|  |  |
| --- | --- |
|  | **T.C.**  **KIRŞEHİR AHİ EVRAN ÜNİVERSİTESİ**  **MHENDİSLİK MİMARLIK FAKÜLTESİ**  **BİLGİSAYAR MÜHENDİLİĞİ**  **LİSANS** |

**Efficient Deep Learning: A Survey on Making Deep Learning Models Smaller, Faster, and Better**

**202511017**

**Kutluhan AZAFLI**

**LİSANS / DERS PROJESİ**

**KIRŞEHİR**

**2024**

Derin öğrenme, bilgisayarlı görü, doğal dil işleme, konuşma tanıma gibi alanlarda devrim yaratırken; bu modellerin parametre sayısı, gecikmesi ve eğitim için gereken kaynaklar giderek artmaktadır. Bu sebeple modelin kalitesinin yanı sıra, modelin boyutu da önemli hale gelmiştir. Derin öğrenme modellerinin doğrudan gerçek dünyada dağıtımı yeterince verimli olmayabilir veya dağıtırken çeşitli zorluklar yaşanabilir. Bunlar arasında sunucu ölçeklendirme, cihazlara özgü dağıtım, gizlilik ve veri hassasiyeti gibi konular bulunmaktadır. Bu sebeple makalede verimlilik kavramı üzerinde durulmuştur.

Çalışmada, modelin çıkarım ve eğitim verimliliği olmak üzere iki ana başlık altında inceleniyor. Çıkarım verimliliği, bir modelin çıkarım sırasında ne kadar hızlı ve küçük olduğunu, eğitim verimliliği ise bir modelin eğitim süresi ve gereksinimleri gibi faktörleri ele alıyor. Ayrıca, pareto-optimalite kavramı üzerinde durularak, model seçiminde dengeli bir yaklaşımın önemi vurgulanıyor.

Derin öğrenmede genel olarak verimliliği arttırmak üzerine ise 5 ana başlık önerilmektedir. Bunlar:

1) Sıkıştırma Teknikleri: Genel olarak modelin mimarisini optimize etmeye yönelik teknikler ve algoritmaları içerir. Tipik bir örnek, bir katmanın ağırlık matrislerini sıkıştırarak modelin boyutunu azaltmaya çalışan kantizasyondur. Bu, ağırlıkların hassasiyetini azaltarak (örneğin, 32 bit kayan nokta değerlerinden 8 bit işaretsiz tamsayılara) kalite kaybını minimumda tutar.

2) Öğrenme Teknikleri: Daha az tahmin hatası yapmak, daha az veri gerektirmek, daha hızlı bir şekilde yakınsamak gibi modelin farklı şekillerde eğitilmesine odaklanan algoritmaları içerir. İyileştirilmiş kalite daha sonra ihtiyaç duyulması halinde parametre sayısının azaltılmasıyla değiştirilebilir. Öğrenme tekniklerinin bir örneği, küçük bir modelin doğruluğunu büyük bir modele benzeterek artıran distilasyondur.

3) Otomasyon: Verilen modelin temel ölçütlerini otomasyon kullanarak iyileştiren araçları içerir. Örneğin, hiper-parametre optimizasyonu, doğruluğu artırmaya yardımcı olan hiper-parametreleri optimize eder, bu da daha az parametreye sahip bir modelle değiştirilebilir. Benzer şekilde, mimari arama da bu kategoriye girer, burada mimari kendisi ayarlanır ve arama, kayıp/doğruluk ve model gecikme, model boyutu gibi diğer metriklerin optimize edildiği bir model bulmaya yardımcı olur.

4) Verimli Mimariler: Temel blokların olduğu kategoridir ve önceden kullanılan temel yöntemlerin (tam bağlantılı katmanlar ve RNN'ler) önemli bir ilerlemesi olarak tasarlanmıştır. Örneğin, evrişimli katmanlar, görüntü sınıflandırması için kullanılan ayrı ağırlıkları öğrenmekten kaçınarak her giriş pikseli için ayrı ağırlıkların öğrenilmesini önler ve aşırı uyumu önler. Benzer şekilde, dikkat katmanları, Seq2Seq modellerindeki Bilgi Darboğazı sorununu çözmüştür. Bu mimariler, verimlilik kazanımları için doğrudan kullanılabilir.

5) Altyapı: Verimli modellerin oluşturulması ve kullanılmasına yardımcı olan altyapı ve araçlar için bir temel sağlamamız gerekir. Bu, Tensorflow, PyTorch gibi model eğitim çerçevelerini içerir (TFLite, PyTorch Mobile gibi verimli modellerin dağıtılması için gereken özel araçlarla birlikte). Verimli modellerden gelen kazanımları değerlendirmek için altyapı ve araçlara bağımlıyız. Örneğin, kantize edilmiş modellerle hem boyut hem de gecikme iyileştirmeleri elde etmek için, çoğu yaygın sinir ağı katmanlarını kantize modda destekleyen çıkarım platformuna ihtiyacımız vardır.

Model optimizasyonunda ise üç ana aşama bulunur: büyütme (Grow), küçültme (Shrink) ve iyileştirme (Improve). Büyütme aşamasında, model kapasitesi artırılarak model kalitesi iyileştirilir. Küçültme aşamasında, modelin boyutu azaltılarak modelin boyutu düşürülürken model kalitesi hafifçe düşebilir. İyileştirme aşamasında ise, modelin kalitesi artırılarak modelin boyutu sabit tutulur.

Makale, bu üç aşamanın kombinasyonundan oluşan iki stratejiyi önermektedir:

1. **Küçült ve İyileştir (Shrink-and-Improve)**: Model boyutunu azaltarak kaliteyi sabit tutmak isteyenler için ideal bir stratejidir. Bu strateji, öncelikle modelin kapasitesini azaltarak başlar, ardından modelin kalitesini iyileştirmek için çeşitli teknikler uygulanır.
2. **Büyüt, İyileştir ve Küçült (Grow-Improve-and-Shrink)**: Kaliteyi artırmak isteyenler için uygundur. Model önce büyütülür, sonra kalitesi artırılır ve son olarak model küçültülerek boyutu azaltılır.

Deneyler, düşük, orta ve yüksek seviye akıllı telefonlarda damıtma ve kuantizasyon tekniklerinin doğruluğu korurken gecikmeyi azaltabileceğini göstermiştir.

Bir modelin boyutu 47.3 kat azaltılmış ve hızı 5.31 kat artırılmıştır, bu da dağıtım verimliliğinde önemli bir iyileşme sağlamıştır.