

Konuşma Duygu Tanıma Uygulamalarında Hiper Parametre Optimizasyonu ile Derin Öğrenme Metotlarının Geliştirilmesi

Cevahir PARLAK^{1*} 

Öz

Bu çalışmada derin öğrenme uygulamalarında oldukça yeni ve önemli bir aşama olan hiper parametre ayarlama metotlarının bir karşılaştırılması verilecektir. Veriseti olarak yeni duygu verisetlerinden NEMO duygusal konuşma veriseti kullanılacak olup, KerasTuner ile CNN, LSTM ve DNN modelleri Rassal arama, Hiperkomite ve Bayesçi optimizasyon metotları kullanılarak karşılaştırılacaktır. Genel olarak makine öğrenmesi ve özellikle de derin öğrenme uygulamalarında başarılı bir model üretebilmek zaman ve hesaplama gücü açısından oldukça pahalı ve zorlu bir işlem olarak araştırmacıların karşısına çıkmaktadır. Hiper parametre optimizasyonunun genel olarak iki temel aşamadan oluştuğu kabul edilebilir. Birinci aşamada öncelikle değişkenlik gösteren parametrelerin alabilecekları değerlere dayalı bir arama uzayı belirlenir. Bu parametreler öğrenme katsayısı, nöron sayısı, katman sayısı, aktivasyon fonksiyonu ve benzeri değişkenler olabilir. İkinci aşama ise bu parametreleri kullanarak yapay zekâ modellerini oluşturur ve belirlenen bir başarı kriterine göre test eder. Optimizör bu modelleri çalıştırırken işlemi hızlandırmak için değişik algoritmalar kullanılabilir. Hiper parametre optimizasyon uygulamaları bu konuda gün geçtikçe daha iyi çözümler sunmakta ve insan faktörünü kademeli olarak aradan çıkarmaktadırlar. Izgara arama mevcut bütün konfigürasyonları bütün kaynakları sonuna kadar tüketerek çalıştırırken, Rasgele arama ise mevcut kümeden tesadüfi olarak seçilen belli konfigürasyonları dener. Rassal arama her ne kadar bütün olası konfigürasyonları denemese bile genellikle Izgara aramaya yakın sonuçlar üretebilmektedir. Ardışık arama, Asenkron Ardışık arama, Populasyon-Tabanlı Eğitim, Hiperkomite ve Bayesçi yaklaşımlarda diğer hiper parametre optimizasyon metotları arasında sayılabilir. Bu çalışmada NEMO konuşma duygu veriseti 4 duygu ile CNN, LSTM ve DNN derin öğrenme sınıflandırıcılarıyla çalıştırılmış ve KerasTuner'in Rassal Arama, Bayesçi Arama ve Hiperkomite Arama metotlarıyla otomatik üretilen metotların performansları karşılaştırılmıştır. Hiper parametre optimizasyon metotlarından Bayesçi Optimizasyon metodunun diğerlerine göre daha iyi ve hızlı sonuçlar ürettiği görülmüştür.

Anahtar Kelimeler: Konuşma duygu tanıma, Derin öğrenme, Hiper parametre optimizasyonu.

Improving Deep Learning Models Using Hyper Parameter Optimization in Speech Emotion Recognition

Abstract

In this research article, a comparison of newly introduced hyper parameter optimization methods is presented. We used NEMO emotional speech dataset which is a very novel dataset with 4 emotional classes. KerasTuner is used to compare the performances of CNN, DNN, and LSTM classifiers by running the Random Search, Bayesian Search and HyperBand Search to optimize the model hyper parameters. Broadly speaking, it is a very arduous, expensive, time consuming and formidable task for the researchers to produce a successful model for machine learning applications, namely deep learning applications. Hyper-parameter optimization consists of two main steps. The first step constructs a search space setting the values of available parameters. Learning rate, number of hidden neurons, layers, activation functions are among the most notable hyper parameters. The Second step involves generating the best machine model using these parameters and specified performance metrics. Optimizers may use different algorithms while running these models. Hyper parameter optimization applications are presenting better solutions in this field day by day and are eliminating the human factor in-between gradually. Grid search runs all possible parameter sets consuming all available resources. On the other hand, Random search runs only some randomly selected configurations. Although Random search does not try all possible actions it can reach the same level of performance. Successive Halving, Asynchronous Successive Halving, Population Based Training, Hyperband, Bayesian approaches are among the prominent optimization methods. This study uses NEMO emotional speech dataset with 4 emotions using CNN, DNN, and LSTM models. Models are generated using Random Search, Bayesian Search, and Hyperband Search methods of KerasTuner and the results are compared. Experiments show that Bayesian Search outperforms Random and Hyperband search algorithms.

Keywords: Speech emotion recognition, Deep learning, Hyperparameter optimization.

¹Fenerbahçe Üniversitesi Bilgisayar Mühendisliği Bölümü, İstanbul, Türkiye, cevahir.parlak@fbu.edu.tr

*Sorumlu Yazar/Corresponding Author

Geliş/Received: 01.07.2024

Kabul/Accepted: 18.11.2024

Yayın/Published: 15.12.2024

1. Giriş

Makine öğrenmesi, yapay zekâ ve bilhassa son zamanlarda derin öğrenme uygulamaları beklenmedik gelişmelerin önünü açmaktadır. Bu uygulamaların geliştirilmesi oldukça fazla zaman, makine gücü ve insan çabası gerektirmektedir. Zira bu uygulamalar verisetleri, algoritmalar, yapay zekâ modelleri, aktivasyon fonksiyonları ve benzeri pek çok parametreyi içlerinde barındırmaktadırlar. Bu parametreler en genel olarak model parametreleri ve hiper parametreler olarak ikiye ayrılabilir. Model parametreleri modele özgü sabit ve katsayıları (nöron ağırlıkları, sabitler), hiper parametreler ise öğrenme katsayısı, momentum, demet sayısı, nöron sayısı, çevrim sayısı, eğitim ve doğrulama seti oranı, aktivasyon fonksiyonları, katman sayısı, çekirdek boyutu ve nöron seyreltme (dropout) gibi interaktif ve başlangıçta belli bir değer alan değişkenleri içermektedir. Bunun yanı sıra başarımları kriteri olarak eğitim kayıp fonksiyonu, doğrulama kayıp fonksiyonu, başarı oranı, eğri altındaki alan (Area Under Curve), F1 skoru veya diğer herhangi bir metrik kullanılabilir. Model parametreleri modele göre çok büyük farklılıklar gösterebilmektedir. ResNet-101'de [He ve ark., 2016] yaklaşık 45 milyon model parametresi olmasına karşın GPT-4'teki [Achiam ve ark., 2023] model parametre sayısı 1,76 trilyon kadardır. Bu parametreler eğitim aşamasında kayıp fonksiyonlarını minimize edecek ve başarı oranlarını maksimize edecek şekilde kalibre edilirler. Yalnız bu noktada dikkat edilmesi gereken nokta şudur ki model hatasının veya kayıp fonksiyonunun küçük olması her zaman en yüksek başarı oranını sağlayacağı anlamına gelmemektedir. Bazı durumlarda kayıp oranı yüksek olan modeller düşük olan modellerden daha yüksek başarı oranlarına ulaşabilmektedirler. Zira modeller eğitildikten sonra daha önce hiç görmedikleri bir test verisi üzerinde denemeye maruz bırakılmaktadırlar ve bu veri üstündeki başarı eğitim ve doğrulama verisi üzerindeki başarıdan çok daha farklı olabilmektedir. Burada aşırı öğrenme veya ezberleme ile yetersiz öğrenme arasındaki ince çizgiye azami dikkat edilmelidir. Model ağırlıkları yapay sinir ağlarındaki bağlantıların önemini ve etkisini gösteren temel parametrelerdir. Eğitim aşamasında ağırlıklar giriş verisi ile çarpım işlemine tabi tutulduktan sonra bir aktivasyon fonksiyonu ile nöronun çıkış verisi ortaya çıkar. Sabitler ise aktivasyon fonksiyonunu orijinden öteleyerek yapay sinir ağının daha esnekliğini ve eğitim aşamasında daha karmaşık yapıları öğrenebilmesini sağlar. Ağırlıklar ve sabitler yapay sinir ağlarının temel yapıtaşlarıdır ve modelin başarılı olabilmesi için çok iyi kalibre ve optimize edilmeleri gereklidir.

Hiper parametreler modelin başarısını çok büyük ölçüde etkilediğinden ayarlamalarında azami çaba ve özen gösterilmelidir. Bu parametrelerin seçiminde pek çok faktör etkili olmaktadır ve genel bir kural belirlemek mümkün değildir. Örnek olarak katman sayısını artırmak veya nöron sayısını arttırmak bazen performansı iyileştirebildiği gibi bazen de olumsuz sonuçlar üretebilmektedir. Dahası eğitim aşamasında modelin başarılı olması test aşamasında da benzer başarıyı gösterebileceği

anlamına gelmemektedir. Bu parametreler sadece kendi barındıkları modele değil kullandıkları veri setine göre de farklı sonuçlar üretebilmektedirler. Bir verisetinde çok yüksek başarı sergileyen bir model başka bir verisetinde çok düşük sonuçlar üretebilmektedir. Farklı verisetleri üstünde yapılan çapraz deneyler bu konuda hala önemli bir genelleştirme sorunu olarak çözülmeyi beklemektedirler. Optimal parametreleri bulabilmek araştırmacıların çok yoğun, meşakkatli ve uzun çabasını gerektiren bir alan olarak halen devam etmektedir. Araştırmacılar derin öğrenme ağlarını otomatize etme konusunda oldukça istekli ve gayretli olmalarına rağmen bu alanda tamamlanmayı bekleyen hala çok iş bulunmaktadır. Veri toplama, veri işleme, öznelik bulma ve seçme, deney tasarımı, model tasarımı ve seçimi, model değerlendirme, kaynak ayırma ve hiper parametre optimizasyonu (HPO) otomatize ve optimize edilmesi gereken konular arasındadır. Elbette ki bu konuların tamamının bir anda otomatize edilmesi beklenemez ancak uzun vadeli bir araştırma hedefi olmak üzere bu alanların her birinin kısmen otomatize edilerek zamanla insan katkısını yavaş yavaş aradan çıkarmak hedeflenebilir. Bu sayede derin öğrenme çalışmaları daha düzenli, hızlı ve etkili bir şekilde yürütülebilir. Makine öğreniminin otomatikleştirilmesi başka alanlardaki insanların ve araştırmacıların bu metotları kendi alanlarına uygulayabilmelerini de kolaylaştıracaktır. Makine öğrenmesi veya daha özel bir ifadeyle derin öğrenme konusunda yeterli kaynak ve bilgiye sahip olmayan kişilerde makine öğrenmesini kendi alanlarında uygulayabilecektir. Otomatik makine öğrenmesi herkesin kolayca ulaşıp başarılı sonuçlar elde edebileceği bir araç kutusu haline getirilebilecektir. Böylece insanların hayatını daha da kolaylaştırıyor olacağı gibi yapılan çalışmalarda da daha iyi sonuç ve performans elde etmek mümkün olacaktır. Son zamanlarda büyük şirketler otomatik makine öğrenmesi alanına büyük ilgi göstermekte ve kullanıcılarına bu yönde hizmetler sunarak hem zaman hem işgücü hem de parasal tasarruf sağlamaktadırlar [Hutter ve ark., 2019]. Otomatik makine öğrenmesinin temel hedefleri aşağıdaki gibi özetlenebilir:

- 1- İnsan çabasını mümkün mertebe azaltmak,
- 2- Makine öğrenme performansını arttırmak,
- 3- Mümkün olan en basit, en hızlı ve parametre sayısı en az modeli elde etmek,
- 4- Kaynak kullanımını minimuma indirmek,
- 5- Zaman tasarrufu sağlamak,
- 6- Farklı branşlarda yapay zekâ konusunda yeterli bilgi ve kaynağı olmayanlara da yapay zekâyı kullanabilme imkânı sağlamak,
- 7- Deneysel sonuçların doğruluk, güvenilirlik ve tekrar üretilebilirliğini geliştirmek.

Bu amaçlara ulaşmak için otomatik derin öğrenmenin önünde pek çok engel bulunmaktadır. Bunlar arasında verisetlerindeki düzensizlik ve yetersizlikler, kayıp fonksiyonlarının elverişli olmayışı, konfigürasyon kümesinin aşırı karmaşık ve büyük boyutta oluşu ve geniş aralıklara dağılmış olması, büyük modellerde büyük verisetlerinde fonksiyon değerlendirmelerinin çok

maliyetli oluşu sayılabilir. Hiper parametre optimizasyon işlemi verilen bir veriseti üstünde aşağıdaki gibi formüle edilebilir.

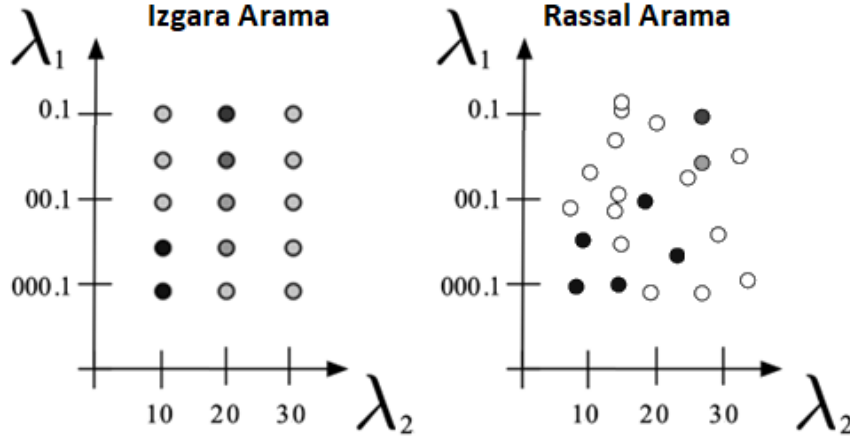
$$\lambda^* = \underset{\lambda \in \mathbb{C}}{\operatorname{argmin}} \mathbb{E}_{(D_{\text{eğitim}}, D_{\text{doğrulama}}) \sim \mathbb{D}} V(\mathcal{L}, A_\lambda, D_{\text{eğitim}}, D_{\text{doğrulama}})$$

Burada $V(\mathcal{L}, A_\lambda, D_{\text{eğitim}}, D_{\text{doğrulama}})$ kayıp fonksiyonunu temsil etmektedir. Kayıp fonksiyonu λ parametrelerine sahip A algoritması tarafından üretilmektedir. A algoritması, $D_{\text{eğitim}}$ verisi üzerinde eğitilmekte ve $D_{\text{doğrulama}}$ verisi üzerinde doğrulanmakta ve eğitim aşamasının sonunda bir D_{test} verisi üzerinde test edilmektedir. $D_{\text{eğitim}}$, $D_{\text{doğrulama}}$ ve D_{test} verileri, \mathbb{D} verisinden elde edilmektedir. \mathbb{C} ise hiper parametre kümesini temsil etmektedir.

Bu çalışmada hiper kalibrasyon ile ilgili güçlü bir arka plan oluşturulmaya çalışılacaktır. Yapay sinir ağlarında öğrenme oranı ve düzenleme, konvolusyonel yapay sinir ağlarında katman sayısı, nöron sayısı, filtre sayısı, karar ağaçlarında ağaç ayırma kriterleri (Gini indeksi, information gain vb.), rasgele ormanlarda karar ağacı sayısı, özellik uzayı, örnekleme yüzdeleri, SVM’lerde doğrusal, polinomsal veya RBF çekirdek kullanımı gibi pek çok farklı hiper parametre bulunmaktadır. SVM’ler özellikle doğrusal olmayan çekirdek kullanıldığı zaman hiper kalibrasyona oldukça fazla ihtiyaç duyarlar; zira SVM’lerde hem doğru çekirdeği hem de bu çekirdekle ilgili doğru katsayı ve sabitleri (derece ve gamma) bulamazsanız başarı oranınızda ciddi kayıplar yaşamanız doğaldır. Hiper kalibrasyonda çok sık kullanılan metotlardan ikisi Rassal arama ve Izgara arama metotlarıdır. Izgara arama önce mümkün bütün hiper parametrelerin bir kümesini oluşturur ve küme elemanlarının hepsi için bir model çalıştırarak en yüksek başarı oranına sahip modeli seçerken, Rassal arama ise önce bütün hiper parametrelerden oluşan kümeyi oluşturur ancak bunlar içinden sadece rastgele belli bir adedine ait modelleri çalıştırarak en iyi modeli keşfetmeye çalışır. Her ne kadar Rassal arama, Izgara arama gibi bütün olasılıkları deneyen tam kapsamlı bütün kaynakları tüketen bir arama değilse de çok daha hızlıdır ve genellikle tipik olarak benzer başarı oranlarına ulaşmakta zorluk çekmez. Bu durum her ne kadar sağduyuya aykırı gözükse de bunun nedeni herhangi bir yapay zekâ problemi için tek bir altın parametre seti olmamasıdır. En iyi başarı oranını veya oldukça yakın değerleri verebilen genellikle birden çok parametre seti bulunur ve bu setler içindeki parametreler belli bir dağılım içinde olurlar. Bu doğal dağılım ortamının içine girebilmeyi tesadüfi olarak da olsa başarabildiğimizde tam kapsamlı bir Izgara arama olmaksızın da yüksek başarı oranlarına ulaşmak mümkündür. Izgara arama ve Rassal aramanın karşılaştırmasını Şekil 1.’de verilmektedir [Bischl ve ark., 2023; Vincent ve Jidesh, 2023].

Hiper optimizasyonla ilgili çalışmalar gün geçtikçe yaygınlaşmaktadır ve araştırmacılar farklı kütüphaneler kullanarak bu alanda başarılı sonuçlar elde etmeye çalışmaktadırlar. Wu [Wu ve ark., 2022] konuşma duygu tanıma için bir yapay sinir ağı arama modeli önermiş ve IEMOCAP veri seti üzerinde 4 duygu ile deneyler gerçekleştirmişlerdir. KYSA, RNN ve Kapsül yapay sinir ağlarıyla

yaptıkları deneylerde hiper kalibrasyonun başarısını sergilemişlerdir. KYSA ve Kapsül network ile %70,54 ağırlıklı başarı oranı, %56,94 ağırlıksız başarı oranlarına ulaşabilmişlerdir.



Şekil 1. Izgara arama ve Rassal arama metotlarının temsili gösterimi. Burada λ_1 ve λ_2 kalibre edilecek hiper parametreleri (öğrenme katsayısı, çevrim sayısı vb.) göstermektedir

Farklı yapay sinir ağlarının bir karşılaştırması Gökalp [Gökalp ve Aydın, 2021] tarafından verilmiştir. Gökalp ve Aydın, RAVDESS [Livingstone ve Russo, 2018] ve TESS [Pichora-Fuller ve ark., 2020] verisetleri üzerinde yaptıkları çalışmada en iyi sonuçları SqueezeNet ile elde ederken en düşük sonuçlar MobileNet ile gelmiştir.

Sun [Sun ve ark., 2022] IEMOCAP veriseti üzerinde iki akışlı bir yapay sinir ağı arama kütüphanesi olan EmotionNAS'ı sunmaktadırlar. IEMOCAP veri setinin doğaçlama kısmındaki kızgın, üzgün, mutlu ve nötr duygu türlerinden oluşan örneklerle CNN-GRU, CTC-RNN, ResNet sınıflandırıcılarıyla yaptıkları deneylerde %69,1 ağırlıksız başarı oranlarına ulaşabilmişlerdir. Bu oran ResNet ile elde edilen %59,1 başarı oranına göre oldukça yüksek bir ilerlemeyi ifade etmektedir. Öznitelik olarak ses sinyallerinden Mel spektrogramlarını ve Wav2Vec akustik öznitelikleri kullanmışlardır. EmotionNAS sadece başarı oranını arttırmakla kalmamış modelin parametre sayısını da oldukça azaltmıştır.

Rajapakshe [Rajapakshe ve ark., 2024] konuşma duygu tanıma uygulamaları için emoDARTS hiper kalibrasyon kütüphanesini önermişler ve IEMOCAP, MSP-IMPROV, MSP-Podcast veri setleri üzerinde deneyler gerçekleştirmişlerdir. Deneylerinde KYSA ve LSTM sınıflandırıcıları birleştirerek bir CNN-SeqNN sinir ağı modeli geliştirmiş ve emoDARTS ile 128 MFCC spektrogramlarını öznitelik olarak kullanarak oldukça yüksek başarı oranlarına sahip sonuçların elde edilebileceğini göstermişlerdir. EmoDARTS, IEMOCAP verisetinde $76,55 \pm 4,03$, MSP-IMPROV verisetinde $65,63 \pm 8,85$, MSP-Podcast veri setinde ise $61,15 \pm 2,41$ ağırlıksız başarı oranları ile en yüksek sonuçları elde edebilmiştir ve taban modelin %53,55 olan başarı oranına göre oldukça büyük bir ilerleme sağlamıştır.

2. Hiper Parametre Optimizasyon Metotları

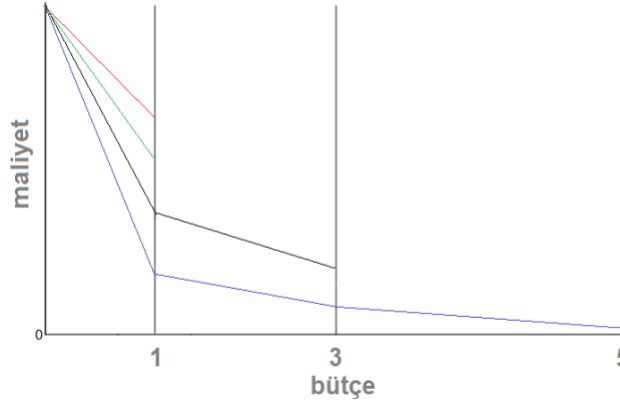
Bu bölümde hiper kalibrasyon metotlarından genel hatlarıyla bahsedilecektir. Hiper ayarlama işlemine başlamadan önce mukayese yapabileceğimiz bir taban metoda ihtiyacımız vardır. Bu taban metodu ölçüt olarak uygulayacağımız hiper ayarlamamızın etkisini olumlu veya olumsuz test edebilme imkânımız olacaktır. Izgara arama ve Rassal arama ile hiper ayarlama bir başlangıç yapabiliriz. Hiper ayarlama tamamlandıktan sonra bakmamız gereken üç önemli bilgi çıktısı bulunmaktadır. Bunlardan ilki performans, ikincisi zaman ve üçüncüsü de hiper parametrelerdir. Mümkün olan en iyi performans, mümkün olan en az sayıda parametre ile en hızlı bir şekilde elde edilebilmelidir. Hiper ayarlama yaparken parametre aralıklarını belirlemek için kullanılacak yöntemler aşağıdaki gibi sıralanabilir:

1. Tahmin
2. Tecrübe
3. Algoritma

Yapay sinir ağlarında hiper ayarlama yapılabilecek parametreler nöron sayısı, katman sayısı, öğrenme oranı, seyreltme oranı, demet sayısı, çevrim sayısı, aktivasyon fonksiyonu, çekirdek sayısı, filtre sayısı, adım sayısı, toplama sayısı olarak sayılabilir. Hiper ayarlama başarı oranlarında %30'a varabilen olağanüstü sıçramalar sağlayabilir. Rassal arama, Izgara aramaya benzer başarı oranlarını yakalayabildiği gibi zaman olarak da oldukça düşük sürelerde tamamlanabilmektedir. Rassal arama ve Izgara aramaya ilave olarak daha yeni ve gelişmiş hiper ayarlama algoritmaları olarak Bayesçi optimizasyon [Brochu ve ark., 2010] ve hiperkomite [Li ve ark., 2018] kullanılabilir. Bu kalibrasyon metotlarında belli bir süre boyunca yeterli ilerleme göstermeyen modeller erken durdurma ile kısa devre sonlandırılarak zaman tasarrufu sağlanabilmektedir.

Yeni hiper ayarlama metotları parametre kümesini daraltmaktan daha çok konfigürasyon çalıştırmanın hızlandırılmasına odaklanarak zayıf ilerleme gösteren metotları hızlı bir şekilde elimine etmek prensibine dayanmaktadırlar. Bu metotlar adaptif hesaplama, ümit verici parametre kümelerine daha fazla kaynak ayırma gibi yaklaşımları kullanmaktadırlar. Kullanılan kaynaklar öznitelik sayısı, eğitim veriseti hacmi gibi farklı formlarda ortaya çıkabilmektedir. Bayesçi optimizasyonu adaptif kaynak ayırma ile buluşturan bazı metotlar bulunduğu gibi hiperkomite optimizasyon metodu da Rassal arama metodunu adaptif kaynak atamayla geliştirmeyi hedeflemektedir. Bayesçi optimizasyon da hedef fonksiyon tahmininin oldukça maliyetli olmasından dolayı buna bir çare olarak çok değişkenli (Multi-Fidelity) optimizasyon metotları önerilmiştir [Li ve Li, 2024]. Çok değişkenli optimizasyon metotları arasında Ardışık Yarılama (Successive Halving, SH) [Jamieson ve Talwalkar, 2016], Hiperkomite (HyperBand) ve Bayesçi Optimizasyon Hiperkomite (Bayesian Optimization HyperBand, BOHB) [Falkner ve ark., 2018] gösterilebilir. Hiperkomite optimizasyon problemleri

genel olarak kara kutu (Blackbox) ve çok deęişkenli (Multi-Fidelity) olarak ikiye ayrılabilir. Izgara arama, Rassal arama, Bayesçi optimizasyon, Simülasyonlu kaynaştırma, Genetik algoritmalar karakutu problemlerinde kullanılırken, modern öğrenme eğrileri, komite temelli, ardışık yarılama, hiperkomiteli Bayesçi optimizasyon ise çok deęişkenli hiperparametre optimizasyon yöntemleridir. Ardışık yarılama en umut verici metotlara bütçeden en fazla pay vermeyi amaçlar. Ardışık yarılama tüm mümkün parametre konfigürasyonlarının erken sonlandırılabilceğini varsayar ve bütçenin belli aşamalarında mevcut konfigürasyonun en kötü yarısını eler ve bu şekilde tek bir küme kalıncaya kadar devam eder ve en son kalan kümeyi bütçenin sonuna kadar çalıştırır. Ardışık yarılama başlangıçta seçilecek küme sayısı ve her aşamada ne kadarının elimine edileceği bir problem olarak karşımıza çıkmaktadır. Hiperkomite buna bir çözüm olarak sunulmaktadır. Hiperkomite, Ardışık Yarılama metodunu farklı bütçelerle sıklıkla uygulayarak en iyi konfigürasyonu bulmayı amaçlar. Ardışık yarılamanın temsili bir gösterimi Şekil 2’de verilmektedir.



Şekil 2. Ardışık yarılama metodu her adımda model sayısını başarı metriğine baęlı olarak belli bir oranda azaltır ve bütçeden ayrılan payı belli bir oranda arttırır.

Hiper optimizasyon günümüzde paralel çalışmayı çok büyük oranda kullanan yeni bir model olarak karşımıza çıkmaktadır. Özellikle de Rassal ve Izgara aramanın paralel hale getirilebilmesi çok basit bir konu olduğu için araştırmacılar bu metotların paralelleştirilmesini geliştirmeye çalışmaktadırlar. Bu bağlamda asenkron erken sonlandırma yaklaşımları olumlu sonuçlar verebilmektedir. Ardışık yarılama ilk başta bütün konfigürasyonlara aynı benzer bütçeler ayrılarak başlar. Bütün performanslar değerlendirilir ve en üstteki konfigürasyonları bir sonraki basamağa gönderir ve bir önceki adımdaki bütçeyi belli bir oranda (2 katı vb.) arttırır. Ardışık arama, performans ve zaman açısından Rassal aramayı geride bırakabilir. Asenkron Ardışık arama (Asynchronous Succesive Halving Algorithm, ASHA), senkron ardışık aramanın zayıf taraflarını alttan yukarıya doğru büyütür ve konfigürasyonları mümkün oldukları zamanda bir üst seviyeye yükselterek çözmeye çalışır. Bunları yaparken de geniş bir set olmadan başlar ve aramayı mümkün olduğunca daraltmayı amaçlar. ASHA algoritmasında her çalışan en alt basamakta yeni bir

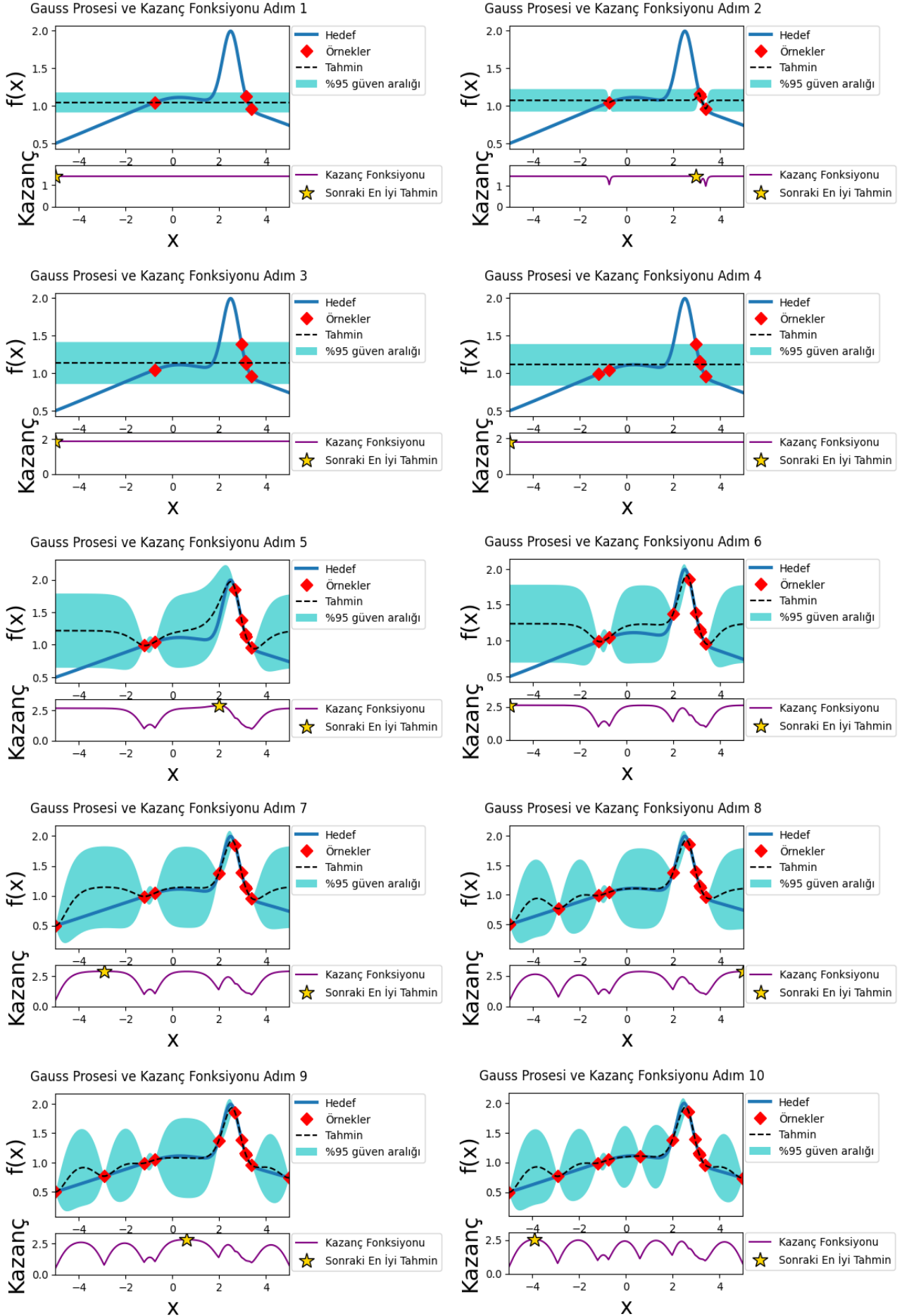
konfigürasyon ekler ve işi bitince yeni bir iş talep eder. Bu esnada yukarıdan aşağıya doğru basamaklara bakılır ve belirli basamaklarda bir üst basamağa eklenebilecek bir konfigürasyonun olup olmadığı kontrol edilir. Eğer yoksa çalışan en alt basamağa yeni bir konfigürasyon ekleyip basamağı genişletir; öyle ki daha fazla konfigürasyon üst basamağa terfi ettirilebilsin [Bischi ve ark., 2023]. ASHA'da denemeler herhangi bir zamanda bir sonraki basamağa ilerleyebilir. Bunun için denemelerin ilerlemesi ve adım tamamlamanın birbirinden ayrıştırılmış olması gerekir. SHA ve ASHA'daki önemli bir varsayım, bir deneme kısa bir süre içinde başarılı oluyorsa daha uzun sürelerde de başarılı olacağı düşüncesidir. Ancak bunu geçersiz kılabilecek bir örnek olarak öğrenme oranındaki ayarlama verilebilir. Genel bir kaide olarak büyük öğrenme oranları küçüklere nazaran daha kısa sürede daha yüksek başarı oranlarına ulaşabilmektedirler. Bu durum küçük öğrenme oranları ile yapılan kalibrasyon denemelerinin hatalı bir şekilde sonlandırılmasına neden olabilmektedir.

Karakutu optimizasyon problemlerinde sıklıkla başvurulan yöntemlerden birisi de Bayeşçi yaklaşımlardır ve SMBO (Sequential Model Based Optimization) ile formüle edilir. Bayeşçi optimizasyon (BO), optimize edilecek parametre kümesi içinden seçilen bir kısmını çalıştırdıktan sonra elde edilen sonuçları bir sonraki adımdaki denemeleri geliştirmek için kullanır. BO'da hedef fonksiyon bilinmez olduğundan yüksek seviye bir karakutu optimizasyon metodudur. BO'nun iki temel içeriği vekil fonksiyon ve kazanç fonksiyonudur. Vekil fonksiyon genellikle bir Gauss fonksiyondur. BO bilinmeyen hedef fonksiyona bir ön tahmin (prior) ataması yapar ve denemeler sonrası elde edilen sonuçlara bakarak bu ön tahmini güncelleyerek hedef fonksiyonun son tahmin (posterior) dağılımlarını elde etmeye çalışır. Son tahmin dağılımları ile deneme sayısını azaltacak bir kazanç fonksiyonu oluşturulmaya çalışılır. Gauss vekil fonksiyonlarının bazı avantajları olduğu gibi yüksek boyutlu verilerle kullanıldığında ölçeklenebilirlik sorunları bulunmaktadır. Rassal ormanlar ve son yıllarda derin yapay sinir ağları bu soruna bir çözüm olarak önerilmektedir. BO genel olarak basit, gürültüsüz veri, hiper dörtgen gibi bazı varsayımlara dayandığı için zayıf yönleri bulunmaktadır ve daha gelişmiş türleri önerilmektedir. Bunların yanısıra BO paralelleştirmeye uygun değildir. BO kimya, robotik, derin öğrenme, bilgisayarla görme, parçacık fiziği, takviyeli öğrenme ve ilaç geliştirme gibi konularda oldukça yaygın uygulama alanları bulabilmiştir. Şekil 3'te bir fonksiyonun BO ile elde edilme aşamaları gösterilmektedir. BOHB ise Hiperkomite ve BO metodlarının iyi taraflarını birleştiren bir algoritmadır. BO algoritması sürekli değerli fonksiyonlar için kullanılabilir ancak kategorik değerli hiper parametre optimizasyon problemleri için uygun değildir [Garrido-Merchán ve Hernández-Lobato, 2020]. Ağaç yapılı Parzen tahmincileri kategorik değerli hiper parametre problemleri için bir çözüm olarak düşünülebilir. Örnek olarak bir yapay sinir ağındaki gizli katman sayısı ve her katmandaki nöron sayısı bir ağaç yapısı oluşturmaktadır. Çünkü herhangi bir katmanın nöron sayısından bahsedebilmek için öncelikle o katmanın var olması

gereklidir. Parzen tahmincileri deęişkenler arasındaki ilişkilerden yeterince yararlanamazlar [Watanabe ve Hutter, 2022].

PBT (Population-Based Training) [Jaderberg ve ark., 2017] optimizasyon işlemini verisetindeki belli örnekler için belli bir zaman aralıklarında çalıştırır ve her zaman aralığından sonra performansları karşılaştırır. İyi performanslı metotların ağırlıklarını ve ayarlarını düşük performanslı çalışmalara aktarır. Böylece onların da çalışma performanslarının iyi olmalarını sağlamaya çalışır ve en kötü performansların da sonlandırılmasını emreder. PBT algoritmasının arkasındaki mantık basit gözükmesine rağmen sıfırdan bunu inşa edebilmek oldukça güçtür. Zira algoritmayı ilerletebilmek ve geliştirmek için oldukça komplike matematiksel optimizasyon işlemlerine ihtiyaç bulunmaktadır. Kötü parametrelerle başlayan işler PBT tarafından parametre mutasyonu denen bir yöntemle optimize edilir. Hiper parametre mutasyonu PBT zamanlayıcısı tarafından yönetilir ve her t zaman aralığı sonrası hiper parametre değerleri mutasyona uğratarak istenen başarı metriğinin daha da geliştirilmesi sağlanır. PBT bir keşfet (explore) ve kullan (Exploit) metodolojisini temel alır ve önce bir arama uzayı keşfeder, bu uzaydaki iyi parametreleri bulur, onları başka yerlerde kullanarak değerlendirir ve kötü çalışmaları günceller. Hiper parametre optimizasyon algoritmalarının arama algoritması ve planlama algoritması olmak üzere 2 temel içeriği bulunmaktadır. Arama algoritması örnek olarak Rassal arama hiper parametre kümesinin nasıl oluşturulacağı ve optimize edileceği ile ilgilenirken, planlama algoritması ise parametre uzayındaki çalıştırmaların nasıl optimize edilebileceği ile ilgilenir. Örnek olarak kötü performanslı çalışmaların erken durdurulması verilebilir.

Hiperkomite, senkron SHA'nın başlangıç performansı zayıflığını azaltmaya çalışır. Hiperkomite algoritması, SHA üzerinde deęişik yarılama oranlarıyla dolaşır ve erken durdurmayı daha fazla kaynak ayırarak başlangıç performansına bakmadan dengelemeye çalışır. Asenkron Hiperkomite, Hiperkomiteye benzerdir ve o da ASHA üzerinde dolaşarak erken durdurmayı dengelemeye çalışır. PBT arama ve planlayıcının işlerini birleştirdiği için hibrit bir metot olarak kabul edilebilir. PBT, Genetik algoritmayla benzerlik gösteren bir algoritmadır. Her bir çalışana bir iş ayrılmıştır ve düşük performanslı konfigürasyonlar yüksek performanslı konfigürasyonlarla deęiştirilir (exploitation) ve rastgele olarak seçilir (exploration). HPO'da objektif fonksiyon belirsiz olduğu için bir karakutu optimize işlemi gereklidir. BO'da vekil hedef fonksiyonu modellerken yakalama fonksiyonu yeni örnekler ve yeni hiper parametreleri tespit ve bulma için kullanılır. GPBT (Generalized Population Based Training) [Bai ve Cheng, 2024], PBT'nin yüksek performanslı çalışanlara olan tarafsızlığından dolayı oluşan eksikliğini gidermek amacıyla önerilmiş bir metottur. Çiftli öğrenme (Pairwise Learning) performans farklılıklarını tespit için çiftli bir strateji kullanan bir öğrenme metodudur. PL, zayıf performanslı çalışanlara rehberlik sağlar. GPBT, PL ile birlikte kullanılınca PBT'ye oranla hesaplama, adaptasyon ve maliyet açısından özellikle takviyeli öğrenmede daha olumlu sonuçlar üretebilmektedir.



Şekil 3. Bilinmeyen bir fonksiyonun Bayes optimizasyon yaklaşımıyla adım adım tahmin edilmesi

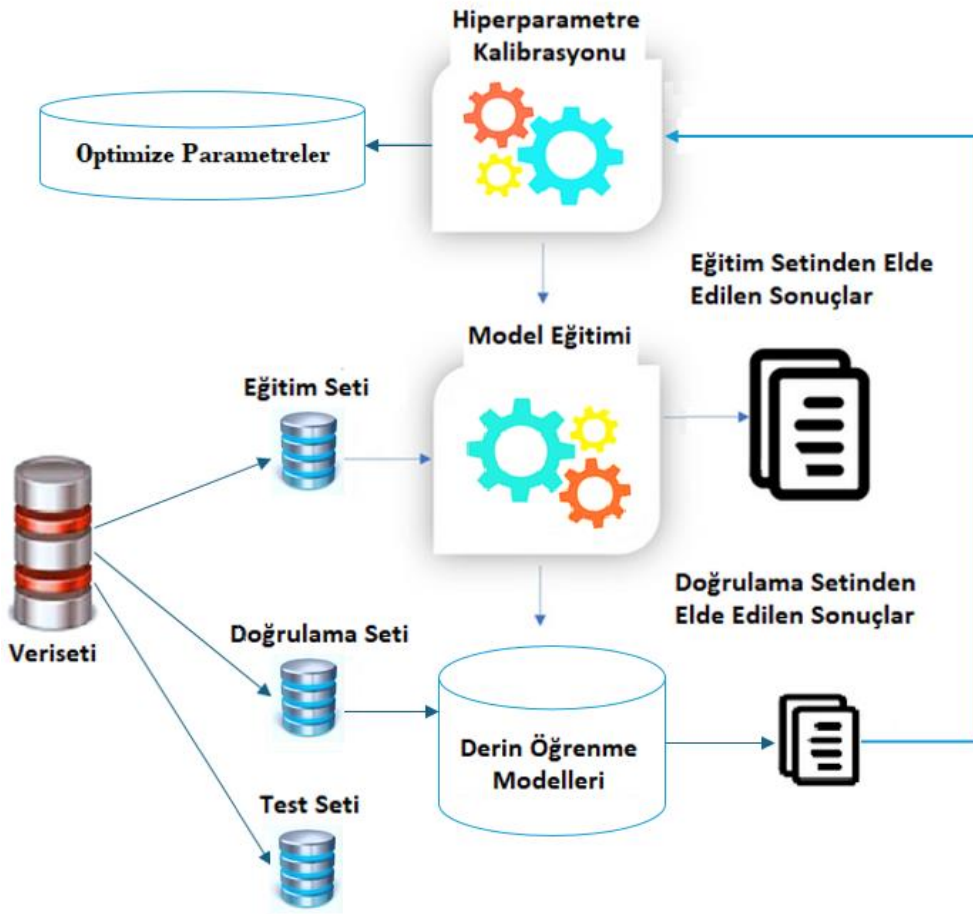
3. Konvolusyonel Sinir Ağları, Çevrimli Sinir Ağları ve Derin Sinir Ağları

Bu bölümde uygulanacak olan derin öğrenme metotlarından bahsedilecektir. Hiper parametre optimizasyonu herhangi bir makine öğrenmesine uygulanabilir. Bunlar arasında karar ağaçları, rassal ormanlar, SVM ve bütün yapay sinir ağları sayılabilir. Konvolusyonel Yapay Sinir Ağları [LeCun ve ark., 1995] günümüzde en yaygın kullanılan yapay zekâ metotlarından biridir. KYSA'lar resim görüntüleri için geliştirilmiş olmalarına rağmen konvolüsyon işleminin 1-Boyutlu, 2-Boyutlu veya 3-Boyutlu uygulanabilir olması sayesinde günümüzde zamansal veriler hatta video verileri üzerinde de kullanılmaktadır. Derin öğrenme çalışmaları her ne kadar RBM'lerle [Hinton ve Salakhutdinov, 2006] başlamış ise de günümüzde KYSA'lar RBM'leri tahtından indirmiş ve alanı domine eder hale gelmiştir. Standart yapay sinir ağlarının aksine KYSA'larda giriş verisi konvolusyonel katmanların tamamına bağlantı vermez ve belli noktalara odaklanmaya çalışırlar. KYSA'ların ölçekleme ve ötelemeye karşı dirençli oldukları söylenmesine karşın döndürmeye karşı dirençli değildir ve bunun için veri çoğaltma ile döndürülmüş verilerle de eğitilerek direnç kazanmaları sağlanabilir. KYSA'larda konvolusyonel katmanlar esasen bir özellik çıkarma işlemi olarak düşünülebilir ve bu işlem bir veya birden fazla tam bağlantılı yapay sinir ağı ile devam eder ve bir softmax sınıflandırıcı ile sonlandırılır. Aydın [Aydın ve ark., 2022] duygu tanımada konvolusyonel sinir ağlarını uygulamışlardır.

LSTM [Hochreiter ve Schmidhuber, 1996] zaman serisi veriler için geliştirilmiş bir yapay sinir ağı modelidir ve hafızalı bir çevrimsel model olarak ele alınabilir. LSTM'ler patlayan ve yok olan gradyanlara iyi bir çözüm olarak önerilmişlerdir. GRU [Cho ve ark., 2014], LSTM'nin biraz daha basitleştirilmiş halidir ve genel olarak performans açısından yakın hatta bazen daha iyi sonuçlar üretebilmektedir. Genel olarak bir LSTM modelinin yapısı hafıza, giriş kapısı, çıkış kapısı ve unut kapısından oluşur. Bu kapıların her biri ayrı birer yapay sinir ağından oluşmaktadır. LSTM modeller çift yönlü olarak da tasarlanabilmektedirler. LSTM modeller dikkat mekanizması (attention), dönüştürücü (transformer) ve son olarak da xLSTM [Beck ve ark., 2024] ile daha da geliştirilmişlerdir. Donuk [Donuk ve Hanbay, 2022] RAVDESS ve TESS verisetleri üzerinde LSTM tabanlı modelleri deneyerek %88,92 başarı oranlarına ulaşmışlardır.

Derin Yapay Sinir ağları standart tek katmanlı yapay sinir ağlarında birden fazla katman kullanılmasıyla ortaya çıkmıştır. Birden fazla katmanlarda patlayan ve yok olan gradyanlar sorunu oldukça uzun bir süre araştırmacıları meşgul eden bir problem olarak varlığı sürdürmüş ve sonraları RELU aktivasyon fonksiyonları ve LSTM gibi modellerle üstesinden gelinmiştir. Çok katmanlı sistemler her ne kadar genel olarak performans ve başarı oranlarına olumlu etkide bulunuyor olsa bile bunu her zaman garanti etmediklerini akıldta tutmak gereklidir. Daha dar ve daha yüzeysel ağlar bazen geniş ve derin ağlardan daha yüksek performans sergileyebilmektedirler.

Bu çalışmada el ile kendi modelimizi oluşturmak yerine KerasTuner hiper parametre optimizasyon kütüphanesi ile kalibre edilmiş bir model oluşturacağız. KerasTuner genel amaçlı bir model oluşturma kütüphanesidir. KerasTuner ile model geliştirmeden önce basit bir başlangıç modelini taban model olarak seçerek daha sonra geliştireceğimiz modellerle kıyaslayacağız. KerasTuner ayrıca filtre sayısı, adım sayısı, havuzlama sayısı gibi parametrelerin de optimize edilmesine imkân tanımaktadır. Hiper kalibrasyon kütüphaneleriyle model arama ve optimize etme Şekil 4’te tasvir edilmektedir.



Şekil 4. Hiper optimizasyon işleminin genel görünüşü

4. Veriseti ve Özellik Çıkarma

Çalışmamızda konuşma ses duygu tanıma örneklerinden oluşan 2020 yılında hazırlanan çok yeni bir veriseti olan NEMO veriseti [Christop, 2024] kullanılmıştır. NEMO veriseti yapay bir veri setidir ve 5 erkek 4 kadın olmak üzere 9 farklı konuşmacıdan elde edilen 6 kategorik duygu sınıfından (şaşkınlık, öfke, üzüntü, korku, mutluluk, nötr) oluşmaktadır. Verisetinde günlük konuşmalarda sık kullanılan 90 cümle bulunmaktadır ve her cümle Lehçe dilinde az rastlanan bir heceyi içermektedir. Bu şekilde verisetinin mümkün olduğunca dilbilimsel olarak yeterli zenginlik seviyesinde olması

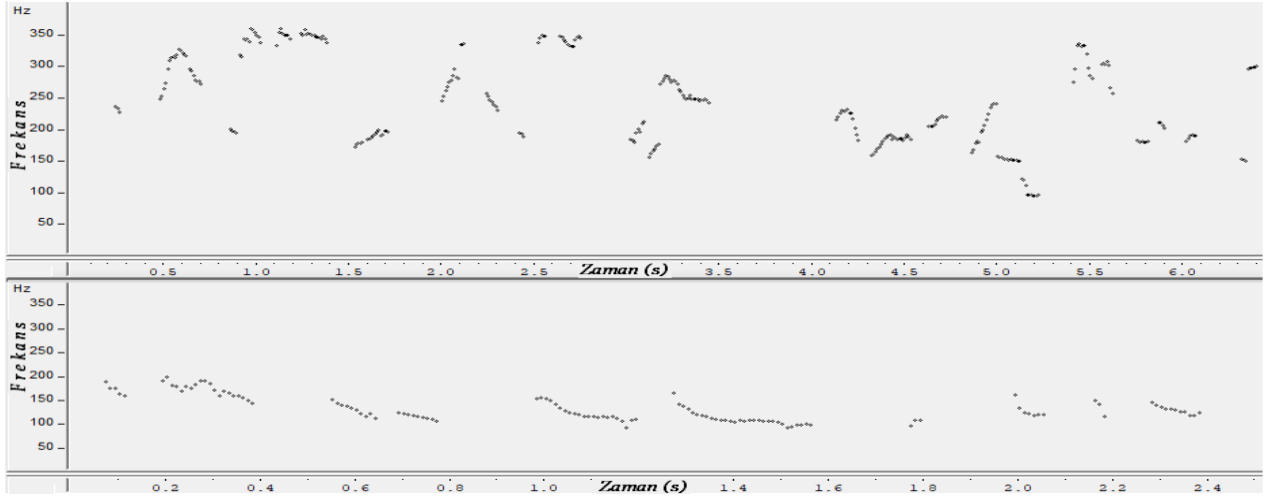
amaçlanmıştır. Veriseti konuşmacı kimliği, yaşı ve cinsiyeti bilgilerini de sağlamaktadır. Duygu sınıflarında 749 mutlu, 749 kızgın, 769 üzgün, 736 korku, 669 sürpriz ve 809 nötr olmak üzere toplam 4481 konuşma ses örneği bulunmaktadır. NEMO veriseti, maksimum uzunluk 5,66 saniye, minimum uzunluk 0,87 saniye, ortalama uzunluk 2,47 saniye, standart sapma 0,65 saniye olan toplam 3,07 saatlik, Lehçe dilinde, 24 kHz örnekleme frekanslı, tek kanallı duygusal konuşma ses örnekleri içerir. Verisetinden özellik çıkarımında ses örnekleri 31,25 milisaniyelik parçalara ayrılarak ve %75 örtüşen adımlarla ilerletilerek işlenmektedir. Her parça için temel frekans ile beraber 13 MFCC katsayısı hesaplanmakta ve bütün parçalara ait maksimum, minimum, standart sapma, aralık, ortalama, maksimum noktası, minimum noktası, 1. ve 2. Doğrusal regresyon katsayıları, doğrusal regresyon aritmetik ve kuadratik hatalar, diklik, yatıklık, 1., 2., ve 3. Çeyrekler arası aralıklar, 90 ve 98'lik yüzdelikler, varyans olmak üzere toplam olarak 22 istatistiksel fonksiyon hesaplanmaktadır. Birinci derece delta parametreleriyle beraber toplam öznelik sayısı 704'e ulaşmaktadır. NEMO veriseti çok yeni bir veriseti olduğu için üzerinde henüz çalışma yapılmamış farklı dilde bir verisetidir. Bunun yanısıra diğer verisetleriyle kıyaslandığında sınıflar arasında dengeli bir dağılıma sahip olması, örnek sayısı açısından orta ölçekte bir veriseti olması diğer avantajlarıdır. Tablo 1'de NEMO veriseti diğer yaygın kullanılan verisetleriyle değişik yönlerden kızgın, mutlu, üzgün, nötr örnek sayısına göre kıyaslanmaktadır. Tablo 1'den görüleceği gibi IEMOCAP ve MELD [Poria ve ark., 2018] verisetlerinde duygu sınıflarının dağılımlarındaki eşitsizlik oldukça belirgindir. EmoDB [Burkhardt ve ark., 2005] ve EmoSTAR [Parlak ve ark., 2014] verisetlerinde de duygu sınıfları arasında dengesiz bir dağılım mevcuttur ve aynı zamanda bu verisetleri küçük hacimlidirler. TESS veriseti dengeli bir dağılıma sahip olmasına rağmen küçük ölçeklidir ve sadece 2 kadın konuşmacıdan oluşmaktadır. SAVEE [Haq ve Jackson, 2010] veriseti duygu sınıfları arasında nötr hariç eşit bir dağılıma sahip olmasına rağmen hem küçük ölçekli hem de konuşmacı sayısı açısından yeterli değildir ve sadece erkek konuşmacılardan oluşmaktadır. Bu verisetleri arasında NEMO dengeli sınıf dağılımı, orta ölçekli hacmi, nispeten yeterli konuşmacı sayısı ve kadın-erkek konuşmacı dengesi bakımından diğerlerinden üstün yönere sahiptir.

Tablo 1. NEMO verisetinin sık kullanılan diğer verisetleriyle kızgın, mutlu, nötr ve üzgün örnek sayıları için karşılaştırılması

	EmoDB	EmoSTAR	TESS	SAVEE	MELD	IEMOCAP	NEMO
Toplam Örnek Sayısı	339	393	1600	300	11353	4490	3076
Uzunluk (saat)	0,26	0,70	0,92	0,32	10,20	5,60	2,10
Konuşmacı Sayısı	10 5 Erkek 5 Kadın	106 61 Erkek 45 Kadın	2 0 Erkek 2 Kadın	4 4 Erkek 0 Kadın	407 - -	10 5 Erkek 5 Kadın	9 5 Erkek 4 Kadın
Kızgın Örnek Sayısı	127	103	400	60	1607	1103	749
Üzgün Örnek Sayısı	62	82	400	60	1002	1084	749
Mutlu Örnek Sayısı	71	82	400	60	2308	595	769
Nötr Örnek Sayısı	79	126	400	120	6436	1708	809

Konuşma ses verilerinde yaygın olarak kullanılan öznelikler Mel filtreleri, MFCC ve Gammatone filtreleri sayılabilir. Mel filtreleri konuşma sinyalinin frekans spektrumunu belli adet (genelde 40) aralıklara bölerek insan kulağındaki modellemeyi taklit etmeye çalışır. Bu aralıklar insan işitme sisteminin yapısı ile uyumlu olarak belli bir frekansa kadar doğrusal aralıklı (1000 Hz veya 1200 Hz), daha ötesinde ise logaritmik olarak artan aralıklıdır. Konuşma ses tanıma uygulamalarında genellikle 16000 Hz örnekleme frekans kullanılır. Daha yüksek örnekleme frekansları ses kalitesini artırsa da konuşma tanıma performansı üstünde etkili değildir ve hesaplama ve hafıza maliyetini büyük oranda arttırdığı için tercih edilmez. Kaynak darlığı olan alanlarda ise bu aralık daraltılarak 4000 Hertz'e kadar düşürülebilir. Televizyon veya bilhassa telefonlarda ise spektrumun başlangıç bölgesi de 300 Hz veya 400 Hz'e kadar bir bant silen (bandstop) filtre ile silinebilir. Zira bu frekans bölgelerinin silinmesi konuşmanın anlaşılabilirliğini pek etkilemez ve bant genişliğinin daha verimli olarak kullanılabilmesine olanak sağlar. MFCC, Mel filtrelerinden Doğrusal Kosinüs Dönüşümü (Discrete Cosine Transform) [Ahmed ve ark., 1974] ile elde edilir ve elde edilen katsayılardan 13 adedini kullanır. MFCC, Mel filtrelerine PCA uygulama gibi de düşünülebilir ve esasen bir özellik dönüştürme ve indirgeme metodudur ve jpg (Joint Photographic Experts Group) resim formatında çok başarılı bir şekilde uygulama alanı bulmuştur. Konuşma duygu tanımda kullanılan öznelikler, önişleme ve sınıflandırıcılar Özseven [Özseven, 2019] ve Çolakoğlu [Çolakoğlu ve ark., 2021] tarafından ayrıntılı incelenmiştir.

Konuşma sesinden duygu tanıma uygulamalarında özellik çıkartma işlemi genellikle konuşma tanıma uygulamalarındaki özellik çıkarma işleminden biraz daha farklı uygulanır. Zira duygu tespitinde özelliklerin zaman içinde değişimleri de büyük önem taşımaktadır. Kızgın tonda söylenen bir cümleyle nötr tonda söylenen bir cümle arasında özelliklerin zamansal değişimi oldukça büyük farklılıklar göstermektedir. Konuşma tanıma uygulamalarında temel frekans olarak da adlandırılan f_0 , konuşma tanıma açısından bir önemi olmadığından kullanılmazken, konuşma duygu tanımda oldukça önemli bir özellik olarak karşımıza çıkmaktadır. Temel frekansın zaman içindeki değişimi bazı duygu sınıflarında oldukça belirgin ayırt edici bir niteliktir. Şekil 4'te kızgın ve nötr duygulu bir cümlenin temel frekans grafikleri ile buna bir örnek verilmektedir. Şekil 4'ten de görüldüğü gibi kızgın cümledeki temel frekans çizgileri ile nötr cümledeki temel frekans çizgileri arasında büyük fark bulunmaktadır. Kızgın cümlede çok büyük dalgalanmalar meydana gelmesine karşın nötr cümlede temel frekans sabit yatay bir çizgi izlemektedir. Bu nedenle konuşma ses verilerinden özellik çıkarma işlemlerinde MFCC ve Mel filtreleri ses özelliklerinin istatistiksel fonksiyonları (ortalama, maksimum, minimum, standart sapma, varyans, yüzdellikler, çeyreklikler vb.) tercih edilmektedir. Bunun yanında doğrudan MFCC ve Mel filtrelerini veya Mel filtrelerinin spektrogramlarını kullanan çalışmalarda vardır. Duygu tanımda da delta katsayıları 1. derece veya 2. derece olarak kullanılabilir [Dala, 2023; Dikbıyık ve ark., 2022].



Şekil 5. Kızgın bir konuşma ses sinyalinin (üstte) temel frekans (f_0) çizgisi ile nötr bir konuşma ses sinyalinin temel frekans çizgisinin (altta) karşılaştırılması.

5. Sınıflandırma

Bu bölüm mevcut modelleri doğrudan kullanmak yerine hiper parametre optimizasyon kütüphanelerinden KerasTuner ile model geliştirmeye ayrılmıştır. Öncelikle bir taban model olarak basit bir konvolusyonel model seçilmiş ve hiper parametre optimizasyonunun katkısını göstermek için üzerinde herhangi bir iyileştirme çalışması yapılmamıştır. Taban modelin sonuçları KerasTuner sonuçları ile karşılaştırılacaktır. KerasTuner Rassal, Bayeşçi ve Hiperkomite arama metotlarını kullanıma sunmuştur. Çalışmamızda bu optimizasyon metotları, KYSA, LSTM ve DNN yapay sinir ağları ile denenecek ve sonuçlar taban metotla karşılaştırmalı olarak verilecektir. Sınıflandırma çalışmalarında verisetleri genellikle %70 eğitim ve %30 test seti olarak ikiye ayrılmaktadır. Ancak yapılan çalışmalar bu tür ayırmanın konuşma duygu tanıma uygulamalarında konuşmacı bağımlı özelliklerden dolayı aşırı öğrenmeye neden olduğunu göstermektedir. Bu nedenle araştırmacılar verisetlerini konuşmacı bağımsız eğitim ve test yapılabilecek şekilde düzenlemeye başlamışlardır. Bir Konuşmacıyı Dışarıda Bırak (Leave One Speaker Out, LOSO) olarak adlandırılan bu metotta eğitim seti ile test setindeki konuşmacıların farklı olması amaçlanmaktadır. LOSO yönteminde, eğitim verisetindeki konuşmacıların hiçbiri test verisetinde bulunmamaktadır. Bu çalışmada da NEMO veriseti LOSO yöntemiyle eğitim ve test setlerine ayrılmıştır. Eğitim setinde 2357 örnek, test setinde 719 örnek bulunmaktadır. Ses örneklerinin dağılımı Tablo 2’de gösterildiği gibi oldukça dengelidir ve herhangi bir duygu sınıfına eğilim göstermemektedir. Ayrıca az sınıflı olması ve küçük bir veriseti olmaması çapraz-geçerleme (cross-validation) uygulama ihtiyacını bertaraf etmektedir. Deneyler KerasTuner 2.10.0 ile Colab V100 GPU’larından yararlanılarak Python 3.9.13 ile, özellik çıkarma işlemi ise MATLAB 2019a ve Auditory Toolbox [Slaney, 1998] kullanılarak gerçekleştirilmiştir.

Tablo 2. Deneylerde kullanılan NEMO verisetinde eğitim ve test setlerindeki duygu örneklerinin dağılımı

	Kızgın	Üzgün	Nötr	Mutlu	Toplam
Eğitim Seti	569	569	630	589	2357
Test Seti	180	180	179	180	719
Toplam	749	749	809	769	3076

Eğitim aşamasında Rassal, Bayeşçi ve Hiperkomite arama yöntemleri her denemede 50 çevrim olmak üzere 20 deneme yapılmıştır. Öğrenme oranı 0,001 ile başlayıp her 20 çevrimlik düzlükte %90 azaltılmıştır. Optimizasyon metodu olarak Adam [Kingma ve Ba, 2017] optimizasyon metodu kullanılmıştır. KYSA deneylerinde katman sayısı ve filtre sayısı, DNN ve LSTM modellerde katman sayısı ve nöron sayısı hiper parametre olarak kullanılmıştır. Katman sayısı 5'e kadar, DNN nöron sayısı 256-1024 aralığında 256 artışlı, LSTM nöron sayısı 32-128 aralığında 32 artışlı, KYSA filtre sayısı 32-128 aralığında 32 artışlı, öğrenme katsayısı 0.001 ve 0.0001 olarak aranmıştır. Genel olarak Hiperkomite aramasının diğerlerine göre daha fazla zaman gerektirdiği söylenebilir. Test aşamasında en iyi 5 hiper parametrelili model 100 çevrim çalıştırılmış ve en iyi modeller seçilmiş ve başarı oranları Tablo 3'te sunulmuştur. Sonuçlardan görülebileceği gibi hiper parametre arama metotlarının hepsi taban modele göre çok büyük ilerlemeler kaydetmektedirler. Bu çalışmada Bayeşçi arama LSTM modellerle en yüksek başarı oranına ulaşmıştır. Metotlar arasında KYSA modeli için Colab ortamındaki toplam tamamlanma süresi ve bellek tüketimi açısından yapılan bir karşılaştırma Tablo 4'te sunulmaktadır. Tablo 4'ten görüleceği gibi Hiperkomite metodu her yönden en pahalı metottur ve özellikle bellek tüketimi açısından çok büyük dezavantaja sahiptir. Bayeşçi arama ve Rassal arama neredeyse başa baş bir performans sergilemektedirler. Rassal arama oldukça basit ve kolay uygulanabilen bir metot olmasına rağmen hem hız ve bellek tüketimi hem de başarı oranları açısından Bayeşçi metotla çok sıkı bir şekilde rekabet edebilmektedir.

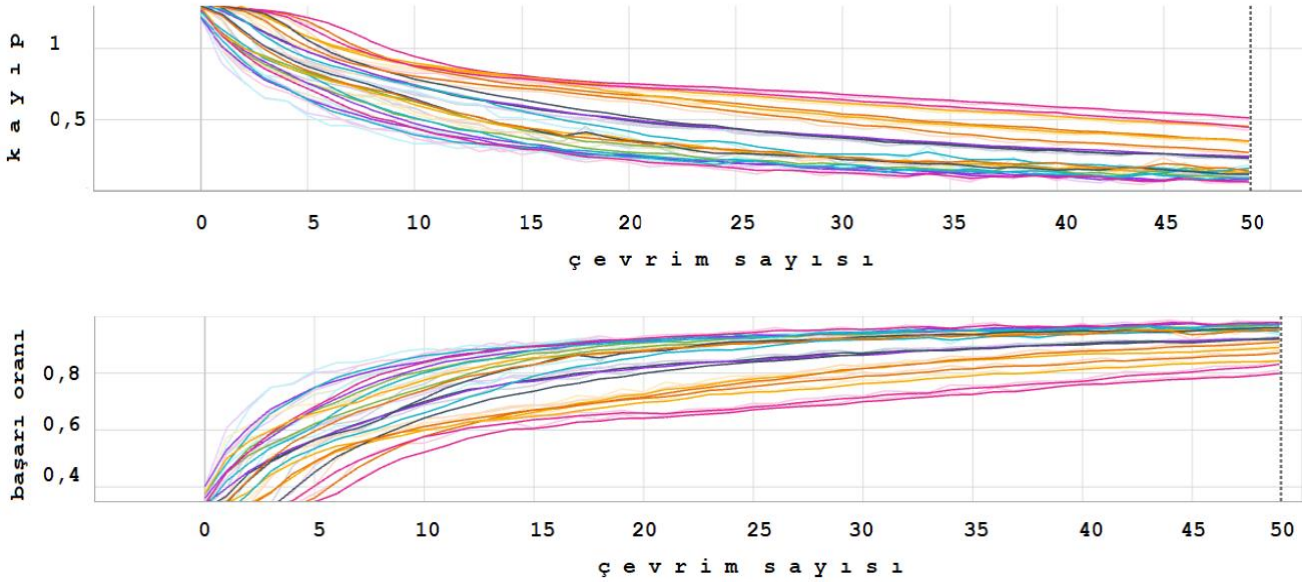
Tablo 3. KYSA, LSTM ve DNN modellerin farklı hiper parametre arama metotlarına göre başarı oranları

	Taban Model	Rassal Arama	Bayeşçi Arama	Hiperkomite Arama
KYSA	38,80	56,05	57,30	53,82
LSTM	42,83	63,28	64,53	63,56
DNN	37,55	61,89	61,47	61,89

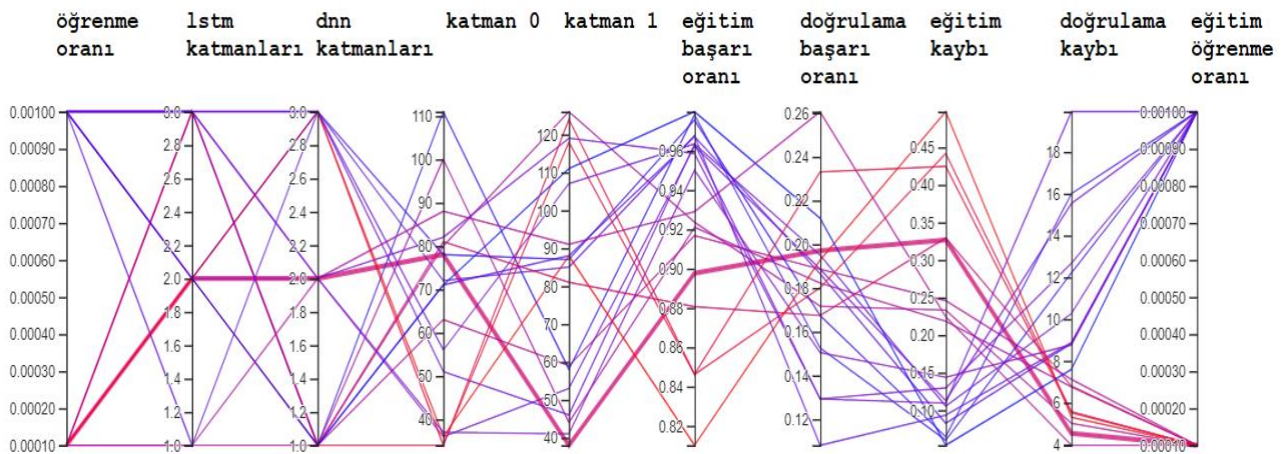
Tablo 4. KYSA modeli için farklı hiper parametre arama metotlarının Colab ortamındaki toplam tamamlanma süresi, bellek ve disk kullanımları

	Rassal Arama	Bayeşçi Arama	Hiperkomite Arama
Süre (Dakika)	32	32	54
Sistem RAM (GB)	6,5	6,3	20,8
GPU RAM (GB)	4,3	4,3	16,8
Disk (GB)	2,0	2,3	9,8

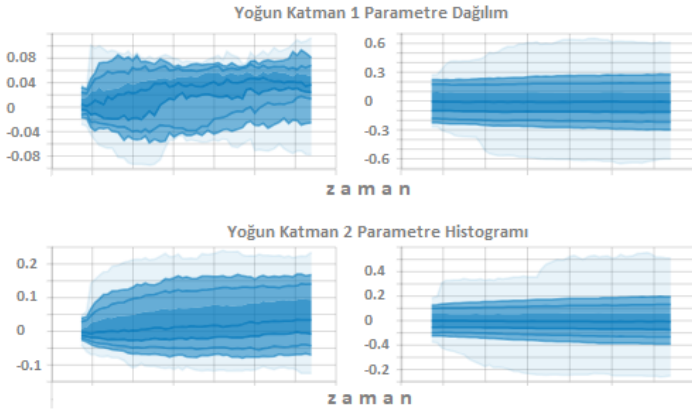
Bu çalışmadaki deneylerde Colab ve KerasTuner'den yararlanılmıştır. KerasTuner, Tensorboard ile hiper parametre aramalarının grafiksel olarak gösterimini de sağlamakta ve model geliştirme aşamalarında parametrelerdeki değişimleri daha yakından takip edebilmenizi sağlamaktadır. Şekil 6'da Bayeşçi aramada Tensorboard ile her denemede elde edilen kayıp ve başarı oranları resmedilmektedir. Şekil 7 Bayeşçi aramada hiper parametre optimizasyonu aşamalarında katmanların ve performans metriklerinin aldıkları değerleri göstermektedir. Şekil 8 yoğun katman parametrelerinin zamana bağlı gelişmelerini, Şekil 9 ise Bayeşçi aramada yoğun katman parametrelerinin belli bir denemede çevrimlere göre aldıkları değerlerin değişimlerini ve histogramlarını gözlemlemektedir. Arama sırasında toplam deneme sayısı 20, her denemede 50 çevrim olacak şekilde başlangıç parametreleri seçilmiştir.



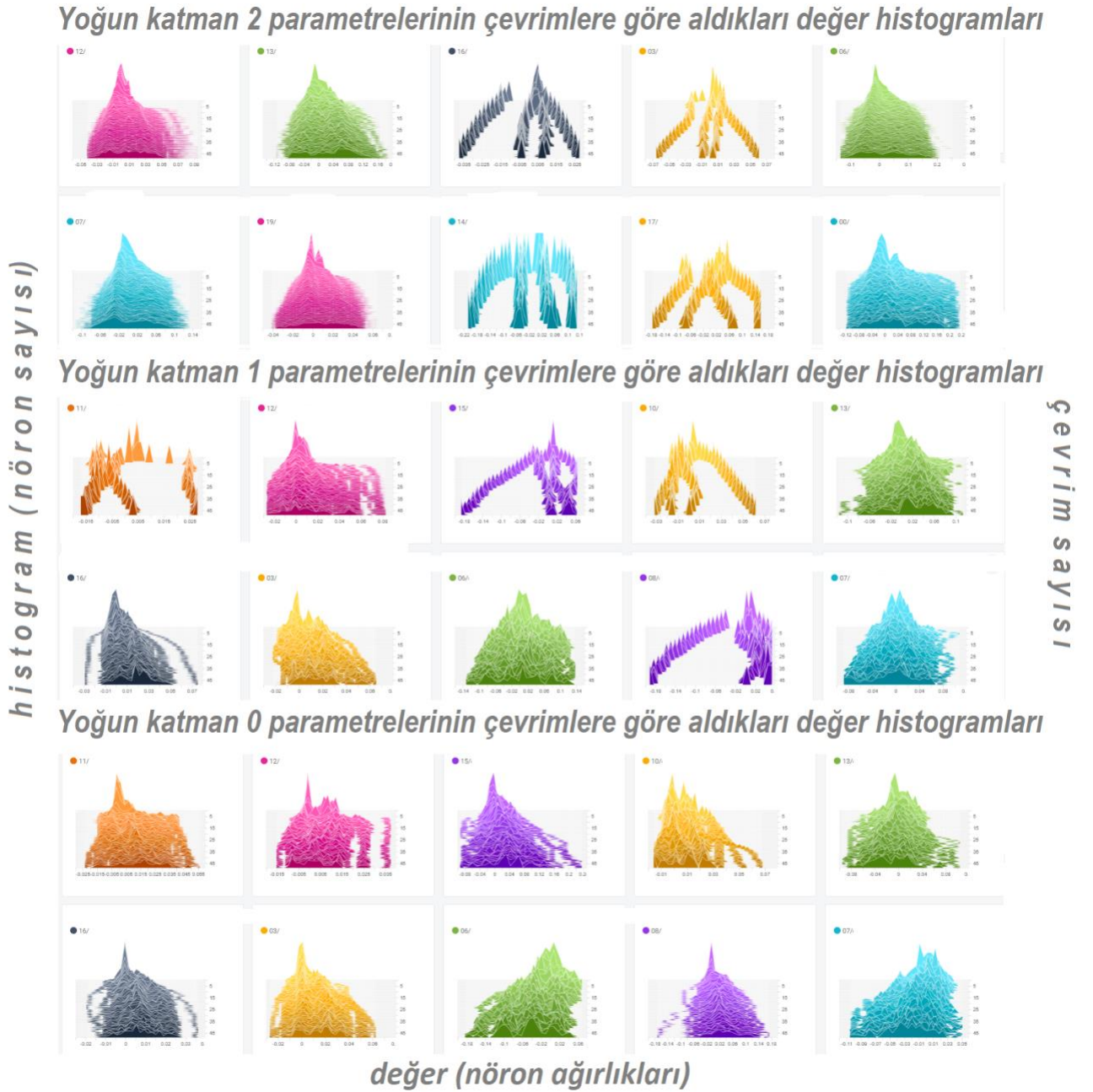
Şekil 6. Bayeşçi aramada çevrim sayısına göre denemelerdeki kayıp (üstte) ve başarı oranları (altta). Her deneme 50 çevrim olmak üzere toplam 20 deneme gerçekleştirilmiştir.



Şekil 7. Bayeşçi aramada hiper parametre optimizasyonu aşamalarında katmanların ve başarı metriklerinin aldıkları değerler



Şekil 8. Bayeşçi aramada yoğun katman parametrelerinin aldıkları değerlerin zamana göre değişimleri



Şekil 9. Bayeşçi aramada yoğun katmanların parametrelerinin belli denemelerde aldıkları değerlerin çevrim sayısına göre değişimleri

6. Sonular ve neriler

Bu alıřmada KerasTuner ile hiper parametre optimizasyon metodlarından Rassal Arama, Bayeřçi Arama ve Hiperkomite Arama incelenmiř olup yeni bir veriseti olan NEMO konuřma duygu veriseti üzerinde KYSA, LSTM ve DNN modelleri ile deneyler yapılmıřtır. Deneyler literatürde sık kullanılan kızgın, üzgün, nötr ve mutlu sınıfları üzerinde gerekleřtirilmiřtir. NEMO veriseti ok yeni bir veriseti olması, orta büyüklükte örnek sayısına sahip olması, üzerinde henüz derin ğrenme ile alıřılmamıř olması, duygu sınıflarının örnek sayıları ve konuřmacı cinsiyet sayısı aısından dengeli bir yapıda olması, nispeten yeterli konuřmacı sayısı aısından tercih edilmiřtir. apraz geerleme yerine, eđitim ve test setlerinde tamamen farklı konuřmacıların bulunmasını gerektiren LOSO metodu kullanılmıř ve konuřmacı bađımlı özelliklerden kaynaklanabilecek ařırı ğrenme probleminin önüne geilmeye alıřılmıřtır. alıřmamızın sonunda KerasTuner ile olduđca bařarılı modeller geliřtirilebileceđi görölmüř ve taban modele göre bařarı oranlarında 25 puana varan ilerlemeler görölmüřtür. Taban model bařarı oranları model üstünde herhangi bir iyileřtirme denemesi yapılmadıđı için olduđca düřüktür. Esasen hiper parametre optimizasyonu iřleminden beklenen de herhangi bir model denemeden en iyi modelin otomatik olarak bulunabilmesidir. alıřmamızda konvolusyonel, LSTM ve derin yapay sinir ađı modelleri denenmiř ve LSTM modeller diđer modellere göre daha bařarılı olmuřtur. Genel olarak Bayeřçi hiper parametre optimizasyon metodunun daha bařarılı olduđu söylenebilir. Diđer bir ilgin sonuç Rassal aramanın olduđca iyi sonular üretebilmiř olmasıdır. Rassal arama süre ve bellek tüketimi aısından Bayeřçi aramayla rekabet edebilmiř, bařarı oranı aısından Bayeřçi aramaya olduđca yakın bařarı oranlarına ulařabilmiř ve hatta DNN sınıflandırıcıda Bayeřçi metottan bile daha iyi bir performans gösterebilmiřtir. Yukarıda da deđindiđimiz gibi bunun nedeni herhangi bir optimizasyon problemi için kesin belirli bir Altın Parametre Seti olmamasıdır. Aksine en iyi veya ona ok yakın bařarı oranını sađlayabilen birden fazla parametre kümesi ve sınıflandırıcı modelleri bulunabilir ve bu parametreler bir dađılım içinde bulunabilirler. Diđer bir dikkat eken sonuç ise Hiperkomite metodunun ařırı bellek ve zaman harcamasına rađmen beklenen bařarıyı gösterememesidir. Hiper parametre optimizasyonu olduđca fazla hesaplama gücü gerektiren bir alıřma alanıdır. Bu alıřmada parametre sayısı ve model türü nispeten sınırlı tutulmuřtur. Parametre sayısı ve aralıkları arttırılarak daha iyi sonulara ulařmak da mümkündür. İleriki alıřmalarda Bayeřçi Hiperkomite (BOHB), SHA, ASHA, GPBT, meta-sezgisel ve biyo-kökenli optimizasyon yöntemleri hiper parametre aramalarında kullanılabilir. Ayrıca hiper parametre sayısı arttırılabileceđi gibi aktivasyon fonksiyonları ve daha yeni modellerden dikkat (attention) ve dönüřtürücü (transformer) modellerde aramalara eklenebilir.

Araştırma ve Yayın Etiği Beyanı

Yazar, makalenin tüm süreçlerinde “Yükseköğretim Kurumları Bilimsel Araştırma ve Yayın Etiği Yönergesi” kapsamında uyulması gerekli tüm kurallara uyulduğunu, karşılaşılabilecek etik ihlallerden Karadeniz Fen Bilimleri Dergisi ve yayın kurulunun herhangi bir sorumluluğunun bulunmadığını, bu çalışmanın Karadeniz Fen Bilimleri Dergisi dışında herhangi bir akademik yayın ortamında değerlendirilmediğini beyan eder.

Kaynaklar

- He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning For Image Recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 770-778.
- Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., ... & McGrew, B. (2023). GPT-4 Technical Report. *arXiv preprint arXiv:2303.08774*.
- Hutter, F., Kotthoff, L., & Vanschoren, J. (2019). *Automated Machine Learning: Methods, Systems, Challenges* (219). Springer Nature.
- Bischl, B., Binder, M., Lang, M., Pielok, T., Richter, J., Coors, S., ... & Lindauer, M. (2023). Hyperparameter Optimization: Foundations, Algorithms, Best Practices, And Open Challenges. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 13(2), e1484.
- Vincent, A. M., & Jidesh, P. (2023). An Improved Hyperparameter Optimization Framework For AutoML Systems Using Evolutionary Algorithms. *Scientific Reports*, 13(1), 4737.
- Wu, X., Hu, S., Wu, Z., Liu, X., & Meng, H. (2022, May). Neural Architecture Search For Speech Emotion Recognition. *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (6902-6906). IEEE.
- Gökalp, S., & Aydın, İ. (2021). Farklı Derin Sinir Ağı Modellerinin Duygu Tanımadaki Performanslarının Karşılaştırılması. *Muş Alparslan Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi*, 2(1), 35-43.
- Livingstone, S. R., & Russo, F. A. (2018). The Ryerson Audio-Visual Database Of Emotional Speech And Song (RAVDESS): A Dynamic, Multimodal Set Of Facial And Vocal Expressions In North American English. *PLoS One*, 13(5), e0196391.
- Pichora-Fuller, M. Kathleen; Dupuis, Kate, 2020, Toronto Emotional Speech Set (TESS), <https://doi.org/10.5683/SP2/E8H2MF>.
- Sun, H., Lian, Z., Liu, B., Li, Y., Sun, L., Cai, C., ... & Cheng, Y. (2022). EmotionNAS: Two-stream Architecture Search For Speech Emotion Recognition. *arXiv preprint arXiv:2203.13617*.
- Rajapakshe, T., Rana, R., Khalifa, S., Sisman, B., Schuller, B. W., & Busso, C. (2024). emoDARTS: Joint Optimization Of CNN & Sequential Neural Network Architectures For Superior Speech Emotion Recognition. *arXiv preprint arXiv:2403.14083*.
- Brochu, E., Cora, V. M., & De Freitas, N. (2010). A Tutorial On Bayesian Optimization Of Expensive Cost Functions, With Application To Active User Modeling And Hierarchical Reinforcement Learning. *arXiv preprint arXiv:1012.2599*.
- Li, L., Jamieson, K., DeSalvo, G., Rostamizadeh, A., & Talwalkar, A. (2018). Hyperband: A Novel Bandit-Based Approach To Hyperparameter Optimization. *Journal of Machine Learning Research*, 18(185), 1-52.
- Li, K., & Li, F. (2024). Multi-Fidelity Methods for Optimization: A Survey. *arXiv preprint arXiv:2402.09638*.
- Jamieson, K., & Talwalkar, A. (2016, May). Non-stochastic Best Arm Identification And Hyperparameter Optimization. *Artificial Intelligence and Statistics* (240-248). PMLR.
- Falkner, S., Klein, A., & Hutter, F. (2018, July). BOHB: Robust And Efficient Hyperparameter Optimization At Scale. *International Conference on Machine Learning* (1437-1446). PMLR.
- Garrido-Merchán, E. C., & Hernández-Lobato, D. (2020). Dealing With Categorical And Integer-Valued Variables In Bayesian Optimization With Gaussian Processes. *Neurocomputing*, 380, 20-35.

- Watanabe, S., & Hutter, F. (2022). c-TPE: Generalizing Tree-Structured Parzen Estimator With Inequality Constraints For Continuous And Categorical Hyperparameter Optimization. *arXiv preprint arXiv:2211.14411*, 240.
- Jaderberg, M., Dalibard, V., Osindero, S., Czarnecki, W. M., Donahue, J., Razavi, A., ... & Kavukcuoglu, K. (2017). Population Based Training Of Neural Networks. *arXiv preprint arXiv:1711.09846*.
- Bai, H., & Cheng, R. (2024). Generalized Population-Based Training For Hyperparameter Optimization In Reinforcement Learning. *IEEE Transactions on Emerging Topics in Computational Intelligence*.
- LeCun, Y., Jackel, L. D., Bottou, L., Cortes, C., Denker, J. S., Drucker, H., ... & Vapnik, V. (1995). Learning Algorithms For Classification: A Comparison On Handwritten Digit Recognition. *Neural Networks: the Statistical Mechanics Perspective*, 261(276), 2.
- Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing The Dimensionality Of Data With Neural Networks. *Science*, 313(5786), 504-507.
- Aydin, M., Tuğrul, B., & Ar, Y. (2022). Emotion Recognition System From Speech Using Convolutional Neural Networks. *Computer Science, IDAP-2022: International Artificial Intelligence and Data Processing Symposium*, 137-143. <https://doi.org/10.53070/bbd.1174033>
- Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, 9(8), 1735-1780.
- Cho, K., Van Merriënboer, B., Bahdanau, D., & Bengio, Y. (2014). On The Properties Of Neural Machine Translation: Encoder-Decoder Approaches. *arXiv preprint arXiv:1409.1259*.
- Beck, M., Pöppel, K., Spanring, M., Auer, A., Prudnikova, O., Kopp, M., ... & Hochreiter, S. (2024). xLSTM: Extended Long Short-Term Memory. *arXiv preprint arXiv:2405.04517*.
- Donuk, K., & Hanbay, D. (2022). Konuşma Duygu Tanıma İçin Akustik Özelliklere Dayalı LSTM Tabanlı Bir Yaklaşım. *Computer Science*, Vol:7(Issue:2), 54-67. <https://doi.org/10.53070/bbd.1113379>
- Christop, I. (2024). nEMO: Dataset of Emotional Speech In Polish. *arXiv preprint arXiv:2404.06292*.
- Poria, S., Hazarika, D., Majumder, N., Naik, G., Cambria, E., & Mihalcea, R. (2018). Meld: A Multimodal Multi-Party Dataset For Emotion Recognition In Conversations. *arXiv preprint arXiv:1810.02508*. <https://doi.org/10.48550/arXiv.1810.02508>
- Burkhardt, F., Paeschke, A., Rolfes, M., Sendlmeier, W. F., & Weiss, B. (2005, September). A Database Of German Emotional Speech. *Interspeech* (Vol. 5, pp. 1517-1520).
- Parlak, C., Diri, B., & Gürgeç, F. (2014, September). A Cross-Corpus Experiment In Speech Emotion Recognition. *SLAM@INTERSPEECH* (pp. 58-61).
- S. Haq and P.J.B. Jackson, "Multimodal Emotion Recognition", *W. Wang (ed), Machine Audition: Principles, Algorithms and Systems, IGI Global Press*, ISBN 978-1615209194, chapter 17, pp. 398-423, 2010.
- Ahmed, N., Natarajan, T., & Rao, K. R. (1974). Discrete Cosine Transform. *IEEE Transactions On Computers*, 100(1), 90-93.
- Özseven, T. (2019). Konuşma Tabanlı Duygu Tanımda Ön İşleme ve Öznitelik Seçim Yöntemlerinin Etkisi. *Dicle Üniversitesi Mühendislik Fakültesi Mühendislik Dergisi*, 10(1), 99-112. doi: 10.24012/dumf.498727
- Çolakoğlu, E., Hızlısoy, S., & Arslan, R. S. (2021). Konuşmadan Duygu Tanıma Üzerine Detaylı bir İnceleme: Özellikler ve Sınıflandırma Metotları. *Avrupa Bilim ve Teknoloji Dergisi*, (32), 471-483.
- Dala, Ö. Ç. (2023). A Literature Review On Emotion Recognition In Speech. *Researcher*, 3(02), 46-52.
- Dikbıyık, E., Demir, Ö., & Doğan, B. (2022). Derin Öğrenme Yöntemleri ile Konuşmadan Duygu Tanıma Üzerine Bir Literatür Araştırması. *Gazi University Journal of Science Part C: Design and Technology*, 10(4), 765-791. <https://doi.org/10.29109/gujsc.1111884>
- M. Slaney, "Auditory Toolbox," Interval Research Corporation, Tech. Rep, 10(1998) 1194
- Kingma, D. P., & Ba, J. (2014). Adam: A Method For Stochastic Optimization. *arXiv preprint arXiv:1412.6980*. <https://doi.org/10.48550/arXiv.1412.6980>