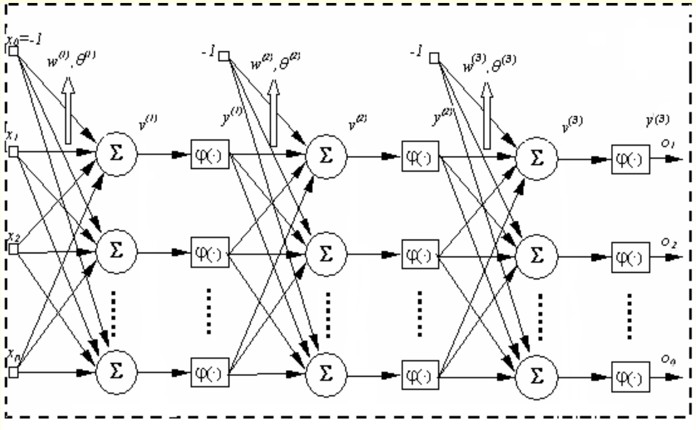
BİLECİK ŞEYH EDEBALİ ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

BİLGİSAYAR MÜHENDİSLİĞİ Anabilim Dalı



BM508 Yapay Sinir Ağları

Ders5

Doç. Dr. Cihan KARAKUZU

Momentumlu geri yayılım

- Standart geri yayılımdan farklı olarak bir/iki önceki adımdaki güncelleme miktarı da belli bir oranda (ki buna momentum denir) katılır.
- Bu strateji, eğitim sırasında oluşacak salınımları azaltılır ve yerel minimuma takılıp kalmaktan kurtarır.

$$\Delta w_{ji}(n) = \eta \delta_j y_i + \alpha \Delta w_{ji}(n-1) + \beta \Delta w_{ji}(n-2)$$

α , β : momentum katsayısı

Bu katsayılar -1< α , β <1 aralığında sezgisel (heuristic) olarak seçilir…

Esnek Yayılım (Resilient Propagation-RP) Algoritması (1)

Martin Riedmiller und Heinrich Braun: Rprop - A Fast Adaptive Learning Algorithm. Proceedings of the International Symposium on Computer and Information Science VII, 1992

M. Riedmiller & H. Braun (1993). A direct adaptive method for faster backpropagation learning: The RPROP algorithm, *Proc. Of the IEEE International Conference on Neural Networks (ICNN), San Francisco, USA, pp.586-591*.

- ***MLP' lerde, genellikle ara katmanlarda sigmoidal transfer fonksiyonları kullanılır. Sonsuz genişlikteki bir aralıkta yer alan giriş değerleri sınırlı bir aralığa sıkıştırıldığı için bu fonksiyonlar sıkıştırıcı (squashing) fonksiyonlar olarak adlandırılmaktadır.***
- ***Sigmoidal fonksiyonlar, eğimleri çok büyük giriş değerleri için sıfıra yakınsayacak şekilde karakterize edilmişlerdir. Bu ise ağırlık ve eşik değerlerin henüz optimum değerlere erişmemişken (çok katmanlı YSA' da sigmoidal transfer fonksiyonlarının kullanılmasından dolayı eğim değerinin çok yavaş değişebilme olasılığı karşısında) yüksek eğim azaltma ile öğrenmede problemlere neden olur.***

Esnek Yayılım (Resilient Propagation-RP) Algoritması (2)

- ***Bu öğrenme algoritmasının amacı kısmi türevlerin olumsuz etkilerini öğrenme sürecinden uzaklaştırmaktır. Bu algoritma Riedmiller ve Braun (1993) tarafından geliştirilmiştir. Ağırlıkların güncelleştirilme yönü için sadece türevlerin işaretleri kullanılır. Ağırlıkların güncelleştirilmesinde türev değerinin öneminin olmaması bu algoritmayı diğerlerinden ayıran en önemli özelliktir.***
- ***Bu özelliği RP' ye, hızlı çözüme ulaşma yeteneği kazandırmıştır.***
- ***Ağırlıkların değişimi, her adımda ölçüt fonksiyonu $\varepsilon(k)$' nın ağırlık değişim değeri $\Delta w_{ji}(k)$' ya göre kısmi türevinin işaretinin bulunmasıyla başarılır.***
- ***Ağırlıklardaki değişim ise sonraki yansıda verilen formülasyonla bulunur.***

Esnek Yayılım Algoritması (3)

$$\Delta_{ji}(n)=\begin{cases}\eta^{+}.\Delta_{ji}(n-1), & \dfrac{\partial\varepsilon(n)}{\partial w_{ji}(n)}.\dfrac{\partial\varepsilon(n-1)}{\partial w_{ji}(n-1)}>0\ ise\\[2ex] \eta^{-}.\Delta_{ji}(n-1), & \dfrac{\partial\varepsilon(n)}{\partial w_{ji}(n)}.\dfrac{\partial\varepsilon(n-1)}{\partial w_{ji}(n-1)}<0\ ise\\[2ex] \Delta_{ji}(n-1), & \dfrac{\partial\varepsilon(n)}{\partial w_{ji}(n)}.\dfrac{\partial\varepsilon(n-1)}{\partial w_{ji}(n-1)}=0\ ise\end{cases}$$

$$\Delta w_{ji}(n)=\begin{cases}-\Delta_{ji}(n), & \dfrac{\partial\varepsilon(n)}{\partial w_{ji}(n)}>0\ ise\\[2ex] \Delta_{ji}(n), & \dfrac{\partial\varepsilon(n)}{\partial w_{ji}(n)}<0\ ise\\[2ex] 0\quad, & \dfrac{\partial\varepsilon(n)}{\partial w_{ji}(n)}=0\ ise\end{cases}$$

$$0<\eta^{-}<1<\eta^{+}$$

$$\frac{\partial\varepsilon(n)}{\partial w_{ji}(n)}=\frac{1}{2}\sum_{p=1}^{P}\frac{\partial\varepsilon_p(n)}{\partial w_{ji}(n)}$$

Esnek Yayılım (Resilient Propagation-RP) Algoritması (4)

- ***Ardışık iki iterasyonda, ölçüt (başarım) fonksiyonu ile türev işaretleri aynı ise her bir ağırlık ve eşik değeri için güncelleme faktörü değeri, 1'den büyük bir katsayı ile çarpılır.***
- ***Eğer başarım fonksiyonu ile türev işaretleri farklı ise güncelleme faktörü değeri 1'den küçük bir katsayı faktörü kadar azaltılır.***
- ***Türev değeri sıfır ise güncelleme değeri sabit kalır.***
- ***Böylece, ağırlıklar salınım yaptığında, ağırlık değişimi azalacaktır. Eğer ağırlıklar bir kaç iterasyon boyunca aynı yönde değişim gösterirse ağırlık değişimi artar.***
- ***Bu öğrenme algoritması genel olarak standart eğim azaltma algoritmalarından çok hızlıdır. Momentumlu öğrenmeye göre çok daha az hafıza gerektirir. YIĞIN (BATCH) TİPİ EĞİTİMDE KULLANILIR.***

Delta-Bar-Delta (DBD) Algoritması (1) (R. A. Jacobs, 1988, “Increased rates of convergence trhough learning rate adaptation”, Neural Networks, 1,295-307)

- ***Çok katmanlı algılayıcılarda bağlantı ağırlıklarının yakınsama hızını arttırmak için kullanılan sezgisel bir yaklaşımdır.***
- ***Deneysel çalışmalar, ağırlık uzayının her boyutunun tüm hata yüzeyi açısından tamamen farklı olabileceğini göstermiştir. Hata yüzeyindeki değişimleri açıklamak için, özellikle, ağın her bağlantısı kendi öğrenme katsayısına sahip olmalıdır.***
- ***Her bir bağlantıya bir öğrenme katsayısı atanırsa ve bu katsayı zamanla değişirse, yakınsama zamanı azaltılır ve daha çok serbestlik derecesi sağlanmış olur.***

Delta-Bar-Delta (DBD) Algoritması (2)

- ***İleri beslemeli YSA'daki her bağlantı için en uygun öğrenme katsayı kümesini belirlemek oldukça zaman alıcı olabilir.***

- ***Eğimin geçmişteki değerlerini kullanarak, yerel hata yüzeyinin eğimini çıkarmak için sezgisellik uygulanabilir.***

- ***Bir bağlantı için ağırlık değişimlerinin işareti, birkaç ardışık zaman adımlarında sırayla değiştiğinde, bağlantı için öğrenme katsayısı azalmalıdır. Bağlantı ağırlık boyutu, hata yüzeyi ile ilgili büyük bir eğriliğe sahiptir.***

- ***Bağlantı ağırlık değişimleri bir kaç ardışık zaman adımları için aynı işarete sahip olduğundan bağlantı için öğrenme oranı arttırılmalıdır. Ağırlık boyutu hata yüzeyi ile ilgili küçük bir eğime sahiptir ve yüzey önemli bir mesafe için aynı doğrultudaki eğime göre devam eder.***

Delta-Bar-Delta (DBD) Algoritması (3)

DBD öğrenme kuralında, her bağlantı için değişken birer öğrenme oranı $\mu(n)$ atanır ve bağlantı ağırlığı **w(n + 1) =w(n) + μ(n+1)g(n)** eşitliği ile güncellenir. Yığın (batch) öğrenme modunda kullanılmalıdır.

Standart geri yayılım algoritmasında ***<u>eğim bileşeni</u>*** (gradyen) : $g(n) = \dfrac{\partial \varepsilon(n)}{\partial w_{ji}(n)}$

$$\bar{g}(n) = (1-\theta)g(n) + \theta g(n-1), \qquad 0 < \theta < 1$$

$$\mu_{ij} \rightarrow w_{ij} \text{ icin ögrenme orani}$$

$$\mu_{ij}(n+1) = \begin{cases} \mu_{ij}(n) + \kappa & , \quad \bar{g}(n-1)g(n) > 0 \\ (1-\gamma)\mu_{ij}(n) & , \quad \bar{g}(n-1)g(n) < 0 \\ \mu_{ij}(n) & , \quad \bar{g}(n-1)g(n) = 0 \end{cases}$$

$$\theta, \kappa, \gamma \quad \text{keyfi secilir}$$

Algoritmanın dezavantaları : parametrelerinin nasıl seçileceğini bilmek kolay değil, fazla hafıza elemanı kullanmayı gerektirir, online eğitimde çalışmaz/kullanılmaz…

Algoritmanın detayı Simon Haykin'in kitabının (1. baskısında) 194-197 sayfalarından öğrenilebilir. Algoritmanın genişletilmiş (Extended) versiyonu (EDBD) da vardır!