İSTANBUL TEKNİK ÜNİVERSİTESİ
ELEKTRİK- ELEKTRONİK FAKÜLTESİ

DERİNLİKLİ ÖĞRENME İLE KONUŞMACI DUYGU DURUMUNUN SINIFLANDIRILMASI

BİTİRME ÖDEVİ

Ahmet Haluk AÇARÇİÇEK

040100487

Bölümü: Elektronik ve Haberleşme Mühendisliği

Programı: Telekomünikasyon Mühendisliği

Danışmanı: Prof. Dr. Bilge GÜNSEL

MAYIS 2015

ÖNSÖZ

İstanbul Teknik Üniversitesi Elektronik ve Haberleşme Mühendisliği bitirme projesi kapsamında hazırlanan bu tez boyunca değerli zamanını ve tecrübelerini esirgemeyip motivasyonumu sürekli yüksek tutmamda yardımcı olan tez danışmanım Prof. Dr. Bilge Günsel'e, zorlu süreç boyunca kaprislerimi çeken arkadaşlarım ve aileme sonsuz teşekkürlerimi sunarım.

Mayıs 2015 Ahmet Haluk Açarçiçek

İÇİNDEKİLER

KISALTMALAR

CD : Karşıtlıklı Iraksay(Contrastive Divergence)
ÇSD : Çerçevenin Seslilik Değeri
DBN : Derinlikli İnanç Ağları(Deep Belief Networks)
GPU : Grafiksel İşlemci Ünitesi(Graphical Processing Unit)
KL : Kullback-Leibler Ayrışması
NIZD : Normalize Edilmiş İzgesel Zarf Farkı
NSL : Uykulu Değil(Non-Sleepy)
RBM : Kısıtlı Boltzmann Makinaları(Restricted Boltzmann Machines)
SGD : Stokastik Gradyan Azaltma(Stochastic Gradient Descent)
SL : Uykulu(Sleepy)

TABLO LİSTESİ

ŞEKİL LİSTESİ

DERİNLİKLİ ÖĞRENME İLE KONUŞMACI DUYGU DURUMUNUN SINIFLANDIRILMASI

ÖZET

2002'de başlayan dijital çağ ile birlikte dünya çapındaki saklanan veriler her geçen yıl katlanmaktadır. Günümüzde kişisel bilgisayarların bile terabaytlarca veri saklayabildikleri ve her geçen günde exabaytlar mertebesinde(10^{18} bayt) yeni verinin üretildiği göz önüne alınarak dijital ortamdaki tüm(ses, görüntü, yazı…) analiz edilebilir verilere büyük veri ismi verilmiştir. Büyük veri ile birlikte de daha önce kullanılan veri analizi yöntemleri çoğu anlamda yetersiz kalmıştır. Aynı zamanda büyük verinin teknoloji şirketleri ve insan hayatı açısından öneminin anlaşılması ile birlikte daha doğru, hızlı ve anlamlı analizlerin yapılması önem kazanmıştır. Ayrıca daha önce kısıtlı sayıda müşterilere, dolayısıyla veriye sahip olan şirketler analizlerini insan aklı sayesinde yapabilmekteyken; e-ticaret, teknolojik gelişmeler ve talep sebebiyle artan analiz ihtiyaçları daha "akıllı" yöntemlerin ihtiyacını doğurmuştur. Yapay zeka konusundaki gelişmeler ile birlikte duruma göre üretilen algoritmalar yerine, ihtiyaca göre, veriden çıkarım yapabilecek akıllılıkta yöntemler pratikte uygulanabilecek hızlara kavuşunca var olan ihtiyaca yönelik çözümler ortaya konmuştur. Duygu sınıflandırma problemi de bu gibi sebeplerden dolayı ortaya çıkmış psikolog veya psikiyatrların yerini alabilecek ve pratikte kullanılabilecek bir problemdir.

Duygular insan hayatında ve tercihlerinde önemli roller oynamaktadır. Beyin ve hormonların etkisiyle ortaya çıkan düşünce ve davranışların farklı örüntülerdeki hallerine farklı duygu isimleri verilmiştir. İnsan beyni tarafından yapılan bu sınıflandırmanın yapay zeka ile gerçekleştirilebilir olup olmaması duygu sınıflandırma probleminin temelini oluşturmaktadır. Duygu sınıflandırma probleminin çözümleri insan hayatında birçok noktada kendine yer bulmaktadır. Havaalanları, konsolosluk vb. yerlerde güvenlik amacı ile, insan psikolojisinin daha

iyi kavranmasında ve psikolojik rahatsızlıkların tespit edilmesinden, ticari uygulamalarda kullanıcı tepkilerinin ölçülmesine hatta duygu durumuna göre tüketilebilir içerikler önerilmesine kadar bir çok alanda uygulamaya geçmiş olan duygu sınıflandırması, görüntü, ses ve biyolojik sensör verileri yardımıyla gerçekleştirilebilir.

Günlük hayatta ise sadece karşıdaki kişinin sesinden o anki duygularına dair çıkarımlar yapabilmekteyiz. Bu motivasyonla çalışma boyunca sadece ses verisi ile duygu sınıflandırma problemi ele alınmıştır. Çalışmalarda kullanılacak veriler SLC(Sleepy Language Corpus) olarak adlandırılmakla birlikte, profesyonel kişiler tarafından uykulu(SL) ve uykulu değil(NSL) olarak ikiye ayrılmış, farklı kişilere ait konuşmalardan oluşmaktadır. Konuşmacıların uyku durumunun sınıflandırılması ise özellikle otomotiv sektöründe sürüş güvenliği açısından olduğu gibi iş güvenliği açısından da fayda sağlayabilecek bir problemdir. Tez çalışmasında kullanılacak SLC verisi derinlikli öğrenme yapıları ile modellenmeye çalışılacaktır.

2006'dan bu yana gitgide önem kazanan derinlikli öğrenme (Deep Learning), insan beyninin hiyerarşik yapısından esinlenilerek ortaya çıkarılmış bir makine öğrenmesi uygulaması olarak, çok boyutlu verilerin modellenmesinde diğer yöntemlere göre daha başarılı sonuçlar vermektedir. DBN(Derinlikli İnanç Ağları)'ler ise derinlikli ağların özel bir çeşidi olup üst üste eklenen RBM(Kısıtlanmış Boltzmann Makineleri) katmanlarının ön-eğitim ve ince-ayar adlı iki aşamada eğitilmesiyle meydana gelmektedirler. Temelleri 1940'lardan beri kullanılmakta olan yapay sinir ağlarına dayanan DBN'lerin kapasitesi ise yapısını oluşturan sanal nöronların sayısı ile artmaktadır. Özellikle bu alanda yapılan yeni çalışmalar ve artan işlemci kapasiteleri sayesinde DBN'lerin gün geçtikçe daha hızlı eğitilmesi ve daha büyük ağ yapılarının kullanılabilmesine olanak tanımaktadır. Artan kapasiteyle birlikte DBN'ler daha karmaşık yapay zeka problemlerinin çözümünde rol almaya başlamışlardır. Bu çalışmada da DBN'ler ile konuşmacıların uykulu(SL)/uykulu değil(NSL) olma durumları sınıflandırılarak, başarım sonuçları analiz edilmiştir. Başarım sonuçlarına ve teorik altyapıya göre DBN yapıları optimize edilmeye çalışılmıştır.

EMOTION RECOGNITION FROM VOICE BY USING DEEP LEARNING

SUMMARY

When the start of the digital age at 2002, data storage capacities starts to increase very rapidly every other year. Nowadays, even personal computers can store terabytes of data and every past day, Exabyte (10^{18} byte) of data are created. According to this fact, all of the data at the digital medium (sound, image, text etc.) that can be analyzed, is called big data. With the presence of big data, most of the existing data analysis methods become inadequate. Also, when the importance of the data is comprehend by the tech companies and human beings, the need of more accurate, fast and meaningful data analysis emerge. Besides of that, when arise of the things like, technological improvements, demand of quality service and e-commerce, companies start to reach more customers than before which led to more data than before. Result of that, companies depends on human labor at data analysis become obsolete and companies are head towards to more "smart" ways to analyze the data. With the improvements in artificial intelligence, classical algorithms that depends on the problem, replaces their places to data centric solutions that can make deductions from data, according to problem. This breakthrough, addresses the problem described before when the speeds of this techniques reaches the practical levels. Emotion classification is also a problem occurs because of this reasons like this which can replace or assist psychologists and psychiatrist in some situations.

Emotions take very important place in the human life and preferences. Different behavioral and mentalistic patterns with the effect of brain and hormones are called different emotions. The doability of this classification with artificial intelligence like in the human brain forms the emotion classification problem. Solution of this problem are applicable in many different areas like security, health, psychology, commerce with the voice, image and biological sensor data.

In daily life, we can make emotion deductions from people's voice. Through this thesis, this motivation leds to solve emotion classification with only voice data. In this work, SLC (Sleepy Language Corpus) data which contains speech from different people that classified as sleepy (SL) or non-sleepy (NSL) by professionals is used. Detection of the speaker's sleepy state only from voice can be helpful in the business and automotive industry. SLC database is modeled with deep learning structures in this thesis.

Deep learning models are inspired from human brain's hierarchical structure. As a machine learning method, deep learning performs state of the art results when modeling high dimensional data. And deep belief networks (DBN) is a special type of deep learning models which is consist of restricted Boltzmann Machines (RBM) stack on to each other. DBN's trained in two steps which is pretraining the RBM's and fine-tuning. DBN is fundamentally rely on the neural network structures which are used since 1940's. Modeling capacity is dependent to the number of artificial neuron units in the structure like neural networks. With the increasing processor speeds and recent studies, larger and more efficient models can be trained day by day. This rising capacity leads to solve much more complicated artificial intelligence problems with DBN, like the classification problem of speaker's sleepiness levels in this thesis. After suggestion of a solution to the problem, DBN's are optimized according to underlying theory and results are analyzed and reported like in technical literature.

1. GİRİŞ

Zeka kavramı tarih boyunca gerek filozofların gerek diğer insanların ilgisini çekmiştir. Zeka öznel bir kavram olmakla birlikte, her düşünürün de zekaya olan yaklaşımı farklı olmuştur. İnsan zekası modern anlamda IQ puanlarıyla somut bir zemine indirgenmeye çalışılarak ölçülmesi 1912 yılına kadar dayanmaktadır. Nitekim testlerde kullanılan Raven matrislerini insan beyninin çözme süresi ile zekası arasında doğru bir ilişki olduğu kabul edilmektedir. Fakat modern bilgisayarlar binlerce basamaklı sayıları, çok boyutlu matrisleri saniyeler almadan çarpıp, toplayabilmekteyken insan beyni için bu işlemlerin günler hatta aylar sürmesi insan zekasıyla, makine zekası arasında bir ayrıma yol açmıştır.

Matematikçi Alan Turing'in meşhur makalesi[1] ile ortaya koyduğu gibi yapay zeka olarak adlandırılan makine zekası ile insan zekası birbirinden farklı ve Turing testleriyle ayırt edilebilmekteydiler. Küçük bir çocuğun bile saniyeler içerisinde yüzlerce objeyi tanıyıp ayırt edebilmesi, duyduğu sesten kim olduğu hatta hislerini anlayabilmesi o yıllardaki yapay zekalar için olanaklı değillerdi. Fakat gün geçtikçe insan beyni esinlenerek geliştirilen algoritmalar ile birlikte daha "zeki" hale gelen yapay zekalar ile birlikte Turing testi de yavaş yavaş geçerliliğini kaybetmektedir.

Özellikle 2006 yılından bu yana insan beynine işleyiş açısından daha yakın olan derinlikli ağların ortaya çıkışıyla yapay zekalar, birçok karmaşık sınıflandırma, kümeleme gibi problemleri insana yakın başarım oranlarında gerçekleştirmeye başlamıştır. Derinlikli ağlar doğal dil işleme, görüntü işleme, ses işleme gibi birçok problemde kullanılmıştır.

1.1 Çalışmanın Amacı

İletişim, sosyal bir varlık olan insan hayatında büyük öneme sahiptir. Toplulukta iletişim gerçekleşirken iletilmek istenen mesaj çeşitli yollarla karşı tarafa aktarılmaktadır. İnsanlar ses, görüntü, dokunma gibi birçok yoldan biri veya bir kaçını kullanarak iletişim kurmaktadırlar. İnsan beyni iletişim sırasında karşısındaki

kişilerin ruh hallerini, konuştuklarını ve amaçlarını aynı anda analiz edebilmektedir. Bu analizlerden biri de duygu sınıflandırmasıdır. Bireylerdeki bu psikofizyolojik kompleks değişimler bir insan için bile uzmanlık isteyen bir alan olmuştur. Derinlikli öğrenme sayesinde bu uzmanlık yapay zekalara yüksek başarımlarla öğretilebilirse toplumsal güvenlik, kişisel sağlık gibi bir çok konuda makineler insanlara yardımcı olabilirler. Bu amaçla çalışmalar boyunca 3.1. bölümde detaylı açıklanan uykulu ve uykulu olmayan iki farklı sınıfa ait konuşmalardan oluşan SLC veri tabanını başarılı bir şekilde modellemek hedeflenmiştir.

1.2 Literatür Araştırması

Ses dosyaları dijital ortamda analog seslerin saniyede 10 binlerce ayrık örneği olarak saklanmaktadır. Saatlerce kayıt içeren SLC veritabanının böylelikle milyonlarca ayrık veriye sahip olduğu görülebilir. Derinlikli öğrenme yöntemleri ise veri üzerinden birden fazla geçerek yani iterasyonlar yaparak uygulanmakta olduğundan her iterasyon milyonlarca verinin işlenmesi pratik olarak mümkün olmamaktadır. Bu yüzden literatürde de ses ile yapılan araştırmalarda sesi daha az veri ile modelleme yoluna gidilmiştir.

Ses birçok bilgiyi taşımakla birlikte taşıdığı bilgilerin tümü insan kulağı için değer taşımamaktadır. Çalışmalar boyunca da insan beyni modellenmeye çalışıldığından konuşmalar, insan kulağına uygun bir şekilde sesi temsil edecek öznitelikler ile ifade edilebilirler. Daha önce başka makine öğrenmesi yöntemleri ile aynı veri tabanında yapılan çalışmalarda, ses özniteliklerinden saniye başına 9 farklı öznitelik ile başarılı sonuçlar elde edilebileceği gösterilmiştir[2].

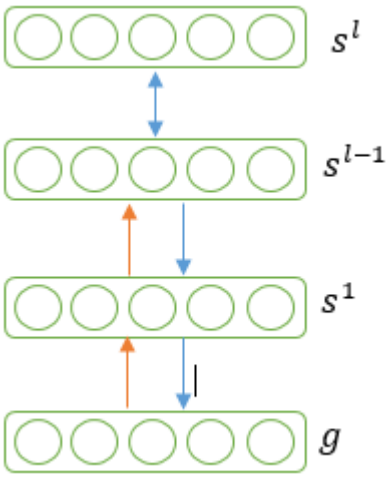
Derinlikli öğrenme terimi 2006 yılında Hinton tarafından çok katmanlı yapay sinir ağlarının verimli bir şekilde eğitilebileceği ortaya koyulduktan sonra literatürde sıklıkla geçmeye başlamıştır. Derinlikli öğrenmenin ses verilerinin modellemesinde de başarılı sonuçlar vermesi sebebiyle tanımlanan probleme iyi bir çözüm oluşturmaktadır[3].

2. DERİNLİKLİ İNANÇ AĞLARI(DBN)

Derinlikli inanç ağları, girişine uygulanan verinin hiyerarşik bir biçimde oluşturulmasını sağlayan graf-tabanlı modellerdir. Birden fazla RBM'in arka arkaya bağlanmasından meydana gelen DBN(derinlikli inanç ağları), yapısını oluşturan RBM'lerin sırasıyla eğitilmesi sayesinde öğrenilirler. Bu sayede girişe uygulanan veri(x***)*** ile aradaki saklı ℓ katman arasında ortak bir olasılık dağılımı (2.1) deki gibi modellenir. (2.1) de s^{ℓ}, ℓ. saklı katmandaki veriye karşı düşen vektördür. g görünür giriş katmanındaki veriye karşı düşen vektördür. Giriş katmanında eğitim amaçlı öznitelik vektörleri uygulandığından

$$P(g, s^1, \ldots, s^{\ell}) = \left(\prod_{k=0}^{\ell-2} P(s^k | s^{k+1}) P(s^{\ell-1}, s^{\ell}\right) \tag{2.1}$$

k=0 iken $g = s^0$olarak alınır. $P(s^k | s^{k+1})$ k. görünür katman için bir sonraki saklı katmanla oluşturduğu koşullu olasılıktır. $P(s^{\ell-1}, s^{\ell})$ ise son katmandaki görünür ve saklı RBM katmanlar arasındaki ortak olasılık dağılımıdır.



Şekil 2.1: Derinlikli Ağların Yapısı

2.1 Ön eğitim Aşaması

DBN ağının öğreniminin Hinton tarafından iki aşamada gerçekleştirilmesi önerilmiştir[4]. İlk aşama olan ön eğitim aşamasında eğiticisiz (unsupervised) sınıflandırma ile giriş verisini modelleyen ağırlık parametreleri öğrenilir. İkinci aşama ince ayar (finetuning) olarak adlandırılır ve öğreticili (supervised) sınıflandırma ile eğitim setini sınıflandıran ağ parametrelerini öğrenilmesini hedefler. DBN'deki ağırlık parametrelerinin ince ayar (finetuning) öncesi tayini için kullanılırlar. RBM'in ilk katmanı görünür katman olarak adlandırılır ve girişine $V_{eğitim}$ uygulanır bu nedenle $x = g = s^0$ olarak alınır. Tek saklı katmandan oluşan RBM'lerin üst üste konularak eğitilmesi fikri ilk olarak Hinton tarafından ortaya atılmıştır[5]. Bu tarz bir yaklaşımın matematiksel olarak doğruluğu iki saklı katmanlı bir DBN ağı ele alınarak açıklanabilir. DBN için olasılık dağılım fonksiyonu (2.2) şeklinde logaritmik halde yazılabilir.

$$\log p(g) = KL(Q(s^{(1)}|g)||p(s^{(1)}|g)) + H_{Q(s^{(1)}|g)} + \sum_s Q(s^{(1)}|g)(logp(s^{(1)}) + logp(g|s^{(1)})) \quad (2.2)$$

$Q(s^{(1)}|g)$ fonksiyonu, $p(s^{(1)}|g)$ fonksiyonunun sonsal olasılığıdır.

$$Q(s^{(1)}|g) = \frac{P(s^{(1)}|g)P(g)}{P(s^{(1)})} \quad (2.3)$$

$H_{Q(s^{(1)}|g)}$ sonsal olasılık dağılımının entropisidir. KL(Kullback–Leibler) ayrışması ise iki olasılıksal dağılımın farkını ölçmeye yarayan bir ayrışım fonksiyonudur.

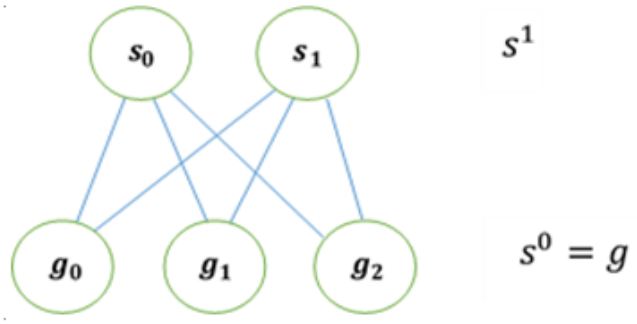
$$KL(Q(s^{(1)}|g)||p(s^{(1)}|g)) = \sum_s Q(s^{(1)}|g) logp\left(\frac{Q(s^{(1)}|g)}{p(s^{(1)}|g)}\right) \quad (2.4)$$

İki saklı katmanı ağırlık parametreleri $W^{(2)} = W^{(1)^T}$ olacak şekilde başlatırsak, $Q(s^{(1)}|g) = p(s^{(1)}|g)$ olacağı görülür. Dolasıyla KL ayrışması sıfıra eşit olur. Bu şekilde gözlenen veri kullanılarak eğitilen ilk katmanın ardından, $W^{(1)}$ ağırlıkları sabit tutulup, eğitilen ilk katmanı bir sonraki katmanın girişi olarak kullanarak $W^{(2)}$

ye göre ikinci katman eğitilebilir. Bu şekilde istenilen ℓ katman boyunca yukarı doğru ilerlenerek bütün katmanların ön eğitimi tamamlanmış olur. Alt bölümde ise tek bir RBM katman için eğitimin matematiksel modeli anlatılmaktadır.

2.1.1 Kısıtlanmış Boltzmann Makineleri (RBM)

Kısıtlanmış Boltzmann Makineleri (RBM), enerji fonksiyonlarının serbest parametreler için lineer olduğu logaritmik lineer Markov Tesadüfi Alanları'nın özel bir çeşididir. Çoğu zaman elimizde olan veriler istenilen modeli oluşturmak için yeterli değildir. İstenilen olasılık dağılımının tümünün gözlenmesi zordur. Bu yüzden saklı birimler yardımıyla hem modelin gücünü arttırarak hem de gözlenmemiş yeni değişkenler tanımlayarak bu sorun çözülmeye çalışılmıştır. RBM'ler görünür katmanının kendi aralarında ve saklı her katmanın kendi aralarında olan bağlantılarını kısıtlayarak ortaya çıkan özel bir yapıdır. RBM'lerin daha karmaşık dağılımları temsil edebilmeleri için saklı birimlerin sayısını arttırarak modelleme kapasitesi arttırılabilirken parametre sayısı arttığından aşırı-öğrenmeye(Overfitting) elverişli hale gelmemesi için dikkatli olunmalıdır. 3 görünür ve 2 saklı birimden oluşan bir RBM Şekil 2.2 deki gibi gösterilebilir.



Şekil 2.2: RBM ağının grafiksel gösterimi

RBM enerji tabanlı bir model olduğundan, modeldeki parametrelerin dağılımını bir skaler enerji olarak ilişkilendirir. Bu modellerde öğrenme, enerjinin probleme uygun olarak tanımlanması ve maliyet fonksiyonunu en küçükleyen model parametrelerinin kestirilmesidir. Bu amaçla enerji tabanlı olasılıksal modellerde I görünür işlemci birimi sayısı; J saklı işlemci birimi sayısı; enerji w_{ij} parametresi, g_i görünür işlem

birimi ile s_j saklı işlemci birimi arasındaki ağırlık katsayısı ; a_i ile b_j terimleri ise yardımcı terimler olacak şekilde enerji fonksiyonu (2.5) de görüldüğü gibi tanımlıdır.

$$E(g,s) = -\sum_{i=0}^{I} a_i g_i - \sum_{j=0}^{J} b_j s_j - \sum_{i=0}^{I}\sum_{j=0}^{J} g_i s_j w_{ij} \tag{2.5}$$

RBM ağındaki görünür ve saklı vektörler arasındaki olasılık, (2.5)'deki enerji fonksiyonunun üstel terimi olarak (2.6) daki gibi tanımlanabilir. (2.6)'daki Z terimine fiziksel sistemlerdeki benzerliği sebebiyle bölümleme fonksiyonu adı verilmiştir ve normalizasyon terimi görevi görmektedir.

$$p(g,s) = \frac{1}{Z} e^{-E(g,s)} \tag{2.6}$$

Z bölümleme fonksiyonu (2.7) deki gibi tanımlı olup olası bütün görünür ve saklı vektörlerin kombinasyonlarının enerjileri toplamıdır. Bu sayede toplam olasılığın bire eşit olmasını sağlar.

$$Z = \sum_{g}\sum_{s} e^{-E(g,s)} \tag{2.7}$$

Böylelikle RBM ağının g görünen vektörüne atadığı bileşik olasılık (2.8) de olduğu gibi tüm olası saklı birim vektörünün olasılıkları toplamına eşit olur.

$$P(g) = \sum_s P(g,s) = \sum_s \frac{e^{-E(g,s)}}{Z} \tag{2.8}$$

RBM ağının görünen vektör g'ye atadığı olasılığı arttırmak için (2.8) den de görülebileceği gibi enerji fonksiyonu minimize edilmelidir. (2.5) deki enerji fonksiyonunu minimize edecek fonksiyon parametrelerinin elde edilmesi SGD ile mümkün olmaktadır. (2.8) denklemi üzerinden w_{ij} için kısmi türev aşağıdaki gibi (2.9) şeklinde elde edilebilir [6].

$$\frac{\partial \log(p(g))}{\partial \mathrm{w_{ij}}} = -\sum_{s} p(s|g) \frac{\partial \mathrm{E}(g,s)}{\partial \mathrm{w_{ij}}} + \sum_{g,s} p(g,s) \frac{\partial \mathrm{E}(g,s)}{\partial \mathrm{w_{ij}}} \tag{2.9}$$

RBM'lerde saklı/görünür katmanlar kendi aralarında bağlı olmadığı için her bir birimin, olasılıksal olarak birbirinden bağımsız olduğu kabul edilebilir. Bayes eşitliğinden yararlanarak saklı birimler vektörünün(s) görünür birimler vektörüne(g) bağlı olan koşullu olasılığı (2.10) ve görünür birimler vektörünün(g) saklı birimler vektörüne(s) bağlı koşullu olasılıkları (2.11) şeklinde elde edilir.

$$P(\mathrm{s}|g) = \frac{P(g,s)}{P(g)} = \frac{\frac{1}{Z} e^{-E(g,s)}}{\sum_s \frac{e^{-E(g,s)}}{Z}} = \prod_{i=1}^{I} P(\mathrm{s_i}|g) \tag{2.10}$$

$$P(g|s) = \prod_{j=1}^{J} P(\mathrm{g_j}|s) \tag{2.11}$$

(2.9) denklemi negatif ve pozitif iki ayrı terimden oluşmaktadır. Literatürde bu terimlere sırasıyla pozitif faz ve negatif faz adı verilmektedir. Buradaki ayrımı sağlayan, terimlerin işareti değil model tarafından belirlenen olasılık dağılımına yaptıkları etkidir. Pozitif faz $V_{eğitim}$ setinin olasılığını yükseltip modelin enerjisini düşürürken, negatif faz ise model tarafından oluşturulan örneklerin olasılığını düşürür. Pozitif faz terimi, $V_{eğitim}$ seti gözlendiğinde beklenen enerji toplamının gradyanı olduğundan $E(g_i s_j)_{veri}$ olarak, negatif faz terimi ise model tarafından belirlenen enerji gradyanı toplamı olduğundan $E(g_i s_j)_{model}$ ile gösterilebilir. Böylelikle (2.9) denklemi (2.12) şeklinde ifade edilebilir.

$$\frac{\partial \log(p(g))}{\partial \mathrm{w_{ij}}} = E(g_i s_j)_{veri} - E(g_i s_j)_{model} \tag{2.12}$$

(2.12) deki ilk terim olan $E(g_i s_j)_{veri}$ yi analitik olarak hesaplamak mümkündür. İkili sistemli birimlerin bulunduğu(g ve $\mathbf{s_i} \in \{\mathbf{0}, \mathbf{1}\}$) ağlarda saklı birimin aktif olma olasılığı (2.10) denkleminden (2.13) formuna sadeleşebilir, görünür birimin aktif olma olasılığı (2.11) denkleminden (2.14) formuna sadeleştirilebilir[7]. Böylelikle pozitif faz için (2.13) ile görünür birimler vektörü bilinirken saklı birimler örneklenebilir.

$$P(\mathrm{s_j} = 1|g) = \frac{1}{1 + e^{(-(\mathrm{b_j} + \sum_i g_i w_{ij}))}} = sigm(\mathrm{b_j} + \sum_i g_i w_{ij}) \tag{2.13}$$

Aynı şekilde (2.14) ile saklı birimler vektörü bilinirken görünür birimler örneklenebilir.

$$P(\mathrm{g_i} = 1|s) = \frac{1}{1 + e^{(-(\mathrm{a_i} + \sum_j s_j w_{ij}))}} = sigm(\mathrm{a_i} + \sum_j s_j w_{ij}) \tag{2.14}$$

(2.12) deki negatif faz $E(g_i s_j)_{model}$ i analitik olarak hesaplamak ise RBM ağının boyutuyla eksponansiyel olarak artmaktadır. Bu yüzden $E(g_i s_j)_{model}$ hesaplanırken Hinton tarafından önerilen Contrastive Divergence(CD) ıraksayı kullanılacaktır[8].

2.1.1.1 RBM'lerde örnekleme

p(g) dağılımından hesaplanacak örnekler, Gibbs örneklemesini geçiş operatörü olarak kullanıp Markov Zinciri işletilerek bulunabilir. **N** adet rastgele değerin $S = (S_1, \dots, S_N)$ Gibbs örneklenmesi, $S_i \sim p(S_i|S_{-i})$ formunda **N** örneklenme adımıyla gerçekleşir. Bu adımlar sırasında S_{-i} , S_i hariç **N-1** adet diğer değeri içerir.

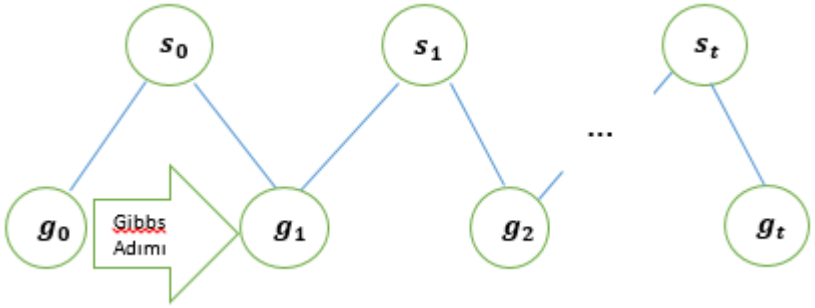
RBM'ler için ise ***S*** görünür ve saklı birimleri içerir. Fakat bu birimler olasılıksal olarak birbirinden bağımsız olduklarından ayrı olarak Gibbs örneklenmesi gerçekleştirilebilir. Bu şekilde görünür birimler, saklı birimlerin sabit değerleri kullanılarak örneklenirken aynı zamanda saklı birimler de görünür birimlerden örneklenebilirler. Bu şekilde gerçekleşen bir Markov Zinciri adımı sırasıyla (2.15) ve (2.16) uygulanarak sonuçlanır.

$$s^{(n+1)} \sim sigm(\mathrm{W}'.g^{(n)} + \mathrm{b}) \tag{2.15}$$

$$g^{(n+1)} \sim sigm(\mathrm{W}.s^{(n+1)} + \mathrm{a}) \tag{2.16}$$

Yukarıda $s^{(n)}$ Markov zincirinin n. Basamağındaki bütün saklı birimleri temsil eder. Yukarıdaki denklemi örnek olarak ikili birimli RBM'ler için açıklanacak olursa, $s_i^{(n+1)}$ in 0 yerine 1 olması $sigm(W_i'.g^{(n)} + \mathrm{b}_i)$ olasılıkla rastgele seçilmektedir. Benzer bir biçimde $g_j^{(n+1)}$ in de 0 değil de 1 olma olasığı $sigm(W_j.s^{(n+1)} + \mathrm{a}_j)$ dir.

Grafiksel olarak göstermek gerekirse:



Şekil 2.3: Gibbs örneklenmesi şeması

"***t***" sonsuza gittikçe elde edilen $g^{(t)}$ve $s^{(t)}$örneklerinin ***p(g,s)*** olasılıksal dağılımına ait güvenilir örnekler olma şansı artmaktadır. Teorik olarak öğrenme aşamasındaki her bir parametre güncellemesi için bu zincirin yakınsaması uzun sürecektir. Bu yüzden ***p(g,s)*** dağılımından örneklerin daha verimli elde edilmesi için RBM'lerde uygulanabilecek CD algoritması kullanılmaktadır.

2.1.1.2 Karşılaştırmalı Iraksama (CD)

Karşılaştırmalı Iraksama yöntemi temel olarak 2 adet varsayım kullanarak DBN'lerde örnekleme işlemini daha kısa hale getirmektedir. Bu varsayımlar aşağıdaki gibidir:

- Modelden istenen olasılık dağılımı son durumda $V_{eğitim}$ setinin olasılık dağılımına yakın olması beklendiğinden, Markov zincirimizi $V_{eğitim}$ setinden bir örnekle başlatılabilir. Bu sayede istenilen dağılıma yakın bir dağılım kullanılacak olduğundan örnekleme zinciri sonuç olarak beklenen dağılıma yakınsar.
- CD, Markov zincirinin yakınsamasını beklemez. Yani sadece "k" adım Gibbs örneklemesi sonucu elde edilen örnekler kullanılır. Hatta pratikte 1 adım Gibbs örneklemesi kullanılarak bile yeterli sonuçlar elde edilebilmektedir.

2.1.1.3 RBM Ağındaki Parametrelerin Güncellenme Denklemleri

CD algoritmasıyla birlikte (2.25) de kullanılacak parametre güncelleme denklemleri bütün parametreler için, $\tilde{g}$ Gibbs örneklemesi sonucu elde edilen örnekler olacak şekilde (2.17) şeklinde elde edilir.

$$-\Delta \mathrm{w_{ij}} = \sum_{s} p(s|g)\frac{\partial \mathrm{E}(g,s)}{\partial \mathrm{w_{ij}}} - \sum_{g,s} p(g,s)\frac{\partial \mathrm{E}(g,s)}{\partial \mathrm{w_{ij}}} \tag{2.17}$$

$$= P(\mathrm{s_j}=1|g)\mathrm{g_j} - \sum_{i} p(g)\,P(\mathrm{s_j}=1|g)\mathrm{g_j}$$

$$= sigm\left(\mathrm{b_j} + \sum_{i} g_i w_{ij}\right) g_i - sigm\left(\mathrm{b_j} + \sum_{i} \tilde{g}_i w_{ij}\right)\tilde{g}$$

$$= E(g_i s_j)_{veri} - E(g_i s_j)_{model}$$

$$-\Delta \mathrm{a_i} = \mathrm{g_j} - \sum_{i} p(g)\,\mathrm{g_j} = \mathrm{g_j} - \tilde{g} = E(g_i)_{veri} - E(g_i)_{model}$$

$$-\Delta \mathrm{b_j} = P(\mathrm{s_j}=1|g)\mathrm{g_j} - \sum_{i} p(g)\,P(\mathrm{s_j}=1|g)$$

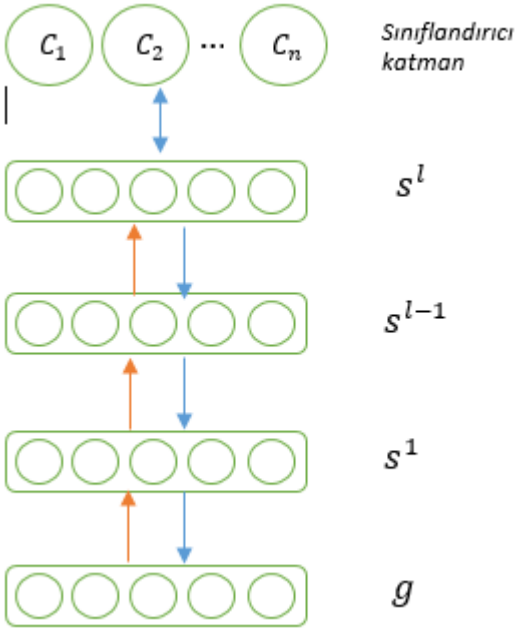
$$= sigm\left(\mathrm{b_j} + \sum_{i} g_i w_{ij}\right) - sigm\left(\mathrm{b_j} + \sum_{i} \tilde{g}_i w_{ij}\right)$$

$$= E(s_j)_{veri} - E(s_j)_{model}$$

DBN ağının parametrelerinin gözetimsiz eğitimi istenilen SGD iterasyonu kadar gerçekleştirildikten sonra ön eğitim aşaması son bulur.

2.2 İnce ayar (Finetuning) Aşaması

RBM olarak katman katman eğitilen DBN de en uygun parametreleri belirlemek için ince-ayar aşamasına geçilir. İnce-ayar aşamasında ön eğitim aşamasında öğrenilen ağ yapısını, makine öğrenmesi yöntemlerinden bir sınıflandırıcı katmanı ile sonlandırarak, çok katlı RBM modeli bir sınıflandırıcıya dönüştürülür.



Şekil 2.4: n adet sınıf için ince-ayar aşamasında DBN yapısı

DBN ağının sınıflandırıcı performansını, yani girişine uygulanan herhangi bir veriyi doğru sınıflandırma ihtimalini arttırmak için ağ parametreleri güncellenerek maksimum başarımı sağlayacak parametreler belirlenmelidir. Maksimum başarımı sağlayacak ağ parametrelerini hesaplamak için önce kullanılacak sınıflandırıcı katman dikkate alınmalıdır.

2.2.1 Lojistik Regresyon

Makine öğrenmesi uygulamalarında en çok kullanılan yöntemlerden biri olan Lojistik Regresyon basit ve güçlü bir lineer sınıflandırıcıdır. ***W*** ağırlık matrisi ve ***b*** yardımcı vektörüyle parametrik halde Lojistik Regresyon denklemi aşağıdaki gibidir.

$$P(C_i|g,W,b) = softmax_i(Wg+b) = \frac{e^{W_i g+b_i}}{\sum_j e^{W_j g+b_j}} \tag{2.18}$$

Softmax aktivasyon fonksiyonu sayesinde çıkışta beklenen sınıf sayısı boyutunda yeni bir sınıflandırıcı katman RBM katmanlarının üstüne eklenir. Softmax fonksiyonu, çıkış katmanının toplam sonucunun 1 olmasını sağlayacak şekilde çıkışları normalize eder. Bu sayede her bir sınıfa ait olan çıkış aktivasyonunun değeri, girişin o sınıfa ait olma olasılığını vermiş olur. Sınıflandırma temel olarak **x** giriş vektörünü, sınıflara karşılık gelen hiper-düzlemlere yansıtarak, girişin hiper-düzleme olan uzaklığına göre o sınıfa(**i**) ait olma olasılığını belirler. Bu şekilde herhangi bir giriş vektörünün hangi sınıfa ait olduğu tahmin edilmek istenirse (2.18)' e göre maksimum olasılığa sahip olan sınıf (2.19) deki gibi belirlenir.

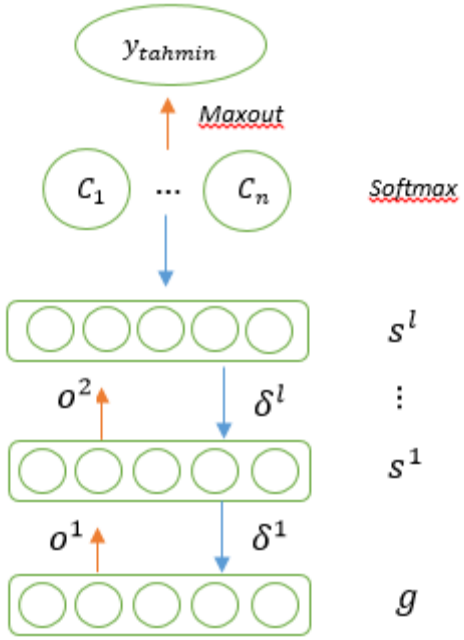
$$y_{tahmin} = argmax_i P(C_i|g,W,b) \tag{2.19}$$

2.2.2 Geri Yayma Yöntemi (Backpropagation)

İnce ayar aşamasında, görünür katmanın son saklı katmanı s^{ℓ} , lojistik regresyon kullanılarak sınıflandırılıp ortaya çıkacak maliyet fonksiyonunun minimize edilmesi amaçlanır. DBN ağına giriş olarak $V_{eğitim}$ veri seti uygulanarak, SGD ile parametrelerin güncellenmesi gereklidir. Fakat çıkış katmanından geriye doğru her iterasyonda (2.17) gibi türevleri hesaplamak maliyetli bir işlemdir. Geri yayma algoritması sayesinde parametre güncellemelerindeki gradyanı hesaplamak yerine çıkış katmanından başlayarak giriş katmanına doğru sırasıyla yaklaşık türev değerinin hesaplanması sağlanır. Hinton tarafından önerilen bu yaklaşım [4], DBN üzerinde son katmandan başlayarak geriye doğru modelin tahminiyle o verinin gerçek sonucu arasındaki farkı hata kabul edip (2.20), her katmandaki hatayı geriye

doğru hesaplayarak maliyet fonksiyonundaki kısmi türevin yaklaşık ifadesinin hesaplanmasıyla parametrelerin güncellenmesi amaçlanır.

$$\frac{\partial \mathrm{E}(g,s)}{\partial {w_{ij}}^{l}} = {\delta^{l}}_{j} {o_{i}}^{l} \tag{2.20}$$



Şekil 2.5: Geri-Yayma algoritması uygulama şeması

Tek bir giriş örneği için geri yayma algoritması Şekil 2.5 deki gibi çalışmaktadır. Buna göre öncelikle giriş örneğine göre ileriye doğru her katman için o çıkış vektörleri (2.21) deki gibi hesaplanır.

$${o_{j}}^{l} = sigm({\mathrm{b_j}}^{l} + \sum_{i} {s_{i}}^{l} {w_{ij}}^{l}) \tag{2.21}$$

Ardından DBN ağından elde edilen y_{tahmin} için son katmanın hatası (2.22) e göre hesaplanır.

$$\delta^{l} = (y_{tahmin} - y_{gerçek})(1 - y_{tahmin})\, y_{tahmin} \tag{2.22}$$

Ardından (2.21) ile elde edilen son katman hatası geriye doğru en alt katmana kadar (2.22) ile yayılır.

$$\delta^{l-1}{}_j = o_j{}^{l-1}(1 - o_j{}^{l-1}) \sum_i \delta_i{}^l w_{ij}{}^l \tag{2.23}$$

Her katman için elde edilen hatalar böylelikle (2.20)'de yerine konularak parametre güncellemeleri sağlandıktan sonra, bu işlemler sıradaki örnek için tekrardan uygulanır. Böylelikle son katmanın hatası düştüğü sürece geri-yayma algoritmasıyla birlikte SGD işletilmeye devam ettikten sonra ince-ayar aşaması da son bulmuş olur.

2.2.3 Sınıflandırma

İnce-ayar aşaması sonucunda kesinleşen ağ parametreleriyle birlikte DBN son halini aldıktan sonra daha önce görülmemiş veri setlerinin sınıflandırılmasında kullanılabilir. Daha önceden V_{test} olarak ayrılan test veri seti, ince ayar aşamasında olduğu gibi DBN in bir lojistik regresyon katmanı ile sonlandırılmasıyla sınıflandırılabilir.

V_{test} deki örnekler DBN in girişi olacak şekilde (2.18) in sonucu, örneğin o sınıfa ait olma olasılığını vermektedir. Bu sayede (2.19) ile en yüksek olasılığa sahip sınıf, girişe uygulanan örneğin sınıfı olarak belirlenmiş olur.

3. DBN'LERİN BAŞARIM ÖLÇÜMÜ

Herhangi bir makine öğrenmesi yönteminin başarısının belirlenmesi için daha önceden başarısı kanıtlanmış yöntemlerle karşılaştırılması gerekmektedir. DBN yöntemi bu amaçla halihazırda başka yöntemlerle sınıflandırılmış ve başarım oranları ölçülmüş gerçek bir veri seti kullanılarak test edilmelidir.

3.1 Sleepy Language Corpus Veritabanı

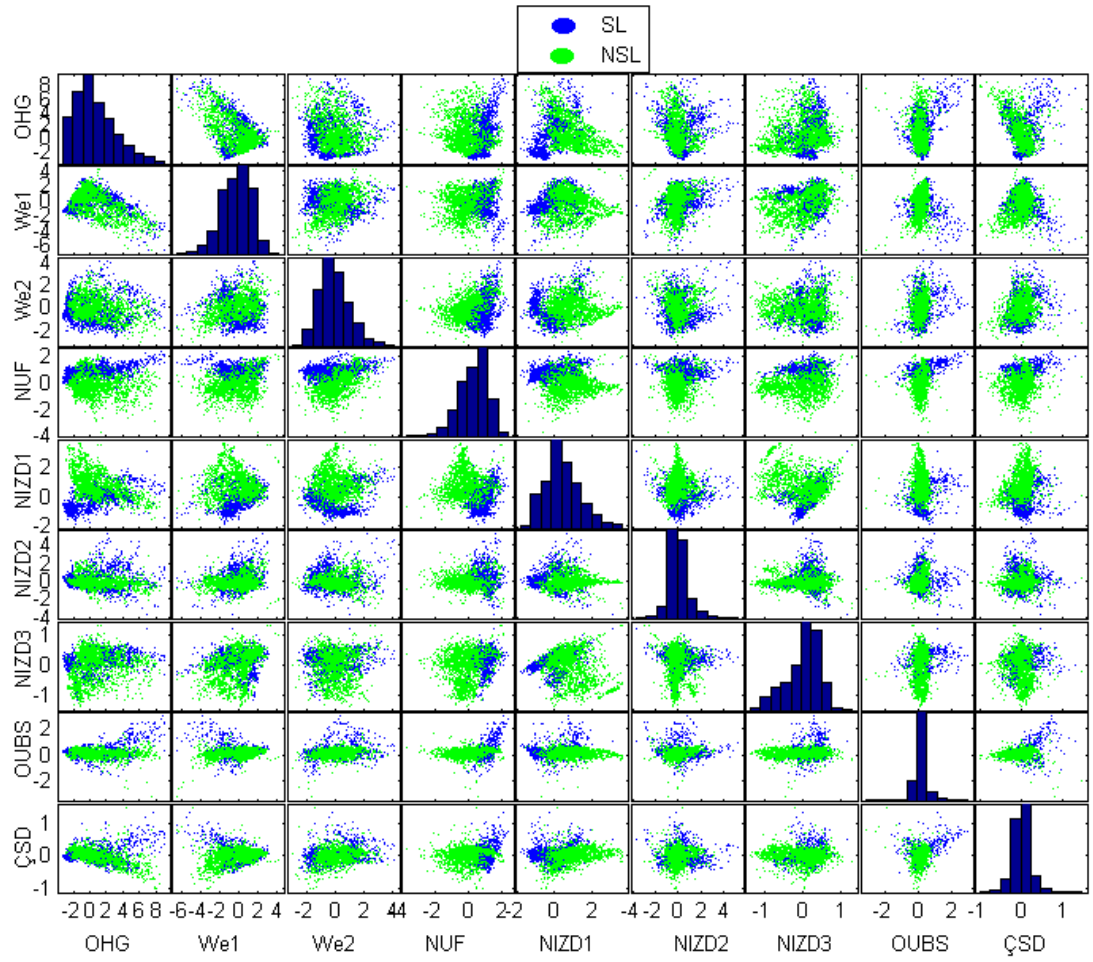
Bu amaçla DBN ağının başarım testleri için "Sleepy Language Corpus" [9] veri seti kullanılmıştır. Bu veri seti uzmanlar tarafından uykulu(SL) veya uykusuz(NSL) olarak etiketlenmiş seslerden oluşmaktadır. 2 sınıftan oluşan bu veritabanının DBN

yöntemiyle öğrenilebilmesi için giriş verisi olan seslerin yönteme uygun halde derinlikli ağın girişine verilmesi gerekir. DBN'ler her ne kadar kompleks verileri modelleyebilecek güçlü bir ağ olsalar da ses verileri girişe direk uygulanmak için fazlasıyla büyüktürler. Bu yüzden uygulamalarda ses yerine sesin belli özelliklerini temsil eden ses öznitelikleri kullanılmaktadır[10]. Bu sayede ses saniye başına, SL-NSL sınıflandırma problemi için başarısını kanıtlamış Tablo 3.1.'de gösterilen 9 öznitelik ile temsil edilecektir.

Tablo 3.1: Sesi temsil eden 9 öznitelik

	Ses Ölçeği	**Öznitelik Adı**	**Öznitelik Açıklaması**
1	**Hz**	Ortalama Harmoniklik Genliği(OHG)	Logaritmik izgedeki uykululuk değişimlerinin korelasyonlarından elde edilen temel frekansların ortalaması
2		10db Algısal Bant Genişliği (W_{E1})	Gürültü eşiğini en az 10dB aşan en yüksek frekans bileşeni
3		5db Algısal Bant Genişliği (W_{E2})	Gürültü eşiğini en az 5dB aşan en yüksek frekans bileşeni
4	**Bark**	Normalize Edilmiş Uykululuk Düzeyi Farkı(NUF)	SL/NSL konuşmaların perde örüntülerinin ve bir ses çerçevesinden bark ölçeğinde hesaplanan referans sesler arasındaki maskelenmiş değişimlerin ortalaması
5		Normalize Edilmiş İzgesel Zarf Farkı(NIZD1)	Her bir kritik banttaki ardışık çerçevelerden elde edilen referans sesler için SL/NSL perde örüntülerinin normalize edilmiş zarf değişimleri
6		Normalize Edilmiş İzgesel Zarf Farkı(NIZD2)	NIZD1 özniteliğinin tüm kritik bantlardaki ortalaması
7		Normalize Edilmiş İzgesel Zarf Farkı(NIZD3)	NIZD1 özniteliğinin ardışık Y ses çerçevesindeki zamansal ortalaması
8		Ortalama Uykusuz Blok Sayısı (OUBS)	Belirli bir zaman aralığındaki beklenen NSL bloğu sayısı
9		Çerçevenin Seslilik Değeri (ÇSD)	Bir ses çerçevesindeki, dış kulağın tüm kritik bantlarına göre ağırlıklandırılmış seslilik değerlerinin toplamı

21 saatlik SLC veritabanından öznitelikler elde edildikten sonra, özniteliklere göre verinin dağılım grafiğini çizdirdiğimizde verinin sınıflandırmaya uygun bir biçimde ayrıştığı Şekil 3.1'de görülebilir.



Şekil 3.1: SLC veri tabanının özniteliklere göre dağılımı

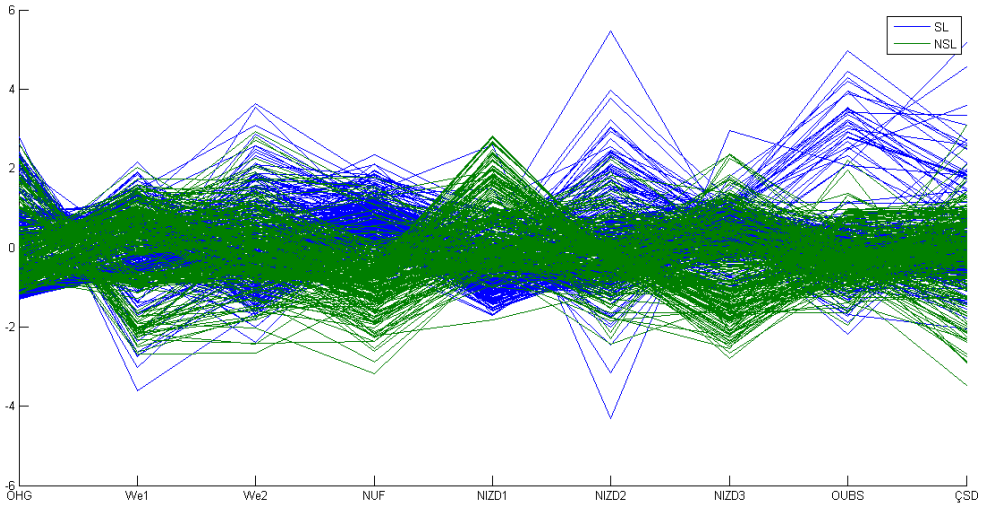
3.1.1 Test Senaryoları

Veri literatürdeki sonuçlarla karşılaştırabilmek açısından 2011 yılındaki "Speaker State Challenge" yarışmasında belirtilen şekilde eğitim, test ve doğrulama kümesi olarak üçe bölünmüştür[9][11]. Ardından ilk test senaryosu için veri SL ve NSL sınıfından örnekler için rastgele Tablo 3.2'de belirlenen miktarlarda her sınıf için eşit sayıda örneklerek en az sayıda örnekle en yüksek başarım hedeflenmiştir.

Tablo 3.2: Veri kümelerine göre test senaryoları

Test	Eğitim Kümesi	Eğitimde kullanılan örnek sayısı			Test Kümesi	Test için kullanılan örnek sayısı		
		SL	NSL	Toplam		SL	NSL	Toplam
1	Öbeklenmiş Eğitim Kümesi	50	50	100	Doğrulama Kümesi	300301	241593	541894
		100	100	200				
		500	500	1000				
		1500	1500	3000				
		4000	4000	8000				
		10000	10000	20000				
		23974	23980	47954				
2	Öbeklenmiş Eğitim Kümesi + Öbeklenmiş Doğrulama Kümesi	50	50	100	Test Kümesi	114810	104902	219712
		100	100	200				
		500	500	1000				
		1500	1500	3000				
		4000	4000	8000				
		10000	10000	20000				
		23974	23980	47954				

Öbeklenmiş eğitim kümesinde 50'şer adet rastgele örneklenen veriyi paralel koordinat grafiği olarak çizersek 100 örnekte bile ayrışım sağlandığı Şekil 3.2'de görülebilir.



Şekil 3.2: Öbeklenmiş eğitim kümesinin paralel koordinat grafiği

3.2 DBN algoritmasının gerçeklenmesi

Teorik olarak bahsedilen DBN'lerin bilgisayar ortamında gerçeklenmesi mümkündür. Son yıllarda DBN'lerin gerçeklenmesi için birçok ortam kullanılabilmektedir. Testler boyunca DBN'lerin gerçeklenmesi için Theano adlı kütüphane kullanılacaktır.

3.2.1 Theano

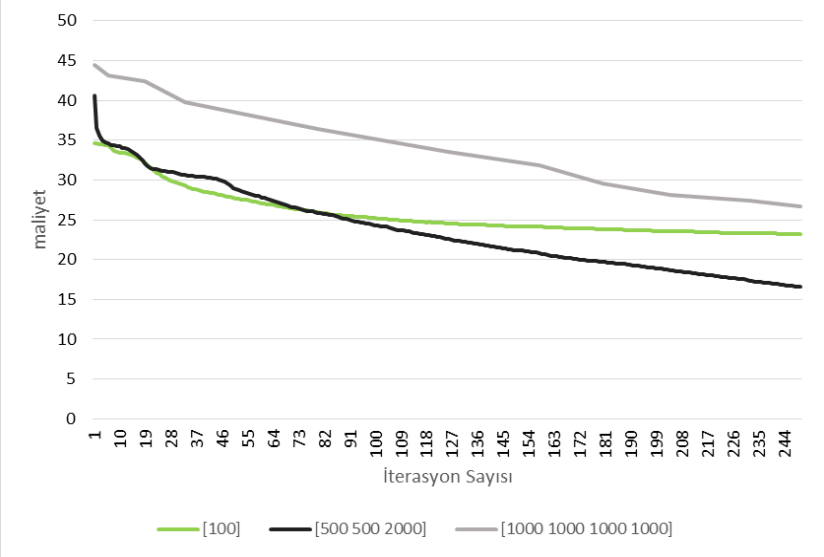
Theano özellikler derinlikli ağların gerçeklenmesi için oluşturulmuş bir Python kütüphanesidir[12][13]. Derinlikli ağların yüksek işlem gücüne ihtiyaç duymasından dolayı geliştirilen Theano, matematiksel ifadelerle işlem yapmayı sağlaması ve bu işlemlerin numerik optimizasyonunu otomatik yapması sebebiyle DBN'lerin gerçeklenmesine uygundur. Ayrıca GPU kullanılmasına imkan tanıyarak matris çarpım işlemlerini hızlandırması sebebiyle daha derin ve büyük ağların daha hızlı eğitilmesine imkan tanımaktadır. Gerekli kurulumlar yapıldıktan sonra, ortamda ikinci bölümde anlatılan DBN yapısı gerçeklenmiş, ardından da SLC veritabanı 3.1.'bölümde anlatıldığı şekilde modellenmeye uygun hale getirilmiştir. Sonuçlar raporlanırken ise yine Python'da bulunan Scikit kütüphanesi kullanılarak başarım sonuçları literatüre uygun ölçümler şeklinde elde edilmiştir

3.2.2 DBN'lerde uygun parametrelerin seçilmesi

Derinlikli ağların oluşturulması ve eğitilmeleri sırasında momentum katsayısı, ön-eğitim adım boyutu, ince-ayar (finetuning) adım boyutu , katman sayısı, katman genişliği, Gibbs örneklenme adımı vb. gibi bir çok parametre kullanılmaktadır. Bu parametreler probleme uygun bir biçimde seçilen kayıp fonksiyonunu minimize edecek şekilde seçilmelidirler. Parametrelerinin seçilmesinin analitik bir yolu olmadığı gibi ızgara arama yöntemi benzeri tüm parametre kombinasyonlarının denenemeyeceği kadar büyük bir domene sahip olmalarından dolayı DBN'lerde parametre seçimi bu yöntem gerçeklenirken en çok dikkat edilmesi gereken kısımlardan biridir.

3.2.2.1 Ağ Derinliğinin ve İşlem Birimi Sayısının Seçilmesi

İşlemci birimi ve katman sayısı arttığında DBN'lerin modelleme kapasitesi artmaktadır. Fakat her problemin karmaşıklığı farklı olduğundan kullanılacak ağ yapıları da farklı olmaktadır. Problemde daha fazla modelleme kapasitesinin kullanımın sonuca pek etkisi olmasa da yapılan iterasyonlar daha uzun sürecek ve aşırı öğrenme probleminin aşılması daha zor olacaktır. Problemin karmaşıklığından daha az sayıda birim kullanılırsa da DBN'in modelleme kapasitesi yetersiz kalacağından az-öğrenme sorunu ortaya çıkacak ve verimsiz bir öğrenme olacaktır.
Bu yüzden Şekil 3.3 de görüldüğü gibi literatürde kullanılmış birkaç farklı ağ yapısı problemde denenmiştir.



Şekil 3.3: Farklı ağ yapılarının DBN modelinin eğitimine etkileri

Şekil 3.3'de görüleceği üzere 100 birimlik tek katmanlı yapı, problemin karmaşıklığı için yeterli olmadığından belli bir noktadan itibaren gelişim göstermemekte iken, 1000'er birimli 4 katmanda oluşan yapının ise gelişim göstermesi çok yavaş gerçekleşmektedir. Bu yüzden probleme en uygun olan 500, 500, 2000 işlemci birimli, 3 katmanlı bir yapıda karar kılınmıştır.

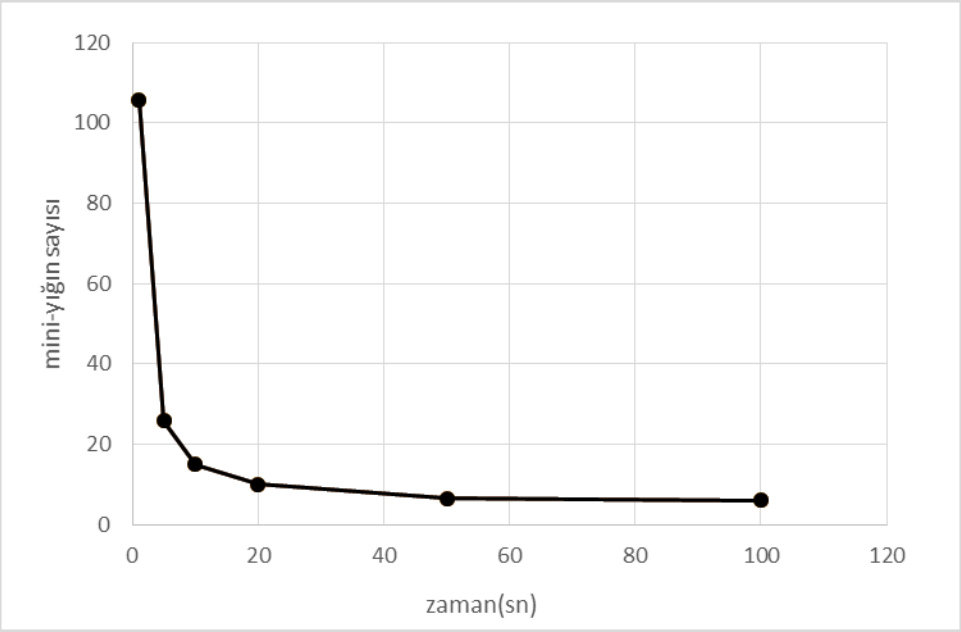
3.2.2.2 Gradyan Azaltma Algoritması

Gradyan azaltması(GD) optimizasyon algoritması tek bir adım gradyan hesaplaması için bile bütün veriye ihtiyaç duyar. Özellikle iterasyonlarla birçok defa GD hesaplanması gereken uygulamalarda bu işlem zaman ve performans kaybı

oluşturmaktadır. GD yerine kullanılacak SGD algoritması ise veriyi rastgele karıştırıp tek bir örneğin gradyanı hesaplanarak parametreleri güncelleneme esasına dayanır. Böylelikle hem bütün örnekleri beklemeden işlem gerçek zamanlı olarak gerçeklenebilirken, aynı zamanda iterasyonlar sonucu çözülmeye çalışılan optimizasyon problemi daha hızlı bir şekilde fonksiyonun global minimumuna yakınsar.

Günümüz bilgisayarlarının çalışma prensipleri vektörel çarpımlara göre optimize edildiğinden özellikle GPU(Graphical Processing Unit) üzerinden yapılan işlemlerde SGD gibi teker teker güncellemeler yerine, örnekler rastgele gruplar halinde GD algoritmasına uygulanır. Bu algoritmaya ise Mini-yığın gradyan azaltma adı verilir. ϵ adım büyüklüğü, $\boldsymbol{n}$ tek seferde kullanılan veri sayısı olacak şekilde parametre güncellemeleri (2.25) deki gibi hesaplanılabilir.

$$\theta := \theta - \epsilon \sum_{i=1}^{n} \left(-\frac{\partial \log(p(g))}{\partial \theta} \right) = \theta - \epsilon . \Delta\theta(t) \tag{2.25}$$



Şekil 3.4: Mini-yığın sayısı ile iterasyon süresinin değişimi

Yapılan testler sonucu Şekil 3.4'de görüldüğü gibi, mini-yığın sayısı ile tek bir parametre güncellemesi için yapılan iterasyon süresi ters orantı ile değişmektedir. Her ne kadar mini-yığın sayısının minimumda tutulması hız açısından olumlu gibi

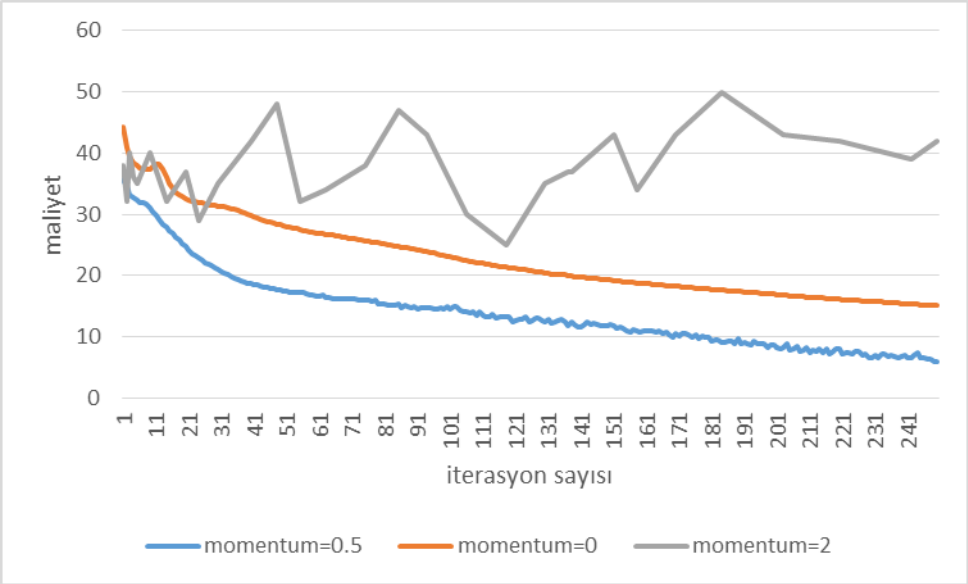
gözükse de, tek seferde dikkate alınacak veri miktarı düştüğünden Gibbs örneklenmesi sonucu elde edilecek değerler toplam veriye dağılım olarak daha az uygun olacaktır. Bu yüzden çalışmalar boyunca genellikle ***n*** parametresi 64 olarak seçilmiştir. Bu sayede bilgisayar mimarisine daha uygun matris çarpımları sayesinde daha yüksek performanslar elde edilmiştir.

3.2.2.3 Momentum

Momentum, öğrenme aşamasını daha da hızlandırmak adına uygulanan bir optimizasyon yöntemidir. Fizik alanındaki momentumdan esinlenilerek oluşturulan bu yöntem maliyet fonksiyonu üzerinde mini-yığın gradyan azaltması ile minimuma doğru ilerlerken aynı yönde atılan arka arkaya adımları hafızada tutarak bir sonraki adımda o yöne doğru olan hızı koruma esasına dayanır. Bu sayede fonksiyonun minimum noktasına daha çabuk ulaşılması hedeflenir. ϵ adım büyüklüğü, ***n*** tek seferde kullanılan veri sayısı, a ise momentum katsayısı olacak şekilde momentumlu Mini-yığın gradyan azaltma algoritmasıyla parametreler (2.26) deki gibi güncellenir.

$$\Delta\theta(t) = v(t) = a.v(t-1) - \epsilon \sum_{i=1}^{n} \left(-\frac{\partial \log\left(p(g)\right)}{\partial \theta}\right) \tag{2.26}$$

Momentum uygulanmasıyla birlikte problemin daha hızlı minimuma ulaştığı 500,500,2000 nörönlü 3 katmanlı bir ağda yapılan testlerde Şekil 3.5 teki gibi gösterilmiştir.



Şekil 3.5: Momentum yönteminin kıyaslanması

Momentum katsayısı birden büyük seçildiğinde maliyet fonksiyonunda dalgalanmalar gerçekleşmekte ve fonksiyon optimize edilememektedir. Başarım testlerinde bu sebeple momentum katsayısı 0.5 olarak tercih edilmiştir.

3.2.2.4 GPU kullanımı

GPU(Grafik İşlemci Ünitesi)'lar bilgisayarlarda görüntü işleme işlemlerini daha hızlı yapabilmek adına özelleşmiş işlemci birimleridir. Nvidia'nın CUDA kütüphanesini duyurmasının ardından GPU'lar görsel işlemlerin yanı sıra matematiksel hesaplama işlemlerinde de CPU'lardan daha avantajlı konuma gelmiştir. GPU'lar aynı anda birçok işlemi yapmak için optimize birimler olduğundan matris işlemlerinde CPU'lara oranla inanılmaz bir hız artışı sağlamıştır. GPU'ların paralel programlamada etkin hale gelmesiyle birlikte akademik alanda da sağladığı hız artışı sebebiyle günlerce sürecek işlemleri saatlere indirgemiştir.

Theano, CUDA destekli bir kütüphane olduğunda işlemler için GPU'lar kullanılabilmektedir. Bunun için gerekli ayarlamalar yapıldıktan sonra DBN için sağladığı hız artışı 500, 500, 2000 işlemci birimli bir ağda bir iterasyon için Tablo

3.3’deki sonuçlar elde edilmiştir. Buna göre GPU ile kurulan ağlarda 20 kattan fazla hız artışı sağladığı görülmüştür.

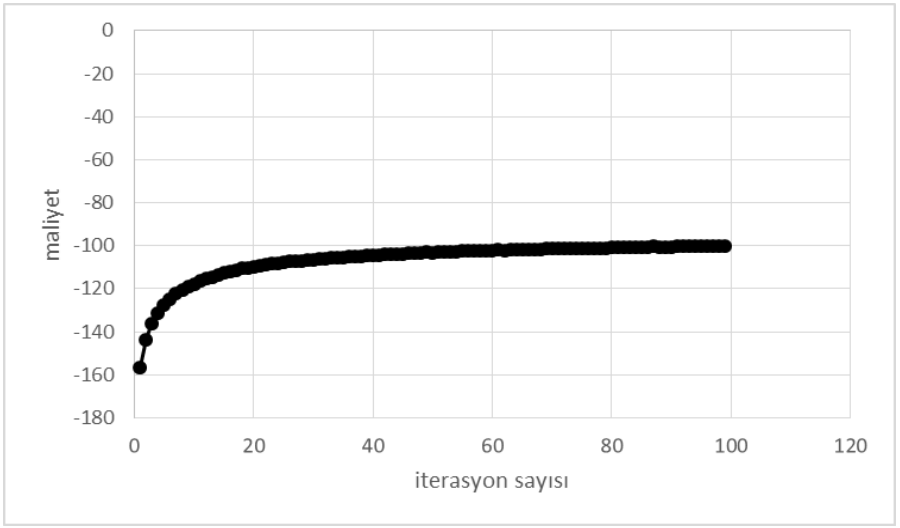
Tablo 3.3: Çeşitli işlemcilerde [500 500 200]’lik yapı için iterasyon hızları

	BLAS versiyonu		
İşlemci birimi	**Numpy**	**ATLAS**	**CuBLAS**
Intel i5 4200u CPU	28'22"	12'50"	13'29"
Intel i7 4790 CPU	13'55"	05'05"	04'55"
Nvidia GeForce GT 740M GPU	05'35"	02'58"	02'02"
Nvidia GeForce GTX 780 GPU	01'48"	00'56"	00'50"

DBN’ler için bilgisayardaki diğer kritik nokta ise BLAS adı verilen lineer cebir işlemleridir. Daha yüksek performans elde edilebilmesi için testlerde 3 farklı BLAS versiyonu denenmiş olup CUDA işlemcileri için optimize edilmiş olan CuBLAS’ın kullanılmasına karar verilmiştir. Tablo 3.3.’e göre BLAS versiyonunun DBN performansı için sağladığı avantajlar görülebilir.

3.2.2.5 Ön-Eğitim Parametrelerinin Seçilmesi

Ön-eğitim aşamasında SGD için gerekli adım boyutu seçilirken maliyet fonksiyonunun iterasyonlar boyunca azaldığı kontrol edilmelidir. Maliyet fonksiyonunu minimize edecek en büyük adım boyutu seçilirse her iterasyonda fonksiyonun minimumuna daha hızlı yakınsaması sağlanabilir. Şekil 3.6’de görülebileceği gibi maliyet fonksiyonun ön-eğitim aşamasında uygun bir şekilde optimize edilmesi için ön-eğitim adım boyutu 0.001 seçilmiştir.



Şekil 3.6: Ön-eğitim maliyet fonksiyonunun iterasyonlar boyunca değişimi

Ön-eğitim boyunca DBN parametreleri gözlemlenerek RBM'lerin aşırı öğrenme durumu engellenebilir. Ayrıca kullanılan özniteliklerinin hangisinin daha çok önem taşıdığı da gözlemlenebilir. Şekil 3.7'de iterasyonlar boyunca ağırlık parametreleri birer piksel ile aşağıdaki gibi görselleştirilerek düşey eksende giriş öznitelikleri, yatay eksen işlemci birimi için W ağırlıkları Şekil 3.7'deki gibi çizilmiştir. 10. İterasyondan sonra görülmeye başlanan aşırı öğrenme bu noktada durulması gerektiğinin göstergesidir.

(a) Ön-eğitim aşaması öncesi rastgele olarak başlatılan ağırlıklar

(b) 1. Ön-eğitim iterasyonundan sonra ağırlıklar

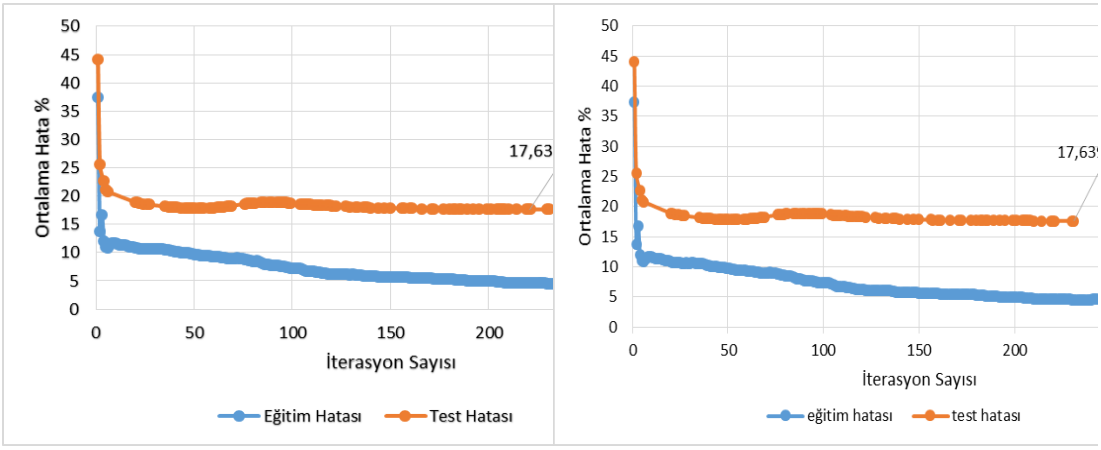
(c) 10. Ön-eğitim iterasyonu sonucu ağırlıklar

(d) 50. Ön-eğitim iterasyonu sonucu ağırlıklar

Şekil 3.7: Ön-eğitim aşaması boyunca W ağırlık matrisinin değişimi

3.2.2.6 Gibbs Örneklemesi

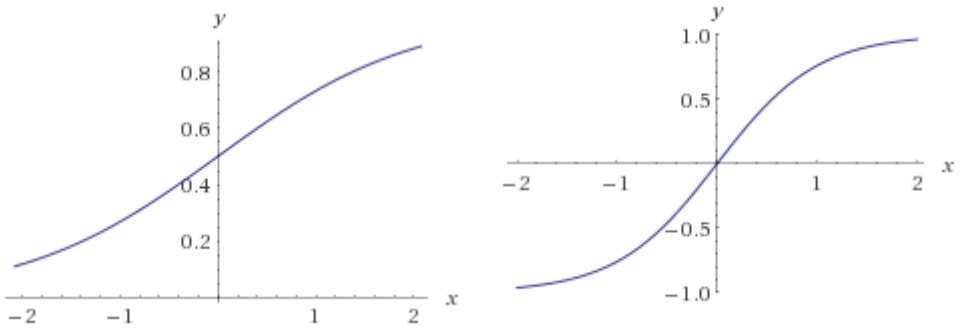
2.1.1.1 de anlatıldığı gibi Gibbs örneklemesi adımı önemli bir parametre olmasına karşın yapılan testlerde Şekil 3.8'de görüldüğü gibi fazla bir fark yaratmadığı görülmüştür. Bu sebeple testler sırasında tek adım Gibbs örneklemesi kullanılmıştır.



Şekil 3.8: Solda (a)tek adım Gibbs örneklemesi, sağda (b) 3 adım Gibbs örneklemesi

3.2.2.7 Aktivasyon Fonksiyonları

DBN işlemci birimlerin kullanılan sigmoid aktivasyon fonksiyonu yerine başka aktivasyon fonksiyonlarının kullanılması literatürde tavsiye edilen bir yöntemdir [14]. Şekil 3.9.b'de gösterilen tanh(x) hiperbolik fonksiyonun, sigmoid aktivasyon fonksiyonu yerine kullanılması testler sırasında denemiştir.



Şekil 3.9: Solda (a) sigmoid(x), sağda (b) tanh(x)

Tanh hiperbolik aktivasyon fonksiyonun kullanılması maliyet fonksiyonun aynı iterasyon sayısında daha fazla yakınsamasına sebep olduysa da işlem gücü olarak daha maliyetli olduğunda eğitim süresini olumsuz etkilemiştir. Bu sebepten dolayı aktivasyon olarak testler sırasında sigmoid fonksiyonu kullanılmıştır.

3.2.3 Başarım sonuçları

Tablo 3.4'de görülebileceği gibi, 3.1.1'de açıklanan test senaryolarından ilkine göre yapılan testler sonucunda, örnek sayısının artması, sonucu dramatik bir biçimde etkilememektedir. Bu demektir ki, oluşturulan DBN modeli, farklı verilerdeki konuşmacı farklılıklarından çok fazla etkilenmemiştir.

Literatürde kullanılan hata raporlama türlerinden hassasiyet hatası belirtilen sınıfa göre doğru tahmin edilen örneklerin, tüm örneklerden belirlenen sınıfa ait olduğu tahmin edilen örneklere oranıyla elde edilir. Hatırlama oranı ise yine belirtilen sınıfa ait doğru tahmin edilen örneklerin, o sınıfa ait tüm örneklere oranıyla hesaplanır.

Tablo 3.4: İlk test senaryosunda elde edilen başarımın değişimi

eğitim sırasında kullanılan örnek sayısı	**Ortalama hata**	**Hassasiyet (%)**	**Hatırlama (%)**	**f1-skoru (%)**	**İterasyon sayısı**	**zaman**
100	37,76%	NSL: 57 SL : 67	NSL: 61 SL : 62	NSL: 59 SL : 64	195	8'43"
200	34,78%	NSL: 54 SL : 71	NSL: 76 SL : 47	NSL: 63 SL : 57	72	6'12"
1000	34,78%	NSL: 60 SL : 70	NSL: 65 SL : 65	NSL: 62 SL : 67	15	2'27"
2000	34,98%	NSL: 59 SL : 71	NSL: 68 SL : 62	NSL: 64 SL : 66	250	12'02"
3000	34,84%	NSL: 60 SL : 70	NSL: 67 SL : 64	NSL: 63 SL : 67	250	13'55"
8000	34,50%	NSL: 62 SL : 68	NSL: 60 SL : 70	NSL: 61 SL : 69	250	15'32"
20000	29,53%	NSL: 73 SL : 69	NSL: 54 SL : 84	NSL: 62 SL : 76	250	22'44"
47955	30,54%	NSL: 62 SL : 77	NSL: 77 SL : 62	NSL: 69 SL : 69	143	32'13"
47955(PCA yok)	31,93%	NSL: 59 SL : 78	NSL: 77 SL : 61	NSL: 66 SL : 68	320	71'58"

Başarım ölçütü olarak kullanılan F1 ölçütü ise hatırlama ve hassasiyet oranlarının dengesini göz önüne alarak literatürde kullanılan (2.27)ye göre hesaplanmıştır.

$$F1 = 2.\frac{Hatırlama.Hassasiyet}{Hatırlama + Hassasiyet} \tag{2.27}$$

İkinci senaryo olan eğitim sırasında eğitim ve doğrulama kümelerinin, test sırasında ise test kümesinin kullanıldığı senaryoda Tablo 3.5 den görülebileceği üzere daha az veri ile aynı hatta daha iyi başarım oranlarına daha hızlı ulaşılabileceği görülmektedir. İlk senaryoya göre eğitim verisinin genişletilmesinin sonuçlara yaptığı pozitif etki gözlenmiştir.

Tablo 3.5: İkinci test senaryosu: eğitim+doğrulama-test

eğitim sırasında kullanılan örnek sayısı	**Ortalama hata**	**Hassasiyet (%)**	**Hatırlama (%)**	**f1-skoru (%)**	**İterasyon sayısı**	**zaman**
100	31,93%	NSL: 85 SL : 67	NSL: 51 SL : 92	NSL: 64 SL : 78	126	6'56''
200	26,86%	NSL: 90 SL : 74	NSL: 63 SL : 94	NSL: 74 SL : 82	133	8'13''
1000	18,95%	NSL: 84 SL : 79	NSL: 74 SL : 87	NSL: 79 SL : 83	233	15'34''
2000	16,54%	NSL: 90 SL : 78	NSL: 71 SL : 92	NSL: 79 SL : 85	94	8'54''
3000	14,44%	NSL: 93 SL : 81	NSL: 76 SL : 95	NSL: 83 SL : 87	66	7'55''
8000	17,52%	NSL: 90 SL : 77	NSL: 70 SL : 93	NSL: 79 SL : 84	28	8'38''
20000	19,88%	NSL: 87 SL : 75	NSL: 68 SL : 90	NSL: 62 SL : 82	38	14'12''
83943	20,72%	NSL: 80 SL : 79	NSL: 76 SL : 82	NSL: 69 SL : 79	109	54'33''
83943(PCA yok)	19,88%	NSL: 94 SL : 67	NSL: 75 SL : 91	NSL: 82 SL : 77	132	68'22''

Çalışma sonucunda elde edilen başarım sonuçları literatürle karşılaştırıldığında, duygu sınıflandırılma problemine derinlikli öğrenmenin bir çözüm oluşturduğu kanaatine varılabilir. 2011'de gerçekleşen yarışmanın sonuçları ile yapılan çalışmanın sonuçlarının karşılaştırılması Tablo 3.6'de görülebilir.

Tablo 3.6: Literatürdeki sonuçlar ile tez sonuçları

	Kullanılan Yöntem	Eğitim-Doğrulama		Eğitim+doğrulama-Test	
		SL	NSL	SL	NSL
Tez Çalışması	DBN(Derinlikli İnanç Ağı)	%62	%77	%95	%76
Interspeech 2011 en yüksek başarım[15]	Hibrid	%60.3	%75.7	%64.2	%79.1
İTÜ MSPR Lab. [2]	SVM	%89.1	%97.2	%79.9	%80.1

4. SONUÇLAR

Derinlikli ağlar her ne kadar başarılı sonuçlar verseler de sahip oldukları parametre çeşitleri sebebiyle öğrenilmesi ve probleme uygulanması karmaşıklık yaratmaktadır. Ayrıca parametre seçimlerinin çoğunun analitik çözümü olmaması en iyi sonucun elde edilip elde edilmediğini belirsiz kılmakla birlikte her zaman gelişime açık olması mühendislik açısından önemli bir avantajdır. Gelişen bilgisayar ve çip teknolojileriyle birlikte bu gelişime açık olma durumunun sonuçları daha da geliştireceği öngörülebilir.

Her ne kadar literatüre bakınca başarılı sonuçlar elde edildiği görülse de sonuçlar saniyelik öznitelik vektörleri olarak değil de konuşma kaydı bazında raporlanarak daha yüksek başarım oranları elde edilebilir. Ayrıca çalışma sırasınca kullanılan ağ gelişime açıktır. Her katman için ayrı adım büyüklükleri seçilip SGD katman bazında daha verimli hale getirilebilir. Üstelik adım büyüklükleri iterasyonlar boyunca mini-yığında bulunan verilere göre seçilerek maliyet fonksiyonun minimumuna daha doğru yakınsamalar elde edilebileceği gibi girişe uygulanan veriye göre daha uygun başlangıç ağırlıkları seçilerek başarım sonuçları daha da geliştirilebilir.

Derinlikli ağlarda ise son yıllarda yapılan çalışmalar sonucu ön-eğitim aşamasının sağlamış olduğu aşırı-öğrenme ve kaybolan gradyan problemleri doğrultucu aktivasyon birimleri ve konvolüsyonel ağlar kullanılarak çözülebilmektedir. Böylelikle ön-eğitim aşaması kaldırılarak daha başarılı sonuçlar elde edilebileceği gösterilmiştir[16].

KAYNAKLAR

[1] **Turing, A.**, 1950. *Computing machinery and intelligence*. Mind: 433-460.

[2] **Gunsel, B., Sezgin, C. and Krajewski, J.**, 2013. Sleepiness detection from speech by perceptual features, in Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference, Vancouver, BC, pp. 788-792.

[3] **Hinton, G. E.**, 2012. *Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups*. Signal Processing Magazine, IEEE 29.6: 82-97.

[4] **Hinton, G. E., Osindero, S., and Teh, Y.**, 2006. *A Fast Learning Algorithm For Deep Belief Nets*, Neural Computation, vol. 18.

[5] **Hinton, G. E. and Salakhutdinov, R. R.,** July 2006. *Reducing the Dimensionality of Data with Neural Networks*, Science Magazine**,** vol. 313, no. 5786, pp. 504-507.

[6] **Krizhevsky, A. and Hinton, G. E.**, 2009. *Learning multiple layers of features from tiny images*. Computer Science Department, University of Toronto, Tech. Rep 1.4: 7.

[7] **Fischer, A. and Christian, I.**, 2014. *Training restricted Boltzmann machines: An introduction. Pattern Recognition.* 47.1: 25-39.

[8] **Hinton, G. E.**, 2002. *Training Products of Experts by Minimizing Contrastive Divergence*, Neural Computation,no. 14, pp. 1771-1800.

[9] **Schuller, B., Batliner, A., Steidl, S., Schiel, F. and Krajewski, J.**, 2011. *Speaker State Challenge,* in *Proc. INTERSPEECH 2011*, Florence, Italy.

[10] **Gunsel, B, Sezgin, C, and Krajewski, J,** 2015. *Medium term speaker state detection by perceptually masked spectral features*, Speech Communication 67:26-41.

[11] **Schuller, B.**, 2013. *Medium-Term Speaker States - A Review on Intoxication, Sleepiness and the First Challenge, in Computer Speech and Language*, Special Issue on Broadening the View on Speaker Analysis, pp. 48-55.

[12] **Bastien, F., Lamblin, P., Pascanu, R., Bergstra, J., Goodfellow, I., Bergeron, A., Bouchard, N., Warde-Farley, D. and Bengio, Y.**, 2012. *Theano: new features and speed improvements*. *NIPS 2012 deep learning workshop*.

[13] **Bergstra, J., Breuleux, O., Bastien, F., Lamblin, P., Pascanu, R., Desjardins, G., Turian, J., Warde-Farley, D. and Bengio, Y.**, 2010. *Theano: A CPU and GPU Math Expression Compiler*. Proceedings of the Python for Scientific Computing Conference (SciPy). June 30 – July 3, Austin, TX.

[14] **LeCun, Y. A.**, 2012. *Efficient backprop*. Neural networks: Tricks of the trade. Springer Berlin Heidelberg. 9-48.

[15] **Dong-Yan, H., Zhengchen, Z., and Shuzhi, S. G.**, March 2014. *Speaker state classification based on fusion of asymmetric simple partial least squares (SIMPLS) and support vector machines*, Computer Speech & Language, vol. 28, no. 2, pp. 392-419.

[16] **Zeiler, Matthew, D.**, 2013. *On rectified linear units for speech processing. Acoustics*, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on. IEEE.

ÖZGEÇMİŞ

Ahmet Haluk Açarçiçek

Doğum Tarihi: 27.03.1992

Yeri: Bursa

e-posta: hlkcrcck@gmail.com

telefon: 0507 883 37 79

Eğitim Bilgileri:

Lisans: İstanbul Teknik Üniversitesi Elektronik ve Haberleşme Müh. (2015)

Lise: Emine Örnek Koleji (2010)

Stajlar:

Turkcell

Türk Telekom

Çemtaş

Yetkinlikler:

C, C++, Python, Matlab, Javascript, HTML5,CSS, Google App Engine.

Sertifikalar:

Game Theory(Oyun Teorisi), Cryptography(Kriptoloji), Design: Creation of Artifacts in Society(Ürün Tasarımı), CCNA(1 kur)