

W świecie komputerów i sieci neuronowych

Waldemar

Dylewski

SZCZECIN 2023

W świecie komputerów i sieci neuronowych

Waldemar Dylewski

1

Wprowadzenie do komputerów

- 1.1. Czym jest komputer?
- 1.2. Dlaczego komputery są ważne?
- 1.3. Historia komputerów
- 1.4. Różne rodzaje komputerów
(np. komputerów osobistych, tabletów,
smartfonów)

2

Jak działa komputer neuronowy?

- 2.1. Co to jest komputer neuronowy?
- 2.2. Porównanie z naszym mózgiem
- 2.3. Neurony w komputerze neuronowym
- 2.4. Działanie komputera neuronowego

3

Tworzenie prostego komputera neuronowego

- 3.1. Jak zbudować prosty komputer
neuronowy
- 3.2. Użycie bloków i kabli
- 3.3. Programowanie komputera
neuronowego

W świecie komputerów i sieci neuronowych

Waldemar Dylewski

4

Sieci neuronowe

- 4.1. Czym są sieci neuronowe?
- 4.2. Przykłady zastosowań sieci neuronowych
- 4.3. Jak sieci neuronowe uczą się

5

Wizualizacja sieci neuronowych

- 5.1. Jak wygląda sieć neuronowa?
- 5.2. Diagramy i rysunki przedstawiające sieci neuronowe

6

Jak sieci neuronowe rozwiązują problemy

- 6.1. Przykłady, jak sieci neuronowe rozwiązują problemy (np. rozpoznawanie obrazów, tłumaczenie języka)
- 6.2. Jak działają w praktyce

W świecie komputerów i sieci neuronowych

Waldemar Dylewski

Etyka i bezpieczeństwo w świecie komputerów neuronowych

7

- 7.1. Dlaczego ważne jest dbanie o etykę w technologii
- 7.2. Bezpieczeństwo w sieciach neuronowych

Zabawki i gry związane z komputerami neuronowymi

8

- 8.1. Gry i zabawki, które wykorzystują komputery neuronowe
- 8.2. Tworzenie własnych prostych projektów

Przyszłość komputerów neuronowych

9

- 9.1. Jak technologia się rozwija
- 9.2. Co przyniesie przyszłość?

W świecie komputerów i sieci neuronowych

Waldemar Dylewski

Jak samodzielnie stworzyć prosty projekt z wykorzystaniem komputera neuronowego

10

10.1. Instrukcje krok po kroku do stworzenia prostego projektu

10.2. Rozwinięcie własnych pomysłów

Podsumowanie

11

11.1. Ważne punkty, które warto pamiętać
11.2. Inspiracja do dalszego eksplorowania świata komputerów neuronowych

Dodatki

12

12.1. Słownik pojęć
12.2. Źródła i zalecane materiały do dalszej nauki
12.3. Przykłady prostych projektów do samodzielnego wykonania

1 Wprowadzenie do komputerów

1.1. Czym jest komputer?



Komputer jest elektronicznym urządzeniem zdolnym do przetwarzania danych, wykonywania operacji logicznych oraz przechowywania, odczytywania i przekazywania informacji. Jest to uniwersalne urządzenie, które może wykonywać różnorodne zadania zgodnie z programami i instrukcjami, które są mu podawane. Oto kilka kluczowych cech komputera:

1.Przetwarzanie danych: komputer przetwarza dane, co oznacza, że wykonuje różnorodne operacje na danych wejściowych, generując wyniki na wyjściu. Operacje te obejmują obliczenia matematyczne, manipulacje tekstowe, analizę grafiki, itp.

2.Pamięć: posiada pamięć, która umożliwia przechowywanie danych tymczasowych lub trwałych. Dzięki pamięci komputer może przechowywać informacje potrzebne do wykonywania operacji.

3.Procesor: jest to jednostka centralna, która wykonuje instrukcje, zarządza operacjami i kontroluje przepływ danych wewnątrz komputera.

4.Wejście/Wyjście: komputer ma interfejsy wejścia i wyjścia, które umożliwiają mu komunikację z użytkownikiem lub innymi urządzeniami. Przykłady to klawiatura, mysz, monitor, głośniki, porty USB itp.

5. Oprogramowanie: komputer działa za pomocą oprogramowania, czyli zestawu instrukcji i procedur, które kontrolują jego działanie. Oprogramowanie to może być w postaci systemu operacyjnego, aplikacji, programów lub skryptów.

6. Uniwersalność: jedną z najważniejszych cech komputera jest jego zdolność do wykonywania różnych zadań w zależności od oprogramowania, które jest na niego załadowane. Może to być przetwarzanie tekstu, obliczenia matematyczne, projektowanie graficzne, analiza danych, gry komputerowe itp.

Komputery są nieodłączną częścią współczesnego życia, wykorzystywane w wielu dziedzinach od biznesu po naukę, rozrywkę czy medycynę. Ich wszechstronność i możliwości sprawiają, że są niezwykle ważnym narzędziem we współczesnym świecie.

Komputery odgrywają kluczową rolę we współczesnym świecie i mają ogromny wpływ na różne dziedziny życia.

W biznesie komputery są wykorzystywane do zarządzania bazami danych, analizy danych, księgowości, tworzenia raportów, planowania zasobów przedsiębiorstwa (ERP), a także do komunikacji biznesowej i handlu elektronicznego.

Komputery są niezbędne do analizy danych, symulacji numerycznych, modelowania matematycznego, badania genomiki, biologii molekularnej, fizyki, chemii, a także do prowadzenia zaawansowanych eksperymentów naukowych.

W sektorze rozrywki komputery są używane do tworzenia gier komputerowych, animacji, filmów, edycji dźwięku i obrazu, a także do strumieniowania treści wideo i audio.

Komputery są integralną częścią edukacji, wykorzystywane w nauczaniu zdalnym, dostarczaniu materiałów edukacyjnych, e-learningu, symulacjach edukacyjnych, tworzeniu interaktywnych lekcji itp.

W sektorze medycznym komputery są używane do analizy obrazów medycznych (np. tomografia komputerowa, rezonans magnetyczny), diagnozowania chorób, przetwarzania danych klinicznych, prowadzenia badań naukowych, a także do zarządzania informacjami pacjentów.

W przemyśle komputery są wykorzystywane do sterowania i monitorowania procesów produkcyjnych, automatyzacji, analizy danych dotyczących wydajności i jakości, a także do planowania zapasów i logistyki.

Komputery umożliwiają komunikację online, korzystanie z mediów społecznościowych, przesyłanie wiadomości e-mail, prowadzenie wideokonferencji, a także dostęp do informacji na temat wydarzeń na całym świecie.

Te przykłady tylko skrawek pokazują, jak szeroko komputery są wykorzystywane we współczesnym świecie. Ich wszechstronność, szybkość przetwarzania danych oraz możliwość rozwiązywania różnorodnych zadań sprawiają, że stanowią niezastąpione narzędzie w wielu dziedzinach życia i mają ogromny wpływ na rozwój społeczny, gospodarczy i technologiczny.

1 Wprowadzenie do komputerów

1.2. Dlaczego komputery są ważne?



Komputery są ważne z wielu powodów, które mają istotny wpływ na różne aspekty życia jednostek, społeczeństwa i gospodarki.

Komputery potrafią przetwarzać ogromne ilości danych w bardzo krótkim czasie, co umożliwia szybkie obliczenia, analizę danych, generowanie raportów i rozwiązywanie złożonych problemów matematycznych czy naukowych.

Dzięki automatyzacji wielu zadań, komputery pozwalają na zwiększenie efektywności pracy i redukcję błędów, co jest szczególnie ważne w wielu obszarach, od przemysłu po medycynę.

Komputery umożliwiają globalną komunikację, pozwalając ludziom na kontakt z całym światem w czasie rzeczywistym, dzięki internetowi, mediom społecznościowym i komunikatorom.

Komputery stanowią fundament dla wielu innowacji technologicznych. Wraz z postępowaniem w dziedzinie sztucznej inteligencji, uczenia maszynowego, robotyki, analizy danych i innych obszarów, komputery otwierają nowe możliwości dla rozwoju technologicznego.

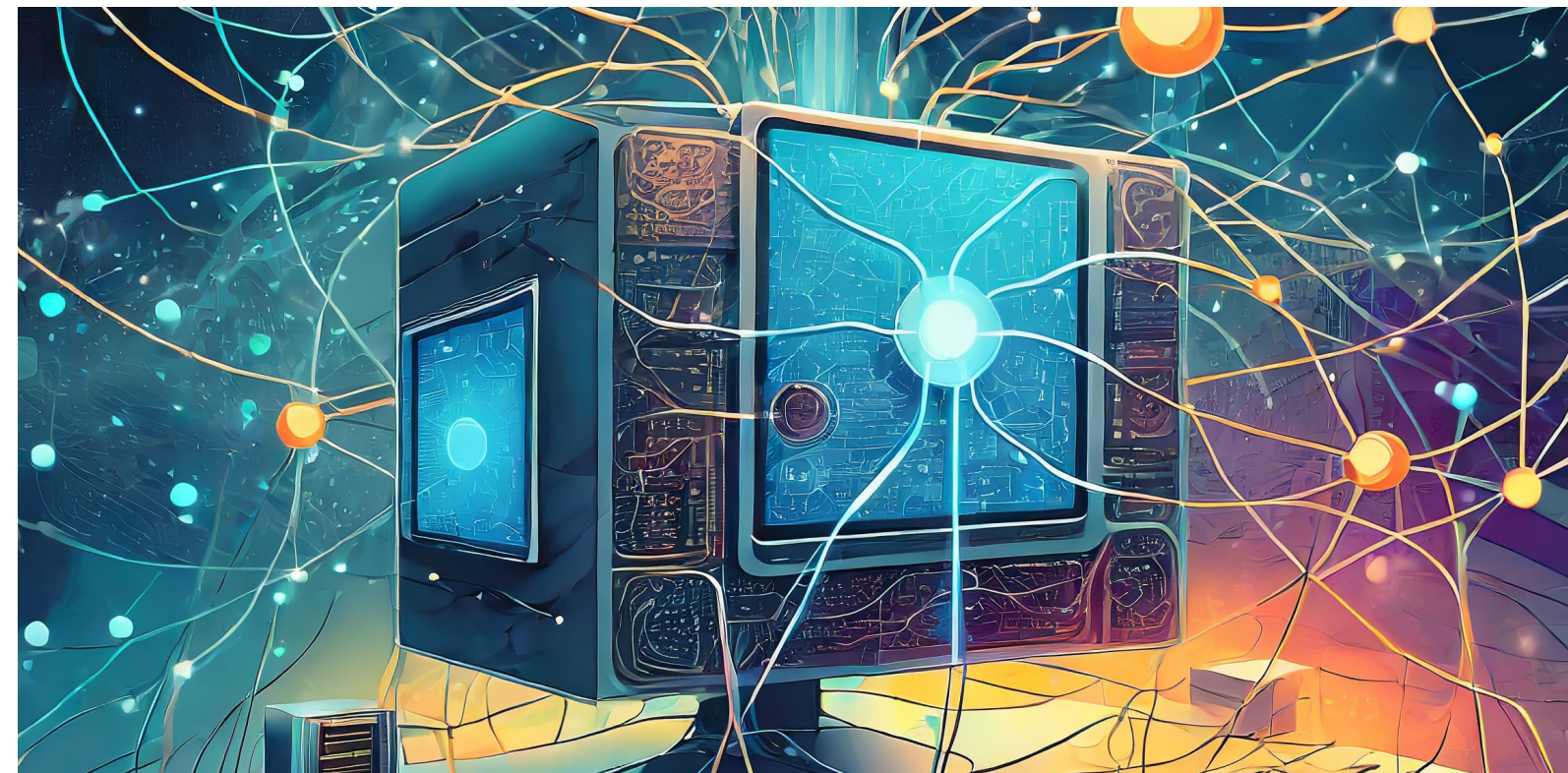
W obszarze edukacji komputery są niezbędne do dostępu do informacji, materiałów edukacyjnych, e-learningu, symulacji naukowych oraz eksperymentów, co ułatwia procesy nauki i kształcenia.

W biznesie komputery są kluczowe dla wielu procesów, od zarządzania bazami danych, księgowości, zarządzania projektami, po komunikację biznesową i handel elektroniczny.

Komputery mają duży wpływ na rozwój społeczny, tworząc nowe możliwości zatrudnienia, ułatwiając dostęp do informacji, poprawiając jakość życia ludzi i zapewniając narzędzia do rozwiązywania problemów społecznych.

W medycynie komputery są niezwykle ważne do analizy obrazów medycznych, diagnozowania chorób, badania genomiki, prowadzenia badań naukowych i zarządzania informacjami medycznymi.

Te powody to tylko fragment pełnego zakresu znaczenia komputerów. Ich wszechstronność, szybkość, możliwość przechowywania, przetwarzania i przekazywania informacji czynią je niezastąpionym narzędziem we współczesnym świecie, wpływając na wiele aspektów życia jednostek i społeczeństwa.





1 Wprowadzenie do komputerów

1.3. Historia komputerów

Historia komputerów jest fascynującą podróżą przez rozwój technologii, która sięga daleko wstecz, aż do starożytności.

Pierwsze kroki w kierunku komputacji zostały poczynione w starożytnym Egipcie i Mezopotamii, gdzie wykorzystywano proste narzędzia do wykonywania obliczeń arytmetycznych, takie jak kamienne płytki i kości liczące.

W okresie XVII – XIX wieku wynaleziono wiele mechanicznych urządzeń obliczeniowych, takich jak maszyna dodawcza Blaise'a Pascala (Pascaline) czy maszyna różnicowa Charles'a Babbage'a.

Rozwój technologiczny, szczególnie w zakresie elektroniki, doprowadził do powstania pierwszych elektronicznych maszyn obliczeniowych. Jednym z przełomowych wynalazków był "Automat tezaurusowy" Vanna M. Beziera, który był wczesnym przykładem programowalnego komputera.

Okres II wojny światowej przyniósł znaczący postęp w dziedzinie komputerów, głównie ze względu na zapotrzebowanie na obliczenia wojskowe. Komputer ENIAC (Electronic Numerical Integrator and Computer), uruchomiony w 1946 roku, był jednym z pierwszych elektronicznych uniwersalnych komputerów.

W latach 50. i 60. XX wieku pojawiły się innowacje, takie jak komputery UNIVAC (UNIVersal Automatic Computer), IBM 700, a później wprowadzono pierwsze języki programowania, takie jak Fortran czy COBOL.





1 Wprowadzenie do komputerów

1.3. Historia komputerów

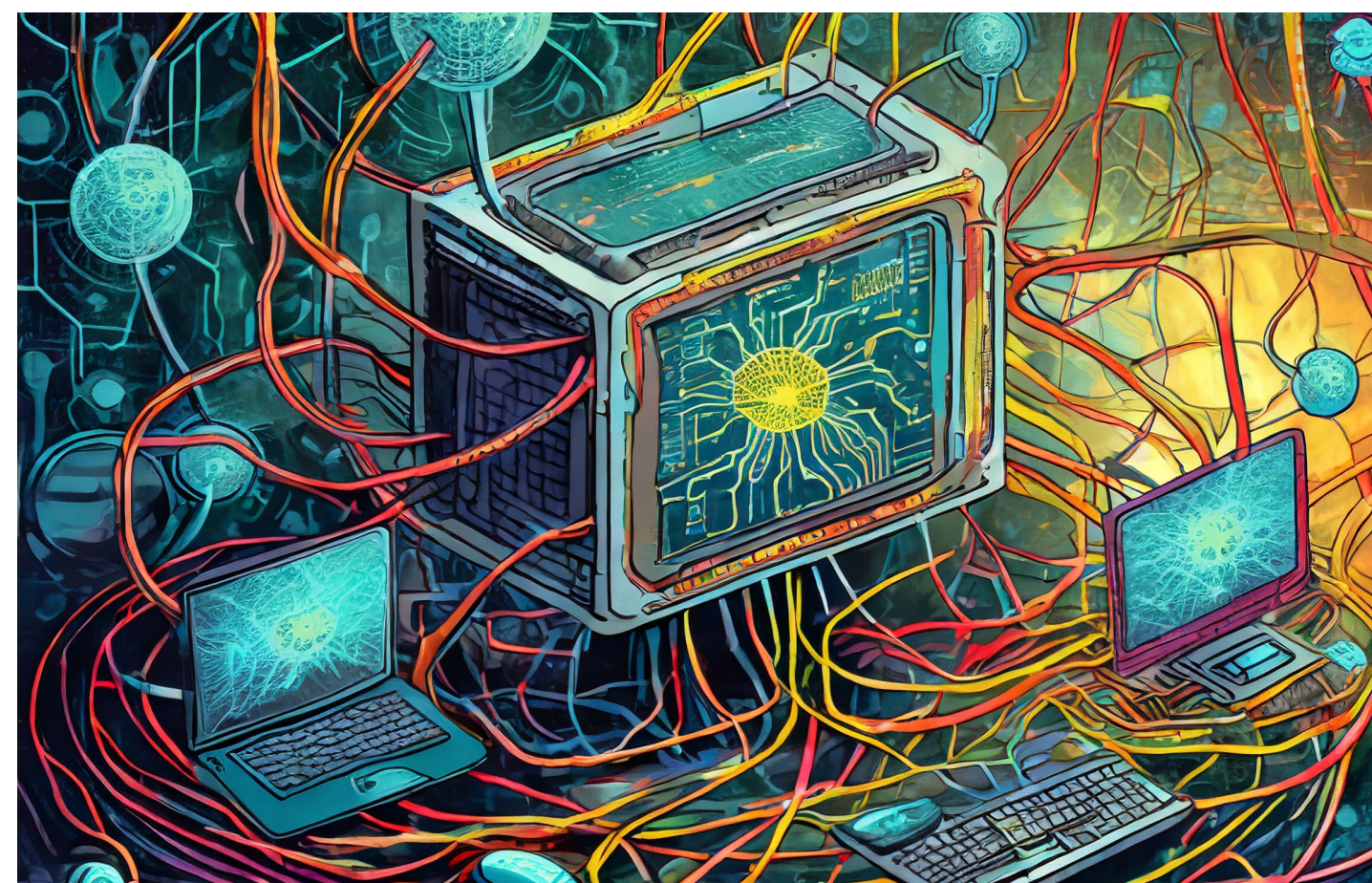
Lata 70. były okresem, w którym pojawiły się pierwsze mikroprocesory, co doprowadziło do rozwoju komputerów osobistych (PC). Apple I został wprowadzony na rynek w 1976 roku, a IBM PC w 1981 roku.

Od lat 80. do lat 90. XX wieku komputery osobiste stawały się coraz bardziej powszechne w gospodarstwach domowych i firmach, rozwijając się w szybkim tempie pod względem mocy obliczeniowej i funkcji.

W latach 90. Internet stał się powszechnie dostępny, co zapoczątkowało erę komunikacji online i przesyłania informacji na skalę globalną, co miało olbrzymi wpływ na rozwój komputerów i technologii.

Obecnie komputery są wszechobecne, od smartfonów i tabletów po superkomputery. Rozwój sztucznej inteligencji, uczenia maszynowego i sieci neuronowych to kolejne kroki w ewolucji komputerów.

Ta krótka historia pokazuje ewolucję komputerów od ich wczesnych, prymitywnych form do współczesnych zaawansowanych maszyn, które zmieniły sposób, w jaki żyjemy, pracujemy i komunikujemy się.





Nie samym komputerem żyje człowiek!

1 | Wprowadzenie do komputerów

1.4. Różne rodzaje komputerów (np. komputerów osobistych, tabletów, smartfonów)

Istnieje wiele różnych rodzajów komputerów, z których każdy ma swoje specyficzne cechy, zastosowania i funkcje. Oto kilka głównych typów komputerów:

1. Komputery osobiste (PC - Personal Computers): są to tradycyjne komputery, takie jak stacjonarne komputery biurowe i laptopy, które są przeznaczone do użytku osobistego. Są wykorzystywane do różnorodnych zadań, od pracy zawodowej po rozrywkę.

2. Tablety: są to przenośne komputery wyposażone w ekran dotykowy, które często działają na systemach operacyjnych mobilnych, takich jak iOS (Apple iPad) czy Android (tablety wielu producentów). Są popularne w zastosowaniach rozrywkowych, edukacyjnych i biznesowych.

3. Smartfony: są to komputery przenośne, które łączą funkcje telefonu komórkowego z możliwościami komputerowymi. Oprócz funkcji telefonii, smartfony obsługują różnorodne aplikacje, umożliwiają dostęp do internetu, obsługę poczty elektronicznej, media społecznościowe i wiele innych.

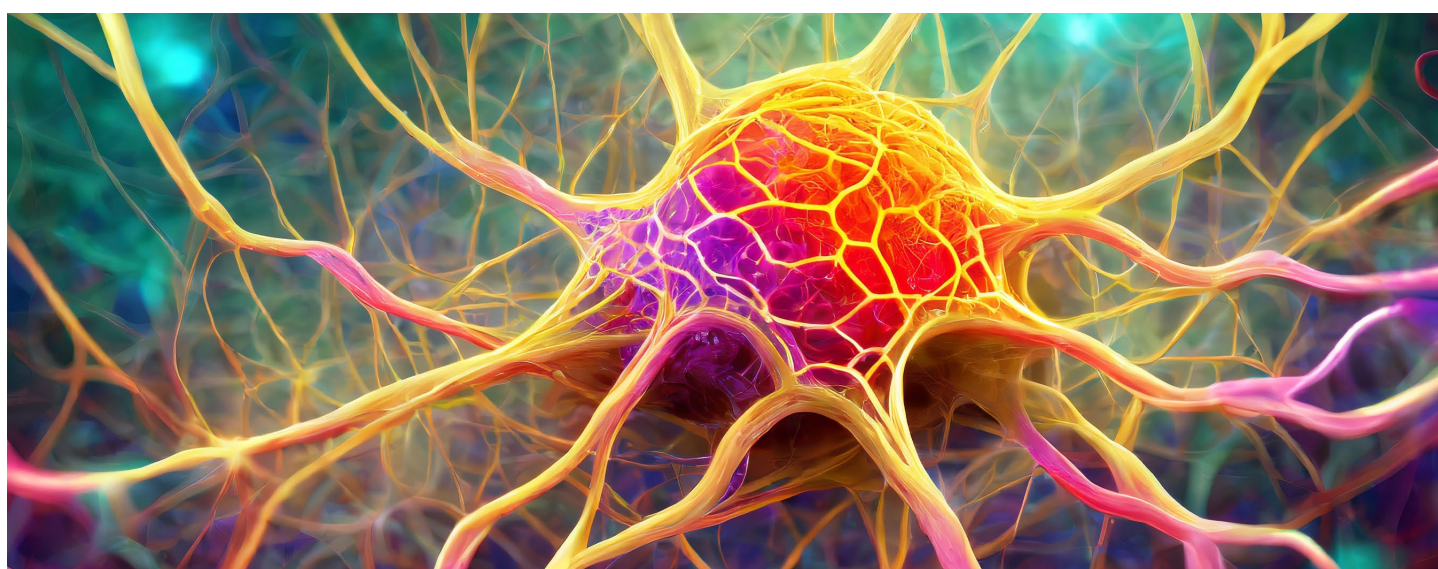
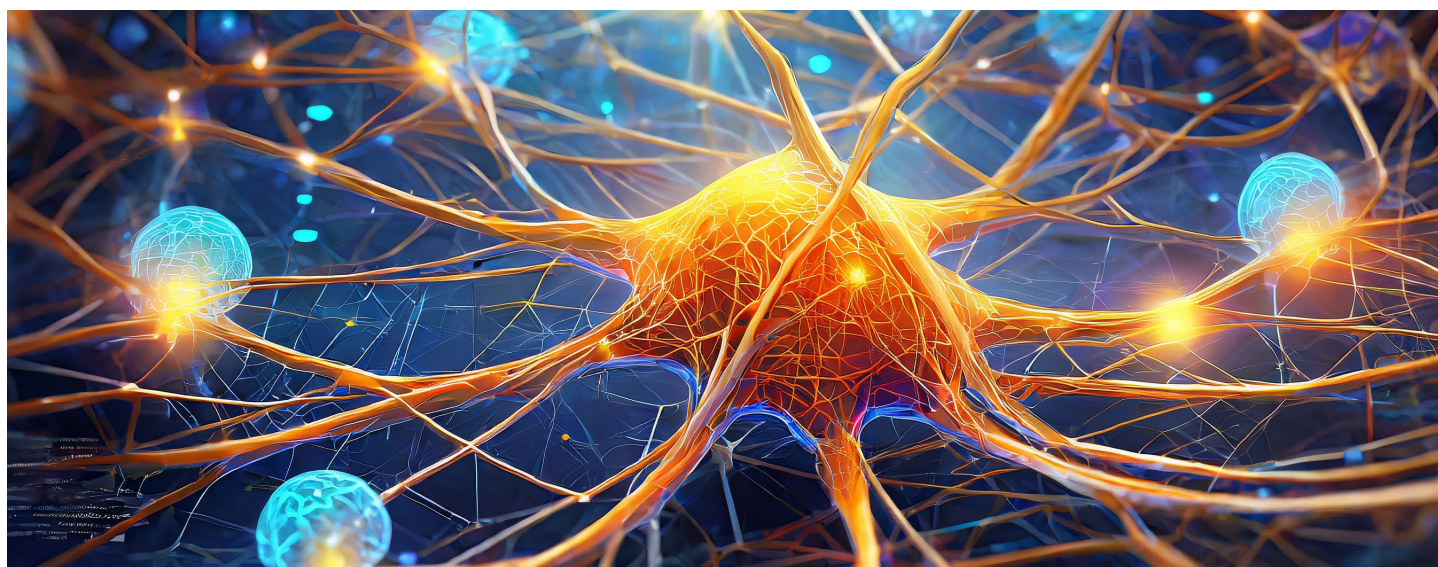
4. Serwery: to komputery, które są przeznaczone do obsługi wielu zadań jednocześnie, takich jak przechowywanie danych, hostowanie stron internetowych, obsługa usług sieciowych, zarządzanie siecią, udostępnianie zasobów dla innych komputerów w sieci, itp.

5. Superkomputery: są to najbardziej zaawansowane i wydajne komputery, które wykorzystywane są do rozwiązywania bardzo skomplikowanych zadań obliczeniowych, takich jak symulacje naukowe, badania meteorologiczne, projekty kosmiczne czy analizy dużych zbiorów danych.

6. Komputery wbudowane (Embedded Computers): są to komputery zintegrowane w urządzeniach, które nie są tradycyjnie postrzegane jako komputery, np. w samochodach, telewizorach, kamerach, smartwatchach, systemach nawigacyjnych, a nawet w niektórych urządzeniach medycznych.

7. Konsolowe systemy do gier: są to specjalnie zaprojektowane komputery przeznaczone do gier komputerowych, takie jak PlayStation, Xbox czy Nintendo Switch. Mają zaawansowane funkcje graficzne i interfejsy użytkownika przystosowane do gier.

Każdy rodzaj komputera ma swoje zastosowania, charakterystyczne cechy i przewagi w różnych dziedzinach. Rozwój technologii komputerowej przyczynia się do powstawania coraz to nowszych rodzajów i form komputerów, które zaspokajają potrzeby różnych użytkowników i branż.



2 | Jak działa komputer neuronowy?

2.1. Co to jest komputer neuronowy?

Termin „komputer neuronowy” jest zazwyczaj używany jako skrót od „sztucznych sieci neuronowych”. Sieci neuronowe są modelem matematycznym inspirowanym strukturą i działaniem ludzkiego mózgu. Są one używane do przetwarzania informacji, rozpoznawania wzorców, wykrywania zależności, klasyfikacji danych i podejmowania decyzji, podobnie jak funkcje wykonywane przez ludzki mózg.

Sztuczne sieci neuronowe składają się z połączonych ze sobą sztucznych neuronów (zwanymi również węzłami), które przetwarzają dane i wykonują obliczenia. Te sztuczne neurony organizowane są w warstwy, które składają się z warstw wejściowych, ukrytych (jeśli są obecne) i warstw wyjściowych.

Podstawowymi elementami w sieciach neuronowych są:

1.Neurony: sztuczne neurony lub węzły są podstawowymi jednostkami przetwarzania w sieciach neuronowych. Każdy neuron otrzymuje dane wejściowe, wykonuje na nich obliczenia zgodnie z pewnymi wagami i funkcją aktywacji, a następnie przekazuje wynik do kolejnych neuronów.

2.Wagi: każda połączona krawędź między neuronami ma przypisaną wagę. Wagi te decydują o wpływie danych wejściowych na wynik obliczeń przeprowadzonych przez neuron.

3.Funkcja aktywacji: jest to funkcja matematyczna, która decyduje, czy i w jakim

stopniu sygnał przekazywany jest z neuronu do kolejnych neuronów. Popularne funkcje aktywacji to np. sigmoidalna, ReLU (Rectified Linear Unit), tanh (tangens hiperboliczny), softmax itp.

Komputer neuronowy (sztuczna sieć neuronowa) może być stosowany w różnych dziedzinach, takich jak rozpoznawanie obrazów, przetwarzanie języka naturalnego, analiza danych, przewidywanie, klasyfikacja czy rozwiązywanie problemów związanych z uczeniem maszynowym. Te sieci są w stanie wykrywać zależności w danych, ucząc się na podstawie próbek i dostosowując swoje wagi, aby dokonywać lepszych predykcji czy klasyfikacji.

“

Gołata E., Informatyka, ekonometria i statystyka w społeczeństwie informacyjnym, Wydawnictwo Uniwersytetu Ekonomicznego, Poznań 2008

Sieci neuronowe należą do kategorii systemów uczących się. W odróżnieniu od klasycznych systemów informatycznych, w ich działaniu wyraźnie wyodrębnić można dwie fazy: fazę treningową oraz fazę reakcji na określony bodziec zewnętrzny. Sposób przeprowadzania procesu dostosowania wag nazywany jest algorytmem uczenia, a jego kolejne kroki polegają na modyfikacji parametrów sieci w taki sposób, aby optymalizować pewne, charakterystyczne dla danej metody uczenia, kryterium.

Podstawowym sposobem uczenia sieci jednokierunkowej jest algorytm wstecznej propagacji błędów. Na podstawie informacji o błędzie modyfikowane są wagi neuronu wyjściowego tak, aby zminimalizować pewną funkcję błędu.

(Gołata E. 2008, s. 113)

”

2 | Jak działa komputer neuronowy?

2.2. Porównanie z naszym mózgiem

Porównanie sztucznych sieci neuronowych z ludzkim mózgiem jest fascynującym zagadnieniem, ponieważ inspiracją do stworzenia tych sieci była biologiczna struktura i funkcje mózgu. Istnieją jednak istotne różnice między sztucznymi sieciami neuronowymi a ludzkim mózgiem:

Mózg ludzki jest niesamowicie złożonym i elastycznym systemem biologicznym. Zawiera miliardy neuronów, które są połączone w bardzo skomplikowany sposób. Sztuczne sieci neuronowe, chociaż rozwijają się i stają się coraz bardziej złożone, wciąż są znacznie mniej złożone niż ludzki mózg.

Ludzki mózg ma zdolność do ciągłego uczenia się, adaptacji i elastyczności, co oznacza, że może szybko reagować na nowe sytuacje i kontynuować uczenie się na przestrzeni całego życia. Sztuczne sieci neuronowe wymagają danych treningowych i algorytmów uczących, które nie są w stanie replikować takiej uniwersalności w procesie uczenia się.

Mózg ludzki jest bardzo efektywny energetycznie, zużywając relatywnie mało energii w porównaniu do swojej wydajności. Obecne sztuczne sieci neuronowe wymagają dużych zasobów obliczeniowych i energii w porównaniu do mózgu.

Choć sztuczne sieci neuronowe są zdolne do przetwarzania dużych ilości danych, mogą mieć problemy z elastycznością w analizie

abstrakcyjnych koncepcji, jakie potrafi zrozumieć i przetwarzać ludzki mózg.

Sieci neuronowe mają swoje zastosowania w wielu obszarach, takich jak rozpoznawanie obrazów, analiza danych, język naturalny itp., ale ich wydajność i zastosowanie w porównaniu z mózgiem ludzkim są wciąż ograniczone.

Mimo różnic między sztucznymi sieciami neuronowymi a ludzkim mózgiem, inspirowanie się biologiczną strukturą mózgu pozwoliło na rozwinięcie nowych metod i technologii w dziedzinie sztucznej inteligencji i uczenia maszynowego. Rozwój w tej dziedzinie nadal trwa, a naukowcy starają się coraz bardziej zbliżyć sieci neuronowe do elastyczności i wydajności mózgu ludzkiego.

Jak działa komputer neuronowy?

2.3. Neurony w komputerze neuronowym



W komputerach neuronowych, czyli sztucznych sieciach neuronowych, neurony są podstawowymi jednostkami przetwarzania informacji, na wzór neuronów biologicznych w ludzkim mózgu. Te sztuczne neurony są zazwyczaj reprezentowane jako węzły (najczęściej okręgi lub prostokąty) i są zorganizowane w warstwy, które przetwarzają dane wejściowe i generują wyniki na wyjściu.

Podstawowe elementy neurona w sztucznej sieci neuronowej to:

1. Wartości wejściowe: neuron otrzymuje sygnały wejściowe, które są przekazywane do niego z innych neuronów lub jako dane wejściowe do sieci.

2. Wagi: każde połączenie między neuronami ma przypisaną wagę. Wagi te decydują o wpływie danych wejściowych na wynik obliczeń przeprowadzonych przez neuron. Podczas procesu uczenia się sieć neuronowa dostosowuje wagi, aby lepiej dopasować się do danych treningowych i wykonywać lepsze predykcje.

“

Proces uczenia sieci neuronowej stanowi etap jej konstrukcji. Podczas tego procesu sieć zdobywa wiedzę dotyczącą określonego zjawiska (modelowanego obiektu) w oparciu o zbiór przykładów zawartych w ciągu uczącym. Etap ten polega na odpowiednim dostrojeniu wartości wag sieci, zgromadzonych w zbiorze W.

(Lula P., Stal J., Tadeusiewicz R., Morajda J., Paliwoda-Pękosz G., Wilusz W. 2012, s. 138)

”

3. Sumowanie ważonych wartości: neuron oblicza sumę wartości wejściowych pomnożonych przez odpowiadające im wagi. Jest to etap agregacji danych wejściowych z ich wagami.

4. Funkcja aktywacji: po obliczeniu sumy ważonych wartości, neuron stosuje funkcję aktywacji, która decyduje, czy i w jakim stopniu neuron ma aktywować się i przekazać sygnał do kolejnych neuronów. Funkcja aktywacji może wprowadzać nieliniowość do działania sieci, co pozwala na bardziej skomplikowane przetwarzanie informacji.

5. Wartość wyjściowa: po zastosowaniu funkcji

aktywacji neuron generuje swoją wartość wyjściową, która jest przekazywana do kolejnych neuronów w sieci lub jest wynikiem końcowym obliczeń sieci.

Neurony w sieciach neuronowych są wzajemnie połączone za pomocą warstw, które mogą być warstwami wejściowymi (przyjmującymi dane), ukrytymi (przetwarzającymi informacje) oraz wyjściowymi (generującymi końcowe wyniki). Te sztuczne neurony i ich połączenia są podstawowymi elementami, które umożliwiają sieciom neuronowym przetwarzanie informacji, ucząc się na podstawie danych treningowych i wykonywanie zróżnicowanych zadań, takich jak klasyfikacja, predykcja czy analiza danych.



Jedna z korporacji branży IT opracowała dyski twarde o nieskończonej pojemności. Mają trafić na rynek jak tylko skończą się formatować. 😊

2 | Jak działa komputer neuronowy?

2.4. Działanie komputera neuronowego

Działanie komputera neuronowego, czyli sztucznej sieci neuronowej, można opisać w kilku krokach, które obejmują przetwarzanie danych, uczenie się na podstawie danych treningowych i wykonanie konkretnego zadania. Oto ogólny schemat działania komputera neuronowego:

1. Przygotowanie danych: Dane wejściowe są przygotowywane do przetwarzania przez sieć neuronową. Mogą to być zdjęcia, tekst, dane numeryczne lub inne typy danych zależnie od zadania, które sieć ma wykonać.

2. Przekazywanie danych przez warstwy: dane wejściowe są przekazywane przez kolejne warstwy neuronów. Pierwsza warstwa to warstwa wejściowa, która przyjmuje dane, następnie dane te przetwarzane są przez warstwy ukryte (jeśli są obecne) i końcowo przez warstwę wyjściową, która generuje wynik.

3. Obliczenia w neuronach: każdy neuron przetwarza otrzymane dane wejściowe, sumując ich wartości pomnożone przez odpowiednie wagi. Następnie stosuje funkcję aktywacji, aby wygenerować swoją wartość wyjściową.

4. Uczenie się (trening): sieć neuronowa jest trenowana na podstawie danych treningowych. W procesie uczenia sieć dostosowuje wagi w połączeniach między neuronami, aby minimalizować błąd między przewidywanymi wynikami a rzeczywistymi wynikami. To jest etap, w którym sieć dostosowuje swoje parametry, aby lepiej dopasować się do charakterystyk danych.

5. Generowanie wyniku: po procesie uczenia sieć jest gotowa do generowania wyników dla nowych danych wejściowych. Działanie sieci

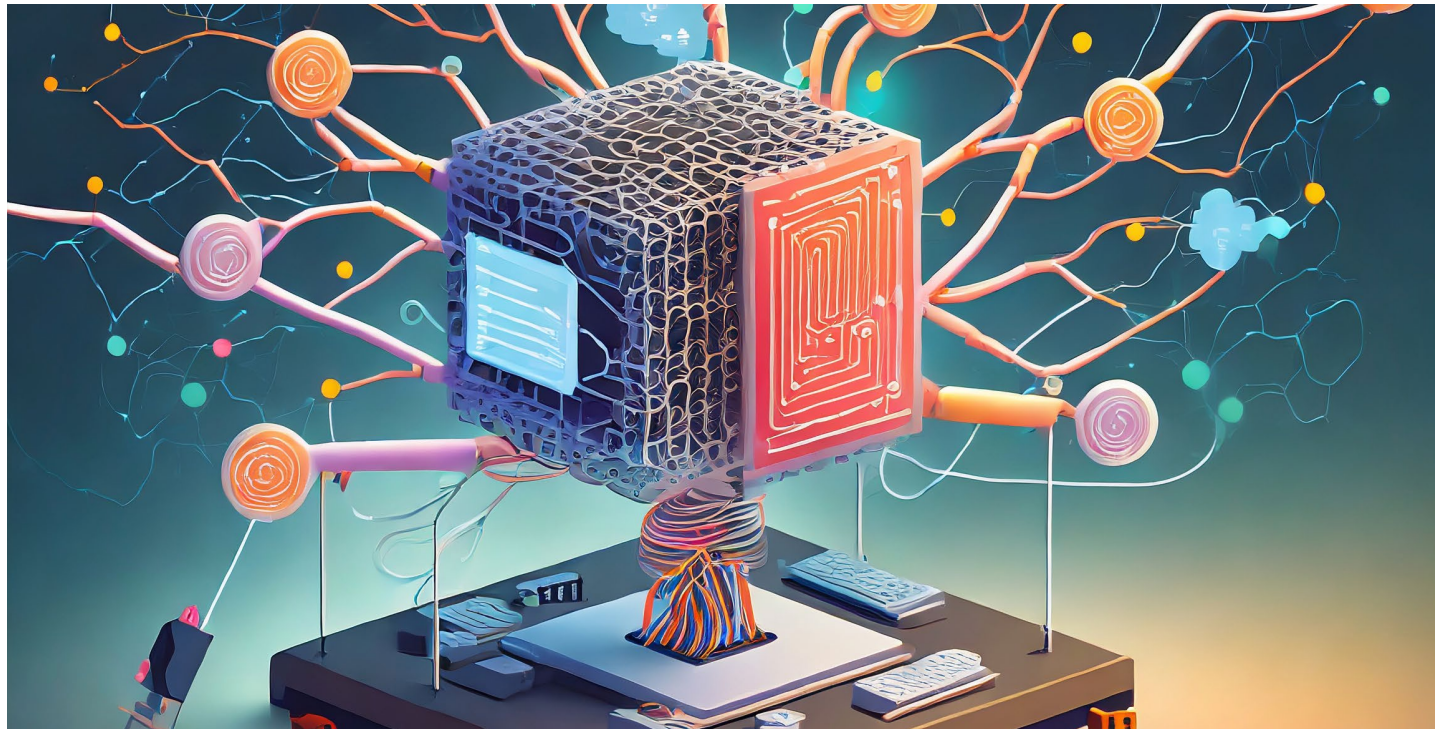
polega na przekazywaniu danych przez węzły i warstwy, aż do wygenerowania końcowego wyniku na wyjściu.

6. Ewaluacja wyników: wyniki wygenerowane przez sieć są oceniane pod kątem skuteczności i dokładności w wykonaniu określonego zadania. To pozwala na ocenę jakości działania sieci i ewentualne wprowadzanie poprawek lub optymalizacji.

Proces działania komputera neuronowego jest oparty na przetwarzaniu danych przez połączone sztuczne neurony, gdzie wagi między nimi są dostosowywane w procesie uczenia. To pozwala sieci na analizę danych, wykrywanie wzorców i podejmowanie decyzji na podstawie przetwarzanych informacji.

3 Tworzenie prostego komputera neuronowego

3.1. Jak zbudować prosty komputer neuronowy



Aby zbudować prosty komputer neuronowy, należy skorzystać z gotowych bibliotek do uczenia maszynowego dostępnych w różnych językach programowania, które oferują narzędzia do tworzenia i trenowania sieci neuronowych. Popularne biblioteki takie jak TensorFlow, Keras, PyTorch czy Scikit-learn oferują interfejsy programistyczne (API), które ułatwiają konstrukcję i trenowanie sieci neuronowych.

Ogólny sposób tworzenia komputera neuronowego wygląda następująco:

1. Wybierz język programowania, z którym się najlepiej czujesz. Python jest często używany ze względu na szeroką gamę bibliotek do uczenia maszynowego.
2. Zainstaluj biblioteki do uczenia maszynowego, takie jak TensorFlow, Keras, PyTorch lub Scikit-learn, w zależności od preferencji.
3. Wybierz typ sieci neuronowej (np. sieć gęsta, konwolucyjna, rekurencyjna), określ liczbę warstw i neuronów w każdej warstwie oraz funkcje aktywacji.

4. Przygotuj dane treningowe i testowe. Może to obejmować czyszczenie, normalizację, kodowanie danych, podział na zestawy treningowe i walidacyjne.

5. Wykonaj proces uczenia, dostosowując wagi w sieci, aby minimalizować błąd predykcji. W procesie tym wykorzystaj odpowiednie algorytmy optymalizacyjne i funkcje straty.

6. Po zakończeniu procesu trenowania ocen jakość sieci neuronowej na podstawie zestawu walidacyjnego. Sprawdź jej wydajność na danych testowych, aby ocenić jej skuteczność.

To ogólny zarys procesu, który obejmuje budowanie i trenowanie prostego komputera neuronowego. W praktyce szczegóły mogą być bardziej złożone, ale korzystanie z gotowych bibliotek do uczenia maszynowego znacznie ułatwia budowę i eksperymentowanie z sieciami neuronowymi. Oprócz tego istnieją różne kursy online, podręczniki i materiały edukacyjne, które mogą pomóc w zrozumieniu procesu tworzenia komputerów neuronowych od podstaw.

Oto krótki opis popularnych bibliotek do uczenia maszynowego i sieci neuronowych:

TensorFlow: jest to otwarta platforma do uczenia maszynowego opracowana przez Google. Znana jest ze swojej elastyczności i skalowalności, która umożliwia tworzenie różnych rodzajów sieci neuronowych. TensorFlow oferuje wsparcie dla budowy zarówno prostych, jak i skomplikowanych

modeli uczenia maszynowego.

Keras: jest to wysokopoziomowy interfejs programistyczny (API) dla sieci neuronowych, który działa na bazie TensorFlow, ale jest także wspierany przez inne platformy takie jak Microsoft Cognitive Toolkit czy Theano. Keras jest łatwy w użyciu i umożliwia szybkie prototypowanie modeli, dzięki czemu jest często wybierany przez początkujących.

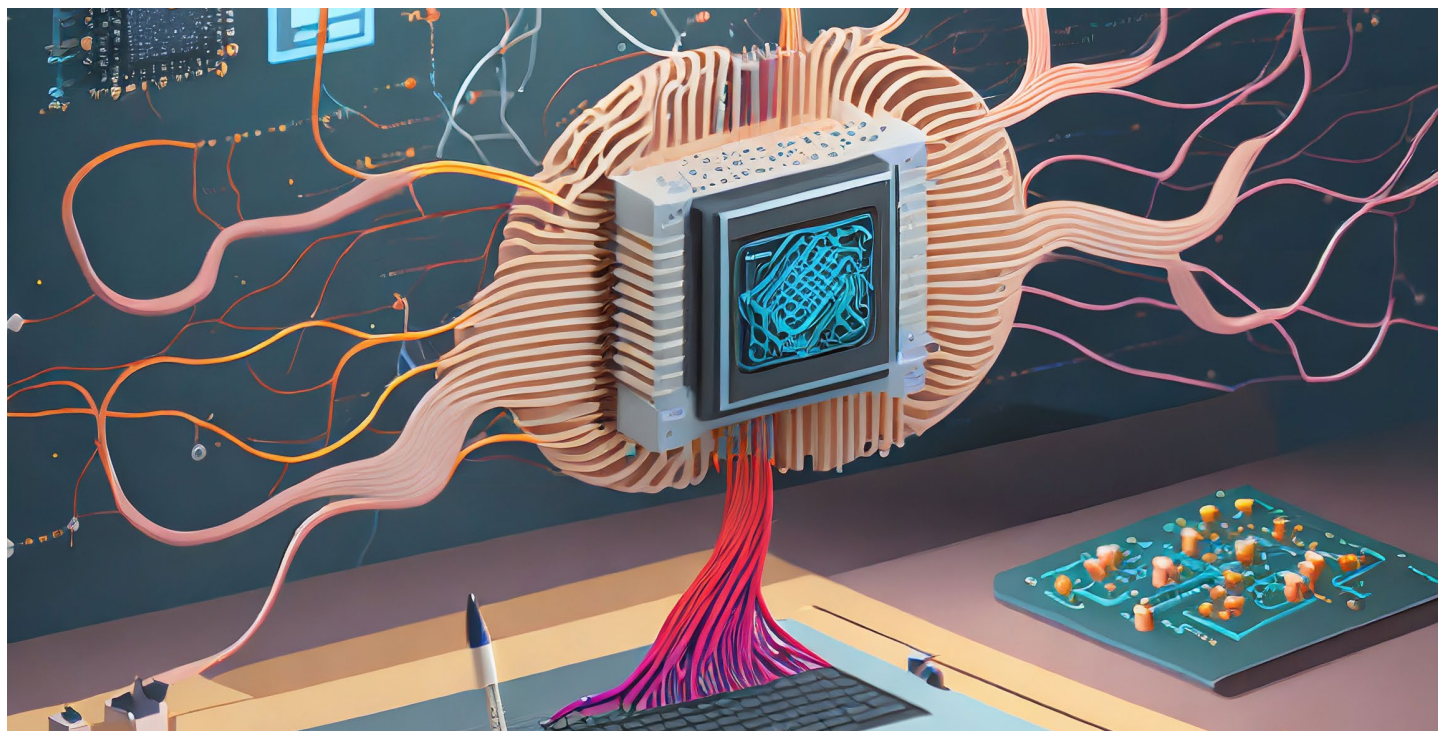
PyTorch: jest to kolejna popularna platforma do uczenia maszynowego opracowana przez Facebook. Wyróżnia się elastycznością i łatwością obsługi, co sprawia, że jest popularny wśród badaczy naukowych. PyTorch pozwala na dynamiczne tworzenie grafów obliczeniowych, co ułatwia eksperymentowanie z modelem.

Scikit-learn: to biblioteka do uczenia maszynowego dla języka Python. Znana jest z prostoty i łatwości obsługi, szczególnie dla osób, które dopiero zaczynają pracę z uczeniem maszynowym. Scikit-learn oferuje zestaw narzędzi do klasyfikacji, regresji, grupowania, redukcji wymiarowości, selekcji cech i wiele innych.

Te biblioteki oferują szeroki zakres narzędzi i funkcji, które ułatwiają pracę z uczeniem maszynowym i sieciami neuronowymi. Każda z nich ma swoje zalety i charakterystyczne cechy, które można wykorzystać w zależności od konkretnych potrzeb projektu czy preferencji programistycznych.

3 Tworzenie prostego komputera neuronowego

3.2. Użycie bloków i kabli



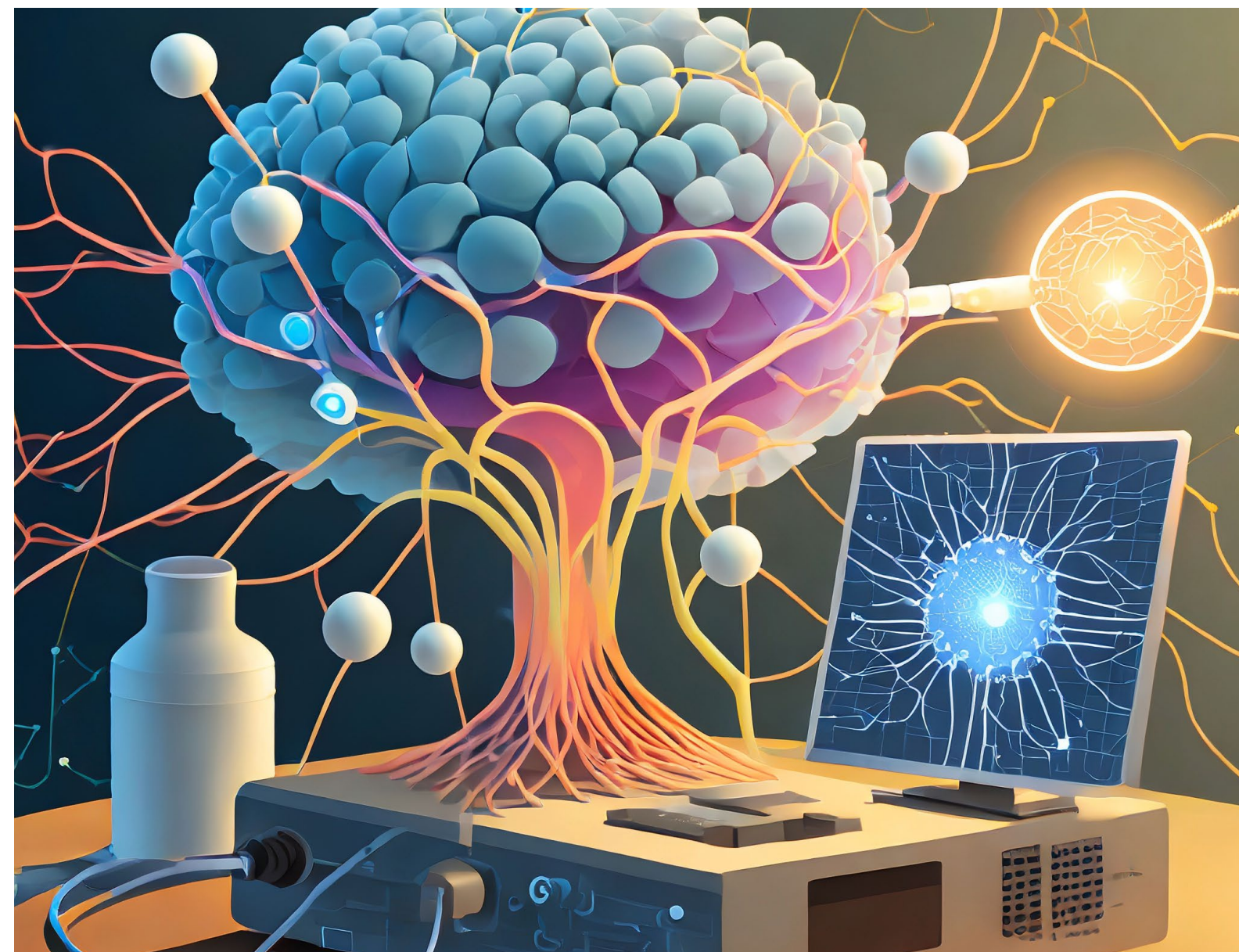
W kontekście sieci neuronowych, bloki i kable odnoszą się zazwyczaj do struktury i połączeń między neuronami w architekturze sieci.

1. Bloki: bloki w sieciach neuronowych odnoszą się do warstw lub grup neuronów, które wykonują określone obliczenia. Na przykład, w sieciach konwolucyjnych (CNN), blok może obejmować warstwy konwolucyjne, normalizacyjne, aktywacyjne i warstw poolingowych. W sieciach gęstych (fully connected networks), blokiem może być pojedyncza warstwa zawierająca zestaw neuronów połączonych z każdym neuronem z poprzedniej warstwy.

2. Kable: kable odnoszą się do połączeń między neuronami w sieci. Każdy neuron w warstwie jest połączony z neuronami w kolejnej warstwie za pomocą tych połączeń, które posiadają wagi. W procesie uczenia się sieć dostosowuje wagi tych połączeń (kable), aby lepiej dopasować się do charakterystyk danych wejściowych i wykonywać lepsze predykcje.

Podczas procesu projektowania sieci neuronowej, architektura składa się z bloków, które są ze sobą połączone poprzez kable. Przykładowo, w sieciach neuronowych stosowanych w zadaniach klasyfikacji obrazów, bloki mogą składać się z warstw konwolucyjnych, warstw poolingowych, warstw normalizacyjnych i warstw w pełni połączonych (fully connected layers). Połączenia między tymi blokami (kable) reprezentują przepływ danych i informacji w sieci, które są przetwarzane w celu wygenerowania ostatecznych wyników na wyjściu sieci.

Dobór odpowiednich bloków i kabli, ich liczba, typy warstw i sposób połączeń między nimi są kluczowe dla skuteczności sieci neuronowej w wykonywaniu określonych zadań. Optymalizacja architektury sieci, w tym doboru bloków i kabli, jest ważnym procesem podczas tworzenia efektywnych modeli uczenia maszynowego.





3 Tworzenie prostego komputera neuronowego

3.3. Programowanie komputera neuronowego

Programowanie komputera neuronowego polega na tworzeniu, konfigurowaniu i trenowaniu sieci neuronowych przy użyciu odpowiednich narzędzi programistycznych i bibliotek do uczenia maszynowego.

Ogólny proces programowania komputera neuronowego obejmuje następujące kroki:

1. Wybór biblioteki lub frameworka do uczenia maszynowego zależy od celów projektu i preferencji programistycznych. Na przykład, TensorFlow, PyTorch, Keras, Scikit-learn, czy też inne biblioteki oferują szeroki zakres funkcji i narzędzi do pracy z sieciami neuronowymi.
2. Następnie należy zdefiniować architekturę sieci neuronowej, co obejmuje wybór rodzaju sieci (np. CNN, RNN, MLP), liczbę warstw, typy warstw (np. konwolucyjne, poolingowe, w pełni połączone) oraz funkcje aktywacji.
3. Dane treningowe muszą być odpowiednio przygotowane do pracy z modelem. To może obejmować skalowanie, normalizację, kodowanie danych kategorycznych oraz podział danych na zestawy treningowe, walidacyjne i testowe.

4. Stworzenie instancji modelu sieci neuronowej na podstawie zdefiniowanej architektury. Należy zainicjować odpowiednie parametry, wagi i funkcje aktywacji w sieci.

5. Proces trenowania polega na dostosowywaniu wag połączeń między neuronami w sieci, aby model lepiej dopasował się do danych treningowych i minimalizował błąd predykcji. Podczas trenowania stosuje się odpowiednie algorytmy optymalizacyjne i funkcje straty.

6. Po zakończeniu procesu trenowania warto ocenić wydajność modelu na zestawie walidacyjnym lub testowym. Ocena modelu obejmuje analizę miar wydajności, takich jak dokładność, precyzja, odsetek fałszywych pozytywów itp.

7. Na podstawie wyników oceny modelu można dostosować parametry sieci, takie jak architektura, liczba warstw, liczba neuronów czy funkcje aktywacji, aby poprawić wydajność modelu.

Programowanie komputera neuronowego jest procesem iteracyjnym, który wymaga eksperymentowania, testowania różnych konfiguracji i optymalizacji modelu w celu uzyskania najlepszych wyników. Wsparcie programistyczne oferowane przez biblioteki do uczenia maszynowego znacznie ułatwia tworzenie, trenowanie i testowanie sieci neuronowych.





4 Sieci neuronowe

4.1. Czym są sieci neuronowe?

Sieci neuronowe są modelem matematycznym, który naśladuje strukturę i działanie ludzkiego mózgu. Są one używane do przetwarzania informacji, uczenia się na podstawie danych, rozpoznawania wzorców i podejmowania decyzji, podobnie jak to robi mózg.

Na ogół sieć neuronowa składa się z połączonych ze sobą sztucznych neuronów (lub węzłów), które przetwarzają dane i wykonują obliczenia. Te sztuczne neurony są zorganizowane w warstwy, które przekazują dane wejściowe i generują odpowiedzi na wyjściu.

Elementy składowe sieci neuronowych:

Neurony: sztuczne neurony to podstawowe jednostki przetwarzania w sieciach neuronowych. Każdy neuron otrzymuje dane wejściowe, wykonuje na nich obliczenia zgodnie z pewnymi wagami i funkcją aktywacji, a następnie przekazuje wynik do kolejnych neuronów.

Wagi: wagi to parametry, które decydują o znaczeniu danych wejściowych dla neuronu. Każde połączenie między neuronami ma przypisaną wagę, która reguluje wpływ danych wejściowych na obliczenia wykonywane przez neuron.

Funkcja aktywacji: jest to funkcja matematyczna, która decyduje, czy i w jakim stopniu sygnał przekazywany jest z neuronu do kolejnych neuronów. Funkcje aktywacji wprowadzają nieliniowość do sieci, co umożliwia bardziej skomplikowane przetwarzanie informacji.

Warstwy: sieci neuronowe składają się z różnych warstw: warstw wejściowych, warstw ukrytych (jeśli są obecne) i warstw wyjściowych. Warstwy te są połączone ze sobą i realizują różne etapy przetwarzania danych.

Sieci neuronowe są używane w różnych dziedzinach, takich jak rozpoznawanie obrazów, przetwarzanie języka naturalnego, analiza danych, rozwiązywanie problemów w uczeniu maszynowym, przewidywanie, klasyfikacja i wiele innych. Są w stanie wykrywać zależności w danych, uczyć się na podstawie próbek i dostosowując swoje wagi, aby dokonywać lepszych predykcji czy klasyfikacji.





Bez biologii ani rusz!

4 | Sieci neuronowe

4.2. Przykłady zastosowań sieci neuronowych

Sieci neuronowe znajdują zastosowanie w wielu obszarach ze względu na swoją zdolność do wykrywania wzorców, uczenia się na podstawie danych oraz podejmowania decyzji.

Sieci neuronowe, zwłaszcza sieci konwolucyjne (CNN), są używane do rozpoznawania obiektów na zdjęciach i wideo. Znane są z wykorzystania w aplikacjach do rozpoznawania twarzy, rozpoznawania znaków drogowych w samochodach autonomicznych, diagnozowania chorób na zdjęciach medycznych itp.

W obszarze NLP sieci neuronowe są stosowane do tłumaczenia maszynowego, analizy nastroju w mediach społecznościowych, generowania tekstu, analizy sentymentu, przetwarzania mowy oraz tworzenia chatbotów i asystentów głosowych.

Sieci neuronowe są wykorzystywane do systemów rekomendacyjnych w serwisach streamingowych, sklepach internetowych i platformach społecznościowych. Dzięki analizie zachowań użytkowników potrafią one proponować spersonalizowane sugestie produktów, muzyki, filmów czy treści.

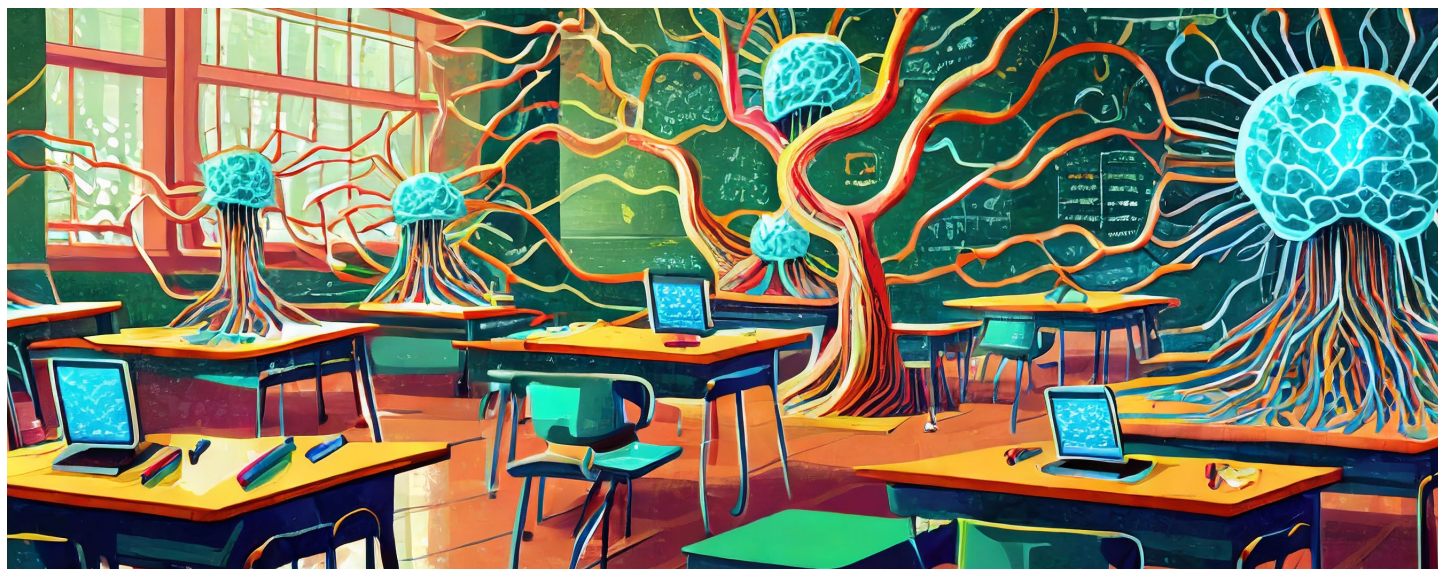
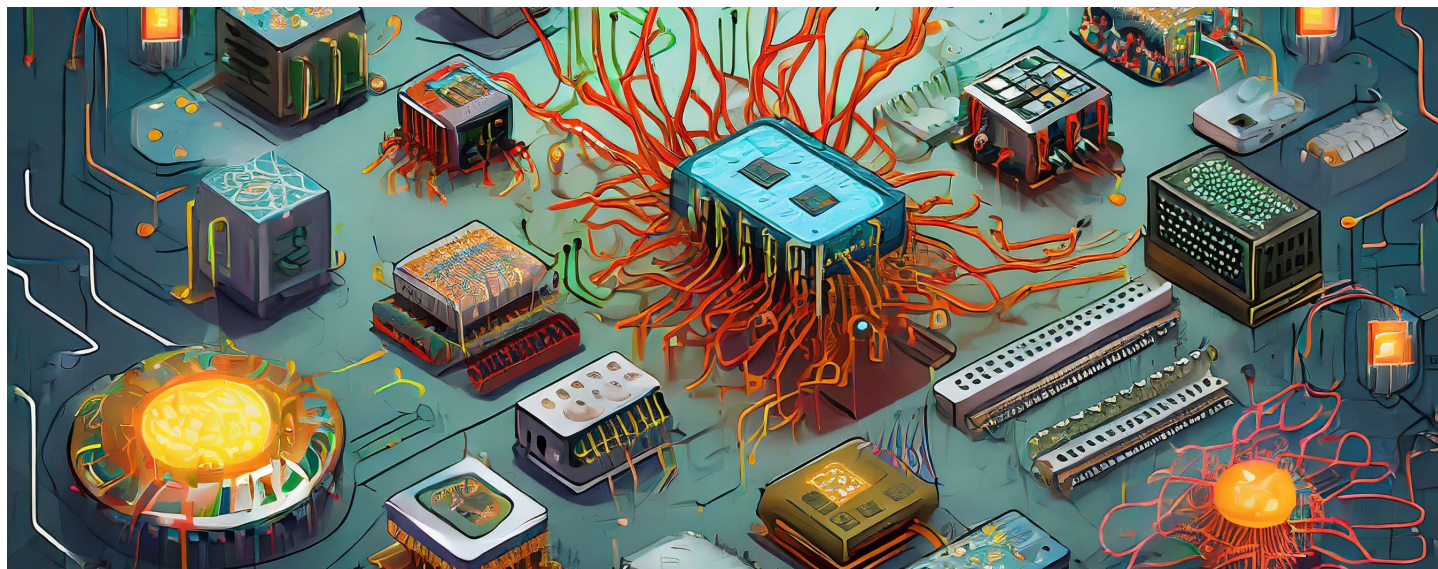
Sieci neuronowe pomagają w diagnozowaniu chorób na podstawie obrazów medycznych, analizie danych klinicznych, prognozowaniu rozwoju chorób, a także w identyfikacji nowych leków i odkrywaniu wzorców w danych biomedycznych.

W technologiach autonomicznych, sieci neuronowe są wykorzystywane do rozpoznawania drogi, innych pojazdów, znaków drogowych i kierowanie ruchem

w pojazdach autonomicznych.

W finansach sieci neuronowe pomagają w prognozowaniu cen akcji, analizie ryzyka, wykrywaniu nadużyć finansowych, analizie rynków i prognozowaniu trendów ekonomicznych.

Te zastosowania to tylko część obszarów, w których sieci neuronowe znajdują zastosowanie. Ich uniwersalność i zdolność do adaptacji do różnych rodzajów danych sprawia, że są one kluczowym narzędziem w dziedzinie sztucznej inteligencji i uczenia maszynowego.



4 | Sieci neuronowe

4.3. Jak sieci neuronowe uczą się

Sieci neuronowe uczą się poprzez proces adaptacji wag połączeń między neuronami w odpowiedzi na prezentowane dane treningowe. Proces ten polega na minimalizowaniu błędu predykcji przez sieć, aż do osiągnięcia zadowalającej wydajności.

Oto główne etapy procesu uczenia się sieci neuronowych:

1. Inicjalizacja wag: na początku, wagi połączeń między neuronami są zazwyczaj inicjowane losowo lub za pomocą określonych technik, takich jak He initialization czy Xavier initialization.

2. Przekazywanie danych treningowych: dane treningowe, które składają się z par wejście-wyjście, są prezentowane sieci neuronowej.

3. Propagacja w przód (forward propagation): dane wejściowe są przekazywane przez sieć, a każdy neuron przetwarza dane zgodnie z wagami połączeń. Sieć generuje wynik na wyjściu.

4. Obliczenie błędu: porównuje się uzyskane wyniki z oczekiwanymi wartościami (znanymi jako etykiety lub oznaczenia) w danych treningowych, co pozwala na określenie błędu predykcji.

5. Propagacja wsteczna (backpropagation): błąd wyznaczony w kroku 4 jest wstecznie propagowany przez sieć, aby określić, jak zmienić wagi połączeń w celu minimalizacji błędu. Wykorzystuje się zasadę gradientu prostego (gradient descent), aby dostosować wagi w taki sposób, aby zmniejszyć błąd predykcji.

6. Aktualizacja wag: wagi połączeń są dostosowywane na podstawie obliczonych gradientów, aby minimalizować funkcję kosztu (błąd predykcji). Zwykle stosuje się różne

warianty algorytmów optymalizacyjnych, takie jak SGD (Stochastic Gradient Descent), Adam, RMSprop itp., aby zoptymalizować proces aktualizacji wag.

7. Iteracyjny proces: proces propagacji w przód i wstecznych poprawek wag jest powtarzany wielokrotnie (przez wiele epok) aż do osiągnięcia satysfakcjonującej wydajności modelu.

Poprzez iteracyjne dostosowywanie wag na podstawie danych treningowych, sieci neuronowe uczą się zależności między danymi wejściowymi a oczekiwanymi wynikami wyjściowymi. W efekcie sieć staje się coraz lepiej przystosowana do wykonywania konkretnego zadania, co nazywane jest procesem uczenia się.

“

Sieci neuronowe są metodami obliczeniowymi, których działanie inspirowane jest sposobem funkcjonowania układów nerwowych organizmów żywych. Podobnie jak układ nerwowy składa się z komórek nerwowych zwanych neuronami, tak również wykorzystywane w analizie danych sztuczne sieci neuronowe składają się z elementów przetwarzających nazywanych sztucznymi neuronami.

(Lula P., Stal J., Tadeusiewicz R., Morajda J., Paliwoda-Pękosz G., Wilusz W. 2012, s. 92)

”

5 | Wizualizacja sieci neuronowych

5.1. Jak wygląda sieć neuronowa?

Wygląd sieci neuronowej zależy od jej architektury, czyli sposobu połączenia neuronów i ich układu w warstwy. Najczęściej spotykanymi typami architektur sieci neuronowych są:

1. Sieć neuronowa w pełni połączona (Fully Connected Neural Network):

w takiej sieci każdy neuron w jednej warstwie jest połączony z każdym neuronem w kolejnej warstwie. Wizualnie taka sieć może być przedstawiona jako zestaw warstw, w których neurony są połączone z neuronami w kolejnych warstwach za pomocą linii reprezentujących wagi połączeń.

2. Sieć konwolucyjna (Convolutional Neural Network - CNN):

CNN charakteryzują warstwy konwolucyjne, poolingowe i w pełni połączone. Warstwy konwolucyjne są reprezentowane przez zestawy filtrów, które przesuwane są po obrazie w celu wykrywania cech. Warstwy poolingowe redukują rozmiar przestrzenny map cech. Warstwy w pełni połączone występują na końcu sieci. Wizualizacja CNN może przedstawiać się jako zestaw warstw z filtrami i mapami cech.

W praktyce, wizualizacja sieci neuronowej często polega na przedstawieniu grafu, który pokazuje architekturę sieci w sposób graficzny. Graf ten wykazuje warstwy sieci i połączenia między neuronami, które mogą być reprezentowane jako strzałki z wagami między warstwami. Wizualizacja taka może być złożona i zależy od liczby warstw, rodzaju połączeń oraz rozmiaru sieci.

3. Rekurencyjne sieci neuronowe (Recurrent Neural Network - RNN):

RNN posiadają pętle zwrotne, które pozwalają na przetwarzanie sekwencji danych. Ich wizualizacja może pokazywać strukturę warstw rekurencyjnych połączonych w sposób sekwencyjny.

Ponadto, dostępne są różne narzędzia, biblioteki i oprogramowanie, które umożliwiają wizualizację sieci neuronowych, pomagając zrozumieć ich strukturę i sposób działania.



Diagramy i rysunki przedstawiające sieci neuronowe są używane do wizualizacji architektury oraz połączeń między neuronami w sieci. Oto kilka typowych sposobów przedstawiania sieci neuronowych za pomocą diagramów:

Warstwowe diagramy sieci neuronowej: to najczęstszy sposób przedstawienia architektury sieci. W tym rodzaju diagramów neurony w poszczególnych warstwach są reprezentowane jako prostokąty lub okręgi, a połączenia między warstwami są zazwyczaj reprezentowane za pomocą strzałek. Wagi połączeń między neuronami mogą być również uwzględnione jako etykiety na strzałkach.

Schematy blokowe (Block diagrams): przedstawiające warstwy lub bloki w sieci neuronowej jako prostokąty lub figury geometryczne. Każdy blok reprezentuje warstwę lub grupę neuronów, a linie łączące bloki pokazują przepływ danych między nimi.

“

Jedną z pierwszych prób zastosowania sztucznych sieci neuronowych do przewidywania bankructwa firm podjęli M. Odom i R. Sharda (1990). W swoich badaniach zastosowali 5 wskaźników finansowych z klasycznego modelu Altmana, które zostały użyte zarówno do budowy modelu opartego na sztucznych sieciach neuronowych, jak i dla modelu dyskryminacyjnego.

(Hołda A. 2006, s. 149)

”

Schematy konwolucyjnej sieci neuronowej: dla sieci konwolucyjnych (CNN), diagramy te zwykle zawierają warstwy konwolucyjne, warstwy poolingowe i warstwy w pełni połączone. Filtry konwolucyjne są reprezentowane jako macierze, a linie łączące warstwy pokazują przepływ danych.

Grafy połączeń neuronów: to szczegółowe diagramy pokazujące wszystkie połączenia między poszczególnymi neuronami w sieci. Te grafy mogą być szczególnie skomplikowane w dużych sieciach, ale są użyteczne do zrozumienia połączeń i przepływu danych w sieci.

Wizualizacja wag: dodatkowo, wizualizacja wag połączeń między neuronami może być reprezentowana za pomocą map cieplnych, histogramów, lub innych technik w celu zobrazowania ich znaczenia i wpływu na funkcjonowanie sieci.

Te diagramy są używane do lepszego zrozumienia struktury, połączeń między warstwami i działania sieci neuronowych. Wybór konkretnego rodzaju diagramu zależy od złożoności sieci oraz celu, jakim jest komunikowanie struktury i funkcjonowania sieci w sposób klarowny i zrozumiały.



Czy człowiek jest jeszcze potrzebny?

6 | Jak sieci neuronowe rozwiązują problemy

6.1. Przykłady, jak sieci neuronowe rozwiązują problemy (np. rozpoznawanie obrazów, tłumaczenie języka)

Oto kilka przykładów, jak sieci neuronowe rozwiązują różne problemy:

1. Rozpoznawanie obrazów:

Konwolucyjne sieci neuronowe (CNN): Wykorzystywane są do klasyfikacji obiektów na obrazach. Sieci te uczą się rozpoznawać cechy hierarchiczne w obrazach, takie jak krawędzie, tekstury, a następnie wykorzystują je do identyfikacji obiektów.

Segmentacja obrazu: Sieci neuronowe mogą być stosowane do identyfikacji i wyodrębnienia konkretnych części obrazu, np. segmentacji obiektów na obrazie medycznym czy samochodzie.

2. Przetwarzanie języka naturalnego (NLP):

Tłumaczenie maszynowe: sieci neuronowe, zwłaszcza rekurencyjne lub uwagowe, są stosowane do tłumaczenia jednego języka na inny. Przetwarzają one duże ilości danych tekstowych, uczą się zależności między słowami i tłumaczą zdania z jednego języka na drugi.

Generowanie tekstu: modele generatywne, takie jak GPT (Generative Pre-trained Transformer), są zdolne do generowania płynnych, sensownych zdań na podstawie podanego kontekstu.

3. Systemy rekomendacyjne:

Rekomendacje produktów: wykorzystują informacje o zachowaniach użytkowników, aby przewidzieć preferencje i proponować użytkownikom produkty lub treści, które mogą ich zainteresować.

4. Diagnostyka medyczna:

Analiza obrazów medycznych: sieci neuronowe pomagają w diagnozowaniu chorób na podstawie obrazów medycznych, np. wykrywanie raka na zdjęciach mammograficznych czy analiza tomografii komputerowej (CT) w celu identyfikacji zmian patologicznych.

5. Przetwarzanie dźwięku:

Rozpoznawanie mowy: sieci neuronowe wykorzystywane są do konwersji mowy na tekst, rozpoznawania komend głosowych, a także do generowania syntetycznego dźwięku lub mowy.

6. Predykcja i klasyfikacja:

Predykcja cen akcji: sieci neuronowe mogą być wykorzystane do prognozowania cen akcji na podstawie analizy historycznych danych rynkowych.

Klasyfikacja tekstu: Klasyfikują teksty na różne kategorie, takie jak sentyment analizy opinii, identyfikacja kategorii tekstu itp.

Warto zauważyć, że sieci neuronowe są wszechstronne i mogą być dostosowywane do różnych zastosowań. Oprócz wyżej wymienionych przykładów, ich elastyczność pozwala na rozwiązywanie szeregu innych problemów w różnych dziedzinach.

6

Jak sieci neuronowe rozwiązują problemy

6.2. Jak działają w praktyce



Sieci neuronowe działają w praktyce poprzez procesy uczenia maszynowego, w których są trenowane na zbiorze danych w celu rozpoznawania wzorców, podejmowania decyzji lub wykonywania predykcji na podstawie nowych danych.

Pierwszym krokiem jest przygotowanie odpowiednich danych treningowych. W przypadku sieci neuronowych dane muszą być odpowiednio przetworzone, skalowane i podzielone na zestawy treningowe, walidacyjne i testowe.

Następnie należy zdecydować o architekturze sieci, tj. rodzaju sieci, liczbie warstw, rodzajach warstw i ich rozmiarze. W zależności od problemu mogą to być sieci konwolucyjne, rekurencyjne, sieci w pełni połączone itp.

Wagi sieci są inicjalizowane, a następnie sieć jest trenowana na danych treningowych. Proces ten polega na propagowaniu danych przez sieć (przez wiele iteracji zwanych epokami), obliczaniu błędów i aktualizacji wag w celu minimalizacji tego błędów.

Po zakończeniu trenowania sieć jest oceniana na zestawie walidacyjnym lub testowym w celu

określenia jej wydajności. Ocena modelu obejmuje analizę różnych metryk zgodnie z charakterystyką problemu, takich jak dokładność, precyzja, odsetek błędów itp.

Na podstawie wyników oceny modelu, można dostosować parametry sieci (takie jak architektura, funkcje aktywacji, liczba warstw) w celu poprawy jej wydajności.

Po zakończeniu procesu trenowania i optymalizacji, gotowy model może być wykorzystywany do prognozowania na nowych danych. Na przykład, w przypadku klasyfikacji obrazów, model może rozpoznawać obiekty na nowych zdjęciach.

W praktyce, sieci neuronowe mogą być stosowane do różnorodnych zastosowań, od analizy danych biznesowych po rozpoznawanie obrazów czy generowanie tekstu. W miarę rozwijania się technologii, pojawiają się nowe architektury sieci, algorytmy trenowania i techniki optymalizacyjne, co prowadzi do coraz bardziej skutecznych i precyzyjnych modeli sieci neuronowych.

7 Etyka i bezpieczeństwo w świecie komputerów neuronowych

Dlaczego ważne jest dbanie o etykę w technologii



Dbanie o etykę w technologii jest kluczowe z kilku powodów:

Wpływ społeczny: technologia ma ogromny wpływ na społeczeństwo. Etyczne decyzje podejmowane w obszarze technologii mogą bezpośrednio wpłynąć na ludzi, ich życie prywatne, wolność, bezpieczeństwo i prawa.

Bezpieczeństwo i prywatność: ochrona danych osobowych, prywatności użytkowników oraz zapewnienie bezpieczeństwa w technologii są kluczowe. Nieetyczne praktyki, takie jak nadużywanie danych czy niewłaściwe wykorzystanie informacji, mogą prowadzić do poważnych naruszeń prywatności i bezpieczeństwa.

Sprawiedliwość i równość: technologia powinna być tworzona w sposób sprawiedliwy i równościowy, zapewniając równy dostęp do zasobów i możliwości dla wszystkich, bez dyskryminacji ze względu na rasę, płeć, orientację seksualną, pochodzenie czy inne cechy.

Zaufanie i zaangażowanie użytkowników: etyczne podejście do technologii buduje zaufanie użytkowników. Kiedy technologie są rozwijane i stosowane w sposób etyczny, użytkownicy są bardziej skłonni do zaufania i akceptacji tych rozwiązań.

Unikanie negatywnych skutków ubocznych: nieprzewidziane negatywne skutki uboczne technologii mogą mieć poważne konsekwencje. Etyczne podejście może pomóc w identyfikacji, zapobieganiu i minimalizacji tych skutków.

Przestrzeganie zasad moralnych: dbanie o etykę w technologii to również przestrzeganie zasad moralnych i społecznych, takich jak uczciwość, uczciwość intelektualna i poszanowanie wartości ludzkich.

Regulacje i standardy: w wielu krajach powstają regulacje i standardy dotyczące etyki w technologii. Przestrzeganie tych przepisów jest istotne dla legalności i zgodności działań z prawem.

Podsumowując, dbanie o etykę w technologii jest nie tylko kwestią moralną, ale także ma ogromne znaczenie dla społeczeństwa, bezpieczeństwa danych, równości, zaufania użytkowników oraz zgodności z przepisami i normami społecznymi. Etyczne podejście jest fundamentalne dla odpowiedzialnego rozwoju technologicznego.





7 Etyka i bezpieczeństwo w świecie komputerów neuronowych

7.2. Bezpieczeństwo w sieciach neuronowych

Bezpieczeństwo w sieciach neuronowych obejmuje szereg zagadnień związanych z ochroną modeli przed różnymi rodzajami ataków, zapewnienie prywatności danych, zapobieganie nadużyciom oraz utrzymanie integralności modeli.

Ataki wstrzykiwania szumu (Adversarial Attacks): sieci neuronowe są podatne na ataki wstrzykiwania szumu, co prowadzi do zmiany wyników klasyfikacji lub predykcji, nawet w przypadku niewielkich modyfikacji wejściowych danych.

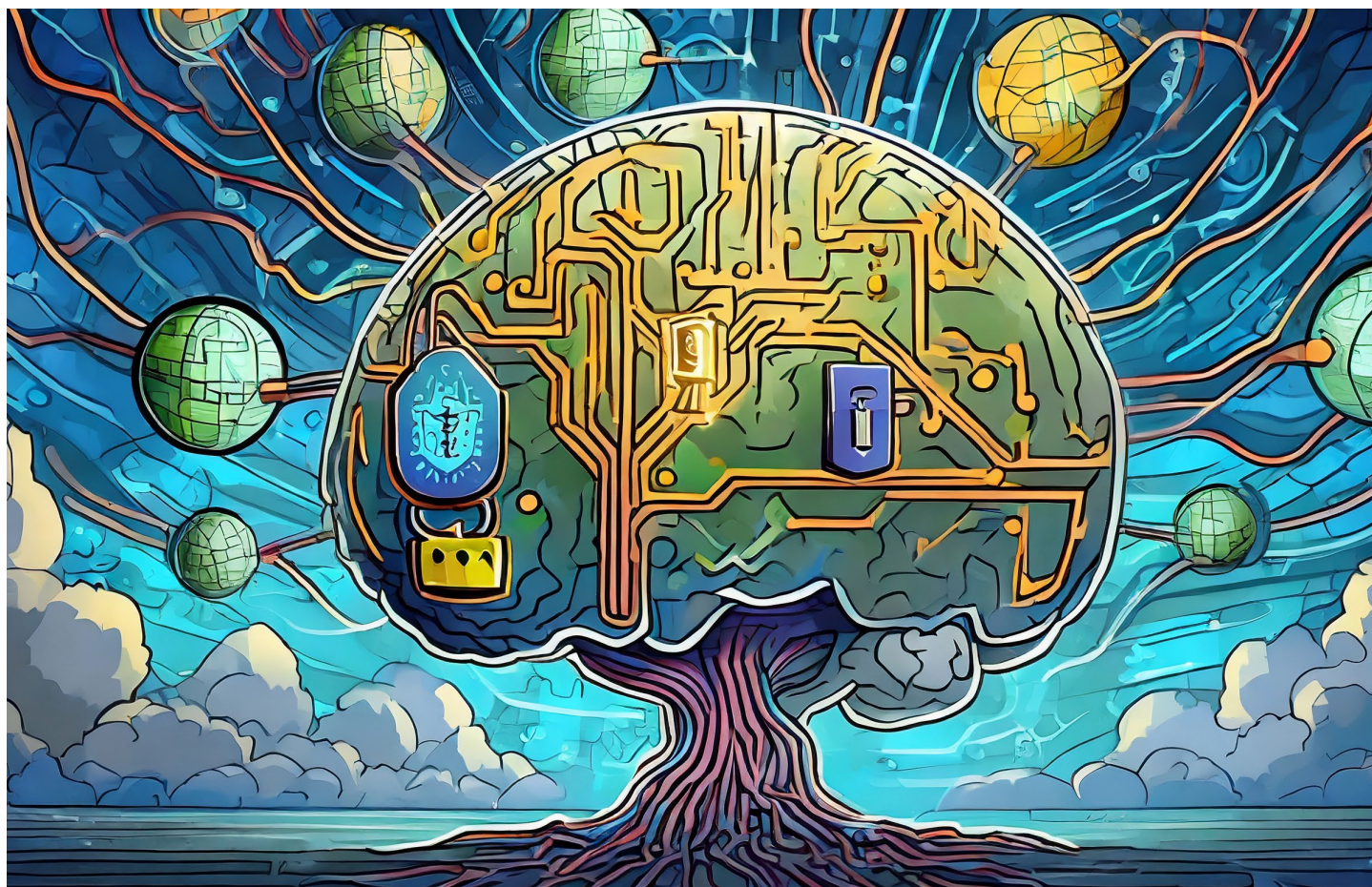
Zabezpieczenia przed atakami wstrzykiwania danych: konieczne jest rozwijanie metod odporności na ataki, które zapobiegają lub minimalizują skutki ataków wstrzykiwania danych, takich jak wstrzykiwanie szumu, zmiany w danych wejściowych itp.

Ochrona prywatności: w kontekście sieci neuronowych, szczególnie w dziedzinach medycznych czy analizie danych osobowych, konieczne jest zapewnienie ochrony danych, aby uniknąć wycieku informacji osobistych lub wrażliwych.

Ryzyko nadużyć i nieuczciwości: sieci neuronowe mogą być wykorzystywane do manipulacji informacją, tworzenia deepfake'ów, rozprzestrzeniania dezinformacji czy innego rodzaju nadużyć, dlatego istnieje potrzeba rozwijania metod wykrywania i przeciwdziałania takim praktykom.

Zabezpieczenie przed atakami zewnętrznymi: modele sieci neuronowych powinny być zabezpieczone przed atakami z zewnątrz, takimi jak próby przejęcia kontroli nad modelem, ataki typu Denial of Service (DoS) czy próby włamania.





7 Etyka i bezpieczeństwo w świecie komputerów neuronowych

7.2. Bezpieczeństwo w sieciach neuronowych

Odporność na ataki adversarialne: istotne jest rozwijanie modeli odpornych na ataki adversarialne poprzez metody takie jak wzmocnienie danych treningowych, stosowanie technik normalizacji czy architektur odpornych na ataki.

Zachowanie integralności modeli: konieczne jest monitorowanie i zapewnienie integralności modeli, aby uniknąć ich manipulacji lub zmian, które mogą prowadzić do fałszywych wyników lub błędnych predykcji.

Bezpieczeństwo w sieciach neuronowych to ważny obszar badawczy, którego celem jest zwiększenie odporności modeli na różnego rodzaju zagrożenia oraz zachowanie ich bezpieczeństwa, integralności i niezawodności w różnych środowiskach i zastosowaniach.





8

Zabawki i gry związane z komputerami neuronowymi

8.1. Gry i zabawki, które wykorzystują komputery neuronowe

Komputery neuronowe znalazły zastosowanie w różnych grach i zabawkach, zarówno w celach edukacyjnych, jak i rozrywkowych

Wiele gier wideo wykorzystuje technologie oparte na komputerach neuronowych do udoskonalenia sztucznej inteligencji przeciwników w grze. Te modele mogą uczyć się z interakcji z graczem, dostosowując swoje zachowanie, takie jak strategię, umiejętności czy reakcje na działania gracza.

Komputery neuronowe mogą być wykorzystane do analizy ruchów w grach planszowych takich jak szachy czy Go, aby stworzyć boty lub algorytmy, które uczą się strategii i stają się coraz lepsze w grze, nawet konkurując z najlepszymi ludzkimi graczami.

Istnieją zabawki, które wykorzystują technologie oparte na komputerach neuronowych do interakcji z dziećmi, ucząc się preferencji i zachowań, aby dostosowywać się do potrzeb użytkownika. Mogą to być interaktywne lalki, zwierzątka czy roboty, które reagują na bodźce otoczenia.

Niektóre gry edukacyjne lub platformy do nauki korzystają z technologii komputerów neuronowych, aby dostosowywać treści do indywidualnych zdolności i potrzeb uczniów, zapewniając spersonalizowane i efektywne metody nauczania.

Istnieją zestawy zabawek, które uczą dzieci podstaw programowania i algorytmów przy użyciu prostych interfejsów, często wykorzystując komputery neuronowe do uproszczenia procesu nauki i interakcji.

Zabawki robotyczne, takie jak roboty interaktywne, wykorzystujące technologie komputerów neuronowych, aby nauczyć się rozpoznawania mowy, reagowania na polecenia, czy nawet tworzenia emocjonalnej więzi z użytkownikiem.



8.2. Tworzenie własnych prostych projektów

```
import tensorflow as tf
from tensorflow.keras.datasets import cifar10
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense, Dropout

# Wczytanie danych CIFAR-10
(x_train, y_train), (x_test, y_test) = cifar10.load_data()

# Normalizacja danych
x_train = x_train.astype('float32') / 255.0
x_test = x_test.astype('float32') / 255.0

# Definicja modelu
model = Sequential([
    Conv2D(32, (3, 3), activation='relu', input_shape=(32, 32, 3)),
    MaxPooling2D((2, 2)),
    Conv2D(64, (3, 3), activation='relu'),
    MaxPooling2D((2, 2)),
    Conv2D(64, (3, 3), activation='relu'),
    Flatten(),
    Dense(64, activation='relu'),
    Dropout(0.5),
    Dense(10, activation='softmax')
])

# Kompilacja modelu
model.compile(optimizer='adam', loss='sparse_categorical_crossentropy', metrics=['accuracy'])

# Trening modelu
model.fit(x_train, y_train, epochs=10, batch_size=64, validation_data=(x_test, y_test))

# Ocena dokładności modelu
test_loss, test_accuracy = model.evaluate(x_test, y_test)
print(f"Dokładność modelu na danych testowych: {test_accuracy}")
```

Prosty przykład wykorzystania biblioteki Keras z backendem TensorFlow do stworzenia prostego modelu do rozpoznawania obiektów na zdjęciach z wykorzystaniem zestawu danych CIFAR-10. Ten zbiór danych zawiera 60 000 kolorowych obrazów o wymiarach 32x32 podzielonych na 10 klas.

Tworzenie prostych projektów z wykorzystaniem komputerów neuronowych może być fascynującym i edukacyjnym doświadczeniem. Oto kilka prostych projektów, które można zrealizować:

Rozpoznawanie obrazów:

Stwórz prosty model do rozpoznawania obiektów na zdjęciach. Wykorzystaj gotowe biblioteki, takie jak TensorFlow lub Keras, aby nauczyć model rozpoznawać różne obiekty.

Klasyfikacja tekstu:

Zbuduj model klasyfikujący teksty. Może to być projekt, który rozpoznaje emocje w tekście (np. czy dany tekst jest pozytywny czy negatywny), czy też kategorie wiadomości.

Gry wideo z elementami AI:

Stwórz prostą grę wideo, w której przeciwnik sterowany jest przez sztuczną inteligencję. Zaimplementuj algorytm decyzyjny, który pozwoli komputerowemu przeciwnikowi reagować na działania gracza.

Generowanie tekstu:

Wykorzystaj modele językowe do stworzenia prostego generatora tekstu. Możesz zbudować model, który nauczy się na podstawie przykładowych tekstów i będzie potrafił generować nowe zdania.

Sieci neuronowe w robotyce:

Zbuduj prostego robota lub wykorzystaj zestaw do robotyki, który jest w stanie reagować na otoczenie. Wykorzystaj sieci neuronowe do nauczenia go rozpoznawania prostych wzorców lub wykonywania prostych zadań.

Klasyfikator rysunków:

Stwórz aplikację, która pozwoli użytkownikowi rysować obiekty, a następnie wykorzystaj sieć neuronową do rozpoznawania i klasyfikowania tych rysunków.

Prosty chatbot:

Zbuduj prostego chatbota, który będzie reagować na pytania użytkownika lub pomoże w podstawowych zadaniach.

Predykcja danych:

Użyj danych historycznych, aby stworzyć prosty model przewidujący np. cenę akcji, temperaturę lub inne wartości na podstawie wcześniejszych danych.

Projekty te można realizować, korzystając z dostępnych bibliotek i frameworków do tworzenia modeli sieci neuronowych, takich jak TensorFlow, Keras, PyTorch czy Scikit-learn. Istnieje wiele darmowych zasobów, kursów online i przykładowych projektów, które mogą pomóc w rozpoczęciu pracy nad własnymi projektami z wykorzystaniem komputerów neuronowych.

```
*C:\Program Files\Notepad++\change.log - Notepad++
Plik Edycja Szukaj Widok Format Składnia Ustawienia Narzędzia Makra Uruchom Wtyczki Karty ?
change.log x
1 import tensorflow as tf
2 from tensorflow.keras.datasets import imdb
3 from tensorflow.keras.preprocessing.sequence import pad_sequences
4 from tensorflow.keras.models import Sequential
5 from tensorflow.keras.layers import Embedding, LSTM, Dense, Dropout
6
7 # Ustawienia parametrów
8 max_features = 10000 # ograniczenie liczby słów (top 10 000 słów)
9 maxlen = 200 # ograniczenie długości recenzji do 200 słów
10
11 # Załaduj dane z bazy IMDB
12 (x_train, y_train), (x_test, y_test) = imdb.load_data(num_words=max_features)
13
14 # Ustawienie długości sekwencji (pad sequences)
15 x_train = pad_sequences(x_train, maxlen=maxlen)
16 x_test = pad_sequences(x_test, maxlen=maxlen)
17
18 # Zbudowanie modelu
19 model = Sequential([
20     Embedding(max_features, 128, input_length=maxlen),
21     LSTM(64),
22     Dense(1, activation='sigmoid')
23 ])
24
25 # Kompilacja modelu
26 model.compile(optimizer='adam',
27               loss='binary_crossentropy',
28               metrics=['accuracy'])
29
30 # Trenowanie modelu
31 model.fit(x_train, y_train, epochs=3, batch_size=64, validation_data=(x_test, y_test))
32
33
Normal text length: 1 081 lines: 33 Ln: 32 Col: 1 Pos: 1 080 Windows (CR LF) UTF-8 INS
```

Przykład zbudowania modelu klasyfikującego teksty za pomocą biblioteki TensorFlow/Keras.

Prosty przykład zbudowania modelu klasyfikującego teksty za pomocą biblioteki TensorFlow/Keras. W tym przypadku, wykorzystamy zbiór danych o recenzjach filmowych zawierający zdania, które są etykietowane jako pozytywne lub negatywne. Będziemy budować model, który będzie klasyfikować recenzje filmowe jako pozytywne lub negatywne na podstawie tekstu.

Opis kroków w kodzie:

1. Dane z bazy IMDB (recenzje filmowe) są wczytywane i ograniczane do maksymalnej liczby słów i długości sekwencji (recenzji).
2. Model sieci neuronowej składa się z warstw Embedding, LSTM i gęstej (Dense) z funkcją aktywacji sigmoidalnej na końcu.
3. Model jest kompilowany z określonym optymalizatorem, funkcją straty oraz metryką (dokładnością).

4. Trenowanie modelu odbywa się na danych treningowych, ocenianie jest wykonywane na danych testowych przez 3 epoki.

Ten prosty model klasyfikuje recenzje filmowe jako pozytywne lub negatywne na podstawie tekstu, ale może być rozbudowany lub zoptymalizowany w zależności od konkretnych potrzeb projektowych. Dodatkowo, istnieje wiele innych podejść do klasyfikacji tekstu, w tym wykorzystanie różnych architektur sieci neuronowych czy technik przetwarzania języka naturalnego.

Prosty przykład gry wideo w Pythonie, w którym komputerowy przeciwnik reaguje na działania gracza. W tej grze, komputerowy przeciwnik będzie próbował przewidzieć ruch gracza i podejmować odpowiednie decyzje w zależności od tego, gdzie znajduje się piłka. Do implementacji tej gry wykorzystamy bibliotekę **pygame**, która ułatwia tworzenie prostych gier wideo w Pythonie.

Ten kod stworzy prostą grę z dwoma platformami (gracz i przeciwnik), które poruszają się w poziomie w celu odbijania piłki. Przeciwnik będzie próbował przewidzieć ruch piłki i reagować na nią. To bardzo podstawowy przykład, który może być rozwinięty i ulepszony, na przykład poprzez bardziej zaawansowany algorytm decyzyjny dla przeciwnika czy dodanie elementów takich jak punkty, poziomy, czy efekty dźwiękowe.

```
*C:\Program Files\Notepad++\change.log - Notepad++
Plik Edycja Szukaj Widok Format Składnia Ustawienia Narzędzia Makra Uruchom Wtyczki Karty ?
change.log x
1 import pygame
2 import random
3
4 # Inicjalizacja modułu pygame
5 pygame.init()
6
7 # Ustawienia ekranu
8 screen_width, screen_height = 800, 600
9 screen = pygame.display.set_mode((screen_width, screen_height))
10 pygame.display.set_caption("Prosta gra z SI")
11
12 # Kolory
13 white = (255, 255, 255)
14 black = (0, 0, 0)
15 red = (255, 0, 0)
16
17 # Parametry gracza i przeciwnika
18 player_width, player_height = 100, 20
19 player_x, player_y = (screen_width - player_width) // 2, screen_height - 2 * player_height
20 enemy_width, enemy_height = 100, 20
21 enemy_x, enemy_y = (screen_width - enemy_width) // 2, player_height
22
23 # Parametry piłki
24 ball_radius = 10
25 ball_x, ball_y = screen_width // 2, screen_height // 2
26 ball_dx, ball_dy = 5, 5
27
28 # Szybkość przeciwnika
29 enemy_speed = 7
30
31 # Funkcja rysująca gracza i przeciwnika
32 def draw_objects():
33     screen.fill(black)
34     pygame.draw.rect(screen, white, (player_x, player_y, player_width, player_height))
35     pygame.draw.rect(screen, red, (enemy_x, enemy_y, enemy_width, enemy_height))
36     pygame.draw.circle(screen, white, (ball_x, ball_y), ball_radius)
37
38 # Pętla główna gry
39 running = True
40 while running:
41     for event in pygame.event.get():
42         if event.type == pygame.QUIT:
43             running = False
44
45     keys = pygame.key.get_pressed()
46     if keys[pygame.K_LEFT] and player_x > 0:
47         player_x -= 5
48     if keys[pygame.K_RIGHT] and player_x < screen_width - player_width:
49         player_x += 5
50
```

Prosty przykład gry wideo w Pythonie

```
*C:\Program Files\Notepad++\change.log - Notepad++
Plik Edycja Szukaj Widok Format Składnia Ustawienia Narzędzia Makra Uruchom Wtyczki Karty ?
change.log x
50
51 # Ruch przeciwnika (algorytm decyzyjny)
52 if ball_y < screen_height // 2:
53     if enemy_x + enemy_width // 2 < ball_x:
54         enemy_x += enemy_speed
55     elif enemy_x + enemy_width // 2 > ball_x:
56         enemy_x -= enemy_speed
57
58 # Ruch piłki
59 ball_x += ball_dx
60 ball_y += ball_dy
61
62 # Kolidacja piłki z granicami ekranu
63 if ball_x <= 0 or ball_x >= screen_width:
64     ball_dx = -ball_dx
65 if ball_y <= 0 or ball_y >= screen_height:
66     ball_dy = -ball_dy
67
68 # Kolidacja piłki z graczem i przeciwnikiem
69 if ball_y >= player_y - ball_radius and player_x <= ball_x <= player_x + player_width:
70     ball_dy = -ball_dy
71 if ball_y <= enemy_y + enemy_height + ball_radius and enemy_x <= ball_x <= enemy_x + enem
72     ball_dy = -ball_dy
73
74 draw_objects()
75 pygame.display.update()
76
77 pygame.quit()
78
```

```
*C:\Program Files\Notepad++\change.log - Notepad++
Plik Edycja Szukaj Widok Format Składnia Ustawienia Narzędzia Makra Uruchom Wtyczki Karty ?
change.log x
1 import nltk
2 from nltk.chat.util import Chat, reflections
3
4 # Zdefiniowanie par wzorców i odpowiedzi
5 pairs = [
6     ['(czego|czemu) nie rozumiesz (.*)',
7      ["Przepraszam, nie jestem w stanie odpowiedzieć na to pytanie.",
8       "Nie mam odpowiedzi na to pytanie. Może spróbuj zapytać o coś innego?"]],
9
10    ['(jak|jaki|co to) (.*)',
11     ["Przepraszam, nie jestem pewien co to oznacza.",
12      "Niestety, nie mam informacji na ten temat."]],
13
14    ['(czy potrafisz|czy umiesz) (.*)',
15     ["Tak, potrafię rozmawiać i odpowiadać na pytania.",
16      "Jestem stworzony do pomocy w prostych zadaniach."]],
17
18    ['(.*)',
19     ["Rozumiem.",
20      "Czy możesz powiedzieć coś więcej?"]],
21 ]
22
23 # Stworzenie chatbota na bazie zdefiniowanych par wzorców i odpowiedzi
24 chatbot = Chat(pairs, reflections)
25
26 # Uruchomienie interaktywnej sesji z chatbotem
27 print("Cześć, w czym mogę Ci pomóc? (Aby zakończyć rozmowę, napisz 'quit')")
28 while True:
29     user_input = input("Ty: ")
30     if user_input.lower() == 'quit':
31         break
32     response = chatbot.respond(user_input)
33     print("Bot:", response)
34
Normal text length: 1 148 lines: 35 Ln: 34 Col: 1 Pos: 1 147 Windows (CR LF) UTF-8 INS
```

**Prosty chatbot, który będzie reagować na pytania użytkownika
lub pomoże w podstawowych zadaniach.**

Zbudowanie prostego chatbota można osiągnąć przy użyciu narzędzi do przetwarzania języka naturalnego i prostych reguł dialogowych. Poniżej znajdziesz przykładowy kod wykorzystujący bibliotekę Python **nltk** do stworzenia prostego chatbota, który odpowie na proste pytania. Jeśli nie masz biblioteki **nltk** zainstalowanej, możesz ją zainstalować używając **pip install nltk**.

Ten prosty chatbot wykorzystuje zdefiniowane pary wzorców i odpowiedzi do reagowania na pytania użytkownika. Możesz dostosować listę par wzorców i odpowiedzi, aby lepiej odpowiadać na konkretne pytania lub potrzeby użytkownika. Ta implementacja opiera się na prostych regułach i może być rozwinięta z większą ilością reguł, aby obsługiwać szerszy zakres pytań. Jednakże, jeśli chcesz zbudować bardziej zaawansowanego chatbota zdolnego do zrozumienia kontekstu i bardziej skomplikowanych zadań, konieczne będzie wykorzystanie bardziej zaawansowanych metod przetwarzania języka naturalnego oraz modeli uczenia maszynowego.


```

1 import numpy as np
2 from sklearn.linear_model import LinearRegression
3 import matplotlib.pyplot as plt
4
5 # Przykładowe dane historyczne - ceny akcji (symulowane)
6 # Tutaj należy podać rzeczywiste dane historyczne
7 data_points = 100
8 X = np.array(range(data_points)).reshape(-1, 1) # Dni (przykładowo)
9 y = np.random.randint(50, 150, size=data_points) # Ceny akcji (symulowane)
10
11 # Utworzenie modelu regresji liniowej i dopasowanie go do danych
12 model = LinearRegression()
13 model.fit(X, y)
14
15 # Predykcja kolejnych cen akcji
16 future_days = 10 # Liczba dni do przewidzenia
17 future_X = np.array(range(data_points, data_points + future_days)).reshape(-1, 1)
18 future_predictions = model.predict(future_X)
19
20 # Wykres danych historycznych i przewidywanych wartości
21 plt.figure(figsize=(10, 6))
22 plt.scatter(X, y, color='blue', label='Historical Data')
23 plt.plot(np.concatenate([X, future_X]), np.concatenate([model.predict(X), future_predictions]))
24 plt.xlabel('Days')
25 plt.ylabel('Stock Price')
26 plt.title('Stock Price Prediction')
27 plt.legend()
28 plt.show()
29

```

Prosty model przewidujący np. cenę akcji, temperaturę lub inne wartości na podstawie wcześniejszych danych.

Rozwiązanie tego rodzaju problemu można podjąć wykorzystując uczenie maszynowe, a konkretnie regresję liniową lub inne bardziej zaawansowane modele, takie jak sieci neuronowe rekurencyjne (RNN) lub sieci neuronowe rekurencyjne długoterminowe (LSTM). Poniżej znajdziesz prosty przykład wykorzystania regresji liniowej do przewidywania wartości na podstawie danych historycznych.

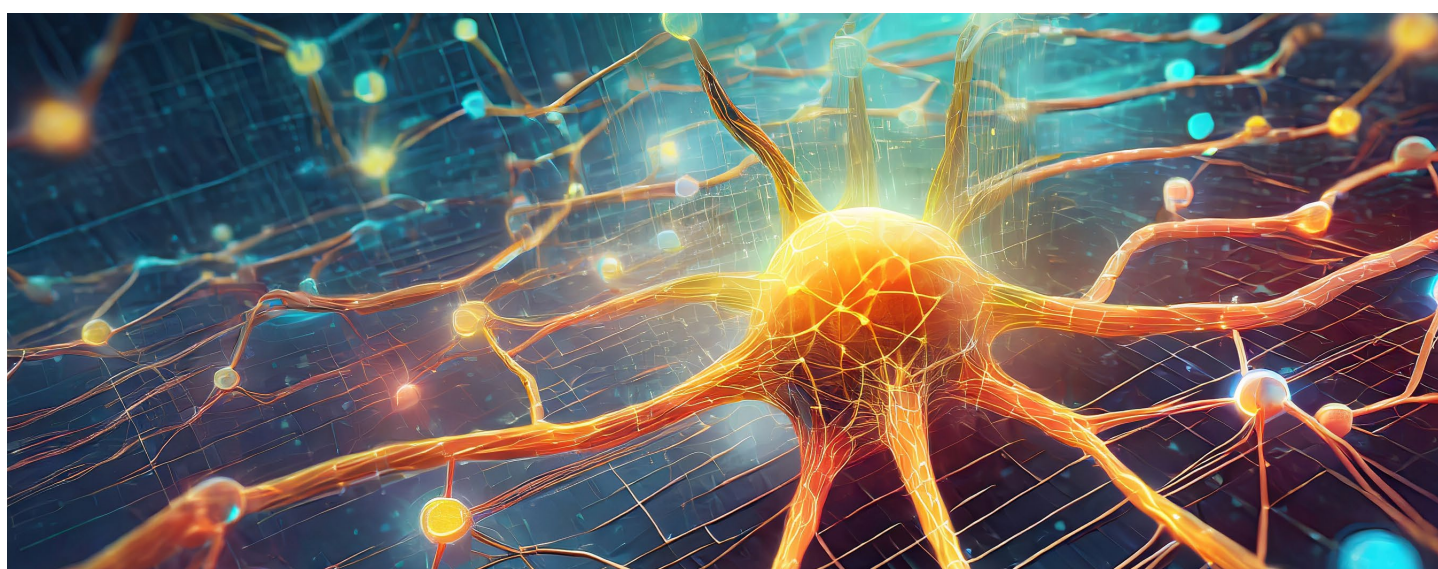
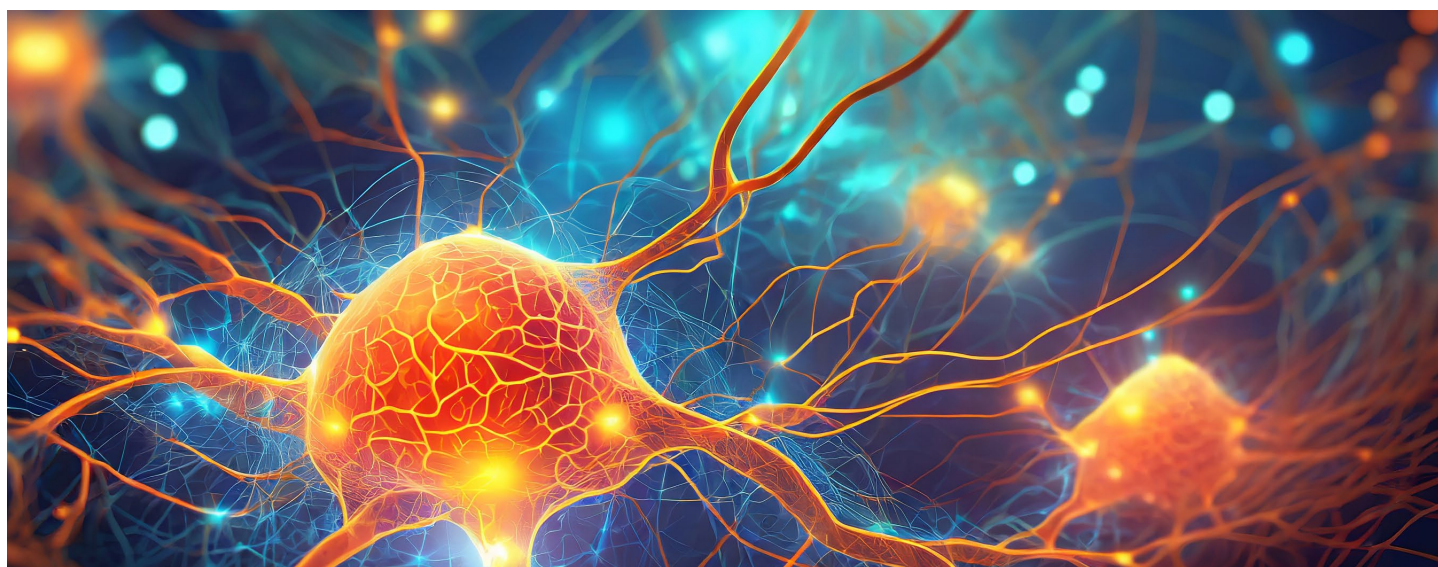
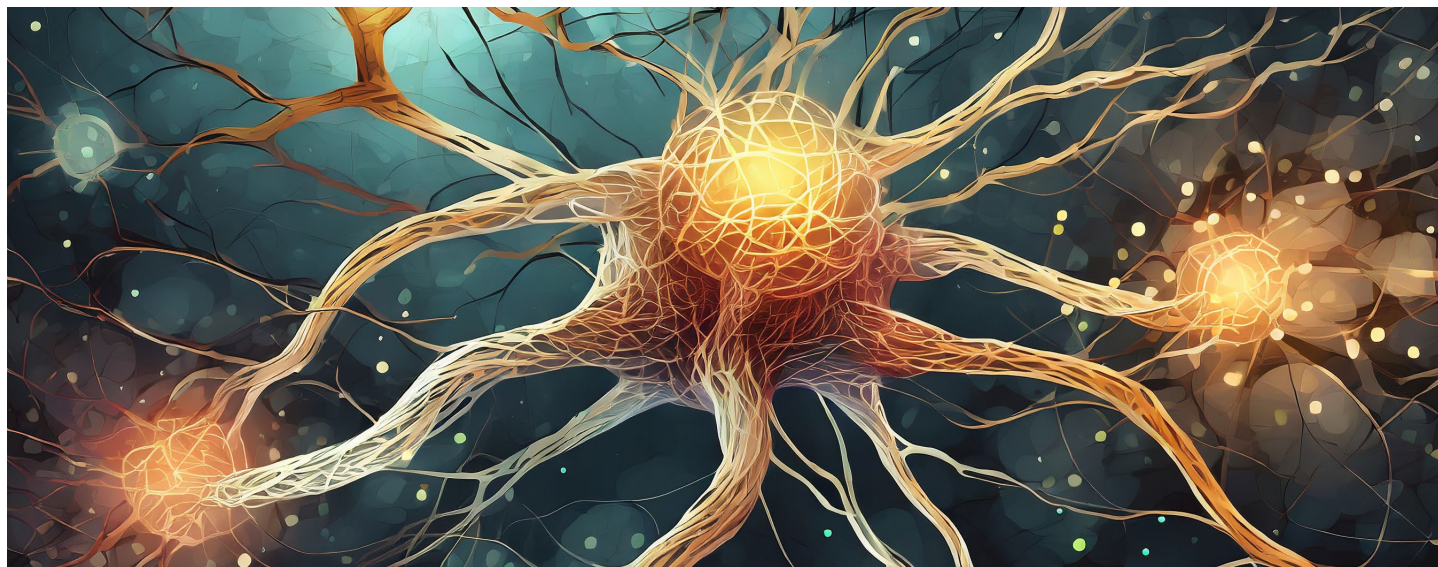
Załóżmy, że mamy zbiór danych zawierający historyczne ceny akcji. Wykorzystamy pakiet **scikit-learn** do zbudowania modelu regresji liniowej i przewidzenia kolejnych cen na podstawie wcześniejszych danych.

Ten kod tworzy model regresji liniowej na podstawie historycznych danych cen akcji i używa go do przewidywania cen na kolejne dni (określona liczbę dni w przyszłość).

W rzeczywistości dane historyczne mogą być bardziej złożone i składać się z wielu cech, a model regresji liniowej może nie być wystarczająco skuteczny. W takim przypadku, bardziej zaawansowane modele, takie jak sieci neuronowe, mogą zapewnić lepsze wyniki przewidywań.

Należy pamiętać, że wykorzystując rzeczywiste dane, należy odpowiednio je przetworzyć, oczyścić i dostosować do potrzeb modelu. Ponadto, ocena wydajności

modelu wymaga stosowania odpowiednich metryk i walidacji.



9 Przyszłość komputerów neuronowych

9.1. Jak technologia się rozwija

Rzeczywistość to dynamiczny proces, który stale ewoluuje i przyczynia się do zmian w różnych dziedzinach życia. Oto kilka kluczowych obszarów, w których technologia stale się rozwija:

Sztuczna inteligencja i uczenie maszynowe: stały postęp w dziedzinie sztucznej inteligencji (SI) i uczenia maszynowego przynosi nowe techniki, modele i algorytmy, które mają zastosowanie w szerokim zakresie, od analizy danych po diagnostykę medyczną i automatykę.

Internet Rzeczy (IoT): rozwój technologii związanych z Internetem Rzeczy umożliwia połączenie urządzeń codziennego użytku z internetem, co otwiera nowe możliwości w dziedzinie zdrowia, transportu, rolnictwa, przemysłu i innych sektorach.

5G i sieci komunikacyjne: wdrażanie technologii 5G zmienia sposób, w jaki ludzie łączą się ze sobą i z siecią. Prędkość, niskie opóźnienia i większa przepustowość mają wpływ na rozwój mobilności, inteligentnych miast, gier, zdrowia i wielu innych dziedzin.

Blockchain i kryptowaluty: Blockchain, będący podstawą kryptowalut takich jak Bitcoin i Ethereum, rozwija się i ma potencjał do rewolucjonizacji sektorów finansowych, handlu, zdrowia, zarządzania łańcuchem dostaw i wielu innych.

Biotechnologia i genomika: postęp w dziedzinie biotechnologii, w tym w sekwencjonowaniu genomu, terapiach genowych czy nowoczesnych technologiach medycznych, otwiera drzwi do nowych sposobów leczenia chorób oraz diagnozowania.

Energetyka odnawialna i technologie ekologiczne: Coraz większe zaangażowanie

w energię odnawialną, rozwój technologii zielonych, baterii, paneli słonecznych, czy efektywności energetycznej, mają na celu zmniejszenie wpływu człowieka na środowisko.

Robotyka i automatyzacja: rozwój robotyki i automatyzacji zmienia sposób, w jaki wykonywane są różne prace, od linii produkcyjnych w fabrykach po dostawę, opiekę zdrowotną czy rolnictwo.

Cyberbezpieczeństwo: w miarę jak technologia się rozwija, wzrasta również zapotrzebowanie na lepsze strategie cyberbezpieczeństwa, aby chronić dane, systemy i infrastrukturę przed atakami cybernetycznymi.

Te obszary stanowią jedynie część dynamicznego spektrum zmian, jakie zachodzą w świecie technologii. Ciągły rozwój w tych dziedzinach przynosi ze sobą zarówno nowe wyzwania, jak i niezliczone możliwości.

“

*Już jesteśmy cyborgami.
Twój telefon i komputer są
twoim przedłużeniem,
choć interfejs wymaga
użycia ruchów palców lub
mowy, które są bardzo
wolne ...*

Elon Musk

”

9

Przyszłość komputerów neuronowych

9.2. Co przyniesie przyszłość?

Przyszłość niesie ze sobą wiele niezwykłych możliwości i zmian, które mogą znacząco wpłynąć na nasze życie. Oto kilka obszarów, które mogą mieć istotny wpływ na przyszłość:

1. Sztuczna inteligencja i uczenie maszynowe: kontynuacja postępu w dziedzinie sztucznej inteligencji może prowadzić do coraz bardziej zaawansowanych systemów, które będą miały zastosowanie w wielu obszarach życia, od medycyny po mobilność i handel.

2. Internet Rzeczy (IoT) i inteligentne miasta: rozwój Internetu Rzeczy sprawi, że coraz więcej urządzeń będzie łączyło się ze sobą, co stworzy podstawę dla inteligentnych miast, poprawiających efektywność, bezpieczeństwo i jakość życia mieszkańców.

3. Mobilność: rozwój technologii, w tym autonomicznych pojazdów i technologii 5G, może zmienić sposób, w jaki podróżujemy, transportujemy towary i organizujemy przestrzeń miejską.

4. Zdrowie i biotechnologia: postęp w biotechnologii, terapiach genowych i diagnostyce medycznej może umożliwić bardziej spersonalizowane podejście do leczenia chorób oraz poprawę jakości opieki zdrowotnej.

5. Energetyka odnawialna: dalszy rozwój energii odnawialnej, nowych źródeł energii i technologii efektywności energetycznej może odegrać kluczową rolę w zmniejszeniu emisji gazów cieplarnianych i przystosowaniu się do zmian klimatu.

6. Rozwój technologii kwantowych: postęp w dziedzinie technologii kwantowych może umożliwić znaczący skok w dziedzinie komputacji, kryptografii, materiałów i nauk przyrodniczych.

7. Edukacja i nauczanie zdalne: zwiększone wykorzystanie technologii w edukacji może prowadzić do bardziej dostępnej, elastycznej i spersonalizowanej edukacji dla osób na całym świecie.

8. Cyberbezpieczeństwo: w miarę jak technologia się rozwija, istnieje coraz większe zapotrzebowanie na bardziej zaawansowane rozwiązania z zakresu cyberbezpieczeństwa, aby chronić dane i systemy przed atakami.

To tylko kilka z potencjalnych kierunków rozwoju, jakie przyniesie przyszłość. Obserwowanie tych obszarów i ich wpływ na nasze życie będzie stanowiło fascynującą podróż wraz z postępowaniem technologicznym. Kluczem jest zachowanie elastyczności, gotowości do adaptacji oraz etycznego i odpowiedzialnego wykorzystywania nowych technologii dla dobra społeczeństwa.

Jak samodzielnie stworzyć prosty projekt z wykorzystaniem komputera neuronowego

10.1. Instrukcje krok po kroku do stworzenia prostego projektu

Krok 1: Wybierz temat projektu

Wybierz obszar, który Cię interesuje (np. aplikacja internetowa, aplikacja mobilna, robotyka, analiza danych, sztuczna inteligencja itp.).

Krok 2: Określ cel projektu

Sprecyzuj, czego chcesz osiągnąć poprzez swój projekt. To może być stworzenie aplikacji do zarządzania zadaniami, stworzenie prostego robota, analiza danych dotyczących ruchu na stronie internetowej itp.

Krok 3: Zdobądź niezbędne narzędzia i zasoby

W zależności od tematu projektu, zidentyfikuj narzędzia, technologie i zasoby potrzebne do jego realizacji. Na przykład, jeśli planujesz tworzyć aplikacje internetowe, możesz potrzebować wiedzy na temat HTML, CSS, JavaScript, a także frameworka takiego jak React lub Angular.

Krok 4: Zaprojektuj projekt

Stwórz szkic lub plan projektu, zidentyfikuj podstawowe funkcje i etapy realizacji.

Krok 5: Rozpocznij implementację

Zacznij od podstawowego kodu, wykonaj pierwsze kroki w swoim projekcie. Jeśli to aplikacja, rozpocznij od stworzenia prostego interfejsu użytkownika lub bazowej funkcjonalności. Dla projektów związanymi z analizą danych, zbierz i przygotuj dane.

“

Postęp technologii polega na dostosowaniu jej tak, abyś nawet jej nie zauważał i tak by mogła stać się częścią codziennego życia ...

Bill Gates

”

Krok 6: Testuj i rozwijaj

Regularnie testuj projekt podczas jego tworzenia. Poprawiaj błędy i dodawaj nowe funkcje na bieżąco.

Krok 7: Dokumentuj projekt

Ważne jest, aby dokumentować postępy w projekcie. Zapisuj zmiany, twórz dokumentację kodu lub instrukcje, które mogą być pomocne dla Ciebie lub innych osób, które będą pracować nad projektem.

Krok 8: Finalizacja i publikacja (jeśli to aplikacja)

Gdy Twój projekt osiągnie oczekiwany stan lub funkcjonalność, wykonaj ostateczne testy. Jeśli to aplikacja, przygotuj ją do publikacji, np. w sklepie aplikacji.

Krok 9: Optymalizacja i doskonalenie

Po publikacji kontynuuj pracę nad projektem. Optymalizuj kod, reaguj na opinie użytkowników (jeśli to aplikacja), doskonal swoje umiejętności lub technologie wykorzystane w projekcie.

Krok 10: Dziel się swoim projektem

Udostępnij swój projekt innym, opublikuj go na platformach open source lub wśród społeczności, aby inni mogli z niego skorzystać lub się zainspirować.

Zachęcamy do elastycznego podejścia do realizacji projektu oraz nieustannego uczenia się i rozwijania swoich umiejętności. Każdy krok w tworzeniu projektu przynosi nowe doświadczenia i możliwości nauki!



Jesteś geniuszem! Spróbuj!

10 | Jak samodzielnie stworzyć prosty projekt z wykorzystaniem komputera neuronowego

10.2. Rozwinięcie własnych pomysłów

Rozwinięcie własnych pomysłów na projekty może być ekscytującym procesem. Oto kilka kreatywnych sposobów na rozwinięcie własnych pomysłów:

1. Badanie rynku i potrzeb: przeprowadź badanie rynku, aby zidentyfikować luki lub potrzeby, na które Twój projekt mógłby odpowiedzieć. Zrozumienie, czego ludzie potrzebują lub czego brakuje, może pomóc w stworzeniu projektu o większym potencjale sukcesu.

2. Połączenie różnych pomysłów: połącz kilka różnych pomysłów w jeden projekt. Czasem połączenie elementów z różnych dziedzin lub koncepcji może prowadzić do unikalnych i interesujących projektów.

3. Brainstorming:

Zbierz grupę osób i przeprowadź sesję brainstorimingu. Dyskutujcie nad pomysłami, podzielcie się swoimi pomysłami i połączcie je, aby stworzyć coś nowego.

4. Analiza konkurencji:

Zbadaj istniejące projekty lub rozwiązania związane z Twoim pomysłem. Zidentyfikuj ich mocne i słabe strony, a następnie pomyśl, jak można usprawnić lub wdrożyć nowe podejście w Twoim projekcie.

5. Inspiracja z życia codziennego:

Skup się na codziennych problemach, z którymi spotykasz się na co dzień. Czasami najlepsze pomysły na projekty rodzą się z prób znalezienia rozwiązania dla własnych codziennych wyzwań.

6. Patrzenie w przyszłość:

Zastanów się, jak będzie wyglądać świat za 5, 10 lub 20

lat. Próbuj przewidzieć, jakie będą potrzeby ludzi i jakie nowe technologie lub podejścia mogą być potrzebne w przyszłości.

7. Połączenie pasji z projektem:

Znajdź obszar, który Cię pasjonuje, i spróbuj stworzyć projekt związany z tą pasją. Tworzenie projektów, które są zbieżne z Twoimi zainteresowaniami, może być inspirujące i satysfakcjonujące.

8. Testowanie i iterowanie:

Nie bój się eksperymentować i testować różne pomysły. Czasami pierwszy pomysł nie jest najlepszy, ale poprzez iterowanie i testowanie można go ulepszyć.

Znalezienie własnego pomysłu może być ekscytującym wyzwaniem. Ważne jest, aby być elastycznym i otwartym na eksperymentowanie, uczenie się i ewoluowanie pomysłu w trakcie jego realizacji. Często najlepsze pomysły rodzą się poprzez aktywne działanie i próbowanie różnych podejść.

11 Podsumowanie

11.1. Ważne punkty, które warto pamiętać



Oto kilka istotnych punktów, które warto pamiętać podczas tworzenia projektów lub podczas pracy nad nimi:

1. Określenie celu projektu:

Zdefiniuj cel: sprecyzuj, czego chcesz osiągnąć poprzez swój projekt. Wartość klarownego celu jest kluczowa dla skutecznej realizacji projektu.

2. Planowanie i dokumentacja:

Szkic projektu: sporządź plan, który

obejmuje etapy realizacji projektu i kluczowe funkcje.

Dokumentacja: twórz dokumentację projektu, która będzie pomocna dla Ciebie lub dla innych osób pracujących nad projektem.

3. Elastyczność i dostosowywanie się:

Bądź elastyczny: projektowanie i rozwijanie projektu to proces dynamiczny. Bądź gotowy na dostosowywanie się do zmieniających się potrzeb i okoliczności.

4. Regularne testowanie i optymalizacja:

Testuj projekt: regularne testowanie pomoże zidentyfikować błędy i obszary, które wymagają ulepszeń.

Optymalizacja: stosuj optymalizacje w celu poprawy wydajności i funkcjonalności projektu.

5. Kreatywne podejście do rozwiązywania problemów:

Kreatywność: bądź otwarty na różnorodne podejścia i nie bój się eksperymentować w celu znalezienia najlepszych rozwiązań.

6. Wsparcie społeczności i wiedza zespołu:

Wsparcie społeczności: korzystaj z wiedzy innych poprzez forum internetowe, grupy dyskusyjne, społeczności programistyczne itp.

Praca zespołowa: jeśli pracujesz w zespole, buduj zgrany zespół, który będzie się wzajemnie wspierał i dzielił wiedzą.

7. Odpowiedzialność i etyka:

Odpowiedzialność: bądź odpowiedzialny za swoje działania, terminy realizacji projektu oraz za jakość końcowego produktu.

Etyka: pamiętaj o zasadach etycznych, szczególnie w dziedzinach, gdzie Twoje rozwiązania mogą mieć wpływ na innych ludzi lub środowisko.

8. Ciągłe uczenie się i rozwój:

Nauka i rozwój: technologia stale się zmienia, dlatego nieustanne poszerzanie wiedzy i doskonalenie umiejętności są kluczowe dla skutecznego prowadzenia projektów.

Przestrzeganie tych kluczowych punktów może pomóc w zapewnieniu sukcesu projektu oraz w utrzymaniu wysokiej jakości pracy podczas jego realizacji. Każdy projekt to okazja do nauki i rozwoju, dlatego warto zwracać uwagę na te aspekty podczas pracy nad nim.

11 Podsumowanie

11.2. Inspiracja do dalszego eksplorowania świata komputerów neuronowych



Oto kilka inspirujących punktów do dalszego eksplorowania świata komputerów neuronowych:

1. Praktyczne projekty:

Projekty do samodzielnego wykonania: spróbuj zaimplementować proste modele sieci neuronowych do rozpoznawania obrazów, klasyfikacji tekstu lub tworzenia prostych gier. To świetny sposób na zrozumienie działania sieci neuronowych.

2. Zasoby online i kursy:

Kursy online: Udemy, Coursera, Khan Academy i wiele innych platform oferuje kursy dotyczące komputerów neuronowych i uczenia maszynowego. Możesz również korzystać z darmowych materiałów na platformach takich jak YouTube czy Medium.

3. Wspólnoty i fora:

Aktywność w społeczności: dołącz do grup dyskusyjnych, społeczności online lub forum związanych z uczeniem maszynowym. Udział w dyskusjach i wymiana doświadczeń z innymi pasjonatami może być bardzo inspirująca.

4. Konferencje i wydarzenia:

Udział w konferencjach: wielu konferencji i wydarzeń poświęconych sztucznej inteligencji oferuje wykłady, warsztaty i możliwość nawiązania kontaktów z ekspertami w dziedzinie.

5. Rozwiązywanie problemów:

Rozwiązywanie problemów: zajmij się konkursami, wyzwaniem lub problemami związanymi z uczeniem maszynowym. Platformy takie jak Kaggle oferują konkursy z nagrodami, które mogą być inspirujące do dalszego rozwoju.

6. Literatura i artykuły:

Czytanie książek i artykułów: Wiele książek, artykułów naukowych i blogów jest poświęconych

tematyce uczenia maszynowego. Czytanie na ten temat może być inspirujące i pogłębiające wiedzę.

7. Eksperymentowanie i twórcze podejście:

Eksperymentowanie: nie bój się eksperymentować z różnymi technikami, modelami i danymi. Czasem najbardziej kreatywne podejścia prowadzą do innowacyjnych rozwiązań.

8. Analiza istniejących aplikacji i projektów:

Analiza istniejących rozwiązań: przejrzyj istniejące aplikacje lub projekty wykorzystujące komputery neuronowe. Zobacz, jak są one zbudowane i jak działają, aby wyciągnąć inspirację do własnych działań.

Eksplorowanie świata komputerów neuronowych jest ekscytującą podróżą, pełną nieustannego rozwoju i nowych odkryć. Różnorodność dostępnych zasobów i możliwości pozwoli Ci na eksperymentowanie i rozwój w tej fascynującej dziedzinie.



5. Wagi (Weights) i Bias:

Wagi są parametrami, które regulują siłę sygnału przekazywanego przez połączenia między neuronami w sieci. Bias jest dodatkowym parametrem, który zmienia równanie wyjściowe neuronu.

6. Funkcja aktywacji (Activation Function):

Funkcja aktywacji kontroluje, czy dane wejściowe przekazywane do neuronu mają być aktywowane (wygenerowane wyjście) czy nie. Przykłady funkcji aktywacji to ReLU (Rectified Linear Activation), sigmoid czy tanh.

7. Przeuczenie (Overfitting) i Niedouczenie (Underfitting):

- Przeuczenie:** Model uczy się na zbyt małym zbiorze danych treningowych, co powoduje, że staje się zbyt dopasowany do tych konkretnych danych i traci zdolność do generalizacji.

- Niedouczenie:** Model jest zbyt prosty lub zbyt ograniczony, przez co nie potrafi dobrze odwzorować zależności w danych treningowych ani generalizować wzorców w danych testowych.

8. Zbiór treningowy, walidacyjny i testowy:

- Zbiór treningowy:** Zbiór danych używany do treningu modelu.
- Zbiór walidacyjny:** Zbiór danych używany do oceny skuteczności modelu podczas jego treningu.
- Zbiór testowy:** Zbiór danych używany do ostatecznej oceny skuteczności modelu po treningu.

9. Gradient Descent:

Jest to algorytm optymalizacyjny, który wykorzystuje pochodne do aktualizacji wag w sieci neuronowej, aby minimalizować funkcję błędów.

Kluczowe pojęcia związane z komputerami neuronowymi:

1. Neuron:

Neuron w kontekście komputerów neuronowych jest podstawową jednostką przetwarzającą informacje. Symuluje on działanie neuronów w ludzkim mózgu poprzez przyjmowanie, przetwarzanie i przekazywanie danych.

2. Sieć neuronowa:

Sieć neuronowa to zbiór połączonych ze sobą neuronów, które pracują wspólnie w celu przetwarzania danych. Składa się z warstw, w których neurony przekazują informacje poprzez połączenia zwane wagami.

3. Uczenie maszynowe (Machine Learning):

To dziedzina sztucznej inteligencji, w której modele komputerowe są uczące się na podstawie danych. Komputer "uczy się" na podstawie doświadczeń i jest w stanie dokonywać predykcji lub podejmować decyzje na podstawie tego, co nauczył się z danych.

4. Uczenie nadzorowane i nienadzorowane:

- Uczenie nadzorowane:** Jest to rodzaj uczenia maszynowego, w którym algorytm jest uczony na podstawie danych zawierających pary wejście-wyjście. Algorytm próbuje przewidzieć wyjście na podstawie danych treningowych.

- Uczenie nienadzorowane:** W tym rodzaju uczenia maszynowego algorytm jest uczony na podstawie danych, ale bez etykietowanych wyjść. Algorytm sam próbuje znaleźć wzorce lub zależności w danych.

12 | Dodatki

12.2. Źródła i zalecane materiały do dalszej nauki

Wideo i materiały multimedialne:

1.YouTube: Kanały takie jak "3Blue1Brown", "Sentdex", "Stanford University" oferują wyjaśnienia i kursy związane z uczeniem maszynowym i sieciami neuronowymi.

2.Podcasty: Posłuchaj podcastów poświęconych uczeniu maszynowemu, takich jak "Talking Machines" czy "Data Skeptic".

Kursy na platformach e-learningowych:

1.Udemy: Oferuje szeroki wybór kursów z uczenia maszynowego i głębokiego uczenia.

2.Pluralsight: Platforma oferuje kursy dla programistów i specjalistów IT w obszarze uczenia maszynowego.

Pamiętaj, że kluczem do skutecznego nauki jest praktyka. Wykorzystaj dostępne materiały do nauki teorii, ale również podejmuj się praktycznych projektów, aby pogłębić swoje zrozumienie i zdobyć praktyczne umiejętności w dziedzinie uczenia maszynowego i sieci neuronowych.

Kursy online:

1.Coursera: Kursy oferowane przez uniwersytety na platformie Coursera, takie jak "Machine Learning" prowadzony przez Andrew Ng.

2.Udacity: Udacity oferuje kursy, w tym "Deep Learning" oraz "Artificial Intelligence for Trading".

3.edX: Na platformie edX znajdziesz kursy oferowane przez uniwersytety i instytucje takie jak MIT, UC San Diego, np. "Deep Learning Explained".

Materiały do samodzielnego nauki:

1.Książki: "Deep Learning" autorstwa Ian Goodfellow, Yoshua Bengio i Aaron Courville, "Python Machine Learning" autorstwa Sebastian Raschka i "Neural Networks and Deep Learning" autorstwa Michael Nielsen.

2.Artykuły i blogi: Czytaj artykuły naukowe opublikowane w czasopismach takich jak "Nature", "Science" oraz śledź blogi takie jak "Towards Data Science", "Distill.pub" czy "OpenAI Blog".

Platformy do praktycznych ćwiczeń:

1.Kaggle: Udostępnia konkursy, zbiory danych oraz notatniki, które pozwalają na praktyczne ćwiczenia z uczenia maszynowego.

2.GitHub: Na GitHubie znajdziesz repozytoria z kodem źródłowym różnych projektów związanych z uczeniem maszynowym, w tym implementacje sieci neuronowych.



Kto ćwiczy ten staje się mistrzem...

12

Dodatki

12.3. Przykłady prostych projektów do samodzielnego wykonania

Oto kilka prostych projektów, które można wykonać samodzielnie w obszarze komputerów neuronowych i uczenia maszynowego:

1. Klasyfikator obrazów:

Rozpoznawanie cyfr: stwórz prosty model, który będzie rozpoznawał ręcznie pisane cyfry ze zbioru danych MNIST.

Klasyfikacja zwierząt: utwórz model, który będzie klasyfikował obrazy różnych zwierząt, wykorzystując zbiór danych ze zwierzętami.

2. Prosta gra:

Gra w kółko i krzyżyk: Zbuduj model, który będzie grał w kółko i krzyżyk z graczem, wykorzystując uczenie maszynowe.

Gra Flappy Bird: Zaimplementuj model, który będzie uczył się grać w grę Flappy Bird, a następnie zobacz, jak dobrze radzi sobie w tej grze.

3. Analiza tekstu:

Klasyfikacja sentymentu: Stwórz model, który będzie klasyfikował teksty jako pozytywne lub negatywne, na podstawie emocji wyrażonych w tekście.

Generowanie tekstu: Wykorzystaj modele językowe do stworzenia prostego generatora tekstu, który będzie tworzył teksty na podstawie zadanego wzorca.

4. Przetwarzanie danych:

Przewidywanie cen akcji: Wykorzystaj dane historyczne i zbuduj model do przewidywania cen akcji na rynku finansowym.

Analiza danych klienci: Zastosuj uczenie nienadzorowane do grupowania klientów na podstawie ich zachowań i preferencji.

5. Detekcja obiektów:

Detekcja twarzy: Wykorzystaj gotowe modele do detekcji twarzy na obrazach lub wideo.

Detekcja ruchu: Stwórz prosty model do detekcji ruchu na podstawie sekwencji obrazów.

6. System rekomendacji:

Rekomendacje filmów/książek: Stwórz prosty system rekomendacji, który sugeruje filmy lub książki na podstawie preferencji użytkownika.

Rekomendacje produktów: Utwórz system rekomendacji produktów dla użytkowników na podstawie ich wcześniejszych zakupów lub zachowań.

Projekty te oferują dobrą okazję do eksperymentowania z różnymi technikami uczenia maszynowego i sieciami neuronowymi. Pamiętaj, że praktyka i eksperymentowanie są kluczowe w nauce w tych obszarach. W miarę zdobywania doświadczenia w prostych projektach, będziesz mógł podjąć się bardziej zaawansowanych zadań.

Dziękuję za uwagę

Waldemar Dylewski