Approximation with sigmoid basis

In[990]:=
```
x = 0; Remove["Global`*"]; DateList[Date[]] // Most
```

Out[990]=
```
{2020, 5, 11, 21, 31}
```

The sigmoid function is a basic function used in neural nets. I will first examine its value in doing standard regression.

Basis functions

Define sigmoid basis

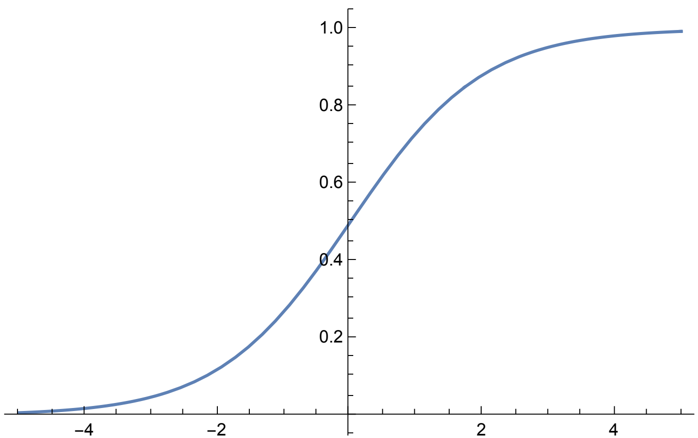
In[991]:=
```
Clear[G]
G[x_] = 1 / (1 + Exp[-x])
G[list_List] := G /@ list
```

Out[992]=
$$\frac{1}{1 + e^{-x}}$$

Choose range and plot basic sigmoid function

In[1019]:=
```
xmin = -5; xmax = 5;
Plot[G[x], {x, -5, 5}]
```

Out[1020]=



Set up approximation scheme. We shall use nnodes sigmoids centered at points in [xmin,xmax]

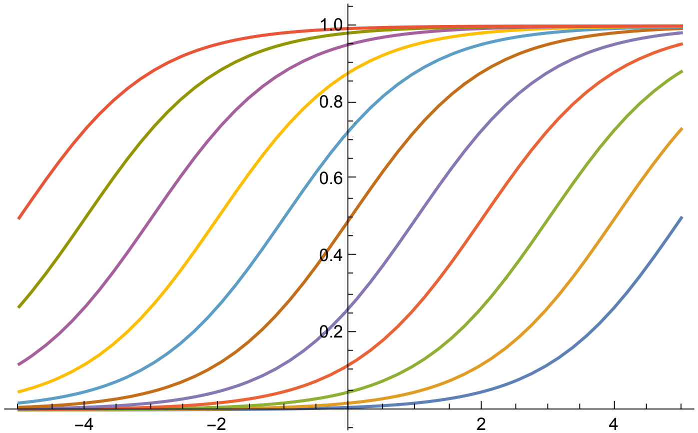
In[1021]:=
```
nnodes = 11; dx = (xmax - xmin) / (nnodes - 1);
nodes = Range[xmin, xmax, dx];
basis = Table[G[x + z], {z, nodes}]
```

Out[1023]=

$$\left\{\frac{1}{1+e^{5-x}}, \frac{1}{1+e^{4-x}}, \frac{1}{1+e^{3-x}}, \frac{1}{1+e^{2-x}}, \frac{1}{1+e^{1-x}}, \frac{1}{1+e^{-x}}, \frac{1}{1+e^{-1-x}}, \frac{1}{1+e^{-2-x}}, \frac{1}{1+e^{-3-x}}, \frac{1}{1+e^{-4-x}}, \frac{1}{1+e^{-5-x}}\right\}$$

In[1024]:=
```
Plot[basis, {x, -5, 5}]
```

Out[1024]=



Define model

In[1025]:=
```
model = Sum[a[i] × basis[[i]], {i, Length[basis]}]
```

Out[1025]=
$$\frac{a[1]}{1+e^{5-x}}+\frac{a[2]}{1+e^{4-x}}+\frac{a[3]}{1+e^{3-x}}+\frac{a[4]}{1+e^{2-x}}+\frac{a[5]}{1+e^{1-x}}+\frac{a[6]}{1+e^{-x}}+\frac{a[7]}{1+e^{-1-x}}+\frac{a[8]}{1+e^{-2-x}}+\frac{a[9]}{1+e^{-3-x}}+\frac{a[10]}{1+e^{-4-x}}+\frac{a[11]}{1+e^{-5-x}}$$

In[1026]:=
```
model /. x → 0;
vars = Variables[%]
```

Out[1027]=
```
{a[1], a[2], a[3], a[4], a[5], a[6], a[7], a[8], a[9], a[10], a[11]}
```

In[1028]:=
```
init = vars - vars;
varsin = {vars, init} // Transpose;
```

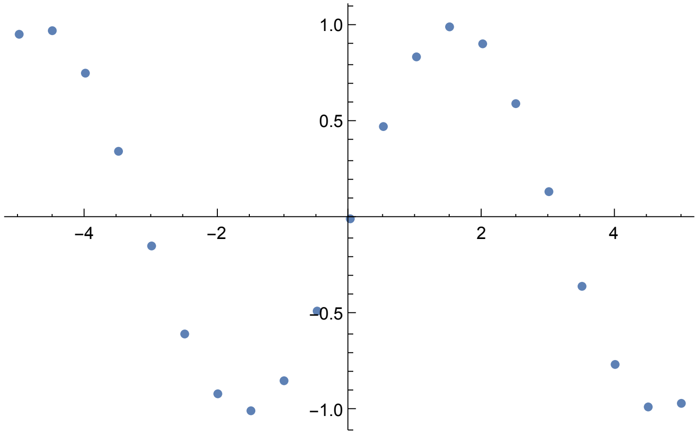
Fit an example function

In[1030]:=
```
f[x_] = Sin[x]
```

Out[1030]=
```
Sin[x]
```

In[1031]:=
```
xpts = (Range[21] - 11) / 2; ypts = f /@ xpts;
data = {xpts, ypts} // Transpose;
ListPlot[data]
```

Out[1033]=



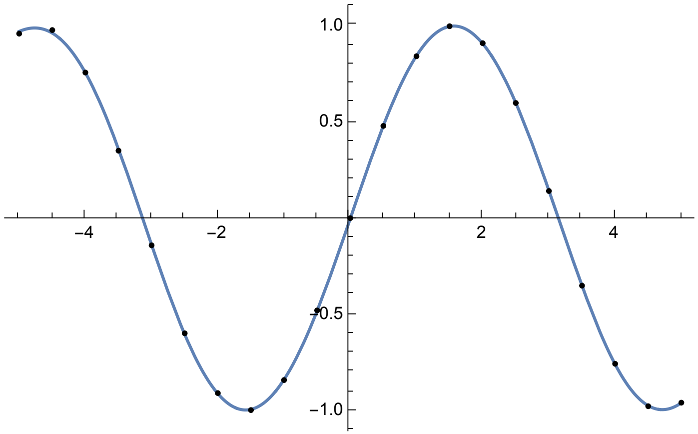
```
In[1034]:= fit = FindFit[data, model, varsin, x]

Out[1034]= {a[1] → 6.69448, a[2] → -6.04341, a[3] → -2.28347,
  a[4] → -1.1444, a[5] → 0.217219, a[6] → 6.6338, a[7] → -1.95719,
  a[8] → 2.95803, a[9] → -7.62965, a[10] → -0.931633, a[11] → 3.96544}

In[1035]:= modelf = Function[{x}, Evaluate[model /. fit]];

In[1036]:= Plot[modelf[x], {x, xmin, xmax}, Epilog → Map[Point, data]]
```

Out[1036]=



Not bad. Problems at the endpoints.

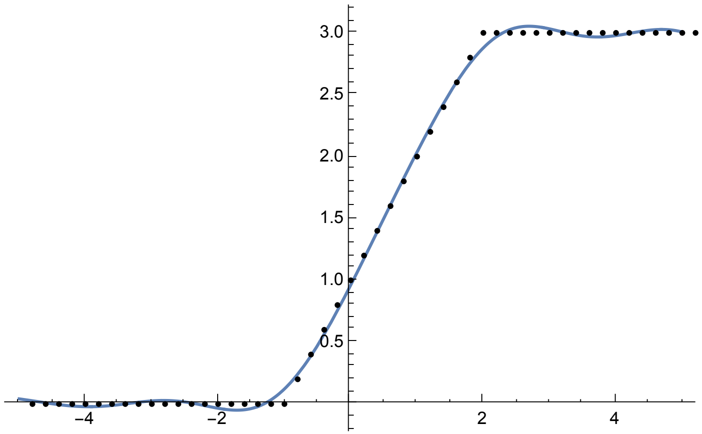
Define script

In[1037]:=
```
script := (
  xpts = (Range[51] - 25) / 5;
ypts = f /@ xpts;
data = {xpts, ypts} // Transpose;
fit = FindFit[data, model, varsin, x];
modelf = Function[{x}, Evaluate[model /. fit]];
Plot[modelf[x], {x, xmin, xmax}, Epilog → Map[Point, data]])
```

Fit an example function

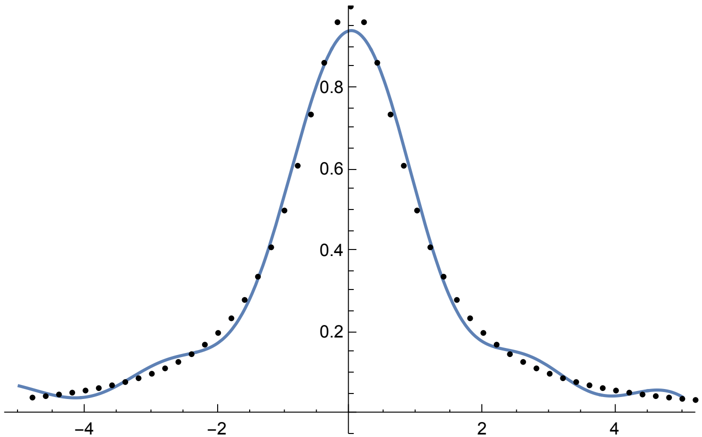
In[1038]:=
```
f[x_] = Min[Max[x + 1, 0], 3];
script
```

Out[1039]=



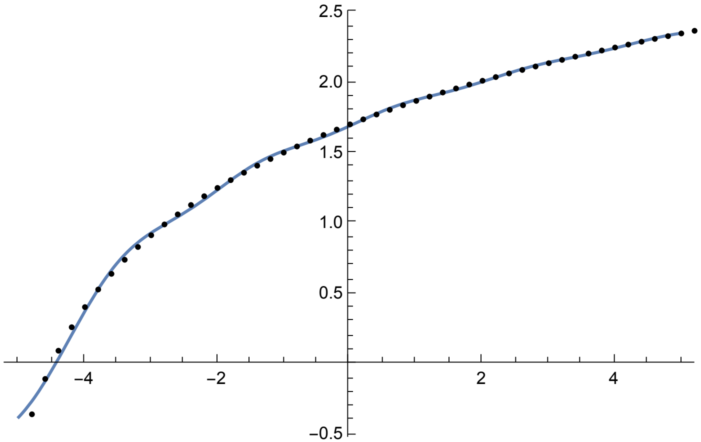
In[1040]:=
```
f[x_] = 1 / (1 + x^2);
script
```

Out[1041]=



In[1042]:=
```
f[x_] = Log[x + 5.5];
script
```

Out[1043]=



In[1044]:=
```
f[x_] = Sin[x^2 / 3 + x];
script
```

Out[1045]=