**Evrişimli Sinir Ağlarında Havuzlama (Pooling) Teknikleri: Maxpooling, Minpooling ve Average Pooling**

Evrişimli sinir ağları (CNN'ler), görüntü işleme ve benzeri görevlerde yaygın olarak kullanılan derin öğrenme modelleridir. CNN'lerin temel bileşenlerinden biri olan havuzlama (pooling) teknikleri, öznitelik haritasının(Featuremap) boyutunu azaltarak varyans ve hesaplama maliyetini düşürmek ve genel model performansını artırmak için kullanılır. Bu yazıda, maxpooling, minpooling ve average pooling gibi popüler havuzlama tekniklerini inceleyeceğiz.

Üç tür havuzlama işlemi vardır:

1. Maksimum havuzlama(Max pooling): Yığının(batch) maksimum piksel değeri seçilir.
2. Min havuzlama(Min pooling) : Yığının minimum piksel değeri seçilir.
3. Ortalama havuzlama(Average pooling) : Yığındaki tüm piksellerin ortalama değeri seçilir.

Burada yığın, görüntünün boyutuna göre karar verilen filtre boyutuna eşit boyutta bir piksel grubu anlamına gelir. Aşağıdaki örnekte 9x9 boyutunda bir filtre seçilmiştir. Havuzlama yönteminin çıktısı filtre boyutunun değişen değerine göre değişir.

**1. Maxpooling**

Maxpooling, bir bölgenin (genellikle 2x2 veya 3x3'lük bir pencere) en büyük özelliğini alarak öznitelik haritasını küçültür. Örneğin, 2x2'lik bir pencere için, her dört piksel arasından en büyük olanı seçilir ve yeni bir öznitelik haritası oluşturulur. Bu işlem, özniteliklerin öne çıkan özelliklerini korurken boyutunu azaltır. Maxpooling'in matematiksel formülasyonu şu şekildedir:

Maxpooling(𝑥)=max⁡(𝑥𝑖,𝑗)Maxpooling(*x*)=max(*xi*,*j*​)

Maxpooling, özellikle nesne tanıma gibi görevlerde kullanışlıdır çünkü belirgin özelliklerin korunmasına yardımcı olur.

**2. Minpooling**

Minpooling, maxpooling'in tam tersidir. Her bir bölge için en küçük özelliği alarak öznitelik haritasını küçültür. Ancak genellikle nadiren kullanılır çünkü çoğu durumda en belirgin özelliklerin en büyük değerler olduğu kabul edilir.

Minpooling'in matematiksel formülasyonu şu şekildedir:

Minpooling(𝑥)=min⁡(𝑥𝑖,𝑗)Minpooling(*x*)=min(*xi*,*j*​)

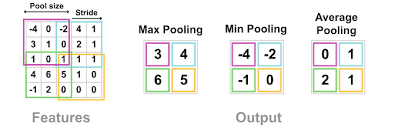
**3. Average Pooling**

Average pooling, her bir bölgenin ortalamasını alarak öznitelik haritasını küçültür. Bu yöntem, özellikle gürültüyü azaltmak ve daha düzenli bir öznitelik haritası elde etmek için kullanışlıdır. Average pooling'in matematiksel formülasyonu şu şekildedir:

Average Pooling(𝑥)=1𝑛∑𝑖,𝑗𝑥𝑖,𝑗Average Pooling(*x*)=*n*1​∑*i*,*j*​*xi*,*j*​

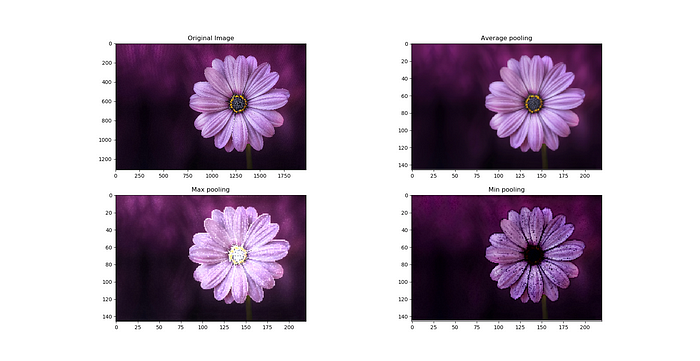
Burada 𝑛*n*, bölge boyutunun (örneğin, 2x2 için 4) birbirine bölünen piksel sayısını temsil eder.

Bu üç tekniğin yapılış şekli aşağıdaki resimde gösterilmiştir:

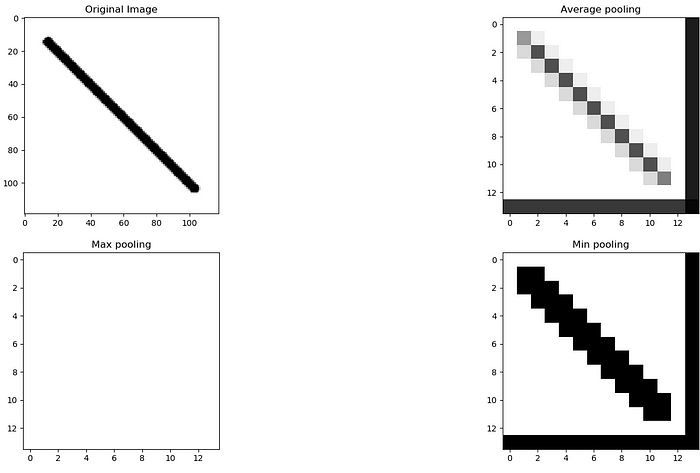


Belirli bir havuzlama yönteminin diğerlerine kıyasla daha iyi olduğunu söyleyemeyiz. Her birinin kendine özgü avantajları ve dezavantajları vardır. Genellikle, maxpooling en yaygın olarak kullanılanıdır çünkü en belirgin özellikleri korurken hesaplama maliyetini azaltır. Avarage pooling yöntemi ise görüntüyü yumuşatır ve dolayısıyla bu havuzlama yöntemi kullanıldığında keskin özellikler tespit edilemeyebilir. Ancak, belirli bir görev için en uygun havuzlama tekniğini belirlemek için eldeki verilere göre deneysel olarak test edilmelidir.

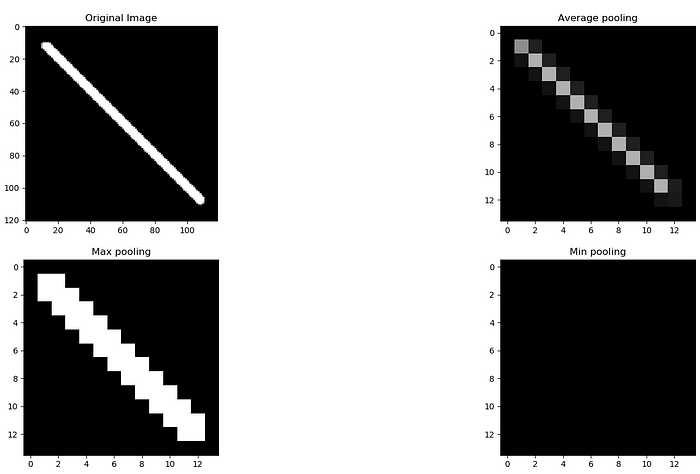
İşlemlerin orijinal resim ile karşılaştırması aşağıdaki şekillerde gösterilmektedir.



Maksimum havuzlama görüntüden daha parlak pikselleri seçer. Görüntünün arka planı koyu olduğunda ve görüntünün yalnızca daha açık pikselleriyle ilgilendiğimizde kullanışlıdır. Örneğin: MNIST veri kümesinde rakamlar beyaz renkle temsil edilir ve arka plan siyahtır. Bu yüzden maksimum havuzlama kullanılır. Benzer şekilde, tam tersi durumda da min havuzlama kullanılır.



Min havuzlama, beyaz arka planlı ve siyah nesneli görüntüler için daha iyi sonuç verir.



Maksimum havuzlama, siyah arka planlı ve beyaz nesneli görüntüler için daha iyi sonuç verir (Örn: MNIST veri seti).

MNIST rakam veri kümesini CNN kullanarak sınıflandırırken, maksimum havuzlama kullanılır çünkü bu görüntülerdeki arka plan hesaplama maliyetini azaltmak için siyah yapılır.

Aşağıdaki python kodu, bir girdi görüntüsü üzerinde her üç havuzlama türünü de gerçekleştirecek ve sonuçları gösterecektir.

|  |  |
| --- | --- |
|  | from skimage.measure import block\_reduce |
|  | import numpy as np |
|  | import matplotlib.pyplot as plt |
|  | from matplotlib.image import imread |
|  |  |
|  | img = imread('black\_stick.png') |
|  | mean\_pool=block\_reduce(img, block\_size=(9,9,1), func=np.mean) |
|  | max\_pool=block\_reduce(img, block\_size=(9,9,1), func=np.max) |
|  | min\_pool=block\_reduce(img, block\_size=(9,9,1), func=np.min) |
|  |  |
|  | plt.figure(1) |
|  | plt.subplot(221) |
|  | imgplot = plt.imshow(img) |
|  | plt.title('Original Image') |
|  |  |
|  | plt.subplot(222) |
|  | imgplot3 = plt.imshow(mean\_pool) |
|  | plt.title('Average pooling') |
|  |  |
|  | plt.subplot(223) |
|  | imgplot1 = plt.imshow(max\_pool) |
|  | plt.title('Max pooling') |
|  |  |
|  | plt.subplot(224) |
|  | imgplot1 = plt.imshow(min\_pool) |
|  | plt.title('Min pooling') |
|  |  |
|  | plt.show() |

Bu yazıda, maxpooling, minpooling ve average pooling gibi popüler havuzlama tekniklerini inceledik ve her birinin işlevselliği hakkında bilgi verdik.