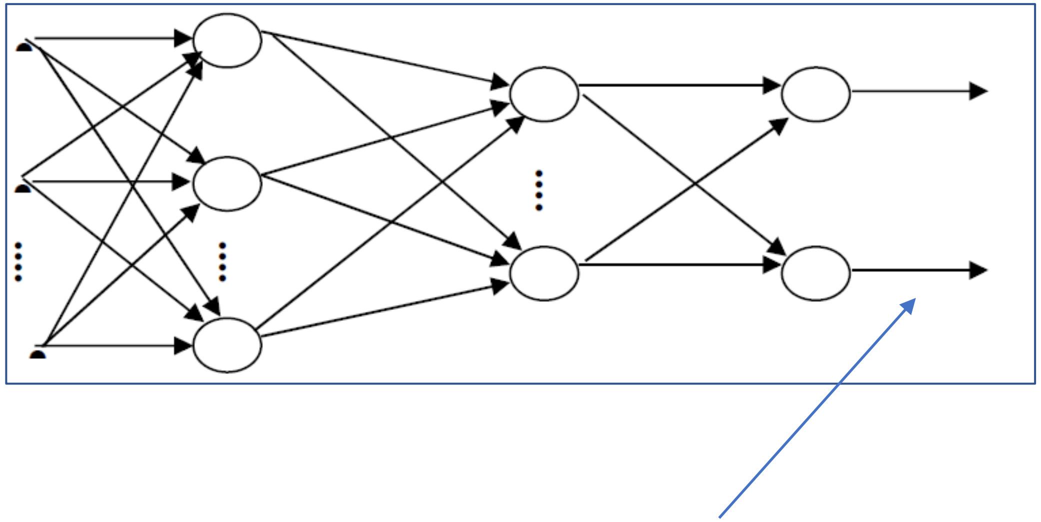
MLP OUTPUT REPRESENTATION CLASSIFICATION



Target Value representation:

For two class problem can use single output 0 or 1
or two outputs [1 0] [0 1]

For three classes use [1 0 0] [0 1 0] [0 0 1]

…………

BACK-PROPAGATION COST FUNCTION

For classification problems a good alternative to mean squared error is cross-entropy

BUT performance should be measured using classification error rate

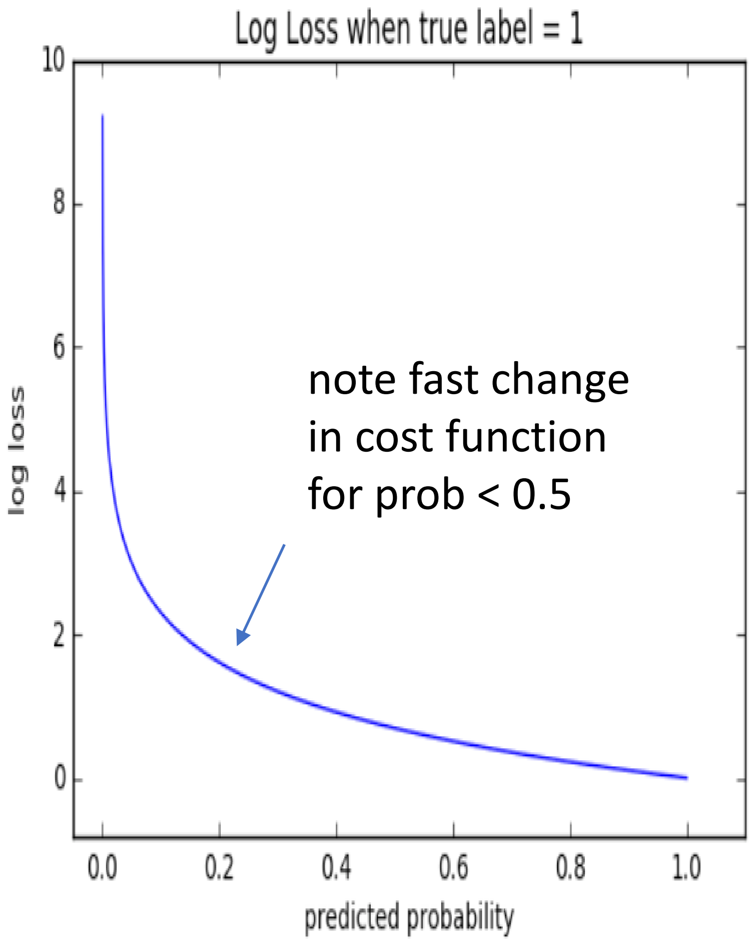
crossentropy in matlab (default performance function in patternet.m)

perf = crossentropy(net,targets,outputs). The function returns a result that heavily penalizes outputs that are extremely inaccurate (y near 1-t), with little penalty for fairly correct classifications (y near t). y=output t=target

Cross-entropy for each pair of output-target elements is ce = -t .* log(y). Special case (N = 1): If an output consists of only one element, then the outputs and targets are interpreted as binary encoding. That is, there are two classes with targets of 0 and 1, whereas in 1-of-N encoding, there are two or more classes. The binary cross-entropy expression is:

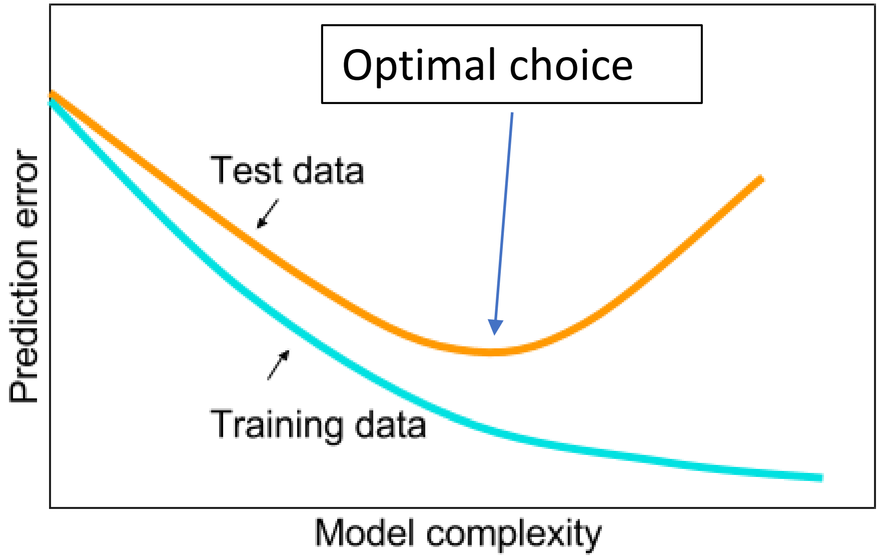
ce = -t .* log(y) - (1-t) .* log(1-y) .

The target matrix columns consist of all zeros and a single 1 in the position of the class being represented by that column vector. When N = 1, the software uses cross entropy for binary encoding, otherwise it uses cross entropy for 1-of-N encoding. e.g. [0 0 1] for class 3.



note fast change in cost function for prob < 0.5

HOW TO AVOID OVERFITTING EXP 1)



There are many ways to avoid overfitting and choosing optimal complexity – known as regularisation methods

Two simple ways for MLP

- changing the number of hidden nodes
- changing number of training epochs, that is early stopping (before convergence, and typically with zero error as convergence criteria)

MULTIPLE CLASSIFIER SYSTEMS (ENSEMBLE CLASSIFIERS) EXP 2)

- Conventional approach
  - design diverse classifiers
  - select the best, but
    - different classifiers often misclassify different samples

- Multiple classifier system approach
  - Also called ensemble method/multiple expert/fusion
  - combine designs to achieve greater accuracy

Note: accuracy/diversity trade-off – if all classifiers are same, no advantage in combining

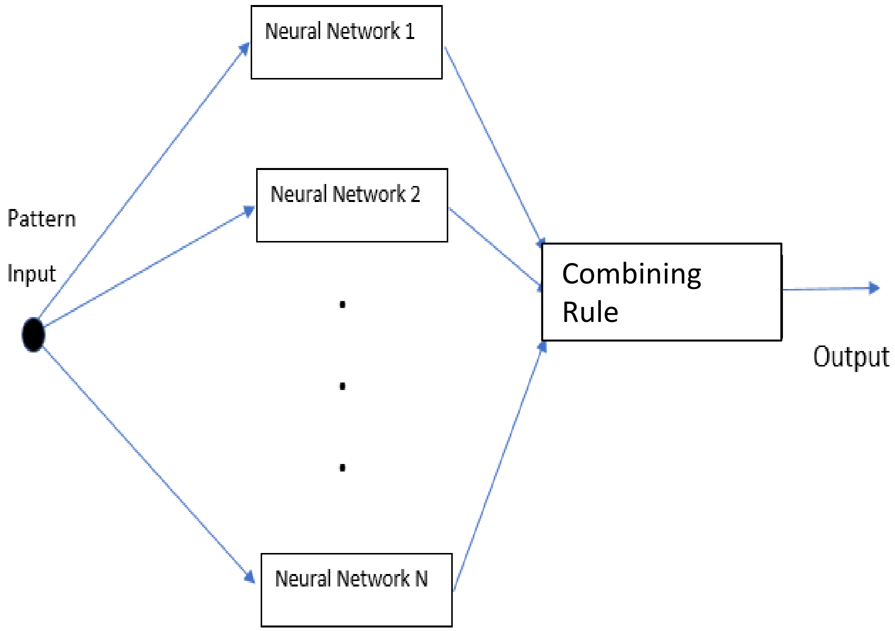
Many ways to diversify classifiers

– e.g. randomise feature space, training set, classifier

Many ways to combine

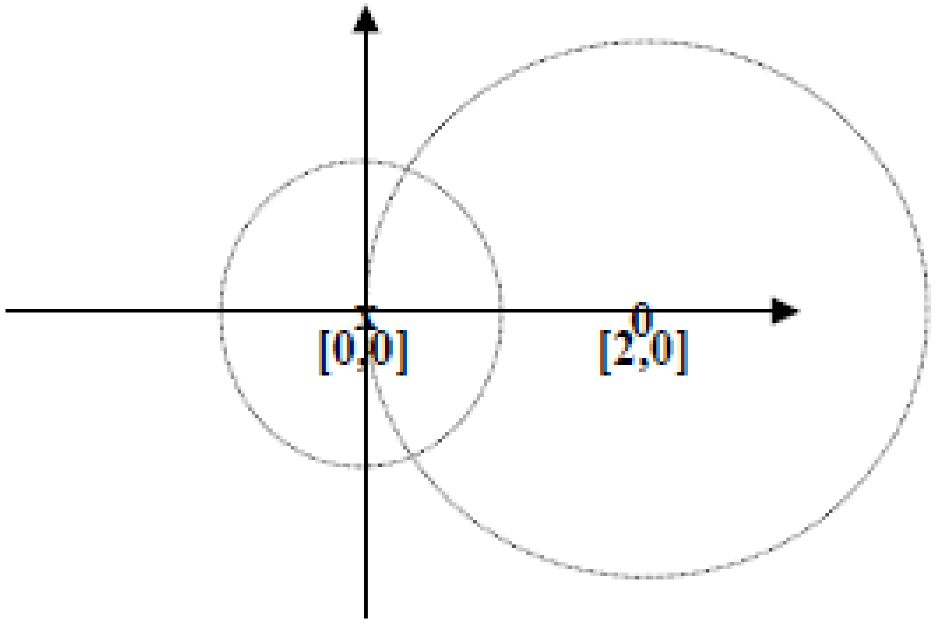
For Neural Networks a simple strategy is to use different starting weights and combine with majority vote (for two class requires > 50 percent of votes)

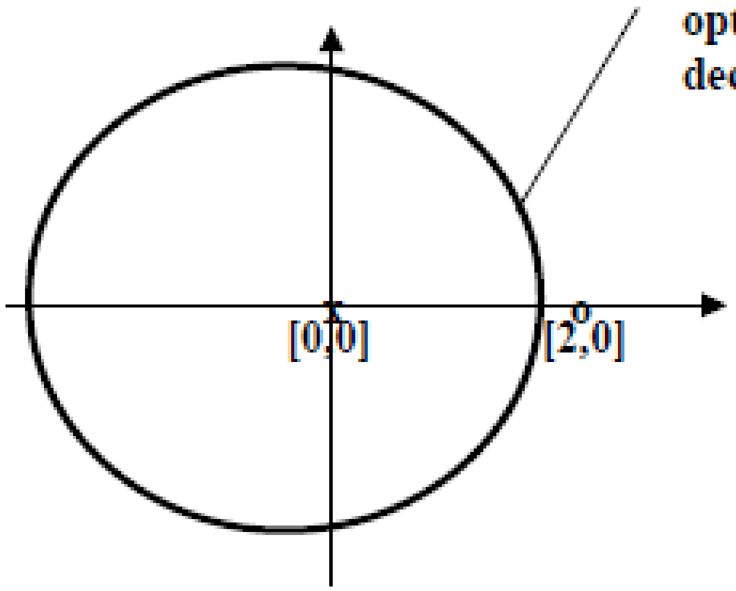
Architecture – can be considered form of regularisation



e.g. 2D 2-class overlapping Gaussian

x mean [0,0] variance 1 **o** mean [2,0] variance 4





optimum (Bayes)
decision boundary