# Pytania z wykładu

## Kiedy chcemy skupić się na wyciąganiu wzorców lepiej ejst zastosować AveragePooling czy MaxPooling?

Lepiej jest stosować MaxPooling, gdyż wyciągamy informację o wyróżniającym wzorcu który występuje w danym fragmencie obrazu.

## Po co kilka konwolucji pod rząd

Dwie konwolucje 3x3 działają podobnie jak jedna 5x5, ale przy dwóch konwolucjach mamy do zapamiętania mniej informacji, a oprócz tego mamy możliwość wciśnięcia dodatkowej funkcji aktywacji

## Po co stosować konwolucję 1x1

Pozwala tanim kosztem manipulować liczbą kanałów - np ze 100 zrobić 1. Wyciąga kluczowe cechy z wielu warstw.

## Jak liczyć forward pass?

Przeprowadzać normalne obliczenia na macierzach, w zależności od operacji. Na przykład na konwolucji wykonywać mnożenia.

## Jak liczyć backpropagation?

Poprzez pochodne cząstkowe.

## Czemu nie ma konwolucji parzystych 2x2, 4x4, 6x6

Nie robi się konwolucji parzystej, bo kwadraty wtedy nie mają środka. Samą operacje teoretycznie dałoby się wykonać, ale byłby problem z określeniem gdzie wpisać wynik konwolucji - kombinowanie.

## Jakie jest zastosowanie konwolucji 1x1

- Zmiana liczby kanałów: Konwolucja 1x1 może zmieniać liczbę kanałów w mapie cech. Na przykład, jeśli wejściowa mapa cech ma 256 kanałów, a chcemy zredukować je do 64, używamy konwolucji 1x1 z 64 filtrami. To pozwala na zmniejszenie złożoności modelu i jego wymagań obliczeniowych.
- Zwiększenie nieliniowości: Pomimo swojej prostoty, konwolucje 1x1 wprowadzają dodatkową nieliniowość do modelu, gdy są stosowane z funkcjami aktywacji, takimi jak ReLU. To pomaga w wyłapywaniu bardziej skomplikowanych wzorców w danych.
- Efektywna integracja informacji z różnych kanałów: Konwolucja 1x1 efektywnie łączy informacje z różnych kanałów mapy cech. Dzięki temu model może lepiej uczyć się zależności między kanałami.

## Po co stosować podwójną konwolucję

- Lepsze Wyłapywanie Cech: Stosowanie dwóch konwolucji po sobie pozwala modelowi na wyłapywanie bardziej złożonych cech w danych. Pierwsza warstwa może wyłapywać proste wzorce, takie jak krawędzie czy kolory, a druga warstwa może integrować te proste wzorce w bardziej skomplikowane struktury.

- Zwiększenie Głębokości Sieci bez Nadmiernego Zwiększania Złożoności: Przez stosowanie wielu lżejszych warstw konwolucyjnych zamiast jednej ciężkiej, można zwiększyć głębokość sieci bez znacznego wzrostu liczby parametrów i obciążenia obliczeniowego.
- Nieliniowość i Abstrakcja: Między warstwami konwolucyjnymi często stosuje się funkcje aktywacji (np. ReLU). Dzięki temu system może wprowadzać nieliniowość do procesu uczenia, co jest kluczowe dla efektywnego modelowania złożonych relacji w danych.
- Zastosowania w Sieciach Segregacyjnych: W sieciach służących do segmentacji obrazów, takich jak U-Net, podwójne konwolucje są wykorzystywane do stopniowego zmniejszania wymiarów przestrzennych, jednocześnie zwiększając liczbę cech. To pozwala na efektywną analizę i klasyfikację różnych regionów obrazu.
- Ograniczenie Przesadnego Dopasowania: Stosując dwie lub więcej warstw konwolucyjnych zamiast jednej dużej, można potencjalnie zmniejszyć ryzyko przeuczenia się (overfitting) sieci do danych treningowych, co jest szczególnie ważne w zadaniach z ograniczoną ilością danych.

## Zastosowanie funkcji flatten i jaki jest Jego zamiennik

**Flatten:**

- Przekształcanie Map Cech na Wektor: flatten służy do przekształcenia wielowymiarowych map cech (wynikających z warstw konwolucyjnych i poolingowych) na jednowymiarowy wektor. Jest to kluczowe, ponieważ warstwy w pełni połączone (fully connected layers), które zwykle następują po warstwach konwolucyjnych, wymagają danych wejściowych w formie jednowymiarowej.
- Przygotowanie do Klasyfikacji: Po przetworzeniu obrazu przez warstwy konwolucyjne i poolingowe, flatten umożliwia przekształcenie wynikającej z tego mapy cech na format, który może być użyty do klasyfikacji (np. rozpoznawanie obiektów na obrazie).

**GlobalAveragePooling**

- stosowane jako alternatywa do warstwy Flatten
- Redukcja Parametrów: GAP znacznie redukuje liczbę parametrów, przekształcając każdą mapę cech w pojedynczą liczbę przez obliczenie średniej wartości wszystkich jej elementów. Dzięki temu zmniejsza ryzyko przeuczenia (overfitting).
- Utrzymanie Przestrzennych Informacji: W przeciwieństwie do flatten, GAP zachowuje przestrzenną informację o cechach, co jest ważne w niektórych aplikacjach, takich jak lokalizacja obiektów na obrazie.
- Elastyczność Rozmiaru Wejściowego: GAP umożliwia sieci radzenie sobie z obrazami o różnych rozmiarach, ponieważ niezależnie od rozmiaru wejściowej mapy cech, wyjście GAP jest stałe (jedna wartość na kanał).

## Auxiliary classifier

- pomocniczy klasyfikator który ma na celu poprawić proces uczenia i generalizacji modelu.
- są trenowane na tych samych danych co główna siec ale skupiają się na cechach z warstw pośrednich.
- Funkcja straty jest wyliczana jako suma głównego klasyfikatora i klasyfikatorów pośrednich.

- Zastosowane w sieciach Inception. (Google)

### Plusy resnet'a i depthneta

**ResNet**

- Rozwiązanie Problemu Zanikającego Gradientu: ResNet wprowadza połączenia rezydualne, które umożliwiają propagację gradientu wstecz przez wiele warstw, pomagając w treningu bardzo głębokich sieci.
- Możliwość Budowy Głębszych Sieci: Dzięki połączeniom rezydualnym, ResNet umożliwia skuteczne trenowanie sieci o znacznie większej głębokości niż to było możliwe wcześniej (np. ResNet-152).
- Wysoka Skuteczność w Zadaniach Klasyfikacyjnych: Sieci ResNet osiągają bardzo dobre wyniki w zadaniach klasyfikacji i rozpoznawania obrazów.
- Lepsza Generalizacja: Połączenia rezydualne pomagają w generalizacji modelu, co jest korzystne przy przenoszeniu nauczonych cech na nowe, niewidziane dane.

**DenseNet**

- Efektywne Wykorzystanie Parametrów: W DenseNet każda warstwa otrzymuje jako wejście dane z wszystkich poprzednich warstw, co prowadzi do bardziej efektywnego wykorzystania parametrów i zmniejsza ryzyko przeuczenia.
- Ulepszona Propagacja Cech i Gradientów: Dzięki gęstym połączeniom, cechy z wcześniejszych warstw są bezpośrednio przekazywane do późniejszych warstw, co poprawia przepływ informacji i gradientów w sieci.
- Zmniejszenie Zapotrzebowania na Liczbę Kanałów: Gęste połączenia pozwalają na redukcję liczby kanałów w każdej warstwie konwolucyjnej, ponieważ sieć może korzystać z cech zgromadzonych z poprzednich warstw.
- Dobry w Wykrywaniu Detali: Dzięki zachowaniu wszystkich cech z poprzednich warstw, DenseNet jest szczególnie dobry w wykrywaniu drobnych detali w obrazach.

# Pytania wymyślone

# Opisz co to jest operacja Pooling. Opisz co to jest konwolucja ze stride=2. Do czego służą takie operacje?

### Pooling

Operacja redukcji wymiaru poprzez przeprowadzenie agregacji na sąsiadujących elementach, dzięki czemu zmniejsza się liczba parametrów w sieci neuronowej. Wykorzystuje się dwa typy operacji:

- maxpooling - wybiera maksymalną wartość z każdego okna danych
- averagepooling - oblicza średnią wartość każdego okna.

### Konwolucja ze stride = x

Jest to dodanie przesunięcia się kernela. Domyślnie przesuwamy się o co jedną kolumnę jak ustawimy Stride = 2 to będziemy się przemieszczać co dwie kolumny.

### Zastosowanie

Zarówno operację pooling jak i stride używa się do redukcji liczby parametrów i rozmiaru danych stosowanych do nauki sieci neuronowej.

## Co to jest i do czego służy regularyzacja w kontekście sieci neuronowych. Wymień trzy metody regularyzacji.

Regularyzacja są to metody mające na celu pomóc sieci nauczyć się wzorców występujących w danych uczących a nie samych danych.

Metody regularyzacji

- L1
- L2
- Dropout

## Opisz krótko "Multi Head Attenction" z Transformer-ów.

Technika pozwalająca modelowi równocześnie skupić się na różnych aspektach informacji wejściowej, co zwiększa jego zdolność do rozumienia złożonych zależności w danych.

Działanie:

- **Podział na Głowice**: wejściowe dane są najpierw dzielone na kilka "głowic". Każda z nich przetwarza dane niezależnie, co pozwala modelowi równocześnie zwracać uwagę na różne aspekty informacji.
- **Obliczenie Uwagi**: Dla każdej głowicy wykonywana jest operacja uwagi (attention), która polega na obliczeniu wag między każdym słowem w sekwencji a każdym innym słowem. Wagi te są wykorzystywane do stworzenia ważonej sumy reprezentacji słów, co pozwala modelowi zrozumieć, jakie słowa są ważne w danym kontekście.
- **Zastosowanie Wzorów Skalowanej Uwagi**: W każdej głowicy używany jest mechanizm tzw. "scaled dot-product attention", który pomaga w stabilizacji treningu modelu.
- **Konsolidacja Wyników**: Wyniki z wszystkich głowic są następnie łączone i przekazywane do kolejnych warstw modelu.

## Co to jest strata (loss) i po co ją obserwować w procesie uczenia?

### Strata

Strata to kara za złe przewidywanie. Oznacza to, że strata jest liczbą wskazującą, jak zła była prognoza modelu na pojedynczym przykładzie. Jeśli przewidywanie modelu jest doskonałe, strata wynosi zero; w przeciwnym razie strata jest większa.

### Przyczyny obserwacji

- Dzięki zmianom straty możemy zauważyć czy nasza sieć neuronowa się uczy.
- Możemy zobaczyć czy nasz model się nie przeucza poprzez sprawdzenie czy na danych treningowych funkcja straty maleje a na walidacyjnych rośnie.

## Opisz krótko za co odpowiedzialna jest sieć neuronowa w deep Q-learning.

W tradycyjnym Q-learning, tabela Q (tablica wartości Q dla każdego stanu i akcji) jest używana do przechowywania i aktualizacji wartości Q, które pomagają agentowi decydować, jaką akcję podjąć w danym stanie.

Sieć neuronowa zatem jest używana zamiast tabeli Q, która uczy się na bazie otrzymywanych wydarzeń stanów Q i na ich podstawie potem agent podejmuje decyzje od tej wartości Q, jaką przewidziała sieć neuronowa.

## Po co stosuje się "Depthwise Separable Convolution". Za co odpowiedzialna jest konwolucja "Depthwise", a za co "Pointwise".

Depthwise Separable convolution jest to przeprowadzanie konwolucji na każdym pojedynczym kanale koloru w obrazie a następnie połączenie wyników konwolucją 1x1. Zalety:

- zmniejszona liczba operacji. Mniej operacji mnożenia, gdyż iterujemy po każdym kanale osobno
- zmniejszona liczba parametrów, co zmniejsza ryzyko przeuczenia się sieci.

Operacja DepthWise jest to właśnie przeprowadzenie konwolucji na każdym oddzielnym kanale w celu wyodrębnienia cech dla konkretnego kanału, a konwolucja Pointwise jest tą konwolucją 1x1 łączącą cechy z każdego kanału.

## Wymień kolejne kroki, które modyfikują "pamięć długotrwałą"(cell state) w LSTM.

1. **Zapomnienie Informacji (Forget Gate)**: model podejmuje decyzję, które informacje należy odrzucić z pamięci długotrwałej. LSTM używa tzw. **forget gate**, który korzysta z aktualnego wejścia i poprzedniego stanu ukrytego, aby wygenerować wartość między 0 a 1 dla każdego numeru w pamięci długotrwałej. Wartość 1 oznacza "zachowaj tę informację", a wartość 0 oznacza "zapomnij o tej informacji".
2. **Aktualizacja Pamięci (Input Gate)**: LSTM decyduje, jakie nowe informacje będą dodane do pamięci długotrwałej. Składa się to z dwóch części:
    - Input gate: tworzy wektor wartości kandydujących, które mogą zostać dodane do stanu komórki.
    - Kombinacja z aktualnym wejściem i poprzednim stanem ukrytym, w celu zdecydowania, które wartości z kandydujących faktycznie aktualizują stan komórki.
3. **Aktualizacja Stanu Komórki (Cell State)**: Stary stan komórki jest mnożony przez wartości z forget gate (co może spowodować zapomnienie niektórych starych informacji), a następnie dodawane są nowe kandydujące wartości (filtrowane przez input gate). Efektem jest nowy stan komórki, który zawiera zarówno zachowane stare informacje, jak i nowe informacje.
4. **Generowanie Wyjścia (Output Gate)**: LSTM używa swojego stanu komórki do wygenerowania wyjścia. Output gate decyduje, jaką część stanu komórki należy wyjść. Stan komórki przechodzi przez funkcję aktywacji (zazwyczaj **tanh**), a jego wynik jest mnożony przez wynik output gate, dając ostateczne wyjście LSTM.

## Jak Progressive GAN umożliwia generację obrazów w dużej rozdzielczości?

Progressive GAN jest uczony najpierw generować obrazy o mniejszej rozdzielczości. Gdy zostaną osiągnięte zadowalające wyniki są, następnie dokładane kolejne warstwy do dyskryminatora i generatora aby je uczyć generować obrazy o podwyższonej rozdzielczości. Proces można by było uogólnić do następujących kroków:

- Uczymy GAN'a na obrazkach małej rozdzielczości. Chcemy by się nauczył generować podstawowe wzorce i cechy.
- Następnie gdy osiągniemy stabilność w generacji i dyskryminacji, są dokładane kolejne warstwy do generatora i dyskryminatora aby uczył się dodawać więcej szczegółów.
- Kroki te powtarzamy aż nie uzyskamy porządanej rozdzielczości.

## Opisz krótko "residual/skip connection" z modelu ResNet.

Są to połączenia które przekazują dane z jednej warstwy sieci neuronowej do n'tej warstwy, pomijając przy tym kilka warstw pośrednich sieci. Zostały zastosowane aby rozwiązać problem degradacji w głębokich sieciach neuronowych a konkretnie problem z liczeniem gradientu w propagacji wstecznej który zanikał przy zbyt głębokich sieciach.

## Jakiej techniki się używa, żeby usunąć liczne (z reguły więcej niż dwie) warstwy "Linear"/"Dense"/"Fully connected" po "Flatten" w CNN? Opisz jak działa ta technika. Dlaczego stosowanie tych wielu warstw sprawiało problemy?

Używa się techniki Global Average Pooling (GAP). Łączy wszystkie mapy cech w jedną, poprzez zastosowanie operacji average pooling na mapach cech dopóki wymiary przestrzenne nie zredukują się do wymiaru 1. np tensor (8, 10, 64) zostałby zredukowany do (1, 1, 64) Wcześniejsze zastosowania miały następujące problemy:

- spłaszczanie map cech po funkcji flatten doprowadzało do utraty informacji przestrzennej
- więcej warstw wymaga więcej danych uczących, które nie zawsze są dostępne
- dłuższy czas uczenia sieci neuronowej oraz większe ryzyko przeuczenia sieci przez dużą liczbę parametrów.

## Jak rozwiążesz poniższy problem.

Opisz w kilku zdaniach wymyślone rozwiązanie będąc przy tym możliwie precyzyjnym (np. podając nazwy konkretnych metod, modeli). Proszę odpowiedzieć na podstawie obecnie posiadanej wiedzy, tego jak się Państwu wydaje, że należałoby to zrobić.

1. Wykrycie czy na filmie (około 30s) występuje konkretna osoba, rozpoznając ją po ubraniu. Na filmie może pojawiać się wiele osób jednocześnie - np. nagranie z wejścia do budynku PWr, C-1.Jako dane wejściowe mają Państwo 5 zdjęć poszukiwanej osoby z różnych kadrów/perspektyw (np. zdjęcie z przodu, z tyłu, itd.).
2. Model do klasyfikacji ras królików na podstawie zdjęcia. Niestety, dostępna jest baza danych z małą liczbą przykładowych zdjęć. Jak poradzisz sobie z takim problemem?
3. Aplikacja, która musi być uruchomiana na telefonie komórkowym służąca do zliczania ludzi na zdjęciach.
4. Klasyczny model do klasyfikacji dokumentów tekstowych. Opisz jedno z podejść (1) oparte na sieciach rekurencyjnych(uczonych od zera) lub (2) oparte na Transformer-ach (transfer learning).

1.

Najpierw należałoby wyciągnąć konkretne obrazy z nagrania. Zakładając na przykład wyciąganie co sekundę 1 klatkę z nagrania wykorzystując do tego narzędzie. Następnie do każdej klatki można by było zastosować model który wykrywałby osoby na obrazie na przykład YOLO i zwracałby wykryte osoby. Kolejnym krokiem

powinno być przygotowanie danych do przetworzenia przez sieć neuronową do rozpoznania osoby po ubraniu. Każde ze zdjęć należałoby przepuścić przez sieć głęboką celem wyciągnięcia cech ubrania. Następnie należałoby porównać cechy ubrań do wykrycia danej osoby z cechami wykrytymi w klatce filmu. Można by było zastosowac miare taką jak odległość cosinusowa, bądź mahalanobisa. Jeżeli by dla danego progu dane były wystarczająco blisko, mozna by było zwracać informację że dana osoba na nagraniu jest.

2.

Najpierw należałoby spróbować wygenerować dodatkowe zdjęcia królika, zacząłbym od prostych transformacji na obrazach np przesunięcia, obrotu, zmiany kolorystyki. Podeszedłbym do problemu iteratywnie, dla danej techniki generacji dodatkowych danych bym testował jak uczona architektura się zachowuje. Jeżeli te techniki nie byłyby skuteczne skorzystałbym z transfer learning'u czyli dostosowałbym nauczoną już sieć do nowego problemu na przykład ResNet'a wyuczonego na ImageNet'cie. Dostosować musiałbym jedynie klasyfikator a nastepnie "douczyć" model aby dobrze się dostosował do nowego problemu. Zastanowiłbym się nad zastosowaniem jeszcze technik regularyzacji, gdyż mając mały zbiór danych model mógłby się nauczyć obrazków a nie cech na nich występujących.

3.

Jeżeli chciałbym wydać aplikację zarówno na IOs'a i Androdida to zastosowałbym Flutter'a do napisania samej aplikacji mobilnej. Zezwala on na napisanie szybko aplikacji, która będzie działała na obydwu platformach. Nastepnie wykorzystałbym do samego rozpoznawania osób na model YOLO-tiny który jest dostosowany do działania na telefonach komórkowych. Jeżeli moje rozwiązanie miałoby się stosować do tworzenia globalnych statystyk, np mamy kilka osób które zlicza ile osób średnio do galerii dodałbym przekazywanie informacji z klasyfikatora do jakiegoś serwera co jakiś czas albo zapisywać je w pamięci telefonu, które później byłyby przechowywane na serwerze.

4.2

WYbranie modelu obejmowało by prawdopodobnie transformera z rodziny BERT, który już został wytrenowany na dużym zbiorze tekstowym. Następnie przygotowałbym teksty, poprzez lematyzację usunięcie stopwords, zastosowanie tokenizacji. Przeprowadziłbym następnie dostosowanie modelu, modyfikacje warstw odpowiedzialnych za klasyfikacje aby był w stanie określić czy dokument to artykuł czy np pozew sądowy. Odpalę następnie trenowanie danych. Trzeba by było jeszcze pomyśleć o sposobie ewaluacji. Zastosowałbym pewnie miarę F1-Score, accuracy oraz wykres straty, tak aby wiedzieć jak bardzo transformer się wyucza i czy na przykład nie potrzebne będą dodatkowe kroki w obróbce danych bądź dodatkowe zmiany w sieci, takie jak na przykład zastosowanie DropOut'u bądź na przykład Cross Validation.

# Opisz dwa dowolne modele, służące do klasyfikacji obrazów które zostały zaproponowane 2017r lub później. Opisz tylko najważniejsze kwestie, które zostały wprowadzone