

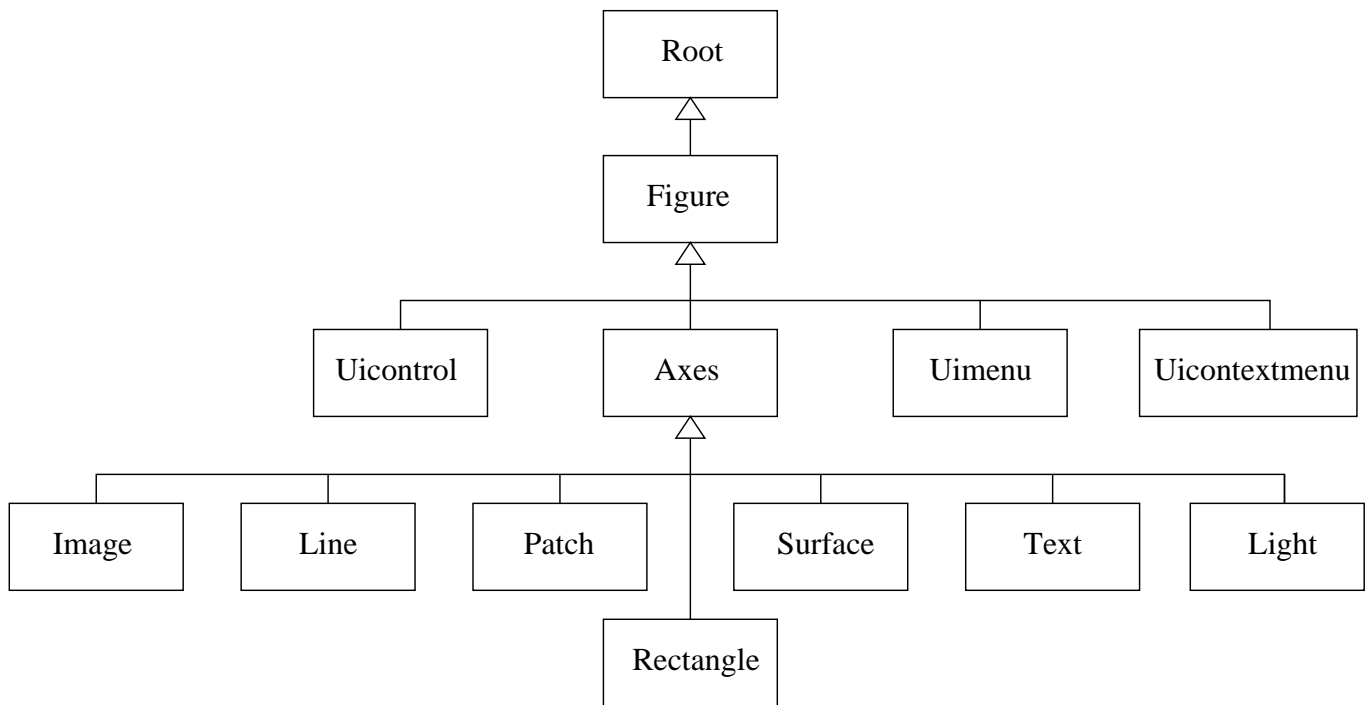
Purpose:

How to train an MLP neural network in MATLAB environment!

that is

For good computations,
we need good formulae
for good algorithms;
and good visualization
for good illustration
of good methods
and successful applications!

Hierarchy of Graphical Objects:



Basics:

- typical parent-child -like inheritance (class) hierarchy
- object handles (pointers) for accessing the properties (attributes) of objects using `get` and `set` methods
- created as a result of some drawing command or by hand:
`object('Property', Value, ...)`
`figure('PaperType', 'a4')`

Objects I:

Root:

- represents the computer screen
- unique
- has Figures as children
- (active) properties, e.g.

`CallbackObject, Children, CurrentFigure, Diary, DiaryFile, Echo, Format, PointerLocation, PointerWindow, ScreenSize, UserData, ...`

Figure: Syntax

```
figure
figure('PropertyName',PropertyValue,...)
h = figure(...)
figure(h)
```

- single drawing/GUI window (frame) on the screen
- number of figures on screen not restricted by MATLAB
- `Axis`, `Uicontrol`, `Uimenu` and `Uicontextmenu` as children
- appears automatically when e.g. `plot` command is given
- creation by hand useful (necessary) when steering different illustrations to different windows (otherwise they appear consecutively in the same frame)
- properties like

`BusyAction, ButtonDownFcn, Children, Color, Colormap, CurrentAxes, CurrentObject, HandleVisibility, KeyPressFcn, MenuBar, Name, NextPlot, PaperPosition, PaperSize, Tag, Parent, Position, WindowButtonUpFcn, ...`

Objects II:

Axes: Syntax

```
axes
axes('PropertyName',PropertyValue,...)
h = axes(...)
axes(h)
```

- defines one drawing area inside (father-)Figure
- contains the necessary attributes for defining the coordinate axis (scales, names of axis, styles, points, etc.) in 2D or 3D for lower level (child) objects Image, Line, Patch, Surface, Text, Light and Rectangle
- usually created automatically as a result of some drawing command
- properties e.g.

```
Box, ButtonDownFcn, CameraPosition, Children, Color,
DataAspectRatio, DrawMode, FontName, FontSize,
GridLineStyle, XGrid, YGrid, ZGrid, XLabel, YLabel,
ZLabel, XLim, YLim, ZLim, ...
```

Uicontrol: Dialog-object between UI-elements and their functionality (cf. *Seeheim*- and MVC-models). More precise treatment in connection with GUIs and MATLAB...

Uimenu: Definition of appearance and functionality of Figures menus, related to GUIs also...

Line:

- basic graphical primitive, whose properties define most of 2D and 3D drawings
- child of (active or activated) Axes
- created automatically with plot, plot3, contour, countour3 etc.
- in addition to similar properties than above defines, e.g.,
LineStyle (-/~/:/-/none)
LineWidth
Marker-type (+/o*/./x/□/◇/▷/◁)
...

Objects III:

Patch:

- coloured 2D or 3D polygon on active axes using corner coordinates
- e.g., as a result of commands `fill` and `fill3`

Surface:

- 3D surface presentation based on data matrices for (x_i, y_j, z_k)
- colour- (using suitable `colormap`) or line-based representation
- appears, e.g., as a result of command `pcolor`, `surf` and `mesh`
- properties include

`CData`, `CDataMapping`, `CreateFcn`, `EdgeColor`, `FaceColor`,
`MeshStyle`, ...

Image:

- graphical object that either
 - interpolates defining matrix using the given `colormap` (*indexed image*)
 - represents directly the given RGB-values (*truecolor*).
- every Image-object has own `colormap` (*palet*), that defines the used colors
- basic type for representing and storing pictures *8-bit unsigned integer* (`uint8`)
- contrary to Surface-object all Image-objects are two-dimensional so that the view point is always along with the *z*-axis
- coordinate system defined by the father-Axes
- creation by hand using `image`; other useful commands, e.g., `imread`, `imwrite`, and `imfinfo` that allow reading, writing, and inspection of different basic formats (e.g., 'bmp', 'hdf', 'jpg'/'jpeg', 'pcx', 'tif'/'tiff', 'xwd')
- properties include

`CData`, `CDataMapping`, `EraseMode`, `Selected`, `Tag`,
`UserData`, ...

Objects IV:

Text:

- addition of texts (with various styles) for suitable places on Axes
- e.g., names of coordinate axis, title of figure, legends etc.
- properties include

Color, Extent, FontAngle, Fontname, Fontsize, HorizontalAlignment, Interpreter, Position, Rotation, String, VerticalAlignment, ...

Light:

- defines the lightning model for Patch- and Surface objects (to learn more, take a course on *Computer graphics* by Tuomo R.)
- properties include Style, Position, ...

To Learn more on objects (especially on Uicontextmenu, Rectangle etc.) go to

<http://www.mathworks.com/access/helpdesk/help/techdoc/matlab.shtml> -> Graphics

Predefined handles:

gcf (*Graphical Current Figure*): handle to active Figure
(stored in Roots CurrentFigure)

gca (*Graphical Current Axes*): active Figures active Axes
(stored in Figures CurrentAxes)

gco (*Graphical Current Object*): currently active graphical object

For manipulation e.g. findobj, copyobj and delete.

Unique identity using Tag

Interactive editing using *Property Editor*, command propedit or propedit(h) with suitable graphical handle

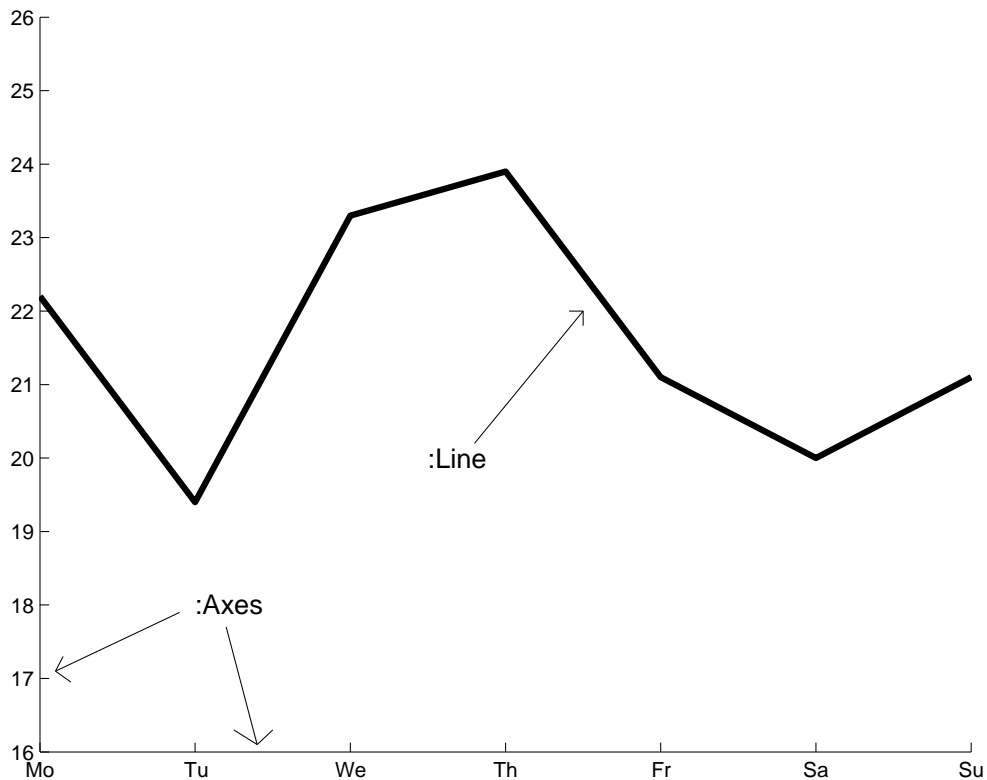
Objects/Graphics V:

- *Tick Marks* for x -, y - and z -axes: `xtick`, `ytick`, `ztick`
- ratio between different axis (*Aspect ratio*):
`axis normal/square/equal/tight`
- clarifying texts
`title`, `xlabel`, `ylabel`, `zlabel`, `legend`, `text`, `gtext`
- predefined colors

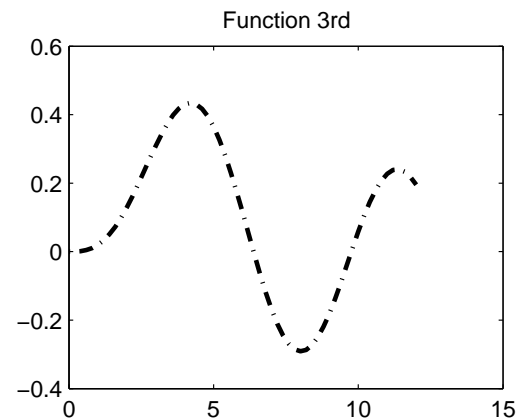
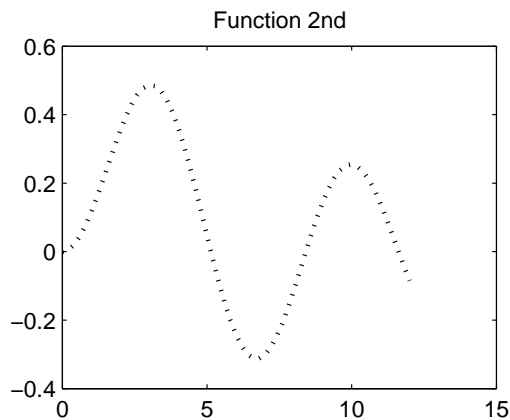
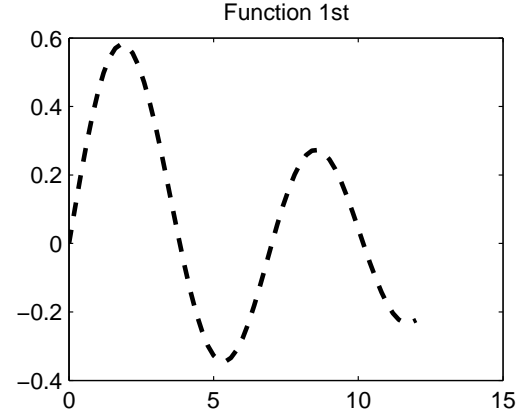
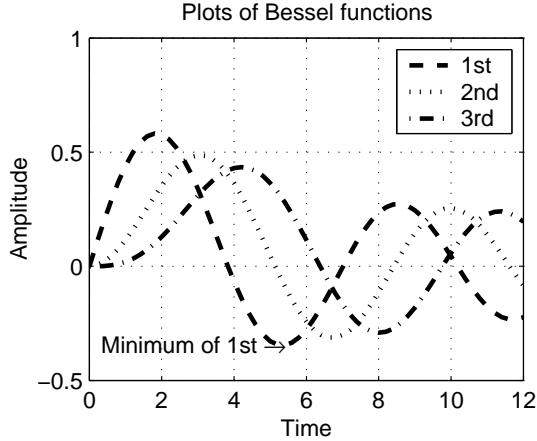
RGB-value	Abbreviation	Color
[0 0 0]	k	<i>black</i>
[0 0 1]	b	<i>blue</i>
[0 1 0]	g	<i>green</i>
[0 1 1]	c	<i>cyan</i>
[1 0 0]	r	<i>red</i>
[1 0 1]	m	<i>magenta</i>
[1 1 0]	y	<i>yellow</i>
[1 1 1]	w	<i>white</i>

- and much more...

First Example:



```
clear; close all;
days = ['Mo'; 'Tu'; 'We'; 'Th'; 'Fr'; 'Sa'; 'Su'];
temp = [22.2 19.4 23.3 23.9 21.1 20 21.1];
f = figure;
a = axes('YLim',[16 26],'Xtick',1:7,'XTickLabel',days);
h = line(1:7,temp);
set(h,'Color',[0 .8 .8],'LineWidth',3);
text(2,18,':Axes','FontSize',13)
text(3.5,20,':Line','FontSize',13)
%grid on
hold on
plot([1.1 1.9],[17.1 17.9],'-');
plot([1.1 1.2],[17.1 16.95],'-');
plot([1.1 1.15],[17.1 17.3],'-');
plot([2.2 2.4],[17.7 16.1],'-');
plot([2.4 2.25],[16.1 16.3],'-');
plot([2.4 2.5],[16.1 16.3],'-');
...
hold off
print -deps graf_ex.eps
close all
```

```

x = 0:.2:12;
y1 = bessell(1,x); y2 = bessell(2,x); y3 = bessell(3,x);
%
figure(1); subplot(2,2,1); h = plot(x,y1,x,y2,x,y3);
set(h,'LineWidth',2,{'LineStyle'},{'--';':';'-.'});
set(h,{'Color'},{'r';'g';'b'});
axis([0 12 -.5 1]); set(gca,'XTick',[0 2 4 6 8 10 12]);
%
grid on
xlabel('Time'); ylabel('Amplitude');
legend(h,'1st','2nd','3rd');
title('Plots of Bessel functions');
[y,ix] = min(y1);
text(x(ix),y,'Minimum of 1st \rightarrow',...
'HorizontalAlignment','right')
%
subplot(2,2,2)
plot(x,y1,'r--','LineWidth',2); title('Function 1st');
subplot(2,2,3);
plot(x,y2,'g:', 'LineWidth',2); title('Function 2nd');
subplot(2,2,4)
plot(x,y3,'b-.','LineWidth',2); title('Function 3rd');
%
print -deps fig1.eps

```