UKSB Sinir Ağları ile Eksik Veri Kümesi Üzerinde Sıralı Bağlanım

Sequential Regression with Missing Data Using LSTM Networks

Safa Onur Şahin

Elektrik Elektronik Mühendisliği Bölümü, İhsan Doğramacı Bilkent Üniversitesi, Ankara, Türkiye
ASELSAN Yapay Zeka ve Bilişim Tekn. Prg. Mdl., Ankara, Türkiye
ssahin@ee.bilkent.edu.tr

Özetçe —Bu bildiride, içerisinde eksik bilgi bulunan veri kümesinin Uzun Kısa-Soluklu Bellek (UKSB) sinir ağları ile sıralı bağlanımı çalışılmıştır. UKSB sinir ağını kullanan sıralı bağlanım uygulamalarında veri kümesi genellikle eksiksiz olarak kabul edilir. Ancak, eksik veri problemi sıralı veri içeren gerçek hayat uygulamalarında sıklıkla karşılaşılan bir sorundur. Bu probleme çözüm amacıyla sunulan yöntemlerde eksik veri, sıralı verideki örüntüyü yakalayacak derecede modellenememekte ve bu yüzden yüksek performans artışları görünmemektedir. Bu bildiride, eksik veri, bağlanımı gerçekleştiren UKSB ağı tabanlı sinir ağının kendisi tarafından modellenmekte ve bağlanım sırasında üretilen bu veri kullanılmaktadır. Gerçek hayat uygulamalarından elde edilmiş sıralı veri kümeleriyle yapılan deneylerde, önerilen algoritmanın geleneksel metotlar karşısında üstün performans artışına sahip olduğu gözlemlenmiştir.

Anahtar Kelimeler—Uzun Kısa-Soluklu Bellek Sinir Ağları, Eksik Veri, Sıralı Bağlanım, Tekrarlayan Sinir Ağları

Abstract—We study regression for variable length sequential data suffering from missing samples and introduce a long short-term memory (LSTM) based sequential regression algorithm. In most sequential regression studies, one considers data sequence is complete, i.e., does not contain any missing data. However, the missing data problem appears in a large number of areas such as finance and medical imaging. The remedies to resolve this problem depends on certain statistical assumptions and imputation techniques. However, the statistical assumptions does not hold in real life and the imputation of artificially generated inputs results in sub-optimal solutions. In our experiments, we achieve significant performance gains with respect to the classical algorithms.

Keywords—Long Short-Term Memory neural networks, Missing Data, Sequential Regression, Recurrent Neural Networks

I. GİRİŞ

Bu bildiride, değişken uzunluktaki sıralı veri üzerinden bağlanım problemi çalışılmıştır. Bu problemde, eksik bilgi içeren veri sıralı olarak elde edilmekte ve bu veri ile ilgili bilinmeyen bir sinyalin tahmini yapılmaktadır. Sıralı veri içeren bağlanım uygulamalarında genellikle veri kümesinin eksiksiz olduğu varsayımı yapılır [1]. Ancak, birçok gerçek hayat uygulamasında çeşitli aksaklıklar veya maliyet sebebiyle

eksik veri ile karşılaşılmaktadır [2]. Ek olarak, çeşitli finans [3] ve medikal görüntüleme [4] uygulamalarında düzgün olmayan örnekleme kullanılmaktadır ve tür veriler eksik bilgi olarak modellenebilmektedir [2].

Yapay sinir ağları, üstün modelleme kapasitelerinden dolayı gerçek hayat uygulamalarında yüksek performans göstermektedir [5]. Bir çok farklı sinir ağı mimarisi arasında geçmişten gelen bilgiyi depolayan bellek yapısı ile tekrarlayan sinir ağları (TSA), sıralı veri kümelerinin işlenmesinde sıklıkla kullanılmaktadır [5]. Basit TSA yapısı bir çok uygulamada zaman-sal örüntüyü yakalayabildiği halde, uzun zamanlı örüntülerde oluşan çok sayıda tekrarlanan çarpma işlemlerinden dolayı gradyanların aşırı hızda büyümesi veya küçülmesi problemiyle karşılaşılır [6]. Bu sorunlara çözüm olarak, her biri özelleşmiş bir göreve sahip düzenleyici yapılarla desteklenen bir tekrarlayan sinir ağları olan uzun kısa-soluklu bellek (UKSB) ağları ortaya çıkmıştır [6].

UKSB sinir ağları sıralı veri içeren uygulamaların genelinde üstün performans göstermesine rağmen eksik veri bulunduran durumlarda önemli performans düşüşleri yaşayabilmektedir [2], [7]. Literatürde, bu sorunların çözümü için eksik olan verinin yerini bir önceki girdi vektörü [7] veya sıfır vektörüyle [8] ile doldurma gibi çeşitli yerine koyma teknikleri kullanılarak UKSB sinir ağı çalıştırılmıştır. Ancak bu yaklaşımlar, yerine koyma vektörlerinin sıralı verideki örüntüden bağımsız olarak veya tam olarak modellemeyecek şekilde seçilmesinden dolayı sınırlı bir performans artışı gösterir. Bu problemlerin çözümü için, bu bildiride, doğrusal olmayan bağlanım problemi üzerinden UKSB sinir ağlarının eksik veri içeren sıralı veri kümesi ile öğrenimi eksik veri yerine sinir ağı tarafından üretilmiş tahmin koyularak çalışılmıştır.

II. PROBLEM VE MODEL TANIMI

Bu bildiride, tüm vektörler sütun vektörleri olarak kullanılmıştır. Vektörler küçük kalın harflerle gösterilmekte iken matrisler büyük kalın harflerle temsil edilmektedirler.

Burada, $\mathbf{X} = [\mathbf{x}_{t_1}, \dots, \mathbf{x}_{t_n}] \in \mathcal{X}$ sıralı olarak elde edilmektedir. $\mathbf{x}_{t_k} \in \mathbb{R}^m$ girdi vektörünü, n ise $\mathbf{X}$ sıralı verisinin uzunluğunu temsil etmektedir. $\mathbf{X}$ sıralı verisi belirli bir örnekleme frekansıyla elde edilmiştir ancak içerisinde eksik örnekler barındırmaktadır. Belirli $\mathbf{x}_{t_k}$ 'lar sıralı veri içerisinde

bulunmamaktadır. Burada amaç, her t_k anında, bu t_k anına kadar elde edilen girdi vektörleri t_k anındaki hedef sinyali olan y_{t_k} arasındaki ilişkiyi öğrenerek y_{t_k} için tahminde bulunmaktır. Burada, t_k anındaki tahmini belirten $\hat{y}_{t_k}$, t_k anına kadar olan geçmiş ve şimdiki gözlemlerin ve hedef sinyallerin bir fonksiyonu olarak ifade edilebilir:

$$\hat{y}_{t_k} = f_t(x_{t_k}, \dots, x_{t_1}, y_{t_{k-1}}, \dots, y_{t_1}). \quad (1)$$

$f_{t_k}(\cdot)$, t_k anındaki zamanla değişen ve uyarlanabilir tahmin fonksiyonunu temsil etmektedir. Burada t_k anında gelen girdi vektörü x_{t_k} için üretilen tahmin olan $\hat{y}_{t_k}$ için zarar $l(y_{t_k}, \hat{y}_{t_k})$ olarak gösterilmektedir. Toplam zarar ise tekil zararların tüm veri kümesi üzerinde ortalaması olarak

$$L = \frac{1}{n} \sum_{k=1}^n l(y_{t_k}, \hat{y}_{t_k}) \quad (2)$$

şeklinde tanımlanmaktadır.

Örneğin, kamera kullanılarak yapılan bir cisim takibi ve pozisyon kestirimi uygulamasında, x_{t_k} lar sıralı olarak gelmekte ve takip edilen cismin üç boyutlu koordinatlarını barındırmaktadır. Burada, hedef cismin bir sonraki andaki pozisyonunu tahmin etmektir, $\hat{y} = x_{t_{k+1}}$. Bu uygulama için zarar fonksiyonu ortalama kareli hata $l(y_{t_k}, \hat{y}_{t_k}) = \frac{1}{m} \sum_{i=1}^m (y_{t_k,i} - \hat{y}_{t_k,i})^2$ şeklinde tanımlanabilir. Kamera görüntü karesi kaçırıldığında veya cisim ile kamera arasına başka bir cisim girdiğinde ise oluşan durum ise model tanımındaki eksik bilgiye karşılık gelmektedir.

Burada sıralı tahmin etme algoritmasında tekrarlayan sinir ağları (TSA) kullanılmaktadır. Basit TSA yapısı aşağıdaki şekilde ifade edilmektedir:

$$\begin{aligned} h_{t_k} &= f(W_h x_{t_k} + R_h h_{t_{k-1}}) \\ y_{t_k} &= g(R_y h_{t_k}). \end{aligned} \quad (3)$$

$x_{t_k} \in \mathbb{R}^m$, t_k anındaki girdi vektörünü, $h_{t_k} \in \mathbb{R}^q$ durum vektörünü, $y_{t_k} \in \mathbb{R}^p$ ise çıktıyı temsil etmektedir. $W_h \in \mathbb{R}^{q \times m}$ ve $R_h \in \mathbb{R}^{q \times q}$ girdi ağırlık matrislerini, $R_y \in \mathbb{R}^{p \times q}$ is çıktı ağırlık matrisini belirtmektedir. $f(\cdot)$ ve $g(\cdot)$ lineer olmayan noktasal operasyonlar yapan fonksiyonlardır.

Bu çalışmada, özel bir TSA yapısı olan UKSB sinir ağları kullanılmıştır. UKSB sinir ağı mimarisi en basit şekliyle aşağıdaki denklemler ile modellenmektedir [6]:

$$z_{t_k} = g(W_z x_{t_k} + R_z h_{t_{k-1}}) \quad (4)$$

$$i_{t_k} = \sigma(W_i x_{t_k} + R_i h_{t_{k-1}}) \quad (5)$$

$$f_{t_k} = \sigma(W_f x_{t_k} + R_f h_{t_{k-1}}) \quad (6)$$

$$o_{t_k} = \sigma(W_o x_{t_k} + R_o h_{t_{k-1}}) \quad (7)$$

$$c_{t_k} = i_{t_k} \odot z_{t_k} + f_{t_k} \odot c_{t_{k-1}} \quad (8)$$

$$h_{t_k} = o_{t_k} \odot g(c_{t_k}), \quad (9)$$

Burada t_k anındaki girdi vektörü x_{t_k} , çıktı vektörü h_{t_k} , durum vektörü c_{t_k} olarak gösterilmektedir. z_{t_k} blok girdi kapısını, i_{t_k} , f_{t_k} ve o_{t_k} ise sırasıyla girdi, unutma ve çıktı kapılarını göstermektedir. Lineer olmayan aktivasyon fonksiyonları $g(\cdot)$, $h(\cdot)$ ve $\sigma(\cdot)$ vektörler üzerinde noktasal bazlı işlemler gerçekleştirilmektedirler. $\sigma(\cdot)$ sigmoid fonksiyonunu ifade etmekte olup $\sigma(x) = \frac{1}{1+\exp(-x)}$ şeklinde tanımlanmaktadır. $\odot$ aynı boyuttaki iki vektörün elemanlarının noktasal olarak (Hadamard) çarpımını ifade etmektedir. W_i , W_z , W_f , $W_o \in$

Şekil 1: UKSB sinir ağı mimarisi.

$\mathbb{R}^{q \times m}$ girdi ağırlık matrislerini, R_i , R_z , R_f , $R_o \in \mathbb{R}^{q \times q}$ tekrarlanan girdi ağırlık matrislerini ifade etmektedir. Burada, formül gösterimlerini sadeleştirmek amacıyla yanlılık vektörleri b_z , b_i , b_f , $b_o \in \mathbb{R}^q$ girdi ağırlık vektörlerinin içerine eklenmiştir. Burada, hedef sinyali olan y_{t_k} 'nin kestirimi

$$\hat{y}_{t_k} = W_{t_k} h_{t_k} \quad (10)$$

şeklinde yapılmaktadır. $W_{t_k} \in \mathbb{R}^{p \times q}$ UKSB sinir ağı mimarisi kullanılarak hedef sinyalin kestirimini yapan ağırlık matrisini temsil etmektedir. Burada, W_{t_k} matrisinin eğitimi, uygulamanın gereksinimlerine göre çevrimiçi veya toplu olarak yapılabilmektedir.

III. EKSİK VERİ İLE SİRALI BAĞLANIM ALGORİTMASI

Klasik UKSB sinir ağı mimarisi düzgün şekilde örneklenmiş sıralı veriyi işleyecek şekilde tasarlanmıştır. Eksik veriyi veya düzgün olmayan örnekleme modelleyecek yapıları içermemektedir. Bu duruma çözüm olarak sunulan [9], [10] gibi yapılar klasik UKSB sinir ağı, zaman bilgisini işleyecek ek kapılarla genişletirler. Takip edilen bir diğer çözüm yolu ise eksik veriyi yerine koyma metodu ile modellemektir. [7], [8] eksik girdi vektörü yerine bir önceki girdi vektörünü veya sıfır vektörünü koymaktadır.

Bu bildiride sunulan yerine koyma algoritması bir sonraki vektörü tahmin etme uygulamalarında çalışmak üzere tasarlanmıştır. Önerilen algoritmada eksik veri, kestirim yapan UKSB tabanlı sinir ağı mimarisi çıktı vektörü olarak modellenmektedir. Girdi vektörü x_{t_k} 'nin eksik olması durumunda girdi vektörü olarak $\hat{x}_{t_k} = x_{t_{k-1}}$ kullanılmaktadır. Önerilen eksik veri ile kestirim metodu Algoritma 1'de gösterilmektedir.

Algoritma 1 Kendi Tahminini Yerine Koyma Tabanlı Bağlanım Algoritması

```

1: for  $k = 1$  to  $n$  do
2:   if  $x_{t_k}$  exists then
3:      $h_{t_k} \leftarrow \text{UKSB}(x_{t_k})$ 
4:   else
5:      $h_{t_k} \leftarrow \text{UKSB}(\hat{x}_{t_k})$ 
6:   end if
7:    $\hat{x}_{t_{k+1}} = W h_{t_k}$ 
8: end for

```

Şekil 2: NYSE veri kümesi üzerinde kestirim performansı.

IV. SAYISAL ÖRNEKLER

Bu bölümde, önerilen kendi tahminini yerine koyma ile kestirim algoritmasının geleneksel yöntemler karşısındaki performansı New York Borsası (NYSE) [11], Elevators [12] gibi farklı gerçek hayat veri kümeleri üzerinden değerlendirilmiştir. Bu bölüm boyunca, kendi tahminini yerine koyma ile kestirim algoritması KTY, bir önceki girdi vektörünü yerine koyma algoritması ÖGY olarak gösterilmektedir. Bütün durumlarda sinir ağıları olasılıksal gradyan inişi (OGİ) algoritması [13] ile eğitilmiştir.

Tamamen sıfırlardan oluşmuş bir vektörü eksik girdi vektörlerinin yerine koyma algoritması sayısal örnekler içerisinde değerlendirilmemiştir. Bunun nedeni, algoritma, verinin kendisinden veya uygulamadan hiç bir girdi almadığı için ön işleme sırasındaki bir değişim algoritmanın performansında büyük değişimlere sebep olabilmektedir. Örneğin, veri üzerinde düzleştirme işlemi yapıldığında ortalama vektörleri sıfır vektörüne dönüştüğü için bu algoritma kullanıldığında aynı zamanda eksik veri için ortalama vektörü yerine koyulmuş olur. Bu durum sayesinde algoritmanın performansı iyileşme gösterse bile verinin kendisinin devamlı veya ayırık olarak dağılması gibi özellikler performansta farklılıklara sebep olabilmektedir. Ek olarak, bu bölümde karşılaştırılan algoritmalar veri kümesinin kendisinden bilgi alırken sadece sıfır vektörü ile yerine koyma işlemi gerçekleştiren algoritma veri kümesinden bilgi almamaktadır. Bu durum adil bir karşılaştırmanın önüne geçeceği için sıfır vektörünü girdi olarak kullanan algoritma sayısal örneklerle konulmamıştır.

İlk olarak, New York Borsası (NYSE) veri kümesi kullanılmıştır. Bu veri kümesinde IBM, HP, Ford gibi New York Borsası'nda işlem gören 36 farklı şirketin 3 Temmuz 1962 - 31 Aralık 1984 tarihleri arasındaki 5651 günlük (22 yıl) borsa işlem değerleri bulunmaktadır. 36 şirketin hisse değerleri günlük olarak alınmakta ertesi günün hisse değerleri tüm şirketler için tek bir sinir ağı tarafından tahmin edilmektedir. Burada, girdi vektörü $x_{t_k} \in \mathbb{R}^{36}$, girdi vektörünün boyutu $m = 36$, hedef vektörü $y_{t_k} = x_{t_{k+1}} \in \mathbb{R}^{36}$, $p = 36$, şeklindedir. Bu deney için eğitim ve test veri kümeleri günlerin ilk %60'ı eğitim setinde, devamındaki %40'ı test setinde kalacak şekilde oluşturulmuştur. İlk %60 ve sonraki %40 şeklinde ayrılıp karıştırma işlemi yapılmamasının sebebi veri kümesinde sıralı bilginin korunmak istenmesidir. OGİ algoritmasının öğrenme hızı $\eta = 0.001$ olarak çarpaz-geçerleme ile $\{0.1, 0.01, 0.001\}$

Şekil 3: Elevators veri kümesi üzerinde kestirim performansı.

arasından seçilmiştir. Aynı şekilde UKSB sinir ağına kullanılan vektörlerin boyutu $\{3, 5, 10\}$ arasından $q = 10$ olarak belirlenmiştir. Deneylerde, KTY ve ÖGY yerine koyma metotlarıyla kestirim algoritmaları sıralı verideki farklı eksiklik oranları için karşılaştırılmıştır. Deneylerde, eksik veri her bir örneğin eşit olasılık ile veriden çıkartılması ile oluşturulmuştur. -0.1 ve -0.9 sırayla %10 ve %90 eksiklik oranlarını ifade etmektedir. Şekil 2'de gösterilen sonuçlarda, eksik veri oranı %10 iken algoritmaları çok yakın performans gösterdiği gözlemlenmiştir. %10 eksik veri oranı için bildiride sunulan metod olan KTY algoritması ihmal edilebilecek düzeyde daha iyi performans göstermektedir. Ancak, Şekil 2'de de görüldüğü gibi eksiklik oranı %90 olduğu durumda algoritmalar arasındaki performans farkları daha belirginleşmektedir. Bu sonuçlar ile NYSE veri kümesi üzerinde, bildiride önerilen kendi tahminini yerine koyma algoritmasının bir önceki girdi vektörünü yerine koyma metotundan daha iyi performans verdiği görülmektedir.

Bunun dışında kullanılan bir diğer veri kümesi Elevators veri kümesidir. Bu veri kümesi bir F-16 savaş uçağının kontrolü sırasında elde edilen 18 farklı özelliği ve bunlarla ilgili bir hedef sinyali içermektedir. Yerine koyma algoritmasının uygulamasında hedef vektörü ile girdi vektörünün aynı karakterde olması gereksinimi bulunması nedeniyle hedef sinyali ek bir özellik gibi düşünülerek girdi vektörüne eklendi. Bu durumda, veri kümesinde girdi vektörü $x_{t_k} \in \mathbb{R}^{19}$, $m = 19$, çıktı vektörü $y_{t_k} \in \mathbb{R}^{19}$, $p = 19$, haline geldi. Bu veri kümesi için de eğitim ve test veri kümeleri ilk %60'ı eğitim setinde, devamındaki %40'ı test setinde kalacak şekilde kalacak şekilde oluşturulmuştur. Çarpaz geçerleme ile OGİ algoritmasının öğrenme hızı $\eta = 0.001$ ve UKSB sinir ağı vektörlerinin boyutu $q = 10$ olarak sırasıyla $\{0.1, 0.01, 0.001\}$ ve $\{3, 5, 10\}$ setlerinden seçilmiştir. Şekil 3'de -0.6 ve -0.8 sırayla %60 ve %80 eksik veri oranını göstermektedir. Şekil 3'e bakıldığında ilk olarak %80 eksik veri ile alınan sonuçların %60 eksik veri ile alınan sonuçlardan daha iyi olduğu görülmektedir. Veri incelendiğinde, veri kümesinin örneklerini elerken genel örüntüye uymayan örneklerin de elenmesinin bu duruma sebep olduğu gözlemlenmiştir. Verilerin %80'i elendiğinde kestirim yapılması görece daha zor olan örnekler de elenmiş ve %60 durumuna göre daha tahmin edilebilir örnekler kalmıştır. Ancak, algoritmaların birbirlerine göre olan performansları NYSE veri kümesi ile yapılan deneylerin sonuçları ile paralellik göstermektedir. Veri eksikliğinin daha az olduğu %60 deneyinde iki algoritmanın da kararlı durum

hatası birbirine yakın değerdedir. Yakınsama hızı açısından ise bildiride önerilen KTY algoritmasının daha iyi performans gösterdiği gözlemlenmiştir. Daha fazla eksik data barındıran %80 deney sonuçlarında ise kararlı durum hatası bakımından da, önerilen KTY algoritmasının daha iyi bir performansa sahip olduğu belirlenmiştir.

V. SONUÇLAR

Bu bildiride, eksik bilgi barındıran sıralı veri kümesinin UKSB sinir ağları ile sıralı bağlanımı çalışılmıştır. Gelecekteki bir değer tahminini amaçlayan uygulamalarda kullanılmak üzere eksik veri yerine yeni bir girdi vektörü koyma algoritması sunulmuştur. Algoritmamız, eksik verinin yerine koyacağı girdi vektörünü sıralı verideki örüntüyü kullanarak tahmin etmekte ve bu şekilde girdi olarak sinir ağına beslemektedir. Bu durumda, eksik veri için yapılan tahmin zarar fonksiyonuna etki etmezken, girdi olarak başka bir sinyalin tahmininde kullanıldığı için gradyanları iletmektedir. Böylece sistem birden fazla adımda tekrarlanan bir kestirim yapmaktadır. Yapılan deneylerde klasik algoritmalara göre kestirim performansını önemli şekilde arttırdığı gözlemlenmiştir.

KAYNAKLAR

- [1] H. Kang, "The prevention and handling of the missing data," *Korean journal of anesthesiology*, vol. 64, no. 5, pp. 402–406, 2013.
- [2] Z. Che, S. Purushotham, K. Cho, D. Sontag, and Y. Liu, "Recurrent neural networks for multivariate time series with missing values," *Scientific reports*, vol. 8, no. 1, p. 6085, 2018.
- [3] F. Eng and F. Gustafsson, "Algorithms for downsampling non-uniformly sampled data," in *Signal Processing Conference, 2007 15th European*. IEEE, 2007, pp. 1965–1969.
- [4] J. J. Benedetto and H. C. Wu, "Nonuniform sampling and spiral mri reconstruction," in *Wavelet Applications in Signal and Image Processing VIII*, vol. 4119. International Society for Optics and Photonics, 2000, pp. 130–142.
- [5] 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*, 2017.
- [6] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [7] Z. C. Lipton, D. Kale, and R. Wetzel, "Directly modeling missing data in sequences with rnns: Improved classification of clinical time series," in *Machine Learning for Healthcare Conference*, 2016, pp. 253–270.
- [8] Z. C. Lipton, D. C. Kale, C. Elkan, and R. Wetzell, "Learning to diagnose with lstm recurrent neural networks," *arXiv preprint arXiv:1511.03677*, 2015.
- [9] S. O. Sahin and S. S. Kozat, "Nonuniformly sampled data processing using lstm networks," *IEEE transactions on neural networks and learning systems*, 2018.
- [10] D. Neil, M. Pfeiffer, and S.-C. Liu, "Phased lstm: Accelerating recurrent network training for long or event-based sequences," in *Advances in Neural Information Processing Systems*, 2016, pp. 3882–3890.
- [11] "New york stock exchange dataset," http://www.cs.technion.ac.il/~rani/portfolios/NYSE_Dataset.htm, accessed: 2018-03-8.
- [12] L. Torgo, "Regression data sets," <http://www.dcc.fc.up.pt/ltorgo/Regression/DataSets.html>, accessed: 2018-03-8.
- [13] H. Jaeger, *Tutorial on training recurrent neural networks, covering BPPT, RTRL, EKF and the "echo state network" approach*. GMD-Forschungszentrum Informationstechnik, 2002, vol. 5.