```
(*Author: Mohamed Kamal AbdElrahman
  Date Apr 24, 2022

 Regression with Neural Networks in Wolfram Langauge

 Goals:
      1- what neural newtworks are ?  Layers ? Functions ? Compositions ?
      2- linear regression with linear layers
      3- why we need nonlinearity ?
      4- why we need to  go to higher dimensions ?
*)
```

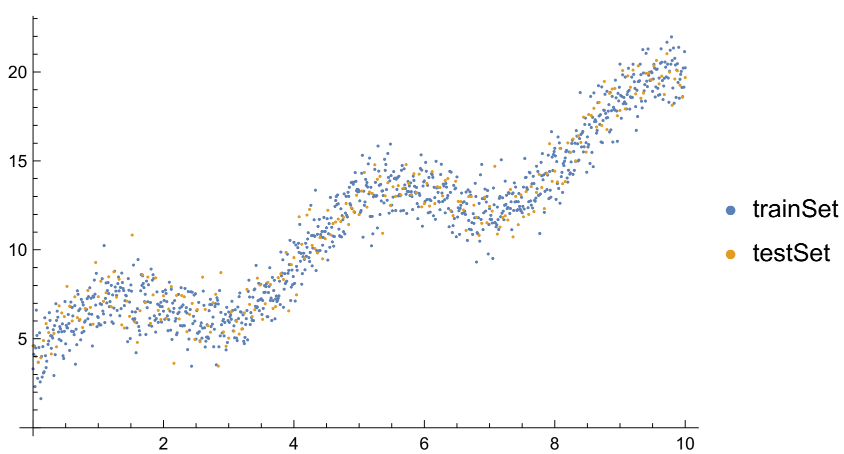
Linear/Nonlinear Regression with Neural Nets

In[359]:=
```
SetDirectory[NotebookDirectory[]]
```
Out[359]= /home/mk/Desktop/DeepLearningWithMathematica/Linear Regression

In[360]:=
```
FileNames[]
```
Out[360]= {RegressionNN.nb, test.csv, train.csv}

In[363]:=
```
trainSet = Import["train.csv"];
testSet = Import["test.csv"];
```

In[365]:=
```
ListPlot[{trainSet, testSet}, PlotLegends → {"trainSet", "testSet"}]
```

Out[365]=


Linear Regression

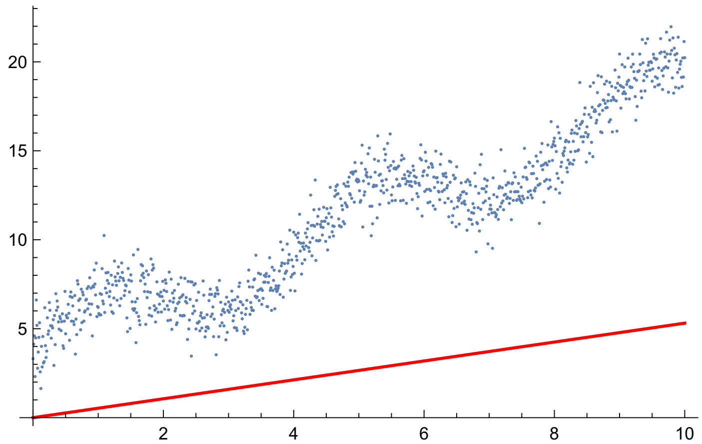
$f(x) = w\,x + b$

In[371]:=
```
f = NetInitialize[LinearLayer[{}, "Input" → {}]]
```

Out[371]= LinearLayer[Input: real Output: real]

In[396]:=
```
Show[ListPlot[trainSet], Plot[f[x], {x, 0, 10}, PlotStyle → Red]]
```

Out[396]=

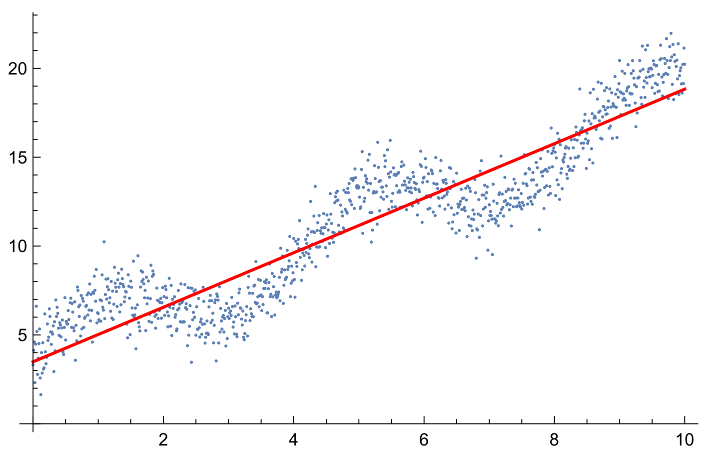


In[394]:=
```
trainedf = NetTrain[f, trainSet〚All, 1〛 → trainSet〚All, 2〛, TimeGoal → 10]
```

Out[394]= LinearLayer[Input: real Output: real]

In[395]:=
```
Show[ListPlot[trainSet], Plot[trainedf[x], {x, 0, 10}, PlotStyle → Red]]
```

Out[395]=



In[400]:=
```
NetMeasurements[trainedNet, testSet〚All, 1〛 → testSet〚All, 2〛, "MeanSquare"]
```

Out[400]= 3.20441

Adding more Linear Layers

In[472]:=
```
f11 = NetInitialize[NetChain[{{}, {}, {}}, "Input" → {}]]
```

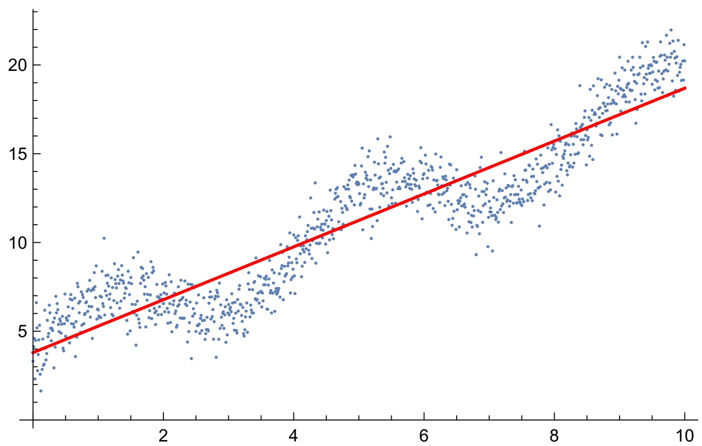
Out[472]=
```
NetChain[ Input port: real
          Output port: real ]
```

In[473]:=
```
trainedf11 = NetTrain[f11, trainSet[[All, 1]] → trainSet[[All, 2]], TimeGoal → 10]
```

Out[473]=
```
NetChain[ Input port: real
          Output port: real ]
```

In[474]:=
```
Show[ListPlot[trainSet], Plot[trainedf11[x], {x, 0, 10}, PlotStyle → Red]]
```

Out[474]=



Adding a Nonlinearity

In[432]:=
```
f2 = NetInitialize[NetChain[{{}, LogisticSigmoid, {}}, "Input" → {}]]
```

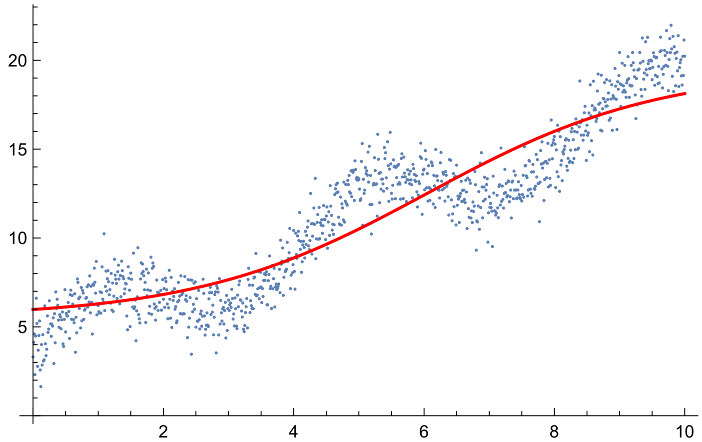
Out[432]=
```
NetChain[ Input port: real
          Output port: real ]
```

In[429]:=
```
trainedf2 = NetTrain[f2, trainSet[[All, 1]] → trainSet[[All, 2]], TimeGoal → 40]
```

Out[429]=
```
NetChain[ Input port: real
          Output port: real ]
```

In[430]:= `Show[ListPlot[trainSet], Plot[trainedf2[x], {x, 0, 10}, PlotStyle → Red]]`

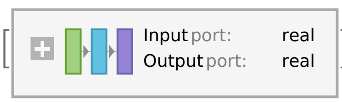
Out[430]=



Adding More Nonlinearity

In[454]:= `f3 = NetInitialize[NetChain[{{}, Tanh, {}, Tanh, {}, Tanh, {}, Tanh, {}}, "Input" → {}]]`

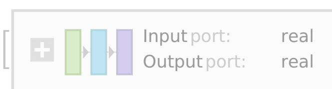
Out[454]= NetChain[



]

`trainedf3 = NetTrain[f3, trainSet⟦All, 1⟧ → trainSet⟦All, 2⟧, TimeGoal → 40]`

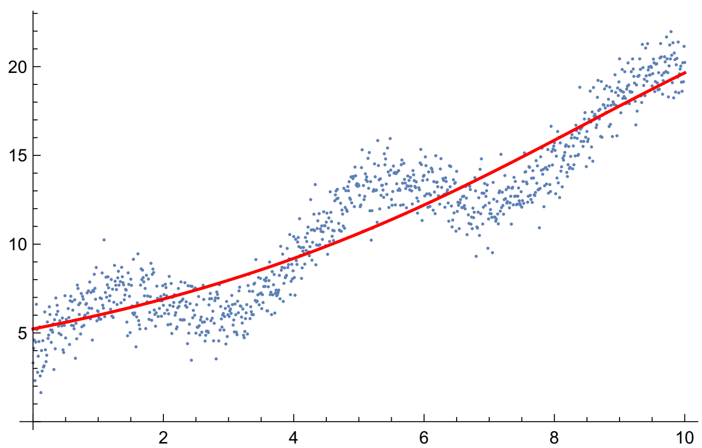
Out[455]= NetChain[



]

In[456]:= `Show[ListPlot[trainSet], Plot[trainedf3[x], {x, 0, 10}, PlotStyle → Red]]`

Out[456]=



Going to Higher Dimensions

In[469]:=
```
f4 = NetInitialize[NetChain[{{10}, Tanh, {}}, "Input" → {}]]
```

Out[469]= NetChain[Input port: real Output port: real]

In[470]:=
```
trainedf4 = NetTrain[f4, trainSet[[All, 1]] → trainSet[[All, 2]], TimeGoal → 40]
```

Out[470]= NetChain[Input port: real Output port: real]

In[476]:=
```
Show[ListPlot[trainSet], Plot[trainedf4[x], {x, 0, 10}, PlotStyle → Red]]
```

Out[476]=