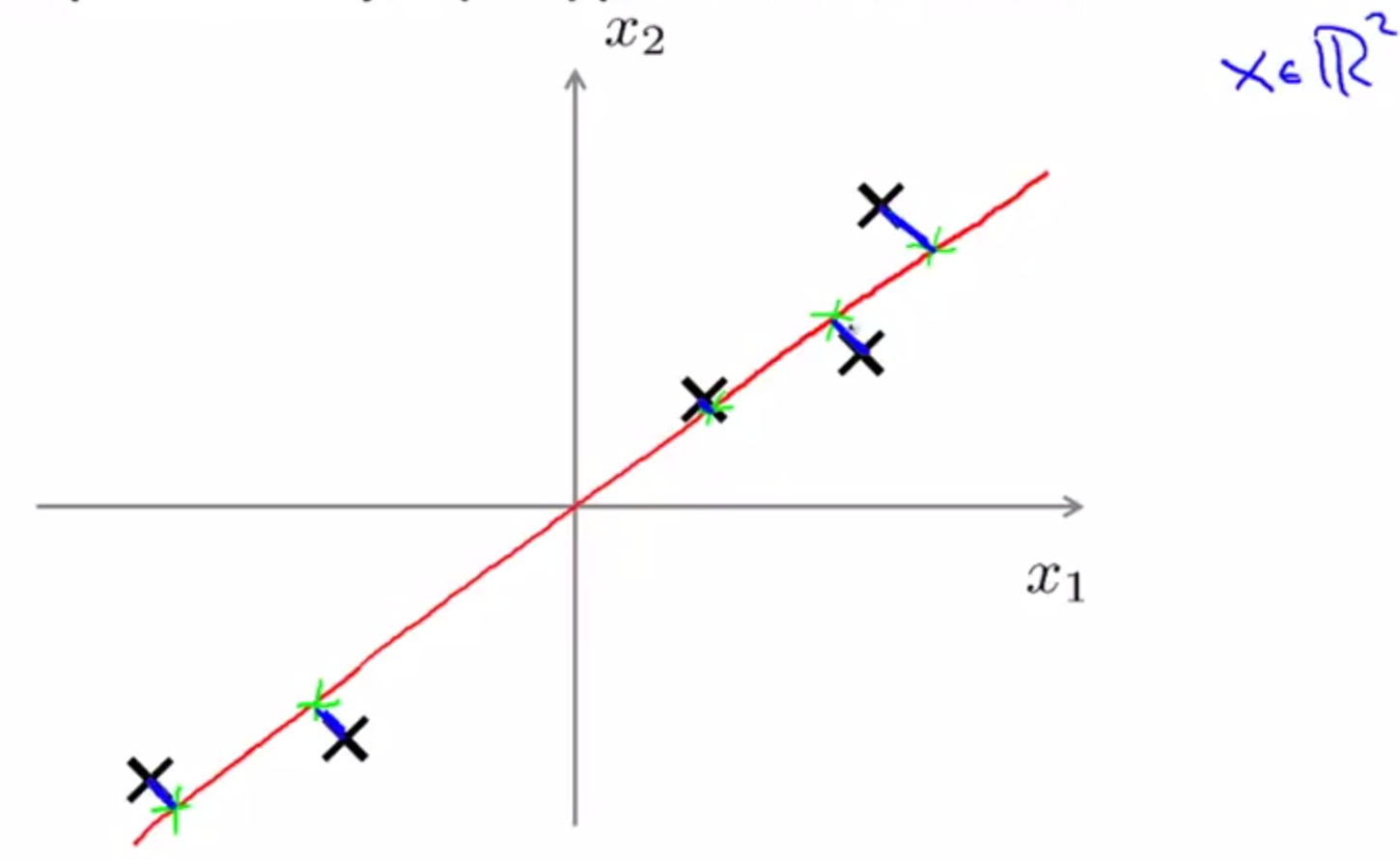
***Dimensionality Reduction***

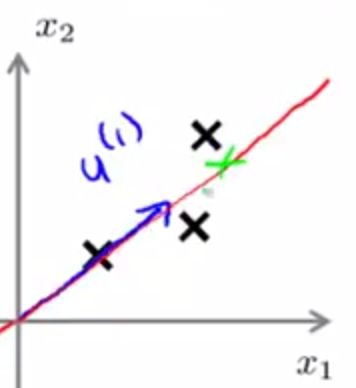
***Principal Component Analysis***

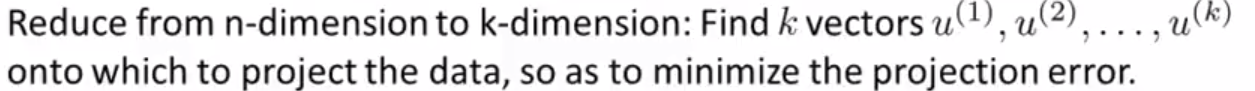
**I. PRINCIPAL COMPONENT ANALYSIS PROBLEM FORMULATION**

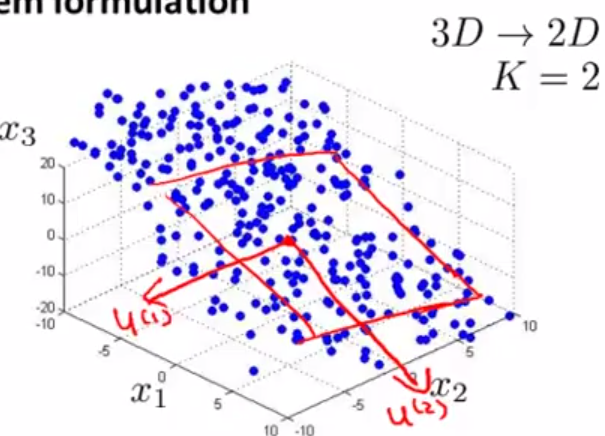
* For the problem of dimensionality reduction, by far the most commonly used algorithm is **principle components analysis** (**PCA**)
* Ex: Data set of examples x in R2 + we want to reduce the dimensions of the data from 2D to 1D (find a line onto which to project the data)



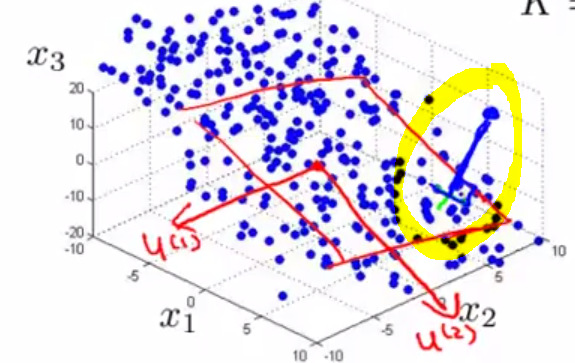
* A good line onto which to project the data might be like above b/c for the projected versions of the points (green), the distance between each point + the projected version (blue) is pretty small/short
* What PCA does is try to find a lower-dimensional surface (a line in this case) onto which to project the data so that the sum of squares (**SSE**) of these segments is minimized (**the projection error**).
* Before applying PCA, it's standard practice to first perform **mean normalization** + **feature scaling** so features x1 + x2 should have 0 mean + comparable ranges of values.
* The goal of PCA, if we want to reduce data from 2D to 1D, is to try find a direction/vector u1 in R(n) onto which to project the data to minimize the projection error even when extending out the vector



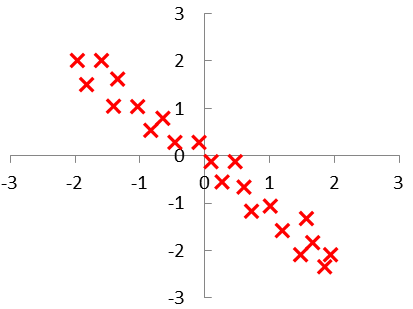
* ***Note:*** Whether PCA gives a positive or negative vector doesn't matter b/c each vector defines the same line onto which we’re projecting data.
* 
* In the more general case of n-dimensional data reduced to k-dimensions, we want to find not just a *single* vector but to find k vectors/dimensions onto which to project the data
* If we have a 3D point cloud like this, maybe we want to do is find a pair of vectors that can define a plane/2D surface onto which I am going to project my data.



* The formal definition of this: Find the **set** of vectors u(1), u(2), … u(k) + project the data onto the **linear subspace** *spanned by this set of k vectors.*
* We want to find a k-dimensional surface (a 2D plane in this case) where we can define the position of the examples in a plane using k directions to minimize the of projection distance
* In a 3D example, given a point, we would take it + project it onto a 2D surface + the projection error would be the distance between the point + where it gets projected down to my 2D surface.



* PCA tries to find the line/plane/whatever onto which to project data to try to minimize that error
* *How does PCA relate to linear regression*?
* When explaining PCA, sometimes it looks a bit like linear regression.
* Despite some cosmetic similarity, these are totally different algorithms.
* If doing linear regression, we’d be trying to predict the value of some variable Y given some info features X by fitting a straight line so as to minimize the squared error between a point + that straight line
* These error lines are the vertical distance between the point and the value predicted by the hypothesis.
* Whereas in contrast, in PCA, it tries to minimize the magnitude of these lines *drawn at an angle,* the **shortest orthogonal distances**/distance between the point x and this red line.
* This gives very different effects depending on the dataset.
* More generally, w/ linear regression, there is a distinguished variable Y we're trying to predict.
* We’re taking all the values of X and try to use that to predict Y.
* Whereas in PCA, there is no distinguished variable Y we're trying to predict + instead we have a list of features X-X(n) + all these features are treated equally
* 



* 

**II. PRINCIPAL COMPONENT ANALYSIS ALGORITHM**