Markov Chain Monte Carlo İle Hata-Parametre Yüzeyleri

*

Muharrem Küçükyılmaz
Bilgisayar Mühendisliği
Yıldız Teknik Üniversitesi
İstanbul, Türkiye
muharrem.kucukyilmaz@gmail.com

***Abstract*—Hata-parametresi optimize edilmiş bir modelde Markov Chain Monte Carlo(MCMC) yöntemi ile hata-paremetre uzayı elde edilmiş ve bu yöntem ile daha başarılı model elde etmek amaçlanmıştır.**

***Index Terms*—Markov Chain Monte Carlo, MCMC, Hata-parametre uzayı, Metropolis**

I. Giriş

Makine Öğrenmesi/Derin öğrenme modellerin başarısının artmasıyla birlikte bir sistemin işleyebilmesi için algoritmalar yerine modeller geliştirilmeye başlanmıştır. Modeller eğitilirken hata-parametre uzayında çok nadir global minimum noktaya ulaşmakla birlikte genellikle lokal minimum noktasına yakınsarlar. Ama çoğu kez tam lokal minimum noktasına ulaşamazlar. Bu çalışmada Lokal minimuma yakınsayan modelin MCMC ile hata-parametre uzayı incelenerek tam lokal minimum noktasını bulmak ve böylece model başarısını arttırmak amaçlanmıştır.

II. Veri Kümesi

UCI veri kümesinden Tablo I'deki waveform, vowel, vehicle, splice ve zoo veri kümeleri seçilmiştir.

TABLE I: UCI Veri Kümesi

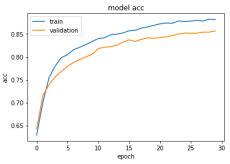
Table Head	**Veri Kümesi**				
	waveform	***vowel***	***vehicle***	***splice***	***zoo***
Özellik	40	11	18	288	16
Örnek sayısı	5000	990	846	3190	84

III. Model

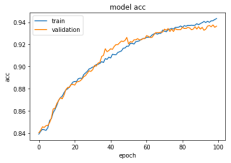
Bu projede tek gizli katmanlı figure 1'deki sequential model kullanılmıştır. Gizli katman 64 düğümden oluşmaktadır. Model ağırlık değerleri -5 ile +5 arasında rastgele olarak başlatılmıştır.Aktivasyon fonksiyonu için gizli katmanda sigmoid, çıkış katmanında softmax kullanılmıştır.Optimizer için Öğrenme katsayısı 0.001 olan Adam Optimizer tercih edilmiştir.

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense (Dense)                (None, 64)                2624
_________________________________________________________________
dense_1 (Dense)              (None, 3)                 195
=================================================================
Total params: 2,819
Trainable params: 2,819
Non-trainable params: 0
```

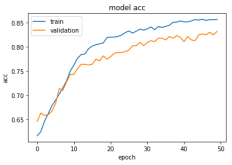
Fig. 1: Model



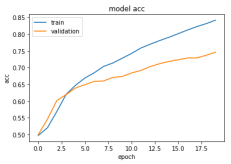
(a) waveform acc



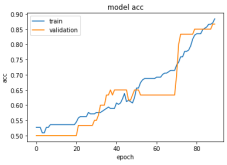
(b) vovel acc



(c) vehicle acc



(d) splice acc



(e) zoo acc

Fig. 2: Acc grafikleri

A. Model Başarıları

5 farklı veri kümesi ile model eğitilmiştir. Doğruluk-iterasyon grafikleri figure 2'deki gibidir.

IV. Metropolis Hasting Algoritmasi

İstatistik de , Metropolis–Hastings Algoritması direkt örneklenmesi zor olasılıksal dağılımdan rastgele noktalar serisi elde etme metodu olan Markov Chain Monte Carlo algoritmasıdır.Metropolis algoritması ise Metropolis-Hasting algoritmasının olaslıksal dağılımın simetrik olduğu özel bir durumudur.

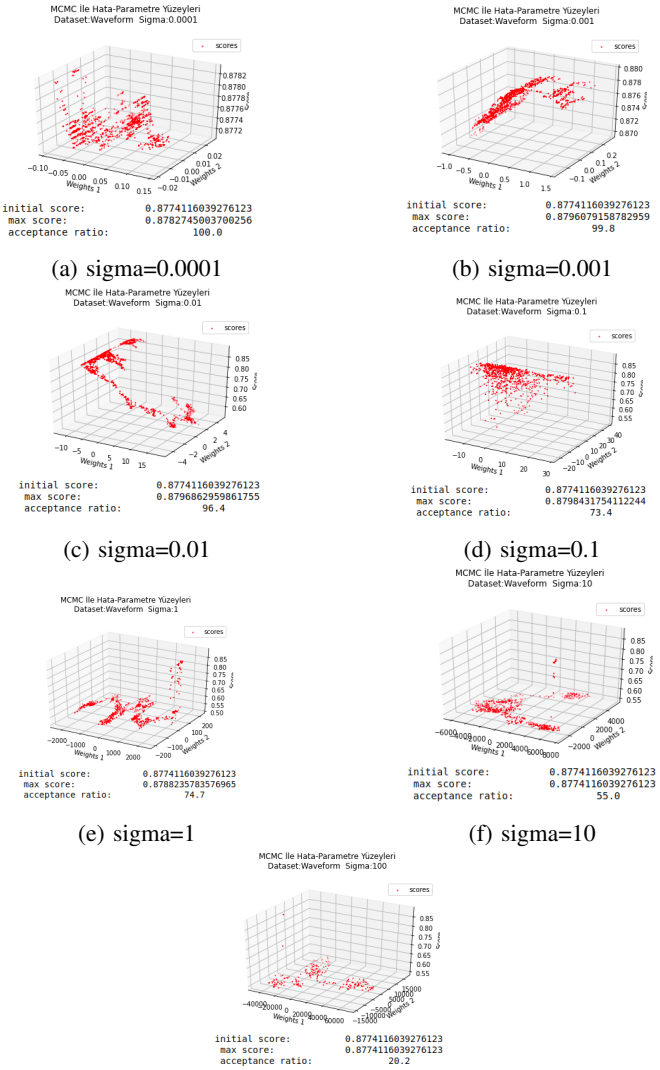
A. Metropolis Algotitması

$f(x)$, istenilen olasılıksal dağılım $P(x)$ 'e orantılı bir fonksiyon olsun. Algoritma adımları aşağıdaki gibidir.

- simetrik olasılık dağılımı seç.
- başlangıç noktasını tanımla.
- her iterasyon t de:
  - olasılık dağılımı $g(x^{'}|x_t)$ 'ndan yeni aday nokta $x^{'}$ üret.
  - kabul edilme oranını $\alpha = \frac{P(x^{'})}{P(x_t)}$ hesapla.
  - Kabul et / Reddet aşamasında;
    - rastgele sayı üret. $u \in [0, 1]$
    - eğer $u \leq \alpha$ ise kabul et ve $x_{t+1} = x^{'}$ olarak ayarla.
    - eğer $u > \alpha$ ise reddet ve $x_{t+1} = x_t$ olarak ayarla.

V. Deneysel Analiz

Metropolis algoriması için, model eğitildikten sonra hatanın en az olduğu son ağırlık değerleri başlangıç noktası olarak tanımlanmıştır. Olasılıksal dağılım için normal dağılım seçilmiş ve farklı sigma değerlerine göre hata-parametre yüzeyleri elde edilmiştir. Kabul et/reddet aşamasında α 1 den büyükse kabul edilmiş, küçükse $\%(\alpha)^n$ ihtimalle kabul edilmiştir.(α)'nın üssel değerini alma sebebimiz α'nın 1'den küçük olduğu durumlarda kabul edilme ihtimalini azaltmaktır. figure 3'de waveform veri kümesinin hata-parametre yüzeyi görülmetedir. Tablo II'de de görüldüğü gibi sigma arttıkça kabul edilebilirlik oranı azalmaktadır.Çünkü random seçtiğimiz yeni nokta önceki noktaya göre mesafesi arttığı için başarı farkı da artmaktadır.Bu iki başarıyı oranladığımızda da fark arttıkça alphanın küçüldüğünü görürüz. Ayrıca tablodan eğitilmiş modelin lokal minimuma yakınsadığını ama tam lokal minimum noktasına ulaşamadığını görüyoruz. Bunun sebebi de adam optimizer da accuracy belli eşik değerinin altındaysa model sonlandırıyoruz. yani tam lokal minimuma ulaşmıyoruz.



(a) sigma=0.0001 (b) sigma=0.001

(c) sigma=0.01 (d) sigma=0.1

(e) sigma=1 (f) sigma=10

(g) sigma=100

Fig. 3: waveform hata-parametre yüzeyi

VI. Sonuç

Bu projede eğitilmiş bir modelin çevresinde model başarısını arttıracak ağırlık değerlerinin olup olmadığı metropolis algoritması ile araştırılmıştır. Tablo II' de görüldüğü gibi çok az da olsa başarının arttığı gözlemlenmiştir.

VII. Kod

Kod: https://github.com/Mky07/mcmc_ile_hata_parametre_uzayi_kesfi.git

TABLE II: Sigma-Kabul Oranı İlişkisi

	sigma	*başlangıç başarı*	*max. başarı*	*kabul oranı%*
waveform	0.0001	0.877411	0.878274	100
	0.001	0.877411	0.879607	99.8
	0.01	0.877411	0.879686	96.4
	0.1	0.877411	0.879843	73.4
	1	0.877411	0.878823	74.7
	10	0.877411	-	55
	100	0.877411	-	20.2
vowel	0.0001	0.941844	0.942060	100
	0.001	0.941844	0.944222	99.9
	0.01	0.941844	0.943778	99.1
	0.1	0.941844	0.944006	98.1
	1	0.941844	-	98.6
	10	0.941844	-	99.7
	100	0.941844	-	99.6
vehicle	0.0001	0.854311	0.855006	100
	0.001	0.854311	0.856050	99.6
	0.01	0.854311	0.855702	96.5
	0.1	0.854311	0.855006	73.3
	1	0.854311	-	100
	10	0.854311	-	94.4
	100	0.854311	-	96.2
splice	0.0001	0.828599	0.829337	100
	0.001	0.828599	0.829337	99.8
	0.01	0.828599	0.829337	93.2
	0.1	0.828599	0.829337	74.2
	1	0.828599	-	39.2
	10	0.828599	0.829336	56.2
	100	0.828599	-	74.4
zoo	0.0001	0.880281	-	100
	0.001	0.880281	0.883802	100
	0.01	0.887323		96.8
	0.1	0.880281	0.887323	78.4
	1	0.880281	-	89.6
	10	0.880281	-	94.3
	100	0.880281	-	94.7