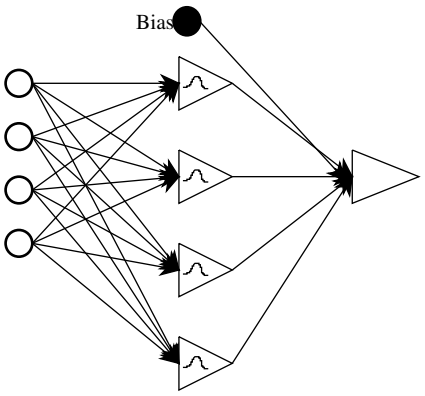
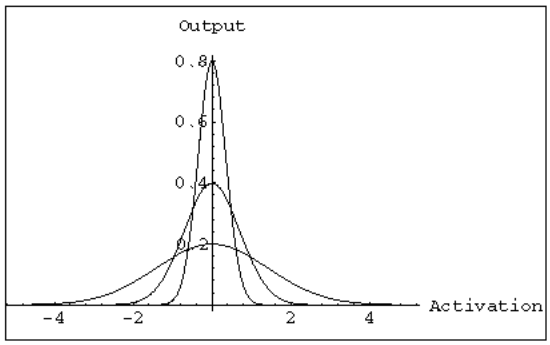
Radial Basis Functions.

A different approach to the use of feedforward neural networks. Though the architecture looks like a standard 3-layer BP network, there is a crucial difference.



The difference is that the hidden units embody a different type of neuron altogether. These units have a *receptive field*: that is, they have their maximal response (generally 1) for some particular point in the input space, and this response tails off as the input value moves away from this point.

One can show such a response for a 1-dimensional input space:



In general, Gaussian units are used: that is the unit has a Gaussian response function.

Each hidden unit, h, computes

Gauss((**I** - $\mathbf{I}_h$,$\mathbf{r}_h$), where

I is the input vector
$\mathbf{I}_h$ is the centre of the receptive field of this hidden unit
$\mathbf{r}_h$ is the radius of the Gaussian.

Gauss(**0**, r) =1, and its value tails away to 0 as the first argument increases in distance away from 0. The speed at which it tends to 0 will depend on **r**.

The output unit (there may be many: only one is shown here, for clarity), may be a simple linear unit, or it may be a logistic unit or a tanh unit.

Notes:

1: Each RBF in the hidden layer responds to input only in some subspace of the input space. When the input is far away from it own centre, $\mathbf{I}_h$, (many radii (standard deviations) away), then the output of that unit will be so small as to be ignorable.

2: The hidden layer recodes the input space using the RBF outputs. Each RBF has a receptive field: that is, an area of the input space to which it responds. Not only is this more biologically plausible (since many sensory neurons respond only to some small subspace of the input space, and are silent in response to all other inputs), but it also allows us to have large numbers of units in the hidden layer without adding (much) to the complexity of the total mapping.

3: There is considerable evidence that neurons in the visual cortex display ***receptive fields*** of this form: that is, they are maximally sensitive to some specific stimulus, and their output falls off as the presented stimulus moves away from this "best" stimulus.

Training the Radial Basis Function system.

To train the system, we need to select:

$\mathbf{I}_h$ for each hidden unit (the centres of the RBFs)
$\mathbf{r}_h$ for each hidden unit (the radii of the RBFs)
W (the weight vector from the RBF layer to the output unit) for each output unit.

There are a number of techniques used:

1: Simplest technique.

Choose the $\mathbf{I}_h$ to coincide with the different training inputs.
Choose $\mathbf{r}_h$ = r (i.e. fixed for all the units, and the same in all directions) so that

$$r = \frac{d}{\sqrt{2M}}$$

where d is the maximal distance between the $\mathbf{I}_h$, and M is the number of RBF units.

2: Cluster the inputs first:

If there is a great deal of training data, using 1 per RBF unit may be impossible. Instead, one can choose M points in the input space, selecting them so that they cover the same region as the input.

After this, one simply trains the **W** using the delta rule.

This type of training clearly takes place in two distinct steps: choosing the centres and radii of the RBF layer, then training up the weights.
The way in which the centres are chosen may involve the use of a *clustering technique,* of which more later.

2: Supervised technique.

In this technique, the RBF centres and radii are initially selected in a similar way to the above (clustered), and the output layer of weights trained.

Then, one attempts to alter the centers and the radii using a gradient descent technique to "fine-tune" the centres, radii, and weights.

Precisely how to select the centres and radii is still an area of development.

RBF's do offer an effective alternative to BP: with the more sophisticated training system, their results are certainly comparable to those of BP.

RBFs , BP and Explanations.

BP:

It is difficulty to provide an explanation for why BP systems perform a particular classification.

All the rules are implicit: inside the network.

(Telling a manager or client that *it's all in the weights* is entirely useless)

RBF:

One can see why the RBF has selected the centres for the RBFs. Generally, these provide the centres of the different clusters of input to the network.

In this way, it is easier to get an explanation for the behaviour of an RBF system than it is for a BP system.

The values of the hidden (RBF) units can be explained in terns of the location of that particular input with respect to the general training set input. The weights are single layer, and can be seen as weighting the hidden unit outputs.

Notes:

RBF's are usable for both classification and for mapping, in the same way as BP systems are. The concepts of training them and testing them is precisely the same.

The RBF technique has a good mathematical background based in the theory of functional approximation. For more information, see Haykin, chapter 7 (not for the faint of heart!).

Note that if $\mathbf{r}_h$ is small, then one is really training up the system just to use each RBF unit for just one input in the training set. Thus, one can get a lack of generalisation, just as one can in BP. If $\mathbf{r}_h$ is large, one can get overgeneralisation.
Using the supervised learning algorithm to choose the centres and the radii can optimise performance.

If one knows that a problem is particularly sensitive to some small area of the input space

-- or if one is having perticular problems with data from some part of the input space

one can choose to place an RBF unit with centre there: thus *domain knowledge* can be used to guide network construction.

BP, RBFs and Fuzzy Sets/ Fuzzy Logic.

Fuzzy set theory is the theory of sets in which elements are members of the set with a set-membership function, whihc has a value between 0 and 1, where
0 means definately NOT a member
1 means definately a member
and intermediate values have intermediate meanings.

A BP output unit with a logistic output could be interpreted as a set-membership function: the output could be thought of as the set-membership function for some set defined by a predicate implied by the network itself.
Another interpretation would be to use the actual level of the output as the srength of the networks conviction in some decision.

RBF units have outputs between 0 and 1. Again this can be considered as encoding fuzzy-set information.

If the output of the RBF network is also between 0 and 1 (as would be the case for a logictic unit), the output could be interpreted in the same way as that for the BP network.