

**ADNAN MENDERES ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
MATEMATİK ANABİLİM DALI
2013-YL-057**

**ZEKİ ÖĞRETİM SİSTEMLERİNDE
ÖĞRETİM KAVRAMLARININ OTOMATİK TESPİTİ**

Özcan YILDIRIM

**Tez Danışmanı:
Yrd. Doç. Dr. Korhan GÜNEL**

AYDIN

ADNAN MENDERES ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ MÜDÜRLÜĞÜNE
AYDIN

Matematik Anabilim Dalı Yüksek Lisans Programı öğrencisi Özcan YILDIRIM tarafından hazırlanan Zeki Öğretim Sistemlerinde Öğretim Kavramlarının Otomatik Tespiti başlıklı tez, 03.09.2013 tarihinde yapılan savunma sonucunda aşağıda isimleri bulunan jüri üyelerince kabul edilmiştir.

	Ünvanı Adı Soyadı	Kurumu	İmzası
Başkan :	Yrd. Doç. Dr. Korhan GÜNEL	ADÜ Fen-Ed. Fakültesi	
Üye :	Yrd. Dr. Mehmet KURT	İzmir Üni. Fen-Ed. Fakültesi	
Üye :	Yrd. Doç. Dr. Rıfat AŞLIYAN	ADÜ Fen-Ed. Fakültesi	

Jüri üyeleri tarafından kabul edilen bu Yüksek Lisans tezi, Enstitü Yönetim Kurulunun sayılı kararıyla .../.../2013 tarihinde onaylanmıştır.

Prof. Dr. Cengiz ÖZARSLAN
Enstitü Müdürü

**ADNAN MENDERES ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ MÜDÜRLÜĞÜNE
AYDIN**

Bu tezde sunulan tüm bilgi ve sonuçların, bilimsel yöntemlerle yürütülen gerçek deney ve gözlemler çerçevesinde tarafımdan elde edildiğini, çalışmada bana ait olmayan tüm veri, düşünce, sonuç ve bilgilere bilimsel etik kuralların gereği olarak eksiksiz şekilde uygun atıf yaptığımı ve kaynak göstererek belirttiğimi beyan ederim.

03.09.2013

Özcan YILDIRIM

ÖZET

ZEKİ ÖĞRETİM SİSTEMLERİNDE ÖĞRETİM KAVRAMLARININ OTOMATİK TESPİTİ

Özcan YILDIRIM

Yüksek Lisans Tezi, Matematik Anabilim Dalı
Tez Danışmanı: Yrd. Doç. Dr. Korhan GÜNEL
2013, 47 sayfa

Günümüzde bilgisayar teknolojisinin hızla büyümesine paralel olarak, eğitim destek sistemlerine yapay zeka unsurlarının entegre edilmesi konusunda yapılan araştırmalar gittikçe derinleşmiştir. Eğitim teknolojilerinde yapay zeka kullanılması çalışmaları, öğrencinin kendisine sunulan konuyla ilgili neyi öğrenip öğrenmediğinin tespiti, algı kapasitesinin ölçülmesi, öğrenme hızının ve öğrenme stilinin belirlenmesi gibi konunun uzmanı bir kişi tarafından bile zorlukla karar verilen problemleriyle ilgidir. Yapılan bu çalışmaların genel amacı, klasik sınıf içi eğitimi referans alarak eğitim destek sistemlerini daha verimli hale getirmektir.

Bu tezde, klasik sınıf içi eğitime destek sağlayan zeki öğretim sistemlerinde bir eğitim içeriğindeki öğrenim kavramlarının nasıl tespit edileceği problemi incelenmiştir. Öğrenim kavramının doğru olarak belirlenmesi zeki öğretim sistemlerinin işlevini yerine getirebilmesi için kritik bir sorundur. Böylece eğitim içeriklerinin sınıflandırılabilmesi, öğrenciye ne öğretilmesi gerektiği ya da önerilen kaynağın gerçekte konu ile ilgili olup olmadığının belirlenmesi sağlanabilir.

Tezde, “Soyut Cebir”, “Analiz” ve “Bilgisayar Bilimleri” başlıkları altında toplanan üç farklı konuda eğitim içerikleri kullanılarak derlemler oluşturulmuştur. Derlemler ve dokümanlar ön işlemlerden geçirilip her bir sözcük dizisi için öznitelik vektörleri oluşturulmuştur. Öznitelik vektörlerinin bileşenleri oluşturulurken terim frekansı ve ters doküman frekansı gibi istatistiksel veriler kullanılmıştır. Öznitelik vektörü oluşturan her bir sözcük dizilimi, konuyla ilgili öğretim kavramı olmaya adaydır. Çalışmada aday kavramlar kümesinden öğretim kavramlarını seçme işlemi için, ağırlıklı oylama kullanan k -En Yakın Komşu algoritması ve Levenberg-Marquardt optimizasyonu ile Çok Katmanlı Algılayıcı yöntemleri uygulanmıştır. Yöntem başarısını sorgulamak için ise duyarlılık, kesinlik ve f-ölçüsü değerleri kullanılmıştır.

Anahtar Sözcükler

Zeki Öğretim Sistemleri, Eğitim Teknolojileri, Yapay Zeka, Makine Öğrenmesi

ABSTRACT

Automatically Detection of Learning Concepts in Intelligent Tutoring System

Özcan YILDIRIM

M.Sc. Thesis, Department of Mathematics
Supervisor: Assist. Prof. Dr. Korhan GÜNEL
2013, 47 pages

Nowadays, researches regarding the integration of intelligent issues on educational support systems has increased in parallel with the rapid growth of computer technology. The researches are associated with problems such as detecting what to teach to students, learning speed, learning style and perception capacity of students. The overall objective of these studies is to develop more efficient educational support systems promoted by traditional classroom training. The solving these issues are difficult, controversial, time consuming and highly non-trivial processes, even for an expert in this field. This thesis analyzes the problem how to determine the learning concepts from an educational material with the intelligent tutoring systems. The extraction of learning concepts from an educational contents is a critical problem. Thus, the problem about the classification of the educational contents can be overcome. In addition, the question of “what to teach to students” can be answered, and the issues of “whether the recommended educational material is related to the learning concepts actually required to teach or not” can be handled by an educational support system.

In this study, firstly, three different corpora have been constructed. The corpora correspond to the “Algebra”, “Calculus” and “Computer Science” as learning domains. The educational contents as documents and corpora has been pre-processed. In the next step, the feature vectors have been generated for each word sequence in the documents. The feature vectors corresponds some statistical data such as term frequency and inverse document frequency. Each word string referenced by a feature vector is candidate for learning concepts within the learning domain. In this study, two different methods have been applied: k-nearest neighbour with majority voting algorithm and Multi-Layer Perceptron (MLP) with Levenberg-Marquardt optimization. Recall, precision and f-measure scores have been used for measuring the system performance.

Key Words

Intelligent Tutoring Systems, Educational Technology, Artificial Intelligence, Machine Learning

ÖNSÖZ

Yüksek lisans öğrenim süresince beni hep destekleyen, umut aşılayan, yol gösteren, bilgisini ve deneyimini her zaman paylaşan örnek aldığım danışman hocam değerli bilim insanı Yrd. Doc. Dr. Korhan Günel'e ve yardımlarını esirgemeyen Yrd. Doc. Dr. Rıfat Aşlıyan hocama teşekkür ederim.

Maddi manevi destekleriyle hep yanımda olan ve ayakta durmamı sağlayan Anneme, Babama, Kardeşim Bircan'a ve Fikriye Gümüş'e en içten teşekkürlerimi sunarım.

Özcan YILDIRIM

İÇİNDEKİLER

KABUL ONAY SAYFASI	iii
BİLİMSEL ETİK BİLDİRİM SAYFASI	v
ÖZET	vii
ABSTRACT	ix
ÖNSÖZ	xi
1. GİRİŞ	1
2. MATERYAL VE METOT	5
2.1. Öznitelik Vektörlerinin Oluşturulması	5
2.2. K-En Yakın Komşu Algoritması	8
2.2.1. Ağırlıklı oylama	9
2.3. Çok Katmanlı Yapay Sinir Ağı Ve Levenberg-Marquardt Optimizasyonu	11
2.3.1. Çok Katmanlı Yapay Sinir Ağı	12
2.3.2. Levenberg-Marquardt Optimizasyonu	18
3. DENEYSEL SONUÇLAR	23
4. SONUÇ	27
KAYNAKLAR	29
ÖZGEÇMİŞ	33

1. GİRİŞ

Dünya nüfusundaki artış ve taleplerin farklılaşması çoğu sistemi etkilediği gibi devletlerin ve çeşitli kurumların eğitim sistemlerine de farklı bir bakış açısı getirmesine sebep olmuştur. Bu talepleri karşılayacak eğitim gücünün sağlanılmasında çeşitli sıkıntılar doğmuştur. Bilginin ve eğitimin artan nüfus ve kişilerin farklı talepleri doğrultusunda yenilenmesi ve şekil değiştirmesi, eğitim sistemlerinin bilgi aktarımı konusunda uzaktan eğitim sistemleri, sınıf içinde kullanılan çeşitli eğitim ekipmanları ya da internet üzerinden erişilebilen eğitim içerikleri gibi yeni yöntemler geliştirmesi sağlamıştır.

Fakat günümüzde öğrenciler ya da var olan bilgiyi edinmek isteyen kişiler artan bilgi miktarı ve hızla değişen teknolojiye entegre edilen bilgi edinme yöntemleri altında ezilmektedirler ve yeterince etkin bir şekilde yararlanamamaktadırlar. Bu sebepten bilgiyi daha rahat, daha etkin ve hızlı bir biçimde elde etmek için farklı eğitim teknolojileri geliştirilmesi zorunlu bir ihtiyaç haline gelmiştir. Son yıllarda hükümetlerin ve eğitim kurumlarının eğitim teknolojileri politikalarını geliştirmeye başlamalarıyla, eğitim teknolojilerine yapay zeka tekniklerini entegre etmeye yönelik bilimsel araştırmaların sayısında da artış gözlenmektedir [1, 4]. Buna paralel olarak, literatürde eğitim teknolojilerinin klasik sınıf içi eğitime destek sağlayarak öğrenci başarısında olumlu yönde artış sağladığına dair araştırmalara rastlamak mümkündür [5, 7].

Bu bakımdan eğitim destek sistemlerinin bir öğretmenin yerini alamasa dahi, öğretmenin davranışlarını taklit edebilme yeteneğine sahip olması gerekir. Bu taklit yeterli seviyede olmasada, eğitim destek sistemleri öğrenciye ne öğretmesi gerektiğini tespit edebilmelidir. Geliştirilecek sistemler, konuyla ilgili kavramları öğrencilerin öğrenim stilleri ve öğrenme hızlarına göre öğretmelidir. Örneğin

bir öğretmenin sınıf içinde öğrencilerin seviyelerine ya da kavrama şekillerine göre öğretim yöntemlerini belirlemesi sürecini eğitim destek sistemleri de taklit edelidir. Ancak, çoğu zaman belli bir konudaki öğrenim kavramlarının konunun uzmanı bir kişi tarafından bile tespiti zor, karmaşık ve zaman alıcı süreçtir [8]. Özellikle çok fazla sayıda farklı dokümana sahip olan bir sistemde, öğrenciye önerilecek dokümanın gerçektende öğretilmesi gereken kavram ile ilgili olduğunun saptanması sıkıntılı bir süreçtir.

Bir dokümandaki kavram veya terimleri belirlemeye yönelik çalışmalar oldukça eskiye dayanır. Bu konuda pek çok bilimsel çalışmaya ulaşmak mümkündür [9,15]. Ancak bu yaklaşımların eğitim teknolojileri içinde kullanılmaya başlaması oldukça yenidir. Daille, dil filtreleme ve istatistiksel metotları birarada kullanarak herhangi bir derlem içindeki kavramları tespit etmeye yönelik bir çalışma gerçekleştirmiştir [16]. Frantzi ve Arkadaşları, *C/NC* değerlerini kullanarak birden fazla sözcükten oluşan terimleri otomatik olarak tespit etmeye çalışmışlardır [17]. Cimiano ve Völker, Olasılıksal Ontoloji Modeli ile özel seçilmiş bir alanda ontolojik bir yapı oluşturabilmek için terimleri otomatik tespit edecek bir araç sunmuşlardır [18]. Günel ve Aşlıyan istatistiksel dil modeli ve kavramsal harita modelini birlikte kullanarak verilen bir eğitim içeriğindeki öğrenim kavramları kümesini minimize etmeye çalışmışlardır [21]. Qasim ve arkadaşları benzerlik yayılım algoritması kullanarak alan kavramlarını çıkarmaya yönelik bir metot sunmuşlardır [8].

Bu çalışmanın temel amacı, eğitim destek sistemleri ile öğrenciye sunulan eğitim içeriklerinden öğrenim kavramlarını otomatik olarak tespit etmektir. Böylece eğitim içerikleri sistem tarafından anlamlı hale getirilmiş olur ve sistem öğrenciye ne öğretilmeli sorusuna yanıt verebilir. Aynı zamanda doküman sınıflandırması kolaylıkla gerçekleştirilebilir ve konu ile ilgisi olmayan dokümanlar elenebilir. Bu amaçla çalışmada yapay zeka yöntemlerinde iyi bilinen gözetimli öğrenme algoritmalar kullanılmıştır.

Çalışmada öğretim kavramlarını tespit edebilmek için öncelikle üç farklı konuda derlem oluşturulmuştur. Sistem öncelikle verilen bir eğitim içeriğinden aday kavramları tespit eder ve her bir aday kavramın öznitelik vektörlerini oluşturur. Öznitelik vektörleri, dokümanı temsil edebilecek öğelerden oluşmalıdır. Normalize edilen ve çeşitli istatistiksel yöntemler kullanılarak çeşitli dönüşümler uygulanan öznitelik vektörleri çalışmada sunulan modeller için girdi vektörü olarak kabul edilir. Böylece farklı dokümanlar için farklı ayırt edici öğrenim kavramları tespiti yapılmış olur ve bir aday kavramın gerçekten birer öğrenim kavramı olup olmadığına sistem karar verir.

Çalışmanın geri kalan kısmı şu şekilde organize edilmiştir: 2. Bölümle öncelikle üç farklı konu başlığı için oluşturulan derlemlerin nasıl elde edildiği anlatılmıştır ve bu derlemler hakkında bazı istatistiksel bilgiler verilmiştir. Daha sonra verilen bir eğitim içeriğinden aday öğrenim kavramlarının nasıl çıkarıldığı ve Her bir aday kavram için öznitelik vektörlerinin nasıl oluşturulduğundan bahsedilmiştir. Takip eden kısımda aday kavramlar arasından öğrenim kavramı olanları tespit etmek için kullanılan gözetimli öğrenme metotlarından bahsedilmiştir. 3. kısımda deneysel sonuçlar verilmiş ve tartışılmıştır.

2. MATERYAL VE METOT

2.1. Öznitelik Vektörlerinin Oluşturulması

Bu çalışmada üç farklı derlem ve her bir derlem için o derlem konusu ile ilgili yedi farklı döküman kullanılmıştır. Bu derlem konuları "Soyut Cebir, "Analiz" ve Yapay Sinir Ağları" dır.Bu derlemlerden "Soyut Cebir, "Analiz" ve Yapay Sinir Ağları için sırasıyla 803490, 1777067, 866490 tane kelime ve 92870, 241315, 122027 adet cümle elde edilmiştir.

Yapılan çalışmanın genel amacı, varolan dökümanların sınıflandırılması olduğundan, sınıflandırmanın temel taşı öznitelik vektörlerinin, yani dökümanı temsil edecek olan ayırt edici özelliklerin, belirlenmesi gerekmektedir. Bu sebepten belirlenen ayırt edici özellikleri elde etmek amacıyla, dökümanlar üzerinde ön işlem yapılmıştır. Dökümanlar konularına göre birleştirilip her bir konunun derlemi elde edilmiştir. Ön işlem aşamasında derlemlerden noktalama işaretleri, sayılar ve özel semboller çıkartılmıştır. İki sözcük arasında sadece bir boşluk bırakılıp bütün sözcükle küçük harflere dönüştürülmüştür. Derlemlerdeki durak sözcükleri yani cümlelerin için sıkca kullanılan, fakat anlama çok katkısı bulunmayan "am, is, the, this..." gibi kelimeler derlemlerden çıkartılmamıştır. Çünkü bu kelimeler daha sonra öznitelik vektörünün bir girdisi olarak kullanılacaktır. Ek olarak her bir derlemden öznitelik vektörünün başka bir bileşenini hesaplamak için bütün cümleler ayıklanarak elde edilmiştir.

Daha sonraki adımda $1 \leq n \leq 5$ için derlemlerden en çok tekrar eden n -gramlar çıkartılmıştır. N -gram modeli olasılıksal dil modelidir. Bu model denklem (2.1.1)' de verildiği gibi bir w_i sözcüğü için w_i sözcüğünden önce gelen $n - 1$ tane kelimeye $w_{i-(n-1)} \dots w_{i-1}$ bakarak sıradaki kelimeyi tahmin etmeye çalışır.

$$P(w_i | w_{i-(n-1)} \dots w_{i-1}) = \frac{C(w_{i-(n-1)} \dots w_{i-1} w_i)}{C(w_{i-(n-1)} \dots w_{i-1})} \quad (2.1.1)$$

(2.1.1) denkleminde, $C(w_{i-(n-1)} \dots w_{i-1})$ ifadesi bu kelime dizisinin eğitim içeriğinde görüntülenme frekansıdır. $C(w_{i-(n-1)} \dots w_{i-1} w_i)$ ifadesi ise w_i sözcüğünün $w_{i-(n-1)} \dots w_{i-1}$ sözcük dizisinin ard arda görüntülenme sayısıdır.

Çalışmada denklem (2.1.1) kullanılarak derlemede en çok tekrar eden ilk 500 n -gramlar elde edilmiştir. Bu n -gramlar kümesine G_1 denilecektir.

Bir eğitim içeriğinin hangi öğrenim kavramına ilişkili olduğunu belirlemek için derlemlere uygulanan ön işlem aşaması eğitim içerikleri içinde uygulanır. Daha sonra dokümanın en çok tekrar eden n -gramları belirlenip ve bu n -gramlar G_2 kümesi olarak adlandırılmıştır. G_2 kümesinin elemanlarının her biri dokümanın aday öğrenim kavramları olarak sayılır. Bu aşamada, oluşturulan G_2 kümesinin elemanlarının gerçekten öğrenim kavramı olup olmadığı uzmana danışılarak belirlenmiştir. Uzman görüşleri, çalışmada gözetimli öğrenim metotları kullanıldığı için sistemin eğitimi aşamasında kullanılacaktır.

Daha sonra dokümandan elde edilen G_2 kümesi içindeki her bir eleman için öznitelik vektörleri oluşturulacaktır. Bu öznitelik vektörlerinin ilk ögesi terim frekansı-ters döküman frekansı ($tf \times idf$) dır. $tf \times idf$ değeri iki farklı hesaplamadan oluşmaktadır. tf terim frekansıdır, terimin doküman içinde tekrar sayısıdır. idf ise toplam döküman sayısının ilgili terimi içeren doküman sayısına oranıdır. İlgili terimin ($tf \times idf$) değeri (2.1.2) denkleminde verildiği gibi hesaplanır.

$$tf \times idf(t, d, D) = tf(t, d) \times \log \left(\frac{|D|}{1 + |d \in D : t \in d|} \right) \quad (2.1.2)$$

(2.1.2) denkleminde $|D|$ toplam doküman sayısıdır. $tf \times idf$, d dökümanından çıkartılmış G_2 kümesindeki terimler için bir frekans yani sıklık göstergesidir.

Öznitelik vektörünün ikinci ögesi ise doküman içerisindeki toplam cümle sayısının ilgili terimi içeren cümle sayısına oranıdır. Bu ögenin kullanılması ile ilgili terimin doküman içerisindeki dağılımı kabaca hesaplanmaktadır. Bunu anlamak içinse dokümanda yer alan cümlelerin toplam sayısına ve bu cümlelerin kaçında ilgili

terimin geçtiğine denklem (2.1.3)' de görüldüğü gibi bakılır.

$$r_1(t, d) = \log \left(\frac{|s|}{1 + |s \in d : t \in s|} \right) \quad (2.1.3)$$

Denklem (2.1.3)' de $|s|$ d dokümanındaki toplam cümle sayısıdır ve $t \in G_2$ dir.

Öznitelik vektörümüzün diğer bir ögesi ise bir ilişki matrisidir. Bu matrisi $x \in G_1$ ve $t \in G_2$ arasındaki ilişki ile belirlenir. Öznitelik vektörününün bu ögesi, G_1 kümesinin terimleri ile ilgili dökümanın G_2 kümesindeki terimlerinin derlemde kaç cümlede beraber geçtiği araştırır. Bu hesaplama denklem (2.1.4) ve verilir.

$$r_2(t, d) = \sum_{x \in G_1} M(t, x), t \in G_2 \quad (2.1.4)$$

Denklem (2.1.4) M ilişki matrisi denklem (2.1.5)' deki gibi hesaplanır.

$$M(t, x) = \begin{cases} 1 & \text{t ve x aynı cümlenin içinde iseler} \\ 0 & \text{aksi takdirde} \end{cases} \quad (2.1.5)$$

Özellik vektörünün en son ögesi ise denklem (2.1.6) görüldüğü gibi G_2 kümesindeki ilgili n -gram terimini oluşturan sözcük sayısının, G_2 kümesi içindeki terimlerin içerdikleri sözcük sayılarının maksimumuna oranıdır.

$$r_3(t) = \frac{|t|}{\max(|t'| : t' \in G_2)}, t \in G_2 \quad (2.1.6)$$

(2.1.6) denkleminde kullanılan $l.l$ operatörü n -gramdaki kelime sayısını verir.

Öznitelik vektörünün bütün ögeleri hesaplandıktan sonra öznitelik vektörünün ögelerine normalizasyon yapılır ve bu değerler $[0,1]$ aralığında olması sağlanır. Daha sonra ise öznitelik vektörüne Temel Bileşenler Analizi (PCA) uygulanır. PCA çok değişkenli bir analiz yöntemidir. Bu yöntem $p < q$ olmak üzere, bağlılık gösteren q adet değişkeni ortogonal, doğrusal ve birbirinden bağımsız p adet değişkene çevirir. Bu çalışmada PCA uygulanmasının genel nedeni ise değişken sayısının çok olmasıdır. Aynı zamanda öznitelik vektörde kullanılan ögelerin

birbiriyle ilişkisi bulunduğundan öğeler metotlar için farklı ayırt edici özellik olmaktan çıkmaktadır. Bu durum sistemin eğitimde olumsuz sonuç vermektedir. Bu sorun PCA ile aşılmaktadır.

2.2. K-En Yakın Komşu Algoritması

k -En Yakın komşu Algoritması (k -nn) sınıflandırma ve kümeleme alanlarında etkin ve yaygın bir şekilde kullanılan, algoritmik olarak basit bir metottur. Metot sınıflandırma ya da kümeleme işlemlerini genel olarak sadece örnek kümesini baz olarak yapar. Bu sebepten dolayı k -nn metodu örnek tabanlı bir metottur. Aynı zamanda bu yöntem parametrik olmayan bir metottur; yani örnek ve test kümesindeki veriler için herhangi bir dağılım varsayımı yapılamaz. Pratikte var olan verilerin dağılımı teorik varsayımlara her zaman uymayabilir. Bu durumda parametrik olmayan metotların kullanılması çok yarar sağlamaktadır.

K -nn metodu $n \in \mathbb{Z}^+$ boyuta kadar genişleyebilen öznitelik vektörleri arasındaki uzaklıkları hesaplayarak sınıflandırma veya kümeleme yapmaktadır. Genelde kullanılan uzaklık Euclid uzaklığıdır, fakat Minskowski, Manhattan, Hamming, Mahalanobis gibi daha farklı metriklerle uzaklık hesaplanabilir.

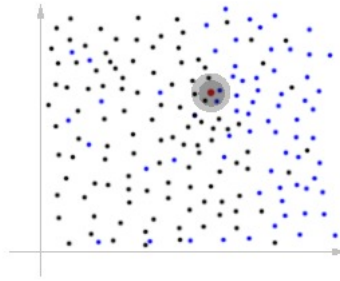
Herhangi $p = (p_1, p_2, \dots, p_n)$ ve $q = (q_1, q_2, \dots, q_n)$ gibi iki öznitelik vektörü arası öklid uzaklığı (2.2.7) eşitliğindeki gibi hesaplanmaktadır.

$$d(\mathbf{p}, \mathbf{q}) = \sqrt{\sum_{i=1}^n (p_i - q_i)^2} \quad (2.2.7)$$

k -nn ile sınıflandırma yapılırken eğitim kümesinde bulunan bütün öznitelik vektörlerinin sınıfları bilinmektedir, yani k -nn gözetimli öğrenim metotları arasındadır. Bu sayede eğitim aşaması hızlı bir biçimde geçilmektedir. Test aşamasında ise girilen yeni bir öznitelik vektörü ile diğer bütün öznitelik vektörleri arasında ki uzaklıklar hesaplanmaktadır. Eğitim kümesinde yer alan örneklerin sayısı fazlaysa ve çok boyutlu uzayda temsil ediliyorlarsa test aşaması uzun süre

alabilir. Çünkü m adet öznitelik vektörü bulunan ve her bir öznitelik vektörünün boyutu da n bir veri grubunda karmaşıklık $O(mn)$ olmaktadır. Dolayısı ile k -nn yöntemi büyük bir eğitim kümesi için çok da hızlı bir yöntem değildir.

k -nn algoritmasının tek parametresi yönteme ismini veren k parametresidir. Test aşamasında, yeni bir öznitelik vektörünün eğitim kümesinde bulunan bütün öznitelik vektörlerinin arasında uzaklık hesaplanmaktadır. Hesaplanan bu uzaklıkların en küçük k tanesi seçilir. Seçilen bu k tane öznitelik vektörünün sınıfları incelenir ve sınıf sayısı en fazla olan sınıf test için girilen öznitelik vektörünün temsil ettiği girdinin sınıfıdır. k parametresinin seçimi için genel bir kural bulunmaktadır. Fakat genelde $k = 3$ veya $k = 5$ alınmaktadır. Test aşamasının bir kaç defa uygulanması ile en uygun k değeri kolaylıkla tespit edilebilir.



Şekil 2.1. k parametre seçimi

Şekil 2.1 de k parametresi ile değişen komşu sayıları görülmektedir. Küçük çemberde komşu sayısının $k = 3$ olduğu görülüyor. Büyük çemberde ise beş komşu öznitelik vektörü görülüyor yani $k = 5$ seçilmiştir. Şekil incelendiğinde verilerin iki sınıfa ayrıldığı görülür. $k = 3$ iken kırmızı ile gösterilen verimizin mavi sınıfa, $k = 5$ iken siyah sınıfa dahi edildiği görülmektedir.

2.2.1. Ağırlıklı oylama

k -nn metodu yukarıda da bahsedildiği gibi genel olarak yeni bir öznitelik vektörü ile diğer bütün öznitelik vektörleri arasındaki uzaklıklar hesaplanır. Ardından

seçilen k parametresi ile yeni öznitelik vektörümüze en yakın k tane vektörünün seçimi yapılır. Seçilen bu k tane öznitelik vektörü içinde sınıf sayısı en fazla olan sınıf yeni öznitelik vektörünün de sınıfı olarak belirlenir.

Yeni öznitelik vektörünün hangi sınıfa dahil olacağı k parametresinin büyük oranda etkisi bulunmaktadır. Seçilen k komşu arasında sayısal olarak fazla olan sınıfın yeni vektörün de sınıfı olarak seçilmesi sorun teşkil eder. Şekil 2.2'deki birinci çember incelendiğinde $+$ ile gösterilen vektörünün iki komşusu gri sınıfa ait iken üç tanesi ise siyah sınıfa aittir. Bu durumda sayısal üstünlük olarak $+$ vektörümüzün sınıfı siyahtır. Siyahların ve grilerin dağılımları birbirine benzemektedir ve genel anlamda k -en yakın komşu yönteminin uygulanmasında bir sıkıntı yoktur. İkinci çember incelendiğinde gri noktalar sayısal üstünlük sağlamaktadır, fakat siyah noktalar $+$ vektörüne çok daha yakındır. Açıkça gri noktalar sadece sayısal olarak fazladır. $+$ vektörünün sınıfının gri olması hatalı bir sınıflandırma olacaktır. Bu sorun ya k parametresi küçülterek ya da ağırlıklı oylama yöntemini uygulanarak aşılabilir.



Şekil 2.2. Komşuluk durumları

Ağırlı oylama yönteminde, sınıf seçimi komşulukların sayısal olarak üstünlüğünü değil, yakınlık derecelerini baz almaktadır. Euclid uzaklıklarının hesaplanmasından sonra k parametresine göre komşular belirlenir ve her bir komşunun sınıfı tespit edilmeye çalışılan vektöre uzaklığı denklem 2.2.8 deki gibi değiştirilir. Aynı sınıfa dahil olan komşuların ağırlıkları toplanır ve toplam ağırlık bulunur.

$$d(p)' = \frac{1}{d(p)^2} \quad (2.2.8)$$

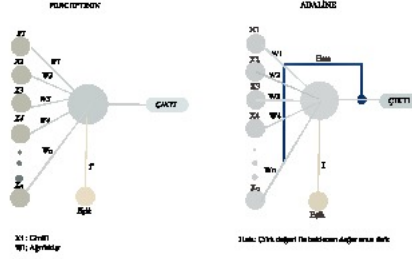
Toplam ağırlık bulunduktan sonra, yeni öznitelik vektörünün sınıfı ağırlığı en fazla olan sınıf olarak belirlenir.

2.3. Çok Katmanlı Yapay Sinir Ağı Ve Levenberg-Marquardt Optimizasyonu

Yapay sinir ağları insan beynini simüle etmeyi amaçlayan bilgisayar sistemleridir. İnsan beyninin öğrenme yoluyla edinmiş olduğu bilgileri kullanarak yeni bilgiler üretebilmesi ve bu bilgiler yardımı ile bir takım sorulara ya da sorunlara çözüm bulması, yapay sinir ağlarının insan beynini simüle etmeye çalışmasının en büyük nedenlerinden biridir. Yapay zeka yöntemlerinden biri olan yapay sinir ağları geleneksel algoritmik yapılarla çözülemeyen veya çok zahmetli çözümlere sahip olan bir çok probleme yeni bir bakış açısı getirmiştir.

Yapay sinir ağlarındaki ilk çalışmalar 1940' lı yıllarında psikolog ve nöro-fizyolog olan Warren McCulloch ve matematikçi Walter Pitts yapılmıştır. McCulloch ve Pitts basit hesaplama yapan ilk sinir hücresini oluşturdu ve herhangi bir mantıksal fonksiyonun nöron ağları ile tasarlanabileceğini göstermişlerdir. Hebb 1949 yılında ilk öğrenme kuralını oluşturdu. Hebb öğrenme kuralı, birbirine bağlı olan iki nöronun aynı yönde eğilim gösterdiğinde, bu iki nöronu birbirine bağlayan sinaps değerinin artırılması gerektiği belirtilmiştir. Rosenblatt 1958 yılında basit algılayıcı yapısını ortaya koydu. Basit algılayıcı aktivasyon fonksiyonu ve ayarlanabilir sinaptik ağlara sahip olan tek bir nörondur. Widrow ve Hoff, 1960 yılında en küçük kareler (LMS) adlı bir öğrenme yöntemi geliştirdiler ve Adaline (Adaptive Linear Neuron) nı eğitmek için kullandılar. Adaline eşik değeri yerine lineer bir aktivasyon fonksiyonu kullanır. Bunun dışında basit algılayıcı ile aralarında bir fark yoktur. Minsky and Papert 1969' lu yıllarda basit algılayıcı, adaline gibi tek katmanlı ağların XOR problemini çözemediğini ortaya

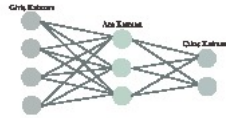
attı. XOR problemi OR yada AND gibi doğrusal ayrılabilen bir fonksiyon değildir. Rumelhart, Hinton ve Williams, 1986 yıllarda geliştirdikleri Çok Katmanlı Yapay Sinir Ağı metodu (Multilayer Perceptron, ÇKA) ile doğrusal ayrılamayan XOR probleminin çözüleceği göstermişlerdir.



Şekil 2.3. Perceptron

2.3.1. Çok Katmanlı Yapay Sinir Ağı

Günlük hayat problemlerinin bir çoğu doğrusal olmayan problemlerdir bundan dolayı ÇKA gibi doğrusal olmayan problemlere çözüm getirebilen ağ yapıları yaygın ve etkin bir biçimde kullanılmaktadır. ÇKA'nın temelleri geriye yayılım algoritmasıyla Werbos tarafından 1974 yılında atılmıştır. Fakat şu andaki hali yukarıda da bahsedildiği gibi Hinton ve Williams tarafından geliştirilmiştir. ÇKA' da giriş katmanı ve çıkış katmanı arasında en az bir ara katman bulunmaktadır.



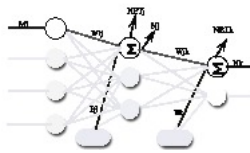
Şekil 2.4. ÇKA

Basit algılayıcı lineer ayrılabilen problemlerin çözümünde kullanılan ve bu tür fonksiyonlara çözüm getiren bir ağ yapısıydı, ancak doğrusal ayrılamayan fonksiyonlarda örneğin XOR problemine çözüm getiremiyordu. Doğrusal ayrılabilen fonksiyonlarda çıktı değerleri bulunduğu sınıfa göre bir doğru ile

ayrılabilirken doğrusal olmayanlar ise ayrılamamaktadır. Bu durumu ÇKA ile aşılmaya çalışılmış ve aktivasyon fonksiyonunda, ağ yapısında, öğrenme algoritmasında basit algılayıcıya göre büyük değişiklikler yapılmıştır.

ÇKA'nın işlem yapısı perceptrona benzemektedir fakat aktivasyon fonksiyonu olarak türevi alınabilen lineer olmayan fonksiyonlar kullanılır (lineer fonksiyonlar $y = ax + b$ gibi fonksiyonlardır lineer olmayanlar ise örneğin $y = ax^2 + bx + c$ yada $y = ax^3 + bx^2 + cx + d$ gibi fonksiyonlardır). ÇKA'de en çok kullanılan aktivasyon fonksiyonlarından bir kaç; sigmoid $y = \frac{1}{1+e^x}$, hiperbolik tanjant fonksiyonu $y = \frac{e^x - e^{-x}}{e^x + e^{-x}}$ gibi fonksiyonlardır. Bu fonksiyonlar ile girdi ve çıktı değerleri arasındaki bağlantı doğrusal olmaktan çıkmaktadır ve bu sayede doğrusal ayrılamayan problemlerin yapısına uygun çıktı değerleri elde edilebilir. Ara katmanlardaki kullanılan lineer olmayan aktivasyon fonksiyonları ÇKA'yı etkili kılan en büyük yanlarından birisidir.

ÇKA'nın öğrenme kuralı Genelleştirilmiş Delta kuralıdır. Gözetimli öğrenme kuralı olan Delta Kuralı ileri besleme ve geri yayılım safaharından oluşan bu öğrenme kuralı ileri besleme aşamasında ağın çıktı değerlerini hesaplarırken geri yayılımda ise ağırlıkların bu çıktı değerine göre değişimini sağlar. İleri yayılımda algoritma şöyle çalışmaktadır.



Şekil 2.5. ÇKA

1. m_i , i. nöronun girdisi
2. w_{ij} , i. nöronla j. nöron arasındaki ağırlık
3. net_j , j. ara katman nöronunun yan çıktısı

$$net_j = \sum(m_i w_{ij})$$

4. n_j , j. nöronun ana çıktısı, b_j j. nöronun eşik değeri

$$n_i = f(net_j) = \frac{1}{1 + \exp^{-(net_j + b_j)}}$$

5. w_{jk} j. nöron ile k. nöron arasındaki ağırlık

6. net_k k. nöronun yan çıktısı

$$net_k = \sum (n_i w_{jk})$$

7. n_k , k. nöronun ana çıktısı, b_k j. nöronun eşik değeri

$$n_k = f(net_k) = \frac{1}{1 + \exp^{-(net_k + b_k)}}$$

İleri yayılım algoritması ile çıktı değerleri (n_1, n_2, \dots) hesaplanmış durumdadır. Geri yayılım kısmı; ilk safhada üretilen çıktılarının ağırlık beklenen çıktılarını (b_1, b_2, \dots) ile karşılaştırıldığı ve hata miktarının hesaplandığı kısımdır. Bu kısımda hata miktarları hesaplanıp, bu hata miktarı yardımı ile ağırlık sahip olduğu ağırlıklar değiştirilir ve hata miktarı minimize edilmeye çalışılır. Çıktı katmanındaki k'inci nöronun hata miktarı h_k şu şekilde hesaplanır.

$$h_k = b_k - n_k \quad (2.3.9)$$

Kullanılan ağ yapısına göre çıktı katmanı bir çok nörondan oluşabilir. Bundan dolayı ağırlık toplam hatası hesaplanmalıdır. Aynı zamanda beklenen çıktı değeri ağırlık çıktı değerinden küçük olabileceğinden hata değeri negatif çıkabilir ve toplamsal olarak sağlıklı sonuçlar vermeyebilir bu sebepten çıkan hata değerinin karesi alınır. Toplam hata (2.3.10) eşitliğindeki gibi hesaplanır.

$$th = \sum_{i=1}^k (h_i)^2 \quad (2.3.10)$$

Hatanın azaltılması için ağırlıkların değiştirilmesi ve minimum hataya ulaşana kadar ağırlık değişiminin devam etmesi gerekmektedir. Ağırlık değişimi iki

aşamada yapılmaktadır. İlk olarak, çıktı katmanı ile ara katman arasında ağırlık değişimi yapılır. Daha sonra ise girdi katmanı ile ara katman arasındaki ağırlıklar güncellenir. Birden fazla ara katman kullanılmışsa o ara katmanlar arasındaki ağırlıklar yenilenir.

Herhangi bir iterasyonda ağırlıkların değişimi (2.3.11) denklemi ile yapılır.

$$\Delta w_{jk}(t) = \lambda \delta_k n_k + \alpha \Delta w_{jk}(t-1) \quad (2.3.11)$$

(2.3.11) denklemindeki kullanılan λ öğrenme katsayısıdır, ve ağırlık değişim miktarını belirler. Öğrenme katsayısındaki değişim ağırlık performansı üzerinde büyük bir etkiye sahiptir. Seçimi ise ağırlık yapısına göre değişmektedir. Genelde sıfır ile bir arasında değerler alır. Öğrenme katsayısı büyük seçildiğinde ağırlık iterasyon sayısı azalırken seçim küçüldüğünde ise iterasyon sayısı artmaktadır. Yinede bu katsayıyı büyük seçmek ağırlık hassasiyetini azaltabilir. (2.3.11) denkleminde kullanılan α momentum katsayısıdır. Momentum katsayısı ise ağırlık yerel minimum bir hataya takılıp salınım yapmasını engeller. Ana mantığı ağırlıklardaki daha önceki değişiminin bir miktarını ağırlık eklemeye dayanır. Genelde sıfır ile bir değeri arasında değerler alır. Denkleminde gösterilen δ_k değeri ise k . nöronun çıktısındaki hata miktarını göstermektedir. Ağırlık genel amacı hatayı minimize etmek olduğundan bu değeri hesaplamak için toplam hata değerinin türevinden yararlanılır.

$$\frac{\partial(th)}{\partial(w_{jk})} = \frac{\partial(th)}{\partial(n_k)} \frac{\partial(n_k)}{\partial(net_k)} \frac{\partial(net_k)}{\partial(w_{jk})}$$

Toplam hatanın türevi alınırken bağlı olduğu değişkenlere göre zincir kuralı uygulanmıştır. th değeri n_k değerine bağlı olduğundan n_k ya göre türev alınır. Zincirin $\frac{\partial(th)}{\partial(n_k)}$ kısmı elde edilmiş olur.

$$th = \frac{1}{2} \sum (b_k - n_k)^2$$

$$\frac{\partial(th)}{\partial(n_k)} = \frac{1}{2} \cdot 2(b_k - n_k)$$

$\frac{\partial(n_k)}{\partial(net_k)}$ kısmının hesaplanmasından önce . n_k değerinin

$$n_k = \frac{1}{1 + \exp^{-net}}$$

bu şekilde hesaplanıldığı daha önce gösterilmişti. n_k değeri net_k ' ya bağlı olduğundan

$$\begin{aligned} \frac{\partial(n_k)}{\partial(net_k)} &= (-1) \cdot (1 + \exp^{-net_k})^{-2} \cdot (-1) \cdot \exp^{-net_k} \\ &= \frac{\exp^{-net_k}}{(1 + \exp^{-net_k})^2} \\ &= (1 - n_k)n_k \end{aligned}$$

$\frac{\partial(n_k)}{\partial(net_k)}$ değeri bu şekilde hesaplanılmıştır. Son zincir olan $\frac{\partial(net_k)}{\partial(w_{jk})}$ değeri ise

$$net_k = \sum(w_{jk} \cdot n_i)$$

$$\frac{\partial(net_k)}{\partial(w_{jk})} = n_i$$

şeklinde hesaplanılmıştır ve bu durumda $\frac{\partial(th)}{\partial(w_{jk})}$ değeri hesaplanılmış olur. Türevler yerine yazıldığında ise

$$\frac{\partial(th)}{\partial(w_{jk})} = (n_k - b_k)(1 - n_k)n_k$$

denklemini elde edilir. Elde edilen bu denklem ile δ_k değeri (2.3.12) denkleminde verildiği gibi hesaplanır.

$$\delta_k = h_k(1 - n_k)n_k \quad (2.3.12)$$

δ_k yani k 'inci üitedeki hata miktarı da hesaplandıktan sonra ağırlıkların değişim miktarı ile t 'inci iterasyondaki yeni ağırlıklar (2.3.13) denkleminle güncellenir.

$$w_{jk}(t) = w_{jk}(t-1) + \Delta w_{jk}(t) \quad (2.3.13)$$

Böylelikle ara katman ile çıkış katmanı arasındaki ağırlık değişimi tamamlanmış olur.

Ağırlık değişiminden sonra ara katman ve çıktı katmanı arasındaki eşik değerleri değişim miktarı ve t 'inci iterasyondaki yeni değeri ise

$$\Delta B_k(t) = \lambda \delta_k + \alpha \Delta B_k(t-1)$$

$$B_k(t) = B_k(t-1) + \Delta B_k(t)$$

şeklinde hesaplanır. Bu durumda ara katman ile çıktı katmanı arasındaki ağırlıklar ve eşik değerleri değiştirilmiştir. İleri yayılım algoritmasının tamamlanması için ara katman ile girdi katmanı arasındaki ağırlıkların ve eşik değerlerinin de değiştirilmesi gerekmektedir.

Ara katman ile girdi katmanı arasındaki ağırlıkların değiştirilmesi, ara katman ile çıktı katmanı arasındaki ağırlıkların değişim mantığıyla hemen hemen aynıdır. Ara katman ile çıktı katmanı arasındaki ağırlıkların değişim miktarı yani δ hesaplanırken çıktı katmanındaki bir proses elemanının hatası göz önüne alınarak hesaplanılmıştır. Fakat ara katman ile girdi katmanını arasındaki ağırlıkların değişim miktarının hesaplanılmasında, çıktı katmanındaki bütün proses elemanlarının hatası göz önüne alınır. Çünkü çıktı katmanındaki her bir proses elemanının hata miktarında girdi katmanı ile ara katmanı arasındaki ağırlıklarının da katkısı vardır. Ara katman ile girdi katmanı arasındaki ağırlıkların değişim miktarı aynı ara katman ile çıktı katmanı arasındaki ağırlıkların değişim miktarı hesaplanması (2.3.14) eşitliği ile bulunur.

$$\Delta w_{ij}(t) = \lambda \delta_j n_i + \alpha \Delta w_{ij}(t-1) \quad (2.3.14)$$

Hata miktarı hesaplanırken çıktı katmanındaki hataların toplamı bütün ağa yayılır ve hata miktarı (2.3.14) eşitliğinde verildiği gibi

$$\delta_j = n_j(1 - n_j) \sum_k \delta_k w_{jk} \quad (2.3.15)$$

belirlenir. Hata miktarı belirlendikten sonra ağırlıkların değişimi ise (2.3.16) ile hesaplanır.

$$w_{ij}(t) = w_{ij}(t-1) + \Delta w_{ij}(t) \quad (2.3.16)$$

Aynı şekilde eşik değişim miktarları ve yeni değerleri

$$\Delta B_j(t) = \lambda \delta_j + \alpha \Delta B_j(t-1)$$

$$B_j(t) = B_j(t-1) + \Delta B_j(t)$$

şeklinde hesaplanır. ÇKA geri yayılımı da bu şekilde tamamlanmış olur. Hata miktarı istenilen miktarda minimize edilene kadar ağırlıklar değiştirilir.

2.3.2. Levenberg-Marquardt Optimizasyonu

Levenberg-Marquardt algoritması lineer olmayan problemlere nümerik çözüm getirir. Bu algoritma geri yayımlı çok katmanlı yapay sinir ağlarında kullanılan Gradyan düşümü ile Gauss-Newton algoritmasının bir kombinasyonudur. Bu kombinasyon sayesinde Gauss-Newton algoritmasından daha dayanıklıdır ve Gradyan düşümü yönteminden daha hızlı çalışır.

Gradyan Düşümü

Yapay sinir ağlarında öğrenmenin gerçekleşmesi için istenen çıktılarla ağın çıktıları arasındaki farkın yani hata miktarının azaltılması gerekmektedir. Gradyan düşümü algoritması bir optimizasyon yöntemi olduğundan hata miktarının minimize edilmesi için uygun yöntemlerden biridir. Yöntem genel itibarı ile eğimden yani fonksiyonun türevinden yararlanır. Adım adım değişken yenilenmesi ile fonksiyonun türevinin küçülen yönünde ilerler ve minimum noktaya doğru değişken değişimi yapılır.

ε minimize edilmeye çalışılan hata fonksiyonu olsun. Hata fonksiyonu w_i 'lere yani ağırlıklarına bağlı olduğundan algoritmanın amacı

$$\min(\varepsilon(w_i))$$

şeklinde ifade edilebilir. Gradyan olarak adlandırılan operatör ise fonksiyonun bütün değişkenlerine göre türevidir. ε fonksiyonunun w_i 'lere göre gradyanı (2.3.17) denklemi ile tanımlanır.

$$\nabla\varepsilon(w_i) = \left[\frac{\partial\varepsilon}{\partial w_1}, \frac{\partial\varepsilon}{\partial w_2}, \dots, \frac{\partial\varepsilon}{\partial w_n} \right] \quad (2.3.17)$$

Algoritma herhangi bir w_i ağırlığından w_{i+1} ağırlığına geçişi (2.3.18) eşitliği ile sağlar.

$$w_{i+1} = w_i - \lambda d_i \quad (2.3.18)$$

(2.3.18) denkleminde kullanılan λ adım büyüklüğü ya da öğrenme katsayısı olarak adlandırılan pozitif bir değerdir. d_i ise hata fonksiyonunun, w_i noktasında hesaplanan türevidir, yani gradyan vektörüdür. Hata fonksiyonuna birinci dereceden Taylor açılımı uygulanarak bir adım ilerideki hata fonksiyonu (2.3.19) eşitliği ile ifade edilir.

$$\varepsilon(w_{i+1}) = \varepsilon(w_i) - \lambda d_i^t d_i \quad (2.3.19)$$

w_i yardımıyla hata fonksiyonunda azalma olduğu tahmin edilir. Öğrenme katsayısı küçük tutulduğunda hata fonksiyonunun azalması sağlıklı bir şekilde gerçekleşmektedir. Fakat işlem süresi bakımından zaman alıcı bir algoritmadır.

Gauss-Newton Algoritması

Gauss-Newton algoritması lineer olmayan problemlerin çözümünde kullanılan Newton algoritmasının geliştirilmiş bir versiyonudur. Newton-Gauss algoritması sadece $\sum \varepsilon^2(w_i)$ gibi fonksiyonları minimize etmek için kullanılır. Newton algoritmasına göre avantajı ise ikinci türevin de kullanılmasıdır.

Minimize edilmek istenen fonksiyon $f(w) = \frac{1}{2} \sum_{i=1}^m \varepsilon_i^2(w)$ şeklinde olsun.

Algoritmanın amacı gradyan düşümünde olduğu gibi $f(w)$ fonksiyonunu minimize etmektir.

Fonksiyonun gradyanı (2.3.20) denklemi ile belirlenir.

$$\nabla f(w) = \frac{\partial f}{\partial w_i} = \sum_{j=1}^m \varepsilon_j \nabla \varepsilon_j = J(w)^T \varepsilon(w) \quad (2.3.20)$$

(2.3.20) eşirliğinde $J(w)$ ile gösterilen ifade Jacobian matrisidir ve bu matris (2.3.21) eşitliği ile gösterilir.

$$J(w) = \begin{pmatrix} \frac{\partial \varepsilon_1}{\partial w_1} & \cdots & \frac{\partial \varepsilon_1}{\partial w_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial \varepsilon_m}{\partial w_1} & \cdots & \frac{\partial \varepsilon_m}{\partial w_n} \end{pmatrix} \quad (2.3.21)$$

Gauss-Newton algoritmasında tanımlanan Hessian matrisi yani minimize edilmeye çalışılan $f(w)$ fonksiyonun ikinci türevinin kullanıldığı kısım (2.3.22) eşitliği ile hesaplanır.

$$\begin{aligned} \nabla^2 f(w) &= \sum_{j=1}^m \nabla \varepsilon_j(w) \nabla \varepsilon_j(w)^T + \sum_{j=1}^m \varepsilon_j(w) \nabla^2 \varepsilon_j(w)^T \quad (2.3.22) \\ &= J(w_i)^T J(w_i) + \sum_{j=1}^m \varepsilon_j(w) \nabla^2 \varepsilon_j(w)^T \\ &= A(w) + S(w) \end{aligned}$$

$S(x)$ fonksiyonu çok küçük bir değer çıkarsa $\nabla^2 f(w)_{(hessian)} \approx J(w_i)^T J(w_i)$ şeklinde gösterilir ve Hessian matrisi (2.3.23) eşitliğindeki gibi hesaplanır.

$$H(w) = \begin{pmatrix} \frac{\partial^2 \varepsilon_1}{\partial^2 w_1} & \cdots & \frac{\partial^2 \varepsilon_1}{\partial w_1 \partial w_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 \varepsilon_m}{\partial w_1 \partial w_n} & \cdots & \frac{\partial^2 \varepsilon_m}{\partial^2 w_n} \end{pmatrix} \quad (2.3.23)$$

Jacobian matrisi ve Hessian matrisi hesaplandıktan sonra Gauss-Newton metodu ile ağırlık değişimi (2.3.24) eşitliği ile gerçekleştirilir.

$$w_{j+1} = w_j - (J(w_j)J(w_j)^T + S(w_j))^{-1} + J(w_j)^T \varepsilon(w_j) \quad (2.3.24)$$

Levenberg-Marquardt algoritması Gauss-Newton ve Gradyan Düşümü algoritmalarının kombinasyonudur. İki algoritmanında avantajlarını kullanır. Gradyan Düşümü algoritmasının hassasiyetini ve Gauss-Newton algoritmasının hızını. Levenberg-Marquardt algoritmasında ağırlık değişimi (2.3.25) eşitliği ile gerçekleştirilir.

$$w_{j+1} = w_j - (J(w_j)J(w_j)^T + \lambda I)^{-1} \nabla \mathcal{E}(w_j) \quad (2.3.25)$$

(2.3.25) eşitliğinde, λ daha öncede Gradyan Düşümünde tanımlanan adım büyüklüğüdür. Algoritma şu şekilde çalışır.

- λ , λ_{ust} , λ_{alt} parametreleri seçilir, $J(w_j)$ ve $\mathcal{E}(w_j)$ değerleri hesaplanır. $C = 1/2\mathcal{E}^2(w_i)$ hesaplanır.
- $w_{j+1} = w_j - (J(w_j)J(w_j)^T + \lambda I)^{-1} \nabla \mathcal{E}(w_j)$ ve $C_{yeni} = 1/2\mathcal{E}^2(w_{i+1})$ yani yeni hata fonksiyonları kareleri toplamı hesaplanır.
- Eğer $C_{yeni} < C$ ise $w_i = w_{i+1}$, $C_{yeni} = C$, $\lambda = \lambda / \lambda_{alt}$ değerleri yenilenir.
- Eğer $C_{yeni} > C$ ise C ve w_i değerleri korunur fakat $\lambda = \lambda * \lambda_{ust}$ yapılır.
- Hata fonksiyonu istenen değer kadar minimize edilmişse algoritma iterasyon yapmaz. Eğer minimize edilememişse algoritma adımları tekrarlanır.

3. DENEYSEL SONUÇLAR

Bu bölümde, belirlenen eğitim dokümanları üzerinde materyal ve metot kısmında anlatılan gözetimli öğrenme yöntemlerini kullanarak elde edilen deneysel sonuçlar verilecektir. Eğitim aşamasında dört farklı konuda altışar doküman kullanılmıştır. Bu dokümanların konuları Grup Teori, Limit ve Süreklilik, Türev, Çok Katmanlı Algılayıcılar ve Geri Beslemeli Algoritmalar olarak belirlenmiştir. Toplam doküman boyutu 3.56 MB dır. Her bir doküman ön işlemden geçirilmiş ve her bir doküman konusu için toplamda 2000 farklı sözcük dizilimi belirlenmiştir. Bu bu sözcük dizilimleri arasından her bir konu başlığı için sırasıyla 918, 1154, 672, 1530 tane aday kavram seçilmiştir. Her bir dokümanın kavramlarının bu doküman için aday öğrenim kavramı olup olmadığı uzman tarafından belirlenmiş ve bu aday kavramları içeren G_2 adında kümeler oluşturulmuştur.

Bu çalışmada eğitim ve test aşamasında dört farklı metot uygulanmıştır. Bu metotlar 3-NN ile ağırlıklı oylama, 5-NN ile ağırlıklı oylama ve Levenberg-Marquardt metotlarıdır. Eğitim aşamasında d_1 ve d_2 dokümanlarından elde edilmiş G_2 kümeleri ile çalışılmıştır. Fakat G_2 kümesi çok sayıda elemana sahip olduğundan, aday kavramların ayırt edici olmasını sağlamak amacı ile uzman tarafından seçilerek azaltılmış ve bu kavramlar eğitim aşamasında kullanılmıştır. Bu uygulama yapılarak eğitim aşamasında daha iyi sonuçlar elde edilmiştir.

Çizelge 3.1. Her bir konu için G_2 kümesindeki aday kavram oranları

<i>Döküman</i>	Öğrenim konuları			
	<i>Grup Teori</i>	<i>Türev</i>	<i>Limit Süreklilik</i>	<i>ÇKA ve Geri Beslemeli Algoritmalar</i>
d_1	0.140	0.244	0.120	0.280
d_2	0.156	0.178	0.080	0.274
d_3	0.160	0.194	0.174	0.252
d_4	0.148	0.232	0.144	0.190
d_5	0.156	0.188	0.090	0.250
d_6	0.150	0.122	0.062	0.280

Çizelge 3.2. F-ölçüsü skoru ile yöntemlerin doğrulama aşaması

Metot	Öğrenim konuları			
	Grup Teori	Türev	Limit ve Süreklilik	ÇKA ve Geri Beslemeli Algoritmalar
3-NN Ağırlıklı Oylama	1.000	1.000	1.000	1.000
5-NN Ağırlıklı Oylama	1.000	1.000	1.000	1.000
Levenberg Marquardt	0.993	0.977	1.000	0.975

Daha sonraki eğitim modellerinde ise G_2 kümesindeki elemanların yarısı rasgele seçilmiştir. Çizelge 3.1 de herbir konu için oluşturulan G_2 kümelerindeki aday kavram oranları verilmiştir . Çizelge 3.2 ile anlatılan yöntemler kullanılarak eğitim aşamasındaki yöntem doğrulaması F-ölçüsü skoru ile herbir konu için hesaplanmıştır.

Test aşamasında ise d_3, d_4, d_5, d_6 dökümanlarından G_2 kümesi çıkartılmıştır ve eğitilmiş metotlara sunulmuştur. Alınan sonuçlara göre başarı ölçümlendirme duyarlılık, Kesinlilik ve F-ölçüsü ile yapılmıştır. Kesinlilik doğru tahmin edilen kavramların tahmin edilen bütün kavramlara oranıdır. Duyarlılık ise doğru tahmin edilen kavramların test dökümanındaki bütün doğru kavram miktarına oranıdır. F-ölçüsü ise duyarlılık ve kesinlilik ölçülerinin ikisinde iyi olmasını amaçlayan ölçüm tipidir.

$$F = 2 \cdot \frac{\text{Kesinlilik} \cdot \text{Duyarlılık}}{\text{Kesinlilik} + \text{Duyarlılık}} \quad (3.0.1)$$

F-ölçüsünün denklem (3.0.1)' deki gibi hesaplanır .

Çizelge 3.3. 3-NN ile Ağırlıklı Oylama metodunun Duyarlılık ,Kesinlilik ve F-ölçüsü test sonuçları.

Konu	3-NN ile ağırlıklı oylama			
	Döküman	Duyarlılık	Kesinlilik	F-ölçüsü
Grup Teori	d_3	1.00	0.95	0.97
	d_4	1.00	0.96	0.98
	d_5	1.00	1.00	1.00
	d_6	1.00	0.96	0.98
Limit ve Süreklilik	d_3	1.00	0.98	0.99
	d_4	1.00	1.00	1.00
	d_5	1.00	1.00	1.00
	d_6	1.00	0.96	0.98
Türev	d_3	1.00	0.97	0.98
	d_4	1.00	1.00	1.00
	d_5	1.00	0.98	0.99
	d_6	1.00	1.00	1.00
ÇKA ve Geri Beslemeli Algoritmalar	d_3	1.00	0.91	0.95
	d_4	1.00	0.85	0.92
	d_5	1.00	0.98	0.99
	d_6	1.00	1.00	0.99

Çizelge 3.4. 5-NN ile Ağırlıklı Oylama metodunun Duyarlılık, Kesinlilik ve F-ölçüsü test sonuçları.

Konu	5-NN ile ağırlıklı oylama			
	Döküman	Duyarlılık	Kesinlilik	F-ölçüsü
Grup Teori	d_3	1.00	0.95	0.97
	d_4	1.00	0.96	0.98
	d_5	1.00	1.00	1.00
	d_6	1.00	0.96	0.98
Limit ve Süreklilik	d_3	1.00	0.98	0.99
	d_4	1.00	1.00	1.00
	d_5	1.00	1.00	1.00
	d_6	1.00	0.96	0.98
Türev	d_3	1.00	0.97	0.98
	d_4	1.00	1.00	1.00
	d_5	1.00	0.98	0.99
	d_6	1.00	1.00	1.00
ÇKA ve Geri Beslemeli Algoritmalar	d_3	1.00	0.91	0.95
	d_4	1.00	0.85	0.92
	d_5	1.00	0.98	0.99
	d_6	1.00	0.97	0.98

Çizelge 3.5. Levenberg-Marquardt optimizasyonu Duyarlılık, Kesinlik ve F-ölçüsü test sonuçları.

Konu	Levenberg-Marquardt			
	Döküman	Duyarlılık	Kesinlik	F-ölçüsü
Grup Teori	d_3	0.60	0.24	0.34
	d_4	0.64	0.20	0.31
	d_5	0.47	0.29	0.36
	d_6	0.64	0.20	0.30
Limit ve Süreklilik	d_3	0.75	0.20	0.32
	d_4	0.64	0.17	0.27
	d_5	0.58	0.10	0.18
	d_6	0.68	0.08	0.14
Türev	d_3	0.98	0.26	0.41
	d_4	0.65	0.30	0.27
	d_5	0.67	0.21	0.23
	d_6	0.93	0.16	0.28
ÇKA ve Geri Beslemeli Algoritmalar	d_3	0.57	0.28	0.38
	d_4	0.69	0.20	0.31
	d_5	0.62	0.27	0.38
	d_6	0.74	0.27	0.40

4. SONUÇ

Bu çalışma ile yapay zeka yöntemleri kullanılarak bir eğitim içeriğinden öğrenim kavramlarının tespit edilebileceği gösterilmiştir. Öğrenim kavramlarının belirlenmesi, konuyla ilgili kavram hiyerarşisinin ve kavram haritasının çıkarılması işlemlerinin ilk aşama olarak kabul edilebilir. Kavram hiyerarşisinin ve kavram haritasının konunun uzmanı bir kişi tarafından bile belirlenmesi zaman alıcı bir süreç olmakla beraber karmaşık ve yorucudur. Bu durumun otomatik olarak aşılabilmesi öğrenim sistemleri için önemli bir gelişme olacaktır.

Öğrenim kavramlarının tespit edilebilmesinin diğer bir avantajı ise bir eğitim dokümanının içerdiği öğrenim kavramlarının ağırlığına göre temsil edilebilmesidir. Bu sayede eğitim içeriğinin hangi öğretim kavramlarını içerdiğinin ve ağırlıklı olarak hangi öğretim kavramları üzerinde odaklandığının tespiti yapılmış olur. Örneğin uzaktan eğitim yapan bir sistemde "türev" konusu hakkında iki farklı doküman var olsun. Bu iki dokümandan birisi diğerine göre "kısmi türev" kavramı hakkında daha çok örnek ve anlatım içersin. Sistem, "kısmi türev" konusunda sıkıntı çeken bir öğrenciye "türev" konusundan daha özel olarak, belirtilen öğretim kavramı üzerine odaklanan eğitim içeriğinin çalışılmasını önerebilir. Doğal olarak öğrenci sistemden daha hızlı, daha verimli ve ihtiyaçlarına doğru bir şekilde yanıt alarak kişisel başarısını arttırabilir. Benzer olarak, öğretmenler de sınıf içinde anlaşılması güç olan bir kavramın öğretilmesi amacıyla sistemin önerdiği kaynağı kullanabilirler.

Bu çalışmada öğretmenli öğrenim algoritması kullanan iki farklı yöntem kullanılmış ve birbiri ile karşılaştırılmıştır. Bu yöntemler Ağırlıklı Oylamayla k -En Yakın Komşu ve Levenberg-Marquardt optimizasyonu ile Çok Katmanlı Algılayıcı algoritmalarıdır. Yapılan deneysel çalışmanın sonucunda Ağırlıklı Oylamayla k -En Yakın Komşuluk metodu diğer yönteme göre daha verimli sonuçlar vermiştir.

Daha genel olarak, yapay sinir ađı modelleri ile öğrenim kavramları tespitinin yapılabildiđi gösterilmiştir. Tezde, daha farklı öznitelik vektörleri oluşturularak öğrenim kavramlarını eğitim içeriğindeki diğer sözcük dizilimlerinden ayırt edecek çalışmaların yapılabileceđi ve öğrenim kavramlarının tespitinde farklı yapay sinir ađı modelleri oluşturulabileceđi sonucuna varılmıştır.

KAYNAKLAR

- [1] Culp, K . M. , Honey, M. , Mandinach, E. 2005. A Retrospective on Twenty Years of Education Technology Policy. **Journal of Educational Computing Reserch**, 32(3):pp. 279-307.
- [2] Kozma, R. B. 2011. ICT, Education Transformation, and Economic Development: an analysis of the US National Educational Technology Plan. **E-Learning and Digital Media**, 8(2):pp. 106-120.
- [3] Rose, R. , Wskd, L. 2012. The National Educational Technogy Plan: continuing the dialogue.**E-Learning and Dgital Media**, 9(1):pp 96-99.
- [4] Xiao, Y., Meier, E. B 2011. Education Technology as a Catalyst for Education Development in Chine: A Policy Perspective. In Huang T, Wiseman AW editors, The Impact and Transformation of Education Policy in Chine (International Perspectives on Education and Society), **Emerald Group Publishing Limited**, 15:pp. 313-343.
- [5] Dell, C. A., Low, C. , Wilker, J. F. 2010. Comparing Student Achievement in Online and Face-to-Face Class Formats, **Journal of Online Learning and Teaching**, 6(1):pp. 30-42.
- [6] Fortune, M. F., Shifflett, B. , Sibley, R. E. 2006. Acomparison of online (high tech) and traditional (high touch) learning in business communication courses in Silicon Valley, **Journal of Education for Business**, 81(4):pp. 210-214.
- [7] Weber, J. M., Lennon, R. 2007. Multi-course comparison of traditional versus web-based course delivery systems, **The Journal of Education Online**, 4(2):pp. 1-19.
- [8] Qasim, I. , Jeong, J. W. , Khan, S. , Lee, D. H. 2011. Exploiting Affinity Propagation for Automatic Acquisition of Domain Concept in Ontology Learning. **International Conference on Emerging Technologies ICET**, 7:pp. 1-6.
- [9] Bracewell, D. B., Jiajun, Y. , Fuji, R. 2008. Single document keyword extraction for Internet was article, **International Journal of Innovative Computing Information and Control**, 4(4):pp: 905-903.
- [10] Gonenc, E. , Cicekli, Y. 2007 Using lexical chains for keyword extraction, **Informatim Processing and Management**, 43(6):pp 1705-1714

- [11] HaCohen-Kerner, Y. , Gross Z. , Masa, A. 2005. Automatic Extraction and Learning of Keyphrases from Scientific Articles. **In Gelbukg A editor. CICling, Lecture Notes in Computer Science. Springer-Verlag, Berlin Heidelberg**, 3406:pp. 657-669.
- [12] Hulth, A. , Karlgren, J. , Jonsson, A. , Bostrom, H. , Asker, L. , 2001. Automatic keyword extraction using domain knowledge, Computational Linguistics and Intelligent Text Processing, **Lecture Notes in Computer Science**, 2004:pp. 472-482.
- [13] Martinez- Fernandez, J. L. , Garcia-Serrano, A., Martinez, P., Villena, J. 2004. Automatic keyword extraction for news finder, Adaptive Multimedia Retrieval, **Lecture Notes in Computer Science**, 3094:pp. 99-119.
- [14] Turney, P. D., 2000. Learning algorithm for keyphrase extraction, **Journal of Information Retrieval**, 2(4):pp. 303-336.
- [15] Vivaldi, J. , Rodriguez, H. 2007 Evaluation of term and term extraction system, **A practical approach, Terminology**, 13(2):pp. 225-248.
- [16] Daille, B. 1996. Study and implementation of combined techniques for automatic extraction of terminology, **The Balancing Act: Combining Aymbolic and Statistical Approaches to Language**, 1:pp. 49-66. Page 17 of 25.
- [17] Frantzi, K. , Ananiadou, S. , Mima, H. 2000. Automatic recognition of multi-word terms:the C-value/NC-value method, **International Journal on Digital Libraries**, 3:pp. 115-130.
- [18] Cimiano, P. , Völker, J. 2005. Texttoonto, **Natural Language Processing and Information System** ,:pp. 227-238.
- [19] Zouaq, A. , Nkambou, R. , 2008. Building Domain Ontologies from Text for Educational Purposes, **IEEE Transactions on Learning Technologies**, 1(1):pp. 49-62.
- [20] Villalon, J. , Calvo, R. A. , July 15-17-2009. Concept Extraction from student essays, towards Concept Map Mining, **Ninth IEEE International Conference on Advanced Learning Technologies** ,:pp. 221-225, Riga, Latvia.
- [21] Gunel, K. , Asliyan, R. , 2010. Extracting Learning Concepts from Educational Texts in Intelligent Tutoring Systems Automatically, **Expert System with Applications**, 37(7):pp. 5017-5022.

- [22] Fix, E. , Hodges, J. L. 1951 Discriminatory analysis, nonparametric discrimination: Consistency properties, Technical Report 4, USAF School of Aviation Medicine, Randolph Field, Texas.
- [23] Aha, D. W., Kibler, D., Albert, M. K. 1991. Instance-based learning algorithms, **Machine Learning**, 6(1):pp. 37-66.
- [24] McSherry, D. , Stretch C. , 2010. An analysis of order dependence in k-nn, **Artificial intelligence and cognitive science**, L. Coyle and J. Freyne (Editors), 6206:pp. 207-218 Springer Berlin Heidelberg.
- [25] Levenberg, K. , 1944. A Method for the Solution of Certain Non-Linear Problems in Least Squares, **Quarterly of Applied Mathematics**, 2:pp. 164-168.
- [26] Marquardt, D. , 1963. An ALgorithm for Least-Squares Estimation of Nonlinear Parameters, **SIAM Journal on Applied Mathematics**, 11(2):pp. 431-441. bibitem27 Kohonen, T. , 1998. Learning Vector Quantization, In: M.A. Arbib, editor, **The Handbook of Brain Theory and Neural Networks**, MIT Press, :pp. 537-540. Cambridge, MA.
- [27] Öztemel, E. , 1992. Integrating Expert System and Neural Networks for Intelligent On-Line Statistical Process Control, University of Wales, College of Cardiff, Ph. D. Thesis.

ÖZ GEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı : Özcan Yıldırım

Doğum Yeri ve Tarihi : Kars 09.06.1986

EĞİTİM DURUMU

Lisans Öğrenimi : Ege Üniversitesi Fen Bilimleri Fak., Matematik Böl., Bilgisayar Ops.

Yüksek Lisans Öğrenimi : Adnan Menderes Üniversitesi Fen Bilimleri Fak., Matematik Böl.

Bildiği Yabancı Diller : İngilizce

BİLİMSEL FAALİYETLERİ

- a) Yayınlar
- b) Bildiriler
- c) Katıldığı Projeler

Aşlıyan R., Günel K., Yılmaz, R., Kolyiğit, Ö., Yıldırım, Ö., Web Sayfalarının Sınıflandırılması, Bilimsel Araştırma Projeleri Kurulu, ADU, FEF-11007, Aydın, Türkiye, 29/03/2011, 29/09/2012.

İLETİŞİM

E-posta Adresi : ozzcanyldrm@gmail.com

Tarih : 11.06.2013