

KARADENİZ TEKNİK ÜNİVERSİTESİ  
BİLGİSAYAR MÜHENDİSLİĞİ  
YAPAY SİNİR AĞLARI ÇOKLU KATMAN  
ÖDEV RAPORU  
Orkhan ALİYEV  
338396

## ÇOKLU KATMAN

Öncelikle arayüze picturebox ekleyip **Paint event** ile koordinat sistemi çiziyoruz. Daha sonra Oluşturduğumuz altı sınıfdan birinin **radiobutton**'una tıklayıp **koordinat** sisteminde örnekleriekliyoruz. Aldığımız sınıf sayısını **numericupdown**'da da belirtiyoruz. Örnekleri alırken de **Paint event** ile örneklerin çizimini yapıyoruz. Burada **mouseclick event**'ı devreye giriyor. Koordinat sisteminde herhangi noktaya tıkladığımızda hangi **radiobutton** seçili ise o sınıfın koşulu çalışacaktır. **Noktalar** sınıfından oluşturduğumuz **referansnoktalar** nesnesinin **constructure** fonksiyonuna o noktanın **x**, **y** konumları ve sınıf **etiketi** gönderiliyor. Arayüzde arakatmandaki **nöron sayısını ve sınıf sayısını** alacağımız **numericupdown** oluşturuyoruz. Sınıf sayısını **K** değişkenine, ara katmandaki nöron sayısını **J** değişkenine atıyoruz. Girdi katmanı ve ara katman arasındaki ağırlık değerleri **V** matrisinde, ara katman ve çıktı katmanı arasındaki ağırlık değerleri **W** matrisinde tutulacaktır. Bunun için **V** matrisinin satır ve sütun değerlerinin  $[J \times (I + 1)]$  bu şekilde tanımlıyoruz. Burada **J** ara katmandaki nöron sayısı **I** ise girdi katmanındaki nöron sayısıdır. Girdi olarak koordinat sisteminde tıklanan noktanın **x** ve **y** koordinatları verileceğinden **I** değeri **2** olarak belirlenir. **V** ve **W** matrislerinin ağırlık değerleri arayüzde “**Verileri dosyadan al**” **checkBox**'ına tıklanmışsa dosyadan, yok eğer tıklanmamışsa rastgele olarak ağırlıklar setlenir. Örnek alınan noktalar **normalize** edilir. Her örnek **x**, **y** ve **bias** katsayısı olarak  $[3 \times 1]$  lik **z** matrisine atanır. **Bias** katsayısı 1 alınır.

$$y = \Gamma[Vz] \quad (1)$$

**V** matrisinin **z** matrisi ile çarpımından çıkan **net** değerini **sigmoid** fonksiyonuna verip (Eşitlik 1), çıktı olarak  $[J \times 1]$  lik matris **y**'i elde ediyoruz. Elde edilen matrise **bias** katsayısını ekliyoruz. Sonuçta  $[(J+1) \times 1]$  lik matris oluşturuyoruz.

$$o = \Gamma[Wy] \quad (2)$$

**W** matrisinin **y** matrisi ile çarpımından çıkan **net** değerini **sigmoid** fonksiyonuna verip (Eşitlik 2), çıktı olarak  $[K \times 1]$  lik matris **o**'u elde ediyoruz. Her örnek için beklenen değerleri oluşturduğumuz  $[K \times 1]$  lik **d** matrisine ekliyoruz. İlk aşamada

$$E \leftarrow E + \frac{1}{2} \|d - o\|^2 \quad (3)$$

Eşitlik 3'deki formül ile **döngü hata değerini** hesaplıyoruz.

$$\delta_o = \frac{1}{2} [(d_k - o_k)(1 - o_k^2)]^* \quad (4)$$

Eşitlik 4'deki formül üzerinden  $\delta_o$  matrisini hesaplıyoruz.

$$W \leftarrow W + \eta \delta_o y^t \quad (5)$$

Eşitlik 5'deki formül üzerinden **W** ağırlık değerlerini güncelliyoruz.

$$f'_y = \frac{1}{2} [1 - y_j^2] \quad (6)$$

Eşitlik 6'daki formül üzerinden  $f'_y$  değerini hesaplıyoruz.

$$\delta_y = w_j^t \delta_o f'_y, \quad (7)$$

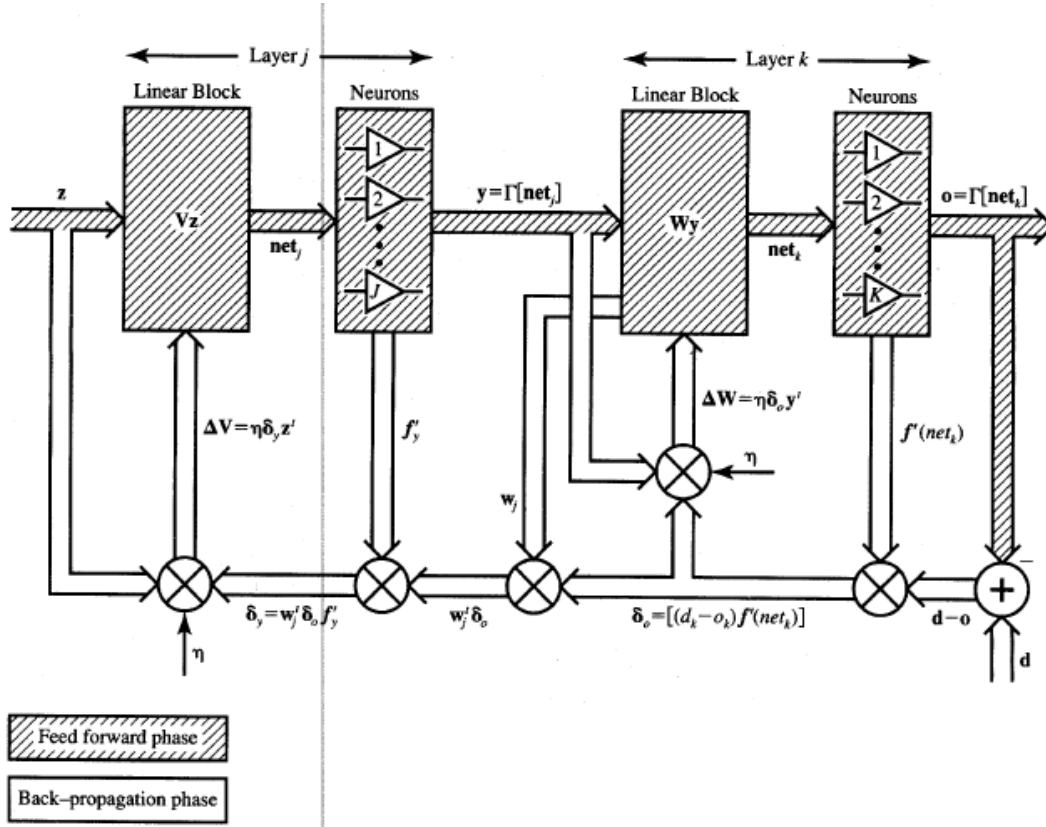
Bulunan  $f'_y$  değerini Eşitli 7'deki formülde yerine yazarak  $\delta_y$  matrisini hesaplıyoruz.

$$V \leftarrow V + \eta \delta_y z' \quad (8)$$

Hesaplanan  $\delta_y$  matrisi  $V$  ağırlığını güncellemede kullanılır (Eşitlik 8).

$\mu$  değeri hesaplama işlemlerinde 1 olarak alınmıştır.

Bu akış tüm örnekler için uygulanır. Örnekler bittiğinde toplam  $E$  değeri kontrol edilir: belirlediğimiz hata payından küçük ise işlem tamamlanır, büyük ise  $E$  değeri sıfıra eşitlenir ve ilk adımdan itibaren tüm örnekler için akış tekrar olarak uygulanır.



İşlem tamamlandığında **paint event**'inde Eşitlik 1 ve Eşitlik 2 uygulanması sonucu ortaya çıkan  $o$  matrisinde maksimum değere sahip sınıfı belirleyip **PictureBox**'da renklendirme işlemini gerçekleştiriyoruz. Renklendirme işlemi **PictureBox**'ın satır ve sütununda üçer piksel atlanarak ilerliyor. Bunun sebebi hız kazanmak.

Sonuç olarak bir problemin ara katman koyularak modellenmesini ve katmanlı mimari ile nasıl çözüm üretilceğini anladık.