Uniwersytet Warszawski
Wydział Fizyki

Mariusz Budziński
Nr albumu: 348587

Analiza sygnałów EEG z zastosowaniem głębokich sieci neuronowych

Praca magisterska
na kierunku Fizyka,
specjalność: metody fizyki w ekonomii (ekonofizyka)

Praca wykonana pod kierunkiem
dr hab. Jarosława Żygierewicza
Zakład Fizyki Biomedycznej

Warszawa, czerwiec 2018r.

Oświadczenie kierującego pracą

Oświadczam, że niniejsza praca została przygotowana pod moim kierunkiem i stwierdzam, że spełnia ona warunki do przedstawienia jej w postępowaniu o nadanie tytułu zawodowego.

Data Podpis kierującego pracą

Oświadczenie autora (autorów) pracy

Świadom odpowiedzialności prawnej oświadczam, że niniejsza praca dyplomowa została napisana przeze mnie samodzielnie i nie zawiera treści uzyskanych w sposób niezgodny z obowiązującymi przepisami.
Oświadczam również, że przedstawiona praca nie była wcześniej przedmiotem procedur związanych z uzyskaniem tytułu zawodowego w wyższej uczelni.
Oświadczam ponadto, że niniejsza wersja pracy jest identyczna z załączoną wersją elektroniczną.

Data Podpis autora (autorów) pracy

Streszczenie

Celem, który postawiono w niniejszej rozprawie była konstrukcja narzędzia do przeprowadzania automatycznej klasyfikacji stadiów snu wykorzystującego sztuczne sieci neuronowe. W pracy zaprezentowano rozważania metodologiczne oraz przetestowano wiele architektur sieci konwolucyjnych i przedyskutowano ich właściwości. Najlepszy rezultat jaki uzyskano, to 68,3 % poprawnie sklasyfikowanych przypadków dla zbioru testowego liczącego kilkanaście tysięcy przykładów. Warto wziąć pod uwagę, że wynik możliwy do osiągnięcia był z góry ograniczony.

Słowa kluczowe

EEG, uczenie głębokie, deep learning, sieci neuronowe, sieci konwolucyjne

Dziedzina pracy (kody wg programu Socrates-Erasmus)

13.2 Fizyka

Tytuł pracy w języku angielskim

Analysis of EEG signals using deep neural networks

Spis treści

1. Wstęp

1.1. **Badania nad fizjologią snu.** Centralnym obiektem zainteresowań tej pracy jest sen. Myślę, że nikogo nie trzeba przekonywać co do jego znaczenia w prawidłowym funkcjonowaniu większości żywych istot. Pomimo wielkich postępów dokonanych w drodze do zrozumienia jego natury, wciąż jawi się jako mistyczny. Podobne odczucia towarzyszyły ludziom już w starożytności, gdzie fenomen snu podniesiono do boskiej rangi. Chociażby w starożytnej Grecji spotkamy się z bogiem *Hipnosem*, czy jeszcze wcześniej w starożytnym Egipcie z boginią *Tawaret*. Wraz z upływem czasu rozumienie snu podlegało naturalnej ewolucji. Przykładowo w 18 wieku za przyczynę jego powstawania podawano ciśnienie jakie krew wywiera na mózg, a na początku 20 wieku neurotoksynę, która gromadzi się w ciągu dnia i w nocy jest sukcesywnie usuwana [21]. Bardzo dużym problemem był brak odpowiednich narzędzi i pojęć do jego pomiaru, przez co również możliwości przeprowadzenia obiektywnych badań. Dużego postępu dokonano w połowie 20 wieku dzięki zastosowaniu w badaniach sygnału EEG, czyli metody do rejestracji fal mózgowych. Pozwoliło to po raz pierwszy na przeprowadzenie obserwacji snu bez zakłócania stanu badanej osoby. Wtedy to spojrzano na sen jako na aktywny proces, który podzielono na następujące po sobie stadia. Dało to również podstawy nowej dziedzinie medycyny badającej zaburzenia snu. Żeby uzmysłowić sobie jej znaczenie wystarczy zauważyć, że [4] [50–70] mln Amerykanów cierpi na tego typu schorzenia. Jest to jedna z motywacji tej pracy. Przejdźmy do krótkiego omówienia sygnału EEG, jako podstawowego obiektu badań.

1.2. **Sygnał EEG.** W dużym uproszczeniu i jak powszechnie wiadomo, mózg jest siecią połączeń neuronów, przez które nieustannie przepływają sygnały elektryczne. Sieć, o której mowa zawiera rzędu 10^{12} neuronów i 10^4 połączeń przypadających średnio na jeden neuron, co jest imponującą liczbą. Już w 19 wieku, bo w roku 1875 angielski lekarz Richard Caton z wykorzystaniem elektroskopu zauważył obecność różnic potencjałów na powierzchni kory mózgowej królików i małp. Dało to początek Elektroencefalografii, która koncentruje się na bioelektrycznej aktywności mózgu wywołanej głównie przez potencjały czynnościowe. Czytelnika zainteresowanego szczegółami powstawania i historią sygnału EEG odsyłam do [22]. Sygnał EEG posiada ogromną użyteczność w wielu dziedzinach. Znalazł zastosowanie między innymi w diagnostyce chorób – przykładowo zaburzeń snu, jako nośnik informacji w interfejsie mózg-komputer, badaniu wariografem. W przypadku zaburzeń snu sygnał EEG pozwala śledzić stadia snu śpiącej osoby. Na tej podstawie tworzony jest *hipnogram*, czyli wykres snu. Porównując uzyskany *hipnogram* z referencyjnym osoby zdrowej, lekarz jest w stanie wykryć odstępstwa od normy i podać diagnozę. W całym procesie leczenia bardzo istotnym jawi się moment oznaczania stadiów snu. Przejdźmy to sformułowania problemu badawczego.

1.3. **Formulacja problemu.** Jak zaznaczono powyżej, oznaczenie stadiów snu śpiącej osoby jest kluczowe na drodze diagnozy schorzeń snu. Praca jaką wykonują lekarze w tym celu jest powolna i uciążliwa. Stąd pomysł na stworzenie narzędzia do zautomatyzowania tego procesu. Doskonałym kandydatem do osiągnięcia tego celu jest komputer i dziedzina *sztucznej inteligencji*, która skupia się na tego typu problemach. Dlatego zdecydowano się na zastosowanie sztucznych sieci neuronowych w celu uzyskania automatycznej klasyfikacji stadiów snu. Stanowi to problem badawczy tej rozprawy.

Warto w tym miejscu dodać, że sztuczne sieci neuronowe i w szczególności konwolucyjne były już z sukcesem stosowane do analizy sygnału EEG. Przykładowo w publikacji [15] przedstawiono architektury służące do analizy EEG pod kątem zastosowania w interfejsie mózg-komputer. W kwestii klasyfikacji stadiów snu zaproponowano również wydajne rozwiązania. W pracy [27] przedstawiono sieć konwolucyjną opartą o jednowymiarową konwolucję jednokanałowego sygnału. Bardziej zbliżoną analizę do zaprezentowanej przeze mnie przedstawiono w [28]. Autorzy wykorzystali tam gotową architekturę sieci VGG i ideę *transfer learningu*. Danymi wejściowymi były reprezentacje czas-częstość poszczególnych fragmentów wielokanałowego sygnału EEG wyznaczone z wykorzystaniem wielookienkowej metody estymacji widma. Analizę przeprowadzono na kolorowych, tzn. trójkanałowych obrazkach. Tamtejsza metodologia mimo podobieństw, znacznie różni się od zaproponowanej w tej pracy. Przejdźmy do omówienia posiadanych danych.

2. Dane

Dane doświadczalne stanowi 109 całonocnych zapisów polisomnograficznych. Dane zostały zebrane w Laboratorium Snu Warszawskiego Uniwersytetu Medycznego, a stadia snu zostały opisane przez specjalistów z tego laboratorium. Do analizy wykorzystano zapisy (około 8h od każdej osoby) zebrane podczas drugiej nocy badania polisomografem. Osoby badane pochodziły z grup o następującej liczebności:

(1) bezsenność – 21 osób,
(2) nadciśnienie –11 osób,
(3) leki nasenne – 24 osoby,
(4) depresja – 9 osób,
(5) grupa referencyjna – 44 osoby.

Rejestrowanych było 28 kanałów sygnału: 21 kanałów elektroencefalograficznych (EEG) z systemu 10-20, dwóch kanałów z elektrod na płatkach uszu, elektrookulogram (EOG) - prawy i lewy, elektromiogram (EMG), elektrokardiogram (ECG) i sygnał z czujnika oddechu (RES). Do pomiaru sygnałów elektrofizjologicznych wykorzystano srebrne elektrody z pastą przewodzącą. Zebrane sygnały przefiltrowano filtrem sprzętowym o przepustowości [0.15–30] Hz. Następnie sygnał spróbkowano z częstością 128 Hz za pomocą 12-bitowego, analogowego konwertera. Maksymalna rezystancja wynosiła 5kΩ. Referencja sygnałów EEG została przeliczona do średniej wartości na elektrodach usznych. Ostatecznie analizę przeprowadzono na 26 kanałowym sygnale. Przykładowy fragment sygnału z kanału C4 przedstawiono poniżej na Rysunku 1.

Każdy 20 sekundowy odcinek snu został zaklasyfikowany przez lekarza do jednej z 8 kategorii:

(1) MUSC — wzmożona aktywność mięśniowa,
(2) WAKE — wybudzenie,
(3) STADIUM I — pierwsze stadium snu,
(4) REM — stadium snu z szybkimi ruchami gałek ocznych,
(5) STADIUM II — drugie stadium snu,
(6) STADIUM III — trzecie stadium snu,
(7) STADIUM IV — czwarte stadium snu,

zgodnie z kryteriami zawartymi w [12]. Łącznie mamy do dyspozycji 139214, 20-sto sekundowych odcinków snu z otagowaniem. Podkreślmy, że nie posiadamy dokładnych informacji co do samego procesu jego nadania. Nie jesteśmy w stanie stwierdzić ilu lekarzy brało udział w tym procesie i jaką zgodność prezentowali między sobą. Jednak dla podzbioru około 20 snów z naszej bazy i dwóch lekarzy biorących udział w badaniu wyniosła ona 76 % [18]. Zatem wynik jaki możemy uzyskać jest z góry ograniczony do podobnej wartości. Przejdźmy do omówienia narzędzi wykorzystanych w rozwiązaniu sformułowanego problemu.

3. Metodologia

3.1. **Podstawowe definicje.** W rozdziale tym przedstawiono ciąg logiczny prowadzący do rozwiązania problemu sformułowanego w podrozdziale 1.3. Ponadto ze względu na niezbyt obszerną i niekompletną polską literaturę postanowiłem przedstawić kilka podstawowych faktów z dziedziny analizy sygnałów i uczenia głębokiego (*ang. deep learningu*).

Jak uważny czytelnik zdążył zauważyć sygnał EEG jest sygnałem. Bardzo trafną i ogólną definicję tego pojęcia można znaleźć w [11]:

Definicja 1. *Sygnał jest czynnikiem przekazującym informację o zmiennym w czasie zachowaniu lub atrybucie pewnego zjawiska*

Powyższa definicja odnosi się do pojęcia abstrakcyjnego *a priori* reprezentowanego przez funkcję $s\colon \mathbb{R} \ni t \to s(t) \in \mathbb{R}$. W wyniku pomiaru otrzymujemy przeliczalną liczbę wartości funkcji s dla czasu z przedziału $[t_0, t_1]$, kiedy to przeprowadzono pomiar.

Wyróżniamy sygnały deterministyczne i stochastyczne. W pierwszym przypadku potrafimy podać formułę, która w każdej chwili czasu poda nam jego wartość. Druga sytuacja obowiązuje gdy brak jest dostatecznej wiedzy na temat mechanizmu powstawania sygnału lub jest on zbyt skompilowany do opisu explicite. Wtedy do opisu stosujemy aparat probabilistyczny. Zauważmy, że każdy sygnał można uznać za stochastyczny, bo determinizm to prawdopodobieństwo równe 1! W naszym przypadku wykorzystamy ogólny opis stochastyczny sygnału EEG – wartościami są zmienne w czasie napięcia elektryczne występujące na czaszce. Teraz krótkie przypomnienie z probabilistyki i teorii procesów stochastycznych.

Definicja 2. *Zmienna losowa* X *to odwzorowanie mierzalne z przestrzeni probabilistycznej* $\mathrm{X}\colon (\Omega, \mathcal{F}, \mathrm{P}) \to (\mathbb{R}^n, \mathcal{B}(\mathbb{R}^n))$*, gdzie* $\mathcal{B}(\mathbb{R}^n)$ *to* σ *ciało zbiorów borelowskich na* $\mathbb{R}^n$*. Mierzalność oznacza, że* $\bigwedge_{B\in\mathcal{B}(\mathbb{R}^n)} \mathrm{X}^{-1}(B) \in \mathcal{F}$*. Odwzorowanie* X *indukuje de facto nową przestrzeń probabilistyczną* $(\mathbb{R}^n, \mathcal{B}(\mathbb{R}^n), \mathrm{P}^{\mathrm{X}})$*, gdzie* $\bigwedge_{B\in\mathcal{B}(\mathbb{R}^n)} \mathrm{P}^{\mathrm{X}}(B) = \mathrm{P}(\mathrm{X}^{-1}(B))$.

Wszystkie zdarzenia związane ze zmienną losową są dobrze określone. Pomijamy definicję przestrzeni probabilistycznej jako obiektu podstawowego. Zauważmy, że zmienna losowa przyporządkowuje zdarzeniom losowym wartości rzeczywiste. Jak wpleść do probabilistyki zmienność w czasie, aby uzyskać związek z definicją sygnału? Odpowiedzi dostarcza poniższa:

Definicja 3. *Procesem stochastycznym* $\mathbb{Y}_{\mathrm{X}}(t)$ *nazywamy funkcję* $f(\mathrm{X}, t)$ *określoną na zmiennej losowej* X *i czasie* $t \in \mathbb{R}$*, która*

$$\bigwedge_{t_0\in\mathbb{R}} f(\mathrm{X}, t_0)$$

jest zmienną losową.

Możemy myśleć o procesie stochastycznym jako o rodzinie zmiennych losowych indeksowanych czasem. Jeśli wybierzemy zdarzenie $\omega \in \Omega$, to funkcja $f(\mathrm{X}(\omega), .) = f(x, .)\colon t \in \mathbb{R} \to \mathbb{R}$ jest realizacją procesu stochastycznego, którą nazywamy szeregiem czasowym. Szereg czasowy jest funkcją o pochodzeniu losowym, tzn. nadaje się do teoretycznej reprezentacji sygnału EEG. W dalszej analizie przyda nam się bardzo ważna

Definicja 4. *Proces stochastyczny nazwiemy stacjonarnym, gdy* $\bigwedge_{\tau\in\mathbb{R}} \bigwedge_{n,i\in\hat{1,n}}$

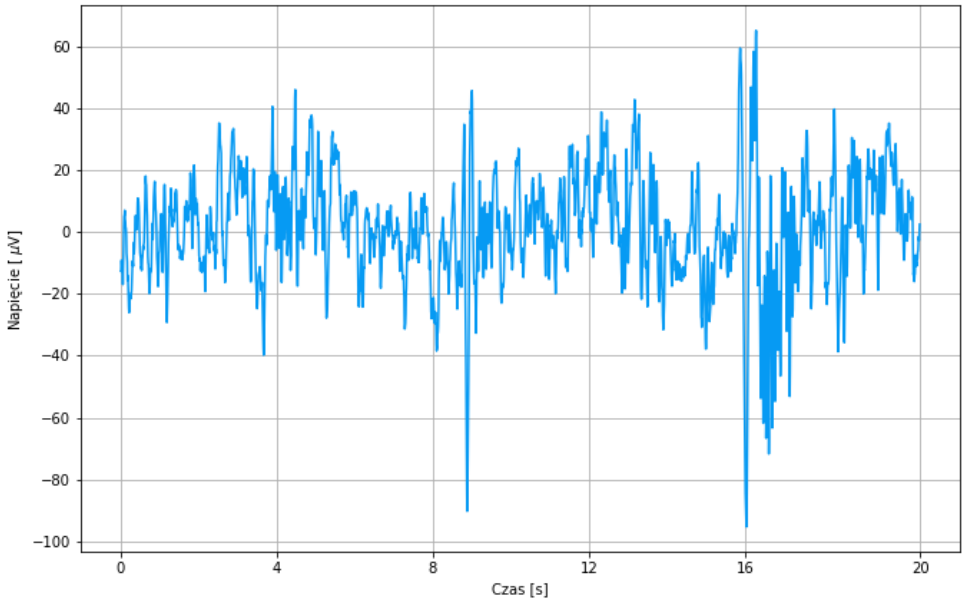
$$\mathbb{E}(\mathbb{Y}_{\mathrm{X}}(t_1+\tau)\mathbb{Y}_{\mathrm{X}}(t_2+\tau)...\mathbb{Y}_{\mathrm{X}}(t_n+\tau)) = \mathbb{E}(\mathbb{Y}_{\mathrm{X}}(t_1)\mathbb{Y}_{\mathrm{X}}(t_2)...\mathbb{Y}_{\mathrm{X}}(t_n))$$

Stacjonarność zapewnia możliwość estymacji punktowej momentów procesu stochastycznego w każdej chwili czasu z wykorzystaniem tylko jednej realizacji. Przykładowo estymatory próbkowe dwóch pierwszych momentów są następujące:

$$\hat{\mathbb{E}}(\mathbb{Y}_{\mathrm{X}}(t)) = \frac{1}{n}\sum_{i=1}^{n}\mathbb{Y}_{X=x_i}(t) = \frac{1}{mn}\sum_{j=1}^{m}\sum_{i=1}^{n}\mathbb{Y}_{X=x_i}(t_j) \overset{\mathrm{n=1}}{=} \frac{1}{m}\sum_{j=1}^{m}\mathbb{Y}_{X=x_1}(t_j),$$

$$\hat{\mathbb{E}}(\mathbb{Y}_{\mathrm{X}}(t)\mathbb{Y}_{\mathrm{X}}(t')) = \frac{1}{n}\sum_{i=1}^{n}\mathbb{Y}_{X=x_i}(t)\mathbb{Y}_{X=x_i}(t') = \frac{1}{mn}\sum_{j=1}^{m}\sum_{i=1}^{n}\mathbb{Y}_{X=x_i}(t_j)\mathbb{Y}_{X=x_i}(t_j+|t'-t|) \overset{\mathrm{n=1}}{=}$$
$$\frac{1}{m}\sum_{j=1}^{m}\mathbb{Y}_{X=x_1}(t_j)\mathbb{Y}_{X=x_1}(t_j+|t'-t|),$$

gdzie dla każdego m oraz j, t_j jest dowolne i takie, żeby powyższe równości miały sens dla posiadanych danych pomiarowych. Stacjonarność jest bardzo pożądaną własnością procesu, gdyż pojedyncza realizacja niesie całą informację o nim. W ogólności sygnał EEG nie jest stacjonarny, co można zauważyć na wykresie fragmentu snu z naszej bazy danych (Rys. 1).



Rysunek 1. Wykres przykładowego fragmentu sygnału EEG ze stadium II snu na kanale C4 względem średniej potencjału na uszach.

Sygnały dzielimy na przyjmujące wartości ciągłe i dyskretne. W drugim przypadku wartości przeciwdziedziny szeregu czasowego reprezentującego sygnał są dyskretne. Sygnał taki można otrzymać w wyniku próbkowania sygnału ciągłego, co w praktyce zawsze ma miejsce. De facto wszystkie sygnały, które przechowuje komputer są dyskretne. Próbkując sygnał ciągły chcemy w efekcie otrzymać wierną jego reprezentację. Nie uda się to w przypadku gdy próbkowanie jest zbyt rzadkie. Sposób w jaki należy próbkować sygnał ciągły, aby zachować całkowitą informację o nim zawarty jest w poniższym

Twierdzenie 3.1. *Nyquist–Shannon*

Niech sygnał będzie reprezentowany przez $s \in \mathrm{L}_1(\mathbb{R})$ *i* $\sum_{n\in\mathbb{Z}} |s(\frac{n}{2f_B})| < \infty$ *oraz supp* $\hat{s} \in [-f_B, f_B]$*, gdzie* $\hat{s}$ *jest transformatą Fouriera funkcji s. Wówczas*

$$s(t) = \sum_{n\in\mathbb{Z}} s\left(\frac{n}{2f_B}\right) \frac{\sin\left(2f_B(t-\frac{n}{2f_B})\right)}{2f_B\left(t-\frac{n}{2f_B}\right)}. \tag{3.1}$$

Powyższa równość oznacza zbieganie w przestrzeni $\mathrm{L}_1(\mathbb{R})$*.*

Dowód powyższego twierdzenia znajduje się w Dodatku A.

Maksymalne wartości częstotliwości sygnału bioelektrycznej aktywności mózgu nie powinny przekraczać 100 Hz. W przypadku snu oczekiwane częstości powinny być mniejsze niż 40 Hz. Sygnały jakimi dysponujemy zostały spróbkowane z częstością 128 Hz i w celu uniknięcia zakłóceń w aparaturze EEG zastosowano filtr dolnoprzepustowy. Na podstawie Twierdzenia 3.1 i wniosków zawartych w Dodatku A, wiemy, że dane które posiadamy są rzetelne.

3.2. **Reprezentacje czas-częstość.** Dla szeregów niestacjonarnych efektywnym opisem są reprezentacje czas-częstość. Z każdym sygnałem możemy stowarzyszyć pojęcie energii:

Definicja 5. *Jeżeli dla sygnału s(t) całka*

$$E_s = \int_{\mathbb{R}} |s(t)|^2 dt$$

jest skończona to $s(t)$ *jest o skończonej energii. W skrócie, wtedy* $s(t) \in \mathrm{L}_2(\mathbb{R})$*.*

Jeżeli dodatkowo $\hat{s} \in \mathrm{L}_2(\mathbb{R})$ to Twierdzenie Plancherela-Parsevala mówi, że

$$E_s = \int_{\mathbb{R}} |\hat{s}(f)|^2 df.$$

Dla $s(t)$ będącą funkcją z okresem $\frac{1}{f_0}$, mamy, że E$= \sum_{n\in\mathbb{Z}} |\hat{s}_n|^2$, gdzie $|\hat{s}_n|^2$ jest n-tym współczynnikiem Fouriera i energią związaną z częstością nf_0. Zatem $\{|s_n|^2\}_{n\in\mathbb{N}}$ definiuje sygnał z dokładnością do fazy. To co tracimy w opisie sygnału poprzez $\{|s_n|^2\}_{n\in\mathbb{N}}$ to informacja o czasie wystąpienia częstości nf_0. Nie ma to znaczenia w przypadku sygnałów stacjonarnych, które są nieczułe na przesunięcia w czasie. Dla sygnałów bez tej własności taki opis nie jest odpowiedni. Potrzebujemy wtedy znać energię sygnału dla częstości w zależności od czasu. Jeżeli sygnał, którym dysponujemy podzielimy na wiele części i do każdej z nich wyestymujemy widmo, to informacja o czasie nie zostanie w pełni stracona. Sposób w jaki należy podzielić sygnał opiszemy później. Właśnie to mamy na myśli mówiąc reprezentacja czas-częstość. Niestety opis w reprezentacji czas-częstość jest ograniczony w następującym sensie.

Definicja 6. *Niech* $s(t)$ *i* $\hat{s}(f) \in \mathrm{L}_2(\mathbb{R})$ *wówczas dwie pierwsze wielkości*

$$\begin{aligned}
m_s &= \frac{1}{E_s}\int_{\mathbb{R}} t|s(t)|^2 dt,\\
m_{\hat{s}} &= \frac{1}{E_s}\int_{\mathbb{R}} f|\hat{s}(f)|^2 df,\\
\sigma_s^2 &= \frac{1}{E_s}\int_{\mathbb{R}} (t-m_s)^2|s(t)|^2 dt,\\
\sigma_{\hat{s}^2} &= \frac{1}{E_s}\int_{\mathbb{R}} (f-m_{\hat{s}})^2|\hat{s}(f)|^2 df,
\end{aligned}$$

określamy mianem średniej sygnału i ostatnie jako wariancja, odpowiednio w reprezentacji czas i częstość.

Można udowodnić [3], że

Twierdzenie 3.2. *Nierówność Heisenberga*
Dla obiektów Definicji 6

$$\sigma_s \sigma_{\hat{s}} \geqslant \frac{1}{4\pi}. \tag{3.2}$$

Zatem im dokładniejszy jest opis widma w czasie, tym mniej dokładny jest on w dziedzinie częstości. Oznaczmy przez $s(t)$ analizowany sygnał, wówczas

$$s(t)w(t-b)$$

jest sygnałem wyciętym przez szybko znikającą poza chwilą b funkcję okienka $w(t)$. Wycięty sygnał zawiera głownie informację o $s(t)$ w chwili b.

Fakt 1. *Dla okienka gaussowskiego $w(t) \sim e^{-ct^2}$, gdzie c to stała, nierówność Heisenberga przechodzi w równość.*

Dla zadanego okna $w(t)$ zdefiniujmy funkcję

$$W_s(f,b) = \int\limits_{\mathbb{R}} s(t)w(t-b)e^{-2\pi i f t}dt = < s|w^*_{f,b} >,$$

gdzie $< | >$ oznacza iloczyn skalarny. Można udowodnić [3], że

Twierdzenie 3.3. *Gdy spełnione są warunki*

(1) $w \in \mathrm{L}_1(\mathbb{R}) \cap \mathrm{L}_2(\mathbb{R})$,
(2) $\int\limits_{\mathbb{R}} |w(t)|^2 dt = 1$,
(3) $|\hat{w}|$ *jest parzysta,*

dla każdego sygnału $s(t) \in \mathrm{L}_2(\mathbb{R})$

$$\iint\limits_{\mathbb{R}\times\mathbb{R}} |W_s(f,b)|^2 df db = \int\limits_{\mathbb{R}} |s(t)|^2 dt = E_s. \tag{3.3}$$

oraz

$$\lim_{A\to\infty} \int\limits_{\mathbb{R}} \left| s(t) - \iint\limits_{\{|f|<A\}\times\mathbb{R}} W_f(s,b) w_{f,b} df db \right|^2 dt = 0. \tag{3.4}$$

Warunek (3) w powyższym Twierdzeniu 3.3 jest spełniony gdy $w(t)$ jest rzeczywista. Zatem na podstawie $W_s(f,b)$ otrzymujemy rzetelną aproksymację $s(t)$.

Za Faktem 1, najbardziej rozsądne wydaję się wykorzystanie $w^*_{f,b}$ z gaussowską funkcją okienka i zastosowanie algorytmu *Matching Pursuit* [19]. Procedura taka jest jednak bardzo kosztowna obliczeniowo, dlatego wykorzystujemy reprezentacje falkową (ang. *wavelet representation*) opisaną poniżej.

Definicja 7. *Transformacją falkową funkcji* $s \in \mathrm{L}_2(\mathbb{R})$ *nazwiemy funkcję* $C_s \colon (\mathbb{R}-0)\times\mathbb{R} \to \mathbb{R}$

$$C_s(a,b) = < s|\psi_{a,b} > = \int\limits_{\mathbb{R}} s(t)\psi^*_{a,b}(t)dt, \text{ gdzie}$$

$$\psi_{a,b} = |a|^{-\frac{1}{2}} \psi\left(\frac{t-b}{a}\right), \ a,b \in \mathbb{R}, a \neq 0.$$

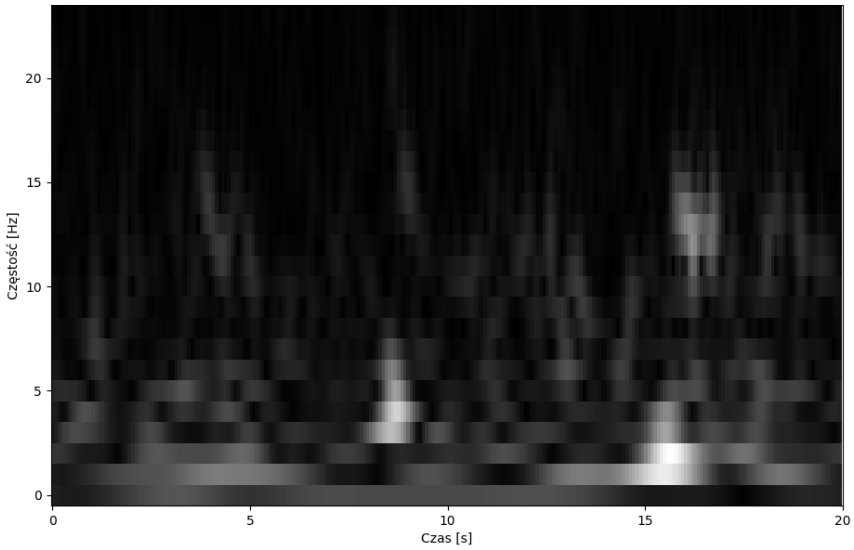
Falka $\psi_{a,b}(t)$ zadaje okienko [3] $\left[b \pm a(m_\psi - \sigma_\psi), \frac{m_{\hat{\psi}} \pm \sigma_{\hat{\psi}}}{a}\right]$. Parametr b zadaje lokalizację okienka w dziedzinie czasu, a parametr a, czyli *skala*, określa rozmiar okienka w czasie i przestrzeni skali a. Z dokładnością do stałych i założenia, że $\int\limits_{\mathbb{R}} \frac{|\hat{\psi}(\mu)|}{|\mu|} d\mu = K < \infty$, (3.3) jest w mocy oraz spełniona jest formuła odwrotna [3]

$$s_\epsilon(t) = \frac{1}{K} \iint\limits_{\{|a|>\epsilon\}\times\mathbb{R}} C_s(a,b)\psi_{a,b}(t)\frac{dadb}{a^2} \to s(t) \ w \ \mathrm{L}_2. \tag{3.5}$$

Istnieje duża dowolność w wyborze funkcji bazowych ψ. Naszym wyborem jest falka Morleta o składowej bazowej jak następuje

$$\psi(t) = \pi^{-0.25}(e^{iwt} - e^{-0.5w^2})e^{-0.5t^2}. \tag{3.6}$$

To co przemawia za wyborem falek Morleta, to nierówność Hesienberga, która przechodzi w równość [3]. Jest zatem optymalną reprezentacją w przestrzeni czas-częstość. Teraz wykorzystując (3.3) możemy dla każdego a, b wyznaczyć $|C_f(a, b)|^2$, czyli gęstość energii w reprezentacji czas-skala. W praktyce łatwo możemy przejść do reprezentacji czas-częstość, korzystając z relacji: $f = \frac{2awr}{M}$, gdzie f to częstość [Hz], a to współczynnik skalowania, r to częstość próbkowania [Hz], M długość sygnału, w to parametr falki Morleta. Zwiększając w zmniejszamy niepewność w dziedzinie częstości, ale zwiększamy w dziedzinie czasu. Zdecydowaliśmy się na w= 10. Taka wartość zapewnia dobrą rozdzielczość w dziedzinie częstości i użyteczną w dziedzinie czasu. Poniżej przedstawiono reprezentację czas-częstość sygnału z Rysunku 1, na której przeprowadzano analizę.



Rysunek 2. Wykres reprezentacji czas-częstość sygnału z Rysunku 1 uzyskany za pomocą ciągłej transformaty falkowej z parametrem $w = 10$.

Bardziej szczegółowy opis powstania tego obrazka zawarto w rozdziale Wyniki.

Z otrzymanych obrazków i z wykorzystaniem *konwolucyjnych sieci neuronowych* dokonano klasyfikacji stadiów snu. W kolejnym podrozdziale opisano dokładniej wykorzystywane narzędzie.

3.3. **Sztuczne Sieci Neuronowe i Uczenie Głębokie.** Dziedzinę Sztucznych Sieci Neuronowych od zawsze motywował ludzki mózg i jego własność generalizowania przetwarzanych informacji. Bardzo dobrze jest to widoczne w pierwszych dwóch akapitach pierwszego rozdziału przełomowej książki Davida E. Rumelharta i Jamesa L. McClellanda [25] z roku 1986, parafrazując:

> What makes people smarter than machines? They certainly are not quicker or more precise. Yet people are far better at perceiving objects in natural scenes and noting their relations, at understanding language and retrieving contextually appropriate information from memory, at making plans and carrying out contextually appropriate actions, and at a wide range of other natural cognitive tasks. People are also far better at learning to do these things more accurately and fluently through processing experience.

> What is the basis for these differences? One answer, perhaps the classic one we might expect from artificial intelligence, is "software" If we only had the right computer program, the argument goes, we might be able to capture the fluidity and adaptability of human information processing.

Przedstawiony cel jest z pewnością bardzo ambitny, a jego realizacja jeszcze odległa. Mimo to sieci neuronowe stały się potężnym **narzędziem inżynierskim** szeroko stosowanym, m.in w :

(1) analizie obrazu,
(2) analizie dźwięku,
(3) analizie szeregów czasowych,
(4) samosterujących samochodach.

Przejdźmy zatem do omówienia ich własności i sposobu w jaki zostaną wykorzystane w rozwiązaniu naszego problemu.

Obiektem naszych zainteresowań jest ludzki mózg. Załóżmy, że jest on systemem połączeń neuronów, czyli podstawowych jednostek przetwarzających i przekazujących informacje. Dlatego też punktem wyjścia w budowie modelu mózgu powinien być model samego neuronu. Pierwszy matematyczny schemat jego działania został zaproponowany przez Warren S. McCullocha i Walter Pittsa w roku 1943 [20]. Publikację tę uznaje się za początek całej dziedziny Sztucznych Sieci Neuronowych. Przedstawiony przez nich model można podsumować w formie prostego wzoru:

$$f(x) = \Theta(w^T x + w_0), \tag{3.7}$$

gdzie w, $x \in \mathbb{R}^n$ i w jest wektorem ustalanych wag, x jest wektorem sygnału wejściowego, w_0 to wyraz wolny, a $\Theta(.)$ to funkcja Heaviside'a. Model ten okaże się pożyteczny w dalszym ciągu pracy. Taki neuron, dla odpowiedniego wyboru wag jest zdolny reprezentować podstawowe funkcje logiczne, czyli rozwiązywać problemy separowalne liniowo. Kilka lat później bo w roku 1949 Donald O. Hebb [6] powiązał proces zmian wag neuronu z uczeniem. Przedstawiona przez niego formuła jest do dziś stosowana i określana jego nazwiskiem. Pozwoliło to spojrzeć na sztuczny neuron jako element obliczeniowy zdolny do generalizowania napływającej wiedzy w celu rozwiązania konkretnego problemu. Zaczęto traktować sztuczną sieć neuronową, czyli graf skierowany sztucznych neuronów jako narzędzie zdolne do przeprowadzania równoległych obliczeń. Żeby uwypuklić interpretację sieci neuronowej jako równoległej maszyny liczącej rozważmy prosty przykład (sekcja 3.3.1).

3.3.1. *Sztuczna sieć neuronowa jako maszyna licząca.* Niech dany będzie homoskedastyczny model liniowy: $\mathrm{Y}|\mathrm{X} = \mathrm{x} \sim \mathcal{N}(x\beta, \sigma^2 I)$, gdzie $\mathrm{x} \in \mathbb{R}^n$. Naszym zadaniem jest podanie parametru β. Wówczas $\hat{\beta} = (x^T x)^{-1} x^T Y$ jest estymatorem o najmniejszej wariancji w klasie estymatorów nieobciążonych. Zanotujmy, że $\hat{\beta} = \arg\min_{\beta} \mathbb{E}(\mathrm{Y} - x\beta)^2$ i estymatorem punktowym $\mathbb{E}(\mathrm{Y} - x\beta)^2$ jest

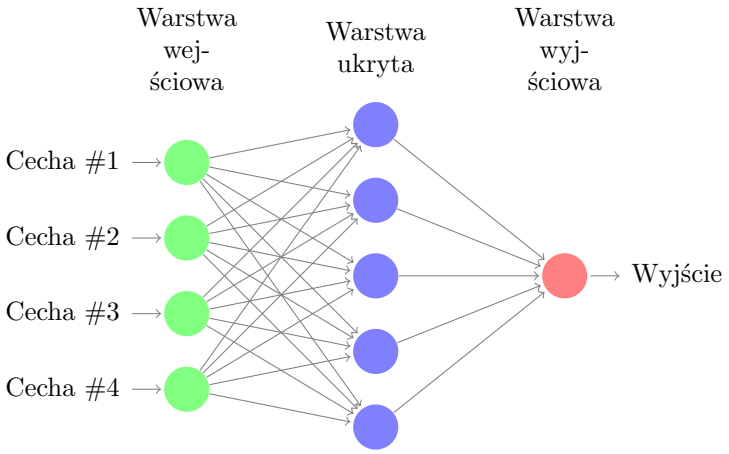
$$R(\beta) = \frac{1}{N} \sum_{i=1}^{N} (y_i - x_i \beta)^2, \tag{3.8}$$

gdzie y_i jest odpowiedzią dla cech x_i, a β nieznanym parametrem. Minimalizując ostatnią sumę znajdujemy poszukiwany estymator $\mathbb{E}(Y|X = x) = x\beta$, który jest odpowiedzią modelu dla zestawu cech x. Jak tego dokonać? Możliwe rozwiązanie tego problemu jest następujące:

(1) Wylosuj wektor β z dowolnego rozkładu
(2) Aktualizuj wagi $\beta_t = \beta_{t-1} - \eta \nabla_\beta R(\beta_{t-1})$ dla pewnej liczby naturalnej η
(3) Powtarzaj punkt (2) do spełnienia kryterium stopu np. gdy $R_t - R_{t-1} < 10^{-6}$

Dlatego, że funkcja R jest wypukła, algorytm zbiegnie dowolnie blisko rozwiązania prawidłowego w zależności od kryterium stopu. Powyższa procedura jest realizowana przez sieć neuronową, gdy funkcję θ zastąpimy przez id, gdzie $id(x) = x$ oraz algorytm zmian wag przejdzie na (2) w powyższym wyliczeniu. Przykłady (x_i, y_i) przekazywane są jeden po drugim. Gdybyśmy zainicjowali k jednostek do realizacji tego zadania z różnymi początkowymi parametrami β, wówczas

$\sum_{j=0}^{k-1} w_j \mathrm{id}(x_i\beta_j)$ zastąpiłby $x_i\beta$ w (3.8). Schemat sieci tego typu dla $n = 4$ i $k = 5$ przedstawiono poniżej:



RYSUNEK 3. Schemat jednowarstwowej sieci nueronowej. Na podstawie: http://www.texample.net/tikz/examples/neural-network/

W tym przypadku mamy $(n+1) * k + k + 1$ parametrów (wagi dla aktywacji z poprzedzającej warstwy i dla wyrazu stałego), które aktualizujemy gradientowo. Okazałoby się, że taka sieć szybciej zbiega do rozwiązania, a obliczenia na każdej jednostce są równoległe i mogą być realizowane jako mnożenie macierzowe.

Metoda uczenia przedstawiona w powyższym punkcie (2) jest współcześnie nazywana gradientową. Naturalnym uogólnieniem jest dodanie do sieci kolejnych warstw ukrytych, gdzie warstwą ukryta jest zdefiniowana na Rysunku 3. Wynikający z gradientowej reguły uczenia algorytm wstecznej propagacji został zaproponowany do zastosowania w sieciach neuronowych w roku 1986 [24]. Stał się źródłem przełomu, otworzył nowe możliwości i pchnął dziedzinę na bardziej współczesne tory.

Powyższy przykład miał uwypuklić interpretację sieci neuronowej jako obiektu przeprowadzającego obliczenia. Nie należy przy tym brać pod uwagę tego, że istniało rozwiązanie analityczne $\hat{\beta} = (x^T x)^{-1} x^T Y$. W poniższym akapicie przedstawiono inną i bardzo ważną własność- uniwersalnej aproksymacji sieci neuronowych. Pomijamy okres do lat 90, który został dość zgrabnie opisany w [26].

3.3.2. *Sztuczna sieć neuronowa jako uniwersalny aproksymator.* W latach 90 chciałbym skoncentrować uwagę czytelnika głównie na pracach statystyka Kurta Hornika. W roku 1989 Kurt Hornik wraz z współpracownikami [8] udowodnili, że zarówno wielo-, jak i jedno- warstwowa sztuczna sieć neuronowa jest uniwersalnym aproksymatorem w klasie rzeczywistych, borelowskich funkcji wektorowych dla naturalnej topologii, gdy w (3.7) zamiast Θ, funkcją aktywacji będzie dowolna *squashing function* Ψ, tzn. taka funkcja, że:

$$\Psi \colon \mathbb{R} \to [0,1]$$
$$\lim_{x \to \infty} \Psi(x) = 1$$
$$\lim_{x \to -\infty} \Psi(x) = 0.$$

Warto zauważyć, że Θ jest szczególnym przypadkiem Ψ. Jest to wspaniały wynik, wskazuje na istnienie architektury wielo- i jedno- warstwowej i takich wag, że aproksymacja z wykorzystaniem tej sieci i skończonej liczby przykładów x_i, y_i jest dowolnie blisko oryginalnej funkcji w sensie pewnej metryki. Zatem na podstawie danych treningowych jesteśmy w stanie aproksymować **rozkłady prawdopodobieństwa**. Z punktu widzenia wielu zastosowań jest to nieoceniona własność. Niestety nie było nic wiadomo na temat wpływu liczby jednostek w warstwie na stopień aproksymacji.

Liczba warstw nie była również teoretycznie określona. Następnie w pracy z roku 1990 Hornik [9] udowodnił już, że na dowolnym zwartym podzbiorze $\mathbb{R}^k$, X, wielowarstwowa sieć neuronowa z ciągłą i ograniczoną funkcją aktywacji pozwala w normie supremum aproksymować funkcje ciągłe na X. Ważnego uogólnienia dokonano później w pracy zatytułowanej "Multilayer Feedforward Networks With a Nonpolynomial Activation Function Can Approximate Any Function" [17]. Wykazano w niej, że sieci neuronowe o niewielomianowej funkcji aktywacji $\sigma \in \mathbb{L}^{\infty}_{\text{loc}}(\mathbb{R})$, dla której domknięcie zbioru nieciągłości ma zerową miarę Lebesque'a są gęste w zbiorze funkcji ciągłych $\mathbf{C}(\mathbb{R}^n)$. Zatem ograniczenia na funkcję aktywacji są minimalne. Pojawiły się również prace podające oszacowania na dokładność aproksymacji pewnych klas funkcji poprzez jednowarstwowe sieci neuronowe w zależności od liczby neuronów [2], [10]. Skoro jednowarstwowa sieć neuronowa jest uniwersalnym aproksymatorem dlaczego potrzebujemy wielu warstw - *uczenia głębokiego*?

W przypadku aktywacji Heaviside'a Θ sieci jednowarstwowe wyznaczają pojedyncze płaszczyzny decyzyjne, dwuwarstwowe obszary jednospójne i wypukłe, a trzywarstwowe – dowolne. Zwiększając liczbę warstw ukrytych reguła decyzyjna może przybrać praktycznie dowolny kształt. Zatem dodatkowe warstwy pozwalają na realizację coraz to bardziej złożonych powierzchni klasyfikacyjnych. Kolejnej odpowiedzi udzielono w [1], gdzie autorzy za wyższość wielowarstwowych sieci obarczają skończoną dokładność numeryczną komputerów. Wartość każdej funkcji jaką posługuje się komputer pochodzi ze skończonego rozwinięcia Taylora tej funkcji, czyli skończonego wielomianu. Oczywiste jest też, że sieć z aktywacją wielomianową pewnego stopnia będzie w stanie reprezentować tylko funkcje wielomianowe do tego rzędu. Zwiększenie liczby warstw powinno zwiększać maksymalny stopień wielomianowy jaką może odtworzyć sieć. Stąd obserwowana wyższość sieci wielowarstwowych. Jednak nie zostało to jeszcze powszechnie rozsądzone. W kwestii aproksymacji powyższe pytanie o liczbę warstw nie doczekało się jeszcze teoretycznie ugruntowanej odpowiedzi.

Warto tu przytoczyć jeszcze jedną, dość popularną interpretację sieci neuronowej, jako narzędzia pozwalającego na stworzenie odpowiedniejszej reprezentacji danych wejściowych. Minimalizując błąd, sieć dla zapostulowanej architektury odnajduje optymalną reprezentację wejścia. Kolejne warstwy można interpretować jako kolejne reprezentacje, które posłużą do wytworzenia ostatecznej, zawartej w warstwie końcowej, z wykorzystaniem której odbędzie się ostateczna analiza. Zakładając, że dane wejściowe są próbą z pewnego rozkładu, kolejne przekształcenia realizowane przez sieć neuronową prowadzą ostatecznie do wytworzenia nowej przestrzeni statystycznej w porównaniu do pierwotnej zdefiniowanej przez dane wejściowe. Zatem interpretacja zawarta w tym akapicie zawiera się w uniwersalnej własności aproksymacyjnej sieci neuronowej wszelakich funkcji, bo również rozkładów prawdopodobieństwa, a zatem i przestrzeni statystycznych.

Empirycznie obserwuje się, że sieci głębokie potrzebują mniej parametrów do osiągania tych samych wyników, są bardziej odporne na błędy w danych. Ponadto lepiej generalizują przetwarzane informacje. Uznajemy, że stosowanie *uczenia głębokiego* jest w pewnym stopniu uzasadnione. Jednym z problemów dotyczących głębokich sieci neuronowych jest tzw. "problem zanikających gradientów". Dodajmy, że funkcją aktywacji powszechnie stosowaną w latach 90 była sigmoida, dla której bardzo szybko gradient dąży do zera. W przypadku funkcji aktywacji, które szybko się nasycają, tzn. zerują gradient propagacja wsteczna zawiedzie i proces uczenia zatrzyma się. Jest to oczywiste, bo informacja o błędzie jest przekazywany wstecz z proporcjonalnością do gradientu [24]. Niech y^k będzie wektorem wyjściowym z k-tej warstwy i wejściem dla neuronów z warstwy kolejnej i N numerem warstwy ostatniej. Ponadto niech zadaniem sieci będzie minimalizacja ryzyka empirycznego $R(Y_{\text{true}}, Y_{\text{pred}})$, nazywaną często *funkcją straty*. Wówczas zachodzi poniższe

Z powodu nasycenia ostatnich warstw gradient na pierwszych warstwach będzie jeszcze mniejszy, a sam proces aktualizacji wag w celu minimalizacji empirycznego ryzyka R, zatrzymany. W roku 2006 Hinton rozwiązał ten problem dla sieci bayesowskich [7]. Współcześnie istnieje wiele metod pozwalających na ominięcie tego problemu. Głownie z tego powodu najczęściej stosowaną funkcją aktywacji jest funkcja

$$\text{ReLU}(x) = \max(0, x). \tag{3.9}$$

W kwestii zadań klasyfikacji najczęściej wykorzystywaną funkcją straty jest enropia krzyżowa (*ang. cross entropy*), która czerpie motywacje bezpośrednio z teorii informacji. Niech p i q będą gęstościami prawdopodobieństw określonymi na tym samym zbiorze, z którego losowane są

elementy tworzące zdarzenia. Jeżeli to losowanie odbywa się z rozkładu p, a jego opis następuje poprzez q, to średnia liczba bitów potrzebna do poprawnego zidentyfikowania zjawiska wynosi $H(p,q) = \mathrm{E}_p(-log_2 q)$ i określana jest mianem entropii wzajemnej. W kontekście uczenia sieci neuronowych rozkład p opisuje prawdziwe etykiety przykładów, zaś rozkład q aktualnie przewidywane przez model. Entropia krzyżowa jest miarą rozbieżności między dystrybucjami p i q, zaś celem uczenia jest jej minimalizacja. Więcej szczegółów można znaleźć w [23].

Wiemy, że sieć neuronowa jest narzędziem do przeprowadzania obliczeń o uniwersalnych zdolnościach aproksymacyjnych. Ważne jest też, że mówiąc sztuczna sieć neuronowa mamy na myśli skierowany graf pojedynczych neuronów. Arbitralne są: model pojedynczego neuronu, architektura połączeń, liczba jednostek i warstw ukrytych, funkcja ryzyka i algorytm aktualizacji wag. Wszystkie te czynniki wpływają na uzyskany wynik. W przypadku modelu neuronu współcześnie wykorzystuje się zaproponowany przez Warren S. McCullocha i Walter Pittsa z ogólna funkcją aktywacji f najczęściej funkcją ReLU (3.9). Jeżeli chodzi o resztę parametrów swobodnych jest ona zależna od problemu i niejednoznacznie zadana. Skupmy się teraz na sieciach splotowych, gdzie zostanie uwypuklone znaczenie liczby warstw ukrytych.

3.4. **Sieci Splotowe.** Splotowe sieci neuronowe mimo, że pojawiły się już 30 lat temu, bo w roku 1989 [16] ostatnio cieszą się szczególną popularnością. Głównie dzięki spektakularnym wynikom w wielu konkursach dotyczących analizy danych o zadanej topologii siatki. Chodzi tu przede wszystkim o szeregi czasowe i zdjęcia. Przykładowo sieć **AlexNet** [14] osiągnęła wynik o 10,8 % lepszy niż zdobywca drugiego miejsca. Poniżej krótko opisano sieć konwolucyjną wykorzystując elementy z [5]. Klasycznie, warstwy sieci konwolucyjnej można podzielić na: konwulucyjne, zwężające i w pełni połączone. Prześledźmy proces przekształcania danych przez sieć.

3.4.1. *Warstwa konwolucyjna.* Dla ustalenia uwagi przedmiotem analizy jest tensor V_{ljk}, gdzie l jest kanałem wejścia, j i k to numer wiersza i kolumny. W przypadku zdjęcia kolorowego $i \in \{1,2,3\}$. Warstwa konwolucyjna w najprostszym przypadku jest reprezentowana jest przez 'tensor' K_{ijmn}, który zadaje związek i-tego kanału wyjścia z j-tym kanałem wejścia, indeksy m i n przyjmują w praktyce wartości do 7. Dla każdego i, K_{ilmn} nazywamy i-tym filtrem i naszym zadaniem jest znaleźć ten tensor dla zadanej przez użytkownika liczby filtrów. W pierwszej warstwie zachodzi następujące przekształcenie:

$$Z_{ijk} = V_{l,m+j,n+k} K_i^{lmn} \tag{3.10}$$

Zastosowano tu konwencję Einsteina i numerację od 0, a indeksy j, k są takie, aby suma miała sens. Zatem filtr przesuwa się po wejściu mnożąc odpowiadające piksele przez swoje wagi. Oczywiście pojawia się duża dowolność w jaki sposób ma odbywać się to przesuwanie, bo np. dla pewnych s_1,s_2 (zwykle $s_{1,2} \in \{1,2,3\}$)

$$Z_{ijk} = V_{l,m+j*s_1,n+k*s_2} K_i^{lmn}, \tag{3.11}$$

zadaje przesunięcia o pewną liczbę pikseli dla każdej zmiany j lub k. Zabieg ten znany jest pod nazwą *padding.* Zauważmy również, że wymiar wyjścia Z_{ijk} ulega zmniejszeniu. Jeżeli będzie on indeksowany od 0, to

$$dim(Z) = dim(K)_1 \text{ x int}(\frac{dim(V)_1 - dim(K)_2}{s_1} + 1) \text{ x int}(\frac{dim(V)_2 - dim(K)_3}{s_2} + 1). \tag{3.12}$$

Przykład 1. *Jeśli dim(V)=3x28x28, dim(K)=10x3x5x5, $s_1 = s_2 = 1$ dim(Z)=10x24x24. Wynik możemy zinterpretować jako obrazek 24x24 z 10 kanałami. Stąd warstwa konwolucyjna pozwala na tworzenie nowych kanałów.*

W celu uniknięcia zmniejszania się wymiaru obrazka często na krawędziach $V_{i,j,k}$ dodaje się zera i zwiększa tym samym $dim(V)_1$ i $dim(V)_2$. Taki zabieg pozwala na przeprowadzenie splotu o tym samym rozmiarze, tzn. wymiar wyjścia jest taki sam jak wejścia. Dwie podstawowe własności wyróżniają warstwę konwolucyjną:

(1) współdzielenie parametrów,
(2) rzadkie oddziaływania.

Pierwsza z nich jest bezpośrednią konsekwencją (3.10). Te same parametry służą do przekształcania różnych obszarów wejścia. Dzięki temu filtr wyszukuje w wejściowych danych pewnych charakterystycznych 'cech', których uczy się reprezentować. Dobrym przykładem są krawędzie na zdjęciach. Zauważmy, że warstwa konwolucyjna ma $dim(K)_0 * dim(K)_1 * dim(K)_2 * dim(K)_3$ parametrów. Co dla przykładu z sekcji 3.3.1 daje 750 parametrów. Gdyby warstwa była zwykłą warstwą w pełni połączoną mielibyśmy 13547520 parametrów, aby uzyskać wyjście tego samego rozmiaru co w (3.10). Oszczędność w liczbie parametrów przeciwdziała przuczeniu. Druga własność wiąże się z $dim(K)_2$ i $dim(K)_3$, bo są one niewielkie i należą w praktyce do zbioru $\{1, .., 7\}$. Następnie najczęściej każdy piksel wyjścia Z_{ijk} jest przekształcany przez funkcję aktywacji f, a całość przechodzi dalej przez warstwę zwężającą, która zmniejsza wymiar przestrzenny.

3.4.2. *Warstwa zwężająca.* Dla zadanego obszaru wejścia, zwykle jest on niewielki i zachodzi z różnym skokiem s, obliczane są statystyki podsumowujące. Najczęściej stosuje się funkcję max. Na mocy jest wzór z (3.12). Warstwę zwężającą wyróżnia *ekwiwariantność*, czyli niewrażliwość na niewielkie przesunięcia wejścia. Jeśli wykorzystujemy funkcję max i dokonamy niewielkiego przesunięcia wejścia, to nie zmieniamy istotnie wyniku warstwy zwężającej. Jest to przydatna własność w przypadku zdjęć, bo niewielkie przesunięcia nie wpływają na ich zawartość. Te trzy cechy wyróżniają sieci splotowe. Sieć konwolucyjna jest w uproszczeniu ciągiem następujących po sobie warstw: konwolucyjnych i zwężających. Ostatecznie pojawiają się warstwy: spłaszczająca i w pełni połączona.

3.4.3. *Warstwa spłaszczająca i w pełni połączone.* Niech wynikiem przekształcenia wejścia V_{ijk} przez kompozycję warstw kowolucyjnych i zwężających będzie W_{pqr}. Warstwa spłaszczająca transformuje W_{pqr} w wektor o $p * q * r$ wierszach. Wartości pojawiające się na jej wyjściu to cechy, na podstawie których zostanie dokonana właściwa analiza np. klasyfikacja. Dalej pojawiają się warstwy w pełni połączone, takie jak w sieci w pełni połączonej. Ich liczba i liczba neuronów w nich zawarta są uzależnione od problemu. Zatem sieć konwolucyjna pozwala na tworzenie nowych cech dopasowanych do problemu i przeprowadzenie automatycznej analizy na nich. Jak zatem znaleźć właściwą architekturę? Niestety nie istnieje teoretycznie ugruntowana odpowiedź na to pytanie. Mimo to, gdy zostanie zapostulowana architektura sieci znalezienie K_{ilmn} dla każdej z warstw ukrytych nie stanowi problemu. Proces ten następuje na drodze spadku gradientowego, który został *explicite* opisany w [5]. Warto dodać, że przed rozpoczęciem procesu uczenia wartości tensora K_{ilmn} są inicjalizowane z dowolnie wybranego rozkładu. W naszym przypadku sieć splotowa została wykorzystana do stworzenia odpowiedniej reprezentacji dla reprezentacji czas-częstość 20-sto sekundowych odcinków snu w obrębie każdego kanału i przeprowadzeniu zadania klasyfikacji z wykorzystaniem *entropii krzyżowej* jako funkcji straty. Warstwa klasyfikacyjna występuje na końcu sieci i z góry ma ustaloną liczbę neuronów, dla naszego problemu $k = 8$. Dodajmy, że w dla zadań klasyfikacyjnych najczęściej stosowaną i użytą przeze mnie funkcją aktywacji w ostatniej warstwie jest *softmax*

$$\sigma(x)_j = \frac{e^{x\beta_j}}{\sum_{i=1}^{k} e^{x\beta_i}}, \tag{3.13}$$

gdzie $\sigma(x)_j$ jest aktywacją na j- tym neuronie reprezentującym j klasę, β_j zestawem wag tego neuronu, a k liczbą klas. Ostatecznie dla ustalonego przykładu wybieramy klasę, której neuron dał największą aktywację.

3.5. **Wstępne przetworzenie sygnałów.** Naszym zadaniem jest przeprowadzić jak najlepszą klasyfikację z wykorzystaniem głębokich sztucznych sieci neuronowych. W tym celu 20 sekundowe odcinki snu każdej osoby w obrębie każdego kanału przetransformowano do reprezentacji czas-częstość z wykorzystaniem reprezentacji falkowej z parametrem w= 10 (3.6). Taka wartość dla rozważanych sygnałów daje wystarczająco dobrą rozdzielczość w dziedzinie częstości i akceptowalną w czasie. Następnie wartość na każdym pikselu została przetransformowana funkcją pierwiastkową, aby zmniejszyć skośność rozkładu wartości pikseli (oryginalny rozkład jest prawo skośny). Reprezentacja czas-częstość sygnału z Rysunku 1 znajduje się na Rysunku 2. Ostatecznie dla każdego 20 s odcinka otrzymujemy obrazek 24x2560x26, gdzie 24 to zakres częstości: [1-25] Hz, 2560 to liczba punktów pomiarowych w czasie w danym kanale, a 26 to liczba kanałów. Następnie

wymiar czasu został przeskalowany 10 krotnie z wykorzystaniem interpolacji funkcjami sklejanymi rzędu 3 z wykorzystaniem biblioteki pythonowej *scipy*. Stąd otrzymujemy obrazek o wymiarze 24x256x26. Zakres częstości, który znalazł się w reprezentacji dobrano *ad hoc* z uwzględnieniem zakresu, który może pojawić się w sygnale EEG podczas snu. Ustalmy znaczenie pojęcia *kanał*, którego przykładem są kolory RGB w zwykłych kolorowych obrazkach. Do tej pory rolę tę odgrywało 26 kanałów sygnału. Jako, że skład częstościowy obrazka ma *a priori* największy wpływ na jego klasyfikację do konkretnej kategorii, rolę kanału przypisujemy teraz częstości. Ostatecznie dla każdego 20s odcinka snu otrzymujemy obrazek o wymiarze 26x256x24. Reprezentację sygnału z Rysunku 1 przedstawiono na Rysunku 2, gdzie dla lepszej widoczności zaprezentowano obrazek z odpowiednio przeskalowanym wymiarem czasu. Dodatkowe sygnały nie-EEG niosą *a priori* użyteczną informację, dlatego zostały uwzględnione w analizie. Wygenerowane obrazki stanowiły wejście dla sieci konwolucyjnej, która była klasyfikatorem.

3.6. **Miary dokładności.** W pierwszej kolejności zbiór danych został podzielony na część treningową, testową i walidacyjną. Podział został dokonany losowo w obrębie wszystkich 109 osób z równym prawdopodobieństwem wylosowania każdej osoby. W zbiorze treningowym znalazły się 75 osoby, walidacyjnym 21 i testowym 10. Trenowanie modelu odbywało się na zbiorze treningowym, część walidacyjna posłużyła do wyboru hiperparametrów modelu, a testowa przetestowaniu modelu. Zanotujmy, że dane służące walidacji zostały wybrane ze zbioru walidacyjnego przez rozpoczęciem uczenia. Otóż wybrano w sposób losowy kilka osób ze zbioru walidacyjnego i z tego zbioru wybrano losowo 624 obrazki. Miejmy również na uwadze, że zbiór treningowy liczył 94944 przykładów. Do oceny użyteczności ostatecznego modelu wykorzystana zostanie dokładność *ang. accuracy*. Jest to procent poprawnie sklasyfikowanych przykładów. Natomiast do oceny postępów uczenia i monitorowania generalizacji wykorzystano *skumulowane accuracy* zdefiniowane jak poniżej:

$$\mathrm{Acc}(n+1) = \begin{cases} \frac{1}{2}\left(\frac{x_0(n)+y_0(n)+x_0^{'}(n+1)}{x(n)+y(n)+x^{'}(n+1)} + \frac{x_0^{'}(n+1)}{x^{'}(n+1)}\right) & \text{dla zbioru treningowego,} \\ \frac{1}{2}\left(\frac{x_0(n+1)+y_0(n)+y_0^{'}(n+1)}{x(n+1)+y(n)+y^{'}(n+1)} + \frac{y_0^{'}(n+1)}{y^{'}(n+1)}\right) & \text{dla zbioru walidacyjnego,} \end{cases}$$

gdzie $\mathrm{Acc}(n)$ to *skumulowane accuracy* po n krokach uczenia, $x(n)$ i $x_0(n)$ to odpowiednio liczba przekazanych przykładów i poprawnie sklasyfikowanych do chwili n w zbiorze treningowym, zaś $x^{'}(n)$ i $x_0^{'}(n)$ to te same wielkości, ale dla danych pobranych w chwili n. Jeżeli chodzi o wielkości oznaczone jako y, to definiujemy je analogicznie, ale odnosimy do zbioru walidacyjnego.

Przejdźmy do omówienia uzyskanych wyników.

4. Wyniki

4.1. **Model pierwszy – przykładowy model z biblioteki tensorflow.** Budowę modelu rozpoczęto od próby wykorzystania architektury przedstawionej w samouczku sieci kowolucyjnych biblioteki *tensorflow* [https://www.tensorflow.org/tutorials/layers]. Jest to struktura z dwiema warstwami kowolucyjnymi, odpowiednio z 32 i 64 filtrami, gdzie rozmiar filtrów ustalono na [3,3]. Oryginalny rozmiar filtrów [5,5] zmieniono ze względu na lepsze wyniki osiągane w przypadku mniejszego ich rozmiaru. Więcej szczegółów związanych z tą architekturą czytelnik znaj

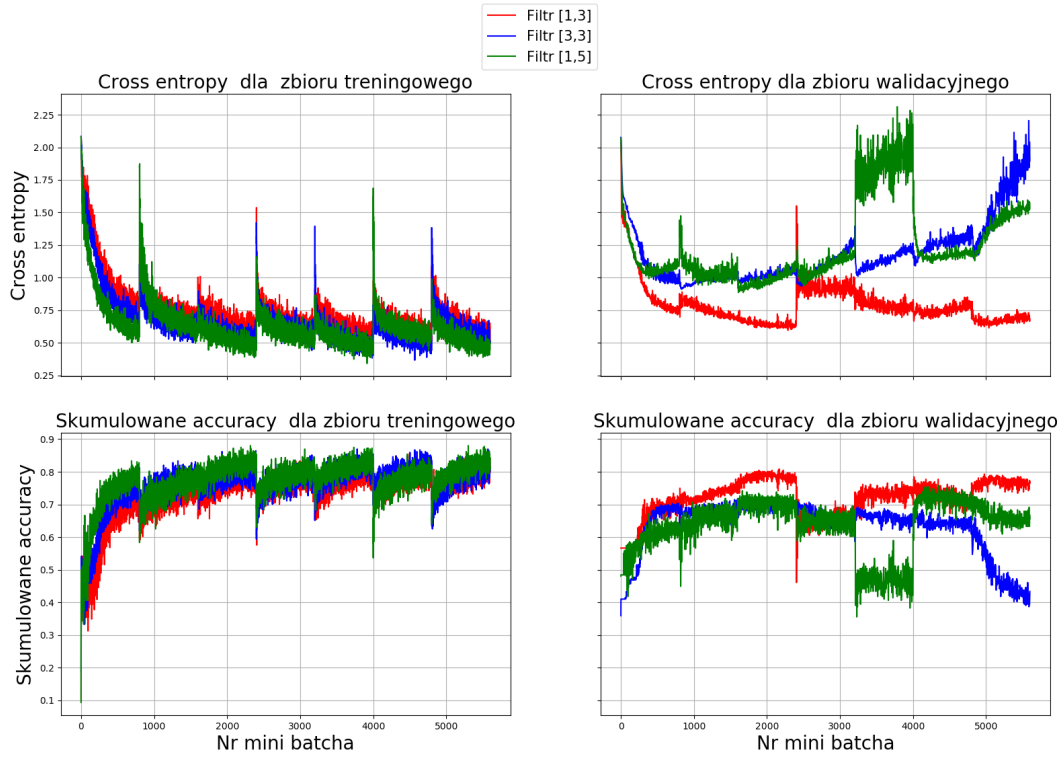
Pomimo zastosowania wielu technik regularyzacji model charakteryzuje się dużą zmiennością. Porażka techniki *batch normalisation* wskazuje, że sen posiada istotnie personalny charakter, dlatego próba normowania danych w obrębie pewnej grupy przynosi marne wyniki na innej. Ponadto pobieranie kolejnych *batchy* wiązało się z obniżeniem straty na zbiorze walidacyjnym, co uznajemy za pozytywne zjawisko. Architektura z filtrami o rozmiarze [1,3], tzn. filtrami które na poziomie konwolucji nie mieszają kanałów badania polisomnograficznego, wykazuje lepsze zachowanie na zbiorze walidacyjnym w porównaniu do wejściowego rozmiaru filtrów. Na powyższym Rysunku 4.1 nie uwzględniono kilku innych przetestowanych architektur z innymi wielkościami filtrów. Nie mniej jednak zwiększanie ich rozmiaru wpływało negatywnie na osiągane wyniki. Dlatego uznajemy, że filtry o rozmiarze [1,3] są o odpowiedniejszym rozmiarze niż początkowe. Warto w tym momencie zwrócić uwagę na wielkość zbioru danych jaki wykorzystano. Zbiór treningowy składał się z 94944 przykładów, dlatego rozsądnym wydaje się ograniczenie liczby trenowalnych parametrów do tej liczby. Przedstawiona powyżej struktura zawiera dziesiątki razy więcej parametrów trenowalnych, bo 25200488, co mogło być jednym z powodów jej nieregularności.

4.2. **Sekwencyjne dodawanie warstw.** Kolejnym pomysłem była sekwencyjna budowa modelu głębokiego warstwa po warstwie. Rozpoczęto zatem od budowy jednowarstwowej sieci konwolucyjnej. Wykorzystując wcześniej zdobyte intuicje ustalono parametry jak poniżej:

(1) warstwa konwolucyjna:
   (a) 5 filtrów [1,3],
   (b) padding *same*,
   (c) funkcja aktywacji *relu*,
(2) warstwa zwężająca:
   (a) rozmiar: [1,2],
   (b) przesunięcia: $[s_1, s_2] = [2, 2]$,
(3) warstwa płaska,
(4) warstwa w pełni połączona:
   (a) 24 jednostki,
   (b) funkcja aktywacji *relu*,
(5) warstwa w pełni połączona:
   (a) 8 jednostek,
   (b) funkcja aktywacji *softmax* (3.13).

Jako stratę wykorzystano entropię wzajemną. Optymalizatorem był algorytm *Adam* [13] z domyślnymi parametrami z biblioteki *tensorflow*. Zwiększono również 4 krotnie rozmiar *mini batcha*, co może pomóc w regularyzacji modelu. Liczba parametrów trenowalnych tego modelu zmalała do 200269. Zatem została znaczenie ograniczona. Zwiększono od tego momentu częstotliwość zmiany *batchy*. Następowała ona po pobraniu 800 *mini batchy*.

Pierwszym krokiem było ustalenie rozmiaru filtrów warstwy konwolucyjnej. W tym celu wykorzystując opisaną powyżej architekturę zbudowano kilka modeli różniących się wyłącznie rozmiarem zastosowanych filtrów. Na Rysunku 4 zaprezentowano uzyskane wyniki.

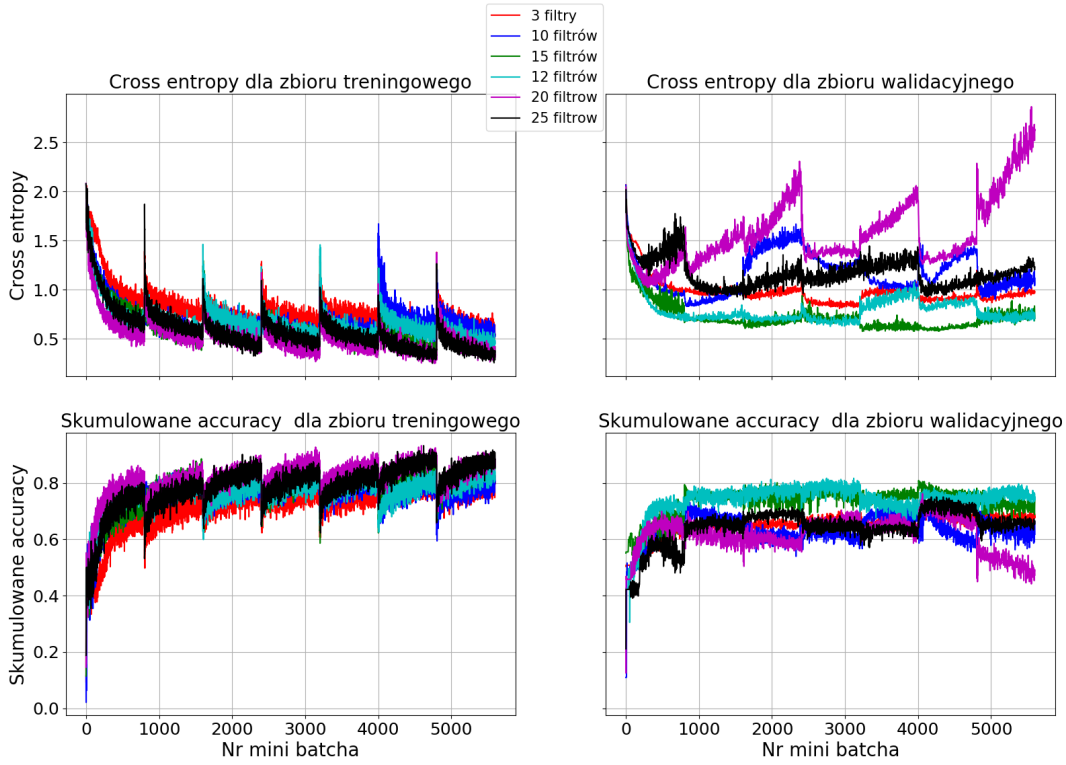


Rysunek 4. Wpływ wielkości filtrów na model pierwszy.

Filtry o rozmiarze [1,3] wydają się najodpowiedniejsze. Oprócz tego, że zachowanie błędu na zbiorze walidacyjnym ma tendencję spadkową, to również charakteryzuje się największą stabilnością miary *skumulowane accuracy*. W przypadku modelu z filtrami [1,5] zauważalny jest moment, w którym pobrano *batch* wpływający negatywnie na proces uczenia. Miejmy tę obserwację na uwadze. W kolejnym etapie manipulowano liczbą jednostek w przedostatniej warstwie w pełni połączonej. Ze względu na osiągnięte wyniki ich liczbę z 24 zmniejszono do 16. Pozwoliło to zredukować liczbę trenowalnych parametrów modelu do 133637. Następnym krokiem było ustalenie liczby filtrów w warstwie konwolucyjnej. Na poniższym Rysunku 5 zaprezentowano wyniki modelu dla różnej ich liczby.

Wynika stąd, że 12 filtrów wydaje się najodpowiedniejszą ich liczbą spośród przetestowanych. Dla ustalonego modelu, to jest z 16 jednostkami w pełni połączonymi na przed ostatniej warstwie i 12 filtrami [1,3] w warstwie konwolucyjnej, wypróbowano wiele technik regularyzacji. Manipulowano również algorytmem zbiegania do rozwiązania. Mimo to nie udało się zmniejszyć fluktuacji modelu. Kolejna próba uzyskania bardziej regularnego rozwiązania polegała na manipulacji wielkością *batcha*, rozmiarem i sposobie przekazywania *minibatchy* na wejście sieci. Pierwotny sposób okazał się dawać najlepsze wyniki. Uzyskany model został kilkukrotnie przeliczony. Okazało się, że nie jest stabilny. Wnioskujemy stąd, że skład *batcha* ma tu fundamentalne znaczenie. Dalej dla ustalonego składu *batchowego* dającego najlepsze wyniki zbadano znaczenie zawartości *mini batcha*. Również w tym przypadku dla kilku przeliczeń zaobserwowano znacznie różne charakterystyki. Potwierdza to ogólne przekonanie o znaczeniu sposobu przekazywania danych modelowi. Poniżej zaprezentowano główne wnioski jakie nasuwają się z przeprowadzonych do tej pory badań:

(1) zbyt duża liczba parametrów powoduje duże wahania charakterystyk modelu,
(2) sen jest bardzo indywidualnym procesem,
(3) techniki regularyzacji jawią się jako bezużyteczne,

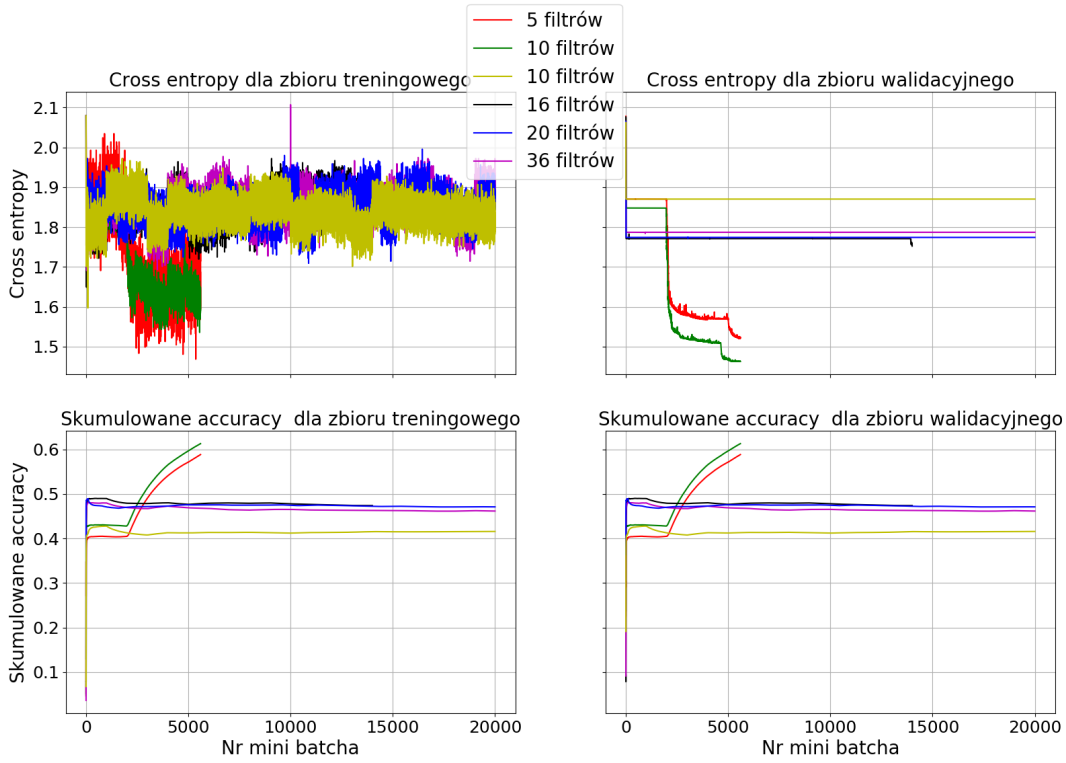


Rysunek 5. Wpływ różnej liczby filtrów na rozważany model

(4) sposób przekazywania danych na wejście modelu ma ogromne znaczenie na uzyskiwane wyniki.

4.3. **Podział objętości wejściowej w zależności od typu kanału.** Kolejnym i jak się okaże, udanym pomysłem było wykorzystanie oddzielnych filtrów do analizy kanałów EEG, EOG, EMG, ECG i RES. Początkowo rozmiar filtrów na kanałach EEG ustalono na [3,3], a na pozostałych na [1,3]. Po przeprowadzeniu konwolucji wyniki zostały ze sobą skonkatenowane, dlatego liczba filtrów była taka sama dla każdego z 5 typów kanałów wejściowych. Dalsza architektura sieci pozostaje bez zmian. Zanotujmy, że od tego momentu zwiększono częstotliwość zmiany *batchy*, która od teraz następowała po pobraniu 100 *mini batchy*. Poniżej, na Rysunku 6 zaprezentowano wykres kilku modeli różniących się liczbą filtrów.

W porównaniu do poprzednich wyników otrzymane modele charakteryzują się wręcz idealną regularnością. Dwa z nich wyróżniają się w oczywisty sposób. Ciekawy jest schodkowy charakter spadku miary *cross entropy*. Reszta modeli wykazuje stałą funkcją straty i miarą *skumulowane accuracy* na zbiorze walidacyjnym. Prawdopodobnie spowodowane jest to utknięciem w dolinie funkcji straty. Można by pomyśleć, że odpowiada za to zwiększona liczba filtrów, czyli parametrów trenowalnych, co nie jest do końca prawdą. Bardzo duże znaczenie ma inicjalizacja wartości początkowych i jak zaznaczono wcześniej sposób prezentowania danych modelowi. Doskonale widać to w przypadku modelu z 10 filtrami. W jednym przypadku proces uczenia utknął w dole potencjału, natomiast w drugim osiągnął zadowalające wyniki. Dodajmy, że manipulacja krokiem uczenia tj. jego zmniejszanie i zwiększanie, dodatkowe techniki jego przyspieszania, to jest *exponential learning rate decay* z początkowym dużym krokiem uczenia oraz manipulacja wielkością *mini batchy* nie zwiększała prawdopodobieństwa wyjścia procesu uczenia z *plateau*. Żeby zaradzić temu problemowi zwiększono rozmiar filtrów na kanale EEG, początkowo do [3,20], przez co filtry obejmowały większy obszar czasu podczas konwolucji. Wydaje się to sensownym zabiegiem, bo



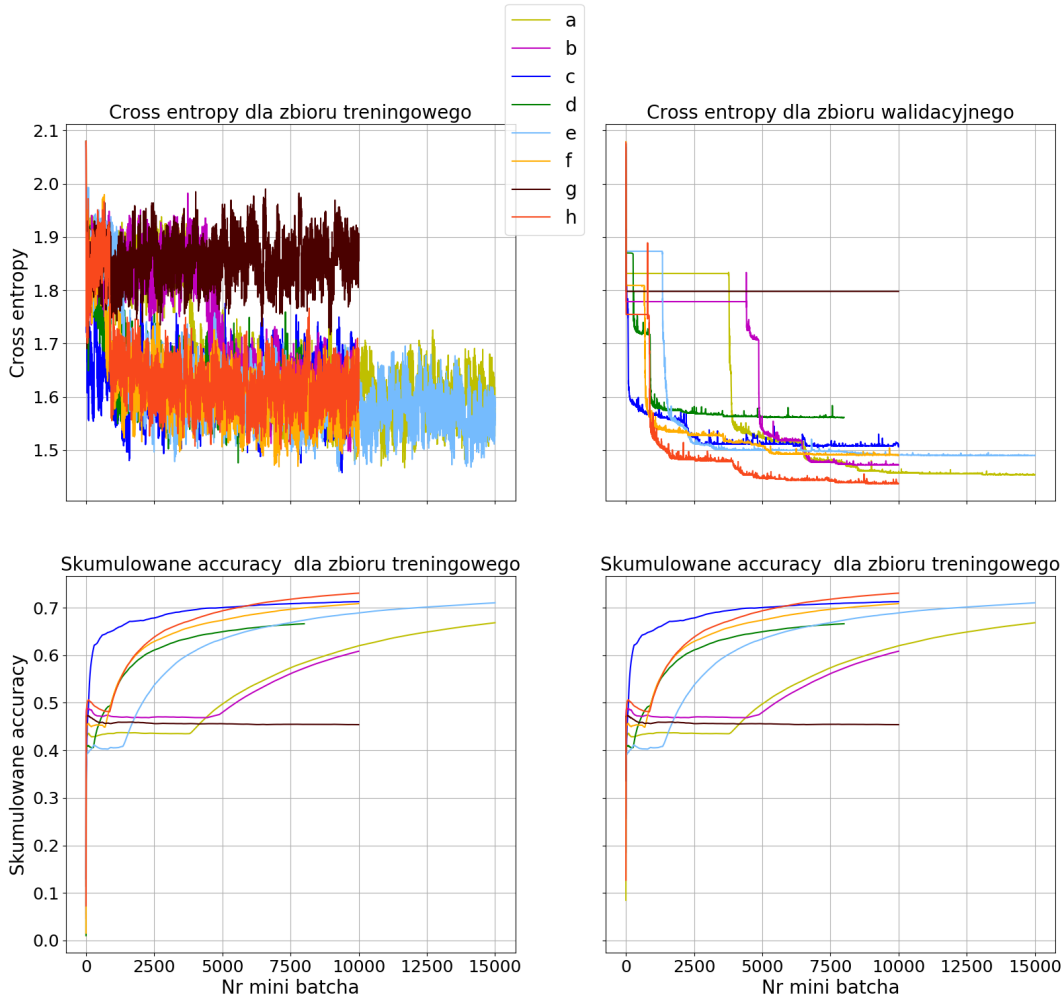
RYSUNEK 6. Wpływ oddzielnej konwolucji w obrębie poszczególnych kanałów.

obszary energetyczne mogące świadczyć o stadium snu obrazka są rozciągłe w czasie i zajmują zwykle czas rzędu $1s$. Zwiększa to niestety liczbę trenowalnych parametrów modelu. Równocześnie przetestowano dwie nowe metody przekazywania przykładów. W pierwszej z nich z równym prawdopodobieństwem losowano jedną z pięciu dostępnych grup:

(1) bezsenność,
(2) nadciśnienie,
(3) leki,
(4) depresja,
(5) grupa referencyjna,

następnie w obrębie wylosowanej grupy wybierano jednego pacjenta również z równym prawdopodobieństwem wylosowania każdego z nich. Losowanie kolejnej osoby odbywało się z grupy innej niż poprzednio wylosowana. W drugiej metodzie dla ustalonego składu *batcha* z równym prawdopodobieństwem wybierano: jedno z 8 stadiów, następnie obrazek z *batcha* sklasyfikowany jako to stadium. Kolejne, różne od poprzedniego stadium wybierano losowo z prawdopodobieństwem przejścia wyestymowanym empirycznie z danych stanowiących zbiór treningowy. Jako, że niektóre przejścia były nieobserwowane, np. wake → stadium II, całą procedurę przerywano po kilku przejściach i ponownie losowano jedno z 8 stadiów. Metoda ta odtwarza w pewnym stopniu czasową strukturę snu, dzięki czemu mamy nadzieję na rzadsze grzęźnięcie modelu w dole potencjału. Poniżej na Rysunku 7 zaprezentowano kilka z przetestowanych modeli, gdzie liczba filtrów warstwy konwolucyjnej została ustalona na 10, ze względu na charakterystyki modeli z Rysunku 6.

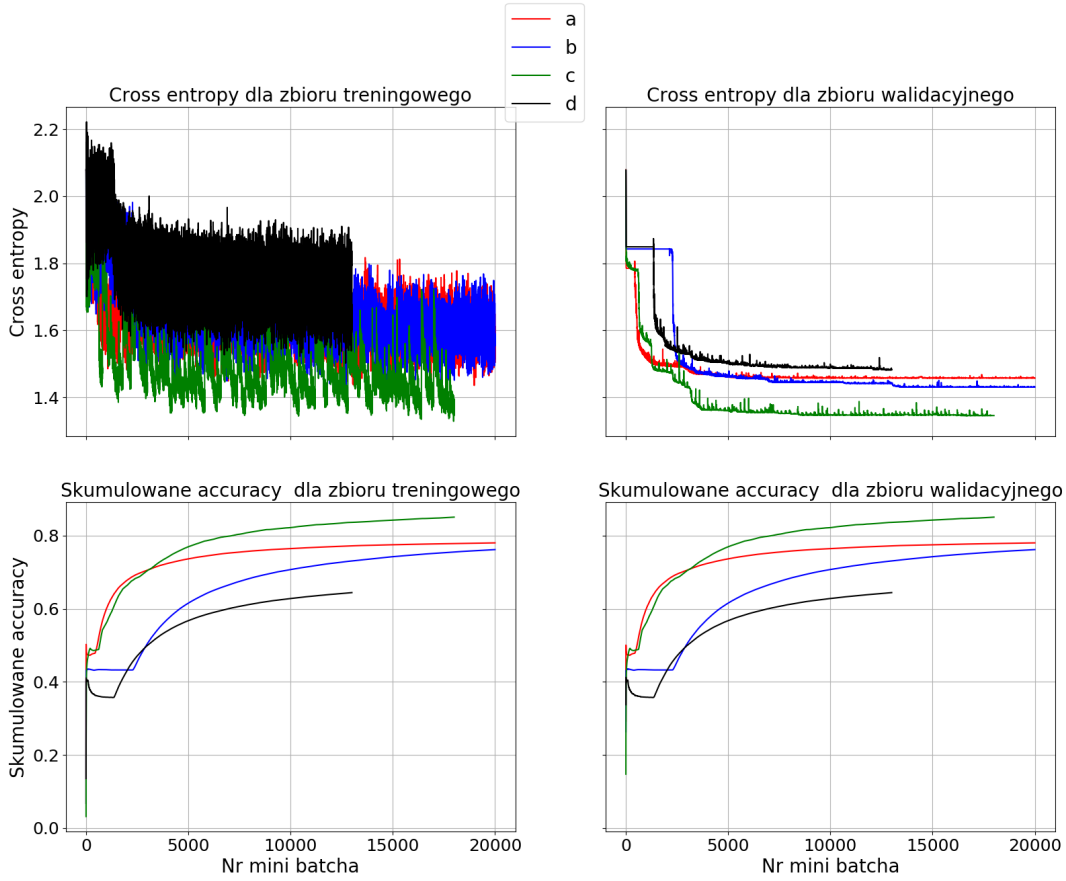
Spójrzmy na Rysunek 7. Dodajmy, że w legendzie określenie 'Filtry' odnosi się do pierwszej warstwy konwolucyjnej i kanałów EEG. W przypadku architektur z dwoma warstwami konwolucyjnymi rozmiar filtra nawiązuje do kanałów EEG w pierwszej warstwy konwolucyjnej oraz do drugiej warstwy konwolucyjnej. Dla pozostałych klas kanałów rozmiar filtrów wynosił [1,3] we wszystkich architekturach. Skupmy tymczasowo uwagę na dwóch pierwszych wykresach względem



RYSUNEK 7. Test metod poprawiających zbieżność, gdzie: a – 'Filtry[3,3] + łańcuch sny', b – 'Filtry[3,3] + łańcuch sny', c – 'Filtry [3,20] ', d – ' Tylko EEG+Filtry [3,20]+łańcuch sny ', e –'Filtry [3,12]+ łańcuch sny', f – 'Filtry [3,12]+łańcuch obrazki-długość 5', g – '10 i 20 filtrów [3,12] łańcuch sny + łańcuch obrazki', h – '5 i 10 filtrów [3,12] + łańcuch sny+ łańcuch obrazki

legendy. Są to charakterystyki takiego samego modelu, który został przeliczony dwukrotnie, a wniosek jaki się nasuwa to: powtarzalność, za którą chcemy obarczać pierwszą z kolei opisaną powyżej metodę przekazywania danych, oznaczoną jako 'łańcuch sny'. Rzeczywiście gdy spojrzymy na uzyskane charakterystyki dla obu zbiorów: treningowego i walidacyjnego, zauważymy wyraźne podobieństwo. Zatem potencjalnie nowy sposób tworzenia *batchy* pozwala osiągnąć większą powtarzalność uzyskiwanych wyników. Drugą metodę przekazywania danych nazwano 'łańcuch obrazki'. Kolejna para wykresów zdaje się potwierdzać zwarte na początku tej sekcji twierdzenie o użyteczności dodatkowych kanałów nie-EEG. Na uwagę zasługuje wykres zarówno funkcji straty, jak również miary *skumulowane accuracy* dla modelu z filtrami o rozmiarze [3,20]. Nie można tutaj stwierdzić, że model utknął w *plateau*. Zatem zwiększenie wymiaru czasowego filtra jest potencjalnie dobrym pomysłem. Kolejne dwa modele z jedną warstwą konwolucyjną i z filtrami o rozmiarze [3,12] wskazują na wyższość drugiej metody podawania danych oraz użyteczność filtrów o rozmiarze [3,12]. W końcu ostatnia para to struktury z dwiema warstwami konwolucyjnymi. Dla

pierwszej, czyli z większą liczbą filtrów nie obserwujemy efektów uczenia. Sytuacja przeciwna jest dla struktury z 5 i 10 filtrami na warstwach konwolucyjnych. Potencjalnie za tę sytuację odpowiada różna liczba parametrów. Nadmieńmy, że dla architektur z więcej niż dwoma warstwami konwolucyjnymi nie udało się opuścić *plateau*. Możliwe, że jest to spowodowane problemem zanikających gradientów. Zwróćmy jeszcze uwagę na Rysunek 6 i model z 10 filtrami. Jak zauważono wcześniej tylko jeden z nich jest użyteczny, a fakt braku efektów uczenia ma wyraźnie losowy charakter. Z tego powodu najbardziej obiecujące modele zostały kilkukrotnie przeliczone. Poniżej na Rysunku 8, dla każdego przeliczonego modelu, zaprezentowano najlepsze uzyskane próby. Dodajmy, że opis



Rysunek 8. Najlepsze znalezione modele, gdzie a –'5 filtrów [3,3]', b– '10 filtrów [3,3]', c– '20 filtrów [3,3] EEG', d–'5 i 10 filtrów [3,12]+ łańcuch sny+ łańcuch obrazki'

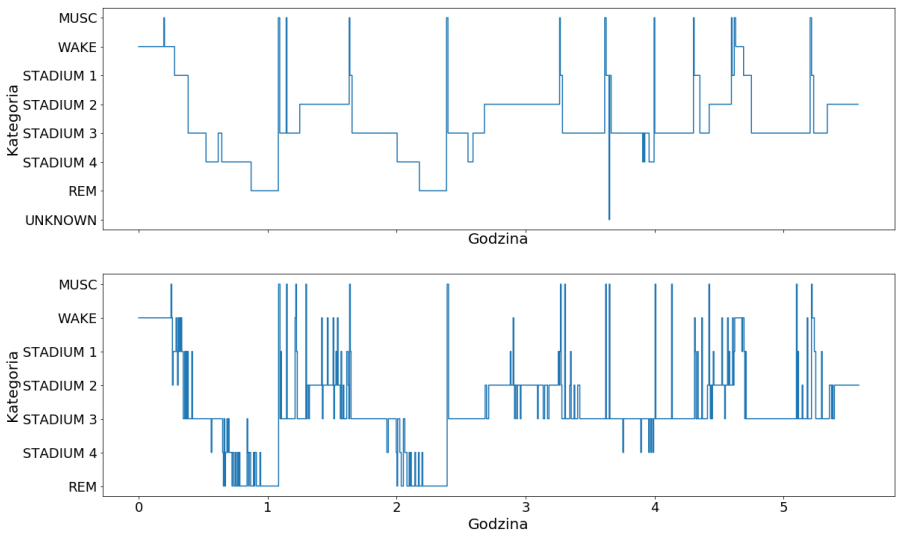
legendy jest zgodny z tym z Rysunku 7. Jak wynika z powyższego, najlepszym modelem okazała się sieć z jedną warstwą konwolucyjną uwzględniająca tylko sygnały EEG. Zauważalny jest również bardziej regularny charakter funkcji straty na zbiorze treningowym. Związane jest to z nadmiarem informacji zawartym we wszystkich kanałach. Spójrzmy jeszcze na tabele krzyżową modelu dla zbioru testowego.

Dla powyższej Tabeli 1 otrzymujemy 66,8 % poprawnie sklasyfikowanych przypadków. Warto też dodać, że w obrębie osób uzyskiwano bardzo różniące się wyniki. Najlepszy wynik poprawnie sklasyfikowanych przypadków osiągnięty w 13 osobowej grupie testowej wynosił 77,9 %, najgorszy 53,4 %, natomiast mediana wyniosła 68,3 %. Taka relacja wcale nie jest zadziwiająca biorąc pod uwagę wcześniejszą obserwacje o wyjątkowo personalnym charakterze snu. Doskonale było to widać

	UNKNOWN	REM	STADIUM 1	STADIUM 2	STADIUM 3	STADIUM 4	WAKE	MUSC
UNKNOWN	0	10	4	23	9	4	7	2
REM	0	902	430	166	3	3	0	13
STADIUM 1	0	507	492	656	5	11	5	17
STADIUM 2	0	170	391	6053	355	193	60	125
STADIUM 3	0	4	23	705	1879	205	62	34
STADIUM 4	0	0	1	385	120	195	410	55
WAKE	0	3	0	17	26	90	1520	101
MUSC	0	1	3	40	9	13	125	235

TABELA 1. Tabla krzyżowa dla najlepszego modelu z Rysunku 7. Wiersze odpowiadają prawdziwym wartościom, zaś kolumny to przewidywania modelu.

w momencie porażki techniki *batch normalisation*. Jak widać w Tabeli 1, klasa 'UKNOWN' jest jedyną, dla której nie udało się przewidzieć poprawnie żadnego przypadku. Wynika to bezpośrednio z niewielkiej liczby dostępnych przykładów dla tej konkretnej klasy. Nie bez usprawiedliwienia pozostał by zabieg usunięcia tych przypadków ze zbioru treningowego, co potencjalnie dałoby szanse na lepsze wyniki. Warto również podkreślić, że model wykazywał dużą tendencję do klasyfikacji przykładów z konkretnej klasy do sąsiednich, np. Stadium 3 jako Stadium 2 i Stadium 1. Zadziwiająca pozostaje duża mylność modelu w przypadku klasyfikacji Stadium 4, które najczęściej było mylone z fazą Wake, czyli wybudzeniem. Fakt ten będzie wymagał dalszej analizy. Stadium 'Wake' jako bardzo charakterystyczne, bo cechujące się szumem o dużej amplitudzie, zostało bardzo dobrze sklasyfikowane. Poniżej na Rysunku 9 zaprezentowano hipnogram dla osoby z najlepszym wynikiem klasyfikacji osiąganym przez model. Porównując hipnogram stworzony przez eksperta



RYSUNEK 9. Hipnogramy pewnej osoby, pierwszy z nich przewidziany przez lekarza, drugi przez model.

i najlepszy z uzyskanych modeli widzimy, że hoć oba wykazują bardzo podobną strukturę ogólną, to hipnogram ekspercki jest znacznie gładszy. Może to wskazywać na fakt, iż ekspert bierze pod uwagę to jakie stadia występowały w poprzedzających chwilach czasu i korzysta z wiedzy o prawdopodobieństwach przejść między stadiami. Tych dodatkowych informacji nie uwzględniano w ostatecznym modelu.

5. Dyskusja

Podsumowując, w pracy zaprezentowano wiele głębokich architektur sieci konwolucyjnych użytecznych w klasyfikacji stadiów snu. Najlepszy model sklasyfikował poprawnie 68,3 % przykładów ze zbioru testowego liczącego 16940, 20 s odcinków snu. Należy przy tym uwzględnić jakość posiadanych danych. Przykładowo zgodność pomiędzy dwoma lekarzami biorącymi udział w tagowaniu snów wynosiła 76 %. Nie posiadamy informacji dla reszty personelu badawczego biorącego udział w badaniu. Stanowi to poważne ograniczenie narzucone na jakąkolwiek automatyczną analizę. Warto w tym miejscu odwołać się do [18], gdzie przedstawiono algorytm do klasyfikacji stadiów snu w formie drzewa binarnego. Przeprowadzona tam analiza odbyła się dla części (ok 25 %) wykorzystanego przeze mnie zbioru danych i dawała 73 % poprawnie sklasyfikowanych przypadków. Nie można jednak porównywać obu modeli ze względu na różne zbiory danych, na których przeprowadzono obie analizy.

Zaprezentowana przeze mnie analiza z pewnością wykazuje potencjał. Zaskakująco, najlepszy wynik dała architektura z jedną warstwą konwolucyjną. Możliwe, że spowodowane jest to zastosowaniem w analizie reprezentacji czas-częstość, która stanowi znaczą edycję danych wejściowych. Jednak bardzo istotna jest jakość dostępnych danych, które z góry warunkują możliwy do osiągnięcia wynik. Warto też wyciągnąć wnioski odnośnie samego narzędzia jakie zastosowałem. Z pewnością metody *uczenia głębokiego* są bardzo użyteczne, ale jeszcze nie zrozumiane. Bardzo duża swoboda w budowie całego modelu sieci neuronowej, brak odpowiednich pojęć i dogłębnego zrozumienia istoty ich działania wskazują, że ta obiecująca dziedzina znajduje się dopiero na początkowym stadium rozwoju. Autor wyraża tym samym ogromną potrzebę i nadzieję na lepsze zrozumienie tego obszaru, póki co **inżynierskich** rozważań, budowę nowych pojęć i wyznaczanie nowych kierunków badań, co poskutkuje lawinowym wzrostem zastosowań i zauważalnym przełomem w życiu każdego z nas.

Dodatek A

W dowodzie bardzo pomocny okaże się poniższy lemat będący Twierdzeniem 2.2 w [3]

Lemat .1. *Niech $s(t) \in \mathrm{L}_1([0,T])$ i współczynniki Fouriera $\{\hat{s}_n\}$ spełniają:*

$$\sum_{n\in\mathbb{Z}} |\hat{s}_n| < \infty,$$

wówczas dla prawie każdego $t \in \mathbb{R}$

$$s(t) = \sum_{n\in\mathbb{Z}} \hat{s}_n e^{2\pi i \frac{n}{T} t}.$$

Jeżeli założymy dodatkowo, że $s(t)$ jest ciągła to równość zachodzi dla każdego t.

Fakt 2. *Sygnał reprezentowany przez funckję z $\mathrm{L}_1(\mathbb{R})$ nazywamy stabilnym, a gdy z $\mathrm{L}_1(T)$ na odcinku T, lokalnie stabilnym.*

Dowód. Twierdzenia Nyguista-Shannona

Skoro, $\hat{s}$ ma zwarty nośnik i z Lematu Riemanna-Lebesgue'a wynika, że jest ciągła, więc jest całkowalna, a nawet $\hat{s} \in \mathrm{L}_2([-f_B, f_B])$. Wykorzystując odwrotną transformatę Fouriera:

$$s(x) = \int_{-f_B}^{f_B} \hat{s}(f) e^{2\pi i f x} dk.$$

Wynika stąd, że dla s istnieje pochodna dowolnego rzędu, a z Lematu Riemanna-Lebesgue'a, że s jest ciągła. Skoro s jest ciągła, to $\bigwedge_{x\in\mathbb{R}} s(x)$ jest jednoznacznie określona, dlatego w powyższym twierdzeniu $s(\frac{n}{2f_B})$ ma sens . Ponieważ $\hat{s} \in \mathrm{L}_1([-f_B, f_B])$ to możemy ją rozwinąć w szereg Fouriera:

$$\sum_{n\in\mathbb{Z}} b_n e^{\frac{i\pi n f}{f_B}}, \tag{.1}$$

gdzie

$$\begin{aligned} b_m &= \frac{1}{2f_B} \langle \hat{s} | e^{\frac{-i\pi m f}{f_B}} \rangle \\ &= \frac{1}{2f_B} \int_{\infty}^{\infty} \hat{s}(f) e^{\frac{-2\pi i m f}{2f_B}} dk = \frac{1}{2f_B} s(\frac{-m}{2f_B}) \end{aligned}$$

W ostatniej równości wykorzystano fakt, że funkcja s posiada transformatę Fouriera. Na mocy Lematu .1 i założenia

$$\hat{s}(f) = \sum_{n\in\mathbb{Z}} \frac{1}{2f_B} s(\frac{n}{2f_B}) e^{\frac{-i\pi n f}{f_B}}, \tag{.2}$$

gdzie równość oznacza zbieżność w $\mathrm{L}_1([-f_B, f_B])$. Aplikując do (.2) odwrotną transformatę Fouriera otrzymujemy

$$\begin{aligned} s(x) = &\sum_{n\in\mathbb{Z}} \frac{1}{2f_B} s(\frac{n}{2f_B}) \int_{-f_B}^{f_B} e^{2i\pi f(x - \frac{n}{2f_B})} df = \\ &\sum_{n\in\mathbb{Z}} \frac{1}{2\pi i f_B (x - \frac{n}{f_B})} s(\frac{n}{2f_B}) [e^{2i\pi(x - \frac{n}{2f_B}) f_B} - e^{-2i\pi(x - \frac{n}{2f_B}) f_B}] = \\ &\sum_{n\in\mathbb{Z}} s(\frac{n}{2f_B}) \frac{sin(2\pi f_B (x - \frac{n}{2f_B}))}{2\pi f_B (x - \frac{n}{2f_B})} = \\ &\sum_{n\in\mathbb{Z}} s(\frac{n}{2f_B}) sinc(2\pi f_B (x - \frac{n}{2f_B})) \end{aligned}$$

□

Wniosek .1. *Próbkując sygnał o ograniczonym widmie z częstością $f_{Nyq} := 2f_B$ jesteśmy w stanie wyznaczyć jego wartość w dowolnym punkcie z dowolną dokładnością.*

Powyższe sformułowanie Twierdzenia Nyquista-Shanona nie uwożliwia wglądu w przypadki próbkowania z częstością różną niż f_{Nyq}. Zaprezentujemy teraz równoważne sformułowanie i alternatywny dowód oparty na formule sumacyjnej Poissona. Dostarczy to wystarczających narzędzi w dyskusji dowolnego próbkowania. Zacznijmy zatem od

Twierdzenie .2. *O formule sumacyjnej Poissona*

Niech $s \in \mathrm{L}_1(\mathbb{R})$ i $f_B > 0$. Wówczas sygnał $\sum_{n\in\mathbb{Z}} s(x + \frac{n}{f_B})$ zbiega prawie wszędzie do funkcji oznaczonej przez $\delta \in \mathrm{L}_1([0, \frac{1}{f_B}])$. Współczynniki rozwinięcia Fouriera są jak następuje: $f_B \hat{s}(nf_B)$.

Dowód. Dowód polega na przeprowadzeniu podstawowych przekształceń i jest pozostawiony czytelnikowi. Można go znaleźć również w [3]. □

Twierdzenie .3. *Niech s będzie ciągłym, stabilnym sygnałem i $\sum_{n\in\mathbb{Z}} |s(\frac{n}{2f_B})| < \infty$ dla $0 < f_B < \infty$. Wówczas*

$$\sum_{n\in\mathbb{Z}} \hat{s}(f + 2f_B n) = \sum_{n\in\mathbb{Z}} \frac{1}{2f_B} s(\frac{n}{2f_B}) e^{\frac{-2\pi i n f}{2f_B}}, \tag{.3}$$

Niech $T(f) \in \mathrm{L}_1(\mathbb{R})$. Oznaczmy

$$h(t) = \int_{\mathbb{R}} T(f) e^{2\pi i f t} df. \tag{.4}$$

Wówczas dla sygnału

$$\tilde{s}(t) = \sum_{n\in\mathbb{Z}} \frac{1}{2f_B} s(\frac{n}{2f_B}) h(t - \frac{n}{2f_B}) \tag{.5}$$

zachodzi

$$\tilde{s}(t) = \int_{\mathbb{R}} e^{2\pi i f t} T(f) \sum_{n\in\mathbb{Z}} \hat{s}(f + 2f_B n)\ df \tag{.6}$$

Dowód. Lewa strona (.3) spełnia założenia Twierdzenia .2 równość wszędzie zachodzi na mocy Lematu .1. Sygnał $h(t)$ jest odwrotną transformatą Fouriera funkcji $T(f)$. Z Lematu Riemanna-Lebesgue'a wynika, że $h(t)$ jest ciągła i znikająca w nieskończoności. Ponadto jest też ograniczona. Zatem prawa strona (.5) jest bezwzględnie zbieżna, więc i zbieżna do ciągłej funkcji $\tilde{s}(t)$. Równość (.6) jest połączeniem (.3) i (.5).

□

Uwaga 1. *W powyższym Twierdzeniu .3 funkcja $T(f)$ jest bezpośrednio związana z urządzeniem próbkującym i opisuje filtrowanie sygnału w reprezentacji częstości co chcemy stricte kontrolować. Urządzenie próbkujące oprócz pobierania punktów również filtruje sygnał. Funkcja $h(t)$ opisuje ten sam proces w reprezentacji czasu. Sygnał (.5) jest sygnałem spróbkowanym z częstością $2f_B$.*

Powyższa Uwaga 1 rzuca nowe światło na Twierdzenie Nyquista–Shannona. Zanotujmy, że $\mathcal{F}^{-1}(\frac{1}{2f_B}\mathbf{1}_{[-f_B,f_B]}(f))(t) = sinc(2f_B t)$, gdzie $\mathcal{F}$ oznacza Transformatę Fouriera. Wówczas z (.4) i (.5) wynika, że równość (3.1) odpowiada filtrowaniu opisanym w przestrzeni częstości przez funkcję $\frac{1}{f_B}\mathbf{1}_{[-f_B,f_B]}(f)$ i z próbkowaniem z częstością $2f_B$. W tym przypadku, gdy widmo jest ograniczone przez f_B filtr jest bezużyteczny, bo widmo i tak jest już ograniczone.

Aliasing

Sygnał o widmie ograniczonym przez $\tilde{f}_B$ przefiltrujmy filtrem $T(f)$ próbkując z częstością $2f_B < 2\tilde{f}_B$. Oznaczmy $2f_B = (2-\alpha)\tilde{f}_B$ i przez $\tilde{s}$ sygnał wynikowy. Przyjrzyjmy się (.6) i w szczególności wyrażeniu

$$\sum_{n\in\mathbb{Z}} \hat{s}(f + (2-\alpha)\tilde{f}_B n) T(f). \tag{.7}$$

Zakładamy dodatkowo, że $\tilde{s}$ jest całkowalny. Jeżeli spróbkowany sygnał ma wiernie reprezentować wyjściowy to $T(f) = \mathbf{1}_{[-\tilde{f}_B,\tilde{f}_B]}(f)$. Dla $f = \tilde{f}_B$ powyższa suma wynosi $\hat{s}(\tilde{f}_B) + \hat{s}(-(1-\alpha)\tilde{f}_B) \neq \hat{s}(\tilde{f}_B)$. Zatem widmo spróbkowanego sygnału nie pokrywa się z pierwotnym, a zjawisko to nazywamy *aliasingiem*. Próbkując sygnał możemy zaakceptować brak wysokich częstości w widmie

pod warunkiem, że w zakresie które obejmuje pokrywa się ono dokładnie z widmem wyjściowego sygnału. Dlatego też zakładamy, że $T(f) = \mathbf{1}_{[-f_\alpha, f_\alpha]}(f)$, gdzie wartości f_α są zależne od α. Wykorzystując powyższą sumę otrzymujemy, że $f < (1-\alpha)f_B$. Przejdźmy do dyskusji *nadpróbkowania*, dla którego *a priori* powinniśmy otrzymać wierną reprezentację, bo im dokładniejszy pomiar tym pewniejszy wynik.

Nadpróbkowanie

W przypadku gdy $f_B > 2\tilde{f}_B$ i niech $f_B = (2+\alpha)\tilde{f}_B$, $\alpha > 0$. W równości (.7), gdzie $-\alpha \to \alpha$, również zakładamy, że $T(f) = \mathbf{1}_{[-\tilde{f}_B, \tilde{f}_B]}(f)$. Prosty rachunek pozwoli nam stwierdzić, że

$$\sum_{n\in\mathbb{Z}} \hat{s}(f + (2+\alpha)\tilde{f}_B n)\mathbf{1}_{[-\tilde{f}_B, \tilde{f}_B]}(f)) = \hat{s}(f).$$

Widmo spróbkowanego sygnału pokrywa się z widmem pierwotnym i zachodzi (3.1).

Literatura

[1] Chitta Baral, Olac Fuentes, and Vladik Kreinovich. *Why Deep Neural Networks: A Possible Theoretical Explanation*, pages 1–5. Springer International Publishing, Cham, 2018.

[2] A. R. Barron. Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Trans. Inf. Theor.*, 39(3):930–945, May 1993.

[3] Pierre Bremaud. *Mathematical Principles of Signal Processing: Fourier and Wavelet Analysis*. Springer, 2010.

[4] Harvey Colten. *Sleep disorders and sleep deprivation : an unmet public health problem*. Institute of Medicine National Academies Press, Washington, D.C, 2006.

[5] Ian Goodfellow. *Deep learning : systemy uczące si*. Wydawnictwo Naukowe PWN SA, Warszawa, 2018.

[6] D.O. Hebb. *The Organization of Behavior: A Neuropsychological Theory*. Psychology Press, 2002.

[7] Geoffrey E. Hinton, Simon Osindero, and Yee-Whye Teh. A fast learning algorithm for deep belief nets. *Neural Comput.*, 18(7):1527–1554, July 2006.

[8] K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural Netw.*, 2(5):359–366, July 1989.

[9] Kurt Hornik. Approximation capabilities of multilayer feedforward networks. *Neural Netw.*, 4(2):251–257, March 1991.

[10] Kurt Hornik, Maxwell Stinchcombe, Halbert White, and Peter Auer. Degree of approximation results for feedforward networks approximating unknown mappings and their derivatives. *Neural Comput.*, 6(6):1262–1275, November 1994.

[11] Szabatin Jerzy. *Podstawy Teorii Sygnałów*. Wydawnictwa Komunikacji i Lacznosci WKL.

[12] A. Kales, A. Rechtschaffen, Los Angeles. Brain Information Service University of California, and NINDB Neurological Information Network (U.S.). *A Manual of Standardized Terminology, Techniques and Scoring System for Sleep Stages of Human Subjects*. NIH publication. U. S. National Institute of Neurological Diseases and Blindness, Neurological Information Network, 1968.

[13] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.

[14] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'12, pages 1097–1105, USA, 2012. Curran Associates Inc.

[15] Vernon J. Lawhern, Amelia J. Solon, Nicholas R. Waytowich, Stephen M. Gordon, Chou P. Hung, and Brent J. Lance. Eegnet: A compact convolutional network for eeg-based brain-computer interfaces, 2016.

[16] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Comput.*, 1(4):541–551, December 1989.

[17] Moshe Leshno and Shimon Schocken. Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. *Neural Networks*, 6:861–867, 1993.

[18] Urszula Malinowska, Hubert Klekowicz, Andrzej Wakarow, Szymon Niemcewicz, and Piotr Durka. Fully parametric sleep staging compatible with the classical criteria. 7:245–53, 11 2009.

[19] S. Mallat and Z. Zhang. Matching pursuits with time-frequency dictionaries. *IEEE Transactions on Signal Processing*, 41:3397–3415, 1993.

[20] Warren S. McCulloch and Walter Pitts. Neurocomputing: Foundations of research. chapter A Logical Calculus of the Ideas Immanent in Nervous Activity, pages 15–27. MIT Press, Cambridge, MA, USA, 1988.

[21] William H. Moorcroft. *Understanding Sleep and Dreaming (Springerlink Behavioral Science)*. Springer, 2005.

[22] Ernst Niedermeyer. *Electroencephalography : basic principles, clinical applications, and related fields*. Lippincott Williams & Wilkins, Philadelphia, 2005.

[23] Reuven Y. Rubinstein and Dirk P. Kroese. *The Cross Entropy Method: A Unified Approach To Combinatorial Optimization, Monte-carlo Simulation (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg, 2004.

[24] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Neurocomputing: Foundations of research. chapter Learning Representations by Back-propagating Errors, pages 696–699. MIT Press, Cambridge, MA, USA, 1988.

[25] David E. Rumelhart, James L. McClelland, and PDP Research Group. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition: Foundations (Volume 1)*. A Bradford Book, 1986.

[26] Ryszard Tadeusiewicz. *Sieci neuronowe*. Akademicka Oficyna Wydawnicza, Warszawa, 1993.

[27] Orestis Tsinalis, Paul M. Matthews, Yike Guo, and Stefanos Zafeiriou. Automatic sleep stage scoring with single-channel eeg using convolutional neural networks, 2016.

[28] Albert Vilamala, Kristoffer H. Madsen, and Lars K. Hansen. Deep convolutional neural networks for interpretable analysis of eeg sleep stage scoring, 2017.