

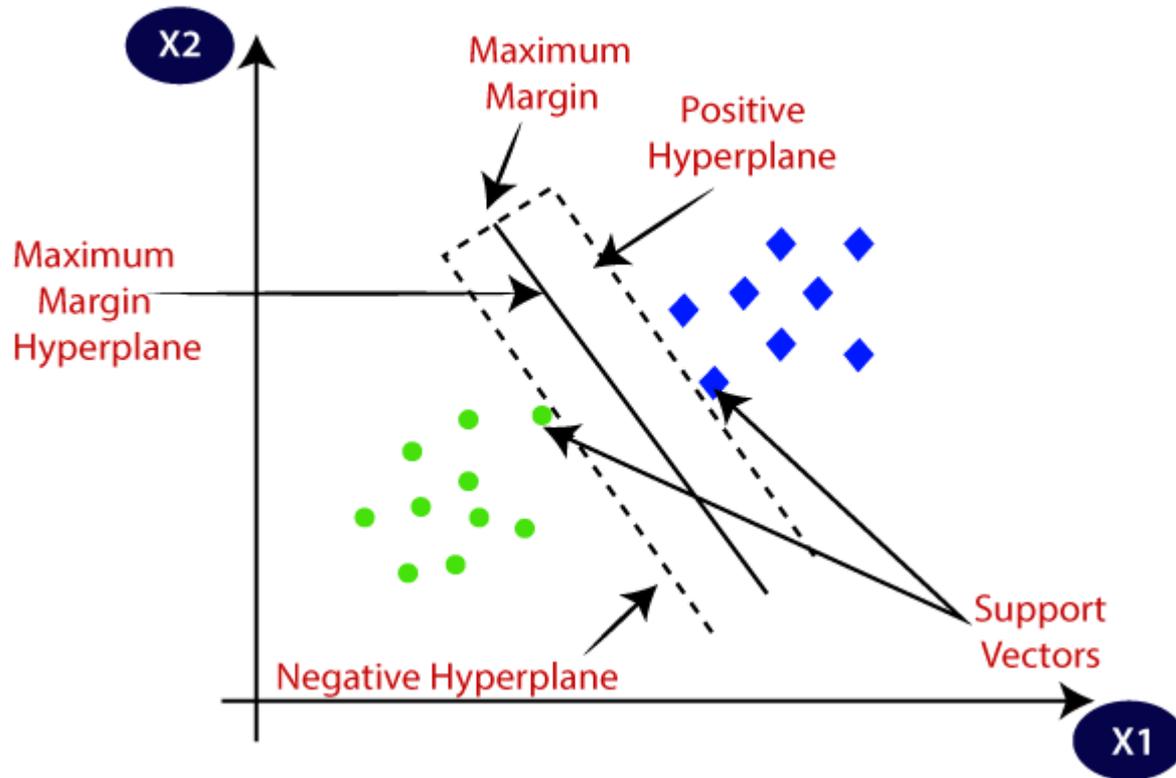
Support Vector Machines

Support Vector Machine Algorithm

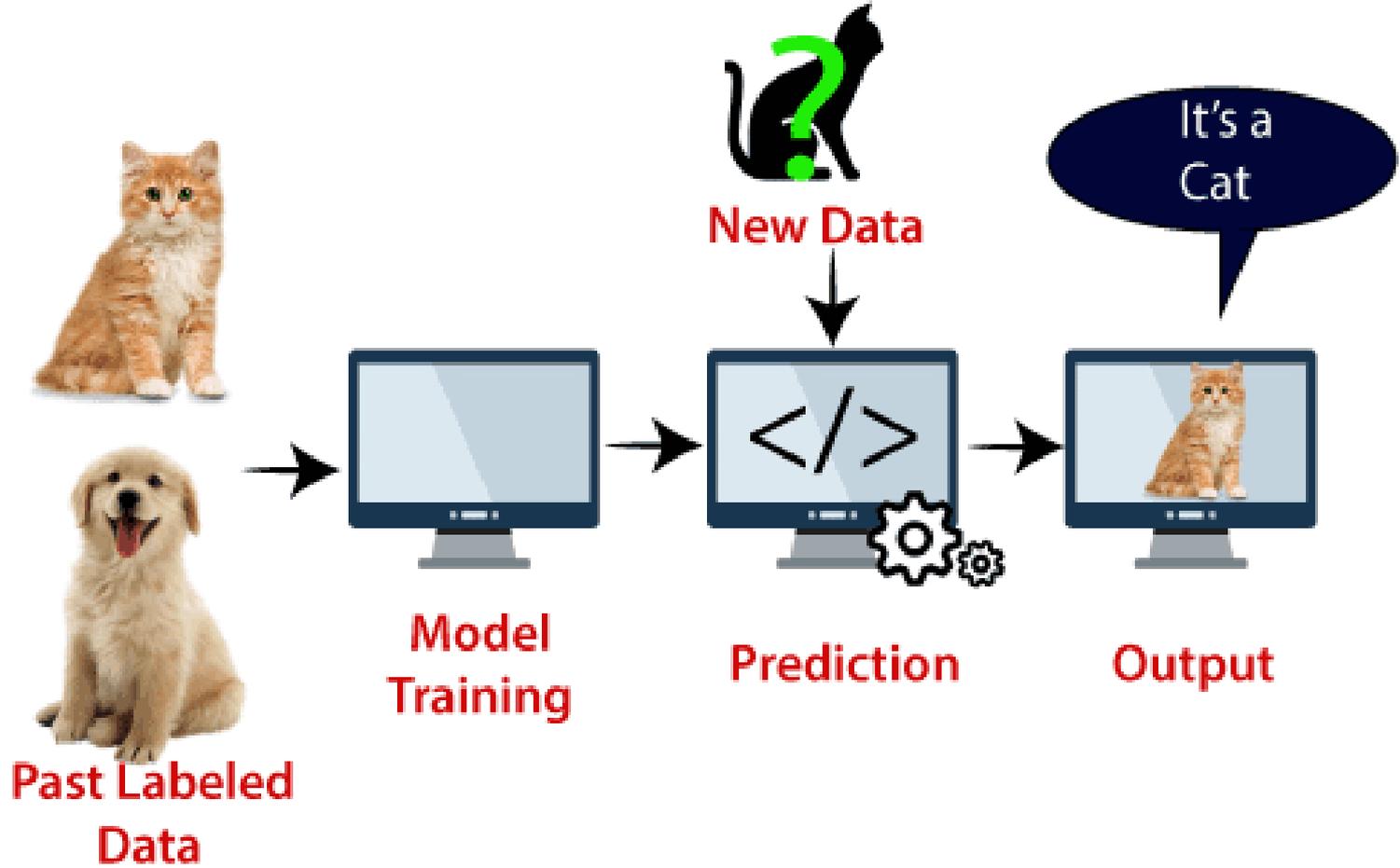
Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning.

The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane.

SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called as support vectors, and hence algorithm is termed as Support Vector Machine. Consider the below diagram in which there are two different categories that are classified using a decision boundary or hyperplane:



Example: SVM can be understood with the example that we have used in the KNN classifier. Suppose we see a strange cat that also has some features of dogs, so if we want a model that can accurately identify whether it is a cat or dog, so such a model can be created by using the SVM algorithm. We will first train our model with lots of images of cats and dogs so that it can learn about different features of cats and dogs, and then we test it with this strange creature. So as support vector creates a decision boundary between these two data (cat and dog) and choose extreme cases (support vectors), it will see the extreme case of cat and dog. On the basis of the support vectors, it will classify it as a cat. Consider the below diagram:



VM algorithm can be used for **Face detection, image classification, text categorization**, etc.

Types of SVM

SVM can be of two types:

- Linear SVM:** Linear SVM is used for linearly separable data, which means if a dataset can be classified into two classes by using a single straight line, then such data is termed as linearly separable data, and classifier is used called as Linear SVM classifier.
- Non-linear SVM:** Non-Linear SVM is used for non-linearly separated data, which means if a dataset cannot be classified by using a straight line, then such data is termed as non-linear data and classifier used is called as Non-linear SVM classifier.

Hyperplane and Support Vectors in the SVM algorithm:

Hyperplane: There can be multiple lines/decision boundaries to segregate the classes in n-dimensional space, but we need to find out the best decision boundary that helps to classify the data points. This best boundary is known as the hyperplane of SVM.

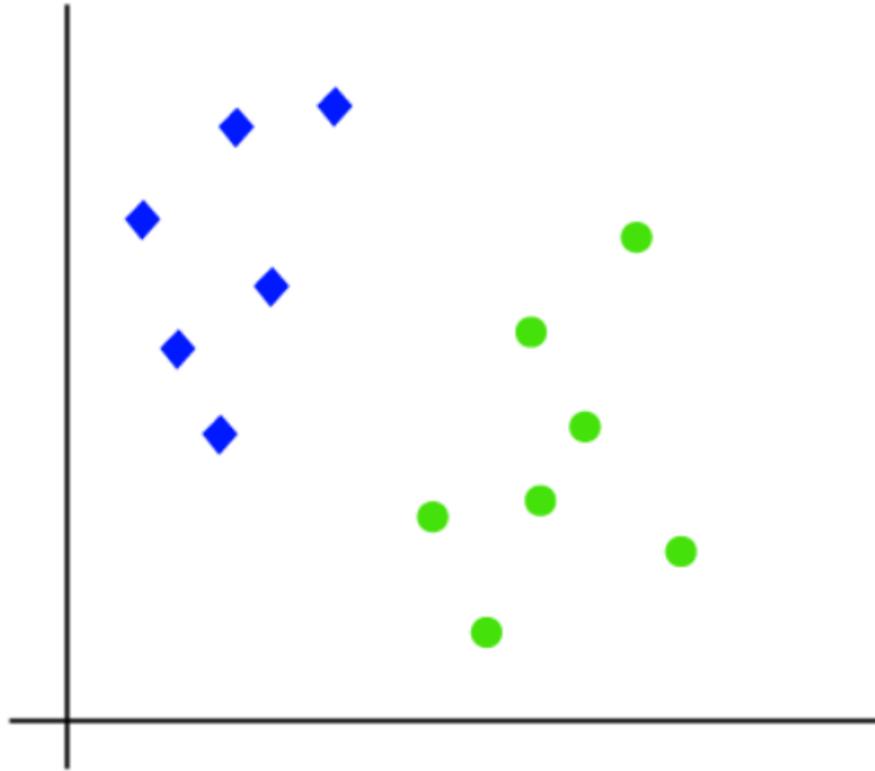
The dimensions of the hyperplane depend on the features present in the dataset, which means if there are 2 features (as shown in image), then hyperplane will be a straight line. And if there are 3 features, then hyperplane will be a 2-dimension plane.

We always create a hyperplane that has a maximum margin, which means the maximum distance between the data points.

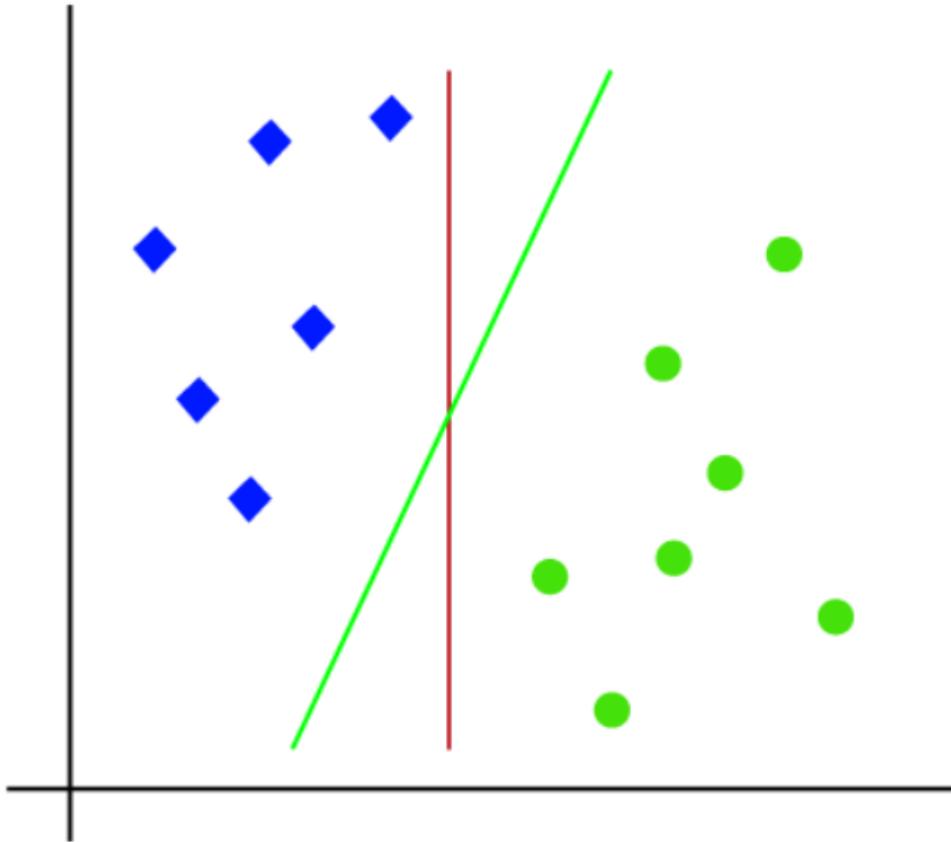
Support Vectors:

The data points or vectors that are the closest to the hyperplane and which affect the position of the hyperplane are termed as Support Vector. Since these vectors support the hyperplane, hence called a Support vector.

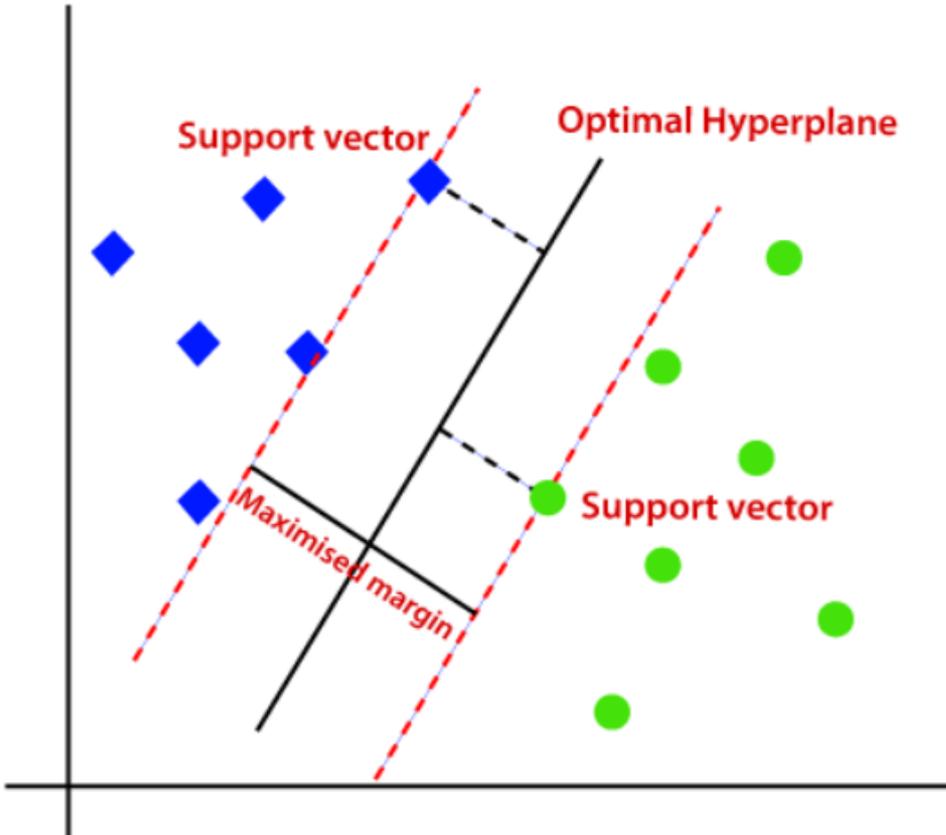
The working of the SVM algorithm can be understood by using an example. Suppose we have a dataset that has two tags (green and blue), and the dataset has two features x_1 and x_2 . We want a classifier that can classify the pair(x_1, x_2) of coordinates in either green or blue. Consider the below image:



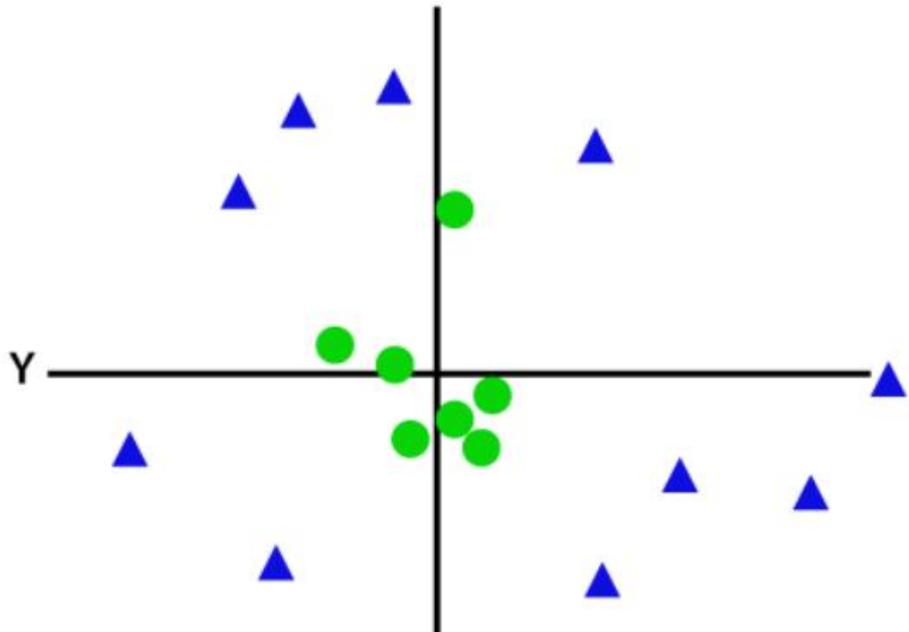
So as it is 2-d space so by just using a straight line, we can easily separate these two classes. But there can be multiple lines that can separate these classes. Consider the below image:



Hence, the SVM algorithm helps to find the best line or decision boundary; this best boundary or region is called as a **hyperplane**. SVM algorithm finds the closest point of the lines from both the classes. These points are called support vectors. The distance between the vectors and the hyperplane is called as **margin**. And the goal of SVM is to maximize this margin. The **hyperplane** with maximum margin is called the **optimal hyperplane**.



If data is linearly arranged, then we can separate it by using a straight line, but for non-linear data, we cannot draw a single straight line. Consider the below image:



A way to choose a model class

- We want to get a low error rate on unseen data.
 - This is called “structural risk minimization”
- It would be really helpful if we could get a guarantee of the following form:

Test error rate \leq train error rate + $f(N, h, p)$

Where N = size of training set,

h = measure of the model complexity,

p = the probability that this bound fails

We need p to allow for really unlucky test sets.

- Then we could choose the model complexity that minimizes the bound on the test error rate.

Training a linear SVM

- To find the maximum margin separator, we have to solve the following optimization problem:

$$\mathbf{w} \cdot \mathbf{x}^c + b > +1 \quad \textit{for positive cases}$$

$$\mathbf{w} \cdot \mathbf{x}^c + b < -1 \quad \textit{for negative cases}$$

$$\textit{and } \|\mathbf{w}\|^2 \textit{ is as small as possible}$$

- This is tricky but it's a convex problem. There is only one optimum and we can find it without fiddling with learning rates or weight decay or early stopping.
 - Don't worry about the optimization problem. It has been solved. Its called quadratic programming.
 - It takes time proportional to N^2 which is really bad for very big datasets
 - so for big datasets we end up doing approximate optimization!

Testing a linear SVM

- The separator is defined as the set of points for which:

$$\mathbf{w} \cdot \mathbf{x} + b = 0$$

so if $\mathbf{w} \cdot \mathbf{x}^c + b > 0$ say its a positive case

and if $\mathbf{w} \cdot \mathbf{x}^c + b < 0$ say its a negative case

Introducing slack variables

- Slack variables are constrained to be non-negative. When they are greater than zero they allow us to cheat by putting the plane closer to the datapoint than the margin. So we need to minimize the amount of cheating. This means we have to pick a value for lambda ([this sounds familiar!](#))

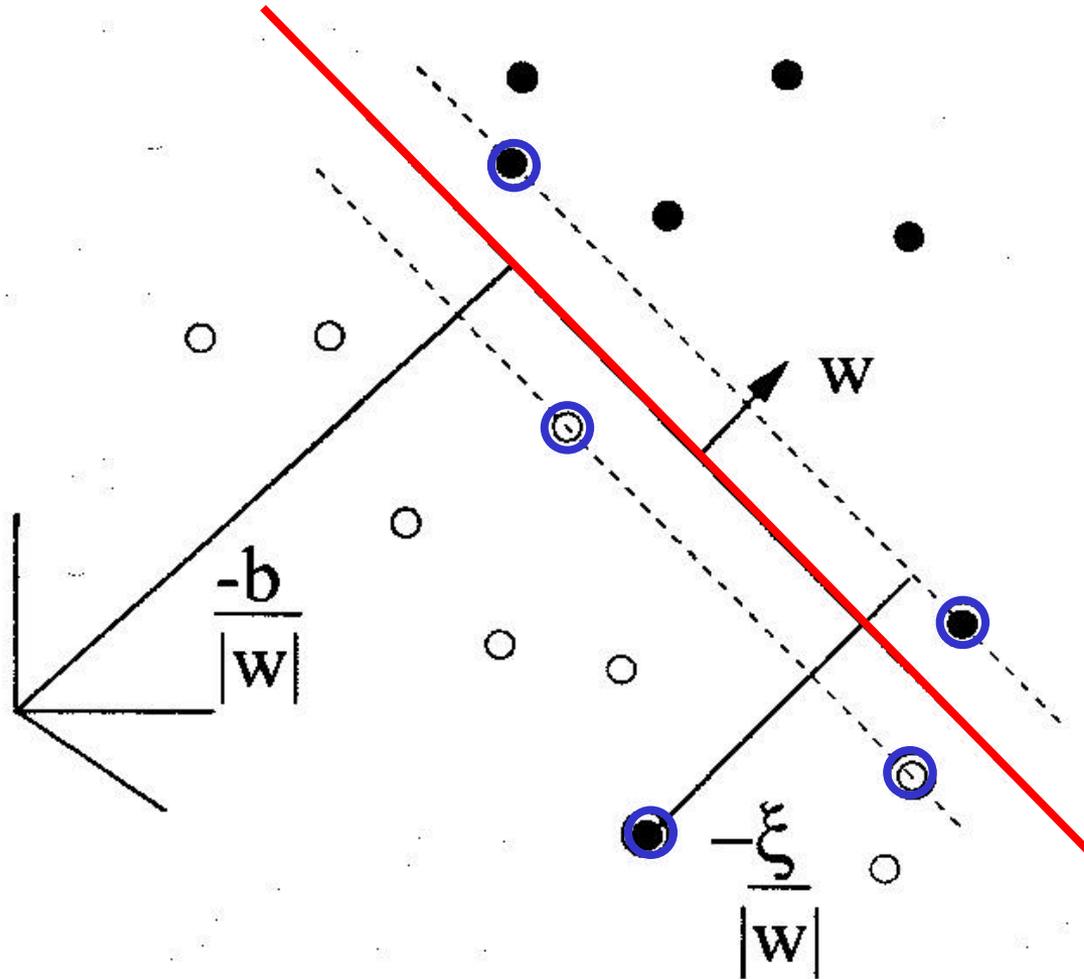
$$\mathbf{w} \cdot \mathbf{x}^c + b \geq +1 - \xi^c \quad \text{for positive cases}$$

$$\mathbf{w} \cdot \mathbf{x}^c + b \leq -1 + \xi^c \quad \text{for negative cases}$$

$$\text{with } \xi^c \geq 0 \quad \text{for all } c$$

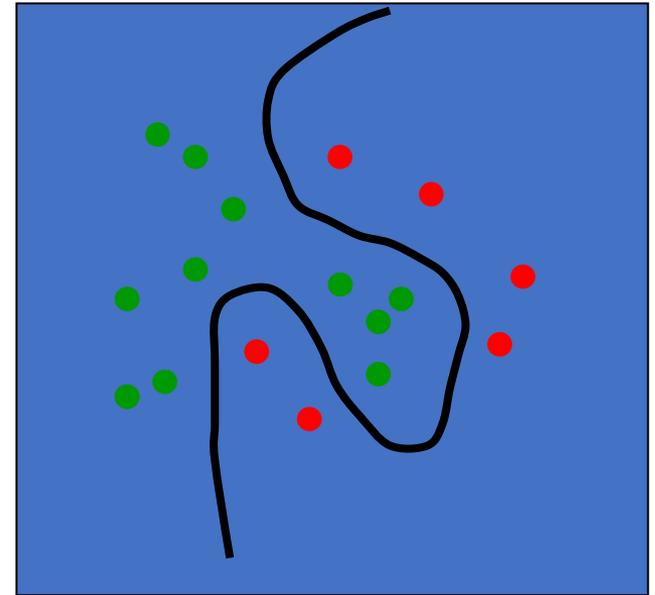
$$\text{and } \frac{\|\mathbf{w}\|^2}{2} + \lambda \sum_c \xi^c \quad \text{as small as possible}$$

A picture of the best plane with a slack variable



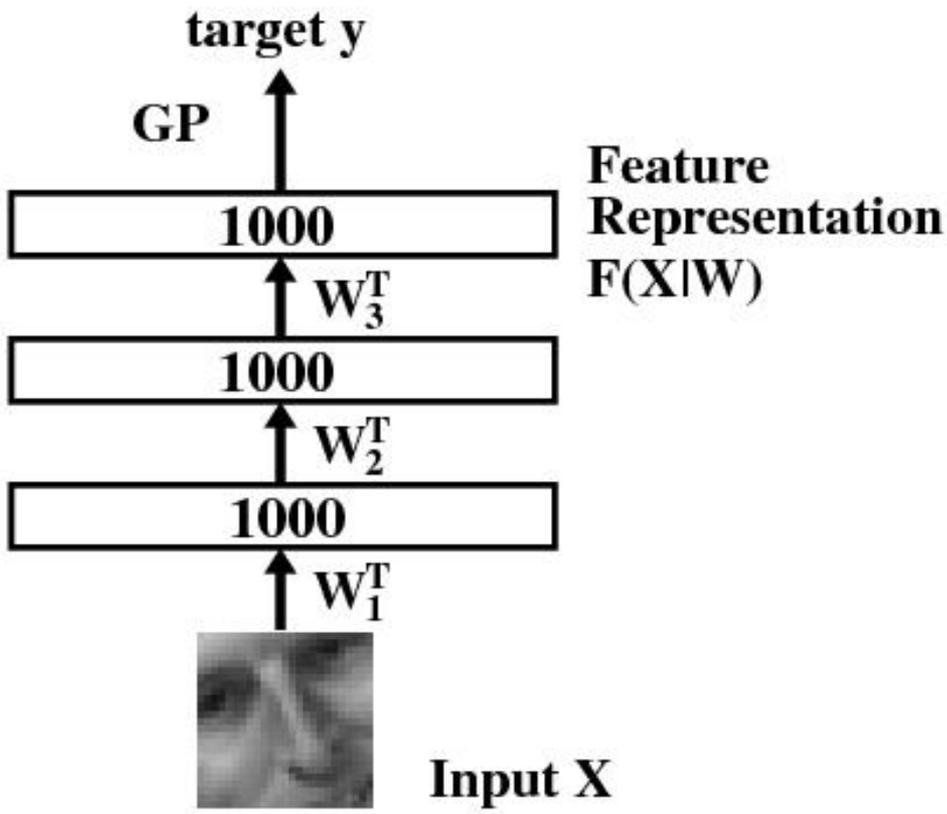
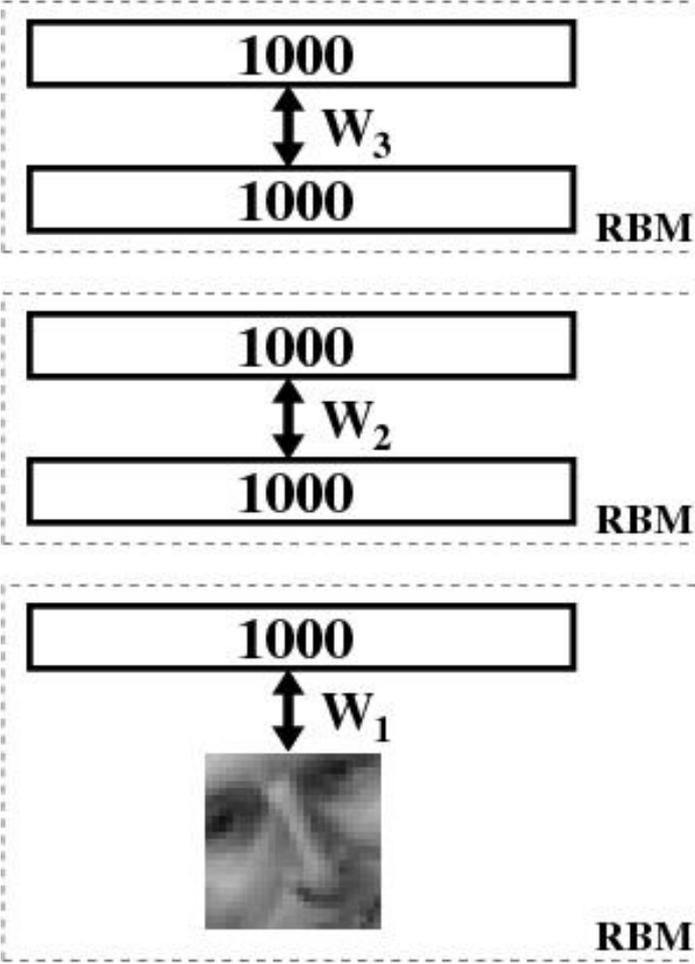
How to make a plane curved

- Fitting hyperplanes as separators is mathematically easy.
 - The mathematics is linear.
- By replacing the raw input variables with a much larger set of features we get a nice property:
 - A planar separator in the high-dimensional space of feature vectors is a curved separator in the low dimensional space of the raw input variables.



A planar separator in a 20-D feature space projected back to the original 2-D space

Learning to extract the orientation of a face patch (Ruslan Salakhutdinov)



The training and test sets



