Artificial Neural Networks

AUTO-ENCODERS

Topics

1. Introduction
   a) Motivation
   b) Feature Selection Problem
   c) Dimensionality Reduction
2. Auto-encoder
   a) Auto-encoders as feature selectors
   b) Auto-encoders used for data compression
   c) Using auto-encoders to reduce dimension
   d) Noise Removal
   e) Sparse auto-encoders
   f) Comparison with PCA

Introduction

Neural networks are learning machines which learn from data.

In practice the data can be large is size and redundant. Besides, some parts may be irrelevant in classification.

Hence, instead of data itself, distinguishing features of each class extracted from data are used for classification.

Features

Selecting appropriate features is a major challenge in classification.

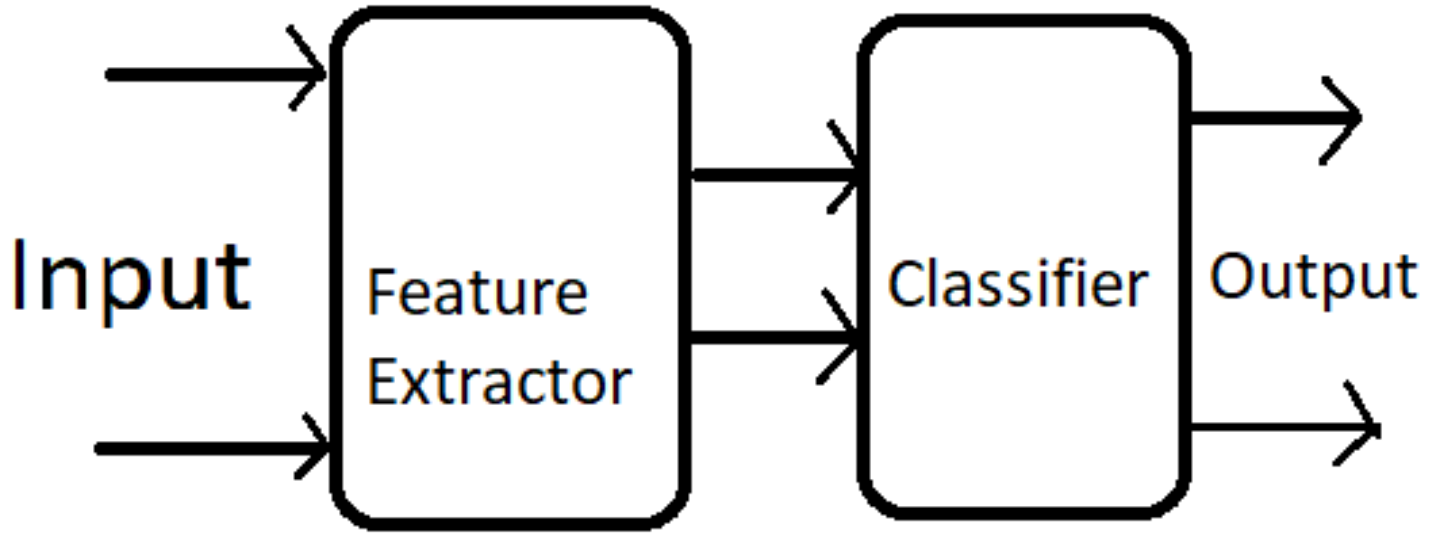
The selected features do not need to correspond to some physical property of the objects (such as color, shape, texture, etc.)

Features

The dimensionality of the feature space can reduce the efficiency of the classifier (curse of dimensionality)

Choosing a subset of the features by mapping the patterns to a lower dimensional space after transforming the features can be a solution

Multilayer Classifiers



Pre-Training for Feature Extraction

Unsupervised Pre-Training uses unsupervised learning in the deep layers to transform the inputs into features that are easier to learn by a final supervised model

Unsupervised training between layers can decompose the problem into distributed sub-problems to be further decomposed at subsequent layers

Often not a lot of labeled data available while there may be lots of unlabeled data. Unsupervised Pre-Training can take advantage of unlabeled data.

Self Training vs Unsupervised Learning

In using Unsupervised Learning as a pre-processor to supervised learning we are given examples from the same distribution

We are supposed to assume that the examples come from a set containing just examples from a pre-defined possible output classes, but the label is not available

Self Training vs Unsupervised Learning

In Self-Taught Learning we do not require that the later supervised instances come from the same distribution

- e.g., We perform self-taught learning with any images, even though later we will apply supervised learning with just cars, trains and motorcycles.
- These types of distributions are more readily available than ones which just have the classes of interest (i.e. not labeled as car *or* train *or* motorcycle)
- However, if distributions are *very* different the classification will not be accurate

Self Training vs Unsupervised Learning

New tasks share concepts/features from existing data and statistical regularities in the input distribution that many tasks can benefit from.

So we can re-use well-trained nets as starting points for other tasks

Both unsupervised and self-taught approaches are applicable by deep learning models

Auto-Encoders

A type of unsupervised learning which discovers generic features of the data

- Learn identity function by learning important sub-features
- Compression, etc. – Under-complete $|\mathbf{h}| < |\mathbf{x}|$
- For $|\mathbf{h}| \geq |\mathbf{x}|$ (Over-complete case more common in deep nets) use regularized auto-encoding: Loss function includes regularizer to make sure we don't just pass through the data (e.g. sparsity, noise robustness, etc.)

Auto-Encoders

An auto-encoder is a feed-forward neural network which consists of three layers:

- Input layer
- Hidden layer
- Output layer

The network provides the same input pattern as its output

Auto-Encoder Applications

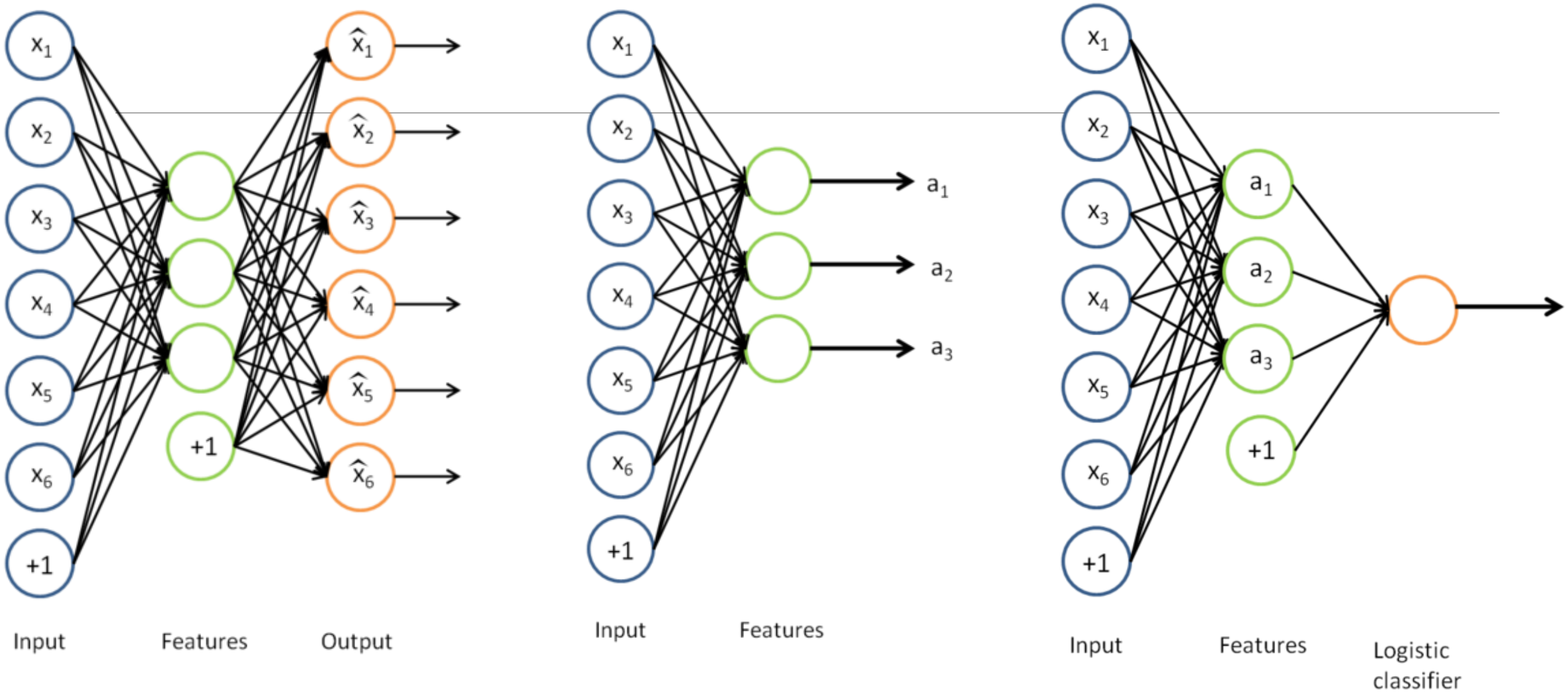
A type of unsupervised learning which discovers generic features of the data. Auto-encoders are used for:

1. Learning important sub-features
2. Data compression
3. Noise removal
4. Generating data from model

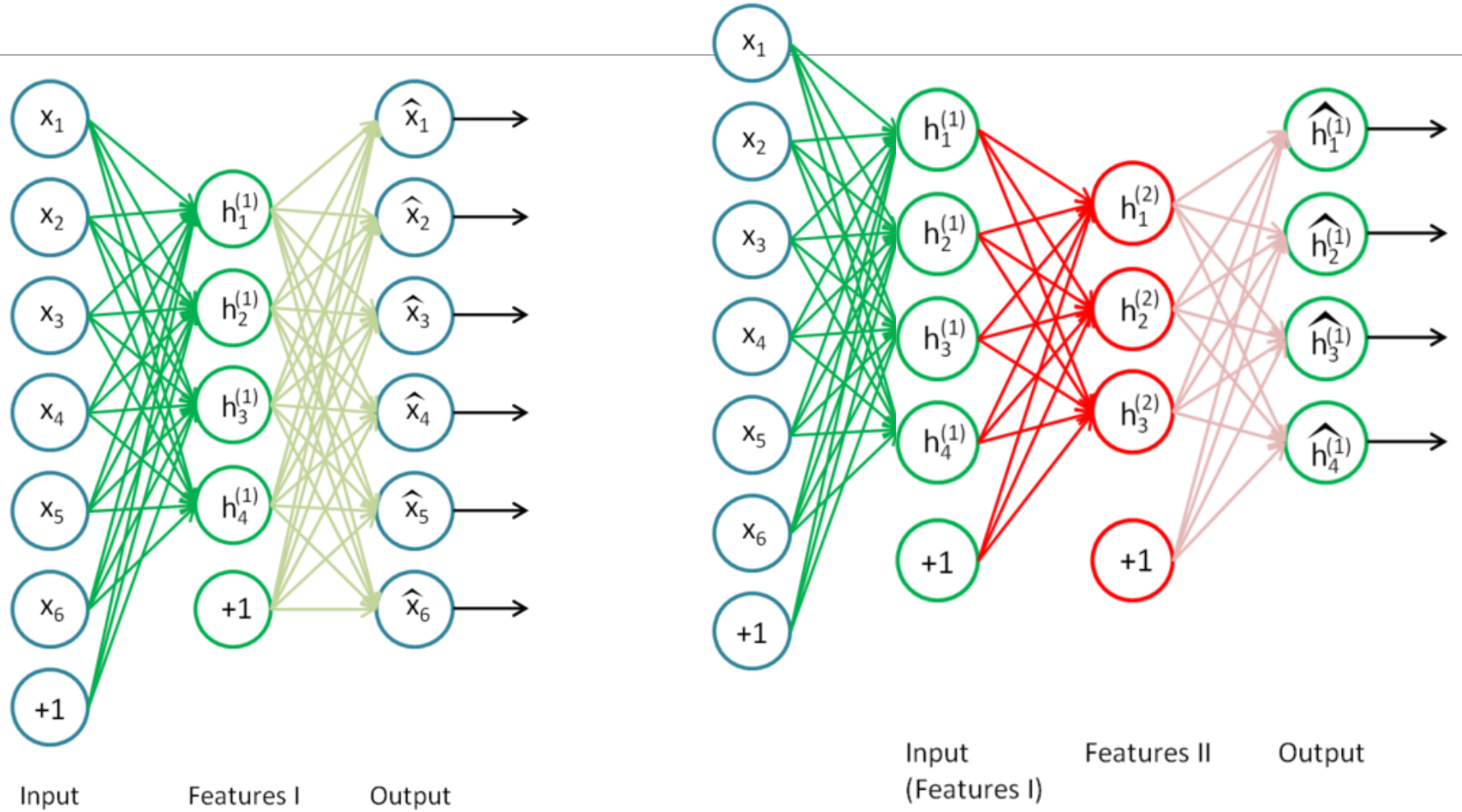
Feature Extraction

Feature extraction is the referred to the set of method for processing initial set of measured data to builds derived values (features) intended to be

1. Informative
2. non-redundant,
3. facilitating the subsequent learning and generalization steps,
4. and in some cases leading to better human understanding


x_1
x_2
x_3
x_4
x_5
x_6
+1
$\hat{x}_1$
$\hat{x}_2$
$\hat{x}_3$
$\hat{x}_4$
$\hat{x}_5$
$\hat{x}_6$
Input
Features
Output
a_1
a_2
a_3
Input
Features
Logistic classifier

Stacked Auto-Encoders

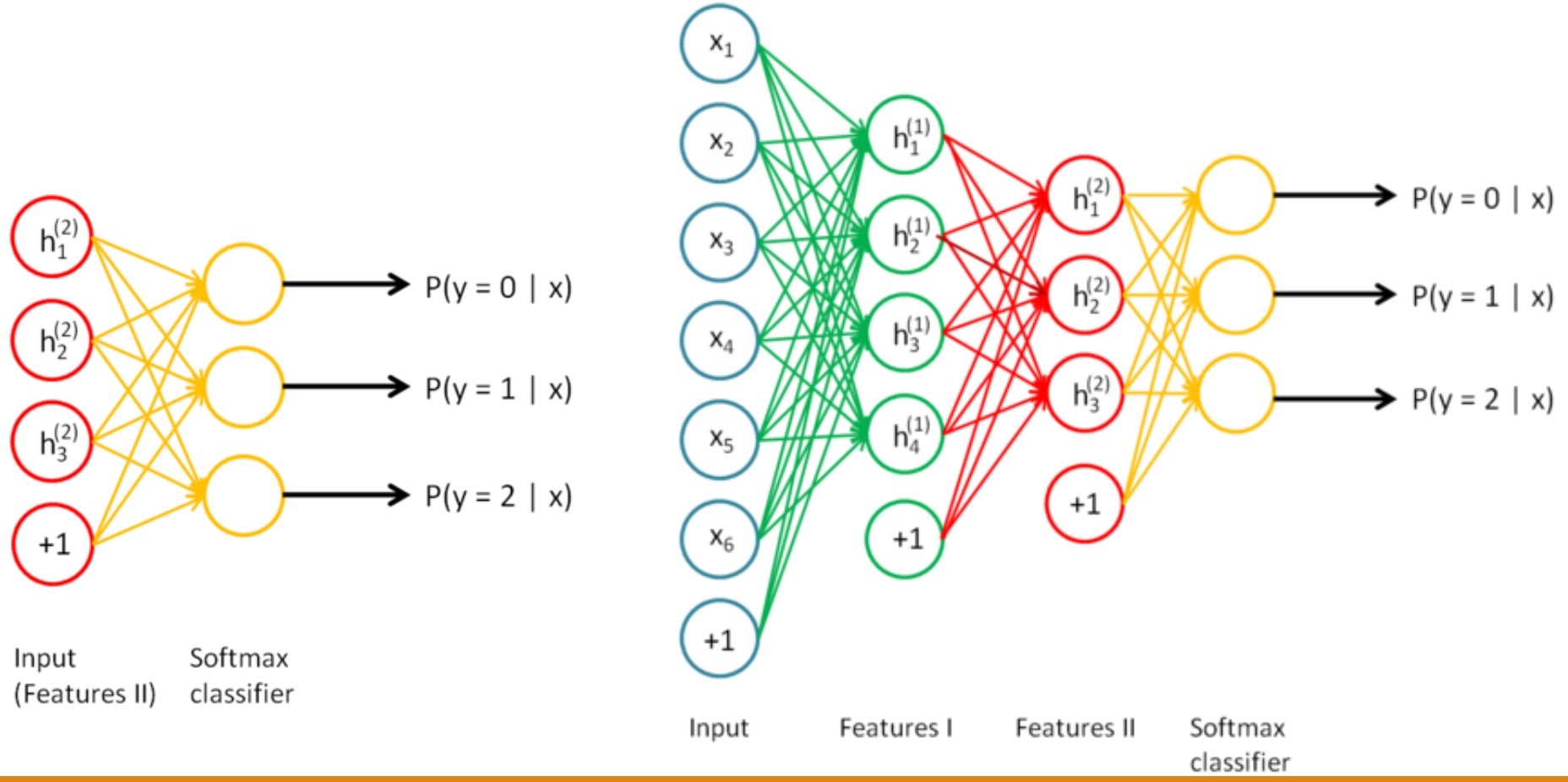


Stacked Auto-Encoders

Apply supervised training on the last layer using final features

Then perform supervised training on the entire network to fine-tune all weights

Stacked Auto-Encoders



Data Compression

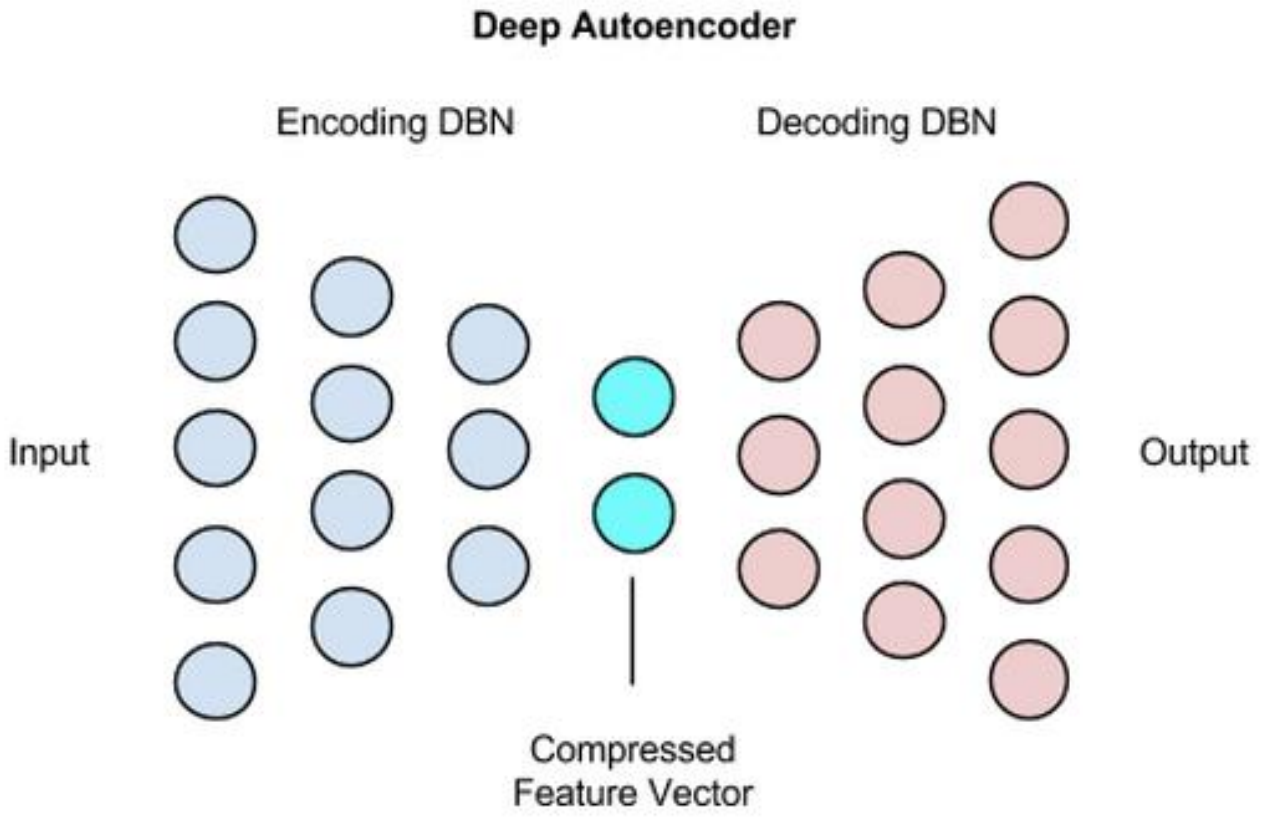
Data can be compressed if it includes redundancy

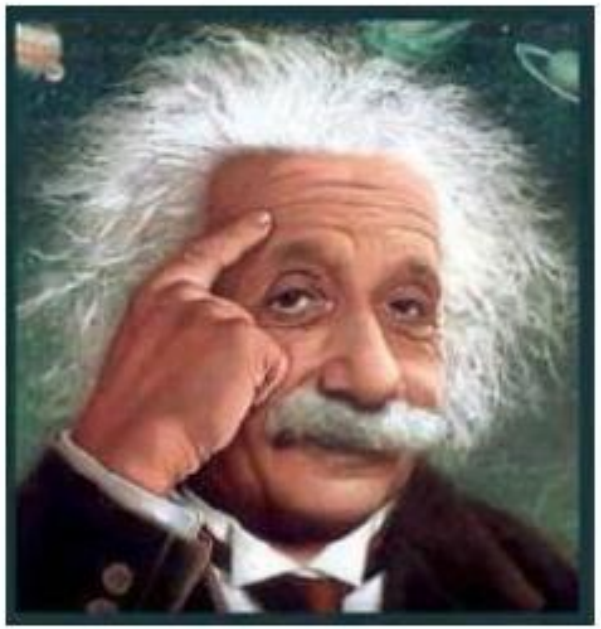
Redundancy is the number of bits used to represent or transmit data minus the actual number of information bits

Data Compression

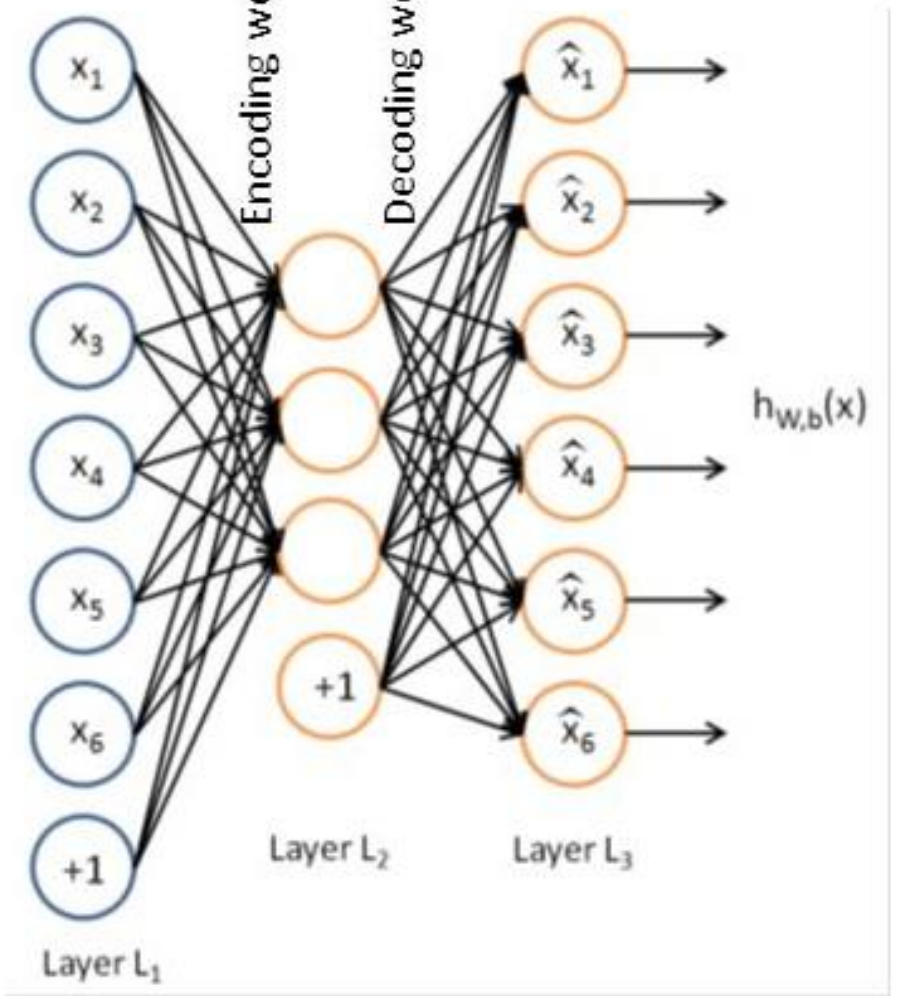
If the number of nodes in the hidden layers is smaller than the number of data elements in input then the input can be represented with a smaller number of data items

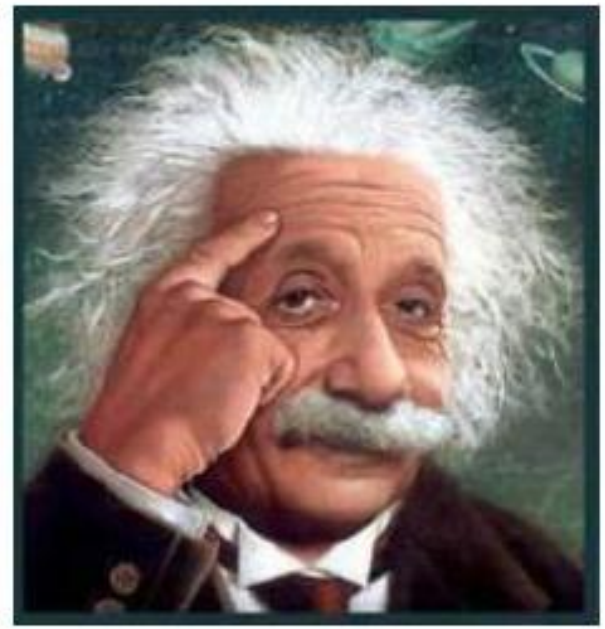
A Deep Auto-Encoder





input





target output

Noise Removal

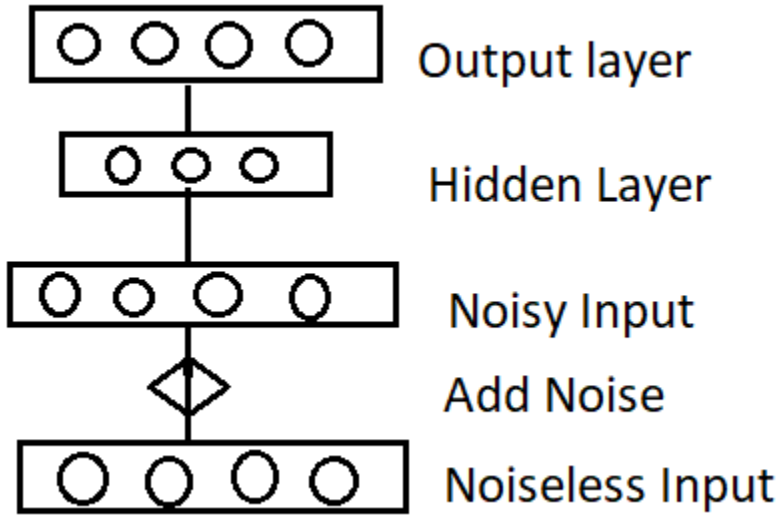
In real-life, data is generally noisy

Examples are: Image, and audio data

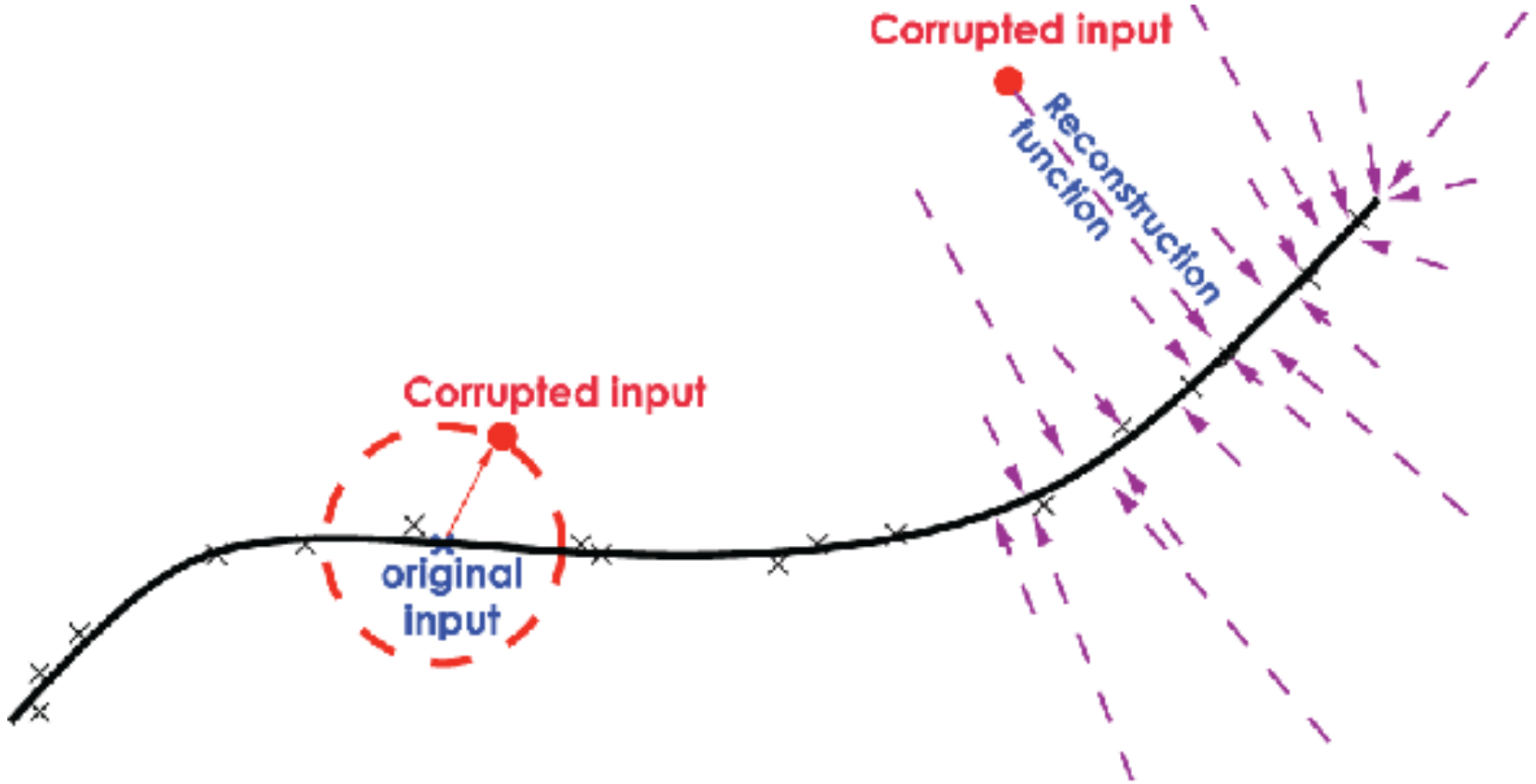
If an auto-encoder is given noisy data as input, but the loss function is defined using noiseless data, it will try to map noisy data to its noiseless counterpart

Noise Removal

The loss function is defined as the difference between the noiseless input and the output



Noise Removal



Sparse Encoders

Auto encoders will often do a dimensionality reduction

- PCA-like or non-linear dimensionality reduction

This leads to a "dense" representation

All features typically have non-zero values for any input and the combination of values contains the compressed information

However, this distributed and entangled representation can often make it more difficult for successive layers to pick out the salient features

Sparse Encoders

A *sparse* representation uses more features where at any given time many/most of the features will have a 0 value

- Thus there is an implicit compression each time but with varying nodes
- This leads to more local variable length encodings where a particular node (or small group of nodes) with value 1 signifies the presence of a high-order feature
- This is easier for subsequent layers to use for learning

How to make a sparse Auto-Encoder?

Use more hidden nodes in the encoder

Use regularization techniques which encourage sparseness (e.g. a significant portion of nodes have 0 output for any given input)

- Penalty in the learning function for non-zero nodes
- Weight decay
- etc.

Comparison with PCA

Principal Component Analysis (PCA) is used to lower the dimensionality of the feature space.

Auto-encoders with a small number of hidden layer nodes can perform the same task.

The experimental studies show that auto-encoders outperform PCA in many cases

Overview - Recap

Principal Component Analysis (PCA) is a way to reduce data dimensionality

PCA projects high dimensional data to a lower dimension

PCA projects the data in the least square sense– it captures big (principal) variability in the data and ignores small variability

Principal Components

Principal components are give by the eigenvectors of the covariance matrix

Main principal component is given by the eigenvector corresponding to the largest eigenvalue.

Covaiance Matric: More than two attributes

$$C^{nxn} = (c_{ij} \mid c_{ij} = \text{cov}(Dim_i, Dim_j))$$

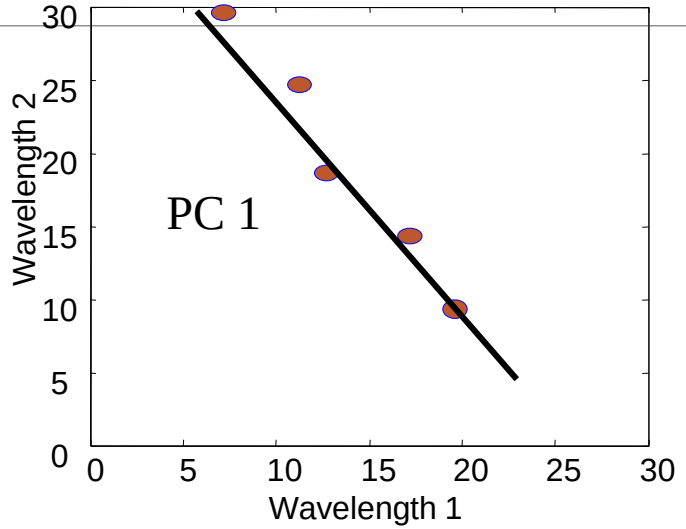
Example for three attributes (x,y,z):

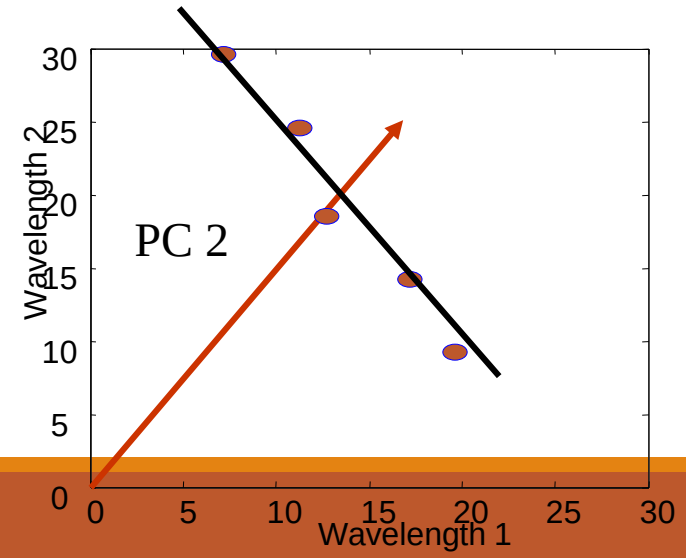
$$C = \begin{pmatrix} \text{cov}(x,x) & \text{cov}(x,y) & \text{cov}(x,z) \\ \text{cov}(y,x) & \text{cov}(y,y) & \text{cov}(y,z) \\ \text{cov}(z,x) & \text{cov}(z,y) & \text{cov}(z,z) \end{pmatrix}$$

Principal Components

First PC is direction of maximum variance from origin

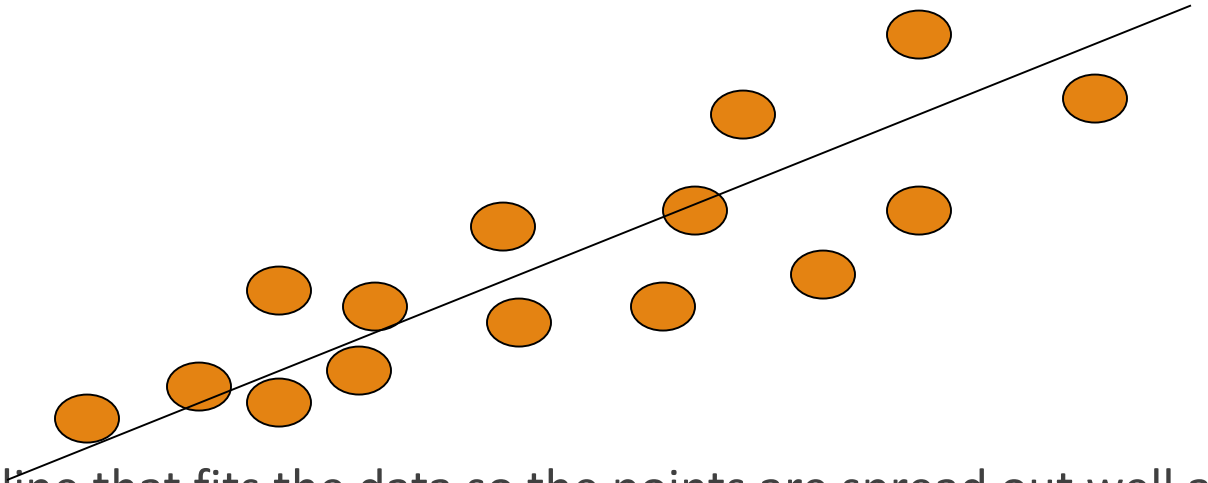
Subsequent PCs are orthogonal to 1st PC and describe maximum residual variance





Algebraic Interpretation – 1D

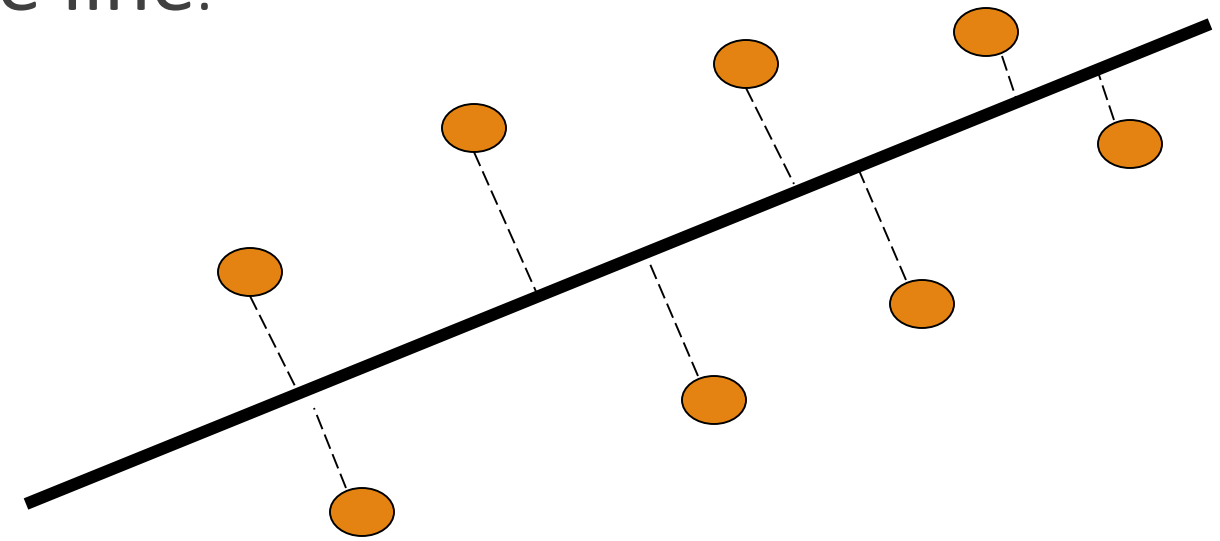
Given m points in a n dimensional space, what is the best line to represent this data?

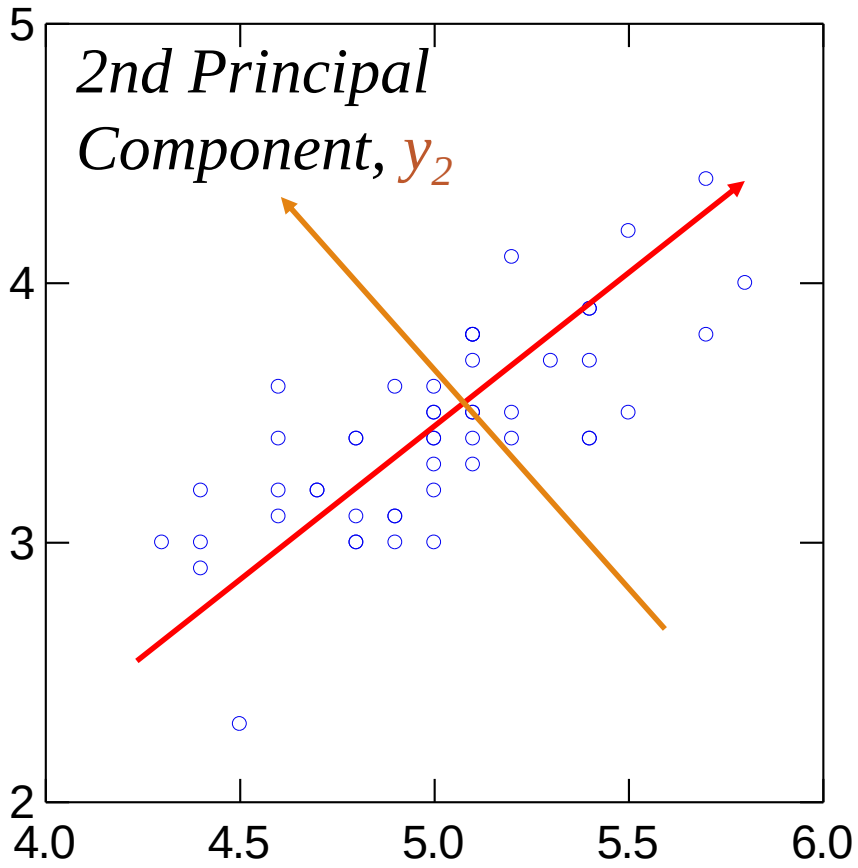


Choose a line that fits the data so the points are spread out well along the line

Algebraic Interpretation – 1D

Formally, minimize sum of squares of distances to the line.

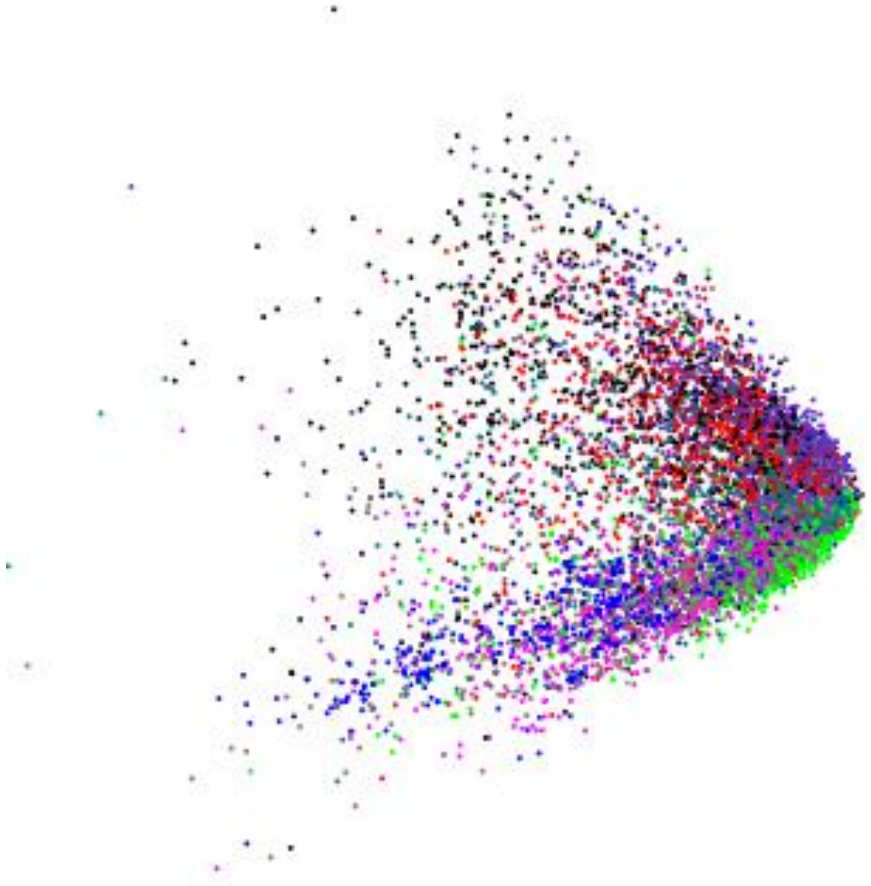



2nd Principal Component, y_2
1st Principal Component, y_1
5
4
3
2
4.0
4.5
5.0
5.5
6.0

Sample Result

The result of reducing dimensionality on a text document categorization application.

PCA (LSA)



Auto-Encoder