

**KARADENİZ TEKNİK ÜNİVERSİTESİ BİLGİSAYAR MÜHENDİSLİĞİ**  
**YAPAY SİNİR AĞLARI ÖDEV RAPORU**

**Orkhan ALİYEV**

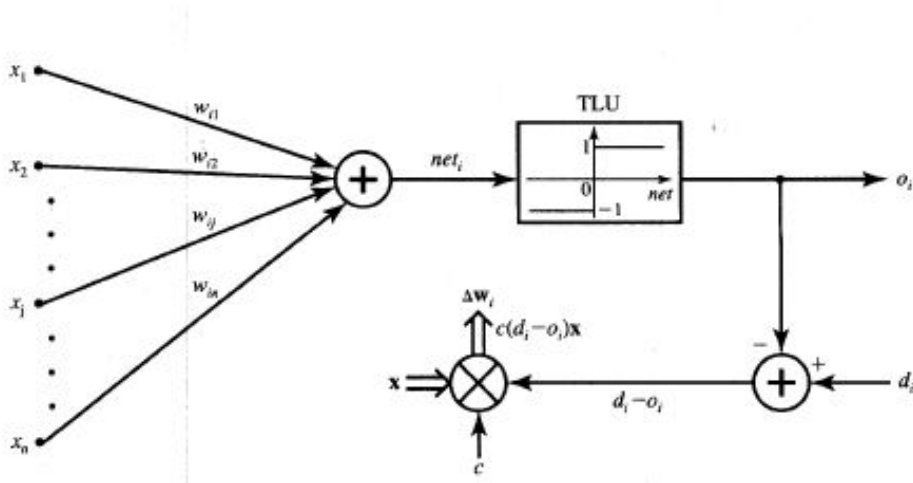
**338396**

# Tek Katman İki Sınıf

Öncelikle arayüze **pictureBox** ekleyip **Paint event** ile koordinat sistemi çiziyoruz. Daha sonra oluşturduğumuz iki sınıftan birinin **radioButton**'una tıklayıp koordinat sisteminde örnekleri alıyoruz. Örnekleri alırken de **Paint event**'i ile örneklerin çizimini yapıyoruz. Burada **MouseClicked event**'i devreye giriyor. Bu fonksiyon sayesinde koordinat sisteminde tıklanan noktanın **x, y** konumlarını ve **sınıf etiketini** oluşturarak örnekler dizisine ekliyoruz. Bu örnekleri **pointer** ile bellekte tutuyoruz. Bu şekilde örnekleri aldıktan sonra bu örnekleri ayrıştıracak **doğruyu** çizme aşamasında ilk olarak **perceptron öğrenme** kuralını uyguluyoruz. Öncelikle **ağırlığı rastgele** atıyoruz. Daha sonra **perceptron öğrenme** kuralına göre gerekiyorsa ağırlıkları güncelliyoruz. Bunun için **ağırlık güncelleme** fonksiyonumuz var. **Güncellenen ağırlıklar** diğer adımlarda yukarıdaki koşulu sağlıyorsa **döngü sonlanır** ve **doğru çizilir**. Bunun için de **doğru çizme** fonksiyonumuz var.

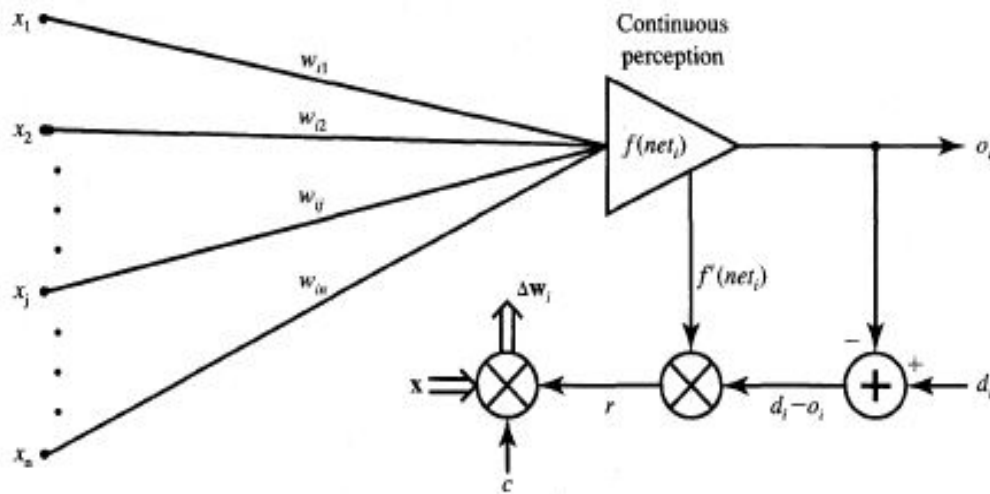
**Delta öğrenme** yaptığımız zaman ise **normalize** edilmiş değerler üzerinde işlem yapmamız gerekiyor. **Normalize işlemi** yaptıktan sonra örneklerimizi **ayrı bir dizide** tutuyoruz ve **delta öğrenme** kuralını bu örneklerle uyguluyoruz. **Normalize** sonucu değerler **0** ile **1** arasına taşınıyor. Bu örnekleri görebilmek için **zoom değeri** setliyoruz ve **trackBar** üzerinden bu değeri azaltıp arttırabiliyoruz. Bu işlem sonucu ise örnekleri koordinat sisteminde büyütüp küçültebiliriz.

**Perceptron öğrenme** kuralını uygularken aşağıdaki şema mantığına göre kodlama yaptım.



Örnekler ağı giriyor ve net değerleri hesaplanıyor daha sonra  $f(net)$  hesaplanıyor ve output üretiliyor. Üretilen çıktı ile beklenen değer arasındaki fark 0'a eşit ise ağırlık güncelleniyor ve diğer örnek için bu iş aynı şekilde devam ediyor. Yok eğer tam tersi ise ağırlıklar güncellenmeden diğer örneğe geçiliyor.

**Delta öğrenme** kuralını uygularken aşağıdaki şema mantığına göre kodlama yaptım.



Örnekler ağa giriyor ve net değerleri hesaplanıyor daha sonra  $f(net)$  hesaplanıyor ve output üretiliyor. Üretilen çıktı ile beklenen değer arasındaki fark belirlediğimiz hata payından büyük ise ağırlık güncelleniyor ve diğer örnek için bu iş aynı şekilde devam ediyor. Yok eğer tam tersi ise ağırlıklar güncellenmeden diğer örneğe geçiliyor.

## Tek Katman Çok Sınıf

Öncelikle arayüze **pictureBox** ekleyip **Paint event** ile koordinat sistemi çiziyoruz. Daha sonra oluşturduğumuz altı sınıftan birinin **radioButton**'una tıklayıp koordinat sisteminde örnekleri alıyoruz. Aldığımız sınıf sayısını **numericUpDown**'da da belirtiyoruz. Örnekleri alırken de **Paint event**'ı ile örneklerin çizimini yapıyoruz. Burada **MouseClicked 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. Bu şekilde **örnekleri** aldıktan sonra bu örnekleri **ayrıştırarak doğruları** çizme aşamasında ilk olarak **perceptron öğrenme** kuralını uyguluyoruz. **Ağırlık matrisimizin** boyutunu da **sınıf boyutu değerine** göre setliyoruz ve **ağırlıkları rastgele** atıyoruz. Daha sonra **nöronlara** göre **beklenen değer matrislerini** setliyoruz. Devamında **perceptron öğrenme** kuralına göre öğrenmeyi gerçekleştiriyoruz. Matematiksel işlemlerin hepsini **Matris sınıfına** göre yapıyoruz. Orada tüm **operator overloading** işlemleri gerçekleştirilmiştir. **Tek katman iki sınıf** örneğinde olduğu gibi **delta öğrenme** kuralına göre öğrenme yaptığımız zaman **normalize checkBox**'ını işaretliyoruz. **Önceki örnekte** olduğu gibi burada da değerler **sıfır ila bir** arasında yerleştiriliyor. Fakat burada zoom değeri statik olarak setlenmiştir. **Her iki uygulamada** da **Temizle butonu** ile arayüzü temizleyip başlangıç değerlere setleme yapıyoruz.