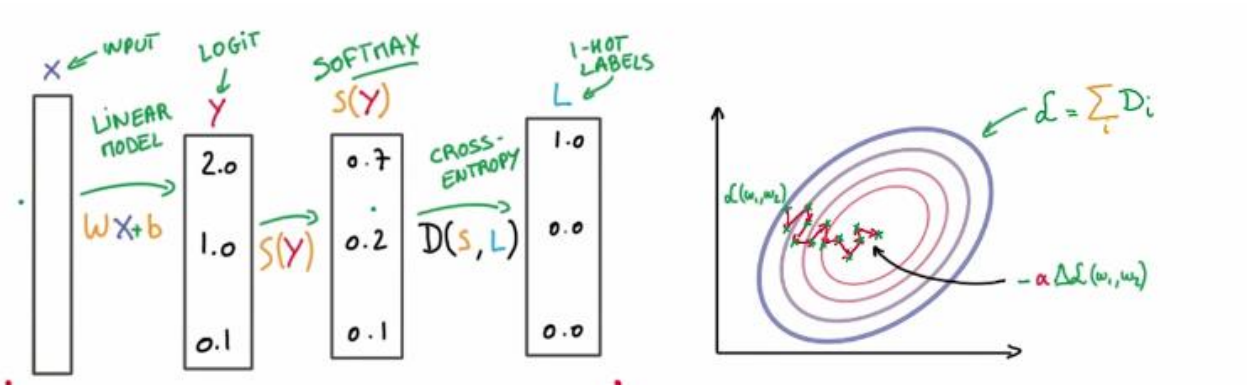
Seostatavates võrkudes kasutatakse funktsiooni ligikaudset määramist.

Olulised terminid:

- *Cost function, loss, (stochastic) gardient descent*
- *Forward propagation*
- *Back propagation*
- Õppimise suurus (*Learning rate*) - kui suuri samme astutakse lokaalse optimumi suunas igal iteratsioonil.
- Servade mittekasutamine (*Dropout*) - vältimaks overfittimist, vältimaks ühest seostest liigsest sõltuvust (*fault redundacy*) ja leidmaks globaalset optimumi, kasutatakse igal iteratsioonil juhuslikult ainult pooli servi kahe kihi vahel.

10.3.2.3. Logistiline regressioon

Toimimispõhimõte



Joonis: Entroopia miinimumi leidmine. [3]

Kategoriseerib lineaarse funktsiooni abil.

WX + b = Y, kus W on number, X on sisendparameetrid, b on number ja Y on tõenäosus kuhugi klassi kuulumiseks. W-d ja b-d “treenitakse” näiteandmete abil.

Treenimiseks kasutatakse teadaolevaid õiged tulemusi, võrreldakse arvutatud tulemusega, leitakse vahe argumendi muudu, tulemuse muudu, arvutustulemuse ja oodatud tulemuse vahel ja väikeste sammudega (sõltuvalt *learning rate* väärtusest) abil parandatakse W-d ja b-d.

Kui gradienti laskumise (*gardient descent*) puhul tehakse elnevalt loetletud sammud kogu andmete peal, siis stohhastiline gradienti laskumine kasutab igal iteratsioonil väikest juhuslikku hulka näiteid (1-100) võimaldades vähem arvutusi ja kiiremat tulemus. Vähesemate näidete puhul

pole tee lokaalsesse optimumi sirgjooneline, et seda teed otsemaks teha, jäetakse varasem üldine inertsi suund meelde (*momentum*). Mõistlik on ka õppimise sammu suurust tehistliku närvivõrgu treenimise käigus vähendada, sest see võimaldab täpsema tulemuseni jõuda.

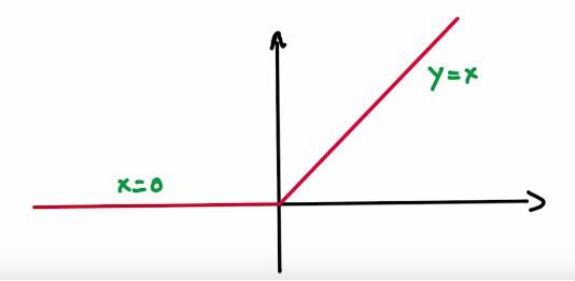
ADAGRAD on SGD täiendus, mis sisemiselt teeb esialgsete juhuslike kaalude ja biaste määramine, õpisammu vähendamist (*learning rate decay*) ja inertsi suuna arvessevõtmist.

Lineaarse klassifitseerimise piirangud

- Nõuab palju sisemisi parameetreid (ja mälu). (sisendparameetrite arv + 1) * kategooriate arv.
- Suudab avastada ainult lineaarseid seoseid (nt $x_1 + x_2$, aga mitte $x_1 * x_2$)

Mittelineaarne klassifitseerimine

Tänu aktivatsioonifunktsiooni sõlmedes kasutamisele, mille tulemus on 0, kui sisendväärtus ei ületa piiri (*threshold*) on närvivõrk võimeline avastamine mittelineaarseid seoseid.



Joonis: Näide mittelineaarsest aktivatsioonifunktsioonist. [3]

Joonis: Näide tehisliku närvivõrgu komponentidest (sisendkiht, aktivatsioonifunktsioon ja väljunud). [3]