



**YAPAY SİNİR AĞLARININ DALGACIK
DÖNÜŞÜMLÜ AKTİVASYON FONKSİYONLARI
İLE İYİLEŞTİRİLMESİ**

Hatice OKUMUŞ

Danışman: Doç. Dr. Mesut KARABACAK
Yüksek Lisans Tezi
Matematik Ana Bilim Dalı
2025

(Her hakkı saklıdır.)

T.C.
ATATÜRK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
MATEMATİK ANA BİLİM DALI

**YAPAY SİNİR AĞLARININ DALGACIK DÖNÜŞÜMLÜ AKTİVASYON
FONKSİYONLARI İLE İYİLEŞTİRİLMESİ**

(Improvement of Artificial Neural Networks With Wavelet Transform Activation Functions)

YÜKSEK LİSANS TEZİ

Hatice OKUMUŞ

Danışman: Doç. Dr. Mesut KARABACAK

Erzurum
Ocak, 2025



FEN BİLİMLERİ ENSTİTÜSÜ
Graduate School of Natural and
Applied Sciences

T.C.
ATATÜRK ÜNİVERSİTESİ
Fen Bilimleri Enstitüsü Müdürlüğü

TEZ KABUL VE ONAY TUTANAĞI

YAPAY SİNİR AĞLARININ DALGACIK DÖNÜŞÜMLÜ AKTİVASYON FONKSİYONLARI İLE İYİLEŞTİRİLMESİ

Doç. Dr. Mesut KARABACAK danışmanlığında, Hatice OKUMUŞ tarafından hazırlanan bu çalışma, 24/01/2025 tarihinde aşağıdaki jüri tarafından Matematik Anabilim Dalı, Uygulamalı Matematik Bilim Dalı'nda Yüksek Lisans tezi olarak **oybirliği / oy çokluğu (.../...)** ile kabul edilmiştir.

Jüri Başkanı:	Prof. Dr. Murat SUBAŞI <i>Atatürk Üniversitesi</i>	Aslı Islak İmzalıdır
Danışman:	Doç. Dr. Mesut KARABACAK <i>Atatürk Üniversitesi</i>	Aslı Islak İmzalıdır
Jüri Üyesi:	Dr. Öğr. Üyesi. Muhammed YİĞİDER <i>Erzurum Teknik Üniversitesi</i>	Aslı Islak İmzalıdır

Enstitü Yönetim
Kurulunun .../.../.... tarih
ve sayılı kararı.

Bu tezin Atatürk Üniversitesi Lisansüstü Eğitim ve Öğretim Yönetmeliği'nin ilgili maddelerinde belirtilen şartları yerine getirdiğini onaylarım.

Prof. Dr. Alper NUHOĞLU
Enstitü Müdürü

Aslı Islak İmzalıdır

Not: Bu tezde kullanılan özgün ve başka kaynaklardan yapılan bildiriş, çizelge, şekil ve fotoğrafların kaynak olarak kullanımı, 5846 sayılı Fikir ve Sanat Eserleri Kanunundaki hükümlere tabidir.



T.C.
ATATÜRK ÜNİVERSİTESİ REKTÖRLÜĞÜ
FEN BİLİMLERİ ENSTİTÜSÜ MÜDÜRLÜĞÜ

ETİK BİLDİRİM VE İNTİHAL BEYAN FORMU

Yüksek Lisans Tezi olarak *Doç. Dr. Mesut KARABACAK* danışmanlığında sunulan “Yapay Sinir Ağlarının Dalgacık Dönüşümlü Aktivasyon Fonksiyonları ile İyileştirilmesi” başlıklı çalışmanın tarafımızdan bilimsel etik ilkelere uyularak yazıldığını, yararlanılan eserlerin kaynakçada gösterildiğini, Fen Bilimleri Enstitüsü tarafından belirlenmiş olan Turnitin Programı benzerlik oranlarının aşılmadığını ve aşağıdaki oranlarda olduğunu beyan ederiz.

Tez Bölümleri	Tezin Benzerlik Oranı (%)	Maksimum Oran (%)
Giriş	3	30
Kuramsal Temeller	15	30
Materyal ve Metot	7	35
Araştırma Bulguları ve Tartışma	0	20
Sonuçlar ve Öneriler	0	20
Tezin Geneli	12	25

Not: Yedi kelimeye kadar benzerlikler ile Başlık, Kaynakça, İçindekiler, Teşekkür, Dizin ve Ekler kısımları tarama dışı bırakılabilir. Yukarıdaki azami benzerlik oranları yanında tek bir kaynaktan olan benzerlik oranlarının %5'den büyük olmaması gerekir.

Sunulan bilgilerin doğru olduğunu, aksi halde doğacak hukuki sorumlulukları kabul ettiğimizi beyan ederiz.

Tez Yazarı (Öğrenci)	Tez Danışmanı
Hatice OKUMUŞ	Doç. Dr. Mesut KARABACAK
24.1.2025	24.1.2025
Aslı Islak İmzalıdır	Aslı Islak İmzalıdır

* Tez ile ilgili YÖKTEZ’de yayımlanmasına ilişkin bir engelleme var ise aşağıdaki alanı doldurunuz.

☐ Tezle ilgili patent başvurusu yapılması / patent alma sürecinin devam etmesi sebebiyle Enstitü Yönetim Kurulunun/....../.... tarih ve sayılı kararı ile teze erişim 2 (iki) yıl süreyle engellenmiştir.

☐ Enstitü Yönetim Kurulunun/....../.... tarih ve sayılı kararı ile teze erişim 6 (altı) ay süreyle engellenmiştir.

TEŞEKKÜR

Tez çalışmam boyunca bana her türlü kolaylığı sağlayan ve gösterdiği yol ve yöntemlerle desteğini esirgemeyen sayın hocam Doç. Dr. Mesut KARABACAK'a ve maddi manevi desteklerini esirgemeyen annem Şengül OKUMUŞ ve babam Yusuf OKUMUŞ'a teşekkür ederim.

Yüksek Lisans eğitimim boyunca “Yurt İçi Yüksek Lisans Burs Programı” ile tarafıma vermiş olduğu destek için TÜBİTAK’a şükranlarımı sunmayı bir borç bilirim.

Hatice OKUMUŞ



ÖZET

YÜKSEK LİSANS TEZİ

YAPAY SİNİR AĞLARININ DALGACIK DÖNÜŞÜMLÜ AKTİVASYON FONKSİYONLARI İLE İYİLEŞTİRİLMESİ

Hatice OKUMUŞ

Danışman: Doç. Dr. Mesut KARABACAK

Amaç: Bu çalışmada YSA'nın çözümlerinde kullanılan standart aktivasyon fonksiyonlarını dalgacık dönüşümlü aktivasyon fonksiyonları ile değiştirerek sonuçlar üzerine bir karşılaştırma yapılması amaçlanmıştır.

Yöntem: Bu çalışmada iki farklı örnek problem ele alınarak YSA ile çözümlmeleri yapılmıştır. Bu örneklerin ilk kısmında gizli katmanda standart bilinen aktivasyon fonksiyonları kullanılarak çözümlenmiştir. İkinci kısmında ise gizli katmanda kullanılan aktivasyon fonksiyonlarından ilkinin dalgacık dönüşümlü aktivasyon fonksiyonları ile değiştirilerek yeniden çözümlleme yapılarak sonuçlar karşılaştırılmıştır. Dalgacık aktivasyon fonksiyonlarının avantajlı ve dezavantajlı durumları incelenmiştir.

Bulgular: Çözümlleme sonucunda elde edilen sonuçlar karşılaştırıldığında kullanılan aktivasyon fonksiyonlarının değişimi YSA üzerinde etkileri şöyledir. Dalgacık dönüşümlü aktivasyon fonksiyonları ile çözüm yapılması YSA'nın yerel özellikleri yakalama konusunda daha duyarlı olduğunu göstermiştir. Karmaşık modeller ve veri setlerinde aktivasyon fonksiyonu olarak dalgacık dönüşümlü aktivasyon fonksiyonlarının kullanılması daha avantajlıdır.

Sonuç: YSA modellerinde sinyal işleme, görüntü tanıma, hastalık veri analizi, kişisel öneri sistemleri, kişisel veri güvenlik sistemleri gibi karmaşık veri setleri üzerinde yapılacak çözümlmelerde dalgacık dönüşümlü aktivasyon fonksiyonlarının kullanılması daha hassas ve doğru sonuçlar elde etmemizi sağlar.

Anahtar Kelimeler: Yapay sinir ağları, Dalga dönüşümlü aktivasyon fonksiyonları, Aktivasyon fonksiyonları

Ocak 2025, 79 sayfa

ABSTRACT

MASTER THESIS

IMPROVEMENT OF ARTIFICIAL NEURAL NETWORKS WITH WAVELET TRANSFORM ACTIVATION FUNCTIONS

Hatice OKUMUŞ

Supervisor: Assoc. Prof. Dr. Mesut KARABACAK

Purpose: In this study, it is aimed to make a comparison on the results by replacing the standard activation functions used in ANN solutions with wavelet transformed activation functions.

Method: In this study, two different sample problems were taken and analyzed with ANN. In the first part of these examples, they were solved using standard known activation functions in the hidden layer. In the second part, the results were compared by re-analysis by replacing the first of the activation functions used in the hidden layer with wavelet transform activation functions. Advantages and disadvantages of wavelet activation functions were examined.

Findings: When the results obtained as a result of the analysis are compared, the effects of the changes in the activation functions used on the ANN are as follows. Solving with wavelet transformed activation functions showed that ANN is more sensitive in capturing local features. It is more advantageous to use wavelet transformed activation functions as activation functions in complex models and data sets.

Results: Using wavelet transform activation functions in analysis of complex data sets such as signal processing, image recognition, disease data analysis, personal recommendation systems, personal data security systems in ANN models allows us to obtain more sensitive and accurate results.

Keywords: Artificial neural networks, Wave transform activation functions, Activation Functions

January 2025, 79 pages

İÇİNDEKİLER

TEZ KABUL VE ONAY TUTANAĞI	i
ETİK BİLDİRİM VE İNTİHAL BEYAN FORMU	ii
TEŞEKKÜR	iii
ÖZET	iv
ABSTRACT	v
İÇİNDEKİLER.....	vi
TABLolar DİZİNİ.....	viii
ŞEKİLLER DİZİNİ	x
KISALTMALAR DİZİNİ	xi
GİRİŞ.....	1
KURAMSAL TEMELLER.....	4
Yapay Sinir Ağları	4
Yapay Sinir Ağının Kullanıldığı Alanlar	5
Biyolojik Sinir Ağının Yapısı	5
Yapay Sinir Ağının Çalışması.....	6
Yapay Sinir Ağının Eğitilmesi	10
Yapay Sinir Ağlarının Sınıflandırılması	10
Katman Sayısına Göre Yapay Sinir Ağları	11
1. Tek katmanlı yapay sinir ağı	11
2. Çok katmanlı yapay sinir ağı.....	11
Yapay Sinir Ağlarının Bağlantı Türlerine Göre Sınıflandırılması	12
İleri beslemeli yapay sinir ağları	12
Geri beslemeli yapay sinir ağı modeli.....	12
Dalgacık (Wavelet) Dönüşümleri	13
Dalgacık Dönüşümlü Sinir Ağları (Wavelet Neural Networks)	14
Dalgacık Dönüşümlü Aktivasyon Fonksiyonları	14
MATERYAL ve YÖNTEM	17
Yapay Sinir Ağlarının Çözüm Yöntemi.....	17
İleri Beslemeli Yapay Sinir Ağları ile Diferansiyel Denklemlerin Çözümleri	18
Dalgacık Dönüşümlü Aktivasyon Fonksiyonlarının Çözüm Metotları	24
Aktivasyon Fonksiyonu Tasarımı	24
Dalgacık Aktivasyon Fonksiyonlarının Matematiksel Yapısı	24
Genocchi Yapay Sinir Ağının Çözüm Metodu	27

ARAŖTIRMA BULGULARI	29
Test Problemi 1 (Diyabet Örneđi).....	29
Test Problemi 2 (KiŖisel Veri Güvenliđi Örneđi)	46
TARTIŖMA ve SONUÇ	63
KAYNAKLAR.....	65
ÖZGEÇMİŖ.....	66



TABLÖLAR DİZİNİ

Tablo 1. Aktivasyon Fonksiyonları 1	8
Tablo 2. Aktivasyon Fonksiyonları 2	9
Tablo 3. Diyabet Teşhisi İçin Kullanılan Özellikler	33
Tablo 4. Hasta Özellik Değer Tablosu.....	33
Tablo 5. Ağırlık ve Bias Değerleri	34
Tablo 6. Gizli Katman Net Girdi Değerleri	34
Tablo 7. İlk Aktivasyon Fonksiyonundan Elde Edilen Değerler	34
Tablo 8. Gizli Katmanın İkinci Ağırlık ve Bias Değerleri.....	35
Tablo 9. Gizli Katmanın İkinci Net Toplam Değerleri.....	35
Tablo 10. Bulunan İkinci Net Çıktı Değerlerinin Aktivasyon Fonksiyonu ile İşlem Sonuçları	35
Tablo 11. Çıkış Katmanı Ağırlık ve Bias Değerleri	36
Tablo 12. Diyabet Teşhisi İçin Kullanılan Özellikler	37
Tablo 13. Hasta Özellik Değer Tablosu.....	37
Tablo 14. Ağırlık ve Bias Değerleri.....	38
Tablo 15. Gizli Katman Net Girdi Değerleri	38
Tablo 16. İlk Aktivasyon Fonksiyonundan Elde Edilen Değerler	38
Tablo 17. Gizli Katmanın İkinci Ağırlık ve Bias Değerleri.....	39
Tablo 18. Gizli Katmanın İkinci Net Toplam Değerleri.....	39
Tablo 19. Bulunan İkinci Net Çıktı Değerlerinin Aktivasyon Fonksiyonu ile İşlem Sonuçları	39
Tablo 20. Çıkış Katmanı Ağırlık ve Bias Değerleri	40
Tablo 21. Özellik Değerleri	48
Tablo 22. Ağırlık Değerleri.....	49
Tablo 23. Bias Değerleri.....	49
Tablo 24. Birinci Net Girdi Değerleri.....	49
Tablo 25. Aktivasyon Fonksiyonunun Değerleri.....	50
Tablo 26. İkinci Ağırlık Değerleri	50
Tablo 27. İkinci Bias Değerleri.....	50
Tablo 28. İkinci Gizli Katman Hesaplama Değerleri	51
Tablo 29. İkinci Gizli Katman Aktivasyon Fonksiyonu Değerleri.....	51
Tablo 30. Çıkış Değerlerinin Ağırlıkları.....	52
Tablo 31. Çıkış Değerleri Bias Değerleri	52

Tablo 32. Özellik Değerleri	53
Tablo 33. Ağırlık Değerleri.....	53
Tablo 34. Bias Değerleri	54
Tablo 35. Birinci Gizli Katman Değerleri	55
Tablo 36. Aktivasyon Fonksiyon Değerleri.....	56
Tablo 37. İkinci Gizli Katman Ağırlıkları	56
Tablo 38. İkinci Gizli Katman Bias Değerleri	57
Tablo 39. İkinci Gizli Katman Sonuçları	57
Tablo 40. İkinci Gizli Katman Aktivasyon Fonksiyon Değerleri.....	58
Tablo 41. Çıkış Katmanı Ağırlık Değerleri	58
Tablo 42. Çıkış Katmanı Bias Değerleri.....	58
Tablo 43. Dalgacık Aktivasyon Fonksiyonu ve ReLU Aktivasyon Fonksiyonlarının Karşılaştırılması.....	61

ŞEKİLLER DİZİNİ

Şekil 1. Biyolojik sinir hücresi	6
Şekil 2. Sinir hücreleri arasındaki sinaptik bağlantı.....	6
Şekil 3. Yapay sinir ağının ilkel yapısı.....	7
Şekil 4. Yapay sinir ağının yapısı.....	7
Şekil 5. Yapay Sinir Ağının Katmanlarının Yapısı.....	10
Şekil 6. Tek katmanlı yapay sinir ağı modeli.....	11
Şekil 7. Çok katmanlı yapay sinir ağlarının modeli	12
Şekil 8. İleri beslemeli yapay sinir ağı modeli	12
Şekil 9. Geri beslemeli yapay sinir ağının modeli.....	13
Şekil 10. Yapay sinir ağı modeli	17
Şekil 11. İleri belemeli yapay sinir ağı modeli.....	18
Şekil 12. İleri beslemeli yapay sinir ağının topolojik yapısı	21
Şekil 13. Genocchi yapay sinir ağının yapısı	27
Şekil 14. Genocchi aktivasyon fonksiyonunun yapısı.....	28
Şekil 15. A- genocchi dalgacık aktivasyon fonksiyonlu ve B-sadece ReLU aktivasyon fonksiyonlu ROC eğrileri	42
Şekil 16. A- Genocchi Dalgacık aktivasyon fonksiyonlu eğitim doğrulama eğrisi B- ReLU aktivasyon fonksiyonlu eğitim doğrulama eğrisi.....	43
Şekil 17. A- Genocchi dalgacık aktivasyon fonksiyonlu doğrulama kaybı eğrisi B-ReLU aktivasyon fonksiyonlu doğrulama kaybı eğrisi.....	44
Şekil 18. A- Genocchi dalgacık aktivasyon fonksiyonlu modelin karışıklık matrisi B- ReLU aktivasyon fonksiyonlu modelin karışıklık matrisi	45
Şekil 19. Sahte hesap modelinde ReLU aktivasyon fonksiyonunun eğitim doğrulama grafiği	59
Şekil 20. Sahte hesap modelinde morlet dalgacığı aktivasyon fonksiyonunun eğitim doğrulama grafiği	60

KISALTMALAR DİZİNİ

FFANN	İleri beslemeli yapay sinir ağı
ReLU	Düzeltilmiş Doğrusal Birim (Rectified Linear Unit)
WNN	Dalgacık Dönüşümlü Aktivasyon Fonksiyonu
YSA	Yapay sinir ağı (Neural Network)



GİRİŞ

Yapay zekâ biliminin önemli araştırma alanlarından biri olan Yapay Sinir Ağları (YSA), bilgisayarların öğrenmesine ve karmaşık ilişki ağlarını keşfetmesine olanak sağlayan bir yaklaşım sunar. Günümüzde bilgisayarlar, geçmişte yalnızca basit hesaplamalar veya veri transferleri yapabilen araçlar olmaktan çıkmış; büyük miktarlardaki verileri analiz edebilen, sorunlara yönelik kararlar alabilen, hatta olaylar arasındaki çok karmaşık ilişkileri öğrenebilen sistemlere dönüşmüştür. Bu dönüşüm, yapay zekâ ve özellikle yapay sinir ağlarının farklı disiplinlerdeki hızlı gelişimiyle doğrudan ilişkilidir. YSA'lar, insan beyninin çalışma prensiplerinden esinlenerek oluşturulan bilgi işleme yapılarıdır. Bu yapılar, çok katmanlı ve uyarlanabilir ağırlıklara sahip olmaları sayesinde, farklı problem türlerini yüksek doğrulukla modelleyebilme potansiyeline sahiptir.

Sayısal analiz ve sinyal işleme alanlarında ise dalgacık teorisi ve dalgacık dönüşümleri (wavelet transforms) giderek önem kazanan güçlü araçlardır. Dalgacıklar, kompakt destekli ve farklı çözünürlük seviyelerinde yerel özellikleri yakalayabilen özel fonksiyonlardır. Fourier analizinden farklı olarak dalgacık dönüşümleri, zaman-frekans düzleminde hem yerel hem de küresel özellikleri aynı anda inceleyebilme imkânı sunar. Bu özelliği, dalgacık dönüşümlerini özellikle sinyal işleme, veri analizi, görüntü işleme, sistem tanımlama ve zaman serisi tahmini gibi alanlarda son derece kullanışlı hâle getirmektedir. Dalgacık teorisi, 1900'lü yılların başına dayanan temellerine rağmen 1970'lerin sonu ve 1980'lerin başında Jean Morlet, Alex Grossmann ve Yves Meyer gibi matematikçilerin öncü çalışmalarıyla ivme kazanmıştır. Daha sonra Mallat (1989) ve Daubechies (1992) gibi araştırmacıların katkılarıyla kuramsal altyapı sağlamlaştırılmış, hızlı ve verimli dalgacık tabanlı algoritmalara giden yol açılmıştır.

Yapay sinir ağları ile dalgacık teorisinin kesişim noktasında ise dalgacık sinir ağları (Wavelet Neural Networks, WNN) adı verilen yeni bir yaklaşım ortaya çıkmıştır. WNN'ler, yapay sinir ağlarının uyarlanabilirlik ve öğrenme yeteneklerini dalgacık dönüşümlerinin çok çözünürlüklü analiz kabiliyetiyle birleştirerek birçok alanda umut vadeden sonuçlar elde etmektedir. Zhang ve Benveniste (1992) tarafından erken dönemde yapılan çalışmalar, dalgacık tabanlı sinir ağlarının özellikle işaret tanıma ve zaman serisi öngörüsü gibi görevlerde yüksek performans sergileyebildiğini göstermiştir. Sonraki yıllarda dalgacık sinir ağları; sinyal işleme, görüntü tanıma, sistem tanımlama ve görüntü işleme gibi çeşitli alanlarda kullanılmaya

başlanmıştır. Bu kapsamda, sinyalin farklı frekans bileşenlerini öğrenebilme kabiliyeti sayesinde, gürültü azaltma ve detay yakalama noktasında da üstünlük sağlamaktadır.

Dalgacık teorisinden yararlanarak geliştirilmiş bu sinir ağı modelleri, veriyi farklı ölçeklerde inceleyerek hem makro düzeydeki (küresel) hem de mikro düzeydeki (yerel) özellikleri anlamlandırabilmektedir. Dolayısıyla, yüksek boyutlu ve karmaşık veri setlerinin analizinde dalgacık tabanlı filtreleme, özellik çıkarımı ve sınıflandırma süreçleri daha verimli hâle gelebilmektedir. Literatürde dalgacık sinir ağlarının kullanıldığı farklı çalışmalarda; örneğin, sağlık alanında elektromiyogram (EMG) sinyallerinin sınıflandırılmasından görüntü işleme alanında kenar tespiti ve sıkıştırmaya kadar pek çok uygulama rapor edilmiştir (örn. Zhao ve ark., 2017; Özbay ve ark., 2019).

Dalgacık teorisinin pratik uygulama sahası oldukça geniştir. Razzaghi ve Yousefi (2001) gibi araştırmacılar, dalgacıkların sayısal analiz, optimal kontrol, sistem analizi ve sinyal işleme başta olmak üzere pek çok disiplinde etkin bir şekilde kullanılabileceğini göstermiştir. Bu fonksiyonlar, farklı zaman ve frekans ölçeklerinde hızlı analiz yapılmasına imkân tanır. Uygulama alanlarındaki çeşitlilik ve yüksek performans, dalgacık teorisinin yapay sinir ağlarıyla bütünleştirilmesi fikrini daha da cazip hâle getirmiştir. Bu anlamda literatürde önemli başka bazı çalışmalar da aşağıda listelenmiştir.

- Pati ve Krishnaprasad (1993), dalgacık tabanlı bir yaklaşım kullanarak dinamik sistemlerin modellenmesi üzerine çalışmıştır.
- Deng ve Zhang (2013), derin öğrenme mimarilerinde dalgacık dönüşümlerinin kullanımı üzerine bir inceleme sunmuştur.
- Çelik (2015), dalgacık dönüşümlü yapay sinir ağlarının zaman serisi tahminindeki performansını değerlendirmiştir.

Bu tez çalışmasında, Yapay Sinir Ağları' nın performansını artırmak amacıyla dalgacık dönüşümlü aktivasyon fonksiyonlarının kullanımı ele alınacaktır. Literatürdeki çalışmalar, dalgacık tabanlı yaklaşımların sinir ağlarının genelleme kabiliyetine ve öğrenme hızına olumlu katkılar sunduğunu göstermektedir. Bu noktada amaç, standart sinir ağı yapılarının benimsediği klasik aktivasyon fonksiyonlarının yerine dalgacık tabanlı fonksiyonlar entegre ederek ağın karar verme yeteneğini geliştirmektir. Özellikle karmaşık veri setlerinde ve yüksek boyutlu problemlerde dalgacık dönüşümlü aktivasyon fonksiyonlarının sinyalin farklı özelliklerini aynı anda modelleyebilmesi, potansiyel olarak daha düşük hata oranlarına ve daha hızlı öğrenme süreçlerine olanak sağlayacaktır.

Görüldüğü üzere, yapay sinir ağılarına dalgacık dönüşümlerinin entegrasyonu giderek genişleyen bir araştırma sahası olup, bu alandaki çalışmalar hâlâ devam etmektedir. Dolayısıyla, bu tezde dalgacık dönüşümlü aktivasyon fonksiyonları ile zenginleştirilmiş yapay sinir ağıları çerçevesinde hem kuramsal arka plan hem de pratik uygulamalar ele alınacak, elde edilen sonuçlar niceliksel ve niteliksel analizlerle değerlendirilecektir.



KURAMSAL TEMELLER

İnsan beyni, üzerinde en çok araştırma yapılan fakat bu araştırmalardan çok az bilgi edinilen organdır. İnsanların en önemli özelliklerinden biri aklı kullanarak kendisine uygun davranışlar sağlamak ve karar verebilmek için model oluşturma çabası içerisinde. Bu durumları sağlayabilmek için öğrenmenin gerçekleşmesi gerekir. Öğrenme, hatırlama, düşünme, karar verme, dönüt verebilme gibi tüm insan davranışlarını beyinde bulunan sinir hücreleri yardımı ile gerçekleştirir. Son dönemde yapılan araştırmalar insan beyni gibi davranış gösteren makinalar üzerine olmuştur. Bu çalışmalar ışığında yapay zekanın alt dalı olan yapay sinir ağlarının önemi artmıştır.

Yapay Sinir Ağları

Yapay sinir ağları temelde tamamen insan beyninin örneklendirilmesi olan sinir hücrelerinden oluşmaktadır. Bu sinir hücreleri insan beyni gibi çalışan yani öğrenme sağlayan öğrendiği bilgiyi depolayan bu bilgilerle alakalı karar verip dönüşüm yapabilen bir sistemdir. Yapay sinir ağları günümüzde insan beyni gibi çalışmasından kaynaklı olarak toplumun yükünü hafifletmesi daha kolay sonuca ulaşılmasından dolayı talep edilen bir teknoloji haline almıştır. Bu teknoloji beyin temelli öğrenmenin daha yaygın kullanılmasına olanak sağlamış oluyor.

YSA insan beyninden yararlanarak oluşturulmuştur. Öğrenme süreçleri matematiksel modellere dayandırılarak inşa edilmiştir. Yapay Sinir Ağı ile insan beynine benzer bilgisayara öğrenebilme yeteneği kazandırılması amaçlanmıştır. Yapay Sinir Ağları öncelikle insan beynindeki nöronların modellenmesi ile başlamıştır. İnsan öğrenmesini dikkate alınırsa beynin elde ettiği tecrübeleri kullanarak öğrenme sağlanır. Öğrendiği bilgileri geliştirebilir, yeni bilgi oluşturabilir, bu bilgileri yorumlayıp sınıflandırabilir ve karar verme gibi kazanımları elde eder. Bu bilgiler yardımı bilgisayar için de bu kazanımları elde edebilecek çeşitli yaklaşımlar vardır. Bunlardan birisi de yapay sinir ağıdır.

Yapay sinir ağları, biyolojik sinir sistemlerinden esinlenildiği için insan sinir hücrelerine benzer davranışlar gösterir. Bu davranışlar ile karmaşık yapıda olan problemlerin çözümü hakkında bilgi sahibi olmak ve yorumlama üzerine daha kolay sonuçlar elde edilir.

İnsan sinir hücrelerinin çalışmasının basit bir modeli olarak hayatımıza giren yapay sinir ağları insan sinir hücrelerine göre çok ilkel kalsa da zamanın teknolojisi olarak adlandırılabilir.

Yapay Sinir Ağının Kullanıldığı Alanlar

Yapay sinir ağları gerçek hayat problemlerinin çözümü için kullanılan sistemdir. Yapay sinir ağları problemleri çözerken beklenen değer ile çıktı değeri arasındaki farkı minimal hale getirerek hatayı düşürerek problem hakkında bilgi verir. Yapay sinir ağlarının kullanıldığı alanları şöyle sıralayabiliriz.

Tahmin: Gelecekteki olaylar hakkında veya sonuçları hakkında bilgi verir. Örnek verilecek olursa finansal zaman serisi tahminlerinde kullanılabilir.

Sınıflandırma: Verileri kategorize etmek için kullanılabilir. Eğitim de öğrenci sınav sonuçlarını kategorilere ayırmak için kullanılabilir.

Dil İşleme: Genellikle web sitelerinde kullanılır. Girilen verileri doğru anlamlandırma ve cevaplamalarda kullanılır.

Optimizasyon: Yapay sinir ağlarının en çok kullanıldığı alandır. Karmaşık problemlerde kullanılır. Verilen problemleri optimize etmek için hatayı en aza indirmek için kullanılır.

Tanı: Hastalık teşhisinde kullanılır. Örneğin kişilerin diyabete olan yatkınlığının ölçülmesinde kullanılır.

Kontrol: Makine kontrollerinde kullanılır. Yapılan makinaların işlevselliğinin belirlenmesinde kullanılabilir.

Yapay sinir ağları gün geçtikçe önemi artan bir teknoloji olduğu için kullanım alanları zamanla artış gösterecektir.

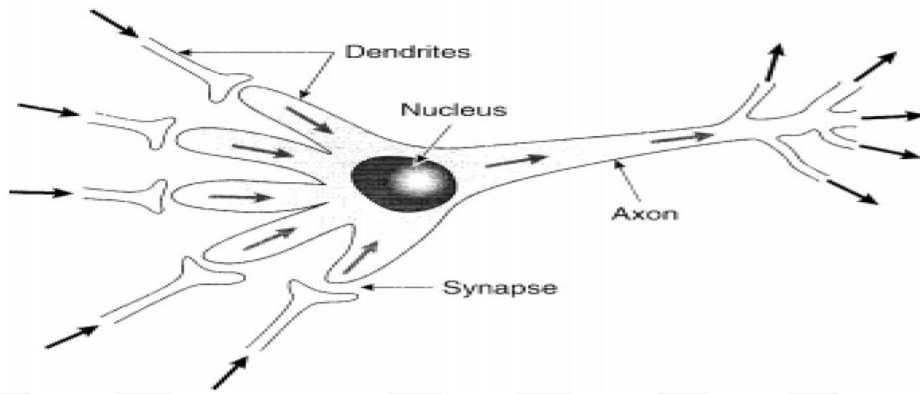
Biyolojik Sinir Ağının Yapısı

İnsan beyninin bilgiyi biyolojik olarak nasıl öğrendiğini inceleyerek Yapay Sinir Ağlarının çalışması tasarlanmıştır.

Biyolojik bir sinir hücresi; bir gövde, bir akson, çok sayıda sinir ucu (dendrit) ve akson ile diğer sinir hücresinin sinir ucu arasında kalan ince uzantılar (sinaps) olmak üzere dört bölümden oluşmaktadır. Dendritler, gelen sinyalleri çekirdeğe iletir. Çekirdek dendritten gelen sinyalleri bir araya toplar. Gelen uyarılar o sinir hücresinin eşik değerine ulaşırsa bir elektriksel çıktı oluşturulur ve aksona iletilir. Eşik değeri, çıktı oluşması için gereken minimum uyarı

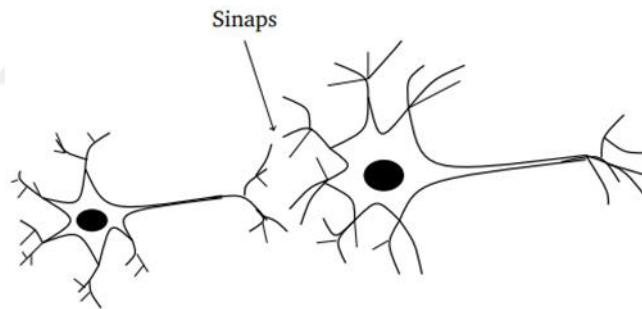
şiddetidir. Her bir nöronun kendine has bir eşik değeri vardır. Eşik değerine ulaşamayan uyarılara sinir hücresi yanıt vermez.

Eşik değerine ulaşan sinir hücreleri toplanan bu sinyalleri akson tarafından işleyerek sinapslara gönderir. Sinapslar da yeni üretilen sinyalleri diğer sinir hücrelerine iletir.



Şekil 1. Biyolojik sinir hücresi

Biyolojik sinir ağlarında aksonlar ile diğer sinir ağları arasında sinaptik boşluklar vardır. Bir önceki sinir ağında oluşan bilgiler bu boşluktan geçerek yeni sinir ağına aktarılır.



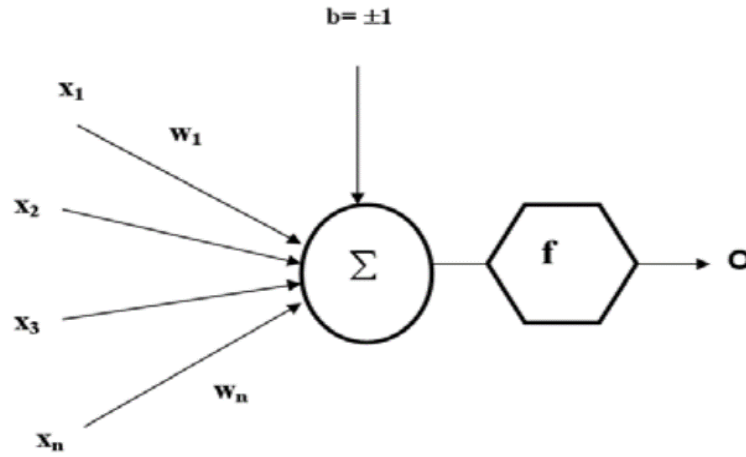
Şekil 2. Sinir hücreleri arasındaki sinaptik bağlantı

Yapay Sinir Ağı'nın Çalışması

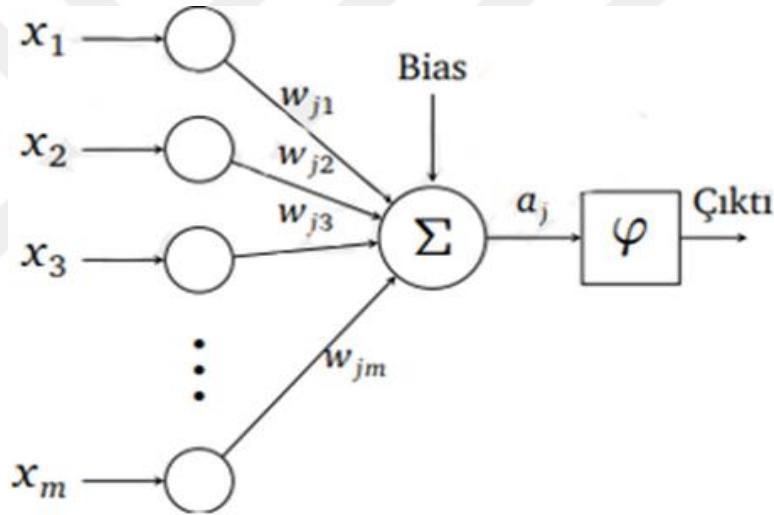
Biyolojik sinir ağlarının çalışmalarından esinlenerek yapay sinir ağları elde edilmiştir. Yani gerçek sinir hücrelerinin simule edilmiş hali Yapay Sinir Ağlarıdır. Yapay Sinir Ağları biyolojik sinir ağlarının simule edilmiş hali olduğu için çalışma yöntemleri benzerdir ve şu şekilde gelen bilgi/örnek iletilir:

Dış ortamdan veya diğer hücrelerden alınan girdiler, ağırlıklar yardımıyla hücreye bağlanır. Toplama fonksiyonu ile net girdi hesaplanır. Net girdinin aktivasyon fonksiyonundan geçirilmesiyle net çıktı hesaplanır. Bu işlem aynı zamanda bir hücrenin çıkışını verir. Her bağlantının bir ağırlık değeri vardır. Yapay Sinir Ağı, kendisine örnekler gösterildikçe bu

ağırlık değerlerini değiştirir. Hedef, ağı gösterilen örnekler için doğru çıktıları verecek ağırlıkları bulmaktır. Bu yöntem ile makine öğrenmeleri sağlanmış olur.



Şekil 3. Yapay sinir ağının ilkel yapısı



Şekil 4. Yapay sinir ağının yapısı

Yapay sinir ağlarının çalışma prensibini katmanlarını göstermeden önce yapay sinir ağının bölümlerinden bahsedip yapay sinir ağı üzerindeki görevlerinin açıklamasını yapalım.

Ağırlıklar: Yapay sinir ağına gelen bilgilerin üzerindeki etkiyi göstermek için kullanılan değerlerdir. Yapay sinir ağına gelen bilgilerin önemini gösterir. Ağı eğitilmesinde kullanılan ağırlıklar güncellenerek hatanın minimum hale getirilmesinde kullanılır. Ağırlıklar değiştikçe çıktı değerleri de değişir. Ağırlık değerlerimiz sabit veya değişken değerler olabilir. Bu değerlerin pozitif olması etkiyi artırırken negatif olası etkiyi azaltır. Şekil 4 w_{jm} değerleri yapay sinir ağının ağırlıklarıdır.

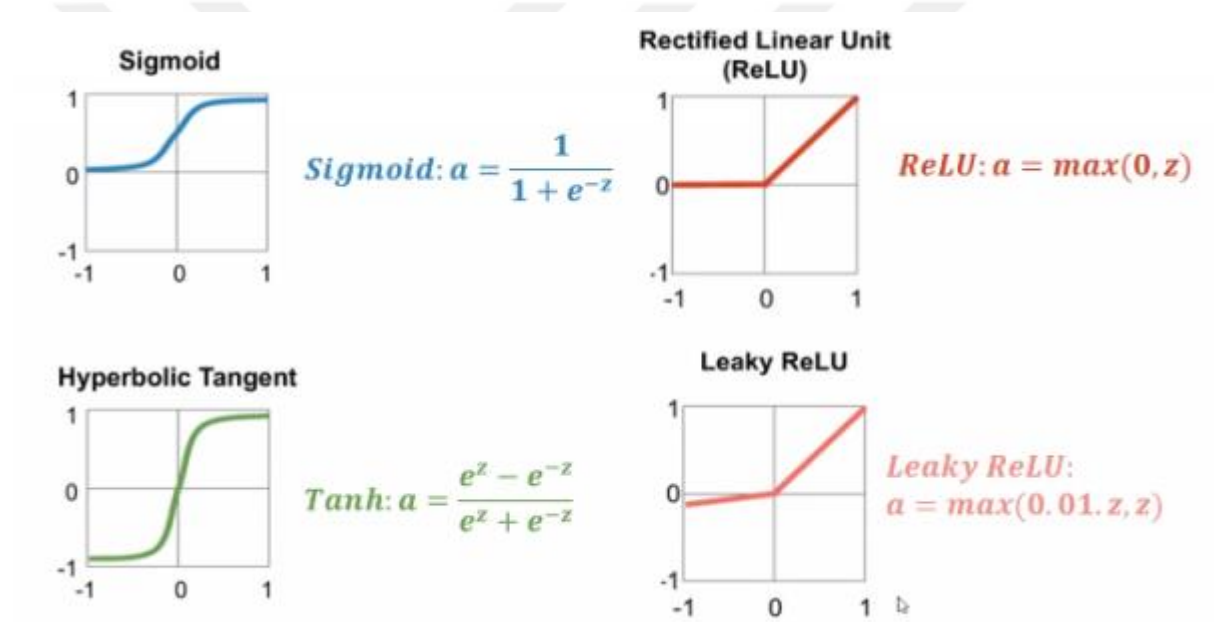
Toplama Fonksiyonu: Yapay sinir ağına gelen girdilerin ağırlığı ile çarpılarak toplanması ile elde edilen net girdilerdir. Bu değer hesaplanırken her bir girdi değeri için verilen ağırlık değeri çarpılır ve bu sonuçlar toplanarak toplama fonksiyonu elde edilir. Bu ifade;

$$a_j = \sum_{i=1}^m x_i w_{ji} \quad (1)$$

eşitliği ile ifade edilir.

Aktivasyon Fonksiyonu: Yapay sinir ağının oluşturduğu net toplam değerinden üreteceği çıktı değerini belirleyen fonksiyondur. Literatürde en çok kullanılan aktivasyon fonksiyonları aşağıdaki tablolarda gösterilmiştir.

Tablo 1. Aktivasyon Fonksiyonları 1



Tablo 2. Aktivasyon Fonksiyonları 2

Aktivasyon Fonksiyonu	Türev	Açıklama
Lineer Fonksiyon $\varphi(x) = x$	$\varphi'(x) = 1$	Yapay nöronun çıktısı net girdiye eşit olur.
Eşik Fonksiyonu $\varphi(x) = \begin{cases} 1, & \text{if } x \geq 0 \\ 0, & \text{if } x < 0 \end{cases}$	$x=0$ noktasında türevi yoktur.	Net girdinin büyüklüğüne göre yapay nöron 0 veya 1 değerini alır. Sınıflandırma problemlerinde kullanılır.
Sigmoid Fonksiyonu $\varphi(x) = \frac{1}{1+\exp(-x)}$	$\varphi'(x) = \varphi(x)[1 - \varphi(x)]$	Çok sık kullanılan bir aktivasyon fonksiyonudur. $[0, 1]$ aralığında değerler alır. Sürekli ve lineer olmayan bir fonksiyondur.
Hiperbolik tanjant Fonksiyonu $\varphi(x) = \tanh(x) = \frac{\exp(x) - \exp(-x)}{\exp(x) + \exp(-x)}$	$\varphi'(x) = 1 - \varphi^2(x)$	$[-1, 1]$ aralığında değer alır.
RELU $\varphi(x) = \max(0, x)$	$\varphi'(x) = \begin{cases} 1, & \text{if } x \geq 0 \\ 0, & \text{if } x < 0 \end{cases}$	$[0, \infty)$ aralığında değerler alır. Hesaplama kolaylığı nedeniyle sık kullanılan bir aktivasyon fonksiyonudur.
Gauss Radyal Temel Fonksiyonu $\varphi(x) = \exp(-\frac{\ x-c\ ^2}{2\delta^2})$	$\varphi'(x) = \frac{-2(x-c)\varphi(x)}{\delta^2}$	Radyal tabanlı fonksiyon yapay sinir ağları (RBFNN) için sık kullanılan temel fonksiyonlardan biridir. c ve δ parametreleri probleme göre belirlenmektedir.

Aktivasyon fonksiyonlarının değeri aşağıdaki denklem ile hesaplanır.

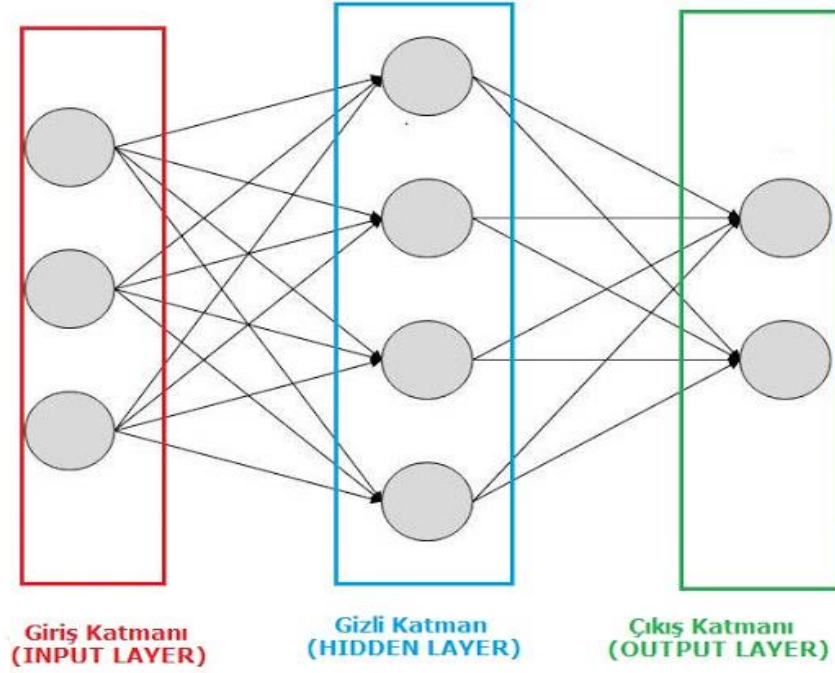
$$y_i = \varphi(a_i) \quad (2)$$

Bias Değeri: Net girdi hesaplanırken sonuca eklenen sabit değerdir. Ekleme zorunluluğu yoktur. Gerekli olan durumlarda eklenir. Bias değeri aktivasyon fonksiyonunun sonucunu artırıp azaltabilir. Beklenen sonuçtan sapma olduğu zaman yararlılıkları vardır. Bias değeri hata oranını azaltmak için yardımcı olan bir değerdir.

$$a_j = \sum_{i=1}^m x_i w_{ji} + b_j \quad (3)$$

Bu verilen toplama fonksiyonu sonucuna eklenen b_j bias değeridir.

Yapay sinir ağlarının genel yapısı bu kısımlardan oluşur. Yapay sinir ağları katmanlardan oluşur ve bu katmanlar birbirine bağlıdır. Bu katmanlar; giriş katmanı, ara katman veya gizli katman, çıkış katmanı olarak üç ana katmandan oluşur. Şekil 5 te yapay sinir ağının katmanları gösterilmiştir.



Şekil 5. Yapay Sinir Ağının Katmanlarının Yapısı

Giriş Katmanı: Dışardan gelen bilgilerin bulunduğu kısım girdi katmanıdır. Girdi katmanında herhangi bir işlem yapılmaz. Tamamen çevrede bulunan bilgilerden oluşur.

Gizli Katman/Ara Katman: Girdi katmanı ile çıktı katmanı arasındaki katmana denir. Ara katmanlar birden fazla olabilir. Bu katman işlem yapılan katmandır. Gelen bilgiler toplama fonksiyonundan geçerek bir aktivasyon fonksiyonu ile işleme tabi tutularak net girdi değeri hesaplanır. Bu hesaplamaları çıktı katmanına gönderir.

Çıktı Katmanı: Ara katmandan gelen bilgileri işleyerek girdi setine gelen bilgilerin çıktısını oluşturur. Bu bilgiler yapay sinir ağının sonucu olabilirken eğer hata oranı fazla ise bu duruma dayanarak yeni girdiler oluşturabilir.

Yapay Sinir Ağının Eğitilmesi

Yapay sinir ağları eğitilirken dışardan alınan girdiler ağırlıklandırılır. Toplam değer hesaplanarak aktivasyon fonksiyonuyla işleme tabi tutulur. Bu işlem ile çıktılar oluşur. Oluşan çıktılar ile hata hesaplanır, optimizasyon yapılır. Eğer girdi ve çıktı değerleri uyum sağlıyorsa amaca ulaşmış ise ağırlıklar saklanır. Bu süreçte geçen zaman öğrenme olarak adlandırılır. Tüm sürece ise öğrenme algoritmaları denmektedir.

Yapay Sinir Ağlarının Sınıflandırılması

Yapay sinir ağlarında öğrenme algoritmalarına göre 3 grupta toplanır.

Denetimli Öğrenme/ Öğretmenli Öğrenme: Sisteme öğrenilmesi istenen girdi ve çıktı değerleri eğitici tarafından yüklenerek oluşturulur. Ağırlıkları güncelleyerek eğitici hatayı düşürmeye çalışır. Bu öğrenme çeşidinde sonuçlar bilinir amaç sonuçlara en az hata ile ulaşmaktır.

Denetimsiz Öğrenme/ Öğretmensiz Öğrenme: Bu durumda çıktı değerleri bilinmemektedir. Cevabın doğruluğu veya yanlışlığı girdilerin verdiği çıktılarına göre belirlenir. Bu durum genelde sınıflandırma yöntemlerinde kullanılır. Örnek verilirse sınıftaki öğrencilerin aldığı puanları sınıflandırmak istenirse kullanılır.

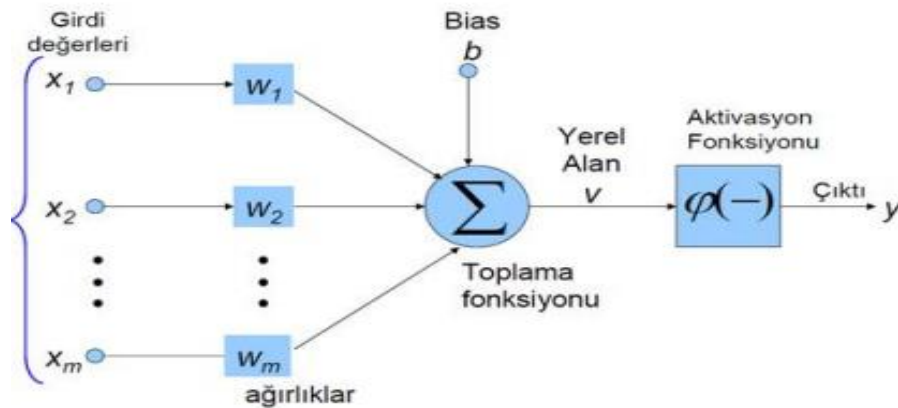
Destekleyici Öğrenme: Ağın davranışının doğru olup olmadığını belirten bir bilgi mevcuttur. Bu bilgiye göre ağırlık ayarlanır. Bu durumda yapay sinir ağı deneme yanılma ile eğitilir.

Katman Sayısına Göre Yapay Sinir Ağları

Yapay sinir ağlarının ara katman sayısına göre iki gruba ayrılır.

1. Tek katmanlı yapay sinir ağı

Tek katmanlı yapay sinir ağlarında giriş ve çıkış katmaları vardır. Bu tarz yapay sinir ağları en ilkel yapay sinir ağıdır. Doğrusal problemlerin çözümlerinde kullanılır. Çıktı fonksiyonu doğrusaldır. -1 veya +1 değerlerini alır. Şekil 6 da tek katmanlı yapay sinir ağının modeli gösterilmiştir.

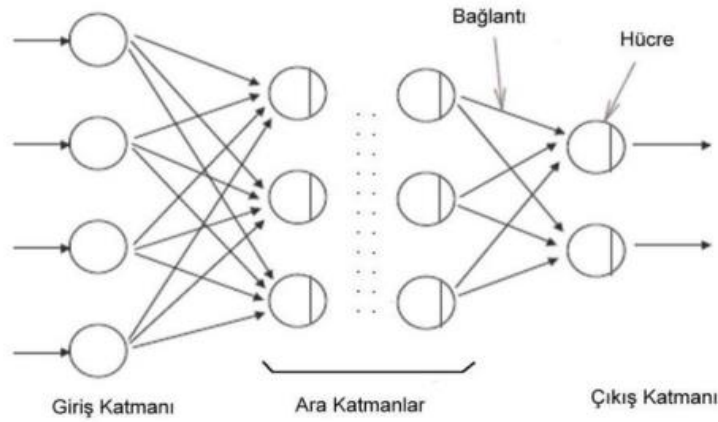


Şekil 6. Tek katmanlı yapay sinir ağı modeli

2. Çok katmanlı yapay sinir ağı

Çok katmanlı yapay sinir ağları giriş katmanı, ara katmanlar ve çıkış katmanından oluşmaktadır. En yaygın kullanılan modeldir. Doğrusal olmayan ve daha karmaşık olan ilişkileri öğrenmek için kullanılan modeldir. Tek katmanlı yapay sinir ağları doğrusal olan

problemleri çözebildiği için doğrusal olmayan daha karmaşık problem çözümlerinde yetersiz kalıyor. Şekil 7 de çok katmanlı yapay sinir ağlarının modeli gösterilmiştir.

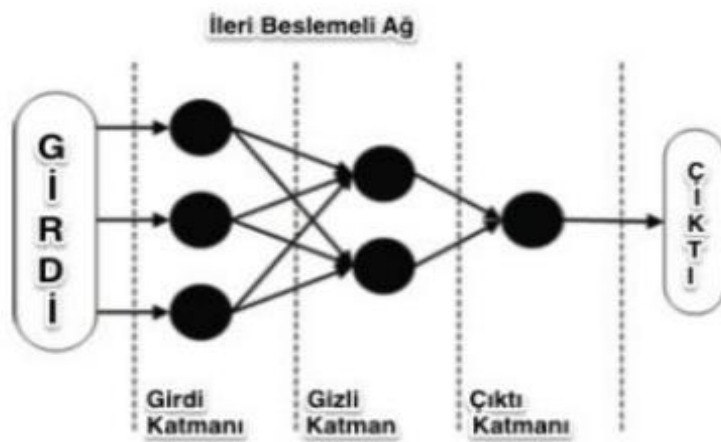


Şekil 7. Çok katmanlı yapay sinir ağlarının modeli

Yapay Sinir Ağlarının Bağlantı Türlerine Göre Sınıflandırılması

İleri beslemeli yapay sinir ağları

İleri beslemeli yapay sinir ağlarında her nöron bitişiğinde bulunan nöronlardan etkilenir. Aynı katman için de nöronların bağlantısı yoktur. Bu ise verilen sinir ağının daha hızlı çalışmasına olanak sağlar. Bu sinir ağı giriş katmanından başlayarak sıra ile ara katmana ve son olarak çıkış katmanına ilerler. Geri dönüş yoktur. Yani sadece bir önceki katmandan yararlanılabilir

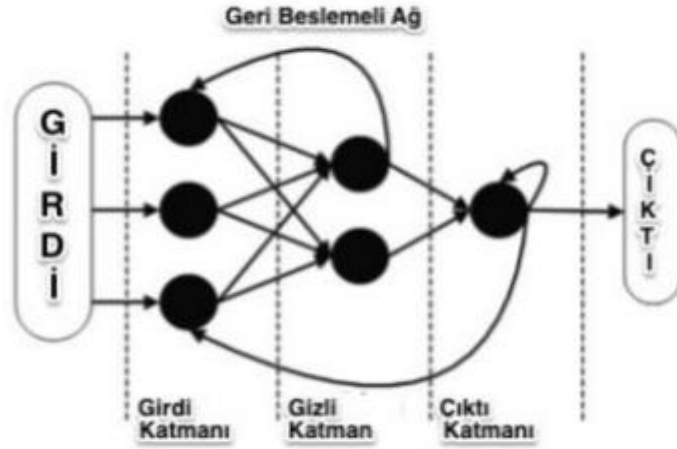


Şekil 8. İleri beslemeli yapay sinir ağı modeli

Geri beslemeli yapay sinir ağı modeli

Geri beslemeli yapay sinir ağı modeli ileri beslemeli gibi sadece bir önceki hücreden değil kendinden önceki diğer hücrelerden de beslenebilir. Aynı ağ içerisinde çıkış ve ara

katmanlardaki giriş katmanına veya önceki ara katmanlara geri beslendiği bir ağ yapısına sahiptir. Bu sebeple yapay sinir ağı doğrusal olmayan dinamik davranışlar gösterir.



Şekil 9. Geri beslemeli yapay sinir ağının modeli

Dalgacık (Wavelet) Dönüşümleri

Sayısal analiz alanında, dalgacık teorisi ve dalgacık dönüşümleri sinyal işleme, veri analizi ve hesaplamalı matematik için güçlü araçlar olarak ortaya çıkmıştır. Küçük dalga benzeri fonksiyonlar olarak tanımlanan dalgacıklar, sinyalleri farklı ölçeklere ayırtmak için benzersiz bir yaklaşım sağlar ve hem yerel hem de küresel özellikleri dikkate değer bir verimlilik ve hassasiyetle etkili bir şekilde yakalar. Dalgacık teorisinden türetilen dalgacık dönüşümleri, sinyalleri çoklu çözünürlüklerde analiz etmek için sağlam bir matematiksel çerçeve sunarak çeşitli alanlarda geniş bir uygulama yelpazesi sağlar.

Dalgacık teorisi 1900 lü yılların başlangıcına dayansa da pratik uygulamalar 1970'lerin sonu ve 1980'lerin başında Jean Morlet, Alex Grossmann ve Yves Meyer gibi matematikçilerin öncü çalışmalarına dayandırılabilir. Fourier analizi ve sinyal işlemeden esinlenen dalgacık teorisi, sinyal gösterimi ve analizi için farklı ölçeklerdeki sinyal özelliklerini yakalamaya yönelik uyarlanabilir yeteneği ile karakterize edilen yeni bir paradigma ortaya koymaktadır.

Bu genel bakış hem teorik temelleri hem de pratik uygulamaları kapsayacak şekilde, dalgacıkların ve dalgacık dönüşümlerinin sayısal analizde kullanımına ilişkin kapsamlı bir araştırma yapmayı amaçlamaktadır.

Teorik bilgileri pratik uygulamalarla sentezleyen bu genel bakış, dalgacıkların ve dalgacık dönüşümlerinin sayısal analizdeki rolünün kapsamlı bir şekilde anlaşılmasını sağlamayı amaçlamaktadır. Deneysel çalışmalar, örnek problemler ve karşılaştırmalı analizler aracılığıyla bu çalışma, karmaşık sayısal analiz problemlerinin ele alınmasında dalgacık tabanlı

tekniklerin etkinliğini ve çok yönlülüğünü göstermeyi ve bu dinamik alanda daha fazla araştırma ve yeniliğe ilham vermeyi amaçlamaktadır.

Dalgacık Dönüşümlü Sinir Ağları (Wavelet Neural Networks)

Son yıllarda, dalgacık teorisi ve yapay sinir ağlarının kesişimi, dalgacık sinir ağları (WNN'ler) olarak bilinen yeni ve umut verici bir paradigmanın ortaya çıkmasına neden olmuştur. Sinir ağlarının uyarlanabilirlik ve öğrenme yeteneklerini dalgacık dönüşümlerinin sunduğu çok çözünürlüklü analiz ile birleştiren WNN'ler, sinyal işleme, zaman serisi tahmini, örüntü tanıma ve sistem tanımlama gibi çeşitli alanlardaki karmaşık problemleri ele almak için güçlü bir çerçeve sunmaktadır.

WNN kavramı hem dalgacık teorisi hem de sinir ağı teknolojisi tarafından atılan temel üzerine inşa edilmiştir. Jean Morlet ve Yves Meyer gibi matematikçilerin öncülük ettiği dalgacık dönüşümleri, sinyalleri birden fazla çözünürlükte analiz etmek için esnek ve verimli bir araç sağlayarak hem ince hem de kaba ayrıntıları aynı anda yakalar. Bu arada, biyolojik beynin yapısı ve işlevinden esinlenen yapay sinir ağları, yinelemeli eğitim süreçleri yoluyla verilerden karmaşık örüntüleri ve ilişkileri öğrenmede mükemmeldir.

Dalgacık Dönüşümlü Aktivasyon Fonksiyonları

Dalgacık dönüşümlü aktivasyon fonksiyonları, genellikle YSA larda kullanılan bir aktivasyon fonksiyonu türlerindendir. Bu fonksiyonlar, modelin öğrenme yeteneğini artırmak ve daha karmaşık ilişkileri yakalamak için kullanılır. Bu aktivasyon fonksiyonları temelde şu özelliklere sahip olmalıdır.

Dalgacık dönüşümlü aktivasyon fonksiyonları, giriş verilerini belirli bir düzende dönüştürüp değiştirerek çıkış verir. Bu dönüşüm, standart bilinen aktivasyon fonksiyonları ve dalgacık (wavelet) dönüşümlü aktivasyon fonksiyonlarının kullanılmasıyla gerçekleştirilir.

Dalgacık dönüşümlü YSA ların özelliklerine değinilecek olursa Non-lineerlik durumlarında dalgacık dönüşümlü aktivasyon fonksiyonları, verilen modelin non-lineer (linier olmayan) ilişkileri öğrenmesine yardımcı olur.

Farklı türde dalgacık fonksiyonları (örneğin, Haar, Daubechies) kullanılarak farklı öğrenme süreçleri oluşturur ve bu sonuçlar ışığında verilen modelin eğitiminde hangi tür dalgacığın kullanılacağına karar verilir. Bu dalgacıklar ile modelde bulunan verilerin farklı ölçeklerdeki yapısını analiz etme ve modelleme yeteneğini göz önüne sermiş olur.

Dalgacık dönüşümlü YSA ların kullanım alanları şöyledir:

Görüntü İşleme: Görüntü verilerinin daha iyi temsil edilmesi için.

Zaman Serisi Analizi: Zaman serisi verilerinin daha iyi modellenmesi için.

Sinyal İşleme: Karmaşık sinyallerin analizi ve sınıflandırılması için.

Dalgacık dönüşümlü YSA larda kullanılan bazı dalgacık türlerini sıralayalım:

1. Haar Dalgacıkları

En basit dalgacık türüdür. İki seviyeli bir yapıya sahiptir ve verileri keskin değişim noktalarıyla temsil eder. Görüntü sıkıştırma ve kenar tespiti uygulamaları basit yapısıyla hızlı hesaplamalar sağlar, bu da gerçek zamanlı uygulamalar için kullanılması avantajlı bir dalgacık türüdür.

Haar dalgacığı aşağıdaki gibi tanımlanır:

$$\varphi(t) = \begin{cases} 1 & 0 \leq t < \frac{1}{2} \\ -1 & \frac{1}{2} \leq t < 1 \\ 0 & \text{diğer durumlar} \end{cases} \quad (4)$$

şeklindedir.

2. Daubechies Dalgacıkları

Daha az sayıda sıfıra sahip dalgacıklardır. Farklı boyutlarda (Daubechies 2, 4, 6 vb.) versiyonları vardır. Sinyal işleme ve görüntü işleme, gürültü giderme gibi modellerde farklı versiyonlarından dolayı kullanışlı bir dalgacık türüdür.

Bu dalgacık için genel bir formül yoktur. Çünkü farklı sıfır sayısına sahip bir dizi vardır. Örnek verilecek olursa Daubechies 4 dalgacığı için;

$$\vartheta(t) = \frac{1}{4}(\sqrt{3} + 2\sqrt{2}t + 3t^2 - 2\sqrt{2}t^3 - \sqrt{3}t^4) \quad (0 \leq t < 1) \quad (5)$$

şeklindedir.

3. Symlet Dalgacıkları

Daubechies dalgacılarına benzer, simetrik yapıya sahiptir. Dalgacıkların daha simetrik bir biçimde yeniden yapılandırılmasına olanak tanır. Görüntü analizi ve sinyal işleme alanlarında kullanılır. Özellikle zaman-frekans analizi gerektiren uygulamalarda faydalıdır. Belirli bir genel formülü yoktur. Symlet 4 için:

$$\sigma(t) = (1 - 2t^2 + t^4) \quad (0 \leq t < 1) \quad (6)$$

4. Coiflet Dalgacıkları

Hem simetrik hem de daha fazla sifıra sahip olan bu dalgacıklar, daha iyi bir zaman-frekans analizi sağlar. Gürültü giderme ve veri analizi alanlarında kullanılır. Özellikle daha karmaşık sinyal özelliklerini temsil etme kabiliyeti ile öne çıkan bir dalgacık türüdür. Coiflet 1 için:

$$\partial(t) = (1 - 2t + 2t^2) \quad (0 \leq t < 1) \quad (7)$$

5. Morlet Dalgacıkları

Bir sinüs dalgası ile bir Gauss dalgasının çarpımıdır. Sürekli dalgacık dönüşümünde yaygın olarak kullanılır. Zaman-frekans analizi ve sinyal işleme alanlarında kullanılır. Bu durum daha geniş alanda çalışma fırsatı tanır. Morlet dalgacığı aşağıdaki gibi gösterilir:

$$\partial(t) = \frac{1}{\sqrt{2\pi}} e^{-\frac{t^2}{2}} \cos(5t) \quad (8)$$

6. Genocchi Dalgacıkları

Non-lineer modellerde Genocchi dalgacığı, non-lineer dönüşümler sağlayarak ağına daha karmaşık ilişkileri öğrenmesine yardımcı olabilir. Dalgacıkların frekans içerikleri sayesinde, veri analizi ve özellik çıkarımı süreçlerinde faydalı olabilir. Görüntü işleme, sinyal analizi ve zaman serisi tahminlerinde kullanılabilir.

Genocchi dalgacıkları bir aralık üzerinde birçok faydalı özelliğe sahiptir. Bu özellikleri Genocchi polinomlarının özelliklerinden çıkarabiliriz.

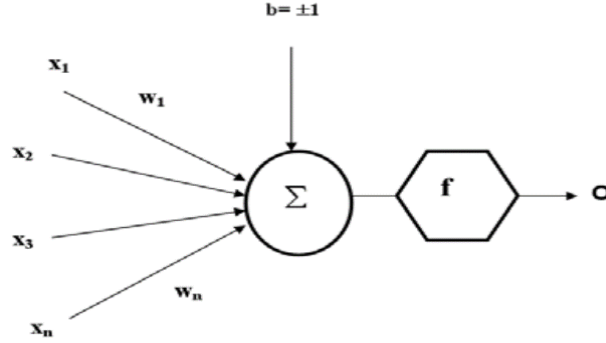
Genocchi polinomlarının bireysel terimleri birer tam sayıdır. Bu sebepten dolayı hesaplama hatası yoktur. Diğer dalgacıkların polinomlarının katsayıları tam sayı olmaması Genocchi polinomlarını daha üstün hale getiriyor. Genocchi polinomlarının terimleri diğer polinomların terimlerinden daha az sayıdadır. Bu sebeple, rastgele bir fonksiyona yaklaşırken diğer fonksiyonlara göre daha kolay sonuç elde etmemize yardımcı olur. Genocchi dalgacıkları aşağıdaki gibi ifade edilir:

$$\rho(t) = \frac{1}{\pi} \sin(\pi t) \quad 0 \leq t < 1 \quad (9)$$

Bu dalgacıkları verilen modellerimizde aktivasyon fonksiyonu olarak kullanmak YSA daha iyileştirebilir. YSA deneme yanılma ile belirlenen modeller olduğu için dalgacıkların kullanımı için en uygun dalgacığı kullanmak önemlidir. Bu nedenle YSA da kullanacağımız dalgacık dönüşümlü aktivasyon fonksiyonlarını deneme yoluyla en iyi sonuç veren dalgacık kullanılmalıdır.

MATERYAL ve YÖNTEM

Yapay Sinir Ağlarının Çözüm Yöntemi



Şekil 10. Yapay sinir ağı modeli

Biyolojik sinir ağları beynimizde bulunan birçok sinir hücresinin koleksiyonudur. Yapay Sinir Ağı ise biyolojik sinir ağlarının basit bir yapısıdır. Şekil 10 da görüldüğü gibi girdiler ağırlıklar toplama fonksiyonu ve çıktılarından oluşmaktadır.

Şekil 10 da bulunan x_i ler yapay sinir ağı için kullanılacak girdileri ifade eder w_i ler ise yapay sinir ağlarında katmanlar arasındaki ağırlıkları ifade eder. Σ ve b girdi ve ağırlıkların toplama fonksiyonlarının yapay sinir ağları çözümünde eşik değeri ile işleme tabi olduğunu gösterir. Yapay Sinir Ağlarının çıktısını elde edebilmek için kullanılan aktivasyon fonksiyonu f ile ifade edilir.

Bir yapay sinir ağı dışarıdan gelen bilgileri girdi olarak kabul eder, ağırlıklar ise bilginin önemini ağ üzerindeki etkisi ile gösterir. Ağa gelen net bilgi toplama fonksiyonu ile hesaplanır. Genelde girdi değeri kendi ağırlığıyla çarpılarak bütün girdilerin toplamı hesaplanarak net girdi sayısı belirlenir.

$$net = \sum_{i=1}^n x_i w_i \quad (10)$$

$$x_i \rightarrow \text{Girdi}$$

$$w_i \rightarrow \text{Ağırlık}$$

Çıktıyı hesaplamak için toplama fonksiyonunda olduğu gibi çeşitli aktivasyon fonksiyonları kullanılır. Genelde kullanılan aktivasyon fonksiyonu sigmoid fonksiyonudur ve şöyle ifade edilir.

$$f(net) = \frac{1}{1 + e^{-net}} \quad (11)$$

Burada bulduğumuz net işlem elemanına gelen toplam fonksiyonunu kullanarak belirlenen girdi değerlerini gösterir.

Bu hesaplamalar yapıldıktan sonra geriye doğru hesaplama aşamasına geçilerek hata payı hesaplanır. Beklenen değerlerimiz ($B_1, B_2 \dots$) ve çıktı değerlerimiz ($C_1, C_2 \dots$) olsun.

Beklenen değerlerimiz ile çıktı değerlerimiz arasındaki fark bize hata değerlerini verir.

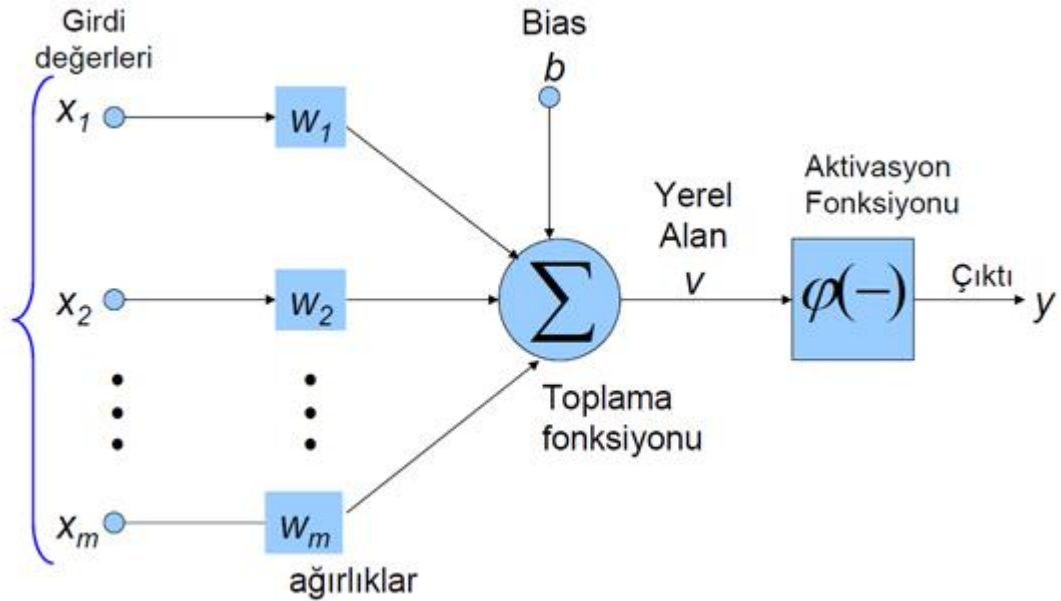
Hata değerlerimizi ($H_1, H_2 \dots$) ile gösterelim.

$$H_i = B_i - C_i \quad (12)$$

ile i . değerin hata değeri hesaplanır. Toplam Hata değerlerimiz aşağıdaki eşitlik ile hesaplanmaktadır.

$$TH = \frac{1}{2} \sum_m H_m^2 \quad (13)$$

İleri Beslemeli Yapay Sinir Ağları ile Diferansiyel Denklemlerin Çözümleri



Şekil 11. İleri belemeli yapay sinir ağı modeli

İleri beslemeli yapay sinir ağlarının çözümü yapılırken girdi katmanında bulunan değerler ile ağırlıklar çarpılır ve toplam değeri bulunur. Bu değer aktivasyon fonksiyonundan işleme tabi tutularak çıktı hesaplanır. Beklenen değerden bulunan değerin farkı alınarak hata değeri hesaplanır. Hata değerini geri yayılım ile ağırlıkları tekrar güncelleyerek işlem tekrar edilir. Bu yöntem ile hata değerinin en küçük hali hesaplanır.

$$\sum = (x_1 \cdot w_1) + \dots + (x_n \cdot w_n) \quad (1)$$

Bulunan toplam değerlerine yanlılık değeri β eklenir. Bu değere α diyelim.

$$\alpha = \sum_{i=1}^n (x_i \cdot w_i + \beta)$$

Aktivasyon fonksiyonları nöronların doğrusal olmayan fonksiyonlar olduğunu belirtir. Eğer aktivasyon fonksiyonu kullanılmazsa sinir ağı sadece doğrusal olacaktır. Genellikle kullanılan sigmoid fonksiyonunu kullanarak adımları ilerleteceğiz.

$$\varphi^{\sim} = \sigma(\alpha) = \frac{1}{1 + e^{-\alpha}} \quad (14)$$

İstenilen çözümden ne kadar uzak olduğunu bulmak için hata değeri hesaplanır. Her çıktı değeri için hata değeri hesaplanarak toplam hata değeri bulunur.

$$H_i = (\varphi_i - \varphi_i^{\sim}) \quad (15)$$

$$H = \frac{1}{n} \sum_{i=1}^n (\varphi_i - \varphi_i^{\sim})^2 \quad (16)$$

Bulunan toplam hata değeri ile en iyi ağırlıkları hesaplamak için yeni ağırlık değerlerini hesaplamamız lazım. Hata değerleri ve gradyanlar yardımıyla yeni ağırlıklar hesaplanır. Toplam hata fonksiyonunun gradyanını, kısmi türev kullanarak ağırlık w_i 'ye göre hesaplayalım. Hata fonksiyonu sadece ağırlık ile ilgili olmadığından zincir kuralı kullanılır.

$$\frac{\partial H}{\partial w_i} = \frac{\partial H}{\partial \varphi^{\sim}} \cdot \frac{\partial \varphi^{\sim}}{\partial \alpha} \cdot \frac{\partial \alpha}{\partial w_i} \quad (17)$$

$$\frac{\partial H}{\partial \varphi^{\sim}} = \frac{\partial}{\partial \varphi^{\sim}} \frac{1}{n} \sum_{i=1}^n (\varphi_i - \varphi_i^{\sim})^2 = 2 \cdot \frac{1}{n} \sum_{i=1}^n (\varphi_i - \varphi_i^{\sim})$$

$$\frac{\partial \varphi^{\sim}}{\partial \alpha} = \frac{\partial}{\partial \alpha} (\varphi^{\sim}) = \frac{\partial}{\partial \alpha} \sigma(\alpha) = \frac{\partial}{\partial \alpha} \left(\frac{1}{1 + e^{-\alpha}} \right)$$

$$= \frac{e^{-\alpha}}{(1 + e^{-\alpha})^2} = \left(\frac{1}{1 + e^{-\alpha}} \right) \cdot \left(\frac{e^{-\alpha}}{1 + e^{-\alpha}} \right)$$

$$= \left(\frac{1}{1 + e^{-\alpha}} \right) \cdot \left(1 - \frac{1}{1 + e^{-\alpha}} \right)$$

$$= \varphi^{\sim} \cdot (1 - \varphi^{\sim})$$

$$\frac{\partial \alpha}{\partial w_i} = \frac{\partial}{\partial w_i} \alpha = \frac{\partial}{\partial w_i} \sum_{i=1}^n (x_i \cdot w_i + \beta) = x_i$$

Zincir kuralı kullanılarak elde edilen kısmi türevleri tek tek alındıktan sonra genel fonksiyonda yerine yazıldı.

$$\frac{\partial H}{\partial w_i} = \frac{\partial H}{\partial \varphi} \cdot \frac{\partial \varphi}{\partial \alpha} \cdot \frac{\partial \alpha}{\partial w_i} \quad (18)$$

$$\frac{\partial H}{\partial w_i} = 2 \cdot \frac{1}{n} \sum_{i=1}^n (\varphi_i - \varphi_i^{\sim}) \cdot \varphi_i^{\sim} \cdot (1 - \varphi_i^{\sim}) \cdot x_i \quad (19)$$

elde edilir.

Elde edilen hata değerini minimum hale getirebilmek için kısmi türevlerini aldık. Optimizasyonu artık yapabiliriz. Optimizasyon elde bulunan verilerden en iyi elemanın seçilme işini yapmaktır. Yani bu yapay sinir ağı için en iyi ağırlığı seçmektir. Yeni ağırlık değerimizin hesaplaması şu şekilde yapılır.

$$\omega_i = w_i - \left(\theta \cdot \frac{\partial H}{\partial w_i} \right) \quad (1)$$

Bu işlem sonucu bize hata oranı en kullanışlı olan ağırlığı vermiş olur. Bu ise bizim yapay sinir ağıımızın çalışmasını sağlar. İleri beslemeli yapay sinir ağı çözümlerini verilen birinci mertebeden diferansiyel denklem üzerinde şöyle gösteririz:

Birinci mertebeden diferansiyel denklem için başlangıç değer problemi $t \in [a, b]$ olmak üzere $t = t_0 = a$ için

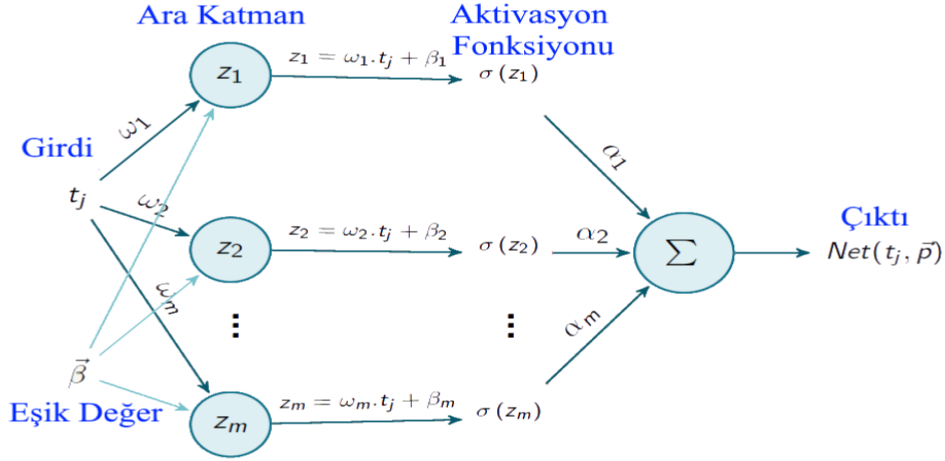
$$\begin{cases} y' = f(t, f(t)) \\ y(t_0) = y_0 \end{cases} \quad (21)$$

olarak tanımlıdır.

Tanımı verilen başlangıç değer probleminin yapay sinir ağı ile çözümünü belirleyelim. n toplam girdi sayısı ve m ara katmandaki nöron sayısı olsun. İleri beslenmeli yapay sinir ağının parametrelerini $\alpha \omega \beta \in R^m$ ve $\vec{p} = (\vec{\alpha}, \vec{\omega}, \vec{\beta})$ olarak belirleyelim. İleri beslenmeli yapay sinir ağının çıktısı p parametre değeri için $Net(t_j, \vec{p})$ ile gösterilir. $j = 1, 2, 3, \dots, n$ ve $t_j \in [a, b]$ aralığındadır.

Verilen başlangıç değer problemini çözümü için yapay sinir ağı çıktısına bağlı olarak başlangıç koşullarını sağlayan deneme fonksiyonumuz.

$$y(t, p) = y_0 + (t - t_0)Net(t, p) \quad (22)$$



Şekil 12. İleri beslemeli yapay sinir ağının topolojik yapısı olarak ifade edilir.

Verilen şekilde adi diferansiyel denklemlerin ileri beslemeli yapay sinir ağının gösterimidir. Burada m tane ara katmandaki nöron sayısını belirtmektedir. $i = 1, 2, 3, \dots, m$ değerleri için z_i ara katmanındaki i . nöronu temsil eder. t_j girdisi için z_i nöronunun çıktısının hesaplaması ω_i bu nöronun ağırlığı ve β_i eşik değeri olsun. $z_i = t_j \omega_i + \beta_i$ denklemi ile z_i nöronun çıktısı elde edilmiş olur. Bu çıktı aktivasyon fonksiyonu ile işleme tabi tutularak ağırlıklandırılır. Ağırlıklandırılmış çıktıların toplamı yapay sinir ağının çıktısını belirler. Yani $\vec{p} = (\vec{\alpha}, \vec{\omega}, \vec{\beta})$ bilinmeyen parametreleri olmak üzere $Net(t_j, \vec{p}) = \sum_{i=1}^m \alpha_i \sigma(z_i)$ bu toplam ile ağırlıklandırılmış ileri beslemeli yapay sinir ağının çıktısı oluşturulur.

Burada belirtilen $\sigma(z_i)$ aktivasyon fonksiyonudur. Literatürde en çok kullanılan aktivasyon fonksiyonu sigmoid fonksiyonudur. Bu sebeple burada aktivasyon fonksiyonu olarak sigmoid fonksiyonu kullanılmıştır. Bu fonksiyonun ifadesi şu şekildedir

$$\sigma(z_i) = \frac{1}{1 + e^{-z_i}} \quad (23)$$

Başlangıçta rasgele belirlenen parametre değerleri ile ağın çıktısı hesaplanıyor. Bu değerler içinde belirli bir hata değeri oluşuyor. Bu hata değerini minimum hale getirmek amacıyla genellikle geri yayılım algoritması kullanılır. Bu yöntem $\vec{p}$ bilinmeyen parametreleri güncellenerek oluşan hata tüm ağa yayılır. İleri beslemeli yapay sinir ağlarında hata

$$H = \frac{1}{2} \sum_{i=1}^n (d_i - y_i)^2 \quad (24)$$

d_i değeri t_j girdisi için beklenen sonuç y_i değeri ise ağın gerçekte ürettiği çıktı değeridir. Adi diferansiyel denklemin çözümünde oluşan hatanın hesaplanması için maliyet fonksiyonu t_j için

$$h_j = \frac{1}{2} \left(\frac{\partial y_t}{\partial t_j} - f(t_j, y_j) \right)^2 \quad (25)$$

eşitliği ile hesaplanır. Bütün girdiler için oluşan toplam hata değerlerinin maliyet fonksiyonu

$$H = \sum_{j=1}^n h_j = \sum_{j=1}^n \frac{1}{2} \left(\frac{\partial y_t}{\partial t_j} - f(t_j, y_j) \right)^2 \quad (26)$$

ile hesaplanır.

Bu maliyet fonksiyonunu minimum hale getirilmesi öğrenme düzeyinin arttığını gösterir. Bu sebeple amacımız bu fonksiyonu minimum hale getirmektir. φ 0 ile 1 arasında değer alan öğrenme katsayımız olsun. Oluşan hatayı minimum hale getirebilmek için bilinmeyen $\alpha_i, \omega_i, \beta_i$ değerlerini güncellemek gerekir. Bu amaçla Gradyan Düşüm Algoritması kullanılır.

Bu yaklaşıma göre yeni parametre değerleri aşağıdaki gibi hesaplanır.

$$\alpha_i = \alpha_i - \varphi \frac{\partial H}{\partial \alpha_i} \quad (2)$$

$$\omega_i = \omega_i - \varphi \frac{\partial H}{\partial \omega_i}$$

$$\beta_i = \beta_i - \varphi \frac{\partial H}{\partial \beta_i}$$

şeklinde yeni parametreler güncellenir.

Parametre değerleri güncellenirken maliyet fonksiyonun bilinmeyen parametrelere göre kısmi türevleri $i = 1, 2, 3, \dots, m$ için

$$\frac{\partial y_t}{\partial t_j} = Net(t_j, \vec{p}) + (t_j - t_0) \frac{\partial Net(t_j, \vec{p})}{\partial t_j} \quad (3)$$

$$\frac{\partial Net(t_j, \vec{p})}{\partial t_j} = \sum_{i=1}^m \alpha_i \omega_i \sigma(z_i) (1 - \sigma(z_i))$$

ve

$$\frac{\partial Net^2(t_j, \vec{p})}{\partial t_j^2} = \sum_{i=1}^m \alpha_i \omega_i^2 \sigma(z_i)(1 - \sigma(z_i))(1 - 2\sigma(z_i))$$

olmak üzere aşağıdaki eşitlikler sıfıra eşitlenir.

$$\frac{\partial H}{\partial \alpha_i} = \sum_{j=1}^n \left(\frac{\partial y_T}{\partial t_j} - f(t_j, y_j) \right) \left(\frac{\partial^2 y_T}{\partial \alpha_i \partial t_j} - \frac{\partial f(t_j, y_j)}{\partial \alpha_i} \right) = 0$$

$$\frac{\partial H}{\partial \omega_i} = \sum_{j=1}^n \left(\frac{\partial y_T}{\partial t_j} - f(t_j, y_j) \right) \left(\frac{\partial^2 y_T}{\partial \omega_i \partial t_j} - \frac{\partial f(t_j, y_j)}{\partial \omega_i} \right) = 0$$

$$\frac{\partial H}{\partial \beta_i} = \sum_{j=1}^n \left(\frac{\partial y_T}{\partial t_j} - f(t_j, y_j) \right) \left(\frac{\partial^2 y_T}{\partial \beta_i \partial t_j} - \frac{\partial f(t_j, y_j)}{\partial \beta_i} \right) = 0$$

Kısmi türevlerdeki $\frac{\partial^2 y_T}{\partial \alpha_i \partial t_j}$, $\frac{\partial^2 y_T}{\partial \omega_i \partial t_j}$ ve $\frac{\partial^2 y_T}{\partial \beta_i \partial t_j}$ değerleri sırasıyla

$$\frac{\partial Net(t_j, \vec{p})}{\partial \alpha_i} = \sigma(z_i)$$

ve

$$\frac{\partial Net^2(t_j, \vec{p})}{\partial \alpha_i \partial t_j} = \omega_i \sigma(z_i)(1 - \sigma(z_i))$$

olmak üzere;

$$\frac{\partial^2 y_T}{\partial \alpha_i \partial t_j} = \frac{\partial Net(t_j, \vec{p})}{\partial \alpha_i} - (t_j - t_0) \frac{\partial Net^2(t_j, \vec{p})}{\partial \alpha_i \partial t_j}$$

$$\frac{\partial^2 y_T}{\partial \beta_i \partial t_j} = \frac{\partial Net(t_j, \vec{p})}{\partial \beta_i} - (t_j - t_0) \frac{\partial Net^2(t_j, \vec{p})}{\partial \beta_i \partial t_j}$$

$$\frac{\partial^2 y_T}{\partial \omega_i \partial t_j} = \frac{\partial Net(t_j, \vec{p})}{\partial \omega_i} - (t_j - t_0) \frac{\partial Net^2(t_j, \vec{p})}{\partial \omega_i \partial t_j}$$

Şeklinde hesaplamalar yapılır.

Yapay sinir ağlarının çözüm metotları bu şekilde yapılmaktadır. Yapay sinir ağının çözüm metoduna ek olarak aktivasyon sayısını artırarak yeni bir çözüm yapmak istiyoruz. Artırdığımız aktivasyon fonksiyonu dalgacık dönüşümleri içerisinde olacaktır. Aktivasyon fonksiyonunun yapay sinir ağına etkisi neler olabileceği hakkında bir sonuca varmak, yararları

ve zararları hakkında bilgi sahibi olabilmek için yapay sinir ağı üzerinde bu değişimi yapmak istiyoruz.

Dalgacık Dönüşümlü Aktivasyon Fonksiyonlarının Çözüm Metotları

Dalgacık dönüşümlü aktivasyon fonksiyonları son zamanların dikkat çeken popüler bir çözüm modeli halini almaya başlamıştır.

Dalgacık fonksiyonları, fonksiyonların yerel özelliklerini yakalamada ve çeşitli ölçeklerdeki bilgiyi temsil etmede oldukça etkilidir. Bu nedenle, özellikle zaman veya mekân açısından sürekli değişkenlik gösteren problemler için kullanımı uygundur. Bu nedenle sinyal işleme ve zaman-frekans analizinde dalgacık dönüşümleri kullanıldığında, ağı öğrenme kapasitesini artırabilir.

Aktivasyon Fonksiyonu Tasarımı

Aktivasyon fonksiyonu tasarımı yapılırken bilinen aktivasyon fonksiyonlarından farklı olarak kullanılan fonksiyonlar verilen modeller için uygulanabilir olmalıdır. Geri yayılım algoritması kullanılacağı için seçilen aktivasyon fonksiyonu sürekli olmalıdır. Geri yayılım gradyanlarının sorunsuz hesaplanabilmesi için türevlenebilir olmalıdır. Fonksiyonun çıkış aralığı öğrenme sürecini etkilediği için çıkış aralığı belirlenmelidir. Fonksiyonun yüksek derecede non-linner olması YSA'nın karmaşık ilişkilerini çözmede yardımcı olmasını sağlar.

Dalgacık özellikleri ile aktivasyon fonksiyonu oluştururken YSA da şu şekilde değişimler gözlenebilir:

1. Dalgacıklar, verileri farklı ölçeklerde ve çözünürlüklerde analiz edebildiği için sinir ağlarında verilerin detaylarını öğrenmek için faydalı olabilir.
2. Dalgacıklar genellikle doğrusal olmayan özelliklere sahip olması nedeniyle sinir ağlarında doğrusal olmayan ilişkilerini öğrenmek için uygun olabilirler.
3. Dalgacıklar, sinyal ve görüntü işleme uygulamalarında başarılı olması özellikle görüntü işleme görevlerinde sinir ağlarıyla kombinasyonlarını kullanmak için kullanılacak güzel bir yöntem olabilir.

Dalgacık Aktivasyon Fonksiyonlarının Matematiksel Yapısı

Dalgacıklar aktivasyon fonksiyonu olarak kullanıldığında bilinen standart aktivasyon fonksiyonları gibi çalışır. Aktivasyon fonksiyonu YSA'nın girdi değerleri ile ağırlıklandırıldıktan sonra net girdinin istenilen aralığa indirgeyebilmek için kullanılan yardımcı fonksiyondur. Bu fonksiyonu biz iki farklı aktivasyon fonksiyonu kullanarak

modelimizin eğimi üzerindeki etkiyi gözlemlemek istedik. İlk kullanılan aktivasyon fonksiyonumuz dalgacık dönüşümlü aktivasyon fonksiyonu olurken ikinci kullandığımız aktivasyon fonksiyonu bilinen standart aktivasyon fonksiyonlarından birisi olarak belirlendi. İlk aktivasyon fonksiyonumuzun dalgacıklardan seçilmesi küçük frekanslara karşı duyarlı olması ve bu duyarlılığı ile girdi değerlerinde temizleme veya normal aktivasyon fonksiyonunun atladığı değerleri yakalama amaçlanmıştır. Aktivasyon fonksiyonu olarak kullanılan dalgacık dönüşümlü aktivasyon fonksiyonlarının genel yapısı vardır. Önceki bölümde bahsettiğimiz dalgacıkların aktivasyon fonksiyonu olarak kullanımının matematiksel formları şu şekildedir:

Dalgacık(wavelet) aktivasyon fonksiyonlarının genel ifadesi

$$\varphi(t) = \rho(t) \cdot g(t) \quad (4)$$

Burada;

$\rho(t)$: Ölçek Fonksiyonu (bilgi çıkarımı sağlar)

$g(t)$: Dalgacık Fonksiyonu (detay çıkarımını sağlar)

Morlet Dalgacı

$$\varphi(t) = \cos(2\pi t) \cdot e^{-t^2} \quad (5)$$

Mexican Hat

$$\varphi(t) = (1 - t^2) \cdot e^{-\frac{t^2}{2}} \quad (6)$$

Haar Dalgacı

Parça parça fonksiyonlar olduğu için ayrık işlemlerde kullanılır.

Genocchi Dalgacı

$$\psi(t) = G_n(t) \cdot e^{-t^2} \quad (7)$$

Burada;

$\psi(t)$: Genocchi dalgacık fonksiyonu

$G_n(t)$: Genocchi polinomu n -ci dereceye bağlıdır

e^{-t^2} : Fonksiyonun yerel etkisini artırabilmek için kullanılan Gaussian ağırlığıdır.

Genocchi polinomları Bernoulli polinomlarının türeviyle ilişkilidir.

$$G_n(t) = 2 \cdot n \sum_{k=0}^{n-1} \binom{n-1}{k} \cdot B_k t^{n-1-k}$$

B_k : k -cı derece Bernoulli sayısı

n : Polinomun derecesi

$$G_0(t) = 0$$

$$G_1(t) = 1$$

$$G_2(t) = 2x$$

$$G_4(t) = 6x^2 - 1$$

örnek olarak ilk 4 terimini bu şekilde yazabiliriz.

Sinir ağlarında Genocchi aktivasyon fonksiyonları girdiyi dalgacık bileşenine ve polinom tabanlı özellik çıkarımına göre dönüştürüyor. Polinom terimleri olması girdiye hatasızlık kazandırır. Polinom kat sayılarının tam sayı olması işlemlerin daha hızlı çözümlenmesini sağlar. Gaussian ağırlık teriminin kullanılması küçük giriş değerleri için hassasiyet oluşturur ve bu değerlere odaklanmayı kolaylaştırır. Çok ölçekli yaklaşımlarda Genocchi polinomları sinyalleri daha iyi ayrıştırır.

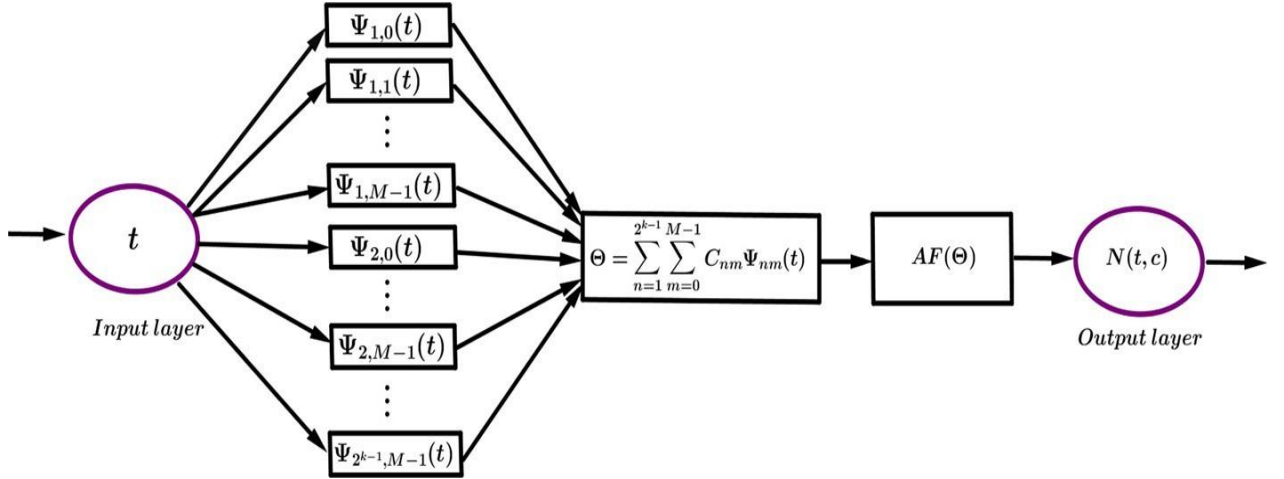
Bu bilgiler ile Genocchi aktivasyon fonksiyonu modelin sinyallerinin hem yerel hem de küresel özelliklerini analiz etmeye izin verir. Polinom derecesi ve diğer parametreler değiştirilebildiği için farklı problemler için kullanım olanağı sağlar ve optimize edilebilir.

Sinir ağının eğitiminde bu tarz aktivasyon fonksiyonları kullanıldığında ağın parametre güncellemelerinde türevleri hesaplanır. Bu dalgacık fonksiyonunun türevi şu şekilde ifade edilir:

$$\psi(t) = G_n(t) \cdot e^{-t^2} \quad (8)$$

$$\psi'_n(t) = G'_n(t) \cdot e^{-t^2} - 2xG_n(t) \cdot e^{-t^2} \quad (9)$$

Genocchi Yapay Sinir Ağının Çözüm Metodu



Şekil 13. Genocchi yapay sinir ağının yapısı

Yukarıda verilen şekilde aktivasyon fonksiyonu olarak ilk önce Genocchi Dalgacıkları alınmıştır. Ek olarak standart bilinen aktivasyon fonksiyonları ile modelin çözümüne devam edilmiştir. Genocchi Dalgacıklarının seçimi ise katsayılarının tam sayı olması ve terim sayısının az olması çözümü daha kolaylaştıracığı için seçilmiştir.

Verilen GWNN sinir ağının yapısını şöyle açıklayabiliriz:

- İlk katman, tek bir t düğümüne sahip olan ağın giriş katmanıdır ve bu düğüm ρ veri sayısı $\{t_1, t_2, \dots, t_\rho\}$.
- İkinci katman ağın gizli katmanıdır; bu katmana aktivasyon fonksiyonu olarak Genocchi dalgacıkları uygulanmıştır.
- Üçüncü katman ağın çıkış katmanıdır, girişi ikinci katmanın Genocchi dalgacıklarının doğrusal bir kombinasyonudur ve çıktısı bir aktivasyon fonksiyonunu ($AF(.)$) girişine yerleştirir.

Bu nedenle, t giriş verisi ve C parametresi ile GWNN'nin çıkışı aşağıdaki gibidir.

$$N(t, C) = AF(\Theta) \quad (10)$$

Burada Θ , Genocchi dalgacıklarının doğrusal bir kombinasyonudur ve şu şekilde yazılır.

$$\Theta = \sum_{n=1}^{2^{k-1}} \sum_{m=1}^{M-1} c_{n,m} \psi_{n,m}(t) = C^T \psi(t) \quad (11)$$

Burada C ve $\psi(t)$ vektörleri şu şekilde belirlenir.

$$C = [c_{1,0}, c_{1,1}, c_{1,2}, c_{1,3}, \dots, c_{2,0}, c_{2,1}, c_{2,M-1}, \dots, c_{2^{k-1},M-1}]^T, \quad (12)$$

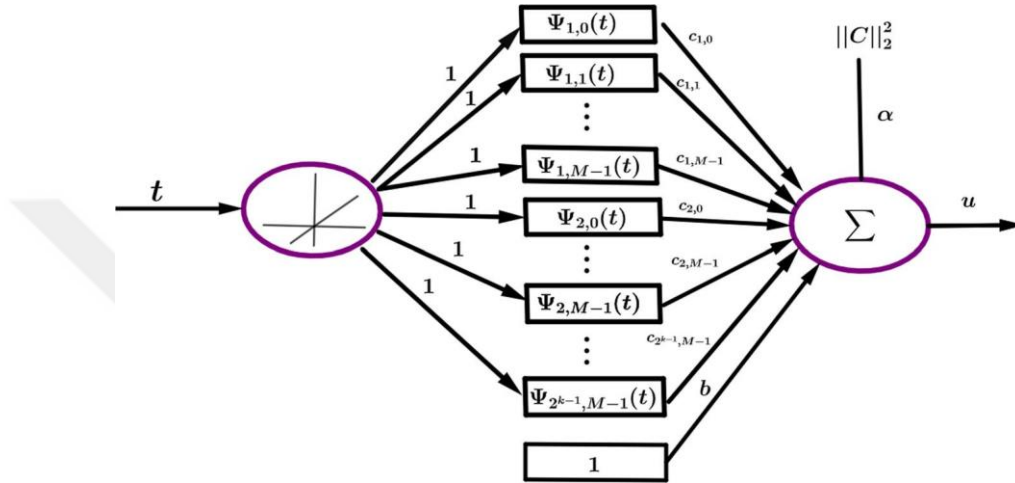
Ve

$$\psi(t) = [\psi_{1,0}(t), \psi_{1,1}(t), \psi_{1,M-1}(t), \dots, \psi_{2,0}(t), \psi_{2,1}(t), \psi_{2,M-1}(t), \dots, \psi_{2^{k-1},M-1}(t)]^T, \quad (13)$$

Burada $\psi_{n,m}(t)$, ($n = 1, 2, \dots, 2^{k-1}, m = 0, 1, \dots, M-1$) Genocchi dalgacıklarıdır.

Ayrıca,

$AF(.)$ Aktivasyon fonksiyonu etkileyen başka bir aktivasyon fonksiyonudur.



Şekil 14. Genocchi aktivasyon fonksiyonunun yapısı

Şekil 14 te Genocchi aktivasyon fonksiyonunun daha derin olarak gösterimi verilmiştir.

Dalgacık aktivasyon fonksiyonları genellikle zaman-frekans, finansal zaman seri tahminleri, kişisel veri analizi, sağlık veri analizi ve hastalık tahminlerinde, otoyol trafik tahmini gibi modellerde kullanılır.

ARAŞTIRMA BULGULARI

Bu bölümde Pima İndian Diabets veri seti ve Kişisel Veri Güvenliği veri setleri üzerine iki farklı örnek hesaplaması yapılacaktır.

Test Problemi 1 (Diyabet Örneği)

Pima İndian Diabets verileri ile çözümlenmiş bir YSA modeline Genocchi aktivasyon fonksiyonunu ekleyerek yeni bir çözüm elde etmeye çalışmayı amaçladık. Öncelikle gizli katmanda standart bilinen ReLU aktivasyon fonksiyonunu çift katmanlı olarak kullandık. Daha sonra aynı veri seti üzerinde ağırlık ve bias değerleri değiştirilmeden sadece gizli katmanda ilk aktivasyon fonksiyonunu Genocchi dalgacık aktivasyon fonksiyonu ile değiştirip yeniden çözümlene yaptık. Bu iki örnek arasında karşılaştırma yapılarak dalgacık aktivasyon fonksiyonlarının avantajları ve dezavantajları üzerine yorumlamalar yaptık. Bu örneğimize benzer kişisel güvenlik sistemleri üzerine bir örnek eklemesi daha yaptık. Bu örnekte ise dalgacık aktivasyon fonksiyonu olarak Morlet dalgacık aktivasyon fonksiyonunu seçtik. Hastalık verisi üzerine belirlediğimiz modelin çözüm yöntemi tek aktivasyon fonksiyonu ile aşağıdaki gibidir:

Diyabet tahmini için kullanılan veri seti Pima İndian Diabets dir. Bu veri setinin şu özellikleri vardır.

- Yaş
- Cinsiyet
- Vücut Kitle İndeksi
- Aile geçmişi
- Glikoz Seviyesi

Diyabet olma olasılığı:

Diyabet yok :0

Diyabet var:1

Bu sistemin YSA çözüm yöntemi şu şekilde olacak:

Girdi Katman: 5 özellik (girdi boyutu)

Gizli Katman: 8 katmanlı olacak aktivasyon fonksiyonu ise RELU

Çıkış Katman: Çıkış Katmanı ise diyabetli olarak verilmesi için güncellemeler yapılacaktır. Çünkü verilen veri seti 0.50 diyabetli hastalar üzerine yapılmış bir YSA modelidir.

Model Denklemi:

Giriş Katmanı:

$$z_i = w_i x_i + b_i \quad (14)$$

$$A_i(z_i) = \text{RELU}(z_i) \quad (15)$$

Çıkış Katmanı: Çıkış katmanında istediğimiz değerin (0,1) aralığı olmasından dolayı bulunan değer sigmoid fonksiyonu ile tekrar işleme tabi tutulacak.

$$z'_i = w'_i A_i + b'_i \quad (16)$$

w_i : Ağırlık Değerleri

x_i : Girdi Değerleri

b_i : Bias Değerleri

$A_i(z_i)$: RELU aktivasyon fonksiyonu $\text{RELU}(z_i) = \max(0, z_i)$

$$\text{Sigmoid}(z_i) = \frac{1}{1 + e^{-z_i}} \quad (17)$$

Modelin eğitimi sırasında kayıp fonksiyonu optimizasyon algoritması kullanılır. Eğitim süresince ağırlıklar güncellenerek istenilen değere yakın sonuçlar optimize edilerek ulaşılır.

Kayıp Fonksiyonu:

Modelin istenilen değere yakınlığı hakkında bilgi verir. Bu model için Binary cross-entropy kayıp fonksiyonu kullanılmaktadır. Aşağıdaki gibi tanımlanır:

$$H(Y, Y') = -((Y \log(Y') - (1 - Y) \cdot \log(1 - Y')) \quad (18)$$

Y : Gerçek Değer (0 veya 1)

Y' : Modelin Tahmin Ettiği Değer

Ağırlıkların Güncellemesi:

Model geri yayılım algoritmasıyla (backpropagation) güncellenir. Ağırlık güncellemeleri şu şekildedir:

$$W'_i = W_i - \varphi \frac{\partial H}{\partial W_i}$$

$$W'_j = W_j - \varphi \frac{\partial H}{\partial W_j}$$

φ : Öğrenme Katsayısı (0,1) arasında değerler alır.

İlk güncelleme RELU aktivasyon fonksiyonu ikinci ise Sigmoid fonksiyonu için yazılmıştır.

Bu örneğimizin daha derin denklemleri şu şekilde ifade edilir:

Girdi Katmanı

$$\mathbf{x}_i = [x_1, x_2, x_3, x_4, x_5]^T \quad (19)$$

Gizli Katman

Ağırlık Matrisi:

$$\mathbf{w}_{ij} = \begin{bmatrix} w_{11} & w_{12} & w_{13} & w_{14} & w_{15} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ w_{n1} & w_{n2} & w_{n3} & w_{n4} & w_{n5} \end{bmatrix} \quad (20)$$

Bias Matrisi:

$$\mathbf{b}_i = \begin{bmatrix} b_1 \\ \vdots \\ b_n \end{bmatrix} \quad (21)$$

Gizli katmandan çıkan nöronlarımız:

$$\mathbf{z}_i = \mathbf{w}_i \mathbf{x}_i + \mathbf{b}_i \quad (22)$$

$$\mathbf{z}_i = \begin{bmatrix} w_{11} & w_{12} & w_{13} & w_{14} & w_{15} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ w_{n1} & w_{n2} & w_{n3} & w_{n4} & w_{n5} \end{bmatrix} \cdot [x_1, x_2, x_3, x_4, x_5]^T + \begin{bmatrix} b_1 \\ \vdots \\ b_n \end{bmatrix} \quad (23)$$

Aktivasyon fonksiyonumuz ile işleme tabi tutulur. Giriş katmanından çıkan değerlerimiz:

$$\mathbf{A}_i(\mathbf{z}_i) = \mathbf{RELU}(\mathbf{z}_i) \quad (49)$$

$$\mathbf{A}_i \left(\begin{bmatrix} w_{11} & w_{12} & w_{13} & w_{14} & w_{15} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ w_{n1} & w_{n2} & w_{n3} & w_{n4} & w_{n5} \end{bmatrix} \cdot [x_1, x_2, x_3, x_4, x_5]^T + \begin{bmatrix} b_1 \\ \vdots \\ b_n \end{bmatrix} \right) = \mathbf{RELU}(\mathbf{z}_i) = \mathbf{max}(\mathbf{0}, \mathbf{z}_i) \quad (24)$$

Buradan elde edilen değerlerimizle çıkış katmanını belirledikten sonra sigmoid aktivasyon fonksiyonu ile işleme tabi tutarak net çıktı hesaplanır. Burada oluşan değerlerimiz sigmoid fonksiyonu için yeni girdiler olduğu için ağırlık ve bias değerlerini bu işlemler ışığında şöyle yazabiliriz.

$$\mathbf{w}_j = [w_{j1} \quad w_{j2} \quad w_{j3} \quad \dots \quad \dots \quad w_{jn}] \quad (25)$$

b_j skaler sayı

$$\mathbf{z}_j = \mathbf{w}_j \mathbf{A}_i + \mathbf{b}_j \quad (26)$$

$$\mathbf{z}_j = [\mathbf{w}_{j1} \quad \mathbf{w}_{j2} \quad \mathbf{w}_{j3} \quad \dots \quad \mathbf{w}_{jn}]. \mathbf{A}_i + \mathbf{b}_j \quad (53)$$

Bu işlemler ile modelin tahmini değeri hesaplanmış olur. Tahmini değer ile gerçek değer arasındaki fark ile kayıp fonksiyonu elde edilir. Kayıp fonksiyonun değerini bulurken kullandığımız Binary cross-entropy kayıp fonksiyonudur.

$$H(Y, Y') = -((Y \log(Y')) - (1 - Y) \cdot \log(1 - Y'))$$

Kayıp fonksiyonunun sonuçları ile ağırmızı yeniden güncelleriz.

$$W'_i = W_i - \varphi \frac{\partial H}{\partial W_i}$$

$$W'_j = W_j - \varphi \frac{\partial H}{\partial W_j}$$

şeklinde hesaplanır.

Pima Indian Diabetes veri seti, diyabet teşhisi için kullanılan bir dizi sağlık özelliği içeren bir modeldir. Veri setinde her örnek 8 özellik ve tek hedef değişkeni içeren bir modeldir. Bu modelin katmanları hakkında detaylı bilgi verelim.

1. Giriş Katmanı:

- Giriş verileri (örneğin, hastanın yaş, cinsiyet gibi özellikleri) bu katmana beslenir.
- Her bir özellik, ağırlık matrisi ile çarpılır ve çarpım sonucu bias değeri ile toplanır.

2. Gizli Katman:

- Giriş katmanından gelen verilerin doğrusal kombinasyonunu elde etmemizi sağlar.
- ReLU aktivasyon fonksiyonu, bu doğrusal kombinasyonu negatif olmayan bir değere dönüştürür ve modelin non-lineer özellikler öğrenmesini sağlar.

3. Çıkış Katmanı:

- Gizli katmanın çıktısını besler.
- Sigmoid aktivasyonu verileri 0 ile 1 değeri arasında bir değere dönüştürür.
- Bu değer, diyabet olma olasılığını elde etmemizi sağlar.

Bu bilgiler ile Pima Indians Diabetes Dataset veri setini göz önüne alarak bir örnek belirtelim.

Pima Indian Diabetes veri seti, diyabet teşhisi için kullanılan bir dizi sağlık özelliği içeren bir modeldir. Veri setinde her örnek 8 özellik ve tek hedef değişkeni içeren bir modeldir. Örnek özellikleri şu şekilde tabloya yerleştirebiliriz.

Tablo 3. Diyabet Teşhisi İçin Kullanılan Özellikler

ÖZELLİK	AÇIKLAMA
Glucose	Glikoz seviyeleri
BloodPressure	Kan basıncı
Pregnancies	Hamilelik sayısı
Insulin	İnsülin seviyesi
Age	Yaş
BMI	Vücut kitle indeksi (BMI)
DiabetesPedigreeFunction	Aile geçmişi puanı
Outcome	Diyabet durumu (0: yok, 1: var)
SkinThickness	Cilt kalınlığı

Özellikler standartlaştırılarak ön eleme yapılır. Eğitim ve test verileri ayrılır. %70 eğitim %30 test olarak ayrılır.

Örneğimizde hasta özellikleri şu şekilde verilsin;

$$x = [6,148,72,35,0,33.6,0.625,50]$$

Tablo 4. Hasta Özellik Değer Tablosu

Özellik	Değer
x_1	6
x_2	148
x_3	72
x_4	35
x_5	0
x_6	33.6
x_7	0.625
x_8	50

Gizli katmanın boyutunu 16 boyutludur. Ağırlık matrisi W^1 ve bias B^1 vektörü rastgele başlatılmış değerlerdir. Bu veri setinin 3 nöronu için net girdi şöyle hesaplanır:

$$Z^1(x) = W^1x + B^1$$

Tablo 5. Ağırlık ve Bias Değerleri

Ağırlık W^1	Bias B^1
$w_{11} = 0.1, w_{12} = -0.2, w_{13} = 0.3, \dots$	$b_1 = 0.5$
$w_{21} = -0.4, w_{22} = 0.5, w_{23} = -0.6, \dots$	$b_2 = -0.1$
$w_{31} = 0.7, w_{32} = -0.8, w_{33} = 0.9, \dots$	$b_3 = 0.2$
.....	

Ağırlık ve bias değerlerine göre gizli katman hesaplamaları yapılır.

$$z_i = \sum_{j=1}^8 w_{ji}x_i + b_j$$

Net girdi değerleri bu denklem ile hesaplanır. Aşağıdaki tabloda birinci girdi değerine karşılık gelen net girdi değerlerinin hesaplaması yapılmıştır:

Tablo 6. Gizli Katman Net Girdi Değerleri

Nöron	Hesaplama	Sonuç z_i
1	$z_1 = 0.1 \cdot 6 + (-0.2) \cdot 148 + 0.3 \cdot 72 + \dots + 0.5$	$z_1 = 10.2$
2	$z_2 = -0.4 \cdot 6 + 0.5 \cdot 148 + (-0.6) \cdot 72 + \dots + (-0.1)$	$z_2 = -5.3$
3	$z_3 = 0.7 \cdot 6 + (-0.8) \cdot 148 + 0.9 \cdot 72 + \dots + 0.2$	$z_3 = 15.1$
...	...	...

Aktivasyon fonksiyonu ile bulduğumuz net girdi değerleri işleme tabi tutulur. Gizli katmandan çıkacak değerler belirlenir

$$h_j = ReLU(z_i) = \max(0, z_i)$$

Tablo 7. İlk Aktivasyon Fonksiyonundan Elde Edilen Değerler

Nöron	z_i	$h_j = ReLU(z_i)$
1	10.2	10.2
2	-5.3	0
3	15.1	15.1
...	...	...

İki aktivasyon fonksiyonu ile işlem yapıldığı için elde edilen çıktı değerleri yeni aktivasyon fonksiyonunun girdi değerleri olarak tekrar işleme tabi tutulur. Bu işlem sonucu ile elde edilen değerler çıktı fonksiyonuna yönlendirilecek olan değerlerdir.

Gizli katmanın ikinci aktivasyon fonksiyonu ile değerlerinin hesabını yukarıdaki çözüm metodu ile tekrar yaparak sonuçlar elde edilir. Bu değerler için de yine ilk üç nöronun çözüm tabloları şöyledir:

Tablo 8. Gizli Katmanın İkinci Ağırlık ve Bias Değerleri

Ağırlık W^2	Bias B^2
$w_{11} = 0.2, w_{12} = -0.3, w_{13} = 0.4, \dots$	$b_1 = 0.1$
$w_{21} = -0.5, w_{22} = 0.6, w_{23} = -0.7, \dots$	$b_2 = -0.2$
$w_{31} = 0.8, w_{32} = -0.9, w_{33} = 1.0, \dots$	$b_3 = 0.3$
...	...

Ağırlık ve bias değerleri ile ikinci net girdi sonuçları hesaplanır.

$$H^2 = Z^2 = W^2 h^1 + B^2$$

$$z_i = \sum_{l=1}^8 w_{ji} h_i + b_j$$

Tablo 9. Gizli Katmanın İkinci Net Toplam Değerleri

Nöron	Hesaplama	Sonuç
1	$z_1 = 0.2 \cdot 10.2 + (-0.3) \cdot 0.0 + 0.4 \cdot 15.1 + \dots + 0.1$	$z_1 = 8.5$
2	$z_2 = -0.5 \cdot 10.2 + 0.6 \cdot 0.0 + (-0.7) \cdot 15.1 + \dots + (-0.2)$	$z_2 = -12.3$
3	$z_3 = 0.8 \cdot 10.2 + (-0.9) \cdot 0.0 + 1.0 \cdot 15.1 + \dots + 0.3$	$z_3 = 20.1$
...	...	...

Bulunan bu değerlere aktivasyon fonksiyonu uygulanır.

$$h_j = ReLU(z_i) = \max(0, z_i)$$

Tablo 10. Bulunan İkinci Net Çıktı Değerlerinin Aktivasyon Fonksiyonu ile İşlem Sonuçları

Nöron	z_i	$h_j = ReLU(z_i)$
1	8.5	8.5
2	-12.3	0
3	20.1	20.1
...	...	...

Bu sonuçlar elde edildi. Bu sonuçlar çıkış katmanına yönlendirilir. Bulunan sonuç sigmoid fonksiyonu ile işleme tabi tutularak elde edilen değer diyabet oranını gösterir.

Çıkış katmanının ağırlıkları 8×1 , bias değeri ise 1 boyutludur.

Tablo 11. Çıkış Katmanı Ağırlık ve Bias Değerleri

Ağırlık W^3	Bias B^3
$w_1 = 0.1, w_2 = -0.2, w_3 = 0.3, \dots$	$b_1 = 0.1$

Son doğrusal kombinasyon ile çıkış katmanı net verileri hesaplanır.

$$Z^3 = W^3 h^2 + B^3$$

$$z^3 = 0.1 \cdot 8.5 + (-0.2) \cdot 0.0 + 0.3 \cdot 20.1 + \dots + 0.1 = 7.2$$

Bulunan çıktı değeri sigmoid fonksiyonu ile işleme alınır. Çünkü diyabet olasılığı 0 ile 1 arasında değer alır. Bu aralığa uygun aktivasyon fonksiyonu sigmoid fonksiyonudur.

$$y^* = \sigma(z^3) = \frac{1}{1 + e^{-z^3}}$$

$$y^* = \sigma(z^3) = \frac{1}{1 + e^{-7.2}} \cong 0.999$$

Bu sonuç bize verilen 1000 kişi içindeki veri setinin diyabetli olduğunu gösterir.

Bu örneğin ilk aktivasyon fonksiyonunu Genocchi aktivasyon fonksiyonu ile değiştirip yeniden çözümleme yapılarak sonuçlar analiz edilir.

Aynı örneğin birinci giriş katmanını Genocchi aktivasyon fonksiyonu ile ikinci gizli katmanda ReLU aktivasyon fonksiyonu kullanılacaktır. Elde edilen sonuçlar çıkış katmanına aktarılır. Çıkış katmanında sigmoid fonksiyonu kullanılarak sonuçlandırılır.

Modelin yapısı şu şekildedir:

Giriş Katmanı: 8 özellik

Gizli Katman 1: 16 nöron Genocchi Aktivasyon Fonksiyonu

Gizli Katman 2: 8 nöron ReLU aktivasyon fonksiyonu

Çıkış Katmanı: 1 nöron sigmoid fonksiyonu

Pima Indian Diabetes veri seti, diyabet teşhisi için kullanılan bir dizi sağlık özelliği içeren bir modeldir. Veri setinde her örnek 8 özellik ve tek hedef değişkeni içeren bir modeldir. Örnek özellikleri şu şekilde tabloya yerleştirebiliriz.

Tablo 12. Diyabet Teşhisi İçin Kullanılan Özellikler

ÖZELLİK	AÇIKLAMA
Glucose	Glikoz seviyeleri
BloodPressure	Kan basıncı
Pregnancies	Hamilelik sayısı
Insulin	İnsülin seviyesi
Age	Yaş
BMI	Vücut kitle indeksi (BMI)
DiabetesPedigreeFunction	Aile geçmişi puanı
Outcome	Diyabet durumu (0: yok, 1: var)
SkinThickness	Cilt kalınlığı

Özellikler standartlaştırılarak ön eleme yapılır. Eğitim ve test verileri ayrılır. %70 eğitim %30 test olarak ayrılır.

Giriş Katmanı:

İlk örnekte verilen girdi değerleri ile işlemler tekrarlanır.

$$x = [6, 148, 72, 35, 0, 33.6, 0.625, 50]$$

Tablo 13. Hasta Özellik Değer Tablosu

Özellik	Değer
x_1	6
x_2	148
x_3	72
x_4	35
x_5	0
x_6	33.6
x_7	0.625
x_8	50

Gizli katmanın boyutunu 16 boyutludur. Ağırlık matrisi W^1 ve bias B^1 vektörü rastgele başlatılmış değerlerdir. Bu veri setinin 3 nöronu için net girdi şöyle hesaplanır:

$$Z^1(x) = W^1x + B^1$$

Tablo 14. Ağırlık ve Bias Değerleri

Ağırlık W^1	Bias B^1
$w_{11} = 0.1, w_{12} = -0.2, w_{13} = 0.3, \dots$	$b_1 = 0.5$
$w_{21} = -0.4, w_{22} = 0.5, w_{23} = -0.6, \dots$	$b_2 = -0.1$
$w_{31} = 0.7, w_{32} = -0.8, w_{33} = 0.9, \dots$	$b_3 = 0.2$
.....	

Ağırlık ve bias değerlerine göre gizli katman hesaplamaları yapılır.

$$z_i = \sum_{j=1}^8 w_{ji} x_i + b_j$$

Net girdi değerleri bu denklem ile hesaplanır. Aşağıdaki tabloda birinci girdi değerine karşılık gelen net girdi değerlerinin hesaplaması yapılmıştır:

Tablo 15. Gizli Katman Net Girdi Değerleri

Nöron	Hesaplama	Sonuç z_i
1	$z_1 = 0.1 \cdot 6 + (-0.2) \cdot 148 + 0.3 \cdot 72 + \dots + 0.5$	$z_1 = 10.2$
2	$z_2 = -0.4 \cdot 6 + 0.5 \cdot 148 + (-0.6) \cdot 72 + \dots + (-0.1)$	$z_2 = -5.3$
3	$z_3 = 0.7 \cdot 6 + (-0.8) \cdot 148 + 0.9 \cdot 72 + \dots + 0.2$	$z_3 = 15.1$
...	...	...

Aktivasyon fonksiyonu ile bulduğumuz net girdi değerleri işleme tabi tutulur. Gizli katmandan çıkacak değerler belirlenir

$$h_j = \psi(z_j) = z_j \cdot e^{-\frac{z_j^2}{2}}$$

Tablo 16. İlk Aktivasyon Fonksiyonundan Elde Edilen Değerler

Nöron	z_j	$h_j = \psi(z_j)$
1	10.2	$10.2 \cdot e^{-\frac{10.2^2}{2}} \approx 0.0$
2	-5.3	$5.3 \cdot e^{-\frac{(-5.3)^2}{2}} \approx 0.0$
3	15.1	$15.1 \cdot e^{-\frac{15.1^2}{2}} \approx 0.0$
...	...	...

İki aktivasyon fonksiyonu ile işlem yapıldığı için elde edilen çıktı değerleri yeni aktivasyon fonksiyonunun girdi değerleri olarak tekrar işleme tabi tutulur. Bu işlem sonucu ile elde edilen değerler çıktı fonksiyonuna yönlendirilecek olan değerlerdir.

Gizli katmanın ikinci aktivasyon fonksiyonu ile değerlerinin hesabını yukarıdaki çözüm metodu ile tekrar yaparak sonuçlar elde edilir. Bu değerler için de yine ilk üç nöronun çözüm tabloları şöyledir.

Tablo 17. Gizli Katmanın İkinci Ağırlık ve Bias Değerleri

Ağırlık W^2	Bias B^2
$w_{11} = 0.2, w_{12} = -0.3, w_{13} = 0.4, \dots$	$b_1 = 0.1$
$w_{21} = -0.5, w_{22} = 0.6, w_{23} = -0.7, \dots$	$b_2 = -0.2$
$w_{31} = 0.8, w_{32} = -0.9, w_{33} = 1.0, \dots$	$b_3 = 0.3$
...	...

Ağırlık ve bias değerleri ile ikinci net girdi sonuçları hesaplanır.

$$H^2 = Z^2 = W^2 h^1 + B^2$$

$$z_i = \sum_{l=1}^8 w_{ji} h_i + b_j$$

Tablo 18. Gizli Katmanın İkinci Net Toplam Değerleri

Nöron	Hesaplama	Sonuç
1	$z_1 = 0.2 \cdot 0.0 + (-0.3) \cdot 0.0 + 0.4 \cdot 0.0 + \dots + 0.1$	$z_1 = 0.1$
2	$z_2 = -0.5 \cdot 0.0 + 0.6 \cdot 0.0 + (-0.7) \cdot 0.0 + \dots + (-0.2)$	$z_2 = -0.2$
3	$z_3 = 0.8 \cdot 0.0 + (-0.9) \cdot 0.0 + 1.0 \cdot 0.0 + \dots + 0.3$	$z_3 = 0.0$
...	...	...

Bulunan bu değerlere aktivasyon fonksiyonu uygulanır.

$$h_j = ReLU(z_i) = \max(0, z_i)$$

Tablo 19. Bulunan İkinci Net Çıktı Değerlerinin Aktivasyon Fonksiyonu ile İşlem Sonuçları

Nöron	z_i	$h_j = ReLU(z_i)$
1	0.1	0.1
2	-0.2	0
3	0.3	0.3
...	...	...

Bu sonuçlar elde edildi. Bu sonuçlar çıkış katmanına yönlendirilir. Bulunan sonuç sigmoid fonksiyonu ile işleme tabi tutularak elde edilen değer diyabet oranını gösterir.

Çıkış katmanının ağırlıkları 8×1 , bias değeri ise 1 boyutludur.

Tablo 20. Çıkış Katmanı Ağırlık ve Bias Değerleri

Ağırlık W^3	Bias B^3
$w_1 = 0.1, w_2 = -0.2, w_3 = 0.3, \dots$	$b_1 = 0.1$

Son doğrusal kombinasyon ile çıkış katmanı net verileri hesaplanır.

$$Z^3 = W^3 h^2 + B^3$$

$$z^3 = 0.1 \cdot 0.1 + (-0.2) \cdot 0.0 + 0.3 \cdot 0.3 + \dots + 0.1 = 0.2$$

Bulunan çıktı değeri sigmoid fonksiyonu ile işleme alınır. Çünkü diyabet olasılığı 0 ile 1 arasında değer alır. Bu aralığa uygun aktivasyon fonksiyonu sigmoid fonksiyonudur.

$$y^* = \sigma(z^3) = \frac{1}{1 + e^{-z^3}}$$

$$y^* = \sigma(z^3) = \frac{1}{1 + e^{-0.2}} \cong 0.55$$

Bu sonuç bize verilen 1000 kişi içindeki veri setinin diyabetli olduğunu gösterir.

Bu sonuçlar göz önüne alınarak iki çözüm arasında yorumlama yapılmadan önce kullanılan aktivasyon fonksiyonlarının avantajları ve dezavantajları hakkında şöyle yorumlar yapılır.

ReLU Aktivasyon Fonksiyonu

Avantajları:

ReLU, matematiksel olarak basit bir fonksiyon olmasından dolayı hesaplamaları kolaydır hızlı sonuç verir. Eğimin daha hızlı olmasını sağlar.

ReLU, pozitif değerler için gradyanı sabit (1) tutar, bu da gradyan kaybı (vanishing gradient) problemini azaltır.

ReLU, negatif değerleri sıfıra eşitler, bu da modelin daha seyrek bir temsil öğrenmesini sağlamaktadır.

Dezavantajları

ReLU, negatif değerler için sıfır çıktı verir. Bu durumda, bazı nöronlar "ölür" ve eğitim sürecine katkıda bulunmaz bu ise eğitimi zorlaştırır.

ReLU, sadece pozitif değerler için non-lineer davranır. Negatif değerler için bu davranışı gösteremez. Karmaşık problemlerde sonuç vermeyebilir.

2. Genocchi Dalgacık Aktivasyon Fonksiyonu

Avantajları:

Genocchi dalgacık fonksiyonu hem pozitif hem de negatif değerler için non-lineer bir dönüşüm sağlar. Bu, modelin daha karmaşık veri setlerini öğrenmesini sağlar.

Dalgacık fonksiyonları, sinyal işleme ve örüntü tanıma gibi alanlarda başarılı olduğu bilinen yapılardır. Bu özellik, modelin verilerdeki ince detayları yakalamasını sağlar ve öğrenmenin doğruluğunu artırır.

Genocchi dalgacık fonksiyonu, farklı giriş değerleri için farklı gradyanlar üreten aktivasyon fonksiyonlarıdır. Modelin daha esnek öğrenmesini sağlar.

Dezavantajları:

Genocchi dalgacık fonksiyonu, ReLU'ya göre daha karmaşık bir matematiksel yapısı olduğu için hesaplama maliyetini artırır.

Genocchi dalgacık fonksiyonu, modelin eğitim verilerine aşırı uyum sağlama riskini artırabilir. Bu durum, düzenlileştirme ile kontrol edilmelidir.

Karşılaştırma ve Yorumlama:

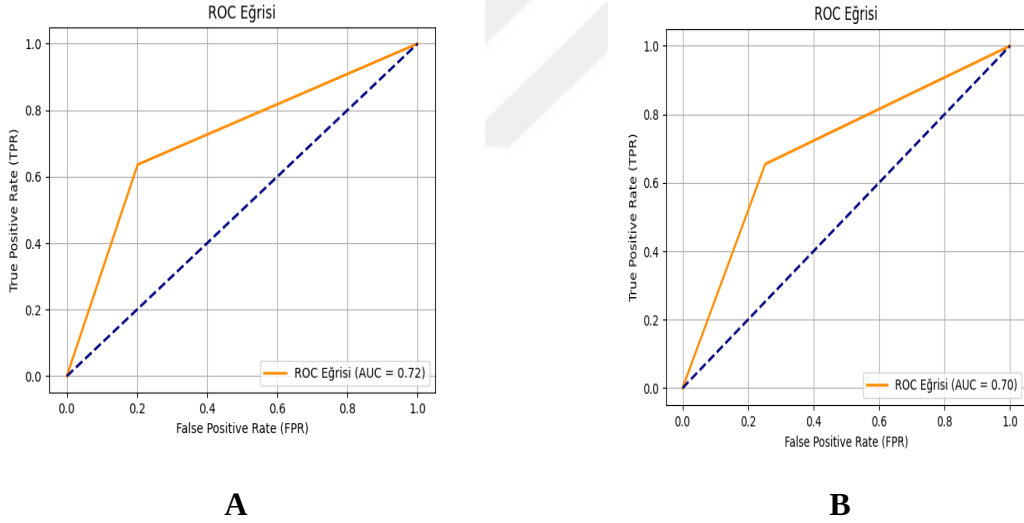
- ReLU: Genellikle daha hızlı eğitim süreleri ve daha düşük hesaplama maliyeti sunar. Ancak, ölü ReLU problemi ve sınırlı non-lineerlik nedeniyle bazı karmaşık problemlerde yetersiz kalabilir.
- Genocchi Dalgacık: Daha karmaşık non-lineer dönüşümler sağlar ve modelin daha ince detayları öğrenmesine yardımcı olur. Ancak, hesaplama maliyeti daha yüksektir ve aşırı uyum riski taşır.
- ReLU: Genel amaçlı yapay sinir ağı modellerinde yaygın olarak kullanılır. Özellikle derin öğrenme modellerinde tercih edilir.
- Genocchi Dalgacık: Sinyal işleme, örüntü tanıma ve karmaşık veri setleri gibi alanlarda daha etkili olabilir. Özellikle verilerdeki ince detayların önemli olduğu durumlarda tercih edilir.

- ReLU: Daha basit bir yapıya sahip olduğu için, genelleme yeteneği yüksek olabilir. Ancak, ölü ReLU problemi nedeniyle bazı durumlarda performans düşebilir.
- Genocchi Dalgacık: Daha karmaşık bir yapıya sahip olduğu için, eğitim verilerine aşırı uyum sağlama riski taşır. Bu durum, düzenleme teknikleri ile kontrol edilmelidir.

Bu bilgiler ile yaptığımız örneğin sonuç ve grafik yorumlamalarını şu şekilde yapabiliriz:

Sonuçlar göz önüne alınırsa sadece ReLU aktivasyon fonksiyonunun kullanımı ile değerimizi 0.999 olarak elde edilmiştir. İlk aktivasyon fonksiyonumuzu Genocchi dalgacığı ile değiştirince sonucumuz 0.55 olmuştur. Aldığımız veri setinin sonucu 0.5 doğruluk değeri ile oluşturulan veri seti olduğu için ikinci çözümümüz daha yakın sonuç vermiştir. Yani dalgacıklar daha hassas olmalarından dolayı daha yakın değerler elde etmemize yardımcı olmuştur. Bu örneğimizle alakalı grafiklerimiz şöyledir:

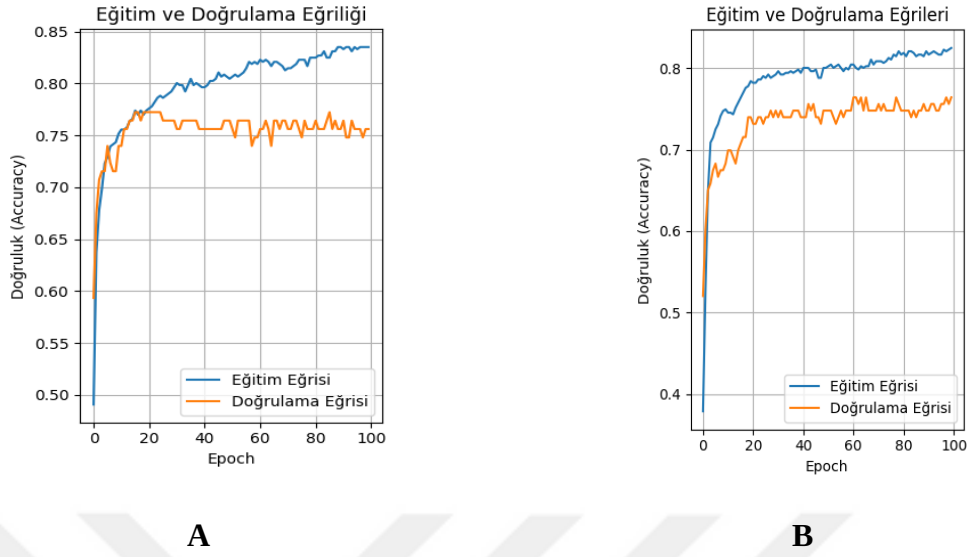
ROC grafiklerinin yorumu:



Şekil 15. A- genocchi dalgacık aktivasyon fonksiyonlu ve B-sadece ReLU aktivasyon fonksiyonlu ROC eğrileri

Bu çalışmada, ReLU ve Genocchi dalgacık aktivasyon fonksiyonları kullanılarak iki farklı yapay sinir ağı modeli oluşturulmuş ve Pima Indians Diabetes Dataset üzerinde karşılaştırmalı bir analiz yapılırsa sonuçlar, ReLU aktivasyon fonksiyonu kullanılan modelin AUC değerinin (0.70) Genocchi dalgacık aktivasyon fonksiyonu kullanılan modelin AUC değerinden (0.72) olduğunu görülmektedir. Ayrıca, Genocchi kullanılan modelin ROC eğrisi, sol üst köşeye daha belirgin bir eğilim göstermiştir. Bu bulgular, Genocchi aktivasyon

fonksiyonunun bu veri seti üzerinde daha iyi bir sınıflandırma performansı sergilediğini göstermektedir.



Şekil 16. A- Genocchi Dalgacık aktivasyon fonksiyonlu eğitim doğrulama eğrisi B- ReLU aktivasyon fonksiyonlu eğitim doğrulama eğrisi

Eğitim ve Doğrulama Doğruluğu eğrilerinin yorumlaması:

- Sadece ReLU aktivasyon fonksiyonu kullanılan:

Eğitim Doğruluğu: Yaklaşık 0.80 (veya %80).

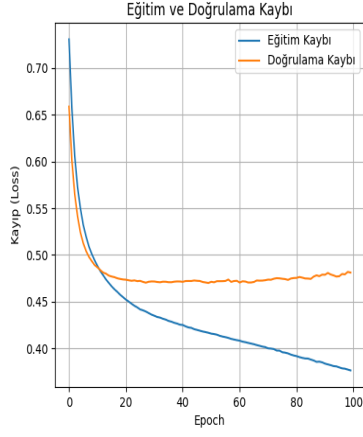
Doğrulama Doğruluğu: Yaklaşık 0.75 (veya %75)

- Genocchi aktivasyon fonksiyonu kullanılan:

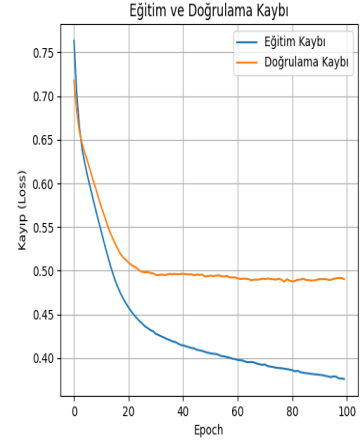
Eğitim Doğruluğu: Yaklaşık 0.85 (veya %85).

Doğrulama Doğruluğu: Yaklaşık 0.80 (veya %80)

Bu sonuçlar ile şu yorum yapılır. Genocchi dalgacık aktivasyon fonksiyonu kullanılan model, ReLU kullanılan modele göre daha yüksek doğruluk değerleri elde etmiştir. Bu, Genocchi dalgacık fonksiyonunun daha karmaşık desenleri öğrenme konusunda daha etkili olduğunu gösterir.



A



B

Şekil 17. A- Genocchi dalgacık aktivasyon fonksiyonlu doğrulama kaybı eğrisi B-ReLU aktivasyon fonksiyonlu doğrulama kaybı eğrisi

Eğitim ve Doğrulama Kaybı (Loss):

- Sadece ReLU aktivasyon fonksiyonu kullanılan:

Eğitim Kaybı: Yaklaşık 0.45 civarında.

Doğrulama Kaybı: Yaklaşık 0.50 civarında.

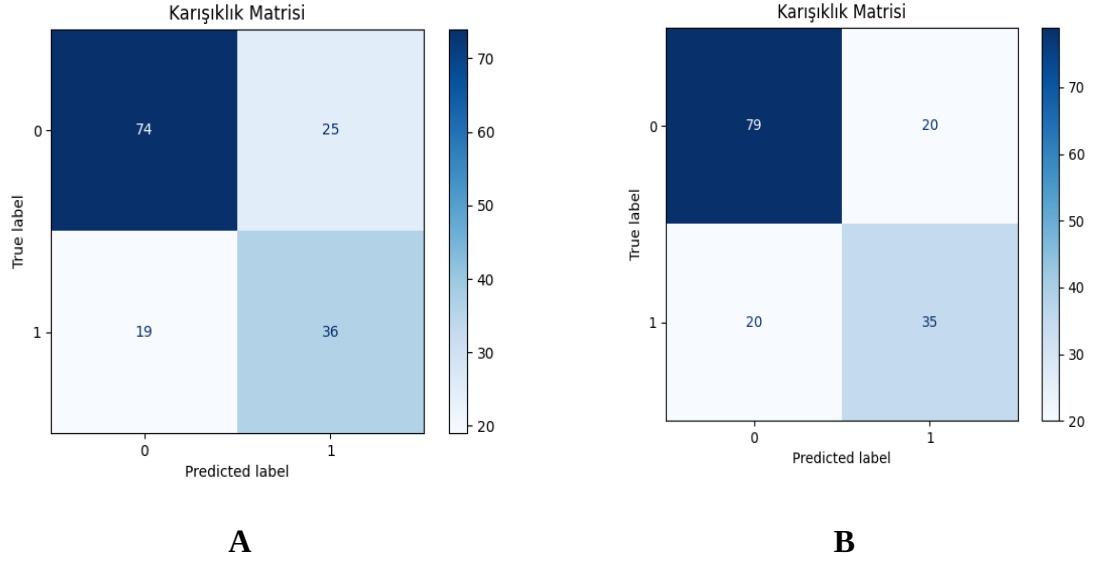
- Genocchi dalgacık aktivasyon fonksiyonu kullanılan:

Eğitim Kaybı: Yaklaşık 0.40 civarında.

Doğrulama Kaybı: Yaklaşık 0.45 civarında.

Elde edilen veriler ile şu yorumlama yapılır. Genocchi dalgacık aktivasyon fonksiyonu kullanılan model hem eğitim hem de doğrulama kaybı açısından daha düşük değerler elde etmiştir. Bu, modelin daha iyi öğrendiğini ve daha doğru tahminler yaptığını gösterir.

ReLU kullanılan modelin kayıp değerleri, Genocchi dalgacık kullanılan modele göre daha yüksektir. Bu, ReLU kullanılan modelin daha fazla hata yaptığını gösterir.



Şekil 18. A- Genocchi dalgacık aktivasyon fonksiyonlu modelin karışıklık matrisi B- ReLU aktivasyon fonksiyonlu modelin karışıklık matrisi

ReLU kullanılan model, negatif sınıfı (TN) daha iyi sınıflandırmıştır. 79- 74

Genocchi dalgacık kullanılan model, pozitif sınıfı (TP) biraz daha iyi sınıflandırmıştır 36- 35. Her iki model de TN ve TP değerleri açısından benzer performans göstermiştir.

ReLU kullanılan model, daha az yanlış pozitif (FP) tahmin yapmıştır 20- 25. Bu, modelin negatif sınıfı daha iyi ayırt ettiğini gösterir.

Genocchi dalgacık kullanılan model, daha az yanlış negatif (FN) tahmin yapmıştır 19- 20. Bu, modelin pozitif sınıfı daha iyi tespit ettiğini gösterir.

Bu çalışma ile ReLU ve Genocchi dalgacık aktivasyon fonksiyonları kullanılarak iki farklı yapay sinir ağı modeli oluşturuldu ve Pima Indians Diabetes Dataset üzerinde karşılaştırmalı bir analiz yapıldı. Karışıklık matrisi sonuçları, ReLU kullanılan modelin negatif sınıfı daha iyi sınıflandırdığını (daha az FP), Genocchi dalgacık kullanılan modelin ise pozitif sınıfı daha iyi tespit ettiğini (daha az FN) göstermiştir. Bu sebeple iki aktivasyon fonksiyonunun beraber kullanımı daha etkilidir.

Dalgacık dönüşümlü aktivasyon fonksiyonu olarak sadece Genocchi aktivasyon fonksiyonu kullanmak uygun olmayabilir. YSA ların yapısı gereği deneme-yanılma olduğu için aktivasyon fonksiyonu seçerken en iyi öğrenmeyi sağlayan aktivasyon fonksiyonunun seçimi YSA kullanımını daha sağlıklı hale getirir. Bu sebeple farklı bir alanda kullanılan bir modelin çözümünde farklı dalgacık aktivasyon fonksiyonu kullanılmalıdır. Bu aktivasyon fonksiyonunu kullanmanın sebebi sadece tek aktivasyon fonksiyonu değil birden fazla farklı tip aktivasyon fonksiyonu olduğunu göstermektir. Aktivasyon fonksiyonları YSA nın ağırlıkları gibidir. Eğer

aktivasyon fonksiyonunun eğitim sürecine etkisi olumlu yönde değilse değiştirilerek tekrar yeni bir aktivasyon fonksiyonu ile işleme tabi tutulur. Dalgacık aktivasyon fonksiyonları YSA ların daha hassas olan verilerinin yakalanmasına veya kullanılmasının önemli olmadığı verileri temizlemesine yardımcı olur. Bu sebeple kullanılan aktivasyon fonksiyonu seçimi çok önemlidir.

Test Problemi 2 (Kişisel Veri Güvenliği Örneği)

Kişisel veri güvenliği üzerine veri seti oluşturalım. Bu örnekte kişinin oturumuna girişinin sahte olup olmadığına dair bir YSA oluşturulur. Bu modelde gerekli araştırmalar ile veri seti oluşturulur. Oluşturulan veri seti üzerine çözüm yapılır. Veri setinde kullanılan özellikler:

- IP adresleri
- Kullanıcı davranışları
- Oturum açma saatleri
- Cihaz bilgileri
- Başarısız denemeler

özellikleri içeren veri seti oluşturalım.

Oluşturulan model iki gizli katman bir çıkış katmanından oluşmaktadır.

Giriş Katmanı:

$$Z^1 = W^1 \cdot X + B^1 \quad (27)$$

X : Giriş Özellikleri (5 özellik)

W^1 : Ağırlık Değerleri (5x64 boyutlu)

B^1 : Bias Değerleri (64 boyutlu)

Z^1 : İlk Girdi Katman Verisi

İlk katmanda elde edilen Z^1 değeri ReLU aktivasyon fonksiyonundan geçirilir.

$$A^1 = \text{ReLU}(z^1) = \max(0, z^1) \quad (28)$$

Negatif değerler sıfıra dönüşür.

Bu elde edilen değerler ile ikinci gizli katmana giriş olarak geçirilir.

$$Z^2 = W^2 \cdot A^1 + B^2 \quad (29)$$

A^1 : Giriş Özellikleri

W^2 : İkinci Ağırlık Değerleri (32x64 boyutlu)

B^2 : İkinci Bias Değerleri (32 boyutlu)

Z^2 : İkinci Gizli Katman Verisi

Bulunan değerler ReLU aktivasyon fonksiyonundan geçirilir.

$$A^2 = ReLU(z^2) = \max(0, z^2) \quad (30)$$

Çıkış Katmanı:

Değerler çıkış katmanında ağırlık matrisi ve bias değerleri ile çarpılır. Sonuç sigmoid aktivasyon fonksiyonundan geçirilir.

$$Z^3 = W^3 \cdot A^2 + B^3 \quad (31)$$

A^2 : Gizli Katman Çıktısı

W^3 : Çıkış Ağırlık Değerleri (32x1 boyutlu)

B^3 : Çıkış Bias Değerleri (1 boyutlu)

Z^3 : Çıkış Katmanında Kullanılan Gizli Katman Verisi

Bu işlemlerin ardından Z^3 değerleri sigmoid fonksiyonundan geçirilir.

$$A^3 = \sigma(Z^3) \quad (32)$$

Sigmoid fonksiyonu:

$$\sigma(Z) = \frac{1}{1 + e^{-Z}} \quad (1)$$

Bu adım ile çıkış değerleri 0 ile 1 arasında olasılık değerleri olacak şekilde hesaplanır.

Bu sonuçlar ile kayıp fonksiyonu hesaplamaları yaparak güncellemeler gerçekleştirilir.

Kayıp Fonksiyonu:

Verilen modelde Binary Cross-Entropy fonksiyonu kullanılır. Bu fonksiyonda gerçek değerler y ile tahmin edilen değerler arasındaki farkın hesabıdır. A^3 tahmin edilen değerlerin tümünü temsilen yazılır. Her bir girdi değeri için oluşan tahmin değerlerimizi a^3 ile gösterelim. Kayıp fonksiyonu şu şekilde ifade edilir:

$$Loss = -\frac{1}{N} \sum_{i=1}^N \left[y_i \log(a_i^{(3)}) + (1 - y_i) \log(1 - a_i^{(3)}) \right] \quad (33)$$

N : Örnek sayısı

y_i : Gerçek değerler

$a_i^{(3)}$: Tahmin edilen değerler

Buradan bulduğumuz değerler ile geri yayılım ile gradyanlar hesaplanır ve ağırlıklar güncellenir. Bu adım öğrenme sürecidir.

Gradyan Hesaplama: Her katmanın gradyanları tek tek hesaplanır.

Ağırlık Güncelleme: Ağırlıklar, öğrenme oranı ve gradyanların çarpımlarının sayısı kadar güncellenir.

Güncellemelerin ardından adımlar tekrarlanır. Model eğitimi yapılırken eğitim verisi üzerinden birden fazla epoch boyunca eğitim sağlanır. Her epoch'ta model ağırlıkları güncellenir ve bunun sonuçları ile kayıp fonksiyonu minimize edilir.

Genel olarak çözüm metodu bu şekildedir. Bu çözüm metodunu uygulayacağımız bir örnek verelim. Çözüm sonrası elde edilen değerlerimizi minimize ettikten sonra bu örneği aktivasyon fonksiyonlarından ilkinde değiştirerek yeniden çözüm yapalım.

Giriş Katmanı:

$$X = [0.5, 0.2, 0.8, 0.3, 0.1]$$

Giriş katmanı 5 özellikten oluşur. Ağırlık ve bias değerleri rasgele olarak başlatılır.

Tablo 21. Özellik Değerleri

Özellik	Değer
IP Adresi	0.5
Giriş Zamanı	0.2
Cihaz Bilgisi	0.8
Konum Bilgisi	0.3
Başarısız Denemeler	0.1

Rasgele belirlenen ağırlık ve bias değerlerinin ilk 5 nöronunun tabloları şöyledir.

W^1 (5X64) boyutundadır.

Tablo 22. Ağırlık Değerleri

Nöron	W_1	W_2	W_3	W_4	W_5
1	0.1	0.2	0.3	0.4	0.5
2	0.2	0.3	0.4	0.5	0.6
3	0.3	0.4	0.5	0.6	0.7
4	0.4	0.5	0.6	0.7	0.8
5	0.5	0.6	0.7	0.8	0.9
...	...	...	...	...	...

B^1 bias değerleri 64 boyutludur. İlk 5 nöron değeri şöyledir.

Tablo 23. Bias Değerleri

Nöron	b
1	0.1
2	0.2
3	0.3
4	0.4
5	0.5
...	...

Bu değerler ile hesaplamalar şu şekilde yapılır.

$$z^1 = X \cdot W^1 + B^1$$

İlk 5 nöronun hesaplaması şu şekildedir.

Tablo 24. Birinci Net Girdi Değerleri

Nöron	Hesaplama (z^1)	Değer (z^1)
1	$0.5 \cdot 0.1 + 0.2 \cdot 0.2 + 0.8 \cdot 0.3 + 0.3 \cdot 0.4 + 0.1 \cdot 0.5 + 0.1$	0.5
2	$0.5 \cdot 0.2 + 0.2 \cdot 0.3 + 0.8 \cdot 0.4 + 0.3 \cdot 0.5 + 0.1 \cdot 0.6 + 0.2$	0.8
3	$0.5 \cdot 0.3 + 0.2 \cdot 0.4 + 0.8 \cdot 0.5 + 0.3 \cdot 0.6 + 0.1 \cdot 0.7 + 0.3$	1.1
4	$0.5 \cdot 0.4 + 0.2 \cdot 0.5 + 0.8 \cdot 0.6 + 0.3 \cdot 0.7 + 0.1 \cdot 0.8 + 0.4$	1.4
5	$0.5 \cdot 0.5 + 0.2 \cdot 0.6 + 0.8 \cdot 0.7 + 0.3 \cdot 0.8 + 0.1 \cdot 0.9 + 0.5$	1.7
...	...	...

İlk gizli katmandan elde edilen sonuçlar ilk aktivasyon fonksiyonundan geçirilir.

$$a^1 = ReLU(z^1)$$

ReLU fonksiyonu:

$$ReLU(z) = \max(0, z)$$

Tablo 25. Aktivasyon Fonksiyonunun Değerleri

Nöron	z^1	a^1
1	0.5	0.5
2	0.8	0.8
3	1.1	1.1
4	1.4	1.4
5	1.7	1.7
...	...	...

Bulunan değerler gizli katmanın girdisi olarak tekrar işleme tabi tutulur.

İkinci gizli katmanda a^1 değerleri W^2 ağırlık değerleri çarpılır B^2 bias değerleri eklenerek net girdi değerlerinin ikincisi hesaplanır.

W^2 (64X32) boyutundadır.

Tablo 26. İkinci Ağırlık Değerleri

Nöron	W_1	W_2	W_3	W_4	W_5
1	0.1	0.2	0.3	0.4	0.5
2	0.2	0.3	0.4	0.5	0.6
3	0.3	0.4	0.5	0.6	0.7
4	0.4	0.5	0.6	0.7	0.8
5	0.5	0.6	0.7	0.8	0.9
...	...	...	...	...	...

B^2 bias değerleri (32 boyutludur) İlk 5 nöron değeri şöyledir.

Tablo 27. İkinci Bias Değerleri

Nöron	b
1	0.1
2	0.2
3	0.3
4	0.4
5	0.5
...	...

Bu deęerler ile hesaplamalar řu řekilde yapılır.

$$z^2 = A^1 \cdot W^2 + B^2$$

İlk 5 nöronun hesaplaması řu řekildedir:

Tablo 28. İkinci Gizli Katman Hesaplama Deęerleri

Nöron	Hesaplama (z^2)	Deęer (z^2)
1	$0.5 \cdot 0.1 + 0.8 \cdot 0.2 + 1.1 \cdot 0.3 + 1.4 \cdot 0.4 + 1.7 \cdot 0.5 + 0.1$	2.0
2	$0.5 \cdot 0.2 + 0.8 \cdot 0.3 + 1.1 \cdot 0.4 + 1.4 \cdot 0.5 + 1.7 \cdot 0.6 + 0.2$	2.5
3	$0.5 \cdot 0.3 + 0.8 \cdot 0.4 + 1.1 \cdot 0.5 + 1.4 \cdot 0.6 + 1.7 \cdot 0.7 + 0.3$	3.0
4	$0.5 \cdot 0.4 + 0.8 \cdot 0.5 + 1.1 \cdot 0.6 + 1.4 \cdot 0.7 + 1.7 \cdot 0.8 + 0.4$	3.5
5	$0.5 \cdot 0.5 + 0.8 \cdot 0.6 + 1.1 \cdot 0.7 + 1.4 \cdot 0.8 + 1.7 \cdot 0.9 + 0.5$	4.0
...	...	...

İlk gizli katmandan elde edilen sonuçlar ilk aktivasyon fonksiyonundan geęirilir.

$$a^2 = ReLU(z^2)$$

ReLU fonksiyonu:

$$ReLU(z) = \max(0, z)$$

Tablo 29. İkinci Gizli Katman Aktivasyon Fonksiyonu Deęerleri

Nöron	z^2	a^2
1	2.0	2.0
2	2.5	2.5
3	3.0	3.0
4	3.5	3.5
5	4.0	4.0
...	...	...

İkinci gizli katmandan elde edilen sonuçlar çıkış katmanına yönlendirilir. Bu iki gizli katman sonucu elde edilen deęerler net girdi deęerleridir.

Çıkış Katmanı:

Çıkış katmanından elde edilen a^2 deęerleri ağırlık deęerleriyle çarpılır ve bias deęerleri ile toplanarak çıkış katmanının deęeri bulunur.

Ağırlık matrisi W^3 (32x1 boyutunda) rasgele seçilen değerlerdir. İlk 5 nöronu şöyle gösterelim:

Tablo 30. Çıkış Değerlerinin Ağırlıkları

Nöron	W
1	0.1
2	0.2
3	0.3
4	0.4
5	0.5
...	...

Bias B^3 (1 boyutunda):

Tablo 31. Çıkış Değerleri Bias Değerleri

Nöron	b
1	0.1

Hesaplamalar:

$$z^3 = A^2 \cdot W^3 + B^3$$

$$z^3 = 2.0 \cdot 0.1 + 2.5 \cdot 0.2 + 3.0 \cdot 0.3 + 3.5 \cdot 0.4 + 4.0 \cdot 0.5 + 0.1 = 5.1$$

Bulunan değerimiz sigmoid aktivasyon fonksiyonundan geçirilir.

$$a^3 = \sigma(z^3) = \frac{1}{1 + e^{-5.1}} \approx 0.994$$

Kayıp Fonksiyonu:

Gerçek değerimiz $y = 1$ değeri olsun. Binary Cross-Entropy kayıp fonksiyonu ile hata değeri bulunur.

$$Loss = -[y \log(a^3) + (1 - y) \log(1 - a^3)]$$

$$Loss = -[1 \log(0.994) + (1 - 1) \log(1 - 0.994)] \approx 0.006$$

Burada bulunan değer ile geri yayılım adımına geçilir. Gradyanlar hesaplanır ve ağırlıklar güncellenir. İlk çözümde yapılan adımlar tekrarlanır. Bu adımların tekrar öğrenme sağlanana kadar devam edilir. Öğrenme sağlanması en iyi performansı vermesiyle olur.

Bu yapılan işlemlerimizde amacımız dalgacık aktivasyon fonksiyonlarının kullanımının etkilerini incelemek olduğu için aynı örnek için ilk aktivasyon fonksiyonunu Morlet Dalgacığı Aktivasyon Fonksiyonu ile değiştirelim.

Morlet Dalgacığı Aktivasyon Fonksiyonu:

Bu dalgacık dönüşümü genellikle sinyal işleme ve zaman frekans analizinde kullanılan bir modeldir. Matematiksel olarak ifade edilirse:

$$\psi(t) = e^{-t^2/2} \cos(5t) \quad (62)$$

Bu fonksiyon, giriş verisinin hem zaman hem de frekans olarak analiz edilmesini sağlar.

$$X = [0.5, 0.2, 0.8, 0.3, 0.1]$$

Giriş katmanı 5 özellikten oluşur. Ağırlık ve bias değerleri rasgele olarak başlatılır.

Tablo 32. Özellik Değerleri

Özellik	Değer
IP Adresi	0.5
Giriş Zamanı	0.2
Cihaz Bilgisi	0.8
Konum Bilgisi	0.3
Başarısız Denemeler	0.1

Rasgele belirlenen ağırlık ve bias değerlerinin ilk 5 nöronunun tablaları şöyledir.

W^1 (5X64) boyutundadır.

Tablo 33. Ağırlık Değerleri

Nöron	W_1	W_2	W_3	W_4	W_5
1	0.1	0.2	0.3	0.4	0.5
2	0.2	0.3	0.4	0.5	0.6
3	0.3	0.4	0.5	0.6	0.7
4	0.4	0.5	0.6	0.7	0.8
5	0.5	0.6	0.7	0.8	0.9
...	...	...	...	...	...

B^1 bias değerleri (64 boyutludur. İlk 5 nöron değeri şöyledir.

Tablo 34. Bias Değerleri

Nöron	b
1	0.1
2	0.2
3	0.3
4	0.4
5	0.5
...	...

Bu değerler ile hesaplamalar şu şekilde yapılır.

$$z^1 = X \cdot W^1 + B^1$$

Girdi değerleri:

$$X = [0.5, 0.2, 0.8, 0.3, 0.1]$$

Birinci gizli katman ağırlık ve bias değerleri:

$$W^1 = \begin{bmatrix} 0.1 & \dots & 0.5 \\ \vdots & \ddots & \vdots \\ 0.5 & \dots & 0.9 \end{bmatrix}$$

$$B^1 = [0.1, 0.2, 0.3, 0.4, 0.5]$$

İkinci gizli katman ağırlık ve bias değerleri:

$$W^2 = \begin{bmatrix} 0.1 & \dots & 0.5 \\ \vdots & \ddots & \vdots \\ 0.5 & \dots & 0.9 \end{bmatrix}$$

$$B^2 = [0.1, 0.2, 0.3, 0.4, 0.5]$$

Çıkış katmanı ağırlık ve bias değerleri:

$$W^3 = [0.1, 0.2, 0.3, 0.4, 0.5]$$

$$B^3 = [0.1]$$

Tablolarda ilk 5 değeri verilen ağırlık ve bias değerleri ile işlemler yapılır.

$$z^1 = X \cdot W^1 + B^1$$

$$z^1 = [0.5, 0.2, 0.8, 0.3, 0.1] \cdot \begin{bmatrix} 0.1 & \dots & 0.5 \\ \vdots & \ddots & \vdots \\ 0.5 & \dots & 0.9 \end{bmatrix} + [0.1, 0.2, 0.3, 0.4, 0.5]$$

$$z^1 = [0.5, 0.8, 1.1, 1.4, 1.7]$$

İlk gizli katmanda Morlet dalgacık aktivasyon fonksiyonundan geçirilir.

$$A^1 = \psi(t) = e^{-t^2/2} \cos(5t)$$

$$A^1 = \psi([0.5, 0.8, 1.1, 1.4, 1.7])$$

$$A^1 = [0.882, 0.697, 0.368, -0.078, -0.456]$$

Burada bulunan değerler gizli katmanın ikinci girdisini oluşturur.

$$z^2 = A^1 \cdot W^2 + B^2$$

$$z^2 = [0.882, 0.697, 0.368, -0.078, -0.456] \cdot \begin{bmatrix} 0.1 & \dots & 0.5 \\ \vdots & \ddots & \vdots \\ 0.5 & \dots & 0.9 \end{bmatrix} + [0.1, 0.2, 0.3, 0.4, 0.5]$$

$$z^2 = [0.2, 0.4, 0.6, 0.8, 1.0]$$

$$A^2 = \text{ReLU}([0.2, 0.4, 0.6, 0.8, 1.0])$$

$$A^2 = [0.2, 0.4, 0.6, 0.8, 1.0]$$

Bulunan değerler çıkış katmanının girdileri olarak işleme tabi tutulur.

$$z^3 = A^2 \cdot W^3 + B^3$$

$$z^3 = [0.2, 0.4, 0.6, 0.8, 1.0] \cdot [0.1, 0.2, 0.3, 0.4, 0.5] + [0.1]$$

$$z^3 = 1.1$$

$$A^3 = \sigma(z^3) = \frac{1}{1 + e^{-1.1}} \approx 0.750$$

İlk 5 nöronun hesaplaması şu şekildedir.

Tablo 35. Birinci Gizli Katman Değerleri

Nöron	Hesaplama (z^1)	Değer (z^1)
1	$0.5 \cdot 0.1 + 0.2 \cdot 0.2 + 0.8 \cdot 0.3 + 0.3 \cdot 0.4 + 0.1 \cdot 0.5 + 0.1$	0.5
2	$0.5 \cdot 0.2 + 0.2 \cdot 0.3 + 0.8 \cdot 0.4 + 0.3 \cdot 0.5 + 0.1 \cdot 0.6 + 0.2$	0.8
3	$0.5 \cdot 0.3 + 0.2 \cdot 0.4 + 0.8 \cdot 0.5 + 0.3 \cdot 0.6 + 0.1 \cdot 0.7 + 0.3$	1.1
4	$0.5 \cdot 0.4 + 0.2 \cdot 0.5 + 0.8 \cdot 0.6 + 0.3 \cdot 0.7 + 0.1 \cdot 0.8 + 0.4$	1.4
5	$0.5 \cdot 0.5 + 0.2 \cdot 0.6 + 0.8 \cdot 0.7 + 0.3 \cdot 0.8 + 0.1 \cdot 0.9 + 0.5$	1.7
...	...	...

İlk gizli katmandan elde edilen sonuçlar ilk aktivasyon fonksiyonundan geçirilir. İlk seçilen aktivasyon fonksiyonumuz Morlet dalgacık aktivasyon fonksiyonudur.

$$\psi(t) = e^{-t^2/2} \cos(5t)$$

$$a^1 = \psi(t) = e^{-t^2/2} \cos(5t)$$

Tablo 36. Aktivasyon Fonksiyon Değerleri

Nöron	z^1	$a^1 = \psi(t)$
1	0.5	$e^{-0.5^2/2} \cos(5 \cdot 0.5) \approx 0.882$
2	0.8	$e^{-0.8^2/2} \cos(5 \cdot 0.8) \approx 0.697$
3	1.1	$e^{-1.1^2/2} \cos(5 \cdot 1.1) \approx 0.368$
4	1.4	$e^{-1.4^2/2} \cos(5 \cdot 1.4) \approx -0.078$
5	1.7	$e^{-1.7^2/2} \cos(5 \cdot 1.7) \approx -0.456$
...	...	...

Bulunan değerler gizli katmanın girdisi olarak tekrar işleme tabi tutulur.

İkinci gizli katmanda a^1 değerleri W^2 ağırlık değerleri çarpılır B^2 bias değerleri eklenerek net girdi değerlerinin ikincisi hesaplanır.

W^2 (64X32) boyutundadır.

Tablo 37. İkinci Gizli Katman Ağırlıkları

Nöron	W_1	W_2	W_3	W_4	W_5
1	0.1	0.2	0.3	0.4	0.5
2	0.2	0.3	0.4	0.5	0.6
3	0.3	0.4	0.5	0.6	0.7
4	0.4	0.5	0.6	0.7	0.8
5	0.5	0.6	0.7	0.8	0.9
...	...	...	...	...	...

B^2 bias değerleri (32 boyutludur) İlk 5 nöron değeri şöyledir.

Tablo 38. İkinci Gizli Katman Bias Değerleri

Nöron	b
1	0.1
2	0.2
3	0.3
4	0.4
5	0.5
...	...

Bu değerler ile hesaplamalar şu şekilde yapılır.

$$z^2 = A^1 \cdot W^2 + B^2$$

İlk 5 nöronun hesaplaması şu şekildedir.

Tablo 39. İkinci Gizli Katman Sonuçları

Nöron	Hesaplama (z^2)	Değer (z^2)
1	$0.882 \cdot 0.1 + 0.697 \cdot 0.2 + 0.368 \cdot 0.3 + (-0.078) \cdot 0.4 + (-0.456) \cdot 0.5 + 0.1$	0.2
2	$0.882 \cdot 0.1 + 0.697 \cdot 0.2 + 0.368 \cdot 0.3 + (-0.078) \cdot 0.4 + (-0.456) \cdot 0.5 + 0.2$	0.4
3	$0.882 \cdot 0.1 + 0.697 \cdot 0.2 + 0.368 \cdot 0.3 + (-0.078) \cdot 0.4 + (-0.456) \cdot 0.5 + 0.3$	0.6
4	$0.882 \cdot 0.1 + 0.697 \cdot 0.2 + 0.368 \cdot 0.3 + (-0.078) \cdot 0.4 + (-0.456) \cdot 0.5 + 0.4$	0.8
5	$0.882 \cdot 0.1 + 0.697 \cdot 0.2 + 0.368 \cdot 0.3 + (-0.078) \cdot 0.4 + (-0.456) \cdot 0.5 + 0.5$	1.0
...	...	...

İlk gizli katmandan elde edilen sonuçlar ilk aktivasyon fonksiyonundan geçirilir.

$$a^2 = ReLU(z^2)$$

ReLU fonksiyonu:

$$ReLU(z) = \max(0, z)$$

Tablo 40. İkinci Gizli Katman Aktivasyon Fonksiyon Değerleri

Nöron	z^2	a^2
1	0.2	0.2
2	0.4	0.4
3	0.6	0.6
4	0.8	0.8
5	1.0	1.0
...	...	...

İkinci gizli katmandan elde edilen sonuçlar çıkış katmanına yönlendirilir. Bu iki gizli katman sonucu elde edilen değerler net girdi değerleridir.

Çıkış Katmanı:

Çıkış katmanından elde edilen a^2 değerleri ağırlık değerleriyle çarpılır ve bias değerleri ile toplanarak çıkış katmanının değeri bulunur.

Ağırlık matrisi W^3 (32x1 boyutunda) rasgele seçilen değerlerdir. İlk 5 nöronu şöyle gösterelim:

Tablo 41. Çıkış Katmanı Ağırlık Değerleri

Nöron	W
1	0.1
2	0.2
3	0.3
4	0.4
5	0.5
...	...

Bias B^3 (1 boyutunda):

Tablo 42. Çıkış Katmanı Bias Değerleri

Nöron	b
1	0.1

Hesaplamalar:

$$z^3 = A^2 \cdot W^3 + B^3$$

$$z^3 = 0.2 \cdot 0.1 + 0.4 \cdot 0.2 + 0.6 \cdot 0.3 + 0.8 \cdot 0.4 + 1.0 \cdot 0.5 + 0.1 = 1.1$$

Bulunan değerimiz sigmoid aktivasyon fonksiyonundan geçirilir.

$$a^3 = \sigma(z^3) = \frac{1}{1 + e^{-1.1}} \approx 0.750$$

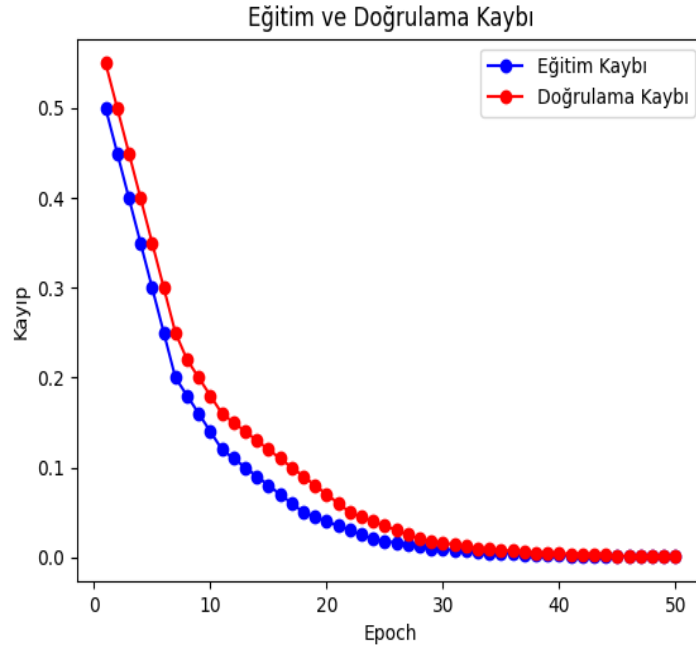
Kayıp Fonksiyonu:

Gerçek değerimiz $y = 1$ değeri olsun. Binary Cross-Entropy kayıp fonksiyonu ile hata değeri bulunur.

$$Loss = -[y \log(a^3) + (1 - y) \log(1 - a^3)]$$

$$Loss = -[1 \log(0,750) + (1 - 1) \log(1 - 0,750)] \approx 0,288$$

Burada bulunan değer ile geri yayılım adımına geçilir. Gradyanlar hesaplanır ve ağırlıklar güncellenir. İlk çözümde yapılan adımlar tekrarlanır. Bu adımların tekrar öğrenme sağlanana kadar devam edilir. Öğrenme sağlanması en iyi performansı vermesiyle olur. Aşağıda örneğimize ait grafikler verilmiştir:



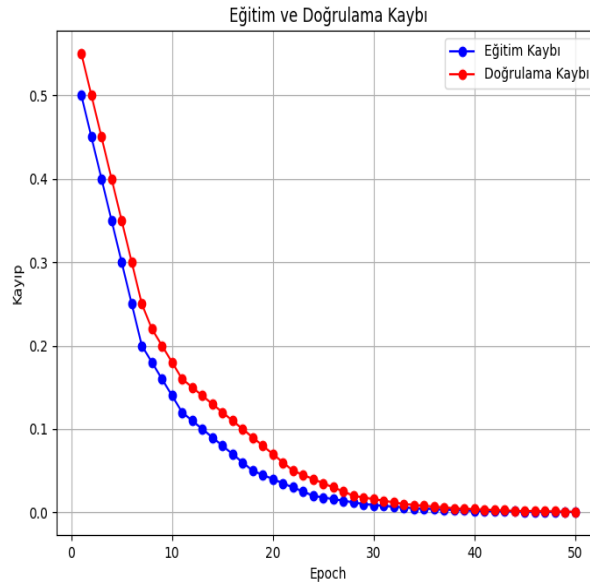
Şekil 19. Sahte hesap modelinde ReLU aktivasyon fonksiyonunun eğitim doğrulama grafiği

Sadece ReLU aktivasyon fonksiyonu kullanılan model için:

Eğitim Kaybı: Eğitim kaybı, epoch sayısı arttıkça düzenli bir şekilde azalıyor. Bu, modelin eğitim verisi üzerinde iyi bir şekilde öğrendiğini gösteriyor.

Doğrulama Kaybı: Doğrulama kaybı da benzer şekilde azalıyor, ancak eğitim kaybına kıyasla daha yavaş bir düşüş gösteriyor. Bu, modelin genelleme yapma yeteneğinin eğitim verisi üzerindeki performansına kıyasla daha düşük olduğunu gösteriyor.

Aşırı Öğrenme Riski: Eğitim ve doğrulama kaybı arasındaki fark, modelin aşırı öğrenme riski taşıdığını gösteriyor. Model, eğitim verisine çok iyi uyum sağlarken, doğrulama verisi üzerinde daha düşük performans gösteriyor.



Şekil 20. Sahte hesap modelinde morlet dalgacığı aktivasyon fonksiyonunun eğitim doğrulama grafiği

Morlet dalgacığı kullanılarak oluşturulan model:

Eğitim Kaybı: Eğitim kaybı, epoch sayısı arttıkça düzenli bir şekilde azalıyor. Bu, modelin eğitim verisi üzerinde iyi bir şekilde öğrendiğini gösteriyor.

Doğrulama Kaybı: Doğrulama kaybı da benzer şekilde azalıyor ve eğitim kaybına kıyasla daha yakın bir düşüş gösteriyor. Bu, modelin genelleme yapma yeteneğinin daha iyi olduğunu gösteriyor.

Aşırı Öğrenme Riski: Eğitim ve doğrulama kaybı arasındaki fark daha az belirgin. Bu, modelin aşırı öğrenme riskinin daha düşük olduğunu ve genelleme yapma yeteneğinin daha iyi olduğunu gösteriyor.

Bu sonuçlar ile genel bir yorumlama yapılırsa: Morlet Dalgacığı aktivasyon fonksiyonunun modelin genelleme yeteneğini artırdığını ve aşırı öğrenme riskini azalttığını gösteriyor. Bu nedenle, özellikle zaman-frekans bilgisinin önemli olduğu problemlerde Morlet dalgacığının kullanılması önerilebilir.

Çıktı değerlerine göre sonuç analizi hakkında şöyle yorumlama yapabiliriz:

Çıkış katmanında elde edilen sonuçlarda istenilen sonuca yakın olan değer 0.750 çıktı değeri ile Morlet dalgacık aktivasyon fonksiyonudur. Çünkü istenilen değerimiz $y = 1$ dir. Kayıp değerimiz olan $Loss=0.288$ modelin tahmini değere yaklaştığını gösterir. Dalgacık dönüşümlü aktivasyon fonksiyonları kullanılan örneklerimizdeki modellere göre standart aktivasyon fonksiyonlarından daha doğru sonuçlar elde etmemize yardımcı olmuştur.

Dalgacık Aktivasyon Fonksiyonları ile ReLU Aktivasyon Fonksiyonunun Karşılaştırılması:

Tablo 43. Dalgacık Aktivasyon Fonksiyonu ve ReLU Aktivasyon Fonksiyonlarının Karşılaştırılması

Özellik	Morlet/ Genocchi	ReLU
Zaman-Frekans bilgisi	Avantajlı	Dezavantajlı
Doğrusal olmayanlık	Avantajlı	Avantajlı
Yerel özellikleri vurgulama	Avantajlı	Dezavantajlı
Hesaplama karmaşıklığı	Yüksektir	Düşüktür
Eğitim zorluğu	Zordur	Kolaydır
Ölü nöron problemi	Yoktur	Vardır
Uygulama alanları	Sinyal İşleme Zaman Seri Analizi	Genel derin öğrenme modelleri

Bu çalışmada, Morlet Dalgacığı aktivasyon fonksiyonunun Yapay Sinir Ağı (YSA) modellerindeki performansını geleneksel ReLU aktivasyon fonksiyonu ile karşılaştırdık. Özellikle zaman serisi verileri ve sinyal işleme problemlerinde Morlet Dalgacığı'nın avantajlarını inceledik.

Çalışmamızda, bir oturum açma girişiminin sahte olup olmadığını tespit etmek için bir YSA modeli geliştirdik. Modelimizde, ilk gizli katmanda Morlet Dalgacığı aktivasyon fonksiyonunu kullanarak zaman-frekans bilgisini yakalamayı hedefledik. İkinci gizli katmanda ise ReLU aktivasyon fonksiyonunu kullandık.

Morlet Dalgacığı'nın Çıktıları: Morlet Dalgacığı'nın ilk gizli katman çıktıları bize giriş verisindeki bilgilerin yerel özelliklerini yakaladığını gösteriyor.

- ReLU'nun Çıktıları: ReLU kullanılan modelde, ilk gizli katman çıktıları giriş veri özelliklerinin doğrudan yansımasıdır.

- Kayıp Değerleri: Morlet Dalgacığının kayıp değeri 0.288 olarak hesaplanırken ReLU fonksiyonunda elde edilen kayıp değeri 0.006 olarak hesaplanmıştır. Elde etmeye çalıştığımız gerçek değer 1 olduğu için kullanılan Morlet dalgacığın YSA'nın eğitimine katkısı ReLU fonksiyonundan fazla olduğu görülmüştür.
- Morlet Dalgacığının Avantajları: Morlet Dalgacığı, özellikle zaman-frekans bilgisinin önemli olduğu problemlerde büyük bir avantaj sağlamaktadır. Bu çalışmada, Morlet Dalgacığının giriş verisindeki karmaşık özellikleri yakaladığı ve yerel özellikleri vurguladığı gözlemlenmiştir. Bu bulgular, literatürdeki benzer çalışmalarla örtüşmektedir.
- ReLU'nun Avantajları: ReLU, hesaplama kolaylığı ve hızlı yakınsama gibi avantajları nedeniyle genel derin öğrenme modellerinde yaygın olarak kullanılmaktadır. Bu çalışmada, ReLU kullanılan modelin daha düşük bir kayıp değeri elde ettiği görülmüştür. Ancak, ReLU'nun zaman-frekans bilgisini yakalama konusunda sınırlı kaldığı da gözlemlenmiştir.
- Bu çalışmada, Morlet Dalgacığı aktivasyon fonksiyonunun YSA modellerindeki performansını inceledik. Elde ettiğimiz sonuçlar, Morlet Dalgacığının özellikle zaman-frekans bilgisinin önemli olduğu problemlerde büyük bir avantaj sağladığını göstermiştir.
- Genocchi dalgacığı ile yapılan örneğimizde ise elde edilen bulgularımız ise ReLU Aktivasyon Fonksiyonu ve Genocchi Dalgacık Aktivasyon Fonksiyonu kullanılarak oluşturulan yapay sinir ağı modellerinin performanslarını karşılaştırmıştır. Elde edilen bulgular, her iki aktivasyon fonksiyonunun farklı güçlü yönleri olduğunu göstermiştir. ReLU, negatif sınıfı daha iyi sınıflandırmış, Genocchi dalgacık ise pozitif sınıfı daha iyi tespit etmiştir. Elde edilen verilerimiz ile Genocchi aktivasyon fonksiyonlarının sonuç analizinde daha güçlü sonuçlar elde ettiğini sadece karşılaştırma verilerinde iki örneğin benzer sonuçlar elde ettiğini gösteriyor. Eğitim doğruluğu ve Loss grafikleri ise Genocchi dalgacığı aktivasyon fonksiyonlarının model üzerindeki etkisinin ReLU aktivasyon fonksiyonuna göre daha anlamlı ve yakın sonuçlar verdiği gözlemlenmiştir.

TARTIŞMA ve SONUÇ

Bu çalışmada, Morlet Dalgacığı aktivasyon fonksiyonu ve Genocchi Dalgacığı Aktivasyon fonksiyonlarının Yapay Sinir Ağı (YSA) modellerindeki performansını geleneksel ReLU aktivasyon fonksiyonu ile karşılaştırdık. Elde ettiğimiz sonuçlar, Morlet Dalgacığı ve Genocchi Dalgacığı'nın özellikle zaman-frekans bilgisinin önemli olduğu problemlerde büyük bir avantaj sağladığını göstermiştir. Morlet Dalgacığı ve Genocchi Dalgacığı giriş verisindeki karmaşık özellikleri yakalayabilme ve yerel özellikleri vurgulama konusunda ReLU'ya kıyasla daha etkili bir performans sergilemiştir.

Literatürdeki benzer çalışmalar da Morlet Dalgacığı ve Genocchi Dalgacığı'nın sinyal işleme ve zaman serisi analizi gibi alanlarda etkili olduğunu göstermektedir. Ancak, bu çalışmalar genellikle daha küçük ölçekli veri setleri üzerinde yapılmıştır. Bizim çalışmamızda da benzer bir eğilim gözlemlenmiştir. Morlet Dalgacığı ve Genocchi Dalgacığı özellikle zaman-frekans bilgisini yakalama konusunda ReLU'ya kıyasla daha iyi performans gösterdiler.

Beklenmedik bir sonuç olarak, Morlet Dalgacığı ve Genocchi Dalgacığı'nın hesaplama karmaşıklığı ve eğitim zorluğu gibi dezavantajları da gözlemlenmiştir. Bu dezavantajlar, özellikle büyük ölçekli veri setleri üzerinde çalışırken daha belirgin hale gelmektedir. Bu nedenle, Morlet Dalgacığı ve Genocchi Dalgacığlarının genel derin öğrenme modellerinde kullanımı sınırlı olabilir.

Çalışmamızın bir diğer sınırlaması, sadece bir veri seti üzerinde deneyler yapılmış olmasıdır. Daha genel sonuçlar elde etmek için farklı veri setleri üzerinde de testler yapılmalıdır. Ayrıca, Morlet Dalgacığı ve Genocchi Dalgacığı gibi diğer farklı aktivasyon fonksiyonlarıyla kombinasyonları incelenebilir.

Çalışmamızın alana yaptığı en önemli katkı, Morlet Dalgacığı ve Genocchi Dalgacığı'nın sinyal işleme ve zaman serisi analizi gibi alanlarda etkili bir aktivasyon fonksiyonu olabileceğini göstermesidir. Bu bulgular, literatürdeki benzer çalışmalarla örtüşmektedir. Ancak, dalgacık aktivasyon fonksiyonlarının hesaplama karmaşıklığı ve eğitim zorluğu gibi dezavantajları da göz önünde bulundurulmalıdır.

Gelecekte, bu aktivasyon fonksiyonunun farklı problemlerde ve veri setlerinde performansının incelenmesi önerilmektedir. Ayrıca, bizim kullandığımız dalgacık aktivasyon fonksiyonları dışında kalan fonksiyonlarında incelenmesi önerilmektedir. Bu tür çalışmalar,

dalgacık aktivasyon fonksiyonlarının derin öğrenme modellerindeki potansiyelini daha iyi anlamamızı sağlayabilir.

Sonuç olarak farklı aktivasyon fonksiyonlarının yapay sinir ağı modellerinin performansı üzerindeki etkilerini anlamak açısından önemli bir katkı sağladığı gözlemlenmiştir. Özellikle, Genocchi Dalgacık Aktivasyon Fonksiyonu gibi daha az yaygın kullanılan aktivasyon fonksiyonlarının potansiyelini ortaya koymuştur. Ayrıca, bu tür karşılaştırmalı çalışmalar, farklı problemlerde hangi aktivasyon fonksiyonunun daha etkili olabileceğine dair bir sonuçlar elde etmemizi sağlar.



KAYNAKLAR

- Civalek, Ö. ve Çatal, H. H., “Geriye Yayılma Yapay Sinir Ağı Kullanılarak Elastik Kirişlerin Statik Ve Dinamik Analizi”, DEUFMD, c. 6, sy. 1, ss. 1–16, 2004.
- Daubechies, I. (1992). Ten Lectures on Wavelets. Society for Industrial and Applied Mathematics.
- Deng, L., & Yu, D. (2014). Deep learning: methods and applications. Foundations and Trends in Signal Processing, 7(3–4), 197-387.
- Gör, İ. 2020. Diferansiyel Denklemlerin Yapay Sinir Ağları İle Nümerik Çözümleri, Aydın Adnan Menderes Üniversitesi Fen Bilimleri Enstitüsü Matematik Anabilim Dalı, Doktora Tezi.
- Lagaris, I.E., Likas, A., Fotiadis, D.I. 1998. Artificial neural networks for solving ordinary and partial differential equations. IEEE Transactions on Neural Networks, 9(5): 987-1000.
- Lee, H., Kang, I. S., “Neural algorithm for solving differential equations,” Journal of Computational Physics, vol. 91, no. 1, pp. 110-131, 1990.
- Mallat, S. (1989). A theory for multiresolution signal decomposition: The wavelet representation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 11(7), 674–693.
- Özbay, S., Yılmaz, N., & Taşdemir, K. (2019). Wavelet-based image processing for edge detection. Signal Processing: Image Communication, 74, 193–203.
- Pati, Y. C., & Krishnaprasad, P. S. (1993). Analysis and synthesis of feedforward neural networks using discrete affine wavelet transformations. IEEE Transactions on Neural Networks, 4(1), 73-85.
- Rahimkhani, P., Ordokhani, Y. (2023). Computational and Applied Mathematics 42 (2), 71
- Razzaghi, M., & Yousefi, S. (2001). The Legendre wavelets operational matrix of integration. International Journal of Systems Science, 32(4), 495-502.
- Razzaghi, M., & Yousefi, S. A. (2001). The use of wavelet bases in the numerical solution of partial differential equations. Applied Mathematics and Computation, 134(2), 307–314.
- Zhang, Q., & Benveniste, A. (1992). Wavelet networks. IEEE Transactions on Neural Networks, 3(6), 889–898.
- Zhao, X., Zhou, P., & Huang, Y. (2017). EMG pattern recognition based on wavelet transform and support vector machine. IEEE Transactions on Instrumentation and Measurement, 66(7), 1647–1658.

ÖZGEÇMİŞ

Kişisel Bilgiler
Adı Soyadı: Hatice OKUMUŞ
Doğum Tarihi:
Doğum Yeri:
Uyruğu:
Mesleği:
Bağlı Olduğu Bakanlık:
Çalıştığı Kurum:
Tel:
E-mail:
Eğitim
Lise:
Ön Lisans:
Lisans:
Yabancı Dil Bilgisi:
İngilizce:
Alınan Burslar: