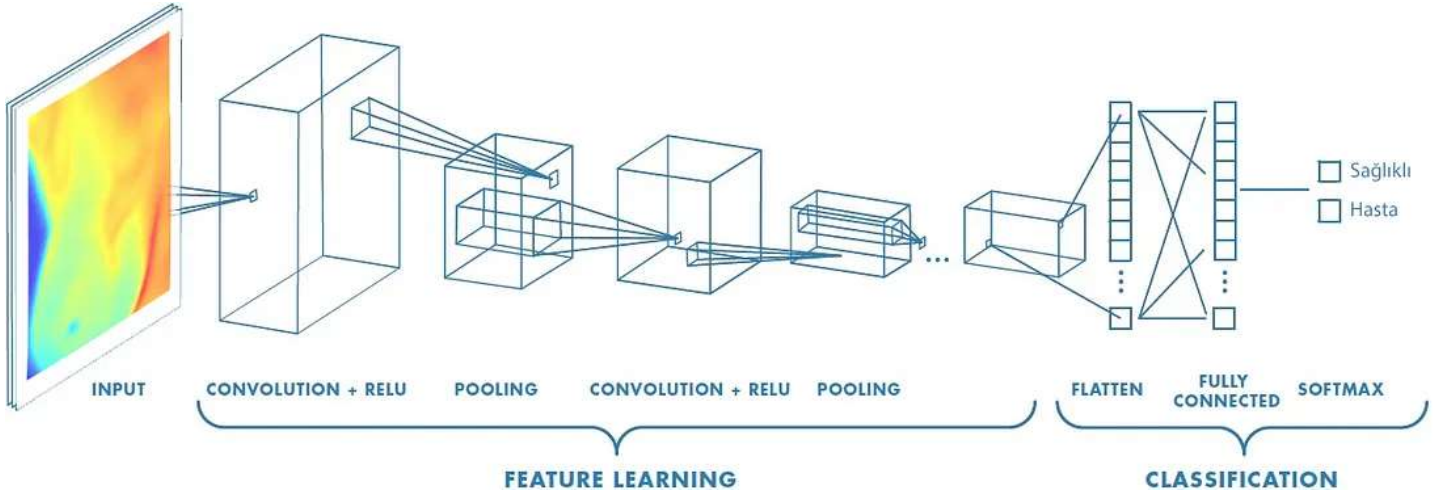


★ Get unlimited access to the best of Medium for less than \$1/week. [Become a member](#)



Convolutional Neural Network (ConvNet yada CNN) nedir, nasıl çalışır?



Tuncer Ergin · [Follow](#)

6 min read · Oct 2, 2018



Share



More

Facebook'a bir resim yüklediğinizde size "X kişisini etiketlemek istermisiniz?" şeklinde bir önermede bulunur. Peki o kişiyi nasıl bildiğini merak ettiniz mi? Veya Google'ın resim arama algoritması nasıl çalışır hiç düşündünüz mü? İşte tüm bunların arkasında bir sinir ağı vardır. Daha net olmak gerekirse, Convolutional Neural Networkten (ConvNet yada CNN) söz ediyoruz. (Türkçeye Evrişimsel Sinir Ağları olarak çevriliyor). CNN biyoloji ve bilgisayar bilimlerinin garip bir karışımı gibi görünse de, bu, resim tanıma için kullanılan çok etkili bir mekanizmadır.



Google Vision Api

Fikrin Arkasındaki Biyoloji

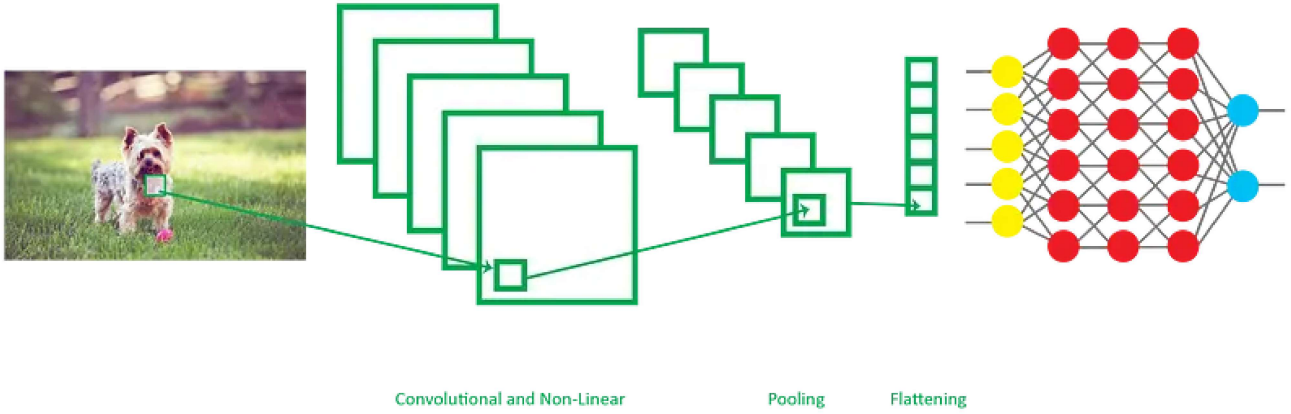
Cnn verilen görüntüleri ayırt etmek için bir uçağı uçak ya da yılanı yılan haline getiren benzersiz özellikleri kullanırlar. Aslında bu süreç beynimizde de bilinçsizce oluyor.

Örneğin, bir uçak resmine baktığımızda, iki kanat, motor, pencere gibi özellikleri birbirinden ayırarak uçağı tanımlayabiliriz. Cnn de aynı şeyi yapar, ancak daha önce eğriler ve kenarlar gibi alt düzey özellikleri tespit ederler ve daha soyut kavramlara kadar bunları oluştururlar.

Evrişimsel Sinir Ağlarının Yapısı

Konuştuğumuz işlevselliği elde etmek için, Cnn görüntüyü çeşitli katmanlarla işler. Bu katmanları yazının bir sonraki bölümünde ayrıntılı olarak inceleyeceğiz ancak şu an sadece bu katmanlara ve amaçlarına genel bir bakış yapalım:

- **Convolutional Layer** — Özellikleri saptamak için kullanılır
- **Non-Linearity Layer** — Sisteme doğrusal olmayanlığın (non-linearity) tanıtılması
- **Pooling (Downsampling) Layer** — Ağırlık sayısını azaltır ve uygunluğu kontrol eder
- **Flattening Layer** — Klasik Sinir Ağı için verileri hazırlar
- **Fully-Connected Layer** — Sınıflamada kullanılan Standart Sinir Ağı



ConvNet şeması

Temel olarak, Cnn, sınıflandırma sorununun çözümü için standart Sinir Ağı kullanır, ancak bilgileri belirlemek ve bazı özellikleri tespit etmek için diğer katmanları kullanır.

Haydi her katmanın ve işlevlerinin detaylarına dalalım.

Convolutional Layer

Bu katman CNN'nin ana yapı taşıdır. Resmin özelliklerini algılamaktan sorumludur. Bu katman, görüntüdeki düşük ve yüksek seviyeli özellikleri çıkarmak için resme bazı filtreler uygular. Örneğin, bu filtre kenarları algılayacak bir filtre olabilir. Bu filtreler genellikle çok boyutludur ve piksel değerleri içerirler.(5x5x3) 5 matrisin yükseklik ve genişliğini, 3 matrisin derinliğini temsil eder. Şimdi mu filtrenin nasıl uygulandığına bakalım;

1	1	1	0	0
0	1	1	1	0
0	0	1	1	1
0	0	1	1	0
0	1	1	0	0

Örnek resim

Örneğin basit olması için sadece 1 kanal işlenecektir.

Resimin 5×5 boyutunda ve 1 ve 0 'lardan oluşan bir resim olduğunu varsayalım.

Filtremizi 3×3 boyutunda oluşturalım.

1	0	1
0	1	0
1	0	1

Filtre

Şimdi, filtrenin nasıl uygulandığına bir bakalım,

1 _{x1}	1 _{x0}	1 _{x1}	0	0
0 _{x0}	1 _{x1}	1 _{x0}	1	0
0 _{x1}	0 _{x0}	1 _{x1}	1	1
0	0	1	1	0
0	1	1	0	0

Image

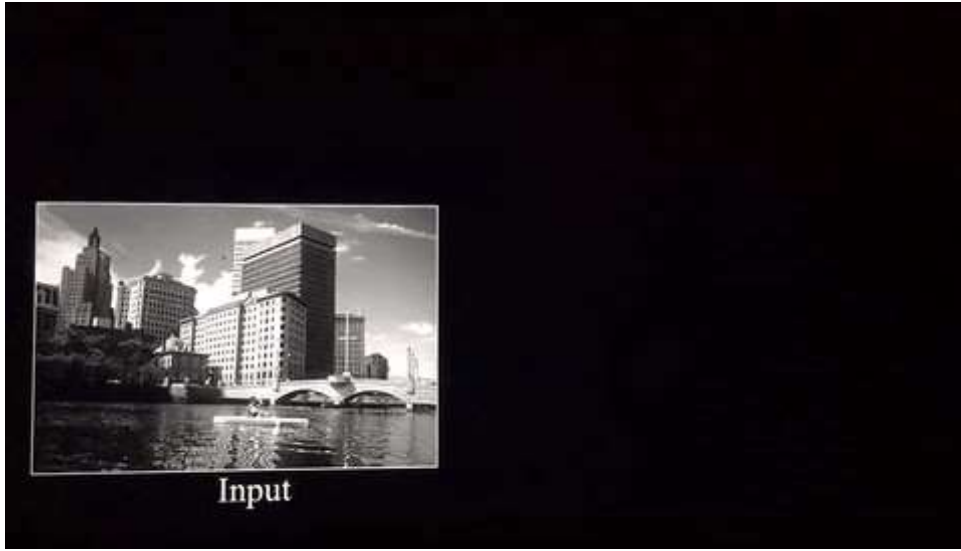
4		

Convolved
Feature

Filtrenin uygulanaşı

Öncelikle, filtre görüntünün sol üst köşesine konumlandırılır. Burada, iki matris arasında (*resim ve filtre*) indisler birbirisi ile çarpılır ve tüm sonuçlar toplanır, daha sonra sonucu çıktı matrisine depolanır. Ardından, bu filtreyi sağa 1 piksel (“basamak” olarak da bilinir) kadar hareket ettirip işlemi tekrarlanır. 1. Satır bittikten sonra 2 satıra geçilir ve işlemler tekrarlanır. Tüm işlemler bittikten sonra çıktı matrisi oluşturulur. Burada çıktı matrisinin 3×3 olmasının nedeni 5×5 matrisinde 3×3 filtresi yatayda ve dikeyde 3 kez hareket etmesinden kaynaklanır. Eğer resim 6×4 ve filtre 3×3 boyutunda olsaydı çıkış matrisi 4×2 boyutunda olurdu. Peki çıktı matrisi bize ne anlatıyor? Bu matrise genellikle Feature Map denir. Filtre tarafından temsil edilen özellikte görüntünün bulunduğu yeri gösterir. Kısacası, filtreyi görüntü üzerinden hareket ettirerek ve basit matris çarpımını kullanarak, özelliklerimizi tespit ediyoruz.

Genellikle, birden çok özelliği tespit etmek için birden fazla filtre kullanılır, yani bir Cnn ağında birden fazla konvolüsyonel (Convolutional) katman bulunur. Aşağıdaki animasyona bir göz atın, burada bu işlem biraz daha görsel olarak anlatılıyor:

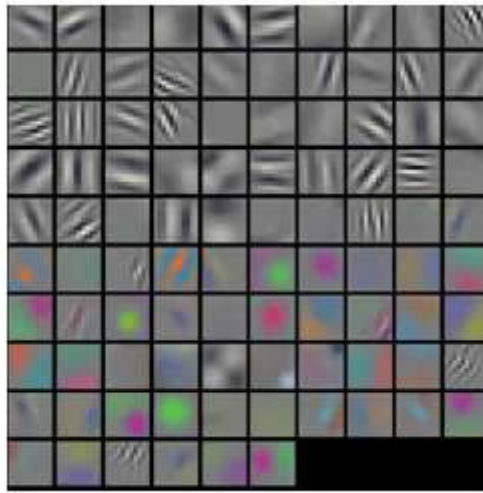


Convolution işlemi

Bir adım daha

İlk filtreyi uyguladığımızda, bir Feature Map oluşturuyor ve bir özellik türünü tespit ediyoruz. Ardından, ikinci bir filtre kullanıp başka bir özellik türünü algılayan ikinci bir Feature Map oluştururuz.

Yukarıdaki örnekte görebildiğimiz gibi bu filtreler basit olabilir, ancak görüntüde bazı karmaşık özellikler çıkarmak istiyorsanız bu filtreler karmaşık hale gelebilirler. Daha karmaşık olan filtreleri görmek için aşağıdaki resme gözatabilirsiniz.



Örnek filtreler

Daha önce bahsettiğimiz, ancak ayrıntılı olarak açıklamadığımız bir başka şey, stride (büyük adım).

Bu terim genellikle padding terimi ile birlikte kullanılır. Stride, filtrenin giriş görüntüsünün etrafında nasıl evrildiğini denetler. Yukarıdaki örnekte Stride 1 piksel idi, ancak daha büyük olabilir. Bu, Feature Map'ın çıktısının boyutunu etkiler. Cnn'nin ilk aşamalarında, ilk filtreleri uygularken, diğer Convolutional Katmanlar

için mümkün olduğunca çok bilgiyi korumamız gerekir. İşte padding bu nedenden dolayı kullanılır. Feature Map'ın orijinal giriş görüntüsünden daha küçük olduğunu fark etmişsinizdir. Bu nedenle Padding, (aşağıdaki resimde olduğu gibi)resmin boyutunu korumak için bu haritaya sıfır değerler katacaktır:

Padding

Non-linearity

Tüm Convolutional katmanlarından sonra genellikle Non-Linearity(doğrusal olmayan) katmanı gelir. Peki görüntüdeki doğrusallık neden bir problemdir? Sorun şu ki, tüm katmanlar doğrusal bir fonksiyon olabildiğinden dolayı Sinir Ağı tek bir perception gibi davranır, yani sonuç, çıktıların linear kombinasyonu olarak hesaplanabilir.

Bu katman aktivasyon katmanı (Activation Layer) olarak adlandırılır çünkü aktivasyon fonksiyonlarından birini kullanılır. Geçmişte, sigmoid ve tahn gibi doğrusal olmayan fonksiyonlar kullanıldı, ancak Sinir Ağı eğitiminin hızı konusunda en iyi sonucu Rectifier(ReLU) fonksiyonu verdiği için artık bu fonksiyon kullanılmaya başlanmıştır.

ReLU Fonksiyonu $f(x) = \max(0, x)$

ReLU fonksiyonunun Feature Map'a uygulandığında aşağıdaki gibi bir sonuç üretilir.

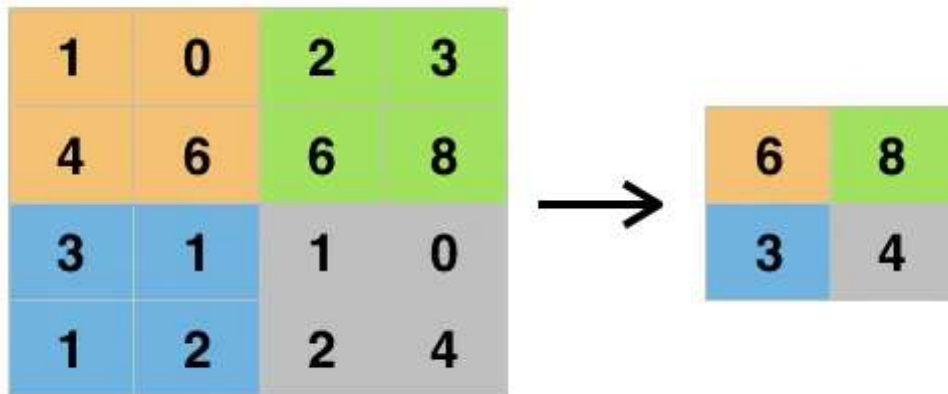


Feature Map'taki siyah değerler negatiftir. Relu fonksiyonu uygulandıktan sonra siyah değerler kaldırılır onun yerine 0 konur.

Pooling Layer

Bu katman, CovNet'teki ardışık convolutional katmanları arasına sıklıkla eklenen bir katmandır. Bu katmanın görevi, gösterimin kayma boyutunu ve ağ içindeki parametreleri ve hesaplama sayısını azaltmak içindir. Bu sayede ağdaki uyumsuzluk kontrol edilmiş olur. Birçok Pooling işlemleri vardır, fakat en popülerleri max pooling'dir. Yine aynı prensipte çalışan average pooling, ve L2-norm pooling algoritmaları da vardır.

Bu işlemi şekiller üzerinden açıklayarak gidelim. Öncelikle 2×2 boyutunda bir filtre oluşturalım. Bu filtreyi aşağıdaki (4×4) resim üzerinde görebilirsiniz. Resimde gördüğünüz gibi, filtre, kapsadığı alandaki en büyük sayıyı alır. Bu sayede, sınır ağının doğru karar vermesi için için yeterli bilgiyi içeren daha küçük çıktıları kullanmış olur.



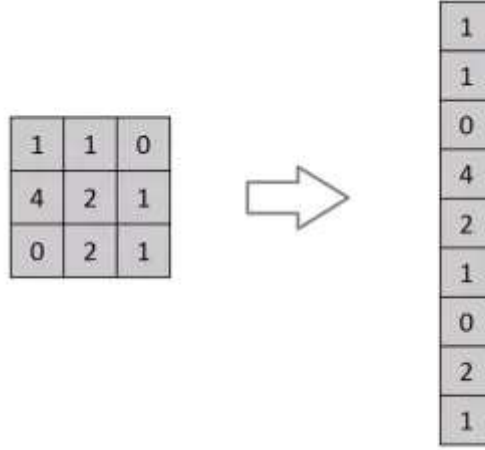
Maxpooling işlemi

Bununla birlikte birçok kişi bu katmanı kullanmayı tercih etmez. Bunun yerine Convolutional katmanında daha büyük Stride (Filtreyi kaydırma işlemi) tercih edilir.

Ayrıca variational autoencoders (VAEs) or generative adversarial networks (GANs) gibi daha üretken modellerde pooling katmanını tamen çıkartırlar.

Flattening Layer

Bu katmanın görevi basitçe, son ve en önemli katman olan Fully Connected Layer'ın girişindeki verileri hazırlamaktır. Genel olarak, sinir ağı, giriş verilerini tek boyutlu bir diziden alır. Bu sinir ağındaki veriler ise Convolutional ve Pooling katmanından gelen matrixlerin tek boyutlu diziye çevrilmiş halidir.



Flatting

Fully-Connected Layer

Bu katman ConvNet'in son ve en önemli katmanıdır. Verileri Flattening işleminden alır ve Sinir ağı yoluyla öğrenme işlemini gerçekleştirir.

Bu katman başlı başına bir derya olduğundan dolayı bu yazıda incelenmeyecektir.

[Open in app](#) ↗



Search



A

katmanı(ları) ve Flattening katmanı eklenmelidir. daha sonra ReLu katmanı kadar Fully-Conncted katmanı eklenir. Aklınızda bulunsun Cnn'nin en son katmanı Fully Connected katmanı olmalıdır. Örnek olarak;

Giriş Resmi -> $[[\text{Conv} \rightarrow \text{ReLU}]^N \rightarrow \text{Pool?}]^M \rightarrow \text{Flattening} \rightarrow [\text{FC} \rightarrow \text{ReLU}]^K \rightarrow \text{FC}$ şeklinde tanımlanabilir.

Bazı yaygın ConvNet isimleri;

LeNet — Bu ağ, Convolutional Networks'ün ilk başarılı uygulaması sayılır. 1990'lı yıllarda Yann LeCun tarafından geliştirildi ve posta kodlarını, basit basamakları vb. okumak için kullanıldı.

AlexNet — Bu ağ, 2012'de ImageNet ILSVRC challenge'ta sunuldu. Diğer ağlardan

oldukça başarılı bir performans göstermiştir.

GoogLeNet — ILSVRC 2014'ün kazananı bu ağ olmuştur. Ağdaki parametrelerin sayısını önemli ölçüde azaltmak için avarage pooling katmanlarını kullandılar.

VGGNet — Bu ağ, ağ derinliğinin Sinir ağları için ne kadar önemli olduğunu kanıtlamıştır. 16 tane Convolutional katman bulunur.

Kaynak: <https://rubikscodet.net/2018/02/26/introduction-to-convolutional-neural-networks/>

Convolutional Network

Deep Learning

TensorFlow

Machine Learning

Artificial Intelligence



Follow

Written by Tuncer Ergin

320 Followers

Software Engineer 🖥️ 😊

More from Tuncer Ergin



Tuncer Ergin

Keras ile Derin Öğrenme Modeli Oluşturma

Derin öğrenme, makine öğreniminin giderek popülerleşen bir alt kümesidir. Derin öğrenme modelleri yapay sinir ağları kullanılarak inşa...

7 min read · Oct 2, 2018



155



3



Tuncer Ergin

Evrişimsel Sinir Ağında Niçin Dropout Kullanmamalısınız

Dropout katmanı ne iş yapar?

4 min read · Oct 2, 2018



141



1



Tuncer Ergin

Yapay Zeka'da Hold-out & Cross validation Nedir?

Cross validation genellikle en çok tercih edilen yöntemdir, çünkü modelinize birden fazla tren testi bölmesi için eğitim olanağı verir.

2 min read · Oct 5, 2018



47





Tuncer Ergin

Konvolüsyonel Sinir Ağı ve OpenCV Kullanarak Yaş ve Cinsiyet Tahmini

“Otomatik yaş ve cinsiyet sınıflandırması, özellikle sosyal platformların ve sosyal medyanın yükselişiyle birlikte iyice rağbet görmeye...

6 min read · Apr 23, 2019



36



1

[See all from Tuncer Ergin](#)

Recommended from Medium



Vaibhav Rastogi

Fully Connected Layer

Fully Connected Layer:

2 min read · Sep 8, 2023



34



1



Juan C Olamendy

Back to Basics: Feature Extraction with CNN

If you've ever wondered how computers can see and understand the world through images, you're in for a treat!

4 min read · Oct 19, 2023



Lists



Predictive Modeling w/ Python

20 stories · 886 saves



Natural Language Processing

1177 stories · 654 saves



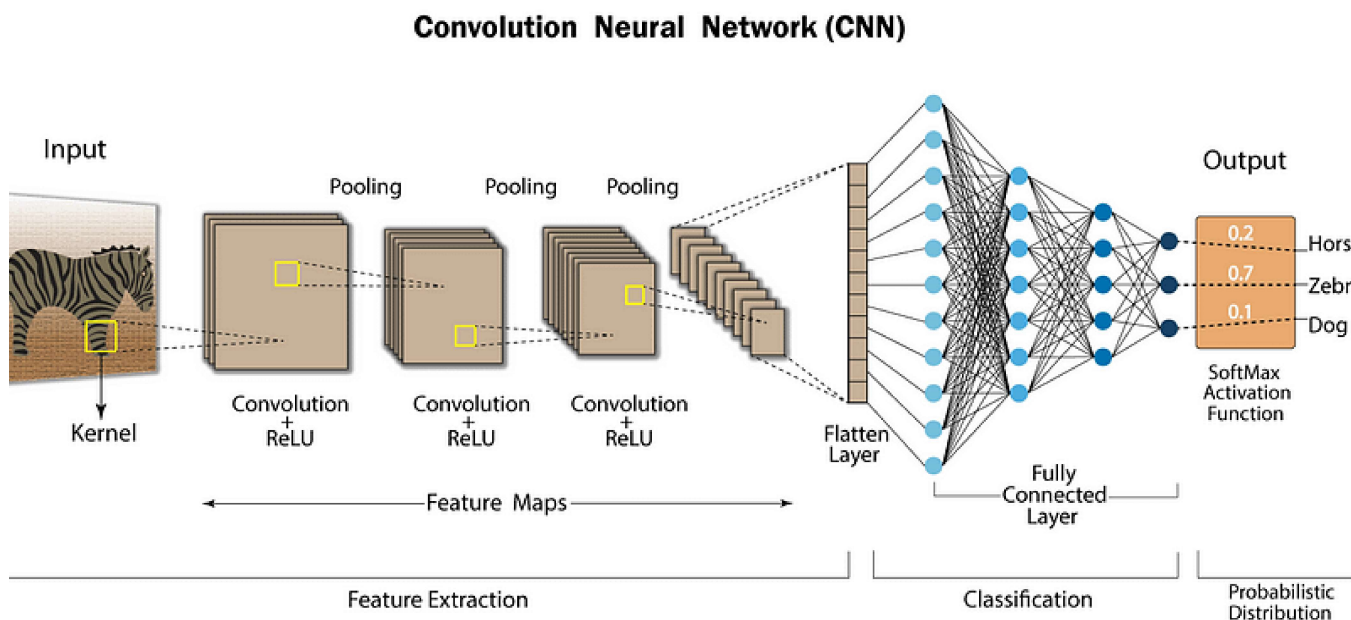
Practical Guides to Machine Learning

10 stories · 1028 saves



AI Regulation

6 stories · 311 saves



Koushik

Understanding Convolutional Neural Networks (CNNs) in Depth

Convolutional Neural Networks skillfully capturing and extracting patterns from data, revealing the hidden artistry within pixels.

12 min read · Nov 28, 2023



580



5



Cristian Leo in Towards Data Science

The Math behind Adam Optimizer

Why is Adam the most popular optimizer in Deep Learning? Let's understand it by diving into its math, and recreating the algorithm.

16 min read · Jan 30



1.2K



8





Prathammodi

Convolutional Neural Networks for Dummies

So you want to learn about Convolutional Neural Networks, CNNs, huh? Well, you've come to the right place.

6 min read · Oct 14, 2023



149



Azeem - I

Understanding ResNet Architecture: A Deep Dive into Residual Neural Network

Residual Network is a deep Learning model used for computer vision applications. The ResNet (Residual Neural Network) architecture was...

6 min read · Nov 14, 2023



12



See more recommendations