UNIVERZA V LJUBLJANI
FAKULTETA ZA MATEMATIKO IN FIZIKO
ODDELEK ZA FIZIKO

PRAKTIKUM STROJNEGA UČENJA V FIZIKI

7. naloga: VAE in nova fizika

Žiga Šinigoj, 28222025

Ljubljana, februar 2024

1 Uvod

Metode strojnega učenja kažejo velik potencial za uporabo prav v iskanju nove fizike v kontekstu visokoenergijskih trkalnikov. V tem primeru ne poznamo oblike signala, zato je potrebno uporabiti metode nenadzorovanega učenja. Primer take metode so Variacijski AvtoEnkoderji (VAE), ti so zelo uporabni v fiziki delcev. Cilj je s pomočjo latentnega prostora VAE izluščiti signal od ozadja za različne fizikalne procese. V naši nalogi bomo iskali dvocurkovni signal v morju dogodkov ozadja.

Naloga je razdeljena na 3 dele. V prvem delu se ukvarjamo z implementacijo binarne mešanice in klasifikacijo podatkovnega seta števk MNIST. V drugem delu naloge za klasifikacijo MNIST uporabimo VAE. V tretjem delu bomo s pomočjo VAE iskali dvocurkovni signal.

2 Binarna mešanica in MNIST

Množico MNIST sestavljajo slike števk od 0 do 9 velikosti 28 × 28 pikslov od vrednosti 0 do 255. Matrike najprej pretvorimo v vektor in normaliziramo vrednosti med 0 in 1. Vsako izmed 10 gruč opišemo s produktom Bernullijevih porazdelitev

$$P(x|\mu) = \prod_{i=1}^{28\times 28} \mu_i^{x_i}(1-\mu_i)^{(1-x_i)} \tag{1}$$

Celotna porazdelitev je linearna kombinacija posameznih porazdelitev

$$P(x) = \sum_{j=1}^{10} P(x|\mu^j)\pi_j \ , \tag{2}$$

kjer so π_j uteži, za katere velja $\sum_{j=1}^{10}\pi_j = 1$. V splošnem parametre π_j, μ_k^j dobimo z maksimizacijo log-likelihooda. V primeru binarnih mešanic pa sledimo algoritmu:

- Izberemo μ_k^j po enakomerni porazdelitvi med 0 in 1, izberemo π_j po Dirichletovi porazdelitvi
- Ponavljamo koraka:
  1. Izračunamo $P(x_l, j) = \frac{P(x_l|j)}{P(x_l)}$ za vse slike in gruče
  2. Posodobimo parametre

$$\mu^j = \frac{\sum_x P(x|j)x}{\sum_x P(x|j)} \tag{3}$$

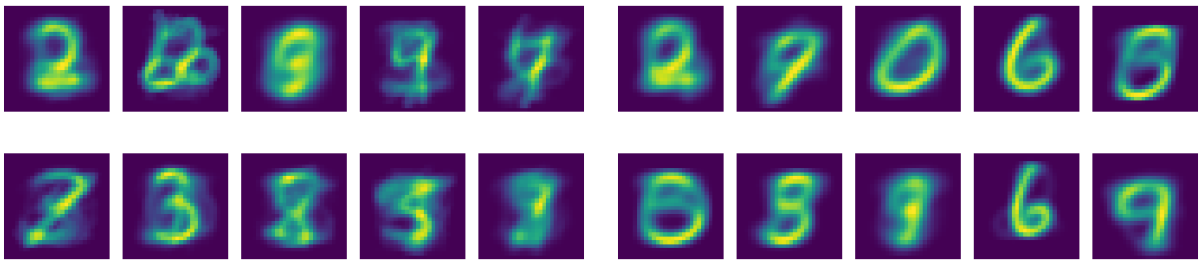
$$\pi_j = \frac{\sum_x P(x|j)}{\sum_{j=1}^{10}\sum_x P(x|j)} \tag{4}$$

Za boljšo numerično stabilnost sem enačbo (1) logaritmiral in računal vsoto namesto produkta. Preden sem apliciral eksponentno funkcijo, sem še odštel faktor $B = max(log(P(x_l|\mu^j))$, kjer je maksimum vzet po vseh gručah. Na ta način sem verjetnost, da dana slika pripada dani gruči, računal kot

$$P(x_l, \mu^j) = exp(log(P(x|\mu)) - B) \ . \tag{5}$$

Iskane parametre μ^j pri različnih iteracijah algoritma prikazujejo slike (1), (2), in (3a). Algoritem slabo reproducira števke po prvi iteraciji. Po drugi iteraciji pa je že mogoče prepoznati števke 6,9,0,8 in 2. Z večanjem števila iteracij se nekoliko izboljšujejo slike, ampak se izboljševanje neha že pri 5 oziroma 10 iteraciji. Zanimivo je, da je mogoče razbrati različne števke pri različnih iteracijah algoritma. Po 10 iteracijah približno dosežemo končno stanje. Rezultati 10 in 20 iteracije vsebujejo približno enake števke (razlika je ta, da se ena trojka spreminja v petko in neznana števka v 10 iteraciji se spreminja v 6 v 20 iteraciji).

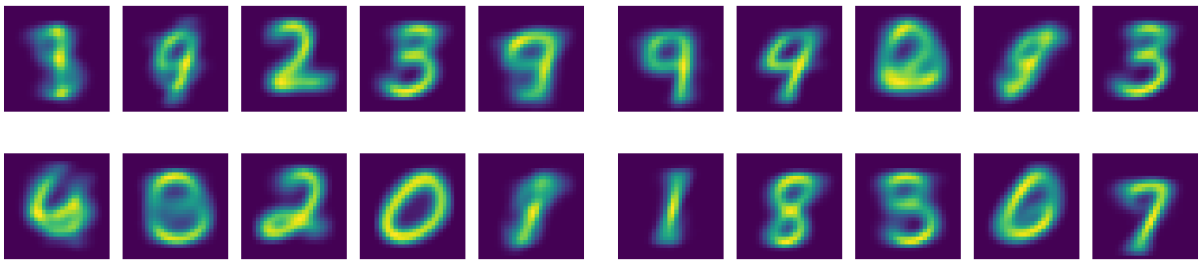
Nato sem za dve števki (gruči) iz slike (2a) (števka 2 in neznana števka v drugem stolpcu in drugi vrstici) generiral slike (slika 5) iz binomske porazdelitve. Opazimo lahko, da se z večanjem n slika izboljšuje. V primeru, da je mogoče razbrati števko, večji n naredi števko še bolj prepoznavno. V primeru, ko števke ni



(a) (b)

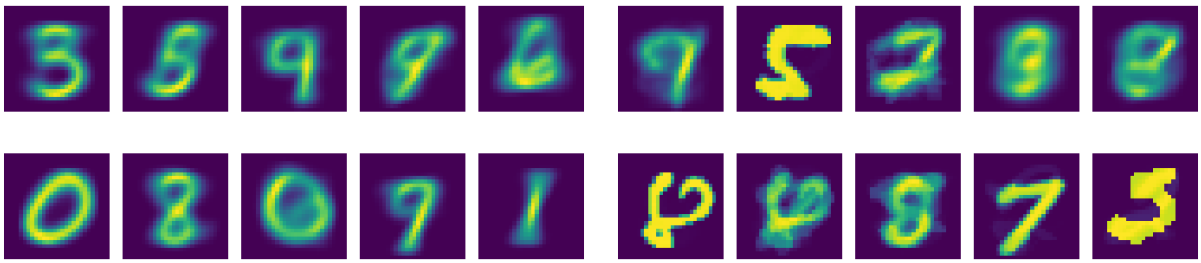
Slika 1: a) Uteži μ_i po 1 iteraciji algoritma. b) Uteži μ_i po 2 iteracijah algoritma.

mogoče razbrati, pa tudi večji n ne pomaga. Da bi preveril, če binomske mešanice res generirajo lepše slike, sem uporabil binomsko mešanico $n = 2$ namesto binarne in pognal algoritem. Rezultate prikazujejo slike (3b), (4). Algoritem je deloval do 3. iteracije, potem je začel divergirati. Opazimo lahko, da so nekatere števke bolj izrazite, nekatere pa še manj kot pri binarni mešanici. Razlog je mogoče ta, da algoritem prepozna dve enaki števki, kot različni gruči, če sta napisani na različna načina.



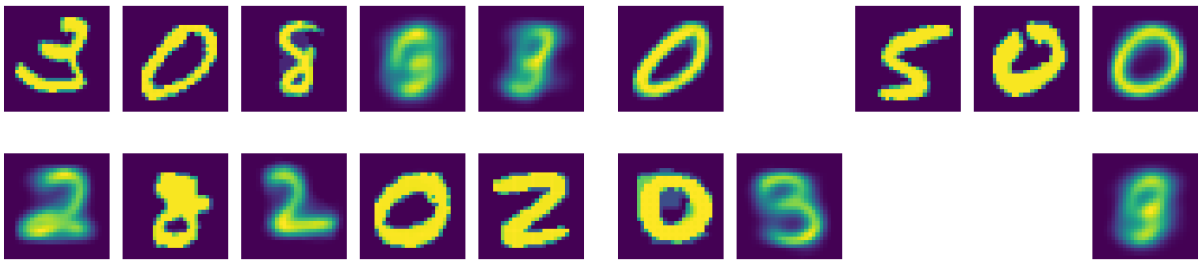
(a) (b)

Slika 2: a) Uteži μ_i po 5 iteracijah algoritma. b) Uteži μ_i po 10 iteracijah algoritma.

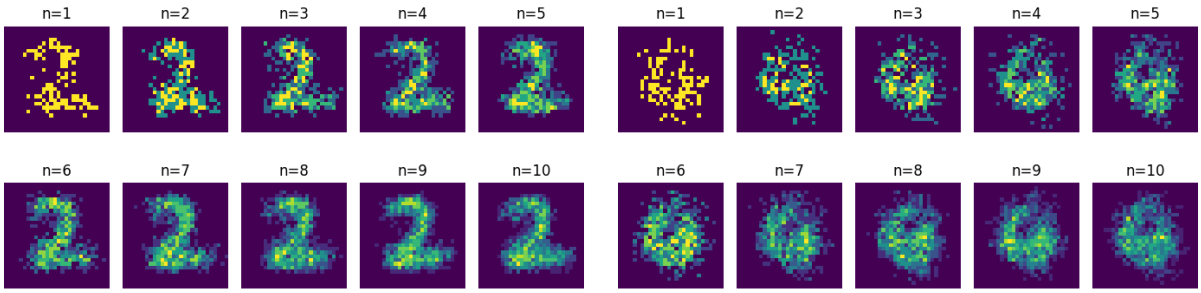


(a) (b)

Slika 3: a) Uteži μ_i po 20 iteracijah algoritma. b) Uteži μ_i po 1 iteraciji algoritma, binomska porazdelitev ($n = 2$).



Slika 4: a) Uteži μ_i po 2 iteracijah algoritma, binomska porazdelitev ($n = 2$). b) Uteži μ_i po 3 iteracijah algoritma, binomska porazdelitev ($n = 2$). Nekatere gruče so divergirale že pri treh iteracijah, zato za njih ni slik.

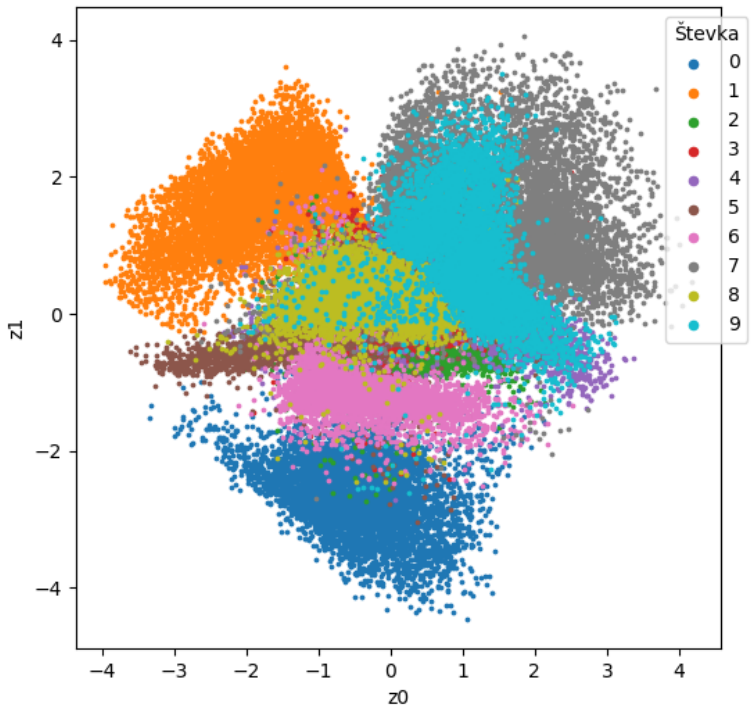


Slika 5: a) Generiranje novih slik za različno binomsko porazdelitev n po petih iteracijah algoritma. Žrebanje iz porazdelitve $P(x, \mu^j)$, ki ustreza števki 2 (1 vrstica, 3 stolpec iz slike 2a). b) Generiranje novih slik za različno binomsko porazdelitev n po petih iteracijah algoritma. Žrebanje iz porazdelitve $P(x|\mu^j)$, ki ustreza neznani števki (2 vrstica, 2 stolpec iz slike 2a).

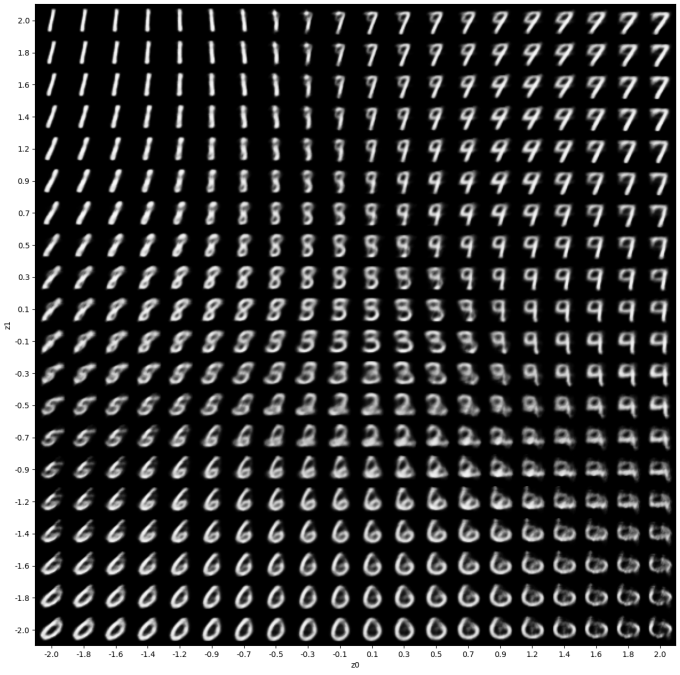
3 VAE in MNIST

V drugem delu naloge še vedno poskušamo klasificirati podatkovni set MNIST. Po navodilih iz naloge sem implementiral VAE. Uporabil sem *adam* optimizator ter treniral do 100 epoh. Števke smo transformirali v 2D latentni prostor (z0,z1). Latentni prostor in mesta posameznih števk v njem prikazuje slika (6). V latentnem prostoru so si najbolj oddaljene števke 0,1 in 7. Očitno imajo te števke najmanj podobnosti med sabo, vsaj tako predvideva VAE. Veliko je prekrivanja med števkama 7 in 9. Veliko slabše so ločene preostale števke, ki se nahajajo okrog izhodišča. Relativno dobro izstopata še števki 5 in 6. V moji implementaciji skoraj vedno ločim števke 0,3,7,9 in mogoče tudi števko 8 in 2. Pri VAE se števke 8,2,3 zelo prekrivajo.

Če sedaj uporabimo dekoderski del VAE, lahko dobimo števke v prvotnem prostoru za vsako točko iz latentnega prostora. Da približno zajamem vse števke sem vzel mrežo velikosti 2×2 centrirano v izhodišču s 400 točkami in rekonstruiral števko za vsako točko na mreži. Rezultat prikazuje slika (7). Rekonstruiramo lahko skoraj vse števke. Števke 4 nisem našel na sliki. Na sliki so lepo vidni zvezni prehodi med števkami. Nekatere števke (npr. 0,1,7,9) so lepo vidne, druge malo manj, odvisno od pozicije, sosedov in razdalj med njimi v latentnem prostoru.



Slika 6: Gručenje podatkov MNIST v latentnem prostoru VAE.



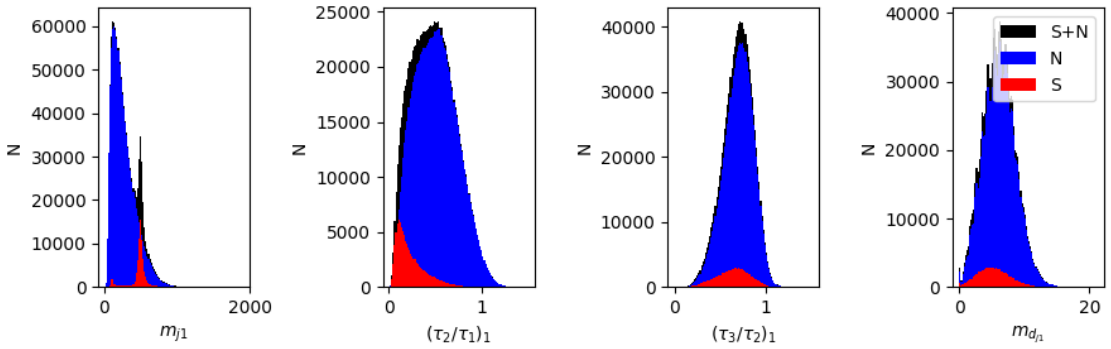
Slika 7: Generiranje slik iz latentnega prostora VAE. Mreža (20×20) zajema interval $z_0, z_1 \in [-2, 2]$ v latentnem prostoru.

4 VAE in mase neznanih delcev

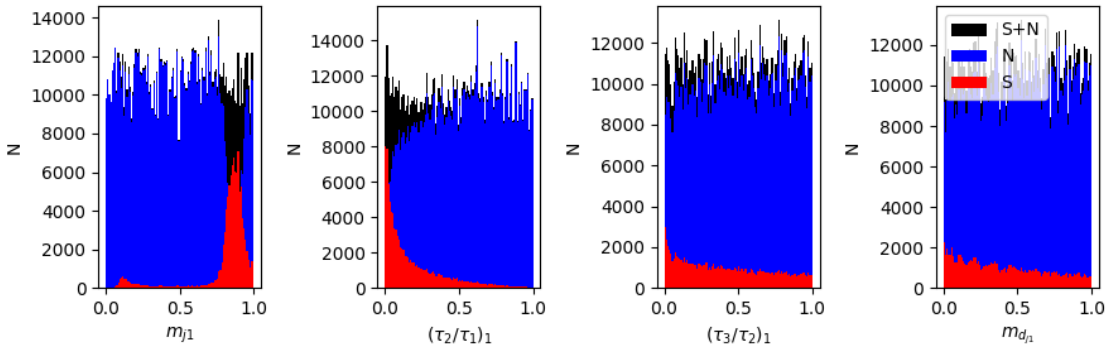
V zadnjem delu naloge smo uporabili VAE za izračun mas neznanih delcev pri trku dveh protonov. Najprej smo natrenirali model na klasificiranih dogodkih (testni podatki) in iz rezultatov in analize, lahko sklepamo nekaj o neznanih podatkih (podatki črne škatle).

4.1 LHCO podatki

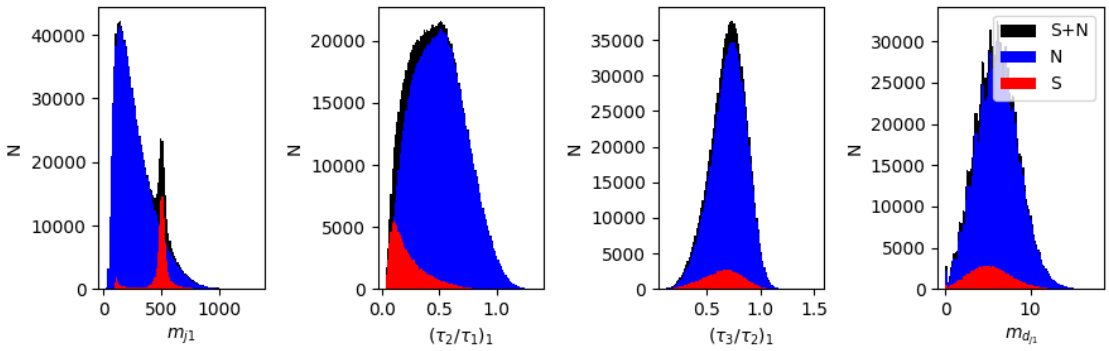
Testni podatki so označeni in vsak dogodek pripada signalu ali ozadju (1 ali 0). Na voljo sta bila 2 seta podatkov, en set z razmerjem $S/B = 10\%$ in en set z razmerjem $S/B = 0.1\%$. Najprej si oglejmo porazdelitev prvih 4 spremenljivk pri $S/B = 10\%$, ki pripadajo enemu curku (slika 8). Nato smo podatke transformirali s $QuantileTransformer$ v porazdelitve na sliki (9). Porazdelitve lahko z inverzno transformacijo pretvorimo v originalne (slika 10). Začetne in dvakrat transformirane porazdelitve na prvi pogled izgledajo enako, vendar bolj podroben ogled kaže na izgubo informacije ali pa na širitev porazdelitev. Amplitude se na vseh porazdelitvah malo zmanjšajo, oblika ostaja približno enaka. Mogoče bi bila tukaj potrebna bolj podrobna analiza in primerjava končne in začetne porazdelitve.



Slika 8: Porazdelitev signala, ozadja in obojega skupaj prvih 4 spremenljivk za podatke $S/B = 10\%$



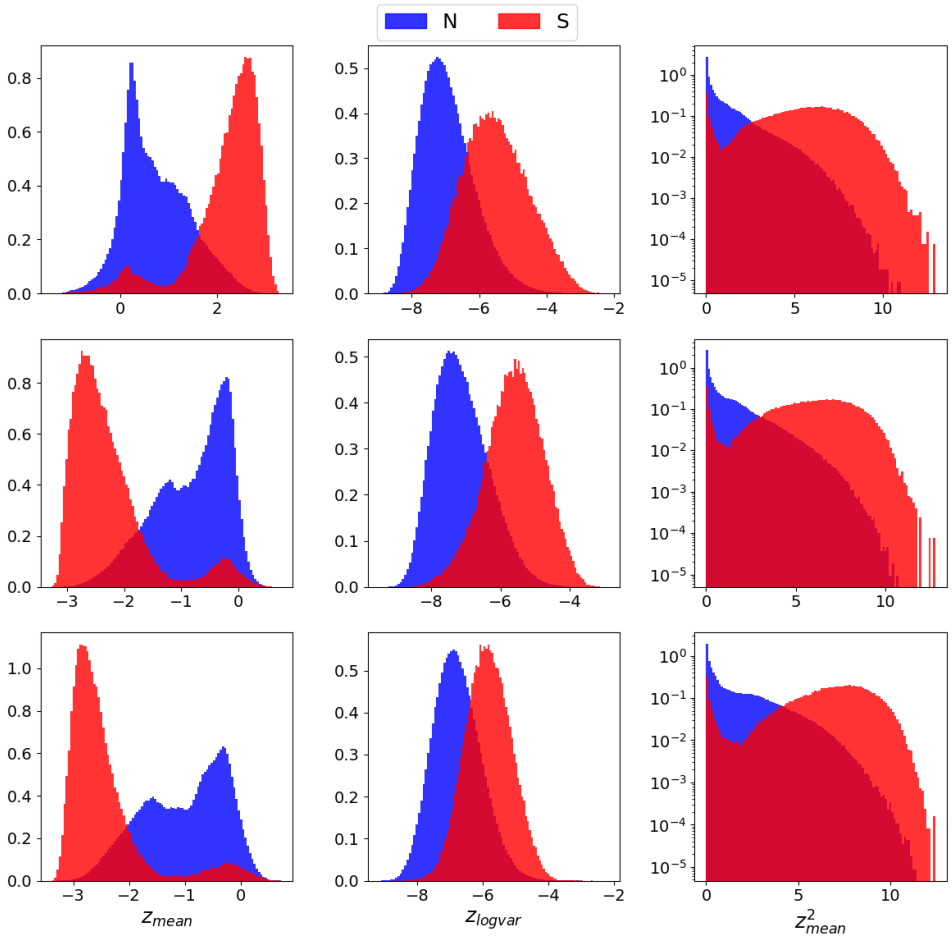
Slika 9: Porazdelitev signala, ozadja in obojega skupaj prvih 4 spremenljivk za podatke $S/B = 10\%$ po transformaciji $QuantileTransformer$.



Slika 10: Porazdelitev signala, ozadja in obojega skupaj prvih 4 spremenljivk za podatke $S/B = 10\%$ po transformaciji $QuantileTransformer$, ter po inverzni transformaciji v začetno porazdelitev.

4.2 Latentni prostor in AUC

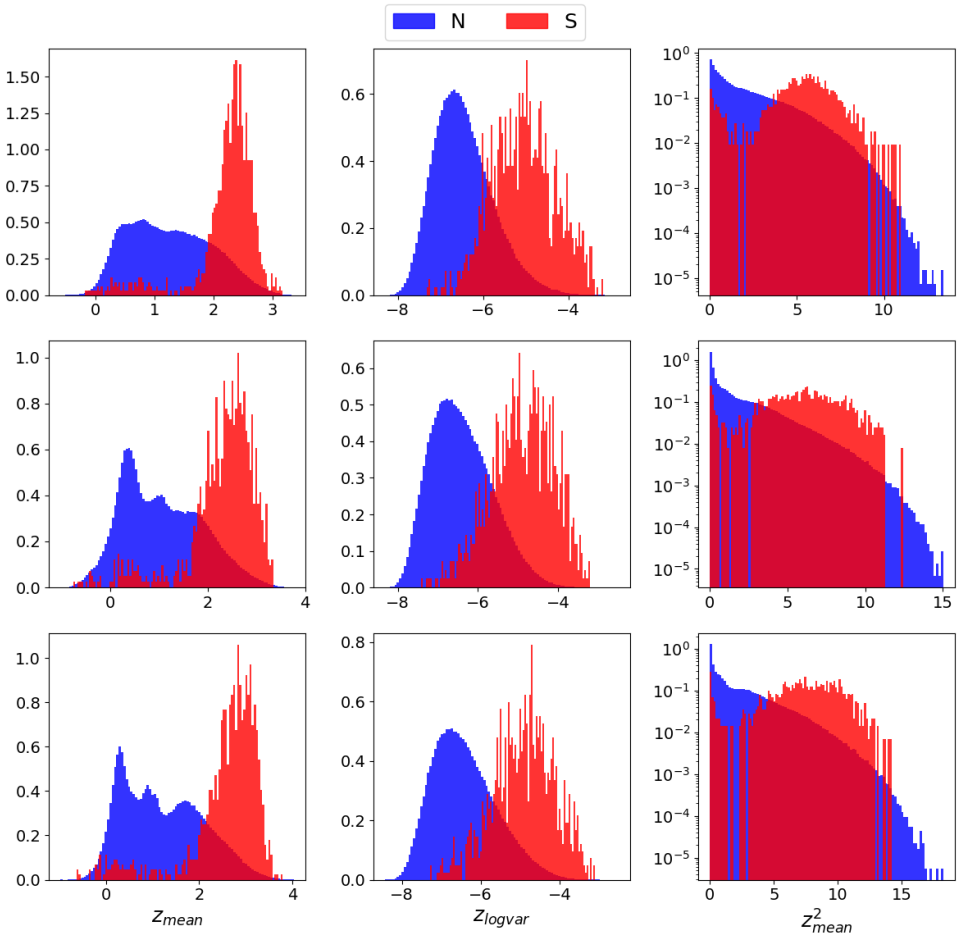
Za implementacijo VAE lahko sedaj uporabimo VAE iz prejšnje naloge, ki ga spremenimo po navodilih. Sedaj uporabimo optimizator *Adadelta* in treniramo do 100 epoch. Natreniral sem 3 VAE z enakimi parametri. Sedaj transformiramo dogodke pri $S/B = 10\%$ in $S/B = 0.1\%$ v latentni prostor za vse 3 VAE modele in za dane klasifikatorje (slika 11 in 12). Vsaka vrstica predstavlja rezultate enega naučenega modela $VAEi, i \in$



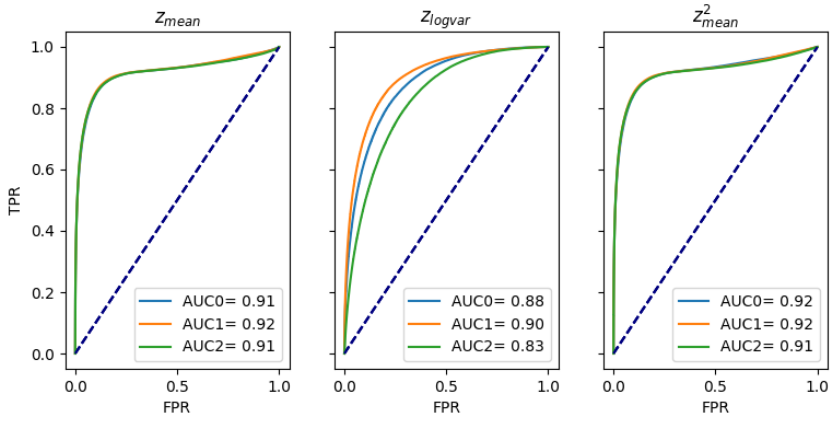
Slika 11: Normalizirana porazdelitev latentnih spremenljivk $z_{mean}, z_{logvar}, z^2_{mean}$ signala in ozadja za tri različne VAE z enakimi parametri, $S/B = 10\%$.

$0, 1, 2$. V splošnem lahko opazimo, da klasifikator z_{mean} ni optimalen, saj lahko zasuka porazdelitvi signala in ozadja in je odvisen od inicializacije uteži VAE. Pri klasifikatorju z_{logvar} in z^2_{mean} pa se porazdelitev

signala vedno nahaja na desni in porazdelitev ozadja na levi. Pri $S/B = 10\%$ je nekoliko manj prekrivanja kot pri $S/B = 0.1\%$, sploh za klasifikator z^2_{mean}.



Slika 12: Normalizirana porazdelitev latentnih spremenljivk $z_{mean}, z_{logvar}, z^2_{mean}$ signala in ozadja za tri različne VAE z enakimi parametri, $S/B = 0.1\%$.

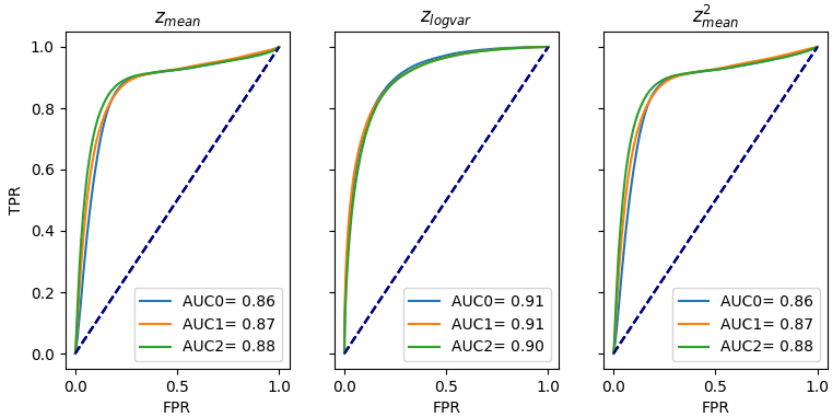


Slika 13: ROC krivulja za vsak VAE (0,1,2) za različne klasifikatorje $z_{mean}, z_{logvar}, z^2_{mean}$ in pripadajoča ploščina pod krivuljo (AUCi, $i = 0, 1, 2$), $S/B = 10\%$.

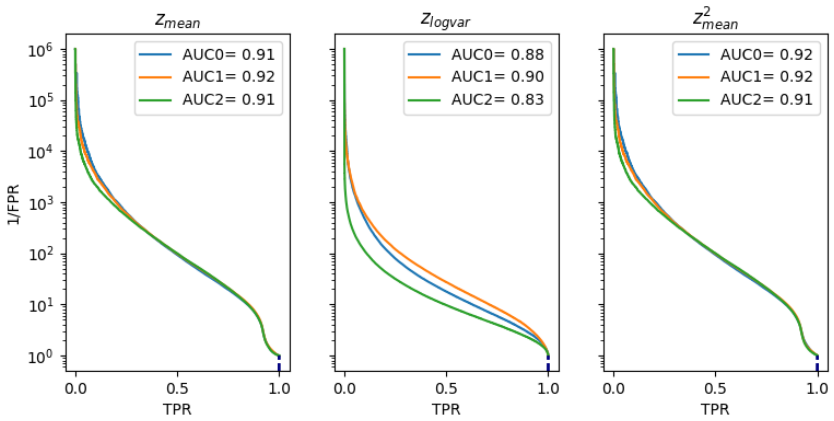
Za vsak VAE model sem narisal ROC krivuljo za vse 3 klasifikatorje na podatkih $S/B = 10\%$ in $S/B = 0.1\%$ (slika 13 in 14). V povprečju se najslabše obnese klasifikator z_{mean} in najboljše klasifikator z_{logvar}. Tudi klasifikator z^2_{mean} da dobre rezultate in je primerljiv z z_{logvar}.

Na podoben način sem narisal tudi grafe 1/FPR v odvisnosti od TPR (slika 15-$S/B = 10\%$ in 16-$S/B =$

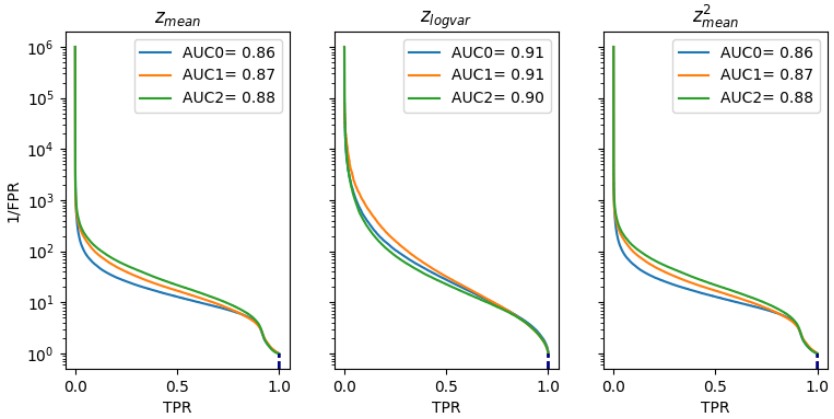
0.1%). Za optimalen klasifikator sem izbral z_{logvar} in 2. model iz ansambla (indeks 1). Ta model bom uporabil za nadaljnje izračune.



Slika 14: ROC krivulja za vsak VAE (0,1,2) za različne klasifikatorje $z_{mean}, z_{logvar}, z^2_{mean}$ in pripadajoča ploščina pod krivuljo (AUCi, $i = 1, 2, 3$), $S/B = 0.1\%$.



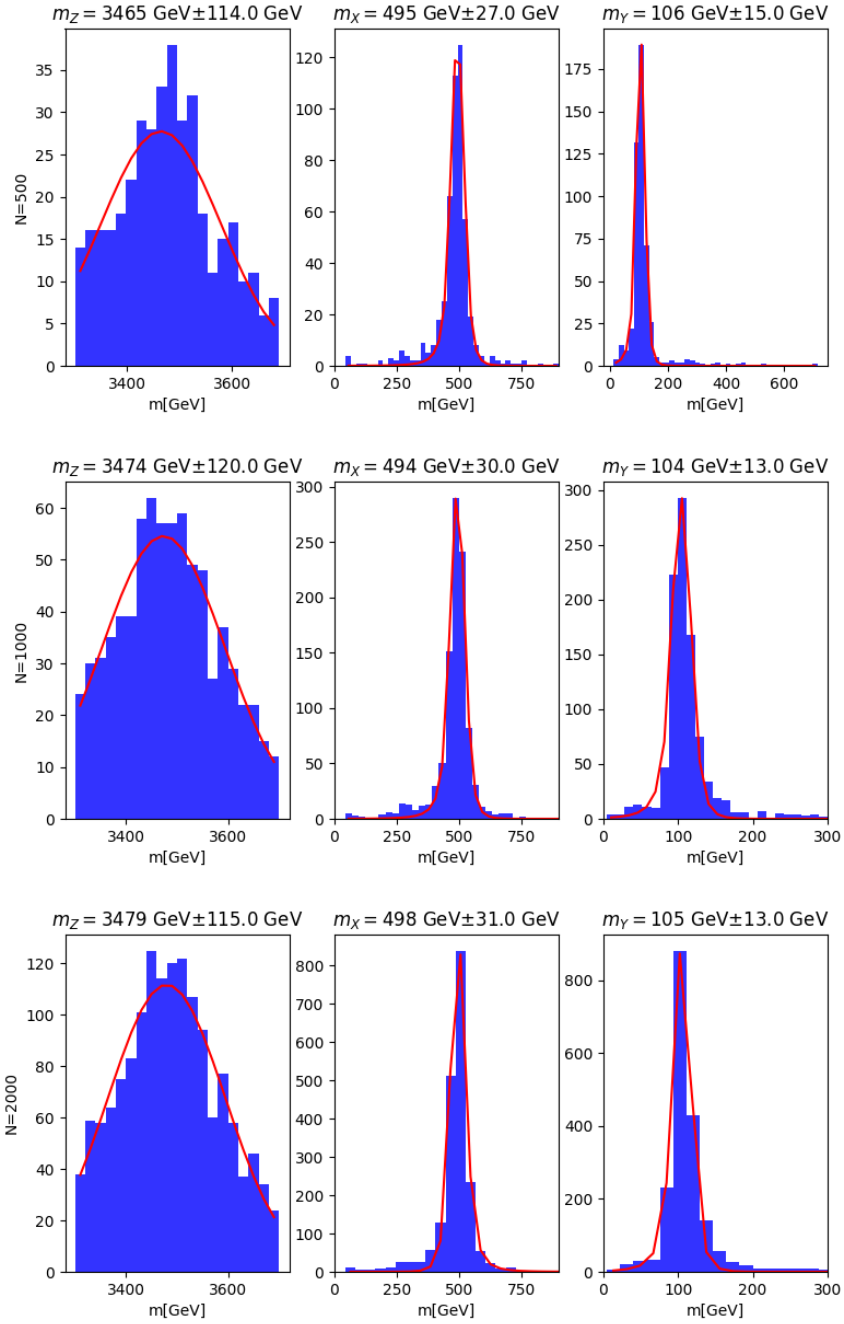
Slika 15: Graf krivulje 1/FPR v odvisnosti od TPR za vsak VAE in za različne klasifikatorje. $S/B = 10\%$



Slika 16: Graf krivulje 1/FPR v odvisnosti od TPR za vsak VAE in za različne klasifikatorje. $S/B = 0.1\%$

4.3 Izračun mase in rekonstrukcija podatkov

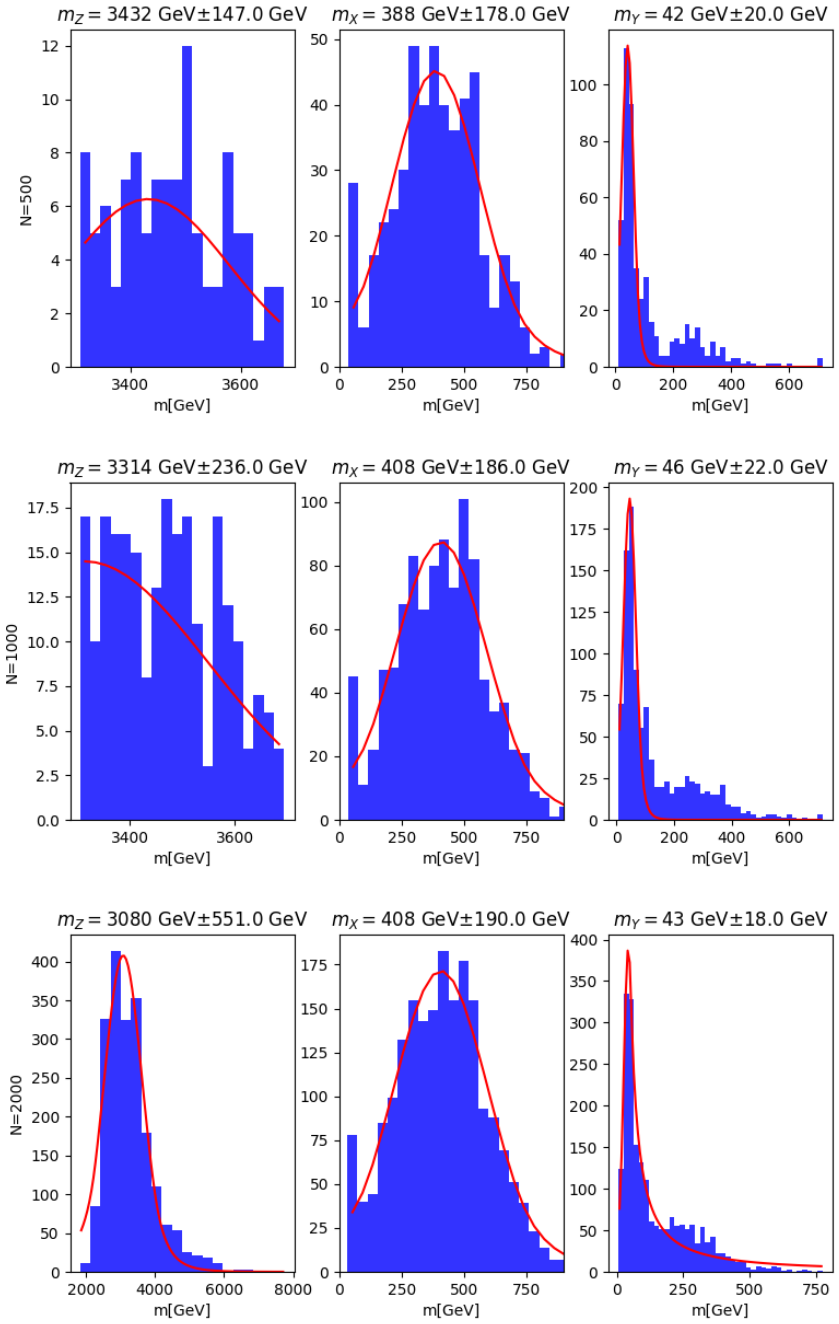
Z izbranim VAE (2. model, rezultati se nanašajo na indeks 1, z_{logvar} klasifikator) sem podatke preslikal v latentni prostor in določil 500, 1000, 2000 največjih vrednosti klasifikatorja z_{logvar} in na ta način dobil indekse 500, 1000, 2000 dogodkov, ki najverjetneje predstavljajo signal. Narisal sem histograme invariantne mase m_Z in mase m_X ter m_Y. Na histograme sem prilagajal CB funkcijo in na ta način določil iskane mase delcev (slika 17, 18). Pri prilagajanju sem začetne parametre ročno spreminjal in glede na občutek določil 'optimalno' prilagajanje, ki ni nujno optimalno. Za končne rezultate sem vzel rezultate pri $N = 500$, saj gre za najbolj čist vzorec. Pri določevanju napake sem za napako mase vzel kar σ_{CB}. Čeprav dobim



Slika 17: Histogrami porazdelitev mas m_Z, m_X, m_Y za različne $N \in 500, 1000, 2000$. Nad vsakim histogramom je prikazana masa in njena nedoločenost, kot rezultat prilagajanja CB funkcije podatkom (rdeča krivulja), $S/B = 10\%$.

pri prilagajanju CB funkcije še napako povprečja (mase) in napako σ_{CB} sta ti običajno manjši od σ_{CB}. Mogoče bi bilo smiselno če bi jih seštel kvadratično, ampak sem naposled kar vzel širino porazdelitve σ_{CB} za napako mase. Končni rezultat za $S/B = 10\%$ je tako $m_Z = (3465 \pm 114)\ GeV$, $m_X = (495 \pm 27)\ GeV$,

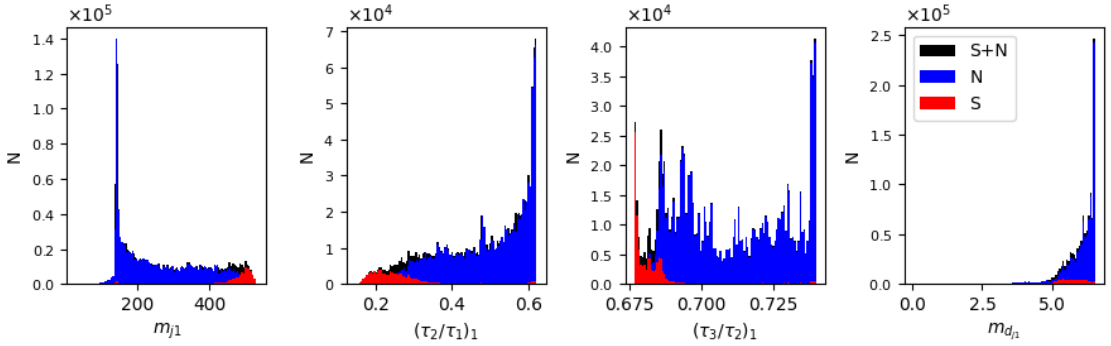
$m_Y = (106 \pm 15)\ GeV$. Pravi rezultati so $m'_Z = 3500\ GeV$, $m'_X = 500\ GeV$, $m'_Y = 100\ GeV$.



Slika 18: Histogrami porazdelitev mas m_Z, m_X, m_Y za različne $N \in 500, 1000, 2000$. Nad vsakim histogramom je prikazana masa in njena nedoločenost, kot rezultat prilagajanja CB funkcije podatkom (rdeča krivulja), $S/B = 0.1\%$.

Ko imamo signala manj, $S/B = 0.1\%$ (slika 18), so rezultati precej slabši. Končni rezultati so $m_Z = (3432 \pm 147)\ GeV$, $m_X = (388 \pm 178)\ GeV$, $m_Y = (42 \pm 20)\ GeV$. Edini rezultat, ki se ne sklada v okviru napake, je masa m_Y.

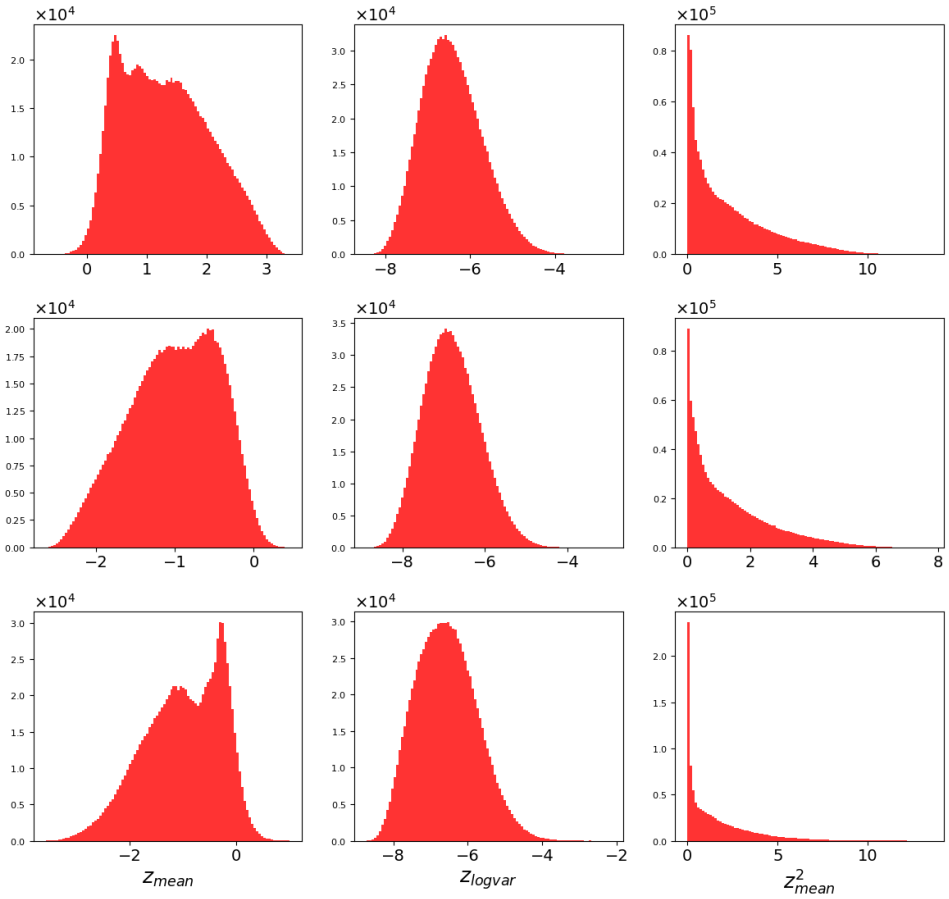
Če sedaj poskušamo rekonstruirati porazdelitve prvih 4 spremenljivk iz latentnega prostora z najboljšim VAE dobimo rezultat na sliki (19). Porazdelitve se precej razlikujejo od porazdelitev na sliki (10). Iz tega lahko zaključimo, da VAE ni najboljši v rekonstrukciji porazdelitev.



Slika 19: Histogrami porazdelitev prvih štirih spremenljivk po preslikavi z enkoderjem v latentni prostor in nato z dekoderjem v prvotni prostor in inverzno transformacijo $QuantileTransformer$, $S/B = 10\%$.

4.3.1 Podatki črne škatle

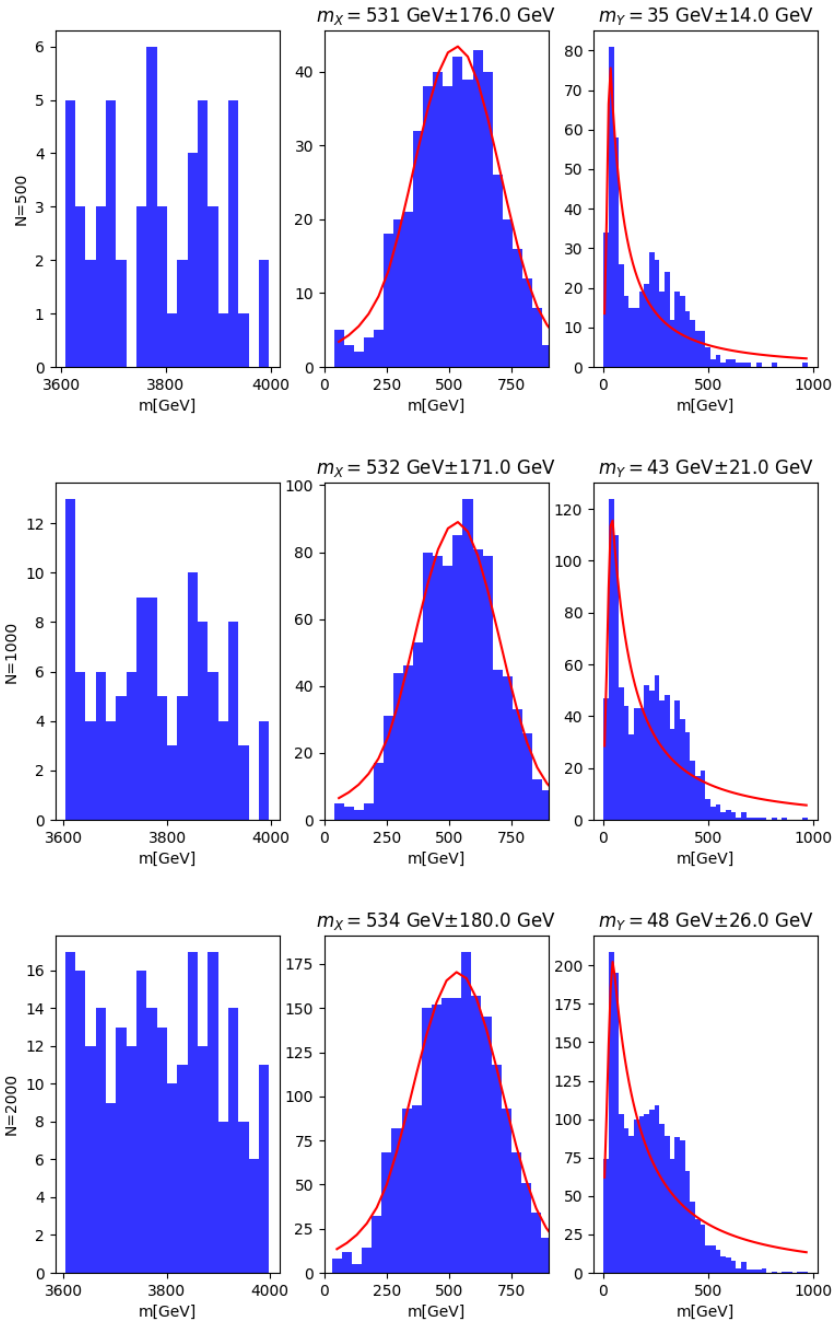
V zadnjem delu je bilo potrebno določiti mase m_Z, m_X, m_Y na neoznačenih podatkih. Podobno kot prej sem učil 3 VAE modele na podatkih črne škatle. Porazdelitve klasifikatorjev prikazuje slika (20). V tem primeru je mogoče razbrati več vrhov samo iz klasifikatorja z_mean. Vemo pa da klasifikator z_mean ni zanesljiv. Za določanje mase bom najprej uporabil najboljši klasifikator za označene podatke z_{logvar} (slika 21).



Slika 20: Porazdelitev vseh dogodkov črne škatle v latentnem prostoru za različne klasifikatorje $(z_{mean}, z_{logvar}, z^2_{mean})$ in vse (tri) natrenirane VAE.

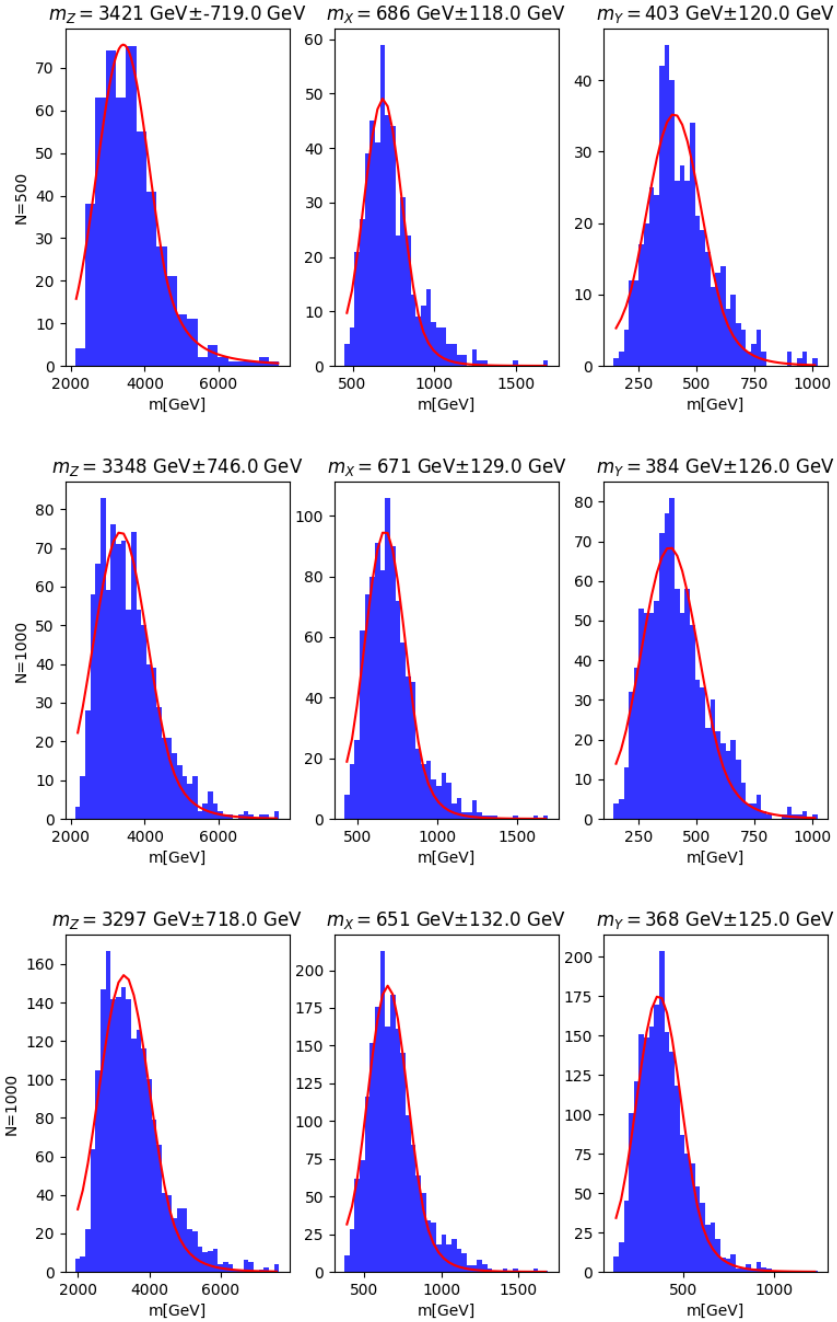
V tem primeru nisem mogel določiti mase m_Z za podan interval mas, kjer naj bi se nahajala. Za ostali masi pa dobim rezultat $m_X = (531 \pm 176)\ GeV$, $m_Y = (35 \pm 14)\ GeV$. Ti rezultati so v okviru napake podobni

rezultatom ko imamo testne podatke z $S/B = 0.1\%$. Vseeno pa je nisem prepričan če je to prava rešitev, saj že namig določa drugačno okno invariantne mase. V ta namen sem pogledal še mase delcev, če vzamem klasifikator z^2_{mean}.



Slika 21: Histogrami porazdelitev mas m_Z, m_X, m_Y na podatkih iz črne škatle za različne $N \in 500, 1000, 2000$. Nad vsakim histogramom je prikazana masa in njena nedoločenost, kot rezultat prilagajanja CB funkcije podatkom (rdeča krivulja).

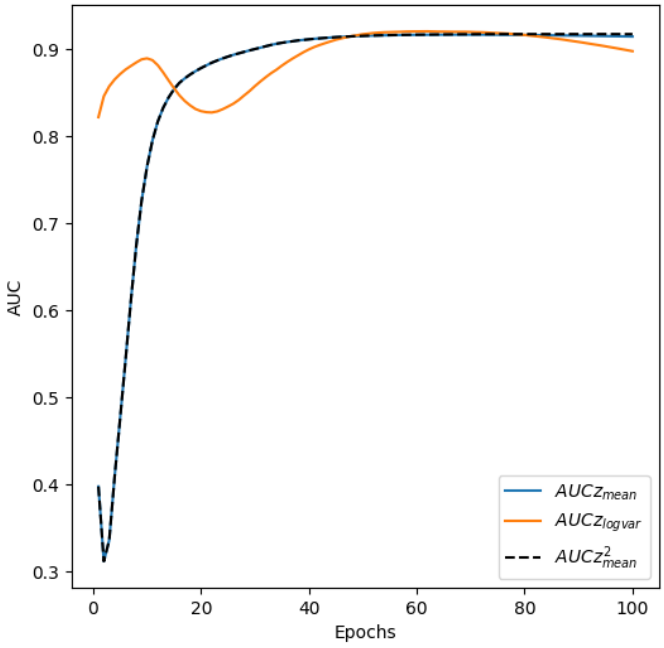
Rezultati so prikazani na sliki 22. Končni rezultati so tako $m_Z = (3421 \pm 719)\ GeV$, $m_X = (686 \pm 118)\ GeV$, $m_Y = (403 \pm 120)\ GeV$. Ker je m_Z v okviru napake v danem oknu, kjer pričakujemo maso, je to najverjetneje pravi rezultat.



Slika 22: Histogrami porazdelitev mas m_Z, m_X, m_Y na podatkih iz črne škatle za različne $N \in 500, 1000, 2000$ in uporabljenim klasifikatorjem z^2_{mean}. Nad vsakim histogramom je prikazana masa in njena nedoločenost, kot rezultat prilagajanja CB funkcije podatkom (rdeča krivulja)

5 AUC med učenjem

Preveril sem še kako se spreminja AUC tekom učenja. Za vsako epocho sem izračunal AUC za vse 3 klasifikatorje. VAE sem učil na labeliranih podatkih $S/B = 10\%$. Rezultate prikazuje slika (23). V tem primeru sta celo klasifikator z_{mean} in z^2_{mean} boljša pri 100 epohah. Pri okrog 60 epohah pa je boljši klasifikator z_{logvar}, kateremu kasneje začne padati uspešnost.



Slika 23: AUC v odvisnosti od epohe tekom učenja VAE za vse tri klasifikatorje.

6 Zaključek

V nalogi smo najprej implementirali algoritem binarne mešanice in klasificirali podatkovno množico MNIST. Algoritem je bil precej uspešen, kljub preprostosti. Klasifikacijo MNIST smo opravili tudi z VAE, ki je pričakovano dal boljše rezultate. V drugem delu naloge smo s pomočjo VAE določili mase iskanih delcev. S pomočjo preslikave v latentni prostor je bilo mogoče učinkovito ločiti ozadje od signala.