1) Görüntü İşleme (Basic)

Dijital görüntü nedir? (Sayısal Görüntü)

Gerçek yeryüzüne ait renkli bir resmin sayısal bilgisayar verisine dönüştürülmüş halidir.

Söz konusu görüntüde detaylar yan yana sıralanmış çok küçük kare şeklindeki elemanların birleşiminden oluşmuştur ve aslında sürekli değil; kesiklidir.

↳ Bu çok küçük karelere **piksel** adı verilir.

Sayısal Görüntü (Dijital), analog (gerçek resim) görüntünün M sütun ve N satırdan oluşacak şekilde örneklenmesi sonucu oluşur.

↳ Satır ve Sütunun kesiştiği her Bölgeye **Piksel** denir.

* Sonuç olarak, sayısal görüntüye çevrilen resimde N x M adet piksel bulunur.

Çözünürlük nedir?

Görüntünün kaç piksele bölündüğünü, yani kaç piksel ile temsil edildiğini gösterir.

↳ (128 x 128) = 128 piksel x ekren, 128 piksel y ekse

* Analog görüntünün bilgisayar ortamında işlenebilmesi için sayısallaştırılması gerekir.

Renkli Görüntü

Renkli görüntüler; R(Kırmızı), G(Yeşil), B(Mavi) ile kodlanmıştır. Renkli görüntüyü oluşturan bu 3 renk Bant olarak isimlendirilir.

↳

Renkli görüntülerin her bir pikseli bilgisayar ekranlarında 24 bitlik veri olarak görüntülenir. Şöyle, her bir renk 8 bit ile (2^8) kodlanacağından 3 renk (RGB) 3x8=24 bit ile kodlanır. Bu durumda RGB görüntülerin her bir pikseli $2^8 \cdot 2^8 \cdot 2^8 \approx 17$ milyon farklı renge sahip olabilir. Bu 3 rengin birleşiminin değer aralığı RGB = (0,0,0), ---, (255,255,255) biçiminde gösterilir.

Matris anlamında düşünüldüğünde doğal renkli 2 boyutlu bir RGB görüntü, her biri (N x M) büyüklüğünde (piksele sahip) 3 matrisin bileşiminden oluşur.

I(i, j, 1); i = 1, 2, ..., N , j = 1, 2, ..., M → Kırmızı banda ilişkin matris.

I(i, j, 2); i = 1, 2, ..., N , j = 1, 2, ..., M → Yeşil banda ilişkin matris

I(i, j, 3); i = 1, 2, ..., N , j = 1, 2, ..., M → Mavi banda ilişkin matris

Sayısal görüntünün kapladığı Bellek alanı?

N satır ve M sütundan oluşan, her bir pikseli "m" bit ile temsil edilen (N x M) büyüklüklü (pikselli) sayısal bir görüntüyü depolamak için gerekli bit sayısı:

(Bit sayısı = N x M x m)

Not:

256 renk (8 bit) = 2^8 — 16 renk (4 bit)

128 renk (7 bit) — 8 renk (3 bit) = 2^3

64 renk (6 bit) — 4 renk (2 bit)

32 renk (5 bit) — 2 renk (1 bit)

Analog görüntü (Sürekli görüntü), (Orijinal)

Dijital // (Sayısal görüntü)

* Sayısal görüntüyü oluşturan piksellerin sayısı (N x M) ne kadar fazla olursa orijinal (Analog) görüntüye o kadar fazla yaklaşır.

(Piksel Sayısı ↓, Orijinallik ↓)

(Bit sayısı ↓, Kalite ↓) } Daha az renk tonu olur çünkü.

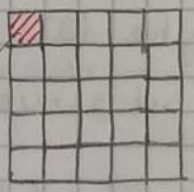
11) Convolutional Neural Networks

Vertical, horizontal, 10°, 128° ... edge detection ile high-level feature'lar extract edilir.

Padding: 2 problemi çözmek için kullanılır.

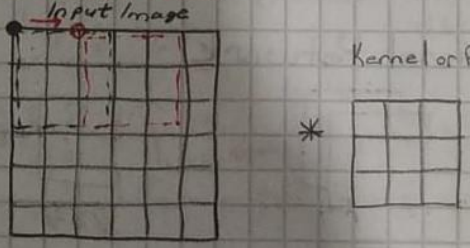
1) We dont want the image to shrink on every step because if we have maybe a hundred layer of deep net, after a hundred layers we end up with a very small image.

2) Throwing away (ignoring) a lot of information from the edges of the image.



→ "Padding" ile image'ye additional border'lar eklenir. (Padded done with zeros)

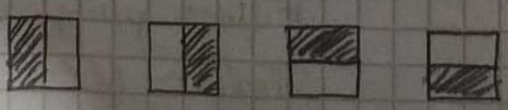
Strided Convolution:



Stride 2 olsun örneğin; bu durumda jump ederken bir yana veya aşağı gitmek yerine 2 gider.

↳ filter ⇒ look for a certain type of edge in original photo.

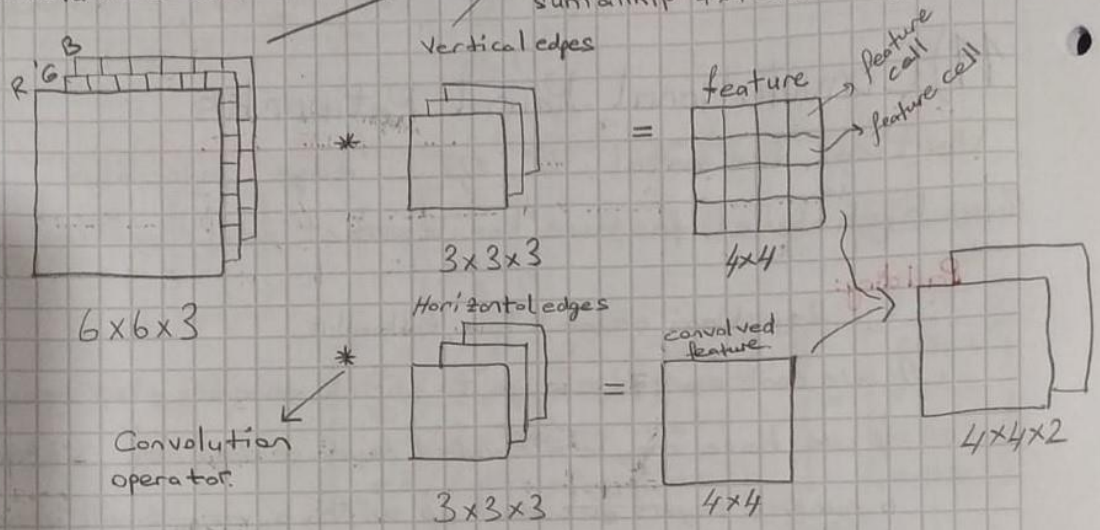
ÖR



} filters olabilir.

Convolutions over Volume:

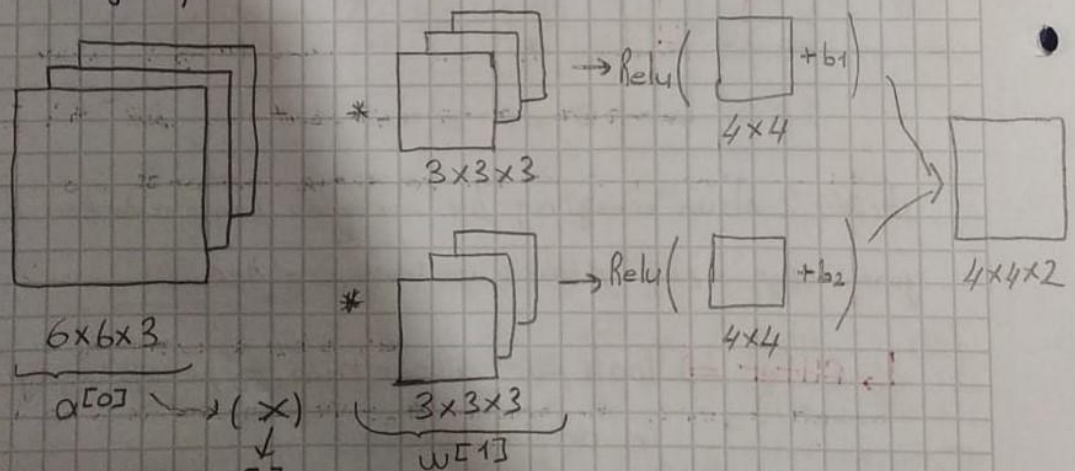
(3×3×3 'ün her bir depth'i sırası ile R,G,B 'de kullanılır; sonrasında sumalınıp 4×4 elde edilir)



→ We now detect 2 features (Horizontal and Vertical edges) but maybe 10°, 130° or several hundreds of different features we can extract.

★ The objective of the "Convolution Operation" is to extract the high-level features such as edges from the input image.

One Layer of a Convolutional Network

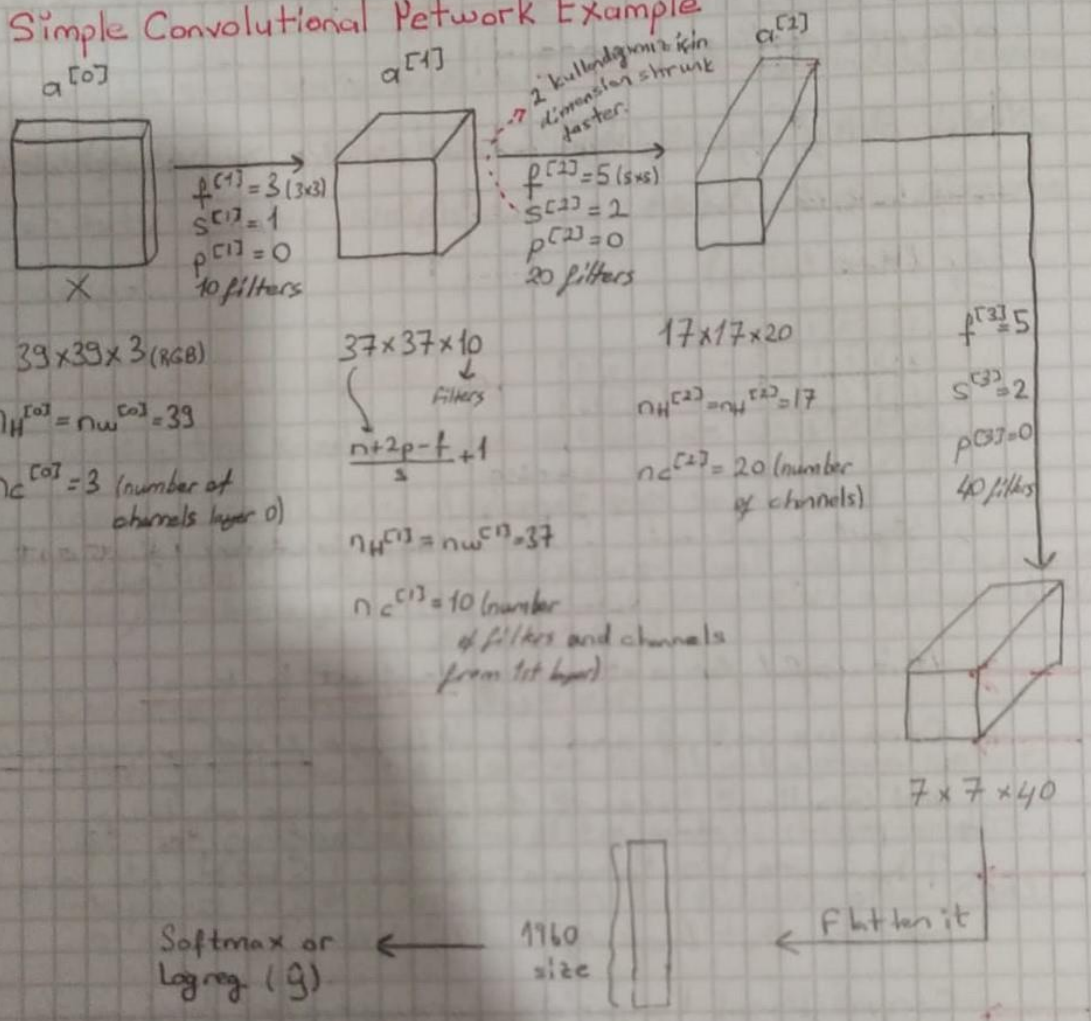


$$Z^{[1]} = W^{[1]} a^{[0]} + b^{[1]}$$

$$a^{[1]} = g(Z^{[1]})$$

→ filters play a role similar W.

Simple Convolutional Network Example



$n_H^{[0]} = n_W^{[0]} = 39$

$n_C^{[0]} = 3$ (number of channels layer 0)

$\frac{n+2p-f}{s}+1$

$n_H^{[1]} = n_W^{[1]} = 37$

$n_C^{[1]} = 10$ (number of filters and channels from 1st layer)

$n_H^{[2]} = n_W^{[2]} = 17$

$n_C^{[2]} = 20$ (number of channels)

Neyaptık? Taking $39 \times 39 \times 3$ input image, then Compute $7 \times 7 \times 40$ features for this image.

Note: Number of channels generally increase (3, 10, 20, 40)

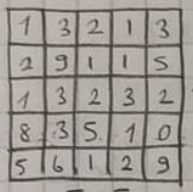
- Generally first Convlayer is responsible for capturing the Low-Level features such as edges, color, etc. With added layers, the architecture adopts to the High-level features as well.

Pooling Layers:

Other than Convolutional Layers, ConvNets often also use pooling layers.

"Max Pooling" ; pooling type'larından biridir ve literatürde en çok kullanılandır.

ÖR (Max Pooling)



5×5 input.

filter size

f = 3 (3×3)

s = 1 ↳ stride

} Hyperparameters. (Just set them)

→ It is useful for extracting dominant features.

→ Once we set f and s; it is just fixed computation and Gradient Descent does not change anything.

→ There are No Parameters to LEARN. No parameters that backprop will adapt.

→ Convolutional Layer'da parametreleri backprop ile kendisi öğreniyordu. (Weight parametrelerini)

→ It reduces the size of the representation to speed the computation

→ # Parameters = 0

Fully Connected Layers:

Birkaç kez evrişimli katmandan ve pooling katmanından geçen ve matris halinde olan görselimiz düz bir vektör haline getirilir.

Girdi resminizi sinir ağları ile eğitebileceğimiz kıvama getirdikten sonra geriye sadece klasik sinir ağlarındaki çalışma mantığı kalıyor. Yine katmanlardaki node'larda özellikler tutuluyor ve weight, bias değiştirilerek öğrenme başlıyor.

Convolutional Layers:

Serving the same purpose of feature extraction.

Resmin özelliklerini algılamaktan sorumlu.

Görüntüdeki yüksek ve düşük seviyeli özellikleri çıkarmak için filtreler uygular. (Bu filtre örneğin kenarları algılayacak bir filtre olabilir)

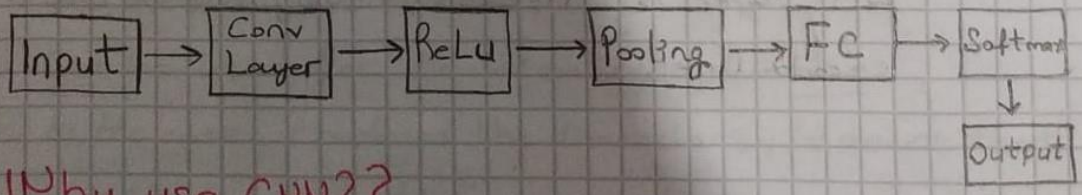
Convolved feature'lar oluşturur.

Filtre görsel üzerinde gezer ve özellik yakalamaya çalışır.

Öğrenilen parametreler bu filtrelerdeki değerlerdir.

Ana amacı negatif değerlerden kurtulmak olan ReLu CNN'de çok önemlidir.

★ Flattened hale getirilinceye kadar olan kısım tamamen complex görsellerde accuracy artırmak için. Sonrasında normal bildiğimiz multi-level perceptron kurulur.



Why use CNN??

Input 28×28 olsun → Bu problemi MLP ile çözebiliriz, ancak Input 1000×1000 iken MLP tek başına işe yaramaz. Bu yüzden CNN, extract the feature of image and Convert it into Lower Dimension without loosing its Characteristics.

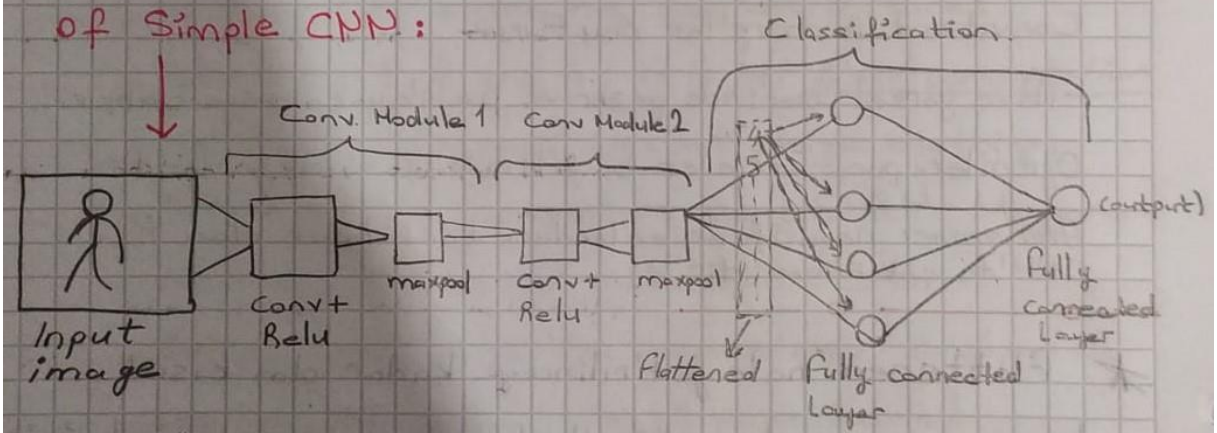
Örneğin: Initial image size $224 \times 224 \times 3$ olsun.
Convolution olmadan çözmeye çalışır isem 100352 nörona input layer da ihtiyacım olur. Ancak convolution uyguladıktan sonra input tensor boyutu $1 \times 1 \times 1000$'e düşürülebilir. Bu ne demek? First input layer'da feed forward için 1000 nörona ihtiyacım var demek.

Two main advantages of Convolutional layers over just using fully connected layers.

1) Parameter sharing
2) Sparsity of connections

Andrew'e göre.

End-to-end Structure of Simple CNN:



→ "Fully connected" means that every node in the first layer is connected to every node in the second layer.

ResNets Nedir? (Residual Network)

Aşırı derin network'lerde gradient'in 0'a yakınsaması problemini ortadan kaldırma amaçlı geliştirilmiştir. Mantık oldukça kolaydır. Birkaç layer önde yer alan aktivasyon fonksiyonunu, şu anki aktivasyon fonksiyonunun inputu ile toplayıp aktive eder. Böylece şu anki layer'ın lineer transform sonucu 0 olsa bile değer üretir. Bu da identity function öğrenmek için mükemmel bir detaydır. Sonuç olarak; "Vanishing Gradients" problemi olmadan çok daha derin ağlara sahip olabiliriz.

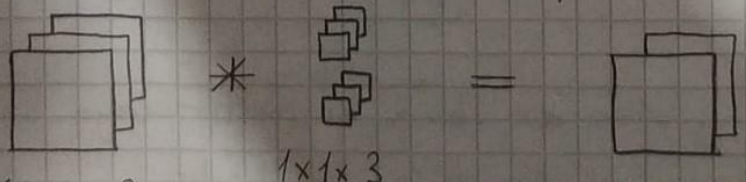
2 önemli problemi çözer: a) Vanishing / Exploding Gradients (Eğitim sırasında bazı nöronlar ölür, bu durum "backprop" sırasında güçlü aktivasyonları olmayan nöronların değerlerinin erimesi ve ağın eğitimine katkı verememesi olarak özetlenir.

b) Optimizasyonun zorlaşması (parametre sayısı artışı eğitimi zorlaştırır; Resnet'de daha fazla katman olmasına rağmen, daha az parametre vardır.)

What does 1x1 convolution do?

a) Can be used for dimensionality reduction.

b) In the case, it reduces the dimensions; it means that that it reduces the computations.



6 x 6 x 3
* Depth: 3

1 x 1 x 3
* Depth: 3
filters: 2

6 x 6 x 2
* Depth: 2

Inception Module Nedir? (GoogLeNet)

Inception module, allow more efficient computation through dimensionality reduction with 1x1 convolutions.

Ne yaptığı çok basit

↳ Take multiple filter sizes within CNN and sıralı olarak çalıştırmak yerine aynı seviyede çalışmalarını emretmek.

* it allows to use multiple types of filter size, instead of being restricted to a single filter size.

* Tek bir type filter size seçmek yerine, multiple seçim yapıp, outputları concatenate eder.

Neural Network with Many Convolutional Layers

Input → Conv+ReLu → Pooling → Conv+ReLu → Pooling → Flatten → FC+Softmax

Feature Learning | Classification

Non Linearity (ReLu)

ReLu(| 15 | 20 | -10 | 35 | / | 18 | -2 | 25 | 100 | / | 20 | -1 | 25 | -10 | / | 101 | 75 | 18 | 23 |) → ReLu → | 15 | 20 | 10 | 33 | / | 18 | 2 | 25 | 100 | / | 20 | 1 | 25 | 10 | / | 101 | 75 | 18 | 23 |

4x4

We want our ConvNet to learn non-negative values.

Transfer Learning

↳ * Computer Vision'da çokça kullanım alanı bulur.

* Training size az iken kullanılır veya hesaplama gücün az ise kullanılır.