



MAKİNE ÖĞRENMESİ İLE CERRAHİ HAREKET TANIMA

Simge Nur KABATAŞ

YÜKSEK LİSANS TEZİ

BİLGİSAYAR MÜHENDİSLİĞİ ANA BİLİM DALI

**GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

AĞUSTOS 2021

ETİK BEYAN

Gazi Üniversitesi Fen Bilimleri Enstitüsü Tez Yazım Kurallarına uygun olarak hazırladığım bu tez çalışmada;

- Tez içinde sunduğum verileri, bilgileri ve dokümanları akademik ve etik kurallar çerçevesinde elde ettiğimi,
- Tüm bilgi, belge, değerlendirme ve sonuçları bilimsel etik ve ahlak kurallarına uygun olarak sunduğumu,
- Tez çalışmada yararlandığım eserlerin tümüne uygun atıfta bulunarak kaynak gösterdiğimi,
- Kullanılan verilerde herhangi bir değişiklik yapmadığımı,
- Bu tezde sunduğum çalışmanın özgün olduğunu,

bildirir, aksi bir durumda aleyhime doğabilecek tüm hak kayıplarını kabullendiğimi beyan ederim.

Simge Nur KABATAŞ

13/08/2021

MAKİNE ÖĞRENMESİ İLE CERRAHİ HAREKET TANIMA

(Yüksek Lisans Tezi)

Simge Nur KABATAŞ

GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

Ağustos 2021

ÖZET

Robotik cerrahi hareketlerin otomatik olarak sınıflandırılması ve tanınması, hem cerrahi eğitimde geri bildirim sağlama hem de ameliyatlarda istenmeyen durum ve hataları önlemede önemli bir adımdır. Cerrahi robotlardan kaydedilen kinematik veriler, cerrahi hareketler hakkında bilgi içerir. Bu verileri kullanarak cerrahların hareketlerini otomatik olarak modelleyebilir ve tanıyabiliriz. Bu çalışmada, cerrahi hareketlerin tanınması için kinematik veriler ile son zamanlarda zaman serisi verileri üzerindeki çalışmalarda tekrarlayan sinir ağlarına göre daha iyi başarı gösteren Transformer modeli kullanılmıştır. Bu çalışmada öğrenilen model literatürde sıkça kullanılan Long Short-Term Memory (LSTM) modeli ile karşılaştırılmıştır. Transformer modelinin ortalama olarak doğruluk payı %77' dir. Sonuçlara göre Transformer modelinin doğruluk payının literatürdeki LSTM modelleri sonuçları ile karşılaştırılabilir olduğu, kendi geliştirdiğimiz LSTM yaklaşımlara göre daha iyi sonuç verdiği görülmüştür. Ayrıca standart sapması daha azdır. Bildiğimiz kadarıyla çalışmamız bir Transformer modelini kinematik veri kullanarak cerrahi hareket tanıma için kullanan ilk çalışmadır. Deney sonuçlarımız Transformer modelinin bu alanda kullanımını desteklemektedir.

Bilim Kodu : 92432

Anahtar Kelimeler : Cerrahi hareket tanıma, kinematik veri, derin öğrenme, transformer ağı

Sayfa Adedi : 71

Danışman : Dr. Öğr. Üyesi Duygu SARIKAYA

SURGICAL GESTURE RECOGNITION WITH MACHINE LEARNING
(M. Sc. Thesis)

Simge Nur KABATAS

GAZİ UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

August 2021

ABSTRACT

Automatic classification and recognition of surgical activities is an important step towards providing automatic feedback in surgical training, and preventing adversarial events and medical errors in surgeries. Kinematic data recorded from surgical robots contains information about surgical gestures. Using this data, we can model and recognize surgeons' gestures automatically. In this study, the Transformer model, which has shown better performance than Recurrent Neural Networks (RNNs) with time series data, has been used to recognize surgical gestures with kinematic data. The model learned in this study is compared with the Long Short-Term Memory (LSTM) model, which is widely used in the literature. The average accuracy of the Transformer model is 77%. According to the results, Transformer model is comparable to the state of the art LSTM methods, and has outperformed the LSTM model we have developed in this study as part of the benchmark, and the standard variation is lower. To our knowledge, our study is the first to use Transformer model for surgical activity recognition with kinematic data. Our experiments show the promise of Transformer Network in this domain.

Science Code : 92432

Key Words : Surgical gesture classification, kinematic data, deep learning, transformer network

Page Number : 71

Supervisor : Assist. Prof. Dr. Duygu SARIKAYA

TEŞEKKÜR

Bu günlere gelmemde emeği olan beni yetiştiren, her zaman yanımda olan sevgili ailem başta olmak üzere ve tez çalışmamın gerçekleşmesinde, değerli bilgisini, önerilerini ve zamanını benimle paylaşan, çok değerli tecrübeleri ile çalışmamın gidişatına yön veren, saygıdeğer danışman hocam Dr. Öğr. Üyesi Duygu SARIKAYA (Gazi Üniversitesi)' ya sonsuz teşekkürlerimi sunarım. Ayrıca hiçbir zaman kendime olan inancımı kaybetmememi sağlayan, motivasyon kaynağım biricik hayat arkadaşım Semih TERCİ (ICterra)' ye teşekkür ederim.



İÇİNDEKİLER

	Sayfa
ÖZET.....	iv
ABSTRACT.....	v
TEŞEKKÜR.....	vi
İÇİNDEKİLER	vii
ÇİZELGELERİN LİSTESİ.....	ix
ŞEKİLLERİN LİSTESİ	x
RESİMLERİN LİSTESİ	xi
SİMGELER VE KISALTMALAR.....	xii
1. GİRİŞ.....	1
2. LİTERATÜR TARAMASI	5
3. MATERYALLER.....	13
3.1. JIGSAWS Veri Seti.....	13
3.2. Kinematik Veri.....	16
3.3. Kullanılan Araçlar	18
3.3.1. Python programlama dili	18
3.3.2. Google colaboratory	20
4. YÖNTEM.....	21
4.1. Derin Öğrenme.....	21
4.2. Yapay Sinir Ağları	22
4.2.1. Derin sinir ağları.....	23
4.2.2. Tekrarlayan sinir ağları (Recurrent Neural Network - RNN).....	25
4.2.3. Gated recurrent units	28
4.2.4. Long short- term memory	28
4.2.5. Transformer network	30

	Sayfa
4.3. Hiperparametreler	33
4.3.1. Öğrenme oranı	34
4.3.2. Batch boyutu.....	34
4.3.3. Epoch sayısı.....	34
4.3.4. Aktivasyon fonksiyonu.....	34
4.3.5. Kayıp fonksiyonu	37
4.3.6. Dropout.....	38
4.3.7. Optimizasyon algoritması.....	39
4.4. Veri İşleme Yöntemleri.....	40
4.4.1. Kayan pencere tekniği	40
4.4.2. Leave one user out (LOUO)	41
4.5. Performans Değerlendirme Yöntemleri	41
4.5.1. Karışıklık matrisi	41
4.5.2. Doğruluk (Accuracy).....	42
4.5.3. Kesinlik (Precision)	42
4.5.4. Geri çağırma (Recall)	43
4.5.5. F1 skoru	43
4.5.6. Levensthein mesafesi.....	43
5. DENEY VE SONUÇLARI.....	45
5.1. Veri Seti Hazırlama.....	45
5.2. Modellerin Eğitim Süreci.....	52
6. SONUÇ	57
6.1. Bu Tezden Çıkarılan Yayınlar	58
KAYNAKLAR	61
ÖZGEÇMİŞ	71

ÇİZELGELERİN LİSTESİ

Çizelge	Sayfa
Çizelge 3.1. Kinematik verinin değişkenleri.....	17
Çizelge 4.1. Karışıklık matrisi [108].....	42
Çizelge 5.1. Hareket etiketlerine karşılık gelen sayısal etiketler	50
Çizelge 5.2. DiPietro ve diğerlerinin [18], cerrahi hareket tanıma çalışmasındaki sonucu.....	53
Çizelge 5.3. LSTM modelinin her test kullanıcısına uygulama sonuçları.....	54
Çizelge 5.4. Dikiş atma cerrahi görevini tanımak için geliştirilen Transformer modelinin sonucu.....	54
Çizelge 5.5. Dikiş atma cerrahi görevini tanımak için uygulanan modellerin sonuçları .	55

ŞEKİLLERİN LİSTESİ

Şekil	Sayfa
Şekil 3.1. Dikiş atma (sol), düğüm atma (orta), iğne geçirme (sağ) görevlerine ait hareket sırası.....	16
Şekil 4.1. Yapay sinir ağı modelinin temeli.....	22
Şekil 4.2. Girdi, gizli ve çıktı katmanından oluşan derin sinir ağı mimarisi	23
Şekil 4.3. Basit bir tekrarlayan sinir ağı.....	26
Şekil 4.4. Tekrarlayan sinir ağının katlanmış ve katlanmamış durumu.....	26
Şekil 4.5. LSTM mimarisinde hücre yapısı	29
Şekil 4.6. Transformer modeli mimarisi	31
Şekil 4.7. Sigmoid fonksiyonu grafiği	35
Şekil 4.8. Tanh fonksiyonu grafiği	36
Şekil 4.9. ReLU fonksiyonu grafiği.....	36
Şekil 4.10. Softmax fonksiyonu grafiği	37
Şekil 4.11. Dropout modeli gösterimi.....	39
Şekil 4.12. Kayan pencere gösterimi	41
Şekil 5.1. Zaman serisi verilerinde kayan pencere tekniği	50

RESİMLERİN LİSTESİ

Resim	Sayfa
Resim 3.1. Da Vinci cerrahi sistemi	13
Resim 3.2. JIGSAWS veri setindeki 3 cerrahi görev (soldan sağa): dikiş atma, düğüm atma, iğne geçirme	14
Resim 5.1. Dikiş atma görevi B cerrahı tarafından denemesi yapıldığı videodan bir çerçeve	46
Resim 5.2. İndirilen veri setinin içindeki klasör ve dosyalar	46
Resim 5.3. Dikiş atma görevine ait olan kinematik veri içeren dosyalar	47
Resim 5.4. B001 denemesine ait başlangıç – bitiş çerçeve numaraları ve etiket bilgisi.	47
Resim 5.5. Veri setinin 0-7 sütun aralığında ilk 10 satır	48
Resim 5.6. Veri setinin 8-15 sütun aralığında ilk 10 satır	48
Resim 5.7. B001 denemesine ait veri setinde null değer olup olmadığı.....	49
Resim 5.8. B001 denemesine ait veri setinde her bir etiketin veri setindeki dağılımı....	49
Resim 5.9. B001 denemesine ait veri setindeki her bir özelliğin özet bilgileri	49
Resim 5.10. Veri setine 16’ lık kayan pencere şeklinde okunmasında oluşan veri	51
Resim 5.11. Her bir çerçeveye karşılık gelen gerçek referans etiketi ile model sonuçlarında tahmin edilen etiketlerin gösterimi.....	56

SİMGELER VE KISALTMALAR

Bu çalışmada kullanılmış simgeler ve kısaltmalar, açıklamaları ile birlikte aşağıda sunulmuştur.

Kısaltmalar	Açıklamalar
3D	Three Dimensional
API	Application Programming Interface
AS-DTW	Asymmetric Sub-Sequence Dynamic Time Warping
AVI	Audio Video Interleave
BIRTH SIM	Birth Simulator
BPTT	Backpropagation Through Time
CAM	Class Activation Map
CNN	Convolutional Neural Network
CPU	Central Processing Unit
CRF	Conditional Random Fields
DCC-CSM	Descriptive Curve Coding-Common String Model
DCT	Discrete Cosine Transform
DDTW	Derivative Dynamic Time Wrapping
DFT	Discrete Fourier Transform
DTW	Dynamic Time Wrapping
DVSS	Da Vinci Surgical System
GAP	Global Average Pooling
GMM	Gaussian Mixture Model
GMP	Global Max Pooling
GPU	Graphic Processing Unit
GRU	Gated Recurrent Unit
HMM	Hidden Markov Model
ISI	Intuitive Surgical, Inc.
JIGSAWS	JHU-ISI Gesture and Skill Assessment Working Set
KNN	K-Nearest Neighbour
LC-SC-CRF	Latent-Convolutional Skip-Chain CRF
LDA-HMM	Latent Dirichlet Allocation – Hidden Markov Model

Kısaltmalar**Açıklamalar**

LOUO	Leave One User Out
LSTM	Long Short-Term Memory
MIS	Minimally Invasive Surgery
MISTIC-SL	Minimally Invasive Surgical Training and Innovation Center Science of Learning
MS-RNN	Multi-Scale Recurrent Neural Network
MsM-CRF	Markov/semi-Markov – Conditional Random Fields
MSE	Mean Squared Error
NLP	Natural Language Processing
PCA	Principal Component Analysis
RAMIS	Robot-Assisted Minimally Invasive Surgery
RIOUS	Robotic Intra-Operative Ultrasound
RL	Reinforcement Learning
RNN	Recurrent Neural Network
S-HMM	Sparse Hidden Markov Model
SC-CRF	Skip-Chain CRF
SDSDL	Shared Discriminative Sparse Dictionary Learning
SELU	Scaled Exponential Linear Unit
SGD	Stochastic Gradient Descent
TCN	Temporal Convolutional Network
TDNN	Time Delay Neural Network
TPU	Tensor Processing Unit
TSC	Transition State Clustering

1. GİRİŞ

Robotik cerrahi, son yıllarda tıp alanında önemli ilerlemeler kaydetmiştir. Bu alandaki ilerlemeler, hastanede kalış süresinin kısalması, daha az enfeksiyon ihtimali ile hastayı daha hızlı iyileştirme potansiyeline sahiptir [1]. Otomatik olarak cerrahi durum değerlendirmesi yapabilmek ve hangi cerrahi hareketlerin yapıldığını anlamak, cerrahlara gerçek zamanlı geri bildirim vererek ameliyatlarda yaşanan istenmeyen durumları ve hataları engellemek adına önem taşımaktadır. İşlem sırasında cerrahın performansının otomatik olarak değerlendirilmesi ve cerraha özgü geri bildirimler sağlanması eğitim sürecini ve genel performansı iyileştirebilmektedir [2]. Örneğin, bilgisayar destekli bir cerrahi eğitim sistemi, hareketlerin doğru sırada yapılıp yapılmadığının gözlemlenmesini veya bir stajyerin en çok hangi hareketlerde zorlandığının tespit edilmesini sağlayabilir.

Cerrahi işlemlerin karmaşıklığı, cerrahın hareketlerindeki değişkenlik ve hastaya özgü anatomi gibi nedenler otomatik cerrahi hareket tanımda zorluk oluşturmaktadır. Cerrahi görevleri (dikiş atma, iğne geçirme, düğüm atma) tanımlamak için bu görevlere ait olan hareket verilerinin analiz edilmesi gerekmektedir. Farklı cerrahi hareketler farklı karmaşıklık düzeylerine sahiptir ve cerrahın beceri düzeyi değişebilmektedir [3]. Bu nedenle, öğrenilen becerileri değerlendirmek için cerrahın hareketlerinin modellenmesi önemlidir.

Robot destekli cerrahi sistemler, cerrahi robotlardan sensörler aracılığıyla veri toplayabilmektedir [4]. Bu veriler, cerrahın hareketleri ile ilgili bilgileri içerir. Da Vinci® gibi mevcut sistemler [5] ile cerrah tarafındaki konsolun ve hasta tarafındaki robotik kolun hareketlerinin kinematik verisi kaydedilir. Bu veriler cerrah komutlarıyla ve cihaz hareketleriyle doğrudan ilişkili olduğundan cerrahi hareket modelleme ve analizi alanlarında kullanılmaktadır [6]. Kinematik veriler genellikle robotik kolların konumu, dönüşü ve hızı gibi birden fazla özellik içermektedir. Kinematik verinin boyutu, video verilerinden daha düşüktür ve herhangi bir ön işlem yapılmadan da kullanılabilir [7]. Bu verilerden, cerrahi görevlerin ve hareketlerin tanınması [8, 9] üzerine çalışmalara ağırlık verilmiştir.

Makine öğrenmesi alanında görülen son zamanlardaki gelişmeler, cerrahi prosedürleri sistematik olarak daha iyi anlama imkânı sunmaktadır [10]. Bu nedenle, cerrahi hareket

tanımda temel adım olan dikiş atma, iğne geçirme ve düğüm atma gibi temel cerrahi hareketlerin doğru bir şekilde tanınabilmesini sağlayan teknikler geliştirilmektedir. Ayrıca, cerrahi hareketlerin gerçek zamanlı olarak tanınması, cerrahi olası hatalardan haberdar eden veya cerraha bir görev sırasında yardımcı olan akıllı asistanların oluşturulmasını sağlayabilir. Bu tür karmaşık cerrahi verilerin yüksek boyutsallığı beceri değerlendirmeleri ve cerrahi hareket analizleri için zorluk oluşturmaktadır [11]. Bu amaca yönelik olarak, daha etkili tekniklerin geliştirilmesine ihtiyaç duyulmaktadır [12].

Makine öğrenmesi, yapay zekânın bir alt alanıdır. Sabit kodlanmış kurallar yerine örneklerden ve deneyimlerden çıkarım yapan algoritmalar ile modelleme ve veri analizi yöntemlerini içerir. Yapay zekânın mevcut hedeflerinin geniş yelpazesinde, makine öğrenmesi tabanlı sağlık hizmetleri uygulamaları yer almaktadır. Makine öğrenmesi, sağlık sektörüne tıbbi görüntülemelerde teşhis, ilaç keşfi, karar destek sistemleri, kişiselleştirilmiş tıp ve çok daha fazlasını içeren çok çeşitli uygulamalarda kullanılmıştır [13]. Makine öğrenmesi, özellikle bilgisayarla görü ve derin öğrenme teknikleri, cerrahi hareketlerin sınıflandırılması ve tanınmasında kullanılabilir. Bu çalışmada, bildiğimiz kadarıyla ilk kez cerrahi hareket tanıma için kinematik veri ile zaman serisi verilerinde başarı gösteren Transformer modeli kullanılmıştır.

Son zamanlarda, doğal dil işleme bağlamında dikkat mekanizması ele alınarak RNN tabanlı kodlayıcı kod çözücü modeli yeniden tasarlanmıştır. Vaswani ve diğerleri [14], tarafından geliştirilen bir model olan Transformer dikkat mekanizmasını ve ileri besleme katmanlarını birleştirerek RNN ihtiyacını ortadan kaldırmıştır. Transformer modeli özellikle uzun metin dizilerinde çeviri performansını artırmıştır. Transformer modeli eğitim sırasında daha fazla operasyonun paralel olarak yürütülmesine izin verdiğinden RNN tabanlı sıralı modellerden daha hızlı eğitilebilmektedir [15].

Sıraya göre hizalanmış modellerin aksine Transformer, verileri sıralı bir şekilde işlemez. Bunun yerine, tüm veri dizisini işler ve dizideki bağımlılıkları öğrenmek için self-attention mekanizmalarını kullanır. Bu nedenle, Transformer tabanlı modeller, sıralı modeller için zorlayıcı olan zaman serisi verilerinin karmaşık dinamiklerini modelleme potansiyeline sahiptir. Transformer, başlangıçta makine çevirisi görevi için kullanılsa da daha sonra zaman serisi verileri üzerinde de mevcut çözümlerden daha iyi performans göstermiştir [15]. Bu çalışmada bir tür zaman serisi verisi olan JHU-ISI Gesture and Skill Assessment

Working Set (JIGSAWS) veri setindeki dikiş atma görevine ait kinematik veriler ile cerrahi hareket için Transformer modeli kullanılmıştır. Çalışma sonucunda, cerrahi hareket tanıma için Transformer modelinin zaman serisi tahmini görevine başarıyla uygulanabileceğini ve mevcut tahmin yöntemleri ile karşılaştırılabilir performans verdiği görülmüştür.

Çalışmada önerdiğimiz Transformer modelinin başarısının karşılaştırmak üzere cerrahi hareket tanıma literatürde sıkça kullanılan LSTM modeli ele alınmıştır. Bu modellerden biri bu alanda literatürde sıkça kullanılan LSTM modelidir. Her iki model de JIGSAWS verisetinde test edilmiş ve elde edilen sonuçlara göre Transformer modelinin mevcut yaklaşımlarla karşılaştırılabilir performans gösterdiği sonucuna varılmıştır.

Cerrahi hareket tanıma, üzerinde aktif olarak çalışılan bir alandır. Cerrahlara geri bildirim sağlamak ve yüksek doğrulukla cerrahi hareketleri tanımakta çalışmanın motivasyonunu oluşturmaktadır. Bu çalışma geliştirdiğimiz modelin performansının test edilmesi için cerrahi hareket tanıma için hazırlanmış JIGSAWS veri seti kullanılmıştır. Bu tezin amacı, zaman serisi tahmininde başarı gösteren Transformer modelinin bir tür zaman serisi olan kinematik veri ile cerrahi hareket tanıma kullanılmasıdır.

Özetle, bu çalışmanın aşağıdaki gibi üç önemli katkısı bulunmaktadır:

- Literatürde ilk kez Transformer modeli kinematik veri ile cerrahi hareket tanıma kullanılmıştır.
- Sonuçlara göre Transformer modelinin doğruluk payının literatürdeki LSTM modelleri sonuçları ile karşılaştırılabilir olduğu görülmüştür.
- Transformer modelinin standart sapması LSTM modellerinden daha azdır ve böylece Transformer modellerinin güvenilirliğini desteklemektedir.

Tez çalışmasının geriye kalan kısmı şu şekilde oluşturulmuştur: İkinci bölümde literatürdeki çalışmalardan bahsedilmiştir. Üçüncü bölümde, çalışmada kullanılan materyallerden olan veri setinden, kinematik veriden ve kullanılan yazılım geliştirme kütüphanelerinden bahsedilmiştir. Dördüncü bölümde, LSTM ve Transformer Network modellerinin temellerinden bahsedilmiştir, daha sonra bu modeller daha kapsamlı bir şekilde açıklanmıştır, ardından modellerin performansını etkileyen hiperparametrelerden ve modellerin performansını ölçmek için kullanılan yöntemlerden bahsedilmiştir. Beşinci

bölümde, veri setinin girdi olarak işlenmesi ve ön hazırlığından sonra modellerin, eğitimi ve modelin yeni veriler ile testinden bahsedilerek, elde edilen deneysel sonuçları ve karşılaştırmalı analizler gösterilmektedir. Altıncı bölümde bu çalışma sonuçlandırılıp gelecekteki çalışmalar tartışılmıştır.



2. LİTERATÜR TARAMASI

Klinik cerrahi cihazlardan kaydedilen kinematik veriler, analiz edilerek cerrahi beceri değerlendirmesinde [16, 17] ve cerrahi hareket tanımada [3, 18, 19] kullanılmıştır. Genel olarak bakıldığında cerrahi robotlar tarafından kaydedilen kinematik ve video verileri kullanılarak hareket tanıma ve cerrahi beceri değerlendirmesi için geliştirilen modeller arasında Hidden Markov Model (HMM) [20] ve Conditional Random Fields (CRFs) [21] modelleri ile RNN [18], Temporal Convolutional Network (TCN) [22] gibi derin öğrenme tabanlı modeller ve Reinforcement Learning (RL) [23] modelleri kullanılmıştır.

Dynamic Time Warping (DTW), zaman serisi sınıflandırması için iyi bilinen bir tekniktir ve cerrahi prosedür uygulamasında cerrahi süreçleri ve cerrahi hareketleri sınıflandırmak için kullanılmıştır [24, 25]. DTW birçok alanda başarılı bir şekilde kullanılmış olsa da zaman içinde sinyal fonksiyonlarında önemli bir fark olduğunda iki sekans arasındaki doğal hizalamaları bulamama durumu olduğu için Derivative Dynamic Time Wrapping (DDTW) [26]' da önerilmiş ve bu konuyu ele almak için ümit verici sonuçlar sağladığı gösterilmiştir. Reiley ve Hager [20], cerrahın becerisini sınıflandırmak için alt görev seviyesinde birden fazla Sparse Hidden Markov Model (S-HMM) oluşturmuşlardır. Sonuçlar cerrahın becerisini %100 doğruluk payı ile tanıdığını göstermiştir.

Tao, Elhamifar, Khudanpur, Hager ve Vidal [27], cerrahi hareketlerin ve becerilerin sınıflandırılması için S-HMM yöntemini tanıtmıştır. Önerilen yöntemin son teknoloji yöntemlere eşit ya da daha yüksek performans gösterdiği belirtilmiştir. Lea, Hager ve Vidal [28], hareketler arası geçişleri daha iyi yakalamak için Skip-Chain CRF varyasyonu geliştirmiştir. Yaklaşım dikiş atma, düğüm atma, iğne geçirme görevlerinde test edilmiştir. Sonuç olarak, mikro ve makro doğruluğunun son teknolojiye göre %18,46 ve %44,13 oranında arttığını belirtmişler. Van Amsterdam, Nakawala, De Momi ve Stoyanov [17], cerrahi hareketlerin tanınması için, zayıf denetimli Gaussian Mixture Model (GMM) tabanlı yeni bir yaklaşım önermiştir. Böylece, girdileri optimize ederek tanıma doğruluğunu iyileştirmesindeki etkisi gösterilmiştir.

Sefati, Cowan ve Vidal [29], da Vinci robotik cerrahi sistemi tarafından kaydedilen karmaşık cerrahi hareketleri etkili bir şekilde analiz etmek için kullanılabilen Shared Discriminative Sparse Dictionary Learning (SDSDL)' e dayalı cerrahi hareket

sınıflandırma yöntemi önermiştir. SDSDL sonuçları sadece kinematik veriler kullanılarak sınıflandırılmasında S-HMM' den daha iyi sonuç verdiği ve SDSDL' in ayrıca Latent Dirichlet Allocation – Hidden Markov Model (LDA-HMM) ve Markov/semi-Markov – Conditional Random Fields (MsM-CRF) dâhil olmak üzere zamanının diğer son teknoloji istatistik yöntemlerinden daha iyi performans gösterdiği belirtilmiştir.

Krishnan ve diğerleri [30], doğrusal dinamik hareketler arasındaki geçişleri tespit ederek ve kümeleyerek bir Transition State Clustering (TSC) yöntemi önermişlerdir. Önerilen yöntemin, GMM tabanlı yöntemlerden %20' ye kadar daha doğru sonuç verdiği ve iğne geçiş görevinde %83, dikiş atma görevinde %73 doğruluğunun olduğu belirtilmiştir. Islam, Kahol, Li, Smith ve Patel [31], Minimally Invasive Surgery (MIS) için verimli, etkili ve ekonomik video tabanlı beceri değerlendirme tekniğine sahip sistem yani, hem gerçek zamanlı ekran geri bildirimi hem de ameliyat sonunda bir performans puanı sağlama yeteneği ihtiva eden bir sistem önerilmiştir. Bu çalışmadan önceki yaklaşımların dezavantajının cerrahların beceri değerlendirme çıktılarını anlamada ve bu çıktıları performanslarını iyileştirmek için kullanmakta güçlük çekmeleri olarak belirtilmiştir. Bu sorunu çözmek için web tabanlı cerrahi beceri eğitimi ve değerlendirme aracı önerilmiştir. Önerilen yöntem, çıkarılan özellikler ile cerrahın skoru arasındaki ilişkiyi bulmak için çok değişkenli bir doğrusal regresyon modeli ile cerrahın el ve cerrahi alet hareketlerinin videosunu analiz eden bir hareket tanıma algoritmasına dayanmaktadır.

Fard, Ameri ve Ellis [32], robotik destekli cerrahinin kinematik sinyali üzerinde hareket tanıma modeli geliştirmiş. Geliştirilen model iki temel bileşenden oluşmaktadır. İlk bileşen, farklı cerrahlar tarafından gerçekleştirilen ameliyatlarda arasındaki benzerliği ölçmek ve bunları k-Nearest Neighbors (kNN) yaklaşımına göre tanımaktadır. Diğer bileşen ise, sınıflandırma sonucuna göre cerrahın her hareketi yapma performansını değerlendiren kısımdır. Otomatik kişiselleştirilmiş Robot-Assisted Minimally Invasive Surgery (RAMIS) hareket eğitimi çerçevesi oluşturmuşlardır. Geliştirilen model dikiş atma görevinde %82, iğne geçirme görevinde %70, düğüm atma görevinde %85 doğruluk göstermiştir. Son teknoloji yöntemlerden daha iyi veya eşit performans gösterdiği belirtilmiştir.

Tao, Zappella, Hager ve Vidal [21], kinematik ve video verilerinden cerrahi hareketlerin ortak olarak tanınması için bir model önermişlerdir. Önceki çalışmalardan farklı olarak, MsM-CRF modeli kullanarak her iki modelden de yararlandıklarını belirtmişler.

Deneylerden elde edilen bilgilere göre, yalnızca kinematik verilerde kullanıldığında son teknoloji yöntemlerle karşılaştırılabilir performansa sahip olduğu belirtilmiştir. Zia ve Essa [33], yalnızca robot kinematik verilerini kullanarak otomatik beceri değerlendirme için farklı bütünsel özelliklerin kullanımını araştırmış ve beceri tanıma performansını artırmak için ağırlıklı bir özellik füzyon tekniği önermiştir. Sonuçlara göre, önceki tüm HMM tabanlı son teknoloji yöntemlerden daha iyi performans gösterdiği belirtilmiştir.

Makine öğrenmesinin amacı büyük miktarda veriden öğrenmek, çıkarım yapmaktır. Makine öğrenmesi, yorumlama ve gerçek zamanlı karar verme ile büyük miktarda verinin verimli bir şekilde işlenmesine izin verdiği için popülerdir [5]. Özellikle, bilişsel cerrahi robotlar kavramında ezber bozan bir unsur haline gelmiştir. Bilişsel becerilere sahip bu tür robotlar, ameliyat ekibine bilişsel düzeyde de yardımcı olmaktadır. Makine öğrenmesi, insan uzmanlar tarafından gösteri yoluyla öğrenilen cerrahi becerinin tanınmasına yardımcı olabilir ve bunu robotik becerilere aktarabilir.

Fard ve diğerleri [34], cerrahın performansını otomatik olarak değerlendirmek için makine öğrenme yöntemli bir model önermiştir. Değerlendirmede 6 önemli öznelik; tamamlanma süresi (görevi tamamlamak için gereken toplam süre), yol uzunluğu (görevi gerçekleştirirken aletin ucu tarafından açıklanan eğrinin uzunluğu), derinlik algısı (aletin eksen boyunca kat ettiği toplam mesafe), hız (önceki zaman adımından konum değişikliği oranı), hareket pürüzsüzlüğü (hızlanma ve yavaşlamanın ritmik modelinin ölçüsü) ve eğrilik kullanılmıştır. Sonuç olarak, önerilen çerçevenin doğruluğu %85,7 olarak belirtilmiştir. Despinoy ve diğerleri [35], cerrahi hareketlerin tanınması ve değerlendirilmesi için kinematik verileri otomatik olarak bölümlere ayırabilen yeni bir denetimsiz algoritma önermişlerdir. Önerilen algoritma, denetimsiz segmentasyon ile cerrahi hareketlerin öğrenilmesi ve tanınmasının sıralı olarak uygulanması ile elde edilmiştir. Yaklaşımları %97,4 doğruluk ile sonuçlanmıştır.

Zia ve diğerleri [36], cerrahın beceri seviyesini sınıflandırmak için frekans özellikleri Discrete Cosine Transform (DCT) ve Discrete Fourier Transform (DFT) ile hareketlerde tekrarlanabilirliği kullanmıştır. Çalışmalarındaki tüm deneyler Leave-one-out cross-validation (LOOCV) uygulanarak yapılmıştır. Elde edilen deneysel sonuçlarda, DFT ve DCT kullanılarak bilinen en gelişmiş tekniklere göre önemli performans iyileştirmeleri gösterdiği belirtilmiştir. Ahmidi ve diğerleri [37], eşzamanlı cerrahi hareket tanıma ve

beceri deęerlendirmesi iin Descriptive Curve Coding-Common String Model (DCC-CSM) nermiřlerdir. Hareketleri ve beceriyi sırasıyla %87,81 ve %91,12 doęrulukla sonulanmıřtır.

Gao ve dięerleri [19], bir sorgu ile hareketleri otomatik olarak tespit etmek iin bir ierięe dayalı sorgulama yntemi (Query-by-Example) geliřtirmiřlerdir. Aynı cerrah aynı deneme, aynı cerrah farklı deneme ve farklı cerrah deney kurulumlarında doęruluk deęerlerinin sırasıyla %72, %61, %53 olduęu gzlemlenmiřtir. Gao ve dięerleri [38], cerrahi aktiviteleri tanımak iin denetimsiz bir ęrenme yntemi geliřtirmiř ve deęerlendirmiřlerdir. Yaklařımları, sırasıyla Asymmetric Sub-Sequence Dynamic Time Warping (AS-DTW) ve řablon eřleřtirmeye dayalı iki puanlama modln ve bir algılama modln kullanan bir Stacked Denoising Autoencoder (SDAE) kullanan denetimsiz bir zellik ęrenme modl ierir. Bu alıřmada, cerrahi kayıtlarda kinematik verilerin bilgilendirici bir temsilini elde etmek iin doęrusal olmayan bir znitelik ęrenme yntemi uygulanmıřtır. alıřmanın sonucunda, JIGSAWS zerinde aynı cerrah aynı deneme, aynı cerrah farklı deneme ve farklı cerrah deney kurulumları iin sırasıyla %93, %91, %88 doęruluk deęerleri, Minimally Invasive Surgical Training and Innovation Center - Science of Learning (MISTIC-SL) zerinde aynı cerrah aynı deneme, aynı cerrah farklı deneme ve farklı cerrah deney kurulumları iin sırasıyla %72, %61, %53 doęruluk deęerleri elde etmiřlerdir.

Bani ve Jamali [39], robotik cerrahi iin son teknoloji veri madencilięi ve zaman serisi analiz yntemlerini birleřtiren bir model nermiřtir. Bu erevenin ilk adımı, DDTW kullanarak bir zaman serisi mesafe tabanlı benzerlik lm geliřtirmektir. Daha sonra sınıflandırmak iin kNN algoritması kullanılmıřtır. Deneyisel sonulara gre nerilen erevenin performansı %99,4' tr. Anh, Nataraja ve Chauhan [15], otomatik cerrahi beceri deęerlendirmesi zerinde zellik ıkarma tekniklerinin performansını ele almıřlardır. Bu performansın neredeyse gerek zamanlı olarak karřılařtırılmasını saęlamıřlardır. Bilinen dokuz zellik ıkarma teknięi zerinde karřılařtırmalı bir analiz yapmıřlardır. Otomatik cerrahi beceri deęerlendirmesinde farklı zellik ıkarma tekniklerinin (Statistical Features, Principal Component Analysis, Discrete Fourier / Cosine Transform, Codebook, Deep Learning Models, Auto-Encoder) performanslarını karřılařtıran bir alıřma nerilmiřtir. Principal Component Analysis ile dikiř atma, dęm

atma ve iğne geçişi için sırasıyla %95,63, %90,17, %90,26; Convolutional Neural Network (CNN) ile %96,84, %92,75, %95,36 doğruluk sonucu verdiği belirtilmiştir.

Çoğu sınıflandırma probleminde iyi bir performans sağlamak için yapay sinir ağı yöntemi kullanılmıştır ve sinir ağları kullanan çalışmaların çoğu durağan, hareket içermeyen el hareketlerini tanımaya odaklanmıştır [40]. DiPietro ve diğerleri [18], önceki çalışmaların HMM ve CRF ile kısa süreli, düşük seviyeli faaliyetleri veya hareketleri tanımaya odaklandığını ve bu yüzden uzun süreli dinamikleri verimli olarak kullanmadıklarını belirtmişlerdir. Çalışmalarında HMM veya CRF tabanlı yöntemler yerine RNN ile modelleme yapmışlardır. Hem JIGSAWS hem MISTIC-SL veriseti üzerinde çalışmışlardır. Yüksek bir tanıma doğruluğu ve düzenleme mesafesi elde etmek için bir LSTM ve Bidirectional LSTM önerilmiştir. Bilindiği kadarıyla bu çalışma ile kinematik veri üzerine ilk kez RNN uygulanmıştır. Her iki verisetinde de deneylerde her cerrah test kullanıcısı olarak çalıştırılarak tüm sonuçların ortalaması alınmıştır. Her çalıştırma için 80 epoch kullanılmıştır. İlk 40 epoch için öğrenme oranı 1,0 iken daha sonra her 5 epochta bir öğrenme oranı yarıya inmiştir. Sonuçlara göre, hareket tanıma için doğruluk payı %81,7 ile %89,5 arası, düzenleme mesafesi %29,7 ile %19,5 arası değişmiştir. Bu çalışmadaki sonuçlar, bizim çalışmamızda da karşılaştırma amacıyla kullanılmıştır.

Lea, Vidal, Reiter ve Hager [22], düşük, orta ve yüksek düzey zaman ölçeklerinde ilişkileri hiyerarşik olarak yakalayan TCN modeli önermişlerdir. Modelin bir RNN modelini eğitmek için gereken sürenin çok altında eğitilebildiği ve sonuçların rekabetçi performans gösterdiği belirtilmiştir. Menegozzo, Dall'Alba, Zandonà ve Fiorini [3], bir cümlede önceki kelimeler üzerinde çalışmak yerine hareket tanıma için kinematik verileri üzerinde çalışan bir Time Delay Neural Network (TDNN) önermiştir. Elde edilen sonuçların literatürde bulunan diğer yöntemlerle karşılaştırılabilir olduğu belirtilmiştir.

Liu ve Jiang [23], önceki çalışmalardan farklı olarak, cerrahi hareket sınıflandırması için farklı bir RL yaklaşımı önermiştir. Görsel veya kinematik veri dizisiyle etkileşime girecek bir ajan ederek, başlangıçta veri dizisinin başında konumlanarak uygun bir adım boyutu seçilir ve her bir zaman adımında ileriye doğru hareket ederek eşzamanlı olarak sınıflandırma yapar. Önerilen yöntem JIGSAWS veri setinin dikiş atma görevi üzerinde test edilmiştir ve son teknoloji yöntemlerden daha iyi performans göstermiştir. Sarıkaya, Guru ve Corso [41], video ve optik akış kullanan robot destekli cerrahi videolarında aynı

anda düşük seviye hareket ve cerrahi görev sınıflandırması için yeni birçok modlu ve çok görevli mimari önermişlerdir. Önerilen mimari, önceki çalışmalardan farklı olarak farklı görevlerdeki ortak hareketleri tanımaya odaklanmış ve geleneksel mimarilere kıyasla önemli ölçüde daha iyi performans göstermiştir.

Wang ve Fey [42], derin öğrenme tabanlı yeni bir uçtan uca analitik model sunmuşlardır. Önerilen model, dikiş atma, iğne geçirme, düğüm atma görevlerinde sırasıyla %92,5, %95,4, %91,3 doğruluk elde etmiştir. Funke ve diğerleri [43], ardışık video karelerinden uzamsal-zamanlama ipuçlarını yakalamak için Three Dimensional- CNN (3D CNN) modeli önermişlerdir. Model, karşılaştırılabilir modellerden daha iyi performans göstererek, dikiş atma görevi %84' ün üzerinde doğruluğa ulaşmıştır. Cifuentes, Boulanger, Pham, Prieto ve Moreau [44], laparoskopik forseps aleti ile elde edilen üç boyutlu uzaydaki konumlarını belirten tıbbi hareketleri, LSTM kullanarak sınıflandırmışlardır. BirthSIM (Birth Simulator) simülatörü ile birleştirilmiş aletli bir obstetrik forseps, her forseps için 3D pozisyon verileri kullanılarak sonuçlara göre %99,1' lik bir tanıma oranı elde edilmiştir. Castro, Pereira, Zanchettin, Macêdo ve Bezerra [16], otomatik robot destekli beceri değerlendirmesi için yeni bir CNN mimarisi önermişlerdir. Scaled Exponential Linear Unit (SELU) aktivasyon fonksiyonunun kullanımını, Global Average Pooling (GAP) ve Global Max Pooling (GMP) katmanlarına dayanan GMP yaklaşımını araştırmışlardır. Gurcan ve Van Nguyen [45], hem dalgacık saçılma işlemlerinin hem de LSTM' in gücünü birleştiren Multi-Scale Recurrent Neural Network (MS-RNN) önermişlerdir. Sonuçlara göre son teknoloji yöntemlerden daha iyi performans gösterdiği ve önerilen ağın hareket tanıma doğruluğu %90 olarak belirtilmiştir.

Itzkovich, Sharon, Jarc, Refaely ve Nisky [12], dikiş atma görevini hareketlere ayırmak için RNN kullanmıştır. Bu çalışmada, cerrahi veri biliminde kinematik verilerin analizinde veri artırmayı (Augmentation) kullanma olasılığını vurguladığı belirtilmiştir. DiPietro ve diğerleri [46], kinematik veriden cerrahi hareket tanıma için dört RNN mimarisini MISTIC-SL ve JIGSAWS verisetleri üzerinde karşılaştırmışlardır: Standart RNN, LSTM, Gated Recurrent Unit (GRU), Mixed History RNN. GRU' lar için manevra tanıma görevinde hata oranı $8,6 \pm 3,4$, düzenleme mesafesi $9,3 \pm 4,3$ olarak sonuçlanmıştır. Fawaz, Forestier, Weber, Idoumghar ve Muller [47], robotik cerrahi sırasında cerrahların hareketlerinde gizli kalıpları çıkararak cerrahi becerileri sınıflandırmak için yeni bir CNN modeli tasarlamışlardır. Zamansal kinematik verileri kullanarak cerrahi hareket

değerlendirmesi için özel olarak tasarlanan modelin orijinal hiyerarşik yapısı, cerrahi beceri seviyesine özgü bilgileri yakalamaya ve aynı zamanda gizli düşük seviyeli özelliklerdeki hareketlerin temsil edilmesine olanak tanımıştır. Sonuçlar, derin sinir ağlarının JIGSAWS veri setinde son teknoloji performansa ulaşabilen bir model olduğunu göstermiştir. Ayrıca, yorumlanabilir bir geri bildirim sağlamak için, çoğu geleneksel derin öğrenme mimarisi gibi yoğun bir katman kullanmak yerine [48], Class Activation Map (CAM)’ den yararlanmayı sağlayan GAP katmanının kullanımı önerilmiştir.

Qin ve diğerleri [49], da Vinci cerrahi sisteminden çeşitli cerrahi görevler için birleşik tahmin modeli olan Fusion-KVE’ yi önermişlerdir. Modeller ile hem JIGSAWS veriseti dikiş atma görevi hem de Robotic Intra Operative Ultrasound (RIOUS) veri seti için %89,4’ e varan üstün çerçeve bazlı durum tahmin doğruluğu elde edilmiştir. Sarıkaya ve Jannin [50], cerrahi hareket tanıma için videolardaki cerrahi aletlerin uzaysal zamansal grafik temsillerine dayanarak modelleme yapmışlardır ve önerdikleri modelin cerrahi hareket tanımda iyi performans gösterdiğini belirtmişlerdir. Van Amsterdam, Clarkson ve Stoyanov [51], kinematik veri ile cerrahi hareketlerin eşzamanlı olarak tanınması için Multi-Task RNN önermişlerdir. Önceki çalışmalardan farklı olarak, kinematik özellik varyasyonunu azaltmak amacıyla zaman yerine temel hareket dizisi üzerinde tanımlanmıştır.

Transformer ağları, ilk olarak doğal dil çevirisi görevi için önerilen [14], yakın zamanda geliştirilmiş bir derin öğrenme modelidir. Transformer ağlarının doğal dil işlemedeki yaygın başarısı için önemli bir faktör, denetimsiz ön eğitim yoluyla doğal dili nasıl temsil edeceklerini öğrenme yetenekleridir [52]. Doğal dil işleminin yanı sıra, Transformer ağları polifonik müzik kompozisyonu gibi çeşitli dizi oluşturma alanlarında da kullanılmıştır. Huang ve diğerleri [53], dikkat mekanizmasını değerlendirmişlerdir ve bu modelleme için Transformer’ ın uygun olduğunu göstermişlerdir. Song, Rajan, Thiagarajan ve Spanias [54], ilk kez zaman serisi modellemesi için dikkat mekanizmasından yararlanmışlardır. Önerilen yöntemin, son teknoloji performansa ulaştığı, LSTM modellerinden daha iyi performans gösterdiği belirtilmiştir. Li ve diğerleri [55], zaman serisi tahmini gibi tahmin sorunlarını Transformer modeli ile çözmeyi önermişlerdir. Hem sentetik hem gerçek veri setleri üzerindeki deney sonuçlarına göre son teknoloji ile olumlu bir şekilde karşılaştırıldığını belirtmişlerdir. Ma ve diğerleri [56], çok değişkenli, coğrafi etiketli zaman serilerinde eksik değerlerin tespiti için öz dikkat mekanizmasına sahip bir

kodlayıcı- kod çözücü mimarisi ilk kez kullanmışlardır. Son teknolojiye sahip bu mimari, RNN tabanlı ispat yöntemlerinden daha iyi performans göstermektedir. Wu, Green, Ben ve O'Banion [57], zaman serisi tahminine yeni bir yaklaşım sunmuşlardır. Bu çalışmada, zaman serisi verilerini tahmin etmek için Transformer tabanlı makine öğrenme modellerini kullanan yeni bir yöntem geliştirilmiştir. Sonuçlara göre, LSTM ve GRU sıralı modeline dikkatle kıyasla son teknoloji ile olumlu bir şekilde karşılaştırılabilir olduğu belirtilmiştir. Czempiel ve diğerleri [58], yakın zamanda Transformer modelini video verileri ile cerrahi hareketleri tanımak için çalışma yapmışlardır ancak bizim çalışmamızda, bildiğimiz kadarıyla, literatürde ilk kez Transformer modeli kinematik veri ile cerrahi hareket tanımada kullanılmıştır.



3. MATERYALLER

Bu bölümde çalışmamızın ele aldığı JIGSAWS veri setinden, kinematik veri özelliklerinden ve kullandığımız kütüphaneler ve geliştirme ortamlarından bahsedilecektir.

3.1. JIGSAWS Veri Seti

Dünyadaki hastanelerde ve diğer kurumlarda 500' den fazla da Vinci cerrahi sistemi kullanılmaktadır [59]. Bu gibi robotik sistemler ile cerrahlar cerrahi görevleri daha hassas gerçekleştirebilmektedir. Cerrah, Resim 3.1' deki konsola oturarak operasyon bölgesini endoskop tarafından sağlanan 3 boyutlu görüntüsünü bir göz merceğinden izler [60]. Cerrahın parmakları ekranın altındaki tutamaçları kavrar ve el hareketleri, tutamaçlardan cerrahi sistem üzerindeki Resim 3.1' deki cerrahi aletlerin uçlarına iletilir. Sistem, cerrahın hareketlerini, hastanın içindeki cerrahi aletlerin hassas, gerçek zamanlı hareketlerine çevirir. Bu teknolojinin avantajı, bilgisayarın cerrahın el hareketini arttırması ve daha az titreme ile daha hassas hale getirmesidir [60].



Resim 3.1. Da Vinci cerrahi sistemi [60]

Bu çalışmada JIGSAWS veri seti kullanılmıştır. JIGSAWS verisetini oluşturan cerrahi görevler, Johns Hopkins Üniversitesi' ndeki bir grup fakülte cerrahı tarafından gerçekleştirilmiştir. Johns Hopkins Üniversitesi ve Intuitive Surgical®, Inc (ISI) arasında ortak bir proje kapsamında, insan hareket modellemesi için toplanan cerrahi hareket veri

setidir [61]. Bu veri seti kullanılarak yapılan araştırma çalışmalarında kontrol teorisi ve bilgisayarlı görü ile ilgili teknikler geliştirilmiş ve uygulanmıştır.

JIGSAWS veri seti temel cerrahi görev olan dikiş atma, düğüm atma, iğne geçirme görevleri cerrahlar tarafından gerçekleştirilirken da Vinci cerrahi sisteminin kaydettiği video ve kinematik verileri içerir [61]. JIGSAWS veri seti, uzman, orta ve acemi olmak üzere farklı deneyime sahip sekiz farklı cerrahın gerçekleştirdikleri görevlere kinematik ve video verilerini içerir. Cerrahlar deneyimlerine göre {"B", "C", "D", "E", "F", "H", "I"} ile kodlanmıştır. D ve E cerrahları 100 saatten fazla, B, G, H ve I cerrahları 10 saatten az, C ve F cerrahları 10 ila 100 saat arasında robotik cerrahi deneyimi olduğunu bildirmiştir [61]. Tüm cerrahlar sağ elini kullanmıştır. Bu görevler bir model üzerinde gerçekleştirilmiştir. Üç görevin birer örneği Resim 3.2' de gösterilmektedir. Her cerrah, üç cerrahi görevden her biri için beş deneme gerçekleştirmiştir.



Resim 3.2. JIGSAWS veri setindeki 3 cerrahi görev (soldan sağa): dikiş atma, düğüm atma, iğne geçirme [61]

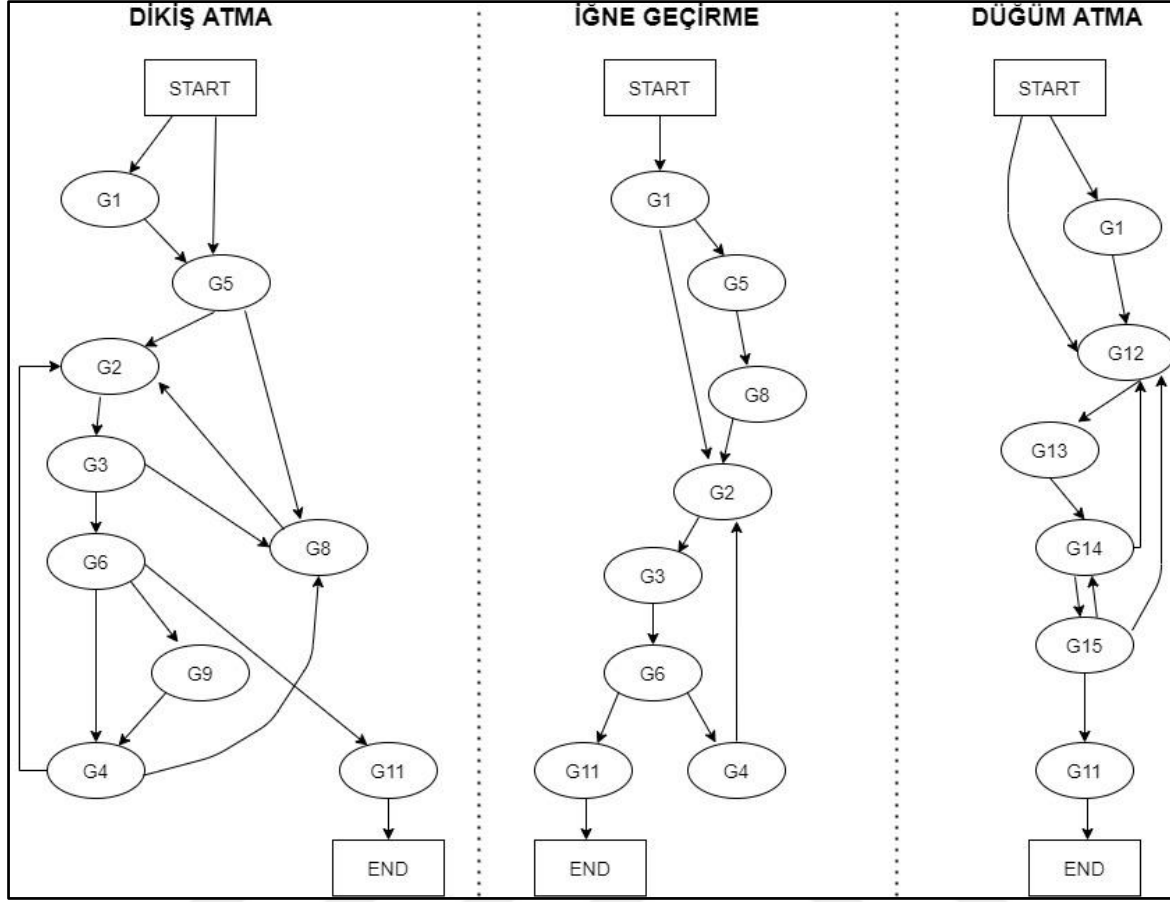
Üç görevden ilki dikiş atma görevidir; bu görevde cerrah iğneyi alır, dikey bir çizgiyi takip eder, bu çizgi kesiği temsil etmektedir. Bu kesiğin etrafında işaretli bir noktadan dokuya geçirilir, bu iğneyi ve diğer tarafında işaretli bir diğer noktadan çıkarılır, yani atılacak dikiş bu çizgi ve noktalara göre yapılmaktadır. İlk iğne geçişinden sonra cerrah iğneyi dokudan çıkarıp sağ eline alıp aynı işlemi 3 kez tekrar etmiştir [61]. Diğer görevlerden düğüm atma görevinde cerrah, düğümü düğüme bağlar. Plastik bir çubuğun etrafına basit ilmekli bir düğüm bağlanır. Son olarak iğne geçirme görevinde, cerrah iğneyi alır ve sağdan sola dört küçük metal halkadan geçirir.

JIGSAWS veri setinde kullanılan cerrahi hareketler, Johns Hopkins Üniversitesi' ndeki bir grup fakülte cerrahı tarafından tanımlanmıştır. Hareketler:

- (G1) Sağ el ile iğneye ulaşmak
- (G2) İğnenin ucunu konumlandırma
- (G3) İğneyi doku içinden itme
- (G4) İğnenin soldan sağa aktarılması
- (G5) İğne kavrama ile çalışma alanının ortasına hareket
- (G6) Sol elle dikişin çekilmesi
- (G7) Dikişi sağ elle çekme
- (G8) İğneyi yönlendirme
- (G9) Dikişin sıkılmasına yardımcı olmak için sağ elin kullanılması
- (G10) Daha fazla dikiş gevşetme
- (G11) Dikişin düşürülmesi ve uç noktalara taşınması
- (G12) Sol elle iğne için uzanma
- (G13) Sağ el çevresinde C halkası yapılması
- (G14) Sağ elle dikiş için uzanma
- (G15) Dikişi iki el ile çekme

Şekil 3.1, her bir görevin yürütülmesi sırasında farklı hareketlerin ilişkisini, sırasını ve akışını temsil eder. Dikiş atma görevinde G7 hariç G1' den G11 ' e kadar hareketler kullanılır. İğne geçişinde G1 ila G6, G8 ve G11 kullanılır. Düğüm atma G1 ve G11 ila G15 kullanılır.

JIGSAWS kinematik veriler, video verileri ve manuel ek açıklamalar olarak üç bileşenden oluşmaktadır. 30 Hz' de ve 640 x 480 çözünürlükte da Vinci Surgical System (dVSS)' nin her iki endoskopik kamerasından stereo video yakalanmıştır. Video ve kinematik veriler, her video çerçevesi aynı anda yakalanan bir kinematik veri çerçevesine karşılık gelecek şekilde senkronize edildi. Serbest bırakılan veri setindeki videolar, DX50 koduyla dört karakter kodu biçiminde kodlanmış Audio Video Interleave (AVI) dosyaları olarak kaydedilir.



Şekil 3.1. Dikiş atma (sol), düğüm atma (orta), iğne geçirme (sağ) görevlerine ait hareket sırası

3.2. Kinematik Veri

Kinematik özelliğin boyutu, videodan çıkarılanlardan daha düşüktür ve herhangi bir ön işlem yapılmadan da kullanılabilir. JIGSAWS veri setinde, kinematik veriler hem cerrah hem de hasta tarafındaki sol ve sağ manipülatörlerinin değişkenlerini içerir (saniyede 30 saniyelik görüntü sayısı toplanan 76 hareket değişkeni: araç ipucu konumları ve yönü, doğrusal ve dönme hızları ve tutucu açısı). Stereo video, 640x480 piksel çözünürlüğünde saniyede 30 kare hızında kaydedilir. Video ve kinematik veriler, ilgili ve tamamlayıcı bilgileri taşır. Kinematik veriler, doğrudan videoda ölçülemeyen robot kollarının üç boyutlu yörüngelerini, üç boyutlu hızlarını içerir. Video verileri, kinematik verilerde doğrudan bulunmayan araçlar ve doku arasındaki etkileşim gibi bağlamsal ve anlamsal bilgiler sağlar [61]. Her yönlendiricinin hareketi için 19 kinematik değişken bulunmaktadır: 4 araç yönlendiricisini dikkate alan, kartezyen konumları (xyz ile gösterilen 3 değişken), bir döndürme matrisi (R ile gösterilen 9 değişken), doğrusal hızları

($x'y'z'$ ile gösterilen 3 değişken), dönme hızları ($\alpha'\beta'\gamma'$ ile gösterilen 3 değişken) ve kavrayıcı açısı (θ).

Tüm kinematik değişkenler ortak bir koordinat sistemi içinde temsil edilir. Çizelge 3.1' de, kinematik veri setinde yer alan değişkenlerin ayrıntılarını açıklamaktadır.

Çizelge 3.1. Kinematik verinin değişkenleri

Sütun İndeksleri	Değişken Sayısı	Değişkenlerin Tanımı
1-3	3	Sol cerrah araç yönlendiricisi ucu konumu (xyz)
4-12	9	Sol cerrah araç yönlendiricisi ucu döndürme matrisi (R)
13-15	3	Sol cerrah araç yönlendiricisi ucu doğrusal hızı ($x'y'z'$)
16-18	3	Sol cerrah araç yönlendiricisi ucu dönme hızı ($\alpha'\beta'\gamma'$)
19	1	Sol cerrah araç yönlendiricisi ucu kavrayıcı açısı (θ)
20-38	19	Sağ cerrah araç yönlendiricisi ucu kinematığı
39-41	3	Hasta tarafı yönlendiricisi1 araç ucu konumu (xyz)
42-50	9	Hasta tarafı yönlendiricisi1 araç ucu döndürme matrisi (R)
51-53	3	Hasta tarafı yönlendiricisi1 araç ucu doğrusal hızı ($x'y'z'$)
54-56	3	Hasta tarafı yönlendiricisi1 araç ucu dönme hızı ($\alpha'\beta'\gamma'$)
57	1	Hasta tarafı yönlendiricisi1 kavrayıcı açısı (θ)
58-76	19	Hasta tarafı yönlendiricisi2 kinematığı

Verilerin hazırlanması, ön işlemden geçirilmesi, temizlenmesi gibi işlemler projenin en kritik adımını oluşturmaktadır. Çalışmaya başlamadan önce, mevcut verileri anlamak, amaca yönelik olarak hazırlamak önemlidir. Veri temizleme, verileri normalize etmek, şekillendirme süreci, verinin analiz aşamasına hazır hale gelmesini sağlar. Toplanan veriler, tutarsızlıklar, eksik veriler, hatalar, geçersiz girdiler içerebilmektedir. Veriler farklı ve çeşitli kaynaklardan toplanabilir, elde edilebilir. Bu kaynaklar, aynı bilgileri içerebilse bile aynı formatta olmayabilir. Bu gibi nedenlerden dolayı da verilerin temizlenmesi, model oluşturulmasına hazır ve kullanılabilir hale getirilmesi gerekmektedir. Bölüm 5' te bu çalışma için veri setinin hazırlanma sürecinden bahsedilmiştir (Bkz. Bölüm 5.1).

Çalışmamızda sadece dikiş atma görevi üzerinde çalışılmıştır ve bunun için JIGSAWS veri setindeki dikiş atma verileri kullanılmıştır. Veri seti, dikiş atma işleminin gerçekleştirildiği videoların her bir video çerçevesine karşılık gelen robot hareketlerinden alınan kinematik verilerden oluşmaktadır.

3.3. Kullanılan Araçlar

Bu çalışmada Python programlama dili kullanılmıştır. Bu bölümde de Python' a ait olan çalışmada kullanılan kütüphanelerden ve kullanılan ortamlardan bahsedilmiştir.

3.3.1. Python programlama dili

Çalışmamızdaki modelleri kodlamak ve uygulamak için Python programlama dili kullanılmıştır. Python, kullanımı kolay bir programlama dilidir. Ayrıca, birçok kütüphane, modül ve uygulama geliştirme çerçeveleri ile uyumludur. Bu durum, çok sayıda çalışma alanında, özellikle veri bilimi ve makine öğrenmesi alanları için Python programlama dilini tercih edilen bir dil haline getirmektedir. Ek kütüphanesi NumPy, çok boyutlu dizileri destekler ve veri bilimi için yararlı olan yüksek seviyeli matematiksel fonksiyonlar ekler [62, 63].

Python programlama dili, veri işleme ve makine öğrenimi algoritmaları için çok sayıda bilimsel kütüphaneye sahip olduğu ve birçok derin öğrenme kütüphanesinin Python ile uyumlu olduğu için tercih edilmiştir. Çalışmamızı gerçekleştirmek için Keras, Scikit-learn gibi birkaç çerçeve / kütüphane kullanılmıştır.

Keras

Keras, Python programlama dilinde yazılmış yüksek seviyeli bir açık kaynaklı derin öğrenme Application Programming Interface (API)' sidir. Keras [64], öncelikle sinir ağı modellemesi için tasarlanmış bir makine öğrenmesi kütüphanesidir. Aynı zamanda açık kaynaklıdır ve son yıllarda popüler hale gelmiştir. Ayrıca, kütüphane Tensorflow ve Theano gibi birden fazla makine öğrenimi platformunun üzerinde çalıştırılabilir ve sinir ağı modelleri geliştirirken karmaşıklığı azaltır. Minimalist, ölçeklenebilir ve derin sinir ağlarıyla hızlı çalışmaları destekler. Keras, standart derin öğrenme yapı taşlarının çoğu için kullanıma hazır çözümler sunar [64].

Çalışmamızda Keras, basit bir API ile hızlı tasarlandığı için seçildi. Sinir ağları kavramlarına çok uygun olduğu ve sıralı modeller gibi bazı güçlü modellerin yardımıyla basit veya karmaşık sinir ağlarının oluşturulması kısa sürdüğü için bu araştırma için Keras kütüphanesi seçilmiştir.

Matplotlib

Matplotlib, verilerin görselleştirilmesi için kullanılan açık kaynaklı bir Python kütüphanesidir. Geniş bir yelpazede farklı grafikler sağlar [65].

Scikit-Learn

Scikit-learn, veri analizi için araçlar sağlayan açık kaynaklı bir Python kütüphanesidir. Sınıflandırma, regresyon, kümeleme veya boyut azaltma gibi ön işleme ve genel makine öğrenmesi problemleri için kullanılır. NumPy ve Matplotlib üzerine inşa edilmiştir [66].

Pandas

Pandas, açık kaynaklı, veri analizi için kullanılan bir Python kütüphanesidir [67]. Veri düzenlemeleri için iyi bir arayüz sağlar. Kütüphane, özellikle zaman serisi analizi için iyi bir şekilde tasarlanmıştır. Zaman serisi verileri için Pandas, zaman adımlarını otomatik olarak çıkarabilir ve ayrıştırabilir [67].

Ayrıca, yapılandırılmış verilerden, tablo şeklindeki veri setlerinden ve vektörleştirilmiş işlemlerden yararlanır ve verileri Pandas veri çerçeveleri veya Pandas serilerinde temsil eder. DataFrame sınıfı, tablo şeklindeki verileri temsil etmek, veri ön işleme, değiştirme veya dilimlemeye yardımcı olmak için kullanılan bir yöntemdir [68]. Bu özelliklerin çoğu, özellikle veri analizi için bu tezde yoğun bir şekilde kullanılmaktadır. Cerrahi verilerin analizinin karmaşıklığını azaltılması, açıklayıcı istatistiklerin ve görselleştirmelerin oluşturulmasını kolaylaştırır. Örneğin bu çalışmada zaman serilerini Pandas kütüphanesi ile verimli bir şekilde bir araya getirip, filtreleyip, yeniden örnekleyebildik.

PyTorch

PyTorch, grafik işleme birimlerinin gücünü kullanarak hızlandırma desteğine sahip tensör hesaplamaları olan açık kaynaklı Python tabanlı bir derin öğrenme kütüphanedir [69]. Kullanımı kolay API sunar. Dinamik hesaplama grafikleri sunan bir platform sağlayarak çalışma sırasında yapılan değişiklikleri desteklemektedir [69]. Bir sinir ağı modeli oluşturmak için ne kadar bellek gerekeceği hakkında fikir olmadığında oldukça kullanışlıdır. PyTorch bu çalışmada Transformer Modelini uygulamak için kullanıldı.

3.3.2. Google colaboratory

Yaygın olarak Google Colab olarak bilinen Google Colaboratory, Google tarafından Gmail hesabı olan herhangi bir kişiye sağlanan açık kaynaklı bir hizmettir. Makine öğrenmesi eğitimi ve araştırmasının yaygınlaştırılmasına yardımcı olmak için oluşturulmuştur. Herhangi bir kurulum gerektirmeden kullanılabilen ve tamamen bulut üzerinde çalışan bir uygulamadır ve Jupyter notebook ile geliştirme ortamı sağlamaktadır [70].

Google Colab, yeterli hesaplama kaynağına sahip olmayan kişilere araştırma için Grafik İşleme Birimi (Graphic Proceesing Unit - GPU) sağlar. Google Colab hizmeti, bir çalışma zamanında 12,72 GB RAM ve 358,27 GB sabit disk alanı sağlar [71]. Her çalışma süresi 12 saat sürer, ardından çalışma zamanı sıfırlanır ve kullanıcının yeniden bağlantı kurması gerekir. Bu, insanların GPU hizmetini kripto para madenciliği ve diğer yasadışı amaçlar için kullanmamasını sağlamak içindir. Aşağıda farklı çalışma durumları gösterilmektedir [71]. Kullanıcı bir Google Colab dosyasını açtığında, bir çalışma zamanı türü seçmesi gerekir. 3 seçenek vardır: None (kullanıcının kullandığı bilgisayarın Merkezi İşlem Birimi (Central Processing Unit – CPU)' sunu kullanır), GPU, Tensor İşleme Birimi (Tensor Processing Unit -TPU) (özellikle tensör işleme için).

4. YÖNTEM

Bu çalışmada bir tür zaman serisi olan kinematik veri ile cerrahi hareketlerin tanınması için makine öğrenmesi kullanılmıştır. Öncelikle kinematik veri ile cerrahi hareketler modellenmiş ve eğitilen model sınıflandırma için kullanılmıştır. Zaman serisi sınıflandırması, zaman serisi verileri ve bu verilerin ait olduğu sınıflar arasındaki ilişkiyi denetimli olarak öğrenilmesi ve öğrenilen modelin yeni gözlemleri sınıflandırmada kullanılmasıdır. Bu amaca yönelik, bu çalışmada iki farklı derin öğrenme modeli kullanılmıştır: LSTM ve Transformer modeli. Bu bölümde JIGSAWS veriseti kinematik verilerinin zaman serisi verisi olarak ele alınması, verilerin öğrenme ve test aşamaları için ayrılmasından, her iki derin öğrenme modelinin temellerinden, modellerin eğitimi ve yeni gözlemler ile test edilmesinden, test aşamasında öğrenilen modellerin performanslarını ölçmek için kullanılan metriklerden bahsedilmiştir.

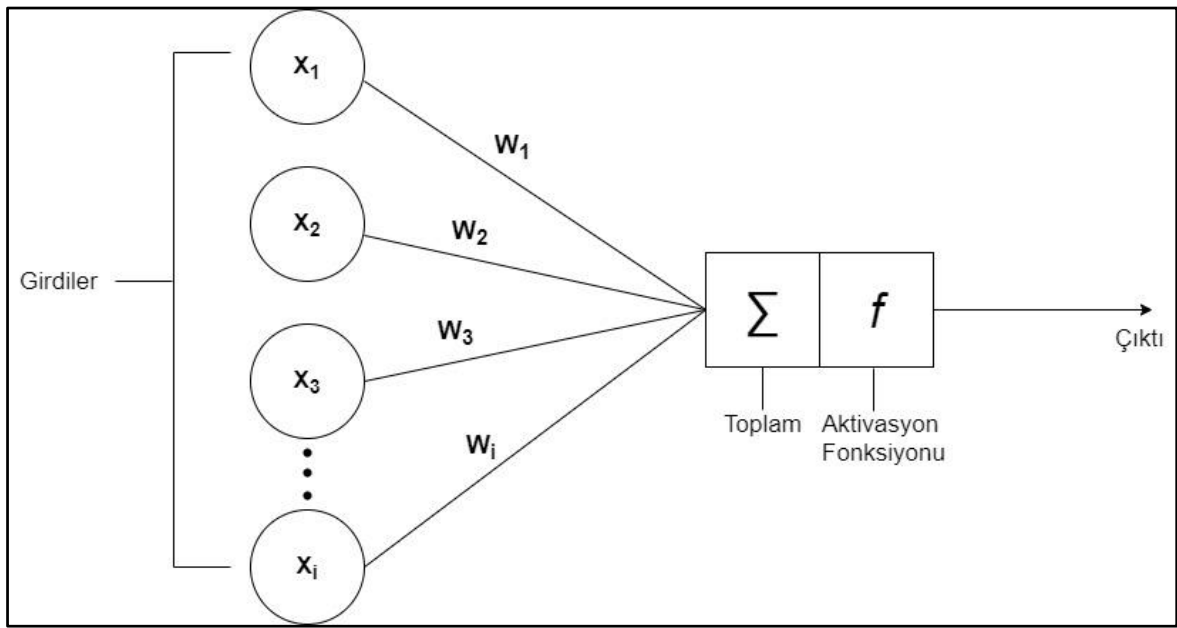
Makine öğrenmesi temel olarak sınıflandırma, kümeleme yöntemleri gibi tahmine dayalı yöntemlerden oluşur. Makine öğrenmesi, denetimli öğrenme ve denetimsiz öğrenme yöntemleri olmak üzere 2 yöntemden oluşmaktadır. Sınıflandırma ve regresyon analizi, denetimli öğrenme; kümeleme modelleri, denetimsiz öğrenme yöntemleridir. Eğer tahmin edilecek değişken kategorik veri tipindeyse sınıflandırma modeli, eğer sürekli değişken veri tipine sahipse de regresyon modeli uygulanır. Bizim çalışmamızda cerrahi hareketler kategorik olarak (G0, G1, G2, ...) tahmin edilmektedir. Bu nedenle cerrahi hareket tanıma için sınıflandırma yapılmıştır.

4.1. Derin Öğrenme

Derin öğrenme, son yıllarda, duygu analizi, metin özetleme [72] ve soru-cevap [73] gibi doğal dil işleme görevleri için en yaygın kullanılan yaklaşım olmuştur. Görüntü, metin ve ses verilerinde doğrusal olmayan gözlemler yapmak ve çıkarımlarda bulunmak için tercih edilen bir yöntemdir. Artan hesaplama altyapısı ve büyük miktarda veriye erişimin mümkün olması ile son yıllarda bu alanda büyük ilerlemeler görülmüştür. Büyük etiketli veri setlerinin kullanılabilirliği, bilgisayar mühendisliğinde yapılan ilerlemeler, dağıtılmış sistemler ve GPU' lar dahil hesaplama gücü gibi birçok faktör sayesinde başarı göstermektedir.

4.2. Yapay Sinir Ağları

Derin öğrenme modelleri çok katmanlı sinir ağlarından oluşur. Yapay sinir ağları, veri odaklı bir tekniktir ve son yıllarda mevcut verilerdeki artış ve bilgisayar donanımındaki optimizasyonlar sayesinde popüler hale gelmiştir. Bu ağlar görüntü sınıflandırma, metin oluşturma, örüntü tanıma, zaman serisi tahmini, nesne algılama gibi çok sayıda problemi çözmek için kullanılmaktadır. Temeli biyolojik sinir ağlarına dayanarak insan beyninin çalışmasını ilham alan denetimli bir öğrenme yöntemidir [74].



Şekil 4.1. Yapay sinir ağı modelinin temeli [74]

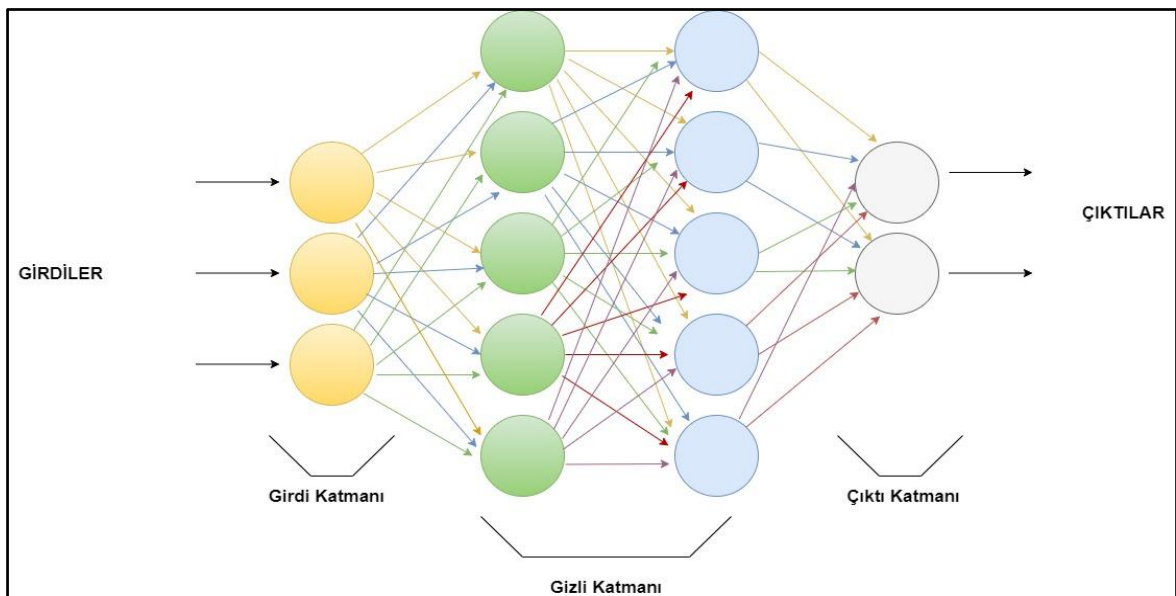
Bir nörona gelen tüm girdiler alınarak, girdilere karşılık gelen ağırlıklarıyla çarpılır ve ardından ağırlıklar toplanarak, aktivasyon fonksiyonu adı verilen doğrusal olmayan bir fonksiyon uygulanır ve bir çıktı üretilir. Bir nöronun çalışması Şekil 4.1’ de görülmektedir. Matematiksel olarak vektör Eş. 4.1 ile temsil edilmektedir. Denklemden sırasıyla x , ω , b , $\odot$, f , y giriş vektörünü, ağırlık vektörünü, nöron eğilimini, Hadamard Çarpımını (eleman – eleman matris çarpımı), aktivasyon fonksiyonunu ve nöron çıkışını temsil eder.

$$y(x) = f(\omega \odot x + b) \quad (4.1)$$

4.2.1. Derin sinir ağı

Derin sinir ağı, birden çok gizli katmanın bir araya getirildiği sinir ağıdır. Her katman, önceki katmanın çıktısını alan doğrusal olmayan bir modüldür. Derin sinir ağı, doğrusal olmayan görevlerde kullanılmak üzere eğitilir. Genellikle regresyon veya sınıflandırma sorunlarını çözmek için eğitilir. Bizim problemimizde bir sınıflandırma problemi olduğu için derin sinir ağından yararlanılmıştır.

Basit bir ileri beslemeli ağı, Şekil 4.2’ de görülmektedir. Bir derin sinir ağı, her biri N sayıda nörona sahip olan çok sayıda katmana sahiptir [75]. Her katmandaki nöronlar, bir dizi yönlendirilmiş kenar ile önceki katmandaki tüm nöronlara bağlanır. Her kenar, kendisine karşılık gelen bir ağırlığa sahiptir. Üç tür katman vardır: giriş katmanı, gizli katman ve çıktı katmanı. İlk katman olan giriş katmanı, girişi alır. Çıkış katmanı olarak adlandırılan son katman, sinir ağının çıktısını üretir. Arada kalan katmanlar gizli katmanlardır. Bilgi, girdi katmanından çıktı katmanına doğru ilerler. Ağı çıktısı en üst katmanda üretilinceye kadar alt katmandan yukarı doğru hareket ederken her bir nöronun çıktıları hesaplanır. Çok katmanlı sinir ağı, tüm nöronların önceki katmanının her nöronuna bağlı olduğu temel bir derin sinir ağıdır. Her bir nöronun çıktısının önceki katmanlardaki nöronlardan toplam ağırlıklı bir girdi aldığı ve bunu bir aktivasyon fonksiyonu aracılığıyla beslediği katmanlar halinde organize edilmiş birçok nörondan oluşur [75].



Şekil 4.2. Girdi, gizli ve çıktı katmanından oluşan derin sinir ağı mimarisi

Algılayıcı

Sinir ağlarının temel birimi, algılayıcı adı verilen yapay nöron türlerinden biridir. Bir algılayıcı sabit sayıda girdi alır ve tek bir çıktı üretir. Çıktıyı hesaplamanın yolu; girdi alıp, ağırlıklarını ve bias terimini ekleyerek ağırlıklı toplamı hesaplama ve aktivasyon fonksiyonunu uygulama gibi birkaç adıma sahiptir. Bu süreç Resim 4.3' te görülmektedir. Algılayıcı, bir giriş vektörü ($x = (x_1, x_2, \dots, x_n)$) olarak temsil edilen N girişe sahiptir. Her giriş, ağırlık vektörü ($w = (w_1, w_2, \dots, w_n)$) ile tanımlanan atanmış bir ağırlığa sahiptir. Sonuç olarak, ağırlıklı girdi değerleri birleştirilerek ağırlıklı toplamı Eş. 4.2' deki gibi hesaplanır [75]:

$$\varepsilon = \omega \cdot x = \sum_{i=1}^n \omega_i \cdot x_i \quad (4.2)$$

Bu adımda, çıktıyı hesaplamak için aktivasyon fonksiyonu (σ) (Bkz. Bölüm 4.3.4.) ağırlıklı toplama uygulanır ve ağırlıklı toplam, eşiği aşp aşmamasına bağlı olarak 0 veya 1 olan bir çıktı y üretmek için bir Eş. 4.3' teki gibi eşikle karşılaştırılır:

$$y = \sigma(\varepsilon) = \begin{cases} 1, & \varepsilon \geq \theta \\ 0, & \varepsilon < \theta \end{cases} \quad (4.3)$$

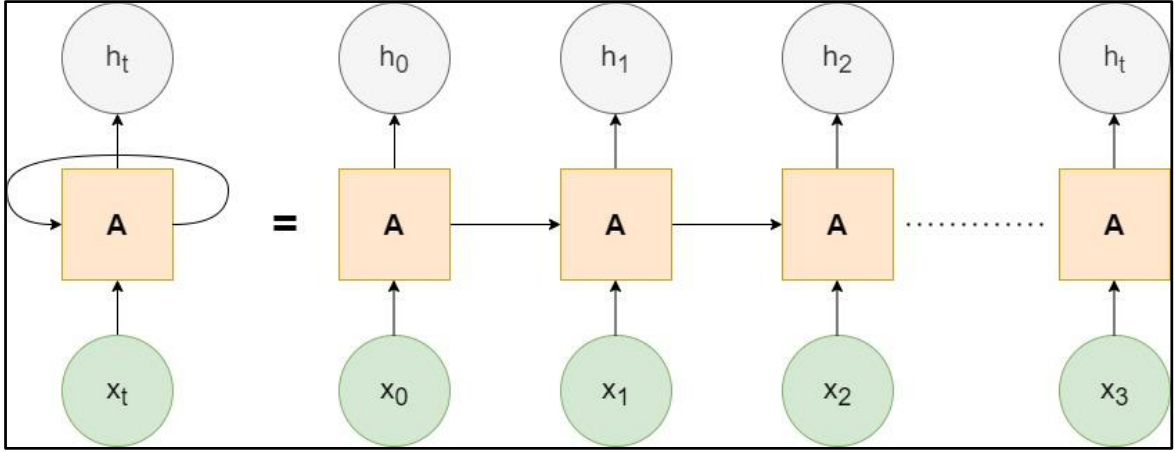
İleri ve geri yayılım

Bir ağı eğitmek için öncelikle toplam hatanın hesaplanması gerekir. İleri yayılım ve geri yayılım olmak üzere iki adımı bulunan algoritma, ağırlıkların rastgele başlatılmasından sonra başlamaktadır. Geri yayılım algoritması, sinir ağlarını eğitmek için kullanılan denetimli bir öğrenme algoritmasıdır. Algoritmanın amacı, ağırlıkları optimize etmektir. Böylece sinir ağı, tahmin edilen ve doğru hedef değerler arasındaki hata oranını en aza indirerek ulaşılabilen daha doğru tahminler elde eder. İleri yayılım süreci, algılayıcının çıktısının hesaplanmasıyla başlar ve kayıp fonksiyonunu bularak devam eder. Daha sonra geriye doğru yayılım sürecinde elde edilen kayıp fonksiyonu (Bkz. Bölüm 4.3.5) türevler kullanılarak en aza indirilir ve daha sonra ağırlıklar güncellenir [75, 76].

4.2.2. Tekrarlayan sinir ağıları (Recurrent Neural Network - RNN)

Tekrarlayan sinir ağıları, 1980' lerde önerilen vektör dizileri üzerinde işlem yapılmasına izin veren yapay sinir ağı modelleridir. Metin çevirisinde çoklu girdi çoklu çıktıya izin vermesi, video sınıflandırmasında ardışık video çerçevelerini girdi olarak alıp tekli çıktı olarak tahmin edebilmesinden dolayı bu gibi alanlarda uygulanmaktadır. Liu ve diğerleri [77] tarafından yapılan araştırmaya göre, tekrarlayan sinir ağlarının geliştirilmesinin ana nedeni sıralı bilgileri işlemektir. Standart sinir ağları ve tekrarlayan sinir ağları arasındaki önemli fark, tekrarlayan sinir ağlarının önceki girişlerden hesaplanan bilgileri depoladıkları bir belleğe sahip olmasıdır, yani son çıktı yalnızca önceki giriş tarafından değil, beslenen tüm girişler tarafından da kullanılır. Ayrıca, tekrarlayan sinir ağları, zaman boyutu üzerindeki bilgileri yakalamak ve bunları ağ içinde depolamak için etkili bir yöntemdir. Zaman serisi verilerinde sıralı model bulunduğundan, standart sinir ağı zaman serisi tahmini için verimli sonuçlar vermemektedir. Tekrarlayan sinir ağları, veriler arasındaki bağımlılığı öğrenmek için daha etkilidir. Tekrarlayan sinir ağlarının zaman serisi tahmini için önemli sonuçlar verdiği kanıtlanmıştır [78]. Bu nedenle bu ağın tekrarlayan kısmı, önceki hesaplamalara bağlı çıktı ile bir dizinin her ögesi için aynı görevi gerçekleştirme kabiliyetinin bir sonucudur. Teorik olarak, tekrarlayan sinir ağlarının çalışma yolları, önceden işlenmiş bilgilerin sınırsız geçmişini depolamak için belleğe sahip olan tekrarlayan sinir ağları olarak gösterilebilir.

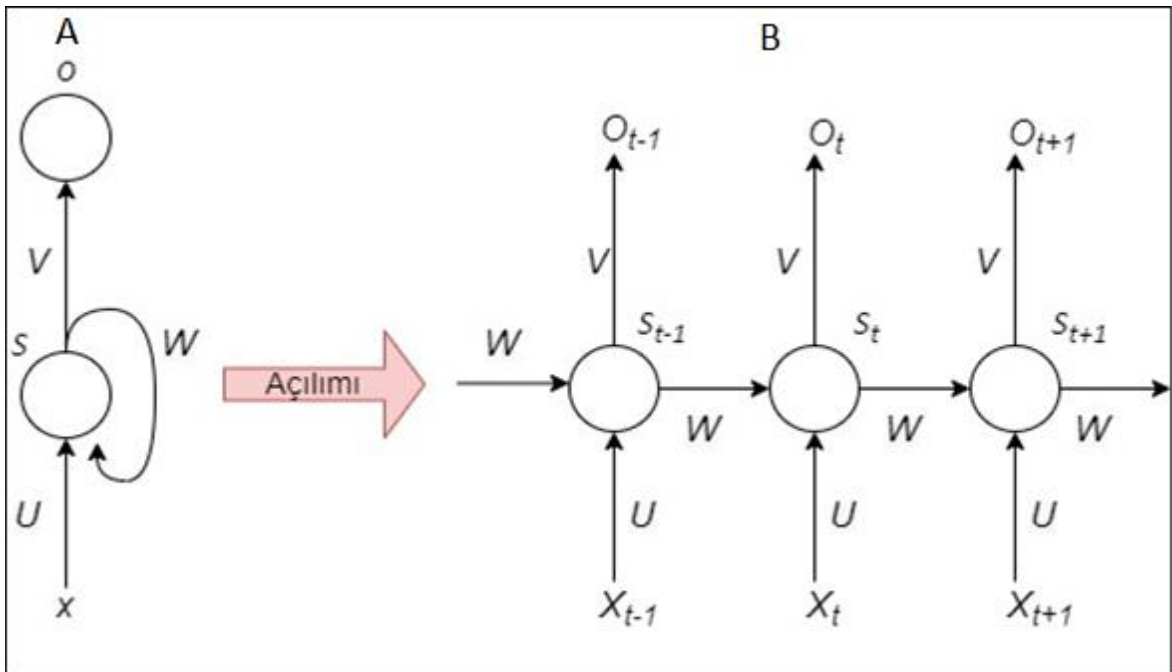
Standart sinir ağlarının bir girişi ve bir çıkışı vardır. Ancak bu pek çok uygulama için doğru değildir. Örneğin, bir cümledeki bir kelimeyi tahmin etmek için, kendisinden önce gelen kelimeleri bilmek zorundadır. Bunun yerine, tekrarlayan bir sinir ağı, iki ağ arasında bazı verileri korurken, ağı yeni veriler üzerinde yeniden çalıştırarak birden çok girdiye ve birden çok çıktıya sahip olabilir. Nöronların ağırlıkları aynı kalır, ancak aktivasyon fonksiyonları farklıdır. Giriş verileri farklıdır ve durum bilgisi tutulur. Bu ağlar, keyfi bir süre önce meydana gelen verileri depolayabilir ve bunlar üzerinde işlem yapabilir. Bu şekilde, tek bir tekrarlayan sinir ağı katmanını iki girişe ve iki çıkışa sahip olur: veri ve zaman. Zaman ve zaman adımları, ağın tek bir tekrarına atıfta bulunur [75].



Şekil 4.3. Basit bir tekrarlayan sinir ağı [76]

Basit bir tekrarlayan sinir ağı, kalıcı bilgilerin ağın bir adımından diğerine aktarılmasına izin veren döngüleri olan bir ağıdır. Bu döngü Şekil 4.3' te görülmektedir. Burada ağ, her bir zaman adımı için bir x_t girişine ve bir h_t çıkışına sahiptir. A gizli durumdur. Gizli durumda, girdisini önceki adımın gizli durumundan ve mevcut adım x_t ' nin çıkışından alan bir aktivasyon fonksiyonu bulunur (Eş. 4.4). Tekrarlayan ağın ağırlıkları zaman adımlarında aynı kalır, ancak dahili bellek zamanla değiştiğçe ve giriş zamanla farklılık gösterdiğinden etkinleştirmeler farklılık gösterir [75].

$$A_t = f(A_{t-1}, x_t) \quad (4.4)$$



Şekil 4.4. Tekrarlayan sinir ağının katlanmış ve katlanmamış durumu [75]

Şekil 4.4' te, Bölüm A, tekrarlayan sinir ağının zaman içinde açılmadan önceki durumunu göstermektedir. Bölüm B, tekrarlayan sinir ağının zaman içindeki katlanmamış sonraki durumunu gösterir. Tekrarlayan sinir ağının katlanmamış durumu, basitçe sıralı ağ gösterir. B bölümüne bakıldığında görüldüğü üzere buradaki sinir ağı, üç katmanlı bir sinir ağıdır. Açıldıktan sonra basit bir geri bildirim bağlantısına sahip bir ağ, her zaman adımında yeni bir katman eklenmesi ile ileri beslemeli sinir ağına dönüştürülebilmektedir. Buradaki ağ, birden fazla gizli katmana sahip olduğu için derin öğrenme sinir ağı olarak adlandırılabilir. Tekrarlayan sinir ağının hesaplamalarında kullanılan çeşitli parametrelerin U, V ve W nöronların ağırlığını temsil eder. x_t , s_t ve o_t , sırasıyla adım t zamanındaki girdi, gizli durum ve çıktı durumudur. Tekrarlayan sinir ağı modellerine karşılık gelen ileri yayılma formülleri Eş. 4.5' teki gibi tanımlanabilir [75]:

$$s_t = f(Ux_t + Ws_{t-1} + b_s) \text{ ve } o_t = g(Vs_t + b_o) \quad (4.5)$$

b_s ve b_o bias terimleridir. f ve g , f' nin genellikle doğrusal olmayan, gizli nöronlara uygulanan türevlenebilir olduğu ve g' nin belirli bir göreve bağlı olarak seçilen aktivasyon fonksiyonlarıdır.

Öğrenme zorluğu

Standart sinir ağları zamansal bağımlılıkları dikkate almadığından, tekrarlayan sinir ağlarının sıralı problemleri çözmek için tasarlanmış bir mimari olduğundan daha önce bahsedilmişti. Tekrarlayan sinir ağları, eğitim sırasında ağırlıkları optimize etmek için zaman içinde geri yayılım yöntemini kullanır. Bu yöntem, en son zaman adımından önceki adımlara geri dönmek için zincir kuralını kullanır. Sonuç olarak, diziler uzun olduğunda, geri yayılım süreci çok uzun olacak ve hesaplaması çok maliyetli olacaktır. Teoride, tekrarlayan sinir ağlarının keyfi olarak uzun dizilerdeki bilgileri kullanma konusunda herhangi bir sınırı yoktur, ancak pratik olarak standart tekrarlayan sinir ağları, kaybolan gradyan problemi nedeniyle yalnızca birkaç adıma bakabilir [79]. Tekrarlayan sinir ağlarından her zaman adımı için ağırlıkların gradyanlarının toplanması gerekmektedir [80]. Gradyanlar ağda geriye doğru giderken küçülme eğilimindedir. Bu durum tekrarlayan sinir ağlarındaki yaygın bir sorun olan ağların daha uzun zaman aralıklarında geçici bağımlılıkları yakalamasını zorlaştıran, ortadan kaybolan gradyan problemine sebep olmaktadır [81, 82]. Bu nedenle tekrarlayan sinir ağlarında kaybolan bir gradyan problemi

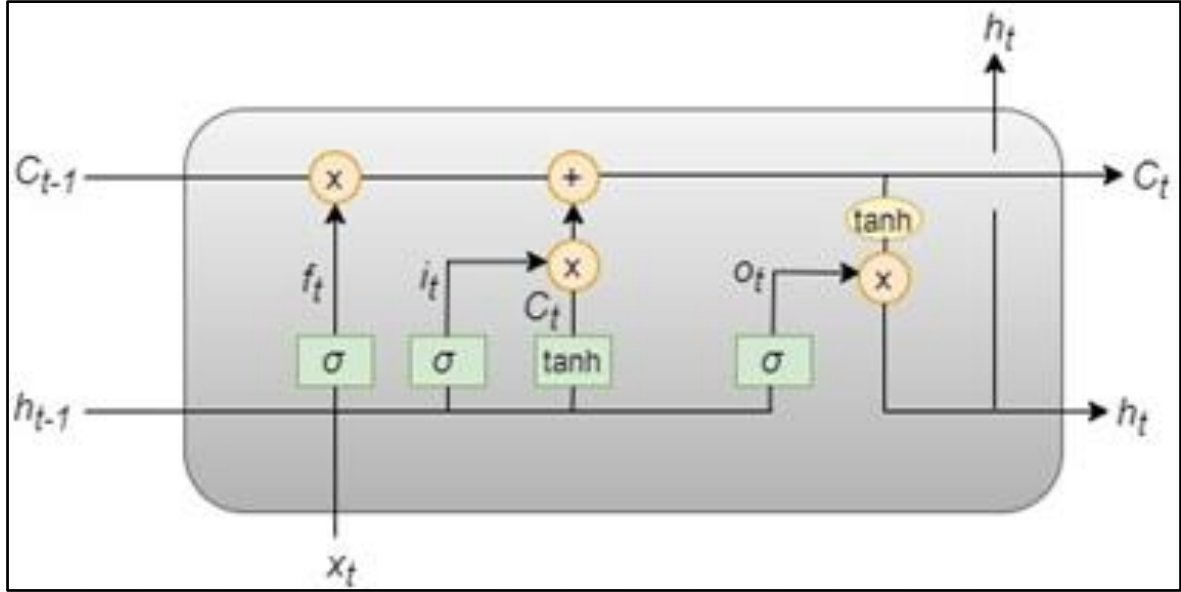
vardır ve uzun vadeli bağımlılıklarda öğrenme sorununa yol açar. Problemin üstesinden gelmek için alternatif yaklaşımlar önerilmiştir: LSTM ve GRU [82, 83].

4.2.3. Gated recurrent units

GRU, 1997' de tanıtılan LSTM birimlerine kıyasla nispeten yeni bir yöntemdir ve oldukça benzerdir [84]. Başlangıçta Cho ve diğerleri [85] ve ayrıca son yıllarda en az LSTM tabanlı ağlar kadar iyi olduklarını göstermişlerdir. GRU mimarisi etrafındaki popülerlik, öncelikle daha az karmaşık yapıları nedeniyle. Ünitenin genel karmaşıklığını azaltarak LSTM' den farklıdır. Genel olarak, hücre durumundan çıkış akışını kontrol eden bir çıkış geçidi içermez. Bunu yaparak, optimize edilebilir parametrelerin toplam sayısı azaltılır ve dolayısıyla model karmaşıklığı azaltılır.

4.2.4. Long short- term memory

Tekrarlayan sinir ağlarının bir çeşidi olarak, LSTM ağı ilk olarak Hochreiter ve Schmidhuber [86] tarafından insan beyninin sıralı verileri işleme şeklini taklit etmekten yola çıkarak önerilmiştir [87]. LSTM ağları, son teknoloji performansla makine çevirisi, konuşma tanıma [88] gibi farklı modelleme görevlerine uygulanmıştır. Tekrarlayan sinir ağlarının uzun vadeli bağımlılıkları ile ilgili sorunları varken, LSTM bu sorunu ek özelliklerle çözer. Uzun süreli bağımlılık probleminden kaçınmak için tasarlanmış özel bir tekrarlayan sinir ağı türüdür. Standart tekrarlayan sinir ağlarının aksine, LSTM modeli bilgileri uzun süre hatırlama özelliğine sahiptir [89]. Bu da gradyanların ağıdaki her katman boyunca yayılmasına izin verir. LSTM ağı, hücre durumunun zaman içindeki verilerini biriktirmektedir. Hatanın türevlerini zaman içinde topladıklarından, hızla kaybolmazlar.



Şekil 4.5. LSTM mimarisinde hücre yapısı [90]

LSTM, bir hafıza hücresinden oluşur ve yapısı Şekil 4.5' te gösterilmektedir. Her hafıza hücresi, hücreye giren ve çıkan bilgi akışını kontrol eden kapılar içerir. Hafıza bloğunun üç ana bileşeni vardır: giriş kapısı, unutma kapısı ve çıkış kapısı. Kapıların her birinin kendi sorumluluğu vardır. Adlarından da anlaşılacağı gibi,

- Giriş kapısı, hücrede hangi bilgilerin depolanacağına karar verir, hangi değerlerin güncelleneceğine karar verir,
- Unutma kapısı, önceki gizli durumdan hangi bilgilerin ağa iletilmesi gerektiğine karar verir, önceki hücre durumunun ne kadarının unutulacağına karar verir,
- Çıkış kapısı, yeni hesaplanmış gizli durumun hangi bilgisinin ağın çıkış vektörüne gittiğini kontrol eder ve mevcut hücre durumunun ne kadarının mevcut gizli duruma ekleneceğine karar verir.

LSTM- RNN

Önceki kısımda, tekrarlayan sinir ağlarının geri yayılım yöntemi ile eğitilmesiyle ilişkili kaybolan gradyan sorununu çözmek için LSTM kullanıldığından bahsedildi. LSTM, bu geriye yayılım problemini, tekrarlayan sinir ağları sorunları ve etkilerini uzun zaman adımları boyunca öğrenmesine izin vermek için sabit bir hatanın sürdürülmesini sağlayarak çözebilir. LSTM, kapılı hücresi sayesinde bunu başarabilmektedir. Bu kapılı hücrenin, açık ve kapalı olmak üzere iki durumu vardır. Bu durumlar da onun bir bilgisayarın hafızası

gibi davranmasını sağlar; yani kendisine hangi verilerin yazılmasına, ondan okunmasına ve saklanmasına izin verileceği konusunda kararlar verir. Bu özellik, eğitim sürecinden öğrenilen cerrahi hareketlerin ayrıntılarını tutmasına ve kapılı hücrede depolanan bu bilgilere dayanarak sınıflandırma için kararları vermesine olanak tanır.

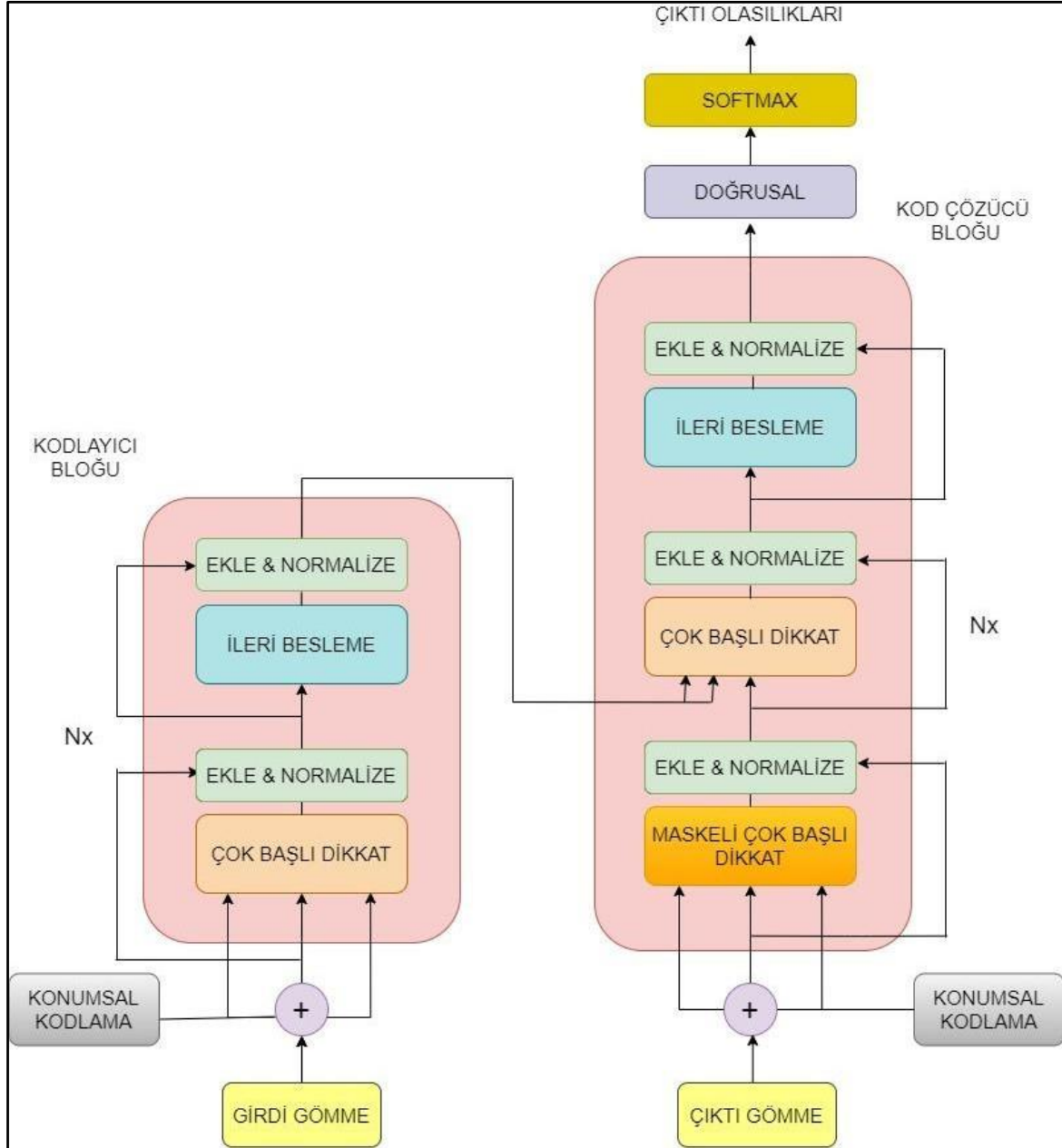
4.2.5. Transformer network

Zaman serisi tahmini, insanların kaynakları yönetmelerine ve kararlar almalarına yardımcı olmak için günlük yaşamda önemli bir rol oynar. Zaman serisi verileri birçok bilimsel ve mühendislik disiplinde yaygındır. Zaman serisi tahmini, zaman serisi verilerini modellemede çok önemli bir görevdir ve makine öğreniminin önemli bir alanıdır. Bu çalışmada, zaman serisi verilerini sınıflandırmak için Transformer tabanlı makine öğrenme modellerini kullanarak geliştirilen modeli kullandık. Bu yaklaşım, zaman serisi verilerinden karmaşık kalıpları ve dinamikleri öğrenmek için self-attention mekanizmalarından yararlanarak çalışır.

Transformer Ağları, 2017’ de Attention is All You Need isimli makalenin yayınlanmasıyla tanıtılmıştır [14]. Öncelikle doğal dil işleme (Natural Language Processing - NLP) alanında çeviri, metin özetleme gibi görevler için tasarlanan bir derin öğrenme modeli olarak önerilmiştir. Transformer ağları, LSTM gibi tekrarlayan sinir ağı modellerinin yerini alarak, NLP problemleri için hızla tercih edilen bir model haline gelmiştir [91]. Transformer ismi, bir transdüksiyon modelidir. Transdüksiyon, NLP bağlamında, bir diziyi girdi olarak alıp, başka bir diziyeye çıktı olarak veren modellerdir [92]. Bu durumda, bir dizinin diğer bir diziyeye dönüşmesi yani transformasyon söz konusu olduğu için ismi buradan gelmektedir. Transformer’ ın ana fikri, transdüksiyon modellerinin kalbi olan tekrarlayan (RNN, LSTM, GRU) veya CNN ağlarını kullanmayarak bunu tamamen dikkat mekanizması ile değiştirmektir [14].

Transformer modeli, tekrarlayan sinir ağı yapısı olmadan dikkat mekanizmasını temel olarak oluşturulmuştur. Dikkat mekanizmalarının tek başına, peş peşe hesaplamalara ihtiyaç duymadan tekrarlayan sinir ağlarının performansının güçlü olduğu vurgulanmaktadır [91]. Transformer modelinde Şekil 4.6’ da görüldüğü gibi biri solda biri sağda olmak üzere 2 temel blok vardır. Kodlayıcı bloğu, girdiyi özyinelemeli olarak işleyen, girdiyi vektöre eşleyen ve içindeki fonksiyon ile kodlayan, bir dizi kodlama

katmanından oluşmaktadır [14]. Her kodlayıcı katmanı, girdilerin hangi bölümlerinin birbiriyle ilişkili olduğu hakkında bilgi içeren kodlamalar oluşturmak için girdiyi işlemektedir. Bu katmanın çıktıları daha sonra ileri beslemeli sinir ağı katmanına girdi olarak gelir ve her konuma bağımsız olarak uygulandıktan sonra, sinir ağını besler.



Şekil 4.6. Transformer modeli mimarisi [14]

Kod çözücü bloğu, kodlayıcı bloğunun çıkışını girdi olarak alır ve içindeki fonksiyonlarla belirtilen boyutlara dönüştürür. Kod çözücü bloğunda kodlayıcı bloğundan farklı olarak bir dikkat mekanizması bulunmaktadır. Her iki blok da, çıktıların işleminden geçirilmesi için bir

sinir ağına sahiptir. Bu iki bloğun içindeki sinir ağı LSTM ya da dikkat mekanizması gibi herhangi ileri beslemeli bir ağıdır [14].

Transformer modelinin ortaya çıkmasında en etkili kısım dikkat mekanizmasıdır. Dikkat mekanizması, girdi dizisine bakar ve dizinin her adımda önemini koruyan bölümlerini seçerek, dizinin hangi bölümlerinin önemli olduğu bilgisini korur. Dikkat mekanizması, anlık olarak birkaç diğer girdi verisini de hesaba katar ve bu girdilere farklı ağırlıklar atayarak önceliklendirme sağlar. Tüm kodlayıcı ve kod çözücü katmanları, her bir girdi için diğer tüm girdilerin uygunluğunu ölçen ve çıktıyı üretmek için uygun bilgileri alan bir dikkat mekanizması kullanır. Ardından dikkat mekanizması sonucunda çıktı olarak gönderilen ağırlıkları ve kodlanmış şekilde olan diziyi girdi olarak alır. Bu ağırlar, tekrarlanan çok başlı dikkat bloklarından ve ileri beslemeli katmanlardan oluşur. Çok başlı dikkat, dikkat mekanizmasındaki işlemleri paralel olarak çalıştırır ve sonuçları birleştirir. Böylece, farklı ilişkilerin öğrenilmesi sağlanmış olur [14]. Girdiye ait özelliklerden doğrusal dönüşümler ile sorgu (Q), anahtar (K) ve değer (V) matrisi Eş. 4.6 - 4.8' deki gibi oluşturulur [93]:

$$Q = XW^Q \quad (4.6)$$

$$K = XW^K \quad (4.7)$$

$$V = XW^V \quad (4.8)$$

$X \in \mathbb{R}^{T \times N}$ (T: girdi uzunluğu, N: özellik sayısı), W^Q , W^K ve W^V eğitilebilir ağırlık matrisleridir. Daha sonra, ölçeklendirilmiş skaler çarpım dikkati uygulanır. Ardından hesaplanan sorgu, anahtar matrisleri Softmax fonksiyonunda geçirildikten sonra değer matrisi ile çarpılır ve dikkat hesaplanır (Eş. 4.9):

$$Dikkat(Q, K, V) = softmax\left(\frac{Q^T K}{\sqrt{d_k}}\right)V \quad (4.9)$$

Girdi özelliklerinin boyutsallığı ile payda, Softmax' ın büyük gömmeler için skaler çarpımı ölçeklendirmesi sağlanır [93]. Çok başlı dikkat, girdilere paralel olarak birkaç kez dikkat uygular. Bir katmandaki tüm başların çıktıları birleştirilir, normalize edilir ve bir sonraki katman için girdi olarak ileri besleme ağı tarafından işlenir.

Transformer – RNN farkı

Transformer modelini tekrarlayan sinir ağlarından ayıran üç faktör olduğu belirtilmiştir:

- Paralel çalışmaya uygun olması: Tekrarlayan sinir ağlarının aksine, Transformer ağları sıralı verilerin sırayla işlenmesini gerektirmez. Örneğin, LSTM kullanırken mimarinin yapısı gereği bir işlemi yapmadan önce başka bir işlemi yapmak gerekir. 2. Adımı hesaplamak için 1. Adımın gizli durumuna ihtiyaç duyulabilmekte fakat Transformer ağlarında ise işlemlerin çoğunu paralel olarak yapılabilir. Bu özellik nedeniyle Transformer ağları, çok daha fazla paralelizasyona izin vermektedir. Sıralı hesaplamanın aksine, paralel hesaplama mevcuttur [91].
- Daha hızlıdır: Aynı görevi diğer mimarilere göre daha az işlemle yapmaktadır. Paralel çalışma ile eğitim sürelerini kısaltmaktadır [91].
- Daha büyük veri setleri üzerinde çalışabilmektedir: Eğitim sırasında daha fazla paralelleştirmeyi kolaylaştırdığından, mümkün olandan daha büyük veri setleri üzerinde eğitim sağlanmaktadır [91].

4.3. Hiperparametreler

Sinir ağı modellerinin başarısını etkileyen parametreler bulunmaktadır. Hem model tanımında hem de eğitimde kullanılan birçok parametre vardır ve bu parametreler hiperparametre olarak adlandırılır. Sinir ağlarının aşırı öğrenmesini önlemek için Srivastava, Hinton, Krizhevsky, Sutskever ve Salakhutdinov [94], belirli hiperparametrelerin (öğrenme hızı, momentum, maksimum norm, vb.) modelin davranışını nasıl etkilediğini açıklamıştır. Hiperparametreler performansı iyileştirecek şekilde ayarlanmalıdır. Bu çoğunlukla, her bir değerin sırayla denendiği ve her ağın nihai performansının karşılaştırılmasıyla yapılır. Hiperparametrelerin ayarları, modelin öğrenip öğrenmediği veya tahminlerinin ne olduğu üzerinde büyük bir etkiye sahiptir. Bu parametrelerden bazıları şunları içerir:

- Öğrenme Oranı,
- Batch Boyutu,
- Epoch Sayısı,
- Aktivasyon Fonksiyonu,

- Kayıp Fonksiyonu,
- Optimizasyon Fonksiyonu,
- Dropout değeri

4.3.1. Öğrenme oranı

Ağın ağırlıklarını, kayıp gradyanına göre ne kadar ayarlandığını kontrol eden bir hiperparametredir. Parametrelerin güncellenme hızını ayarlamaktadır. Genellikle değer (0, 1) aralığındadır. Değer ne kadar düşükse ya da optimum değerlerden çok daha küçükse, aşağı doğru eğim boyunca o kadar yavaş ilerler ve ideal sonuca ulaşmak çok daha uzun sürer. Çok yüksek bir öğrenme oranı yani optimum değerden çok daha büyükse, ağın öğrenmesini hızlandırırken ideal sonucu aşacaktır.

4.3.2. Batch boyutu

Batch boyutu, modeli besleyen örneklerin sayısıdır. Batch boyutu, modelin ne kadar hızlı yakınsadığını belirler ve sonuçları kökten etkileyebilir. Büyük batch boyutu, modelin genelleştirme yeteneğinin kaybolmasına neden olabilir. Bir batchteki örnekler ne kadar farklı olursa, model o kadar hızlı öğrenir [95].

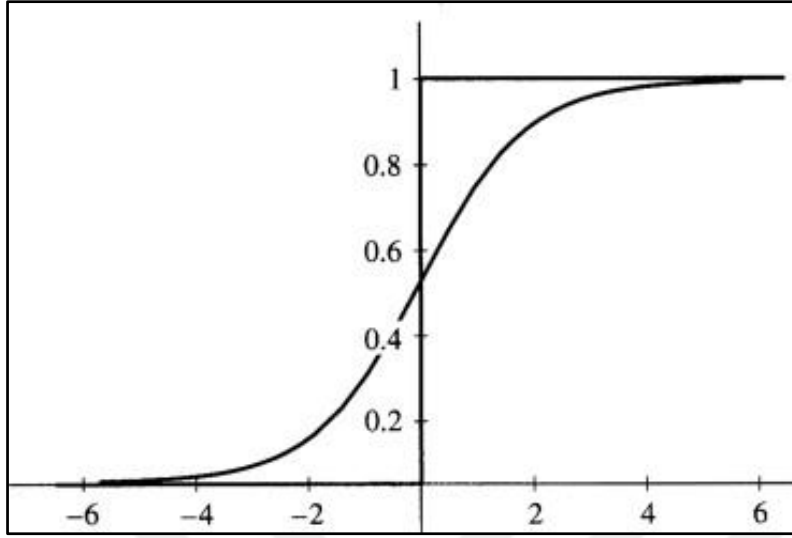
4.3.3. Epoch sayısı

Eğitim sırasında tüm eğitim verilerinin ağda gösterilme sayısıdır. Modelin veriyi iyi tutması, ancak aşırıya kaçmaması için epoch sayısını doğru vermek önemlidir. Aynı model ve hiperparametrelerle birkaç deneme yapıp, sadece epoch sayısını değiştirmek, en iyi performansı sağlayacak miktarı seçmek olabilir. Bu yaklaşımın bir dezavantajı, tüm modelleri denemek ve hesaplama verimsizliğinin uzun sürmesidir.

4.3.4. Aktivasyon fonksiyonu

Aktivasyon fonksiyonu, yapay sinir ağı modeli mimarisinde çok önemli bir rol oynar. Sinir ağları doğrusal olmayan aktivasyon fonksiyonları kullanırlar, böylece daha karmaşık yapıları ve karar sınırlarını öğrenmeleri için giriş verisine doğrusal olmayan dönüşüm gerçekleştirir [96]. Aktivasyon fonksiyonunu yaygın örnekleri arasında Sigmoid, Tanjant, Rectified Linear Unit ve Softmax bulunur [75].

Sigmoid fonksiyonu



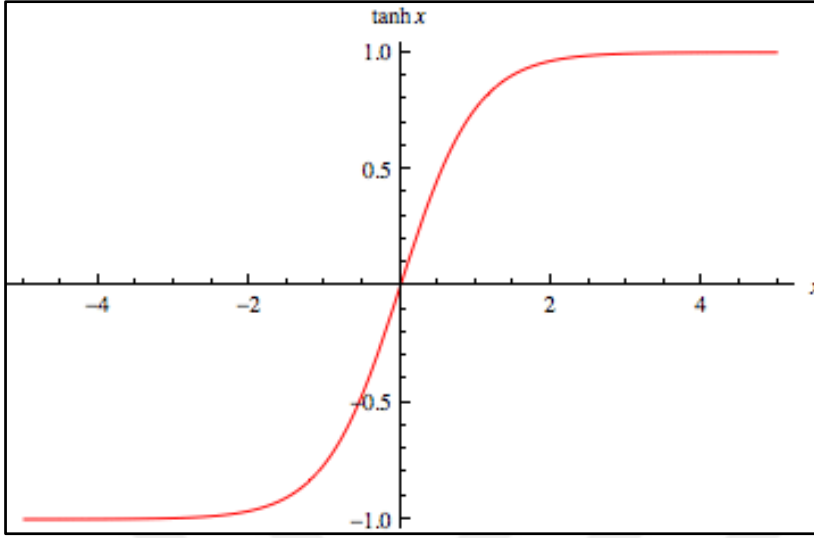
Şekil 4.7. Sigmoid fonksiyonu grafiği [97]

Sigmoid fonksiyonu, en yaygın kullanılan aktivasyon fonksiyonlarından biridir ve “Lojistik Fonksiyon” olarak da adlandırılır. Monoton olarak artan bir fonksiyondur ve Eş. 4.10’ daki gibi tanımlanır [97]:

$$\sigma(x) = \text{sigmoid}(x) = \frac{1}{1+e^{-x}} \quad (4.10)$$

Sigmoid fonksiyonu, gerçek değerli bir sayı alıp, Şekil 4.7’ de görüldüğü gibi (0,1) aralığında bir değere dönüştürür. Bununla birlikte, 0 veya 1 de iken, doygunluk noktalarına yakın gradyan değerlerinin kaybolması veya küçük olması nedeniyle Sigmoid fonksiyonunu kullanmanın ağırlık daha fazla öğrenmek için yavaşlayabilmesi gibi dezavantajları vardır.

Tanjant fonksiyonu

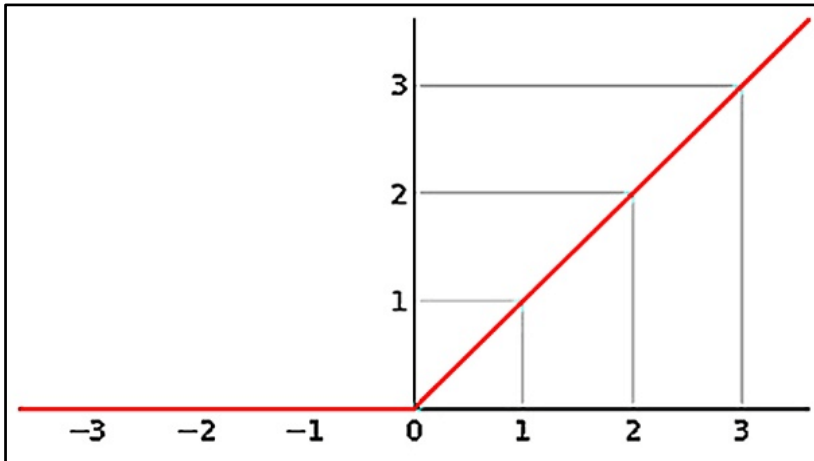


Şekil 4.8. Tanh fonksiyonu grafiği [98]

Tanjant (tanh) fonksiyonu, başka bir aktivasyon fonksiyonu türüdür. Sigmoid fonksiyonu olarak doğrusal olmayan S şekilli bir fonksiyondur. Sigmoid fonksiyonu ile aralarındaki temel fark, Tanh fonksiyonunun çıktı aralığının $[0,1]$ yerine Şekil 4.8’ de görüldüğü gibi $[-1,1]$ aralığında sıfır merkezli olmasıdır. Bu nedenle, Tanjant fonksiyonu daha çok tercih edilmektedir ve Eş. 4.11’ deki gibi tanımlanır [98]:

$$\sigma(x) = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (4.11)$$

ReLU fonksiyonu

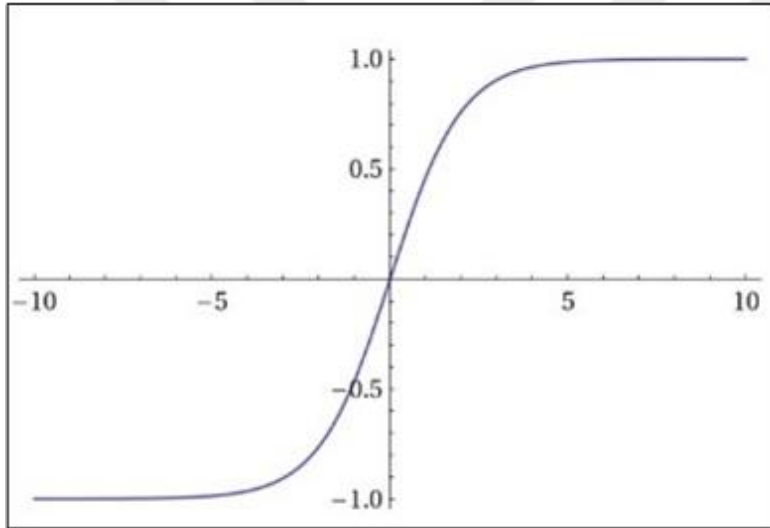


Şekil 4.9. ReLU fonksiyonu grafiği [99]

Eş. 4.12' de matematiksel olarak ifade edilen ReLU fonksiyonu, negatif giriş değerlerini sıfıra ayarlayarak net girdiyi Şekil 4.9' da görüldüğü gibi sıfırdan büyük veya sıfıra eşit bir değere sıkıştırır. Sigmoid ve Tanjant fonksiyonları ile karşılaştırıldığında, ReLU hesaplamalarında aktivasyonlarda üstel fonksiyonun hesaplanmasına gerek olmadığı için maliyeti daha azdır [99]. ReLU' yu sinir ağlarında kullanmak, yakınsaması daha hızlı olduğu için ve gradyan kaybolma problemiyle karşılaşmadığı için avantajdır. Eş. 4.12' deki gibi tanımlanır.

$$\sigma(x)=\text{ReLU}(x)=\max(0, x) \quad (4.12)$$

Softmax fonksiyonu



Şekil 4.10. Softmax fonksiyonu grafiği [100]

Softmax fonksiyonu, sayıları olasılıklara göre ölçeklendiren bir aktivasyon fonksiyonudur [100]. Bir Softmax fonksiyonunun çıktısı, her olası sonucun olasılıklarına sahip bir vektördür ve bu vektördeki olasılıklar, tüm olası sonuçlar Şekil 4.10' da görüldüğü gibi olacak şekilde $[0,1]$ aralığında ve toplamı bire eşittir. Diğer aktivasyon fonksiyonlarından farklı olarak genellikle derin öğrenme modelinde son katman olarak yerleştirilir ve çoklu sınıflandırma problemlerinde kullanılır [100].

4.3.5. Kayıp fonksiyonu

Bir makine öğrenmesi modelinin tahmin kapasitesini belirlemek, maliyet fonksiyonu olarak da adlandırılan bir kayıp fonksiyonuna dayanır. Eğitim sırasında ağ, girdi verileriyle

birlikte çıktı üretir. Ağ çıktısı ile istenen çıktı arasındaki farka hata denmektedir ve kayıp fonksiyonunda amacı, tahmin edilen ve doğru hedef değerler arasındaki hata oranını ölçmektir. Öğrenmeyi kolaylaştırmak için de hatayı ölçen bir kayıp fonksiyonu oluşturulur. Bu nedenle, en iyi gerçekleştirilen makine öğrenmesi modelini elde etmek için, kayıp fonksiyonunun çıktısı en aza indirilmelidir. Çeşitli kayıp fonksiyonları vardır.

Mean Squared Error

Bir regresyon problemi için tahmin edilen ve gerçek hedef değerler arasındaki ortalama kare farkını hesaplamak için kullanılan yaygın olarak kullanılan kayıp fonksiyonu, Mean Squared Error (MSE)' dir. Eş. 4.13' teki gibi tanımlanır [101]:

$$MSE(y, \hat{y}) = \frac{1}{N} \sum_{i=1}^N (\hat{y}_i - y_i)^2 \quad (4.13)$$

Burada N, gözlemlerin sayısıdır, y_i gerçek değeri belirtir ve tahmin edilen değer $\hat{y}$ ile gösterilir. MSE genellikle Tanjant ve doğrusal aktivasyon fonksiyonları ile kullanılır. Hataların normal dağıldığını varsayar.

Cross Entropy

Hata oranını ölçmek için yaygın olarak kullanılan bir başka kayıp fonksiyonu, Eş. 4.14' te verilen Cross Entropy (CE)' dir [102]:

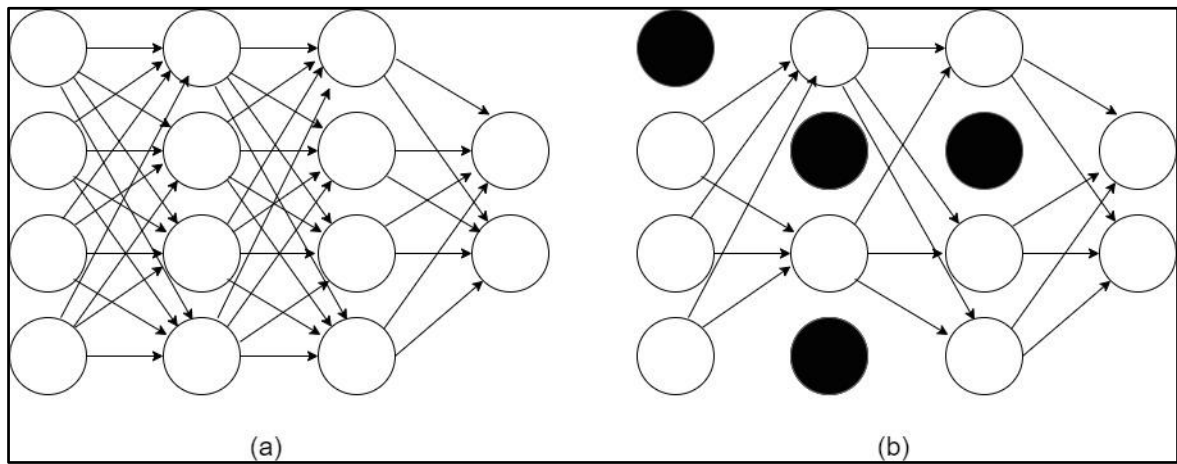
$$CE(y, \hat{y}) = \frac{1}{N} \sum_{i=1}^N y_i \log(\hat{y}) \quad (4.14)$$

Buna göre Cross Entropy, aktivasyon nöronları olasılıkları temsil ettiğinde kullanılır ve genellikle sinir ağlarında sigmoid aktivasyon fonksiyonu ile kombinasyon halinde kullanılır. Uygulamada CE, sınıflandırma hatası oranları açısından çoğunlukla daha hızlı yakınsama ve MSE' den daha iyi sonuçlar vermektedir [103].

4.3.6. Dropout

Modelin aşırı öğrenmesini engellemek için eğitim aşamasında rastgele seçilen belirli nöron kümesinin rastgele nöronlarını her yinelemede kaldırılması dropout ile sağlanır [94]. Şekil

4.11’ deki gibi her yinelemede, (a) kısmında standart sinir ağı görülmektedir. (b) kısmında ise bu sinir ağına dropout uygulanarak her katmandaki bazı nöronları rastgele kapatılır ve bu nöronlar hem ileri yayılmada hem de geri yayılmada kullanılmaz. Her yinelemede çıkarılacak nöronlar rastgele olacağından, öğrenme algoritmasının her yinelemede hangi nöronların kapatılacağı hakkında hiçbir fikri olmaz; bu nedenle, öğrenme algoritmasını ağırlıkları dağıtmaya ve bazı belirli özelliklere odaklanmaya zorlar. Dropout, nöronları daha sağlam olmaya, diğer birimlere bağlı kalmadan kendi başlarına öğrenme özelliğine zorlar.



Şekil 4.11. Dropout modeli gösterimi

4.3.7. Optimizasyon algoritması

Ağırlıklar bir optimizasyon algoritması ile güncellenir. Amaç, performansı en iyi olan sonucu bulmaktır. SGD [104], Adam [105] ve diğerleri gibi optimizasyon algoritmaları için birkaç yöntem vardır.

Stochastic Gradient Descent (SGD) algoritması

Gradyan, bir yüzeyin eğimi demektir. Gradyan descent, o yüzeydeki en alçak noktaya ulaşmak için bir eğimden inmek anlamından yola çıkarak, girdilerin bir dizi parametresinin kısmi türevi olan dışbükey bir fonksiyondur. Eğim ne kadar büyükse, eğim o kadar diktir. Rastgele bir başlangıç değerinden başlayarak, Gradyan Descent, verilen kayıp fonksiyonunun minimum olası değerini ve parametrelerin optimum değerlerini bulmak için yinelemeli olarak çalıştırılır [104]. Ağın hatasını azaltmak için bir makine öğrenimi ağ yapılandırmasında yinelemeli olarak küçük ayarlamalar yapar. SGD, her eğitim

noktasından sonra ağın yapılandırmasını ayarlayarak minimum değeri bulmaya çalışır. Tüm veri seti için hatayı azaltmak veya gradyanı bulmak yerine, bu yöntem sadece stochastic anlamında olduğu gibi rastgele seçilen bir batch için gradyan yaklaştırarak hatayı azaltır [104].

Adam algoritması

Adam algoritması, standart SGD algoritmasının bir uzantısıdır. SGD' de, tüm ağırlıkları güncelleyen ve eğitim süreci boyunca sabit kalan tek bir öğrenme oranı vardır. Buna karşılık Adam algoritması, her parametre için uyarlanabilir öğrenme oranlarını hesaplar [105].

4.4. Veri İşleme Yöntemleri

4.4.1. Kayan pencere tekniği

Zaman serisi verileri, belirli bir zaman aralığında alınan verilerdir ve gözlem süresi boyunca büyük ölçüde değişebilir ve bu nedenle doğrusal olmayan hale gelir. Kayan pencere, zaman serisi verilerinin gerçek değerinin geçici bir tahminidir [106]. Zaman penceresi çok büyükse, birden çok hareket hakkında bilgi içerebilir ve tanıma sisteminin doğruluğu azaltır ve hesaplama yükünü artırır. Aksine, zaman penceresi çok küçükse, bazı hareketler birden fazla ardışık pencereye bölünebilir. Her seferinde pencere boyutu kadar veri alınır ve işlem, tüm zaman serisi verileri bölümlere ayrılana kadar tekrarlanır [107]. Kayan pencere, yapılan cerrahi görevin çerçeveye karşılık gelen hareketini tahmin etmek için geçmiş zaman serisi verilerini kullanır. Bu çalışmada, pencere boyutu 16 olarak kullanılmıştır. Şekil 4.12, pencere boyutu 16 olan kayan pencere sürecini göstermektedir. Her sayı (1, 2, 3, ... , 10) sırasıyla 1, 2, 3, ... ,10 çerçevelerine ait zaman serisi verilerinin temsil eder. Başlangıçta, 1' den 16' ya kadar olan aralık, sonraki hareketin sınıfının tahmin edilmesi için 16 çerçevelik geçmiş verilerinin kullanıldığını temsil eder, ardından pencere, sonraki hareketi tahmin etmek için diğer 16 çerçeve gözlemlerini kapsayacak şekilde bir çerçeve sağa kayar.

İlk Pencere																								
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	
Pencere Sürgüsü																								
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	

Şekil 4.12. Kayan pencere gösterimi

4.4.2. Leave one user out (LOUO)

Eğitim / test verilerini bölmek için kullanılır. Her örnek bir kez test seti olarak kullanılırken, geri kalan örnek eğitim setini oluşturur. Bizim çalışmamızda 8 cerrahın denemesinden oluştuğu için her bir eğitim / test işleminde 1 cerrahı dışarıda test verisi olarak değerlendirip, geriye kalan 7 cerrahı eğitim verisi olarak kullanılması sağlanarak her eğitim / test sürecinde modelleri farklı verilerle karşılaştırılmıştır ve modellerin aşırı öğrenmesinin önüne geçilmiştir.

4.5. Performans Değerlendirme Yöntemleri

Derin öğrenmede kullanılan modellerin performansını ölçmek için kullanılan metrikler vardır. Bu çalışma, sınıflandırma problemine çözüm ürettiği için sınıflandırma modellerini değerlendirmeye uygun performans metrikleri kullanmıştır.

4.5.1. Karışıklık matrisi

Bir karışıklık matrisi, modelin performansını görselleştirmemizi sağlar. Çizelge 4.1' de karışıklık matrisindeki satırlar, model tarafından tahmin edilen sınıf etiketlerini temsil eder; sütunlar, test veri setindeki orijinal sınıf etiketlerini temsil eder. Karışıklık matrisi ile doğruluk, kesinlik, geri çağırma, F-1 skoru gibi ölçümleri hesaplanabilmektedir [108].

Çizelge 4.1. Karışıklık matrisi [108]

		Gerçek Değer	
		Pozitif	Negatif
Tahmin Edilen Değer	Pozitif	TP (True Positive- Gerçek Pozitif)	FP (False Positive- Yanlış Pozitif)
	Negatif	FN (False Negative- Yanlış Negatif)	TN (True Negative- Gerçek Negatif)

Gerçek pozitif (TP), pozitif bir gerçek değer doğru şekilde pozitif olarak sınıflandırıldığında tanımlandığında, yanlış pozitif (FP), pozitif olarak sınıflandırılan negatif bir gerçek değerdir, gerçek negatif (TN), negatif olarak doğru şekilde sınıflandırılan negatif bir gerçek değerdir ve yanlış negatif (FN), negatif olarak sınıflandırılan pozitif bir gerçek değerdir [108].

4.5.2. Doğruluk (Accuracy)

Doğruluk, tüm değerlere göre doğru şekilde sınıflandırılmış değerlerin yüzdesidir ve model performansını ölçmek için en yaygın kullanılan ölçüttür. Ancak, performansın doğru bir ölçütünü vermeyebilir ve hatalı olabilir. Örneğin, pozitif değerlerin negatif değerlere ağır bastığı veya tersinin olduğu dengesiz veri setleri üzerine eğitim, doğruluğu etkiler ve genel performansa genel bir bakış sunmaz [109]. Eş. 4.15' teki gibi hesaplanır:

$$\text{Doğruluk} = \frac{TP+TN}{TP+TN+FP+FN} \quad (4.15)$$

4.5.3. Kesinlik (Precision)

Gerçek pozitifin oranı da denilmektedir. Kesinlik, modelin aslında negatif olan bir örneği pozitif olarak etiketlememe yeteneğidir. Kesinlik, tahminlerin alaka düzeyini ölçmekle ilgili bir ölçüdür. Pozitif olarak tahmin edilen tüm sınıflar arasında pozitif tahmin edilen sınıfların oranını, dolayısıyla alternatif adı, pozitif tahmin değerini ölçer. Pozitif sınıflar için yüksek düzeyde yanlış sınıflandırmalar (yanlış pozitifler) alaka düzeyini etkileyebilir. Kesinlik ayrıca, tahminler gerçek değerlere yaklaştıkça, kesinlik değeri ne kadar yüksekse,

sonuçlardaki değişiklikleri de yakalar. Bu, aşağıdaki formülle hesaplanabilir [109]. Eş. 4.16' daki gibi hesaplanır:

$$Kesinlik = \frac{TP}{TP+FP} \quad (4.16)$$

4.5.4. Geri çağırma (Recall)

Geri çağırma puanı bize gerçekte pozitif olan tüm örneklerin yüzdesini ve doğru şekilde sınıflandırılan yüzdeyi verir. Tüm pozitif sınıflar arasında doğru tahmin edilen pozitif sınıflar olan gerçek pozitif oranı ölçer. Bu, her iki sınıfın da yanlış sınıflandırmalarındaki artıştan kaynaklanmaktadır [109]. Eş. 4.17' deki gibi hesaplanır:

$$Geri \text{ Çağırma} = \frac{TP}{TP+FN} \quad (4.17)$$

4.5.5. F1 skoru

F1 skoru, kesinlik ve geri çağırmanın ağırlıklı bir harmonik ortalamasıdır. Yukarıda bahsedilen ölçütlerin çoğu sınıflara bağlı ve sınıf dengesizliğine duyarlı olsa da F1 skoru, sınıf dengesizliğini dikkate alan ortak bir metriktir [109]. Eş. 4.18' deki gibi hesaplanır:

$$F1 \text{ skoru} = \frac{Doğruluk \cdot Geri \text{ Çağırma}}{Doğruluk + Geri \text{ Çağırma}} \quad (4.18)$$

4.5.6. Levenshtein mesafesi

Levenshtein mesafesi, iki kaynak arasındaki benzerliğin bir ölçüsüdür [110]. Levenshtein [111] tarafından geliştirilen metrik yöntem, birinci kaynağı ikinci kaynağa dönüştürmek için asgari sayıda temel düzenleme işlemi için gereken işlem sayısıdır. Levenshtein tarafından ele alınan temel düzenleme işlemleri şunlardır: Ekleme, Silme, Permütasyon. Levenshtein mesafesi ne kadar büyükse, o kadar farklılık vardır. Bu çalışmada, kaynaklardan biri modellerimizin tahmin ettiği etiket değeri ile gerçek etiket değerlerinin benzerliğini bulmak için Levenshtein mesafesinden yararlanılmıştır.

Bu çalışmada, cerrahi hareket tahmininde modelinde ne kadar iyi çalıştıklarını belirlemek için doğruluk, F1-skoru, Levenshtein mesafesi değerlendirme ölçütleri kullanılmıştır.



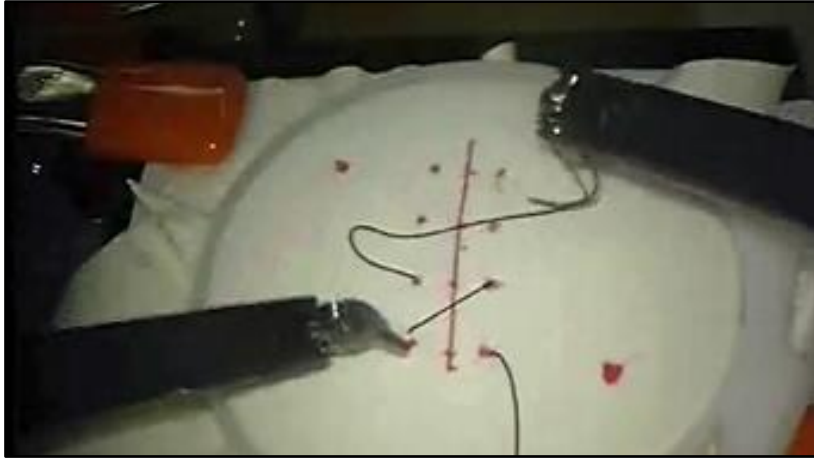
5. DENEY VE SONUÇLARI

Bu çalışmada, kinematik veri ile cerrahi hareketleri tanımda literatürde bu konuda sıklıkla kullanılan bir yöntem olan LSTM modeli [18], bu çalışma için geliştirdiğimiz Transformer modeli (Bkz. Bölüm 5.2.) ile kıyaslamak için kullanılmaktadır. Bu bölümde önce modelleri eğitmeden önce kullanılacak olan veri setinin hazırlanma sürecinden bahsedilip, ardından modellerin eğitim sürecinden bahsedilecektir. Modeller eğitilerek daha sonra performansları değerlendirilmiştir. Her iki model için de sadece JIGSAWS veri setindeki dikiş atma görevine ait veriler üzerinde çalışılmıştır.

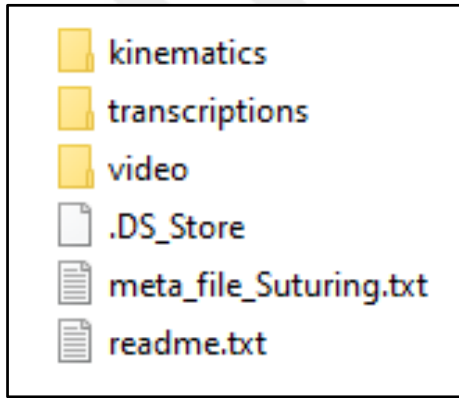
5.1. Veri Seti Hazırlama

Kullandığımız veriler zaman serisi formatındadır. Bu, her değerın cerrahi robotların kollarından okunan zaman içinde belirli bir anda meydana gelen bir okuma olduğu anlamına gelir. Zaman serileri, sıralı yapıya sahip özel bir veri türüdür. Veri noktalarının bağımlılığını ve verilerin anlamını etkilediği için sıralama özelliği çok önemlidir. Zaman serilerini oluşturan veri noktalarının daha önceden tanımlanmış bazı özellikleri vardır. Özelliklerden biri, veri noktalarının eşit aralıklarla zaman içinde tekrarlanan ölçümlerle elde edilmesinin gerekmesidir. Veri noktalarının zaman aralığı sürekli olmalı ve her bir zaman birimi gözlemi en fazla bir veri noktasına sahip olmalıdır.

Zaman serileri, istatistik, ekonomi, örüntü tanıma, kontrol mühendisliği, sinyal işleme, astronomi, meteoroloji gibi çeşitli alanlarda kullanılmaktadır. Zaman serileri analizinin araştırma ve uygulama alanları gelişmektedir. Zaman serilerindeki gürültü, ana problemlerden biridir ve yanlış tahminlere neden olur [112]. Verinin boyutunu azaltmak ve gürültüyü gidermek için, verilere çeşitli boyutsallık azaltma teknikleri (Principal Component Analysis - PCA) uygulanabilir. Ancak, bu prosedürler genellikle uzman bilgisi gerektirir ve verilerin bilgilendirici kısımlarını kaybedebilir. Veri seti 76 değişkenden oluşmakta ve bu çalışmada yakın zamanlı ve iyi performans gösteren Recognizing Surgical Activities with Recurrent Neural Networks [18] adlı makaleyi örnek alarak cerrah ve hasta tarafındaki sağ ve sol robot kollarının x, y, z pozisyonları; x, y, z hızları, gripper olmak üzere toplam 14 değişken kullanılmıştır. Böylece verinin boyutunun azaltılması sağlanmıştır.

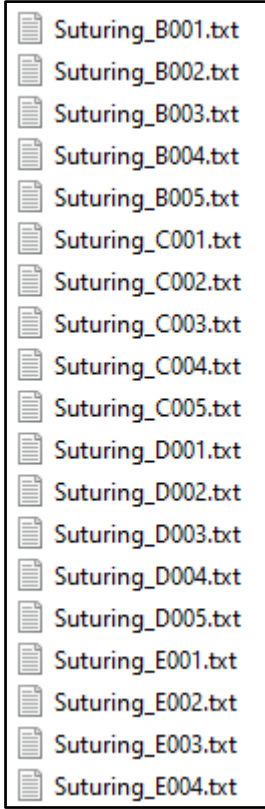


Resim 5.1. Dikiş atma görevi B cerrahı tarafından denemesi yapıldığı videodan bir çerçeve [61]



Resim 5.2. İndirilen veri setinin içindeki klasör ve dosyalar

Dikiş atma veri seti Johns Hopkins Üniversitesi tarafından erişime açık olarak sunulmuştur [113] ve içerisinde Resim 5.2’ deki dosyalar bulunmaktadır. readme.txt dosyasında veri seti ile ilgili bilgilendirmeleri okuduktan sonra, önce transcriptions klasöründe Resim 5.3’ te görüldüğü gibi her bir cerrahın denemeleri olan videolara ait, kinematik verileri kullanılabilir hale getirici bilgiler içermektedir. Resim 5.4’ teki gibi ilk sütunda başlangıç çerçevesi, ikinci sütunda bitiş çerçevesi ve son sütunda cerrahi hareket etiketi belirtilmiştir.



Resim 5.3. Dikiş atma görevine ait olan kinematik veri içeren dosyalar

80	219	G1
220	370	G5
371	590	G8
591	660	G2
661	954	G3
955	1097	G8
1098	1124	G2
1125	1401	G3
1402	1439	G2
1440	1698	G8
1699	1783	G2
1784	2285	G3
2286	2495	G6
2496	2679	G4
2680	2750	G2
2751	2976	G3
2977	3155	G6
3156	3308	G4
3309	3659	G2
3660	4011	G3
4012	4321	G8
4322	4374	G2
4375	4704	G3
4705	4846	G6
4847	4983	G4
4984	5059	G2
5060	5354	G3
5355	5447	G6
5448	5635	G11

Resim 5.4. B001 denemesine ait başlangıç – bitiş çerçeve numaraları ve etiket bilgisi

Buradan yola çıkarak her bir deneme için kinematics dosyasının içinde yer alan kinematik veriler, bir önceki transcriptions dosyasındaki bilgilere göre veri seti oluşturularak etiketlendirilmiş veri seti oluşturulmuştur. Resim 5.5- 5.6' da oluşturulan 15 boyutlu veri setinin ilk 10 satırı görülmektedir.

	pos_x	pos_y	pos_z	vel_x	vel_y	vel_z	gripper
0	0.027341	0.025038	0.002805	0.008277	-0.01157	0.016445	0.227176
1	0.02741	0.024581	0.00334	-0.005032	-0.010106	0.010453	0.536448
2	0.027352	0.024415	0.003517	-0.005765	-0.005372	0.005293	0.521589
3	0.027132	0.024301	0.00364	-0.006597	-6.9e-05	0.000453	0.521589
4	0.026901	0.024417	0.003522	-0.007098	0.003515	-0.005346	0.521589
5	0.026726	0.02459	0.003337	-0.001013	0.006278	-0.0043	0.521426
6	0.026696	0.024713	0.003266	-0.001146	0.006994	-0.003906	0.519956
7	0.026665	0.024961	0.003163	-0.001143	0.006983	-0.003915	0.520119
8	0.026586	0.025123	0.002977	-0.000453	0.00329	-0.005953	0.520283
9	0.02654	0.02518	0.002825	-0.000192	0.001896	-0.006724	0.520446

Resim 5.5. Veri setinin 0-7 sütun aralığında ilk 10 satır (B001 denemesine ait)

_pos_x	_pos_y	_pos_z	_vel_x	_vel_y	_vel_z	_gripper	label
0.086707	0.046216	-0.019131	-0.017022	0.001125	0.008558	0.302558	G1
0.086357	0.046115	-0.01893	-0.006825	-0.006246	0.005107	0.522184	G1
0.086285	0.046029	-0.018911	-0.000973	-0.0035	-0.0013	0.522674	G1
0.086293	0.045811	-0.018932	0.004548	-0.005484	-0.002439	0.522348	G1
0.086342	0.045655	-0.018996	-0.000769	-0.002795	-0.001053	0.522511	G1
0.086323	0.045617	-0.019005	-2.3e-05	0.000139	2.2e-05	0.522674	G1
0.086297	0.045635	-0.018991	0.000374	0.001346	0.000511	0.522837	G1
0.086269	0.04569	-0.018978	0.000118	0.000421	0.000159	0.522837	G1
0.086286	0.045723	-0.018977	0.000144	0.000519	0.000196	0.523327	G1
0.086296	0.045756	-0.018991	0.000395	0.005143	-0.00515	0.523164	G1

Resim 5.6. Veri setinin 8-15 sütun aralığında ilk 10 satır (B001 denemesine ait)

Veri setindeki veriler hakkında istatistiksel bilgi sahibi olmak için önce veri setindeki her bir sütunda null değere olup olmadığı bakılmıştır ve null değer bulunmadığı görülmüştür (Resim 5.7.). Sonra, her bir etiket değerine ait veri sayısına bakılmıştır ve Resim 5.8' de görüldüğü üzere B001 denemesinde en fazla G3 hareketi ve en az G1 hareketi kullanılmıştır. Resim 5.9' da B001 veri setindeki her bir özelliğine dair özet bilgileri, ortalama, minimum, maksimum ve bazı yüzdeliklerin değeri görülmektedir. Dağılım hakkında fikir edinmek için de her bir girdi değişkeninin histogramı oluşturulmuştur. Özellikle, _gripper değişkeninde, pos_y ve pos_z değişkenlerinde yoğun bir dağılım olduğu görülmüştür.

```
1 trainxdata.isna().sum()
pos_x      0
pos_y      0
pos_z      0
vel_x      0
vel_y      0
vel_z      0
gripper    0
_pos_x     0
_pos_y     0
_pos_z     0
_vel_x     0
_vel_y     0
_vel_z     0
_gripper   0
label      5556
dtype: int64
```

Resim 5.7. B001 denemesine ait veri setinde null değer olup olmadığı

```
label
G1      140
G2      771
G3     2276
G4      474
G5      151
G6      624
G8      932
G11     188
dtype: int64
```

Resim 5.8. B001 denemesine ait veri setinde her bir etiket in veri setindeki dağılımı

```

      pos_x    pos_y    ...    _vel_z    _gripper
count  5556.000000  5556.000000  ...  5556.000000  5556.000000
mean     0.039051    0.044925  ...   -0.000007   -0.369338
std      0.007840    0.010192  ...    0.009937    0.572877
min      0.007943    0.023186  ...   -0.038327   -1.173425
25%      0.033401    0.037050  ...   -0.002075   -1.025892
50%      0.041461    0.044427  ...   -0.000001   -0.280634
75%      0.044894    0.053989  ...    0.001347    0.177233
max      0.055304    0.074417  ...    0.067254    0.524960

[8 rows x 14 columns]
```

Resim 5.9. B001 denemesine ait veri setindeki her bir özelliğin özet bilgileri

Veri ön işleme, ölçeklendirme ya da normalize etmek, ağın hızlı öğrenmesini ve yakınsamasını sağlamak için önemli bir adımdır [114]. Ayrıca modelin öğrenmesi gereken

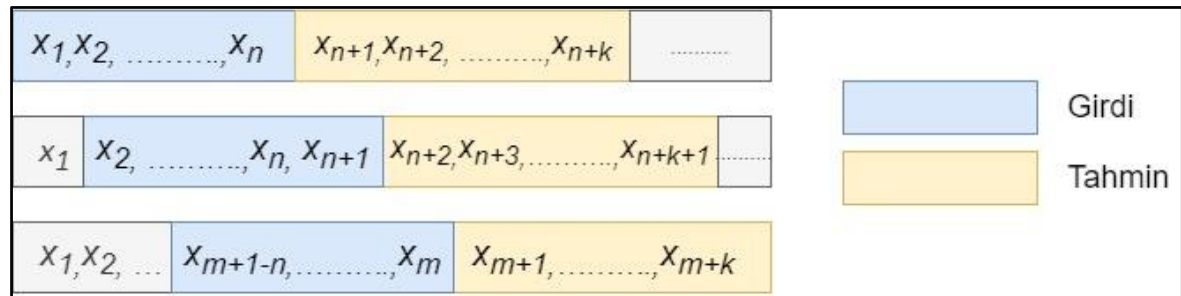
tahminlerin aralığını sınırlandırıcı zaman serisi tahmin analizi için önemli bir adımdır. Bu yüzden hem eğitim hem test verileri normalize edilmiştir. Önce ortalaması hesaplanarak ardından standart sapması ve son olarak her bir veriden ortalama ile farkı alınarak standart sapmaya bölünerek normalize edilmesi sağlanmıştır. Eş. 5.1' deki gibi hesaplanır ve x_i bir veri noktasını ($x_1, x_2, \dots, x_n$), $\bar{x}$ ortalamayı, s ise standart sapmayı göstermektedir:

$$Z = \frac{x_i - \bar{x}}{s} \quad (5.1)$$

Her bir çıktı etiketine, matematiksel hesaplamalarda kullanılması için Çizelge 5.1' deki gibi her birine karşılık bir sayısal değer atanmıştır. Diğer cerrahi görevlerden farklı olarak dikiş atma görevinde G7 hareketi kullanılmamaktadır.

Çizelge 5.1. Hareket etiketlerine karşılık gelen sayısal etiketler

Hareket Etiket	G1	G2	G3	G4	G5	G6	G8	G9	G10	G11
Sayısal Etiket	0	1	2	3	4	5	6	7	8	9



Şekil 5.1. Zaman serisi verilerinde kayan pencere tekniği

Eğitim ve test verileri, Şekil 5.1' de açıklanan kayan pencere tekniği kullanılarak giriş ve çıkış bileşenlerine bölünmüştür. Kayan pencere tekniğinde, önceki n (pencere boyutu) zaman adımları bir girdi olarak kullanılır ve sonraki adım k (tahmin sayısı) zaman adımları bir çıktı değişkeni olarak kullanılır. LSTM ve Transformer ağları için denetimli öğrenmeyi etkinleştirmek için kayan pencere tekniği uygulanır. Zaman serilerini denetimli öğrenme problemine dönüştürdükten sonra, veriler çalışmada kullandığımız ağlar için belirtilen şekilde [örnekler, pencere boyutu, özellikler] girdi olarak alınmıştır. Tüm bu veri ön işleme adımları her zaman serisi için uygulanmıştır. Verilerin m satır ve n sütunları vardır. Cerrahi hareket için alınan zaman damgası veya okuma sayısına karşılık n satırdır. Verimizi Resim

5.10’ da görüldüğü gibi hem eğitim hem test verileri 16’ lık pencereler şeklinde ve birbirine yakın veriler içerdiği için her 3 çerçeveden yani her 3 satırdan birini ele aldık. Verimizi 16’ lık pencerelere böldüğümüz için de her zaman adımındaki kullanılan matris boyutu 16 x 14 ‘tür.

```
[[-1.851393 -1.5946742 -0.04097755 -0.24939446 1.887121 -1.9239887
 2.4276872 0.7287024 0.54377043 1.7271677 -0.6754436 1.855242
 -2.1509397 1.517876 ]
 [-1.8632562 -1.5576823 -0.09748917 -0.20164138 1.5817941 -2.0688028
 2.3659961 0.7204078 0.59534127 1.6624837 0.01453841 2.048842
 -2.8193903 1.5207268 ]
 [-1.8739713 -1.5254983 -0.15971208 -0.16184714 1.3308908 -2.1895416
 2.311924 0.71543103 0.6388616 1.5969323 -0.06187935 1.7926035
 -2.9035254 1.5067592 ]
 [-1.8780533 -1.5087194 -0.19498174 -0.15132093 1.2603744 -2.2224047
 2.2833662 0.715668 0.6574335 1.5625305 -0.12706283 1.172996
 -2.4124024 1.5007731 ]
 [-1.8890237 -1.4740825 -0.27453887 -0.12693092 1.1076379 -2.2964368
 2.1889267 0.7182749 0.688918 1.4901855 -0.2263195 0.8432967
 -2.5265281 1.494787 ]
 [-1.8968049 -1.438366 -0.36241242 -0.17185989 1.5003054 -3.047713
 2.0423226 0.72408116 0.7125912 1.4085174 0.3509494 0.26107562
 -2.7667556 1.4537379 ]
 [-1.8960395 -1.4097145 -0.45329192 -0.04785592 0.6026124 -2.5441737
 1.8709643 0.7290579 0.7251053 1.3173091 0.238977 0.28403214
 -3.4689207 1.3476932 ]
 [-1.8937434 -1.4011779 -0.497379 -0.03668786 0.53077936 -2.5812502
 1.7982314 0.7307168 0.72685885 1.2717773 0.21836026 0.25189298
 -3.4890487 1.2884011 ]
 [-1.8891512 -1.3974493 -0.5687199 0.04533961 -0.09684498 -1.9491476
 1.7304515 0.7275175 0.72175753 1.18982 0.20515072 -0.6028548
 -2.394287 1.1971813 ]
 [-1.8873653 -1.4086351 -0.6230272 0.13429898 -0.7807946 -1.3458154
 1.7148386 0.71720845 0.70828694 1.1317128 -0.44581994 -0.6885591
 -1.6856813 1.1860646 ]
 [-1.8863449 -1.4167793 -0.6470747 0.1305763 -0.7600201 -1.3382317
 1.7102679 0.7102173 0.6978453 1.1101032 -0.4026112 -0.5422933
 -1.6336504 1.2102937 ]
 [-1.8804771 -1.4365999 -0.68815583 0.11491534 -0.68174994 -1.3046464
 1.704937 0.695287 0.6725779 1.0793151 -0.3885375 -1.2946112
 -0.49963945 1.2941028 ]
 [-1.871165 -1.455243 -0.725229 0.9222248 -0.3369223 -1.1995567
 1.7018914 0.6818971 0.6451585 1.0579946 -0.33927953 -1.1286684
 -0.43925548 1.3516858 ]
 [-1.8632562 -1.4678026 -0.7574928 0.02659781 -0.090993 -1.0077953
 1.7053171 0.66755927 0.6195724 1.0470815 -0.96247315 -0.654015
 -0.22760966 1.3884579 ]
 [-1.8580261 -1.4726106 -0.76911575 0.12415787 -0.81678426 -0.42528844
 1.7037942 0.6602126 0.60801476 1.0453469 -0.9029685 -0.45790076
 -0.15082137 1.4129732 ]
 [-1.856623 -1.4947861 -0.7891553 -0.6649233 -1.2982556 -0.60741967
 1.7037942 0.6469412 0.591595 1.0438291 -0.1223716 -0.38039517
 -0.15404186 1.4582976 ]]
```

Resim 5.10. Veri setine 16’ lık kayan pencere şeklinde okunmasında oluşan veri

5.2. Modellerin Eğitim Süreci

LOUO yöntemi ile her test verisi için farklı cerrahlar seçilerek 8 deneme gerçekleştirilmiştir. Deneylerde her bir modelin tanıma performansını ve çalışmanın katkısını göstermek için doğruluk payı hesaplanmış ve Levenshtein mesafesi bir değerlendirme kriteri olarak kullanılmıştır.

Sinir ağları, sürekli olarak gradyan descent kavramına dayalı olarak eğitim almaktadır ve eğitim sürecinde kayıp değerinin sürdürülmesi gereklidir. Kayıp fonksiyonu, eğitilecek epochlar ve nöron sayısı gibi parametreleri artırdıkça modelin öğrenip öğrenmediğini, kayıp fonksiyonu değeri düşmeye başladığında modelin doygun olduğu ve yeni bir şey öğrenmediği anlamına gelir. Model eğitimini olumlu etkileyebilecek parametrelerden bazıları, öğrenme hızı, optimize edici, batch boyutudur. Bu çalışma için geliştirdiğimiz için hem LSTM hem Transformer modellerinde kayıp fonksiyonunu izleyerek, farklı parametre kombinasyonları deneyerek parametreler ayarlanmıştır.

LSTM modeli için kullanılan hiperparametreler yakın zamanda iyi performans gösteren DiPietro ve diğerlerinin [18] çalışması örnek alınmıştır. DiPietro' nun çalışmasında oluşturulan LSTM modeli, JIGSAWS ve MISTIC-SL veri setleri üzerinde dikiş atma görevinde cerrahi hareket tanıma için kullanılmıştır ve sonuçlar MsM-CRF, SDSDL, SC-CRF, LC-SC-CRF modelleri ile karşılaştırılmıştır. MsM-CRF modeli [21], zamansal hareket segmentasyonu ve tanıma için Markov ve semi-Markov CRF modellerinin birleştirilmesiyle oluşturulmuştur. Model, çerçeveler arasındaki her bir hareketin genel özelliklerini ve hareketler arasındaki etkileşimleri yakaladığı belirtilmiştir ve CRF / semi-CRF modellerine göre gelişmiş olduğu belirtilmiştir. SDSDL modeli [29], cerrahi hareketleri etkin bir şekilde analiz etmek için kullanılabilen, paylaşılan, ayırt edici, seyrek sözlük öğrenimine dayalı bir cerrahi hareket segmentasyonu ve sınıflandırma yöntemi olarak önerilmiştir. Her hareketi sınıflandırmak için çok sınıflı bir doğrusal destek vektör makinesi ile birlikte tüm hareketler için ortak bir sözlüğü öğrenmek için önerilmiştir. Önerilen yöntem JIGSAWS veriseti üzerinde yapılan deneylerde, yalnızca kinematik verilere dayanan en son yöntemlerden eşit veya daha iyi performans gösterdiği ortaya koyulmuştur. SC-CRF (Skip-Chain CRF) modeli [28], hareketler arasındaki uzun menzilli durum geçişlerini yakalamak için daha yüksek dereceli zamansal ilişkiler kullanarak geliştirilmiştir. LC-SC-CRF (Latent-Convolutional Skip-Chain CRF) modeli [115], zaman

serisi modeli olarak sensör verilerinden yorumlanabilir ve birleştirilebilir hareket ilkelerini öğrenmeye dayandığı belirtilmiştir. Modelin alan bilgisi gerektirmeden iyi performans gösterdiği belirtilmiştir.

Çizelge 5.2. DiPietro ve diğerlerinin [18], cerrahi hareket tanıma çalışmasındaki sonucu

Veri Seti	JIGSAWS	MISTIC-SL
	Accuracy (%)	Accuracy (%)
MsM-CRF [21]	72.6	-----
SDSDL [29]	78.7	-----
SC-CRF [28]	80.3	-----
LC-SC-CRF [115]	82.5 ± 5.4	81.7 ± 6.2
Forward LSTM [18]	80.5 ± 6.2	87.8 ± 3.7
Bidir.LSTM [18]	83.3 ± 5.7	89.5 ± 4.0

Çizelge 5.2’ de görüldüğü üzere LSTM modelinin JIGSAWS veri setinde doğruluk payı %83.3 iken MISTIC-SL veri setinde doğruluk payı %89.5 olarak sonuçlanarak diğer modellere göre daha iyi sonuç verdiği görülmüştür. Bizde çalışmamızda LSTM modelinin yakın zamanda diğer modellere göre başarı göstermesi sebebiyle bu çalışma için geliştirdiğimiz Transformer modelinin karşılaştırmasını yapmak için oluşturduğumuz LSTM modelinde DiPietro ve diğerlerinin [18] çalışması ele alınmıştır. Böylece, LSTM modeli oluşturulurken DiPietro ve diğerlerinin [18] çalışmasında belirtildiği gibi batch boyutu 5, dropout değeri 0.5, epoch sayısı 80 olarak belirlenmiştir ve SGD optimize edici, aktivasyon fonksiyonu Softmax kullanılmıştır ve ilk 40 epocha kadar öğrenme oranı 1.0 ve ardından her 5 epochta bir yarıya düşürülmüştür.

Her bir kullanıcıya ait denemeye uygulanan bu çalışma için geliştirdiğimiz LSTM modelinin sonuçları aşağıdaki Çizelge 5.3’ de görülmektedir. Tabloya göre en yüksek doğruluk payı Test F kullanıcılarını denemesi, en düşük doğruluk payı Test G kullanıcılarını denemeleri olduğu görülmektedir. Doğruluk payının ortalaması 0.729, standart sapması 0.058 iken Levensthein mesafesinin ortalaması 33.36, standart sapması 10.03’ tür.

Çizelge 5.3. LSTM modelinin her test kullanıcısına uygulama sonuçları

User Out	Doğruluk	Kayıp	F1 Skoru	Kesinlik	Geri Çağırma	Levensthein Mesafesi
B	0.7205	0.8595	0.6896	0.7101	0.6797	36.6
C	0.7521	0.6611	0.7301	0.7537	0.7192	41.3
D	0.6599	1.2976	0.6365	0.6716	0.6210	32.9
E	0.7188	0.7878	0.6790	0.7089	0.6648	32.0
F	0.8373	0.6218	0.8173	0.8356	0.8084	52.2
G	0.6538	1.5069	0.6358	0.6540	0.6267	21.8
H	0.7569	0.7831	0.7218	0.7517	0.7076	23.8
I	0.7388	0.7915	0.7056	0.7291	0.6938	26.3

Transformer modelini zaman serisi verileri üzerinde uygulamak için yaptığımız araştırmalara göre öncelikle var olan mimaride birkaç değişiklik yapmak gerekmektedir [116]. İlk adım olarak, giriş verileri zaten sayısal değerlere sahip olduğu için embedding işlemi kaldırılmıştır. Bir embedding genellikle belirli bir tamsayıyı N boyutlu bir uzaya eşler. Burada bu işlemi gerçekleştirmek için de 14 boyutlu verileri N boyutlu bir uzaya dönüştürmek için basitçe bir doğrusal bir dönüşüm kullanılmıştır. Geriye dönük dikkat sınırlandırmak için 16' lık pencereler kullanılmıştır. Böylece, self-attention mekanizması tüm diziyi birleştiremez. Zaman adımlarını yalnızca dikkate alınan zaman adımına kadar dahil edebilmesi sağlanır. Gerisi oldukça standart bir dikkat mekanizmasını takip etmektedir. Transformer modelinde, farklı hiperparametre kombinasyonu denenmiştir. Batch boyutu 10, 32, 64, 128 değerleri ile; dropout değeri 0.1 ile 0.010; öğrenme oranı 0.001-0.010 aralığında değerlendirilerek en iyi sonuç veren hiperparametreler belirlenmiştir. Batch boyutu 10 ve dropout değeri 0.07, optimize edici olarak SGD öğrenme oranı için 0.003 değeri kullanılmıştır. Kayıp fonksiyonu olarak MSE kullanılmıştır. Transformer modelinin 8 kullanıcıdaki çalışma sonucu Çizelge 5.4' te gösterilmiştir.

Çizelge 5.4. Dikiş atma cerrahi görevini tanımak için geliştirilen Transformer modelinin sonucu

User Out	B	C	D	E	F	G	H	I
Doğruluk	0.75	0.78	0.70	0.78	0.88	0.72	0.79	0.76

Çizelge 5.4’ te görüldüğü üzere dikiş atma görevi tanımak için geliştirilen Transformer modelinin sonucu görülmektedir. En yüksek doğruluk payı, test F kullanıcısında doğruluk payı %88, en düşük doğruluk payı test kullanıcısı D doğruluk payı %70 elde edilmiştir. Ortalama doğruluk payı %77’ dir.

Bu çalışma için geliştirdiğimiz her iki modelin doğruluk metriği sonuçları daha önce JIGSAWS veri seti üzerinde LSTM modelleri uygulanan DiPietro ve diğerlerinin [18]’ in makale sonuçları ile karşılaştırılmıştır. Çizelge 5.5’ te literatürdeki sonuçlar ile kendi geliştirdiğimiz LSTM modelinin sonuçlarının karşılaştırması görülmektedir. Çizelgede DiPietro ve diğerlerinin [18]’ in çalışmasına göre en yüksek doğruluk payı %86.7, en düşük doğruluk payı %74.3’ tür. Bu çalışmada gerçekleştirilen LSTM modelinin en yüksek doğruluk payı %83, en düşük doğruluk payı %66’ dır. Tüm deneysel sonuçlar dikkate alındığında, bu çalışma için geliştirdiğimiz LSTM modelinde ortalama doğruluk payı %72, Transformer modelinde ortalama doğruluk payı %77’ dir. Transformer modeli ile elde edilen sonuçların standart sapması diğer yöntemlere göre daha azdır, bu da modelin güvenilirliğini desteklemektedir.

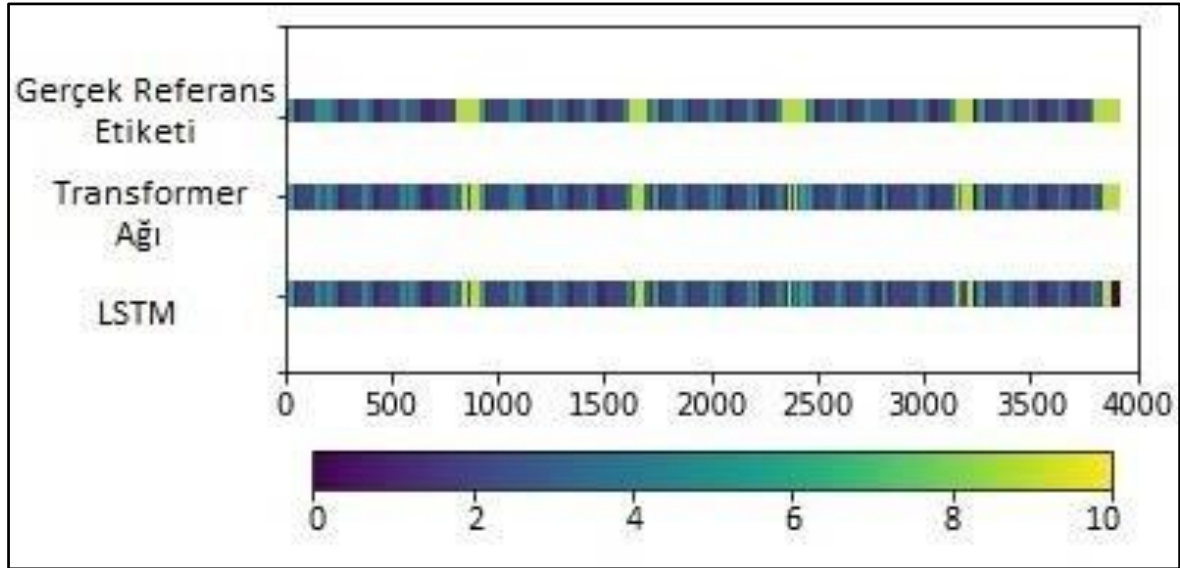
Çizelge 5.5. Dikiş atma cerrahi görevini tanımak için uygulanan modellerin sonuçları

Model	Doğruluk Payı (%)
LSTM DiPietro [18]	80.5 ± 6.2
LSTM ^a	72.9 ± 5.8
Transformer Network	77.0± 5.4

^aÇalışmadaki adaptasyonu

Modellerin veri setindeki performansını ölçmek için tahmin edilen bir dizi etiketin, gerçek etiket dizisine dönüştürülmesi için gereken hareket sayısına oranına karşılık gelen Levenshtein mesafesi hesaplanmıştır. Resim 5.11’ de F cerrahının birinci denemesine ait dikiş atma denemeleri için her iki model için de gerçekte referans etiketi ile her modelin tahmin sonucunda çıkan etiket değerlerinin renklendirilmesi görülmektedir. X eksenini ile çerçeve sayısı ve altta bulunan sütun grafikte etiketlerin (0-10) her birine karşılık gelen renkler görülmektedir. Gerçek referans etiketi ile her bir modelin çalışma sonucunda tahmin edilen etiketler renklendirilmiştir. Her renge karşılık gelen etiketler incelendiği

zaman, Transformer modelinin sonucunda tahmin edilen etiketler, gerçek etiket değerine LSTM modelinin tahmin ettiği etiketlere göre daha yakın görülmektedir.



Resim 5.11. Her bir çerçeveye karşılık gelen gerçek referans etiketi ile model sonuçlarında tahmin edilen etiketlerin gösterimi

6. SONUÇ

Robotik cerrahide görevlerin otomatik olarak tanınması ve sınıflandırılması, otomatik cerrahi ve cerrahi eğitim için önemli bir basamaktır. Robotik destekli cerrahi, cerrahların geleneksel cerrahi ile mümkün olandan daha fazla hassasiyetle birçok karmaşık operasyonu gerçekleştirmesine olanak tanır. Her yıl artan cerrahi prosedür sayısı ile, cerrahi uygulamayı geliştirmek için stajyerlerin cerrahi becerilerinin doğru, objektif ve otomatik bir değerlendirmesini sağlamak için benzeri görülmemiş bir ihtiyaç söz konusudur.

Bu çalışmanın odak noktası video, robot kinematiki bilgileri verilen bir cerrahi eylemi tahmin etmeye yönelik yapılan çalışmaların araştırması olmuştur. Bununla ilgili önceki çalışmalar ya HMM 'ler ve CRF' ler gibi değişken grafik modellerine ya da RNN, LSTM ve TCN gibi derin öğrenme modellerine dayanmaktadır.

Video tabanlı cerrahi hareket tanımda temel zorluk, yüksek boyutsallık, yüksek yedekleme düzeyi ve video verilerinin yüksek karmaşıklığıdır. Kinematik veriler, veri toplama sırasında çevrimiçi işlemeyi mümkün kılacak hareket tanıma için düşük boyutlu özellikler sağlamaktadır. Cerrahi robotlardan kaydedilen kinematik yörüngeler cerrahi hareketler hakkında bilgi içerir ve potansiyel olarak cerrahın beceri seviyeleri hakkında ipuçlarını kodlar. Bu yörüngelerin anlamlı hareket tanımlama, cerrahi beceri değerlendirmesi için yeni metrikler geliştirmenin yanı sıra cerrahi otomasyonu basitleştirmeye yardımcı olabilmektedir. Bundan dolayı, gelecek çalışmada, cerrahi hareket tanımak için sadece kinematik veriler üzerinde derin öğrenme yöntemlerinden biri ile cerrahi hareket tanımlama çalışması yapılarak literatürdeki sonuçların karşılaştırmasının yapılması, bir sorunu çözerken elde edilen bilgiyi farklı bir araştırma problemine uygulamaya odaklanan transfer öğrenimi ile başka bir veri setini JIGSAWS veri setine uygulayarak cerrahi hareket algılamaya dair çalışma yapılabilirliği araştırılacaktır.

Cerrahi hareketleri tanımak, mevcut cerrahi durumu anlamak ve cerraha otomatik geri bildirim sağlamak için ön adım oluşturmaktadır. Bilgisayarlı tıp eğitimi sistemlerinin gelişimi, otomatik değerlendirme yöntemleri üzerine araştırmaları desteklemiştir. Transformer modeli son zamanlarda doğal dil işlemede önemli ilerleme göstermiştir. Transformer modeli ayrıca zaman serisi verilerinde paralel çalışmaya uygun bir modeldir.

Bu çalışmada, bildiğimiz kadarıyla, literatürde ilk kez Transformer modeli kinematik veri ile cerrahi hareket tanımda kullanılmıştır. Geliştirdiğimiz Transformer modeli, kinematik veri ile JIGSAWS veri setinde tanımlanan cerrahi hareketleri tanımda test edilmiştir. Elde edilen sonuçlar literatürde sıklıkla kullanılan LSTM modelleri sonuçları ile karşılaştırılmıştır. Sonuçlara göre Transformer modelinin doğruluk payının literatürdeki LSTM modelleri sonuçları ile karşılaştırılabilir olduğu, kendi geliştirdiğimiz LSTM modeline göre daha iyi sonuç verdiği görülmüştür. Ayrıca, Transformer modelinin standart sapması LSTM modellerinden daha azdır ve bu durum Transformer modellerinin güvenilirliğini desteklemektedir. Transformer modelinin kinematik veri ile cerrahi aktivite tanımda kullanılabileceği görülmüş ve bu durum elde edilen sonuçlarla desteklenmiştir. Gelecek çalışmalarda farklı Transformer mimariler ve bu mimarilerin kinematik veri ile nasıl daha etkin bir şekilde eğitilebileceği araştırılacaktır.

6.1. Bu Tezden Çıkarılan Yayınlar

Danışmanım Dr. Öğr. Üyesi Duygu SARIKAYA ile 9-11 Haziran tarihinde düzenlenen 29. Sinyal İşleme ve İletişim Uygulamaları Kurultayında “Transformer Ağı ile Cerrahi Hareket Tanıma” adlı bildirimiz yayınlanmıştır.

Özetle, Robotik cerrahi hareketlerin otomatik olarak sınıflandırılması ve tanınması, hem cerrahi eğitimde geri bildirim sağlama hem de ameliyatlarda istenmeyen durum ve hataları önlemede önemli bir adımdır. Cerrahi robotlardan kaydedilen kinematik veriler, cerrahi hareketler hakkında bilgi içerir. Bu verileri kullanarak cerrahların hareketlerini otomatik olarak modelleyebilir ve tanıyabiliriz. Bu çalışmada, cerrahi hareketlerin tanınması için kinematik veriler ile son zamanlarda zaman serisi verileri üzerindeki çalışmalarda tekrarlayan sinir ağlarına göre daha iyi başarı gösteren Transformer modeli kullanılmıştır. Bu çalışmada öğrenilen model literatürde sıkça kullanılan Long Short-Term Memory (LSTM) modeli ile karşılaştırılmıştır. Transformer modelinin JHU-ISI Gesture and Skill Assessment Working Set (JIGSAWS) veri seti için ortalama olarak doğruluk payı %77’ dir. Sonuçlara göre Transformer modelinin doğruluk payının literatürdeki LSTM modeline göre ise daha iyi sonuçları ile karşılaştırılabilir olduğu, kendi geliştirdiğimiz LSTM modeline göre daha iyi sonuç verdiği, ayrıca LSTM modellerine göre daha az standart sapmaya sahip olduğu görülmüştür. Bildiğimiz kadarıyla çalışmamız bir Transformer

modelini kinematik veri ile cerrahi hareket tanıma için kullanan ilk çalışmadır. Deney sonuçlarımız Transformer ağının bu alanda kullanımını desteklemektedir.





KAYNAKLAR

1. Fox, R., Krishnan, S., Stoica, I., and Goldberg, K. (2017). Multi-level discovery of deep options. *Representations*, 5(6), 7-8.
2. Ettehad, N., Manaffam, S., and Behal, A. (2017, October). Learning from demonstration: generalization via task segmentation. *Materials Science and Engineering Conference Series*, 261(1), 012001.
3. Menegozzo, G., Dall'Alba, D., Zandonà, C., and Fiorini, P. (2019, April). *Surgical gesture recognition with time delay neural network based on kinematic data*. International Symposium on Medical Robotics (ISMR), Atlanta, 125-132.
4. Moorthy, K., Munz, Y., Dosis, A., Hernandez, J., Martin, S., Bello, F., Rockall, T.A. and Darzi, A. (2004). Dexterity enhancement with robotic surgery. *Surgical Endoscopy*, 18(5), 790-795.
5. Kassahun, Y., Yu, B., Tibebu, A. T., Stoyanov, D., Giannarou, S., Metzen, J. H., and Vander Poorten, E. (2016). Surgical robotics beyond enhanced dexterity instrumentation: a survey of machine learning techniques and their role in intelligent and autonomous surgical actions. *International Journal of Computer Assisted Radiology and Surgery*, 11(4), 553-568.
6. Ellis, R. D., Munaco, A. J., Reisner, L. A., Klein, M. D., Composto, A. M., Pandya, A. K., and King, B. W. (2016). Task analysis of laparoscopic camera control schemes. *International Journal of Medical Robotics and Computer Assisted Surgery*, 12(4), 576-584.
7. Fard, M. J. (2016). *Computational modeling approaches for task analysis in robotic-assisted surgery*, Doctoral dissertation, Wayne State University, Detroit.
8. DiPietro, R., Rupprecht, C., Navab, N., and Hager, G. D. (2017). *Analyzing and exploiting NARX recurrent neural networks for long-term dependencies*. *arXiv:170207805*.
9. Forestier, G., Petitjean, F., Senin, P., Despinoy, F., Huauilmé, A., Fawaz, H. I., Weber, J., Idoumghar, L., Muller, P.A. and Jannin, P. (2018). Surgical motion analysis using discriminative interpretable patterns. *Artificial Intelligence in Medicine*, 91, 3-11.
10. El-Saig, D., Elek, R. N., and Haidegger, T. (2018, November). *A Graphical Tool for Parsing and Inspecting Surgical Robotic Datasets*. 18th International Symposium on Computational Intelligence and Informatics (CINTI), Hungary, 000131-000136.
11. Castro, D., Pereira, D., Zanchettin, C., Macêdo, D., and Bezerra, B. L. (2019, July). *Towards optimizing convolutional neural networks for robotic surgery skill evaluation*. International Joint Conference on Neural Networks (IJCNN), Budapest, 1-8.

12. Itzkovich, D., Sharon, Y., Jarc, A., Refaely, Y., and Nisky, I. (2019, May). *Using augmentation to improve the robustness to rotation of deep learning segmentation in robotic-assisted surgical data*. International Conference on Robotics and Automation (ICRA), Montreal, 5068-5075.
13. Zhao, B., Waterman, R. S., Urman, R. D., and Gabriel, R. A. (2019). A machine learning approach to predicting case duration for robot-assisted surgery. *Journal of Medical Systems*, 43(2), 32.
14. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. and Polosukhin, I. (2017). *Attention is all you need*. Proceedings of the 31st International Conference on Neural Information Processing Systems, California, 6000-6010.
15. Anh, N. X., Nataraja, R. M., and Chauhan, S. (2020). Towards near real-time assessment of surgical skills: A comparison of feature extraction techniques. *Computer Methods and Programs in Biomedicine*, 187, 105234.
16. Castro, D., Pereira, D., Zanchettin, C., Macêdo, D., and Bezerra, B. L. (2019, July). *Towards optimizing convolutional neural networks for robotic surgery skill evaluation*. International Joint Conference on Neural Networks (IJCNN), Budapest, 1-8.
17. van Amsterdam, B., Nakawala, H., De Momi, E., and Stoyanov, D. (2019, May). *Weakly supervised recognition of surgical gestures*. International Conference on Robotics and Automation (ICRA), Montreal, 9565-9571.
18. DiPietro, R., Lea, C., Malpani, A., Ahmidi, N., Vedula, S. S., Lee, G. I., Lee, M.R. and Hager, G. D. (2016, October). *Recognizing surgical activities with recurrent neural networks*. 19th International Conference on Medical Image Computing and Computer-Assisted Intervention, Shenzhen, 551-558.
19. Gao, Y., Vedula, S. S., Lee, G. I., Lee, M. R., Khudanpur, S., and Hager, G. D. (2016). Query-by-example surgical activity detection. *International Journal of Computer Assisted Radiology and Surgery*, 11(6), 987-996.
20. Reiley, C. E., and Hager, G. D. (2009, September). *Task versus subtask surgical skill evaluation of robotic minimally invasive surgery*. International Conference on Medical Image Computing and Computer-Assisted Intervention, London, 435-442.
21. Tao, L., Zappella, L., Hager, G. D., and Vidal, R. (2013, September). *Surgical gesture segmentation and recognition*. International Conference on Medical Image Computing and Computer-Assisted Intervention, Nagoya, 339-346.
22. Lea, C., Vidal, R., Reiter, A., and Hager, G. D. (2016, October). *Temporal convolutional networks: A unified approach to action segmentation*. European Conference on Computer Vision, Amsterdam, 47-54.
23. Liu, D., and Jiang, T. (2018, September). *Deep reinforcement learning for surgical gesture segmentation and classification*. International Conference on Medical Image Computing and Computer-Assisted Intervention, Cham, 247-255.

24. Forestier, G., Lalys, F., Riffaud, L., Trelhu, B., and Jannin, P. (2012). Classification of surgical processes using dynamic time warping. *Journal of Biomedical Informatics*, 45(2), 255-264.
25. Lalys, F., Riffaud, L., Bouget, D., and Jannin, P. (2011). A framework for the recognition of high-level surgical tasks from video images for cataract surgeries. *IEEE Transactions on Biomedical Engineering*, 59(4), 966-976.
26. Keogh, E. J., and Pazzani, M. J. (2001, April). *Derivative dynamic time warping*. Proceedings of the SIAM International Conference on Data Mining, Chicago, 9-11.
27. Tao, L., Elhamifar, E., Khudanpur, S., Hager, G. D., and Vidal, R. (2012, June). *Sparse hidden markov models for surgical gesture classification and skill evaluation*. Information Processing in Computer-Assisted Interventions, Berlin, 167-177.
28. Lea, C., Hager, G. D., & Vidal, R. (2015, January). *An improved model for segmentation and recognition of fine-grained activities with application to surgical training tasks*. IEEE Winter Conference on Applications of Computer Vision (WACV), 1123-1129.
29. Sefati, S., Cowan, N. J., and Vidal, R. (2015). *Learning shared, discriminative dictionaries for surgical gesture segmentation and classification*. Medical Image Computing and Computer Assisted Intervention-MICCAI Workshop: M2CAI, Germany, Munich, 4.
30. Krishnan, S., Garg, A., Patil, S., Lea, C., Hager, G., Abbeel, P., and Goldberg, K. (2017). Transition state clustering: Unsupervised surgical trajectory segmentation for robot learning. *International Journal of Robotics Research*, 36(13-14), 1595-1618.
31. Islam, G., Kahol, K., Li, B., Smith, M., and Patel, V. L. (2016). Affordable, web-based surgical skill training and evaluation tool. *Journal of Biomedical Informatics*, 59, 102-114.
32. Fard, M. J., Ameri, S., and Ellis, R. D. (2016). Toward personalized training and skill assessment in robotic minimally invasive surgery. *arXiv:1610.07245*.
33. Zia, A., and Essa, I. (2018). Automated surgical skill assessment in RMIS training. *International Journal of Computer Assisted Radiology and Surgery*, 13(5), 731-739.
34. Fard, M. J., Ameri, S., Chinnam, R. B., Pandya, A. K., Klein, M. D., and Ellis, R. D. (2016). Machine learning approach for skill evaluation in robotic-assisted surgery. *arXiv:1611.05136*.
35. Despinoy, F., Bouget, D., Forestier, G., Penet, C., Zemiti, N., Poignet, P., and Jannin, P. (2015). Unsupervised trajectory segmentation for surgical gesture recognition in robotic training. *IEEE Transactions on Biomedical Engineering*, 63(6), 1280-1291.

36. Zia, A., Sharma, Y., Bettadapura, V., Sarin, E. L., Clements, M. A., and Essa, I. (2015, October). *Automated assessment of surgical skills using frequency analysis*. 18th International Conference on Medical Image Computing and Computer-Assisted Intervention - MICCAI, 430-438.
37. Ahmidi, N., Gao, Y., Béjar, B., Vedula, S. S., Khudanpur, S., Vidal, R., and Hager, G. D. (2013, September). *String motif-based description of tool motion for detecting skill and gestures in robotic surgery*. International Conference on Medical Image Computing and Computer-Assisted Intervention: MICCAI, Berlin, 26-33.
38. Gao, Y., Vedula, S. S., Lee, G. I., Lee, M. R., Khudanpur, S., and Hager, G. D. (2016, May). *Unsupervised surgical data alignment with application to automatic activity annotation*. IEEE International Conference on Robotics and Automation (ICRA), Stockholm, 4158-4163.
39. Bani, M. J., and Jamali, S. (2017). A new classification approach for robotic surgical tasks recognition, *arXiv:1707.09849*.
40. Maung, T. H. H. (2009). Real-time hand tracking and gesture recognition system using neural networks. *International Journal of Computer and Information Engineering*, 3(2), 315-319.
41. Sarikaya, D., Guru, K. A., and Corso, J. J. (2018). Joint surgical gesture and task classification with multi-task and multimodal learning, *arXiv:1805.00721*.
42. Wang, Z., and Fey, A. M. (2018). Deep learning with convolutional neural network for objective skill evaluation in robot-assisted surgery. *International Journal of Computer Assisted Radiology and Surgery*, 13(12), 1959-1970.
43. Funke, I., Bodenstedt, S., Oehme, F., von Bechtolsheim, F., Weitz, J., and Speidel, S. (2019, October). *Using 3d convolutional neural networks to learn spatiotemporal features for automatic surgical gesture recognition in video*. International Conference on Medical Image Computing and Computer-Assisted Intervention, 467-475.
44. Cifuentes, J., Boulanger, P., Pham, M. T., Prieto, F., and Moreau, R. (2019, July). *Gesture classification using LSTM recurrent neural networks*. International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Berlin, 6864-6867.
45. Gurcan, I., and Van Nguyen, H. (2019, May). *Surgical Activities Recognition Using Multi-scale Recurrent Networks*. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2887-2891.
46. DiPietro, R., Ahmidi, N., Malpani, A., Waldram, M., Lee, G. I., Lee, M. R., Vedula, S. S. and Hager, G. D. (2019). Segmenting and classifying activities in robot-assisted surgery with recurrent neural networks. *International Journal of Computer Assisted Radiology and Surgery*, 14(11), 2005-2020.

47. Fawaz, H. I., Forestier, G., Weber, J., Idoumghar, L., and Muller, P. A. (2019). Accurate and interpretable evaluation of surgical skills from kinematic data using fully convolutional neural networks. *International Journal of Computer Assisted Radiology and Surgery*, 14(9), 1611-1617.
48. Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., and Torralba, A. (2016). *Learning deep features for discriminative localization*. Proceedings of the IEEE Conference on Computer Cision and Pattern Recognition, Las Vegas, 2921-2929.
49. Qin, Y., Pedram, S. A., Feyzabadi, S., Allan, M., McLeod, A. J., Burdick, J. W., and Azizian, M. (2020, May). *Temporal segmentation of surgical sub-tasks through deep learning with multiple data sources*. IEEE International Conference on Robotics and Automation (ICRA), Paris, 371-377.
50. Sarikaya, D., and Jannin, P. (2020). Towards generalizable surgical activity recognition using spatial temporal graph convolutional networks, *arXiv:2011.03728*.
51. van Amsterdam, B., Clarkson, M. J., and Stoyanov, D. (2020, May). *Multi-task recurrent neural network for surgical gesture recognition and progress prediction*. Proceedings IEEE International Conference on Robotics and Automation (ICRA), Paris, 1380-1386.
52. Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou Y. and Liu, P. J. (2019). Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140), 1-67.
53. Huang, C. Z. A., Vaswani, A., Uszkoreit, J., Shazeer, N., Hawthorne, C., Dai, A. M., Dinculescu, M. and Eck, D. (2018). *Music transformer: Generating music with long-term structure*. International Conference on Learning Representations, Vancouver, *arXiv:1809.04281*.
54. Song, H., Rajan, D., Thiagarajan, J., and Spanias, A. (2018). *Attend and diagnose: Clinical time series analysis using attention models*. Proceedings of the AAAI Conference on Artificial Intelligence, New Orleans, 4091-4098.
55. Li, S., Jin, X., Xuan, Y., Zhou, X., Chen, W., Wang, Y. X., and Yan, X. (2019). Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting. *Advances in Neural Information Processing Systems*, 32, 5243-5253.
56. Ma, J., Shou, Z., Zareian, A., Mansour, H., Vetro, A., and Chang, S. F. (2019). CDSA: cross-dimensional self-attention for multivariate, geo-tagged time series imputation, *arXiv:1905.09904*.
57. Wu, N., Green, B., Ben, X., and O'Banion, S. (2020). Deep transformer models for time series forecasting: The influenza prevalence case, *arXiv:2001.08317*.
58. Czempiel, T., Paschali, M., Ostler, D., Kim, S. T., Busam, B., and Navab, N. (2021). OperA: Attention-Regularized Transformers for Surgical Phase Recognition, *arXiv:2103.03873*.

59. İnternet: Intuitive surgical. URL: <http://www.intuitivesurgical.com/>, Son Erişim Tarihi: 18.05.2021
60. İnternet: About da Vinci Systems. URL: <https://www.davincisurgery.com/da-vinci-systems/about-da-vinci-systems##>, Son Erişim Tarihi: 01.05.2021
61. Gao, Y., Vedula, S. S., Reiley, C. E., Ahmidi, N., Varadarajan, B., Lin, H. C., Lingling, T., Zappella, L., Bejar, B., Yuh., D.D., Chen, C.C.C., Vidal R., Khudanpur S. and Hager, G. D. (2014, September). *Jhu-isi gesture and skill assessment working set (jigsaws): A surgical activity dataset for human motion modeling*. Medical Image Computing and Computer Assisted Intervention-MICCAI Workshop: M2CAI, USA, Boston, 3(3).
62. İnternet: Python. URL: <https://www.python.org/>, Son Erişim Tarihi: 10.05.2021
63. İnternet: NumPy. URL: <https://www.numpy.org/>, Son Erişim Tarihi: 10.05.2021
64. İnternet: Keras. URL: <https://keras.io/>, Son Erişim Tarihi: 10.05.2021
65. İnternet: Matplotlib. URL: <https://matplotlib.org/>, Son Erişim Tarihi: 10.05.2021
66. İnternet: Scikit learn. URL: <https://scikit-learn.org/stable/>, Son Erişim Tarihi: 10.05.2021
67. İnternet: Pandas. A Foundational Python Library for Data Analysis and Statistics. URL: dlr.de/sc/en/Portaldata/15/Resources/dokumente/pyhpc2011/submissions/pyhpc2011_submission_9.pdf, Son Erişim Tarihi: 10.05.2021
68. İnternet: Python Data Analysis Library. URL: <https://pandas.pydata.org/>, Son Erişim Tarihi: 10.05.2021
69. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., steiner, B., Fang, Lu and Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems*, 32, 8026-8037.
70. İnternet: Colab Notebooks. URL: <https://magenta.tensorflow.org/demos/colab/>, Son Erişim Tarihi: 10.05.2021
71. İnternet: Colaboratory. URL: <https://colab.research.google.com/notebooks/welcome.ipynb#scrollTo=-Rh3-Vt9Nev9>, Son Erişim Tarihi: 10.05.2021
72. Rush, A. M., Chopra, S., and Weston, J. (2015). *A neural attention model for abstractive sentence summarization*. Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Lisbon, 379-389.

73. Golub, D., and He, X. (2016). Character-level question answering with attention, *arXiv:1604.00727*.
74. İnternet: Artificial Neural Network. Introduction to Artificial Neural Networks, URL: <https://www.theprojectspot.com/tutorial-post/introduction-to-artificial-neural-networks-part-1/7introduction-to-artificial-neural-networks-part-1/7>, Son Erişim Tarihi: 21.05.2021
75. LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436-444.
76. İnternet: Recurrent Neural Network: Maths In 5 Minute. <https://vidishajitani.medium.com/recurrent-neural-network-maths-69214e4d69e1>, Son Erişim Tarihi: 21.05.2021
77. Liu, S., Yang, N., Li, M., and Zhou, M. (2015). *A recursive recurrent neural network for statistical machine translation*. Proceedings of The 52nd Annual Meeting of the Association for Computational Linguistics, Baltimore, 1491-1500.
78. Medsker, L., and Jain, L. C. (1999). *Recurrent neural networks: design and applications* (First Edition). ABD: CRC Press, 2-6.
79. Bengio, Y., Simard, P., and Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, 5(2), 157-166.
80. Williams, R. J., and Zipser, D. (1995). Gradient-based learning algorithms for recurrent. *Backpropagation: Theory, architectures, and applications*, 433, 17.
81. Hochreiter, S. (1998). The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 6(02), 107-116.
82. Pascanu, R., Mikolov, T., and Bengio, Y. (2013, May). *On the difficulty of training recurrent neural networks*. International Conference on Machine Learning, Atlanta, 1310-1318.
83. Martens, J., and Sutskever, I. (2011, January). *Learning recurrent neural networks with hessian-free optimization*. International Conference on Machine Learning, Washington, 1033-1040.
84. Chung, J., Gulcehre, C., Cho, K., and Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling, *arXiv:1412.3555*.
85. Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation, *arXiv:1406.1078*.
86. Hochreiter, S., and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8), 1735-1780.

87. Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. *Advances in Neural Information Processing Systems*, 27, 3104-3112.
88. Graves, A., Mohamed, A. R., and Hinton, G. (2013, May). *Speech recognition with deep recurrent neural networks*. IEEE International Conference on Acoustics, Speech and Signal Processing, Vancouver, 6645-6649.
89. Sak, H., Senior, A. W., and Beaufays, F. (2014). *Long short-term memory recurrent neural network architectures for large scale acoustic modeling*. Fifteenth Annual Conference of the International Speech Communication Association, Singapore, 338-342.
90. İnternet:LSTM Architecture, Understanding LSTM Networks, URL: <http://colah.github.io/posts/2015-08-Understanding-LSTMs/>, Son Erişim Tarihi: 26.04.2021
91. Wolf, T., Chaumond, J., Debut, L., Sanh, V., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., Platen, P.v., Ma, C., Jernite, Y., Plu, J., Xu, C., Scao, T.L., Gugger, S., Drame, M., Lhoest, Q. and Rush, A. M. (2020, October). *Transformers: State-of-the-art natural language processing*. Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, Online, 38-45.
92. İnternet: Transformers. How Transformers Work, URL: <https://towardsdatascience.com/transformers-141e32e69591>, Son Erişim Tarihi: 10.05.2021
93. De Coster, M., Van Herreweghe, M., and Dambre, J. (2020). *Sign language recognition with transformer networks*. 12th International Conference on Language Resources and Evaluation, Marseille, 6018 – 6024.
94. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1), 1929-1958.
95. Tran, Q., and Firl, J. (2014, June). *Online maneuver recognition and multimodal trajectory prediction for intersection assistance using non-parametric regression*. IEEE Intelligent Vehicles Symposium Proceedings, Dearborn, 918-923.
96. Nwankpa, C., Ijomah, W., Gachagan, A., and Marshall, S. (2018). *Activation functions: Comparison of trends in practice and research for deep learning*. 2nd International Conference on Computational Sciences and Technology, Suzhou, 124-133.
97. İnternet: Sigmoid. URL:https://en.wikipedia.org/wiki/Sigmoid_function, Son Erişim Tarihi: 10.05.2021
98. İnternet: Hiperbolik Fonksiyon, URL: https://tr.wikipedia.org/wiki/Hiperbolik_fonksiyon, Son Erişim Tarihi: 10.05.2021

99. Glorot, X., Bordes, A., and Bengio, Y. (2011, June). *Deep sparse rectifier neural networks*. Proceedings of the 14th International Conference on Artificial Intelligence and Statistics, Lauderdale, 315-323.
100. Gao, B., and Pavel, L. (2017). On the properties of the softmax function with application in game theory and reinforcement learning, *arXiv:1704.00805*.
101. İnternet: Mean Squared Error. URL: https://en.wikipedia.org/wiki/Mean_squared_error, Son Erişim Tarihi: 10.05.2021
102. İnternet: Cross Entropy, URL: https://en.wikipedia.org/wiki/Cross_entropy, Son Erişim Tarihi: 10.05.2021
103. Golik, P., Doetsch, P., and Ney, H. (2013, August). Cross-entropy vs. squared error training: a theoretical and experimental comparison. *Interspeech*, 13, 1756-1760.
104. Bottou, L. (1991). Stochastic gradient learning in neural networks. *Proceedings of Neuro-Nimes*, 91(8), 12.
105. Kingma, D. P., and Ba, J. (2014). Adam: A method for stochastic optimization, *arXiv:1412.6980*.
106. Yahmed, Y. B., Bakar, A. A., Hamdan, A. R., Ahmed, A., and Abdullah, S. M. S. (2015). Adaptive sliding window algorithm for weather data segmentation. *Journal of Theoretical and Applied Information Technology*, 80(2), 322.
107. Vafaeipour, M., Rahbari, O., Rosen, M. A., Fazelpour, F., and Ansarirad, P. (2014). Application of sliding window technique for prediction of wind velocity time series. *International Journal of Energy and Environmental Engineering*, 5(2), 1-7.
108. İnternet: Confusion Matrix, URL: https://en.wikipedia.org/wiki/Confusion_matrix, Son Erişim Tarihi: 23.05.2021
109. Hossin, M., and Sulaiman, M. N. (2015). A review on evaluation metrics for data classification evaluations. *International Journal of Data Mining and Knowledge Management Process*, 5(2), 1.
110. Haldar, R., and Mukhopadhyay, D. (2011). Levenshtein distance technique in dictionary lookup methods: An improved approach, *arXiv:1101.1232*.
111. Levenshtein, V. I. (1966, February). Binary codes capable of correcting deletions, insertions, and reversals. *Soviet physics doklady*, 10 (8), 707-710.
112. Yang, Q., and Wu, X. (2006). 10 challenging problems in data mining research. *International Journal of Information Technology and Decision Making*, 5(04), 597-604.
113. İnternet: JIGSAWS.Downloads, URL: https://cirl.lcsr.jhu.edu/research/hmm/datasets/jigsaws_release/, Son Erişim Tarihi: 20.05.2021

114. Masum, S., Liu, Y., and Chiverton, J. (2018, June). *Multi-step time series forecasting of electric load using machine learning models*. International Conference on Artificial Intelligence and Soft Computing, Bangalore, 148-159.
115. Lea, C., Vidal, R., & Hager, G. D. (2016, May). *Learning convolutional action primitives for fine-grained action recognition*. IEEE international conference on robotics and automation (ICRA), Stockholm, 1642-1649.
116. İnternet: Transformer for metamodels, URL:
<https://timeseriestransformer.readthedocs.io/en/latest/README.html#adaptations-for-time-series>, Son Erişim Tarihi: 10.06.2021





GAZİ GELECEKTİR...