

Genel Talimatlar

İlgili python betiğine ekli proje PDF dosyanızı göndermek için lütfen aşağıda verilen talimatları izleyin.

Öğrencilerin bir Google Colab not defteri ile çalışmak, çıktılarını PDF dosyası olarak kaydetmek ve belirlenen Learn ögesine göndermek için izleyebilecekleri adım adım bir kılavuz. Bu talimatları doğrudan öğrencilerinizle paylaşabilirsiniz:

Google Colab ile Çalışmak ve Çıktıları Göndermek için Adım Adım Talimatlar Adım 1:

Google Colab'da .ipynb Dosyasını Açın

1. Sağlanan .ipynb dosyasını Learn'den indirin.
2. Google Colab'ı ziyaret ederek [Google Colab](https://colab.research.google.com/)'ı açın.
3. Colab'da:
 - Dosya > Not Defterini Aç'a tıklayın.
 - Yükle sekmesine gidin.
 - Dosya Seç'e tıklayın ve indirdiğiniz .ipynb dosyasını yükleyin.

Adım 2: Not Defteri Üzerinde Çalışın

1. Talimatları izleyin ve not defterindeki gerekli tüm görevleri tamamlayın.
2. Her bir hücrenin yanındaki oynat düğmesine (▶) tıklayarak veya Shift + Enter tuşlarına basarak çalıştırın.
3. Dosya > Kaydet'i seçerek veya kaydet simgesine tıklayarak ilerleme durumunuzu sık sık kaydedin.

Adım 3: Not Defterini PDF Olarak Kaydetme

1. Tüm görevleri tamamladıktan sonra not defterini PDF'ye aktarın. ◦ Dosya > Yazdır'a tıklayın.
- Yazdır iletişim kutusunda, hedef yazıcıyı PDF olarak kaydet olarak değiştirin.
- Tüm içeriğin düzgün bir şekilde sığığından emin olmak için gerekirse ayarları (örneğin, yatay/dikey, kenar boşlukları) yapın. ◦ Kaydet'e tıklayın ve PDF dosyasını saklamak için bilgisayarınızda bir konum seçin.

Alternatif Yöntem:

Doğrudan "PDF'ye Yazdır" yöntemi çalışmazsa veya sorunlarla

karşılaşırsanız: • Not defterini HTML olarak indirin:

- Dosya > İndir > HTML olarak indir (.html).
- HTML dosyasını bir tarayıcıda açın ve PDF olarak

yazdırın. Adım 4: PDF'yi Doğrulayın

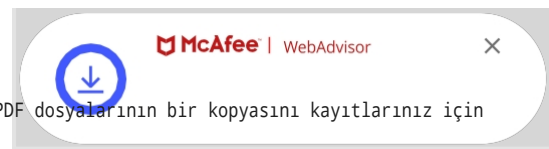
1. Kaydedilen PDF'yi açın ve kontrol edin:
 - Tüm hücreler, çıktılar ve grafikler dahildir.
 - İçerik açıkça görülebiliyor ve doğru şekilde biçimlendirilmiş.
2. Herhangi bir içerik eksik veya belirsizse, not defterini tekrar gözden geçirin ve tekrar dışa aktarın.

Adım 5: PDF'yi Learn'e gönderin

1. Learn platformunuzda oturum açın.
2. İlgili gönderim ögesine gidin.
3. Ödevi Gönder veya eşdeğer bir seçeneğe tıklayın.
4. PDF dosyanızı yükleyin.
5. Ekrandaki talimatları izleyerek gönderiminizi onaylayın.

Adım 6: Çalışmanızı Saklayın

Daha sonra yeniden göndermeniz veya gözden geçirmeniz gerekmesi ihtimaline karşı hem .ipynb hem de PDF dosyalarının bir kopyasını kayıtlarınız için saklayın.



Yaygın Sorun Giderme İpuçları

- Google Colab Zaman Aşımı: Google Colab hareketsizlik nedeniyle zaman aşımına uğrayabilir. İlerlemenizi kaydedip sık kaydedin ve yeniden çalıştırarak Gerekirse yaş.
- Dosya Boyutu: PDF dosyanızın yüklenemeyecek kadar büyük olmadığından emin olun. Gerekirse ücretsiz çevrimiçi araçları kullanarak sıkıştırın es s a ^{Your} ^{download'd's being taranmıştır.}
- Destek ile İletişime Geçin: Teknik sorunlarla karşılaşırsanız, derhal eğitmeninize veya destek ekibine ulaşın. Bir sorun olursa size haber verimiz.

□ Gerekli Paketler:

Ana kütüphaneleri kullanma

Yaygın olarak kullanılan özel bir paketimiz var

İşte veri bilimi için en yaygın kullanılan beş Python kütüphanesi:

1. NumPy

- **Amaç:** Özellikle büyük çok boyutlu dizileri ve matrisleri işlemek için verimli sayısal hesaplamalar. Bu diziler üzerinde işlem yapmak için matematiksel fonksiyonlar sağlar.
- **Kullanım Örnekleri:** Veri manipülasyonu, lineer cebir, Fourier dönüşümleri, rastgele sayı üretimi.

2. Pandalar

- **Amaç:** Özellikle yapılandırılmış veriler için veri manipülasyonu ve analizi. Tablo halindeki verilerle (örn. CSV, Excel) iyi çalışır.
- **Kullanım Örnekleri:** Veri temizleme, filtreleme, grupta ve veri kümelerini birleştirme.

3. Matplotlib

- **Amaç:** Statik, animasyonlu ve etkileşimli grafikler aracılığıyla veri görselleştirme.
- **Kullanım Örnekleri:** Çizgi grafikler, çubuk grafikler, dağılım grafikleri ve histogramlar oluşturma.

4. Seaborn

- **Amaç:** matplotlib üzerinde daha yapısal Veri görselleştirme
- **Kullanım Örnekleri:** Çizgi grafikler, çubuk grafikler, dağılım grafikleri ve histogramlar oluşturma.

Bu kütüphaneler veri bilimi için temeldir ve genellikle tüm veri analizi ve makine öğrenimi hattını kapsayacak şekilde birlikte kullanılır.

PyTorch Kullanımı

1. PyTorch

- **Amaç:** Sinir ağları oluşturmak ve eğitmek için derin öğrenme çerçeveleri.
- **Kullanım Örnekleri:** Bilgisayarla görme, doğal dil işleme, takviyeli öğrenme ve diğer derin öğrenme uygulamaları.

```
import numpy as np          # Sayısal hesaplamalar için NumPy'yi içe aktar
pandas'ı pd olarak içe aktar # Veri manipülasyonu için Pandas'ı içe aktarın
import matplotlib.pyplot as plt # Çizim için Matplotlib'i içe aktarın
import seaborn as sns # Gerekirse matplotlib'in ötesinde çizim için Seaborn'u içe aktarın from
sklearn import datasets # Makine öğrenimi için scikit-learn'i içe aktarın
```

Alıştırma 0 (15 puan): Ana kavramlar hakkında bilgi

Sinir ağı kavramı ile uyumlu olarak verilen teknik kelimeleri açık ve kısa bir şekilde tanımlayın;•

Öğrenme oranı

- Etkinleştirme işlevi
- Gizli katman ve gizli birimler

□ Çözüm

Alıştırma 1 (10 puan): Tensör İşlemleri

Hedef: Tensör oluşturma ve manipüle etme pratiği.

1 ile 10 arasında rastgele tam sayılarla dolu (3, 3, 3) şeklinde bir 3B tensör

oluşturun. Aşağıdaki işlemleri gerçekleştirin:

- İlk iki boyuttan bir 2D matris çıkarmak için tensörü dilimleyin. •

Tensörü 1 boyutlu bir vektöre yeniden şekillendirin.

Aynı boyutta rastgele oluşturulmuş başka bir tensör ile eleman bazında toplama işlemi gerçekleştirin.

Alıştırma 2 (10 pts): Doğrusal Dönüşüm

Hedef: Temel matris çarpımını uygulamak ve sinir ağlarındaki rolünü anlamak.

Rastgele sayılarla dolu (3, 4) şeklinde bir A tensörü ve (4, 2) şeklinde başka bir B tensörü oluşturun.

- C çıkış tensörünü üretmek için A ve B arasında bir matris çarpımı gerçekleştirin.
- Yayın kullanarak C'ye bir sapma terimi ekleyin.

Çözüm

Çözüm için İlgili Kod

Alıştırma 3 (15 pts): Aktivasyon Fonksiyonları

Hedef: Yaygın aktivasyon fonksiyonlarını uygulamak ve görselleştirmek.

torch.linspace fonksiyonunu kullanarak -5 ila 5 arasında bir girdi

aralığı oluşturun. Aşağıdaki aktivasyon fonksiyonlarını PyTorch

fonksiyonları olarak uygulayın:

- Sigmoid
- ReLU
- Tanh

Matplotlib kullanarak giriş aralığı için bu aktivasyon fonksiyonlarının çıkışlarını çizin.

Çözüm

Çözüm için İlgili Kod

PyTorch ile Sinir Ağları Oluşturma ve Eğitim

Bu bölümde, aşağıdaki adımları izleyerek PyTorch kullanarak bir sinir ağı uygulayacağız:

- **Adım 1: Sinir Ağı Sınıfını Tanımlayın**
- **Adım 2: Verileri Hazırlayın**
- **Adım 3: Modeli, Kayıp Fonksiyonunu ve Optimize Ediciyi Oluşturun**
- **Adım 4: Modeli Eğitin**
- **Adım 5: Modelin Test Edilmesi**

Alıştırma 4 (10 puan): Sinir Ağının Tanımlanması

Hedef: Verilen yapıdan NN sınıfınızı oluşturun.

Verilen görselleştirmeyi kullanın ve daha önce yaptığımıza benzer şekilde (gerekli tüm modüller ve kütüphaneler dahil) Sinir Ağı (NN) Sınıfınızı bir python kodu olarak yazmak için aşağıdaki ayrıntıları varsayın. Bir ön varsayım olarak, Sigmoid fonksiyonu aktivasyon olarak her bir uzay için kullanılır.

•

Çözüm



McAfee | WebAdvisor

İndirme işleminiz taranıyor. Bir sorun olursa size haber vereceğiz.

×

FEATURES

Which properties do you want to feed in?



+ - 2 HIDDEN LAYERS

+ -
3 neurons

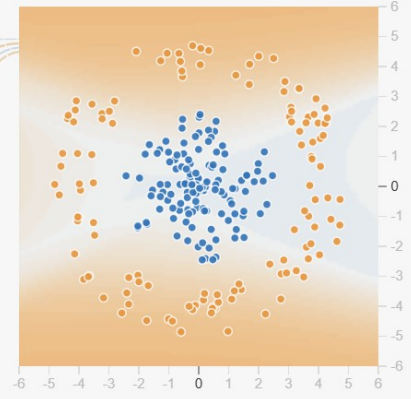
+ -
4 neurons

The outputs are mixed with varying weights, shown by the thickness of the lines.

This is the output from one neuron. Hover to see it larger.

OUTPUT

Test loss 0.392
Training loss 0.407



Colors shows data, neuron and weight values.

- `nn.Sequential()` kuralını kullanmadan sınıf işlevini yazın Bunun yerine
- `nn.Sequential()` işlevini kullanarak benzer bir sınıf oluşturun.

Çözüm

İlgili kod

Alıştırma 5 (10 puan): Tanımlanan NN'nin eğitilmesi

Hedef: Modeli eğitim seti üzerinde eğitmek için simüle edilmiş bir veri seti oluşturmak

- İlk olarak, tanımladığınız NN sınıfına dayalı olarak tensör olarak uygun bir rastgele veri oluşturun (`nn.Sequential()` kullanılmadığı durum konvansiyonu). İlk olarak `X_train` ve `y_train` oluşturmaya odaklanın.

İpucu: Örneklem büyüklüğünü 700 olarak kullanarak hem girdi hem de yanıt verileriniz için `torch.rand` işlevinden yararlanabilirsiniz.

- İlgili ara adımları ekleyerek NN sınıf modelinizi eğitin
 - Modeli Oluşturun, Kayıp Fonksiyonunu ve Optimize Ediciyi Tanımlayın. Problemin regresyon olduğunu varsayarak veri setinize dayalı herhangi bir uygun kayıp fonksiyonunu kullanabilirsiniz. Dikkate alınan optimize edici için varsayılan parametreleri de kullanın
 - Modeli 10 epok kullanarak eğitin
 - Eğitim verileri için tahminlerinizi alın ve hatalarınızı hesaplamak için orijinal yanıtla karşılaştırın

Çözüm

İlgili kod

Alıştırma 6 (10 puan): Yeni bir veri seti üzerinde performans testi

Hedef: Şimdi yeni bir veri üzerindeki performansı ölçmek için eğitim verisine benzer yeni bir test verisi oluşturmanız gerekir

Bu amaçla, öncelikle veri setinizi 300 boyutunda oluşturmanız gerekir• `X_test` ve

`y_test`'i oluşturun

- Orijinal `y_test` değerleriyle karşılaştırmak için şimdi test verileri üzerinde eğitilmiş modelin tahminlerini alın



McAfee | WebAdvisor



İndirme işleminiz taranıyor. Bir sorun olursa size haber vereceğiz.

İlgili kod

Alıştırma 7 (20 puan): Epok ve öğrenme hızının etkisi

Hedef: Yukarıdaki eğitim prosedürüne bakarak, farklı parametrelerin etkisini inceleyin

- Şimdi epok sayısını 100, 500 olarak iki farklı durum olarak düşünün
- Öğrenme oranı parametresi η için varsayılan parametre yerine alternatif değerler olarak $\eta=0.1$, $\eta=0.01$ değerlerini kullanın

Yukarıdaki parametre ayarlarını değiştirerek yaptığınız deneye dayanarak, bunların yukarıda oluşturulan modelinize dayalı **eğitim** ve **test** kaybı hesaplaması üzerindeki etkilerini tartışın (Alıştırma 5'te oluşturduğunuz aynı eğitime verileri üzerinden)

☐ Çözüm

İlgili kod

- Yukarıdaki duruma benzer şekilde hatanızı tekrar hesaplayın.

☐ Çözüm



McAfee | WebAdvisor

×

İndirme işleminiz taranıyor. Bir sorun olursa size haber vereceğiz.