MmaGMDH

MmaGMDH is a *Mathematica* program for the implementation of the traditional GMDH algorithm.

For more on the GMDH see:
http://www.GMDH.net

MmaGMDH utilizes the power inherent in *Mathematica*'s Functional Programming language, resulting in a revolutionary implementation of the GMDH algorithm. In a few lines of *Mathematica* code MmaGMDH produces both numeric and symbolic solutions for the GMDH. An option is provided that allows the user to turn off the generation of a symbolic solution - allowing for faster solution times when only numeric results are necessary.

All of the intermediate results for the GMDH solution are available for "interactive" post processing.

The data sets that need to be initialized are as follows:

xtr[1]	Input training data.	{{xtr_1},{xtr_2},...,{xtr_m}}
xte[1]	Input testing data.	{{xte_1},{xte_2},...,{xte_m}}
xva[1]	Validation data.	{{xva_1},{xva_2},...,{xva_m}}
ztr	Output training data.	{ztr}
zte	Output testing data.	{zte}
zva	Output validation data.	{zva}

Where "m" is the number of inputs and:
Length[xtr_i]=Length[ztr] = number of training data for fitting nodal surfaces using least squares.
Length[xte_i]=Length[zte] =number of testing data for pruning nodes at each layer.
Length[xva_i]=Length[zva]=number of validation data.

An example for using the MmGMDH is provided that utilizes Jerome Friedman's "add10" database.

Description of add10 database:
The synthetic dataset "add10" uses a function suggested by Jerome Friedman in "Multivariate Adaptive Regression Splines", technical Report No. 102, November 1988, Laboratory for Computational Statistics, Department of Statistics, Stanford University.

The function is:
f(x1,...,x10) = 10*sin(pi*x1*x2) + 20*(x3-0.5)^2 + 10*x4 + 5*x5 + n, where n is zero mean unit variance Gaussian noise. The inputs x1...x10 are sampled independently from a uniform [0; 1] distribution.

The "add10" dataset is intended to provide an example where some inputs are irrelevant for predictions and where the function has a predominantly additive structure.

Number of instances: 8181.

see:
http://www.work.caltech.edu/cs156/96/projects/optimization/add10.html
http://www.cs.toronto.edu/~delve/data/add10/desc.html

Note: The average value for the sum-squared-error over the validation set of 3.65 (Layer 4) compares favorably with the value 3.9 reported in:
Voss, M.S. (2003) "Social Programming on MARS",
Proceedings 2003 Genetic and Evolutionary Computation Conference,
July 12-16, Chicago, Illinois (Late Breaking Papers).

see:
http://www.evolutionarystructures.com/papers/voss_gecco2003.pdf

Please send any comments/corrections or extensions of MmaGMDH to:
MarkVoss@EvolutionaryStructures.com

Any submitted comments/corrections/improved versions, will be review and posted to www.gmdh.net per webmaster approval.

Import "add10.dat" and initialize global data sets.

In[42]:=

```
SetDirectory["D:\\MmaGMDH2"];
data1 = Import["add10.dat", "Table"];
Length@data1
data2 = Take[data1, 1400];
zdata = #[[11]] & /@ data2;
xdata = Take[#, {1, 10}] & /@ data2;
```

Out[44]= 8181

In[48]:=

```
xtr[1] = Transpose@Take[xdata, {1, 200}];
xte[1] = Transpose@Take[xdata, {201, 400}];
xva[1] = Transpose@Take[xdata, {401, 1400}];
ztr = Take[zdata, {1, 200}];
zte = Take[zdata, {201, 400}];
zva = Take[zdata, {401, 1400}];
```

MmaGMDH - Module and Function Definitions.

In[54]:=

```
GMDHleastSquares[{x_, y_, z_}] := Module[{o, Y, A, b, a},
  o = Table[1, {Length[x]}];
  Y = {o, x, y, x y, x x, y y};
  A = Y.Transpose[Y];
  b = z.Transpose[Y];
  a = LinearSolve[A, b]]

GMDHerror[{x_, y_, z_, a_}] := Module[{xy, zp, e},
  xy = Transpose[{x, y}];
  zp = (a[[1]] + a[[2]] #[[1]] + a[[3]] #[[2]] +
       a[[4]] #[[1]] #[[2]] + a[[5]] #[[1]]^2 + a[[6]] #[[2]]^2) & /@ xy;
  e = z - zp;
  {e.e, zp} ]

Nchoose2[n_] := Module[{t1, t2, t3, t4},
  t = Range[n] ; t1 = Flatten[Outer[List, t, t], 1]; t2 = Map[Sort, t1]; t3 = Union[t2];
  t4 = Cases[t3, {x_, y_} /; (x =!= y)] ]

GMDHpoly[{x_, y_}, a_] := a[[1]] + a[[2]] x + a[[3]] y + a[[4]] x y + a[[5]] x^2 + a[[6]] y^2

GMDHeval[n_, xv_] := xso[n][[1]] /. MapThread[(#1 → #2) &, {xso[0], xv}]

GMDH[numLayers_, option_] := Module[{nunInputs, nc2, rngnc2},
  numInputs = Length[xtr[1]];
  nc2 = Nchoose2[numInputs];
  rngnc2 = Range@Length[nc2];
  xso[0] = Table[ToExpression[StringJoin["x" <> ToString[i]]], {i, 1, numInputs}];
  Table[
   tr[i] = Part[xtr[i], #] & /@ nc2;
   te[i] = Part[xte[i], #] & /@ nc2;
   va[i] = Part[xva[i], #] & /@ nc2;
   av[i] = GMDHleastSquares[{#[[1]], #[[2]], ztr}] & /@ tr[i];
   trer[i] = MapThread[GMDHerror[ {#1[[1]], #1[[2]], ztr, #2}] &, {tr[i], av[i]}];
   teer[i] = MapThread[GMDHerror[ {#1[[1]], #1[[2]], zte, #2}] &, { te[i], av[i]}];
   vaer[i] = MapThread[GMDHerror[ {#1[[1]], #1[[2]], zva, #2}] &, { va[i], av[i]}];
   trfit[i] = Sort@Thread[{#[[1]] & /@ trer[i], rngnc2}];
   tefit[i] = Sort@Thread[{#[[1]] & /@ teer[i], rngnc2}];
   vafit[i] = Sort@Thread[{#[[1]] & /@ vaer[i], rngnc2}];
   inputs[i] = Take[#[[2]] & /@ tefit[i], numInputs];
   xtr[i + 1] = Part[#[[2]] & /@ trer[i], inputs[i]];
   xte[i + 1] = Part[#[[2]] & /@ teer[i], inputs[i]];
   xva[i + 1] = Part[#[[2]] & /@ vaer[i], inputs[i]];
   If[Symbolic /. option,
    xsi[i] = Part[xso[i - 1], #] & /@ Part[nc2, inputs[i]];
    xso[i] = MapThread[GMDHpoly, {xsi[i], Part[av[i], inputs[i]]}];
   ];
   Print["Layer=", i]
   , {i, 1, numLayers}];
 ]
```

GMDH run with timing (8 layers with symbolic solution turned off)

```
In[60]:= Timing[GMDH[numLayers = 8, Symbolic -> False]][[1]]

         Layer=1

         Layer=2

         Layer=3

         Layer=4

         Layer=5

         Layer=6

         Layer=7

         Layer=8

Out[60]= 3.234 Second
```

GMDH run with timing (8 layers with symbolic solution turned on)

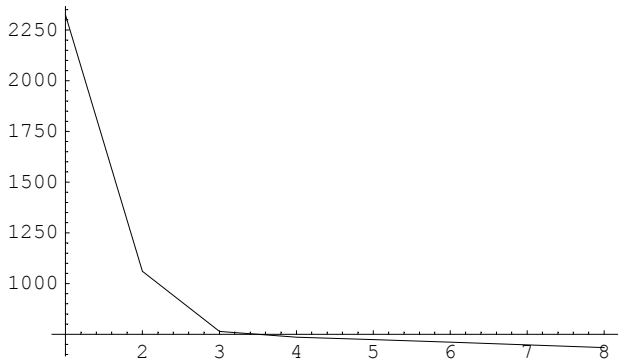
```
In[61]:= Timing[GMDH[numLayers = 8, Symbolic -> True]][[1]]

         Layer=1

         Layer=2

         Layer=3

         Layer=4

         Layer=5

         Layer=6

         Layer=7

         Layer=8

Out[61]= 8.344 Second
```

Plot of test,training and validation errors.

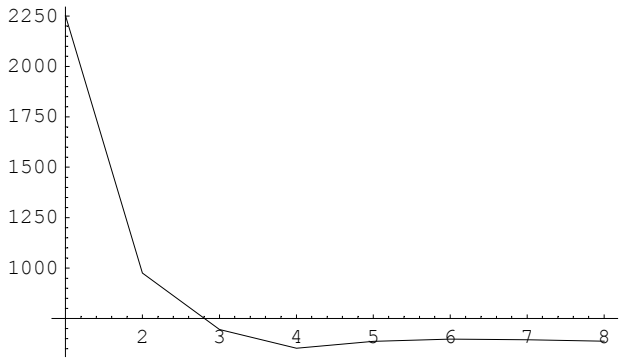
In[62]:=

```
Table[tefit[i][[1, 1]], {i, 1, numLayers}]
ListPlot[%, PlotJoined → True];

Table[trfit[i][[1, 1]], {i, 1, numLayers}]
ListPlot[%, PlotJoined → True];

Table[vafit[i][[1, 1]], {i, 1, numLayers}]
ListPlot[%, PlotJoined → True];
```

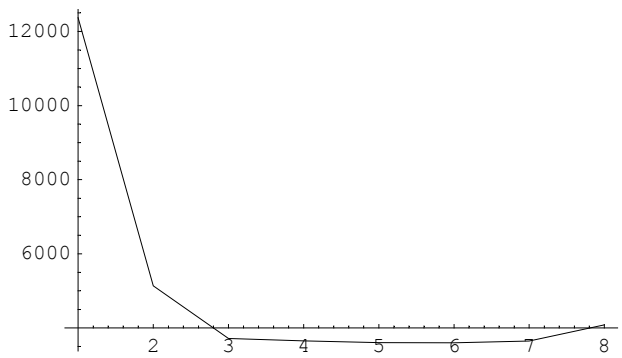
Out[62]= {2325.17, 1060.5, 763.558, 735.425, 724.19, 710.685, 697.781, 684.166}



Out[64]= {2253.33, 975.253, 695.652, 602.66, 635.892, 648.039, 644.673, 636.921}



Out[66]= {12371.4, 5136.65, 3712., 3651.53, 3601.29, 3596.65, 3648.95, 4083.27}



Best symbolic polynomials from layer's 1 and 2.

```
In[68]:= xso[1][[1]]
         FullSimplify@xso[2][[1]]

Out[68]= 4.21185 + 22.5828 x1 - 16.9661 x1^2 + 6.1141 x4 + 0.357527 x1 x4 + 5.00612 x4^2

Out[69]= -5.97124 + 3.82042 x1^2 + x1 (-5.99842 - 0.508844 x2) +
          5.66311 (-1.63108 + x2) (0.146746 + x2) +
          0.0475591 (3.5934 - 10.1279 (-1.5701 + x1) x1 + 22.2842 x2 + 1.34894 x1 x2 - 15.0129 x2^2)^2 +
          1.89929 x4^2 - 1.74499 (-4.34335 + x5) (1.21316 + x5) + x4 (7.11168 + 7.20351 x5) -
          0.00208979 (3.5934 - 10.1279 (-1.5701 + x1) x1 + 22.2842 x2 + 1.34894 x1 x2 - 15.0129 x2^2)
           (7.7153 + 1.5937 x4 (3.74438 + x4) + 4.58332 x5 + 6.0445 x4 x5 - 1.46423 x5^2) -
          0.00785011 (7.7153 + 1.5937 x4 (3.74438 + x4) + 4.58332 x5 + 6.0445 x4 x5 - 1.46423 x5^2)^2
```

Output of layer one's best symbolic polynomials in Fortran and C form.

```
In[70]:= CForm@xso[1][[1]]

Out[70]//CForm=
       4.2118460115651475 + 22.582786418223098*x1 -
          16.96606941533943*Power(x1,2) + 6.1141004542769375*x4 +
          0.35752719088834684*x1*x4 + 5.0061219373888965*Power(x4,2)

In[71]:= FortranForm@xso[1][[1]]

Out[71]//FortranForm=
                4.2118460115651475 + 22.582786418223098*x1 -
             -  16.96606941533943*x1**2 + 6.1141004542769375*x4 +
             -  0.35752719088834684*x1*x4 + 5.0061219373888965*x4**2
```

Calculation of data set using symbolic GMDH solution.

```
In[72]:= zpolypredictGMDH = GMDHeval[4, #] & /@ xdata;
```

Validation of symbolic GMDH solution over validation set.

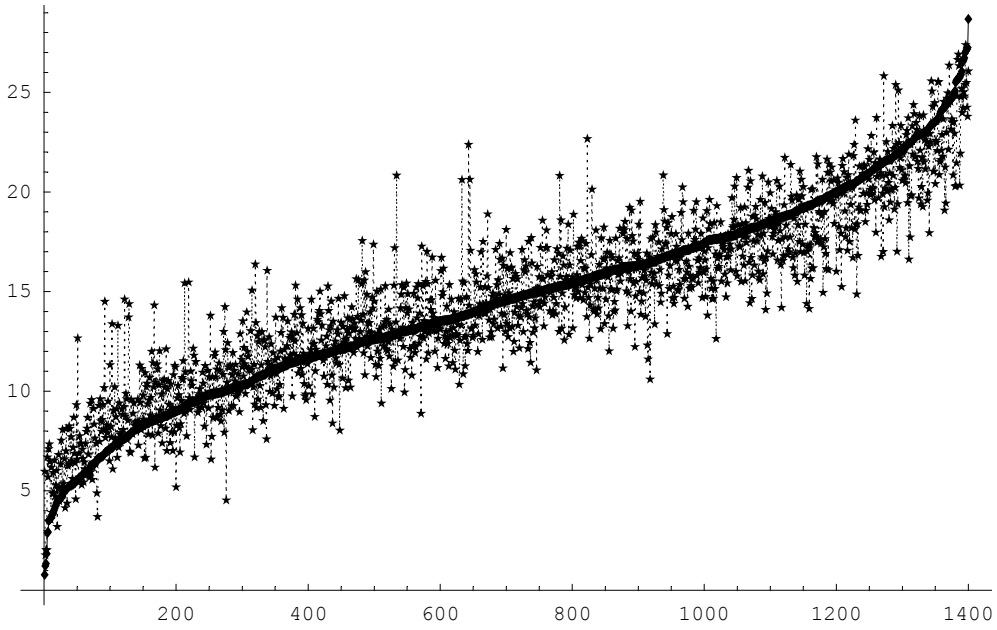
In[73]:=
```
zpolypredictGMDH1000 = Take[zpolypredictGMDH, {401, 1400}];
zze1000 = zpolypredictGMDH1000 - xva[5][[1]];
zze1000.zze1000
```

Out[75]= 3.09529×10^{-26}

Plot of symbolic solution vs add10.
Sorted on add10.

In[76]:=
```
<< Graphics`MultipleListPlot`
```

In[77]:=
```
zvaVSzpoly = Transpose@Sort[Drop[Transpose[{zdata, zpolypredictGMDH}], 0]];
MultipleListPlot[zvaVSzpoly[[1]], zvaVSzpoly[[2]], PlotJoined → True];
```



In[79]:=
```
<< Statistics`LinearRegression`
```

In[80]:=
```
data = Transpose[{xtr[1][[1]], xtr[1][[2]], ztr}];
```

In[81]:=
```
Timing[Table[Regress[data, {1, x1, x2, x1 x2, x1 x1, x2 x2}, {x1, x2}];, {1000}];]
```

Out[81]= {9.266 Second, Null}

In[82]:=
```
Timing[Table[GMDHleastSquares[{xtr[1][[1]], xtr[1][[2]], ztr}];, {1000}];]
```

Out[82]=
```
{1.125 Second, Null}
```