

Konwolucyjna sieć neuronowa, w skrócie CNN, to wyspecjalizowany typ modelu sieci neuronowej przeznaczony do pracy z dwuwymiarowymi danymi obrazu.

Nazwa „konwolucyjna/splotowa sieć neuronowa” wskazuje, że sieć wykorzystuje operację matematyczną zwaną splotem. Sieci splotowe to wyspecjalizowany typ sieci neuronowych, które używają splotu zamiast ogólnego mnożenia macierzy w co najmniej jednej z warstw sieci. Sieci te najczęściej wykorzystywane są do przetwarzania obrazu.

Sieci konwolucyjne mogą uczyć się przestrzennej hierarchii wzorców

Oznacza to, że pierwsze warstwy sieci uczą się takich małych lokalnych wzorców jak np krawędzie i kolejne warstwy uczą się coraz to wyższych/większych struktur składających się z elementów rozpoznawanych przez warstwy. Czyli w tych ostatnich warstwach mogą być np. takie cechy (przy rozpoznawaniu samochodu na zdjęciu tak koło pojazdu, zderzak, maska pojazdu i na wyjściu dostajemy predykcje, jaki jest to element.

Warstwa Splotowa

Warstwy splotowe są głównymi elementami budulcowymi używanymi w konwolucyjnych sieciach neuronowych.

Splot to proste zastosowanie filtru na wejściu, które powoduje aktywację. Wielokrotne zastosowanie tego samego filtru do wejścia skutkuje mapą aktywacji zwaną mapą cech, wskazującą lokalizację i siłę wykrytej cechy na wejściu, takiej jak obraz.

Mamy tutaj także przedstawioną prostą implementację sieci konwolucyjnej

```
model.add(Conv2D(32, (3,3), input_shape=(8, 8, 1)))
```

Pierwszy z parametrów, określany jako **filters** to głębina wyjściowej mapy cech, czyli liczba filtrów, obliczana przez konwolucję. Jest to liczba filtrów, które chcemy, aby nasza sieć się nauczyła i za ich pomocą ekstraktowała cechy z obrazka i tworzyła mapy cech, które przekazywane są do kolejnej warstwy.

Drugi z parametrów to **kernel_size**, jest to rozmiar łąt wyciągniętych z danych wejściowych. Najprościej mówiąc jest to łąta, która przechodzi po obrazie w celu wyciągnięcia pewnych cech. Najczęściej stosowane łąty to (3,3) lub (5,5).

Kolejny parametr **input_shape** określa parametry obrazu wejściowego. W tym przypadku te dwie pierwsze 8 określają rozmiar obrazka, a ta 1 określa, że obrazek jest czarno biały. Gdyby na końcu była 3 oznaczałoby to, że obrazek jest kolorowy. Parametr ten jest stosowany tylko w pierwszej warstwie.

W warstwie tej następuje operacja konwolucji, które polega na przesuwaniu okna o wymiarach 3x3 lub 5x5 po trójwymiarowej mapie cech, której działanie zostało przedstawione na animacji. Okno jest zatrzymywane w każdym miejscu, w którym da się je umieścić i dokonywana jest ekstrakcja trójwymiarowej łąty otaczającej je cech. Taka łąta ma kształt (wysokość_okna, szerokość_okna, głębina_wejscia)

Operacja Max-Pooling:

Operacja Max-Pooling - to prosta operacja arytmetyczna, której zadaniem jest skalowanie obrazu przeprowadzane przy użyciu okien ekstrakcji przetwarzających mapy cech.

Okna te zwracają maksymalne wartości każdego z kanałów.

Warstwy tego typu umożliwiają zmniejszenie liczby wymiarów danych, co jest niezbędne w celu uzyskania rozsądnych rozmiarów map cech przy wzroście liczby cech.

Rozwiązanie to pozwala również kolejnym warstwom konwolucji „widzieć” większy przestrzenny obszar danych wejściowych.

```
model.add(layers.MaxPooling2D(pool_size=(2,2)))
```

Operacja Max-Pooling różni się od warstwy konwolucji tym, że operacja max-pooling jest przeprowadzana zwykle przy użyciu okien o wymiarach 2x2 przy kroku równym 2 (ma to na celu dwukrotne zredukowanie mapy cech). Konwolucje są natomiast przeprowadzane przy użyciu okien o wymiarach 3x3.

Warstwa Flatten:

zadaniem tej warstwy jest spłaszczenie tensora trójwymiarowego do takiej postaci, aby warstwa Dense nie miała problemu z jego przetworzeniem.

```
model.add(layers.Flatten())
```

Dlaczego Dane się spłaszczają?

Dane są spłaszczane, w celu przekazania ich do warstwy Dense, która będzie za chwilę omówiona. Warstwa ta może przyjąć tylko tensor jednowymiarowy, a więc zdjęcie musi zostać przekształcone w taki sposób, aby sieć je „przeczytała”.

Różnica między warstwą spłotową, a warstwą gęstą:

Różnica między sieciami konwolucyjnymi, a sieciami gęstymi jest taka, że warstwy gęste (Dense) uczą się cech/parametrów globalnie na podstawie danych wejściowych, natomiast warstwy konwolucyjne uczą się lokalnych wzorców.

Co to oznacza?

Jeżeli podamy w zbiorze treningowym dane, gdzie uszkodzona lampa była tylko po lewej stronie. Natomiast, gdy w zbiorze testowym podamy obrazek z uszkodzoną prawą lampą, to warstwa Dense nie rozpozna tego uszkodzenia, ponieważ w zbiorze treningowym, nie został umieszczony obrazek z takim uszkodzeniem.

W sieciach konwolucyjnych zaletą jest fakt, że nawet jeśli uszkodzona lampa pojawiła się po lewej stronie, a testując sieć podajemy zdjęcie, gdzie uszkodzona lampa znajduje się po prawo to sieć wykorzystująca warstwy konwolucyjne nie będzie miała problemu z rozpoznaniem tej cechy.

Wniosek:

Sieci konwolucyjne są niewrażliwe na przesunięcia.

