



Az Örnekle Öğrenme Problemleri için MAML ve ProtoNet Algoritmalarının İncelenmesi

Ayla Gülcü¹, Muhammet Alkan²

¹ Fatih Sultan Mehmet Vakıf Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, İstanbul, Türkiye, (ORCID: 0000-0003-3258-8681), agulcu@fsm.edu.tr

² Fatih Sultan Mehmet Vakıf Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, İstanbul, Türkiye (ORCID: 0000-0001-5188-2742), malkan@fsm.edu.tr

(İlk Geliş Tarihi 2 Ekim 2020 ve Kabul Tarihi 10 Ocak 2021)

(DOI: 10.31590/ejosat.834647)

ATIF/REFERENCE: Gülcü, A. & Alkan, M. (2021). Az Örnekle Öğrenme Problemleri için MAML ve ProtoNet Algoritmalarının İncelenmesi. *Avrupa Bilim ve Teknoloji Dergisi*, (21), 113-121.

Öz

Derin sinir ağları ile özellikle görüntü veri kümeleri üzerinde çok başarılı sonuçlar elde edilmektedir. Ancak bu başarının arkasında büyük ölçekli etiketli veri kümeleri yatmaktadır. Derin öğrenme ağlarının özelleşmiş bir türü olan evrişimli sinir ağları birçok görüntü tanıma görevlerinde çok iyi sonuçlar vermiş ve görüntüleri modellemek için standart araçlar haline gelmişlerdir. Özellikle ImageNet gibi büyük ölçekli veri kümelerinin kullanılmaya başlanmasıyla geliştirilen ağların başarıları daha da artmıştır. Ancak ImageNet gibi büyük bir veri kümesi oluşturabilmek maliyetli olması nedeniyle pek kolay değildir. Eldeki problem ile ilgili çok sayıda örnek toplanamadığı durumlarda kullanılan sinir ağı yeterince iyi eğitilememekte, dolayısıyla istenen başarı elde edilememektedir. Bu nedenle derin sinir ağlarının genelleme yeteneklerinin geliştirilmesi ve büyük etiketli veri kümelerine olan ihtiyacın azaltılması çok önemlidir. Tasarlanan bir makina öğrenmesi modelinin büyük bir veri kümelerine ihtiyaç duyulmadan, az sayıda örnek ve önceki görevlerden öğrenilmiş olan meta-bilgilerle genelleştirme kabiliyetinin artırılması ve aynı zamanda yeni görevler için hızlı adaptasyon sağlanması meta-öğrenme ile gerçekleşebilir. Meta-öğrenme algoritmalarında hızlı öğrenme iki aşamada gerçekleştirilir; ilk aşamada meta-bilgiler çıkarılır, ikinci aşamada ise bu meta-bilgiler kullanılarak yeni görevlere hızlı adaptasyon sağlanır. Derin sinir ağları temelli meta-öğrenme yaklaşımlarının az örnekli görüntü veri kümelerinde başarılı sonuçlar vermesiyle meta-öğrenme kavramı tekrar önem kazanmış; özellikle az örnek öğrenme problemleri için literatürde çok sayıda yeni algoritma ortaya çıkmaya başlamıştır. Bu çalışmada, literatürdeki başarılı meta-öğrenme algoritmalarından Model-Agnostic Meta-Learning (MAML) ve Prototypical Networks (ProtoNet) algoritmaları az örnek öğrenme problemlerine uygulanarak performansları incelenmiştir. Veri kümesi olarak literatürde az örnek öğrenme problemleri için oluşturulmuş MiniImageNet ve CIFAR100 görüntü veri kümeleri kullanılmış; farklı meta-öğrenme parametreleri ve farklı algoritma hiper-parametreleri için algoritmaların verdikleri sonuçlar değerlendirilmiştir. Buna göre, MAML algoritmasının örnek sayısının daha az durumda ProtoNet' ten daha iyi sonuçlar verdiği; ancak ProtoNet' in fazla örnek durumunda daha iyi genelleme yapabildiği belirlenmiştir. Bu durum, MAML algoritmasının bütün sınıfları ayırt edebilecek nitelikte genel başlangıç ağırlıklarını öğrenmeye çalışması; ProtoNet algoritmasının ise her bir sınıfı daha iyi temsil edebilmek için her sınıfa ait farklı bir prototip bulmaya çalışması ile açıklanabilir.

Anahtar Kelimeler: az örnek sınıflandırma, meta-öğrenme, MAML, ProtoNet.

Investigating MAML and ProtoNet Algorithms for Few-shot Learning Problems

Abstract

Deep neural networks have proven to be very effective for image-related problems. However, their success is mainly attributed to the large-scale annotated datasets that have been used to train them. Convolutional neural networks which are special type of neural networks have achieved very good results for visual recognition problems and therefore have become the standard tool for these tasks. The use of large-scale annotated datasets like the ImageNet have even improved the results obtained by these networks. However,

¹ Sorumlu Yazar: agulcu@fsm.edu.tr

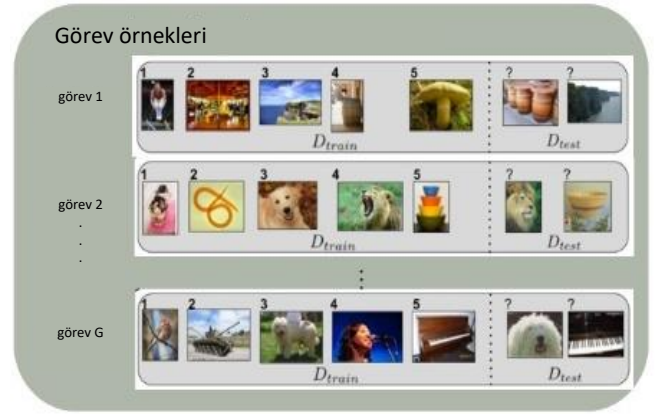
creating an annotated dataset of that scale is very difficult due to its cost. In some cases, even if there are enough resources, it is sometimes impossible to obtain such large datasets. It is shown that the neural networks cannot be trained well if there is not enough training data. As these networks require large amounts of annotated data to be able to generalize well, it is very important to develop new models that can be trained well even if the training data is not abundant. Meta-learning paradigm addresses this problem of few-shot learning by proposing models that can utilize the experience from the previous tasks to learn new tasks. Meta-learning algorithms gain the fast adaptation ability by using the meta-data obtained from the previous tasks. The meta-learning concept has regained its popularity after the success of some deep neural networks-based meta-learning algorithms for the few-shot image classification problems. In this study, two meta-learning algorithms, namely, Model-Agnostic Meta-Learning (MAML) and Prototypical Networks (ProtoNet) are applied to few-shot learning problems and their performance is evaluated. MiniImageNet and CIFAR100 few-shot learning image classification datasets have been used as the test bed, and the two algorithms have been evaluated under different meta-learning and algorithm hyper-parameter settings. The results suggest that MAML results in better classification accuracy than ProtoNet when the number of shot is taken as 1, and ProtoNet results in better accuracy when the number of shot is increased. The main reason for this is that while MAML tries to find common initial weights that can easily distinguish all classes, ProtoNet tries to find a different prototype for each class, and the number of shots definitely increases the representation power of that prototype.

Keywords: few-shot classification, meta-learning, MAML, ProtoNet.

1. Giriş

Yeterli miktarda etiketlenmiş veri bulunduğu derin sinir ağları modellerinin çok iyi performans verdiği gösterilmiştir (He vd., 2016). Ancak Finn vd. yaptıkları çalışmada etiketli örnek sayısının azaltıldığında, sadece birkaç adet örnek kullanıldığında derin ağlarının yeni sınıfları yeterince öğrenemediklerini ortaya koymuşlardır (Finn vd., 2017). Yine benzer bir çalışmada da evrişimli sinir ağlarının her bir sınıfa ait çok az sayıda örneğin verildiği görüntü tanıma problemlerinde yeni sınıflar için adapte olamadıkları, dolayısıyla kötü sonuçlar verdikleri gösterilmiştir (Ravi & Larochelle, 2017). Literatürde *az örnekle sınıflandırma* (*few-shot classification*) olarak adlandırılan bu problemler için klasik yöntemlerden farklı olarak öncelikle orta büyüklükte değişik veri kümeleri ile eğitilmiş bir modelin daha sonra az sayıda örnek içeren yeni sınıflar için hızlıca adapte olacağı meta-öğrenme olarak adlandırılan yöntemler kullanılmaya başlanmıştır. Az örnekle sınıflandırma problemlerinde örnek sayısı olarak genellikle 1 ve 5 alınmaktadır (1-shot, 5-shot).

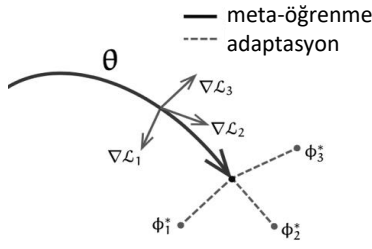
Meta-öğrenme literatürde daha çok öğrenmeyi öğrenme şeklinde anılmaktadır. Çünkü bu yaklaşım, önceki görevlerden/eğitimlerden elde edilen meta-bilgilerin kullanılmasıyla yeni görevlerin çok daha hızlı bir şekilde öğrenilmesini hedeflemektedirler. Bu sayede eğitim süreci yeni bir görev kümesi için ya da yeni bir problem için sıfırdan başlamak yerine belirli bir tecrübe üzerine geliştirilen bir süreç haline gelir. İnsanların öğrenme süreci bu konuda en güzel örnek oluşturmaktadır. Bir çocuk yeni bir nesneyi öğrenirken o nesnenin parçaları ile daha önceden öğrendiği nesnelere arasında bir bağlantı kurarak çok hızlı bir şekilde öğrenme gerçekleştirilebilir. Örnek vermek gerekirse, bisikleti ilk defa gören bir çocuk tekerleklerini daha önceden öğrendiği parçalara benzetecek, ya da direksiyonunu daha önceden oynadığı bir oyuncağın direksiyonuna benzetecek, bu sayede bisikleti diğer nesnelere ayıran özellikleri hemen fark edecektir. Sadece tek bir bisiklet örneği gösterilen bir çocuk bundan sonra bütün bisikletleri tanıyabilir duruma gelmektedir. İşte meta-öğrenme yaklaşımlarının temel amacı da her defasında çok sayıda örneğe ihtiyaç duymadan, sadece birkaç adet örnekle, önceki görevlerden edinilmiş bilgileri kullanarak yeni görevleri çok hızlı bir şekilde öğrenebilen modellerin oluşturulmasıdır.



Şekil 1: Eğitim veri kümesindeki farklı görevler

Meta-öğrenme tekniklerinin eğitim ve test süreçleri klasik makine öğrenmesi süreçlerinden farklıdır. Bu süreçte kullanılan terminoloji de diğer yaklaşımlardan farklıdır. Eğitim kümesi *görev* (*task*) adı verilen N adet sınıfın gösterildiği (N-yollu) ve her sınıfa ait K adet örnek verildiği bir veri kümesinden oluşur. Eğitim sürecinde öncelikle bu görevler üzerindeki hata değerlerine göre modelin parametreleri iyileştirilmeye çalışılır ancak asıl parametre güncellemesi test görevleri üzerinde elde edilen hataya göre yapılır. Dolayısıyla, her bir göreve ait test hatası, meta-öğrenme aşamasındaki eğitim hatası olarak alınarak model güncellenir.

Şekil 1’de her biri eşit büyüklükte destek ve sorgu veri kümelerinden oluşan G adet görevden oluşan bir meta-eğitim veri kümesi gösterilmektedir. Buradan da görülebileceği gibi bir meta-öğrenme eğitim süreci, önceden belirlenmiş G adet görev kullanılarak gerçekleştirilir. Öğrenmeyi öğrenme tamamen burada yer alan veriler kullanılarak gerçekleşir. Şekilde de görüldüğü gibi her bir görev, *destek* (*support*) ve *sorgu* (*query*) olarak iki ayrı veri kümesinden oluşmaktadır. Şekilde D_{train} destek, D_{test} ise sorgu kümesini göstermektedir. Her bir görev için aslında eğitim ve test verilerini oluşturan bu küçük veri kümeleri ana eğitim veri kümesinden rastgele olarak seçilir. Bir görev içindeki destek veri grubu, N sınıfa ait (N-yollu sınıflandırma için) K adet örnekten oluşmaktadır. Sorgu veri grubunda ise destek grubundaki veriler için eğitilen modelin gerçek performansının test edileceği veri kümesi bulunmaktadır. Sorgu veri grubunda ortak yaklaşım olarak 15 adet örnek kullanılmaktadır. Aynı eğitim kümesi gibi görevlerden oluşan test veri kümesi ve ayrıca istenirse doğrulama veri kümesi de oluşturulmakta, bu test veri kümesi ile modelin rapor edilen gerçek test performansları elde edilmektedir.



Şekil 2: Meta-öğrenme ve adaptasyon süreçlerinin gösterimi (Kaynak: Finn vd., 2017).

Meta-öğrenme aşamasından sonra adaptasyon aşamasına geçilir. Şekil 2’ de gösterilen adaptasyon sürecinde her bir görevdeki eğitim kümesindeki (destek) sınıflar için öğrenilmiş ağırlıklar ile başlanarak test kümesindeki (sorgu) sınıflar öğrenilmeye çalışılır. Bu adaptasyon aşaması sayesinde yeni sınıfların öğrenilmesi çok hızlı olur. Ancak test kümesindeki örnekler ile eğitim kümesindeki verilerin benzerliği ne kadar fazla olursa adaptasyon süreci o kadar verimli olur. Şekil 2’ de, Θ meta-modelin parametrelerini gösterirken kalın çizgi ise eğitim verileri ile gerçekleşen meta-öğrenme sürecinin ardından Θ ’ nın güncellenmesini göstermektedir. Her bir farklı görev için hatanın gradyan değerleri ($\nabla L_1, \nabla L_2, \nabla L_3$) ile Θ güncellenmiş olur. Φ_i ’ nin yeni bir i görevi için optimal parametreyi temsil ettiği düşünülürse, yeni meta-model parametresinin, Θ , tüm görevlerin optimal parametrelerine yakın olduğu (Φ_1, Φ_2, Φ_3) görülebilir. Tek bir görev değil, birden fazla farklı görev üzerinden güncellenen meta-model genelleştirme özelliğini gösteren başlangıç ağırlıkları ile yeni görevlere hızlı bir şekilde adapte olmayı başarır. Meta-öğrenme sürecinde Θ ’ nın optimize edilmesi amaçlanır. Adaptasyonda ise amaç Φ ’ nin optimize edilmesidir. Bu açıdan meta-öğrenme yaklaşımlarının iç ve dış olmak üzere iki döngü ile öğrenme gerçekleştirdikleri söylenebilir. Dış döngü meta-öğrenmenin gerçekleştiği, Θ ’ nın optimize edilerek genel özelliklerin çıkarıldığı; iç döngü ise adaptasyonun gerçekleştiği, Φ ’ nin optimize edilmeye çalışıldığı döngüdür. Dış döngüde çıkarılan genel özellikler sayesinde içeride yeni görevler için hızlı adaptasyon sağlanır.

Bu çalışmada, en temel meta-öğrenme alt yapısı haline gelen Model-Agnostic Meta-Learning (MAML) (Finn vd., 2017) ve Prototypical Networks (ProtoNet) (Snell vd., 2017) algoritmaları incelenmiştir. Algoritmaların performansı literatürdeki az örnekle öğrenme problemleri için oluşturulmuş veri kümelerinden MiniImageNet ve CIFAR100 kullanılarak farklı hiper-parametre kümeleri için değerlendirilmiş, bu sayede performansa etkisi en fazla olan hiper-parametreler belirlenmeye çalışılmıştır.

2. Materyal ve Metod

Bu bölümde ilk olarak seçilen iki meta öğrenme algoritması olan MAML ve ProtoNet algoritması anlatılmaktadır. Ardından, çalışmada kullanılan az örnekle öğrenme veri kümelerinden bahsedilmektedir.

2.1. MAML Algoritması

Model-Agnostic Meta-Learning (MAML) (Finn vd., 2017), en temel meta-öğrenme algoritması haline gelmiştir ve yeni ortaya atılan tüm yöntemlerde bu yöntem temel kıyaslama algoritması olarak kullanılmaktadır. Gradyanlı iniş ile eğitilebilen bir model üzerinde çalışabilen bu yaklaşım, tam bağlantılı model, evrişimli ya da yinelemeli sinir ağlarında kullanılabilir. Ayrıca algoritmanın en büyük avantajı, bir güncelleme fonksiyonu ya da bir öğrenme kuralı

gerektirmemesidir. Ancak MAML algoritmasının (Antoniou vd., 2018)’ nin belirttiği gibi bazı dezavantajlarından dolayı çok çeşitli varyasyonları geliştirilmiştir. İlk olarak, MAML’ ın yazarları meta-güncelleme sırasındaki hesaplama maliyeti yüksek olan ikinci türev hesabından kaçınmak için “First-order MAML (FOMAML)” ortaya atmışlardır. İkinci türev hesabı ortadan kalktığı durumda da hala başarılı sonuçlar elde edildiğini, buna karşın hesaplama yükünün azaldığını ortaya koymuşlardır. Bu nedenle literatür çalışmalarında MAML dendiğinde aksi belirtilmedikçe “FOMAML” kastedilmektedir.

En temel hatta ilk ortaya atılan derin-öğrenme temelli meta-öğrenme algoritması olan MAML algoritmasının çalışma prensibi Şekil 2’ de belirtilen yaklaşım ile aynıdır. Hatta bu resim orijinal MAML makalesinden alınmıştır. Dış ve iç döngüden oluşan bu algoritmada dış döngüde genel meta-bilgiler çıkarılarak uygun başlangıç ağırlıkları bulunmaya çalışılır. İç döngüde ise bu başlangıç ağırlıkları kullanılarak yeni görevlerdeki az sayıda örnek için hızlı adaptasyon sağlanmaya çalışılır.

Algorithm 1: MAML

```

input :  $p(T)$  : Görevlerin dağılımı
         $\alpha, \beta$  : adım genişliği hiper-parametreleri
1  $\theta$  parametresini rastgele başlat
2 while durma kriterleri sağlanmadığı sürece do
3   bir yığın görev örnekle:  $T_i \sim p(T)$ 
4   for each  $T_i$  do
5     Gradyan değerlerini hesapla:  $\nabla_{\theta} \mathcal{L}_{T_i}(f_{\theta})$ 
6     Gradyan iniş algoritması ile parametre güncelle:
7      $\theta'_i = \theta - \alpha \nabla_{\theta} \mathcal{L}_{T_i}(f_{\theta})$ 
8   end
9   Meta parametreleri güncelle:
10   $\theta \leftarrow \theta - \beta \nabla_{\theta} \sum_{T_i \sim p(T)} \mathcal{L}_{T_i}(f_{\theta'_i})$ 
11 end

```

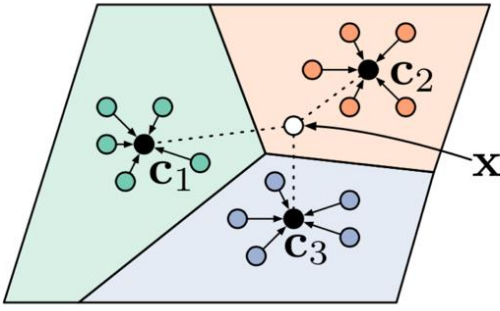
Şekil 3: MAML yöntemi sözde-kodu (Kaynak: Finn vd., 2017).

Şekil 3’te Finn vd.’ nin MAML makalesinde paylaştıkları sözde-kod gösterilmektedir. Satır 1’ de gösterildiği gibi rastgele değerler ile başlatılan Θ , satır 2’ de başlatılan her bir dış döngü adımında güncellenir (satır 8’ e bakınız). Her bir döngü adımında T görev kümesinden rastgele seçilen i adet görevin olduğu bir grup görev (batch) için eğitim K örnek kullanılarak gerçekleştirilir (satır 5). Bir sonraki adımda ise satır 6’ da görüldüğü gibi bu görev kümesindeki test örnekleri kullanılarak adaptasyon aşamasındaki parametre, Θ , güncellenir. Mevcut yığındaki tüm görevler bittiğinde (iç döngü) yeni güncellenmiş parametreler ile hesaplanan hata değeri meta-öğrenme aşamasındaki eğitim hatası olarak değerlendirilir ve bu değere göre meta-model parametreleri, Θ , güncellenir (satır 8). Ağırlık güncellemesi Stokastik Gradyan İniş (SGD) yöntemi ile yapılır ve istenen hata değerine ulaşılan kadar ya da tanımlanan tüm görev yığınları bitene kadar algoritma devam eder.

2.2. ProtoNet Algoritması

Kısaca ProtoNet olarak adlandırılan Prototypical Networks algoritması her bir sınıfın bir prototip ile temsil edilmesi ve yeni gelen bir örneğin kendisine en yakın sınıfa atanması ilkesiyle çalışmaktadır (Snell vd., 2017). Her ne kadar ProtoNet algoritması basitliği ile göze çarpsa da alınan sonuçlar bu algoritmanın sınıflandırma kabiliyetinin yüksek olduğunu göstermiştir. Algoritmada bir f_0 fonksiyonu kullanılarak her bir sınıf için M boyutlu bir prototip vektörü çıkarılmaya çalışılır. Problemden K adet sınıf var ise, her bir k sınıfı için M boyutlu bir

c_k vektörü oluşturulur. Her bir sınıfı temsil eden c_k vektörünün oluşturulmasında ise orijinal algoritmada o sınıfa ait olan veri örnekleri için f_0 fonksiyonu ile elde edilen gösterimlerin ortalaması kullanılmaktadır. Ardından yeni gelen bir örnek için gene aynı f_0 fonksiyonu ile elde edilen gösterim kullanılarak bu gösterimin en yakın olduğu c_k 'nın olduğu sınıf o örneğin ait olduğu sınıf olarak seçilir. Şekil 4' te 3 ayrı sınıf için o sınıfa ait örnekler kullanılarak elde edilmiş olan c_1 , c_2 ve c_3 olmak üzere 3 ayrı prototip gösterilmektedir. Yeni bir X örneği seçilen bir uzaklık metriğine göre kendisine en yakın prototipin olduğu sınıfa atanır. Her bir yeni örnek atanması işlemi ardından hemen o sınıfa ait prototip güncellenir. Doğru sınıfa atanmış bir örnek prototipi iyileştirirken yanlış bir örnek prototipi kötüleştirir. Vektör gösterimleri arasındaki benzerliğin bulunması için makalede en iyi sonuçları *Öklit* mesafesinin verdiği belirtilmiştir. Ancak eldeki probleme göre farklı uzaklık metrikleri de denenebilir. Yanlış sınıflara atanan örnekler için doğan hata değeri kullanılarak SGD algoritmasıyla ağırlıklar güncellenir, dolayısıyla öğrenme süreci geliştirilmiş olur.



Şekil 4: ProtoNet algoritmasında X örneğinin en yakın prototiple temsil edilen sınıfa atanması.

2.3. Kullanılan Veri Kümeleri

Literatürde az örnekle öğrenme problemleri için çeşitli veri kümeleri bulunmaktadır. Az örnekle öğrenme problemleri için yakın zamanda etkili yaklaşımlar geliştirilmiş olmasıyla paralel olarak yeni veri kümeleri de hızla ortaya çıkmaktadır. Bu alanda en çok kullanılan veri kümesi olan Omniglot kümesi (Lake vd., 2011) 50 farklı alfabe için 28x28 boyutunda siyah beyaz el ile yazılmış, her biri 20 örnek içeren 1623 adet farklı karakterden oluşmaktadır. Ancak bu veri kümesi çok basit olması ve dolayısıyla farklı algoritmaların performanslarını değerlendirmede yetersiz oluşu nedeniyle eleştirilmektedir. Bu nedenle çalışmada az örnekle öğrenme veri kümelerinden MiniImageNet ve FC100 veri kümeleri tercih edilmiştir. Bunların yanında, renkli olarak 84x84 boyutunda 200 farklı sınıftan toplam 11788 adet kuş resminden oluşan ve kısaca CUB olarak adlandırılan Caltech-UCSD Birds veri kümesi de az örnekle öğrenme problemlerinde sıklıkla kullanılan veri kümeleri arasındadır. Bu veri kümesi kullanılarak yapılan çalışmalar da literatürde mevcuttur (Alkan, 2020).

2.3.1. MiniImageNet Veri Kümesi

Temel olarak ImageNet olarak bilinen çok büyük ölçekli ILSVRC-12 (Russakovsky vd., 2015) veri kümesinin bir alt kümesi olarak düşünülebilir. Bu büyük veri kümesinin gerektireceği çok yüksek hesaplama ve hatta depolama yükünden kaçınabilmek için oluşturulan çok daha küçük ölçekli bir veri kümesidir. Vinyals vd. (2016) tarafından ImageNet veri kümesinden rastgele olarak 100 adet sınıfın ve her bir sınıfa ait gene rastgele olarak 600 örneğin seçilmesi şeklinde oluşturulmuştur. Ancak örneklerin oluşturulması, az örnekle

öğrenme için görevlerin oluşturulması için değişik örnekleme teknikleri kullanılmaktadır.

Literatürde bu konuda bir ortak kabul bulunmamakla birlikte daha çok Ravi & Larochelle (2017) tarafından tanımlanmış sınıf ayrımları kullanılmaktadır. Buna göre, 100 sınıfın 64 tanesi eğitim, 16 tanesi doğrulama ve 20 tanesi test için kullanılmaktadır. Görüntüler ise 84x84 piksel boyutundadır.

2.3.2. Fewshot-CIFAR100 Veri Kümesi

Kısaca FC100 olarak anılan veri kümesi, MiniImageNet' in oluşturulmasından ilham alınarak oluşturulmuştur. MiniImageNet veri kümesinin az örnekle öğrenme problemleri için Omniglot veri kümesinden daha uygun olduğu görülmüş, bu sebeple benzer yöntemlerle yeni veri kümeleri geliştirilmiştir. FC100 veri kümesi, orijinal CIFAR100 (Krizhevsky, 2009) veri kümesi kullanılarak oluşturulmuş, 100 farklı sınıftan sınıf başına 600 görüntünün olduğu 32x32 boyutunda renkli görüntülerden oluşan bir veri kümesidir (Oreshkin vd., 2018).

Görüntü boyutlarının küçük olması, bu veri kümesini daha zorlu bir hale getirmektedir. Bu veri kümesini zorlu yapan bir diğer özellik de kullanılan sınıf ayrımı ile farklı sınıflar arasında daha bilgi örtüşmesi dolayısıyla benzerliğin daha az indirilmiş olmasıdır. Bu sınıf ayrımının yapılabilmesi için orijinal CIFAR100 veri kümesindeki 100 sınıf bir üst seviyeye çıkılarak 20 ana sınıf altına toplanmıştır. Bu sayede eğitim ve test için seçilen örnekler çok farklı ana sınıflara ait olduklarından aradaki benzerlikler de daha az olacaktır. Eğitim için bu 20 adet üst sınıftan 12 üst sınıf (60 alt sınıf) sadece eğitim için kullanılmaktadır.

3. Araştırma Sonuçları ve Tartışma

Bu bölümde ilk olarak algoritmaların performanslarının incelenmesinde kullanılan deney ortamı anlatılmaktadır. Ardından MAML ve ProtoNet algoritmalarının her birinin kullanılan iki ayrı veri kümesi için verdikleri sonuçlar gösterilmektedir.

3.1. Deney Ortamı

Bölüm 2.3.1' de bahsedildiği gibi az örnekle öğrenme veri kümelerinin oluşturulması kullanılan örnekleme teknikleri açısından farklı olabilmektedir. Ancak sunulan algoritmaların performanslarının adil bir şekilde kıyaslanabilmesi için tüm deneyler için aynı örnekleme yöntemleri ile elde edilmiş görevlerden oluşan veri kümeleri kullanılmalıdır. Az örnekle öğrenme problemleri için başarılı olarak öne sürülen bir algoritmanın belirli bir veri kümesi için verdiği sonuçları literatürde farklı algoritmaların sunduğu sonuçlar ile karşılaştırılırken görev kümelerinin farklı örnekleme teknikleri ile oluşturulmuş olması nedeniyle adil olmayabilmektedir (Chen vd., 2019). Bu nedenle, sadece meta-öğrenme yaklaşımları için görev kümelerinin oluşturulmasının hedeflendiği, bu algoritmalar için veri katmanı sağlayan kütüphaneler yazılmaktadır. Bu çalışmadaki deneyler için Pytorch derin öğrenme kütüphanesi ile uyumlu açık kaynak Torchmeta kütüphanesi seçilmiştir (Deleu vd., 2019). Torchmeta ve benzeri kütüphaneler ile algoritma ile veri sağlama katmanları tamamen birbirlerinden ayrılmakta, tüm algoritmalar için ortak bir alt yapı sunulmaktadır. Bu kütüphaneye veri kümesi tanımlanabileceği gibi, literatürde tanımlanmış olan çok sayıda veri kümesi de zaten bu veri yüklemeye kütüphanesinde hazır olarak bulunmaktadır.

MAML ve ProtoNet algoritmaları hakkında bazı hiper-parametre değerleri seçilmelidir. Örneğin, başlangıç ağırlıklarının nasıl belirleneceği, kaç örnek ya da kaç sınıf kullanılacağı, bunun yanında algoritmaya spesifik değerlerin ne olacağı gibi kararlar verilmelidir. İlk olarak iki çeşit eğitim başlangıç yöntemi kullanılmaktadır: rastgele ve önceden eğitilmiş (pre-train). Başlangıç ağırlıklarının rastgele oluşturulması sırasında ise aynı “seed” değeri kullanılarak adil bir kıyaslama yapılması hedeflenmiştir. Önceden eğitilmiş ağırlıklar ile meta-öğrenmenin başlatılması için ise öncelikle modelin belirli bir veri kümesi kullanılarak standart bir şekilde eğitilmesi gerçekleştirilir. Literatürde ön eğitim işlemi için ImageNet’ in kullanıldığı çalışmalar mevcuttur ve bu eğitim ile bulunan ağırlıklar deney için çok iyi başlangıç durumu sunmaktadırlar. Ancak bu büyüklükteki verinin işlenmesi hem zaman hem de donanım gereksinimi açısından çok zordur. Bu

nedenle bu çalışmada ImageNet veri kümesinin bir alt kümesi olan Tiny ImageNet (Le & Yang, 2015) veri kümesi kullanılarak önceden eğitilmiş ağırlıklar elde edilmiştir. Bu veri kümesinde her biri 500 eğitim, 50 doğrulama ve 50 test görüntüsünden oluşan 200 sınıftan 120000 farklı görüntü bulunmaktadır.

Evrişimli Sinir Ağı (ESA) mimarisi (backbone) olarak ise her iki algoritma için literatürde en çok tercih edilen 4 katmanlı bir konvolüsyon blok yapısı kullanılmıştır (Vinyals vd., 2016). Her bir blokta 3x3 boyutunda 64 adet filtre ve bunu takip eden normalizasyon (Ioffe & Szegedy, 2015) ve 2x2 havuzlama katmanları bulunmaktadır. Aktivasyon fonksiyonu olarak ReLU kullanılmaktadır. Optimizasyon algoritması olarak yine literatürdeki çalışmalarda kullanıldığı gibi 10^{-3} öğrenme oranı ile Adam seçilmiştir. MAML ve ProtoNet algoritmalarının test edilmesi için belirlenmiş olan tüm hiper-parametreler Tablo 1’ de gösterilmektedir.

Tablo 1. Seçilen hiper-parametreler ve değerleri.

Hiper-parametre	Algoritmada Kullanılan Hiper-parametre Değerleri	
	MAML	ProtoNet
Sınıf sayısı (way)	5 – 10	5 – 10
Örnek sayısı (shot)	1 – 5	1 – 5
Eğitim adım sayısı	1 – 5	–
Test adım sayısı	5 – 10	–
Adım genişliği	0,1 – 0,2 – 0,4	–
Yığın boyutu (batch)	8	8
Adam Öğrenme oranı	10^{-3}	10^{-3}

3.2. MAML Sonuçları

Her iki veri kümesi için de MAML eşit sayıda iterasyon (epoch) boyunca eğitilmiştir. Algoritmaların gidişatına göre 200 iterasyonun yeterli olduğuna karar verilmiştir. 200 iterasyon boyunca işlenecek görüntü sayısı ise seçilen yığın boyutuna ve seçilen sınıf ve örnek sayısına göre değişiklik göstermektedir. Bu nedenle tüm veri kümesinin kullanılabilceği bir iterasyon sayısı seçilmelidir. Daha somut bir şekilde ifade etmek gerekirse, MiniImageNet 64 sınıf ve her sınıftan 600 görüntü ile toplam $64 \times 600 = 38400$ adet görüntüden oluşmaktadır. Sınıf sayısının 5 (5-yollu), örnek sayısının da 5 olarak alındığı durumda her bir iterasyonda $5 \times 5 \times (\text{yığın boyutu})$ adet görüntü işlenecektir. Yığın boyutu 8 alınır ise, 1 iterasyonda 200 görüntü işlenecek; toplam iterasyon sayısı 200 olarak belirlendiğinde toplam 40000 görüntü işlenmiş olacaktır. Eğer eldeki eğitim veri kümesinde bu sayıdan daha fazla sayıda görüntü var ise bu görüntüler kullanılmamış olacaktır.

3.2.1. MiniImageNet Üzerindeki Sonuçları

MAML algoritması rastgele ağırlıklarla başlatılarak 200 iterasyon süresince eğitilmiş, ardından test kümesi üzerinde 600 iterasyon süresince test edilmiştir. Literatürdeki çalışmalarda olduğu gibi %95 güven aralığı içerisindeki test doğruluk oranları rapor edilmiştir. Tablo 2’ de gösterildiği üzere 3 çeşit adım

genişliği ve 4 çeşit adım sayısı değerleri için elde edilen sonuçlar sunulmuştur.

Adım genişliği hiper-parametresi için 0,1 değerinin en iyi değer olarak ortaya çıktığı görülmektedir. 5-yönlü deneyler için sonuçlar incelendiğinde farklı adım genişlikleri için doğruluk değerinin %4 ila %7 arasında değişiklik gösterdiği görülmektedir. Tek bir eğitim adımının gerçekleştiği durumda test için 5 yerine 10 örnek kullanılmasının MiniImageNet veri kümesi için alınan sonuçları arttırdığı gözlemlenmektedir. Ancak bu artış genellikle %1 seviyelerinde kalmıştır. Yine 5-yönlü sonuçlara göre, tüm durumlar için eğitim örnek sayısının artırılmasının (1-örnek yerine 5 örnek kullanılması) doğruluk oranını artırdığı gözlemlenmiştir. Sadece adım genişliğinin 0,1 olduğu satırlar dikkate alındığında bu iyileşme oranının %15’ e kadar çıkabildiği görülmektedir. Benzer bir durum 10-yönlü sonuçlar için de geçerlidir.

Tablo 3’ te orijinal MAML makalesinde (Finn vd., 2017) belirtilen hiper-parametre değerleri kullanıldığında alınan sonuçlar ve orijinal makalede belirtilen sonuçlar gösterilmektedir. Kurulan deney ortamında yapılan deneylerde orijinal makaledeki değerlerden oldukça düşük değerler elde edilebilmiştir. Bunun üzerine MAML makalesinde de kullanıldığı gibi önceden eğitilmiş ağırlıklar kullanılmıştır. Ancak önceden eğitim için bölüm 3.1’de anlatıldığı üzere Tiny

ImageNet veri kümesi kullanılmış; çok büyük boyutlu ImageNet verisi kullanılmamıştır.

Tablo 2. MAML'ın MiniImageNet için doğruluk oranları.

Adım sayıları	Adım genişliği	5-yönlü		10-yönlü	
		1- örnek	5- örnek	1- örnek	5- örnek
Eğitim: 1 Test: 5	0,1	32,91 ± 0,25	48,01 ± 0,24	21,11 ± 0,14	32,99 ± 0,15
	0,2	31,34 ± 0,24	46,70 ± 0,26	21,23 ± 0,14	32,94 ± 0,15
	0,4	30,80 ± 0,23	42,52 ± 0,24	19,45 ± 0,13	28,97 ± 0,14
Eğitim: 1 Test: 10	0,1	33,34 ± 0,24	49,09 ± 0,25	21,64 ± 0,14	33,24 ± 0,16
	0,2	31,47 ± 0,24	46,80 ± 0,25	21,29 ± 0,13	32,91 ± 0,15
	0,4	30,92 ± 0,24	42,63 ± 0,25	19,96 ± 0,13	29,52 ± 0,15
Eğitim: 5 Test: 5	0,1	33,29 ± 0,23	50,89 ± 0,25	21,83 ± 0,14	35,48 ± 0,16
	0,2	31,54 ± 0,24	48,06 ± 0,25	21,53 ± 0,13	34,65 ± 0,15
	0,4	28,91 ± 0,23	46,51 ± 0,26	20,12 ± 0,13	32,85 ± 0,15
Eğitim: 5 Test: 10	0,1	34,22 ± 0,24	50,69 ± 0,25	22,98 ± 0,14	36,21 ± 0,16
	0,2	32,33 ± 0,24	48,78 ± 0,26	22,48 ± 0,13	35,32 ± 0,16
	0,4	30,49 ± 0,22	47,18 ± 0,27	21,12 ± 0,13	33,54 ± 0,15

Tablo 3. Kurulan deney ortamında alınan sonuçların orijinal sonuçlarla karşılaştırılması.

Veri Kümesi: MiniImageNet	5-yönlü	
	1- örnek	5- örnek
MAML makalesi	48,70 ± 1,84	63,11 ± 0,92
Kurulan deney düzeneği	45,90 ± 0,31	59,10 ± 0,26

Model, Tiny ImageNet veri kümesi ile eğitilmiş ağırlıklar ile başlatıldığında orijinal MAML makalesinde açıklanan sonuçlara yakın değerler elde edilmiştir. MAML makalesinde Tablo 3' te de gösterildiği gibi 5-yönlü 1-örnek için $48,70 \pm 1,84$ aralığındaki doğruluk oranı; 5-yönlü 5-örnek verildiği durumda ise $63,11 \pm 0,92$ doğruluk oranı elde edilmiştir. Tiny ImageNet veri kümesi ile eğittiğimiz modelde ise sırasıyla $45,90 \pm 0,31$ ve $59,10 \pm 0,26$ değerleri elde edilmiştir. Orijinal çalışmada kullanılan ImageNet veri kümesine göre çok küçük boyutlu bir veri kümesi kullanıldığından elde edilen %3' lük fark makul görünmektedir. Önceden eğitilmiş ağırlıklarla başlamanın önemini de net olarak ortaya çıktığı bu sonuçlar ile ayrıca kurulan düzeneğin de doğru çalıştığı doğrulanmış olmaktadır.

3.2.2. Fewshot-CIFAR100 Üzerindeki Sonuçları

MAML algoritmasının MiniImageNet üzerinde yapılan tüm deneyleri kısaca FC100 olarak bilinen Fewshot-CIFAR100 üzerinde de tekrarlanmıştır. Gene 0,1 – 0,2 ve 0,4 olmak üzere 3 farklı adım genişliği ve 4 farklı adım değeri denenmiştir. Algoritma, yine 200 iterasyon boyunca eğitilmiş ve elde edilen sonuçlar 600 iterasyon boyunca test edilmiştir. Test doğruluk oranları için %95 güven aralığı içindeki değerler rapor edilmiştir. Tablo 4' te MAML algoritması rastgele ağırlıklar ile başlatıldığında elde edilen sonuçlar görülmektedir.

Yapılan deneyler sonucunda, adım genişliği parametresi için MiniImageNet ile benzer sonuçların elde edilmiştir. Sadece 1 adet deney konfigürasyonu hariç genel olarak 0,1 adım genişliğinin en iyi değer olduğu söylenebilir. 5-yönlü deneyler için 0,1 adım genişliği incelendiğinde, hem 1-örnek hem de 5-örnekli deneylerde daha iyi sonuçlar elde edilmiştir. Bu iyileşmenin %5 ila %10 arasında olduğu tabloda görülmektedir.

Hem 5-yönlü, hem 10-yönlü deney sonuçlarına göre tek bir eğitim adımının seçildiği durumda test sürecinde 5 yerine 10 adım değerinin kullanılmasının iyileşme yönünde çok kısıtlı bir

etkisinin olduğu gözlemlenmektedir. Yine hem 5-yönlü, hem 10-yönlü sonuçlara göre örnek sayısının artırılması doğruluk oranında belirgin bir artışa sebep olmaktadır. 0,1 adım genişliği değerinin kullanıldığı durumlarda aradaki farkın yaklaşık olarak %15 olduğu görülmektedir.

ProtoNet algoritmasının orijinal makalesindeki sonuçlar ile kurulan deney düzeneğinde elde edilen sonuçlar da karşılaştırılmıştır. Orijinal çalışmada büyük ölçekli ImageNet veri kümesi ile eğitilmiş başlangıç ağırlıkları ile başlanmasıyla 5-yönlü 1-örnek ve 5-yönlü 5-örnek deneyler için sırasıyla $58,9 \pm 1,9$ ve $71,5 \pm 1,0$ doğruluk oranları elde edilmiştir. Bu çalışmada ise Tiny ImageNet ile elde edilen başlangıç ağırlıklarının kullanıldığı durumda orijinal makalede belirtilen değerlerden yaklaşık %3 daha düşük değerler elde edilebilmiştir. Aradaki bu farkın önceden eğitim için kullanılan veri kümesi büyüklüğü açısından makul olduğu düşünülmektedir. Ayrıca, önceden eğitilmiş ağırlıklar ile algoritma başlatıldığında rastgele ağırlıklarla başlatmaya göre doğruluk oranlarında 5-yönlü 1-örnek için %12' lik; 5-yönlü 5-örnek deneyler için neredeyse %9' luk bir iyileşme elde edilmiştir.

3.3. ProtoNet Sonuçları

ProtoNet algoritması deneyleri için de 200 iterasyon boyunca model eğitilmiştir.

3.3.1. MiniImageNet Sonuçları

MiniImageNet veri kümesi üzerinde ProtoNet algoritmasının almış olduğu doğruluk oranları hem algoritmanın rastgele başlangıç durumunda hem de eğitilmiş ağırlıklara başlatıldığı durumlar için Tablo 5' te verilmiştir. Eğitilmiş ağırlıklarla başlangıç deneyleri için Tiny ImageNet veri kümesi üzerinden elde edilen ağırlıklar kullanılmıştır. Tablo 5' te verilen sonuçlar, 200 iterasyon boyunca eğitim veri kümesi üzerinde eğitilen ilgili modellerin ardından 600 iterasyon süresince test veri kümesi

üzerinde verdikleri doğruluk oranlarının %95 güven aralığı içerisindeki değerlerdir.

Tablo 4. MAML' in FC100 için doğruluk oranları.

Adım sayıları	Adım genişliği	5-yönlü		10-yönlü	
		1- örnek	5- örnek	1- örnek	5- örnek
Eğitim: 1 Test: 5	0,1	40,90 ± 0,31	56,17 ± 0,28	27,58 ± 0,17	41,34 ± 0,18
	0,2	40,26 ± 0,30	54,41 ± 0,29	27,34 ± 0,18	38,32 ± 0,17
	0,4	35,59 ± 0,28	49,38 ± 0,28	23,77 ± 0,14	33,27 ± 0,15
Eğitim: 1 Test: 10	0,1	40,89 ± 0,30	56,69 ± 0,29	28,07 ± 0,17	42,41 ± 0,18
	0,2	40,08 ± 0,29	54,33 ± 0,29	27,59 ± 0,18	38,89 ± 0,18
	0,4	35,77 ± 0,28	49,45 ± 0,28	24,18 ± 0,16	34,97 ± 0,17
Eğitim: 5 Test: 5	0,1	41,67 ± 0,30	58,73 ± 0,29	28,90 ± 0,17	44,98 ± 0,17
	0,2	35,10 ± 0,26	58,06 ± 0,28	29,32 ± 0,17	43,30 ± 0,18
	0,4	31,08 ± 0,24	52,40 ± 0,28	21,08 ± 0,14	37,13 ± 0,15
Eğitim: 5 Test: 10	0,1	42,65 ± 0,31	59,49 ± 0,29	29,58 ± 0,18	46,02 ± 0,17
	0,2	38,47 ± 0,29	58,49 ± 0,29	29,54 ± 0,17	44,42 ± 0,18
	0,4	34,20 ± 0,28	52,06 ± 0,30	21,47 ± 0,16	38,65 ± 0,18

Tablo 5. ProtoNet' in MiniImageNet için doğruluk oranları.

Başlangıç	5-yönlü		10-yönlü	
	1- örnek	5- örnek	1- örnek	5- örnek
Rastgele	26,83 ± 0,17	45,06 ± 0,22	16,98 ± 0,11	31,14 ± 0,13
Eğitilmiş	38,25 ± 0,25	50,62 ± 0,23	20,48 ± 0,12	35,88 ± 0,14

Alınan 5-yönlü ve 10-yönlü deney sonuçları incelenirse, eğitim sürecinde örnek sayısının artırılmasının bütün deneylerdeki doğruluk oranlarını artırdığı görülmektedir. Bütün parametrelerin sabit tutularak sadece örnek sayısının artırılması, yapılan bütün deneylerde alınan doğruluk oranlarını neredeyse 2 katına çıkarmıştır. Doğruluk oranlarındaki bu yükseliş durumu hem 5-yönlü hem de 10-yönlü deneylerde aynı şekilde göze çarpmaktadır. Benzer bir durum MAML algoritmasının almış olduğu doğruluk oranlarında da görülmüştü ve modele eğitim sürecinde artan sayıda örnek verildiğinde alınan doğruluk oranlarının arttığı yapılan deneylerde gösterilmiştir.

ProtoNet algoritmasının başlangıcı rastgele seçildiğinde, yani sıfırdan eğitime başlandığında alınan doğruluk oranları ile eğitilmiş ağırlıklarla başlatıldığında alınan doğruluk oranları karşılaştırıldığında belirgin farklılıklar gözlemlenmiştir. Eğitilmiş ağırlıklarla başlangıç durumunda doğruluk oranlarında elde edilen artış başarısının benzeri MAML algoritmasının yine aynı veri kümesi üzerindeki sonuçlarında da gözlemlenmişti. Bu sonuçları incelediğimizde 5-yönlü 1-örnek sonuçları için %12' lik bir artış gözlemlenirken 5-yönlü 5-örnek sonuçları için ise kabaca %5' lik bir iyileşme gözlemlenmiştir.

Orijinal makalede elde edilen sonuçlarla karşılaştırıldığında, bu sonuçlara en yakın değerlerin eğitilmiş ağırlıklarla başlangıç için Tiny ImageNet veri kümesi üzerinden elde edilen ağırlıkların kullanıldığı durumda elde edildiği gözlemlenmiştir. Büyük ImageNet veri kümesi üzerinden elde edilmiş ağırlıklarla başlatılan orijinal makalede 5-yönlü 1-örnek için $42,90 \pm 0,6$ değeri gösterilmiştir. Bu çalışmada elde edilen değerler orijinal makalede sunulan bu sonuçlarla karşılaştırıldığında, bu sonuçlar arasında yaklaşık %4' lük bir fark gözlemlenmiştir. Başlangıç ağırlıkları orijinal makalede büyük ImageNet veri kümesi üzerinden elde edilirken bizim yaptığımız çalışmada ise daha

küçük ölçekli olan Tiny ImageNet veri kümesi üzerinden elde edilmesinden dolayı aradaki %4' lük bu farkın kabul edilebilir olduğu düşünülmektedir.

Bütün deneylerde modelin temsil boyutu ve aynı zamanda gizli katman boyutu 64 olarak belirlenip sonuçlar bu değere göre raporlanmıştır. Sadece 5-yönlü deneyler için, bu iki parametrenin 64 yerine 128 değeriyle çalıştırıldığında elde edilen doğruluk oranları Tablo 6' da paylaşılmıştır. Tablo 6' da kullanılan TB kısaltması temsil boyutunu belirtirken SK kısaltması ise gizli katman boyutunu belirtmektedir. Temsil boyutunun 64' ten 128' e çıkartılması ile rastgele başlangıca sahip 1-örnek deneylerinde yaklaşık olarak %2' lik bir artış gözlemlenirken rastgele başlangıca sahip 5-örnek deneylerinde ise yaklaşık %2' lik bir azalma gözlemlenmiştir.

3.3.2. Fewshot-CIFAR100 Sonuçları

Fewshot-CIFAR100 ya da kısaca FC100 olarak adlandırılan veri kümesi üzerinde ProtoNet algoritmasının almış olduğu doğruluk oranları hem algoritmanın rastgele başlangıç durumunda hem de eğitilmiş ağırlıklarla başlatıldığı durumlar için Tablo 7' de verilmiştir. Eğitilmiş ağırlıklarla başlangıç deneyleri için Tiny ImageNet veri kümesi üzerinden elde edilen ağırlıklar kullanılmıştır. Tablo 7' de verilen sonuçlar, 200 iterasyon boyunca eğitim veri kümesi üzerinde eğitilen ilgili modellerin ardından 600 iterasyon süresince test veri kümesi üzerinde verdikleri doğruluk oranlarının %95 güven aralığı içerisindeki değerlerdir.

Alınan 5-yönlü ve 10-yönlü deney sonuçlarını incelersek, eğitim sürecinde örnek sayısının artırılmasının bütün deneylerdeki doğruluk değerini artırdığı görülmektedir. Bütün parametrelerin sabit tutularak sadece örnek sayısının 1' den 5' e çıkartılması yapılan bütün deneylerde alınan doğruluk oranlarını

yaklaşık olarak %20 arttırmıştır. Doğruluk oranlarındaki bu yükseliş durumu hem 5-yönlü hem de 10-yönlü deneylerde aynı şekilde gözükmektedir. Benzer bir durum MAML algoritmasının almış olduğu doğruluk oranlarında da gözlemlenmişti ve modele eğitim için daha fazla sayıda örnek verildiğinde alınan doğruluk oranlarının arttığı yapılan deneylerde gösterilmiştir.

ProtoNet algoritmasının rastgele ağırlıklarla başlatılması yerine eğitilmiş ağırlıklarla başlatıldığında alınan doğruluk oranları karşılaştırıldığında yine aynı şekilde belirgin artışlar

gözlemlenmiştir. Eğitilmiş ağırlıklarla başlangıç durumunda doğruluk oranlarında elde edilen artış başarısının benzeri MAML algoritmasının yine aynı veri kümesi üzerindeki sonuçlarında da gözlemlenmişti. Bu sonuçları incelediğimizde 5-yönlü 1-örnek sonuçları için %10' luk artış gözlemlenirken 5-yönlü 5-örnek sonuçları için ise %7' lik bir iyileşme gözlemlenmiştir. 10-yönlü 1-örnek sonuçları için %10' luk bir artış gözlemlenirken 10-yönlü 5-örnek sonuçları için ise %8' lik iyileşme gözlemlenmiştir.

Tablo 6. ProtoNet' in MiniImageNet üzerinde ek parametreler için sonuçları

Başlangıç	Örnek sayısı	5-yönlü		
		TB:64, SK: 64	TB:128, SK:64	TB:64, SK:128
Rastgele	1-örnek	26,83 ± 0,17	28,07 ± 0,18	28,45 ± 0,19
	5-örnek	45,06 ± 0,22	44,95 ± 0,23	42,86 ± 0,24
Eğitilmiş	1-örnek	38,25 ± 0,25	32,54 ± 0,22	32,41 ± 0,23
	5-örnek	60,10 ± 0,25	53,31 ± 0,25	51,98 ± 0,24

Tablo 7. ProtoNet' in FC100 için doğruluk oranları.

Başlangıç	5-yönlü		10-yönlü	
	1-örnek	5-örnek	1-örnek	5-örnek
Rastgele	40,86 ± 0,28	60,62 ± 0,28	27,49 ± 0,16	48,24 ± 0,18
Eğitilmiş	50,07 ± 0,31	67,67 ± 0,25	37,05 ± 0,19	56,47 ± 0,17

Orijinal makalede elde edilen sonuçlarla karşılaştırıldığında, bu sonuçlara en yakın değerlerin eğitilmiş ağırlıklarla başlangıç için Tiny ImageNet veri kümesi üzerinden elde edilen ağırlıkların kullanıldığı durumda elde edildiği gözlemlenmiştir. Büyük ImageNet veri kümesi üzerinden elde edilmiş ağırlıklarla başlatılan orijinal makalede 5-yönlü 1-örnek için $55,5 \pm 0,7$ değeri gözlemlenirken 5-yönlü 5-örnek için $72,0 \pm 0,6$ değeri gözlemlenmiştir. Elde edilen değerlerle orijinal ProtoNet makalesinde verilen sonuçları karşılaştırdığımızda, bu sonuçlar arasında yaklaşık %5' lik bir fark gözlemlenmiştir. Başlangıç ağırlıkları orijinal makalede büyük ImageNet veri kümesi üzerinden elde edilirken bizim yaptığımız çalışmada ise daha küçük ölçekli olan Tiny ImageNet veri kümesi üzerinden elde edilmesinden dolayı aradaki %5' lik bu farkın kabul edilebilir olduğu görülmektedir.

Yapılan bütün deneylerde modelin temsil ve aynı zamanda gizli katman boyutu değerleri 64 olarak belirlenip sonuçlar bu değere göre raporlanmıştır. Sadece 5-yönlü deneyler için, bu iki parametrenin 64 yerine 128 değeriyle çalıştırıldığında elde edilen doğruluk oranları Tablo 8' de paylaşılmıştır. Tablo 6' da kullanılan TB kısaltması temsil boyutunu belirtirken SK kısaltması ise gizli katman boyutunu belirtmektedir. Temsil boyutunun 64' ten 128' e çıkartılması ile rastgele başlangıç durumunda 1-örnek ve 5-örnek kullanılarak yapılan deneylerde %1' lik bir azalma görülmüştür. Yalnız gizli katman boyutunun 64' den 128' e yükseltilmesi ile rastgele başlangıca sahip 1-örnek deneylerinde yaklaşık olarak %2' lik bir artış gözlemlenirken rastgele başlangıca sahip 5-örnek deneylerinde ise aynı sonuçlara ulaşıldığı gözlemlenmiştir. Eğitilmiş ağırlıklarla başlangıç yapılan deneylerde ise daha farklı durumlar gözlemlenmiştir. Yalnız temsil boyutunun 64' ten 128' e çıkartılması ile eğitilmiş ağırlıklarla başlangıç yapılan 1-örnek deneylerinde kabaca %6 civarında bir azalma görülürken eğitilmiş ağırlıklarla başlangıç

yapılan 5-örnek deneylerinde ise yaklaşık %4' lük bir azalma görülmüştür.

4. Sonuç

Klasik makine öğrenmesi yöntemleri ile elde edilen modele, eğer bu eğitim sürecinde görmediği farklı bir görüntü verilirse modelin bu görüntüye adaptasyon zorluğu klasik makine öğrenmesi yöntemlerindeki en temel problem olarak gözükmektedir. Bu sebepten dolayı her problem için yeni bir model sıfırdan eğitim sürecinden geçirilir. Diğer bir deyişle, oluşturulan bu modeller genel bir çözüm sunabilmekten çok özel bir çözüm olarak kalmışlardır ve bu sebepten dolayı genelleme performansları düşüktür. Modelin farklı veri kümeleri için bu şekilde sıfırdan sürekli olarak eğitilmesi hem zaman hem de donanım açısından zorlayıcı bir problem olmaktadır. Bu gibi zorlukların üstesinden gelebilmek ve benzer problemler için daha genel modeller eğitebilmek için meta-öğrenme yaklaşımları önerilmiştir. Bu yaklaşımların eğitim süreçlerine getirdiği değişiklikler sayesinde oluşacak olan modelin genelleme performansı artmaktadır. Eğitim süreci boyunca, aynı yığın içerisinde çok sayıda görev oluşturulup modelin bu görevler üzerindeki genel başarısı arttırılmaya çalışılır. Bu sayede, ortaya çıkan modelin adaptasyon zorluğu ortadan kalkacaktır ve daha önce eğitim sürecinde görmediği farklı bir görüntüye daha çabuk ve doğru bir şekilde adapte olacaktır.

Farklı parametre değerlerini deneyerek yaptığımız deneyler sonucunda MAML algoritması için en etkili parametrenin adım genişliği olduğu gözlemlenmiştir. Uygun bir adım genişliği değeri ile algoritmanın başarısı daha da artarken uygun olmayan bir adım genişliği değeri ile algoritmanın başarısının azaldığı gözlemlenmiştir. Adım genişliğinin dikkatli bir şekilde denenecek uygun değerinin belirlenmesi alınacak sonuçlar açısından daha etkili olacaktır ve sonuçlar üzerindeki olumsuz

etkisinin daha aza indirilebileceği gözlemlenmiştir. Bir başka önemli durum ise her görevin kaç adet örnek görüntü içereceğidir. Hem MAML hem de ProtoNet algoritmaları için, eğitim sürecinde modele gösterilen örnek sayısının artırılmasının olumlu sonuçları gözlemlenmiştir ve daha başarılı sonuçlar elde edilmiştir. Yapılan bütün deneylerde, her bir parametrenin ayrı bir değişken şeklinde ele alınması elimizde halihazırda bulunan hesaplama gücü açısından mümkün

olmadığından dolayı literatürde yapılan çalışmalarda sıklıkla kullanılan ve iyi sonuçlar elde ettiği belirlenen parametre değerleri seçilmiştir. Gerektiğinde elimizdeki hesaplama gücüne uygun olacak şekilde bu parametreler değiştirilmiştir. Bu duruma örnek vermek gerekirse, elimizdeki kaynakların limiti aşılmadan deney sonuçlarını doğru bir şekilde elde edebilmek adına hem MAML algoritması hem de ProtoNet algoritması için daha küçük bir yığın boyutu seçilmiştir.

Tablo 8. ProtoNet' in FC100 üzerindeki ek parametre sonuçları

Başlangıç	Örnek sayısı	5-yönlü		
		TB:64, SK: 64	TB:128, SK:64	TB:64, SK:128
Rastgele	1-örnek	40,86 ± 0,28	40,27 ± 0,28	42,18 ± 0,28
	5-örnek	60,62 ± 0,28	59,64 ± 0,27	60,72 ± 0,28
Eğitilmiş	1-örnek	50,07 ± 0,31	49,65 ± 0,30	44,41 ± 0,29
	5-örnek	67,67 ± 0,25	70,09 ± 0,25	63,18 ± 0,26

Hem MAML hem de ProtoNet algoritmalarının elde ettikleri sonuçlara baktığımızda, modele gösterilen örnek sayısı arttıkça ProtoNet algoritmasının elde ettiği doğruluk oranlarının MAML algoritmasına yaklaştığı ve hatta daha iyi sonuçlar elde ettiği gözlemlenmiştir. Bu artışı gözlemlediğimizde, ProtoNet algoritmasının öğrenilmesi istenen her sınıf için o sınıf özelliklerini en iyi şekilde temsil edecek bir prototip oluşturmaya çalışması ve prototip oluşturulması için gösterilen örnek sayısının artması sonucunda oluşturulan bu prototiplerin daha doğru sonuçlar elde ettiği sonucuna varılmıştır. MAML algoritmasında ise bu durum daha farklı şekilde işlemektedir. Algoritma, eğitim süreci boyunca tüm sınıfların ortak özelliklerini en iyi şekilde temsil edecek ortak başlangıç ağırlıkları çıkarmaya, bu şekilde öğrenmeyi genelleştirmeye çalışır.

Yaptığımız bu çalışmada elimizdeki hesaplama kısıtlarından dolayı sabit bir omurga kullandığımız için sonraki çalışmalarda ise bu değişimin sonuçları nasıl etkilediğinin incelenmesi düşünülmektedir. Omurga değiştirilerek yapılacak deneylerin asıl amacı kullanılan ağ modeline göre doğruluk oranlarının nasıl ve neye göre değiştiğini gözlemleyebilmektir. Bu deneylerde omurga modellerinin derinliği ve parametre sayıları gözlemlenerek sonuçların incelenmesi düşünülmektedir. Bunun yanında çok daha yüksek gizli katman boyutları ve temsil boyutları için ve farklı veri kümeleri için algoritmaların performanslarının incelenmesi ileride yapılması düşünülen çalışmalar arasındadır.

Kaynakça

Alkan, M. (2020). Az Örnekle Öğrenme Problemlerinde Derin Öğrenme Temelli Meta-Öğrenme Algoritmalarının Karşılaştırılması (Master's thesis, Fatih Sultan Mehmet Vakıf Üniversitesi, Lisansüstü Eğitim Enstitüsü).

Antoniou, A., Edwards, H., & Storkey, A. (2018). How to train your MAML. 7th International Conference on Learning Representations, ICLR 2019. <http://arxiv.org/abs/1810.09502>

Chen, W.-Y., Liu, Y.-C., Kira, Z., Wang, Y.-C. F., & Huang, J.-B. (2019). A Closer Look at Few-shot Classification. <http://arxiv.org/abs/1904.04232>

Deleu, T., Würfl, T., Samiei, M., Cohen, J. P., & Bengio, Y. (2019). Torchmeta: A Meta-Learning library for PyTorch. <http://arxiv.org/abs/1909.06576>

Finn, C., Abbeel, P., & Levine, S. (2017). Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. 34th International Conference on Machine Learning, ICML 2017, 3, 1856–1868. <http://arxiv.org/abs/1703.03400>

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 770–778. <https://doi.org/10.1109/CVPR.2016.90>

Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. 32nd International Conference on Machine Learning, ICML.

Krizhevsky, A. (2009). Learning Multiple Layers of Features from Tiny Images. ... Science Department, University of Toronto, Tech. <https://doi.org/10.1.1.222.9220>

Lake, B. M., Salakhutdinov, R., Gross, J., & Tenenbaum, J. B. (2011). One shot learning of simple visual concepts. In Proceedings of the 33rd Annual Conference of the Cognitive Science Society.

Le, Y., & Yang, X. (2015). Tiny ImageNet Visual Recognition Challenge.

Oreshkin, B. N., Rodriguez, P., & Lacoste, A. (2018). TADAM: Task dependent adaptive metric for improved few-shot learning. Advances in Neural Information Processing Systems. <http://arxiv.org/abs/1805.10123>

Ravi, S., & Larochelle, H. (2017). Optimization as a Model for Few-Shot Learning. Proceedings of the 5th International Conference on Learning Representations (ICLR 2017), 1–11.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., & Fei-Fei, L. (2015). ImageNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision. <https://doi.org/10.1007/s11263-015-0816-y>

Snell, J., Swersky, K., & Zemel, R. S. (2017). Prototypical Networks for Few-shot Learning. Advances in Neural Information Processing Systems. <http://arxiv.org/abs/1703.05175>

Vinyals, O., Blundell, C., Lillicrap, T., Kavukcuoglu, K., & Wierstra, D. (2016). Matching networks for one shot learning. Advances in Neural Information Processing Systems, Nips, 3637–3645.