

Sinir Ağları ile Çevrimiçi Dağıtılmış Doğrusal Olmayan Bağlanım

Online Distributed Nonlinear Regression via Neural Networks

Tolga Ergen ve Süleyman S. Kozat
Elektrik ve Elektronik Mühendisliği Bölümü
İhsan Doğramacı Bilkent Üniversitesi
Ankara, Türkiye
{ergen,kozat}@ee.bilkent.edu.tr

Özetçe —Bu bildiri, içinde düğümler bulunan bir ağda doğrusal olmayan bağlanım problemi çalışılmıştır ve uzun-kısa soluklu bellek (UKSB) ağları merkezli bir algoritma sunulmuştur. UKSB yapısının parametrelerini çevrimiçi olarak öğrenebilmek için, UKSB yapısının denklemleri doğrusal olmayan durum uzay formuna koyulmuştur ve sonra dağıtılmış parçacık süzme (DPS) merkezli parametre öğrenme algoritması sunulmuştur. Simülasyonlarda, sunulan algoritma tarafından klasik metotlara kıyasla erişilen performans iyileştirmesi gösterilmiştir.

Anahtar Kelimeler—doğrusal olmayan bağlanım, çevrimiçi öğrenim, dağıtılmış parçacık süzme, uzun kısa soluklu bellek ağı.

Abstract—In this paper, we study the nonlinear regression problem in a network of nodes and introduce long short term memory (LSTM) based algorithms. In order to learn the parameters of the LSTM architecture in an online manner, we put the LSTM equations into a nonlinear state space form and then introduce our distributed particle filtering (DPF) based training algorithm. Our training algorithm asymptotically achieves the optimal training performance. In our simulations, we illustrate the performance improvement achieved by the introduced algorithm with respect to the conventional methods.

Keywords—nonlinear regression, online learning, distributed particle filtering, long short term memory network.

I. GİRİŞ

Dağıtılmış öğrenim algoritmaları birçok gerçek hayat uygulamalarını başarılı bir şekilde modelleyebildikleri için sinyal işleme literatüründe kapsamlı olarak çalışılmıştır [1]. Bu algoritmalar genelde içinde düğümler bulunduran ağlar hakkında bilgi edinmek için kullanılmıştır [1]. Bu bildiri, özellikle içinde düğümler bulunduran ağlar için doğrusal olmayan bağlanım problemine odaklanılmıştır. Bu yapıda, her bir düğüm ardışık olarak değişken uzunluklu bağlanım vektörü ve bu vektörün etiketini almaktadır. Daha sonra, vektör ve etiketi arasındaki ilişkiye bakarak, bir sonraki bağlanım vektörünün etiketi tahmin edilmeye çalışılmaktadır [2].

Sinyal işleme literatüründe çeşitli doğrusal olmayan bağlanım metotları bulunmaktadır [2]. Ancak, bu metotlar yeterli

performans gösterememekten ve yüksek hesaplama karmaşıklığından mağdur olmaktadır. Bu sorunları çözmek için sinir ağları merkezli bağlanım algoritmaları sunulmuştur [3]. Bu amaç için yüksek derece doğrusal olmayan yapıları modelleme ve veri içindeki hem uzun hem kısa dönemli bağımlılıkları yakalama yeteneklerinden dolayı, genellikle uzun kısa soluklu bellek (UKSB) sinir ağları kullanılmaktadır [3].

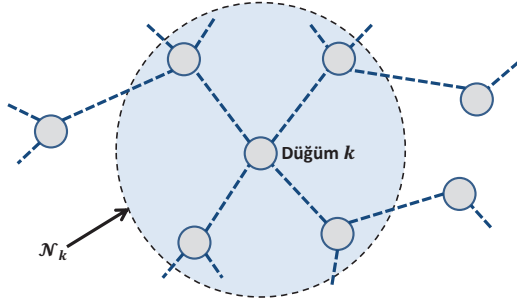
UKSB yapısını kullanmanın esas sorunu bu yapının parametrelerini öğrenmektir. UKSB yüksek derecede karmaşık bir yapıya sahip olduğundan dolayı, parametreleri öğrenmek amacıyla genellikle birinci dereceden bayır merkezli algoritmalar kullanılmaktadır [4]. Ancak, birinci dereceden bayır merkezli algoritmalar, daha karmaşık algoritmalara göre yavaş yaklaşım göstermekten ve kötü performanstan mağdur olmaktadır [2].

Bu bildiri, literatürdeki problemler UKSB merkezli bir öğrenim algoritması sunularak ve UKSB yapısının parametreleri önerilen çevrimçi dağıtılmış parçacık süzme (DPS) [5] algoritması ile öğrenilerek çözülmektedir. Bu bildiri işlenen ağ yapısında, her bir düğüm bir sonraki bağlanım vektörü etiketini tahmin edebilmek için UKSB yapısını kullanmaktadır. Buna ek olarak, her bir düğüm UKSB yapısının parametrelerini önerilen DPS algoritması ile öğrenmektedir. Bu ağda, UKSB yapısı kullanıldığından dolayı veri içindeki hem uzun hem kısa dönemli bağımlılıklar yakalanabilmektedir. Ayrıca, DPS algoritması ile UKSB yapısının parametreleri etkili ve çevrimiçi bir şekilde öğrenilmektedir.

II. PROBLEM TANIMI

Bu bildiri, Şekil 1'de gösterildiği gibi bir ağ yapısı incelenmektedir. Bu ağ yapısında K tane düğüm bulunmaktadır. Her bir düğüm k 'nin komşular kümesi \mathcal{N}_k ile gösterilmektedir ve düğüm k komşuları ile veri alışverişinde bulunabilmektedir. Buna ek olarak her bir düğüm k 'nin komşu sayısı η_k ile gösterilmektedir. Bu düğümlerden her biri ardışık olarak $\{d_{k,t}\}_{t \geq 1}$, $d_{k,t} \in \mathbf{R}$ istenen sinyalini ve $\{\mathbf{X}_t\}_{t \geq 1}$ bağlanım matrisini almaktadır. Bağlanım matrisi \mathbf{X}_t aşağıdaki şekilde tanımlanmaktadır:

$$\mathbf{X}_t = [\mathbf{x}_{t,1} \ \mathbf{x}_{t,2} \ \dots \ \mathbf{x}_{t,m_t}].$$



Şekil 1: Çift taraflı bağlantıları olan dağıtılmış bir ağ örneği.

Burada, $\mathbf{x}_{t,l} \in \mathbf{R}^p$, $\forall l \in \{1, 2, \dots, m_t\}$ bağlantım matrisinin sütunlarını ve $m_t \in \mathbf{Z}^+$ bağlantım matrisinin zaman indeksi t 'ye göre değişebilen sütun sayısını temsil etmektedir. Buradaki amaç şimdiki ve geçmiş gözlemlerimize $\{\dots, \mathbf{X}_{t-1}, \mathbf{X}_t\}$ dayanarak $d_{k,t}$ 'yi tahmin etmektir. Her bir düğüm k 'deki tahmin $\hat{d}_{k,t}$ şeklinde gösterilmektedir. Bu bağlamda, $\hat{d}_{k,t}$, $\{\dots, \mathbf{X}_{t-1}, \mathbf{X}_t\}$ ve $\{\dots, d_{t-2,k}, d_{t-1,k}\}$ 'in bir fonksiyonu olarak görülebilir. Buna ek olarak, her bir zaman indeksi t 'de, düğüm k 'nin hata fonksiyonu $l(d_{k,t}, \hat{d}_{k,t})$ şeklinde gösterilmektedir.

Bu bildiriye, $\hat{d}_{k,t}$ 'yi tahmin edebilmek için UKSB sınır ağı kullanılmaktadır. Bu ağın yapısı aşağıdaki denklemler ile açıklanmaktadır [3]:

$$\mathbf{z}_t = g(\mathbf{W}^{(z)}\mathbf{x}_t + \mathbf{R}^{(z)}\mathbf{y}_{t-1} + \mathbf{b}^{(z)}) \quad (1)$$

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

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

$$\mathbf{c}_t = \mathbf{i}_t \odot \mathbf{z}_t + \mathbf{f}_t \odot \mathbf{c}_{t-1} \quad (4)$$

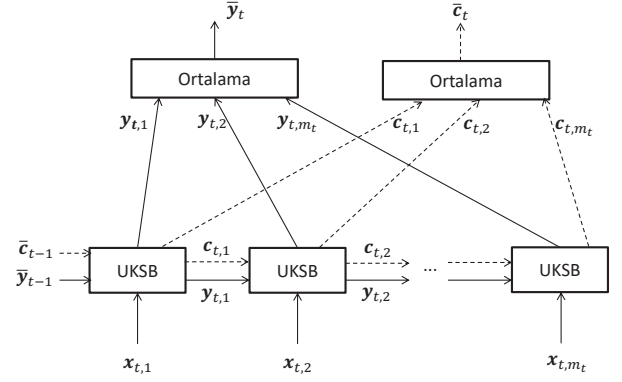
$$\mathbf{o}_t = \sigma(\mathbf{W}^{(o)}\mathbf{x}_t + \mathbf{R}^{(o)}\mathbf{y}_{t-1} + \mathbf{b}^{(o)}) \quad (5)$$

$$\mathbf{y}_t = \mathbf{o}_t \odot h(\mathbf{c}_t). \quad (6)$$

Yukarıdaki denklemlerde, giriş vektörü $\mathbf{x}_t \in \mathbf{R}^p$, çıkış vektörü $\mathbf{y}_t \in \mathbf{R}^n$ ve durum vektörü $\mathbf{c}_t \in \mathbf{R}^n$ şeklinde gösterilmektedir. Buna ek olarak, \mathbf{o}_t , \mathbf{f}_t ve \mathbf{i}_t sırasıyla çıkış, unutma ve giriş kapıları temsil etmektedir. $g(\cdot)$ ve $h(\cdot)$ fonksiyonları için hiperbolik tanjant fonksiyonu $\tanh(\cdot)$ kullanılmaktadır ve bu fonksiyonlar her bir vektöre noktabanlı olarak uygulanmaktadır. Benzer bir şekilde, $\sigma(\cdot)$ fonksiyonu için ise sigmoit fonksiyonu kullanılmaktadır. Ayrıca, yukarıdaki denklemlerdeki \odot işlemi aynı boyutta iki vektörün her bir elementini ayrı ayrı çarparak yeni bir çarpım vektörü oluşturmaktadır. Bu denklemlerdeki diğer matris ve vektörler için tanımlamalar şu şekildedir: $\mathbf{W}^{(z)} \in \mathbf{R}^{n \times p}$, $\mathbf{R}^{(z)} \in \mathbf{R}^{n \times n}$, $\mathbf{b}^{(z)} \in \mathbf{R}^n$, $\mathbf{W}^{(i)} \in \mathbf{R}^{n \times p}$, $\mathbf{R}^{(i)} \in \mathbf{R}^{n \times n}$, $\mathbf{b}^{(i)} \in \mathbf{R}^n$, $\mathbf{W}^{(f)} \in \mathbf{R}^{n \times p}$, $\mathbf{R}^{(f)} \in \mathbf{R}^{n \times n}$, $\mathbf{b}^{(f)} \in \mathbf{R}^n$, $\mathbf{W}^{(o)} \in \mathbf{R}^{n \times p}$, $\mathbf{R}^{(o)} \in \mathbf{R}^{n \times n}$ ve $\mathbf{b}^{(o)} \in \mathbf{R}^n$. UKSB'nin çıkışını kullanarak her bir düğüm k için tahmin aşağıdaki şekilde hesaplanmaktadır:

$$\hat{d}_{k,t} = \mathbf{w}_t^T \bar{\mathbf{y}}_t. \quad (7)$$

Burada, $\mathbf{w}_t \in \mathbf{R}^n$ bağlantım katsayıları vektörünü ve $\bar{\mathbf{y}}_t \in \mathbf{R}^n$ ortalama metodu [3] Şekil 2'deki gibi hesaplanan çıkış vektörünü temsil etmektedir. Bu ağ için yapılmak istenen UKSB'nin parametrelerini toplam hata fonksiyonu $\sum_{i=1}^t l(d_{i,k}, \hat{d}_{i,k})$ 'nu minimuma indirecek şekilde öğrenmektir.



Şekil 2: Ağdaki her bir düğümün detaylı bir şematiği. Burada, şekildedeki her bir UKSB yapısının parametreleri aynıdır, ancak sunum kolaylığı açısından ayrı yapılar olarak gösterilmektedir.

III. DAĞITILMIŞ ÇEVİRİMİÇİ PARAMETRE ÖĞRENİM ALGORİTMASI

Bu bölümde, öncelikle UKSB yapısının denklemleri durum uzay formunda tanımlanmaktadır. Daha sonra bu form kullanılarak, PS ve DPS merkezli çevrimiçi parametre öğrenim algoritmaları sunulmaktadır.

Şekil 2'deki yapı ve UKSB denklemleri (1), (2), (3), (4), (5) ve (6) beraber değerlendirildiğinde, düğüm k için aşağıdaki gibi bir durum uzay formuna ulaşılabilir:

$$\bar{\mathbf{c}}_t = \Omega(\bar{\mathbf{c}}_{t-1}, \mathbf{X}_t, \bar{\mathbf{y}}_{t-1}) \quad (8)$$

$$\bar{\mathbf{y}}_t = \Theta(\bar{\mathbf{c}}_t, \mathbf{X}_t, \bar{\mathbf{y}}_{t-1}) \quad (9)$$

$$\boldsymbol{\theta}_t = \boldsymbol{\theta}_{t-1} \quad (10)$$

$$d_{k,t} = \mathbf{w}_t^T \bar{\mathbf{y}}_t + \varepsilon_{k,t}. \quad (11)$$

Yukarıdaki denklemlerde, $\Omega(\cdot)$ ve $\Theta(\cdot)$ fonksiyonları Şekil 2'de gösterilen ardışık UKSB ve ortalama işlemleri tarafından yapılan doğrusal olmayan işlemleri temsil etmektedir. Buna ek olarak, $\boldsymbol{\theta}_t \in \mathbf{R}^{n_\theta}$ vektörü $\{\mathbf{w}, \mathbf{W}^{(z)}, \mathbf{R}^{(z)}, \mathbf{b}^{(z)}, \mathbf{W}^{(i)}, \mathbf{R}^{(i)}, \mathbf{b}^{(i)}, \mathbf{W}^{(f)}, \mathbf{R}^{(f)}, \mathbf{b}^{(f)}, \mathbf{W}^{(o)}, \mathbf{R}^{(o)}, \mathbf{b}^{(o)}\}$ parametrelerinden oluşmuş bir vektördür ve boyutu $n_\theta = 4n(n+p) + 5n$ şeklinde tanımlanmaktadır. Burada amaç sistem parametrelerini öğrenmek olduğu için denklem (10) bir durum vektörü olarak sisteme dahil edilmiştir. Bundan başka, $\varepsilon_{k,t}$ rastlantısal bir değişkendir ve $R_{k,t}$ onun varyansını temsil etmektedir.

A. Genel Parçacık Süzme Algoritması Merkezli Parametre Öğrenimi

Bu bölümde, genel PS algoritması merkezli parametre öğrenim algoritması sadece bir düğüm olan $K = 1$ bir ağ için sunulmaktadır. PS algoritması sadece rastlantısal değişkenlerin bağımsız olmasına ihtiyaç duymaktadır. Bundan dolayı, (8), (9), (10) ve (11)'deki sistem, aşağıdaki gibi değiştirilerek daha sade bir formda yazılabilmektedir:

$$\mathbf{a}_t = \varphi(\mathbf{a}_{t-1}) + \boldsymbol{\epsilon}_t \quad (12)$$

$$d_t = \mathbf{w}_t^T \bar{\mathbf{y}}_t + \xi_t. \quad (13)$$

Yukarıdaki denklemlerde, $\boldsymbol{\epsilon}_t$ ve ξ_t sırasıyla durum ve ölçüm gürültü örneklerini temsil etmektedir. Ayrıca, $\varphi(\cdot)$ fonksiyonu (8), (9) ve (10) denklemlerindeki doğrusal olmayan işlemi

temsil etmektedir. Denklem (12)'ye sonradan eklenen gürültü değişkeni, parametrelerin rastgele yürüme şeklinde güncellenmelerini sağlamaktadır. Buna ek olarak, denklem (12)'deki durum vektörü \mathbf{a}_t aşağıdaki gibi tanımlanmaktadır:

$$\mathbf{a}_t = \begin{bmatrix} \bar{\mathbf{c}}_t \\ \bar{\mathbf{y}}_t \\ \bar{\boldsymbol{\theta}}_t \end{bmatrix}. \quad (14)$$

Denklem (12) ve (13)'teki sistem için amaç ortalama karekök hatası açısından en iyi parametre tahmini olan $\mathbf{E}[\mathbf{a}_t|d_{1:t}]$ 'ni elde etmektir. Bunu gerçekleştirebilmek için öncelikle sonsal yoğunluk fonksiyonu olan $p(\mathbf{a}_t|d_{1:t})$ bulunmaktadır. Bu sonsal yoğunluk fonksiyonu bulunduktan sonra PS algoritması [6] aşağıdaki gibi uygulanmaktadır.

Bu algoritmada, sonsal yoğunluk fonksiyonunun örnekleri ve bu örneklerin ağırlıkları mevcut bulunmaktadır ve bunlar $\{\mathbf{a}_t^i, \omega_t^i\}_{i=1}^N$ şeklinde gösterilmektedir. Bu örnekler kullanılarak, sonsal yoğunluk fonksiyonu aşağıdaki gibi hesaplanabilir:

$$p(\mathbf{a}_t|d_{1:t}) \approx \sum_{i=1}^N \omega_t^i \delta(\mathbf{a}_t - \mathbf{a}_t^i). \quad (15)$$

İstenilen sonsal yoğunluk fonksiyonundan örnekleme genellikle zordur, bu yüzden, örnekleme $q(\mathbf{a}_t|d_{1:t})$ fonksiyonunu kullanarak gerçekleştirilmektedir [7]. Bu fonksiyon literatürde önem fonksiyonu olarak da bilinmektedir. Denklem (15)'teki ağırlıkları hesaplamak için aşağıdaki formül kullanılmaktadır:

$$w_t^i \propto \frac{p(\mathbf{a}_t^i|d_{1:t})}{q(\mathbf{a}_t^i|d_{1:t})} \text{ ve } \sum_{i=1}^N \omega_t^i = 1. \quad (16)$$

Denklem (16) çarpanlarına ayrılarak aşağıdaki tekrarlanan formül elde edilmektedir [7]:

$$\omega_t^i \propto \frac{p(d_t|\mathbf{a}_t^i)p(\mathbf{a}_t^i|\mathbf{a}_{t-1}^i)}{q(\mathbf{a}_t^i|\mathbf{a}_{t-1}^i, d_t)} \omega_{t-1}^i. \quad (17)$$

Denklem (16)'daki önem fonksiyonu öyle bir şekilde seçilmelidir ki ağırlıkların varyansı minimum değerine ulaşsın. Böylece, bütün parçacıklar büyük ölçüde denklem (15)'e katkıda bulunacaktır [7]. Bu açıdan, önem fonksiyonu $p(\mathbf{a}_t^i|\mathbf{a}_{t-1}^i)$ olarak seçilmektedir çünkü bu fonksiyon kolayca hesaplanabilir ve ağırlıklar için küçük bir varyans değeri oluşturmaktadır [6]. Bu değişikliklerle birlikte denklem (17) aşağıdaki gibi değişmektedir:

$$\omega_t^i \propto p(d_t|\mathbf{a}_t^i)\omega_{t-1}^i. \quad (18)$$

Denklem (15) ve (18) kullanılarak, durum vektörü için aşağıdaki tahmin elde edilmektedir:

$$\begin{aligned} \mathbf{E}[\mathbf{a}_t|d_{1:t}] &= \int \mathbf{a}_t p(\mathbf{a}_t|d_{1:t}) d\mathbf{a}_t \\ &\approx \int \mathbf{a}_t \sum_{i=1}^N \omega_t^i \delta(\mathbf{a}_t - \mathbf{a}_t^i) d\mathbf{a}_t = \sum_{i=1}^N \omega_t^i \mathbf{a}_t^i. \end{aligned} \quad (19)$$

Burada, önem fonksiyonunu ağırlıkların varyansını küçük bir değer haline getirecek şekilde seçilmesine rağmen zaman geçtikçe ağırlıkların varyansı artmaktadır [6], [7]. Bunu önlemek için [7]'de önerilen yeniden örnekleme algoritması kullanılmaktadır.

B. Dağıtılmış Parçacık Süzme Algoritması Merkezli Parametre Öğrenimi

Bu bölümde, DPS algoritması merkezli çevrimiçi parametre öğrenim algoritması sunulmaktadır. Bir önceki bölümün aksine, bu bölümde düğüm sayısı birden fazla olabilir. Bundan dolayı, her bir düğüm (12) ve (13)'teki sisteme sahiptir. Bu ağırlık parametrelerini öğrenmek için Markov zinciri dağıtılmış parçacık süzme (MZDPS) [5] algoritması kullanılmaktadır. Bu algoritmada, parçacıklar ağ içinde rastgele olarak ağırlık topolojisine göre değişik düğümlere hareket etmektedir. Her bir adımda, parçacıklar o anda bulunduğu düğüm komşularından birine rastgele hareket etmektedirler. Bu hareket esnasında, her bir tanecığın ağırlığı düğüm k 'de iken $p(d_{k,t}|\mathbf{a}_t)$ kullanılarak güncellenmektedir ve böylece bu parçacıklar düğüm k 'deki gözleme dayalı bilgiyi elde etmektedir. Bu ağ yapısı köşeler V ile gösterilen ve kenarları E ile gösterilen bir grafik $G = (V, E)$ olarak düşünülebilir. Buna ek olarak, parçacık i 'nin düğüm k 'yi n adımda ziyaret etme sayısı $M^i(k, n)$ ile gösterilmektedir. Bu ağ yapısında, parçacıkların hareketleri komşuluk matrisi \mathcal{A} ile belirlenmektedir. Bu bağlamda, eğer ağdaki toplam kenar sayısı $|E(G)|$ ile gösterilirse, toplamda n adım atıldığında her bir parçacık her düğüm k 'yi ziyaret ettiğinde kendisinin ağırlığını $p(d_{k,t}|\mathbf{a}_t)^{\frac{2|E(G)|}{n\eta_k}}$ ile çarpmaktadır [5]. Denklem (18) kullanılarak, toplamda n adım atıldığında parçacık i için düğüm k 'de aşağıdaki güncelleme yapılmaktadır:

$$w_{k,t}^i \propto w_{k,t-1}^i \prod_{j=1}^K p(d_{j,t}|\mathbf{a}_t^i)^{\frac{2|E(G)|}{n\eta_j}} M^i(j, n). \quad (20)$$

Yukarıdaki denklem kullanılarak, düğüm k 'deki sonsal yoğunluk fonksiyonu aşağıdaki gibi hesaplanmaktadır:

$$p(\mathbf{a}_t|d_{k,1:t}) \approx \sum_{i=1}^N w_{k,t}^i \delta(\mathbf{a}_t - \mathbf{a}_t^i). \quad (21)$$

Sonsal yoğunluk fonksiyonu (21) ile hesaplandıktan sonra, düğüm k 'nin durum vektörü \mathbf{a}_t için yaptığı tahmin aşağıdaki formül ile hesaplanmaktadır:

$$\begin{aligned} \mathbf{E}[\mathbf{a}_t|d_{k,1:t}] &= \int \mathbf{a}_t p(\mathbf{a}_t|d_{k,1:t}) d\mathbf{a}_t \\ &\approx \int \mathbf{a}_t \sum_{i=1}^N \omega_{k,t}^i \delta(\mathbf{a}_t - \mathbf{a}_t^i) d\mathbf{a}_t = \sum_{i=1}^N \omega_{k,t}^i \mathbf{a}_t^i. \end{aligned} \quad (22)$$

Bu bölümde kullandığımız metod, en iyi parametre tahminine yakınsamayı garanti etmektedir [5]. Bu algoritma ile bütün düğümler için tahminler Algoritma 1'de gösterdiği gibi yapılabilmektedir. Algoritma 1'da $N(j)$ j düğümündeki parçacık sayısını ve $\mathcal{I}_{l \rightarrow j}$ ise l düğümünden j düğümüne hareket eden parçacıkların indekslerini temsil etmektedir.

IV. SAYISAL ÖRNEKLER

Bu bölümde, önerilen algoritmanın geleneksel algoritmalara göre performans değerlendirmesi yapılmaktadır. Burada, dört farklı algoritma karşılaştırılacaktır. Bunlardan birincisi olasılıksal bayır inişi (OBİ) merkezli algoritmasıdır ve bu algoritma UKSB yapısının parametrelerini öğrenmek için geleneksel bir algoritmadır. PS merkezli algoritma ikinci algoritmadır

Algorithm 1 Dağıtılmış Parçacık Süzme Algoritması Merkezli Parametre Öğrenim Algoritması

- 1: **Giriş:** Bütün düğümlerdeki bütün parçacıklar ve onların ağırlıkları
- 2: **for** n adım **do** % Parçacıkları n adım hareket ettirir
- 3: Parçacıkları \mathcal{A} 'ya göre hareket ettir
- 4: **for** $j = 1 : K$ **do**
- 5: $\{\mathbf{a}_{j,t}^i\}_{i=1}^{N(j)} = \bigcup_{l \in \mathcal{N}_j} \{\mathbf{a}_{l,t}^i\}_{i \in \mathcal{I}_{l \rightarrow j}}$
- 6: $\{w_{j,t}^i\}_{i=1}^{N(j)} = \bigcup_{l \in \mathcal{N}_j} \{w_{l,t}^i\}_{i \in \mathcal{I}_{l \rightarrow j}}$
- 7: $\{w_{j,t}^i\}_{i=1}^{N(j)} \leftarrow \{w_{j,t}^i\}_{i=1}^{N(j)} p(d_{j,t} | \{\mathbf{a}_{j,t}^i\}_{i=1}^{N(j)})^{\frac{2|E(G)|}{n\eta_j}}$
- 8: **end for**
- 9: **end for**
- 10: Her bir düğümdeki parçacıkları [7]'yi kullanarak yeniden örnekle
- 11: **Çıkış:** Her bir düğümdeki güncellenmiş parçacıklar ve onların ağırlıkları

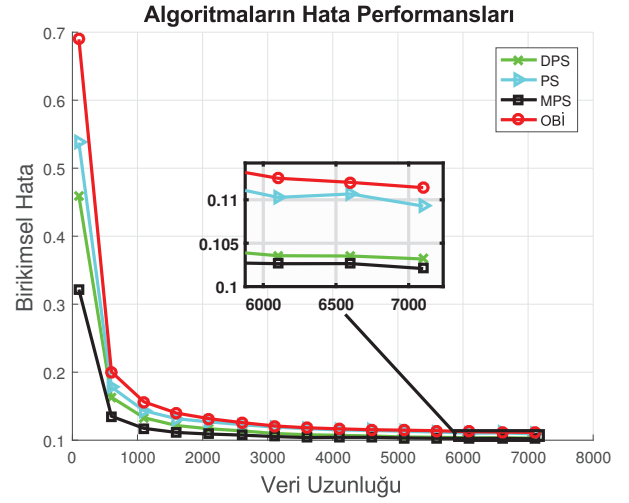
ve bu algoritmada düğümler arasında veri alışverişi yoktur. Üçüncü algoritma önerilen DPS merkezli algoritmasıdır. Son olarak, dördüncü algoritma merkezleştirilmiş PS (MPS) algoritmasıdır ve bu algoritmada her bir düğüm sistemdeki diğer bütün düğümlerin gözlemlerine ulaşabilmektedir.

Bu algoritmaları karşılaştırmak için el yazısı ile yazılmış rakamlara ait bir veri kümesi kullanılmaktadır. Bu veri kümesinde, her bir piksele ait iki boyutlu pozisyon vektörleri ve bu vektörlere ait etiketler bulunmaktadır. Bu bağlamda, $\mathbf{X}_t \in \mathbf{R}^{2 \times m_t}$ bağlantım matrisini ve m_t bağlantım matrisindeki zamana göre değişebilen sütun sayısını temsil etmektedir. Buradaki amaç daha önce alınan bağlantım matrisleri ve etiketlerini inceleyerek, gelecekteki etiketleri tahmin etmektir. Bu problem için 4 düğümü olan $K = 4$ bir ağ kullanılmıştır. Önerilen DPS algoritması için Markov adım sayısı $n = 10$ seçilmiştir ve düğümler arasındaki komşuluk ilişkileri aşağıdaki gibidir:

$$\mathbf{A} = \begin{bmatrix} 0 & \frac{1}{2} & 0 & \frac{1}{2} \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ 0 & \frac{1}{2} & 0 & \frac{1}{2} \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 \end{bmatrix}. \quad (23)$$

PS merkezli algoritmalar için parçacık sayısı $N = 50$ olarak seçilmiştir. OBI merkezli algoritma için öğrenme hızı $\mu = 0.005$ olarak seçilmiştir. Ayrıca, bütün algoritmalar için ϵ_t ve $\xi_{k,t}$ sıfır ortalamalı ve sırasıyla $\text{Cov}[\epsilon_t] = 0.0004\mathbf{I}$ ve $\text{Var}[\xi_{k,t}] = 0.01$ istatistiklerine sahip olan Gauss gürültüleri olarak ele alınmıştır. Bu değişken değerleri algoritmaların benzer bir durağan durum hatasına ulaşması için seçilmiştir. Böylece, algoritmalar adil bir şekilde karşılaştırılmıştır.

Şekil 3'te, algoritmaların birikimsel hataları karşılaştırılmaktadır. Beklenildiği üzere, OBI merkezli algoritma sadece birinci dereceden bayır bilgisi kullandığı için bu algoritmalar arasında en kötü performansı göstermektedir. PS merkezli algoritma OBI'den az oranda daha iyi olsa da, düğümler arasında iletişim eksikliğinden dolayı o da kötü bir performans göstermektedir. Önerilen DPS merkezli algoritma PS ve OBI algoritmalarına göre hayli üstün bir performans göstermektedir. Bunun yanında, önerilen DPS algoritmasında, düğümler sadece komşularıyla iletişim kurabilmesine rağmen, bu algoritma MPS algoritmasıyla neredeyse aynı performansı göstermektedir. Böylece, önerilen DPS algoritması PS ve OBI'den



Şekil 3: Algoritmaların birikimsel hata performansları.

çok daha iyi bir performans gösterirken, MPS algoritmasının performansına çok yaklaşılmaktadır.

V. SONUÇLAR

Bu bildiriye, içinde düğümler bulunduran bir ağ için doğrusal olmayan bağlantım problem incelenmiş ve değişken veri uzunluklu bağlantım için UKSB merkezli çevrimiçi bir algoritma sunulmuştur. Daha sonra, UKSB yapısının parametrelerini öğrenmek için çevrimiçi parametre öğrenim algoritmaları sunulmuştur. Bunun için öncelikle her bir düğümde UKSB yapısının denklemleri doğrusal olmayan durum uzay formuna konup daha sonra bu form üzerinde önerilen DPS merkezli parametre öğrenim algoritması çıkarılmıştır. Böylece, UKSB yapısı içeren bir ağ için en iyi parametre tahminine yakınsamayı garanti eden bir algoritma elde edilmiştir. Sayısal örneklerde, önerilen DPS algoritmasının geleneksel algoritmalar göre üstün bir performans ortaya koyduğu gözlenmiştir.

KAYNAKLAR

- [1] F. S. Cattivelli and A. H. Sayed, "Diffusion strategies for distributed kalman filtering and smoothing," *IEEE Transactions on Automatic Control*, vol. 55, no. 9, pp. 2069–2084, Sept 2010.
- [2] N. Cesa-Bianchi and G. Lugosi, *Prediction, learning, and games*. Cambridge university press, 2006.
- [3] 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.
- [4] J. Mazumdar and R. G. Harley, "Recurrent neural networks trained with backpropagation through time algorithm to estimate nonlinear load harmonic currents," *IEEE Transactions on Industrial Electronics*, vol. 55, no. 9, pp. 3484–3491, 2008.
- [5] S. H. Lee and M. West, "Markov chain distributed particle filters (mcdpf)," in *Proceedings of the 48th IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*, Dec 2009, pp. 5496–5501.
- [6] P. M. Djuric, J. H. Kotecha, J. Zhang, Y. Huang, T. Ghirmai, M. F. Bugallo, and J. Miguez, "Particle filtering," *IEEE signal processing magazine*, vol. 20, no. 5, pp. 19–38, 2003.
- [7] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking," *IEEE Transactions on signal processing*, vol. 50, no. 2, pp. 174–188, 2002.