T.C.
ERCİYES ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI

ATEŞLEME SİNİR AĞI MİMARİLERİNDE ALGORİTMİK PERFORMANS İNCELEMESİ

Hazırlayan

Obeid Mohamednoor Obeid ABDALLA

Danışman
Prof. Dr. Recai KILIÇ

Yüksek Lisans Tezi

Temmuz 2024
KAYSERİ

T.C.
ERCİYES ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI

ATEŞLEME SİNİR AĞI MİMARİLERİNDE ALGORİTMİK PERFORMANS İNCELEMESİ

(Yüksek Lisans Tezi)

Hazırlayan
Obeid Mohamednoor Obeid ABDALLA

Danışman
Prof. Dr. Recai KILIÇ

Bu çalışma, Erciyes Üniversitesi Bilimsel Araştırma Projeleri Birimi tarafından FBA-2024-13580 kodlu proje ile desteklenmiştir.

Temmuz 2024
KAYSERİ

BİLİMSEL ETİĞE UYGUNLUK

Bu çalışmadaki tüm bilgilerin, akademik ve etik kurallara uygun bir şekilde elde edildiğini beyan ederim. Aynı zamanda bu kural ve davranışların gerektirdiği gibi, bu çalışmanın özünde olmayan tüm materyal ve sonuçları tam olarak aktardığımı ve referans gösterdiğimi belirtirim.

Obeid Moamednoor Obeid ABDALLA

“Ateşleme Sinir Ağı Mimarilerinde Algoritmik Performans İncelemesi” adlı Yüksek Lisans Tezi, Erciyes Üniversitesi Lisansüstü Tez Önerisi ve Tez Yazma Yönergesi’ ne uygun olarak hazırlanmıştır.

Hazırlayan	**Danışman**
Obeid Mohamednoor Obeid ABDALLA	Prof. Dr. Recai KILIÇ

Elektrik-Elektronik Mühendisliği ABD Başkanı

Prof. Dr. Enis GÜNAY

TEŞEKKÜR

Tez çalışmam süresince planlamadan yürütülmesine ve sonuçlandırılmasına kadar her aşamada yakın ilgi ve yardımlarını esirgemeyen, farklı bakış açıları ve bilimsel katkılarıyla beni aydınlatan, değerli hocalarım Prof. Dr. Recai KILIÇ ve Dr. Öğr. Üyesi Nimet KORKMAZ'a en içten teşekkürlerimi sunarım. Bu süreçte verdiği destek ve yönlendirmeler için minnettarım.

Lisans eğitimimden bugüne kadar yaşadığım zorluklarda beni motive eden ve bana yardımcı olan Arş. Gör. Ahmet Yasin BARAN hocama da ayrıca teşekkür etmek isterim.

Bu tez çalışmasına maddi destek sağlayan Erciyes Üniversitesi Bilimsel Araştırma Projeleri Birimi'ne (Proje No: FBA-2024-13580) ve sağladıkları burs ve destekler için teşekkürlerimi sunarım.

Hayatım boyunca maddi ve manevi desteklerini benden esirgemeyen ve bana her zaman yardımcı olan herkese sonsuz teşekkürlerimi sunarım. Eğitim hayatım boyunca bana verdikleri destek ve emekleri için minnettarım. Umarım beklentilerinizi karşılayabilir ve sizi mutlu edebilirim.

Obeid Mohamednoor Obeid ABDALLA

Temmuz 2024, KAYSERİ

ATEŞLEME SİNİR AĞI MİMARİLERİNDE ALGORİTMİK PERFORMANS İNCELEMESİ

Obeid Mohamednoor Obeid ABDALLA
Erciyes Üniversitesi, Fen Bilimleri Enstitüsü
Yüksek Lisans Tezi, Temmuz 2024
Danışman: Prof. Dr. Recai KILIÇ

ÖZET

Geleneksel programlama yöntemleri ile yapılan problem çözümlerinin karmaşıklığını azaltmak için insan beynindeki bilgi sürecinden esinlenen çeşitli nöral ağ teknikleri geliştirilmiştir. Bu teknikler arasında birinci nesil ve ikinci nesil yapay sinir ağlarının (YSA) devamı olarak biyolojik anlamlılığa sahip üçüncü nesil YSA'ları olarak Ateşlemeli Sinir Ağlar (Spiking Neural Networks-SNN'ler) tanımlanmaktadır. SNN'ler, zar potansiyeli bir eşiği aştığında ateşlenen nöron katmanlarından oluşan biyolojik nöronların iletişim kurallarını taklit eder. SNN ağ yapılarının başarım odaklı performans değerlerini artırmak için literatürde yenilikçi çalışmalara ihtiyaç duyulmaktadır. Bu tez çalışmasında denetimli ve denetimsiz öğrenme paradigmaları kullanılarak çeşitli görevleri gerçekleştirebilen SNN ağ mimarilerinin tasarlanması ve bu ağ mimarilerine eğitim için farklı algoritmaların uygulanması ve performans değerlendirilmesi amaçlanmaktadır. Bu bağlamda bu tez çalışmasında; biyolojik anlamlılığı yüksek ve az değişkenle tanımlanan FitzHugh-Nagumo nöron modeli (FHN) ile farklı katmanlardan oluşan SNN yapıları oluşturulmuştur. Oluşturulan ağları eğitmek için denetimsiz öğrenme paradigması altında Ateşleme Zamanlamaya Bağlı Plastisite (STDP) algoritmasıyla XOR uygulaması yapılmıştır. Denetimli öğrenme paradigması altında da Multi-STIP ve Multi-ReSuMe algoritmaları kullanılarak İris çiçeği ve Wisconsin göğüs kanseri (WBC) sınıflandırılması gerçekleştirilmiştir. Nümerik simülasyonlardan edinilen veriler, literatürde bulunan veriler ile karşılaştırarak kıyaslaması yapılmıştır. Sonuçlara göre, geliştirilen SNN'lerin kullanılan nöron sayısına göre yüksek doğruluk elde ettiğini görülmüştür. Multi-ReSuMe algoritması İris çiçeği veri setinde Multi-STIP algoritmasından daha iyi performans gösterirken, Multi-STIP algoritması Wisconsin göğüs kanseri veri setinde daha iyi performans sergilemiştir. Bu sonuçlar, SNN'nin güç tüketimini azaltmasında ve performansını artırmasında SNN optimizasyonunun önemini göstermektedir.

Anahtar Kelimeler: Ateşlemeli sinir ağlar (spiking neural networks-SNN'ler),Makine Öğrenmesi, Denetimli öğrenme, Denetimsiz öğrenme,

ALGORITHMIC PERFORMANCE REVIEW IN SPIKING NEURAL NETWORK ARCHITECTURES

Obeid Mohamednoor Obeid ABDALLA
Erciyes University, Graduate School of Natural and Applied Sciences
Master Thesis, July 2024
Supervisor: Prof. Dr. Recai KILIÇ

ABSTRACT

Various neural network techniques inspired by the information process of the human brain have been developed to simplify problem-solving beyond traditional programming methods. Among these techniques, Spiking Neural Networks (SNNs) are defined as the third generation of artificial neural networks with good biological significance. They are considered a continuation of the first-generation and second-generation artificial neural networks (ANNs). SNNs mimic the communication rules of biological neurons, consisting of layers of neurons that fire when the membrane potential exceeds a threshold. Innovative studies are needed in the literature to increase the design-oriented performance values of SNN network structures. This thesis aims to design SNN architectures capable of performing various tasks using supervised and unsupervised learning paradigms and to evaluate the performance of different training algorithms. In this context, SNN structures with different layers were constructed using the FitzHugh-Nagumo (FHN) neuron model which is defined with high biological significance and few variables. The XOR problem was addressed using the Spike Time Dependent Plasticity (STDP) algorithm under the unsupervised learning paradigm. The iris flower and Wisconsin breast cancer classification problems were conducted using the Multi-STIP and Multi-ReSuMe algorithms under the supervised learning paradigm. The data obtained from the numerical simulations were compared with the data found in the literature. According to the results, the developed SNNs achieved high accuracy in relation to the low number of neurons used. The developed SNNs were found to achieve high accuracy depending on the number of neurons used. While the Multi-ReSuMe algorithm outperformed the Multi-STIP algorithm on the iris flower dataset, the Multi-STIP algorithm performed better on the Wisconsin breast cancer dataset. These results indicate the importance of optimization of SNN which lowers power consumption of SNN and enhances its performance.

Keywords: Spiking neural network, Machine learning, Supervised learning, Unsupervised learning.

İÇİNDEKİLER

ATEŞLEME SİNİR AĞI MİMARİLERİNDE ALGORİTMİK PERFORMANS İNCELEMESİ

YÖNERGEYE UYGUNLUK....iii
KABUL VE ONAY....iv
TEŞEKKÜR....v
ÖZET....vi
ABSTRACT....vii
İÇİNDEKİLER....viii
KISALTMALAR....x
TABLOLAR LİSTESİ....xi
ŞEKİLLER LİSTESİ....xii
GİRİŞ....1

1. BÖLÜM

ATEŞLEMELİ SİNİR AĞ MİMARİLERİ

1.1. Biyolojik Nöron Fizyolojisi....9
1.2. Ateşlemeli Nöronlar....11
1.2.1. Nöral Ağırlıklar....15
1.3. Ateşlemeli Ağ Mimarileri ve Giriş Sinyallerin Kodlanması....21

2. BÖLÜM

SNN'LERDE ÖĞRENME STRATEJİLERİ VE ALGORİTMALAR

2.1. Denetimsiz Öğrenme....29

2.1.1. Denetimsiz Öğrenme Paradigmasına Dayalı Örüntü Sınıflandırma Uygulaması 33

2.2. Denetimli Öğrenme 37

2.2.1. Multi-STIP Algoritması 38

2.2.2. Multi-ReSuMe Algoritması 39

3. BÖLÜM

SNN'LERDE DENETİMLİ ÖĞRENME ALGORİTMALARININ PERFORMANSI

3.1. Iris Veri Setine Dayalı Uygulama 42

3.2. WBC Veri Setine Dayalı Uygulama 46

3.3. FHN Tabanlı Farklı Ateşlemeli Sinir Ağı Mimarilerinin Performans İncelemesi 51

4. BÖLÜM

TARTIŞMA-SONUÇ VE ÖNERİLER

KAYNAKÇA 58

ÖZGEÇMİŞ 67

KISALTMALAR

YSA	:	Yapay Sinir Ağları
DSA	:	Derin Sinir Ağları
DNN	:	Derin Sinir Ağları
MLP	:	Multi-Layer Perceptron
ASN	:	Ateşleme Sinir Ağlar
SNN	:	Ateşleme Sinir Ağlar
HH	:	Hodgkin–Huxley
LIF	:	Leaky Integrate-And-Fire
ADEX	:	Adaptive Exponential İntegrate-And-Fire
ML	:	Morris-Lecar
HR	:	Hindmarsh-Rose
FHN	:	Fitzhugh-Nagumo
IF	:	Entegrasyon ve Ateşleme
RNN	:	Yinelemeli Sinir Ağları
SCNN	:	Ateşlemeli Evrişimsel Sinir Ağı
WBC	:	Wisconsin Göğüs Kanseri
SGD	:	Stokastik Gradyan Düşüş
BPTT	:	Zaman İçinde Geri Yayılım
STDP	:	Ateşleme Zamanlamaya Bağlı Plastisite
mHR	:	memristif Hindmarsh-Rose
NMSE	:	Normalize Edilmiş Ortalama Senkronizasyon Hata
WC	:	Wilson Cowan
SRM	:	Ateşlemeye Tepki Model
TTFS	:	İlk Ateşleme Zamanı
ML	:	Makine Öğrenimi
SWAT	:	Sinaptik Ağırlık İlişkilendirme Eğitimi
WTA	:	Kazanan Hepsini Alır

TABLOLAR LİSTESİ

Tablo 1.1 Farklı ateşlemeli nöron modellerinin karşılaştırılması 12
Tablo 2.1 Literatürde bulunan farklı algoritmalara dayalı ateşlemeli sinir ağlarının karşılaştırılması 28
Tablo 2.2 2x2 SNN örüntü sınıflandırmasında kullanılan parametreler 35
Tablo 3.1 Fisher's Iris veri seti 42
Tablo 3.2 İris probleminde kullanılan algoritma parametreleri 43
Tablo 3.3 WBC veri seti 47
Tablo 3.4 WBC probleminde kullanılan algoritma parametreleri 48
Tablo 3.5 Farklı mimariler kullanılarak Iris veri setinde elde edilen doğruluklar 51
Tablo 3.6 Farklı mimariler kullanılarak WBC veri setinde elde edilen doğruluklar 52
Tablo 3.7 Test grubunda elde edilen en yüksek doğruluklar 53

ŞEKİLLER LİSTESİ

Şekil a Sinir ağların nesilleri 2
Şekil 1.1 Biyolojik nöronun şematiği 9
Şekil 1.2 Aksiyon potansiyel görseli 10
Şekil 1.3 Bir ağda ateşlemeli nöron etkileşimi kavramı 12
Şekil 1.4 FitzHugh-Nagumo nöron modelinin I_{ext}=3,5 genlikli bir darbe sinyali ile uyarıldığında verdiği dinamikleri 14
Şekil 1.5 FitzHugh-Nagumo nöron modelinin I_{ext}=3,5 genlikli birim adım sinyali ile uyarıldığında verdiği dinamikleri 15
Şekil 1.6 Ateşlemeli nöron model öğeleri 16
Şekil 1.7 Üç mHR nöron tabanlı ateşlemeli sinir ağı 17
Şekil 1.8 Sabit başlangıç koşulları için ağın ateşleme dinamikleri [x_1;0;0; x_2; 0; 0; x_3; 0; 0; -2;-2;-2], (a-b) membran potansiyeli zaman serisi, (c-d) zaman içinde x_1-x_2 (yeşil çizgi) ve x_1-x_3 (kırmızı çizgi) arasındaki mutlak fark, (e-f) senkronizasyon faz portreleri (a, c) k = 0,2, (b, d) k = 3. 19
Şekil 1.9 Normalize edilmiş ortalama senkronizasyon hata haritası mHR sinir ağı. 20
Şekil 1.10 Sinir ağlarının tasarımları, (a) tek katmanlı ileri beslemeli sinir ağı, (b) tekrarlayan sinir ağı, (c) çok katmanlı ileri beslemeli sinir ağı (çok çıkışlı) (d) kısmen bağlı çok katmanlı ileri beslemeli sinir ağı, (e) çok bağlantılı ileri beslemeli sinir ağı (f) çok katmanlı ileri beslemeli sinir ağı (tek çıkışlı). 22
Şekil 1.11 Giriş sinyallerin kodlanma teknikleri, (a) oransal kodlanma (b) zamansal kodlanma 24
Şekil 2.1 Makine öğrenimi türleri ve kullanımları 26
Şekil 2.2 Denetimsiz öğrenme 30
Şekil 2.3 Ateşleme zamanlamaya bağlı plastisite STDP ağırlık değişimi 32
Şekil 2.4 XOR problemi ve onu çözmek için kullanılan sinir ağı yapısı 34
Şekil 2.5 Ateşlemeli sinir ağının dinamiği, (a),(b) giriş nöronlarının membran potansiyeli, (c),(d) çıkış nöronlarının membran potansiyeli 36
Şekil 2.6 Zamanla nöral ağırlıkların gelişimi 36
Şekil 2.7 Denetimli öğrenme 37
Şekil 3.1 Iris veri setinde kullanılan ağ mimarisi 43

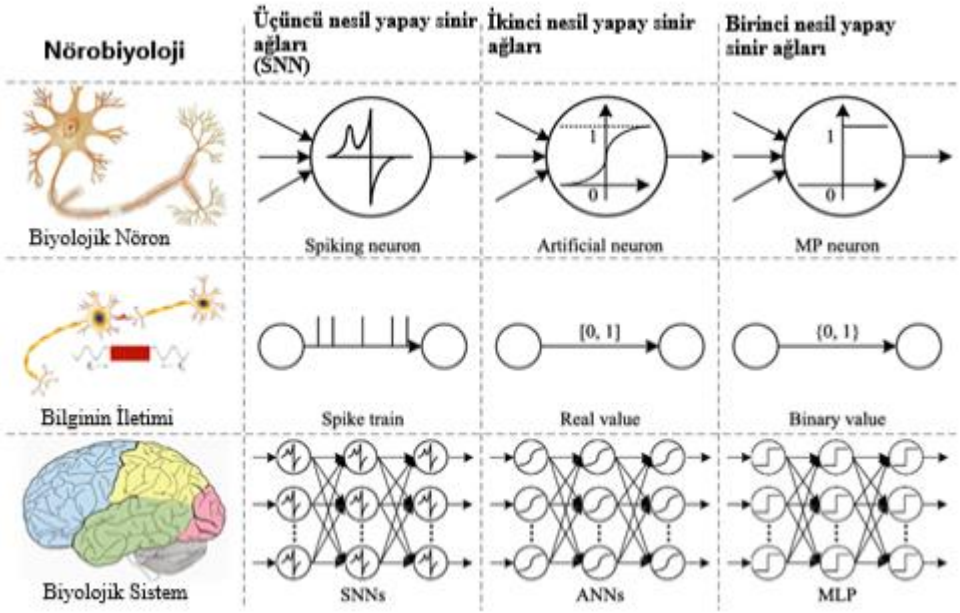
Şekil 3.2 FHN (4×12×1) ateşlemeli sinir ağına ilişkin Fisher Iris veri setine ait eğitim grubun sonuçları: (a_i),(b_i) zaman içindeki ağırlık gelişimidir, (a_{ii}),(b_{ii}) [20700-22500] ms aralığında çıkış nöronunun gerçek sinyali u[17] ve istenen sinyali u[18], (a_{iii}),(b_{iii}) hata fonksiyonu gelişimi. (a_i)-(a_{iii}) Multi-ReSuMe sonuçları (b_i)-(b_{iii}) Multi-STIP sonuçları 44
Şekil 3.3 FHN (4×12×1) ateşlemeli sinir ağında Fisher Iris veri setine ait test grubun sonuçları (a_i),(b_i) çıkış nöronunun membran potansiyeli (gerçek), (a_{ii}),(b_{ii}) hata fonksiyonu gelişimi (a_i)-(a_{ii}) Multi-ReSuMe sonuçları (b_i)-(b_{ii}) Multi-STIP sonuçları 45
Şekil 3.4 WBC veri setinde kullanılan ağ mimarisi 48
Şekil 3.5 FHN (9×18×1)ateşlemeli sinir ağına ilişkin WBC veri setine ait eğitim grubun sonuçları: (a_i),(b_i) zaman içindeki ağırlık gelişimidir, (a_{ii}),(b_{ii}) [99600-102600] ms aralığında çıkış nöronunun gerçek sinyali u[28] ve istenen sinyali u[29], (a_{iii}),(b_{iii}) hata fonksiyonu gelişimi. (a_i)-(a_{iii}) Multi-ReSuMe sonuçları (b_i)-(b_{iii}) Multi-STIP sonuçları 49
Şekil 3.6 FHN (9×18×1) ateşlemeli sinir ağında WBC veri setine ait test grubun sonuçları (a_i),(b_i) çıkış nöronunun membran potansiyeli (gerçek), (a_{ii}),(b_{ii}) hata fonksiyonu gelişimi (a_i)-(a_{ii}) Multi-ReSuMe sonuçları (b_i)-(b_{ii}) Multi-STIP sonuçları 50
Şekil 3.7 FHN tabanlı farklı ateşlemeli sinir ağı mimarilerine göre sonuçların doğruluğu: (a) İris veri seti, (b) WBC veri seti. 52

GİRİŞ

İnsan beyni birbiriyle bağlantılı birden fazla biyolojik nöron ağını kullanarak bilgiyi işleme ve depolama yeteneğine sahiptir. Bu yapıdan esinlenerek geliştirilen Yapay Sinir Ağları (YSA), problem çözümlemelerinin karmaşıklığını azaltmak için 1980'lerden beri yaygın olarak kullanılmaktadır [1, 2]. YSA'lar canlı vücudundaki karar verme mekanizmalarına benzer davranış gösteren basit modeller ile oluşturulur. Sinir ağları, nöron adı verilen birbirine bağlı düğümlerden oluşur. Sinir ağındaki nöronlar katmanlar halinde düzenlenir ve her katman belirli bir işlevi yerine getirmektedir. Yapay Sinir Ağları oluşturuldukları katman yapılarına ve bu katmanlardaki yapıların eğitilme tekniklerine göre birinci, ikinci ve üçüncü nesil sinir ağları olmak üzere üç ayrı şekilde incelenmektedir. Birinci nesil yapay sinir ağları, biyolojik nöronun basit bir modeli olan McCulloch-Pitts nöron modelinden esinlenmiştir [1, 3]. Birinci nesil yapay sinir ağları birbirine bağlı bir veya daha fazla nöron katmanından oluşan basit ağlardır. Katmanlar çok sayıda düğümden veya nöronlardan oluşmaktadır. Her düğüm, iki seviyeli aktivasyon fonksiyonu olarak bilinen belirli bir çıkış fonksiyonu aracılığıyla çalışmaktadır. Birinci nesil yapay sinir ağları basit sınıflandırma problemlerini çözmek için kullanılabilir, ancak daha karmaşık ilişkileri öğrenemezler. Bunun nedeni birinci nesil yapay sinir ağlarının yalnızca girdi ve çıktı değişkenleri arasındaki doğrusal ilişkileri öğrenebilmesidir. Son yıllarda YSA'daki önemli gelişmelerle birlikte birinci nesil sinir ağları yerini ikinci nesil derin sinir ağlarına (DSA)'lar bırakmıştır. İkinci nesil sinir ağları, Derin Sinir Ağları (DNN) (Deep Neural Networks) olarak bilinen çok katmanlı algılayıcılardır. DNN'ler, seviyeli aktivasyon fonksiyonlarının kullanımını içerir ve bu durum onları birinci nesil sinir ağlarından daha karmaşık hale getirmektedir[4, 5]. Seviyeli aktivasyon fonksiyonlarının kullanılması girdi ve çıktı değişkenleri arasındaki daha karmaşık ilişkileri öğrenmelerine olanak tanır. Multi Layer Perceptron (MLP)'ler görüntü sınıflandırma, konuşma yapılarının tanınması ve doğal dil işleme gibi çeşitli görevler için kullanılmaktadır. Bu alanlardaki dikkat çeken görevler arasında desen sınıflandırma,

görüntü segmentasyonu, metin çevirisi ve soru cevaplama yer almaktadır[6]. DNN'lerin çıktısı ağın bağlantılarına, ağırlık değerlerine ve kullanılan aktivasyon fonksiyonlarına bağlı olarak değişir. Bu durum YSA'ların bilgiyi işlemede ve yorumlamada karmaşıklığını artırmaktadır.

Son yıllarda derin öğrenme modelleri ve çok miktardaki eğitim verisiyle elde edilen önemli performans sonuçlarına rağmen, bu sistemlerin geniş çapta benimsenmesini engelleyen iki temel zorluk bulunmaktadır: Birincisi, DNN'lerin genellikle yüksek güç tüketimi gerektirmesidir[7]. Bu güç tüketiminin sebebi Şekil a'da görüldüğü gibi, DNN katmanlardaki nöronların iletişim kurarken sürekli zamanlı değerleri kullanmasıdır. İkinci öne çıkan zorluk, gürültü artışıyla birlikte sistemlerin performansının önemli ölçüde düşmesidir. Yapay sinir ağları (YSA)'lar ve derin sinir ağları (DSA)'lar, canlı vücudundaki karar mekanizmalarından esinlenmiş olsa da yapısal olarak, nöral hesaplamaları ve öğrenme kuralları açısından biyolojik sinir ağlarından temel farklılıklar gösterirler[8]. Bu farklılık, mevcut YSA'ların sınırlamalarını aşma potansiyeline sahip olan üçüncü nesil sinir ağları olarak kabul edilen ateşleme sinir ağlarının (ASA)'lar geliştirilmesine yol açmıştır[9, 10].



Şekil a Sinir ağların nesilleri

Üçüncü nesil sinir ağları, Ateşleme Sinir Ağlar (SNN'ler) (Spiking Neural Networks) [11], biyolojik nöral yapılar ile beyin benzeri hesaplamalar (neurocomputing) arasındaki

boşluğu kapatmayı amaçlamaktadır [12]. SNN'lerde nöronlar ateşlenen sinyalleri ateşleme şeklinde iletmektedir. Ateşlemelerin ani yükselişlerin zamanlaması bilgi işleme açısından önemlidir. Çıkış katmanlarındaki nöronların çıktısı, önceki nesillerde olduğu gibi bir aktivasyon fonksiyonu yerine bir dizi diferansiyel denklem tarafından belirlenmektedir. Birinci ve ikinci nesil bilinen yapay sinir ağlar (YSA)'lar ve SNN'ler arasındaki temel fark, modellenen nöronal yapıların kendi aralarındaki iletişim kurma biçimleridir. YSA'larda sürekli değerler kullanılırken, SNN'ler ateşlenmeleri kullanmaktadır. SNN'ler ayrıca YSA'lardan daha az enerji harcamaktadır. Bunun nedeni, SNN'lerin yalnızca gerektiğinde ateşlenmeleri iletmesi, YSA'ların ise her zaman sürekli değerler iletmesidir. Bu davranış özelliği, SNN'lerin robotları ve otonom araçları gömülü sistemlere entegre ederek kontrol etme potansiyelini göstermektedir. Bununla birlikte, SNN'lerin eğitimi, YSA'ların eğitimi ile kıyasla daha karmaşıktır[13]. Bu karmaşıklığın sebebi, elde edilmesi zor ve maliyetli olabilen etiketli verilere ihtiyaç duyulmasıdır. SNN'ler aynı zamanda ateşlenmelerin zamanlamasına bağlı oldukları için gürültüye karşı YSA'lardan daha duyarlıdırlar. Son biyolojik keşiflerle birlikte ateşlemeler tarafından yönetilen ve ateşlenen nöron modellerine uygun olan sinaptik plastisitenin çeşitli biçimlerini tanıtılmıştır. Biyolojik öğrenme mekanizmaları tam olarak belirsiz kalsa da nöronların ateşlenmesi veya aksiyon potansiyelleri, nöronlar arasındaki bilgi aktarımının birincil yolu olarak kabul edilmektedir. Sinaptik ağırlık ve ateşleme tabanlı öğrenmenin bu biçimleri, ateşleme sinir ağları (SNN'ler) ile uyumlu olmasına karşı geleneksel sinir ağı modelleri için uygun değildir. Bu kapsamda üçüncü nesil sinir ağlarının nörobiyolojiğe daha yakın olması sebebiyle de bu tez çalışmasında da biyolojik anlamlılığı yüksek SNN yapıları tasarlanmıştır. Tasarlanmış 2 farklı SNN yapıları beyindeki biyolojik nöron yapılarının sinapslar ile kurduğu haberleşme kurallarından ilham almaktadır[14]. Tasarlanmış SNN yapılarıyla biyolojik nöronların zengin dinamiklerini yakalama ve zaman, frekans ve faz gibi çeşitli bilgi boyutlarını temsil etme yetenekleriyle gelecek vadeden bir hesaplama paradigması sunulacaktır. Ek olarak tasarlanan SNN yapıları beyin benzeri karmaşık bilgi işlemeyi modelleyebilir, büyük veri hacimlerini işleyebilir ve bilgi gösterimi için ateşleme dizilerini kullanabilirler. Bu durum onları düşük güçlü donanım uygulaması için uygun hale getirmiştir[15].

Ateşleme Sinir Ağları (SNN'ler), nöronal bilgi işlemenin zamansal dinamiklerini yakalamak için uyarlanmış nöron modelleri ile tanımlanır. SNN'ler tipik olarak sinapslar

aracılığıyla birbirine bağlanan ateşlenebilen nöron katmanlarından oluşmaktadır. SNN'lerde iç zar potansiyel eşiğini geçtiğinde ateşleme üreten biyolojik nöronlardan ilham alan nöron modeli yapıları kullanır. Hodgkin–Huxley (HH), Leaky Integrate-and-Fire (LIF), Adaptive exponential integrate-and-fire (ADEX) ve memristif nöron modelleri gibi ateşleme nöron modelleri ateşleme desenleri arasındaki göreli zamanlamayı [16] veya bazı referans sinyallere göre kontrol edilen zamanlamayı modelleyebilir[17, 18].

1952'de literatüre sunulan Hodgkin-Huxley nöron modeli biyolojik bir nöronun elektriksel özelliklerinin araştırılmasında, özellikle aksiyon potansiyelinin üretilmesinde başarılı çıktılar veren detaylı bir modeldir [19]. Bu model, farklı iyon kanallarının membran kapasitesi ve direnci üzerindeki etkilerini birleştirerek, farklı akım uyarıları altında nöronların çeşitli ateşleme olaylarını açıklamaktadır. Hesaplamalı sinirbilimde biyolojik olarak en makul modellerden biri olarak kabul edilen Hodgkin-Huxley modeli, membran potansiyelini ve iyonik geçiş değişkenlerinin aktivasyonunu ve inaktivasyonunu tanımlayan dört sıradan diferansiyel denklemden oluşan karmaşık, doğrusal olmayan bir sistemdir. Hodgkin-Huxley (HH) modeli son derece gerçekçi ve biyofiziksel açıdan doğru olmasına rağmen hesaplama açısından karmaşıktır ve analitik olarak çözülemez. Bu yüzden HH nöron modelinin değişkenlerini azaltmak için birtakım çalışmalar yapılmıştır. Sonuç olarak Morris-Lecar (ML), Hindmarsh-Rose (HR) ve FitzHugh-Nagumo (FHN) modelleri gibi basitleştirilmiş nöron modelleri geliştirilmiştir[20–22]. Bunlara ek olarak Entegre et ve ateşle (IF) modeli (Integrate and Fire) ve Sızdıran entegre et ve ateşle (LIF) modeli gibi biyolojik iyon kanalı davranışını dikkate almadan, membran potansiyeli bir eşiğe ulaşana kadar giriş akımını zaman içinde entegre eden modeller tanıtılmıştır. Bu modeller değişken tanımlanması bakımından basit modeller olmasına karşın, biyolojik anlamlılığı oldukça düşüktür. Üçüncü nesil sinir ağlarının birinci ve ikinci nesil sinir ağlarına göre biyolojik anlamlılığı daha yüksektir. Bununla birlikte, literatürde üçüncü nesil ağ yapılarını kullanarak çözüm üretilmek istenen problemlerdeki ağ yapıları incelendiğinde, hesaplama kolaylığı dikkate alınarak IF ve LIF gibi nöron modelleri kullanımının tercih edildiği görülmektedir [23]. Öte yandan, önceki nesil ağ yapıları ile kıyaslandığında biyolojik anlamlılık kavramını da gözeten bir ağ yapısında biyolojik anlamlılık açısından da beklentileri karşılayabilecek nöron modellerinin kullanımının problem çözümlerindeki performansı etkileyeceği

düşünülmektedir. Bu kapsamda, bu tez çalışmasında biyolojik anlamlılığı yüksek ve az değişkenle tanımlanan FHN nöron modeli ile farklı katmanlardan oluşan SNN yapıları oluşturmak ve bu ağ yapılarının performanslarını gözlemlemek hedeflenmektedir.

SNN'ler genel olarak Tek katmanlı ateşlenebilen sinir ağı (SNN), Çok katmanlı beslemeli SNN, Yinelemeli Sinir Ağları (RNN) (Recurrent Neural Networks) ve Ateşlemeli Evrişimsel Sinir Ağı (SCNN) (Spiking Convolutional Neural Networks) yapılarından oluşturulmaktadır[24–27]. Tek katmanlı ateşleme sinir ağı (SNN) bir giriş katmanı ve bir çıkış katmanından oluşan hiçbir gizli katmanı olmayan bir SNN türüdür. Tek bir sinaptik ağırlık katmanına sahiptir[26]. Bu bağlamda denetimli öğrenmeyi ateşleyen nöronlarla gerçekleştirir. Basit mimarisi nedeniyle tek katmanlı SNN'ler simülasyonlarda yaygın olarak kullanılmaktadır. Buna karşılık, başka bir yaygın SNN mimarisi olan çok katmanlı ileri beslemeli SNN, her nöronun yalnızca önceki katmandakilere bağlandığı hiyerarşik olarak düzenlenmiş nöronlara sahiptir[27]. İlk katman, herhangi bir nöron hesaplaması olmadan giriş verilerini temsil eden giriş katmanından oluşmaktadır. Her bir nöronun ateşleme dizisi, belirli bir problem için giriş verilerini kodlamaktadır. Bu kodlanmış veriler daha sonra bir sonraki katmana aktarılır. Son katman, her bir nöron tarafından üretilen ateşleme dizileriyle ağın çıktısını oluşturduğu çıkış katmanıdır. Giriş ve çıkış katmanları arasında bir veya daha fazla gizli katman bulunabilir. Geri besleme bağlantılarını içeren bir diğer sinir ağı ise Tekrarlayan Sinir Ağları (RNN) olarak bilinmektedir[24]. Bu geri besleme bağlantıları tekrarlayan SNN'lerin dinamik zamansal bilgileri etkili bir şekilde işlemesine olanak tanıyarak zaman içinde değişen karmaşık örüntülerin modellenmelerine olanak tanır. Tekrarlayan SNN'ler doğrusal olmayan hesaplamaları verimli bir şekilde yürütebilmektedir. Deneysel bulgular, bu ağların karmaşık zamana bağlı hesaplamaları gerçekleştirebilmek için ateşleme dinamiklerini optimize ettiğini göstermektedir. Bu SNN yapılarına ek olarak geleneksel CNN'lere (Convolutional Neural Networks) benzer şekilde çalışan Ateşlemeli Evrişimli Sinir Ağı (SCNN) de vardır. Geleneksel CNN'ler genellikle hata geri yayılım yöntemlerini kullanarak çekirdek filtre parametrelerini eğitir. Buna karşılık, SCNN'ler genellikle eğitim sırasında sabit parametrelere sahip el yapımı evrişim çekirdeklerini kullanır. Bu SCNN'ler çeşitli sınıflandırma görevlerinde ve görüntü tanıma görevlerinde başarılı bir performans göstermiştir. Bu SCNN yapısının diğer SNN yapılarından temel farkı uygulanan öğrenme şekilleridir. SNN yapılarında öğrenme şekilleri veya tercih edilen

algoritmalar en önemli unsurlardandır [25]. Öğrenme şekillerine göre yapılar birbirinden ayrılmaktadır. Sinir ağlarında öğrenme, skaler değerli sinaptik ağırlıkları ayarlayarak gerçekleştirilir. Buna dayalı olarak denetimli öğrenme, denetimsiz öğrenme, takviyeli öğrenme ve yarı denetimli öğrenme gibi çeşitli öğrenme paradigmaları geliştirilmiştir [28, 29]. Denetimsiz öğrenme, genellikle kümeleme ve boyutluluk azaltma gibi görevler için kullanılan, açık etiketler olmadan veriler içindeki kalıpları veya yapıyı bulmaya odaklanır. Denetimli öğrenme, ağların görünmeyen girdileri tahmin etmek veya sınıflandırmak için etiketli veriler üzerinde eğitildiği köklü bir yaklaşımdır [30]. Denetimli öğrenme büyük ölçekli etiketlenmemiş veri kümelerinin kullanılabilirliğinden yararlanarak geniş bir yelpazede akış görevleri ve veri kümeleri için temsiller öğrenebilir. Çoğu mevcut görsel tespit sistemleri işaretlenmiş eğitim veri setlerini kullanarak denetimli öğrenmeyi kullanmaktadır [31–34].

Öğrenme algoritmaları da Ateşleme Sinir Ağı (SNN) yapılarının yeteneklerini ve performansını şekillendirmede çok önemli bir rol oynar. Bu algoritmalar SNN'lerin verilerden bilgi edinmesini, gelişmesini ve uyarlamasını sağlar. Özellikle, öğrenme algoritmasının seçimi ağın yeteneğini üç şekilde önemli ölçüde etkiler: i) zaman içinde değişen örüntüleri yakalamak, ii) belirli görevleri gerçekleştirmek ve iii) sinaptik ağırlıkları optimize etmek. SNN'ler biyolojik sistemlerden türetilen farklı algoritma türleri kullanılarak eğitilir. Bu algoritmalar gradyan düşüş algoritmaları, sinaptik plastisite algoritmaları ve ateşleme dizilerine dayalı algoritmaları şekilde özetlenebilir. Bu öğrenme algoritmalarının seçimi SNN yapılarının tüm potansiyelinden yararlanmada kritik öneme sahiptir. Ateşleme Sinir Ağları (SNN'ler) bağlamındaki gradyan düşüş algoritmaları, belirli bir kayıp veya hata fonksiyonunu en aza indirmek için sinaptik ağırlıklar gibi parametrelerini ayarlayarak ağın performansını optimize etmeyi amaçlar[35]. Gradyan düşüş yöntemleri geleneksel sinir ağlarında yaygın olarak kullanılırken, bunları SNN'lere uyarlamak da zordur[36]. Bununla birlikte, son gelişmeler SNN'lerde Stokastik Gradyan Düşüş (SGD) (Stochastic Gradient Descent) ve Zaman İçinde Geri Yayılım (BPTT) (Backpropagation) gibi teknikleri kullanıma sunmuştur. Bu yöntemler, gerçek gradyanlara yaklaşan stokastik gradyanlar sunarak, SNN'lerde denetimli öğrenmeyi mümkün kılar[37–39]. Sinaptik plastisite algoritmaları ise nöronlar arasındaki bağlantıların (sinapsların) gücünün zaman içinde nasıl geliştiğini yöneten Ateşleme Sinir Ağları (SNN'ler) dahil olmak üzere tüm sinir ağlarının temel

bileşenleridir. Bu algoritmalar, ağın aktivitesine ve giriş verilerine göre sinaptik ağırlıkların nasıl ayarlanacağını belirlemektedir. En önemli sinaptik plastisite algoritmalarından birisi olan Ateşleme-Zamanlamaya Bağlı Plastisite (STDP) (Spike Time Dependent Plasticity), sinaptik öncesi ateşlenmelerin sinaptik sonrası ateşlenmelerden önce gelmesi durumunda sinapsları güçlendirirken ters senaryoda ise sinaps bağlantılarını zayıflatarak zaman içinde değişen örüntülerin yakalanmasını sağlamaktadır[40].

Bu tez çalışmasında denetimsiz öğrenme paradigmasının bir türü olan STDP algoritması kullanılarak 4 FHN nöronundan (2 giriş nöronu × 2 çıkış nöronu) oluşan bir SNN yapısı tasarlanmıştır. Bu yapıyla makine öğreniminin en temel uygulamalarından birisi olan EX-OR çalışması test edilmiştir. Ek olarak denetimli öğrenme paradigması altında Multi-STIP ve Multi-ReSuMe algoritmaları kullanılarak iki farklı SNN yapısı ile iki farklı uygulama gerçekleştirilmiştir. İlk uygulamada İngiliz istatistikçi ve biyolog Ronald Fisher'in 1936 yılında literatüre sunduğu İris çiçeği veri seti kullanılmıştır[41]. Veri seti, üç İris türünün (İris Setosa, İris Virginica ve İris Versicolor) her birinden 50 örneği içerir. Her bir örneğin dört özelliği ölçülmüştür: sepallerin ve taç yapraklarının uzunluğu ve genişliği, hepsi santimetre cinsinden ölçülmüştür. Bu veri seti, destek vektör makineleri gibi çeşitli istatistiksel sınıflandırma tekniklerinin değerlendirilmesi için standart bir test durumu haline gelmiştir. Bu çalışmada 17 FHN nöronundan (4 giriş nöronu × 12 gizli katman nöronu × 1 çıkış nöronu) oluşan bir SNN yapısıyla İris çiçeği veri seti kullanılarak üç İris türünün sınıflandırılması gerçekleştirilmiştir. İkinci uygulamada ise 1992 yılında K. P. Bennett tarafından literatüre sunulan Wisconsin göğüs kanseri (WBC) veri seti kullanılmıştır[42]. 28 FHN nöronundan (9 giriş nöronu × 18 gizli katman nöronu × 1 çıkış nöronu) oluşan bir SNN yapısıyla göğüs kanseri tespiti yapılmıştır. Ardından her bir veri sete en optimum nöral ağ ve algoritma belirlemek için farklı gizli katman nöron sayısına sahip nöral ağlar simüle edilmiştir. Iris veri setinde (4 × 8 × 1), (4 × 12 × 1) ve (4 × 24 × 1) nöral ağlar ve Wisconsin göğüs kanseri veri setinde (9 × 12 × 1), (9 × 18 × 1) ve (9 × 27 × 1) nöral ağlar simüle edilip analiz edilmiştir.

Bu tez çalışmasında denetimli ve denetimsiz öğrenme paradigmaları kullanılarak çeşitli görevleri gerçekleştirebilen SNN ağ mimarilerinin tasarlanması ve bu ağ mimarilerine eğitim için uygulanacak farklı algoritmaların performans değerlendirilmesi amaçlanmaktadır. Bu amaç doğrultusunda ilk olarak nöral ağ yapılarının biyolojik

temelleri ve üçüncü nesil sinir ağı olarak literatürde yer alan SNN yapılarının avantajları sunulmuştur. Ek olarak literatürde mevcut nöron modelleri ele alınmıştır. Bu kapsamda öncelikle nöronların ateşleme davranışlarını iki adet diferansiyel denklemle tanımlayabilen FitzHugh-Nagumo Nöron Modeli test edilerek SNN ağına adapte edilmiştir. Bununla birlikte FHN nöronları ile oluşturulan SNN yapıları için sinaptik ağırlık tanımlanarak farklı mimariler tanıtılmıştır. Ardından, tasarlanacak SNN yapıları için Makine öğrenmesinin türleri olan denetimli ve denetimsiz öğrenmeler tanıtılmıştır. Denetimsiz öğrenme paradigması altında STDP algoritması benzetim programlarıyla test edilmiştir. Denetimli öğrenme paradigması altında da Multi-STIP ve Multi-ReSuMe algoritmaları incelenmiştir. Konu ile ilgili temel kavramların incelenmesinden sonra FHN nöron modeliyle birlikte gerekli algoritmalarında nümerik simülasyonları yapılmış ve aşağıdaki konfigürasyonlar nümerik simülasyon sonuçları ile gözlemlenmiştir:

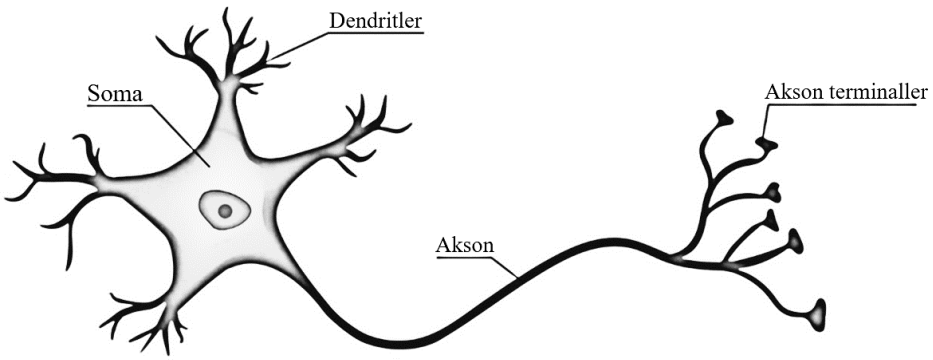
- 4 FHN nöronundan (2 giriş nöronu × 2 çıkış nöronu) oluşan SNN yapısı STDP ve winner takes all (WTA) algoritmalarıyla eğitilerek EX-OR uygulaması yapılmıştır.
- 17 FHN nöronundan (4 giriş nöronu × 12 gizli katman nöronu × 1 çıkış nöronu) oluşan bir SNN yapısıyla İris çiçeği veri seti kullanılarak üç İris türünün sınıflandırılması Multi-STIP ve Multi-ReSuMe algoritmalarıyla gerçekleştirilmiştir.
- 28 FHN nöronundan (9 giriş nöronu × 18 gizli katman nöronu × 1 çıkış nöronu) oluşan bir SNN yapısıyla Wisconsin göğüs kanseri veri seti kullanılarak Multi-STIP ve Multi-ReSuMe algoritmalarıyla gerçekleştirilmiştir.

1. BÖLÜM

ATEŞLEMELİ SİNİR AĞ MİMARİLERİ

1.1. Biyolojik Nöron Fizyolojisi

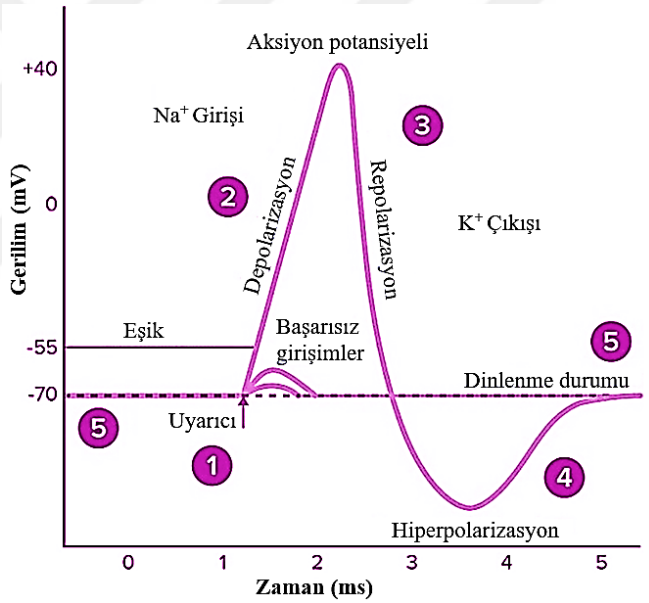
Sinir sisteminin temel yapı taşı olan nöronlar beyindeki haberleşme aşamalarında hayati bir rol oynamaktadır. Nöronlar kendisine gelen sinyali aksiyon potansiyeli adı verilen elektriksel bir sinyal ile diğer nöronlara iletmekle görevlidir. Nöronlar, Şekil1'de görüldüğü üzere soma, dendrit ve akson olmak üzere üç bölümden oluşmaktadır. Dendritler algılayıcı görevi yaparak diğer nöronlardan gelen sinyalleri alır ve somadan çıkarak diğer nöronların aksonlarına bağlanarak sinaptik akımları somaya iletir. Başlangıçta pasif bir yapı olarak düşünülen dendritlerin yapılan çalışmalar ve incelemeler sonunda aksonal süreçlerde yer alan aktif bir mekanizmaya sahip olduğu anlaşılmıştır[43].



Şekil 1.1 Biyolojik nöronun şematiği

Aksiyon potansiyeli aksonlar aracılığıyla sinaps adı verilen bağlantı noktalarında diğer nöronlara iletilir. Bu iletişim kimyasal nörotransmiterler aracılığıyla gerçekleşmektedir. Aksonlar, nöronun sinyallerini vücudun farklı bölgelerine iletebilir. Aksonların uzunluğu hedeflenen bölgenin nörona olan uzaklığına bağlı olarak değişir. Beyindeki herhangi bir

nöronun aksonu, omurilik ve oradan da kaslara kadar uzanabilir. Bu sayede merkezi sinir sistemi kaslar arasında iletişim kurmaktadır. Soma olarak bilinen nöronun çekirdek mekanizması nöronun bilgi işleme merkezi olarak görev yapar ve nöronun tüm işlevlerini yerine getirmesi için hayati bir rol oynar. Bu işlevler, nöronun diğer hücrelerle etkili bir şekilde iletişim kurmasını ve sinir sisteminin genel işleyişini sağlar. Soma, dendritlerden gelen elektriksel sinyalleri işleyerek bir aksiyon potansiyeli oluşturur. Şekil 1.2'de görülen aksiyon potansiyeli, akson boyunca ilerleyerek hedef hücrelere aktarılır. Aksiyon potansiyeli, bir nöronun hücre zarında meydana gelen hızlı ve geçici bir elektriksel değişikliktir. Şekil 1.2'de görüldüğü gibi aksiyon potansiyeli sırasıyla dinlenme modu, depolarizasyon, repolarizasyon, hiperpolarizasyon ve dinlenme potansiyeline dönüş olmak üzere 5 aşamadan oluşmaktadır.



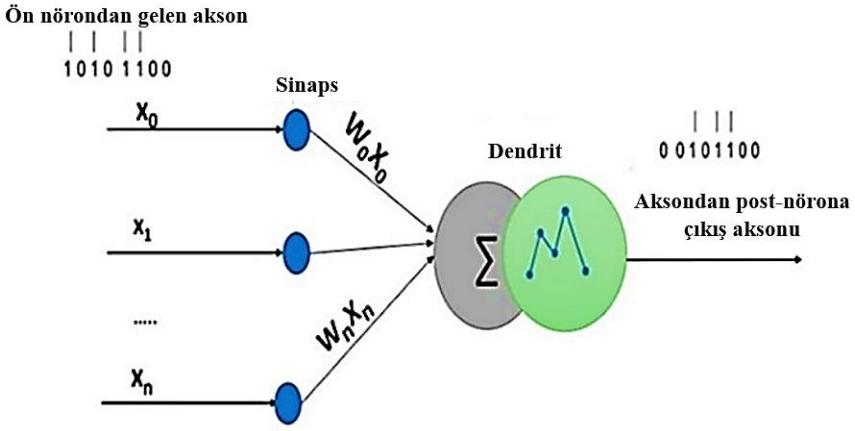
Şekil 1.2 Aksiyon potansiyel görseli

Başlangıçta soma dinlenme halindeyken hücre içindeki sodyum (Na^+), potasyum (K^+) ve kalsiyum (Ca^{+2}) yoğunluklarını düzenleyen iyon pompaları dinlenme potansiyeli olarak bilinen negatif voltajı korur. Dendritik akımlar geldikçe hücreyi depolarize ederler. Somanın voltajı eşik gerilimine ulaştığında voltajla aktifleşen sodyum kanalları açılır ve sodyum iyonlarının hücreye girmesine izin verir. Bu şekilde hücre depolarize olmaya başlar. Bu süreç, sodyum iyonlarının akışı kimyasal gradyanı dengeleyene kadar devam eder, bu da dinlenme voltajıyla karşılaştırıldığında önemli ölçüde daha pozitif bir yüke neden olur. Bu gerçekleşen depolarizasyon aşaması aynı zamanda voltaj kapılı potasyum

kanallarını da aktive ederek potasyum iyonlarının hücreden çıkmasına izin verir. Buna eş zamanlı olarak sodyum kanalları da inaktive olur ve repolarizasyon süreci başlar. Açık potasyum kanalları sonunda hücreyi, dinlenme voltajının altında bir voltaja sahip olan hiperpolarizasyon durumuna getirir. Ani yükselişin ardından sodyum kanalları devre dışı kalır ve potasyum kanalları kısa bir süre açık kalır, bu durum nöronun yeniden ateşlenmesini neredeyse imkânsız hale getirir. Tek bir artış sırasında hücre içindeki iyonik konsantrasyonlardaki değişiklik minimum düzeydedir. Ancak iyon pompaları birçok ani yükselişte uygun sodyum ve potasyum seviyelerini korumak için gereklidir. Ek olarak bazı nöronlarda başka akımlar, özellikle de kalsiyum akımları mevcuttur. Bir aksiyon potansiyeli ile ilişkili hızlı depolarizasyon, somatik voltaj potansiyelini arttırır ve ayrıca somaya yakın akson segmentlerinde depolarizasyona neden olur. Bu depolarizasyon akson kısımlarındaki sodyum kanallarını aktive eder ve ~~bu da~~ akson boyunca daha fazla sodyum kanalını tetikler. Bu zincirleme reaksiyon, akson boyunca ilerleyen bir voltaj dalgası üretir, sonuçta akson terminallerinin yakınındaki sinaptik veziküllere ulaşır ve onların nörotransmiterleri serbest bırakmasına neden olur[44].

1.2. Ateşlemeli Nöronlar

Üçüncü nesil sinir ağları olarak bilinen Ateşlemeli Sinir Ağları (SNN'ler) Yapay Sinir Ağlarının (YSA'lar) standart işleyişinden farklı bir değişime sahiptir. SNN'ler nöral dinamiklerin doğruluğuna öncelik vererek ve olay odaklı tepki verme olanağı sunarak biyolojik sinir aktivitelerini modellerler. Geleneksel YSA'ların aksine, SNN'ler verileri zaman kodlu sinyaller olarak almaktadırlar. Giriş bilgilerini yalnızca gelen ani sinyaller olarak tanımlanan ateşlemeler alındıktan sonra işlerler. Bu da SNN'leri biyolojik olarak daha makul hale getirmektedir. SNN'lerde somalar, dendritler ve sinapslar Şekil 1.3'te görüldüğü gibi modellenir. Sinyalin ulaştığı nöron, önceki katmanda bulunan nöronlardan bir zaman serisi olarak çok sayıda ateşlenme girdisi alır. Daha sonra sinyalin ulaştığı nöron alınan giriş kombinasyonuna bağlı olarak kendine ait bir çıktı sinyali üretir. Şekil 1.3'te görülen temel konsept, gerçekleştirilmesi gereken uygulamaya bağlı olarak birçok farklı biçimde inşa edilmektedir[45].



Şekil 1.3 Bir ağda ateşlemeli nöron etkileşimi kavramı

Ateşleme Sinir Ağları (SNN'ler), sinirsel bilgi işlemenin zamansal dinamiklerini yakalamak için uyarlanmış nöron modellerini içermektedir. SNN'ler tipik olarak sinapslar aracılığıyla birbirine bağlanan ateşlenebilen nöron katmanlarından oluşmaktadır. Biyolojik sinir ağlarına benzeyen SNN'lerin tasarlanmasında iç zar potansiyeli eşiğini geçtiğinde ateşleme üreten biyolojik nöronlardan ilham alan nöron yapıları kullanır. Hodgkin–Huxley (HH), Leaky Integrate-and-Fire (LIF), Adaptive exponential integrate-and-fire (ADEX) ve memristif nöron modelleri gibi ateşleme nöron modelleri ateşleme desenleri arasındaki göreli zamanlamayı [16] veya bazı referans sinyallere göre kontrol edilen zamanlamayı modelleyebilir[17, 18]. Bu nedenle literatürdeki nöron modellerini listeleyen ve karşılaştıran Tablo 1.1 oluşturulmuştur.

Tablo 1.1 Farklı ateşlemeli nöron modellerinin karşılaştırılması

Nöron Model	Değişken Sayısı	Matematiksel İşlem Sayısı	Karmaşıklık	Biyolojik Olabilirlik
HH	4	1200	Çok Yüksek	Evet (Yüksek)
ML	2	600	Yüksek	Evet (Yüksek)
WC	2	180	Orta	Evet (Küçük)
HR	3	120	Orta	Evet (Küçük)
FHN	2	72	Düşük	Evet (Küçük)
SRM	1	50	Düşük	Evet (Küçük)
LIF	1	5	Çok Düşük	Hayır (Sıfır)

Hodgkin ve Huxley tarafından 1952 yılında literatüre sunulan model dört adet durum değişkenini kullanarak nöronun biyofiziksel ve biyolojik olarak anlamlı özelliklerini modelleyebilmektedir[19]. Bu model, farklı iyon kanallarının membran kapasitesi ve direnci üzerindeki etkilerini birleştirerek, farklı akım uyarıları altında nöronların çeşitli ateşleme olaylarını açıklamaktadır. Morris ve Lecar ise midyenin kas liflerinde gözlenen ve kalsiyum (Ca^{+2}) ve potasyum (K+) iletkenliğinden etkilenen bir dizi salınım davranışını kopyalayan bir nöron modeli sunmuştur[20]. Bu model, membran potansiyeli ile iyon kanallarının aktivasyonu arasındaki ilişkiyi iki adet durum değişkeni kullanarak açıklar. Wilson ve Cowan sundukları nöron modelinde (WC) emel uyarıcı ve inhibitör nöron popülasyonları arasındaki etkileşimlerin dinamiklerini tanımlamışlardır[46]. Bu model nöron dinamiklerinin pek çoğunu sergileyebilmektedir ve bu modelin çeşitli genişletilmiş versiyonları nöron popülasyonlarının simüle edilmesinde kullanılmıştır[47, 48]. Hindmarsh ve Rose iki adet durum değişkeni ile tanımlanan FHN nöron modeline, bir durum değişkeni daha ekleyerek yeni bir nöron modeli geliştirmişlerdir[49]. Ateşlemeye tepki model (SRM) nöron modeli (Spike Response Model) ateşlemelerin algılanması ve emisyonu nöronun membran potansiyelini nasıl etkilediğini tasvir etmek için çekirdek fonksiyonlarını kullanan nöron modelidir[50]. Bu model nöron dinamiklerini modellemek için bir adet durum değişkeni kullanır. Bu model basitliği nedeniyle tüm nöron dinamiklerini modelleyememektedir. Benzer olarak (IF) nöron modeli ve (LIF) nöron modeli biyolojik iyon kanalı davranışını dikkate almadan, membran potansiyeli bir eşiğe ulaşana kadar giriş akımını zaman içinde entegre eden modeller tanıtılmıştır[51]. Bu yüzden bu modellerin de biyolojik anlamlılığı SRM nöron modelinde olduğu gibi çok düşüktür.

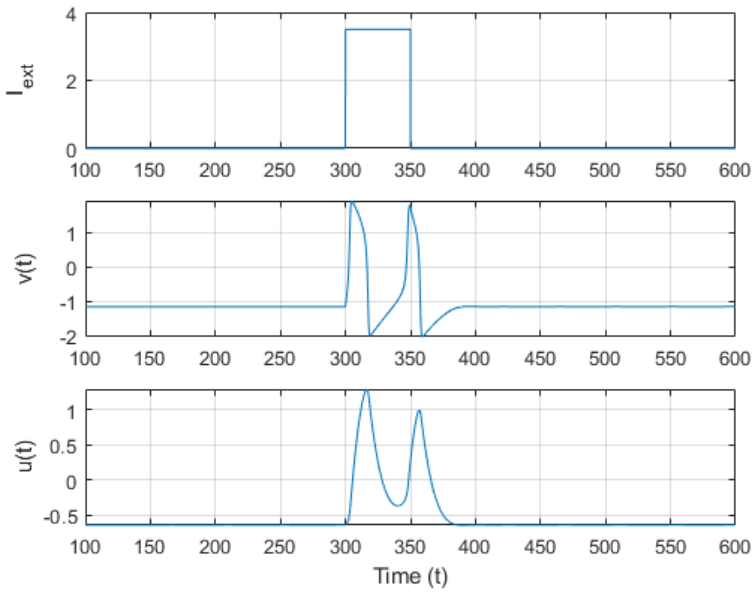
FitzHugh-Nagumo (FHN) modeli, Richard FitzHugh tarafından Hodgkin ve Huxley'nin kalamarın aksondaki aksiyon potansiyeli üretimi modelinin indirgenmesi olarak oluşturulan, nöronal uyarılabilirliğin basitleştirilmiş bir matematiksel temsilidir[22]. Fitzhugh ve Nagumo HH nöron modelini kapsamlı basitleştirmeler kullanarak değişkenlerini azaltmıştır. Bu şekilde iki adet durum değişkenli önceki modellere göre daha az karmaşıklıkla ateşleme davranışı sergileyebilmektedir[22, 52]. FitzHugh, HH nöron modelini daha basit tanımlama açısından Van Der Pol osilatör modeli üzerinde değişiklikler yapılarak, Denklem (1.1) ve (1.2)'le tanımlanan FitzHugh-Nagumo Nöron modeli geliştirmiştir.

Gerçek bir sinire daha iyi benzemek için Van der Pol modeli, mevcut uyarım gibi bir parametredeki değişikliklere yanıt olarak bir davranış sergileyecek şekilde ayarlanmıştır. FHN modelinin temel formu iki bağlantılı doğrusal olmayan adi diferansiyel denklemden oluşur. Bunlardan biri nöronal membran voltajındaki hızlı değişiklikleri tanımlar, diğeri ise sodyum kanalı deaktivasyonu ve potasyum kanalı deaktivasyonunu içeren daha yavaş iyileşme süreçlerini temsil eder. Bu nöron modelini tanımlayan diferansiyel denklem sistemi aşağıdaki gibi ifade edilir:

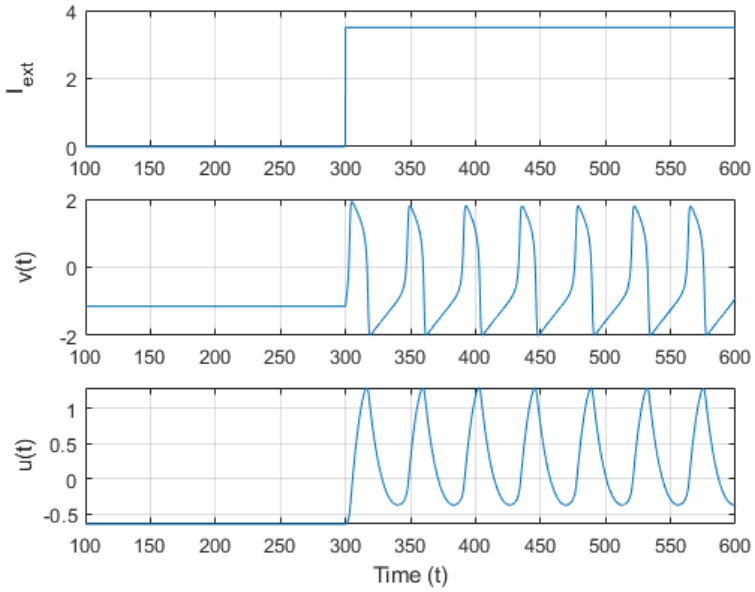
$$\dot{v} = v - \frac{v^3}{3} - u + RI_{ext} \tag{1.1}$$

$$\dot{u} = \frac{1}{\tau}(v - a + bu) \tag{1.2}$$

Burada v membran potansiyelini, u nöronun dinlenme durumuna dönmesine neden olan toparlanma parametresini temsil etmektedir. a, b, R ve τ parametreleri sabit olup $a=0,7$, $b=07$, $R=0,1$ ve $\tau=12,5$ olarak ayarlanmıştır. I_{ext} terimi, harici uyarıyı temsil eden nörona beslenen harici bir akımdır. FHN nöron modelinin dinamik davranışı Şekil 1.4 ve 1.5'te verilmektedir.



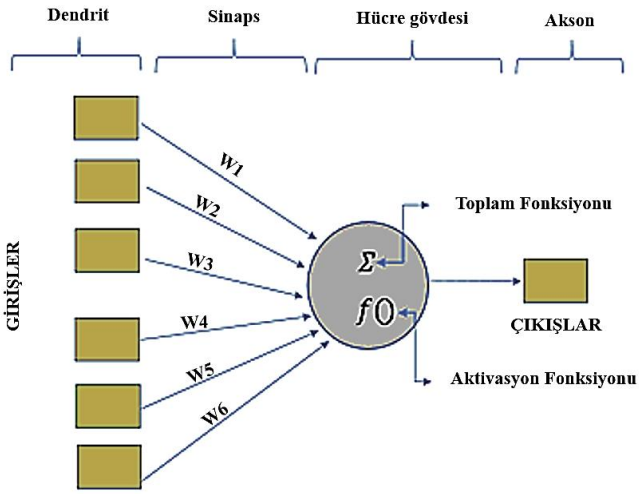
Şekil 1.4 FitzHugh-Nagumo nöron modelinin I_{ext}=3,5 genlikli bir darbe sinyali ile uyarıldığında verdiği dinamikleri



Şekil 1.5 FitzHugh-Nagumo nöron modelinin I_{ext}=3,5 genlikli birim adım sinyali ile uyarıldığında verdiği dinamikleri

1.2.1. Nöral Ağırlıklar

Biyolojik sistemler, öğrenme, hafıza konuları ve sinaptik plastisite üzerine son yıllarda çok sayıda deneysel sonuç kaydedilmiştir. Bu çalışmalardan beyindeki öğrenme ve hafıza sürecinin sinapsların ağırlıklarının değiştirerek gerçekleştirildiği çıkarımı elde edilmiştir. Sinaptik plastisite; nöral yapıların ele alındığı çalışmalarda bilgiyi alan nöronların (post-sinaptik nöronun) cevabının genliğini etkileyen ağırlık ile tanımlanmaktadır. Nöronlar kuplajlı oldukları diğer nöronlarla haberleşme işlemini sinapslar aracılığı ile gerçekleştirmektedir. Nöronların birbiri ile kuplajlanması sırasında sinaptik ağırlık parametresinin değişmesiyle nöronlar arasında bilgi transfer ve işleme sürecinin gerçekleştiği düşünülmektedir. 2000'li yılların başlarına kadar sinirbilimciler sinaptik ağırlığın, bilgiyi gönderen nöronların (pre-sinaptik nöron) ve bilgiyi alan nöronların (post-sinaptik nöronun) ortalama ateşleme hızlarıyla orantılı olarak arttığını sunmuşlardır [53]. Bununla birlikte, araştırmacılar tarafından son zamanlarda sinaptik ağırlığın; nöronların ateşlenme zamanı arasındaki farka bağlı olarak da gerçekleşebileceği görüşü kabul edilmiştir [54].Nöral ağ yapılarının paralel işleme süreci bir "sinaptik ağırlık matrisinin kullanımına dayanmaktadır[55]. Birbirlerine sinaps yapısı ile bağlı olan nöronlar arasında bilgi iletim süreci, bahsedilen sinaptik ağırlık değerlerinin değişimine bağlı olarak gerçekleşmektedir. *(çıkış=ağırlık x giriş).*



Şekil 1.6 Ateşlemeli nöron model öğeleri

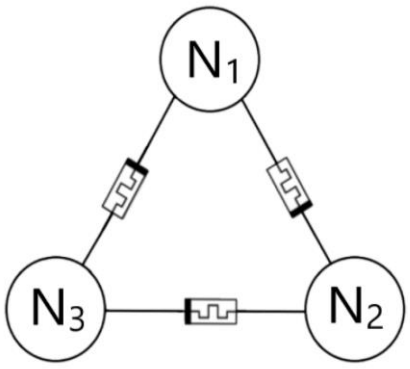
Şekil 1.6'da görüldüğü gibi her nöron bilgiyi bir veya daha fazla bağlantı yoluyla alır. *x* bilgisini taşıyan sinyaller başlangıçta nöronlar arasında bulunan *w* ağırlıkları aracılığıyla işlenir. Bu işlem *c.x* doğrusal çarpımı şeklinde olabileceği gibi *W(x)=tanh(x)* gibi doğrusal olmayan bir aktivasyon fonksiyonu şeklinde de olabilmektedir. Post-sinaptik nörona gelen tüm sinyaller bu kurala göre işlenir ve bir araya toplanarak post-sinaptik nöronun uyarı akımını oluşturur. Sinir ağının eğitim süreci boyunca, nöronlar arasındaki ağırlıklar, sinir ağı çıktısının optimize edilmesini sağlamak için özelleştirilmiş algoritmalar aracılığıyla değiştirilir. Bu algoritmalar, tahmin edilen çıktı ile istenen çıktı arasındaki farkı en aza indirecek şekilde ağırlıkları ayarlar, böylece sinir ağının doğruluğunu ve verimliliğini artırır.

Literatürde son yapılan çalışmalarda, biyolojik sinapsların sağladığı davranışa daha çok benzeyen doğrusal olmayan bir davranış sağlayan memristör elemanları sinaptik ağırlık olarak kullanılmıştır[17, 56]. Şekil 1.6'deki her ağırlık bir memristör elemanı ile yer değiştirilirse memristörler sinyalleri işleme sürecinden sorumlu olur. Her memristör için eşdeğer ağırlık, SNN'nin nöronları ateşleme davranışı sergiledikçe sürekli olarak güncellenir. Memristör elemanı, manyetik akı ile elektrik yükü arasındaki ilişki boşluğunu dolduran yeni bir devre elemanı ilk olarak 1971'de Leon Chua tarafından tanıtılmıştır[57]. Memristörlerin, voltaj ve akım arasında bir histeresis ilişkisi göstermeleri nedeniyle bilgiyi depolamak için hafıza benzeri bir yeteneğe sahip oldukları düşünülmektedir. Bu davranışın biyolojik sinapsların biyolojik davranışına uygun olduğu düşünülmektedir. Hem memristörler hem de biyolojik sinapslar, bağlantı güçleri şunlara

bağımlılığını gösterir: i) voltaj/membran potansiyel farkına ve ii) memristör/sinapsın durumuna (*W(x)* ve *x* değerleri). Ayrıca, çalışmalar memristörlerin faz dışı nöronlar arasında senkronize olma yeteneğini göstermektedir[58, 59]. Bu bağlamda, yapılan tez çalışmasının bir bölümü yukarıda anlatılan memristif ağırlık mekanizmaların etkilerinin incelenmesine ayrılmış olup, memristörleri ağırlık olarak kullanan bir ateşlemeli sinir ağı, memristörlerin ağın nöronlarını senkronize etme yeteneğini test etmek için tasarlanmıştır[60]. Tasarlanan sinir ağı, aşağıdaki ODE set ile tanımlanan memristif Hindmarsh Rose nöron modeli (mHR) temeline dayanmaktadır:

$$\begin{aligned} \dot{x} &= y - x^3 + 3x^2 + \tanh(z)\,x \\ \dot{y} &= 1 - 5x^2 - y \\ \dot{z} &= x \end{aligned} \tag{1.3}$$

Zaman bağımlı değişkenler x, y ve z sırasıyla membran potansiyelini, ateşleme değişkenini ve nöronun memristif elemanının manyetik akısını ifade eder. Şekil 1.7'de görüldüğü gibi, memristör elemanları kullanılarak mHR modelinin üç nöronunun kapalı döngü şeklinde bağlanmasıyla bir ateşlemeli sinir ağ oluşturulur. Bao ve ark. tarafından önerilen yapıya kıyasla bu yapının seçilmesiyle senkronizasyon durumuna ulaşma olasılığının daha da artırıldığına inanılmaktadır[61].



Şekil 1.7 Üç mHR nöron tabanlı ateşlemeli sinir ağı

Yukarıdaki nöronların her birinin N_1, N_2ve N_3'ün $N_i = [x_i; y_i; z_i]$ ($i \in 1,2,3$) ifade ile tanımlandığını varsayalım. Ayrıca, K'nin aşağıdaki tanıma sahip 3x3'lük bir matris olduğunu varsayalım:

$$= \begin{bmatrix} k & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{1.4}$$

Burada k, memristörün bağlantı gücüdür. Ve son olarak, mHR nöron fonksiyonu $G(N)$'yi şu şekilde tanımlayalım:

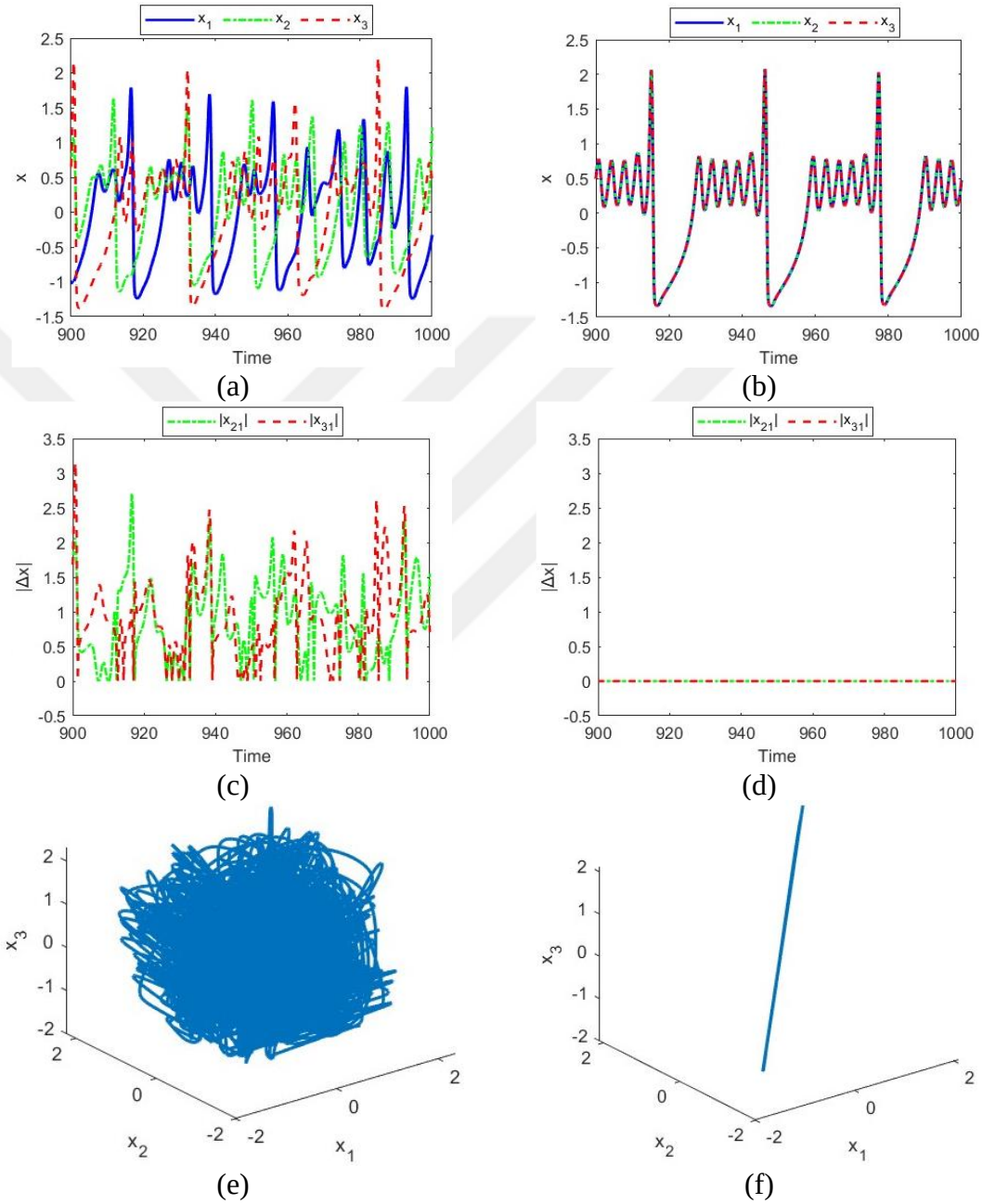
$$G(\boldsymbol{N}) = \begin{bmatrix} y - x^3 + 3x^2 + \tanh(z)\, x \\ 1 - 5x^2 - y \\ x \end{bmatrix} \tag{1.5}$$

Bu, Şekil 1.8'de görülen sistemin Denklem (1.6)'da görüldüğü gibi modellenmesini sağlar:

$$\begin{aligned} \dot{N}_1 &= G(N_1) + K\tanh(\varphi_1)\,(N_1 - N_2) + K\tanh(\varphi_3)\,(N_1 - N_3) \\ \dot{N}_2 &= G(N_2) + K\tanh(\varphi_2)\,(N_2 - N_3) + K\tanh(\varphi_1)\,(N_2 - N_1) \\ \dot{N}_3 &= G(N_3) + K\tanh(\varphi_3)\,(N_3 - N_1) + K\tanh(\varphi_2)\,(N_3 - N_2) \\ \dot{\varphi}_1 &= x_1 - x_2 \\ \dot{\varphi}_2 &= x_2 - x_3 \\ \dot{\varphi}_3 &= x_3 - x_1 \end{aligned} \tag{1.6}$$

Zaman bağımlı olan değişkenler φ_1, φ_2 ve φ_3, sırasıyla her nöron çifti (N_1, N_2), (N_2, N_3) ve (N_3, N_1) arasında bulunan memristörlerin manyetik akısını ifade eder. Önerilen sistem için senkronizasyon yeteneği, Lyapunov kararlılık teoremi ve Lipschitz teoremi kullanılarak teorik olarak analiz edilmiş ve doğrulanmıştır. Hem memristif kuplajlama gücü yeterince yüksek olduğunda ve aynı zamanda hem de başlangıç akı değeri yeterince negatif değere ayarlandığında üstel senkronizasyona ulaşıldığı düşünülmektedir. Teorik analizden sonrasında, öngörülen üstel senkronizasyon davranışını doğrulamak için nümerik simülasyonlar yürütülmüştür. İki farklı memristif kuplaj gücüne sahip ancak benzer başlangıç koşullarına sahip iki durum dikkate alınmıştır [x_1;0;0; x_2; 0; 0; x_3; 0; 0; -2;-2;-2], burada x_1, x_2 ve x_3, [-1.3, 2] arasında rastgele değerlerdir. İlk durum, 0,2 değerinde bir memristif kuplaj gücü k'yi ele alırken, ikinci durum 3'e eşit bir k değerine sahiptir. İlk durumda k = 0,2, zar potansiyeli diyagramı (a) her biri bağımsız dinamiklere sahip üç farklı sinyal gösterir. Mutlak hata diyagramı (c) hata değerinin 0'dan büyük olduğunu gösterir. Öte yandan faz diyagramı sinyaller arasında bir ilişki olmadığını gösterir. Öte yandan ikinci durumda k = 3, zar potansiyeli diyagramı (b) birbirleriyle

çakışan üç sinyal gösterir. (d)'deki mutlak hata hatanın 0'a eşit olduğunu gösterir. Faz diyagramı (f) nöronlar arasındaki ilişkinin x1 = x2 = x3'ün doğrusal olduğu tam senkronizasyon olduğunda beklenen bir durumdur.

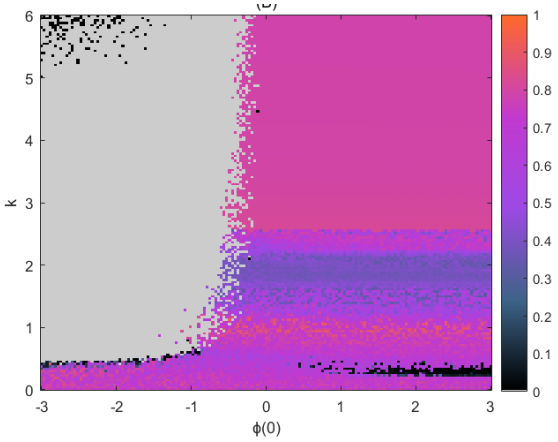


Şekil 1.8 Sabit başlangıç koşulları için ağın ateşleme dinamikleri [x_1;0;0; x_2; 0; 0; x_3; 0; 0; -2;-2;-2], (a-b) membran potansiyeli zaman serisi, (c-d) zaman içinde x_1-x_2 (yeşil çizgi) ve x_1-x_3 (kırmızı çizgi) arasındaki mutlak fark, (e-f) senkronizasyon faz portreleri (a, c) k = 0,2, (b, d) k = 3.

Bu ateşlemeli sinir ağı için normalize edilmiş ortalama senkronizasyon hatası NMSE şu şekilde yazılır:

$$NMSE = \frac{1}{n}\sum_{i=1}^{n} \frac{\sqrt{||N_1(i)-N_2(i)||^2+||N_2(i)-N_3(i)||^2+||N_3(i)-N_1(i)||^2}}{\sqrt{2\left(||N_1(i)||^2+||N_2(i)||^2+||N_3(i)||^2\right)}} \quad (1.7)$$

Bu denklem kullanılarak, mHR tabanlı ateşlemeli sinir ağı için normalize edilmiş ortalama senkronizasyon hatası, [0; 6] aralığındaki memristif kuplaj gücü değerleri ve [-3; 3] aralığındaki başlangıç memristif akı değerleri için hesaplanmıştır. Yukarıda belirtilen denklem için simülasyon sonuçları, Şekil 1.9'da görüldüğü gibi bir hata haritasına dönüştürülmüştür. En küçük normalize edilmiş ortalama senkronizasyon hatası $NMSE \approx 0$ değerine sahip bölge, gri ile gösterilen bölgedir. Bu, sinir ağının tüm nöronlarının aynı x, y ve z değerlerine sahip olmasıyla ortaya çıkan dinamiklere dönüşür ve bu da denklem (1.6)'nın sıfırlanması ile sonuçlanır. Bu bölgenin k>0,5 ve $\varphi(0)$<0'dan uzandığı görülür ki bu da teorik analizle bulunan sonucu teyit eder. Memristörlerin başlangıç akı değerlerini negatif bir değerde tutarken memristif kuplajlama gücü değeri k'yi yüksek değerlere ayarlayarak, sinir ağı tam senkronizasyon durumuna ulaşabilir. Ayrıca, pozitif başlangıç akısı yönünde siyah bölge ile gösterilen, tüm nöronların birbirinden 120° kaydırıldığı bir tür faz kilitlemesine sahip olduğu bulunan başka bir bölge daha vardır.

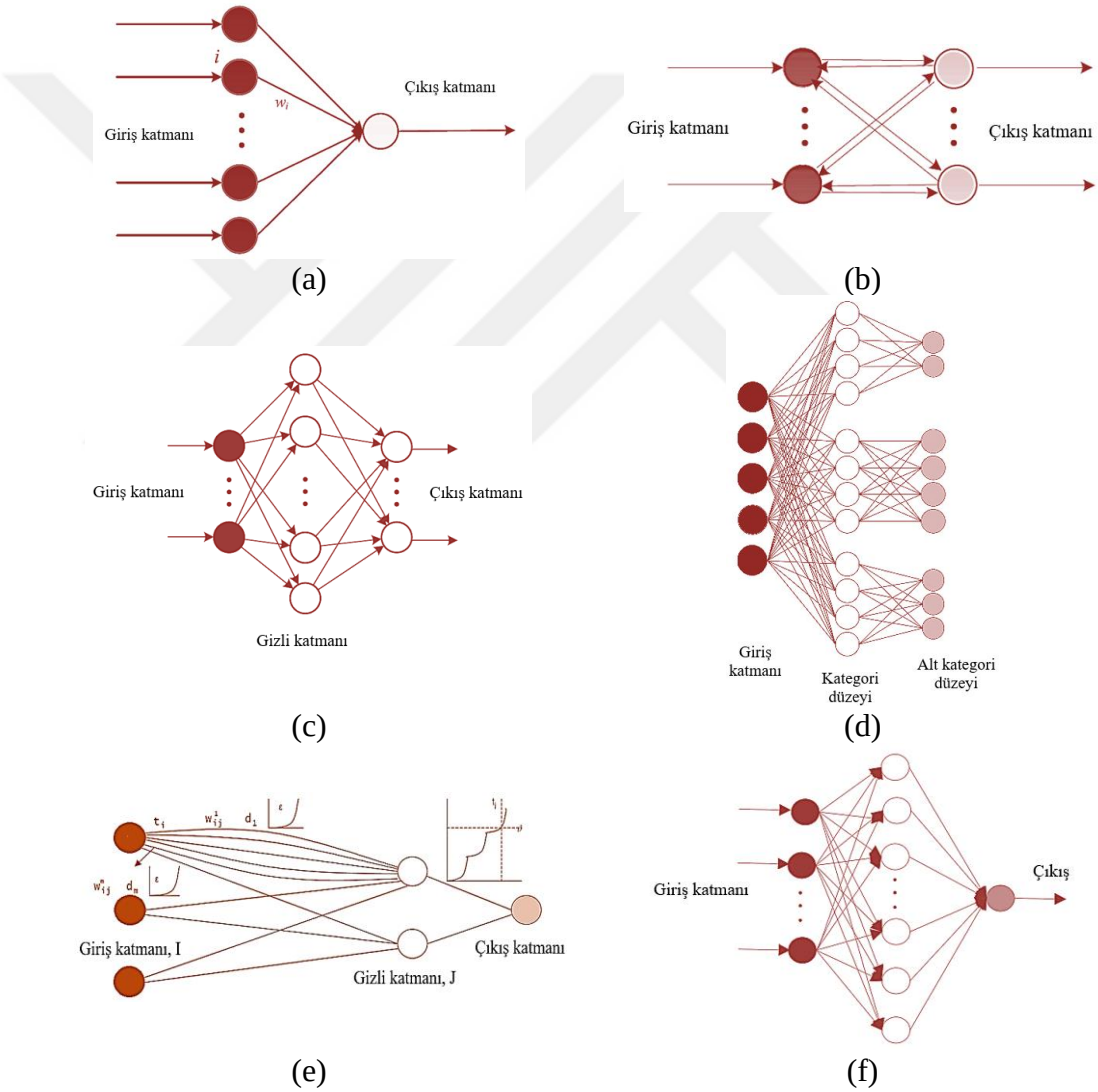


Şekil 1.9 Normalize edilmiş ortalama senkronizasyon hata haritası mHR sinir ağı.

1.3. Ateşlemeli Ağ Mimarileri ve Giriş Sinyallerin Kodlanması

Şekil 1.10'de görüldüğü gibi SNN'ler de benzer şekilde ağın karmaşıklığını ve kullanım kolaylığını belirleyen birçok farklı bağlantı türleriyle oluşturulmuştur. Bu ağ yapıları sırasıyla tek katmanlı ileri beslemeli sinir ağı, tekrarlayan sinir ağı, çok katmanlı ileri beslemeli sinir ağı (çok çıkışlı), kısmen bağlı çok katmanlı besleme -ileri sinir ağı, çok bağlantılı ileri beslemeli sinir ağı, çok katmanlı ileri beslemeli sinir ağı (tek çıkışlı) şeklinde sıralanabilir. Tek katmanlı ileri beslemeli sinir ağının giriş katmanında birden fazla nöron, çıkış katmanında ise tek bir nöron bulunmaktadır. Bu ağ, kurulabilecek ve eğitilebilecek en basit sinir ağı türü olarak görülmektedir. Bu ağda yer alan ağırlıkların sayısı, giriş nöronlarının sayısına eşittir. Bu ağ, ağın ağırlık değerlerini haritalandırarak kendisine beslenen girdilerden görsel veri çekebilmektedir. Öte yandan, tekrarlayan sinir ağı, çok sayıda giriş nöronu ve çıkış nöronundan oluşur. Bu ağda bulunan toplam ağırlık sayısı(2 × giriş nöronlarının sayısı × çıkış nöronlarının sayısı) giriş ve çıkış nöronlarının sayısına göre değişmektedir. Bu ağ yapısı tek katmanlı ileri beslemeli sinir ağına göre daha karmaşıktır. Aynı sayıda girdiyi ve aynı veri kümesini kullanarak onu aynı şekilde eğitmek için yaklaşık iki kat daha fazla hesaplama gücü gerektirir. Bu ağ, zamansal verileri tanıma ve daha fazla kategoriyi tanıma konusunda gelişmiş bir yeteneğe sahiptir. Çok katmanlı ileri beslemeli sinir ağı (çok çıkışlı) giriş, gizli ve çıkış katmanı olmak üzere en az üç katmana sahiptir. Her katman, toplam ağırlık sayısını (giriş nöronlarının sayısı × gizli nöronların sayısı + gizli nöronların sayısı × çıkış nöronlarının sayısı) oluşturan birden fazla nöron oluşturur. Bu ağın eğitilmesi gereken birden fazla ağırlık çeşidi vardır ve bu da hesaplama gücünü zorlaştırmaktadır. Bu tür bir ağ, kendisini uygulamalarda kullanımını daha pratik hale getirecek şekilde eğitmek için özel algoritmalara sahiptir. Öte yandan, kısmen bağlı çok katmanlı ileri beslemeli sinir ağı, bir katmandaki nöronların gruplar oluşturarak bir sonraki veya önceki katmana bağlandığı çok katmanlı bir sinir ağı biçimidir. Bu gruplar genellikle sınıflandırılacak toplam kategori sayısıyla aynı sayıda olacak şekilde ayarlanır. Bu ağda eğitilecek ağırlıkların sayısı değişir ve çok katmanlı ileri beslemeli sinir ağındaki ağırlıktan daha az olma eğilimindedir. Bu ağ, birçok kategoriye sahip verileri verimli bir şekilde sınıflandırmak için kullanılır. Çoklu katmanlı ileri beslemeli sinir ağları, nöron çiftleri arasında birden fazla bağlantı içeren sinir ağlarıdır. Her nöron, tipik olarak farklı zaman gecikmelerine sahip birden fazla bağlantı aracılığıyla başka bir katmanın nöronuna bağlanır. Bu tür ağlarda sayı ağırlıkları önemli ölçüde artmasına rağmen, bu tür ağ yapısının ağın öğrenme yeteneğini arttırdığına inanılmaktadır. Son olarak, çok katmanlı ileri beslemeli sinir ağı (tek çıkışlı) en az üç

katmana sahiptir: giriş katmanı, gizli katman ve çıkış katmanı. Bir nörona sahip olan çıktı katmanı dışında tüm katmanlarda birden fazla nöron bulunur. Tasarım, toplam ağırlık sayısına (giriş nöronlarının sayısı × gizli nöronların sayısı + gizli nöronların sayısı) eşit olan hibrit bir şekil ve şekil biçimidir. Bu ağ, önceki ağın özelliklerine ve daha kolay eğitilebilecek daha basit bir tasarıma sahiptir. Bu ağın çıkış nöronu, sinir ağına girilen bilgilerin kategorisini belirleyen ve görüntüleyen kullanır. Bu tezde, Şekil 1.10 (f)'de görülen tasarım, kalıpları öğrenmek ve kategorize etmek için kullanılan FHN nöron tabanlı SNN ağı gerçekleştirmek için kullanılmıştır.



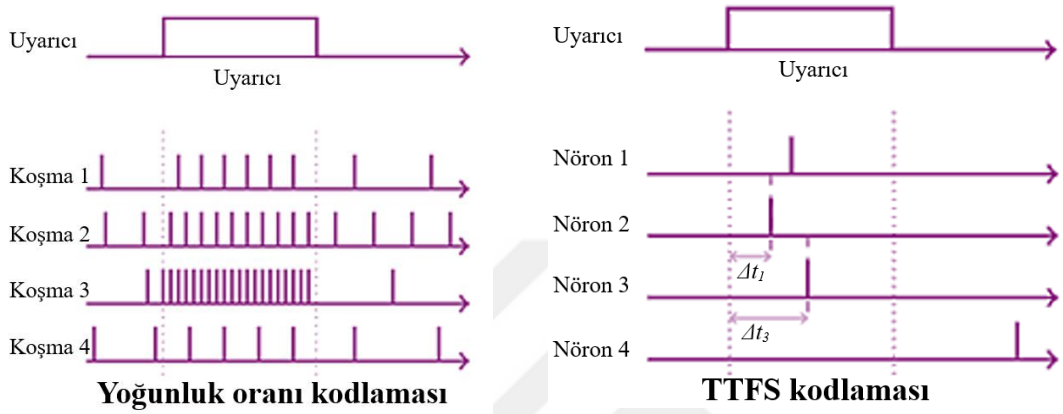
Şekil 1.10 Sinir ağlarının tasarımları, (a) tek katmanlı ileri beslemeli sinir ağı, (b) tekrarlayan sinir ağı, (c) çok katmanlı ileri beslemeli sinir ağı (çok çıkışlı) (d) kısmen bağlı çok katmanlı ileri beslemeli sinir ağı, (e) çok bağlantılı ileri beslemeli sinir ağı (f) çok katmanlı ileri beslemeli sinir ağı (tek çıkışlı).

Yapay sinir ağlarında (YSA), ilk katmanın girdisi kayan nokta değerlerinin bir vektörüdür, oysa ateşlemeli sinir ağlarında (SNN'ler), Dirac delta fonksiyonlarının zamana bağlı bir vektörüdür. Beyinde bilgi, aksiyon potansiyeli dizileri veya ani artış dizileri yoluyla kodlanır. Bu aksiyon potansiyelleri, nöron gövdesi boyunca ilerleyen ani yükselmelerin voltajındaki değişikliklere rağmen, Boolean ya hep ya hiç değerleri olarak yorumlanabilir. Beyin bölgesi ve nöron tipine göre çeşitli kodlama mekanizmaları kullanılmaktadır. Genel olarak tüm kodlama teknikleri iki ana kategoriye ayrılabilir: oransal ve zamansal kodlama[62, 63].

i) Oransal kodlama, bir analog değerin bir ani artışlar dizisine dönüştürülmesinin, ardışık ani artışların ateşleme hızı veya sıklığı tarafından belirlendiği böyle bir mekanizmadır[64]. Örneğin motor nöronlarda kas kasılma derecesi, ani artış frekansıyla doğru orantılıdır. Hız kodları ayrıca üç alt kategoriye ayrılabilir: sayım, yoğunluk ve nüfus oranı kodları. Bu alt kategoriler sırasıyla zaman içindeki ortalama ani sayısını (ani ne sayısını bir isim gelmeli?), birkaç çalıştırmadaki ortalama ani sayısını ve birkaç nöron üzerindeki ortalama ani sayısını belirler. Robotik uygulamalarda sınıflandırma görevlerinin yanı sıra hız kodlu bilgiler sıklıkla kullanılır. Çoğu uygulama, gerçek nöronların özelliklerini yakından taklit etmek için Poisson dağıtılmış ateşleme dizilerinden yararlanır. Oransal kodlama, sinir kodlamasına ilişkin ilk varsayımlardan biri olsa da son araştırmalar, ani artış dizilerinde, ani artış sayısının ötesinde daha fazla bilginin kodlanabileceğini öne sürer. Spesifik olarak, beyinde gözlemlenen bir öğrenme mekanizması olan ani yükseliş zamanlamasına bağlı plastisite (STDP), bireysel ani yükselişlerin zamanlamasına dayanır. Ek olarak, sinir sisteminin belirli uyaranlara hızlı tepki vermesi, bilgi işleme için uzun spike trenlere güvenmeyebileceğini düşündürmektedir.

ii) Zamansal kodlama, ani artışların göreceli konumlarını sayısal değerlere çevirir ve sinirbilimsel bir varsayımdan çok hesaplamaya dayalı bir çözüm olarak hizmet eder. En temel zamansal kodlama şeması, keyfi bir zaman aralığı uzunluğunun seçildiği ve bu aralık içindeki ani yükselişin konumunun çok önemli olduğu ilk ateşleme zamanı TTFS'dir[65]. Bilgi, uyaranın başlangıcı ile bir nöronun ilk ateşleme arasındaki Δt zaman farkıyla kodlanır. Genel olarak büyük bir genlik, erken ateşleme süresine yol açarken, düşük genlikler, büyük bir aralığa veya hiç

ani yükselmeye yol açmaz. Hata, istenen ve gerçek ani yükselme konumları arasındaki fark olarak tanımlanır. Bu yöntemi kullanarak, geri yayılım algoritması gibi algoritmalar, bu varsayımlardan türetilen denetimli bir öğrenme kuralı olan (SpikeProp'ta olduğu gibi) bir artış ağına uygulanır ve XOR işlevini hesaplamadaki etkinliğini gösterir.



(a) (b)

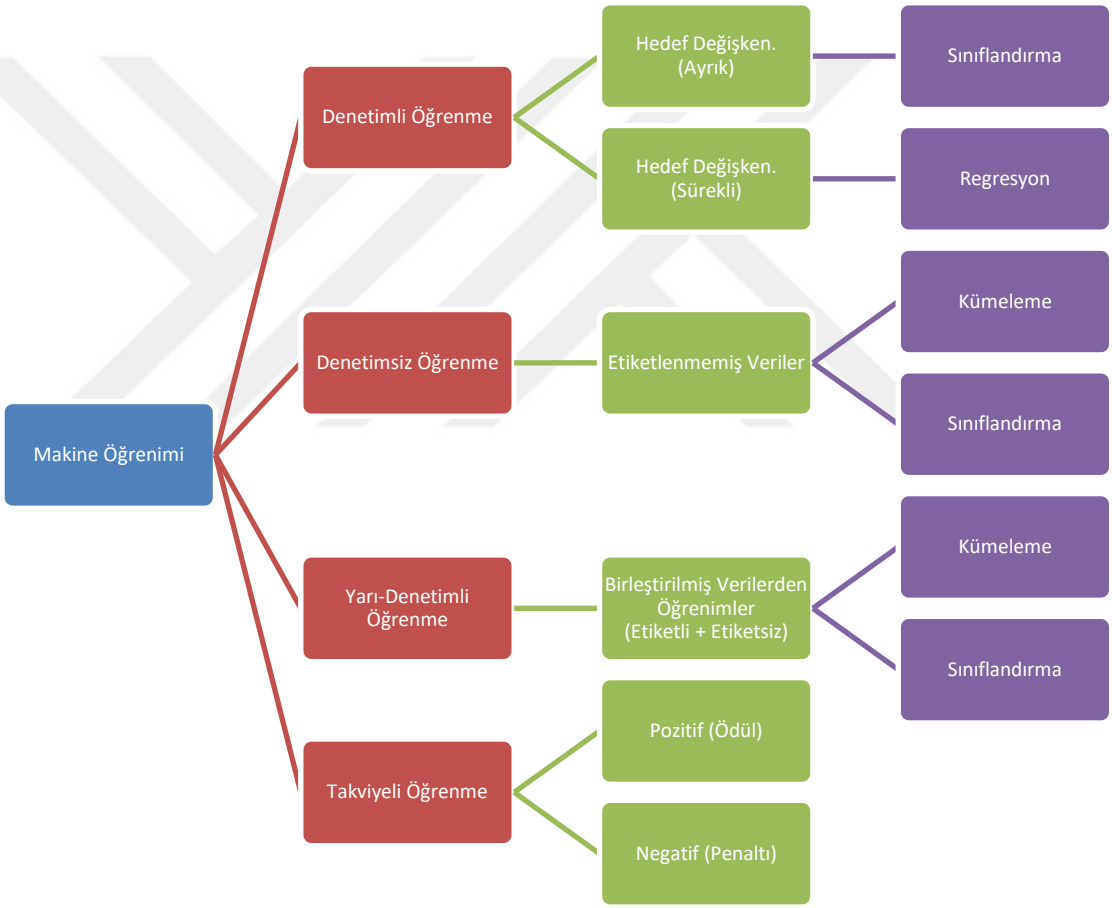
Şekil 1.11 Giriş sinyallerin kodlanma teknikleri, (a) oransal kodlanma (b) zamansal kodlanma

2. BÖLÜM

SNN'LERDE ÖĞRENME STRATEJİLERİ VE ALGORİTMALAR

Sinir ağlarında öğrenme genellikle makine öğrenimi (ML) teknikleri aracılığıyla gerçekleştirilir. Veri analizi ve hesaplama uygulamaların akıllı bir şekilde çalışmasına imkân tanıyarak son yıllarda hızla büyümektedir[66]. ML sistemlere özel bir programlama işlemi yapmaksızın deneyimlerden otomatik olarak öğrenme ve geliştirme yeteneği sunar. Genel olarak bir makine öğrenimi çözümünün etkinliği ve verimliliği; verilerin doğasına, özelliklerine ve öğrenme algoritmalarının performansına bağlıdır. Makine öğrenimi algoritmaları alanında, veriye dayalı sistemleri etkili bir şekilde oluşturmak için sınıflandırma analizi, regresyon, veri kümeleme, özellik mühendisliği, boyutluluk azaltma, birliktelik kurallı öğrenme veya takviyeli öğrenme teknikleri mevcuttur. Ateşlemeli veya ateşlemesiz, her iki tip sinir ağları da nöronlar arasındaki skaler değerli sinaptik ağırlıkları ayarlamak için makine öğrenimi kurallarını içerir. Geri yayılım tabanlı gradyan düşüşün geleneksel YSA'larda oldukça etkili olduğu kanıtlanmıştır. Bununla birlikte, SNN'lerin eğitimi spike davranışının türevlenememesi nedeniyle güçtür. Biyolojik nöronların doğrusal olan ve doğrusal olmayan işlemleri gerçekleştirirken, geri yayılım algoritması öncelikle doğrusal mekanizmaları ele alır. Bununla birlikte, beyinde bir tür geri yayılımın var olduğunu gösteren bazı kanıtlar vardır. Biyolojik nöronlar bilgiyi ani spikelar aracılığıyla işleyip iletse de biyolojik sinir ağlarında öğrenmeyi yöneten mekanizmalar araştırma konularının başlıkları olmaya devam etmektedir. Sinirbilimciler bu öğrenme sürecinin çeşitli biçimlerini ani zamanlamaya bağlı esneklik (STDP) altında sınıflandırmışlardır. STDP yapısının temel özelliği, sinaps öncesi ve sinaps sonrası nöronları birbirine bağlayan sinaptik ağırlığın spike zamanlarına göre ayarlanmasıdır [67].

Bahsedilen çalışmalara ilaveten, bahsedilen sistemler için uygun bir öğrenme algoritması bulma ihtiyacı halen devam etmektedir. Makine Öğrenimi algoritmaları Şekil 2.1'daki gibi temel olarak dört tipe ayrılır: denetimli öğrenme, denetimsiz öğrenme, yarı denetimli öğrenme ve gösterilen takviyeli öğrenme. Bu öğrenme tiplerinden her bir öğrenme tekniği ve gerçek dünyadaki zorlukların üstesinden gelmede pratik uygulanabilirliği yönünden farklılık göstermektedir.



Şekil 2.1 Makine öğrenimi türleri ve kullanımları

- Denetimli öğrenme, etiketli eğitim verilerini kullanarak girdileri çıktılarla eşleştiren bir fonksiyonun türetildiği makine öğrenimidir [68]. Bu yaklaşım, girdi verilerinden belirli hedeflere ulaşmak için bilinen sonuçları olan örneklerden yararlanır. Denetimli

öğrenmedeki birincil hedefler, verileri kategorilere ayıran sınıflandırma ve veri noktaları içindeki ilişkileri modelleyen regresyon gibi görevleri içerir. Örneğin, ürün incelemeleri gibi verilerin duyarlılığını tespit etme veya sınıf etiketini tahmin etme gibi çalışmalar denetimli öğrenmeye örnektir.

- Denetimsiz öğrenme, insan müdahalesi olmadan yalnızca veriye dayalı süreçlere dayanarak, etiketlenmemiş veri kümelerinin bağımsız analiz edilmesi yöntemidir[69]. Özellikleri çıkarmak, önemli kalıpları ve yapıları belirlemek, verileri kümeler halinde gruplamak ve önceden tanımlanmış hedefler olmadan verileri keşfetmek için yaygın olarak kullanılır. Denetimsiz öğrenmedeki temel görevler arasında kümeleme, yoğunluk tahmini, özellik öğrenme, boyutluluğun azaltılması, birliktelik kurallarının keşfedilmesi, anormallik tespiti ve benzeri konular yer alır.
- Yarı denetimli öğrenme, etiketli ve etiketsiz verilerden yararlanarak denetimli ve denetimsiz yaklaşım unsurlarını birleştirir [29]. Tamamen denetlenen ve denetlenmeyen öğrenme paradigmaları arasındaki boşluğu doldurur. Birçok gerçek dünya senaryosunda, etiketli veriler az olabilirken, etiketlenmemiş veriler bol olabilmektedir. Bu durum yarı denetimli öğrenmeyi değerli kılar. Yarı denetimli öğrenme modellerinin temel amacı, tahmin sonuçlarını yalnızca etiketli veriler kullanılarak elde edilebilecek sonuçların ötesinde geliştirmektir. Yarı denetimli öğrenme uygulamaları, veri etiketleme verimliliğinin optimize edilmesinin hayati önem taşıdığı makine çevirisi, sahtekarlık tespiti ve metin sınıflandırma gibi çeşitli alanları kapsar.
- Takviyeli öğrenme, yazılım aracılarının ve makinelerin, verimliliği artırmak için belirli ortamlardaki optimum davranışları özerk bir şekilde belirlemelerine olanak tanıyan bir makine öğrenme algoritmasıdır [70]. Bu yaklaşım, kararların ödüller veya cezalar tarafından belirlendiği çevreyle etkileşimler tarafından yönlendirilir. Ödülleri en üst düzeye çıkarmak veya riskleri en aza indirmek için bu etkileşimlerden elde edilen içgörülerden yararlanılmaktadır. Takviyeli öğrenme robotik, otonom sürüş, üretim ve tedarik zinciri lojistiği gibi karmaşık sistemlerde görevleri olan; otomatikleştirmeyi veya operasyonel verimliliği optimize etmeyi amaçlayan yapay zekâ modellerinin eğitiminde özellikle etkilidir. Ancak basit veya anlaşılır sorunların çözümünde tercih edilen bir seçenek olmayabilir.

Bu tez konusunda, denetimli öğrenme ve denetimsiz öğrenmeden elde edilen algoritmaların her ikisine de odaklanılmıştır. Tablo 2.1, çeşitli öğrenme algoritmalarını,

her algoritma için kullanılan ilgili sinir ağlarını ve bunların performanslarını göstermektedir. Sinir ağı yapıları ve bu yapılarla elde edilen doğruluk tespitleri ilgili çalışma sonuçlarından alınmıştır [71–74]. Tabloda yazılan Multi-STIP, Multi-ReSuMe, SpikeProp, SWAT ve BP-STDP algoritmaları denetimli öğrenme algoritmaları olarak sınıflandırılırken, STDP algoritması bir denetimsiz öğrenme algoritmasıdır.

Tablo 2.1 Literatürde bulunan farklı algoritmalara dayalı ateşlemeli sinir ağlarının karşılaştırılması

Yazarlar	Nöron Modeli	Kodlama Yöntemi	Mimarinin Düzeni	Öğrenme Algoritması	Veri seti	Doğruluk
Lin et al.	SRM	Zamansal	4-50-1	Multi-STIP	Iris	96.7%±2.0
Sporea et al.	SRM	Zamansal	4-90-1	Multi-ReSuMe	Iris	94.0%±2.8
Bohte et al.	LIF	Zamansal	64–15–2	SpikeProp	WBC	97.0%±0.6
Wade et al.	LIF	Zamansal	9–117–2	SWAT	WBC	96.7%±2.3
Querlioz et al.	LIF	Oransal	784-300	STDP	MNIST	93.5%
Tavanaei et al.	IF	Oransal	784-1000-10	BP-STDP	MNIST	96.6%

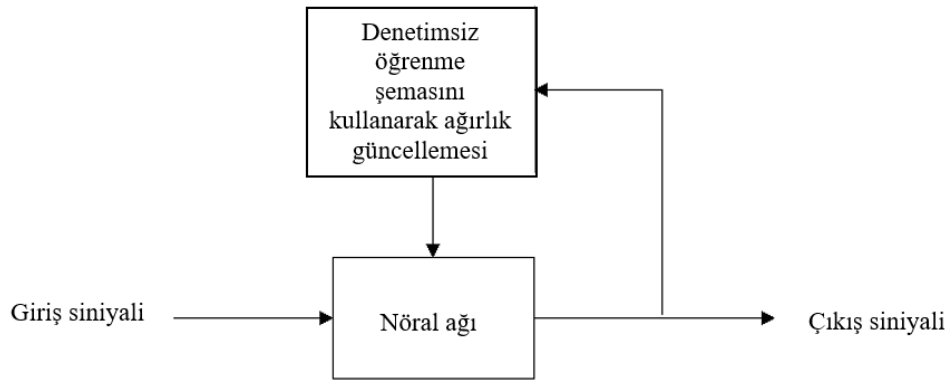
Bohte ve ark. doğrusal olmayan XOR problemini çözmek üzere bir SNN'yi eğitmek için gradyan düşüş öğrenme kuralını kullanan SpikeProp algoritmasını sunmuşlardır[72]. Bununla birlikte, SpikeProp algoritması bilgiyi kodlamak için tek bir artış kullanır ve düşük bir öğrenme oranına sebep olur. Bu durumda ağı eğitmek için gereken toplam epoch sayısını artar. Wade ve ark. Bienenstock-Cooper-Munro ve STDP mekanizmalarını birleştirerek çok katmanlı ileri beslemeli SNN'ler için bir sinaptik ağırlık ilişkilendirme eğitimi (SWAT) algoritması önermişlerdir ve sonuçlarını SpikeProp algoritmasının sonuçlarıyla karşılaştırmışlardır[75]. Sporea ve ark. STDP ve anti-STDP öğrenme kurallarını birleştirerek çok katmanlı SNN'lerin sinaptik ağırlığını güncelleyen Multi-ReSuMe algoritmasını geliştirmişlerdir[76]. Bu algoritma, orijinal olarak Ponulak ve

diğerleri tarafından geliştirilen ReSuMe algoritmasının bir uzantısıdır. Lin ve ark. çok katmanlı SNN'lere öğrenmeyi dahil etmek için STKLR algoritmasını genişleterek Multi-STIP algoritmasını önermişlerdir[71]. Bu algoritmada öğrenme işlemi, spike trenlerinin iç çarpımı hesaplanarak gerçekleştirilir. Lin ve ark. bu SNN'yi kullanarak elde ettikleri sonuçlarını Multi-ReSuMe algoritmasıyla da karşılaştırmışlardır. Querlioz ve ark. memristör elemanının iki katmanlı bir SNN'deki sinaps bağlantılarını modelleme konusundaki yeteneğini göstermişlerdir[74]. STDP öğrenme kuralını kullanarak MNIST'nin elle yazılan sayılarını yüksek doğrulukla sınıflandırabilen bir denetimsiz öğrenme uygulaması ortaya koymuşlardır. Tavanaei ve diğerleri. STDP ve gradyan düşüş öğrenme kurallarını kullanan hibrit bir algoritma geliştirmişlerdir. Algoritmalarının sonuçlarını çeşitli veri kümeleri üzerinde karşılaştırmışlardır[73].

2.1. Denetimsiz Öğrenme

Denetimsiz öğrenme, bir ağın giriş desenlerinin tüm veri setinin istatistiksel yapısını taklit eden bir yöntemin giriş desenlerini temsil etmek için nasıl öğrenebileceğini anlatır. [68]. Bu, etiketlenmemiş verilerden temel yapıyı ortaya çıkaracak bir fonksiyonun çıkarılmasını içeren bir makine öğrenimi görevidir. Görevin veya değişimin farkında olunamayan yerel olaylara dayanan süreçlerinin 'iyi' veya 'kötü' olarak değerlendirilmesini içerir. Bu algoritmalar eğitim sürecini yönlendirmek için etiketler olmadan çalışır. Denetimsiz öğrenmede, algoritmaya girdi olarak büyük miktarda veri ve her gözlemin özellikleri sağlanır, ancak bunlara karşılık gelen istenen çıktılar sağlanmaz. Öğrenme, Hebb'in sinaptik bağlantı değişikliklerileriyle ilgili 1949 tarihli varsayımından esinlenerek, yerel etkinliğe uyum sağlamayı içerir. Denetimsiz öğrenme yöntemleri çeşitli bileşenlerden oluşabilir: (a) Aktivite olmadığında ağırlıkların kendiliğinden ayarlanır. (b) Presinaptik ani artışlardan bağımsız olarak yalnızca postsinaptik ani artışlardan kaynaklanan etkiler. (c) Kısa süreli sinaptik plastisitede görüldüğü gibi, postsinaptik değişkenlerden bağımsız olarak yalnızca presinaptik ani artışlardan kaynaklanan etkiler. (d) postsinaptik ani artışlar veya depolarizasyon (Hebbian terimleri) ile birlikte presinaptik ani artışlardan kaynaklanan etkiler. (e) Tüm bu etkiler mevcut sinaptik ağırlığa bağlı olarak değişebilir. Ağırlıklar maksimuma yaklaştıkça değişiklikler azalabilir. Şekil 2.2 eğitim ağı bağlantısını göstermektedir. Yapı, harici rehberlik veya bilgi kaynakları olmadan çalıştığı için kendi kendini organize eden veya uyarlanabilir öğrenme algoritması olarak da bilinir. Bunun yerine, girdi verileri içindeki temel bilgileri

öğrenmek ve bunlara uyum sağlamak için yerel verilere ve dahili mekanizmalara dayanır. Sistem, işlediği verilere göre kategoriler veya kümeler halinde düzenlediği eğitim örnekleri ve giriş kalıplarıyla sunulur.



Şekil 2.2 Denetimsiz öğrenme

Hebb'in varsayımı, yapay sinir ağlarının (YSA) nasıl öğrendiğini keşfetmede temel oluşturmuş ve biyolojik sinir sistemlerinin hem teorik analizlerinde hem de hesaplamalı modellerinde etkili olmuştur. Hebb kuralının uygulamaları farklılık gösterebilir [77]. Temel ilişkisel Hebbian öğrenme, presinaptik ve postsinaptik nöronlar arasındaki aktivitelerin senkronizasyonuna dayanır. Hebbian öğrenme dinamiklerinin değişimi aşağıdaki diferansiyel denklemle tanımlanır:

$$\frac{dw_{ij}}{dt} = \alpha\, u_i\, u_j \tag{2.1}$$

Burada w_{ij}, nöral ağ yapısında etkin olan j giriş biriminden i çıkış birimine olan bağlantının ağırlığıdır ve bir öğrenme oranıdır. Sinaps öncesi nöron u_j ile sinaps sonrası nöron u_i arasındaki ağırlık w_{ij}, u_j, u_i ve öğrenme oranı α ile çarpılarak güncellenir. u_j ve u_i aynı işarete sahipse Δw_{ij} değişikliği pozitif olacaktır ve w_{ij}'nin pekiştirildiğini gösterir. İşaretleri farklıysa sıfırdan küçük olacaktır. $\alpha < 0$ olan bir öğrenme kuralının genellikle anti-Hebbian olarak adlandırıldığına dikkat edilmelidir, çünkü pre- ve postsinaptik nöronlar aynı anda aktifse sinaps zayıflar; bu, Hebb tarafından öne sürülenin tersi bir davranıştır. Genel olarak α katsayısı, w_{ij} ağırlığının mevcut değerine bağlı olarak değişebilir. Bu değişkenlik, ağırlık artışını maksimum w_{max} değerine sınırlamak için kullanılır. Bu ağırlık bağımlılığına 'sert sınır' ve 'yumuşak sınır' olarak bilinen iki standart yaklaşım vardır. Sert sınır yaklaşımında $\alpha = \gamma 2$, $0<w_{ij}<w_{max}$ aralığında sabit kalır, aksi

takdirde sıfır olur. Dolayısıyla eğer w_{ij} w_{max}'ın üst sınırına ulaşırsa ağırlık artışı aniden durur. Öte yandan, sinaptik ağırlık artışı için yumuşak bir sınır, Denklem (2.2)'teki α parametresinin, w_{ij} maksimum değeri w_{max}'a yaklaştıkça sıfıra yaklaşacak şekilde ayarlanmasıyla elde edilebilir,

$$\alpha\ (w_{ij})\ =\ \gamma_2 (w^{max}\ -\ w_{ij})^{\beta} \tag{2.2}$$

pozitif sabitler γ_2 ve β ile. Üssün tipik değeri β=1'dir ancak diğer değerler de kullanılabilir [78]. β→0 için yumuşak sınır kuralı sert sınır kuralına yakınsar.

Mevcut ateşlemeli sinir ağı (SNN) öğrenme algoritmalarında Hebbian plastisite öne çıkarken, son yıllarda Spike Zamanlamaya Bağlı Plastisite (STDP) algoritması büyük ilgi görmektedir. STDP algoritması, Hebbian denetimsiz öğrenme kuralının bir varyasyonunu temsil eder. Sinaptik ağırlık değişikliklerinin, sinaps öncesi ve sinaps sonrası ani spikelar arasındaki zamanlama ilişkisi tarafından belirlendiğini önermektedir. Presinaptik bir nöron j'den gelen bir sinapsın ağırlık değişimi Δw_{ij}, presinaptik ani spikeları ile postsinaptik ani spikeları arasındaki göreceli zamanlamaya bağlıdır. t^f, j sinapsına presinaptik ani spike'ının varış zamanlarını göstersin, burada f = 1, 2, 3,… M, presinaptik ani spikelarını sayar. Benzer şekilde, n=1, 2, 3, … N ile t^n, postsinaptik nöronun ateşleme zamanlarını temsil etsin. Pre- ve postsinaptik ani çiftler içeren bir stimülasyon protokolü tarafından indüklenen toplam ağırlık değişimi Δw_{ij} daha sonra şu şekilde hesaplanır[79, 80]:

$$\Delta w_{ij}\ =\ \sum_{j=1}^{M} \sum_{i=1}^{N} W(t_i^n\ -\ t_j^f) \tag{2.3}$$

$W(t)$ fonksiyonu (burada $t = t_i^n\ -\ t_j^f$), Şekil 2.3'te gösterilen STDP fonksiyonlarından (öğrenme penceresi de denir) birini belirtir. STDP fonksiyonu $W(t)$için alttaki gösterim popülerdir:

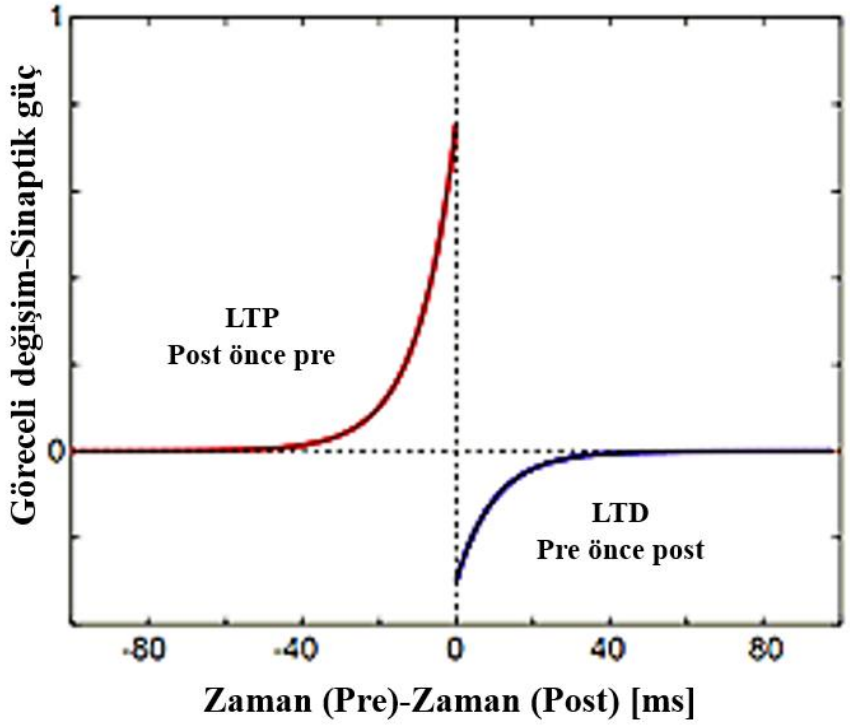
$$W(t) = \begin{cases} -A_{-} \exp\left(\frac{t}{\tau_{-}}\right) & if\ t \le 0 \\ A_{+} \exp\left(\frac{-t}{\tau_{+}}\right) & if\ t > 0 \end{cases} \tag{2.4}$$

Bu gösterim deneysel verilere ve modellere uygun olarak tanımlanmıştır[81, 82]. STDP'ye göre, eğer presinaptik bir artış, postsinaptik bir artıştan kısa bir aralık kadar önce gelirse (t = post zamanı – ön zamanı > 0), sinaptik ağırlık kuvvetlenir. Bu duruma Uzun Vadeli Güçlenme LTP adı verilir ve kuvvetlenme derecesi üstel bir bozunuma fonksiyonu

$A_+ \exp(-t/\tau_+)$ tarafından yönetilir. Burada τ_+ bir zaman sabitidir ve A_+ olası maksimum sinaptik değişimi belirtir. Tersine, eğer postsinaptik bir artıştan kısa bir süre sonra presinaptik bir artış meydana gelirse, sinaptik güç azalır. Bu durum Uzun Vadeli Depresyon LTD olarak bilinir ve bu azalmanın büyüklüğü başka bir üstel bozunuma fonksiyonu olan $A_- \exp(t/\tau_-)$'yi takip eder. Burada τ_- bir zaman sabitidir ve A_- maksimum depresyonu temsil eder. Şekil 2.3, sinaptik değişikliklerin t zaman aralığına göre nasıl değiştiğini göstermektedir. Daha da önemlisi, parametreler değişebileceğinden, STDP işlevinin şekli bir ağ içindeki sinapslar arasında farklılık gösterebilir. Aşağıda verilen denklemde görüldüğü gibi STDP algoritmasının basitleştirilmiş bir versiyonu da mevcuttur. [83].

$$\Delta w_{ij} = \begin{cases} -A_- w_{ij}\,(w_{max} - w_{ij}) & if\ t \leq 0 \\ A_+ w_{ij}\,(w_{max} - w_{ij}) & if\ t > 0 \end{cases} \tag{2.5}$$

Fizyolojik kanıtlar, presinaptik ve postsinaptik ani artışların zamanlama sırasının yalnızca belirli bağlamlarda kritik olması ve ateşlenme oranı gibi presinaptik aktiviteye bağlı olması nedeniyle STDP'nin genel uygulanabilirliğinin tartışıldığını göstermektedir.

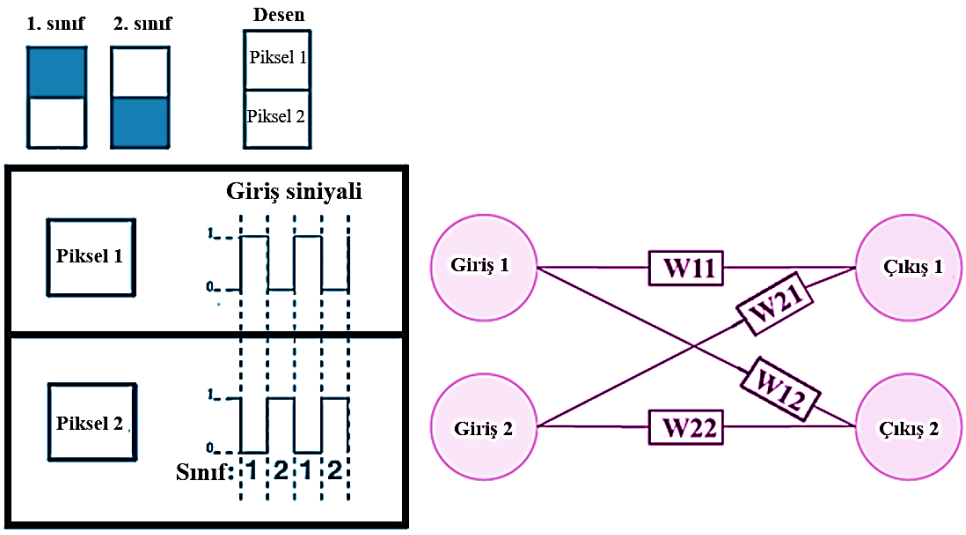


Şekil 2.3 Ateşleme zamanlamaya bağlı plastisite STDP ağırlık değişimi

Denetimsiz öğrenmede yaygın olarak kullanılan bir diğer algoritma, kazanan hepsini alır (WTA) algoritmasıdır[84]. Rekabetçi bir nöron ağından uygun bir nöronu seçen, nesne tanıma gibi bilişsel işlevlerin temel bir ilkel olduğu varsayılan bir mekanizmadır. Kazanan hepsini alır, nöronların aktivasyon için rekabet ettiği sinir ağı modellerinde kullanılan bir hesaplama ilkesidir. Klasik biçiminde, yalnızca en yüksek aktivasyona sahip nöron aktif kalırken diğer tüm nöronlar inhibe edilir. Bununla birlikte, birden fazla nöronun aktif kalmasına izin vermek için bir güç fonksiyonunun uygulandığı yumuşak kazanan hepsini alır gibi varyasyonlar da mevcuttur[85]. Ağdaki çıkış düğümleri, dönüşlü bağlantılar yoluyla kendilerini etkinleştirirken birbirlerini engeller. Zamanla, çıktı katmanındaki en güçlü girdiye karşılık gelen yalnızca bir düğüm aktif kalır. Bu mekanizma, ağın bir kümeden en büyük girişi seçmek için doğrusal olmayan engellemeyi kullanmasına olanak tanır. Kazanan hepsini alır hem sürekli zamanlı hem de artan ağlar dâhil olmak üzere çeşitli sinir ağı modelleri kullanılarak uygulanabilen çok yönlü bir hesaplama yöntemidir. Pek çok pratik durumda aktif hale gelen yalnızca tek bir nöron yoktur, aynı zamanda sabit bir k sayısı için aktif hale gelen tam olarak k adet nöron vardır. Bu prensibe k-kazanan hepsini alır denir.

2.1.1. Denetimsiz Öğrenme Paradigmasına Dayalı Örüntü Sınıflandırma Uygulaması

Örüntü tanımada STDP denetimsiz öğrenme şemasına yönelik bir uygulama olarak, yaygın olarak XOR deseni olarak bilinen 2 farklı girişi tanımlamak için tamamen bağlı bir 2x2 SNN oluşturulmuştur. Problem Şekil 2.4'te görülebilir, doğrudan sinir ağına beslenen bir giriş sinyali iki piksele uygulanmaktadır. Giriş dalga biçimi, her biri ağ üzerinde iki piksellik bir desen görüntüleyen birden fazla zaman dilimine bölünebilir. Karşılıkları presinaptik nöronlar olan siyah pikseller "1" düzeyiyle etkinleştirilirken, beyaz pikseller "0" giriş düzeyiyle temsil edilir. İki çıkış nöronunun, giriş ani artışlarına cevap olarak ilk önce ani yükselişe geçmek için rekabet etmesi beklenir. Ayrıca, kazanan hepsini alır konsepti uygulanarak, ilk yükselen nöron aktif nöron olarak alınırken diğer çıkış nöronları bastırılır. Bu süreç daha sonra sinapslara karşılık gelen güçlenme ve depresyon davranışı uygular.



Şekil 2.4 XOR problemi ve onu çözmek için kullanılan sinir ağı yapısı

Sinir ağı, membran potansiyeli dinamiklerini simüle etmek için FHN nöron modelini kullanır (Bkz. Denklem (1.1) ve (1.2)). 2x2 sinir ağı varsayıldığında ağ değişkenleri $\overrightarrow{N_i} = \{u_i, v_i\}$ olarak endekslenecektir; burada $\vec{N}$ u ve v parametrelerine sahip FHN nöronudur ve i, nöron indeksidir (i=1,2,3,4'tür). i=1,2 endeksleri giriş nöronlarını, i=3,4 endeksleri ise çıkış nöronlarını ifade etmektedir. Çözülmesi istenen problemde, zaman adımı 0,08ms olarak alınmıştır ve simülasyon süresinin t [0,40000] ms aralığına ayarlanmıştır. Sistem Euler Explicit yöntemi kullanılarak simüle edilmiştir. Ayrıca simülasyon boyunca tüm nöronların ani spike dizileri kaydedilmiştir. Piksellerden gelen giriş akımı aşağıdaki gibi modellenmiştir:

$$I_1(t) = \begin{cases} 5 & , \quad mod\left(\frac{t}{250}, 2\right) < 1 \\ 0 & , \quad mod\left(\frac{t}{250}, 2\right) \geq 1 \end{cases} \tag{2.6'a}$$

$$I_2(t) = \begin{cases} 5 & , \quad mod\left(\frac{t}{250}, 2\right) \geq 1 \\ 0 & , \quad mod\left(\frac{t}{250}, 2\right) < 1 \end{cases} \tag{2.6b}$$

Burada $I_1(t)$ ve $I_2(t)$ terimlerinin her biri, ilgili nöronlar için I_{ext} terimi olarak kabul edilir. Öte yandan, çıkış nöronları $I_3(t)$ ve $I_4(t)$ için I_{ext}her adımda aşağıdaki şekilde hesaplanır:

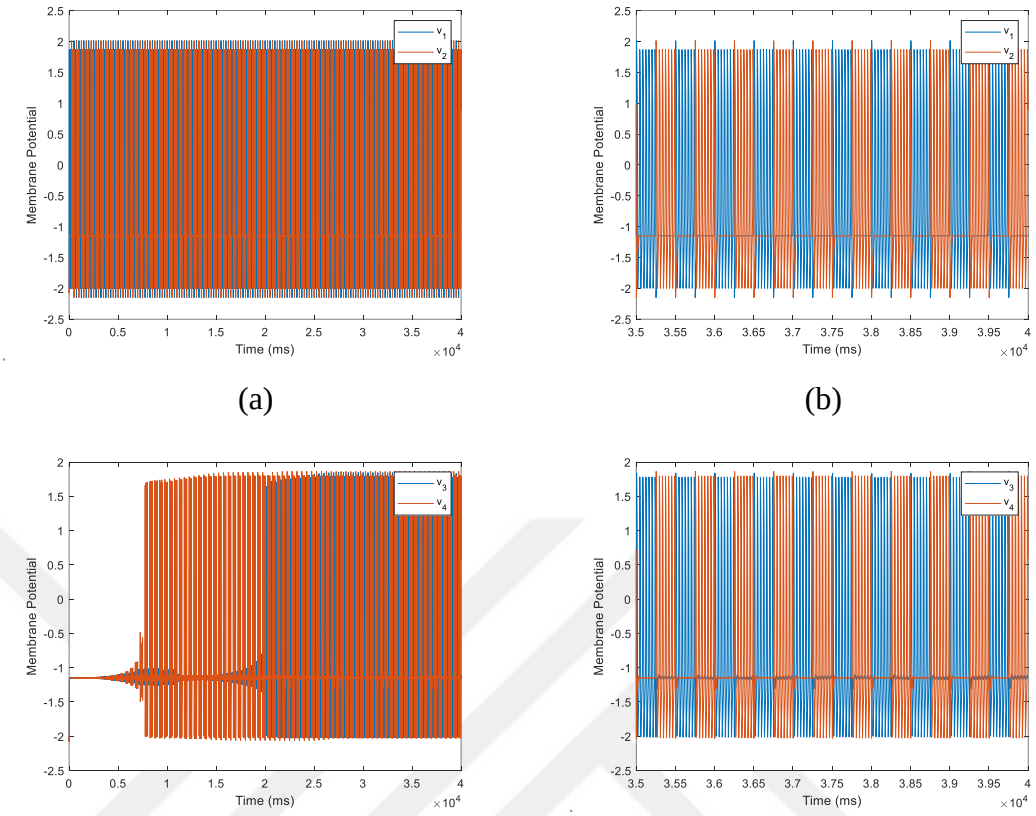
$$I_3(t) = \sum_{i=1}^{2}(u_i - u_3)\, W_{i1} \tag{2.7a}$$

$$I_4(t) = \sum_{i=1}^{2}(u_i - u_4)\, W_{i2} \tag{2.7b}$$

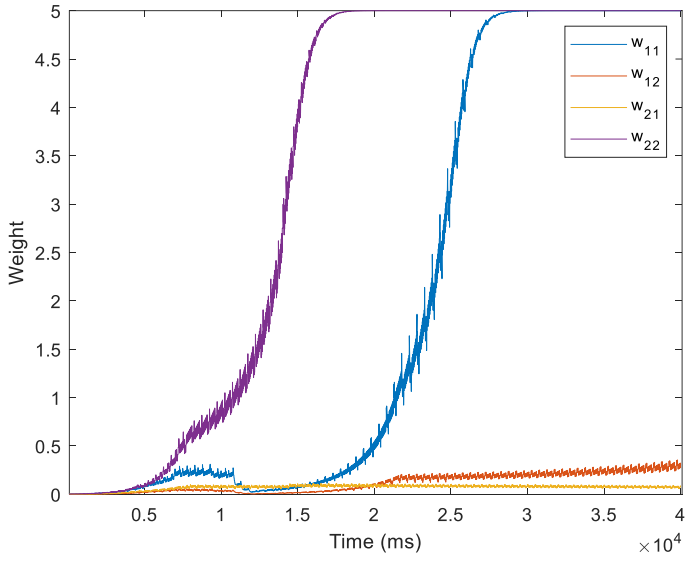
$\{W_{11}, W_{21}, W_{12}, W_{22}\}$ ağırlıkları, Denklem (2.6a) & (2.6b) kullanılarak STDP öğrenme algoritmaları aracılığıyla her 250 ms'de güncellenen [0,5] aralığındaki değerlere sahip olacak şekilde modellenmiştir. Önceki bölümde bahsedilen STDP algoritması için kullanılan parametreler ve nöron modeli parametreleri Tablo 2.2'da verilmiştir. Elde edilen sonuçlar Şekil 2.5 ve 2.6'da görülebilir. Giriş pikselleri Sınıf 1 sinyali verdiğinde N_4 çıkış nöronu bastırılırken N_3 çıkış nöronu yükselir ve aktif hale gelir. Benzer şekilde, Sınıf 2 giriş sinyali verdiğinde N_3 çıkış nöronu bastırılır ve N_4 çıkış nöronu yükselir ve ateşleme davranış sergiler. Simülasyon sırasında ağırlık değerlerini gösteren Şekil 2.6 incelendiğinde, W_{11} ve W_{22} ağırlıklarının w_{max} doyma sınırına yaklaştığı, diğer ağırlıklar olan W_{21}ve W_{12}'nin ise fazla bir değişim yaşamadığı görülmüştür. Bu gözlem, SNN'nin yakınsamaya ulaştığını ve ağın artık tamamen eğitildiğini gösterir. Tüm bu sonuçlar, oluşturulan SNN'nin iki farklı giriş sınıfını denetimsiz bir şekilde ayırt edebildiğini göstermektedir.

Tablo 2.2 2x2 SNN örüntü sınıflandırmasında kullanılan parametreler

STDP Parametreleri			FHN Parametreleri			
A_-	A_+	w_{max}	a	τ	b	R
0.195	0.25	5	0.7	12.5	0.7	0.1



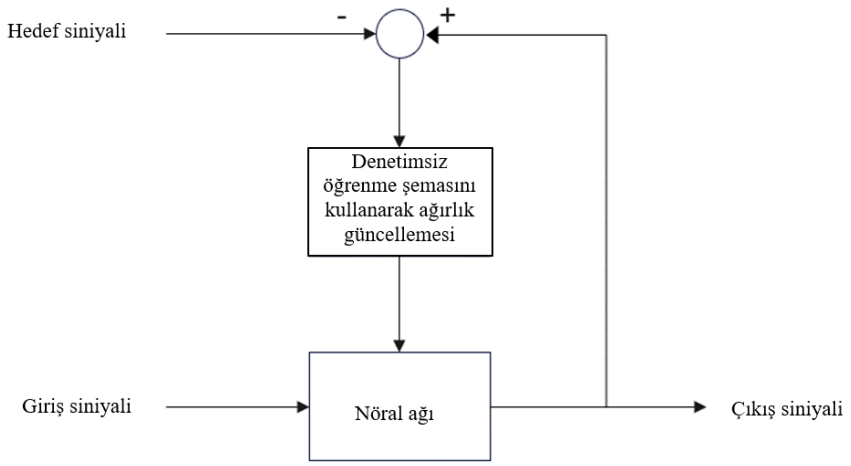
Şekil 2.5 Ateşlemeli sinir ağının dinamiği, (a),(b) giriş nöronlarının membran potansiyeli, (c),(d) çıkış nöronlarının membran potansiyeli



Şekil 2.6 Zamanla nöral ağırlıkların gelişimi

2.2. Denetimli Öğrenme

Denetimsiz bir Hebbian öğrenme kuralı sinaptik ağırlıkları, presinaptik ve postsinaptik nöronlar arasındaki korelasyonlara dayalı olarak ayarlar. Ancak ağın amacı yalnızca verileri temsil etmek yerine belirli bir işlevi gerçekleştirmek olduğunda bu yöntem verimsizdir. Belirli bir görevi etkili bir şekilde gerçekleştirmek için ağın o görevle ilgili bazı bilgileri alması gerekir. Denetimli Öğrenme, bir dizi eğitim örneğinden oluşan etiketli eğitim verilerinden bir işlev türetmeyi amaçlayan, örnek giriş-çıkış çiftlerine dayalı olarak giriş verilerini çıkış verileriyle eşleştirerek bu sorunu giderir. Bu öğrenme sisteminde, harici bir kaynak, çıktının önceden bilindiği geri bildirim uyaranlarıyla ağa rehberlik eder. Eğitim sürecinde çıktılar sürekli olarak istenilen bilgilerle karşılaştırılır. Birkaç iterasyondan sonra, gradyan düşüş kuralı, bağlantı ağırlıklarını ayarlamak için gerçek çıkış sinyali (-) ile hedef sinyal (+) bilgisi arasındaki hatayı kullanır. Bu ayarlama, kümülatif hatayı en aza indirmeyi ve Şekil 2.7'te gösterildiği gibi gerçek çıktıyı hedef sinyale yaklaştırmayı amaçlamaktadır. Bu nedenle denetimli öğrenme, verinin gerçek sınıfının bilindiği gerçeklere dayanır.



Şekil 2.7 Denetimli öğrenme

Ağın eğitilme süreci, denetimsiz öğrenme süreciyle bazı benzerliklere sahiptir. Giriş verileri, özel kodlama yöntemleri aracılığıyla ani artış dizilerine kodlanır. Daha sonra girdi SNN'ye beslenir ve ardından ağı çalıştırmak için belirli bir simülasyon stratejisi kullanılır. Çıkış ani artış dizisi elde edilir ve ardından denetimli öğrenme temel adımları takip eder. İlk önce gerçek çıkış artış dizileri ile istenen çıkış artış dizileri arasındaki hata bulunur. Ağ daha sonra hesaplanan hatayı bir sonraki iterasyonda uygulanması gereken

ağırlık değişimini ΔW bulmak için kullanır ve ağırlığı buna göre $W \leftarrow W + \Delta W$ günceller. Bu süreç, öğrenme süreciyle elde edilen ateşlemeli sinir ağının (SNN) hatası önceden belirlenen minimum hataya ulaşıncaya veya öğrenme epoch'larının üst sınırı aşılıncaya kadar devam eder. Sonlandırma koşulu karşılanmazsa öğrenme işlemi tekrarlanır. Denetimli öğrenmenin ardından, ağın çıkış ani spike trenlerinin kodu, belirli bir kod çözme yöntemi kullanılarak çözülür.

2.2.1. Multi-STIP Algoritması

Multi-STIP öğrenme algoritması, spike trenleri için yeni bir hata fonksiyonu oluşturarak ve bunu birden fazla katmana yayarak hatanın geriye yayılmasını entegre eder [71]. Bu, tüm katmanlarda birden fazla ani ateşleyen nöronlara uygulanabilen, ani spike trenlerinin iç çarpımlarındaki farklılığa dayanan bir sinaptik ağırlık öğrenme kuralıyla birleştirilir. Nörofizyolojik kayıtların bir soyutlamasını temsil eden bir spike dizisi, esasen sıralı spike zamanlarının bir dizisidir. Analiz ve hesaplama kolaylığı için, spike trenlerini sürekli işlevlere dönüştürmek için özel bir evrişim çekirdeği kullanılır ve bunlar üzerinde standart matematiksel işlemlerin kullanılmasına olanak tanır. Bir çekirdek fonksiyonu kullanılarak bir spike dizisi üzerinde evrişim hesaplamaları gerçekleştirilerek, ateşleme dizisi, bir nöronun postsinaptik potansiyeli veya spike ateşleme yoğunluğu fonksiyonu gibi spesifik bir sinirsel fizyolojik sinyal olarak yorumlanabilir [86]. Çok katmanlı SNN'ler için denetimli öğrenme algoritması olan evrişim çekirdeği yoluyla spike dizileri arasındaki ilişkiyi değerlendirerek, çekirdekleştirilmiş ateşleme dizilerindeki farklılıklara dayalı olarak Multi-STIP geliştirilmiştir. Tipik olarak ani artış sürelerinin iç çarpımı, simetrik ve pozitif tanımlı bir çekirdek fonksiyonu seçilerek belirlenir [86, 87]. Ateşleme zamanları t_m ve t_n'ye karşılık gelen iki ani artış için, aralarındaki iç çarpım, bir çekirdek fonksiyonu $\kappa(t_m, t_n)$ kullanılarak tanımlanabilir. Çekirdek fonksiyonu κ, simetrik, kaymayla değişmeyen ve pozitif tanımlı olma koşullarını karşılayan bir Gauss çekirdeği olacak şekilde seçilir.

$$\langle \delta(t - t_m), \delta(t - t_n) \rangle = \kappa(t_m, t_n) = \exp\left(-\frac{|t_m - t_n|^2}{2\sigma^2}\right) \tag{2.8}$$

Ayrıca, Laplace'ın ters çoklu ikinci dereceden (Inverse multiquadratic) ve α-fonksiyonu gibi çekirdek fonksiyonları olmak için gereken koşulları karşılayan çeşitli fonksiyonlar

da mevcuttur. Multi-STIP algoritması, her bir çağ yinelemesinde yaşanan ağırlık değişimini şu şekilde tanımlar:

$$\Delta w_{oh} = \eta \left[\sum_{m=1}^{N_o^d} \sum_{n=1}^{N_h} \kappa(t_m^d, t_n^h) - \sum_{m=1}^{N_o^a} \sum_{n=1}^{N_h} \kappa(t_m^a, t_n^h) \right] \quad (2.9a)$$

$$\Delta w_{hi} = \eta \sum_{O=1}^{N_O} \left[\sum_{m=1}^{N_o^d} \sum_{n=1}^{N_i} \kappa\left(t_m^d, t_n^i\right) - \sum_{m=1}^{N_o^a} \sum_{n=1}^{N_i} \kappa\left(t_m^a, t_n^i\right) \right] w_{oh} \quad (2.9b)$$

Burada Δw_{oh} ve Δw_{hi} Δw_{hi} sırasıyla h gizli katman ile o çıktı katmanı arasında bulunan bir ağırlık ve i giriş katmanı ile h gizli katman arasında bulunan bir ağırlık için ağırlık modifikasyonlarıdır. N_i, N_h, giriş nöronu i ve gizli nöron h tarafından sergilenen toplam ani artış sayısıdır, N_o^a, N_o^d, sırasıyla çıkış nöronu o tarafından sergilenen gerçek toplam ani artış sayısı ve arzu edilen toplam sivri uç sayısıdır. t_n^i, t_n^h, giriş nöronu i'nin n'inci ani artışıdır ve gizli nöron h, t_m^a, t_m^d, sırasıyla çıkış nöronunun o gerçek m'inci sivri ucu ve arzu edilen m'inci sivri ucudur. N_O is çıkış nöronlarının toplam sayısıdır, w_{oh} h gizli ve o çıkış katmanı arasında bulunan ağırlıkların vektörüdür ve η öğrenme oranıdır.

2.2.2. Multi-ReSuMe Algoritması

Multi-ReSuMe, Bohte ve diğerleri tarafından geliştirilen SpikeProp'un çok katmanlı yeteneklerini, Ponulak ve Kasiński tarafından geliştirilen, birden fazla spike ve çeşitli nöron modellerini barındıran ReSuMe'nin çok yönlülüğüyle birleştiren denetimli bir öğrenme algoritmasıdır[88]. Sporea ve Grüning, birden fazla katmana sahip tamamen bağlantılı ileri beslemeli spiking sinir ağları için bu algoritmayı tanıtmıştır[76]. Ağ hatasının geriye yayılması yoluyla ReSuMe algoritmasını birden çok katmana genişletir. Ağırlıklar, STDP ve anti-STDP süreçlerine göre güncellenir ve SpikeProp'tan farklı olarak Multi-ReSuMe, tüm katmanlarda birden fazla artış ateşleyen nöronlara uygulanabilir. Bu çok katmanlı ReSuMe algoritması, hız nöronları için geri yayılım öğrenme algoritmasına benzer, ancak yükselen nöronları kullanır. Algoritma, her epoch için çıkış nöronlarının ağırlık değişikliklerini hesaplar ve bunu, değişikliklerin ağırlık sayısıyla orantılı olmasını sağlayarak presinaptik bağlantı sayısına göre normalleştirir. Algoritmanın, eğitim sürecini gecikmelere karşı dayanıklı hale getiren gürültülü ani spike trenlere dayanabildiği kanıtlanmıştır. Algoritma, biri LTP durumu $W_{post}(s)$ tanımına, diğeri ise LTD durumu $W_{pre}(-s)$ tanımına sahip iki çekirdek işlevi türetmek için STDP fonksiyonu Denklem (2.10)'i kullanır.

$$(s) = \begin{cases} W_{pre}(-s) = -A_{-} \exp\left(\frac{s}{\tau_{-}}\right) & if\ s \leq 0 \\ W_{post}(s) = A_{+} \exp\left(\frac{-s}{\tau_{+}}\right) & if\ s > 0 \end{cases} \quad (2.10)$$

Burada s değişkeni, postsinaptik ani yükseliş ile presinaptik ani yükseliş (t_{post} - t_{pre})arasındaki zaman farkını temsil eder. Multi-ReSuMe, ağırlık değişikliklerini her epoch'ta aşağıdaki denklemler aracılığıyla gerçekleştirir:

$$\Delta w_{oh}(t) = \frac{1}{N_H}\ s_h(t)\left[\int_0^{\infty} W_{pre}(s)[s_o^d(t-s) - s_o^a(t-s)]ds\ \right]$$

$$+\frac{1}{N_H}\ [s_o^d(t) - s_o^a(t)] * \left[a + \int_0^{\infty}\ W_{post}(s)s_h(t-s)ds\ \right] \quad (2.11a)$$

$$\Delta w_{hi}(t) = \frac{1}{N_i N_H}\ s_i(t) \sum_{O=1}^{N_O}\left[\int_0^{\infty} W_{pre}(s)[s_o^d(t-s) - s_o^a(t-s)]ds\ \right] w_{oh}$$

$$+\frac{1}{N_i N_H}\ \sum_{O=1}^{N_O}[s_o^d(t) - s_o^a(t)]\left[a + \int_0^{\infty}\ W_{post}(s)s_i(t-s)ds\ \right]\ w_{oh} \quad (2.11b)$$

Multi-STIP'e benzer şekilde, Δw_{oh} ve Δw_{hi} terimleri sırasıyla h gizli katman ile o çıktı katmanı arasında bulunan bir ağırlık ve i giriş katmanı ile h gizli katman arasında bulunan bir ağırlık için ağırlık modifikasyonudur. N_i, N_H ve N_o sırasıyla i giriş, h gizli ve o çıkış katmanlarında bulunan toplam nöron sayısıdır, $s_o^a(t)$, $s_o^d(t)$, giriş nöronu i ve gizli nöron h için ateşleme dizileridir, $s_o^a(t)$, $s_o^d(t)$ çıkış nöronu o için gerçek spike dizisi ve istenen spike dizisidir . w_{oh}, h gizli ve o çıktı katmanı arasında bulunan ağırlıkların vektörüdür ve a, ağırlık değişiminin doğru yönde gerçekleşmesini sağlayan pozitif, Hebbian olmayan bir terimdir.

3. BÖLÜM

SNN'LERDE DENETİMLİ ÖĞRENME ALGORİTMALARININ PERFORMANSI

Bu bölümde FHN nöron tabanlı, tek çıkışlı, çok katmanlı ileri beslemeli SNN yapıları tasarlanmıştır. Bu yapılar denetimli öğrenme algoritmaları kullanılarak iki farklı veri seti üzerinde eğitilmiştir. Oluşturulan ağlar Multi-STIP ve Multi-ReSuMe algoritmaları kullanılarak Iris veri seti ve Wisconsin Göğüs Kanseri veri seti (WBC) gibi sınıflandırma problemleri üzerine uygulanmıştır. Farklı sayıda gizli katman nöronuna sahip çoklu koşmalar geliştirilmiş ve her ağın performansı analiz edilmiştir. Ağ doğrulukları çoklu çalıştırmalar için ölçülmüş ve her ağın ortalama doğruluğu ve standart sapması hesaplanmıştır. Bu bölümde sunulan SNN yapıları, Bölüm 2'de sunulan XOR problemi için kullanılan FHN nöron modeliyle aynı parametrelerle oluşturulmuştur. (bkz. Tablo 2.2). Her örnek ve bir örnek sonrasındaki arasında, bir epoch uzunluğuna eşit bir gecikme süresi ayarlanmıştır. Her bir epoch için ayrılan zaman dilimi, sinir ağındaki bir nöronun tamamen uyarıldığında toplam 4 ateşleme (yaklaşık 150 ms) vermesine olanak tanıyacak şekilde ayarlanmıştır. Bu simülasyonlarda kullanılan veri setleri kontrol edilmiş ve eksik bilgi içeren örnekler veri kümesinden çıkarılmıştır. Ayrıca, karakter veya dizi içeren herhangi bir alan sayısal olarak kodlanmıştır. Bu durum her dizenin belirli bir sayıya karşılık geldiği anlamına gelmektedir. Veri setlerinde değişiklikler yapıldıktan sonra her bir nöron için ateşleme süreleri bütün epochlar için kaydedilmiştir. Bu kaydedilen bilgiler ile iki algoritma için sinaptik ağırlık değişiklikleri hesaplanmıştır. Giriş katmanında bulunan nöronların ateşleme süreleri 0 ms'ye, gizli katmanda bulunan nöronlar 300 ms'ye ve çıkış nöronları ise 600 ms'ye ayarlanmıştır

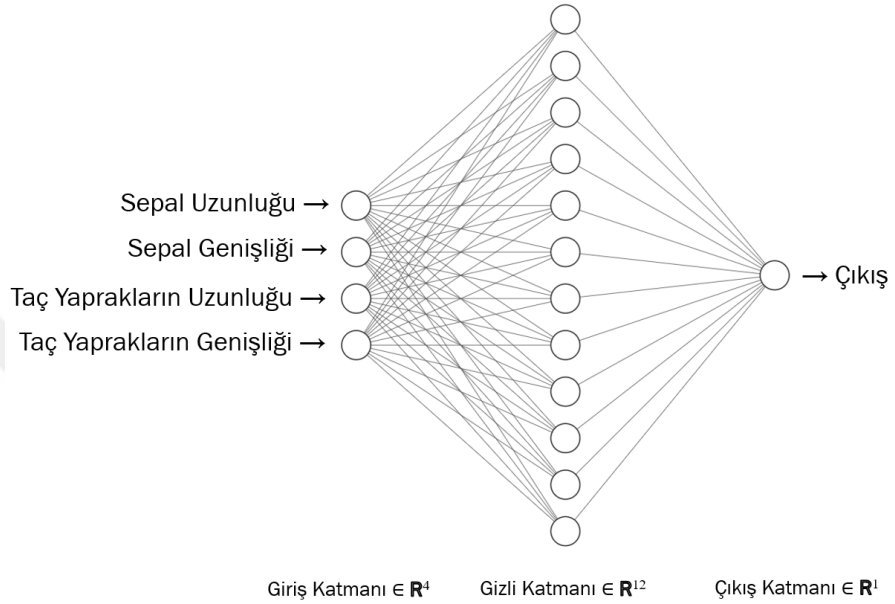
3.1. Iris Veri Setine Dayalı Uygulama

İstatistikçi ve biyolog Ronald Fisher'in adını taşıyan Fisher Iris veri seti sınıflandırma problemine önerilen çözümlerde yaygın olarak kullanılmaktadır. Veri seti Setosa, Versicolor ve Virginica şeklinde üç farklı çiçek türü içermektedir. Veri setinde her bir çiçek türü için 50 örnek, toplamda ise 150 örnek bulunmaktadır. Tablo 3.1'de görüldüğü gibi her örnek için veriler çiçeğin sepallerin uzunluğu, sepallerin genişliği, taç yaprakların uzunluğu ve taçyaprakların genişliği olmak üzere 4 özelliğini temsil etmektedir. Çeşit alanı (iris sınıfı) Setosa=1, Versicolor=0,3 ve Virginica=0 olarak kodlanmıştır. Daha sonra tüm alanlar, ilgili alanlarda bulunan maksimum değere göre normalleştirilir. Her iris sınıfı rastgele seçilir ve daha sonra veri seti, her biri 75 örnek içeren bir eğitim seti ve bir test seti olmak üzere iki gruba bölünür. Eğitim seti, 1. örnek Setosa, 2. örnek Versicolor, 3. örnek Virginica, 4. örnek Setosa,…vb. şeklinde düzenlenmiştir. Öte yandan test seti ise ilk 25 örnek Setosa, ikinci 25 örnek Versicolor ve üçüncü 25 örnek Virginica şeklinde düzenlenmiştir. Her grup için toplam 75 epoch (eğitim ve test) ile simülasyon zaman aralığı [0-22500] ms olarak ayarlanır ve bu da bir epoch için 300 ms'ye denk gelir (150 ms örnek ve 150 ms gecikme). Her sınıfı tanımlayan dört nitelik için dört giriş nöronu ve bir kontrol nöronu tanımlanmıştır. Sisteme beslenen tüm değerler, tam uyarı genliği olarak alınan 'gain=4.5'e eşit bir kazançla çarpılmıştır.

Tablo 3.1 Fisher's Iris veri seti

Sepal Uzunluğu	Sepal Genişliği	Taç Yaprakların Uzunluğu	Taç Yaprakların Genişliği	Çeşit
5,1	3,5	1,4	0,2	Setosa
4,9	0,3	1,4	0,2	Setosa
4,7	3,2	1,3	0,2	Setosa
0,7	3,2	4,7	1,4	Versicolor
6,4	3,2	4,5	1,5	Versicolor
6,9	3,1	4,9	1,5	Versicolor
6,3	3,3	0,6	2,5	Virginica
5,8	2,7	5,1	1,9	Virginica
7,1	0,3	5,9	2,1	Virginica

Her iki algoritmanın simülasyonları 4×12×1 yapısına sahip tam bağlantılı bir ağ Şekil 3.1 için gerçekleştirilmiştir. Her algoritma için kullanılan parametreler Tablo 3.2'te görüldüğü gibi alınmıştır. Sinir ağı için başlangıç ağırlıkları W_0 her durumda [0,19-0,20] aralığında rastgele değerler olarak alınmıştır.



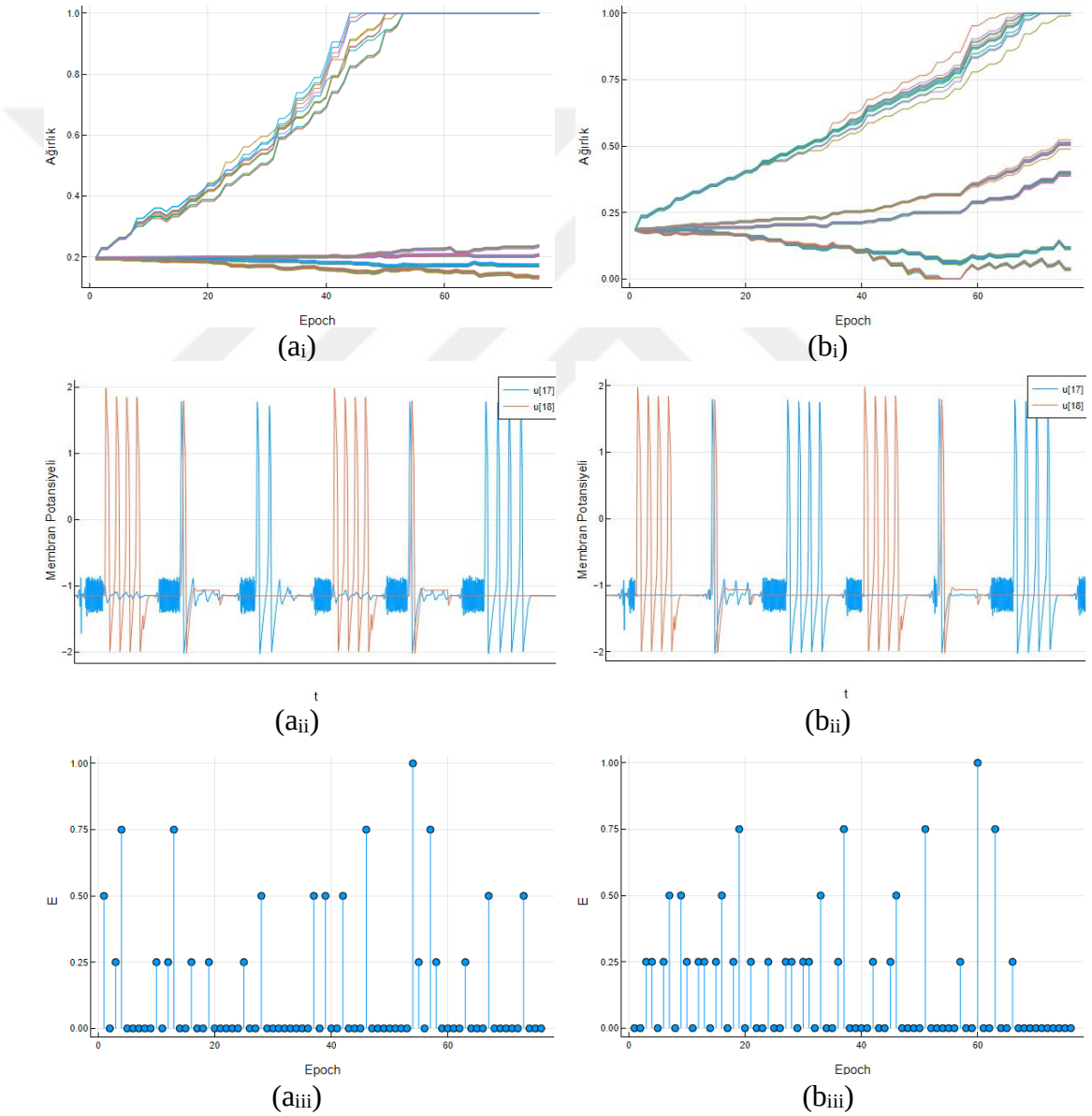
Şekil 3.1 Iris veri setinde kullanılan ağ mimarisi

Tablo 3.2 İris probleminde kullanılan algoritma parametreleri

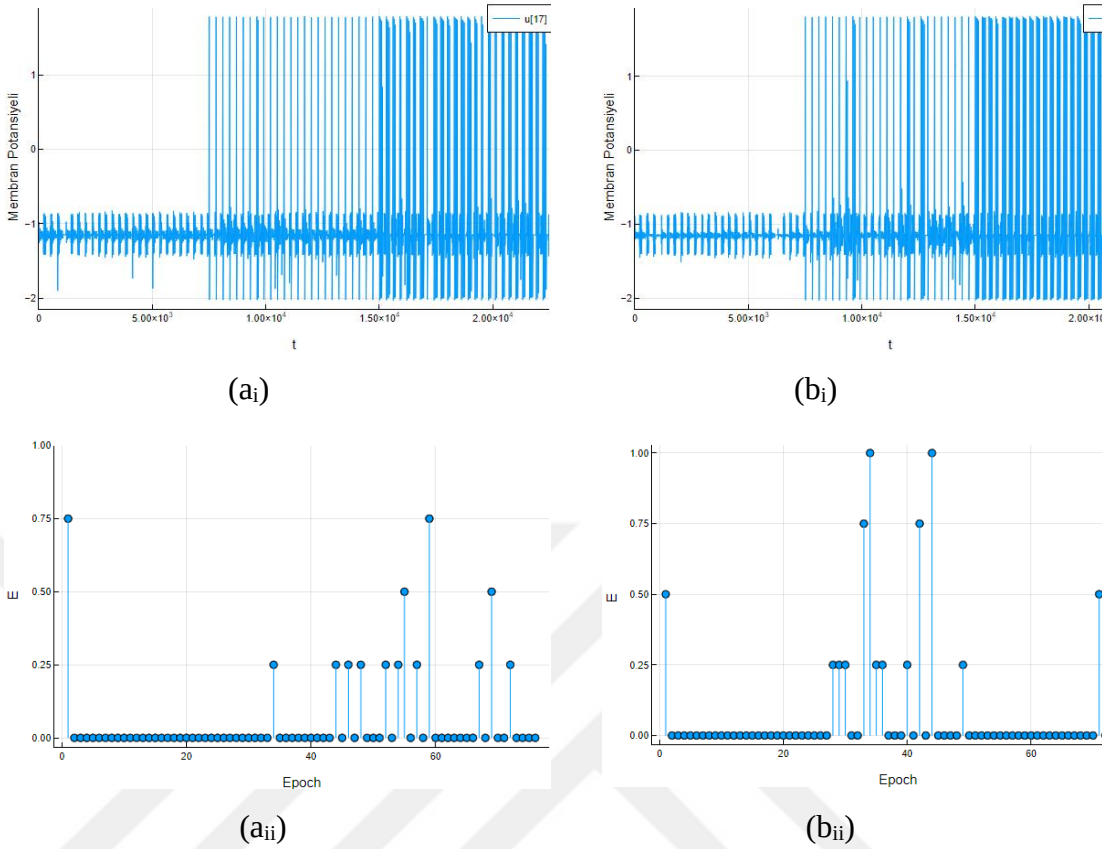
Multi-ReSuMe				Multi-STIP		
A-	A+	a	W_0	σ	H	W_0
0.1	0.4	0.005	[0,19-0,20]	20	0.0120	[0,19-0,20]

Daha sonra eğitim simülasyonlar bu ayarlar kullanılarak gerçekleştirilir ve eğitim sürecinin sonuçları Şekil 3.2'da gösterilmektedir. Multi-ReSuMe algoritmasının sonuçları (a) etiketiyle indekslenirken, Multi-STIP algoritmasının sonuçları (b) ile etiketlenir. Her epoch'taki ağırlıkların gelişimi hem (a_i) hem de (b_i) üzerinde verilmektedir. Tüm simülasyon periyodu boyunca çıkış nöronunun membran potansiyeli kaydedilir ve [20700-22500] ms'ye eşdeğer son altı epoch'a ait sinyaller (a_{ii}) ve (b_{ii})'de çizilir. u[17] etiketiyle indekslenen sinyal, eğitim süreci sırasında sinir ağının gerçek çıktısını, u[18] ise sinir ağından istenen çıktıyı ifade eder. Son olarak eğitim sürecinde her epoch sonunda hata fonksiyonu hesaplanır ve (a_{iii}) ve (b_{iii})'de görselleştirilmiş olarak verilir. Daha sonra yeni elde edilen ağırlık matrisi kaydedilir ve geri kalan kullanılmayan veri seti örnekleri (test grubu) üzerinde test simülasyonları gerçekleştirilir. Bu, W0 = W(epoch=75)

ayarlanarak ve öğrenme süreci devre dışı bırakılarak yapılır. Şekil 3.3, test simülasyonlarının sonuçlanmasından sonra elde edilen sonuçları göstermektedir. Şekil 3.2'ya benzer şekilde, Multi-ReSuMe algoritması sonuçlarına (a) etiketi verilmiştir ve Multi-STIP algoritması (b) ile indekslenmiştir. Simülasyon periyodu [0-22500] ms sırasında çıkış nöronunun sergilediği membran potansiyeli sinyali (ai) ve (bi)'de verilmiştir. Öte yandan hata fonksiyonu epoch bazında verilir ve bu grafiklerden elde edilen sonuçlar daha sonra çalışmanın doğruluğunu hesaplamak için kullanılır.



Şekil 3.2 FHN (4×12×1) ateşlemeli sinir ağına ilişkin Fisher Iris veri setine ait eğitim grubun sonuçları: (a_i),(b_i) zaman içindeki ağırlık gelişimidir, (a_{ii}),(b_{ii}) [20700-22500] ms aralığında çıkış nöronunun gerçek sinyali u[17] ve istenen sinyali u[18], (a_{iii}),(b_{iii}) hata fonksiyonu gelişimi. (a_i)-(a_{iii}) Multi-ReSuMe sonuçları (b_i)-(b_{iii}) Multi-STIP sonuçları



(a_i) (b_i)

(a_{ii}) (b_{ii})

Şekil 3.3 FHN (4×12×1) ateşlemeli sinir ağında Fisher Iris veri setine ait test grubun sonuçları (a_i),(b_i) çıkış nöronunun membran potansiyeli (gerçek), (a_{ii}),(b_{ii}) hata fonksiyonu gelişimi (a_i)-(a_{ii}) Multi-ReSuMe sonuçları (b_i)-(b_{ii}) Multi-STIP sonuçları

Şekil 3.2 (a_i) (b_i) her iki algoritmanın eğitim sürecinin ağırlık gelişim sonuçlarını göstermektedir. Grafiklerde her iki algoritmada nöral ağ eğitildikçe sinaptik ağırlıkların gruplar veya bantlar oluşturma eğiliminde olduğunu göstermektedir. Bu gruplar eğitim süreci boyunca artış ve azalışları göstermede tek bir grup gibi davranırlar. Yalnızca bir grup doyum sınırına ulaşırken diğerleri ağa verilen farklı örnekleri değiştirmeye devam etmektedir. Yukarıda belirtilen parametrelerin kullanılmasından sonra elde edilen çıkış nöronunun membran potansiyeli dinamiklerine bakıldığında, sinir ağlarının bu iki algoritmadan herhangi biri üzerinde eğitilmesinin bastırılmış dinamiklerle sonuçlandığı görülmüştür. Çıkış nöronunun gerçek sinyali, kontrol nöronu tarafından ifade edilen istenen sinyalin tersidir. Şekil 3.2 (a_{ii}) ve (b_{ii}) 'e bakıldığında, kontrol nöronu tamamen etkinleştirildiğinde (4 ateşlenme), çıkış nöronu ateşlenmez. Benzer şekilde, kontrol nöronu ateşlendiğinde (0 artış), çıkış nöronu tamamen etkinleştirilir (dört artış). Kontrol nöronu yarım aktivasyon (bir ateşlenme) gösterdiğinde, çıkış nöronu da yarı aktivasyon

yani yarı bastırma (bir ateşlenme gösterir. Simülasyonlarda kullanılan hata fonksiyonu bahsedilen üç farklı duruma göre belirlenmiştir. Hata fonksiyonu E şu şekilde hesaplanır:

$$E = \begin{cases} \frac{(4-|N_o^d - N_o^a|)}{4} & if\ N_o^d \neq 1 \\ \frac{|N_o^d - N_o^a|}{4} & if\ N_o^d = 1 \end{cases} \quad (3.1)$$

N_o^d, N_o^a terimleri daha önce sırasıyla çıkış nöronunun istenen ateşlenme sayısı ve gerçek ateşlenme sayısı (oluşması gereken veya oluşmaması gereken ateşlenmeler) ile meydana gelebilecek toplam ani artış ateşlenme sayısı arasındaki oranı geri verir. Hata fonksiyonu, Setosa ve Virginica sınıfları ($N_o^d \neq 1$) ve Versicolor sınıfı ($N_o^d = 1$) için doğru oranı verecek şekilde modellenmiştir. Şekil 3.2 (a_{iii}) 'e bakıldığında, sondan (sağdaki) dördüncü epochun 0,5'e eşit bir hatası vardır, bu da o epoch'ta ateşlenmenin meydana gelmemesi gerektiğini göstermektedir. Bu, Şekil 3.2 (a_{ii})'de aynı epocha karşılık gelen sondan dördüncü görüntülenen sinyalde görülmektedir. Bu durumda çıkış sınıfı Virginica'dır ve çıkış nöronunun gerçek sinyalinin toplam dört ateşlenmeye sahip olması gerekmektedir. Ancak görüntülenen sinyalde yalnızca iki artış var ve 2 ateşlenmenin daha oluşması gerekmektedir. Bu yüzden Denklem (3.1)'de tanıtılan hata fonksiyonunu kullanarak bu epochun hatası 0,5 olarak hesaplanmıştır. Şekil 3.3, eğitim süreci tamamlandıktan sonra test grubunun sınıflandırma sonuçlarını göstermektedir. Daha önce de belirtildiği gibi sınıflandırma sırasında oluşan hataların net bir şekilde ayırt edilebilmesi için örnekler Setosa, Versicolor ve Virginica olarak düzenlenmiştir. Hata grafikleri de ilgili sonuçlara eklenmiş ve her iki algoritmanın doğruluğu hesaplanmıştır. Bu koşturmada Multi-ReSuMe algoritması test grubu üzerinde %94,67 doğruluk verirken, Multi-STIP test grubu üzerinde %92,33 doğruluk vermiştir. Burada Multi-ReSuMe algoritması ile eğitilen FHN nöron modeli ile oluşturulan 4×12×1'lik SNN yapısı Iris veri seti sınıflandırma uygulamasında daha başarılı olduğu görülmüştür.

3.2. WBC Veri Setine Dayalı Uygulama

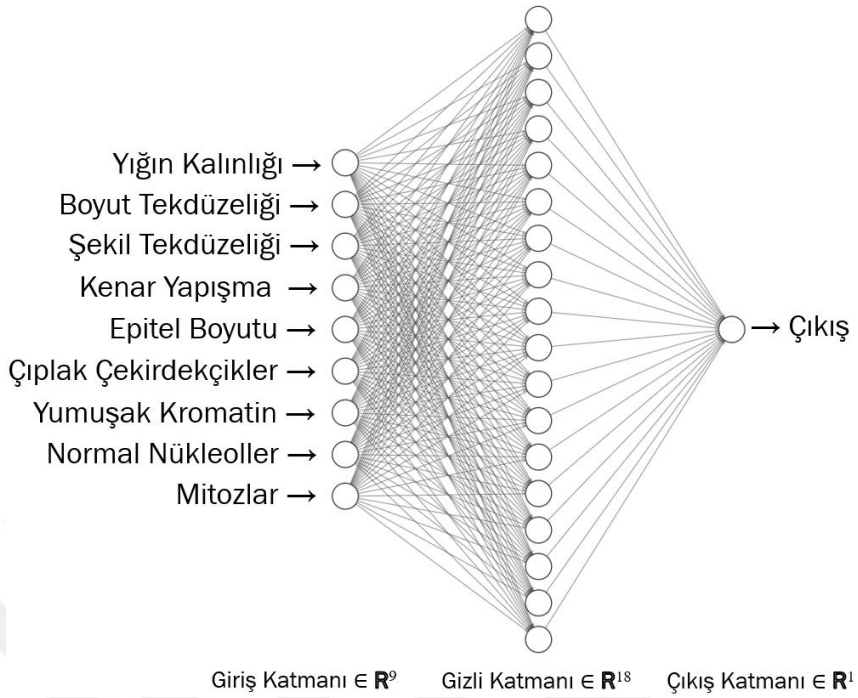
Wisconsin Üniversitesi Hastanelerinden Dr. Wolberg tarafından toplanan Wisconsin Göğüs Kanseri veri seti, model sınıflandırma problemlerinin çözümü için önerilen yöntemleri test etme alanında yaygın olarak kullanılmaktadır. Veri seti, 458 iyi huylu tümör ve 241 kötü huylu tümör içeren 699 örnek içermektedir. Veri setinde bazı verileri eksik olan 16 örnek çıkarılmıştır. Her örnek 9 özellik değeri ve bir çıktı değerinden oluşur (bkz. Tablo 3.3). Her özellik değeri [1,10] aralığında bir tamsayı ile temsil edilir, çıkış değerleri 2 ve 4 sırasıyla iyi huylu tümörü ve kötü huylu tümörü temsil etmektedir.

Tümörün büyüklüğü ve diğer özellikleri gibi dokuz özelliğine göre tümörün iyi veya kötü huylu olup olmadığına karar verilmektedir. İris problemine benzer şekilde veri setinde yer alan değerler, ilgili alanlarda bulunan maksimum değere göre normalize edilmiş ve daha sonra her değer 'gain=4.5' kazançla çarpılmıştır. Ayrıca, her örneğin ve sınıfının 9 özelliğini sırasıyla alacak şekilde 9 giriş nöronu ve bir kontrol nöronu yapılandırılmıştır. Wisconsin Göğüs Kanseri veri seti 342 örnekten oluşan eğitim seti ve 341 örnekten oluşan test setinden oluşmaktadır. Iris veri setindeki gibi epoch başı 300 ms (150 ms örnek ve 150 ms gecikme) ayarlanıp eğitim seti ve test setine toplam simülasyon zaman aralıkları sıra ile [0-102600] ms ve [0-102300] ms atanmıştır. Eğitim grubu rastgele düzenlenirken test grubunda ilk 222 örnek iyi huylu tümör ve son 119 örnek kötü huylu tümörlerden oluşmaktadır.

Tablo 3.3 WBC veri seti

İd	Yığın Kalınlığı	Boyut Tekdüzeliği	Şekil Tekdüzeliği	Kenar Yapışma	Epitel Boyutu	Çıplak Çekirdekçikler	Yumuşak Kromatin	Normal Nükleoller	Mitozlar	Sınıf
1000025	5	1	1	1	2	1	3	1	1	2
1002945	5	4	4	5	7	10	3	2	1	2
1015425	3	1	1	1	2	2	3	1	1	2
1016277	6	8	8	1	3	4	3	7	1	2
1017023	4	1	1	3	2	1	3	1	1	2
1017122	8	10	10	8	7	10	9	7	1	4
1041801	5	3	3	3	2	3	4	4	1	4
1044572	8	7	5	10	7	9	5	5	4	4
1047630	7	4	6	4	6	1	4	3	1	4
1050670	10	7	7	6	4	10	4	1	2	4

Bu veri seti için simülasyon işlemi sırasında alınan parametreler Tablo 3.4'te verilmiştir. Bu simülasyonlar için kullanılan ağ, 9×18×1 yapısına sahip, Şekil 3.4 görüldüğü gibi tamamen bağlantılı bir ağdır. Her durumda sinir ağı için başlangıç ağırlıkları W_0, [0,24-0,25] aralığında rastgele değerler olarak alınır.



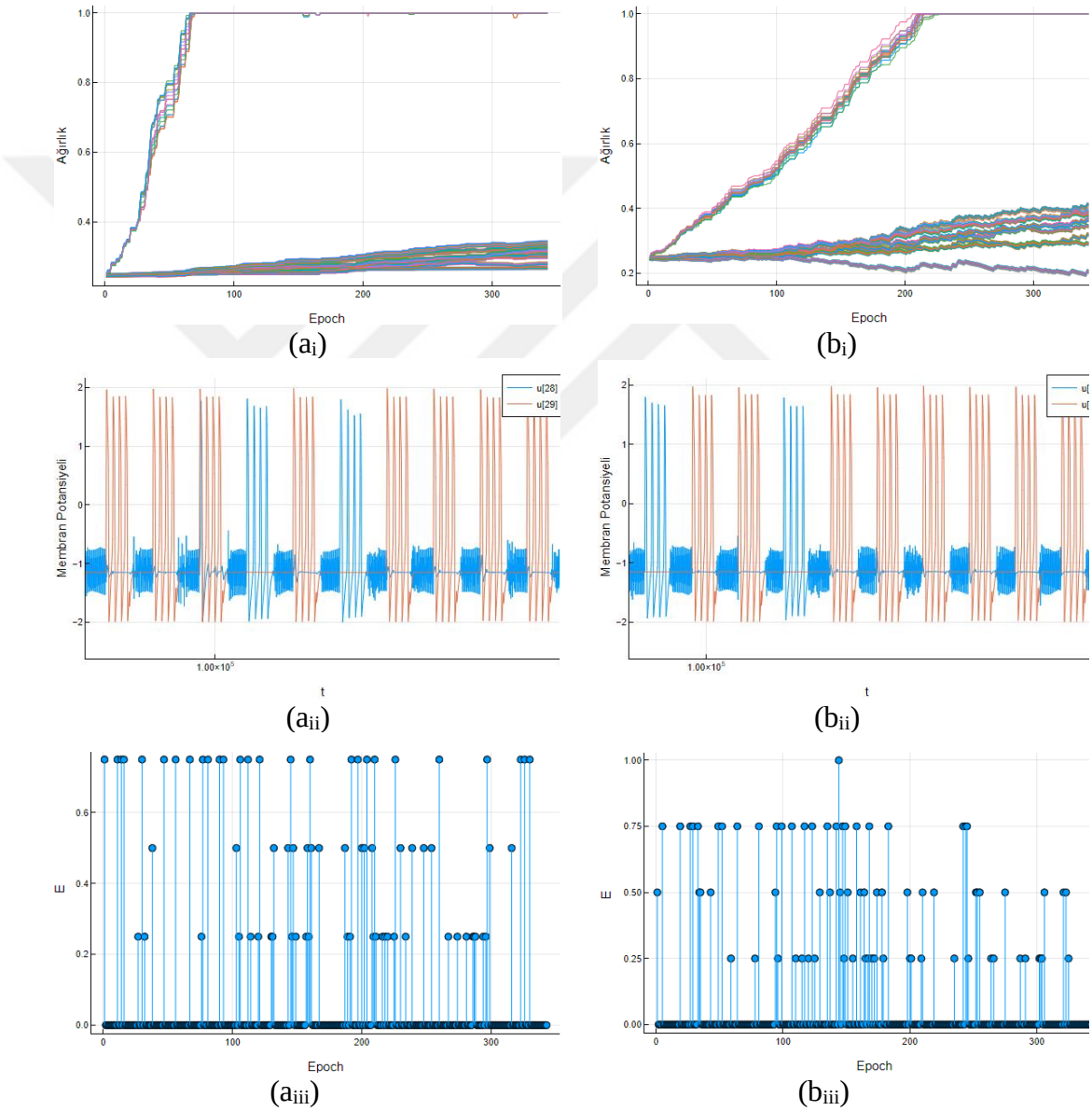
Şekil 3.4 WBC veri setinde kullanılan ağ mimarisi

Tablo 3.4 WBC probleminde kullanılan algoritma parametreleri

Multi-ReSuMe				Multi-STIP		
A_-	A_+	a	W_0	σ	η	W_0
0.1	0.4	0.005	[0,24-0,25]	10	0.0035	[0,24-0,25]

Benzer şekilde eğitim simülasyonlar bu ayarlar kullanılarak gerçekleştirilir ve eğitim sürecinin sonuçları Şekil 3.5'te verilmektedir. Multi-ReSuMe algoritma sonuçları (a) etiketiyle etiketlenip Multi-STIP algoritma sonuçları (b) ile indislenmiştir. Her epoch'taki ağırlıkların gelişimi hem (a_i) hem de (b_i) üzerinde verilmektedir. Tüm simülasyon süreci boyunca çıkış nöronunun membran potansiyeli kaydedilip ve [99600-102600] ms'ye eşdeğer son on epoch'a ait sinyaller (a_{ii}) ve (b_{ii})'de çizdirilmiştir. u[28] etiketiyle indekslenen sinyal, eğitim süreci sırasında sinir ağının gerçek çıktısını, u[29] ise sinir ağından istenen çıktıyı ifade eder. Son olarak eğitim sürecinde her epoch döngüsünün sonunda hata fonksiyonu hesaplanıp (a_{iii}) ve (b_{iii}) indislenen grafikler ile verilir . Daha sonra en son alınan ağırlık değer matrisi kaydedilir ve daha kullanılmayan veri seti örnekleri (test grubu) üzerinde test simülasyonları koşturulur. Bu, W0 = *W(epoch=342)* ayarlanarak ve öğrenme prosesi devre dışı bırakılarak yapılır. Şekil 3.6, test

simülasyonlarının sona eridiğinde elde edilen sonuçları göstermektedir. Şekil 3.2'ya benzer şekilde, Multi-ReSuMe algoritması sonuçlarına (a) etiketi verilmiştir ve Multi-STIP algoritması (b) ile indekslenmiştir. Simülasyon periyodu [0-102300] ms sırasında çıkış nöronunun sergilediği membran potansiyeli sinyali (ai) ve (bi)'de verilmiştir. Öte yandan hata fonksiyonu epoch bazında verilir ve yine de bu grafiklerden elde edilen sonuçlar ile yapılan koşturmanın doğruluğunu hesaplanır.

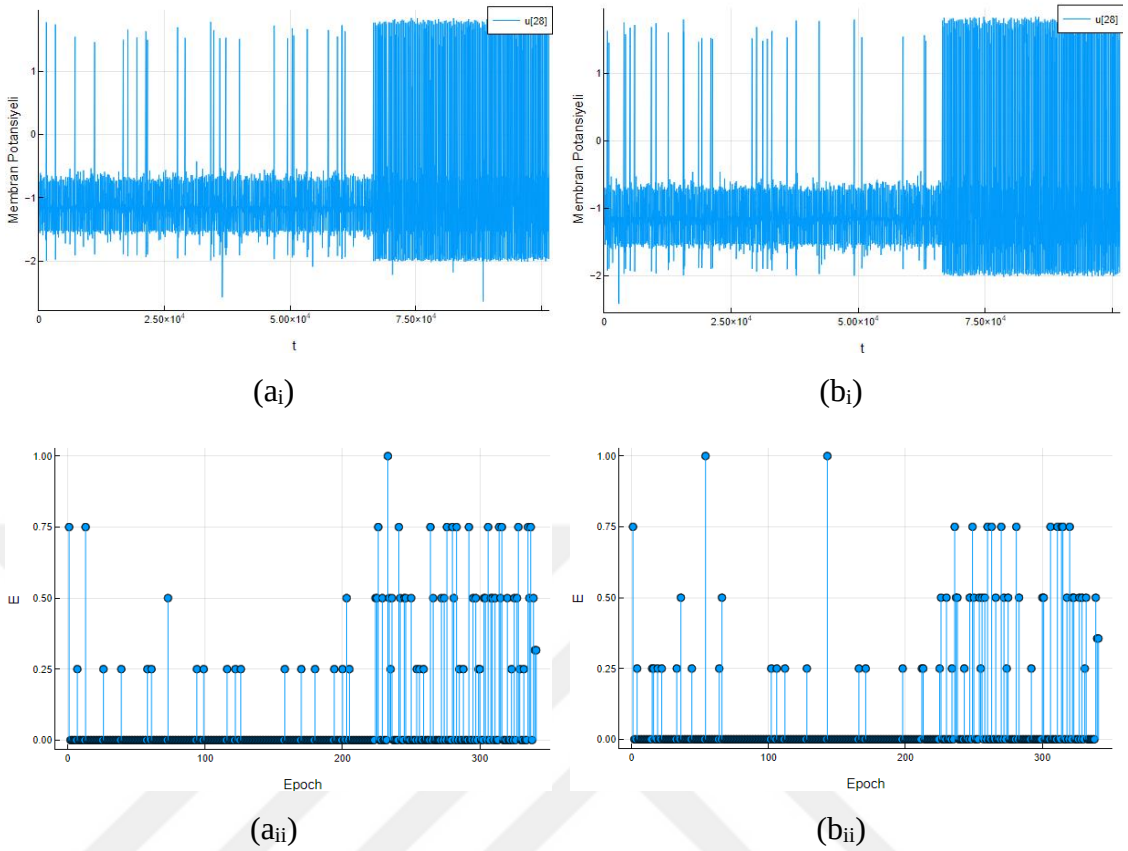


(a_i) (b_i)

(a_{ii}) (b_{ii})

(a_{iii}) (b_{iii})

Şekil 3.5 FHN (9×18×1)ateşlemeli sinir ağına ilişkin WBC veri setine ait eğitim grubun sonuçları: (a_i),(b_i) zaman içindeki ağırlık gelişimidir, (a_{ii}),(b_{ii}) [99600-102600] ms aralığında çıkış nöronunun gerçek sinyali u[28] ve istenen sinyali u[29], (a_{iii}),(b_{iii}) hata fonksiyonu gelişimi. (a_i)-(a_{iii}) Multi-ReSuMe sonuçları (b_i)-(b_{iii}) Multi-STIP sonuçları



(a_i) (b_i)

(a_{ii}) (b_{ii})

Şekil 3.6 FHN (9×18×1) ateşlemeli sinir ağında WBC veri setine ait test grubun sonuçları (a_i),(b_i) çıkış nöronunun membran potansiyeli (gerçek), (a_{ii}),(b_{ii}) hata fonksiyonu gelişimi (a_i)-(a_{ii}) Multi-ReSuMe sonuçları (b_i)-(b_{ii}) Multi-STIP sonuçları

Eğitim süreci sonunda elde edilen sonuçlar Şekil 3.5'te görülmektedir. İris problemine benzer şekilde, sisteme daha fazla veri beslendikçe ağırlıkların zaman içindeki gelişimi gruplar oluşturma eğilimindedir ve yalnızca bir ağırlık grubu üst doyma sınırına ulaşmaktadır. Ayrıca, (a_{ii}),(b_{ii}) şekillerinde gösterilen membran potansiyeli sonuç grafikleri de İris probleminde gözlemlenen aynı durumu göstermektedir. Kontrol nöronu tamamen uyarıldığında, çıkış nöronu ateşlenmez ve bunun tersi olarak kontrol nöronu uyarılmadığında da çıkış nöronunun ateşlendiği görülmektedir. Bu nedenle bu veri seti için Denklem (3.1)'de görülen hata fonksiyonu benzer şekilde geliştirilmiştir. Iris veri setinin Versicolor sınıfında olduğu gibi, WBC veri seti sadece 2 sınıftan oluştuğu için kontrol nöronunun ateşleme sayısının çıkış nöronunun ateşleme sayısına eşit olduğu bir durum mevcut değildir.

$$E = \frac{(4-|N_o^d - N_o^a|)}{4} \tag{3.2}$$

Bu hata fonksiyonu E, daha önce tartışılan önceki karşılığı ile aynı anlama sahiptir. Şekil 3.6 (a_i) ve (b_i) ise eğitim süreci bittikten sonra simüle edilen test grubu sonuçlarını göstermektedir. Minör nöral aktiviteye sahip olan bölge, iyi huylu tümör sınıfına karşılık gelir. Majör nöral aktiviteyi içeren bölge ise kötü huylu tümör sınıfına karşılık gelmektedir. Test sonuçlarına ilişkin hata grafikleri de her algoritma için dahil edilmiş ve aynı şekilde her algoritmanın doğruluğu elde edilmiştir. Bu koşturmada Multi-ReSuMe algoritması test grubu üzerinde %90,36 doğruluk verirken, Multi-STIP test grubu üzerinde %91,51 doğruluk vermiştir. Sonuç olarak Multi-STIP algoritması ile eğitilen FHN nöron modeli ile oluşturulan 9×18×1'lik SNN yapısı Wisconsin Göğüs Kanseri veri seti sınıflandırma uygulamasında daha başarılı olmuştur.

3.3. FHN Tabanlı Farklı Ateşlemeli Sinir Ağı Mimarilerinin Performans İncelemesi

En uygun ağ yapısını bulmak amacıyla, önceki veri setleri için simülasyonlar, gizli katmanda her ağda farklı nöron sayısına sahip farklı sinir ağları için tekrarlanmıştır. Performans analizi için iris veri seti için 4×8×1, 4×12×1 ve 4×24×1 sinir ağları seçilmiştir. Öte yandan WBC veri setinde performans analizi için 9×12×1, 9×18×1 ve 9×27×1 sinir ağları dikkate alınmıştır. Her ağ için simülasyonlar daha önce tanımlanan aynı ayarlar kullanılarak yürütülür. Simülasyonlar her ağ için 30 kez tekrarlanarak her çalıştırmanın doğruluğu kaydedilerek ortalama doğruluk ve standart sapma buna göre hesaplanır. Sonuçlar daha sonra Tablo 3.5 ve 3.6 üzerinde kaydedilip sonuçlar daha iyi anlaşılması için Şekil 3.7'de görülen görsele dönüştürülmüştür. En yüksek doğruluğa sahip ağlar şunlardır:

- Iris probleminde: 4×12×1 (Multi-STIP) ve 4×24×1 (Multi-ReSuMe).
- WBC probleminde: 9×27× 1 (Multi-STIP) ve 9×12×1 (Multi-ReSuMe).

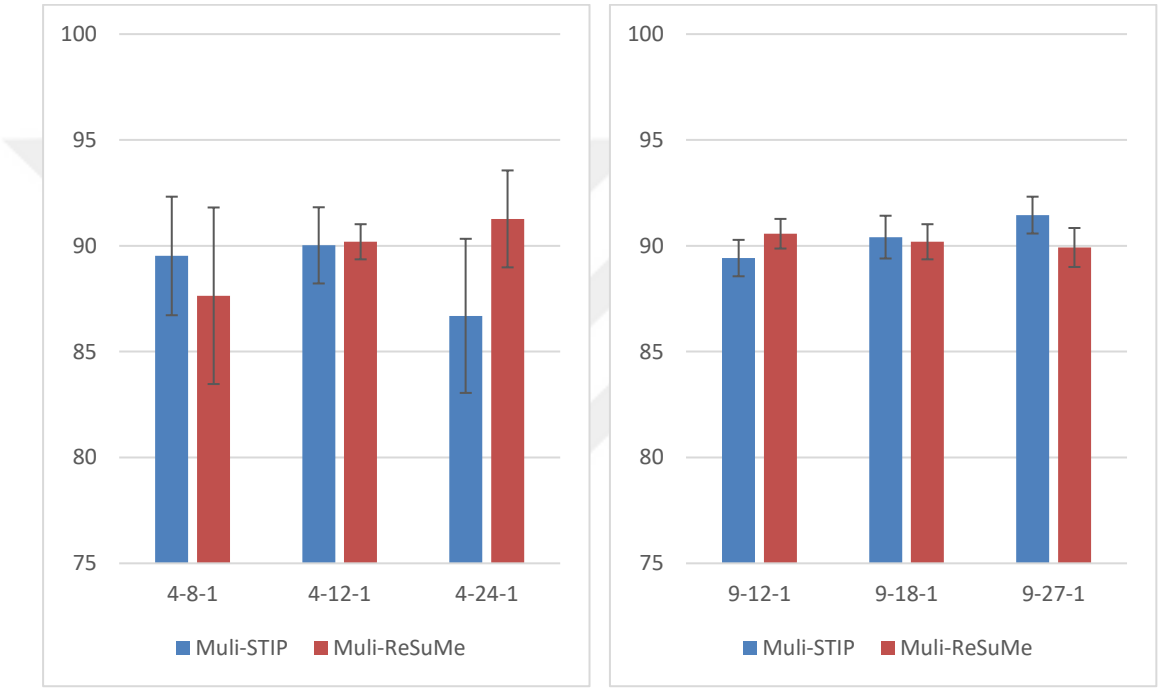
Bu ağlara ilişkin doğruluklar, SRM modeli üzerinde yapılan [71, 89] çalışmada bulunan doğruluklarla birlikte Tablo 3.7'de sunulmuştur.

Tablo 3.5 Farklı mimariler kullanılarak Iris veri setinde elde edilen doğruluklar

Algoritma	Mimarilere Göre Doğruluk Değerleri		
	4-8-1	4-12-1	4-24-1
Multi-STIP	%89,52±2,8	%90,02±1,8	%86,69±3,64
Multi-ReSuMe	%87,64±4,17	%90,19±0,83	%91,27±2,29

Tablo 3.6 Farklı mimariler kullanılarak WBC veri setinde elde edilen doğruluklar

Algoritma	Mimarilere Göre Doğruluk Değerleri		
	9-12-1	9-18-1	9-27-1
Multi-STIP	%89,42±0,86	%90,41±1,01	%91,45±0,87
Multi-ReSuMe	%90,57±0,70	%90,19±0,83	%89,92±0,92



(a) (b)

Şekil 3.7 FHN tabanlı farklı ateşlemeli sinir ağı mimarilerine göre sonuçların doğruluğu: (a) İris veri seti, (b) WBC veri seti.

Tablo 3.7 Test grubunda elde edilen en yüksek doğruluklar.

Veri Set	Yazarlar	Algoritma	Ağ Mimarisi	Doğruluk
Iris	Bu Çalışma	Multi-ReSuMe	4-24-1	%91,27±2,29
		Multi-STIP	4-12-1	%90,02±1,8
	Lin ve ark.[71]	Multi-ReSuMe	4-90-1	%94,0±2,8
		Multi-STIP	4-50-1	%96,7±2,0
	Lin ve ark.[89]	Multi-ReSuMe	4–80–1	%86.3±0.8
		Multi-STIP	4–80–1	%94.2±2.0
WBC	Bu Çalışma	Multi-ReSuMe	9-12-1	%90,57±0,70
		Multi-STIP	9-27-1	%91,45±0,87
	Lin ve ark.[71]	Multi-ReSuMe	9-60-1	%94,8±1,4
		Multi-STIP	9-60-1	%97,1±0,7
	Lin ve ark.[89]	Multi-ReSuMe	9–50–1	%94.4±1.4
		Multi-STIP	9–50–1	%96.7 ± 0.7

4. BÖLÜM

TARTIŞMA-SONUÇ VE ÖNERİLER

SNN ağ yapılarının; kullanılan nöron ve eğitim algoritmaları açısından kapsamlı başarım odaklı performans kıyaslaması literatürde mevcut olmayıp bu ağ yapılarının başarım odaklı performans değerlerini artırmak için literatürde yenilikçi çalışmalara ihtiyaç duyulmaktadır. Bu kapsamda, Bölüm 3'te sonuçların analizi, sinir ağlarını güçlendirmede kullanılan nöron modeli ve algoritma hakkında bir bakış açısı sunulmaktadır. Lin ve ark. çalışmasındaki doğruluk sonuçları yüksek olmasına rağmen, simülasyonlarında biyolojik anlamlılığı düşük SRM nöron modeli kullanmışlardır. Ek olarak gizli katmandaki nöron sayısı yüksek seçmişlerdir (Bkz. Tablo 3.7). Bu çalışmada ise, FHN nöron modeli kullanılarak yapılan simülasyonlar, daha az sayıda nöron kullanarak yüksek doğrulukta sonuçlar elde etmeye odaklanmaktadır. Böylece gizli katmandaki nöron sayısının yüksek olmasının her zaman yüksek sınıflandırma doğruluğu elde edilmesini sağlamadığı sonucuna varılmıştır. Üstelik birçok farklı çalışmada Multi-STIP algoritmasının Multi-ReSuMe algoritmasından daha başarılı olduğu değerlendirilmektedir [14, 71, 89]. Ancak bu tezde bu algoritmalar kullanılan veri setine bağlı olarak farklı performanslar vermektedir. Bunun nedeni SRM nöron modeli yerine FHN nöron modelinin kullanılmasıdır. İris veri seti için, Multi-ReSuMe'nin 4×12×1 ve 4×24×1 mimarilerine sahip sinir ağlarında daha yüksek performans sağladığı bulunurken, Multi-STIP'in Wisconsin göğüs kanseri veri seti için 9×18×1 ve 9×27×1 mimarilerine sahip sinir ağlarında daha iyi performans gösterdiği tespit edilmiştir.

FHN tabanlı SNN ağ yapılarında beslenen giriş değerleri, önceki çalışmalar da olduğu gibi oransal kodlama tekniği kullanılarak kodlanmıştır. Bu durum SNN tarafından görüntülenen her çıktının sınıfı, diğer sınıflardan farklı, özel bir ateşleme oranına sahip olduğu anlamına gelmektedir. Ek olarak tez sonuçlarına göre Wisconsin göğüs kanseri veri seti daha büyük örnek gruplarından oluştuğundan Wisconsin göğüs kanseri veri seti için doğruluk sonuçlarının Iris probleminde bulunandan daha düşük bir standart sapma değerine sahip olduğu kolaylıkla gözlemlenmiştir.

Bu tez çalışmasında denetimli ve denetimsiz öğrenme paradigmaları kullanılarak çeşitli görevleri gerçekleştirebilen SNN ağ mimarilerinin tasarlanması ve bu ağ mimarilerine eğitim için uygulanacak farklı algoritmaların performans değerlendirilmesi amaçlanmaktadır. Bu kapsamda FitzHugh-Nagumo nöron modeli tabanlı ateşleme sinir ağları STDP, Multi-STIP ve Multi-ReSuMe algoritmaları kullanılarak tasarlanmış ve eğitilmiştir. SNN yapıları, i) Denetimsiz öğrenme paradigması altında STDP algoritması kullanılarak XOR problemi, ii) Multi-STIP ve Multi-ReSuMe denetimli algoritmalarını kullanarak İris çiçeği problemi, iii) Multi-STIP ve Multi-ReSuMe denetimli algoritmalarını kullanarak Wisconsin göğüs kanseri problemi olmak üzere 3 farklı problem üzerinde test edilmiştir. Sinir ağları, her ağda kullanılan toplam nöron sayısına bağlı olarak yüksek doğruluklara ulaşmıştır. 4×12×1 mimarisine sahip SNN yapısıyla İris çiçeği veri seti kullanılarak üç İris türünün sınıflandırılması Multi-STIP ve Multi-ReSuMe algoritmalarıyla gerçekleştirilerek sırasıyla ortalama %90,02±1,8 ve %90,19±0,83 başarı oranı elde edilmiştir. 9×18×1 SNN yapısıyla Wisconsin göğüs kanseri veri seti kullanılarak Multi-STIP ve Multi-ReSuMe algoritmalarıyla gerçekleştirilerek göğüs kanseri tespiti yapılmıştır. Sırasıyla ortalama %90,41±1,01ve %90,19±0,83 başarı oranı elde edilmiştir. Elde edilen sonuçlara göre Multi-ReSuMe algoritmasının daha başarılı olduğu görülmüştür. Bunun yanı sıra Wisconsin göğüs kanseri veri seti üzerinde ise elde edilen sonuçlarda Multi-STIP algoritmasıyla daha iyi sonuç alındığı gözlenmiştir. Ayrıca her bir veri setinde farklı yapılar kullanılmıştır ve tüm kullanılan yapıların performansı kıyaslanıp söz konusu veri sete göre en uygun olan yapı teyit edilmiştir. İris veri seti için Multi-ReSuMe algoritması kullanıldığında en uygun yapının 4×24×1 olduğu, Wisconsin göğüs kanseri veri seti için ise Multi-STIP algoritmasının 9×27×1 yapısının uygun olduğu çıkarımı yapılmıştır.

KAYNAKÇA

1. McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. **The bulletin of mathematical biophysics**, **5**(4) : 115-133.

2. Cheng, B., & Titterington, D. M. (1994). Neural networks: A review from a statistical perspective. **Statistical science**, **9** (1) : 2-30.

3. Marsalli, M. (2006). Mcculloch-pitts neurons, *The 2008 Annual Meeting of the consortium on cognitive science instruction (ccsi)*, s. 1162.

4. Cios, K. J. (2018). Deep neural networks—A brief history, 183-200. *In*: Advances in Data Analysis with Computational Intelligence Methods: Dedicated to Professor Jacek Żurada (Eds: A. Gawęda, J. Kacprzyk, L. Rutkowski, G.Yen). Springer Cham.

5. Hinton, G. E., Osindero, S., & Teh, Y.-W. (2006). A fast learning algorithm for deep belief nets. **Neural computation**, **18**(7), 1527-1554.

6. Abiodun, O. I., Jantan, A., Omolara, A. E., Dada, K. V., Mohamed, N. A., & Arshad, H. (2018). State-of-the-art in artificial neural network applications: A survey. **Heliyon**, **4**(11), e00938.

7. Dampfhoffer, M., Mesquida, T., Valentian, A., & Anghel, L. (2022). Are SNNs really more energy-efficient than ANNs? An in-depth hardware-aware study. **IEEE Transactions on Emerging Topics in Computational Intelligence**, **7**(3), 731-741.

8. Kim, Y., Chough, J., & Panda, P. (2022). Beyond classification: directly training spiking neural networks for semantic segmentation**. Neuromorphic Computing and Engineering**, **2**(4), 044015.

9. Davidson, S., & Furber, S. B. (2021). Comparison of artificial and spiking neural networks on digital hardware. **Frontiers in Neuroscience**, **15**, 651141.

10. Ghosh-dastidar, S., & Adeli, H. (2009). Spiking neural networks. **International Journal of Neural Systems**, 19(04), 295-308.

11. Nandakumar, S. R., Kulkarni, S. R., Babu, A. V, & Rajendran, B. (2018). Building brain-inspired computing systems: examining the role of nanoscale devices. **IEEE Nanotechnology Magazine**, **12**(3), 19-35.

12. Camuñas-Mesa, L. A., Linares-Barranco, B., & Serrano-Gotarredona, T. (2019). Neuromorphic spiking neural networks and their memristor-CMOS hardware implementations. **Materials**, **12**(17), 2745.

13. Javanshir, A., Nguyen, T. T., Mahmud, M. A. P., & Kouzani, A. Z. (2022). Advancements in algorithms and neuromorphic hardware for spiking neural networks. **Neural Computation**, **34**(6), 1289-1328.

14. Wang, X., Lin, X., & Dang, X. (2020). Supervised learning in spiking neural networks: A review of algorithms and evaluations. **Neural Networks**, **125**, 258-280.

15. Pham, Q. T., Nguyen, T. Q., Hoang, P. C., Dang, Q. H., Nguyen, D. M., & Nguyen, H. H. (2021). A review of SNN implementation on FPGA. *2021 international conference on multimedia analysis and pattern recognition (MAPR)*, IEEE, (ss. 1-6).

16. Gütig, R., & Sompolinsky, H. (2006). The tempotron: a neuron that learns spike timing–based decisions. **Nature Neuroscience**, **9**(3), 420-428.

17. Zhou, P., Choi, D.-U., Eshraghian, J. K., & Kang, S.-M. (2022). A fully memristive spiking neural network with unsupervised learning. *2022 IEEE International Symposium on Circuits and Systems (ISCAS)*, IEEE, (ss. 634-638).

18. Montemurro, M. A., Rasch, M. J., Murayama, Y., Logothetis, N. K., & Panzeri, S. (2008). Phase-of-firing coding of natural visual stimuli in primary visual cortex. **Current biology**, **18**(5), 375-380.

19. Hodgkin, A. L., & Huxley, A. F. (1952). A quantitative description of membrane current and its application to conduction and excitation in nerve. **The Journal of Physiology**, **117**(4), 500.

20. Morris, C., & Lecar, H. (1981). Voltage oscillations in the barnacle giant muscle fiber. **Biophysical Journal**, **35**(1), 193-213.

21. Hindmarsh, J. L., & Rose, R. M. (1982). A model of the nerve impulse using two first-order differential equations. **Nature**, **296**(5853), 162-164.

22. FitzHugh, R. (1961). Impulses and physiological states in theoretical models of nerve membrane. **Biophysical Journal**, **1**(6), 445-466.

23. Taherkhani, A., Belatreche, A., Li, Y., Cosma, G., Maguire, L. P., & McGinnity, T. M. (2020). A review of learning in biologically plausible spiking neural networks. **Neural Networks**, **122**, 253-272.

24. Kim, R., Li, Y., & Sejnowski, T. J. (2019). Simple framework for constructing functional spiking recurrent neural networks. **Proceedings of the National Academy of Sciences**, **116**(45), 22811-22820.

25. Wang, C., Yan, H., Huang, W., Sheng, W., Wang, Y., Fan, Y.-S., … Chen, H. (2023). Neural encoding with unsupervised spiking convolutional neural network. **Communications Biology**, **6**(1), 880.

26. Mohemmed, A., & Kasabov, N. (2012). Incremental learning algorithm for spatio-temporal spike pattern classification. *The 2012 International Joint Conference on Neural Networks (IJCNN)*, IEEE, (ss. 1-6).

27. Zhao, D., Zeng, Y., Zhang, T., Shi, M., & Zhao, F. (2020). GLSNN: A Multi-Layer Spiking Neural Network Based on Global Feedback Alignment and Local STDP Plasticity. **Frontiers in Computational Neuroscience**, 14, 576841.

28. Morales, E. F., & Escalante, H. J. (2022). Chapter 6 - A brief introduction to supervised, unsupervised, and reinforcement learning, ss. 111-129. *In:* Biosignal Processing and Classification Using Computational Learning and Intelligence (Eds: A. A. Torres-García, C. A. Reyes-García, L. Villaseñor-Pineda, & O. Mendoza-Montoya). Academic Press.

29. Van Engelen, J. E., & Hoos, H. H. (2020). A survey on semi-supervised learning. **Machine Learning**, **109**(2), 373-440.

30. Tavanaei, A., Ghodrati, M., Kheradpisheh, S. R., Masquelier, T., & Maida, A. (2019). Deep learning in spiking neural networks. **Neural Networks**, **111**, 47-63.

31. Meftah, B., Lézoray, O., Chaturvedi, S., Khurshid, A. A., & Benyettou, A. (2013). Image Processing with Spiking Neuron Networks, ss. 525-544. *In:* Artificial Intelligence, Evolutionary Computing and Metaheuristics: In the Footsteps of Alan Turing (Eds: X.-S. Yang). Springer Berlin Heidelberg.

32. Liu, F., Zhao, W., Chen, Y., Wang, Z., Yang, T., & Jiang, L. (2021). SSTDP: Supervised spike timing dependent plasticity for efficient spiking neural network training. **Frontiers in Neuroscience**, **15**, 756876.

33. Mostafa, H. (2017). Supervised learning based on temporal coding in spiking neural networks. **IEEE Transactions on Neural Networks and Learning Systems**, **29**(7), 3227-3235.

34. Kulkarni, S. R., & Rajendran, B. (2018). Spiking neural networks for handwritten digit recognition—Supervised learning and network optimization. **Neural Networks**, **103**, 118-127.

35. Urbanczik, R., & Senn, W. (2009). A gradient learning rule for the tempotron. **Neural Computation**, **21**(2), 340-352.

36. Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. **Nature**, **323**(6088), 533-536.

37. Stewart, K. M., & Neftci, E. O. (2022). Meta-learning spiking neural networks with surrogate gradient descent. **Neuromorphic Computing and Engineering**, **2**(4), 044002.

38. Guo, W., Fouda, M. E., Eltawil, A. M., & Salama, K. N. (2023). Efficient training of spiking neural networks with temporally-truncated local backpropagation through time. **Frontiers in Neuroscience**, **17**, 1047008.

39. Zhang, M., Qu, H., Belatreche, A., & Xie, X. (2018). EMPD: An efficient membrane potential driven supervised learning algorithm for spiking neurons. **IEEE Transactions on Cognitive and Developmental Systems**, **10**(2), 151-162.

40. Taherkhani, A., Belatreche, A., Li, Y., Cosma, G., Maguire, L. P., & McGinnity, T. M. (2020). A review of learning in biologically plausible spiking neural networks. **Neural Networks**, **122**, 253-272.

41. Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. **Annals of Eugenics**, **7**(2), 179-188.

42. Bennett, K. P., & Mangasarian, O. L. (1992). Robust linear programming discrimination of two linearly inseparable sets. **Optimization Methods and Software**, **1**(1), 23-34.

43. Parker, A. J., & Newsome, W. T. (1998). Sense and the single neuron: probing the physiology of perception. **Annual Review of Neuroscience**, **21**(1), 227-277.

44. Bean, B. P. (2007). The action potential in mammalian central neurons. **Nature Reviews Neuroscience, 8**(6), 451-465.

45. Wang, Y.-H., Gong, T.-C., Ding, Y.-X., Li, Y., Wang, W., Chen, Z.-A., … Luo, Q. (2022). Redox memristors with volatile threshold switching behavior for neuromorphic computing. **Journal of Electronic Science and Technology**, **20**(4), 100177.

46. Wilson, H. R., & Cowan, J. D. (1972). Excitatory and inhibitory interactions in localized populations of model neurons. **Biophysical Journal**, **12**(1), 1-24.

47. Jirsa, V. K., & Haken, H. (1996). Field theory of electromagnetic brain activity. **Physical Review Letters**, **77**(5), 960.

48. Robinson, P. A., Rennie, C. J., & Wright, J. J. (1997). Propagation and stability of waves of electrical activity in the cerebral cortex. **Physical Review E**, **56**(1), 826.

49. Hindmarsh, J. L., Rose, R. M., & Huxley, A. F. (1984). A model of neuronal bursting using three coupled first order differential equations. **Proceedings of the Royal Society of London. Series B. Biological Sciences**, **221**(1222), 87-102.

50. Gerstner, W., & Kistler, W. M. (2002). Spiking Neuron Models: Single Neurons, Populations, Plasticity. Cambridge university press, ss. 480.

51. Abbott, L. F. (1999). Lapicque's introduction of the integrate-and-fire model neuron (1907). **Brain Research Bulletin**, **50**(5), 303-304.

52. Nagumo, J., Arimoto, S., & Yoshizawa, S. (1962). An active pulse transmission line simulating nerve axon. **Proceedings of the IRE**, **50**(10), 2061-2070.

53. Turrigiano, G. G., Leslie, K. R., Desai, N. S., Rutherford, L. C., & Nelson, S. B. (1998). Activity-dependent scaling of quantal amplitude in neocortical neurons. **Nature**, **391**(6670), 892-896.

54. Izhikevich, E. M., Gally, J. A., & Edelman, G. M. (2004). Spike-timing Dynamics of Neuronal Groups. **Cerebral Cortex**, **14**(8), 933-944.

55. LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. **Nature**, **521**(7553), 436-444.

56. Yang, R., Huang, H.-M., & Guo, X. (2019). Memristive synapses and neurons for bioinspired computing. **Advanced Electronic Materials**, **5**(9), 1900287.

57. Chua, L. (1971). Memristor-The missing circuit element. **IEEE Transactions on Circuit Theory**, **18**(5), 507-519.

58. Xu, Q., Tan, X., Zhu, D., Chen, M., Zhou, J., Wu, H., & Pham, V. T. (2020). Synchronous behavior for memristive synapse-connected chay twin-neuron network and hardware implementation. **Mathematical Problems in Engineering**, **2020**(1), 8218740.

59. Xu, Y., Jia, Y., Ma, J., Alsaedi, A., & Ahmad, B. (2017). Synchronization between neurons coupled by memristor. **Chaos, Solitons and Fractals**, **104**, 435-442.

60. Abdalla, O., Baran, A. Y., Korkmaz, N., & Kılıç, R. (2024). Investigation of Synchronization Control of the Memristive/Resistive-Coupled Neural Network with Noise Effect. **International Journal of Bifurcation and Chaos**, **34**(08).

61. Bao, H., Zhang, Y., Liu, W., & Bao, B. (2020). Memristor synapse-coupled memristive neuron network: synchronization transition and occurrence of chimera. **Nonlinear Dynamics**, **100**(1), 937-950.

62. Auge, D., Hille, J., Mueller, E., & Knoll, A. (2021). A survey of encoding techniques for signal processing in spiking neural networks. **Neural Processing Letters**, **53**(6), 4693-4710.

63. Guo, W., Fouda, M. E., Eltawil, A. M., & Salama, K. N. (2021). Neural coding in spiking neural networks: a comparative study for robust neuromorphic systems**.** **Frontiers in Neuroscience**, **15**, 638474.

64. Adrian, E. D., & Zotterman, Y. (1926). The impulses produced by sensory nerve endings. **The Journal of Physiology**, **61**(4), 465-483.

65. Gollisch, T., & Meister, M. (2008). Rapid neural coding in the retina with relative spike latencies. **Science**, **319**(5866), 1108-1111.

66. Jordan, M. I., & Mitchell, T. M. (2015). Machine learning: Trends, perspectives, and prospects. **Science**, **349**(6245), 255-260.

67. Caporale, N., & Dan, Y. (2008). Spike timing–dependent plasticity: a Hebbian learning rule. **Annual Review of Neuroscience**, **31**(1), 25-46.

68. Dike, H. U., Zhou, Y., Deveerasetty, K. K., & Wu, Q. (2018). Unsupervised learning based on artificial neural network: a review. *2018 IEEE International Conference on Cyborg and Bionic Systems (CBS)*, IEEE, (ss. 322-327).

69. Cunningham, P., Cord, M., & Delany, S. J. (2008). Supervised learning, ss. 21-49. *In:* Machine learning techniques for multimedia: case studies on organization and retrieval (Eds: M. Cord, P. Cunningham). Springer Berlin, Heidelberg.

70. Wang, X., Wang, S., Liang, X., Zhao, D., Huang, J., Xu, X., … Miao, Q. (2022). Deep reinforcement learning: A survey. **IEEE Transactions on Neural Networks and Learning Systems**, **35**(4), 5064-5078.

71. Lin, X., Wang, X., & Hao, Z. (2017). Supervised learning in multilayer spiking neural networks with inner products of spike trains. **Neurocomputing**, **237**, 59-70.

72. Bohte, S. M., Kok, J. N., & La Poutre, H. (2002). Error-backpropagation in temporally encoded networks of spiking neurons. **Neurocomputing**, **48**(1-4), 17-37.

73. Tavanaei, A., & Maida, A. (2019). BP-STDP: Approximating backpropagation using spike timing dependent plasticity. **Neurocomputing**, **330**, 39-47.

74. Querlioz, D., Bichler, O., Dollfus, P., & Gamrat, C. (2013). Immunity to device variations in a spiking neural network with memristive nanodevices. **IEEE Transactions on Nanotechnology**, **12**(3), 288-295.

75. Wade, J. J., McDaid, L. J., Santos, J. A., & Sayers, H. M. (2010). SWAT: A spiking neural network training algorithm for classification problems. **IEEE Transactions on Neural Networks**, **21**(11), 1817-1830.

76. Sporea, I., & Grüning, A. (2013). Supervised learning in multilayer spiking neural networks. **Neural Computation**, **25**(2), 473-509.

77. Sejnowski, T. J., & Tesauro, G. (1989). The Hebb rule for synaptic plasticity: algorithms and implementations, ss. 94-103. *In:* Neural models of plasticity (Eds: J. Byrne, W. Berry). Academic Press.

78. Gütig, R., Aharonov, R., Rotter, S., & Sompolinsky, H. (2003). Learning input correlations through nonlinear temporally asymmetric Hebbian plasticity. **Journal of Neuroscience**, **23**(9), 3697-3714.

79. Gerstner, W., Kempter, R., van Hemmen, J. L., & Wagner, H. (1996). A neuronal learning rule for sub-millisecond temporal coding. **Nature**, **383**(6595), 76-78.

80. Kempter, R., Gerstner, W., & van Hemmen, J. L. (1999). Hebbian learning and spiking neurons. **Physical Review E**, **59**(4), 4498-4514.

81. Song, S., Miller, K. D., & Abbott, L. F. (2000). Competitive Hebbian learning through spike-timing-dependent synaptic plasticity. **Nature Neuroscience**, **3**(9), 919-926.

82. Zhang, L. I., Tao, H. W., Holt, C. E., Harris, W. A., & Poo, M. (1998). A critical window for cooperation and competition among developing retinotectal synapses. **Nature**, **395**(6697), 37-44.

83. Masquelier, T., & Thorpe, S. J. (2007). Unsupervised learning of visual features through spike timing dependent plasticity. **PLOS Computational Biology**, **3**(2), e31.

84. Ferré, P., Mamalet, F., & Thorpe, S. J. (2018). Unsupervised feature learning with winner-takes-all based stdp. **Frontiers in computational neuroscience**, **12**, 24.

85. Majani, E., Erlanson, R., & Abu-Mostafa, Y. (1988). On the k-winners-take-all network. *Advances in neural information processing systems*, 9.

86. Park, I., Seth, S., Paiva, A., Li, L., & Principe, J. (2013). Kernel methods on spike train space for neuroscience: a tutorial. **IEEE Signal Processing Magazine**, **30**(4), 149-160.

87. Park, I. M., Seth, S., Rao, M., & Príncipe, J. C. (2012). Strictly positive-definite spike train kernels for point-process divergences. **Neural Computation**, **24**(8), 2223-2250.

88. Ponulak, F., & Kasiński, A. (2010). Supervised learning in spiking neural networks with resume: sequence learning, classification, and spike shifting**. Neural Computation**, **22**(2), 467-510.

89. Lin, X., Zhang, M., & Wang, X. (2021). Supervised learning algorithm for multilayer spiking neural networks with long-term memory spike response model. **Computational Intelligence and Neuroscience**, **2021**(1), 1-16.

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı: Obeid Mohamednoor Obeid ABDALLA
Uyruğu: Sudan
Doğum Tarihi ve Yeri:
Medeni Durum:
e-mail:
Yazışma Adresi:

EĞİTİM

Derece	Kurum	Mezuniyet Tarihi
Yüksek Lisans	Erciyes Üniversitesi, Elektrik Elektronik Mühendisliği	2024
Lisans	Erciyes Üniversitesi, Makine Mühendisliği	2022
Lisans	Erciyes Üniversitesi, Elektrik Elektronik Mühendisliği	2022
Lise	Bahri Model Lisesi, Hartum/SUDAN	2015

YABANCI DİL

Arapça (Anadil)
Türkçe
İngilizce

YAYINLAR

1. Abdalla, O., Baran, A. Y., Korkmaz, N., & Kılıç, R. (2024). Investigation of Synchronization Control of the Memristive/Resistive-Coupled Neural Network with Noise Effect. International Journal of Bifurcation and Chaos, 2450099.