POLITECHNIKA WROCŁAWSKA
WYDZIAŁ ELEKTRONIKI

KIERUNEK: Informatyka

SPECJALNOŚĆ: Grafika i Systemy Multimedialne

PRACA DYPLOMOWA MAGISTERSKA

Detekcja manipulacji zawartości zdjęć przy pomocy metod uczenia głębokiego

Image manipulation detection using deep learning techniques

AUTOR:
Jarosław Ciołek-Żelechowski

PROMOTOR:
dr inż. Paweł Ksieniewicz

OCENA PRACY:

WROCŁAW 2020

SPIS TREŚCI

1. WSTĘP

2. UCZENIE MASZYNOWE

Alan Turing w swojej pracy z 1950 roku *Computing Machinery and Inteligence*[14] zdefiniował pojęcie *obiekcji lady Lovelace*. Odnosiło się ono do krótkiej notatki[6] jaką lady Ada Lovelace poczyniła w 1843 roku podczas tłumaczenia na język angielski artykułu Luigi Menabrea[8], który to był streszczeniem wykładu Charlesa Babbage'a wygłoszonego w Turynie w 1841 roku. Wykład dotyczył projektu maszyny analitycznej której zadaniem było zautomatyzowanie niektórych obliczeń związanych z analizą matematyczną. Pojęcie to brzmi następująco:

Maszyna analityczna nie ma na celu zapoczątkowania czegokolwiek. Może wykonywać operacje, które możemy kazać jej przeprowadzać... Jej celem jest zwiększanie dostępności tego co umiemy już wykonać[14]

Turing, przywołał to pojęcie, zastanawiając się nad tym, czy komputery mogą się uczyć, tworzyć nowe rzeczy. Jak piszę on dalej w swoim artykule[14]: problem jest natury programistycznej i wymaga on stworzenia zupełnie innego, jak na tamte czasy, środowiska i sposobu pojmowania nauczania jako takiego. Wierzył jednak, że jest to możliwe.

Termin *Uczenie Maszynowe* po raz pierwszy pojawił się w 1959 roku w pracy naukowej autorstwa Arthura Samuela:

Uczenie maszynowe to dziedzina nauki dająca komputerom możliwość uczenia się bez konieczności ich jawnego programowania.[12]

Praca ta dotyczyła maszyny, która przez ok. 8 godzin *uczyła się* gry w warcaby znając jedynie jej zasady, posiadając pewnego rodzaju funkcję celu(zbijanie pionów przeciwnika), oraz macierz losowych liczb, której to zmiany i przekształcenia miały na celu reprezentować inne podejścia(nową wiedzę). Sprawdzianem sukcesu maszyny, było pokonanie jej twórcy.

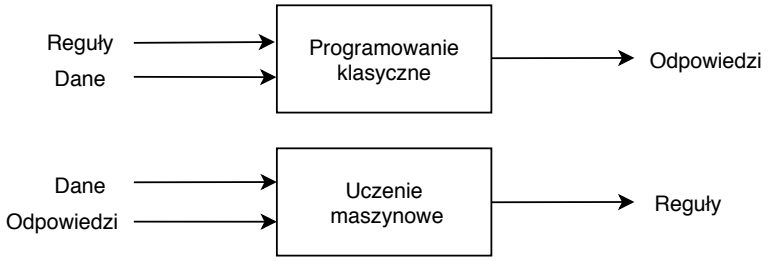
Bardziej techniczną definicję podał w 1997 roku Tom Mitchel, w rozdziale otwierającym swoją książki pt. *Machine Learning*:

Mówimy, że program komputerowy uczy się na podstawie doświadczenia E w odniesieniu do jakiegoś zadania T i pewnej miary wydajności P, jeśli jego wydajność (mierzona przez P) wobec zadania T wzrasta wraz z nabywaniem doświadczenia E.[9]

Tym samym uczenie maszynowe zostało sprowadzone do problemu, w którym to posiadamy trzy elementy T, P i E. W dalszej części Tom Mitchel podaje przykład powyżej definicji zrealizowanej dla gry w warcaby(wyraźny ukłon w stronę pracy Arthura Samuela[12]):
— Zadanie T: gra w warcaby,
— Miara Wydajności P: procent gier wygranych na oponentów,
— Doświadczenie E: granie partii przeciwko samemu sobie.

Tym samym można rozumieć uczenie maszynowe jako nowy paradygmat programowania. W programowaniu klasycznym, programista definiuje reguły według których program przetwarza dane wejściowe, generujący tym samym dane wyjściowe - odpowiedź pracy programu(patrz

rysunek 2.1). W przypadku uczenia maszynowego mamy sytuację w której programista wprowadza dane oraz odpowiedzi i oczekuje uzyskać od programu zestawu reguł, według których dane odpowiedzi zostały przypisane do konkretnych próbek. Reguły te, mają posłużyć w dalszej części do przetwarzania nowych danych.



Rys. 2.1. Uczenie maszynowe: nowy model programowania

Podsumowując, system uczenia maszynowego jest trenowany, a nie programowany w sposób jawny. Celem programisty jest przedstawienie mu odpowiedniej dużej ilości przykładów wyników, tak by sam system określił ich statystyczną strukturę, co w dalszej części pozwoli na ustalenie reguł umożliwiających automatyzację całego procesu. Ważne, by podkreślić tutaj znaczenie danych wejściowych, które to często określa się terminem *Big Data*[13].

2.1. BIG DATA

Popularyzację tego terminu przypisuję się do wykładu autorstwa Johna Mashey'a[7] jaki wygłosił w 1998 roku. Zauważył on, że wraz ze spadkiem cen nośników do przechowywania i gromadzenia danych, ilość zbieranych przez ludzkość informacji wzrasta z każdym rokiem. Co więcej, ilość tych danych sprawia, że ich analiza przestała być możliwa do realizacja w sposób inny niż automatyczny. Za kryterium, czy dany zbiór informacji można określić jako Big Data, często przywołuje się te podane przez Douglasa Laney'ego[4] i określane mianem *3V*, które to rozwija się jako rozmiar(ang. *volume*), różnorodność(ang. *variety*) i prędkość (ang. *velocity*). Oznaczają one kolejno:

— **rozmiar** - dane są zbyt duże by mieściły się na standardowych dyskach twardych,
— **różnorodność** - dane pochodzą z różnych źródeł, są niejednorodne i słabo ustrukturyzowane,
— **prędkość** - tempo napływu nowych danych jest znaczące, co w rezultacie utrudnia ich analizowanie.

Jednym z kluczowych zjawisk przyczyniającym się do procesu nagłego przyrostu ilości i źródeł danych jest Internet przedmiotów (ang. *Internet of Things*[3]). Według tej koncepcji, różnego typu urządzenia osadzone w urządzeniach codziennego użytku(np. odkurzacze, żarówki, instalacje grzewcze), czy też w maszynach przemysłowych, zbierają dane o otoczeniu, czy samym procesie w którym uczestniczą, a ponadto mają możliwość komunikacji pomiędzy sobą, ale również z jednostką centralną jeśli taka występuje. Wprowadza to nowy wymiar w możliwościach automatyzacji procesów i tworzy nową przestrzeń do tworzenia się złożonych, inteligentnych systemów.

2.2. RODZAJE SYSTEMÓW UCZENIA MASZYNOWEGO

Istnieje wiele metod podziału uczenia maszynowego na kategorię. Jedną z najpopularniejszych jest podział ze względu na sposób uczenia się oraz zadanie do wykonania[11]:
— Uczenie nadzorowane:
  — klasyfikacja,
  — regresja.
— Uczenie nienadzorowane:
  — klasteryzacja,
  — redukcja wymiarowości,
  — uczenie przy użyciu reguł asocjacyjnych.
— Uczenie ze wzmocnieniem.

W uczeniu z nadzorem(ang. *supervised learning*), którym zajmę się w poniżej pracy, dane trenujące przekazane algorytmowi zawierają dołączone do nich etykiety, czyli rozwiązania problemu. Celem algorytmu jest stworzenie funkcji(nazywanej też hipotezą[1]), która będzie maksymalizować swoją skuteczność względem zadanych kryteriów.

Na zbiór uczący Z_u składa się zbiór n wektorów, gdzie każdy z nich opisuje pojedynczy obiekt(patrz wzór 2.1):

$$Z_u = \{(x_1, y_1), ..., (x_n, y_n)\} \tag{2.1}$$

I tak w powyższym równaniu2.1) i-ty wektor oznaczony byłby jako x_i i odpowiadałaby mu etykieta oznaczona jako y_i. W zależności od typu danych przechowywanych pod etykietą będziemy mieć do czynienia z zadaniem klasyfikacji(etykieta należy do skończonego i przeliczalnego zbioru) lub regresji(wartości przyjmowane przez etykietę należą do przestrzeni ciągłej). Ważne żeby dodać że na wektor x_i może składać się m-liczb, gdzie każdą z tych liczb będziemy nazywać atrybutem lub cechą danego wektora(patrz wzór 2.2):

$$x_i = \{x_{1,m}, ..., x_{i,m}\} \tag{2.2}$$

Etykietą y_i, w tym rozumowaniu, oznaczamy prawdziwą wartość funkcji, którą to chcemy by nasz algorytm odwzorował. Algorytm ten, nazywamy również modelem i opisujemy go jako następującą funkcję f(patrz wzór 2.3):

$$f(x) : x \in X \mapsto y \in Y \tag{2.3}$$

Jak widać zadaniem powyższej funkcji jest przyporządkowanie wektorom wejściowym etykiet. Jej skuteczność jest mierzona przy użyciu wybranej przez nas metryki na zbiorze testowym. Zbiór testowy musi posiadać tą samą strukturę co zbiór trenujący. Można traktować metrykę M jako funkcję wyższego rzędu, której pierwszym argumentem jest sam model f, a drugim zbiór testowy Z_t. Dziedziną metryki jest zazwyczaj podzbiór liczb rzeczywistych z zakresu od 0 do 1(patrz wzór 2.4)

$$M(f, Z_t) : (f, Z_t) \mapsto m \in [0, 1] \tag{2.4}$$

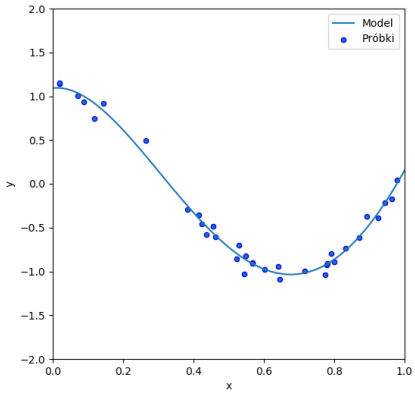
Przykładem metryk stosowanych w uczeniu maszynowym są między innymi dokładność, czułość, precyzja, a także miara *F* (ang. *F-score*), które omówione zostały szczegółowo w dalszej części rozdziału.

2.3. PROBLEM PRZEUCZANIA, NIEDOUCZANIA

Wykorzystanie osobnego zbioru do badania jakości modelu w uczeniu nadzorowanym związane jest bezpośrednio z takimi problemami jak nadmierne dopasowanie, przeuczenia(ang. *overfitting*) oraz niedouczenie (ang. *underfitting*)[2]. Zadaniem, które dobrze nadaje się do graficznej ilustracji powyższych problemów jest regresja liniowa. Jak już zostało opisane powyżej w problemie tym, etykiety przykładów są liczbami należącymi do przestrzeni ciągłej, a zadaniem algorytmu jest wyznaczenie funkcji f(patrz wzór 2.5):

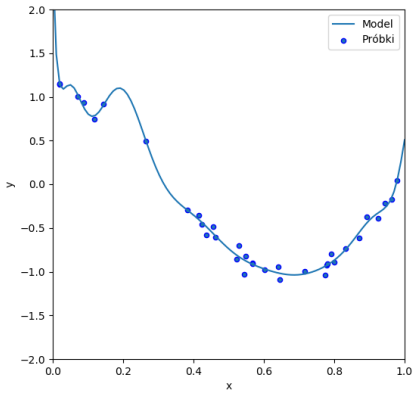
$$f(x) : x \mapsto y \in \mathbb{R} \tag{2.5}$$

Dodatkowo dla lepszej interpretacji graficznej każdy wektor w przestrzeni x będzie posiadał tylko jedną cechę. Przykład prawidłowego dopasowania wygląda następująco(patrz rysunek 2.2):



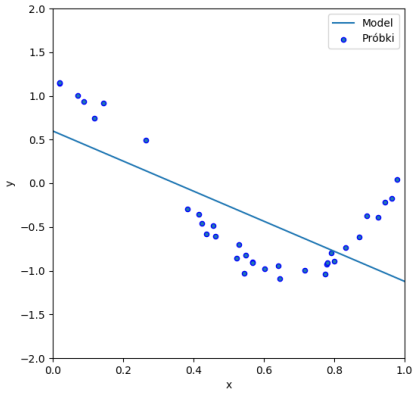
Rys. 2.2. Przykład prawidłowego dopasowania

Problem nadmiernego dopasowania cechuje się tym, że taki model zbyt mocno generuje charakterystykę dla danych trenujących, zawierając w niej również występujące tam szumy(patrz rysunek 2.3):



Rys. 2.3. Przykład nadmiernego dopasowania

Odwrotnym problem jest problem niedouczenia. Występuję on na przykład wtedy, kiedy zastosujemy zbyt prosty model w stosunku do poziomu skomplikowania zbioru danych. Wynik takiej operacji widoczny jest na rysunku 2.4



Rys. 2.4. Przykład niedostatecznego dopasowania

2.4. ZADANIE KLASYFIKACJI BINARNEJ

W niniejszej pracy będę zajmował się zadaniem należącym do problemów klasyfikacji binarnej, czyli takiej w której przestrzeń etykiet ogranicza się do dwóch elementów(patrz wzór 2.6):

$$y = \{0, 1\} \tag{2.6}$$

Klasy 0 i 1 w powyższym wzorze2.6 są przykładowe i bardziej niż ich wartość interesuje Nas liczność zbioru y. Tym samym mając zdefiniowaną przestrzeń możliwych wartości modelu można zdefiniować szereg miar, które są wykorzystywane do oceny jego wyników. Podstawowym elementem zadania ewaluacji modelu w zadaniu klasyfikacji binarnej jest macierz, tablica błędów widoczna w tabeli 2.1. Do jej skonstruowania potrzebujemy oczywiście przetestować działanie naszego modelu na zbiorze testowym Z_t. Poszczególnym reprezentantom tego zbioru, przypisywane są pozytywne lub negatywne etykiety, teraz w zależności od tego czy dany element zbioru x był faktycznie pozytywny czy negatywny można go wpisać w macierz błędów 2.1.

	Klasyfikacja pozytywna	**Klasyfikacja negatywna**
Stan pozytywny	Prawdziwie dodatnia (ang. *true positive*, TP)	Fałszywie ujemna (ang. *false negative*, FN)
Stan negatywny	Fałszywie dodatnia (ang. *false positive*, FP)	Prawdziwie ujemna (ang. *true negative*, TN)

Tabela 2.1. Tablica pomyłek, możliwe wyniki klasyfikacji binarnej

2.5. EWALUACJA MODELU - MIARY JAKOŚCI

Korzystając z opisanej powyżej macierzy błędów 2.1 w łatwy sposób można przedstawić definicję szeregu miar do oceny modelu, z których to skorzystałem w niniejszej pracy:

— **Dokładność** - procent poprawnych klasyfikacji, opisanych wzorem 2.7:

$$\mathbf{accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.7}$$

— **Czułość** - stosunek prawidłowych wyników pozytywnych do sumy prawidłowych wyników pozytywnych oraz błędnych wyników negatywnych, który można rozumieć jako zdolność modelu do poprawnego etykietowania(patrz wzór 2.8),

$$\mathbf{recall} = \frac{TP}{TP + FN} \tag{2.8}$$

— **Precyzja** - stosunek prawidłowych wyników pozytywnych do sumy prawidłowych wyników pozytywnych oraz błędnych wyników pozytywnych, który można rozumieć jako zdolność modelu do niepoprawnej klasyfikacji próbek negatywnych jako pozytywne(patrz wzór 2.9),

$$\mathbf{precision} = \frac{TP}{TP + FP} \tag{2.9}$$

— **miara *F*** - będąca średnią ważoną z czułości i precyzji(patrz wzór 2.10).

$$\mathbf{F1} = \frac{2 * precision * recall}{precision + recall} \tag{2.10}$$

2.6. SPLOTOWE SIECI NEURONOWE

Splotowe, inaczej konwolucyjne, sieci neuronowe (ang. *CNN - convolutional neural networks*) stanowią wynik badań nad korą wzrokową i od 1980 roku są używane w zadaniach rozpoznawania obrazów[2]. Podstawową różnicą w stosunku do sieci neuronowych jest stosowanie wielowymiarowej operacji splotu, realizowanej za pomocą szeregu filtrów, których to parametry dobierane są podczas trenowania sieci. Pozwala to siecią konwolucyjnym na nie przetwarzanie obrazów piksel po pikselu, a raczej poprzez zauważanie ogólnych cech obrazu i budowaniu z nich nowych struktur.

W analizie matematycznej operacją splotu(reprezentowana jako $*$) dwóch funkcji f i g, jest trzecia funkcja s, patrz wzór 2.11:

$$\begin{aligned} s(t) = (f * g)(t) &= \int_{-\infty}^{\infty} f(\tau) g(t - \tau) d\tau = \\ &= \int_{-\infty}^{\infty} f(t - \tau) g(\tau) d\tau \end{aligned} \tag{2.11}$$

, a w przypadku operacji dyskretnych, patrz wzór 2.12:

$$s(t) = (f * g)(t) = \sum_{\tau=-\infty}^{\infty} f(\tau)(g - \tau) \tag{2.12}$$

W literaturze opisującej sieci neuronowych przyjęło się określać $f(\tau)$ jako wejście, $g(t-\tau)$ jako jądro lub filtr(z ang. *kernel*, *filter*), a wynik samej operacji jako mapę atrybutów lub aktywacji (z ang. *feature map*, *activation map*)[10]. Zadanie dla typowego problemu klasyfikacji obrazów, operującego na dwuwymiarowym obrazie wejściowym M przy pomocy filtra K z założeniem wykorzystania wielowymiarowego splotu $S(i, j)$, można sformułować następująco 2.13:

$$\begin{aligned} S(i, j) = (M * K)(i, j) &= \sum_k \sum_l M(k, l) K(i - k, j - l) \\ = (K * M)(i, j) &= \sum_k \sum_l M(i - k, j - l) K(k, l) \end{aligned} \tag{2.13}$$

2.7. UCZENIE GŁĘBOKIE

Budowa sieci splotowych, a dokładnie fakt, że nie wymagały one połączeń pomiędzy wszystkimi neuronami w każdej z warstw oraz fakt, że zastosowanie operacji splotu, pozwala na zmniejszenie wymiarów zdjęcia pozwoliło budować sieci o większej ilości warstw ukrytych [11]. Za pierwszą pracę z stosującą uczenie głębokie(funkcje splotu, wsteczną propagację) uważa się tą 1989 roku, której autorzy stworzyli model rozpoznający kody pocztowe[5]. Sam model okazał się sukcesem, problemem był jednak czas uczenia - $\sim$3 dni. Z racji jednak że w ostatnim czasie moc obliczeniowa komputerów wzrosła kilkukrotnie, oraz dodatkowo same zbiory danych zwiększyły swoje rozmiary i jakość - uczenie głębokie staje coraz popularniejsze[2].

3. ANALIZA ISTNIEJĄCYCH METOD W OBRĘBIE DZIEDZINY

4. ZAŁOŻENIA METODOLOGICZNE

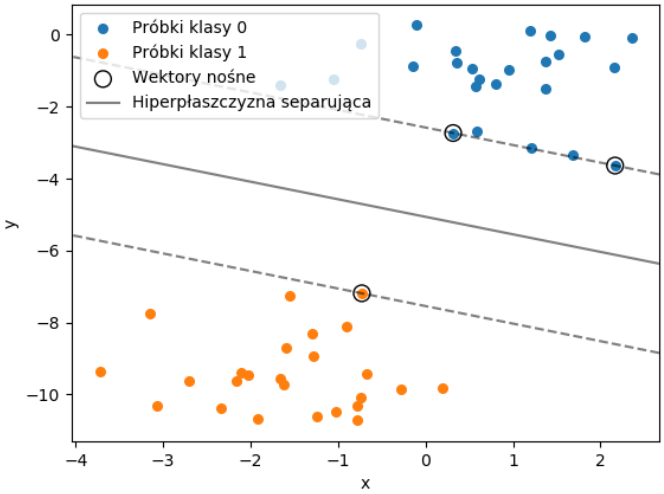
4.1. WYBRANIE ZBIORÓW DANYCH

4.2. STRATYFIKOWANA WALIDACJA KRZYŻOWA

4.3. PAROWE TESTY STATYSTYCZNE

4.4. OPIS WYKONANYCH EKSPERYMENTÓW

4.4.1. Maszyna wektorów nośnych(*SVM*)



Rys. 4.1. Przykład klasyfikacji maszyny wektorów nośnych

4.4.2. VGG Net

4.4.3. Autorska metoda wykorzystania sieci konwolucyjnej w detekcji falsyfikacji zdjęć

5. IMPLEMENTACJA I INTERPRETACJA WYNIKÓW BADAŃ

5.1. IMPLEMENTACJA ŚRODOWISKA RAMOWEGO WYKORZYSTUJĄCEGO ISTNIEJĄCE ARCHITEKTURY SIECI

5.2. IMPLEMENTACJA AUTORSKIEJ METODY WYKORZYSTANIA UCZENIA GŁĘBOKIEGO W ZADANIU KLASYFIKACJI

6. PODSUMOWANIE

BIBLIOGRAFIA

[1] Cichosz, P., *Systemy uczące się* (Wydawnictwo Naukowo-Techniczne, 2000).

[2] Géron, A., *Hands-On Machine Learning with Scikit-Learn and TensorFlow* (O'Reilly Media, 2017).

[3] Holler, J., Tsiatsis, V., Mulligan, C., Avesand, S., Karnouskos, S., Boyle, D., *From machineto-machine to the internet of things: Introduction to a new age of intelligence*, Academic Press. 2014.

[4] Laney, D., *3d data management: Controlling data volume, velocity, and variety*, tech. rep., META Group. 2001.

[5] LeCun, Y., Boser, B., Denker, J.S., Henderson, D., Howard, R.E., Hubbard, W., Jackel, L.D., *Backpropagation applied to handwritten zip code recognition*, Neural Computation. 1989.

[6] Lovelace, A., Menabrea, L.F., *Sketch of the analytical engine invented by charles babbage... with notes by the translator. translated by ada lovelace*, Scientific Memoirs. 1843.

[7] Mashey, J.R., *Big data... and the next wave of infrastress*, Slides from invited talk. 1998.

[8] Menabrea, L.F., *Sketch of the analytical engine invented by charles babbage*, Bibliothèque Universelle de Genève, No. 82. 1842.

[9] Mitchell, T.M., *Machine Learning* (McGraw-Hill, Inc., 1997).

[10] Piczak, K., *Klasyfikacja dźwięku za pomocą splotowych sieci neuronowych*, Prasa akademicka. 2018.

[11] Raschka, S., *Python Machine Learning* (Packt Publishing, 2015).

[12] Samuel, A.L., *Some studies in machine learning using the game of checkers*, IBM Journal of Research and Development(Volume: 3, Issue: 3). 1959.

[13] Singh, S., Singh, N., *Big data analytics*, Communication, Information Computing Technology (ICCICT). 2012.

[14] Turing, A.M., *Computing machinery and inteligence*, Mind(Volume: LIX, Issue: 236). 1950.

SPIS RYSUNKÓW

SPIS TABEL