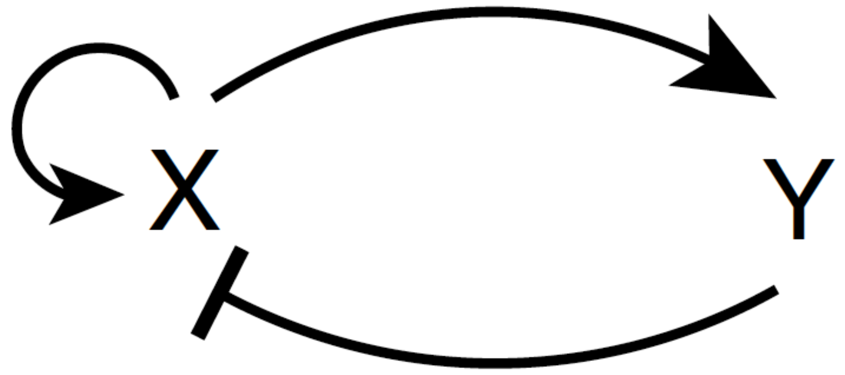
Learning theory and Neural Network models (for GRN and perturbation prediction)

LALEH HAGHVERDI

(P12) Exercises:

a) Show that the following motif can result in oscillation around an attractor.

oscillator motif



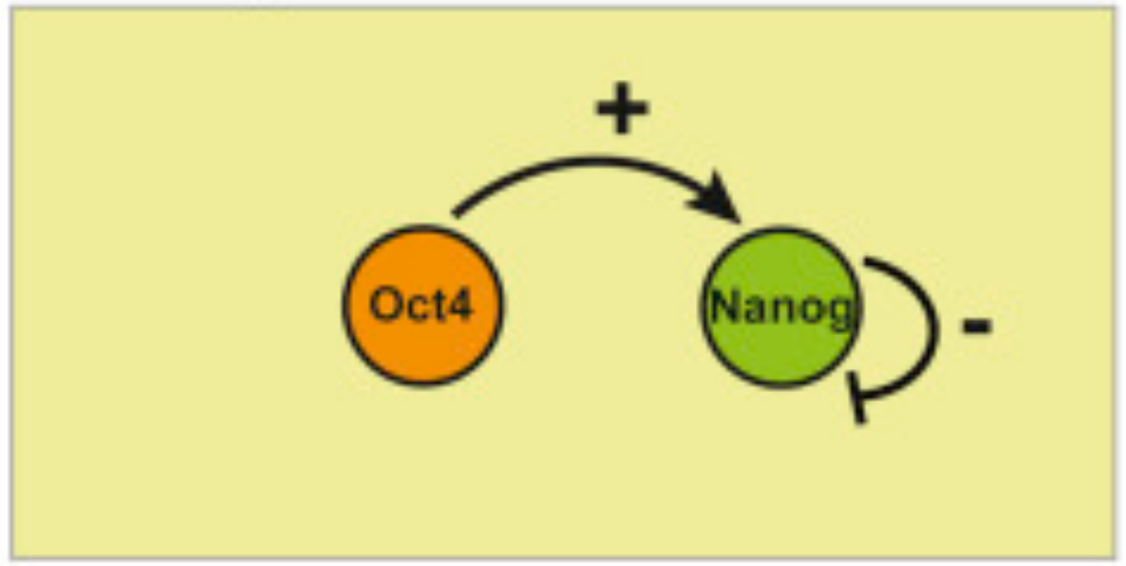
b) Show the conditions on alpha, beta that result in damped oscillation around the attractor

c) What if Y has self activation instead of X?

d) What if both X and Y have self activation?

e) Which of these motifs can be assumed for cell type transitions (i.e., exiting one attractor and entering another)?

f) Analyse the attractor states of the following motif of pluripotency (stemness). Under what parameter ranges is we get an stable attractor?



Direct disruption of core pluripotency transcription motif using CRISPRd

(P14) Exercises:

We saw how a toggle GRN with two hill functions can give rise to two attractors

We also saw oscillator and damped oscillator GRNs

a) Can you make a GRN with chaotic dynamics? E.g. based on the Roessler attractor?

b) Is it likely or unlikely that such a dynamics appears in natural GRNs?

c) How will the dynamics of a chaotic system (e.g. Roessler) be with added noise?

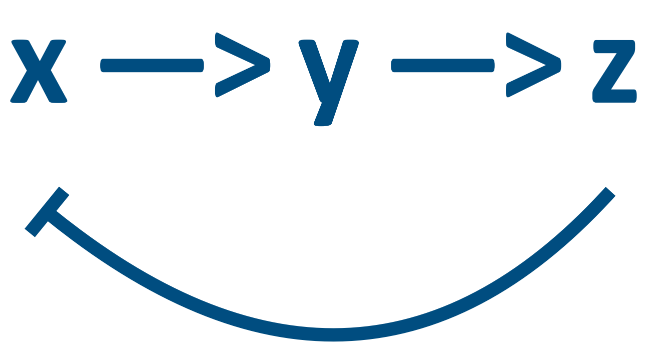
d) How can we characterise the dynamics around an attractor of a D-dimensional GRN? (So far we have considered 2 genes only)

e) Can you by considering D genes, each with a Sigmoid activation function create a 3D chaos system (Roessler, Lorenz, etc.)?

(P19) Exercises:

a) Analyse and simulate (using Beeline and BoolODE) the dynamics of the following GRN and show it can result in cell state transition.

b) What attractor states does this GRN motif have?



(P22) Exercises:

a) Can you suggest another simple model with either improved ground truth capture? Or improved prediction power?

b) Any idea for a nonlinear but still simple model?

Possible project title:

Analysis of different parameter regimes and the resulting dynamics in linear Gene Regulatory Networks approximation with two or three genes.

Exercises:

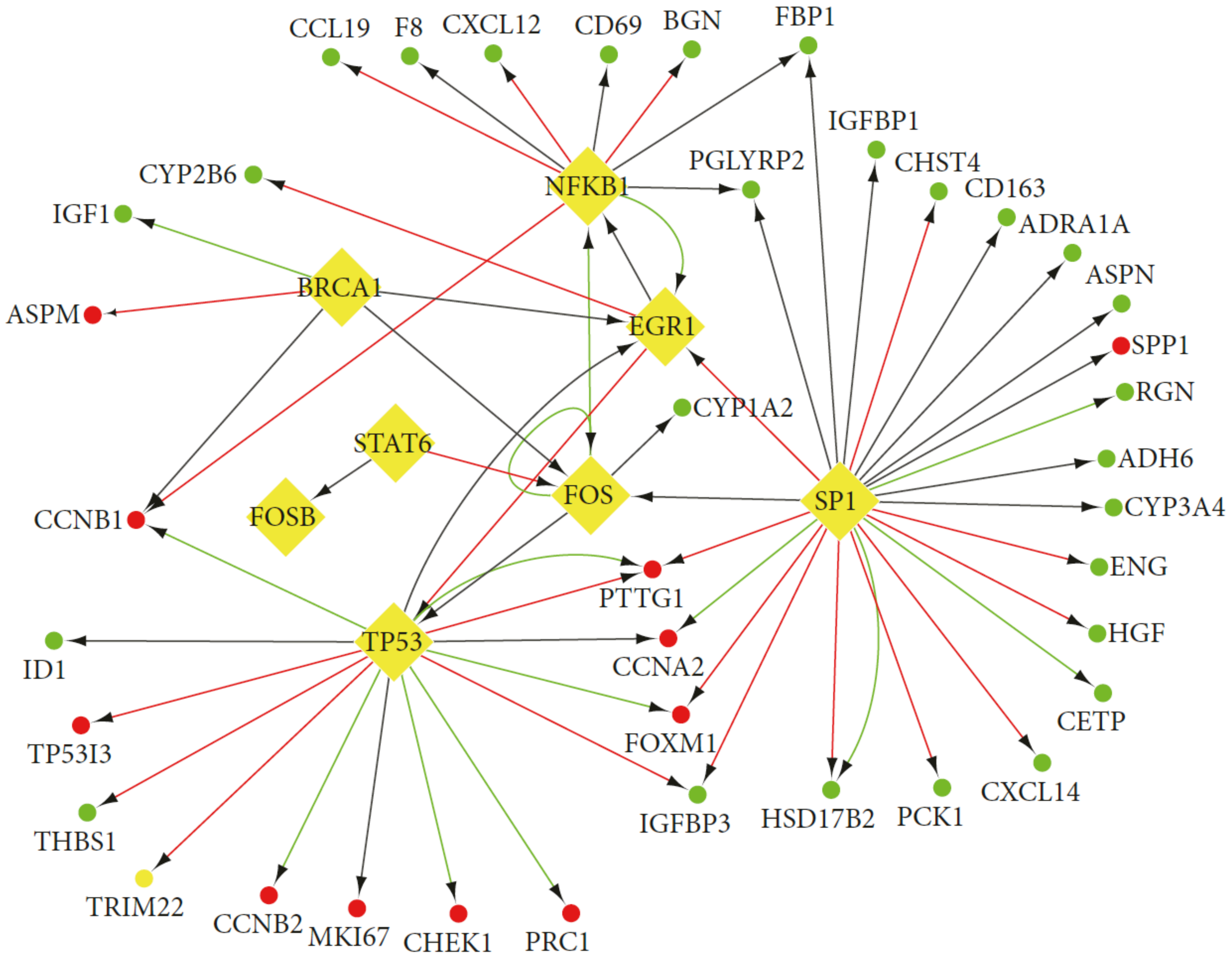
a) How can you make a function with a similar shape to a cooperative Hill function from ReLU (Rectified Linear Unit) functions?

b) Typical Neural Net architectures do not consider interaction terms among the input features explicitly. However, the nonlinearity introduced by activation function implicitly accounts for interaction terms. Can you prove this?

c) Build a minimal NN architecture for modelling the GRN (e.g. among 1000 TFs) of an attractor cell state that captures each gene's activation by cooperative binding of 3 TFs to its promoter, and accounts for autoregulation and degradation as well.
d) Test your method on the Haematopoetic data from CellOracle: Kamimoto, et al. *Nature* (2023).
e) Why does CellOracle model disregards autoregulation and degradation rates?

GRNs and TF networks

- Human and mouse: ~20k genes
- ~4000 TFs that directly bind to DNA
- Lowly expressed to regulate other genes



References

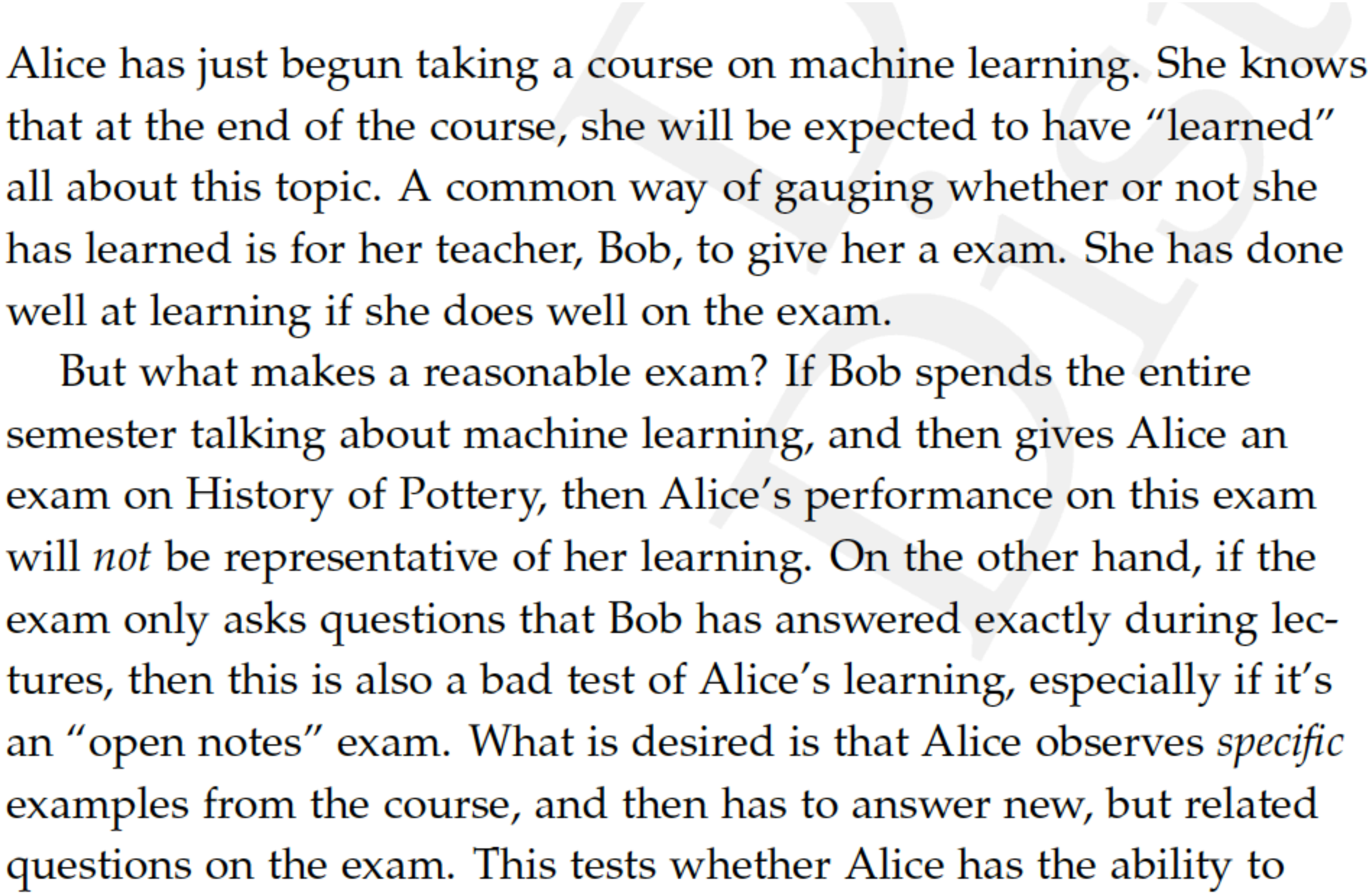
- A Course in Machine Learning,
Hal Daume
https://people.cs.umass.edu/~pinar/ciml-v0_9-ch01.pdf

- Understanding Machine Learning: From Theory to Algorithms,
Shai Shalev-Shwartz and Shai Ben-David
https://www.cs.huji.ac.il/~shais/UnderstandingMachineLearning/understanding-machine-learning-theory-algorithms.pdf

- Dive into deep learning,
Zhang, Aston, Zachary C. Lipton, Mu Li, and Alexander J. Smola.
arXiv preprint arXiv:2106.11342 (2021)

What does it mean to learn?

- Generalisation
- Predictability



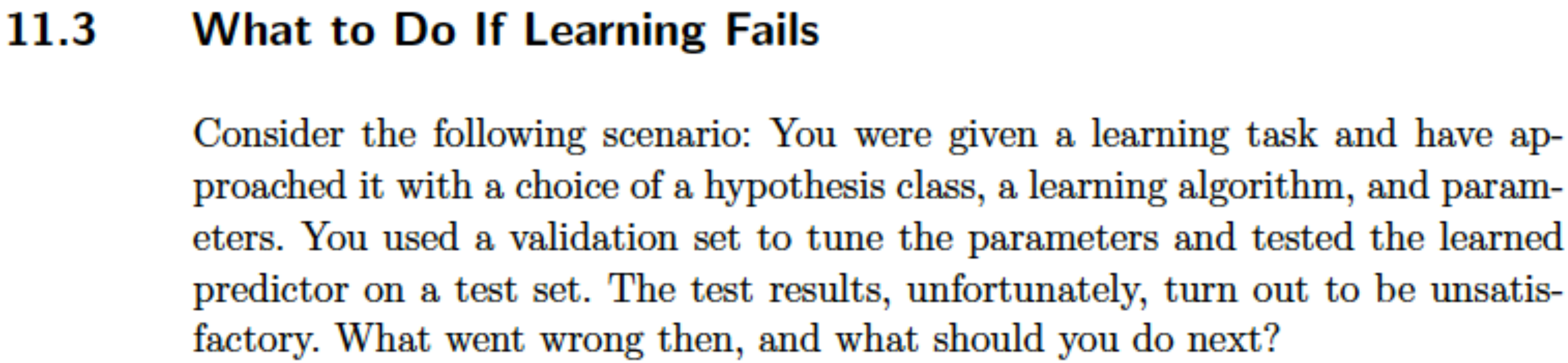
Alice has just begun taking a course on machine learning. She knows that at the end of the course, she will be expected to have “learned” all about this topic. A common way of gauging whether or not she has learned is for her teacher, Bob, to give her a exam. She has done well at learning if she does well on the exam.

But what makes a reasonable exam? If Bob spends the entire semester talking about machine learning, and then gives Alice an exam on History of Pottery, then Alice’s performance on this exam will *not* be representative of her learning. On the other hand, if the exam only asks questions that Bob has answered exactly during lectures, then this is also a bad test of Alice’s learning, especially if it’s an “open notes” exam. What is desired is that Alice observes *specific* examples from the course, and then has to answer new, but related questions on the exam. This tests whether Alice has the ability to

Data devisions: Train, Validation, Test set

- The performance of the learning algorithm should be measured on unseen “test” data
- The training set should be a good representative of the test set.
  - Random sampling
  - Large enough training set

Learning Theory



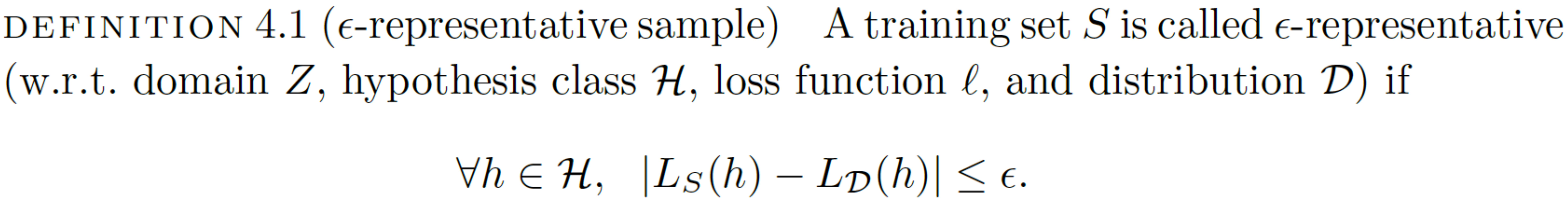
11.3 What to Do If Learning Fails

Consider the following scenario: You were given a learning task and have approached it with a choice of a hypothesis class, a learning algorithm, and parameters. You used a validation set to tune the parameters and tested the learned predictor on a test set. The test results, unfortunately, turn out to be unsatisfactory. What went wrong then, and what should you do next?

- Learning Theory
- Understand the cause of bad performance

Learning Theory

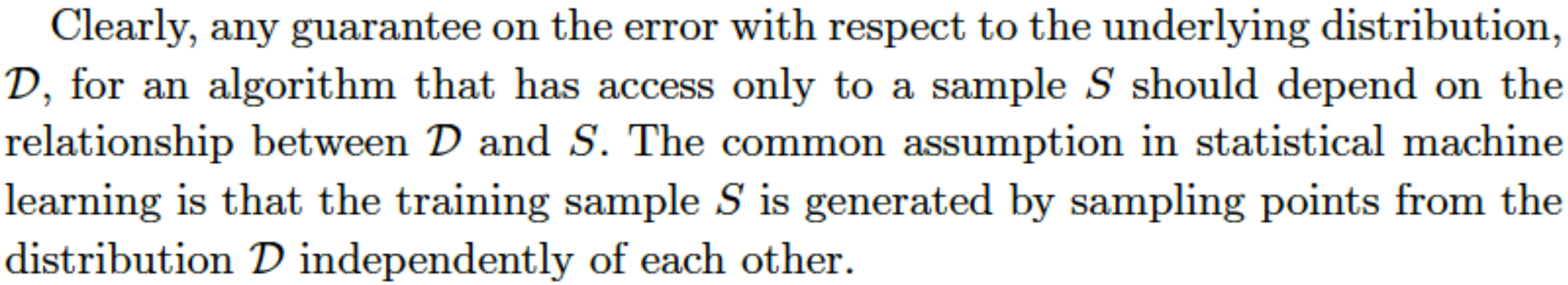
- Unknown distribution of all available data out there $\mathcal{D}$
- Unknown target function f
- Training set $S : \mathcal{X} \times \overline{\mathcal{Y}}$
- Hypothesis class $\mathcal{H}$
- Model $h_S : \mathcal{X} \rightarrow \mathcal{Y}$



DEFINITION 4.1 (ϵ-representative sample) A training set S is called ϵ-representative (w.r.t. domain Z, hypothesis class $\mathcal{H}$, loss function ℓ, and distribution $\mathcal{D}$) if

$$\forall h \in \mathcal{H}, \quad |L_S(h) - L_{\mathcal{D}}(h)| \leq \epsilon.$$

Independently and Identically Distributed (iid) according to the true data distribution D

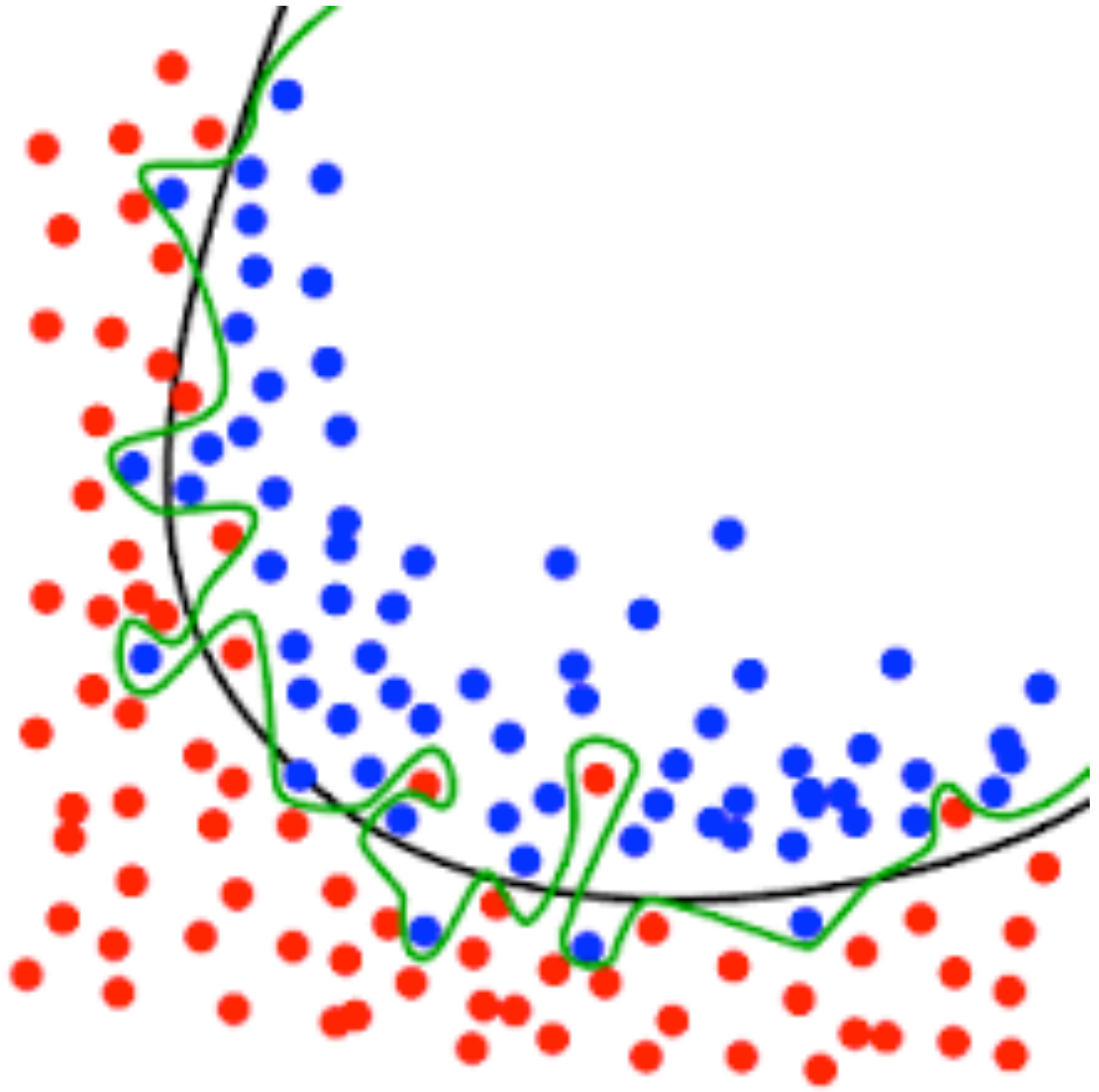


Clearly, any guarantee on the error with respect to the underlying distribution, $\mathcal{D}$, for an algorithm that has access only to a sample S should depend on the relationship between $\mathcal{D}$ and S. The common assumption in statistical machine learning is that the training sample S is generated by sampling points from the distribution $\mathcal{D}$ independently of each other.

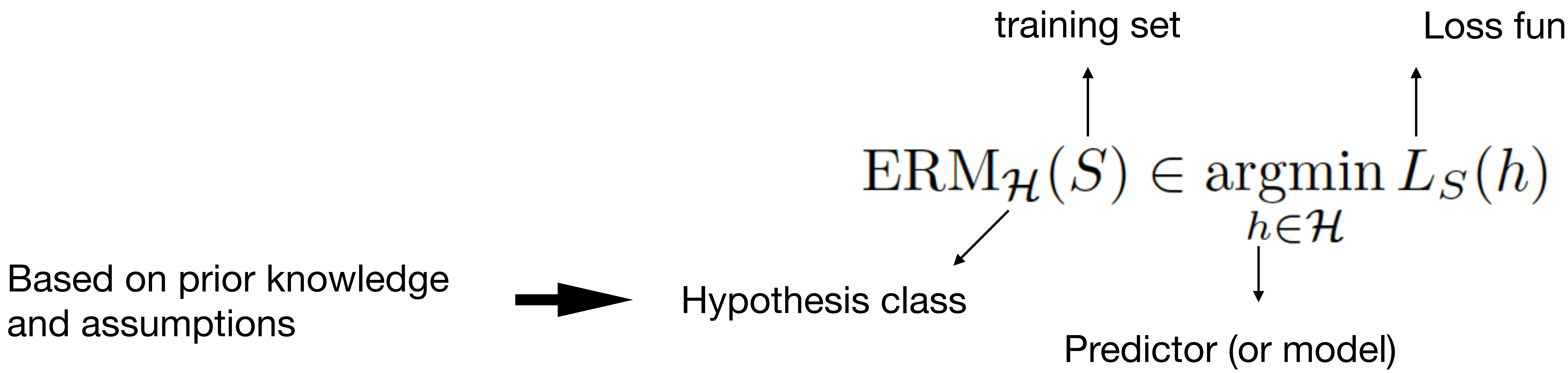
Exercises:

a) Can you calculate the error dependence in e-representativeness of iid data with as a function of sample size ?
b) How well does iid sampling represent minority groups in data sampling?
c) Do you know any problem where iid sampling is not recommended?

Learning by Empirical Risk Minimization (ERM)



- Hypothesis class (set of possible models)
- Loss (cost, risk, …) function
- Model



training set

Loss function

$$\mathrm{ERM}_{\mathcal{H}}(S) \in \operatorname*{argmin}_{h \in \mathcal{H}} L_S(h)$$

Based on prior knowledge and assumptions

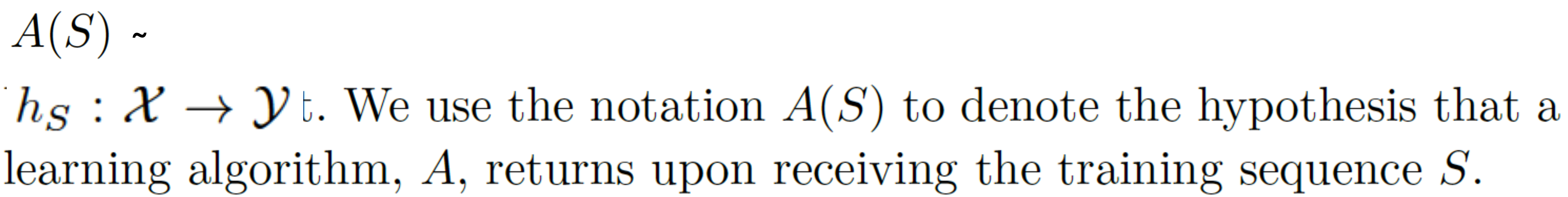
Hypothesis class

Predictor (or model)

Exercises:

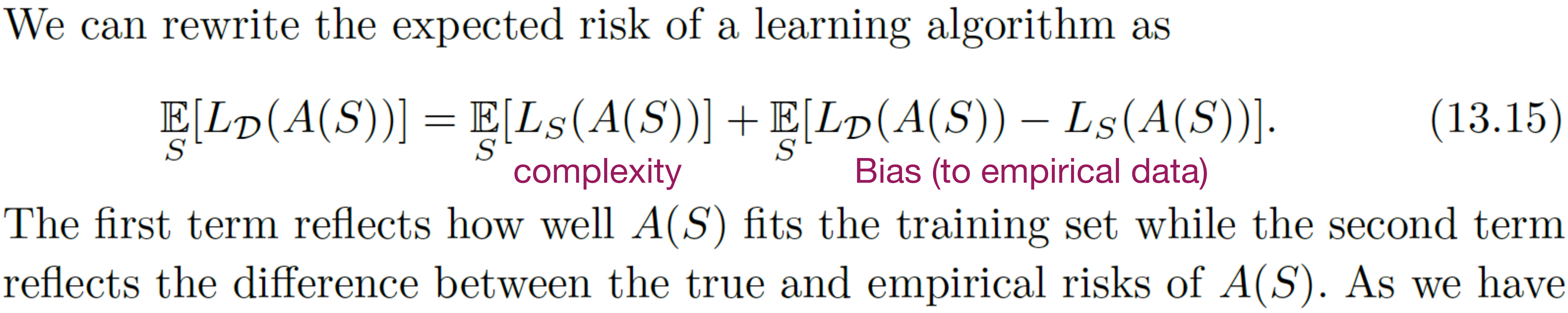
a) What is learning by Structural Risk Minimisation (SRM)?
b) Do you know any example of SRM learning?

Bias-Complexity tradeoff



$A(S)$ ~

$h_S : \mathcal{X} \to \mathcal{Y}$ t. We use the notation $A(S)$ to denote the hypothesis that a learning algorithm, A, returns upon receiving the training sequence S.



We can rewrite the expected risk of a learning algorithm as

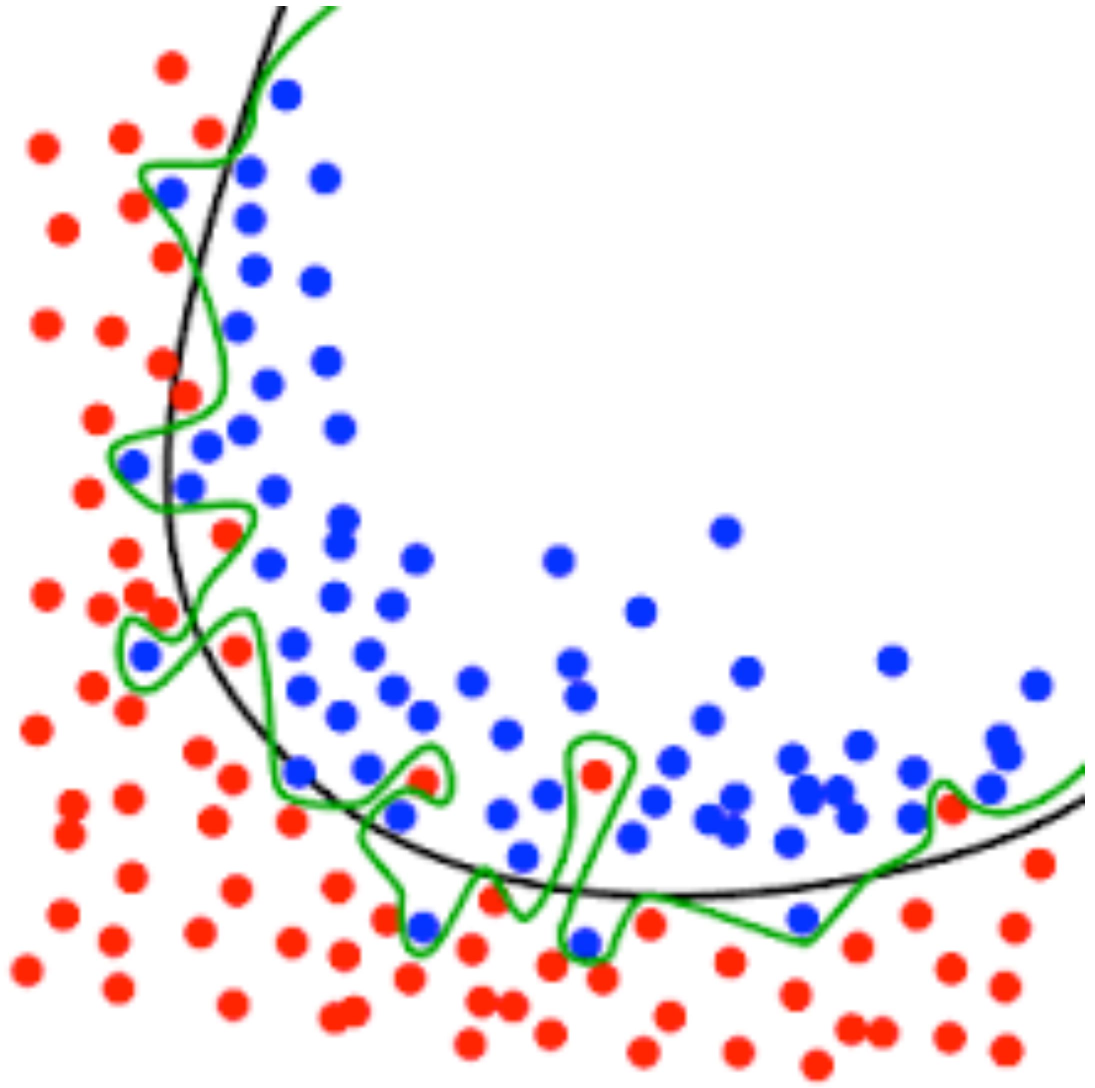
$$\mathop{\mathbb{E}}_{S}[L_{\mathcal{D}}(A(S))] = \mathop{\mathbb{E}}_{S}[L_S(A(S))] + \mathop{\mathbb{E}}_{S}[L_{\mathcal{D}}(A(S)) - L_S(A(S))]. \qquad (13.15)$$

complexity

Bias (to empirical data)

The first term reflects how well $A(S)$ fits the training set while the second term reflects the difference between the true and empirical risks of $A(S)$. As we have

Challenges in Learning

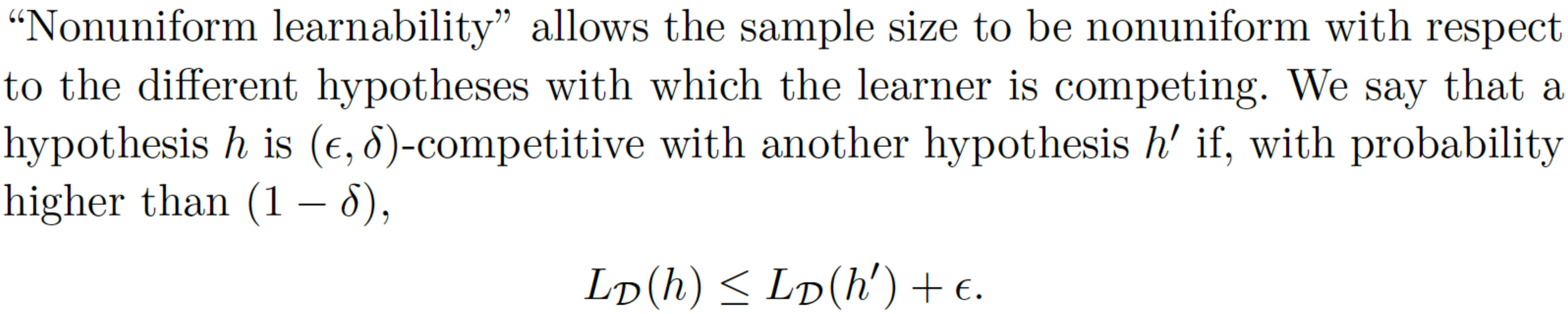


- Overfitting
  - Finite hypothesis class
  - Regularization
- Realizability: with the given feature set, can a predictor be constructed?
- Identifiability
- Convex optimisation
- S adequately represents D?

Exercises:

a) What measures and statistics do you need to gather to check the performance of a model with respect to each bullet point ?

Unidentifiability



"Nonuniform learnability" allows the sample size to be nonuniform with respect to the different hypotheses with which the learner is competing. We say that a hypothesis h is (ϵ, δ)-competitive with another hypothesis h' if, with probability higher than $(1 - \delta)$,

$$L_{\mathcal{D}}(h) \leq L_{\mathcal{D}}(h') + \epsilon.$$

Exercises:

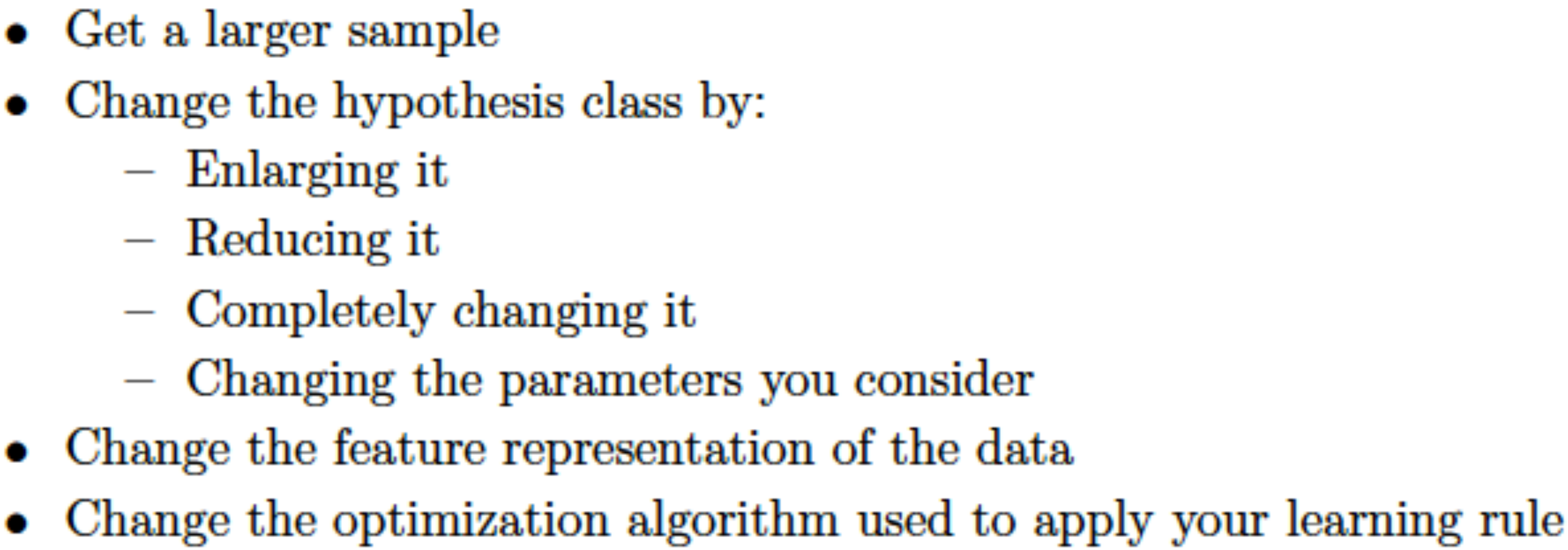
a) Assume you have learned a model h on a training set data S. How can you test if there are other closely competing models available?

b) What is cross validation? And how does it help to consider unidentifiability issues for a learning problem?

c) Can adding noise to the data also reveal existence of unidentifiability?

d) Show your propositions on a data set under an unidentifiable model. E.g. linear regression X . W = Y , solving for W, when X and Y are given, with corresponding matrix sizes [N, G], [G,G], [N, G] with N<<G.

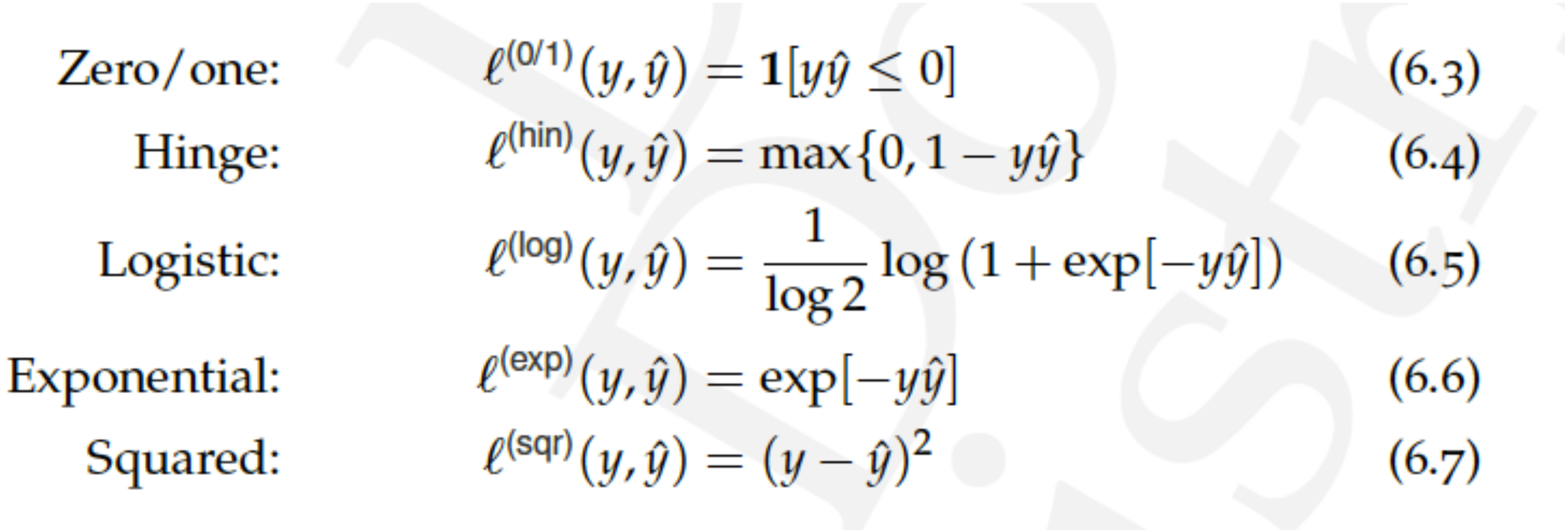
Some of the things to do if learning fails



- Get a larger sample
- Change the hypothesis class by:
  - Enlarging it
  - Reducing it
  - Completely changing it
  - Changing the parameters you consider
- Change the feature representation of the data
- Change the optimization algorithm used to apply your learning rule

Learning components

- Train set
- The Model, e.g. $\hat{y} = \boldsymbol{w} \cdot \boldsymbol{x} + b$
- Loss (also known as Cost, Risk or Penalty) function



$$\text{Zero/one:} \quad \ell^{(0/1)}(y, \hat{y}) = \mathbf{1}[y\hat{y} \le 0] \tag{6.3}$$

$$\text{Hinge:} \quad \ell^{(\text{hin})}(y, \hat{y}) = \max\{0, 1 - y\hat{y}\} \tag{6.4}$$

$$\text{Logistic:} \quad \ell^{(\log)}(y, \hat{y}) = \frac{1}{\log 2} \log\left(1 + \exp[-y\hat{y}]\right) \tag{6.5}$$

$$\text{Exponential:} \quad \ell^{(\exp)}(y, \hat{y}) = \exp[-y\hat{y}] \tag{6.6}$$

$$\text{Squared:} \quad \ell^{(\text{sqr})}(y, \hat{y}) = (y - \hat{y})^2 \tag{6.7}$$

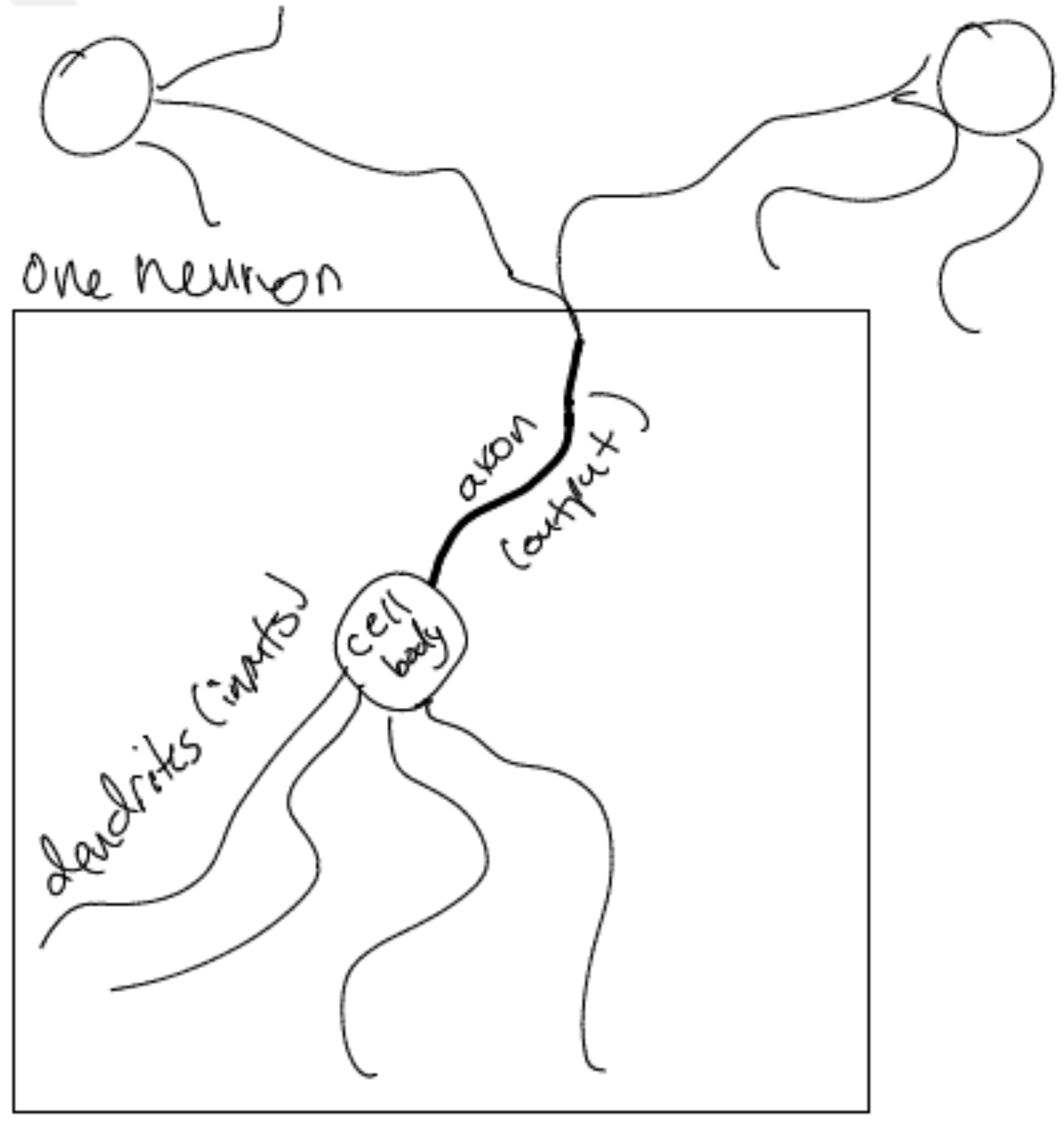
Perceptron

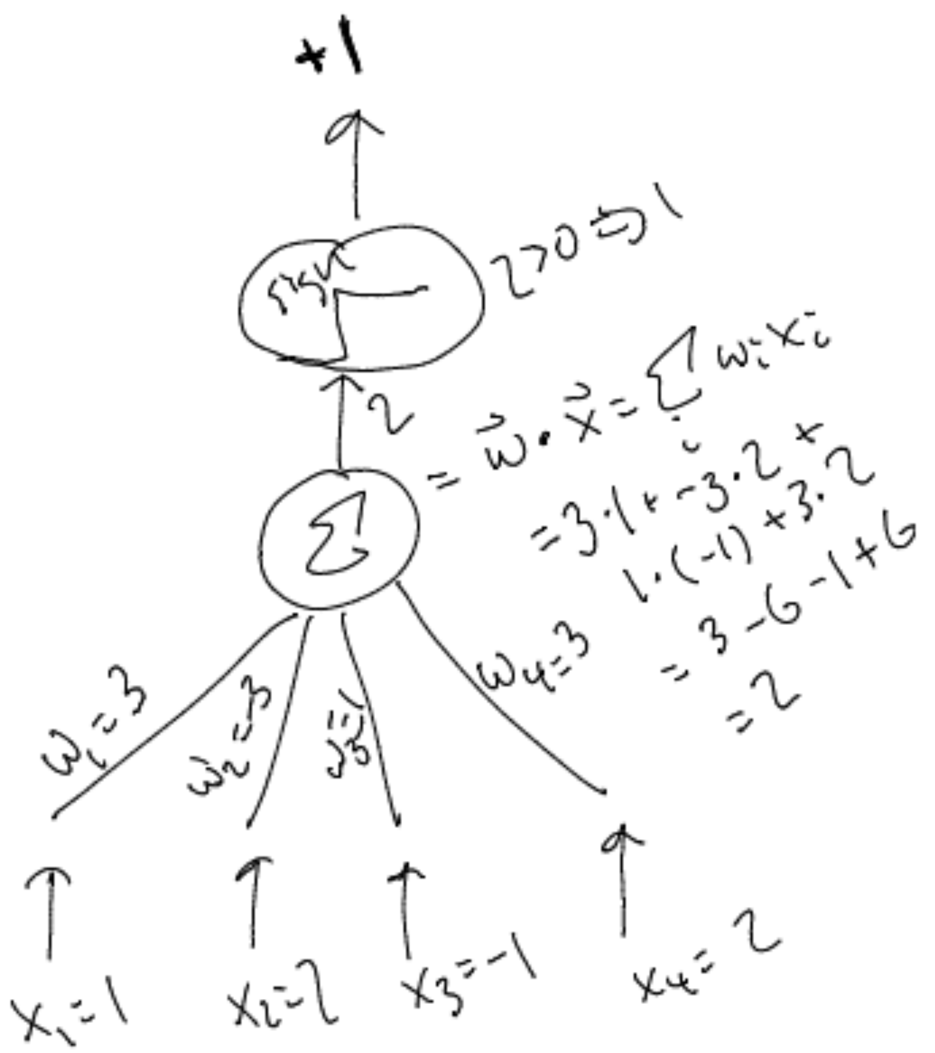
- A single (linear) unit of Neural Networks

- Activation is calculated as:

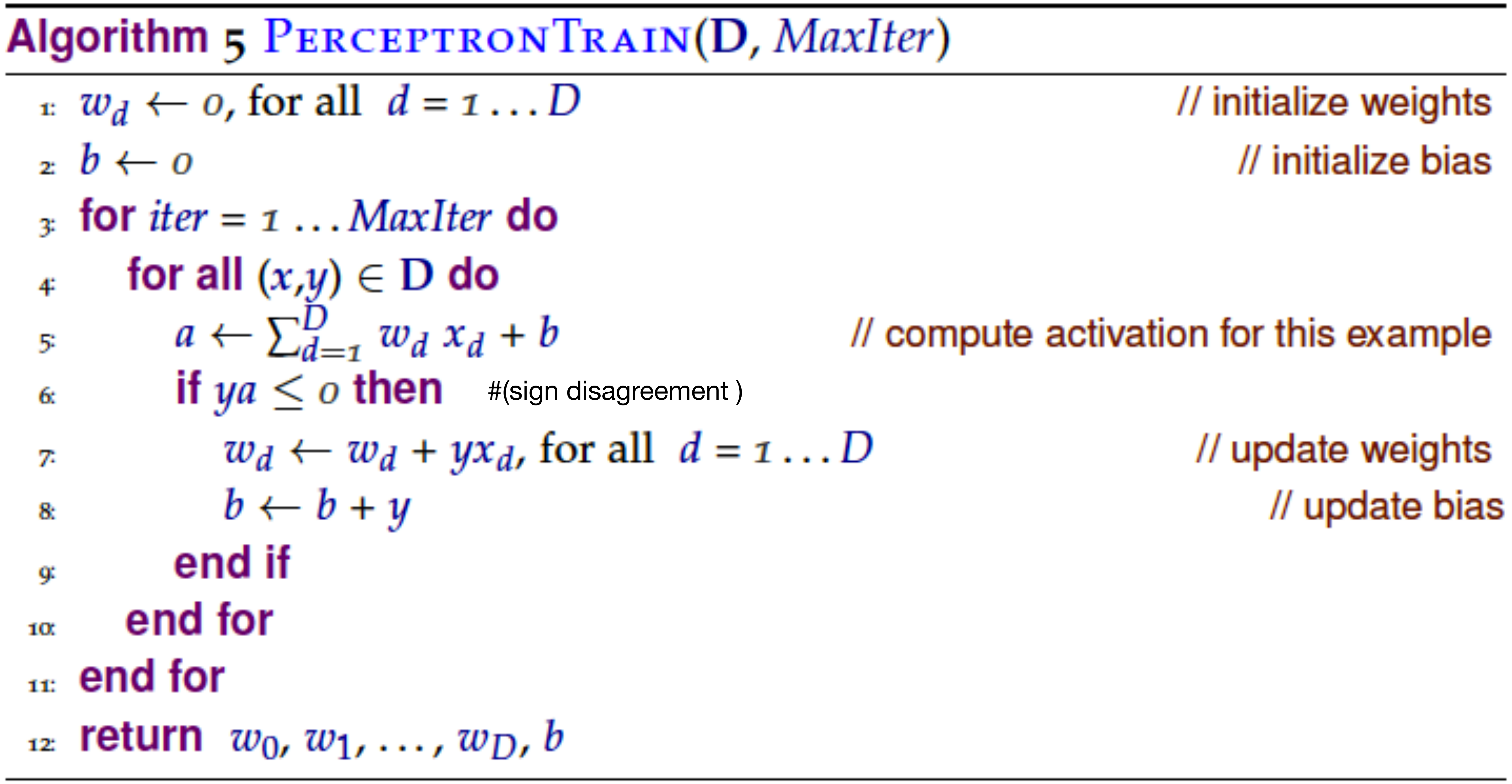
$$a = \left[\sum_{d=1}^{D} w_d x_d\right] + b$$

$$\hat{y} = sign(\sum_{i=1}^{D} w_i x_i + b)$$



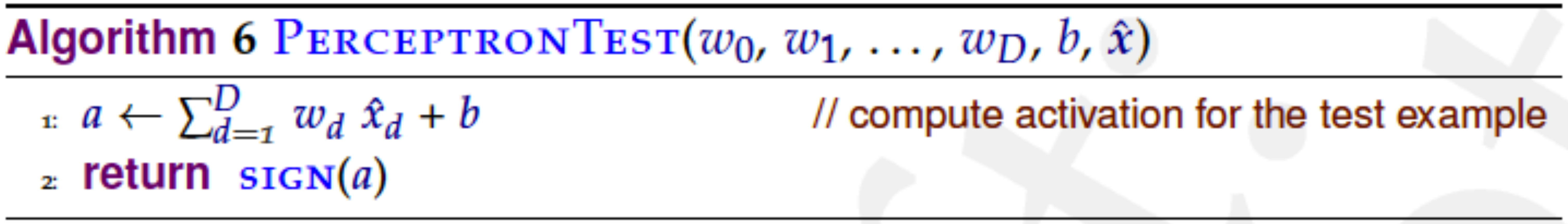


Single-layer Perceptron (for binary classification)



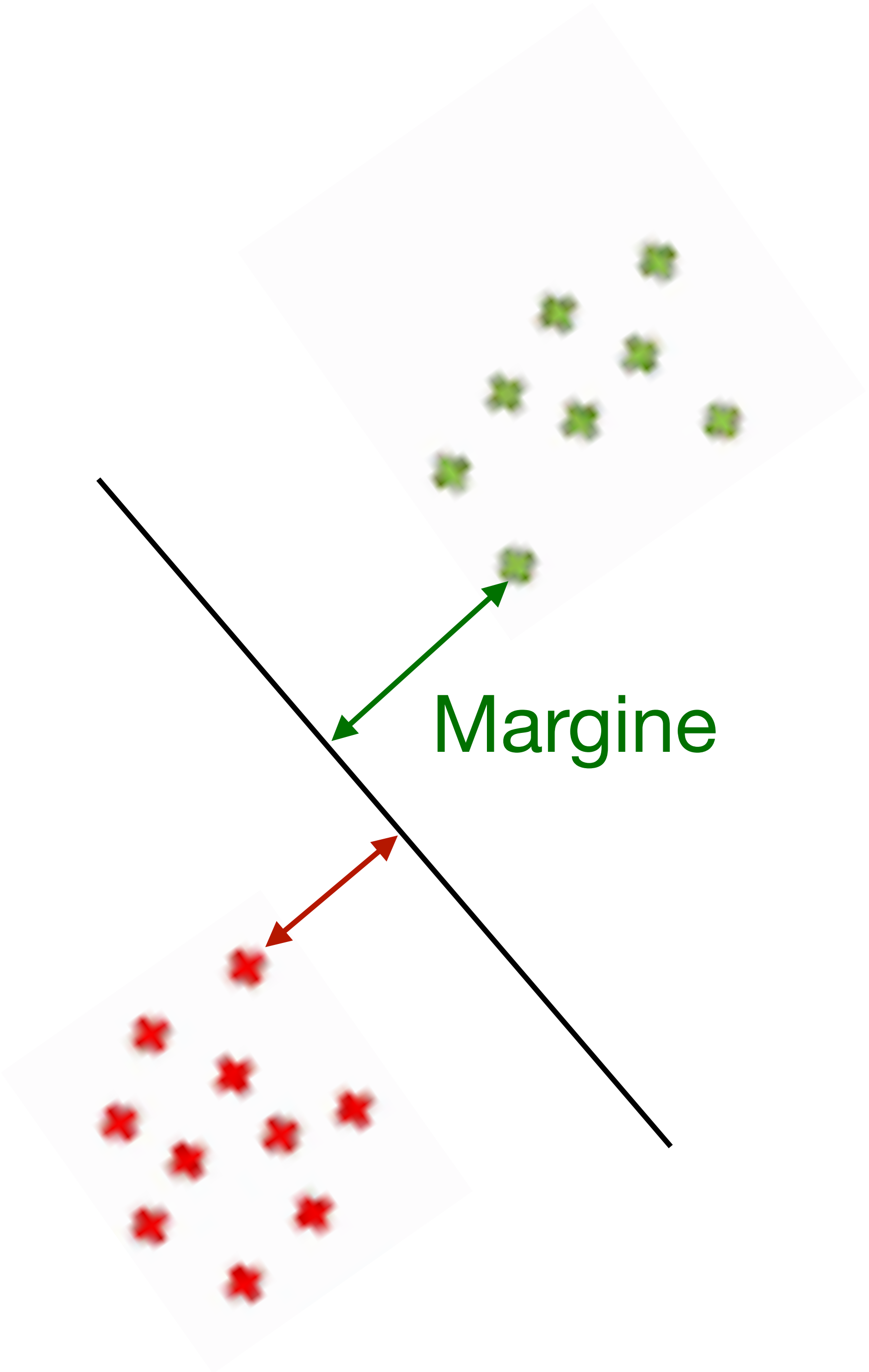
Algorithm 5 PerceptronTrain(D, *MaxIter*)

```
$w_d \leftarrow 0$, for all $d = 1 \dots D$                      // initialize weights
$b \leftarrow 0$                                                  // initialize bias
for $iter = 1 \dots MaxIter$ do
    for all $(x,y) \in$ D do
        $a \leftarrow \sum_{d=1}^{D} w_d\, x_d + b$                // compute activation for this example
        if $ya \leq 0$ then     #(sign disagreement )
            $w_d \leftarrow w_d + y x_d$, for all $d = 1 \dots D$  // update weights
            $b \leftarrow b + y$                                   // update bias
        end if
    end for
end for
return $w_0, w_1, \dots, w_D, b$
```



Algorithm 6 PerceptronTest($w_0, w_1, \dots, w_D, b, \hat{x}$)

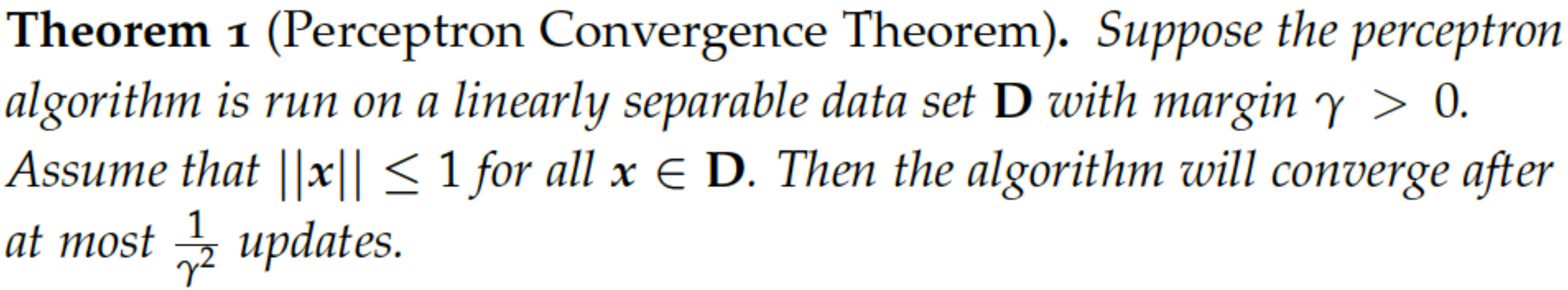
```
$a \leftarrow \sum_{d=1}^{D} w_d\, \hat{x}_d + b$     // compute activation for the test example
return sign($a$)
```



Exercises:

a) Prove that the update line
   - increases activation, if activation<0 and y=1
   - decreases activation, if activation>0 and y=-1

b) Prove the perceptron convergence theorem



Theorem 1 (Perceptron Convergence Theorem). *Suppose the perceptron algorithm is run on a linearly separable data set* **D** *with margin* $\gamma > 0$. *Assume that* $||\boldsymbol{x}|| \leq 1$ *for all* $\boldsymbol{x} \in \mathbf{D}$. *Then the algorithm will converge after at most* $\frac{1}{\gamma^2}$ *updates.*

Challenges in modelling GRNs from single-cell omics

- Too many features
- Missing components
- Highly complex interactions
- No real time-series data, only snapshots (so no LLMs)
- Non-deterministic GRNs
    - uncertainty models

Exercises:

a) What data with what quality would you ask the Human Cell Atlas (HCA) consortium to produce for a powerful predictive/generative model of GRNs and perturbation effects?
b) How could you ensure that rare cell types are well considered in the model?
c) Is there a smart way of sampling perturbation data or do you recommend an iid sampling of all genes perturbation?