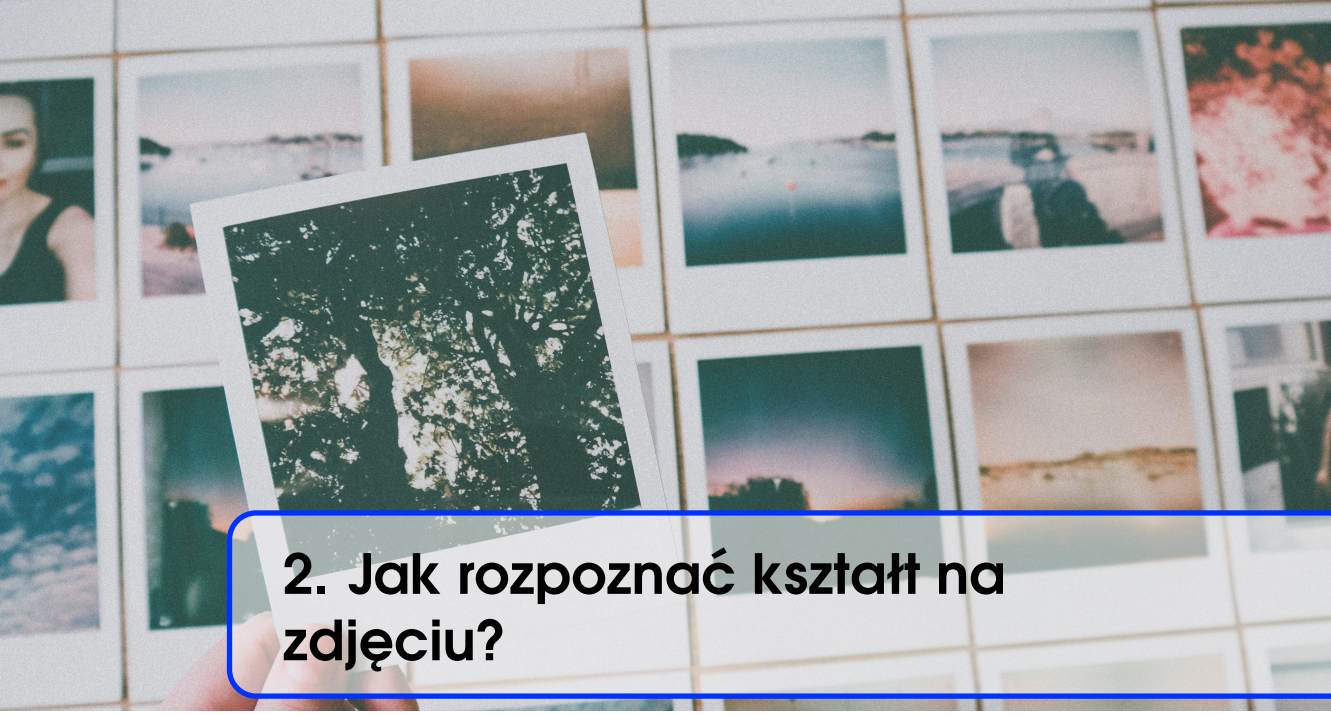


2. Jak rozpoznać kształt na zdjęciu?

Łukasz Woźniak

2.1 Człowiek istota społeczna

Człowiek - istota społeczna - od zawsze żył w rzeczywistości wizualnej. Świadczą o tym rysunki w jaskiniach z epoki kamienia łupanego. Nasuwają się więc następujące pytania. Jak postrzegamy obrazy? W jaki sposób mózg interpretuje to, co widzi? Jak przetwarzamy informację wizualną?

Procesy poznawcze, kształtujące człowieka już w bardzo wczesnym etapie rozwoju, skupiają się na obrazach. Dziecko rozumie oraz rozpoznaje obraz matki, jeszcze zanim zacznie rozumieć słowo „matka". Obrazy stanowią nieodłączną sferę życia, wobec czego bezsprzeczny jest fakt, że tekst w bardzo małym stopniu jest w stanie z nimi rywalizować. Przetwarzanie grafiki to sposób, w jaki mózg człowieka analizuje i przekazuje informacje zwane percepcją wzrokową lub informacją wizualną. Dzięki tej zdolności odbiorca interpretuje i przetwarza znaczenie informacji wizualnych, które dostarcza mu wzrok. Tak więc postrzeganie wizualne odgrywa ogromną rolę w codziennym życiu. Dlatego też niezwykle istotne jest zrozumienie, w jaki sposób interpretowane są obrazy. Może okazać się to przydatne w procesie projektowania komputerowego systemu rozpoznawania i przetwarzania obrazu.

Wybitny psycholog Albert Mahrabian w latach 70. ubiegłego wieku w pozycji „Silent Messages" pisał, że komunikacja niewerbalna stanowi

Norway grants

Projekt pt.: „MODELOWE ROZWIĄZANIA NA TRUDNE WYZWANIA – Plan Rozwoju Lokalnego i Instytucjonalnego Stalowej Woli" współfinansowany jest ze środków Norweskiego Mechanizmu Finansowego 2014-2021 (85%) oraz Budżetu Państwa (15%).

Wspólnie działamy na rzecz Europy zielonej, konkurencyjnej i sprzyjającej integracji społecznej.

93% całej komunikacji człowieka, jak również informacje wizualne stanowią 90% wszystkich informacji przesyłanych do mózgu. Obrazy są przetwarzane przez mózg dużo szybciej niż tekst. Szacunkowo ludzie pamiętają 80% z tego, co widzą, a jedynie 20% z tego, co czytają. Zespół neurologów z Instytutu Technologicznego w Massachusetts (MIT) odkrył, że mózg jest w stanie przetwarzać całe obrazy, które ludzkie oko widzi przez zaledwie 13 milisekund. Człowiek może zatem zidentyfikować kilkanaście, połączonych konkretną koncepcją, obrazów migających w ułamku sekundy. Zespół naukowców z Uniwersytetu w Toronto w pracy na temat neuronalnych korelacji epizodycznego kodowania obrazów i słów (Neural correlates of the episodic encoding of pictures and words) dowiódł, że ludzie są w stanie zapamiętać 2000 zdjęć z dokładnością przynajmniej 90% przez okres kilku dni, nawet przy bardzo krótkim czasie ich prezentacji podczas nauki. Badania wskazują zatem, że człowiek posiada dużo doskonalszą pamięć do zapamiętywania treści wizualnych niż tekstu pisanego, co może wynikać z faktu, że dzięki obrazom automatycznie pojawiają się powiązania z inną wiedzą o świecie, czyli bardziej skomplikowanym kodowaniem niż w przypadku słów.

2.2 Informacja wizualna

Informacja wizualna to podstawowa postać informacji komunikowania się człowieka z otoczeniem, jak również odbierania informacji z otoczenia. Zmysł wzroku pozwala na odbieranie bodźców świetlnych. Umożliwia dostrzeganie i rozróżnianie przedmiotów, skalowanie ich wielkości i formy, przestrzennego usytuowania, jak również rozpoznawanie barwy i ruchu. W tym wypadku narzędziem poznawczym jest oko, natomiast receptorami czopki i pręciki tworzące siatkówkę oka. **Wzrok** to szczególny zmysł, który pozwala człowiekowi na pobieranie aż 80% informacji z otoczenia. Dzięki temu człowiek jest w stanie poznawać świat, głównie w postaci informacji wizualnej. Postacie, w jakich możemy je odczytać są typem informacji:

- obrazowej – informacja pierwotna, nieuwarunkowana znaczeniowo;
- tekstowej – informacja symboliczna, uwarunkowana kulturowo.

Zmysł wzroku przetwarza bodźce świetlne - oddziałujące na oko - na potencjały czynnościowe komórek nerwowych i przekazuje je do mózgu, gdzie są przetwarzane, a w konsekwencji interpretowane. Takie działanie pozwala na odbieranie informacji o świecie, rozróżnianie kształtu, ruchu, emocji oraz wrażeń estetycznych.

Norway grants
Projekt pt.: „MODELOWE ROZWIĄZANIA NA TRUDNE WYZWANIA – Plan Rozwoju Lokalnego i Instytucjonalnego Stalowej Woli" współfinansowany jest ze środków Norweskiego Mechanizmu Finansowego 2014-2021 (85%) oraz Budżetu Państwa (15%).
Wspólnie działamy na rzecz Europy zielonej, konkurencyjnej i sprzyjającej integracji społecznej.

Podstawowym czynnikiem pozwalającym na widzenie przedmiotów jest światło. Bez światła nie jest możliwe zaobserwowanie barwy przedmiotu ani jego kształtu. Pionierem w tej dziedzinie był Izaak Newton, który już w 1666 roku odkrył, że światło słoneczne stanowi mieszaninę siedmiu **barw widmowych**:

1. czerwonej,
2. pomarańczowej,
3. żółtej,
4. zielonej,
5. niebieskiej,
6. granatowej,
7. fioletowej.

Każda z barw widmowych ma inną długość fali. Razem jednak stanowią światło białe, które w momencie, gdy trafia do detektora - takiego jak oko - może zostać pochłonięte lub odbite i wywołać wrażenie barwy. Zatem barwa obiektu zależy od światła padającego i światła odbitego. Przedmioty, które odbijają cały zakres widma rozpoznawane są jako białe. Natomiast te, które pochłaniają cały zakres widma odbierane są jako czarne. Subiektywne mieszanie barw odbitych, polegające na nakładaniu się na siebie barw składowych światła białego, decyduje zatem o rejestrowaniu barw powierzchni.

■ **Przykład 2.1** W tym wypadku przykładem mogą być zielone liście. Zawierają one chlorofil, który intensywnie pochłania barwę fioletową. Białe światło odbite od powierzchni liści jest pozbawione koloru fioletowego. Wywołuje to wrażenie zieleni i właśnie dlatego liście są postrzegane jako koloru zielonego (rys. 2.1).

Rysunek 2.1: Postrzeganie koloru liści

■

Norway grants
Projekt pt.: „MODELOWE ROZWIĄZANIA NA TRUDNE WYZWANIA – Plan Rozwoju Lokalnego i Instytucjonalnego Stalowej Woli" współfinansowany jest ze środków Norweskiego Mechanizmu Finansowego 2014-2021 (85%) oraz Budżetu Państwa (15%).
Wspólnie działamy na rzecz Europy zielonej, konkurencyjnej i sprzyjającej integracji społecznej.

2.3 Uczenie maszynowe

Pisanie aplikacji - związanych ze sztuczną inteligencją - na potrzeby widzenia komputerowego wymaga podstawowej wiedzy z **uczenia maszynowego** oraz **uczenia głębokiego**. Podczas tworzenia kodu aplikacji łączymy koncepcje uczenia maszynowego i uczenia głębokiego z elementami języka naturalnego oraz odwzorowaniem zachowań ludzkich dla danego problemu. Stworzone w wyniku takiego działania oprogramowanie jest czymś w rodzaju sztucznej inteligencji.

Uczenie maszynowe wyewoluowało od tradycyjnego podejścia do programowania. Tradycyjne programowanie przebiega etapowo i polega na opracowaniu struktury reguł wyrażonych w języku programowania, pobraniu danych, przetworzeniu ich, a w konsekwencji zwróceniu wyniku. Podejście takie jest wykorzystywane w prawie każdej sytuacji, gdy zaprogramowanie pewnej sekwencji pozwala osiągnąć zamierzony wynik. Ma ono jednak swoje ograniczenia. Rozwiązanie jest możliwe do osiągnięcia tylko wówczas, gdy jest możliwość zdefiniowania reguł.

■ **Przykład 2.2** Weźmy pod uwagę przykład z ceną i zyskiem sklepu. Mamy dostęp do danych dotyczących ceny produktów i zysku netto ze sprzedaży. Możemy wyznaczyć wówczas wskaźnik atrakcyjności zakupu na podstawie obrotu danym produktem. Kod wczytuje cenę zakupu oraz zysk netto i zwraca wynik, który jest efektem dzielenia dwóch składników. Odpowiedź jest wynikiem operowania na danych przez zadanie reguł. Powyższa sytuacja odpowiada większości tworzonych programów zdefiniowanych za pomocą reguł. W innych przypadkach kod byłby zbyt złożony, aby móc rozwiązać takie problemy. ■

■ **Przykład 2.3** Inny przykład może stanowić sytuacja rozpoznawania aktywności fizycznej. Mając do dyspozycji prędkość, jesteśmy w stanie określić czy osoba spaceruje, czy biegnie. Oczywiście wymaga to wprowadzenia reguł określających zakres prędkości dla danej aktywności. Możemy założyć, że mamy odczynienia ze spacerem w przypadku, gdy prędkość jest mniejsza od 6 km/h, powyżej jest to już bieg. Co w przypadku, gdy osoba jedzie na rowerze? Zakładamy tu również regułę, że czynność ta jest wykonywana przy prędkości powyżej 12 km/h. Mamy więc zdefiniowane 3 czynności, ale załóżmy, że chcemy poszerzyć tą bazę i wprowadzić do niej grę w golfa. Aktywność ta wymaga jednocześnie spacerowania, przemieszczania się i czasem biegania. Z zaistniałych reguł nie jesteśmy w stanie wywnioskować czy dana aktywność jest właśnie grą w golfa. Co w wypadkach, gdy nie jesteśmy w stanie określić reguł? Wówczas musimy odwrócić sytuację i przypisać

Projekt pt.: „MODELOWE ROZWIĄZANIA NA TRUDNE WYZWANIA – Plan Rozwoju Lokalnego i Instytucjonalnego Stalowej Woli" współfinansowany jest ze środków Norweskiego Mechanizmu Finansowego 2014-2021 (85%) oraz Budżetu Państwa (15%).
Norway grants
Wspólnie działamy na rzecz Europy zielonej, konkurencyjnej i sprzyjającej integracji społecznej.

dane wyjściowe łącznie z etykietami. Dzięki temu możemy brać pod uwagę nowe przypadki - na tym właśnie polega uczenie maszynowe. Schemat działania takiego oprogramowania jest inny. Zastanówmy się, co powinniśmy zmienić, by uzyskać poprawną odpowiedź. Wiemy, że każdy program potrzebuje danych, na których ma operować. Wiemy również, jakie rozwiązania chcemy uzyskać, a więc w takim wypadku niewiadomą są reguły, jakie muszą zaistnieć. To podejście nazywamy **uczeniem maszynowym**. ■

Podsumowując, procesem wstępnym jest etykietowanie danych, a wyjściowym dopasowanie reguł, które pasują do już zaetykietowanych danych. Dzięki temu, podczas wykonywania różnych aktywności, otrzymujemy reguły, które jednoznacznie definiują typ.

Dziedziny sztucznej inteligencji są rozległym i abstrakcyjnym tematem. W tym wypadku uczenie maszynowe możemy zakwalifikować jako **sztuczną inteligencję**, gdyż dzięki niej maszyna może rozpoznawać świat tak jak człowiek. W dalszej części zajmiemy się rozwojem rozpoznawania obrazu z wykorzystaniem uczenia maszynowego.

2.4 Widzenie komputerowe

Podczas opracowywania przetwarzania obrazu za pomocą Machine Lerningu, posłużymy się platformą TensorFlow. Jest to platforma oparta na zasadzie Open Source, w szczególności przeznaczona do tworzenia modeli uczenia maszynowego. Wewnątrz biblioteki zostało zaszytych wiele algorytmów oraz wzorców, wobec czego użytkownik może się skupić na rozwiązaniu problemu. Może być użyta do szerokiego zakresu zadań, ale skupia się w szczególności na uczeniu i wnioskowaniu sieci neuronowych. Ma wbudowanych wiele typowych algorytmów i wzorców, co w konsekwencji pozwala się skupić przede wszystkim na rozwiązaniu problemu, a nie na poszukiwaniu drogi do rozwiązania. Pozwala również na obsługę przetwarzania obrazu, dźwięku, tekstu oraz wielu innych zagadnień. W dalszej części skupimy się na przetwarzaniu obrazu, a w szczególności na rozpoznawaniu elementów garderoby na zdjęciach.

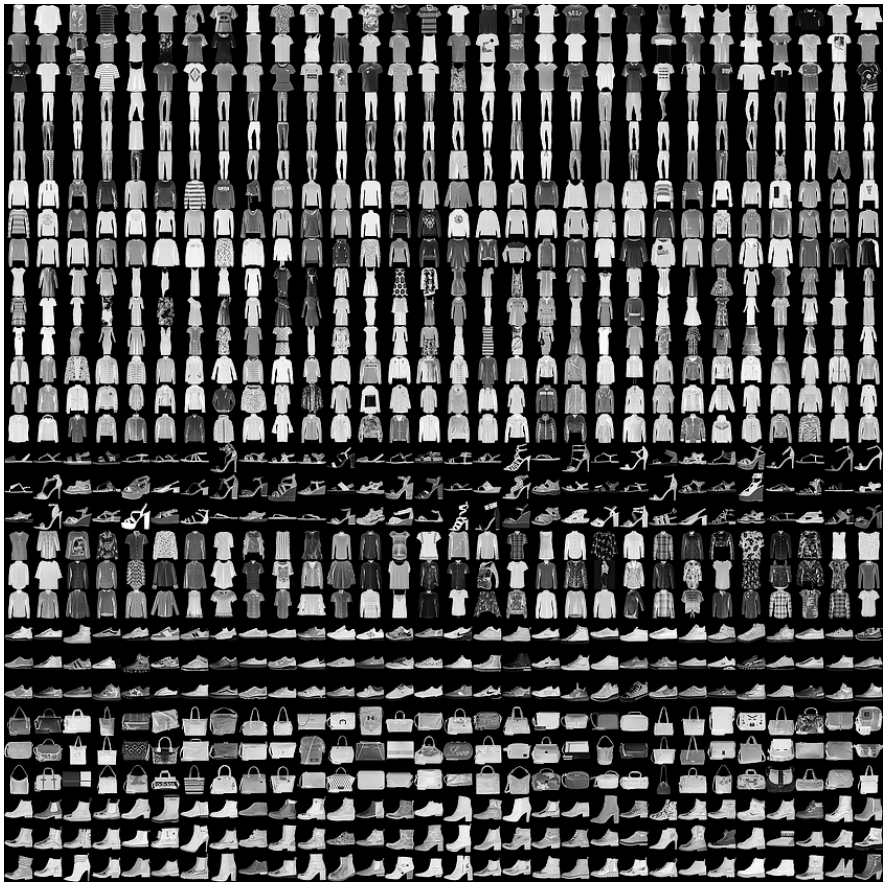
Proces tworzenia i przygotowania danych dla modeli uczenia maszynowego, podczas którego jednostka obliczeniowa korzysta z szeregu algorytmów (w celu rozpoznania danych wejściowych oraz sposobów ich rozróżniania), nazywamy **trenowaniem**. Wynikiem jest proces rozpoznawania lub kategoryzowania danych wejściowych, który jest nazywany **wnioskowaniem**.

Norway grants
Projekt pt.: „MODELOWE ROZWIĄZANIA NA TRUDNE WYZWANIA – Plan Rozwoju Lokalnego i Instytucjonalnego Stalowej Woli" współfinansowany jest ze środków Norweskiego Mechanizmu Finansowego 2014-2021 (85%) oraz Budżetu Państwa (15%).
Wspólnie działamy na rzecz Europy zielonej, konkurencyjnej i sprzyjającej integracji społecznej.

Zastanówmy się chwilę, czego musielibyśmy użyć, aby móc rozpoznawać elementy garderoby. W tym celu możemy posłużyć się zdjęciami, które pozwolą nam na utworzenie modelu. Model taki, aby był dobrze wytrenowany, musi przetworzyć wiele obrazów, by osiągnąć jak najlepszy wynik.

2.5 Wykrywanie cech w obrazach

W tej części zajmiemy się właściwym widzeniem komputerowym, w którym model nauczy się rozpoznawać elementy garderoby. W tym celu przygotujemy obrazy zawarte w postaci gotowej biblioteki danych oraz zaprogramujemy sieć neuronową tak, aby dopasować dane do etykiet oraz poznamy reguły potrzebne do rozróżniania elementów.

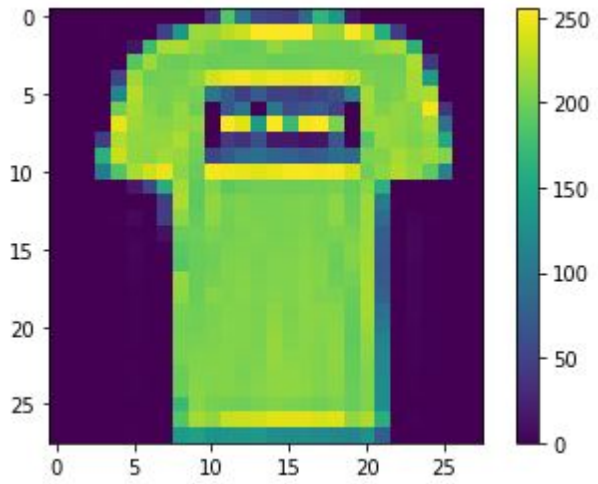


Rysunek 2.2: Zbiór Fashion-MNIST (MIT Licencja)

Norway grants
Projekt pt.: „MODELOWE ROZWIĄZANIA NA TRUDNE WYZWANIA – Plan Rozwoju Lokalnego i Instytucjonalnego Stalowej Woli" współfinansowany jest ze środków Norweskiego Mechanizmu Finansowego 2014-2021 (85%) oraz Budżetu Państwa (15%).
Wspólnie działamy na rzecz Europy zielonej, konkurencyjnej i sprzyjającej integracji społecznej.

Czym zatem jest widzenie komputerowe? Jak wytłumaczyć je komputerowi? Zastanów się, w jaki sposób podczas wielu lat rozpoznawano obrazy i kształtowano ich nazwy. Wobec czego, jak elementy garderoby były rozpoznawane w czasie oraz jak je selekcjonowano i dobierano. Tę samą drogę możemy zastosować dla komputera, jednak wymaga ona nałożenia innych ograniczeń. W tym celu posłużymy się zbiorem danych przewidzianym do uczenia i analizy porównawczej algorytmów Fashion MNIST. Zbór posiada 70 tysięcy obrazów (60 tys. obrazów treningowych, 10 tys. obrazów testowych) w skali szarości od 0 do 255 i rozmiarze 28×28 pikseli, podzielone na 10 kategorii. Upraszczając rozumowanie, będziemy rozpoznawać na obrazie podstawowe 10 różnych rodzajów garderoby, takie jak: koszule, spodnie, sukienki czy buty.

Jak widać na rysunku 2.2, każdy pojedynczy element z prostokątnych elementów siatki przedstawia wycinek ze zbioru danych biblioteki i zawiera się w rozmiarze 28×28 pikseli. Skala szarości poszczególnych pikseli mieści się w przedziale od 0 do 255. Na rysunku 2.3 pokazano jeden z obrazów biblioteki Fashion MNIST.



Rysunek 2.3: Jeden z elementów zbioru Fashion-MNIST

Przeanalizujmy teraz, jakie informacje możemy odczytać z danego zbioru danych. Każdy obraz to złożenie 784 wartości (28×28), a każdy piksel mieści się w przedziale od 0 do 255 w skali szarości. Przyjmijmy zatem, że te niewiadome stanowią wartość X. Kolejną informacją jest zbiór, który posiada 10 różnych typów obrazów, a zatem jest to wartość Y. W dalszej części chcemy się dowiedzieć, w jaki sposób Y jest funkcją

Norway grants
Projekt pt.: „MODELOWE ROZWIĄZANIA NA TRUDNE WYZWANIA – Plan Rozwoju Lokalnego i Instytucjonalnego Stalowej Woli" współfinansowany jest ze środków Norweskiego Mechanizmu Finansowego 2014-2021 (85%) oraz Budżetu Państwa (15%).
Wspólnie działamy na rzecz Europy zielonej, konkurencyjnej i sprzyjającej integracji społecznej.

X. W tym celu stworzymy większą ilość neuronów, z których każdy zostanie nauczony parametrów, wobec czego powstanie syntetyczna funkcja. Takie podejście zapewni nam możliwość dopasowania wzorca do pożądanej odpowiedzi.

Etykieta	Nazwa
0	T-shirt
1	Trouser
2	Pullover
3	Dress
4	Coat
5	Sandal
6	Shirt
7	Sneaker
8	Bag
9	Ankle boot

Tablica 2.1: Numery 10 typów obrazów

Podczas trenowania sieci neuronowej wczytujemy do każdego neuronu z osobna wartości X, natomiast wszystkie wagi M oraz przesunięcia C zostają zainicjalizowane losowo. Podczas sumowania wartości wyjściowych dla każdego z neuronów, każdy z nich wygeneruje pewną wartość. Działanie to jest wykonane dla wszystkich neuronów w warstwie wyjściowej. Wobec czego neuron zerowy poda wartość prawdopodobieństwa, z jaką piksele odpowiadają etykiecie zerowej, neuron pierwszy poda wynik prawdopodobieństwa dla etykiety pierwszej i tak po kolei dla każdego neuronu. W końcowym etapie będziemy chcieli dopasować uzyskaną wartość do żądanego wyniku. Poszukiwanym obrazem jest koszulka but.

Przyjrzyjmy się i przeanalizujmy kod programu przedstawiony na rysunku 2.4. W niniejszym artykule będziemy opisywać strukturę programu w języku Python. Na początek dodajemy obsługę biblioteki TensorFlow oraz obsługę pakietu Kears. Sam pakiet Kears zawiera obsługę wbudowanych zbiorów danych, w tym zbiory treningowe i testowe. Pakiet został wykorzystany na potrzeby niniejszego opracowania, w celu przedstawienia jak najprostszej formy obsługi uczenia maszynowego, aby nie obciążać odbiorcy nadmierną liczbą nowych nazw i koncepcji.

Norway grants
Projekt pt.: „MODELOWE ROZWIĄZANIA NA TRUDNE WYZWANIA – Plan Rozwoju Lokalnego i Instytucjonalnego Stalowej Woli" współfinansowany jest ze środków Norweskiego Mechanizmu Finansowego 2014-2021 (85%) oraz Budżetu Państwa (15%).
Wspólnie działamy na rzecz Europy zielonej, konkurencyjnej i sprzyjającej integracji społecznej.

```
# TensorFlow and tf.keras
import tensorflow as tf

# Helper libraries
import numpy as np
import matplotlib.pyplot as plt

fashion_mnist = tf.keras.datasets.fashion_mnist

(train_images, train_labels),
(test_images, test_labels) = fashion_mnist.load_data()

train_images = train_images / 255.0
test_images = test_images / 255.0

model = tf.keras.Sequential([
    tf.keras.layers.Flatten(input_shape=(28, 28)),
    tf.keras.layers.Dense(128, activation='relu'),
    tf.keras.layers.Dense(10)
])

model.compile(optimizer='adam',
              loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True),
              metrics=['accuracy'])

model.fit(train_images, train_labels, epochs=10)
```

Rysunek 2.4: Kod programu

Kolejna linijka kodu odpowiada za uzyskanie zbiorów treningowych i testowych (rys. 2.5).

```
(train_images, train_labels),
(test_images, test_labels) = fashion_mnist.load_data()
```

Rysunek 2.5: Kod programu

Jako wynik operacji - *data.load_data* otrzymujemy dwa zbiory danych. Pierwszy, treningowy zbiór danych stanowi tablica o nazwie *training_images*, która zawiera w swojej strukturze 60 tys. tablic treningowych, wczytane rysunki o rozmiarach 28x28 pikseli, a także tablicę *training_labels*, która zawiera 60 tys. wartości w przedziale od 0 do 9. Drugi testowy zbiór stanowi tablica o nazwie *test_images*, która zawiera w swojej strukturze 10 tys. tablic testowych, wczytane rysunki o rozmiarach 28×28 pikseli, a także tablicę *training_labels*, która zawiera 10 tys. wartości w przedziale od 0 do 9.

Zadaniem, jakie stoi przed nami, jest dopasowanie obrazów treningowych do etykiet treningowych, aby uzyskać jak najlepsze dopasowanie.

Norway grants
Projekt pt.: „MODELOWE ROZWIĄZANIA NA TRUDNE WYZWANIA – Plan Rozwoju Lokalnego i Instytucjonalnego Stalowej Woli" współfinansowany jest ze środków Norweskiego Mechanizmu Finansowego 2014-2021 (85%) oraz Budżetu Państwa (15%).
Wspólnie działamy na rzecz Europy zielonej, konkurencyjnej i sprzyjającej integracji społecznej.

```
train_images = train_images / 255.0
test_images = test_images / 255.0
```

Rysunek 2.6: Kod programu

Przy tym zabezpieczymy sieć na wypadek dostępu do danych testowych. Następnym krokiem jest normalizowanie obrazu. Polega ono na tym, że podzielimy każdą tablicę przez wartość 255, co odpowiada poziomowi szarości, a w konsekwencji da reprezentację pojedynczego piksela wartością 0 lub 1. Proces ten przeprowadza się w celu polepszenia wydajności w czasie trenowania sieci neuronowej (rys. 2.6). Następnie definiujemy sieć neuronową, która tworzy model (rys. 2.7). Część ta została opisana poniżej.

```
model = tf.keras.Sequential([
    tf.keras.layers.Flatten(input_shape=(28, 28)),
    tf.keras.layers.Dense(128, activation='relu'),
    tf.keras.layers.Dense(10)
])
```

Rysunek 2.7: Kod programu

Przyszedł czas na najważniejszą część, definiującą kompilacje modelu, gdzie wybieramy funkcję straty i optymalizator (rys. 2.8). Wybór odpowiedniego optymalizatora wiąże się z odpowiednią wiedzą na temat rozwiązywania problemów sztucznej inteligencji. W zadanym przypadku posłużymy się funkcją straty, którą jest rzadka kategoryczna entropia krzyżowa. Polega ona na tym, że zamiast próbować przewidywać pojedynczą liczbę, która ma być rozwiązaniem, ustalamy kategorię.

```
model.compile(optimizer='adam',
              loss=tf.keras.losses.
              SparseCategoricalCrossentropy(from_logits=True),
              metrics=['accuracy'])
```

Rysunek 2.8: Kod programu

Podobnie rzecz się ma do wyboru optymalizatora. Ważne jest, aby pracował on z odpowiednią wydajnością, gdyż będziemy analizować

Norway grants
Projekt pt.: „MODELOWE ROZWIĄZANIA NA TRUDNE WYZWANIA – Plan Rozwoju Lokalnego i Instytucjonalnego Stalowej Woli" współfinansowany jest ze środków Norweskiego Mechanizmu Finansowego 2014-2021 (85%) oraz Budżetu Państwa (15%).
Wspólnie działamy na rzecz Europy zielonej, konkurencyjnej i sprzyjającej integracji społecznej.

zbiór danych o wielkości 60 tys. obrazów treningowych. W naszym wypadku zastosujemy optymalizator adam, który jest ulepszeniem stochastycznego spadku wzdłuż gradientu.

```
model.fit(train_images, train_labels, epochs=10)
```

Rysunek 2.9: Kod programu

Kolejna linijka programu odpowiada za dopasowanie - w czasie dziesięciu epok - obrazów treningowych do etykiet treningowych (rys. 2.9), po czym uruchamiamy skrypt.

Rysunek 2.10: Zbiór pojedynczych grafik

Na koniec możemy ocenić dokładność uczenia sieci neuronowej. W tym celu przekażemy do wytrenowanego modelu obrazy i etykiety przeznaczone do testów (rys. 2.10), aby móc prognozować, co widzi na każdym z obrazów.

Norway grants
Projekt pt.: „MODELOWE ROZWIĄZANIA NA TRUDNE WYZWANIA – Plan Rozwoju Lokalnego i Instytucjonalnego Stalowej Woli" współfinansowany jest ze środków Norweskiego Mechanizmu Finansowego 2014-2021 (85%) oraz Budżetu Państwa (15%).
Wspólnie działamy na rzecz Europy zielonej, konkurencyjnej i sprzyjającej integracji społecznej.

2.6 Rozpoznawanie kształtów

Dotarliśmy do ostatniego punktu. Przystąpimy więc do uruchomienia naszego programu, który zaczyna trenować sieć - epoka po epoce. Na koniec uzyskujemy wynik, jak widać na rysunku 2.11.

```
Epoch 1/10
1875/1875 [==============================] - 5s 2ms/step - loss: 0.5003 - accuracy: 0.8246
Epoch 2/10
1875/1875 [==============================] - 4s 2ms/step - loss: 0.3746 - accuracy: 0.8648
Epoch 3/10
1875/1875 [==============================] - 4s 2ms/step - loss: 0.3367 - accuracy: 0.8788
Epoch 4/10
1875/1875 [==============================] - 4s 2ms/step - loss: 0.3126 - accuracy: 0.8854
Epoch 5/10
1875/1875 [==============================] - 4s 2ms/step - loss: 0.2943 - accuracy: 0.8922
Epoch 6/10
1875/1875 [==============================] - 4s 2ms/step - loss: 0.2805 - accuracy: 0.8961
Epoch 7/10
1875/1875 [==============================] - 4s 2ms/step - loss: 0.2681 - accuracy: 0.9009
Epoch 8/10
1875/1875 [==============================] - 4s 2ms/step - loss: 0.2561 - accuracy: 0.9048
Epoch 9/10
1875/1875 [==============================] - 4s 2ms/step - loss: 0.2484 - accuracy: 0.9082
Epoch 10/10
1875/1875 [==============================] - 4s 2ms/step - loss: 0.2401 - accuracy: 0.9111
<keras.callbacks.History at 0x7fd2f99f6490>
```

Rysunek 2.11: Trenowanie zbioru

Model, korzystając z danych treningowych, uzyskał dokładność bliską 91% podczas obliczeń w zaledwie 10 epokach. Spójrzmy teraz jak wygląda sytuacja z danymi testowymi.

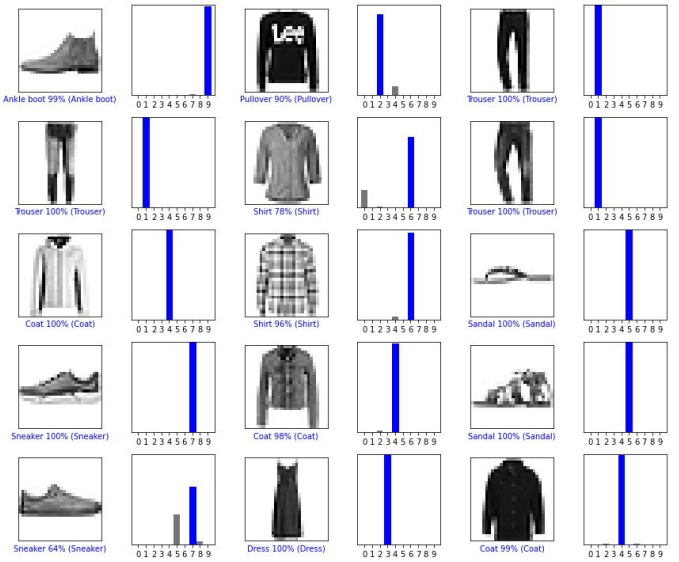
```
313/313 - 1s - loss: 0.3366 - accuracy: 0.8812 - 529ms/epoch - 2ms/step

Test accuracy: 0.8812000155448914
```

Rysunek 2.12: Trenowanie zbioru testowego

Tu, po jednej epoce, otrzymaliśmy wynik 88,12%, co jest wynikiem gorszym od wyniku dla danych treningowych, lecz nie odbiega znacząco od niego. Różnica taka zdarza się dość często. Wynika to z faktu, że podczas uczenia sieci neuronowej, potrafi ona dobrze dopasować dane wejściowe do danych wyjściowych, za pomocą których została ona wytrenowana.

Samo uczenie przebiega na zasadzie wnioskowania na podstawie przeanalizowanych przykładów, gdzie proces rozpoznaje jak wygląda dany element, taki jak sweter czy but (rys. 2.13).

Norway grants
Projekt pt.: „MODELOWE ROZWIĄZANIA NA TRUDNE WYZWANIA – Plan Rozwoju Lokalnego i Instytucjonalnego Stalowej Woli" współfinansowany jest ze środków Norweskiego Mechanizmu Finansowego 2014-2021 (85%) oraz Budżetu Państwa (15%).
Wspólnie działamy na rzecz Europy zielonej, konkurencyjnej i sprzyjającej integracji społecznej.



Rysunek 2.13: Trenowanie zbioru testowego

2.7 Analiza wyników

Znamy już dokładność wytrenowania modelu podczas użycia zbioru testowego. Teraz zastanówmy się, jak przeprowadzane jest testowanie i wnioskowanie. Otrzymany zbiór klasyfikacji, który otrzymaliśmy po przekazaniu obrazów testowych do metody model.predict został przedstawiony poniżej.

```
array([1.2184351e-07, 6.0797112e-10, 1.5039088e-08, 1.7623728e-09,
       1.9998125e-09, 1.3366874e-03, 1.8967589e-07, 3.8073987e-02,
       3.3286156e-08, 9.6058887e-01], dtype=float32)
```

Rysunek 2.14: Wynik dla trenowanej sieci

Otrzymana tablica zawiera wyniki dla 10 neuronów wyjściowych. Przypisana etykieta jest faktycznym odzwierciedleniem klasy elementu odzieży. Wartości oznaczają prawdopodobieństwo, z jakim dany obraz odpowiada danej etykiecie. Wobec czego uzyskujemy informację, że z prawdopodobieństwem wynoszącym 96,05% element z indeksem 0 będzie miał etykietę równą 9. Dokonując oceny wiemy, że ten element ma taką etykietę, wobec czego prognoza jest prawidłowa.

Powyższe wyniki odnoszą się do przypadku, w którym trenowaliśmy sieć tylko przez czas dziesięciu epok. Oznacza to w praktyce, że

Norway grants
Projekt pt.: „MODELOWE ROZWIĄZANIA NA TRUDNE WYZWANIA – Plan Rozwoju Lokalnego i Instytucjonalnego Stalowej Woli" współfinansowany jest ze środków Norweskiego Mechanizmu Finansowego 2014-2021 (85%) oraz Budżetu Państwa (15%).
Wspólnie działamy na rzecz Europy zielonej, konkurencyjnej i sprzyjającej integracji społecznej.

pięć razy wykonaliśmy całą pętlę treningową. W tym czasie losowo zainicjalizowaliśmy neurony porównaniu ich z etykietami, sprawdzeniu wydajności za pomocą funkcji straty oraz wykonaniu aktualizacji. Otrzymane wyniki są całkiem dobre i wynoszą 91% dla danych treningowych i 88,12% dla danych testowych.

Norway grants

Projekt pt.: „MODELOWE ROZWIĄZANIA NA TRUDNE WYZWANIA – Plan Rozwoju Lokalnego i Instytucjonalnego Stalowej Woli" współfinansowany jest ze środków Norweskiego Mechanizmu Finansowego 2014-2021 (85%) oraz Budżetu Państwa (15%).

Wspólnie działamy na rzecz Europy zielonej, konkurencyjnej i sprzyjającej integracji społecznej.