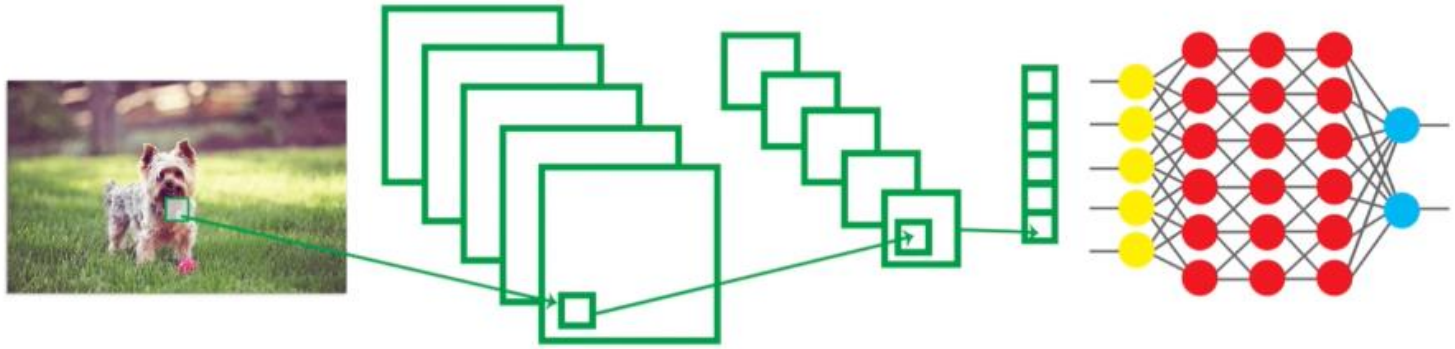
CNN

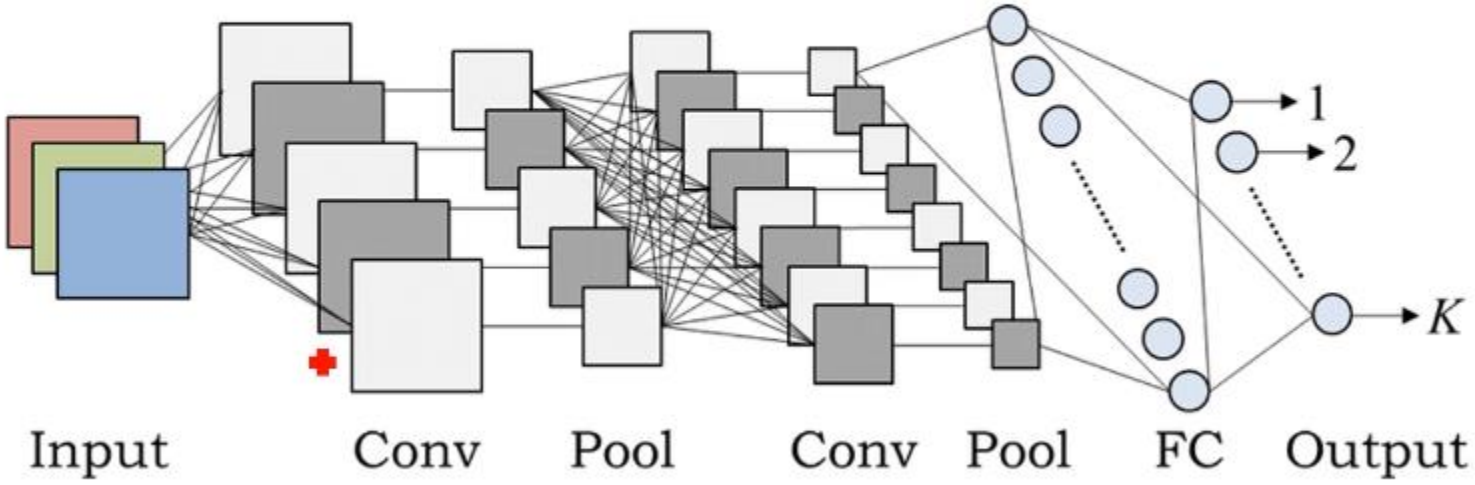
CNN (Convolutional Neural Network - Evrişimli Sinir Ağı)

- CNN (Convolutional Neural Network - Evrişimli Sinir Ağı), özellikle görsel veri işleme için tasarlanmış derin öğrenme modellerinin bir türüdür. CNN'ler, görüntü tanıma, sınıflandırma ve segmentasyon gibi görevlerde yaygın olarak kullanılırlar ve karmaşık görsel verilerden özellikleri otomatik olarak öğrenebilirler.



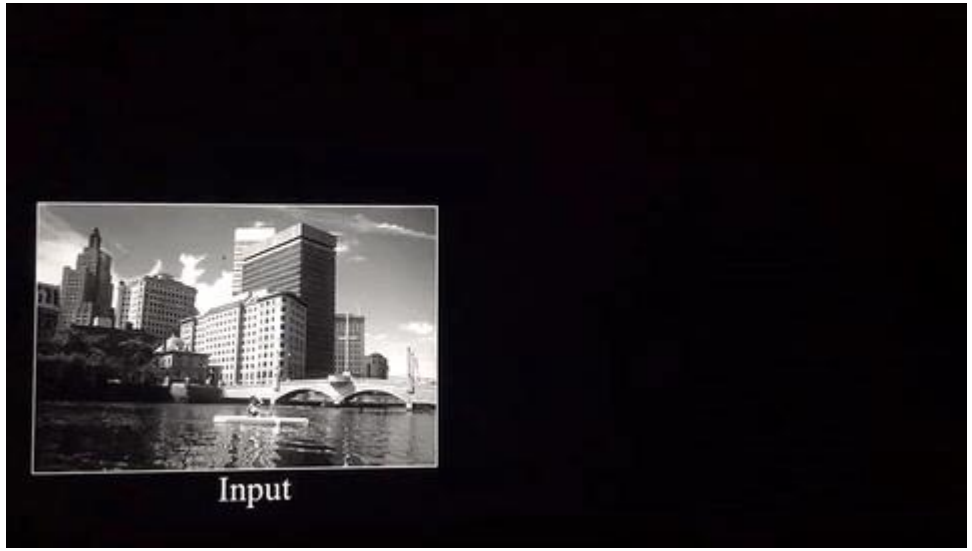
ConvolutionNeural Network

- CNN: Convolutional Neural Network



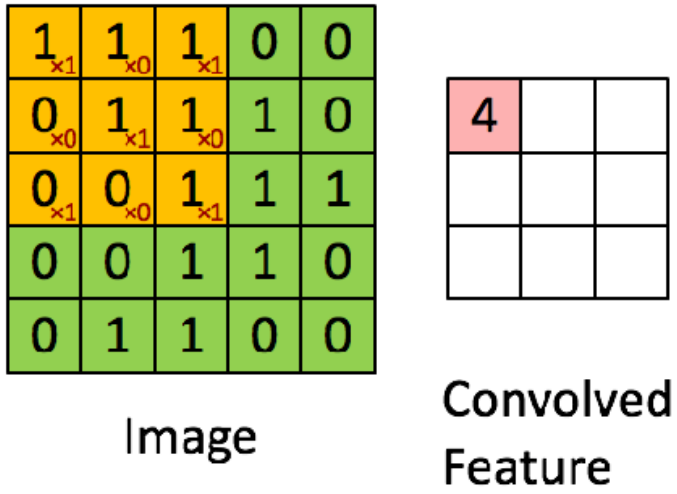
Evrişim Katmanları (Convolutional Layers):

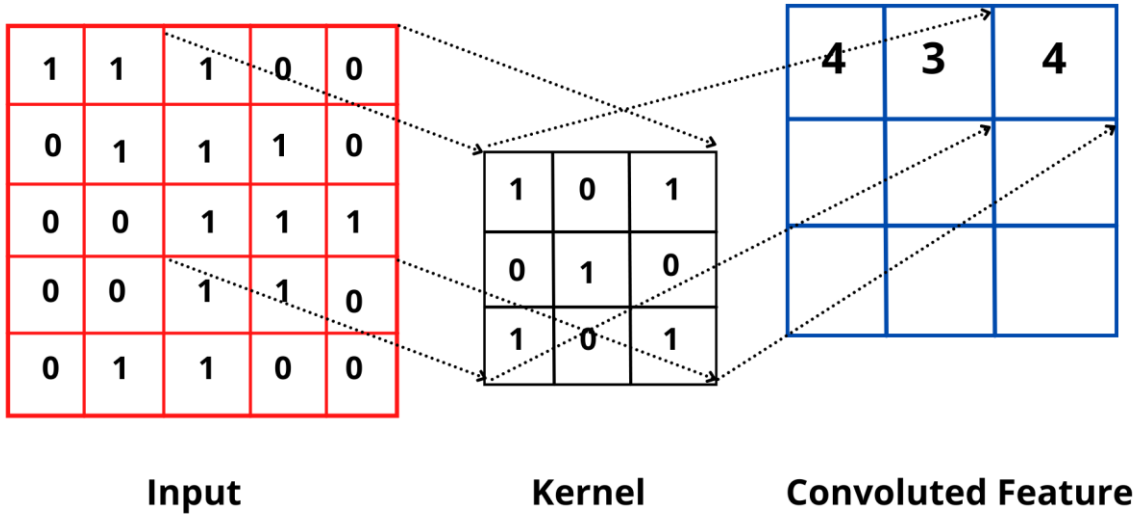
- Bu katmanlar, filtre veya çekirdek adı verilen ağırlık setlerini kullanarak, giriş verisi üzerinde yerel desenleri tespit eder. Her filtre, belirli bir özelliği (örneğin kenarlar, doku, vb.) yakalamak için tasarlanmıştır.



Kernel Size (Çekirdek Boyutu)

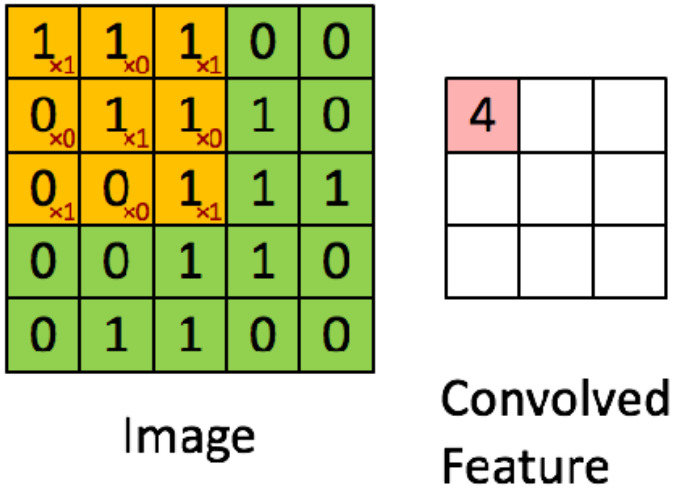
- Filtrelerin boyutunu belirler (örneğin, 3x3, 5x5). Bu boyut, her bir evrişim işlemi sırasında girdi üzerindeki kayan pencerenin boyutudur.





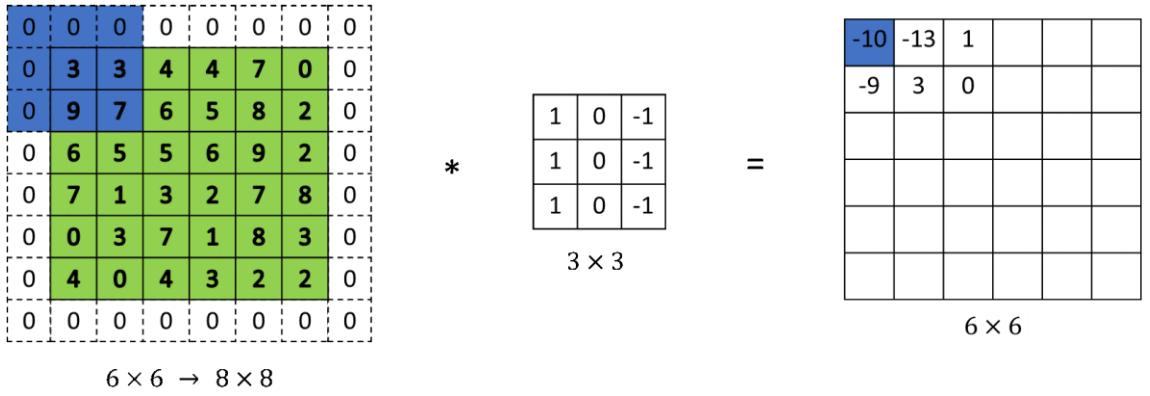
Strides (Adımlar)

- Filtrenin her seferinde girdi üzerinde ne kadar hareket edeceğini belirler. Örneğin, strides=(2, 2) ayarı, filtrenin hem yatay hem de dikey yönde iki piksel atlayarak hareket etmesi anlamına gelir.



Padding (Dolgu)

- Girdinin kenarlarına dolgu eklenip eklenmeyeceğini ve nasıl ekleneceğini belirler. padding='same', girdi ve çıktı özellik haritalarının aynı boyutlarda olmasını sağlamak için girdiye dolgu ekler. padding='valid', dolgu eklenmeyeceği anlamına gelir ve bu durumda çıktı boyutu genellikle girdiden daha küçük olur.



Activation Function (Aktivasyon Fonksiyonu)

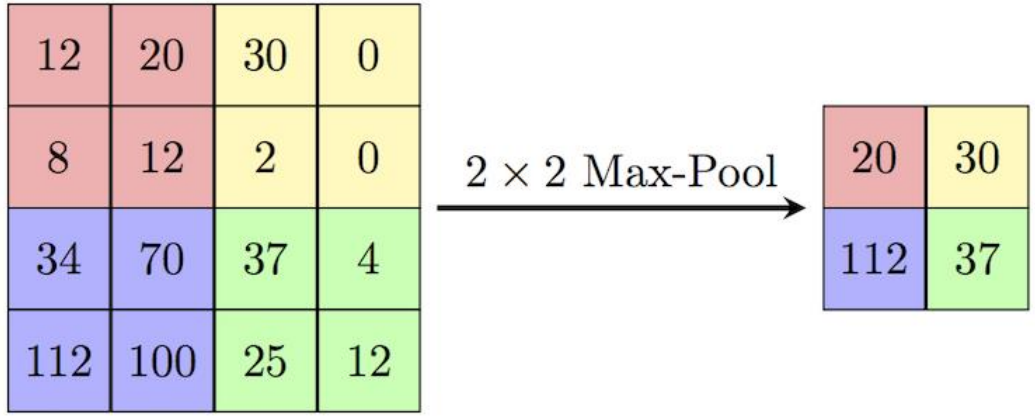
- Katmandan geçen verilerin aktivasyon fonksiyonuna tabi tutulup tutulmayacağını belirler. Çoğu zaman ReLU (Rectified Linear Unit) fonksiyonu kullanılır.
- ReLU, hesaplama açısından diğer aktivasyon fonksiyonlarına göre çok daha verimlidir. Nöronun çıktısı ya sıfırdır ya da girdi değeri kadardır.

Input Shape (Girdi Şekli)

- İlk evrişim katmanında, girdi verisinin boyutlarını belirtir. Örneğin, 28x28 piksellik gri tonlamalı bir görüntü için input_shape=(28, 28, 1) olur.

Pooling Katmanı

- Pooling katmanı, Convolutional Neural Network (CNN) mimarisinin bir parçası olarak kullanılan bir katmandır ve genellikle evrişim (convolution) katmanlarından sonra gelir. Pooling katmanının temel amacı, giriş olarak aldığı özellik haritalarının (feature maps) boyutunu azaltmaktır. Bu işlem ağın parametre sayısını ve hesaplama yükünü azaltır.

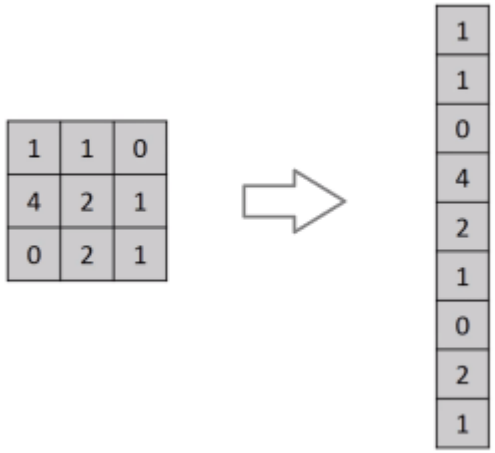


Pooling katmanı

1. **Max Pooling:** Max pooling, bir özellik haritasındaki belirli bir pencere boyutu içindeki maksimum değeri alır ve geri kalan bilgiyi atar. Bu, genellikle en önemli özelliği (en yüksek aktivasyonu) korurken, diğer tüm özellikleri yok sayar.
2. **Average Pooling:** Average pooling, belirli bir pencere boyutu içindeki tüm değerlerin ortalamasını alır ve böylece o penceredeki genel aktivasyon seviyesini temsil eden bir değer üretir.

Flatten Katmanı

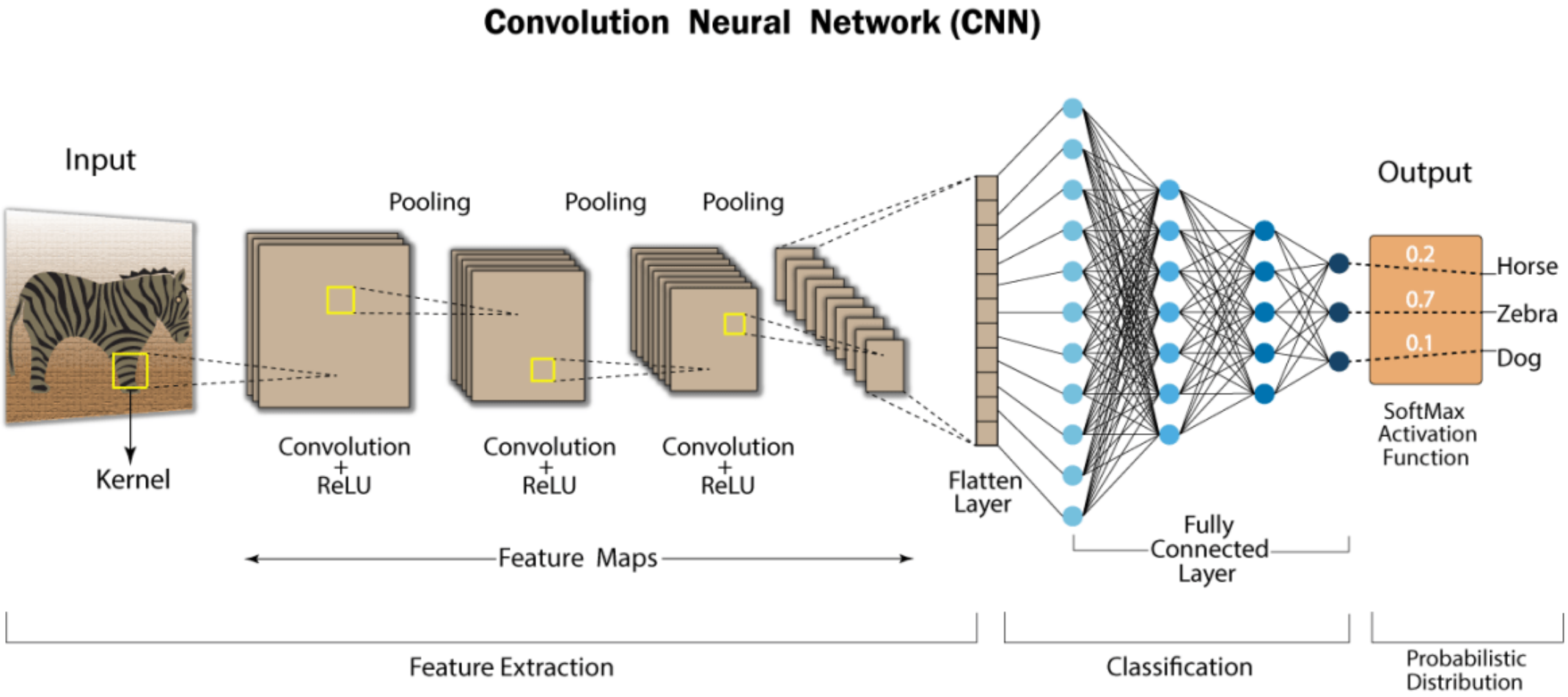
- Evrişimli ve pooling katmanlarından elde edilen özellik haritaları (feature maps) çok boyutludur. Flatten katmanı bu çok boyutlu özellik haritalarını tek boyutlu bir vektöre dönüştürür. Bu, özellik haritalarını tam bağlantılı katmanlara uygun hale getirir.



Tam Bağlantılı Katmanlar(Dense)

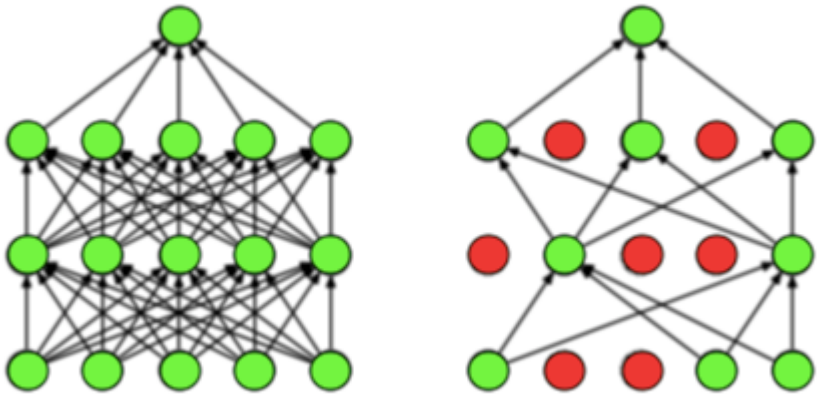
- Bir yapay sinir ağında, "Tam Bağlantılı Katman", ağın öğrenmesi ve karar vermesi için kullanılan bir bölümdür. Bu katmanda yer alan nöronlar, ağın önceki katmanlarından gelen tüm bilgileri alır. Her bir nöron, bu bilgileri birleştirip analiz eder ve sonunda bir sonuç (örneğin, bir resmin hangi sınıfa ait olduğu gibi) üretir. Bu katmanlar, genellikle ağın en son kısmında bulunur ve ağın öğrendiği her şeyi kullanarak nihai bir karar vermesini sağlar. Örneğin, bir fotoğrafta bir kedi olup olmadığını belirlemek gibi. Bu katmanlar, yapay sinir ağlarının "düşünme" ve "karar verme" kısmı gibidir.

Tam Bağlantılı Katmanlar



Dropout

- Dropout, yapay sinir ağlarındaki bir düzenleme (regularization) tekniğidir ve aşırı uyum (overfitting) sorununu azaltmaya yardımcı olur. Dropout işlemi, eğitim süreci sırasında rastgele seçilen nöronları (ve bunların bağlantılarını) geçici olarak ağdan çıkarmak anlamına gelir.

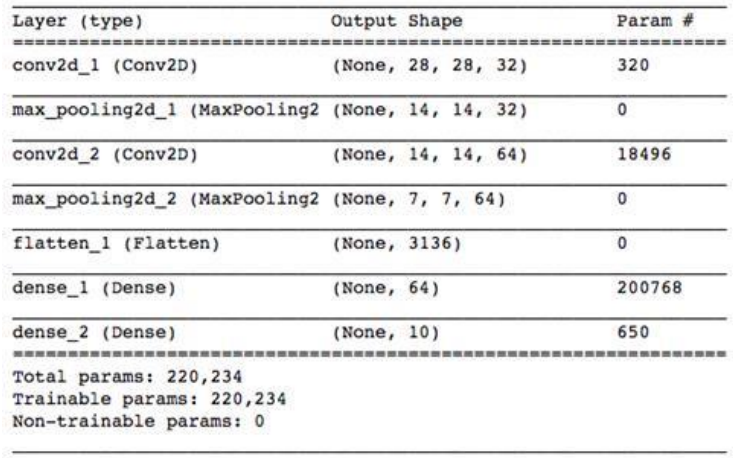


(a) Standard Neural Net

(b) After applying dropout.

Model Özeti (summary())

- Bir yapay sinir ağı modelinin mimarisini özetleyerek, modelin katmanları, her katmandaki parametre sayısı ve diğer önemli bilgileri gösterir. Bu özet, modelinizi analiz etmenize ve anlamanıza yardımcı olur, özellikle modelin boyutu ve karmaşıklığı hakkında hızlı bir fikir edinmek istediğinizde kullanışlıdır.



Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 28, 28, 32)	320
max_pooling2d_1 (MaxPooling2	(None, 14, 14, 32)	0
conv2d_2 (Conv2D)	(None, 14, 14, 64)	18496
max_pooling2d_2 (MaxPooling2	(None, 7, 7, 64)	0
flatten_1 (Flatten)	(None, 3136)	0
dense_1 (Dense)	(None, 64)	200768
dense_2 (Dense)	(None, 10)	650

Total params: 220,234
Trainable params: 220,234
Non-trainable params: 0

Compile işlemi

- Yapay sinir ağları ve derin öğrenme kontekstinde, "compile" işlemi, bir modelin eğitim öncesi hazırlık aşamasını ifade eder. Bu aşama, özellikle Keras gibi yüksek seviye derin öğrenme kütüphanelerinde yaygındır. Compile işlemi sırasında, modelin nasıl eğitileceği ve güncelleneceği belirtilir.
- **Optimizer Belirleme**
- **Loss Fonksiyonu Seçimi**
- **Metriklerin Tanımlanması**

```
model.compile(optimizer='adam',
              loss='categorical_crossentropy',
              metrics=['accuracy'])
```

Eğitim

- CNN'de (Convolutional Neural Network) modeli "fit etmek", modelin eğitim verileri üzerinde eğitilmesi sürecini ifade eder. Bu süreçte, model veri setinden örnekler alır, bu örnekler üzerinden tahminlerde bulunur ve tahminlerin doğruluğuna göre kendini iteratif olarak günceller.
- **epochs:** Epoch sayısı. Bir epoch, eğitim veri setinin tamamının modele tam olarak bir kez sunulduğu bir dönemdir. Daha fazla epoch, genellikle modelin eğitim verilerini daha iyi öğrenmesini sağlar, ancak overfitting riskini de artırabilir.

Eğitim

- **batch_size:** Eğitim sırasında her bir adımda (iteration) modele verilen örnek sayısıdır. Büyük bir batch_size, her adımda daha fazla örneğin işlenmesini sağlar, bu da genellikle daha hızlı eğitim süreçleri anlamına gelir. Ancak, daha büyük bir batch boyutu daha fazla bellek gerektirir.

- **steps_per_epoch:** Bir eğitim döngüsü (epoch) içinde model tarafından gerçekleştirilecek toplam adım sayısıdır. Bir adım, bir batch verinin işlenmesi anlamına gelir. Yani steps_per_epoch, bir epoch içinde kaç kez ağırlık güncellemesi yapılacağını belirler.

Eğitim

- Eğer bir eğitim veri setiniz varsa ve her epoch'ta tüm veri setini modele sunmak istiyorsanız, **steps_per_epoch'u** veri setinizin toplam boyutunu batch_size'a bölerek hesaplayabilirsiniz.

- Örneğin, 1000 örneklik bir veri setiniz ve batch_size=20 ayarınız varsa, her epoch için 50 adım yapmanız gerekir (1000 / 20 = 50). Bu durumda steps_per_epoch=50 olarak ayarlanabilir.