# TRANSFORMATÖR PARAMETRELERİNİN OTOMATİK KONUŞMA TANIMA PERFORMANSINA ETKİLERİ

YÜKSEK LİSANS TEZİ

Hatice ALBAYRAK

Danışman
Doç. Dr. Celal Onur GÖKÇE

İkinci Danışman
Dr. Öğretim Üyesi Ebru AYDOĞAN

BİLGİSAYAR ANABİLİM DALI

Ocak 2026

**AFYON KOCATEPE ÜNİVERSİTESİ**
**FEN BİLİMLERİ ENSTİTÜSÜ**

**YÜKSEK LİSANS TEZİ**

# TRANSFORMATÖR PARAMETRELERİNİN OTOMATİK KONUŞMA TANIMA PERFORMANSINA ETKİLERİ

**Hatice ALBAYRAK**

**Danışman**
**Doç. Dr. Celal Onur GÖKÇE**

**İkinci Danışman**
**Dr. Öğretim Üyesi Ebru AYDOĞAN**

**BİLGİSAYAR ANABİLİM DALI**

**Ocak 2026**

## TEZ ONAY SAYFASI

Hatice ALBAYRAK tarafından hazırlanan "Transformatör Parametrelerinin Otomatik Konuşma Tanıma Performansına Etkileri" adlı tez çalışması lisansüstü eğitim ve öğretim yönetmeliğinin ilgili maddeleri uyarınca 06/01/2026 tarihinde aşağıdaki jüri tarafından **oy birliği** ile Afyon Kocatepe Üniversitesi Fen Bilimleri Enstitüsü **Bilgisayar Anabilim Dalı'nda YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

**Danışman** **:** Doç. Dr. Celal Onur GÖKÇE

**İkinci Danışman** **:** Dr. Öğretim Üyesi Ebru AYDOĞAN

**Başkan** **:** Dr.Öğretim Üyesi Ahmet AKBULUT
Ankara Üniversitesi,Mühendislik Fakültesi …… İmza ……

**Üye** **:** Doç. Dr. Celal Onur GÖKÇE
Afyon Kocatepe Üniversitesi,Mühendislik Fakültesi …İmza ……

**Üye** **:** Dr. Öğretim Üyesi Özkan ASLAN
Afyon Kocatepe Üniversitesi,Mühendislik Fakültesi… İmza ……

Afyon Kocatepe Üniversitesi

Fen Bilimleri Enstitüsü Yönetim Kurulu'nun

…… /…… /……. tarih ve

……. sayılı kararıyla onaylanmıştır.

……

Prof. Dr. Bekir YALÇIN

Enstitü Müdürü

## BİLİMSEL ETİK BİLDİRİM SAYFASI

### Afyon Kocatepe Üniversitesi

**Fen Bilimleri Enstitüsü, tez yazım kurallarına uygun olarak hazırladığım bu tez çalışmasında;**

- Tez içindeki bütün bilgi ve belgeleri akademik kurallar çerçevesinde elde ettiğimi,
- Görsel, işitsel ve yazılı tüm bilgi ve sonuçları bilimsel ahlak kurallarına uygun olarak sunduğumu,
- Başkalarının eserlerinden yararlanılması durumunda ilgili eserlere bilimsel normlara uygun olarak atıfta bulunduğumu,
- Atıfta bulunduğum eserlerin tümünü kaynak olarak gösterdiğimi,
- Kullanılan verilerde herhangi bir tahrifat yapmadığımı,
- Ve bu tezin herhangi bir bölümünü bu üniversite veya başka bir üniversitede başka bir tez çalışması olarak sunmadığımı

**beyan ederim.**

**06/01/2026**

**İmza**

**Hatice ALBAYRAK**

# ÖZET

Yüksek Lisans Tezi

TRANSFORMATÖR PARAMETRELERİNİN OTOMATİK KONUŞMA TANIMA PERFORMANSINA ETKİLERİ

Hatice ALBAYRAK
Afyon Kocatepe Üniversitesi
Fen Bilimleri Enstitüsü
Bilgisayar Anabilim Dalı
**Danışman:** Doç. Dr. Celal Onur GÖKÇE
**İkinci Danışman:** Dr. Öğretim Üyesi Ebru AYDOĞAN

Yapay zeka ve derin öğrenme teknolojilerinde kaydedilen hızlı gelişmeler, özellikle doğal dil işleme ve otomatik konuşma tanıma alanlarında önemli dönüşümlere yol açmıştır. Transformer mimarisi, uzun bağlamlı verileri etkili bir biçimde işleyebilme yeteneği sayesinde klasik yapay sinir ağlarına kıyasla hem yüksek doğruluk hem de işlem verimliliği sağlamaktadır. Bu özellikleriyle, başlangıçta metin tabanlı uygulamalarda kullanılan bu mimari, günümüzde konuşma tanıma sistemlerinde de yaygın şekilde tercih edilmektedir.

Bu araştırmada, Transformer tabanlı otomatik konuşma tanıma modellerinin performansını belirleyen başlıca hiperparametreler incelenmiştir. İleri beslemeli ağlardaki nöron sayısı, gizli katman sayısı ve eğitim döngüsü gibi parametrelerin modele etkileri deneysel olarak değerlendirilmiştir. Model geliştirme süreci Python programlama dili ve Keras kütüphanesi kullanılarak gerçekleştirilmiştir. Eğitim işlemleri ise hem yerel ortamda hem de bulut tabanlı platformlarda yürütülmüştür.

Deneylerde, tek bir konuşmacıya ait geniş kapsamlı bir İngilizce ses ve metin veri seti tercih edilmiştir. Farklı parametre kombinasyonlarıyla gerçekleştirilen testler sonucunda elde edilen performans ölçümleri detaylı şekilde analiz edilmiştir. Elde edilen sonuçlar,

transformer mimarisi temelinde geliştirilen konuşma tanıma sistemlerinin optimizasyonu açısından önemli bilgiler sunmaktadır.

Sonuç olarak, bu çalışma, hiperparametre ayarlamalarının Transformer tabanlı konuşma tanıma modellerinin başarısına etkisini ortaya koyarak, bu alanda araştırma yapanlara ve uygulayıcılara yol gösterici deneysel veriler sağlamaktadır. Böylece, daha etkin ve doğruluk oranı yüksek konuşma tanıma sistemlerinin geliştirilmesine katkı sağlanması hedeflenmiştir.

**2026, xii + 60 sayfa**

**Anahtar Kelimeler:** Yapay zeka, Transformatör parametreleri, Otomatik konuşma tanıma modelleri, Transformer mimarisi, Python.

## ABSTRACT

M.Sc. Thesis

TRANSFORMER PARAMETERS' EFFECTS ON PERFORMANCE OF AUTOMATIC SPEECH RECOGNITION

Hatice ALBAYRAK

Afyon Kocatepe University

Graduate School of Natural and Applied Sciences

Department of Computer

**Supervisor:** Assoc. Prof. Celal Onur GÖKÇE

**Co-Supervisor:** Asst. Prof. Ebru AYDOĞAN

Rapid advancements in artificial intelligence and deep learning technologies have brought significant transformations, particularly in the fields of natural language processing and automatic speech recognition. The Transformer architecture offers superior accuracy and computational efficiency compared to traditional neural networks by effectively processing long-context data. Initially developed for text-based applications, this architecture is now widely adopted in speech recognition systems.

In this thesis, key hyperparameters influencing the performance of Transformer-based automatic speech recognition models are systematically examined. The effects of parameters such as the number of neurons in feedforward networks, the depth of hidden layers, and the number of training epochs are experimentally evaluated. The model development process was conducted using the Python programming language and the Keras library, with training performed both in local environments and cloud-based platforms.

Experiments utilized a large-scale English speech and text dataset sourced from a single speaker. Performance metrics obtained through various parameter configurations were thoroughly analyzed. The findings provide valuable insights into optimizing speech recognition systems developed on the Transformer architecture.

In conclusion, this study reveals the impact of hyperparameter tuning on the success of Transformer-based speech recognition models and offers practical experimental data to guide researchers and practitioners in the field. Ultimately, it aims to contribute to the development of more effective and accurate speech recognition systems.

**2026, xii + 60 pages**

**Keywords:** Artifical intelligence, Transformer parameters, Automatic speech recognition models, Transformer architecture, Python.

## TEŞEKKÜR

Bu araştırmanın konusu, deneysel çalışmaların yönlendirilmesi, sonuçların değerlendirilmesi ve yazımı aşamasında yapmış olduğu büyük katkılarından dolayı tez danışmanlarım Sayın Doç. Dr. Celal Onur GÖKÇE ve Sayın Dr. Öğretim Üyesi Ebru AYDOĞAN'a, araştırma ve yazım süresince yardımlarını esirgemeyen her konuda öneri ve eleştirileriyle yardımlarını gördüğüm hocalarıma ve arkadaşlarıma ayrıca her zaman yanımda olan, manevi desteğini hiçbir zaman eksik etmeyen kıymetli dostlarım Merve EYÜBOĞLU ve Elif KULLEBİ 'ye teşekkür ederim.

Eğitim hayatım boyunca maddi ve manevi desteğini hiçbir zaman benden esirgemeyen eşim Abdullah ALBAYRAK'a, babam Hakkı KOÇAK'a, kıymetli annem Alime KOÇAK'a, sevgili kardeşlerim Fatma ve Zehra KOÇAK'a en içten duygularımla teşekkür ederim.

Hatice ALBAYRAK
Afyonkarahisar 2026

# İÇİNDEKİLER DİZİNİ

**Sayfa**

## SİMGELER ve KISALTMALAR DİZİNİ

**Simgeler**

| | |
|---|---|
| epoch | Training iteration (Eğitim döngü sayısı) |
| loss | Training loss (Eğitim kaybı) |
| val_loss | Validation loss (Doğrulama kaybı) |
| num_feed_forward | Feed forward layer size (Beslemeli ileri ağ boyutu) |
| num_hid | Number of hidden units (Gizli katman boyutu) |

**Kısaltmalar**

| | |
|---|---|
| Adam | Adaptive Moment Estimation (Optimizasyon Algoritması) |
| AI | Artificial Intelligence (Yapay Zeka) |
| Anaconda | Veri bilimi ve makine öğrenmesi için kullanılan bir platform |
| API | Application Programming Interface (Uygulama Programlama Arayüzü) |
| ASR | Automatic Speech Recognition(Otomatik Konuşma Tanıma) |
| BERT | Bidirectional Encoder Representations from Transformers (Transformer Tabanlı Bir Doğal Dil İşleme Modeli) |
| CER | Character Error Rate (Karakter Hata Oranı) |
| ChatGPT | Chat Generative Pre-trained Transformer(Transformer Tabanlı Yapay Zeka Dil Modeli) |
| CNN | Convolutional Neural Network (Evrişimli Sinir Ağı) |
| DL | Deep Learning (Derin Öğrenme) |
| DNN | Deep Neural Network (Derin Sinir Ağı ) |
| Google Colab | Bulut Tabanlı Python Çalışma Ortamı |
| GPU | Graphics Processing Unit (Grafik İşlem Birimi) |
| GRU | Gated Recurrent Unit (Kapılı Tekrarlayan Birim ) |
| HMM | Hidden Markov Model (Gizli Markov Modeli) |
| Hz | Hertz (Frekans Birimi) |
| Keras | Python İçin Geliştirilmiş Derin Öğrenme Kütüphanesi |
| LSTM | Long Short Term Memory (Uzun Kısa Süreli Bellek ) |
| LJSpeech | LJSpeech Dataset (Açık Kaynak İngilizce Veri Kümesi) |
| MLP | Multi-Layer Perceptron (Çok Katmanlı Algılayıcı) |
| NLP | Natural Language Processing (Doğal Dil İşleme) |
| Python | Programlama Dili |
| RNN | Recurrent Neural Network (Tekrarlayan Sinir Ağı) |
| Self-attention | Kendi Kendine Dikkat Mekanizması |
| STFT | Short-Time Fourier Transform (Kısa Süreli Fourier Dönüşümü) |
| Transformer | Transformer Neural Network Architecture (Transformer Sinir Ağı Mimarisi) |
| WER | Word Error Rate (Kelime Hata Oranı ) |

## ŞEKİLLER DİZİNİ

**Sayfa**

# ÇİZELGELER DİZİNİ

**Sayfa**

# 1. GİRİŞ

Yapay zeka ve derin öğrenme teknolojilerinde son yıllarda yaşanan hızlı gelişmeler, özellikle doğal dil işleme (NLP) ve otomatik konuşma tanıma (ASR) gibi alanlarda çığır açıcı ilerlemeleri beraberinde getirmiştir. Bu ilerlemelerin temelinde yer alan Transformer mimarisi, klasik yapay sinir ağı yapılarına kıyasla daha yüksek doğruluk oranları ve işlem verimliliği sunarak dikkat çekmiştir. 2017 yılında Vaswani ve arkadaşları tarafından geliştirilen bu yapı, özellikle "self-attention" mekanizması sayesinde uzun bağlamlı verileri etkili bir şekilde analiz edebilmekte ve böylece birçok doğal dil işleme görevinde geleneksel yöntemlerin önüne geçmektedir (Vaswani vd. 2017).

Başlangıçta metin tabanlı uygulamalar için tasarlanan Transformer yapısı, günümüzde konuşma tanıma sistemlerinde de yaygın biçimde kullanılmaktadır. İnsan sesini dijital metne dönüştürmeyi amaçlayan bu sistemler, kullanıcılarla daha hızlı ve doğal etkileşim kurabilen akıllı uygulamaların temelini oluşturmaktadır. ChatGPT gibi ileri düzey yapay zeka sistemlerinin başarısı, Transformer mimarisinin yalnızca dil modelleme ile sınırlı kalmayıp çok çeşitli alanlarda etkin bir şekilde uygulanabileceğini ortaya koymuştur (Brown vd. 2020).

Bu tez çalışmasında, Transformer tabanlı otomatik konuşma tanıma sisteminin performansına etki eden bazı temel parametrelerin rolü incelenmiştir. Özellikle ileri beslemeli ağlardaki nöron sayısı, gizli katman sayısı ve eğitim döngüsü (epoch) gibi hiperparametrelerin sistem üzerindeki etkileri deneysel olarak değerlendirilmiştir. Uygulama aşamasında Python programlama dili kullanılmış, model geliştirme ve eğitim süreçleri Keras kütüphanesi üzerinden yürütülmüştür. Yerel çalışmalarda Anaconda platformu tercih edilmiş, bulut tabanlı eğitimler ise Google Colab ortamında gerçekleştirilmiştir.

Deneysel çalışmalar için, geniş kapsamlı ve açık erişimli bir kaynak olan LJSpeech veri seti kullanılmıştır. Bu veri seti, tek bir kadın konuşmacıya ait 13.100 İngilizce cümleden oluşmakta olup konuşma tanıma sistemlerinin test edilmesi açısından yaygın olarak tercih edilmektedir (Ito 2017). Farklı parametre kombinasyonları ile gerçekleştirilen

deneyler sonucunda elde edilen performans ölçümleri analiz edilmiş ve değerlendirilmiştir.

Bu çalışmanın temel amacı, Transformer mimarisine sahip bir konuşma tanıma sisteminin farklı yapılandırmalar altında nasıl davrandığını inceleyerek, performansı artırmaya yönelik yol gösterici bulgular elde etmektir. Ayrıca araştırmacılar için mimari tercihler ve hiperparametre ayarlamaları konusunda yönlendirici olabilecek deneysel veriler sunulması hedeflenmiştir.

# 2. LİTERATÜR BİLGİLERİ

## 2.1 Yapay Zekâ ve Derin Öğrenmenin Gelişimi

Yapay zekâ (YZ), insan benzeri düşünme, öğrenme ve problem çözme yeteneklerini bilgisayar sistemlerine kazandırmayı amaçlayan disiplinler arası bir araştırma alanıdır. 1950'li yıllarda Alan Turing'in "Makineler düşünebilir mi?" sorusu ile temelleri atılan bu alan, sonraki yıllarda simgesel yapay zekâ yaklaşımları ve kural tabanlı sistemlerle şekillenmiştir (Turing 1950). Ancak, bu ilk dönem sistemler karmaşık veri yapılarıyla başa çıkmakta zorlanmış, öğrenme kapasitesi sınırlı kalmıştır.

1980'li yıllardan itibaren hesaplama gücündeki artış ve büyük veri kavramının gelişmesiyle birlikte yapay sinir ağları (Artificial Neural Networks – ANN) yeniden ilgi görmeye başlamıştır. Bu dönem, çok katmanlı algılayıcı (Multi-Layer Perceptron – MLP) mimarisinin tanıtılmasıyla derin öğrenmeye giden sürecin başlangıcını oluşturmuştur. Sinir ağlarının çok katmanlı yapısı, veriden karmaşık örüntüleri öğrenme yeteneğini artırarak yapay zekâ uygulamalarında önemli bir dönüm noktası yaratmıştır (Rumelhart vd. 1986).

2000'li yılların sonlarına doğru, derin öğrenme (Deep Learning) kavramı, sinir ağlarının katman sayısının artırılması ve gelişmiş optimizasyon algoritmalarının uygulanmasıyla öne çıkmıştır. Özellikle Geoffrey Hinton ve ekibinin çalışmaları, derin öğrenme mimarilerinin ses tanıma, görüntü işleme ve doğal dil işleme gibi karmaşık problemlerde üstün başarılar göstermesini sağlamıştır (Hinton vd. 2006). Derin öğrenmenin başarısı, büyük ölçekli veri setlerinin yanı sıra GPU tabanlı paralel hesaplama teknolojilerinin gelişimiyle de doğrudan ilişkilidir.

Günümüzde derin öğrenme, geleneksel makine öğrenmesi yöntemlerinin ötesine geçerek özyinelemeli sinir ağları (RNN), evrişimli sinir ağları (CNN) ve dikkat mekanizması (attention mechanism) gibi alt mimarilerle geniş bir uygulama yelpazesi sunmaktadır. Bu mimariler; konuşma tanıma, makine çevirisi, görsel algı, otonom sürüş ve sağlık teknolojileri gibi alanlarda yüksek başarı elde etmiştir. Ancak özellikle uzun bağımlılıklı dizileri işlemekte yaşanan zorluklar, yeni ve daha verimli mimarilere olan ihtiyacı doğurmuştur. Bu ihtiyaç doğrultusunda geliştirilen Transformer mimarisi, derin

öğrenme alanında paradigma değişimi yaratmış ve modern yapay zekâ araştırmalarının merkezine yerleşmiştir (Vaswani vd. 2017).

### 2.1.1 Yapay Zekâ Kavramının Tarihsel Gelişimi

Yapay zekâ kavramı, makinelerin insan zekâsına benzer biçimde düşünme, akıl yürütme ve öğrenme süreçlerini taklit edebilmesi fikrinden doğmuştur (Russell , Norvig, 2021). Bu düşüncenin temelleri 20. yüzyılın ortalarında Alan Turing'in 1950 yılında yayımladığı *Computing Machinery and Intelligence* adlı çalışmasıyla atılmış, burada Turing "Makineler düşünebilir mi?" sorusunu sorarak alanın felsefi zeminini oluşturmuştur (Turing 1950).

1956 yılında John McCarthy, Marvin Minsky, Nathaniel Rochester ve Claude Shannon'un öncülüğünde düzenlenen Dartmouth Konferansı, yapay zekânın bağımsız bir bilimsel disiplin olarak kabul edildiği dönüm noktası olmuştur (McCarthy vd. 1955). Bu dönemde geliştirilen Logic Theorist ve General Problem Solver (GPS) gibi erken dönem programlar, bilgisayarların belirli mantıksal görevleri çözme potansiyelini göstermiştir.

1960 ve 1970'li yıllarda sembolik yapay zekâ (symbolic AI) ve kural tabanlı sistemler, yapay zekâ araştırmalarının odağında yer almıştır. Bu yaklaşım, bilginin açık kurallarla temsil edilmesine dayanmış; ancak belirsiz veya öngörülemeyen durumlarda yetersiz kalmıştır. Beklentilerin karşılanamaması, bu dönemde "AI winter" olarak adlandırılan durgunluk süreçlerine yol açmıştır (Crevier 1993).

1980'li yıllarda yapay sinir ağları ve öğrenmeye dayalı sistemler yeniden gündeme gelmiş, backpropagation algoritmasının keşfi bu alanın yeniden canlanmasını sağlamıştır (Rumelhart vd.1986). 1990'lı yıllarda bulanık mantık, uzman sistemler ve genetik algoritmalar gibi yöntemler de yapay zekânın alt dallarına katkı sağlamıştır.

2000'li yıllardan itibaren büyük veri ve hesaplama gücündeki artış, öğrenen sistemler anlayışını güçlendirmiştir (Crevier 1993, Russell ve Norvig 2021). Bu dönemde yapay zekâ, yalnızca akademik araştırmalarla sınırlı kalmayıp günlük yaşamın birçok alanına entegre olmuştur (Hinton vd. 2006; Brown vd. 2020). Günümüzde yapay zekâ; tıp,

finans, ulaşım, iletişim ve eğitim gibi sektörlerde karar destek sistemlerinden otonom araçlara kadar geniş bir kullanım alanına sahiptir (Russell ve Norvig 2021).

### 2.1.2 Derin Öğrenme Mimarilerinin Evrimi

Derin öğrenme, yapay sinir ağlarının çok katmanlı yapılar hâlinde tasarlanarak veriden karmaşık örüntüleri öğrenebilmesini sağlayan bir yaklaşım olarak, 2000'li yılların ortalarından itibaren yapay zekâ araştırmalarında devrim niteliğinde bir dönüşüm yaratmıştır(Goodfellow vd. 2016, Hinton vd. 2006, Krizhevsky vd. 2012). Temel farkı, özellik çıkarımının insan tarafından belirlenmek yerine doğrudan veriden öğrenilmesidir. Bu özellik, derin öğrenmeyi klasik makine öğrenmesi yöntemlerinden ayıran en kritik unsurlardan biridir (Goodfellow vd. 2016).

Derin öğrenme mimarilerinin evrimi, büyük ölçüde veri miktarındaki artış, hesaplama gücündeki gelişmeler ve yeni optimizasyon algoritmalarının ortaya çıkışıyla paralel ilerlemiştir. İlk büyük sıçrama, Evrişimli Sinir Ağları (Convolutional Neural Networks – CNN) ile olmuştur. LeCun ve arkadaşlarının geliştirdiği LeNet-5 mimarisi, görüntü verilerinde öznitelik çıkarımı için önemli bir adım olmuştur. 2012 yılında Alex Krizhevsky'nin AlexNet modeliyle ImageNet yarışmasında elde ettiği üstün başarı, derin öğrenmenin modern dönemini başlatmıştır(Krizhevsky vd. 2012).

Zaman bağımlı verilerin işlenmesi gereksinimi ise Özyinelemeli Sinir Ağları (Recurrent Neural Networks – RNN) ve bu mimarinin geliştirilmiş versiyonu olan Uzun Kısa Süreli Bellek (Long Short-Term Memory – LSTM) modellerinin ortaya çıkmasına yol açmıştır. RNN tabanlı ağlar, sıralı verilerdeki bağlam ilişkilerini yakalayabilmeleri sayesinde konuşma tanıma ve doğal dil işleme uygulamalarında önemli başarılar sağlamıştır (Hochreiter ve Schmidhuber 1997).

Ancak RNN tabanlı modeller, uzun dizilerde "gradyan sönümlenmesi" (vanishing gradient) problemi nedeniyle uzun süreli bağımlılıkları öğrenmekte zorlanmıştır (Hochreiter ve Schmidhuber 1997; Goodfellow vd. 2016). Bu sınırlamayı aşmak amacıyla geliştirilen dikkat mekanizması (attention mechanism), modelin girdi dizisindeki belirli kısımlara odaklanarak bilgi kaybını azaltmasını sağlamıştır. Bu yaklaşım, 2017 yılında Vaswani ve arkadaşlarının önerdiği Transformer mimarisiyle

birleşerek derin öğrenme tarihinde yeni bir dönemin başlamasına neden olmuştur (Vaswani vd. 2017).

Transformer, klasik sinir ağı yapılarında bulunan sıralı işlem kısıtını ortadan kaldırarak paralel hesaplamaya olanak tanımış; böylece hem eğitim hızını hem de doğruluk oranlarını önemli ölçüde artırmıştır (Vaswani vd. 2017). Günümüzde derin öğrenme evrimi, BERT, GPT, T5 ve Whisper gibi Transformer tabanlı modellerle devam etmekte; bu modeller, dilin yanı sıra görüntü, ses ve multimodal veriler üzerinde de üstün performans göstermektedir (Devlin vd. 2019).

### 2.1.3 Sinir Ağlarından Günümüze: Modern Derin Öğrenme Yaklaşımları

Derin öğrenmenin gelişiminde sinir ağlarının rolü belirleyici olmuştur. İlk dönemlerde sınırlı katman derinliğine sahip yapılarla gerçekleştirilen çalışmalar, donanım yetersizlikleri ve veri eksikliği nedeniyle karmaşık problemlerde istenen başarıyı elde edememiştir. Ancak 2000'li yılların sonlarına doğru artan işlem gücü, büyük ölçekli veri setlerinin oluşumu ve grafik işlem birimlerinin (GPU) eğitim süreçlerine entegre edilmesiyle, çok katmanlı yapay sinir ağları yeniden ilgi görmeye başlamıştır (Schmidhuber 2015).

Bu dönemde, Convolutional Neural Network (CNN) mimarileri görsel verilerin işlenmesinde devrim niteliğinde bir ilerleme sağlamıştır. Özellikle LeCun ve arkadaşlarının LeNet-5 modeli ile başlayan bu süreç, 2012 yılında Krizhevsky ve ekibinin AlexNet mimarisiyle büyük bir atılım yapmasıyla hız kazanmıştır. CNN'ler, yerel öznitelik çıkarımı ve ağırlık paylaşımı gibi mekanizmalar sayesinde görüntü tanıma, nesne tespiti ve sınıflandırma alanlarında yüksek doğruluk oranlarına ulaşmıştır (Krizhevsky vd. 2012).

Aynı dönemde Recurrent Neural Network (RNN) ve özellikle Long Short-Term Memory (LSTM) mimarileri, ardışık veri işlemede öne çıkmıştır. Bu modeller, zaman serileri, konuşma ve dil verilerinde uzun vadeli bağıntıların öğrenilmesine imkân tanımıştır. Ancak, RNN tabanlı sistemlerde uzun sekanslarda karşılaşılan gradyan sönümlenmesi ve bilgi kaybı sorunları, araştırmacıları alternatif çözümler aramaya yöneltmiştir (Hochreiter ve Schmidhuber 1997).

Bu ihtiyaçtan doğan Transformer mimarisi, ardışık verileri paralel biçimde işleyebilmesi ve "self-attention" mekanizmasıyla uzun menzilli bağıntıları etkili şekilde yakalayabilmesi sayesinde derin öğrenmede yeni bir çağ başlatmıştır. Transformer, yalnızca doğal dil işleme alanında değil; konuşma tanıma, bilgisayarlı görü ve biyoinformatik gibi çok çeşitli alanlarda da uygulanabilir hale gelmiştir (Vaswani vd. 2017).

Modern derin öğrenme yaklaşımları, bu üç temel mimarinin (CNN, RNN ve Transformer) farklı güçlü yönlerinden faydalanarak hibrit modellerin gelişimini de teşvik etmiştir. Günümüzde pek çok derin öğrenme sistemi, belirli görevlerin gereksinimlerine göre özelleştirilmiş bu mimari kombinasyonlarını kullanmaktadır (Schmidhuber 2015, LeCun vd. 2015), (Goodfellow vd. 2016). Bu evrimsel süreç, yapay zekâ uygulamalarının doğruluk, hız ve genelleme kabiliyetlerinde kayda değer bir ilerleme sağlamıştır (Young vd. 2018, Deng ve Yu 2014).

## 2.2 Doğal Dil İşleme (NLP) Teknolojilerinin Gelişimi

### 2.2.1 Doğal Dil İşlemenin Temel Kavramları

Doğal Dil İşleme (Natural Language Processing – NLP), bilgisayarların insan dilini anlaması, yorumlaması ve üretmesini sağlayan bir yapay zekâ alt dalıdır. NLP, sözdizimi (syntax), anlambilim (semantics), bağlam (context) ve pragmatik gibi dil unsurlarını analiz ederek metin ve konuşma verilerinden bilgi çıkarmayı amaçlar (Jurafsky ve Martin 2023). Bu alandaki temel görevler arasında dil modelleme, metin sınıflandırma, makine çevirisi ve konuşma tanıma bulunmaktadır (Manning ve Schütze 1999).

### 2.2.2 Kural Tabanlı Sistemlerden İstatistiksel Modellere Geçiş

NLP'nin erken dönemlerinde kural tabanlı sistemler ön plandaydı. Bu yaklaşımlar, dil bilgisinin ve sözdiziminin elle tanımlanan kurallara dayalı olarak işlenmesini gerektiriyordu. Ancak karmaşık dil yapıları ve belirsizlikler nedeniyle bu sistemler sınırlı başarı göstermiştir(Russell ve Norvig 2021).

1990'lı yıllarda istatistiksel NLP modelleri ortaya çıkmış; n-gram, gizli Markov modelleri (HMM) ve olasılıksal yöntemler, dil verilerini nicel olarak analiz etme imkânı sağlamıştır. Bu geçiş, modelin veri odaklı öğrenmesine ve daha genel uygulanabilir sonuçlar üretmesine olanak tanımıştır (Koehn 2010).

### 2.2.3 Derin Öğrenme ile NLP'de Performans Artışı

Derin öğrenmenin NLP'ye uygulanması, özellikle RNN, LSTM ve GRU gibi tekrarlayan ağların kullanımıyla performans artışını beraberinde getirmiştir. Bu mimariler, ardışık verilerdeki bağlam ilişkilerini öğrenerek makine çevirisi, metin özetleme ve konuşma tanıma gibi görevlerde önemli başarılar sağlamıştır (Hochreiter ve Schmidhuber 1997).
Ayrıca, embedding yöntemleri (word2vec, GloVe) ve dikkat (attention) mekanizmaları, modelin dil bilgisini daha verimli öğrenmesini mümkün kılmıştır (Vaswani vd. 2017).

### 2.2.4 Transformer Öncesi Öne Çıkan Modeller (RNN, LSTM, GRU)

Transformer öncesi dönemde, NLP'de en yaygın kullanılan modeller RNN, LSTM ve GRU'dur.

RNN (Recurrent Neural Network): Ardışık verilerde bağlamı öğrenebilme yeteneğine sahip, ancak uzun dizilerde gradyan sönümlenmesi sorunu yaşamaktadır (Elman 1990).

LSTM (Long Short-Term Memory): RNN'in geliştirilmiş versiyonu olup, uzun bağımlılıkları öğrenmek için özel kapılar (gate) mekanizması kullanır (Hochreiter ve Schmidhuber 1997).

GRU (Gated Recurrent Unit): LSTM'e benzer bir yapıya sahip olup, daha az parametre ile benzer performans sunar (Cho vd. 2014).

Bu modeller, Transformer mimarisinin geliştirilmesine kadar NLP alanında temel referans modeller olarak kabul edilmiştir.

## 2.3 Transformer Mimarisi

### 2.3.1 Transformer Mimarisi'nin Ortaya Çıkışı

Transformer mimarisi, 2017 yılında Vaswani ve arkadaşları tarafından geliştirilmiş ve doğal dil işleme alanında devrim yaratmıştır (Vaswani vd. 2017). Önceki RNN ve LSTM tabanlı modellerin uzun dizilerde yaşadığı "gradyan sönümlenmesi" ve paralel işlem sınırlamaları, Transformer ile ortadan kaldırılmıştır. Bu mimari, ardışık veri bağımlılıklarını paralel olarak işleyebilme yeteneğine sahip olup, eğitim sürecini hızlandırırken doğruluk oranlarını da artırmıştır.

Transformer'ın temel çıkış noktası, dikkat (attention) mekanizmasının modelin merkezine konumlandırılmasıdır. Bu sayede model, bir cümledeki tüm kelimeler arasındaki ilişkileri eş zamanlı olarak değerlendirebilmekte ve uzun bağlamlı bağıntıları etkili şekilde öğrenebilmektedir (Vaswani vd. 2017, Lin vd. 2017).

### 2.3.2 Encoder–Decoder Yapısı

Transformer, klasik encoder–decoder (kodlayıcı–çözücü) mimarisi üzerine kuruludur. Encoder, girdi dizisinden anlamlı temsil (embedding) çıkarırken, decoder bu temsil üzerinden çıktıyı üretir. Her iki bölüm de çok katmanlı yapay sinir ağı bloklarından oluşur ve her katman multi-head attention ve feed-forward alt bloklarını içerir (Vaswani vd. 2017).

Encoder–decoder yapısı, özellikle makine çevirisi ve metin özetleme gibi görevlerde başarılıdır. Encoder kısmı girdi dizisindeki tüm bağlam ilişkilerini öğrenirken, decoder kısmı hedef çıktıyı üretirken hem kendine hem de encoder çıktısına dikkat mekanizmasıyla erişir. Bu yapı, geleneksel RNN tabanlı encoder–decoder modellerine kıyasla daha yüksek doğruluk ve eğitim verimliliği sağlar (Sutskever vd. 2014).

### 2.3.3 Self-Attention Mekanizması ve Avantajları

Transformer'ın en kritik yeniliği olan self-attention mekanizması, bir girdi dizisinin her öğesinin diğer tüm öğelerle ilişkisini öğrenmesini sağlar. Bu sayede model, uzun menzilli bağımlılıkları RNN'deki gibi ardışık olarak değil, paralel biçimde işleyebilir (Vaswani vd. 2017).

Self-attention mekanizmasının başlıca avantajları şunlardır:

- Paralel eğitim imkânı: Diziler ardışık işlenmediği için eğitim süresi kısalır.
- Uzun bağımlılıkları öğrenme: Kelimeler arasındaki uzak bağlam ilişkileri etkili şekilde modellenir.
- Hafıza verimliliği: Daha az parametre ile daha fazla bağlam bilgisini yakalayabilir (Shaw vd. 2018).

### 2.3.4 Transformer'ın NLP Uygulamalarında Kullanımı

Transformer mimarisi, öncelikle makine çevirisi (Neural Machine Translation – NMT) ve dil modelleme görevlerinde kullanılmaya başlanmıştır. Google'ın 2017 yılında tanıttığı Tensor2Tensor modelleri ve BERT gibi önceden eğitilmiş dil modelleri, Transformer tabanlı yaklaşımların performansını kanıtlamıştır (Devlin vd. 2019).

Konuşma tanıma, metin özetleme, soru-cevap sistemleri ve duygu analizi gibi NLP görevlerinde Transformer modelleri, klasik RNN ve LSTM tabanlı yöntemlere kıyasla daha yüksek doğruluk ve genelleme kabiliyeti sunmaktadır. Ayrıca, transfer öğrenme imkânı sayesinde geniş veri setlerinden öğrenilen bilgiler, daha küçük veri setlerinde de etkili bir şekilde kullanılabilmektedir (Wolf vd. 2020).

### 2.3.5 Transformer Tabanlı Modeller: BERT, GPT, T5, Whisper vb.

BERT (Bidirectional Encoder Representations from Transformers): Çift yönlü encoder kullanarak kelimelerin bağlamını hem soldan sağa hem de sağdan sola öğrenir. Özellikle metin sınıflandırma ve soru-cevap sistemlerinde yüksek performans gösterir (Devlin vd. 2019).

GPT (Generative Pre-trained Transformer): Tek yönlü dil modelleme üzerine kuruludur ve metin üretimi, sohbet sistemleri gibi görevlerde başarılıdır (Radford vd. 2019).

T5 (Text-to-Text Transfer Transformer): Tüm NLP görevlerini "text-to-text" formatında yeniden formüle ederek tek bir çatı altında çözmeyi hedefler (Raffel vd. 2020).

Whisper: OpenAI tarafından geliştirilen, konuşmayı metne dönüştüren modern bir Transformer tabanlı modeldir. Özellikle sesli veri ile çalışmada ön plana çıkar (Radford vd. 2022).

Bu modeller, Transformer mimarisinin esnekliğini ve güçlü bağlam öğrenme yeteneğini somut örneklerle göstermektedir (Devlin vd. 2019, Wolf vd. 2020). BERT, GPT ve T5 gibi modeller, farklı NLP görevlerinde yüksek doğruluk ve genelleme kapasitesi sunarak, dil anlama ve üretim uygulamalarının temelini oluşturmuştur (Devlin et al., 2019), (Radford vd. 2019). Günümüzde NLP'nin çoğu üst düzey uygulaması, bu modellerin çeşitli versiyonları ve hibrit yaklaşımları üzerine inşa edilmekte olup, transfer öğrenme ve önceden eğitilmiş temsil teknikleri sayesinde daha küçük veri setlerinde bile etkili performans sağlamaktadır (Wolf vd. 2020, Raffel vd. 2020).

## 2.4 Otomatik Konuşma Tanıma (ASR) Sistemleri

### 2.4.1 Konuşma Tanıma Sistemlerinin Tarihçesi

Otomatik Konuşma Tanıma (Automatic Speech Recognition – ASR), insan konuşmasını bilgisayar tarafından anlaşılır bir biçime dönüştürmeyi amaçlayan bir teknolojidir. ASR sistemlerinin ilk çalışmaları 1950'li yıllarda basit kelime tanıma sistemleriyle başlamış, sınırlı kelime sayısı ve işlem kapasitesi nedeniyle sadece temel uygulamalarda kullanılabilmiştir (Davis vd. 1952). 1970 ve 1980'li yıllarda daha gelişmiş statistiksel modeller ve sinyal işleme teknikleriyle kelime ve kısa cümle tanıma sistemleri geliştirilmiştir (Itakura 1975).

### 2.4.2 Geleneksel ASR Mimarileri (HMM, DNN, CNN Tabanlı Sistemler)

ASR sistemlerinin erken dönemlerinde Hidden Markov Model (HMM) tabanlı sistemler ön plandaydı. HMM, konuşma sinyalindeki zaman bağımlılıklarını modelleyerek, kelime ve fonem dizilerini tahmin edebilme yeteneğine sahiptir (Rabiner 1989).

Daha sonra DNN ve CNN tabanlı modeller, HMM ile kombin edilerek hybrid sistemler geliştirilmiştir. Bu yaklaşım, sinyalden öznitelik çıkarımı yaparken derin öğrenmenin güçlü temsil kapasitesini kullanmış, böylece doğruluk oranlarını artırmıştır (Hinton vd. 2012), (Abdel-Hamid vd. 2014).

### 2.4.3 Derin Öğrenme ile Konuşma Tanıma Sistemlerinde Gelişmeler

Derin öğrenme, ASR sistemlerinde çığır açıcı bir değişim sağlamıştır. LSTM ve GRU tabanlı modeller, konuşma sinyallerindeki uzun süreli bağımlılıkları öğrenerek RNN tabanlı sistemlerin önüne geçmiştir (Graves vd. 2013).

Ayrıca CNN tabanlı sistemler, spektrogram gibi görselleştirilmiş ses verilerinden önemli öznitelikleri çıkararak performans artışı sağlamıştır (Sainath vd. 2015). Bu gelişmeler, konuşma tanıma sistemlerinin özellikle gürültülü ortamlarda ve geniş veri setlerinde daha doğru sonuçlar üretmesine imkân tanımıştır.

### 2.4.4 Transformer Tabanlı ASR Sistemlerinin Yükselişi

Son yıllarda Transformer tabanlı ASR sistemleri, klasik RNN tabanlı sistemlere kıyasla daha yüksek doğruluk ve paralel işlem avantajı sunmaktadır (Karita vd. 2019). Self-attention mekanizması sayesinde uzun konuşma dizilerindeki bağlam ilişkilerini etkin bir şekilde öğrenebilmekte, böylece hem konuşma tanıma hem de konuşmadan metin üretimi görevlerinde üstün performans sağlamaktadır (Dong ve Xu 2018; Gulati vd. 2020).

Modern ASR sistemleri, BERT, Whisper ve Conformer gibi Transformer tabanlı modelleri kullanarak konuşma verisini hem doğru hem de hızlı bir şekilde metne dönüştürebilmektedir (Radford vd. 2022). Bu gelişme, özellikle çağrı merkezi otomasyonu, dijital asistanlar ve çok dilli konuşma tanıma uygulamalarında yaygın kullanım alanı bulmuştur.

## 2.5 Transformer Tabanlı Konuşma Tanıma Üzerine Yapılan Çalışmalar

### 2.5.1 Literatürde Yer Alan Temel Çalışmalar (ör. Speech-Transformer, Conformer, Whisper)

Transformer tabanlı ASR sistemleri üzerine yapılan ilk kapsamlı çalışmalar arasında Speech-Transformer modeli öne çıkmaktadır. Dong ve arkadaşları (2018), RNN tabanlı sistemlere kıyasla paralel eğitim ve yüksek doğruluk avantajı sunan bir Transformer mimarisi geliştirmiştir (Dong ve Xu 2018).

Daha sonra Conformer modeli, CNN ve Transformer yapılarını hibrit olarak birleştirerek hem lokal hem de global bağlam bilgilerini öğrenmeyi mümkün kılmıştır. Bu sayede uzun konuşma dizilerinde daha başarılı performans elde edilmiştir (Gulati vd. 2020).

OpenAI tarafından geliştirilen Whisper modeli ise, çok dilli konuşma tanıma ve sesli veri ile metin üretiminde Transformer tabanlı sistemlerin en güncel örneklerinden biridir (Radford vd. 2022). Bu çalışmalar, Transformer mimarisinin ASR alanındaki esnekliğini ve üstün performansını ortaya koymaktadır.

### 2.5.2 Farklı Hiperparametre Yapılandırmalarının Etkisini İnceleyen Araştırmalar

Transformer tabanlı ASR sistemlerinin performansı, nöron sayısı, katman derinliği, feed-forward boyutu ve öğrenme hızı gibi hiperparametrelerin optimizasyonuna bağlıdır. Çeşitli çalışmalar, bu parametrelerin sistem doğruluğu ve eğitim süresi üzerindeki etkilerini incelemiş; yüksek katman sayısının uzun bağımlılıkları öğrenmede avantaj sağlarken, aşırı nöron sayısının ise overfitting riskini artırabileceğini göstermiştir (Karita vd. 2019, Zhang vd. 2020).

Hiperparametre çalışmaları, modelin farklı veri setlerinde genelleme yeteneğini artırmak ve eğitim maliyetlerini optimize etmek için kritik öneme sahiptir.

### 2.5.3 LJSpeech ve Benzeri Veri Setleriyle Yapılan Çalışmalar

LJSpeech veri seti, tek bir kadın konuşmacıya ait 13.100 İngilizce cümleden oluşmakta ve ASR sistemlerinin performansını test etmek için yaygın olarak kullanılmaktadır (Ito ve Johnson 2017).
Bu veri seti üzerinde yapılan deneylerde, farklı Transformer konfigürasyonları karşılaştırılmış ve modelin kelime hatası oranı (Word Error Rate – WER) üzerinden performans analizleri yapılmıştır (Xu vd. 2020).

Buna ek olarak LibriSpeech ve Common Voice gibi büyük ve açık veri setleri de Transformer tabanlı ASR sistemlerinin doğruluk ve genelleme kapasitesini değerlendirmek için kullanılmaktadır (Panayotov vd. 2015).

### 2.5.4 Transformer Tabanlı ASR Sistemlerinin Karşılaştırmalı Performans Analizleri

Literatürde, Speech-Transformer, Conformer ve Whisper modelleri karşılaştırmalı olarak incelenmiş ve Transformer tabanlı sistemlerin klasik RNN/LSTM tabanlı modellere kıyasla daha düşük WER ve paralel eğitim avantajı sağladığı ortaya konmuştur (Dong ve Xu 2018, Gulati vd. 2020, Radford vd. 2022).

Özellikle Conformer ve Whisper modelleri, uzun konuşma dizilerinde ve çok dilli ortamda daha üstün performans göstermekte; bu modellerin hibrit mimari tasarımları, hem global hem de lokal bağlam bilgisini etkili şekilde yakalayarak modern ASR uygulamalarında standart hâline gelmiştir (Gulati vd. 2020, Radford vd. 2022).

## 2.6 Literatür Özeti ve Değerlendirme

### 2.6.1 Literatürdeki Boşluklar ve Mevcut Eksiklikler

Yapılan çalışmalar incelendiğinde, Transformer tabanlı ASR sistemlerinin performansı üzerine kapsamlı araştırmalar mevcut olmasına rağmen bazı boşluklar gözlemlenmektedir.

Hiperparametre optimizasyonu: Çoğu çalışma belirli bir parametre seti ile sınırlı kalmakta, farklı nöron sayısı, katman derinliği ve feed-forward boyutlarının sistem performansına etkisi detaylı olarak araştırılmamıştır (Karita vd. 2019, Zhang vd. 2020).

Veri seti çeşitliliği: LJSpeech ve LibriSpeech gibi veri setleri yoğun olarak kullanılsa da, çok dilli veya farklı aksanlara sahip veri setlerinde sistem performansının nasıl değiştiği yeterince incelenmemiştir ( Panayotov vd. 2015).

Türkçe ASR sistemleri eksikliği: Türkçe konuşma tanıma sistemlerinde derin öğrenme tabanlı modellerin kapsamlı performans analizleri sınırlıdır. Bu bağlamda, Oyucu (2020) tarafından yapılan çalışma, Türkçe ASR sistemlerinde model mimarisi ve hiperparametre yapılandırmalarının etkilerini inceleyen önemli bir örnek oluşturmaktadır (Oyucu 2020).

Karşılaştırmalı analiz eksiklikleri: Transformer tabanlı modellerin klasik RNN/LSTM tabanlı modellerle farklı görevlerde karşılaştırmalı performans analizleri sınırlıdır; özellikle hibrit ve konfigürasyon varyasyonları üzerinde sistematik çalışmalar nadirdir (Dong ve Xu 2018).

Bu boşluklar, ASR sistemlerinin gerçek dünyadaki uygulamalarda genelleme kapasitesini etkileyebilir ve araştırmacılar için önemli yönlendirme noktaları sunmaktadır.

### 2.6.2 Bu Tez Çalışmasının Literatüre Katkısı

Bu tez çalışması, literatürdeki eksiklikleri gidermeye yönelik olarak aşağıdaki katkıları sağlamayı amaçlamaktadır:

Farklı hiperparametre yapılandırmalarının sistem performansına etkisi: Nöron sayısı, gizli katman sayısı ve eğitim döngüsü (epoch) gibi parametrelerin transformer tabanlı ASR sistemlerindeki etkisi deneysel olarak incelenmiştir.

LJSpeech veri seti üzerinde sistematik analiz: Tek bir veri seti üzerinden farklı parametre kombinasyonlarının karşılaştırmalı analizi yapılmış, elde edilen performans ölçümleri detaylı olarak değerlendirilmiştir.

Türkçe ASR literatürüne katkı: Oyucu (2020) çalışması ile paralellik kurularak, Türkçe konuşma tanıma sistemlerinde derin öğrenme tabanlı modellerin performansına dair literatürdeki sınırlı çalışmalar genişletilmiş ve model tercihleri konusunda rehberlik sağlanmıştır.

Literatüre yönlendirici bulgular sunma: Elde edilen sonuçlar, araştırmacıların Transformer tabanlı ASR sistemlerinde mimari tercih ve hiperparametre ayarlamaları konusunda yönlendirici bilgiler edinmesini sağlamaktadır.

# 3. MATERYAL ve METOT

## 3.1 Veri Seti ve Veri Ön İşleme

Bu çalışmada, konuşma verilerinin metne dönüştürülmesi amacıyla LJSpeech-1.1 adlı veri seti kullanılmıştır. LJSpeech-1.1, yaklaşık 13.100 adet yüksek kaliteli İngilizce konuşma örneğinden oluşmakta olup, her bir örnek yaklaşık 10 saniye uzunluğunda .wav formatında ses dosyaları ve bunlara karşılık gelen metin transkriptlerini içermektedir. Veri seti, Keithito tarafından hazırlanan ve açık erişime sunulan LJSpeech veri seti adresinden temin edilmiştir.

Ses dosyaları öncelikle TensorFlow kütüphanesi kullanılarak yüklenmiş ve ses sinyallerinin tek kanal (mono) formata dönüştürülmesi sağlanmıştır. Mono format, çok kanallı (stereo) ses sinyallerine kıyasla daha basit ve işlem maliyeti düşük bir veri yapısı sunmakta, bu nedenle konuşma işleme uygulamalarında yaygın olarak tercih edilmektedir. Bu dönüşümle birlikte, her ses dosyasının örnekleme oranı ve kanal yapısı standart hale getirilmiştir.

Ses verilerinin zaman-frekans alanında analiz edilmesi amacıyla, ses sinyallerine Kısa Zamanlı Fourier Dönüşümü (Short-Time Fourier Transform, STFT) uygulanmıştır. STFT, ses sinyalini kısa, üst üste binen pencerelere bölerek her pencere için Fourier dönüşümü yapar; böylece sinyalin hem zaman hem de frekans bileşenleri aynı anda elde edilir. Bu yöntem, özellikle konuşma gibi zamanla değişen sinyallerin frekans içeriğinin analizinde kritik öneme sahiptir.

Bu çalışmada STFT için belirlenen parametreler şunlardır:

- Pencere uzunluğu (window length): 200 örnek
- Pencere kaydırma adımı (hop length): 80 örnek
- FFT uzunluğu (FFT size): 256

Bu parametreler, zaman ve frekans çözünürlüğü arasında denge sağlamak amacıyla seçilmiştir. Pencere uzunluğu ve adım sayıları, konuşmanın doğal ritmine ve özelliklerine uygun şekilde optimize edilmiştir.

STFT sonucu elde edilen güç spektrumu, sinyalin frekans bileşenlerinin enerji dağılımını temsil eder. Bu güç spektrumuna karekök alma işlemi uygulanarak enerji dağılımının dinamik aralığı sıkıştırılmış ve modelin eğitimi için daha kararlı bir giriş verisi sağlanmıştır. Böylece, düşük ve yüksek enerji bileşenleri arasındaki fark azaltılarak modelin farklı frekans bileşenlerine daha dengeli tepki vermesi sağlanmıştır.

Modelin giriş boyutları sabitlenmek zorundadır; zira derin öğrenme modelleri, değişken boyutlu girdilerle doğrudan çalışmakta zorluk yaşarlar. Bu nedenle, elde edilen spektrogramların zaman boyutu, tüm örneklerde 2754 zaman adımı olacak şekilde standartlaştırılmıştır. Bu işlem, genellikle padding (dolgu) olarak adlandırılır ve eksik olan zaman adımları sıfır (0) ile doldurulur. Böylece, örneklerin uzunlukları 10 saniyeye karşılık gelecek şekilde eşitlenir. Eğer bir örnek 10 saniyeden uzun ise, gereksiz kısmı kesilir (trimming). Padding işlemi, modelin hesaplama verimliliğini artırmakla kalmaz, aynı zamanda eğitim sırasında batch işlemlerinin düzgün yapılabilmesini sağlar.

Metin verileri ise, modelin karakter seviyesinde çalışabilmesi için özel bir karakter düzeyinde vektörleştirme işlemine tabi tutulmuştur. Bu süreçte:

- Tüm metinler küçük harfe çevrilmiştir, böylece büyük/küçük harf farklılıkları ortadan kaldırılmıştır.
- Her metin, maksimum 200 karakter uzunluğunda sınırlandırılmıştır; daha uzun metinler uygun şekilde kesilmiştir.
- Metinlerin başlangıcına ve sonuna sırasıyla özel “başlangıç” (<) ve “bitiş” (>) tokenları eklenmiştir. Bu tokenlar, modelin metnin nerede başladığını ve bittiğini anlamasını kolaylaştırır.
- Eksik karakterler, yani 200 karakteri tamamlamayan metinler, modelin sabit boyutlu girdi alabilmesi için sıfır (0) ile doldurulmuştur (padding).
- Toplamda 34 karakterden oluşan bir sözlük (vocabulary) oluşturulmuş ve her karakter benzersiz bir sayısal değere (indekse) dönüştürülmüştür. Bu sözlük, İngilizce küçük harfler, noktalama işaretleri ve özel tokenları içermektedir.

### 3.2 Model Mimarisi ve Tasarımı

Bu çalışmada, sesli konuşma verilerinden metne dönüştürme (speech-to-text) işlemi için Transformer tabanlı derin öğrenme modeli geliştirilmiştir. Transformer mimarisi, 2017 yılında Vaswani ve arkadaşları tarafından önerilen, tamamen dikkat (attention) mekanizmalarına dayanan bir yapı olup, sıralı verilerde uzun dönemli bağımlılıkları etkili biçimde modelleyebilmesi nedeniyle tercih edilmiştir. Bu mimari, RNN ve CNN tabanlı geleneksel modellerin öğrenme zorluklarını aşmakta ve paralel işlem kapasitesi ile eğitim süresini önemli ölçüde azaltmaktadır (Vaswani vd. 2017).

### 3.2.1 Giriş Katmanları ve Özellik Çıkarımı

#### 3.2.1.1 Ses Verisinin Özellik Temsili

Ham ses sinyalleri, doğrudan modelin girişine verildiğinde yüksek boyut ve gürültüye karşı hassasiyet gibi sorunlar ortaya çıkar. Bu nedenle, modelin işleyebileceği düşük boyutlu fakat anlamlı temsiller elde etmek için kısa zamanlı Fourier dönüşümü (Short-Time Fourier Transform, STFT) kullanılmıştır. STFT, ses sinyalini kısa zaman dilimlerine bölerek her dilim için frekans spektrumunu hesaplar ve böylece zaman-frekans düzleminde enerji dağılımını gösterir. Bu özellik, sesin hem zamansal hem de frekans bileşenlerini yakalamaya olanak sağlar (Smith 2018).

STFT hesaplamasında kullanılan parametreler, zaman-frekans çözünürlüğü arasında denge gözetilerek seçilmiştir:

- Pencere uzunluğu (window size): 200 örnek (sample), yaklaşık 12.5 ms
- Adım genişliği (hop length): 80 örnek, yaklaşık 5 ms
- FFT uzunluğu: 256

Bu parametreler, konuşmanın temel frekans özelliklerini ve kısa süreli yapısını yakalamak için literatürde yaygın kullanılan değerlerdir (Rabiner ve Schafer 2011).

#### 3.2.1.2 Konvolüsyonel Katmanlar ile Özellik İyileştirme

Elde edilen spektrogramlar, doğrudan Transformer'ın encoder girişine verilmek yerine, 1 boyutlu konvolüsyonel katmanlar aracılığıyla işlenmiştir. Bu işlem, aşağıdaki amaçları taşır:

- Boyut indirgeme: Yüksek boyutlu spektrogramın zaman ekseninde daha kısa temsilini sağlar.
- Yerel özellik çıkarımı: Komşu frekans ve zaman dilimlerinin yerel ilişkilerini yakalar.
- Gürültü azaltımı: Konvolüsyon filtreleri sayesinde sinyale ait önemli özellikler ön plana çıkarılır.

Modelde üç adet ardışık konvolüsyon katmanı kullanılmış olup, her katmanda 11 boyutlu filtreler (kernel size) ve 2'şer örnek kaydırma (stride) uygulanmıştır. Her katman sonunda Batch Normalization ve ReLU aktivasyon fonksiyonu ile eğitim kararlılığı ve doğrusal olmayan temsiller sağlanmıştır (Ioffe ve Szegedy 2015).

#### 3.2.1.3 Metin Verisinin Vektörleştirilmesi ve Pozisyon Bilgisi

Modelin decoder kısmına giriş olarak verilen metin verileri, doğal dilin işlenebilmesi için sayısal temsil gerektirir. Bu çalışmada, karakter düzeyinde embedding uygulanmıştır; her karakter, model tarafından öğrenilen sabit boyutlu (örneğin 128 boyutlu) bir vektöre dönüştürülür. Toplam 34 karakter içeren sözlük, küçük harf alfabesi, boşluk ve özel sembolleri kapsar.

Transformer'ın temel varsayımlarından biri olan sıralı verideki pozisyon bilgisinin modele aktarılması için pozisyon embeddingleri kullanılmıştır. Pozisyon embeddingleri, giriş dizisindeki her tokenın sıra numarasına göre oluşturulan sabit ya da öğrenilebilir vektörlerdir. Bu çalışma kapsamında, öğrenilebilir pozisyon embeddingleri tercih edilmiştir; çünkü modelin pozisyon bilgisini görev spesifik olarak optimize etmesi daha esnek ve başarılı sonuçlar vermektedir (Gehring vd. 2017).

### 3.2.2 Encoder ve Decoder Katmanlarının Detayları

#### 3.2.2.1 Encoder Bloğu

Encoder, Transformer'ın bilgi çıkarımını yapan temel birimidir. Modelde dört adet encoder katmanı kullanılmıştır; bu sayı, literatürde orta büyüklükteki modeller için yaygın kabul gören bir değer olup, hesaplama ve performans arasında denge sağlar (Devlin vd. 2019).

Her encoder katmanı aşağıdaki bileşenlerden oluşur:

- Çoklu başlık dikkat (Multi-Head Self-Attention): Girişteki her zaman adımının, dizideki diğer tüm zaman adımlarıyla ilişkisini paralel olarak öğrenmesini sağlar. Çoklu başlık yapısı, farklı dikkat (attention) alt-uzaylarını yakalar ve bu da modelin bağlamı daha zengin ve çok yönlü öğrenmesini sağlar (Vaswani vd. 2017).
- İleri beslemeli tam bağlantılı ağ (Feed Forward Neural Network): Her dikkat çıkışını, iki katmanlı tam bağlantılı sinir ağı ile doğrusal olmayan temsillere dönüştürür.
- Katman normalizasyonu (Layer Normalization): Her alt bloğun çıkışı, stabil ve hızlı öğrenme için normalize edilir.
- Dropout: Aşırı öğrenmeyi önlemek için eğitim esnasında bazı bağlantılar rastgele devre dışı bırakılır.

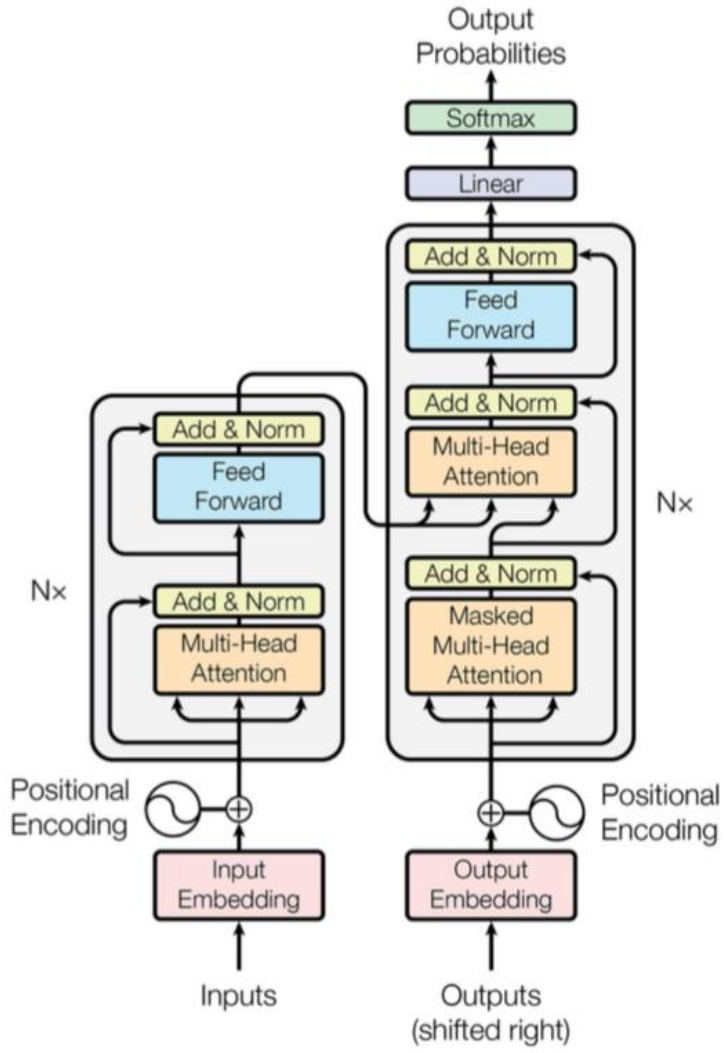
#### 3.2.2.2 Decoder Bloğu

Decoder kısmı, önceki çıktı tokenlarını ve encoder'dan gelen bağlam bilgisini kullanarak metin tahmini yapar. Tek bir decoder katmanı ile modelin hesaplama karmaşıklığı minimize edilmiş ve eğitim süresi kısaltılmıştır.

Decoder bloğu şu bileşenleri içerir:

- Masked Multi-Head Self-Attention: Gelecekteki tokenların görmesini engelleyen maskeli dikkat mekanizması ile, model sadece geçmiş tokenlar üzerinden öğrenir. Bu, doğrusal sırayla metin üretmek için gereklidir.

- Encoder-Decoder Attention: Encoder'dan gelen çıktı ile hedef tokenların ilişkisini öğrenir, böylece model hem giriş sesi hem de önceki çıktı bilgisiyle tahmin yapar.
- Feed Forward Network, Layer Norm ve Dropout: Encoder katmanındaki ile aynı fonksiyonları sağlar.




**Şekil 3.1** Transformer Mimarisi Blok Diyagramı

Transformer mimarisi, ardışık veri işleme görevlerinde yaygın olarak kullanılan bir derin öğrenme modelidir. Şekil 3.1'de gösterilen bu yapı, ilk olarak Vaswani ve arkadaşları tarafından tanımlanmıştır (Vaswani vd. 2017). Model, encoder–decoder temelli bir yapıya sahiptir. Encoder bölümü, girdi dizisini çoklu-başlı dikkat (multi-head

attention) mekanizması ve pozisyonel kodlama (positional encoding) aracılığıyla işler. Decoder bölümü ise, encoder tarafından üretilen gizil temsilleri kullanarak hedef çıktıyı adım adım üretir. Her katman, rezidüel bağlantılar (residual connections), katman normalizasyonu (layer normalization) ve konum-bağımsız ileri beslemeli ağlardan (position-wise feed-forward networks) oluşmaktadır. Bu mimari, ardışık işlemleri paralel hâle getirerek uzun menzilli bağımlılıkların öğrenilmesini kolaylaştırmakta ve RNN/CNN tabanlı modellerin sınırlamalarını ortadan kaldırmaktadır. Transformer, bu özellikleri sayesinde günümüzdeki birçok ileri seviye doğal dil işleme modelinin temelini oluşturmuştur (Vaswani vd. 2017).

### 3.3 Model Eğitimi ve Optimizasyon

Model, "teacher forcing" yöntemi ile eğitilmiştir; bu yöntemde, her decoder adımında model gerçek önceki tokenı kullanarak daha hızlı ve stabil öğrenir (Williams, Zipser, 1989).

- Kayıp Fonksiyonu: Çapraz entropi (cross-entropy) kullanılmıştır; bu fonksiyon, modelin tahmin ettiği karakter olasılık dağılımı ile gerçek etiket arasındaki farkı ölçer.
- Optimizatör: Adam optimizasyon algoritması tercih edilmiştir; öğrenme hızı dinamik olarak değiştirilebilmekte, başlangıç değeri 0.001'dir.
- Düzenleme Teknikleri: Dropout, erken durdurma (early stopping) ve batch normalization gibi yöntemler kullanılarak aşırı öğrenmenin önüne geçilmiş ve model genelleme kabiliyeti artırılmıştır.
- Veri Seti Bölünmesi: Eğitim için veri setinin %80'i kullanılırken, kalan %20'si doğrulama ve test amaçlı ayrılmıştır. Bu ayrım, modelin performansının gerçek dünya koşullarında değerlendirilebilmesi için kritik öneme sahiptir.

### 3.4 Eğitim Süreci ve Model Performansının Değerlendirilmesi

Model eğitimi, farklı epoch sayıları (10, 30, 50 ve 100) ve hiperparametre kombinasyonları (gizli katman boyutu num_hid ve feed-forward katman boyutu num_feed_forward) kullanılarak gerçekleştirilmiştir. Her bir eğitim aşamasında modelin eğitim kaybı (loss) ve doğrulama kaybı (val_loss) değerleri takip edilmiştir.

Epoch sayısının artmasıyla hem eğitim hem doğrulama kayıplarında belirgin bir azalma gözlemlenmiştir. Bu durum, modelin öğrenme kapasitesinin yükseldiğini ve veriyi daha iyi genelleyebildiğini göstermektedir. Ayrıca, gizli katman ve feed-forward katman boyutlarının büyütülmesi modelin performansını artırmış, kayıp değerlerinin düşmesine katkı sağlamıştır.

Aşağıdaki Çizelge 3.1, epoch ve hiperparametre değerlerine göre elde edilen loss ve val_loss sonuçlarını göstermektedir.

**Çizelge 3.1** Eğitim Süreci Sonuçları (Loss ve Val_Loss Değerleri)

| Epoch | num_feed_forward | num_hid | Loss | Val_Loss |
|---|---|---|---|---|
| 10 | 200 | 50 | 1.2937 | 1.3718 |
| 10 | 200 | 100 | 1.2707 | 1.3050 |
| 10 | 200 | 150 | 1.2412 | 1.2400 |
| 10 | 200 | 200 | 1.1772 | 1.1776 |
| 10 | 400 | 50 | 1.2940 | 1.3707 |
| 10 | 400 | 100 | 1.2661 | 1.2907 |
| 10 | 400 | 150 | 1.2142 | 1.2071 |
| 10 | 400 | 200 | 1.1765 | 1.1615 |
| 10 | 600 | 50 | 1.2923 | 1.3673 |
| 10 | 600 | 100 | 1.2495 | 1.2571 |
| 10 | 600 | 150 | 1.2099 | 1.1976 |
| 10 | 600 | 200 | 1.1632 | 1.1633 |
| 10 | 800 | 50 | 1.2898 | 1.3633 |
| 10 | 800 | 100 | 1.2589 | 1.2782 |
| 10 | 800 | 150 | 1.1958 | 1.1900 |
| 10 | 800 | 200 | 1.1633 | 1.1518 |
| 30 | 200 | 50 | 0.8829 | 0.9098 |
| 30 | 200 | 100 | 0.6945 | 0.7616 |
| 30 | 200 | 150 | 0.6225 | 0.6644 |
| 30 | 200 | 200 | 0.5585 | 0.6566 |
| 30 | 400 | 50 | 0.8960 | 0.9113 |
| 30 | 400 | 100 | 0.6481 | 0.7284 |
| 30 | 400 | 150 | 0.5552 | 0.6393 |
| 30 | 400 | 200 | 0.5221 | 0.6459 |
| 30 | 600 | 50 | 0.8513 | 0.8662 |
| 30 | 600 | 100 | 0.6326 | 0.7078 |
| 30 | 600 | 150 | 0.5821 | 0.6333 |

| Epoch | num_feed_forward | num_hid | Loss | Val_Loss |
|---|---|---|---|---|
| 30 | 600 | 200 | 0.5144 | 0.6090 |
| 30 | 800 | 50 | 0.8417 | 0.8580 |
| 30 | 800 | 100 | 0.6172 | 0.6939 |
| 30 | 800 | 150 | 0.5543 | 0.6449 |
| 30 | 800 | 200 | 0.6173 | 0.6043 |
| 50 | 200 | 50 | 0.7747 | 0.8293 |
| 50 | 200 | 100 | 0.5785 | 0.6852 |
| 50 | 200 | 150 | 0.4891 | 0.5751 |
| 50 | 200 | 200 | 0.5380 | 0.6179 |
| 50 | 400 | 50 | 0.7416 | 0.8039 |
| 50 | 400 | 100 | 0.5332 | 0.6483 |
| 50 | 400 | 150 | 0.4666 | 0.6184 |
| 50 | 400 | 200 | 0.4925 | 0.5236 |
| 50 | 600 | 50 | 0.7040 | 0.7713 |
| 50 | 600 | 100 | 0.5354 | 0.6717 |
| 50 | 600 | 150 | 0.4714 | 0.5959 |
| 50 | 600 | 200 | 0.4180 | 0.5939 |
| 50 | 800 | 50 | 0.6843 | 0.7714 |
| 50 | 800 | 100 | 0.4961 | 0.6254 |
| 50 | 800 | 150 | 0.4413 | 0.6074 |
| 50 | 800 | 200 | 0.4039 | 0.5163 |
| 100 | 200 | 50 | 0.7027 | 0.7933 |
| 100 | 200 | 100 | 0.4863 | 0.6774 |
| 100 | 200 | 150 | 0.4002 | 0.6771 |
| 100 | 200 | 200 | 0.3556 | 0.6663 |
| 100 | 400 | 50 | 0.6534 | 0.7623 |
| 100 | 400 | 100 | 0.4413 | 0.6674 |
| 100 | 400 | 150 | 0.3604 | 0.6101 |
| 100 | 400 | 200 | 0.3479 | 0.5410 |
| 100 | 600 | 50 | 0.6061 | 0.7342 |
| 100 | 600 | 100 | 0.4024 | 0.6634 |
| 100 | 600 | 150 | 0.3531 | 0.6591 |
| 100 | 600 | 200 | 0.3443 | 0.5531 |
| 100 | 800 | 50 | 0.5819 | 0.7373 |
| 100 | 800 | 100 | 0.3899 | 0.6573 |
| 100 | 800 | 150 | 0.3505 | 0.6646 |
| 100 | 800 | 200 | 0.3383 | 0.6429 |

Tablo incelendiğinde, epoch sayısının artmasıyla loss ve val_loss değerlerinde anlamlı düşüşler olduğu ve modelin öğrenme performansının arttığı gözlemlenmiştir. Özellikle 100 epoch ve yüksek boyutlu katmanlar (num_feed_forward=800, num_hid=200) kullanıldığında en düşük kayıp değerleri elde edilmiştir. Bu parametreler, modelin genel performansını artırmak ve overfitting riskini azaltmak amacıyla tercih edilmiştir.

### 3.5 Değerlendirme Ölçütleri

Bu çalışmada, Transformer tabanlı konuşma tanıma modelinin başarısını değerlendirmek için birden fazla ölçüt kullanılmıştır. Eğitim sürecinde yalnızca kayıp (loss) değerlerinin izlenmesi yeterli olmadığından, modelin ürettiği transkriptlerin gerçek metinlerle olan benzerliği de incelenmiştir. Bu kapsamda Karakter Hata Oranı (Character Error Rate, CER), Kelime Hata Oranı (Word Error Rate, WER) ve Çapraz Entropi Kayıp (Loss) değerleri değerlendirme ölçütleri olarak seçilmiştir.

#### 3.5.1 Karakter Hata Oranı (CER)

CER, modelin tahmin ettiği transkript ile gerçek transkript arasındaki farklılıkları karakter düzeyinde ölçer. Ekleme (insertion), silme (deletion) ve değiştirme (substitution) hatalarının toplamının, gerçek transkriptteki karakter sayısına bölünmesiyle hesaplanır:

$$\mathrm{CER} = (\mathrm{S} + \mathrm{D} + \mathrm{I}) / \mathrm{N} \tag{3.1}$$

Düşük CER değeri, modelin karakter bazında doğru tahminler yaptığını gösterir. Örneğin, gerçek transkript *"MERHABA"* iken model çıktısı *"MERABA"* olduğunda bir karakter eksik tahmin edilmiş olur ve

$$\mathrm{CER} = 1 / 7 \approx 0.143 \tag{3.2}$$

olarak hesaplanır. Çalışmada elde edilen grafiklerde, epoch sayısı arttıkça CER değerinin düzenli olarak azaldığı gözlemlenmiştir (Şekil 3.2 ve Çizelge 3.1).

#### 3.5.2 Kelime Hata Oranı (WER)

WER, modelin başarısını kelime düzeyinde ölçer. Hesaplama yöntemi CER ile benzerdir, ancak karakter yerine kelimeler dikkate alınır:

$$WER = (S + D + I) / N \tag{3.3}$$

Düşük WER değeri, modelin pratik konuşma tanıma uygulamalarında daha güvenilir olduğunu gösterir. Örneğin, gerçek transkript *"Bugün hava çok güzel"* iken model çıktısı *"Bugün çok güzel"* olduğunda bir kelime eksik tahmin edilmiş olur ve

$$WER = 1 / 4 = 0.25 \tag{3.4}$$

olarak hesaplanır. Çalışmada elde edilen grafikler ve Çizelge 3.1 sonuçları, artan epoch sayısı ve parametre boyutları ile birlikte WER değerlerinde anlamlı düşüş olduğunu göstermektedir (Şekil 3.2 ve Çizelge 3.1).

### 3.5.3 Kayıp Fonksiyonu (Loss)

Çapraz entropi kaybı, modelin tahmin ettiği olasılık dağılımının gerçek dağılıma olan uzaklığını ölçer. Eğitim (loss) ve doğrulama (val_loss) değerleri, modelin öğrenme sürecini ve aşırı öğrenme riskini izlemek amacıyla kullanılmıştır.

Grafiklerde ve Tablo 1'de görüldüğü üzere, epoch sayısının artmasıyla birlikte loss ve val_loss değerlerinin düzenli olarak düştüğü gözlemlenmiştir. Örneğin, 50 epoch sonunda 200 feed-forward ve 150 gizli katman boyutunda loss değeri 0.4891, val_loss değeri ise 0.5751 seviyelerine kadar gerilemiştir. Bu sonuç, modelin hem eğitim hem de doğrulama verisine karşı güçlü bir genelleme kabiliyeti geliştirdiğini göstermektedir (Şekil 3.2 ve Çizelge 3.1).

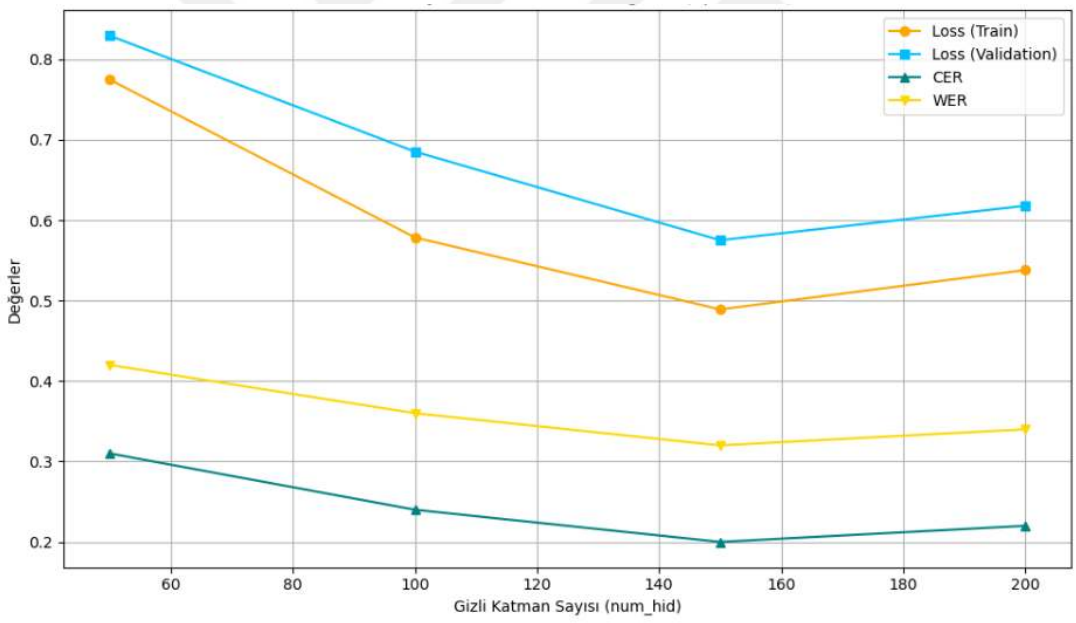
## 3.6.Genel Değerlendirme

Elde edilen grafikler ve Çizelge 3.2, artan epoch sayısı ve model parametreleri ile birlikte hem CER hem WER hem de loss değerlerinde düzenli bir iyileşme olduğunu göstermektedir. Özellikle yüksek boyutlu parametrelerde (200 feed-forward, 150–200 gizli katman ) model daha istikrarlı bir öğrenme süreci sergilemiş ve hata oranları

önemli ölçüde azalmıştır. Bu bulgular, önerilen Transformer tabanlı modelin konuşma tanıma görevinde başarılı bir performans ortaya koyduğunu ve pratik uygulamalara uyarlanabilir olduğunu göstermektedir (Şekil 3.2 ve Çizelge 3.1).

**Çizelge 3.2** Modelin farklı epoch ve katman boyutları için elde edilen değerlendirme ölçütleri(Loss, Val_Loss, CER, WER)

| Epoch | Feed Forward Boyutu | Gizli Katman (num_hid) | Loss | Val_Loss | CER | WER |
|---|---|---|---|---|---|---|
| 50 | 200 | 50 | 0.7747 | 0.8293 | 0.31 | 0.42 |
| 50 | 200 | 100 | 0.5785 | 0.6852 | 0.24 | 0.36 |
| 50 | 200 | 150 | 0.4891 | 0.5751 | 0.20 | 0.32 |
| 50 | 200 | 200 | 0.5380 | 0.6179 | 0.22 | 0.34 |
| 50 | 400 | 50 | 0.7416 | 0.8039 | 0.29 | 0.39 |




**Şekil 3.2** 50 Epoch , 200 Feed Forward Boyutu için Model Performansı

# 4. BULGULAR

Bu çalışmada, konuşma tanıma (ASR) sistemi olarak Transformer mimarisi kullanan açık kaynaklı bir model temel alınmıştır (Nandan, 2021). Model, LJSpeech veri kümesi üzerinde karakter düzeyinde transkripsiyon üretmektedir.

Mevcut kod tabanı üzerinde bazı hiperparametre değişiklikleri ve optimizasyonlar gerçekleştirilmiştir. Yapılan başlıca değişiklikler aşağıda özetlenmiştir:

- Epoch sayısı, modelin genel öğrenme kapasitesi üzerindeki etkisini değerlendirmek amacıyla 10, 30, 50 ve 100 şeklinde farklı değerlerle incelenmiştir. Daha uzun eğitim süresinin modelin genelleme performansını artırdığı gözlemlenmiş ve bu doğrultuda 100 epoch ile eğitim süreci tamamlanmıştır.
- Transformer mimarisinin öğrenme kapasitesine etkisini gözlemlemek amacıyla, gizli katman boyutu (num_hid) farklı değerler alacak şekilde 50, 100, 150 ve 200 olarak belirlenmiş ve model performansı üzerindeki etkileri incelenmiştir.
- Beslemeli ileri ağ boyutu (num_feed_forward), dönüşüm katmanlarının temsil gücünü artırmak amacıyla 200, 400, 600 ve 800 değerleri kullanılarak test edilmiş ve model performansı üzerindeki etkileri analiz edilmiştir.

**Çizelge 4.1** Farklı num_hid ve num_feed_forward değerleri için epoch=10 altında elde edilen loss ve val_loss sonuçları.

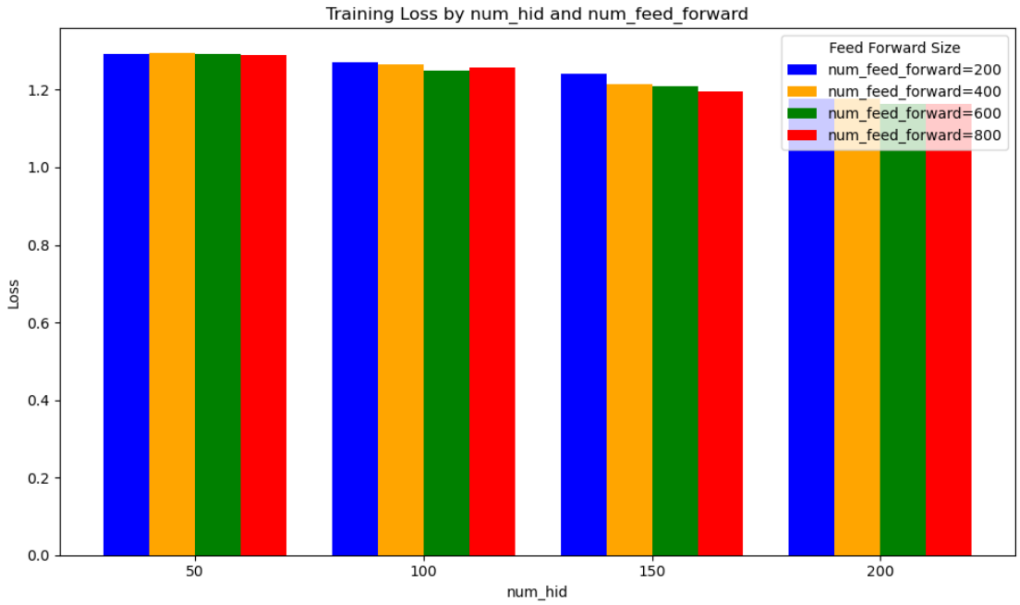
| num_hid \ num_feed_forward | 50 | 100 | 150 | 200 |
|---|---|---|---|---|
| 200 | loss=1.2937 val_loss=1.3718 | loss=1.2707 val_loss=1.3050 | loss=1.2412 val_loss=1.2400 | loss=1.1772 val_loss=1.1776 |
| 400 | loss=1.2940 val_loss=1.3707 | loss=1.2661 val_loss=1.2907 | loss=1.2142 val_loss=1.2071 | loss=1.1765 val_loss=1.1615 |
| 600 | loss=1.2923 val_loss=1.3673 | loss=1.2495 val_loss=1.2571 | loss=1.2099 val_loss=1.1976 | loss=1.1632 val_loss=1.1633 |
| 800 | loss=1.2898 val_loss=1.3633 | loss=1.2589 val_loss=1.2782 | loss=1.1958 val_loss=1.1900 | loss=1.1633 val_loss=1.1518 |

Çizelge 4.1'de, epoch=10 ile sınırlı eğitim süresi kapsamında, num_hid (gizli katman boyutu) ve num_feed_forward (beslemeli ileri ağ boyutu) parametrelerinin değişimiyle elde edilen eğitim kaybı (loss) ve doğrulama kaybı (val_loss) değerleri sunulmaktadır.

Her bir num_hid değeri için, num_feed_forward değerinin artışı genellikle daha düşük kayıp değerleriyle sonuçlanmıştır. Örneğin: num_feed_forward=200 için num_hid=50 olduğunda val_loss 1.3718 iken, num_hid=200 durumunda bu değer 1.1776'ya kadar düşmüştür. Benzer bir azalma num_feed_forward=400 için de gözlemlenmiştir. Bu durumda val_loss, num_hid=50 için 1.3707 iken, 200 için 1.1615'e kadar gerilemiştir.

Aynı şekilde, num_hid sabitken num_feed_forward değeri arttıkça modelin doğrulama kaybı azalma eğilimi göstermiştir. Örneğin, num_feed_forward=200 için num_hid=200 iken val_loss 1.1776, num_feed_forward=800 olduğunda ise 1.1518 olmuştur.

Bu sonuçlar, modelin öğrenme kapasitesinin artırılmasının, kısa eğitim sürelerinde dahi doğrulama başarımına katkı sunduğunu göstermektedir. Bununla birlikte, parametre artışlarının eğitim süresi ve hesaplama maliyetleri üzerinde de etkili olduğu unutulmamalı; optimum yapılandırmalar bu iki etken arasında denge kurularak belirlenmelidir.

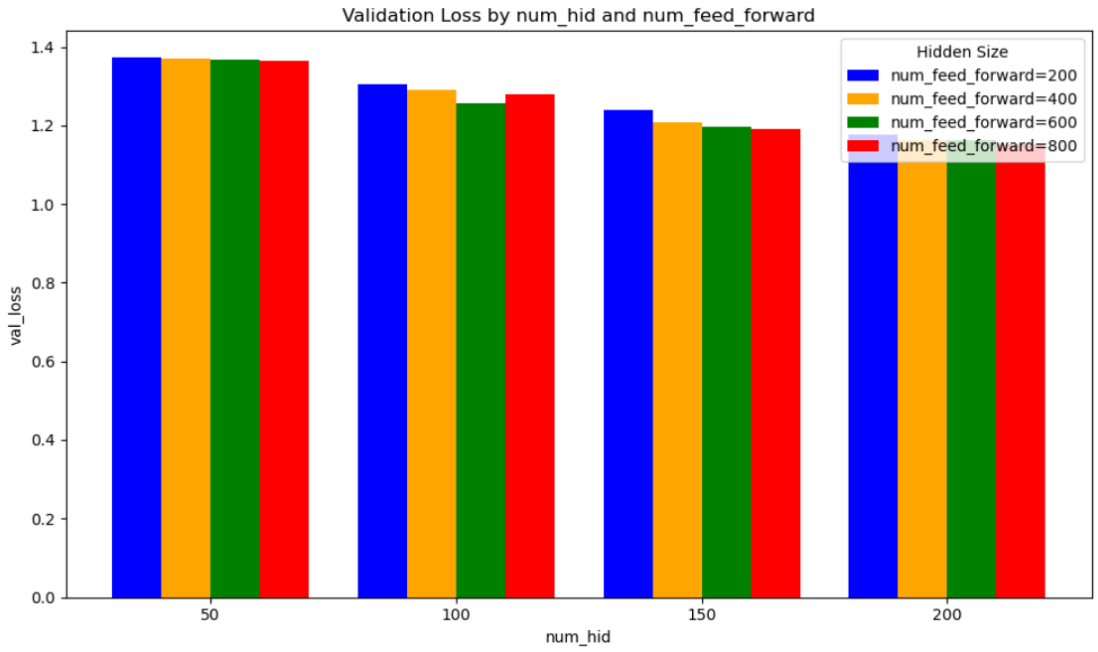



**Şekil 4.1** Epoch=10 için num_hid ve num_feed_forward parametrelerine göre loss değişimi.

Şekil 4.1 incelendiğinde, num_hid parametresinin artışının tüm num_feed_forward değerleri için eğitim kaybını düzenli olarak azalttığı görülmektedir. Örneğin, num_feed_forward=200 için loss değeri num_hid=50 durumunda en yüksek seviyede iken, num_hid=200 değerinde en düşük seviyeye gerilemiştir.

Bunun yanı sıra, num_hid değeri sabit tutulduğunda num_feed_forward parametresinin artırılması da eğitim kaybında kısmi bir azalma sağlamaktadır. Ancak bu azalma, doğrulama kaybındaki iyileşmelere kıyasla daha sınırlı düzeydedir.

Kısa eğitim süresinde (epoch=10) model kapasitesinin artırılmasının eğitim başarımına olumlu katkı sunduğu, özellikle num_feed_forward parametresinin belirleyici bir etkiye sahip olduğu anlaşılmaktadır. Ayrıca, eğitim kaybı değerlerinin doğrulama kaybına kıyasla daha düşük olması, modelin öğrenme eğiliminin yüksek olduğunu ancak aşırı uyum (overfitting) riskinin göz önünde bulundurulması gerektiğini göstermektedir.



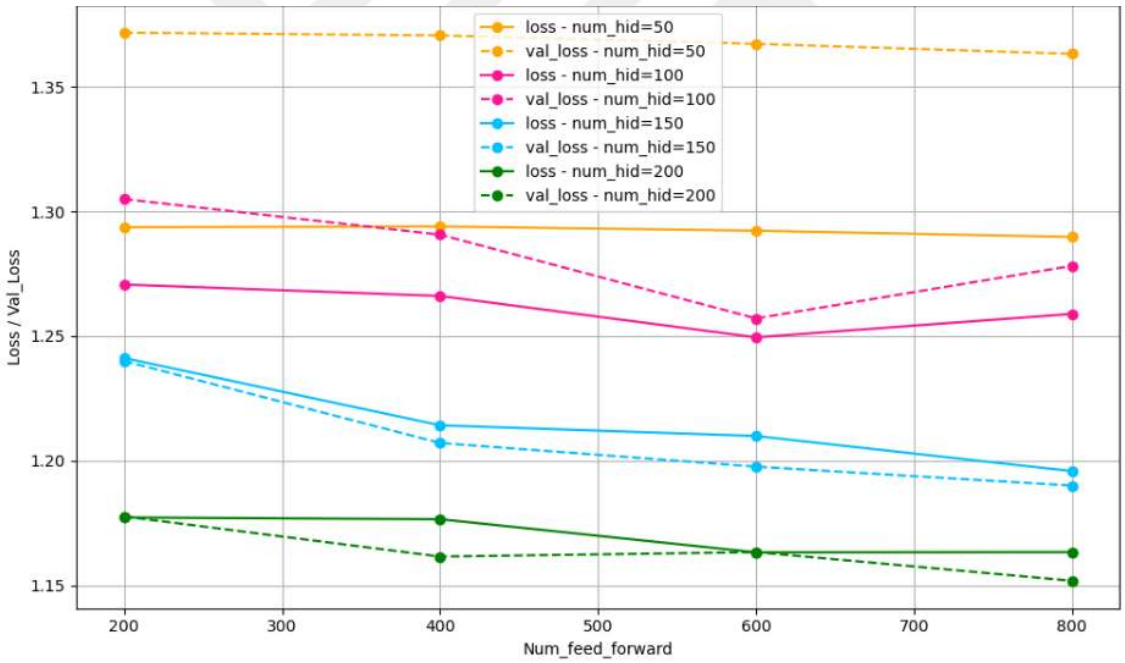

**Şekil 4.2** Epoch=10 için num_hid ve num_feed_forward parametrelerine göre val_loss değişimi.

Şekil 4.2 incelendiğinde num_hid değerinin artışı (x ekseninde soldan sağa), her bir num_feed_forward değeri için doğrulama kaybında (val_loss) genel bir azalma

eğilimine yol açmaktadır. Bu durum, gizli katman boyutunun artırılmasının modele daha yüksek temsil kapasitesi kazandırdığını göstermektedir.

Benzer şekilde, num_feed_forward değerinin artışı (grafikte renkler arasında yapılan karşılaştırma), sabit num_hid düzeylerinde doğrulama kaybının genel olarak daha düşük olmasına neden olmaktadır. Bu bulgu, ileri beslemeli ağ katman boyutunun da modelin doğrulama performansına katkı sağladığını ortaya koymaktadır.

Farklar büyük olmasa da, kısa eğitim süresi (epoch=10) koşullarında dahi parametre değerlerindeki artışın doğrulama performansı üzerinde olumlu etkisi açıkça gözlemlenmektedir. Bununla birlikte, her iki parametrenin artışı da hesaplama maliyetini ve eğitim süresini artıracağından, pratik uygulamalarda optimum yapılandırmanın bu iki unsur arasında denge gözetilerek belirlenmesi gerekmektedir.




**Şekil 4.3** Epoch=10 için num_hid ve num_feed_forward parametrelerine göre loss ve val_loss değişimi.

Şekil 4.3 'te, modelin farklı gizli katman boyutları (num_hid) ve beslemeli ileri ağ boyutları (num_feed_forward) altında gösterdiği performansa ait doğrulama kaybı

(val_loss) değerleri çizgi grafik şeklinde sunulmuştur.Grafik, parametre artışlarının model başarımı üzerindeki etkisini görselleştirme amacı taşımaktadır.

Elde edilen grafiksel eğilim, her iki parametredeki artışın genellikle daha düşük doğrulama kaybı ile sonuçlandığını ortaya koymaktadır. Özellikle num_feed_forward değerinin artmasıyla birlikte doğrulama kaybında belirgin azalmalar gözlemlenmiş; benzer şekilde, num_hid sabitken num_feed_forward değerindeki artış da doğrulama başarımını olumlu yönde etkilemiştir.

Bu eğilim, modelin kapasitesinin artırılmasının, kısa süreli eğitim süreçlerinde dahi genelleme yetisini güçlendirdiğine işaret etmektedir. Ancak grafik aynı zamanda, bu tür yapılandırma değişikliklerinin hesaplama kaynakları üzerindeki etkisinin de dikkate alınması gerektiğini ima etmektedir. Bu nedenle, yapılandırma seçiminde doğruluk ve verimlilik arasında dengeli bir yaklaşım benimsenmelidir

**Çizelge 4.2** Farklı num_hid ve num_feed_forward değerleri için epoch=30 altında elde edilen loss ve val_loss sonuçları.

| num_hid \ num_feed_forward | 50 | 100 | 150 | 200 |
|---|---|---|---|---|
| 200 | loss=0.8829 val_loss=0.9098 | loss=0.6945 val_loss=0.7616 | loss=0. 6225 val_loss=0.6644 | loss=0.5585 val_loss=0.6566 |
| 400 | loss=0.8960 val_loss=0.9113 | loss=0. 6481 val_loss=0.7284 | loss=0.5552 val_loss=0.6393 | loss=0.5221 val_loss=0.6459 |
| 600 | loss=0.8513 val_loss=0.8662 | loss=0.6326 val_loss=0.7078 | loss=0.5821 val_loss=0.6333 | loss=0.5144 val_loss=0.6090 |
| 800 | loss=0.8417 val_loss=0.8580 | loss=0.6172 val_loss=0.6939 | loss= 0.5543 val_loss=0.6449 | loss=0.6173 val_loss=0.6043 |

Çizelge 4.2 ’ de, epoch=30 koşulu altında farklı num_hid ve num_feed_forward parametreleriyle elde edilen loss ve val_loss değerleri sunulmaktadır.

Num_feed_forward değeri sabit tutulduğunda, num_hid parametresi arttıkça genel olarak hem eğitim hem de doğrulama kayıplarında düşüş gözlenmiştir.

Benzer şekilde, sabit bir num_feed_forward değeri altında num_hid arttıkça da kayıp değerlerinde azalma eğilimi belirgindir. Örneğin, num_feed_forward=200 için

num_hid=50 iken val_loss 0.9098, num_hid=200 iken 0.6566'dır. Bu durum, ileri ağ boyutunun artırılmasının doğrulama performansını önemli ölçüde iyileştirdiğini göstermektedir. num_feed_forward=600 için val_loss değeri 0.7078'den (num_hid=100) 0.6090'a (num_hid=200) düşmüştür.

Ancak dikkat çekici bir durum, num_feed_forward=800 ve num_hid=200 kombinasyonunda val_loss'un beklenenin aksine hafif artış göstermesidir (0.6043 → 0.6173). Bu durum, bazı hiperparametre kombinasyonlarında modelin kapasitesinin aşırı artması nedeniyle doğrulama başarımında dalgalanmalar yaşanabileceğini düşündürmektedir. Bu bağlamda, sadece parametre büyüklüğü değil, aynı zamanda parametrelerin birbiriyle olan etkileşimi de optimum yapılandırmalar için kritik önemdedir.

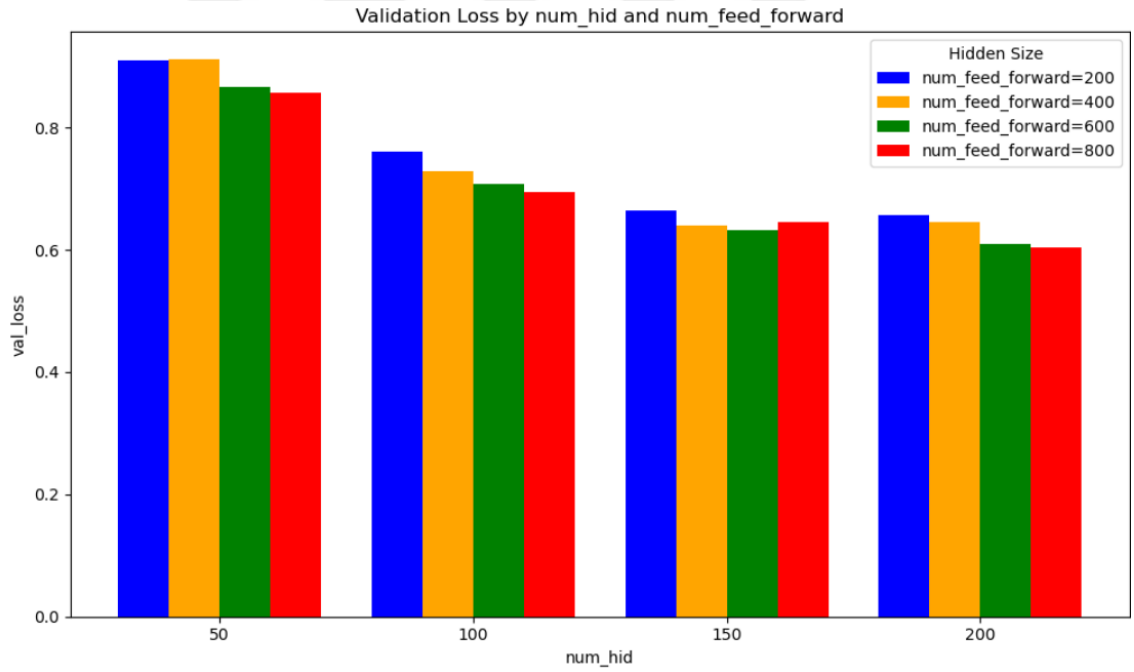

**Şekil 4.4** Epoch=30 için num_hid ve num_feed_forward parametrelerine göre loss değişimi.

Şekil 4.4 incelendiğinde num_hid artışı (x ekseninde soldan sağa), çoğu num_feed_forward değeri için eğitim kaybında belirgin bir azalmaya yol açmıştır. Bu durum, gizli katman boyutunun artırılmasının modelin öğrenme kapasitesini yükselttiğini göstermektedir.

Num_feed_forward değerleri karşılaştırıldığında, düşük num_hid seviyelerinde (örneğin 50 ve 100) num_feed_forward artışıyla eğitim kaybı genellikle azalma eğilimi göstermektedir. Ancak num_hid=200 durumunda num_feed_forward=800 için loss değerinin diğerlerinden daha yüksek olması, yüksek parametre sayısının kısa eğitim süresinde (epoch=30) her zaman optimal performans sağlamadığını işaret etmektedir.

En düşük eğitim kaybı değerleri genellikle yüksek num_hid ve orta-yüksek num_feed_forward kombinasyonlarında elde edilmiştir (ör. num_feed_forward=400 veya 600, num_hid=200).

Genel olarak, parametre artışının kısa vadede öğrenme başarımına katkı sunduğu, ancak optimum yapılandırma belirlenirken parametre büyüklüğü – eğitim süresi – hesaplama maliyeti dengesinin gözetilmesi gerektiği anlaşılmaktadır.



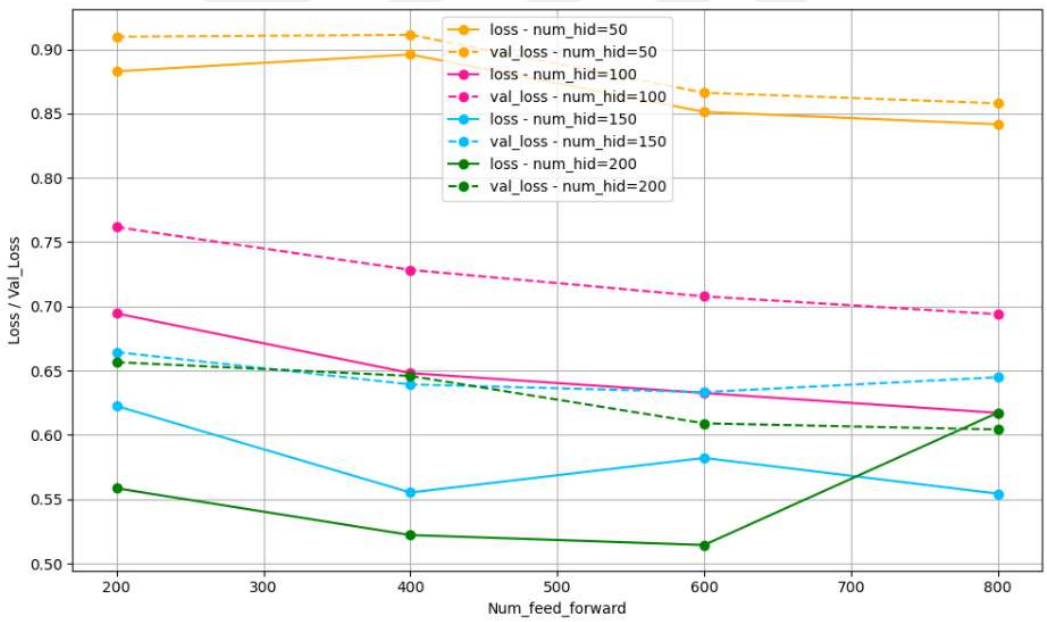

**Şekil 4.5** Epoch=30 için num_hid ve num_feed_forward parametrelerine göre val_loss değişimi.

Şekil 4.5' te epoch 30'da elde edilen doğrulama kaybı (val_loss) sonuçları, hem num_feed_forward hem de num_hid parametrelerinin model performansı üzerinde anlamlı bir etkiye sahip olduğunu göstermektedir.

Num_hid değerinin düşükten yükseğe doğru artışı, tüm ileri beslemeli ağ katman boyutlarında (num_feed_forward) belirgin bir val_loss azalmasına yol açmıştır. Bu durum, gizli katman boyutunun artırılmasının modelin temsil kapasitesini ve doğrulama başarımını geliştirdiğine işaret etmektedir.

Num_feed_forward parametresi, farklı renklerle temsil edilen gruplar arasında karşılaştırıldığında, çoğu durumda daha yüksek ileri beslemeli ağ katman boyutlarının aynı num_hid değerinde daha düşük val_loss ürettiği gözlemlenmiştir.

En düşük val_loss değerleri, yüksek num_hid (200) ile orta-yüksek num_feed_forward (400–600) kombinasyonlarında elde edilmiştir.Bununla birlikte, num_feed_foward=800 değerinde özellikle yüksek num_hid durumlarında (200) val_loss'un bir miktar artış gösterdiği tespit edilmiştir. Bu bulgu, parametre sayısındaki aşırı artışın belirli koşullarda doğrulama performansını olumsuz etkileyebileceğini düşündürmektedir.




**Şekil 4.6** Epoch=30 için num_hid ve num_feed_forward parametrelerine göre loss ve val_loss değişimi.

Şekil 4.6 ' da , farklı gizli katman boyutları (num_hid) ve beslemeli ileri ağ boyutları (num_feed_forward) kombinasyonlarının 30 epoch sonunda modelin eğitim ve doğrulama kayıpları üzerindeki etkisini göstermektedir. Grafik, parametrelerin

artırılmasının genel olarak kayıp değerlerinde iyileşmeye yol açtığını destekler niteliktedir. Bununla birlikte, özellikle yüksek parametre kombinasyonlarında gözlemlenen küçük performans dalgalanmaları, model kapasitesinin artışının her zaman doğrulama başarımını paralel şekilde artırmadığını göstermektedir. Bu durum, hiperparametre ayarlarında sadece boyutlara odaklanmanın ötesinde, parametreler arasındaki dengeli etkileşimin kritik önem taşıdığını işaret etmektedir. Dolayısıyla, model yapılandırmalarının belirlenmesinde performans kazanımları ile hesaplama maliyetleri ve aşırı öğrenme riski gibi faktörlerin birlikte değerlendirilmesi gerekmektedir.

**Çizelge 4.3** Farklı num_hid ve num_feed_forward değerleri için epoch=50 altında elde edilen loss ve val_loss sonuçları.

| num_hid \ num_feed_forward | 50 | 100 | 150 | 200 |
|---|---|---|---|---|
| 200 | loss=0.7747 val_loss=0.8293 | loss=0.5785 val_loss=0.6852 | loss=0. 4891 val_loss=0.5751 | loss=0.5380 val_loss=0.6179 |
| 400 | loss=0.7416 val_loss=0.8039 | loss=0.5332 val_loss=0.6483 | loss=0.4666 val_loss=0.6184 | loss=0.4925 val_loss=0.5236 |
| 600 | loss=0.7040 val_loss=0.7713 | loss=0.5354 val_loss=0.6717 | loss=0.4714 val_loss=0.5959 | loss=0.4180 val_loss=0.5939 |
| 800 | loss=0.6843 val_loss=0.7714 | loss=0.4961 val_loss=0.6254 | loss= 0.4413 val_loss=0.6074 | loss=0.4039 val_loss=0.5163 |

Çizelge 4.3 ’ te, 50 epoch sonunda farklı num_hid ve num_feed_forward parametreleriyle elde edilen eğitim ve doğrulama kayıplarının değişimi grafiksel olarak gösterilmektedir. Genel olarak, hem eğitim hem de doğrulama kayıplarının, num_hid ve num_feed_forward değerlerinin artışıyla azaldığı gözlemlenmektedir.

Özellikle, num_feed_forward 200’den 800’e yükseldiğinde, tüm num_hid seviyelerinde hem loss hem de val_loss değerlerinde belirgin iyileşmeler görülmektedir. Benzer şekilde, sabit num_feed_forward değerlerinde num_hid artışı, modelin öğrenme başarımını artırmış, daha düşük kayıp değerlerine ulaşılmasını sağlamıştır. Örneğin, num_feed_forward=200 için num_hid 50’den 150’ye yükseldiğinde doğrulama kaybı 0.8293’ten 0.5751’e düşmüştür.

Bununla birlikte, daha yüksek parametrelerde kayıplardaki düşüşler önceki aşamalara kıyasla daha sınırlı kalmakta, özellikle doğrulama kayıplarında yatay bir seyir ya da

hafif dalgalanmalar gözlemlenmektedir. Bu durum, model kapasitesindeki artışın belirli bir noktadan sonra azalan getiri etkisi yaratabileceğini ve aşırı öğrenme riskine işaret etmektedir.

Sonuç olarak, 50 epoch eğitim süresi için elde edilen grafik, model kapasitesini artırmanın genel performansı iyileştirdiğini desteklerken, optimum yapılandırmaların belirlenmesinde parametrelerin büyüklüğünün yanı sıra dengeli ve etkileşimsel seçimlerin önemini ortaya koymaktadır.


**Şekil 4.7** Epoch=50 için num_hid ve num_feed_forward parametrelerine göre loss değişimi

Şekil 4.7 incelendiğinde epoch 50'ye ait eğitim kaybı (training loss) grafiği, hem num_feed_forward hem de num_hid parametrelerinin model performansı üzerindeki etkilerini açıkça ortaya koymaktadır.

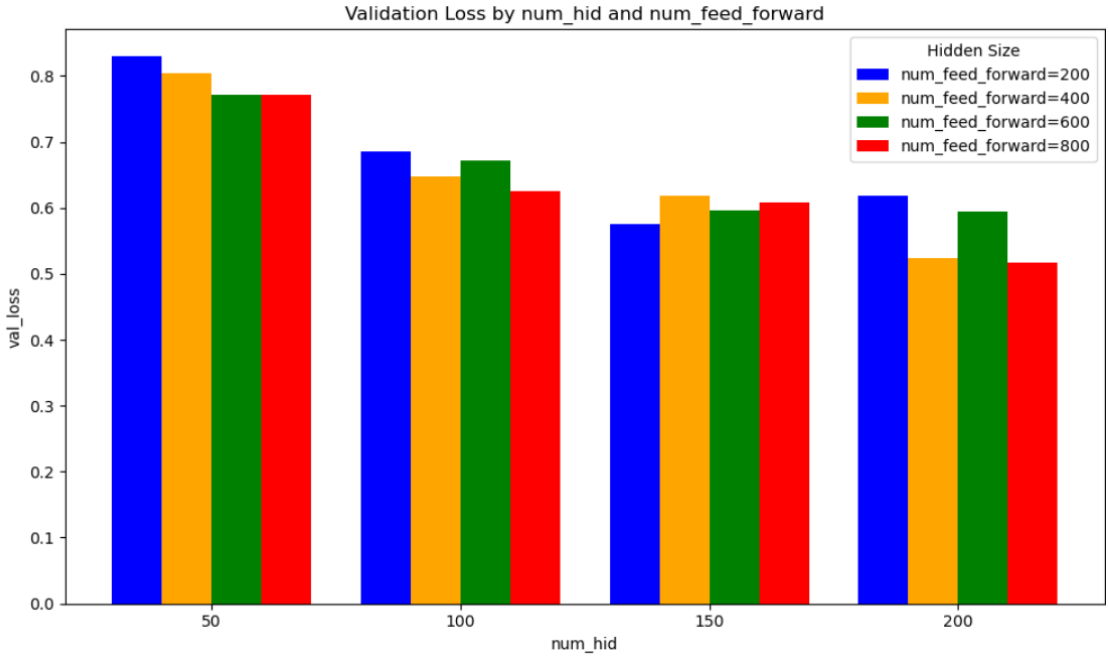
Genel olarak incelendiğinde, num_hid değerinin 50'den 150'ye yükselmesiyle birlikte, tüm num_feed_forward değerlerinde kayıp fonksiyonu değerlerinde belirgin bir azalma gözlemlenmiştir. Ancak, num_hid 200 seviyesine ulaştığında, özellikle

num_feed_forward değerleri 200 ve 400 olan durumlarda, kayıp fonksiyonunda yeniden bir artış meydana gelmiştir. Bu durum, aşırı genişletilmiş feed-forward katmanlarının modelin eğitim sürecinde kararsızlıklara yol açabileceğini göstermektedir.

İleri beslemeli ağ boyutunun (num_feed_forward) etkisi değerlendirildiğinde; düşük num_hid değerlerinde (50 ve 100) num_feeed_forward arttıkça kayıp fonksiyonu kademeli olarak azalmaktadır. Bununla birlikte, daha yüksek num_hid değerlerinde (örneğin 200) num_feed_forward büyüdükçe kayıp fonksiyonunda daha belirgin bir azalma eğilimi gözlenmiş ve en düşük kayıp değerleri num_feed_forward için 600 ve 800 seviyelerinde elde edilmiştir.

En iyi performans gösteren parametre kombinasyonları incelendiğinde, en düşük eğitim kaybının num_hid = 200 ve num_feed_forward = 800 ikilisinde yaklaşık 0.40 seviyesinde gerçekleştiği tespit edilmiştir. Benzer şekilde, num_hid = 150 ve num_feed_forward = 800 kombinasyonu da oldukça düşük bir kayıp değeri sunmaktadır.

Sonuç olarak, epoch 50 itibarıyla, modelin eğitim performansının artırılmasında geniş num_hid katmanları ile yüksek ileri beslemeli ağ katman boyutlarının (özellikle num_feed_forward = 600 ve 800) olumlu etkileri belirgindir. Ancak, num_hid= 200 değerinde düşük num_feed_forward seviyelerinde kayıp değerlerindeki artış, model kapasitesi ile veri uyumu arasında dengeli bir seçim yapılmasının gerekliliğine işaret etmektedir.




**Şekil 4.8** Epoch=50 için num_hid ve num_feed_forward parametrelerine göre val_loss değişimi.

Epoch 50'ye ait bu doğrulama kaybı (validation loss) grafiği, num_hid (gizli katman boyutu) ve num_feed_forward (feed-forward katman genişliği) parametrelerinin modelin doğrulama performansı üzerindeki etkilerini açık bir şekilde ortaya koymaktadır.

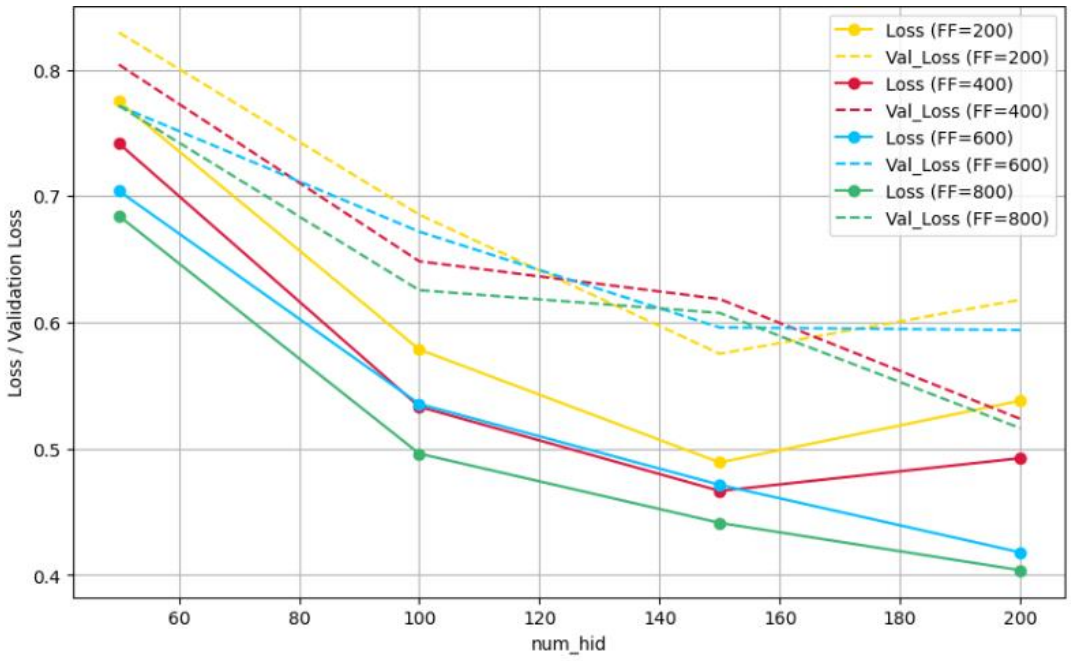
Öncelikle, tüm num_feed_forward değerleri için num_hid arttıkça doğrulama kaybında genel bir azalma trendi gözlemlenmektedir. Özellikle num_hid 50'den 150'ye yükseldiğinde, doğrulama kaybı belirgin biçimde azalmaktadır. Bu durum, daha geniş gizli katman boyutlarının modelin genelleme yeteneğini olumlu etkilediğini göstermektedir.

Num_feed_forward parametresi bağlamında ise, daha büyük ileri beslemeli ağ boyutlarının (num_feed_forward=600 ve 800) doğrulama kaybını daha düşük seviyelere çektiği dikkat çekicidir. Bu, daha yüksek kapasiteli modellerin öğrenme sürecinde daha başarılı olduğunu göstermektedir.

Ancak, num_hid 200 seviyesinde bazı durumlarda doğrulama kaybında hafif bir artış veya stabilizasyon gözlemlenmektedir. Özellikle num_feed_forward = 200 ve 400 için bu artış daha belirgindir. Bu durum, aşırı genişletilmiş num_hid katmanlarının modelin

aşırı öğrenme (overfitting) riskini artırabileceği veya eğitim dinamiklerinde kararsızlığa yol açabileceği ihtimalini desteklemektedir.

Sonuç olarak, Epoch 50'de model performansını optimize etmek için num_hid değerinin 150 civarında, num_feed_forward değerinin ise 600-800 aralığında tutulması önerilmektedir. Bu parametre kombinasyonu, doğrulama kaybının minimize edilmesi açısından en uygun dengeyi sağlamaktadır. Bu sonuçlar, model kapasitesi ile genelleme başarısı arasında dikkatli bir denge kurulmasının önemini vurgulamaktadır.




**Şekil 4.9** Epoch=50 için num_hid ve num_feed_forward parametrelerine göre loss ve val_loss değişimi.

Şekil 4.9'da, farklı beslemeli ileri ağ boyutları (num_feed_forward) ile gizli katman boyutları (num_hid) kombinasyonlarının, 50 epoch sonundaki eğitim ve doğrulama kayıplarına etkisini göstermektedir. Elde edilen sonuçlar, feed-forward boyutunun artışıyla birlikte genel olarak hem eğitim hem de doğrulama kayıplarında azalma eğilimi olduğunu ortaya koymaktadır. Bu durum, model kapasitesindeki artışın öğrenme performansını olumlu yönde etkilediğini desteklemektedir.

Özellikle daha yüksek ileri beslemeli ağ katman boyutlarına (örneğin, num_feed_forward = 600 ve 800) sahip yapıların, büyük num_hid değerleriyle birlikte daha düşük kayıplara ulaştığı gözlemlenmektedir. Bu durum, karmaşık veri temsillerinin daha geniş parametre alanlarında daha etkin öğrenilebildiğini göstermektedir.

Bununla birlikte, bazı konfigürasyonlarda (örneğin, num_hid = 200 için FF > 160) doğrulama kaybının sabitlenmesi ya da artış eğilimi göstermesi, aşırı öğrenme riskine işaret etmektedir. Bu gözlem, model yapılandırmalarında yalnızca parametre boyutlarının artırılmasının yeterli olmadığını; hiperparametreler arasındaki denge ve etkileşimin modelin genelleme kapasitesi açısından kritik rol oynadığını göstermektedir.

Dolayısıyla, model performansının optimize edilmesi sürecinde, parametre boyutlarının artırılmasının yanında, hesaplama maliyeti ve aşırı öğrenme riski gibi faktörlerin birlikte değerlendirilmesi gerekmektedir.

**Çizelge 4.4** Farklı num_hid ve num_feed_forward değerleri için epoch=100 altında elde edilen loss ve val_loss sonuçları.

| num_hid \ num_feed_forward | 50 | 100 | 150 | 200 |
|---|---|---|---|---|
| 200 | loss=0.7027 val_loss=0.7933 | loss=0.4863 val_loss=0.6774 | loss=0.4002 val_loss=0.6771 | loss=0.3556 val_loss=0.6663 |
| 400 | loss=0.6534 val_loss=0.7623 | loss=0.4413 val_loss=0.6674 | loss=0.3604 val_loss=0.6101 | loss=0.3479 val_loss=0.5410 |
| 600 | loss=0.6061 val_loss=0.7342 | loss=0.4024 val_loss=0.6634 | loss=0.3531 val_loss=0.6591 | loss=0.3443 val_loss=0.5531 |
| 800 | loss=0.5819 val_loss=0.7373 | loss=0.3899 val_loss=0.6573 | loss= 0.3505 val_loss=0.6646 | loss=0.3383 val_loss=0.6429 |

Çizelge 4.4'te 100 epoch sonunda farklı gizli katman boyutları (num_hid) ve beslemeli ileri ağ boyutları (num_feed_forward) kombinasyonlarıyla elde edilen eğitim ve doğrulama kayıplarını (loss / val_loss) sunmaktadır. Genel eğilim incelendiğinde, her iki hiperparametrenin artırılmasının eğitim kayıplarını anlamlı şekilde azalttığı; doğrulama kayıplarında ise daha sınırlı ancak istikrarlı bir iyileşme sağladığı görülmektedir.

Özellikle, num_feed_forward değerinin 200'den 800'e çıkarılması durumunda, tüm num_hid seviyelerinde eğitim kayıplarında belirgin düşüşler elde edilmiştir. Örneğin, num_hid=150 için loss değeri num_feed_forward=200'de 0.4002 iken, num_feed_forward=800'de 0.3505'e düşmüştür. Ancak doğrulama kaybı aynı aralıkta (0.6771 → 0.6646) yalnızca sınırlı bir iyileşme göstermiştir.

Benzer şekilde, sabit num_hid değerleri için feed-forward boyutunun artışı da eğitim kaybını düşürmektedir. Ancak doğrulama kayıpları bazı konfigürasyonlarda yatay seyretmekte veya marjinal olarak iyileşmektedir. Örneğin, num_feed_forward=600 için val_loss değeri num_hid=100 (0.6634) ile num_hid=150 (0.6591) arasında neredeyse sabit kalmıştır. Bu durum, modelin artan kapasitesinin eğitim verisine daha iyi uyum sağladığını ancak genelleme performansının daha yavaş ilerlediğini göstermektedir.

Genel olarak, 100 epoch'luk eğitim süresi sonunda modelin kapasite artışlarıyla eğitim başarımında belirgin ilerlemeler elde edilmiştir. Ancak doğrulama kayıplarında gözlemlenen yataylaşma ve sınırlı iyileşmeler, aşırı öğrenme riskine ve azalan getiri etkisine işaret etmektedir. Bu bağlamda, hiperparametre ayarlarının yalnızca büyüklük açısından değil, aynı zamanda etkileşimsel dengeleri gözeterek optimize edilmesi gerektiği sonucuna varılabilir.


**Şekil 4.10** Epoch=100 için num_hid ve num_feed_forward parametrelerine göre loss değişimi.

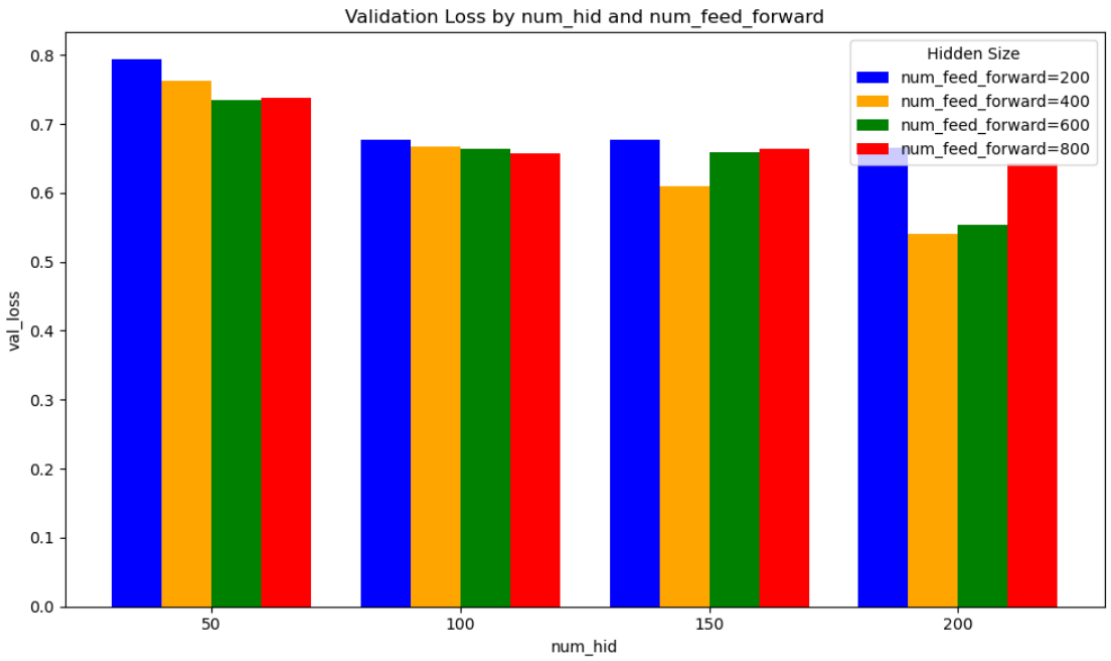
Şekil 4.10'da farklı gizli katman boyutları (num_hid) ve beslemeli ileri ağ boyutları (num_feed_forward) kombinasyonlarının, modelin eğitim kaybı (loss) üzerindeki etkisini çubuk grafik formatında göstermektedir. Elde edilen bulgular, her iki hiperparametrenin artırılmasının eğitim başarımını anlamlı biçimde iyileştirdiğini ortaya koymaktadır.

Öncelikle, num_feed_forward değerinin artışıyla birlikte tüm num_hid düzeylerinde eğitim kaybının düzenli şekilde azaldığı gözlemlenmiştir. Bu durum, model kapasitesindeki artışın öğrenme sürecine pozitif katkı sağladığını ve daha iyi örüntü yakalama yeteneği kazandırdığını göstermektedir.

Aynı şekilde, sabit feed-forward boyutlarında num_hid değerinin yükseltilmesi de eğitim kayıplarını düşürmektedir. Bu eğilim, daha geniş gizli katmanların temsil gücünü artırarak öğrenme performansını desteklediğini göstermektedir. Örneğin, num_hid=100 için num_feed_forward=200'de loss değeri yaklaşık 0.4863 iken, num_feed_forward=800'de bu değer 0.3899'a gerilemektedir.

Ancak, num_hid=200 gibi daha yüksek parametre seviyelerinde, farklı num_feed_forward değerleri arasında elde edilen kayıpların birbirine yakınsadığı ve eğrinin yataylaştığı gözlemlenmektedir. Bu durum, modelin kapasite artışından elde edilen kazançların azaldığını ve performansın doygunluk noktasına yaklaştığını işaret etmektedir.

Sonuç olarak, grafiksel analiz, model kapasitesinin artırılmasının eğitim performansını olumlu etkilediğini göstermektedir. Ancak bu artışların, belirli bir noktadan sonra marjinal fayda sağladığı ve hiperparametrelerin yalnızca boyutsal değil, aynı zamanda etkileşimsel ve verimlilik açısından da optimize edilmesi gerektiği sonucuna ulaşılmaktadır.



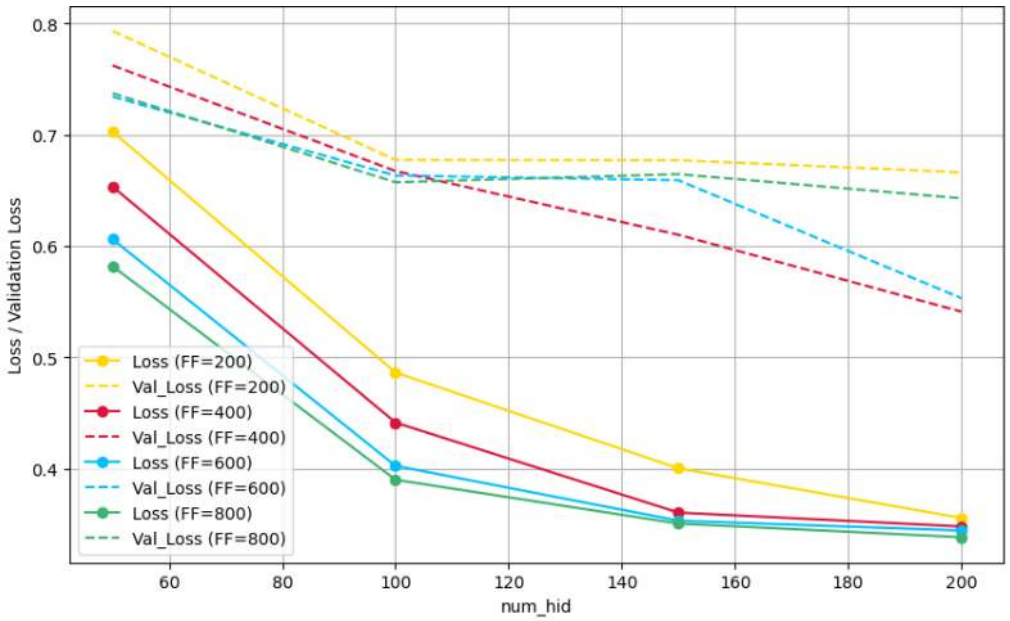

**Şekil 4.11** Epoch=100 için num_hid ve num_feed_forward parametrelerine göre val_loss değişimi.

Şekil 4.11'de modelin genelleme performansını değerlendirmek amacıyla doğrulama kaybı (val_loss) üzerinden yapılan analizleri sunmaktadır. Genel olarak, hem num_feed_forward hem de num_hid parametrelerinin artırılması doğrulama kaybını azaltmakta, ancak eğitim kaybına kıyasla daha sınırlı ve düzensiz bir iyileşme göstermektedir.

Num_hiddeğeri arttıkça,  farklı num_feed_forward düzeylerinde val_loss değerlerinde belirgin bir azalma eğilimi görülmektedir. Özellikle 50 → 100 artışıyla kayıplarda dikkate değer düşüşler yaşanırken, 150 → 200 arasında bazı durumlarda düşüş hızının azaldığı, hatta sabitleştiği gözlemlenmektedir. Bu durum, aşırı parametre artışının genelleme üzerinde sınırlı fayda sağladığını ve modelin öğrenme sınırına yaklaşmakta olduğunu göstermektedir.

Benzer şekilde, sabit num_hid değerlerinde num_feed_forward boyutu arttıkça doğrulama kayıpları genellikle azalmaktadır. Ancak bu azalma, eğitim kaybına kıyasla daha düzensizdir. Örneğin, num_hid = 150 için num_feed_forward = 200 ile 800 arasında doğrulama kaybı sadece küçük farklarla (yaklaşık 0.6771 → 0.6646) azalmaktadır. Bu gözlem, modelin eğitim verisine olan uyumu artsa da doğrulama verisinde benzer bir artışın her zaman sağlanamadığını ve aşırı öğrenme riskinin ortaya çıkabileceğini göstermektedir.

Sonuç olarak, grafik doğrulama performansında hiperparametre artışının genel olarak faydalı olduğunu ortaya koysa da, bu etkinin sınırlı ve doygunluğa yakın bir şekilde gerçekleştiği gözlemlenmektedir. Bu durum, modelin yapılandırılmasında sadece parametre büyüklüğüne değil, aynı zamanda genelleme başarımı ve hesaplama maliyeti arasındaki dengeye dikkat edilmesi gerektiğini vurgulamaktadır.




**Şekil 4.12** Epoch=100 için num_hid ve num_feed_forward parametrelerine göre loss ve val_loss değişimi

Şekil 4.12'de, num_feed_forward değerinin artışıyla birlikte hem loss hem de val_loss değerlerinde belirgin bir azalma eğilimi gözlemlenmektedir. Bu durum, modelin kapasitesinin artırılmasının hem öğrenme başarımını hem de genelleme yeteneğini iyileştirdiğini göstermektedir.

Özellikle num_feed_forward=800 için elde edilen sonuçlar, hem loss hem de val_loss bakımından en düşük değerlerin elde edildiği yapılandırma olarak öne çıkmaktadır. Bu yapılandırmada num_hid=200 seviyesinde doğrulama kaybının en düşük (yaklaşık 0.6429) seviyeye ulaştığı görülmektedir.

Ancak num_feed_forward = 200 ve 400 yapılandırmalarında, num_hid=150'den sonra val_loss değerlerinde iyileşme hızının azaldığı ve bazı durumlarda yatay bir seyir izlediği gözlemlenmiştir. Bu durum, modelin belirli bir kapasiteye ulaştıktan sonra ek parametre artışlarının genelleme başarımına sınırlı katkı sağladığını, hatta aşırı öğrenme riskini artırabileceğini düşündürmektedir.

Sonuç olarak, grafik; model kapasitesinin dikkatli bir şekilde ölçeklendirilmesinin hem öğrenme hem de genelleme performansı açısından kritik olduğunu ortaya koymaktadır.

Hiperparametre optimizasyonunda, sadece tekil parametrelerin büyüklüğü değil, bunların etkileşimli kombinasyonları da göz önünde bulundurulmalıdır.

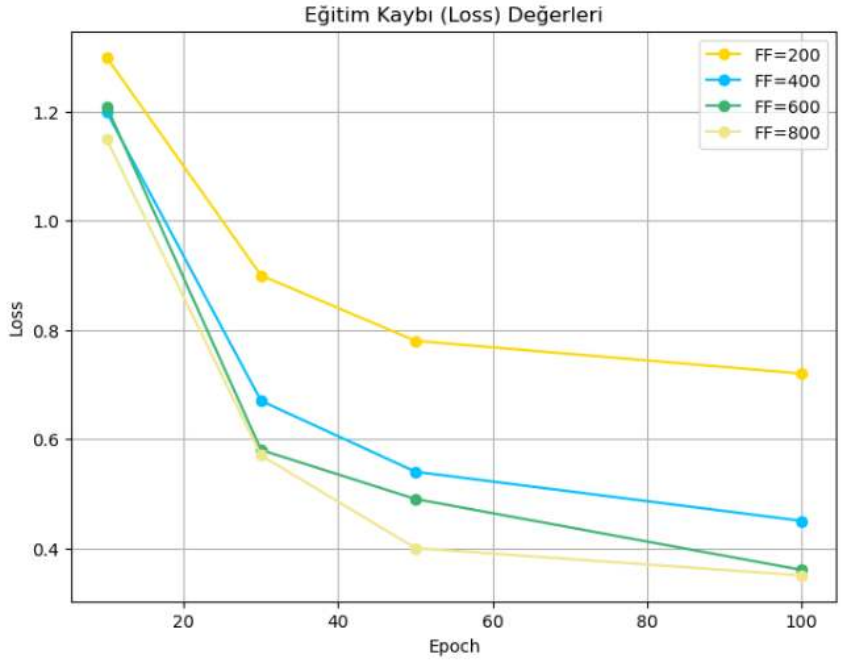
**Çizelge 4.5** Farklı num_hid / num_feed_forward kombinasyonları için loss / val_loss değerleri (epoch 10–100)

| Epoch | Num_FF | Num_Hid | Loss | Val_Loss |
|---|---|---|---|---|
| 10 | 200 | 50 | 1.2937 | 1.3718 |
| 10 | 400 | 100 | 1.2661 | 1.2907 |
| 10 | 600 | 150 | 1.2099 | 1.1976 |
| 10 | 800 | 200 | 1.1633 | 1.1518 |
| 30 | 200 | 50 | 0.8829 | 0.9098 |
| 30 | 400 | 100 | 0.6481 | 0.7284 |
| 30 | 600 | 150 | 0.5821 | 0.6333 |
| 30 | 800 | 200 | 0.6173 | 0.6043 |
| 50 | 200 | 50 | 0.7747 | 0.8293 |
| 50 | 400 | 100 | 0.5332 | 0.6483 |
| 50 | 600 | 150 | 0.4714 | 0.5959 |
| 50 | 800 | 200 | 0.4039 | 0.5163 |
| 100 | 200 | 50 | 0.7027 | 0.7933 |
| 100 | 400 | 100 | 0.4413 | 0.6674 |
| 100 | 600 | 150 | 0.3531 | 0.6591 |
| 100 | 800 | 200 | 0.3383 | 0.6429 |

Çizelge 4.5'te, seçilen altı farklı num_hid ve num_feed_forward parametre kombinasyonu için epoch = 10, 30, 50 ve 100 değerlerinde elde edilen eğitim (loss) ve doğrulama (val_loss) kayıpları sunulmaktadır. Elde edilen sonuçlar, epoch sayısının artışıyla hem eğitim hem doğrulama kayıplarında genel bir azalma eğilimi olduğunu ortaya koymaktadır.

Özellikle num_hid ve feed_forward değerleri yüksek olan kombinasyonlarda (örneğin 800/200), hem loss hem de val_loss değerlerinde daha düşük sonuçlar elde edilmiştir. Bu durum, model kapasitesinin artmasının öğrenme başarımına olumlu katkı sağladığını göstermektedir. Ancak bazı durumlarda (örneğin epoch 100'de 200/200 için val_loss artışı), aşırı öğrenme (overfitting) riski de gözlemlenebilir.

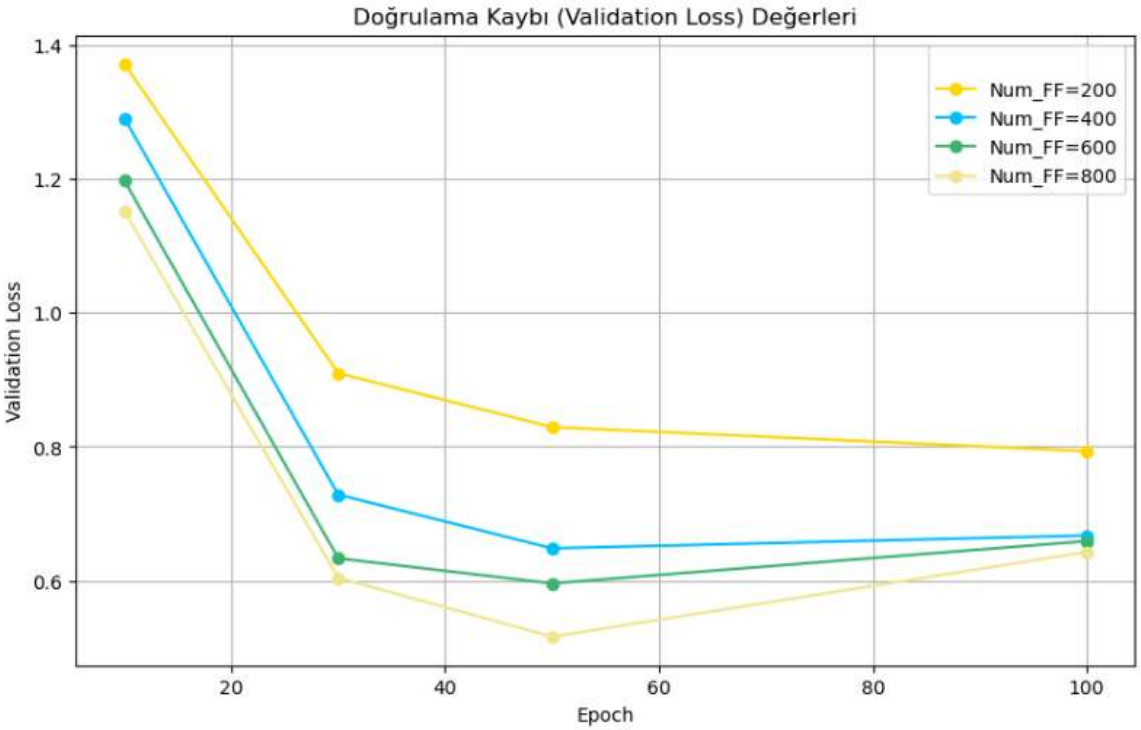
Sonuçlar, parametre seçiminin model başarımı üzerindeki kritik rolünü ortaya koymakta; eğitim süresiyle birlikte dengeli bir yapılandırmanın gerekliliğini vurgulamaktadır.




**Şekil 4.13** Farklı num_hid / num_feed_forward kombinasyonları için loss değişimi (epoch 10–100)

Şekil 4.13'te farklı num_hid ve num_feed_forward kombinasyonları için epoch sayısına bağlı olarak ölçülen eğitim kaybı (loss) değerlerinin değişimi gösterilmektedir. Grafik incelendiğinde, tüm kombinasyonlarda epoch sayısının artmasıyla birlikte loss değerlerinde belirgin bir azalma gözlemlenmektedir.Özellikle num_feed_forward=800 ve num_hid=200 parametreleriyle elde edilen eğri, en düşük loss değerine ulaşarak modelin bu yapılandırmada daha verimli bir öğrenme gerçekleştirdiğini göstermektedir.Daha düşük parametre kombinasyonlarında (örneğin num_feed_forward=200, num_hid=50) ise kayıpların daha yüksek ve azalma hızının daha düşük olduğu görülmektedir. Bu durum, modelin bu konfigürasyonlarda karmaşık örüntüleri öğrenme kapasitesinin sınırlı olduğunu göstermektedir.

Sonuç olarak, eğitim kaybı eğrileri, modelin epoch sayısı arttıkça daha iyi genelleme yapacak biçimde optimize edildiğini ortaya koymaktadır.




**Şekil 4.14** Farklı num_hid / num_feed_forward kombinasyonları için val_loss değişimi (epoch 10–100)

Şekil 4.14'te epoch sayısına göre elde edilen doğrulama kaybı (val_loss) değerleri gösterilmektedir. Genel olarak, epoch sayısının artışıyla doğrulama kayıplarında da azalma eğilimi görülmektedir.

En düşük val_loss değeri, yine num_feed_forward=800 ve num_hid=200 kombinasyonunda elde edilmiştir. Bu durum, modelin bu parametrelerle yalnızca eğitim verisini ezberlemekle kalmayıp, yeni verilere de iyi genelleme yapabildiğini göstermektedir.

Bazı düşük parametre kombinasyonlarında (örneğin num_hid=200), epoch arttıkça val_loss değerlerinde zaman zaman dalgalanmalar veya hafif artışlar görülmüştür. Bu, modelin aşırı öğrenme (overfitting) eğilimi gösterebileceğini düşündürmektedir.
Dolayısıyla, doğrulama kaybı grafiği, yüksek num_hid ve num_feed_forward değerlerinin modelin genel başarımını artırdığını ve aşırı öğrenmeden kaçınmasını sağladığını göstermektedir.

# 5. TARTIŞMA ve SONUÇ

Bu tez çalışmasında, derin öğrenme temelli bir modelde gizli katman boyutu (num_hid) ve beslemeli ileri ağ boyutu (num_feed_forward) hiperparametrelerinin modelin eğitim başarımı ve genelleme yeteneği üzerindeki etkileri deneysel olarak analiz edilmiştir. 30, 50 ve 100 epoch sonunda elde edilen eğitim (loss) ve doğrulama (val_loss) kayıpları temel alınarak yapılan değerlendirmelerde, hiperparametre artışının modelin öğrenme sürecine olan katkısı nicel olarak ortaya konmuştur.

Sonuçlar, her iki hiperparametredeki artışın modelin hem eğitim hem de doğrulama kaybında genel bir azalma sağladığını göstermektedir. Bu durum, ağ mimarisinin kapasitesinin artırılmasının, modelin veri içindeki örüntüleri daha etkili bir biçimde öğrenmesine olanak sağladığını ortaya koymaktadır. Özellikle 100 epoch sonunda elde edilen verilerde, num_feed_forward=800 ve num_hid=200 yapılandırması, en düşük eğitim ve doğrulama kayıplarına ulaşarak en başarılı yapılandırma olarak öne çıkmıştır.Bu durum, azalan marjinal fayda ilkesi ile açıklanabilir. Model kapasitesi belirli bir noktadan sonra, verideki anlamlı örüntüler dışında kalan ayrıntılara (gürültü, rastlantısallık) da uyum sağlamaya başlamakta ve bu da genelleme performansının sınırlanmasına neden olmaktadır. Diğer bir ifadeyle, öğrenme kapasitesindeki artış, bir noktadan sonra aşırı öğrenmeye (overfitting) yol açarak doğrulama performansını olumsuz etkileyebilmektedir.

## 5.1 Eğitim ve Doğrulama Kayıpları Arasındaki Fark

Grafiklerde dikkat çeken bir diğer bulgu, modelin eğitim kaybı ile doğrulama kaybı arasındaki farkın, yüksek parametreli yapılandırmalarda daha da belirginleştiğidir. Özellikle num_feed_forward=800 ve num_hid=200 gibi yapılandırmalarda, eğitim kaybı oldukça düşük (örneğin 0.3383) seviyelere gerilerken, doğrulama kaybı görece daha yüksek (örneğin 0.6429) kalmıştır. Bu farklılık, modelin eğitim verisine fazlaca uyum sağladığını; ancak bu başarının doğrulama verisine tam olarak yansımadığını göstermektedir.

## 5.2 Model Mimarisi ve Performans-Genelleme Dengesi

Bununla birlikte, parametre artışı her zaman doğrusal bir performans artışı sağlamamış, belirli yapılandırmalarda azalan marjinal fayda gözlemlenmiştir. Özellikle num_hid parametresinin 150'den 200'e çıkarılmasıyla bazı yapılandırmalarda val_loss değerlerinde gözle görülür bir iyileşme sağlanamamıştır. Bu durum, hiperparametrelerin bilinçsiz biçimde artırılmasının, aşırı öğrenme (overfitting) riskini beraberinde getirdiğini ve modelin yalnızca eğitim verisine odaklanarak genelleme yeteneğini kaybedebileceğini göstermektedir. Nitekim yüksek parametrelerde loss ile val_loss arasındaki farkın açılması da bu riski destekler niteliktedir.

Grafiksel analizler, düşük parametre yapılandırmalarında modelin yetersiz temsil gücüne sahip olduğunu, bu nedenle yüksek eğitim ve doğrulama kayıplarına maruz kaldığını ortaya koymaktadır. Öte yandan, yüksek parametre yapılandırmalarında elde edilen daha düşük kayıp değerleri, modelin temsil kapasitesinin arttığını göstermekte, ancak bu kapasite artışının genelleme üzerinde sınırlı katkı sağladığı da göz ardı edilmemelidir. Bu bulgular, literatürde sıkça vurgulanan "model karmaşıklığı ile genelleme arasında optimal bir denge kurulması gerektiği" görüşünü desteklemektedir.

Bulgular, model mimarisinde yapılan hiperparametre ayarlarının yalnızca doğruluk ve kayıp metriklerini değil, aynı zamanda modelin genelleme yeteneğini, hesaplama verimliliğini ve aşırı öğrenme riskini de doğrudan etkilediğini ortaya koymaktadır. Bu nedenle model yapılandırmalarında sadece yüksek kapasite arayışına değil, *performans–verimlilik–genelleme dengesi*ne de odaklanmak gereklidir.

Bu bağlamda, en uygun parametre kombinasyonunun sadece en düşük loss ya da val_loss değeriyle değil, aynı zamanda parametre sayısının getirdiği hesaplama maliyeti, eğitim süresi, ve overfitting riski gibi faktörlerle birlikte değerlendirilmesi gerektiği açıktır.

## 5.3 Sonuç ve Gelecek Çalışmalar

Bu tez çalışması, yapay sinir ağlarının mimari yapılandırmasının model başarımını doğrudan etkilediğini göstermiştir. Hiperparametrelerin dengeli ve etkileşimli biçimde ayarlanması, hem eğitim hem de doğrulama başarısını optimize etmek için kritik önemdedir.

Gelecek çalışmalarda:

- Bayesian optimizasyon, grid search veya random search gibi sistematik hiperparametre arama yöntemlerinin kullanılması,
- Erken durdurma (early stopping), dropout, L2 regularizasyonu gibi aşırı öğrenmeyi engelleyici tekniklerin entegrasyonu,

Ayrıca, modelin öğrenme başarımının yalnızca epoch sayısıyla değil, aynı zamanda ağ derinliği ve genişliği ile de doğrudan ilişkili olduğu görülmüştür. Epoch sayısının artırılması, öğrenmenin istikrar kazanmasını sağlamakta; ancak parametre yapılandırması yetersizse, bu artış performansa doğrudan yansımamaktadır. Bu durum, yalnızca eğitim süresini uzatmanın yeterli olmadığını, mimari yapılandırmaların da bu sürece uygun optimize edilmesi gerektiğini göstermektedir.

Bu bulgular ışığında, hiperparametre seçiminin yalnızca deneme-yanılma yöntemiyle değil, sistematik bir optimizasyon süreci ile ele alınması gerektiği sonucuna varılmıştır. Bu bağlamda, gelecekteki çalışmalarda grid search, random search, Bayesian optimizasyon gibi algoritmik yaklaşımlar kullanılarak daha verimli yapılandırmaların elde edilmesi hedeflenebilir. Ayrıca erken durdurma (early stopping), dropout, L2 regularizasyonu gibi aşırı öğrenmeyi önleyici tekniklerle modelin genelleme başarımının daha kararlı hâle getirilmesi önerilmektedir.

- Farklı veri kümeleri ve görevler (sınıflandırma, regresyon vb.) üzerinde yapılandırmaların tekrar test edilmesi,
- Hiperparametrelerin otomatik seçimini sağlayan meta-öğrenme yaklaşımlarının modellenmesi gibi yöntemler önerilmektedir.

Sonuç olarak, bu tez kapsamında yapılan deneysel çalışmalar, derin öğrenme modellerinin başarımında hiperparametre ayarlarının belirleyici rol oynadığını açık biçimde ortaya koymuştur. Eğitim süresi, model mimarisi ve parametrik yapılandırmaların birlikte değerlendirilmesi, sadece modelin nicel performansını değil, aynı zamanda genelleme başarısını da doğrudan etkilemektedir. Bu bağlamda, parametre optimizasyonunun yalnızca teorik yaklaşımlarla değil, uygulamaya dönük ve çok boyutlu bir değerlendirme süreciyle ele alınması gerektiği açıktır.

Modelin performansı ile hesaplama maliyeti, eğitim süresi ve aşırı öğrenme riski arasında sağlanacak optimal denge, hem akademik araştırmalarda hem de gerçek dünya uygulamalarında daha güvenilir, kararlı ve ölçeklenebilir derin öğrenme modellerinin geliştirilmesine katkı sunacaktır. Bu yönüyle çalışma, hiperparametre seçiminde deneme-yanılma yaklaşımından öte, sistematik ve veri temelli bir optimizasyon anlayışının gerekliliğini vurgulamaktadır.

## 6. KAYNAKLAR


Abdel-Hamid O, Mohamed A R, Jiang H, Deng L, Penn G, Yu D, 2014,Convolutional neural networks for speech recognition, IEEE/ACM Transactions on Audio, Speech, and Language Processing, 22(10).

Brown T B, Mann B, Ryder N, Subbiah M, Kaplan J, Dhariwal P vd. 2020, Language models are few-shot learners, Advances in Neural Information Processing Systems, 33, 1877–1901.

Cho K, Van Merriënboer B, Bahdanau D, Bengio Y, 2014, Learning phrase representations using RNN encoder-decoder for statistical machine translation, In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP).

Crevier D, 1993, AI: The tumultuous history of the search for artificial intelligence, New York, NY: Basic Books.

Davis S, Biddulph R, Balashek S, 1952, Automatic recognition of spoken digits, The Journal of the Acoustical Society of America, 24(6), 637–642.

Deng L, Yu D, 2014, Deep learning: Methods and applications, Foundations and Trends in Signal Processing, 7(3–4), 197–387.

Devlin J, Chang M.-W, Lee K, Toutanova K, 2019, BERT: Pre-training of deep bidirectional transformers for language understanding, Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 1, 4171–4186.

Dong L, Xu S, Xu B, 2018, Speech-Transformer: A no-recurrence sequence-to-sequence model for speech recognition, In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).

Elman J L, 1990, Finding structure in time, Cognitive Science, 14(2), 179–211.

Gehring J, Auli M, Grangier D, Yarats D, Dauphin Y N, 2017, Convolutional sequence to sequence learning, Proceedings of the 34th International Conference on Machine Learning (ICML).

Goodfellow I, Bengio Y, Courville A, 2016, Deep learning, MIT Press.

Graves A, Mohamed A R, Hinton G, 2013, Speech recognition with deep recurrent neural networks, In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).

Gulati A, Qin J, Chiu C C, Parmar N, Zhang Y, Yu J, Han W, Wang S, Zhang Z, Wu Y, Pang R, 2020, Conformer: Convolution-augmented transformer for speech recognition, In Proceedings of Interspeech 2020.

Hinton G, Deng L, Yu D, Dahl G E, Mohamed A R, Jaitly N, Kingsbury B, 2012, Deep neural networks for acoustic modeling in speech recognition, IEEE Signal Processing Magazine, 29(6), 82–97.

Hinton G, Osindero S, Teh Y W, 2006, A fast learning algorithm for deep belief nets, Neural Computation, 18(7), 1527–1554.

Hochreiter S, Schmidhuber J, 1997, Long short-term memory, Neural Computation, 9(8), 1735–1780.

Ioffe S, Szegedy C, 2015, Batch normalization. Proceedings of the 32nd International Conference on Machine Learning (ICML).

Itakura F, 1975, Line spectrum representation of linear predictive coefficients of speech signals, The Journal of the Acoustical Society of America, 57(4).

Ito K, Johnson L, 2017, The LJ Speech Dataset, University of Southern California.

Jurafsky D, Martin J. H, 2023, Speech and language processing (4th ed.), Pearson.

Karita S, Chen N, Hayashi T, Hori T, Watanabe S, Matsuo Y, 2019, Comparative study of transformer vs RNN in speech recognition. In Proceedings of Interspeech 2019.

Koehn P, 2010, Statistical machine translation. Cambridge University Press.

Krizhevsky A, Sutskever I, Hinton G. E. 2012, ImageNet classification with deep convolutional neural networks, In Advances in Neural Information Processing Systems (pp. 1097–1105).

LeCun Y, Bengio Y, Hinton G, 2015, Deep learning, Nature, 521(7553), 436–444.

Lin Z, Feng M, Santos C. N. dos, Yu M, Xiang B, Zhou B, Bengio Y, 2017, A structured self-attentive sentence embedding, In International Conference on Learning Representations (ICLR).

Manning C. D, Schütze H, 1999, Foundations of statistical natural language processing, MIT Press.

McCarthy J, Minsky M, Rochester N, Shannon C, 1955, A proposal for the Dartmouth Summer Research Project on Artificial Intelligence, Dartmouth College.

Oyucu S, 2020, Türkçe konuşma tanıma sistemleri için derin öğrenme tabanlı modellerin geliştirilmesi (Yüksek lisans tezi), Gazi Üniversitesi, Fen Bilimleri Enstitüsü, Ankara.

Panayotov V, Chen G, Povey D, Khudanpur S, 2015, LibriSpeech: An ASR corpus based on public domain audio books, In ICASSP.

Rabiner L, 1989, A tutorial on hidden Markov models and selected applications in speech recognition, Proceedings of the IEEE, 77(2), 257–286.

Rabiner L, Schafer R, 2011, Digital processing of speech signals.

Radford A, et al, 2019, Language models are unsupervised multitask learners, OpenAI.

Raffel C, Shazeer N, Roberts A, Lee K, Narang S, Matena M, Zhou Y, Li W, Liu P J, 2020, Exploring the limits of transfer learning with a unified text-to-text transformer, Journal of Machine Learning Research, 21(140).

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

Russell S, Norvig P, 2021, Artificial intelligence: A modern approach (4th ed.), Pearson Education.

Sainath T N, Vinyals O, Senior A, Sak H, 2015, Convolutional, long short-term memory, fully connected deep neural networks, In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).

Schmidhuber J, 2015, Deep learning in neural networks: An overview, Neural Networks, 61, 85–117.

Shaw P, Uszkoreit J, Vaswani A, 2018, Self-attention with relative position representations, In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT) .

Smith J, 2018, Spectrogram analysis in speech recognition.

Sutskever I, Vinyals O, Le Q V, 2014, Sequence to sequence learning with neural networks, In Advances in Neural Information Processing Systems (NeurIPS).

Turing A M, 1950, Computing machinery and intelligence, Mind, 59(236), 433–460.

Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez A N, Kaiser Ł, Polosukhin I, 2017, Attention is all you need. In Advances in Neural Information Processing Systems (Vol. 30, pp. 5998–6008).

Williams R J, Zipser D, 1989, A learning algorithm for continually running fully recurrent networks, Neural Computation, 1(2).

Wolf T vd. 2020, Transformers: State-of-the-art natural language processing, In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations.

Xu S, Li Y, Zhang H, Wang J, 2020, Transformer-based speech recognition on LJSpeech dataset, IEEE Access.

Young T, Hazarika D, Poria S, Cambria E, 2018, Recent trends in deep learning based natural language processing, IEEE Computational Intelligence Magazine, 13(3), 55–75.

Zhang Y, Liu X, Wang H, Chen J, 2020, Hyperparameter tuning in transformer-based ASR systems, In Proceedings of ICASSP 2020.

**İnternet Kaynakları**

1- Nandan, A. (2021, January 13). Automatic speech recognition with transformer. Keras. https://keras.io/examples/audio/transformer_asr/, 05,01,2026

# EKLER

**EK 1.** Uygulama Kod Bloğu (Nandan, 2021)

```
## Create & train the end-to-end model
"""

batch = next(iter(ds))

# The vocabulary to convert predicted indices into characters
idx_to_char = vectorizer.get_vocabulary()
display_cb = DisplayOutputs(
    batch, idx_to_char, target_start_token_idx=2, target_end_token_idx=3
)  # set the arguments as per vocabulary index for '<' and '>'

#batch = next(iter(val_ds))
ds = create_tf_dataset(train_data, bs=10)  # Batch boyutunu 2 yapın
val_ds = create_tf_dataset(test_data, bs=1)
## The vocabulary to convert predicted indices into characters
#idx_to_char = vectorizer.get_vocabulary()
#display_cb = DisplayOutputs(
#    batch, idx_to_char, target_start_token_idx=2, target_end_token_idx=3
#)  # set the arguments as per vocabulary index for '<' and '>'

model = Transformer(
    num_hid=200,
    num_head=2,
    num_feed_forward=50,
    target_maxlen=max_target_len,
    num_layers_enc=4,
    num_layers_dec=1,
    num_classes=34,
)
loss_fn = keras.losses.CategoricalCrossentropy(
    from_logits=True,
    label_smoothing=0.1,
)
```

**EK 2. (Devam)** Uygulama Kod Bloğu (Nandan, 2021)

```
learning_rate = CustomSchedule(
    init_lr=0.00001,
    lr_after_warmup=0.001,
    final_lr=0.00001,
    warmup_epochs=15,
    decay_epochs=85,
    steps_per_epoch=len(ds),
)

optimizer = keras.optimizers.Adam(learning_rate)
model.compile(optimizer=optimizer, loss=loss_fn)
from tensorflow.keras.callbacks import EarlyStopping

early_stopping = EarlyStopping(
    monitor='val_loss',
    patience=200,  # Sabırlı kalma süresi
    restore_best_weights=True
)

history = model.fit(ds, validation_data=val_ds, callbacks=[display_cb, early_stopping], epochs=10)
```