BİLGİSAYAR MÜHENDİSLERİ İÇİN DİFERANSİYEL DENKLEMLER ÖDEV RAPORU

Berke Bünyamin Süle [bunyamin.sule@std.yildiz.edu.tr](mailto:bunyamin.sule@std.yildiz.edu.tr)

Youtube video linki: <https://www.youtube.com/watch?v=jeyxHMTZaUE>

Youtube: Berke Süle

A: Görüntü sınıflandırma

* 2 sınıflı her sınıfta en az 100 örnek içeren gri tonlu görüntü veri kümesi (her pikselin değeri 0-255 arası ise 0-1 arası dönüşüm) oluştur: (bonus 4 sınıflı)

Veri kümesi olarak Kaggle’dan **handwritten digit** **MNIST in CSV** veri kümesini kullandım.

Link: <https://www.kaggle.com/datasets/oddrationale/mnist-in-csv>

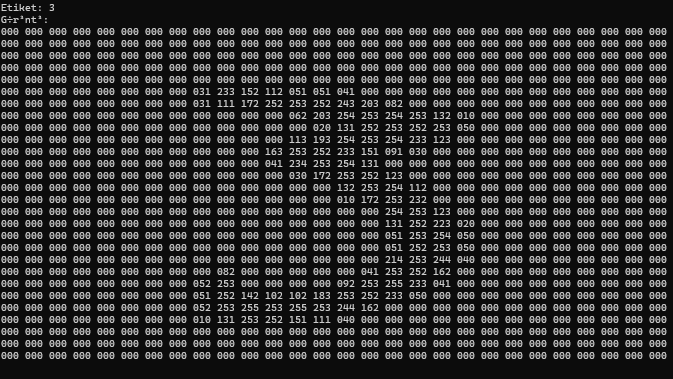
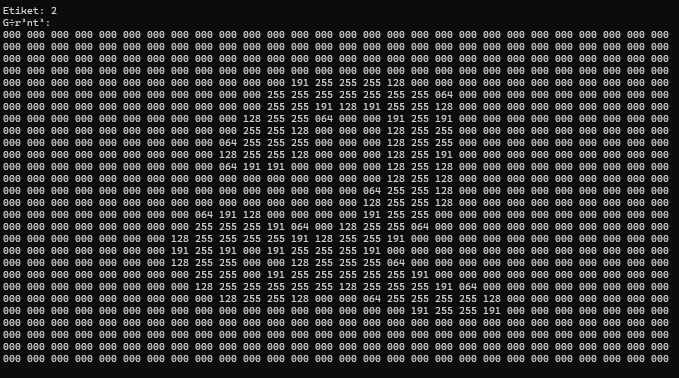
Bu csv dosyasını C’de kolay işleyebilmek için Python kullanarak txt formatına çevirdim:

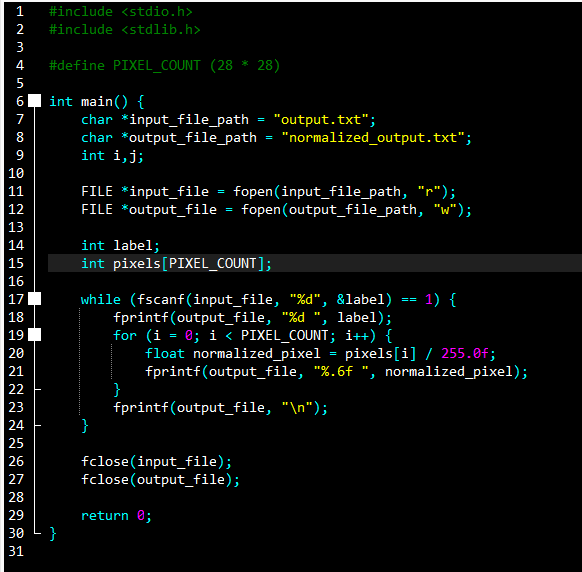


Normalde veri kümesi 10 sınıflı ben sadece ilk 4 sınıfı kullandım. Her sınıftan 500 görüntü aldım. Output dosyasında başta 1 satırda label onun altında 28\*28’lik pixel değerleri yer alıyor

Normalize Edilmiş Görüntüler: (dosya içeriği bu şekilde değil okunabilsin diye normalize edilmemiş halini bu formatta ekrana yazdırdım)

kalıp, desen, düzen, ekran görüntüsü, monokrom, tek renkli, siyah beyaz içeren bir resim

Açıklama otomatik olarak oluşturuldu

Görüntülerin pixel değerleri 0-255 arasındaydı. Aşağıdaki C kodu ile 0-1 arasına getirdim

* Görüntüleri N\*N’lik vektörler haline getir (N>20)

Görüntüler hazır 28\*28’lik halde.

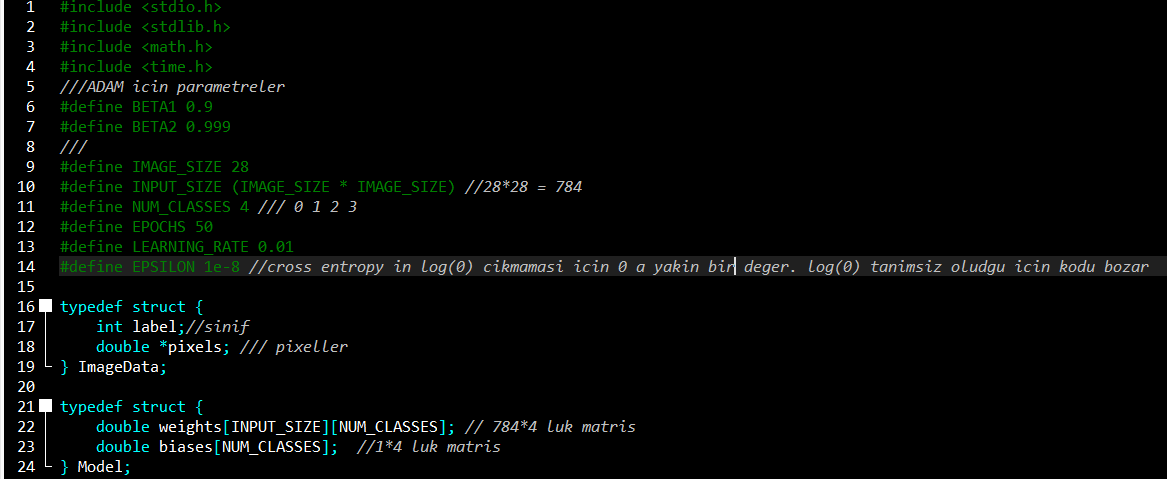
* Eğitim ve test kümesine böl (%80 / %20)

Aşağıdaki C kodu ile bu işlemi gerçekleştirdim: metin, ekran görüntüsü, yazılım, multimedya yazılımı içeren bir resim

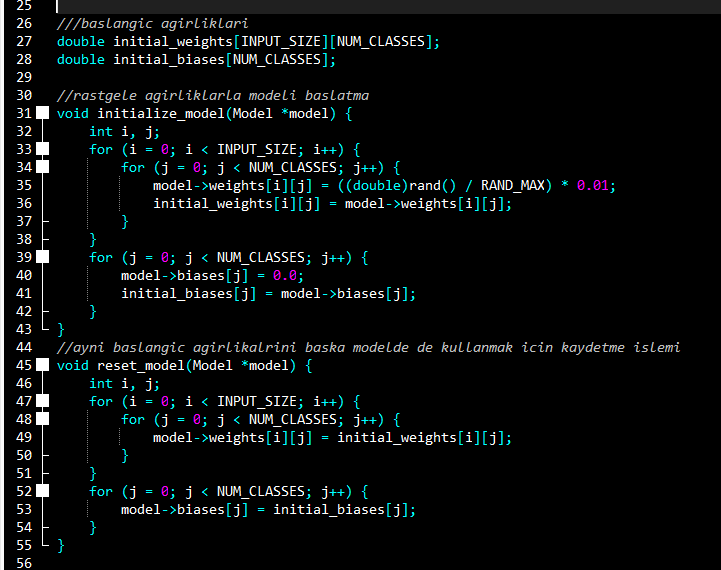
Açıklama otomatik olarak oluşturuldu

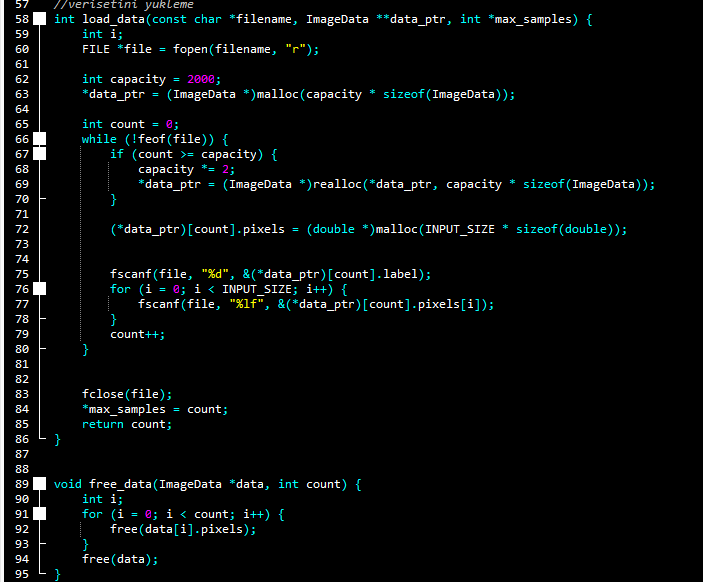
* Model Eğitimi

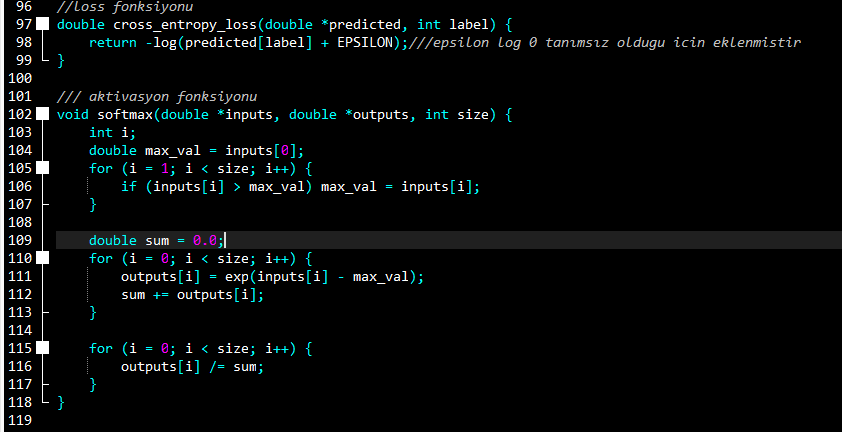
Aşağıdaki görsellerde model eğitiminde kullandığım C kodu var:



Modele başlangıçta rastgele ağırlıklar verdim. Reset model fonksiyonuyla başlangıçta verdiğim ağırlıkların aynısını GD, SGD ve Adam’da bire bir kullanmak için kopyalıyoruz.



Veri Yükleme: (hazır kod)

Aktivasyon ve Kayıp Fonksiyonları:

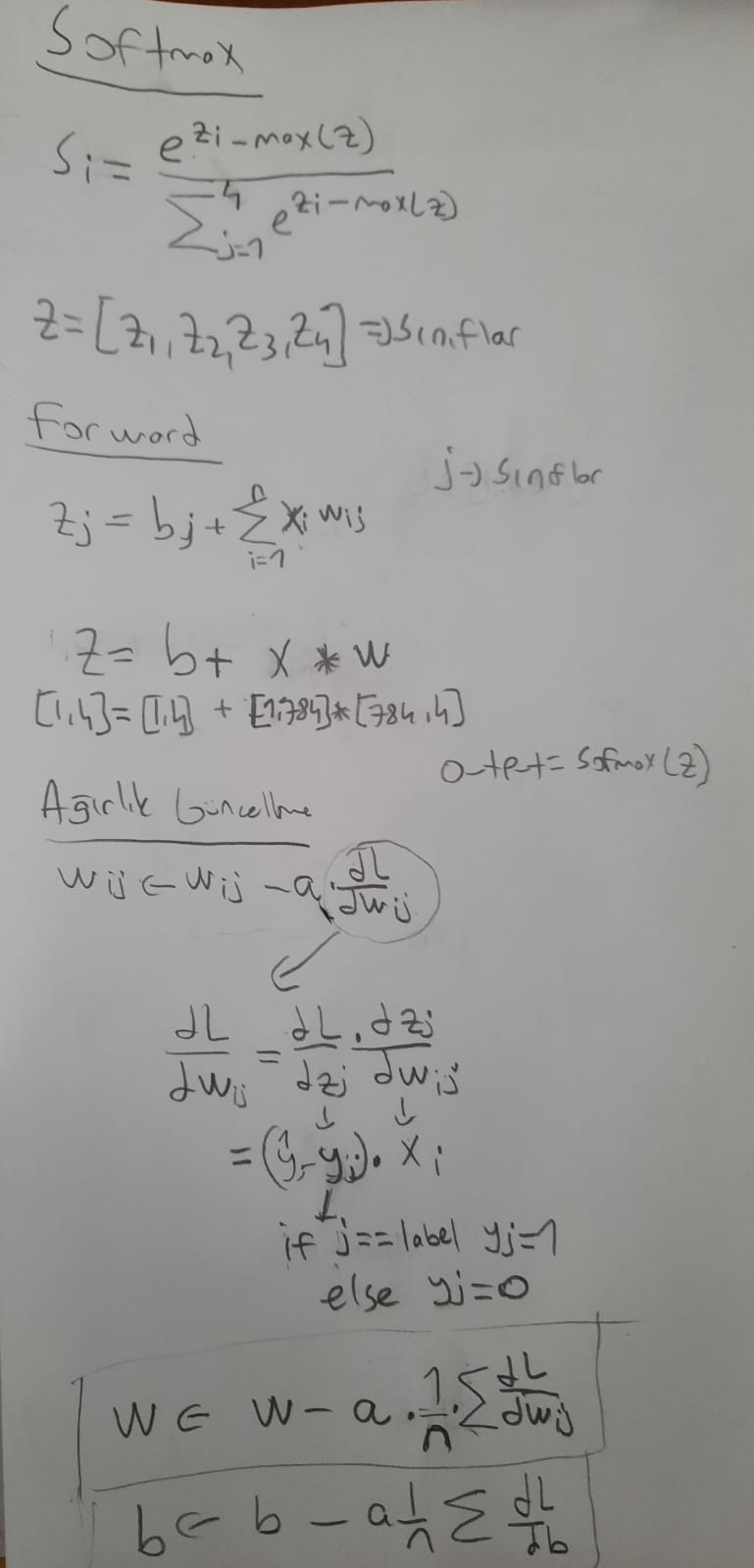
metin, ekran görüntüsü, yazılım içeren bir resim

Açıklama otomatik olarak oluşturuldu

Evaluate\_accuracy ile modelin test verisi üzerindeki doğruluğunu hesaplıyoruz.

metin, ekran görüntüsü, yazılım, ekran, görüntüleme içeren bir resim

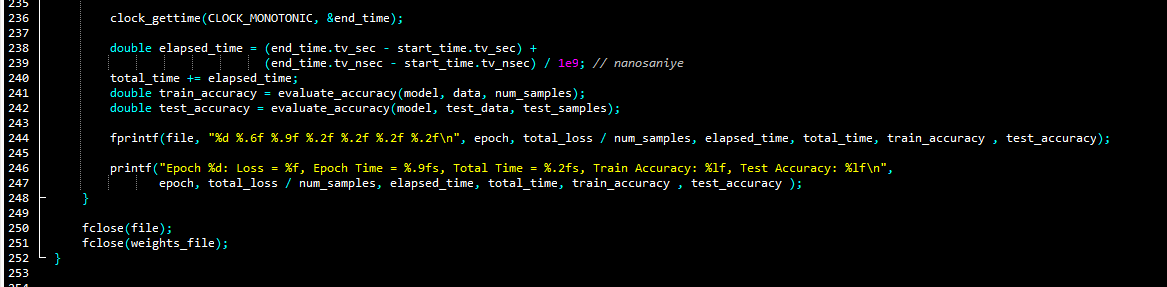
Açıklama otomatik olarak oluşturuldu



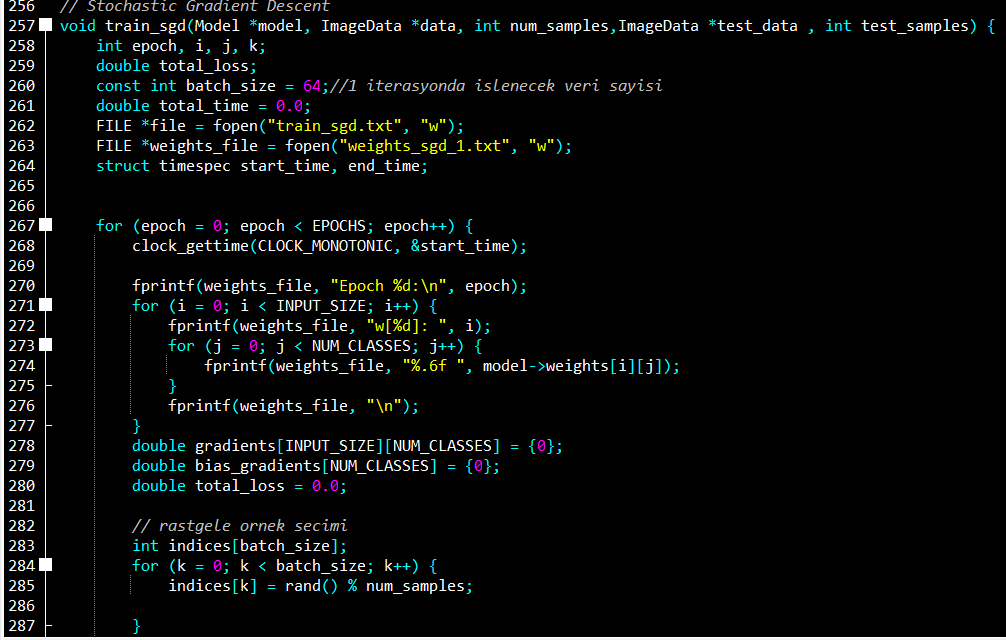
* Gradient Descent ile Eğitim:

metin, ekran görüntüsü, yazılım, multimedya yazılımı içeren bir resim

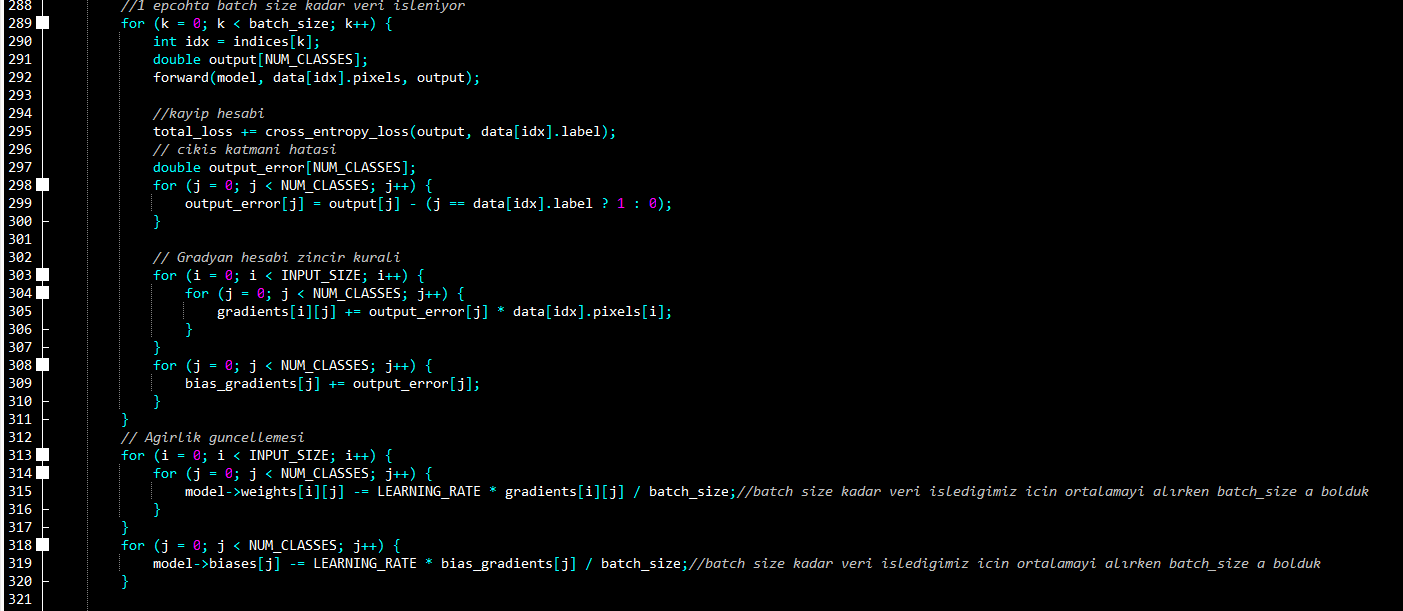
Açıklama otomatik olarak oluşturuldumetin, ekran görüntüsü, yazılım içeren bir resim

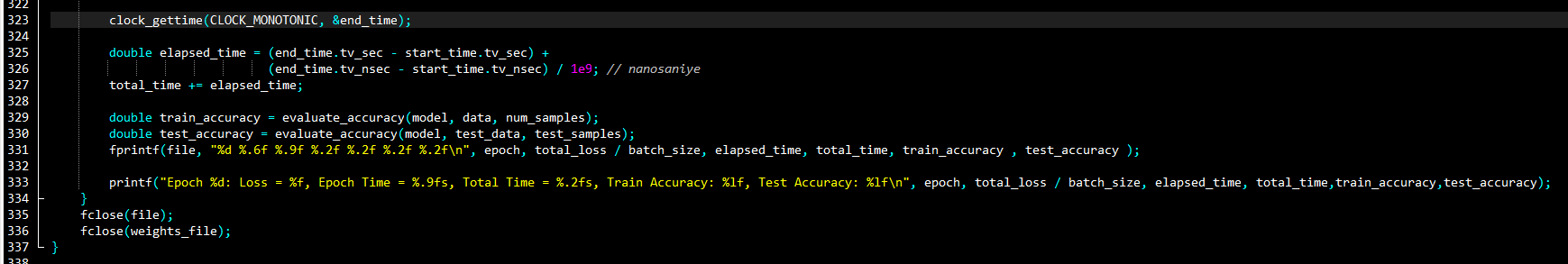
Açıklama otomatik olarak oluşturuldu

SGD ile Eğitim:

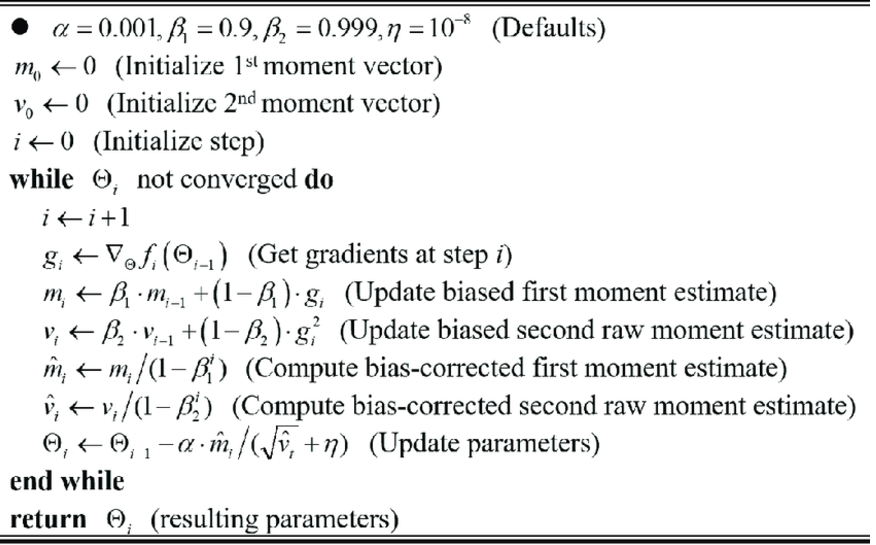


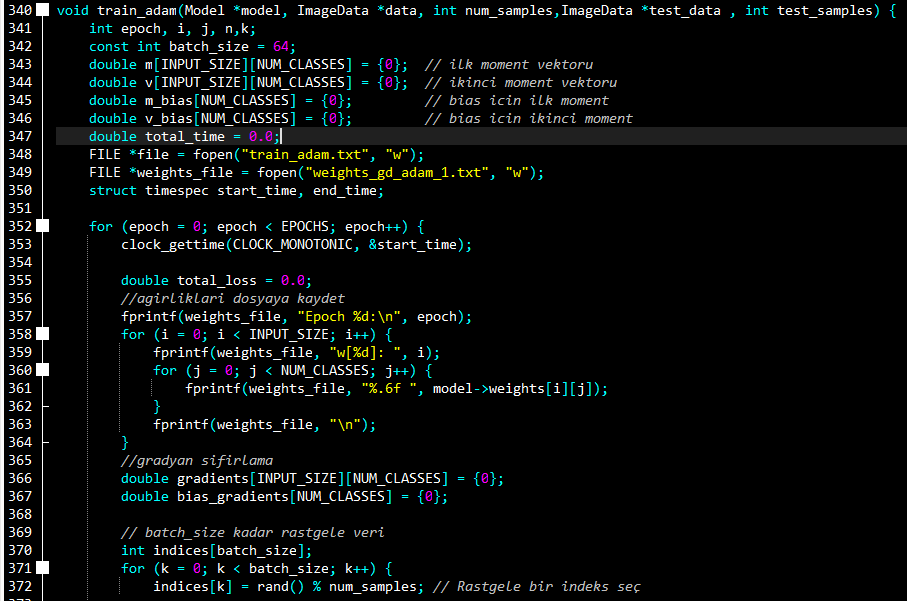
SGD’de GD’deki gibi bir epochta tüm veriyi işlemek yerine veri kümesinden her iterasyonda rastgele seçtiğimiz batch\_size boyutundaki veriyi kullanıyoruz.

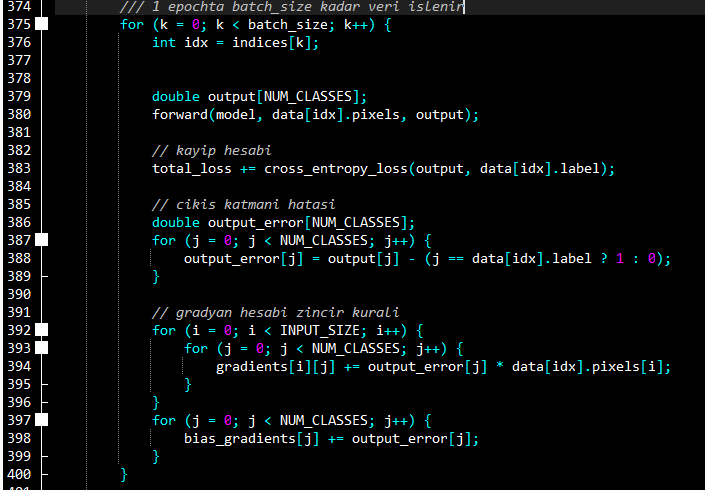


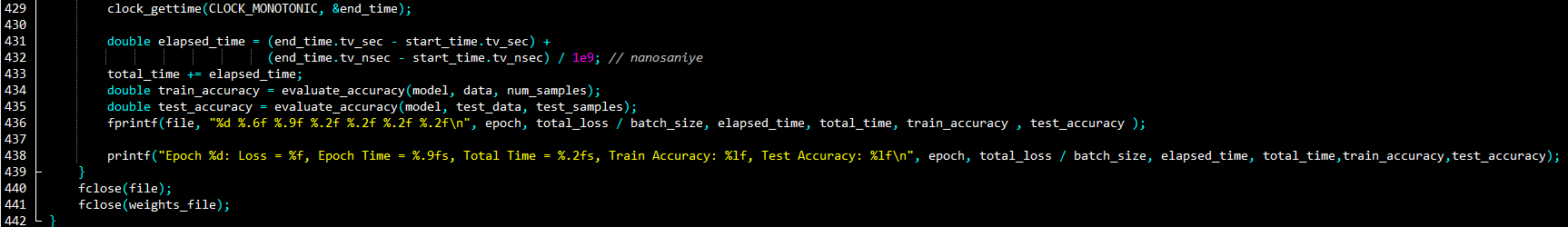
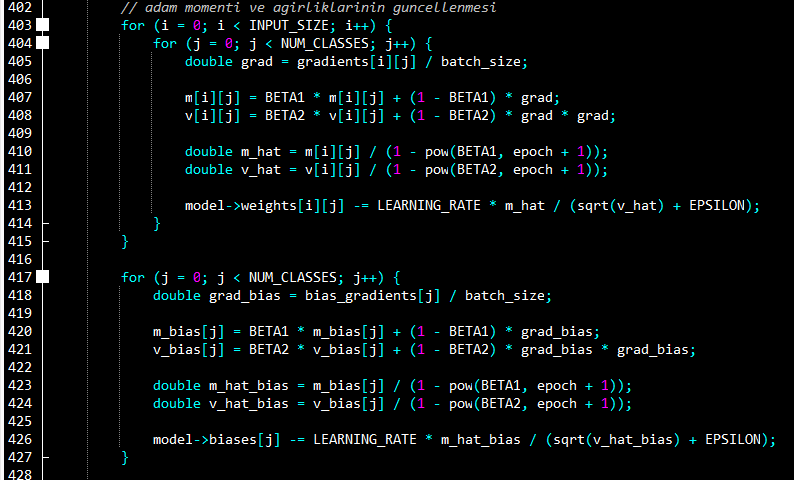


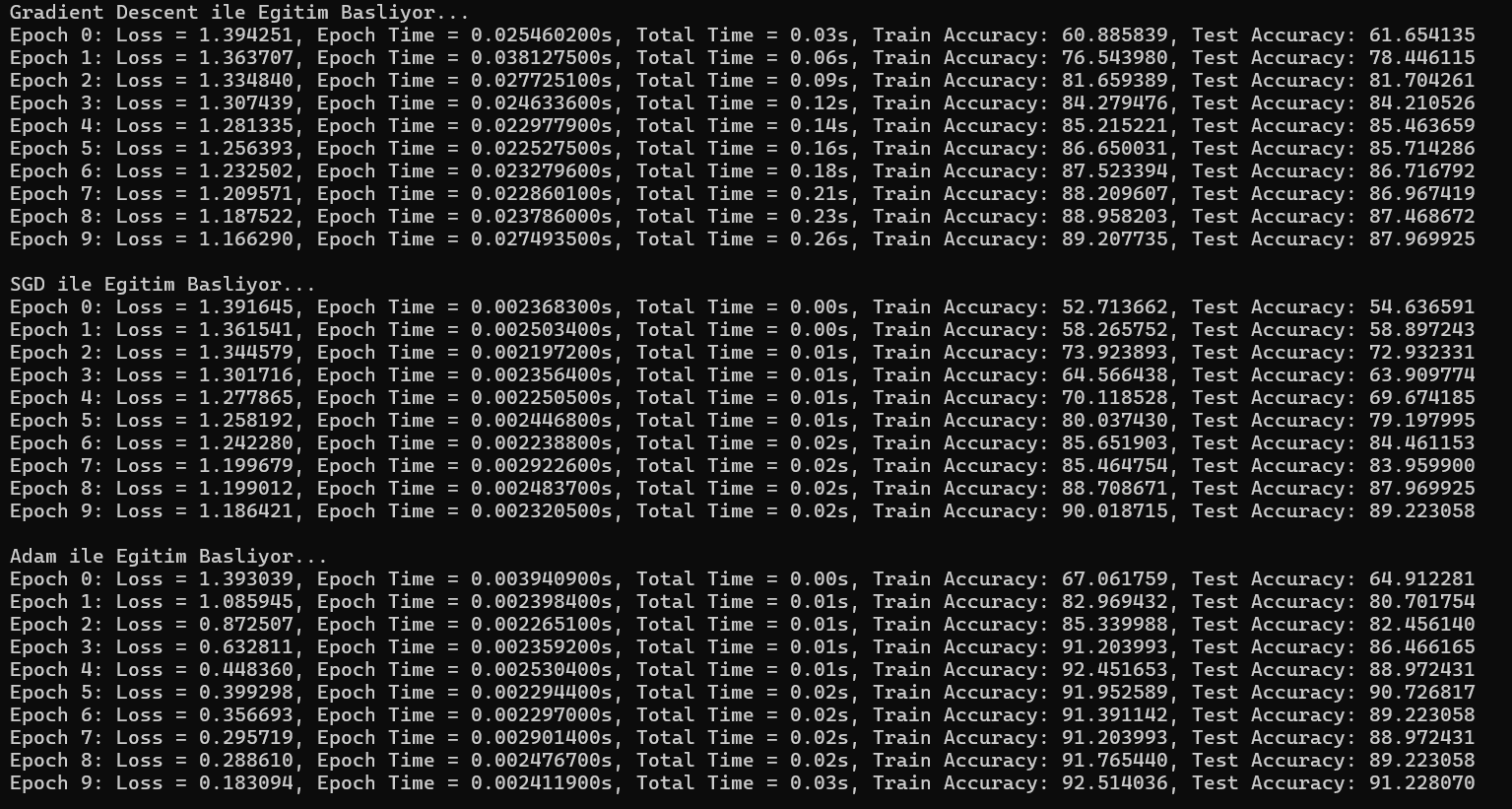
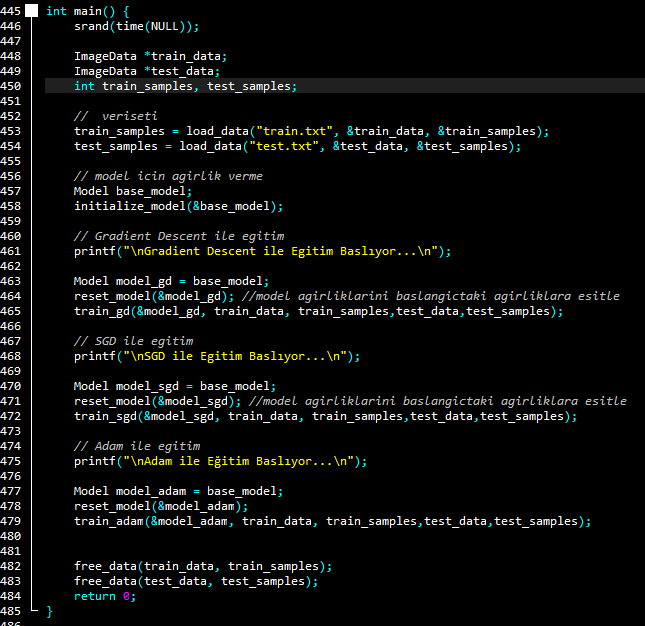
ADAM ile Eğitim:





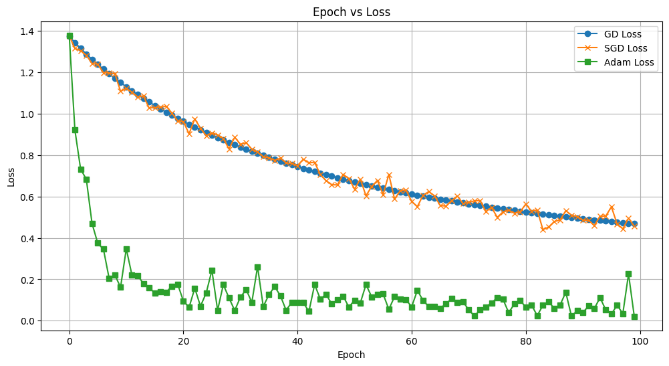




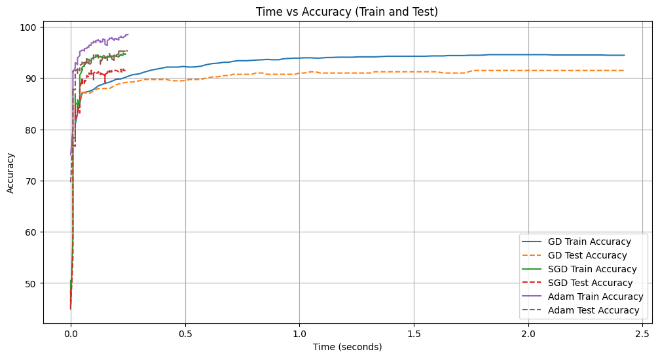
Main fonksiyon

Grafikler:

W1:

 öykü gelişim çizgisi; kumpas; grafiğini çıkarma, çizgi, diyagram, metin içeren bir resim

Açıklama otomatik olarak oluşturuldumetin, öykü gelişim çizgisi; kumpas; grafiğini çıkarma, çizgi, diyagram içeren bir resim

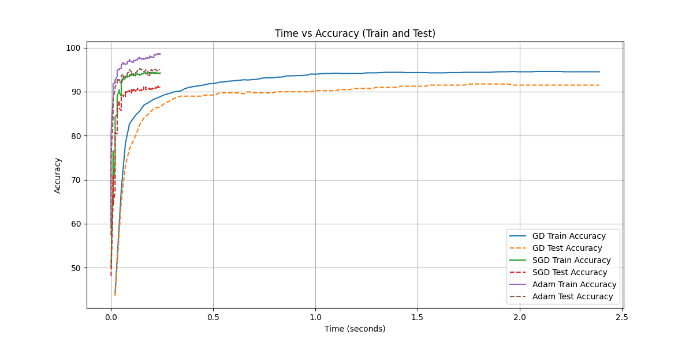
Açıklama otomatik olarak oluşturuldu

W2:

çizgi, metin, öykü gelişim çizgisi; kumpas; grafiğini çıkarma, diyagram içeren bir resim

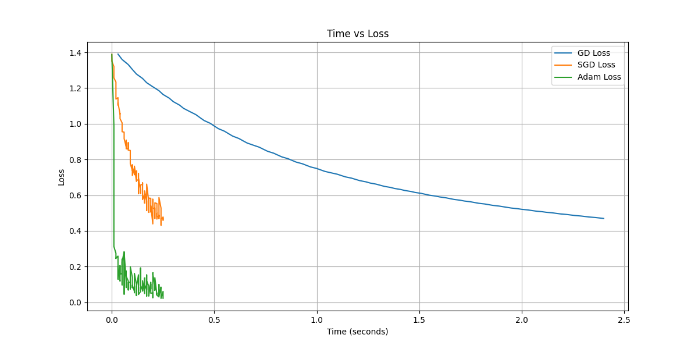
Açıklama otomatik olarak oluşturuldumetin, çizgi, öykü gelişim çizgisi; kumpas; grafiğini çıkarma, ekran görüntüsü içeren bir resim

Açıklama otomatik olarak oluşturuldumetin, ekran görüntüsü, çizgi, diyagram içeren bir resim

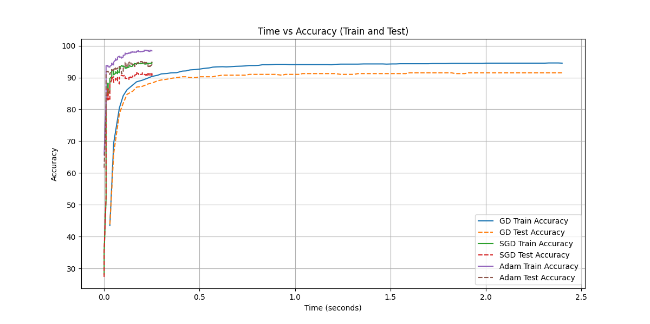
Açıklama otomatik olarak oluşturuldu

W3:

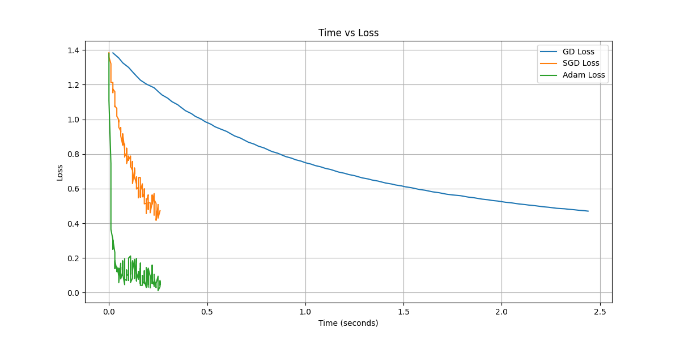
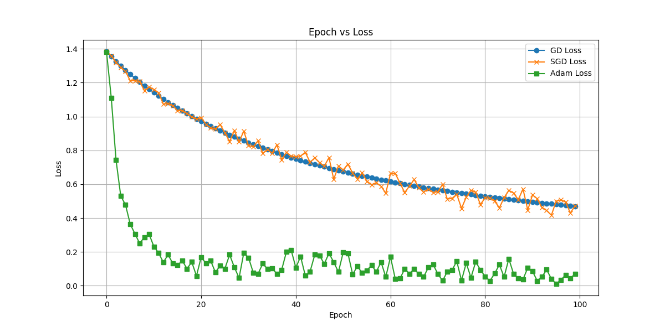
metin, çizgi, öykü gelişim çizgisi; kumpas; grafiğini çıkarma, diyagram içeren bir resim

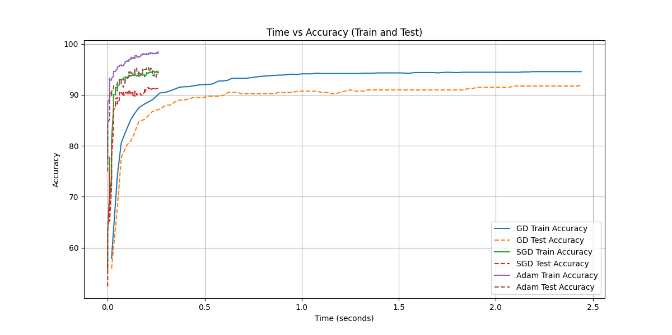
Açıklama otomatik olarak oluşturuldu

metin, ekran görüntüsü, çizgi, diyagram içeren bir resim

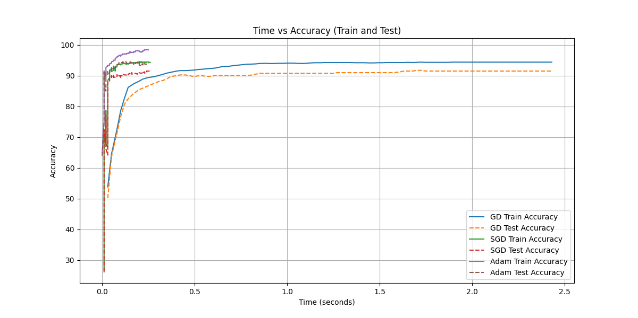
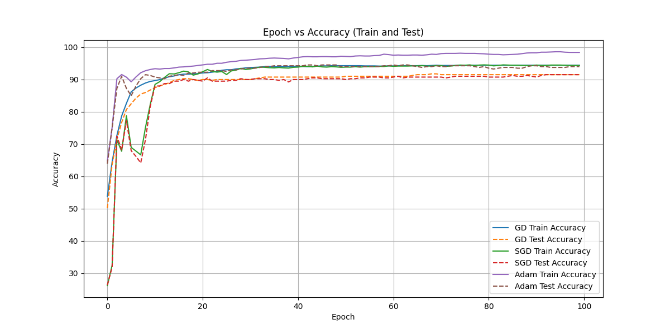
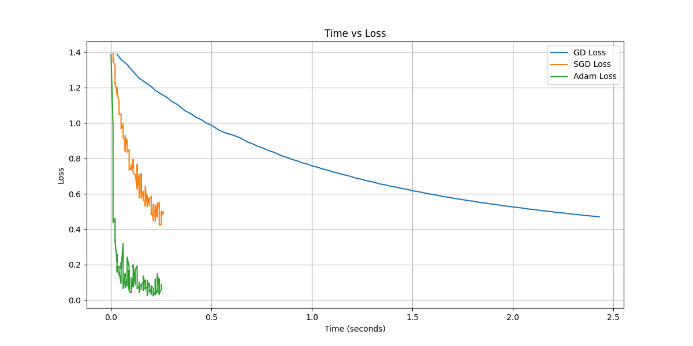
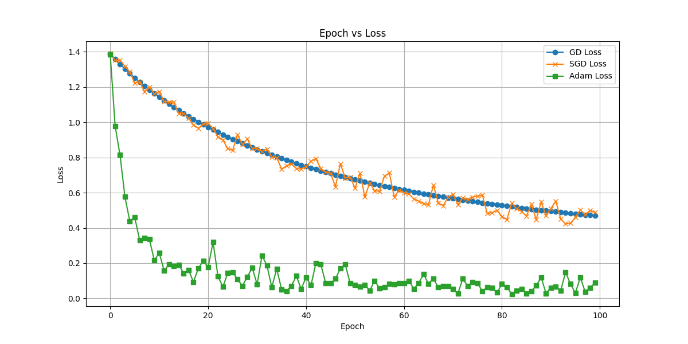
Açıklama otomatik olarak oluşturuldu

W4:

metin, ekran görüntüsü, diyagram, çizgi içeren bir resim

Açıklama otomatik olarak oluşturuldu

W5:



Grafiklerinden yola çıkarak bu değerlendirmeleri yaptım:  
GD, düzgün bir kayıp inişi gösterirken SGD ve ADAM daha düzensiz bir kayıp inişi sergiliyor. Bunun sebebi, SGD ve ADAM’ın her iterasyonda rastgele seçilmiş batch\_size boyutunda veri kullanarak gradyan hesaplaması yapması ve bu rastgele verilerin gradyan ortalamasını almasıdır. Rastgelelik, kayıp inişinde dalgalanmalara yol açar. GD ise tüm veri kümesinin gradyan ortalamasını hesaplar, bu nedenle daha düzenli bir kayıp inişi sağlar. Ancak GD, bir epoch sırasında tüm veri kümesini kullanır, bu yüzden SGD ve ADAM’a göre epoch süreleri daha uzundur.

ADAM, SGD ve GD’den farklı olarak momentum ve adaptif öğrenme oranlarını birleştirir. Momentum, gradyan yönünü stabilize ederken adaptif öğrenme oranları, parametreler için bireysel öğrenme hızları ayarlayarak optimizasyonu daha verimli hale getirir. Bu özellikler, ADAM’ın SGD ve GD’ye kıyasla daha düşük loss ve daha yüksek accuracy değerlerine ulaşmasını ve daha iyi genelleme yeteneğine sahip olabilmesini sağlar

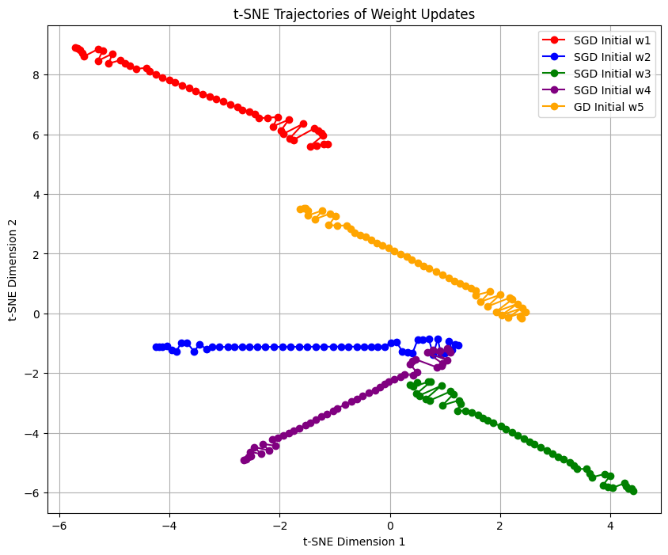
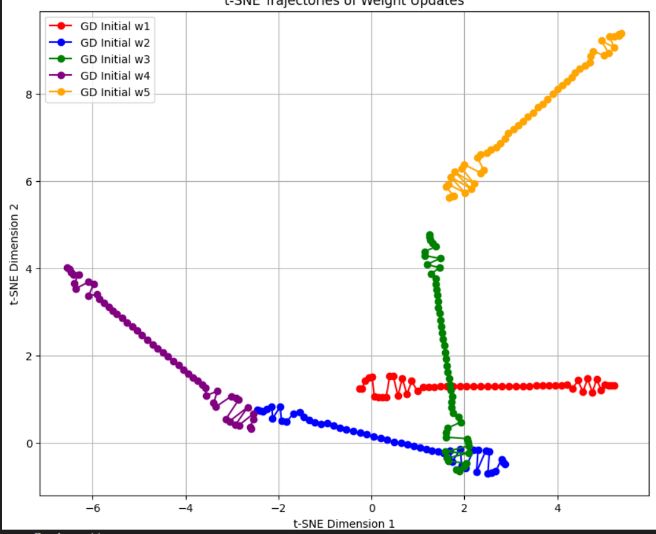
Grafik Çizdirme Kodları (Python):

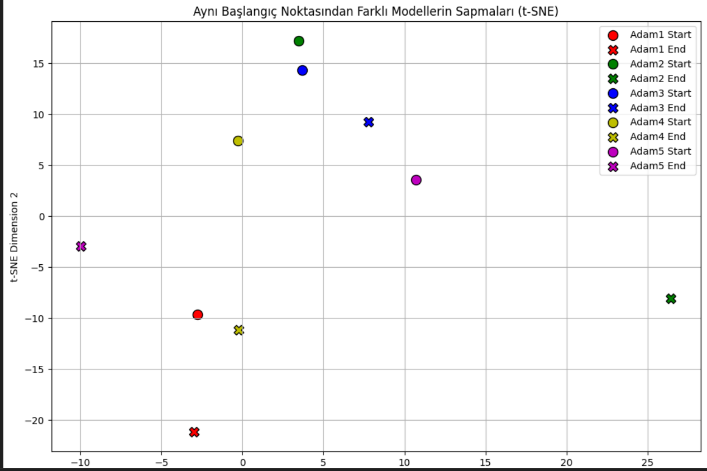
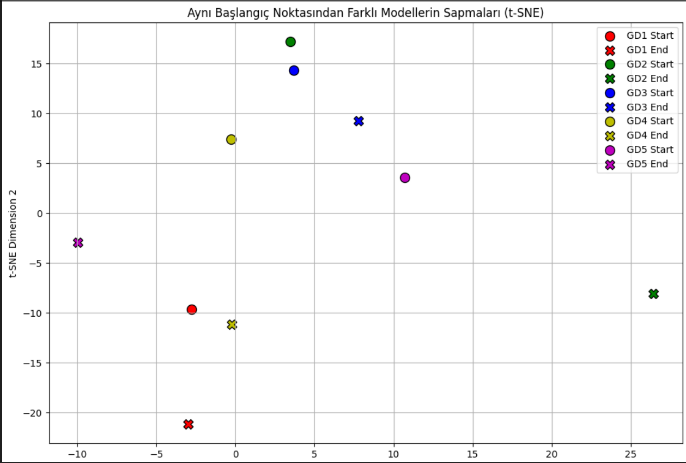
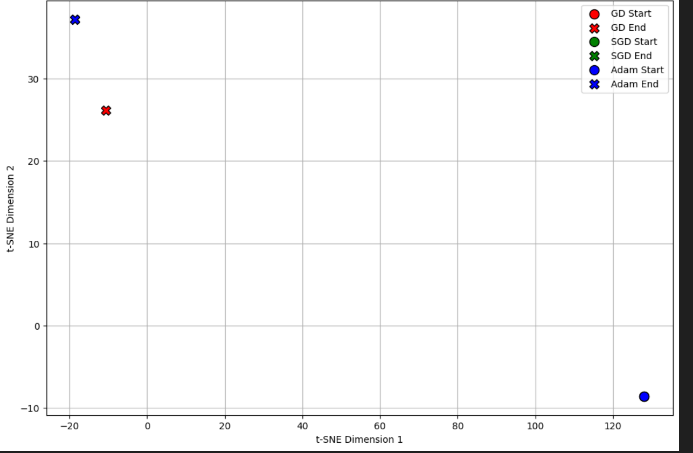
metin, ekran görüntüsü içeren bir resim

Açıklama otomatik olarak oluşturuldumetin, ekran görüntüsü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu

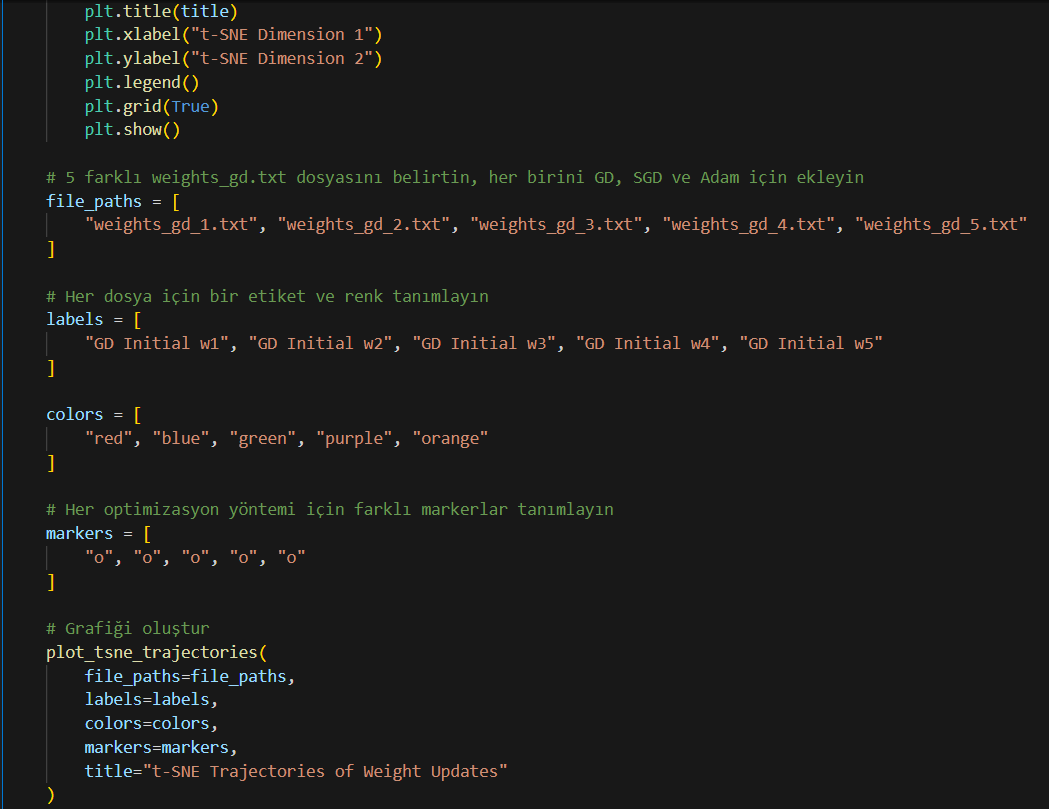
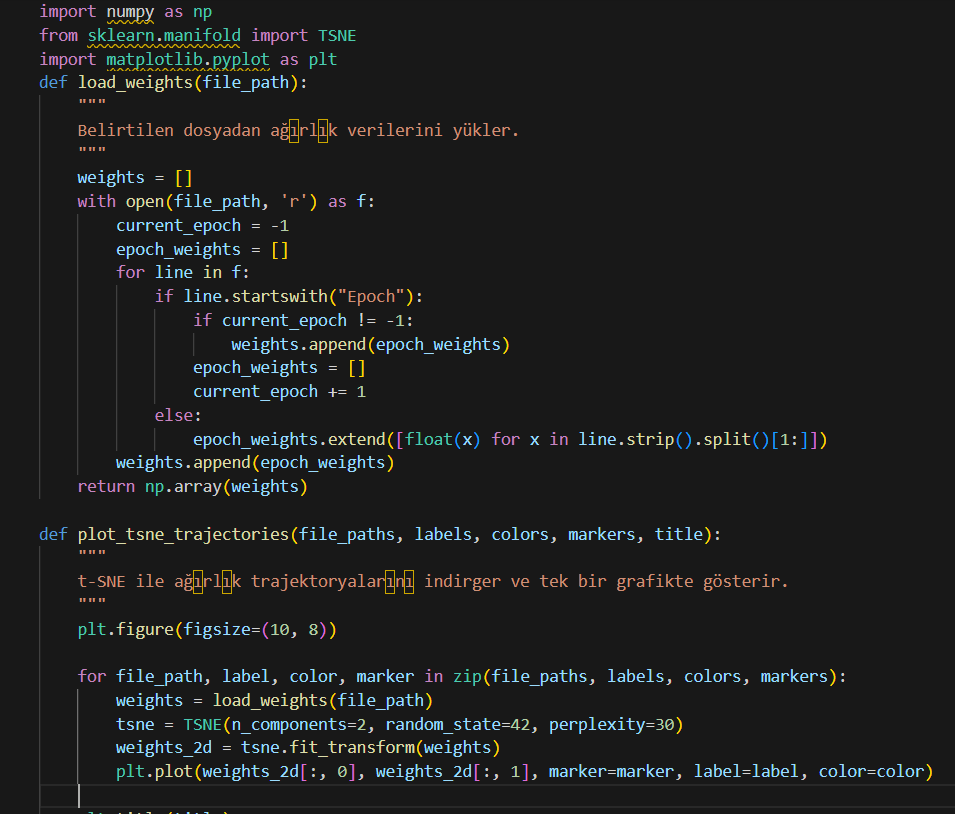
B: Optimizasyon sürecinin 2 boyutta gösterimi:

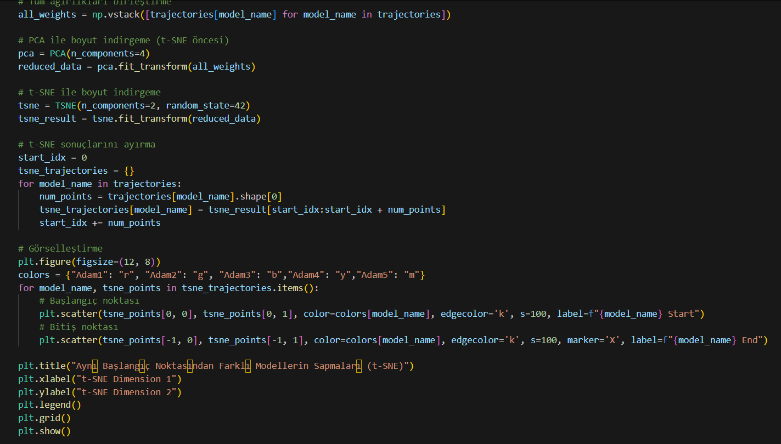
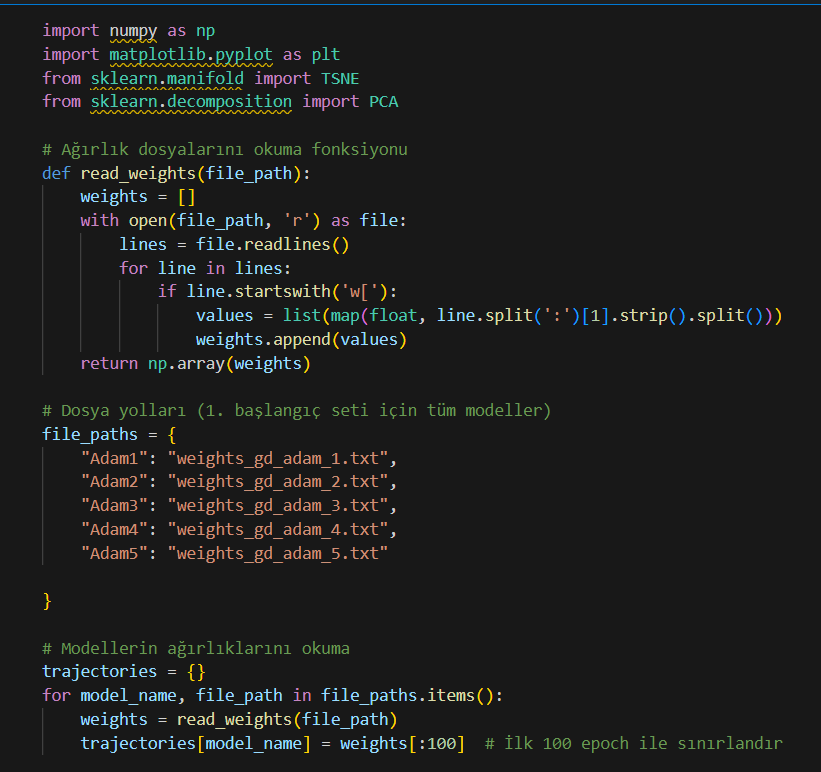
metin, öykü gelişim çizgisi; kumpas; grafiğini çıkarma, ekran görüntüsü, diyagram içeren bir resim

Açıklama otomatik olarak oluşturuldu

Bu T-SNE grafiklerinde aynı modelin 5 farklı başlangıç ağırlıklarıyla (W) eğitildiğinde farklı local minimum değerlerine yakınsadığı gözlemlenir. Aynı zamanda aynı başlangıç ağırlıklarıyla farklı modeller eğitildiğinde farklı modellerin hemen hemen aynı local minimuma yakınsadığı gözlemlenir

T-SNE kodu:





T-SNE grafiklerini bu kodlarla oluşturdum. Her grafik için aynı kodları farklı modeller veya ağırlıklar için çalıştırdım.

Kaynakça:

T-SNE:

[https://www.researchgate.net/figure/a-tSGD-training-Dotted-lines-illustrate-the\_fig1\_366247247](https://www.researchgate.net/figure/a-t-SNE-plot-of-the-weight-space-during-SGD-training-Dotted-lines-illustrate-the_fig1_366247247)

Softmax ve Gradyan Hesabı:

<https://ogunlao.github.io/2020/04/26/you_dont_really_know_softmax.html#numerical-stability-of-softmax>

<https://www.youtube.com/watch?v=LLux1SW--oM>

<https://www.youtube.com/watch?v=ueO_Ph0Pyqk>

<https://madalinabuzau.github.io/2016/11/29/gradient-descent-on-a-softmax-cross-entropy-cost-function.html>

<https://medium.com/@shredder67/softmax-for-noobies-how-we-train-models-for-multi-class-classification-with-some-math-e53d1d413be4>

GD , SGD ve ADAM:

<https://sites.google.com/view/mfatihamasyali/optimization-techniques>