Uniwersytet Wrocławski
Wydział Fizyki i Astronomii
Fizyka komputerowa

Praca Magisterska

Tytuł polski

Tytuł angielski

Autor:
Rafał Skrzypiec

Promotor:
dr hab. Krzysztof Graczyk

Wrocław, 2018

Streszczenie

Tekst streszczenia

Abstract

Tekst streszczenia

Spis treści

Rozdział 1

Wstęp

Proton to trwała cząsteczka subatomowa o dodatnim ładunku elektrycznym o wartości $+e$ i masie spoczynkowej około 938 MeV [53], nieznacznie mniejszej od masy neutronu. Według teorii modelu standardowego jest fermionem o spinie $1/2$, składa się z dwóch kwarków górnych i jednego kwarka dolnego związanych ze sobą dzięki oddziaływaniom silnym, których pośrednikiem są gluony. Jest to podstawowy składnik materii a jądro atomowe każdego pierwiastka zawiera jeden lub więcej protonów. Nie wszystkie właściwości protonu są dobrze poznane a ich zrozumienie ich stanowi istotny problem fizyki cząstek elementarnych.

Funkcje postaci protonu opisują przestrzenny rozkład ładunku elektrycznego oraz magnetycznego we wnętrzu protonu, są zatem dobrym dostarczycielem informacji o jego wewnętrznej strukturze i jednym z kluczowych składników, które mogą pomóc ją poznać i zrozumieć. Funkcje postaci protonu uzyskiwane są z analizy elastycznego rozpraszania elektronów na nukleonach oraz lekkich jądrach. Wyróżniamy elektryczną G_{E_p} i magnetyczną G_{M_p} funkcje postaci, które dla niewielkich wartości przekazu czteropędu mogą być utożsamiane z transformatami Fouriera gęstości ładunku elektrycznego i magnetyzacji wewnątrz protonu.

Znajomość funkcji postaci pozwala na oszacowanie promienia protonu czyli jednej z wciąż jeszcze niewystarczająco dobrze określonej właśności cząstki. Wielkość tę można zmierzyć poprzez opisane powyżej rozpraszanie elektron-proton lub spektroskopię atomową obserwując niewielkie przesunięcia w spektrum wodoru spowodowane fizycznym rozmiarem protonu. Analiza spektroskopii atomu wodoru składającego się z protonu i elektronu daje wyniki spójne z pomiarami rozpraszania, a szacunki promienia protonu obarczone są niepewnością rzędu 0.6% [17]. Komitet Danych dla Nauki i Techniki (CODATA) uwzględniając pomiary z obu metod rekomenduje wartość [53] $R_E = 0.8775(51)$ fm. Okazuje się, że do badania rozmiaru protonu można wykorzystać również miony, które są około 200-krotnie cięższe od elektronów. Eskperymenty obserwacji przesunięcia Lamba 2S-2P w spektrum wodoru mionowego zostały opublikowane w 2010 roku [56]. To podejście skutkuje wynikami obarczononymi 10 razy mniejszą niepewnością niż wcześniej [17]. Ostatnie publikacje [6] wskazują na wartość $R_E = 0.84087(39)$ fm. To wartość aż o 4% mniejsza od rekomendowanej przez CODATA. Takie rozbieżności wskazują na poważne braki w dotychczasowej teorii co równocześnie wzbudza zainteresowanie tą dziedziną fizyki.

Celem tej pracy magisterskiej jest zbudowanie modelu statystycznego, który wykorzystując pomiary eksperymentalne rozpraszania elektron-proton da w wyniku przewidywanie elektrycznej i magnetycznej funkcji postaci. Przedstawione analizy prezentują wykorzystanie trzech

modelów, które w różny sposób starają się rozwiązać to zadanie. Pierwszy z nich do analizy wykorzystuje tylko pomiary całkowitych przekrójów czynnych w zależności od kwadratu przekazu czteropędu Q^2 oraz czynnika kinematycznego ϵ i estymuje całkowity przekrój czynny. Następnie operacje różniczkowania pozwalaja na separacje funkcji postaci. Drugi z modelów wykorzystuje dodatkowo pomiary stosunków funkcji postaci, jego wynikiem są explicite elektrtyczna i magnetyczna funkcja postaci. Uzyskanie lepszej zgodności między modelem a pomiarami eksperymentalnymi wymaga uwzględnienia dodatkowej poprawki po za klasycznymi poprawkami radiacyjnymi. Trzeci model oprócz szacowania funkcji postaci przewiduje wartość poprawki dwufotonowej.

Zainspirowany poprzednimi badaniami [2], [34] wykorzystałem do tego celu popularny typ sztucznych sieci neuronowych - perceptron wielowarstwowy. Sieci neuronowe są przykładem systemów uczących się, i znalazły szerokie zastosowanie w problemach klasyfikacyjnych oraz regresyjnych. Swoją popularność zawdzięczają biologicznym analogiom, ich konstrukcja inspirowana jest budową znajdujących się w mózgu struktur stworzonych przez neurony - komórkę nerwową, która na podstawie sygnałów wejściowych wyznacza wartość wysyłanego przez nią sygnału wyjściowego [64]. Tak zdefiniowane bardzo ogólnie neurony możemy łączyć w rozbudowane struktury tworząc sieć neuronową. Połączone w dowolny sposób komórki tworzą graf, w którym każda krawędź ma przypisaną liczbę nazywaną wagą. Jej wartość świadczy o istotności połączenia i jest ustalana podczasa nauki modelu. Uczenie modelu wymaga przygotowania odpowiedniego zestawu danych, który składa się ze zmiennych objaśniających czyli danych wejściowych oraz zmiennej objaśnianej czyli oczekiwanej wartości wyjściowej. Następnie uczący algorytm propagacji wstecznej modyfikuje wagi łączące neurony tak aby zminimalizować wartość błędu pomiędzy oczekiwanymi wartościami wyjściowymi oraz wynikiem modelu.

Przedłożona praca została podzielona na sześć rozdziałów, następny zawiera wprowadzenie do tematyki elastycznego rozpraszania elektronów na protonie. Przedstawia rys historyczny eksperymentów, istotne mierzalne wielkości fizyczne oraz sposób ich pomiarów skupiając się na metodzie separacji Rosenblutha [65] oraz pomiarze polaryzacji protonu [1]. Rozdział 3 to wstęp do dziedziny sieci neuronowych, przedstawiona jest w nim ich historia, struktura i sposób działania. Następnie zaprezentowano wizualne działanie twierdzenia mówiącego, że sieć neuronowa może realizować aproksymację z dowolnie małym błędem oraz problem osiągnięcia kompromisu pomiędzy obciążeniem i wariancją modelu. Rozdział 4 to wprowadzenie do metodologii budowy i nauki modelu sieci neuronowych wraz z omówieniem wykorzystanych algorytmów m.in. generowania sztucznych danych, wczesnego zatrzymania nauki modelu oraz walidacji krzyżowej . W rozdziale 5 zostaną przedstawione wyniki trzech analiz wynikające z wykorzystania różnych technik modelowania oraz rodzajów pomiarów eksperymentalnych. Uzyskane wyniki zostaną porównane z rezultatami przedstawionymi w pracy [34]. Praca kończy się podsumowaniem w rozdziale 6.

Rozdział 2

Elastyczne rozpraszanie elektronów na protonie

2.1 Rys historyczny

W roku Ernest Rutherford wraz ze swoimi studentami Hansem Geigerem i Ernestem Marsdenem badali rozpraszanie cząstek naładowanych na cienkich foliach różnych pierwiastków. Do eksperymentu wykorzystano cząstki α emitowane przez naturalne pierwiastki radioaktywne.

Cząstka α to jądro podstawowego izotopu helu składające się z dwóch protonów i dwóch neutronów. Cząstka wysyłana jest przy rozpadzie niestabilnych jąder z prędkością $10^7 \frac{\mathrm{m}}{\mathrm{s}}$ i może przelecieć kilka centymetrów w powietrzu lub około 0,1 mm w ciele stałym zanim zostanie zatrzymana z powodu zderzeń. Masa cząstek α jest około 7300 razy większa od masy elektronu. Atomy folii można sobie wyobrazić jako upakowane dość blisko siebie kule. Ponieważ są neutralne, poza atomami właściwie nie ma oddziaływania z cząstkami α. Wewnątrz atomu cząstki mogą oddziaływać zarówno z dodatnim jak i z ujemnym ładunkiem. Ponieważ ujemny ładunek jest związany z elektronami, które są o cztery rzędy wielkości lżejsze od cząstek α możemy zaniedbać całkowicie oddziaływanie z elektronami. Oddziaływanie z ładunkiem dodatnim zależy w istotny sposób od rozkładu tego ładunku w atomie. Wyniki były zaskakujące jeśli przyjąć za słuszny model Thompsona z jednorodnym rozkładem masy i ładunku. Podczas eksperymentu zaobserwowano rozpraszanie pod dużymi kątami, dla niektórych cząstek pod kątem prawie 180°. Oczekiwania związane z eksperymentem były oparte o obliczenia w ramach klasycznej elektrodynamiki. Dawały się jednak wyjaśnić w ramach tej samej teorii klasycznej jeśli założyć, że cały ładunek dodatni i (prawie) cała masa atomy skupione są w bardzo małym obszarze rzędu 10^{-14}m w środku atomu. Eksperyment potwierdzał więc hipotezę, że atom posiada jądro, bardzo gęstą, bardzo małą strukturę niosącą cały dodatni ładunek atomu i prawie całą (99,95%) jego masę.

2.2 Różniczkowy przekrój czynny

Wyniki eksperymentów rozproszeniowych charakteryzujemy przy pomocy różniczkowego przekroju czynnego. Żeby wprowadzić to pojęcie ustalmy układ współrzędnych w ten sposób, że centrum potencjału rozpraszającego znajduje się w początku układu, a oś OZ skierowana jest wzdłuż padającej wiązki cząstek. Różniczkowy przekrój czynny $\frac{d\sigma}{d\Omega}(\theta, \varphi)$ definiujemy przy

pomocy ilorazu:

$$\frac{d\sigma}{d\Omega}(\theta,\varphi) = \frac{\begin{array}{c}\text{liczba cząstek rozproszonych w jednostce}\\ \text{czasu w kąt bryłowy } d\Omega \text{ wokół kierunku } (\theta,\varphi)\end{array}}{\text{strumień cząstek wiązki padającej } \times\ d\Omega}. \tag{2.1}$$

Strumień cząstek wiązki padającej to liczba cząstek przechodzących w jednostce czasu przez jednostkę powierzchni prostopadłą do osi OZ, w obszarze $z \ll 0$, a więc tam gdzie cząstki wiązki nie oddziałują jeszcze z centrum rozpraszającym. Wielkość ta ma wymiar $\left[\frac{1}{\mathrm{m}^2\,\mathrm{s}}\right]$, zatem różniczkowy przekrój czynny ma wymiar pola powierzchni. Często stosowaną jednostką, w której wyraża się przekroje czynne jest 1 barn $= 10^{-28}$ m^2. Definicję różniczkowego przekroju czynnego można interpretować następująco: liczba cząstek rozproszonych w jednostce czasu w kąt bryłowy $d\Omega$ jest równa liczbie cząstek pochłoniętych przez tarczę o powierzchni $\frac{d\sigma}{d\Omega}(\theta,\varphi)\,d\Omega$ umieszczoną prostopadle do kierunku propagacji wiązki padającej. Całkując po kątach otrzymujemy całkowity przekrój czynny:

$$\sigma = \int d\Omega \frac{d\sigma}{d\Omega}(\theta,\varphi) = \int_0^{\pi} \sin\theta d\theta \int_0^{2\pi} d\varphi \frac{d\sigma}{d\Omega}(\theta,\varphi)\,. \tag{2.2}$$

W definicji przekroju czynnego występuje tylko liczba cząstek rozproszonych. Dla kątów $\theta > 0$ możemy przyjąć, że detektor mierzący liczbę cząstek rozproszonych jest na tyle daleko od centrum rozpraszającego, że jest już poza wiązką padającą, która ma wymiar poprzeczny większy od rozmiarów obszaru z potencjałem istotnie różnym od zera, ale jednak skończony. Im mniejszy kąt θ tym trudniej oddzielić cząstki rozproszone od nierozproszonych. Dla małych kątów można mówić o różniczkowym przekroju czynnym tylko w sensie ekstrapolacji z obszaru $\theta > 0$. Pojęcie różniczkowego i całkowitego przekroju czynnego nie jest ograniczone do rozpraszania elastycznego, stosuje się je także do opisu rozpraszania nieelastycznego.

Zakładając rozpraszanie cząstki α w polu siły centralej $\sim \frac{1}{r^2}$, wykorzystując zasadę zachowania momentu pędu oraz energii mechanicznej otrzymujemy wzór na różniczkowy przekrój czynny, którym posługiwał się Rutherford [4]

$$\frac{d\sigma}{d\Omega}(\theta) = \left(\frac{e^2}{8\pi\epsilon_0 E_\alpha}\right)^2 \frac{1}{4\sin^4\frac{\theta}{2}}, \tag{2.3}$$

gdzie e to ładunek elementarny, ϵ_0 to przenikalność elektryczna próżni, E_α to energia padającej cząstki α natomiast θ to kąt rozpraszania.

Z racji skali cząstek, które biorą udział w tym doświadczeniu, zagadnienie należy jednak rozpatrywać z punktu widzenia mechaniki kwantowej. Szukając stanów stacjonarnych hamiltonianu

$$\hat{H} = \frac{\hat{p}^2}{2\mu} - \frac{e^2}{4\pi\varepsilon_0}\frac{1}{r}, \tag{2.4}$$

przy wykorzystujaniu przybliżenia Borna, czyli jednej z metod przybliżonego rozwiązywania równania na stacjonarne stany rozproszeniowe, otrzymujemy wzór Rutherforda [41] na różniczkowy przekrój czynny na rozpraszanie na potencjale kulombowskim

$$-\frac{\alpha}{r} = -\frac{e^2}{4\pi\epsilon_0}\frac{1}{r}, \tag{2.5}$$

$$\frac{d\sigma}{d\Omega}(\theta) = \left(\frac{\alpha}{2E_\alpha}\right)^2 \frac{1}{4\sin^4\frac{\theta}{2}} = \left(\frac{e^2}{8\pi\epsilon_0 E_\alpha}\right)^2 \frac{1}{4\sin^4\frac{\theta}{2}}. \tag{2.6}$$

W granicy $\theta \to 0$ różniczkowy przekrój czynny staje się nieskończony. Całka po kątach jest rozbieżna i całkowity przekrój czynny jest nieskończony. Źródłem tych nieskończoności jest długozasięgowy charakter potencjału kulombowskiego, w sytuacjach fizycznych nie spotykamy czystego potencjału kulombowskiego, jest on zwykle mniej lub bardziej skutecznie ekranowany. To, że wynik otrzymany w ramach mechaniki kwantowej jest taki sam jak w mechanice klasycznej miało zasadniczy wpływ na rozwój mechaniki kwantowej. To dzięki temu zbiegowi okoliczności wnioski Rutherforda dotyczące budowy atomu wyciągnięte na podstawie wzorów klasycznych okazały się słuszne, a pytanie o stabilność układu składającego się z dodatnio naładowanego jądra i krążących wokół niego elektronów stało się podstawowym problemem teoretycznym dla ówczesnych fizyków.

Okazuje się, że badanie rozpraszania cząstek mikroskopowych to jeden z najlepszych eksperymentalnych sposobów poznania ich struktury oraz oddziaływań pomiędzy nimi. W trakcie zderzeń może ulec zmianie struktura wewnętrzna cząstek, a nawet ich rodzaj i liczba. Dzieje się tak gdy energia zderzenia jest dostatecznie wysoka. Takie zdarzenia nazywamy nieelastycznymi, w elastycznych procesach rozpraszania liczba cząstek się nie zmienia i wewnętrzne stopnie swobody nie ulegają wzbudzeniu, można więc zaniedbać strukturę wewnętrzną cząstek.

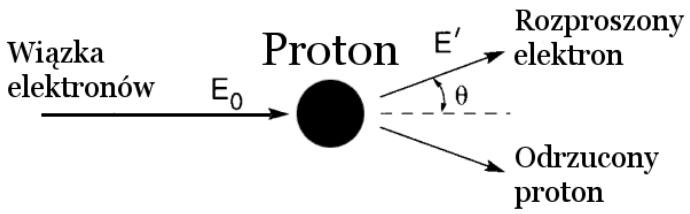
Wraz z budową akceleratorów cząstek oraz rozwojem aparatury eksperymentalnej dokonał się ogromny postęp w poznaniu budowy podstawowych składników materii. Możliwość produkcji wiązki elektronów o wysokiej enegii i dużym natężeniu spowodowały, że od lat 60 XX wieku, rozpraszanie elektronów i protonów stało się skuteczną metodą badania struktury protonu. Rozpraszanie elektronu na protonie polega na wymianie wirtualnego fotonu między naładowanymi elektrycznie elektronem i protonem. Wynik rozpraszania $e^- p \to e^- p$ jest ściśle zależny od długości fali $\lambda = hc/E$ wymienionego podczas oddziaływania fotonu, większa energia pozwala spojrzeć bardziej wgłąb struktury cząstki.

- Dla bardzo małej energii elektronu, $\lambda \gg r_p$, gdzie r_p jest promieniem protonu, rozpraszanie jest równoważne rozpraszaniu na punktowej cząstce.
- Dla małej energii elektornu, $\lambda \sim r_p$, rozpraszanie jest równoważne rozpraszaniu na naładowanej cząstce niepunktowej.
- Dla dużej energii elektronu, $\lambda < r_p$, długość fali jest wystarczająco mała aby dostrzec strukturę protonu, źródłem rozpraszania są kwarki.
- Dla bardzo dużych energii elektornu, $\lambda \ll r_p$, źródłem rozpraszania jest morze kwarkowo-gluonowe

2.3 Metoda Rosenblutha

Energia rozproszonego elektronu E' jest mniejsza niż energia początkowa E_0 o wartość energii przekazanej protonowi o masie M i wynosi:

$$E' = \frac{E_0}{1 + \frac{2E_0}{M}\sin^2\frac{\theta}{2}} \tag{2.7}$$



Rysunek 2.1: Prosty schemat rozpraszania elastycznego elektronu o energii początkowej E_0 i energii kończowej E' na jądrze atomu wodoru, θ to kąt rozpraszania.

Zdefiniujmy $Q^2 \equiv -q^2$, q to przekaz czteropędu.

$$\begin{aligned} Q^2 \equiv -q^2 = -(p^\mu - p'^\mu)^2 &= 2M\,(E_0 - E') \\ &= 4E_0E' \sin^2\frac{\theta}{2} \end{aligned} \tag{2.8}$$

Otrzymana formuła Rutherforda z równania 2.6 dobrze sprawdza się przy opisie rozpraszania nierelatywistycznych elektronów. Przekrój czynny szybkich elektronów $p_e = E_e$ wymaga uwzględnienia dodatkowych czynników. Są nimi: efekty relatywistyczne, odrzut protonu oraz oddziaływanie spin-spin. Należy pamiętać, że poniższe równania odpowiadają układowi jednostek miar HEP (*high energy physics*) powszechnie stosowanemu w fizyce cząstek elementarnych. Zmodyfikowany przekrój czynny opisywany jest przez formułę Motta [4],

$$\begin{aligned} \left(\frac{d\sigma}{d\Omega}\right)_M &= \left(\frac{d\sigma}{d\Omega}\right)_{Rutherford} \times 4\frac{E'}{E}\left(\cos^2\frac{\theta}{2} + \frac{Q^2}{2M^2}\sin^2\frac{\theta}{2}\right) \\ &= \frac{\alpha^2}{4E^2\sin^4\frac{\theta}{2}}\frac{E'}{E}\left(\cos^2\frac{\theta}{2} + \frac{Q^2}{2M^2}\sin^2\frac{\theta}{2}\right). \end{aligned} \tag{2.9}$$

Powyższe równanie bierze jednak pod uwagę oddziaływania elektronu z protonem traktując proton jako obiekt punktowy, przy dostacznie wysokiej energii eletronu proton ujawnia swoje fizyczne wymiary oraz strukturę wewnętrzną. Ich skutkiem jest rozkład ładunku elektrycznego i momentu magnetycznego protonu.

Struktura protonu opisywana jest więc przez dwie funkcje postaci, elektryczną G_{E_p} i magnetyczną G_{M_p} które są transformatami Fouriera, odpowiednio rozkładu ładunku elektrycznego i rozkładu momentu magnetycznego protonu. Funkcje postaci protonu opisują przestrzenny rozkład ładunku elektrycznego we wnętrzu protonu, są zatem dobrym dostarczycielem informacji o jego wewnętrznej strukturze i jednym z kluczowych składników, które mogą pomóc ją poznać i zrozumieć. W granicy nierelatywistycznej definiowane są w następujący sposób [17]:

$$G(Q^2) = \int d^3\vec{r}\,\rho(\vec{r})e^{i\vec{Q}\cdot\vec{r}}, \tag{2.10}$$

całkując po całej objętości protonu. W praktyce podczas rozpraszania elektron z protonem wymieniają między sobą nieskończoną ilość fotonów, w pierwszym przybliżeniu można założyć, że pośrednikiem oddziaływania jest tylko jeden foton. Zmodyfikowane równanie 2.9 nosi nazwę przekroju czynnego Rosenblutha [73]:

$$\begin{aligned}\left(\frac{d\sigma}{d\Omega}\right)_R &= \frac{\alpha^2}{4E^2\sin^4\frac{\theta}{2}}\frac{E'}{E}\left(\frac{G_{E_p}^2+\tau G_{M_p}^2}{1+\tau}\cos^2\frac{\theta}{2}+2\tau G_{M_p}^2\sin^2\frac{\theta}{2}\right)\\ &= \frac{\alpha^2}{4E^2\sin^4\frac{\theta}{2}}\frac{E'}{E}\cos^2\frac{\theta}{2}\times\left[G_{Ep}^2+\frac{\tau}{\epsilon}G_{M_p}^2\right]\frac{1}{(1+\tau)}\\ &= \left(\frac{d\sigma}{d\Omega}\right)_0\times\left[G_{E_p}^2+\frac{\tau}{\epsilon}G_{M_p}^2\right]\frac{1}{(1+\tau)},\end{aligned} \tag{2.11}$$

gdzie ϵ jest czynnikiem kinematycznym i także polaryzacją wirtualnego fotonu

$$\epsilon = \left[1+2(1+\tau)\,\mathrm{tg}^2\left(\frac{\theta}{2}\right)\right]^{-1},$$

ponadto

$$\tau = \frac{Q^2}{4M^2}.$$

Funkcje G_{E_p}, G_{M_p} są zależne tylko od przekazu czteropędu Q^2 i spełniają poniższe warunki brzegowe:

$$\begin{aligned}G_{E_p} &= G_{E_p}(Q^2) \quad G_{E_p}(0)=1\\ G_{M_p} &= G_{M_p}(Q^2) \quad G_{M_p}(0)=\mu_p,\end{aligned}$$

gdzie μ_p to moment magnetyczny protonu. Wartości tych istotnych funkcji są wyznaczane eksperymentalnie.

Zmierzony w laboratorium całkowity przekrój czynny rozpraszania w przybliżeniu jednofotonowym wyraża się formułą [2]:

$$\begin{aligned}\sigma_R(\epsilon,Q^2) &\equiv \epsilon\,(1+\tau)\,\frac{\sigma\,(\epsilon,Q^2)}{\sigma_0\,(\epsilon,Q^2)}\\ &= \tau G_{M_p}^2(Q^2)+\epsilon G_{E_p}^2(Q^2).\end{aligned} \tag{2.12}$$

Metoda Rosenblutha to pierwsza poznana technika pozwalająca na otrzymanie wartości funkcji G_E i G_M dla protonu. Wymaga ona pomiarów przekroju czynnego rozpraszania elektron-proton podczas wielu eksperymentów, dla ustalonej wartości Q^2 i różncyh wartości ϵ. Zmianę tych parametrów można otrzymać poprzez korygowanie energii wiązki oraz kąta rozpraszania elektronu w tak dużym zakresie, jak to jest wykonalne eksperymentalnie. Metoda separacji Rosenblutha pozwala zapisać zredukowany przekrój czynny jako kombinację liniową funkcji postaci G_{E_p} oraz G_{M_p} [55], funkcje postaci stają się wtedy odpowiednimi współczynnikami prostej z równania (2.12), które można wyznaczyć metodą regresji.

2.4 Metoda transferu polaryzacji

Kolejna metody oparte są na pomiarach polaryzacji odbitego protonu podczas rozpraszania spolaryzowanych elektronów lub na pomiarach asymetrii rozpraszania, pozwalają one określić

wzajemny stosunek elektrycznego oraz magnetycznego czynnika postaci

$$\mathcal{R}\left(Q^2\right) \equiv \mu_p \frac{G_{E_p}\left(Q^2\right)}{G_{M_p}\left(Q^2\right)},$$

gdzie $\mu_p = 2,793$ to moment magnetyczny protonu. W przybliżeniu jednofotonowym, otrzymujemy tylko dwa niezerowe składniki wektora polaryzacji, poprzeczny P_t oraz podłużny P_l, przedstawione na rysunku 2.2. Stosunek czynników postaci możemy otrzymać bezpośrednio ze stosunku składowych polaryzacji, otrzymujemy [55]:

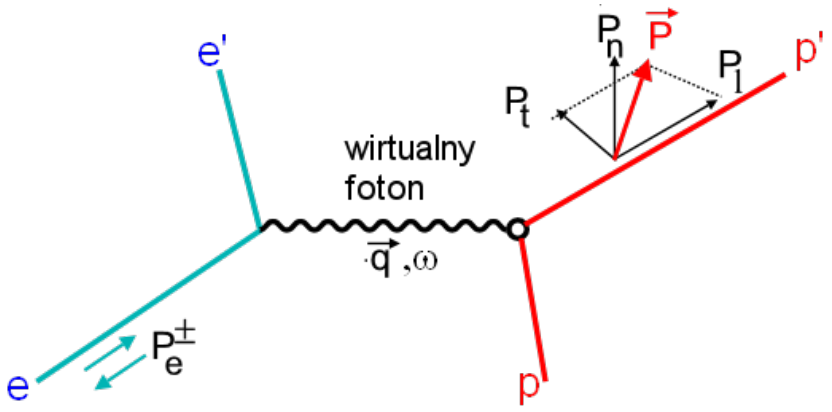
$$\mathcal{R}\left(Q^2\right) \equiv -\mu_p \frac{P_t}{P_l} \frac{E+E'}{2M} \operatorname{tg}^2\left(\frac{\theta}{2}\right)$$

gdzie P_l i P_t to podłużny i poprzeczny składnik wektora polaryzacji odrzuconego protonu. E oraz E' to poczatkowa i końcowa energia elektronu, θ to kąt rozpraszanai elektronu i M to masa protonu. Schemat procesu pokazany został na rysunku 2.2

Współczynnik $\mathcal{R}\left(Q^2\right)$ może także zostać wyznaczony na podstawie pomiaru asymetrii podczas sprężystego rozpraszania elektron-proton [2, 55]

$$\frac{\sigma_+ - \sigma_-}{\sigma_+ + \sigma_-} = -2\mu_p \sqrt{\tau\left(1+\tau\right)} \operatorname{tg}\left(\frac{\theta}{2}\right) \frac{\mathcal{R} \sin\theta^* \cos\phi^* + \mu_p \sqrt{\tau\left[1+\left(1+\tau\right)\operatorname{tg}^2\left(\frac{\theta}{2}\right)\right]} \cos\theta^*}{\mathcal{R}^2 + \mu_p \tau/\epsilon},$$

gdzie σ_+ i σ_- to przekroje czynne dla dodatniej i ujemnej skrętności, θ^* i ϕ^* to kąty polarny i azymutalny polaryzacji protonu względem wektora przekazu pędu $\vec{q}$ i płaszczyzny rozpraszania.



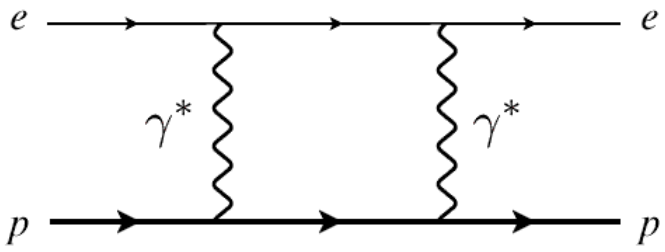
Rysunek 2.2: Schemat rozpraszania spolaryzowanego podłużnie elektronu na protonie w przybliżeniu jednofotonowym.

2.5 Poprawka dwu-fotonowa

Jak zaznaczono w [7], najnowsze pomiary przekrojów czynnych wskazują, że do uzyskania całkowitej zgodność z danymi pomiarów transferu polaryzacji niezbędne jest uwzględnienie

dodatkowych poprawek do równania 2.12. Po za klasycznymi poprawkami radiacyjnymi należy rozważyć poprawkę dwufotonową, która może zostać zapisana jako dodatkowy składnik równania.

$$\sigma_R \rightarrow \sigma_R + \delta_{TPE} \tag{2.13}$$



Rysunek 2.3: Diagram dwufotonowy dla elastycznego rozpraszania elektron-proton.

Diagram Feynmana dla tego zjawisku został przedstawiony na rysunku 2.3. Obliczenie poprawki jest trudne i zależne od modelu, wiedząc jednak, że amplituda rozpraszania musi spełniać ogólne zasady symetrii możemy założyć, że

$$\sigma_R \rightarrow \sigma_R + \delta_{TPE}\left(Q^2, \epsilon\right) \tag{2.14}$$

Wpływ poprawki dwufotonowej zostanie uwzględniony w jednej z przedstawionej poniżej analiz.

2.6 Promień protonu

Zgodnie z równianiem (2.10), funkcje postaci w granicy nierelatywistycznej definiowane są w następujący sposób:

$$G(Q^2) = \int d^3\vec{r}\, \rho(\vec{r}) e^{i\vec{Q}\cdot\vec{r}},$$

Ponadto rozwinięcie w szereg dla niewielkich Q^2 daje:

$$G(Q^2) = 1 - \frac{1}{6}\langle r^2\rangle Q^2 + \ldots, \tag{2.15}$$

gdzie $\langle r^2\rangle$ to kwadrat średniej kwadratowej promienia protonu. Dokładne oszacowanie przebiegów funkcji postaci G_{E_p} i G_{M_p} jest niezwykle ważne, poniważ ich znajomość może posłużyć do oszacowania promienia protonu, który wyraża się poniższym wzorem:

$$r^2_{E,M} \equiv \langle r^2_{E,M}\rangle = -6\frac{dG_{E,M}}{dQ^2}\bigg|_{Q^2=0} \tag{2.16}$$

Wielkość tę można zmierzyć poprzez opisane powyżej rozpraszanie elektron-proton lub spektroskopię atomową obserwując niewielkie przesunięcia w spektrum wodoru spowodowane fizycznym rozmiarem protonu. Rezultaty otrzymane w pomiarach spektroskopowych atomu wodoru składającego się z protonu i elektronu są spójne z pomiarami rozpraszania i dają

wynik z niepewnością rzędu 0.6% [17]. Jedne z najdokładniejszych pomiarów rozpraszania zostały opublikowane w [12], natomiast wartość promienia protonu została oszacowana na

$$R_E = 0.879(8) \text{ fm}. \tag{2.17}$$

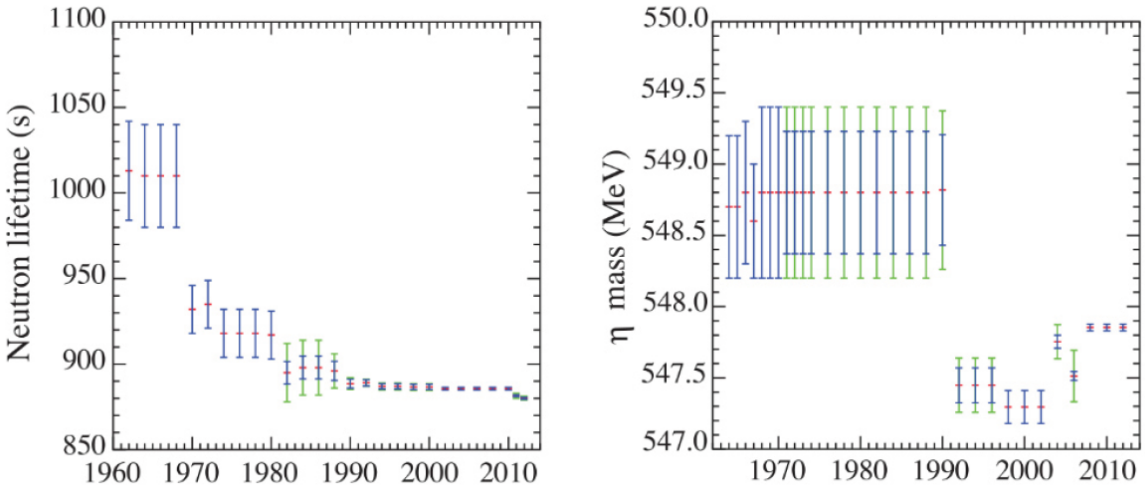
Komitet Danych dla Nauki i Techniki (CODATA) uwzględniając dodatkowo pomiary spektroskopowe rekomenduje wartość [53]

$$R_E = 0.8775(51) \text{ fm}. \tag{2.18}$$

Okazuje się, że do badania rozmiaru protonu można wykorzystać również miony. Dysponujemy wtedy jedną metodą, która daje rezultaty obarczone bardzo niewielką niepewnością. Pierwsze wyniki zostały opublikowane w 2010 roku [56] i zawierały promień protonu zmierzony przez obserwacje przesunięcia Lamba 2S-2P w spektrum wodoru mionowego. Mion, z powodu masy około 200-krotnie większej od elektronu orbituje znacznie bliżej protonu niż elektron i rozmiar protonu ma znacznie większy wpływ na jego poziomy energetyczne. Pozwala to na pomiar promienia protonu z niepewnością 10 razy mniejszą niż przy wynikach otrzymanych z eksperymentów z wykorzystaniem elektronów [17]. Obecne rezultaty wskazują na wartość [6]

$$R_E = 0.84087(39) \text{ fm}. \tag{2.19}$$

To wartość o 4% mniejsza od rekomendowanej przez CODATA (2.18), będąca rozbieżnością na poziomie 7σ. Wartości stałych fizycznych bardzo często znacząco zmieniały swoją wartość wraz z zmianą techniki pomiaru i wzrostem czułości urządzeń pomiarowych. Rysunek 2.4 przedstawia ewolucje wartości średniego czasu życia neutronu oraz masy mezonu eta w czasie. Możemy zauważyć, że w obu tych przykładach najnowsza wartość nie znajduje się nawet w obszarze niepewności pomiarowej początkowych wskazań.



Rysunek 2.4: Przykłady zmian szacowanych wielkości fizycznych w czasie. Na wykresie po lewej stronie średni czas życia swobodnego neutronu, po prawej stronie masa mezonu eta. Rysunek zapożyczony z pracy [11].

2.7 Cel pracy i pomiary rozproszeniowe

Podczas analizy wykorzystano dane otrzymane na skutek pomiarów dwóch opisanych powyżej wielkości fizycznych. Pierwszy zestaw danych to 24 niezależne zbiory danych, opublikowane w pracach [40, 9, 3, 49, 31, 10, 59, 14, 8, 44, 68, 74, 5, 15, 72, 54, 69, 70, 62, 24, 18], które zawierają pomiary przekrojów czynnych σ_R wraz z niepewnością pomiarową $\Delta\sigma_R$ w zależności od czynnika kinematycznego ϵ oraz kwadratu przekazanego czteropędu Q^2. Ponadto w jednej z analiz zostanie wykorzystany powyższy zbiór danych z wartościami przekrojów czynnych σ_R zmodyfikowanych o poprawkę wynikającą z wymiany dwóch fotonów podczas rozpraszania. Rozmiar danych to razem 426 punktów pomiarowcyh.

Drugi zestaw tworzą zbiory opublikowane w pracach [52, 43, 23, 57, 28, 29, 60, 38, 51, 42, 63, 19], które zawierają stosunek elektrycznej i magnetycznej funkcji postaci $\mathcal{R}$ wydobyty z tzw. pomiarów transferu polaryzacji w zależności od Q^2. Kolejna kolumna zawiera informacje nt. niepewności pomiarowej $\Delta\mathcal{R}$. Drugi zbiór to 68 punktów pomiarowych.

Dodatkowo każdy z 24 zbiorów danych zawierających pomiary całkowitego przekroju czynnego ma określoną niepewność systematyczną $\Delta\eta$ wynikającą z rozbieżności pomiędzy pomiarami całkowitych przekrojów czynnych i pomiarami podłużnej i poprzecznej składowej polaryzacji protonów dającymi w rezultacie stosunki funkcji postaci protonu.

Celem tej pracy jest zbudowanie modelu statystycznego, który wykorzystując opisane powyżej pomiary eksperymentalne da w wyniku przewidywanie elektrycznej i magnetycznej funkcji postaci. Zainspirowany poprzednimi badaniami [2], [34] wykorzystałem do tego celu popularny typ sztucznych sieci neuronowych - perceptron wielowarstwowy.

Rozdział 3

Sieci Neuronowe

3.1 Historia i rozwój sieci neuronowych

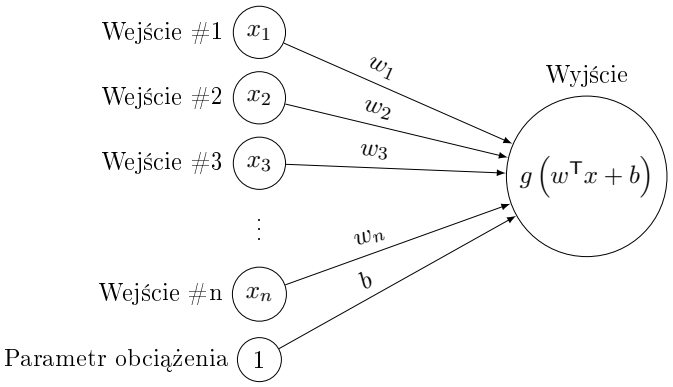
Z biologicznego punktu widzenia neuron to komórka, która odbiera, przetwarza i przesyła informacje wykorzystując elektryczne i chemiczne sygnały. Z neuronu wychodzą wypustki, z których jedna, przesyłająca sygnał to akson, pozostałe, odbierające sygnały to dendryty. Jeden neuron może więc otrzymywać informacje od wielu komórek nerwowych, które połączone są ze sobą przez synapsy. Jeśli suma sygnałów elektrycznych z wejściowych połączeń przekroczy pewien próg, zależne od napięcia kanały sodowe otwierają się pozwalając neuronowi na dalszą transmisję sygnału elektrycznego [50]. Inspirując się opisanym powyżej, uproszczonym schematem działania neuronów, Frank Rosenblatt w 1957 roku zaproponował pojęcie perceptronu. Dziś rozumiane jest jako najprostszy matematyczny model neuronu służący nauce binarnej klasyfikacji czyli funkcji, która jako wejście przyjmuje wektor liczb rzeczywistych x a jej wyjściem jest 0 lub 1

$$g(x) = \begin{cases} 1 & \text{gdy} \quad w^\mathsf{T} x + b > 0 \\ 0 & \text{gdy} \quad w^\mathsf{T} x + b \leqslant 0. \end{cases}$$

Gdzie w^T to wektor wartości wag symbolizujących synapsy odpowiadające parametrom wejściowym x, $w^\mathsf{T} x$ to iloczyn skalarny $\sum_i w_i^\mathsf{T} x_i$. Analogicznie do uproszczonego modelu biologicznego neuronu, wynik wyjściowy jest pozytywny po przekroczeniu przez sumę sygnałów wejściowych pewnego progu b - parametru obciążenia.

Tak zdefiniowany perceptron ma ograniczone możliwości klasyfikacji, jednak użycie bardziej skomplikowanych, szczególnie nieliniowych funkcji aktywacji opisanych w rozdziale 3.2 pozwala imitować perceptronowi każdą bramkę logiczną. Ponadto okazuje się, że połączenie wielu perceptronów w jedną warstwę daje nieskończone możliwości klasyfikacyjne i regresyjne, co opisano w sekcji 3.4. Następny przełom w dziedzinie perceptronów nastąpił w latach 80 wraz ze zdefiniowaniem efektywnej metody nauki perceptronów wielowarstwowych (MLP - Multilayer Perceptron) – metody propagacji wstecznej przedstawionej w rozdziale 3.3.

Pomysł wykorzystania perceptronu jako podstawowej komórki budującej zaowocował stworzeniem wielu różnych struktur sieci neuronowych różniących się sposobem połączenia neuronów oraz metodą przetwarzania danych wejściowych przez neuron. Po za tym, dzięki wykorzystaniu procesorów graficznych w obliczeniach, w ostatnich latach szczególne zainteresowanie



Rysunek 3.1: Schemat perceptronu o n wejściach, wartość wyjściowa jest wynikiem działania funkcji Heaviside'a g na argument $w^\mathsf{T}x + b$.

wzbudziły konwolucyjne sieci neuronowe (CNN - Convolution Neural Network). Ich głównym wykorzystaniem jest przetwarzanie obrazów lub dźwięków. Czarno-biały obraz o wymiarach 100×100 pikseli to 10 tysięcy potencjalnych wag każdego neuronu w wartstwie ukrytej. Inspirując się działaniem ludzkiego narządu wzroku zaproponowano [46] wykorzystanie w sieci neuronowej warstw konwolucyjnych oraz łączących, które skutecznie zmniejszają rozmiar wektora wejściowego przetworzonego obrazu wraz z zachowaniem jego kluczowych cech. Tak przetworzony wektor jest równocześnie wejściem do dołączonej w następnych wartstwach sieci MLP.

Kolejną klasę sztucznych sieci neuronowych tworzą rekurencyjne sieci neuronowe (RNN - Recurrent Neural Network), gdzie komórki zawierają pętlę łączącą wyjście danego neuronu z jego wejściem [25]. Taki neuron przetwarza więc informacje nie tylko z poprzednich warstw ale także tę, którą zawierał w przeszłości. RNN są skutecznie wykorzystywane szczególnie w przypadku przetwarzania danych sekwencyjnych takich jak rozpoznawanie mowy. Przełomem, który znacznie ułatwił naukę modelów RNN oraz zwiększył ich skuteczność jest koncept komórki LSTM [35], która ma bramki wejścia, wyjścia oraz zawiera bramkę zapomnij. Bramka wejściowa decyduje jak dużo informacji z poprzedniej warstwy zostaje zapamiętanych przez neuron, bramka wyjściowa decyduje jak dużo informacji z neuronu zostaje przekazanych do następnej warstwy. Bramka zapomnij determinuje ilość informacji z przeszłości przechowywanych przez neuron.

Stosunkowo nowym pomysłem, który wywodzi się z 2014 roku są generatywne sieci antagonistyczne (GAN - Generative Adversarial Network) [33]. Składają się one z dwóch sieci, które są równocześnie uczone. Pierwsza nazywana dyskryminatorem przymjmuje jako wejście np. obraz Y i generuje wyjście w skalarnej postaci $D(Y)$, które wskazuje czy obraz Y wygląda naturalnie czy nie. O $D(Y)$ możemy myśleć jak o funkcji, która ma niską wartość bliską zeru gdy Y jest prawdziwym obrazem i dużą wartość dodatnią jeśli mamy do czynienia z nienaturalnym np. zaszumionym obrazem. Druga sieć nazywana jest generatorem, oznaczmy ją przez $G(Z)$, gdzie Z to wektor losowy, próbkowany z dowolnego rozkładu. Rolą generatora jest wytworzenie obrazów, które będą wejściem dyskryminatora. W czasie treningu D uczy się na prawdziwych obrazach i dopasowuje swoje parametry tak aby minimalizować funkcję $D(Y)$, następnie wejściem D jest obraz wyprodukowany z wektora Z, wtedy sieć jest uczona aby maksymalizować wartość $D(G(Z))$. Podczas tego procesu generator może nauczyć się produkować takie obrazy $G(Z)$ aby minimalizować wartość wyjścia dyskryminatora a co za tym idzie oszkuać go nt. prawdziwości tych obrazów. Główne zastosowanie GAN to generowanie fotorealistycznych obrazów oraz filmów wideo.

Wśród innych ciekawych i najpopularniejszych konceptów wykorzystujących sieci neuronowe wywodzące się z idei perceptronu znajdują się autoenkodery (AE - Autoencoder) [16], których zadaniem jest kompresja wejściowej informacji. Sieci Hopfielda [36] wykorzystywane do modelowania pamięci skojarzeniowej, sieci Kohonena nazywane inaczej samoorganizującym się odwzorowaniem (SOM - Self-Organizing Map) [45], których wynikiem jest reprezentacja danych wejściowych o zmniejszonej liczbie wymiarów oraz wiele innych sieci zawierających modyfikacje przedstawionych powyżej pomysłów.

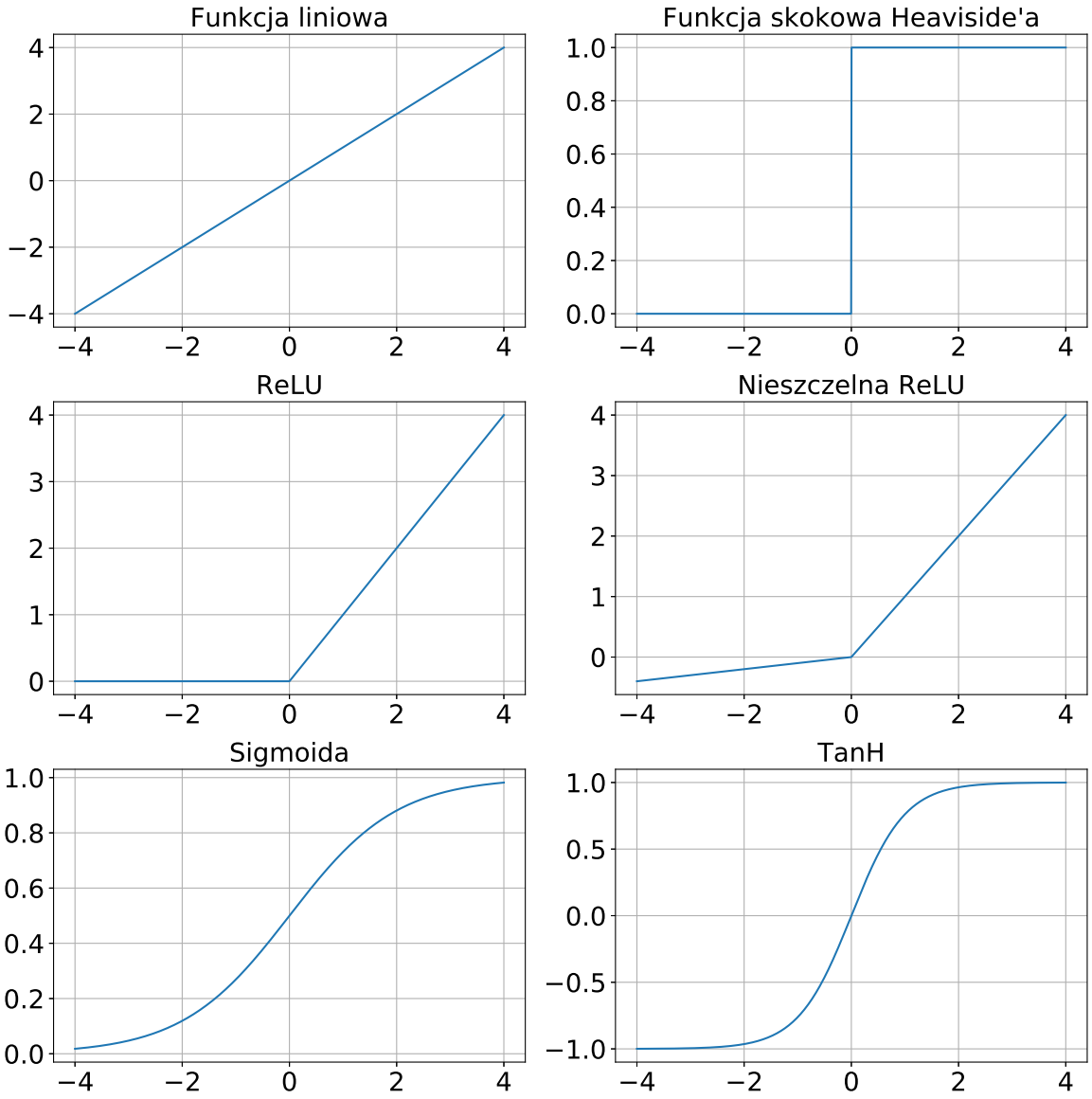
3.2 Funkcje aktywacji

Jednym z podstawowych problemów podczas wyboru struktury sieci neuronowej jest wybór typu ukrytej jednostki. Funckja aktywacji to funkcja, która działa na każdy neuron ukryty w sieci neuronowej, jako argument przyjmuje sumę iloczynów wartości neuronów z warstwy poprzedzającej i odpowiadających im wag. Każda z warstw sieci neuronowej lub nawet każdy neuron może mieć zdefiniowaną inną funckję aktywacji.

Perceptron, który był inspiracją powstania sieci neuronowcyh został skonstruowany jako uproszczony model biologicznego neuronu, który do pewnej progowej wartości napięcia nie jest aktywny. Wykorzystanie funkcji Heaviside'a jako funkcji aktywacji przedstawionej w prawym górnym rogu na rysunku 3.2 pozwoliło naśladować mechanizm propagacji sygnału przy spełnieniu określonych warunków. Funkcja przyjmuję wartość jeden jeśli suma wartości wejściowych jest większa od zera, w innym przypadku funkcja przyjmuje wartość zero i neuron nie propaguje sygnału. O funkcji skokowej Heaviside'a zdecydowanie warto wspomnieć ze względów historycznych, jednak obecnie budowane sieci z niej nie korzystają. Główną wadą jest brak ciągłości funkcji oraz pochodna, która uniemożliwia skorzystanie z wykorzystywanego podczas nauki algorytmu propagacji wstecznej[67]. Pochodna funkcji jest równa zero wszędzie po za argumentem równym zero, gdzie osiąga nieskończoną wartość. Wśród pożądanych cech funkcji aktywacji znajdują się takie atrybuty jak nieliniowość, jest to cecha, która pozwala sieci neuronowej odwzorować nieliniowe zależności [47], [37]. Jedynym wyjątkiem od reguły jest stosowanie w problemach regresyjnych funkcji tożsamościowej w ostatniej warstwie wyjściowej.

W ogólności od funkcji aktywacji nie jest wymagana ciągłość pochodnej, grupę szeroko stosowanych funkcji aktywacji zajmują poprawiona jednostka liniowa (ReLU) oraz jej modyfikacje. ReLU jest zdefiniowana jako $g(z) = \max\{0, z\}$ i jej pochodna nie jest określona w $z = 0$, zauważmy jednak, że funkcja ma dobrze zdefiniowaną pochodną lewostronną równą zero oraz prawostronną równą jeden. Implementacja programistyczna może zapewnić zwracanie w zerze podczas nauki sieci jedną z tych wartości. Zauważmy również, że jest bardzo mało prawdopodobne aby algorytm szukający minimum funkcji straty osiągnął punkt minimum, w którym gradient wynosi zero. Arymetyka zmiennoprzecinkowa zapewnia, że będzie to raczej mała wartość ϵ. Wśród zalet funkcji ReLU wymienia się dokładniejsze odwzorowanie obserwowanego w neurobiologii zjawiska – tylko neurony, które otrzymały odpowiednio silny sygnał są aktywowane. Ponadto brak podatności na przeuczenie, podczas inicjalizacji sieci losowymi wagami, tylko około 50% ukrytych neuronów jest aktywowanych. Dodatkowo w funkcjach ReLU nie spotykamy na problem znikającego gradientu uniemożliwiającego uczenie, który pojawia się w funkcjach sigmoidalnych. Funkcja ReLU oraz "Nieszczelna" ReLU zdefiniowana jako $g(z) = \max\{0.1z, z\}$ przedstawione zostały w środkowym rzędzie rysunku 3.2.

Następną grupę stosowanych funkcji aktywacji zajmują funkcje sigmoidalne. Jest to grupa



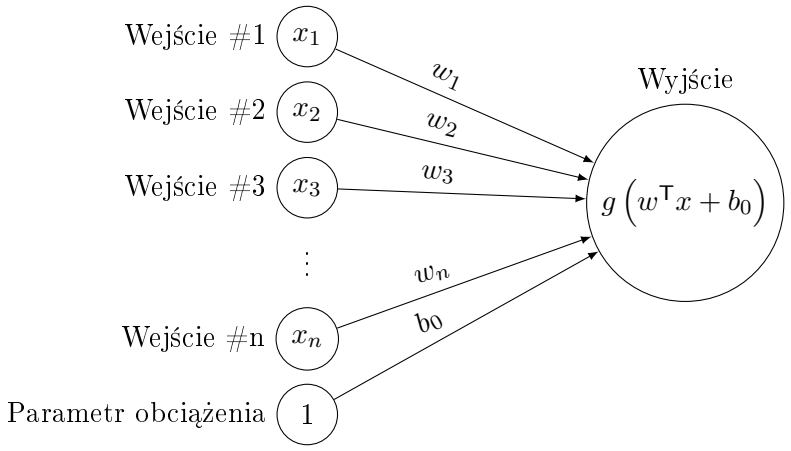
Rysunek 3.2: Kilka przykładów często stosowanych funkcji aktywacji.

monotonicznie rosnących funkcji, których zbiór wartości jest ograniczony przez asymptoty o skończonych wartościach, do których wartość funkcji dąży w $\pm\infty$ [47]. Jednym z najczęściej wykorzystywanych przykładów funkcji sigmoidalnych jest sigmoida zdefiniowana równaniem:

$$\sigma(z) = \frac{1}{1 + e^{-z}}. \tag{3.1}$$

Sigmoida jest różniczkowalna w każdym punkcie co pozwala używać podczas procesu uczenia metod optymizacji wykorzysujących gradient. Ponadto pochodna względem argumentu x wyraża się prostą relacją

$$\frac{d}{dz}\sigma(z) = \sigma(z)\left(1 - \sigma(z)\right). \tag{3.2}$$



Rysunek 3.3: Reprezentacja funkcji dyskryminacyjnej $y(x)$ w postaci diagramu sieci neuronowej, mającej n wejść, parametr obciążenia i jedno wyjście.

Innym przykładem często wykorzystywanej w sztucznych sieciach neuronowych funkcji sigmoidalnej jest tangens hyperboliczny (prawy dolny róg Rys. 3.2). Wzór tej funkcji możemy wyrazić korzystając z definicji sigmoidy

$$\mathrm{tgh}(z) = 2\sigma(2z) - 1. \tag{3.3}$$

Jedną z zalet tej funkcji jest symetryczność względem początku układu współrzędnych. Głównym problemem funkcji sigmoidalnych jest problem znikającego gradientu, który pojawia się dla silnie ujemnych lub silnie dodatnich z. Mała wartość gradientu powoduje, że algorytm uczący napotyka wtedy na problemy ze szkoleniem sieci, problemu można uniknąć kontrolując i ograniczając wartości wag podczas nauki.

Interpretacja probabilistyczna sigmoidy

Poniższy fragment przedstawia interesującą właściwość simoidy. Zastosowanie sigmoidy jako funkcji aktywacji naturalnie wynika z postaci prawdopodobieństwa a posteriori w Bayesowskim podejściu do problemu klasyfikacji dwóch cech [13].

Rozważmy sztuczną sieć neuronową z jedną warstwą ukrytą oraz funkcję dyskryminacyjną $y(x)$ taką, że wektor x jest przypisany do klasy C_1 jeśli $y(x) > 0$ i do klasy C_2 jeśli $y(x) < 0$. W najprostszej, liniowej formie funkcja może być zapisana jako:

$$y(x) = w^{\mathsf{T}}x + b_0. \tag{3.4}$$

Wektor w, to d-wymiarowy wektor wag, natomiast parametr b_0 to parametr obciążenia. Rozważmy funkcję $g(\cdot)$ nazywaną dalej funkcją aktywacji, która jako argument przyjmuje jako argument sumę z równania (3.4):

$$y = g\left(w^{\mathsf{T}}x + b_0\right) \tag{3.5}$$

Załóżmy, że funkcja rozkładu prawdopodobieństwa danych pod warunkiem klasy C_k zadana jest przez wielowymiarowy rozkład normalny z równymi macierzami kowariancji $\Sigma_1 = \Sigma_2 = \Sigma$

$$p(x|C_k) = \frac{1}{(2\pi)^{\frac{d}{2}}\,|\Sigma|^{\frac{1}{2}}} \exp\left[-\frac{1}{2}\left(x - \mu_k\right)^{\mathsf{T}}\Sigma^{-1}\left(x - \mu_k\right)\right]. \tag{3.6}$$

Prawdopodobieństwo a posteriori klasy C_1 można zapisać używając twierdzenia Bayesa:

$$\begin{aligned} p(C_1|x) &= \frac{p(x|C_1)p(C_1)}{p(x|C_1)p(C_1)+p(x|C_2)p(C_2)} \\ &= \frac{1}{1+\frac{p(x|C_2)p(C_2)}{p(x|C_1)p(C_1)}} \\ &= \frac{1}{1+\exp(-a)}, \end{aligned} \tag{3.7}$$

gdzie

$$\begin{aligned} a &= \ln\frac{p(x|C_1)p(C_1)}{p(x|C_2)p(C_2)} \\ &= (\mu_1-\mu_2)^{\mathsf{T}}\Sigma^{-1}x - \frac{1}{2}\mu_1^{\mathsf{T}}\mu_1 + \frac{1}{2}\mu_2^{\mathsf{T}}\Sigma^{-1}\mu_2 + \ln\frac{p(C_1)}{p(C_2)}, \end{aligned} \tag{3.8}$$

pamiętając o tym, że macierz kowariancji jest symetryczna otrzymujemy

$$x = \Sigma^{-1}(\mu_1-\mu_2) \tag{3.9a}$$

$$b_0 = -\frac{1}{2}\mu_1^{\mathsf{T}}\mu_1 + \frac{1}{2}\mu_2^{\mathsf{T}}\Sigma^{-1}\mu_2 + \ln\frac{p(C_1)}{p(C_2)} \tag{3.9b}$$

Zatem widzimy, że użycie funkcji aktywacji w postaci sigmoidy pozwala nie tylko dokonać decyzji klasyfikacji ale również interpretować wynik funkcji dyskryminacyjnej jako prawdopodobieństwa a posteriori.

3.3 Uczenie sieci neuronowej

W biologii, uczeniem nazywamy proces pozyskiwania nowej lub modyfikacji nabytej wiedzy, zachowania lub umiejętności pod wpływem doświadczeń [58]. Systemy uczące się (*machine learning*) to jedna z gałęzi dziedziny sztucznej inteligencji, która zajmuje się badaniem algorytmów oraz modeli matematycznych wykorzystywanych przez systemy komputerowe do stopniowego zwiększania efektywności wykonywania określonego zadania. Wśród metod nauki możemy wyróżnić uczenie nienadzorowane oraz nadzorowane. Pierwsze z nich do nauki wykorzystuje zbiór danych zawierający cechy, a następnie stara się nauczyć użytecznych właściwości dotyczących struktury tego zbioru danych. Algorytmy wykorzystujące uczenie nienadzorowane stosowane są na przykład w problemach segmentacji danych lub detekcji anomalii. Nadzorowane algorytmy uczące się poznają zbiór danych zawierający cechy lecz do każdego przykładu przypisana jest również etykieta, zadaniem algorytmu jest nauczenie się w jaki sposób trafnie przypisywać etykiety [32]. Ten rodzaj algorytmów wykorzystywany jest najczęściej w problemach klasyfikacyjnych oraz regresyjnych. Z matematycznego punktu widzenia, uczenie się to problem optymalizacyjny polegający na poszukiwaniu ekstremum funkcji, która jest kryterium jakości nauki. W poniższym rozdziale opisuję proces nadzorowanego uczenia się jenokierunkowej sieci neuronowej o jednej warstwie ukrytej, której zadaniem będzie nauka relacji $f(X) = Y$.

Dane

U podstaw każdego modelu statystycznego leży zbiór danych służący do nauki modelu nazywany zbiorem treningowym, to on bardzo często determinuje jego wybór i jest źródłem wiedzy dla modelu. Jakość modelu powstałego modelu jest ściśle zależna od jakości i spójności dostarczonych danych. Niech przykładowy zbiór danych treningowych składa się dwóch wektorów, które reprezentują zmienne objaśniające oraz oczekiwane wyniki m jednowymiarowych próbek. Niech pierwszy z nich to wektor $X \in \mathbb{R}^{1\times m}$, a odpowiadające mu wyniki to $Y \in \mathbb{R}^{1\times m}$.

Parametry

Niech sieć przedstawiona na rysunku 3.4 składa się z dwóch warstw: 1) ukrytej zawierającej L neuronów i 2) warstwy wyjściowej składającej się z 1 neuronu. Warstwy są zdefiniowane przez:

1. parametry wartstwy ukrytej, które odwzorowują 1-wymiarowe wektory wejściowe w aktywacje L neuronów: macierz wag $W^h \in \mathbb{R}^{L\times 1}$ i wektor parametru obciążenia $b^h \in \mathbb{R}^{L\times 1}$,
2. parametry wartstwy wyjściowe, które odwzorowują L-wymiarowy wektor aktywacji neuronów ukrytych w jeden neuron wartstwy wyjściowej: macierz wag $W^o \in 1 \times L$ i wektor parametru obciążenia $b^o \in \mathbb{R}^{1\times 1}$.

Inicjalizacja wartości początkowych parametrów pełni istotną rolę w skuteczności procesu uczenia, na przykład inicjalizacja wag z wartością zero uniemożliwi jakąkolwiek naukę. Wagi powinny być są inicjalizowane losowo, jedną z metod skutecznej inicjalizacji wag jest metoda Xaviera [30]. Parametry nauczonego modelu powinny skutkować najlepszą aproksymacją funkcji f.

Propagacja sygnału

Wejście każdego neuronu w warstwie ukrytej jest iloczynem danych wejściowych i odpowiadającej im wagi plus parametr obciążenia. Na przykład dla i-tej próbki danych wejściowych, w l-tym neuronie mamy

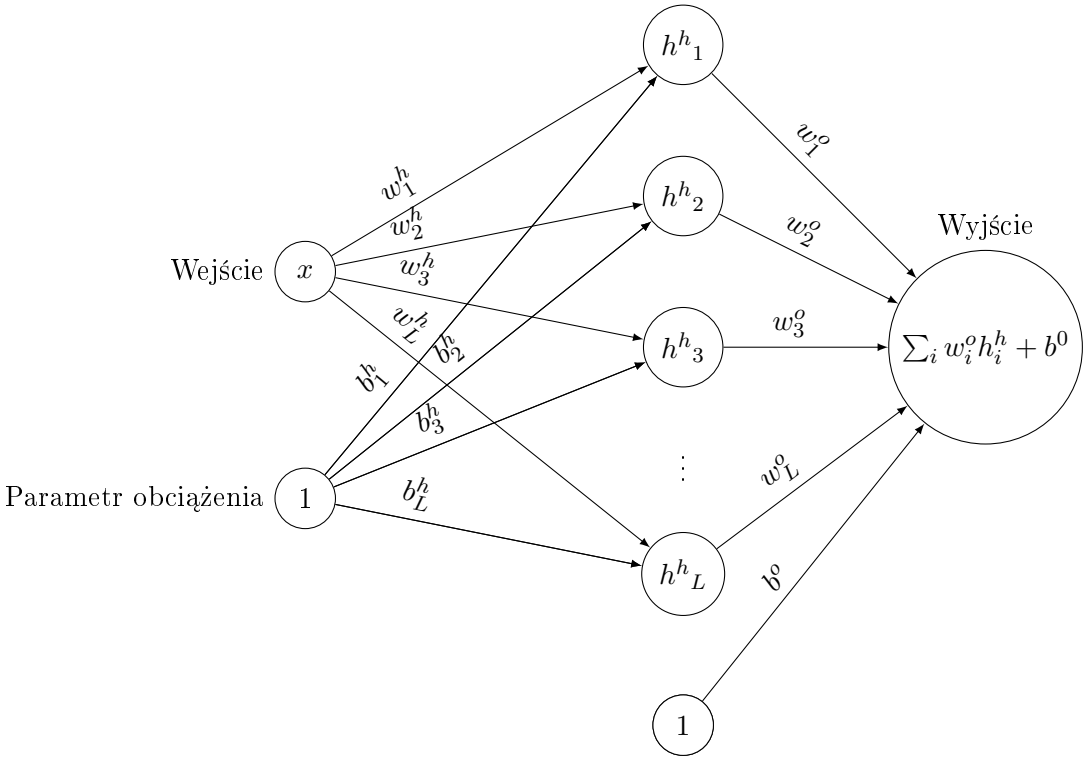
$$a^{h}{}_{l}^{(i)} = W^{h}{}_{l} x^{(i)} + b^{h}{}_{l} \tag{3.10}$$

Niech funkcją aktywacyjną neuronów jest sigmoida $\sigma(a) = 1/(1+e^{-a})$, jako argument przyjmuje ona wejście neuronów:

$$h^{h}{}_{l}^{(i)} = \sigma(a^{h}{}_{l}^{(i)}) \tag{3.11}$$

Neuron warstwy wyjściowej zawiera sumę iloczynów aktywacji neuronów i odpowiadających im wag plus parametr obciążenia.

$$\begin{aligned} a^{o(i)} &= \sum_l W^{o}{}_{l} h^{h}{}_{l}^{(i)} + b^o \\ &= \sum_l W^{o}{}_{l} \sigma(a^{h}{}_{l}^{(i)}) + b^o \\ &= \sum_l W^{o}{}_{l} \sigma(W^{h}{}_{l} x^{(i)} + b^{h}{}_{l}) + b^o \end{aligned} \tag{3.12}$$



Rysunek 3.4: Schemat sieci neuronowej o jednej warstwie ukrytej

Funkcja straty

Funkcja straty lub inaczej funkcja kosztu $J(\Theta)$ to z definicji funkcja przyporządkowująca nieujemną wielkość kary poprzez porównanie zmiennej objaśnianej do wyliczonego estymatora. Nauka modelu polega na umiejętnym minimalizowaniu wartości tej funkcji poprzez modyfikacje parametrów modelu. W problemach regresyjnych bardzo często jako funkcje straty wykorzystuje się błąd średniokwadratowy. Jest to wartość oczekiwana kwadratu błędu pomiędzy estymatorem i wartością estymowaną, w przedstawianym przykładzie funkcja ma postać błędu średniokwadratowego przemnożonego przez czynnik $1/2$ co okazuje się przydatne podczas dalszych obliczeń.

$$\begin{aligned} J^{(i)}(\Theta) &= \frac{1}{2}\left(y^{(i)} - a^{o(i)}\right)^2, \\ J(\Theta) &= \frac{1}{m}\sum_{i=1}^{m} J^{(i)}(\Theta) = \frac{1}{2m}\sum_{i=1}^{m}\left(y^{(i)} - a^{o(i)}\right)^2. \end{aligned} \tag{3.13}$$

Propagacja wsteczna

Po obliczeniu skalarnej straty wynikającej z równania (3.13) należy przekazać tę informację do poprzednich warstw sieci i zaktualizować parametry. Algorytm propagacji wstecznej [67] pozwala informacji o błędzie płynąć wstecz przez sieć, aby obliczyć gradient. Użycie reguły

łańcuchowej umożliwia zapisanie procedury propagacji wstecznej i obliczenie gradientu funkcji straty względem parametrów sieci neuronowej.

Na początku policzmy pochodną funkcji kosztu względem wyniku wartstwy wyjściowej.

$$\frac{\partial J}{\partial a^{o(i)}} = \frac{1}{m}\left(y^{(i)} - a^{o(i)}\right), \tag{3.14}$$

następnie policzmy gradient wyjścia neuronów ukrytych:

$$\frac{\partial J}{\partial h^{h}_{l}{}^{(i)}} = \frac{\partial J}{\partial a^{o(i)}}\frac{\partial a^{o(i)}}{\partial h^{h}_{l}{}^{(i)}} = \frac{\partial J}{\partial a^{o(i)}} W^{o}{}_{l}, \tag{3.15}$$

co umożliwia obliczenie gradientu względem wejścia neruonów ukrytych:

$$\frac{\partial J}{\partial a^{h}_{l}{}^{(i)}} = \frac{\partial J}{\partial h^{h}_{l}{}^{(i)}}\frac{\partial h^{h}_{l}{}^{(i)}}{\partial a^{h}_{l}{}^{(i)}} = \frac{\partial J}{\partial h^{h}_{l}{}^{(i)}} h^{h}_{l}{}^{(i)}(1 - h^{h}_{l}{}^{(i)}), \tag{3.16}$$

gdzie została wykorzystana relacja:

$$\frac{\partial \sigma(x)}{\partial x} = \sigma(x)(1 - \sigma(x)).$$

Ostatecznie możemy policzyć gradienty względem parametrów sieci, np. dla warstwy wejściowej:

$$\frac{\partial J}{\partial W^{o}{}_{l}} = \sum_{i} \frac{\partial J}{\partial a^{o(i)}}\frac{\partial a^{o(i)}}{\partial W^{o}{}_{l}} = \sum_{i} \frac{\partial J}{\partial a^{o(i)}} h^{h}_{l}{}^{(i)}, \tag{3.17}$$

$$\frac{\partial J}{\partial b^{o}} = \sum_{i} \frac{\partial J}{\partial a^{o(i)}}\frac{\partial a^{o(i)}}{\partial b^{o}} = \sum_{i} \frac{\partial J}{\partial a^{o(i)}}. \tag{3.18}$$

Ostatnim, bardzo istotnym elementem tworzonego modelu jest wybór algorytmu uczącego, poszukującego minimum funkcji straty (3.13). Algorytm propagacji wstecznej odnosi się do metody obliczania pochodnych i pozwala na propagację informacji o błędzie do parametrów sieci lecz realizacja uczenia się za jego pomocą wykorzystywana jest przez szereg algorytmów nazywanych algorytmami uczenia się. Ich przegląd i analiza znajdują się w rozdziale 4.5. Proces uczenia modelu sieci neuronowej zazwyczaj kończy się po ustalonej liczbie epok lub osiągnięciu odpowiedniej wartości funkcji straty, metody tzw. wczesnego zatrzymania algorytmu są dokładniej opisana w rozdziale **??**. Jedna epoka oznacza jedną iterację po wszystkich elementach zbioru treningowego.

3.4 Uniwersalne twierdzenie aproksymacyjne

Według uniwersalnego twierdzenia aproksymacyjnego jednokierunkowa sieć neuronowa z jedną warstwą ukrytą i skończoną ale wystarczająco dużą liczbą neuronów, może przybliżyć dowolną mierzalną funkcję borelowską z jednej przestrzeni o skończonej liczbie wymiarów do drugiej z dowolnym niezerowym poziomem błędu. W tej części prezentuję dowód wizualny działania uniwersalnego twierdzenia aproksymacyjnego pokazując wpływ ilości neuronów w warstwie ukrytej oraz stosowanej funkcji aktywacji na jakość przybliżenia.

W 1989 roku Cybenko [20] udowodnił uniwersalne twierdzenie aproksymacyjne dla jednokierunkowego perceptronu wielowarstwowego z sigmoidalną funkcją aktywacji. Funkcje sigmoidalne to rodzina funkcji szeroko stosowanych w jednokierunkowych sieciach neuronowych, szczególnie tych stworzonych w celach regresji.

Definicja 3.4.1. Funkcja $\sigma : \mathbb{R} \to \mathbb{R}$ jest funkcją sigmoidalną jeśli

$$\sigma(x) \to \begin{cases} 1 & \text{gdy} \quad x \to +\infty \\ 0 & \text{gdy} \quad x \to -\infty \end{cases}$$

Definicja 3.4.2. Funkcja σ jest funkcją dyskryminacyjną jeśli dla miary $\mu \in M(I_n)$ zachodzi

$$\int_{I_n} \sigma\left(w^{\mathsf{T}} x + b_0\right) d\mu(x) = 0 \tag{3.19}$$

dla każdego $w \in \mathbb{R}$ i $b_0 \in \mathbb{R}$ co implikuje, że $\mu = 0$.

Twierdzenie 3.4.1. *Każda ograniczona, mierzalna funkcja sigmoidalna σ jest funkcją dyskryminacyjną. W szczególności każda ciągła funkcja sigmoidalna jest dyskryminacyjna. [20]*

Twierdzenie 3.4.2 (Uniwersalne twierdzenie aproksymacyjne wg. Cybenki)**.** *Niech σ będzie ciągłą funkcją dyskryminacyjną, wtedy skończona suma*

$$G(x) = \sum_{i=1}^{N} w_i^o \sigma\left({w_i^h}^{\mathsf{T}} x + b_i^h\right) \tag{3.20}$$

jest gęsta w $C(I_n)$. Innymi słowy, dla danej funkcji $f \in C(I_n)$ i $\epsilon > 0$, istnieje suma $G(x)$ mająca powyższą postać, dla której

$$|G(x) - f(x)| < \epsilon \qquad \forall x \in I_n$$

Jeszcze w tym samym roku, po pracy Cybenki ukazała się praca Hornika, Stinchcombe'a i White'a [37], którzy udowodnili prawdziwość powyższego twierdzenia dla dowolnej "ściskającej" funkcji aktywacji. Do tej klasy należy zaliczyć funkcje, które mają ograniczony zbiór wartości i są niemalejące jako przykład może posłużyć nieciągła funkcja skokowa Heaviside'a.

Definicja 3.4.3. Funkcja $\Psi : \mathbb{R} \to [0, 1]$ jest funkcją ściskającą jeśli zachodzą wszystkie poniższe warunki:

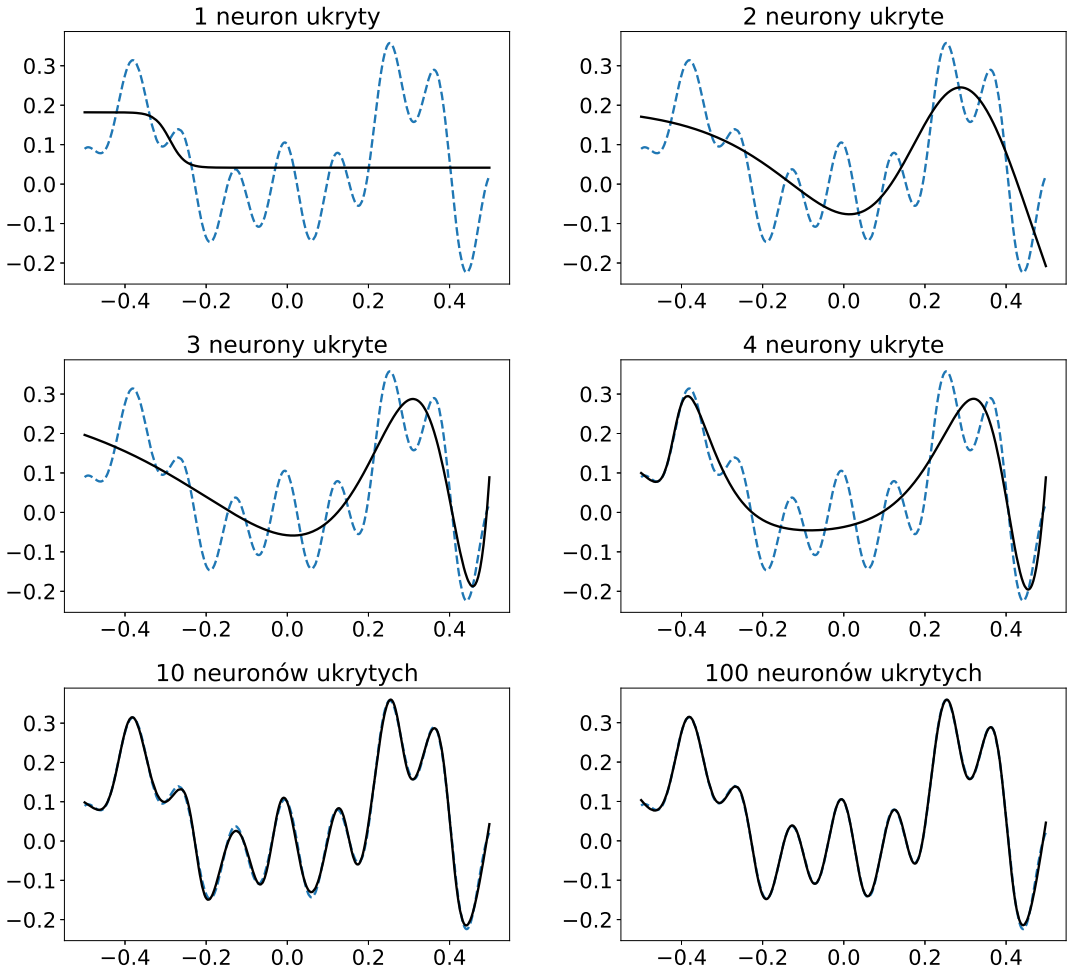
1. Ψ jest niemalejąca,
2. $\lim_{\lambda \to \infty} \Psi(\lambda) = 1$,
3. $\lim_{\lambda \to -\infty} \Psi(\lambda) = 0$.

Następnie uniwersalne twierdzenie aproksymacyjne zostało rozszerone o bardzo szeroką klasę funkcji aktywacji niebędących wielomianami [48]. Obejmuje ona bardzo często obecnie stosowaną poprawioną jednostkę liniową (ReLU).

Aby zademonstrować działanie uniwersalnego twierdzenia aproksymacyjnego nauczono sieci neuronowe o jednej warstwie ukrytej lecz o różnej ilości neuronów przebiegu funkcji zdefiniowanej poniższym równaniem:

$$f(x) = 3 \log(x + 2) \left(0.4x^2 + 0.3x \sin(15(x - 1)) + 0.05 \cos(50(x - 1))\right) \tag{3.21}$$

Dane uczące składały się z tysiąca par punktów $(x, f(x))$ z przedziału $x \in [-0.5, 0.5]$, zadaniem modelu była jak najdokładniejsza aproksymacja wartości funkcji poprzez minimalizację błędu średniokwadratowego. Rysunek 3.6 przedstawia uzyskaną zależność wartości funkcji błędu

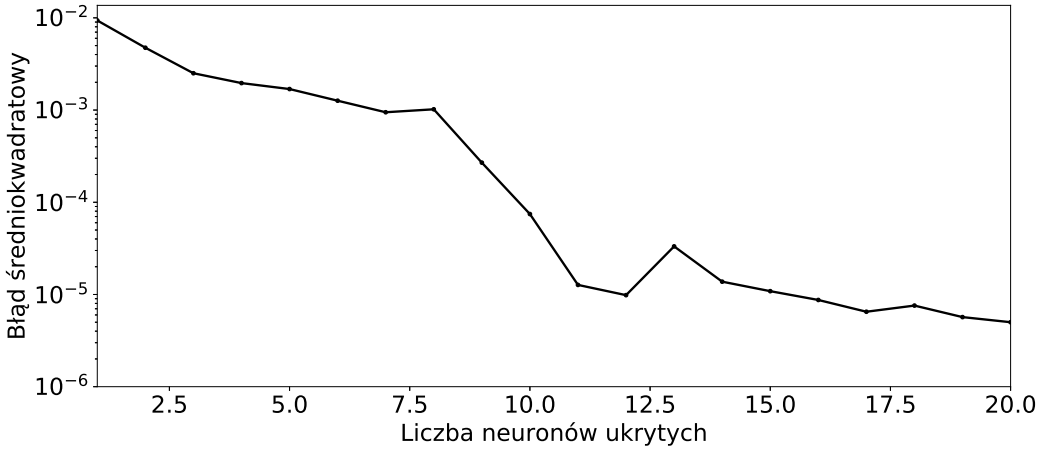


Rysunek 3.5: Aproksymacja funkcji o równaniu (3.21) przez sieć neuronową w zależności od liczby neuronów w warstwie ukrytej. Rysunki przedstawiają aproksymacje funkcje otrzymane dla 1, 2, 3, 4, 10 i 100 neuronów. Przebieg funkcji (3.21) to niebieska przerywana linia, wynik modelu to ciągła czarna linia.

od liczby neuronów w warstwie ukrytej przy wykorzystaniu sigmoidy jako funkcji aktywacji. Wartość błędu maleje eksponencjalnie wraz ze wzrostem liczby neuronów osiągając wartość $\simeq 10^{-5}$ dla 20 neuronów. Natomiast przebieg aproksymowanych funkcji możemy zaobserwować na rysunku 3.5. Dla 1 neuronu w wartstwie ukrytej, funkcją wyjściową jest po prostu sigmoida o postaci

$$G(x) = w_1^o \sigma\left(w_1^h x + b_1\right),$$

następne neurony dodają do funkcji $G(x)$ kolejne składniki tej postaci. Dla przedstawianego przykładu, sieć o 10 neuronach ukrytych bardzo dobrze odwzorowuje przebieg funkcji uwzględ-



Rysunek 3.6: Błąd średniokwadratowy w zależności od liczby neuronów w warstwie ukrytej podczas aproksymacji funkcji z równania (3.21).

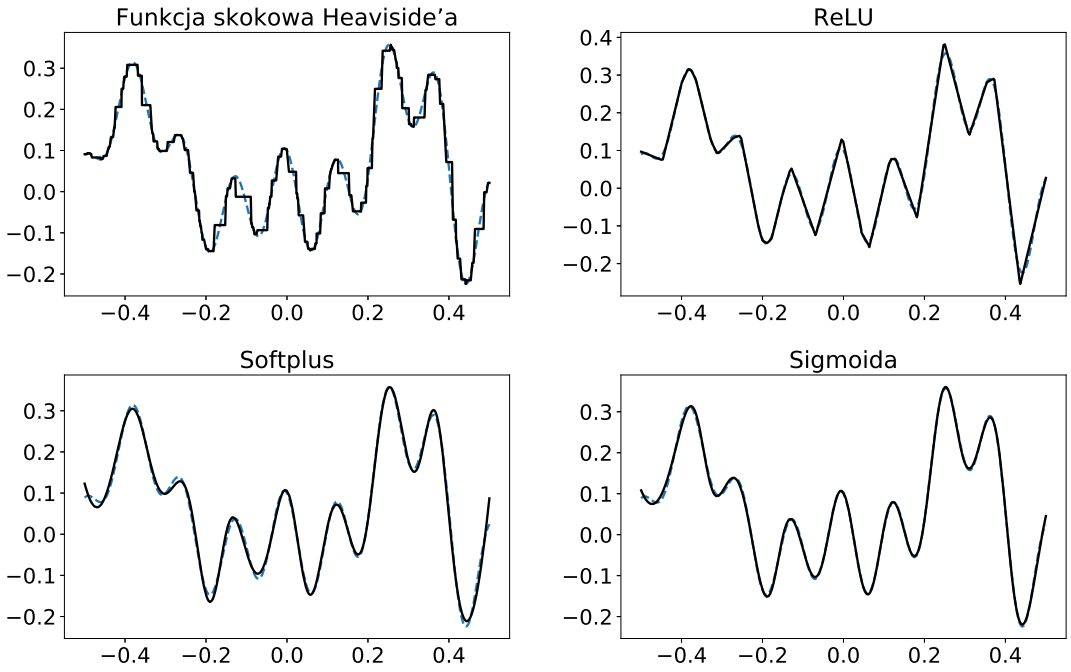
niając wszystkie wypukłości.

Zgodnie z treścią twierdzenia udowodnionego w 1993 roku [48], sieć neuronowa ma zdolność przybliżania funkcji nawet przy wykorzystaniu nieciągłych i nieograniczonych funkcji aktywacji. Rysunek 3.7 pokazuje przybliżenia otrzymane przy użyciu funkcji: (*a*) Heaviside'a, (*b*) ReLU, (*c*) softplus oraz (*d*) sigmoidy dla 100 neuronów w warstwie ukrytej. Można zauważyć, że zapewnienie niskiej wartości błędu przy wykorzystaniu nieciągłych funkcji aktywacji wymaga znacznie większej ilości neuronów niż w przypadku użycia ciągłych funkcji jak softplus czy sigmoida. Należy również pamiętać, że wykorzystanie funkcji Heaviside'a uniemożliwia skorzystanie z wprowadzonej w poprzednim podrozdziale metody propagacji wstecznej, ponieważ pochodna w punkcie $x = 0$ jest nieskończona. Chociaż uniwersalne twierdzenie aproksymacyjne zapewnia istnienie perceptronu wielowarstwowego, który odwzoruje tę funkcję z dowolnie małym błędem to nauka modelu może się nie powieść, ponieważ algorytm uczący może nie być w stanie znaleźć wartości parametrów odpowiadających szukanej funkcji.

3.5 Kompromis między obciążeniem i wariancją

Jednym z głównych wyzwań podczas budowy predykcyjnych modelów statystycznych jest stworzenie modelu, który będzie miał możliwość uogólnienia. Uogólnienie to zdolność algorytmu do dobrego działania zarówno na danych, które posłużyły do nauki modelu i na nowych, wcześniej nieznanych pomiarach. Aby to osiągnąć zestaw dostępnych danych dzieli się na dwa zbiory: szkoleniowy oraz walidacyjny. Minimalizacja wartości funkcji straty obliczonej na podstawie zbioru szkoleniowego prowadzi do nauki modelu, natomiast obserwacja wartości błędu obliczonego na podstawie zbioru walidacyjnego pozwala sprawdzić jakość uogólnienia modelu.

Błąd predykcyjnych modelów statystycznych może być rozłożony na dwa główne składniki – błąd spowodowany obciążeniem i wariancją. Zrozumienie tych błędów umożliwia skonstruowanie modelu, który równocześnie minimalizuje zarówno obciążenie jak i wariancję, co pozwala uniknąć nadmiernego dopasowania lub niedopasowania modelu. W poniższej sekcji



Rysunek 3.7: Porównanie jakości aproksymacji funkcji o równaniu (3.21) w zależności od zastosowanej funkcji aktywacji. Przebieg funkcji (3.21) to niebieska przerywana linia, wynik modelu o 100 neuronach ukrytych to ciągła czarna linia.

zdefiniuję te błędy i wyjaśnię dlaczego dobry model powinien zachowywać kompromis pomiędzy obciążeniem i wariancją.

Obciążenie

Obciążenie estymatora to różnica pomiędzy oczekiwaną wartością estymatora i prawdziwą wartością estymowanego parametru. Estymator, którego obciążenie wynosi zero nazywamy estymatorem nieobciążonym, w przeciwnym razie estymatorem obciążonym. Załóżmy, że model statystyczny opisywany jest przez parametr θ, tak że $P_\theta(x) = P(x \mid \theta)$, natomiast $\hat{\theta}$ to estymator θ będący funkcją danych x. Zakładamy, że dane są zgodne z nieznanym rozkładem $P_\theta(x) = P(x \mid \theta)$, natomiast θ ma ustaloną wartość i jest parametrem tego rozkładu, następnie konstruujemy estymator $\hat{\theta}$.

Definicja 3.5.1. Obciążenie estymatora $\hat{\theta}$ względem parametru θ definiujemy jako:

$$\text{bias}_\theta\left[\hat{\theta}\right] = \mathbb{E}_{x|\theta}\left[\hat{\theta}\right] - \theta = \mathbb{E}_{x|\theta}\left[\hat{\theta} - \theta\right]. \tag{3.22}$$

Przykład:
Rozważmy zbiór niezależnych i równomiernie rozłożonych zmiennych losowych $\{x^{(1)}, \ldots, x^{(m)}\}$ zgodnie z rozkładem zero-jedynkowym o średniej θ,

$$P(x^{(i)}; \theta) = \theta^{x^{(i)}} (1 - \theta)^{(1 - x^{(i)})}. \tag{3.23}$$

Niech estymatorem wartości średniej rozkładu będzie średnia arytmetyczna próbki:

$$\theta_m = \frac{1}{m}\sum_{i=1}^{m} x^{(i)} \tag{3.24}$$

Wtedy obiążenie średniej arytmetycznej próbki wynosi:

$$\begin{aligned}
\text{bias}\left(\hat{\theta}_m\right) &= \mathbb{E}\left[\hat{\theta}_m\right] - \theta \\
&= \mathbb{E}\left[\frac{1}{m}\sum_{i=1}^{m} x^{(i)}\right] - \theta \\
&= \frac{1}{m}\sum_{i=1}^{m} \mathbb{E}\left[x^{(i)}\right] - \theta \\
&= \frac{1}{m}\sum_{i=1}^{m} \sum_{x^{(i)}=0}^{1} \left(x^{(i)}\theta^{x^{(i)}}(1-\theta)^{(1-x^{(i)})}\right) - \theta \\
&= \frac{1}{m}\sum_{i=1}^{m} (\theta) - \theta \\
&= \theta - \theta = 0.
\end{aligned} \tag{3.25}$$

Ponieważ obciążenie estymatora równe jest zero, średnia arytmetyczna próbki jest nieobciążonym estymatorem średniej rozkładu zero-jedynkowego.

Wariancja

Wariancja to kolejna właściwość estymatora, którą warto wziąć pod uwagę. W statystyce definiuje się ją jako wartość oczekiwaną kwadratu różnicy odchyleń wartości cechy od wartości oczekiwanej.

Definicja 3.5.2. Wariancja zmiennej losowej X o wartości oczekiwanej μ, zdefiniowana jest wzorem

$$\text{Var}(X) = \mathbb{E}\left[(X-\mu)^2\right] \tag{3.26}$$

Przykład:
Ponownie rozważmy zbiór niezależnych i równomiernie rozłożonych zmiennych losowych $\{x^{(1)}, \ldots, x^{(m)}\}$ zgodnie z rozkładem zero-jedynkowym o średniej θ. Obliczmy wariancję estymatora średniej rozkładu $\theta_m = \frac{1}{m}\sum_{i=1}^{m} x^{(i)}$

$$\begin{aligned}
\text{Var}\left(\hat{\theta}_m\right) &= \text{Var}\left(\frac{1}{m}\sum_{i=1}^{m} x^{(i)}\right) \\
&= \frac{1}{m^2}\sum_{i=1}^{m} \text{Var}\left(x^{(i)}\right) \\
&= \frac{1}{m^2}\sum_{i=1}^{m} \theta\,(1-\theta) \\
&= \frac{1}{m^2} m\theta\,(1-\theta) \\
&= \frac{1}{m}\theta\,(1-\theta)\,.
\end{aligned} \tag{3.27}$$

Otrzymana wariancja jest malejącą funkcją zależną od m, które jest wielkością próbki losowej. Jest to zgodny z intuicją rezultat, wariancja estymatora maleje wraz ze zwiększeniem liczby przykładów w zbiorze danych.

Kompromis między obciążeniem i wariancją

Celem modelu statystycznego zbudowanego na potrzeby tej pracy jest estymacja funkcji. Jeśli oznaczymy przez Y zmienną objaśnianą i przez X zmienną objasniającą, możemy przyjąć, że istnieje relacja łącząca obie te zmienne

$$Y = f(X) + \varepsilon, \tag{3.28}$$

gdzie ε to część Y niemożliwa to przewidzenia na podstawie X, jest to błąd losowy o rozkładzie normalnym i wartości oczekiwanej zero, $\varepsilon \sim \mathcal{N}(0, \sigma_\varepsilon)$. Następnie możemy stworzyć model, którego wynikiem będzie estymacja funkcji $f(X)$ – $\hat{f}(X)$, błąd średniokwadratowy predykcji modelu (MSE) w punkcie x to:

$$\mathrm{MSE}(x) = \mathbb{E}\left[\left(Y - \hat{f}(x)\right)^2\right] \tag{3.29}$$

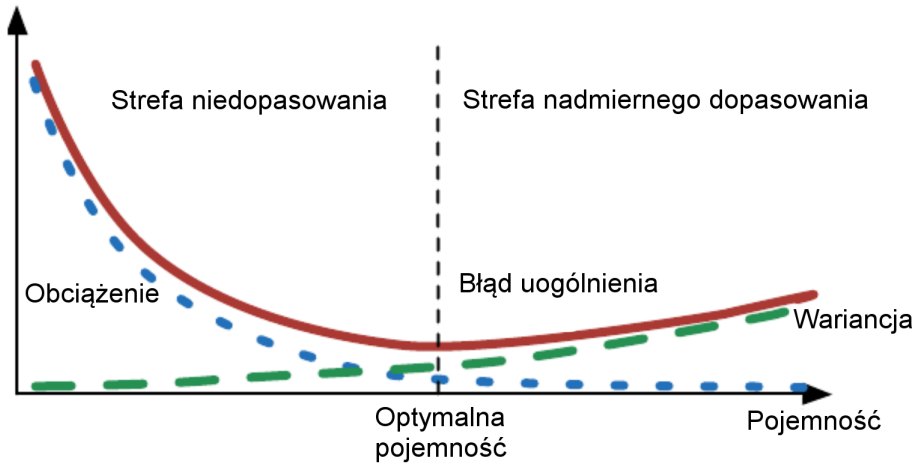
Możemy go rozłożyć na składniki opisujące obiążenie i wariancję.

$$\mathrm{MSE}(x) = \left(\mathbb{E}\left[\hat{f}(x)\right] - f(x)\right)^2 + \mathbb{E}\left[\left(\hat{f}(x) - \mathbb{E}\left[\hat{f}(x)\right]\right)^2\right] + \sigma_\varepsilon^2 \tag{3.30}$$

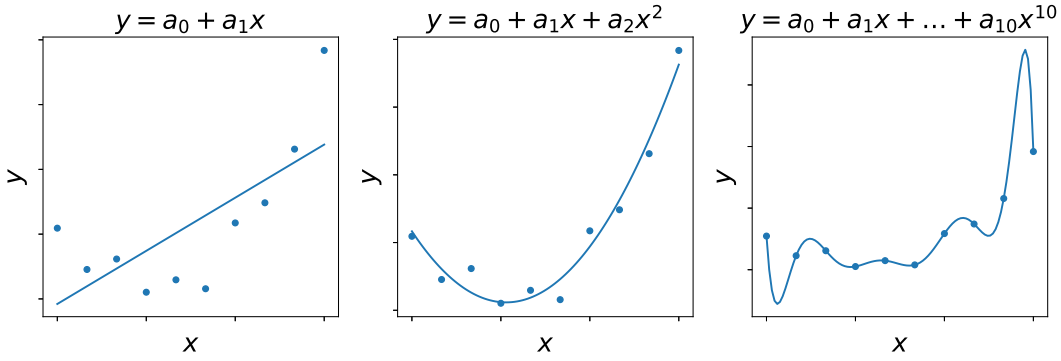
$$\mathrm{MSE}(x) = \mathrm{Obciążenie}^2 + \mathrm{Wariancja} + \mathrm{Nieusuwalny\ błąd}$$

Pojemność modelu to jego zdolność do dopasowania do wielu różnych funkcji [32]. Modele o dużej pojemności są zdolne odwzorować bardziej złożone zależności między zmiennymi lecz zagrożeniem płynącym z ich strony jest nadmierne dopasowanie modelu do danych treningowych. Rysunek 3.8 przedstawia typową zależność obciążenia i wariancji w zależności od pojemności modelu. Wraz ze wzrostem pojemności obciążenie ma tendencję malejącą, natomiast wariancja rosnącą co skutkuje charakterystycznym U-kształtem sumy tych błędów - błędu uogólnienia. Model o optymalnej pojemności minimalizuje błąd uogólnienia co charakteryzuje się zachowanym kompromisem między wielkością obciążenia i wariancji.

Wysokie obciążenie może powodować pominięcie istotnych zależności pomiędzy zmienną objaśnianą i zmiennymi objaśniającymi co prowadzi do niedopasowania. Wysoka wariancja może powodować dopasowanie modelu do szumu znajdującego się w danych co skutkuje nadmiernym dopasowaniem. Rysunek 3.8 przedstawia przykładowy zbiór szkoleniowy tzn. dziesięć wylosowanych punktów spełniających zależność $y = x^2 + \varepsilon$. Do zbioru danych dopasowyjemy trzy modele, funkcję liniową, kwadratową oraz wielomian 10 stopnia. Łatwo zauważyć, że funkcja liniowa nie może odwzorować odpowiedniego parabolicznego kształtu, chociaż poprawnie wskazuje trend jest to model o dużym obciążeniu. Krzywa będąca wielomianem 10 stopnia mimo tego, że przechodzi przez wszystkie punkty pomiarowe prawdopodobnie będzie skutkowała największym błędem w przypadku dostarczenia nowych danych, model charakteryzuje się bardzo dużą wariancją. Wielomian drugiego stopnia dobrze odwzorowuje strukture danych i spodziewamy się, że nie zmieni się to w przypadku uwzględnienia nowych danych, model ma małe obciążenie i wariancję



Rysunek 3.8: Zależność błędu modelu spowodowanego obciążeniem i wariancją od pojemności. Rysunek przetłumaczony na język polski z [32].



Rysunek 3.9: Wykresy punktowe przedstawiające dziesięć wygenerowanych punktów spełniających zależność $y = x^2 + \varepsilon$ linie ciągłe oznaczają dopasowane krzywe, od lewej odpowiednio wielomian pierwszego, drugiego i dwudziestego stopnia.

Rozdział 4

Metodologia analizy

4.1 Keras

Keras jest interfejsem API wysokiego poziomu służącym do tworzenia i szkolenia modeli głębokiego uczenia. Początkowo Keras został opracowany dla naukowców, którzy mogli dzięki niemu dokonywać szybkich eksperymentów i symulacji. Ponieważ jest rozpowszechniany pod licencją MIT (może być za darmo wykorzystywany w projektach komercyjnych) zdobył dużą popularność. Dziś ma on kilka set tysięcy użytkowników, od nauczycieli akademickich po inżynierów oprogramowania pracujących zarówno w start-upach jak i dużych firmach, i hobbystów. Jego zalety są wykorzystywane między innymi w wiodących ośrodkach naukowych takich jak Europejska Organizacja Badań Jądrowych CERN i setkach firm, z których największe to Google, Netflix, Uber, Yelp, Opera Software [27]. Ponadto jego zalety zostały docenione przez analityków całego świata, którzy chętnie wykorzystują jego możliwości w konkursach Kaggle. Kaggle to platforma internetowa, która organizuje konkursy na najlepsze modele służące do przewidywania i opisywania zbiorów danych przesyłanych przez firmy i użytkowników, wiele z rywalizacji zostało wygranych przez modele zbudowane przy użyciu wspomnianego interfejsu API. Opisywane w następnych rozdziałach modele sieci neuronowych zostały zaprogramowane przy użyciu biblioteki Keras.

Do największych zalet Keras należą:

- posiada przyjazny użytkownikowi interfejs, który ułatwia szybkie prototypowanie modeli sieci neuronowych;
- prosty i spójny interfejs zoptymalizowany pod kątem typowych przepadków użycia;
- zapewnia przejrzyste informacje zwrotne dotyczące błędów użytkownika;
- obsługuje dowolne architektury sieciowe: modele z wieloma wejściami lub wieloma wyściami;
- posiada wbudowane wsparcie dla splotowych sieci neuronowych oraz rekurencyjnych sieci neuronowych;
- pozwala na bezproblemowe działanie tego samego kodu na CPU oraz GPU.

Keras jest biblioteką, o której można powiedzieć, że zapewnia narzędzia służące do zbudowania modelu sieci neuronowych natomiast w minimalnym stopniu pozwala użytkownikom na

ingerencję w ich strukturę. W zamian wykorzystuje wyspecjalizowaną i dobrze zoptymalizowaną bibliotekę wyspecjalizowaną w operacjach na tensorach. Szczególnie szybko wykonują się obliczenia numeryczne typowe dla algorytmów uczenia maszynowego takich jak mnożenie macierzy i obliczanie gradientu. Można wybierać wśród trzech istniejących implementacji, każda z nich ma otwarte źródło. Pierwsza z nich wykorzystuje Tensorflow opracowany i rozwijany przez Google'a, druga korzysta z Theano opracowanego i rozwijanamego przez LISA Lab w Uniwersytecie Montrealskim, ostatnia i najmniej popularna wykorzystuje CNTK opracowane i rozwijane przez Microsoft. Obecnie najczęściej wykorzystywany jest TensorFlow, został on także wykorzystany w tej pracy. Alternatywą dla Kerasa jest niedawno powstały, zdobywający coraz większą popularność projekt Torch finansowany przez Facebooka.

Proces budowy oraz treningu modelu sieci neuronowej jest bardzo prosty i wymaga wykonania następujących kroków

1. Zdefiniuj swoje dane treningowe: dane wejściowe i dane wyjściowe
2. Zdefiniuj warstwy swojej sieci neuronowej, które przekształcają dane wyjściowe w wyjście
3. Skonfiguruj proces uczenia poprzez wybranie funkcji straty, algorytmu szukającego minimum funkcji straty
4. Przeprowadź odpowiednią do wytrenowania sieci ilość iteracji

Ponizszy przykład prezentuje jak proste jest zbudowanie i wytrenowanie bardzo podstawowego, wymagającego minimum zaangażowania przykładu sieci neuronowej przy użyciu biblioteki Keras.

```
#Zaimportuj wymagane pliki
from keras import models
from keras import layers

#Zainicjalizuj model
model = models.Sequential()

#Dodaj pierwszą warstwę
model.add(layers.Dense(units = 10, activation = 'sigmoid',
    input_shape = 2))

#Dodaj drugą warstwę
model.add(layers.Dense(units = 5, activation = 'tanh'))

#Dodaj warstwę wyjsciową
model.add(layers.Dense(units = 1))

#Skompiluj model
model.compile(optimizer = 'rmsprop', loss='mse')

#Trenuj model
model.fit(inputs = X, outputs = Y, epochs = 100)
```

Zdefiniowana powyżej sieć składa się z dwóch warstw ukrytych o odpowiednio 10 i 5 neuronach. Funkcją aktywacji w pierwszej warstwie jest sigmoida, dane wejściowe zawierają dwie

cechy, które posłużą do zbudowania modelu, druga warstwa wykorzystuje tangens hiperboliczny jako funkcję aktywacji. Wynikiem sieci jest zgodnie z definicją warstwy wyjściowej jednowymiarowy wektor. Model podczas nauki minimalizuje błąd średniokwadratowy, wykorzystuje do tego algorytm *rmsprop*, trenownanie modelu skończy się po 100 pełnych iteracjach zbioru danych.

```
#X_new - nowe dane

#Stwórz predykcje
Y_new = model.predict(X_new)
```

Mając gotowy model i zapewniając nowe dane wejściowe możemy z łatwością wygenerować przewidywania sieci używając funkcji *predict*.

4.2 Szacowanie niepewności przewidywania modelu

Każdy pomiar eksperymentalny obarczony jest niepewnością wyniku. Niedokładność pomiaru pochodzi nie tylko z niedoskonałości aparatury i zmysłów obserwatora, ale jest nieodłączną cechą takiej operacji. W przypadku analizowanych poniżej danych mamy doczynienia z niepewnościami pomiarów oraz z błędami systematycznymi wynikającymi z rozbieżności pomiędzy pomiarami całkowitych przekrojów czynnych i pomiarami podłużnej i poprzecznej składowej polaryzacji protonów dającymi w rezultacie stosunki funkcji postaci protonu. Na pierwszy zestaw analizowanych danych składa 426 pomiarów przekrojów czynnych σ, każdy punkt pomiarowy oprócz zmiennej objaśnianej zawiera przypisaną do niej niepewność pomiaru $\Delta\sigma$. Następne dwie kolumny to zmiennej objaśniające Q^2 oraz ϵ. Bazując na idei zaproponowanej w [26], wykorzystując niepewność pomiaru oraz błędy systematycze możemy wygenerować następne zestawy danych, które będą zawierały wartości przekrojów czynnych z zakresów, w których były możliwe do zmierzenia dla zadanych wartości Q^2 i ε. Następnie każda z replik posłuży do treningu osobnej sieci neuronowej, co pozwoli otrzymać rozkład funkcji $\sigma^{(net)}(Q^2, \varepsilon)$. W pracy [26] pokazano, że tak otrzymana przestrzeń funkcyjna zapewnia nieobciążoną estymację wszystkich pomiarów doświadczalnych przy równoczesnym otrzymaniu gładkiej interpolacji.

Jako przykład posłużą dane wykorzystane do treningu pierwszego modelu statystycznego, natomiast należy pamiętać, że wszystkie zbiory danych posiadające wyznaczoną niepewność pomiaru mogą zostać wykorzystane do powtórzenia poniższej procedury.

Zakładając, że pomiary eksperymentalne $\sigma^{(exp)}$ są wartościami zmiennej losowej o rozkładzie normalnym $\mathcal{N}\left(\sigma^{(exp)}, \Delta\sigma_i^{(exp)}\right)$ możemy wygenerować kolejne pomiary korzystając z poniższego równania:

$$\sigma_{a,i}^{(art)(k)} = \sigma_i^{(exp)} + \mathcal{N}\left(0, \Delta\sigma_i^{(exp)}\right)^{(k)}, \tag{4.1}$$

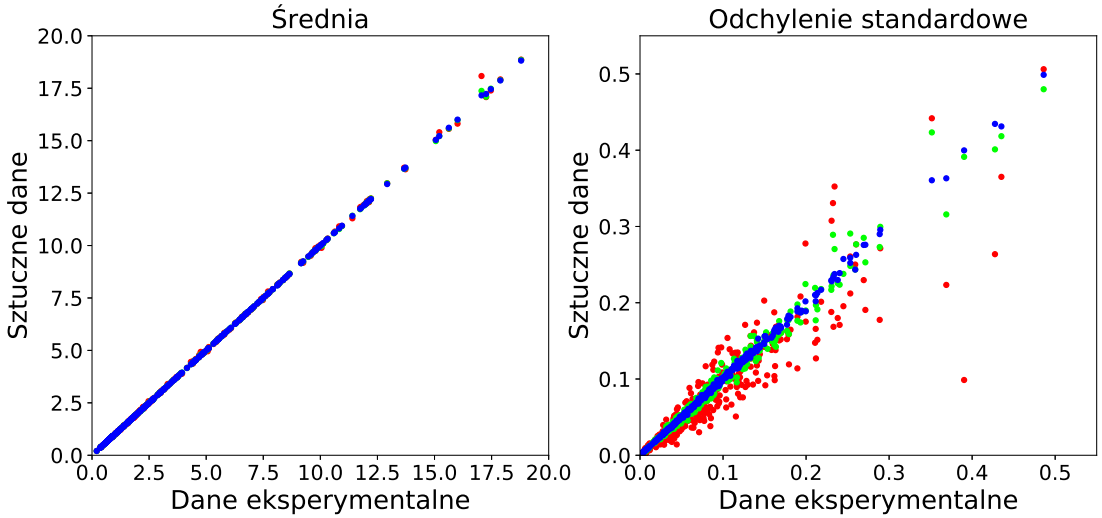
gdzie $\sigma_i^{(exp)}$ to kolejny pomiar eksperymentalny a k to kolejny wygenerowany zestaw danych. Korzystając z opisanej wyżej metody kluczowy jest wybór optymalnej wartości liczby replik N_{rep} tak aby rozkład wygenerowanych danych zawierał charakterystyki zgodne z danymi eksperymantalnymi. Aby dokonać wyboru odpowiedniej wartości N_{rep} porównano wartości średnie oraz odchylenie standardowe próbek po wygenerowaniu sztucznych danych. Rysunek 4.1

przedstawia dwa wykresy punktowe powyższych wartości dla 10, 100 oraz 1000 replik. Szczególny wpływ ilości wygenerowanych danych widoczny jest w części przedstawiającej porównanie odchyleń standardowych. Większa liczba klonów powoduje, że charakterystyki rozkładów wygenerowanych oraz eksperymentalnych danych są bardziej zgodne, co na wykresach prezentuje się jako ułożenie punktów wzdłuż prostej $y = x$. Wartość średnia oraz odchylenie standardowe sztucznych danych zostały zdefiniowane w równaniach (4.2a) oraz (4.2b).

$$\left\langle \sigma_{a,i}^{(art)} \right\rangle_{rep} = \frac{1}{N_{rep}} \sum_{k=1}^{N_{rep}} \sigma_{a,i}^{(art)(k)} \,, \tag{4.2a}$$

$$\left\langle \Delta\sigma_{a,i}^{(art)} \right\rangle_{rep} = \sqrt{\left\langle \sigma_{a,i}^{(art)2} \right\rangle_{rep} - \left\langle \sigma_{a,i}^{(art)} \right\rangle_{rep}^{2}} \,. \tag{4.2b}$$

Aby wskazać jak bardzo wygenerowane dane różnią się od danych eksperymentalnych zdefi-



Rysunek 4.1: $\left\langle \sigma_{a,i}^{(art)} \right\rangle$ vs. $\sigma_i^{(exp)}$ po lewej oraz $\left\langle \Delta\sigma_{a,i}^{(art)} \right\rangle$ vs. $\Delta\sigma_i^{(exp)}$ po prawej dla $N_{rep} = 10$ (czerwony), 100 (zielony), 1000 (niebieski).

niowano średnią wariancję oraz średni błąd względny dla wszystkich punktów pomiarowych ($N_{dat} = 426$):

$$\left\langle V\left[\left\langle \sigma_{a,i}^{(art)} \right\rangle_{rep}\right] \right\rangle_{dat} = \frac{1}{N_{dat}} \sum_{i=1}^{N_{dat}} \left(\left\langle \sigma_{a,i}^{(art)} \right\rangle_{rep} - \sigma_i^{(exp)} \right)^2 \,, \tag{4.3a}$$

$$\left\langle PE\left[\left\langle \sigma_{a,i}^{(art)} \right\rangle_{rep}\right] \right\rangle_{dat} = \frac{1}{N_{dat}} \sum_{i=1}^{N_{dat}} \left| \frac{\left\langle \sigma_{a,i}^{(art)} \right\rangle_{rep} - \sigma_i^{(exp)}}{\sigma_i^{(exp)}} \right| \,. \tag{4.3b}$$

Analogicznie możemy zdefiniować $\left\langle V\left[\left\langle \Delta\sigma_{a,i}^{(art)} \right\rangle_{rep}\right] \right\rangle_{dat}$ oraz $\left\langle PE\left[\left\langle \Delta\sigma_{a,i}^{(art)} \right\rangle_{rep}\right] \right\rangle_{dat}$.

Tabela 4.1 przedstawiona różnice między zbiorami danych dla 10, 100 oraz 1000 replik danych. Wariancja wartości średniej zachowuje się zgodnie z przewidywaniami wynikającymi

Tabela 4.1: Porównanie pomiędzy danymi eksperymentalnymi i danymi sztucznie wygenerowanymi

N_{rep}	10	100	1000
$\left\langle V\left[\left\langle \sigma_{a,i}^{(art)}\right\rangle_{rep}\right]\right\rangle_{dat}$	3.6×10^{-3}	4.3×10^{-4}	4.0×10^{-5}
$\left\langle PE\left[\left\langle \sigma_{a,i}^{(art)}\right\rangle_{rep}\right]\right\rangle_{dat}$	0.60%	0.20%	0.06%
$\left\langle V\left[\left\langle \Delta\sigma_{a,i}^{(art)}\right\rangle_{rep}\right]\right\rangle_{dat}$	1.5×10^{-3}	9.4×10^{-5}	1.4×10^{-5}
$\left\langle PE\left[\left\langle \Delta\sigma_{a,i}^{(art)}\right\rangle_{rep}\right]\right\rangle_{dat}$	19.4%	5.8%	1.7%

z teorii Monte Carlo i jest proporcjonalna do $1/N_{rep}$. Podobnie jest w przypadku wariancji odchylenia standardowego, które powinno maleć wraz ze wzrostem N_{rep} proporcjonalnie do $1/\sqrt{N_{rep}}$ [26]. Aby osiągnąć ponad 99% zgodność w wartości średniej oraz około 99% zgodność w niepewności pomiarowej należy wygenerować około 1000 replik danych. Ponadto, każdy z 14 niezależnych zbiorów danych ma określoną procentową niepewność systematyczną $\Delta\eta$, która powinna zostać uwzględniona podczas następnego etapu generowania replik danych. Dla każdego z 14 zbiorów losowana jest jedna wartość $\mathcal{N}(0,\Delta\eta)$ i ostatecznie generowane punkty przyjmują postać

$$\begin{aligned}\sigma_i^{(art)(k)} &= \sigma_{a,i}^{(art)(k)}\times\left(1+\mathcal{N}(0,\Delta\eta)^{(k)}\right)\\ &= \left(\sigma_i^{(exp)}+\mathcal{N}\left(0,\Delta\sigma_i^{(exp)}\right)^{(k)}\right)\times\left(1+\mathcal{N}(0,\Delta\eta)^{(k)}\right).\end{aligned}\tag{4.4}$$

Przedstawiona w dalszej części pracy analiza opiera się na aż trzech różnych zbiorach danych. Oprócz powyższego zbioru dysponujemy zależnością przekroju czynnego od Q^2 i ϵ zmodyfikowaną o poprawkę dwufotonową oraz stosunkiem funkcji postaci w zależności od Q^2. Do tych zbiorów danych zostały zastosowane analogiczne kroki służące wygenerowaniu sztucznych danych.

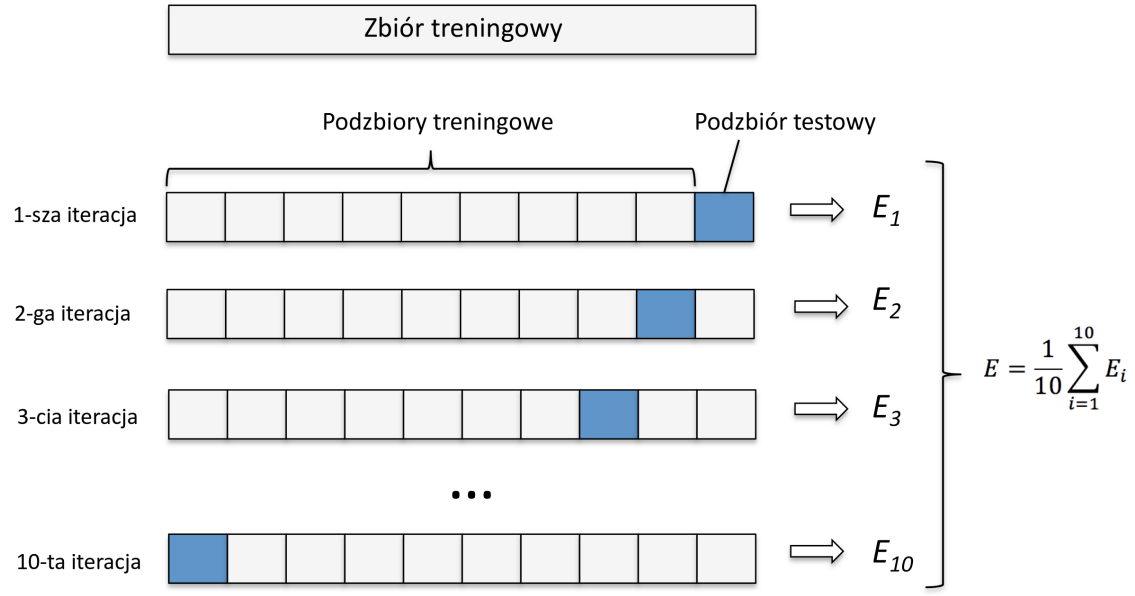
4.3 Walidacja krzyżowa i wczesne zatrzymanie

Algorytm wykorzystywany podczas nauki modelu ma za zadnianie znalezienie takich parametrów, które sprawiają, że model odwzorowuje dane wykorzystane do nauki w sposób jak najlepszy z możliwych. Jeśli do walidacji modelu wykorzystamy inną, niezależną próbkę danych pochodzącą z tego samego zbioru co podzbiór uczący, zazwyczaj okaże się, że model nie działa aż tak dobrze jak przy użyciu zbioru uczącego. Rozmiar tej różnicy zwiększa się,

szczególnie wtedy gdy wielkość zbioru treningowego jest niewielka, lub gdy liczba parametrów modelu jest bardzo duża. Walidacja krzyżowa to metoda statystyczna, która ma za zadanie zminimalizować tę różnicę przez co pomaga ocenić i zwiększyć trafność przewidywań modelu predykcyjnego. Jest to jeden ze sposobów na zachowanie kompromisu między obciążeniem i wariancją, problemu opisanego w rozdziale 3.5.

W najprostszym przykładzie walidacji krzyżowej zbiór danych dzieli się na dwa podzbiory: uczący i walidacyjny. Podczas gdy zbiór uczący służy do nauki modelu, zbiór walidacyjny wykorzystuje się aby mierzyć błąd modelu na nieznanym zbiorze danych.

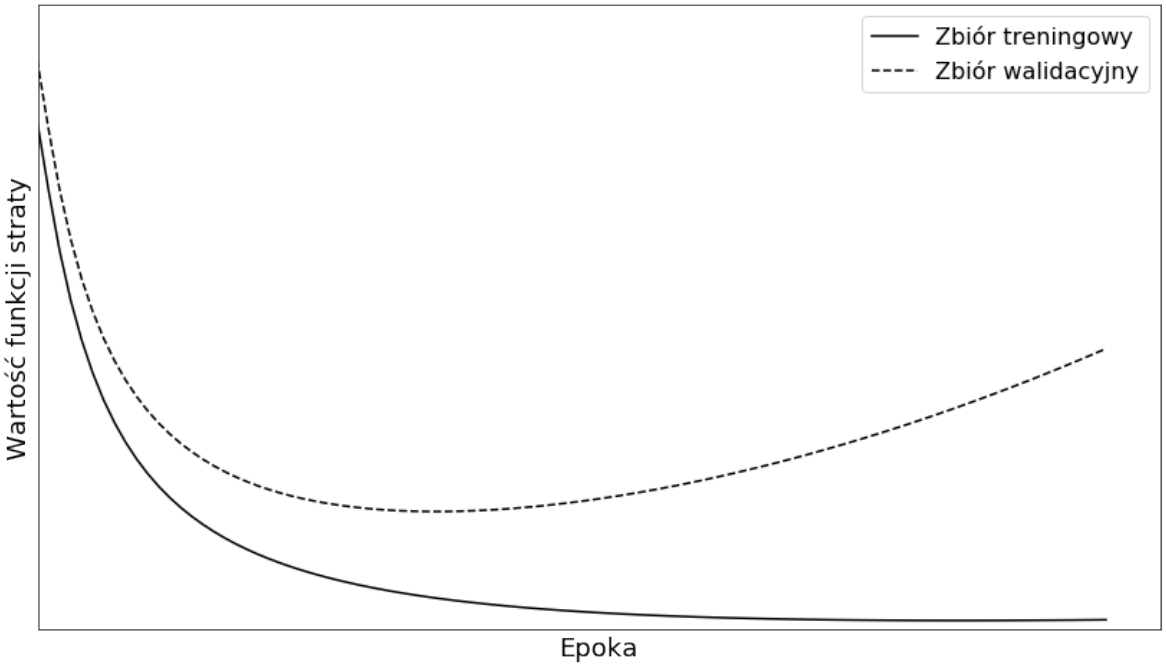
W algorytmie k-krotnej walidacji krzyżowej zbiór danych jest losowo dzielony na k równych wielkością podzbiorów. Jeden z k podzbiorów jest przeznaczany na zbiór walidacyjny, pozostałe $k-1$ podzbiorów służą jako dane teningowe. Powyżej opisana procedura jest powtarzana k razy, a każdy k podzbiorów dokładnie raz zostaje wykorzystany jako zbiór testowy. Następnie k wyników modelu jest uśrednianych dając w rezultacie jeden wynik. Rysunek 4.2 przedstawia sposób działania 10-krotnej walidacji krzyżowej.



Rysunek 4.2: Przykład 10-krotnej walidacji krzyżowej, kolor niebieski oznacza podzbiór testowy, pozostała część zbioru to podzbiór treningowy. E_i to wartości mówiące o wydajności modelu, np. wartości funkcji straty. Wynik końcowy jest średnią z wyników wszystkich iteracji. Rysunek przetłumaczony na język polski z [61].

Cytując [39]: "(...) istnieje pewien kompromis między obciążeniem a wariancją, związany z wyborem parametru k w k-krotnej walidacji krzyżowej. Zazwyczaj stosuje się wartości z przedziału od 5 do 10, ponieważ pokazano empirycznie, że w takim wypadku otrzymujemy przewidywania, które nie cierpią nadmiernie ani z powodu dużego obciążenia ani dużej wariancji." Podczas treningu modelu wybierano więc takie k z zakresu $[5, 10]$, dla którego liczba próbek w zbiorze danych jest całkowicie podzielna przez k co zapewnia równy rozmiar wszystkich zbiorów treningowych i walidacyjnych.

Algorytmy uczenia maszynowego dopasowują parametry modelu na podstawie danych treningowych o skończonym rozmiarze. Podczas procesu szkolenia model jest oceniany na podstawie tego, jak dobrze przewiduje obserwacje zawarte w tym zbiorze. Jednak celem uczenia maszynowego jest stworzenie modelu, który ma zdolność do przewidywania uprzednio niewidzianych obserwacji. Nadmierne dopasowanie to zjawisko pojawiające się wtedy gdy model za bardzo dopasowuje się do danych w zbiorze uczącym co powoduje zmniejszenie wartości błędu na tym zbiorze lecz równocześnie jest przyczyną wzrostu błędu na zbiorze testowym. Nadmierne dopasowanie modelu to problem, który może się pojawiać gdy model zawiera więcej parametrów niż wymagałaby tego natura modelowanego zjawiska. Sieć neuronowa to struktura skłonna do przeuczania. Podczas gdy obserwowany błąd obliczany w oparciu o dane treningowe spada, w pewnym momencie wartość błędu dla zbioru walidacyjnego zaczyna wzrastać. Rysunek 4.3 przedstawia często zamieszczane w literaturze, wyidealizowane krzywe zmiany wartości funkcji straty w czasie, dla zbiorów treningowego i walidacyjnego. Najlepszy model predykcyjny miałby parametry, które odpowiadają momentowi globalnego minimum dla zbioru walidacyjnego.



Rysunek 4.3: Wyidealizowane przykłady krzywych przedstawiających zmianę wartości funkcji straty na zbiorach treningowym i walidacyjnym, podczas nauki modelu

W dziedzinie uczenia maszynowego, metoda wczesnego zatrzymania to forma regularyzacji, która pozwala uniknąć problemu przeuczenia, zatrzymując naukę modelu gdy wartość funkcji straty na zbiorze walidacyjnym zaczyna wzrastać. Rzeczywisty przebieg wartości funkcji straty ma wiele lokalnych minimów, dlatego na podstawie obserwacji krzywych uczenia dokonano wyboru kryteriów zatrzymania nauki modelu. Niech $J_{wa}(t)$ to wartość funkcji straty na zbiorze walidacyjnym po t epokach, $J_{min}(t)$ to dotychczasowe minimum funkcji straty na zbiorze walidacyjnym po t epokach, definiowane jako:

$$J_{min}(t) \equiv \min_{t'<t} J_{wa}(t') \tag{4.5}$$

Niech $J_{sr}(t)$ będzię średnią wartością funkcji straty dla zbioru walidacyjnego z ostatnich 10 epok.

$$J_{sr}(t) \equiv \frac{1}{10} \sum_{i=0}^{10} J_{wa}(t-i) \tag{4.6}$$

Oraz zdefiniujmy pomocniczy parametr $GL(t)$

$$GL(t) \equiv \frac{J_{sr}(t)}{J_{min}} - 1 \tag{4.7}$$

Podczas nauki przedstawionych modeli statystycznych oprócz wykorzystania metody wczesnego zatrzymania została ustalona minimalna wymagana liczba epok. Z powodu startu algorytmu uczącego z losowymi parametrami, szczególnie w pierwszych iteracjach nauki funkcja błędu może być poddana dużym fluktuacjom. Po przekroczeniu minimalnej liczby epok do wczesnego zatrzymania wystarczyło spełnienie jednego z dwóch obowiązujących warunków:

- $J_{min}(t) = J_{min}(t + 200)$ dla wszystkich $t \in [t, t + 200]$, brak zmniejszenia minimalnej wartości funkcji straty dla zbioru walidacyjnego przez 200 epok
- $GL(t) > 2$, względny wzrost średniej wartości funkcji straty przez ostatnie 10 epok względem osiągniętego minimum jest większy niż 200%

Po skończeniu nauki, wybierany jest model, który ma najmniejszą wartość funkcji straty na zbiorze testowym.

4.4 Ilość neuronów

Architektura sieci neuronowej, tzn. ilość warstw ukrytych oraz ilość neuronów w warstwach ukrytych jest zdeterminowana przez wymiar danych wejściowych, rodzaj rozwiązywanego problemu (klasyfikacja czy regresja) oraz relację między zmiennymi objaśniającymi i zmienną objaśnianą.

Uogólniony model liniowy przydatny w szerokim zakresie zastosowań, nie potrzebuję żadnej warstwy ukrytej. Bywa szczególnie przydatny gdy zbiór zawiera mało danych lub są one obarczone dużą niedokładnością. Nawet w przypadku gdy relacja między zmiennymi jest lekko nieliniowa, użycie prostego modelu liniowego może skutkować lepszym uogólnieniem problemu niż skomplikowany model będący wrażliwy na każdy szum znajdujący się w danych. Zgodnie z uniwersalnym twierdzeniem aproksymacyjnym jedna warstwa ukryta z wystarczająco dużą liczbą neuronów wystarcza aby z dowolną dokładnością przybliżyć dowolną ciągłą funkcję [20], [37], [48]. Jeśli zmienna objaśniająca jest jednowymiarowa, wydaje się, że nie odniesiemy żadnej korzyści z skonstruowania sieci neuronowej o więcej niż jednej warstwie ukrytej. Sprawy komplikują się jednak gdy zmienna wejściowa jest dwu lub więcej wymiarowa. Dwuwarstowa sieć neuronowa zachowuje właściwości jednowarstwowej sieci neuronowej oraz osiąga zdolność nauki każdego problemu klasyfikacyjnego [13], ponadto wielowarstwowa sieć neuronowa z dwoma warstwami może skutkować dokładniejszymi wynikami wykorzystując mniejszą

Tabela 4.2: Liczba parametrów sieci neuonowej z dwoma warstwami ukrytymi w zależności od liczby neuronów w warstwach

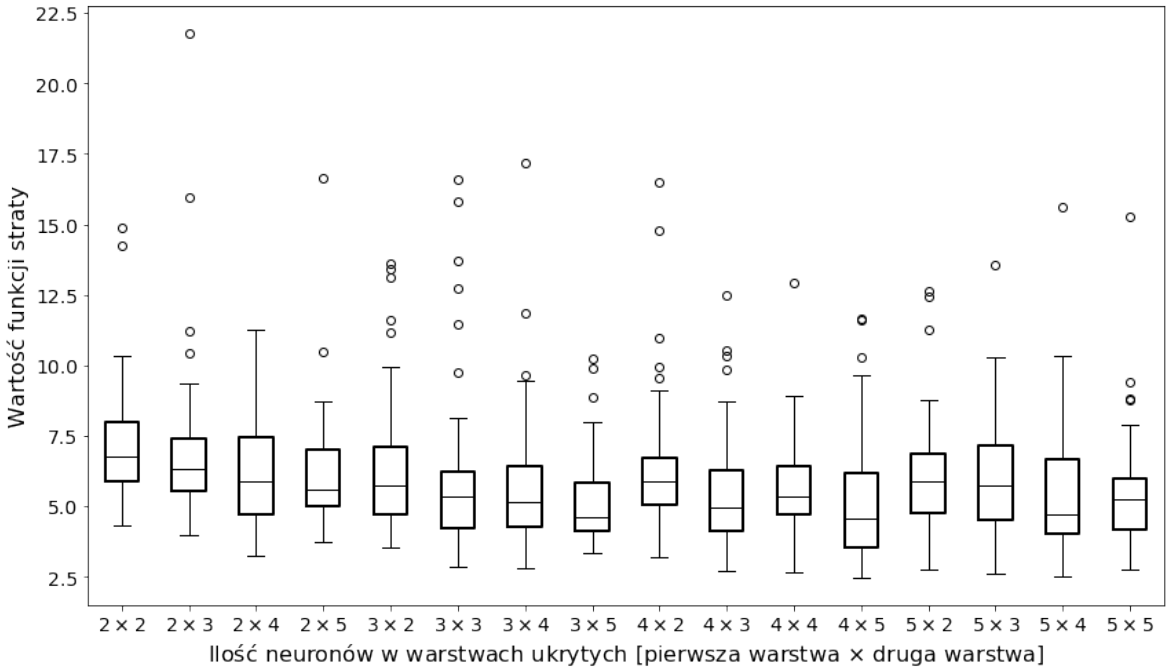
I warstwa \ II warstwa	2	3	4	5
2	14	18	22	26
3	19	24	29	34
4	24	30	36	42
5	29	36	43	50

ilość parametrów niż jednowarstwowa sieć [21]. Na tej podstawie, do rozwiązania problemu regresji gdzie wejściem jest para liczb (ε, Q^2) postanowiłem wybrać sieć neuronową z dwoma warstwami ukrytymi.

Aby znaleźć odpowiednią liczbę neuronów w dwóch warstwach ukrytych, stworzyłem siatkę [2, 3, 4, 5] $\times$ [2, 3, 4, 5] neuronów i sprawdziłem, która konfiguracja daje najmniejszy błąd zbioru walidacyjnego. Dane zostały podzielone na zbiór treningowy i testowy w stosunku 2:1. Dla każdej konfiguracji wytrenowano 50 sieci i sprawdzono jak wygląda statystyka błędu. Tabela 4.2 zawiera porównanie liczby parametrów sieci neuronowej w zależności od liczby neuronów w warstwach ukrytych. Do eksperymentów wybrano konfiguracje charakteryzujące się rozsądną w porównaniu do rozmiaru danych wejsciowych liczbą parametrów. Rysunek 4.4 przedstawia rozkłady minimalnej wartości funkcji straty uzyskanej na danych walidacyjnych uzyskanej z 50 treningów sieci dla każdej konfiguracji ilości neuronów. Wykres pudełkowy to forma graficznej prezentacji rozkładu, która pozwala w łatwy sposób ukazać położenie, rozproszenie oraz kształt empirycznego rozkładu badanej cechy statystycznej. Konfiguracja 3 $\times$ 5 charakteryzuje się najniższą medianą wartości funkcji straty oraz małą liczbą wartości odstających. Ta obserwacja pozwoliła zdecydować, że liczby neuronów będą wynosiły 3 i 5 w odpowiednio pierwszej i drugiej warstwie ukrytej, co za tym idzie sieć będzie miała 36 parametrów.

4.5 Algorytm uczący

Bardzo istotnym elementem tworzonego modelu jest wybór algorytmu poszukującego minimum funkcji straty oznaczonej na potrzeby tego paragrafu jako $J(\theta)$. Na podstawie jego wyników aktualizowane będą parametry tworzonej sieci neuronowej. Bardzo pomocną koncepcją pozwalającą zrozumieć istotę trudności problemu jest powierzchnia błędu. "Każda z N wag i wartości progowych sieci (tzn. wszystkie wolne parametry modelu) traktowana jest jako jeden z wymiarów przestrzeni. W ten sposób każdy stan sieci, wyznaczony przez aktualne wartości jej N parametrów może być traktowany jako punkt na N-wymiarowej hiperpłaszczyźnie. $N + 1$ wymiarem (zaznaczanym jako wysokość ponad wspomnianą wyżej hiperpowierzchnią) jest błąd, jaki popełnia sieć. Dla każdego możliwego zestawu wag i progów może więc zostać narysowany punkt w przestrzeni $N + 1$-wymiarowej, w taki sposób, że stan sieci wynikający z aktualnego zestawu jej parametrów lokuje ten punkt na wspomnianej wyżej N-wymiarowej hiperpłaszczyźnie zaś wartość błędu, jaki popełnia sieć dla tych właśnie wartości parametrów stanowi wysokość umieszczenia punktu ponad tą płaszczyzną. Gdybyśmy opisaną procedurę powtórzyli dla wszystkich możliwych wartości kombinacji wag i progów sieci, wówczas otrzy-



Rysunek 4.4: Wykresy pudełkowe przedstawiające rozkład wartości funkcji straty w zależności od ilości neuronów w pierwszej i drugiej warstwie ukrytej

malibyśmy "chmurę" punktów rozciągających się ponad wszystkimi punktami N-wymiarowej hiperpłaszczyzny parametrów sieci, tworzącą właśnie rozważaną powierzchnię błędu. Celem uczenia sieci jest znalezienie na tej wielowymiarowej powierzchni punktu o najmniejszej wysokości, czyli ustalenie takiego zestawu wag i progów, który odpowiada najmniejszej wartości błędu. Przy stosowaniu modeli liniowych z funkcją błędu opartą na sumie kwadratów powierzchnia błędu ma kształt paraboloidy (funkcji kwadratowej), ma więc kształt kielicha o gładkich powierzchniach bocznych i o jednym wyraźnym minimum. Z tego powodu wyznaczenie w tym przypadku wartości minimalnej nie stwarza większych problemów."[71]

Jeżeli dysponujemy niewielkim zbiorem danych treningowych, do znalezienia optimum funkcji doskonale sprawdzą się metody quasi-Newtonowskie. Ich zaletą jest bardzo szybka zbieżność, niestety obliczenie hesjanu funkcji wielu zmiennych charakteryzuje się dużą złożonością pamięciową $O(n^2)$ i jeszcze większą złożonością obliczeniową $O(n^3)$. Z tego powodu możliwość ich zastosowania ogranicza się do niewielu przypadków. Najbardziej znane algorytmy quasi-Newtonowskie to m.in: *LM-BFGS, Levenberg-Marquardt.* Dysponując dużym zbiorem danych należy wybrać inny algorytm. Po za losowym poszukiwaniem parametrów, najłatwiejszym z nich i bardzo intuicyjnym jest metoda gradientu prostego (*gradient descent*). Parametry θ aktualizowane są w następujący sposób:

$$\theta^{k+1} = \theta^k - \alpha \nabla J(\theta^k) \tag{4.8}$$

gdzie α to wybrany odpowiednio parametr szybkości uczenia (*learning rate*) odpowiedzialny za stopień zmiany parametrów w kolejnych iteracjach. Jeśli θ^0 znajduje się odpowiednio blisko minimum funkcji, i parametr α jest wystarczcająco niewielki, algorytm osiąga liniową

zbieżność [22]. W ogólności metoda gradientu prostego gwarantuje zbieżność do globalnego minimum w przypadku funkcji błędu o wypukłej powierzchni i do lokalnego minimum dla funkcji błędu o powierzchni nie wypukłej. Algorytm jednak jest bardzo wolny, co jest jego największą słabością. Ze względu na częstość aktualizacji wag, metodę gradientu prostego możemy podzielić na *batch gradient descent* oraz *astic gradient descent*. W pierwszym przypadku wagi są dostosowywane po przetworzeniu pełnego zbioru danych, w metodzie stochastycznej zbiór uczący dzielony jest na podzbiory a wagi aktualizowane są po przetworzeniu każdego z podzbiorów. Druga metoda jest szczególnie użyteczna dla dużych zbiorów danych. Spodziewamy się, że dla dobrze przygotowanych danych kierunek podążania wartości wag będzie podobny jeśli policzymy gradient zarówno dla 10% jak i dla 100% zbioru treningowego.

Wyobraźmy sobie, że poszukiwanie minimum powierzchni błędu to przemierzanie przestrzeni pełnej dolin, pagórków, wąwozów. W kolejnych iteracjach przeskakujemy między tymi obszarami, w pewnym momencie może się zdarzyć, że gradient zaniknie lub będzie bardzo słaby a nasze poszukiwania zatrzymają się nie osiągając wystarczającego minimum. Idea pędu inspirowana zjawiskami fizycznymi to nadanie gradientowi krótkotrwałej pamięci. Posługując się kolejną analogią, popchnięta w dół piłka nabierając prędkości zwiększa swój pęd. To samo dzieje się z parametrami sieci, wartość pędu wzrasta dla wymiarów, których gradienty wskazują te same kierunki i zmniejsza modyfikacje wartości dla wymiarów, w których gradienty zmieniają kierunki. W rezultacje otrzymujemy szybszą zbieżność i mniejsze oscylacje.

$$v^{k+1} = \beta v^k + \nabla J(\theta^k) \tag{4.9}$$

$$\theta^{k+1} = \theta^k - \alpha v^{k+1} \tag{4.10}$$

Zmiana jest niewielka, gdy $\beta = 0$, otrzymujemy zwykłą metodę gradientu prostego, zazwyczaj jednak ustala się wartość parametru β, zwanego pędem na około 0.9 [67].

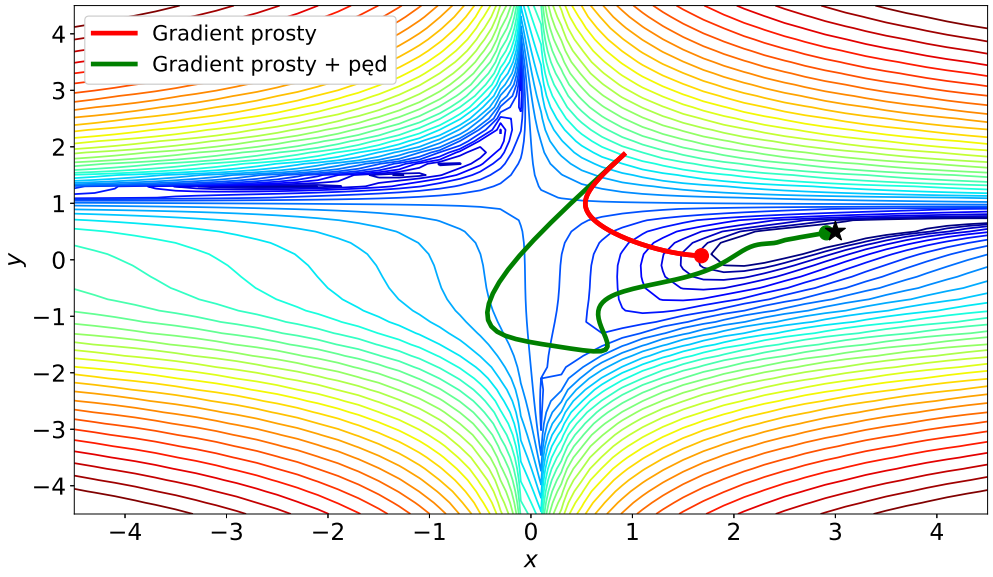
Porównanie efektywności przedstawionych wyżej algorytmów znajduje się na rysunku 4.5, w zaprezentowanym przykładzie metoda gradientu prostego potrzebuje około 10 razy więcej iteracji od modyfikacji z pędem aby dotrzeć do minimum zaprezentowanej funkcji. Jest to przykład świadczący o tym jak duży wpływ na szybkość działania algorytmu wywiera ta niewieka modyfikacja.

Wykorzystany podczas treningu modelu algorytm korzysta jednak z jeszcze z jednej modyfikacji. Nie chcielibyśmy aby piłka spuszczona w dół ślepo podążała za zboczem widząc, że za niedługo mocno się ono podniesie. Przyspieszenie Nesterova (NAG) jest sposobem na uwzględnienie podczas obliczania gradientu przybliżonej przyszłej pozycji parametrów sieci. Algorytm opisują równania (4.11) i (4.12) [66].

$$v^{k+1} = \beta v^k + \nabla J(\theta^k - \beta v^k) \tag{4.11}$$

$$\theta^{k+1} = \theta^k - \alpha v^{k+1} \tag{4.12}$$

Niezywkle istotnym parametrem algorytmu jest α, jego niezmienność wraz z postępem iteracji powoduje bardzo niską efektywność algorytmu. Ze względu na metodę zmiany tego parametru, który może być indywidualnie ustalany dla każdej wagi powstało wiele szeroko wykorzystywanych algorytmów. Do najpopularniejszych należą między innymi *Adam, Nadam, Adagrad, Adadelta, AMSGrad, RMSprop.*

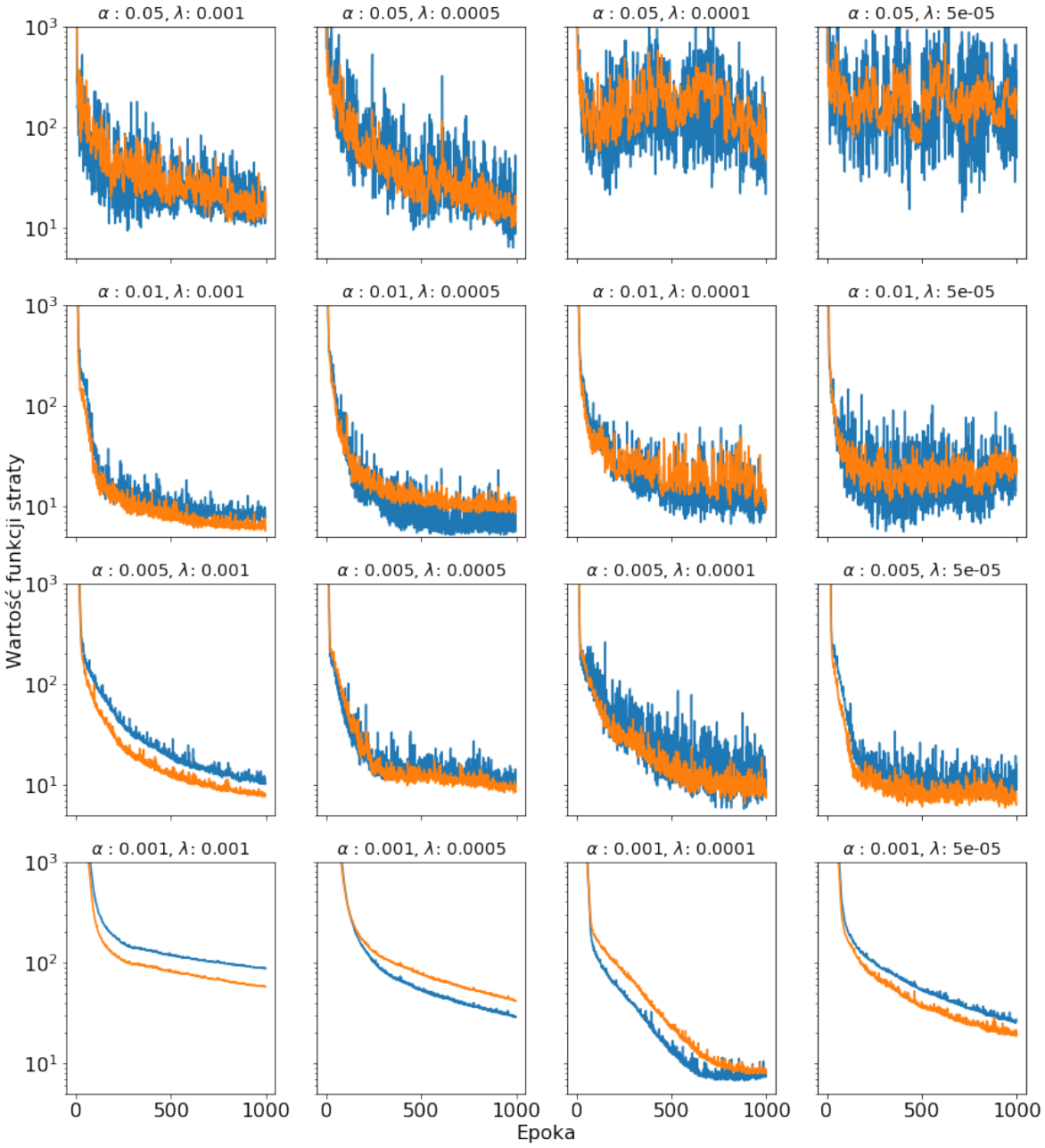


Rysunek 4.5: Funkcja $f(x, y) = (1.5 - x + xy)^2 + (2.25 - x + xy^2)^2 + (2.625 - x + xy^3)^2$, osiąga minimum równe 0, w punkcie $(3, 0.5)$ oznaczonym czarną gwiazdą. Grafika przedstawia porównanie działania metody gradientu prostego oraz jego modyfikacji poprzez dodanie pędu. Przyjmując, że punkt początkowy to $(2, 1)$, $\alpha = 0.001$ i $\beta = 0.9$, możemy prześledzić trajektorie algorytmów przez pierwsze 500 iteracji działania.

W swoim algorytmie postanowiłem dokonywać zmiany parametru α wraz ze wzrostem iteracji. Ponadto szybkość uczenia zależna jest od wybranego parametru λ decydującego o tym z jaką szybkością maleje.

$$\alpha(i) = \alpha_0 \times \frac{1}{1 + \lambda \times i} \tag{4.13}$$

Rysunek 4.6 przedstawia porównanie przykładowych krzywych zmian wartości funkcji straty w czasie dla różnych wartości α i λ. Na ich podstawie widać jak duży wpływ wnosi parametr α w proces nauki modelu. Zbyt duża szybkość uczenia powoduje bardzo duże oscylacje krzywej funkcji straty, za mała wartość α bardzo mocno spowalnia proces nauki. Pewien kompromis przynosi wybranie odpowiednio dużej początkowej wartości szybkości uczenia, co przynosi szybkie przejście algorytmu w obszar minimum i następnie zmniejszenie go do wartości potrafiącej efektywnie dalej poszukiwać optimum. Zadowalający przebieg mają krzywe o parametrach $\alpha = 0.005$, $\lambda = 0.001$, które przedstawiają porządany, eksponencjalny kształt o niewielkiej oscylacji. Na podstawie powyższej analizy to właśnie te hiperparametry zostały wykorzystane w modelu, dodatkowo parametr pędu β został ustalony na wartość 0.9



Rysunek 4.6: Porównanie przykładowych krzywych zmiany wartości funkcji straty w czasie dla zbiorów treningowego (kolor pomarańczowy) i walidacyjnego (kolor niebieski) ze względu na parametry α (*learning rate*) oraz λ (*decay*)

Rozdział 5

Wyniki analizy

5.1 Analiza nr 1

Celem pierwszej analizy jest modelowanie elektrycznego i magnetycznego czynnika postaci przy wykorzystaniu wyłącznie danych przekrojów czynnych rozpraszania elektron-proton.

Dane wejściowe i funkcja straty

Na zbiór analizowanych danych składa się 24 niezależnych zbiorów danych z eksperymentów, w których dokonywano rozpraszania elektron-proton, razem daje to 426 punktów pomiarowych. Zestaw danych składa się z 4 kolumn, które zawierają kolejno zmienną objaśnianą σ - przekrój czynny, niepewność pomiaru zmiennej objaśnianej $\Delta\sigma$ oraz dwie zmienne objaśniające Q^2 - kwadrat przekazanego czteropędu i czynnik kinematyczny ϵ. Ponadto, każdy z niezależnych zbiorów ma określoną niepewność systematyczną $\Delta\eta$. Dodatkowo do każdego ze zbiorów dodano sztuczny punkt pomiarowy, który korzysta z założenia, że $\sigma\left(Q^2 = 0 \text{ GeV}^2, \epsilon = 1\right) = 1$, niepewność pomiarowa punktu wynosi $\Delta\sigma = 0.01$, zwiększa to liczbę wszystkich punktów pomiarowych do 450. Funkcja straty to z definicji funkcja przyporządkowująca nieujemną wielkość kary poprzez porównanie zmiennej objaśnianej do wyliczonego estymatora. W przedstawionym modelu, wykorzystana została zmodyfikowana postać funkcji chi-kwadrat, która bierze pod uwagę zarówno nipewność pomiarową oraz systematyczną

$$\chi^2 = \frac{1}{n}\chi^2_\sigma \tag{5.1}$$

$$\chi^2_\sigma = \sum_{k=1}^{N_\sigma}\left[\sum_{i=1}^{n_k}\left(\frac{\eta_k\sigma^{th}_{ki} - \sigma^{ex}_{ki}}{\Delta\sigma_{ki}}\right)^2 + \left(\frac{\eta_k - 1}{\Delta\eta_k}\right)^2\right], \tag{5.2}$$

gdzie N_σ to liczba zbiorów danych z niezależnych eksperymentów, n_k to liczba punktów w k-tym zbiorze danych, $n = \sum_{k=1}^{N_\sigma} n_k$ to liczba wszystkich punktów pomiarowych, η_k to parametr normalizacyjny dla k-tego zbioru danych, $\Delta\eta_k$ to błąd systematyczny. σ^{ex}_{ki} to wartość eksperymentalna przekroju czynnego i-tego pomiaru z k-tego zbioru danych, zmierzona dla określonych par Q^2_{ki}, ϵ_{ki}. $\Delta\sigma^{ex}_{ki}$ oznacza odpowiadającą niepewność pomiaru, σ^{th}_{ki} to przewidywanie modelu statystycznego. η_k, $k = 1, 2, \ldots, N_\sigma$ to parametry normalizacyjne. Ich wartosci

Tabela 5.1: Hiperparametry modelu

Kategoria	Parametr	Wartość
Generowanie danych	N_{rep}	500
k-krotna walidacja krzyżowa	k	5
Algorytm uczący	α (*learning rate*)	0,003
	λ (*decay*)	0,0005
	β (*pęd*)	0,9
Sieć neuronowa	Liczba warstw	2
	Ilość neuronów	(3,5)

są aktualizowane podczas każdej itereacji nauki modelu [34], powinny one spełniać warunek

$$\frac{\partial \chi^2_\sigma}{\partial \eta_k} = 0,\ k = 1, \ldots, N_\sigma, \tag{5.3}$$

co można zapisać jako

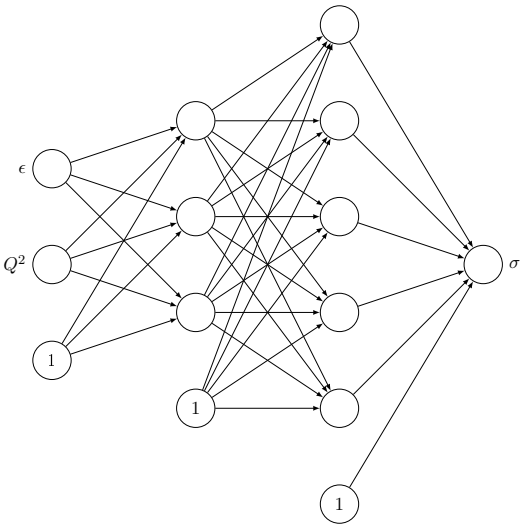
$$\eta_k = \frac{\sum_{i=1}^{n_k} \frac{\sigma^{th}_{ki}\sigma^{ex}_{ki}}{(\Delta\sigma_{ki})^2} + \frac{1}{(\Delta\eta_k)^2}}{\sum_{i=1}^{n_k} \frac{\left(\sigma^{th}_{ki}\right)^2}{(\Delta\sigma_{ki})^2} + \frac{1}{(\Delta\eta_k)^2}}. \tag{5.4}$$

Parametry i nauka sieci

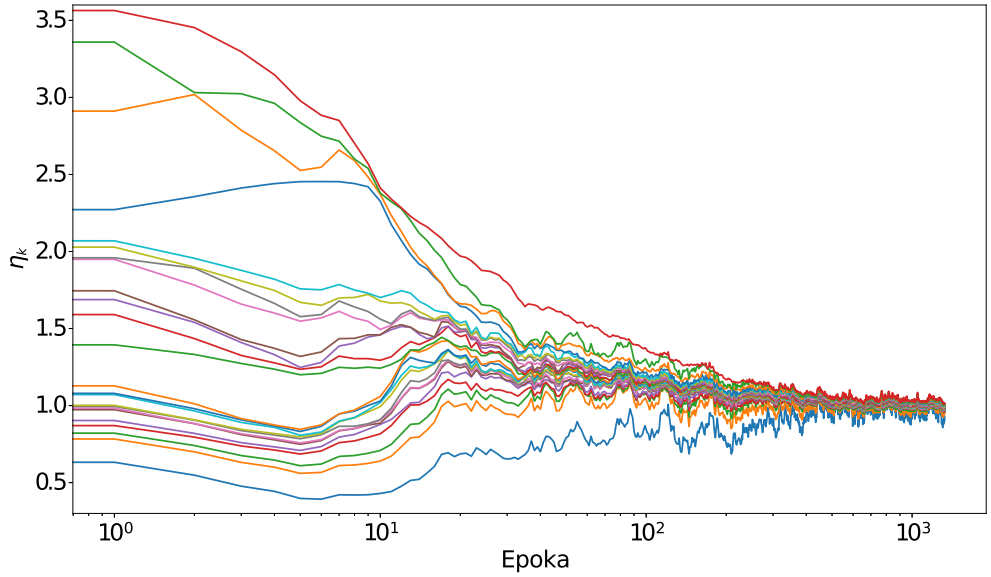
Sieć neuronowa o dwóch wartstwach ukrytych i hiparametrach przedstawionych w Tabeli 5.1 daje w rezultacie wartość σ w zależności od zmiennych Q^2 oraz ϵ. Podczas treningu elementy ze zbioru parametrów η_k dążą do wartości bliskich 1. Rysunek 5.2 przedstawia ewolucje parametrów η_k wraz z nauką sieci dla każdego z 24 niezależnych zbiorów danych. Łącznie zostało wytrenowanych 2500 ($N_{rep} \times k$) modeli. Przykładowy przebieg wartości funkcji straty dla zbioru treningowego i walidacyjnego został przedstawiony na Rysunku 5.3, możemy zauważyć, że około 1100 epoki funkcja straty osiąga minimum, zgodnie z opisanymi wcześniej zasadami wczesnego zatrzymania model kończy naukę po następnych 200 epokach.

Wyniki

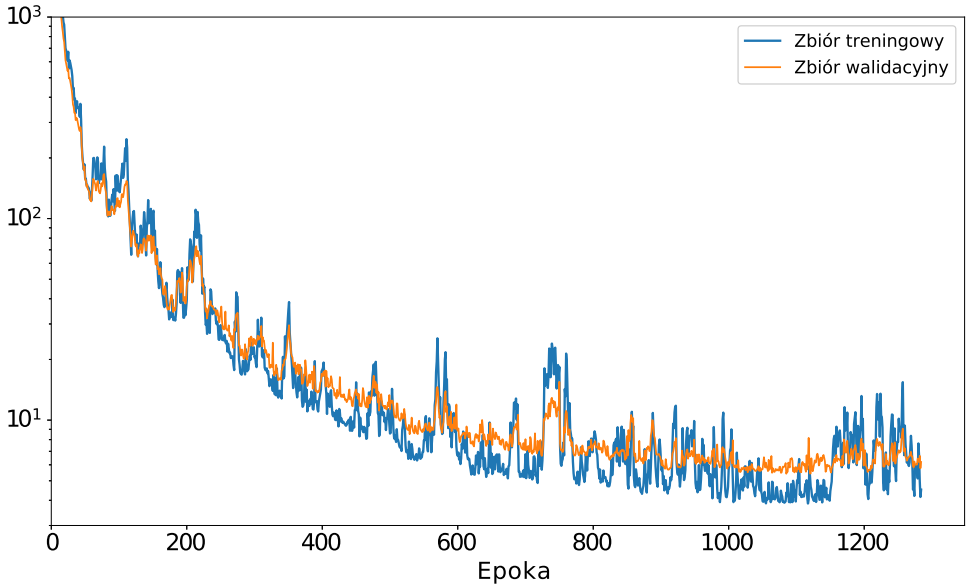
Wyniki wytrenowanych modeli tworzą statystykę, której najważniejszymi parametrami są średnia przedstawiana jako wynik analizy, oraz odchylenie standardowe, które wyznacza zakres bardzo prawdopodobnych wyników, na wykresie przedstawiany jako zacieniowany obszar. Rysunek 5.5 przedstawia zależność przekroju czynnego $\sigma(\epsilon)$, dla kilku ustalonych wartości Q^2. Otrzymane funkcje mają przebieg liniowy o bardzo podobnym współczynniku kierunkowym a błąd modelu 1σ znacznie wzrasta wraz ze wzrostem Q^2. Zależność przekroju czynnego $\sigma(Q^2)$ przy ustalonym parametrze ϵ znajduje się na rysnku 5.5. Możemy zauważyć, że im niższa wartość ϵ tym mniejszy przekrój czynny dla $Q^2 = 0$, następnie krzywe mają bardzo podobny przebieg, niezależnie od ustalonego parametru ϵ zbiegają do tej samej maksymalnej wartości σ wraz ze wzrostem Q^2. Ponadto, wraz ze wzrostem Q^2 rośnie niepewność otrzymanego wyniku, dla $Q^2 = 10$ GeV^2 obszar błędu modelu 1σ wynosi aż ± 1. Po zbadaniu podstawowych zależności estymowanej funkcji od kwadratu przekazu czteropędu oraz czynnika kinematycznego



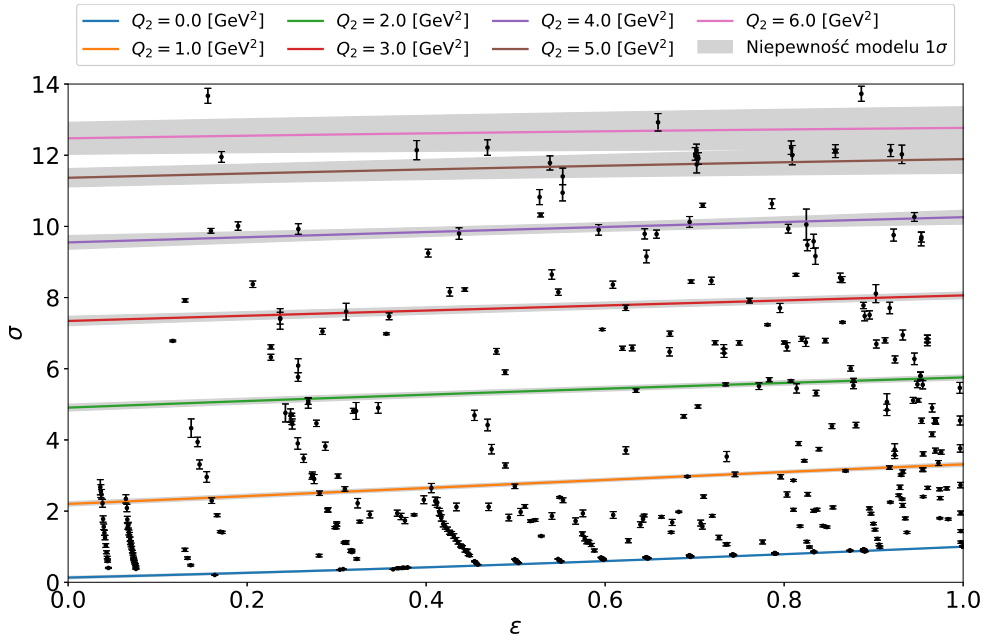
Rysunek 5.1: Schemat sieci neuronowej zastosowanej w pierwszej analizie, która składa się z: i) warstwy wejściowej z dwoma neuronami, ii) dwóch warstw ukrytych z odpowiednio trzema i pięcioma neuronami, iii) warstwy wyjściowej z jednym neuronem. Linie zakończone strzałką oznaczają wagi odpowiadające każdej z par neuronów



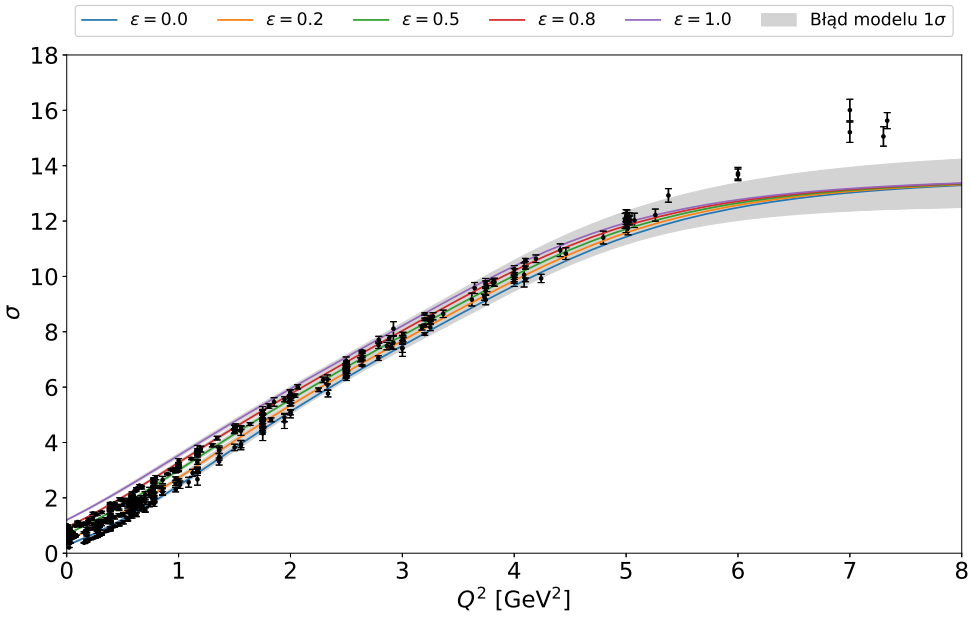
Rysunek 5.2: Ewolucja parametrów η_k podczas jednego z treningów modelu. W ciągu kolejnych epok, wartości parametrów ustalane na podstawie równania 5.4 zbiegają do wartości bliskich 1.



Rysunek 5.3: Zmiana wartości funkcji straty podczas nauki modelu. Wartość funkcji obliczana na podstawie zbioru treningowego oznaczona jest kolorem niebieskim, dla zbioru walidacyjnego - pomarańczowym.



Rysunek 5.4: Zależność przekroju czynnego σ od czynnika kinematycznego ϵ przy ustalonym przekazie czteropędu Q^2. Linia ciągła wyznacza średnią wartość po wszystkich wytrenowanych sieciach. Kolor szary wyznacza obszar niepewności 1σ.



Rysunek 5.5: Zależność przekroju czynnego σ od przekazu czteropędu Q^2 przy ustalonym czynniku kinematycznym ϵ. Linia ciągła wyznacza średnią wartość po wszystkich wytrenowanych sieciach. Kolor szary wyznacza obszar niepewności 1σ.

wyznaczono elektryczny i magnetyczny funkcji postaci. Wiemy, że

$$\sigma_R(\epsilon, Q^2) = \tau G_{M_p}^2(Q^2) + \epsilon G_{E_p}^2(Q^2), \tag{5.5}$$

obliczając pochodną po parametrze ϵ otrzymamy kwadrat elektrycznej funkcji postaci protonu, więc

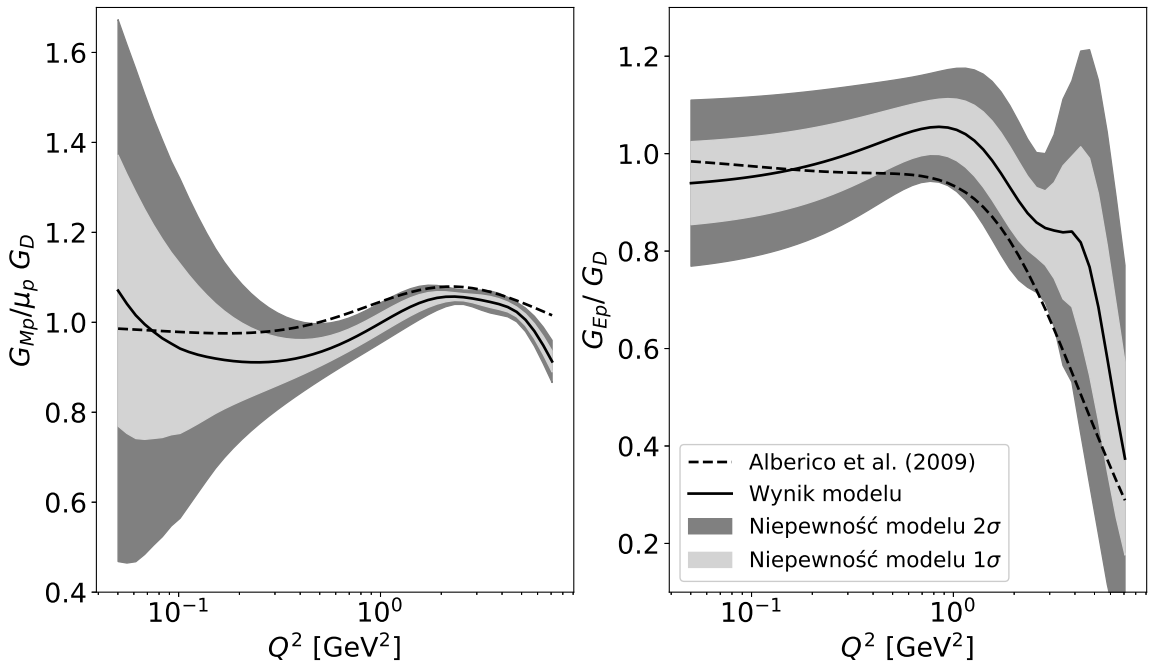
$$G_{E_p}(Q^2) = \sqrt{\frac{\partial \sigma_R(\epsilon, Q^2)}{\partial \epsilon}}. \tag{5.6}$$

Następnie magnetyczna funkcja postaci protonu wyraża się wzorem

$$G_{M_p}(Q^2) = \sqrt{\frac{\sigma_R(\epsilon, Q^2) - \epsilon \frac{\partial \sigma_R(\epsilon, Q^2)}{\partial \epsilon}}{\tau}} \tag{5.7}$$

Należy pamiętać, że pochodna $\frac{\partial \sigma_R(\epsilon, Q^2)}{\partial \epsilon}$ musi zostać wyznaczona dla ustalonej wartości czynnika kinematycznego ϵ. Ponieważ przyjmuje on wartość z przedziału $[0, 1]$, pochodną wyznaczano dla ϵ będących 11 pierwszymi wyrazami ciągu arytmetycznego o przepisie: $\epsilon_n = (n-1) \times 0.1$. Powtarzając te procedurę dla każdego wytrenowanego modelu otrzymujemy 2500×11 wyników, ich średnia tworzy wynik modelu. Wykresy 5.6 przedstawiają obliczone w ten sposób funkcje postaci protonu. Opisane działanie jest przyczyną niepewności otrzymanego wyniku, która ma dwa źródła. Pierwsze z nich to błąd pochodnej, czyli rozbieżność wyników w zależności od wybranego parametru ϵ podczas liczenia pochodnej. Drugie z nich spowodowane jest liczbą wytrenowanych modeli, z których każdy przyjmuje nieco inne dane wejściowe, tworzy to

dużą statystykę wyników pochodzących ze wszystkich nauczonych modeli. Otrzymane funkcje postaci zostały porównane z przewidywaniami z publikacji [2] i ich graficzne przedstawienie znajduje się na rysunku 5.6. Wartość funkcji $G_{E_p}\left(Q^2 = 0\ \mathrm{GeV}^2\right)$ powinna wynosić 1, mimo że wynik modelu jest niedoszacowany to wartość 1 znajduje się w obszarze niewielkiego błędu. Niestety wraz ze wzrostem Q^2 wzrasta rozbieżność między porównywanymi funkcjami, dla $Q^2 \simeq 2.0\ \mathrm{GeV}^2$ obserwujemy nienaturalną zmianę wypukłości funkcji, której się niespodziewamy, i która nie występuje w wyniku z [2]. Jest to jedyny obszar, w którym porównywana funkcja znajduje się po za obszarem niepewności modelu 2σ. Przebieg magnetycznej funkcji postaci G_{M_p} charakteryzuje się bardzo dużą niepewnością w rejonach niskiej wartości Q^2, mimo to średni wynik jest bliski pożądanej wartości bliskiej 1. Powyżej wartości $Q^2 \simeq 0.2\ \mathrm{GeV}^2$ niepewność pomiaru znacznie się zmniejsza, porównywany rezultat znajduje się na granicy niepewności 2σ różniąc się od wyniku modelu o stałą wartość około 0.05, modele zaczynają się znacznie różnić dopiero dla wartości Q^2 przekraczających $10\ \mathrm{GeV}^2$. Całkowity rezultat należy jednak ocenic pozytywnie, porównywana funkcja rzadko znajduje się po za obszarem niepewności modelu 2σ.



Rysunek 5.6: Elektryczna (a) i magnetyczna (b) funkcja postaci. Linia ciągła przedstawia średnią ze wszystkich modeli, zacieniowane regiony wyznaczają obszary 1σ powstałe z dwóch różnych przyczyn. Jasnoszary obszar opisany jako błąd modelu to odchylenie standardowe opisujące rozkład wyników wszystkich wytrenowanych modeli. Ciemnoszary obszar wyznacza odchylenie standardowe opisujące rozkład wyników powstałych na skutek obliczeń pochodnej dla różnych wartości ϵ z zakresu $[0, 1]$. Przerywana linia to wyniki przedstawione w [2].

5.2 Analiza nr 2

Dane wejściowe i funkcja straty

Dane wejściowe w następnej analizie to 450 punktów pomiarowych analizowanych w pierwszej analizie powiększone o zbiór 68 pomiarów stosunku funkcji postaci G_{E_p}/G_{M_p} wraz z niepewnością pomiarową w zależności od kwadratu przekazu czteropędu Q^2. Ponieważ znamy jeden z więzów stosunku funkcji postaci, do zbioru dodany został 69 i 70 punkt: $\mathcal{R}\left(Q^2 = 0 \text{ GeV}^2\right) = 1$, $\Delta\mathcal{R} = 0,001$. Razem otrzymujemy 520 pomiarów co sugeruje wybranie parametru k-krotnej walidacji jako $k = 5$. Wykorzystana podczas nauki drugiego modelu funkcja straty χ^2 (5.8) jest modyfikacją funkcji wykorzystanej w pierwszej analizie. Do użytej wcześniej funkcji błędu dodany został składnik (5.9) uwzględniający błąd estymacji stosunku $\mathcal{R} = G_{E_p}/G_{M_p}$ oraz składniki dbające o zachowanie więzów $G_{E_p}\left(Q^2 = 0 \text{ GeV}^2\right) = 1$, $G_{M_p}\left(Q^2 = 0 \text{ GeV}^2\right) = 1$ - równanie (5.10).

$$\chi^2 = \frac{1}{n}\left[\chi^2_\sigma + \chi^2_{PT} + \chi^2_{G_M} + \chi^2_{G_E}\right], \tag{5.8}$$

$$\chi^2_{PT} = \sum_{i=1}^{n_k^{PT}} \left(\frac{\mathcal{R}_i^{th} - \mathcal{R}_i^{ex}}{\Delta\mathcal{R}_i}\right)^2, \tag{5.9}$$

$$\chi^2_G = \left(\frac{G^{th} - 1}{\Delta G}\right)^2, \tag{5.10}$$

gdzie n_k^{PT} to liczba pomiarów stosunków funkcji postaci, $\mathcal{R}_i^{ex}$ to i-ty pomiar zmierzony dla odpowiadającej wartości Q_i^2, z niepewnością pomiaru $\Delta\mathcal{R}_i$ i $\mathcal{R}_i^{th}$ to estymowana przez model wartość. $\chi^2_G = \chi^2_{G_{E_p}}$ lub $\chi^2_{G_{M_p}}$, natomiast $G^{th} = {G_{E_p}}^{th}\left(Q^2 = 0 \text{ GeV}^2\right)$ lub ${G_{M_p}}^{th}\left(Q^2 = 0 \text{ GeV}^2\right)$ oraz $\Delta G = 0,001$.

Parametry i nauka sieci

Wynikiem działania sieci neuronowej o dwóch wartstwach ukrytych i hiparametrach przedstawionych w Tabeli 5.2 są funkcje postaci G_{Mp} oraz G_{Ep} w zależności od zmiennej Q^2. Rysunek 5.7 przedstawia schemat wykorzystanej sieci. Podczas treningu elementy ze zbioru parametrów η_k aktualizowane są podczas każdej epoki zgodnie z wzorem 5.4 i ponownie dążą do wartości bliskich 1, przykładowe wartości będące wynikiem nauki modelu znajdują się w Tabeli 5.3. Łącznie zostało wytrenowanych 2500 ($N_{rep} \times k$) modeli. Dane wejściowe służące do treningu sieci zostały znormalizowane, wykorzystano do tego standaryzację polegającą na wyzerowaniu średniej oraz skalowaniu dającym w rezultacie jednostkową wariancję zgodnie ze wzorem 5.11

$$x' = \frac{x - \bar{x}}{\sigma}, \tag{5.11}$$

gdzie $\bar{x}$ to średnia cechy x, a σ to odchylenie standardowe cechy x. Wartości początkowe wag odpowiadających poszczególnym parom neuronów zostały zainicjalizowane przy użyciu rozkładu jednostajnego z przedziału $[-0,6; 0,6]$. Zbyt duże lub zbyt niskie wagi skutkują znalezieniem się na płaskich krańcach sigmoidy. Prowadzi to do niewielkiego gradientu co skutecznie spowolnia lub nawet uniemożliwia trening sieci.

Rysunek 5.7: Schemat sieci neuronowej zastosowanej w drugiej analizie, która składa się z: i) warstwy wejściowej z dwoma neuronami, ii) dwóch warstw ukrytych z odpowiednio trzema i pięcioma neuronami, iii) warstwy wyjściowej z dwoma neuronami. Linie zakończone strzałką oznaczają wagę odpowiadającą każdej z par neuronów

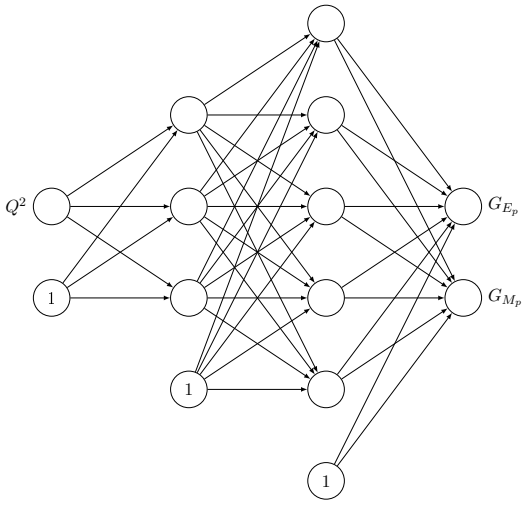


Tabela 5.2: Hiperparametry modelu podczas drugiej analizy

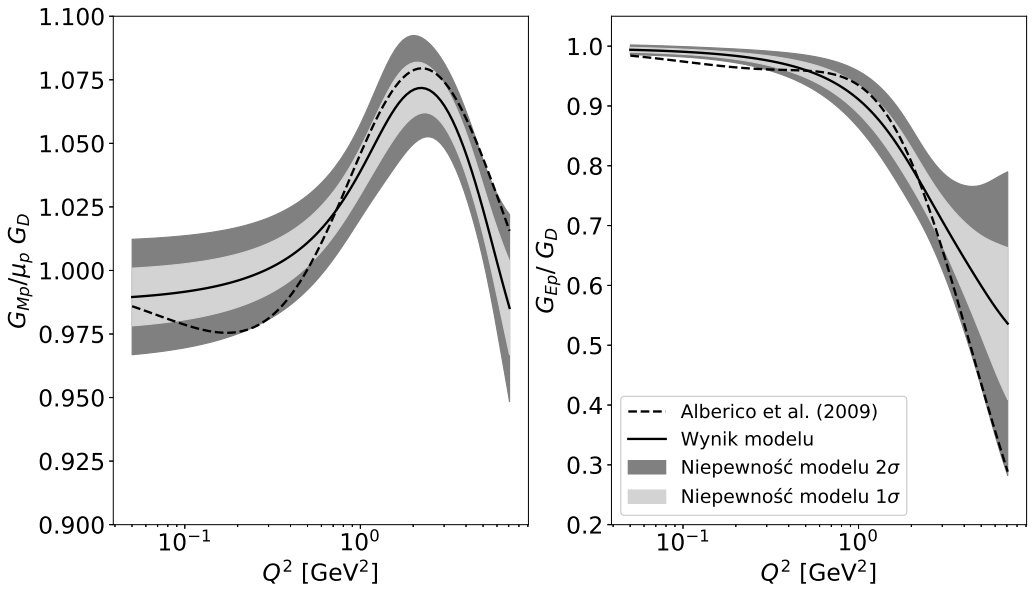
Kategoria	Parametr	Wartość
Generowanie danych	N_{rep}	500
k-krotna walidacja krzyżowa	k	5
Algorytm uczący	α (*learning rate*)	0,02
	λ (*decay*)	0,001
	β (*pęd*)	0,9
Sieć neuronowa	Liczba warstw	2
	Ilość neuronów	(3,5)

Tabela 5.3: Przykładowe końcowe wyniki parametrów η_k podczas nauki modelu w drugiej analizie

L.p	Zbiór danaych	$\Delta\eta_k[\%]$	η_k
1	And94_000.dat	1,77	1,014
2	And94_100.dat	2,70	1,018
3	Arn86_1500.dat	3,00	1,017
4	Bar66_1700.dat	2,50	0,981
5	Bar73_200.dat	2,10	0,971
6	Bar73_300.dat	2,10	0,998
7	Bar73_400.dat	2,10	0,980
8	Ber71_700.dat	4,00	0,947
9	Bor74_1900.dat	2,00	0,971
10	Chr03_500.dat	1,50	0,959
11	Dut03_1400.dat	1,90	0,989
12	Goi70_1800.dat	3,80	0,936
13	Jan66_1600.dat	1,60	0,965
14	Kir73_1100.dat	4,00	1,050
15	Lit67_600.dat	4,00	0,941
16	Mur74_2200.dat	4,60	1,016
17	Nic99_1300.dat	1,90	1,025
18	Pri71_900.dat	1,90	1,009
19	Qat05_2300.dat	3,00	0,981
20	Sil93_1000.dat	3,00	0,997
21	Sim80_2100.dat	0,50	1,017
22	Sim81_2000.dat	0,50	0,997
23	Ste75_800.dat	2,40	0,983
24	Wal94_1200.dat	1,90	0,964

Wyniki

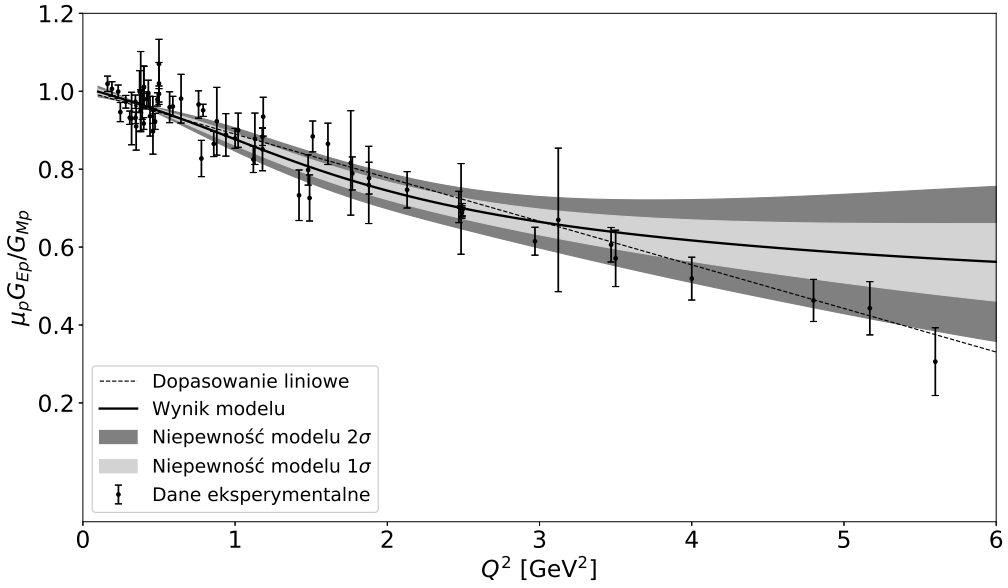
Wyniki nauczonych modeli tworzą statystykę, której najważniejszymi parametrami są średnia przedstawiana jako wynik modelu, oraz odchylenie standardowe, które wyznacza zakres bardzo prawdopodobnych wyników, na wykresie przedstawiany jako zacieniowany obszar. Rysunek 5.8 przedstawia wykresy otrzymanych funkcji postaci protonu. Lewa część obrazka przedstawia magnetyczną funkcję postaci protonu w zależności od kwadratu przekazu czteropędu $G_{M_p}(Q^2)$, prawa część rysunku przedstawia przebieg elektrycznej funkcji postaci $G_{E_p}(Q^2)$. Otrzymane wyniki są znacznie bliższe przebiegom funkcji z publikacji [2] i prawie całkowicie mieszczą się w zaznaczonych ciemnym kolorem obszarach $\pm 2\sigma$. Wariancja magnetycznej funkcji postaci jest w przybliżeniu stały na całej długości przedstawionego przebiegu funkcji i wynosi około 0.015, odchylenie standardowe dla elektrycznej funkcji postaci w pobliżu $Q^2 \to 0$ jest bardzo niewielkie i znacznie wzrasta wraz z Q^2 osiągając wysoką wartość 0.1. Może to być spowodowane mniejszą ilością pomiarów doświadczalnych dla dużych wartości Q^2. Najbardziej zauważalna różnica pomiędzy funkcjami występuje dla wartości $Q^2 \simeq 0.15$ GeV^2,



Rysunek 5.8: Magnetyczna i elektryczna funkcja postaci. Linia ciągła oznacza średnią ze wszystkich modeli, zacienione pola wyznaczają obszary 1σ (jasny) oraz 2σ (ciemny). Linia przerywana przedstawia wyniki z publikacji [2].

wynik modelu zarówno dla G_{M_p} jak i dla G_{E_p} jest przeszacowany względem przerywanej krzywej. Porównywana elektryczna funkcja postaci w tym obszarze nieznacznie wychodzi po za pokazany obszar niepewności modelu. Dodatkowym źródłem danych służącym do nauki sieci neuronowej w stosunku do poprzedniego modelu były dane zawierające eksperymentalne pomiary stosunków elektrycznej i magnetycznej funkcji postaci w zależności do kwadratu przekazu czteropędu. Rysunek 5.9 przedstawia wspomniane powyżej pomiary wraz z zaznaczoną ich niepewnością. Ponadto przerywaną linią zaznaczono dopasowanie liniowe, które dobrze opisuje

relację pomiędzy funkcjami postaci oraz wynik modelu wraz z odchyleniami standardowymi. Iloraz $\mu G_{E_p}/G_{M_p}$ szczególnie dobrze opisywany jest przez model dla niewielkich wartości Q^2 i cechuje się w tym obszarze bardzo niewieką niepewnością. Wraz ze wzrostem Q^2 rezultat modelu jest coraz mniej liniowy, G_{E_p} staje się przeszacowane, a odchylenie standardowe znacznie wzrasta. Większość pomiarów doświadczalnych pozostaje jednak w obszarze niepewności -2σ.



Rysunek 5.9: Pomiary stosunku elektrycznej i magnetycznej funkcji postaci wraz z dopasowaniem liniowym oraz dopasowaniem modelu statystycznego. Linia ciągła oznacza średni wynik ze wszystkich wytrenowanych modeli, zacienione pola wyznaczają obszary 1σ oraz 2σ.

5.3 Analiza nr 3

Dane wejściowe i funkcja straty

Wykorzystane podczas analizy dane składają sie z dwóch zestawów. Pierwszy to 450 punktów pomiarowych przekrojów czynnych, których wartości zostały zmodyfikowane o poprawkę dwufotonową. Struktura danych wejściowych jest analogiczna do tych wykorzystywanych w powyższych analizach - cztery kolumny zawierają zmienną objaśnianą σ , niepewność pomiaru przekroju czynnego $\Delta\sigma$ i dwie zmienne objaśniające Q^2 oraz ϵ. Drugi zestaw to 70 pomiarów doświadczalnych stosunków funkcji postaci G_{E_p}/G_{M_p} wraz z niepewnością pomiarową w zależności od kwadratu czteropędu Q^2. Razem otrzymujemy 520 pomiarów co ponownie sugeruje wybranie parametru k-krotnej walidacji jako $k = 5$. Zastosowana podczas nauki trzeciego modelu funkcja straty χ^2 (5.13) składa się się z tych samych składników co funkcja wykorzystana podczas drugiej analizy (5.8), jednak do całkowitego przekroju czynnego należy dodać poprawkę zależną od kwadratu przekazu czteropędu Q^2 i czynnika kinematycznego ϵ jak przedstawiono w równaniu 5.12.

$$\sigma_R(\epsilon, Q^2) = \tau G_{M_p}^2(Q^2) + \epsilon G_{E_p}^2(Q^2) + \Delta C_{2\gamma}\left(Q^2, \epsilon\right), \tag{5.12}$$

$$\chi^2 = \frac{1}{n}\left[\chi_\sigma^2 + \chi_{PT}^2 + \chi_{G_M}^2 + \chi_{G_E}^2\right]. \tag{5.13}$$

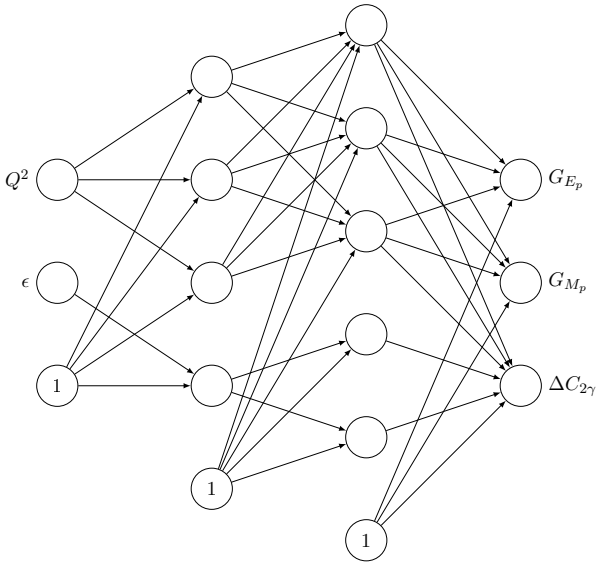
Parametry i nauka sieci

Podczas ostatniej analizy na szczególną uwagę zasługuję struktura sieci neuronowej przedstawiona na rysunku 5.10. Wynikiem działania sieci są funkcje postaci G_{M_p} oraz G_{E_p} w zależności od Q^2 oraz poprawka $\Delta C_{2\gamma}$ zależna od Q^2 i ϵ, takie wymagania wymuszają przerwanie połączeń między niektórymi neuronami. Sieć składa się z warstwy wejściowej, dwóch warstw ukrytych oraz warstwy wyjściowej, wejściowa składa się z dwóch neuronów (Q^2, ϵ). Pierwsza warstwa ukryta zawiera trzy neurony połączone wyłącznie z Q^2 oraz jeden połączony wyłącznie z ϵ (3-1). Druga warstwa ukryta to pięć neuronów, ponownie trzy z nich połączone są z neuronami łączącymi się wyłącznie z kwadratem przekazu czteropędu, pozostałe dwa połączone są z neuronem z poprzedniej warstwy połączonym z czynnikiem kinematycznym (3-2). Ostatnia warstwa wyjściowa składa się z trzech neuronów, dwie funkcje postaci otrzymują sygnał pochądzący z trzech neuronów przekazujących sygnał zależny od wartości Q^2 natomiast poprawka $\Delta C_{2\gamma}$ połączona jest ze wszystkimi neuronami w poprzedniej warstwie, zatem jej wartości są zależne zarówno od Q^2 i ϵ.

Tabela 5.4 przedstawia wykorzystane hiperparametry sieci neuronowej. Normalizacja danych oraz inicjalizacja wag zostały wykonane identycznie jak w poprzednich analizach.

Wyniki

Rysunek 5.11 przedstawia wykresy otrzymanych funkcji postaci protonu. Lewa część obrazka przedstawia magnetyczną funkcję postaci protonu w zależności od kwadratu przekazu czteropędu $G_{M_p}\left(Q^2\right)$, prawa część rysunku przedstawia przebieg elektrycznej funkcji postaci $G_{E_p}\left(Q^2\right)$. Otrzymane wyniki zostały ponownie porównywane są z rezultatami publikacji [2], które całkowicie zawierają się w obszarze $\pm 2\sigma$ otrzymanych wyników. W przypadku magnetycznej funkcji postaci główna różnica to brak zmiany monotoniczności funkcji będącej wynikiem modelu, w

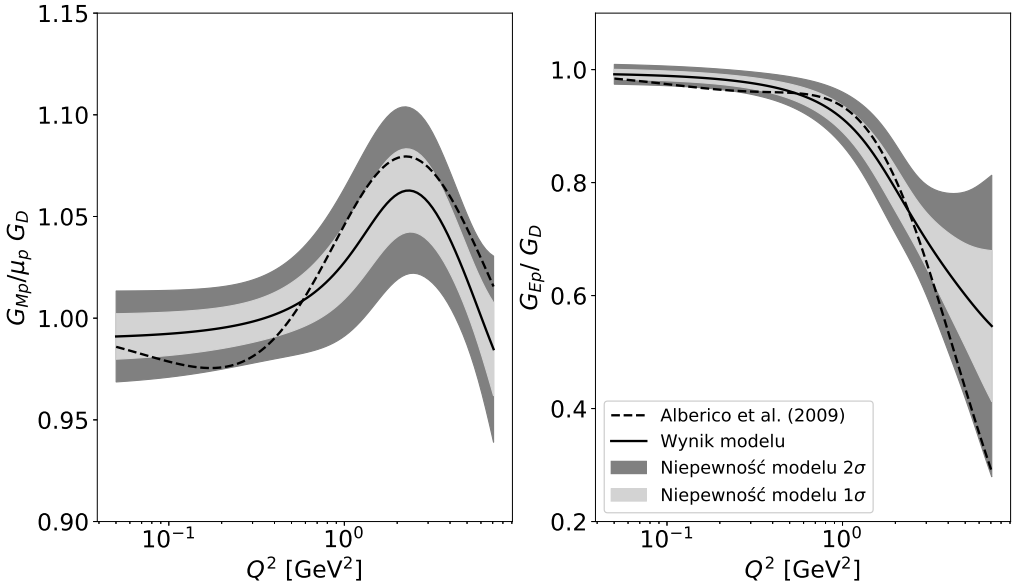


Rysunek 5.10: Schemat sieci neuronowej zastosowanej w trzeciej analizie, która składa się z: i) warstwy wejściowej z dwoma neuronami, ii) dwóch warstw ukrytych z odpowiednio czterema i pięcioma neuronami, iii) warstwy wyjściowej z trzema neuronami. Linie zakończone strzałką oznaczają wagę odpowiadającą każdej z par neuronów

Tabela 5.4: Hiperparametry modelu

Kategoria	Parametr	Wartość
Generowanie danych	N_{rep}	500
k-krotna walidacja krzyżowa	k	5
Algorytm uczący	α (*learning rate*)	0,02
	λ (*decay*)	0,001
	β (*pęd*)	0,9
Sieć neuronowa	Liczba warstw	2
	Ilość neuronów	((3-1),(3-2))

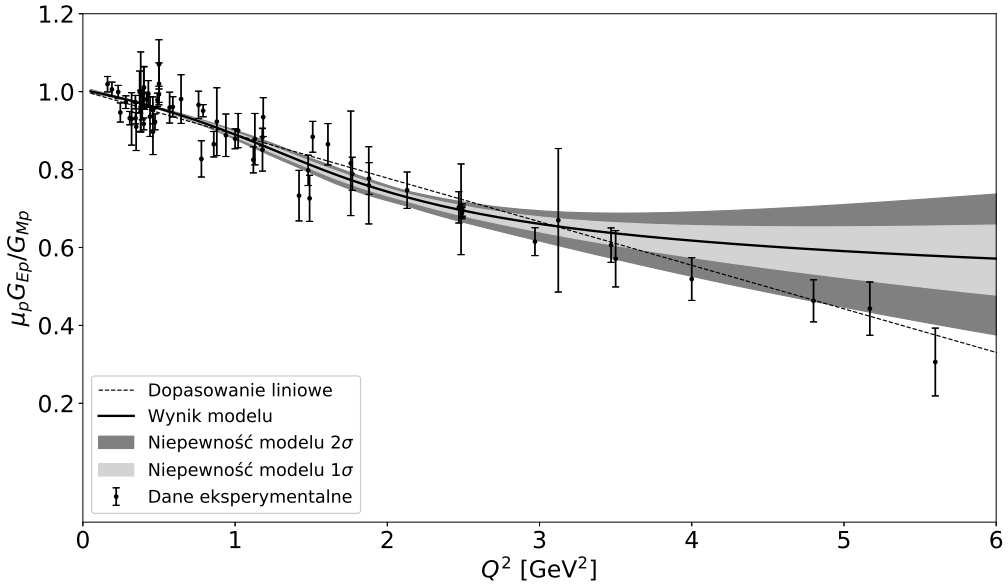
okolicach $Q^2 \simeq 0.15$ GeV^2 oraz późniejsze niedoszacowanie względem porównywanej funkcji, nie przekraczające różnicy 0.02. Elektryczna funkcja postaci również pozbawiona jest pierwszego niewielkiego przecięcia, następnie spadek wartości zaczyna się wcześniej, tzn. dla nieco mniejszej wartości Q^2, jednak jest on mniej gwałtowny. Odchylenie standardowe magnetycznej funkcji postaci jest w przybliżeniu stałe na całej długości przedstawionego przebiegu funkcji i wynosi około 0.01, wariancja elektrycznej funkcji postaci w pobliżu $Q^2 \to 0$ jest bardzo niewielka i znacznie wzrasta wraz z Q^2.



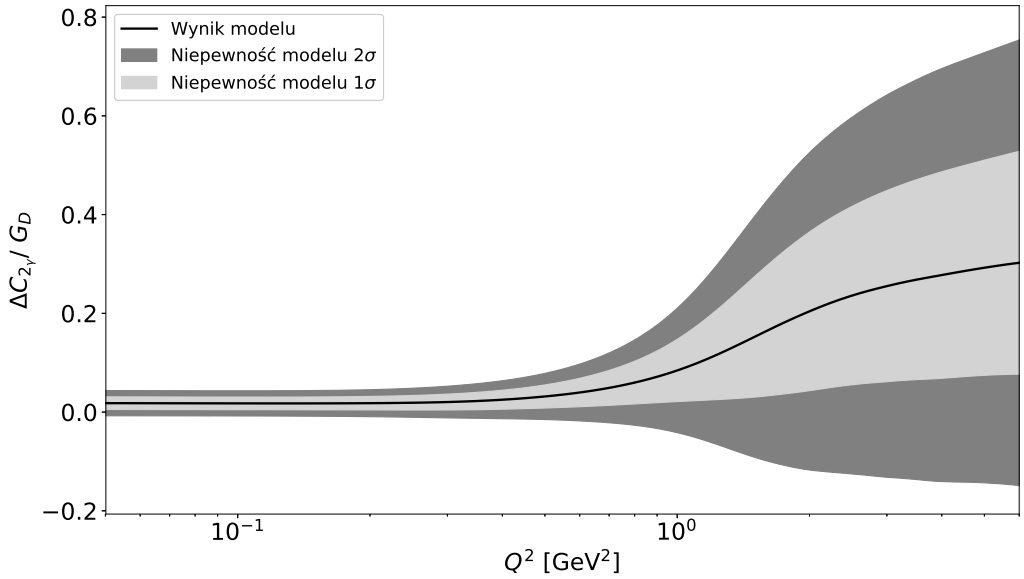
Rysunek 5.11: Magnetyczna i elektryczna funkcja postaci. Linia ciągła oznacza średnią ze wszystkich modeli, zacienione pola wyznaczają obszary 1σ (jasny) oraz 2σ (ciemny). Linia przerywana przedstawia wyniki z publikacji [2].

Rysunek 5.12 przedstawia pomiary eksperymentalne stosunku funkcji postaci wraz z zaznaczoną niepewnością pomiarow, przerywaną linią zaznaczono dopasowanie liniowe oraz wynik modelu wraz z odchyleniami standardowymi. Iloraz $\mu G_{E_p}/G_{M_p}$ szczególnie dobrze opisywany jest przez model dla niewielkich wartości Q^2 i cechuje się w tym obszarze bardzo niewieką niepewnością. Wraz ze wzrostem Q^2 maleje gęstość pomiarów, G_{E_p} staje się przeszacowane, a odchylenie standardowe znacznie wzrasta. Większość pomiarów doświadczalnych znajduje się jednak w obszarze niepewności -2σ.

Jednym z wyników sieci nieuwzględnionym podczas wcześniejszych analiz jest poprawka dwufotonowa $\Delta C_{2\gamma}$, rysunek 5.13 przedstawia jej zależność od kwadratu przekazu czteropędu dla ϵ będących 11 pierwszymi wyrazami ciągu arytmetycznego o przepisie: $\epsilon_n = (n-1) \times 0.1$. Dla wartości $Q^2 < 1$ GeV^2 poprawka ma niewielką wartość i niewielkie odchylenie standardowe, zatem ma niewielki wpływ na wartość całkowitego przekroju czynnego. Dla $Q^2 > 1$ GeV^2 jej średnia wartość zaczyna rosnąć wraz z Q^2 do około 0.3, odchylenie standardowe również gwałtownie wzrasta i osiąga wartość około 0.2.



Rysunek 5.12: Pomiary stosunku elektrycznej i magnetycznej funkcji postaci wraz z dopasowaniem liniowym oraz dopasowaniem modelu statystycznego. Linia ciągła oznacza średni wynik ze wszystkich wytrenowanych modeli, zacienione pola wyznaczają obszary 1σ oraz 2σ.



Rysunek 5.13: Poprawka dwufotonowa w zależności od Q^2, ciągła linia to wynik modelu, kolory szarne oznaczją odchylenie standardowe 1σ oraz 2σ.

Rozdział 6

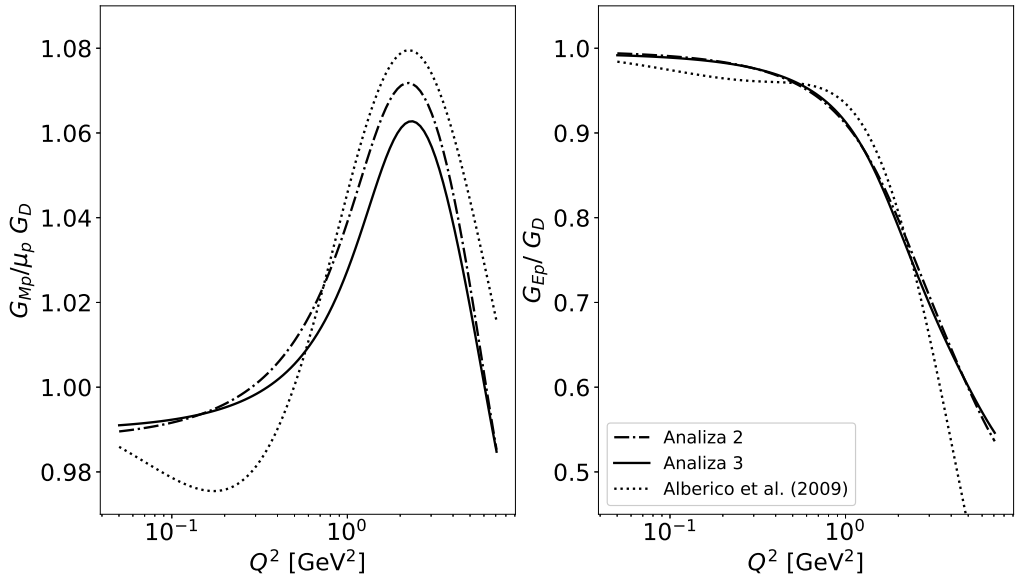
Zakończenie

W powyższej pracy zbudowano modele statystyczne, które wykorzystując pomiary eksperymentalne rozpraszania elektron-proton dają w wyniku przewidywanie elektrycznej i magnetycznej funkcji postaci. W tym celu wykorzystano popularny typ sztucznych sieci neuronowych - perceptron wielowarstwowy. Do nauki modelów posłużyły dwa typy danych, pierwszy z nich zawiera pomiary całkowitych przekrojów czynnych w zależności od kwadratu przekazu czteropędu Q^2 oraz czynnika kinematycznego ϵ, drugi zawiera pomiary stosunków funkcji postaci w zależności od Q^2. Wyniki przeprowadzonych analiz zgadzają się jakościowo z wcześniej przeprowadzonymi badaniami [2], przebiegi funkcji postaci w zależności od Q^2 zawierają się w zakresach niepewności pomiarowych zaprezentowanych na wykresach.

Proces analizy wymagał zrozumienia zarówno fizyki badanego zjawiska jak i dziedziny nauki zajmującej się algorytmami uczenia maszynowego. Opisano metody budowy sieci neuronowych, w tym szczególnie perceptronu wielowarstwowego wraz z metodą nauki modelu. Zwrócono uwagę na istotną zdolność uniwersalnej aproksymacji oraz na charakterystyczne problemy predykcyjnych modelów statystycznych takie jak kompromis między obciążeniem i wariancją. Nauka model statystycznego została zaprogramowana w języku Python przy wykorzystaniu biblioteki Keras. Zaimplementowano działanie niestandardowej funkcji straty, która bierze pod uwagę wyniki operacji przeprowadzanych na dwóch różnych zbiorach danych (równanie 5.13). Dodatkowo zaprogramowano algorytm walidacji krzyżowej oraz wczesnego zatrzymania nauki sieci, które pomogły uniknąć przeuczenia modelu statystycznego. Bazując na idei zaproponowanej w [26], wykorzystując niepewności pomiarowe oraz błędy systematyczne wygenerowano zestawy sztucznych zestawów danych, których celem było zapewnienie gładkiej interpolacji funkcji postaci przy równoczesnym zapewnieniu nieobciążonej estymacji wszystkich pomiarów. Ponadto opisano proces wyboru hiperparametrów modelu i szczegóły działania wybranego algorytmu uczącego.

Przedstawione analizy prezentują wyniki działania trzech modelów, które w różny sposób starają się przewidywać elektryczną i magnetyczną funkcje postaci. Pierwszy z nich do nauki wykorzystuje tylko pomiary całkowitych przekrójów czynnych w zależności od kwadratu przekazu czteropędu Q^2 oraz czynnika kinematycznego ϵ i estymuje całkowity przekrój czynny. Następnie operacje różniczkowania pozwalaja na separacje funkcji postaci. Drugi z modelów wykorzystuje dodatkowo pomiary stosunków funkcji postaci, jego wynikiem są explicite elektrtyczna i magnetyczna funkcja postaci. Trzeci model oprócz szacowania funkcji postaci przewiduje wartość poprawki dwufotonowej. Podczas nauki każdego z modelów skorzystano z metody

5-krotnej walidacji krzyżowej oraz 500 wygenerowanych zestawów sztucznych danych co spowodowało, że jeden model wymagał wytrenowania 2500 sieci neuronowych. Wyniki nauczonych modeli tworzą statystykę, której najważniejszymi parametrami są średnia przedstawiana jako wynik modelu, oraz odchylenie standardowe, które wyznacza zakres bardzo prawdopodobnych wyników.



Rysunek 6.1: Porównanie funkcji postaci protonu wynikających z analiz nr 2 (linia przerywana) oraz 3 (linia ciągła) z wynikami przedstawionymi w [2] (linia kropkowana).

Analiza nr 1 prezentowała najprostszy sposób rozwiązania problemu, korzystała tylko z jednego rodzaju danych i jej wynikiem był całkowity przekrój czynny. Ekstrakcja funkcji postaci z przekroju czynnego była obarczona dużym błędem co negatywnie odbiło się na wynikach modelu i skutkowało dużą wariancją otrzymanych rezultatów. Uwzględnienie pomiarów doświadczalnych stosunków funkcji postaci znacznie podniosło jakość wyników. Rysunek 6.1 przedstawia porównanie funkcji postaci otrzymanych w analizach nr 2 oraz 3. Różnice między analizami nr 2 i 3 wynikające z uwzględnienia poprawki dwufotonowej widoczne są przede wszystkim dla magnetycznej funkcji postaci. Największa różnica w magnetycznych funkcjach postaci w porównaniu do krzywej pochodzącej z [2] pojawia się dla małych wartości Q^2, w wynikach analizy w tym obszarze nie występuje zmiana monotoniczności funkcji. Elektryczna funkcja postaci również pozbawiona jest pierwszego niewielkiego przecięcia, następnie spadek wartości zaczyna się dla nieco mniejszej wartości Q^2, jednak jest on mniej gwałtowny. Całkowite różnice są niewielkie a wyniki przeprowadzonych analiz zgadzają się jakościowo z wcześniej przeprowadzonymi badaniami [2], przebiegi funkcji postaci w zależności od Q^2 zawierają się w zakresach niepewności pomiarowych co przedstawiono na rysunkach 5.8, 5.11.

Bibliografia

[1] A. I. Akhiezer and M. Rekalo. Polarization effects in the scattering of leptons by hadrons. *Sov. J. Part. Nucl.*, 4:277, 1974. [Fiz. Elem. Chast. Atom. Yadra4,662(1973)].

[2] W. M. Alberico, S. M. Bilenky, C. Giunti, and K. M. Graczyk. Electromagnetic form factors of the nucleon: New fit and analysis of uncertainties. *Phys. Rev. C*, 79(6):065204, June 2009.

[3] W. Albrecht, H.-J. Behrend, H. Dorner, W. Flauger, and H. Hultschig. Some Recent Measurements of Proton Form Factors. *Physical Review Letters*, 18:1014–1015, June 1967.

[4] C. Amsler. *Nuclear and Particle Physics*. 2053-2563. IOP Publishing, 2015.

[5] L. Andivahis, P. E. Bosted, A. Lung, L. M. Stuart, J. Alster, and et al. Measurements of the electric and magnetic form factors of the proton from Q^2=1.75 to 8.83 $(GeV/c)^2$. *Physical Review D*, 50:5491–5517, Nov. 1994.

[6] A. Antognini, F. Nez, K. Schuhmann, F. D. Amaro, F. Biraben, and et. al. Proton structure from the measurement of 2s-2p transition frequencies of muonic hydrogen. *Science*, 339(6118):417–420, 2013.

[7] J. Arrington, W. Melnitchouk, and J. A. Tjon. Global analysis of proton elastic form factor data with two-photon exchange corrections. *Physical Review C*, 76(3):035205, Sept. 2007.

[8] W. Bartel, F.-W. Büsser, W.-R. Dix, R. Felst, D. Harms, , and et al. Measurement of proton and neutron electromagnetic form factors at squared four-momentum transfers up to 3 $(GeV/c)^2$. *Nuclear Physics B*, 58:429–475, July 1973.

[9] W. Bartel, B. Dudelzak, H. Krehbiel, J. M. McElroy, U. Meyer-Berkhout, and et al. Small-Angle Electron-Proton Elastic-Scattering Cross Sections for Squared Momentum Transfers Between 10 and 105 F^{-2}. *Physical Review Letters*, 17:608–611, Sept. 1966.

[10] C. Berger, V. Burkert, G. Knop, B. Langenbeck, and K. Rith. Electromagnetic form factors of the proton at squared four-momentum transfers between 10 and 50 fm^{-2}. *Physics Letters B*, 35:87–89, Apr. 1971.

[11] J. Beringer, J. F. Arguin, R. M. Barnett, K. Copic, O. Dahl, and et.ąl. Review of particle physics. *Physical Review D*, 86:010001, Jul 2012.

[12] J. C. Bernauer et al. High-precision determination of the electric and magnetic form factors of the proton. *Physical Review Letters*, 105:242001, 2010.

[13] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[14] F. Borkowski, P. Peuser, G. G. Simon, V. H. Walther, and R. D. Wendling. Electromagnetic form factors of the proton at low four-momentum transfer. *Nuclear Physics A*, 222:269–275, Apr. 1974.

[15] F. Borkowski, G. G. Simon, V. H. Walther, and R. D. Wendling. Electromagnetic form factors of the peoton at low four-momentum transfer (II). *Nuclear Physics B*, 93:461–478, July 1975.

[16] H. Bourlard and Y. Kamp. Auto-association by multilayer perceptrons and singular value decomposition. *Biological cybernetics*, 59:291–4, 02 1988.

[17] C. E. Carlson. The Proton Radius Puzzle. *Prog. Part. Nucl. Phys.*, 82:59–77, 2015.

[18] M. E. Christy, A. Ahmidouch, C. S. Armstrong, J. Arrington, R. Asaturyan, and et al. Measurements of electron-proton elastic cross sections for $0.4< Q^2 <5.5$ $(GeV/c)^2$. *Physical Review C*, 70(1):015206, July 2004.

[19] C. B. Crawford, A. Sindile, T. Akdogan, R. Alarcon, W. Bertozzi, and et al. Measurement of the Proton's Electric to Magnetic Form Factor Ratio from $H \rightarrow 1(e \rightarrow, e' p)$. *Physical Review Letters*, 98(5):052301, Feb. 2007.

[20] G. Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, 2(4):303–314, Dec 1989.

[21] J. de Villiers and E. Barnard. Backpropagation neural nets with one and two hidden layers. *IEEE Transactions on Neural Networks*, 4(1):136–141, Jan 1993.

[22] J. E. Dennis, Jr. and R. B. Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations (Classics in Applied Mathematics, 16)*. Soc for Industrial & Applied Math, 1996.

[23] S. Dieterich, P. Bartsch, D. Baumann, J. Bermuth, K. Bohinc, and et al. Polarization transfer in the ^{4}He(e–>,e'p–>)^{3}H reaction. *Physics Letters B*, 500:47–52, Feb. 2001.

[24] D. Dutta, D. van Westrum, D. Abbott, A. Ahmidouch, T. A. Amatuoni, C. Armstrong, J. Arrington, K. A. Assamagan, and et al. Quasielastic (e, e' p) reaction on ^{12}C , ^{56}Fe , and ^{197}Au. *Physical Review C*, 68(6):064603, Dec. 2003.

[25] J. L. Elman. Finding structure in time. *Cognitive Science*, 14(2):179–211.

[26] S. Forte, L. s. Garrido, J. I. Latorre, and A. Piccione. Neural network parametrization of deep-inelastic structure functions. *Journal of High Energy Physics*, 5:062, May 2002.

[27] C. Francois. *Deep Learning with Python*. Manning Publications, 2017.

[28] O. Gayou, K. A. Aniol, T. Averett, F. Benmokhtar, W. Bertozzi, and et al. Measurement of G_{E_p}/G_{M_p} in e–>p –> ep–> to $Q^2 = 5.6$ GeV^2. *Physical Review Letters*, 88(9):092301, Mar. 2002.

[29] O. Gayou, K. Wijesooriya, A. Afanasev, M. Amarian, K. Aniol, and et al. Measurements of the elastic electromagnetic form factor ratio $\mu_p G_{Ep}/G_{Mp}$ via polarization transfer. *Physical Review C*, 64(3):038202, Sept. 2001.

[30] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *In Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS'10). Society for Artificial Intelligence and Statistics*, 2010.

[31] M. Goitein, R. J. Budnitz, L. Carroll, J. R. Chen, J. R. Dunning, and et al. Elastic Electron-Proton Scattering Cross Sections Measured by a Coincidence Technique. *Physical Review D*, 1:2449–2476, May 1970.

[32] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[33] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative Adversarial Networks. *arXiv e-prints*, June 2014.

[34] K. M. Graczyk. Two-photon exchange effect studied with neural networks. *Phys. Rev. C*, 84:034314, Sep 2011.

[35] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[36] J. J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences of the United States of America*, 79(8):2554–2558, Apr. 1982.

[37] K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, 1 1989.

[38] B. Hu, M. K. Jones, P. E. Ulmer, H. Arenhövel, O. K. Baker, and et al. Polarization transfer in the H2(e$\rightarrow$,e$'$p$\rightarrow$)n reaction up to Q^2=1.61(GeV/c)^2. *Physical Review C*, 73(6):064004, June 2006.

[39] G. James, D. Witten, T. Hastie, and R. Tibshirani. *An Introduction to Statistical Learning: With Applications in R*. Springer Publishing Company, Incorporated, 2014.

[40] T. Janssens, R. Hofstadter, E. B. Hughes, and M. R. Yearian. Proton Form Factors from Elastic Electron-Proton Scattering. *Physical Review*, 142:922–931, Feb. 1966.

[41] Z. Jaskólski. *Notatki do wykładu z mechaniki kwantowej*. 2016.

[42] M. K. Jones, A. Aghalaryan, A. Ahmidouch, R. Asaturyan, F. Bloch, and et al. Proton G_E/G_M from beam-target asymmetry. *Physical Review C*, 74(3):035201, Sept. 2006.

[43] M. K. Jones, K. A. Aniol, F. T. Baker, J. Berthot, P. Y. Bertin, and et al. G_{E_p}/G_{M_p} Ratio by Polarization Transfer in e–>p –> ep–>. *Physical Review Letters*, 84:1398–1402, Feb. 2000.

[44] P. N. Kirk, M. Breidenbach, J. I. Friedman, G. C. Hartmann, H. W. Kendall, and et al. Elastic Electron-Proton Scattering at Large Four-Momentum Transfer. *Physical Review D*, 8:63–91, July 1973.

[45] T. Kohonen. Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 43(1):59–69, Jan 1982.

[46] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, pages 2278–2324, 1998.

[47] Y. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller. Efficient backprop. In *Neural Networks: Tricks of the Trade, This Book is an Outgrowth of a 1996 NIPS Workshop*, pages 9–50, 1998.

[48] M. Leshno, V. Y. Lin, A. Pinkus, and S. Schocken. Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. *Neural Networks*, 6:861–867, 1993.

[49] J. Litt, G. Buschhorn, D. H. Coward, H. Destaebler, L. W. Mo, and et al. Measurement of the ratio of the proton form factors, G_E/G_M, at high momentum transfers and the question of scaling. *Physics Letters B*, 31:40–44, Jan. 1970.

[50] A. Lorenc and J. Bryk. Jak działa neuron. *Delta*, 2005.

[51] G. MacLachlan, A. Aghalaryan, A. Ahmidouch, B. D. Anderson, R. Asaturyan, and et al. The ratio of proton electromagnetic form factors via recoil polarimetry at Q=1.13 GeV. *Nuclear Physics A*, 764:261–273, Jan. 2006.

[52] B. D. Milbrath, J. I. McIntyre, C. S. Armstrong, D. H. Barkhuff, W. Bertozzi, and et al. Erratum: Comparison of Polarization Observables in Electron Scattering from the Proton and Deuteron [Phys. Rev. Lett. 80, 452 (1998)]. *Physical Review Letters*, 82:2221, Mar. 1999.

[53] P. J. Mohr, B. N. Taylor, and D. B. Newell. CODATA recommended values of the fundamental physical constants: 2010. *Reviews of Modern Physics*, 84:1527–1605, Oct. 2012.

[54] J. J. Murphy, Y. M. Shin, and D. M. Skopik. Erratum: Proton form factor from 0.15 to 0.79 fm^{-2}. *Physical Review C*, 10:2111–2111, Nov. 1974.

[55] C. F. Perdrisat, V. Punjabi, and M. Vanderhaeghen. Nucleon electromagnetic form factors. *Progress in Particle and Nuclear Physics*, 59:694–764, Oct. 2007.

[56] R. Pohl, A. Antognini, F. Nez, F. D. Amaro, F. Biraben, and et al. The size of the proton. *Nature*, 466:213–218, July 2010.

[57] T. Pospischil, P. Bartsch, D. Baumann, R. Böhm, K. Bohinc, and et al. Measurement of G_{Ep}/G_{Mp} via polarization transfer at $Q^2 = 0.4$ GeV $/c^2$. *European Physical Journal A*, 12:125–127, Sept. 2001.

[58] praca zbiorowa. *Biologia; jedność i różnorodnoć*. 2008.

[59] L. E. Price, J. R. Dunning, M. Goitein, K. Hanson, T. Kirk, and R. Wilson. Backward-Angle Electron-Proton Elastic Scattering and Proton Electromagnetic Form Factors. *Physical Review D*, 4:45–53, July 1971.

[60] V. Punjabi, C. F. Perdrisat, K. A. Aniol, F. T. Baker, J. Berthot, and et al. Proton elastic form factor ratios to Q^2=3.5GeV^2 by polarization transfer. *Physical Review C*, 71(5):055202, May 2005.

[61] S. Raschka. *Python Machine Learning*. Packt Publishing, 2015.

[62] S. Rock, R. G. Arnold, P. E. Bosted, B. T. Chertok, B. A. Mecking, and et al. Measurement of elastic electron-neutron scattering and inelastic electron-deuteron scattering cross sections at high momentum transfer. *Physical Review D*, 46:24–44, July 1992.

[63] G. Ron, J. Glister, B. Lee, K. Allada, W. Armstrong, and et al. Measurements of the Proton Elastic-Form-Factor Ratio $\mu_p G_E{}^p/G_M{}^p$ at Low Momentum Transfer. *Physical Review Letters*, 99(20):202002, Nov. 2007.

[64] F. Rosenblatt. The perceptron, a perceiving and recognizing automaton : (project para). *Cornell Aeronautical Laboratory report*, 85-460-1.

[65] M. N. Rosenbluth. High energy elastic scattering of electrons on protons. *Physical Review*, 79:615–619, Aug 1950.

[66] S. Ruder. An overview of gradient descent optimization algorithms. *ArXiv e-prints*, Sept. 2016.

[67] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *Nature*, 323:533–536, Oct. 1986.

[68] A. F. Sill, R. G. Arnold, P. E. Bosted, C. C. Chang, J. Gomez, and et al. Measurements of elastic electron-proton scattering at large momentum transfer. *Physical Review D*, 48:29–55, July 1993.

[69] G. G. Simon, C. Schmitt, F. Borkowski, and V. H. Walther. Absolute electron-proton cross sections at low momentum transfer measured with a high pressure gas target system. *Nuclear Physics A*, 333:381–391, Jan. 1980.

[70] G. G. Simon, C. Schmitt, and V. H. Walther. Elastic electric and magnetic e-d scattering at low momentum transfer. *Nuclear Physics A*, 364:285–296, July 1981.

[71] StatSoft. Uczenie perceptronu wielowarstwowego. *Internetowy Podręcznik Statystyki*, 2011.

[72] S. Stein, W. B. Atwood, E. D. Bloom, R. L. A. Cottrell, H. Destaebler, and et al. Electron scattering at 4deg with energies of 4.5-20 GeV. *Physical Review D*, 12:1884–1919, Oct. 1975.

[73] R. Taylor. Deep inelastic scattering: The early years. *Nobel Lecture*, Dec. 1990.

[74] R. C. Walker, B. W. Filippone, J. Jourdan, R. Milner, R. McKeown, and et al. Measurements of the proton elastic form factors for $1<=Q^2<=3$ $(GeV/c)^2$ at SLAC. *Physical Review D*, 49:5671–5689, June 1994.