



T.C.
KTO Karatay Üniversitesi
Mühendislik Fakültesi
Bilgisayar Mühendisliği Bölümü

CE452 - Yapay Sinir Ağlarına Giriş
YSA ile Hava Tahmini Uygulama Raporu

HAZIRLAYAN
21502774 - Mehmet KALAYCI

DERS DANIŞMANI
Prof. Dr. Novruz ALLAHVERDİ

KONYA
Ocak, 2018

İçindekiler

1. GİRİŞ	2
2. Yapay Sinir Ağı, Özellikleri ve Kullanım Alanları [1]	2
2.1. Yapay Sinir Ağı Nedir?	2
2.2. Yapay Sinir Ağı Özellikleri	3
2.3. Yapay Sinir Ağı Kullanım Alanları	3
3. Hava Tahmin Uygulaması	4
3.1. Neural Network Fitting Tool	4
3.2. Neural Network Fitting Tool ile Hava Tahmin Uygulaması	5
4. SONUÇ	9
EK-1	10
EK-2	14
KAYNAKLAR	17

1. GİRİŞ

Zaman değişkeniyle ilişkili bir değişken hakkında, elde edilen gözlem değerlerini zamana göre sıralanmış olarak gösteren serilere, zaman serisi denir. Zaman serisi üzerinde yapay sinir ağları ile ileriye doğru tahmin yürütmektir. Zaman serisinde bir değer kendinden önce gelen değerlerle ilişkili ve dolayısıyla onlarla tahmin edilebilirdir.

Bu çalışmada 15 Haziran 2016 ve 22 Eylül 2016 tarihleri arasındaki nem oranı (%), görüş mesafesi (km), basınç (mb), çiğ noktası (°C), sıcaklık (°C) bilgilerini içeren bir veri seti oluşturularak yapay sinir ağı kullanılarak geleceğe yönelik hava tahmini uygulaması MATLAB programlama dili geliştirilmiştir.

2. Yapay Sinir Ağı, Özellikleri ve Kullanım Alanları [1]

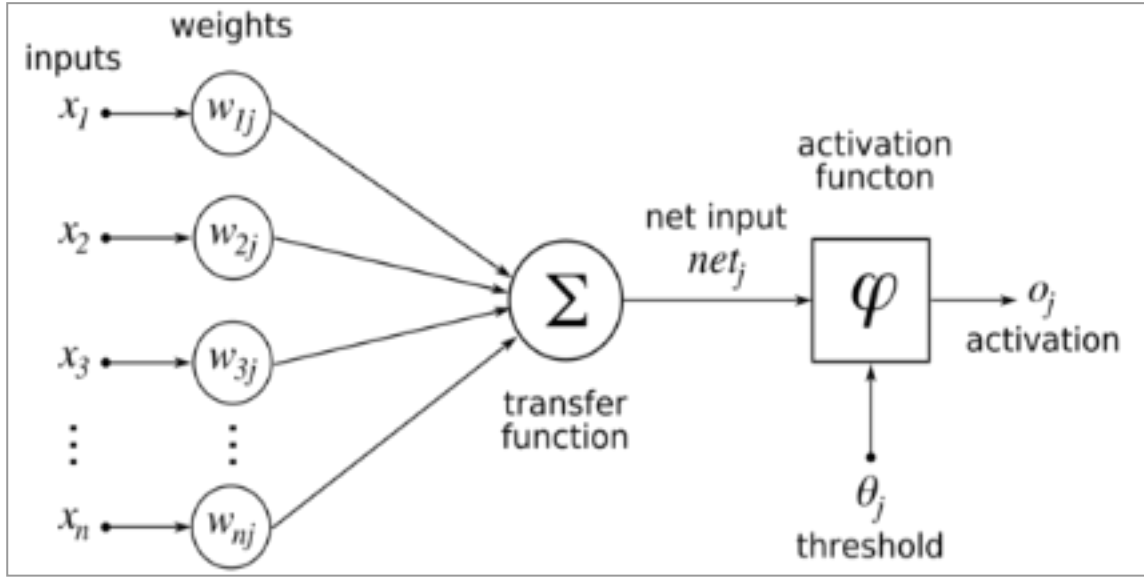
2.1. Yapay Sinir Ağı Nedir?

YSA, insan beyninin çalışma mekanizmasını taklit ederek beynin öğrenme, hatırlama genelleme yapma yolu ile yeni bilgiler türetebilme gibi temel işlevlerini gerçekleştirmek üzere geliştirilen mantıksal yazılımlardır. YSA biyolojik sinir ağlarını taklit eden sentetik yapılardır.

İlk yapay sinir ağı modeli 1943 yılında bir sinir hekimi olan Warren McCulloch ve bir matematikçi olan Walter Pitts tarafından Sinir Aktivitesinde Düşüncelere Ait Bir Mantıksal Hesap (A Logical Calculus of Ideas Immanent in Nervous Activity) başlıklı makale ile ortaya çıkarılmıştır. Sinir sistemi elemanlarının, yapay sinir ağı modelindeki terminolojisi tablo 3.1 de belirtilmiştir. Şekil 3.1 de yapay sinir ağı modeli gösterilmiştir.

Tablo 2.1: Sinir sistemi elemanlarının, Yapay Sinir Ağı modelindeki terminolojisi

Sinir Sistemi	Yapay Sinir Ağı
Nöron	İşlem Elemanı
Dentrit	Toplama Fonksiyonu
Hücre Gövdesi	Aktivasyon Fonksiyonu
Akson	Eleman Çıkışı
Sinaps	Ağırlıklar



Şekil 2.1: Yapay sinir ağı modeli

2.2. Yapay Sinir Ağı Özellikleri

Yapay sinir ağlarının en temel özellikleri aşağıda belirtilmiştir.

- Doğrusal Olmama
- Paralel Çalışma
- Öğrenme
- Genelleme
- Hata Toleransı ve Esneklik
- Eksik Verilerle Çalışma
- Çok Sayıda Değişken ve Parametre Kullanma
- Uyarlanabilirlik

2.3. Yapay Sinir Ağı Kullanım Alanları

Yapay sinir ağları üzerine çalışmalar iki kısım üzerine yoğunlaşmıştır. Bunlar ileri beslemeli çok katmanlı ağlar ve Hopfield ağlarıdır. Günümüzde kullanıldığı alanlar şöyledir:

- Denetim
- Sistem Modelleme
- Ses Tanıma
- El Yazısı Tanıma
- Parmak İzi Tanıma
- Fizyolojik işaretleri tanıma, izleme ve yorumlama

- Meteorolojik yorumlama
- Elektrik işareti tanıma
- Otomatik araç denetimi

Yapay sinir ağlarının başlıca temel uygulama alanları sınıflandırma, tahmin ve modelleme olarak ele alınabilir.

Sınıflandırma: Tıbbi teşhis, imza tetkikleri, ses tanıma, şekil tanıma, hücre tiplerinin sınıflandırılması, mikroplar, modeller vb. sınıf çeşitleri.

Tahmin: İleriki satışlar, enerji ihtiyacı, üretim ihtiyacı, hava tahminleri gibi ileriye yönelik tahminler.

Modelleme: Kimyasal yapılar, işlem kontrolü, dinamik sistemler, robot kontrolü vb. uygulamalar.

3. Hava Tahmin Uygulaması

Bu çalışmada 15 Haziran 2016 ve 22 Eylül 2016 tarihleri arasındaki nem oranı (%), görüş mesafesi (km), basınç (mb), çiğ noktası (°C), sıcaklık (°C) bilgilerini içeren bir veri seti oluşturularak yapay sinir ağı kullanılarak geleceğe yönelik hava tahmini uygulaması MATLAB programlama diliyle geliştirilmiştir.

Projede kullanılacak veriler MS Excel programı kullanılarak veriler Normalize Değer= $(X - X_{min}) / X_{max} - X_{min}$ formülü kullanılarak normalize edilmiş, eğitim için %15 'i, test için % 10 'u kullanılmış doğrulama verileri olarak gruplanmıştır. Bu verilerden nem oranı (%), görüş mesafesi (km), basınç (mb), çiğ noktası (°C) değerleri ağı eğitilmesi için kullanılmış, sıcaklık (°C) bilgisi ise hedef veri olarak kullanılmıştır. EK-1 de normalize edilmiş hava durumu verileri verilmiştir.

Uygulama için hazırlanan veriler MATLAB programlama dili ve neural network fitting aracı kullanılarak, hava tahmini için yapay sinir ağı modellenmiştir. Yapay sinir ağı olarak oluşturulan MATLAB kodu EK-2 de verilmiştir.

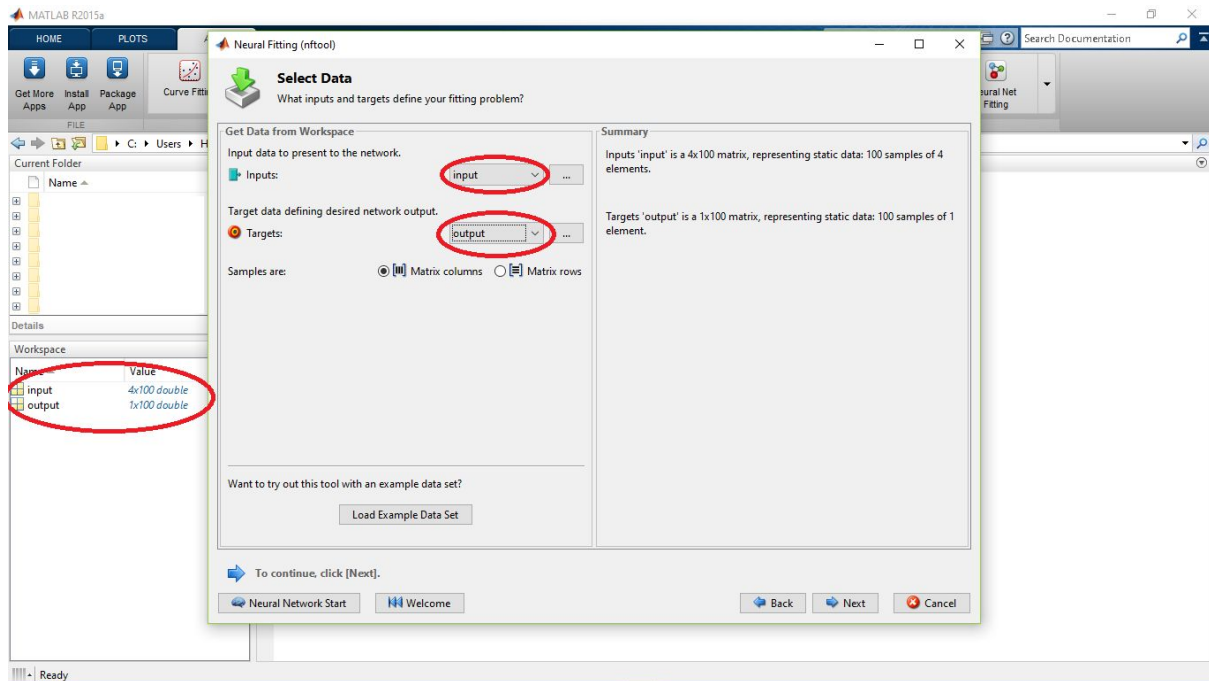
3.1. Neural Network Fitting Tool

Neural fitting uygulaması, verileri seçmenize, bir ağ oluşturmanıza ve eğitmenize ve ortalama karesel hata ve regresyon analizini kullanarak performansını değerlendirmenize yardımcı olacaktır.

Bu projede neural network fitting tool aracı kullanılmıştır. Aracı kullanabilmek için arayüz kullanılabileceği gibi, nftool komutu ile de bu araca erişim sağlanabilir.

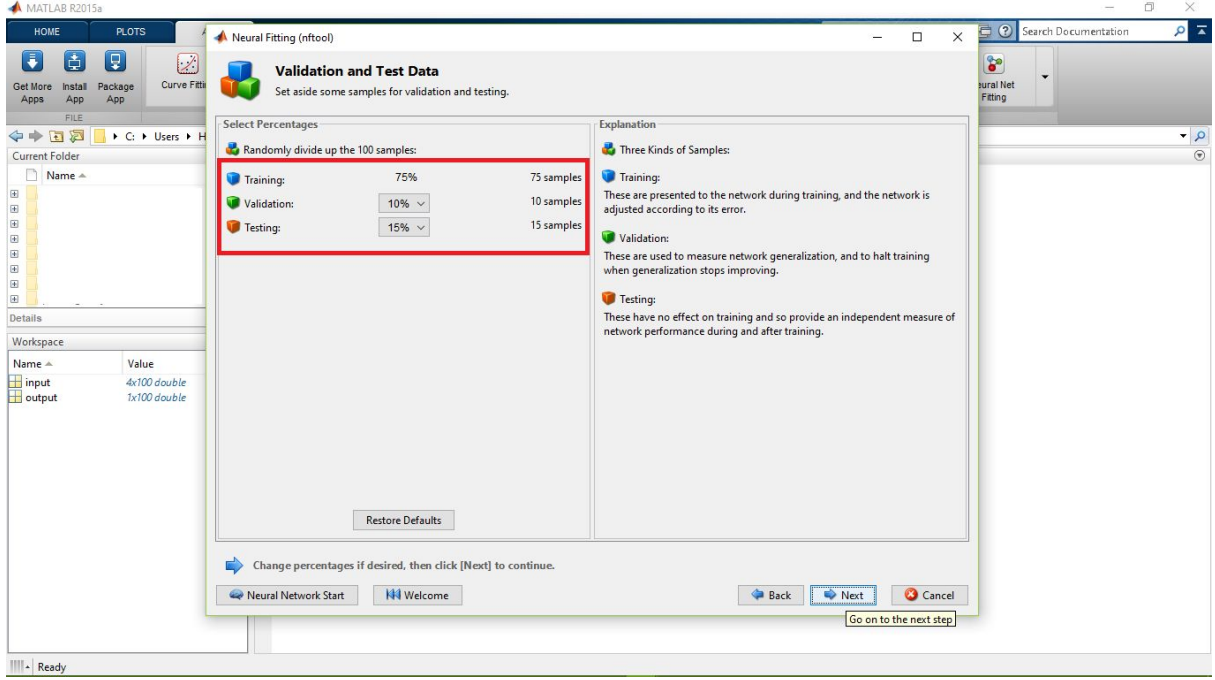
3.2. Neural Network Fitting Tool ile Hava Tahmin Uygulaması

Nftool aracını kullanmadan önce verilerimizi eklemek için MATLAB workspace alanındaki alana sağ tıklayarak input ve output isimli değişkenleri oluşturarak bu değerleri giriyoruz. Ardından veri seçim penceresinden input ve target olarak tanımlı alanlara, daha önceden tanımladığımız değişkenlerimizi Şekil 3.1 'deki belirtiyoruz.



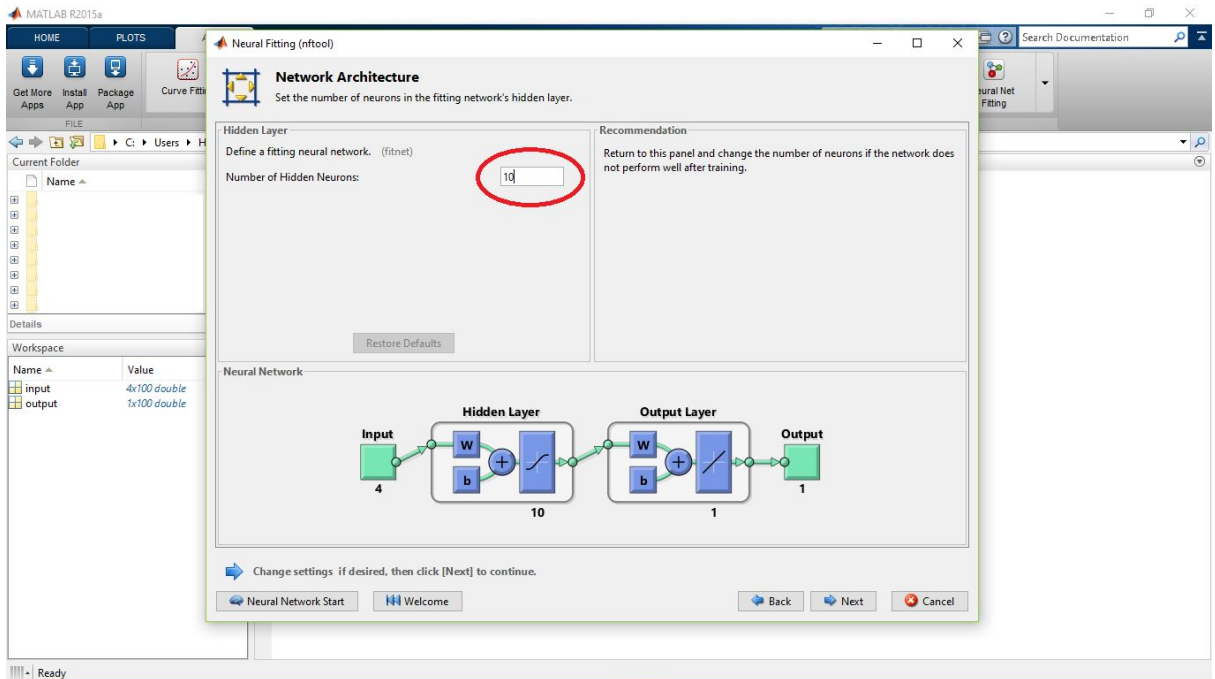
Şekil 3.1: YSA 'da kullanılacak değişkenlerin tanımlanıp, belirtilmesi

Değişkenlerin belirtilmesinden sonra, seçilen verilerin yüzde kaçının hangi amaçla kullanılacağını belirtiyoruz. Bu çalışmada verilerin %75'i eğitim, %10 'u doğrulama ve %15 'i test amaçlı olarak kullanılmıştır. Burada farklı amaçlar için yüzdelik dilim şeklinde ayrıştırılan veriler MATLAB tarafından rastgele seçilmektedir. Şekil 3.2 'de verilerin yüzdelik dilimlere ayrıştırılması görülmektedir.



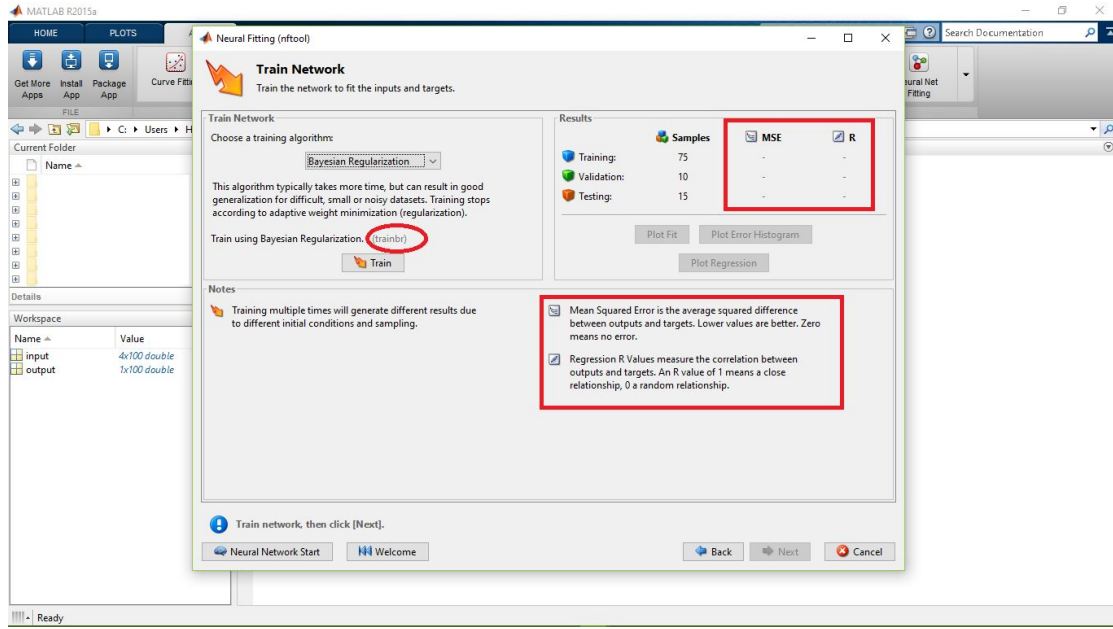
Şekil 3.2: Verilerin kullanım amacı olarak ayrıştırılması

Bu adımda yapay sinir ağı için kullanılacak olan mimari yapı belirlenir. Bu çalışmada kullanılan veriler için en iyi sonuçlar 10 adet nöron olarak belirlenmiştir. Projede kullanılan YSA mimarisi Şekil 3.3 'de belirtilmiştir.



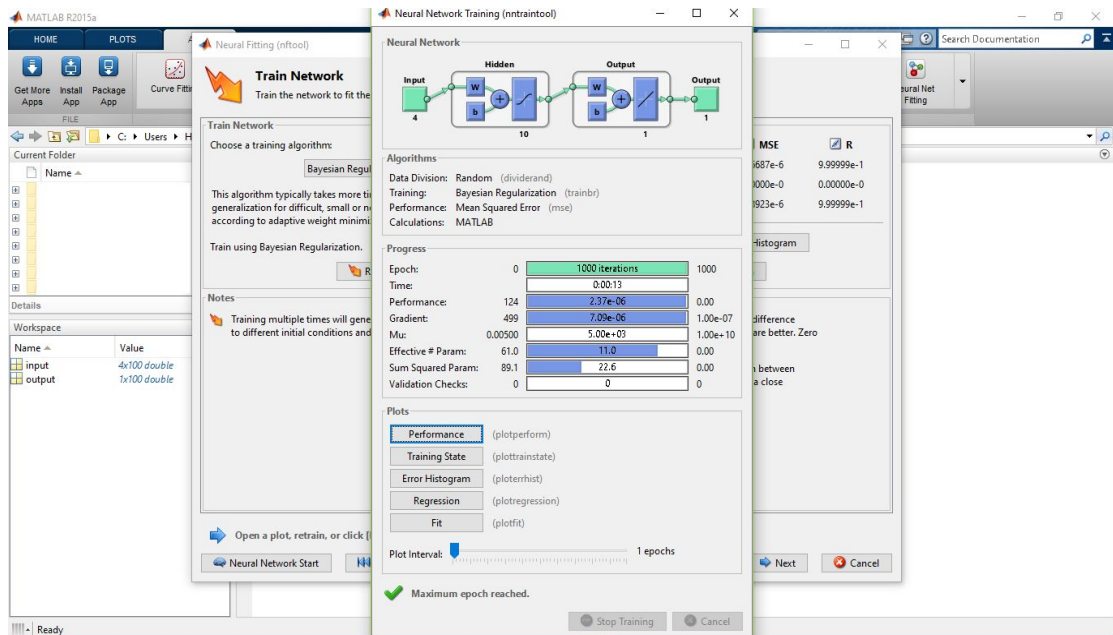
Şekil 3.3: Yapay sinir ağı mimarisinin belirlenmesi

Bir sonraki adım olan ağı eğitilmesi adımında, yapay sinir ağının hani algoritma kullanılarak eğitileceğini seçiyoruz. Bayesian Regularization eğitimi, eğitimde uzun zaman almasına rağmen iyi sonuçlar vermesinden dolayı, proje için bu algoritma seçilmiştir. Şekil 3.4 'de belirtilen MSE (Mean Squared Error) değerinin sıfıra yakın olması sonucun iyi, hata oranının düşük olduğu anlamına gelir. Şekil 3.4 'de belirtilen R (Regression) değerinin bire yakın olması ise veriler arasındaki ilişkinin kuvvetli olduğu anlamına gelmektedir.



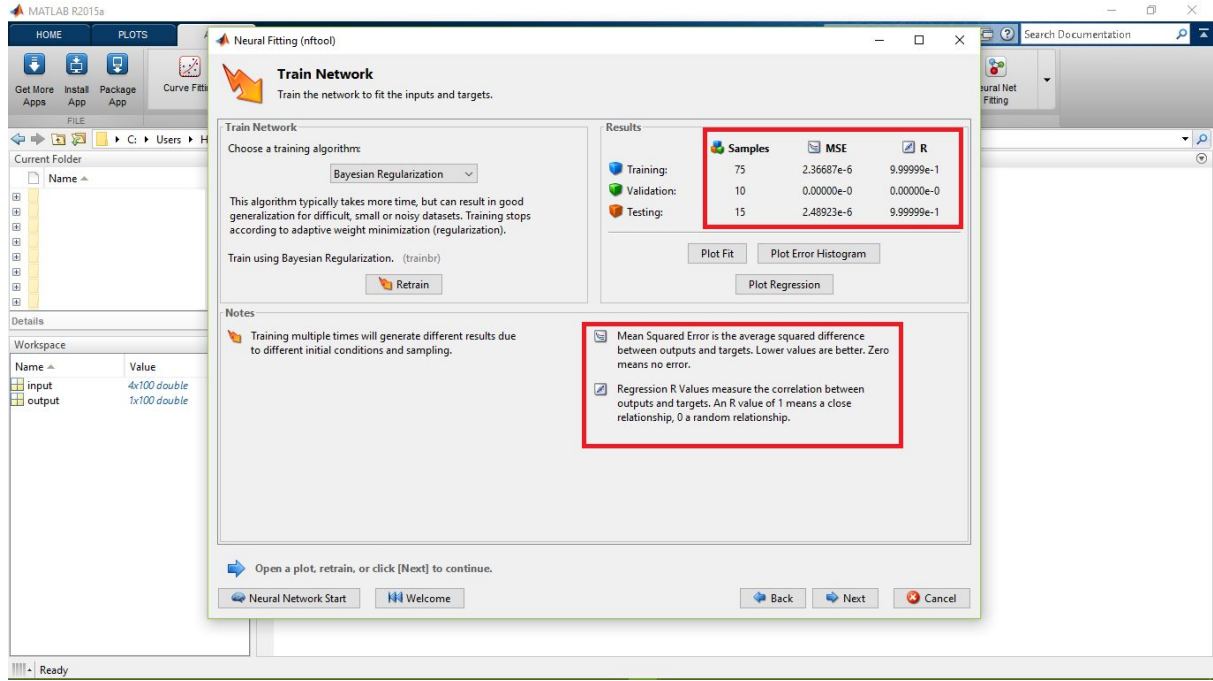
Şekil 3.4: Ağın eğitilmesi

Şekil 3.5 'de ise ağın eğitim süreci görülmektedir.



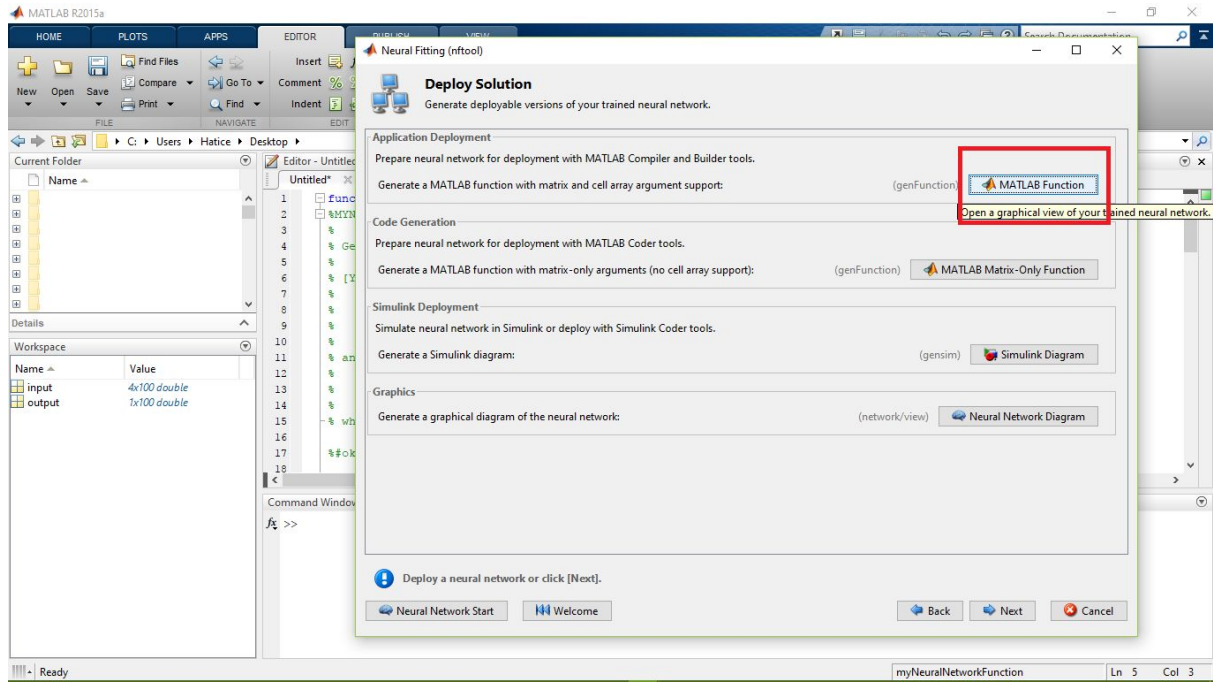
Şekil 3.5: Ağın eğitim süreci

Şekil 3.6 'da eğitilen ağı ait sonuçlar görülmektedir.



Şekil 3.6: Eğitilen ağı ait sonuçlar

Şekil 3.7 'de oluşturulan yapay sinir ağı için MATLAB ile otomatik olarak fonksiyon oluşturulur.



Şekil 3.7: Yapay sinir ağı fonksiyonun oluşturulduğu arayüz

4. SONUÇ

Bu çalışmada Konya iline ait 15 Haziran 2016 ve 22 Eylül 2016 tarihleri arasındaki nem oranı (%), görüş mesafesi (km), basınç (mb), çığ noktası ($^{\circ}\text{C}$), sıcaklık ($^{\circ}\text{C}$) bilgilerini içeren bir veri seti oluşturularak yapay sinir ağı kullanılarak geleceğe yönelik hava tahmini uygulaması MATLAB programlama diliyle geliştirilmiştir. geliştirilen uygulamada 10 adet nöron kullanılmış ve MSE oranı $2.36687\text{e-}6$ olarak çıkmıştır. Verilerin daha da arttırılarak daha iyi eğitim gerçekleştirilmesi sağlanabilir.

EK-1

Nem Oranı (%) Norm	Görüş Mesafesi (km) Norm	Basınç (mb) Norm	Çiğ Noktası (°C) Norm	Sıcaklık (°C) Norm
0.294117647	0.428571429	0.294117647	0.416666667	0.294117647
0.352941176	0.476190476	0.352941176	0.416666667	0.352941176
0.411764706	0.523809524	0.411764706	0.5	0.411764706
0.470588235	0.571428571	0.470588235	0.416666667	0.470588235
0.470588235	0.571428571	0.470588235	0.583333333	0.470588235
0.588235294	0.666666667	0.588235294	0.583333333	0.588235294
0.529411765	0.619047619	0.529411765	0.583333333	0.529411765
0.705882353	0.761904762	0.705882353	0.75	0.705882353
0.647058824	0.714285714	0.647058824	0.916666667	0.647058824
0.588235294	0.666666667	0.588235294	0.833333333	0.588235294
0.647058824	0.714285714	0.647058824	0.916666667	0.647058824
0.647058824	0.714285714	0.647058824	0.583333333	0.647058824
0.705882353	0.761904762	0.705882353	0.75	0.705882353
0.588235294	0.666666667	0.588235294	0.833333333	0.588235294
0.352941176	0.476190476	0.352941176	0.833333333	0.352941176
0.176470588	0.333333333	0.176470588	0.583333333	0.176470588
0.470588235	0.571428571	0.470588235	0.666666667	0.470588235
0.529411765	0.619047619	0.529411765	0.583333333	0.529411765
0.647058824	0.714285714	0.647058824	0.666666667	0.647058824
0.470588235	0.571428571	0.470588235	0.75	0.470588235
0.411764706	0.523809524	0.411764706	0.666666667	0.411764706
0.470588235	0.571428571	0.470588235	0.583333333	0.470588235
0.647058824	0.714285714	0.647058824	0.583333333	0.647058824

0.352941176	0.476190476	0.352941176	0.416666667	0.352941176
0.411764706	0.523809524	0.411764706	0.333333333	0.411764706
0.411764706	0.523809524	0.411764706	0.333333333	0.411764706
0.352941176	0.476190476	0.352941176	0.333333333	0.352941176
0.352941176	0.476190476	0.352941176	0.333333333	0.352941176
0.647058824	0.714285714	0.647058824	0.5	0.647058824
0.823529412	0.857142857	0.823529412	0.75	0.823529412
0.823529412	0.857142857	0.823529412	0.833333333	0.823529412
1	1	1	0.916666667	1
1	1	1	1	1
1	1	1	0.916666667	1
0.647058824	0.714285714	0.647058824	0.833333333	0.647058824
0.470588235	0.571428571	0.470588235	0.75	0.470588235
0.294117647	0.428571429	0.294117647	0.666666667	0.294117647
0.470588235	0.571428571	0.470588235	0.666666667	0.470588235
0.588235294	0.666666667	0.588235294	0.666666667	0.588235294
0.588235294	0.666666667	0.588235294	0.583333333	0.588235294
0.529411765	0.619047619	0.529411765	0.583333333	0.529411765
0.588235294	0.666666667	0.588235294	0.666666667	0.588235294
0.705882353	0.761904762	0.705882353	0.666666667	0.705882353
0.647058824	0.714285714	0.647058824	0.833333333	0.647058824
0.647058824	0.714285714	0.647058824	0.75	0.647058824
0.705882353	0.761904762	0.705882353	0.833333333	0.705882353
0.647058824	0.714285714	0.647058824	0.833333333	0.647058824
0.647058824	0.714285714	0.647058824	0.833333333	0.647058824
0.764705882	0.80952381	0.764705882	0.833333333	0.764705882

0.705882353	0.761904762	0.705882353	0.916666667	0.705882353
0.647058824	0.714285714	0.647058824	1	0.647058824
0.529411765	0.619047619	0.529411765	0.75	0.529411765
0.529411765	0.619047619	0.529411765	0.75	0.529411765
0.529411765	0.619047619	0.529411765	0.75	0.529411765
0.529411765	0.619047619	0.529411765	0.666666667	0.529411765
0.529411765	0.619047619	0.529411765	0.583333333	0.529411765
0.588235294	0.666666667	0.588235294	0.833333333	0.588235294
0.705882353	0.761904762	0.705882353	0.833333333	0.705882353
0.823529412	0.857142857	0.823529412	0.833333333	0.823529412
0.705882353	0.761904762	0.705882353	0.916666667	0.705882353
0.470588235	0.571428571	0.470588235	0.75	0.470588235
0.411764706	0.523809524	0.411764706	0.5	0.411764706
0.470588235	0.571428571	0.470588235	0.583333333	0.470588235
0.470588235	0.571428571	0.470588235	0.583333333	0.470588235
0.529411765	0.619047619	0.529411765	0.583333333	0.529411765
0.647058824	0.714285714	0.647058824	0.5	0.647058824
0.764705882	0.80952381	0.764705882	0.75	0.764705882
0.647058824	0.714285714	0.647058824	0.583333333	0.647058824
0.647058824	0.714285714	0.647058824	0.75	0.647058824
0.705882353	0.761904762	0.705882353	0.833333333	0.705882353
0.647058824	0.714285714	0.647058824	0.916666667	0.647058824
0.470588235	0.571428571	0.470588235	0.75	0.470588235
0.588235294	0.666666667	0.588235294	0.75	0.588235294
0.588235294	0.666666667	0.588235294	0.666666667	0.588235294
0.529411765	0.619047619	0.529411765	0.583333333	0.529411765

0.529411765	0.619047619	0.529411765	0.666666667	0.529411765
0.529411765	0.619047619	0.529411765	0.583333333	0.529411765
0.470588235	0.571428571	0.470588235	0.75	0.470588235
0.352941176	0.476190476	0.352941176	0.583333333	0.352941176
0.294117647	0.428571429	0.294117647	0.583333333	0.294117647
0.176470588	0.333333333	0.176470588	0.333333333	0.176470588
0.235294118	0.380952381	0.235294118	0.333333333	0.235294118
0.352941176	0.476190476	0.352941176	0.416666667	0.352941176
0.470588235	0.571428571	0.470588235	0.416666667	0.470588235
0.529411765	0.619047619	0.529411765	0.5	0.529411765
0.470588235	0.571428571	0.470588235	0.5	0.470588235
0.529411765	0.619047619	0.529411765	0.333333333	0.529411765
0.588235294	0.666666667	0.588235294	0.5	0.588235294
0.588235294	0.666666667	0.588235294	0.5	0.588235294
0.529411765	0.619047619	0.529411765	0.5	0.529411765
0.470588235	0.571428571	0.470588235	0.5	0.470588235
0.411764706	0.523809524	0.411764706	0.5	0.411764706
0.235294118	0.380952381	0.235294118	0.5	0.235294118
0.235294118	0.380952381	0.235294118	0.333333333	0.235294118
0.117647059	0.285714286	0.117647059	0.166666667	0.117647059
0.235294118	0.380952381	0.235294118	0	0.235294118
0.235294118	0.380952381	0.235294118	0.416666667	0.235294118
0.058823529	0.238095238	0.058823529	0.25	0.058823529
0	0.19047619	0	0.083333333	0
0	0	0	0.083333333	0

EK-2

```
function [Y,Xf,Af] = myNeuralNetworkFunction(X,~,~)
%MYNEURALNETWORKFUNCTION neural network simulation function.
%
% Generated by Neural Network Toolbox function genFunction.
%
% [Y] = myNeuralNetworkFunction(X,~,~) takes these arguments:
%
% X = 1xTS cell, 1 inputs over TS timesteps
% Each X{1,ts} = 4xQ matrix, input #1 at timestep ts.
%
% and returns:
% Y = 1xTS cell of 1 outputs over TS timesteps.
% Each Y{1,ts} = 1xQ matrix, output #1 at timestep ts.
%
% where Q is number of samples (or series) and TS is the number of timesteps.

%#ok<*RPMT0>

% ===== NEURAL NETWORK CONSTANTS =====

% Input 1
x1_step1_xoffset = [55;22;31;12];
x1_step1_gain =
[0.0470588235294118;0.0952380952380952;0.117647058823529;0.166666666666667];
x1_step1_ymin = -1;

% Layer 1
b1 =
[0.014638598170407918;-0.0034444794838031042;-0.021331000261126661;-0.001914048
8963371965;-0.009833938543653678;0.011103839845128465;0.0042820870972154175;-0
.013381405270922927;0.0095869323677963986;-0.0017694738057315736];
IW1_1 = [-0.0094387456096692978 -0.024709075429821103 -0.0094387456105427935
-0.034609195775985549;-0.033020447757527931 0.0014999959948119861
-0.033020447758321886 -0.0021659342932196955;0.040474656698281326
-0.0061562313523083638 0.040474656697073286
-0.010293739666005575;0.057266234565672598 -0.0014876369512787351
0.057266234565232353 -0.058314998489566207;-0.049778093046463308
0.00015628164936084019 -0.049778093046967314
0.039330047731581219;0.0020959454239551758 0.0067733691066981247
0.0020959454246050131 0.071010587480362722;0.045308961451659517
-0.00064217240837063027 0.045308961451602132
-0.028561492523628516;0.035109146093514045 -0.0058484465434834555
0.035109146094200822 0.0010201285583462119;0.035120083569370693
```

```

0.00050633644551388904 0.035120083570184438
-0.0049581119074815798;0.015722592564277011 -0.0080795975503425901
0.01572259256324544 0.051790775039465532];

% Layer 2
b2 = 0.019191812763293372;
LW2_1 = [-1.1516867245755784 -1.523944906507245 1.5732907847749225
1.6142539325380161 -1.5590848075965489 1.4100655449522546 1.5489038401908306
1.5540184267881487 1.5164464504713273 1.517315115051356];

% Output 1
y1_step1_ymin = -1;
y1_step1_gain = 0.117647058823529;
y1_step1_xoffset = 22;

% ===== SIMULATION =====

% Format Input Arguments
isCellX = iscell(X);
if ~isCellX, X = {X}; end;

% Dimensions
TS = size(X,2); % timesteps
if ~isempty(X)
    Q = size(X{1},2); % samples/series
else
    Q = 0;
end

% Allocate Outputs
Y = cell(1,TS);

% Time loop
for ts=1:TS

    % Input 1
    Xp1 = mapminmax_apply(X{1,ts},x1_step1_gain,x1_step1_xoffset,x1_step1_ymin);

    % Layer 1
    a1 = tansig_apply(repmat(b1,1,Q) + IW1_1*Xp1);

    % Layer 2
    a2 = repmat(b2,1,Q) + LW2_1*a1;

    % Output 1
    Y{1,ts} = mapminmax_reverse(a2,y1_step1_gain,y1_step1_xoffset,y1_step1_ymin);

```



```

end

% Final Delay States
Xf = cell(1,0);
Af = cell(2,0);

% Format Output Arguments
if ~isCellX, Y = cell2mat(Y); end
end

% ===== MODULE FUNCTIONS =====

% Map Minimum and Maximum Input Processing Function
function y = mapminmax_apply(x,settings_gain,settings_xoffset,settings_ymin)
    y = bsxfun(@minus,x,settings_xoffset);
    y = bsxfun(@times,y,settings_gain);
    y = bsxfun(@plus,y,settings_ymin);
end

% Sigmoid Symmetric Transfer Function
function a = tansig_apply(n)
    a = 2 ./ (1 + exp(-2*n)) - 1;
end

% Map Minimum and Maximum Output Reverse-Processing Function
function x = mapminmax_reverse(y,settings_gain,settings_xoffset,settings_ymin)
    x = bsxfun(@minus,y,settings_ymin);
    x = bsxfun(@rdivide,x,settings_gain);
    x = bsxfun(@plus,x,settings_xoffset);
end

```

KAYNAKLAR

[1] Yapay Sinir Ağları (YSA) Nedir ? (2016, February 11). Retrieved January 1, 2018, from <http://kod5.org/yapay-sinir-aglari-ysa-nedir/>

[2] Durumu, 1. G. (n.d.). Hava durumu. Retrieved January 1, 2018, from <https://www.havadurumu15gunluk.net/>