

Kapitola 9 - Sít' RBF a umělá data $\sin(x)$

Demonstrace použití RBF sítě na umělých datech - aproximace funkce sinus.

Načtení knihovny NeuralNetworks

Nejdříve načteme knihovnu neuronových sítí.

In[19]:=

```
<< NeuralNetworks`
```

Pokud pracujete v Mathematice 8.0, vypněte ještě zobrazování chybové hlášky Remove::rmnsm. Tuto hlášku vyhazují funkce knihovny NeuralNetworks. Na funkci knihovny toto nemá žádný vliv.

In[20]:=

```
Off[Remove::rmnsm]
```

Vytvoření trénovacích dat (Sin(x))

Vygenerujeme si umělá data - navzorkujeme sinusovku.

In[21]:=

```
n = 20;  
x = Table[N[2  $\pi$  / (n - 1) i], {i, 0, n - 1}];  
y = Sin[x];
```

Takto vypadají naše data :

In[24]:=

x

Out[24]=

```
{0., 0.330694, 0.661388, 0.992082, 1.32278, 1.65347,  
1.98416, 2.31486, 2.64555, 2.97625, 3.30694, 3.63763, 3.96833,  
4.29902, 4.62972, 4.96041, 5.2911, 5.6218, 5.95249, 6.28319}
```

In[25]:=

y

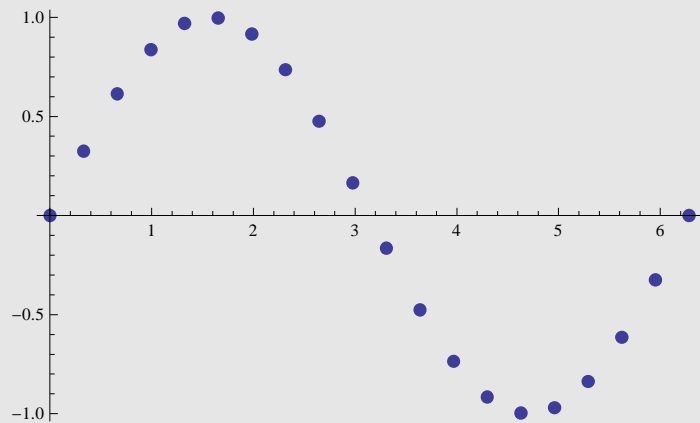
Out[25]=

```
{0., 0.324699, 0.614213, 0.837166, 0.9694, 0.996584, 0.915773,  
0.735724, 0.475947, 0.164595, -0.164595, -0.475947, -0.735724, -0.915773,  
-0.996584, -0.9694, -0.837166, -0.614213, -0.324699, -2.44929  $\times 10^{-16}$ }
```

In[26]:=

```
ListPlot[Transpose[{x, y}], PlotStyle -> PointSize[0.02]]
```

Out[26]=



Zpracování dat neuronovou sítí

Inicializace sítě - zadáme trénovací množinu, rozdělenou na vstupní a výstupní data, a počet neuronů. Počet neuronů se zadává jako kladné celé číslo.

Vytvořenou síť si uložíme do proměnné "net".

In[27]:=

```
net = InitializeRBFNet[x, y, 5, RandomInitialization -> True]
```

Out[27]=

```
RBFNet[{{w1, λ, w2}, χ}, {Neuron -> Exp, FixedParameters -> None,  
  AccumulatedIterations -> 0, CreationDate -> {2011, 5, 23, 0, 27, 47.1839159},  
  OutputNonlinearity -> None, NumberOfInputs -> 1}]
```

Můžeme si nechat zobrazit nějaké další informace o vytvořené síti.

In[28]:=

```
NetInformation[net]
```

Out[28]=

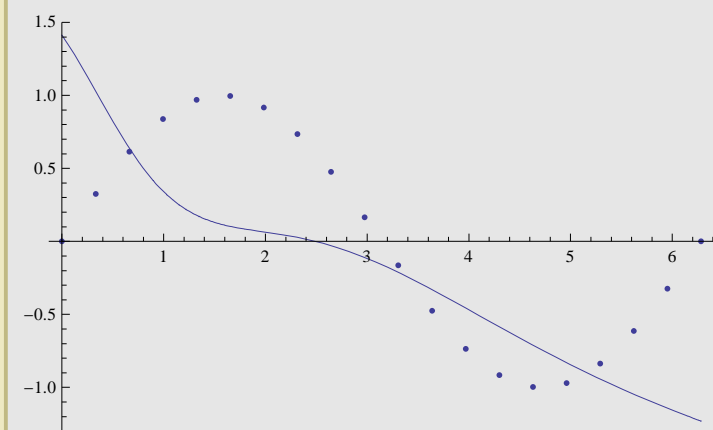
```
Radial Basis Function network. Created 2011-5-23 at  
  0:27. The network has 1 input and 1 output. It consists of 5  
  basis functions of Exp type. The network has a linear submodel.
```

Podíváme se jak naše, zatím náhodně inicializovaná, síť odpovídá na data.

In[29]:=

NetPlot[net, x, y]

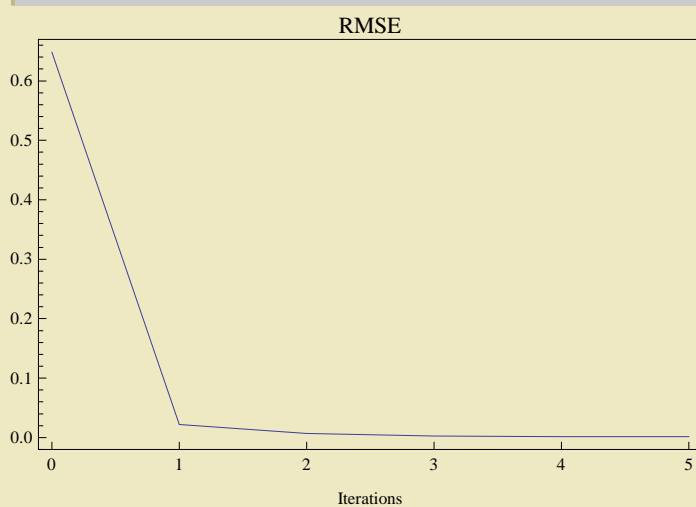
Out[29]=



Ted' síť natrénujeme pomocí funkce "NeuralFit", které zadáme naší síť (proměnná "net"), trénovací množinu ("x" a "y") a počet učících kroků.

Funkce NeuralFit vyprodukuje naučenou síť a záznam o průběhu učení - obě tyto návratové hodnoty si ukládáme (do proměnných "net2" a "record"). Parametr "5" znamená počet prováděných iterací.

In[30]:=

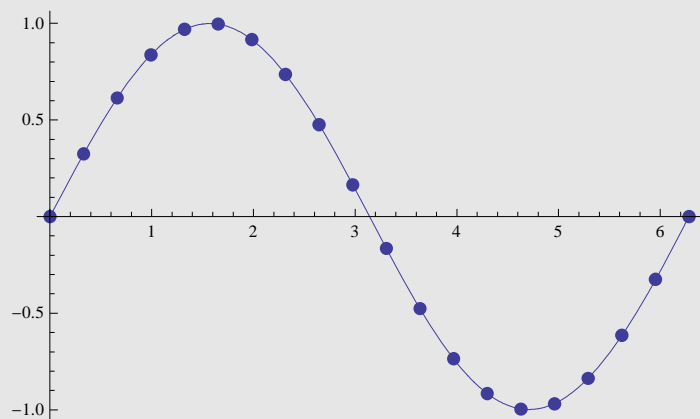
{net2, record} = NeuralFit[net, x, y, 5];

Jak teď odpovídá natrénovaná síť (reprezentováno čarou) na data (body) :

In[31]:=

```
NetPlot[net2, x, y, PlotStyle -> PointSize[0.02]]
```

Out[31]=

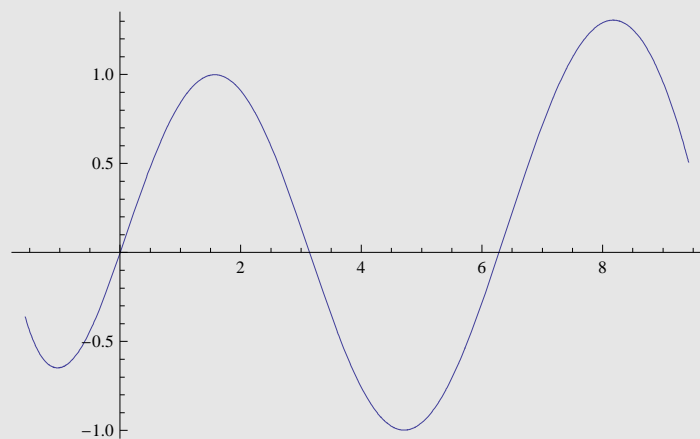


Podíváme se i mimo trénovaný interval :

In[32]:=

```
Plot[net2[{x}], {x, -π/2, 3π}]
```

Out[32]=

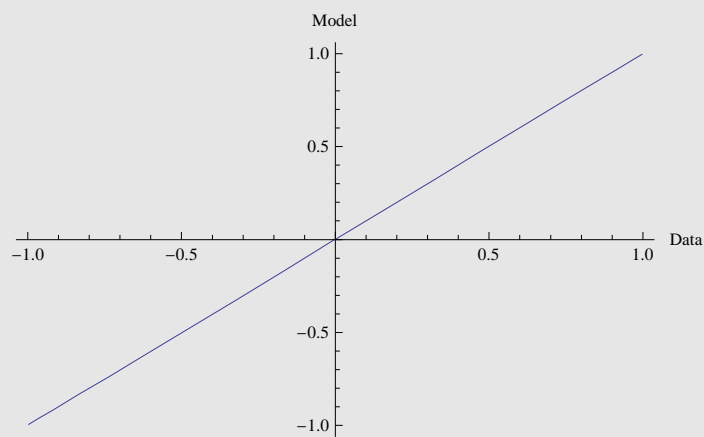


Můžeme se podívat na odpověď sítě na tzv. "data/model" diagramu - ideálně by měl být reprezentován čarou odpovídající ose 1. a 3. kvadrantu.

In[33]:=

```
NetPlot[net2, x, y, DataFormat -> NetOutput]
```

Out[33]=



Takto potom vypadá síť, když ji převedeme do vzorce - určitě poznáváte aktivační funkce neuronu.

In[34]:=

```
net2[{vstup}][[1]]
```

Out[34]=

$$36.548 - 1.12527 e^{-0.219727 (-0.909208 + vstup)^2} +$$

$$44.03 e^{-0.0687002 (-0.792339 + vstup)^2} - 77.8995 e^{-0.0332923 (0.0482398 + vstup)^2} +$$

$$0.117232 e^{-0.424755 (0.357952 + vstup)^2} + 0.054645 e^{-2.71033 (1.15471 + vstup)^2} - 3.43528 vstup$$

Prohlášení

Tento text je součástí bakalářské práce Adama Činčury “Demonstrační aplikace pro podporu kurzu neuronových sítí” na FEL ČVUT 2011. Vznikl úpravou textu Petra Chlumského.