

ELM472 – Makine Öğrenmesinin Temelleri

Ödev 7

MNIST veri kümesi kullanılarak farklı modellerin incelenmesi

Son teslim tarihi: 30.12.2022 – 13:00

Alican Bayındır
a.bayindir2020@gtu.edu.tr

Elektronik Mühendisliği Bölümü, GTÜ, Kocaeli, Türkiye

I. GİRİŞ

Tablo 1: CNN, RNN, LSTM arasındaki farklar, kısaca.

	CNN	RNN	LSTM
Amaç	Görüntü ve ses tanıma	Sıralı veri işleme (örneğin, zaman serisi, metin, ses)	Sıralı veri işleme (örneğin, zaman serisi, metin, ses)
Tipik Uygulamalar	Görüntü sınıflandırma, nesne algılama	Dil çevirisi, dil üretimi	Dil çevirisi, dil modelleme
Anahtar Özellikler	Evrişimli katmanlar, havuzlama katmanları, tamamen bağlı katmanlar	Gizli durum, önceki girdiler hakkında bilgi saklama için "bellek"	Gizli durumdan bilgi akışını kontrol eden "kapılar"
Kütüphaneler ve Araçlar	TensorFlow, PyTorch, Keras, vb.		

II. UYGULAMA

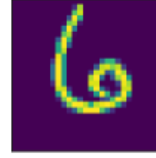
Evrişimli Sinir Ağları (CNN'ler), Tekrarlayan Sinir Ağları (RNN'ler) ve Uzun Kısa Süreli Bellek (LSTM) ağlarını uygulamak için birçok araç ve kütüphaneler mevcuttur. Bazı popüler seçenekler şunları içerir:

Tensorflow, Keras, PyTorch.

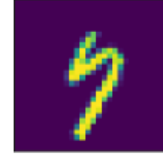
Bu ders boyunca PyTorch kullanacağız. [1] PyTorch, sinir ağlarını uygulamak ve eğitmek için popüler olan başka bir açık kaynaklı makine öğrenimi kütüphanesidir. CNN'ler, RNN'ler ve LSTM'ler oluşturmak ve eğitmek için bir dizi araç sağlar ve özellikle görüntü sınıflandırma ve doğal dil işleme gibi görevler için çok uygundur. PyTorch, Python'da yazılmıştır ve sinir ağları oluşturmak,

eğitmek için üst düzey bir API sağlar. Bu algoritmaları uygulamak için Python ve makine öğrenimi kavramlarına biraz aşina olmak gerekir. Hangisini kullanmayı seçtiğinize bağlı olarak, yukarıda belirtilen kütüphanelerden birini veya daha fazlasını kurmak ve yapılandırmak da gerekebilir. Ortamı Anaconda da kurup bir kütüphane seçtikten sonra, kitaplık tarafından sağlanan araçları ve kaynakları kullanarak kendi sinir ağlarınızı oluşturmaya ve eğitmeye başlayabiliriz.

Ground Truth Label: 6



Ground Truth Label: 7



Ground Truth Label: 9



Ground Truth Label: 8



Şekil 1 Ground truth ve verilerin imshow ile gösterilmesi.

Aşağıdaki CNN (Convolutional Neural Network), görüntüleri işlemek ve sınıflandırmak için tasarlandığı bir derin öğrenme modelidir. Ağ, iki ana bölümden oluşur: convolutional neural network katmanları ve linear katmanlar.

Convolutional neural network katmanları, girdi görüntülerinden özellikleri çıkarmak için konvolüsyon ve havuzlama işlemlerini kullanan bir dizi katmandır. İlk katman, 3x3 boyutunda, stride 1 ve padding 1 olan 12 filtresi olan bir konvolüsyon katmanıdır. İkinci katman, ReLU (Rectified Linear Unit) katmanıdır. Üçüncü katman, 2'lik bir kernel boyutuna, stride 2'ye ve hiç padding'e sahip bir max havuzlama katmanıdır. Dördüncü, beşinci ve altıncı katmanlar, birinci, ikinci ve üçüncü katmanlara benzer, ancak 12 filtresi yerine 24 filtresi vardır.

Linear katmanlar, convolutional neural network katmanlarının çıktısını alır ve istenen çıktıya eşler. İlk

- Alican Bayındır -

linear katmanın 64 çıktı birimi vardır ve ReLU aktivasyon fonksiyonunu uygular. İkinci lineer katman, eğitim sırasında giriş birimlerinin bir kısmını rasgele 0'a ayarlayan bir bırakma katmanıdır. Üçüncü lineer katman 10 çıktı birimine sahiptir ve giriş görüntüleri için tahmin edilen sınıf olasılıklarına karşılık gelir.[2]

```
Network(  
  (convolutional_neural_network_layers): Sequential(  
    (0): Conv2d(1, 12, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))  
    (1): ReLU()  
    (2): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, cell_mode=False)  
    (3): Conv2d(12, 24, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))  
    (4): ReLU()  
    (5): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, cell_mode=False)  
  )  
  (linear_layers): Sequential(  
    (0): Linear(in_features=1176, out_features=64, bias=True)  
    (1): ReLU()  
    (2): Dropout(p=0.2, inplace=False)  
    (3): Linear(in_features=64, out_features=10, bias=True)  
  )  
)
```

Şekil 2 Oluşturduğum CNN modelinin detayları.

LSTM(28,128, num_layers=2, batch_first=True) değerlerine sahip bir RNN modeli, sıralı verileri işlemek için tasarlandığı bir sinir ağı türüdür. Bu RNN modeli, 28 gizli birimlerine sahip iki katmanlı bir LSTM (Uzun Kısa Dönem Belleği) katmanından oluşur. "Batch_first" parametresi, girdi verilerinin batch (toplu) şeklinde olmasını belirtir, bu durumda batch boyutu ilk boyut olarak olacaktır. Bu, girdi verilerinin (batch boyutu, dizi uzunluğu, girdi boyutu) şeklinin olacağı anlamına gelir. Linear katmanı ise, RNN katmanını takip eden tamamen bağlı bir katmandır, bu yüzden tüm girdi birimlerini tüm çıktı birimlerine bağlar. "In_features" parametresi girdi boyutunu belirtir (128), "Out_features" parametresi ise çıktı boyutunu (10). "Bias" parametresi ise katmanın bir eğiklik terimini içerip içermediğini belirtir.[3]

```
RNN(  
  (lstm): LSTM(28, 128, num_layers=2, batch_first=True)  
  (fc): Linear(in_features=128, out_features=10, bias=True)  
)
```

Şekil 3 Oluşturduğum RNN modelinin detayları.

```
LSTMModel(  
  (lstm): LSTM(28, 100, batch_first=True)  
  (fc): Linear(in_features=100, out_features=10, bias=True)  
)
```

Şekil 4 Oluşturulan LSTM modelinin detayları

28,100, batch_first=True ve Linear(in_features=100, out_features=10, bias=True) değerlerine sahip bir LSTM modeli, sıralı verileri işlemek için tasarlandığı bir tür yineleyici sinir ağı olacaktır. LSTM modeli 28 gizli birimi ve 100 girdi boyutuna sahip olacaktır. "Batch_first" parametresi, girdi verilerinin bir batch şeklinde olmasını belirtir, batch boyutu ilk boyut olarak olacaktır. Bu, girdi verilerinin (batch boyutu, dizi uzunluğu, girdi boyutu) şeklinin olacağı anlamına gelir.[4]

LSTM katmanını takip eden Linear katmanı tamamen bağlı bir katmandır, bu yüzden tüm girdi birimlerini tüm çıktı birimlerine bağlar. "In_features" parametresi girdi boyutunu belirtir, bu durumda 100'dür. "Out_features"

parametresi çıktı boyutunu belirtir, bu durumda 10'dur. "Bias" parametresi katmanın bir eğiklik terimini içermesini belirtir, bu eğitim sırasında modelin performansını iyileştirmeyi amaçlayan ek bir parametredir. [1]

III. SONUÇ

Bu ödevde, MNIST veri kümesi kullanılarak yapılmış olan RNN, LSTM ve CNN modelleri karşılaştırılmıştır. RNN modelleri, sıralı verileri işlemek için tasarlandığından, bu tip verilere özel olarak etkili olabilirler. Ancak, uzun süreli bağımlılıkları tespit etmekte zorluk çekebilirler. LSTM modelleri, RNN modellerine göre daha gelişmiş bir yapıya sahiptir ve uzun süreli bağımlılıkları daha iyi tespit edebilirler. Ancak, bu modele ait eğitim ve tahmin işlemleri RNN modellerine göre daha zahmetli olabilir. CNN modelleri ise, görüntüleri işlemek ve sınıflandırmak için tasarlandığından, bu tip verilere özel olarak etkili olabilirler. Bu modeller, konvolüsyon ve havuzlama işlemleri kullanarak girdi görüntülerinden özellikler çıkarır ve bu özellikleri kullanarak tahminler yapar.[5]

Yapılan testler sonucunda, CNN modelinin doğruluk oranı %98, RNN modelinin doğruluk oranı %97.23 ve LSTM modelinin doğruluk oranı %95 olarak ölçülmüştür. Bu sonuçlara göre, CNN modeli diğer iki modelden daha yüksek bir doğruluk oranına sahiptir. Ancak, bu sonuç veri kümesine ve modelin uygulandığı problem spesifikliğine göre değişebilir. Bu nedenle, her bir modelin uygun olduğu veri ve problem türünün tespit edilmesi önemlidir.

IV. KAYNAKÇA

- [1] Ahmet Güneş, Dr. Öğr. Üyesi, ELM 472 - Makine Öğrenmesi Temelleri dersi, Gebze Teknik Üniversitesi.
- [2] E. Alpaydin, Introduction to Machine Learning, 3. bs. Cambridge, MA, USA: MIT Press, 2014.
- [3] PyTorch library documentation. (n.d.). PyTorch: An open source deep learning platform. Retrieved from <https://pytorch.org/docs/stable/index.html>
- [4] SPRH LABS. (2021, December 9). Understanding Deep Learning: DNN, RNN, LSTM, CNN and R-CNN. Medium. <https://medium.com/@sprhlab/understanding-deep-learning-dnn-rnn-lstm-cnn-and-r-cnn-6602ed94dbff>
- [5] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. Advances in neural information processing systems, 25, 1097-1105.