Applied Deep Learning

Dr. Philippe Blaettchen
Bayes Business School (formerly Cass)

www.bayes.city.ac.uk

Learning objectives of today

Goals: Be ready for your group assignment

How will we do this?

- We will learn and practice another way to tune hyperparameters
- Then, we will take a look at autoencoders, a specific type of neural network
- Finally, we will discuss the case underlying the assignment, as well as some of the key challenges

BAYES
BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

Hyperparameter tuning with Keras tuner

Let’s try it together in Python

BAYES
BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

Hyperparameter tuning process with Keras Tuner

1. Define a function that, given a hyperparameter-setter, creates a model
   - Within that function, using the hyperparameter-setter, we define the hyperparameter space

2. Define an instance of the Keras Tuner, specifying the type of hyperparameter search
   - Can use RandomSearch, Hyperband, Sklearn, BayesianOptimization

BAYES
BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

Hyperparameter tuning process with Keras Tuner

1. Define a function that, given a hyperparameter-setter, creates a model
   Within that function, using the hyperparameter-setter, we define the hyperparameter space

2. Define an instance of the Keras Tuner, specifying the type of hyperparameter search
   Can use RandomSearch, **Hyperband**, Sklearn, BayesianOptimization

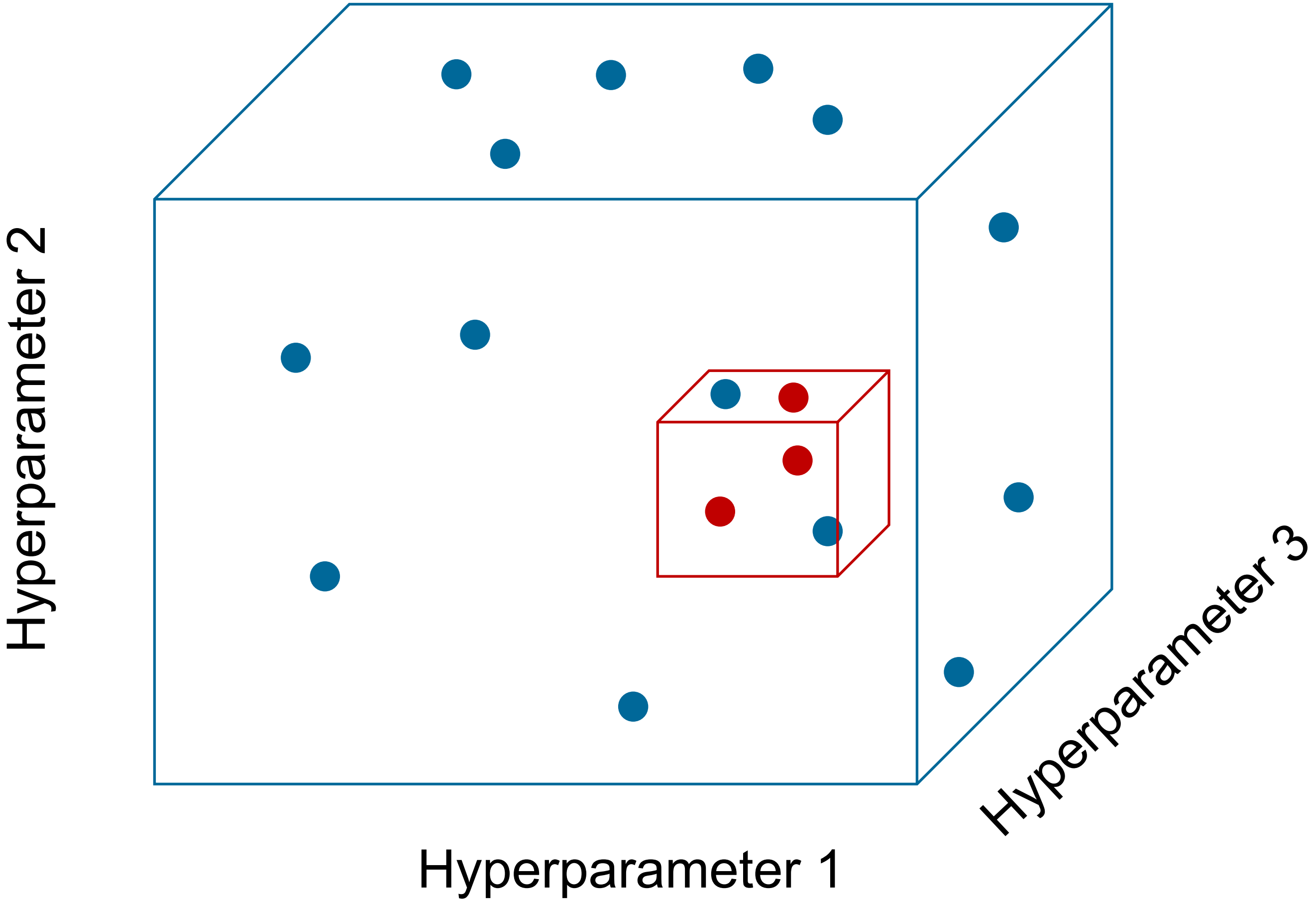
3. Let the tuner do its magic
   The hyperparameter-setter will automatically choose the “correct” hyperparameters

4. Based on the best parameters found, generate a model, train it, and evaluate it

5. Look at the outcome, possibly search in a smaller grid

BAYES
BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

Hyperparameter space



BAYES BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

Autoencoders

What is an autoencoder?

- A neural network that predicts its own inputs
  → So that we can learn a (compact) representation of the data

BAYES
BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

Recall that a neural network learns representations

x

Learn $f(\cdot)$

$f(x)$

Learn $\mathrm{g}(\cdot)$

$g\big(f(x)\big) \approx y$

E.g., $y = 1$, if it's a cat
$y = 0$, if it's a BA student

BAYES
BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

Recall that a neural network learns representations

x

Learn $f(\cdot)$

$f(x)$

Learn $\mathrm{g}(\cdot)$

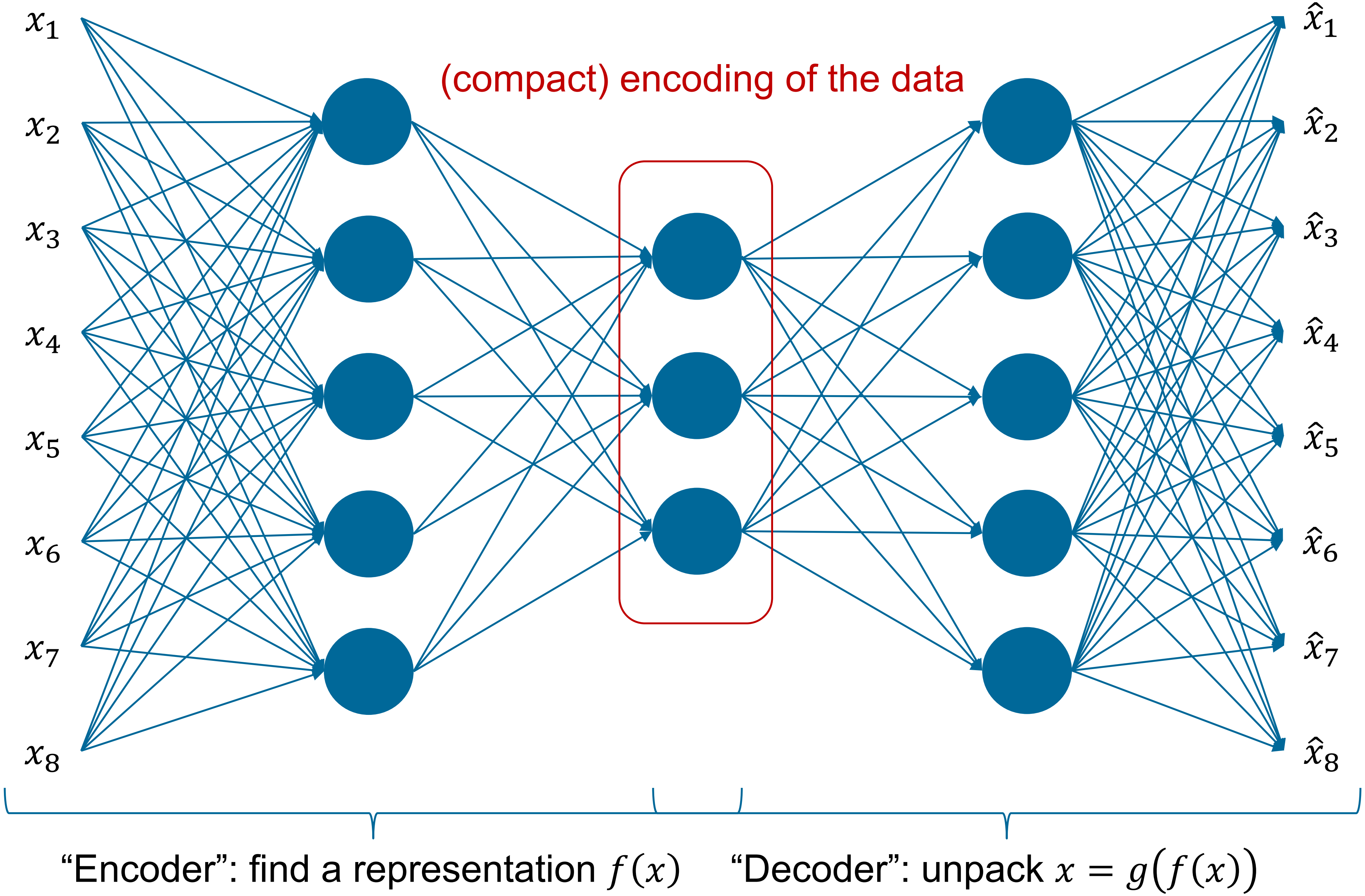
$g(f(x)) \approx x$

Why do we want to “copy” the input?

- We don’t care about the copy itself (which should be good, nevertheless)
- What we care about is a (compact) representation of the data, $f(x)$ that is good enough to recreate x

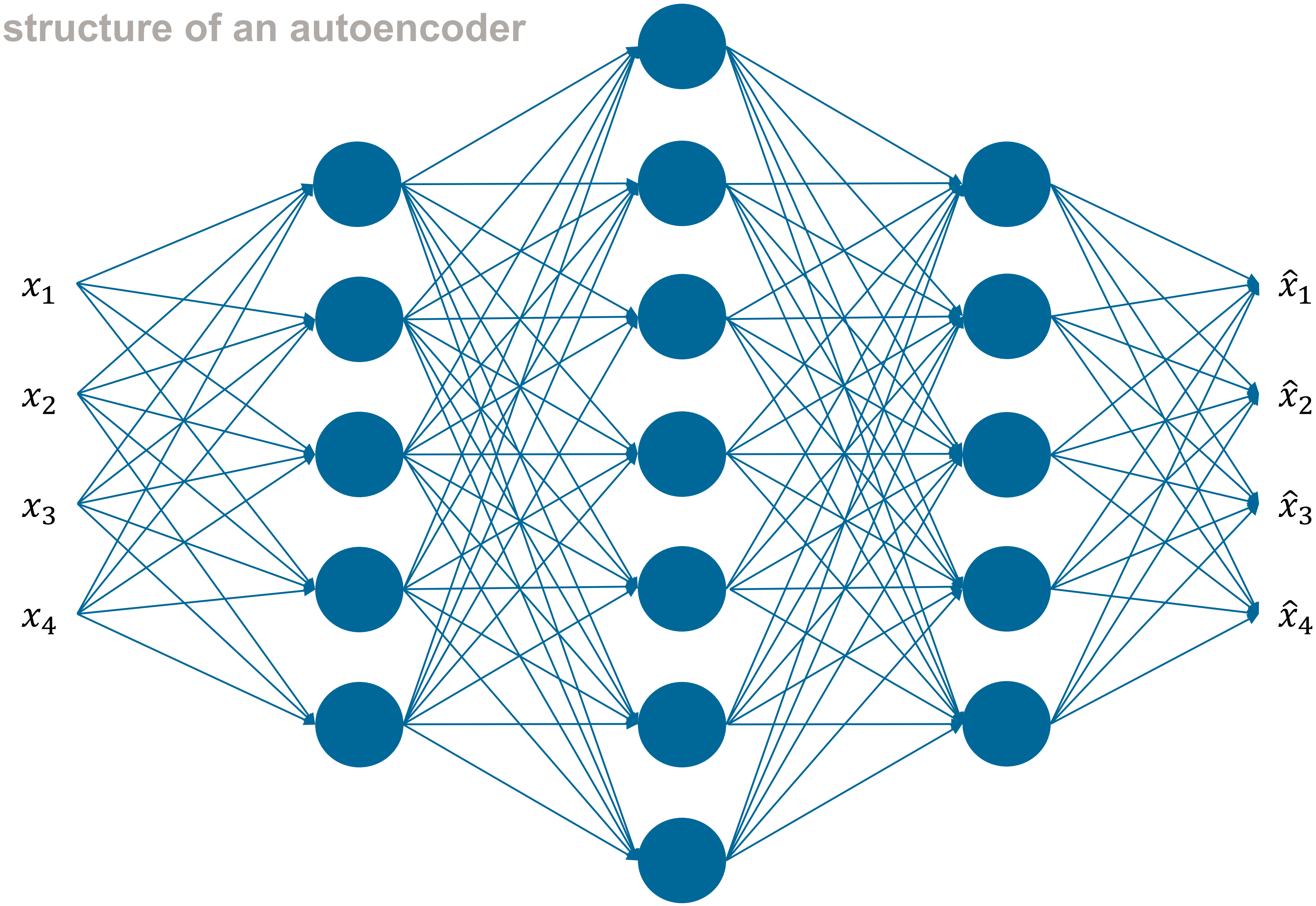
BAYES
BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

The structure of an autoencoder



BAYES BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

NOT the structure of an autoencoder



BAYES
BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

Let's try it together in Python

BAYES
BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

What to use it for?

- Dimensionality reduction ("advanced PCA")
- **Denoising: train to "recover" non-noisy data from noisy data**

- Anomaly detection: train to represent normal data. When data cannot be predicted well, it is likely to be "anormal"
- Generate new content (such as images): variational autoencoders

The process for denoising

- We create artificial noise on our data

- We build an autoencoder that takes the noisy data as input, and tries to build an accurate representation of the original
  - To do so with images, we need convolutional layers. Don't worry about how they work, we will get to them soon. You find all the code on using them in the notebook!
  - Training the autoencoder may take quite a bit of time! (I suggest first getting to the training part, then taking a break in the meantime)

- We then can run the autoencoder on new (noisy) data, to create non-noisy versions

BAYES
BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

Try it out in Python

An overview of the group assignment

The group assignment

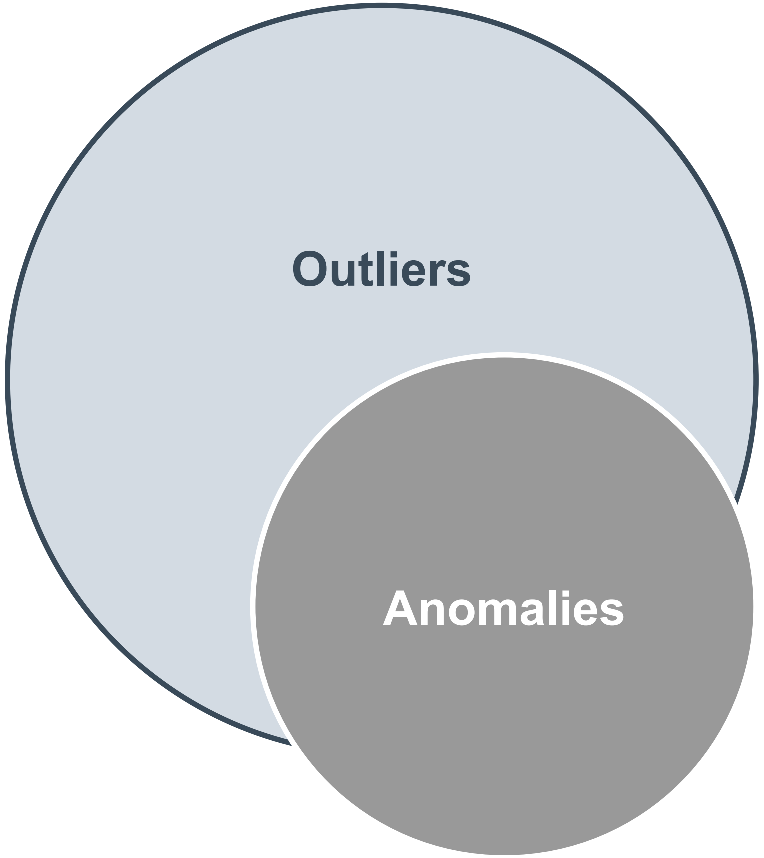
- "To Catch a Thief": A case about "Shift", an InsureTech startup that helps insurance companies uncover fraudulent claims
  - In particular, their algorithm flags suspicious cases for claim handlers to investigate

- We have a dataset of vehicle insurance claims. One issues is that it is "unbalanced"
  → only about 1% of cases are frauds
  → What would be the accuracy of an algorithm that always predicts "no fraud"?

- Your tasks:
  - Pre-processing
  - Understanding the relevant metrics
  - Trialing and comparing models of varying complexity
  - Exploring autoencoders as a tool to detect anormal data
  - Discussing transparency implications of neural networks

Anomaly detection with autoencoders

Outliers and anomalies



Outliers:

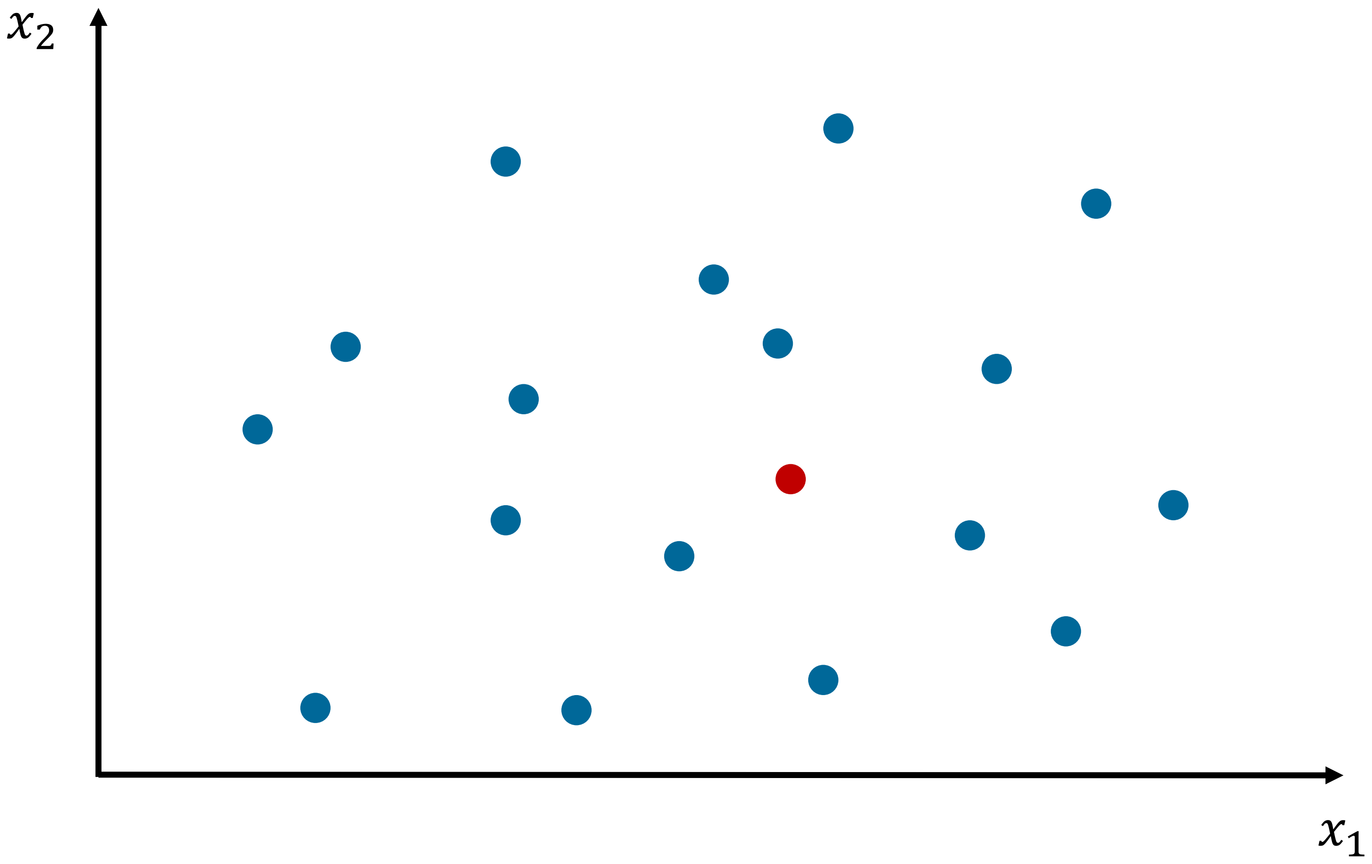
- Data points that are distinctly different from other data points
- Can be caused by unavoidable random errors or by systematic errors relating to how data was sampled

Anomalies:

- Outliers or other values that are not expected to exist
- Can be context- or pattern-based:
    - Context: exceptionally high credit card spending on Black Friday versus near-simultaneous spending in New York and London
    - Pattern: high credit card spending every Saturday versus high spending on a day where spending is low in other weeks

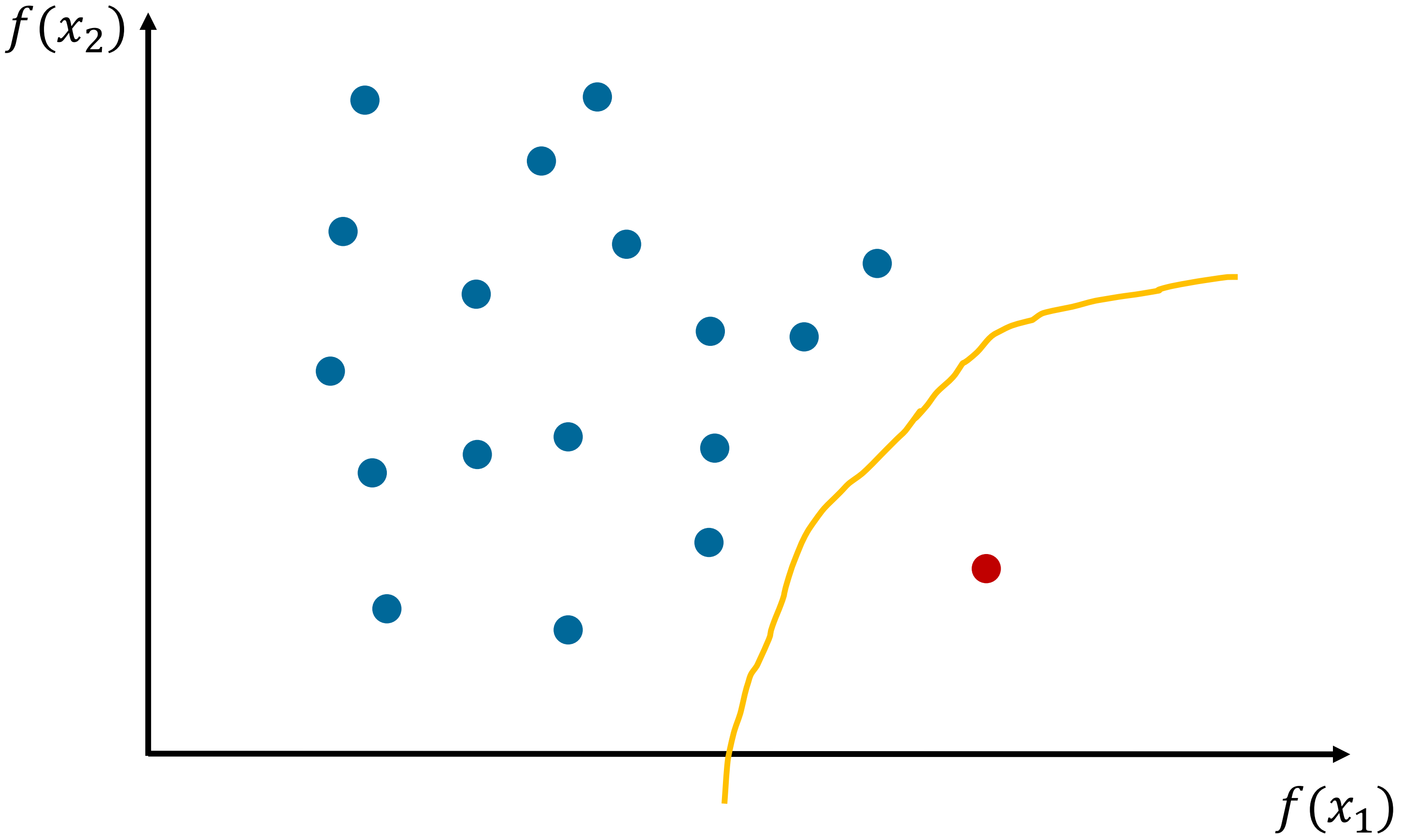
BAYES BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

Differentiating anomalies from normal observations



BAYES
BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

Differentiating anomalies from normal observations



What are possible anomalies and how would we detect them?

Consider the following situations:

- A machine produces thousands of screws per minute, every few days the type of screw is changed
- A software developer for a bank downloads a large number of entries from a customer database
- An intermediary supplies fair trade coffee beans

What is the expected outcome in each case?

What is an anomalous outcome?

What data do we observe?

BAYES BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

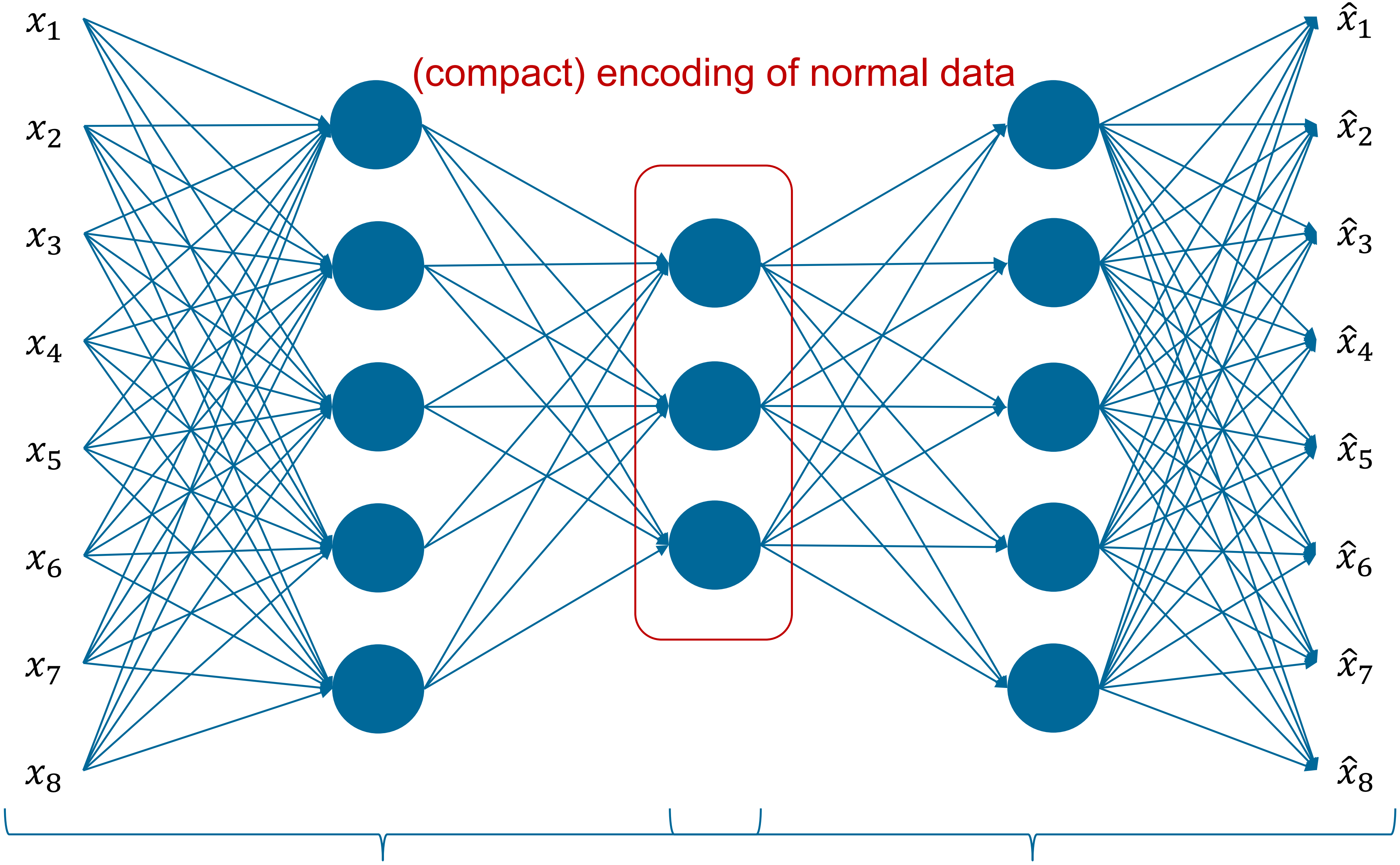
Detecting anomalies

Supervised anomaly detection:
- A fancy way of saying classification – learn to differentiate between two classes
- We can use the standard toolbox
- When feasible, usually the most failsafe method
- Only works if we know how anormal data looks like, i.e., we have enough data points

Semi-supervised anomaly detection:
- Learn an efficient representation of normal data and then try to apply this to new data coming in
- We can use autoencoders and other tools
- We don't need to know how anormal data looks like
- Still need to be sure that our normal data is actually normal

Unsupervised anomaly detection:
- Learn "how far" datapoints are from each other and recognize the ones that are far away from anything else
- We can use isolation forests and other tools
- We can work with any kind of data
- We don't have many guarantees

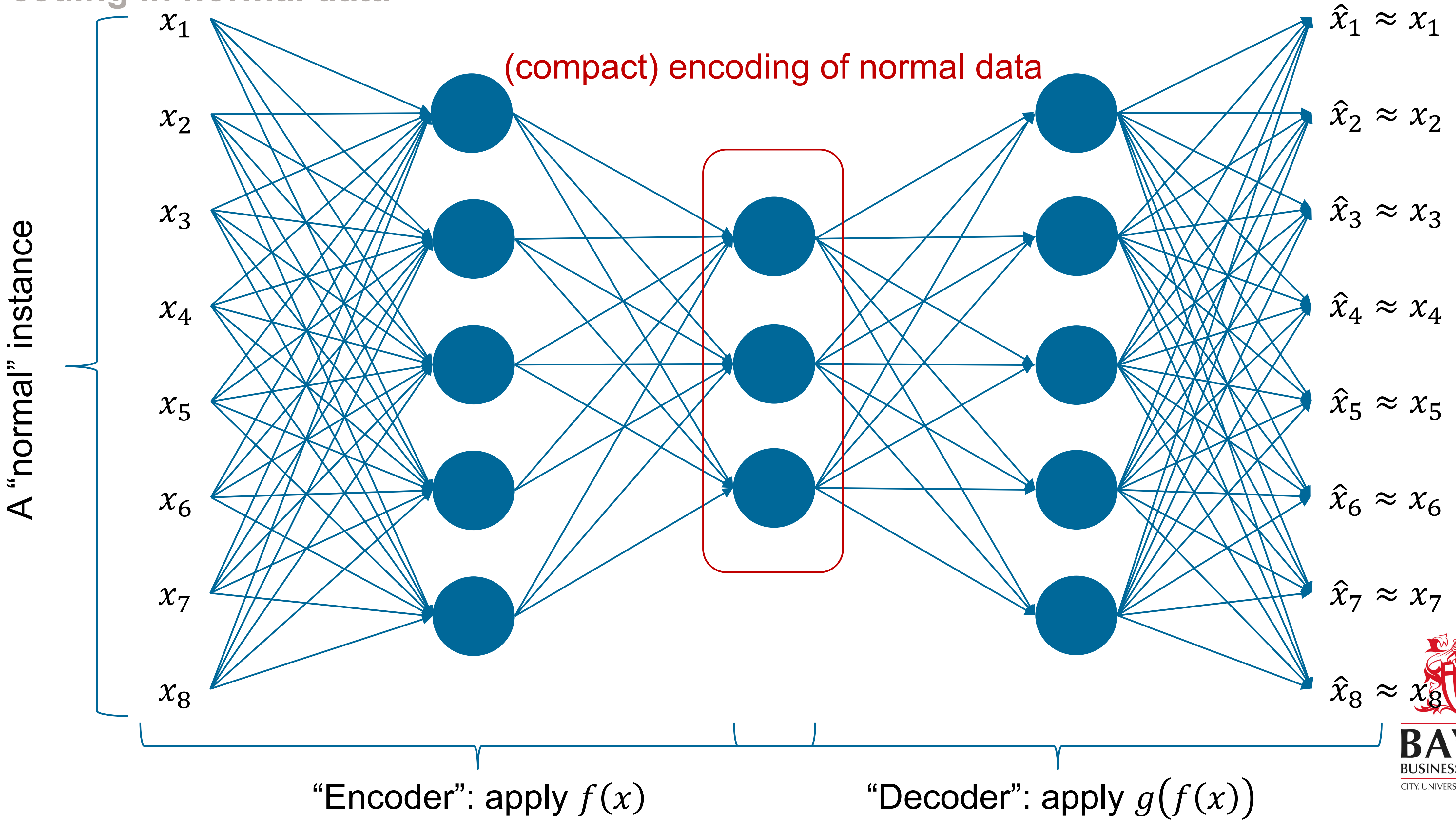
Anomaly detection with an autoencoder



"Encoder": find a representation $f(x)$ "Decoder": unpack $x = g(f(x))$

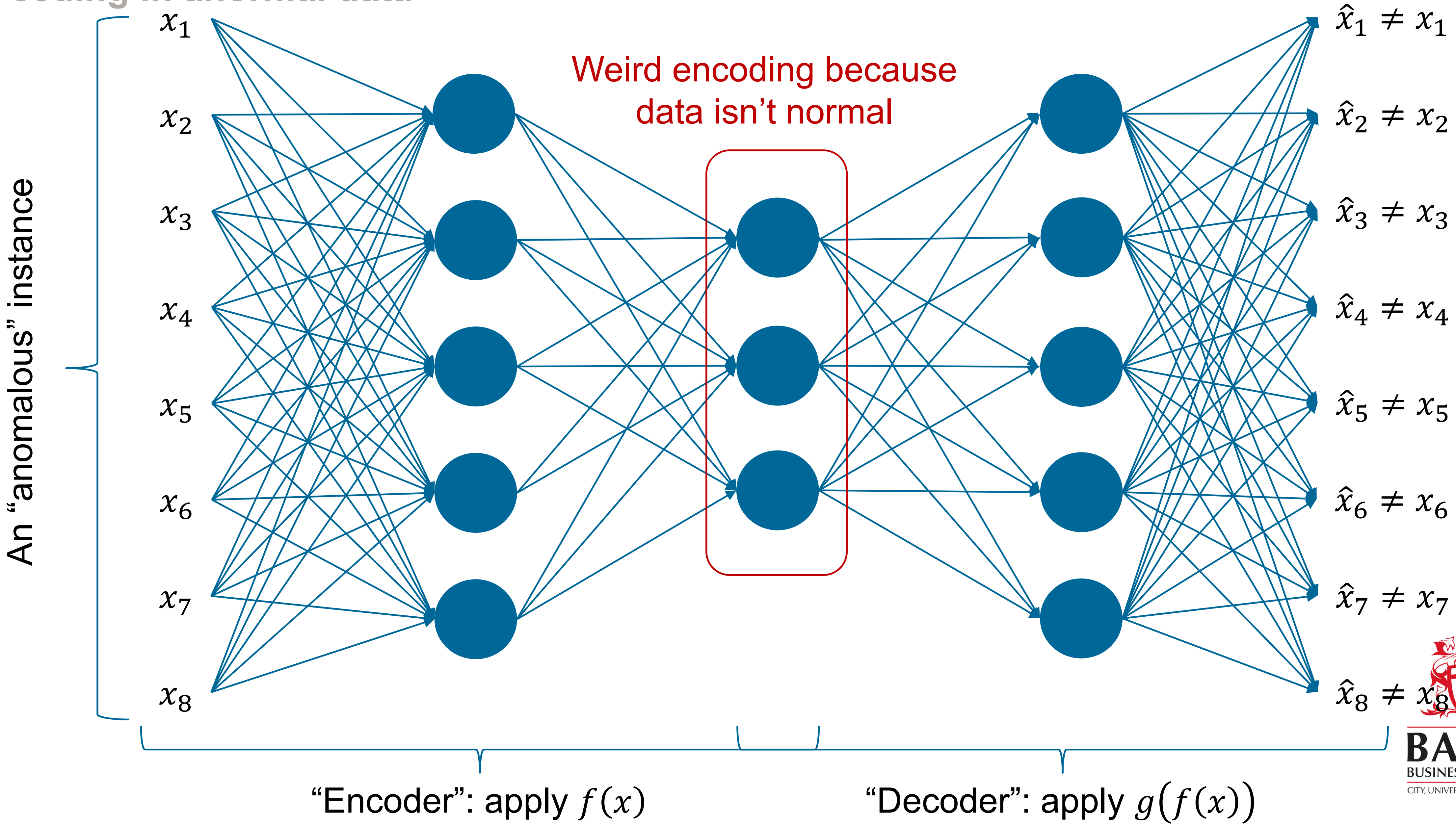
BAYES BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

Feeding in normal data



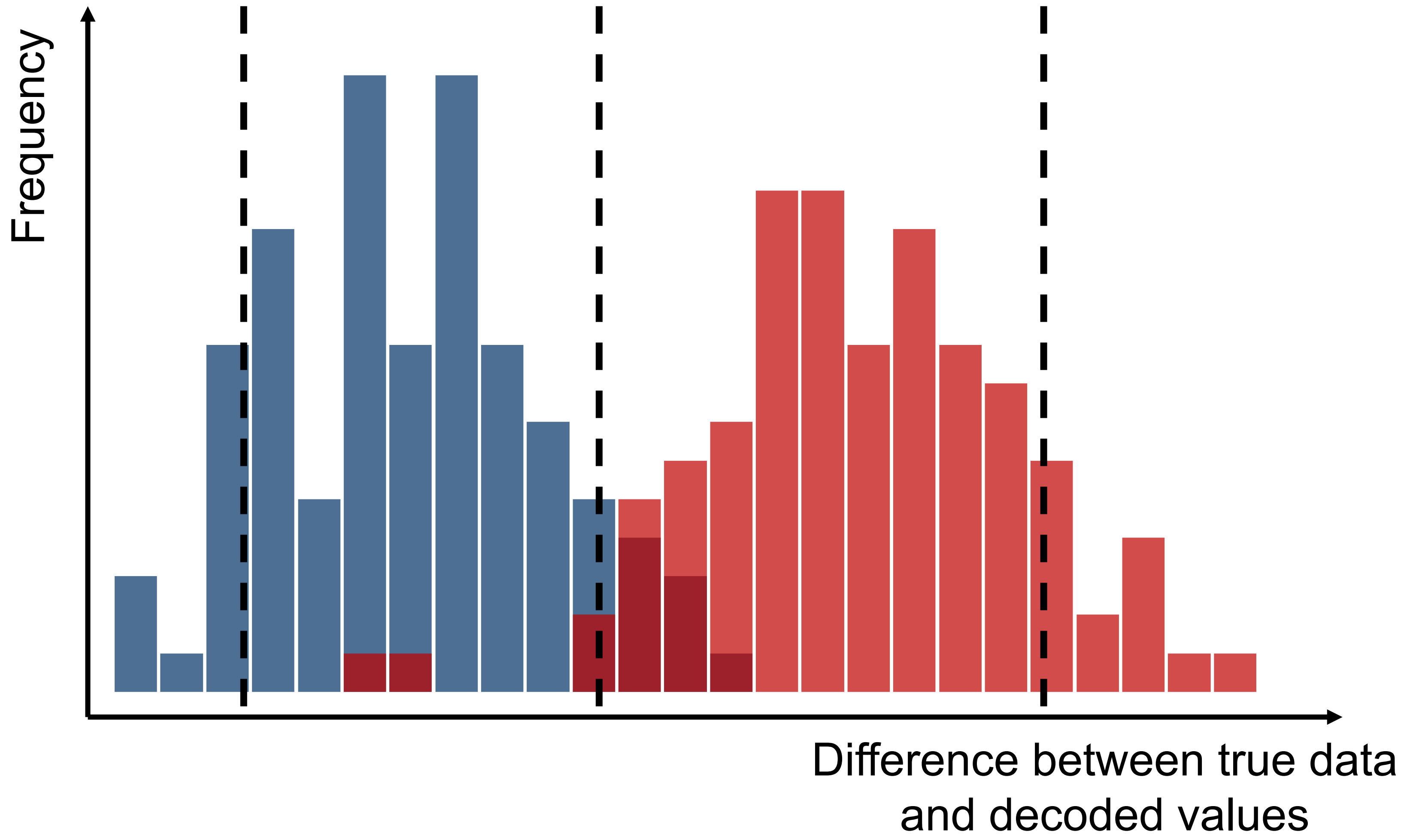
BAYES BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

Feeding in anormal data



BAYES BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

What should we be observing?



BAYES
BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

Good luck with the assignment!