

Zastosowanie autoenkoderów wariacyjnych do rozpoznawania zmian na obrazach medycznych

()

Tomasz Nanowski

Praca inżynierska

Promotor: dr Jan Chorowski

Uniwersytet Wrocławski
Wydział Matematyki i Informatyki
Instytut Informatyki

2 lutego 2019

Tomasz Nanowski

.....

.....

(adres zameldowania)

.....

.....

(adres korespondencyjny)

PESEL:

e-mail:

Wydział Matematyki i Informatyki

stacjonarne studia I stopnia

kierunek: informatyka

nr albumu: 279076

Oświadczenie o autorskim wykonaniu pracy dyplomowej

Niniejszym oświadczam, że złożoną do oceny pracę zatytułowaną *Zastosowanie autoenkoderów wariacyjnych do rozpoznawania zmian na obrazach medycznych* wykonałem/am samodzielnie pod kierunkiem promotora, dr. Jana Chorowskiego. Oświadczam, że powyższe dane są zgodne ze stanem faktycznym i znane mi są przepisy ustawy z dn. 4 lutego 1994 r. o prawie autorskim i prawach pokrewnych (tekst jednolity: Dz. U. z 2006 r. nr 90, poz. 637, z późniejszymi zmianami) oraz że treść pracy dyplomowej przedstawionej do obrony, zawarta na przekazanym nośniku elektronicznym, jest identyczna z jej wersją drukowaną.

Wrocław, 2 lutego 2019

(czytelny podpis)

Streszczenie



W tej pracy chciałem sprawdzić skuteczność wykorzystania metody uczenia bez nadzoru w problemie lokalizowania zmian nowotworowych na zdjęciach wykonanych metodą rezonansu magnetycznego. Bazowałem na podstawie obserwacji, że zmiany patologiczne są rzadkie i są pewnym odstępstwem od normy. Zdecydowałem się na użycie autoenkodera wariacyjnego, który pozwala oszacować prawdopodobieństwo wystąpienia danej próbki, co pozwoliłoby mi na ich klasyfikowanie. Pomysł przetestowałem na danych syntetycznych, wykorzystując do tego zbiór MNIST. Otrzymane wyniki okazały się być zadowalające i zachęcały do przeprowadzenia dalszych eksperymentów już na danych medycznych. Niestety w tym przypadku nie można było uznać tego za sukces, a według mojej analizy model nauczył się jedynie zwracać uwagę na prostą własność, jaką jest jasność próbki. Na tym zakończyłem moją pracę i przygotowałem propozycje rozwiązań, które mogłyby według mnie pozytywnie wpłynąć na poprawę rezultatów.

Spis treści

1. Wstęp	9
1.1. Przedstawienie problemu	9
1.2. Sztuczne sieci neuronowe	9
1.3. Uczenie nadzorowane i bez nadzoru	10
1.4. Autoenkoder	10
1.5. Autoenkoder wariacyjny	12
1.5.1. Opis matematyczny	14
1.6. Ocena jakości modelu	14
1.6.1. Krzywa ROC i AUC	14
2. Eksperyment na danych syntetycznych (MNIST)	17
2.1. Możliwości autoenkodera wariacyjnego	17
2.2. Symulacja docelowego problemu	19
2.3. Wnioski	21
3. Eksperyment na danych właściwych (MRI FLAIR)	23
3.1. Opis danych	23
3.2. Wstępna obróbka	24
3.2.1. Podział na mniejsze kawałki	24
3.2.2. Dodatkowe przygotowanie zbioru	25
3.3. Wyniki	26
3.4. Analiza	27
3.4.1. Wnioski	27

4. Summary**29**

Notes

 Dokończyć	14
 Dodać z-dim	26

Rozdział 1.

Wstęp

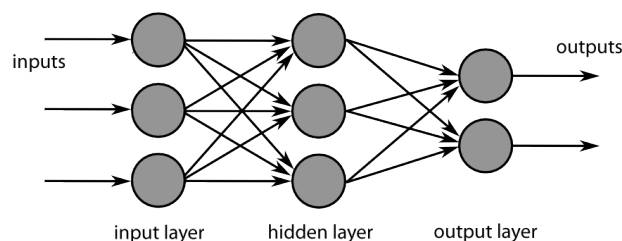
1.1. Przedstawienie problemu

Problem, który zainspirował mnie do napisania tej pracy to lokalizowanie zmian nowotworowych na zdjęciach wykonanych metodą rezonansu magnetycznego (MRI, ang. magnetic resonance imaging). Można podejść do niego od strony nadzorowanej, czyli bazować na danych przeanalizowanych wcześniej przez specjalistów, gdzie każdy przypadek został ręcznie obejrzany i oznaczony. O ile taka metoda ma wiele zalet, jak chociażby korzystanie z rzeczywistej wiedzy eksperckiej, to wykorzystywany zbiór jest bardzo kosztowny w przygotowaniu i dalszym rozwoju, a ponadto może być ograniczony jedynie do pojedynczej partii ciała. Zauważając te wady oraz łącząc je z obserwacją, że zmiany patologiczne tak na prawdę są rzadkie i są pewnym odstępstwem od normy, chciałbym spróbować skorzystać z metody nienadzorowanej, w której to model nauczyłby się oszacowywać prawdopodobieństwo występowania pojedynczej próbki w pewnym kontekście. Przy takim podejściu mógłbym oznaczać obserwacje mało prawdopodobne jako właśnie te nowotworowe. Model, na którego wykorzystanie się zdecydowałem to autoenkoder wariacyjny (VAE, ang. Variational Autoencoder), łączący sztuczne sieci neuronowe z modelowaniem probabilistycznym. Pomysł ten chce przetestować początkowo na danych syntetycznych z wykorzystaniem zbioru MNIST.

1.2. Sztuczne sieci neuronowe

Sztuczne sieci neuronowe mają obecnie bardzo mocno ugruntowaną pozycję szczególnie w dziedzinie problemów związanych z analizą i przetwarzaniem obrazów. Pomimo, iż nie jest to nowy pomysł, dopiero ostatni wzrost w wydajności komputerów pozwolił na ich praktyczne zastosowanie. Z matematycznego punktu widzenia są to sparametryzowane nieliniowe funkcje o pewnej ustalonej strukturze. Składają się z prostych elementów zwanych neuronami, a one natomiast są pogrupowane w

warstwy. Połączenia między warstwami definiują przepływ danych. 'Nauka' sieci neuronowych polega na optymalizacji pewnej funkcji straty, czyli wyznaczeniu takich parametrów, żeby osiągnąć minimalny koszt. Do tego celu często korzysta się z metod opartych na Stochastic Gradient Descent, a przy wybranej strukturze można w efektywny sposób zastosować algorytm propagacji wstecznej. W dalszej części pracy będę używał prostszej nazwy - sieci neuronowe. Przykładowa architektura sieci neuronowych jest zaprezentowana na wykresie 1.1.



Rysunek 1.1: Przykładowy model sieci neuronowej

1.3. Uczenie nadzorowane i bez nadzoru

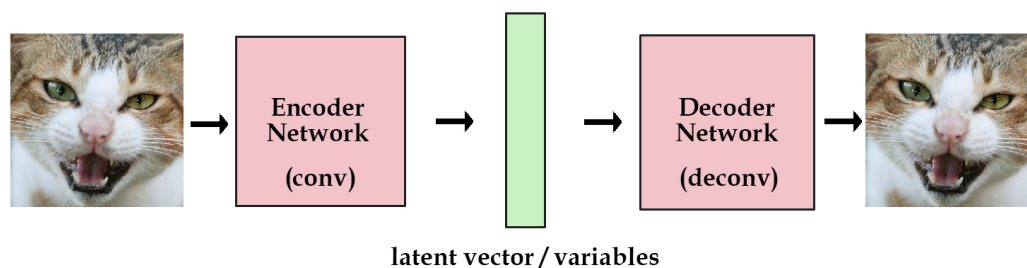
Są to dwa różne rodzaje problemów z dziedziny uczenia maszynowego. W przypadku uczenia nadzorowanego wiemy jaką chcemy uzyskać odpowiedź dla danego wejścia i próbujemy tak dobrać parametry modelu, żeby odpowiadał on z jak największą poprawnością. Przykładowymi zadaniami tego rodzaju są regresja oraz klasyfikacja, która nawiasem mówiąc może zostać uznana za problem regresji, ale na dyskretnym, skończonym zbiorze.

Z grubsza odwrotnie jest w sytuacji bez nadzoru. Tam dla danych wejściowych nie znamy interesującej nas odpowiedzi i staramy się tak zbudować model, żeby był on w stanie sam ją wydobyć. Zazwyczaj interesuje nas rozwiązanie takich problemów jak klasteryzacja, redukcja wymiarowości czy wydajne kodowanie danych. Jednym z modeli wykorzystywanym w tej klasie zadań jest właśnie autoenkoder.

1.4. Autoenkoder

Jest to jeden z rodzajai sieci neuronowych, służący do znajdowania wydajnej reprezentacji danych, co jak wspomniałem wcześniej jest przykładem nauki bez nadzoru. Myślę, że mogę go określić mianem nieliniowej alternatywy dla klasycznej statystycznej metody analizy głównych składowych (PCA, ang. principal component analysis). W autoenkoderach można wyróżnić dwie połączone ze sobą części zwane enkoderem i dekodere. Zadaniem enkodera jest wyprodukowanie właśnie tej reprezentacji, podczas gdy dekodery służy do odtworzenia z niej oryginalnej postaci. Zależy nam na tym, żeby wyjście było w jakimś sensie jak najbardziej podobne

do wejścia przy zachowaniu odpowiednio małej wymiarowości kodowania. Jest to pewnego rodzaju balansowanie pomiędzy ilością informacji, na których przepływać się zgadzamy, a ich jakością. W szczególności można pomyśleć o takiej patologicznej sytuacji jak ustalenie rozmiaru reprezentacji równej rozmiarowi danych wejściowych, co prawdopodobnie będzie skutkować idealnymi rekonstrukcjami, ale przy okazji zerową wiedzą płynącą z takiego modelu. Analogicznie można rozważyć przypadek kiedy obszar kodowania jest za mały. Model w takiej sytuacji skupi się jedynie na przekazaniu wyłącznie trywialnych cech, żeby mimo wszystko jakkolwiek odtworzyć dane. Poglądowy schemat budowy znajduje się na rysunku 1.2.



Rysunek 1.2: Architektura autoenkodera

W przypadku obrazów na funkcję straty często wybierany jest błąd średniokwadratowy (MSE, ang. Mean Squared Error).

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n \left(Y_i - \hat{Y}_i \right)^2 \quad (1.1)$$

gdzie:

Y_i – oryginalne dane

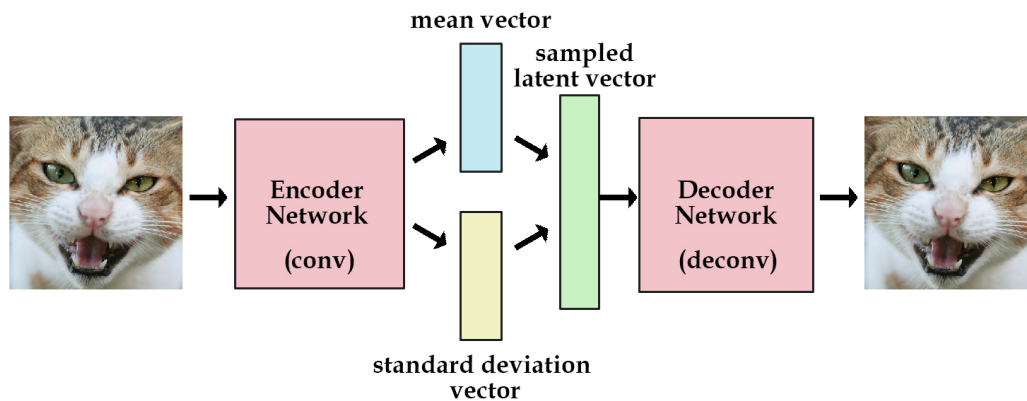
\hat{Y}_i – zrekonstruowane dane

Wymagającym sprostowania może być fakt, iż mimo mówimy jakie oczekujemy wyjście z modelu, w sensie rekonstrukcji, to dalej jest to problem typu nauki bez nadzoru, ponieważ rzeczywistą szukaną przez nas wiedzą jest umiejętność kodowania danych i osobno ich dekodowania. Chciałbym w tym miejscu zastanowić się co dzięki temu zyskujemy. Już teraz zaletą jest oczywiście redukcja wymiarowości, która ułatwia późniejszą analizę danych oraz pozwala na ich kompresję i ewentualne odsumowanie. Dodatkowo wydaje się, że mając wytrenowany dekodery, na konkretnej rodzinie danych, moglibyśmy zaaplikować do niego dowolne kodowanie i w ten sposób może otrzymać wynik pochodzący z oryginalnej dziedziny, czyli wykorzystać go jako generator nowych próbek. Niestety ten pomysł ma taki problem, że podczas nauki nie ma żadnej kontroli nad tym jak dane zostaną rozrzucone w przestrzeni,

która notabene jest nieskończona (z ograniczeniem do pojemności liczb zmiennoprzecinkowych). W takim razie nie wiemy jaka podprzestrzeń odpowiada tym kodowaniom, które są sensownie dekodowane, co jest potrzebne przy generowaniu, a dodatkowo jeśli bałaby nieciągła, to niemożliwe byłoby interpolowanie pomiędzy próbkami. Znaczącym problemem jest natomiast to, że kodowanie jako wektor liczb jest bardzo precyzyjną informacją, co może doprowadzić do przeuczenia modelu i błędnego działania na danych pochodzących z poza zbioru treningowego.

1.5. Autoenkoder wariacyjny

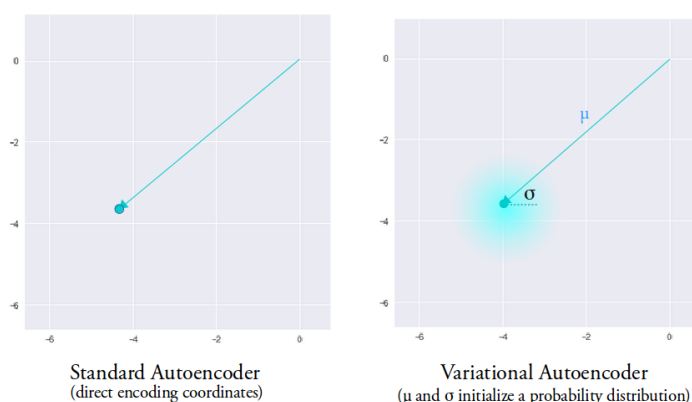
Autoenkoder wariacyjny (VAE, ang. Variational Autoencoder) rozszerza założenia podstawowej architektury o wprowadzenie modelowania rozkładu prawdopodobieństwa dla reprezentacji ukrytej. W odróżnieniu od podstawowej wersji enkoder nie produkuje pojedynczego wektora, ale dwa wektory interpretowane odpowiednio jako średnie μ oraz wariancję σ dla rozkładu Gaussa. Dopiero na podstawie tych parametrów formowane jest kodowanie, gdzie i -ta wartość losowana jest z $\mathcal{N}(\mu_i, \sigma_i^2)$, które aplikowane zostaje do dekodera i rekonstruowana jest na podstawie niego pierwotna próbka. Takie podejście znaczy, że nawet dla tego samego wejścia i podczas gdy średnie oraz odchylenia pozostaną niezmiennie, to i tak reprezentacja będzie się różnić właśnie ze względu na występującą losowość. Zmodyfikowany schemat został przedstawiony na rysunku 1.3.



Rysunek 1.3: Architektura autoenkodera wariacyjnego

Intuicyjnie wektor średnich oznacza miejsce, gdzie zakodowana zmienna powinna być wycentrowana, natomiast odchylenie kontroluje obszar, wokół którego kodowanie może się różnić. Sytuację tę zaprezentowałem na wykresie 1.4, przy czym porównałem z podstawową wersją. W jej przypadku informacja wychodząca z enkodera jest bardzo precyzyjna, natomiast w obecnie rozważanym modelu, dzięki dodaniu szumu, zostaje ona rozmyta. Taki zabieg ma za zadanie ograniczyć zjawisko

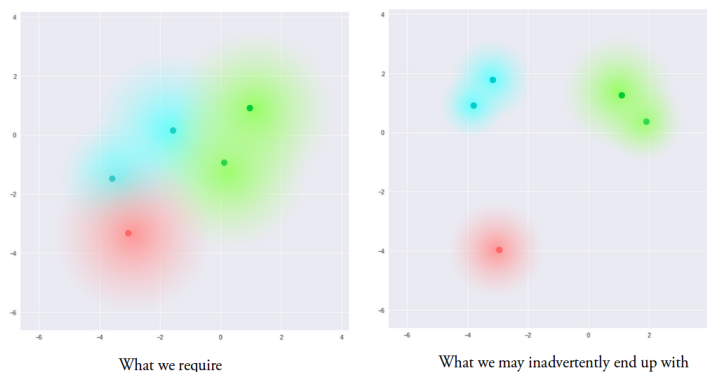
przeuczenia oraz wprowadzić pewną lokalną ciągłość w przestrzeni. Jest to wynikiem tego, że gdy kodowanie losowane jest z wnętrza kuli, notabene o nieskończonym promieniu, ale ustalonej gęstości, to dekodery zmuszony jest nauczyć się, że nie jedynie pojedynczy punkt odnosi się do danej próbki, ale i również jego całe lokalne sąsiedztwo. To pozwala dekodować nie tylko specyficzne reprezentacje (pozostawiając przestrzeń nieciągłą), ale też te trochę różniące się, ponieważ model narażony jest na szereg zmian kodowania tego samego wejścia podczas treningu.



Rysunek 1.4

Model jest teraz wystawiony na pewien stopień lokalnej zmienności przez wprowadzenie szumu do kodowania pojedynczej próbki, co skutkuje ciągłą przestrzenią w reprezentacji ukrytej dla lokalnego sąsiedztwa. Jednak na razie nie ma ograniczeń co do wartości jakie mogą być przyjmowane przez wektory μ i σ , a w rezultacie enkoder może nauczyć się generować bardzo różne μ dla różnych klas i grupować je minimalizując σ , upewniając się przy tym, że samo kodowanie nie różni się zbyt wiele dla tej samej próbki. W szczególności prawdopodobnym jest wystąpienie tak patologicznej sytuacji, gdzie σ zawsze wynosi 0, przez co nowy model nie będzie różnił się w działaniu od jego podstawowej wersji, włącznie z uwzględnieniem wad, które starał się wyeliminować. Omawiany przeze mnie przykład przedstawiony jest na rysunku 1.5. Oczekiwana sytuacja byłoby rozłożenie danych blisko siebie, ale z zachowaniem ich separowalności. Pozwoliłoby to na gładką interpolację pomiędzy próbkami i generowanie nowych danych.

W celu osiągnięcia zamierzonych rezultatów, należy wzbogacić funkcję straty o koszt dywergencji Kullbacka-Leiblera, która mierzy różnicę pomiędzy dwoma rozkładami prawdopodobieństwa. Minimalizowanie go oznacza optymalizowanie parametrów rozkładu (μ i σ), tak żeby jak najbardziej przypominał docelowy rozkład, który w przypadku autoenkodera wariacyjnego będzie standardowym rozkładem normalnym ($\mu = 0$, $\sigma = 1$). Przy takim wyborze dane powinny zostać rozrzucone dookoła



Rysunek 1.5

$\vec{0}$ i nie pozostawiać pustych miejsc w przestrzeni ukrytej.

$$\text{KLD} = \sum_{i=1}^n \sigma_i^2 + \mu_i^2 - \log(\sigma_i) - 1 \quad (1.2)$$

1.5.1. Opis matematyczny

To co do tej pory napisałem odnośnie autoenkodera wariacyjnego raczej trzeba uznać za zbiór intuicji i oczekiwań. W tym rozdziale chciałbym pokazać jego funkcjonowanie od strony matematycznej.

Dokończyć

1.6. Ocena jakości modelu

1.6.1. Krzywa ROC i AUC

Krzywa ROC (Receiver Operating Characteristic) jest narzędziem do oceny poprawności klasyfikatora binarnego. Bazuje ona na wyliczaniu charakterystyki jakościowej modelu predykcji w wielu różnych punktach odcięcia. Działa to na takiej zasadzie, że model dla próbek przewiduje z jaką pewnością pochodzą z klasy 1. Następnie badane są różne progi i klasyfikowane przy ich użyciu obiekty. Dla uzyskanych klasyfikacji liczymy odsetek prawdziwych pozytywnych (TPR) oraz odsetek fałszywych pozytywnych (FPR) i nanosimy te wartości na wykres. Warto zauważyć, że dla losowego modelu jego wykres to prosta przechodząca przez $(0, 0)$ i $(1, 1)$. Dzieje się tak, ponieważ dla każdego progu połowa klasyfikacji będzie nad i połowa pod. Idealny model znajduje się w punkcie $(0, 1)$.

Przydatne jest również obliczenie pola powierzchni pod krzywą AUROC (ang. Area Under the ROC). Interpretuje się ją jako prawdopodobieństwo, że badany mo-

del predykcijny oceni wyżej losowy element klasy pozytywnej od losowego elementu klasy negatywnej.

Rozdział 2.

Eksperyment na danych syntetycznych (MNIST)

Do przetestowania pomysłu posłuży mi zbiór MNIST. Jest to zestaw odręcznie napisanych cyfr, które następnie zostały znormalizowane względem rozmiaru i wycentrowane. Każdy obrazek ma rozmiar 28x28 pikseli, mających wartości całkowite z przedziału $[0, 255]$. Na zbiór składa się 60 tys. danych treningowych i 10 tys. testowych. Przykładowe obrazki znajdują się na rysunku 2.1.

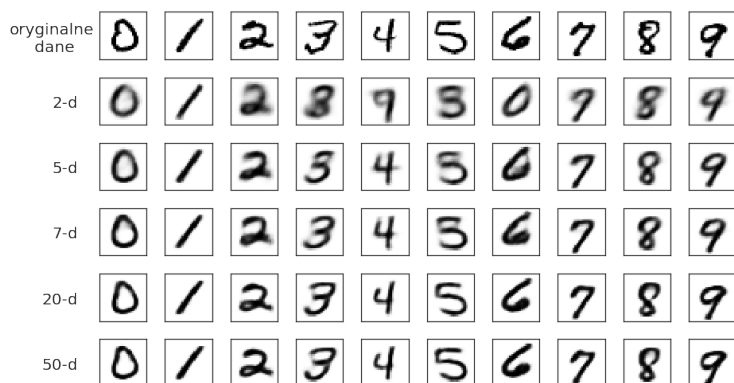


Rysunek 2.1: Przykładowe dane ze zbioru MNIST

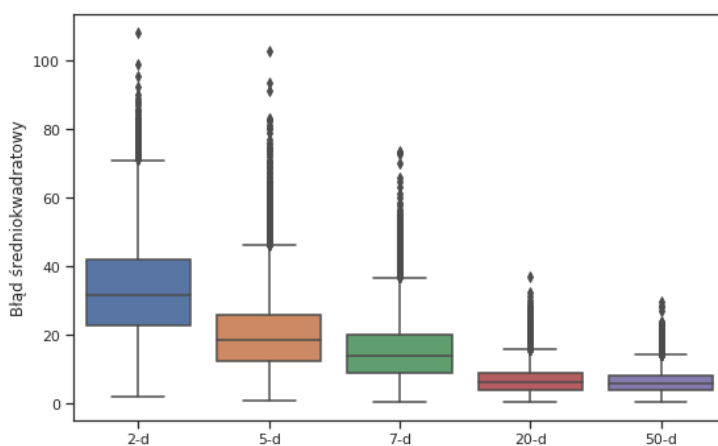
2.1. Możliwości autoenkodera wariacyjnego

Chciałbym zacząć od sprawdzenia podstawowych własności autoenkodera wariacyjnego, przedstawionych w rozdziale 1.5. W tym celu wytrenowałem modele o różnych rozmiarach reprezentacji ukrytej: 2, 5, 7, 20, 50. Następnie sprawdziłem jak

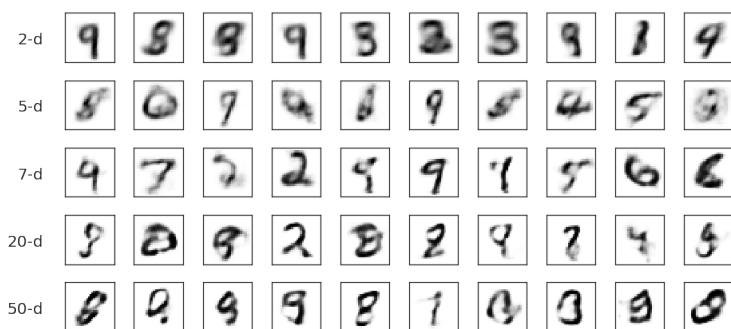
radzą sobie z rekonstrukcją danych ze zbioru testowego oraz generowaniem nowych próbek. Wyniki prezentują się na rysunku 2.2.



(a)



(b)

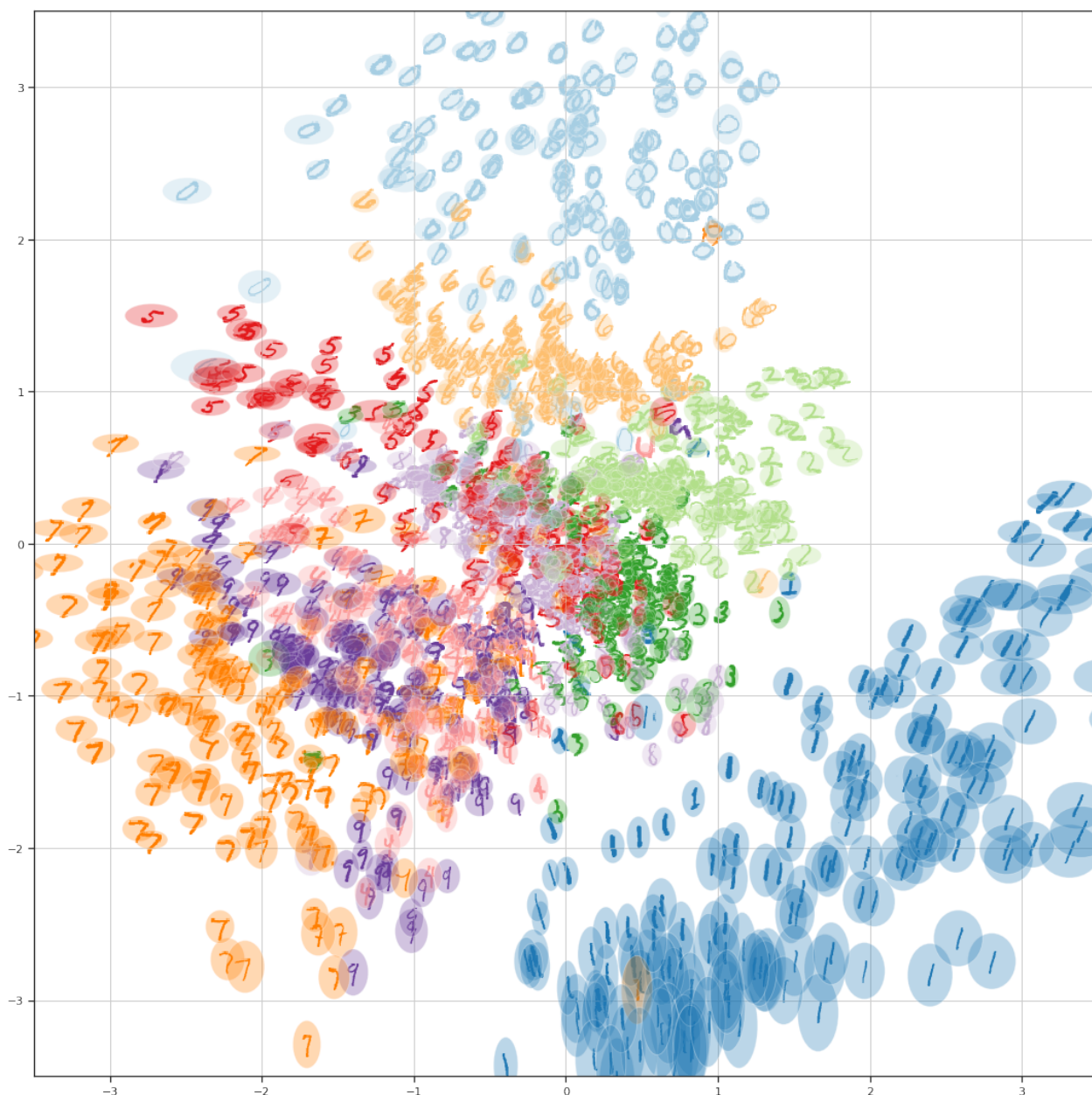


(c)

Rysunek 2.2

Ze względu na łatwość wizualizacji, do sprawdzenia jak dane reprezentowane są w przestrzeni ukrytej wybrałem model bazujący na wymiarze rozmiaru 2. Zdaję sobie sprawę, że rekonstrukcje w tym przypadku nie są najlepszej jakości, ze względu na zbyt ograniczoną ilość informacji możliwych do przekazania, ale zależy mi też na przedstawieniu podstawowych założeń. W tym celu na wykresie dla losowych

próbek zaznaczyłem odpowiadające im średnie oraz odchylenia. Otrzymany rezultat widać na obrazku 2.3, gdzie można zauważyć opisane wcześniej własności wybranego modelu takie jak skoncentrowanie reprezentacji wokół $\vec{0}$ czy ciągłość przestrzeni.



Rysunek 2.3

2.2. Symulacja docelowego problemu

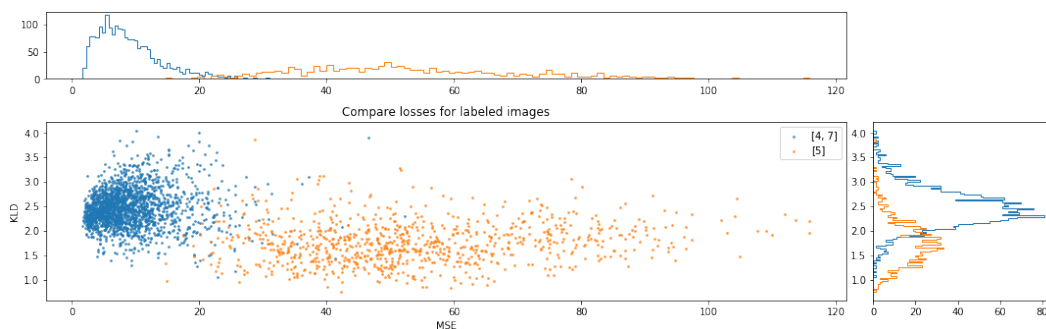
W tym miejscu interesuje mnie odtworzenie oryginalnego problemu na prostszych danych, jakim jest właśnie zbiór MNIST. Pozwoli mi to stwierdzić sensowność skorzystania z wybranego modelu. Zakładam, że jeżeli eksperyment nie powiódłby

się na łatwiejszych danych, to raczej nie powiedzie się również na tych skomplikowanych. Dodatkowo zyskam pewnego rodzaju doświadczenie z wybraną architekturą.

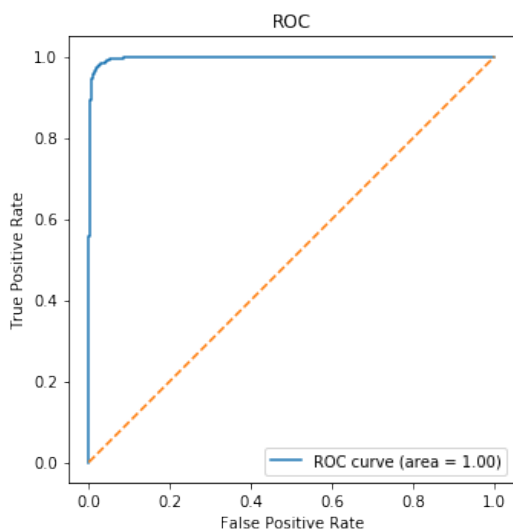
Przygotowana przeze mnie symulacja prezentuje się następująco. Zbiór danych składa się z trzech cyfr: 4, 5, 7, z czego 5 stanowi jedynie 1% wszystkich próbek. Ma to odpowiadać rzadkości sytuacji patologicznych. Natomiast zadaniem jest stwierdzenie czy dana obserwacja jest cyfrą 5 czy nie.

Klasyfikacja odbywać się będzie na podstawie wartości funkcji straty dla autoenkodera wariacyjnego, którą można interpretować jako ograniczenie prawdopodobieństwa. Takie podejście wymaga ustalenia pewnego progu kategoryzacji, dlatego w tym miejscu będę prezentował krzywą ROC, która rozważa wszystkie sensowne wartości. Dodatkowo zaprezentuję dwa składniki błędu oddzielnie, przez co będzie można zaobserwować, który z nich ma największy wpływ.

Na podstawie wniosków z rozdziału 2.1. wybrałem rozmiar warstwy ukrytej równy 10. Wyniki zaprezentowane są na rysunku 2.4 oraz 2.5.



Rysunek 2.4



Rysunek 2.5

2.3. Wnioski

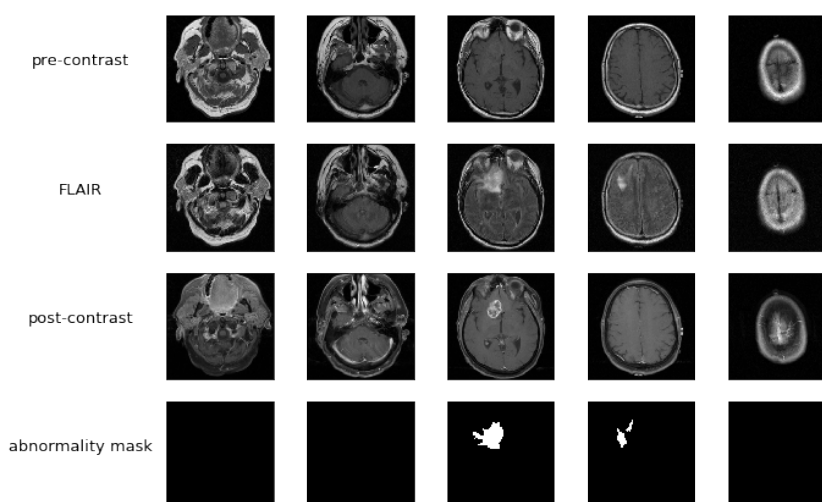
Na podstawie otrzymanych wyników mogę stwierdzić, że eksperyment się powiódł. W przypadku danych MNIST udało się sklasyfikować dane, które są rzadkie i stanowią jedynie odsetek wszystkich. Jest sens w takim razie w użyciu tego modelu dla już prawdziwych danych.

Rozdział 3.

Eksperyment na danych właściwych (MRI FLAIR)

3.1. Opis danych

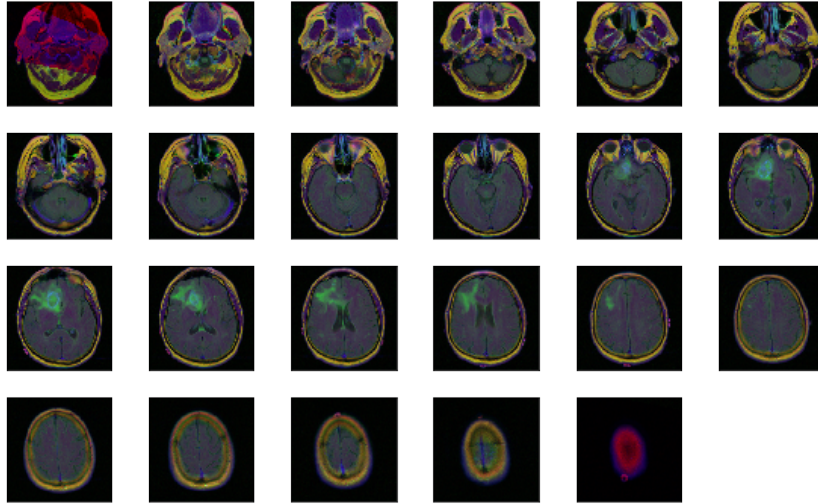
Dane pochodzą z Uniwersytetu Duke’a (ang. Duke University). Są to zdjęcia głowy wykonane metodą obrazowania rezonansu magnetycznego w technice FLAIR (ang. fluid-attenuated inversion recovery) wraz z zaznaczonymi obszarami zmian nowotworowych. Obrazy mają rozmiar 256x256x3 pikseli, gdzie pierwszy kanał odpowiada momentowi przed wprowadzeniem kontrastu, trzeci po, a drugi jest właściwym zdjęciem. Maski mają rozmiar 256x256 pikseli o wartościach odpowiednio 255 dla komórek nowotworowych i 0 w przeciwnym wypadku. Na rysunku 3.1 pokazane są zdjęcia z podziałem na kanały i odpowiadające im maski.



Rysunek 3.1

Dane pogrupowane są dla 110 pacjentów ze zdiagnozowanym nowotworem. Na

rysunku 3.2 zaprezentowałem zdjęcia pojedynczej osoby w trzech kanałach. Łącznie w zbiorze danych znajduje się 3929 par obrazów, przy czym jedynie $\sim 1.02988\%$ pikseli zostało zidentyfikowanych jako komórki nowotworowe, co potwierdza obserwację odnośnie ich rzadkości.



Rysunek 3.2

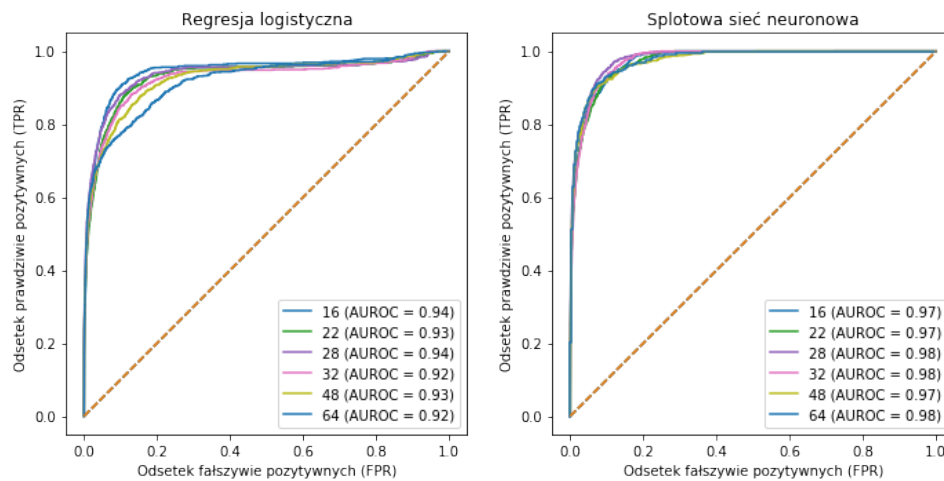
3.2. Wstępna obróbka

Dane podzieliłem na 2 rozłączne zbiory ze względu na pacjentów w stosunku 70% dla zestawu treningowego i 30% dla testowego.

3.2.1. Podział na mniejsze kawałki

Danymi, które rzeczywiście mnie interesują są poszczególne komórki w mózgu. To co chciałbym umieć oceniać to ich patologiczność. Mam zamiar to robić na podstawie kontekstu w jakim się znajdują, czyli lokalnego sąsiedztwa na zdjęciu. W praktyce sprowadzi się to do utożsamiania wycinka obrazu rozmiaru $n \times n$ z wartością komórki leżącą w jego środku. W takim sensie chce podzielić zbiór danych na mniejsze kawałki, żeby każdej komórce odpowiadał jej kontekst. Jednak parametrem wymagającym ustalenia jest rozmiar takiego sąsiedztwa. Muszę wiedzieć czy kawałek o danych wymiarach będzie zawierał wystarczającą ilość informacji potrzebnych do poprawnej klasyfikacji. Teoretycznie za kontekst mógłbym uznać cały obrazek, ale zależy mi też na ograniczeniu rozmiaru modelu i potrzebnej w związku z tym mocy obliczeniowej. Do wyznaczenia tej wartości skorzystam z podejścia nauki nadzorowanej, co może wydawać się sprzeczne z moją początkową deklaracją, ale ten krok służy jedynie we wsparciu wyboru optymalnego rozmiaru i nie jest wymagany.

Decyzję podejmę na podstawie rezultatów osiąganych przez modele takie jak regresja liniowa oraz splotowa sieć neuronowa dla następujących rozmiarów: 16, 22, 28, 32, 48, 64. Zakładam, że jeśli w jakimś przypadku dokładność będzie zadowalająca, to ilość obecnych informacji wystarczy do rekonstrukcji. Warto w tym miejscu wspomnieć, że przy takim podejściu muszę zrównoważyć dane w zbiorze treningowym, ponieważ aktualnie jednych jest około 100 razy mniej niż drugich. Rozwiążę to na takiej zasadzie, że w trakcie uczenia porcję danych wejściowych będę komponował losując połowę próbek z jednej klasy i połowę z drugiej. Może to wpłynąć negatywnie na obciążenie modelu, ale i tak ostateczna klasyfikacja będzie odbywać się z wykorzystaniem progu, co powinno zredukować ten błąd. Wyniki zaprezentowałem przy pomocy krzywych ROC na wykresie 3.3. Jak można zauważyć wszystkie rozważane rozmiary prezentują się równie dobrze, w związku z czym zdecydowałem się wybrać jako parametr liczbę 22.



Rysunek 3.3

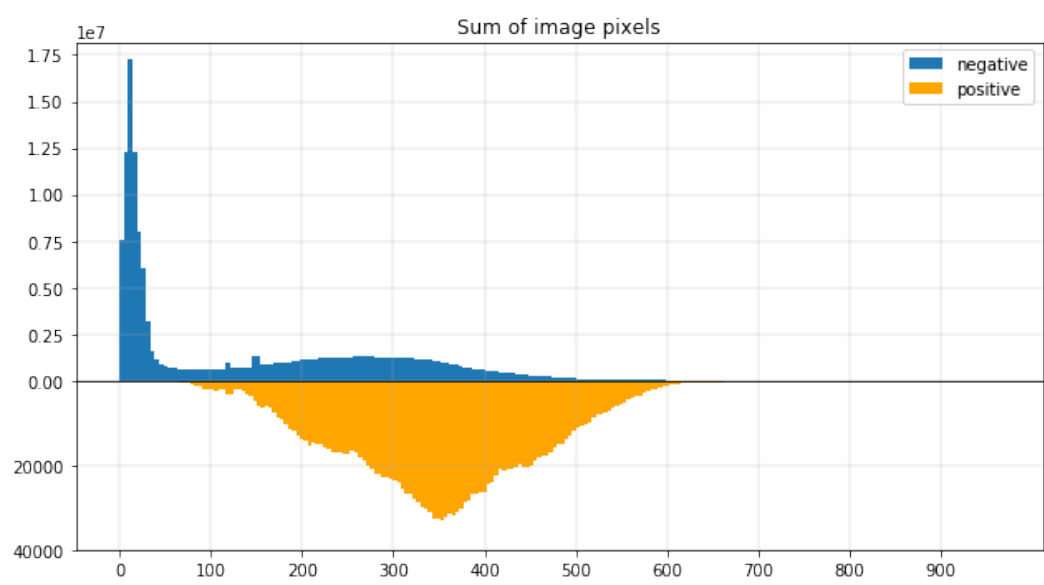
Dodatkowo chciałem sprawdzić jak taki prosty model rzeczywiście poradzi sobie w przypadku problemu identyfikowania komórek nowotworowych u pacjenta. Nie jest to oczywiście najlepsza metoda dla podejścia nadzorowanego, ponieważ jedną z nich jest architektura U-Net, ale powinno to wystarczyć jako punkt odniesienia dla autoenkodera.

Jako model rozważań wybrałem sieć splotową, a próg kategoryzacji jako najlepszy punkt odcięcia na krzywej ROC ograniczający odsetek wyników fałszywie pozytywnych do 1%.

3.2.2. Dodatkowe przygotowanie zbioru

Po zdecydowaniu się na łatki rozmiaru 22 i przygotowaniu takiego zbioru postanowiłem dodatkowo przeanalizować jego zawartość w celu znalezienia trywialnych przypadków. Jednym z nich były kawałki znajdujące się poza czaszką, które nigdy

nie są nowotworowe. Będą to obrazki o niskiej łącznej sumie pikseli, co zaprezentowane jest na histogramie 3.4 (zwracam uwagę na różne skale na osiach). Na początku można zauważyć sporą grupę przypadków niepatologicznych. Sprawdziłem, że minimalna wartość sumy dla klasy odpowiadającej nowotworom wynosi 56.5. Przy tym założeniu usunąłem wszystkie obrazki u sumie mniejszej, niż 42.0 co zmniejszyło ilość danych o 45.5%, pozbywając się trywialnych próbek. Dany próg będę oczywiście uwzględniał w późniejszej klasyfikacji.



Rysunek 3.4

3.3. Wyniki

Przetestuję 4 kombinacje modeli, ze zmienioną funkcją kosztu

W tabeli 3.1 zostały przedstawione wyniki dla czterech kombinacji modeli oraz parametrów. Wartości liczbowe opisują najlepsze AUC ROC podczas uczenia wraz z odpowiadającą epoką. Jak widać w tym zestawieniu najlepiej wypadły podstawowe modele VAE.

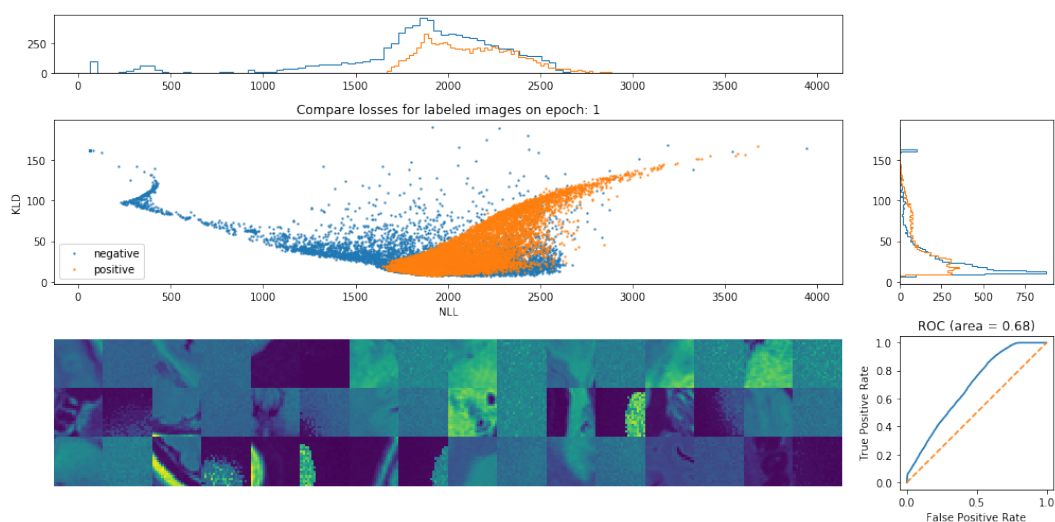
Model	Best ROC AUC
VAE	0.68 (1)
VAE + Softmax	0.68 (1)
C-VAE	0.59 (1)
C-VAE + Softmax	0.67 (1)

Tablica 3.1: My table

3.4. Analiza

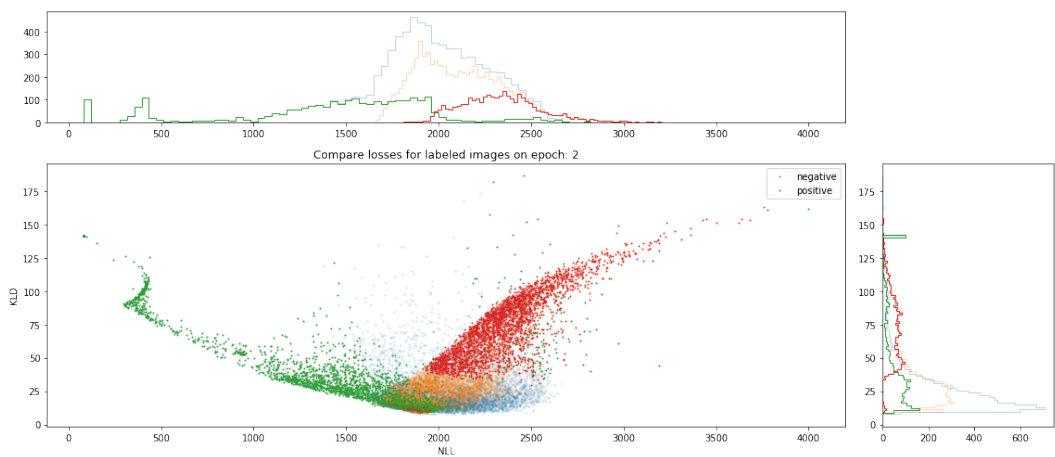
Zajmę się analizą modelu w wersji VAE + Softmax. Na wykresie 3.5 przedstawione są koszty, ich rozkłady, przykłady rekonstrukcji oraz krzywa ROC. Ciekawie wyglądają te dwa zauważalne ogony na wykresie. Postaram się lepiej przyjrzeć ich specyfice.

Pierwszym pomysłem jest, żeby sprawdzić rozkład obrazków w zależności od ich jasności, czyli łącznej sumy pikseli. Ciemne obrazki definiuję jako te o niskiej sumie, a jasne o wysokiej. Na rysunku 3.6 zaznaczyłem negatywne obrazki, dla których suma $\mu = 50$ oraz pozytywne obrazki z sumą co najmniej 150. Zaznaczone są one odpowiednio zielonym i czerwonym kolorem na wykresie. Jak widać po rozkładach składają się one na dokładnie te 2 separowalne ogony. Wniosek z tego jest następujący. Na podstawie tego modelu możemy odesparować jedynie najciemniejsze i najjaśniejsze obrazki z obydwu klas. W takim razie model nie robi nic nadzwyczajnego. Wystarczy przyjrzeć się histogramowi na rysunku 3.7 z zaznaczonymi progami. One już bardzo dobrze separują te klasy.

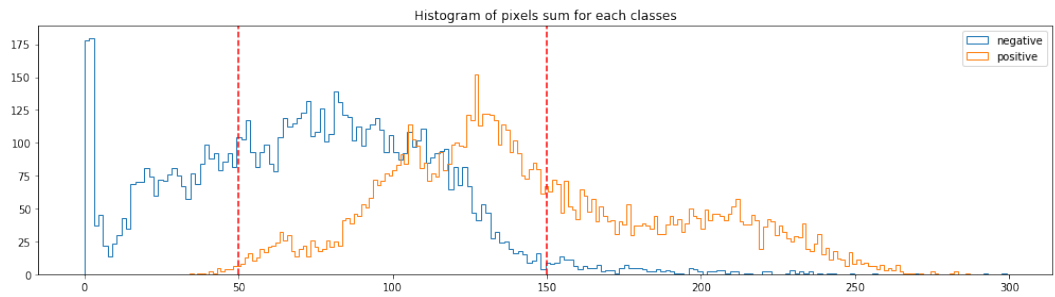


Rysunek 3.5

3.4.1. Wnioski



Rysunek 3.6



Rysunek 3.7

Rozdział 4.

Summary