

Veri Bağlanımı için Yüksek Verimli Yinelemeli Sinir Ağı Yapısı

A Highly Efficient Recurrent Neural Network Architecture for Data Regression

Tolga Ergen ve Emir Ceyani
Elektrik ve Elektronik Mühendisliği Bölümü
İhsan Doğramacı Bilkent Üniversitesi
Ankara, Türkiye
{ergen,ceyani}@ee.bilkent.edu.tr

Özetçe —Bu bildiride, çevrimiçi doğrusal olmayan veri bağlanım problemi çalışmakta ve uzun kısa soluklu bellek (UKSB) ağları merkezli yüksek verimli bir yapısı sunulmaktadır. Ayrıca, önerilen yapının parametrelerini öğrenmek için çevrimiçi öğrenme algoritmaları sunulmaktadır. Önce, veri bağlanımı için UKSB merkezli bir yapı kullanılmaktadır. UKSB ağlarının karmaşıklığını düşürmek için, matris çarpımları yüksek verimli bir işlem (ef işlemi) ile değiştirilmektedir. Sonra, öğrenilmesi gereken parametre sayısını düşürmek için UKSB ağının ağırlık matrislerine ayrıştırma yapılmaktadır. Sunulan yapıların parametrelerini öğrenmek için Üstel Gradyan (ÜD) ve Olasılıksal Bayır İnme (OBİ) merkezli çevrimiçi öğrenme algoritmaları sunulmaktadır. Deney sonuçları sunulan yapı tarafından sağlanan önemli ölçüdeki performans ve verimlilik kazanımlarını göstermektedir.

Anahtar Kelimeler—uzun kısa soluklu bellek ağı, gradyan inişi, üstel gradyan, ef işlemi, matris ayrıştırması.

Abstract—In this paper, we study online nonlinear data regression and propose a highly efficient long short term memory (LSTM) network based architecture. Here, we also introduce online training algorithms to learn the parameters of the introduced architecture. We first propose an LSTM based architecture for data regression. To diminish the complexity of this architecture, we use an energy efficient operator (ef-operator) instead of the multiplication operation. We then factorize the matrices of the LSTM network to reduce the total number of parameters to be learned. In order to train the parameters of this structure, we introduce online learning methods based on the exponentiated gradient (EG) and stochastic gradient descent (SGD) algorithms. Experimental results demonstrate considerable performance and efficiency improvements provided by the introduced architecture.

Keywords—long short term memory network, gradient descent, exponentiated gradient, ef-operator, matrix factorization.

I. GİRİŞ

Yinelemeli sinir ağları (YSA) güçlü modelleme kapasitelerinden ötürü literatürde başlıca üzerinde durulan konulardan biridir [1], [2]. Fakat, geleneksel YSA yapıları bilgi akışını

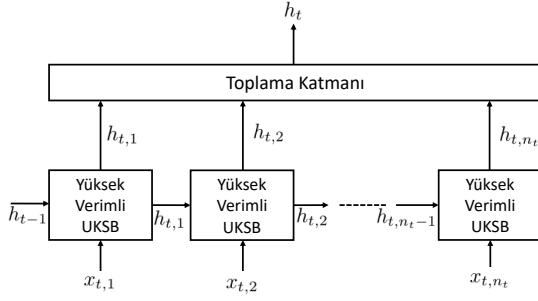
kontrol edemediklerinden dolayı, öğrenme esnasında parametrelerin gradyanlarının normundaki değişimlere karşı oldukça hassastır [3]. Bu yüzden bu yapılar, verideki uzun soluklu bağlantıları yakalamakta güçlük çekmektedir [3]. Bu problemleri çözmek amacıyla, bilgi akışını çeşitli bilgi kapıları ile kontrol edebilen gelişmiş bir YSA yapısı, uzun kısa soluklu bellek (UKSB) sunulmaktadır. [4]. Ancak bu kapıların bilgi kontrolü üzerindeki avantajı UKSB yapılarının eğitimi zorlaştırmış ve karmaşıklığını artırmıştır [4].

Literatürde, UKSB ağları için çeşitli çevrimiçi öğrenme yöntemleri bulunmaktadır [2], [5]. Bu metodlar arasında toplama işlemleri ile parametreleri güncelleyen olasılıksal bayır inme (OBİ) algoritması en yaygın olanıdır [2]. Ancak bu algoritmalar yavaş yakınsama oranı ve düşük performansa sahiptir. Bu sorunları çözmek için, Üstel Gradyan (ÜG) algoritması sunulmaktadır. [6]. Ancak, ÜG algoritması çarpımsal güncellemeler yaptığından dolayı daha fazla işlem yapmaktadır [6], [7].

Yukarıda anlatılan problemler, özgün ve yüksek verimli bir UKSB ağı ile ÜG [6] ve OBİ [2] merkezli çevrimiçi öğrenme algoritmalarıyla çözülmektedir. Öncelikle, değişken uzunluktaki girdileri işlemek için UKSB merkezli bağlanım yapısı sunulmaktadır. Daha sonra, UKSB ağlarında hesaplama karmaşıklığını azaltmak için matris çarpımına kıyasla daha az enerji kullanan bir işlem kullanılmaktadır. Karmaşıklığı ve öğrenilecek parametre sayısını daha da düşürmek amacıyla, UKSB parametrelerine matris ayrıştırması yöntemi [8] uygulanmaktadır. Böylece, literatürdeki diğer metodların [2], [6] aksine, UKSB ağlarının yüksek performansı düşük karmaşıklıkta bir öğrenme algoritması ile elde edilebilmektedir. Simülasyonlar sonucu geleneksel yöntemlere kıyasla önemli derecede performans kazanımı ve karmaşıklık düşüşü gözlemlenmektedir.

II. MODEL VE PROBLEM TANIMI

Bu bildirideki problemde, ardışık olarak $\{d_t\}_{t \geq 1}$, $d_t \in \mathbb{R}$ istenen sinyali ve matrisler $\{\mathbf{X}_t\}_{t \geq 1}$, $\mathbf{X}_t = [\mathbf{x}_{t,1}, \mathbf{x}_{t,2}, \dots, \mathbf{x}_{t,n_t}]$, $\mathbf{x}_{t,j} \in \mathbb{R}^p, \forall j \in \{1, 2, \dots, n_t\}$, alınmaktadır ve $n_t \in \mathbb{Z}^+$ zaman indisi t ile değişebilen \mathbf{X}_t matrisinin sütun sayısıdır. Bu problemde, amaç $\{d_t\}_{t \geq 1}$ ve



Şekil 1: Yüksek Verimli UKSB yapısının detaylı şeması

vektör dizisi \mathbf{X}_t arasında bir ilişki bulmaktır. Bu ilişkiyi bulmak için, her bir zaman indisi t 'de \mathbf{X}_t elde edildikten sonra şuanki ve geçmişteki gözlemlere dayanarak bir tahmin, \hat{d}_t , elde edilir. Buna ek olarak, her bir zaman indisi t 'de hata fonksiyonu $L(\hat{d}_t, d_t)$ hesaplanır. Bu bildiride, \hat{d}_t 'yi elde etmek için UKSB ağları kullanılmaktadır. Girdi değişken uzunlukta olduğundan ötürü UKSB \mathbf{X}_t matrisinin her bir sütununa uygulanmaktadır. Bu ağın yapısı aşağıdaki denklemler ile açıklanmaktadır [1], [4]:

$$\tilde{c}_{t,j} = g(\mathbf{W}^{(\tilde{c})} \mathbf{x}_{t,j} + \mathbf{R}^{(\tilde{c})} \mathbf{h}_{t,j-1} + \mathbf{b}^{(\tilde{c})}) \quad (1)$$

$$\mathbf{i}_{t,j} = \sigma(\mathbf{W}^{(i)} \mathbf{x}_{t,j} + \mathbf{R}^{(i)} \mathbf{h}_{t,j-1} + \mathbf{b}^{(i)}) \quad (2)$$

$$\mathbf{f}_{t,j} = \sigma(\mathbf{W}^{(f)} \mathbf{x}_{t,j} + \mathbf{R}^{(f)} \mathbf{h}_{t,j-1} + \mathbf{b}^{(f)}) \quad (3)$$

$$\mathbf{c}_{t,j} = \mathbf{D}_{t,j}^{(i)} \tilde{c}_{t,j} + \mathbf{D}_{t,j}^{(f)} \mathbf{c}_{t,j-1} \quad (4)$$

$$\mathbf{o}_{t,j} = \sigma(\mathbf{W}^{(o)} \mathbf{x}_{t,j} + \mathbf{R}^{(o)} \mathbf{h}_{t,j-1} + \mathbf{b}^{(o)}) \quad (5)$$

$$\mathbf{h}_{t,j} = \mathbf{D}_{t,j}^{(o)} g(\mathbf{c}_{t,j}), \quad (6)$$

burada $\mathbf{c}_{t,j} \in \mathbb{R}^m$ durum vektörünü, $\mathbf{x}_{t,j} \in \mathbb{R}^p$ giriş vektörünü ve $\mathbf{h}_{t,j} \in \mathbb{R}^m$ çıkış vektörünü temsil etmektedir. Ayrıca, $\mathbf{i}_{t,j}$, $\mathbf{f}_{t,j}$ ve $\mathbf{o}_{t,j}$ sırasıyla giriş, unutma, ve çıkış kapılarıdır. $g(\cdot)$ uygulandıkları vektörlerin her bir elemanını ayrı ayrı işlemekte ve genellikle $\tanh(\cdot)$ fonksiyonu olarak seçilmektedir. Benzer bir fonksiyon olan $\sigma(\cdot)$ ise sigmoid fonksiyonu olarak seçilmektedir ve uygulandıkları vektörlerin her bir elemanını ayrı ayrı işlemektedir. $\mathbf{D}_{t,j}^{(i)} = \text{diag}(\mathbf{i}_{t,j})$, $\mathbf{D}_{t,j}^{(f)} = \text{diag}(\mathbf{f}_{t,j})$ ve $\mathbf{D}_{t,j}^{(o)} = \text{diag}(\mathbf{o}_{t,j})$ şeklinde tanımlanırken, $\text{diag}(\cdot)$ girdi vektörünü matrisin ana köşegenine yerleştirecek şekilde matris oluşturur. Geriye kalan matris ve vektörlerin boyutları girdi ve çıktı vektörlerinin boyutlarına göre belirlenmektedir. Şekil 1'de UKSB ağının tüm sütunlara uygulanmasından sonra sabit uzunlukta çıkış vektörü elde etmek için tüm çıktıların ortalaması alınır [9]. Bu vektör $\mathbf{h}_t \in \mathbb{R}^m$ ile ifade edilmekte ve tahmin vektörlerinin, $\hat{d}_t = \mathbf{w}_t^T \mathbf{h}_t$, oluşturulmasında kullanılmaktadır. Burada $\mathbf{w}_t \in \mathbb{R}^m$ zaman indisi t için bağlanım katsayılarını temsil etmektedir. Bu sistemde amaç parametreleri $\sum_{i=1}^t L(\hat{d}_i, d_i)$ olarak tanımlanan toplam hata fonksiyonunu en küçük değere indirecek şekilde çevrimiçi olarak öğrenmektir.

III. YÜKSEK VERİMLİ YİNELEMELİ SINIR AĞLARI İLE ÇEVİRİMİÇİ ÖĞRENME

Bu bölümde, önce ef işlemi UKSB yapısına uygulanmaktadır. Daha sonra, ef işlemine ilaveten matris ayrıştırması yöntemi ile yüksek verimli UKSB yapısı sunulmaktadır. Son olarak OBİ ve ÜG merkezli çevrimiçi öğrenme algoritmaları sunulmaktadır.

A. Ef İşlemi ile Yinelemeli Sinir Ağları

Bu bölümde, UKSB yapısının denklemleri ef işlemi uygulanarak yeniden tanımlanmaktadır. Herhangi bir $\mathbf{a}, \mathbf{b} \in \mathbb{R}^p$, vektörleri olsun ve bu iki vektör için ef işlemi [7] şu şekilde tanımlıdır:

$$\mathbf{a} \diamond \mathbf{b} := \sum_{i=1}^p \text{sign}(a_i) b_i + \text{sign}(b_i) a_i. \quad (7)$$

Burada $\text{sign}(\cdot)$ fonksiyonu girdinin işaretini çıktı olarak vermektedir. Ef işlemi sadece toplama ve işaret işlemlerini kullandığından ötürü matris çarpımına göre daha verimlidir.

Geleneksel UKSB ağındaki matris çarpımları ef işlemi ile değiştirildiğinde, denklem (1)-(6) aşağıda şu şekilde tanımlanır:

$$\tilde{c}_{t,j} = g(\mathbf{a}_{\tilde{c}} \odot (\mathbf{W}^{(\tilde{c})} \diamond \mathbf{x}_{t,j}) + \mathbf{b}_{\tilde{c}} \odot (\mathbf{R}^{(\tilde{c})} \diamond \mathbf{h}_{t,j-1}) + \mathbf{b}^{(\tilde{c})}) \quad (8)$$

$$\mathbf{i}_{t,j} = \sigma(\mathbf{a}_i \odot (\mathbf{W}^{(i)} \diamond \mathbf{x}_{t,j}) + \mathbf{b}_i \odot (\mathbf{R}^{(i)} \diamond \mathbf{h}_{t,j-1}) + \mathbf{b}^{(i)}) \quad (9)$$

$$\mathbf{f}_{t,j} = \sigma(\mathbf{a}_f \odot (\mathbf{W}^{(f)} \diamond \mathbf{x}_{t,j}) + \mathbf{b}_f \odot (\mathbf{R}^{(f)} \diamond \mathbf{h}_{t,j-1}) + \mathbf{b}^{(f)}) \quad (10)$$

$$\mathbf{c}_{t,j} = \mathbf{i}_{t,j} \blacklozenge \tilde{c}_{t,j} + \mathbf{f}_{t,j} \blacklozenge \mathbf{c}_{t,j-1} \quad (11)$$

$$\mathbf{o}_{t,j} = \sigma(\mathbf{a}_o \odot (\mathbf{W}^{(o)} \diamond \mathbf{x}_{t,j}) + \mathbf{b}_o \odot (\mathbf{R}^{(o)} \diamond \mathbf{h}_{t,j-1}) + \mathbf{b}^{(o)}) \quad (12)$$

$$\mathbf{h}_{t,j} = \mathbf{o}_{t,j} \blacklozenge g(\mathbf{c}_{t,j}). \quad (13)$$

Burada $\mathbf{a}_{(\cdot)}, \mathbf{b}_{(\cdot)} \in \mathbb{R}^m$ ölçekleme katsayılarıdır ve \blacklozenge işlemi aşağıdaki gibi tanımlanmaktadır:

$$\begin{aligned} \mathbf{a} \blacklozenge \mathbf{b} &:= [\mathbf{a}_1 \diamond \mathbf{b}_1 \ \mathbf{a}_2 \diamond \mathbf{b}_2 \ \dots \ \mathbf{a}_p \diamond \mathbf{b}_p]^T \\ &:= \text{sign}(\mathbf{a}) \odot \mathbf{b} + \text{sign}(\mathbf{b}) \odot \mathbf{a}. \end{aligned}$$

Denklem (8)'de, $\mathbf{W}^{(\tilde{c})} \diamond \mathbf{x}_{t,j}$ şu şekilde yazılır:

$$\mathbf{W}^{(\tilde{c})} \diamond \mathbf{x}_{t,j} = [\mathbf{w}_1^{(\tilde{c})} \diamond \mathbf{x}_{t,j} \ \mathbf{w}_2^{(\tilde{c})} \diamond \mathbf{x}_{t,j} \ \dots \ \mathbf{w}_m^{(\tilde{c})} \diamond \mathbf{x}_{t,j}]^T, \quad (14)$$

burada $\mathbf{w}_i^{(\tilde{c})} \diamond \mathbf{x}_{t,j}$ şu şekilde verilmektedir:

$$\mathbf{w}_i^{(\tilde{c})} \diamond \mathbf{x}_{t,j} = \sum_{k=1}^p \text{sign}(x_{t,jk}) w_{ik}^{(\tilde{c})} + \text{sign}(w_{ik}^{(\tilde{c})}) x_{t,jk}$$

ve

$$\mathbf{R}^{(\tilde{c})} \diamond \mathbf{h}_{t,j-1} = [\mathbf{r}_1^{(\tilde{c})} \diamond \mathbf{h}_{t,j-1} \ \mathbf{r}_2^{(\tilde{c})} \diamond \mathbf{h}_{t,j-1} \ \dots \ \mathbf{r}_m^{(\tilde{c})} \diamond \mathbf{h}_{t,j-1}]^T, \quad (15)$$

burada $\mathbf{r}_i^{(\tilde{c})} \diamond \mathbf{h}_{t,j}$ şu şekilde verilmiştir:

$$\mathbf{r}_i^{(\tilde{c})} \diamond \mathbf{h}_{t,j} = \sum_{k=1}^m \text{sign}(h_{t,jk}) r_{ik}^{(\tilde{c})} + \text{sign}(r_{ik}^{(\tilde{c})}) h_{t,jk},$$

burada $\mathbf{w}_i^{(\tilde{c})}$ ve $\mathbf{r}_i^{(\tilde{c})}$ sırasıyla $\mathbf{W}^{(\tilde{c})}$ ve $\mathbf{R}^{(\tilde{c})}$ 'ın i . satırını temsil etmektedir. Diğer çarpım işlemleri için, denklem (14) ve (15)'de gerekli katsayı matrislerine göre parametrelerinin değişmesi yeterlidir.

Açıklama 1. Geleneksel UKSB ağı ile karşılaştırıldığında, Denklem (1) - (6)'daki $4m(m+p) + 3m$ adet çarpma işlemi işaret çarpma ve toplama işlemleriyle değiştirilmektedir. Ancak, denklem (8) - (13)'da ölçekleme katsayıları $8m$ daha fazla çarpma işlemi oluşturmaktadır. Genelde, oldukça büyük m ve p değerler alındığında $8m \ll 4m(m+p) + 3m$ olduğu görülmektedir. Böylece, UKSB ağlarındaki çarpım sayısı ve hesaplama karmaşıklığı büyük bir ölçüde düşürülmektedir.

B. Ağırlık Matrisi Ayırıştırması ile Yüksek Verimli Yinelemeli Sinir Ağları

UKSB ağı matrisleri iki tane düşük dereceli matrisin çarpımı, $\mathbf{W}^{(\cdot)} \approx \mathbf{M}^{(\cdot)}\mathbf{N}^{(\cdot)}$ ve $\mathbf{R}^{(\cdot)} \approx \mathbf{P}^{(\cdot)}\mathbf{Q}^{(\cdot)}$, şeklinde ayrıştırılmaktadır. Burada matrisler $\mathbf{W}^{(\cdot)} \in \mathbb{R}^{m \times p}$, $\mathbf{M}^{(\cdot)} \in \mathbb{R}^{m \times d}$, $\mathbf{N}^{(\cdot)} \in \mathbb{R}^{d \times p}$, $\mathbf{R}^{(\cdot)} \in \mathbb{R}^{m \times m}$, $\mathbf{P}^{(\cdot)} \in \mathbb{R}^{m \times f}$ ve $\mathbf{Q}^{(\cdot)} \in \mathbb{R}^{f \times m}$ olarak tanımlanmakta ve $d, f \ll \min(p, m)$ koşulunu sağlamaktadır. Bu ayrıştırmayı UKSB ağırlık katsayılarına uygulayarak (8)-(13)'daki matrisler ayrıştırılmış halleri ile değiştirilmektedir. Şekil 1'deki j 'inci UKSB yapısına yapılan değişiklikler aşağıdaki gibidir:

$$\mathbf{M}^{(\tilde{c})} \diamond \mathbf{N}^{(\tilde{c})} \diamond \mathbf{x}_{t,j} = [\mu_1^{(\tilde{c})} \diamond \mathbf{x}_{t,j} \mu_2^{(\tilde{c})} \diamond \mathbf{x}_{t,j} \dots \mu_m^{(\tilde{c})} \diamond \mathbf{x}_{t,j}]^T, \quad (16)$$

burada $\mu_i^{(\tilde{c})} \in \mathbb{R}^p$, $\mathbf{M}^{(\tilde{c})} \diamond \mathbf{N}^{(\tilde{c})}$ çıktısının i . satırıdır ve $\mu_i^{(\tilde{c})} \diamond \mathbf{x}_{t,j}$ şöyle ifade edilmektedir:

$$\mu_i^{(\tilde{c})} \diamond \mathbf{x}_{t,j} = \sum_{k=1}^p \text{sign}(x_{t,jk}) \mu_{ik}^{(\tilde{c})} + \text{sign}(\mu_{ik}^{(\tilde{c})}) x_{t,jk}.$$

ve

$$\mathbf{P}^{(\tilde{c})} \diamond \mathbf{Q}^{(\tilde{c})} \diamond \mathbf{h}_{t,j-1} = [\nu_1^{(\tilde{c})} \diamond \mathbf{h}_{t,j-1} \nu_2^{(\tilde{c})} \diamond \mathbf{h}_{t,j-1} \dots \nu_m^{(\tilde{c})} \diamond \mathbf{h}_{t,j-1}]^T, \quad (17)$$

burada $\nu_i^{(\tilde{c})} \in \mathbb{R}^m$ $\mathbf{P}^{(\tilde{c})} \diamond \mathbf{Q}^{(\tilde{c})}$ çıktısının i . satırıdır ve $\nu_i^{(\tilde{c})} \diamond \mathbf{h}_{t,j}$ aşağıdaki gibi ifade edilmektedir:

$$\nu_i^{(\tilde{c})} \diamond \mathbf{h}_{t,j} = \sum_{k=1}^m \text{sign}(h_{t,jk}) \nu_{ik}^{(\tilde{c})} + \text{sign}(\nu_{ik}^{(\tilde{c})}) h_{t,jk}.$$

Diğer ağırlık matrisleri için, seçilen matrisin ayrıştırılmış hali (16) ve (17)'de kullanılmaktadır. Daha sonra, (16) ve (17)'daki işlemler tekrar edilmektedir.

Açıklama 2. Matris ayırıştırması ile UKSB ağına toplam parametre sayısı düşürülmüştür. Denklem (1)-(6)'daki ağırlık matrislerinde, $\mathbf{W}^{(\cdot)}$ ve $\mathbf{R}^{(\cdot)}$, $4m(m+p)$ skaler parametre bulunmaktadır. Ancak, yüksek verimli UKSB ağına $4d(m+p) + 8mf$ skaler parametre bulunmaktadır. Koşul, $d, f \ll \min(m, p)$, sağlandığı sürece yüksek verimli UKSB ağı çok daha az parametreye sahiptir.

C. Çevrimiçi Öğrenme Algoritmaları

Bu bölümde, yüksek verimli ağların parametreleri öğrenmek için OBİ ve ÜG merkezli çevrimiçi güncellemeler türetilmiştir. Öncelikle, her parametre için çevrimiçi güncelleme OBİ algoritması ile elde edilmektedir [2]. Anlık hata, $L(\hat{d}_t, d_t)$, notasyon kolaylığı açısından L ile ifade edilmektedir. \mathbf{w}_t için OBİ güncellemesi aşağıdaki gibi hesaplanmaktadır:

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \eta_t \nabla \mathbf{w}_t L, \quad (18)$$

burada $\nabla \mathbf{w}_t$, hata fonksiyonunun \mathbf{w}_t ye göre gradyanını belirtmektedir. Öğrenme hızı η_t ile ifade edilmiştir. OBİ güncellemeleri eklemeli güncellemelerdir çünkü gradyan bilgisi her adımda eklenmektedir (18). Diğer bir yandan \mathbf{w}_t için ÜG güncellemesi [6] aşağıdaki gibi hesaplanmaktadır:

$$\mathbf{w}_{t+1,i} = \frac{w_{t,i} r_{t,i}}{\sum_{j=1}^m w_{t,j} r_{t,j}}, \quad (19)$$

burada $r_{t,i} = \exp(-\eta_t L'_{w_{t,i}})$ iken, $w_{t,i}$ ve $L'_{w_{t,i}}$ sırasıyla \mathbf{w}_t 'nin i . elemanı ve anlık hata fonksiyonunun $w_{t,i}$ 'ye göre kısmi türevi olarak tanımlanmaktadır. Denklem (19)'da görüldüğü üzere, ÜG güncellemeleri çarpımlı güncellemelerdir çünkü gradyan bilgisi her adımda çarpılmaktadır. Denklem (19)'da çarpma ve üs almak yerine ef işleminin birinci dereceden Taylor serisi açılımı kullanılarak aşağıdaki güncelleme elde edilmektedir:

$$w_{t+1,i} = \frac{w_{t,i} \diamond \hat{r}_{t,i}}{\sum_{j=1}^m (w_{t,j} \diamond \hat{r}_{t,j})}, \quad (20)$$

burada $\hat{r}_{t,i} = 1 - \eta_t L'_{w_{t,i}}$ olarak ifade edilmektedir. Denklem (20)'deki bölen tüm i değerleri için aynı olduğundan bu işlem normal çarpma işleminden daha az işlem yapmaktadır. Hata fonksiyonunun UKSB parametrelerine göre OBİ ve EG algoritmaları için birinci dereceden gradyanları türetilmiştir. Türevler sunum ve notasyon kolaylığı açısından $n_t = 1$ durumu için türetilmiştir. Önce hata fonksiyonunun $w_{ij}^{(\tilde{c})}$ 'ye göre türevi şu şekilde hesaplanmaktadır:

$$\begin{aligned} \frac{\partial L}{\partial w_{ij}^{(\tilde{c})}} &= \frac{\partial L}{\partial \hat{d}_t} \frac{\partial \hat{d}_t}{\partial \mathbf{h}_{t,1}} \frac{\partial \mathbf{h}_{t,1}}{\partial w_{ij}^{(\tilde{c})}} \\ &= -2(d_t - \hat{d}_t) \mathbf{w}_t^T \frac{\partial(\mathbf{o}_{t,1} \diamond \mathbf{g}(\mathbf{c}_{t,1}))}{\partial w_{ij}^{(\tilde{c})}}. \end{aligned} \quad (21)$$

Denklem (21)'deki kısmi türev aşağıdaki gibi hesaplanmaktadır:

$$\begin{aligned} \frac{\partial(\mathbf{o}_{t,1} \diamond \mathbf{g}(\mathbf{c}_{t,1}))}{\partial w_{ij}^{(\tilde{c})}} &= \frac{\partial \mathbf{o}_{t,1}}{\partial w_{ij}^{(\tilde{c})}} \odot \text{sign}(\mathbf{g}(\mathbf{c}_{t,1})) \\ &+ \mathbf{o}_{t,1} \odot 2\delta(\mathbf{g}(\mathbf{c}_{t,1})) \odot \mathbf{g}'(\mathbf{c}_{t,1}) \odot \frac{\partial \mathbf{c}_{t,1}}{\partial w_{ij}^{(\tilde{c})}} \\ &+ 2\delta(\mathbf{o}_{t,1}) \odot \frac{\partial \mathbf{o}_{t,1}}{\partial w_{ij}^{(\tilde{c})}} \odot \mathbf{g}(\mathbf{c}_{t,1}) \\ &+ \text{sign}(\mathbf{o}_{t,1}) \odot \mathbf{g}'(\mathbf{c}_{t,1}) \odot \frac{\partial \mathbf{c}_{t,1}}{\partial w_{ij}^{(\tilde{c})}}. \end{aligned} \quad (22)$$

Denklem (22) için, $\mathbf{o}_{t,1}$ ve $\mathbf{c}_{t,1}$ 'nin $w_{ij}^{(\tilde{c})}$ 'ye göre türevleri hesaplanır. $\lambda_{t-1}^{(R^{(o)h})} = \partial(\mathbf{R}^{(o)} \diamond \mathbf{h}_{t-1,1}) / \partial w_{ij}^{(\tilde{c})}$ varsayımıyla (12)'nin türevi aşağıdaki gibidir:

$$\frac{\partial \mathbf{o}_{t,1}}{\partial w_{ij}^{(\tilde{c})}} = \mathbf{D}_{t,1}^{(\sigma'(\zeta^{(o)}))} (\mathbf{b}_o \odot \lambda_{t-1}^{(R^{(o)h})}). \quad (23)$$

Burada

$$\zeta_{t,1}^{(o)} = \mathbf{a}_o \odot (\mathbf{W}^{(o)} \diamond \mathbf{x}_{t,1}) + \mathbf{b}_o \odot (\mathbf{R}^{(o)} \diamond \mathbf{h}_{t-1,1}) + \mathbf{b}^{(o)} \quad (24)$$

varsayılmaktadır. (22)'yi, hesaplayabilmek için, $\mathbf{c}_{t,1}$ 'nin $w_{ij}^{(\tilde{c})}$ 'ye göre türevi hesaplanmaktadır. Bu türev için, (11)'den

aşağıdaki gibi bir özyinelemeli ilişki elde edilmektedir:

$$\begin{aligned} \frac{\partial c_{t,1}}{\partial w_{ij}^{(\tilde{c})}} &= \text{sign}(\tilde{c}_{t,1}) \odot \frac{\partial i_{t,1}}{\partial w_{ij}^{(\tilde{c})}} + 2\delta(\tilde{c}_{t,1}) \odot \frac{\partial \tilde{c}_{t,1}}{\partial w_{ij}^{(\tilde{c})}} \odot i_{t,1} \\ &+ \text{sign}(i_{t,1}) \odot \frac{\partial \tilde{c}_{t,1}}{\partial w_{ij}^{(\tilde{c})}} + 2\delta(i_{t,1}) \odot \frac{\partial i_{t,1}}{\partial w_{ij}^{(\tilde{c})}} \odot \tilde{c}_{t,1} \\ &+ \text{sign}(c_{t-1,1}) \odot \frac{\partial f_{t,1}}{\partial w_{ij}^{(\tilde{c})}} + 2\delta(c_{t-1,1}) \odot \frac{\partial c_{t-1,1}}{\partial w_{ij}^{(\tilde{c})}} \odot f_{t,1} \\ &+ \text{sign}(f_{t,1}) \odot \frac{\partial c_{t-1,1}}{\partial w_{ij}^{(\tilde{c})}} + 2\delta(f_{t,1}) \odot \frac{\partial f_{t,1}}{\partial w_{ij}^{(\tilde{c})}} \odot c_{t-1,1}. \end{aligned} \quad (25)$$

Denklem (25) için, (8), (9) ve (10)'un $w_{ij}^{(\tilde{c})}$ 'e göre türevi şu şekilde hesaplanmaktadır:

$$\frac{\partial i_{t,1}}{\partial w_{ij}^{(\tilde{c})}} = D_{t,1}^{(\sigma'(\zeta^{(i)}))} \left(b_i \odot \lambda_{t-1}^{(R^{(i)}h)} \right) \quad (26)$$

$$\frac{\partial f_{t,1}}{\partial w_{ij}^{(\tilde{c})}} = D_{t,1}^{(\sigma'(\zeta^{(f)}))} \left(b_f \odot \lambda_{t-1}^{(R^{(f)}h)} \right) \quad (27)$$

$$\begin{aligned} \frac{\partial \tilde{c}_{t,1}}{\partial w_{ij}^{(\tilde{c})}} &= D_{t,1}^{(g'(\zeta^{(\tilde{c})}))} \left((\text{sign}(x_{t,1,j}) + 2\delta(w_{ij}^{(\tilde{c})})x_{t,1,j})e_i \right. \\ &\quad \left. + b_{\tilde{c}} \odot \lambda_{t-1}^{(R^{(\tilde{c})}h)} \right). \end{aligned} \quad (28)$$

Denklem (26)-(28) kullanılarak (25) hesaplanmaktadır. Sonra, (21)'i hesaplamak için (25) ve (23) kullanılır. (21) elde edildikten sonra, parametreler OBİ ve ÜG merkezli algoritmalarla güncellenir.

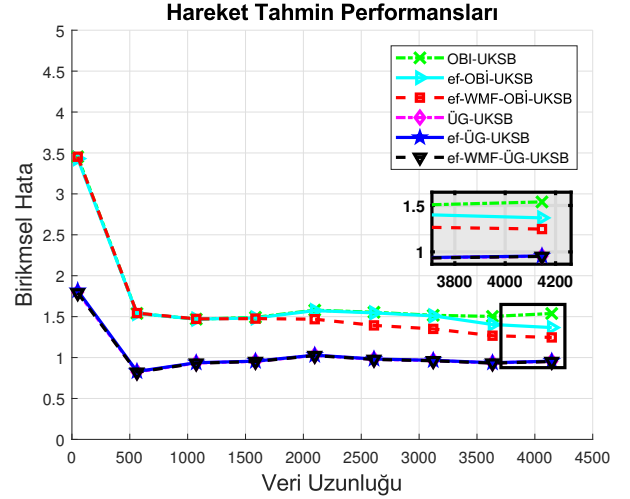
Denklem (25)'deki özyineleme, $n_t \neq 1$ durumunda Şekil 1'de görüldüğü gibi çeşitli UKSB bloklarının çıktılarından hesaplanmaktadır. Ayrıştırılmış UKSB ağı için ise hata fonksiyonun tüm ayrıştırılmış matrislere göre türevlerinin hesaplanması gerekmektedir.

IV. SAYISAL ÖRNEKLER

Bu bölümde, önerilen algoritmaların performansı kıyaslanmaktadır. Bu bölüm boyunca, "Yapı 1" geleneksel UKSB ağını temsil etmektedir. Benzer bir şekilde, "Yapı 2" ef işlemi ile tanımlanmış UKSB ağını (ef-UKSB) ve "Yapı 3" ise "Yapı 2" ile tanımlanmış ağı ağırlık matrislerinin ayrıştırılmasını (ef-WMF-UKSB) temsil etmektedir. Önerilen algoritmaların karşılaştırılmasında F16 savaş uçaklarının hareket anlarına ait bir veri kümesi kullanılmaktadır [10]. Buradaki amaç, veri kümesindeki $X_t \in \mathbb{R}^{18}$ bağlanım vektörü kullanılarak F16 uçaklarının konumunu tahmin etmektir. Deneylerde öğrenme hızı $\eta = 0.1$ olarak kullanılmıştır. Yapı 3 için matris derecesi 2 seçilmiştir. Şekil 2'de görüldüğü üzere, ÜG merkezli algoritmalar OBİ merkezli algoritmalarından daha üstün bir performans göstermektedir. OBİ merkezli algoritmalar Yapı 3 en az hataya sahip iken ÜG merkezli algoritmalar yakın performans göstermişlerdir. Önerilen algoritmalar arasında az parametreye sahip olması sebebiyle Yapı 3 en uygun olanı olarak seçilmiştir.

V. SONUÇLAR

Bu bildiride, çevrimiçi doğrusal olmayan veri bağlanım problemi incelenmiş ve bu problem için UKSB merkezli yüksek verimli bir yapı sunulmaktadır. Bunun için öncelikle matris



Şekil 2: Sunulan algoritmaların hareket tahmini performansları.

çarpımı işlemi yerine daha verimli bir işlem olan ef işlemi kullanılmış daha sonra UKSB ağındaki tüm matrisler, geleneksel UKSB ağıının parametre sayısının düşürülmesi için ayrıştırılmıştır. Bu model için OBİ [2] ve ÜG [6] merkezli çevrimiçi öğrenme algoritmaları sunulmaktadır. Böylece, UKSB merkezli yüksek verimli yapılar ve etkili çevrimiçi öğrenme algoritmaları elde edilmiştir. Sayısal örneklerde, önerilen yapıların geleneksel UKSB yapısına göre üstün bir performans ortaya koyduğu gözlemlenmiştir.

KAYNAKLAR

- [1] K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink, and J. Schmidhuber, "LSTM: A search space odyssey," *IEEE Transactions on Neural Networks and Learning Systems*, vol. PP, no. 99, pp. 1–11, 2016.
- [2] A. C. Tsoi, "Gradient based learning methods," in *Adaptive processing of sequences and data structures*. Springer, 1998, pp. 27–62.
- [3] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE Transactions on Neural Networks*, vol. 5, no. 2, pp. 157–166, Mar 1994.
- [4] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.
- [5] T. Ergen and S. S. Kozat, "Efficient online learning algorithms based on lstm neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. PP, no. 99, pp. 1–12, 2018.
- [6] J. Kivinen and M. K. Warmuth, "Exponentiated gradient versus gradient descent for linear predictors," *Information and Computation*, vol. 132, no. 1, pp. 1–63, 1997.
- [7] H. Tuna, I. Onaran, and A. E. Cetin, "Image description using a multiplier-less operator," *IEEE Signal Processing Letters*, vol. 16, no. 9, pp. 751–753, Sept 2009.
- [8] O. Kuchaiev and B. Ginsburg, "Factorization tricks for LSTM networks," *CoRR*, vol. abs/1703.10722, 2017. [Online]. Available: <http://arxiv.org/abs/1703.10722>
- [9] T. Ergen and S. S. Kozat, "Online training of lstm networks in distributed systems for variable length data sequences," *IEEE Transactions on Neural Networks and Learning Systems*, vol. PP, no. 99, pp. 1–7, 2017.
- [10] J. Alcalá-Fdez, A. Fernández, J. Luengo, J. Derrac, S. García, L. Sánchez, and F. Herrera, "KEEL data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework," *Journal of Multiple-Valued Logic and Soft Computing*, vol. 17, no. 2-3, pp. 255–287, 2011.