

- Google Colab çalışma limitleri

Google Colab'da T4 GPU çalışma alanını seçtiğinizde, Tesla T4 GPU'ya erişim sağlarsınız. Bu GPU, 16GB GDDR6 bellek ve 2.560 CUDA çekirdeği ile donatılmıştır ve derin öğrenme görevleri için uygundur ([DrLee](#)) ([Brev](#)).

Ancak, Google Colab'ın T4 GPU (ve diğer kaynaklar) kullanım sınırları dinamiktir ve çeşitli faktörlere bağlı olarak değişebilir. Genel olarak, ücretsiz kullanıcılar için Colab şunları sağlar:

1. **Sınırlı Oturum Süresi:** Bir oturumun maksimum çalışma süresi genellikle 12 saat civarındadır, ancak yüksek kullanım algılanırsa veya sistem yoğun olduğunda daha kısa olabilir.
2. **Boşta Kalma Zaman Aşımı:** Oturumunuz belirli bir süre (genellikle 90 dakika) boşta kalırsa, kaynakları serbest bırakmak için sonlandırılır.
3. **Kaynak Erişilebilirliği:** T4 GPU'ya erişim garanti edilmez ve dalgalanabilir. Yüksek talep nedeniyle GPU'nun mevcut olmadığı zamanlar olabilir.

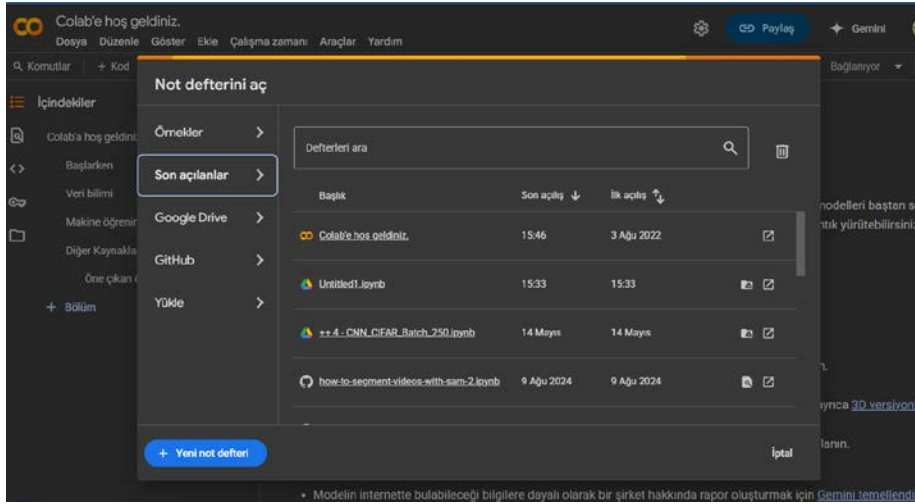
Daha tutarlı erişim ve daha az kısıtlama için Colab Pro veya Colab Pro+ yükseltmesi yapmayı düşünebilirsiniz. Colab Pro, daha uzun çalışma süreleri, daha hızlı GPU'lar, daha fazla RAM ve etkinlik nedeniyle daha az kesinti gibi ek özellikler sunar ([Novita AI](#)) ([Brev](#)).

Özetle, Google Colab, T4 GPU da dahil olmak üzere ücretsiz olarak güçlü hesaplama kaynakları sağlar, ancak oturum süresi ve erişilebilirlik konusunda bazı sınırlamalar vardır. Yoğun ve uzun süreli görevler için Colab Pro'yu düşünmek faydalı olabilir.

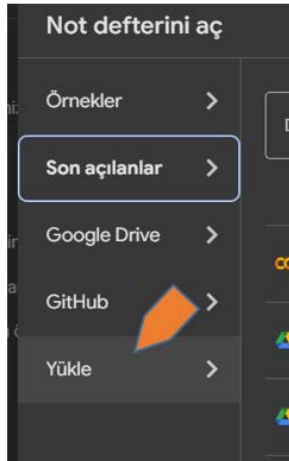
COLAB İŞLEM Sıralaması

- Paylaşılan NB un Colab de açılması için altta gerekenler yaopılacaktır..

1. <https://colab.research.google.com/> adresine giriş yapılır



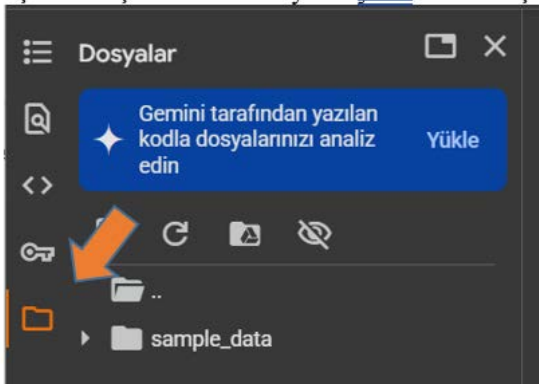
2. Üstteki resimde görülen açılış penceresindeki Yükle seçeneğine girilir. Oradan da Göz At seçeneğine tıklanarak dersten önce öğrencilerle paylaşılan Notebook un olduğu klasöre gidilip çalıştırılacak NB seçilir.



** Yapılan hataların çoğusu ilgili dosya klasörünün yeri ve bu yolun tanıtılması ile sorunlardan kaynaklanmaktadır.

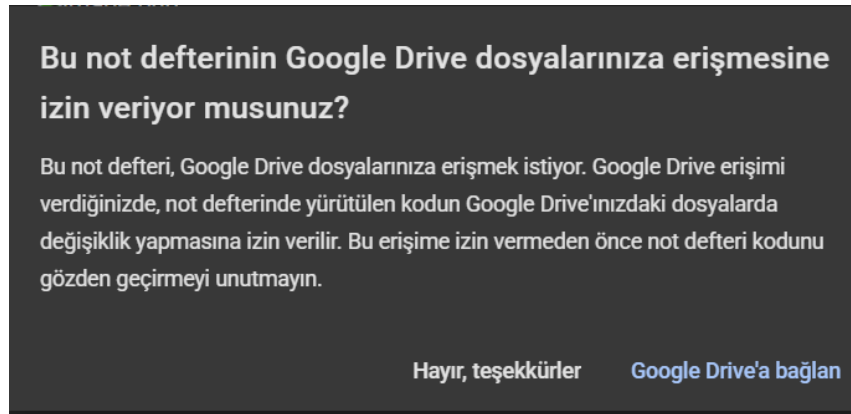
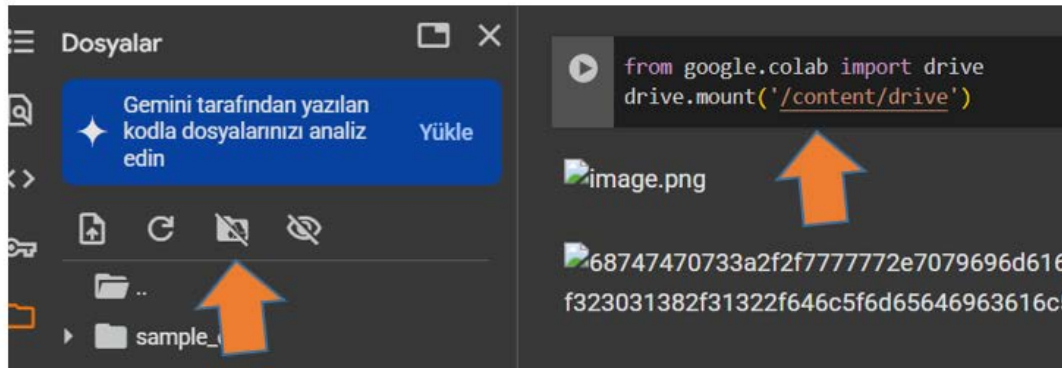
Local deki NB dosyayı çalıştırırken içindeki resimleri göstermiyor. Çünkü Colab de aynı klasörde bu resimler bulunmuyor.

3. Colab da GPU hizmetini kullanmak için geldik buraya, o izini vermelimiz. iki şey yapılacak bir izin vermek diğeri GPU yu aktive etmek. Resimdeki solda klasör resimi olan yere tıklayınca Google in sanal olarka kullandığı yeri gösteriyor. İlgili kısmın açılması için biraz bekleyiniz yada tekrar tuşa tıklayınız.





4. Resimde solda görülen “Drive Ekle” tuşuna tıklayınca otomatik olarak NB a sağda gösterilen kod bloğunu ekler. Bu kod bloğu bizim için Connection sağlayacak. bunu çalıştırınca GDrive a bağlanma yetkisi verilmiş oluyor. bu işme başarılı olunca soldaki Dosyalar kısmında ilgili klasörlerin gelmiş olması gerekir. Tüm izinleri vermeniz gerekir. Sonrasında code altında “Mounted at /content/drive” yazısı görünür. Cell_images klasörü de bu dosyalar içinde olmalı.

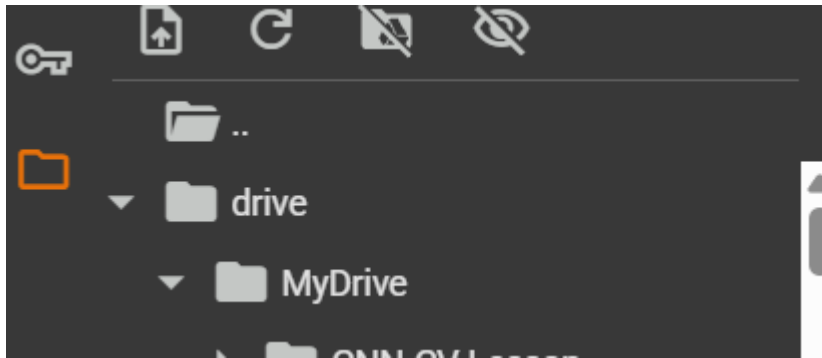
sonra alt resimdeki kısımda “Google Drive a bağlan” tıklayın ve ilgili gmail hesabınızı seçin.



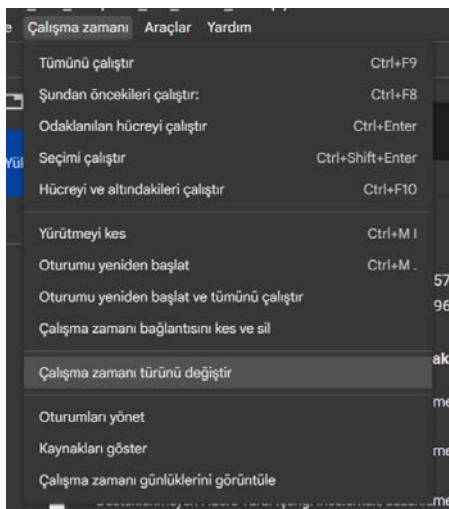
Gelen seçeneklerin Tümünü seçin. Eğer yetkilendirme işlemi başarıyla sağlandı ise alttaki gibi kod satırının altında “Mounted at /content/drive” yazacaktır.

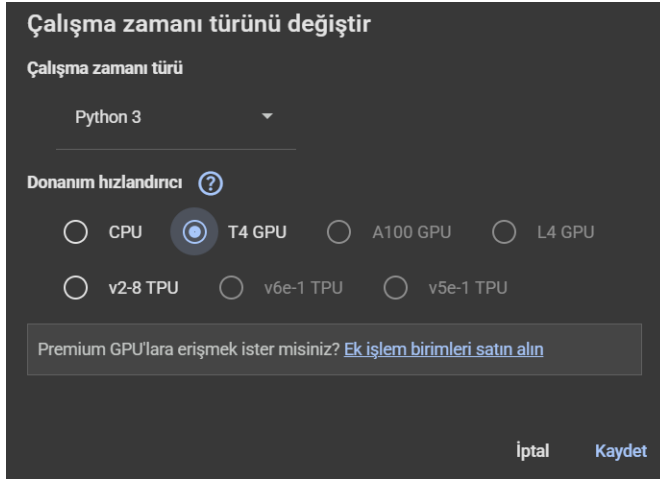
```
1 ✓  
1.  from google.colab import drive  
drive.mount('/content/drive')  
  
 Mounted at /content/drive
```

5. Şimdi sol tarafta resimdeki gibi klör kısmında Drive seçeneği eklendi. Biraz beklemeniz gerekebilir ilk etapta. Yada Klör tuşuna birkaç kez basabilirsiniz. Drive → MyDrive → Böylece dosyalar görünmüş oluyor. Burası bizim Google Drive daki klasörlerimiz. Buradaki klasörler işlemler yaparken bazen kesinti olup gidebiliyor, o zaman üstteki kodu tekrar çalıştırıp tekrar bağlantı yapılabilir. Yada komple NB baştan ilk maddeden itibaren yeniden çalıştırılabilir.

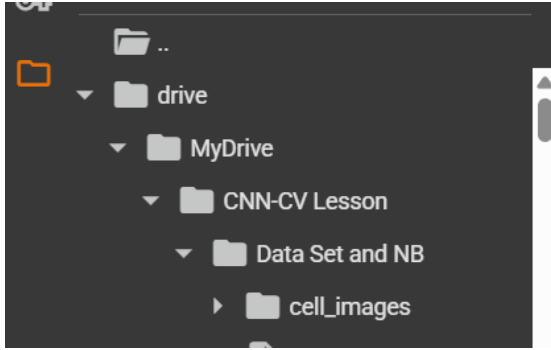


6. İlk işlem yetkilendirme idi ve onu yukarda yapmış olduk. İkinci işlem ise GPU seçimi.. üstteki menülerinden “Çalışma Zamanı” menüsüne girip oradan “Çalışma zamanı türünü değiştir” kısmına girmek ve orada T4 GPU seçeneğini seçmek. sağ üstte bağlanıyor diyecek ve T4 yazısı gelmiş olacak. Bu işlemi en baştan yapmayınca önceki işlemlerinizi sonlanabiliyor, o yüzden burada en başta yapmış olduk.
NOT: BU bağlantıyı seçince klör yapısında soldaki “drive” seçeneği kaybolabilir. O zaman Yetkilendirme kodunu tekrar çalıştırın veya sayfayı yenileyip yeniden bağlanarak tekrar yetkilendirin

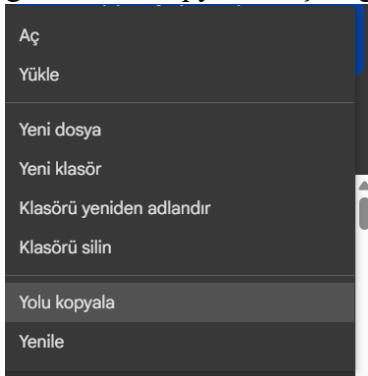




7. NB daki Kütüphaneleri import eden kısmı çalıştırın, pwd kısmını çalıştırın. Üçüncü olarak ise NB daki my_data="cell_images" dediğimiz yerdeki yolu değiştirmemiz lazım. Gdrive daki solda ilgili dosyanın yüklü olduğu yere geldiğimizde sağ tuş yapıp yolu göster diyeceğiz ve bunu kod satırına yeniden yapıştırmış olacağız. şimdi değişmiş olacak.. Bunun için tabiki cell_image klasörünün Google Drive a önceden eklenmiş olması gerek. Resimde benim eklediğim kısım göeülüyor.



Resimde görülen cell image klasör isminin üzerine gelip 3 noktaya tıklayınca resimdeki gibi "Yolu kopyala" seçeneği çıkacak ve oraya tıklayacağız.



Şimdi bu yolu Colab de çalıştırdığımız klasör olarak vereceğiz.

Öncesi:

```
my_data= 'cell_images' # pwd ile belirtilen klasördeki görüntü dosyasına bağlanır.  
# rar dosyamızın adı- NB ile aynı klasörde olmalı
```

Sonrası:

```
my_data= '/content/drive/MyDrive/CNN-CV Lesson/Data Set and NB/cell_images' # pwd ile belirtilen klasördeki  
# rar dosyamızın adı- NB ile aynı klasörde olmalı
```

my_data= '/content/drive/MyDrive/CNN-CV Lesson/Data Set and NB/cell_images'
şeklinde olacak. Artık dosyalar tanımlanmış oldu.

NOT: Dosyanın yerini göstermede genelde hata alınıyor. buraya dikkat edelim. Gerisi zaten aynı şekilde kodların çalıştırılması şeklinde devam edecek.

Şimdi alttaki "os.listdir(my_data)" kodunu çalıştırınca train-test klasörlerini resimdeki gibi göstermesi gerekir. Geri kalan satırları da yine çalıştırıp alt klasörleri de görmemiz gerekir.

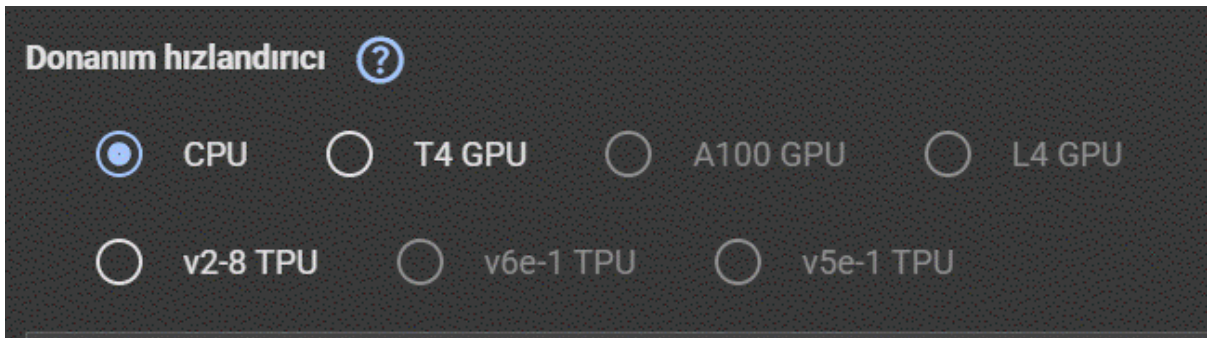
```
os.listdir(my_data) # "cell_images"  
# eger burada 'test', 'train' olursa  
# CNN tarafında NB'larda lazım olanlar
```

['train', 'test']

Geri kalan kod satırları artık tamamen Kocal'de çalıştırdığımız biçimde gerçekleştirilecektir.

NB ile çalışma bitince Çalışma türünü yine CPU olarak seçip kapatabiliriz. Aksi halde Free hakkınızdan gidecektir.

Çalışma Zamanları hakkında bilgi:



Google Colab'daki çalışma zamanı (runtime) seçenekleri, sizin kodunuzu hangi donanım üzerinde çalıştıracığınızı belirler. Görseldeki seçeneklere göre şöyle açıklayabilirim:

1. **CPU (Central Processing Unit)**
 - o Standart işlemci kullanımıdır.
 - o Hesaplama gücü diğer hızlandırıcılara göre daha düşüktür.
 - o Basit veya küçük ölçekli işlemler için yeterlidir.
2. **T4 GPU (NVIDIA Tesla T4 GPU)**
 - o GPU (Grafik İşlem Birimi) hızlandırıcısıdır.
 - o Derin öğrenme, makine öğrenmesi ve paralel hesaplama için uygundur.
 - o CPU'ya göre çok daha hızlıdır.
3. **A100 GPU (Gri renkte, seçilemez durumda)**
 - o NVIDIA'nın daha güçlü GPU modelidir.
 - o Daha büyük ve karmaşık modelleri hızlıca çalıştırmak için kullanılır.
 - o Google Colab'ın ücretsiz sürümünde genelde kullanılamaz.
4. **L4 GPU (Gri renkte, seçilemez)**
 - o Bu da gelişmiş bir GPU modelidir, ancak erişim sınırlı olabilir.
5. **v2-8 TPU (Tensor Processing Unit)**
 - o TPU, Google tarafından derin öğrenme hızlandırması için özel olarak tasarlanmış donanımdır.
 - o Özellikle TensorFlow için optimize edilmiştir.
 - o v2 sürümü orta seviyede TPU kapasitesi sunar.
6. **v6e-1 TPU, v5e-1 TPU (Gri renkte, seçilemez)**
 - o TPU'nun daha yeni veya farklı versiyonlarıdır.
 - o Ücretsiz Colab ortamlarında genellikle erişime kapalıdır.

Özetle:

- **CPU:** Genel amaçlı, en yavaş ama erişimi her zaman açık.
- **GPU (T4 GPU):** Paralel hesaplama ve derin öğrenme için uygun, hızlıdır.
- **TPU:** TensorFlow projeleri için özel hızlandırıcı, paralel ve büyük işlemler için ideal.
- **Diğer GPU ve TPU'lar** genellikle ücretli veya kısıtlı erişime sahiptir.

İhtiyacınıza göre çalışma zamanınızı seçebilirsiniz. Örneğin, basit işlemler için CPU yeterli, derin öğrenme modelleri için T4 GPU veya TPU kullanmak çok daha hızlı sonuç verir.

CNN eğitimi ve büyük derin öğrenme projeleri için TensorFlow kullanıyorsanız **v2-8 TPU** genellikle daha hızlıdır. Ancak framework esnekliği ve küçük/orta ölçek projelerde **T4 GPU** da oldukça iyi bir tercihtir.

- **Ücretsiz kullanıcılar için T4 GPU, v2-8 TPU'ya göre daha stabil ve erişilebilir bir hızlandırıcıdır.**
- TPU genellikle daha hızlı performans sunsa da, ücretsiz kullanımda erişim kısıtlamaları ve kota nedeniyle daha az pratik olabilir.

Eğer ücretsiz Colab kullanıyorsanız ve **kesintisiz GPU erişimi ve stabil kullanım** istiyorsanız, **T4 GPU seçmek genellikle daha iyi olur.**

TPU'ya erişim varsa denemek avantaj sağlar ama sürekli erişim garantisi yoktur.

Kullanım Süreleri:

Google Colab ücretsiz sürümünde TPU ve GPU kullanımına dair kota ve limitlerle ilgili detaylar şu şekildedir:

1. GPU (T4 GPU) Kullanım Kotası ve Limitleri

- **Erişim Süresi:**
Ücretsiz kullanıcılar genellikle günde birkaç saat GPU kullanabilir. Bu süre tam olarak açıklanmasa da, genelde 12 saate kadar GPU kullanımına izin verildiği gözlemlenmiştir.
 - **Oturum Süresi (Session Timeout):**
En fazla 12 saat kesintisiz çalışma hakkınız var, ancak bazen daha kısa sürelerde oturum sonlandırılabilir. Oturum kapanırsa, GPU erişimi kaybedilebilir.
 - **Bekleme Süresi (Cooldown Period):**
Yoğun talep durumunda, GPU erişiminiz geçici olarak kısıtlanabilir. Bu durumda bazı saatler ya da günler GPU erişimi kısıtlanabilir.
 - **Kullanım Yoğunluğu:**
Yoğun saatlerde GPU kaynakları sınırlı olabilir ve erişim zorlaşabilir. Gece veya daha az yoğun zamanlarda erişim daha kolaydır.
-

2. TPU (v2-8 TPU) Kullanım Kotası ve Limitleri

- **Erişim Süresi:**
TPU kullanımı genellikle GPU'ya göre daha kısıtlıdır ve ücretsiz kullanıcılar için erişim süresi daha kısa olabilir (örneğin günde 1-2 saat).
 - **Oturum Süresi:**
TPU ile çalışan oturumlar da 12 saate kadar açık kalabilir ancak TPU kullanım süresi GPU'ya göre çok daha sınırlı olabilir.
 - **Bekleme Süresi:**
TPU kaynakları daha sınırlı olduğu için, erişim sırasında bekleme süreleri oluşabilir.
 - **Yoğunluk ve Erişim:**
TPU'lar ücretsiz sürümde daha nadir tahsis edilir, dolayısıyla her zaman erişim garantisi yoktur.
-

3. Genel Notlar

- **Pro sürüme (Colab Pro/Pro+) geçiş yaparsanız,**
 - GPU ve TPU kaynaklarına erişim süresi artar,

- Bekleme süreleri azalır,
- Daha güçlü GPU ve TPU modellerine erişebilirsiniz (örneğin A100 GPU).
- **Kaynakların kullanım süresi ve erişim kotası Google tarafından dinamik olarak ayarlanır** ve bu kotalar zaman zaman değişebilir.

Özet:

Özellik	T4 GPU (Ücretsiz)	v2-8 TPU (Ücretsiz)
Günlük Kullanım Süresi	Ortalama 8-12 saat	Ortalama 1-2 saat
Oturum Maks Süresi	Yaklaşık 12 saat	Yaklaşık 12 saat
Erişim Sıklığı	Daha sık, daha stabil	Daha sınırlı, bekleme olabilir
Kullanım Yoğunluğu	Yoğun saatlerde erişim zor	Daha sınırlı kaynak, nadiren verilir
Pro Sürümde Durum	Daha fazla erişim, güçlü GPU	Daha fazla erişim, güçlü TPU

İstersen, ücretsiz Colab’da bu limitlerin nasıl kontrol edildiği, oturumun nasıl yönetildiği ve erişim sınırlarını aşmamak için pratik ipuçları da verebilirim.

Kullanım Kontrolü

Google Colab ücretsiz sürümde GPU ve TPU kaynak kullanım sınırlarını aşmamak ve oturumları verimli kullanmak için bazı pratik ipuçları ve kontrol yöntemleri şöyle:

1. Kaynak Kullanımını Kontrol Etme

- **!nvidia-smi Komutu (GPU için):**
Terminal veya kod hücresinde bu komutu çalıştırarak GPU durumunu, kullanım oranını, belleği görebilirsiniz.

`!nvidia-smi`
- **TPU Durumunu Kontrol Etme:**
TPU kullanırken TensorFlow ile TPU bağlandığında durumunu kontrol edebilirsiniz:


```
import tensorflow as tf
resolver = tf.distribute.cluster_resolver.TPUClusterResolver(tpu='')
print("TPU devices:", tf.config.list_logical_devices('TPU'))
```
- **Oturum Süresi ve Bellek Takibi:**
Colab arayüzünde “Runtime” > “Manage sessions” kısmından aktif oturumlarınızı ve kalan sürelerini görebilirsiniz.

2. Oturumları Verimli Kullanma

- **Uzun Süre Boşta Kalmaktan Kaçının:**
Colab, uzun süre etkileşim olmazsa oturumu otomatik kapatır. Kodunuzun aktif çalışması için aralıklarla çıktı veya işlem olması önemli.
 - **Küçük Batch Boyutları Kullanmak:**
Büyük batch boyutları GPU/TPU belleğini hızlı doldurur ve hata verebilir. Optimize ederek belleği aşmamak için küçük batch'lerle çalışabilirsiniz.
 - **Veri ve Model Boyutunu Optimize Etmek:**
Gereksiz büyük veri veya model boyutları kullanımını azaltmak hem hız hem de kota açısından faydalıdır.
 - **Checkpoint Almak:**
Uzun eğitimlerde modelinizi sık sık kaydedin, olası kesintilerde kayıplarınız olmaz.
-

3. Limitlere Takılmamak için İpuçları

- **Kaynaklar Kısıtlanırsa:**
Oturumunuzu kapatıp biraz bekledikten sonra tekrar deneyin. Google, yoğunluk durumuna göre erişimi açar/kapatır.
 - **Farklı Saatlerde Deneyin:**
Yoğun saatler (genelde gündüz, mesai saatleri) dışında GPU/TPU erişimi daha kolay olabilir.
 - **Colab Pro Düşünün:**
Eğer süre ve performans çok önemliyse ücretli Colab Pro veya Pro+ sürümüne geçmek avantaj sağlar.
-

4. TPU ve GPU Arasında Geçiş

- Eğer GPU'nuz veya TPU'nuz kısıtlanmışsa, diğer hızlandırıcıyı deneyebilirsiniz. Projeye göre bazen GPU daha stabil erişim sunar.
-

5. Kullanım Sınırlarını Aşmamak için

- Aynı anda çok fazla oturum açmayın.
 - Oturumlarda ağır hesaplamaları gereksiz yere uzun tutmayın.
 - Hesabınızın Google politikalarına uygun davranın (spam, kötüye kullanım vb. yapmayın).
-

