



GİRESUN
ÜNİVERSİTESİ



FEN BİLİMLERİ ENSTİTÜSÜ

UZUN KISA-SÜRELİ HAFIZA DERİN YAPAY SİNİR
AĞI İLE GİRESUN İLİ MİNİMUM SICAKLIK
ÖNGÖRÜSÜ

İSTATİSTİK
ANABİLİM DALI

Yüksek Lisans Tezi

Onur DERYA
172116001

2021

GİRESUN

Yüksek Lisans Tezi

Onur DERYA

2021

**T.C.
GİRESUN ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**UZUN KISA-SÜRELİ HAFIZA DERİN YAPAY SİNİR
AĞI İLE GİRESUN İLİ MİNİMUM SICAKLIK
ÖNGÖRÜSÜ**

YÜKSEK LİSANS TEZİ

Öğrencinin Adı SOYADI : Onur DERYA
ORCID :
Tezin Enstitüye Verildiği Tarih :

Enstitü Anabilim Dalı : İstatistik Anabilim Dalı

Tez Danışmanı : Dr. Öğr. Üyesi Ali Zafer DALAR
ORCID :

**Ocak 2021
GİRESUN**

T.C.
GİRESUN ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

UZUN KISA-SÜRELİ HAFIZA DERİN YAPAY SİNİR
AĞI İLE GİRESUN İLİ MİNİMUM SICAKLIK
ÖNGÖRÜSÜ

YÜKSEK LİSANS TEZİ

Onur DERYA

Enstitü Anabilim Dalı

:

İstatistik

Bu tez 28/01/2021 tarihinde aşağıdaki jüri tarafından oybirliği ile kabul edilmiştir.

Prof. Dr.
Erol EĞRİOĞLU
Jüri Başkanı

Dr. Öğr. Üyesi
Özge GÜNDOĞDU
Üye

Dr. Öğr. Üyesi
Ali Zafer DALAR
Üye

Doç. Dr.
Bahadır KOZ
Enstitü Müdürü

BEYAN

Tez içindeki tüm verilerin akademik kurallar çerçevesinde tarafımdan elde edildiğini, görsel ve yazılı tüm bilgi ve sonuçların akademik ve etik kurallara uygun şekilde sunulduğunu, kullanılan verilerde herhangi bir tahrifat yapılmadığını, başkalarının eserlerinden yararlanılması durumunda bilimsel normlara uygun olarak atıfta bulunulduğunu, tezde yer alan verilerin bu üniversite veya başka bir üniversitede herhangi bir tez çalışmasında kullanılmadığını beyan ederim.

Onur DERYA

../../2021

TEŞEKKÜRLER

Yüksek lisans tezimin gerçekleştirilmesinde değerli bilgilerini benimle paylaşan, bu çalışmanın her aşamasında desteğini sunan, samimiyetini ve ilgisini hiç esirgemeyen danışmanım Dr. Öğr. Üyesi Ali Zafer DALAR'a teşekkürlerimi sunarım.

Ayrıca yüksek lisans eğitimim boyunca bana kattıkları her bilgi ve güzel anı için tüm hocalarıma, moral verdikleri ve destek oldukları için Ordu Üniversitesi Bilgi İşlem Daire Başkanlığı'nda çalışan tüm iş arkadaşlarıma, yoğun çalışmalarım sırasında sabır gösterdikleri için tüm dostlarıma teşekkürlerimi sunarım.

Hayatımın her noktasında bir izi olan sevgili annem Selma DERYA'ya bu süreçte de sevgisi, pozitif enerjisi ve desteği için sonsuz teşekkür ederim.

İÇİNDEKİLER

TEŞEKKÜRLER	I
İÇİNDEKİLER	II
SİMGELER VE KISALTMALAR LİSTESİ.....	IV
ŞEKİLLER LİSTESİ	V
TABLolar LİSTESİ	VI
ÖZET.....	VII
SUMMARY	VIII
BÖLÜM 1. GİRİŞ.....	1
BÖLÜM 2. GENEL BİLGİLER	7
2.1. Zaman Serisi Analizi	7
2.2. Yapay Sinir Ağları	8
2.2.1. Çok katmanlı algılayıcı.....	10
2.3. Derin Öğrenme Yöntemleri	11
2.3.1. Tekrarlayan sinir ağları	11
2.3.1.1. TSA eğitim problemleri.....	13
2.3.2. Uzun kısa-süreli hafıza	16
2.3.2.1. LSTM yapay sinir ağlarının yapısı.....	17
2.3.2.2. LSTM yönteminin adımları.....	19
2.4. Optimizasyon Algoritmaları	22
2.4.1. AdaGrad optimizasyon algoritması	22
2.4.2. RMSProp optimizasyon algoritması.....	23
2.4.3. Adam optimizasyon algoritması	23
BÖLÜM 3. MATERYAL VE YÖNTEM	26
3.1. Kullanılan Veri Seti	26
3.2. Karşılaştırmada Kullanılan Öngörü Yöntemleri.....	27
3.3. Kullanılan Programlama Dili Kütüphaneleri.....	28
3.4. Uygulanılan Yöntem.....	30

BÖLÜM 4. ARAŞTIRMA BULGULARI	35
BÖLÜM 5. TARTIŞMA VE SONUÇ	42
KAYNAKLAR.....	43
ÖZGEÇMİŞ	47



SİMGELER VE KISALTMALAR LİSTESİ

σ	: Sigmoid Fonksiyonu
α	: Öğrenme Hızı
ϵ	: Epsilon
API	: Uygulama Programlama Arayüzü (Application Programming Interface)
CPU	: Merkezi İşlem Birimi / İşlemci (Central Processing Unit)
DIFTS	: Derin Sezgisel Bulanık Zaman Serisi (Deep Intuitionistic Fuzzy Time Series)
BZS	: Bulanık Zaman Serisi (Fuzzy Time Series)
GPU	: Grafik İşlemci Birimi (Graphics Processing Unit)
HKO	: Hata Kareler Ortalaması (Mean Squared Error, MSE)
HKOK	: Hata Kareler Ortalamasının Karekökü (Root Mean Squared Error, RMSE)
HMO	: Hatanın Mutlak Ortalaması (Mean Absolute Error, MAE)
LSTM	: Uzun Kısa-Süreli Hafıza (Long Short-Term Memory)
MIT	: Massachusetts Teknoloji Enstitüsü (Massachusetts Institute of Technology)
PS	: Pi-sigma
ReLU	: Doğrultulmuş Doğrusal Birim (Rectified Linear Units)
RMSprop	: Kareler Ortalamasının Karekökü Yayılımı (Root Mean Square Propagation)
TSA	: Tekrarlayan Sinir Ağı (Recurrent Neural Network, RNN)
SGİ	: Stokastik Gradyan İnişi (Stochastic Gradient Descent, SGD)
SMN	: Tek Çarpımsal Nöron (Single Multiplicative Neuron)
YAK	: Yapay Arı Kolonisi (Artificial Bee Colony, ABC)
YSA	: Yapay Sinir Ağı (Artificial Neural Network, ANN)

ŞEKİLLER LİSTESİ

Şekil 2.1. Biyolojik sinir hücresi yapısı	8
Şekil 2.2. Basit bir yapay sinir ağı yapısı.....	9
Şekil 2.3. Çok katmanlı algılayıcı yapısı	10
Şekil 2.4. Bir TSA modelinin açık hali	11
Şekil 2.5. TSA çalışma şeması	12
Şekil 2.6. Üç katmanlı bir TSA modeli	14
Şekil 2.7. Tek katmanlı standart bir TSA modeli	16
Şekil 2.8. Dört katmanlı LSTM ağ modeli.....	17
Şekil 2.9. Ağ geçişi çizgi hatları.....	17
Şekil 2.10. LSTM hücre durumu.....	18
Şekil 2.11. LSTM ağ modelinin kapı gösterimi	18
Şekil 2.12. LSTM unutma kapısı katmanı	19
Şekil 2.13. LSTM’de yeni bilginin hücre durumunda saklanması.....	20
Şekil 2.14. LSTM’de bilginin unutulması ve yeni bilginin aktarılması.....	21
Şekil 2.15. LSTM’de çıktı değerinin oluşturulması.....	21
Şekil 3.1. Veri seti grafiği.....	30
Şekil 3.2. Eğitim ve test seti.....	31
Şekil 3.3. LSTM katmanı oluşturma kod parçası.....	32
Şekil 3.4. Erken durdurma kod parçası	33
Şekil 4.1. Döngü sayısı grafiği.....	36
Şekil 4.2. Uygulanan model katman yapısı.....	36
Şekil 4.3. Modelin ağırlıklarının gösterimi	38
Şekil 4.4. Uygulanan yöntemden elde edilen öngörü değerleri	40

TABLÖLAR LİSTESİ

Tablo 3.1. Kütüphane versiyonları	29
Tablo 4.1. Nöron sayısı karşılaştırma tablosu	35
Tablo 4.2. Gecikme sayısı karşılaştırma tablosu	35
Tablo 4.3. Dense katmanı ağırlıklarının değerleri.....	37
Tablo 4.4. LSTM modelinin katman ağırlıklarının değerleri	39
Tablo 4.5. Karşılaştırma tablosu.....	41

UZUN KISA-SÜRELİ HAFIZA DERİN YAPAY SİNİR AĞI İLE GİRESUN İLİ MİNİMUM SICAKLIK ÖNGÖRÜSÜ

ÖZET

Son yıllarda zaman serisi öngörüsünde derin öğrenme yöntemlerinin klasik zaman serisi yöntemleri, bulanık zaman serisi yöntemleri ve diğer yapay sinir ağları yöntemlerine göre daha iyi sonuçlar verdiği görülmektedir. Derin öğrenmenin gelişimi ile yapay sinir ağlarında oluşan yenilikçi ağ modellerinden biri ve aynı zamanda bir derin öğrenme yöntemi olan uzun kısa-sürelili hafıza yöntemi literatürde zaman serisi analizinde kullanılmaya başlanmıştır.

Yapay sinir ağlarında olduğu gibi derin öğrenme yöntemlerinde de oluşturulan modele ait başlangıç değerleri ve modelin içerisindeki ağ ağırlık değerleri modelin zaman serisi öngörü performansına etki etmektedir. Bu değerlerin belirlenmesi öngörü doğruluğu konusunda önem arz etmekte olup literatürde çözülmesi gereken bir problem olarak karşımıza çıkmaktadır.

Bu tez kapsamında; başlangıç değerlerinin üretilmesinde Xavier yöntemini, katman ağırlıklarının optimize edilmesinde ise Adam optimizasyon algoritmasını kullanan tek katmanlı uzun kısa-sürelili hafıza yaklaşımı kullanılmış ve bu yaklaşımın öngörü performansını göstermek için Giresun ili günlük ortalama minimum sıcaklık gerçek hayat zaman serisi kullanılmıştır. Bu yaklaşım, literatürde var olan diğer bazı öngörü yöntemleri ile öngörü performansı açısından karşılaştırılmış ve yaklaşımın diğer yöntemlere göre daha üstün öngörü performansına sahip olduğu görülmüştür.

Anahtar kelimeler: Derin öğrenme, uzun kısa-sürelili hafıza, yapay sinir ağları, öngörü

MINIMUM WEATHER TEMPERATURE FORECASTING OF GİRESUN PROVINCE USING LONG SHORT-TERM MEMORY DEEP ARTIFICIAL NEURAL NETWORK

SUMMARY

In recent years, it has been observed that deep learning methods give better results in time series forecasting compared to classical time series methods, fuzzy time series methods and other artificial neural network methods. With the development of deep learning, long short-term memory, which is one of the innovative network models formed in artificial neural networks and is also a deep learning method, has started to be used in time series analysis in the literature.

As in artificial neural networks, the initial values of the model created in deep learning methods and the weight values in the model affect the time series forecasting performance of the model. It is important to determine these values for forecasting accuracy and they appear as problems to be solved in the literature.

Within the scope of this thesis, the single layered long short-term memory approach, using the Xavier method in generating the initial values of the model and the Adam algorithm for optimizing the layer weights, was utilized. The daily average minimum temperature real-world time series in Giresun province was used to show the forecasting performance of the approach. The approach was compared with some other forecasting methods in the literature in terms of forecasting performance and it was found that the approach has a superior forecasting performance compared to other methods.

Keywords: Deep learning, long-short term memory, artificial neural networks, forecasting

BÖLÜM 1. GİRİŞ

Geçmişte ve günümüzde yaşadığımız olaylarla ilişkiler kurarak gelecek hakkında öngörülerde bulunuruz. Hayatımızdaki tesadüf faktörünü ortadan kaldırmaya çalışarak tecrübelerimizle ileriye dönük adımlarımızı tahmin ederek planlarız. Fizikçi ve filozof Niels Bohr, “tahmin zordur, özellikle de gelecek hakkında” demiştir. Tarihte insanların hava durumunu tahmin etmek için çevrelerindeki canlıların tepkilerinden varsayımlar çıkardığını düşündüğümüzde öngöründe bulunmak gerçekten zor diyebiliriz.

Teknolojik cihazların gelişmesi ile öngöründe bulunmak istediğimiz alan hakkında veri elde edebildiğimiz ölçüm cihazları ortaya çıkmıştır. Elde edilen verilerle istatistiksel analizler ortaya konarak zaman serilerinde öngörü çalışmaları önem kazanmaya başlamıştır. Verinin giderek büyümesi ile analiz yöntemleri de gelişmeye başlamıştır. Yapay sinir ağları (YSA’lar) ve derin öğrenme uygulamaları bu alanda hızla büyüyen tekniklerdir.

YSA modeli ilk kez McCulloch ve Pitts (1943) tarafından ortaya koyulmuştur. Geliştirilen bu tek katmanlı YSA modelleri doğrusal olmayan problemlerin çözümünde başarısız olmaktadır. Rumelhart ve ark. (1986) çalışmasında doğrusal olmayan ilişki gösteren XOR problemi üzerinden çalışmalar yapılarak çok katmanlı algılayıcı (ÇKA) modeli geliştirilmiştir.

YSA modelleri ile birçok alanda zaman serisi tahmin çalışmaları yapılmaktadır. Keskin Benli (2005) çalışmasında, bankalarda mali başarısızlığın öngörülmesinde lojistik regresyon ve YSA karşılaştırılması yapılarak YSA ile daha başarılı sonuçlar alındığı belirtilmiştir. Çuhadar ve ark. (2009) çalışmasında, Antalya ili dış turizm talebi tahminlerinde üstel düzleştirme ve Box-Jenkins yöntemleri ile YSA modeli tahmin doğrulukları karşılaştırılarak YSA ile daha iyi sonuçlar elde edilmiştir. Kaynar ve ark. (2011) çalışmasında, doğalgaz tüketim tahmininde ÇKA ile ARIMA modeli

karşılaştırılarak ÇKA ile daha iyi sonuçlar elde edilmiştir. Zhang (2003)'de, literatürde sıklıkla kullanılan veri setlerinden olan Lynx, Sunspot ve Borsa endeksi veri setleri üzerinde ARIMA, YSA ve ARIMA ile YSA'nın birleşiminden oluşan melez model arasında karşılaştırma yapılmıştır. Melez yöntemin diğer yöntemlerden daha iyi sonuçlar elde ettiği gösterilmiştir. Egrioglu ve ark. (2013) tarafından bulanık kümeleme, yapay sinir ağları ve genetik algoritma yöntemleri kullanılarak birleştirilen öngörü yöntemi ile Alabama Üniversitesi kayıt verileri için yapılan çalışmada literatürde önerilen diğer bulanık zaman serisi yaklaşımları ile karşılaştırılmış ve önerilen yöntem ile daha iyi sonuçlar elde edilmiştir.

Tekrarlayan sinir ağları (TSA), zaman serisi öngörü uygulamalarında daha iyi performans göstermesini sağlayan sıralı ve tekrarlı olma özelliklerine sahip yapay sinir ağlarıdır. TSA ilk kez basit tekrarlayan ağ olarak Elman (1990) tarafından önerilmiştir (Elmas, 2018). Hochreiter (1991) ve Bengio ve ark. (1994) tarafından TSA ile ilgili uzun süreli öğrenme problemleri ile gradyanların güncellenmesi sorunlarına değinilmiştir (Olah, 2015).

Tekrarlayan sinir ağlarında ortaya çıkan gradyan sorununu çözen ve uzun sıralı verileri belleğinde tutabilen uzun kısa-süreli hafıza (LSTM) modeli, ilk olarak Hochreiter ve Schmidhuber (1997) tarafından tanıtılmıştır. LSTM, konuşma tanıma (Cai ve Liu, 2016), dil modelleme ve metin tahmini (Soutner ve Müller, 2013), duygu analizi (Wang ve ark., 2016) ve çeviri (Alla ve ark., 2020) gibi alanlarda kullanılmaktadır. Bunların yanı sıra LSTM ile zaman serisi öngörü çalışmaları farklı alanlarda literatürde sıklıkla kullanılmaktadır. Sezer ve ark. (2020) çalışmasında finansal zaman serileri öngörü uygulamasına ilişkin derin öğrenme çalışmalarının kapsamlı bir literatür taraması sağlanmış ve son yıllarda derin öğrenme yöntemleri ile zaman serileri öngörü çalışmalarının artışına değinilmiştir. Yapılan literatür taraması kapsamında derin öğrenmede en çok kullanılan model türünün TSA olduğu görülmekte ve TSA modelleri arasından da en çok LSTM tercih edilmektedir.

Liu ve ark. (2020)'de, LSTM'nin optimal hiper parametrelerinin genetik algoritma ile belirlendiği grupsal ampirik mod ayrışımı tabanlı LSTM ile petrol üretimi öngörüsü

yapılarak sonuçları grupsal ampirik mod ayrışımı tabanlı YSA ve destek vektör makinesi yöntemleri ile karşılaştırılmış ve LSTM ile daha iyi sonuçlar elde edilmiştir.

Bao ve Yue (2017) çalışmasında hisse senedi fiyatları öngörü çalışmasında ilk olarak zaman serisindeki gürültüyü azaltmak için dalgacık dönüşümü ile seri ayrıştırılarak yığılmış otokodlayıcı içeren bir melez LSTM modeli önerilmiştir. Oluşturulan melez model ile sadece serideki gürültü azaltılarak uygulanan LSTM modeli, TSA ve LSTM ile karşılaştırılmış ve önerilen melez LSTM ile daha iyi sonuçlar elde edilmiştir.

Abdel-Nasser ve Mahmoud (2019), yenilenebilir enerji kaynaklarından biri olan fotovoltaik enerji gücünün öngörüsünün elde edildiği çalışmada LSTM ile doğrusal regresyon, karar ağacı ve YSA yöntemleri karşılaştırılmış ve LSTM ile daha başarılı sonuçlar elde edilmiştir.

Livieris ve ark. (2020), altın fiyatı öngörü çalışmasında bir diğer derin öğrenme modeli olan evrişimli sinir ağları ile LSTM modeli birleştirilerek melez bir model önerilmiştir. Çalışmada destek vektör regresyonu, ileri beslemeli sinir ağı, farklı hiper parametrelerle oluşturulan dört LSTM modeli ve evrişimli sinir ağları ile birleştirilen iki LSTM modeli karşılaştırılmıştır. LSTM modelinin öngörüdeki başarısına vurguda bulunularak önerilen melez LSTM modeli ile daha iyi sonuçlar elde edildiği ve uygulamada denenilen farklı hiper parametrelerle yöntemin tahmin yeteneğinin daha da geliştiği belirtilmiştir. Zaytar ve Amrani (2016)'da, Fas'taki 9 şehir için 24 ve 72 saatlik hava durumu verilerinde öngörüde bulunmak için LSTM modeli kullanılmıştır. LSTM modellenmesinde Theano kütüphanesinin işlevselliği ve web tabanlı uygulamalarda kullanabilirliğine vurgu yapılarak LSTM modelinin hava durumu öngörüsünde geleneksel yöntemlere göre daha iyi performans gösterdiği ve yeni sistemlerde standart haline gelebilecek bir yöntem olduğu belirtilmiştir.

Xu ve ark. (2019) çalışmasında, iç mekan hava sıcaklığını tahmin etmek için hata düzeltme modeli ile birleştirilen LSTM modeli geri yayımlı sinir ağı, destek vektör makinesi, karar ağacı ve LSTM modeli yöntemleri ile karşılaştırılmıştır. LSTM modelinin iç mekan sıcaklığını tahmin etmede diğer yöntemlerden daha iyi olduğu ve

modifiye edilen LSTM modeli ile belirgin bir iyileşme elde edildiği belirtilmiştir. Karevan ve Suykens (2020)'de, Belçika'daki 5 şehre ait minimum ve maksimum sıcaklık, çiy noktası ve rüzgar hızı gibi hava değişkenlerinin öngörüsü çalışmasında test noktası yakınındaki değerlerin modelin eğitimi üzerinde daha yüksek etkiye sahip olduğunun kabul edildiği transdüktif LSTM modeli önerilmiş ve LSTM ile karşılaştırılmıştır. Önerilen modelin birçok durumda LSTM'den daha iyi performans gösterdiği belirtilmiştir. Pandit ve ark. (2020) çalışmasında, dalga boyu ve rüzgar hızı öngörüsü için LSTM ile Markov modeli karşılaştırılmıştır. Dalga boyu tahmininde LSTM modeli ile daha iyi sonuç elde edilirken rüzgar hızı tahmininde Markov modeli ile daha iyi sonuç elde edilmiştir. LSTM modellerinin varyasyonlarına değinilerek farklı modellemeler ile gelişmiş LSTM modellerinde farklı sonuç elde edilebileceği vurgulanmıştır. Kocak ve ark. (2020) çalışmasında Giresun iline ait minimum hava sıcaklığı değerleri öngörüsünde sezgisel bir bulanık zaman serisi ile kullanılan melez LSTM modeli ile literatürdeki sinir ağı modelleri ve bulanık zaman serisi yöntemleri ile karşılaştırmışlar ve önerilen model ile daha iyi sonuçlar elde edilmiştir.

Sinir ağlarında girdi ağırlıklarının sabit kullanılmayıp optimize edilerek işleme alınması çıktı değerlerini iyileştirmede önemli katkı sağlamıştır. Ağırlık değerlerinin optimize edilmesi işlemi derin öğrenme yöntemlerinde de kullanılmaktadır. Chang ve ark. (2018) çalışmasında, elektrik fiyatı tahmininde Adam algoritması ile optimize edilmiş LSTM modeli önerilmiştir. Çalışmada stokastik gradyan inişi (SGİ), RMSprop ve Adam algoritmaları ile optimize edilmiş LSTM modelleri, genelleştirilmiş regresyon sinir ağı ve geri yayılım sinir ağı yöntemleri karşılaştırılmış ve Adam algoritması ile optimize edilmiş LSTM modelinin daha iyi sonuçlar elde ettiği görülmüştür. Chang ve ark. (2019)'da elektrik fiyatı tahmininde dalgacık dönüşümü ile ayrıştırılmış ve Adam algoritması ile optimize edilmiş bir melez LSTM modeli önerilmiştir. İlk adımda LSTM modeline uygun optimize ediciyi bulmak için SGİ, RMSprop ve Adam algoritmaları karşılaştırılmış ve Adam algoritmasının daha iyi performans gösterdiği belirtilmiştir. Sonrasında önerilen melez LSTM model ile istatistik yöntem ARMA ve aşırı öğrenme makineleri ile birleştirilmiş bir başka melez model ile karşılaştırma yapılmış ve önerilen LSTM modeli daha iyi sonuçlar vermiştir. Diferansiyel evrim algoritması ile birleştirilmiş bir başka LSTM modeli ve ARIMA

yöntemi ile birleştirilmiş yapay sinir ağı modeli ile de farklı veri kısımları ile karşılaştırma yapıldığında dalgacık dönüşümü ve Adam algoritması ile optimize edilmiş LSTM modelinin daha iyi sonuçlar verdiği görülmüştür.

Sinir ağlarında girdi ağırlıklarının başlangıç değerleri modelin eğitiminde önemli rol oynamaktadır. Glorot ve Bengio (2010) çalışmasında ağırlık başlangıcının kaybolan gradyan problemi üzerindeki etkilerine, rastgele başlangıç değerlerinin standart gradyan inişinin derin sinir ağlarında zayıf olduğuna ve doğrusal olmayan aktivasyon fonksiyonlarının etkilerine değinilerek yeni bir başlangıç değerleri yöntemi önerilmiştir. Gao ve ark. (2017) çalışmasında tekrarlayan sinir ağlarında girdi değişkenlerinin hisse senedi tahmininde daha etkili bir yöntem sağlayabileceği hipotezi ile Adam algoritması ve Xavier başlangıç değerleri kullandıkları LSTM modeli ile hareketli ortalama, üstel hareketli ortalama ve destek vektör makinesi yöntemleri karşılaştırılmış ve LSTM ile daha iyi sonuçlar elde edilmiştir. Gao ve Chai (2018)'de, hisse senedi kapanış fiyatlarının tahmin edilmesinde temel bileşen analizi ile teknik göstergeler için boyut küçültme yaparak Adam algoritması ve Xavier başlangıç değerleri kullandıkları LSTM modeli önerilmektedir. Önerilen LSTM modeli, dört farklı teknik göstergeye ayrılarak ileri beslemeli sinir ağları, destek vektör makineleri gibi yöntemlerle karşılaştırılmış ve önerilen LSTM modelinin daha iyi performans gösterdiği belirtilmiştir.

Halpern-Wight ve ark. (2020)'de, güneş fotovoltaik enerji sistemi için enerji öngörü çalışmasında tek katmanlı LSTM ile 5 katmanlı LSTM modeli karşılaştırılmıştır. Her iki model de benzer performans göstermiştir. Tek katmanlı LSTM modelinin performansına vurgu yapılarak tek katmanlı LSTM modellerinin kullanımı çok katmanlı modellere göre daha ekonomik olacağından tercih edilebilir olduğu belirtilmiştir.

Bu tez kapsamında; modelin başlangıç değerlerinin üretilmesinde Xavier yöntemini, katman ağırlıklarının optimize edilmesinde ise Adam algoritmasını kullanan tek katmanlı LSTM yaklaşımı uygulanmış ve uygulanan yaklaşımın öngörü performansını göstermek için Giresun ili günlük ortalama minimum sıcaklık gerçek hayat zaman

serisi kullanılmıştır. Uygulanan yaklaşım, literatürde var olan diğer bazı öngörü yöntemleri ile öngörü performansı açısından karşılaştırılmış ve yaklaşımın diğer yöntemlere göre daha üstün öngörü performansına sahip olduğu görülmüştür.

Tez çalışmasının ikinci bölümünde genel bilgiler başlığı altında; zaman serisi analizi, yapay sinir ağları, derin öğrenme yöntemlerinden tekrarlayan sinir ağları ve uzun kısa-süreli hafıza yöntemleri ve optimizasyon algoritmaları olarak AdaGrad, RMSProp ve Adam optimizasyon algoritmaları tanıtılmıştır. Üçüncü bölümde; kullanılan veri setinden, karşılaştırmada kullanılan öngörü yöntemlerinden, uygulamada kullanılan programlama dili kütüphanelerinden ve uygulanan yöntemden bahsedilmiştir. Dördüncü bölümde, uygulanan yöntemin performansını göstermek için Giresun iline ait 01.01.2017 ile 31.04.2017 tarihleri arası günlük ortalama minimum sıcaklık değerleri zaman serisi kullanılarak uygulanan yöntem, literatürde yer alan diğer bazı öngörü yöntemleri ile karşılaştırılmıştır. Son bölümde ise sonuçlar yorumlanarak, uygulanan yöntemin performansına değinilmiştir.

BÖLÜM 2. GENEL BİLGİLER

Bu bölümde bazı zaman serisi analizi yöntemlerinden, bulanık zaman serisi analizi yaklaşımlarından, optimizasyon yöntemlerinden biri olan Adam optimizasyonundan, yapay sinir ağlarından ve derin öğrenme yöntemlerinden bahsedilmiştir.

2.1. Zaman Serisi Analizi

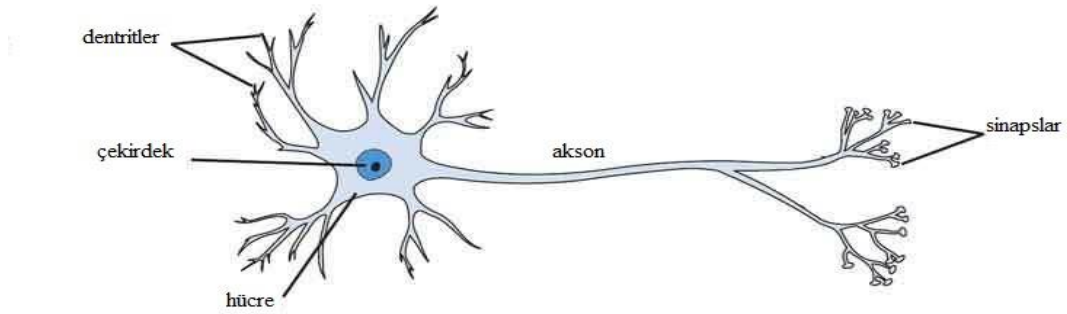
Zaman serisi, her biri belirli bir t zamanında kaydedilen gözlemler kümesidir. İlk çağdan beri insanlar, tıpta, hava durumunda, ekonomide ve astronomi alanında zamansal değişimleri takip ederek veriler oluşturmuş ve bunları anlamlandırmaya çalışmıştır. Günümüzde zaman serisi verileri ve analizi, nesnelerin interneti (Internet of Things), sağlık hizmetlerinin dijitalleşmesi, akıllı şehirlerin yükselişi ve benzeri yenilikçi gelişim ile giderek daha önemli hale gelmektedir. Sürekli izleme (continuous monitoring) ve veri toplama daha yaygın hale geldikçe hem istatistiksel hem de makine öğrenme ve derin öğrenme teknikleriyle zaman serisi analizine olan ihtiyaç artmaktadır. Tüm bu alanlar geniş kapsamlı zaman serisi verileri sunduğundan insan davranışı, özel sektör verileri, çevresel ve iklimsel değişimlerini analiz etmek ve öngöründe bulunmak için yeni zaman serisi analizi yöntemleri de gelişmeye devam etmektedir.

Öngörü, bir değişkenin geçmiş ve günümüz verilerine dayanarak gelecekteki değerlerinin tahmin edilmesidir. Öngörü amaçlı çalışmalar ilk başlarda basit modelleme denemeleri ile gerçekleşmiş daha sonrasında teknolojinin sağladığı avantajlar ve büyük veri kullanılarak daha karmaşık modeller kurularak daha iyi sonuçlar alınmasını sağlamıştır. Literatürde yer alan gelişmiş yapay sinir ağı modelleri ile yapılan çalışmaların klasik zaman serisi yöntemlerinden daha başarılı sonuçlar elde etmesi buna örnektir.

2.2. Yapay Sinir Ağları

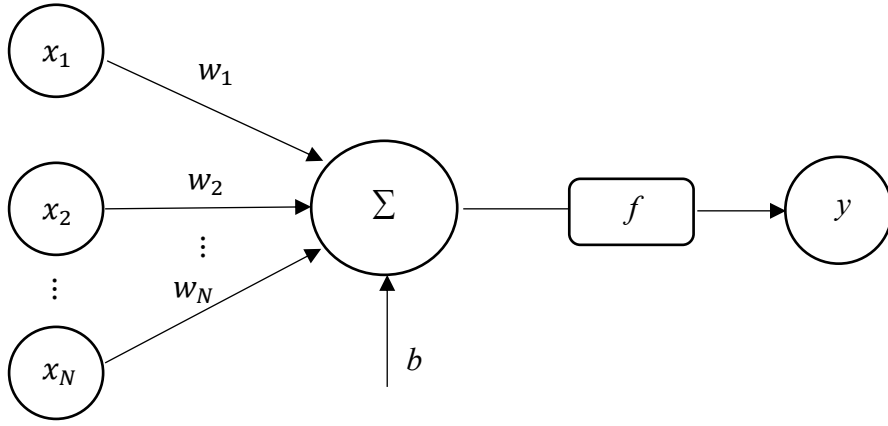
Yapay sinir ağları, beyindeki biyolojik sinir ağlarının çalışmasından esinlenerek geliştirilmiş bilgi işleme sistemleridir. YSA'nın ilk matematiksel modelinin temelleri McCulloch ve Pitts'in 1943 yılında yayınladıkları makaleyle atılmıştır. YSA; öğrenme, ilişkilendirme, sınıflandırma, genelleme, öngörü ve optimizasyon gibi farklı konularda uygulanmaktadır.

Sinir ağının temel unsuruna nöron denir. Şekil 2.1'de gösterildiği gibi bir nöron, dentritler, hücre ve akson olarak üç ana bölümden oluşur. Dentritler her bir dalının bir nörona bağlı olduğu etrafındaki nöronlardan sinyal alan yapılardır. Akson, sinyali bir nörondan diğerine ileten ince bir silindirik yapıya sahiptir. Aksonun sonunda dentritlerle temas sinapslar aracılığı ile oluşur. Sinaptaki nöronlar arası sinyallerin birbirlerine pozitif veya negatif ağırlık etkisi vardır. Bir sinaptan nörona gelen dürtü sinyali sinapsların toplam ağırlık eşiğini aşabiliyorsa aksona sinyal gönderir (Kawaguchi, 2000).



Şekil 2.1. Biyolojik sinir hücresi yapısı

Bir YSA temel olarak giriş katmanı, gizli katman ve çıkış katmanından oluşmaktadır. Basit bir YSA mimarisi Şekil 2.2'de verilmiştir. Giriş katmanında bulunan girdiler $X_1, X_2, \dots, X_N$ şeklindedir. Gizli katmanda her girdi bağlantı ağırlık değerleriyle çarpılır. Ağırlık değerleri, girdiyi pozitif veya negatif yönde etkileyebilir. Ağırlık değerleri ile çarpılan girdiler ön bilgi (bias) ile toplanır ve son olarak elde edilen sonuç bir aktivasyon fonksiyonundan geçerek çıkış katmanına aktarılır.



Şekil 2.2. Basit bir yapay sinir ağı yapısı

Yapay sinir ağı matematiksel olarak ifade edildiğinde;

- N : Girdi sayısı
- x : Girdi değeri
- w : Ağırlık
- b : Ön bilgi
- f : Aktivasyon fonksiyonu
- y : Çıkış değeri

olmak üzere,

$$y = f\left(\sum_{i=1}^N x_i w_i + b\right) \quad (2.1)$$

şeklinde ifade edilir (Zayegh ve Bassam, 2018).

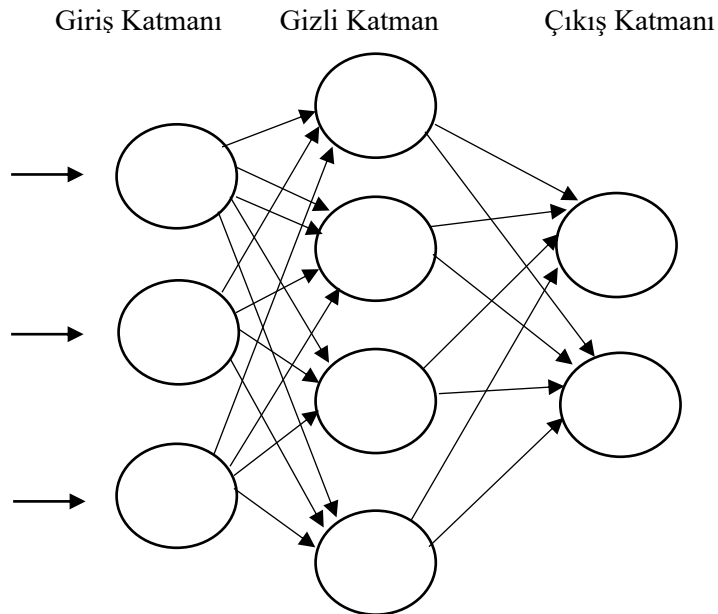
İleri beslemeli sinir ağları, birimler arasında bağlantıların bir döngü oluşturmadığı yapay sinir ağlarıdır. Bilgi, döngü oluşturmadan ağda yalnızca ileriye doğru hareket eder ve önce giriş katmanından sonrasında eğer varsa gizli katmanlardan ve son olarak çıkış katmanından geçer. Tek katmanlı basit bir yapay sinir ağı sadece girdi ve çıktı katmanlarından oluşmaktadır.

Geri yayılım sağlayan yapay sinir ağlarında bir döngü oluşur. Gradyan inişi kullanarak denetimli öğrenme sağlarlar. Ağ üzerinde geriye doğru hareket ettikçe son ağırlık katmanının gradyanından hesaplamaya başlayarak ilk ağırlık katmanının gradyanına doğru ağırlıklar güncellenir.

2.2.1. Çok katmanlı algılayıcı

Tek katmanlı algılayıcıların doğrusal olmayan problemlerin çözümünde başarısız olması ve günlük hayatta karşılaşılan problemlerin çoğunun doğrusal olmayan nitelikler taşıması üzerine doğrusal olmayan bir ilişki gösteren XOR problemini çözmek amacıyla Rumelhart ve arkadaşları (1986) tarafından çok katmanlı algılayıcı modeli geliştirilmiştir.

Çok katmanlı algılayıcılar, Şekil 2.3'te gösterildiği gibi 3 temel bölümden oluşur. Giriş katmanında bilgi işleme gerçekleşmez, girdiler gizli katmana iletilir. Giriş katmanında her bir girdi, gizli katmandaki elemanların tümüne bağlıdır. Gizli katmanda, gelen bilgiler işlenir. Bir adet gizli katman ile bir problem çözülebileceği gibi problem türüne göre birden fazla gizli katmana da ihtiyaç duyulabilir. Çıkış katmanında, gizli katmandan gelen bilgiler işlenerek dışarıya veya bir başka ağa girdi olarak iletilir.



Şekil 2.3. Çok katmanlı algılayıcı yapısı

2.3. Derin Öğrenme Yöntemleri

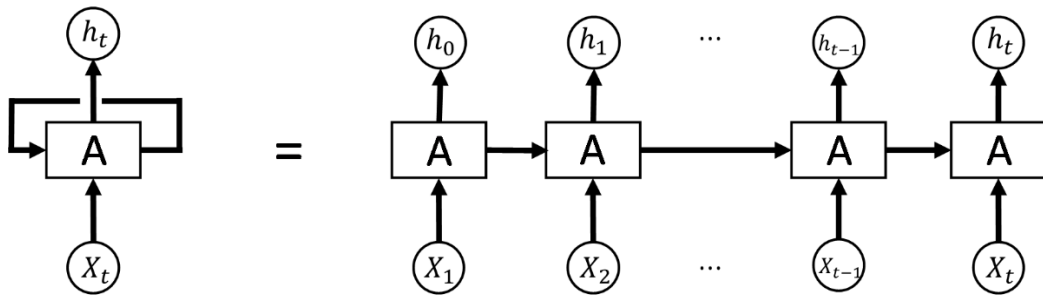
Teknolojideki donanımsal gelişmeler ile daha fazla nöron ve katmana sahip YSA'nın eğitilmesi ve çalıştırılması da mümkün hale gelmiştir. Daha fazla derinliğe ve genişliğe sahip bu ağlar derin sinir ağları, eğitimleri de derin öğrenme olarak adlandırılmaya başlanmıştır (Elmas, 2018).

Derin öğrenme modelleri birçok farklı alanda kullanılmaktadır. Evrişimli sinir ağları; görüntü işleme, örüntü tanıma ve doğal dil işleme gibi alanlarda kullanılır. Derin inanç ağı; örüntü tanıma, bilgi çıkarma ve doğal dil işleme gibi alanlarda uygulanır. Metin, ses, zaman serisi gibi ardışık verilerin işlenmesinde ise çoğunlukla tekrarlayan sinir ağları kullanılmaktadır. Bu çalışmada da zaman serisi analizinde tekrarlayan sinir ağlarının bir türü olan uzun kısa-sürelili hafıza modeli kullanılmıştır.

2.3.1. Tekrarlayan sinir ağları

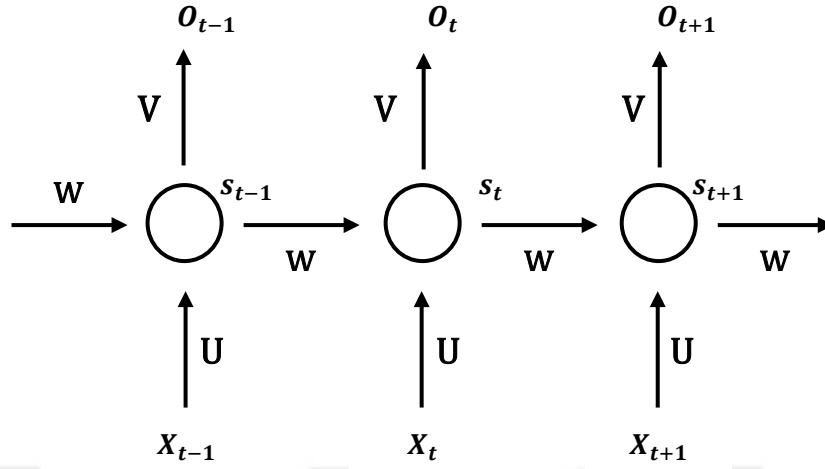
TSA, metin tahmini, dil modelleme, konuşma tanıma, çeviri, duygu analizi ve zaman serisi analizi gibi çalışmalarda kullanılan bir tür yapay sinir ağıdır. TSA, tekrarlama katmanı ile mimariye döngüler eklenmiş bir ÇKA ağı olarak görülebilir.

TSA'nın içinde barındırdığı tekrarlama katmanı, yapay sinir ağının kullanımları arasında durum bilgisinin saklanması ve tekrar kullanılmasını sağlamaktadır. Bu durum ileri beslemeli bir ağın çıkış bilgisinin saklanarak bir sonraki yapay sinir ağı kullanımında girdi olarak sisteme tekrar verilmesi şeklinde düşünülebilir. TSA yapısının açılmış (ileri beslemeli ağ olarak ifade edilen) hali Şekil 2.4'te verilmiştir.



Şekil 2.4. Bir TSA modelinin açık hali

TSA yapısının çalışma sistemi ve sistemdeki parametreler Şekil 2.5'te verilmiştir.



Şekil 2.5. TSA çalışma şeması

Burada;

- x_t : t anındaki girdi değişkeni
- s_t : t anındaki gizli durum (hidden state). Ağın hafıza kısmını oluşturur.

s_t , bir önceki hesaplanan gizli durum ve mevcut girdi değeri ile denklem 2.2'deki gibi bulunur.

$$s_t = f(U x_t + W s_{t-1}) \quad (2.2)$$

Burada f lineer olmayan bir ***tanh*** ya da ***ReLU*** (doğrultulmuş doğrusal birim) fonksiyonu olabilir ve

$$o_t : \text{softmax}(V * s_t) \quad (2.3)$$

ile hesaplanabilir. Softmax fonksiyonu, normalleştirilmiş üstel fonksiyon olarak değerleri 0 ile 1 arasında tutar. Çıktıları olasılık dağılımına normalleştirmek için genellikle sinir ağlarının son aktivasyon fonksiyonu olarak kullanılır.

s_t önceki adımlardaki bilgiyi bir sonraki katmana taşımak için saklı tutar. Adımların sonunda o_t , t anındaki bütün bilgiyi kullanarak elde edilen çıktıyı oluşturur. Diğer

yapay sinir ağlarının aksine TSA, (U, V, W) parametrelerini ağ boyunca kullanır. Burada;

- U: gizli katmanın ağırlık değerlerini
- V: çıkış katmanının ağırlık değerlerini
- W: gizli durum geçişinin ağırlık değerlerini

ifade etmektedir. Bu yönüyle bir dizi halinde girdiler kullanıldığında, aynı hedefe ulaşmak için her bir adım birlikte çalışmış olur. Böylece öğrenme için gerekli çok sayıdaki parametre indirgenmiş olur (Karakoyun, 2018).

2.3.1.1. TSA eğitim problemleri

TSA etkili bir modeldir ancak TSA ile iyi çıktılar alabilmek için uygun mimariyi belirleyebilmek önemlidir. TSA'nın eksi yönlerinden biri, kaybolan ve patlayan gradyan problemidir ve bu yüzden uzun süreli girdilerin öğrenilmesinde güçlük çekmektedir (Hermans ve Schrauwen, 2013).

Uzun süreli bağımlılık problemleri

TSA, dahili hafızası sayesinde geçmiş ile bağlantı kurarak anlamlandırma sağlamakta ve böylece zaman bazlı problemlerde başarılı sonuçlar elde etmektedir. Ancak TSA'da hangi aktivitelerin ve bu aktivitelerin ne kadar süre hafızasında tutulacağı bilinmemektedir. Tüm girdiler TSA'nın dahili hafızasında tutulmaktadır. Hafızasındaki bazı bilgiler önemli iken bazı bilgiler gereksizdir. Bu yüzden bazı aktivitelerin sınıflandırılmasında tüm geçmişin saklanması gerek yoktur.

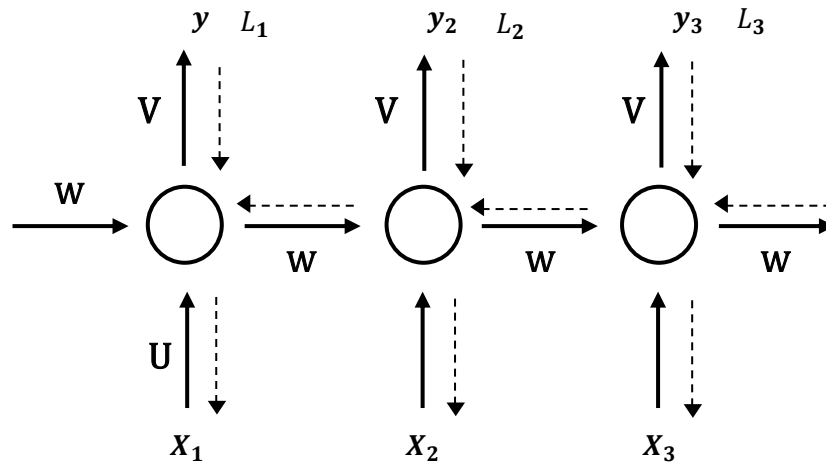
Aktivite sınıflandırılmasında gerekli bilgi çok önceden oluşmuş ise bu bilgiye ulaşamayabilir. Bazı durumlarda ise bir önceki karedeki olay ile şimdiki karedeki olay birbiri ile bağlantılı olabilir. Örneğin; aktivite sınıflandırması yapılacak olan videoda, konferansa gelen katılımcılar salona girip koltuklara oturuyor olsun. TSA böyle bir örnekte önceden sırayla koltuklara oturmuş olan katılımcılardan, bir sonraki katılımcının da koltuğa oturacağını kolaylıkla tahmin edebilir. Tahmin edilen aktivite yakın zamanda gerçekleşmiştir.

Bazı aktiviteler daha karmaşık olabilir. Yukarıda verilen örnekte tahmin edilen aktivite konferans başladıktan sonra gelen bir katılımcının aktivitesi olduğunda geç gelen katılımcının, salona girip koltuğa oturacağını tahmininin yapılması insanlar için zor değildir. Ancak TSA için koltuğa oturma aktivitesi bitip araya başka aktiviteler girdiği için yeni gelen katılımcının oturacağını tahmin etmek zor olabilir (Elit, 2018).

Teorik olarak TSA bu tür uzun süreli bağımlılıkları kontrol etme yeteneğine sahiptir. Ancak bunun için parametrelerin çok dikkatli seçilmiş olması gerekmektedir. Pratikte bu pek mümkün olmadığından TSA uzun süreli geçmişini hatırlamakta problem yaşamaktadır (Olah, 2015).

Kaybolan ve patlayan gradyan problemi

Geleneksel yapay sinir ağlarında olduğu gibi TSA'yı da eğitmek için geri yayılım işlemi gerekmektedir. Parametreler tüm zaman adımlarında kullanıldığından her bir çıktıdaki gradyan sadece mevcut zaman adımına değil kendinden önceki zaman adımlarına da bağlıdır.



Şekil 2.6. Üç katmanlı bir TSA modeli

Şekil 2.6’da gösterilen üç katmanlı TSA modelinin ileri yayılımı sırasında elde edilen değerler ile geri yayılım sırasında yeni ağırlık değerleri hesaplanır. İleri yayılım

sırasında ağ, her zaman adımında L_t kaybını hesaplamak için tahminler üretir. Geri yayılım sırasında ağırlık parametreleri bakımından kaybın gradyanları her bir zaman adımında hesaplanır.

Aşağıdaki (2.4) nolu denklem, gizli durum geçişinin ağırlıklarını temsil eden W 'ye göre kaybın gradyanını göstermektedir.

$$\frac{dL}{dW} = \sum_{t=0}^n \frac{dL_t}{dW} \quad (2.4)$$

Şekil 2.6'da gösterilen TSA modelinin son zaman adımında ($t=3$) kaybın gradyanına baktığımızda bu gradyan, aşağıdaki denklem adımlarındaki gibi zincir kuralı kullanılarak üç alt gradyana ayrılabilir (Anwer, 2017).

$$\frac{\partial L_3}{\partial W} = \frac{\partial L_3}{\partial y_3} \cdot \frac{\partial y_3}{\partial h_2} \cdot \frac{\partial h_2}{\partial W} \quad (2.5)$$

$$\frac{\partial L_3}{\partial W} = \sum_{t=0}^2 \frac{\partial L_3}{\partial y_3} \cdot \frac{\partial y_3}{\partial h_2} \cdot \frac{\partial h_2}{\partial h_t} \cdot \frac{\partial h_t}{\partial W} \quad (2.6)$$

$$\frac{\partial L_3}{\partial W} = \sum_{t=0}^2 \frac{\partial L_3}{\partial y_3} \cdot \frac{\partial y_3}{\partial h_2} \cdot \left(\prod_{j=t+1}^2 \frac{\partial h_j}{\partial h_{j-1}} \right) \cdot \frac{\partial h_2}{\partial W} \quad (2.7)$$

Uzun süreli olarak birden fazla zaman adımında geri yayılım işlemi sürekli yenilenir ise gradyanların birden az olması durumunda sonuç giderek küçülmekte ve kaybolan gradyan problemine neden olmaktadır. Aynı durumda gradyanların birden büyük olması durumunda ise sonuç giderek büyümekte ve patlayan gradyan problemine neden olmaktadır.

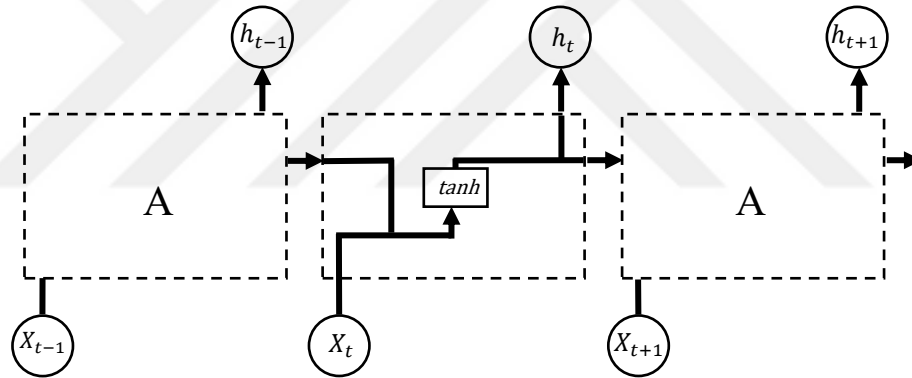
Tahmin edilemeyen büyüklükteki gradyanlar, optimizasyon algoritmalarının kararlılığını etkiler. Aşırı derecede küçük gradyanların ise parametrelere bir etkisi olmadığında öğrenmeyi olumsuz etkiler. Bu yüzden TSA uzun süreli bağımlılıklarda

öğrenme zorluğu yaşar. Bu sorun geleneksel yapay sinir ağlarında da oluşabilir ancak TSA geri yayılımın gerçekleştiği daha fazla katmana sahip olduğundan bu durum TSA’da daha belirgindir (Zhang ve ark., 2020).

2.3.2. Uzun kısa-sürelî hafıza

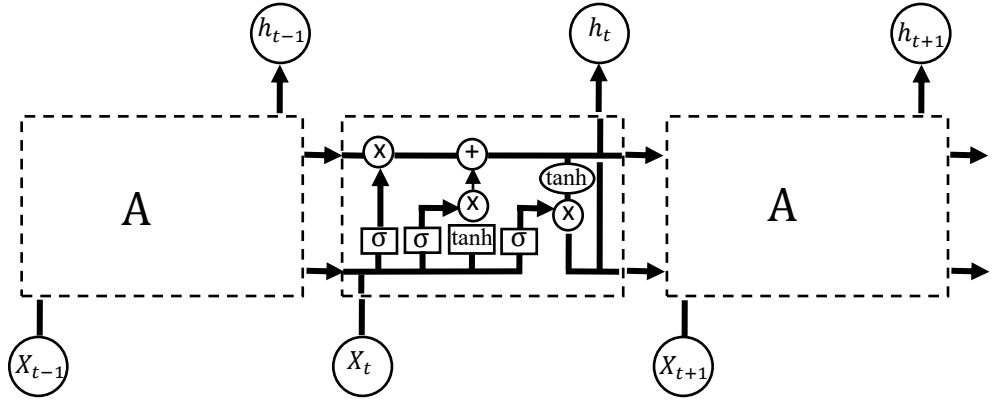
Uzun kısa-sürelî hafıza, uzun süreli bağımlılıkları öğrenebilen özel bir TSA türüdür. LSTM ilk olarak Hochreiter ve Schmidhuber tarafından 1997’de tanıtılmıştır. LSTM, uzun süreli bağımlılık sorununun önüne geçmek için tasarlanmıştır.

Tüm tekrarlayan sinir ağları, sinir ağının modüllerinden oluşan bir zincir biçimindedir. Şekil 2.7’de gösterildiği gibi standart TSA’da bu tekrarlayan modül, tek bir *tanh* katmanı gibi çok basit bir yapıya sahiptir.



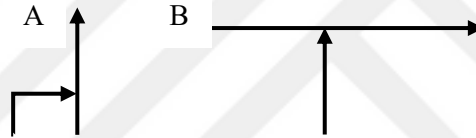
Şekil 2.7. Tek katmanlı standart bir TSA modeli (Olah, 2015)

LSTM de bu zincir benzeri yapıya sahiptir ancak tekrarlayan modülün farklı bir yapısı vardır. Şekil 2.8’de gösterildiği gibi tek bir sinir ağı katmanı yerine çok özel bir şekilde yerleştirilmiş dört katman bulunmaktadır.



Şekil 2.8. Dört katmanlı LSTM ağ modeli (Olah, 2015)

Şekil 2.8'deki her satır, bir düğümün çıktısından diğerlerinin girişlerine kadar tüm vektörleri taşır. Modelin altında bulunan dikdörtgen şeklindeki kutular öğrenilmiş sinir ağı katmanlarını, üstündeki daireler ise vektör eklenmesi gibi noktasal işlemleri temsil eder.



Şekil 2.9. Ağ geçişi çizgi hatları

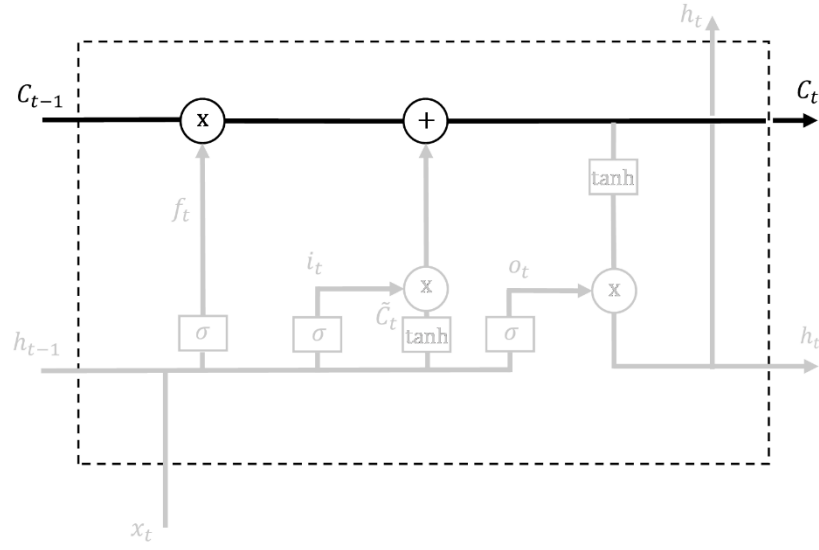
Şekil 2.9 biçim A'da gösterilen çizgiler birleştirme işlemini, biçim B'de ise çizginin çatallanması içeriğin kopyalanması ve kopyaların farklı yerlere gideceği anlamına gelmektedir. Ayrıca;

- $\otimes$: Hadamard çarpımını (vektörün eleman-eleman çarpımı)
- $\oplus$: Elemanter toplamını (vektörün eleman-eleman toplamı)

göstermektedir (Olah, 2015).

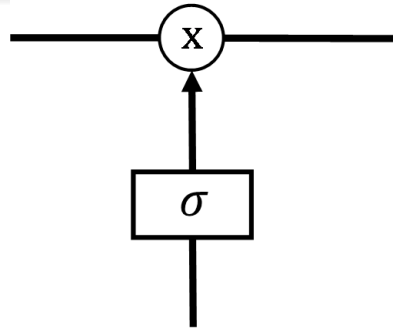
2.3.2.1. LSTM yapay sinir ağlarının yapısı

LSTM yönteminin uzun süreli bağımlılık sorununu çözümlemedeki en önemli bileşeni hücre durumu bilgisidir. Şekil 2.10'da diyagram boyunca üstten yatay olarak geçen çizgi hücre durumunu temsil etmektedir.



Şekil 2.10. LSTM hücre durumu (Olah,2015)

LSTM, kapılar adı verilen yapılar tarafından düzenlenmiş hücre durumuna bilgi ekleme veya kaldırma kabiliyetine sahiptir. Kapılar, isteğe bağlı olarak bilginin aktarılmasına izin vermek için bir yoldur. Şekil 2.11’de gösterilen kapı, sigmoid sinir ağı katmanını ve noktasal çarpma işleminden oluşur.



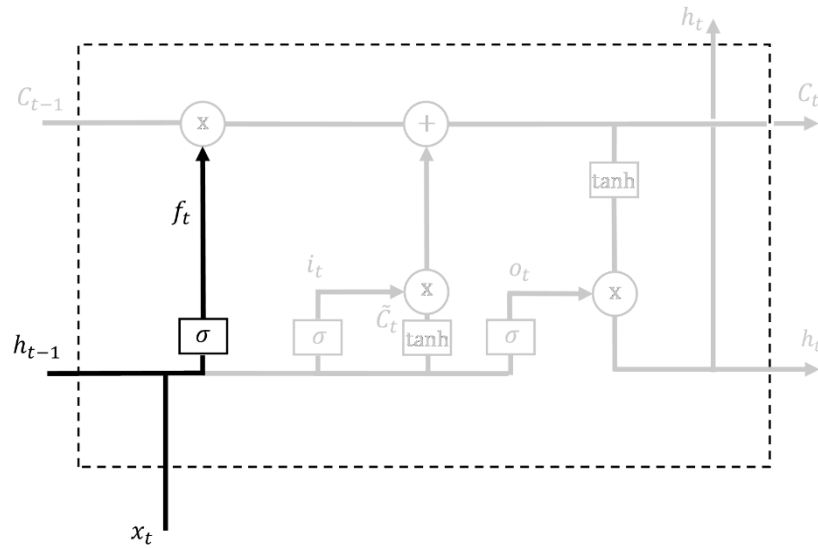
Şekil 2.11. LSTM ağ modelinin kapı gösterimi (Olah,2015)

Sigmoid katman bilgiyi sıfır ile bir arasında sayılara dönüştürür ve her bileşene bilginin ne kadarına geçiş izni verileceğini açıklar. 0 değeri “hiçbir şeyin geçmesine izin verilmemesi”; 1 değeri ise “her şeye izin verilmesi” anlamına gelmektedir. LSTM, hücre durumunu korumak ve kontrol etmek için bu kapılardan üçüne sahiptir (Olah, 2015).

2.3.2.2. LSTM yönteminin adımları

LSTM’de ilk adım, hangi bilgilerin hücre durumundan atılacağına belirlenmesidir. Bu karar, unutma kapısı katmanını (forget gate layer) adı verilen bir sigmoid katmanını tarafından gerçekleştirilir. h_{t-1} bir önceki blok çıktı değeri, x_t girdi değeri ve b_f ön bilgi değeri olmak üzere; f_t unutma kapısı fonksiyonu denklem 2.8 ile hesaplanır. Unutma kapısı, h_{t-1} ve x_t ’ye bakarak C_{t-1} hücre durumundaki her sayı için 0 ve 1 arasında bir sayı verir. 1 değeri “bunu tamamen koru”, 0 değeri ise “bundan tamamen kurtul” anlamına gelmektedir. Tüm hesaplamalar boyunca kullanılan $W_{f,x}$, $U_{f,h}$, $W_{c,x}$, $U_{c,h}$, $W_{i,x}$, $U_{i,h}$, $W_{o,x}$, $U_{o,h}$ değerleri ilgili kapıların ağırlık değerlerine karşılık gelmektedir.

$$f_t = \sigma (W_{f,x} x_t + U_{f,h} h_{t-1} + b_f) \quad (2.8)$$



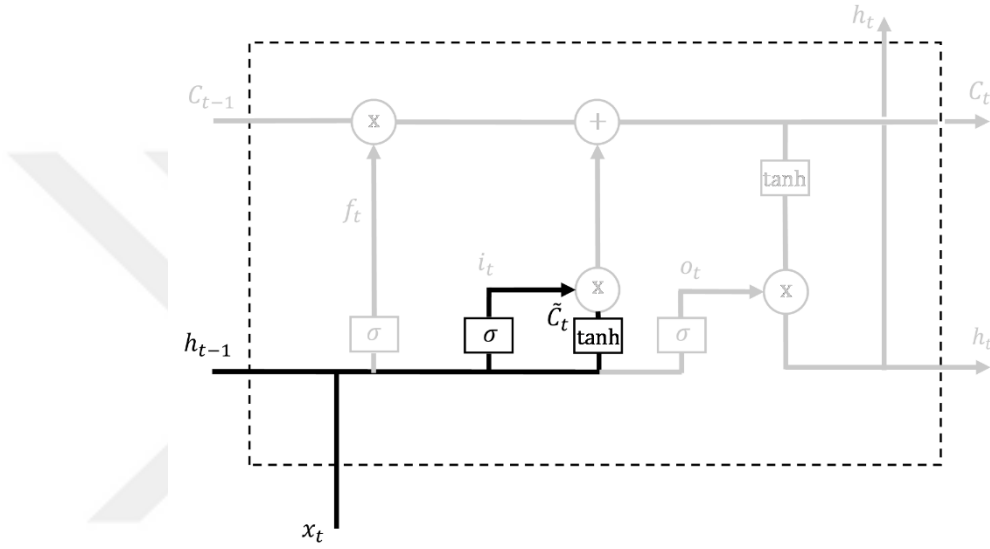
Şekil 2.12. LSTM unutma kapısı katmanı (Olah, 2015)

Sonraki adımda hücre durumunda hangi yeni bilgilerin saklanacağına karar verilmektir. Bu karar verme süreci iki aşamada gerçekleşir. Önce giriş kapısı katmanını (input gate layer) olarak adlandırılan sigmoid katmanını hangi değerlerin güncelleneceğine karar verir. Daha sonra $\tanh$ katmanını hücre durumuna eklenebilecek yeni aday değerlerin ($\hat{C}_t$) vektörünü oluşturur. Sonraki adımda Şekil 2.12’de gösterildiği gibi hücre durumunu güncellemek için bu ikisi birleştirilir. i_t giriş kapısı

değeri, $\hat{C}_t$ aday değer, b_i giriş ön bilgi ve b_c hücre ön bilgi değeri olmak üzere denklem 2.9 ve 2.10'daki gibi hesaplanır.

$$i_t = \sigma(W_{i,x} x_t + U_{i,h} h_{t-1} + b_i) \quad (2.9)$$

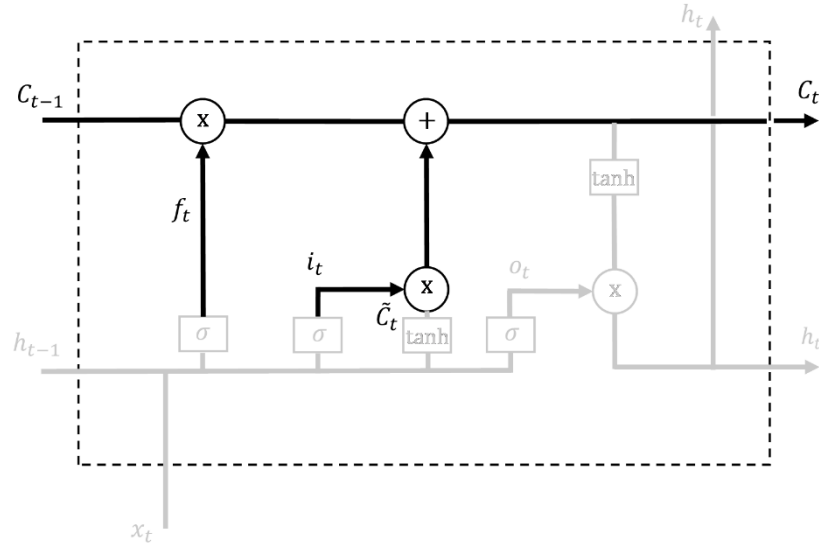
$$\hat{C}_t = \tanh(W_{c,x} x_t + U_{c,h} h_{t-1} + b_c) \quad (2.10)$$



Şekil 2.13. LSTM'de yeni bilginin hücre durumunda saklanması (Olah, 2015)

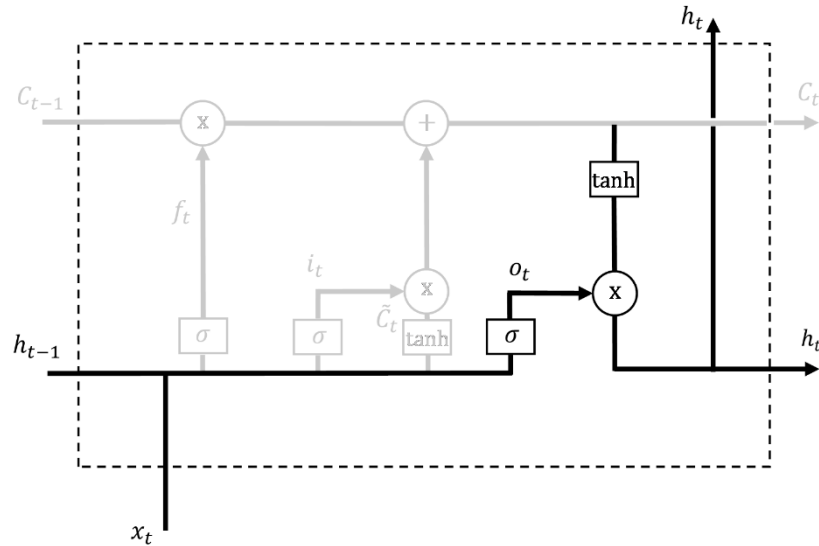
Bu aşamada; C_{t-1} eski hücre durumu, C_t yeni hücre durumuna güncellenmiş olur. C_{t-1} eski durum değeri ile f_t unutma kapısı değeri $\otimes$ işlemi vasıtasıyla çarpılır. Böylece önceki adımda unutulmasına karar verdiğimiz bilgiler unutulur. i_t giriş kapısı değeri ile $\hat{C}_t$ yeni hücre aday değeri $\otimes$ işleminden geçirilir. Sonra bu iki değer toplanır. Yeni elde edilen bu değer, her bir durum değerinin ne kadar güncellenmesine karar verilen ve sonraki bloğa aktarılmak üzere hafızaya alınan değer olur.

$$C_t = f_t \otimes C_{t-1} + i_t \otimes \hat{C}_t \quad (2.11)$$



Şekil 2.14. LSTM’de bilginin unutulması ve yeni bilginin aktarılması (Olah, 2015)

Son aşamada çıktının ne olacağına karar verilmesi gerekmektedir. Çıktı, kapı durumuna bağlı olarak belirlenir ancak bu değer filtrelenmiş bir değerdir. İlk olarak hücre durumunun hangi kısımlarının çıktısının verileceğine karar verilmesini sağlayan bir sigmoid katman çalıştırılır. Sonrasında hücre durumuna $\tanh$ fonksiyonu uygulanarak değerlerin -1 ile 1 arasında olması sağlanır. Ardından sigmoid katman ile çarpılarak çıktının karar verilen kısmı için h_t değeri elde edilmiş olur (Olah, 2015).



Şekil 2.15. LSTM’de çıktı değerinin oluşturulması (Olah, 2015)

o_t ve h_t çıktı değeri, b_o ön bilgi çıktı değeri olmak üzere sırasıyla denklem 2.12 ve 2.13'teki gibi hesaplanır.

$$o_t = \sigma (W_{o,x} x_t + U_{o,h} h_{t-1} + b_o) \quad (2.12)$$

$$h_t = o_t \otimes \tanh(C_t) \quad (2.13)$$

2.4. Optimizasyon Algoritmaları

Optimizasyon, ağıın ürettiği çıkış değeri ile gerçek değer arasındaki farkı minimuma indirmek için kullanılan yöntemdir. Optimizasyon yöntemleri ile hata fonksiyonunun mutlak minimum değeri bulunmaktadır. Bu değer derin öğrenme çalışmalarında, öğrenme işleminin sağlıklı bir şekilde sonuçlanmasını sağlar.

2.4.1. AdaGrad optimizasyon algoritması

AdaGrad algoritması, her parametre için öğrenme oranlarının her birini gradyanın geçmiş karesel toplamının karekökü ile ters orantılı bir şekilde ölçekleyerek uyarlar.

Gradyanların t zaman adımında elde edilmesi,

$$g_{t,i} = \nabla_{\theta_i} J(\theta_{t,i}) \quad (2.14)$$

parametrelerin güncellenmesi,

$$\theta_{t+1,i} = \theta_{t,i} - \frac{\alpha}{\sqrt{G_{t,ii}} + \epsilon} g_{t,i} \quad (2.15)$$

şeklindedir. Güncelleme işleminde karesel gradyanların birikimi, öğrenme oranında erken ve gereğinden fazla azalmalara neden olabilir (Goodfellow ve ark., 2018).

2.4.2. RMSProp optimizasyon algoritması

RMSProp algoritması, AdaGrad yönteminin gradyan birikimi yaklaşımını üstel ağırlıklı hareketli ortalama yaklaşımı ile değiştirilmesiyle oluşturulmuştur. Tüm gradyan setini kullanmak yerine her adımda hesaplanan gradyanlar üzerinde belli bir çerçeve boyutu kullanılmaktadır (Goodfellow, et al., 2018).

Gradyanların t zaman adımında elde edilmesi,

$$g_t = \nabla_{\theta_t} J(\theta_t) \quad (2.16)$$

karesel gradyanların toplanması,

$$E[g^2]_t = \rho E[g^2]_{t-1} + (1 - \rho) g_t^2 \quad (2.17)$$

parametrelerin güncellenmesi,

$$\theta_{t+1} = \theta_t - \frac{\alpha}{\sqrt{E[g^2]_t + \epsilon}} g_t \quad (2.18)$$

şeklinde hesaplanır.

2.4.3. Adam optimizasyon algoritması

İlk olarak 2014 yılında Kingma ve Ba tarafından ortaya atılan Adam optimizasyon algoritmasında popüler iki optimizasyon yönteminin avantajları birleştirilmiştir. AdaGrad'ın seyrek gradyanlarla başa çıkma yeteneği ve RMSProp'un sabit olmayan hedeflerle başa çıkma yeteneği birleştirilerek algoritmanın hızlı, uygulaması kolay ve

makine öğrenimi alanındaki çok çeşitli dışbükey olmayan optimizasyon problemlerine uygun olduğu belirtilmiştir.

Adam optimizasyon algoritması, eğitim verilerine dayalı ağ ağırlıklarını güncellemek için klasik stokastik gradyan iniş yerine kullanılabilen bir yöntemdir. Ağırlıklar güncellenirken aşağıdaki parametreler bazı varsayılan değerler ile kullanılır.

- α : Adım boyutu veya öğrenme hızı. Varsayılan değer 0,001 olarak alınmıştır.
- β_1 : Birinci moment kestirimi için üstel azalma oranı. Varsayılan değer 0,9 olarak alınmıştır.
- β_2 : İkinci moment kestirimi için üstel azalma oranı. Varsayılan değer 0,999 olarak alınmıştır.
- $f(\theta)$: θ parametresi ile stokastik hedef fonksiyonu
- θ_0 : İlk parametre vektörü
- m_0 : Birinci moment vektörü
- v_0 : İkinci moment vektörü
- ϵ : Sıfıra bölünmeyi önlemek için kullanılan değer, epsilon. Varsayılan değer 10^{-8} olarak alınmıştır.

Ağırlıkların güncellenme adımları;

t zaman adımında stokastik hedefe göre gradyanların elde edilmesi,

$$g_t = \nabla_{\theta} f_t(\theta_{t-1}) \quad (2.19)$$

yanlı birinci moment kestiriminin güncellenmesi,

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \quad (2.20)$$

yanlı ikinci ham moment kestiriminin güncellenmesi,

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \quad (2.21)$$

yanlılığı düzeltilmiş birinci moment kestiriminin hesaplanması,

$$m'_t = \frac{m_t}{1 - \beta_1^t} \quad (2.22)$$

yanlılığı düzeltilmiş ikinci moment kestiriminin hesaplanması,

$$v'_t = \frac{v_t}{1 - \beta_2^t} \quad (2.23)$$

ve parametrelerin güncellenmesi

$$\theta_{t+1} = \theta_t - \frac{\alpha}{\sqrt{v'_t} + \epsilon} m'_t \quad (2.24)$$

şeklinde gerçekleştirilir (Kingma ve Ba, 2017).

BÖLÜM 3. MATERYAL VE YÖNTEM

Zaman serilerinde öngörü çalışmaları geleneksel istatistiksel yöntemlerle uzun zamandır gerçekleştirilmektedir. Son zamanlarda popülerliği artan yapay sinir ağları ve derin öğrenme modelleri ile de öngörü çalışmaları yapılmaktadır. Özel mimarisi sayesinde zaman serilerinde sıkça kullanılmaya başlanan LSTM de bu çalışmada literatürdeki diğer bazı yöntemler ile karşılaştırılarak yöntemin zaman serisi öngörüsündeki performansı test edilmiştir.

Bu tez çalışmasında LSTM modeli için Python programlama dili kullanılmıştır. Model ilk olarak çeşitli programlama dillerinde canlı ön izlemeler ve düzenleme için bir ortam sağlayan açık kaynak kodlu bir program olan JupyterLab üzerinde Python 3.8.3 versiyonu ile geliştirilmiştir. Modeli çalıştırma kısmında ise online olarak Python ve Python notebook uzantılı dosyaları açma, düzenleme ve çalıştırma imkânı sunan, GPU desteği sağlayan ve birçok Python kütüphanesini içerisinde barındıran Google Colaboratory (Colab) kullanılmıştır. Colab üzerinde model çalıştırılırken Python 3.6.9 versiyonu ve yapılan çalıştırmalarda kaydedilen donanımsal özelliklerde çoğunlukla Tesla T4 model GPU ve Intel Xeon 2.00GHz model CPU (Merkezi İşlem Birimi) kullanılmıştır.

3.1. Kullanılan Veri Seti

Bu çalışmada hava sıcaklığı değerleri üzerinden öngörü çalışması yapılmıştır. Giresun Meteoroloji Bölge Müdürlüğü tarafından 01.01.2017 ile 30.04.2017 tarihleri arasında saatlik olarak kaydedilen Giresun iline ait minimum hava sıcaklığı değerleri alınarak aritmetik ortalama ile günlük minimum sıcaklık değerleri elde edilmiştir. Dönüştürülen veriler ile zaman serisi öngörü çalışması gerçekleştirilmiştir.

3.2. Karşılaştırmada Kullanılan Öngörü Yöntemleri

Uygulanan yöntemin öngörü performansını karşılaştırmak için Koçak ve ark. (2020) çalışmasında kullanılan aşağıdaki açıklaması yapılan öngörü yöntemleri kullanılmıştır.

- MLP-ANN, verilerin öngörüsü veya tahmin edilmesinde sıklıkla kullanılan ileri beslemeli çok katmanlı algılayıcı yapay sinir ağıdır.
- PS-ANN, verilerin öngörüsü veya tahmin edilmesinde sıklıkla kullanılan Shin ve Gosh (1991) tarafından önerilen yüksek dereceli bir pi-sigma yapay sinir ağıdır. Bu ağ, daha az sayıda ağırlık ve işlem birimi kullanarak çıktı birimlerini oluşturur. Yapılan çalışmada, karmaşık yaklaşım ve sınıflandırma problemlerinin sadece üç veya dört toplama birimi kullanılarak pi-sigma ağı tarafından çözülebileceği görülmektedir. Eğitim sürelerinin önemli ölçüde azalmış bir ağ ile sonuçlanacağı ve toplama birimlerinin karmaşıklık düzeyine ulaşmak için kademeli olarak eklenebileceği görülmektedir.
- S&C-FTS, literatürdeki ilk kez Song ve Chissom (1993) tarafından önerilen bulanık zaman serileri yöntemidir. Karmaşık matris işlemlerine dayanmaktadır. Yapılan çalışmada, öngörülen değerlerin modelin çıktıları hakkındaki yorumlanmasına bağlı olması eksiklik olarak belirtilmesine rağmen bulanık kümeler teorisinin karmaşık problemleri çözmek için başarılı bir yöntem olduğu vurgulanmaktadır.
- C-FTS, literatürde bulanık mantık ilişkilerine dayanan Chen (1996) tarafından önerilen bulanık zaman serisi tahmin modelidir. Yapılan çalışmada, Song ve Chishom (1993) çalışmasındaki karmaşık maksimum-minimum birleşim işlemlerinden ziyade basitleştirilmiş aritmetik işlemler kullanılarak daha etkin bir yöntem olduğu belirtilmektedir.
- FTS-N, Bas ve ark. (2015) tarafından önerilen doğrusal ve doğrusal olmayan zaman serilerini tahmin etmek için farklı bir bulanık zaman serisi ağına sahiptir. Bulanık zaman serileri yönteminin ve otoregresif yöntemin hibrit yaklaşımı önerilmektedir.
- FTS-SMN, tek bir çarpımsal nöron model yapay sinir ağına dayanan Aladağ (2013) tarafından önerilen bir bulanık zaman serisi yöntemidir. Önerilen yöntemde modelin eğitiminde bulanık ilişkileri tanımlamak için çarpımsal

nöron modeli ve tahmin doğruluğunu artırmak için parçacık sürü optimizasyonu kullanılmaktadır.

- FTS-PS, pi-sigma yapay sinir ağına dayanan Bas ve ark. (2018) tarafından önerilen bir bulanık zaman serisi yöntemidir. Bulanık ilişkileri belirlemede değiştirilmiş parçacık sürü optimizasyonu kullanılmıştır.
- IFTS-PS-ABC, pi-sigma yapay sinir ağları ve yapay arı kolonisine dayanan Eğrioğlu ve ark. (2019) tarafından önerilen sezgisel bir yüksek dereceli bulanık zaman serisi tahmin yöntemidir. Sezgisel bulanık zaman serisi modellerinde bir zaman serisini analiz etmede gerçek gözlemleri, üyelik değerleri ve üyelik dışı değerleri bir arada kullanılmaktadır.
- DIFTS-LSTM, LSTM'ye dayalı Kocak ve ark. (2020) tarafından önerilen yeni bir yüksek dereceli derin sezgisel bulanık zaman serisi yöntemidir. Önerilen yöntemde üyelik ve üyelik dışı değerler sezgisel bulanık kümeleme kullanılarak elde edilmektedir. Gecikmeli değerler, LSTM'in girdilerini oluşturmaktadır.

3.3. Kullanılan Programlama Dili Kütüphaneleri

Pandas, veri yapıları sunan ve veri üzerinde analiz yapmayı sağlayan kullanımı kolay, hızlı ve esnek bir açık kaynak kodlu Python kütüphanesidir.

NumPy (Numerical Python), çok boyutlu diziler ve matrislerle çalışmayı sağlayan ve matematiksel işlemler yapabilen açık kaynak kodlu bir Python kütüphanesidir.

Matplotlib, statik, animasyonlu ve etkileşimli görseller oluşturmak için kapsamlı bir Python kütüphanesidir. Çoğunlukla 2D grafikler ve *NumPy* kütüphanesi için bir çizim kütüphanesidir.

Sklearn veya *Scikit-learn*, makine öğrenmesi modelleri oluşturmada kullanılan, regresyon, kümeleme ve sınıflandırma için kullanılan pek çok öğrenme algoritmasına sahip veri işlemede kullanılan ve *NumPy* ile uyumlu çalışabilen bir Python kütüphanesidir.

Keras, Python için geliştirilmiş bir derin öğrenme kütüphanesidir ve derin öğrenme modellerinin oluşturulması ve eğitilmesi için uygun ortam sağlar. Aynı kodun değiştirilmeden hem CPU’da hem de GPU’da çalışmasını sağlar. *TensorFlow* ve *Theano* kütüphaneleri ile uyumlu ve derin öğrenme modellerinin prototipinin oluşmasını hızlıca sağlayan API’ye (Uygulama Programlama Arayüzü) sahiptir. Evrişimli ağlar, yinelemeli ağlar ve her ikisinin beraber kullanımı için önceden tanımlı desteğe sahiptir. Birçok farklı ağ yapısını, çoklu girdi ya da çoklu çıktıyı, katman veya model paylaşımını uygulayabilmek mümkündür. Keras, MIT lisansı (Massachusetts Teknoloji Enstitüsü tarafından hazırlanan açık kaynak ve özgür yazılım lisansı) ile dağıtılmaktadır ve bu sayede ticari projeler dahil her yerde serbest olarak kullanılabilir (Chollet, 2018).

TensorFlow, Google tarafından ilk olarak makine öğrenimi ve derin öğrenme araştırmaları için geliştirilmiş ancak birçok alanda da uygulanabilen veri akış grafiklerini kullanarak sayısal hesaplama yapan açık kaynaklı bir programlama kütüphanesidir. *TensorFlow* ile yapay sinir ağları oluşturabilir ve bu ağlar eğitilebilir.

Bu çalışmada kullanılan kütüphanelere ait versiyon bilgileri aşağıdaki tabloda verilmiştir.

Tablo 3.1. Kütüphane versiyonları

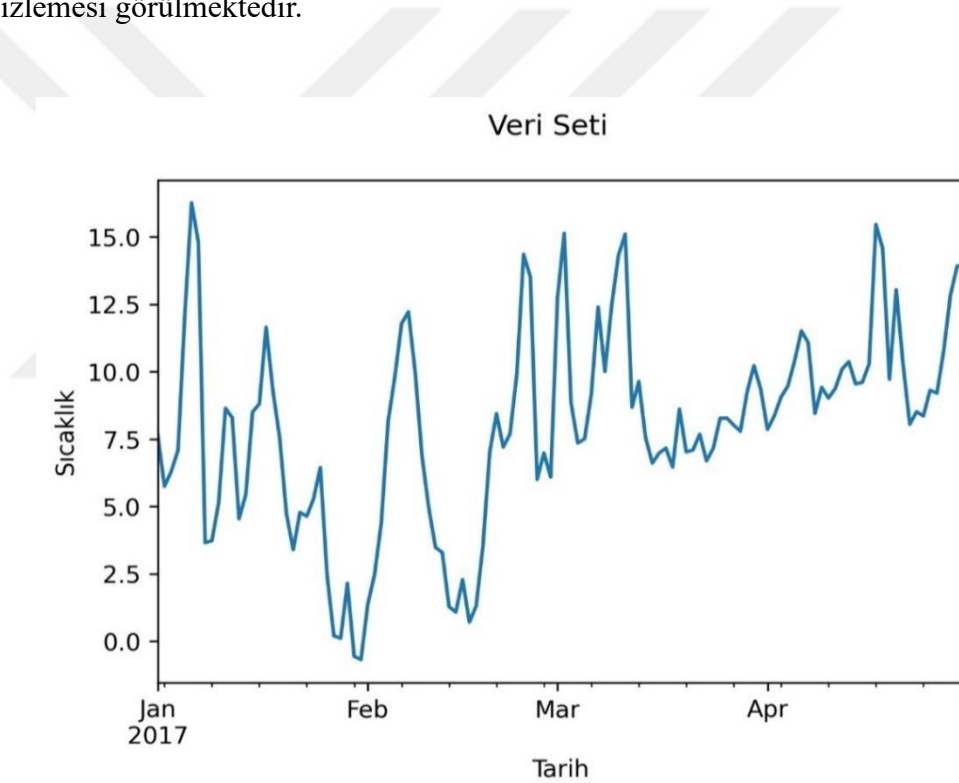
Kütüphane	Versiyon
Pandas	1.1.5
NumPy	1.19.4
Matplotlib	3.2.2
Sklearn	0.22.2
Keras	2.4.3
TensorFlow	2.4.0

3.4. Uygulanılan Yöntem

Uygulanılan yöntem, aşağıdaki adım sıralarıyla oluşturulmuş ve kullanılan program diline aktarılmıştır.

Adım 1. *Zaman serisi veri setinin yüklenmesi ve incelenmesi*

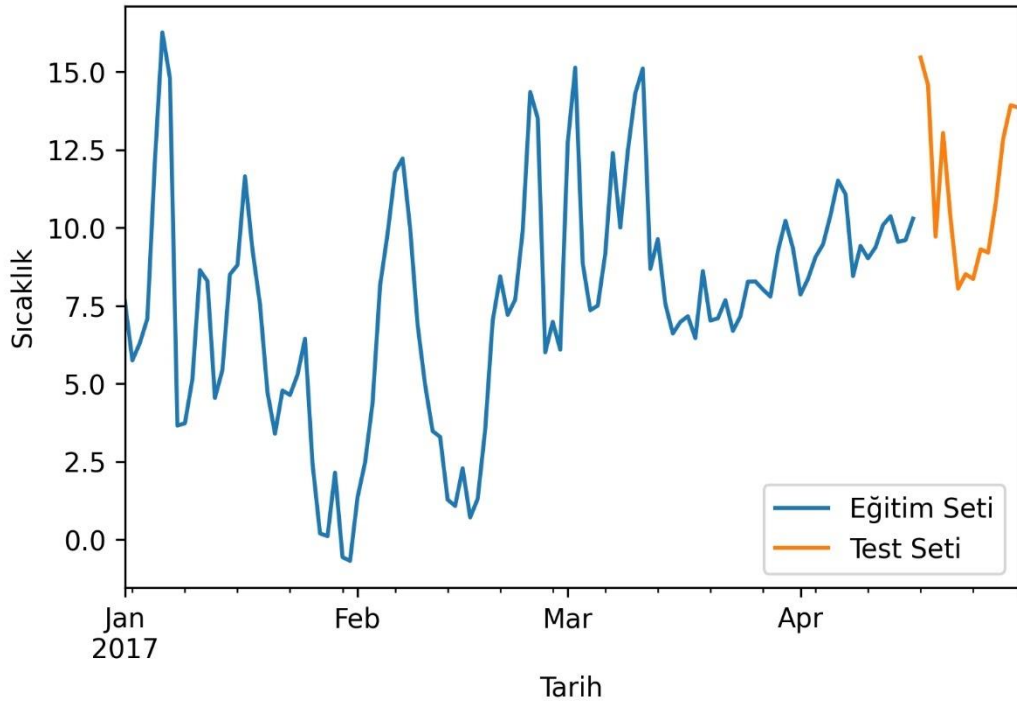
Materyaller bölümünde belirtilen Python kütüphaneleri yüklendikten sonra veri seti, *Pandas* kütüphanesinin farklı formattaki dosyaları *dataframe* adı verilen iki boyutlu heterojen tablo verisi olarak sisteme dahil edilmiştir. Veri setinde boş gözlem olup olmadığı kontrol edilmiş ve boş gözlem olmadığı görülmüştür. Şekil 3.1’de veri setinin ön izlemesi görülmektedir.



Şekil 3.1. Veri seti grafiği

Adım 2. *Veri setinin eğitim ve test seti olarak ayrılması*

Veri seti 01.01.2017 ile 30.04.2017 tarihleri arasında yer alan 120 güne ait sıcaklık değerinden oluşmaktadır. Veri, 106 gözlem (veri setinin yaklaşık %90'ı) eğitim seti ve 14 gözlem (veri setinin yaklaşık %10'u) test seti olacak şekilde iki kısma ayrılmıştır.



Şekil 3.2. Eğitim ve test seti

Adım 3. Verilerin ölçeklendirilmesi

Sklearn kütüphanesinin *MinMaxScaler* komutu ile her iki veri seti -1 ile 1 arasında ölçeklendirilmiştir. Veriler önce denklem 3.1 ile $[0,1]$ aralığına, sonrasında ise denklem 3.2 ile $[-1,1]$ aralığına dönüştürülür. Kullanılan programlama dilinde *axis=0* ifadesi satırları ifade etmektedir.

$$X_{std} = (X - X.min(axis = 0)) / (X.max(axis = 0) - X.min(axis = 0)) \quad (3.1)$$

$$X_{scaled} = X_{std} * (max - min) + min \quad (3.2)$$

NumPy kütüphanesi ile öncelikle veri setleri tek boyutlu dizine ve y , X 'in gecikmeli hali olacak şekilde eğitim seti y ve X şeklinde iki forma dönüştürülmüştür. Girdi değerleri, LSTM modeline aktarılmadan önce üç boyutlu vektör formuna çevrilmiştir.

Adım 4. Modelin oluşturulması

Yapay sinir ağları modellenirken model seçimi, katman belirleme, optimizasyon türü, hata metrikleri, performans ölçüm metrikleri gibi birçok değişken belirlenir. Bunlar

verinin ve problemin türüne göre belirlenirken bazı değişkenleri değiştirerek de farklı performans denemeleri yapılabilir.

Model olarak başlangıçta *Sequential* modeli seçilmiştir. *Sequential*, giriş ve çıkış tensörüne sahip düz katmanlar yığımından oluşur. Belirlenen modele Şekil 3.3'te belirtilen LSTM katmanı eklenmiştir. Katmanda LSTM nöron sayısı (n_{unit}) 1 ile 10 arasında denenmiştir. Bu çalışmada tek LSTM katmanlı bir model uygulandığından tekrar katman ekleme işlemi olan *return_sequences*, *False* olarak belirlenmiştir.

```
lstm_model.add(LSTM(unit=n_unit, input_shape=(1,
X_train_min_3d.shape[1]), activation='relu',
kernel_initializer='GlorotUniform', return_sequences=False))
```

Şekil 3.3. LSTM katmanı oluşturma kod parçası

Adım 5. Aktivasyon fonksiyonu seçimi

Aktivasyon fonksiyonu olarak ReLU fonksiyonu kullanılmıştır. ReLU aktivasyon fonksiyonu girdi pozitif ise doğrudan ileten, negatif olduğu durumlarda da sıfır çıktısı veren parçalı bir doğrusal fonksiyondur. Kaybolan gradyan probleminin üstesinden gelmesine, modellerin hızlı öğrenmesine ve iyi performans göstermesine olanak tanıdığı için sıklıkla kullanılır (Brownlee, 2020).

Adım 6. Başlangıç ağırlıklarının belirlenmesi

Başlangıç ağırlıklarını belirlemede *GlorotUniform* kullanılmıştır. Glorot veya Xavier olarak da geçen başlatıcı, $[-limit, limit]$ sınırları içinde düzgün dağılımdan rastgele sayı üretir. n_{girdi} , ağırlık tensöründeki girdilerin sayısı ve $n_{çıktı}$, çıktı birimlerinin sayısı olmak üzere *limit* değeri aşağıdaki denklemde gösterildiği şekilde hesaplanır.

$$limit = \sqrt{\frac{6}{n_{girdi} + n_{çıktı}}} \quad (3.3)$$

Adım 7. Çıktı katmanı oluşturma

LSTM katmanından sonra tek nöronlu *Dense* katmanı eklenmiştir. *Dense* katmanı, aşağıdaki denklem ile çıktıyı oluşturur. Denklemden *çarpım* işlevi, vektörlerin iç çarpımını gerçekleştirir, *kernel* katman tarafından oluşturulan ağırlık matrisi ve *aktivasyon*, belirlenen aktivasyon fonksiyonudur.

$$\text{çıkıtı} = \text{aktivasyon}(\text{çarpım}(\text{girdi}, \text{kernel}) + \text{ön bilgi}) \quad (3.4)$$

Model eğitilirken hata kareler ortalaması (HKO) kayıp metriği kullanılmıştır. HKO, tahmin edilen değişkenin gözlemlenen değeri ile tahmin edilen değeri arasındaki farkın kareleri toplamının ortalamasıdır.

$$HKO = \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{n} \quad (3.5)$$

Adım 8. Optimizasyon yöntemi ve erken durdurma kriteri belirleme

Optimizasyon yöntemi olarak Adam optimizasyon algoritması kullanılmıştır. Model eğitilirken gradyan güncellemesi için örnek sayısı (batch) 1 ve döngü sayısı (epoch) 100 olarak tanımlanmıştır. Erken durdurma kriteri Şekil 3.4'teki gibi eklenmiştir. *monitör*='loss' ile her döngüdeki kayıp değerleri model adımlarında gösterilir. *patience*, eğitimin durdurulmasına karar veren iyileştirme olmayan döngü sayısı ve *verbose*, erken durdurmanın hangi döngüde gerçekleştiğini, *restore_best_weights*, modelde en iyi ağırlıkların mı yoksa son hesaplanan ağırlıkların mı kullanılacağını ve *min_delta*, iyileştirme olarak nitelendirilecek olan minimum değeri belirtir.

```
early_stop = EarlyStopping(monitor='loss', patience=4, verbose=1,
restore_best_weights=False, min_delta=0.0001)
```

Şekil 3.4. Erken durdurma kod parçası

Adım 9. Verileri geri ölçeklendirme

Oluşturulan model ile test verilerinin tahminleri oluşturulmuştur. İlk başta *Sklearn* kütüphanesi ile ölçeklendirilen veriler tekrar *Sklearn* kütüphanesinin *inverse_transform* işlemi ile geri ölçeklendirmiştir.

Adım 10. *Hata metrikleri ile model performans ölçümü*

Model performans değerlendirmesi için hatanın mutlak ortalaması (HMO), HKO ve hata kareler ortalamasının karekökü (HKOK) hata metrikleri kullanılmıştır.

$$HMO = \frac{\sum_{i=1}^n |y_i - \hat{y}_i|}{n} \quad (3.6)$$

$$HKOK = \sqrt{\frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{n}} \quad (3.7)$$

BÖLÜM 4. ARAŞTIRMA BULGULARI

LSTM modeli oluşturulurken nöron sayısı ve gecikme sayısı 1 ile 10 arasında değiştirilerek denenmiş ve sonuçlar Tablo 4.1 ve Tablo 4.2’de verilmiştir. En iyi hata metrik değerleri nöron sayısı 5, gecikme sayısının 1 olduğu durumda elde edilmiştir.

Tablo 4.1. Nöron sayısı karşılaştırma tablosu

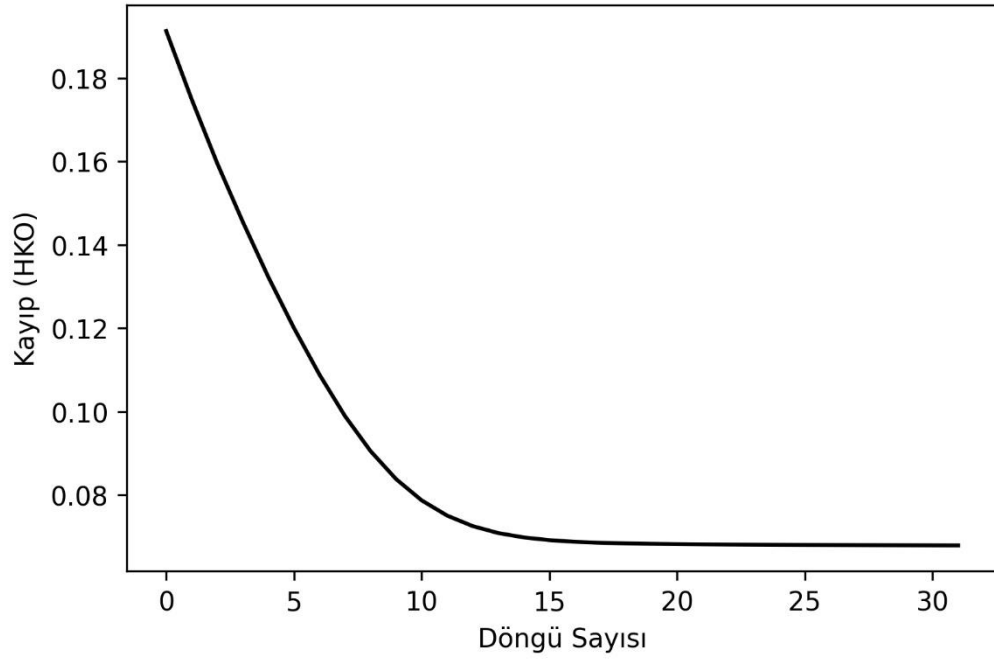
Hata metrikleri	n=1	n=2	n=3	n=4	n=5	n=6	n=7	n=8	n=9	n=10
HKOK	0,7710	0,8231	0,6786	0,8156	0,6638	0,7291	0,8274	0,7002	0,7435	0,6916
HMO	0,6714	0,5552	0,5041	0,5535	0,5026	0,5280	0,5605	0,5260	0,5340	0,5199

Tablo 4.2’de de görüleceği üzere gecikme sayısı arttıkça öngörü performansında düşüşler gözlemlenmiştir.

Tablo 4.2. Gecikme sayısı karşılaştırma tablosu

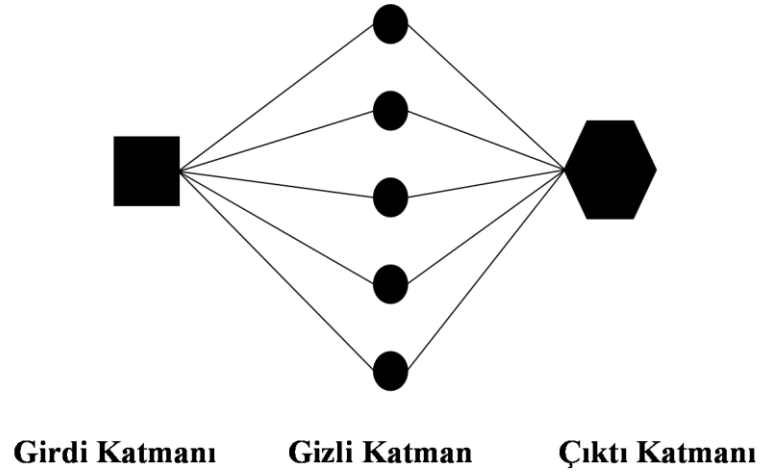
Hata metrikleri	g=1	g=2	g=3	g=4	g=5	g=6	g=7	g=8	g=9	g=10
HKOK	0,6637	1,5511	1,8536	1,8685	1,9328	2,3130	2,7889	2,8960	2,7287	2,5250
HMO	0,5026	1,1544	1,5653	1,6912	1,7678	2,0609	2,4416	2,5456	2,4851	2,3190

En iyi model, nöron sayısı 5 ve gecikme sayısı 1 ile çalıştırıldığında erken durdurma devreye girerek 32’nci adımda eğitimi durdurmuştur. Şekil 4.1’de gösterilen döngü sayısı ile HKO kayıp grafiğini incelediğimizde kayıp değerlerinin 15’ten sonra düşüş miktarının çok az olduğu görülmektedir.



Şekil 4.1. Döngü sayısı grafiği

Test edilen parametre değişiklikleri ile elde edilen en iyi değerler olan 5 nöron ve 1 gecikmeli değişken ile oluşturulan modelin katman yapısı Şekil 4.2’de görüldüğü gibidir.



Şekil 4.2. Uygulanan model katman yapısı

Uygulanan modelin LSTM katmanında denklem 4.2 ile elde edilen 140 tane ağırlık parametresi ve denklem 4.4 ile elde edilen Dense katmanında 6 tane ağırlık parametresi olmak üzere modelde toplam 146 tane ağırlık parametresi bulunmaktadır.

LSTM katmanındaki ağırlık parametreleri sayısı denklem 4.1’deki gibi hesaplanır. Denklemde n , katman sayısını ve i , girdi sayısını göstermektedir. En son toplamaya eklenen katman sayısı, her katmana etki eden ön bilgi değerleridir. Modelde dört farklı kapı olduğundan tüm toplam en son dört ile çarpılır.

$$4((n + i)n + n) \quad (4.1)$$

$$4((5 + 1)5 + 5) = 140 \quad (4.2)$$

Dense katmanındaki ağırlık parametreleri sayısı denklem 4.3’teki gibi hesaplanır. Denklemde n , katman sayısını ve o , çıktı sayısını göstermektedir. En son toplamaya eklenen çıktı sayısı, katmana etki eden ön bilgi değeridir.

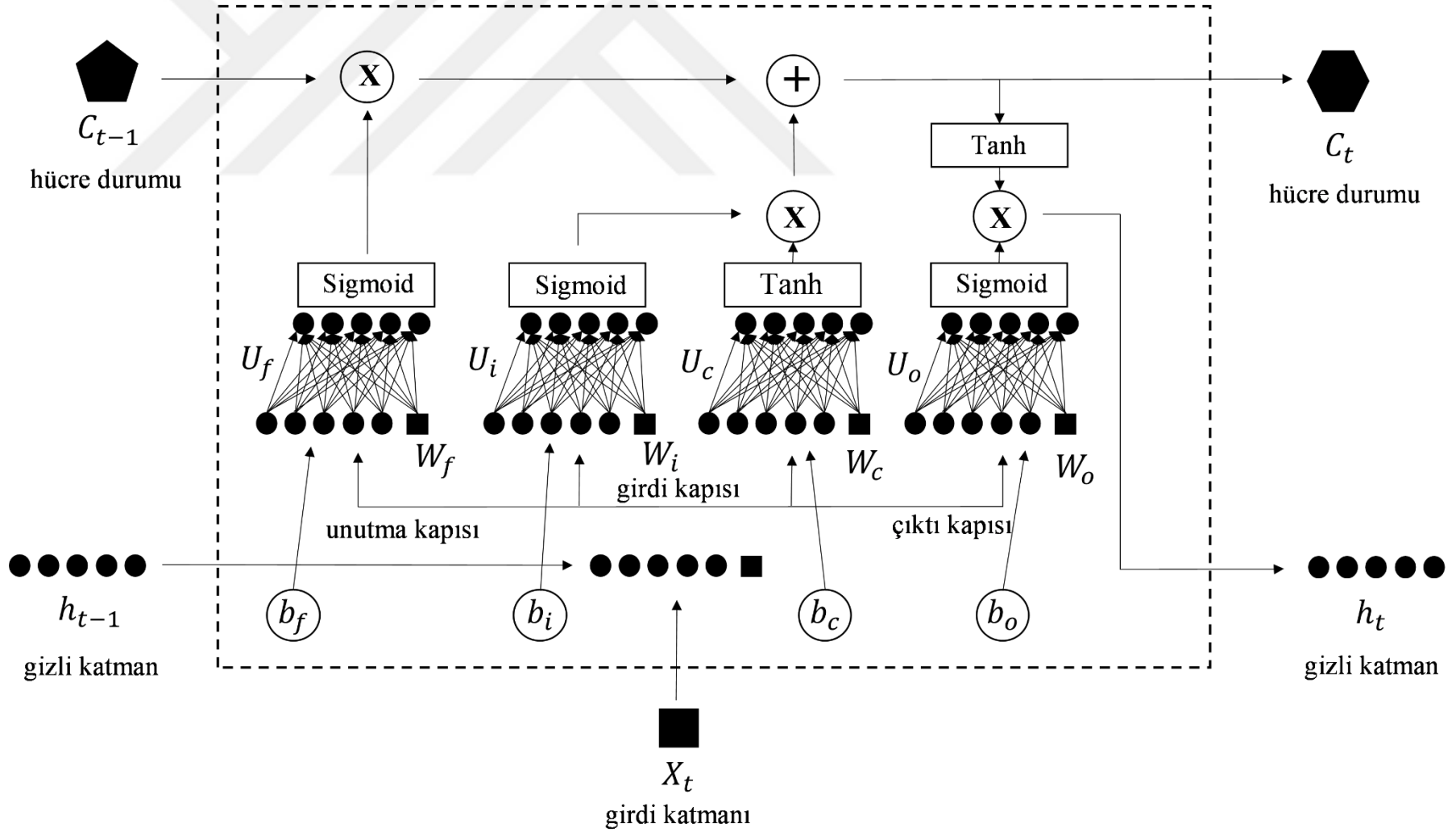
$$(n * o) + o \quad (4.3)$$

$$(5 * 1) + 1 = 6 \quad (4.4)$$

En iyi hata metrik değerlerinin elde edildiği durum için ağırlıkların modelin yapısındaki konumları Şekil 4.3’te, Dense katmanı ağırlık değerleri Tablo 4.3’te ve LSTM katmanı ağırlık değerleri Tablo 4.4’te gösterilmektedir.

Tablo 4.3. Dense katmanı ağırlıklarının değerleri

Ağırlıklar	n=1
d_1	0,8016
d_2	-1,1031
d_3	-0,9385
d_4	0,6926
d_5	0,2865
b_{dense}	0,0318



Şekil 4.3. Modelin ağırlıklarının gösterimi

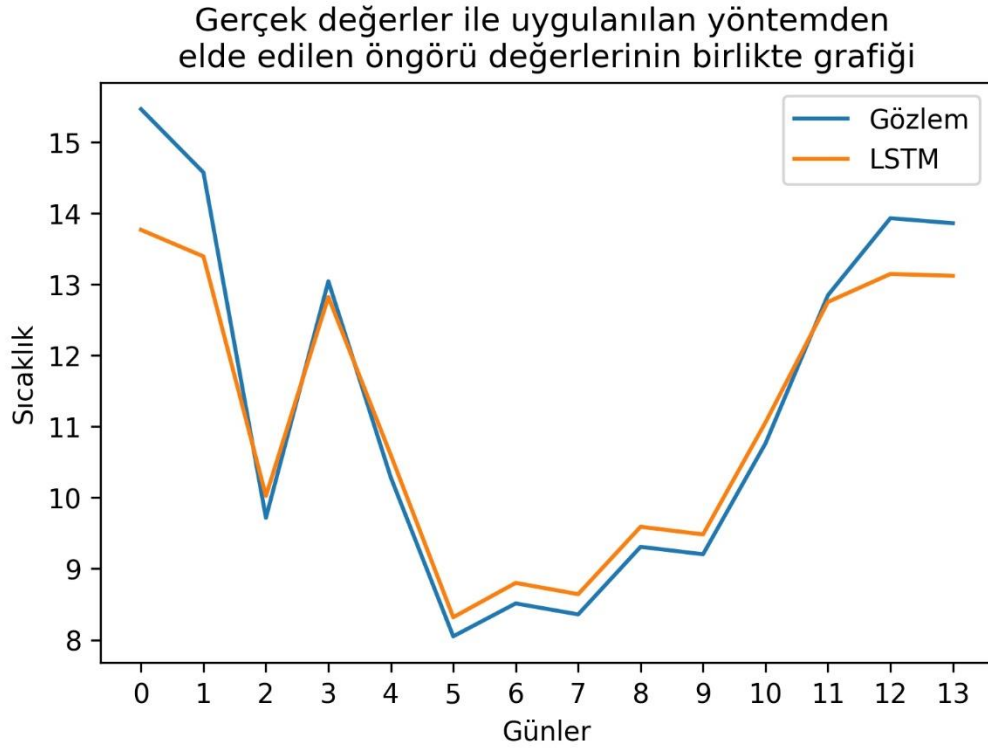
Tablo 4.4. LSTM modelinin katman ağırlıklarının değerleri

Ağırlıklar	n=1	n=2	n=3	n=4	n=5
W_i	0,7612	-0,9623	0,4025	-0,0237	-0,3828
W_f	-0,4802	0,1417	0,1248	-0,0052	-0,5047
W_c	0,6869	-1,0285	0,1286	-0,1966	-0,2797
W_o	0,5364	-0,9116	0,0304	-0,1892	0,2246
U_{i1}	0,0038	-0,0395	-0,0166	-0,1708	-0,2673
U_{i2}	0,5272	-0,0802	0,0404	-0,1613	-0,0392
U_{i3}	-0,2500	0,0729	0,0729	-0,2539	0,2135
U_{i4}	-0,4768	-0,0990	0,0833	-0,1545	-0,2536
U_{i5}	0,0926	-0,0553	0,5341	0,0326	0,2799
U_{f1}	-0,3360	0,0810	-0,2262	-0,2284	-0,2807
U_{f2}	-0,0022	-0,3586	0,0133	-0,0934	0,3815
U_{f3}	-0,4386	-0,0802	-0,1079	0,4473	-0,1461
U_{f4}	0,1685	0,0544	0,1505	0,0406	0,2228
U_{f5}	-0,2744	0,2835	-0,0115	-0,2299	-0,0343
U_{c1}	-0,0698	-0,2842	0,5538	0,0706	-0,2615
U_{c2}	-0,0251	0,1566	0,3976	0,0752	0,3649
U_{c3}	0,2590	0,2778	0,0373	0,3956	0,0417
U_{c4}	-0,1663	-0,1120	0,0419	0,3480	0,4192
U_{c5}	-0,0736	-0,2052	-0,2172	-0,0183	0,4211
U_{o1}	-0,0649	0,1257	-0,1063	-0,2944	0,1107
U_{o2}	-0,2279	-0,0135	0,1690	0,0230	-0,0103
U_{o3}	-0,1443	0,0163	0,2418	0,0319	-0,0216
U_{o4}	-0,1360	0,1855	-0,4051	0,0091	0,0815
U_{o5}	0,0795	-0,0703	0,1421	-0,3178	0,0953

Tablo 4.4. (Devamı)

Ağırlıklar	n=1	n=2	n=3	n=4	n=5
b_i	0,5082	0,5217	0,0555	-0,1145	-0,1414
b_f	1,0000	1,0000	1,0000	1,0000	1,0000
b_c	0,4652	0,2971	-0,0613	-0,1980	-0,2279
b_o	0,4870	0,5235	0,0568	-0,1151	-0,1384

Uygulanılan model adımları çalıştırılıp test verisi için modelin öngörü değerleri elde edilmiştir. Gözlemlenen gerçek değerler ile uygulanılan yöntemden elde edilen öngörü değerlerinin karşılaştırılması Şekil 4.4'te gösterilmiştir.



Şekil 4.4. Uygulanılan yöntemden elde edilen öngörü değerleri

Uygulanılan yöntem ve karşılaştırmada kullanılan diğer yöntemlere ait öngörü ve hata kriteri değerleri Tablo 4.5'te verilmiştir. Kocak ve ark. (2020) çalışmasında karşılaştırmada kullanılan öngörü yöntemlerinin parametre değerleri; model derecesi 1 ile 5 arasında, küme sayısı 3 ile 10 arasında, gizli katman nöron sayısı 1 ile 5 arasında

deneme yanılma yöntemi kullanılarak test kümesine ait en iyi HKOK değerine göre belirlenmiştir.

Tablo 4.5'te tüm yöntemlerden elde edilen öngörü ve hata kriteri değerlerine bakıldığında uygulanan yöntemin hem HKOK hem de HMO değerinde literatürdeki diğer yöntemlere göre daha iyi sonuçlar elde ettiği gözlemlenmiştir.

Tablo 4.5. Karşılaştırma tablosu

Test Verisi	MLP- ANN	PS- ANN	C- FTS	S&C FTS	FTS- N	FTS- SMN	FTS- PS	IFTS- PS- ABC	DIFTS- LSTM	RNN	Uygulanan Yöntem
15,46	10,16	10,65	10,47	9,96	11,40	10,97	9,91	11,27	10,71	14,18	13,76
14,57	11,50	13,58	14,93	14,85	14,76	13,09	14,15	12,72	10,89	13,87	13,39
9,72	11,24	9,80	6,01	8,50	12,48	8,85	9,91	6,34	11,78	9,99	10,02
13,04	9,06	7,88	10,13	9,96	8,76	8,85	9,91	9,37	10,35	12,97	12,82
10,28	11,14	12,65	10,47	14,85	13,13	13,09	9,91	13,16	10,41	10,47	10,60
8,05	9,78	9,14	10,47	9,96	9,60	8,85	9,91	9,90	8,82	8,95	8,32
8,51	8,17	7,96	7,35	8,17	8,63	8,85	9,91	8,03	8,45	9,18	8,80
8,36	9,09	9,64	8,09	8,17	8,79	8,85	9,91	7,92	8,34	9,10	8,64
9,31	8,88	8,83	8,09	8,17	8,65	8,85	9,91	8,04	8,34	9,68	9,59
9,20	9,67	9,95	10,13	9,26	9,38	8,85	9,91	8,35	10,80	9,60	9,48
10,76	9,46	9,35	10,13	9,26	8,75	8,85	9,91	10,02	11,33	10,91	11,06
12,85	10,43	10,98	10,47	10,87	10,41	10,97	9,91	11,68	11,81	12,82	12,75
13,93	11,07	12,06	10,47	14,85	11,90	10,97	14,15	12,04	13,57	13,55	13,14
13,86	11,21	11,08	14,04	9,62	12,63	10,97	9,91	13,46	13,39	13,51	13,12
HKOK	2,4394	2,3438	2,3148	2,5682	2,2254	2,3017	2,3091	2,1728	1,9500	0,8520	0,6637
HMO	1,9764	1,8187	1,7717	1,9242	1,7703	1,8511	1,6961	1,7914	1,3700	0,7490	0,5026

BÖLÜM 5. TARTIŞMA VE SONUÇ

Son yıllarda zaman serisi öngörüsünde derin öğrenme yöntemlerinin klasik zaman serisi yöntemleri, bulanık zaman serisi yöntemleri ve diğer yapay sinir ağları yöntemleriyle rekabet edebilir sonuçlar verdiği görülmektedir. Derin öğrenmenin gelişimi ile yapay sinir ağlarında oluşan yenilikçi ağ modellerinden biri ve aynı zamanda bir derin öğrenme yöntemi olan uzun kısa-süreli hafıza (LSTM) literatürde zaman serisi analizinde kullanılmaya başlanmıştır. Yapay sinir ağlarında olduğu gibi derin öğrenme yöntemlerinde de oluşturulan modele ait başlangıç değerleri ve modelin içerisindeki ağırlık değerleri modelin zaman serisi öngörü performansına etki etmektedir. Bu değerlerin belirlenmesi önem arz etmekte olup literatürde çözülmesi gereken problemler olarak karşımıza çıkmaktadır.

Bu tez kapsamında; modelin başlangıç değerlerinin üretilmesinde Xavier (Glorot) yöntemini, katman ağırlıklarının belirlenmesinde ise Adam optimizasyon yöntemini kullanan tek katmanlı LSTM yaklaşımı uygulanmış ve uygulanan yaklaşımın öngörü performansını göstermek için Giresun ili günlük ortalama minimum sıcaklık gerçek hayat zaman serisi kullanılmıştır. Uygulanan yaklaşım, literatürde var olan diğer bazı öngörü yöntemleri ile öngörü performansı açısından karşılaştırılmış ve yaklaşımın diğer yöntemlere göre hem HKOK hem de HMO değeri bakımından daha üstün öngörü performansına sahip olduğu gözlemlenmiştir.

Gelecek çalışmalarda farklı batch, epoch değerlerinin de denenerek öngörü performansına etkisi araştırılabilir ve aynı zamanda batch, epoch ve gecikmeli değişken sayısının keyfi belirlenmesinin önüne geçebilmek için bu değerlerin optimizasyon ile belirlenmesi sağlanabilir.

KAYNAKLAR

- Abdel-Nasser, M. & Mahmoud, K., 2019. Accurate photovoltaic power forecasting models using deep LSTM-RNN. *The Natural Computing Applications* , 31:2727-2740.
- Aladag, C. H., 2013. Using multiplicative neuron model to establish fuzzy logic relationships. *Expert Systems Applications*, 40(3): 850-853.
- Alla, L., Bamba, D.C., Mamadou, N.E., Ba, B.S.O. & Moussa L., 2020. [arxiv.org. https://arxiv.org/abs/2004.13840](https://arxiv.org/abs/2004.13840) , Eriřim Tarihi: 14.01.2021.
- Anwer, A. M. O., 2017. Derin Öğrenme Yöntemleri ile Göğüs Kanseri Teřhisi. Türk Hava Kurumu Üniversitesi, Fen Bilimleri Enstitüsü, Elektrik ve Bilgisayar Mühendislięi ABD, Yüksek Lisans Tezi.
- Bao, W. & Yue, J., 2017. A deep learning framework for financial time series using stacked autoencoders and long-short term memory. *Plos One*, 12(7): e0180944.
- Bas, E., Egrioglu, E., Aladaę, C. H. & Yolcu, U., 2015. Fuzzy-time-series network used to forecast linear and nonlinear time series. *Applied Intelligence*, 43(2): 343-355.
- Bas, E., Grosan, C., Egrioglu, E. & Yolcu, U., 2018. High order fuzzy time series method based on Pi-Sigma neural network. *Engineering Applications Artificial Intelligence*, 72: 350-356.
- Bengio, Y., Simard, P. & Frasconi, P., 1994. Learning long-term dependencies with gradieny descent is difficult. *IEEE Transactions on Neural Networks*, 5(2): 157-166.
- Brownlee, J., 2020. Machine Learning Mastery. <https://machinelearningmastery.com/rectified-linear-activation-function-for-deep-learning-neural-networks/#:~:text=The%20rectified%20linear%20activation%20function,otherwise%2C%20it%20will%20output%20zero.&text=The%20rectified%20linear%20activation%20func> , Eriřim Tarihi: 06.01.2021.
- Cai, M. & Liu, J., 2016. Maxout neurons for deep convolutional and LSTM neural networks in speech recognition. *Speech Communication*, 77: 53-64.
- Chang, Z., Zhang, Y. & Chen, W., 2018. Effectice Adam-optimized LSTM Neural Network for Electricity Price Forecasting. *IEEE 9th International Conference on Software Engineering and Service Science*.

- Chang, Z., Zhang, Y. & Chen, W., 2019. Electricity price prediction based on hybrid model of adam optimized LSTM neural network and wavelet transform. *Energy*, 187: 115804.
- Chen, S. M., 1996. Forecasting enrollments based on fuzzy time-series. *Fuzzy Sets Systems*, 81(3): 311-319.
- Chollet, F., 2018. *Deep Learning with Python*. NY: Manning Publications Co., Shelter Island, 61-62.
- Çuhadar, M., Güngör, İ. & Göksu, A., 2009. Turizm Talebinin Yapay Sinir Ağları ile Tahmini ve Zaman Serisi Yöntemleri ile Karşılaştırmalı Analizi: Antalya İline Yönelik Bir Uygulama. *Süleyman Demirel Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi*, 14(1): 99-114.
- Egrioglu, E., Aladag, C. H. & Yolcu, U., 2013. Fuzzy time series forecasting with a novel hybrid approach combining fuzzy c-means and neural networks. *Expert Systems with Applications*, 40(3): 854-857.
- Egrioglu, E., Yolcu, U. & Bas, E., 2019. Intuitionistic high-order fuzzy time series forecasting method based on Pi-Sigma artificial neural networks trained by artificial bee colony. *Granul Comput*, 4(4): 639-654.
- Elit, C. A., 2018. Makine Öğrenmesi Yöntemleriyle Akan Görüntülerden Otomatik Aktivite Sınıflandırma. Ankara Üniversitesi, Fen Bilimleri Enstitüsü, Bilgisayar Mühendisliği ABD, Yüksek Lisans Tezi.
- Elman, J. L., 1990. Finding structure in time. *Cognitive Science*, 14(2): 179-211.
- Elmas, Ç., 2018. *Yapay Zeka Uygulamaları*. 4. Cilt. Seçkin Yayıncılık, Ankara, 231.
- Gao, T. & Chai, Y., 2018. Improving Stock Closing Price Prediction Using Recurrent Neural Network and Technical Indicators. *Neural Computation*, 30(10): 2833-2854.
- Gao, T., Chai, Y. & Liu, Y., 2017. Applying long short term memory neural networks for predicting stock closing price. *8th IEEE International Conference on Software Engineering and Service Science*, 575-578.
- Glorot, X. & Bengio, Y., 2010. Understanding the difficulty of training deep feedforward neural networks. *Thirteenth International Conference on AISTATS*, 9: 249-256.
- Goodfellow, I., Bengio, Y. & Courville, A., 2018. *Derin Öğrenme*. Ankara: Buzdağı Yayınevi.
- Halpern-Wight, N., Konstantinou, M., Charalambides, A. G. & Reinders, A., 2020. Training and Testing of a Single-Layer LSTM Network for Near-Future Solar Forecasting. *Applied Sciences*, 10(17).
- Hermans, M. & Schrauwen, B., 2013. Training and Analysing Deep Recurrent Neural Networks. *Advances in Neural Information Processing Systems*, 26: 190-198.
- Hochreiter, S., 1991. *Untersuchungen zu dynamischen neuronalen Netzen*. München Technische Universität, Institut für Informatik, Diploma Thesis.

- Hochreiter, S. & Schmidhuber, J., 1997. Long Short-Term Memory. *Neural Computation*, 9(8): 1735-1780.
- Karakoyun, E. Ş., 2018. Derin Öğrenme ile Zaman Serilerinin Gerçek Zamanlı Tahmini. Necmettin Erbakan Üniversitesi, Fen Bilimleri Enstitüsü, Endüstri Mühendisliği ABD, Yüksek Lisans Tezi.
- Karevan, Z. & Suykens, J. A., 2020. Transductive LSTM for time-series prediction: An application to weather forecasting. *Neural Networks*, 125: 1-9.
- Kawaguchi, K., 2000. UTEP. <http://wwwold.ece.utep.edu/research/webfuzzy/docs/kk-thesis/kk-thesis-html/node10.html> , Erişim Tarihi: 04.01.2021.
- Kaynar, O., Taştan, S. & Demirkoparan, F., 2011. Yapay Sinir Ağları ile Doğalgaz Tüketim Tahmini. Atatürk Üniversitesi İİBF Dergisi, Issue 10. *Ekonometri ve İstatistik Sempozyumu Özel Sayısı*, 463-474.
- Keskin Benli, Y., 2005. Bankalarda Mali Başarısızlığın Öngörülmesi Lojistik Regresyon ve Yapay Sinir Ağır Karşılaştırılması. *Gazi Üniversitesi Endüstriyel Sanatlar Eğitim Fakültesi Dergisi*, (16): 31-46.
- Kingma, D. P. & Ba, J., 2017. arXiv. <https://arxiv.org/abs/1412.6980> Erişim Tarihi:02.01.2021.
- Kocak, C., Egrioglu, E. & Bas, E., 2020. A new deep intuitionistic fuzzy time series forecasting method based on long short-term memory. *The Journal of Supercomputing*, DOI: 10.1007/s11227-020-03503-8.
- Liu, W., Liu, W. D. & Gu, J., 2020. Forecasting oil production using ensemble empirical model decomposition based Long Short-Term Memory neural network. *Journal of Petroleum Science and Engineering*, 189(2): 107013.
- Livieris, I. E., Pintelas, E. & Pintelas, P., 2020. A CNN-LSTM model for gold price time-series forecasting. *Neural Computing and Applications*, 32: 17351-17360.
- McCulloch, W. S. & Pitts, W., 1943. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5: 115-133.
- Olah, C., 2015. <http://colah.github.io/posts/2015-08-Understanding-LSTMs/> Erişim Tarihi: 15.10.2019.
- Pandit, R. K., Kolios, A. & Infield, D., 2020. Data-driven weather forecasting models performance comparison for improving offshore wind turbine availability and maintenance. *IET Renewable Power Generation*, 14(13): 2386-2394.
- Rumelhart, D. E., Hinton, G. E. & Williams, R. J., 1986. Parallel distributed processing: explorations in the microstructure of cognition. MIT Press, Cambridge.
- Sezer, O. B., Gudelek, M. U. & Ozbayoglu, A. M., 2020. Financial time series forecasting with deep learning : A systematic literature review: 2005-2019. *Applied Soft Computing*, 90: 106181.
- Shin, Y. & Ghosh, J., 1991. The pi-sigma network: an efficient higher-order neural network for pattern classification and function approximation. *IJCNN-91-Seattle International Joint Conference on Neural Networks*, 1: 13-18.

- Song, Q. & Chissom, B. S., 1993. Forecasting enrollments with fuzzy time series- Part I. *Fuzzy Sets and Systems*, 1(54): 1-9.
- Soutner, D. & Müller, L., 2013. Application of LSTM Neural Networks in Language Modelling. *International Conference on Text, Speech and Dialogue*, 105-112.
- Wang, J., Yu, L.-C., Lai, R. & Zhang, X., 2016. Dimensional Sentiment Analysis Using a Regional CNN-LSTM Model. *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, 225-230.
- Xu, C. ve diğerleri, 2019. Improving prediction performance for indoor temperature in public buildings based on a novel deep learning method. *Building and Environment*, 148: 128-135.
- Zayegh, A. & Bassam, N. A., 2018. *Digital Systems. İçinde: Neural Network Principles and Applications*. IntechOpen, 115-130.
- Zaytar, M. A. & Amrani, C. E., 2016. Sequence to Sequence Weather Forecasting with Long Short-Term Memory Recurrent Neural Networks. *International Journal of Computer Applications*, 143(11): 7-11.
- Zhang, A., Lipton, Z. C., Li, M. & Smola, A. J., 2020. *Dive into Deep Learning - Release 0.7.1*. d2l.ai, 171.
- Zhang, G. P., 2003. Time series forecasting using a hybrid ARIMA and neural network model. *Neurocomputing* , 50: 159-175.

ÖZGEÇMİŞ

Onur DERYA İlk, orta ve lise eğitimini Gelibolu’da tamamladı. 2006 yılında Gelibolu Anadolu Lisesi’nden mezun oldu. 2007 yılında başladığı Eskişehir Osmangazi Üniversitesi İstatistik Bölümü’nü 2012 yılında bitirdi. 2018 yılında Giresun Üniversitesi İstatistik Anabilim Dalında yüksek lisans eğitimine başladı. 2016 yılında Ordu Üniversitesi’nde bilgisayar işletmeni olarak çalışmaya başladı. Halen bilgisayar işletmeni olarak görev yapmaktadır.