6.6 Supervised Learning

- Supervised learning is typically used for multilayer perceptrons to approximate complex nonlinear mappings.
- In general, it is possible to show that two layers are sufficient to approximate any nonlinear function.
- Therefore, we restrict our considerations to such two-layer networks.
- The structure of the decoding part of the two-layer back-propagation network is presented in Figure (6–4).

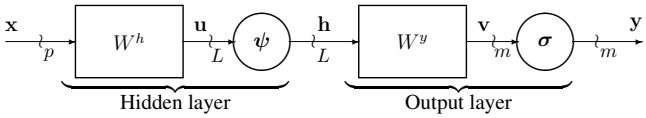
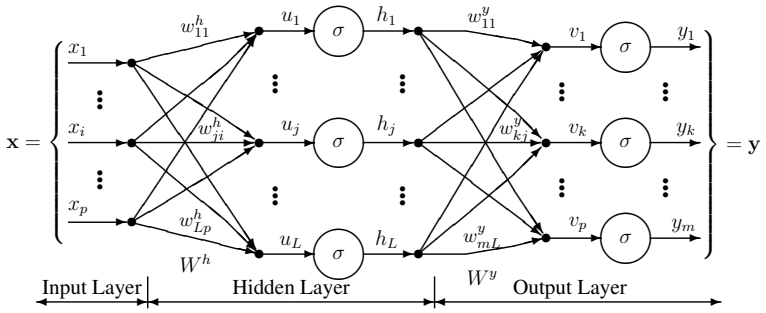


Figure 6–4: A block-diagram of a single-hidden-layer feedforward neural network

- The structure of each layer has been discussed in sec. 4.4.
- Nonlinear functions used in the hidden layer and in the output layer can be different.
- The output activation function can be linear.
- There are two weight matrices: an $L \times p$ matrix W^h in the hidden layer, and an $m \times L$ matrix W^y in the output layer.

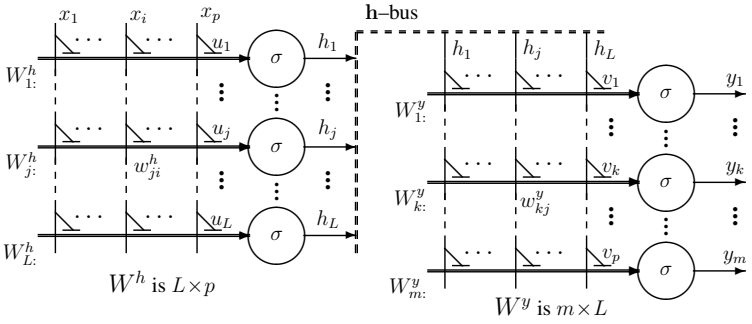
6.6.1 Detailed structure of a Two-Layer Perceptron — the most commonly used feedforward neural network

Signal-flow diagram:



$$u_j = W^h_{j:} \cdot \mathbf{x}\,; \quad h_j = \sigma(u_j)\,; \quad v_k = W^y_{k:} \cdot \mathbf{h}\,; \quad y_k = \sigma(v_k)$$

Dendritic diagram:



We can have:

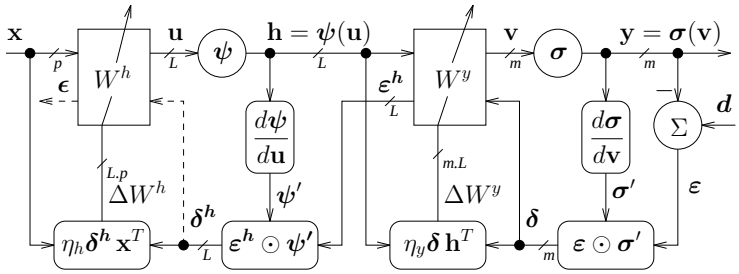
$$x_p = 1$$

and possibly

$$h_L = 1$$

Figure 6–5: Various representations of a Two-Layer Perceptron

6.6.2 The structure of the two-layer back-propagation network with learning



- The structure of the decoding and encoding parts of the two-layer back-propagation network:
- Note the blocks which calculate derivatives, delta signals and the weight update signals.
- The process of computing the signals in the basic **back-propagation algorithm** (pattern mode) during each time step consists of the:

  forward pass in which the signals of the decoding part are determined starting from $\mathbf{x}$, through
  $\mathbf{u} = W^h\mathbf{x}$, $\mathbf{h} = \boldsymbol{\psi}(\mathbf{u})$, $\mathbf{v} = W^y\mathbf{h}$, $\mathbf{y} = \boldsymbol{\sigma}(\mathbf{v})$, and derivatives $\boldsymbol{\psi'}$, and $\boldsymbol{\sigma'}$.

  backward pass in which the signals of the learning part are determined starting from $\mathbf{d}$, through
  $\boldsymbol{\varepsilon} = \mathbf{d} - \mathbf{y}$, $\boldsymbol{\delta} = \boldsymbol{\varepsilon} \odot \boldsymbol{\sigma'}$, $\Delta W^y = \eta_y \boldsymbol{\delta} \mathbf{h}^T$, $\boldsymbol{\varepsilon^h} = W^{yT}\boldsymbol{\delta^h}$ (**back-propagated error**), and
  $\boldsymbol{\delta}^h = \boldsymbol{\varepsilon}^h \odot \boldsymbol{\psi'}$, $\Delta W^h = \eta_h \boldsymbol{\delta^h} \mathbf{x}^T$
- Note that, in general, the weight updates are proportional to the synaptic input signals ($\boldsymbol{x}$, or $\boldsymbol{h}$) and the delta signals ($\boldsymbol{\delta^h}$, or $\boldsymbol{\delta}$).
- The delta signals, in turn, are proportional to the derivatives the activation functions, $\boldsymbol{\psi'}$, or $\boldsymbol{\sigma'}$.

6.7 Model of eye movement control

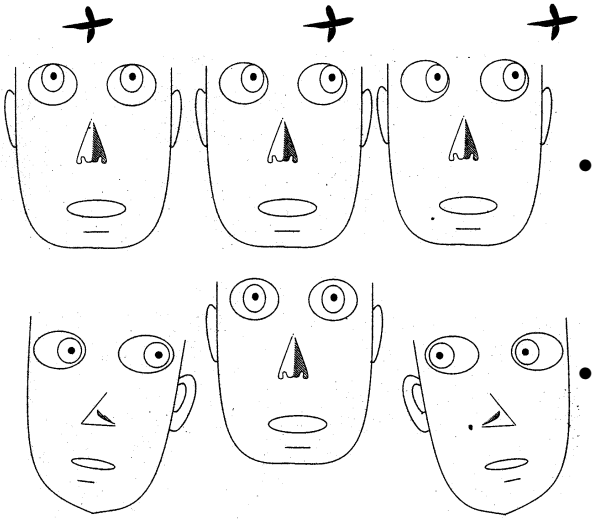
based on sec. 9.6 *Distributed representation in eye movement control.* from Lytton: *From Computer to Brain ...*

- Multi-layer perceptrons, often called back-propagation networks have been used as a tool to model and understand the complex structure of sensory and motor systems.
- Here we look at the mixture of sensory inputs that help **control eye movement**.
- There are several things that cause your eyes to move.
- Fast movements to look at something that has attracted your attention are called saccades.
- Slow movements are used to follow a moving visual target, for example a duck flying overhead. This is called **visual pursuit**.
- There is another type of slow movement that is used for eye stabililization. This is called the **vestibulo-ocular reflex** (VOR).
- Pursuit is based on the movement of an external visual target.

  The VOR is based on the viewer's own movement.
- Although apparent movement of the entire visual field is also used to stabilize the eyes, the major input for the VOR is an inertial sensing mechanism called the **semicircular canals**, located together with the sound sensing organs in the ears.
- The use of inertial sensing means that the VOR works in the dark.

- (The closeness of visual and inertial sensation also accounts for the fact that retinal slip, the movement of the entire visual field, produces the sensation of movement, as for example when a train next to yours leaves the station.)

- The VOR compensates for head movement that could otherwise cause the image to disappear off of the retina.
- Both the VOR and pursuit operate simultaneously, as for example when you watch a duck fly by, while sitting in amoving boat.
- Signals from both eye and ear (semicircular canals) influence the muscles controlling eye movement



- Visual pursuit
- VOR

- The neurons that mediate eye movements are in the brainstem, a place where it is relatively hard to record neuronal activity.
- Before any such recordings had been made, researchers had already been constructing models of eye movement control, based largely on engineering principles.

- They generally assumed that the neuronal systems for pursuit and VOR systems would remain separate until they converged on the muscles of the eye.
- When brainstem recordings were finally made, people were surprised to find that many neurons could not be strictly defined as “pursuit cells” or “VOR cells” but had a mixed response to both visual and movement input.
- Even more surprising was that there were some cells where pursuit inputs were pulling in one direction and VOR inputs were pushing in the opposite direction.
- Tom Anastasio and David Robinson were two researchers who looked into this by applying the then-new back-prop algorithm for two-layer perceptron

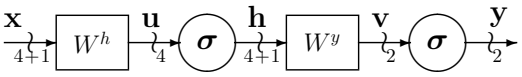
  to the problem of how pursuit and VOR inputs might distribute the information through a bunch of interneurons (hidden units)

  and still be able to produce the correct effects on the eye.
- This represented a different approach to understanding a neural circuit.
- Instead of using top-down ideas to figure out how the system might work, their model was allowed to grow from the bottom up and was then analyzed by the researchers and compared to the real thing.
- This had some appeal since the nervous system has to develop from simpler rules that cannot take account of how the finished brain ought to look.

6.7.1 The two-layer perceptron model

A version of the model can consist of a two-layer feedforward network with:

$n = 4 + 1$ **input** signals $\mathbf{x} = [x_1\ x_2\ x_3\ x_4\ x_5 = 1]$
$L = 4 + 1$ **hidden** signals $\mathbf{h} = [h_1\ h_2\ h_3\ h_4\ h_5 = 1]$
$m = 2$ **output** signals $\mathbf{y} = [y_1\ y_2]$



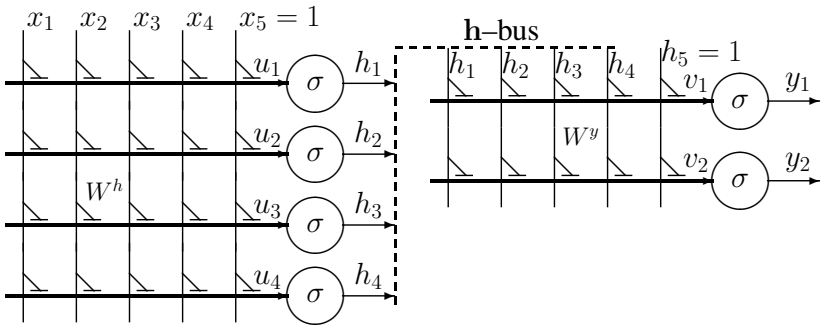
Additional, constant input and hidden signals are called the **bias** signals and are important for the proper working of the network. The sigmoidal activation function $\sigma(v) = \dfrac{1}{1 + e^{-v}}$

Input and output signals are interpreted as follows:

x_1 — left VOR
x_2 — left pursuit
x_3 — right pursuit
x_4 — right VOR

y_1 — left eye
y_2 — right eye

Detailed dendritic diagram



Note that the input matrix W^h is 4×5 and the output matrix W^y is 2×5

The vector of output signals can be calculated as: $\mathbf{y} = \boldsymbol{\sigma}(W^y \cdot \boldsymbol{\sigma}(W^h \cdot \mathbf{x}))$

- The **training input/output pairs** are simple in principle, but keeping track of crossing influences pushing left and right makes it confusing.
- Left pursuit (visual target moving to left) makes the eyes move left, while left VOR (head move to the left) makes the eyes move right.
- Here are the basic input/output mappings in the context of watching a duck flying by.
- There are $K = 4$ training pairs $\mathbf{x}(k); \mathbf{d}(k)$ for $k = 1, \ldots, 4$, where $\mathbf{d}(k)$ is a desired value of the output signal.
- One training pair is stored in a column of matrices X and D, respectively.

$$k = \begin{matrix} \text{h R} & \text{h L} & \text{d L} & \text{d R} \\ 1 & 2 & 3 & 4 \end{matrix}$$

$$X = \begin{bmatrix} 0 & 1 & 0.5 & 0.5 \\ 0.5 & 0.5 & 1 & 0 \\ 0.5 & 0.5 & 0.0 & 1 \\ 1 & 0 & 0.5 & 0.5 \end{bmatrix} \begin{matrix} \text{left VOR} \\ \text{left pursuit} \\ \text{right pursuit} \\ \text{right VOR} \end{matrix}$$

$$D = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix} \begin{matrix} \text{left eye} \\ \text{right eye} \end{matrix}$$

- 0.5 is the value of spontaneous activity for a unit; 1 means full activation; 0 means that the unit is being inhibited.
- Note that each of the training patterns activates only one neuronal output that activates relevant eye muscles.
- For example, the 1 in the first column indicates that head turning right (h R) activates the right VOR input and movement of the eyes left.

- This mapping is easiest to understand if you imagine your own head moving or your eyes tracking a duck.
- Pattern $k = 1$ is pure right VOR.
  This means that the head moves right so that the right VOR unit is activated (1) and the left VOR unit is inhibited (0).
  The pursuit units remain at their resting activation levels (0.5)..
  The output is full activation (1) of the muscles moving the eyes to the left and full inhibition (0) of the muscles moving the eyes to the right.
  Head goes right; eyes go left.

$$
\begin{array}{rcc}
 & \text{h R}\ \ \text{h L}\ \ \text{d L}\ \ \text{d R} & \\
k = & 1 \quad 2 \quad 3 \quad 4 & \\
X = & \begin{bmatrix} 0 & 1 & 0.5 & 0.5 \\ 0.5 & 0.5 & 1 & 0 \\ 0.5 & 0.5 & 0.0 & 1 \\ 1 & 0 & 0.5 & 0.5 \end{bmatrix} & \begin{array}{l}\text{left VOR}\\ \text{left pursuit}\\ \text{right pursuit}\\ \text{right VOR}\end{array} \\
D = & \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix} & \begin{array}{l}\text{left eye}\\ \text{right eye}\end{array}
\end{array}
$$

- Similarly, pattern $k = 2$ is pure left VOR.
- Pattern $k = 3$ is pure left pursuit. The duck goes left. Left pursuit input is activated and right inhibited.
  VOR inputs both stay at rest. Left eye muscles are activated and right inhibited.
  Duck goes left, eyes go left.
- In pattern $k = 4$, duck goes right, eyes go right.

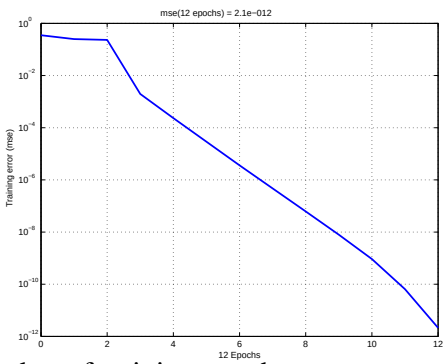
6.7.2 Learning procedure

- In order to obtain synaptic matrices W^h and W^y that implement required input/output mapping we present the input/output patterns to the network,
  calculate the error between the desired output $\mathbf{d}(k)$ and the real output $\mathbf{y}(k)$ generated by the network, and
  update weights according to an applied learning law (algorithm).
- The convergence is monitored by calculating the **mean squared error**.
- There are a number of learning algorithms available, the slowest is the basic back-propagation algorithm presented in p. 6–44, the fastest (but complicated) is the Levenberg-Marquardt algorithm that I have used to obtain the following weights (after rounding):

```
Wh =  4    -4     5    -4    -1
      3    -2     0    -1    -2
      5     6     6    21    24
     -6     7    -5     5     0

Wy = -6   -11    -8    23     1
      4    19     6   -21    -2
                          onek = ones(1,K)
Y = logsig(Wy*[logsig(Wh*[X; onek]) ; onek])

  1.0000    0.0000    1.0000    0.0000
  0.0000    1.0000    0.0000    1.0000
```



Note that we have obtained a perfect mapping after a small number of training epochs.