

UNIWERSYTET ZIELONOGÓRSKI

Wydział Informatyki, Elektrotechniki i Automatyki

Praca magisterska

Kierunek: Informatyka

**WYKORZYSTANIE GŁĘBOKICH SIECI
NEURONOWYCH DO WSPOMAGANIA
DIAGNOSTYKI MEDYCZNEJ**

Krystian Dziędziola

Promotor:
dr inż. Artur Gramacki

Zielona Góra, czerwiec 2019

Streszczenie

Głównym celem pracy jest przedstawienie wyników przeprowadzonych badań dotyczących możliwości zastosowania głębokich sieci neuronowych do wspomagania diagnostyki medycznej. Przykładowym zadaniem, które zostało wybrane jako obiekt badań jest diagnozowanie stanów padaczkowych na podstawie odczytów z elektroencefalogramu (EEG).

// todo: uzupełnić

Słowa kluczowe: Sieci neuronowe, diagnostyka medyczna, EEG, deep learning.

Spis treści

1. Wstęp	1
1.1. Wprowadzenie	1
1.2. Cel i zakres pracy	1
1.3. Przegląd literatury	2
1.4. Struktura pracy	2
2. Wprowadzenie do głębokich sieci neuronowych	3
2.1. Sieci neuronowe a sztuczna inteligencja	3
2.2. Klasyczne sieci neuronowe	5
2.3. Deep learning, czyli głębokie sieci neuronowe	6
2.4. Architektury głębokich sieci neuronowych	8
2.4.1. Splotowe sieci neuronowe	8
2.4.2. Rekurencyjne sieci neuronowe	9
3. Omówienie wybranego problemu diagnostyki medycznej	11
3.1. Przedstawienie problemu	11
3.2. Dostępne dane	11
3.3. Oczekiwane rezultaty	11
4. Przegląd dostępnych bibliotek oraz narzędzi programistycznych	12
4.1. Dostępne narzędzia	12
4.2. Wybrany stos technologiczny	12
5. Próba rozwiązania problemu	13
5.1. Przygotowanie danych	13
5.2. Implementacja procesu uczenia	13
5.3. Metoda oceny wyników	13
5.4. Wybór rodzaju sieci neuronowej	13
5.5. Budowa modelu	13
5.6. Monitorowanie	13
5.7. Optymalizacja	13
5.8. Ocena otrzymanych rezultatów	13
6. Zakończenie	14
A. Płyta DVD	15

Spis rysunków

2.1. Porównanie klasycznego programowania z metodą uczenia maszynowego	4
2.2. Zależność między sztuczną inteligencją, uczeniem maszynowym i sie- ciami neuronowymi	5
2.3. Struktura prostej sieci neuronowej	5
2.4. Schemat procesu uczenia głębokiej sieci neuronowej	7
2.5. Dane liczbowe obrazka w formacie RGB	9
2.6. Schemat działania sieci splotowej	9
2.7. Pętla w rekurencyjnej sieci neuronowej	10

Spis tabel

Rozdział 1

Wstęp

1.1. Wprowadzenie

Współczesna technologia sztucznej inteligencji, której dużą część stanowią sztuczne sieci neuronowe, pozwala na wykonywanie przez maszyny zadań, które do niedawna byli w stanie wykonywać tylko ludzie. Jednym z przykładów jest diagnostyka medyczna, która wymaga specjalistycznej wiedzy oraz doświadczenia.

Dzięki sztucznym sieciom neuronowym maszyna jest w stanie nauczyć się pewnych reguł, na podstawie których podejmuje decyzje, które mogą symulować ludzką inteligencję. W niektórych problemach maszyna jest w stanie podejmować poprawne decyzje z dokładnością dorównującą lub nawet przewyższającą zdolności człowieka. Dodatkowo może to zrobić w znacznie krótszym czasie.

// todo: dokończyć

1.2. Cel i zakres pracy

Celem pracy jest próba wykorzystania głębokich sieci neuronowych do wspomagania diagnostyki medycznej dla wybranego problemu praktycznego oraz osiągnięcie możliwie najlepszych rezultatów.

Zakres pracy obejmuje:

- krótkie wprowadzenie w tematykę sieci głębokich oraz porównanie z klasycznymi sieciami neuronowymi,
- przegląd dostępnych bibliotek i narzędzi programistycznych dla sieci głębokich,
- omówienie praktycznego problemu z dziedziny diagnostyki medycznej i próba rozwiązania z wykorzystaniem sieci głębokich,
- przedstawienie szczegółów technicznych i implementacyjnych,
- wykonanie eksperymentów, ich ocena oraz sformułowanie wniosków końcowych.

1.3. Przegląd literatury

[1] [2] [3] // todo:

1.4. Struktura pracy

// todo:

Rozdział 2

Wprowadzenie do głębokich sieci neuronowych

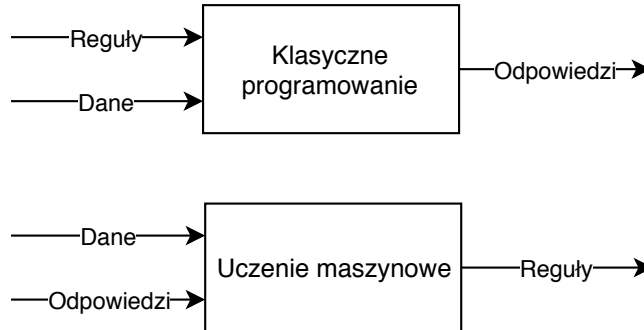
2.1. Sieci neuronowe a sztuczna inteligencja

W ciągu ostatnich kilku lat sztuczna inteligencja stała się tematem dużego zainteresowania w nauce, technologii oraz życiu codziennym. Ma to miejsce za sprawą szerokiego spektrum jej zastosowań oraz możliwości wykorzystania do zadań, które do tej pory były niemal nieosiągalne przez maszyny. Są to m.in. autonomiczne pojazdy, inteligentni wirtualni asystenci, wspomaganie diagnostyki medycznej oraz wiele innych.

Sztuczna inteligencja datuje swoje początki na lata 50-te XX wieku kiedy to zaczęto prowadzić badania nad tym czy komputer jest w stanie nauczyć się "myśleć". Bardziej ścisłą definicją celu prowadzonych badań mogłaby być "próba automatyzacji zadań intelektualnych, które normalnie wykonywane są przez człowieka". Początkowo próby te były poczynione poprzez zaprogramowanie szeregu reguł, którymi miała posługiwać się maszyna w podejmowaniu swoich własnych decyzji. Jest to tzw. symboliczna sztuczna inteligencja (*ang. symbolic artificial intelligence*), czyli klasyczne podejście polegające na zaprogramowaniu określonego algorytmu działania. [4] Zauważono jednak, że zbiór określonych przez programistów reguł sprawdza się do rozwiązywania dobrze zdefiniowanych problemów logicznych takich jak np. gra w szachy. Takie podejście nie jest jednak wystarczające do osiągnięcia przez komputer umiejętności podejmowania decyzji, która mogłaby symulować ludzką inteligencję oraz wykonwania przez nią bardziej abstrakcyjnych zadań takich jak np. rozpoznawanie mowy, obrazu lub tłumaczenie języków.

Nasuwa się pytanie czy komputer jest w stanie wyjść poza ramy ściśle zdefiniowanych reguł i samemu nauczyć się w jaki sposób wykonywać określone zadania? Odpowiedzią na to pytanie oraz niedoskonałości klasycznego programowania było zastosowanie podejścia zwanego dziś uczeniem maszynowym (*ang. machine learning*). Polega ono na tym, że maszynie nie jest dostarczany ściśle określony algorytm działania, a jedynie przedstawiane są dane oraz wyniki jakie powinny powstać po przetworzeniu tychże danych. Maszyna ma za zadanie stworzyć swoje własne reguły, które będą odzwierciedlać sposób w jaki prawidłowo należy wykonać konkretne zadanie. Można więc powiedzieć, że w uczeniu maszynowym, maszyna jest uczona, a nie programowana. Pomimo tego, że uczenie maszynowe zaczęło się roz-

wijać dopiero w latach 90-tych bardzo szybko stało się popularne za sprawą tego, że przynosiło najlepsze wyniki spośród wszystkich podejść w dziedzinie sztucznej inteligencji. Różnicę pomiędzy klasycznym programowaniem a uczeniem maszynowym zaprezentowano na rysunku 2.1.



Rys. 2.1. Porównanie klasycznego programowania z metodą uczenia maszynowego

Mnogości zastosowań uczenia maszynowego dorównuje również ilość algorytmów, które są wykorzystywane w tym podejściu. Dostępnych jest wiele modeli, poczynając od bardzo prostych, kończąc na dosyć skomplikowanych. Kilka najczęściej używanych, to m.in.:

- **drzewa decyzyjne** (*ang. decision trees*),
- **algorytm k-średnich** (*ang. k-means*),
- **sztuczne sieci neuronowe** (*ang. artificial neural networks*). [5]

Przedmiotem tej pracy jest ostatnia z wymienionych metody, czyli sztuczne sieci neuronowe. Jest to zestaw algorytmów, który został zaprojektowany w sposób inspirowany działaniem ludzkiego mózgu. W rzeczywistości jednak, ich działanie nie jest do końca zgodne z tym, w jaki sposób działa umysł człowieka. Sieci neuronowe potrafią przyjmować dane w formie wektorów liczbowych, które mogą zawierać takie informacje jak np. obrazy, dźwięki, tekst lub szeregi czasowe. Na ich podstawie są w stanie dokonywać klasyfikacji, klasteryzacji oraz predykcji. Jednym z typów sztucznych sieci neuronowych są tzw. głębokie sieci neuronowe (*ang. deep neural networks*), które znacząco zwiększają możliwości klasycznych sieci neuronowych. Zostaną one omówione w rozdziale 2.3.

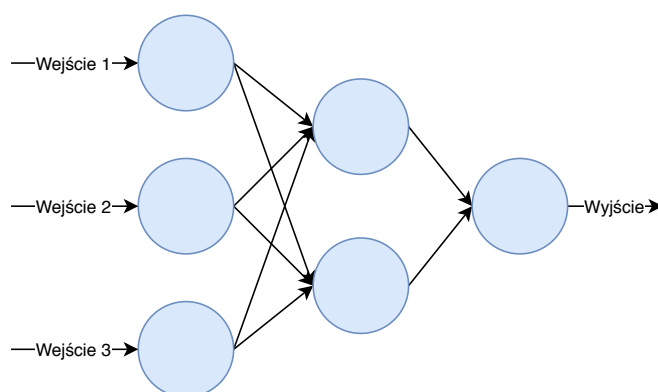
Można zauważyć, że sztuczna inteligencja jest pojęciem bardzo szerokim. W jej skład wchodzi różne metody m.in. uczenie maszynowe. Z kolei specyficznym podejściem w uczeniu maszynowym są sieci neuronowe. Zależności pomiędzy tymi pojęciami przedstawia rysunek 2.2.



Rys. 2.2. Zależność między sztuczną inteligencją, uczeniem maszynowym i sieciami neuronowymi

2.2. Klasyczne sieci neuronowe

Sieci neuronowe to dziedzina uczenia maszynowego, która jako model matematyczny podlegający uczeniu wykorzystuje strukturę sieci składającą się z wielu jednostek obliczeniowych zwanych neuronami. Neurony wykonują podstawowe obliczenia i ich wyniki przekazują do kolejnych neuronów. Operacje, które są wykonywane przez neurony to przeważnie sumowanie otrzymanych informacji oraz zastosowanie prostej, nieliniowej funkcji. W większości sieci neuronowych, neurony zgrupowane są w tzw. warstwy. Obliczenia wykonane przez jedną warstwę są przekazywane do kolejnej, która z kolei wykonuje obliczenia na otrzymanych wynikach. Ostatnia warstwa zwraca końcowy wynik, który jest interpretowany w różny sposób w zależności od wykonywanej operacji np. klasyfikacja lub regresja. Schemat prostej sieci neuronowej przedstawiony został na rysunku 2.3



Rys. 2.3. Struktura prostej sieci neuronowej

Pierwszą koncepcję neuronu datuje się na rok 1943, w którym to powstał model neuronu McCulloch-Pitts. Był on bardzo prosty w porównaniu do współczesnych sieci neuronowych, gdyż pozwalał jedynie na wykorzystanie wartości binarnych na wyjściu z neuronów. Każdy z nich sumował wartości wejściowe i przyrównywał je do zera. Dodatkowo nie istniała żadna reguła aktualizacji wewnętrznych wartości neuronów (tzw. wag), która jest niezbędna do prawidłowego przebiegu procesu adaptacji neuronu do nowych informacji. Bez takiej reguły wszystkie wagi neuronów

musiałyby być ustawiane ręcznie.

W latach 50-tych przedstawiony został perceptron, który jest najprostszą siecią neuronową składającą się z wielu neuronów McCullocha-Pittsa. Implementuje on algorytm uczenia nadzorowanego klasyfikatorów binarnych. Jest funkcją przynależności potrafiąca przydzielić jedną z dwóch klas do danych parametrów wejściowych. W perceptronie została przedstawiona prosta reguła służąca do aktualizacji wag dla kolejnych iteracji. Wzór na wagę w kroku $t + 1$ przedstawia się następująco:

$$w_i(t + 1) = w_i(t) + (d_j - y_j(t))x_{j,i} \quad (2.1)$$

gdzie w_i to waga i -tego neuronu, d_j to oczekiwana wartość dla j -tego wejścia, $y_i(t)$ oznacza wartość obliczoną z j -tego wejścia, natomiast $x_{j,i}$ jest i -tą wartością j -tego wejścia. Oznacza to, że przyszła wartość wagi neuronu obliczona jest przez dodanie błędu (różnicy między wartością oczekiwaną a obliczoną) pomnożonego przez wartość rzeczywistą do wartości aktualnej wagi. Reguła ta może być stosowana jedynie do uczenia jednowarstwowych sieci neuronowych, co znacznie ogranicza zakres jej zastosowań. Perceptron może więc być używany jedynie do problemów liniowo separowalnych.[2]

W latach 60-tych zostało matematycznie udowodnione, że sieć z pojedynczą warstwą nie posiada możliwości klasyfikowania problemów, które nie są liniowo separowalne. Przykładem jest funkcja alternatywy wykluczającej - XOR. Kolejne, bardziej skomplikowane problemy, których się podejmowano, takie jak m.in. rozpoznawanie mowy, kończyły się więc niepowodzeniem. Możliwość uczenia sieci wielowarstwowych była więc konieczna w dalszym rozwoju sieci neuronowych. W tej samej publikacji [3] zostało również udowodnione, że 2-warstwowa sieć jest w stanie zamodelować niemal każdą funkcję. W praktyce jest to jednak ciężkie do osiągnięcia.

W późniejszym okresie przedstawiony został algorytm wstecznej propagacji błędów (*ang. backpropagation*), który umożliwiał uczenie wielowarstwowych sieci neuronowych. Oparty jest on na minimalizacji wartości tzw. funkcji straty z wykorzystaniem optymalizacyjnej metody największego spadku. Błędy obliczane są na warstwie wyjściowej i przekazywane są wstecz, do warstw poprzedzających. Jest to możliwe dzięki wykorzystaniu tzw. metody łańcuchowej (*ang. chain rule*). Dzięki zastosowaniu tego typu podejścia osiągnięty został cel, który zakładał umożliwienie uczenia wielowarstwowych sieci neuronowych. Pozwoliło to w znaczącym stopniu przyspieszyć rozwój w tej dziedzinie.

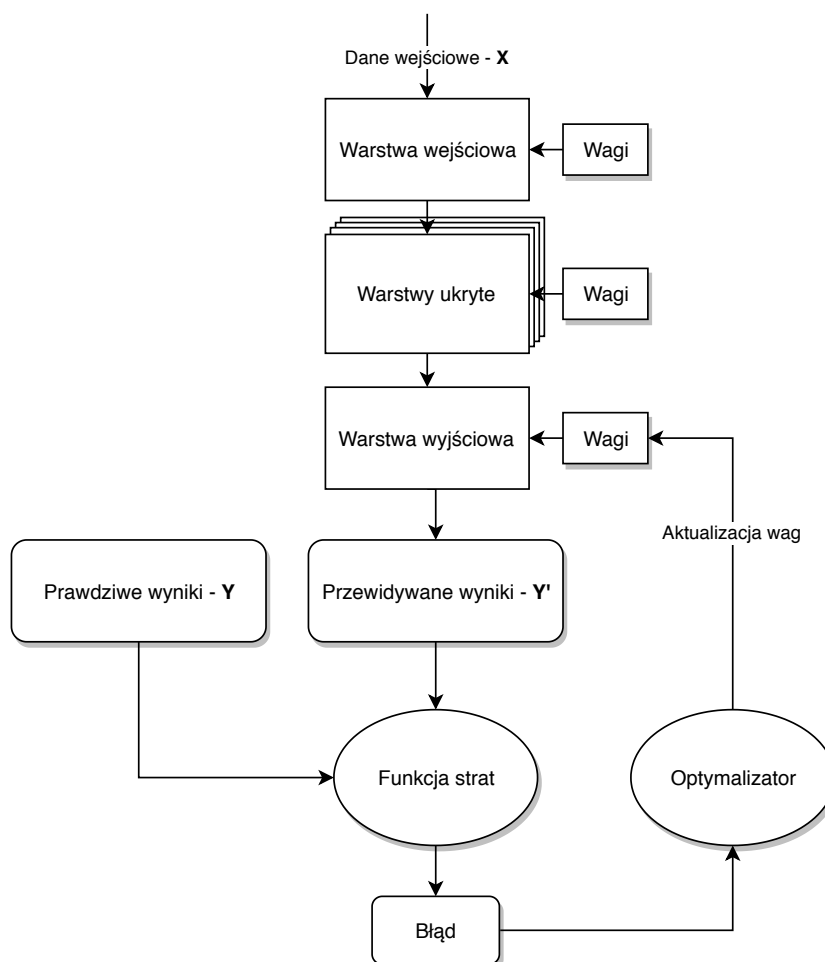
Algorytm wstecznej propagacji błędów stał się najpopularniejszym i najbardziej skutecznym algorytmem do nauki sieci neuronowych. Ma on właśnie zastosowanie w głębokich sieciach neuronowych, które zostały opisane w rozdziale 2.3.

2.3. Deep learning, czyli głębokie sieci neuronowe

Deep learning to dziedzina uczenia maszynowego, która zakłada nowe podejście do nauki wzorców w danych. Kładzie ona nacisk przede wszystkim na naukę kolejnych warstw reprezentacji, które z każdą warstwą są coraz bardziej konkretne. Słowo "głębokie" (*ang. deep*) odnosi się głównie do ilości warstw, które składają się na model sieci. Ta cecha jest zwykle nazywana głębokością sieci (*ang. depth*). Nowoczesne modele głębokich sieci neuronowych często uczą się dziesiątek, a czasem nawet se-

tek kolejnych warstw reprezentacji danych. Każda z nich uczy się samodzielnie na podstawie przedstawionych im informacji.

Inne podejścia, takie jak np. klasyczne uczenie maszynowe skupia się na nauce jedynie jednej lub dwóch warstw reprezentacji danych, dlatego tego typu metody są często nazywane "płytkim uczeniem" (*ang. shallow learning*). W deep learning'u jako model matematyczny prawie zawsze używa się sieci neuronowych, które składają się z nałożonych na siebie wielu warstw neuronów.



Rys. 2.4. Schemat procesu uczenia głębokiej sieci neuronowej

Każda z warstw posiada zestaw tzw. wag (*ang. weights*), które określają sposób w jaki interpretowane są dane wejściowe. Z technicznego punktu widzenia wagi są parametrami transformacji, które są wykonywane przez warstwy na danych wejściowych. W tym kontekście uczenie oznacza poszukiwanie odpowiedniego zestawu wag, który pozwoli na poprawne przekształcanie danych wejściowych w oczekiwane rezultaty. W głębokich sieciach neuronowych mogą istnieć miliony tego typu parametrów, więc znalezienie odpowiedniej wartości dla każdego z nich może być trudnym zadaniem, biorąc pod uwagę, że zmiana jednego parametru może mieć wpływ zachowanie innych.

W celu poprawnej modyfikacji parametrów sieć musi obserwować wyniki wyjściowe i porównywać je z wynikami docelowymi, które dostarczone zostały w procesie uczenia. Pomiaru tego, jak bardzo uzyskane wyniki różnią się od tych prawdzi-

wych zajmuje się tzw. funkcja strat (*ang. loss function*). Zwraca ona konkretną wartość liczbową popełnionego błędu, który następnie może być minimalizowany w celu uzyskiwania coraz to lepszych wyników. Odbywa się to poczynając od warstwy ostatniej, kończąc na warstwie wejściowej stosując algorytm wstecznej propagacji błędów, który implementowany jest przez tzw. optymalizator (*ang. optimizer*). [6]

Początkowo wagi posiadają losowe wartości, które dostosowywane są w ramach wielu iteracji zwanych epokami uczącymi (*ang. epochs*). Przeważnie proces powtarzany jest dziesiątki razy na tysiącach przykładowych danych. Wybierany jest zestaw wag sieci, który posiada najmniejszą wartość funkcji strat, czyli jest najlepiej dostosowany do poprawnego przetwarzania danych wejściowych na wyjściowe. Cały proces uczenia głębokiej sieci neuronowej został zaprezentowany na rysunku 2.4.

Możliwość rozwoju głębokich sieci neuronowych była więc uwarunkowana powstaniem sprzętu o odpowiedniej mocy obliczeniowej, która umożliwiłaby wykonywanie tak dużej ilości obliczeń w skończonym czasie. Obecnie używane są do tego procesory kart graficznych (*GPU - ang. graphics processing unit*), które swoimi możliwościami znacznie przewyższają zwykłe procesory (*CPU - central processing unit*).

2.4. Architektury głębokich sieci neuronowych

Wyróżnia się 5 najczęściej stosowanych architektur głębokich sieci neuronowych. Są to:

- **Sieci splotowe** (*CNN - ang. Convolutional Neural Networks*)
- **Rekurencyjne sieci neuronowe** (*ang. Recurrent Neural Networks*)
- **ResNets** (*Residual Networks*)
- **Autoenkodery** (*ang. Autoencoders*)
- **GAN** (*ang. Generative Adversarial Networks*)

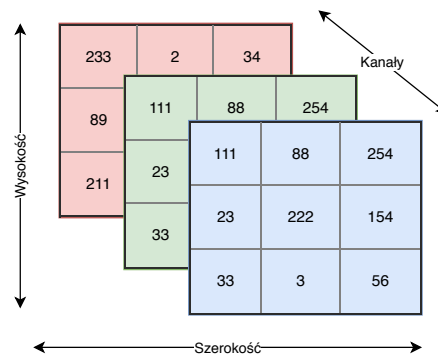
Zostaną omówione jedynie dwie pierwsze architektury, gdyż to one znalazły zastosowanie w niniejszej pracy.

2.4.1. Splotowe sieci neuronowe

Splotowe sieci neuronowe są niewątpliwie najpopularniejszą architekturą głębokich sieci neuronowych. Są one przeważnie używane do zadań związanych z rozpoznawaniem obrazów, jednak równie dobrze sprawdzają się w wielu innych dziedzinach.

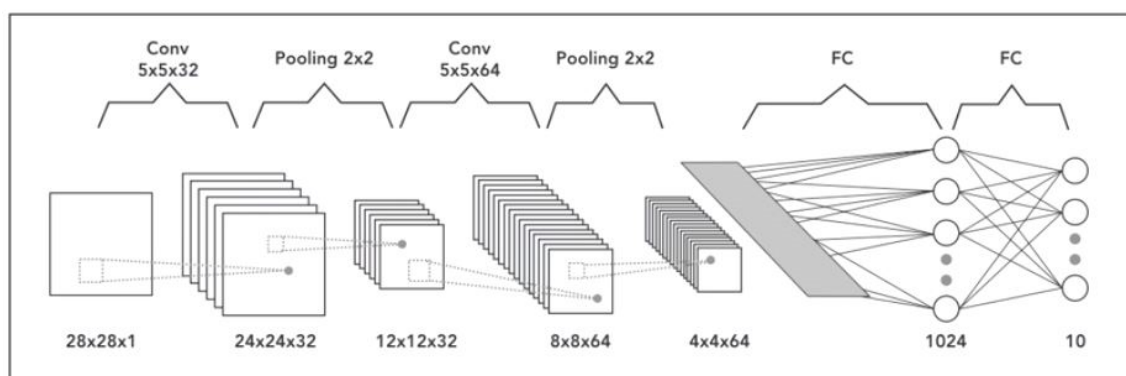
Przewagą tego typu sieci nad klasycznymi sieciami neuronowymi jest możliwość rozpoznawania tymczasowych oraz przestrzennych zależności pomiędzy danymi. Jest to możliwe, dzięki temu, że dane które trafiają do sieci mogą posiadać strukturę wielowymiarową. Przykładowo, obrazki zapisane w formacie RGB są 3-wymiarową macierzą przechowującą informację o wartościach koloru każdego z pikseli 2.5.

Na danych przestrzennych wykonywane są operacje takie jak: konwolucja (*ang. convolution*), zmniejszanie rozmiaru danych (*ang. subsampling*) oraz aktywacja (*ang. activation*). Znalezione cechy są następnie podawane na wejście klasycznej sieci,



Rys. 2.5. Dane liczbowe obrazka w formacie RGB

gdzie neurony połączone są każdy z każdym (*ang. fully connected*), która wykonuje ostateczne operacje i zwraca wynik końcowy. Działanie sieci splotowej polega na tworzeniu kolejnych tzw. map cech (*ang. feature map*), które są abstrakcyjnymi reprezentacjami danych umożliwiającymi wychwycenie podobieństw i zależności pomiędzy danymi. Szczegółowy opis sposobu działania tych operacji nie będzie tutaj omawiany. Został on przedstawiony m.in. w [1] oraz [6]. Przykładowy schemat działania splotowej sieci neuronowej przedstawiony jest na rysunku 2.6.



Rys. 2.6. Schemat działania sieci splotowej

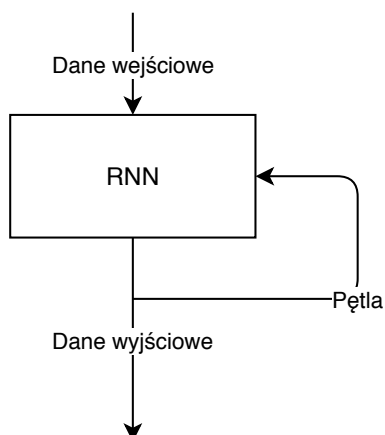
Zaletą tego typu sieci jest bez wątpienia możliwość przetwarzania wielowymiarowych danych, dzięki czemu są one w stanie odnajdywać przestrzenne zależności pomiędzy nimi, co przekłada się na ich wysoką skuteczność i wszechstronność.

2.4.2. Rekurencyjne sieci neuronowe

Rekurencyjne sieci neuronowe są często używane do rozwiązywania problemów, gdzie wykorzystywane są dane sekwencyjne. W praktyce znajdują one zastosowanie w takich problemach jak np. przetwarzanie języka naturalnego, synteza mowy czy automatyczne tłumaczenie języków.

Idea działania rekurencyjnych sieci neuronowych polega na "zapamiętywaniu" pewnego ciągu sekwencji i dokonywanie wyborów nie tylko na podstawie aktualnych danych, ale również tych, które były przetwarzane poprzednio. Jest to realizowane za pomocą wewnętrznej pętli, która polega na przekazywaniu wyników z danej warstwy jako dodatkowych danych wejściowych podczas kolejnej iteracji. Schemat pętli

zaprezentowany został na rysunku 2.7.



Rys. 2.7. Pętla w rekurencyjnej sieci neuronowej

Zaletą rekurencyjnych sieci neuronowych jest to, że współdzielą one jeden zestaw parametrów we wszystkich krokach, dzięki temu pozwala to zredukować ich ilość. Mogą być one również używane w połączeniu z sieciami splotowymi.

Do wad należy m.in. to, że nie są one w stanie zapamiętywać zbyt długich sekwencji. Dodatkowo ich głębokość nie może być zbyt duża, ponieważ używane funkcje aktywacji powodują, że gradient zmian zanika po przejściu przez kilka warstw.

Rozdział 3

Omówienie wybranego problemu diagnostyki medycznej

3.1. Przedstawienie problemu

3.2. Dostępne dane

3.3. Oczekiwane rezultaty

Rozdział 4

Przegląd dostępnych bibliotek oraz narzędzi programistycznych

4.1. Dostępne narzędzia

4.2. Wybrany stos technologiczny

Rozdział 5

Próba rozwiązania problemu

5.1. Przygotowanie danych

5.2. Implementacja procesu uczenia

5.3. Metoda oceny wyników

5.4. Wybór rodzaju sieci neuronowej

5.5. Budowa modelu

5.6. Monitorowanie

5.7. Optymalizacja

5.8. Ocena otrzymanych rezultatów

Rozdział 6

Zakończenie

Dodatek A

Płyta DVD

Do tekstu pracy załączona została płyta DVD z następującą zawartością:

- plik **/praca-dyplomowa.pdf** - tekst pracy dyplomowej,
- katalog **/projekt/** - zawiera wszystkie pliki wykonanego projektu,
- katalog **/oprogramowanie/** - zawiera oprogramowanie wymagane do uruchomienia projektu. W szczególności są to:

// todo: uzupełnić

- katalog **/program/** - program,

Bibliografia

- [1] Francois Chollet. *Deep Learning with Python*. 2018.
- [2] Jacob M. Williams. *Deep Learning and Transfer Learning in the Classification of EEG Signals*. 2017.
- [3] Marvin Minsky and Seymour Paper. *Perceptrons*. 1969.
- [4] futureoflife.org. *Benefits & risks of artificial intelligence*.
<https://futureoflife.org/background/benefits-risks-of-artificial-intelligence>.
- [5] searchenterpriseai.techtarget.com. *Machine learning (ML)*. <https://searchenterpriseai.techtarget.com/definition/machine-learning-ML>.
- [6] Yoshua Bengio Ian Goodfellow and Aaron Courville. Deep learning. Book in preparation for MIT Press, 2016.