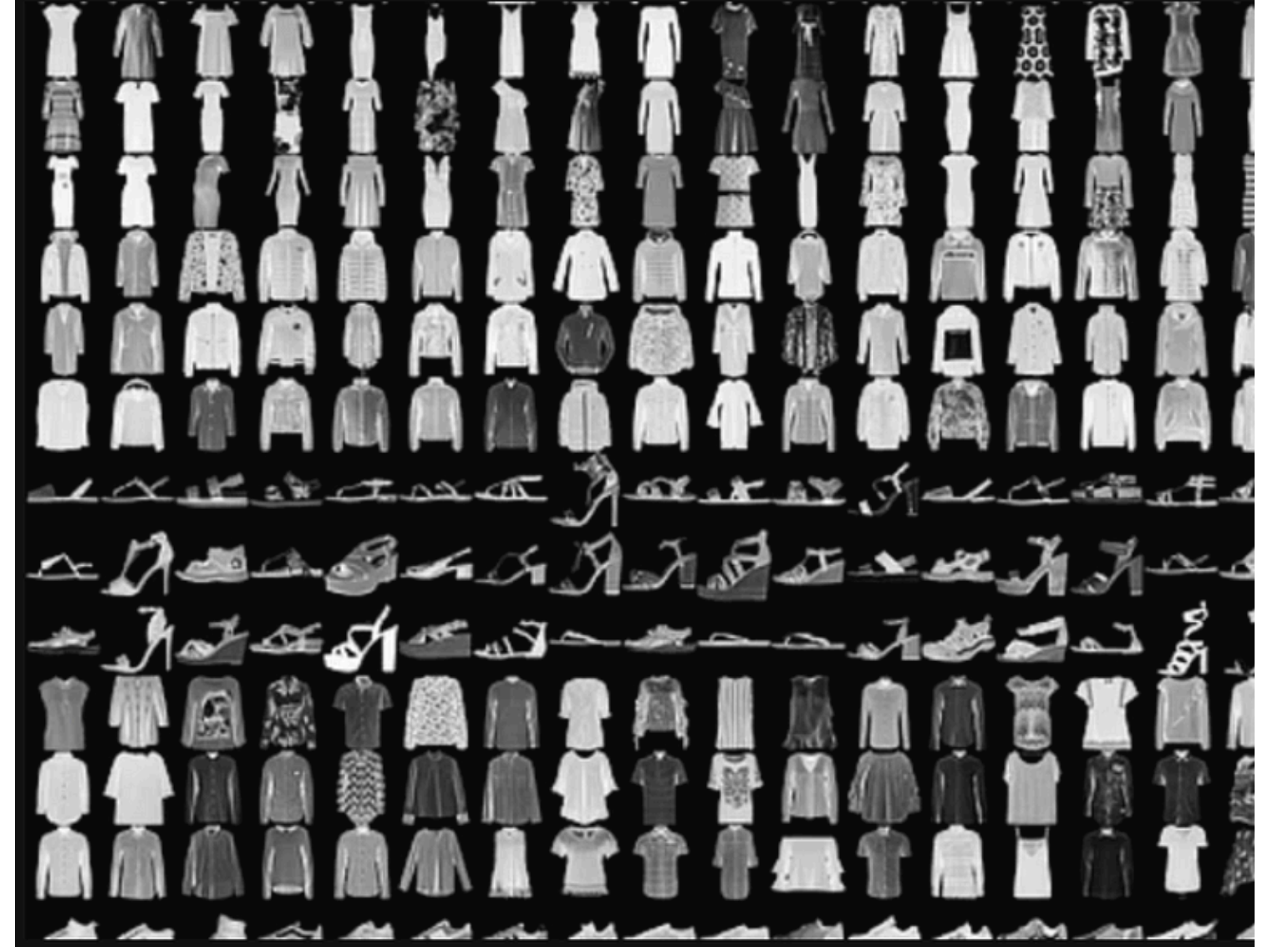


Proje Tanıtımı

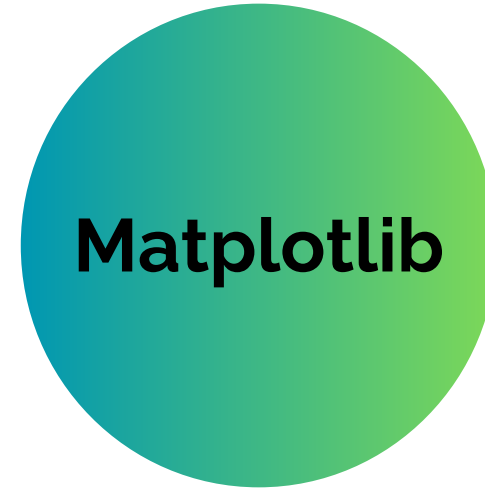
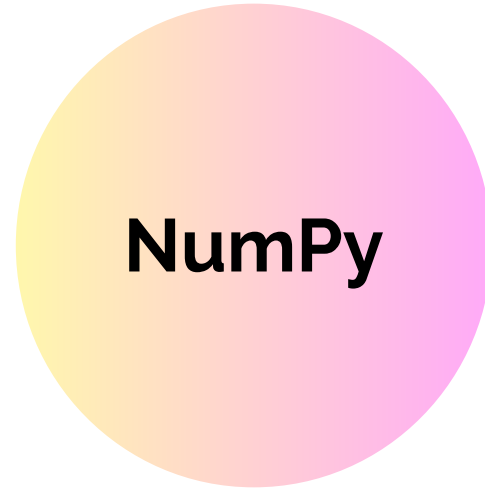
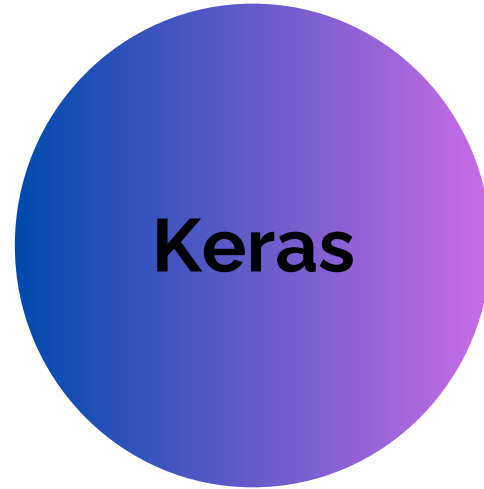
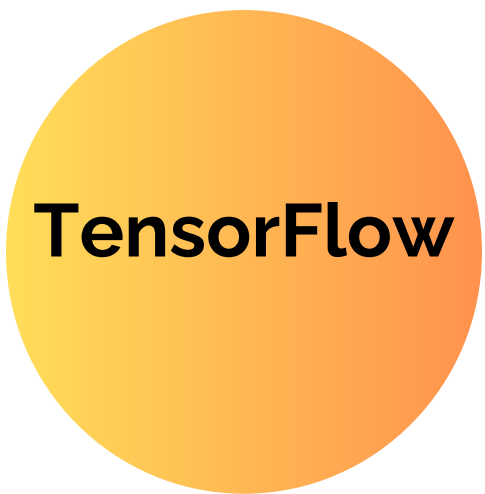


<https://medium.com/@moubine.rabab/fashion-mnist-dataset-427c977358b1>

Projenin Amacı ve Kapsamı:

Projede Fashion MNIST veri setini kullanarak bir Convolutional Neural Network (CNN) modeli eğitilmiştir. Modelim, giyim eşyalarını sınıflandırmak için tasarlanmıştır. Python dili kullanılarak Google Colab'da geliştirilmiştir.

Kullanılan Araçlar ve Teknolojiler:



- Veri Seti -

Fashion MNIST veri seti, toplamda 70,000 adet 28x28 piksel boyutunda gri tonlamalı görüntü içerir.

Veri Seti Özellikleri

Fashion MNIST veri seti, Keras kütüphanesinin içinde hazır olarak bulunur. Veri setinde, 10 sınıf, her sınıfta 60,000 eğitim ve 10,000 test görüntüsü bulunur.

Veri Setinin Yapısı

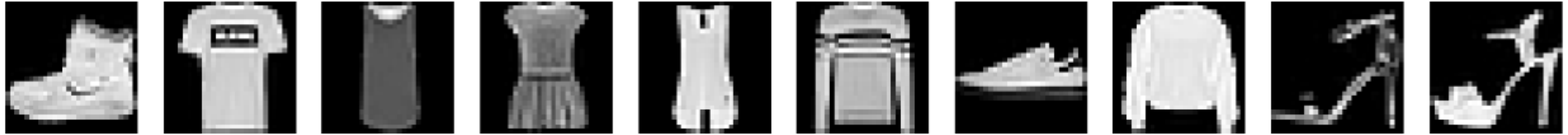
1. Eğitim Seti (Training Set): Modeli eğitmek için kullanılan veri seti.
 - Görüntüler (X_{train}): 60,000 adet 28x28 piksel boyutunda gri tonlamalı görüntü.
 - Etiketler (y_{train}): Bu 60,000 görüntüye karşılık gelen sınıf etiketleri. Her etiket 0 ile 9 arasında bir sayı olup, görüntünün hangi sınıfa ait olduğunu belirtir.
2. Test Seti (Test Set): Modelin performansını değerlendirmek için kullanılan veri seti.
 - Görüntüler (X_{test}): 10,000 adet 28x28 piksel boyutunda gri tonlamalı görüntü.
 - Etiketler (y_{test}): Bu 10,000 görüntüye karşılık gelen sınıf etiketleri.

Etiketler (Labels) Nedir?

Etiketler, her bir görüntünün hangi sınıfa ait olduğunu belirten değerlerdir. Fashion MNIST veri setinde, her görüntü bir giyim eşyasını temsil eder ve bu giyim eşyasının sınıf etiketi 0 ile 9 arasında bir sayı olarak belirlenmiştir. Örneğin: 0-Tişört/Üst (T-shirt/top), 1-Pantolon (Trousers) gibi..

Veri setinin görselleştirilmesi

Colab'da ilk 3 kod bloğunda gerekli kütüphaneler import edildikten sonra Fashion MNIST veri seti, Keras kütüphanesi tarafından otomatik olarak yüklenir ve eğitim ile test setlerine bölünür. Bu bölünme, veri setinin yapısı gereği önceden belirlenmiştir. Veri setinin boyutlarını kontrol ederek ve görüntüleri görselleştirerek, veri setinin içeriğini daha iyi anlayabilir ve model eğitime başlamadan önce veri setinin doğru bir şekilde yüklendiğini doğrularız.



Neden 60,000 ve 10,000?

Bu bölünme, veri setinin oluşturulması sırasında belirlenmiş bir kuraldır ve veri seti bu şekilde yayınlanmıştır. Eğitim ve test veri setlerinin boyutları, modelin eğitim performansını değerlendirmek ve genelleme yeteneğini ölçmek için uygun olacak şekilde belirlenmiştir.

Verilerin normalize edilmesi

$X_{\text{train}} = X_{\text{train}} / 255.0$

$X_{\text{test}} = X_{\text{test}} / 255.0$

Görüntü verileri başlangıçta 0-255 aralığında piksel değerlerine sahiptir. X_{train} ve X_{test} veri setlerini 255'e bölerek tüm piksel değerlerini 0-1 aralığına çekiyoruz. Bu, modelin daha stabil ve hızlı öğrenmesine yardımcı olur.

CNN Modelin Oluşturulması

Convolutional Neural Network (CNN), özellikle görüntü verileri üzerinde kullanılan bir tür derin öğrenme modelidir. CNN'ler, görüntüleri işleyerek bu görüntülerdeki özellikleri tanır ve sınıflandırma gibi görevleri gerçekleştirir. Bu modeller, evrişim (convolution), havuzlama (pooling) ve tam bağlantılı (fully connected) katmanlardan oluşur.

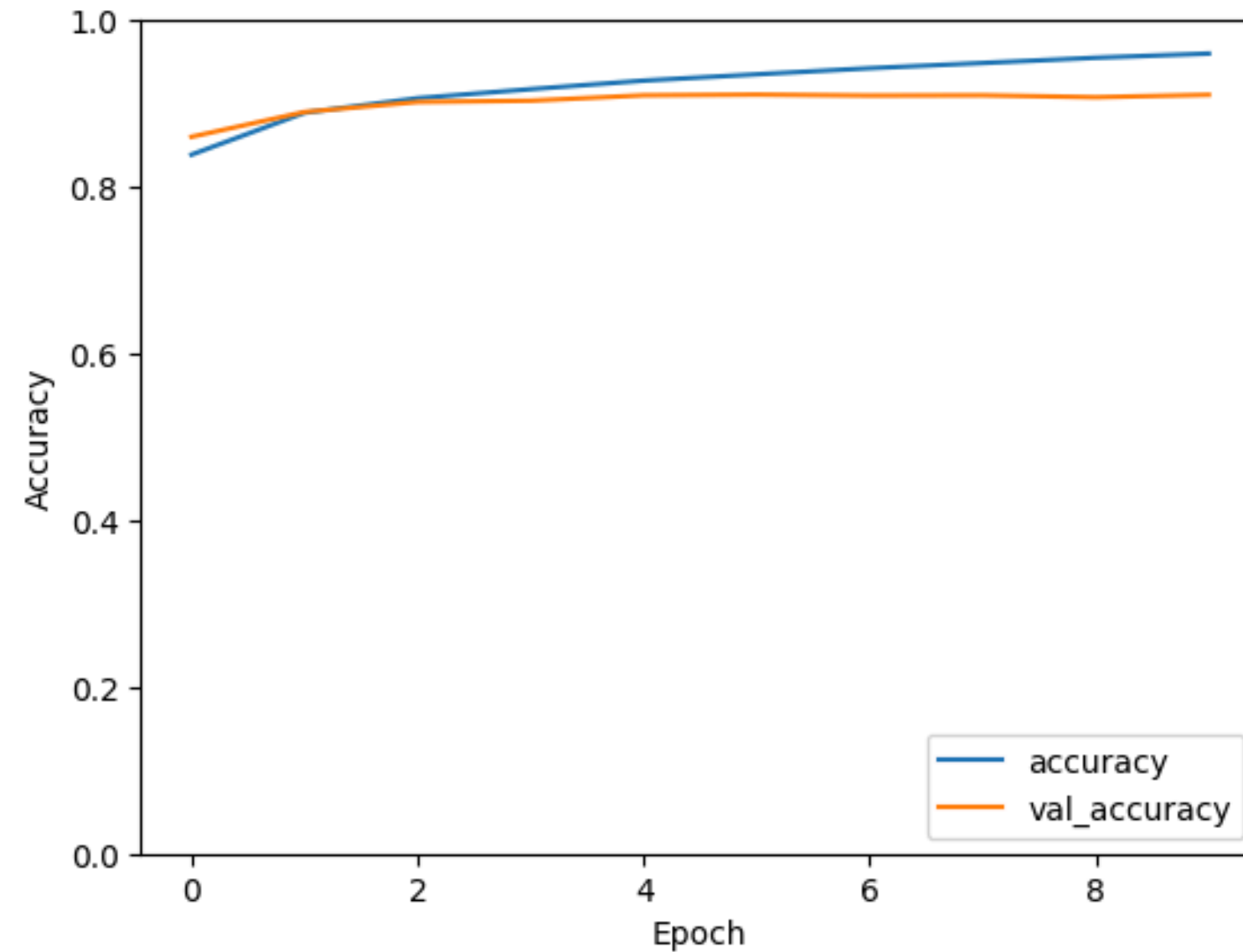
CNN modelinin katmanları ve mimarisi kod bloklarında belirlenmiştir. (Conv2D, MaxPooling2D, Flatten, Dense). Modelin optimizer, loss ve metric parametreleri derlenir.

Özetle:

- Eğitim (Training): CNN Modeli, eğitim veri seti üzerinde eğitilir. Bu süreçte, modelin ağırlıkları güncellenir ve model, veri setindeki görüntüleri ve bunlara ait etiketleri öğrenir.
 - Değerlendirme (Evaluation): Modelin performansı test veri seti üzerinde değerlendirilir. Test veri seti, modelin daha önce görmediği görüntülerden oluşur ve modelin genelleme yeteneğini ölçmek için kullanılır.
-

Sonuç

Grafik, modelin eğitim süreci boyunca elde ettiği doğruluk değerlerini görselleştirir. Bu grafik, eğitim ve doğrulama (validation) veri setleri üzerindeki doğruluk metriklerini gösterir ve her bir epoch (eğitim iterasyonu) için doğruluk değerlerinin nasıl değiştiğini gösterir. Grafiğin x-ekseni epoch sayısını, y-ekseni ise doğruluk değerini temsil eder.



Confusion Matrix (Karmaşıklık Matrisi)

Karmaşıklık matrisi, modelin farklı sınıfları ne kadar doğru veya yanlış tahmin ettiğini açıkça gösterir ve modelin hangi alanlarda daha fazla iyileştirme gerektirdiğini belirlememize yardımcı olur. Bu sayede, modelin performansını daha iyi anlayabilir ve gerekli iyileştirmeleri yapabiliriz.

```
[[852  0  24  12  5  2 101  0  4  0]
 [ 2 975  2  13  4  0  3  0  1  0]
 [ 14  0 882  9  42  0  52  0  1  0]
 [ 14  1  7 940  19  1  17  0  1  0]
 [ 1  0  61  25 880  1  30  0  2  0]
 [ 0  0  0  0  0 993  0  4  0  3]
 [109  0  62  27  85  1 710  0  5  1]
 [ 0  0  0  0  0  29  0 923  0  48]
 [ 5  0  5  1  6  2  6  1 971  3]
 [ 1  0  0  0  0  5  0  16  0 978]]
```

Karmaşıklık matrisindeki her bir hücre, gerçek sınıfları (satırlar) ve modelin tahmin ettiği sınıfları (sütunlar) gösterir. Örneğin, 0. satır 3. sütundaki hücre, gerçek sınıfı tişört olan örneklerin, model tarafından kaç kez kazak olarak tahmin edildiğini ifade eder.

Doğruluk Parametreleri

- Precision (Hassasiyet): Precision, modelin pozitif olarak tahmin ettiği örneklerin ne kadarının gerçekten pozitif olduğunu gösterir. Yani, bu metrik, modelin doğru pozitiflerin toplam pozitif tahminlere oranını ifade eder. Örneğin, sınıf 1 için hassasiyet %100 olduğu için, modelin sınıf 1 olarak tahmin ettiği tüm örneklerin gerçekte sınıf 1 olduğu söylenebilir.
- Recall (Geri Çağırma): Recall, gerçek pozitiflerin (doğru tahminlerin) tüm gerçek pozitiflere oranını ifade eder. Yani, bu metrik, modelin gerçek pozitifleri kaçırma oranını gösterir. Örneğin, sınıf 6 için geri çağırma %71 olduğu için, gerçek sınıf 6 olan tüm örneklerin %71'inin model tarafından doğru bir şekilde sınıflandırıldığı söylenebilir.
- F1-Score: F1-score, hassasiyet ve geri çağırmanın harmonik ortalamasıdır. Bu metrik, dengeli bir performans sergileyip sergilemediğini belirlemek için kullanılır.

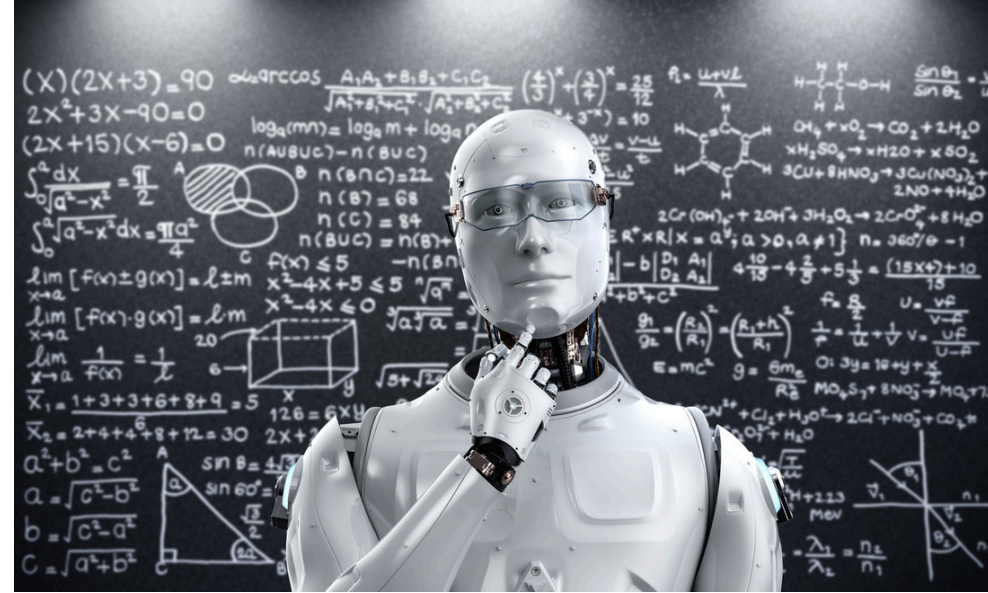
```
313/313 [=====] - 3s 8ms/step
```

	precision	recall	f1-score	support
0	0.85	0.85	0.85	1000
1	1.00	0.97	0.99	1000
2	0.85	0.88	0.86	1000
3	0.92	0.94	0.93	1000
4	0.85	0.88	0.86	1000
5	0.96	0.99	0.98	1000
6	0.77	0.71	0.74	1000
7	0.98	0.92	0.95	1000
8	0.99	0.97	0.98	1000
9	0.95	0.98	0.96	1000
accuracy			0.91	10000
macro avg	0.91	0.91	0.91	10000
weighted avg	0.91	0.91	0.91	10000

- Accuracy (Doğruluk): Doğruluk, modelin doğru tahmin ettiği örneklerin toplam örneklere oranını ifade eder. Bu değer, genellikle sınıf dengesizliği olmayan durumlarda kullanılır.

Özet

Model eğitim sırasında, eğitim veri seti üzerindeki örneklerle beslenir ve bu örneklerin her birinin doğru sınıfları bilinir. Model bu eğitim sürecinde, girdi verilerini ve doğru çıktıları (etiketlenmiş sınıfları) kullanarak ilişkileri öğrenir. Test sırasında, model eğitimde öğrendiği bu ilişkilere dayanarak, henüz görmediği (eğitim veri setinde olmayan) yeni veriler üzerinde tahminler yapar. Yani, test sırasında model gerçek sınıfları bilmez, ancak eğitim sırasında öğrendiği bilgilere dayanarak tahminler yapar. Test veri setindeki her bir örnek için, model tahminlerini yapar ve gerçek sınıfı ile karşılaştırır. Modelin doğruluğu ve güvenilirliği, uygulamanın başarısı üzerinde doğrudan bir etkiye sahiptir. Model, görüntü sınıflandırma problemlerinde, giysi tanıma uygulamalarında veya otomatik sürücüsüz araçlar gibi alanlarda kullanılabilir.



https://github.com/nisaBayhan/fashion_mnist_data



Teşekkürler!
