

Fashion MNIST Üzerinde MLP Modeli ile Çoklu Sınıflandırma: Eğitim, Optimizasyon ve Sonuçlar Raporu

Özet

Bu raporda, Fashion MNIST veri seti üzerinde çok sınıflı görüntü sınıflandırma yapmak için geliştirdiğimiz Yapay Sinir Ağı (MLP) modelinin tasarımı, eğitimi ve değerlendirilmesi ele alınmıştır.

Öncelikle model mimarisi açıklanmış, farklı katmanlar ve aktivasyon fonksiyonlarının seçim nedenleri belirtilmiştir. Ardından, SGD ve Adam optimizasyon algoritmaları kullanılarak modelin performansları karşılaştırılmış ve elde edilen sonuçlar analiz edilmiştir.

Son olarak, modelin eğitim süreci, doğrulama sonuçları ve elde edilen accuracy (doğruluk) ve loss (kayıp) değerleri incelenerek modelin etkinliği değerlendirilmiştir.

Model Mimarisi

Bu modelde, **Batch Normalization** ve **Dropout** gibi düzenleme teknikleri kullanılarak modelin performansı artırılmaya çalışılmıştır. Modelin mimarisi, özellikle derin öğrenme modellerinin eğitiminde sıkça karşılaşılan **aşırı öğrenme (overfitting)** ve **gradyan kaybı** gibi sorunları minimize etmek için tasarlanmıştır.

Model, **28x28 boyutundaki giriş verilerini** düzleştirerek, **Flatten** katmanı ile tek boyutlu hale getirir. İlk gizli katman olarak **300 nöronlu bir Dense katmanı** kullanılır ve burada **ReLU (Rectified Linear Unit)** aktivasyon fonksiyonu tercih edilmiştir. ReLU, negatif değerleri sıfıra indirerek öğrenme sürecini hızlandırır ve gradyan kaybı sorununu engeller. Bu katmandan sonra **Batch Normalization** uygulanır. Batch Normalization, her katman çıktısını normalize ederek öğrenme hızını artırır, ağırlıkların daha hızlı güncellenmesini sağlar ve modelin daha stabil çalışmasını destekler.

Ayrıca, modelde **Dropout** uygulanarak aşırı öğrenme riski azaltılmaya çalışılmıştır. Dropout, her eğitim adımında rastgele olarak bazı nöronları devre dışı bırakır ve böylece modelin fazla uyum yapmasını engeller. Bu modelde, **0.2 oranında bir Dropout** uygulanmıştır, yani her seferinde nöronların %20'si devre dışı bırakılır.

İkinci gizli katman olarak **100 nöronlu bir Dense katmanı** kullanılır ve bu katmanda da **ReLU aktivasyonu** tercih edilmiştir. Yine, bu katmanda da **Batch Normalization** ve **Dropout** uygulanarak modelin genelleme kapasitesi artırılmak istenmiştir. Son olarak, çıktıyı veren **10 nöronlu bir Dense katman** bulunur ve **Softmax aktivasyon fonksiyonu** kullanılarak sınıflandırma işlemi yapılır.

Modelin eğitimi için **Adam optimizasyon algoritması** seçilmiştir ve öğrenme oranı 0.001 olarak belirlenmiştir. Bu optimizasyon algoritması, genellikle derin öğrenme modelleri için verimli ve hızlı sonuçlar verir. Modelin eğitimi, **sparse_categorical_crossentropy** kayıp fonksiyonu ile yapılır ve doğruluk (accuracy) metriği ile modelin başarısı izlenir. Model 30

epoch boyunca eğitilmiş olup, her epoch sonunda doğrulama verisiyle modelin performansı değerlendirilmiştir. Eğitim sırasında **batch size 64** olarak belirlenmiş, böylece her adımda 64 örnek üzerinden eğitim yapılmıştır.

Kullanılan Hiperparametreler, Sonuç Analizleri ve İyileştirme Önerileri

1. Hiperparametreler

Modelin eğitiminde kullanılan önemli hiperparametreler, optimizör, öğrenme oranı, dropout oranı ve batch size'dır. Bu parametreler modelin performansını doğrudan etkileyen faktörlerdir. Optimizasyon algoritması olarak Adam ve SGD arasındaki farklar, öğrenme oranları ve dropout tekniği gibi özellikler üzerinde testler yapılmıştır.

2. Sonuç Analizleri

Adam vs. SGD Optimizer

İlk modelde SGD optimizasyonu kullanıldıktan sonra, Adam optimizasyonunun kullanılması bazı sınıflarda iyileştirmeler sağladı. Özellikle **Shirt** ve **Pullover** gibi sınıflarda recall oranı artarken, precision oranı bir miktar azalmıştır. Bu, recall'ın artırılarak daha fazla doğru tahmin yapılmaya çalışıldığını ancak precision'da bir düşüş görüldüğünü göstermektedir. Diğer sınıflarda ise önemli değişiklikler gözlemlenmemiştir. Özetle, Adam optimizasyonu modelin genel doğruluğunu artırırken, sonuçlar arasında büyük farklar gözlemlenmemektedir.

Adam optimizör genellikle daha hızlı sonuç verirken, SGD daha kararlı bir öğrenme süreci sunar. Test sonuçları, SGD'nin daha düşük test kaybı (daha düşük hata oranı) sağladığını ancak Adam optimizör'ün doğruluk, precision ve recall açısından daha iyi performans gösterdiğini ortaya koymaktadır. Bu doğrultuda, hızlı öğrenme süreci gerektiren durumlarda Adam tercih edilirken, daha stabil ve kararlı bir öğrenme süreci için SGD kullanılabilir.

Öğrenme Oranı

Adam optimizör için farklı öğrenme oranları (0.001 ve 0.005) kullanılarak yapılan testlerde, en iyi performans **0.001** öğrenme oranı ile elde edilmiştir. Bu öğrenme oranı ile modelin doğrulama doğruluğu 0.8914 olarak kaydedilmiştir. Bu, modelin öğrenme sürecinin daha verimli olduğunu ve daha iyi bir genelleme sağladığını göstermektedir.

Dropout ile Model Performansını Artırma

Dropout tekniği, modelin aşırı öğrenmesini (overfitting) engellemek amacıyla kullanılmaktadır. Dropout oranı olarak 0.2 seçildiğinde, modelin doğrulama doğruluğu artmıştır. Ancak, eğitim kaybı arttığı için modelin öğrenmesi biraz daha zorlaşmıştır. Bu durum, modelin eğitim verilerine aşırı uyum sağlamadığını ve daha iyi genelleme yapabilme kapasitesine sahip olduğunu gösterir. Yapılan analizlere göre, **Dropout 0.2** oranı doğrulama doğruluğunu artırmış ve modelin genelleme yeteneğini iyileştirmiştir.

Batch Size'in Etkisi

Batch size parametresi de modelin performansını etkileyen önemli bir faktördür. Batch size 64 kullanıldığında, model daha dengeli bir performans sergileyerek hem doğrulama doğruluğunu yüksek tutmuş hem de kaybı düşük tutmuştur. Batch size 32 ile en yüksek doğrulama doğruluğu elde edilmiş ancak kayıp oranı daha yüksek olmuştur. Batch size 128 ile kayıp oranı düşerken doğrulama doğruluğunda küçük bir azalma gözlemlenmiştir. Bu sonuçlar, batch size 64'ün modelin doğruluğu ve genelleme kabiliyeti açısından daha uygun olduğunu göstermektedir.

Batch Normalization (BN)

Batch Normalization, modelin doğrulama doğruluğunu iyileştirme noktasında önemli bir rol oynamaktadır. Batch Normalization eklenen modelin doğrulama doğruluğu 0.8872 iken, Batch Normalization eklenmeden önceki modelde bu değer 0.8930 olarak kaydedilmiştir. Bu, Batch Normalization eklemenin modelin genel doğrulama doğruluğunu biraz düşürse de, overfitting'i önlemek adına faydalı olduğunu göstermektedir. Ayrıca, Batch Normalization modelin eğitim kaybını bir miktar artırmış olsa da, modelin daha stabil sonuçlar verdiği ve genelleme yeteneğini artırdığı söylenebilir.

3. İyileştirme Önerileri

Modelin doğruluğunu artırmak için hiperparametre optimizasyonu yapılabilir. Grid Search veya Random Search gibi tekniklerle öğrenme hızı, batch size, epoch sayısı gibi hiperparametreler optimize edilerek modelin daha iyi bir doğrulama doğruluğu sağlaması hedeflenebilir.

Modelin daha karmaşık verileri öğrenebilmesi için daha derin ağ yapıları (daha fazla katman ve nöron) eklenebilir. Ancak, daha derin ağ yapıları overfitting riskini artırabileceği için dikkatli olunmalıdır.

Eğitim verisini artırmak için veri augmentasyonu teknikleri (örneğin, resim döndürme, çevirme, zoom, vb.) kullanılabilir. Bu, modelin genelleme yeteneğini artırabilir ve farklı koşullara daha dayanıklı hale gelmesini sağlayabilir.

Veri setinde hatalı, eksik veya uyumsuz veriler varsa, bu verilerin temizlenmesi veya düzeltilmesi modelin başarısını artırabilir. Veri kalitesini artırmak, modelin doğruluğu üzerinde olumlu bir etki yaratacaktır.

Ağırlıkların büyüklüğünü sınırlamak için L2 regularization uygulanabilir. Bu teknik, modelin overfitting yapmasını engellemeye yardımcı olabilir ve modelin genelleme yeteneğini artırabilir.

Modelin erken aşamalarda aşırı öğrenmesini engellemek için early stopping (erken durdurma) tekniği uygulanabilir. Bu, doğrulama kaybı artmaya başladığında modelin eğitiminin sonlandırılmasını sağlayarak overfitting'i önleyebilir.

Daha gelişmiş ve güçlü modeller için Transfer Learning yöntemleri kullanılabilir. Örneğin, önceden eğitilmiş modelleri kullanarak daha az veri ile daha hızlı ve verimli bir eğitim gerçekleştirilebilir.

Daha fazla epoch sayısı ile eğitim yapmak, modelin doğruluğunu artırabilir. Ancak, aşırı öğrenme (overfitting) ile karşılaşmamak için bu sürecin dikkatlice izlenmesi gerekir. Dropout oranı ve batch normalization parametreleri (örneğin momentum, epsilon gibi)

optimize edilerek, modelin eğitim süreci hızlandırılabilir ve doğrulama doğruluğu artırılabilir. Bu, modelin daha sağlam ve genellenebilir olmasına yardımcı olabilir.

Bu iyileştirme önerileri, modelin performansını artırmak, overfitting'i engellemek ve genelleme yeteneğini güçlendirmek için kullanılabilecek farklı stratejilerdir.

Sonuçlar

Yapılan analizler doğrultusunda, modelin doğruluğu ve genelleme yeteneğini artırmak için **Adam optimizer**, **Dropout 0.2** ve **Batch Size 64** önerilmektedir. Ancak, her bir hiperparametre üzerinde yapılacak ince ayarlamalarla modelin performansı daha da geliştirilebilir.

Ek Deneyler

Yeni oluşturduğumuz model, temel bir yapay sinir ağıdır ve üç gizli katmandan oluşmaktadır. İlk katman, giriş görsellerini düzleştirerek 28x28 boyutlarındaki her bir görseli tek boyutlu bir vektöre dönüştürmektedir. Bu, görsellerin model tarafından daha kolay işlenebilmesini sağlar. Gizli katmanlar, sırasıyla 512, 256 ve 128 nöron içerir. Ayrıca her gizli katmanda **Dropout** katmanı bulunur. Dropout, aşırı öğrenmeyi (overfitting) engellemek için nöronların %20'sini rastgele kapatır, böylece modelin genellenebilirliğini artırır. Son olarak, çıktı katmanında 10 nöron bulunur ve **Softmax** aktivasyon fonksiyonu kullanılır, bu da her sınıf için olasılıkları hesaplar.

Modelin eğitimi için **sparse categorical crossentropy** kayıp fonksiyonu ve **Adam** optimizasyon algoritması kullanılmıştır. Adam optimizatörü, öğrenme oranını dinamik bir şekilde ayarlayarak daha hızlı ve etkili bir eğitim süreci sağlar. Eğitim sırasında, modelin doğruluğu **accuracy** metriği ile izlenmiştir. Model, 25 epoch boyunca eğitim verilmiş ve her epoch'ta 128 örnekle işlem yapılmıştır.

Eğitim sonunda elde edilen doğruluk sonuçları oldukça olumlu olmuştur. Eğitim doğruluğu %92.05 olarak gerçekleşirken, doğrulama doğruluğu %88.95 olmuştur. Eğitim doğruluğunun yüksek olması, modelin eğitim verisine iyi uyum sağladığını gösterirken, doğrulama doğruluğunun biraz daha düşük olması ise modelin genel genellenebilirliğinin test edildiğini ve yeni veriler üzerinde de başarılı olduğunu göstermektedir. Ancak, doğrulama doğruluğunun biraz daha düşük olması, modelin daha da geliştirilmesi gerektiğini gösterir. Bu nedenle, modelin genellenebilirliğini artırmak amacıyla veri augmentasyonu, hiperparametre optimizasyonu gibi iyileştirme tekniklerinin değerlendirilmesi faydalı olacaktır.