

Support Vector Machines

When these machines are best known is in handwriting recognition.

Good tutorial – look at

<http://guppy.mpe.nus.edu.sg/burges.ps>

Currently, we have two types of neural nets.

One layer perceptrons – learning algorithm easy, drawback such network can't learn much.

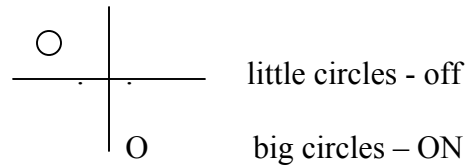
Multilayer networks – learning rule harder

Want some way to transform harder problems to be solved by one layer networks.

What would an example transformation look like?

(points have radius < 1)

(point have radius > 1)



we want to be able to separate these two sets of examples

We could try mapping: $F(X) = (X_1^2, X_2^2)$

So we can reduce to perceptron case if know how to choose F wisely (F is the feature space)

In general, feature space maps work best if map from original number of dimensions.

Minimal number of dimension should map to called VC-dimension.

Example: From above problem, we could have used map $F = (X_1^2, X_2^2, (\sqrt{2})x_1x_2)$

What's the maximum number of dimensions you could need to separate data via a hyperplane.

Turns out need at most an $N-1$ dimensional space to separate N data points.

SVM's are also called Kernel machines

If stayed in same number of dimensions and we had no feature space map, then the optimal linear separator of the on vs not on set is given by minimizing $\sum \alpha_i - \frac{1}{2} \sum \alpha_i \alpha_j y_i y_j (X_i \cdot X_j)$

Optimal linear separator minimizes these distances.