CS 189/289

Today's lecture:

1. General Info (also read www.eecs189.org in full)
2. Introduction to ML (Prof. Malik)

Prof. Malik

Prof. Listgarten

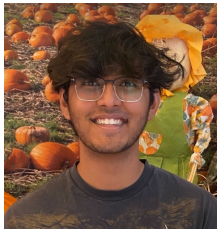
Head GSI: Arvind Rajaraman

Student Instructors

Head GSI
Arvind Rajaraman
4th Year BS
Interests: embodied robots, foundation models

Philip Jacobson
5th Year PhD
Interests: 3D perception, autonomous vehicles

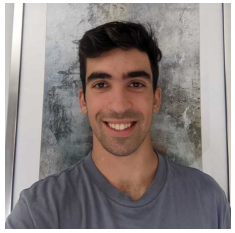
Bear Xiong
3th Year PhD
Interests: machine learning and biology



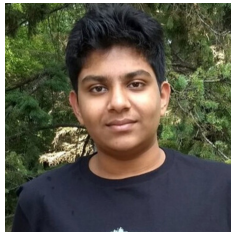
Sandeep Mukherjee
4th Year MS
Interests: robotics, grasping

Sampada Deglurkar
4th Year PhD
Interests: ML safety

Andrew Qin
3rd Year BS
Interests: ML applied to game theory



Samuel Alber
4th Year BS
Interests: computational biology, biophysics



Aryan Jain
4th Year BS
Interests: robot learning, few-shot learning for CV

This class is in person only

1. All lectures are in person. Lecture recordings will only be released in two batches: before (1) the midterm and (2) final.
2. All discussion sections will be in person.
3. NO ALTERNATE EXAM TIMES will be given other than what is listed as the official final exam date and time, other than DSP which gets an extended (but overlapping) version.
4. EXAMS MUST BE TAKEN IN PERSON.
5. "Ed" post with detailed exam policy coming soon.

Midterm

Friday 10/13 from 7-9pm @ Pimentel 1

(DSP will overlap with this)

Enrollment

- The class is full.
- There are no instructor-allocated enrollment codes.
- You might get off the wait list. We don't know (it's automatic).
- Concurrent enrollment: TBD.
- Questions: email cs-enrollments@berkeley.edu, or EECS enrollment drop-in office hours (look on-line for dates and times).
- If you are on a waitlist: complete all assignments as though you were enrolled.

Gradescope & Ed

Ed - https://edstem.org/us/join/fCBF32

Gradescope - https://www.gradescope.com/courses/575262

Auditing

- You may attend lectures if there are empty seats available in the lecture hall, but you may not come to discussion section, or use up staff resources in any other way. You will not get added to bCourses or Ed, however, most course content is on the main web page: www.eecs189.org. Video lectures are for enrolled students only, on bcourses.

Contact information

- Read www.eecs189.org before initiating any contact (WIP).
- Most questions should go to a public Ed post first.
- If needed, private post on Ed to instructors.
- Professors will have office hours after each lecture (with the professor who has been lecturing), as noted on course web page (see calendar).
- DSP: Krystle Simon (krystle@eecs.berkeley.edu)

Pre-requisites

- Courses you should have taken (or equivalent):

  EE16A & EE16B & CS70 & MATH53
- Some core areas needed: linear algebra, vector calculus, probability, basic programming skills.
- Discussion and Homework 0 will be indicative of whether you have the necessary math knowledge to succeed in the class.

Final grade breakdown

CS 189 & 289

- Homework: 20%
- Midterm: 35%
- Final: 45%

CS 289A will be distinguished by having extra exam questions. (No class project).

CS 189 and CS 289A graded on separate curves.

Homeworks

- Combination of pen and paper & coding (python).
- Total of around 8 homeworks (TBD).
- See website for specific due dates.
- HW 0 is still yet to be released but it will be due before the add/drop deadline. Likely to be released by end of weekend, and we will scale the amount of work for the assignment based on how much time you have to complete it.

Homework niceties

- No late homework accepted, but…
- Can drop your two lowest homework grades.
- 80% correctness gives you a full grade on each homework.
- **Intention: to encourage you to learn the material.**

Collaboration

- Encouraged to work on homework problems in study groups of 2-5 people. These people must be named in your write-up.
- **Must always write up the solutions on your own.**
- **You are not permitted to look at anyone's final written solution, even for members of your own study group.**
- You may use books or online resources (not solutions from previous terms, etc.) to help solve homework problems, but you must always credit all such sources in your writeup and you must never copy material verbatim.

Cheating

- Zero-tolerance policy for cheating.
- No online communication outside of Ed to discuss assignment content (other than formally *listed* collaborators).
- Looking at online solutions from previous semesters or other students is forbidden. Staff will specifically be watching for this.
- Sharing of your solution with others is forbidden.
- Full details on course website.

Materials you are responsible for

- Lectures and Discussions are the course material, and you are responsible for all of it.
- No textbook.
- Slides will be posted after each lecture (Ed).
- Pointers may be given to auxiliary material that may help your understanding (see course web site).

CS 189/289

Today's lecture:

1. General Info (also read www.eecs189.org in full)
2. Introduction to ML (Prof. Malik)

CS 189/289 : Introductory Lecture

Jitendra Malik

Examples of learning problems

- Recognizing digits
- Classifying email as spam or not
- Predicting the price of a stock 6 months from now
- Netflix problem – predict rating of a movie j by a customer i
- Determine credit-worthiness for a mortgage or a credit card transaction

Types of problems

- Classification problems, where the output is a label from a finite set (in the simplest case, just binary)
- Regression problems, where the output is real-valued
- Ranking problems, such as an internet query for a document relevant to a search term

Handwritten digit recognition on MNIST

- Has a training set of 60 K examples (6K examples for each digit), and a test set of 10K examples.
- Each digit is a 28 x 28 pixel grey level image. The digit itself occupies the central 20 x 20 pixels, and the center of mass lies at the center of the box
- Error rates of best systems are under 0.5%

The machine learning approach to object recognition

- Training time
  - Compute feature vectors for positive and negative examples of image patches
  - Train a classifier
- Test Time
  - Compute feature vector on image patch
  - Evaluate classifier

Let us take an example…

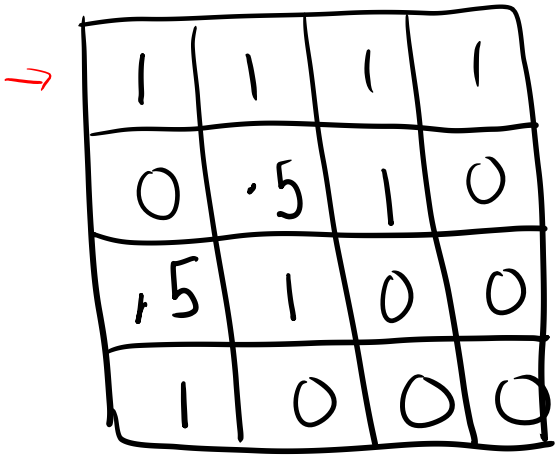
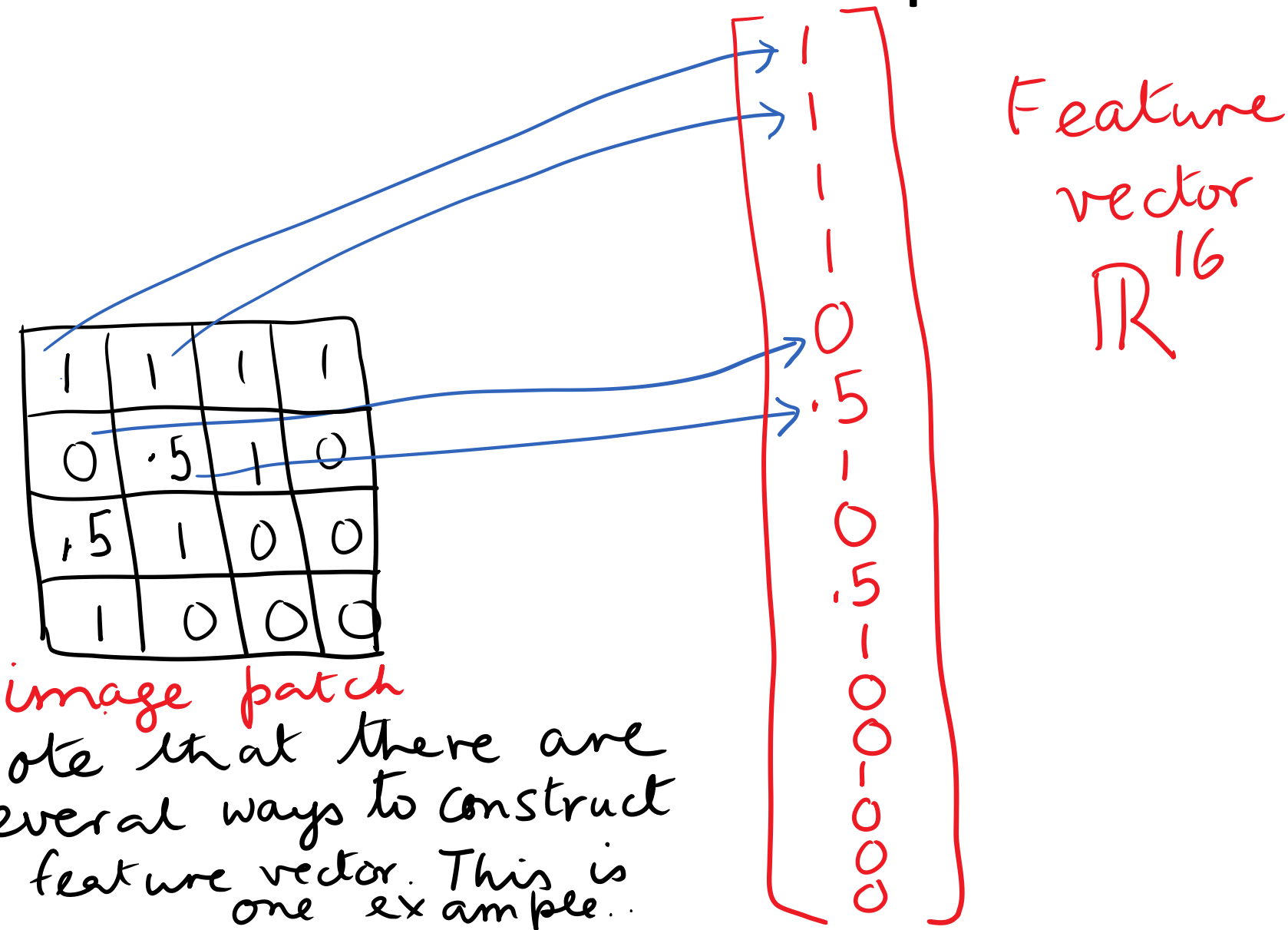


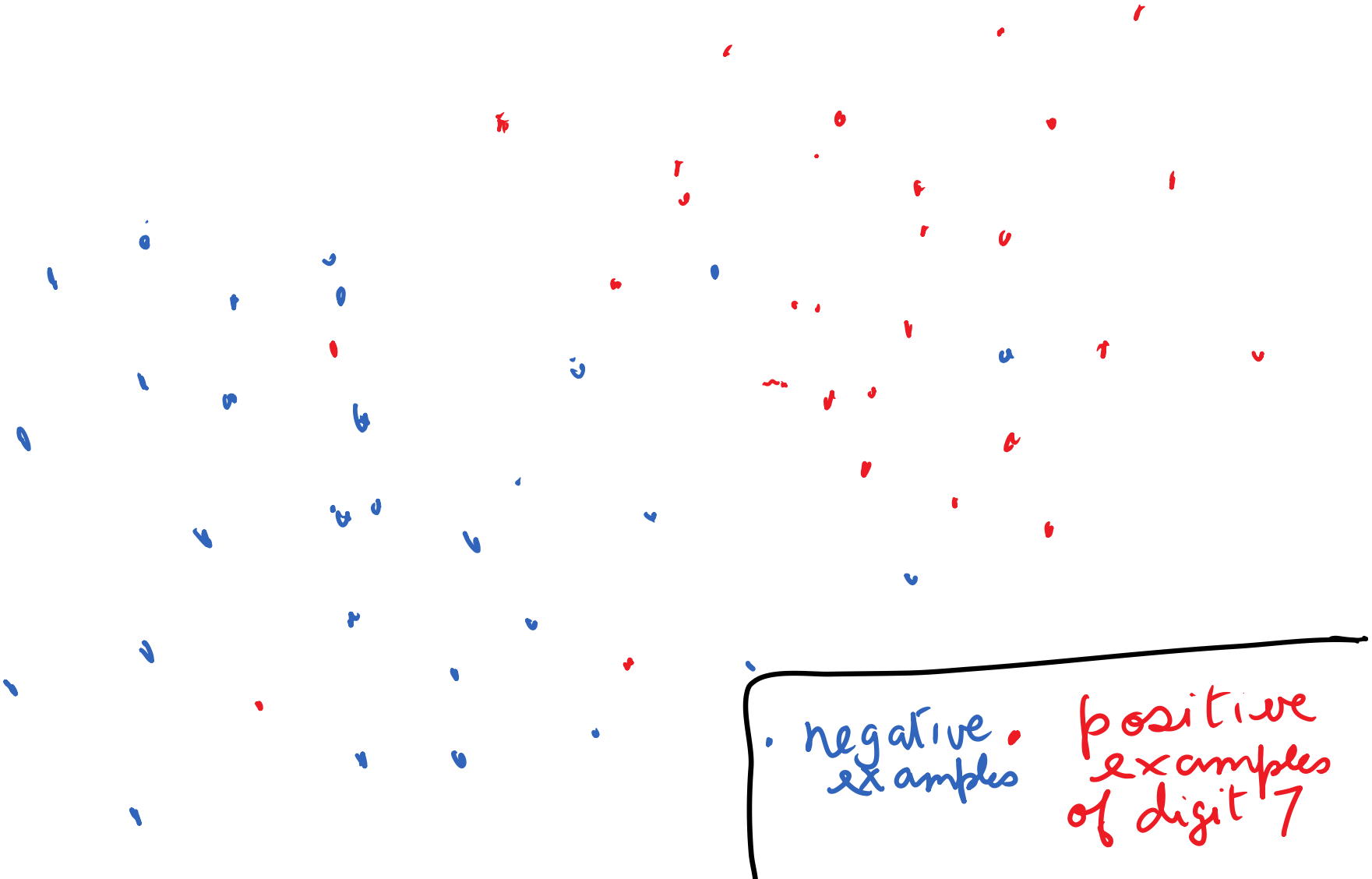
image patch

Let us take an example...

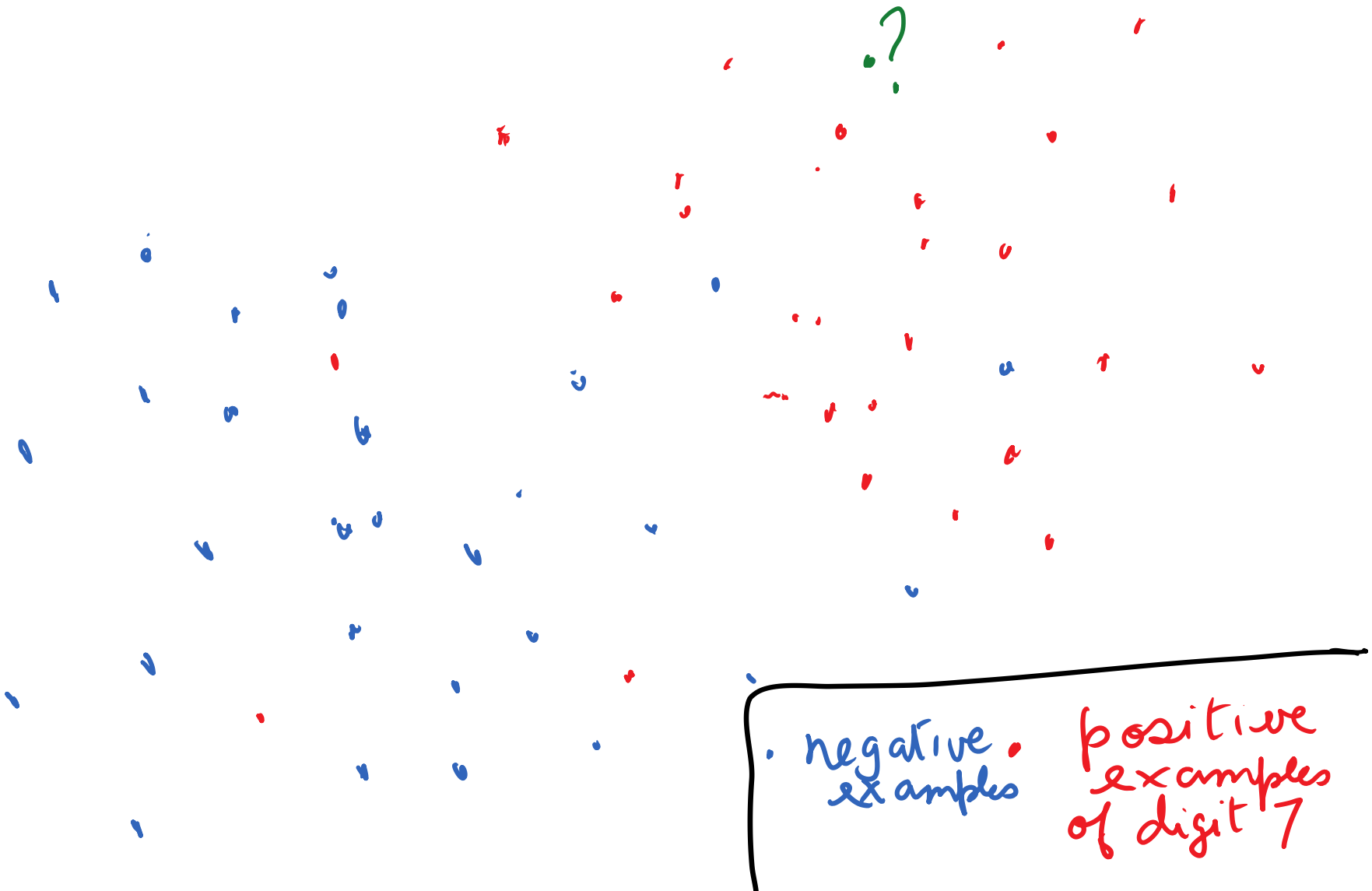


Note that there are several ways to construct a feature vector. This is one example.

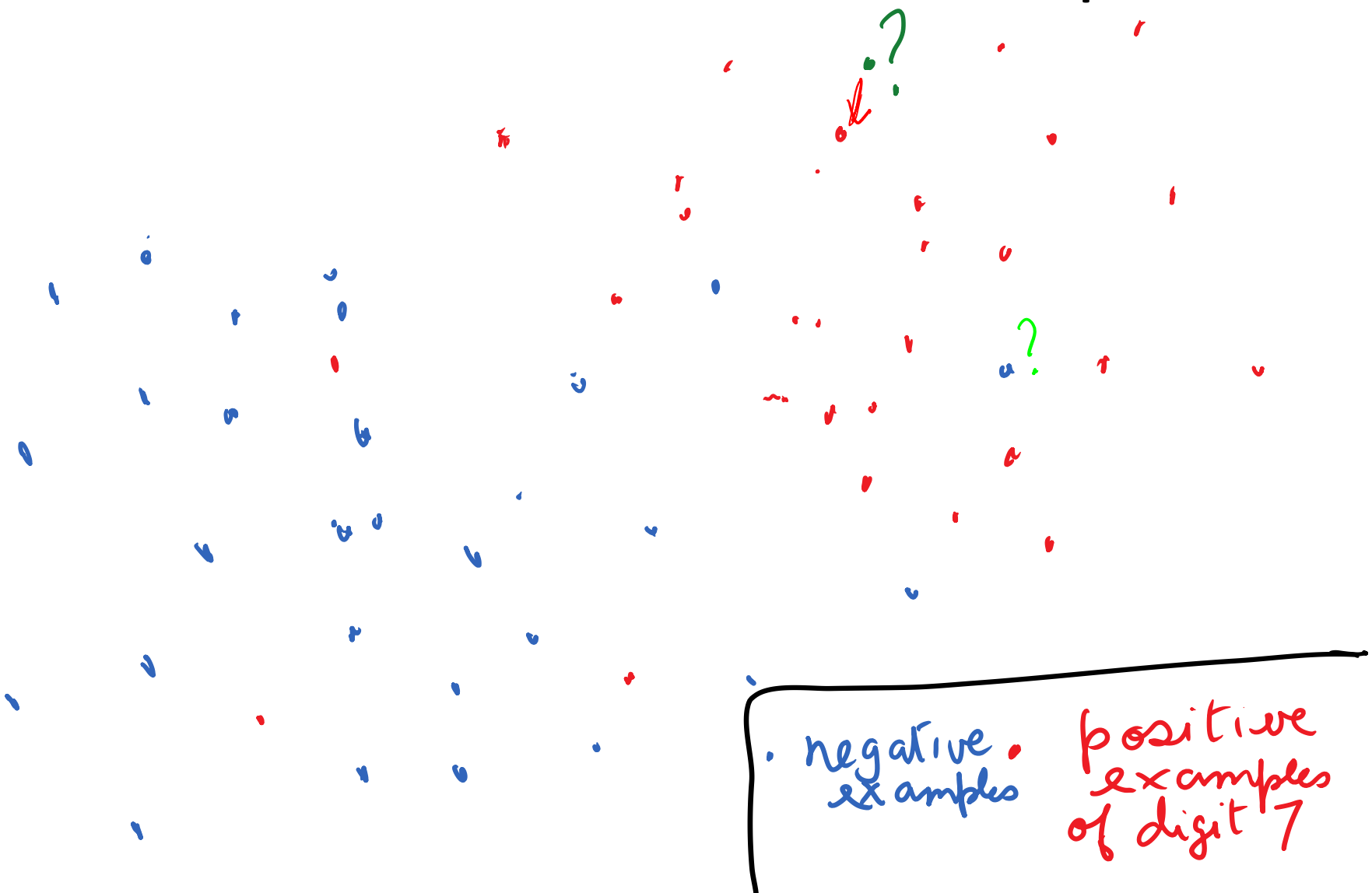
In feature space, positive and negative examples are just points…



How do we classify a new point?



Nearest neighbor rule
“transfer label of nearest example”



Linear classifier rule

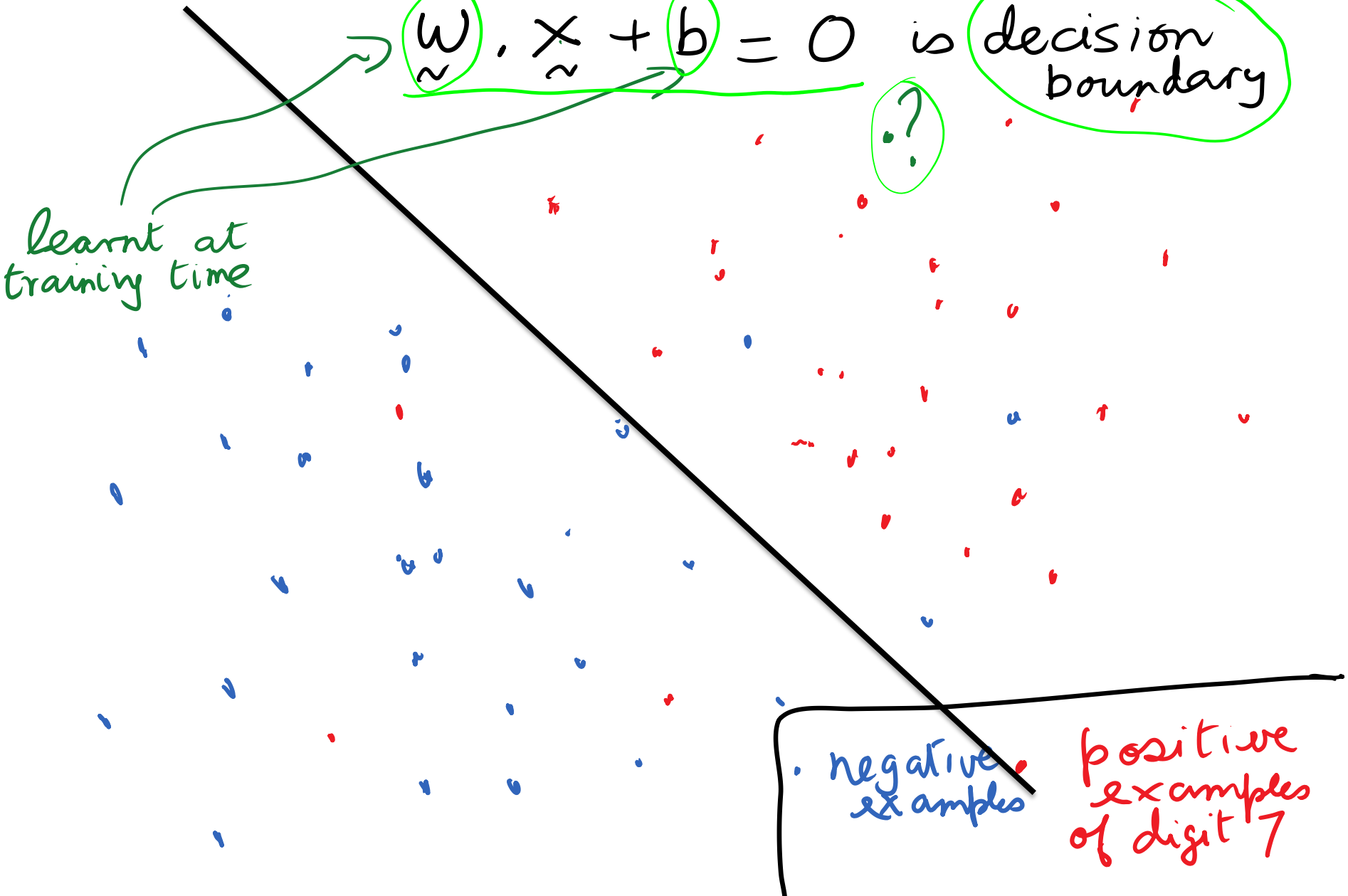
$\underset{\sim}{w} \cdot \underset{\sim}{x} + b = 0$ is decision boundary

?

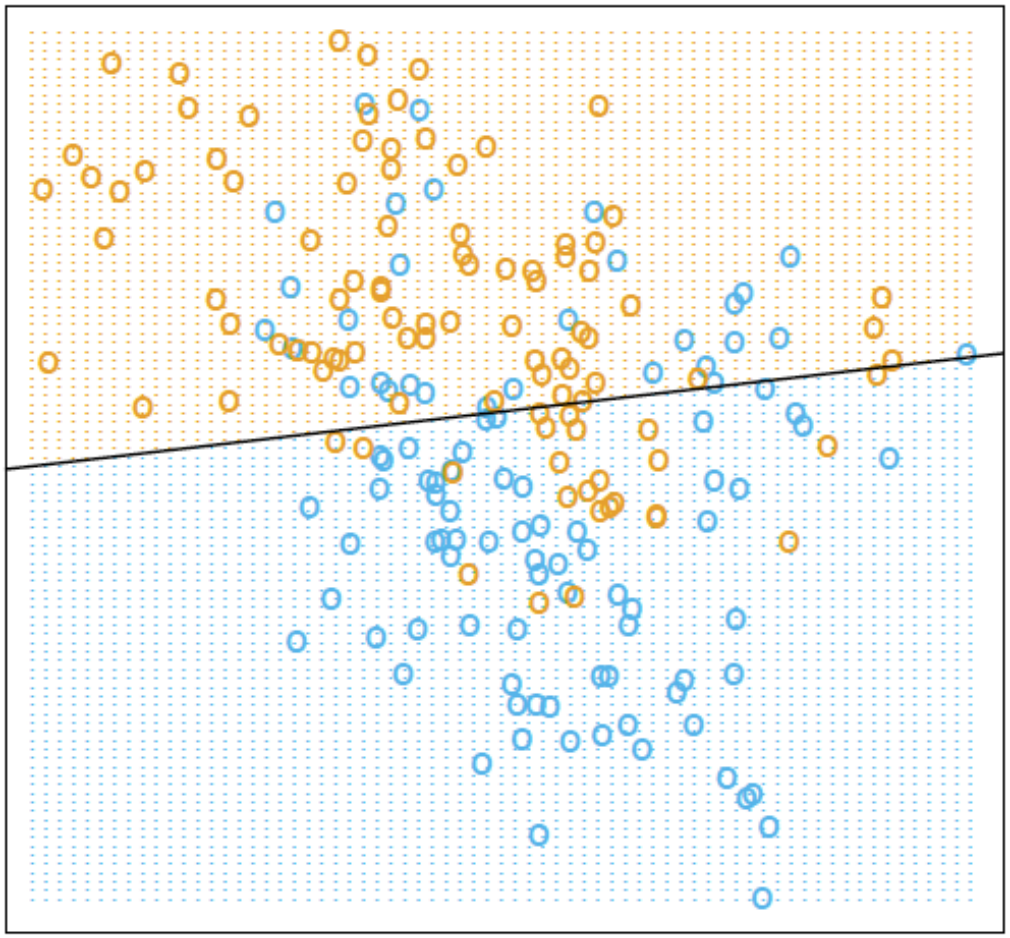
learnt at training time

negative examples

positive examples of digit 7

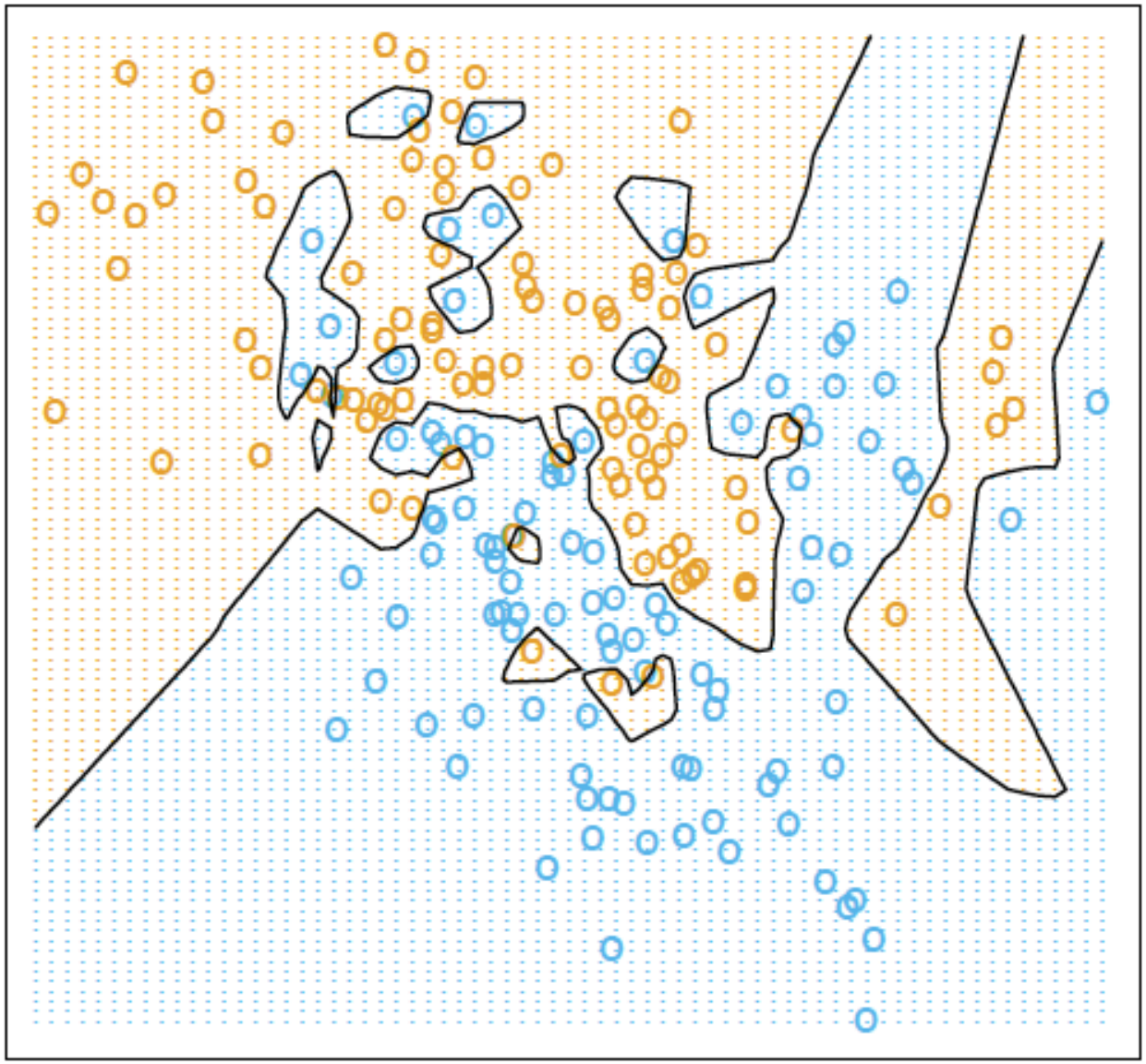


Linear Classifier
[Source : Hastie, Tibshirani, Friedman]

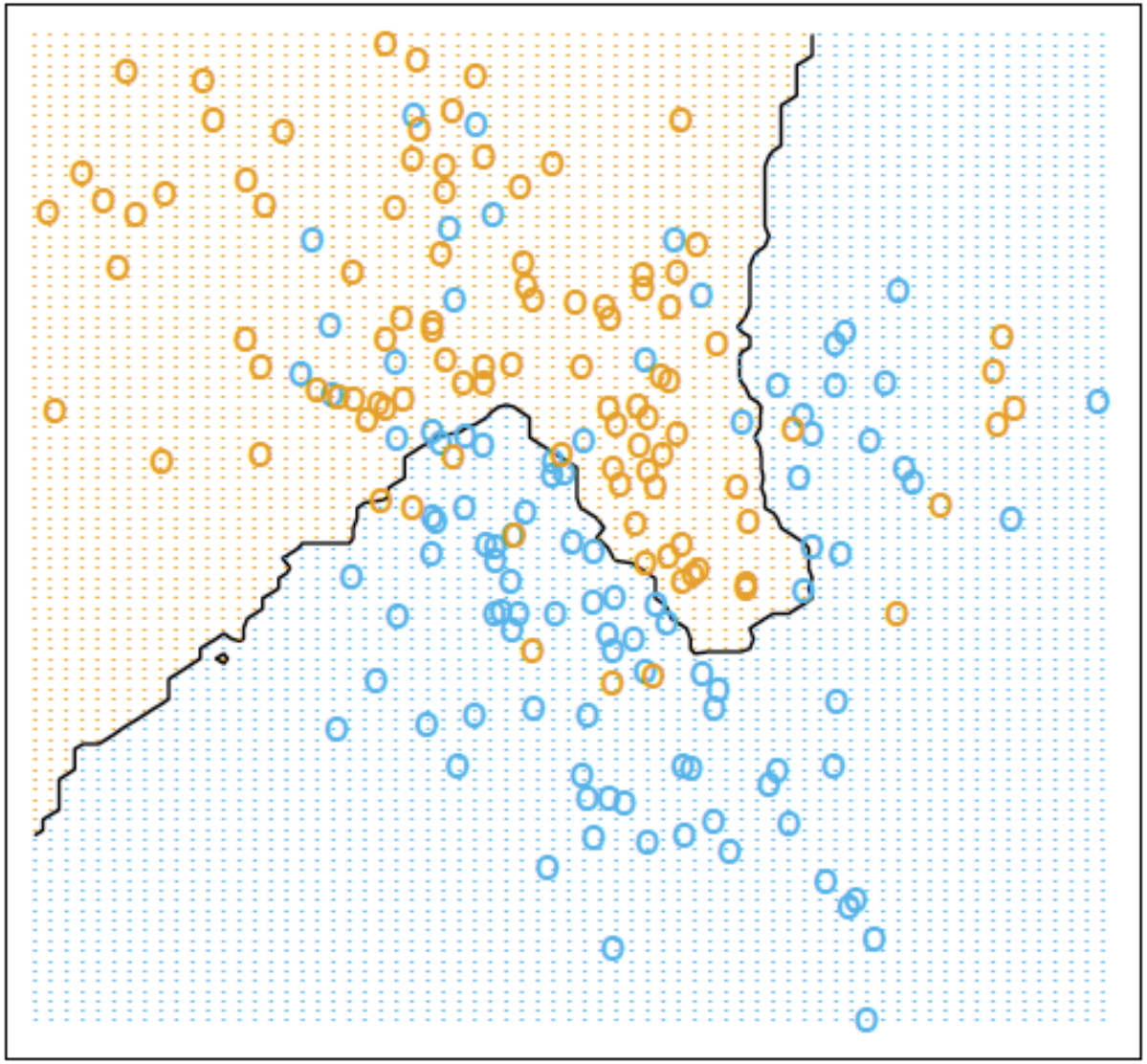


How do we get non-linear decision boundaries?

1-Nearest Neighbor



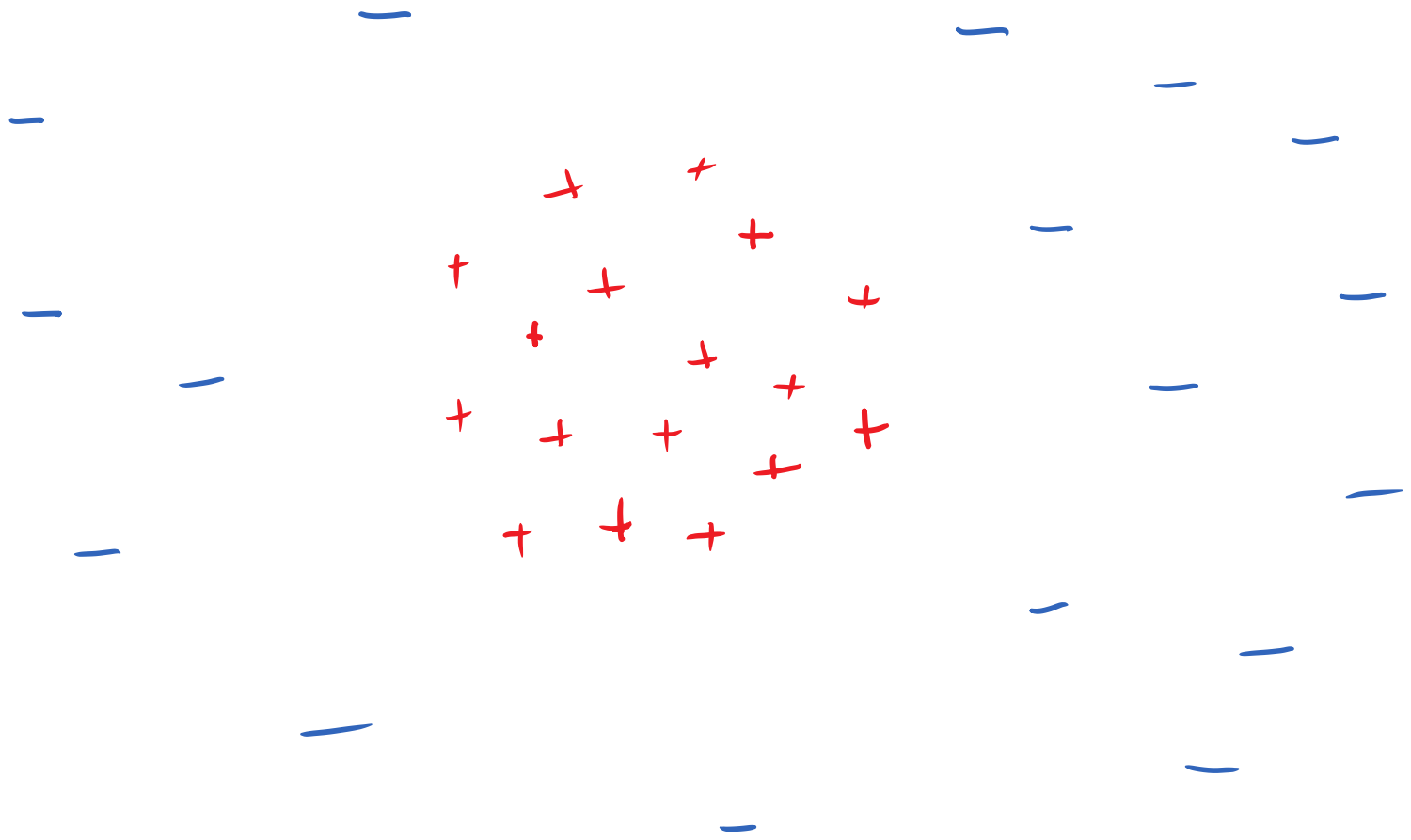
15-Nearest Neighbor



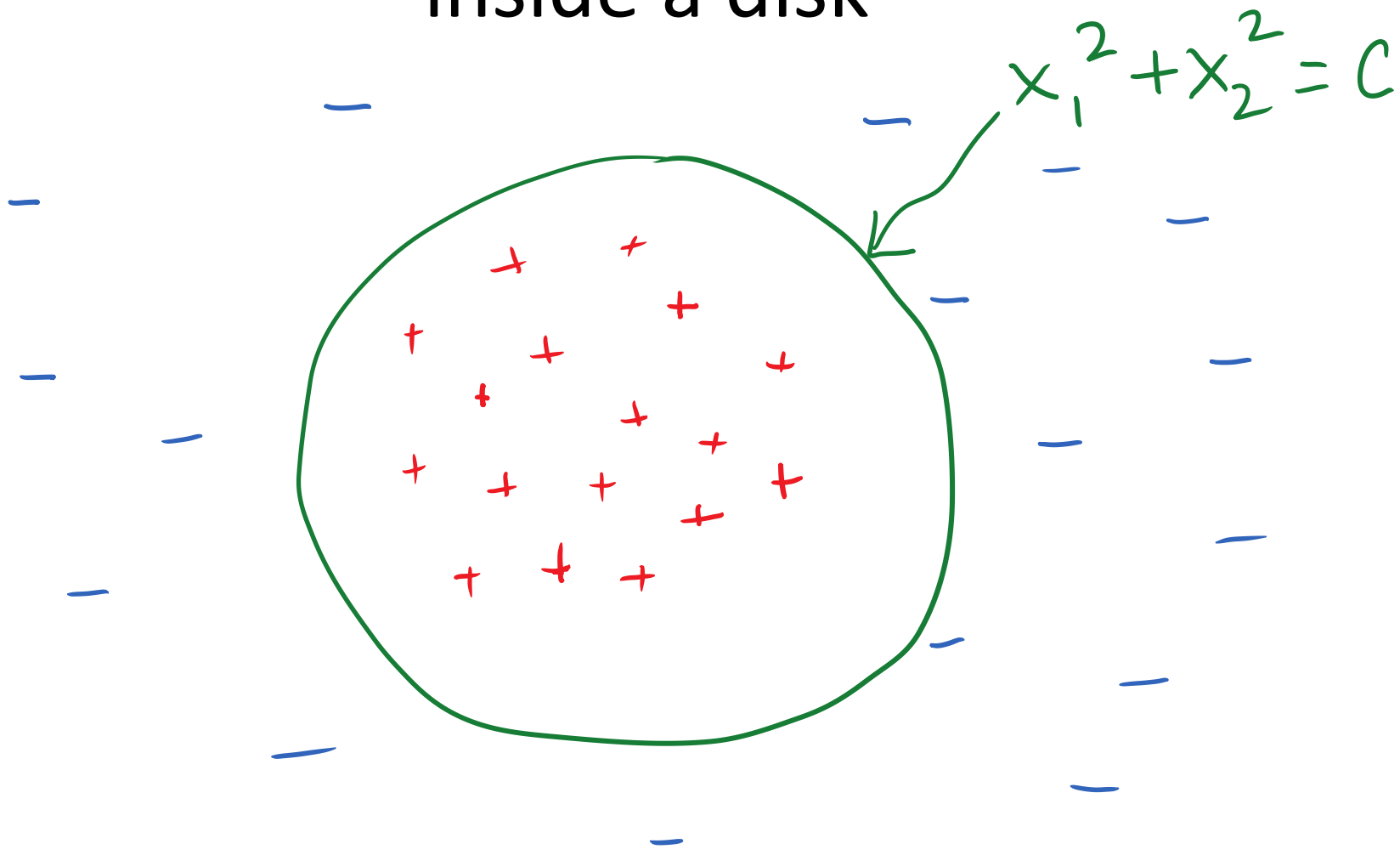
How do we get non-linear decision boundaries?

- Nearest neighbors
- Multilayer perceptrons a.k.a Neural Networks

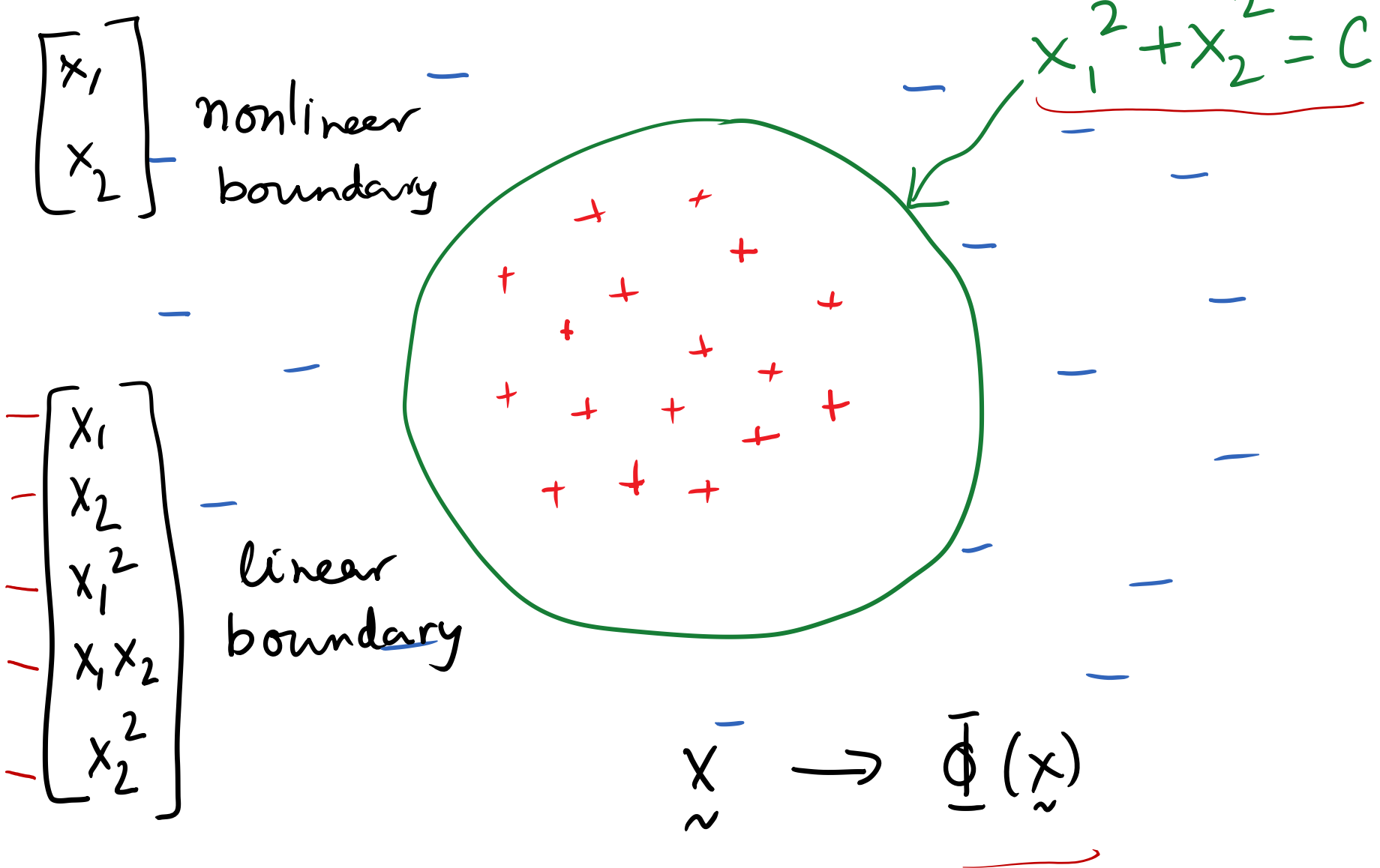
Suppose the positive examples lie inside a disk



Suppose the positive examples lie inside a disk

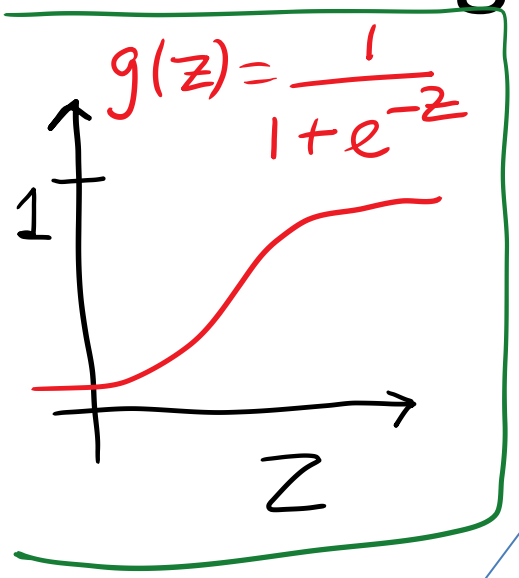


We can construct a new higher-dimensional feature space where the boundary is linear

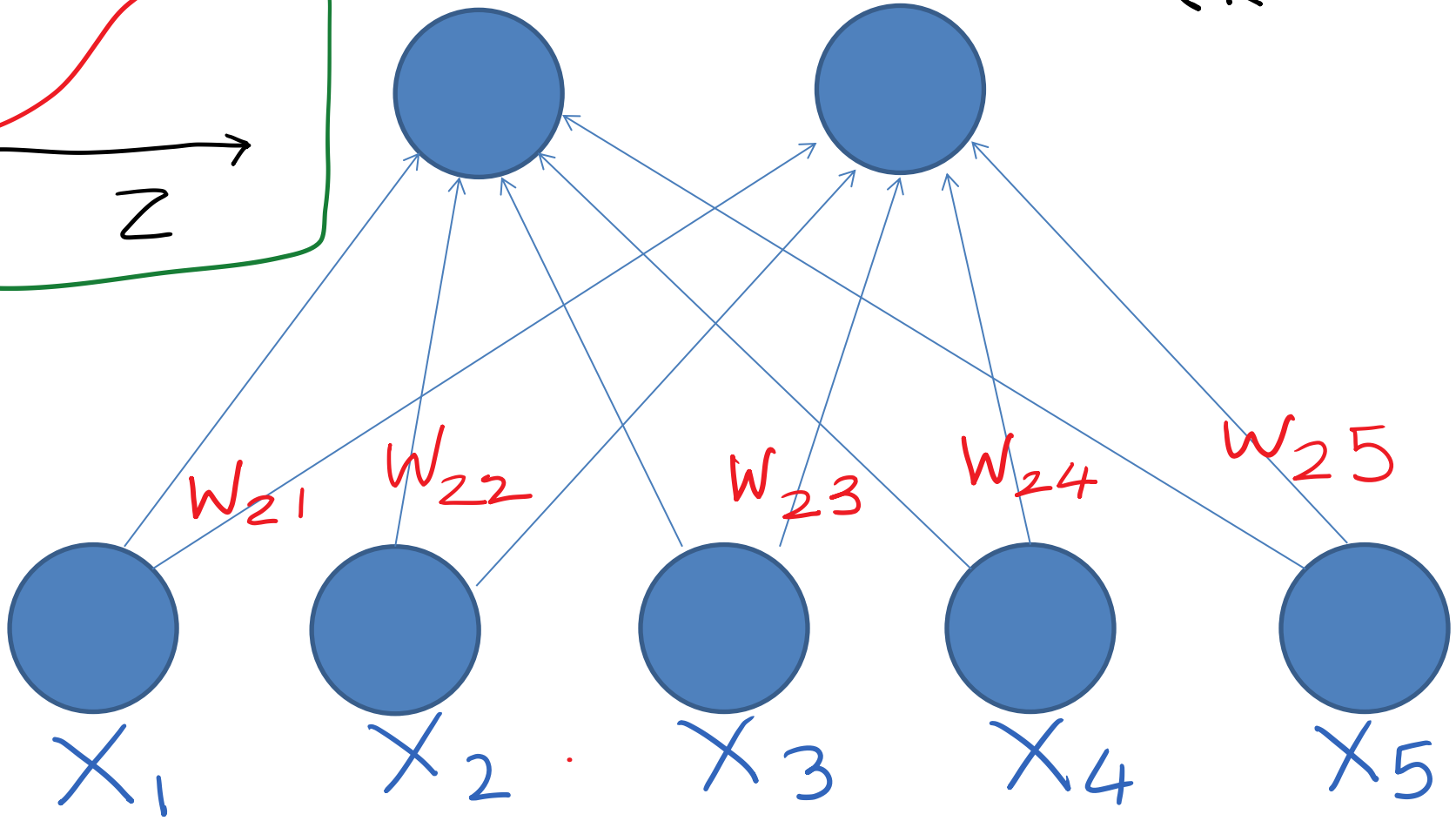


Neural Networks

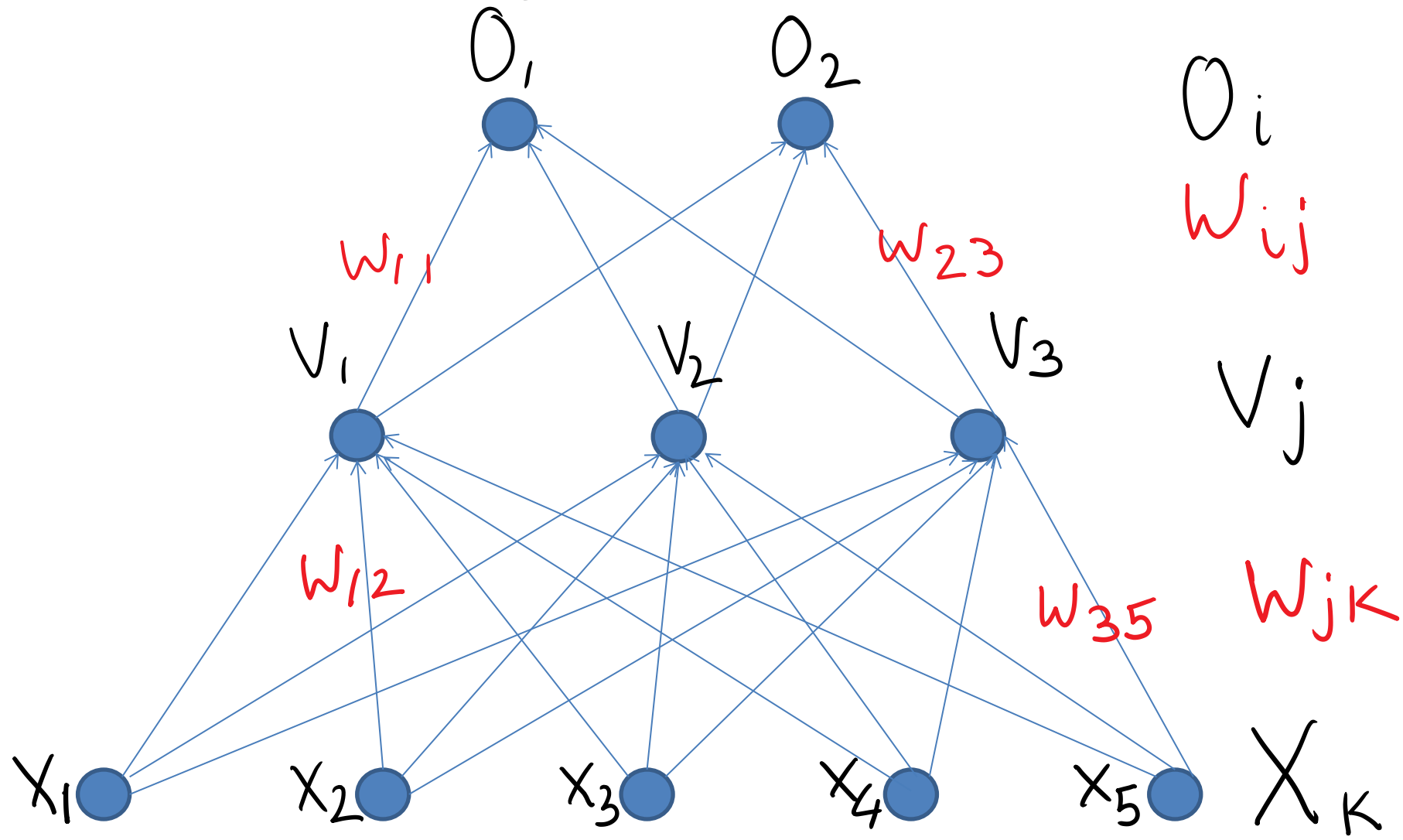
Single layer neural network



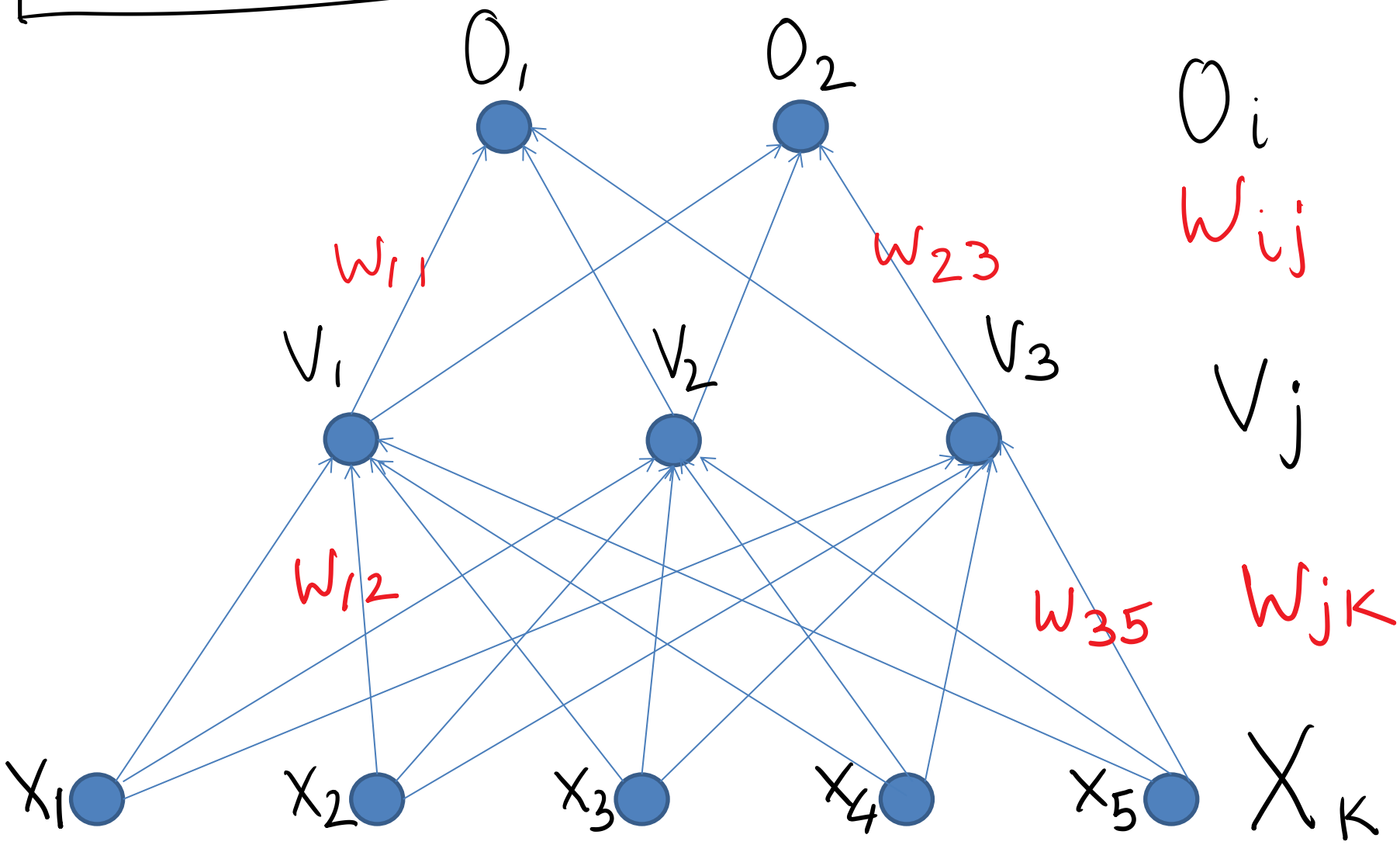
V_1 V_2 $= g\left(\sum_k w_{jk} x_k\right)$



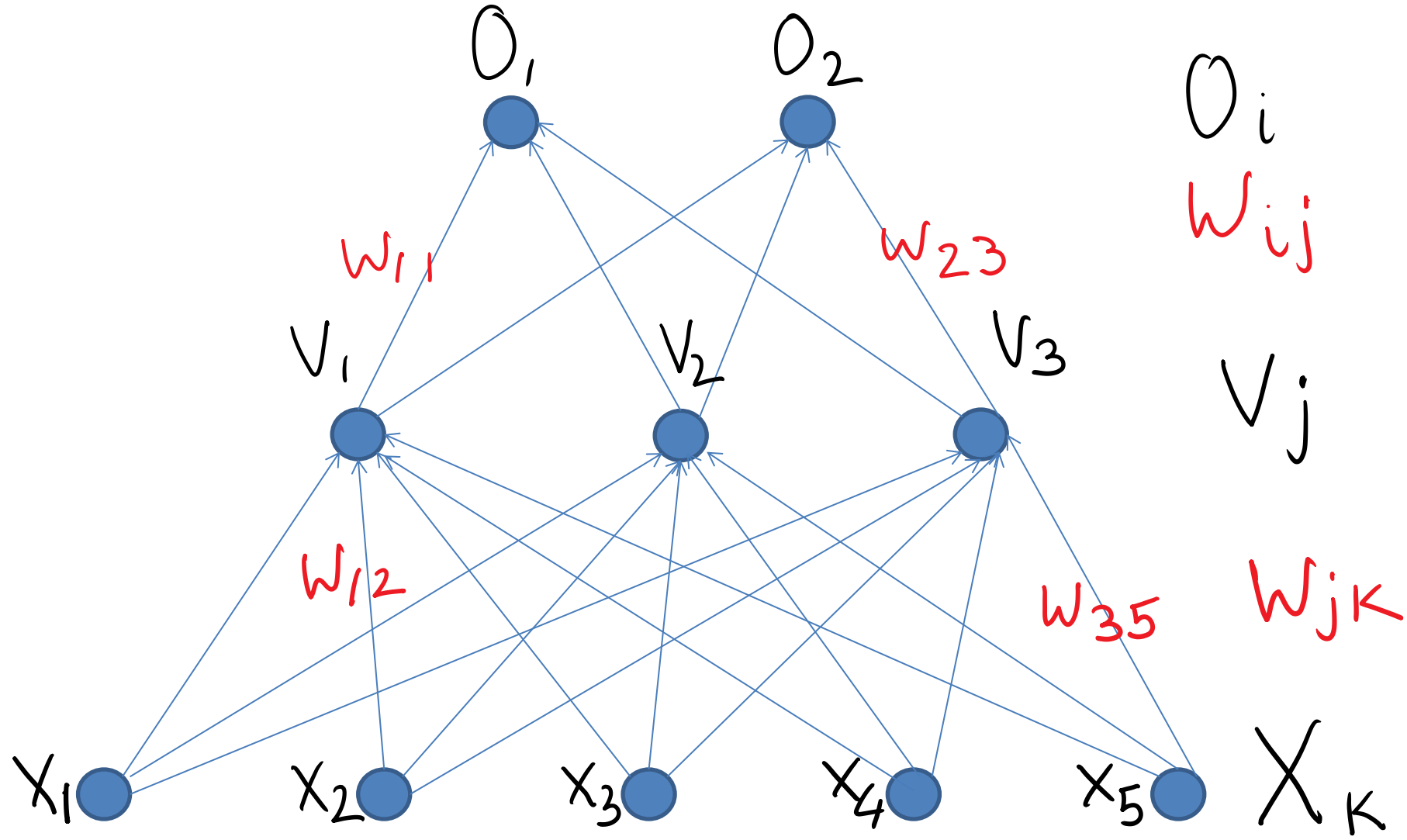
Two layer neural network



$$V_j = g\left(\sum_k w_{jk} x_k\right); \quad O_i = g\left(\sum w_{ij} V_j\right)$$



$$O_i = g\left(\sum_j W_{ij}\, g\left(\sum_k W_{jk}\, X_k\right)\right)$$



O_i

W_{ij}

V_j

W_{jk}

X_k

Training a neural network

Goal: Find W such that O_i is as close as possible to y_i (desired output)

Approach:
- Define loss function $\mathcal{L}(W)$
- Compute $\nabla_W \mathcal{L}$
- $W_{new} \leftarrow W_{old} - \eta \nabla_W \mathcal{L}$

Training a single layer neural network

- A good choice of loss function is the cross entropy

$$L = -\sum_{\text{input data}} \left(y_i \ln O_i + (1-y_i) \ln(1-O_i) \right)$$

- We model the activation function g as a sigmoid

$$g(z) = \frac{1}{1+\exp(-z)}$$

- Finding w reduces to logistic regression!

We can use STOCHASTIC GRADIENT DESCENT.

Training a two layer neural network

(more layers can be trained in the same way)

- We compute the gradient with respect to all the weights: from input to hidden layer, and hidden layer to output layer.
- We can use stochastic gradient descent as before. The loss function is no longer convex, so we can only find local minima. That may be good enough for many applications.
- The complexity of computing the gradient in the naïve version is quadratic in the number of weights. The back-propagation algorithm is a trick that enables it to be computed in linear time.
- We can add a regularization term to penalize large weights; that usually improves the performance.

How do we choose a classifier?

- If we knew the true probability distribution of the features conditioned on the classes, there is a “correct” answer – the Bayes classifier. This minimizes the probability of misclassification.

Bayes Optimal Classifier
(this is unknown in practical situations)

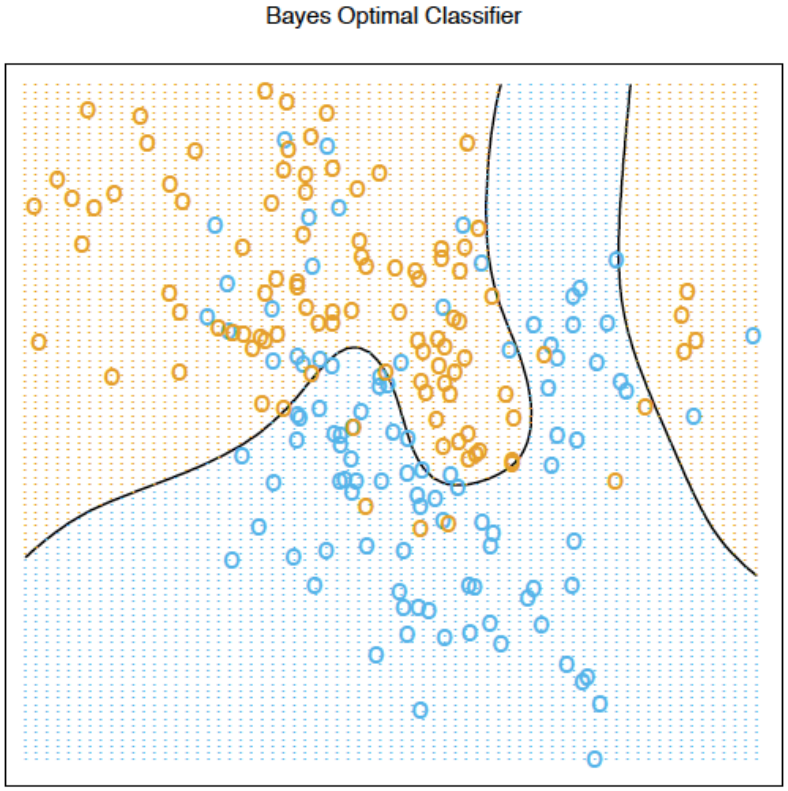


FIGURE 2.5. *The optimal Bayes decision boundary for the simulation example of Figures 2.1, 2.2 and 2.3. Since the generating density is known for each class, this boundary can be calculated exactly (Exercise 2.2).*

Two kinds of error

- Training set error
  - We train a classifier to minimize training set error.
- Test set error
  - At run time, we will take the trained classifier and use it to classify previously unseen examples. The error on these is called test set error.

Validation and Cross-Validation

- If the test set error is much greater than training set error, that is called over-fitting.
- To avoid over-fitting, we can measure error on a held-out set of training data, called the validation set.
- We could divide the data into k-folds, use k-1 of these to train and test on the remaining fold. This is cross-validation