Iskanje nove fizike z variacijskimi avtoenkoderji

Praktikum strojnega učenja v fiziki

Domen Vaupotič

1 Nova fizika

Standardni model predstavlja eno najuspešnejših in najboljših fizikalnih teorij, ki opisujejo fundamentalni ustroj sveta. Model vsebuje 12 delcev in pripadajočih 12 antidelcev, med katerimi posredujejo tri različne vrste sil (elektromagnetna, šibka in močna), za popoln opis pa potrebujemo še 26 parametrov, ki podajajo mase delcev in sklopitvene konstante. Čeprav je standardni model samokonsistenten in ga potrjuje obilica eksperimentalnih dokazov, obstajajo pojavi, ki jih ne more pojasniti. Mednje sodijo med drugim neničelna masa nevtrinov, obstoj temne snovi in gravitacijska sila. Ker obstaja kopica različnih razširitev standardnega modela, ni enostavno začrtati eksperimentov, s katerimi bi izbrali pravo smer raziskovanja. Najstarejša in najrobustnejša metoda je razmeroma trivialna in se imenuje iskanje izboklin (*bump hunt*). Z njo preučujemo spektre invariantnih mas, tako da v gladkih območjih ozadja najdemo resonančne vrhove, ki jih povzročajo novi, še neznani delci. Pomembna lastnost te metode je, da je povsem agnostična glede potencialnega modela, s katerim želimo naknadno razložiti obstoj delca. Napredek na področju strojnega učenja nam tako omogoča še boljše, hitrejše in učinkovitejše iskanje resonančnih vrhov v obilju eksperimentalnih podatkov.

2 Variacijski avtoenkoderji

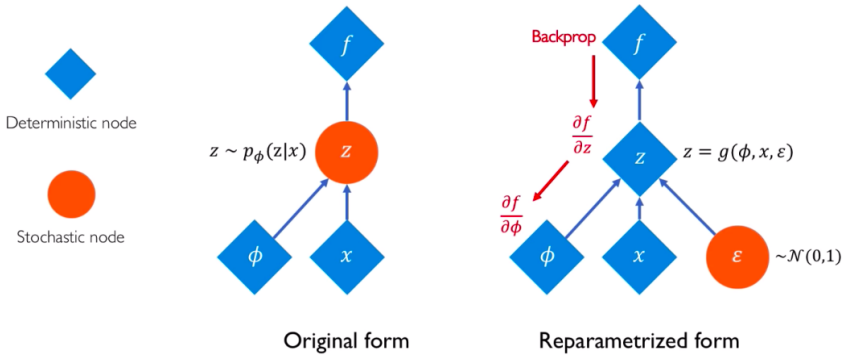
Metoda, s katero se bomo ukvarjali v nalogi, se imenuje variacijski avtoenkoder. Pomudimo se za trenutek pri navadnih avtoenkoderjih. Običajno so sestavljeni iz dveh nevronskih mrež (poljubnega tipa), tako da prva mreža, imenovana enkoder, preslika vhodni podatek v manjdimenzionalni *latentni vektor*, medtem ko druga mreža, imenovana dekoder, latentni vektor preslika nazaj v podatek, ki je čim bolj podoben vhodnemu vektorju. Odvisno od namena uporabe izberemo ustrezno funkcijo izgube, s katero primerjamo vhodni podatek z izhodnim. Avtoenkoderji torej efektivno delujejo kot zgoščevalniki podatkov, saj preslikajo vhodni podatek v zgoščen latentni vektor, ki je manj dimenzionalen. Njihova prednost pred običajnimi zgoščevalniki (npr. JPEG in MPEG) je ta, da se sami naučijo, kako podatke najbolje zgostiti, medtem ko moramo pri klasičnih metodah sami sestaviti algoritem. Njihova sposobnost samoučenja pa se izrazi v slabosti, da so primerni le za uporabo na podatkih, ki so podobni učni množici, medtem ko bodo za bistveno drugačne podatke dajali bistveno slabše rezultate. Avtoenkoderje lahko uporabljamo tudi kot metode za zmanjševanje dimenzij (alternativa PCA, t-SNE) ali za rekonstrukcijo šumnatih podatkov (pri tej rabi avtoenkoder učimo tako, da iz šumnatega vhodnega podatka odstrani šum).

Variacijski avtoenkoderji (VAE) so nadgradnja, pri kateri latentne vektorje zamenjamo z *latentnimi porazdelitvami*. Enkoder tako vhodni podatek preslika v latentno

porazdelitev: običajno izberemo kar Gaussove porazdelitve, ki so torej parametrizirane s srednjo vrednostjo μ in varianco σ. Po enkoderju sledi vzorčenje, pri katerem iz dane porazdelitve žrebamo naključno število, tega pa kot latentno število uporabimo kot vhod dekoderja. Dimenzionalnost latentnih porazdelitev je lahko poljubna: običajno izberemo n-dimenzionalne nekorelirane Gaussove porazdelitve, tako da kot vhod v enkoder nato uporabimo izžrebani latentni vektor. Ključno je, da lahko variacijski avtoenkoder uporabimo ne zgolj kot zgoščevalnik, ampak tudi kot generativni model, ki nam omogoča ustvarjanje novih podatkov.

Funkcija izgube je pri VAE sestavljena iz dveh prispevkov: prvi je enak kot pri običajnih avtoenkoderjih in meri razliko med vhodnim in izhodnim podatkom, drugi prispevek pa je regularizacija, s katero preprečimo, da bi variacijski avtoenkoder latentne porazdelitve razmaknil preveč vsaksebi, s čimer bi zgubili želeno zveznost latentnega prostora. Regularizacijo običajno vpeljemo z Kullback-Leiblerjevo divergenco, s katero latentne porazdelitve silimo k normalizirani Gaussovi porazdelitvi, odstopanje pa ustrezno kaznujemo. Kot že omenjeno, s tem preprečimo, da bi se latentni prostor razgručil v ločene podprostore. Izbira normirane Gaussove porazdelitve kot vnaprejšnje (*prior*) porazdelitve je običajno pogojena s tem, da nam omogoča eksplicitni analitični izračun KL-divergence.

Da lahko VAE naučimo z običajnim vzvratnim razširjanjem, se moramo poslužiti še reparametrizacijskega trika. Z njim vozlišče, ki v VAE predstavlja žrebanje in s svojo naključnostjo predstavlja oviro za vzvratno razširjanje napak, pretvorimo v deterministično vozlišče tako, da žrebanje izvedemo po standardni Gaussovi porazdelitvi, nato pa izžrebano vrednost zgolj ustrezno reskaliramo, da sovpade z ustrezno latentno porazdelitvijo. Reparametrizacijski trik je shematsko prikazan na sliki 1.



Slika 1: Reparametrizacijski trik. [1]

Ker želimo preprečiti, da bi enkoder vračal negativno varianco porazdelitve, običajno izhod enkoderja obravnavamo kot srednjo vrednost, μ, in logaritem variance, $\log \sigma_z$. Reparametrizacijo torej izvedemo tako, da žrebamo ε po standardni Gaussovi porazdelitvi, nato pa z, ki želimo, da je porazdeljen po $\mathcal{N}(\mu, \sigma)$, izračunamo kot:

$$(\mu, \sigma) \text{ in } \varepsilon \sim \mathcal{N}(0,1) \longrightarrow z = \mu + \exp\left(\frac{\varepsilon \log \sigma}{2}\right).$$

Za konec se vrnimo še na izračun izgube. Omenili smo, da je celotna izguba $\mathcal{L}$ sestavljena iz rekonstrukcijske izgube $\mathcal{L}_R$ in izgube zaradi KL-divergence $\mathcal{L}_{KL}$. Izkaže se, da

[1] `https://towardsdatascience.com/reparameterization-trick-126062cfd3c3`

je običajno treba izgubi še primerno utežiti:

$$\mathcal{L} = \zeta\, \mathcal{L}_R + \xi\, \mathcal{L}_{KL}.$$

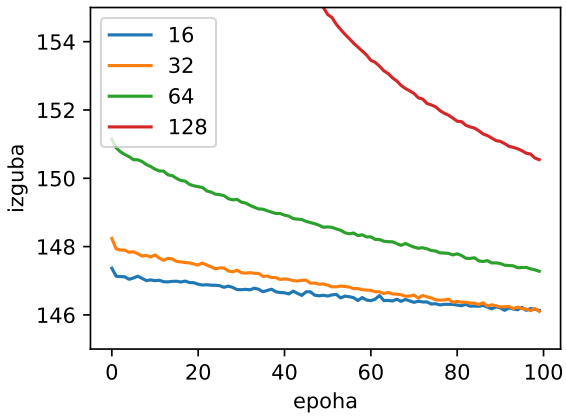
3 Metodologija

Variacijski avtoenkoder smo implementirali v programskem jeziku Python s knjižnicama TensorFlow in Keras.

4 Rezultati

4.1 MNIST

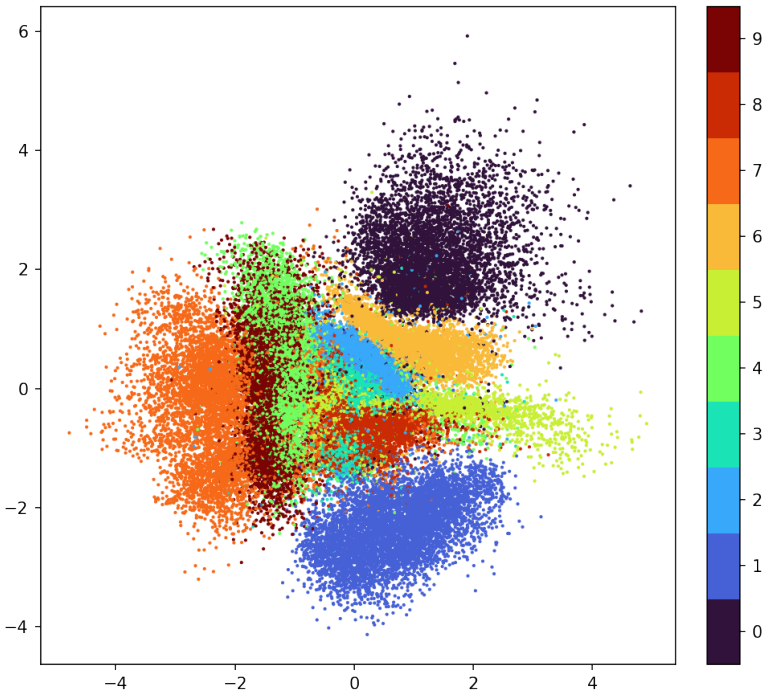
V prvem delu naloge se bomo posvetili podatkovnemu setu MNIST, ki ga sestavljajo ročno zapisane števke 0–9 v obliki slik, velikosti 28×28 pikslov. Vhodne podatke smo normalizirali na vrednosti od 0 do 1 in jih preoblikovali v vektor dolžine 784. Enkoder in dekoder sta bila sestavljena iz ene skrite plasti s 64 nevroni, latentni prostor pa je bil dvodimenzionalen. VAE smo učili z optimizatorjem Adam, in sicer 100 epoh. Na sliki 2 je prikazan potek izgube pri različni velikosti učnega paketa. Opazimo, da z manjšimi učnimi paketi hitreje dosežemo nižje izgube.



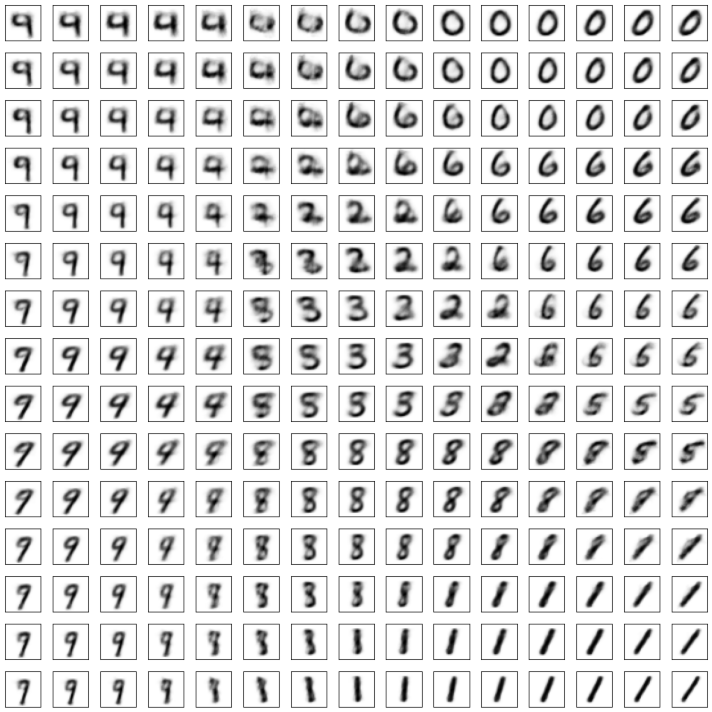
Slika 2: Vpliv velikosti učnega paketa na potek učenja.

Oglejmo si sedaj, kako izgleda latentni prostor, ki je prikazan na sliki 4. Opazimo, da je večina števk ločenih v posamezne gruče, vendar deluje celotni latentni prostor kot povezana, zvezna gruča. Predstavljamo si torej lahko zvezno prehajanje iz ene števke v drugi, kar je vidno tudi na sliki 4, ki prikazuje vzorčen latentni prostor na kvadratu $[-1.8, 1.8]^2$.

Oglejmo si sedaj, kakšen vpliv ima razmerje uteži v celotni izgubi. V zgornjem primeru smo uporabili vrednosti $\zeta = 784, \xi = 1$, kar pomeni, da smo rekonstrukcijsko izgubo utežili z dolžino vhodnega vektorja. Na sliki 5 sta prikazani dve skrajni situaciji. V prvi je uporabljena samo KL-izguba: rezultat je homogena gruča, porazdeljena po dvodimenzionalni standardni Gaussovi porazdelitvi (glej sliko 6), števke pa med seboj niso ločene. V drugem primeru je uporabljena samo rekonstrukcijska izguba: latentni prostor postane veliko bolj nepravilne oblike, števke so dobro ločene, med posameznimi gručami pa so večji prazni prostori, znotraj katerih bi dekoder dajal čudne in nesmiselne rezultate.



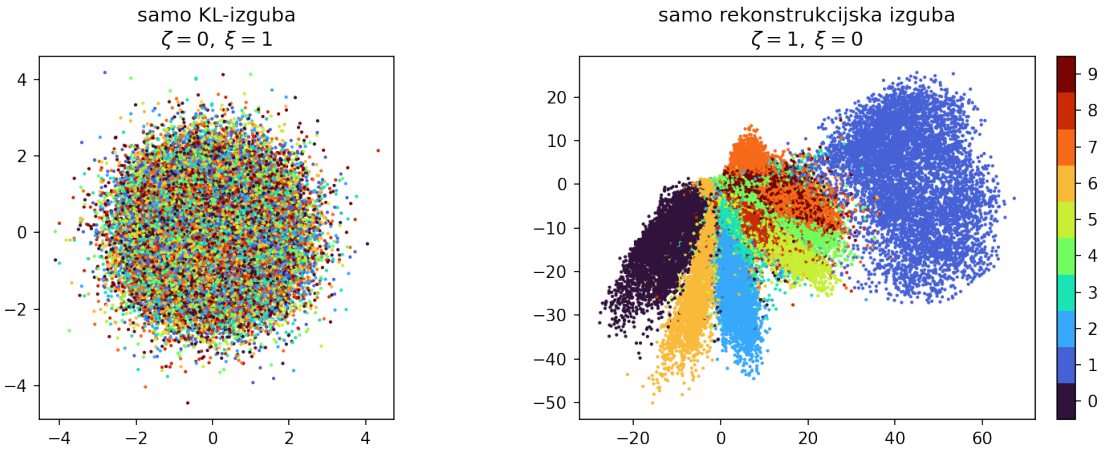
Slika 3: Latentni prostor učnega seta MNIST z utežema $\zeta = 784, \xi = 1$.



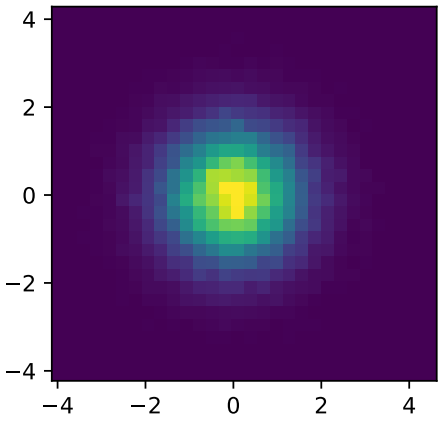
Slika 4: Vzorčen latentni prostor po kvadratu $[-1.8, 1.8]^2$.

4.2 LHCO

V drugem delu naloge se bomo ukvarjali z iskanjem novih delcev v simuliranih dogodkih trkov na LHC. Dogodki bodo sestavljeni iz ozadja (kot ga narekuje kvantna kromodinamika) in razpada namišljenega delca $Z' \to X+Y$. V testnih podatkih so dogodki ustrezno



Slika 5: Latentni prostor v dveh skrajnih primerih: uporabljeni samo KL-izgubi (levo) in uporabljeni samo rekonstrukcijski izgubi (desno).



Slika 6: Histogram latentnega prostora v primeru $\zeta = 0, \xi = 1$.

označeni, delci pa so imeli maso $m_{Z'} = 3{,}5\,\text{TeV}, m_X = 500\,\text{GeV}$ in $m_Y = 100\,\text{GeV}$. Drugi del bo analiza neoznačenih dogodkov (črna škatla), v katerih bomo morali določiti mase vseh treh delcev.

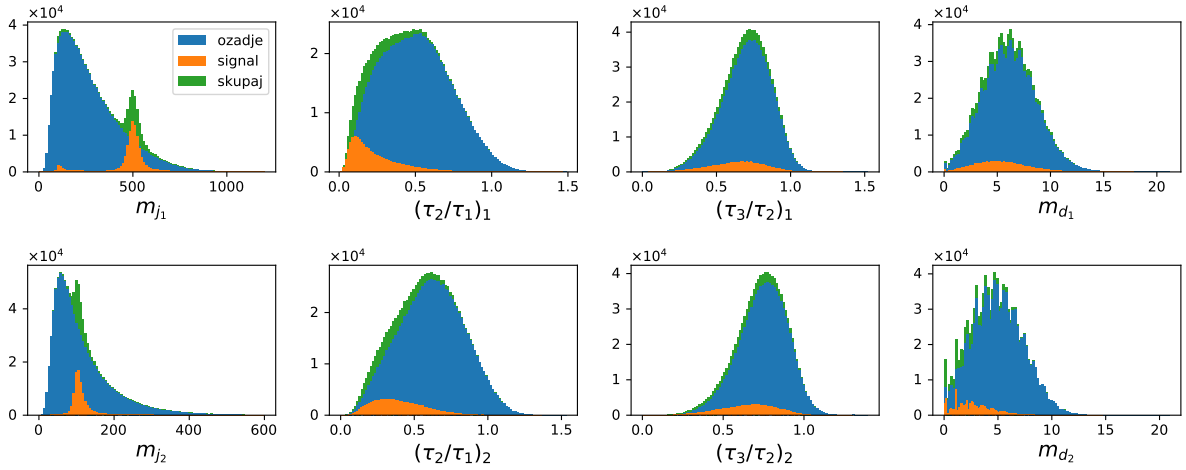
Za vsak dogodek imamo podanih 8 spremenljivk, in sicer po štiri za vsakega izmed hadronskih curkov: invariantna masa curka (m_j), 2-*subjettiness* (τ_2/τ_1), 3-*subjettiness* (τ_3/τ_2) in *massdrop* (m_d). Dodatno imamo podano še invariantno maso celotnega dogodka, m.

Vse spremenljivke najprej transformiramo s transformacijo `QuantileTransformer`, tako da dobimo enakomerno porazdeljene transformirane spremenljivke na enotskem intervalu. VAE je sestavljen iz 3 skritih slojev, vsak ima 64 nevronov, latentni prostor je enodimenzionalen. Rekonstrukcijska izguba bo merjena s povprečnim kvadratom napake (MSE), njena utež pa bo $\zeta = 5000$.

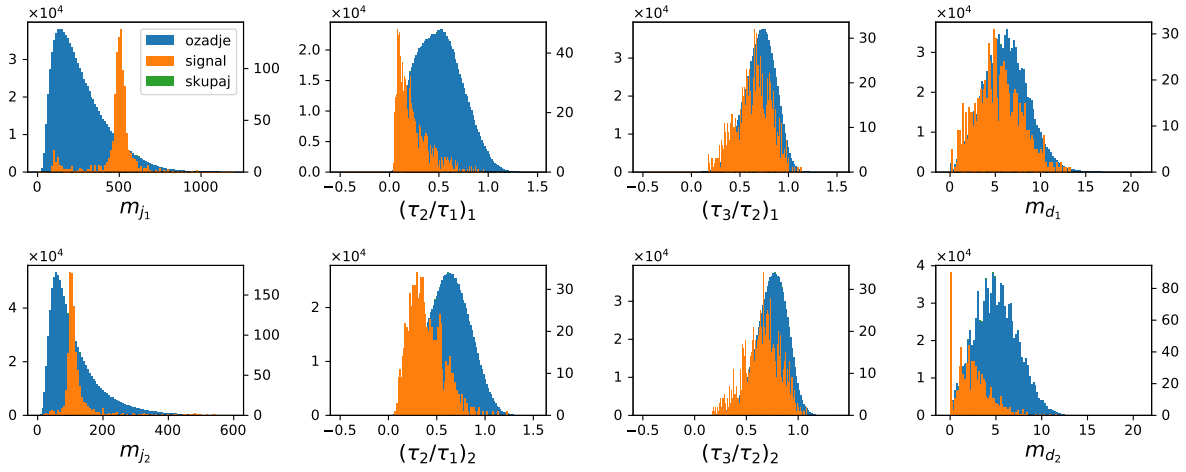
4.3 Testni podatki

Znotraj testnih podatkov imamo dve simulaciji z različnim razmerjem signala in ozadja, S/B. Oglejmo si za začetek strukturo podatkov za obe razmerji signala, kot je prikazano

na slikah 7 in 8. Pri razmerju $S/B = 10\,\%$ je že v neločenih dogodkih mogoče opaziti dva resonančna vrhova pri ustreznih masah delcev X in Y. Nasprotno sta pri razmerju $S/B = 0{,}1\,\%$ vrhova skrita v ozadje, zato sta na grafih prikazana v ločeni y-osi. Na sliki 9 je prikazana še porazdelitev invariantnih mas dogodkov za obe razmerji signala, kjer ponovno opazimo jasen vrh pri razmerju $S/B = 10\,\%$.



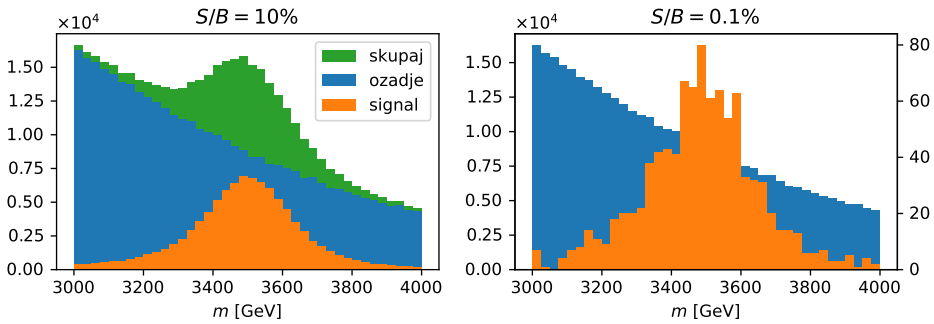
Slika 7: Struktura testnih podatkov pri $S/B = 10\,\%$.



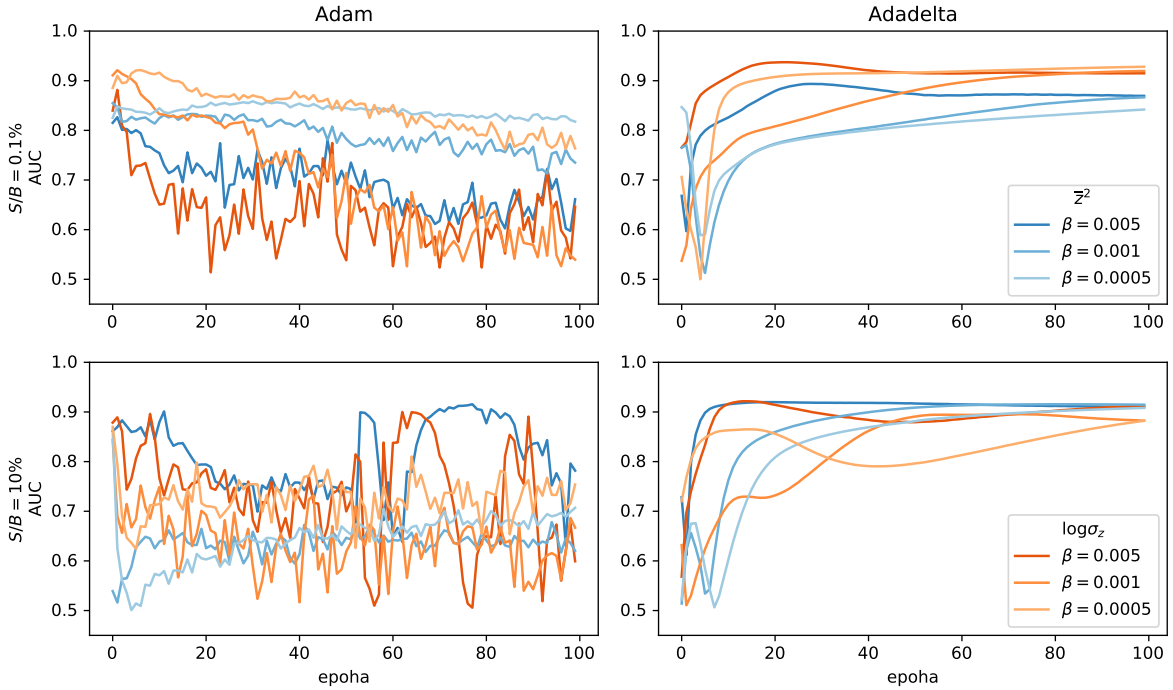
Slika 8: Struktura testnih podatkov pri $S/B = 0{,}1\,\%$.

4.3.1 Vpliv hiperparametrov

Preden se posvetimo učenju, si oglejmo vpliv izbire optimizatorja in hitrosti učenja β, kar je prikazano na sliki 10. Na grafih je prikazana vrednost AUC med potekom učenja za optimizator Adam in Adadelta pri treh različnih hitrostih učenja. Opazimo, da se optimizator Adam obnaša precej nenavadno, zato bomo za vso nadaljnje učenje uporabili optimizator Adadelta.



Slika 9: Porazdelitev invariantnih mas dogodkov na območju med 3000 GeV in 4000 GeV.

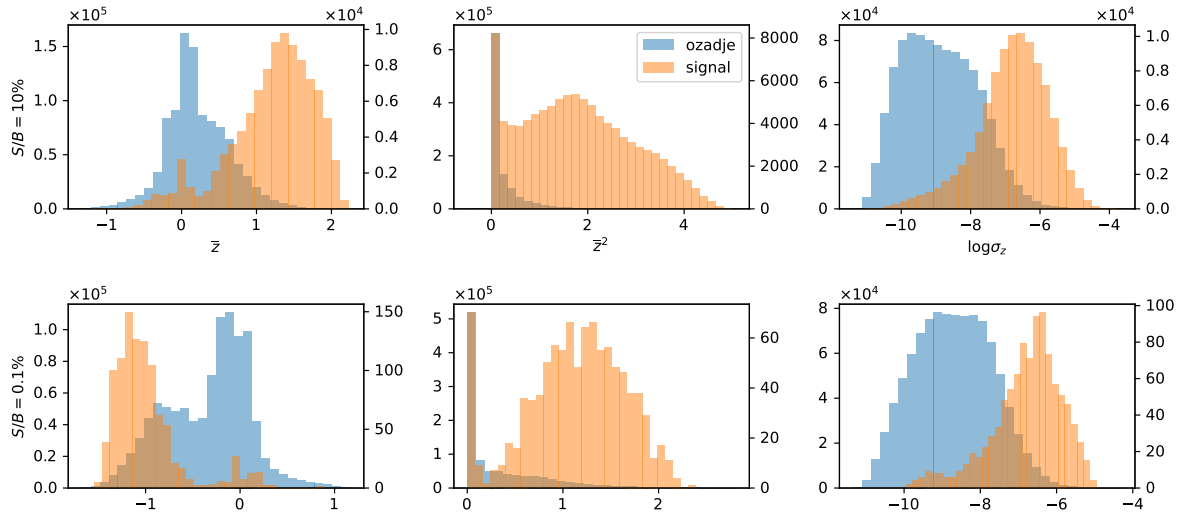


Slika 10: Vpliv hitrosti učenja β in optimizatorja na AUC med učenjem.

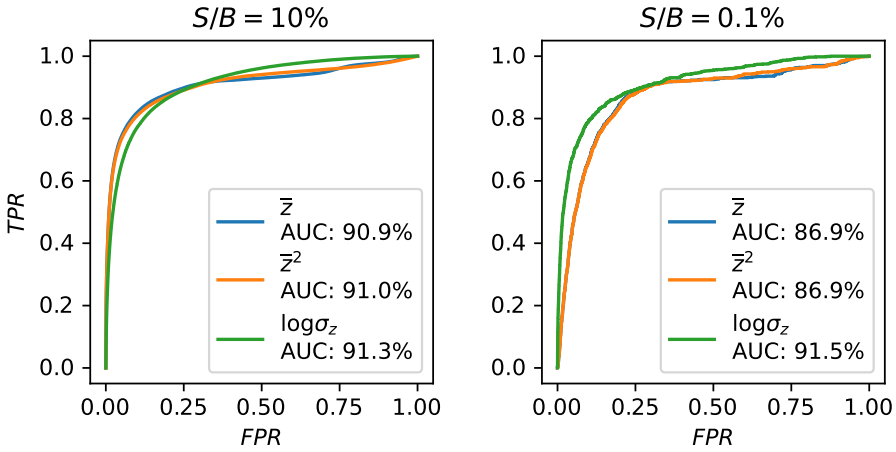
4.3.2 Latentni prostor

VAE smo trenirali 100 epoh in na sliki 11 so prikazane porazdelitve parametrov latentne porazdelitve. Ukvarjamo se torej ponovno z dvema parametroma: srednjo vrednostjo, $\overline{z}$, in logaritmom variance, $\log \sigma_z$. Opazimo, da oba parametra, ki ju bomo uporabili kot klasifikatorja, dobro ločita dogodke ozadja in signala, vredno je omeniti, da se na prikazanih grafih signal večinoma pojavi na desni strani (pri večjih vrednostih klasifikatorja), le v enem primeru se pojavi na levi. Vrstni red je seveda povsem nepomemben in ne vpliva na kakovost klasifikatorja, zavisi zgolj od naključnih začetnih pogojev učenja (inicializacija uteži).

Na sliki 12 so prikazane krivulje ROC za vse tri možne klasifikatorje ($\overline{z}, \overline{z}^2$ in $\log \sigma z$) te pripadajoče ploščine pod krivuljo (AUC). Opazimo, da v obeh primerih klasifikator $\log \sigma_z$ nekoliko prednjači pred ostalima dvema, zato bomo tega uporabili za nadaljnjo klasifikacijo.



Slika 11: Porazdelitev parametrov latentne porazdelitve.



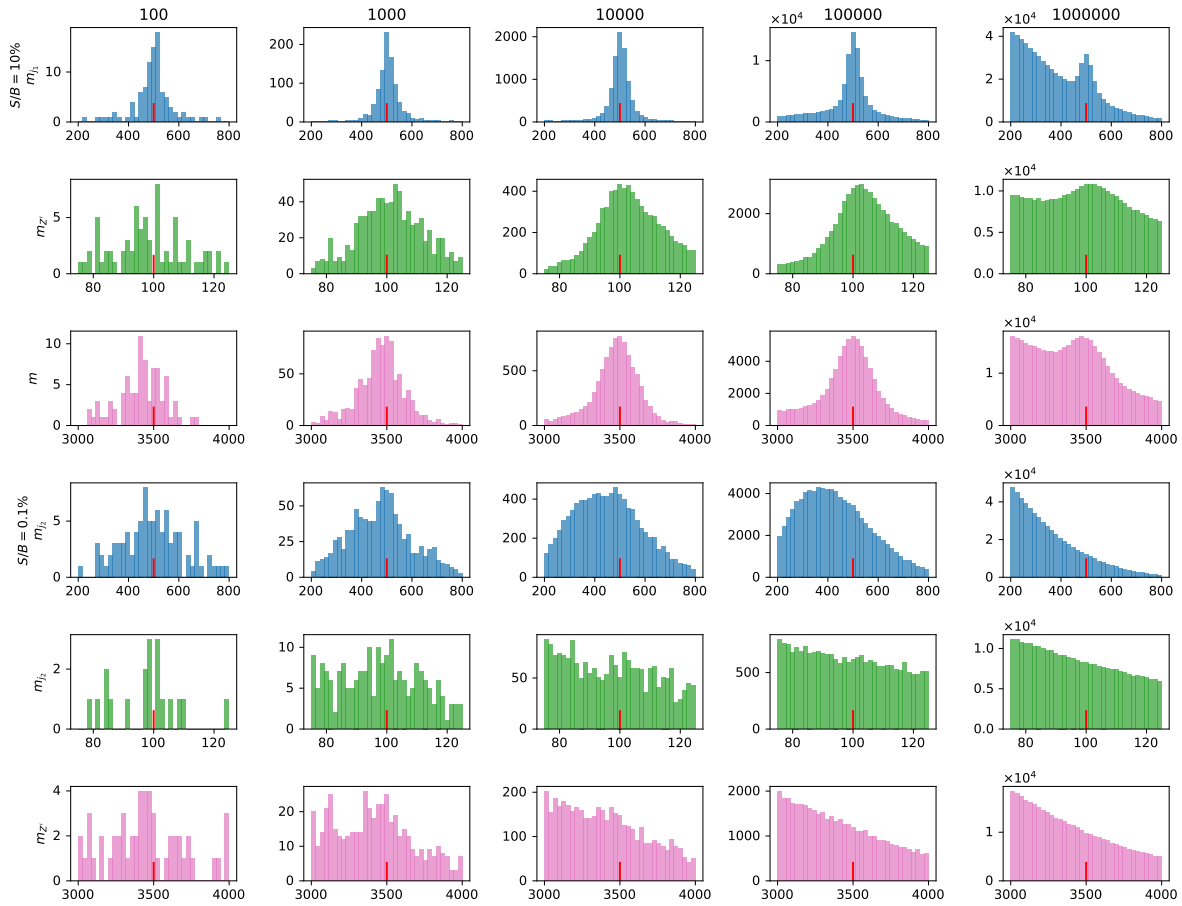
Slika 12: Krivulje ROC za tri različne klasifikatorje in obe razmerji signala.

4.3.3 Porazdelitev parametrov v signalu

Sedaj si bomo ogledali porazdelitev parametrov dogodkov v signalu tako, da bomo pobirali vedno več dogodkov iz ustrezne smeri vrednosti klasifikatorja. Na sliki 13 so prikazane porazdelitve mas za vse tri delce pri naraščajočem številu vključenih dogodkov. Opazimo, da v primeru $S/B = 10\,\%$ klasifikator zelo dobro izlušči resonančne vrhove, ki jih nato začne prekrivati ozadje. Tudi v primeru $S/B = 0{,}1\,\%$ klasifikator najde ustrezne vrhove, vendar ti – pričakovano – hitro zginejo med ozadjem.

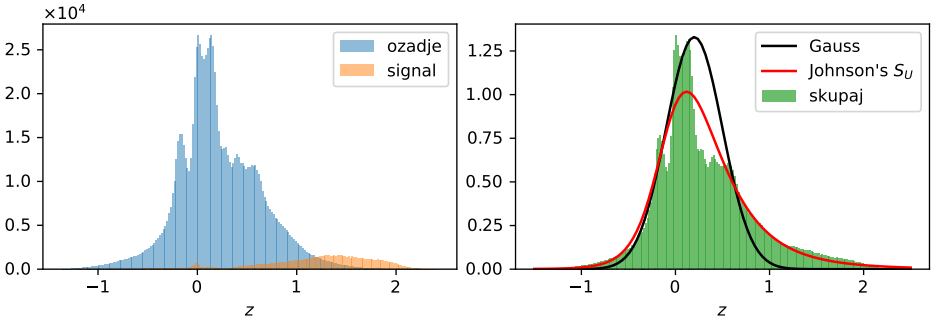
4.3.4 Generiranje novih podatkov

V zadnjem bomo generirali nove podatke in jih primerjali z učnimi podatki, za razmerje signala $S/B = 10\,\%$. Na sliki 14 je prikazana porazdelitev vrednosti latentne spremenljivke z, in sicer ločeno za ozadje in signal (glede na podane oznake). Da bi se čim bolj približali porazdelitvam parametrov v učnem setu, moramo vzorčiti po porazdelitvi, ki bo karseda podobna. V ta namen smo na celotno porazdelitev prilegali Gaussovo porazdelitev, vendar se ta slabo prilega, saj močno podceni predvsem območje s signalom.



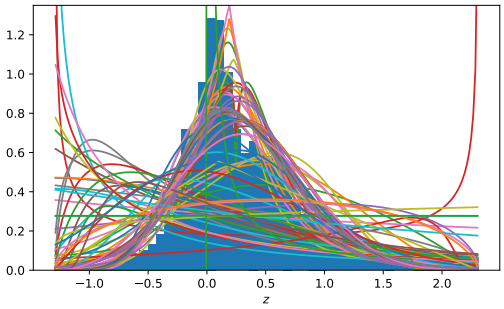
Slika 13: Porazdelitve mas vseh treh delcev s postopnim vključevanjem dogodkov signala. Rdeče črtice označujejo izbrane vrednosti mas, ki so bile uporabljene v simulaciji dogodkov.

Da bi našli boljšo zvezno porazdelitev, smo uporabili iskanje[2] po vseh porazdelitvah iz knjižnice `scipy.stats`, kot je prikazano na sliki 15. Najbolje se je prilegala porazdelitev Johnson S_U.



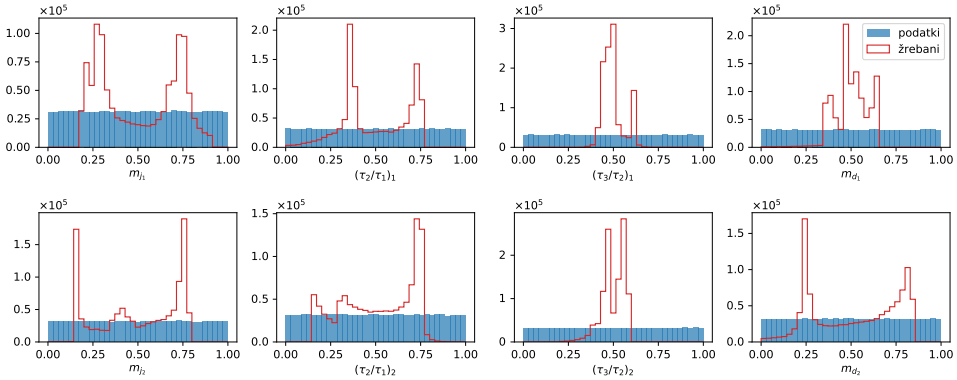
Slika 14: Porazdelitev latentne spremenljivke z za ozadje in signal (levo) ter za vse dogodke skupaj (desno). Prilegani sta Gaussova porazdelitev in porazdelitev Johnson S_U.

[2] `https://stackoverflow.com/a/37616966`



Slika 15: Prilegane raznovrstne zvezne porazdelitve.

Po tej porazdelitvi smo žrebali enako število dogodkov, kot v učnem setu (1100000). Na sliki 16 so prikazane porazdelitve za netransformirane podatke (iz učnega seta in naključno žrebane), na sliki 17 pa ustrezno transformirani. Opazimo, da se porazdelitve v splošnem sicer pojavijo na približno pravih mestih, vendar se po obliki znatno razlikujejo od tistih iz učnega seta. Zanimiva je prisotnost resonančnih vrhov, vendar na napačnih mestih. Najverjetneje je za to krivo prekratko učenje, kar pomeni, da so se tudi po 100 epohah izhodi avtoenkoderja še vedno znatno razlikovali od vhodnih podatkov, kar je vidno tudi na sliki 18, na kateri sta prikazani vhodna in izhodna distribucija. Za boljše ujemanje bi morda bilo treba povečati vrednost uteži ζ in s tem povečati rekonstrukcijsko sposobnost VAE.

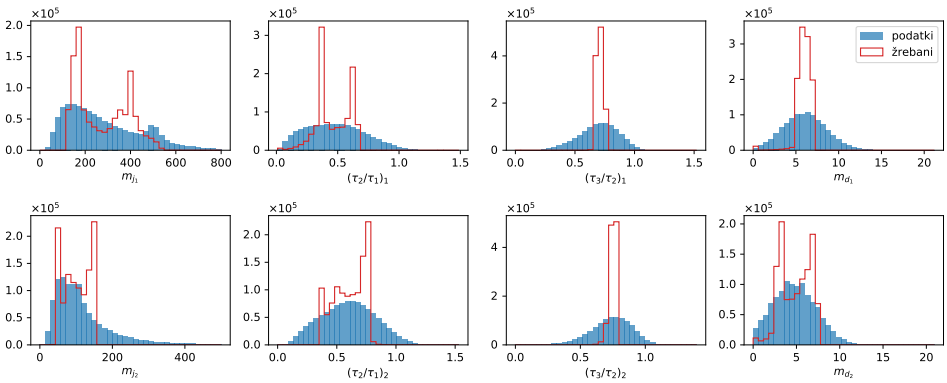


Slika 16: Porazdelitev netransformiranih parametrov za učni set in žrebane dogodke.

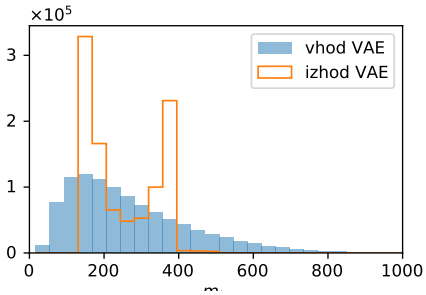
4.4 Črna škatla

V tem delu naloge bomo enak postopek kot na učni množici uporabili še na neoznačenih podatkih. Na sliki 19 so prikazani dogodki za signal, rekonstruirani iz klasifikatorja $\overline{z}$, ki se je v tem primeru bolje odrezal. Opazimo lahko, da se pojavljata resonančna vrhova pri masah približno 400 GeV in 600 GeV. Poleg vrednosti klasifikatorja smo za dogodke signala dodatno uporabili filter, s katerim smo vključili zgolj dogodke s celokupno invariantno maso med 3,6 TeV in 4 TeV.

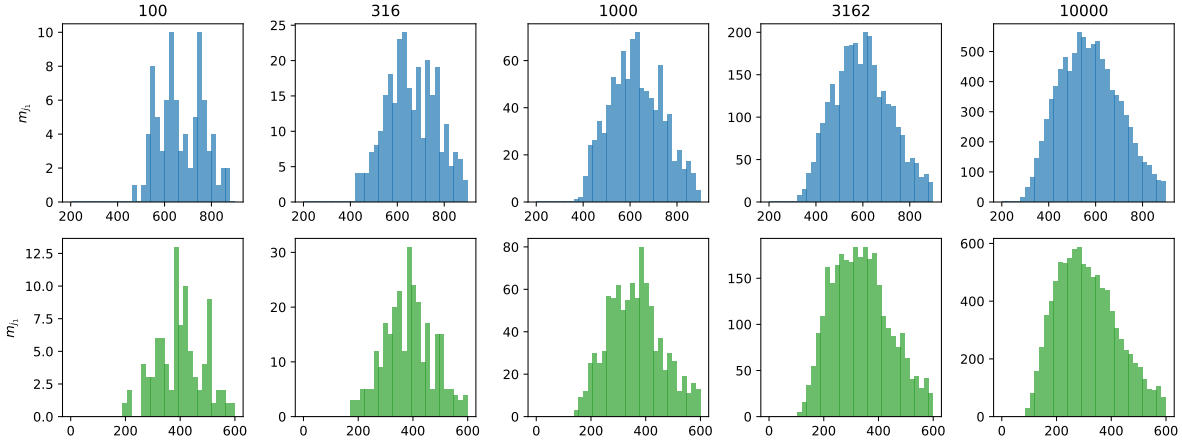
Za boljšo določitev mas smo spreminjali število vključenih dogodkov in spremljali, kje se pojavi vrh histograma. Na sliki 20 je prikazan položaj vrha za vse tri mase pri različnem številu vključenih dogodkov signala. Opazimo, da pri prevelikem številu vključenih dogodkov začnemo zajemati tudi dogodke ozadja, ki prekrijejo signal in s tem se maksimum



Slika 17: Porazdelitev transformiranih parametrov za učni set in žrebane dogodke.



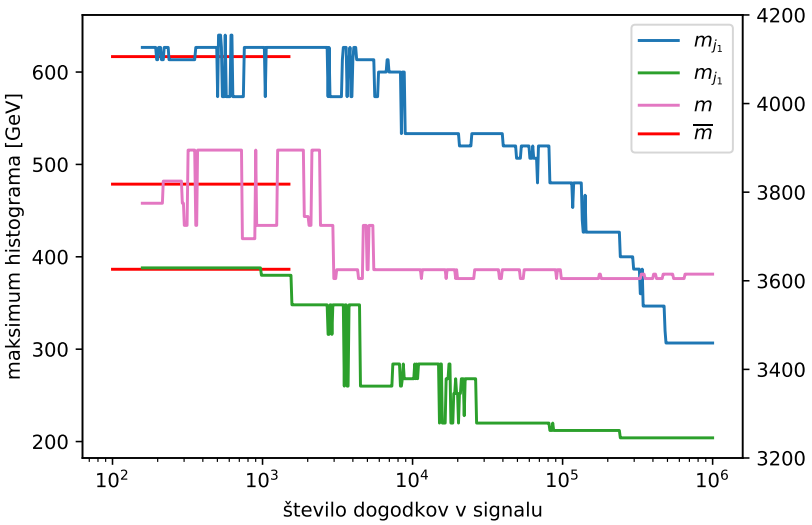
Slika 18: Primerjava vhodne in izhodne porazdelitve VAE za dogodke iz učne množice $S/B = 10\,\%$.



Slika 19: Rekonstrukcija signala iz dogodkov črne škatle.

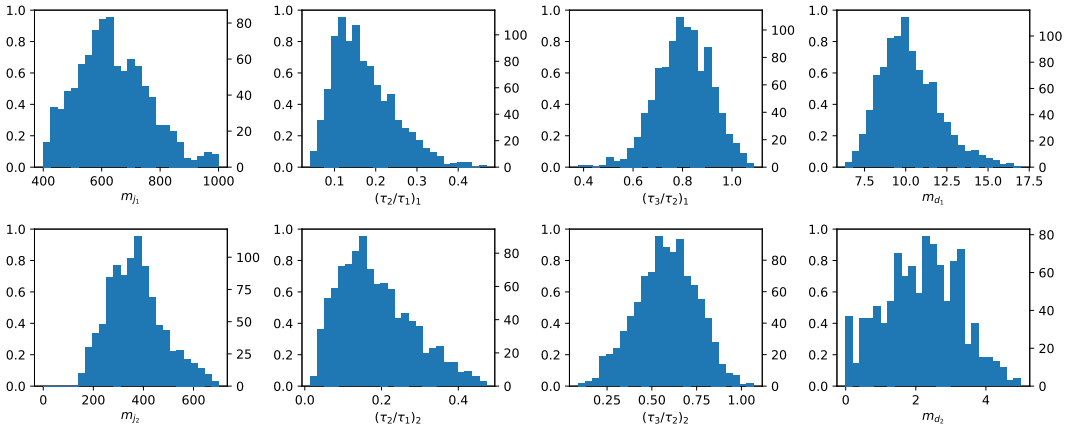
premakne proti nižjim vrednostim. Končno oceno mas smo izračunali s povprečenjem vrednosti od vključenih 100 do 1300 dogodkov, kot je prikazano z rdečo črto na grafu:

$$m_X = (617 \pm 19)\,\mathrm{GeV}$$
$$m_Y = (387 \pm 3)\,\mathrm{GeV}$$
$$m_{Z'} = (3818 \pm 77)\,\mathrm{GeV}$$



Slika 20: Položaj maksimuma resonančnega vrha z naraščanjem števila vključenih dogodkov iz signala.

Na sliki 21 so prikazani še vsi parametri dogodkov pri vključenih 1000 dogodkih signala.



Slika 21: Porazdelitev vseh osmih parametrov v 1000 dogodkih signala.

5 Zaključek

V nalogi smo obravnavali variacijske avtoenkoderje, s katerimi smo v prvem delu generirali dvodimenzionalni latentni prostor števk iz učne množice MNIST, v drugem delu pa jih uporabili za iskanje signala v simuliranih dogodkih LHC. Na primeru števk smo si pogledali vpliv razmerja med pomembnostjo rekonstrukcijske izgube in regularizacijske izgube s KL-divergenco. Uspešno smo določili mase neznanega delca in njegovih razpadnih produktov, tako da smo variacijski avtoenkoder, uporabili kot binarni klasifikator med dogodki ozadja in dogodki signala.