

TuringEQ: Doğrusal Olmayan Problemler Özelinde Yeni bir Yapay Zekâ Mimarisi

Hüseyin Enes OKUTAN^{1*}, Muhammet BAYKARA²

¹ Yazılım Mühendisliği Bölümü, Mühendislik Fakültesi, İnönü Üniversitesi, Malatya, Türkiye

² Yazılım Mühendisliği Bölümü, Teknoloji Fakültesi, Fırat Üniversitesi, Elazığ, Türkiye

*¹ enes.okutan@inonu.edu.tr, ² mbykara@firat.edu.tr

(Geliş/Received: 05/06/2024;

Kabul/Accepted: 20/09/2024)

Öz: Yapay Zekâ alanındaki gelişmelerle birlikte birçok alanda Yapay Zekâ kullanımı yaygınlaşmış ve bu teknolojinin kullanımı ile önemli başarımlar elde edilmiştir. Elde edilen başarımların büyük bir kısmı, Yapay Zekâ içerisindeki alt alanlardan biri olan Yapay Sinir Ağlarına dayanmaktadır. Bu çalışma kapsamında, Yapay Sinir Ağlarının başarımlarını yükseltecek yeni bir Yapay Zekâ Mimarisi üzerinde durulmaktadır. Bu çalışmada, doğrusal ve doğrusal olmayan problemler özelinde incelemeler yapılarak bu problemlerin karakteristiğinin çözüm üzerindeki etkileri incelenmiş ve bu incelemeler sonucunda temel bir doğrusal olmayan problem üzerinde, başarılı bir şekilde çalışacak yeni bir Yapay Zekâ Mimarisi geliştirilmesi amaçlanmıştır. Yapılan çalışmada öncelikle yeni Yapay Zekâ Mimarisi için temel bir problem belirlenmiştir. Temel problemin belirlenmesinin ardından, Yapay Zekâ Mimarisi ile ilgili çalışmalara başlanarak teorik temeller üzerinde mimari tasarım yapılmıştır. Mimari kapsamında, problem katmanlara bölünerek, her katmanın kendi içerisinde düzgün bir şekilde öğrenme işlemini gerçekleştirmesi amaçlanmıştır. Başarımların olumsuz olarak etki eden katmanlar içerisindeki ağırlıkların, katman bozukluğunu giderecek şekilde değiştirilmesi ile katman düzeltimi yapılmıştır. Bu katman düzeltimi ile mimarinin temeli olarak her katmanın genel mimari başarımlarını arttıracak şekilde kendi hatasını düzeltmesi ile öğrenme işleminin hatasız bir şekilde gerçekleştirilmesi amaçlanmıştır. Bütün bu teorik temeller üzerine tasarlanan Yapay Zekâ Mimarisinin, temel bir doğrusal olmayan problem üzerinde test edilmesi sonucunda mimarinin başarılı bir şekilde öğrenme işlemini gerçekleştirdiği görülmüştür.

Anahtar kelimeler: Yapay Zekâ, Yapay Öğrenme, Doğrusal Olmayan Problem.

TuringEQ: A New Artificial Intelligence Architecture for Nonlinear Problems

Abstract: With the developments in the field of Artificial Intelligence, the use of Artificial Intelligence has become widespread in many areas and important achievements have been achieved with the use of this technology. Most of the achievements are based on Artificial Neural Networks, which is one of the sub-fields in Artificial Intelligence. In this study, a new Artificial Intelligence Architecture that can achieve the performance level of Artificial Neural Networks is emphasized.

In this study, the effects of the characteristics of these problems on the solution were examined by examining the linear and nonlinear problems, and because of these examinations, it was aimed to develop a new Artificial Intelligence Architecture that would work successfully on a basic nonlinear problem. In the study, first, a basic problem for the new Artificial Intelligence Architecture was determined. After the determination of the basic problem, studies on Artificial Intelligence architecture were started and architectural design was made on theoretical foundations. Within the scope of architecture, it is aimed to divide the problem into layers and each layer to perform the learning process within itself properly. Layer correction was made by changing the weights in the layers, which had a negative effect on the performance, in a way that would eliminate the layer defect. With these layer corrections, as the basis of the architecture, it is aimed to perform the learning process without error by correcting its own error in a way that increases the overall architectural performance of each layer. As a result of testing the Artificial Intelligence architecture, which was designed on all these theoretical foundations, on a basic nonlinear problem, it was seen that the architecture successfully performed the learning process.

Key words: Artificial Intelligence, Machine Learning, Nonlinear Problem.

1. Giriş

Teknolojinin gelişmesi ve mevcut problemler üzerine uygulanmasıyla birçok probleme etkin ve kalıcı çözümler üretilmeye başlanmıştır. İnsanlar açısından, zamandan ve maliyetten tasarruf sağlamak amacıyla teknolojik gelişmeler doğrudan kullanım alanları bularak, insanların problemlerine çözümler üretilmiştir. Bu kapsamda, kullanım alanı olarak bilimsel problemlerden günlük hayatta karşılaşılan problemlere kadar birçok alanda aktif teknolojik çözümler kullanılmaya başlanmıştır.

İlk zamanlarda temel anlamda komutlar ile çözülebilen problemlerin karmaşık bir yapıya dönüşmesi sonucunda teknolojik gelişmeler bir gereklilik haline gelmiştir. Teknolojinin gelişiminin kaçınılmaz olduğu bu

* Sorumlu yazar: enes.okutan@inonu.edu.tr. Yazarların ORCID Numarası: ¹ 0000-0002-7455-2786, ² 0000-0001-5223-1343

ilerlemeler sonucunda Yapay Zekâ konusu ortaya çıkmış ve insanların problemlerine en az insanlar kadar etkin çözümler üretebilecek teknolojiler üzerinde durulmuştur.

Makinelerin en az insanlar kadar zeki olabileceği fikri 1950 yılında Mind adlı felsefe dergisinde ünlü bir İngiliz matematikçi ve bilgisayar bilimcisi olan Alan Turing tarafından Turing Testi olarak ortaya atılmıştır [1]. Alan Turing tarafından ortaya atılan Turing Testi, temel olarak belirli üç oda üzerinde durmaktadır. Bu odalardan ilki, içerisinde sorular sormaya hazır olan sorgulayıcı bir insanın olduğu bir odadır. Bu odalardan ikinci oda sorulara cevap vermeyi bekleyen cevap verici bir insanın yer aldığı bir odadır. Bu odalardan sonuncu oda ise içerisinde sorular sorulara cevap verebilecek bir bilgisayarın bulunduğu odadır. Turing testine göre, ilk odada bulunan sorgulayıcı insan, sorulara cevapların geleceği odalardan hangisinde bilgisayar hangisinde insan olduğunu bilmemektedir. Bu kapsamda Turing testi, sorgulayıcı olan insanın yönelttiği sorulara verilen cevaplardan cevabı veren odada bir bilgisayar mı yoksa bir insan mı olduğunun tahmin edilememesine dayanmaktadır. Eğer sorgulayıcı insan, bilgisayar ve insan tahminini doğru bir şekilde yapamaz ise bu durum Turing testinin başarılı olduğu anlamına gelmekte ve makinelerin düşünebileceği kanıtlanmış olmaktadır.

Alan Turing tarafından, makinelerin düşünebilmesine yönelik ilk fikirlerin ortaya atılmasının ardından gelen teknolojik gelişmeler ile Yapay Zekâ kavramı ortaya çıkmıştır. Alan Turing'in Turing testi ile ortaya çıkarmayı amaçladığı şekilde bir Yapay Zekâ, bir bilgisayarın veya bilgisayar kontrollü robotun akıllı varlıklarla ilişkili görevleri yerine getirme yeteneğidir [2].

Yapay Zekâ kavramının ortaya çıkmasının ardından akıllı varlıklarla ilişkili görevleri yerine getirebilmek amacıyla çeşitli akıllı algoritmalar geliştirilmiştir. Geliştirilen akıllı algoritmaların mevcut problemleri akıllı bir varlık kadar iyi bir şekilde çözebilmesi sonucunda Yapay Zekâ alanında hızla artan gelişmeler sağlanmış ve mevcut akıllı algoritmaların insan beynine benzer şekilde çalışabilme kabiliyeti üzerinde durulmuştur.

Akıllı algoritmalarının insan beynine olabildiğince yakın olarak modellenmesini sağlayabilecek bilimsel çalışmaların sürdürülmesinin ardından Yapay Sinir Ağı ortaya çıkmıştır. Yapay Sinir Ağı, insan beyni içerisindeki bilgi akışı ve işlenmesinin yapay olarak bilgisayarlar üzerinde modellenmesini içeren bir Yapay Zekâ alt dalıdır [3]. Yapay Sinir Ağları ile beynin bilgi işlem mimarisine benzer bir şekilde katmanlı bir sinir ağı mimarisi temel alınmaktadır. Her bir katman kendi içerisinde bilgiyi işleyen düğümlere sahiptir. Ek olarak bu düğümler diğer düğümlere bilginin aktarılmasını sağlayan ve bilgisayarın öğrenme işlemini gerçekleştiren belirli ağırlık değerlerine sahiptir. Bir Yapay Sinir Ağı, temelde bu bağlantılar üzerindeki ağırlıkları ayarlama işlemini gerçekleştirerek bilgisayarın verilen problemi öğrenmesini sağlamayı amaçlamaktadır. Bir Yapay Sinir Ağı üzerinde ağırlıkların güncellenmesini içeren tüm bu işlemler, genel olarak belirli verilerin Yapay Sinir Ağına gösterilmesi ve Yapay Sinir Ağına gösterilen bu verilerden öğrenme işlemi yapmasını sağlamaktadır [4].

Tüm bu gelişmelerden görülebileceği gibi teknolojik olarak Alan Turing testi ile başlayan gelişmeler günümüz teknolojisinde Yapay Sinir Ağları ile kendisini göstermektedir. Yapay Zekâ alanında yaşanan gelişmeler ile çözülebilecek problem uzayı artmakta ve her açıdan insanların yaklaşımına benzer şekilde etkin ve verimli çözümler üretilebilmektedir.

Bu çalışma kapsamında etkin ve verimli çözümler üretilebilmesine olanak tanıyan Yapay Sinir Ağına ek olarak yeni bir alternatif Yapay Zekâ Mimarisinin oluşturulması amaçlanmıştır. Ek olarak oluşturulmuş olan bu Yapay Zekâ Mimarisinin doğrusal olmayan bir problem olan XOR problemi üzerindeki başarımı analiz edilmiştir.

Oluşturulan Yapay Zekâ Mimarisi denklem sistemlerine ve katman mimarisine dayanmaktadır. Bu katman mimarisi sayesinde problem katmanlar düzeyinde çözüme ulaştırılarak, bozuk katmanların genel başarımı maksimuma ulaştıracak şekilde kendilerini düzeltmesi işlemi gerçekleştirilmiştir. Elde edilen Yapay Zekâ Mimarisinin doğrusal olmayan bir problem olan XOR problemi üzerinde test edilmesi sonucunda, mimarinin tahminleme işlemini düzgün bir şekilde yaptığı görülmüş ve TuringEQ Yapay Zekâ Mimarisinin doğrusal olmayan problemler üzerinde çalıştığı kanıtlanmıştır.

Çalışma kapsamında mevcuttaki Yapay Zekâ mimarilerine bağımlılığı azaltmak ve ileriki aşamalarda performans optimizasyonları yapılabilecek denklem sistemlerine dayalı yeni bir Yapay Zekâ sistemi geliştirilmesi amaçlanmıştır. Geliştirilen bu Yapay Zekâ mimarisi ile yeni mimarilerin Yapay Sinir Ağlarının ilk çağlarında çözmüş olduğu doğrusal olmayan bir problem olan XOR problemini çözdüğü kanıtlanarak Yapay Zekâ alanında sadece Yapay Sinir Ağları özelinde değil, farklı yaklaşımlar ile literatüre yeni yaklaşımların getirilebileceği görülmektedir. Ek olarak çalışma kapsamında denklem sistemlerine dayalı yeni bir mimarinin geliştirilmesinin sebebi, ileriki çalışmalarda denklem sadeleştirme ve denklem optimizasyonu teknikleri ile TuringEQ mimarisinin hesaplama karmaşıklığını azaltarak Yapay Zekâ sistemlerinin hesaplama maliyetlerini düşürmektir.

2. İlgili Çalışmalar

Yapay Zekâ alanında son yıllarda yapılan çalışmalar, insan zekasına benzer davranışların bilgisayarlarda modellenbilmesine dayanmaktadır. Bu kapsamda, literatürde çeşitli yaklaşımlara dayanan mimari ve konseptler geliştirilmiştir. Bu yeni Yapay Zekâ yaklaşımlarından ilki insan beynindeki sinir hücrelerinin ve bu hücreler

arasındaki bağlantıların modellemesini içeren Yapay Sinir Ağlarıdır. Yapay Sinir Ağları, son zamanlarda karmaşık süreçlerin modellenmesinde bir araç olarak uygulanmaya başlayan veriye dayalı modellerden biridir [5]. Yapay sinir ağları ile son zamanlarda yapılan birçok uygulama mevcuttur. Bu uygulamalardan bazıları doğal dil işleme, görüntü işleme ve makine öğrenmesi gibi alanları kapsamaktadır. Yapay Sinir Ağlarının Yapay Zekâ alanında en çok kullanılan mimari yaklaşım olmasına karşın kullanım alanlarına ve amaçlarına göre sinir ağları bağlamında veya tamamıyla özgün bir mimari olarak geliştirilen farklı Yapay Zekâ yaklaşımları da mevcuttur.

Bu yaklaşımlar Yapay Sinir Ağlarını temel alarak farklılaşan mimari bileşenlerine veya onarım mekanizmalarına dayanan çeşitli yeni sinir ağı tasarımlarını içermektedir. Bu tasarımlardan bazıları Neocognitron [6], LeNet-5 [7], AlexNet [8], VGGNet [9], GoogleNet [10] ve ResNet [11] olarak isimlendirilen sinir ağı mimarileridir. Yapay Zekâ alanında üretilen bu yeni mimariler, farklılaşarak ortaya çıkarken genellikle belirli problemlerin üstesinden gelmek veya mevcut yöntemlerin sınırlamalarını aşmak amacıyla tasarlanmıştır. Geleneksel yapay sinir ağları büyük veri kümelerinde etkili olmasına rağmen, hesaplama karmaşıklığı ve eğitim süreleri gibi zorluklarla karşı karşıya kalmasından kaynaklı olarak sinir ağlarını temel alan yeni mimarilerde geliştirilmiştir. Bu sorunları ele almak için geliştirilen Transformers mimarisi, özellikle doğal dil işleme alanında iyi bir performans sergilerken, EfficientNet gibi modeller, sinir ağlarının boyutlandırılmasındaki verimsizlikleri gidermek amacıyla ölçeklendirme stratejilerini optimize etmektedir. Graph Neural Networks ise, düğümler ve kenarlar arasındaki ilişkileri modelleyerek graf yapılarındaki verileri analiz etmede önemli bir adım atmıştır. Her bir yeni mimari, belirli bir problem veya sınırlamayı hedef alarak var olan yöntemlerden farklılaşan özgün yaklaşımlar sunarak Yapay Zekâ uygulamalarının kapsamını ve etkinliğini genişletmektedir. Yapay Sinir Ağlarından türeyen bu mimarilerin yanı sıra sinir ağı mimarilerinin manuel yapılandırmasından kaynaklanan hatalara odaklanan çalışmalarda mevcuttur. Bu çalışmalardan biri Termritthikun ve arkadaşları [12] tarafından geliştirilen ve sinir ağı mimarilerinin manuel olarak yapılandırılması problemini çözmeyi amaçlayan bir süper ağ mimarisidir. Çalışma kapsamında SaINAS ismi verilen bu süper ağ mimarisi ile belirginlik tahmini problemi için manuel yapılandırmadan kaynaklanan hataların engellenmesi amacıyla dinamik bir evrişim entegrasyonu kullanılarak ağ içerisinde otomatik bir yapılandırma gerçekleştirilmektedir. Literatürde sinir ağları temel alınarak geliştirilen birçok yapay zekâ mimarisi olmasına karşın istatistiksel yöntemler baz alınarak geliştirilen farklı Yapay Zekâ yaklaşımları da mevcuttur. Bu yaklaşımlardan biri sınıflandırma problemlerinde kullanılan ve istatistiksel yöntemlere dayanan Naive Bayes algoritmasıdır. Naive Bayes algoritması, Bayes teoremine dayanarak sınıflandırma yapan olasılık tabanlı bir algoritmadır [13]. İstatistiksel yöntemlere dayanan diğer Yapay Zekâ algoritmaları ise bağımlı ve bağımsız değişkenler arasındaki doğrusal ilişkileri modellemeye dayanan doğrusal regresyon ve belirli bir olayın olma olasılığına tahmin etmeye dayalı olan lojistik regresyon algoritmalarıdır. Yapay Zekâ alanında yaygın olarak kullanımda olan diğer yaklaşımlar ise verilerin komşuluk ilişkilerinin analizine dayanan k en yakın komşu algoritması ve k ortalamalar algoritmalarıdır.

Literatür de Yapay Zekâ alanı ile ilgili gerçekleştirilen çalışmalar incelendiğinde çalışmaların büyük bir kısmının Yapay Sinir Ağları temelinde gerçekleştirildiği görülmektedir. Son yıllarda yapılan çalışmalar tamamıyla yeni ve özgün Yapay Zekâ mimarileri geliştirilmesinden ziyade mevcutta var olan Yapay Zekâ mimarilerinin iyileştirilmesine yönelik çalışmalardır. Mevcuttaki Yapay Sinir Ağlarının beyin tümörlerinin sınıflandırılmasını içeren ve Joshi ve arkadaşları [14] tarafından yapılan bir çalışmada parçacık sürü optimizasyonu kullanımını iyileştiren ve Cuckoo arama algoritmasını kullanan PSCS adında yeni bir optimizasyon yaklaşımı önerilmektedir. Önerilen optimizasyon yaklaşımının sinir ağları ile birleştirilmesi sonucunda başarılı sonuçlar elde edilmiştir. Sinir ağı mimarilerinin dışında veri setlerinin mimarilere uygun yapılara dönüştürülmesi için yapılan çalışmalarda mevcuttur. Bu çalışmalardan biri Rosso ve arkadaşları [15] tarafından inşaat yapılarının dayanıklılığını ölçmek için sinir ağlarının kullanımında performans iyileştirmesi için veri ön işleme adımında Fourier dönüşümlü ve dönüşümsüz tekniklerin Yapay Sinir Ağının başarısına katkısını incelemektedir. Literatürde Yapay Zekânın eğitilmesi için kullanılacak yeni veri setlerinin üretimine yönelik çalışmalar da mevcuttur. Bu çalışmalardan biri Aslan ve arkadaşları [16] tarafından yalan söyleyen ve söylemeyen insanların beyinlerinden alınan EEG sinyal örneklerinden oluşan büyük bir veri setinin oluşturulmasına ve Yapay Sinir Ağları ile analizine odaklanmaktadır.

Literatürde gerçekleştirilen bu çalışmalara bakıldığında Yapay Zekâ alanındaki çalışmaların yoğunlukla sinir ağları temelinde gerçekleştirildiği görülmektedir. Bu kapsamda, Yapay Zekâ alanındaki çalışmaların yeni ve özgün mimari ve yaklaşımlarla geliştirilmesi için yapılan bu çalışmanın literatüre önemli bir katkı sunacağı düşünülmektedir.

3. Problem Kümeleri

Bir problem, yanıtının bilimsel yöntemlerle bulunması veya kurallarla çözülmesi gereken bir soru olarak tanımlanmaktadır. İnsanlar geçmişten günümüze kadar birçok problem ile karşılaşmış ve bu problemlere o günün mevcut imkanlarını kullanarak çözümler üretmişlerdir. Elde mevcut olan bir durumun problem olarak nitelenebilmesi için şu özelliklere sahip olması gerekmektedir:

- Durumun benzersiz olması
- Çözümünün hali hazırda biliniyor olmaması
- Durumun zihinde karmaşıklığa sebep olması
- Daha önce edinilen bilgi ve tecrübe ile çözülebilir nitelikte olması

Temel olarak bir problemin problem olarak nitelenebilmesi için belirtilen özellikleri sağlaması gerekmektedir.

Problemlerin ortaya çıkışı, teknolojik geliştirmeleri hızlandırmış ve belirli problem karakteristiklerine ek olarak yeni çözüm yöntemleri ortaya çıkmıştır. Problemlerin karakteristiği çözüme giden yolda önemli bir yer tutmaktadır. Çözümün bilinmesinden ziyade genellenebilir bir çözüm elde edebilmek için üzerinde çalışılan problem karakteristiklerinin doğru bir şekilde analiz edilmesi gerekmektedir. Çalışma kapsamında ele alınacak iki ana problem karakteristiği mevcuttur. Bu problem karakteristiklerinden ilki herhangi bir düzensizlik içermeyen doğrusal problemler iken, ikincisi ise düzensizlikler barındırabilen doğrusal olmayan

3.1. Doğrusal Problemler

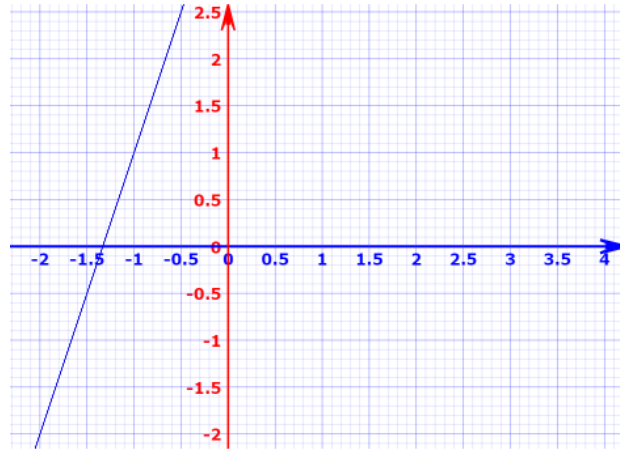
Doğrusal problemler, kendi içerisinde bir doğrusallık içeren ve bu doğrusallığı bozacak herhangi bir düzensizliği barındırmayan problemlerdir. Bu tür problemlerde veriler, doğrusal bir düzlem üzerinde ilerlemektedir.

3.1.1. Doğrusal Problemlerin Karakteristiği

Doğrusal problemler, düzgün bir dağılıma sahiptir. Bu problem türüne dahil olan problemler kendi içerisinde düzenliliği bozacak herhangi bir karakteristik taşımaz. Ayrıca doğrusal bir problem düzlemsel olarak bakıldığında doğrusallık taşımaktadır. Verilerin dağılımı, bu doğrusallığı bozacak herhangi bir düzensizlik içermemektedir. Denklem 1'de basit bir doğrusal denklem problemi verilmiştir. Bu problem, denklem tanımından da anlaşılabilir gibi doğrusal bir yapıya sahip olup, kendi içerisinde herhangi bir düzensizlik barındırmadığından doğrusal problem kategorisine girmektedir.

$$y = 3x + 5 \quad (1)$$

Başka bir ifade ile Denklem 1'de verilen denklem Şekil 1'de görüldüğü gibi düzlem üzerinde doğrusal bir dağılıma sahip olduğundan doğrusal bir problem olarak sınıflandırılabilir.



Şekil 1. Doğrusal bir denklem için düzlem üzerinde grafiksel gösterim

3.1.2. Doğrusal Problemlerin Çözüm Yöntemleri

Doğrusal problemlerin çözülebilmesi için bir amaç fonksiyonu seçilmelidir. Seçilen amaç fonksiyonunun belirli kısıtlara uyması şartıyla, verilerin doğrusallığından faydalanarak doğrusal problemlerin çözülmesi işlemi gerçekleştirilir. Doğrusal problemler çözülmürken izlenmesi gereken adımlar sırasıyla şu şekildedir [17]:

- Karar değişkenleri tanımlanır: Amaç fonksiyonunun kontrolü için gerekli olan karar değişkenlerinin tanımlanması işlemi yapılır. Bu adımda tanımlanan karar değişkenleri amaç fonksiyonu üzerinde ayarlamalar yapmak için kullanılmaktadır.
- Amaç fonksiyonu belirlenir: Karar değişkenleri kullanılarak oluşturulan doğrusal denklem sistemine amaç fonksiyonu denir. Doğrusal problemlerin çözümü için amaç fonksiyonunun karar değişkenleri ile ayarlanması sonucunda istenen çözümün elde edilebileceği bir denklem elde edilir.
- Kısıtlama kümesi belirlenir: Karar değişkenlerinin değer aralıkları için belirli kısıtlamalar bulunmaktadır. Çözüm işleminde belirlenen bu kısıtlamalara uyulmalıdır.
- Doğrusal problemi çözme yöntemi belirlenir: Doğrusal problemi çözebilmek için tek yönlü yöntem, R kullanarak sorunu çözme, problemin grafik yöntemi kullanarak çözülmesi ve açık bir çözücü kullanarak sorunu çözme gibi çözüm yöntemlerinin biri seçilir.

Bütün bu doğrusal problem çözme adımlarının gerçekleştirilmesinin ardından seçilmiş olan doğrusal problem çözüm tekniğine göre, karar değişkenleri üzerindeki kısıtlamalara bağlı kalınarak belirlenmiş olan amaç fonksiyonunun optimum olduğu nokta bulunmaya çalışılır. Çözüm yönteminin kısıtlamalara bağlı kalınarak uygulanması sonucunda elde edilen optimum alan problemin çözümünü içermektedir.

3.2. Doğrusal Olmayan Problemler

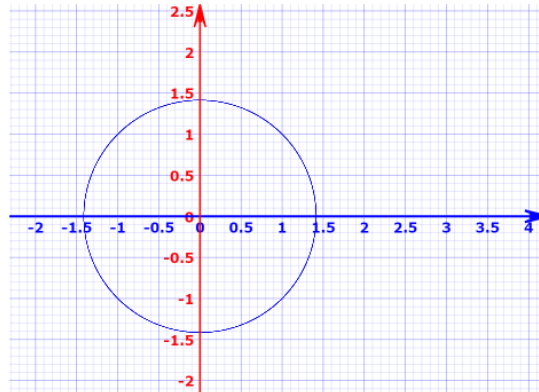
Doğrusal olmayan problemler, kendi içerisinde bir doğrusallık içermeyen ve girdiler ile çıktılarının birbiriyle orantılı olmadığı problemlerdir. Bu tür problemlerde veriler bir düzlem üzerinde doğrusal bir dağılım yerine karmaşık olarak yayılmış halde bulunur.

3.2.1. Doğrusal Olmayan Problemlerin Karakteristiği

Doğrusal olmayan problemler düzgün olmayan bir dağılıma sahiptir. Bu problem türüne dahil olan problemler kendi içerisinde düzenli herhangi bir karakteristik taşımaz. Doğrusal olmayan bir problem düzlemsel olarak bakıldığında doğrusallık yerine rastgele bir dağılıma sahiptir. Verilerin dağılımı, doğrusallığa engel olan düzensizlikler içermektedir. Denklem 2’de basit bir doğrusal olmayan denklem problemi verilmiştir. Bu problem, denklem tanımından da anlaşılabilir gibi doğrusal bir yapıya sahip olmayıp, kendi içerisinde girdi verilerinden çıktı verilerine düzenliliği bozacak dağılımlara sahip olduğundan doğrusal olmayan problem kategorisine girmektedir.

$$2x^2 + 2y^2 = 4 \quad (2)$$

Başka bir ifade ile Denklem 2’de verilen denklem Şekil 2’de görüldüğü gibi düzlem üzerinde doğrusal bir dağılıma sahip olmadığından doğrusal olmayan bir problem olarak sınıflandırılabilir.



Şekil 2. Doğrusal olmayan bir denklem için düzlem üzerinde grafiksel gösterim

3.2.2. Doğrusal Olmayan Problemlerin Çözüm Yöntemleri

Doğrusal olmayan problemlerde doğrusal problemlere benzer şekilde amaç fonksiyonu belirlenerek, bu amaç fonksiyonunun seçilen yöntemle göre çözülmesi sağlanır. Temel anlamda doğrusal olmayan problem çözüm adımları da doğrusal problem çözüm adımlarına benzerlik göstermekle birlikte üzerinde çalışılan problemlerin karakteristik farklılıklarından dolayı çözüm yöntemleri arasında farklılıklar bulunmaktadır. Doğrusal olmayan problemlerde amaç fonksiyonunun çözümü, bu fonksiyonun yapısına göre yapılmaktadır.

Amaç fonksiyonu içbükey (maksimizasyon problemi) veya dışbükey (minimizasyon problemi) ise ve kısıtlama kümesi dışbükey ise, problem dışbükey olarak adlandırılır ve çoğu durumda dışbükey genel yöntemler kullanılabilir. Amaç fonksiyonu ikinci dereceden ve kısıtlamalar doğrusal ise, ikinci dereceden programlama teknikleri kullanılır. Amaç fonksiyonu bir içbükey ve bir dışbükey fonksiyonun oranıysa (maksimizasyon durumunda) ve kısıtlamalar dışbükey ise, o zaman problem kesirli programlama teknikleri kullanılarak bir dışbükey optimizasyon problemine dönüştürülebilir. Dışbükey olmayan problemleri çözmek için çeşitli yöntemler mevcuttur. Bir yaklaşım, doğrusal programlama problemlerinin özel formülasyonlarını kullanmaktır. Başka bir yöntem dal ve sınır kullanımını içerir [18].

Doğrusal olmayan problemlerin çözümü, genel olarak bir noktada maliyeti yaklaşık çözümlerden herhangi biri için elde edilen en iyi alt sınıra eşit olacak şekilde gerçek bir çözüm elde edilerek yapılmaktadır. Bu çözüm, benzersiz olmasa da optimaldir.

Doğrusal olmayan problemlerin bu şekilde çözümünde kullanılan algoritma, mümkün olan en iyi çözümün bulunan en iyi noktadan bir tolerans dahilinde olduğu güvencesiyle erken durdurulabilir; bu tür noktalara ϵ -optimal denir. Sonlu sonlandırmayı sağlamak için ϵ -optimal noktalarda sonlandırma genellikle gereklidir. Bu özellikle büyük türevlenebilirlik ve kısıtlama nitelikleri altında geçerli olan bir durumdur.

Doğrusal olmayan problemlerin, doğrusal problemler gibi kesin çözümlere sahip olmaması ve hata tolere edilebilir yaklaşımlar ile çözüm işleminin gerçekleştirilmesi nedeniyle bu tür problemler Yapay Zekâ ile çözülmesi amaçlanan problemlerdir. Doğrusal olmayan problemler, Yapay Zekâ teknikleri ile ele alınan başlıca problemler arasındadır [19].

4. Materyal ve Metot

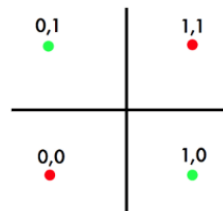
Yapılan çalışmanın temel amacı, Yapay Zekâ tekniklerinden olan ve yaygın olarak her alanda kullanım olanağına sahip olan Yapay Sinir Ağlarına alternatif bir Yapay Zekâ Mimarisi geliştirmektir. Bu amaçla yapılan çalışmanın öncelikli hedefi, geliştirilmiş olan Yapay Zekâ Mimarisinin doğrusal olmayan bir problem olan XOR problemi üzerinde çalıştığının kanıtlanmasıdır.

4.1. Materyal

Yapılan çalışmada veri seti olarak XOR verisi kullanılmıştır. Oluşturulan Yapay Zekâ Mimarisi XOR verisi üzerinde test edilerek mimarinin doğrusal olmayan bir problem üzerinde çalışıp çalışmadığı test edilmiştir.

4.1.1. XOR Verisi

Doğrusal olmayan bir karakteristiğe sahip olması ve mimarinin doğrusal olmayan bir problem üzerinde çalıştığının doğrulanması için XOR problemi kullanılmıştır. XOR problemi, doğrusal olmayan bir problemdir. Bu kapsamda XOR içerisinde girdi verilerinin çıktı verilerine oranı Şekil 3'te görüldüğü gibi düzenli bir dağılım göstermemektedir. Düzenli dağılım göstermeyen XOR verisi, mimarinin doğrusal olmayan problemler üzerindeki yeteneğini kanıtlamada iyi bir başlangıç noktası olacaktır.



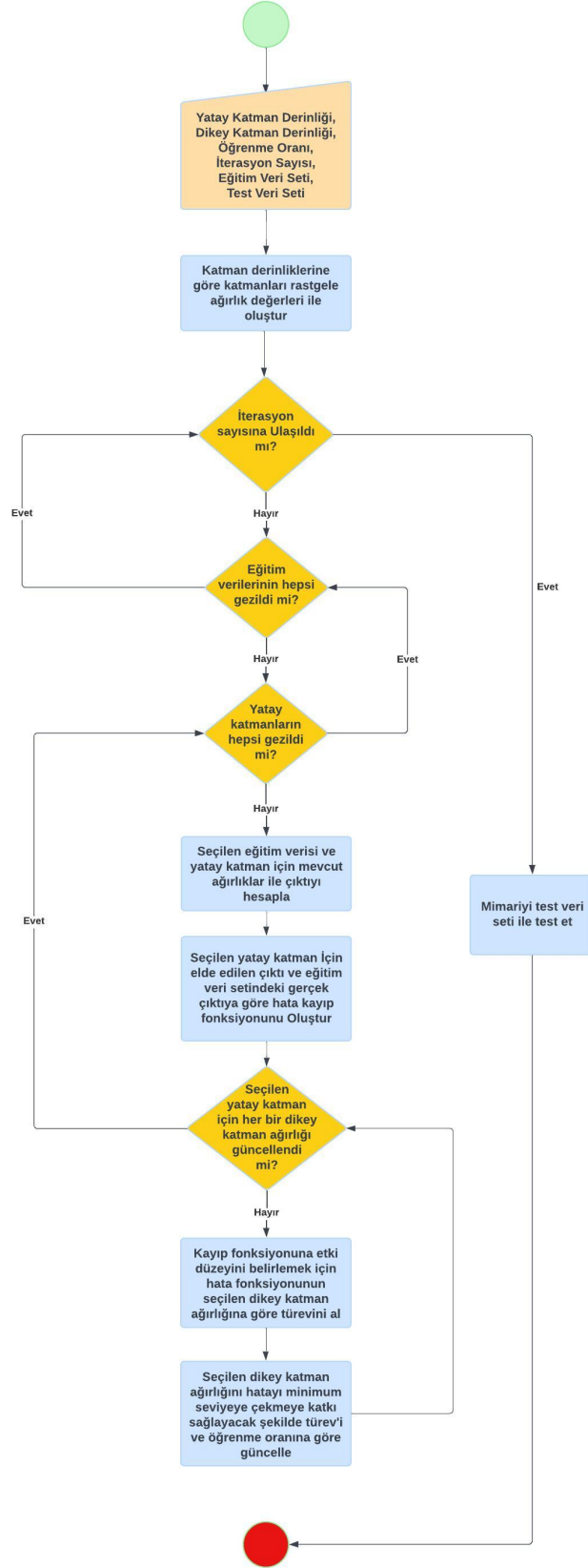
Şekil 3. XOR verisinin düzlem üzerinde gösterimi [20]

4.2. Metot

Yapılan çalışmada öncelikle geliştirilecek olan Yapay Zekâ Mimarisi için teorik temeller tasarlanmıştır. Teorik temellerin tasarlanarak doğrulanmasının ardından mimari tasarım yapılarak, yapılan mimari tasarımın hızlı geliştirme imkânı ve OOP (Object Oriented Programming) mimarisinde kodlama yeteneği sunan PYTHON programlama dili kullanılarak kodlanması işlemi yapılmıştır. PYTHON programlama dili ile OOP mimarisine uygun şekilde TuringEQ kütüphanesi olarak kodlanan mimarinin test işlemleri, XOR veri seti üzerinde yapılarak elde edilen sonuçlardan TuringEQ Yapay Zekâ Mimarisinin doğrusal olmayan problemler üzerinde çalışıp çalışmadığı test edilmiştir

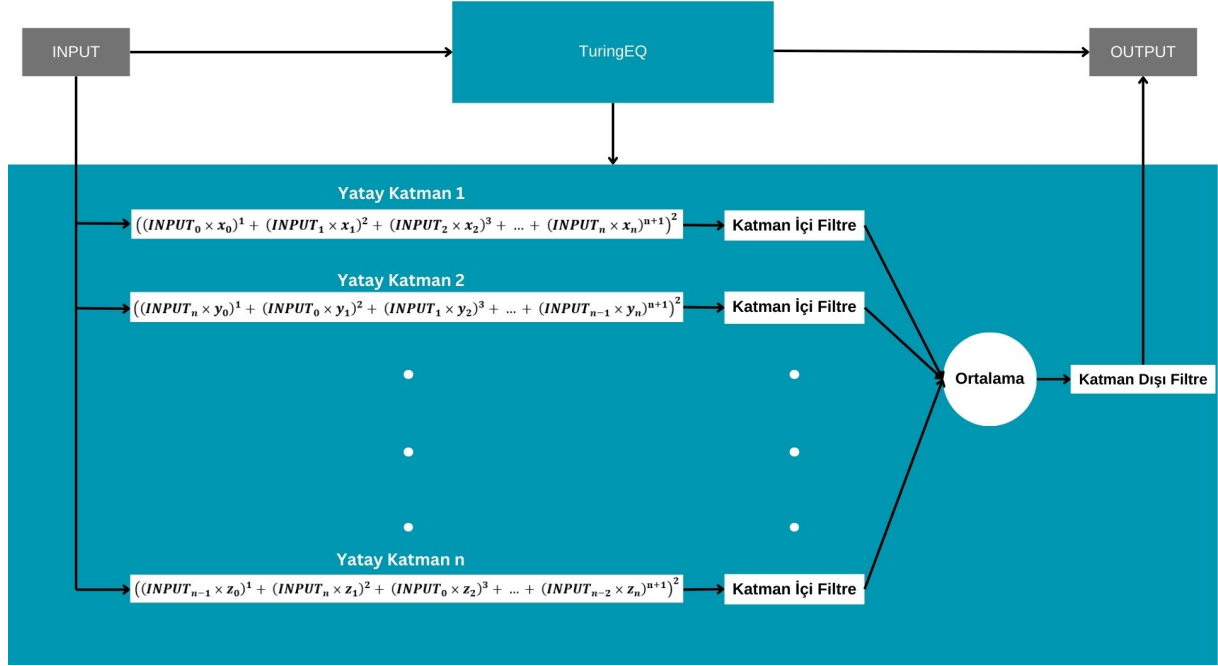
4.2.1. TuringEQ Mimarisi

Yapılan çalışma sonucunda TuringEQ Yapay Zekâ Mimarisi oluşturulmuştur. TuringEQ mimarisi, denklem sistemlerinden faydalanan ve katmanlı bir mimariye sahip olan Yapay Zekâ Mimarisi'dir. Bu mimaride her bir katmanın bir denklem sistemini ifade ettiği yatay katmanlar ve bu yatay katmanlar içerisindeki denklemlerin bilinmeyen değişkenlerini ifade eden dikey katmanlar bulunmaktadır. Mimariye sunulan girdi değerlerinin her bir yatay katmanın farklılaşan dikey katman ağırlıklarına dağıtılması ile girdi verilerinin işlenmesi sağlanmaktadır. Mimarinin genel akış şeması Şekil 4'te gösterilmektedir. Mimariye girdi olarak verilen yatay katman derinliği, dikey katman derinliği, öğrenme oranı, iterasyon sayısı, eğitim veri seti ve test veri seti bulunmaktadır. Bu girilen verilerden yatay katman derinliği mimarinin içerisindeki denklem sistemlerinin sayısını ifade ederken dikey katman derinliği ise bu yatay katmanlar içerisindeki denklem sistemlerinin değişken sayılarını ifade etmektedir. Öğrenme oranı, yatay katmanlar içerisindeki dikey katman ağırlıklarının değişim oranını belirlemektedir. İterasyon sayısı, mimarinin veri seti üzerinde kaç iterasyon da çalıştırılarak öğrenme işleminin gerçekleştirileceğini belirlemektedir. Son olarak eğitim veri seti, mimarinin eğitimi için kullanılırken test veri seti ise eğitilen mimarinin test edilmesi sürecinde başarıyı ölçmek için kullanılmaktadır. TuringEQ mimarisine gerekli girdilerin verilmesinin ardından mimari kendi içerisindeki yatay ve dikey katmanları, dikey katman ağırlıkları rastgele başlangıç değerlerine ayarlanacak şekilde oluşturmaktadır. Katmanların oluşturulmasının ardından mimari girilen iterasyon adedince kendi içerisinde eğitim işlemini gerçekleştirmektedir. TuringEQ mimarisinin eğitim stratejisi, farklılaşan her bir yatay katmanın kendi içerisinde dikey katman ağırlıklarının değiştirilmesi ile onarılmasına dayanmaktadır. Bu kapsamda, seçilen her bir eğitim verisi için her bir yatay katmanın çıktıları hesaplanmakta ve hesaplanan bu çıktılar eğitim veri setindeki gerçek çıktılar ile karşılaştırılarak hata kayıp fonksiyonu oluşturulmaktadır. Belirli bir yatay katmanın hatalı hesaplama durumunu ölçebilmek için oluşturulmuş olan hata kayıp fonksiyonunun oluşturulmasının ardından ilgili yatay katmanın içerisindeki dikey katman ağırlıklarının yatay katmanın hatasına etki düzeyinin belirlenmesi için güncellenmek istenen her bir ağırlığa göre türev alınmaktadır. Alınan türev ile hatayı minimum noktaya yakınsayacak güncellenmek istenen dikey katman ağırlığının değişim yönü belirlenerek, öğrenme oranına göre bu ağırlığın güncellenmesi işlemi yapılmaktadır. TuringEQ mimarisi, her bir eğitim verisi için yatay katmanların her bir dikey katman ağırlıklarını bu stratejiye göre güncelleyerek öğrenme işlemini gerçekleştirmektedir,



Şekil 4. TuringEQ akış şeması

TuringEQ mimarisi Şekil 5'te görüldüğü gibi kendi içerisinde katmanlı bir yapıya sahip olmakla birlikte her bir katman kendi içerisindeki başarımı maksimum düzeye çıkarmaktan sorumludur.



Şekil 5. TuringEQ genel mimarisi

TuringEQ mimarisinde her bir katmanın çıktısı mimarinin çıktısına aritmetik ortalama ile etki etmektedir. Bu şekilde, katmanların genel çıktıya aynı düzeyde etki etmesi sağlanarak katmanlardaki hata düzeltimlerinin genel hatayı minimize etmesi sağlanmaktadır. Denklem 3'te TuringEQ içerisindeki bir yatay katmanın genel yapısı gösterilmektedir. Her bir girdinin bilinmeyen değişkenler ile çarpılarak toplanması sonucunda, denklemi yeterince dağıtık ve türevi alınabilir hale getirebilmek amacıyla sonunda karesi alınmaktadır.

$$((INPUT_0 \times x_0)^1 + (INPUT_1 \times x_1)^2 + (INPUT_2 \times x_2)^3 + \dots + (INPUT_n \times x_n)^{n+1})^2 \quad (3)$$

Şekil 5'te görüldüğü gibi her bir yatay katman kendi içerisinde kendi bilinmeyen değişken değerlerine sahiptir. Bu bilinmeyen değişkenler katman içi hataların onarılması için kullanılmaktadır.

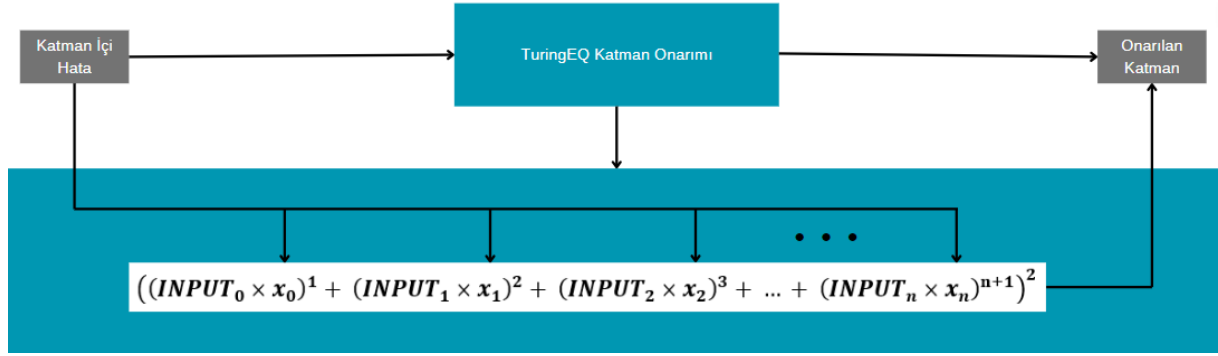
TuringEQ mimarisinde hata onarımının yapılabilmesi için her bir katmandan elde edilen çıktılar gerçek çıktılar ile farkına dayanan ve Denklem 4'te belirtilen bir kayıp fonksiyonu üzerinden hata tespiti gerçekleştirilmektedir.

$$0.5 \times (\text{Tahmin Edilen Değer} - \text{Gerçek Değer})^2 \quad (4)$$

Denklem 5'te gösterildiği gibi kayıp fonksiyonunun her bir yatay katman içerisindeki her bir bilinmeyen değişkene göre türevinin hesaplanması sonucunda bilinmeyen değişkenlerin belirli bir öğrenme oranında değiştirilmesi işlemi gerçekleştirilmektedir.

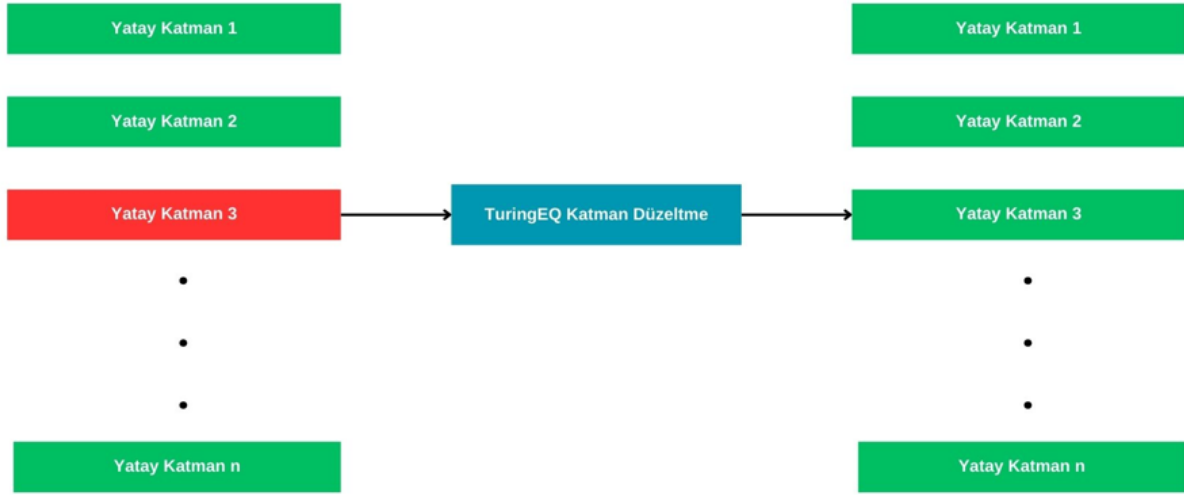
$$\sum_{k=0}^n x_k - \left(\frac{d(0.5 \times (\text{Tahmin Edilen Değer} - \text{Gerçek Değer})^2)}{d(x_k)} \right) \times \text{Öğrenme Katsayısı} \quad (5)$$

Belirtilen hata onarım denklemlerinin kullanılmasıyla Şekil 6'da gösterildiği şekilde yatay katmanlar üzerindeki her bir bilinmeyen değişken için hata onarım işlemi gerçekleştirilmiş olur.



Şekil 6. TuringEQ yatay katman değişkenlerinin düzeltilmesi

TuringEQ içerisindeki her bir yatay katmanın belirtilen şekilde düzenlemesiyle mimari genelinde bozuk katmanların onarım işlemi yapılarak Şekil 7’de görüldüğü gibi genel olarak mimari düzeyinde hata onarımı gerçekleştirilmiş olur.



Şekil 7. TuringEQ bozuk katman onarımı

TuringEQ mimarisinde yatay katmanlara ek olarak dikey katmanlar mevcuttur. Dikey katmanların derinliği her bir yatay katman içerisindeki bilinmeyen değişken sayılarını temsil etmektedir. Yatay katman içerisindeki bilinmeyen değişken sayısı arttıkça dikey katman derinliği artmaktadır. Çok özellikli girdilerde, girdileri her bir katmanda farklı bilinmeyen değişkenler üzerinden ayarlayarak türevlerin çeşitlendirilmesi ve girdiler üzerinden birbirinden farklılaşan denklem sistemleri oluşturulması amaçlanmaktadır. 1. Yatay katmanı temsil eden Denklem 6 ve 2. Yatay katmanı temsil eden Denklem 7’de belirtildiği şekilde TuringEQ mimarisine sunulan girdilerin sıralaması her bir katmanda farklılaşarak birbirinden farklı denklem sistemleri elde edilmektedir.

$$((INPUT_0 \times x_0)^1 + (INPUT_1 \times x_1)^2 + (INPUT_2 \times x_2)^3 + \dots + (INPUT_n \times x_n)^{n+1})^2 \quad (6)$$

$$((INPUT_n \times y_0)^1 + (INPUT_0 \times y_1)^2 + (INPUT_1 \times y_2)^3 + \dots + (INPUT_{n-1} \times y_n)^{n+1})^2 \quad (7)$$

Farklı denklem sistemleri elde etmenin yanı sıra, etkili girdilerin yüksek düzeyli olarak temsil edildiği denklem sistemlerinde girdi karakteristiklerinin net olarak çıkarılması amaçlanmaktadır. Bu kapsamda, yüksek dereceli bilinmeyen değişkenler üzerinden ayarlanan girdilerin etki düzeyleri yüksek olurken, diğer denklemlerde aynı girdinin etki düzeylerinin düşürülmesi sebebiyle girdiler üzerinde farklı düzeylerin test edilmesi ve en uygun dereceye sahip düzey üzerinde ayarlamaların bulunması sağlanmaktadır.

5. Bulgular ve Tartışma

Elde edilen TuringEQ mimarisinin XOR veri seti üzerinde farklı katman derinlikleri ve öğrenme katsayısındaki değişimlerin mimarinin öğrenme durumuna etkisini görebilmek amacıyla oluşturulan ve farklı yatay katman derinliği, dikey katman derinliği ve öğrenme katsayılarını içeren 2 farklı senaryo için test edilmesi sonucunda elde edilen bulgular üzerinden çeşitli analizler yapılmıştır. Farklı senaryolar için TuringEQ mimarisinin XOR veri seti üzerinde elde etmiş olduğu başarımların incelenmesinin ardından TuringEQ mimarisinin Yapay Sinir Ağları ile aynı problem üzerinde elde etmiş olduğu başarımlar açısından karşılaştırmalar yapılarak, yeni oluşturulan Yapay Zekâ Mimarisinin mevcutta yaygın bir biçimde kullanılan bir Yapay Zekâ Mimarisi olan Yapay Sinir Ağlarına karşı olumlu ve olumsuz tarafları incelenmiştir.

5.1. TuringEQ Mimarisinin Senaryo 1 için Test Edilmesi

TuringEQ mimarisi ilk senaryoda 2 yatay katman derinliği, 4 dikey katman derinliği ve 0,1 öğrenme katsayısı için XOR problemi üzerinde çalıştırıldığında Tablo 1’de görüldüğü gibi başarımları yüksek sonuçlar elde edilmiştir.

Tablo 1. TuringEQ mimarisi senaryo 1 için sonuçlar

X	Y	Beklenen Sonuç	Tahmin Edilen Sonuç	Hata
0	0	0	0,0	0
0	1	1	1,135508236674183	-0,135508236674183
1	0	1	0,9840631326887465	0,0159368673112535
1	1	0	0,06507361395531353	-0,06507361395531353

TuringEQ mimarisinin senaryo 1 için eğitim öncesi başlangıç tahmin değerleri Tablo 2’de görülmektedir.

Tablo 2. TuringEQ mimarisi senaryo 1 için eğitim öncesi tahminleri

X	Y	Beklenen Sonuç	Tahmin Edilen Sonuç	Hata
0	0	0	0,0	0
0	1	1	0,7997997799281024	0,20020022007108976
1	0	1	0,30401244023373786	0,69598755976626214
1	1	0	3,1714671185109644	-3,1714671185109644

TuringEQ mimarisinin senaryo 1 için eğitim öncesi başlangıç ağırlık değerleri Tablo 3'te görülmektedir.

Tablo 3. TuringEQ mimarisi senaryo 1 için eğitim öncesi başlangıç ağırlık değerleri

Yatay Katmanlar	1.Dikey Katman Ağırlığı	2.Dikey Katman Ağırlığı	3.Dikey Katman Ağırlığı	4.Dikey Katman Ağırlığı
1.Yatay Katman	0,5535758386896907	0,4251791630662658	0,24124945159930522	0,9140154121540919
2.Yatay Katman	0,8977400944587374	0,7309848405146916	0,22831793153423074	0,13132787610390073

TuringEQ mimarisinin senaryo 1 için eğitim sonrası ağırlık değerleri Tablo 4'te görülmektedir.

Tablo 4. TuringEQ mimarisi senaryo 1 için eğitim sonrası ağırlık değerleri

Yatay Katmanlar	1.Dikey Katman Ağırlığı	2.Dikey Katman Ağırlığı	3.Dikey Katman Ağırlığı	4.Dikey Katman Ağırlığı
1.Yatay Katman	-0,992434997994770	0,379327937039821	0,227173207889428	0,856354088385537
2.Yatay Katman	-1,00326665617092	0,99984477281268	0,148377417701089	0,132736814366460

TuringEQ mimarisinin senaryo 1 için XOR problemi üzerinde eğitim işleminin 100 iterasyon tekrarı için gerçekleştirilmesi sonucunda XOR tahminlemesinin çok yakın düzeylerde gerçekleştiği görülmektedir. Eğitim öncesi ve eğitim sonrasında mimarinin elde ettiği sonuçlar mutlak hataya göre incelendiğinde eğitim öncesinde hata oranı 1,02 iken eğitim sonrasında ise hata oranının 0,05'e gerilediği tespit edilmiştir. Bu kapsamda TuringEQ mimarisinin doğrusal olmayan problemler üzerinde çalıştığının kanıtlanması açısından senaryo 1 için elde edilen sonuçların olumlu düzeyde olduğu görülmüştür.

5.2. TuringEQ Mimarisinin Senaryo 2 için Test Edilmesi

TuringEQ mimarisi, ikinci senaryoda 2 yatay katman derinliği, 8 dikey katman derinliği ve 0,05 öğrenme katsayısı için XOR problemi üzerinde çalıştırıldığında Tablo 5'te görüldüğü gibi başarıyı yüksek sonuçlar elde edilmiştir.

Tablo 5. TuringEQ mimarisi senaryo 2 için sonuçlar

X	Y	Beklenen Sonuç	Tahmin Edilen Sonuç	Hata
0	0	0	0,0	0
0	1	1	1,08214460177223	-0,08214460177223
1	0	1	1,0927197157602675	-0,0927197157602675
1	1	0	0,02014012966719759	-0,02014012966719759

TuringEQ mimarisinin senaryo 2 için eğitim öncesi başlangıç tahmin değerleri Tablo 6'da görülmektedir.

Tablo 6. TuringEQ mimarisi senaryo 2 için eğitim öncesi tahminleri

X	Y	Beklenen Sonuç	Tahmin Edilen Sonuç	Hata
0	0	0	0,0	0
0	1	1	0,5714640914273006	0,4285359085726994
1	0	1	0,8905479121079419	0,1094520878920581
1	1	0	5,385353620277009	-5,385353620277009

TuringEQ mimarisinin senaryo 2 için eğitim öncesi başlangıç ağırlık değerleri Tablo 7’de görülmektedir.

Tablo 7. TuringEQ mimarisi senaryo 2 için eğitim öncesi başlangıç ağırlık değerleri

Yatay Katmanlar	1.Dikey Katman Ağırlığı	2.Dikey Katman Ağırlığı	3.Dikey Katman Ağırlığı	4.Dikey Katman Ağırlığı	5.Dikey Katman Ağırlığı	6.Dikey Katman Ağırlığı	7.Dikey Katman Ağırlığı	8.Dikey Katman Ağırlığı
1.Yatay Katman	0,53639384	0,14171107	0,81874787	0,18594171	0,31970408	0,93385278	0,11875274	0,2928139
2.Yatay Katman	0,55280375	0,67850139	0,61142288	0,02961601	0,35183227	0,82342399	0,61788556	0,05130254

TuringEQ mimarisinin senaryo 2 için eğitim sonrası ağırlık değerleri Tablo 8’de görülmektedir.

Tablo 8. TuringEQ mimarisi senaryo 2 için eğitim sonrası ağırlık değerleri

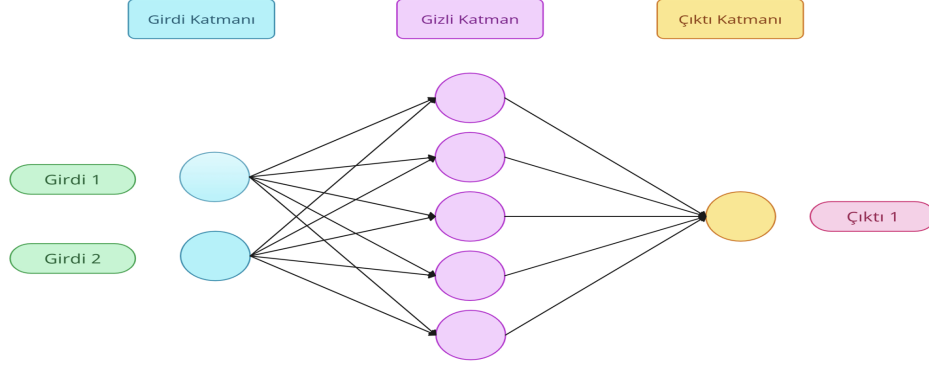
Yatay Katmanlar	1.Dikey Katman Ağırlığı	2.Dikey Katman Ağırlığı	3.Dikey Katman Ağırlığı	4.Dikey Katman Ağırlığı	5.Dikey Katman Ağırlığı	6.Dikey Katman Ağırlığı	7.Dikey Katman Ağırlığı	8.Dikey Katman Ağırlığı
1.Yatay Katman	-1,37037909	0,58164229	0,71769999	0,21688348	0,32124603	0,89375705	0,11875717	0,2928139
2.Yatay Katman	-1,02159386	0,63632502	0,22937971	0,02964162	0,30387616	0,88088905	0,49511525	0,05130254

TuringEQ mimarisinin senaryo 2 için XOR problemi üzerinde eğitim işleminin 100 iterasyon tekrarı için gerçekleştirilmesi sonucunda XOR tahminlemesinin senaryo 1’de görüldüğü gibi çok yakın düzeylerde gerçekleştiği görülmektedir. Eğitim öncesi ve eğitim sonrasında mimarinin elde ettiği sonuçlar mutlak hataya göre incelendiğinde eğitim öncesinde hata oranı 1,48 iken eğitim sonrasında ise hata oranının 0,05’e gerilediği tespit edilmiştir. Bu kapsamda TuringEQ mimarisinin doğrusal olmayan problemler üzerinde çalıştığının kanıtlanması açısından senaryo 1’de olduğu gibi senaryo 2 içinde elde edilen sonuçların olumlu düzeyde olduğu görülmüştür.

5.3. TuringEQ Mimarisi ve Yapay Sinir Ağlarının Karşılaştırılması

TuringEQ mimarisinin belirlenmiş olan senaryolar üzerinden XOR veri seti üzerindeki başarımının doğrulanmasının ardından Yapay Sinir Ağlarının aynı veri seti üzerinde senaryolarda kullanılan aynı iterasyon sayısı ve öğrenme katsayısına göre göstermiş olduğu başarıma göre iki Yapay Zekâ Mimarisi arasında

karşılaştırmalar yapılmıştır. Karşılaştırmada kullanılan Yapay Sinir Ağı Şekil 8’de gösterildiği şekilde 2 düğümünden oluşan 1 girdi katmanı, 5 düğümünden oluşan 1 gizli katman ve 1 düğümünden oluşan 1 çıktı katmanından oluşan çok katmanlı bir sinir ağıdır.



Şekil 8. Yapay Sinir Ağı Mimarisi

Yapay Sinir Ağı mimarisinin senaryo 1 için 0,1 öğrenme katsayısı ve 100 iterasyon eğitim sonucunda mimarinin eğitim sonrası tahminleri Tablo 9’da gösterilmektedir.

Tablo 9. Yapay Sinir Ağı mimarisi senaryo 1 için eğitim sonrası tahminleri

X	Y	Beklenen Sonuç	Tahmin Edilen Sonuç	Hata
0	0	0	0,45077726	-0,45077726
0	1	1	0,44137248	0,55862752
1	0	1	0,55580937	0,44419063
1	1	0	0,5710973	-0,5710973

Yapay Sinir Ağı mimarisinin senaryo 1 için XOR problemi üzerinde eğitim işleminin 100 iterasyon tekrarı için gerçekleştirilmesi sonucunda XOR tahminlemesinin doğru bir şekilde yapılabilmesi için iterasyon tekrarının yetersiz düzeyde kaldığı görülmektedir. Eğitim sonrasında mimarinin elde ettiği sonuçlar mutlak hataya göre incelendiğinde hata oranının 0,5 olduğu tespit edilmiştir. Bu kapsamda senaryo 1 özelinde yapılan karşılaştırmalara göre Yapay Sinir Ağı mimarisinin TuringEQ mimarisine göre aynı iterasyon ve öğrenme katsayıları ile daha yüksek bir hata oranına sahip olduğu görülmektedir.

Yapay Sinir Ağı mimarisinin senaryo 2 için 0,05 öğrenme katsayısı ve 100 iterasyon eğitim sonucunda mimarinin eğitim sonrası tahminleri Tablo 10’da gösterilmektedir.

Tablo 10. Yapay Sinir Ağı mimarisi senaryo 2 için eğitim sonrası tahminleri

X	Y	Beklenen Sonuç	Tahmin Edilen Sonuç	Hata
0	0	0	0,4568027	-0,4568027
0	1	1	0,45977562	0,54022438
1	0	1	0,46615469	0,53384531
1	1	0	0,48125098	-0,48125098

Yapay Sinir Ağı mimarisinin senaryo 2 için XOR problemi üzerinde eğitim işleminin 100 iterasyon tekrarı için gerçekleştirilmesi sonucunda XOR tahminlemesinin doğru bir şekilde yapılabilmesi için iterasyon tekrarının yetersiz düzeyde kaldığı görülmektedir. Eğitim sonrasında mimarinin elde ettiği sonuçlar mutlak hataya göre incelendiğinde hata oranının 0,5 olduğu tespit edilmiştir. Bu kapsamda senaryo 2 özelinde yapılan karşılaştırmalara göre Yapay Sinir Ağı mimarisinin TuringEQ mimarisine göre aynı iterasyon ve öğrenme katsayıları ile daha yüksek bir hata oranına sahip olduğu görülmektedir.

TuringEQ mimarisinin belirlenmiş olan senaryolar üzerinden XOR veri seti üzerindeki başarımının doğrulanmasının ardından Yapay Sinir Ağlarının aynı veri seti üzerinde göstermiş olduğu başarımlar incelendiğinde iki Yapay Zekâ mimarisi arasında yapılan karşılaştırmalara göre TuringEQ mimarisinin daha düşük iterasyon sayılarında başarılı sonuçlara ulaştığı görülürken, Yapay Sinir Ağlarının daha yüksek iterasyon sayılarında başarılı sonuçlara ulaştığı görülmüştür. Yapılan analizlerde iki mimari çalışma hızı açısından karşılaştırıldığında TuringEQ mimarisinin, mimari içerisindeki gradyan geri düzeltme mekanizmasından kaynaklı olarak bir iterasyon için Yapay Sinir Ağlarından daha düşük hızlarda çalıştığı görülmüştür.

TuringEQ mimarisinin başlangıç ağırlık değerlerinin veri setinin aralık genişliğine göre belirlenmesi mimarinin başarımını etkileyen önemli unsurlardan biridir. Ağırlık değerlerinin rastgele olarak atanması mimarinin hesaplama karmaşıklığını arttırmakta ve bazı durumlarda öğrenmeyi olumsuz olarak etkilemektedir. Bu kapsamda, ağırlıkların belirli bir stratejiye göre atanması mimarinin öğrenmesine katkı sağlayacaktır.

6. Sonuç ve Öneriler

Yapılan çalışmada geliştirilen TuringEQ mimarisinin doğrusal olmayan bir problem olan XOR problemi üzerinde test edilmesi sonucunda olumlu sonuçlar elde edilmiştir. Doğrusal olmayan bir problem üzerinde TuringEQ mimarisi, 100 iterasyon sonucunda farklı katman derinlikleri ve öğrenme katsayıları ile olumlu sonuçlar göstererek XOR problemi için gerekli tahminlemeleri doğru bir şekilde gerçekleştirmiştir.

Elde edilen başarımların, oluşturulmuş olan mimarinin XOR problemleri üzerinde yüksek bir başarımla çalıştığını kanıtlamış ve temel düzeyde doğrusal olmayan karakteristiğe sahip problemler için temel mimarinin uygun bir tasarıma sahip olduğu görülmüştür.

Çalışma sonucunda elde edilen kazanımlar şu şekilde sıralanabilir:

- Yeni bir Yapay Zekâ Mimarisi olan TuringEQ mimarisi geliştirilmiştir.
- TuringEQ mimarisinin doğrusal olmayan problem karakteristiklerine uygun olduğu kanıtlanmıştır.
- TuringEQ mimarisinin farklı parametrik değişimler için çalıştırılması sonucunda elde edilen başarımlar olumlu düzeydedir.
- TuringEQ mimarisinin ileri düzey çalışmalarda geliştirilebilecek temel tasarımının doğrulanması yapılmıştır.

Yapay Sinir Ağları ile yapılan karşılaştırmalar sonucunda, TuringEQ mimarisinin her ne kadar daha düşük iterasyonlarda başarılı sonuçlara ulaştığı görülmüş olsa da mimari içerisindeki gradyan geri düzeltme mekanizmasından kaynaklı olarak, iterasyon başına Yapay Sinir Ağlarından daha düşük hızlarda çalıştığı görülmüştür.

Yapılan çalışma sonucunda TuringEQ mimarisi elde edilerek temel bir doğrusal olmayan problem olan XOR problemi üzerinde olumlu sonuçlar alınsa da elde edilen bu mimarinin karmaşık ve büyük doğrusal olmayan problemler üzerindeki başarımı analiz edilmemiştir.

İlerleyen çalışmalarda TuringEQ mimarisinin büyük ve daha karmaşık veri kümeleri üzerinde çalıştırılması sonucunda elde edilen sonuçlar analiz edilerek mimari düzeyde iyileştirme çalışmaları yapılması hedeflenmektedir.

Ayrıca yatay katmanlar içerisindeki hata onarımında türev yaklaşımının yanı sıra, yatay katman bilinmeyen değişkenlerinin derecelerinin ve birbirleri ve diğer katman değişkenleri ile ilişkilerinin ele alındığı yeni bir onarım algoritması tasarlanması düşünülmektedir. Ek olarak tasarlanacak olan bu onarım algoritması ile iterasyon başına çalışma hızlarının daha düşük seviyelere indirgenmesi, mevcut algoritmanın hız noktasındaki verimsizliğinin çözülmesi ve mevcutta kullanılan Yapay Sinir Ağı gibi Yapay Zekâ mimarilerine hız olarak avantaj sağlanabilmesi amaçlanmaktadır.

Teşekkür

H. E. O., fikir sahibi olup çalışmayı gerçekleştirdi, yorumladı ve makaleyi yazdı. M.B., çalışmayı yorumladı.

Kaynaklar

- [1] Moor JH, Turing test, Encyclopedia of Computer Science, ss. 1801-1802, Oca. 2003, Erişim: 05 Haziran 2024. [Çevrimiçi]. Erişim adresi: https://dl.acm.org/doi/abs/10.5555/1074100.1074882#pill-authors__contentcon
- [2] Abitha J. Artificial Intelligence Technology and its Challenges-A Review, Journal of excellence in Computer Science and Engineering, c. 2, sy 1, ss. 11-18, Şub. 2016.
- [3] Abraham. A. H. of measuring system design 2005, "Artificial neural networks", wsc10.softcomputing.net, Erişim: 05 Haziran 2024. [Çevrimiçi]. Erişim adresi: http://wsc10.softcomputing.net/ann_chapter.pdf
- [4] Silva IN, Spatti DH, Flauzino RA, Liboni LHB, Alves SFR. Artificial Neural Networks: A Practical Course, Illustrated. Springer International Publishing, 2016, 2016.
- [5] Asadollahfardi G. "Artificial Neural Network", ss. 77-91, 2015.
- [6] Fukushima K. "Biological Cybernetics Neocognitron: A Self-organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position", Biol. Cybernetics, c. 36, s. 202, 1980.
- [7] LeCun Y, Bottou L, Bengio Y, Haffner P. "Gradient-based learning applied to document recognition", Proceedings of the IEEE, 1998, Erişim: 05 Ağustos 2024. [Çevrimiçi]. Erişim adresi: <https://ieeexplore.ieee.org/abstract/document/726791/>
- [8] Krizhevsky A, Sutskever I, Hinton GE., "ImageNet classification with deep convolutional neural networks", Commun ACM, c. 60, sy 6, ss. 84-90, Haz. 2017.
- [9] Simonyan K, Zisserman A. "Very Deep Convolutional Networks for Large-Scale Image Recognition", 3rd International Conference on Learning Representations, ICLR 2015- Conference Track Proceedings, Eyl. 2014, Erişim: 05 Ağustos 2024. [Çevrimiçi]. Erişim adresi: <https://arxiv.org/abs/1409.1556v6>
- [10] Szegedy C vd. "Going Deeper with Convolutions", Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, c. 07-12-June-2015, ss. 1-9, Eyl. 2014.
- [11] He K, Zhang X, Ren S, Sun J. "Deep Residual Learning for Image Recognition", Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, c. 2016-December, ss. 770-778, Ara. 2015.
- [12] Termritthikun C, Umer A, Suwanwimolkul S, Xia F, Lee I. "SalNAS: Efficient Saliency-prediction Neural Architecture Search with self-knowledge distillation", Eng Appl Artif Intell, c. 136, s. 109030, Eki. 2024.
- [13] Yang FJ. "An implementation of naive bayes classifier", Proceedings - 2018 International Conference on Computational Science and Computational Intelligence, CSCI 2018, ss. 301-306, Ara. 2018.
- [14] Joshi AA, Aziz RM. "Deep learning approach for brain tumor classification using metaheuristic optimization with gene expression data", Int J Imaging Syst Technol, c. 34, sy 2, s. e23007, Mar. 2024.
- [15] Rosso MM, Aloisio A, Randazzo V, Tanzi L, Cirrincione G, Marano GC. "Comparative deep learning studies for indirect tunnel monitoring with and without Fourier pre-processing", Integr Comput Aided Eng, c. 31, sy 2, ss. 213-232, Oca. 2024.
- [16] Aslan M, Baykara M, Alakus TB. "LieWaves: dataset for lie detection based on EEG signals and wavelets", Med Biol Eng Comput, c. 62, sy 5, ss. 1571-1588, May. 2024.
- [17] Steps to Solve a Linear Programming Problem | Superprof. Erişim: 05 Haziran 2024. [Çevrimiçi]. Erişim adresi: https://www.superprof.co.uk/resources/academic/math/linear-algebra/linear-programming/steps-to-solve-a-linear-programming-problem.html#chapter_steps-to-solve-a-linear-programming-problem
- [18] Nonlinear programming- Wikipedia. Erişim: 05 Haziran 2024. [Çevrimiçi]. Erişim adresi: https://en.wikipedia.org/wiki/Nonlinear_programming
- [19] Benderskaya EN. Nonlinear Trends in Modern Artificial Intelligence: A New Perspective, Beyond Artificial Intelligence, ss. 113-124, 2013.
- [20] Demystifying the XOR problem- DEV Community. Erişim: 05 Haziran 2024. [Çevrimiçi]. Erişim adresi: <https://dev.to/jbahire/demystifying-the-xor-problem-1blk>