

SİNİR MANTIĞI

Prof. Dr. Remzi YILDIRIM

(ÖDEV)

2016-AYBÜ-ANKARA

SİNİR MANTIĞI

SİNİR MANTIĞI: TOERİ ve UYGULAMASI

Danışman: Yrd. Doç. Dr. Remzi YILDIRIM

2006 - Ankara

Kullanılan Kısaltmalar

LT:	Doğrusal Eşik Elementi
LTD:	Doğrusal Eşik Fonksiyonu
LTM:	Çok Eşikli Doğrusal Eşik Elementi
S[f]:	En Düşük Ağırlıklı Doğrusal Eşik Fonksiyonu
$\hat{L}T$:	Düşük Ağırlıklı Doğrusal Eşik Fonksiyonları
$\hat{L}T^{0d}$:	$S[f] \leq n^d$ için Doğrusal Eşik Fonksiyonlarının Kümesi
VLSI:	En Geniş Ölçekli Entegrasyon
$\hat{L}T_2$:	Düşük Ağırlıklı kapılardan oluşan Devre
LT[D]:	D büyüklüğündeki Ağırlıklardan oluşan LT Fonksiyonları Kümesi
AND:	Ve Fonksiyonu
OR:	Veya Fonksiyonu
PRODUCT:	Çarpım Fonksiyonu
COMP:	Karşılaştırma Fonksiyonu
NOT:	Değil Fonksiyonu
MAJ:	Çoğunluk Fonksiyonu
AON:	Ve, Veya, Değil gibi Kapılara Verilen Genel İsim
Δ :	Determinant

Özet

İnsan beyinleri temel yapı blokları olarak bilgisayarlara göre birkaç kat daha yavaş olmasına rağmen birleşik iyileştirme ve görüntü ses tanımlama gibi zor problemlerin çözümünde bilgisayarlara göre çok daha üstündür. Bu gözlem yapay nöron alanındaki ilgiyi arttırmaktadır. Sonraki biyolojik sinirlerin hareketinden esinlenen yapay sinirleri bağlayarak kurulmuştur. Bu tezde n ikili girdilerin nörona benzer doğru yanlış fonksiyonlarını hesaplayan doğrusal başlangıç elemanının(LT) olarak adlandırılan yapay nöronun doğru yanlış versiyonunu düşüneceğiz. LT elemanı onun doğru yanlış girdilerinin ağırlıklı toplamının işaretini çıkar. LT devreleri olarak adlandırılan LT elemanlarından oluşan ağ devrelerindeki çalışmanın ana konusu hesaplanabilir kapasitelerinin ve limitlerinin tahminini ve VE, VEYA, DEĞİL(AON devreleri) kapılarına dayandırılmış geleneksel doğru yanlış mantık devrelerinin özelliklerinin kıyaslanmasıdır. Örneğin LT devreleri AON devrelere göre tamsayıların toplaması, çarpması, bölmesi gibi önemli fonksiyonların uygulamasında daha verimlidir.

LT elemanının AON kapısından daha güçlü olduğunu görmek kolay, basitçe birinin ağırlığı seçmedeki özgürlüğü yüzünden. Gerçekten ağırlıkların farklı seçimleri farklı doğru yanlış fonksiyonları üretir. Bu nedenle, basit bir LT elemanı ile tanımlanabilecek n-girdili doğru yanlış fonksiyonlarının sayısı 2^n 'dir. Bu ek üs eklenti karışıklığı ile orantılıdır. Bazı LT fonksiyonları büyüklük olarak farklı ağırlıklar gerektirir karşılık gelen LT elemanlarının zor donanım ve yazılım uygulamalarında yardımcı olarak. Bundan dolayı LT devrelerinin alanında teorik araştırma ağırlıklar üzerine odaklanmıştır kısıtlı ağırlıklarda LT elemanlarının kısmi kuvvetlerinde. 1971'lerde Muroga doğrusal başlangıç elemanının tamsayı ağırlıklarla ifade edilebileceğini ispatladı. Ağırlıkların büyüklüklerini doğal sayılarla kısıtlayarak orijinal LT elemanının gücü kaybedilmiş olmaz. Bu durumu reel sayıların alt kümelerine genelleayebiliriz. Örneğin, ağırlıklar tamsayıların karesi olarak sınırlandırılabilir ve tüm LT fonksiyonları tanına bilinir. Şu soruyu soracağız. Kendi ağırlıklarıyla çizilen ağırlıkları bütün LT fonksiyonlarıyla ifade edilebileceğini garanti eden $D \in \mathbb{R}$ üzerindeki altkümelerin şartları nelerdir? Ağırlıkların karışıklığının diğer bir bakış açısı da girdi sayısı arttıkça büyümesidir. Sayfa [17], [33], [38], [43] de gösterildiği gibi üslü olarak büyüyen ağırlıklarıyla tek bir eşik elementile ifade edile bilinen doğrusal eşik fonksiyonları vardır ama daha küçük polinom olarak büyüyen ağırlıklardaki eşik elementleriyle ifade edilemez. Sonucun ışığı doğrultusunda küçük (polinom olarak büyüyen) ağırlık fonksiyonlarının sınıfı sayfa [43]'te doğrusal eşik fonksiyonları setiyle LT adında bir sınıf tanımlayarak yukarıdaki soruyla ilgilenilmiş oldu. Biz tek bir LT elementi üzerine odaklanacağız. Polinom ve üs ağırlıkları arasında ayırımı daha da rafine ederek aradaki boşlukları doldurmaya izin veren en düşük ağırlıklar ile eşik fonksiyonları kurmak için iki yeni metodumuz olacak. Yani, polinomun derecesi d'ye göre LT alt sınıflarına bölünen polinom boyutlu ağırlıklarla doğrusal eşik fonksiyonlarının sınıfını ispat edeceğiz. Aslında daha genel bir sonuçla herhangi sayıda girdi ve ağırlık için doğrusal eşik fonksiyonu olduğunu ispatlayacağız.

Bazı LT fonksiyonları girdi değişkenlerinin sayısı ile üslü olarak büyümeyi gerektirmektedir sayfa [13], [18], 'de gösterildiği gibi bu tip fonksiyonlar küçük boyutlarda polinom olarak büyüyen LT kapılarının iki katmanlı devreleriyle değiştirilebilir. Sayfa [18]'de büyük katsayılarla belli fonksiyonlara odaklanarak gösterildiği gibi devrenin boyutu üzerindeki en iyi bilinen sınırı geliştireceğiz.

Ayrıca açık iki katmanlı devre üreteceğiz. İki katmanlı LT devreleri farklı doğrusal elementlerden oluşmuştur ama bazı doğru yanlış fonksiyonları için eşitlik, toplama ve çarpma gibi, birinci tabakanın kapıları aynıdır. Bu durumun avantajından faydalanmak için yeni bir doğru yanlış hesaplayan bir element tanıtacağız. İşaret fonksiyonu yerine girdilerinin ağırlıklı toplamalarının doğru yanlış fonksiyonlarını hesaplar. Çoklu geçişli doğrusal eşikler anlamına gelen LTM hesaplama elementi diyeceğiz. VLSI uygulamasında bağlamında LTM'nin avantajları daha belirgin olacaktır. Gerçekten bu yeni model simetrik fonksiyonun yerleşim alanını $O(n^2)$ 'den $O(n)$ 'e düşürmektedir. LT ve LTM elementlerinin VLSI uygulamalarını göstereceğiz. Programlanabilir ve donanımsal olarak bağlanabilir iki çeşit element yapılmıştır. Programlanabilir elementler ağırlıkların değerlerinin depolamak için yüzen kapıların şarjını kullanmaktadır.

Uzun yıllar boyunca doğrusal eşik mantığı konusuna, hesaplanabilir devre karışıklığı, sayfa [38], [56], ve donanım uygulamasını, sayfa [40], [48] iki farklı yolla yaklaşılmıştır. Şaşırtıcı olarak, bu iki yaklaşım arasında çok küçük bir iletişim vardır. Bütün olarak, bu tez eşik devrelerin uygulamasını ve teorisi arasında bir bağlantı kurmaya yönelik bir adımdır. Bu durumun katkısı üç seviyeli. Teorik seviyede, LT ve LTM gibi fonksiyonların yeni sınıfları tanımlandı ve hesaplama güçleri tahmin edildi. Algoritmik seviyede, reel ağırlıkların keyfi reel sayıların, tamsayı ağırlıklarından çizilen ağırlıklara nasıl dönüştüğünü göstereceğiz, ayrıca en düşük ağırlıklarda LT fonksiyonlarını nasıl kuracağımızı ve son olarak LT_2 devreleri(düşük ağırlıklı kapılardan oluşan devre) üreten ve COMP denilen kıyaslama fonksiyonunu hesaplayan bir algoritma göstereceğiz. Ayrıca XOR, toplama, çarpma fonksiyonları gibi yararlı fonksiyonları yapan LTM devreleri sunacağız. Uygulama safhasında tasarımı, yerleşimi ve LT ve LTM'nin VLSI uygulamasının testini göstereceğiz. Eşik mantığının teorik ve pratik yönü arasında bir bağlantı kurmak pratik problemler için çözüm sağlamaya ve uygulama konularından esinlenen yeni teorik soruların tanımına faydalı olacak.

İçindekiler

Özet

1 Giriş	3
1.1 Tanım ve Örnekler LT Fonksiyonu.....	3
1.2 Doğrusal Eşik Elementi.....	7
1.3 Çoklu Eşikler ve VLSI Uygulaması.....	11
1.4 Katkı ve Tezin Organizasyonu.....	14
2 Ağırlıkları Kısıtlamak	16
2.1 Giriş.....	16
2.2 Motivasyon.....	17
2.3 Başlangıç ve Gerekli Çalışma.....	17
2.3.1 $O(2^n)$ n-değişkenli LT fonksiyonları.....	21
2.4 Reel Tamsayılara Ağırlıkları.....	22
2.4.1 Eşik Fonksiyonları Ağırlıkta $O(n \log_2 n)$ bit Gerektirmektedir.....	22
2.4.2 Eşik Fonksiyonları Ağırlıkta $6(n \log_2 n)$ bit Gerektirmektedir.....	25
2.4.3 Reel Sayıları Tamsayılara Dönüştüren Bir Algoritma.....	25
2.5 Ağırlıkları Keyfi Bir Kümeye Dönüştürme.....	29
2.6 Sonuç.....	37
3 En Düşük Ağırlıklar	39
3.1 Giriş.....	39
3.1.1 Motivasyon.....	40
3.1.2 Organizasyon.....	41
3.2 Başlangıç ve Gerekli Çalışma.....	42
3.2.1 Ağırlıkları Asgari Yapma.....	42
3.2.2 $\{0,1\}$ Karşı $\{-1,1\}$	43
3.3 Genelleştirilmiş Çoğunluk Fonksiyonu.....	44
3.3.1 Matematiksel Ayar.....	44
3.3.2 Ağırlık Vektörleri.....	46
3.3.3 Yapı.....	47
3.4 $\{0,1\}$ Arasında Keyfi Ağırlık Fonksiyonu.....	49
3.4.1 Yaklaşım.....	50
3.4.2 Temel Yapı.....	52
3.4.3 Keyfi Sayıda ve Büyüklükte Değişkenler....	55
3.5 Sonuçlar.....	56
4 Devre Derinliği için Ağırlık Boyutu Satışı	57
4.1 Giriş.....	57
4.2 Kıyaslama için LT_2 Devresi.....	58
4.3 Bilgisayar Simülasyonu.....	62
4.4 $LT_d \subset LT_{d+1}$ 'e Genelleme.....	63
4.5 Sonuç.....	64
5 Çoklu Eşiklerle Doğrusal Eşik Elementi	66
5.1 Giriş.....	66
5.1.1 Tanımlar ve Örnekler.....	67
5.1.2 Organizasyon.....	69
5.2 LTM Yapıları.....	69

5.3	<i>LTM</i> 'in Sınıflandırılması.....	72
5.4	Sınıflandırma Teoreminin İspatı.....	73
5.4.1	İlaveler.....	73
5.4.2	Ayrım.....	74
5.5	Sonuçlar.....	76
6	VLSI Uygulaması ile Programlanabilir Sinir Mantığı	78
6.1	Giriş.....	78
6.2	Sinir Mantığı ve Geleneksel Mantık.....	80
6.3	Programlanabilir ve Donanımsal Bağlı Ağırlıklar.....	81
6.4	Uygulama ve Sonuçlar.....	82
6.5	VLSI Yerleşimi.....	85
6.6	Sonuç.....	88
7	Sonuçlar	90
	Kaynaklar	94

Şekil Listesi

1.1	Doğrusal Eşik Elementi $y = \text{sgn}(-t + \sum_{i=1}^n w_i x_i)$	3
5.1	LT'nin şematik Görünümü ve SYM, LTM Hesaplama Elementleri.....	67
5.2	Her bit için tek LTM kapısında iki dört basamaklı tamsayının toplamı.....	69
5.3	$O(n)$ boyutta LT Devresi ve Tek LTM Kapısı.....	70
5.4	MADD: 3-bit Üç Tamsayının Toplamı X, Y and Z – LTM Elementlerinin Katmanının Kullanarak.....	72
5.5	Sınıflar Arasında İlişkiler.....	72
6.1	Sinir ve Geleneksel Mantık. XOR Hesaplayan İki Devre.....	80
6.2	İki 4-bit Tamsayıların Karşılaştırması.....	82
6.3	Programlanabilir Doğrusal Eşik Elementinin Şeması.....	83
6.4	$-w_0 + \sum_{i=1}^{16} w_i x_i$ Doğrusal Toplamın Yerleşimi- Dört Eşik Elementi gösteriliyor, ikisi programlanabilir, ikisi programlanamaz, ikincisi birim ağırlıkta. Gösterilene alan $168\mu X 360\mu$. Çip MOSIS'ten gelen The chip 2μ teknolojisi kullanılarak yapılmıştır.....	84
6.5	V_{dd} — Eşik'e karşı girdideki 1'lerin sayısı.....	85
6.6	Simetrik fonksiyonlar için LTM(Sağ)'nin LT(sol)'e göre avantajları. Ağırlıklı toplam ilk katmandaki her kapı yerine bir kez uygulanmıştır.....	86
6.7	LTM kapısının yüksek seviyedeki şeması.....	87
6.8	16-girdili LTM elementinin yerleşimi. Çıktı 4-bit hafıza hücresi adresleyen 4-bit bustan oluşmuştur. Ağırlıklı toplam voltajların kapasite toplam olarak Nöron MOS'ta implemente edilmiştir. Çip 2μ teknolojisi bulunan MOSIS'ten yapılmıştır.....	89

Tablo Listesi

1.1	2-değişkenli bağlantı, $OR(x_1, x_2) = \text{sgn}(-1 + x_1 + x_2)$	4
1.2	2-değişkenli ayraç, $AND(x_1, x_2) = \text{sgn}(-2 + x_1 + x_2)$	4
1.3	3-değişkenli çoğunluk, $MAJ(x_1, x_2, x_3) = \text{sgn}(-2 + x_1 + x_2 + x_3)$	5
1.4	2-değişkenli eşitlik, $XOR(x_1, x_2) \neq \text{sgn}(w_0 + w_1x_1 + w_2x_2)$	6

Bölüm 1

Giriş

İnsan beyninin temel yapı blokları birkaç kat daha yavaş olmasına rağmen birleşik optimizasyon, resim ve ses tanıma gibi zor problemlerin çözümünde bilgisayarlara göre çok daha fazla üstündür. Bu gözlem yapay sinir ağları sayfa [20], [37] alanına daha fazla eğilmeyi tetikledi. İkincisi biyolojik nöronlardan esinlenerek yapay sinirler birbirine bağlayarak kurulmuştur. Bu tezde n ikili girdilerin sayfa [32] sinire benzer doğru yanlışı fonksiyonlarını hesaplayan doğrusal başlangıç elemanı(LT) olarak adlandırılan yapay nöronun doğru yanlışı versiyonunu düşüneceğiz. Bir LT elementi doğru yanlışı girdilerinin ağırlıklı toplamının işaretini dışarıya çıkarır. LT devreleri denilen LT elementlerinden oluşan ağların çalışma alanındaki ana konu onların hesaplama kapasitelerindeki tahmini, sınırları ve VE, VEYA, DEĞİL (AON devreleri denilmektedir) kapılarına dayandırılmış geleneksel doğru yanlışı mantık devrelerinin özelliklerinin kıyaslanmasını içermektedir. Örneğin LT devreleri AON devrelere göre tamsayıların toplaması, çarpması, bölmesi gibi önemli fonksiyonların uygulamasında daha verimlidir.

Sinirsel ve doğrusal eşik mantığına iki yaklaşım bulunmaktadır: Teori ve uygulama. LT elementlerini uygulama yapan elektronik devreler altmışlı yıllarda önerilmişti. Bu alandaki çalışmalar halen devam etmektedir. Diğer yandan LT ile ilgili son yapılan teorik araştırma hesaplanabilir devre karmaşasının çatısı, sayfa [38], [56] kurulmuştur. Exclusive-OR (XOR) gibi belli doğru yanlışı fonksiyonların sabit derinlikteki polinom büyüklükteki LT devresiyle ifade edilebileceği gösterilmiştir ama eğer klasik AON uygulaması kullanılırsa üslü olarak büyük bir devre gerektirmektedir. Toplama, çarpma, bölme gibi birçok faydalı fonksiyonun temeli olarak XOR, araştırmacılar hesaplamanın doğrusal eşik modelinin sınır ve gücünü araştırması üzerine yoğunlaşmışlardır. Bu görev şaşırtıcı olarak zor olmuştur. Gerçekten alandaki yalnız güçlü alt sınır polinom ağırlık gibi küçük LT elementlerin polinom boyut devrelerinin iki katmanlı olarak uygulanabileceği fonksiyonların sınıfı LT_2 ile ilgilidir. Diğer bir deyimle LT_2 gibi keyfi ağırlıklarda LT elementlerinin kullanımına izin verilirse LT_2 'de olmayan bir fonksiyon bulunabilir, o zaman alt sınır olmaz, keyfi ağırlıklarda LT devrelerinin polinom boyutta devreleriyle iki katmanlı uygulama yapılabilecek fonksiyon yoktur.

Sinir mantığının teorik ve pratik bakış açıları arasında çok küçük bir ilişki vardır. Bu tezin amacı teorik ve uygulama arasındaki bu boşluğu azaltmaktır. Bu bölümün geri kalanında tezde ifade edilen ana fikri adresleyeceğiz. Bölüm 1.1'de LT modellerini kullanarak ortak doğru yanlışı fonksiyonlarını uygulama yapan örnekleri sunacağız ve doğrusal eşik fonksiyonlarını sınıflayan LT'yi tanımlayacağız. Bölüm 1.2 LT elementlerinin ağırlıklarının çalışması üzerine ana fikirleri işlemektedir, bölüm 2, 3 ve 4 de sonuçları anlatmaktadır. Bölüm 1.3.5. ve 6. bölümlerle ilgilidir, VLSI uygulaması kadar iyi olan LT elementinden türetilen yeni bir hesaplama elementi olan LTM'yi anlatmaktadır. Son olarak da bölüm 1.4'te tezin katkılarını özetlemektedir.

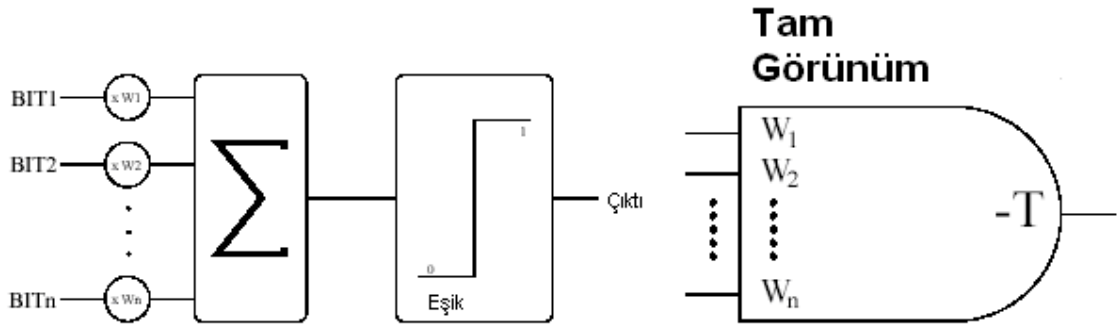
1.1 Tanım ve Örnekler ile LT Fonksiyonu

Bu bölümde doğrusal eşik kapısıyla hesaplanan fonksiyonun resmi bir tanımını vereceğiz. Tek bir LT elementiyle uygulama yapılabilen doğru yanlış fonksiyonları örnekleri göstereceğiz, özellikle VE, VEYA, MAJ ve COMP olarak aşağıda belirtilen hesaplamaları göstereceğiz.

Bu tez doğrusal eşik kapılarından oluşan LT devreleri veya doğrusal eşik devreleri çalışması üzerine odaklanmıştır. Bunlar ikili sayı düzeninde girdiler ve çıktılar içermektedir. Doğrusal eşik fonksiyonlarıyla matematiksel olarak anlatılmaktadır.

Tanım 1.1 (Doğrusal Eşik Fonksiyonu)

n - değişkenli doğrusal eşik fonksiyonu şu tipte bir doğru yanlış fonksiyonudur:
 $f: \{0, 1\}^n \rightarrow \{0, 1\}$



Şekil 1.1: Doğrusal Eşik Elementi $y = \text{sgn}(-t + \sum_{i=1}^n w_i x_i)$.
Herhangi $x \in \{0, 1\}^n$ için

$$f(x) = \text{sgn}(F(x)) = \begin{cases} 1 & F(x) \geq 0 \\ 0 & \text{Diğer} \end{cases}$$

$$F(x) = w_0 + \sum_{i=1}^n w_i x_i$$

$w \in \mathbb{R}^{n+1}$ ile sabitlenerek yazılabilecektir. Şekil 1.1 fikri göstermektedir. Aşağıdaki örnekleri düşünün.

Örnek 1.1 (OR'un LT Gösterimi)

Basit bir doğru yanlış fonksiyonu

$$OR(X_1, \dots, X_n) = \begin{cases} 0 & \text{if } (X_1, \dots, X_n) = (0, \dots, 0) \\ 1 & \text{Diger} \end{cases}$$

n-değişkenden oluşan OR bağlantısıdır.

Eşik kapısıyla uygulanabilir, her n için

$$\forall x \in \{0,1\}^n, OR(x_1, \dots, x_n) = \text{sgn}(w_0 + \sum_{i=1}^n w_i x_i)$$

(w₀, ..., w_n) ağırlık vektörü bulunmaktadır.

OR'u implement etmek için birim ağırlıklara ve w₀ değeri -1 olan bir eşik değere ihtiyaç vardır.

$$W=(-1, 1, \dots, 1)$$

X_1	X_2	$-1 + X_1 + X_2$	$\text{sgn}(-1 + x_1 + x_2)$	$OR(x_1, x_2)$
0	0	-1	0	0
0	1	0	1	1
1	0	0	1	1
0	0	1	1	1

Tablo 1.1: 2-değişkenli bağlantı, $OR(x_1, x_2) = \text{sgn}(-1 + x_1 + x_2)$

X_1	X_2	$-2 + x_1 + x_2$	$\text{sgn}(-2 + x_1 + x_2)$	$AND(x_1, x_2)$
0	0	-2	0	0
0	1	-1	0	0
1	0	-1	0	0
0	0	0	1	1

Tablo 1.2: 2-değişkenli ayrılma, $AND(x_1, x_2) = \text{sgn}(-2 + x_1 + x_2)$

$$OR(X_1, \dots, X_n) = \text{sgn}(-1 + \sum_{i=1}^n x_i)$$

Tablo 1.1 $n = 2$ durumunu göstermektedir.

Örnek 1.2 (AND $\in$ LT)

AND ayrımı da

$$AND(x_1, \dots, x_n) = \text{sgn}(-n + \sum_{i=1}^n x_i)$$

doğrusal bir eşik fonksiyonudur.

Tablo 1.2 $n = 2$ için olan durumu göstermektedir.

Çoğunluk fonksiyonu olan MAJ girdi değişkenlerinin yarısından çoğu 1 ise çıktı olarak 1 veren fonksiyondur.

Örnek 1.3 (MAJ $\in$ LT)

Aşağıda çoğunluk fonksiyonunun tanımı vardır.

$$MAJ(x_1, \dots, x_n) = \begin{cases} 1 & \text{Eger..} \sum_{i=1}^n x_i \geq \left\lceil \frac{n}{2} \right\rceil \\ 0 & \text{Diğer} \end{cases}$$

x_1	x_2	x_3	$-2 + x_1 + x_2 + x_3$	$\text{sgn}(-2 + x_1 + x_2 + x_3)$	$MAJ(x_1, x_2, x_3)$
0	0	0	-2	0	0
0	0	1	-1	0	0
0	1	0	-1	0	0
0	1	1	0	1	1
1	0	0	-1	0	0
1	0	1	0	1	1
1	1	0	0	1	1
1	1	1	1	1	1

Tablo 1.3: 3-değişkenli çoğunluk, $MAJ(x_1, x_2, x_3) = \text{sgn}(-2 + x_1 + x_2 + x_3)$.

Eşik fonksiyonu için doğal bir adaydır, ağırlıkların bir seçimi,

$$w = (-\left\lceil \frac{n}{2} \right\rceil, 1, \dots, 1)$$

$$MAJ(X_1, \dots, X_n) = \text{sgn}\left(-\left\lfloor \frac{n}{2} \right\rfloor + \sum_{i=1}^n x_i\right)$$

tablo 1.3 $n = 3$ olan durumu göstermektedir.

Örnek 1.1, 1.2 ve 1.3 simetrik olanlar için doğru yanlış fonksiyonlarını göstermektedir. Çıktı pozisyonlarından bağımsız olarak girdi vektöründeki 1'lerin sayısına bağlı olan fonksiyonlardır. İyi bilinen simetrik fonksiyon eşitlik ya da XOR fonksiyonudur.

Örnek 1.4 ($XOR \notin LT$)

n-değişkenli eşitlik fonksiyonu

$$XOR(X_1, \dots, X_n) = \begin{cases} 1 & \text{Eger..} \sum_{i=1}^n x_i \text{..tek..ise} \\ 0 & \text{Diğer} \end{cases}$$

şeklindedir.

$n=2$ olsun ve XOR'u implement eden bazı ağırlıklar vardır.

$$w = (w_0, w_1, w_2)$$

$$XOR(X_1, X_2) = \text{sgn}(w_0 + w_1 x_1 + w_2 x_2)$$

X_1	x_2	$-2 + x_1 + x_2 + x_3$	$XOR(x_1, x_2)$	gösterir
0	0	w_0	0	$w_0 < 0$ (1)
0	1	$w_0 + w_2$	1	$w_0 + w_2 > 0$ (2)
1	0	$w_0 + w_1$	1	$w_0 + w_1 > 0$ (3)
1	1	$w_0 + w_1 + w_2$	0	$w_0 + w_1 + w_2 < 0$ (4)

Tablo 1.4: 2-değişkenli eşitlik, $XOR(x_1, x_2) \neq \text{sgn}(w_0 + w_1 x_1 + w_2 x_2)$

Tablo 1.4 x değıştikçe $F(x)$ 'in değerlerini göstermektedir. Fonksiyonun değeri w_i 'nin çözümünün olmadığı eşitsizli sistemi üretmektedir. Gerçekten eşitlik (1) + (4) $2 w_0 + w_1 + w_2 + w_3 < 0$ iken, eşitlik (2) + (3) $2 w_0 + w_1 + w_2 + w_3 > 0$ değeri üretmektedir. O nedenle $n = 2$ için $XOR \notin LT$. Herhangi n için de aynı durum geçerlidir. Gerçekten herhangi n için $XOR \in LT$ düşünün ve XOR fonksiyonu

$$XOR(X_1, X_2) = \text{sgn}(w_0 + \sum_{i=1}^n w_i x_i)$$

şeklindedir.

O zaman

$$XOR(X_1, X_2, 0, \dots, 0) = \text{sgn}(w_0 + w_1 x_1 + w_2 x_2)$$

ancak $n=2$ için $XOR(x_1, x_2, 0, \dots, 0) = XOR(x_1, x_2)$, $XOR \in LT$, doğru olmayacak şekilde.

Doğrusal eşik fonksiyonları için de simetrik tüm fonksiyonlar için $w_1=w_2=\dots=w_n$. Bu durum oldukça uygun çünkü ağırlıkların tümü 1'e eşitlenebilir. Alttaki doğru yanlış fonksiyonları simetrik olmazsa ne olur? Ağırlıklar ne kadar büyük olabilir? Takip eden örnek ağırlıkları farklı olmayı gerektiren bir LT fonksiyonunu göstermektedir ve böylece diğerleriyle kıyaslandığında bazılarının büyük olmasını gerektiriyor. Gerçekte ağırlıklar girdi değişkenlerinin sayısı ile üslü olarak büyümektedir.

Örnek 1.5 ($COMP \in LT$)

Karşılaştırma fonksiyonu X ve Y olan ve ikilik sistemde gösterimleri $(x_1, \dots, x_n)$ ve $(y_1, \dots, y_n)$ olan iki tamsayı kabul etmektedir. Birinci değişken olan X

$$X = \sum_{i=1}^n 2^{i-1} x_i$$

iken ikinci değişken Y

$$Y = \sum_{i=1}^n 2^{i-1} y_i$$

şeklindedir ve onları kıyaslamaktadır. Ve her ikisini

$$COMP(x_1, \dots, x_n, y_1, \dots, y_n) = \begin{cases} 1 & \text{eger } X > Y \\ 0 & \text{diger} \end{cases}$$

fonksiyonu ile karşılaştırıyor. $COMP$ 'un LT uygulaması

$$COMP(x_1, \dots, x_n, y_1, \dots, y_n) = \text{sgn}(\sum_{i=1}^n 2^{i-1} x_i - \sum_{i=1}^n 2^{i-1} y_i)$$

yukarıda tanımlı olduğu gibidir. Aşağıdaki

$$w = (0, 1, 2, \dots, 2^n, -1, -2, \dots, -2^n)$$

ağırlık vektörüne dönüştürmektedir.

1.2 Doğrusal Eşik Elementinin Ağırlıkları

Bu bölümde LT elementlerinin ağırlıklarına ilişkin ana konularına değineceğiz. Farklı ağırlıktaki kümelerin aynı LT fonksiyonunu türetebileceğini göstereceğiz ve asgari ağırlıkları tanımlayacağız. 2, 3 ve 4. bölümlerin konuları tanıtıldı, sırasıyla ağırlıkları sınırlama, asgari ağırlıklarla fonksiyonlar kurma ve büyük ağırlıktaki tek bir elementi düşük ağırlıklı kapılardan oluşan bir devreye dönüştürme. Doğrusal eşik elementi gibi bir elementin verim hesabı nasıl yapılır? Tek bir LT kapısı farklı doğru yanlış fonksiyonlarının kalabalığını uygulama yapabilir. Bu ağırlıklarını değiştirerek yapılabilir. Gerçekten her bir ağırlık seçimi ayrı bir fonksiyona tekabül etmektedir. Bazı farklı ağırlıktaki kümeler aynı doğru yanlış fonksiyonunu üretirken genel olarak ağırlıklar için iki farklı seçim iki farklı fonksiyon vermektedir. N girdili bir LT elementi 2^n farklı doğru yanlış fonksiyonu uygulama edebilir, bölüm 2.3.1'de bu durumu göreceğiz. LT'nin AON'a göre karşılaştırılmış bu ilave gücü ilave bir karışıklığı beraberinde getirmektedir. Bu noktada şu soru sorulabilir: doğrusal bir eşik kapısının bilgi içeriği nedir özellikle bu bilgiyi depolamak için ne kadar bit gerekmektedir?

Tek bir LT elementinin ağırlıkları üzerine odaklanalım. Şunu not edin, bir I fonksiyonu olsun, ağırlık vektörü w benzersiz olmasın. Farklı ağırlıklar aynı fonksiyonu uygulama yapsın.

Örnek 1.6 (Ağırlıkları düşürmek)

Aşağıdaki f fonksiyonu

$$f(x_1, \dots, x_4) = \text{sgn}(2 - 4x_1 + 6x_2 - 2x_3 + 4x_4)$$

aşağıdaki gibi yazılabilir:

$$f(x_1, \dots, x_4) = \text{sgn}(1 - 2x_1 + 3x_2 - x_3 + 2x_4)$$

Çünkü $\text{sgn}(2a) = \text{sgn}(a)$ her $a > 0$ için geçerlidir. Şu fonksiyonu düşünün:

$$f(x_1, \dots, x_5) = \text{sgn}(2 - 4x_1 + 6x_2 - 2x_3 + 4x_4 + x_5).$$

x_5 'e bağlı olmadığı için aşağıdaki gibi yazılabilir.

$$f(x_1, \dots, x_5) = \text{sgn}(1 - 2x_1 + 3x_2 - x_3 + 2x_4).$$

Çünkü $2 - 4x_1 + 6x_2 - 2x_3 + 4x_4$ ikinin katı olduğu için ya ≤ 2 ya da ≥ 0 'dır. Her iki durumda da x_5 'i eklemek işareti değiştirmez.

Benzer bir fikir aşağıdaki iki örneğe de

$$f(x_1, \dots, x_4) = \text{sgn}(-4 + x_1 + x_2 + x_3 + 4x_4) = x_4$$

$$f(x_1, \dots, x_3) = \text{sgn}(3 + 2x_1 - x_2 - x_3) = 1$$

uygulanabilir.

Ama genel olarak daha düşük veya en düşük ağırlıklar bulmak daha zor bir problemdir. Ağırlıkları asgari yapmak için

$$f(x_1, \dots, x_3) = \text{sgn}(-1 + 2x_1 - 3x_2 + 4x_3)$$

aşağıdaki fonksiyonunun implement edilmesi gerekmektedir.

$$f(x_1, x_2, x_3) = \text{sgn}(\overline{x_1 x_2} + x_3).$$

Aşağıdaki fonksiyonu implement eden asgari ağırlıkları türetmektedir.

$$f(x_1, \dots, x_3) = \text{sgn}(-1 + x_1 - x_2 + 2x_3).$$

Örnek eşitlik 1.6 aynı LT elementini implement etmek için farklı ağırlıktaki vektörler kullanılabileceğini göstermektedir. Bu düşüncenin resmi bir tanımı aşağıdaki gibidir.

Tanım 1.2 (Ağırlık Uzayı)

F eşik fonksiyonu verilmiş olsun, Tanım 1.1'i sağlayan tüm ağırlıkların kümesini

$$W = \left\{ W \in R^n : \forall x \in \{0,1\}^n, \text{sgn}(w_0 + \sum_{i=1}^n w_i x_i) = f(x) \right\}$$

W olarak tanımlayalım.

Ağırlıkları çalışmak istiyoruz, özellikle aşağıdaki ortogonal sorular üzerinde duruyoruz.

1. Eğer ağırlıklar sadece tamsayılar olsun şeklinde sınırlandırılırsa ne olur ya da genel olarak ağırlıklar R'nin herhangi bir alt kümesine sınırlandırılacak olursa?

2. Ağırlıkların tam sayı olduğunu ve boyutlarının verildiğini varsayarsak, en düşük ağırlık vektörünü nasıl bulunur?

3. Verilen fonksiyonda, f, minimum ağırlığın büyük olduğunu faz edersek, örneğin girdilerin sayısı ile katlanarak artıyorsa bu fonksiyonu küçük ağırlıklı girişlerden oluşmuş iki-katlı LT devresiyle tamamlayabilir miyiz?

1970'lerin başlarında gösterilmiştir ki herhangi bir LT fonksiyonu tamsayı olarak işaretlenen ağırlık ile tamamlanabilir. Burada müspet olamayan bir ispat yapılmıştır yani tamsayı olan ağırlığı bulmadan tamsayı ağırlıklar kümesinin var olduğu gösterilmiştir. Bölüm 2'de şu soruları soruyoruz: Sonuç elemanının kuvvetini etkilemeden verilen sayı kümesinde isteğe bağlı bir değer olarak ağırlıklar nasıl kısıtlanabilir? Bu sayı kümesinin koşulları nelerdir? Etkili bir dönüştürme algoritması var mıdır?

Bölüm 3 ve Bölüm 4'te ise tamsayı ağırlıkların boyutuna değinilmektedir. Ağırlıkların boyutunun ölçülmesi için L_1 normu kullanıyoruz.

Tanım 1.3 (En Düşük Ağırlık Boyutu)

Bir ağırlık vektörünün boyutunu ağırlığın mutlak değerlerinin toplamı olarak tanımlıyoruz.

Minimum ağırlık boyutunun doğrusal eşik fonksiyonu

$$S[f] = \min_{w \in W} \left(\sum_{i=0}^n |w_i| \right)$$

şeklinde tanımlanır.

En düşük değeri sağlayan belirli vektöre en düşük ağırlık vektörü denir.

Doğal olarak, $S[f]$ n'in fonksiyonudur.

Çeşitli bilgilerin aynı anda işleme tabi tutulduğu bilgisayar işlev türlerinde yapılan birçok deneyin sonucu göstermiştir ki doğrusal eşik değerlerindeki katsayıların değerleri, girdilerin boyutlarıyla birlikte çok hızlı bir şekilde artmıştır. Bu nedenle bilgisayar işlev türlerinin pratikte kullanımı sınırlanır ve bu durumu bir olağan soru takip eder: Eğer biri katsayılarında sadece “küçük” artışları olan eşik değerleri için kendini sınırlıyorsa bu bilgisayar işlev türlerinin bilişimsel gücü nasıl sınırlanabilir? Katlanarak artan ağırlıklı tek bir eşik değeri tarafından tamamlanabilen fakat polinom olarak artan ağırlıklı bir eşik değeri tarafından tamamlanamayan fonksiyon yani sayfa [17], [33], [38], [43]'te gösterilmiştir. Bu fonksiyon $COMP$ 'tur. Örnek 1.5.'te benzer fonksiyon gösterilmiştir. Bu sonuçların ışığında LT nin altsınıfları küçük ağırlıklı fonksiyon sınıfı: LT olarak tanımlanmıştır. n 'in katlanarak artması ve n 'in polinom olarak artması için küçük ve büyük değer girdi sayılarında kalır. LT nasıl bir yoğunluktur, başka bir deyişle küçük ağırlıklı fark edilmeyen fakat polinom derecesi d olan 11 gibi artan ağırlıklar ile tamamlanabilen fonksiyonlar var mıdır? LT ağırlıklarının polinomunun derecesinin artışına endeksli fonksiyon sınıflarına bölünebilir mi? Bölüm 3'te verilen ağırlık boyutlarında LT fonksiyonlarının algoritmik oluşumu gösterilerek bu sorular cevaplanacaktır.

İstenen büyük ağırlıklı LT fonksiyonlarına nasıl değinilebilir? Siu ve Bruck ([43]), $LT_1 \subset L\hat{T}_2$ 'i göstererek ve isteğe bağlı derinliği genelleştirerek $LT_d \subset L\hat{T}_{d+1}$ [13] , $LT_d \subset L\hat{T}_{d+1}$ sınırında geliştirilebileceğini kanıtlamıştır. Metot her ne kadar karmaşık olsa da ve [18]'de ispatı takip etmek güçleşse de [13]'ün sonuçları olan basitleştirilmiş versiyon sunuluyor. Burada $LT_1 \subset L\hat{T}_2$ 'nin gösterilmesine odaklanılıyor. Ağırlığı düşürmek için kullanılan iki yöntem ikinin kuvveti olarak bölmek ve prime modüle göre bölmektir. Eğer yeteri kadar prime kullanılıyorsa ”küçük”- ağırlık girişleri doğru çıktıyı üreten devre ile birleştirilir. Belirli büyük ağırlık fonksiyonunun benzetimini kısıtlayarak sayfa [13] ve [18]'de gösterilen sonuçları daha basitleştirilebilir: $COMP$. Netice itibarıyla sayfa [18]'deki $O(n^{12} \log^{11} n)$ 'un genel sınırları üzerindeki önemli gelişme olan $O(n^4 \log n)$ düzenindeki devrenin kapılarının sayıları sınırlandırılır.

1.3 Çoklu Eşik ve VLSI Uygulaması

Bu kısımda bölüm 6'da tanımlanacak olan donanım uygulamalarının sonuçlarını ana hatlarıyla inceleyecek ve bölüm 5'deki LT yle ilgili yeni bir fonksiyon sınıfı olan LMT gösterilir.

60'larda ve 70'lerde ortaya atılan eşik devreleri uygulamaları sayfa [4], [48], [53] ve daha yakın sayfalarda [28], [39]'dur. Bilgi birikimimiz için eşik devrelerindeki teorik sonuçlar, silikonlu uygulamaları içeren işlere eklenmiyor. Programlanabilir nötron bazlı donanımlarda sayfa [39], [41] öneriliyor.

LT uygulamalarının özellikle birbirinden ayrıldığı iki nokta var: Ağırlıklı toplamı hesaplamak için kullanılan metot ve ağırlığın saklanması için kullanılan usul. Her biri ağırlıklı girdilere denk akımlar seçildi. Buna göre doğru yanlış girdileri

kullanıldı ve kesin çarpmaya gerek duyulmadı.

Girdi mantıksal 0 ve akım w_i mantıksal 1 olduğunda sıfır akım üretildiğinden emin olunur. Böyle bir “çarpım” girdi pimi giriş terminaliyle bağlantılı olan tek bir transistor tarafından yapılabilir. Ağırlığı depolama ve girdileri ölçeklendirme tek bir transistor de birleştirilmiş olması bu şekilde bir yaklaşımın avantajıdır. Fakat ağırlıklar nasıl tamamen depolanıyor? Bu problemi iki yönden ele alabiliriz: Fiziksel bağlantılı ve programlanabilir ağırlıklar. Fiziksel bağlantılı Ağırlıklar devrenin planlandığı anda tanımlanır ve bir kere monte edildikten sonra değiştirilemez. Fiziksel bağlantılı ağırlıklar ile ilgili birçok ilginç soru vardır. Gerçekten çoğu uygulamalarda farklı ağırlıklar farklı planlara karşılık gelirler. Bu farklılıklar LT devresi planlamayı zor bir iş haline getirir. Çünkü farklı elemanların farklı şekilleri vardır. Ancak önceki bölümde gördüğümüz üzere hesaplanan fonksiyonu etkilemeden LT elemanının ağırlığı değiştirilebilir. Bu son faktör elemanların birbirine tamamıyla uyabilecek şekilde planlanmasına yardımcı olur. Öte yandan programlanabilir ağırlıklar böyle zorluklar çıkarmaz. Tüm LT elemanları benzer görünür. Programlanabilir ağırlıkların birçok uygulama şekli vardır. Dijital RAM olarak depolanabilir ya da girdi hattından beslenebilir. Bölüm 6’da LT elemanının iki uygulama yolu gösteriliyor. Birincisinde transistorun boy en oranında depolayan fiziksel bağlantılı ağırlıklar kullanılıyor diğerinde ise geçici olamayan yük olarak dalgalı giriş transistoru yerleştirilmiş programlanabilir ağırlıklar kullanılıyor. İkinci durumda ağırlığın değeri tünelin açılmasına ya da sıcak elektron püskürmesine bağlı olarak değişebilir. Eşitlik gibi genel $LT2$ fonksiyonu için bir LT devresi planlandığında LT gösteriminin gerekenden fazla olduğunun farkına varılabilir. Genelde $LT2$ devresi farklı eşik elemanlarından oluştuğunda, eşitlik, toplama ve çarpma gibi bazı kullanışlı fonksiyonlar halinde birinci katmanın girişi eşikleri açısından farklı olurlar. Bu faktörün avantajlarından yararlanmak için bölüm 5’e giriş yapmak gerekir. *LTM, çarpım eşikleri olan doğrusal eşik elemanı*, yeni bir hesaplama elemanıdır ve şu hesaplama da kullanılır: Doğru yanlış girdilerin ağırlıklı toplamının tek bir eşikle karşılaştırmak yerine eşik kümeleriyle karşılaştırıla bilinir. Geometrik olarak ikiye bölünmüş hiper küp şeklinde kullanılan paralel hiper düzlemler kümesi olarak görülebilir.

Tanım 1.4 (Çoklu Bağlantılı Doğrusal Eşik Girişi – LTM)

Fonksiyon f LTM’ dedir eğer ağırlıklar kümesi $w_i \in Z, 1 \leq i \leq n$ ve a var ise

Fonksiyon $h : Z \rightarrow \{0,1\}$ öyle ki

$$f(x) = h\left(\sum_{i=1}^n w_i x_i\right) \text{ her } x \in \{0,1\}^n \text{ için.}$$

Sadece h ’de kısıtlanır öyle ki polinom olarak girdileri tarayacak birçok $\left[-\sum_{i=1}^n |w_i|, \sum_{i=1}^n |w_i| \right]$ bağlantılara uğrar.

Fark ediliyor ki bağlantı sayılarındaki kısıtlama olmadan LMT girişi hiçbir doğru yanlış fonksiyonunu hesaplayabilir durumda değildir. Aslında verilen keyfi fonksiyon, f , kümesi $w_i = 2^{i-1}$ ve $h\left(\sum_{i=1}^n 2^{i-1} x_i\right) = f(x_1, \dots, x_n)$.

Şimdiki örnekte tek LTM elementi ile nasıl eşitlik fonksiyonunu, XOR ,

hesaplayabileceğimiz gösterilecektir. Örnek 1.4'te XOR' yi hesaplamak için tek bir LT elemanının yeterli olamadığı gösterilmişti.

Örnek 1.7 ($XOR \in LTM$)

$XOR(X)$ çıktıları 1 eğer $|X|$, X 'deki 1'lerin sayıları tektir. Diğer türlü çıktıları 0 olur. Uygulamak için $0 \leq k \leq n$ aralığında $w_i = 1$ ve $h(k) = \frac{1}{2}(1 - (-1)^k)$ seçilir. $h(k)$ 'nin $k < 0$ ve $k > n$ olarak tanımlanmasına gerek yoktur ve birçok polinom şeklinde bağlantısı vardır.

Diğer bir kullanışlı fonksiyon ise iki tam sayının toplanmasıdır. LTM elemanlarının tek katmanı tarafından hesaplanır.

Yukarıda da bahsedildiği gibi eşitlik fonksiyonunun planında gerekli olan alanın geliştirilmesi için LTM araştırması pratik düşünceler üzerine geliştirilmiştir. LT teorik çerçeve içinde kalsa da, bu yeni hesaplama elemanı birçok zor problemin çözülmesini sağlar. Tek bir LMT elemanının tek bir LT elemanından daha güçlü olduğunu görmek çok basittir. Fakat LT_2 ya da $L\hat{T}_2$ karşılaştırmasında nasıl güçlü olabilir?

1.4 Tezin Yazımı ve Organizasyonu

Tezin yazımı üç aşamada gerçekleşmiştir:

- Teorik aşamada $L\hat{T}^{(d)}$ ve LTM gibi yeni fonksiyon sınıfları tanımlanmış ve bunların hesaplama güçleri tahmin edilmiştir.
- Algoritmik aşamada tam sayı ağırlıklarda olduğu gibi her hangi bir reel sayılar kümesinde gerçek ağırlığın ağırlığa nasıl dönüşeceği ve minimum ağırlıklı LT fonksiyonlarının nasıl oluşturulacağı gösterilmiştir. Ayrıca COM P fonksiyonunu hesaplayan $L\hat{T}_2$ devresinin oluşturulduğu algoritma gösterilmiştir. Son olarak XOR , ADD , $PRODUCT$ gibi kullanışlı fonksiyonların LTM ile hesaplanması gösterilmiştir.
- Uygulama aşamasında ise LT ve LMT 'in uygulamaları olan $VLSI$ 'nin tasarım, planlama ve test basamakları gösterilmiştir. Ağırlık değerini depolamak için dalgalı giriş kullanılan programlanabilir LT elemanı tasarlanmıştır.

Tez şu şekilde organize edilmiştir: Bölüm 2'de eşik devresi teorisinin bilinen sonuçları gösterilmiştir. Özellikle herhangi bir doğrusal eşik elemanı tam sayı ağırlık ile tamamlanabilir. Makalemiz bu sonucun herhangi bir reel sayılar kümesinde genelleştirilmesidir. Ağırlığın dönüştürüldüğü algoritma boyunca koşullar tamamlanmış LT fonksiyonunun türetilmesine izin verir. Bölüm 3'te minimum ağırlıkta doğrusal eşik fonksiyonlarının oluşturuluş metodu gösterilmiştir. Bu metot d'ye bağlı $L\hat{T}^{(d)}$ sınıflarının birbirinden ayrılması için kullanılır. d tamsayısı verildiğinde $L\hat{T}^{(d)}$ sınıfı ağırlığın $O(n^d)$ ile tamamlayabildiği

fonksiyon kümesi olarak tanımlanır. Bölüm 4'te örnekteki gibi belli başlı sonuçlar gösterilmiştir: Büyük ağırlıklı tek LT elemanı iki katlı devreden oluşmuş $\hat{LT}$ elemanı yani küçük ağırlıklı doğrusal eşik elemanı ile tamamlanabilir. Bölüm 5'te LMT yani çoklu bağlantılı doğrusal eşik girişi tanıtılmıştır. XOR , ADD , $PRODUCT$ 15'den LT 'ye ilgili LTM'nin kuvvetinin tahmini ve türev sınıfları gibi $\hat{LT}$, $L\hat{T}_2$ ve LT_2 gibi kullanılan doğru yanlış fonksiyonlarının yapıları gösterilmiştir. Son olarak bölüm 6'da LT ve LMT 'nin tamamlayıcısı $VLSI$ tanıtılmıştır. Fiziksel bağlantılı ve programlanabilir sonuçlar gösterilmiştir. Ağırlıklar dalgalı girişteki yükte depolanır ve elektron püskürmesi ve tünel açılmasıyla değiştirilir.

Bölüm 2

Ağırlıkların Kısıtlanması

2.1 Giriş

Şimdiki bölümde tek LT elemanının ağırlığı ile ilgili sorulara odaklanılıyor. Verilen herhangi bir n tane değişkenli eşik fonksiyonunda ağırlığı depolamak için kaç tane bite ihtiyaç vardır. Bu sorunun cevabı herhangi bir LT fonksiyonunun tam sayı ağırlıklar ile tamamlanmasının gösterilmesi ve boyutlarının kısıtlanmasının sağlanması ile 1970'lerin başında Murago, sayfa [32], tarafından verilmiştir. Bölüm 2'de bu düşünde şu sorulara cevap verilerek genelleştirilecektir:

- Ağırlıkların büyüklüğü tam sayılar yerine tamsayıların kareleriyle sınırlandırılırsa ne olur?
- Eğer yalnızca 2'nin kuvvetlerine izin verilirse ne olur?
- Genelde D , pozitif reel sayıların alt kümesi, $D \in \mathbb{R}^+$ ağırlıkların büyüklüğü D 'den olan LT fonksiyon seti $LT[D]$ olarak tanımlanır.

$$LT[D] = \{f : f(x_1, \dots, x_n) = \text{sgn}(w_0 + \sum_{i=1}^n w_i x_i) \mid w_i \in D \text{ olduğu yerde.}$$

$LT[D] = LT$ 'deki D koşulları nelerdir? (örneğin tüm LT fonksiyonlarını tamamlamak için büyüklüğü D 'den olan ağırlıklar yeterlidir.

Bölüm 2.2'de bu tip soruların bazı motivasyonları gösterilecektir. 2.3'te ise şu başlıklar altında toplanabilecek ispatlar ve örnekler verilecektir:

- n değişkenli kaç tane LT fonksiyonu vardır?
- Herhangi bir eşik fonksiyonunda ağırlıkların depo edilebilmesi için gerekli bitlerin üst sınırı nedir?
- Tamamlanan fonksiyonu değiştirmeden her hangi bir reel ağırlığı nasıl bir tam sayıya dönüştürebiliriz?

Bölüm 2.5'te ise temel sonuç sunulacaktır: Tamamlanabilen LT fonksiyonları tarafından garanti edilen D kümesinin koşulları.

2.2 Motivasyon

Eşik devrelerine değinildiği zaman, genellikle mesele belirli ağırlık değerinin aynı girişte ya da değişik girişlerde farklı bölgelerde görülmesidir. Ağırlık değerini depolamanın pahalı olduğu verilen sistemde öncelikle değer depolanmak istenir ve aynı değeri birçok bölgede depolamak yerine ilişkin ağırlıklar ile bağlanır. Bu

kavram eşik devrelerinin uygulamalarındaki hem donanım hem de yazılımda tatbik edilebilir. Öğrenilmiş algoritmaların sonuçları ya da verilen fonksiyonda basit bir ön hesaplama olarak verilmiş ağırlıklar için yukarıdaki yaklaşım edilebilir. Yalnızca çift ağırlıklar değerleri elenir. Daha fazla çift ağırlıklar ortaya çıkartmak için ilgili eşik fonksiyonlarını etkilemeden değiştirilmiş belli ağırlıkların depolama yerlerini muhafaza etme amacımıza ulaşmamıza bir basamak daha kaldı. Bölüm 1.2’de farklı ağırlıklar kümeleri aynı eşik fonksiyonu ile tamamlanabildiği için yukarıdaki durumun yapılabildiğini gördük. Bu bağlamda şu soru sorulabilir: Verilen reel sayılar kümesi tüm LT fonksiyonlarının gösterilmesi için yeterli midir?

2.3 Taslaklar ve ilgili çalışmalar

Farklı bir ağırlık aynı LT fonksiyonu ile tamamlanabilir. Ağırlıklar kümesini karakterize etmenin bir yolu ağırlığın sınırları olarak neyi tanımladığımızdır.

Tanım 2.1 (Ağırlığın sınırları)

$(w_0, \dots, w_n)$ ağırlığın seti olsun ve f de bunların tamamladığı fonksiyon olsun. Kümenin sınır çifti

$$l = \max_{x|f(x)=0} (w_0 + \sum_{i=1}^n w_i x_i)$$

ve

$$h = \max_{x|f(x)=1} (w_0 + \sum_{i=1}^n w_i x_i)$$

olsun.

Burada (l, h) sınırın iki açık özellik

- $l < 0$ ve $h \geq 0$
- her $x \in \{0, 1\}^n$ için, $w_0 + \sum_{i=1}^n w_i x_i \notin]l, h[$

görülmektedir.

Aşağıda gösterilen ispatı yapmak için $(-1, 1)$ sınırlarına ihtiyacımız olacak. Verilen herhangi bir ağırlıklar setini algoritma 2.1 şöyle bir sete çevirir.

Algoritma 2.1 $(-1, 1)$ sınırı

$(u_0, \dots, u_n)$ kümesi (l, h) sınırı ile veriliyor ve

$$w_0 = \frac{2}{h-l} (u_0 - \frac{h+l}{2})$$

$$w_i = \frac{2u_i}{h-l} \text{ her } i \text{ için, } 1 \leq i \leq n$$

algoritma 2.1'in geçerli ağırlıkları ürettiğini gösterelim öyle ki orjinal olanlarla aynı değerleri ürettiğini ve sınırlarının $(-1,1)$ aralığında olduğunu gösterelim.

Yardımcı Teorem 2.1 (Ağırlıkların çevirimi)

$(u_0, \dots, u_n)$ (h,l) sınırlı ağırlıkların keyfi bir kümesi olsun ve f ifade ettiği fonksiyon olsun. Algoritma 2.1 ile elde edilen ve $(-1,1)$ sınırında bulunan fonksiyon olsun ve f fonksiyonunu ifade etsin.

Kanıt:

$$f(x) = \text{sgn}(u_0 + \sum_{i=1}^n u_i x_i) \quad g(x) = \text{sgn}(w_0 + \sum_{i=1}^n w_i x_i)$$

$f(x) = \text{sgn}(u_0 + \sum_{i=1}^n u_i x_i)$ ve g fonksiyonu yeni ağırlıklarla ifade edilen $(w_0, \dots, w_n)$, $g(x) = \text{sgn}(w_0 + \sum_{i=1}^n w_i x_i)$. Her x için $g(x)=f(x)$ olduğunu göstereceğiz. İki duruma bakacağız:

$f(x) = 0$ olsun. Tanım 2.1 ile sınır (l,h) ile ve $h-l>0$ ile sınır

$$u_0 + \sum_{i=1}^n u_i x_i \leq l$$

şeklindedir. Eşitliğin her iki tarafına aşağıdaki yöntemi

$$u_0 - \frac{h+l}{2} + \sum_{i=1}^n u_i x_i \leq \frac{l-h}{2}$$

Uygularsak ve daha sonra $\frac{2}{l-h}$ ile

$$\frac{2}{h-l} (u_0 - \frac{h+l}{2} + \sum_{i=1}^n u_i x_i) \leq \frac{2}{h-l} (\frac{l-h}{2})$$

çarparsak,

$$w_0 + \sum_{i=1}^n w_i x_i \leq -1$$

sonucuna ulaşırız.

$$g(x)=0$$

- $f(x)=1$ yapan x için aynı işlemleri tekrarlırsak

$$u_0 + \sum_{i=1}^n u_i x_i \geq h$$

$$u_0 - \frac{h+l}{2} + \sum_{i=1}^n u_i x_i \leq \frac{h-l}{2}$$

$$\frac{2}{h-l} (u_0 - \frac{h+l}{2} + \sum_{i=1}^n u_i x_i) \leq \frac{2}{h-l} (\frac{h-l}{2})$$

$$w_0 + \sum_{i=1}^n w_i x_i \leq 1$$

elde ederiz.

$$g(x)=1$$

ve $g=f$ olduğunu gösterdik ve öyle $|w_0 + \sum_{i=1}^n w_i x_i| \geq 1$ bir eşitsizlik olan $u_0 + \sum_{i=1}^n u_i x_i$ noktalarında h veya l 'ye eşit olan bir eşitlik olduğunu gösterdik.

Bu durumu bir örneklerle gösterelim.

Örnek 2.1 $((-1,1)$ Sınırı)
2-değişkenli LT fonksiyonunu düşünelim.

$$f(x_1, x_2) = \text{sgn}(-1.2 + 0.5x_1 + 1.1x_2)$$

Ağırlık vektörü $(-1.2, 0.5, 1.1)$. Sınırı olan (l, h) 'yi hesaplayalım. Ağırlıklı toplam aşağıdaki değerleri tahmin etmektedir.

x_1	x_2	$-1.2 + 0.5 x_1 + 1.1 x_2$	$f(x_1, x_2)$
0	0	-1.2	0

0	1	-0.1	0
1	0	-0.7	0
1	1	0.4	1

Tanım 2.1'e başvurarak

$$l = \max_{x|f(x)=0} (-1.2 + 0.5x_1 + 1.1x_2) = -0.1$$

$$h = \max_{x|f(x)=0} (-1.2 + 0.5x_1 + 1.1x_2) = -0.4$$

L ve h'yi elde ederiz.

Algoritma 2.1 ile yeni ağırlıklar

$$w_0 \frac{2}{h-l} (u_0 - \frac{h+l}{2}) = \frac{2}{0.4 - (-0.1)} (1.2 - \frac{0.4 + (-0.1)}{2}) = 5.4$$

$$w_1 = \frac{2x0.5}{0.4 - (-0.1)} = 2.0$$

$$w_2 = \frac{2x1.1}{0.4 - (-0.1)} = 4.4$$

şeklindedir. Yeni ağırlık vektörü aşağıdaki değerleri tahmin etmektedir.

x_1	x_2	$-5.4 + 2.0x_1 + 4.4x_2$	$\text{sgn}(-5.4 + 2.0x_1 + 4.4x_2)$
0	0	-5.4	0
0	1	-1.0	0
1	0	-2.4	0
1	1	1.0	1

Beklendiği gibi yeni ağırlıklar aynı fonksiyonu ifade etmektedir ve sınırı (-1,1)'dir.

Tek elemanlı LT'nin çalışmasıyla ilgili iyi bilinen bazı sorunları belirtelim.

2.3.1 n-Değişkenli $0(2^n)$ LT Elemanı Vardır

n tane değişken verildiğinde toplam doğru yanlış fonksiyonunun 2^{2^n} olduğunu belirlemek kolaydır. Ayrıca genel doğru yanlış fonksiyonu 2^n çiftli girişli gerçek tablosu tarafından özel olarak belirlenir. Bu fonksiyonların kaç tanesi gerçekten eşik fonksiyonlarıdır? Bu soru 1950'lerde birçok yazar tarafından düşünülmüştür. Şu sınırlama sayfa [36] 'da türetilmiştir.

$$|LT| < 2^{n^2}$$

$|LT|$ n tane değişkenli eşik fonksiyonları için sabit kalır. Daha sonra 1850'de L. Schlafli, sayfa [42], tarafından $|LT|$ deki en iyi üst sınırı

$$|LT| < 2 \sum_{i=0}^n \binom{2^n - 1}{i} = 2^{n^2 - n \log_2 n + O(n)}$$

olarak bulunmuştur.

Sayfa [54] 'de $|LT|$ deki ilk alt sınır yayınlandı fakat aynı dergide gösterilmiş benzer bir ispat olan [46] 'nın sunuş tarihi nedeniyle önceliği oldu. Bu alt sınır şudur:

$$|LT| > 2^{\frac{n(n-1)}{2}}$$

Yalnızca 1989'da Zuev tarafından bu sınırlar geliştirildi. Sayfa [34] ve [57] 'deki sonuçların kullanıldığı sayfa [58] görülebilir ve

$$|LT| > 2^{n^2 - \frac{10n^2}{\ln n} - O(n \ln n)}$$

şeklinde bu sınırı tanımlayabiliriz.

Son zamanlarda üst sınır daha da geliştirilmiştir, bunun için sayfa [21] ve [23]'e bakınız:

$$|LT| > 2^{n^2 - n \log_2 n - O(n)}$$

Ve son olarak sayfa [22]'de ispatlandığı gibi

$$|LT| > 2^{n^2 - n \log_2 n - O(n)}$$

olarak gösterilebilir. Sonuç olarak üst ve alt sınır aralığı daraldı. 2^n çevresinde n değişkenli eşik fonksiyonları olduğundan teorik argümanlardaki kullanılabilir. Bu argümanlar en az n^2 olması gereken ağırlığı (eğer bu değer değilse bazı fonksiyonlar diferansiyelleşemiyor.) göstermek için ihtiyaç duyulan toplam bit sayısını belirlemek amacıyla kullanılabilir. Ayrıca 1.2 kısmında gördüğümüz gibi farklı ağırlık vektörleri aynı fonksiyonu tamamlıyor. LT sunumunun depolama açısından en uygun sunum değildir çünkü ağırlığı sunmak için n^2 fazla bite ihtiyaç vardır.

2.4 Reelden Tamsayı Ağırlıklara

Bu bölümde tamsayı ağırlıklarla yazılabilen reel ağırlıklar gibi herhangi bir ağırlık kümesini ile yazılmış fonksiyonu göstereceğiz. İki argüman sunuyoruz:

- Ağırlıkların boyutlarını sınırlamayı sağlayan müspet olmayan, varoluş temelli argüman.
- Ağırlıkların boyutlarını sınırlamayan fakat verilen ağırlık kümesini tam sayıya dönüştüren müspet argüman.

2.4.1 Her Bir $O(n \log_2 n)$ Bit için Ağırlık Sağlayan Eşik Fonksiyonları

Yukarıda bahsedildiği gibi verilen eşik fonksiyonu tahminine göre tek bir ağırlık en az $O(n)$ bit sağlar. Çünkü LT sunumu dağınıktır. Muroga ,sayfa [32] 'nin 1970'lerin başlarında gösterdiği gibi gerçek sayı $O(n \log_2 n)$ 'dur. Tam olarak olamasa da LT sunumu biraz yoğundur; $O(n^2)$ arasındaki $O(n^2 \log_2 n)$ fark daha küçüktür ve eşik izlenerek yapılan ağırlık toplamının kullanılmasıyla uygun hesaplama avantajı elde edilir. Muroga'nın sınır tanımını kavrayabilmek için aşağıda ispatı gösterilmektedir.

Theorem 2.1 (Ağırlıkta $O(n \log_2 n)$ bits) Keyfi bir n -değişkenli LT fonksiyonu için, ağırlık w_i aşağıdaki

$$|w_i| < O(n \log_2 n) \quad \forall 0 \leq i \leq n$$

durumu sağlamaktadır.

Kanıt:

$l, f \in LT$ koşulunu sağlayan bir eşik fonksiyonu olsun. Ağırlıkları $(u_0, \dots, u_n)$ olarak bilinmesin. l 'nin doğruluk tablosundan 2^n adet doğrusal eşitsizlikler çıkartılabilir.

$$\begin{cases} u_0 + u_1 x_1^{(1)} + \dots + u_n x_n^{(1)} \geq 0 \\ \dots \\ \dots \\ u_0 + u_1 x_1^{(2^n)} + \dots + u_n x_n^{(2^n)} < 0 \end{cases}$$

$\{x^{(k)}\}_{k=1}^{2^n}$ hiper küpün üst noktalarıdır ve eşitsizliğin yönü fonksiyonun ilgili noktadaki değerine (alt tercih ihtiyaridir.) bağlıdır. (l, h) ağırlıkların $(u_0, \dots, u_n)$ sınırlarıdır ve $(-1, 1)$ sınırlarında yeni ağırlıkların $(u_0, \dots, u_n)$ elde edilmesi için algoritma 2.1 uygulanır. Eşitsizlik sistemi

$$\begin{cases} w_0 + w_1 x_1^{(1)} + \dots + w_n x_n^{(1)} \geq 1 = 2f(x^{(1)}) - 1 \\ \dots \\ \dots \\ w_0 + w_1 x_1^{(2^n)} + \dots + w_n x_n^{(2^n)} < -1 = 2f(x^{(n+1)}) - 1 \end{cases}$$

şekline gelir.

Doğrusal eşitsizlik teorisinde, sayfa [26], [27] , belli başlı sonuçlar kullanılarak; 2^n eşitsizliği dışında $n + 1$ 'in alt kümesinin var olduğunu öne sürebiliriz. Örneğin eğer eşitsizlik işareti eşitlik işareti ile yer değiştirir ise eşitliğin sonuç sisteminin çözümü aynı zamanda eşitsizlik sistemini de çözer. İlgili işaretlenmiş $n + 1$ eşitsizliğinde $\{z^{(k)}\}_{k=1}^{n+1} \{0, 1\}$ noktalarının bir kümesi olsun.

Aşağıdaki 24 eşitlikler sistemini

$$\begin{cases} w_0 + w_1 z_1^{(1)} + \dots + w_n z_n^{(1)} = 2f(z^{(1)}) - 1 \\ \dots \\ \dots \\ w_0 + w_1 z_1^{(n+1)} + \dots + w_n z_n^{(n+1)} = 2f(z^{(n+1)}) - 1 \end{cases}$$

elde ederiz.

Yukarıda da değinildiği gibi eşitliklerin sağ tarafı f fonksiyonuna bağlıdır. Cramer metodu kullanılarak

$$w_i = \frac{\Delta_i}{\Delta}$$

sonuç bulunabilir.

Burada A determinant iken

$$\Delta = \begin{vmatrix} 1 & z_1^{(1)} & \dots & z_n^{(1)} \\ \dots & & & \\ \dots & & & \\ 1 & z_1^{(n+1)} & \dots & z_n^{(n+1)} \end{vmatrix}$$

ve A_i eşitlik sisteminin sağ tarafında i . kolonun değiştirilmesiyle elde edilmektedir.

Fark ediliyor ki

$$\Delta = \begin{vmatrix} 1 & \dots & z_1^{(1)} & \dots & z_{i-1}^{(1)} & f(z^{(1)})-1 & z_{i+1}^{(1)} & \dots & z_n^{(1)} \\ \dots & & & & & & & & \\ \dots & & & & & & & & \\ 1 & \dots & z_1^{(n+1)} & \dots & z_{i-1}^{(n+1)} & f(z^{(n+1)})-1 & z_{i+1}^{(n+1)} & \dots & z_n^{(n+1)} \end{vmatrix}$$

yukarıdaki matrisler 1 s, 0 s ve -1 s 'den oluşmuştur. Böyle bir matrisin determinanı aşağıdaki sınırı

$$\Delta^{(n \times n)} \leq B_n$$

sağlar.

B_n sınırlama sırası aşağıdaki durumu

$$B_{n+1} \geq n B_n$$

sağlamaktadır.

Daha sonra aşağıdaki durumu

$$\Delta^{(nXn)} \leq O(n!)$$

ifade etmektedir.

Burada tamsayı ağırlıkların boyutu üzerinde ilgileniyoruz, dolayısıyla

$$w_i' = \Delta \mid w_i = \frac{|\Delta|}{\Delta} \Delta_i \leq O(n!) = O(2^{n \log_2 n})$$

tek bir ağırlığı depolamak için $O(n \log_2 n)$ bite ihtiyaç vardır.

2.4.2 Her Bir $O(n \log_2 n)$ Bit için Ağırlık Sağlayan Eşik Fonksiyonları

Bölüm 2.4.1’de herhangi bir LT fonksiyonunda boyutlarında yukarıdan sınır koyarak ağırlığı depolamak için $O(n \log n)$ bit’in yeterli olduğunu kanıtladık.

1961’in başlarında her bir $f_2(n)$ bit için ağırlık gibi $O(2^{\frac{n}{2}})$ boyutlarını sağlayan fonksiyon bulundu. Son dönemlerde sayfa [17] ‘de Hastad boyutu $6(n \log_2 n)$ olan gibi her bir ağırlık için $f_1(n \log n)$ ‘u sağlayan fonksiyonu gösterdi. Bölüm 2.3.1’de gördüğümüz gibi her bir ağırlık için en az n bit deposuna ihtiyaç duyulan yaklaşık 2^n LT fonksiyonu vardır. Daha önce de değinildiği üzere LT sunumu depolama açısından en uygun sunum değildir. Bu sunum eşik fonksiyonunu depolamak için ilave bitlerle $\log n$ faktörünü sağlar. Bununla beraber LT sunumu yoğundur ve kompleksliği hesaplamak için ekstra depolanma arayı kapatır. Depolama için LT fonksiyonlarının spektral sunumu en uygun olanıdır. Aslında sayfa [9] ‘da yazar ilk $n + 1$ spektral katsayılarının özel bir şekilde belirlendiğini göstermiştir.

2.4.3 Reel Sayılardan Tamsayılara Çeviren Bir Algoritma

İspatın geçerliliği boyunca süren algoritmayı göstermeden önce, birkaç örneğe bakalım.

Örnek 2.2 (Reel sayılardan tamsayılara) Aşağıdaki fonksiyon verilmiş olsun.

$$f_1(x_1, x_2) = \text{sgn}(-0.5 + 0.2x_1 + 0.3x_2)$$

Tam sayı ağırlıkları elde etmek için açık olan yol bunları 10 faktörü ile çarpmaktır. Olguyu kullanarak

$$f_2(x_1, x_2) = \text{sgn}(-5 + 2x_1 + 3x_2)$$

denklem aşağıdaki gibi

$$f_2(x_1, x_2) = \text{sgn}(-2.35 + \pi x_1 - \sqrt{2} x_2)$$

olsun.

Bu durumda derecelendirme işe yaramıyor çünkü irrasyonel ağırlıklar var. Bu ağırlıklarla uzlaşmanın bir yolu yer fonksiyonlarını, $\lfloor \cdot \rfloor$, kullanmaktır ve

$$f_2(x_1, x_2) = \text{sgn}(-3 + 3x_1 - 2x_2)$$

aşağıdaki fonksiyonda ise

$$f_3(x_1, x_2) = \text{sgn}(-0.5 + 0.1x_1 + \frac{\sqrt{2}}{3}x_2)$$

şeklindedir.

Ne x10 ne de $\lfloor \cdot \rfloor$ doğru cevabı verdi. Doğru cevabı bulmak için çarpma işlemini ve yer fonksiyonunu birlikte uygulamalıyız ve aşağıdaki örnek

$$f_3(x_1, x_2) = \text{sgn}(-5 + x_1 + 4x_2)$$

reelden tamsayı ağırlıklara dönüştürmek için kullanılan algoritmanın arkasında ki ana fikri gösteriyor. Derecelendirme ve her ağırlık için yer uygulama. Algoritmanın işleyebilmesi için herhangi bir ağırlık vektörünün yeterince büyük ve derecelendirilmiş katsayılarının var olduğunun kanıtlanması gerekir.

Algoritma 2.2 (Reelden tamsayı ağırlıklarına) verilen reel ağırlık kümesi $(u_0, \dots, u_n)$.

1. $(-1, 1)$ sınırında yeni ağırlıkları $(v_0, \dots, v_n)$ 'de etmek için algoritma 2.1 'i uygula.
2. $\mathcal{W}_i = \lfloor (n+2)v_i \rfloor$ 'i yerleştir.

Yardımcı Teorem 2.2 (Reelden tamsayı ağırlıklarına)

Algoritma 2.2 'de oluşturulan ağırlıklar orijinal ağırlıklarla aynı fonksiyonu tamamlar. Şöyle ki,

$$f(x) = \text{sgn}(u_0 + \sum_{i=1}^n u_i x_i) = \text{sgn}(w_0 + \sum_{i=1}^n w_i x_i).$$

İspat:

$(u_0, \dots, u_n)$ reel ağırlıklar kümesi olsun. Yeni ağırlıklar kümesini $(v_0, \dots, v_n)$ elde etmek için algoritma 2.1'i uygulayalım Gösterimi basitleştirmek için $(1, x_1, x_2, \dots, x_n)$ genişletilmiş vektörü yerine x diyelim sonuç olarak $v \cdot x = v_0 + \sum_{i=1}^n v_i x_i$. Yardımcı teorem 2.1 'e göre yeni ağırlıklar da aynı fonksiyonu tamamlar ve sınırları $(-1, 1)$.
Örneğin

$$f(x) = \text{sgn}(u \cdot x) = \text{sgn}(v \cdot x)$$

$$|v \cdot x| \geq 1 \text{ her } x \in \{0,1\}^n.$$

Yukarıdaki eşitsizliği k ile çarpıyoruz,

$$|(kw) \cdot x| \geq k \text{ her } x \in \{0,1\}^n$$

$\lfloor kw \rfloor$ vektörü ifade etsin.

$$|(kw) \cdot x - \lfloor kw \rfloor \cdot x + \lfloor kw \rfloor \cdot x| \geq k$$

$$|(kw - \lfloor kw \rfloor) \cdot x + \lfloor kw \rfloor \cdot x| \geq k.$$

Üçgen eşitsizliği ile,

$$|(kw - \lfloor kw \rfloor) \cdot x| + |\lfloor kw \rfloor \cdot x| \geq k$$

$$|\lfloor kw \rfloor \cdot x| \geq k - |(kw - \lfloor kw \rfloor) \cdot x|$$

$$|\lfloor kw \rfloor \cdot x| \geq k - \sqrt{n+1} \|(kw - \lfloor kw \rfloor)\|$$

$$|\lfloor kw \rfloor x| \geq k - \sqrt{n+1} \quad || (1, \dots, 1 ||$$

$$|\lfloor kw \rfloor x| \geq k - \sqrt{n+1} \sqrt{n+1}$$

$$|\lfloor kw \rfloor x| \geq k - n - 1$$

elde ediliyor.

Bu noktada $k = n + 2$ yapıyoruz ve aşağıdaki denklemi

$$|\lfloor (n+2)w \rfloor x| \geq 1$$

elde ediyoruz.

Yeni ağırlıkların orijinal fonksiyonu f tamamladığını nasıl belirtebiliriz? Verilen herhangi bir x girdi vektörü yukarıdaki basamakları tekrarlar, $u.x \leq l$ durum $f(x)=0$ ya da $u.x \geq h$ durum $f(x) = 1$ için ile başlar ve $\lfloor (n+2)w \rfloor x \leq -1$ ya da $\lfloor (n+2)w \rfloor x \geq 1$ türetir. Mutlak değerin ispatta kullanılması iki durumu birlikte tehdit eder.

Aşağıdaki örnekte algoritma 2.2 uygulanıyor.

Örnek 2.3 (Algoritmayı kullanarak)

Örnek 2.1'de kullanılan aynı 2-değişkenli fonksiyonu

$$f(x_1, x_2) = \text{sgn}(-1.2 + 0.5x_1 + 1.1x_2)$$

kullanalım.

Örnek 2.1'de yapılan Algoritma 2.1'i yapmaya ihtiyaç duyuyoruz. Yeni ağırlık vektörü v ,

$$f(x_1, x_2) = \text{sgn}(-5.4 + 2.0x_1 + 4.4x_2).$$

Bu noktada ağırlıkları $n + 2 = 4$ ile çarpıyoruz ve

$$f(x_1, x_2) = \text{sgn}(-22 + 8x_1 + 17x_2)$$

tabanını alıyoruz.

Yeni tamsayı ağırlıkları aşağıdaki tabloyu veriyor.

x_1	x_2	$-22 + 8x_1 + 17x_2$	$\text{sgn}(-22 + 8x_1 + 17x_2)$
0	0	-22	0
0	1	-5	0
1	0	-14	0
1	1	3	1

Beklenildiği gibi fonksiyon değişmedi. Fakat bulunan ağırlıklar biraz büyük. Özellikle asıl fonksiyon $AND(x_1, x_2)$ 'dir ve çok daha küçük eightlerle

$$f(x_1, x_2) = \text{sgn}(-2 + x_1 + x_2)$$

şeklinde tamamlanabilir.

Bölüm 3'te olası en küçük tamsayı ağırlıklar değerini bulma problemine yönelinecektir.

2.5 Ağırlıkların Herhangi Bir Sayı Kümesine Dönüştürülmesi

Ağırlıkların mutlak değerini D kümesi, $D \in \mathbb{R}$ olarak kısıtlanır. D 'ye ağırlıkların tanım kümesi denir.

Tanım 2.2 $LT(D)$ – D tarafından spawned edilen LT fonksiyonları kümesi)
Verilen D 'ye göre, $\mathbb{R}$ 'nin alt kümesi, $LT(D)$ mutlak değeri yalnız D 'den çekilmiş ağırlıklar ile tamamlanabilen LT fonksiyonlarının kümesi olarak tanımlanır.

Resmen

$$LT(D) = \{f \subseteq LT \mid \exists w \in \mathbb{R}^{n+1}, \text{ öyle ki } f(x) = \text{sgn}(w_0 + \sum_{i=1}^n w_i x_i) \forall x \in \{0,1\}^n\}$$

$$\text{ve } |w_i| \in D \mid 0 \leq i \leq n \text{ için.}$$

Hedefimiz D 'nin özelliklerini ve $LT(D)$ üzerindeki etkilerini irdelemektir. Birkaç kesin durumu eleyerek öncelikle D için aday listesini daraltalım.

- D sonsuzdur.

Aslında eğer D sonlu ise yeterince büyük n bulunabilir. Bu sebeple n değişkenli LT fonksiyonu vardır. Bu fonksiyon D 'nin asallığından daha büyüğüne ihtiyaç duyan ayrı ağırlıkların sayısı sebebiyle

D 'den çekilen ağırlıklarla tamamlanamaz. Örneğin *COMPARISON* fonksiyonu, ağırlıkların yarısı ayrık olması için değişken sayısına $n > 2|D|$. diyelim.

- D sayılabilir.

Bölüm 2.3'te belirtildiği gibi $LT(N) = LT$ 'dır. Herhangi bir LT fonksiyonu tam sayı ağırlıkları kullanılarak tamamlanabilir. N 'den daha yüksek asallığı olan bir küme kullanılması ekstra fonksiyonellik sağlamaz. Aslında eğer küme $]0, \varepsilon[$ aralığını içeriyorsa tam sayı ağırlıkları $LT(D) = LT$ 'ye uygun olmak için küçültülebilir. Diğer yandan $D = [100, 101]$, olduğu düşünülürse fonksiyonların kümesi fazla sayıdaki üretimi çok limitlidir. Bu OR ve birkaç yakından ilgili fonksiyon içerir. Genel sayılamayan D kümesi durumunda D 'nin “en iyi” sayılabilen alt kümesine odaklanırsınız.

- D tam anlamıyla sıralanmıştır.

D sayılabilir olduğu için sıralanabilir. Ayrıca tüm elemanları ayrı olmalıdır. Bu alfabenin rolünü oynar.

D sayılabilir ve sıralanabilir olduğu için endekslenebilir.

$$D = \{d_i \mid i \in N\} \text{ ve } d_i \leq d_{i+1} \dots \forall i.$$

Aşağıda “değiştirilmiş yer” fonksiyonu $d(\cdot)$ tanımlanıyor:

$$d: R \rightarrow R$$

$$d: \dots x_i \dots \rightarrow \dots d(x) = d_i$$

$$\text{Öyleki } d_i \leq x \leq d_{i+1}$$

Örnek 2.4 (Kare ve üstel ağırlıklar)

Gördük ki reel ağırlıklar tamsayı ağırlıklara dönüştürülebilir. Ağırlıkları mükemmel karelere dönüştürmek mümkün müdür?

$$d_i = i^2 \text{ her } i \in \mathbb{N} \text{ için.}$$

5-değişkenli AND fonksiyonunu

$$AND(x_1, \dots, x_5) = \begin{cases} 1 & \text{eger } x_1 = \dots = x_5 = 1 \\ 0 & \text{diger} \end{cases}$$

düşününüz.

LT'nin bir fonksiyon olduğunu gördük ve aşağıdaki gibi

$$AND(x_1, \dots, x_5) = \text{sgn}(-5 + x_1 + x_2 + x_3 + x_4 + x_5)$$

yazılabilir.

Tam kare formunda ağırlıklarla

$$AND(x_1, \dots, x_5) = \text{sgn}(-25 + 4x_1 + 4x_2 + 4x_3 + 4x_4 + 9x_5)$$

şeklinde yazılabilir mi?

Ya 2'nin katları şeklinde yazılabilir mi?

$$d_i = 2^i \text{ her } i \in \mathbb{N} \text{ için}$$

$$AND(x_1, \dots, x_5) = \text{sgn}(-8 + x_1 + x_2 + 2x_3 + 2x_4 + 2x_5)$$

eşitliğinden görülüyor ki, örnek 2.4'de d_i 'nin devamının ne olduğunu önemsemeden D 'den çekilen ağırlıklarla AND tamamlanabilir. Bu durum her hangi bir LT fonksiyonu için doğru değildir. Takip eden teorem gösteriyor ki eğer d_i polinom olarak artıyorsa $LT(D) = LT$ fakat bunlar üstel ise bazı LT fonksiyonları tamamlanamaz. 32, eğer $D = O(i^d)$ ise tüm LT fonksiyonları tarafından tamamlanabilirler. Fakat $D = Q(2^{an})$ ise bu durum gerçekleşmez. Nitekim daha genel sonuçları gösteriyoruz. Mesela bazı süper- polinom artışları serbest bırakılır eğer $d_i = O(n^{\log n})$ ise, $LT(D) = LT$.

Teorem 2.2 (Ağırlıkların Kısıtlanması)

Ağırlıklar sıralanmış $D = \{d_i, i \in \mathbb{N}\}$, $D \in \mathbb{R}^+$ kümesi tarafından kısıtlanmış olsun.

1. $LT(D) = LT$ eğer verilen büyük sabit $C \in \mathbb{R}^+$ için herhangi bir $i > i_0$ de i_0 , var ise

$$C(d_{i+1} - d_i) < d_i.$$

2. $LT(D) \subset LT$ eğer $d_i \in \Omega(2^{mi})$ ve

$$\exists(K, \infty, i_0) \text{ Her } i > i_0 \text{ için } K 2^{\infty i} \leq d_i.$$

İspat:

Birinci bölümü gösterelim. (l, h) sınırlarıyla verilen orijinal ağırlık vektörü u , ile algoritma 2.1 'i uygulayarak $(-1, 1)$ sınırlarında yeni ağırlıklar v elde ederiz.

$$|v.x| \geq 1 \text{ her } x \in \{0, 1\}^n \text{ için.}$$

Yukarıdaki eşitsizliği k ile çarpıyoruz,

$$|(kw).x| \geq k \text{ her } x \in \{0, 1\}^n \text{ için}$$

aşağıdaki eşitsizliğe

$$|(kw).x - d(kw).x + d(kw).x| \geq k$$

dönüşür.

Yukarıda tanımlandığı gibi $d(kw)$, $(d(kw_0), \dots, d(kw_n))$ vektörünü gösteriyorken, d fonksiyonu D kümesi için yer fonksiyonu aşağıdaki gibi

$$|kw - d(kw).x + d(kw).x| \geq k$$

genelleştiriliyor.

Üçgen eşitsizliği ile,

$$|kw - d(kw).x| + |d(kw).x| \geq k$$

$$|d(kw).x| \geq k - |kw - d(kw).x|$$

$$|d(kw).x \geq k - \sum_{j=0}^n |k w_j - d(k w_j)|$$

$$|d(kw).x \geq k - (n+1)(d_{i+1} - d_i)$$

i $d(k|w_{max}|) = d_i$ iken, w_{max} ağırlık en büyük mertebeye ulaşır $|kw_i - d(kw_i)|$ D'nin koşullarına göre

$$C(d_{i+1} - d_i) < d_i$$

i_0 vardır.

Tüm $i > i_0$ ve herhangi bir seçim için C'yi $C = w_{max}(n + 1)$ alalım.

$$w_{max}(n+1)(d_{i+1} - d_i) < d_i$$

eşitsizliği ile d fonksiyonunun tanımı ile $d_i < kw_{max}$

$$w_{max}(n+1)(d_{i+1} - d_i) < kw_{max}$$

$$k - (n+1)(d_{i+1} - d_i) > 0 .$$

Daha sonraki eşitsizliği kullanarak aşağıdaki denklem

$$|d(kw).x| > 0$$

ortaya çıkmaktadır.

Yeni ağırlık vektörünün $d(kw)$, orijinal fonksiyon tarafından tamamlandığını göstermek için bu kadar yeterlidir. Teoremin ispatının ikinci bölümüne gelmeden önce iki örneğe bakalım.

Örnek 2.5 (Kareler) D, örnek 2.4'te tanımlanan tam kareler kümesinde tanımlanan

$$D = \{1, 4, 9, 16, 25, \dots\}$$

küme olsun.

Teorem 2.2'ye göre tüm LT fonksiyonları her $C \in \mathbb{R}^+$ için

$$C(d_{i+1} - d_i) = C(i^2 + 2i + 1 - i^2) = C(2i + 1) < i^2$$

olarak tanımlanabilmektedir.

Her $i > i_0$,

$$i_0 = \left\lfloor 2C + \sqrt{C^2 + C} \right\rfloor + 1$$

teorem 2.2'nin ispatı

$$|d(kw).x \geq k - (n+1)(d_{i+1} - d_i)$$

eşitsizliğinde görüldüğü gibidir.

Olarak,

$$|d(kw).x \geq k - (n+1)(2i+1)$$

$$|d(kw).x \geq k - (n+1)(2\sqrt{k k_{\max}} + 1)$$

seçiyoruz

$$k = n + 2 + \sqrt{n^2 + 3n + 2}$$

ve aşağıdaki eşitsizliği

$$|d(kw).x| > 0$$

elde ediyoruz.

Yeni ağırlıkların orjinal fonksiyonu ifade ettiğini göstermektedir.

Örnek 2.6 (Sayısal örnek)

Örnek 2.7 (2'nin Kuvvetleri) Üstteki ispat 2'nin üsleri için kullanılırsa ne olur:

$$D = \{1, 2, 4, 8, 16, \dots\}$$

Teorem 2.2'deki ifade kullanılırsa

$$C(d_{i+1} - d_i) = C(2^{i+1} - 2^i) = C(2^i) < C d_i$$

durum $\forall C \in \mathbb{R}^+, \exists i_0 / C(d_{i+1} - d_i) < d_i$ her $i \geq i_0$ için sağlanamaz. Teorem kullanılamaz. İspat nerede başarısız olmaktadır?

$$| (kw).x | \geq k \text{ her } x \in \{0, 1\}^n \text{ için}$$

$$| (kw).x - d(kw).x | + | d(kw).x | \geq k$$

$$| (kw - d(kw)).x + d(kw).x | \geq k$$

$$| (kw - d(kw)).x | + | d(kw).x | \geq k$$

$$| d(kw).x | \geq k - | (kw - d(kw)).x |$$

$$| d(kw).x | \geq k - (n+1)(d_{i+1} - d_i)$$

Bu noktada şu ifade kullanılarak $d_{i+1} - d_i = d_i < k$

$$| d(kw).x | \geq k - (n+1)k$$

$$| d(kw).x | \geq -nk$$

ispatımız burada işe yaramadı ancak 2'nin kuvvetleri için sonuçları kanıtlamanın başka bir yolu olmalıdır. Karşıt bir örneğe ihtiyacımız var. Ağırlık vektörü $(-5, 1, 1, 2, 3, 4)$ olan 5 değişkenli LT fonksiyonu

$$f(x_1, x_2, x_3, x_4, x_5) = \text{sgn}(-5 + x_1 + x_2 + 2x_3 + 3x_4 + 4x_5)$$

düşünelim.

2'nin üsleri olarak ifade edebileceğimiz ağırlıkları bulduğumuzu

$$f(x_1, x_2, x_3, x_4, x_5) = \text{sgn}(-2^{a_0} + 2^{a_1}x_1 + 2^{a_2}x_2 + 2 \cdot 2^{a_3}x_3 + 2 \cdot 2^{a_4}x_4 + 2 \cdot 2^{a_5}x_5)$$

farz edelim.

$a_i \in N$ iken. f tanımında yeni ağırlıkların ilişkisine göre şu gözlemleri

$$f(11100) = 0 \text{ ama } f(11010) = 1 \Rightarrow 2^{a_3} < 2^{a_4}$$

$$f(10010) = 0 \text{ ama } f(10001) = 1 \Rightarrow 2^{a_4} < 2^{a_5}$$

$$f(00001) = 0 \text{ ama } f(00110) = 1 \Rightarrow 2^{a_5} < 2^{a_3} + 2^{a_4}$$

yaparız.

Bu eşitsizlik sisteminin tam sayı sonucu olmadığına görülmesi çok basittir.

İspat: (devam)

Teorem 2.2 deki 2. savı ispatlamak için örnek 2.7 deki fikir izlenir. $K \cdot 2^{a_i}$ 'nin ağırlıkları ile tamamlanamayan fonksiyonu

$$w_l < w_{l+1} < \dots < w_{\lfloor l + \frac{1}{a} \rfloor + 1} < \dots < w_{\lfloor l + \frac{2}{a} \rfloor + 1}$$

bulunmak isteniliyor. Bunu sağlayan herhangi bir ağırlık kümesi seçeriz. Ağırlıkların farklı olmasını garanti ederiz ayrıca fonksiyonu düzenlenir.

İlk eşitsizlik kümesi

$$w_l + w_{\lfloor l + \frac{1}{a} \rfloor} > w_{\lfloor l + \frac{2}{a} \rfloor}$$

gösterir.

İlk eşitsizlikler kümesi aşağıdakileri

$$w_l < \frac{1}{2} w_{\lfloor l+\frac{1}{a} \rfloor} < \frac{1}{4} w_{\lfloor l+\frac{2}{a} \rfloor}$$

ifade etmektedir.

İkinci ile birlikte gerekli çelişki sağlanır. Bu örnek 2.7 de $K = a = 1$ olarak yapılmıştır. Fonksiyonları herhangi bir sabit olan K ve a 'ya göre nasıl yapılandırabiliriz? Geliştirme için bölüm 5'te elemanlar kullanırız.

Yardımcı Teorem 2.3 (yoğun ağırlıklar ile fonksiyon kurma)

K ve A sabitleri verilmiş ve aşağıdakileri sağlayan herhangi bir ağırlık kümesi olan fonksiyonumuz

$$w_l < w_{l+1} < \dots < w_{\lfloor l+\frac{1}{a} \rfloor+1} < \dots < w_{\lfloor l+\frac{2}{a} \rfloor+1}$$

vardır.

Ve

$$w_l + w_{\lfloor l+\frac{1}{a} \rfloor} > w_{\lfloor l+\frac{2}{a} \rfloor}$$

İle yardımcı teorem 2.3'ün ispatı Bölüm 3.4.3'te verilmiştir.

2.6 Sonuç

Bölüm 2'de LT fonksiyonlarının tam sayı ağırlıklarla tamamlanması gibi belli başlı sonuçların genelleştirilmesi yapılmıştır. Yoğunluğu yeterli D tarafından çekilen ağırlıkları olan LT fonksiyonlarının verilen herhangi bir alt küme, $D \in \mathbb{R}$ ile tamamlanabileceğini gösterdik. D 'nin polinom olarak ya da süper polinom olarak artması gerekir fakat üstel olarak artıyorsa bazı fonksiyonlar reelleştirilemez. Teorem 2.2 D 'nin doğru şartlarını gösterir. İspatlar müspettir, ağırlıkların oluşumundaki ya da karşıt örneklerin oluşumundaki algoritmalar gösterilmiştir. Sayfa [49] 'da 38 de tamamlanan doğrusal düşünce listesi ile 2 katlı LT devreleri tanıtılmıştır. İkinci kat girişindeki ağırlıkların büyüklükleri 2'nin kuvvetleridir.

Bölüm 3

Minimum Ağırlıklar

3.1 Giriş

Bu bölümde tam sayılı ağırlıklar gibi ikili girdi ve çıktılı tek doğrusal eşik girişi incelenecektir. Bu girişler matematikte *doğrusal eşik fonksiyonu* olarak tanımlanır.

Tanım 3.1 (Doğrusal Eşik Fonksiyonu) n değişkenli doğrusal eşik fonksiyonu doğru yanlış fonksiyonudur $f : \{0,1\}^n \rightarrow \{0,1\}$ her x için $x \in \{0,1\}^n$ ve $w \in \mathbb{R}^{n+1}$ olarak

$$f(x) = \text{sgn}(F(x)) = \begin{cases} 1, & F(x) \geq 0 \text{ için} \\ 0, & \text{Diğer} \end{cases}$$

$$F(x) = W \cdot (-1, x) = -w_0 + \sum_{i=1}^n w_i x_i$$

sabitlenmiştir.

Bölüm 2 de görüldüğü gibi LT fonksiyonları tamsayı ağırlıkları ile reelize edilebilir. Bu bölümün devamında genelliği kaybetmeden ağırlıkların tam sayı olduğunu varsayacağız.

Ayrıca doğrusal eşik fonksiyonu

$$f : \{-1,1\}^n \rightarrow \{0,1\}$$

tamamlanabilir.

$\{0, 1\}$ ve $\{-1, 1\}$ sunumlarına yönlendireceğiz.

Verilen f fonksiyonda ağırlık vektörü özel değildir.

Tanım 3.2 (Ağırlık alanı) Verilen eşik fonksiyonu f ve W tanım 3.1 i karşılayan tüm ağırlıkların kümesi olarak tanımlanıyor.

$$W = \{w \in \mathbb{Z}^n : \forall x \in \{0,1\}^n, \text{sgn}(W \cdot (-1, x)) = f(x)\}$$

Aşağıda ağırlıkların boyutunun bir ölçümü vardır.

Tanım 3.2 (En Düşük Ağırlık Boyutu) Ağırlık boyutunun vektörü ağırlıkların mutlak değerlerinin toplamı olarak tanımlanır. En düşük ağırlık boyutunun doğrusal eşik fonksiyonu şu şekilde

$$S[f] = \min_{w \in W} \left(\sum_{i=0}^n |w_i| \right)$$

tanımlanır.

Asgariyi sağlayan belirli vektöre minimum ağırlık vektörü denir. Doğal olarak $S[f]$ 'nin fonksiyonudur.

3.1.1 Motivasyon

Neden eşik devresindeki ağırlıkların boyutuyla ilgileniyoruz?

Eşik devresi şaşırtıcı derecede güçlü görünüyor. Örneğin polinom boyutunun sabit derinlikli eşik devresi tarafından tam sayı bölmesi tamamlanıyor. AC^0 'deki her fonksiyon polinom benzeri ekseri devrelerin3 derinlikten hesaplanabilir olması [1] ayrıca sağlanmıştır. Bu tüm ACC^0 için gerçektir. Etkili üst sınırlar verildiğinde alt sınırları bulmak elde etmek için güçlüklerle karışılması sürpriz değildir. Aslında eşik devresi için en iyi alt sınır derinlik2 için gerekli olan Inner-Product-Mod-2 (IP2) 'nin sonucudur. Fakat bu alt sınır devrenin küçük ağırlıklar içerdiğini varsayar ve IP2 herhangi bir ağırlıklı 2 derinlikli polinom eşik devresi tarafından hesaplanabilir. Şöyle ki, eşik devreleri için alt sınırların gelişmesinin edinilmesi, büyük ağırlıkların rolünün anlaşılması ilgili olduğu görülüyor. Eğer yalnızca küçük gelişme boyutlarının katsayılarıyla eşik elemanları kısıtlanabiliyorsa, devrenin hesaplanabilir gücü nasıl kısıtlanabilir? Bu durum sayfa [17], [33], [43] de gösteriliyor. Üstel şekilde artan ağırlıklı tek eşik elemanı $S[f] \sim 2^n$ tarafından tamamlanabilen doğrusal eşik fonksiyonları bulunuyor. Fakat bu fonksiyonlar daha küçük polinom olarak artan ağırlıklı eşik elemanı $S[f] \sim n^d$, d tarafından tamamlanamıyor. Bu sonuçların ışığında yukarıdaki soru ile doğrusal eşik fonksiyonlarının kümelerinin bulunduğu sınıfı ("küçük" polinom olarak artan ağırlıklı fonksiyon sınıfı) açıklayarak uzlaşabiliriz. Yakın zamandaki araştırmaların çoğu küçük ağırlıklı devrelerin gücüne ve keyfi ağırlıklı devreyle ilişkilerine odaklanmıştır. Özellikle bunlar gösteriyor ki devredeki derinliği bir arttırmak tüm polinom boyuttaki ağırlıkları düşürmek için yeterlidir. Bu bölümde değişik yaklaşımlar ediniyoruz. Devrelerle uğraşmak yerine tek eşik girişine odaklanılıyor.

3.1.2. Organizasyon

Söz konusu bölümün geri kalanının özet bir çerçevesini vermekteyiz. Bölüm 3.2’de biz etkiyi en aza indirirken karşı karşıya kalınan bazı zorlukları ve çıktı alanı tarafından nasıl etkilendiğini göstermekteyiz. Bölüm 3.3’de biz $\{-1,1\}$ ’de tanımlanan fonksiyonları dikkate alıyoruz. Eşikte olmayan fonksiyonlar sınırlandırılıyor (genelleştirilmiş ana fonksiyon) ve en düşük ağırlığa sahip bu tür fonksiyonların nasıl yapılacağını gösteriyoruz. Bölüm 3.4’de biz $\{0,1\}$ üzerinde tanımlanan her hangi bir eşik fonksiyonu ile temas etmemizi sağlayan minimal fonksiyonları oluşturmanın başka bir yolunu veriyoruz.

3.2 Başlangıçlar ve Örnekler

Bu bölümde bir eşik fonksiyonunun ağırlığını en aza indirmeye çalışırken karşı karşıya kalınan bazı zorlukları açıklamaktayız. Biz ayrıca çıktı alanının etkileme ölçülerinden nasıl etkilendiğini gösteriyoruz. (Örneğin $\{-1,1\}$ ’e karşılık $\{0,1\}$) Bununla ilgili sonuçlara sayfa [25] ‘ten bakınız.

3.2.1 Etkileri En Aza İndirme

Bir eşik unsurunun etkilerini ölçme konusunda analiz yaparken ana zorluk, aşağıdaki örnekte gösterildiği gibi farklı bir etki seti tarafından tek bir doğrusal eşik fonksiyonu gerçekleştirilmesinden kaynaklanmaktadır.

Örnek 3.1 (En Düşük Etkiye Sahip Bir Eşik Fonksiyonu) Aşağıdaki iki etki setini değerlendirelim.(etki vektörleri)

$$w = (4 \ 1 \ 2 \ 5), \quad F_1(x) = -4 + x_1 + 2x_2 + 5x_3$$

$$w' = (8 \ 2 \ 4 \ 10), \quad F_2(x) = -6 + 2x_1 + 4x_2 + 10x_3$$

Her ikisi de aynı eşik fonksiyonunu

$$f(x) = \text{sgn}(F_2(x)) = \text{sgn}(2F_1(x)) = \text{sgn}(F_1(x))$$

vermektedir.

Yakından bir bakış $f(X) = \text{sgn}(-1 + x_3)$, yukarıdaki etki faktörlerinin hiç birisinin minimal ölçüde olmadığını göstermektedir. Gerçekten de en düşük etki $w^N = (1 \ 0 \ 0 \ 1)$ ve $S[f] = 2$ ’dir.

Verilen bir etki setinin minimal düzeyde olup olmadığını belirlemek genel olarak bir sorundur. Bizim tekniğimiz etki vektörlerinin yapılmasından ibarettir ve onun en düşüğü kolaylıkla sağlanabilir. Biz daha sonra daha büyük bir fonksiyonlar seti elde etmek için onları en düşük düzeyde muhafaza ederken nasıl

değiştirileceği gösteriliyor.

3.2.2 $\{0,1\}$ karşılık $\{-1,1\}$

Bizim aynı fonksiyonu $(0,1)$ ve $(-1,1)$ üzerinde gerçekleştirdiğimizi farz edin. Etkilenme nasıl olacaktır? Bir örneğe göz atalım.

Örnek 3.2 (OR Fonksiyonu)

1. $x_i \in \{0,1\}$ için

$$OR(x_1, \dots, x_n) = \text{sgn}(-1 + x_1 + \dots + x_n)$$

olsun.

Ağırlıkların boyutu $S = n + 1$. Bu ağırlıklar en düşük olanıdır.

Kanıt: Ağırlıklar tamsayıdır. Boyutlarını düşürmek OR'un tanımını delemek şekilde 0'lamak demek oluyor.

2. Şimdi, $x_i \in \{-1,1\}$ için

$$OR(x_1, \dots, x_n) = \text{sgn}(n - 2 + x_1 + \dots + x_n)$$

olsun.

Ağırlıkların boyutu $S = 2n - 1$. Bu ağırlıklar da en düşük olanıdır.

Kanıt: OR'u ifade eden tüm ağırlıklar pozitif olmalıdır. Any weights that implement OR have to be positive. $S' < 2n-2$ boyutunda ağırlıkların olduğunu düşünün. Hiçbir ağırlık 0 olamaz, böylece $\sum_1^n w' \geq n$, $-w_0 < (2n - 2) - n = n - 2$ eşik fonksiyonunu ifade eder. w_i' en düşük ağırlık olsun. $X_i = 1$ olsun ve diğer tüm girdiler -1 olsun. $\sum_1^n w' < -w_0(n - 2)$ böylece $F(X) < 0$ OR'un tanımını delerek.

Bu örnekten de görülebileceği gibi $\{0,1\}$ ifadesi $\{-1,1\}$ ifadesinden daha küçük ağırlıktadır. Bu genel olarak doğru mudur?

Örnek 3.3 (MAJ Fonksiyonu) n değişken sayısı tek olsun. Girdilerin yarısından fazlası doğru olursa MAJ fonksiyonu doğru dönmektedir.

• $x_i \in \{0,1\}$ için

$$MAJ(x_1, \dots, x_n) = \text{sgn}(-\frac{n+1}{2} + x_1 + \dots + x_n)$$

olsun.

Ağırlıkların boyutu $S = \frac{3n+1}{2}$. Durum 2'dekine benzer bir ispatla bunların en küçük olduğunu gösterebiliriz.

• Şimdi $x_i \in \{-1, 1\}$ olsun,

$$MAJ(x_1, \dots, x_n) = \text{sgn}(x_1 + \dots + x_n)$$

Onların ölçüsünün indirilmesi, bir veya daha fazlasının 0'ya getirilmesini ifade ettiği içi bu etkiler minimaldir. Bu ise MAJ'ın tanımını değiştirecektir. Etki ölçüsü $S = n$ dir.

Bu ikinci örnek genel olarak $\{0, 1\}$ mi yoksa $\{-1, 1\}$ 'nin mi daha küçük ağırlıkta bir fonksiyon üreteceğini söyleyemeyeceğimizi göstermektedir.

3.3 $\{-1, 1\}$ Üzerinde Genelleştirilmiş Çoğunluk Fonksiyonu

Bu bölümde aşağıdaki modele

$$f : \{-1, 1\} \rightarrow \{0, 1\}$$

$$f(x) = \text{sgn}(\sum_{i=1}^n w_i x_i)$$

çalışılacaktır.

Eşik olmadığına dikkat edin, biz belirli etkilere sahip bir çoğunluk fonksiyonuna göz atıyoruz. Biz minimal etkilere sahip fonksiyonların oluşturduğu fonksiyonlara cevap veriyoruz. Özellikle amacımız belirli bir çıktı n ve ölçü S elde etmek içindir.

3.3.1 Matematiksel Kurgular

Biz minimal etkinin kolaylıkla kurulabileceği fonksiyonlar ile ilgilenmekteyiz. Minimal etkiyi bulma bir araştırmayı kapsar; Biz bu nedenle sınırlı bir etki alanına sahip fonksiyonları bulmak için çabalıyoruz. Aşağıda yazılı olanlar bize w üzerine

sınırlamalar koymamıza izin vermektedir. (Bu bölümün geri kalan kısmında, karışıklığa meydan vermemek için biz açık bir şekilde vektörleri anlatılacaktır.)

Tanım 3.4 (Bir Doğru Yanlış Fonksiyonun Temel Alanı) Bir vektör $G \{ -1, 1 \}^n$ $f(v) = f(-v)$ gibi f 'nin temeli olarak adlandırılır. Biz temel alan R 'yi f 'nin tüm temel setleri olarak tanımlarız.

Tanım 3.5 (Temel Generator Matrisi) Verilen bir etki vektörü $w \in W$ ve bir temel $v \in G \subset R$ için, temel generator matrisi, $G = (g_{ij})$, bir $n \times k$ -matrisidir, $\{-1, 0, 1\}$ kapsamındadır, g dizileri w 'ye doğru düz açılı ve diktir ve tüm sıfır olmayan koordinatlarda v 'ye eşittir, yani,

1. $G\vec{w} = 0 \quad Gw = 0$
2. $g_{ij} = 0$ veya $g_{ij} = v_j$ her i ve j için.

Örnek 3.4 (Temel Generator Matrisi) Bir etki vektörü tarafından belirlenen bir linear fonksiyon verildiğini var sayalım. $w = (1, 1, 2, 4, 1, 1, 2, 4)$. Araştırırken $v = (1, 1, 1, 1, -1, -1, -1, -1)$ olduğunu tespit ederiz. Dikkat edin $w_1 + w_2 - w_7 = 0$ ve $g.w = 0$ olarak yazılabilir, orada $g = (1, 1, 0, 0, 0, 0, -1, 0)$ G setidir $r = v - 2g$. g tüm sıfır olmayan koordinatlarda v 'ye eşit olduğu için $f \in \{-1, 1\}$ 'dir. Ayrıca $r.w = v.w + g.w = 0$ dır. Biz yeni bir temel ortaya atıyoruz. $r = (-1, -1, 1, 1, -1, -1, 1, -1)$.

Lemma 3.1 (G ve W'nin Dikeyliği) Verilen bir etki vektörü $w \in W$ ve bir temel $v \in R$, $uG^T = 0$ her hangi bir etki vektörü $u \in W$ için düzenlenmiştir.

Kanıt: $u \in W$ ve bir dizi için, G 'nin g_i , $v = v - 2g_i$ olsun $\langle f, g \rangle$, eğer $G \{ -1, 1 \}^n$ tanımlarken ve if $\blacksquare w = 0$ ise. Bu $f(if) = f(-if)$ anlamına gelir: Her hangi bir temel vektör için $u \in W$, $\text{sgn}\{u.v\} = \text{sgn}\{-u.v\}$. Bu nedenle, $u \blacksquare (v - 2g_i) = 0$ ve sonuç olarak, $v \cdot u = 0$ olduğu için, $u \cdot c_{ji} = 0$ kabul ederiz.

Yardımcı Teorem 3.2 (Minimalite) $w \in W$ ağırlık vektörü için ve kök $v \in R$ için, eğer, $\text{rank}(G) = n - 1$ (i.e., G $n - 1$ bağımsız satıra sahip) ve $|w_i| = 1$ bazı i 'ler için, o zaman w en düşük ağırlık vektörüdür.

İspat: Yardımcı teorem 3.1 'den her u ağırlık vektörü $uG^T = 0$ eşitliğini sağlar. $\text{rank}(G) = n - 1$ $\dim(W) = 1$ olduğunu gösterir. Örneğin tüm muhtemel ağırlık vektörleri kendi içinde çarpılan tam sayılardır. $|w_i| = 1$ olduğu için tüm vektörler $u = kw$ $k > 1$ değeri için eşitliğinde vardır.

Örnek 3.4 'ü yardımcı teorem 3.2 'nin uygulamaları ile tamamladık.

Örnek 3.5 (Minimalite)

$$\vec{w} = (1, 1, 2, 4, 1, 1, 2, 4)$$

$$\vec{v} = (1, 1, 1, 1, -1, -1, -1, -1)$$

verilmiş olsun.

Aşağıdaki gibi bir yapı kurabiliriz:

$$G = \begin{pmatrix} 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 \\ 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & -1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & -1 \end{pmatrix}$$

Rank(G)'nin minimum olduğunu göstermek oldukça kolay ve $S[f] = 16$.

3.3.2 Ağırlık Vektörleri

Örnek 3.5'te nasıl verilen vektörün minimum olduğunu gösterildiğini işlemiştik. Bu kısımda keyfi girdi değerleri ile minimum ağırlıklı boyutunu içeren doğrusal eşik fonksiyonunun örneklerini gösterilir.

Ağırlık vektörü oluşturmak ve en düşük olduğunu göstermek istiyoruz. Girdilerin sayısına n diyelim ve n çift olsun. w de iki denk bloktan oluşsun:

$$(w_1, w_2, \dots, w_{n/2}, w_1, w_2, \dots, w_{n/2})$$

Açıkça, $v = (1, 1, \dots, 1, -1, -1, \dots, -1)$ bir kök ve G ilgili karşılık gelen matristir.

$$G = \begin{pmatrix} 1 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & -1 & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & -1 & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & \dots & 0 & 0 & 0 & 0 & 0 & -1 & 0 & \dots & 0 & 0 & 0 \\ \dots & & & & & & & & & & & & & & & \\ \dots & & & & & & & & & & & & & & & \\ \dots & & & & & & & & & & & & & & & \\ 0 & 0 & 0 & 0 & \dots & 0 & 1 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 & 0 & 1 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & -1 \end{pmatrix}$$

3.3.3 Yapım

S tamsayısı ve n değişken ile n değişkenli ve s boyutunda minimum ağırlıkta bir fonksiyon olduğunu göstermektedir.

Teorem 3.1 (Ana Sonuç) (s,n) ikilisi için aşağıdakileri sağlayan

1. $n \leq s \leq \begin{cases} 2^{\frac{n}{2}} & n \text{ çift ise} \\ 2^{\frac{n-1}{2}} + 2^{\frac{n-3}{2}} & n \text{ tek ise} \end{cases}$
2. s çift

[f] = s boyutunda en düşük ağırlıklı n değişkenden oluşan doğrusal bir eşik fonksiyonu vardır.

Kanıt: Verilen (s,n) çifti yukarıdaki şartları sağlar. $\sum_{i=1}^n |w_i| = s$ koşulunu sağlayan ağırlık vektörü w, yapılandırılır. Daha sonra $f(x) = \text{sgn}(\vec{w} \cdot \vec{x})$ fonksiyonunun minimum ağırlık vektörü gösterilir. İspat sadece n 'in çift olduğunu gösterir.

YAPIM.

1. $(a_1, a_2, \dots, a_{n/2}) = (1, 1, \dots, 1)$ tanımlanır.
2. Eğer $\sum_{i=1}^{n/2} a_i < s/2$ ise $a_i < 2^{i-2}$ gibi en küçük a_i alınarak arttırılır. (Bağlantı olayında olduğu gibi en küçük i'ye endeksli w_i alınır.)
3. $\sum_{i=1}^{n/2} a_i < s/2$ veya $(a_1, a_2, \dots, a_N) = (1, 1, 2, 4, \dots, 2^{\frac{n}{2}-2})$
4. Set $w = (a_1, a_2, \dots, a_{n/2}, a_1, a_2, \dots, a_{n/2})$ olana kadar , bir önceki basamağı tekrar ederiz.

Çünkü boyutu zamanla arttırarak algoritmada $n < s < 2^{n/2}$ eşitsizliğini sağlayan her s tamsayısı için istenilen sonuca yaklaşılr. w'nin en düşük olduğunu gösterelim.

MINİMALİTE verilen $w = (a_1, a_2, \dots, a_{n/2}, a_1, a_2, \dots, a_{n/2})$ 'den kökü $u = (1, 1, \dots, 1, -1, -1, \dots, -1)$ ve $w_i = w_{i+n/2}$. eşitliği ile ilgili genelleştirici G matrisinin n/2 satırını buluruz. Eklenen satırlardan anlaşılıyor ki ilk k a_i 'leri ikinin kuvvetleri (k s ve n 'ye bağlıken). Bunlar $a_i = \sum_{j=1}^k a_{ij} a_j$ şeklinde yazılabilir ve k-1 satır üretilebilir. Son olarak diğer bütün a_i 'ler $i > k$ iken 2^{k+1} den küçüktür. Bunlar ikili açılım olarak yazılabilir, ($a_{ij} \in \{0,1\}$ iken $a_i = \frac{n}{2} - k$ gibi ağırlıklar vardır. G'nin bağımsız toplam n-1 satırı vardır. $\text{rank}(G) = n - 1$ ve $w_1 = 1$; yardımcı teorem 3.2'ye göre $\vec{w}$ minimumdur ve $S[f] = s$ 'dir.

Örnek 3.6 (10 Değişkenli ve 26 Büyüklükteki Bir Fonksiyon) İle başlıyoruz

$$\vec{a} = (1,1,1,1,1)$$

Öteleme yapıyoruz,

$$(1,1,2,1,1)$$

$$(1,1,2,2,1)$$

$$(1,1,2,2,2)$$

$$(1,1,2,3,2)$$

$$(1,1,2,3,3)$$

$$(1,1,2,4,3)$$

$$(1,1,2,4,4)$$

ve sonuç olarak algoritma

$$\vec{a} = (1,1,2,4,5)$$

sonsuzla gitmektedir.

$$\vec{w} = (\vec{a}, \vec{a}) = (1,1,2,4,5, 1,1,2,4,5)$$

en düşük olduğu iddia edilir.

Gerçekten, $\vec{v} = (1,1,1,1,1, -1, -1, -1, -1, -1)$ ve

$$G = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 \\ 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & -1 \end{pmatrix}$$

9 rankında bir matristir.

Örnek 3.7 (Polinom Boyutlarında Fonksiyonlar) Bu örnekte Teorem 3.1 'in uygulamasını göreceğiz. $S[f] < n^d$ için doğrusal eşik fonksiyonları olan $L\hat{T}^{(d)}$ 'yi tanımlıyoruz. Teorem gösteriyor ki her çift n için n değişkenli ve minimum ağırlıklı $S[f] = n^d$ bir f fonksiyonu vardır. Buradan yapılacak çıkarım tüm d 'ler için $L\hat{T}^{(d)}$ uygun bir LT alt kümesidir.

3.4 {0,1} de Keyfi Eşik Fonksiyonu

Bu kısımda en düşük ağırlıklı eşik fonksiyonunun oluşturulabilmesi için değişik teknikler gösterilecek. Her bir yükün boyutu değişkenlerin sayısı ile fonksiyonları oluşturabiliriz. Fonksiyonları $\{0,1\}$ girdi tanım kümesinde düşünebiliriz fakat keyfi girdi uzayı $\{a,b\}$ için argüman hala vardır. Bölümün geri kalanında karışıklığı önlemek amacıyla vektörleri belirtmek için büyük harfler kullanacağız.

3.4.1 Yaklaşım

Kullandığımız metotlar sayfa [52]'nin sonuçlarını taban almaktadır. Genelleştirmenin dışına çıkmadan ağırlıkların tamamıyla pozitif tamsayılar olduklarını var sayalım. Hedefimiz $S = \sum_0^n |w_i| = \sum_0^n w_i$.

Eşitliğini küçültmektir. Eşitlik [32]'den bildiğimiz gibi aynı fonksiyon tarafından tamamlanan diğer ağırlıklar, U , tamamen pozitif olmalıdır.

$W, \sum_0^n w_i \geq \sum_0^n u_i$ şartları altında göstereceğiz.

Aşağıdaki eşitliklerde olan girdi vektörlerini X ve Y

$$F(X) = -w_0 + \sum_1^n w_i x_i = 0$$

$$F(Y) = -w_0 + \sum_1^n w_i y_i = 0$$

olarak düşünelim.

Matrisin satırlarını A olarak belirleyelim,

$$A = \begin{pmatrix} -1 & X^{(1)} \\ -1 & X^{(2)} \\ \dots & \dots \\ \dots & \dots \\ -1 & X^{(p)} \\ 1 & -Y^{(1)} \\ 1 & -Y^{(2)} \\ \dots & \dots \\ \dots & \dots \\ 1 & -Y^{(q)} \end{pmatrix} = \begin{pmatrix} -1 & x_1^{(1)} & x_2^{(1)} & \dots & x_n^{(1)} \\ -1 & x_1^{(2)} & x_2^{(2)} & \dots & x_n^{(2)} \\ \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ -1 & x_1^{(1)} & x_2^{(1)} & \dots & x_n^{(1)} \\ -1 - y_1^{(1)} - y_2^{(1)} & \dots & -y_n^{(1)} \\ -1 - y_1^{(2)} - y_2^{(2)} & \dots & -y_n^{(2)} \\ \dots & \dots & \dots \\ \dots & \dots & \dots \\ -1 - y_1^{(q)} - y_2^{(q)} & \dots & -y_n^{(q)} \end{pmatrix}$$

satırların tekrarına izin verdik: $X^{(i)} = X^{(j)} = \dots = X^{(k)}$, elde ettik.

Örnek 3.8 (Matris A) Elimizde aşağıdaki ağırlıkların olduğunu varsayalım.

$$W = (13 \ 6 \ 6 \ 3 \ 3 \ 2 \ 2 \ 1 \ 1)$$

Hedefimiz minimum olduğunu göstermektir. Öncelikle matris A'yi oluşturmalıyız.

$$A = \begin{pmatrix} -1 & X^{(1)} \\ -1 & X^{(2)} \\ -1 & -Y^{(1)} \\ -1 & -Y^{(2)} \end{pmatrix} = \begin{pmatrix} -1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 \\ -1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & -1 & 0 & -1 & 0 & -1 & 0 & -1 \\ 1 & -1 & 0 & -1 & 0 & -1 & 0 & -1 & 0 \end{pmatrix}$$

A için pek çok seçenek mevcuttur. Yukarıda gösterilen daha sonra göreceğimiz kadar iyi bir örnek değildir.

Teori 3.2 (En Düşük Değeri Sağlama Koşulu) : Ağırlık vektörü W olarak kabul edildiğinde, A'yı aşağıda belirtilen şekilde açıklarız. Eğer $a > 0$ ve A aşağıdakini sağlamaktaysa

$$(1 \dots 1)A = (a \dots a)$$

ağırlık vektörü W minimum değere sahiptir.

Kanıt: X'lerin ve F'lerin tanımına göre, A matris'i aşağıdakileri

$$A.(w_0 w_1 w_2 \dots w_n)^T = \overset{\overleftarrow{p}}{00\dots 00} \overset{\overleftarrow{q}}{1\dots 11} \quad (3.1)$$

sağlar.

$sgn(0) = 1$ ve $sgn(-1) = 0$ olduğundan, aynı fonksiyonu uygulayan herhangi bir ağırlık vektörü, U, yukarıdaki eşitlikleri "=" yerine ">" ile yerine getirmelidir.

$$A.(u_0 u_1 u_2 \dots u_n)^T = \overset{\overleftarrow{p}}{00\dots 00} \overset{\overleftarrow{q}}{1\dots 11} \quad (3.2)$$

$V = U - W$ olsun ve (3.1)'deki eşitlikleri (3.2)'deki eşitsizliklerden çıkaralım. Sonuç olarak şunları

$$A.(v_0 v_1 v_2 \dots v_n)^T = \overset{\overleftarrow{p+q}}{00\dots 00} \quad (3.3)$$

buluruz.

Şimdi öyle bir A alalım ki

$$\overset{\overleftarrow{p+q}}{11\dots 11}.A = \overset{\overleftarrow{n}}{aa\dots a} \quad (3.4)$$

A pozitif bir tamsayı olduğunda eşitlik (3.3)'teki tüm eşitsizlikleri soldaki tüm 1 vektörüyle çarpıp şu sonucu

$$\overbrace{(1 \ 1 \ \dots \ 1 \ 1)}^{p+q} \cdot A \cdot (v_0 \ v_1 \ v_2 \ \dots \ v_n)^T \geq \overbrace{(1 \ 1 \ \dots \ 1 \ 1)}^{p+q} \cdot \overbrace{(0 \ 0 \ \dots \ 0 \ 0)}^{p+q}^T$$

$$\overbrace{(a \ a \ \dots \ a \ a)}^n \cdot (v_0 \ v_1 \ v_2 \ \dots \ v_n)^T \geq 0$$

$$a \sum_0^n v_i \geq 0$$

elde ederiz.

Tüm $i = 0, \dots, n$ için, $a > 0$, $w_i \geq 0$, $u_i \geq 0$ olduğundan, şunu

$$\sum_0^n u_i \geq \sum_0^n w_i$$

bilmekteyiz.

Girdi kümesinin $\{0,1\}$ olduğu gerçeğini, kanıtın hiç bir aşamasında kullanmadığımıza dikkat ediniz. Hatta yukarıdaki kanıt herhangi bir $\{a, b\}$ girdi kümesi için geçerlidir. Görebileceğiniz gibi kanıt, eşitlik (3.4)'ü sağlayacak bir A'nın yapılandırılması üzerine kurulmuştur. Bir A oluşturabilmek için W seçimine bağlı olan uygun X'lere ve F'lere ihtiyaç vardır.

3.4.2 Basit Oluşum

Bu bölümde W ile, genel oluşum için ağırlık vektörü, karşı karşıya geleceğiz, ve uygun bir A matrisi bularak W'nin en düşük değerde olduğunu kanıtlayacağız. Eşik değeri w_0 'yu isteğe bağlı bir değer olarak kabul edelim. Şunları seçelim:

$$w_1 = \left\lfloor \frac{w_n}{2} \right\rfloor, w_3 = \left\lfloor \frac{w_0 - w_1}{2} \right\rfloor, w_5 = \left\lfloor \frac{w_0 - w_1 - w_3}{2} \right\rfloor, \dots, w_{n-1} = 1 \quad \text{ve} \quad w_{2i} = w_{2i-1} \quad i=1, \dots, n$$

için. Öyle bir "n" seçiyoruz ki $\sum_1^n w_{2i-1} \geq w_0 - 1$ olsun. Bir örnek inceleyelim.

Örnek 3.9 ($w_0 = 13$) Yukarıda tekrarlanan tanımı uygulayarak, Örnek 3.8'deki ağırlık faktörünü elde ediyoruz: $W = (13 \ 6 \ 6 \ 3 \ 3 \ 2 \ 2 \ 1 \ 1)$. Burada, A için olan X ve Y tipi satırları izleyiniz.

$$\begin{cases} \begin{pmatrix} -1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ -1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 \end{pmatrix} \\ \begin{pmatrix} -1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ -1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \end{pmatrix} \\ \begin{pmatrix} -1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ -1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 \end{pmatrix} \\ \begin{pmatrix} -1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ -1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \end{cases} \begin{matrix} sumX_1 = (-2 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 2 \ 2) \\ sumX_2 = (-2 \ 1 \ 1 \ 1 \ 1 \ 2 \ 2 \ 0 \ 0) \\ sumX_3 = (-2 \ 1 \ 1 \ 2 \ 2 \ 0 \ 0 \ 1 \ 1) \\ sumX_4 = (-2 \ 2 \ 2 \ 0 \ 0 \ 0 \ 0 \ 1 \ 1) \end{matrix}$$

$$\begin{pmatrix} 1 & -1 & 0 & -1 & 0 & -1 & 0 & -1 & 0 \\ 1 & 0 & -1 & 0 & -1 & 0 & -1 & 0 & -1 \end{pmatrix} \underbrace{\hspace{10em}}_{sumY_1 = (2 \ -1 \ -1 \ -1 \ -1 \ -1 \ -1 \ -1 \ -1)}$$

1 vektörünün tümünü elde etmek için satırları kopyalayıp topluyoruz. Sadece tek sayılar ile numaralandırılmış sütunlar gösterilmektedir.

$$\begin{pmatrix} -2 & 1 & 1 & 1 & 2 \\ -2 & 1 & 1 & 2 & 0 \\ -2 & 1 & 2 & 0 & 1 \\ -2 & 2 & 0 & 0 & 1 \\ -2 & -1 & -1 & -1 & -1 \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 1 & -1 & 0 \\ 0 & 1 & -1 & -1 & 0 \\ -2 & -1 & -1 & -1 & -1 \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ -2 & -1 & -1 & -1 & -1 \end{pmatrix}$$

Sonuncusu 2 vektörünün tümüne eşit olmaktadır.

Teori 3.3 (Oluşumun En Düşük Değeri) Herhangi bir w_0 için, $S = 3 * w_0 - 2$ boyutlarında en düşük ağırlıkla ve bir çok değişkenle $n = \lceil \log_2 S \rceil$ bir eşik değeri fonksiyonu oluşturabiliriz.

Kanıt: $1A = a1$ 'i sağlayan bir A oluşturacağız ve Teori 3.2'yi uygulayacağız. A'yı oluşturmak için sadece 2 adet Y tipi vektör

$$\begin{pmatrix} 1 & -1 & 0 & -1 & 0 & \dots & -1 & 0 \\ 1 & 0 & -1 & 0 & -1 & \dots & 0 & -1 \end{pmatrix}$$

gereklidir.

İki olası şekilde $(2 \ -1 \ \dots \ -1)$ sonuç elde edilebilir.

X tipi vektörler, ikişer ikişer toplanarak, şu sonucu verirler:

Bu kısmi toplamaları tekrar edip toplayarak, 1 vektörünün tümünü elde edebiliriz. Peki bunu nasıl yapacağız? İki Y tipi ve iki X tipi vektörü toplayarak bir $(0, \dots, 0, 1)$ oluştururuz.

$$-2 \ 1 \ \dots \ 1 \ 2 \ 0 \ \dots \ 0 \ 0$$

veya

$$-2 \ 1 \ \dots \ 1 \ 2 \ 0 \ \dots \ 0 \ 1$$

S_i 'den kastımızın $i = 1 \dots n$, singleton vektörü $(0, \dots, 0, 1, 0, \dots, 0)$, 1. pozisyonda iken, olduğunu ifade edelim. Tüm S_i 'ye V ve F tipi vektörleri toplayarak ulaştığımızı göstermek için tümevarım kullanıyoruz. Hatta $j = 1, \dots, i - 1$ için tüm S_j 'ye ulaştığımızı varsayalım. İki X tipi ve iki Y tipi vektörleri toplayarak S_i üretebiliriz.

$$\begin{pmatrix} 2 & -1 & -1 & -1 & -1 & \dots & -1 & -1 & -1 & -1 \\ -2 & 1 & \dots & 1 & 2 & 0 & \dots & 0 & 0 & 0 \\ 0 & 0 & \dots & 0 & 0 & 1 & 0 & \dots & 0 & 0 \\ 0 & 0 & \dots & 0 & 0 & 0 & 1 & 0 & \dots & 0 \\ \dots & & & & & & & & & \dots \\ \dots & & & & & & & & & \dots \\ \dots & & & & & & & & & \dots \\ 0 & 0 & \dots & 0 & 0 & 0 & 0 & \dots & 0 & 1 \end{pmatrix}$$

Tüm S_i vektörlerini bulduktan sonra, onları $3 \times (2 \ -1 \ \dots \ -1)$ kere toplayıp, 2 vektörünün tümüne ulaşabiliriz.

3.4.3 Keyfi Büyüklük ve Sayıda Değişkenler için Oluşum

Bu bölümde, fazladan bir değişken elde etmek için ağırlığı nasıl bölmemiz gerektiğini göstereceğiz. Aynı zamanda, bir ya da iki değişken birim ağırlıkla toplandığında sonucun en düşük değerli bir fonksiyon olduğunu da kanıtlayacağız.

Yardımcı Teorem 3.3 (Ağırlığı Bölmek) $W = (w_0, w_1, \dots, w_n)$ en düşük değerinde olsun.. O halde $a + b = w_1$ minimum değerinde olduğunda, $\overline{W} = (w_0, a, b, w_2, w_3, \dots, w_{n+1})$ olacaktır.

Kanıt: İkinci sütunu tekrar ederken, bir A oluşturun.

Yardımcı Teorem 3.4 (bir girdiyi birim ağırlıkla toplamak) $W = (w_0, w_1, \dots, w_n)$ minimum değerinde olsun. O halde $w_{n+1} = 1$ minimum değerinde olduğunda, $\overline{W} = (w_1, w_2, w_3, \dots, w_{n+1})$ olacaktır.

Kanıt: W 'nin en düşük değerinde olmadığına yani W için daha iyi bir seçenek olduğunu düşünelim ve buna W diyelim. İki olasılık vardır. Ya $w'_{n+1} = 0$ olacaktır ya da $i < n + 1$ için bazı w 'lar uygun w_i 'den daha küçüktür. İkinci durumda, $x_{n+1}=0$ alıyoruz ve hipoteze ters olarak daha küçük ağırlıklarla uygulanmış olan orijinal fonksiyonu elde ediyoruz. Şimdi $w'_{n+1} = 0$ olduğunu, yani x_{n+1} 'e bağlı olmadığını farz edelim. Bu tüm X girdileri için $\sum_0^n w_i x_i \geq 0$ ya da $\sum_0^n w_i x_i \leq -2$ olduğunu ifade eder. Orijinal fonksiyonun minimum değerinde olmadığını söyleyerek w_0 'ı 1 birim azaltabiliriz.

Bu iki yardımcı teoremi kullanarak, fonksiyonları keyfi sayıda ve büyüklükte değişkenlerle oluşturmak basittir. Hatta bundan fazlasını da yapabiliriz: fonksiyonları sabit ağırlık yapısıyla oluşturabiliriz. Bu fikri Yardımcı Teorem 2.3'ü kanıtlayarak gösterelim.

Yardımcı Teorem 2.3'ün Kanıtı:

İki pozitif sabit K ve α aşağıdaki için bir ağırlık vektörü

$$w_l < w_{l+1} < \dots < w_{\left\lfloor l + \frac{1}{\alpha} \right\rfloor + 1} < \dots < w_{\left\lfloor l + \frac{2}{\alpha} \right\rfloor + 1}$$

$$w_l + w_{\left\lfloor l + \frac{1}{\alpha} \right\rfloor} \geq w_{\left\lfloor l + \frac{2}{\alpha} \right\rfloor}$$

Oluşturmaliyiz.

$\alpha = 1$ olması Örnek 2.7'de incelenmişti. Ağırlık vektörü şu şekilde

$$(-2^p, 1, 1, 2, 4, \dots, 2^p, 1, 1, 2, 4, \dots, 2^p, w, w+1, \dots, w + \left\lfloor \frac{1}{\alpha} \right\rfloor + 1, \dots, w + \left\lfloor \frac{2}{\alpha} \right\rfloor + 1)$$

Olsun.

$p, 2^p \geq w_{\left\lfloor l + \frac{2}{\alpha} \right\rfloor + 1}$ olacak şekilde ve yeterli büyüklükte bir tamsayı olacak şekilde seçilmiştir. Vektörün minimum düzeyde olduğunu kanıtladığımız şekilde, diğer ağırlıklar için de yukarıdaki iki koşulun sağlandığını kanıtlıyoruz.

3.5 Sonular

Keyfi ağırlık, büyüklük ve sayıdaki girdili minimum ağırlık eşiğı fonksiyonları oluşturmak için iki teknik gösterdik. $\{0,1\}$ ve $\{-1,1\}$ 'yi girdi kümeleri olarak kabul ettik. Bu teknikleri kullanarak, üstsel olarak ve polinom olarak büyüyen ağırlıkların arasındaki ayrımı ayrıntılarıyla inceledik. Asıl sorun bu tekniklerin devre boyutunda bulunan değışken ağırlığa sahip fonksiyonların mevcut alt sınırlarını genişletmede yararlı olup olmadığını bulmaktır.

Bölüm 4

Devre Derinliğı için Ağırlık Boyutundan Vazgeçmek

4.1 Giriş

Sinirsel ağ alanındaki deneysel birçok deneysel sonuç, doğrusal eşik değıerlerindeki katsayıların büyüklüğünün girdilerin boyu ile birlikte çok hızlı arttığını ve bu nedenle bu ağın pratikte kullanımını sınırladığını göstermektedir. Doğal olarak şu soru sorulabilir: eğer sadece katsayı büyümesi küçük olan eşik elemanları seçilirse, ağın bilişimsel gücü ne kadar sınırlanır? Bu bölüm büyük ağırlıklar gerektiren LT fonksiyonlarının uygulanması üzerine yoğunlaşmaktadır. Büyük ağırlıklarla tek bir LT geçidi kullanmak yerine, küçük ağırlıklı LT geçişlerinden oluşan iki tabakalı devre kullanıyoruz. Büyük ve küçük sırasıyla n 'nin, girdilerin sayısı, üstel olarak ve polinomial olarak anlamına gelmektedir.

Üstel olarak büyüyen ağırlıklara sahip olan eşik elemanı ile uygulanan bir fonksiyon olduğu sayfa [17], [33], [38], [43], fakat bu fonksiyon polinomial olarak büyüyen ağırlığa sahip olan eşik elemanı ile uygulanamayacağı gösterilmişti. Bu sonucun ışığında LT alt sınıfı küçük ağırlıklı fonksiyonların sınıfı olarak tanımlanmaydı. Siu ve Bruck $LT_d \subset LT_{2d+1}$ sayfa [13]'nin sınırları $LT_d \subset LT_{d+1}$

haline getirdiği ve bunu $LT_1 \subset LT_2$ şeklinde göstererek ve keyfi derinliği genelleyerek kanıtlamıştır. Lakin bu metot çok karmaşıktır ve kanıtları takip etmek zordur. [18], [13]'deki sonuçların basitleştirilmiş versiyonunu göstermektedir. $LT_1 \subset LT_2$ olduğunu göstermeye odaklanır. Bir yere kadar daha basit ve hissel bir oluşum yaratarak birincinin yerini almaktadır. Fikir, iki işlemi ağırlıkları azaltmak, iki ayrı güce ayırmak ve onları modula bir asal olarak bölmek için kullanmaktır. Sonuç olarak elde edilen küçük-ağırlıklı geçitler, yeterli prime kullanılırsa (58) doğru çıktı yaratan bir devreye bağlılardır.

Sadece belirli bir büyük-ağırlık fonksiyonun simülasyonunu kullanarak, sayfa [13] ve [18]'de belirtilen sonuçları daha da basitleştirdik: KARŞILAŞTIRMA. Sonuç olarak sırasıyla $O(n^4 \log n)$ olan devremizdeki bir takım geçitlerde sınırlar ve [18]'deki $O(n^{12} \log^{11} n)$ 'deki genel sınırdaki önemli gelişmeler elde ettik. Ayrıca bir bilgisayar simülasyonu çalıştırdık ve 22 değişene kadar minimum devreler belirledik. Simülasyonun sonuçlarını gösteriyoruz ve ileriki araştırmalar için uygulamalar ve talimatlar sunuyoruz.

4.2 Karşılaştırma için LT_2 devresi

Karşılaştırma fonksiyonunun 2 n-bit sayıdan oluştuğunu farz edelim. $X_i = (x_1, x_3, \dots, x_{2n-1})$ $X_2 = (x_2, x_4, \dots, x_{2n}) \in \{0, 1\}^n$ olsun. X_1 ve X_2 tarafından temsil edilen tam sayı değerleri sırasıyla şunlara eşittir: $\sum_{i=1}^n x_{2i-1} 2^{i-1}$ ve $\sum_{i=1}^n x_{2i} 2^{i-1}$, KARŞILAŞTIRMA fonksiyonu şu şekilde tanımlanır.

$$C(X_1, X_2) = \begin{cases} 1 & X_1 > X_2 \\ 0 & \text{Diğer} \end{cases}$$

Diğer bir değişle,

$$\begin{aligned} C(X_1, X_2) &= \text{sgn}[X_1 - X_2] \\ &= \text{sgn}[\sum_{i=1}^n 2^{i-1} (x_{2i-1} - x_{2i})] \end{aligned}$$

şeklinde ifade edilebilir.

KARŞILAŞTIRMA fonksiyonun LT_1 'ye ait olma fakat LT_1 'e ait olmama gibi ilginç bir özelliği vardır. Armonik analiz araçları kullanılarak, KARŞILAŞTIRMA'nın LT_2 'de olduğu gösterilmiştir. Sayfa [18]'de açıklanan metod kullanılarak KARŞILAŞTIRMA için bir LT_2 devresinin açık oluşumunu sağlıyoruz. KARŞILAŞTIRMA'nın açık oluşumu sayfa [2]'de açıklanmıştır.

KARŞILAŞTIRMA'nın değerinin, X_1 ve X_2 'nin farklılık gösterdiği en yüksek-düzey bit pozisyonuyla bulunduğunu unutmayınız. Eğer bu bit X_1 'de 1 ise ve X_2 'de 0 ise, o halde $C(X_1, X_2) = 1$. Ya da, $C(X_1, X_2) = 0$. ($X_1 = X_2$ iken, 59 tanımı $C(X_1, X_2)=0$.)

$F(X) = \sum_{i=1}^n 2^{i-1}(x_{2i-1} - x_{2i})$ olsun. Birinci basamak, 1'den büyük olan ağırlıkları üsrekli olarak yarıya bölüp 1 ve 0 arasında ağırlığa sahip ağırlıklar sağlayarak bir $F_i(X)$ fonksiyonları sırası oluşturmaktır. Aşağıdakiyle başlıyoruz.

$F_0(X) = F(X)$. n basamak sonra, bölme işlemi sıfıra eşit olan bir fonksiyon

$$F_0 = x_1 - x_2 + 2x_3 - 2x_4 + \dots + 2^{n-1}x_{2n-1} - 2^{n-1}x_{2n}$$

$$F_1 = x_3 - x_4 + \dots + 2^{n-2}x_{2n-1} - 2^{n-2}x_{2n}$$

...

...

$$F_{n-1} = x_{2n-1} - x_{2n}$$

$$F_n = 0$$

verecektir.

Her bölmenin hem X_1 hem X_2 'yi sola kaydırmaya eşit olduğunu unutmayalım.

Yardımcı Teorem 1. $F(X)$ 'inve yukarıda belirtildiği gibi sola kaydırılmış $F_l(X)$, $0 \leq l \leq n$ doğrusal fonksiyonların doğrusal kombinasyonları için,

$$F(X) > 0 \Leftrightarrow \exists l : F_l(X) = 1.$$

Kanıt. Her $X \in \{0, 1\}^{2n}$ için bir t mevcuttur, öyle ki tüm $l \geq t$ için, $F_l(X) = 0$. Şimdi kaydırmayla ortaya çıkacak en büyük hatayı göz önünde bulundurun Şunu

$$\max_{X \in \{0,1\}^{2n}} \max_{l > 0} |F_l(X) - 2 \cdot F_{l+1}(X)| = 1$$

görüyoruz.

Her kaydırmayla en fazla iki değişken (ağırlıkları 1 ve -1 arasında değişen "low-order bit"leri temsil eden) yok edildiğinden. Bu nedenle, $|F(X)|$ sıfır değilse, $|F_l(X)| = 1$ olacak bir l mevcuttur. Eğer $F(X)$ pozitifse, o halde tüm $F_l(X)$ ler pozitif veya

sıfırdır ve eğer $F(X)$ negatifse, o halde $F_l(X)$ ler negatif veya sıfırdır. Sonuç aşağıdaki gibidir.

$X_1 = (x_1, x_3, \dots, x_{2n-1}), X_2 = (x_2, x_4, \dots, x_{2n}) \in \{0,1\}^n$ olsun. Her $0 \leq l \leq n$ için “test” 60 fonksiyonunu aşağıdaki gibi tanımlayın.

$$T_l(X) = \begin{cases} 1 & \dots F_l(X) = 1 \\ 0 & \text{Diğer} \end{cases}$$

Yardımcı Teorem 1 şu şekilde

$$F(X) > 0 \Leftrightarrow \bigvee_{l=0}^n T_l(X) = 1$$

tanımlanabilir.

Deneme olmasına rağmen, modulo asal sayılarını hesaplama fikriye kullanıldığında Yardımcı Teorem 1 yararlı hale geliyor. Modulus işlemini 0’da ortalananmış simetrik bir aralıkta değerler vermesi için tanımlıyoruz. Örneğin bir tam sayı Z için ve pozitif tamsayı k için, $t \in [-k, k]$ ve $t \equiv Z \pmod{2k-1}$ olduğunda, $Z \pmod{2k-1} = t$ olsun. p prime verildiğinde, aşağıdaki gibi her $0 \leq l \leq n$ için bir test işlemi modulo p tanımlayın.

$$T_{pl}(X) = \begin{cases} 1 & \text{eger } F_l(X) \pmod{p} = 1 \\ 0 & \text{diğer} \end{cases}$$

Bir $X \in \{0, 1\}^{2n}$ için ve bir asal p için, sıradaki tüm $F_l(X)$ fonksiyonları için $T_{pl}(X)$ hesapladığımızı varsayalım. Test işlem modulosu asal p her zaman doğru cevabı vermediğinden, bu yeterli olmayacaktır. Fakat aşağıdaki Yardımcı Teorem bize yeterli asal sayıda tekrar edersek, örneğin r kadar, çoğu zaman doğru yanıtı alacağımızı söylemektedir.

Yardımcı Teorem 2. $p_1 < p_2 < \dots$ 3’ten büyük ardışık asallar olsun. $s, p_1 p_2 \dots p_s > 2^{n+l} - 1$ sağlayan en küçük tamsayı olsun. O halde $|Z| < 2^n - 1$ olan her Z tamsayısı için,

$$Z \in [-1, 1] \Rightarrow Z \pmod{p_i} \in [-1, 1] \text{ tüm asallar } > 3,$$

$$Z \notin [-1, 1] \Rightarrow Z \pmod{p_i} \in [-1, 1] \text{ s’den küçük asallar } > 3.$$

Kanıt. Birinci ifade denemedir. İkincisi direkt olarak Chinese Remainder Theorem (bakınız [18])’den alıntıdır. $s = O(n \log n)$.

$X \in \{0, 1\}^{2n}$ için ve bir grup asal sayı $p_1 p_2 \dots p_r$ için, $T_{p_{i,l}}(X)$ $1 \leq i \leq r$ ve $0 \leq l \leq n$ için olduğunu hesaplayan bir elemanlar bütününe sahip olduğumuzu düşünelim.

$T_{p_1,0}(X)$	$T_{p_1,1}(X)$	$\dots$	$T_{p_1,n}(X)$
$T_{p_2,0}(X)$	$T_{p_2,1}(X)$	$\dots$	$T_{p_2,n}(X)$
$\vdots$	$\vdots$	$\ddots$	$\vdots$
$T_{p_r,0}(X)$	$T_{p_r,1}(X)$	$\dots$	$T_{p_r,n}(X)$

$F(X) < 0$ olduğunda bir elemanın 1’e döndüğü bir “yanlış” pozitif belirleyin. $F(X) > 0$ olduğunda bir elemanın 1’e döndüğü bir “doğru” pozitif belirleyin.

- $F(X) > 0$ olduğunda, Yardımcı Teorem 1 her satırda en az bir gerçek pozitif olduğunu söylemektedir. Bu nedenle dizide toplamda en az r tane gerçek pozitif vardır.

- $F(X) < 0$ olduğunda, Yardımcı Teorem 2 her sütunda $3 \cdot s$ ’den az yanlış pozitifin olduğunu söylemektedir. Bu nedenle dizide toplamda $3 \cdot s \cdot n$ ’den az yanlış pozitif bulunmaktadır.

Eğer $r = 3 \cdot s \cdot n$ olduğunu seçersek, $F(X) < 0$ olduğunda 1’e dönen elemanların sayısı her zaman r ’den küçük olacaktır, aynı zamanda $F(X) > 0$ olduğunda 1’e dönen elemanların sayısı r ’ye eşit ya da r ’den küçük olacaktır. Buradaki önemli nokta yanlış pozitiflerin sayısının üst sınırının satır sayısından ve gerçek pozitiflerin sayısının alt sınırının sütun sayısından bağımsız olmasıdır.

KARŞILAŞTIRMA için, test elementlerini bir LT geçişine girdi olarak bağlayabiliriz ve geçişin eşiğini r olarak belirleyebiliriz. Buradaki sorun tek bir tabaka küçük ağırlıklı eşik geçişleri kullanarak, test elementlerini nasıl gerçekleştireceğimizdir.

Bu yaklaşım eşik devre teorisinde standardtır. $1 \leq i \leq r$ ve $0 \leq l \leq n$ için, $F_{p_{i,l}}(X)$ ’i, $F_l(X)$ modulo p_i ’nin ağırlıkları düşürülerek elde edilen doğrusal kombinasyon olarak tanımlayınız.

Her $X \in \{0, 1\}^{2n}$ için,

$$T_{p_{i,l}}(X) = 1 \Leftrightarrow F_{p_{i,l}}(X) \bmod p = 1.$$

Şimdi $F_{p_{i,l}}(X)$ en fazla $n \cdot p$ farklı değer üstleniyor. Bunların en fazla n tanesi modulo p_i alındığında 1’e eşittir. $F_{p_{i,l}}(X)$ ’in değerlerini $v_1, v_2, \dots, v_n$ gibi azaltılmış

modulo p_i 1 iken gösteriniz. Her V_j , $1 \leq j \leq n$ için, ilk tabakaya iki LT geçişi yerleştirip bunlara $G_j^{(1)}(X)$ ve $G_j^{(2)}(X)$ diyoruz.

- $G_j^{(1)}$ ve $G_j^{(2)}$, deki girdi telleri üzerindeki ağırlıklar $F_{p_{i,j}}(X)$ 'nin ağırlıklarına eşit olarak eşitlenmiştir.
- $G_j^{(1)}$ ve $G_j^{(2)}$ 'nin geçişleri sırasıyla v_j ve v_{j+i} olarak ayarlanmıştır.
- $G_j^{(1)}$ ve $G_j^{(2)}$ 'nin çıkış ağırlıkları üzerindeki ağırlıklar sırasıyla 1 ve -1, olarak

$$\sum_{j=1}^n (G_j^{(1)}(X) + G_j^{(2)}(X)) = T_{p_{i,j}}(X)$$

ayarlanmıştır.

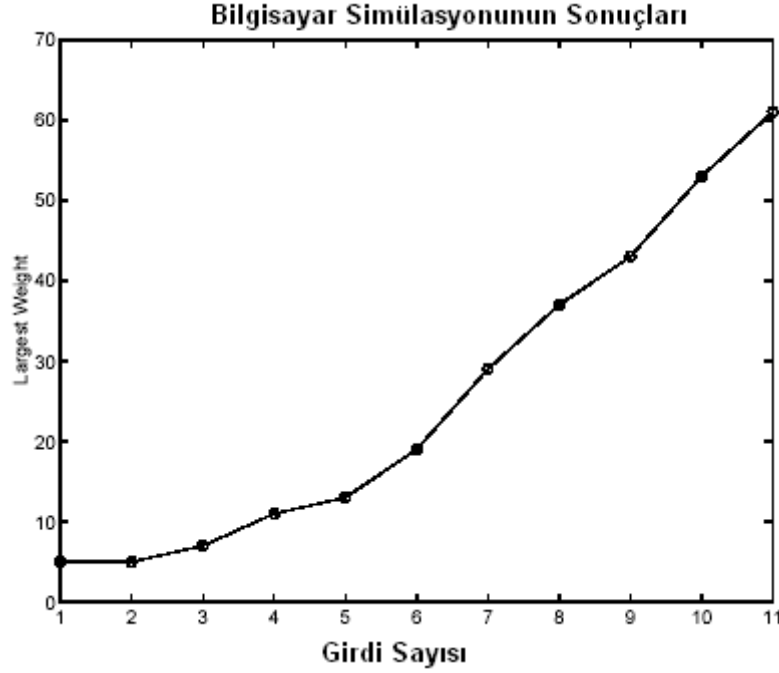
Sonuç olarak, toplamda, her birinin gerçekleştirilmek için $2 \cdot n$ LT geçişine ihtiyacı olan $3 \cdot s \cdot n^2$ tane test elemanımız var. Yani toplamda $6 \cdot s \cdot n^3$ tane LT geçişi lazım. $s = O(n \log n)$ olduğundan, yapımızın büyüklüğü şudur: $O(n^4 \log n)$.

4.3 Bilgisayar Simülasyonu

KARŞILAŞTIRMA'nın LT_2 yapısını simüle etmek için, ekte gösterilen kısa bir Matlab programı kullandık. Her n (değişken sayısının yarısı) için, en küçük asal sayısını ve düzgün bir devre oluşturan en küçük eşiği bulduk. Aşağıdaki tablo sonuçları göstermektedir:

n	Asal Sayısı	Eşik
1	1	0
2	1	0
3	2	1
4	3	2
5	4	3
6	6	5
7	8	7
8	10	9
9	12	11
10	14	13
11	16	15

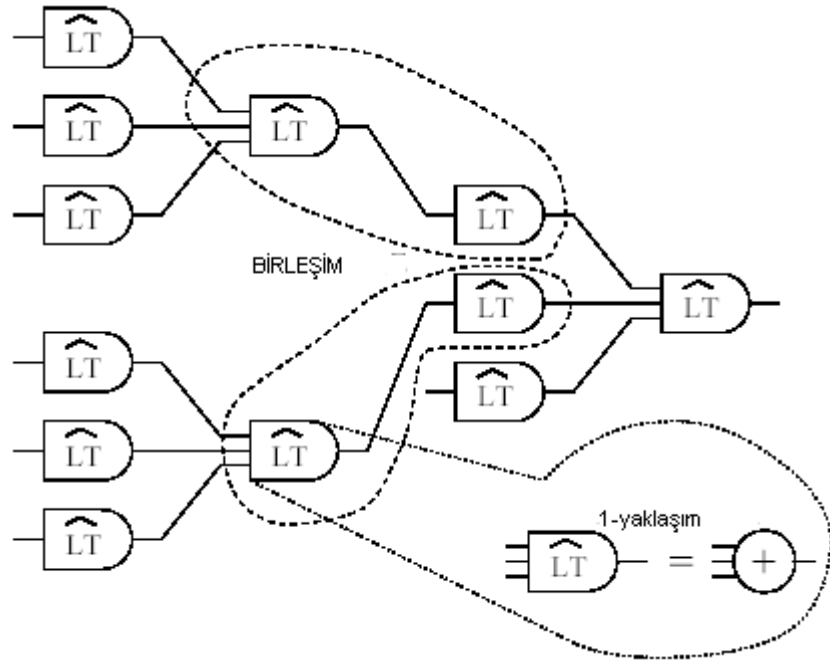
Devredeki en büyük ağırlığı girdi sayısının fonksiyonu olarak çiziyoruz.



4.4 $LT_d \subset \hat{L}T_{d+1}$ 'e Genelleme

"1-yaklaşım," denilen yukarıdaki fonksiyonun önemli bir özelliği çıktı eşik geçişinin ağırlıklı toplamla değiştirilebilmesidir. (i.e., 64 $sgn(.)$ fonksiyonunu iptal edin) böyle bir devrenin çıktısı 0'a ya da 1'e çok yakın olacaktır. Bu durum sayfa [13]'te $LT_d \subset LT_{d+1}$ olduğunu göstermek için kullanılmıştır. Fikir şu şekildedir:

1. LT_d devresinin çıkış geçişini, G_{last} , alalım. Bir LT geçişi olduğu için Bölüm 4.2de görüldüğü gibi bir LT_2 geçişi de onun yerine geçebilir.
2. Ona bağlı olan bütün geçişleri alıp, $G_{last-1}^{(i)}$, onların da yerlerini değiştiriyoruz.
3. Çıkış tabakası $G_{last-1}^{(i)}$ ile giriş tabakası $G_{last}^{(i)}$ 'yi birleştirmek için "1- yaklaşım " özelliğini kullanın.
4. Tüm LT geçişlerinin yerine LT_2 devreleri gelene kadar işleme devam edin.



LT_{d+i} devresiyle bir LT_d fonksiyonu simüle etmek için tabakaları birleştirin.

4.5 Sonuç

$LT_d \subset LT_{d+i}$ çok yararlı bir sonuçtur.. $MADD \in LT_2$ -65 anlamına gelen $MADD \in LTM$ oluşturarak, $LTM \in LT_2$ olduğunu göstermek için kullanabiliriz. Bu çalışmanın ilginç bir yönü ise Bölüm 4.2'de belirtilen yaklaşımı kullanarak KARŞILAŞTIRMA için LT_2 uygulamasının büyüklüğünün elde etmektir.

Bölüm 5

Çok Eşikli Doğrusal Eşik Elementi LTM

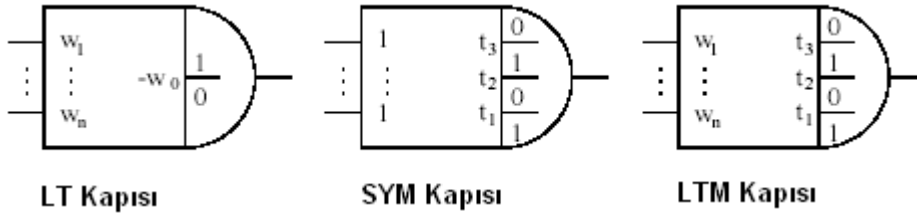
5.1 Giriş

LT elemanlarının VLSI uygulanması sayfa [8] üzerine yaptığımız çalışmadan ilham alarak, bu bölümde LTM çok geçişli doğrusal eşik bkz: Sayfa [15] ve [35],

denilen daha güçlü bir hesaplama elemanından bahsedeceğiz. *LT* elemanındaki işaret fonksiyonu yerine, girdilerinin ağırlıklı toplamının keyfi (polinom olarak çok geçişler) bir doğru yanlış fonksiyonu hesaplanır.

Çalışmadaki LTM devreleri(LTM elemanlarından oluşan devreler) hakkındaki konular, hesaplama kapasitelerinin, sınırlamalarının hesaplanması ve özelliklerinin AON devrelerinin özellikleriyle karşılaştırılmasıdır. Bu çalışmadaki yaklaşım, *LT* devreleri ve *LTM* devreleri arasındaki ilişkiyi anlamaktan geçer. Bu bölümdeki ana katkılarımız şunlardır:

- m tamsayılarının eklenmesi eve iki tamsayının ürünleri için LTM devrelerinin etkin tasarımlarını oluşturarak, LTM'nin gücünü göstermekteyiz.
- LTM devrelerinin uygulamadan LT devrelerinden daha rahat olduğunu gösteriyoruz. Özellikle n girdili doğru yanlış fonksiyonları için VLSI planı *LT* devrelerindeki $O(n^2)$ 'dan LTM devrelerindeki $O(n)$ 'ye düşürüldüğündeki alanlarda.
- LTM devrelerinin hesaplama gücünü LT devreleriyle karşılaştırarak belirtiyoruz.



Şekil 5.1: *LT*, *SYM* ve *LTM* hesaplama elemanlarının şema halinde gösterimi. Daha sonra *LT* ve *LTM* elemanlarının resmi tanımlarını açıklayacağız.

5.1.1 Tanımlar ve Örnekler

Tanım 5.1 (Doğrusal Eşik Geçişi - LT)

Bir doğrusal eşik geçişi iki rakamlı girdileriyle bir doğru yanlış fonksiyonunu hesaplar

$$f(X) = \text{sgn}(w_0 + \sum_{i=1}^n w_i x_i)$$

w_i tamsayılar ve eğer durumu 0'a eşitse ya da 0'dan büyükse $\text{sgn}(\cdot)$ çıktı olarak 1 verir, eğer durum öyle değilse 0 verir.

Sekil 5.1.1 bir n -girdili *LT* elemanı; eğer $\sum_{i=1}^n w_i x_i \geq -w_0$ ise, elemanın çıktı olarak 1 verdiğini, değilse çıktı olarak 0 verdiğini gösterir. İkincisi simetrik fonksiyonların genel sınıfına dahildir-SYM.

Tanım 5.2 (Simetrik Fonksiyonlar - SYM)

Bir doğru yanlış fonksiyonu eğer değeri girdide bulunan 1'lere ($|X|$ olarak gösterilir) bağlıysa

$$|X| = \sum_{i=1}^n x_i$$

simetriktr.

Şekil 5.1.1 simetrik fonksiyonun bir örneğini göstermektedir; 3 geçişi vardır, $|X| < t_1$ ve $t_2 \leq |X| \leq t_3$ için sonuç olarak 1 verir, diğer durumlarda 0 verir. EVET, VEYA ve EŞİTLİK simetrik fonksiyonların örnekleridir. Tek bir LT elemanı sadece simetrik fonksiyonların sınırlı altkümelerini ifade edebilir. LTM 'yi SYM 'nin genellemesi olarak tanımlıyoruz. Çünkü ağırlıkları 1'e sabitlemek yerine LT 'de olduğu gibi onların değişken olmalarına izin veriyoruz (bkz: şekil 5.1.1.)

Tanım 5.3 (Çok Geçişli Doğrusal Eşik Geçişi - LTM)

Eğer ağırlık kümesi $w_i \in \mathbb{Z}, 1 \leq i \leq n$ ise f fonksiyonu LTM 'dedir. Ve fonksiyon $h : \mathbb{Z} \rightarrow \{0,1\}$ öyle ki

$$f(X) = h\left(\sum_{i=1}^n w_i x_i\right) \text{ her } X \in \{0,1\}^n \text{ için}$$

girdisi $\left[-\sum_{i=1}^n |w_i|, \sum_{i=1}^n |w_i|\right]$ taradığı zaman polinomial kadar geçiş aşamasından geçmesi h 'yi sınırlandıran tek şeydir.

Geçiş sayısında bir kısıtlama olmadığında bir LTM geçişi herhangi bir doğru yanlış fonksiyonunu hesaplayabilecek kapasiteye sahiptir. Hatta keyfi fonksiyon f , $w_i 2^{i-1}$ ve $h\left(\sum_{i=1}^n 2^{i-1} x_i\right) = f(x_1, \dots, x_n)$ olsun.

Örnek 5.1 ($XOR \in LTM$)

$XOR(X)$ 1 sonucunu verir, eğer $|X|$ ise ve X 'in içindeki 1'lerin sayısı tekse.

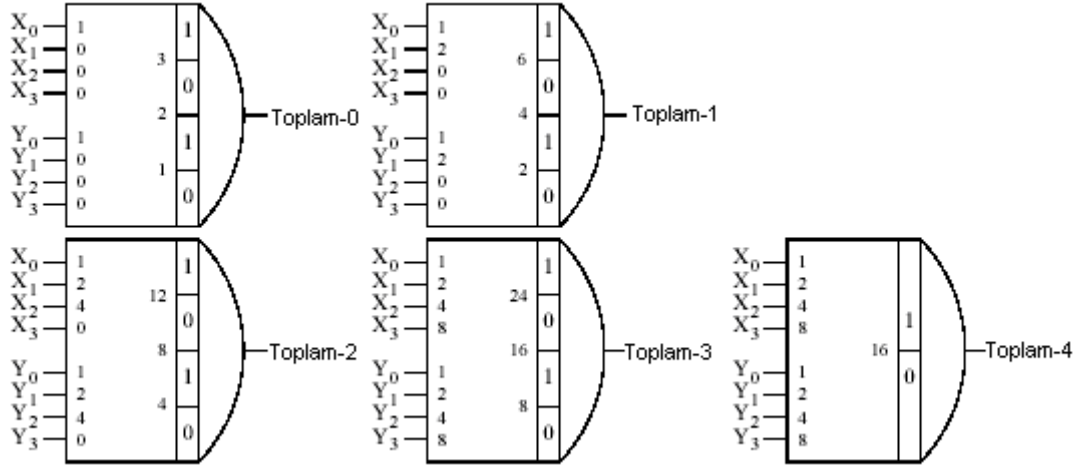
Uygulamak için $0 \leq k \leq n$ için, $w_i = 1$ ve $h(k) = \frac{1}{2}(1 - (-1)^k)$ olmasını seçin. $h(k)$, k

< 0 ve $k > n$ için tanımlanmak zorunda değildir. Ve polinomial pek çok geçişi vardır. LTM 'nin hesaplayabildiği başka yararlı bir fonksiyon da $ADD(X, Y)$ 'dir, iki n -bit tamsayı olan X ve Y 'nin toplamı.

Örnek 5.2 ($ADD \in LTM$) Toplamayı uygulamak için

$$f_l(X, Y) = h_l\left(\sum_{i=1}^l 2^i (x_i + y_i)\right)$$

$k \in [2^l, 2 \times 2^l - 1] \cup [3 \times 2^l,)$ için $hi(k) = 1$ olduğunda $X + Y$ 'nin m . bitini hesaplar. Şekil 5.2 $n = 4$ olduğunda ortaya çıkanların örneğini göstermektedir.



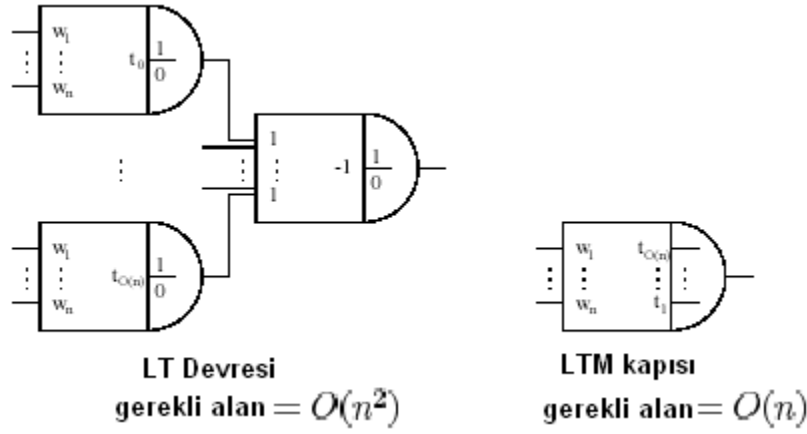
Şekil 5.2: Her çıktı biti için tek bir LTM geçişi kullanarak iki 4-bit-tamsayının toplamı.

5.1.1 Düzenleme

Bu bölüm şu şekilde düzenlenmiştir. Bölüm 5.2'de, LTM devrelerinin uygulamalarını göreceğiz. Özellikle, tek tabakalı LTM elemanlarıyla m tamsayılarının nasıl toplanacağını göreceğiz. Bölüm 5.3'te, LTM'nin tanımlama sonuçlarını kanıtlayacağız, kapsama ilişkileri ve özellikle $LTM \subseteq LT_2$. Ayrıca hangi kapsamaların uygun olduğunu göstereceğiz ve farklılıkları belirtmek için fonksiyonlardan yararlanacağız.

5.2 LTM Yapıları

LTM hakkındaki teorik sonuçlar doğru yanlış fonksiyonlarının VLSI uygulanmasında uygulanabilir. Çoklu eşikleri olan geçiş fikri aklımıza simetrik doğru yanlış fonksiyonlarının etkin bir VLSI uygulamasını ararken geldi. Tek bir LT geçişi simetrik doğru yanlış fonksiyonlarını uygulamada yetersiz olsa bile, iki katmanlı bir LT devresi yeterlidir(Şekil 5.2). Ayrıca böyle bir devrenin VE, VEYA ve DEĞİL üzerine kurulmuş geleneksel mantık devresinden çok daha iyi çalıştığı bilinmektedir. Ayrıca geleneksel devrelerin üstel büyüklükleri vardır (ya da sınırsız derinliği) [51].



Şekil 5.3: LT devresi, $O(n)$ büyüklüğünde vs tek bir LTM geçişi.

Öneri 5.1 (simetrik fonksiyon uygulaması için LT_2 vs LTM)

Bir simetrik fonksiyonun LT_2 düzeni için $O(n^2)$ 'lik bir alana ihtiyaç vardır; oysa LTM için sadece $O(n)$ 'lik bir alana ihtiyaç vardır.

KANIT:

LT_2 'de genellenmiş simetrik bir fonksiyonu uygulamak için birinci tabakada n tane LT geçişine ihtiyaç vardır. w_0 geçişi dışında hepsi aynı w_i ağırlığına sahiptir. Aynı doğrusal toplam olan $\sum_{i=1}^n w_i x_i$ n kere tekrarlamaktansa, bunu bir kere yapıyoruz ve sonuçları n tane farklı geçişle karşılaştırıyoruz. Sonuç olarak ortaya çıkan devre tek bir LTM geçişine tekabül eder.

Şekil 5.2 genellenmiş bir simetrik fonksiyonun uygulanmasından LTM 'nin LT 'ye göre olan avantajlarını göstermektedir. Aslında, LT_2 düzeni gereğinden fazladır, her ağırlığın n tane kopyası vardır ve en az $O(n^2)$ 'lik bir alana ihtiyaç vardır. Diğer taraftan LTM tek bir ağırlıklı toplam vermektedir ve alan ihtiyacı $O(n)$ 'dir. Tek bir LTM geçişi m tane n -bit tamsayısı olan $MADD$ 'in toplamını hesaplayabilir. Tek sorun m 'nin n 'de polinomial olmasıdır.

Teori 5.1 ($MADD \in LTM$)

LTM geçişlerinin tek bir katmanı, m 'nin n 'de en polinomial halinde olması şartıyla, m tane n -bit tamsayıyı toplayabilir.

KANIT:

$MADD$ en fazla $n + \log m$ bitteki tamsayıları vermektedir. Her bit için bir tane LTM geçişine ihtiyaç vardır. En az önemli bit basit bir m -bit XOR ile hesaplanmaktadır. Diğer tüm bitler için

$$f_l(X^{(1)}, \dots, X^{(m)}) = h_l(\sum_{i=1}^l 2^i \sum_{j=1}^m x_i^{(j)})$$

toplamın l . bitini hesaplamak için kullanıyoruz.

Örnek 5.3 (üç 3-bit tamsayının toplamı)

Yukarıdaki yapıyı $m = 3$, $n = 3$ 'e uyguluyoruz. Sonuç şekil 5.2'de gösterilmiştir. Sonucun $\{0,...,21\}$ aralığında olduğuna dikkat ediniz. Bu nedenle, çıktı bitini hesaplayan LTM geçişinin sadece 2 tane eşiğe ihtiyacı vardır.

Sonuç 5.1 (PRODUCT $\in$ PTM) PTM geçişlerinin tek bir katmanı (yukarıda açıklanan), m'in n'de en polinomial halde olması şartıyla, m tane n-bit tamsayının ürününü hesaplayabilir.

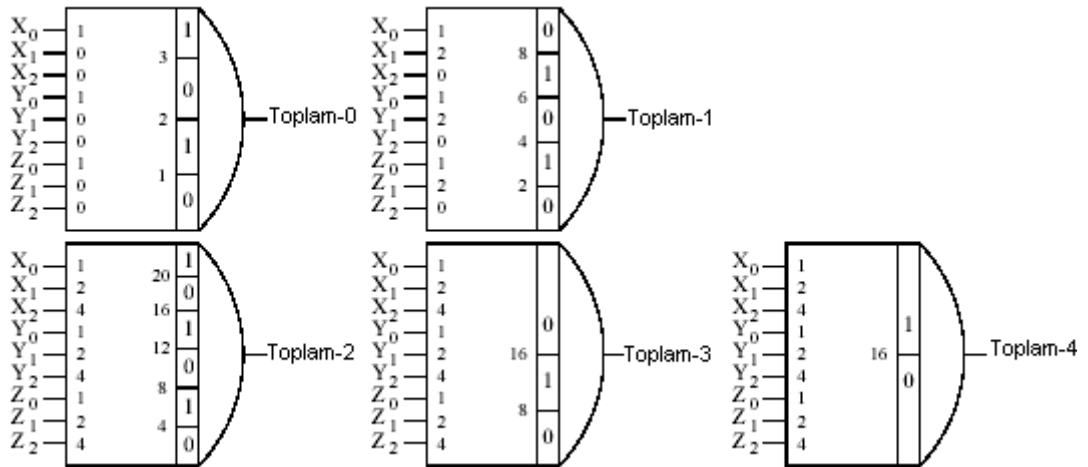
KANIT:

Sayfa [9]'daki PT_1 'le benzer şekilde, PTM_1 'de (ya da sadece PTM) doğrusal yoğunluk polinomial sonuca izin veriyoruz.

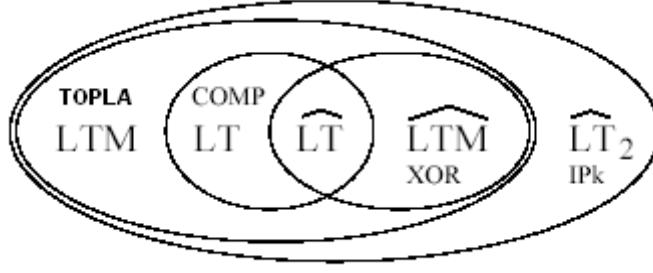
$$f(X) = h(w_1X_1 + ... + W_nX_n + W_{(1,2)} x_1x_2 + ...)$$

Fakat sonucun polinomial olacak kadar çok olmasına izin vermiyoruz(yoksa herhangi bir doğru yanlış fonksiyonu tek bir geçişle gerçekleştirilebilir). İki n-bir tamsayı olan X ve Y'nin ürünü $PRODUCT(X,Y) = \sum_{i=1}^n x_i Y$ şeklinde yazılabilir. PRODUCT'ı uygulamak için MADD yapısını kullanıyoruz.

$$PRODUCT(X,Y) = MADD(x_1Y, x_2Y, ..., x_nY,)$$



Şekil 5.4: MADD: üç tane 3-bit tamsayı olan X, Y ve Z'nin toplaması – LTM elemanlarının bir tabakası kullanılarak.



Şekil 5.5: Sınıflar arasındaki ilişki.

$$f_l(X, Y) = h_l\left(\sum_{j=1}^n \sum_{i=1}^n 2^i x_j y_i\right)$$

f_l ürünün l -inci bitini vermektedir.

5.2 LTM'nin Sınıflandırılması

Küçük ağırlıkları (polinomial olarak büyüyen) göstermek için bir şapka kullanıyoruz, örneğin LT , LTM [6], [43] ve tek bir tabaktan oluşmayan devrenin derinliğini (tabakalarının sayısını) belirtmek için altsimge kullanıyoruz. Bu çalışmada geçen devrelerin hepsinin polinomial büyüklüğü (elemanların sayısı) n (girdi sayısı) cinsindendir. Örneğin LT_2 sınıfı LT elemanlarının derinlik-2 polinomial büyüklüğe sahip olanlarıyla uygulanabilen doğru yanlış fonksiyonlarından oluşur.

Şekil 5.3 doğru yanlış fonksiyonlarının 5 sınıfı, LT , $\hat{LT}$, LTM , $\hat{LTM}$ ve LT_2 , ile farklılıkları kurmak için kullanılan fonksiyonların ilişkisini anlatır.

Bu bölümde Şekil 5.3'te gösterilen ilişkileri kanıtlayacağız.

Teori 5.2 (LTM'nin Sınıflandırılması)

Şekil 5.3'de gösterilen kapsamalar ve farklılıklar aşağıdadır;

- $\hat{LT} \subseteq LT \subseteq LTM$
- $\hat{LT} \subseteq \hat{LTM} \subseteq LTM$
- $LTM \subseteq \hat{LT}_2$
- $XOR \in \hat{LTM}$ ama $XOR \notin LT$
- $COMP \in LT$ fakat $COMP \notin \hat{LTM}$
- $ADD \in LTM$ fakat $ADD \notin LT \cup \hat{LTM}$
- $IP_k \in \hat{LT}_2$ fakat $IP_k \notin LTM$

5.4 Sınıflandırma Teorisinin Kanıtı

Şekil 5.3'te gösterilen ilişkileri kanıtlayalım. İlk başta kapsama ilişkilerini görelim. Daha sonra sınıflar arasındaki farklılıkları gösteren fonksiyonların üzerinde duracağız.

5.4.1. Kapsamalar

Çoğu kapsama ilişkisi şu tanımlardan gelirler: $\hat{L}T \subseteq LT \subseteq LTM$ ve $\hat{L}T \subseteq \hat{L}TM \subseteq LTM$. Sadece bir tanesinin bir kanıtı ihtiyacı vardır:

$$\underline{LTM \subseteq \hat{L}T_2}$$

Yukarıdaki ifadeyi göstermek için sayfa [13]'den bir sonuç kullanacağız. Keyfi ağırlıklı tek bir LT geçişi, bir LT_2 devresiyle gerçekleştirilebilir. Ayrıca ikinci tabakadaki doğrusal olmama durumu devrenin çıktılarını etkilenmeden kaldırılabilir (1-yaklaşım denilen özellik). p n'de polinomial iken ve tüm i'ler için $f \in LT$ iken

$$f_i \in \hat{L}T, f(X) = \sum_{i_1}^p w_i f_{i_1}(X)$$

LTM 'deki bir fonksiyonun LT_2 uygulamasını düşünelim. Birbirinin aynısı olan LT geçişlerinden oluşan bir tabaka ve onları takip eden 1 ve -1 ağırlığında tek bir geçiş ve -1 eşiğinden oluşur. Birinci tabakadaki her LT geçişini, eşiti olan LT geçişleri ve ağırlıklı toplamıyla değiştiriyoruz. Ağırlıklı toplamı birleştiriyoruz. Örneğin ikinci ve üçüncü tabakalar, ortaya çıkan devre LT_2 'de olacaktır.

5.4.2 Farklılıklar

Birinci örnekte $XOR \in \hat{L}TM$ olduğunu ve $XOR \notin LT$ olduğunu bilindiğini gördük. Diğer taraftan, $COMP(X, Y)$, iki tane n-bit tamsayının karşılaştırmasının LT'de

$$COMP(X, Y) = \text{sgn}\left(\sum_{i=1}^n 2^i (x_i - y_i)\right) = \begin{cases} 1 & Y \leq X \text{ ise} \\ 0 & \text{Diğer} \end{cases}$$

olduğunu gördük.

$COMP \notin \hat{L}TM$ olduğunu gösterelim. Bu nedenle bir doğru yanlış fonksiyonun *entropi* özelliğinden bahsedeceğiz. İletişimin karmaşıklığı hakkında eşit bir tanım sayfa [47]'de verilmiştir.

Tanım 5.4 (Entropi)

Doğru yanlıf fonksiyonu için n -değişkeni, S bu değişkenlerin altkümesi ve $s \in \{0,1\}^{|S|}$, $f_s(x_1, \dots, x_{n-|S|})$ f'de s 'nin değeri S 'nin yerine konularak elde edilen fonksiyon. f 'nin entropisi

$$E[f] = \max_s |f_s : s \in \{0,1\}^{|S|}|$$

şeklinde tanımlanır.

Entropi n değişkenin mümkün $2^{|S|}$ tüm değerlerini S kümesine atayarak elde edebileceği alt fonksiyon üzeri $n - |S|$ değişkenlerin maksimum sayısıdır.

Yardımcı Teorem 5.1 (Üstel Entropi, Üstel Ağırlıklar Demektir)

Öyle bir f fonksiyonu ki, $E[f]$ n'de üstel, LTM uygulaması üstel ağırlıkları gerektirmekte. Örneğin $\sum_i^n w_i$ üstel.

Kanıt: Bir alt fonksiyon $W_s \in \sum_{i \in S} w_i s_i$ olduğunda

$$f_s(x_1, \dots, x_{n-|S|}) = f(X, S=s) = h\left(\sum_{i \in X-S} w_i x_i + W_s\right)$$

şeklinde yazılabilir.

Pigonhole prensibine göre ve W_s bir tamsayı ise, $|\{W_s : s\}| \geq E[f]$ den büyük olmalıdır. Eğer değilse W_s 'nin tüm $E[f]$ özel alt fonksiyonlarını tanımlamaya yeterli özel değeri olmayacaktır. Bu da

$$E[f] \leq \sum_{i \in S} |w_i| \leq \sum_{i=1}^n |w_i|$$

demektir.

$$\underline{COMP \notin \hat{LTM}}$$

Kanıt: $E[COMP]$ 'nin üstel olduğunu gösteriyoruz ve 5.1'i kullanıyoruz.

$$f_s(x_1, \dots, x_n) = COMP(X, Y=s) \text{ olsun.}$$

bu fonksiyonlardan 2^n tane mevcut; hepsinin özel olduğunu gösterelim. İki ayrı tamsayı s_1 ve s_2 , öyle bir X_0 seçin ki $s_1 < X_0 < s_2$, ve sonra $f_{s_1}(X_0) \neq f_{s_2}(X_0)$.

$$ADD \in LTM \text{ fakat } ADD \notin LT \cup \hat{LTM}$$

Kanıt: $ADD \in LTM$ olduğunu gördük. Toplamın en az önemli olan kısmı LT'de olmayan XOR'dur. Diğer taraftan $ADD \notin \hat{LTM}$ anlamına gelen COMP'a benzer bir kanıtı olan $E[ADD]$ üsteldir.

$$IP_k \in L\hat{T}, \text{ fakat } IP_k \notin LTM$$

Kanıt: Let $IP(X, Y) = \sum_{i=1}^n x_i y_i$ olsun. Fonskyionu $IP_k(X, Y) = 1$ sadece ve sadece $IP \geq k$, ya da $IP_k = 0$ olarak tanımlayın. $IP_k \notin LTM$ olduğunu iddia ediyoruz. Hatta IP_k LTM'nin içinde olsaydı, ağırlıklı toplam tarafından takip edilen bit LT geçişleri tabakasıyla uygulanabilirdi[13]. O zaman IP_2 (Inner Product mod 2)'yi yanlış olarak bilinen $L\hat{T}_2$ 'de uygulamak için $k = 1..n$ için devreleri birleştirebilirdik.

Sınıflandırmayı tamamlamak için eksik olan şey şudur: $\hat{LT} = LT \cap \hat{LTM}$. Bunun doğru olduğunu varsayıyoruz.

Varsayım 5.1 ($\hat{LT} = LT \cap \hat{LTM}$)

LT değişken değerli doğrusal eşik fonksiyonları sınıfını temsil etsin ve $\hat{LT}$ ağırlıklarında Polinomial büyüme olan fonksiyonların sınıfı olsun ve LTM de Tanım 5.1'de belirtildiği gibi olsun böylece,

$$\hat{LT} = LT \cap \hat{LTM}$$

5.5 Sonuçlar

Asıl amacımız etkin bir genellenmiş simetrik fonksiyon ortaya koymak için teorik sonuçları kullanmaktı. Bu işlem süresince, LT_2 uygulamasının gereğinden fazla olduğunu anladık. Bu da bizi LTM'nin tanımına, yeni ve daha güçlü bir hesaplama elemanına, götürdü. LTM'nin gücünü LT'ye göre sınıflandırdık. VLSI düzenlerinin alanlarının $O(n^2)$ 'dan $O(n)$ 'ye düşürmede ve çoklu toplama ve üründe etkili tasarımlar elde etmede nasıl kullanılabileceğini belirttik. Gelecek çalışmalar için bazı ilginç talimatlar ise şunlardır. (i) $\hat{LT} = LT \cap \hat{LTM}$ varsayımını doğrulama (ii) LTM'nin analizinde özellikle PTM'nin sınıflandırmada nereye düştüğünü göstermede(şekil 5.3), spektral teknikleri uygulama. Doğrusal karar listelerinden, LDL, Bölüm 2'nin sonuç kısmında(77), Bölüm 2.7'de bahsedilmiştir. LTM'nin LDL'nin bir parçası olduğunu görmek kolaydır yani $LTM \subseteq LDL$. İlginç bir problem $LTM \in LDL$ ya da $LTM = LDL$ oluşturmaktır.

Daha mümkün cevap olan öncekini kanıtlamak için, IP_k 'nin tüm LDL yapıları gösterilmeli ve gösterilen fonksiyonlar LTM'de olmamalıdır.

Gelecek çalışmalar için başka bir yön ise 6. Bölümde dile getirilmiştir. Yukarıdaki fikirleri VLSI kümesinde uygulamaktır. 2μ analog çipte, yukarıdaki modeli kullanarak programlanabilir bir genellenmiş simetrik fonksiyon ürettik. Yüzen geçiş teknolojisi ağırlıkları programlamak için kullanıldı. Yüzen geçişe elektronlar enjekte ederek ve elektronları oraya yönlendirerek tek bir geçişe ağırlık yükledik.

Bölüm 6

Programlanabilir Sinirsel Mantık ile VLSI Uygulanması

6.1 Giriş

Neuromorphic analog VLSI alanında, çoğu araştırma bir şekilde öğrenen veya adapte olan sinirlerin uygulanmasıyla ilgilenmekte, sayfa [11], [16], [19]. Bunun nedeni sinirsel sistemlerin gücünün adapte olma yeteneklerinden kaynaklandığına inanılmasıdır. Bir sinir tarafından yerine getirilen fonksiyonun ağırlıklı girdilerin toplamı ve daha sonra bir eşik kendi kendine(öğrenmeden) bir yapı bloğu olduğu kanıtlanmıştır. Uzun yıllar boyunca teorik bilgisayar bilimi bu tarz nöronların gücünü, polinomial büyüklükteki vs üstel büyüklükteki devreler ve NP'nin tam olması sorunuyla ilgili problemler açısından inceledi. Ana sorundogru yanlış girdisi, çıktısı ve eşik devreleri üretmek ve yararlı doğru yanlış fonksiyonlarını etkili bir şekilde hesaplama. Eşik devrelerinin şaşırtıcı derecede güçlü oldukları görülmüştür[1]. Örneğin tamsayı bölümü, sabit derinliği olan bir polinomial büyüklükteki eşik devresiyle yapılabilir. Başka bir deyişle eğer iki n-bit tamsayının bölümünün hesaplanması eşik devreleriyle yapılmak isteniyorsa, polinomial olarak pek çok, n tane, eşik elementlerine ihtiyaç vardır. Diğer yandan VE, YA DA ve DEĞİL gibi geleneksel mantık devreleri kullanmak, üstel sayıda pek çok geçiş kullanmayı gerektirir. XOR ve tamsayı toplaması gibi daha basit fonksiyonlarda da durum aynıdır.

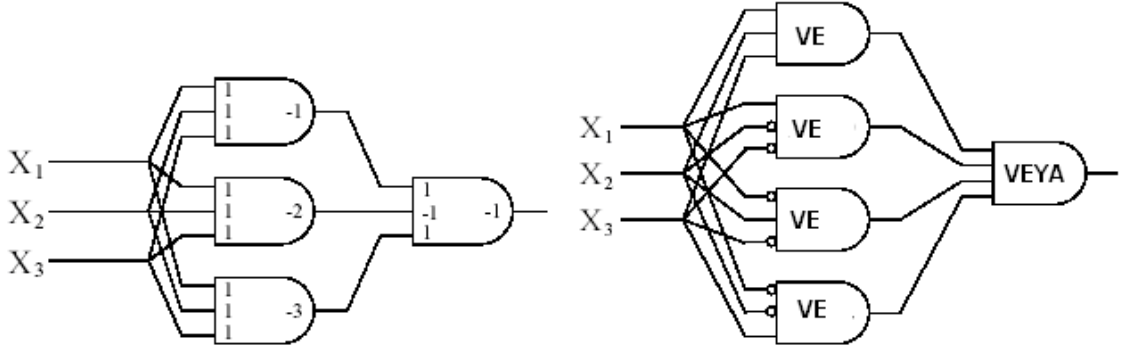
Eşik devreleri teorisinden elde edilen sonuçların çoğu, silikon üzerindeki devrelerde kullanıma uyarlanabilir. Ağırlıklar için en yüksek boyut ve sonuç olarak alınan elementin ya da devrenin gücü arasındaki ilişki gibi sonuçlar[6], [13], *XOR*, *ADD*, *MULTIPLY* ve diğer yararlı fonksiyonların etkin tasarımları için bkz: Sayfa [24], [28], [31]. Örneğin teorinin basit bir uygulaması bizi çoklu eşik elementini tanımlamaya taşıdı, bkz: Bölüm 5. Bu, belirli doğru yanlış fonksiyonlarında özellikle *PARITY* gibi simetrik fonksiyonların alanını $O(n^2)$ 'den $O(n)$ 'ye indirdi.

Araştırmamızın 3 önemli amacı vardır:

1. Uygulama yönü. Silikon üzerinde etkili eşik elementleri tasarlamak ve uygulamak.
2. Teorik yönü. Yüksek performanslı eşik devrelerini sistematik bir şekilde tasarlamak için, teorik bilgisayar biliminde yapılmış çalışmalardan yararlanmak.
3. Programlanabilirlik yönü. Eşik elementlerini FPGA'larda yapı taşları olarak kullanmak.

Eşik devrelerinin kullanılması, 60'ların ve 70'lerin başında önerilmişti sayfa [4], [48], [53], ve daha yakın olarak bkz: sayfa [28], [39]. Bildiğimiz kadarıyla, eşik devrelerinin teorik sonuçları daha önce silikon kullanımını içeren başka bir çalışmayla ilişkilendirilmemişti. Programlanabilir nöron-bazlı donanım son zamanlarda sayfa [39], [41]'de önerilmiştir. Aşağıdaki uygulama bölümünde, bunların çalışmamızla olan ilişkisini anlatıyoruz. FPGA'nın kısa bir tekrarı için bkz: sayfa [50]. Bölüm 6.2'de eşik devrelerini geleneksel mantık devreleriyle karşılaştırıyoruz. Bölüm 6.3'te tasarımın programlanabilirlik yönüne

odaklanıyoruz. Bölüm 6.4'te VLSI uygulamasını ve test sonuçlarını görüyoruz. LTM elementi Bölüm 5'te ve sayfa [7]'de teorik bir açıdan anlatılmıştı. Geleneksel eşik devreleriyle ve (VE, YA DA, DEĞİL) devreleriyle karşılaştırılmıştı. Bölüm 6.5 2μ -teknoloji $2mm \times 2mm$ çipi üzerine LTM'nin kullanımını göstermektedir.



Şekil 6.1: Sinirsel vs. geleneksel mantık. XOR hesaplayan iki devre.

6.2 Sinirsel Mantık ve Geleneksek Mantık

Herhangi bir doğru yanlış fonksiyonu AON devreleriyle sistematik olarak çözülebilirken neden eşik elementleri kullanalım ki? (XOR) gibi bazı fonksiyonlarda girdideki bit sayısı arttıkça AON devresindeki elementlerin sayısı da üstel olarak artacaktır[51]. Diğer taraftan, doğrusal eşik elementleri kullanılıyorsa, geçişlerin sayısı, girdi bitlerinin sayısı içinde doğrusaldır. Bu Şekil 6.2de, 3-bit girdi için kullanılmıştır. Genellikle, n tane bitin XORunu hesaplayan derinlik-2 AON devresinin en az $2^{n-1} + 1$ tane geçişe ihtiyacı vardır. LT kullanılırsa, sadece $n + 1$ geçişe ihtiyaç vardır.

LT devrelerinin AON devrelerinden daha güçlü olduğunu görmek gayet kolaydır. Bunun nedeni her AON geçişi için, onun eşiti olan ve aynı fonksiyonu hesaplayan bir LT geçişinin mevcut olmasıdır. Oysa çoğu LT geçişinin AON eşitleri yoktur.

Örnek 6.1 (ÇOĞUNLUK) ağırlık vektörü $(w_0, ..., w_5) = (-3, 1, 1, 1, 1, 1)$ ile tanımlanmış bir fonksiyon olsun:

$$f(x_1, ..., x_5) = \text{sgn}(-3 + x_1 + x_2 + x_3 + x_4 + x_5).$$

Sadece bir adet 1 vermektedir, oysa 3 ya da daha fazla girdi 1'dir. Tek bir VE veya YA DA geçişiyle uygulanamaz, bazı girdileri reddetsek bile (DEĞİL).

LT devreleri daha güçlü oldukları halde, yapı taşlarının daha karmaşık olduğu ve bu nedenle devre düzeninde daha çok yer kaplayacakları tartışılabilir. Bu konu bir yere kadar doğrudur. Fakat ihtiyaç duyulan geçişlerin sayısında üstelden

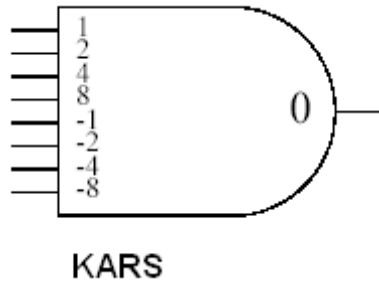
polinomial a giden düşüş, boyutlarındaki büyümeyi görünmez hale getirmektedir. Sıradaki bölüm bu konu üzerinde duracaktır.

6.3 Programlanabilir ve Fiziksel Bağlantılı Ağırlıklar

FPGA'ya mevcut fonksiyonların içinden seçilen, her elementin hesapladığı fonksiyon programlanabilir olan element devreleri olarak bakılabilir. Geleneksel FPGAlarda bu içinden seçilen küme VE, YA DA ve DEĞİL'den oluşmaktadır. Biz daha geniş bir fonksiyonlar kümesi sunuyoruz, Doğrusal Eşik Fonksiyonları, LT. Bir LT geçişi hakkındaki tüm bilgi eşikte ve ağırlıklarda saklıdır. Ağırlıkları kullanmanın 2 yolunu göz önünde bulunduruyoruz.

- Fiziksel bağlantılı ağırlıklar bir transistörün en/boy oranında kodlanmıştır.
- Programlanabilir ağırlıklar, yüzen geçiş üzerine kalıcı bir yük olarak konmuştur.

Fiziksel bağlantılı ağırlıklar, devre üretildikten sonra tekrar değiştirilemezler oysa programlanabilir olanlar değiştirilebilirler. Fiziksel bağlantılı ağırlıklar otomatik düzende ilginç bir sorun teşkil etmekte. Karşılaştırma fonksiyonu, COMP, gibi bazı fonksiyonlar 1'den $2^{n/2}$ 'ye kadar bir ağırlık gerektirmektedir. Şekil 6.3 8-bit COMP fonksiyonunu göstermektedir. VE, VEYA ve tüm simetrik fonksiyonlar küçük ağırlıklarla uygulanabilir. Bu farklılık da fiziksel bağlantılı ağırlıkları ve bazı LT geçişlerini kullanmanın diğerlerinden daha geniş olduğunu gösterir.



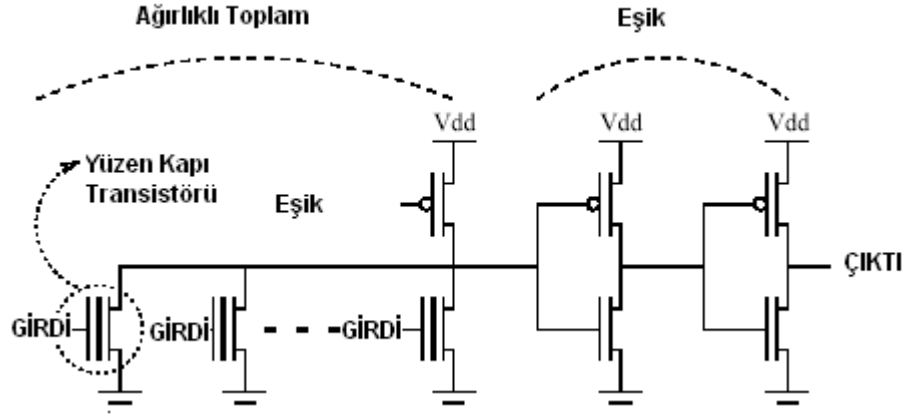
Şekil 6.2: 2 4-bit tamsayının karşılaştırılması.

Programlanabilir ağırlık kullanmak, düzeni basitleştirir ve LT elementinin hesapladığı fonksiyon üzerinde değişiklik yapmayı sağlar. Sıradaki bölümde uygulamanın ayrıntılarından bahsedeceğiz.

6.4 Uygulama ve Sonuçları

Sayfa [41]'de yazarlar, değişken bir doğru yanlış fonksiyonunu uygulayan sinir bazlı bir devre üretmişlerdi. Biz değişken bir eşik elementi (doğru yanlış fonksiyonlarının sınırlı bir kümesi) uyguluyoruz. Asıl fonksiyon ağırlıklar düzeltilerek seçildi. Şekil 6.4'te şemasal uygulama görülebilir. 16-girişli eşik elementi MO-SIS'te mevcut olan standart $2\mu m$ duble - çoklu analog işlemi kullanılarak üretildi. Düzen için Şekil 6.4'e bakınız. 16 girdi tüm 4 geçişe de metal kullanılarak (mor) dağıtıldı; böyle bir düzen eşik elementlerinin yoğun şemalarını oluşturmayı sağlar.

Her ağırlık için tek bir transistör kullanarak, ağırlıkları polisilikon yüzen geçişlerin üzerine yerleştiriyoruz böylece yenilenme olmadan da uzun zaman boyunca hafızada tutma sağlanıyor. Çarpma, girdilerin doğru yanlış olması gerçeği üzerine kurulmuştur, mantıksal bir 0 için 0 Volt, mantıksal bir 1 için X Volt, X 1 ile 5 Volt arasında değişebilir. Bir girdi ilgili ağırlığa akım oranı yaratır. Toplam $\sum_{i=1}^n w_i x_i$ transistörleri aynı yere bağladığımızda doğal olarak ortaya çıkar. Akımların toplamı yerine voltajların toplamaları kullanıldığında, [39]'daki bir diğer yaklaşım ortaya çıkar. Son olarak iki dönüştürücü çıktıları mantıksal 0'a ya da mantıksal 1'e çekmeye uğraşırlar.



Şekil 6.3: Programlanabilir Doğrusal Eşik Elemanın Şeması.

Yeni bir fonksiyonda programlama yapabilmek için, ağırlıkları yönlendirme ve sıcak elektron enjeksiyonu ile değiştirir, yüzen geçişlerin benzer uygulamaları için bkz sayfa [16], [19], [55]. Sayfa [10]'da görülebileceği üzere, burada kullanılan tek transistör belleğinden birazcık daha karmaşık olan analog hafıza hücresi, 14 bite kadar bilgi saklayabilir, ve bu çoğu uygulanabilir eşik fonksiyonu için yeterli bir miktardır.

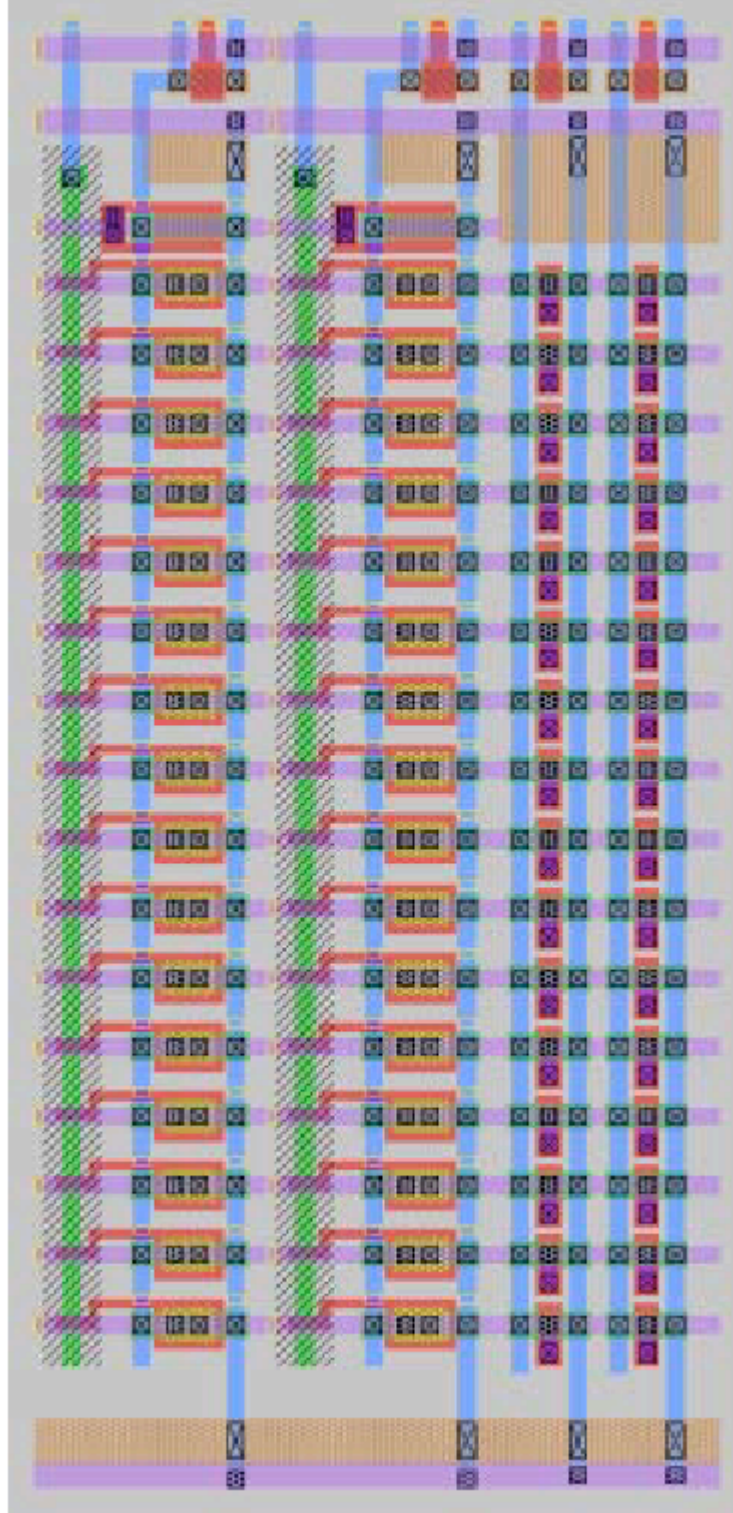
Girdi vektöründeki 1'lerin sayısını değiştirerek $w_0 + \sum_{i=0}^{16} x_i = 0$ iken eşik değerini w_0 belirleyerek, eşik elementimizin doğrusallığını test ettik. Mantıksal 1 değeri için 1 volt kullanıldı. Şekil 6.5 sonuçları göstermekte.

Verinin karekök şekline dikkat ediniz. Bu önemli bir noktayı göstermekte. T'de doğrusal olmayan belirli bir T değeri elde etmek için gerekli olan voltaj. Eşiğin üstünde ya da altında işleyen bir *nFET* için, tek bir girdinin katkıları sırasıyla şöyledir:

$$I = \frac{\beta}{2} (V_g - V_T)^2$$

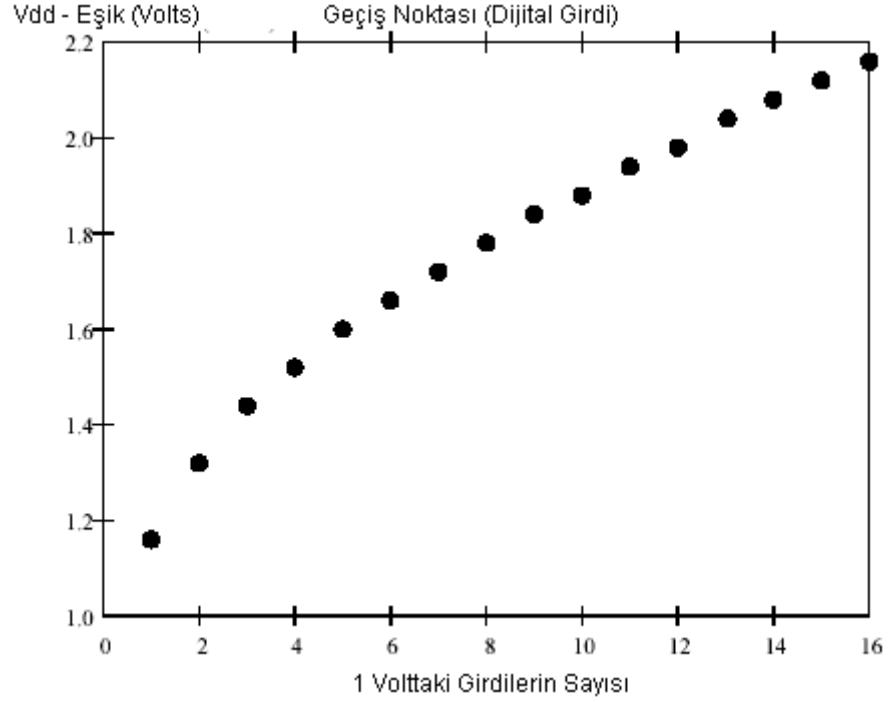
$$I = I_0 e^{\frac{kV_g}{V_T}}$$

V_T termal voltaj iken ve $(3, I_0$ ve κ sabit iken. Fiziksel bağlantılı ağırlıklar, $(5$ ve I_0 'nun oranlı olduğu transistörün W/L oranı olarak kodlanmışlardır[29]. Bu, transistörün neresinde işlem olduğuna bağlı olmadan, ağırlıkların değerlerini W/L 'de doğrusal kılmaktadır. Programlanabilir ağırlıklar durumunda değer,



Şekil 6.4: Doğrusal toplamın düzeni – $w_0 - w_0 \% \sum_{i=1}^{16} w_i x_i$. Dört eşik elementi

gösterilmektedir, iki programlanabilir ve iki programlanamaz, sonuncusunun birim ağırlıkları vardır. Gösterilen alan $168\mu \times 360\mu$ 'dur. Çip MOSIS'te mevcut olan 2μ teknolojisiyle üretilmiştir.



Şekil 6.5: Vdd — Eşik ve girdideki 1'lerin sayısı.

Yüzen geçişte depolanan voltajda ağırlıkların sayıları üstel ya da ikinci dereceden olabilir bkz: Şekil 6.4. Bu tarz doğrusal olmama durumları, geniş dinamik aralıklarda söz konusudur.

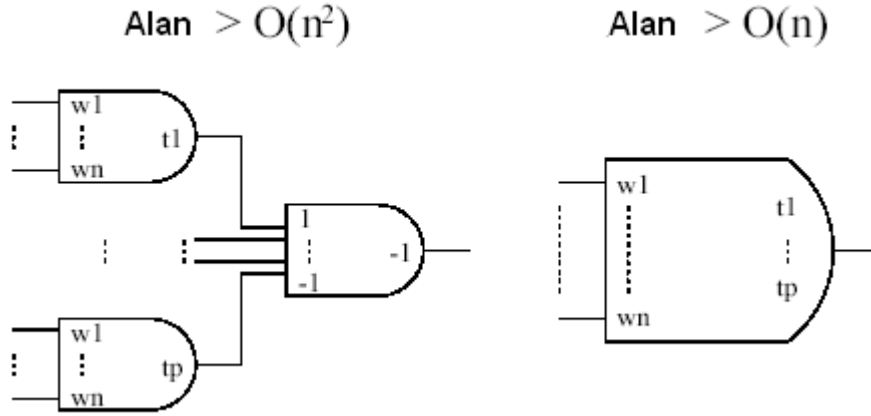
6.4 VLSI Düzeni ile LTM

LTM hakkındaki teorik sonuçlar doğru yanlış fonksiyonlarının VLSI uygulanmasına da uyarlanabilir. Çoklu eşikleri olan geçiş fikri aklımıza simetrik doğru yanlış fonksiyonlarının etkin bir VLSI uygulamasını ararken geldi. Tek bir *LT* geçişi simetrik doğru yanlış fonksiyonlarını uygulamada yetersiz olsa bile, iki katmanlı bir *LT* devresi yeterlidir. Ayrıca böyle bir devrenin VE, VEYA ve DEĞİL üzerine kurulmuş geleneksel mantık devresinden çok daha iyi çalıştığı bilinmektedir. Ayrıca geleneksel devrelerin üstel büyüklükleri vardır.

Öneri 6.1 (Simetrik Fonksiyon Uygulaması için LT_2 vs LTM)

Simetrik bir fonksiyonun LT_2 düzeni $O(n^2)$ kadar bir alan gerektirmektedir oysa

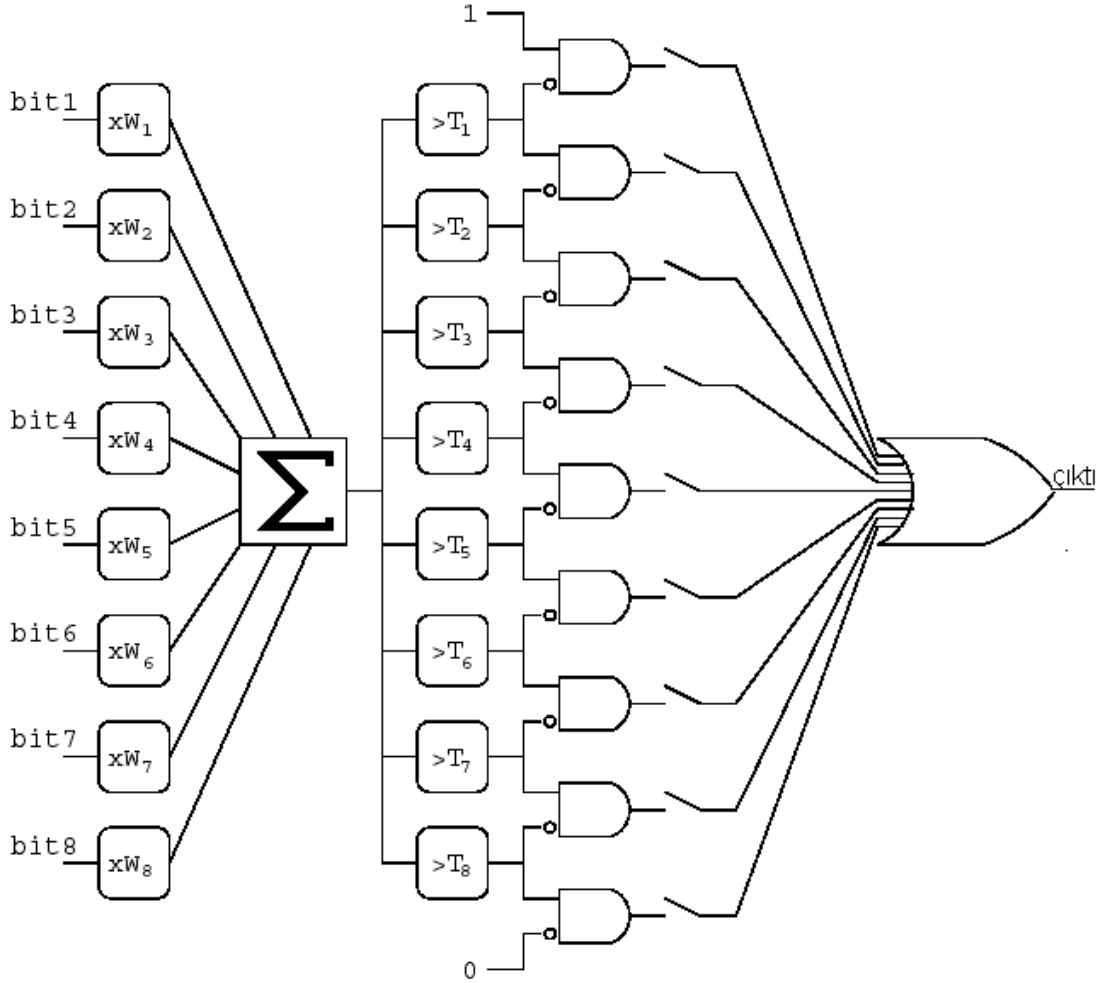
Şekil 6.6: Genellenmiş bir simetrik fonksiyonun uygulanmasında LTM'nin LT'ye göre olan avantajlarını göstermektedir. Ağırlıklı toplam her geçişte tekrar uygulanmaktansa sadece bir kere uygulanır.



KANIT:

LT_2 'de genellenmiş simetrik bir fonksiyonu uygulamak için birinci tabakada n tane LT geçişine ihtiyaç vardır. w_0 geçişi dışında hepsi aynı w_i ağırlığına sahiptir. Aynı doğrusal toplam olan $\sum_{i=1}^n w_i x_i$ 'i n kere tekrarlamaktansa, bunu bir kere yapıyoruz ve sonuçları n tane farklı geçişle karşılaştırıyoruz. Sonuç olarak ortaya çıkan devre tek bir LTM geçişine tekabül eder.

Yukarıdaki önerme Şekil 12'de tasvir edilmiştir. Aslında, LT_2 düzeni gereğinden fazladır, her ağırlığın n tane kopyası vardır ve en az $O(n^2)$ 'lik bir alana ihtiyaç vardır. Diğer taraftan LTM tek bir ağırlıklı toplam vermektedir ve alan ihtiyacı $O(n)$ 'dir. Şekil 6.5 LTM elementinin yüksek düzeyli bir şemasını göstermektedir. Bu element $2mm \times 2mm$ çip üzerine, MOSIS'deki 2μ teknolojisi kullanılarak üretilmiştir. Şekil 14 düzenini göstermektedir. 16 girdisi vardır. Çıktısı 4-bit bellek hücresini gösteren 4-bit bus'tan oluşur. LT geçişinin düzeninde kullanılan akımların toplamının, Şekil 6.4, aksine; ağırlıklı toplam Nöron MOS tarzında uygulanmıştır, voltajların kapasite toplamı için bkz: Sayfa [30], [39]. Ağırlıkların ve eşiklerin değerleri yüzen geçişler üzerine depolanmıştır. Belirli bir ağırlık/eşik seçmek için giriş çizgileri kullanılarak, birlikte ya da ayrı ayrı olmak üzere programlanabilirler.



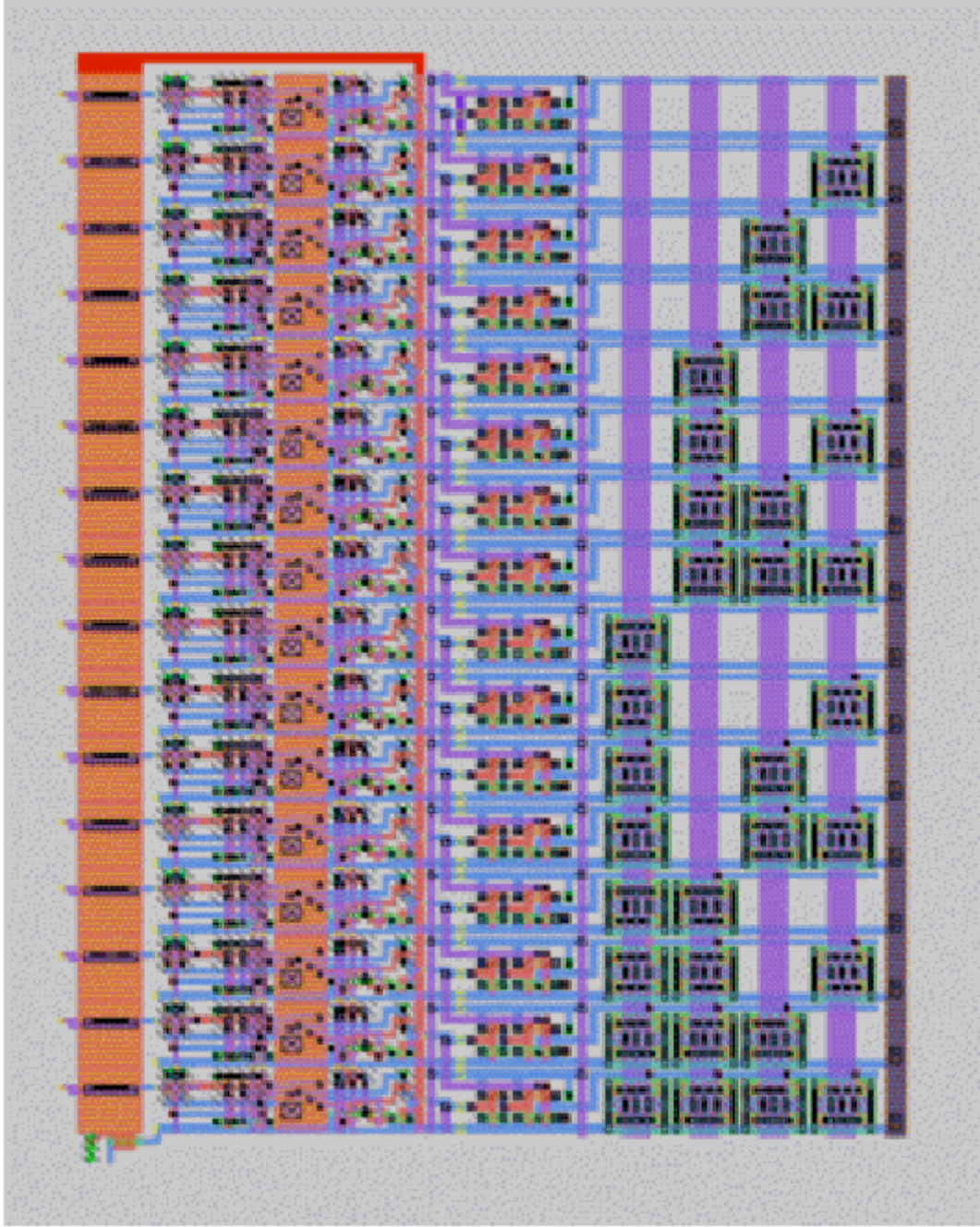
Şekil 6.7: Bir LTM geçişinin yüzey düzeyli şeması.

Eşiklerin yükseldiğini varsayarak, sadece tek bir çizginin VE tabakasının çıktısında mantıksal 1'de olduğu bilinmektedir, bkz: Şekil 13. bu bilgiden yararlanarak, 16 çizgiyi değerleri fonksiyonlara göre depolayan bir bellek hücresi yardımıyla 4-bir bus'lara yüklüyoruz. Genel olarak, $\log_2 1$ bitlerinden bir bus elde edilir, t LTM elementinin geçişlerinin (eşiklerinin) sayısı iken. Simetrik fonksiyonlarda $t = n$, girdi sayısı. Ya da Şekil 6.5'teki devreye bakılabilir, 16-bit girdi ve 4-bit programlanabilir hesaplama elementi.

6.5 Sonuç

Ağırlık depolamak için yüzen geçişler kullanarak 16-girdili programlanabilir doğrusal eşik elementi ürettik ve test ettik. Böyle bir depolama yenileme gerektirmemekte ve ağırlıkların enjeksiyon ya da yönlendirme yoluyla değiştirilmesi sağlanmaktadır. 16-girdili çoklu-eşik elementi uygulayarak ikinci bir çip daha ürettik. Bir tek çoklu-eşik elementi XOR ve tamsayı toplamı yapabilmektedir. Bazı yararlı doğru yanlış fonksiyonlarının ilk tabakadaki tüm geçişleri aynı ağırlığa sahip olan 1-tabakalı LT devreleri tarafından yapılabilmesi bilgisinden yararlanmaktadır. Ağırlıklı toplamı sadece bir kere uygulayarak, alanın n^2 'den n 'e düşürülmesini sağlamaktadır.

Uygulanabilirlik açısından bakarsak, bu çalışmanın devamı, fiziksel bağlantılı ağırlıklı eşik devrelerinin düzenini sistematik (ya da otomatik) bir şekilde üretmeyi planlama yönünde olabilir. Çalışmanın başka bir yöne ise programlanabilir eşik elementlerini FPGA'larda yapı taşı olarak kullanmak olabilir.



Şekil 6.8: 16-girdili bir LTM elementinin düzeni. Çıktı 4-bit bellek hücrelerini gösteren 4-bit bus'tan oluşmaktadır. Ağırlıklı toplam, voltajların kapasite toplamı olarak Nöron MOS tarzında uygulanmıştır. Bu çip MOSIS'de bulunan 2μ teknolojisi ile üretilmiştir.

Bölüm 7

Sonuçlar

Bu tez doğrusal eşik elementlerinin özelliklerini incelemiştir. Bunlar, doğru yanlış girdileri, doğru yanlış çıktıları olan yapay nöronlar, çıktılarının ağırlıklı toplamalarının işareti hesaplamak. 3 şekilde katkımız olmuştur:

- Teorik düzeyde $\hat{L}T^{(d)}$ ve LTM gibi yeni fonksiyon sınıflarını tanımladık ve bunların hesaplama gücünü sınıflandırdık.
 - Algoritmik düzeyde, gerçek ağırlıkların reel sayıların eğişken bir alt kümesinden alınan ağırlıklara nasıl dönüştürülebileceğini gösterdik. LT fonksiyonlarının minimum ağırlıkla nasıl oluşturulabileceğini ve son olarak COMP'u hesaplayan bir $\hat{L}T_2$ devresi üreten bir algoritma gösterdik. Ayrıca, XOR, ADD, PRODUCT gibi yararlı fonksiyonları hesaplayan LTM devrelerinden bahsettik.
 - Uygulama düzeyinde, LT ve LTM'nin VLSI uygulamasının tasarımı, düzenini ve test edilmesini gösterdik. Ağırlıkların değerini depolamak için yüzen geçiş teknolojisini kullanan programlanabilir bir LT elementi tasarladık.

Bölüm 2'de eşik devreleri teorisinin bazı bilinen sonuçlarını gösterdik, özellikle herhangi bir doğrusal eşik elementinin tamsayı ağırlıklarla kullanılabileceğine değindik. Buradaki katkımız, reel sayıların değışken kümesiyle yapılan bir genelleme oldu. Ağırlıkları değıştirmeye yarayan bir algoritma ile bir LT fonksiyonunun uygulanma şartlarını belirttik. Bölüm 3'te minimum ağırlıklarla doğrusal eşik fonksiyonları oluşturmak için bir metot sunduk. Bu d ile işaretlenmiş olan $\hat{L}T^{(d)}$ sınıfları arasındaki farklılıkları ortaya çıkarmak için kullanıldı. $d, O(n^d)$ ağırlıklarıyla uygulanabilen fonksiyonlar kümesi olan bir $\hat{L}T^{(d)}$ sınıfı. Bölüm 4'te bilinen bir sonuç ortaya konuldu, büyük ağırlıklara sahip bir LT elementinin, $\hat{L}T$ elementlerinden oluşan iki tabakalı devreyle, küçük ağırlıklı doğrusal eşik elementleriyle kullanılabileceği. Katkımız, COMP, karşılaştırma fonksiyonu için bu devrelerden oluşturmak oldu. Bölüm 5'te LTM'den ya da çok eşikli doğrusal eşik elementinden bahsedildi. XOR, ADD, PRODUCT gibi yararlı Boolean fonksiyonları için oluşturulmuştur. Ayrıca LTM'nin LT ve $\hat{L}T$, $\hat{L}T_2$ ve LT_2 gibi alt sınıflarına göre gücü hesaplandı. Son olarak Bölüm 6'da LT ve LTM'nin VLSI uygulamasından bahsedildi. Fiziksel bağlantılı ve programlanabilir çözümler sunuldu. Ağırlıklar, yüzen geçişlere yük olarak depolandı ve elektronların yönlendirilmesi ve enjeksiyonuyla değıştirildi.

Açık kalan noktalar ve gelecekteki araştırmalar için ilginç yönler şunlardır. Örneğin bu tezde belirtilen fonksiyon sınıflarıyla doğrusal karar listesindeki fonksiyon sınıfları arasındaki ilişki[49]. LTM ve LT ile ilgili konuları tamamlamak için varsayım 5.1'in kanıtı gereklidir. Algoritmik açıdan, ağırlıkları dönüştürmek ve küçültmek için verimli algoritmalar geliştirmek zor bir sorun gibi görünmektedir. Donanım uygulaması konusunda ise, uzak bir hedef de, bazı özel programlanabilir geçiş şemalarıyla, eşik elementlerinin mantık tasarım kütüphanelerinde yapı taşı olarak kullanılmasıdır.

Ek

```
fonksiyon correct = test(n, r, t)
%% fonksiyon correct = test(n, r, t)
% COMPARISON(X,Y)'nin simulasyonu
% correct = 1 eğer çalışma kurulursa
% n = X (Y) içindeki kullanılan bitlerin sayısı
% r = kullanılan asal sayıların sayısı
% t = kullanılan eşik
% V.Bohossian May, 96

BIG = 2^n;
correct = 1;
load primes.txt; % İlk 1000 asal sayı
p = primes(3:r + 2); % 2 ve 3'ü çıkar
hp = fix(p / 2) + 0.1; % hp : yarı p
p = p * ones(1, n); % kolonları çiftle
hp = hp * ones(1, n);
for i=1:n, L(i) = 2 ^ (i - 1);end;

for x=0 : 2 ^ n - 1,
for y=0 : 2 ^ n - 1,
    Ax = fix((x * ones(1, n)) ./ L);
    Ay = fix((y * ones(1, n)) ./ L);
    A = Ax - Ay;
    A = rem(ones(r, 1) * A + BIG * p, p);
    A = A + ((signChp - A) - 1) / 2 .* p;
    positives = size(find(~(A - 1)),1);
    bit = (positives > t);
    correct = correct & (bit == (x > y));
end;
end;

return;
```

Kaynaklar

- [1] E. Allender. A note on the power of threshold circuits. *Proceedings of the 30th IEEE Symposium on Foundations of Computer Science*, pages 580-584, 1989.
- [2] N. Alon and J. Bruck. Explicit constructions of depth-2 majority circuits for comparison and addition. *SIAM Journal of Discrete Mathematics*, 7(1): 1-8, February 1994.
- [3] E. Amaldi and V. Kann. The complexity and approximability of finding maximum feasible subsystems of linear relations. *Ecole Polytechnique Federale De Lausanne Technical Report*, ORWP 93/11, August 1993.
- [4] J.J. Amodei, R.O. Winder, D. Hampel, and T.R. Mayhew. Digital circuit techniques. *International Solid-State Circuits Conference*, February 1967.
- [5] P.W. Beame, S.A. Cook, and H.J. Hoover. Log depth circuits for division and related problems. *Proceedings of the 25th IEEE Symposium on Foundations of Computer Science*, pages 1-6, 1984.
- [6] V. Bohossian and J. Bruck. Algebraic techniques for constructing minimal weight threshold functions. *Submitted to SIAM Journal of Discrete Mathematics*. Available at <http://paradise.caltech.edu/ETR.html>.
- [7] V. Bohossian and J. Bruck. Multiple threshold neural logic. *Advances in Neural Information Processing Systems*, 10, 1998. Available at <http://paradise.caltech.edu/ETR.html>.
- [8] V. Bohossian, P. Hasler, and J. Bruck. Programmable neural logic. *Proceedings of the second annual IEEE International Conference on Innovative Systems in Silicon*, pages 13-21, 1997. Available at <http://paradise.caltech.edu/ETR.html>.
- [9] J. Bruck. Harmonic analysis of polynomial threshold functions. *SIAM Journal of Discrete Mathematics*, 3(2):168-177, May 1990.
- [10] C. Diorio, S. Mahajan, P. Hasler, B.A. Minch, and C. Mead. A high resolution non-volatile analog memory cell. *Proceedings of the International Conference of Circuits and Systems*, 3:2233-2236, 1995.
- [11] R. Douglas, M. Mahowald, and C. Mead. Neuromorphic analogue VLSI. *Annual Reviews in Neuroscience*, 18:255-281, 1995.
- [12] M. Goldmann, J. Hastad, and A. Razborov. Majority gates vs. general weighted threshold gates. *Computational Complexity*, 2:277-300, 1992.
- [13] M. Goldmann and M. Karpinski. Simulating threshold circuits by majority circuits. *Proceedings of the 25th ACM Symposium on the Theory of Computing*, pages 551-56, 1993.

- [14] A. Hajnal, W. Maass, P. Pudlak, M. Szegedy, and G. Turan. Threshold circuits of bounded depth. *Journal of Computer and System Sciences*, 46(2): 129-154, April 1993.
- [15] D.R. Haring. Multi-threshold threshold elements. *IEEE Transactions on Electronic Computers*, 15(1), February 1966.
- [16] P. Hasler, C. Diorio, B.A. Minch, and C. Mead. Single transistor learning synapses. *Advances in Neural Information Processing Systems*, pages 817-824, 1995.
- [17] J. Hastad. On the size of weights for threshold gates. *SIAM Journal of Discrete Mathematics*, 7:484-492, 1994.
- [18] T. Hofmeister. A note on the simulation of exponential threshold weights. *CON-COON conference*, 1996.
- [19] M. Holler, S. Tarn, H. Castro, and R. Benson. An electrically trainable artificial neural network with 10240 'floating gate' synapses. *International Joint Conference on Neural Networks*, 11:191-196, June 1989.
- [20] J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the USA National Academy of Sciences*, 79:2554-2558, 1982.
- [21] A.A. Irmatov. On the number of threshold functions. *Diskretnaya Matematika (Russian)*, 5(3):40-43, 1993.
- [22] A. A. Irmatov. Estimations of the number of threshold functions. *Discrete Mathematics and Applications*, 6(6):569-583, 1996.
- [23] J. Kahn, J. Komlos, and E. Szemerédi. On the probability that a random $\{\pm 1\}$ -matrix is singular. *Journal of the American Mathematical Society*, 8(1):223-240, 1995.
- [24] W.H. Kautz. The realization of symmetric switching functions with linear-input logical elements. *IRE Transactions on Electronic Computers*, March 1961.
- [25] M. Krause and P. Pudlak. On computing boolean functions by sparse real polynomials. *Proceedings of the 36th Annual Symposium on Foundations of Computer Science*, pages 682-691, October 1995.
- [26] H.W. Kuhn and A.W. Tucker. Linear inequalities and related systems. *Annals of Mathematics Studies*, 38, 1956. Princeton University Press, Princeton, NJ.
- [27] H.W. Kuhn and A.W. Tucker. On systems of linear inequalities. *Linear Inequalities and Related Systems, Annals of Mathematics Studies*, 38:99-156, 1957. Princeton University Press, Princeton, NJ.

- [28] R. Lauwereins and J. Bruck. Efficient implementation of a neural multiplier. *IBM Research Report*, RJ 8138 (74551), May 30, 1991.
- [29] C. Mead. Analog VLSI and neural systems. *Addis on-Wesley*, 1989.
- [30] B.A. Minch, C. Diorio, P. Hasler, and C. Mead. Translinear circuits using subthreshold floating-gate MOS-transistors. *Analog integrated circuits and signal processing*, 9(2): 167-179, March 1996.
- [31] R.C. Minnick. Linear - input logic. *IRE Transactions on Electronic Computers*, March 1961.
- [32] M. Muroga. Threshold logic and its applications. *Wiley-Inter science*, 1971.
- [33] J. Myhill and W.H. Kautz. On the size of weights required for linear-input switching functions. *IRE Transactions on Electronic Computers*, 10:288-290, 1961.
- [34] A.M. Odlyzko. On subspaces spanned by random selections of ± 1 vectors. *Journal of Combinatorial Theory, Series A*(47): 124-133, 1988.
- [35] S. Olafsson and Y.S. Abu-Mostafa. The capacity of multilevel threshold functions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(2), March 1988.
- [36] D.T. Perkins, D.G. Willis, and E.A. Whitmore. Division of space by concurrent hyperplanes. *Internal Report, Missile and Space Division*, 1959. Lockheed Aircraft Corporation, Sunnyvale, California.
- [37] D. Rumelhart and J. McClelland. Parallel distributed processing: Explorations in the microstructure of cognition. *MIT Press*, 1982.
- [38] J.S. Shawe-Taylor, M.H.G. Anthony, and W. Kern. Classes of feedforward neural networks and their circuit complexity. *Neural Networks*, 5:971-977, 1992.
- [39] T. Shibata, K. Kotania, and T. Ohmi. Real-time reconfigurable logic circuits using neuron MOS transistors. *International Solid-State Circuits Conference*, 1993.
- [40] T. Shibata and T. Ohmi. A functional MOS transistor featuring gate-level weighted sum and threshold operations. *IEEE Transactions on Electron Devices*, 39(6), June 1992.
- [41] T. Shibata and T. Ohmi. Neuron MOS binary-logic integrated circuits - part I: Design fundamentals and soft-hardware-logic circuit implementation. *IEEE Transactions on Electron Devices*, 40(3), March 1993.
- [42] L. Shlafli. Gesamelte mathematische abhandlugen. *Band 1*, 1850. Basel: Verlag Birkhauser.

- [43] K. Siu and J. Bruck. On the power of threshold circuits with small weights. *SIAM Journal of Discrete Mathematics*, 4(3):423-435, August 1991.
- [44] K. Siu, J. Bruck, T. Kailath, and T. Hofmeister. Depth efficient neural networks for division and related problems. *IEEE Transactions on Information Theory*, 39(3):423-435, May 1993.
- [45] K. Siu and V.P. Roychowdhury. On optimal depth threshold circuits for multiplication and related problems. *SIAM Journal of Discrete Mathematics*, 7(2):284-292, May 1994.
- [46] D.R. Smith. Bounds on the number of threshold functions. *IEEE Transactions on Electronic Computers*, June 1966.
- [47] M. Szegedy. Algebraic methods in lower bounds for computational models with limited communication. *PhD Thesis*, 1989. Chicago, Illinois.
- [48] T. Tich-Dao. Threshold l^2L and its applications to binary symmetric functions and multivalued logic. *IEEE Journal of Solid-State Circuits*, 12(5), October 1977.
- [49] G. Turan and F. Vatan. Linear decision lists and partitioning algorithms for the construction of neural networks. 1997.
- [50] Villasenor and W.H. Mangione-Smith. Configurable computing. *Scientific American*, pages 66-71, June 1997.
- [51] I. Wegener. The complexity of the parity function in unbounded fan-in unbounded depth circuits. *Theoretical Computer Science*, 85:155-170, 1991.
- [52] D.G. Willis. Minimum weights for threshold switches. *Switching Theory in Space Techniques*, 1963. Stanford University Press.
- [53] B.A. Wooley and C.R. Baugh. An integrated m-out-of-n detection circuit using threshold logic. *IEEE Journal of Solid-State Circuits*, 9(5), October 1974.
- [54] S. Yajima and T. Ibaraki. A lower bound on the number of threshold functions. *IEEE Transactions on Electronic Computers*, 14(6):926-929, December 1965.
- [55] K. Yang and A.G. Andreou. The multiple input floating gate MOS differential amplifier: An analog computational building block. *IEEE ISCAS*, 5, 1994.
- [56] A.C. Yao. On ACC and threshold circuits. *Proceedings of the 31th IEEE Symposium on Foundations of Computer Science*, pages 619-627, 1990.
- [57] T. Zaslavsky. Facing up to arrangements: Face-count formulas for partitions of space by hyperplanes. *Journal of the American Mathematical Society*, 154, Providence, RI, 1975.

- [58] Y.A. Zuev. Methods of geometry and probabilistic combinatorics in threshold logics. *Discrete Mathematics*, pages 427-438, Appl. 2, 1992.