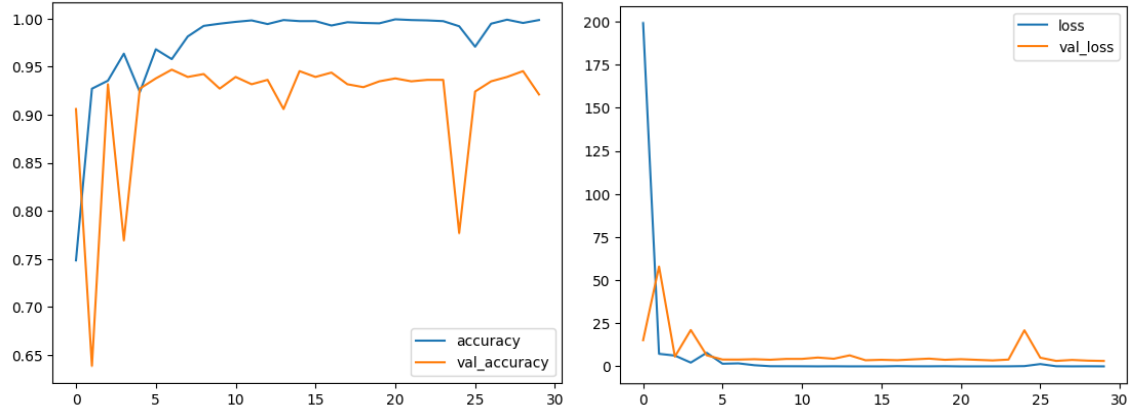


### Soru 1 :

Training log'a göre, modelin yalnızca birkaç epoch içinde yaklaşık %99'luk bir doğruluğa eriştiğini görüyoruz. Bununla birlikte, validation doğruluğu dalgalanıyor ve tutarlı bir şekilde artmıyor, bu da fazla overfitting işaret ediyor olabilir. Test loss'a baktığımızda da bu sonuca varabiliriz. test accuracy oldukça yüksek olmasına karşın test loss'ta oldukça yüksektir.

Test Loss: 3.164019823074341

Test Accuracy: 0.9210925698280334



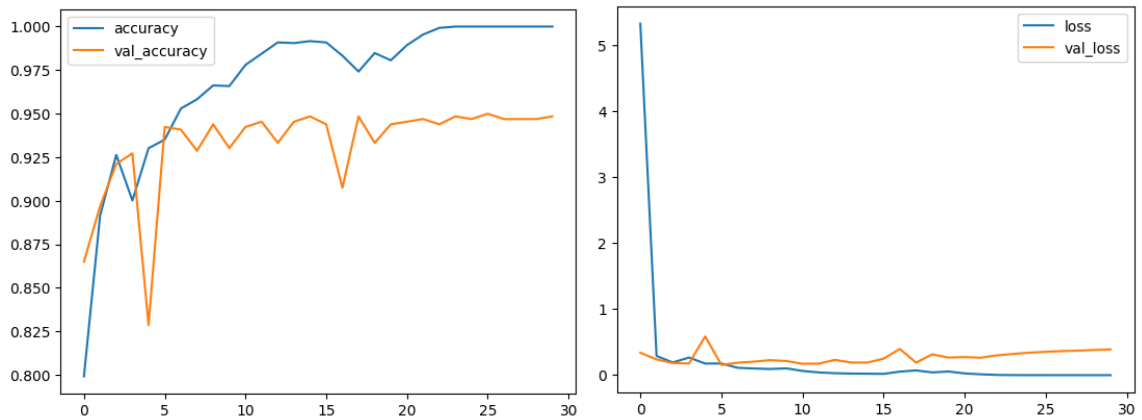
### Soru2:

- Hyperparameter optimizasyonu yapılabilir. Modeldeki gizli katman sayısı, batch\_size sayısı, epoch sayısı, aktivasyon fonksiyonları, optimizier tipleri değiştirilebilir.

#### 1. Optimizasyonda yapılanlar=gizli katman sayısı artırıldı

Elde edilen sonuç= epoch24'e kadar gayet iyi şekilde ilerlense de sonrasında overfitting oldu. çünkü loss,val\_loss'tan çok uzaklaşmaya başladı.

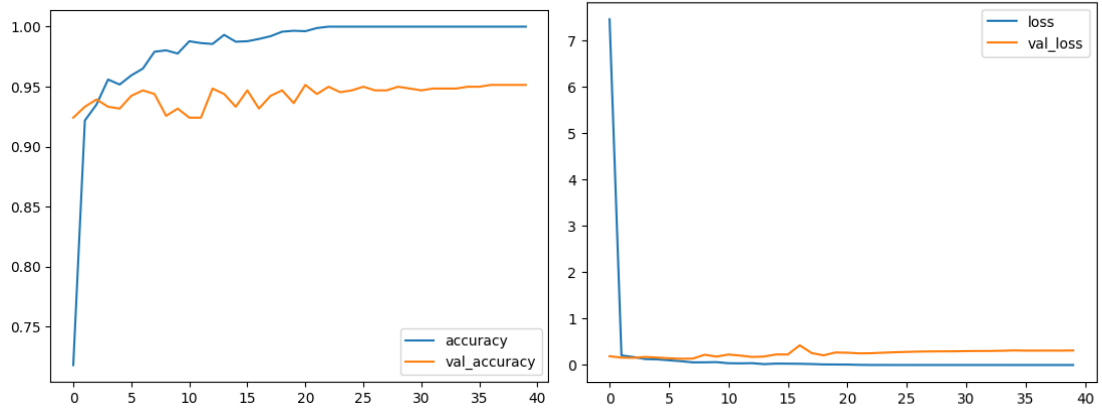
Test sonuç= Test Loss: 0.39007508754730225, Test Accuracy: 0.9484066963195801



2. Optimizasyonda yapılanlar= batch\_size sayısı ve epoch sayısı arttırıldı.

Elde edilen sonuç= epoch23'e kadar gayet iyi şekilde ilerlense de sonrasında overfitting oldu. çünkü loss,val\_loss'tan çok uzaklaşmaya başladı.

Test sonuç=Test Loss: 0.31337684392929077, Test Accuracy: 0.9514415860176086

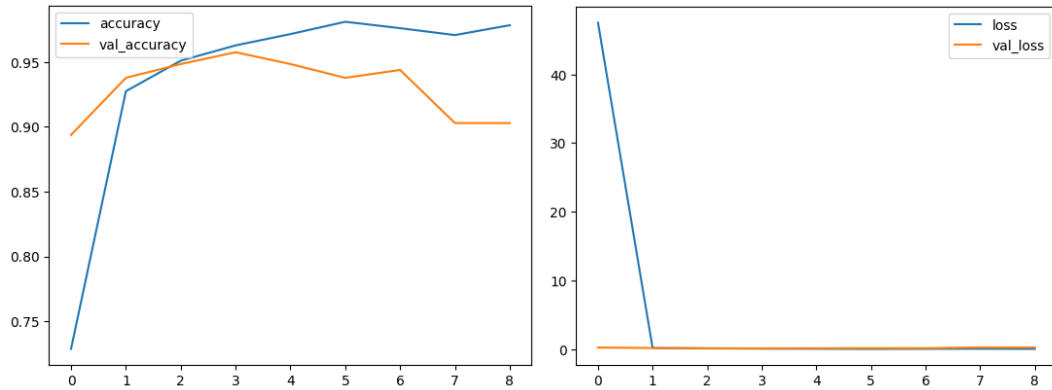


-Data düzgün seçilebilir yada preprocess yapılabilir. şuan ki datanın içerisindeki verilerin boyutları birbirinden farklı bir şekilde.Bu cnn için çok sorun olmasada yinede önemli bir etkidir. Bu yüzden ya verideki bütün resimler aynı boyuta getirilebilir yada aynı boyutlu data seçilebilir. Ben zaten bu işlemi gerçekleştirdim ve 256, 256 olacak şekilde resize ettim. Daha yüksek formatlar seçmek daha fazla feature anlamına geleceğinden boyut arttırılabilir. Early stopping eklenebilir.

3. Optimizasyonda yapılanlar= fotoğraflar 256,256'ya rotate edildi. Early stopping eklendi

Elde edilen sonuç= epoch 14'de loss dalgalanmaya başladı.

Test sonuç= Test Loss: 0.2685064375400543, Test Accuracy: 0.9028831720352173



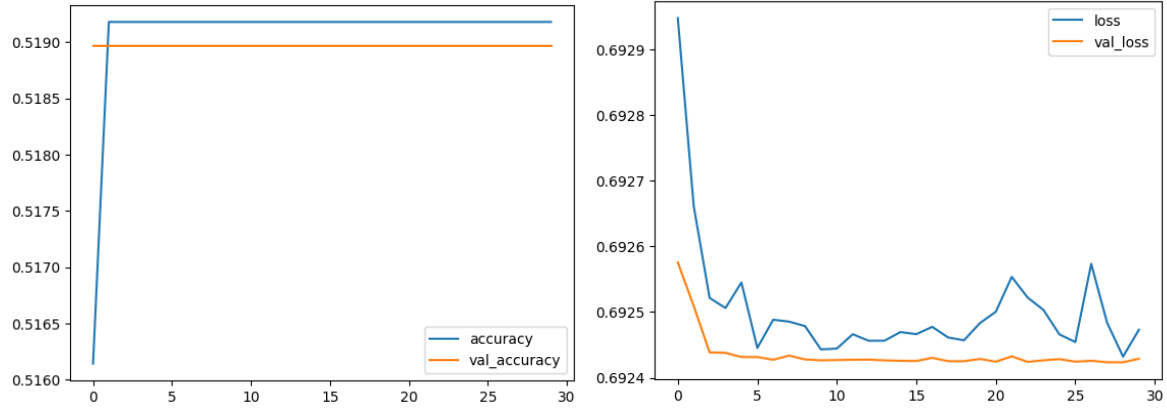
-Model komple baştan değiştirilebilir. Mesela çok daha derin ağa sahip olan resnet50 modeli kullanılabilir. İçinde 50 tane gizli ağ bulunmasının yanı sıra çok kolay bir şekilde imagenet'in pretrain

edilmiş ağırlıklarının üzerine bir eğitim gerçekleştirilebilir. Bu eğitimin çok daha maliyetli olmasına sebep olucaktır ancak overfitting'i önleyebilir. L1 çürütme uygulanabilir. çürütme modelin başlangıçtaki epochlarında çok ilerleyen epochlarında daha az olacak şekilde uygulanabilir. Bu imagenet ağırlıklarına bağlı kalınmasını önler ve bu sayede ağırlıkların genelleştirme yeteneği artar.

4. Optimizasyonda yapılanlar= Model resnet50 olarak değiştirildi.

Elde edilen sonuç= accuracy ve val\_accuracy epoch boyunca çok az değişiklik gösterdi.

Test sonuç= Test Loss: 0.6924288272857666, Test Accuracy: 0.5189681053161621

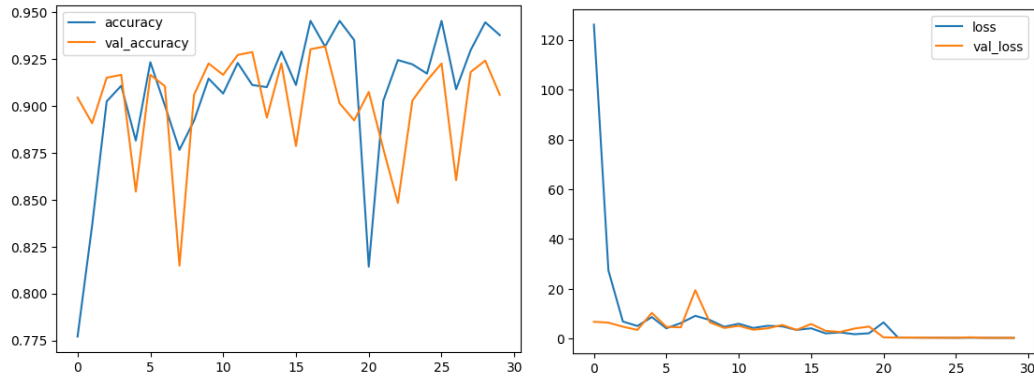


Soru3:

- Eğitim geçmişine baktığımızda, azalan kayıp ve artan doğrulukla kanıtlandığı gibi, modelin hem eğitim hem de doğrulama veri setinin performansının zaman içinde iyileştiğini görebiliriz. Ancak doğrulama kaybının arttığı birkaç epoch vardır ve bu durum overfitting'e işaret edebilir.

Test Loss: 0.3421761393547058

Test Accuracy: 0.9059180617332458



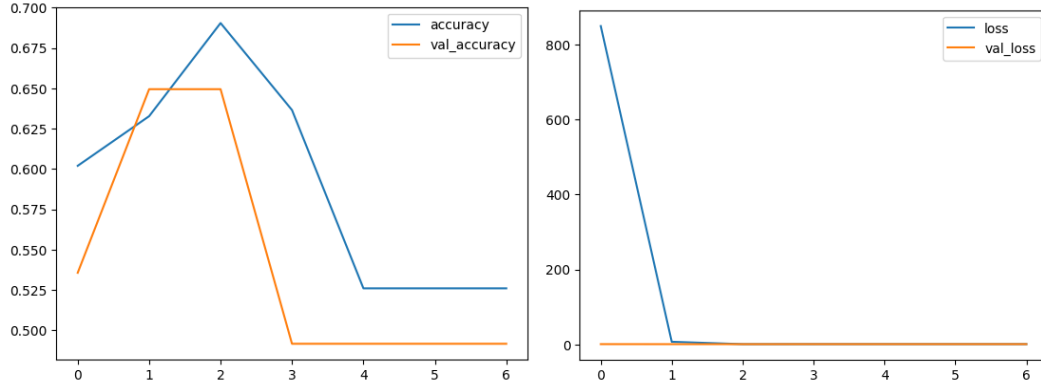
Soru4:

Verinin boyutu 256,256 olarak sabitlenebilir. Early stopping eklenebilir.

1. Optimizasyonda yapılanlar = Boyut değiştirildi. Early stopping eklendi.

Elde edilen sonuç= epoch 7'de erken durdurma devreye girdi.

Test sonuç= Test Loss: 0.6952791810035706, Test Accuracy: 0.49165400862693787

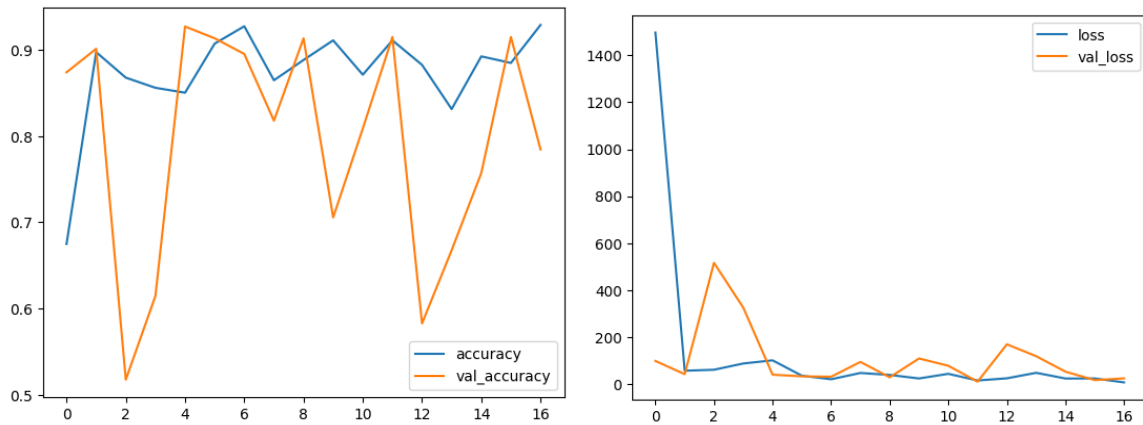


Hyperparameter optimizasyonu yapılabilir. Modeldeki gizli katman sayısı, batch\_size sayısı, epoch sayısı, aktivasyon fonksiyonları, optimizier tipleri değiştirilebilir.

2. Optimizasyonda yapılanlar = gizli katman sayısı artırıldı. Epoch sayısı artırıldı. Batch\_size artırıldı.

Elde edilen sonuç= epoch 17'de erken durdurma devreye girdi.

Test sonuç= Test Loss: 24.13956069946289, Test Accuracy: 0.7845219969749451



Soru5:

Resnet50'nin daha iyi performans gösterdiğini düşünüyorum. Genelleştirmeyi imagenet sayesinde en çok o başardı. Diğer modeller çok hızlı bir şekilde modeli ezberleme tarafındaydılar. Her ne kadar resnet50'nin doğruluğu düşük gibi gözüksede model sayısını arttırarak diğer modelleri kolayca geçebileceğini düşünüyorum.

Kaynak kodlar için github linkim: <https://github.com/ipproo/YapaySinirAglariProje>

COLAB Linkleri:

FCFF:

<https://colab.research.google.com/drive/1gmvcmxzX6ryVWQtxzwd7lnWDBn0jpiXg?usp=sharing>

FCFF\_OPT:

<https://colab.research.google.com/drive/1JLmCIFINM1fYTNysEmpZqNH4riRDQhc?usp=sharing>

CNN:

[https://colab.research.google.com/drive/1\\_9Mw6LUzMb\\_Co88D5bHBLfiDWkTXOOofU?usp=sharing](https://colab.research.google.com/drive/1_9Mw6LUzMb_Co88D5bHBLfiDWkTXOOofU?usp=sharing)

CNN\_OPT:

<https://colab.research.google.com/drive/1Y0XtATU4BpBvg0h3ky6xU2r0EEfpsvCh?usp=sharing>