



GÖRÜNTÜ İŞLEME

DÖNEM PROJESİ

İrem ATILGAN

17061036

09.01.2021

1. Orijinal bir konvolüsyonel ağının tasarlanması, hiper-parametrelerin denenmesi ve sonuçlarının raporlanması

Bu bölümde farklı parametrelerle CNN mimarileri oluşturulmuş ve bu mimarilerin CINIC-10 verisetine göre başarısı test edilmiştir. Modellerin eğitim ve test süreci aşağıdaki adımlarla gerçekleşmiştir :

1. CINIC-10 Verisetinin Kaggle'dan alınarak eğitim, test ve validasyon dizilerine aktarıldı,
2. Verisetleri random şekilde shuffle edildi,
3. Veriler 255'e bölünerek normalize edildi,
4. 32x32 Boyutlarında olan verileri 224x224 boyutuna getirildi
5. Hedef dizisi (etiketler) One Hot Encoding ile sayısallaştırıldı
6. Modeller oluşturuldu ve derlendi,
7. Ön işlemeden geçirilmiş olan veriler modellere eğitilmek üzere verildi.

Modellerin eğitimi ve değerlendirilmesi Google Colab ortamından gerçekleştirilmiştir. RAM limitinin normalizasyon ve eğitim işlemlerinde sıkça dolmasından dolayı veriseti içerisinden, eğitim, test ve validasyon için her birinden 2000 tane olmak üzere, toplam 6000 veri üzerinde işlem yapılabilmiştir. Buna ek olarak, yine aynı sebepten, batch size 16 olarak alınmıştır. Alınan tüm bu kararlar, modelin başarısını kritik olarak etkilemiştir.

Denenen hiperparametrelerde; derleyici olarak "adam" ve "rmsprop"; dropout oranı olarak 0.2 ve 0.7; filtre boyutu olarak 16 ve 32 denenmiştir. Farklı kombinasyonlar için farklı modeller oluşturulmuş olsa da (katman sayısı, başlatıcı fonksiyon, filtre sayısı), donanımsal kaynakların ve zamanın yetersizliğinden dolayı sayılı modeller denenebilmiştir.

Projede kullanılan modüller ise aşağıdaki gibidir:

1. **def upload_images(data_type)** : Fotoğrafların bulunduğu dosyadan okuma yapan fonksiyondur. Parametre olarak "train", "test" veya "validation" string'ini alır ve o isimdeki dosyadan okuma yapar. Etiket (labels) ve fotoğraf dizisi döner.
2. **def get_model(num_filters, filter_size, layer_size, initializer_func, dropout_ratio)** : CNN model kurma fonksiyonudur. Parametre olarak filtre sayısı, filtre boyutu, katman sayısı, başlatıcı fonksiyon ve dropout oranını alır; Son katmanlarda Flatten() fonksiyonu ile özellik matrisi düzleştirilerek (10,1) çıktı verecek şekilde düzenlenmiştir. Oluşan son model çıktı olarak dönlür.
3. **def plot_metric(history, metric)** : Model kayıtlarının grafikleştirildiği fonksiyondur. Parametre olarak eğitim kayıtlarını ve grafiği çizilmesi istenen değer alınır; değer olarak "accuracy" verilirse eğitim ve validasyon doğruluğu, "loss" verilirse eğitim ve validasyon hatası grafikleştirilerek kullanıcıya sunulur.

Ödevde kullanılan modeller, açıklamaları ve başarıları aşağıda detaylı olarak paylaşılmıştır.

1.1 Modeller

Model 1

Model Özellikleri :

Katman Sayısı : 3

Dropout Oranı : 0.2

Filtre Sayısı : 16

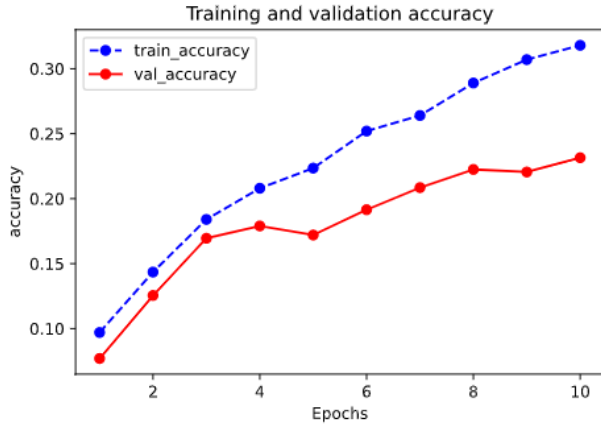
Optimizasyon Fonksiyonu : Adam

Filtre Boyutu : 3x3

Model: "sequential_6"

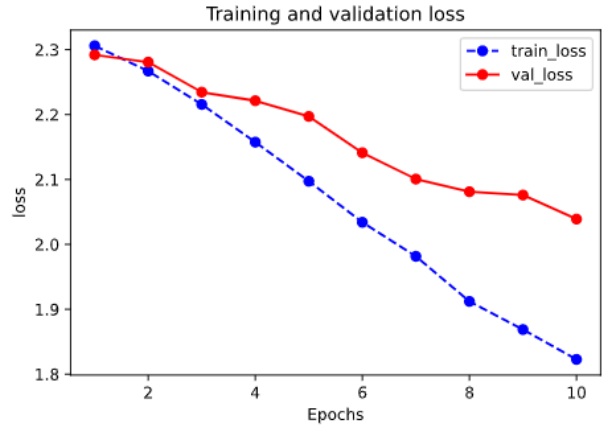
Layer (type)	Output Shape	Param #
conv2d_18 (Conv2D)	(None, 222, 222, 16)	448
max_pooling2d_18 (MaxPooling)	(None, 111, 111, 16)	0
dropout_18 (Dropout)	(None, 111, 111, 16)	0
conv2d_19 (Conv2D)	(None, 109, 109, 16)	2320
max_pooling2d_19 (MaxPooling)	(None, 54, 54, 16)	0
dropout_19 (Dropout)	(None, 54, 54, 16)	0
conv2d_20 (Conv2D)	(None, 52, 52, 16)	2320
max_pooling2d_20 (MaxPooling)	(None, 26, 26, 16)	0
dropout_20 (Dropout)	(None, 26, 26, 16)	0
flatten_6 (Flatten)	(None, 10816)	0
dense_12 (Dense)	(None, 3)	32451
dense_13 (Dense)	(None, 10)	40
Total params: 37,579		
Trainable params: 37,579		
Non-trainable params: 0		

Aşağıda modelin eğitimi sırasında değişen eğitim,validasyon doğruluğu ve eğitim,validasyon hatası gösterilmektedir :



train_accuracy : 0.3241

val_accuracy : 0.2315



train_loss : 1.7894

val_loss : 2.0390

Model 2

Model Özellikleri :

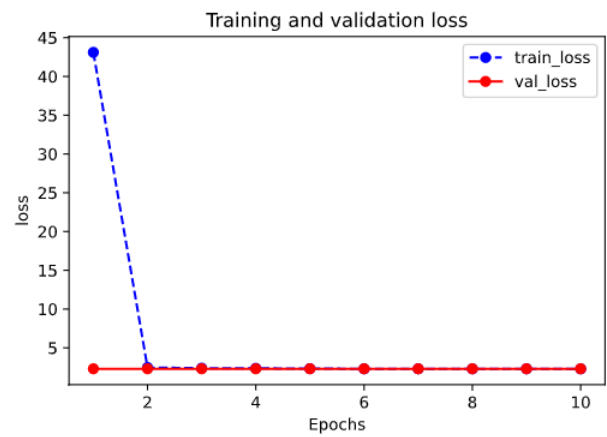
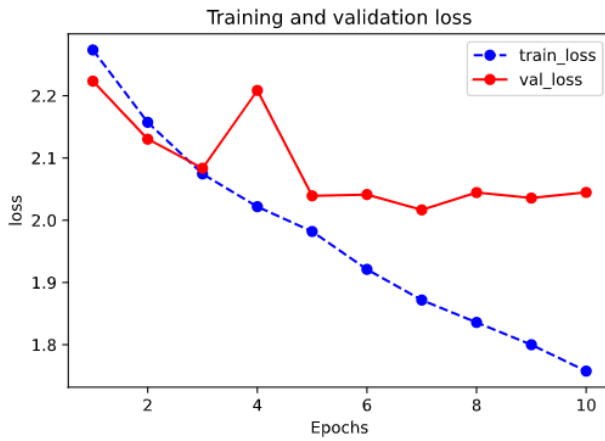
Katman Sayısı : 3

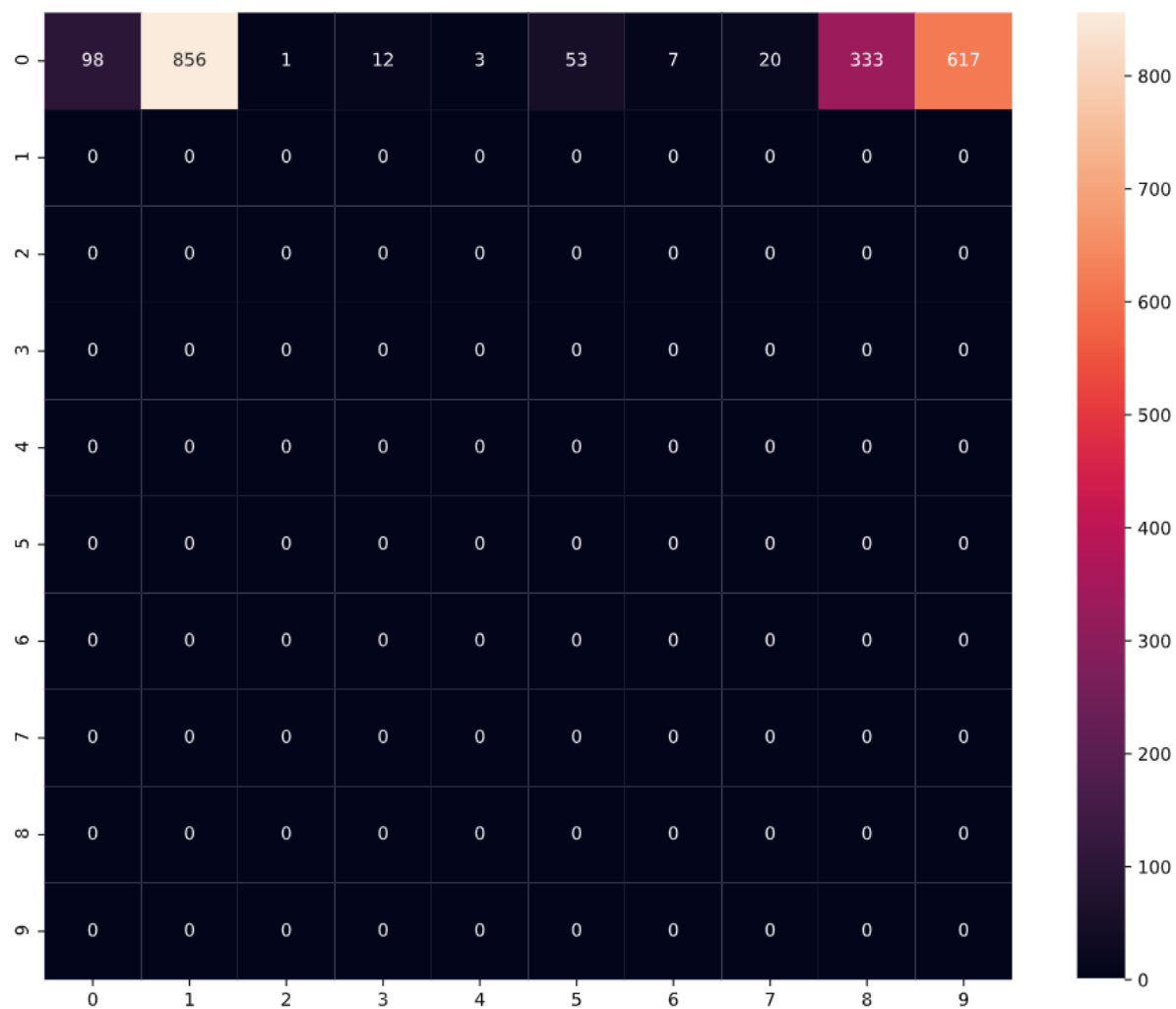
Dropout Oranı : 0.2

Filtre Sayısı : 32

Optimizasyon Fonksiyonu : Adam

Filtre Boyutu : 3x3





	precision	recall	f1-score	support
0	1.00	0.05	0.09	2000
1	0.00	0.00	0.00	0
2	0.00	0.00	0.00	0
3	0.00	0.00	0.00	0
4	0.00	0.00	0.00	0
5	0.00	0.00	0.00	0
6	0.00	0.00	0.00	0
7	0.00	0.00	0.00	0
8	0.00	0.00	0.00	0
9	0.00	0.00	0.00	0
accuracy			0.05	2000
macro avg	0.10	0.00	0.01	2000
weighted avg	1.00	0.05	0.09	2000

train_accuracy : 0.0901

train_loss : 2.2980

val_accuracy : 0.0985

val_loss : 2.3031

Model 3

Model Özellikleri :

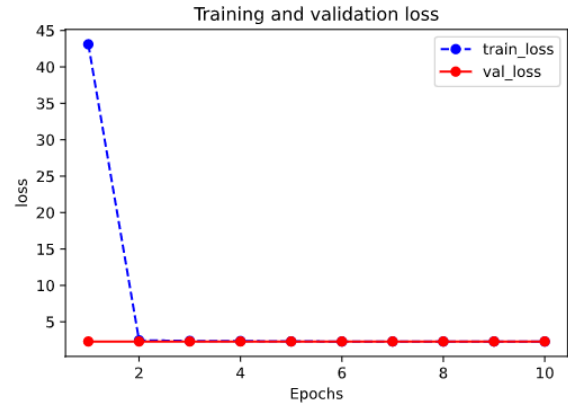
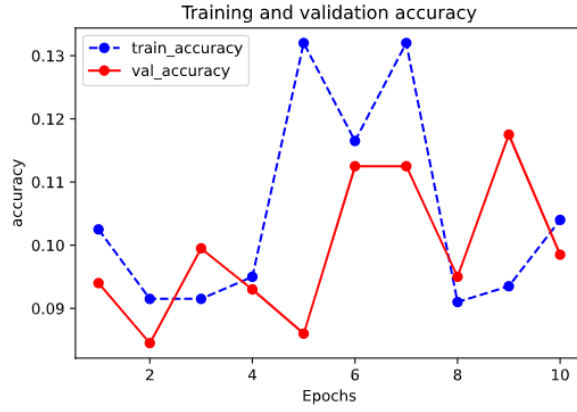
Katman Sayısı : 3

Dropout Oranı : 0.7

Filtre Sayısı : 32

Optimizasyon Fonksiyonu : Adam

Filtre Boyutu : 3x3



	precision	recall	f1-score	support
0	1.00	0.90	0.94	2000
1	0.00	0.00	0.00	0
2	0.00	0.00	0.00	0
3	0.00	0.00	0.00	0
4	0.00	0.00	0.00	0
6	0.00	0.00	0.00	0
7	0.00	0.00	0.00	0
8	0.00	0.00	0.00	0
9	0.00	0.00	0.00	0
accuracy			0.90	2000
macro avg	0.11	0.10	0.10	2000
weighted avg	1.00	0.90	0.94	2000

train_accuracy : 0.0901

train_loss : 2.2980

val_accuracy : 0.0985

val_loss : 2.3031

1.2 Sonuç

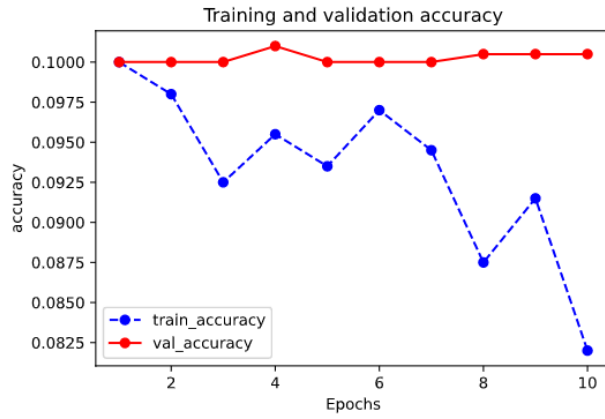
Sonuçlarda da görüleceği üzere, filtre sayısının artırılması ve filtre boyutunun küçülmesi, en yüksek eğitim başarısını getirmektedir. Buna rağmen veri sayısındaki yetersizlik, başarının oldukça az oranda kalmasına sebep olmuştur.

2. Öğrenim Aktarımının Gerçeklenmesi ve Raporlanması

Bu bölümde ImageNet verileri ile eğitilmiş VGG16 ve ResNet50 hazır mimarileri ile modeller oluşturulmuş ve mevcut verilerle eğitimleri gerçekleştirilmiştir. İki mimarinin de son dört katmanı hariç tüm katmanlarının eğitilmesi ve parametrelerinin değiştirilmesi engellenmiştir.

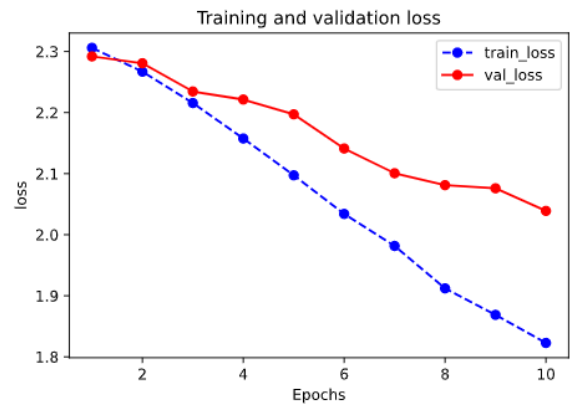
Aktarım Öğrenimi (Transfer Learning) ile eğitim sonucu elde edilen başarı, hazır olmayan mimarilerin başarılarına göre çok daha düşük kalmaktadır. Bunun tahmin edilebilir bazı sebepleri vardır; Bu tür öğrenme, ImageNet veriseti kullanılarak 1000'den fazla sınıf için eğitilmiştir. Her ne kadar görüntüleri en küçük özelliklerine kadar ayırmada başarılı olsalar da, mevcut verisetimizde 10 kategori bulunduğundan ve derin öğrenme için oldukça az veri üzerinde çalışıldığından, modeller genelleştirmede başarılı olamayabilirler. Bu da başarının eğitimle yeterince yükselememesine sebep olmuş olabilir.

ResNet50



train_accuracy : 0.0808

val_accuracy : 0.1005



train_loss : 2.3026

val_loss : 2.0390