Zastosowanie sieci neuronowych do przewidywania popularności filmików

Tomasz Bocheński

Politechnika Warszawska, wydział Elektroniki i Technik Informacyjnych,
Nowowiejska 15/19, 00-665 Warszawa, Polska
T.Bochenski@stud.elka.pw.edu.pl

Streszczenie. Sztuczne sieci neuronowe są szeroko stosowane w przypadku problemu klasyfikacji, prognozowania czy też analizy danych. Wiele różnych architektur tych sieci zostało zaprezentowanych i znalazło swoje wykorzystanie w różnych przypadkach. Konwolucyjne sieci neuronowe mają zastosowanie w klasyfikacji obrazów, a sieci rekurencyjne - w rozpoznawaniu mowy. W tej publikacji omówię podstawowe informacje związane z różnego rodzaju sieciami neuronowymi, a także przedstawię sposób ich wykorzystania, a dokładniej architektury konwolucyjnych sieci rekurencyjnych z pamięcią długotrwałą, do przewidywania popularności filmików. Wyniki badań przeprowadzonych na filmikach z serwisów internetowych takich jak Facebook, Youtube czy Dailymotion pokazują potencjał tego typu podejścia.

Słowa kluczowe: sztuczne sieci neuronowe, konwolucyjne sieci neuronowe, rekurencyjne sieci neuronowe, LRCN, LSTM, popularność filmików

1 Wprowadzenie

1.1 Inspiracje biologiczne

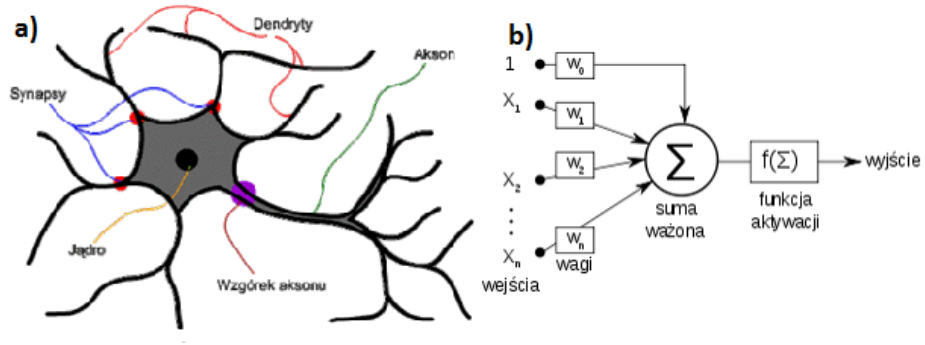
Sztuczne sieci neuronowe mają swoje inspiracje biologiczne w sposobie działania ludzkiego mózgu. W mózgu znajdują się neurony połączone w sieci, które odpowiedzialne są za inteligencję, emocje czy też zdolności twórcze. Na rys.1a zaprezentowany został schemat neuronu naturalnego z najważniejszymi, z punktu widzenia sztucznych sieci neuronowych, częściami składowymi. Są nimi:

- Dendryty, czyli wejścia neuronu;
- Synapsy, czyli zakończenia dendrytów, potrafiące modyfikować trafiający do nich sygnał osłabiając go lub wzmacniając;
- Jądro, czyli centrum obliczeniowe neuronu;
- Akson, czyli wyjście neuronu.

Na rys.1b przedstawiony został schemat sztucznego neuronu. Porównując go z neuronem naturalnym można zauważyć między nimi pewne podobieństwa:

- Wejścia są odpowiednikami dendrytów;

- Wagi cyfrowe są odpowiednikami modyfikacji dokonywanych na impulsach elektrycznych przez synapsy;
- Bloki sumy ważonej i funkcji aktywacji są odpowiednikiem jądra;
- Wyjście jest odpowiednikiem aksonu.



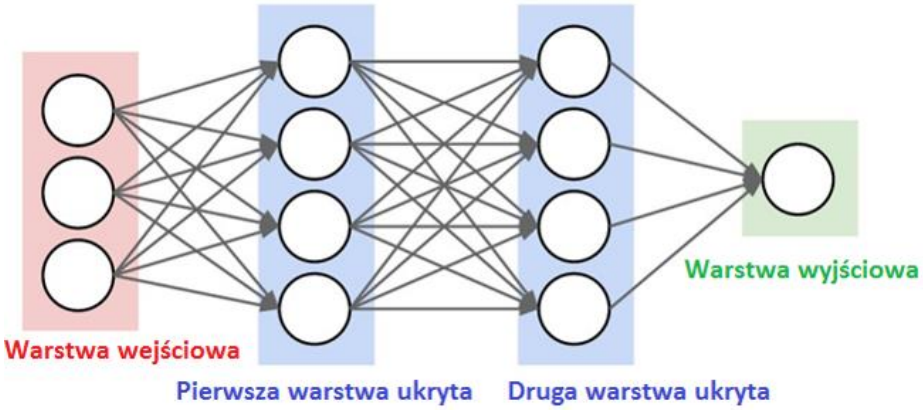
Rysunek 1a) Schemat budowy neuronu naturalnego. b) Schemat budowy neuronu sztucznego. [Źródło: Internet]

1.2 Działanie sztucznego neuronu

Sztuczny neuron to system przetwarzający wartości sygnałów wejściowych w pojedynczą wartość wyjściową. Dane wprowadzone na wejścia neuronu zostają przeskalowane przez wagę danego wejścia. Otrzymane w ten sposób wyniki cząstkowe są sumowane w bloku sumy ważonej. Wyznaczona suma służy jako argument pewnej funkcji, zwanej funkcją aktywacji. Jest to zwykle funkcja nieliniowa, np. funkcja sigmoidalna. Wynik tej funkcji przekazywany jest na wyjście neuronu.

1.3 Podstawy architektury sztucznych sieci neuronowych

Sztuczne sieci neuronowe składają się z warstw, natomiast warstwy (poza warstwą wejściową) z neuronów. W sieciach neuronowych można wyróżnić trzy typy warstw: warstwę wejściową, warstwy ukryte oraz warstwę wyjściową. Warstwa wejściowa reprezentuje dane wprowadzane na wejście sieci, natomiast warstwa wyjściowa wyznacza wyjście sieci. Wszystkie inne warstwy znajdujące się między nimi nazywane są warstwami ukrytymi. Warstwy ukryte oraz warstwa wyjściowa są warstwami w pełni połączonymi. Oznacza to, że neurony znajdujące się w tych warstwach są połączone ze wszystkimi wyjściami warstwy poprzedniej, natomiast połączenia między neuronami z tej samej warstwy nie istnieją. Rys.2 przedstawia przykładowy schemat sztucznej sieci neuronowej. Wagi połączeń pomiędzy neuronami a wyjściami poprzedniej warstwy są parametrami, które należy odpowiednio dobrać. Liczba wag przypadająca na jeden neuron jest równa ilości jego połączeń zwiększonej o 1. Drugi czynnik sumy wynika z faktu, że wejściem do każdego neuronu jest również stała niezależna od danych wejściowych. Nie została ona uwzględniona na rys.2.



Rysunek 2) Schemat sztucznej sieci neuronowej składającej się z czterech warstw: warstwy wejściowej, dwóch warstw ukrytych oraz warstwy wyjściowej. Zaprezentowana sieć zbudowana jest z 9 neuronów, a liczba wag które należy dobrać wynosi 41. [Źródło: Internet]

1.4 Uczenie sieci neuronowej

Aby sieć neuronowa funkcjonowała prawidłowo, konieczne jest dobranie dla niej odpowiednich wartości wag. Proces doboru tych wartości nazywany jest uczeniem sieci neuronowej. Aby nauka była możliwa potrzebna jest funkcja, która potrafiłaby porównać dane zestawy wag i wyznaczyć lepszy z nich. Funkcja taka nazywana jest funkcją kosztu. Określa ona jakość dobranych wag poprzez wyznaczenie błędu między wartością na wyjściu sieci a wartością, która na tym wyjściu powinna być.
Mając zdefiniowaną funkcję kosztu łatwo można zauważyć, że problem nauki sieci sprowadza się do problemu znalezienia minimum funkcji kosztu. Istnieje wiele sposobów na rozwiązanie tego problemu, jednak obecnie najpopularniejszym z nich jest metoda gradientu prostego. Polega ona na wyznaczeniu gradientu funkcji kosztu i odpowiednim uaktualnieniu wartości wag sieci na podstawie tego gradientu.

2 Konwolucyjne sieci neuronowe

2.1 Wstęp

W przypadku regularnych sieci neuronowych nie istnieją ograniczenia nakładane na dane wejściowe. Wektor pewnych danych podawany jest na wejście sieci. Następnie dane te poddawane są pewnym transformacjom przechodząc przez kolejne warstwy ukryte. Na końcu trafiają do warstwy wyjściowej, która wyprowadza je na wyjście sieci.
Regularne sieci neuronowe nie sprawdzają się jednak w przypadku, gdy danymi wejściowymi są obrazy (trójwymiarowe wolumeny posiadające długość, wysokość i szerokość). Przyczyną tego jest zbyt duża liczba połączeń między neuronami wynikająca z faktu, że wszystkie warstwy w regularnych sieciach neuronowych są warstwa-

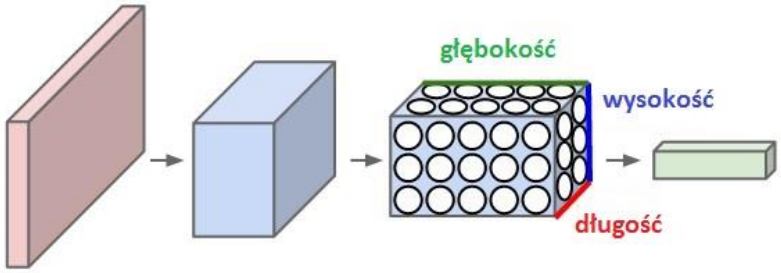
mi w pełni połączonymi. Stosowanie tego typu sieci do analizowania obrazów jest więc niepraktyczne.
Konwolucyjne sieci neuronowe stanowią rozwiązanie powyższego problemu. Danymi wejściowymi do konwolucyjnych sieci neuronowych mogą być tylko obrazy, dlatego sieci te są zoptymalizowane pod kątem ich przetwarzania.

2.2 Porównanie sieci konwolucyjnych z sieciami regularnymi

Po pierwsze konwolucyjne sieci neuronowe składają się ze znacznie większej ilości warstw niż regularne sieci neuronowe. W przypadku sieci regularnych ilość warstw mieści się zwykle w przedziale od 3 do 4. Sieci konwolucyjne nazywane są głębokimi sieciami neuronowymi, ponieważ posiadają od 10 do 20 warstw.
Po drugie w konwolucyjnych sieciach neuronowych neurony tworzą trójwymiarowe wolumeny. Trójwymiarowe ułożenie neuronów jest bardziej naturalne w przypadku przetwarzania obrazów, ponieważ same obrazy to tak naprawdę trójwymiarowe wolumeny o pewnej długości i wysokości oraz szerokości równej 3. Rys.3 przedstawia trójwymiarową strukturę konwolucyjnych sieci neuronowych.
Po trzecie regularne sieci neuronowe składają się tylko z warstw w pełni połączonych. Konwolucyjne sieci neuronowe są bardziej różnorodne i składają się z następujących typów warstw:

- Warstwy wejściowej, zawierającej surowy, jeszcze nie przetworzony obraz, czyli wolumen o pewnej długości i wysokości oraz szerokości równej 3;
- Warstw konwolucyjnych, realizujących główne zadanie jakim jest analizowanie obrazów;
- Warstw rektyfikowanej jednostki liniowej, przeprowadzających operacje zamiany wartości ujemnych na zera;
- Warstw próbkujących, które zajmują się próbkowaniem długość i wysokość wolumenów podawanych na ich wejścia, pozostawiając bez zmian ich trzeci wymiar;
- Warstw w pełni połączonych

Nie wszystkie z tych warstw zawierają połączenia, których wagi należy optymalizować. Zarówno warstwa konwolucyjna jak i w pełni połączona składają się z neuronów, a dodanie ich wiąże się z koniecznością optymalizowania większej ilości wag. Natomiast warstwa rektyfikowanej jednostki liniowej jak i warstwa próbkująca nie zawierają neuronów, zatem dodanie ich nie wpływa na ilość wag.



Rysunek 3) Schemat przedstawiający trójwymiarową strukturę konwolucyjnych sieci neuronowych. Można zauważyć, że każda warstwa ma swoją długość, wysokość oraz głębokość. [Źródło: Internet]

2.3 Opis warstwy konwolucyjnej

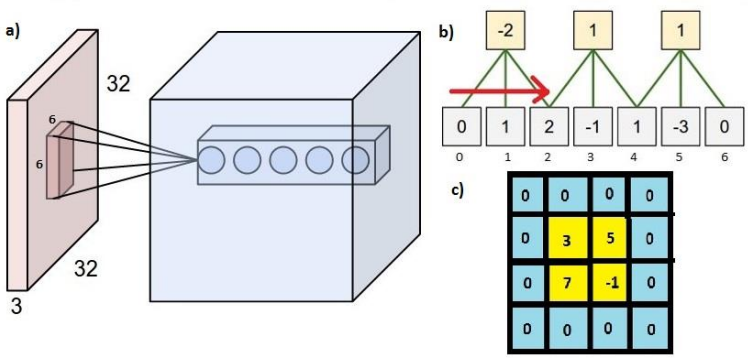
Najważniejszymi warstwami konwolucyjnych sieci neuronowych są warstwy konwolucyjne, które zajmują się wyszukiwaniem cech wizualnych w obrazach. Pomimo dużych rozmiarów danych wejściowych działają one efektywnie, co jest możliwe dzięki lokalności połączeń i współdzieleniu parametrów.
Lokalność połączeń mówi o tym, że połączenia neuronów są lokalne w przestrzeni (długość i wysokość), natomiast pełne jeśli chodzi o trzeci wymiar wolumenu, zwany głębokością. Z lokalnością połączeń związane są następujące hiperparametry:

- Wielkość filtru (F), która określa wymiary przestrzeni z jaką połączony jest pojedynczy neuron. Zostało to zobrazowane na rys.4a.
- Wielkość kroku (S), która określa odległość między lokalnymi obszarami obserwowanymi przez kolejne neurony. Zostało to zobrazowane na rys.4b.
- Wyrównanie (P), które określa o ile należy zwiększyć przestrzeń (długość i wysokość) wolumenu wejściowego, wstawiając w nowo powstałe miejsca wartości zero. Zostało to zobrazowane na rys.4c.

Znając wartości wszystkich hiperparametrów oraz wielkość wolumenu wejściowego (W) możliwe jest wyznaczenie wielkości wolumenu wyjściowego (OUT). Służy do tego wzór:

$$OUT = (W-F+2P)/S+1$$

Współdzielenie parametrów opiera się na założeniu, że jeśli jakaś wizualna cecha obrazu jest warta wyznaczenia w punkcie przestrzeni wolumenu, to warto jej również szukać w dowolnym innym punkcie przestrzeni tego wolumenu (czyli w dowolnym innym punkcie o tej samej trzeciej współrzędnej). W przypadku, gdy głębokość wolumenu z neuronami wynosi K, otrzymujemy K przestrzeni. W obrębie danej przestrzeni szukane są te same wizualne cechy obrazu, natomiast każda z przestrzeni szuka innych cech.



Rysunek 4a) Schemat połączeń pomiędzy warstwą konwolucyjna a danymi wejściowymi przy wielkości filtru równej 6. Niebieski prostopadłościan reprezentuje warstwę konwolucyjna, natomiast bryła czerwona dane wejściowe. [Źródło: Internet] b) Schemat przedstawiający połączenia pomiędzy kolejnymi neuronami a danymi wejściowymi przy wielkości kroku równej 2. Kwadraty żółte to neurony warstwy konwolucyjnej, natomiast kwadraty szare to dane wejściowe. [Źródło: Internet], c) Schemat przedstawiający powierzchnie wolumenu wejściowego przed zastosowaniem wyrównania równego 1 (żółty kwadrat) oraz po zastosowaniu tego wyrównania (kwadrat niebiesko-żółty).

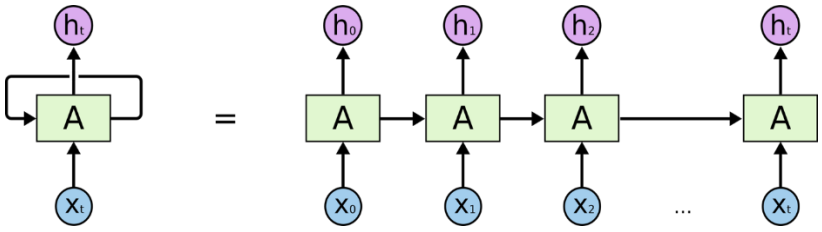
3 Rekurencyjne sieci neuronowe

3.1 Wstęp

Regularne sieci neuronowe charakteryzują się jednokierunkowym przepływem danych. Dane wejściowe wchodzą do sieci, są przez nią przetwarzane, następnie podawane są na wyjście sieci.

Rekurencyjne sieci neuronowe (RNN) wyróżniają się istnieniem sprzężeń zwrotnych między wejściem a wyjściem sieci. Można je zilustrować jako wiele kopii tej samej sieci, z których wyjście każdej kolejnej zależy od wyjść wszystkich poprzednich. Przedstawione to zostało na rys.5. Rekurencyjne sieci neuronowe wykorzystują informacje o danych wejściowych z przeszłości do aktualnie wykonywanych zadań. Jest to typ architektury stworzony do przetwarzania sekwencji.

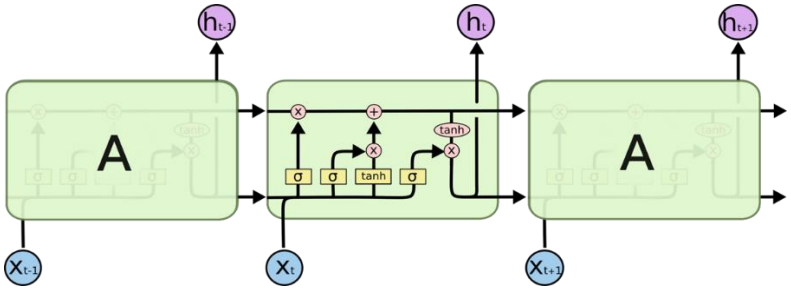
W praktyce rekurencyjne sieci neuronowe działają jedynie w przypadku, gdy zależności czasowe nie są duże. Wraz z ich wzrostem wyniki są coraz gorsze. Problem ten rozwiązują komórki pamięci krótko-długo trwałej (LSTM).



Rysunek 5) Schemat przedstawiający rekurencyjną sieć neuronową. [Źródło: Internet]

3.2 Komórki pamięci krótko-długo trwałej

Komórki pamięci krótko-długo trwałą (LSTM) to specjalny rodzaj rekurencyjnych sieci neuronowych, które potrafią poprawnie działać nawet wtedy, gdy zależności czasowe są duże. Schemat LSTM został przedstawiony na rys.6. Jak można zauważyć, komórki pamięci krótko-długo trwałej składają się z czterech warstw: trzech bramek sigmoidalnych oraz jednej bramki wykonującej operację nieliniową, zwykle tangensa hiperbolicznego.



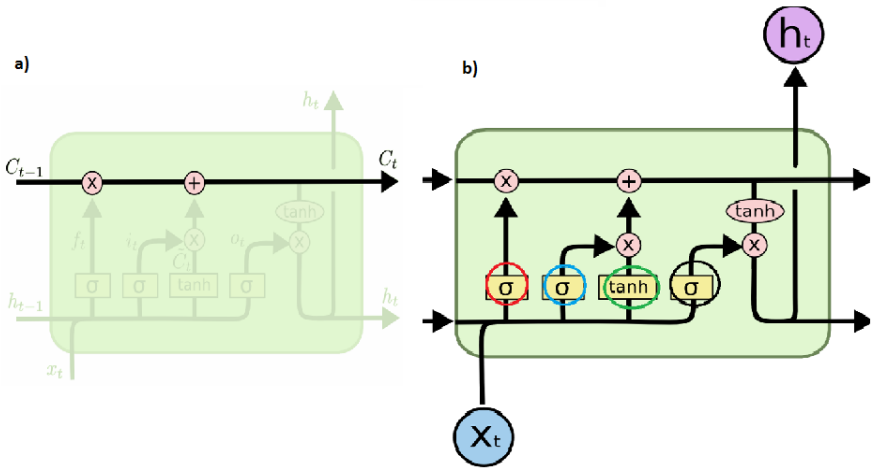
Rysunek 6) Schemat przedstawiający budowę komórki pamięci krótko-długo trwałej (LSTM). [Źródło: Internet]

Wszystkie informacje o stanie komórki LSTM przechowywane są na linii stanu komórki. Linia ta została wyróżniona ciemniejszym kolorem na rys.7a. LSTM posiada umiejętność modyfikowania informacji znajdujących się na tej linii, poprzez usuwanie części z nich oraz dodawanie nowych. Jest to charakterystyczna cecha komórek pamięci krótko-długo trwałej. W przypadku, gdy na wejściu sieci pojawiają się dane, uaktualniany jest zarówno stan komórki jak i wyjście.

Uaktualnienie stanu komórki składa się z dwóch etapów. Pierwszy etap polega na usuwaniu informacji z linii stanu komórki. Służy do tego bramka sigmoidalna nazywana bramką zapominającą, oznaczona na rys.7b czerwonym kółkiem. Decyduje ona o tym, jaka część informacji z linii stanu komórki zostanie zapomniana.

Drugi etap polega na dodawaniu nowych informacji do linie stanu komórki. Przebiega on w dwóch krokach. W pierwszym kroku podejmowana jest decyzja, jaka część stanu komórki będzie modyfikowana. Służy do tego bramka sigmoidalna nazywana bramką wejściową, oznaczona na rys.7b niebieskim kółkiem. W drugim kroku bramka wykonująca operacje nieliniową (tangensa hiperbolicznego) tworzy wektor wartości, którymi uaktualniana jest wybrana w pierwszym kroku część linii stanu komórki. Bramka ta została oznaczona na rys. 7b zielonym kółkiem.

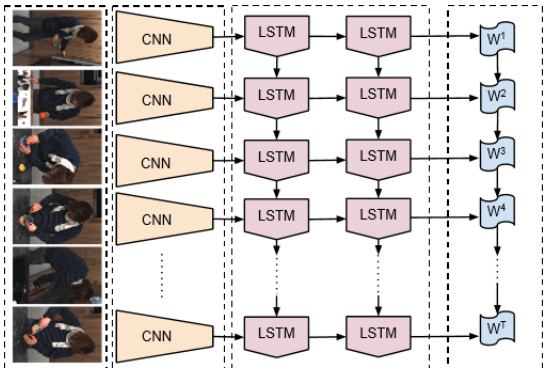
Gdy stan komórki zostanie już uaktualniony, wyznaczane jest jej wyjście. W pierwszym kroku ostatnia z bramek sigmoidalnych, oznaczona na rys. 7b czarnym kółkiem, decyduje jaka część informacji znajdujących się na linii stanu komórki podana zostanie na wyjście. W drugim kroku wartość stanu komórki poddawana jest operacji tangensa hiperbolicznego, a wynik zostaje przeskalowany przez wartość otrzymaną z bramki sigmoidalnej w pierwszym kroku i kierowany jest na wyjście komórki LSTM.



Rysunek 7a) Wyróżniona linia to tak zwana linia stanu komórki. Trzyma ona informacje o bieżącym stanie komórki LSTM. [Źródło: Internet] b) Schemat przedstawiający bramki wchodzące w skład komórki LSTM: bramkę zapominającą (oznaczoną czerwonym kółkiem), bramkę wejściową (oznaczoną niebieskim kółkiem), bramkę transformacji nieliniowej (oznaczoną zielonym kółkiem) oraz ostatnią z bramek (oznaczoną czarnym kółkiem). [Źródło: Internet].

4 Konwolucyjne sieci rekurencyjne z pamięcią długotrwałą

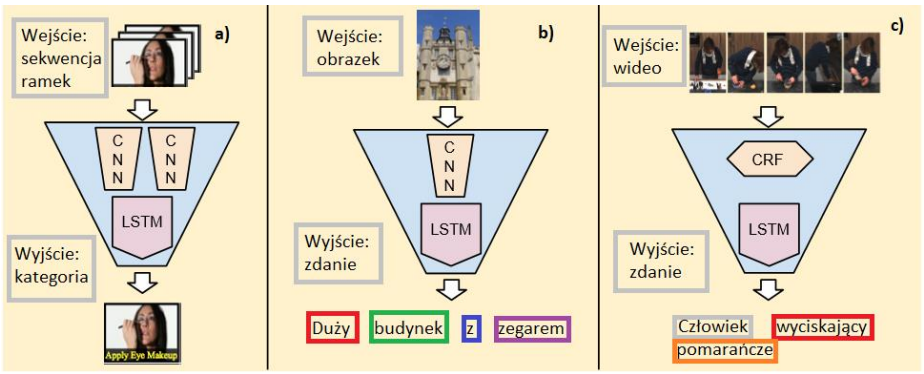
Konwolucyjne sieci rekurencyjne z pamięcią długotrwałą (LRCN) to architektura powstała w wyniku połączenia konwolucyjnych sieci neuronowych z komórkami pamięci krótko-długo trwałej. Ich schemat przedstawiony jest na rys.8.
Na wejście sieci podawane są kolejne obrazy. Konwolucyjna część sieci zajmuje się wydobywaniem cech wizualnych z obrazów, które następnie trafiają jako sekwencja do rekurencyjnej części sieci, czyli komórek z pamięcią krótko-długo trwałą.



Rysunek 8) Schemat architektury LRCN będącej połączeniem konwolucyjnych sieci neuronowych z komórkami pamięci krótko-długo trwałej. [Źródło: publikacja Long-term Recurrent Convolutional Networks for Visual Recognition and Description]

Analizując działanie tego typu sieci można zauważyć, że architektura ta może być wykorzystywana w trzech przypadkach:

- Wejściem jest sekwencja a wyjściem wartość o stałej wielkości. Przykładem może być rozpoznawanie aktywności na filmikach. Na wejście podawana jest sekwencja klatek, natomiast wyjściem jest wartość określająca czynność np. skakanie lub pływanie. Zostało to przedstawione na rys.9a.
- Wejściem jest wartość o stałej wielkości a wyjściem sekwencja. Przykładem może być generowanie opisów do obrazków. Na wejście podawany jest obraz, natomiast wyjściem jest sekwencja słów tworząca opis danego obrazka. Zostało to przedstawione na rys.9b
- Zarówno wejście jak i wyjście jest sekwencją. Przykładem może być generowanie opisów do filmików. Na wejście podawana jest sekwencja klatek z filmiku, natomiast wyjściem jest sekwencja słów tworząca opis danego filmiku. Zostało to przedstawione na rys.9c.



Rysunek 9) Graficzne przedstawienie różnych przypadków wykorzystywania architektury LRCN. a) Rozpoznawanie aktywności. b) Generowanie opisów do obrazów. c) Generowanie opisów do filmików. [Źródło: publikacja Long-term Recurrent Convolutional Networks for Visual Recognition and Description]

5 Przewidywanie popularności filmików – eksperymenty

5.1 Zbieranie danych i ich normalizacja

Dane wykorzystywane do uczenia sieci neuronowej pobrane zostały z trzech serwisów internetowych: Facebook'a, Youtube'a oraz Dailymotion, za pomocą API udostępnionego przez każdy z serwisów. Kryterium oceny popularności filmiku była liczba znormalizowanych punktów obliczana według wzoru:

$$PKT = \log2((\text{liczba wyświetleń} + 1) / \text{liczba osób śledzących kanał})$$

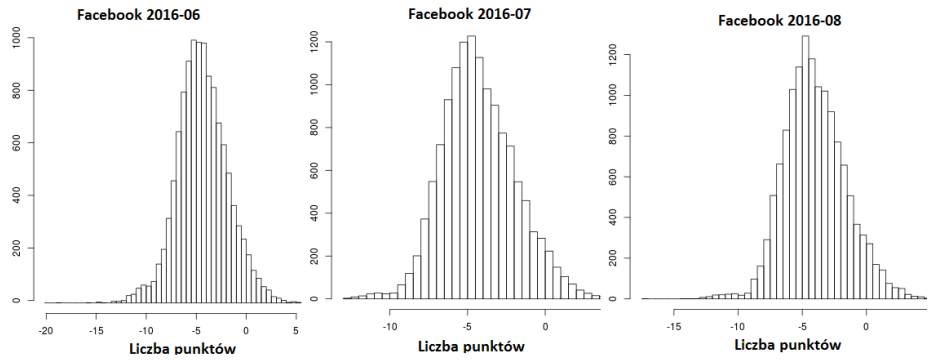
Czym większa liczba punktów, tym popularniejszy jest filmik. Jak widać jest ona proporcjonalna do liczby wyświetleń filmiku i odwrotnie proporcjonalna do liczby osób śledzących kanał na którym opublikowany został filmik.

Ponieważ liczba osób subskrybujących różne kanały nie jest stała w czasie, to pobieranie danych o filmikach, które zostały opublikowane długi czas temu, nie jest wskazane. Liczba przyznanych im punktów może być błędna, wskutek nagłego spadku lub też wzrostu liczby fanów kanału.

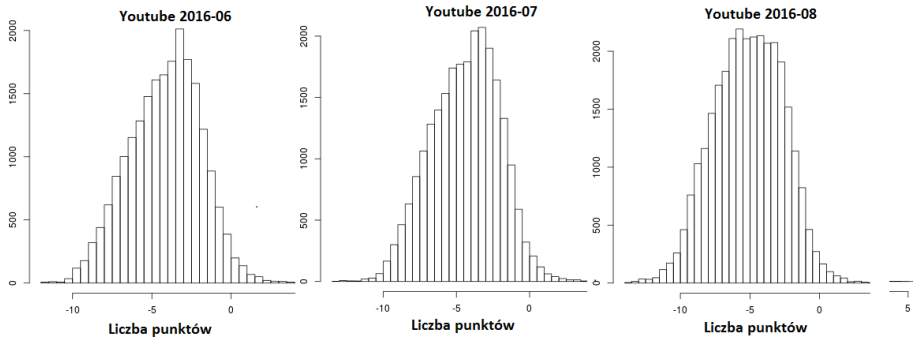
Z tego względu pobrane zostały informacje o filmikach, które znajdowały się w Internecie nie dłużej niż trzy miesiące (06.2016, 07.2016, 08.2016). Dodatkowo na filmiki z Youtube'a oraz Dailymotion zostało nałożone dodatkowe kryterium minimalnej liczby wyświetleń, które w obu przypadkach wynosiło 300.
Ostateczne wielkości zbiorów danych prezentują się w następujący sposób:

- Facebook: 39724 filmików
- Youtube: 86721filmików
- Dailymotion: 55706 filmików

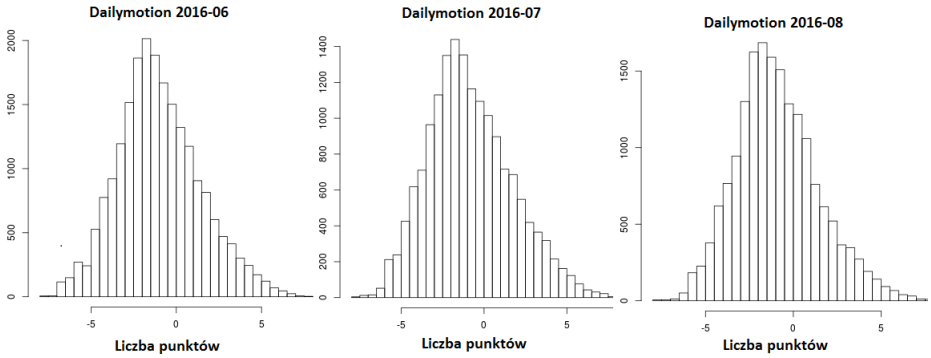
Rys.10,11,12 przedstawiają histogramy liczby znormalizowanych punktów. Histogramy te mają rozkład gaussowski, co pozwala sądzić, że wybrane kryterium oceny popularności jest prawidłowe.



Rysunek 10) Histogramy liczby punktów dla filmików z Facebook'a.



Rysunek 11) Histogramy liczby punktów dla filmików z Youtube'a.



Rysunek 12) Histogramy liczby punktów dla filmików z Dailymotion.

Ze względu na małą liczbą zebranych danych przeprowadzane zostało douczanie sieci, które polega na wykorzystaniu wytrenowanej już sieci i dostosowaniu wartości jej wag do nowych danych, na których ma działać.

5.2 Uczenie na podstawie miniaturek

W eksperymencie tym skorzystano z modelu wytrenowanej wcześniej sieci o nazwie Bvlc Reference Caffenet. Eksperyment obejmował uczenie sieci na podstawie miniaturkach filmików z Facebook'a, Dailymotion oraz Youtube'a.
Rezultaty prezentują się w następujący sposób:

- Dla filmików z Facebook'a otrzymana dokładność wynosiła 0.60312 w przypadku klasyfikacji binarnej (na filmiki popularne i niepopularne) oraz 0.2064 w przypadku klasyfikacji z podziałem na 8 kategorii popularności.
- Dla filmików z Dailymotion otrzymana dokładność wynosiła 0.5867 w przypadku klasyfikacji binarnej (na filmiki popularne i niepopularne).

- Dla filmików z Youtube'a otrzymana dokładność wynosiła 0.59123 w przypadku klasyfikacji binarnej (na filmiki popularne i niepopularne).

5.3 Uczenie na podstawie sekwencji klatek

Eksperyment ten składał się z dwóch etapów. W pierwszym etapie przeprowadzono douczanie modelu hybrydowego podając na jego wejście pierwsze klatki filmików, przy czym traktowane były one jako oddzielna dane wejściowe (nie tworzyły sekwencji). Tak wytrenowany model został w drugim etapie uzupełniony o warstwę LSTM (tworząc LRCN) i ponownie przetrenowany na tych samych danych wejściowych (z tym że traktowanych już jako sekwencje a nie niezależne obrazy).
Eksperyment ten został przeprowadzony tylko dla filmików z Facebook'a. Jego rezultaty są następujące:

- Dokładność sieci po przeprowadzeniu pierwszego etapu wynosiła 0.59254. Po etapie drugim wynosiła ona 0.599792.

6 Podsumowanie

W artykule przedstawiono podstawowe informacje na temat sztucznych sieci neuronowych, działania i architektury konwolucyjnych oraz rekurencyjnych sieci neuronowych, jak również architektury konwolucyjnych sieci rekurencyjnych z pamięcią długotrwałą. Zaproponowany został także sposób wykorzystania tych sieci do przewidywania popularności filmików oraz przedstawione zostały rezultaty niektórych badań. W artykule pokazano, jak można łączyć różne typy podstawowych sieci neuronowych (w tym przypadku konwolucyjnych sieci neuronowych z komórkami pamięci krótko-długo trwałej) aby dostosować architekturę sieci do konkretnego zadania, jakim jest przewidywanie popularności filmików.

7 Literatura

1. Jeffrey Donahue, Lisa Hendricks, Serio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, Trevor Darrell: Long-term Recurrent Convolutional Networks for Visual Recognition and Description
2. Andrej Karpathy blog: Convolutional Neural Networks for Visual Recognition
3. Andrej Karpathy blog: What a Deep Neural Network thinks about your #selfie?
4. Lars Eidnes blog: Auto – Generating Clickbait With Recurrent Neural Networks
5. Christopher Olah blog: Understanding LSTM Networks