



Araştırma Makalesi

Nesne Sınıflandırma İşlemi İçin Tensor İşleme Birimi ve CPU Performans Karşılaştırması

Ayşe Ayyüce Demirbaş *¹, Ahmet Çınar¹

¹Fırat Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği, Elazığ, Türkiye

Anahtar Kelimeler:

TPU
CPU
Derin Öğrenme
Tensorflow
MobileNet v1

ÖZ

Tensor İşleme Birimi (TPU), Google tarafından derin öğrenme görevlerini hızlandırmak için özel olarak geliştirilmiş bir yongadır. Yakın zamana kadar TPU'lar sadece Google Cloud ve Google Colab Platformları üzerinden kullanılabilmekteydi. 2019 yılının başlarında Google firması bu ürünü Coral adı altında donanımsal olarak üretmiştir. Bu sayede, dizüstü bilgisayar ve Raspberry Pi 3 gibi düşük donanım özelliklerine sahip cihazlarda derin öğrenme uygulamaları daha hızlı bir şekilde gerçekleştirilebilmektedir. Bu makalede MobileNet v1 modeli kullanılarak ilk olarak TPU bulduran Google Coral USB Accelerator ile daha sonra ise CPU kullanılarak 5 kategori ve 4326 çiçek görüntülerinden oluşan bir Kaggle veri seti eğitilmiş ve eğitilen bu veri setine dayanarak görüntülerin sınıflandırılması sağlanmıştır. Bu yolla cihazın performansı incelenmiştir. CPU için başarı oranı %84.29252, TPU için başarı oranı ise %99.609 olarak elde edilmiştir.

Comparing Tensor Processing Unit and CPU Performance for Image Classification

Keywords:

TPU
CPU
Deep Learning
Tensorflow
MobileNet v1

ABSTRACT

Tensor Processing Unit (TPU) is a processor to accelerate deep learning tasks developed by Google. Until recently, TPUs have been only used on Google Cloud and Google Colab platforms. In early 2019 Google produces this hardware as Coral. Thus, deep learning tasks can be performed more quickly on low-cost computers such as laptops and Raspberry Pi 3. In this paper, a Kaggle dataset that consists of 5 categories and 4326 flower images have been trained with the Google Coral USB Accelerator that contains TPU and a CPU using the MobileNet v1 model and the images are classified based on this dataset. In this way, we have investigated the performance of Google Coral USB Accelerator. The accuracy of the training with CPU is 84.29252%, for the TPU training, 99.609%.

*Sorumlu Yazar

*(a.ayyuced@gmail.com) ORCID ID 0000-0002-6731-9345
(acinar@firat.edu.tr) ORCID ID 0000-0001-5528-2226

e-ISSN: XXXXXXXXXXXX

Geliş Tarihi: 01/04/2020; Kabul Tarihi: 29/04/2020

Bilgisayar Bilimleri ve Teknolojileri Dergisi

1. GİRİŞ

Makine öğrenmesi ile ilgili ilk çalışmalar 1943 yılında Walter Pitts ve Warren McCulloch tarafından yapılmıştır. Bu çalışma ile bir sinir ağı matematiksel olarak modellenmiştir. (McCulloch ve Pitts, 1943). 1950 yılında ise Alan Mathison Turing, bir bilgisayarın zeki olup olmadığını belirlenmesi için Turing Testi'ni ortaya koymuştur. Bu fikir yapay zekânın ilk kilometre taşı olarak tarihe geçmiştir (Turing, 1950). Modern yapay sinir ağlarının temelini atılması, 1958 yılında Frank Rosenblatt tarafından perceptron algoritmasının yazılmasıyla mümkün olmuştur (Rosenblat, 1958). David E. Rumelhart, Geoffrey E. Hinton ve Ronald J. Williams 1986'da yayımladıkları bir bildiri ile birçok gizli katmana sahip olan bir sinir ağının daha basit bir yolla daha etkili bir şekilde nasıl eğitilebileceğini göstermişlerdir (Rumelhart ve ark., 1988).

TensorFlow; günümüzde çok yaygın olarak kullanılan, Google tarafından geliştirilen açık kaynak kodlu bir makine öğrenmesi sistemidir. (Abadi ve ark., 2016) Çalışma biçimi basitçe açıklanacak olursa, Tensor olarak adlandırılan skaler, vektör veya n-boyutlu diziyi girdi olarak alır ve bir çıktı üretir. "Tensorflow'da tüm işlemler Graph adı verilen art arda gerçekleşen bir hesaplama kümesinde gerçekleştirilir." (URL-1) TPU'lar (Tensor processing unit- Tensör işleme üniteleri) sadece Tensorflow ile kullanılabilirler için bu çalışmada Tensorflow kullanılmıştır.

Derin öğrenme, bir çeşit makine öğrenmesidir. Görüntü tanıma, görüntü işleme, doğal dil işleme, tıbbi görüntü analizi gibi birçok alanda yaygın olarak kullanılır. İnsan beynindeki nöronların çalışma yapısına benzeyen yapay sinir ağları ile modellenen algoritmalarla çalışır. Bu yapay sinir ağları aynı nöronlar gibi, dış dünyadan girdi alır, bu girdiyi işler ve bir çıktı üretirler.

Derin öğrenme için CPU (Central Processing Unit-Merkezi işlem ünitesi), GPU (Graphics Processing Unit- Grafik işlem ünitesi) ve TPU kullanılabilir. CPU bu iş için özelleştirilmiş bir donanım değildir. GPU ise CPU'ya kıyasla daha verimli çalışacaktır. Fakat Tensorflow ile yapılan çalışmalarda TPU kullanılması büyük bir verim sağlayacaktır. Bunun sebebi TPU'nun Tensorflow için özel olarak geliştirilmiş bir işlemci olmasıdır.

Google Colab platformu; CPU, GPU veya TPU kullanılarak derin öğrenme çalışmalarının yapılmasını mümkün kılmaktadır. Bunun yanında Google Cloud üzerinden de TPU kullanılabilir. Fakat bu kullanımlar sırasında veri seti ve yazılan kodlar sadece çalışma yapılan bilgisayarda kalmayacak, karşı taraftaki bilgisayarlarda da veri olarak depolanacaktır. Bu sebepten dolayı çalışmanın yapıldığı bilgisayarda TPU kullanılabilmesi sorunu çözecektir. Google Coral USB Accelerator bu amaç için geliştirilmiş bir modüldür. İçinde TPU bulunduran bu cihaz, USB portundan bilgisayara bağlanabilen bir yardımcı işlemcidir.

Bu yardımcı işlemci, kişisel bilgisayarlarda veya Raspberry Pi gibi tek kart bilgisayarlarda elbette ki Google'ın bilgisayarlarındaki kadar verim elde edemeyecektir. Fakat Tensorflow için CPU'dan daha hızlı çalışacağı gerçektir.

Bu çalışmanın amacı, kişisel bilgisayarlarda yürütülen derin öğrenme çalışmalarını süre açısından incelemektir. Detaylı bir literatür taraması sonucunda bu konuda yeterli sayıda çalışma olmadığı görülmüştür.

Bu çalışmanın yapısı aşağıdaki gibidir;

Giriş bölümünde bu alanda daha önce yapılan çalışmalar ele alınmıştır. Yöntem bölümünde TPU'nun nasıl çalıştığı, çalışmada kullanılan model ve veri seti incelenmiştir. Daha sonra veri seti, bu modelle eğitilmiş ve sınıflandırmalar yapılmıştır. Bulgular bölümünde bir önceki bölümde elde edilen deneysel sonuçlar kullanılarak eğitim süresine göre işlemcilerin performansları karşılaştırılmıştır. Sonuç bölümünde ise bu konuda daha önce yapılan çalışmalara değinilmiş ve karşılaştırmalı sonuçlar verilmiştir. Yapılan çalışmada elde edilen verilere göre TPU'nun avantajlarından söz edilmiştir.

2. YÖNTEM

Derin öğrenme çalışmalarında işlemci olarak genellikle Graphical Processing Unit (GPU) kullanılır. Bu durum verimi artıracaktır. Bu çalışmada Google Coral USB Accelerator ve Intel® Core™ i7-7500U CPU @ 2.70GHz × 4 CPU'nun verisetinin eğitilmesi sırasında gösterdikleri performans ele alınacaktır. Derin öğrenme ile çalışırken Tensorflow kütüphanesi kullanılacaksa işlemci olarak TPU kullanmanın avantajlı olacağı öngörülmektedir.

2.1. TPU Nedir?

Bilgisayarlarda genel amaçlı işlemler için kullanılan CPU'lar skaler verilerle çalışır. "CPU'nun en büyük yararı esnekliğidir. Von Neumann mimarisi ile milyonlarca farklı uygulama için her türlü yazılım yüklenebilir. Bir bilgisayarda kelime işleme, roket motorlarını kontrol etmek, banka işlemlerini gerçekleştirmek veya görüntüleri sinir ağları ile sınıflandırmak için CPU kullanılabilir." (URL-2) Fakat Von Neumann mimarisinin darboğazı CPU'nun bir dezavantajıdır.

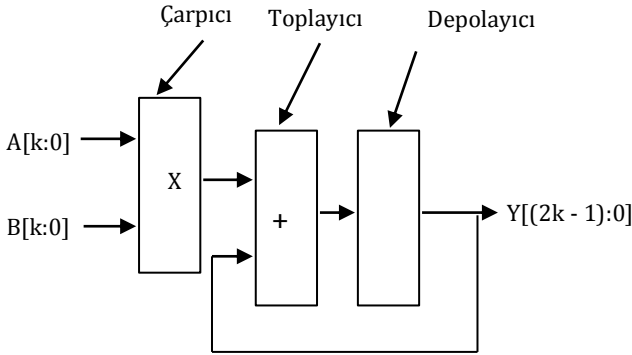
Bunun yanında GPU'lar, CPU'dan farklı olarak çok fazla sayıda Arithmetic Logic Unit (ALU) bulundurlar. Vektörler üzerinde işlem yaptıkları için grafiksel görevleri daha hızlı bir şekilde çalıştırabilirler. CPU'dan daha verimli çalıştıkları için derin öğrenme uygulamalarında sıkça tercih edilirler.

Yapay sinir ağları matrisler ile oluşturuldukları için bu iş için matrisler üzerinde işlem yapan TPU'ları kullanmak çok uygundur. TPU'lar Matris Çarpım Birimi ve Vektör İşleme Birimi

bulundurulur. Yapay sinir ağları için gereken matris işlemlerini Şekil 1’de gösterildiği gibi (çarpma ve toplama) hızlı bir şekilde, daha az güç tüketimi ile gerçekleştirirler.

Basit bir ifadeyle TPU, 8 bitlik donanım çarpanlarının 256x256 matrisini destekleyen donanımsal bir matris çapıcısıdır. (Olsen, 2017) Verinin donanım matrisine akmasını sağlayan sistolik kaydırma devresi sayesinde saniyede 65536 çarpma işlemi yapabilir. (Jouppi ve ark., 2017) TPU’lar binlerce toplama ve çarpma devresi içerirler. Bu devreler birbirine bağlanarak fiziksel bir matris oluşturulmuştur. Bu, sistolik dizi mimarisidir. (Jouppi ve ark., 2017) Bir sistolik dizi, yapay sinir ağı hesaplamalarını şu şekilde yürütür:

“İlk başta, TPU parametreleri bellekten çarpma ve toplama devrelerinden oluşan matrise yükler. Ardından, TPU verileri bellekten yükler. Her çarpma işleminde, toplama işlemi yapılırken sonuç bir sonraki çarpım devresine geçer. Böylece çıktı, veriler ve parametreler arasındaki tüm çarpma sonucunun toplamı olacaktır. Tüm büyük hesaplamalar ve verilerin aktarılması sürecinde, hiçbir hafıza erişimine ihtiyaç duyulmaz.” (URL-2)

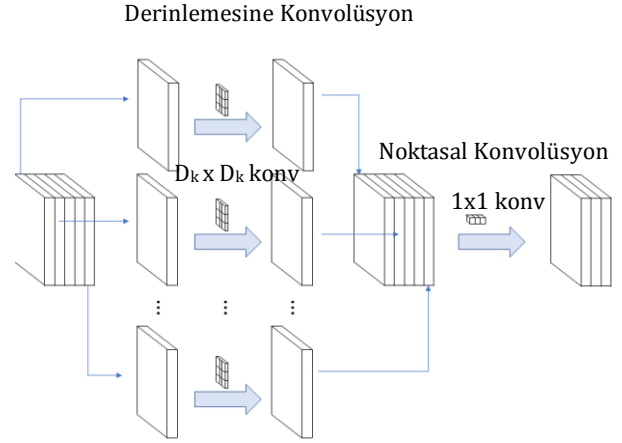


Şekil 1. Matris Çarpım Birimi (URL-3)

2.2. Model

ImageNet, 14 milyondan fazla görüntüden oluşan bir veri kümesidir. (Deng ve ark., 2009) Kullanılan model “ImageNet veri kümesinden 1.000 sınıfı anlamak için önceden eğitilmiş, değiştirilmiş bir MobileNet v1 modelidir.” (Retrain a classification model on-device with weight imprinting)

MobileNet v1 konvolüsyonel sinir ağları yapısını kullanır. (Sheng ve ark., 2018) “Konvolüsyon katmanının işi iki alt göreve ayrılır; girdiyi filtreleyen derinlemesine konvolüsyon katmanı ve yeni özellikler oluşturmak için bu filtrelenmiş değerleri birleştiren 1x1 konvolüsyon katmanı (noktasal konvolüsyon katmanı olarak da adlandırılır).” (URL-4) Bu iki konvolüsyon katmanı, derinlemesine ayrılabilir konvolüsyon bloğu oluştururlar. Şekil 2’de gösterilen derinlemesine ayrılabilir konvolüsyon, model boyutunu ve karmaşıklığını azaltmak için kullanılır.



Şekil 2. Derinlemesine Ayrılabilir Konvolüsyon Bloğu (URL-5)

MobileNet modeli aktivasyon fonksiyonu olarak ReLu6 kullanır. ReLu6, ReLu fonksiyonuna benzer. Ancak ReLu6 fonksiyonunda aktivasyon değeri 6’dan büyük olamamaktadır. Bu fonksiyon matematiksel olarak Denklem (1)’deki gibi tanımlanır,

Denklem (1): ReLu Fonksiyonu

$$y = \min(\max(0, x), 6)$$

2.3 Veri setinin Eğitilmesi

Farklı sayıda kategori ve görüntülerden oluşan veri setleri kolaylıkla bulunabilir. Çalışma için uygun veri seti seçildikten sonra bu veri seti bir model kullanılarak eğitilir. Bu noktada kullanılan işlemci hız açısından büyük öneme sahiptir.

2.3.1 Veri seti

Bu çalışmada veri seti olarak Kaggle’den alınan Papatya, Karahindiba, Gül, Ayçiçeği ve Lale olmak üzere 5 kategoride toplam 4326 çiçek görüntüsü kullanılmıştır. (Kaggle Flowers Recognition) Görüntüler yüksek çözünürlüklü olmayıp, yaklaşık 320x240 boyutlarındadır. Her bir kategoride ortalama 864 görüntü bulunmaktadır.

2.3.2 İşlemci

Veri seti eğitilirken önce MobileNet v1 modeli ile içinde TPU bulduran Google Coral USB Accelerator kullanılmıştır. Daha sonra ise Intel® Core™ i7-7500U CPU @ 2.70GHz × 4 CPU ile aynı veri seti aynı model ile tekrar eğitilmiştir.

Veri setinde bulunan görüntüler ve önceden eğitilmiş model kullanılarak Google tarafından yazılan classification_transfer_learning.py (URL-6) ile eğitilmiştir. Bu işlem 23140 milisaniye sürmüştür.

Daha sonra ise aynı veri seti, aynı model ile Intel® Core™ i7-7500U CPU @ 2.70GHz × 4 CPU kullanılarak eğitilmiştir. Bu işlem için Google tarafından yazılmış retrain.py kodları kullanılmıştır (URL-7). Bu işlem 373800 milisaniyede tamamlanmıştır.

2.3.3 Görüntülerin Sınıflandırılması

Eğitilen model kullanılarak Şekil 3'te gösterilen papatya, karahindiba, gül, ayçiçeği ve lale olmak üzere 5 kategoride veri setinde bulunmayan 30 tane çiçek görüntüsü Google Coral USB Accelerator ile sınıflandırılmıştır. Bu işlem sırasında Google tarafından yazılmış classify_image.py (URL-8) kodları kullanılmıştır. Bu 30 tane testin 23'ünde, eğitilen modelin doğru sonuç verdiği gözlemlenmiştir.



Şekil 3. Test aşamasında kullanılan görüntüler

Google Coral TPU USB Accelerator kullanılarak papatya.jpg dosyası için yapılan sınıflandırmada sonuç aşağıdaki gibi elde edilmiştir,

daisy: 0.99609

Aynı görüntü Intel® Core™ i7-7500U CPU @ 2.70GHz × 4 CPU ile sınıflandırıldığında aşağıdaki skor elde edilmiştir,

daisy
Score : 0.8429252

Her iki işlemci için de elde edilen skorlar birbirine yakın ve çoğu zaman doğru olacaktır. Ancak sonuçlar bölümünde de görüleceği üzere eğitim sırasında Google Coral TPU USB Accelerator daha büyük performans gösterecektir.

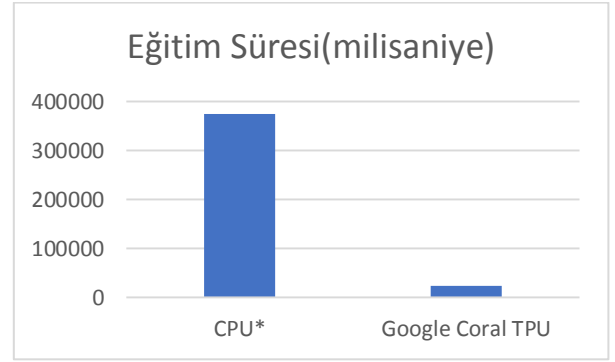
3. BULGULAR

Google Coral USB Accelerator ve Intel® Core™ i7-7500U CPU @ 2.70GHz × 4 CPU'nun MobileNet v1 modeli ile aynı veri seti için başarımları araştırılmıştır.

Aşağıda Tablo 1'de, MobileNet v1 modelinin eğitilmesi sırasında milisaniye cinsinden CPU ve Google Coral USB Accelerator TPU'nun elde ettiği başarımlar gösterilmiştir.

Tablo 1. Verisetinin CPU ve TPU'daki Eğitim Süresi

Model	Intel® Core™ i7-7500U CPU @ 2.70GHz × 4 CPU	Google Coral USB Accelerator TPU
MobileNet v1	373800	23140



Şekil 4. Veri seti eğitim sürelerinin grafiği

* Intel® Core™ i7-7500U CPU @ 2.70GHz × 4 CPU

Şekil 4'te Google Coral USB Accelerator 23,14 saniyede veri setinin eğitilmesi işlemi tamamlarken, CPU aynı işlemi 6.23 dakikada tamamlamıştır. Yani Google Coral USB Accelerator'ın Intel® Core™ i7-7500U CPU @ 2.70GHz × 4 CPU'dan daha hızlı bir şekilde çalıştığı deneysel olarak ispatlanmıştır.

Bu değerlerden de görüldüğü üzere Google Coral USB Accelerator, derin öğrenme işlemlerinde verimi artırmaktadır.

4. SONUÇ

Bu çalışmaya benzer bir çalışma da Google tarafından yapılmıştır. Fakat bu çalışmadan farklı olarak görüntülerin tanınması sırasında ne kadar zaman harcadığı araştırılmıştır.

Tablo 2. Google Tarafından 64-bit Intel(R) Xeon(R) E5-1650 v4 @ 3.60GHz CPU ve Google Coral Accelerator Kullanılarak Elde Edilen Milisaniye Cinsinden Süreler (URL9)

Model Mimarisi	Masaüstü CPU (1)	Masaüstü CPU (1) + USB Accelerator (USB 3.0) Edge TPU ile	Gömülü CPU (2)
DeepLab V3 (513x513)	394	52	1139
DenseNet (*) (224x224)	380	20	1032
Inception v1 (224x224)	90	3.4	392
Inception v4 (299x299)	700	85	3157
Inception-ResNet V2 (299x299)	753	57	2852
MobileNet v1 (224x224)	53	2.4	164
MobileNet v2 (224x224)	51	2.6	122

MobileNet v1 SSD (224x224)	109	6.5	353
MobileNet v2 SSD (224x224)	106	7.2	282
ResNet-50 V1 (*) (299x299)	484	49	1763
ResNet-50 V2 (*) (299x299)	557	50	1875
ResNet-152 V2 (299x299)	1823	128	5499
SqueezeNet (224x224)	55	2.1	232
VGG16 (224x224)	867	296	4595
VGG19 (224x224)	1060	308	5538
EfficientNet-EdgeTpu-S(**)	5431	5.1	705
EfficientNet-EdgeTpu-M(**)	8469	8.7	1081
EfficientNet-EdgeTpu-L(**)	22258	25.3	2717

*“1 Desktop CPU: 64-bit Intel(R) Xeon(R) E5-1650 v4 @ 3.60GHz
2 Embedded CPU: Quad-core Cortex-A53 @ 1.5GHz
3 Performansa engel olması nedeniyle bazı işlemlerin CPU üzerinde yürütülmesi gerekiyor
* Edge TPU çalışma zamanının 10. sürümü tarafından desteklenmiyor.”
** CPU'daki gecikme, bu modeller için yüksektir, çünkü TensorFlow Lite çalışma zamanı, tüm platformlardaki eğitim sonrası niceleme modelleri için tam olarak optimize edilmemiştir. “*

Tablo 2’de çeşitli modeller Google tarafından eğitilmiştir ve daha sonra CPU, Google Coral USB Accelerator ve Gömülü CPU üzerinde görüntü sınıflandırma işlemi yapılmıştır. Burada, kullanılan işlemcinin ve modelin görüntü tanıma çalışmalarında zaman açısından önemi incelenmiştir. Görüleceği üzere Google Coral USB Accelerator yalnızca eğitim sürecinde değil, görüntülerin sınıflandırılması sırasında da önemli başarımlar sağlamaktadır. Burada bazı modeller için CPU ve Google Coral USB Accelerator ile yapılan görüntü tanıma süreleri arasında çok büyük fark vardır. Bu durum modelin yapısından kaynaklanmaktadır.

Yeni bir sınıflandırma yaklaşımı sunan bir çalışmada DenseNet, Dense1_MobileNet, Dense2_MobileNet ve MobileNet modelleri kullanılarak Caltech-101 ve Uebingen Animals veri setlerindeki görüntüler sınıflandırılmıştır. (Wang ve ark. ,2020) MobileNet için sınıflandırma doğrulukları aşağıdaki gibi elde edilmiştir

Tablo 3. Caltech-101 ve Uebingen Animals veri setleri için MobileNet modeli ile elde edilen doğruluk değerleri

Veri seti	Doğruluk
Caltech-101	0.766
Uebingen Animals	0.916

Bu çalışmada bir papatya görüntüsü kullanılarak TPU için 0.99609, CPU için ise 0.8429252 doğruluk değerleri elde edilmiştir. Tablo 3’de görüldüğü üzere Caltech-101 ve Uebingen Animals veri setleri için elde edilen doğruluk değerleri bu çalışmada TPU üzerinde elde edilen doğruluk değerinden düşüktür. CPU için doğruluk değeri değerlendirilecek olursa, Caltech-101 veri seti ile elde edilen değerden yüksek ancak Uebingen Animals veri seti ile elde edilen değerden düşüktür. Bu durum kullanılan veri setinden kaynaklanmaktadır.

Derin öğrenme çalışmalarında genellikle CPU yerine GPU tercih edilir. Fakat çalışmada Tensorflow kütüphanesi kullanılacaksa TPU’nun tercih edilmesi daha uygundur. TPU, Google Coral USB Accelerator cihazı yerine Google Cloud platformu üzerinden veya Google Colab ile kullanılabilir. Buradaki sakıncalardan bir tanesi yapılan çalışmada yazılan kodlar, kullanılan veri seti gibi verilerin karşı taraftaki bilgisayar üzerinde de depolanacak olmasıdır. Bu yolla istenmeden karşı tarafla veriler paylaşılmaktadır. Bunun önüne geçmek için Google Coral USB Accelerator yardımcı işlemcisi kullanılabilir. Böylece düşük donanımsal özelliklere sahip cihazlarda da derin öğrenme çalışmaları daha verimli hale getirilebilir.

KAYNAKÇA

- A. M. Turing (1950). Computing Machinery and Intelligence. Mind 49: 433-460
- D. E. Rumelhart, G. E. Hinton, R. J. Williams (1988): Learning Representations by Back-Propagating Errors, Cognitive modeling
- Eric B. Olsen (2017) Proposal for a High Precision Tensor Processing Unit
- Frank Rosenblatt (1958), The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain, Psychological Review Vol. 65, No. 6
- Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li and Li Fei-Fei (2009) Dept. of Computer Science, Princeton University, USA, ImageNet: A Large-Scale Hierarchical Image Database
- Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek G. Murray, Benoit Steiner, Paul Tucker, Vijay

Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng, Google Brain, (2016) TensorFlow: A system for large-scale machine learning 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI '16) November 2–4, • Savannah, GA, USA

Norman P. Jouppi, Cliff Young, Nishant Patil, David Patterson ve diğerleri (2017) In-Datacenter Performance Analysis of a Tensor Processing Unit Google, Inc. , Mountain View, CA USA

Tao Sheng, Chen Feng, Shaojie Zhuo, Xiaopeng Zhang, Liang Shen, Mickey Aleksic (2018), A Quantization-Friendly Separable Convolution for MobileNets

URL-1:
<https://www.guru99.com/what-is-tensorflow.html> [Erişim Tarihi: 10.03.2020]

URL-2:
<https://cloud.google.com/blog/products/ai-machine-learning/what-makes-tpus-fine-tuned-for-deep-learning> [Erişim Tarihi: 10.02.2020]

URL-3:
<https://medium.com/@antonpaquin/whats-inside-a-tpu-c013eb51973e> [Erişim Tarihi: 10.03.2020]

URL-4:
<https://machinethink.net/blog/mobilenet-v2/> [Erişim Tarihi: 5.02.2020]

URL-5:
<https://towardsdatascience.com/review-mobilenetv1-depthwise-separable-convolutionlight-weight-model-a382df364b69> [Erişim Tarihi: 10.03.2020]

URL-6:
<https://gist.github.com/ayyucedemirbas/6c2d6bd9324834432df02e8083be9031> [Erişim Tarihi: 23.03.2020]

URL-7:
<https://gist.github.com/ayyucedemirbas/37ce6f12deb9db99715ac398a309285c> [Erişim Tarihi: 5.03.2020]

URL-8:
<https://gist.github.com/ayyucedemirbas/2901b48a1b33eec1fd4794a522c7e204> [Erişim Tarihi: 23.03.2020]

URL-9:
<https://coral.withgoogle.com/docs/edgetpu/benchmarks/> [Erişim Tarihi: 15.03.2020]

Warren McCulloch, Walter Pitts (1943): A Logical Calculus of the Ideas Immanent in Nervous Activity. Bulletin of Mathematical Biophysics, Vol. 5, p. 115, 133

Wei Wang, Yutao Li, Ting Zou, Xin Wang, Jieyu You, Yanhong Luo (2020) A Novel Image Classification Approach via Dense-MobileNet Models