

# Kapitola 4 - Dopřední síť a umělá data Sin(x)

*Demonstrace použití dopředné neuronové sítě na umělých datech - aproximace funkce sinus.*

## Načtení knihovny NeuralNetworks

Nejdříve načteme knihovnu neuronových sítí.

In[216]:=

```
<< 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[217]:=

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

## Vytvoření trénovacích dat

Vygenerujeme si umělá data - navzorkujeme sinusovku.

In[218]:=

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

Takto vypadají naše data :

In[221]:=

**x**

Out[221]=

```
{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[222]:=

**y**

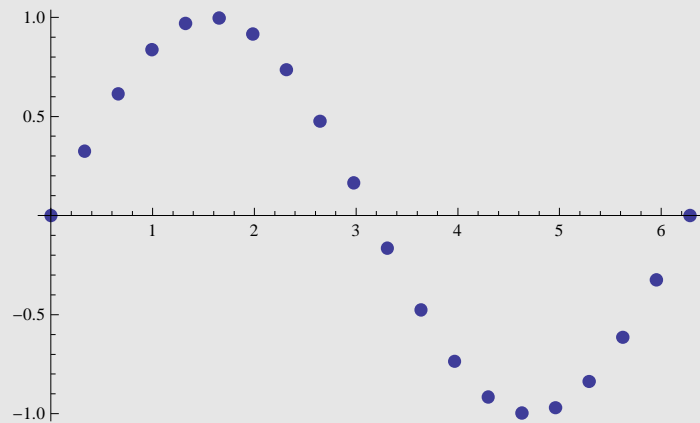
Out[222]=

```
{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[223]:=

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

Out[223]=



## Zpracování dat neuronovou sítí

Inicializace sítě - zadáme trénovací množinu, rozdělenou na vstupní a výstupní data, a počet neuronů. Počet neuronů se zadává jako seznam, každý prvek (číslo) seznamu odpovídá počtu neuronů v jedné skryté vrstvě. {3} znamená jedna skrytá vrstva s třemi neurony. {4,3} znamená dvě skryté vrstvy, jedna se čtyřmi neurony a druhá se třemi neurony.

Vytvořenou síť si uložíme do proměnné "net".

In[224]:=

```
net = InitializeFeedForwardNet[x, y, {3}, RandomInitialization -> True]
```

Out[224]=

```
FeedForwardNet[{{w1, w2}}, {Neuron -> Sigmoid, FixedParameters -> None,
  AccumulatedIterations -> 0, CreationDate -> {2011, 5, 22, 23, 47, 31.2167305},
  OutputNonlinearity -> None, NumberOfInputs -> 1}]
```

Můžeme si nechat zobrazit podrobnější informace o vytvořené síti.

In[225]:=

```
NetInformation[net]
```

Out[225]=

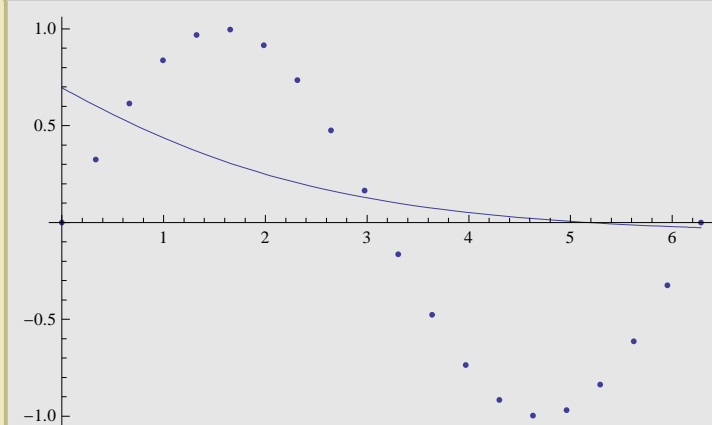
```
FeedForward network created 2011-5-22 at 23:47.
The network has 1 input and 1 output. It consists of 1 hidden
layer with 3 neurons with activation function of Sigmoid type.
```

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

In[226]:=

**NetPlot[net, x, y]**

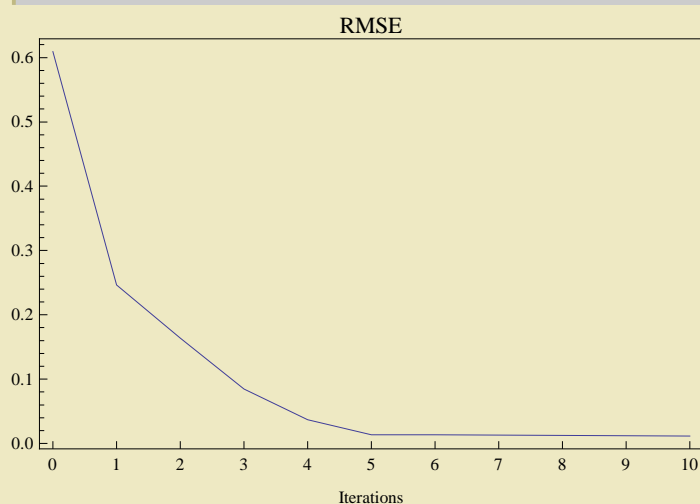
Out[226]=



Ted' síť natrénujeme pomocí funkce "NeuralFit", které zadáme naši síť (proměnná "net"), trénovací množinu "x" a "y", x jsou vstupní data, y jsou výstupní data a ještě zadáme počet učících kroků.

Funkce NeuralFit vyprodukuje naučenou síť a záznam o průběhu učení - obě tyto návratové hodnoty si ukládáme (do proměnné "net2" a "record").

In[227]:=

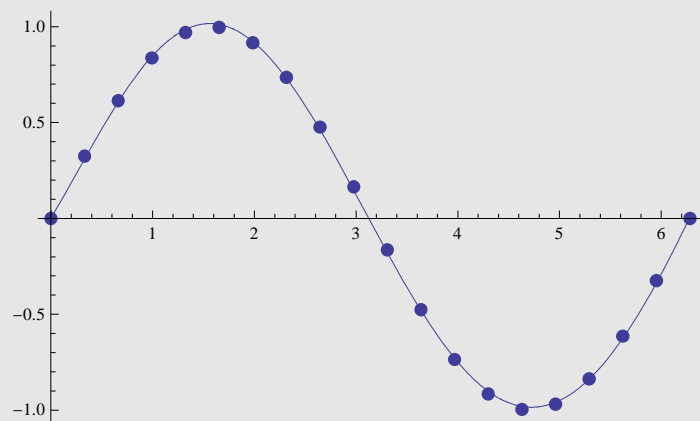
**{net2, record} = NeuralFit[net, x, y, 10];**

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

In[228]:=

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

Out[228]=

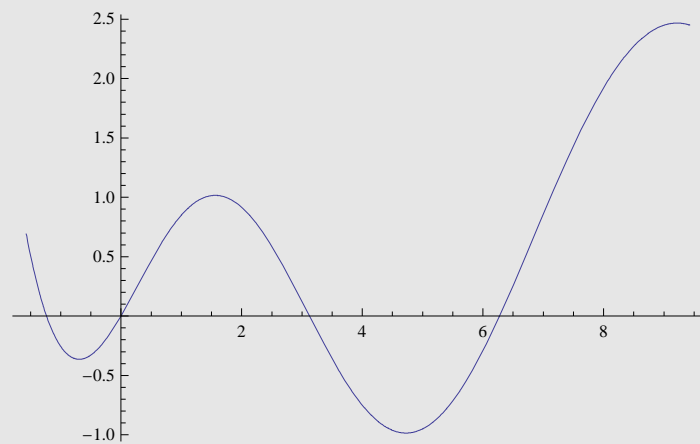


Podíváme se i mimo trénovaný interval :

In[229]:=

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

Out[229]=

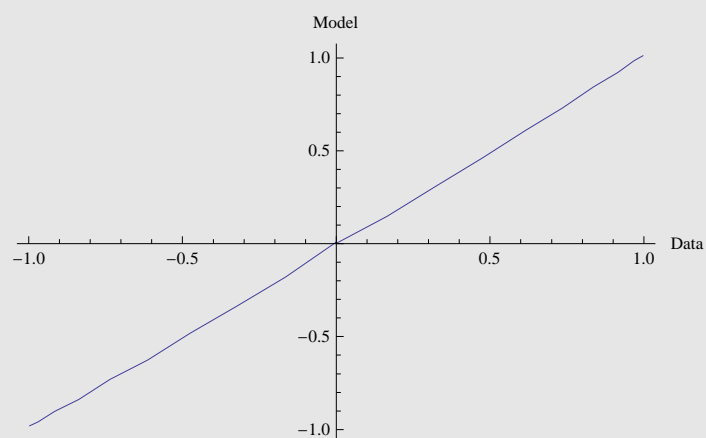


Můžeme se podívat na odpověď sítě na tzv. "data/model" diagramu - ideálně by měl být reprezentován čarou odpovídající ose 1. a 3. kvadrantu.

In[230]:=

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

Out[230]=



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

In[231]:=

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

Out[231]=

$$12903.8 - \frac{12921.2}{1 + e^{0.519832 - 0.300522 \text{ vstup}}} - \frac{9380.26}{1 + e^{-0.771328 + 0.30429 \text{ vstup}}} - \frac{3315.13}{1 + e^{-0.0165886 + 0.352074 \text{ vstup}}}$$

## 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.