BLM404 YAPAY ZEKA

TEXT-TO-IMAGE SENTEZİ
(OXFORD 102 FLOWERS VERİ KÜMESİ KULLANILARAK)

BİLGİSAYAR MÜHENDİSLİĞİ

Egehan Demir	Emre Nihat Durgun	Alparslan Beraat Özdemir
180202040@kocaeli.edu.tr	180202015@kocaeli.edu.tr	170202045@kocaeli.edu.tr

I. ÖZET

Tensroflow kütüphanesi ve "Oxford 102 flowers" veri kümesini kullanarak, verilen İngilizce metinler ile (Türkçeye çeviremedik) ilgili çiçek resimlerini gösterilmesini sağlamak.

II. GİRİŞ

CNN(Convolutional Neural Network) ve RNN(Recurrent Neural Network), LSTM öğrenme algoritmalarını kullanarak eğitilmiş veri kümeleriyle metinden resim sentezi yapılmıştır. CNN bilgisayardaki objeleri ve resimleri tanımak için kullanılan çok yaygın bir sinirsel ağ modelidir. RNN isminden de anlaşılacağı gibi geçici ardaşık bilgileri yorumlar. Bunu yapabilmek için verileri belli bir sıraya sokar ve bu yöntemi de girdileri alarak ve tekrar bu inputları bir önceki veya bir sonraki düğümlerin aktivasyonlarına göre kullanarak tahmin işini gerçekleştirmeye çalışır. RNN daha geçici bilgiler ve veriler için tasarlanmıştır.

RNN ile CNN arasındaki en temel fark ise geçici bilgilerin işlenilebilirliğidir. Veriler, diziler gibi cümleler halinde gelir. CNN geçici bilgileri işlemekte zorlandığı için RNN daha spesifik amaçlar için kullanılır. CNN ve RNN çok farklı alt yapılara sahiptir. Bunların daha farklı amaçları ve kullanım durumları vardır.

III. YÖNTEM

a) Veri Seti

Oxford Flowers 102 veri seti, Birleşik Krallık'ta yaygın olarak görülen 102 çiçek kategorisinden oluşan bir tutarlılıktır. Her sınıf 40 ila 258 görüntüden oluşur. Görüntüler büyük ölçekli, poz ve ışık varyasyonlarına sahiptir. Ayrıca, kategori içinde büyük varyasyonları olan kategoriler ve çok benzer birkaç kategori vardır.

Veri seti eğitim seti, doğrulama seti ve test seti olmak üzere ikiye ayrılır. Eğitim seti ve doğrulama setinin her biri, sınıf başına 10 görüntüden oluşur (her biri toplam 1020 görüntü). Test seti kalan 6149 görüntüden oluşur (sınıf başına en az 20).

Bu resimler büyük ölçeklerde, pozlarda ve renklendirmelerde olabilir. Ayrıca data setleri isomap ile birlikte biçim ve renkleri kullanarak görselleştirilir.

b) Algoritma

Üretken Mualif ağlarda iki tane üretici göz önünde bulundurulur. Üretici " G " ayırıcı " D ". Ayrımcı yapay ve sentetik resimler arasındakileri ayırır. Üretici, ayrımcıyı kandırmaya çalışır. Sonuç olarak D ve G şu matematiksel algoritmada bir oyun oynar;

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{data}(x)}\left[\log D(x)\right] + \mathbb{E}_{x\sim p_z(z)}\left[\log\left(1-D\left(G\left(z\right)\right)\right)\right].$$

Model mimarisi 2 adımdan oluşur;

- Adım 1-GAN Taslak
- Adım 2-GAN Süperçözünürlük

Adım 1-GAN:Taslak

Mimarimiz GAN'ı eğiterek düşük çözünürlükteki resimleri üretmektir. Bu aşamada, metin tanımlarını kodlanmış gömülü metinler " Pt " olarak tanımladık. Bu gömülü metinler için derin altyapılı metin gömme yaklaşımı kullanılmıştır.

Adım 2-GAN:Yüksekçözünürlük

2. adımımızda StackGAN yüksek çözünürlüklü bir araç olarak davranmıştır. 2. Adımda, aşağıda verilen düşük çözünürlüklü örnek ve 1. Adımda bulunan koşullu gömülü metin gibi davranarak ayrımcı D'yi ve üretici G'yi, Ld'yi max alarak ve Lg'yi minimum alarak eğitir.

$$\mathcal{L}_D = \mathbb{E}_{(I,t)\sim \mathrm{p_{data}}}\left[\log D\left(I,\varphi_t\right)\right] + \mathbb{E}_{s_0\sim \mathrm{p}_{G_0},t\sim \mathrm{p_{data}}}\left[\log\left(1-D\left(G\left(s_0,c\right),\varphi_t\right)\right)\right],$$

and

$$\mathcal{L}_G = \mathbb{E}_{s_0\sim \mathrm{p}_{G_0},\mathrm{t}\sim \mathrm{p_{data}}}\left[\log\left(1-D\left(G\left(s_0,c\right),\varphi_t\right)\right)\right] + \lambda D_{\mathrm{KL}}\left(\boldsymbol{\mu}_{\varphi_t,\Sigma_{\varphi_t}}\right),$$

c) Fonksiyonlar

1- Data Loader Python Dosyası

102flowers ve text_c10 klasörünün yolunu bulmaktadır. Her bir resmin captions dosyası içerisinde 10 tane cümlesi mevcut, burada her bir klasörü açıp içindekleri txt dosyalarını bulundurmaktadır. Txt dosyasındaki metinler, tensorflow cümle işleme fonksiyonuna sokulur. Cümlede başlama(<S>) ve bitirme(</S>) kalıplarını ekler ve bunu "*processed_capts*" değişkenine entegre eder. Vocab tensorflow txt dosyasına göre "*processed_capts*" arrayindeki captionlardan metin belgesini oluşturur eğer dosya varsa uyarı yapar. Bütün captionları nltk tokenizer vasıtasıyla parçalara ayırıp bunları "captions_ids" arrayine ekler. Kontrol örnek bir resmin cümlesini, cümlesinin parçalanmış halini amaçlı parçalanmış kelimelerin idsini yazmaktadır.Resimlerin olduğunu belirtilen klasörüden bütün resimleri alınır kaç tane olduğunu yazdırılır ve bu resimleri yeniden boyutlandırmaya başlatılır.

Resimleri tek tek boyutlandırıp float32 tipine çevirir. Eğer stackGAN kullanımı için need_256 boolunu true yapılırsa 256'ya 256 olarak tekrar boyutlandırıp images ve images_256 arrayine atılır.

2- Utils Python Dosyası

load_and_assign_npz() -> oluşturulan .npz uzantılı dosyaları modeli tekrar çalıştırmak için kullanılır.
load_folder_list() -> verilen klasördeki klasör listesini döndürür.
print_dict()->dictinary yapısında bütün anahtarları ve itemları yazdırır.
get_random_int() -> random bir liste döndürür.
preprocess_caption() -> verilen caption hakkında öndüzenlemeye yapar.
Merge() -> resimleri belirtilen boyut içerisinde boyutlandırıp bir array haline getirir
save_images() -> ## resimleri kaydetmesi için imsave fonksiyonuna yönlendirir
prepro_img() -> ##verilen resim verilen mode tipine göre çeşitli değişikliklere uğrar. Bu değişiklikler genelde yönünü değiştirme, yeniden boyutlandırma kırpma vb eylemlerdir.

3- Sonuçlar Python Dosyası

Önceden oluşturulmuş model dosyalarını ilgili dizinden çeker ve tensorflow oturumu oluşturur. Oluşturulan oturum verilmek üzere örnek cümleler alınır, alınan cümlelere tokenize edildikten sonra uygun sonuçlar çıkartılması amacıyla oluşturulan oturuma verilir. Verilen örnek cümleler üzerinde işlem yapıldıktan sonra ise sonuç elde edilerek kod tamamlanır.

4- Text2Image Python Dosyası

Train edebilmek amacıyla numpy arraye çevirilir. Dataların tutulmasını sağlayan altyapının oluşturulur.
Reduce Mean -> Bir tensörün boyutları boyunca öğelerin ortalamasını hesaplar.
RNN Loss = RNN öğrenmesi sırasında farkedilen kayıp oranını belirler.
Random görseller üreten model içerisindeki fonksiyon çağırılır. Generatorun oluşturduğu random görselleri gerçek/sahte diye sınıflandırır. Sahte fotoğraflar oluşturulur. Ayırma fonksiyonu ile sahte fotoğraflar ayrılır. Ayırma fonksiyonu ile gerçek fotoğraflar ayrılır. Gerçek resimdeki metinlerle olan uyuşmazlıklar bulunur.

Tensorlayerlar üzerinden ilgili tensor katmanları çağırılır. Daha sonra en son train edilmiş model kayıtlarını yükler. Array içindeki örnek cümleleri preprocess_caption vasıtasıyla parçalar. word tokenize ile kelimeleri parçalayıp örnek cümleler dizisinin her bir elemanına parçaladığı kelimeleri atar.

pad_sequences -> fonksiyonu ile Array boyutlarını eşit hale getirir.

Veriseti üzerinden kaç defa geçileceği, kaç öğrenme tekrarı olacağı epoch ile belirtilir. Toplam train sayısının bir seferde yapılacak işlem sayısına bölümü

sonucunda epoch başına düen batch sayısı belirlenir. Epochları döngü ile epoch sayısı kadar uygular. Her epochta işlem yapılacak batch sayısı döner.

5- Model Python Dosyası

rnn_embed() -> Cümlelerin girdisini alarak öğrenmeyi sağlayan bir derin öğrenme algoritmasıdır.

generator_txt2img_resnet() -> Oluşturulan katman yapısı vasıtası ile sahte görüntüler üretir.

discriminator_txt2img_resnet() -> Generatorun oluşturduğu random görselleri gerçek/sahte diye sınıflandırır.

d) Kütüphaneler

1- tensorflow.compat.v1

Tensorflow 1.x ve Tensorflow 2.x kütüphanlerine kod yazılmasına izin verir.

2- Tensorlayer

Google TensorFlowdan genişletilmiş Deep Learning ve Reinforcement Learning kütüphanesidir.

3- Nltk

Natural Language Toolkit; insan dili verileriyle çalışmak için Pyhton programlama dili ile geliştirilmiş ve geliştirilmekte olan 50'nin üzerinde derlem(corpus) ve sözcük kaynağı(lexical resources) ile oluşturulmuş açık kaynaklı bir kütüphanedir.

4- Scipy

SciPy, Python'un Numpy uzantısına dayanan matematiksel algoritmalar ve kolaylık işlevlerinin bir koleksiyonudur. Kullanıcıya veriyi manipüle etmek ve görselleştirmek için üst düzey komutlar ve sınıflar sunarak etkileşimli Python oturumuna önemli güç katar.

5- imageio

6- matplotlib

Matplotlib, figürlerin görselleştirilmesinde biz analizcilere yardımcı olan 2 Boyutlu bir çizim kütüphanesidir.

7- scikit-image

Scikit-learn veya Sklearn makine öğrenmesi modelleri oluşturmak için kullanılan Python temelli bir kütüphanedir. Regresyon, kümeleme ve sınıflandırma için kullanılan pek çok öğrenme algoritmasına sahiptir.

IV. SONUÇ

Derin öğrenme yöntemleri bilgisayar görüsünde önemli başarılara yol açmıştır. Bu çalışmada, oldukça küçük ve ince görsel farklılıkları olan sınıflarda bir çalışma yapabilmek için Oxford 102-Flowers veri kümesi kullanılmıştır. GAN algoritmasının yaptığımız çalışmada ve benzer çalışmalarda oldukça yaygın olarak kullanıldığı için proje için oldukça uygun bir algoritma olduğu sonucuna ulaşıldı. GAN – Çekişmeli nöral ağların çalışma mantığını kavranıldı.

Makine öğrenmesi yöntemleri ile model eğitmenin ne kadar büyük bir zaman ve işlemci kaynağı harcadığını ortaya koyuldu, GPU gücünün bu konudaki önemi ortaya çıkmış oldu. Bir veri seti eğitilirken eğitim algoritmasındaki parametrelerin hem eğitim süresini hem de doğruluk sonuçlarını ciddi ölçüde etkilediğini ve bu parametrelerin uygun bir şekilde ayarlanması gerektiğini görülmüştür.

V. KAYNAKÇA

1. OKOKPROJECTS, “Text to Image Synthesis in Python”, https://www.youtube.com/watch?v=sVuop7jBch4,
2. ZSDONGHAO, “Text To Image Synthesis”, https://github.com/zsdonghao/text-to-image
3. RUYUEGUANGHUA, “GAN text_to_image_102flowers – RIEYUGUANGHUA”, https://www.kaggle.com/code/riyueguanghua/gan-text-to-image-102flowers-rieyuguanghua/notebook
4. Tensorflow, “oxford_flowers102”, https://www.tensorflow.org/datasets/catalog/oxford_flowers102
5. Büşra Rümeysa Mete, “Derin Öğrenme ile Görüntü Sınıflandırma”, https://95.183.179.45/xmlui/bitstream/handle/20.500.12831/3899/3899.pdf?sequence=1&isAllowed=y
6. Paarthneekhara, ”Text To Image Synthesis Using Thought Vectors”, https://github.com/paarthneekhara/text-to-image
7. Özgür Doğan, ”CNN (Convolutional Neural Networks) Nedir?”, https://teknoloji.org/cnn-convolutional-neural-networks-nedir
8. Mehmet Fatih AKCA, “RNN Nedir? Nasıl Çalışır?”, https://medium.com/deep-learning-turkiye/rnn-nedir-nasıl-çalışır-9e5d572689e1
9. Mehmet Fatih AKCA, “LSTM Nedir? Nasıl Çalışır?”, https://mfakca.medium.com/lstm-nedir-nasıl-çalışır-326866fd8869
10. Veribilimcisi.com, “Uzun / Kısa Süreli Bellek (Long / Short Term Memory)”, https://veribilimcisi.com/2017/09/26/uzun-kisa-sureli-bellek-long-short-term-memory/
11. Google, “Colab'a hoş geldiniz.”, https://colab.research.google.com/notebooks/welcome.ipynb?hl=tr
12. Randula Koralage, “How to Load a Dataset From the Google Drive to Google Colab”, https://medium.com/geekculture/how-to-load-a-dataset-from-the-google-drive-to-google-colab-67d0478bc634