Uniwersytet Warszawski
Wydział Matematyki, Informatyki i Mechaniki

Jakub Giezgała
Nr albumu: 448134

Analiza właściwości sieci neuronowych do predykcji aktywności obszarów regulatorowych

Praca licencjacka
na kierunku BIOINFORMATYKA I BIOLOGIA SYSTEMÓW

Praca wykonana pod kierunkiem
dr. hab. Bartosza Wilczyńskiego
Instytut Informatyki

Warszawa, Czerwiec 2025

Streszczenie

W pracy przeprowadzono analizę właściwości konwolucyjnej sieci neuronowej (CNN) wykorzystywanej do klasyfikacji aktywności obszarów regulatorowych DNA, takich jak promotory i enhancery, na podstawie sekwencji pochodzących z komórek glejowych. Model wstępnie wytrenowano na zbiorze danych pozyskanych z glejaków, a następnie poddano transferowemu uczeniu na mniejszym, docelowym zbiorze pochodzącym z indukowanych pluripotentnych komórek macierzystych sztucznie zróżnicowanych do astrocytów (iAstrocytes).

Głównym celem badania była ocena wpływu dotrenowywania na zdolność predykcyjną względem czterech klas funkcjonalnych (aktywny promotor, aktywny niepromotor, nieaktywny promotor, nieaktywny niepromotor) oraz identyfikacja mechanizmów adaptacji modelu. Dostrajanie modelu spowodowało niewielkie pogorszenie wydajności na danych źródłowych przy jednoczesnej istotnej poprawie metryk na zbiorze docelowym, co wskazuje na skuteczne dostosowanie reprezentacji sieci do specyfiki sekwencji astrocytarnych.

W celu wyjaśnienia mechanizmów adaptacji szczegółowej analizie poddano 300 filtrów pierwszej warstwy konwolucyjnej: porównano zmiany wag i normy Frobeniusa przed i po dotrenowaniu oraz oceniono wpływ poszczególnych filtrów na predykcję. Aby przetestować plastyczność warstwy wejściowej, pierwotnie wyzerowane filtry reaktywowano przez losowanie wag z rozkładu oszacowanego metodą jądrowej estymacji gęstości (KDE), a wyniki weryfikowano wielokrotnymi testami statystycznymi. Analiza wykazała, że adaptacja sieci do nowego zbioru danych zachodzi głównie przez selektywną modyfikację niewielkiego podzbioru filtrów, podczas gdy wagi większości z nich są powtórnie zerowane. Wyniki sugerują, że efektywny transfer wiedzy osiągany jest poprzez ukierunkowaną adaptację niewielu filtrów, co może mieć implikacje dla transferowego uczenia oraz interpretowalności modeli w genomice.

Słowa kluczowe

Konwolucyjne sieci neuronowe, Analiza filtrów konwolucyjnych, Cecha strukturalna DNA, Predykcja aktywności epigenetycznej, Norma Frobeniusa, Estymacja jądrowa gęstości

Dziedzina pracy (kody wg programu Socrates-Erasmus)

11.9 Inne nauki matematyczne i informatyczne

Klasyfikacja tematyczna

J. Computer Applications
J.3. Life and Medical Sciences

Tytuł pracy w języku angielskim

Analysis of properties of neural networks for prediction of regulatory sequence activity

Spis treści

1. Wprowadzenie . . . 5
1.1. Regulacja ekspresji genów . . . 5
1.1.1. Promotory i enhancery . . . 7
1.1.2. Modyfikacje histonowe . . . 8
1.1.3. Epigenetyka nowotworów mózgu . . . 8
1.1.4. Klasyfikacja obszarów regulatorowych . . . 9
1.2. Uczenie maszynowe . . . 10
1.2.1. Uczenie nadzorowane . . . 11
1.2.2. Spadek gradientowy . . . 12
1.2.3. Miary jakości . . . 13
1.3. Głębokie uczenie . . . 15
1.3.1. Konwolucyjne sieci neuronowe . . . 18
1.3.2. Uczenie transferowe . . . 20
1.4. Architektura *Custom* . . . 20
1.5. Analiza statystyczna danych . . . 24
1.5.1. Estymacja jądrowej gęstości . . . 25

2. Motywacja . . . 27
2.1. Cel pracy . . . 27
2.2. Znaczenie pierwszej warstwy konwolucyjnej . . . 28
2.3. Pytania badawcze . . . 28

3. Wyniki . . . 29
3.1. Porównanie modeli . . . 29
3.2. Dotrenowywanie modelu *alt* . . . 32
3.2.1. Optymalizacja parametrów uczenia . . . 32
3.2.2. Strategia eksperymentalna . . . 33
3.3. Zamiana pozycją istotnych z wyzerowanymi filtrami . . . 36
3.4. Probabilistyczna aktywacja filtrów . . . 37
3.5. Redukcja normy poprzez estymację średnich wartości . . . 40
3.6. Wpływ pierwszej warstwy konwolucyjnej . . . 42

4. Podsumowanie i dyskusja . . . 47

Bibliografia . . . 51
Aneksy . . . 56
A. Wykresy . . . 56
B. Implementacja . . . 60

Rozdział 1

Wprowadzenie

W niniejszej pracy przeprowadzono analizę filtrów pierwszej warstwy sieci konwolucyjnej, ze szczególnym uwzględnieniem wpływu ich normy Frobeniusa na jakość klasyfikacji. W pierwszym rozdziale omówiono kluczowe pojęcia, zarówno z biologii molekularnej, jak i szeroko pojętego uczenia maszynowego, aby przybliżyć problem czytelnikom niezwiązanym bezpośrednio z dziedziną. Punktem wyjścia naszych rozważań jest problem klasyfikacji surowych sekwencji nukleotydowych do czterech klas obszarów regulatorowych. [1].

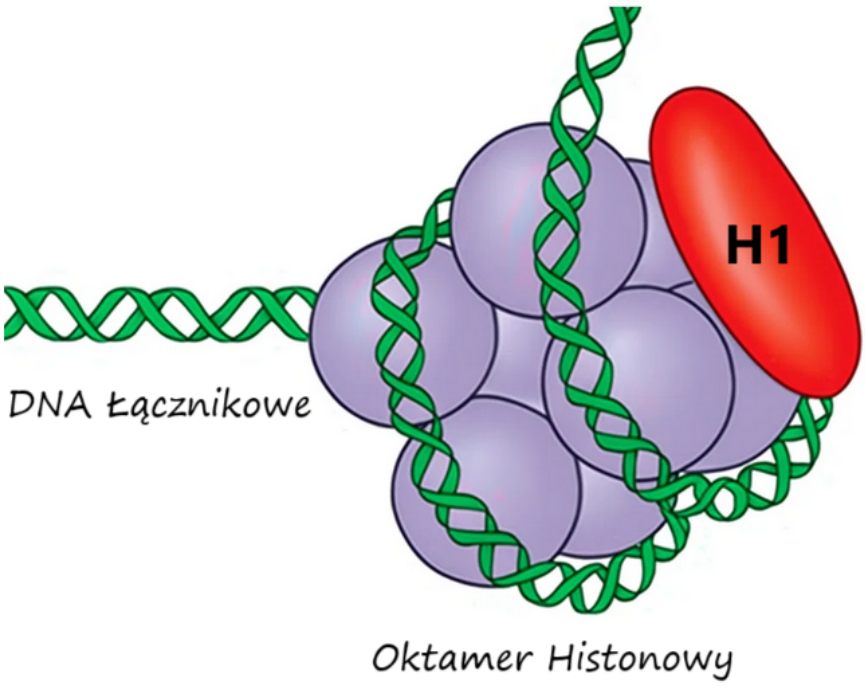
1.1. Regulacja ekspresji genów

Regulacja ekspresji genów zapewnia precyzyjne sterowanie syntezą białek i jest niezbędna dla prawidłowego funkcjonowania organizmów. Mimo że wszystkie komórki somatyczne organizmu posiadają ten sam materiał genetyczny, to właśnie precyzyjny wybór aktywnych genów w danej komórce jest przyczyną ich różnorodności. **Geny**, czyli pojedyncze fragmenty DNA o długości od kilkuset do kilku tysięcy par zasad, zawierają informacje potrzebne do syntezy białek oraz niekodujących RNA. Dzięki regulacji ekspresji komórki mogą adaptować się do zmiennych warunków środowiska i utrzymywać homeostazę. Proces ten nie ogranicza się wyłącznie do eukariontów, ale jest kluczowy także dla prokariotów oraz wirusów.

Helisa DNA, pełniąca funkcje podstawowego nośnika informacji genetycznej, składa się z dwóch komplementarnych nici polinukleotydowych. Połączone są ze sobą przez wiązania wodorowe między parami zasad azotowych: adenina (A) zawsze łączy się z tyminą (T), a guanina (G) z cytozyną (C), których kolejność determinuje kod genetyczny. W obrębie genomu wyróżniamy regiony kodujące (eksony) oraz niekodujące (introny). Dopiero procesy **transkrypcja** (przepisywanie DNA na RNA) i **translacja** (synteza białek) pozwalają fragmentom kodującym pełnić swoje funkcje biologiczne. Natomiast regiony niekodujące precyzyjnie regulują obydwa etapy ekspresji genów.

W komórkach eukariotycznych DNA występuje głównie w formie liniowej, upakowanej w chromosomy. Ich głównym składnikiem, występującym także w cytoplaźmie niektórych komórek prokariotycznych, jest **chromatyna** - włóknista substancja składająca się nie tylko z DNA, ale także z białek, niewielkiej ilości RNA oraz **histonów**. To właśnie zasadowość tych ostatnich jest przyczyną nawijania się wokół nich niespełna dwa razy DNA, tworząc wspólnie z nim **nukleosom**. Około 200 par zasad (dokładnie 146 w przypadku człowieka) owiniętych jest wokół 8 histonów rdzeniowych, między którymi znajduje się *circa* 50 par zasad, tworzących DNA łącznikowe.

Ze względu na pełnioną funkcję biologiczną oraz budowę chemiczną, można wyróżnić dwa stany chromatyny: **euchromatynę** oraz **heterochromatynę**. Euchromatyna jest aktywna transkrypcyjnie, w przeciwieństwie do nieaktywnej heterochromatyny. Oznacza to, że heterochromatyna nie ulega procesowi transkrypcji, czyli przepisaniu DNA na RNA i translacji, która odpowiada za przetłumaczenie kodu genetycznego na sekwencję aminokwasową [17]. Do dnia dzisiejszego jej funkcja nie jest w pełni poznana, jednak obecnie przyjmuje się, że wpływa na regulację ekspresji genów oraz stabilność chromosomów. Nie ogranicza się tylko do wyciszania genów, ale bierze aktywny udział w ich aktywacji [18]. Euchromatyna natomiast, która jest aktywna transkrypcyjnie, zawiera obszerną liczbę genów.



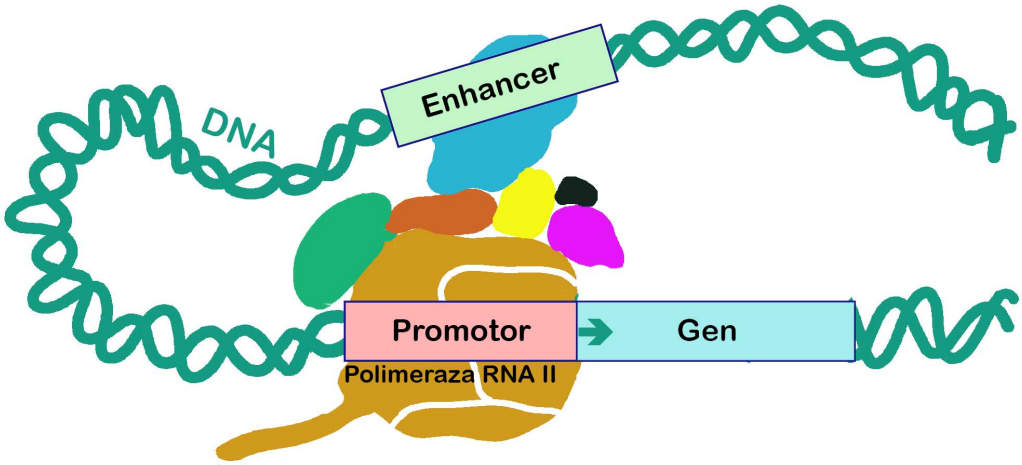
Rysunek 1.1: Helisa dwuniciowego DNA owinięta wokół oktameru histonowego tworzy nukleosom, który jest podstawowym poziomem upakowania chromatyny. Pojedyncza cząstka rdzeniowa składa się z osmiu białek histonowych (po dwa histony H2A, H2B, H3, H4). Stabilizację całego nukleosomu, zapewnia histon H1, które łączy je w zwartą strukturę. Razem z nim tworza strukturę nazywaną chromatosomem. Sąsiednie cząstki rdzenia nukleosomu połączone są bezpośrednio przez DNA łącznikowe.

Zgodnie z centralnym dogmatem biologii molekularnej informacja genetyczna przepływa z DNA przez RNA do białka. Na każdym z tych etapów dochodzi do regulacji ekspresji genów. Ważne są zarówno stopień upakowania chromatyny, jak i dostępność specyficznych czynników transkrypcyjnych. Inicjację transkrypcji warunkują promotory, enhancery oraz kompleksy mediatorowe. Natomiast modyfikacje chromatyny, takie jak metylacja czy acetylacja, hamują albo ułatwiają wiązanie czynników transkrypcyjnych. Na poziomie syntezy białek zachodzą mechanizmy, takie jak alternatywny splicing, czy poliadenylacja, kontrolowane przez niekodujące RNA. Wpływają one na prawidłowe przetworzenie matrycowego RNA przed translacją. Sama inicjacja translacji wymaga obecności małych RNA regulatorowych i odpowiednich modyfikacji białek inicjacyjnych.

1.1.1. Promotory i enhancery

Promotor to sekwencja DNA znajdująca się bezpośrednio przed początkiem transkrypcji genu. Stanowi niezbędny element ekspresji, rekrutując polimerazę RNA i czynniki transkrypcyjne oraz wskazując punkt startu i kierunek transkrypcji [19]. Promotory rozpoznawane są przez charakterystyczne motywy sekwencyjne, takie jak wyspy CpG czy silnie konserwowaną sekwencję konsensusową TATAAA (ang. *TATA box*), które umożliwiają lokalizację genu w genomie.

Enhancer, inaczej wzmacniacz transkrypcji, to zazwyczaj krótki fragment DNA, zlokalizowany w pewnej odległości od promotora regulowanego genu. Pomimo dużej odległości (nawet setki tysięcy par zasad) enhancery mogą zwiększać wydajność transkrypcji danego genu. Enhancery, w przeciwieństwie do wyciszaczy (ang. *silencers*), powodują zwiększenie wydajności, poprzez wzrost prawdopodobieństwa rekrutacji innych czynników transkrypcyjnych oraz enzymów modyfikujących chromatynę. Chociaż mogą się one znajdować nawet w znacznej odległości, to aby być aktywne, muszą być zlokalizowane w bezpośrednim sąsiedztwie promotora. Zbliżają się do niego fizycznie poprzez wytworzenie pętli chromatyny. Występują zarówno u eukariontów, jak i u prokariontów.



Rysunek 1.2: Mechanizm aktywacji transkrypcji poprzez fizyczną interakcję enhancera i promotora. Polimeraza RNA wiąże się z promotorem, aby zainicjować transkrypcję. Wiązanie enhancera z promotorem opiera się na polimerazie RNA II (RNAPII) oraz złożonej strukturze specyficznych dla danego typu komórek czynników transkrypcyjnych, koregulatorów, enzymów, białek architektonicznych, takich jak kohezyna, kondensyna i czynnik wiążący CTCF [17]. Spowodowane jest to faktem, że polimerazy jądrowe RNA u eukariontów nie rozpoznają sekwencji promotora, a jedynie kompleks preinicjacyjny, który jest niezbędny do prawidłowego przebiegu transkrypcji.

1.1.2. Modyfikacje histonowe

Każda komórka danego organizmu współdzieli ze sobą identyczne DNA. Nie oznacza to jednak, że w każdej z nich ekspresji ulegają wszystkie geny. Aby utrzymywać stan homeostazy, w każdej z nich odpowiednie białka muszą być syntetyzowane w odpowiednim momencie. Analogicznie do genów, promotory i enhancery mogą być aktywne, bądź nie. Miarą ich aktywności jest mierzenie poziomu danego genu w komórce. Jeśli jest on podwyższony, to mówimy, że zarówno promotor, jak i enhancer są aktywne. Zależy to od położenia fragmentów DNA ich kodujących w jądrze komórkowym, względem położenia kompleksów histonowych.

Zmiany posttranslacyjne histonów mogą zarówno aktywować, jak i represjonować transkrypcję. Przykładowo trimetylacja histonu H3 w pozycjach K4, K36 czy K79 oraz acetylacja histonów H3 i H4 zwykle korelują z wysoką aktywnością transkrypcyjną. Modyfikacje rozluźniają struktury chromatyny i ułatwiają dostęp czynnikom transkrypcyjnym [9]. Z kolei trimetylacja H3K9 i H3K27 prowadzi do kondensacji chromatyny i wyciszenia genów. Istotnym markerem aktywnych promotorów i enhancerów jest też acetylacja lizyny 27 histonu H3 (H3K27ac).

Dodatkowo w regionach kodujących enhancery obserwuje się trimetylację H3K4me1, bez dodania trzech grup metylowych do czwartej lizyny histonu H3 [20, 21]. Podsumowując, modyfikacje histonowe tworzą kombinatoryczną 'gramatykę' umożliwiającą precyzyjną regulację transkrypcji na poziomie chromatyny.

1.1.3. Epigenetyka nowotworów mózgu

Epigenetyka jest nauką zajmującą się badaniem zmian w ekspresji genów, które nie są powiązane ze zmianami w sekwencji DNA. Jej rozwój pozwolił na skorelowanie modyfikacji histonowych z procesem nowotworzenia. Procesem polegającym na niekontrolowanym podziale komórek, które przestają podlegać naturalnym mechanizmom kontroli wzrostu. Zakłócenie wzorców metylacji genomowego DNA było pierwszą nieprawidłowością epigenetyczną opisaną w ludzkich nowotworach. Ta nierównowaga obejmuje hipermetylację wysp CpG promotorów genów supresorowych, powodującą zaburzenie transkrypcji [22].

Glejaki to najczęstsze nowotwory ośrodkowego układu nerwowego. Światowa Organizacja Zdrowia (WHO) wyróżnia cztery kategorie w zależności od zjadliwości choroby. Obejmują one łagodne gwiaździaki (stopień I wg WHO), gwiaździaki rozlane, które mogą być łagodne (stopień II) lub złośliwe (stopień III), aż po wysoce złośliwe glejaki wielopostaciowe (stopień IV). Zaliczają się do nich także złośliwe astrocytomy, znane również jako gwiaździaki złośliwe, czyli nowotwory mózgu wywodzące się z **astrocytów** - komórek glejowych pełniących istotną i wszechstronną rolę w ośrodkowym układzie nerwowym [10]. Są to zazwyczaj guzy o wysokim stopniu złośliwości (III lub IV stopień) i agresywnym przebiegu [24].

W kontekście nowotworów mózgu astrocyty pełnią dwojaką rolę. Po pierwsze, komórkowe populacje wywodzące się z linii astrocytarnej (lub ich prekursorów) stanowią potwierdzone źródło transformacji prowadzącej do rozwoju nowotworu. Typ rozwijającego się guza zależy zarówno od pierwotnego stanu komórkowego, jak i od towarzyszących mutacji (np. w genie ATRX) oraz zmian epigenetycznych. [23, 25]. Po drugie, modyfikacje struktury chromatyny oraz profile aktywności promotorów i enhancerów silnie kształtują programy ekspresji genów, co odróżnia guzy o niskiej złośliwości od tych wysoce agresywnych [9]. Przykładowo metylacja promotora genu MGMT prowadzi do jego wyciszenia i wiąże się z lepszą odpowiedzią pacjentów na leczenie temozolomidem [23].

W jednym z badań [9] opracowano atlas epigenetyczny różnych stopni glejaków, uwzględniając stopień upakowania chromatyny, modyfikacje histonowe (m.in. H3K27ac, H3K27me3) i metylację DNA. Sekwencje uzyskano za pomocą metody **ChiP-seq** (ang. *chromatin immunoprecipitation sequencing*), która pozwala na identyfikację regionów wzbogaconych w metylacje lub acetylacje. Wykazano, że organizacja chromatyny i układ aktywnych enhancerów mają kluczowe znaczenie dla specyficznej ekspresji genów w guzach o różnym stopniu agresywności. Profil epigenetyczny i trójwymiarowa architektura chromatyny są zatem niezbędne do zrozumienia patogenezy glejaków [9].

1.1.4. Klasyfikacja obszarów regulatorowych

Klasyfikacja i wykrywanie elementów cis-regulatorowych DNA, czyli tych leżących w pobliżu genu, które regulują (m.in. promotory i enhancery), w ciągu ostatnich lat przeżyła prawdziwą rewolucję. Pierwotne metody opierały się na statystyczno-fizycznych modelach służących identyfikacji krótkich motywów DNA rozpoznawanych przez czynniki transkrypcyjne [26]. Równolegle rozwinęła się koncepcja *phylogenetic footprinting*, polegająca na wykorzystywaniu kompleksowych analiz sekwencji międzygatunkowych do wykrywania konserwowanych miejsc regulatorowych. Umożliwiła separację elementów funkcjonalnych od neutralnych sekwencji genomicznych dla poszczególnych gatunków [27]. Miejsca regulatorowe są specyficzne dla poszczególnych gatunków ze względu na powszechną wymianę miejsc wiązania czynników transkrypcyjnych [28].

Kolejnym krokiem było ujęcie regulacji genów w postaci modułów cis-regulatorów (CRM), czyli zbiorów wielu motywów współdziałających w regulacji ekspresji. Modele skupiały się nie wokół pojedynczych motywów, a względnego położenia, uwzględniającego kontekst chromatyny. Pozwoliło to lepiej zrozumieć skomplikowaną współzależność wielu czynników w aktywacji enhancerów.

Pojawienie się technologii sekwencjonowania nowej generacji (NGS) i projekty takie jak *ENCODE* (Encyclopedia of DNA Elements) [55] radykalnie zwiększyły możliwości wykrywania obszarów regulatorowych. Przejście od immunoprecypitacji chromatyny hybrydyzowanej na mikromacierzach (ChIP-chip) do sekwencjonowania wysokoprzepustowego znacznie poprawiło rozdzielczość i czułość mapowania miejsc wiązania białek i modyfikacji histonowych. Jednocześnie metody analizujące dostępność chromatyny (ATAC-seq, Cut&tag) i techniki syntetyzujące długodystansowe kontakty (3C, Hi-C) pozwoliły na projekcję liniowych sekwencji na trójwymiarową funkcjonalność [29, 30].

Integracja wielowymiarowych danych (sekwencja, dostępność chromatyny, modyfikacje histonowe, enhancery, promotory, ekspresja RNA) umożliwiła budowę bardziej precyzyjnych modeli przewidujących lokalizację i aktywność elementów regulatorowych. Przyrównywanie do sekwencji konsensusowych pozwalało na jednoczesne wykrywanie polimorfizmów i unikalnych zmian. Równocześnie pojawiły się metody przewidywania CRM oparte na agregacji dużych zbiorów danych, w tym meta-analizy wielu sygnałów ChIP-seq i oddziaływań chromosomowych [28].

W odpowiedzi na stale rosnącą skalę i złożoność danych biologicznych wprowadzono metody uczenia maszynowego. Klasyczne algorytmy (drzewa decyzyjne, sieci bayesowskie, maszyny wektorów nośnych [2]) zostały uzupełnione przez wciąż wszechobecne w zastosowaniach medycznych lasy losowe oraz głębokie sieci neuronowe. Ich zastosowanie medyczne i bioinformatyczne to między innymi diagnostyka retinopatii cukrzycowej, ekspresja genów na mikromacierzach, analiza ekspresji białek w widmach spektralnych, odkrywanie biomarkerów i przewidywanie interakcji białko-białko [11].

Szczególną uwagę należy zwrócić na **konwolucyjne sieci neuronowe** (CNN), które potrafią automatycznie wydobywać z surowych sekwencji DNA zarówno lokalne motywy, jak i ich wyższe kombinacje, bez potrzeby ręcznej inżynierii cech. Narzędzia oparte na CNN, takie jak Basset, zademonstrowały znaczący postęp w przewidywaniu dostępności chromatyny i identyfikacji regionów regulatorowych bezpośrednio ze sekwencji nukleotydowej [4]. Kolejne rozszerzenia architektur, integrujące kontekst epigenomiczny lub informacje o trójwymiarowej organizacji chromatyny, systematycznie zwiększają trafność predykcji oraz ułatwiają interpretację wyuczonych reprezentacji [31].

W odniesieniu do unikalnych cech regulatorów transkrypcji i astrocytarnych fenotypów, wykazano istnienie licznych enhancerów specyficznych dla gatunku ludzkiego. Na podstawie modyfikacji chromatyny skorelowano geny odpowiedzialne za zaburzenia mózgu u *Homo Sapiens*. Równoległe badania nad astrocytami pochodzącymi z indukowanych pluripotentnych komórek macierzystych (**iAstrocytes**) naczelnych wskazały, że ludzkie astrocyty posiadają charakterystyczne sygnatury transkryptomiczne. Wskazuje to na ewolucyjne przekształcenie modułów astrocytów w linii ludzkiej [10, 32]. Nie można zatem brać pod uwagę jedynie uniwersalnych motywów sekwencji, bez uwzględnienia zmienności gatunkowej. Jest to szczególnie istotna obserwacja w kontekście badań chorób związanych z unikalnymi cechami ludzkiego genomu.

1.2. Uczenie maszynowe

Uczenie maszynowe (ang. *machine learning*) to poddziedzina sztucznej inteligencji, której celem jest konstruowanie algorytmów zdolnych do automatycznego uczenia się na podstawie danych. W odróżnieniu od programowania *explicite*, gdzie reguły są sztywno zdefiniowane przez programistę, algorytmy uczenia maszynowego budują model doskonalący swoje działanie poprzez ekspozycję na dane treningowe, bez bezpośredniego programowania pod konkretne zadanie. W kontekście problemów predykcyjnych wyróżnia się dwa podstawowe typy zadań: **klasyfikację**, gdzie model przyporządkowuje dane wejściowe do dyskretnych etykiet, oraz **regresję**, w którym przewidywaną wartością jest liczba ciągła. W zależności od charakteru dostępnych danych i mechanizmu uczenia, wyróżnia się trzy główne paradygmaty uczenia maszynowego [11]:

- Uczenie nadzorowane polega na obserwacji odpowiadających sobie par danych, a przedmiotem nauki jest funkcja odwzorowująca dane wejściowe w dane wyjściowe. Model na podstawie ciągu uczącego dąży do trafnego przewidywania etykiet przypisywanym nowym, nieznanym wcześniej danym.

- W przypadku uczenia nienadzorowanego, model uczy się rozpoznawania wzorców w danych wejściowych, bez wyraźnego sprzężenia zwrotnego. Najbardziej znanym zadaniem tego typu jest klasteryzacja - wykrywanie sensownego podziału obiektów wejściowych na grupy (klastry).

- Uczenie ze wzmacnianiem bazuje na zestawie nagród i kar. Model uczy się, które z możliwych akcji do podjęcia przed wzmocnieniem są odpowiedzialne za osiągnięcie ściśle określonego wyniku. W trakcie procesu trenowania poprawia swoje działania w zamiarze maksymalizacji nagrody w przyszłości.

W pracy analizujemy problem predykcji funkcji regulatorowych na podstawie surowej sekwencji nukleotydowej, gdzie każda sekwencja musi zostać zaklasyfikowana do jednej z czterech klas: aktywnego promotora, nieaktywnego promotora, aktywnego niepromotora lub nieaktywnego niepromotora. Taka formuła problemu jednoznacznie kwalifikuje go jako zadanie klasyfikacji w ramach uczenia nadzorowanego.

1.2.1. Uczenie nadzorowane

W uczeniu nadzorowanym dysponujemy zbiorem treningowym składającym się z N par $(x_i;\ y_i)$, gdzie x_i jest daną wejściową, a y_i odpowiadającą jej etykietą, czyli daną wyjściową [11]. Zakładamy, że istnieje pewna nieznana funkcja $y = f(x)$, którą model stara się przybliżyć. Celem jest zbudowanie takiego modelu (hipotezy), który na nowych danych będzie jak najdokładniej przewidywał prawdziwe etykiety, a nie tylko doskonale dopasuje się do danych treningowych.

Do rzetelnej ewaluacji modelu służy podział danych na następujące zbiory: treningowy, walidacyjny oraz testowy. Pierwszy z nich służy nauce modelu *sensu stricto*. Drugi wykorzystujemy do wybrania najlepszego modelu w czasie trwania całego treningu, a ostatni, nieużywany w trakcie uczenia, służy do nieobciążonej ewaluacji i sprawdzenia generalizacji.

Przeuczenie (ang. *overfitting*) występuje, gdy model nadmiernie dopasowuje się do danych treningowych, co prowadzi do gorszej wydajności na nowych danych. Niedouczenie (ang. *underfitting*) oznacza natomiast sytuację, gdy model nie jest w stanie dobrze odwzorować zależności nawet w zbiorze treningowym. Celem uczenia maszynowego jest zatem znalezienie nietrywialnego odzwierciedlenia, które jednocześnie jest proste i nieredukowalne, ale generalizuje się na cały problem. W tym celu często nie ograniczamy się do jednego modelu, a testujemy wiele hipotez poprzez dostrojenie **hiperparametrów** - regulowanych parametrów, które umożliwiają precyzyjne sterowanie procesem treningu.

Od prawidłowego wyboru hiperparametrów w dużej mierze zależy ostateczna wydajność modelu. Sam proces ich znajdywania również nie jest trywialny. W przypadku jednego można zastosować metodę bisekcji celem znalezienia jego optymalnej wartości. Natomiast dla wielu, najprostsze jest ręczne dostrajanie metodą prób i błędów. W praktyce, ze względu na istniejące ograniczenia czasowe i obliczeniowe, korzysta się z metody wyszukiwania gridowego. Polega ona na ewaluacji każdej możliwej, wcześniej zaproponowanej kombinacji na zbiorze walidacyjnym. Inną optymalną metodą jest optymalizacja bayesowska, która traktuje proces znajdowania hiperparametrów jako zadanie dla uczenia maszynowego samo w sobie [33]. Udowodniono, że przy odpowiednich założeniach jest ona nawet lepsza niż ręczne dostrajanie przez ekspertów [34].

Proces uczenia modelu polega na minimalizacji **funkcji kosztu** (ang. *cost function*), która mierzy, jak bardzo przewidywania modelu odbiegają od rzeczywistych wartości. Funkcja ta składa się z dwóch składników: funkcji straty (ang. *loss function*), która oblicza błąd pomiędzy przewidywaniami modelu a prawdziwymi etykietami dla pojedynczego przykładu oraz składnika regularyzacyjnego, który karze model za nadmierną złożoność, zapobiegając w ten sposób przeuczeniu.

$$Cost(h) = LossFunction(h) + \lambda \cdot Complexity(h)$$

$$\hat{h} = \underset{h \in \mathcal{H}}{\operatorname{argmin}}\ Cost(h)$$

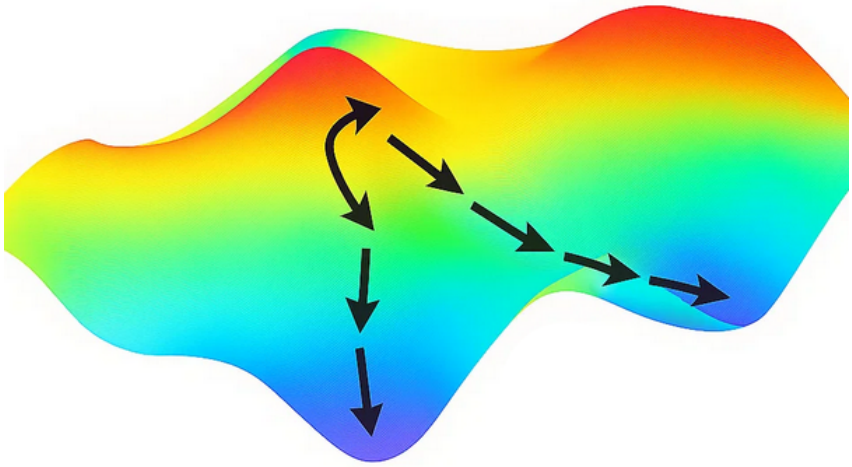
Hiperparametr λ kontroluje siłę regularyzacji. Jego prawidłowe dobranie pozwala znaleźć kompromis pomiędzy dopasowaniem modelu do danych treningowych a jego zdolnością do uogólniania na nowe dane. Proces trenowania można więc przedstawić jako poszukiwanie takiej hipotezy $\hat{h}$ w przestrzeni hipotez $\mathcal{H}$, która minimalizuje łączny koszt, tym samym najlepiej tłumacząc istniejące odwzorowanie pomiędzy danymi wejściowymi a wyjściowymi w zbiorze uczącym.

1.2.2. Spadek gradientowy

W uczeniu nadzorowanym najczęściej stosuje się algorytm spadku gradientowego do minimalizacji funkcji straty. Parametry modelu traktujemy jako wektor wag w, który iteracyjnie dostosowujemy. W każdym kroku oblicza się gradient funkcji straty względem wag i wykonuje krok w kierunku przeciwnym do gradientu:

$$w_i \leftarrow w_i - \eta \cdot \frac{\partial}{\partial w_i} \cdot LossFunction(\mathbf{w})$$

Hiperparametr η, który określa wielkość kroku, nazywamy **współczynnikiem uczenia** (ang. *LR - learning rate*). W trakcie procesu uczenia może być stałą wartością lub być konsekwentnie zmniejszany w miarę postępów. Ze względu na złożoność problemu oraz wybór η, który nie może być za duży, ponieważ wtedy istniałoby ryzyko przekroczenia owego minimum, nie przetwarza się wszystkich przykładów ze zbioru treningowego na raz. Istnieją jednak efektywne odmiany, które zrezygnowały z pełnego determinizmu na rzecz efektywności. Istotą stochastycznego spadku gradientowego (ang. *SGD - Stochastic Gradient Descent*) jest pokazywanie losowo wybranego przykładu ze zbioru uczącego, co pozwala na ciągłe dostosowywanie wag.



Rysunek 1.3: Przestrzeń wagowa pokazana jest schematycznie jako rozkład wartości funkcji straty w zależności od ustawienia wag. Dla problemów wypukłych (np. regresji liniowej) powierzchnia straty ma jedno minimum globalne. Natomiast w zadaniach nieliniowych zazwyczaj występuje wiele minimów lokalnych. Algorytm spadku gradientowego zbiega do jednego z nich zależnie od inicjalizacji, współczynnika uczenia i szumu stochastycznego, dlatego nie ma gwarancji znalezienia rozwiązania globalnego na przedstawionej powierzchni.

Krok, w którym zostanie przetworzony cały zbiór treningowy, nazywamy **epoką**. W praktyce dla dużej liczby przykładów N, wybierany jest podzbiór n przykładów, zwanych miniwsadem (ang. *minibatch*). Ilość obliczeń w każdym kroku redukuje się dzięki temu o czynnik $\frac{n}{N}$, przy jednoczesnym wzroście standardowego błędu estymacji gradientu o $\sqrt{n}$. Zatem algorytm SGD jest dokładnie $\sqrt{n}$ razy szybszy od deterministycznej wersji. Tym samym jest jednym z kluczowych hiperparametrów. Wielkość miniwsadu może być dostosowana do możliwości sprzętowych (np. równoległego przetwarzania na GPU), by przyspieszyć trening [11].

1.2.3. Miary jakości

Celem wszystkich epok, które składają się na proces treningu modelu, jest znalezienie optymalnych wag w kontekście minimalizacji funkcji kosztu. Ostateczna weryfikacja odbywa się jednak na wyłączonym z uczenia zbiorze testowym, poprzez różnorodne metryki. Pomimo tego, że jest to kluczowa kwestia w uczeniu maszynowym, nie osiągnięto jeszcze powszechnego konsensusu w sprawie wybrania jednolitej miary [36].

Rozpatrując klasyfikację binarną (dwie klasy: pozytywna i negatywna) podstawowym narzędziem jest macierz pomyłek (ang. *confusion matrix*), zawierająca liczbę prawdziwych pozytywów (TP), fałszywych negatywów (FN), fałszywych pozytywów (FP) i prawdziwych negatywów (TN). Macierz pomyłek definiuje się następująco:

	Predykcja Pozytywna	**Predykcja Negatywna**
Rzeczywiście Pozytywna	**TP** (prawdziwe pozytywy)	**FN** (fałszywe negatywy)
Rzeczywiście Negatywna	**FP** (fałszywe pozytywy)	**TN** (prawdziwe negatywy)

Błędem pierwszego rodzaju (ang. *false positive rate*) nazywamy odsetek fałszywie zaklasyfikowanych pozytywów spośród wszystkich przypadków negatywnych. Jest to równoważne częstości fałszywych alarmów. Za błąd drugiego rodzaju (ang. *true positive rate*) rozumiemy sklasyfikowanie przypadków pozytywnych jako negatywne. Bezpośrednio z tymi terminami związane są czułość i swoistość. Przyjmują one wartości od 0 do 1, gdzie druga wartość oznacza brak błędu odpowiedniego rodzaju.

- Czułość (ang. *sensitivity*, SN): Prawdopodobieństwo, że klasyfikacja będzie poprawna pod warunkiem, że przypadek jest pozytywny. W terminologii medycznej oznacza to, że test wykonany dla osoby chorej wykaże jej chorobę.

$$\text{Czułość} = \frac{TP}{P} = \frac{TP}{TP + FN}$$

- Swoistość, specyficzność (ang. *specificity*, SP) Prawdopodobieństwo, że klasyfikacja będzie poprawna, pod warunkiem, że przypadek jest negatywny.

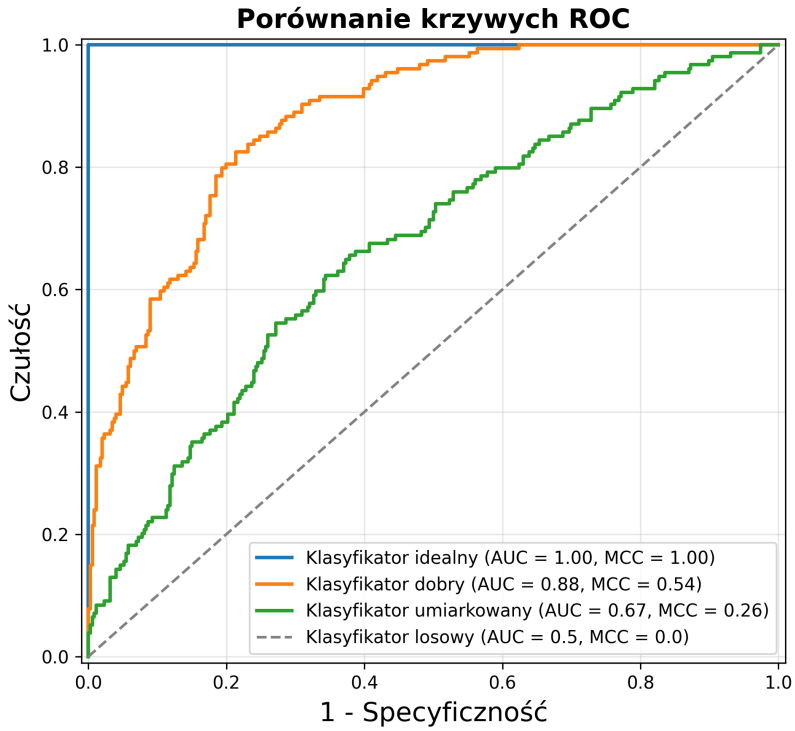
$$\text{Swoistość} = \frac{TN}{N} = \frac{TN}{FP + TN}$$

Powyżej rozważane miary są sparowane, tzn. trzeba obie jednocześnie brać pod uwagę przy ocenie klasyfikatora. Wysoka czułość oznacza wykrycie wszystkich prawdziwie pozytywnych przypadków, ale może prowadzić do wielu fałszywie dodatnich wyników. Z kolei wysoka swoistość pozwala odróżnić przypadki pozytywne od negatywnych, ale może skutkować fałszywie ujemnymi wynikami.

Istnieją jednak miary, które są sprowadzone do jednej liczby, która w pewien sposób opisuje całościowo wynik. Jedną z nich jest powierzchnia pod krzywą charakterystyki roboczej odbiorcy (**ROC**). Każdy punkt na tej krzywej otrzymywany jest dla ustalonej wartości progu i ma współrzędne (1 - specyficzność; czułość). Powierzchnia pod krzywą ROC (ang. *area under curve*, **AUC**) jest miarą bezprogową opisującą zdolność modelu do rozróżnienia klas. Formalnie AUC jest miarą prawdopodobieństwa, że klasyfikator przypisze wyższą ocenę losowo wybranemu przypadkowi pozytywnemu niż losowo wybranemu przypadkowi negatywnemu.

Innym współczynnikiem uwzględniającym błędy fałszywie pozytywne i negatywne jest współczynnik korelacji Matthewsa (ang. **MCC** - *Matthews Correlation Coefficient*). Uchodzi on za zrównoważoną miarę, która może być stosowana nawet wtedy, gdy klasy są silnie niezbalansowane. Przyjmuje wartości od -1 (całkowicie błędna klasyfikacja) przez 0 (losowa) do $+1$ (idealna klasyfikacja). Jest to szczególnie wiarygodna miara przy niezbalansowanych zbiorach danych, ponieważ wymaga dobrych wyników w każdej kategorii macierzy pomyłek, by uzyskać wysoki wynik [35, 36].

$$MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$



Rysunek 1.4: Obrazowe porównanie ROC AUC oraz MCC dla kilku modeli. jest to powszechna miara standardowej oceny klasyfikacji binarnej w większości dziedzin naukowych [35]. AUC może wynosić od 0 (całkowicie błędna predykcja) do 1 (wynik idealny).

1.3. Głębokie uczenie

Głębokie uczenie (ang. *deep learning*) to poddziedzina uczenia maszynowego, w której modele (sieci neuronowe) mają wiele warstw. Jego początki sięgają dawnych prac, w których starano się wymodelować architekturę zbliżoną do działania mózgu [37]. Słowo 'głęboki' odnosi się do istnienia wielu warstw, przez które przechodzi cała ścieżka obliczeń. Wielowarstwowa architektura pozwala na hierarchiczne uczenie reprezentacji danych - od cech niskiego poziomu do coraz bardziej abstrakcyjnych. Sieć neuronowa złożona jest z warstwy wejściowej, warstw ukrytych i warstwy wyjściowej.

Sieci ze sprzężeniem w przód, zgodnie ze swoją nazwą, mają połączenia tylko w jednym kierunku. Ich architekturę można przedstawić jako skierowany graf acykliczny z wyznaczonymi węzłami wejściowymi i wyjściowymi. W każdym węźle, zwanym także neuronem, obliczana jest funkcja jego wejść, a jej wynik przekazywany jest następnikom w sieci. Wszystkie wagi pełnią funkcję parametrów, których iteracyjne dostosowanie pozwala na dopasowanie do danych treningowych podczas procesu uczenia.

Zadaniem każdego węzła jest obliczenie ważonej sumy jego danych wejściowych pochodzących z węzłów poprzedzających. Następnie jej przekształcenie za pomocą wybranej funkcji aktywacji. Dzięki temu model może reprezentować w zasadzie dowolną funkcję za cenę odpowiedniego rozmiaru. W przypadku funkcji aktywacji, która jest nieliniowa i ograniczona, twierdzenie o uniwersalnej aproksymacji gwarantuje, że do przybliżenia dowolnej funkcji ciągłej wystarcza pojedyncza warstwa ukryta. Nie określa jednak praktycznych aspektów, w jaki sposób znaleźć optymalną konfigurację sieci ani wartości jej wag.

Niech a_j oznacza wynik uzyskany przez neuron j i $w_{i,j}$ będzie wagą prowadzącą od jednostki i do jednostki j. Natomiast wyraz wolny (ang. **bias**) b oznacza wyraz wolny danego neuronu, dodawany niezależnie od sygnału wejściowego, a g jest funkcją aktywacji przypisaną do jednostki j. Wówczas zachodzi następująca zależność:

$$a_j = g_j(\sum_i w_{i,j} \cdot a_i + b_j)$$

W praktyce jednak funkcje aktywacyjne wykorzystywane w systemach głębokiego uczenia są zazwyczaj monotonicznie niemalejące, czyli pochodne g' są nieujemne. Najczęściej można spotkać:

- funkcja **ReLU** (ang. *recified linear unit*):

$$\mathrm{ReLU}(x) = \max(0, x)$$

- funkcja **logistyczna**, zwana także sigmoidalną:

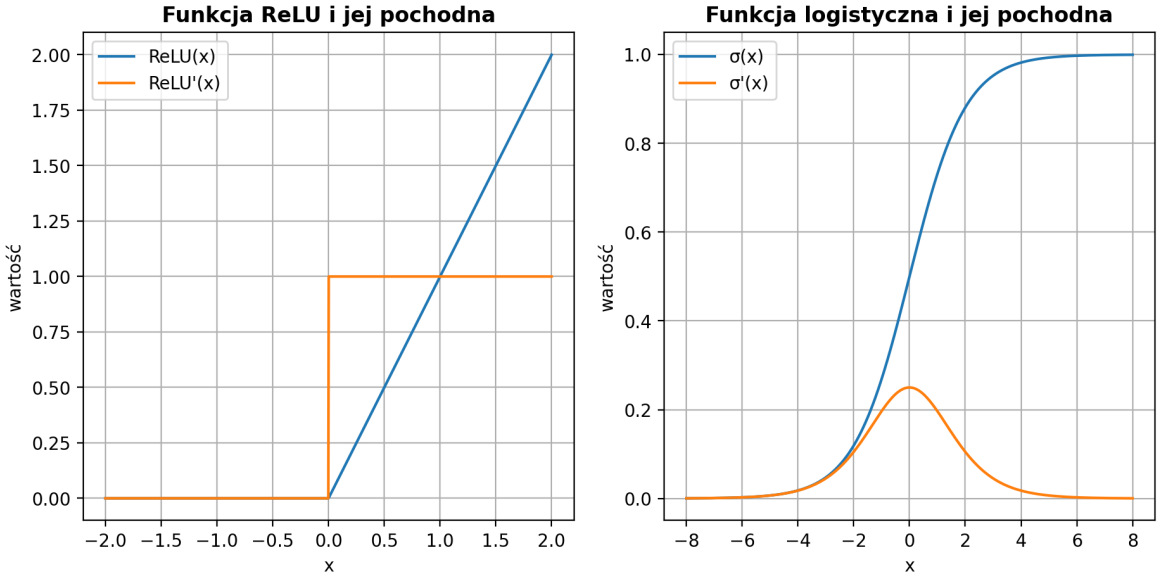
$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

- tangens hiperboliczny:

$$\tanh(x) = \frac{e^{2x} - 1}{e^{2x} + 1} = 2 \cdot \sigma(2x) - 1$$

- funkcja **softplus**, która stanowi *de facto* wygładzoną wersją funkcji $ReLU$:

$$\mathrm{softplus}(x) = ln(1 + e^x)$$



Rysunek 1.5: Dwie najbardziej popularne w głębokim uczeniu maszynowym funkcje aktywacyjne oraz ich pochodne.

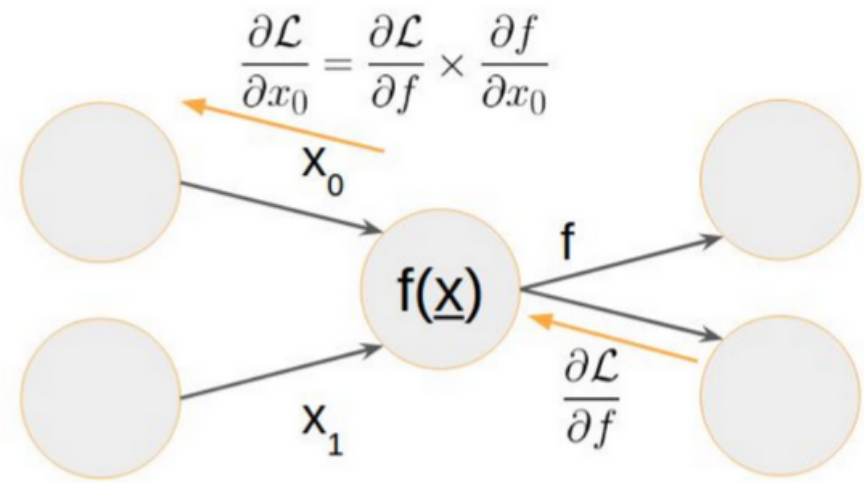
Domyślną strukturą sieci neuronowych jest graf w **pełni połączony** (ang. *fully connected*), to znaczy, że każdy neuron połączony jest z każdym węzłem następnej warstwy. Dzięki temu możemy traktować warstwy jako wektory i zastosować algorytmy oparte na spadku gradientowym. Bezpośrednio stosujemy je w przypadku ostatniej warstwy, czyli tej, która wytwarza informację wyjściową dla całej sieci. Dla prowadzących do nich kolejnych warstw ukrytych, aż do warstwy wejściowej, obliczanie gradientu jest niewiele bardziej skomplikowane i opiera się na regule łańcuchowej, która pozwala na policzenie pochodnej funkcji złożonej.

$$\frac{\partial g(f(x))}{\partial x} = g'(f(x)) \cdot \frac{\partial f(x)}{\partial x}$$

Z tego faktu korzysta podstawowy algorytm uczenia maszynowego wielowarstwowych sieci neuronowych o nazwie **propagacja wsteczna** (ang. *back-propagation*) [38]. Polega on na przekazywaniu w głąb sieci gradientów, które są pochodnymi cząstkowymi funkcji straty względem wag. Propagacja rozpoczyna się od warstwy wyjściowej, gdzie gradient obliczany jest bezpośrednio z formuły wyrażającej funkcję kosztu w zależności od przewidywanych wartości, a kończy na warstwie wejściowej. Ze względu na wymnażanie przez kolejne czynniki zawierające lokalne pochodne może wystąpić zjawisko zanikającego gradientu [39, 11]. Co prawda wszystkie pochodne są nieujemne, ale mogą być bliskie zeru, gdy zmiana wag prowadzących do danego neuronu nie będzie miała żadnego wpływu na daną jednostkę.

Jeżeli natomiast neuron warstwy ukrytej przyczynił się do powstania błędu na węźle warstwy wyjściowej, to trzeba mu dostarczyć informację o błędzie, proporcjonalną do wartości wagi. Tym samym wsteczna propagacja jest efektywnym narzędziem obliczeniowym komplementarnym do metody **największej wiarygodności**. Minimalizując funkcję straty, jednocześnie szukamy takich parametrów naszego modelu, dla których prawdopodobieństwo znalezienia danego ciągu uczącego będzie jak największe.

$$w^* = \underset{w}{\operatorname{argmin}} - \sum_{j=1}^{N} log\mathbb{P}_w(y_j|x_j)$$



Rysunek 1.6: Sygnał wejściowy w sieci przechodzi przez kolejne warstwy od strony lewej do prawej. Natomiast obliczanie wstecznej propagacji przebiega od prawej do lewej.

Jako, że funkcja logarytmiczna jest ściśle rosnąca, maksymalizowanie prawdopodobieństwa jest równoważne maksymalizacji jego logarytmu. Korzystając z faktu, że logarytm iloczynu równa się sumie logarytmów poszczególnych czynników, szukamy wektora $w*$, który minimalizuje sumę zanegowanych logarytmów dla N przykładów. Możemy na to także spojrzeć, jak na minimalizację funkcji mierzącej odmienność dwóch rozkładów od siebie. Funkcję tę nazywamy entropią wzajemną i definiujemy w następujący sposób [11]:

$$H(P,Q) = \mathbb{E}_{z\sim P(\mathbf{z})}[log\ Q(z)\ dz] = H(P) + D_{KL}(P\ ||\ Q) = H(P) + \int P(z) log\ \frac{P(Z)}{Q(z)}\ dz$$

$$D_{KL}(P\ ||\ Q) = \int P(z) log\ \frac{P(Z)}{Q(z)}\ dz \tag{1.1}$$

Gdzie D_{KL} jest **dywergencją Kullbacka-Leiblera**. Dla ustalonego rozkładu P, który domyślnie jest prawdziwym rozkładem naszego zbioru danych, minimalizacja entropii odbywa się poprzez dostosowanie wag, aby rozkład hipotezy Q był jak najbardziej do niego zbliżony. Wtedy interpretujemy wynik otrzymany przez naszą sieć jako prawdopodobieństwo przynależności do odpowiedniej klasy. W uczeniu maszynowym nierzadko zdarza się, że klasyfikatory są wykorzystywane do predykcji wielu kategorii. Jeśli zatem istnieje k możliwości, potrzebujemy dokładnie k węzłów wyjściowych, którym przypisane prawdopodobieństwa sumują się do jednego. Do realizacji tego celu używamy warstwy **softmax**, dzięki której k-ty element wektora wyjściowego zadany jest wzorem:

$$softmax(\text{input})_k := \frac{\exp(\text{input}_k)}{\sum_{k'=1}^{d} \exp(input_{k'})}$$

Wejście input_k do każdego z węzłów wyjściowych odpowiada ważonej kombinacji liniowej wyników z poprzedniej warstwy. Warstwa *softmax* jest nie tylko gładka i różniczkowalna, ale ze względu na wykładniczy charakter uwydatnia różnice w wartościach wejściowych. Na przykład wektor input $= [5;\ 2;\ 0;\ -2]$ transformuje się na wektor $[0.946;\ 0.047;\ 0.006;\ 0.001]$ [11].

1.3.1. Konwolucyjne sieci neuronowe

Sekwencje DNA nie są zwykłymi, nieuporządkowanymi wektorami cech. W kontekście translacji istotne jest pojęcie **ramki odczytu**, ponieważ informacja genetyczna odczytywana jest w trójkach, nienachodzących na siebie nukleotydów (kodonach) po transkrypcji mRNA. Każda z nich jednoznacznie koduje jeden aminokwas. Kodonem startowym jest `AUG`, który inicjuje syntezę białka. Zatem kluczowa nie jest tylko kolejność nukleotydów, ale także ich wzajemne położenie oraz sąsiedztwo.

Wcześniej omówione architektury traktują każdą pozycję wejściową jako indywidualną cechę. Gdyby trenować model składający się z w pełni połączonych warstw, to niezależnie od wprowadzonej permutacji, jego proces treningu przebiegałby identycznie. Jednak zaburzenie pozycji w sekwencji niszczy jej biologiczny kontekst i prowadzi do utraty informacji funkcjonalnej. W konsekwencji model uczony na permutowanych danych nie odzwierciedla żadnych rzeczywistych reguł molekularnych. Dla zadań biologicznych korzystniejsze będą podejścia zakładające niezmienniczość przestrzenną i współdzielenie wag, aby wykrywać wzorce niezależnie od ich położenia. Dzięki temu nasz model będzie zawierał mniejszą liczbę wag, co przyspieszy proces uczenia.

Konwolucyjne sieci neuronowe (ang. *convolutional neural network*, CNN) to wielowarstwowe modele, zawierające przestrzennie lokalne połączenia, uczące się reprezentacji danych poprzez współdzielone **filtry**, **jądra** (ang. *kernels*). Pierwotnie były konstruowane dla problemu klasyfikacji z założeniem, że na wejściu podawany jest obraz [40]. Schemat, według którego wagi łączone są z wejściem, nosi nazwę **konwolucji** lub **splotu**. Filtry przesuwają się po wejściu, ucząc się wykrywać cechy niższego poziomu. Kolejne warstwy agregują je w bardziej abstrakcyjne reprezentacje. Sieć zatem realizuje pewne odwzorowanie w przestrzeń nazw klas obiektów reprezentowanych przez dane wejściowe.

Sieci konwolucyjne mają neurony zorganizowane w trzywymiarową strukturę: szerokość, wysokość, głębokość. Każdy z nich w danej warstwie jest połączony tylko z niewielką liczbą neuronów w warstwie poprzedzającej, a każda warstwa przekształca objętość wejściową w objętość wyjściową. Każdy filtr w warstwie konwolucyjnej składa się z macierzy, która przesuwana jest po wymiarach przestrzennych (szerokość i wysokość) objętości. Odległość między dwoma najbliższymi polami, na których centrujemy jądro, nazywamy **krokiem**, albo **rytmem** konwolucji [11].

Dany filtr produkuje na wyjściu dwuwymiarową macierz odpowiadającą swojej aktywacji dla wszystkich położeń w objętości wejściowej. Uzyskane mapy aktywacji od poszczególnych jąder są składane poprzez ułożenie ich w stos wzdłuż trzeciego wymiaru - głębokości. Może to zatem doprowadzić do zmniejszenia wymiaru wejścia. Aby temu przeciwdziałać, stosuje się wypełnienie (ang. *padding*). Metoda ta umożliwia otrzymanie rozmiaru wyjścia takiego samego jak rozmiar wejścia kosztem dodania ustalonych, syntetycznych, najczęściej zerowych wartości na krawędziach, co jednocześnie zapewnia skuteczność i prostotę.

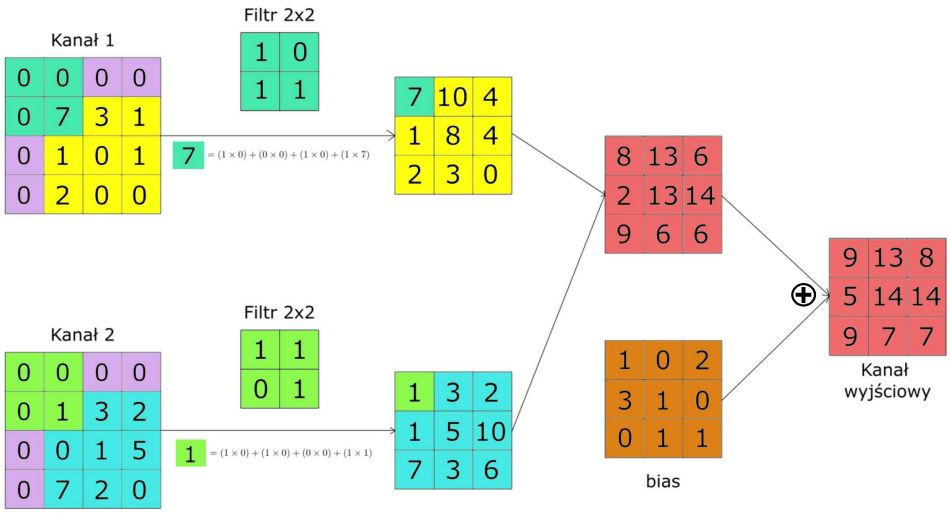
Zazwyczaj między warstwami konwolucyjnymi znajdują się dodatkowe warstwy zbiorcze, zwane inaczej kumulacyjnymi (ang. *pooling layers*). Ich zadaniem jest swoiste podsumowanie działania sąsiadujących jednostek w postaci jednej wartości. Również charakteryzują się posiadaniem jądra o zadanym rozmiarze, krokiem i uzupełnieniem, jednak nie posiadają wag. Operacja jest *a priori* ustalona, a nie wyuczona, jak w przypadku filtrów konwolucyjnych. Ponadto w warstwie kumulacyjnej zazwyczaj nie występuje funkcja aktywacji. Najpopularniejsze rodzaje to:

- **kumulacja maksymalizująca** (ang. *max pooling*) to zastępowanie wartości wejściowych o z góry określonym rozmiarze ich wartością maksymalną. Pozwala ona na zachowanie najważniejszych cech, zwiększając czułość sieci na poszczególne wzorce. Zmniejsza również liczbę wag potrzebnych w kolejnych warstwach (ang. *downsampling*) co prowadzi do zmniejszenia kosztów obliczeń i przyspieszenia procesu uczenia.

- **kumulacja uśredniająca** (ang. *average pooling*) to zastępowanie danych wejściowych poprzez ich średnie. Ułatwia ona rozpoznawanie obiektów w różnych skalach. Jest także wykorzystywana do downsamplingu, ponieważ zmniejsza częstotliwość próbkowania.

Chociaż architektury zawierające warstwy konwolucyjne wykorzystywane są bardzo często w aplikacjach widzenia komputerowego, stanowią uniwersalne narzędzie. W genomice wejściowa sekwencja DNA jest zwykle kodowana z wykorzystaniem **gorącej jedynki** (ang. *one-hot encoding*). Przy użyciu transformacji możemy uzyskać dwuwymiarową macierz zer i jedynek, gdzie wartość 1 w odpowiednim wierszu oznacza rodzaj nukleotydu na odpowiedniej pozycji. Aktywacja każdego filtra świadczy o wystąpieniu odpowiedniego motywu w określonym przesunięciu sekwencji. Tłumaczy to bezpośrednio, dlaczego sieci CNN doskonale sprawdzają się w uczeniu zależności wyższego rzędu i ukrytych wzorców w danych dotyczących ekspresji genów [40, 41, 42, 43].

Jednym z najbardziej znaczących zastosowań konwolucyjnych sieci neuronowych w regulomice było opracowanie narzędzia **Basset**, które zaproponowało wielozadaniowe podejście do przewidywania dostępności chromatyny na podstawie surowej sekwencji DNA [4]. Praktyczne osiągnięcia w tej dziedzinie obejmują odwzorowanie wag filtrów na biologiczne motywy wiązania białek oraz poprawę predykcji czynników transkrypcyjnych. Pozwalają także na ocenę wpływu mutacji metodą *in-silico*, co umożliwia kategoryzację wariantów regulacyjnych w kontekstach chorobowych [43, 44].



Rysunek 1.7: Przykład działania konwolucji dla dwóch różnych kanałów. Widać, jak zastosowanie wypełnienia zerami pozwala zachować oryginalny rozmiar wejścia. Każdy filtr potencjalnie może zawierać także bias, który jest dodawany na wyjściu.

1.3.2. Uczenie transferowe

Rozmaite motywy w sekwencji DNA mogą być do siebie zbliżone w ramach jednego gatunku. Nauka o astrocystach, które dostarczają neuronom wszelkich substancji odżywczych, może dostarczyć nam informacji o genach związanych z chorobami mózgu [10]. W uczeniu maszynowym przeniesienie doświadczenia zdobytego z badania jednego problemu na grunt innego nosi nazwę **uczenia trasferowego**. W uniwersum sieci neuronowych proces uczenia polega wyłącznie na dostrajaniu wag, dlatego ich skopiowanie, a następnie dostrojenie (ang. *fine-tuning*) na nowym problemie wydaje się być naturalnym podejściem. Kluczowe w kontekście uczenia transferowego wydaje się dobór zbliżonych problemów.

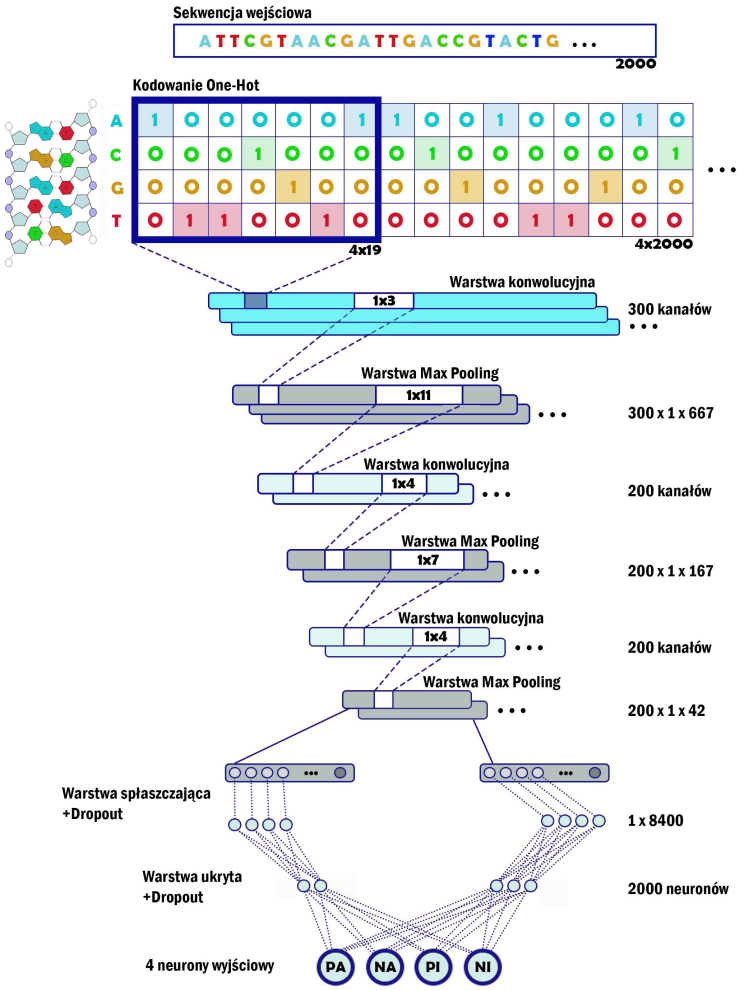
Naszą architekturę wytrenowaliśmy początkowo na próbkach pochodzących od polskich pacjentów z glejakami mózgu [9]. Sekwencje pochodziły od 16 pacjentów o zróżnicowanym stopniu złośliwości histopatologicznej: 8 pacjentów ze stopniem IV oraz po 4 pacjentach ze stopniami I i II/III. Pojedyncza obserwacja miała długość 2000 nukleotydów wraz z informacją o lokalnym poziomie acetylacji chromatyny, użytej jako etykietę określającą aktywność regulatorową. Regiony o podwyższonym poziomie zostały podzielone na dwie klasy: **aktywny promotor** (*promoter active*, *PA*) oraz niepromotor aktywny (*nonpromoter active*, *NA*). W drugiej klasie zawarte są enhancery, za które uznano regiony regulatorowe, nie znajdujące się bezpośrednio przed kodowanym genem. Przeciwnymi klasami są nieaktywny promotor (*promoter inactive*, *PI*) oraz nieaktywny niepromotor (*nonpromoter inactive*, *NI*). Ostatnia klasa zawiera sekwencje obejmujące regiony pozbawione podwyższonej acetylacji, usytuowane za genem i niepowiązane z żadną wcześniej zdefiniowaną klasą.

Opisane dane pozwoliły na stworzenie zbioru uczącego *Alt*, składającego się z 40 tysięcy sekwencji dla każdej klasy. Został on podzielony na podzbiór treningowy, walidacyjny oraz testowy według chromosomów, do których należą sekwencje, co jest zwyczajową praktyką [45]. Należy podkreślić, że przynależność sekwencji do klasy opiera się na założeniu, że istnieje deterministyczna reguła określająca poziom acetylacji. Niekoniecznie odpowiada to biologicznej rzeczywistości, w której komórki tego samego typu o wspólnym genomie mogą wykazywać różne profile ekspresji i różne stany epigenetyczne [3].

Natomiast drugi, mniejszy zbiór, nazwany *HighSignal*, posłużył do douczenia modelu. Został stworzony na tych samych zasadach co zbiór bazowy, jednak zawiera sekwencje pochodzące z astrocytów płodowych, a nie z glejaków. Zostały one pozyskane z indukowanych komórek macierzystych, które przekształciły się w procesie różnicowania. W badaniach [10] wykazano, że geny związane z chorobami mózgu są w ludzkich astrocytach częściej wyciszane niż aktywowane w porównaniu z astrocytami innych naczelnych. Taka różnica epigenetyczna może przyczyniać się do specyficznych funkcji ludzkiego mózgu.

1.4. Architektura *Custom*

W celu analizy sekwencji DNA i predykcji ich właściwości funkcjonalnych, przynależności do jednej z czterech klas zaprojektowano jednokanałową konwolucyjną sieć neuronową [1]. Na wejściu przyjmowana jest sekwencja o stałej długości, reprezentowana jako dwuwymiarowy tensor, aby pozyskać kontekst przestrzenny (podsekcja 1.3.1). Architektura składa się z sekwencyjnego stosu warstw konwolucyjnych, po których następują dwie warstwy w pełni połączone.



Rysunek 1.8: Architektura każdego modelu wykorzystywanego w niniejszej pracy. Ekstrakcję cech niskiego i wysokiego poziomu odbywa się za pomocą stosu warstw konwolucyjnych z poolingiem, po której następuje warstwa klasyfikująca. Ostatnią warstwą jest warstwa *softmax* z czterema jednostkami wyjściowymi, które reprezentują prawdopodobieństwo przynależności do odpowiedniej klasy.

Pierwsza warstwa konwolucyjna składa się z 300 filtrów o rozmiarze jądra (4, 19). Liczba wierszy pozwala na jednoczesne analizowanie pozycji wszystkich nukleotydów, ucząc się motywów w pełnej reprezentacji. Wymiar dopełnienia jest obliczany tak, aby zachować szerokość warstwy wejściowej po operacji splotu. Następne warstwy konwolucyjne składają się z 200 filtrów, skupiając się wyłącznie na wykrywaniu bardziej złożonych, nieliniowych wzorców wzdłuż sekwencji.

Bezpośrednio po każdej warstwie konwolucyjnej zastosowano metodę normalizacji wsadowej. Jest to powszechnie stosowana technika poprawiająca szybkość zbieżności poprzez przeskalowanie wartości wychodzących z wewnętrznych warstw na podstawie przykładów z każdego miniwsadu. Polega ona na wprowadzeniu dwóch parametrów normalizacji: β i γ, które są przedmiotami uczenia, jak inne parametry sieci.

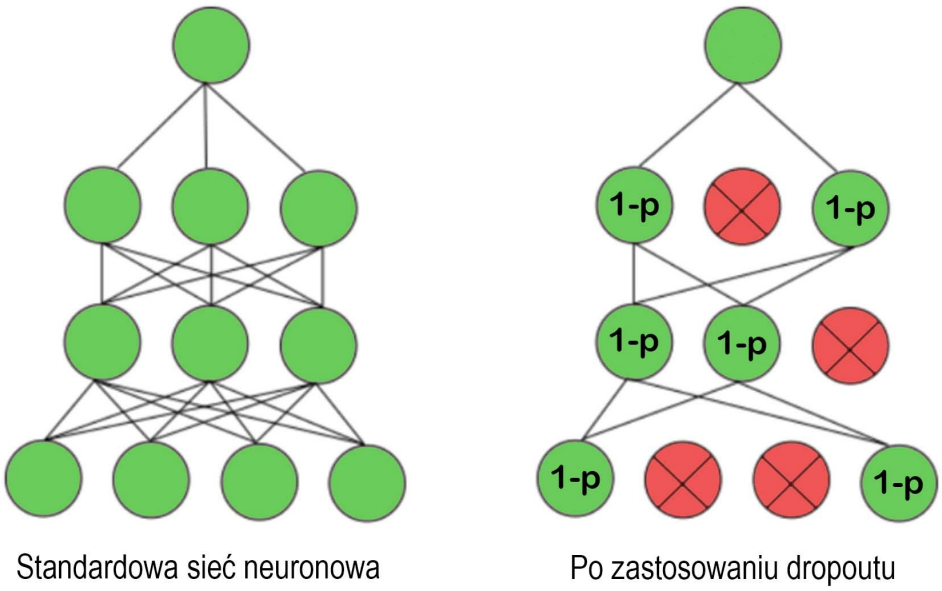
$$\hat{z}_i = \gamma \cdot \frac{z_i - \mu}{\sqrt{\epsilon + \sigma^2}} + \beta$$

Gdzie dla n przykładów, wartości tworzące miniwsad $z_1;\ z_2;\ \ldots; z_n$ zostają zastąpione przez $\hat{z_1};\ \hat{z_2};\ \ldots; \hat{z_n}$. Natomiast μ jest wartością średnią, a σ odchyleniem standardowym powyższego zbioru. Aby zapobiec potencjalnemu dzieleniu przez zero, dodawana jest także niewielka dodatnia stała ϵ. Normalizacja wsadu zapobiega ryzyku utraty informacji, gdy wagi danej warstwy są zbyt małe [11].

Nieliniowość wprowadzana jest za pomocą funkcji *ReLU*, która zastępuje wszystkie ujemne wartości aktywacji zerami. Każdy blok kończy się operacją max pooling w wymiarze odpowiadającym długości sekwencji. Kumulacja redukuje rozmiar reprezentacji i zapewnia inwariantność do małych motywów. Długość sekwencji jest sukcesywnie redukowana z 2000 do 667, następnie 167 i ostatecznie do 42. Jednocześnie liczba kanałów rośnie z 1 do 300, a następnie do 200. Tak otrzymany wektor cech o wymiarze 8400 jest spłaszczany i przekazywany do modułu klasyfikującego, który składa się z sekwencji w pełni połączonych (gęstych) warstw neuronowych (ang. *fully connected layers*).

Pomiędzy warstwami gęstymi zastosowano **dropout** z prawdopodobieństwem $p = 0.5$. Technika ta, będąca formą efektywnej regularyzacji, w naszym przypadku losowo zeruje aktywacje połowy neuronów w warstwach w pełni połączonych podczas każdej iteracji treningowej [46]. W tak zmodyfikowanej sieci podczas treningu wykonywana jest następnie faza propagacji wstecznej, ale na etapie testowania model uruchamiany jest bez udziału dropoutu. Wymuszamy na nim wyuczenie się wielu różnych, solidnych wyjaśnień dla każdej klasy wyjściowej, czym przeciwdziałamy przeuczeniu. Równocześnie wykorzystano dropout z prawdopodobieństwem 0.2 w warstwach konwolucyjnych, aby zwiększyć ogólną wydajność naszej architektury. Wprowadzając do modelu szum, który utrudnia dopasowanie się do zbioru treningowego, czynimy ów model bardziej odporny na szumy zawarte w danych wejściowych.

Dodatkowa regularyzacja, która obejmuje wszystkie parametry modelu, polega na dodaniu czynnika penalizującego $\lambda \cdot \sum_{i,j} W_{i,j}^2$ do funkcji straty, gdzie λ jest hiperparametrem. Pozwala ona na ograniczenie złożoności poprzez mechanizm wygaszania wag (ang. *weight decay*). W praktyce najczęściej stosuje się wartości rzędu 10^{-4} i tak też było w tym przypadku. Wszystkie treningi sieci odbywały się dla $\lambda = 1 \cdot 10^{-4}$. Dzięki temu utrzymujemy wartości sigmoidu w pobliżu części liniowych, z dala od płaskich odcinków, na których zanikają gradienty. W przypadku innych funkcji aktywacyjnych wygaszanie wag również wydaje się korzystne i wpływa znacząco użytecznie na generalizację różnorodnych architektur [11, 13].



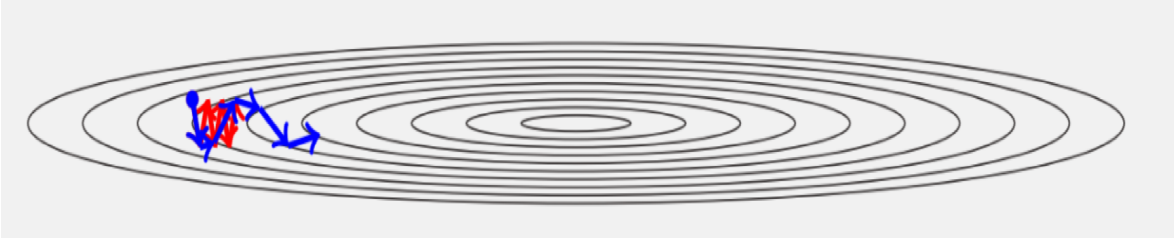
Rysunek 1.9: Przykładowe zastosowanie dropoutu z prawdopodobieńswtem p dla pierwszych trzech w pełni połączonych warstw fragmentu sieci neuronowej.

Proces optymalizacji parametrów modelu został dopełniony za pomocą algorytmu RMSprop [53]. Normalizując skumulowane gradienty, dostosowuje on adaptacyjnie kroki uczenia dla każdego parametru indywidualnie. Algorytm ten został wzbogacony o mechanizm `momentum` [52], który znacząco przyspiesza zbieżność w kierunkach o stabilnym gradiencie oraz tłumi oscylacje w obszarach o wysokiej zmienności. Aktualizacja parametrów odbywa się wtedy według następujących wzorów:

$$v_t = \gamma \cdot v_{t-1} + \eta \cdot \nabla J(\theta_t) \tag{1.2}$$

$$\theta_{t+1} = \theta_t - v_t$$

- v_t to skumulowany gradient w kroku t, który przechowuje bezwładność poprzednich aktualizacji,
- γ to współczynnik `momentum` (domyślnie przyjęty jako $\gamma = 0.1$), który określa udział poprzednich gradientów w bieżącej aktualizacji,
- η to szybkość uczenia podczas bieżącej epoki,
- $\nabla J(\theta_t)$ to gradient funkcji straty względem parametrów θ w kroku t,
- θ_t to wartość parametrów w kroku t.



Rysunek 1.10: Czerwonymi strzałkami zaznaczono położenie wag w przestrzeni wagowej (rys. 1.3) po 6 iteracjach wstecznej propagacji, a niebieskimi wraz z działaniem mechanizmu `momentum`. Działa on anologicznie do fizycznej zasady bezwładności. Gdy gradienty wskazują spójny kierunek podczas kolejnych iteracji, to suma ich udziałów, prowadzi do szybszych aktualizacji w tym kierunku. W obszarach o dynamicznie zmieniających się gradientach tłumi drgania, uśredniając przeciwstawne gradienty. W połączeniu z RMSprop, który adaptacyjnie skaluje kroki uczenia, metoda ta efektywnie radzi sobie z problemami zanikających gradientów oraz szumem stochastycznym, zapewniając szybszą zbieżność [53].

Ostatnia warstwa wykorzystuje funkcję logistyczną, do ograniczenia każdej z czterech wyjściowych wartości do przedziału [0; 1]. Pozwala to interpretować wyjście jako niezależne prawdopodobieństwa przynależności do każdej z czterech klas. Ostatecznym wyjściem naszego modelu, na podstawie sekwencji wejściowej, jest właśnie klasa z największą wartością prawdopodobieństwa.

1.5. Analiza statystyczna danych

Statystyczna analiza stanowiła istotny element walidacji wyników i pozwoliła na weryfikację wielu hipotez. W pracy wykorzystano zarówno metody analizy korelacji, jak i testy nieparametryczne, odpowiednie dla danych o rozkładzie odbiegającym od normalnego. Do sprawdzenia siły i kierunku potencjalnej zależności liniowej między zmiennymi stosuje się współczynnik korelacji Pearsona, który mierzy stopień zależności między dwiema zmiennymi o rozkładzie zbliżonym do normalnego [51]. Dla danych o nieliniowym związku wykorzystuje się natomiast współczynnik korelacji rang Spearmana, który jest odporny na odstępstwa od założeń parametrycznych i wykrywa monotoniczne zależności [50]. Kontrola istotności statystycznej uzyskanych współczynników opiera się na testach rangowych, które nie wymagają założeń o normalności rozkładu.

W wszystkich wykonanych testach statystycznych przyjęto poziom istotności statystycznej (α) na poziomie 0.05. Jest to najczęściej stosowany próg, który wskazuje poniżej jakiej wartości p wynik badania uznaje się za statystycznie istotny. Przyjęty próg oznacza, że istnieje 5% ryzyka popełnienia błędu pierwszego rodzaju (odrzucenia prawdziwej hipotezy zerowej). Jeśli obliczona wartość p dla danego wyniku jest mniejsza lub równa ustalonemu progowi, hipoteza zerowa jest odrzucana na rzecz hipotezy alternatywnej, co sugeruje, że badany efekt lub różnica nie jest dziełem przypadku. W przypadku testu Kruskala-Wallisa hipotezą alternatywną jest, że przynajmniej jeden z analizowanych rozkładów ma inną medianę. Natomiast w przypadku Manna-Whitney'a oznacza, że rozkłady są parami różne i mają różną lokalizację mediany.

W celu porównania rozkładów między więcej niż dwiema grupami zastosowano nieparametryczny test Kruskala-Wallisa [47], który stanowi odpowiednik jednoczynnikowej analizy wariancji. Z kolei dla porównania par wykorzystano test U Manna-Whitney'a [48]. Aby zminimalizować ryzyko wystąpienia błędu pierwszego rodzaju, wynikającego z wielokrotnego testowania, do skorygowania uzyskanych p-wartości zastosowano korektę Benjaminiego-Hochberga, która jest mocniejsza statystycznie w porównaniu do konserwatywnej poprawki Bonferroniego [49]. Pozwala ona na kontrolowanie prawdopodobieństwa zajścia hipotezy zerowej, a tym samym podobieństwa analizowanych rozkładów.

1.5.1. Estymacja jądrowej gęstości

Niektóre filtry konwolucyjne w trakcie procesu treningu mogą zostać uznane za nieprzydatne i wyzerowane (podsekcja 1.2.2). Początkowo wszystkie wagi są inicjowane metodą Kaiminga-He [12], która jest optymalna dla sieci neuronowych wykorzystujących funkcje aktywacji ReLU, biorąc pod uwagę ograniczenia dotyczące stałej wariancji w przejściach do przodu i do tyłu [54]. W trakcie procesu treningu miliony parametrów ulegają ciągłym zmianom, aby jak najlepiej dostosować się do zbioru uczącego. Ostateczny model jest jednak bardzo trudny do interpretacji lub zrozumienia przez człowieka [8].

Najłatwiejsze, zarówno w zrozumieniu, jak i dostępie, są filtry z pierwszej warstwy konwolucyjnej. Norma Frobeniusa danego jądra może posłużyć jako heurystyczne kryterium identyfikacji jego przydatności. Ponadto ich konwolucja odbywa się bezpośrednio z zakodowanymi (rys. 1.4) sekwencjami DNA. Norma Frobeniusa [59] macierzy $\mathbf{W} \in \mathbb{R}^{a \times b}$ zdefiniowana jest jako:

$$||\mathbf{W}||_{\mathrm{F}} = \sqrt{\sum_{i=1}^{a} \sum_{j=1}^{b} \mathbf{W}_{i,j}} = \mathrm{trace}(\mathbf{W}^{\mathbf{T}}\mathbf{W}) \tag{1.3}$$

Do wizualizacji i analizy rozkładu wag pierwszej warstwy konwolucyjnej, zastosowano nieparametryczną estymację jądrową gęstości (ang. *Kernel Density Estimation*, KDE) z jądrem gaussowskim [15, 14]. Metoda ta pozwoliła na aproksymację funkcji gęstości prawdopodobieństwa wyłącznie na podstawie wag, bez narzucenia założeń co do jej postaci analitycznej. Klasyczna metoda KDE skutecznie wygładza dane poprzez wykorzystanie funkcji jądrowej, a następnie normalizację wyniku w celu uzyskania estymacji gęstości.

Sprowadzając filtry do postaci wektorów $\{w_i^n\}$, tym samym ignorując ich strukturę przestrzenną, wykorzystywany przez nas estymator jądrowy gęstości w punkcie x jest postaci:

$$\hat{f}(x) = \frac{1}{n \cdot h_{\mathrm{S}} \cdot \sqrt{2\pi}} \cdot \sum_{i=1}^{n} \exp\left(-\frac{(x - w_i)^2}{2 \cdot h_{\mathrm{S}}^2}\right) \tag{1.4}$$

Gdzie $h_{\mathrm{S}} > 0$ jest parametrem wygładzenia (ang. *bandwidth*). Wybór tego parametru jest istotny, ze względu na kompromis między odchyleniem a wariancją [16]. Większa wartość prowadzi do bardziej gładkich oszacowań o niższej wariancji, ale o wyższym przesunięciu, podczas gdy mniejsze wartości powodują większą wariancję [15], [14]. W praktyce dla rozkładów jednowymiarowych wykorzystuje się estymację Silvermana:

$$h_{\mathrm{S}} = 1.06 \cdot \hat{\sigma} \cdot n^{-\frac{1}{5}}$$

Gdzie $\hat{\sigma}$ to odchylenie standardowe próbki $\{w_i\}$ zawierającej wagi istotnych filtrów z pierwszej warstwy konwolucyjnej. Reguła ta pozwala na dobre, stabilne wygładzenie przy rozmiarach analizowanych danych.

Rozdział 2

Motywacja

Analiza danych neurobiologicznych, szczególnie związanych z guzami mózgu, takimi jak glejaki, stanowi istotne wyzwanie badawcze. Głównymi przyczynami są wysoka złożoność biologiczna oraz ograniczona dostępność dużych, wysokojakościowych zbiorów danych [56]. W kontekście rozpoznawania sekwencji zawiłość. Glejak, należący do najbardziej agresywnych i śmiertelnych nowotworów mózgu, nie jest jednolitą masą komórek. W obrębie jednego guza współwystępują różne subklony komórek nowotworowych, każdy z nieco innym profilem genetycznym i epigenetycznym, co utrudnia identyfikację jednoznacznych wzorców sekwencji. Aktywność enhancerów i promotorów, która decyduje o tożsamości komórki, jest dynamicznie regulowana przez splot czynników: sekwencji DNA, modyfikacji chromatyny, obecności białek transkrypcyjnych i sygnałów z mikrośrodowiska. Jednocześnie astrocyty mogą wpływać na mikrośrodowisko guza i reakcję na terapię, co czyni je cennym przedmiotem badań w neuroonkologii [57].

2.1. Cel pracy

Głównym celem niniejszej pracy licencjackiej jest analiza porównawcza właściwości dwóch konwolucyjnych sieci neuronowych – `alt_last` oraz `high4_last` – w kontekście predykcji aktywności czterech klas obszarów regulatorowych DNA (aktywne/nieaktywne promotory oraz aktywne/nieaktywne enhancery) w danych genomowych nowotworów mózgu oraz astrocytów. Oba modele charakteryzują się identyczną architekturą typu *Custom*, wzorowaną na *Basset network* [1], ale różnią się charakterystyką zbiorów treningowych:

- Model `alt_last` został wytrenowany wyłącznie na zestawie *Alt*, zawierającym dane pochodzące z komórek glejaków (40 000 sekwencji na klasę).
- Model `high4_last` korzysta z rozszerzonego zbioru danych, który zawiera zarówno sekwencje z glejaków, jak i dodatkowe przykłady pochodzące od sztucznie zróżnicowanych do astrocytów komórek iAstrocytes (łącznie *circa* 51 000 sekwencji na klasę).

Wykorzystanie modelu w uczeniu transferowym (podsekcja 1.3.2) na sekwencjach różnego rodzaju może potencjalnie poprawić jakość i odporność jej predykcji. Modele wytrenowane na danych pochodzących z glejaków mogą nauczyć się unikalnych reprezentacji przestrzennych istotnych dla tkanki mózgowej, które mogą być transferowane do zadań związanych z astrocytami i *vice versa*. Sieć neuronowa będzie miała możliwość poznania szerszego kontekstu biologicznego, bez dodatkowego adnotowania sekwencji. Detekcja, a następnie synteza unikalnych informacji, które są zawarte w tych dwóch zbiorach, może pozwolić na znalezienie zupełnie nowych cech.

2.2. Znaczenie pierwszej warstwy konwolucyjnej

Pierwsza warstwa konwolucyjna odgrywa kluczową rolę w sieciach operujących na sekwencjach DNA, ponieważ odpowiada za ekstrakcję podstawowych motywów sekwencyjnych. Przykładem mogą być miejsca wiązania czynników transkrypcyjnych czy specyficzne wzorce nukleotydowe (podsekcja 1.1.3) [4]. W odróżnieniu od wyższych warstw, które kodują abstrakcyjne kombinacje cech, filtry pierwszej warstwy działają jako detektory lokalnych wzorców w surowych danych wejściowych. W kontekście genomowym modyfikacje wag w tej warstwie bezpośrednio przekładają się na zdolność sieci do rozpoznawania ewolucyjnie zachowanych motywów lub nowotworowych aberracji.

W niniejszej pracy największą uwagę poświęcamy filtrom pierwszej warstwy ze względu na ich zasadnicze znaczenie dla interpretacji biologicznej, adaptacyjności modelu oraz stabilności transferu wiedzy. Filtry tej warstwy umożliwiają bezpośrednie powiązanie wykrywanych wzorów z sekwencjami DNA, co jest niezbędne do zrozumienia mechanizmów regulacyjnych na poziomie molekularnym. Dodatkowo podczas procesu dotrenowywania filtry pierwszej warstwy decydują o tym, jak sygnał jest propagowany do głębszych warstw sieci.

Jako prostą miarę siły sygnału filtra zaproponowano normę Frobeniusa jego wag (równanie 1.3). Jej bardzo niska wartość sugeruje, że filtr aktywuje się niezwykle rzadko, praktycznie nie przekazując sygnału dalej. Należy jednak podkreślić, że norma Frobeniusa jest jedynie prostą miarą energii. Filtry o niższych wartościach nadal mogą kodować istotne, lecz subtelne motywy, jak na przykład kontrast pomiędzy kanałami. Duża norma przypuszczalnie odpowiada za generalny motyw, a nie fragment silnie specyficzny dla danej klasy [7].

Analiza wstępna wykazała, że jedynie 46 z 300 filtrów w pierwszej warstwie posiada wagi numerycznie różne od zera, co sugeruje, że sieć podczas klasyfikacji koncentruje się jedynie na wąskim zestawie kluczowych motywów sekwencyjnych. W związku z tym modyfikacja tylko tego niewielkiego podzbioru filtrów w trakcie dotrenowywania modelu na danych z astrocystów pozwala minimalizować ryzyko utraty informacji specyficznych dla danych z komórek glejaków, zachowując przy tym zdolność do generalizacji na całkowicie nowy rodzaj danych [5].

2.3. Pytania badawcze

W oparciu o powyższe cele, kluczowe pytania badawcze dotyczą trzech głównych obszarów: adaptacji modelu do nowych danych pochodzących ze zdrowych komórek glejowych, analizy różnic strukturalnych między filtrami pierwszej warstwy obu sieci oraz oceny wpływu modyfikacji tych filtrów na ostateczną predykcję modelu. Dokładniej, celem iteracyjnego dotrenowania modelu `alt_last` na zbiorze *HighSignal*, który pierwotnie współtworzył zbiór treningowy dla modelu `high4_last` jest:

1. Porównanie skuteczności predykcyjnej: sprawdzenie, czy ekspozycja modelu na sekwencje pochodzące z komórek iAstrocytes pozwala osiągnąć efektywność predykcyjną porównywalną do modelu trenowanego *ab initio* na rozszerzonym zbiorze danych.

2. Dynamika adaptacji filtrów: analiza zmian wag filtrów pierwszej warstwy konwolucyjnej podczas transferu wiedzy między kontekstem nowotworowym (glejaki) a zdrowych komórek glejowych (astrocyty).

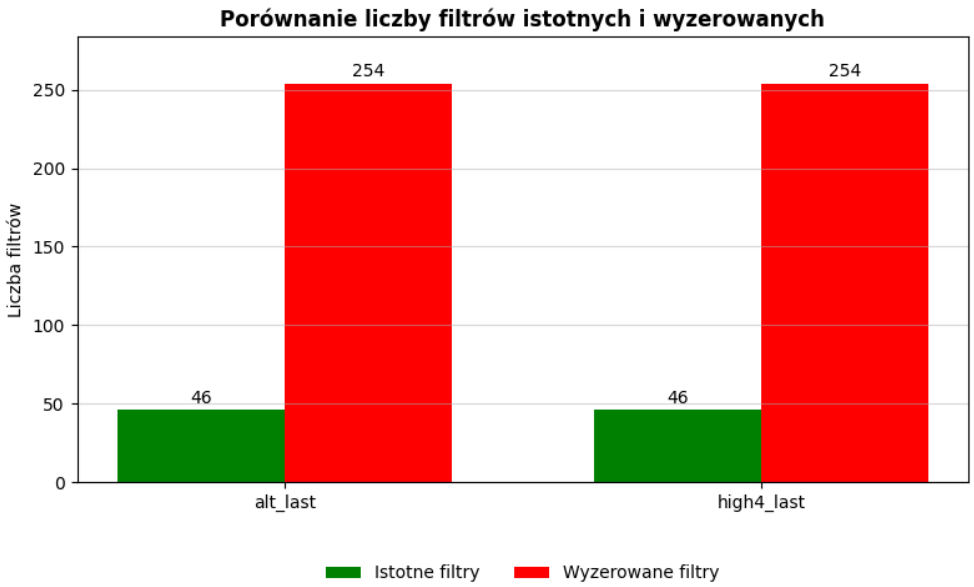
3. Wpływ modyfikacji filtrów na predykcję: ocena, w jaki sposób zmiany w pierwszej warstwie konwolucyjnej (np. zmiana wag lub pozycji filtrów) wpływają na przewidywane przez model wyniki, zarówno na oryginalnym zbiorze *Alt*, jak i na zbiorze *HighSignal*.

Rozdział 3

Wyniki

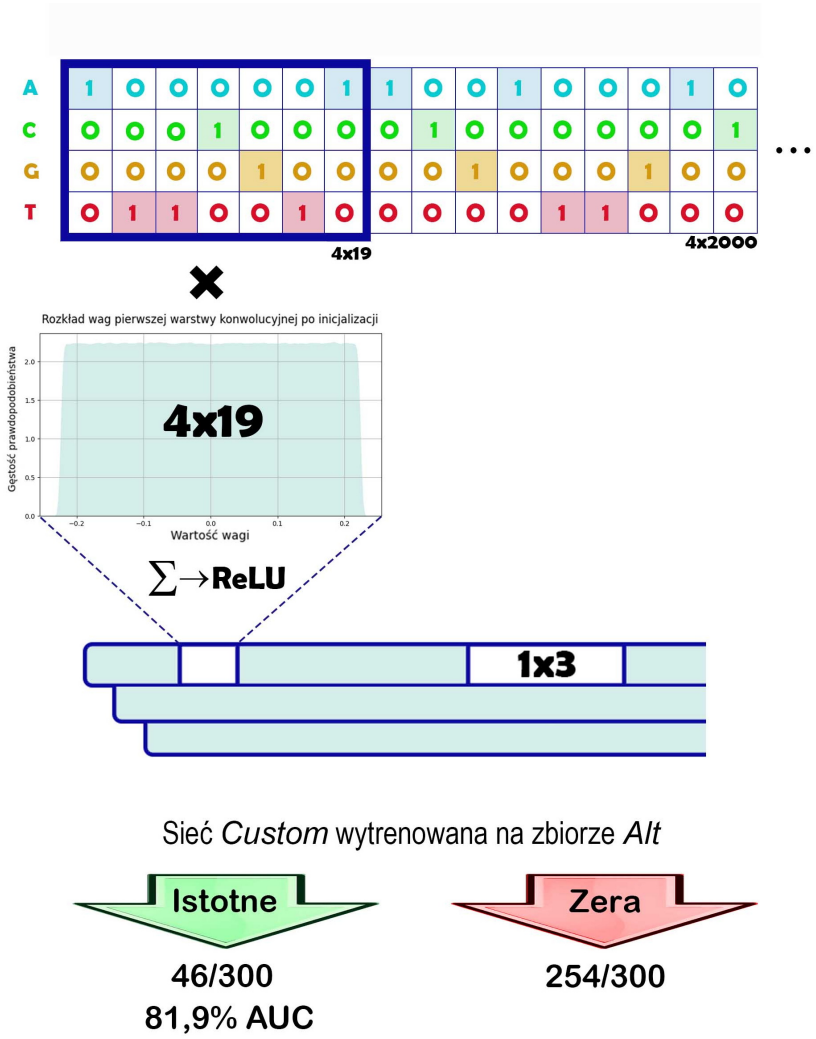
Wszystkie eksperymenty dotrenowania modelu *alt_last* na zbiorze *HighSignal* powtórzono wielokrotnie, aby zapewnić statystyczną istotność wyników. Podział sekwencji na podzbiory treningowe, walidacyjne i testowe (według przynależności do chromosomów) był identyczny, jak opisano w podsekcji 1.3.2 dla wszystkich powtórzeń. W sekwencjach zawierających poniżej stu nieznanych nukleotydów, oznaczonych jako `N`, zastąpiono je losowymi nukleotydami. Każdy uzyskany model, zarówno przed, jak i po procesie dotrenowania, był ewaluowany oddzielnie na zbiorach testowych pochodzących z glejaków i astrocytów. Ponieważ klasy są wewnątrz każdego zbioru rozważane osobno i mają zbliżoną liczbę przykładów, nie stosowano specjalnych metod do wyrównywania nierównomierności klas (podsekcja 1.2.3).

3.1. Porównanie modeli



Rysunek 3.1: Rozkład filtrów pierwszej warstwy konwolucyjnej w modelach *alt_last* oraz *high4_last* na podstawie normy Frobeniusa. Niezerowe wagi są silnie skoncentrowane wokół nielicznych jąder konwolucyjnych. Zapewniają one stabilną i silną zdolność do wykrywania motywów DNA.

Początkowa analiza wykazała, że po wytrenowaniu na zbiorze *Alt* model *alt_last* zachował tylko 46 z 300 filtrów pierwszej warstwy z wagami istotnie różnymi od zera. Analogiczną strukturę obserwujemy w modelu *high4_last* (rysunki 3.2 i 3.1), co potwierdza hipotezę, że w ekstrakcji cech dla danych genomowych istotnych jest stosunkowo niewiele motywów. Filtry, których wszystkie wagi numerycznie odpowiadają zeru, nie uczestniczą w żaden sposób w propagacji sygnału, niezależnie od przypisanych im wartości początkowych. Taki efekt jest konsekwencją działania regularyzacji typu *Weight Decay* (regularyzacji L2), która podczas treningu systematycznie redukuje normę wag filtrów uznawanych za nieistotne dla minimalizacji funkcji kosztu [13].

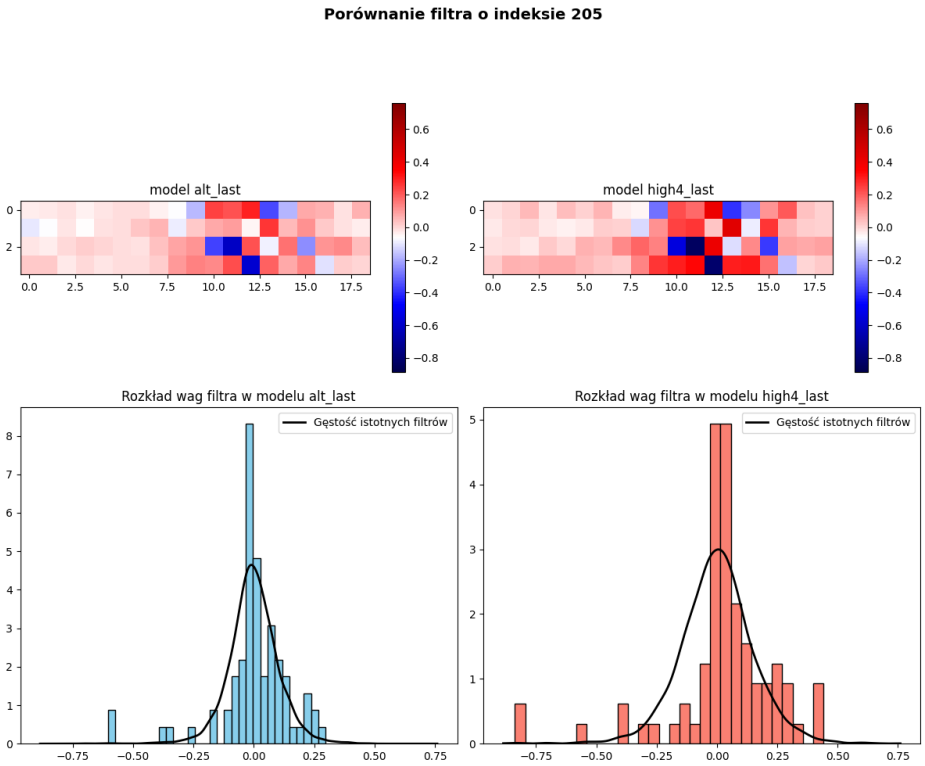


Rysunek 3.2: Początkowa inicjalizacja wag filtrów konwolucyjnych w modelu *alt_last* wykorzystuję metodę Kaiminga-He dla rozkładu jednostajnego [12]. Tymczasem po procesie treningowym na zbiorze *Alt* wiele z nich zostało zdegenerowanych do macierzy zerowych.

Zbiór 300 filtrów poddano analizie porównawczej opartej na normie Frobeniusa, ze względu na fakt, że wewnątrz istotnych filtrów średnia i mediana wag w przybliżeniu do części dziesiętnych wynosi zero. Uzyskane wyniki potwierdzają rezultaty otrzymane w pracy [2]. Dokładniejsze analizy norm oraz wartości wag poszczególnych filtrów ujawniły, że:

- W obu modelach występuję identyczna liczba 46 aktywnych filtrów pierwszej warstwy. Każdy z tych filtrów spełnia warunek $||\mathbf{W}||_{\mathrm{F}} \geq 0.50$, przy czym wszystkie jego składowe wagowe mają wartość bezwzględną $> 10^{-5}$.
- Pozostałe 254 filtry mają wartości wag bliskie zeru (rzędu 10^{-42}), co czyni je nieaktywnymi w procesie klasyfikacji.
- Indeksy aktywnych filtrów w obu modelach są ze sobą w pełni zgodne.
- Średnia norma Frobeniusa aktywnych filtrów w modelu *high4_last* wzrosła o około 45% w porównaniu z modelem *alt_last*. Największą normę we wszystkich przypadkach wykazywał filtr o indeksie 205, którego wartość normy była największa spośród wszystkich badanych filtrów.
- Wewnętrzna kolejność filtrów posortowanych malejąco według normy była zbliżona w obu modelach, co świadczy o zachowaniu struktury kluczowych motywów.

Powyższa strukturalna koincydencja wskazuje, że obie sieci identyfikują te same fundamentalne motywy sekwencyjne jako kluczowe dla klasyfikacji regionów regulatorowych. Jednakże rozbieżności sugerują, że rozszerzenie zbioru treningowego o dane ze zbioru *HighSignal* prowadzi do wzmocnienia odpowiedzi na specyficzne motywy nukleotydowe (rys. 3.3), przy jednoczesnym zachowaniu ekwiwalentnej struktury filtrów.



Rysunek 3.3: Porównanie wag wewnątrz filtra o indeksie 205 między analizowanymi modelami ujawnia zbliżoną strukturę wartości na odpowiadających sobie pozycjach. Model `high4_last`, charakteryzuje się jednak większą wariancją wag oraz obecnością cięższych ogonów w ich rozkładzie, co sugeruje silniejszą adaptację sieci do specyficznych wzorców. Fakt ten przekłada się bezpośrednio na zwiększoną normę Frobeniusa poszczególnych filtrów.

W celu weryfikacji funkcjonalnej roli filtrów przeprowadzono eksperyment dezaktywacji filtrów pierwszej warstwy. Dezaktywacja pojedynczego istotnego filtra, polegająca na wyzerowaniu jego wag konwolucyjnych oraz parametrów normalizacji wsadowej, systematycznie pogarszała zdolności predykcyjne modelu. Ponadto całkowite wyzerowanie wszystkich aktywnych filtrów pierwszej warstwy sprowadzało wynik modelu do poziomu klasyfikatora losowego (rys. 1.4), co potwierdza ich podstawową rolę w ekstrakcji cech. Z kolei dezaktywacja filtrów, które były początkowo zerowe, nie wpływała w żaden sposób na jakość klasyfikacji, ponieważ ich sygnał w pierwszej warstwie praktycznie nie występował. Wyniki te świadczą, że skuteczna predykcja modeli opiera się wyłącznie na wąskim podzbiorze filtrów pierwszej warstwy; identyczność aktywnych indeksów między modelami sugeruje istnienie uniwersalnych mechanizmów ekstrakcji cech dla regionów regulatorowych układu nerwowego.

Rezultaty świadczą o tym, że skuteczna predykcja modeli opiera się wyłącznie na wąskim podzbiorze filtrów pierwszej warstwy konwolucyjnej. Identyczność indeksów tych, które są aktywne, między modelami sugeruje istnienie uniwersalnych mechanizmów ekstrakcji cech dla regionów regulatorowych układu nerwowego. Wyniki pozostają spójne dla obu zbiorów testowych, potwierdzając ogólny charakter zaobserwowanego mechanizmu.

3.2. Dotrenowywanie modelu *alt*

Każde dotrenowanie odbywało się na tym samym podzbiorze treningowym, zbioru *HighSignal*. Selekcja optymalnego modelu opierała się na maksymalizacji wartości wskaźnika AUC osiąganej dla wszystkich klas na zbiorze walidacyjnym, zgodnie z metodologią opisaną w podsekcji 1.3.2. Wszystkie metryki ewaluacyjne podawane są dla chromosomów testowych sekwencji odpowiedniego pochodzenia.

3.2.1. Optymalizacja parametrów uczenia

Fundamentalnym aspektem procesu dotrenowania modelu `alt_last` na nowym zbiorze danych było dobranie optymalnej wartości współczynnika uczenia. Zbyt wysoka wartość może poskutkować całkowitą aktualizacją wag, co w konsekwencji prowadzi do katastroficznego zapominania. Natomiast zbyt niska uniemożliwia efektywną adaptację do nowych danych.

- LR = 0.01 (wartość oryginalna) powodował degradację modelu do klasyfikatora losowego na przestrzeni kilku epok. Jest to klasyczny przejaw katastroficznego zapominania w transferowym uczeniu.
- LR = 0.001 (10-krotnie mniejszy) zapewnił stabilną konwergencję,
- LR = 0.0001 skutkował wydłużeniem procesu uczenia bez poprawy metryk.

Ze względu na wielkość zbioru *HighSignal*, który jest czterokrotnie mniejszy niż zbiór *Alt*, ograniczono maksymalną liczbę epok do 200. Pozostałe hiperparametry pozostawiono bez zmian względem oryginalnej konfiguracji, ze względu na niewielkie różnice w analizowanych metrykach [3]. Zmniejszenie szybkości uczenia, które reguluje wielkość aktualizacji parametrów, zapewniło równowagę między adaptacją a retencją wiedzy.

3.2.2. Strategia eksperymentalna

Dotrenowany model oznaczono jako *alt_high_0*, gdzie sufiks '0' sygnalizuje, że przed etapem transferu wiedzy nie zastosowano żadnych modyfikacji w strukturze filtrów konwolucyjnych pierwszej warstwy. Selekcja optymalnego modelu opierała się na maksymalizacji wartości wskaźnika AUC osiąganej dla wszystkich klas na zbiorze walidacyjnym zbioru *HighSignal*. Ewaluację wyników wykonano na chromosomach testowych, niezależnych od treningowych, osobno dla sekwencji glejaków i astrocytów.

Model	PA			NA			PI			NI		
	Cz.	Sw.	AUC	Cz.	Sw.	AUC	Cz.	Sw.	AUC	Cz.	Sw.	AUC
alt_last	0.86	0.89	0.92	0.56	0.95	0.79	0.55	0.85	0.74	0.71	0.88	0.83
alt_high_0	0.78	0.90	0.85	0.49	0.92	0.74	0.57	0.78	0.71	0.63	0.86	0.79
high4_last	0.84	0.86	0.90	0.62	0.85	0.77	0.37	0.90	0.66	0.60	0.88	0.77

Tabela 3.1: Wyniki modeli dla czterech klas zbioru *Alt*: Promotor Aktywny - PA, Niepromotor Aktywny - NA, Promotor Nieaktywny - PI, Niepromotor Nieaktywny - NI. Najwyższy wynik AUC dla każdej z nich osiąga model *alt_last*.

Bazowy model *alt_last* dominuje w każdej klasie, osiągając najwyższe AUC, co jednoznacznie świadczy o dobrym dopasowaniu architektury do danych pochodzących z nowotworów mózgu. Jednocześnie model dotrenowany wskazuje spadek wydajności dla każdej klasy (Δ AUC: -0.07 dla PA; -0.05 dla NA; -0.03 dla PI; -0.04 dla NI). Oznacza to, że adaptacja wag do nowych sekwencji powoduje utratę wiedzy o danych źródłowych. Natomiast model *high4_last* osiąga niższe wyniki niż *alt_last* w klasach NA i PI, co sugeruje, że trening na połączonym zbiorze może prowadzić do kompromisów w reprezentacjach cech specyficznych dla glejaków i astrocytów.

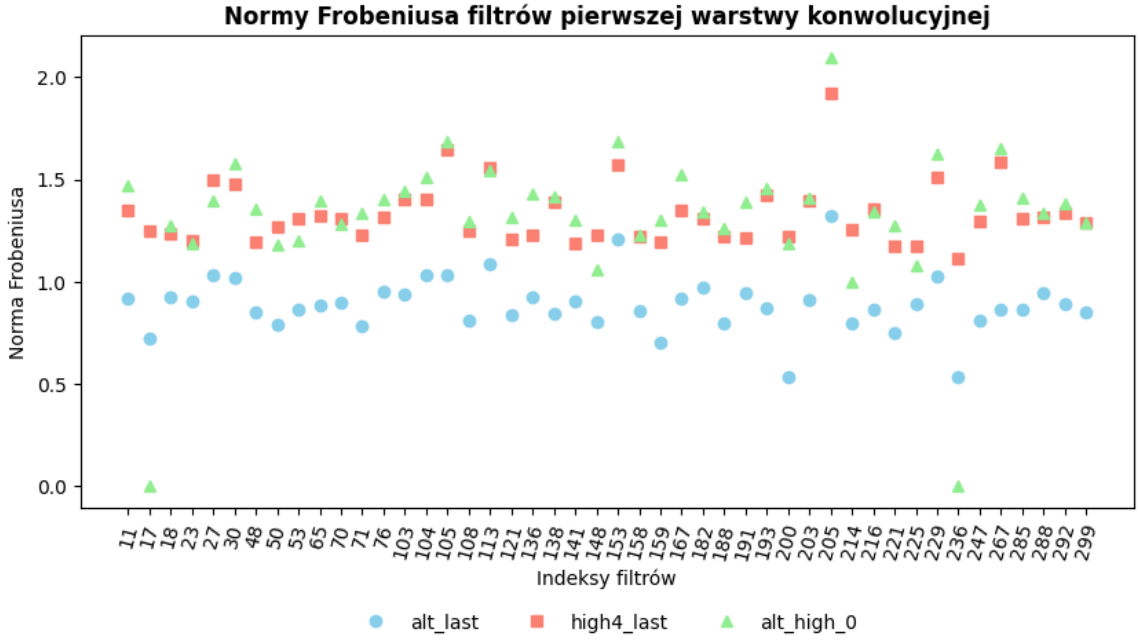
Model	PA			NA			PI			NI		
	Cz.	Sw.	AUC	Cz.	Sw.	AUC	Cz.	Sw.	AUC	Cz.	Sw.	AUC
alt_last	0.76	0.88	0.86	0.41	0.91	0.69	0.55	0.83	0.73	0.61	0.84	0.75
alt_high_0	0.75	0.90	0.87	0.79	0.88	0.88	0.58	0.89	0.77	0.66	0.91	0.84
high4_last	0.81	0.87	0.88	0.87	0.83	0.90	0.40	0.95	0.71	0.67	0.92	0.83

Tabela 3.2: Wyniki modeli dla wszystkich klas zbioru *HighSignal*. Wartości AUC modeli *alt_high_0* oraz *high4_last* są do siebie bardzo zbliżone.

Wyniki przedstawione w tabelach 3.1 i 3.2 ukazują istotną dysproporcję między czułością (Cz) a swoistością (Sw) w klasach, gdzie metryka AUC jest najniższa. Zjawisko to jest najbardziej widoczne w przypadku modelu *high4_last* wobec sekwencji pochodzenia glejakowego oraz *alt_last* na zbiorze *HighSignal*. Sieci trenowane pojedynczo unikają fałszywie pozytywnych klasyfikacji kosztem wysokiej liczby przypadków fałszywie negatywnych. Taka dysproporcja często wynika z nieadekwatnej ekstrakcji cech. Poprawa swoistości w jednej klasie odbywa się kosztem czułości w innej, ze względu na współdzielenie konwolucyjnych wag. Efekt ten jest jednak zauważalnie słabszy w przypadku dotrenowanego modelu.

W tabeli 3.2 można zaobserwować, że model *alt_high_0* osiąga AUC na poziomie 0.88 dla klasy aktywnych enhancerów, co stanowi wzrost o 0.19 względem *alt_last* i sytuuje go blisko wyniku *high4_last*. To dowód skuteczności dotrenowywania w adaptacji do nowych motywów w komórkach iAstrocytes. W klasie nieaktywnych enhancerów uzyskuje najlepszy wynik, przewyższając dwa pozostałe modele. Sugeruje to istnienie synergii między wiedzą z komórek nowotworowych a zdrowymi komórkami glejowymi.

Wszystkie modele mają natomiast trudność z klasą PI, co może wskazywać na różnorodną naturę nieaktywnych promotorów, bez istnienia wyraźnych markerów epigenetycznych. Obserwujemy wyraźny kompromis, w którym spadkowi predykcji na zbiorze testowym *Alt* towarzyszy jednocześnie wzrost na nowo dotrenowanych sekwencjach. Otrzymana korzyść jest jednak średnio $2,6$ razy większa niż strata, potwierdzając tym samym efektywność strategii dotrenowania.

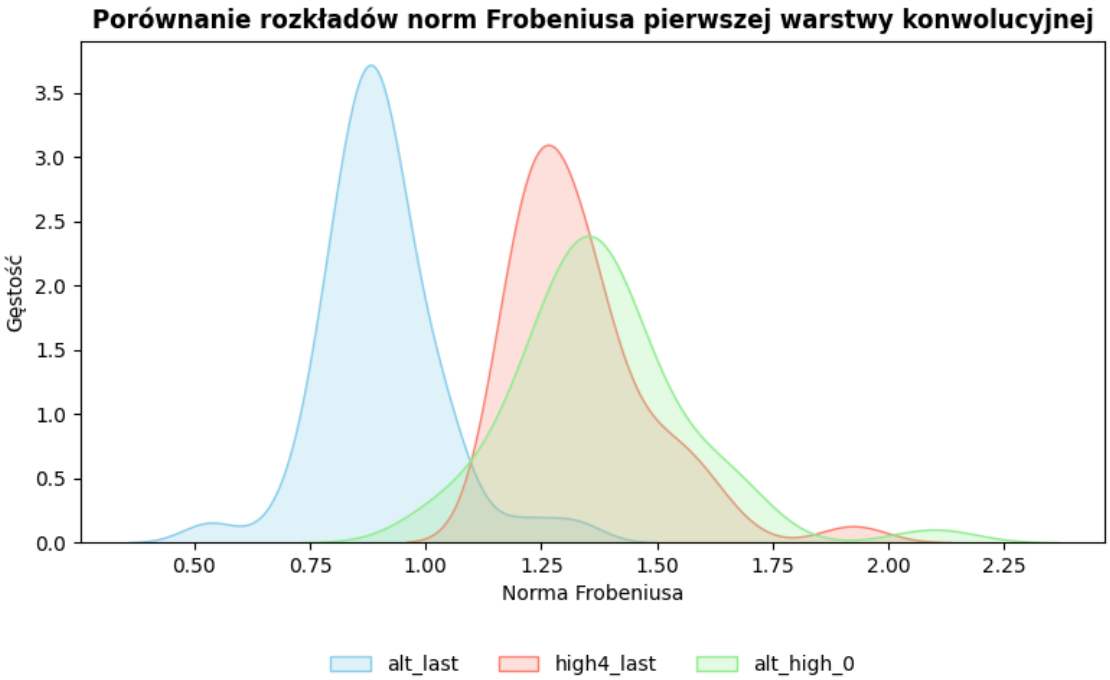


Rysunek 3.4: Porównanie norm Frobeniusa filtrów o wyszczególnionych indeksach z pierwszej warstwy konwolucyjnej. Wagi dwóch z czterdziestu sześciu filtrów po procesie dotrenowania zostałe wyzerowane. Od początku charekteryzowały się one niskimi wartościami (236 najniższą, a 17 czwartą najmniejszą). Normy Frobeniusa pozostałych wzrosły, zbliżając się do wyników charakterystycznych dla modelu *high4_last*. W 25 z 44 pozostałych istotnych filtrów, norma jest większa niż w przypadku modelu wytrenowanego *ab initio* na obu zbiorach danych.

Jednocześnie, podczas procesu douczenia modelu na nowym zbiorze danych, zaobserwowano istotne zmiany w rozkładzie norm Frobeniusa filtrów pierwszej warstwy konwolucyjnej. Już na początkowym etapie dwa filtry o indeksach 17 i 236, które charakteryzowały się od początku niską normą, uległy całkowitej degradacji, osiągając wartości bliskie zeru. Jednocześnie norma pozostałych 44 filtrów zwiększyła się średnio o $0,48$ ($\sigma = 0,127$) zbliżając ich rozkład do modelu *high4_last* (rys. 3.4).

Wyzerowanie filtrów może świadczyć o ich niskiej użyteczności w kontekście sekwencji astrocytów. Zastosowanie techniki regularyzacyjnej *Weight Decay* prowadzi do optymalizacji przestrzeni cech. Skutkiem jest utrata części jednostek, co zostaje zrekompensowane przez wzmocnienie pozostałych. Sieć zamiast dodania nowych filtrów wzmacnia istniejące ścieżki propagacji sygnału, eliminując cechy o niskiej użyteczności względem nowych danych. Przedstawione zmiany wskazują zatem na proces generalizacji modelu w celu przyporządkowania sekwencji do jednej z czterech klas. Strategia ta jest zgodna z zasadą Brzytwy Ockhama, gdzie minimalizacja funkcji kosztu prowadzi do zmniejszenia złożoności, przy jednoczesnym zachowaniu jak najlepszych zdolności predykcyjnych.

W celu scharakteryzowania zmian rozkładów norm Frobeniusa pomiędzy modelami *alt_ last*, *high4_ last* i *alt_ high_ 0*, zastosowano nieparametryczną estymację gęstości jądrowej (podsekcja 1.5.1). Jak pokazano na rysunku 3.5, rozkłady wykazują znaczące podobieństwo w kształcie, choć różnią się między sobą parametrami rozkładu. Wszystkie cechują się grubszymi ogonami niż rozkład normalny, jednocześnie zachowując prawostronną skośność. Filtry modeli trenowanych na zbiorze *HighSignal* posiadają większą wariancję oraz medianę. Dotrenowanie oryginalnego modelu spowodowało wzrost mediany norm istotnych filtrów z 0.89 do 1.365, tym samym zwiększając się o 5% względem wytrenowanego modelu *ab initio* na obu zbiorach. Wzajemna entropia (równanie 1.1) w porównaniu do rozkładu norm modelu *high4_ last* również się zmniejszyła, z 1.69 do 0.886.



Rysunek 3.5: Porównanie rozkładów norm Frobeniusa istotnych filtrów, uzyskanych metodą KDE dla trzech modeli. Kolorem niebieskim oznaczono: *alt_ last*, czerwonym: *high4_ last*, zielonym: *alt_ high_ 0*. Wszystkie rozkłady charakteryzują się ciężkim ogonem oraz przesunięciem szczytu na lewa stronę (są leptokurtyczne i skośne prawostronnie), ale różnią się między sobą medianą, co znajduje uzasadnienie w różnych parametrach ich rozkładu.

Analizę uzupełniono o następujące testy statystyczne (podsekcja 1.5):

- **Test Manna-Whitney'a**: Dla par modeli *alt_ last* vs *alt_ high_ 0* uzyskano p-wartość = 0.117, co nie daje podstaw do odrzucenia hipotezy zerowej o jednakowych rozkładach.
- **Korelacja Spearmana**: Wartość 0.7921 (p-wartość = 1.51×10^{-10}) potwierdza monotoniczną zależność między normami [50].

3.3. Zamiana pozycją istotnych z wyzerowanymi filtrami

Kolejnym eksperymentem było sprawdzenie, czy filtry pierwszej warstwy mające początkowe zerowe wagi mogą mieć znaczenie dla adaptacji sieci. Ze względu na właściwości konwolucji, sieć powinna być zdolna do adaptacji również w przypadku zmiany miejsca działania istotnych filtrów. Dlatego przeprowadzono symulację, w której pozycje wag wszystkich filtrów pierwszej warstwy mieszano losowo pomiędzy istotnymi i wyzerowanymi filtrami. Innymi słowy, każdy nieaktywny filtr mógł z jednakowym prawdopodobieństwem zamienić swoje wagi z innym, istotnym filtrem. Tak zmodyfikowane modele poddano ponownemu treningowi i ewaluacji.

$\mathbb{P}_{\textbf{zamiany}}$	$\mathbf{MCC}_{\text{Alt}}$				$\mathbf{MCC}_{\text{HighSignal}}$				$\|\mathbf{W}_{\boldsymbol{F}}\|$		
	PA	**NA**	**PI**	**NI**	**PA**	**NA**	**PI**	**NI**	**Istotne**	**Max**	**Min**
0.0	0.60	0.45	0.33	0.49	0.66	0.66	0.48	0.55	44	2.10	1.00
0.1	0.59	0.44	0.32	0.49	0.65	0.65	0.47	0.58	42	1.64	0.98
0.2	0.63	0.44	0.34	0.48	0.66	0.66	0.48	0.60	41	1.63	0.23
0.3	0.60	0.42	0.32	0.48	0.66	0.67	0.47	0.57	32	1.63	1.04
0.4	0.62	0.43	0.32	0.47	0.65	0.67	0.48	0.58	31	1.63	0.90
0.5	0.63	0.40	0.33	0.46	0.64	0.64	0.46	0.57	27	1.69	1.04
0.6	0.63	0.44	0.42	0.50	0.66	0.65	0.47	0.56	31	1.80	1.22
0.7	0.62	0.40	0.30	0.47	0.64	0.63	0.45	0.58	24	1.86	1.30
0.8	0.67	0.38	0.30	0.46	0.66	0.62	0.40	0.56	23	1.58	1.15
0.9	0.61	0.40	0.30	0.46	0.65	0.63	0.46	0.57	24	1.81	1.10
1.0	0.64	0.39	0.28	0.45	0.65	0.60	0.41	0.53	21	1.69	1.29

Tabela 3.3: Współczynniki korelacji Matthews'a, dla modeli z zamienionymi filtrami. Zawarto także informację o liczbie istotnych filtrów oraz granicznych wartościach normy w obrębie modelu. Douczanie pozwala zaobserwować trend do zerowania wag, które na innej pozycji były aktywnie wykorzystywane. Nawet dwukrotnie mniejsza ilość wykrywanych motywów, nie pogarsza znacząco predykcji.

W trakcie treningu wszystkie wagi mogą brać udział w uczeniu dzięki algorytmowi wstecznej propagacji (podsekcja 1.2.2). Dzięki właściwościom konwolucji nawet radykalna zmiana pozycji wag (kompletna zamiana pozycji wszystkich aktywnych filtrów) pozwoliła na ponowną adaptację sieci do nowych sekwencji przy zastosowanych parametrach uczenia. Oznacza to, że sieć była w stanie nauczyć się *de novo* wykrywania wzorców nawet po przetasowaniu wag.

Wyniki przedstawiono w tabeli 3.3 - zawiera ona wartości współczynników korelacji Matthews'a (MCC) dla wszystkich klas przy różnych prawdopodobieństwach zamiany pozycji filtrów (od 0,0 do 1,0) dla obu podzbiorów testowych. Zauważono, że transferowe uczenie w tych warunkach wykazuje tendencję do ponownego zerowania wag tych filtrów, które uprzednio były aktywne na innych pozycjach. Nawet pozostawienie jedynie 21 istotnych filtrów nie pogorszyło znacząco ogólnej jakości predykcji.

Zdarzają się nawet okazjonalne przypadki, kiedy tak sztucznie zmieniony model dokonuje klasyfikacji dla niektórych klas lepiej, niż model *alt_ high_ 0*. Współtowarzyszy temu zazwyczaj pogorszenie predykcji dla pozostałych etykiet. Efekt ten jest dostrzegalny w tabeli 3.3 od czterdziestoprocentowej szansy na zmianę pozycji wag. Wzrost prawidłowych przewidywań sekwencji zawierających aktywne promotory jest silnie skorelowany ze spadkiem *MCC* dla pozostałych klas. Niezależnie od porównywanego zbioru, modele najgorzej radzą sobie z klasą PI.

Generalnie, wprowadzanie szumu poprzez przetasowanie wag pierwszej warstwy obniżało odchylenie standardowe wartości norm Frobeniusa wszystkich filtrów. Zakres między maksymalnymi a minimalnymi wartościami zmalał. Normy te pozostały nieco wyższe niż w modelu *alt_ last*, ale różnica między największą a najmniejszą wartością wyraźnie się zmniejszyła. Wyjątkiem jest model, w którym filtry zamieniano z P_zamiany = 0.2. Zawiera on dokładnie jeden filtr o wyjątkowo niewielkiej normie w stosunku do wszystkich wcześniej analizowanych. Niektóre wagi w nim zawarte są o dwa, trzy rzędy wielkości mniejsze od pozostałych (rys. 3.3). Pomimo zaistnienia tej anomalii, model dokonuje średnio najtrafniejszej predykcji.

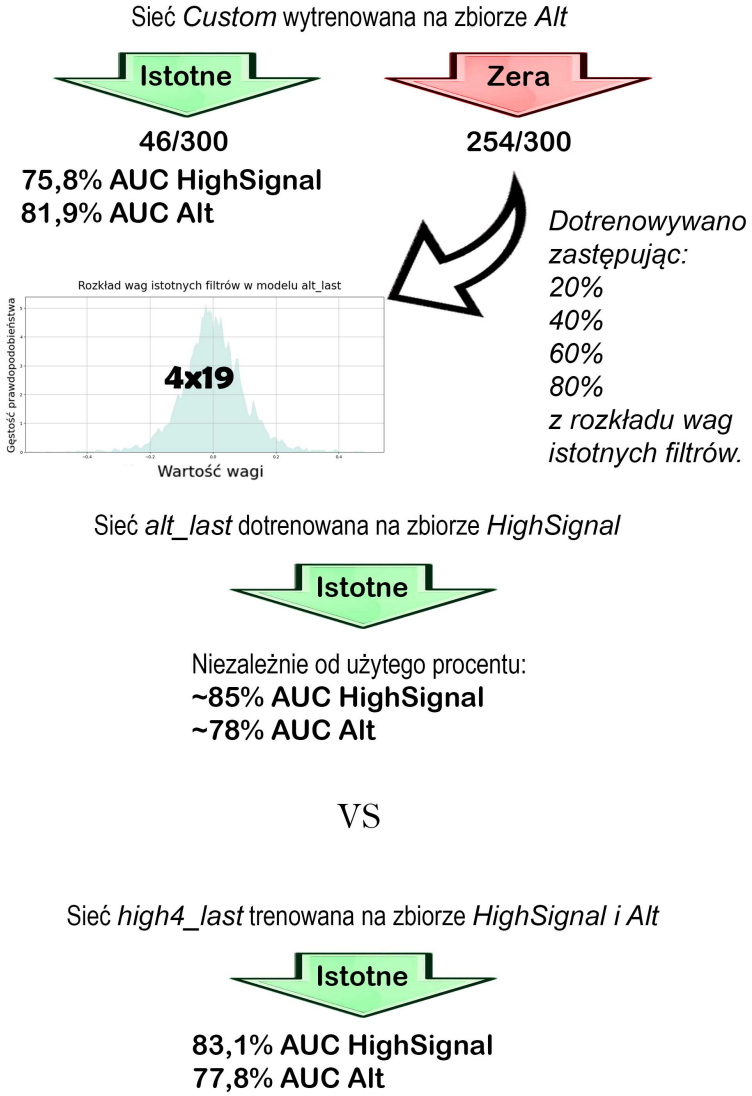
Podsumowując, wprowadzanie losowej zamiany wag filtrów pierwszej warstwy konwolucyjnej pokazało, że sieć potrafi zaadaptować się do takich modyfikacji. Nawet znacznie mniejsza liczba aktywnych filtrów nie musi drastycznie obniżać predykcji. Modele uczyły się wykorzystywać dostępne filtry, a efekt kompensacji sprawiał, że ogólna sprawność klasyfikacji pozostała zbliżona.

3.4. Probabilistyczna aktywacja filtrów

Po stwierdzeniu, że losowa zamiana wag filtrów może poprawić zdolności adaptacyjne modelu podczas dotrenowania na zbiorze *HighSignal*, podjęto próbę stworzenia bardziej uporządkowanego mechanizmu reaktywacji wyzerowanych filtrów. Początkowo, metodą bisekcji, próbowano deterministycznie dobrać procent nowo aktywowanych filtrów, ale ze względu na losowość inicjalizacji wag i nieliniowe interakcje między filtrami metoda ta okazała się niestabilna i nie pozwalała na jednoznaczne określenie optymalnego odsetka.

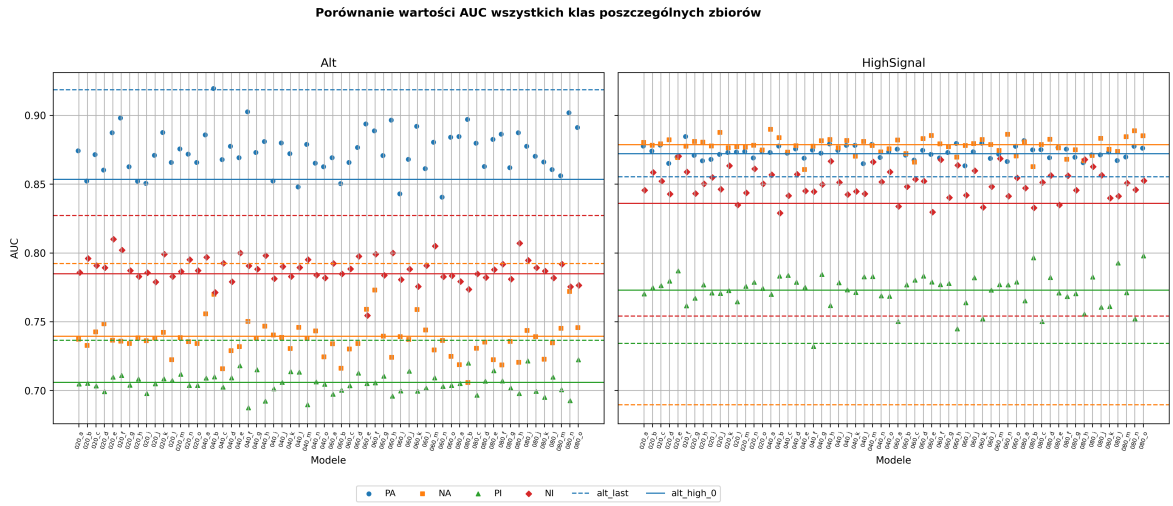
W związku z tym zastosowano podejście probabilistyczne inspirowane gęstością rozkładu wag filtrów w modelu *alt_ last*. Rozkład ten oszacowano metodą KDE na podstawie wag istotnych filtrów oryginalnej sieci. Nowe filtry inicjalizowano losowo, próbkując ich wagi z tego rozkładu. Dzięki temu zachowano statystyczne podobieństwo do już istniejących, istotnych filtrów, a jednocześnie wprowadzono element stochastycznej eksploracji przestrzeni wag. Taka metoda inicjalizacji pozwoliła na bardziej koherentną reaktywację wyzerowanych filtrów podczas dotrenowywania.

Każdy zerowy filtr przed douczeniem na zbiorze *HighSignal* był reaktywowany z określonym prawdopodobieństwem $\mathbf{R}$ w obrębie pojedynczego modelu. Eksperyment przeprowadzono dla czterech różnych wartości: $\mathbf{R} \in \{0.20;\ 0.40;\ 0.60;\ 0.80\}$. Dla każdej z nich wykonano 15 niezależnych powtórzeń, uzyskując łącznie 60 różnych modeli, co pozwoliło na obiektywną ocenę stabilności zaproponowanej metody.



Rysunek 3.6: Schemat procesu reaktywacji i dotrenowania modeli. Preinicjacyjny etap losowania wag metodą KDE prowadzi do nieznacznej poprawy metryk na obu zbiorach danych względem modelu *alt_ high_ 0*.

Na podstawie rysunku 3.7 można zaobserwować, że skuteczność predykcji modeli z reaktywowanymi filtrami jest zbliżona do modelu *alt_ high_ 0*. Zdarzają się jednak przypadki, że douczony w ten sposób model poprawia metryki na całym zbiorze *HighSignal*. Nie występuje jednak ani jeden przypadek, w którym model jest lepszy dla wszystkich klas, niezależnie od rozpatrywanego zbioru sekwencji. Uśredniając wszystkie wyniki, uzyskujemy nawet minimalnie lepsze rezultaty, niezależnie od pochodzenia sekwencji. Dla zbioru *Alt* dotrenowane modele najlepiej radzą sobie z wykrywaniem aktywnych promotorów. W jednym przypadku dotrenowany model uzyskał nawet minimalnie lepszy wynik od modelu *alt_ last*. Natomiast w przypadku sekwencji pochodzących z komórek iAstrocytes predykcja klas PA, NA, NI jest do siebie bardzo zbliżona. Największy potencjał dostrzegalny jest w poprawie klasyfikacji klasy nieaktywnych niepromotorów.



Rysunek 3.7: Otrzymano stabilną metodę dotrenowywania modelu *alt_ last*. Niewielki spadek dla pierwotnego zbioru danych jest rekompensowany przez wzrost poprawności klasyfikacji sekwencji ze zbioru na którym odbywało się douczanie.

Sprawdzono także, czy różne prawdopodobieństwa reaktywacji filtrów wpływają na jakość predykcji. W tym celu zestawiano wyniki dla poszczególnych klas, grupując wykorzystane prawdopodobieństwo reaktywacji. W celu zbadania istnienia istotnych różnic między rozkładami, dla każdej klasy wykorzystano test Kruskala-Wallisa, aby porównać ze sobą ich mediany. Uzyskane p-wartości nie pozwoliły jednak na odrzucenie żadnej hipotezy zerowej. Oznacza to, że % filtrów, których wagi zostały zmienione, nie wpływa w bezpośredni sposób na predykcję po douczeniu na nowym zbiorze danych.

Klasa	**Zbiór**	**p-wartość (Kruskal-Wallis)**
PA	Alt	0.576
NA	Alt	0.363
PI	Alt	0.861
NI	Alt	0.158
PA	HighSignal	0.665
NA	HighSignal	0.422
PI	HighSignal	0.990
NI	HighSignal	0.618

Tabela 3.4: Wyniki testu Kruskala–Wallisa wszystkich klas poszczególnych zbiorów, dla rozkładów AUC pogrupowanych według prawdopodobieństwa reaktywacji nieistotnych filtrów pierwszej warstwy. Modele zostały nazwane zgodnie z następującym schematem: [**R**]_[litera alfabetu łacińskiego].

3.5. Redukcja normy poprzez estymację średnich wartości

W dotychczasowych eksperymentach zmiany wprowadzone w filtrach pierwszej warstwy wpływały na końcową predykcję modelu w minimalnym stopniu. Aby zweryfikować i zwiększyć istotność statystyczną wyników, zdecydowano się przeprowadzić wielokrotne powtórzenia dotrenowania modelu *alt_ last* z inną techniką inicjalizacji wag wyzerowanych filtrów. Zauważono, że podczas dotrenowania sieć bardzo szybko redukowała normę większości reaktywowanych filtrów, prowadząc do ich niemal natychmiastowego wyzerowania i utraty znaczenia.

W odpowiedzi zamiast próbkować nowe wagi filtrów z rozkładu oryginalnych wag postanowiono skorzystać z rozkładu średnich wartości wag istotnych filtrów w modelu *alt_ last*. Ze względu na występowanie wewnątrz poszczególnych jąder wag zarówno dodatnich, jak i ujemnych, zastosowanie średniej pozwoliło na ponad czterokrotne zmniejszenie normy Frobeniusa inicjowanych filtrów (*circa* z 0.86 do 0.19). W efekcie stworzono cztery grupy eksperymentalne modeli:

- Grupa 1: bazowe 60 modeli, te same co poprzednio, ze standardową inicjalizacją wag wyzerowanych filtrów.
- Grupa 2: zawierajaca po 5 modeli dla każdej wartości $\mathbf{R}$. Wartości w_i w równaniu 1.4 zastąpiono średnimi wartościami $\hat{w}_i = \frac{1}{76} \cdot \sum_{j=1}^{m=76} w_j$ każdego istotnego jądra modelu *alt_ last*.
- Grupa 3: 20 modeli, w których średnie wartości mogły zastąpić wagi zarówno filtrów pierwotnie wyzerowanych, jak i istotnych. Ponadto w tej grupie dwukrotnie zwiększono wartość *momentum* (z 0,1 do 0,2), aby przyspieszyć zbieżność sieci.
- Grupa 4: zastosowano większe momentum, ale zmianie wag poddano wyłącznie filtry uznane za nieaktywne w modelu *alt_ last*.

W każdej z grup porównano różnice w uzyskiwanych wartościach AUC dla poszczególnych klas na obu zbiorach testowych. Test Kruskala–Wallisa wykonany dla każdej klasy osobno nie wykazał istotnych różnic między grupami , analogicznie jak w tabeli 3.4. Potwierdziło to hipotezę, że sama liczba reaktywowanych filtrów w obrębie zastosowanych strategii nie wpływa istotnie na końcową skuteczność predykcji modelu.

Skoro metryki nie różniły się między sobą w zależności od prawdopodobieństwa, z jakim reaktywowano filtry, postanowiono zagregować wyniki AUC wewnątrz poszczególnych grup dla każdej z czterech klas. Ponownie testem Kruskala-Wallisa porównano ze sobą wyniki różnych grup w celu stwierdzenia, czy istnieją różnice w jakości predykcji dla poszczególnych klas w zależności od przyjętej strategii reaktywacji.

Wyniki w tabeli 3.5 wskazują, że modele z grupy trzeciej osiągają inną wartość AUC niż pozostałe. Różnice stwierdzono zarówno dla klasy PA zbioru *Alt* oraz klasy PI zbioru *HighSignal*. Testy parowe Mann–Whitney'a z korektą Benjaminiego-Hochberga wykazały, że w obu przypadkach modele z grupy 3 (ingerujące w pierwotnie aktywne filtry) osiągały istotnie gorsze wyniki niż modele grupy 4.

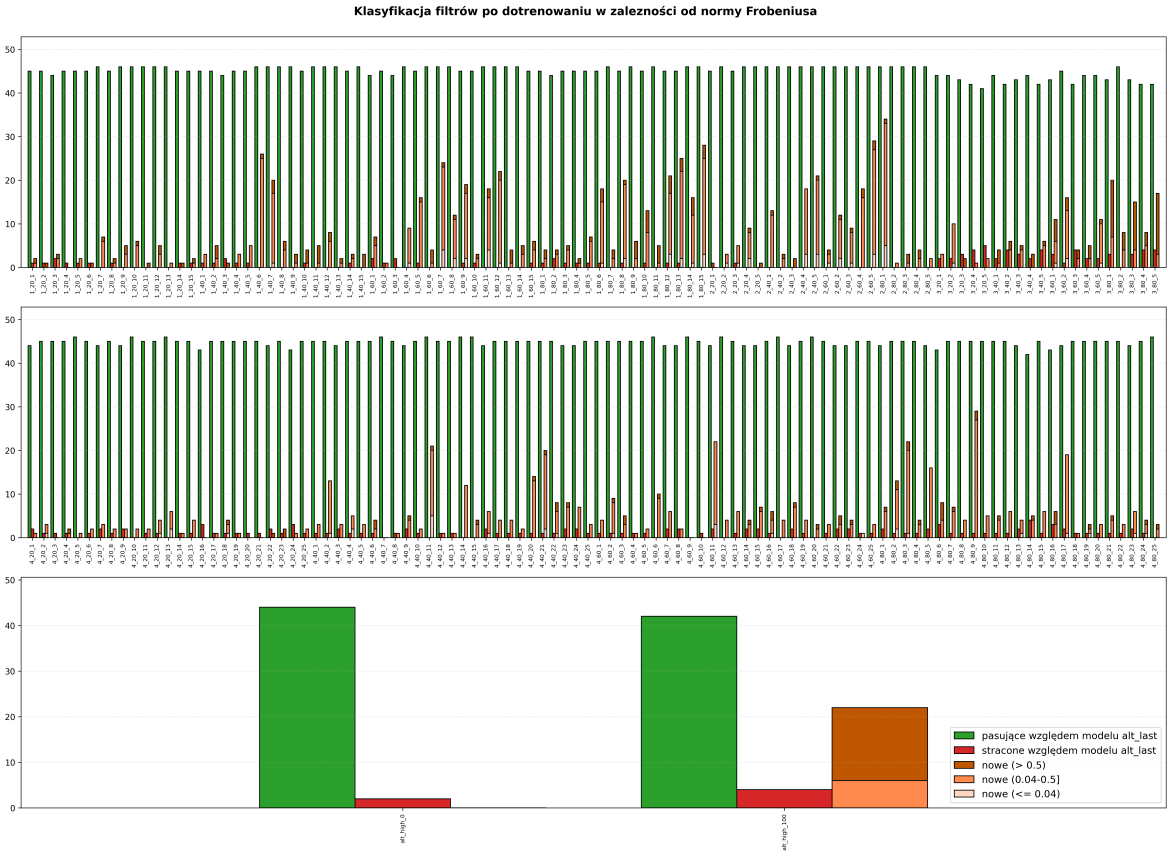
Klasa	Zbiór	p-wartość (Kruskal-Wallis)
PA	Alt	0.016
NA	Alt	0.275
PI	Alt	0.321
NI	Alt	0.395
PA	HighSignal	0.537
NA	HighSignal	0.051
PI	HighSignal	0.017
NI	HighSignal	0.313

Tabela 3.5: Wyniki testu Kruskala–Wallisa czterech grup stategii reaktywacji filtrów między sobą, w obrębie klas poszczególnych zbiorów. istotne róznice wykryto dla klasy nieaktywnych promotorów zbioru *HighSignal* oraz dla klasy PA zbioru *Alt*.

Modele z grupy trzeciej w niektórych przypadkach radzą sobie zatem gorzej od pozostałych. Strategia, w której manipulujemy wagami już wytrenowanych, funkcjonalnych filtrów, może osłabiać predykcję, podczas gdy zmiana wag tylko nieaktywnych filtrów nie ma negatywnego wpływu na jakość klasyfikacji. Natomiast sposób wprowadzenia szumu do pierwszej warstwy konwolucyjnej nie ma bezpośredniego wpływu na jakość predykcji. Zmiana może obejmować zarówno wagi o wiele niższe od tych w jadrach propagujących sygnał dalej, jak i do nich zbliżone.

Następnie skupiono się na analizie liczby i norm Frobeniusa filtrów konwolucyjnych w modelach osiągających najlepsze wyniki walidacyjne. Rysunek 3.8 prezentuje, ile filtrów z pierwotnie aktywnych modeli *alt_last* pozostało aktywnych w dotrenowanych modelach oraz jakie normy Frobeniusa uzyskały filtry reaktywowane.

Dla $\mathbf{R} = 0.20$ rzadko obserwujemy przyrost nowych filtrów. Grupa trzecia charakteryzuje się większą szansą na stracenie istotnego filtru, ze względu na zamianę ich wag na etapie preinicjacyjnym. Nie stwierdzono jednak istnienia bezpośredniej zależności między liczbą istotnych filtrów a szansą reaktywacji. Relatywnie często modele adaptowały około 20 nowych filtrów o niższych normach Frobeniusa. Efekt ten można interpretować jako kompensację: zamiast filtrów, które nie uogólniają się na nowy rodzaj sekwencji, sieć wykorzystuje większą liczbę nowych, słabszych filtrów, które mogą zostać dopasowane w czasie uczenia.



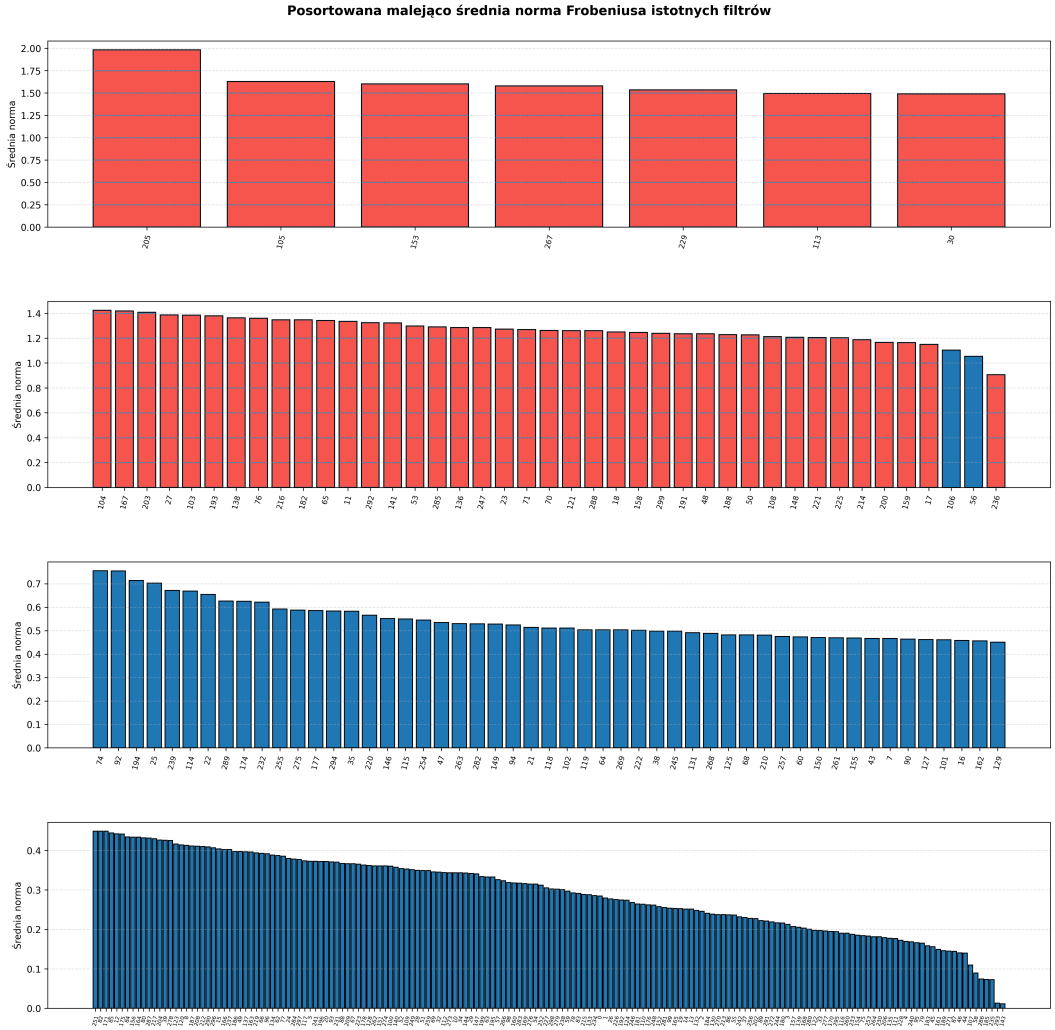
Rysunek 3.8: Dla każdego modelu przedstawiono ilość filtrów współwystępujących w modelu *alt_ last* oraz ile z nich zostało wyzerowanych podczas treningu. Ponadto ukazano normę Frobeniusa nowych filtrów, którym zostały zmienione wagi przed dotrenowaniem. Każdy model został oznaczony według następującego wzorca: [numer grupy]_**R**_[numer porządkowy]. Dodatkowo dla porównania dotrenowano model *alt_ high_ 100*, w którym zmieniono wagi wszystkich nieistotnych filtrów wykorzystując estymację średnimi wagami. Podczas douczania preinicjowane wagi mogą zarówno zwiększać albo zmniejszać swoją normę Frobeniusa.

3.6. Wpływ pierwszej warstwy konwolucyjnej

Oryginalnie istotne filtry pierwszej warstwy konwolucyjnej modelu *alt_ last*, mające uprzednio dostrojone wagi, okazały się silnie konserwowane w kolejnych etapach uczenia. Analiza wykazała, że mimo reaktywacji wielu innych filtrów podczas dotrenowania kluczowe motywy nukleotydowe pozostawały wciąż wykrywane przez wcześniej wyuczone filtry. Tylko filtry o najniższej normie Frobeniusa były czasami zerowane, zwłaszcza w sytuacjach, gdy ingerowano w ich wagi.

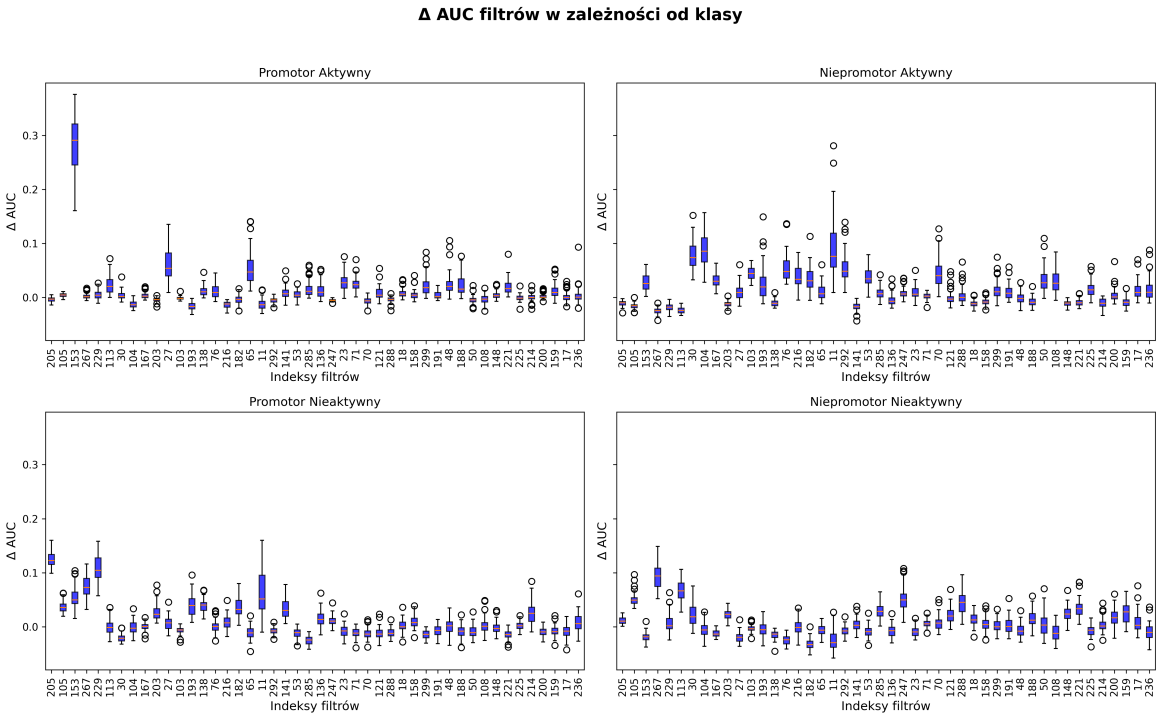
Aby zweryfikować przydatność normy Frobeniusa jako heurystycznej miary istotności filtra, przeprowadzono grupowanie wszystkich filtrów pierwszej warstwy na podstawie ich średnich norm w próbie 200 modeli. Dla każdego filtra obliczono średnią normę, biorąc pod uwagę tylko wartości powyżej ustalonego progu $||W||_F \geq 0.04$ i podzielono według niej filtry na cztery kategorie (rys. 3.9):

1. Filtry o najwyższych wartościach normy występujące we wszystkich modelach
2. Filtry o stosunkowo wysokich wartościach normy
3. Filtry o średnich wartościach normy (grupa plateu)
4. Filtry o najniższych wartościach normy



Rysunek 3.9: Pogrupowanie wszystkich filtrów na cztery kategorie według średnich wartości normy Frobeniusa. Analizowano wyłącznie wartości powyżej przyjętego progu $||W||_F \geq 0.04$. Poniżej niego, wartości normy Frobeniusa były juz o kilka rzędów wielkości mniejsze i dążyły do zera. Kolorem czerwonym zaznaczono filtry oryginalnie istotne występujące w modelu *alt_last*.

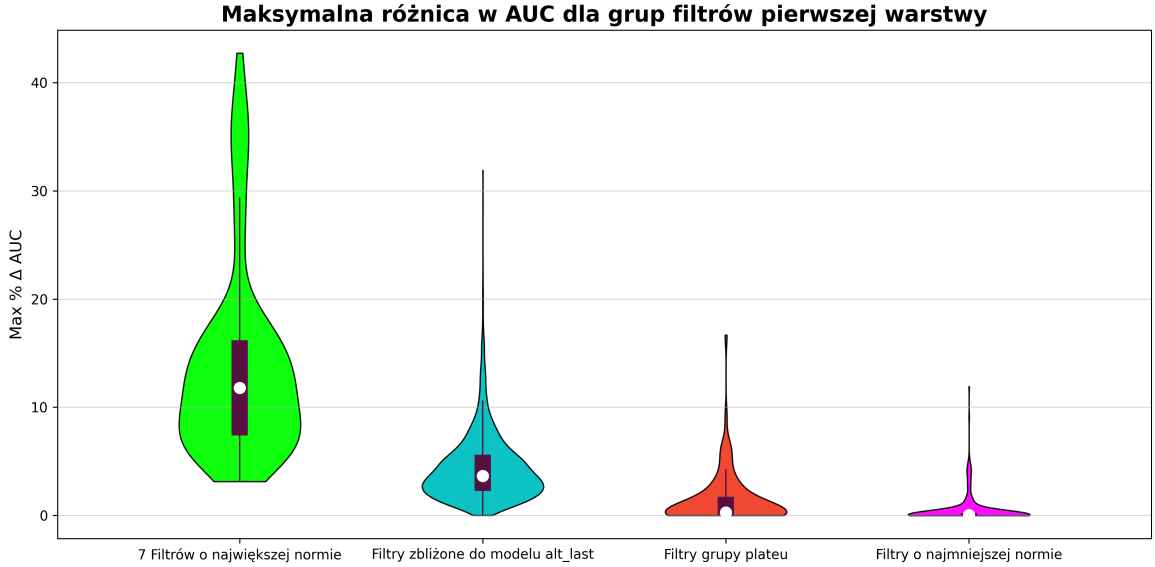
Następnie dla każdego modelu i każdego filtra z normą powyżej progu pojedynczo dezaktywowano każdy z nich, rejestrując zmianę AUC dla wszystkich klas na zbiorze *HighSignal*. Uzyskane różnice przedstawiono na wykresie pudełkowym (rys. 3.10), gdzie osie odpowiadają filtrom uporządkowanym malejąco według średniej wartości normy. Obserwacje wyraźnie wskazywały, że filtry dominujące wartością normy Frobeniusa wywoływały największe spadki jakości predykcji po wyzerowaniu.



Rysunek 3.10: Wpływ wyzerowania oryginalnie istotnych filtrów na metrykę AUC dla sekwencji pochodzących z astrocytów. Dodatnia wartość oznacza, spadek predykcji, a ujemna jej wzrost. Nie istnieje żaden filtr, którego wyzerowanie spowodowałoby pogorszenie predyckji dla każdej klasy. Filtry uporządkowane według malejącej wartości normy Frobeniusa.

W końcowej fazie analizy obliczono dla każdego filtra maksymalną zmianę AUC (największy obserwowany spadek) we wszystkich modelach, a następnie pogrupowano filtry zgodnie z wcześniej przyjętymi kategoriami. Porównanie rozkładów tych maksymalnych zmian pomiędzy grupami (rys. 3.11) wykazało istotne różnice statystyczne. Test Kruskala–Wallisa odrzucił hipotezę zerową, co wskazuje, że rozkłady spadków AUC różnią się między kategoriami. Testy parowe Manna-Whitney'a wykorzystujące korektę Benjamina-Hochberga potwierdziły istotne różnice pomiędzy wszystkimi grupami (tabela 3.6).

Wyniki te sugerują, że norma Frobeniusa jest użyteczną heurystyką do identyfikacji filtrów konwolucyjnych o dużym wpływie na predykcję: filtry o wysokiej wartości normy wywołują zwykle większe spadki AUC po dezaktywacji niż filtry o niskich normach.



Rysunek 3.11: Rozkłady maksymalnych zmian wartości AUC pogrupowane według czterech kategorii filtrów zdefiniowanych na podstawie średniej wartości normy Frobeniusa.

Test	Statystyka	p-wartość	p-wartość BH
Kruskal-Wallis (4 kategorie)	1443.37	$0^{\dagger}$	
Mann-Whitney (porównania parami)			
1 vs 2	885859.50	2.906×10^{-162}	1.743×10^{-161}
1 vs 3	37816.00	5.926×10^{-47}	8.889×10^{-47}
1 vs 4	141456.00	2.571×10^{-122}	5.143×10^{-122}
2 vs 3	181132.50	6.507×10^{-31}	7.808×10^{-31}
2 vs 4	739290.50	1.540×10^{-157}	4.619×10^{-157}
3 vs 4	20772.00	7.688×10^{-7}	7.688×10^{-7}

Tabela 3.6: Wyniki testu Kruskala–Wallisa oraz testów Manna–Whitneya z korekcją Benjamini–Hochberga porównujące cztery kategorie filtrów (zdefiniowane na podstawie normy Frobeniusa). Znaczne wartości statystyk i bardzo małe p-wartości ($p \ll 0.001$) wskazują na istotne różnice pomiędzy grupami filtrów pod względem wpływu na AUC po ich dezaktywacji.

Rozdział 4

Podsumowanie i dyskusja

Niniejsza praca wykazała dwutorowy charakter wyników eksperymentów dotyczących dostrajania konwolucyjnej sieci neuronowej (CNN) zastosowanej do klasyfikacji obszarów cis-regulatorowych w danych genomicznych pochodzących z układu nerwowego.

Po pierwsze, przeprowadzone analizy potwierdziły skuteczność transferu wiedzy pomiędzy pokrewnymi zbiorami danych genomowych. Odpowiedni dobór hiperparametrów, w szczególności współczynnika uczenia, umożliwił skuteczne dostosowanie modelu do danych pochodzących z astrocytów, przy jednoczesnym zachowaniu podstawowej wiedzy o glejakach. Proces dotrenowania zaowocował poprawą miar jakości na zbiorze docelowym, kosztem nieznacznego pogorszenia wydajności na oryginalnym zbiorze źródłowym. Jest to charakterystyczny efekt procesu *fine-tuningu*, w którym model, adaptując się do nowych wzorców, częściowo traci informację o wcześniej wyuczonych cechach. Uwzględniając perspektywę wydajności klas docelowych, kompromis okazał się korzystny.

Po drugie, przeprowadzono szeroki zestaw eksperymentów polegających na modyfikacji wag filtrów pierwszej warstwy konwolucyjnej. Żadna metoda nie pozwoliła jednak na osiągnięcie powtarzalnej i statystycznie istotnej poprawy wyników ponad te uzyskane standardowym dotrenowaniem (model *alt_high_0*). Oznacza to, że przy przyjętej architekturze sieci i dostępnych zbiorach danych, manipulacje wagami pierwszej warstwy konwolucyjnej nie stanowią realnej ścieżki do dalszego zwiększania skuteczności klasyfikacji. Na podstawie wielokrotnie powtarzanych eksperymentów i testów statystycznych można wnioskować, że nie istnieje optymalny, uniwersalny schemat modyfikacji filtrów, który prowadziłby do istotnej poprawy działania modelu.

Warto odnotować, że dotrenowane modele osiągały średnio lepsze wyniki w detekcji większości klas niż model *high4_last* trenowany *ab initio*. Potwierdza to przewagę uczenia transferowego nad uczeniem od podstaw, ponieważ pozwala ono na wykorzystanie i wzmocnienie cennych cech wyekstrahowanych podczas wstępnego treningu na większym zbiorze danych.

Z biologicznego punktu widzenia, szczególnie interesująca jest obserwacja dotycząca związku pomiędzy nieaktywnymi enhancerami ze zbioru *HighSignal* a aktywnymi promotorami ze zbioru *Alt*. Być może przestrzenna reorganizacja chromatyny w astrocytach zachowuje pewne strukturalne motywy (pętle, domeny TAD), które w glejakach ulegają rearanżacjom, pozwalając na zaistnienie nowych interakcji. Dzięki temu proces dotrenowania mógł pozwolić modelowi na lepsze uchwycenie pewnych zależności epigenetycznych, co tłumaczyłoby zaobserwowaną poprawę w predykcji nieaktywnych enhancerów (NI) w astrocytach oraz aktywnych promotorów (PA) w glejakach. Należy jednak pamiętać, że klasa nieaktywnych enhancerów ma częściowo sztuczny charakter, ponieważ grupuje ona wszystkie sekwencje, które nie zostały zaklasyfikowane do innych kategorii.

W niezależnych powtórzeniach eksperymentu główny sygnał klasyfikacyjny pochodził od niewielkiego, silnie konserwowanego zestawu filtrów pochodzących z oryginalnego modelu *alt_last*. Nawet pełna reaktywacja wszystkich uprzednio wyzerowanych filtrów prowadziła jedynie do nieznacznego osłabienia tych uznanych za istotne. Sugeruje to, że filtry te odpowiadają za rozpoznanie fundamentalnych motywów, preferowanych przez sieć w procesie klasyfikacji. Z drugiej strony, filtry 'odzerowywane' w procesie dotrenowania były zazwyczaj ponownie zerowane. Tylko pojedyncze z nich uzyskiwały normę Frobeniusa o wartości zbliżonej do oryginalnie istotnych filtrów. Zdecydowana większość miała ją stosunkowo mniejszą, przy jednoczesnym zachowaniu niezerowej wartości wszystkich wag. Pozostawieniu wielu nowych filtrów często towarzyszy wyzerowanie pojedynczych tych, które pierwotnie uznane zostały za istotne. Wyzerowane filtry w porównianiu do innych, charakteryzowały się stosunkowo niską wartością normy przed dotrenowaniem. Można interpretować to zjawisko jako mechanizm kompensacji utraconej wiedzy na rzecz dostosowania się do nowego zbioru danych.

W reakcji na nowe dane najmocniej adaptują się jednak filtry, które miały największą normę Frobeniusa. Poprzez zwiększenie, co do modułu, wartości swoich wag, dostosowują się do wykrywania specyficznych wzorców astrocytarnych. Dla poszczególnych klas zbioru docelowego podobne wyniki osiągały modele, które zostawiły sobie zaledwie 20 filtrów, jak i te posiadające ich w pierwszej warstwie powyżej 80. Brak drastycznego zmniejszenia ilości pozostawionych istotnych filtrów pozwala jednak lepiej zniwelować efekt zapominania.

Oczekiwanym skutkiem modyfikacji wag istotnych filtrów było pogorszenie zdolności predykcyjnych modelu, co zostało potwierdzone eksperymentalnie. Wyniki te wskazują, że kluczowe motywy są zakodowane w konkretnych, oryginalnych filtrach, a próby zmiany ich wag lub pozycji skutkują obniżeniem jakości klasyfikacji. Co ważne, wykorzystanie metody KDE do sterowania reinicjacją wag nie pozwoliło na zidentyfikowanie optymalnego podzbioru filtrów, którego reaktywacja umożliwiłaby odkrycie nowych, użytecznych motywów. Przyjęta architektura sieci skutecznie wyodrębniła główny zbiór lokalnych motywów sekwencyjnych istotnych dla zadania klasyfikacji na zbiorze *Alt*. Jednak dalsze manipulacje pierwszą warstwą konwolucyjną nie poszerzały tego zbioru w sposób poprawiający działanie modelu. Dodatkowo, sieć wykazuje zjawisko redundancji i kompensacji - nawet po wprowadzeniu perturbacji w pierwszej warstwie, głębsze warstwy są w stanie częściowo zrekonstruować niezbędne reprezentacje na podstawie pozostałego sygnału.

Analiza pierwszej warstwy ujawniła, że filtry o największej normie Frobeniusa zawsze pozostawały aktywne i były odpowiedzialne za najsilniejsze zmiany, kiedy je dezaktywowano. Analogicznie, filtry o najmniejszej wartości normy wpływały zazwyczaj w znikomym stopniu na prawidłowość dokonywanej klasyfikacji. Norma Frobeniusa nie pozwala ściśle określić przydatności filtrów, ale stanowi jej prostą i użyteczną miarę, pozwalając na ich pogrupowanie. Warto podkreślić, że każdy filtr, nawet taki o bardzo niskiej normie, bierze czynny udział w dokonywanej klasyfikacji. Jego przydatność może być minimalna (około 0.1%), ale usunięcie zawsze powoduje spadek AUC w jednej klasie i wzrost w innej. Każde jądro rozpoznaje zatem pewną klasę, zapobiegając błędnej klasyfikacji innej.

Umieszczając otrzymane wyniki w szerszym kontekście literatury, można wskazać kilka konkretnych wniosków popartych ostatnimi pracami. Minnoye i współpracownicy [60] wykazali, że głębokie sieci uczą się jednocześnie konserwowanych motywów oraz gatunkowo specyficznych cech enhancerów, co tłumaczy obserwowaną w naszych eksperymentach stabilizację pewnej ilości filtrów pierwszej warstwy. Nie okazują się one nieistotne po procesie dotrenowania, ponieważ są uniwersalnymi detektorami istotnych motywów.

Inne prace oraz badania nad syntetycznymi enhancerami w kontekście komórkowym [61, 62, 63] pokazują natomiast praktyczne zastosowania wyuczonych reprezentacji. Modele oparte na sekwencji potrafią przewidywać aktywność enhancerów i służyć do projektowania funkcjonalnych sekwencji, a następnie weryfikacji ich aktywności *in vitro* i *in vivo*. Można zatem zastrzec, że wniosek o braku istotnej poprawy poprzez manipulacje warstwą konwolucyjną dotyczy wyłącznie naszej kombinacji: konkretnej architektury *Custom* oraz charakterystyki dostępnych zbiorów danych i przyjętej definicji klas. Inne architektury, a także integracja multimodalnych sygnałów epigenomicznych, mogłyby dostarczyć dodatkowy sygnał i pozwolić na przekroczenie obecnych ograniczeń.

Zastosowanie normy Frobeniusa, jako heurystycznej miary użyteczności filtrów, pojawia się także w literaturze dotyczącej pruningu konwolucyjnych sieci neuronowych. Sam proces polega na upraszczaniu modelu poprzez usuwanie zbędnych lub mniej istotnych elementów. Pozwala on na poprawę wydajności poprzez ograniczenie nadmiernego dopasowania, przy zmniejszeniu liczby parametrów, co pozwala efektywne obniżenie kosztów obliczeniowych. Przykładowo algorytm *OCNNA* [64] wykorzystuje miary oparte na normie Frobeniusa do wyznaczenia ważności filtrów i ich selekcji podczas kompresji modelu. Norma ta ma jednak swoje ograniczenia. Jest miarą skalarną, nie uwzględniającą struktury przestrzennej wektorów wag ani korelacji między ich poszczególnymi wartościami. Powinna być zatem stosowana z innymi kryteriami, oceniającymi istotność poszczególnych filtrów. W przyszłości można rozważyć metody uwzględniające wzajemny kierunek i strukturę filtrów, takie jak analiza podobieństwa oparta na normie cosinusowej [6] lub wielowymiarowe techniki estymacji gęstości [58].

Podsumowując, pierwsza warstwa konwolucyjna badanej sieci wykazała się silną plastycznością funkcjonalną. Pomimo aktywnego udziału każdego filtra, sieć potrafi adaptować swoją wewnętrzną reprezentację bez drastycznej utraty ogólnej dokładności. Jednak z punktu widzenia zastosowanej architektury, najważniejsze są cechy wyższego rzędu - sposób, w jaki głębsze warstwy sieci łączą podstawowe motywy wykrywane przez pierwszą warstwę konwolucyjną w coraz bardziej abstrakcyjne reprezentacje. To właśnie one pozwalają modelowi zachować wysoką skuteczność nawet przy wprowadzeniu znaczących perturbacji na najniższym poziomie. W efekcie, głębsze warstwy demonstrują zdolność do kompensacji lokalnych zaburzeń, co świadczy o pewnym stopniu redundancji w ekstrahowanych cechach oraz podkreśla potencjał głębokiego uczenia w problemach biologicznych.

Bibliografia

[1] Marlena Osipowicz, *Identification of chromatin regions active in human brain using neural networks*, Master's thesis. *Uniwersytet Warszawski Wydział Matematyki, Informatyki i Mechaniki* (2021).

[2] Julia Smolik, *Interpretacja filtrów z warstw konwolucyjnych sieci neuronowych przy pomocy motywów DNA*, Master's thesis. *Uniwersytet Warszawski Wydział Matematyki, Informatyki i Mechaniki* (2023).

[3] Piotr Lembryk, *Analiza wpływu architektury konwolucyjnej sieci neuronowej na jakość klasyfikacji aktywności obszarów regulatorowych DNA*, Master's thesis. *Uniwersytet Warszawski Wydział Matematyki, Informatyki i Mechaniki* (2021).

[4] David R. Kelley, Jasper Snoek, John L. Rinn, *Basset: learning the regulatory code of the accessible genome with deep convolutional neural networks. Genome Research* 26(7):990–999 (2016).

[5] Jason Yosinski, Jeff Clune, Yoshua Bengio and Hod Lipson, *How transferable are features in deep neural networks?* In Advances in Neural Information Processing Systems 27, *NIPS Foundation* (2014).

[6] Arshdeep Singh, Mark D. Plumbley, *A Passive Similarity based CNN Filter Pruning for Efficient Acoustic Scene Classification. arXiv:2203.15751* (2022).

[7] Rhea Chowers, Yair Weiss, *What do CNNs Learn in the First Layer and Why? A Linear Systems Perspective. arXiv:2206.02454* (2022).

[8] Reza Abbasi-Asl, Bin Yu, *Structural Compression of Convolutional Neural Networks with Applications in Interpretability. Frontiers in Big Data*, Volume 4 (2021).

[9] Karolina Stępniak, Magdalena A. Machnicka, Jakub Mieczkowski, Anna Macioszek, Bartosz Wojtaś, Bartłomiej Gielniewski, Katarzyna Poleszak, Malgorzata Perycz, Sylwia K. Król, Rafał Guzik, Michał J. Dąbrowski, Michał Dramiński, Marta Jardanowska, Ilona Grabowicz, Agata Dziedzic, Hanna Kranas, Karolina Sienkiewicz, Klev Diamanti, Katarzyna Kotulska, Wiesława Grajkowska, Marcin Roszkowski, Tomasz Czernicki, Andrzej Marchel, Jan Komorowski, Bożena Kamińska & Bartek Wilczyński, *Mapping chromatin accessibility and active regulatory elements reveals pathological mechanisms in human gliomas*, *Nature Communications*, Volume 12, Article number: 1978 (2021).

[10] Katarzyna Ciuba, Aleksandra Piotrowska, Debadeep Chaudhury, Bondita Dehingia, Eryk Duński, Rüdiger Behr, Karolina Soroczyńska, Małgorzata Czystowska-Kuźmicz, Misbah Abbas, Edyta Bulanda, Sylwia Gawlik-Zawiślak, Sylwia Pietrzak, Izabela Figiel, Jakub Włodarczyk, Alexei Verkhratsky, Marcin Niedbała, Wojciech Kaspera, Tomasz Wypych, Bartosz Wilczyński, Aleksandra Pękowska, *Molecular signature of primate astrocytes reveals pathways and regulatory changes contributing to human brain evolution*, *Cell Stem Cell*, Volume 32, Issue 3, 6 March, Pages 426-444.e14 (2025).

[11] Stuart Russell, Peter Norvig, *Artificial Intelligence: A Modern Approach. Prentice Hall Press*, ISBN: 978-0-13-604259-4 (2009).

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, *Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. Proceedings of the IEEE International Conference on Computer Vision* (2015).

[13] Anders Krogh, John A. Hertz, *A Simple Weight Decay Can Improve Generalization. neurIPS* (1991).

[14] Murray Rosenblatt, *Remarks on Some Nonparametric Estimates of a Density Function. The Annals of Mathematical Statistics*, 27(3):832 – 837 (1956).

[15] Emanuel Parzen, *On estimation of a probability density function and mode. Annals of Mathematical Statistics, 33:1065–1076* (1962).

[16] Bernard W. Silverman, *Density Estimation for Statistics and Data Analysis. CRC Press*, Volume 26 (1986).

[17] Malgorzata I. Srebniak & Agnieszka Tomaszewska, *Badania cytogenetyczne w praktyce klinicznej, I. Warszawa: Wydawnictwo Lekarskie PZWL*, ISBN: 978-83-200-3489-9 (2008).

[18] Shiv I. S. Grewal, Jia Songtao, *Heterochromatin revisited. Nature Reviews Genetics*, Volume 8, no. 1, pp. 35–46 (2007).

[19] Xueyan Wang, Danli Liu, Jing Luo, Dashuai Kong, Yubo Zhang, *Exploring the Role of Enhancer-Mediated Transcriptional Regulation in Precision Biology. International Journal of Molecular Sciences* 24(13), 10843 (2023).

[20] Nathaniel D. Heintzman et al. *Distinct and predictive chromatin signatures of transcriptional promoters and enhancers in the human genome. Nature Genetics* 39, 311–318 (2007).

[21] Andrea Smallwood & Bing Ren, *Genome organization and long-range regulation of gene expression by enhancers. Current Opinion in Cell Biology* 25, 387–394 (2013).

[22] Mario F. Fraga, Manel Esteller, *Towards the human cancer epigenome: a first draft of histone modifications. Cell Cycle* Oct;4(10):1377-8 (2005).

[23] Rebecca Burgess, Robert Jenkins & Zhiguo Zhang, *Epigenetic changes in gliomas. Cancer Biology & Therapy* 7, 1326–1334 (2008).

[24] Karen Uhlmann, Klaus Rohde, Constanze Zeller, Janusz Szymas, Siegfried Vogel, Karola Marczinek, Gundula Thiel, Peter Nürnberg, Peter W. Laird, *Distinct methylation profiles of glioma subtypes. International Journal of Cancer* 106, 52–59 (2003).

[25] Kasthuri Kannan, Akiko Inagaki, Joachim Silber, Daniel Gorovets, Jianan Zhang, Edward R. Kastenhuber, Adriana Heguy, John H. Petrini, Timothy A. Chan, Jason T. Huse, *Whole exome sequencing identifies ATRX mutation as a key molecular determinant in lower-grade glioma. Oncotarget* Oct 11;3(10):1194–1203 (2012).

[26] Otto G. Berg, Peter H. von Hippel, *Selection of DNA binding sites by regulatory proteins: Statistical-mechanical theory and application to operators and promoters. Journal home page for Journal of Molecular Biology*, Volume 193, Issue 4, 20 February, Pages 723-743 (1987).

[27] Emmanouil T. Dermitzakis , Andrew G. Clark, *Evolution of Transcription Factor Binding Sites in Mammalian Gene Regulatory Regions: Conservation and Turnover Free. Molecular Biology and Evolution*, Volume 19, Issue 7, July, Pages 1114–1121 (2002).

[28] Kushal Suryamohan, Marc S. Halfon, *Identifying transcriptional cis-regulatory modules in animal genomes. Wiley Interdisciplinary Reviews-Developmental Biology* Mar-Apr;4(2):59-84 (2015).

[29] Peter J. Park, *ChIP–seq: advantages and challenges of a maturing technology. Nature Reviews Genetics*, Volume 10, pages 669–680 (2009).

[30] Erez Lieberman-Aiden , Nynke L. van Berkum, Louise Williams, Maxim Imakaev, Tobias Ragoczy, Agnes Telling, Ido Amit, Bryan R. Lajoie, Peter J. Sabo, Michael O. Dorschner, Richard Sandstrom, Bradley Bernstein, Irene M. A. Bender, Mark Groudine, Andreas Gnirke, John Stamatoyannopoulos, Leonid A. Mirny, Eric S. Lander, Job Dekker, *Comprehensive mapping of long range interactions reveals folding principles of the human genome. Science* Oct 9; 326(5950):289-93 (2009).

[31] Xuehai Hu, Alisdair R. Fernie, Jianbing Yan, *Deep learning in regulatory genomics: from identification to design. Current Opinion in Biotechnology*, Volume 79, February, 102887 (2023).

[32] Anastasia Levchenko, Alexander Kanapin, Anastasia Samsonova, Raul R. Gainetdinov, *Human Accelerated Regions and Other Human-Specific Sequence Variations in the Context of Evolution and Their Relevance for Brain Development. Genome Biology and Evolution* Jan 1;10(1):166-188 (2018).

[33] Md Riyad Hossain, Douglas Timmer, Hiram Moya, *Machine learning model optimization with hyper-parameter tuning approach. International Conference on Advanced Engineering, Technology and Applications (ICAETA) At: Istanbul, Turkey* (2021).

[34] Jasper Snoek, Hugo Larochelle and Ryan P. Adams, *Practical Bayesian Optimization of Machine Learning Algorithms. arXiv:1206.2944v2 [stat.ML]* 29 Aug (2012).

[35] Davide Chicco, Giuseppe Jurman, *The Matthews correlation coefficient (MCC) should replace the ROC AUC as the standard metric for assessing binary classification. BioData Mining* Feb 17;16:4 (2023).

[36] Davide Chicco & Giuseppe Jurman, *The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. BMC Genomics*, Volume 21, Article number: 6 (2020).

[37] Warren S. McCulloch & Walter Pitts, *A logical calculus of the ideas immanent in nervous activity. Bulletin of Mathematical Biophysics*, Volume 5, pp. 115-133 (1943).

[38] David E. Rumelhart, Geoffrey E. Hinton & Ronald J. Williams, *Learning representations by back-propagating errors. Nature*, Volume 323, pages 533–536 (1986).

[39] Juyoung Yun, *Mitigating Gradient Overlap in Deep Residual Networks with Gradient Normalization for Improved Non-Convex Optimization. arXiv:2410.21564v3 [cs.LG]* 15 Nov (2024).

[40] Yann LeCun, Leon Bottou, Yoshua Bengio & Patrick Haffner, *Gradient Based Learning Applied to Document Recognition. Proceedings of the IEEE*, Volume 86, no. 11, pp. 2278-2324, November (1998).

[41] Akshata Hegde, Tom Nguyen, Jianlin Cheng, *Machine Learning Methods for Gene Regulatory Network Inference. arXiv:2504.12610v1* (2025).

[42] Sai Teja Mummadi, Md Khairul Islam, Victor Busov & Hairong Wei *Gene regulatory network prediction using machine learning, deep learning, and hybrid approaches. Forestry Research* 5, Article number: e014 (2025).

[43] Babak Alipanahi, Andrew Delong, Matthew T Weirauch & Brendan J. Frey, *Predicting the sequence specificities of DNA- and RNA-binding proteins by deep learning. Nature Biotechnology*, Volume 33, pages 831–838 (2015).

[44] Jian Zhou & Olga G. Troyanskaya, *Predicting effects of noncoding variants with deep learning–based sequence model. Nature Methods*, Volume 12, pages 931–934 (2015).

[45] Louisa-Marie Krützfeldt, Max Schubach, Martin Kircher, *The impact of different negative training data on regulatory sequence predictions. PLOS One* Dec 1;15(12):e0237412 (2020).

[46] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, Ruslan Salakhutdinov, *Dropout: A Simple Way to Prevent Neural Networks from Overfitting. Journal of Machine Learning Research* 15 (2014).

[47] William H. Kruskal & W. Allen Wallis, *Use of ranks in one-criterion variance analysis. Journal of the American Statistical Association*, 47(260), 583–621 (1952).

[48] Henry B. Mann & Donald R. Whitney, *On a Test of Whether one of Two Random Variables is Stochastically Larger than the Other. The Annals of Mathematical Statistics* 18(1): 50-60 March (1947).

[49] Yoav Benjamini, Yosef Hochberg, *Controlling the False Discovery Rate: a Practical and Powerful approach to Multiple Testing. Journal of the Royal Statistical Society: Series B (Methodological)*, 57(1), 289–300 (1995).

[50] Charles Spearman, *The Proof and Measurement of Association between Two Things. The American Journal of Psychology*, Volume 15, No. 1 January (1904).

[51] Karl Pearson, *Contributions to the Mathematical Theory of Evolution, II: Skew Variation in Homogeneous Material. Philosophical Transactions of the Royal Society*, 186, 343-414 (1895).

[52] Ilya Sutskever, James Martens, George Dahl, Geoffrey Hinton, *On the importance of initialization and momentum in deep learning. Proceedings of the 30th International Conference on Machine Learning, PMLR* 28(3):1139-1147 (2013).

[53] Tijmen Tieleman & Geoffrey Hinton, *Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. COURSERA: Neural Networks for Machine Learning* (2012).

[54] Xavier Glorot, Yoshua Bengali, *Understanding the difficulty of training deep feedforward neural networks. Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, PMLR* 9:249-256 (2010).

[55] ENCODE Project Consortium, *An integrated encyclopedia of DNA elements in the human genome. Nature* 489, 57–74 (2012).

[56] Faris Rustom, Ezekiel Moroze, Pedram Parva, Haluk Ogmen, Arash Yazdanbakhsh, *Deep learning and transfer learning for brain tumor detection and classification. Biology Methods & Protocols* Nov 19;9(1):bpae080 (2024).

[57] Lulu Gong, Fabio Pasqualetti, Thomas Papouin, ShiNung Ching, *Astrocytes as a mechanism for contextually-guided network dynamics and function. PLOS Computational Biology* May 31;20(5) (2024).

[58] Travis A. O'Brien, Karthik Kashinath, Nicholas R. Cavanaugh, William D. Collins, John P. O'Brien, *A fast and objective multidimensional kernel density estimation method: fastKDE. Computational Statistics & Data Analysis* Volume 101, September, Pages 148-160 (2016).

[59] Gene H. Golub & Charles F. Van Loan, *Matrix Computations. John Hopkins University Press*, ISBN: 978-0801854149 (1996).

[60] Liesbeth Minnoye, Ibrahim I. Taskiran, David Mauduit, Maurizio Fazio, Linde V. Aerschot, Gert Hulselmans, Valerie Christiaens, Samira Makhzami, Monika Seltenhammer, Panagiotis Karras, Aline Primot, Edouard Cadieu, Ellen van Rooijen, Jean-Christophe Marine, Giorgia Egidy, Ghanem-Elias Ghanem, Leonard Zon, Jasper Wouters, Stein Aerts, *Cross-species analysis of enhancer logic using deep learning. Genome Research* Dec, 30(12):1815-1834 (2020).

[61] Bernardo P. de Almeida, Franziska Reiter, Michaela Pagani & Alexander Stark. *DeepSTARR predicts enhancer activity from DNA sequence and enables the de novo design of synthetic enhancers. Nature Genetics* Volume 54, pages 613–624 (2022).

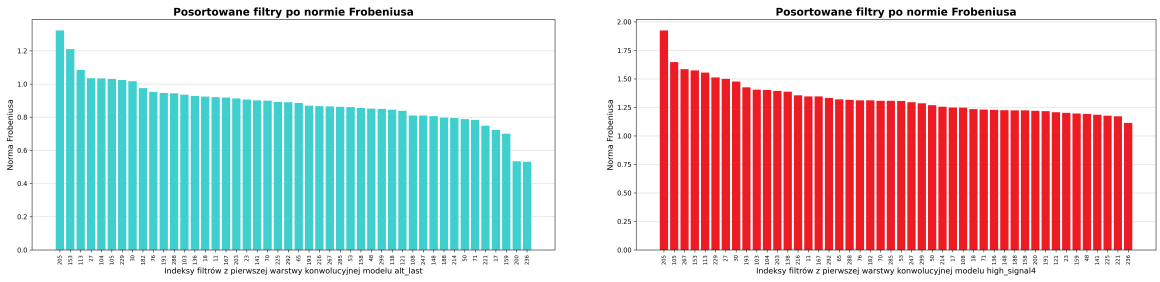
[62] Bernardo P. de Almeida, Christoph Schaub, Michaela Pagani, Stefano Secchia, Eileen E. M. Furlong & Alexander Stark, *Targeted design of synthetic enhancers for selected tissues in the Drosophila embryo. Nature* Volume 626, pages 207-211 (2024).

[63] Ibrahim I. Taskiran, Katina I. Spanier, Hannah Dickmänken, Niklas Kempynck, Alexandra Pančíková, Eren Can Ekşi, Gert Hulselmans, Joy N. Ismail, Koen Theunis, Roel Vandepoel, Valerie Christiaens, David Mauduit & Stein Aerts, *Cell-type-directed design of synthetic enhancers. Nature* Volume 626, pages 212-220 (2024).

[64] Jose M. Benítez, *Optimizing Convolutional Neural Network Architecture. arXiv:2401.01361v1* Dec (2023).

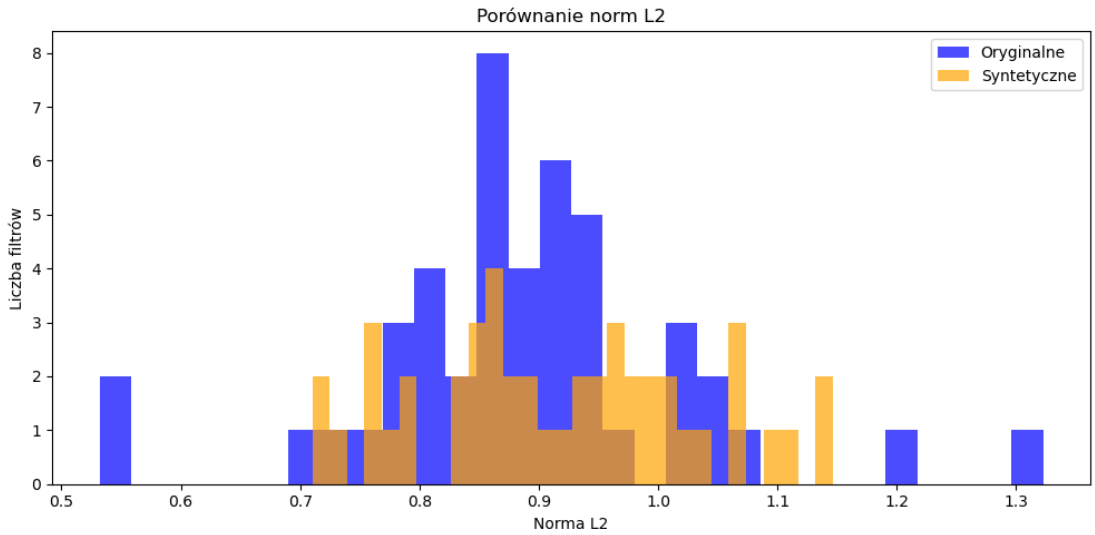
Aneksy

A. Wykresy

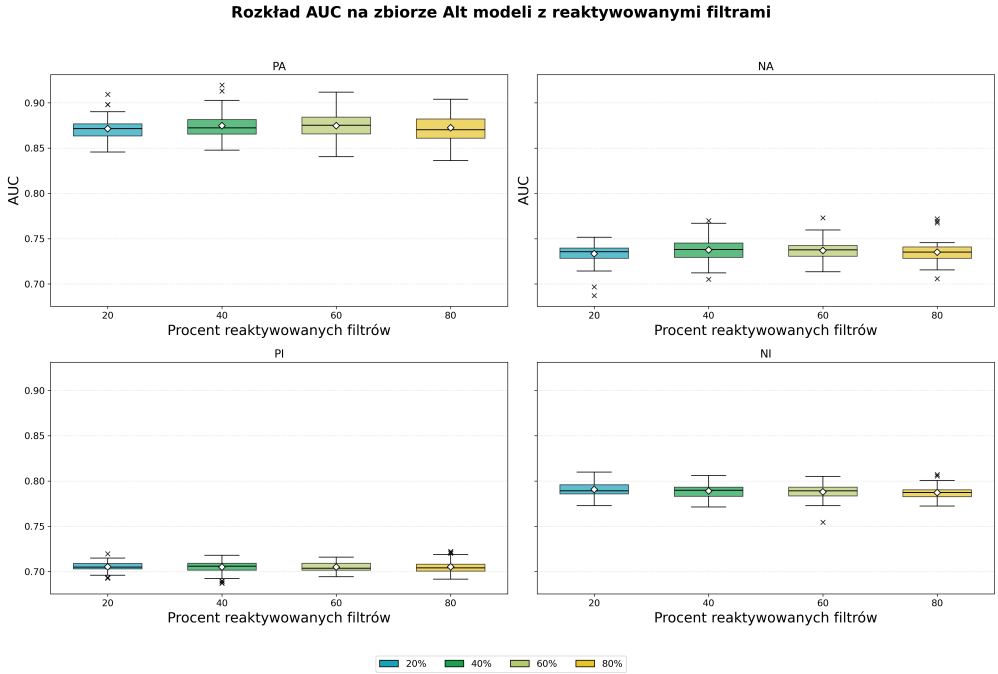


Filtry modelu *alt_ last* Filtry modelu *high4_ last*

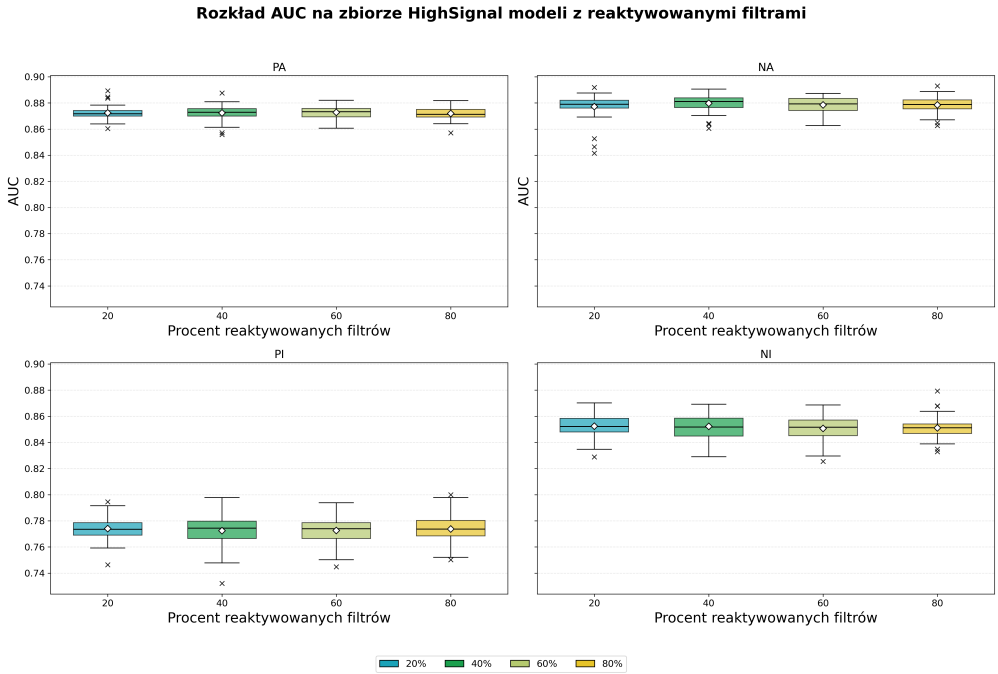
Rysunek 1: Na rysunku przedstawiono histogramy indeksów filtrów pierwszej warstwy konwolucyjnej wraz z odpowiadającymi im wartościami normy Frobeniusa, posortowanymi malejąco. Indeksy istotnych filtrów są ze sobą całkowicie zgodne, jednak istnieją różnice w pozycjach po posortowaniu. Filtry charakterezujące się wysoką normą w modelu *alt_ last*, mają ją jednocześnie w modelu *high4_ last* i *vice versa*. Wszystkie filtry w modelu trenowanym *ab initio* na obu zbiorach danych mają wyższą normę Frobeniusa.



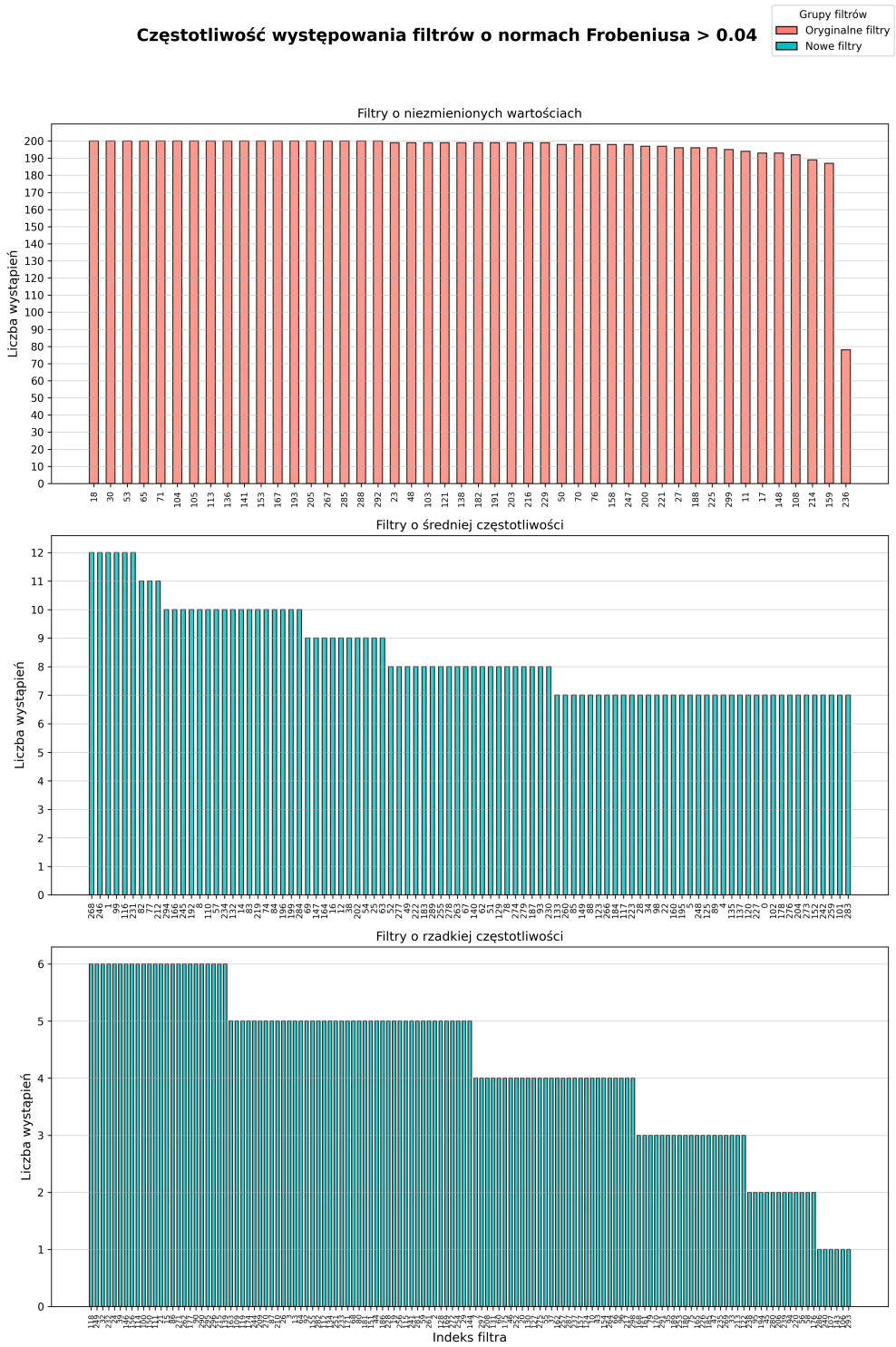
Rysunek 2: Porównanie norm 46 istotnych filtrów modelu *high4_ last* oraz tylu samo syntetycznie wygenerowanych, przy pomocy równania 1.4. Traktując wagi jako rozkład jednowymiarowy metoda pozwala na generowanie wektorów o zbliżonej normie euklidesowej (L2), której uogólnienie stanowi norma Frobeniusa [59]. Wykorzystana metoda ma jednak problem z osiągnięciem skrajnych wartości [16].



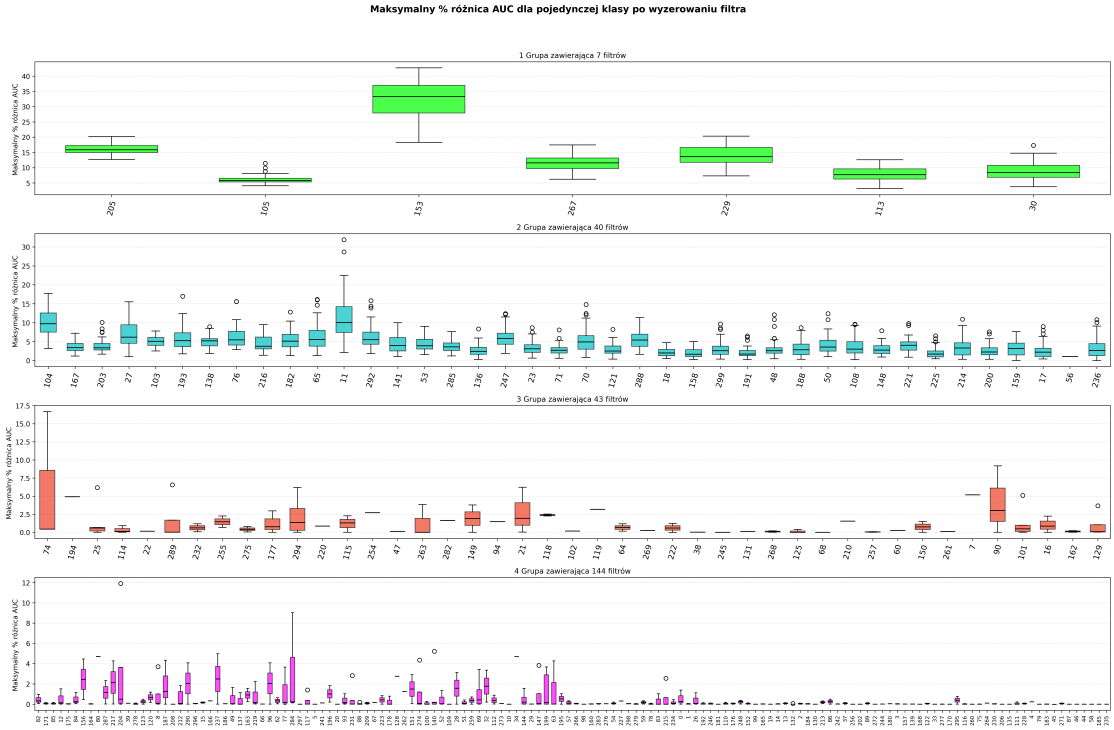
Rysunek 3: Wykres pudełkowy metryki AUC dla zbioru *Alt* pogrupowany w zależności od klasy oraz prawdopodobieństwa reaktywacji filtrów konwolucyjnych z pierwszej warstwy. Do wykrycia istotnych różnic między różnymi wartościami prawdopodobieństwa wykorzystano test Kruskala-Wallisa oraz Manna-Whitney'a (sekcja 1.5). Nie uzyskano jednak żadnej p-wartości poniżej progu 0.05.



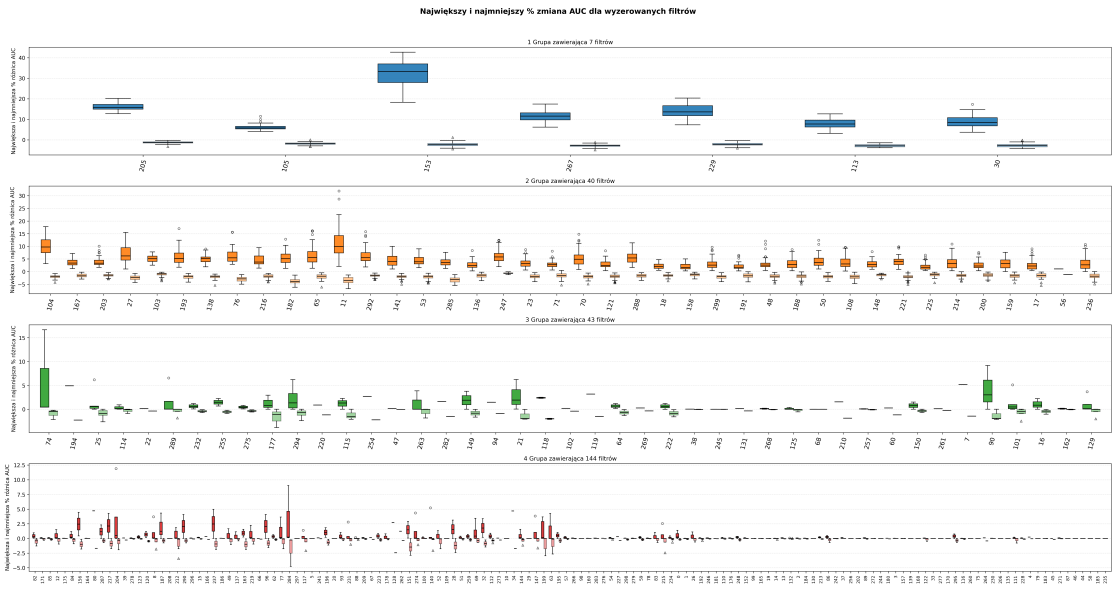
Rysunek 4: Analogiczny wykres pudełkowy dla zbioru *HighSignal* (rys. 3). Wszystkie wykorzystane testy również nie pozwoliły na odrzucenie hipotezy zerowej.



Rysunek 5: Przedstawiono ile razy każdy z filtrów po procesie douczenia, będąc częścią najlepszego modelu na zbiorze walidacyjnym, osiagnął normę Frobeniusa powyżej 0.04. Przyjęto ten próg, dlatego że poniżej niego, wszystkie normy są najwyżej rzędu 10^{-3}. Pociąga to za sobą ich znikomy wpływ na dokonywaną predykcję, będąc artefaktem treningu i sposobu wybrania najlepszego modelu.



Rysunek 6: Na wykresie przedstawiono maksymalne, procentowe różnice w metryce AUC po wyzerowaniu danego filtra dla pojedynczej klasy, modeli pierwszej grupy (podsekcja 3.6). Wykres pudełkowy został pogrupowany na podstawie średnich norm Frobeniusa (rys. 3.9).



Rysunek 7: Widać zarówno największe, jak i najmniesze procentowe zmiany w metryce AUC po wyzerowaniu filtra o odpowiednim indeksie. Wykres ujawnia, że każdy filtr, nieważne jak pomocny w klasyfikacji pewnej klasy, jednocześnie wpływa negatywnie na inną. Znikomy wpływ może być skutkiem redundancji wykrywanych cech przez jądra głębszych warstw konwolucyjnych, spowodowanej zastosowaniem mechanizmu dropout w trakcie treningu.

B. Implementacja

Wszystkie skrypty wykorzystane w projekcie zostały napisane w języku Python, w wersji 3.13.2. Do implementacji oraz trenowania konwolucyjnych sieci neuronowych zastosowano framework `PyTorch` (wersja 2.7.0+cu126), który umożliwia dynamiczne trenowanie modeli oraz efektywne obliczenia na procesorze CPU i większą precyzyjność na karcie graficznej GPU, niezbędnej w przypadku wielokrotnych, kosztownych obliczeń.

Wizualizacje wyników eksperymentów, w tym wykresy zestawiające różnorakie miary jakości, rozkłady danych czy porównania statystyczne, zostały wygenerowane przy pomocy biblioteki `Matplotlib` (wersja 3.10.1), która pozwala na tworzenie dwuwymiarowych wykresów.

Do przetwarzania i manipulowania danymi numerycznymi użyto biblioteki `NumPy` (wersja 2.2.5), która stanowi standardowe narzędzie w analizie danych i uczeniu maszynowym. W celu przeprowadzania testów oraz analiz statystycznych wykorzystano bibliotekę `SciPy` (wersja 1.15.2), oferującą różnorodny zestaw narzędzi do obliczeń naukowych, w tym testy istotności, analizy wariancji czy korelacje.

Wszystkie informacje wykorzystane podczas dotrenowywania modeli wraz z całym użytym kodem są publicznie dostępne w moim repozytorium na platformie GitHub, pod adresem: `https://github.com/kugi8412/RegSeqNet`