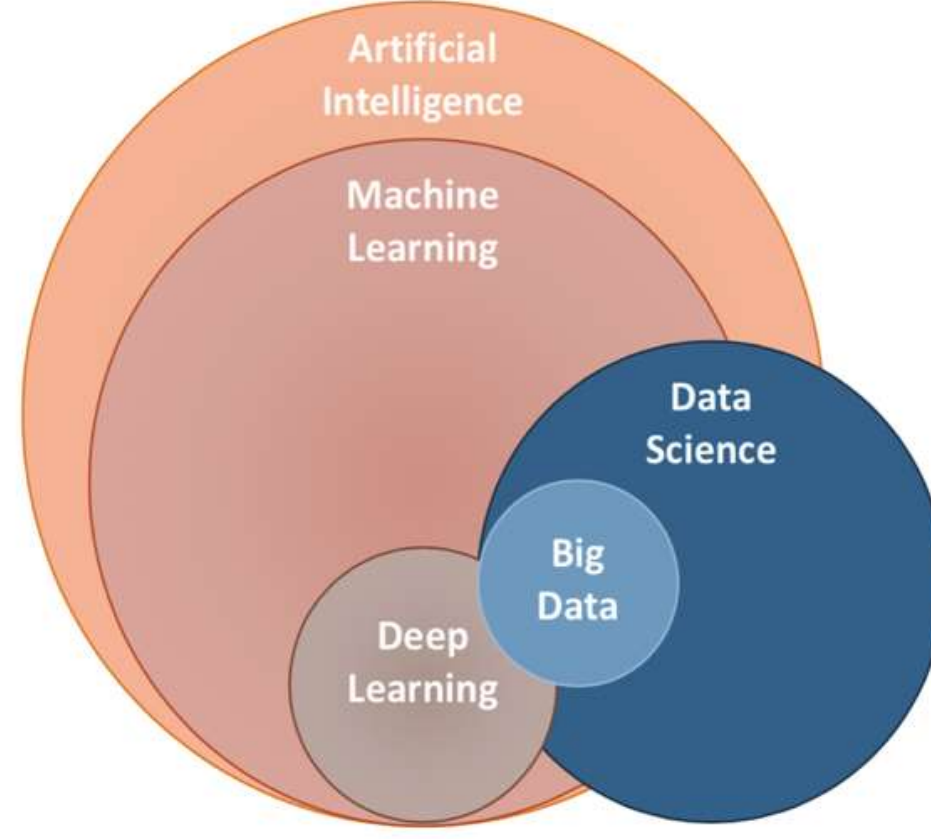


ÖZET

Yapay Zeka; adı son yıllarda çok popülerleşen, hakkında yüzlerce kitap yazılan, binlerce akademik yayın ve araştırmaya konu olan, sinema sektörünün bilim-kurgu senaryolarında sıkça esinlendiği ve bunu yaparken yapay zeka kavramı ile ilgili tek taraflı bir bakış açısı sunup yanılgılara yol açabildiği bir yaklaşımdır. Yapay zekanın gelişmesiyle beraber yapay zeka uygulamaları da gelişmiştir. Bu uygulamalardan biri olan makine öğreniminin öğrenme yaklaşımlarından biri de memelilerin sinir sisteminden esinlenilerek tasarlanan “Yapay Sinir Ağları”dır. Bu çalışmada Yapay Zeka ve onun bir uygulaması olan Makine Öğrenimine değinildikten sonra Yapay Sinir Ağları üzerinde durulacak ve MATLAB’de Yapay Sinir Ağları ile Sınıflandırma Modellemesi örneği verilecektir.



GİRİŞ

Yapay zeka kavramı ilk olarak Alan Turing tarafından “Makineler düşünebilir mi?” sorusuyla ortaya çıkmıştır. Yapay zekâ, bir bilgisayarın ya da bilgisayar destekli bir makinenin, genellikle insana özgü nitelikler, çözüm yolu bulma, anlama, bir anlam çıkartma, genelleme ve geçmişteki deneyimlerinden öğrenme gibi yüksek mantık süreçlere ilişkin görevleri yerine getirme yeteneği olarak bilim dünyasında tanımlanmıştır.

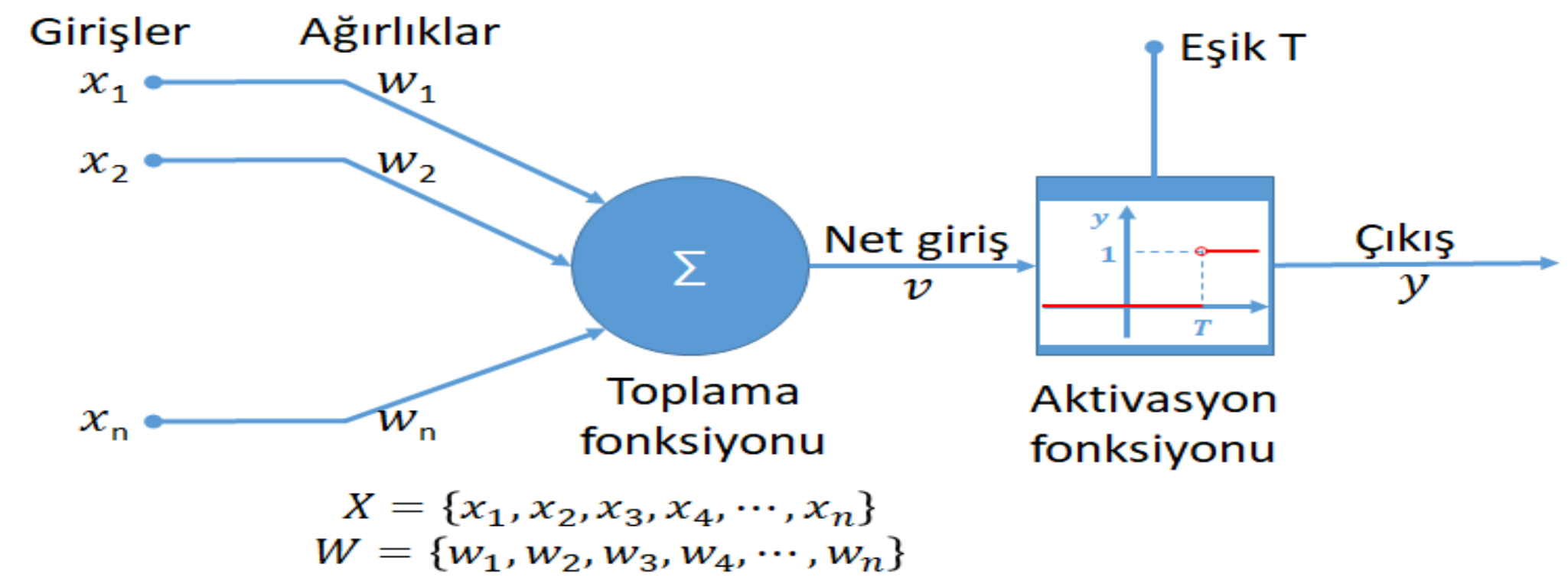
Yapay zekanın insanlar gibi öğrenmesi prensibini esas alan **Makine öğrenimi (ML – Machine Learning)**, yazılım programlarının açık bir şekilde programlanmadan sonuçları tahmin etmede daha doğru olmasını sağlayan bir algoritma kategorisidir. *Makine öğrenmesinin temel dayanağı*, giriş verisini alabilen algoritmalar oluşturmak ve çıktılarını yeni veriler ortaya çıktıkça güncellerken bir çıktıyı tahmin etmek için istatistiksel analiz kullanmaktır.

Yapay Sinir Ağları ise makine öğrenimi yaklaşımlarından biridir. Makine öğreniminin öğrenme çeşitlerinden olan “derin öğrenme” yapay sinir ağları ile yapılır.

YAPAY SİNİR AĞLARI (YSA)

Yapay sinir ağları (YSA), insan beyninin bilgi işleme tekniğinden esinlenilerek geliştirilmiş bir bilgi işlem teknolojisidir. YSA ile basit biyolojik sinir sisteminin çalışma prensibi taklit edilir. YSA'nın öğrenme özelliği sayesinde geleneksel teknikler için çok karmaşık kalan problemlere çözüm sağlanabilmektedir. Yine öğrenme yeteneği sayesinde, bilinen örnekleri kullanarak daha önce karşılaşılmamış durumlarda genelleme yapabilmektedir.

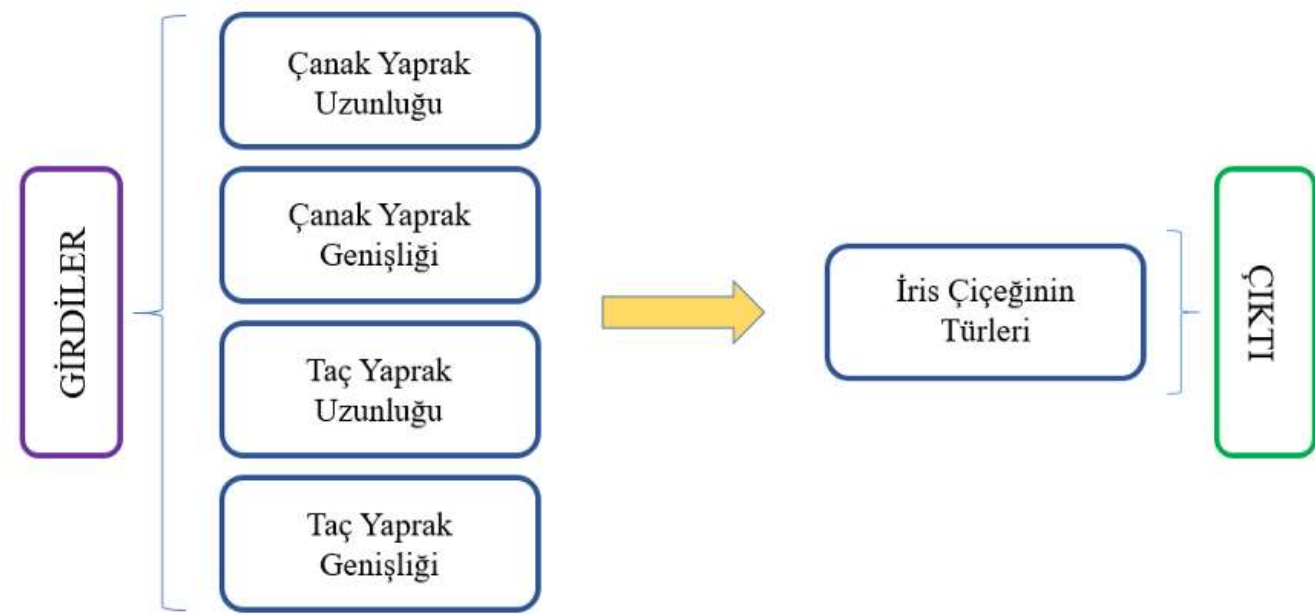
Yapay sinir ağlarındaki nöronlar, genellikle katmanlar hâlinde düzenlenir. Her katmandaki nöronlar kendinden bir önceki ve bir sonraki katmandaki nöronlarla bağlantılıdır. En baştaki katman girdi katmanı, en sondaki katman ise çıktı katmanı olarak adlandırılır. Gizli katmanlardaki nöronlar genellikle bir önceki katmandan aldıkları bilgiyi işler ve bir sonraki katmana iletir.



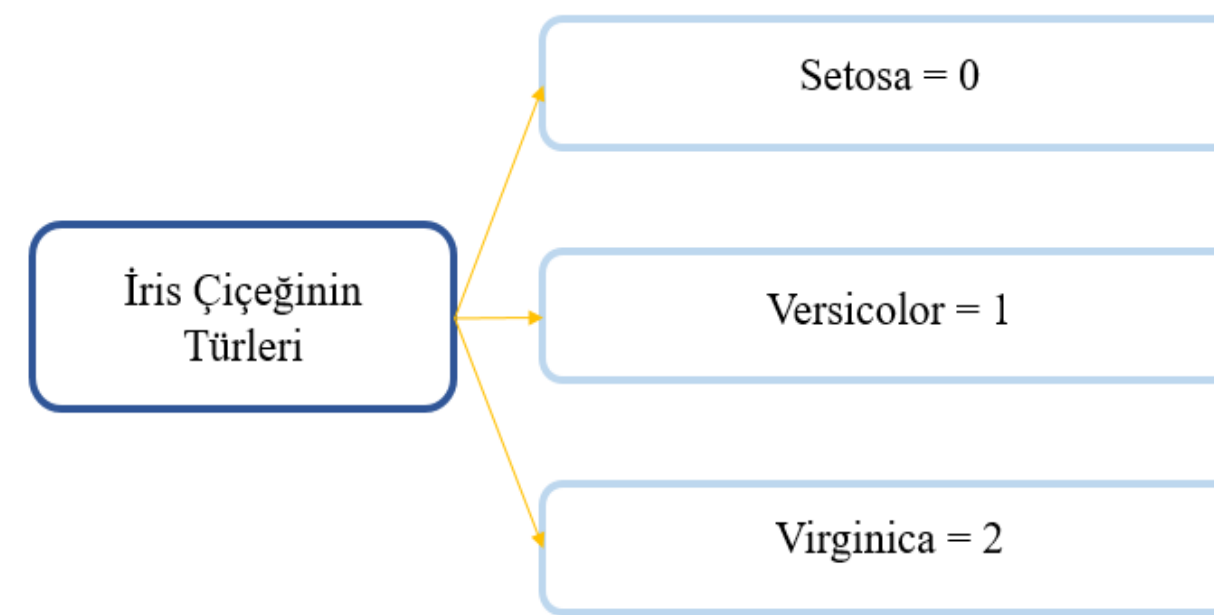
MATLAB ÜZERİNDE YAPAY SİNİR AĞI İLE SINIFLANDIRMA ÖRNEĞİ VE MODELLEMESİ

Bu uygulamada öncelikle preprocessing işlemleri için kod yazılmıştır. Daha sonrasında ağı oluşturmak için **ntool** isimli araç kullanılmıştır. Bu çalışmada yapay zeka uygulamalarında örnek olarak sık sık kullanılan ve yüksek performans veren bir veri seti olan **İris çiçeği veri seti** kullanılmıştır.

Kullanılan veri setinde iris çiçeği toplam 3 türe ayrılarak sınıflandırılmıştır. Bu türlerin sınıflandırılmasında ise 4 tane ölçüt kullanılmıştır. Son olarak bu uygulamada 150 tane ölçüm yapılmıştır ve bu ölçümler cm cinsinden kaydedilmiştir.



String veriler kabul edilmediği için öncelikle bu verilerin nümerik ifadelerle çevrilmesi gereklidir. Bu çalışmada da tür çıktıları için nümerik değerler atanmıştır:



Normalizasyon için aşağıdaki fonksiyon girilir:

```
>> normalized= normalize(iris, 'range')
```

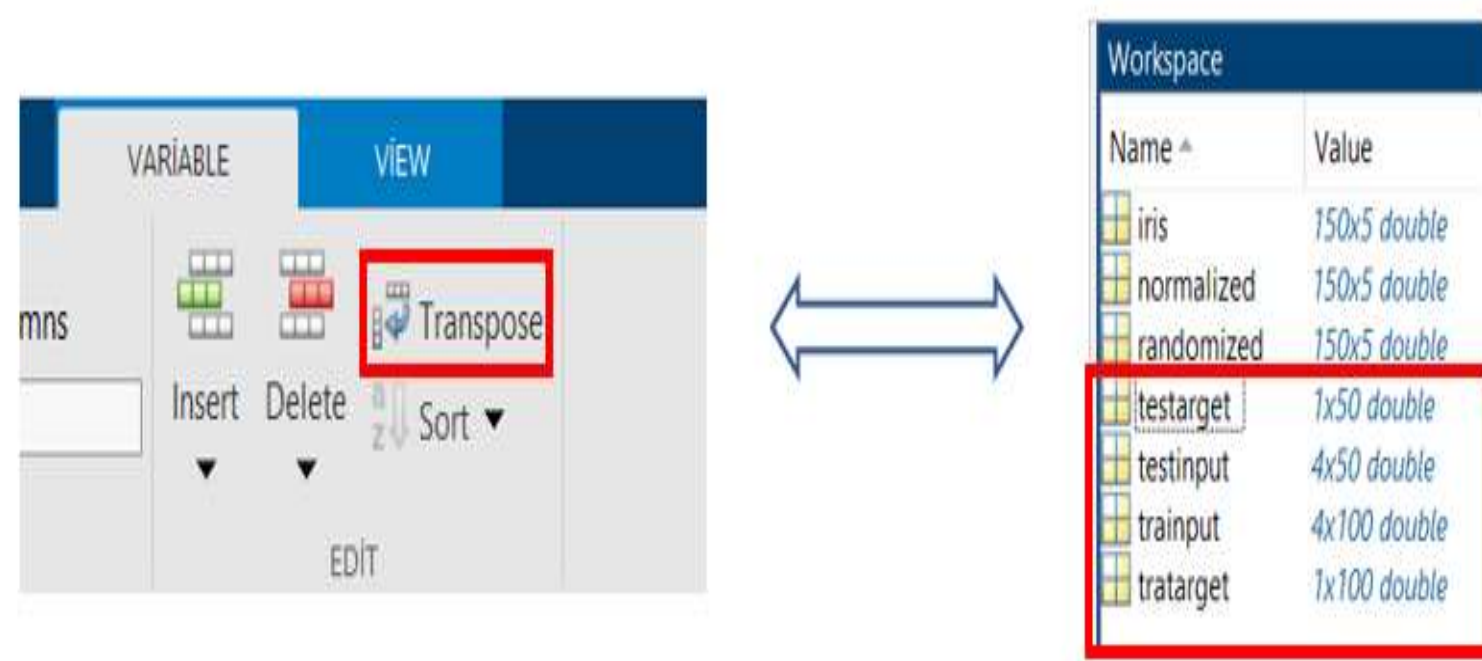
Normalizasyon işleminin amacı verileri belirli bir aralığa sıkıştırıp, sistemin performansını arttırmaktır. İşlemin sonunda veriler 0-1 aralığına sıkıştırılmıştır.

Ardından verileri randomize etmek için aşağıdaki fonksiyon girilir:

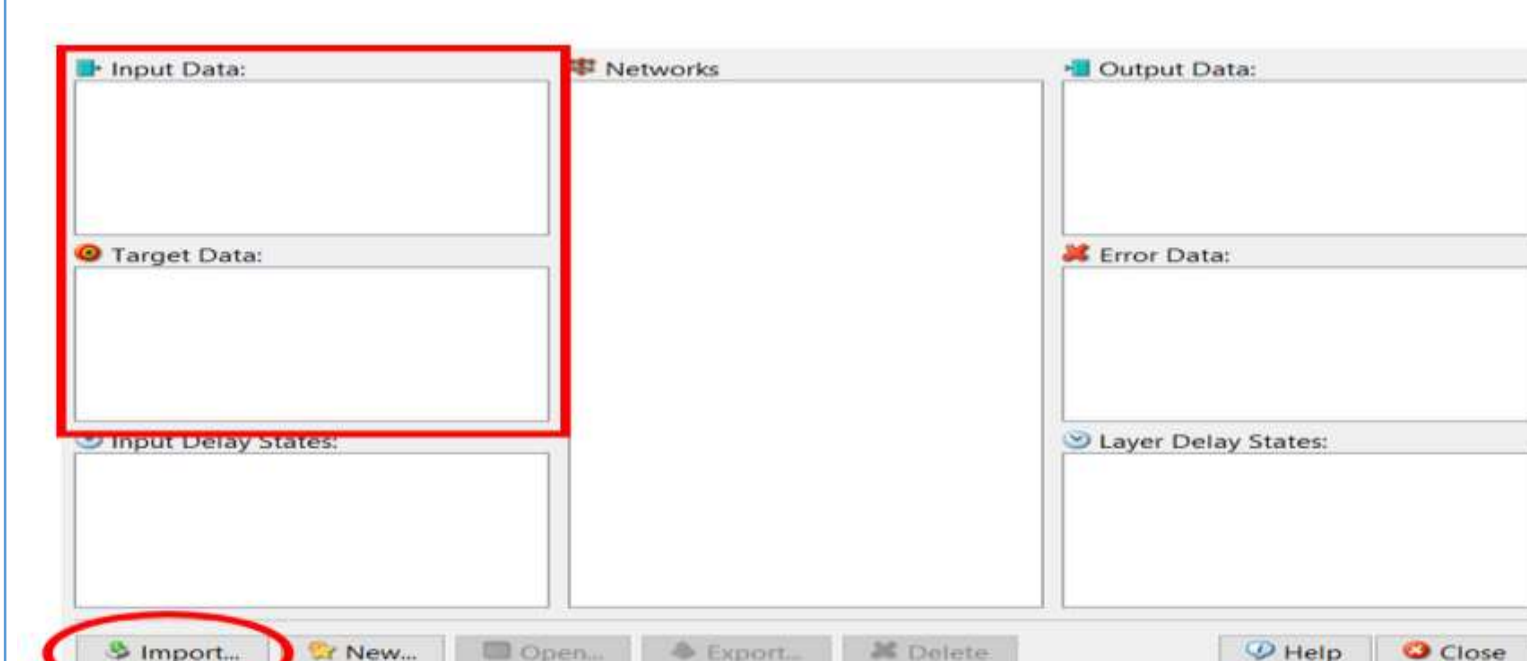
```
>> randomized=normalized(randperm(size(normalized,1)),:)
```

Bu işlemin amacı sıralı olan verileri karıştırıp, daha sağlıklı bir öğrenme ağı oluşturmak. Eğer veriler randomize edilmezse ağı eğitimi esnasında eksik öğrenme oluşabilir ve test bölümünde düşük performanslı bir sonuç elde edilir.

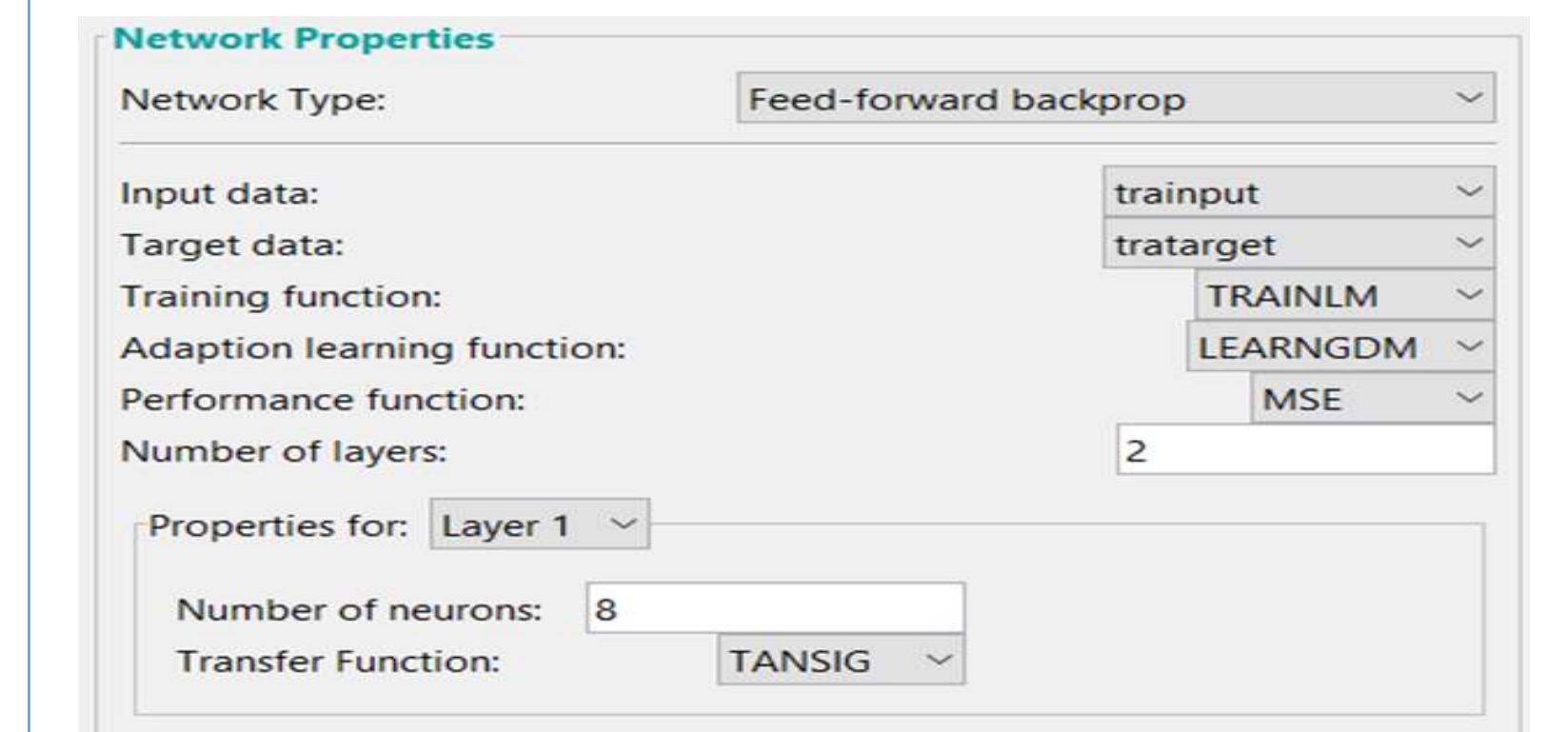
Eğitim işlemine başlamadan önce veri setinin hem ‘training’ ve ‘test’ olarak hem ‘input’ ve ‘target’ olarak bölünmesi gerekir. Training kısmında 100, test kısmında ise 50 adet satır olacak şekilde bu ayırım yapılmıştır. Ayırım işleminin ardından 4 adet veri setine sahip olunur. Son olarak bölünen tüm verilerin transpoznesinin alınması gereklidir.



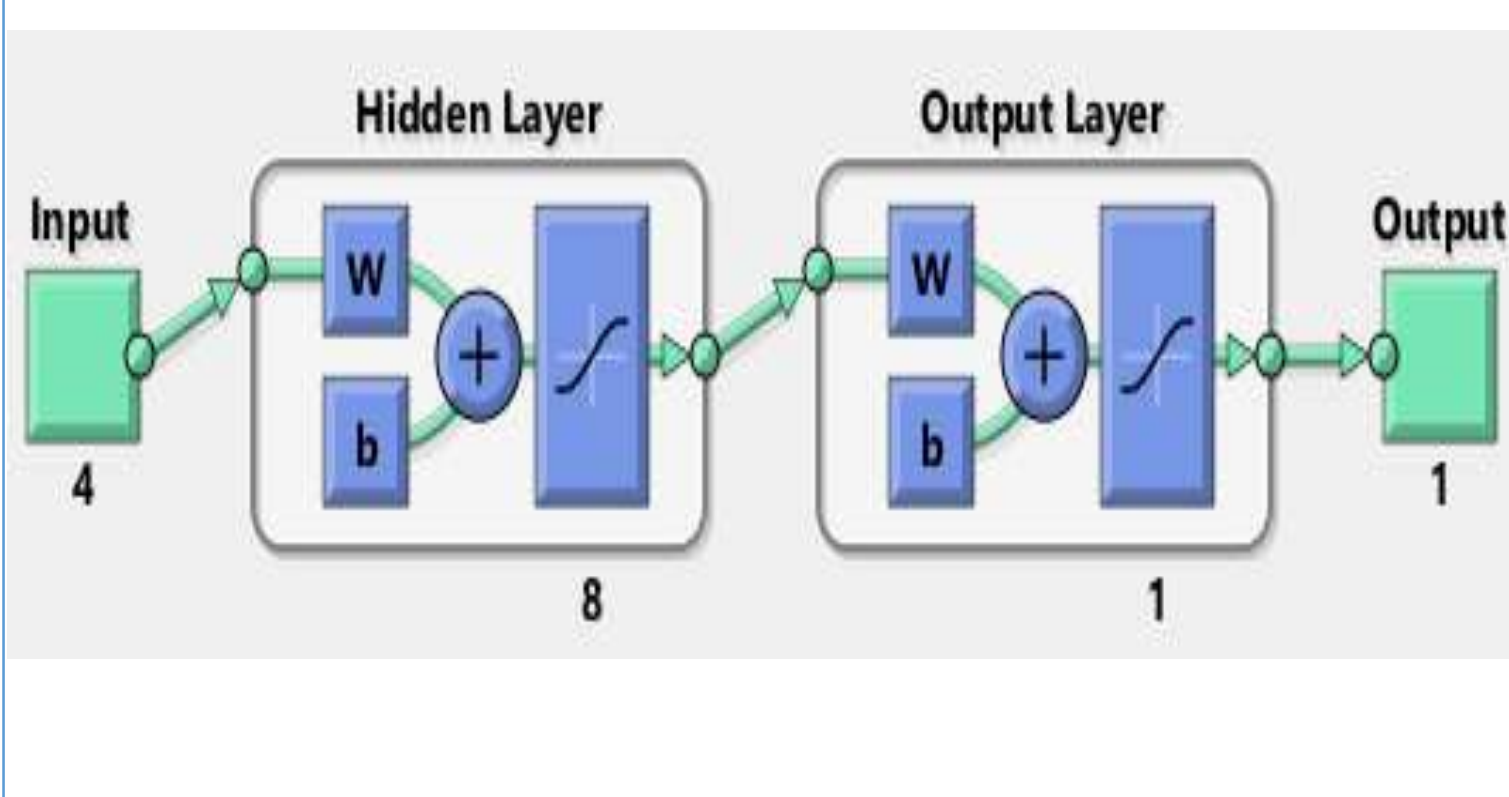
Preprocessing aşaması tamamlanmıştır. Artık oluşturulacak ağı eğitim kısmına geçilebilir. Nntool’u açmak için, command window penceresine **ntool** yazılır. Ardından yapılması gereken işlem bölünen veri setini nntool’a yüklemek olacaktır. Aşağıda görülen input data bölümüne train-input ve test-input veri kümelerini; target data bölümüne ise train-target ve test-target verileri eklenir.



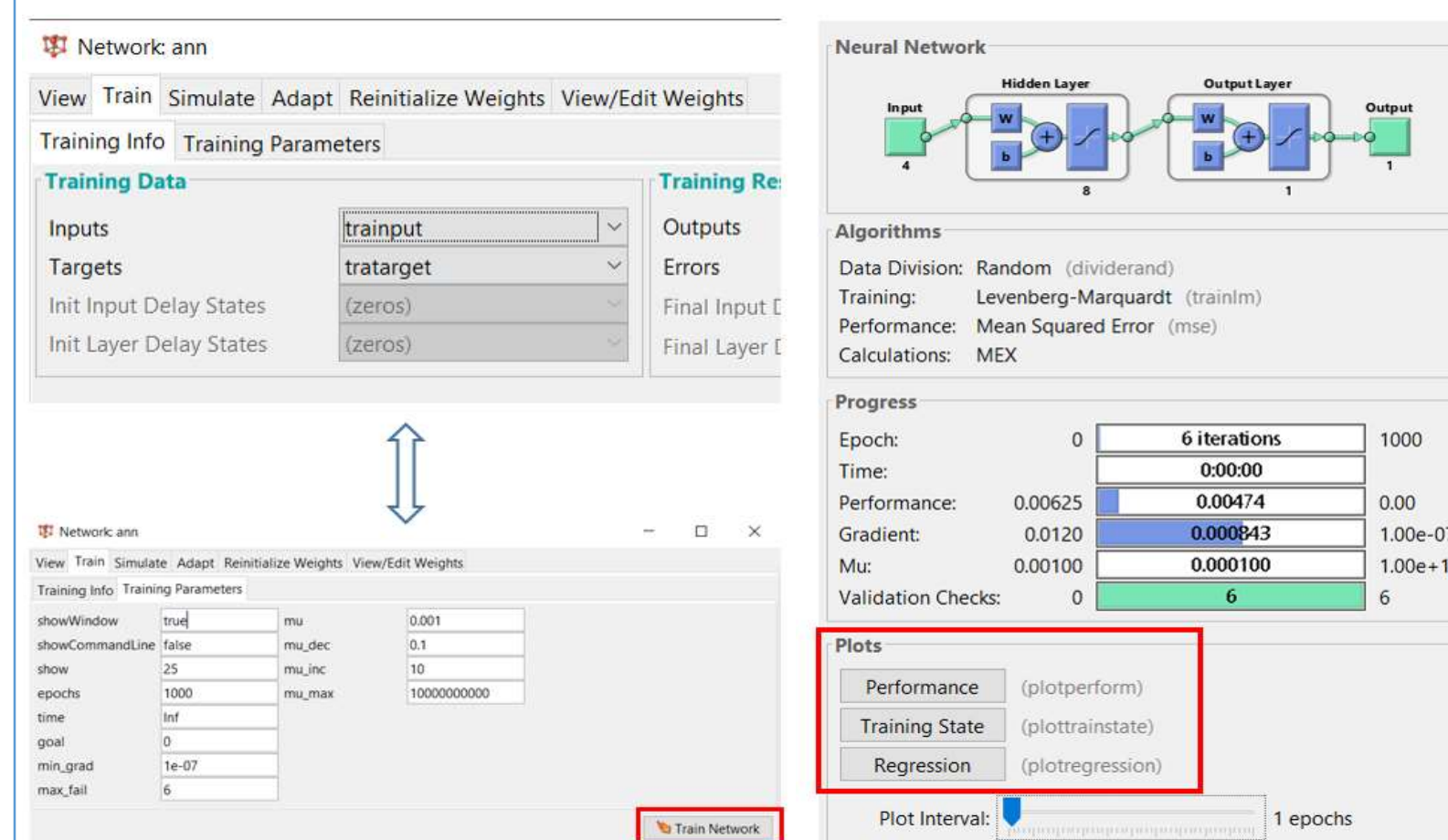
Yükleme işlemi tamamlandıktan sonra oluşturulacak ağı eğitim kısmı için öğrenme algoritması, performans metriği, katman sayısı, nöron sayısı ve transfer fonksiyonu gibi parametreler girilir.



Oluşturulmak istenen ağı topolojisi aşağıda verilmiştir. Topolojide 4 girdi, 1 gizli katman ve 1 çıktı bulunmaktadır. Gizli katmanda 8 nöron bulunmaktadır. Kullanılan nöron sayısı bu veri seti üzerinde optimum sonucu verdiğinden bu sayı kullanılmıştır. Nöron sayısı deneme yoluyla artırılıp azaltılarak optimum sonuca ulaşılır.



Artık ağı eğitimi aşamasına geçilmeye hazırdır. Oluşturulan ağı girildiğinde soldaki gibi bir pencereyle karşılaşılır. Bu bölümde son parametreler girilip ağı eğitimi başlatılır. Eğitim tamamlandıktan sonra elde edilen performans verileri ve regresyon grafikleri gibi verilere ise sağda gösterilen bölümden ulaşılabilir.



Son olarak performans sonucunu görebilmek için aşağıdaki karışıklık matrisi (confusion matrix) kullanılır.

