ASTR 3890 - Selected Topics: Data Science for Large Astronomical Surveys (Spring 2022)

Clustering & Unsupervised Classification

Dr. Nina Hernitschek
April 18, 2022

info: Take Home Exam

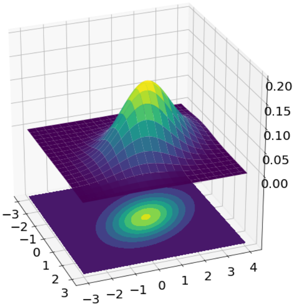
The take home exam will be available on `github` on Monday 25 at 4pm.

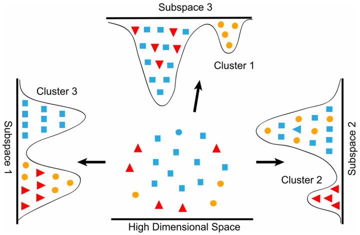
It will consist of multiple problems.

You have 1 week to solve the problems and to submit on `github` - i.e.: The exam is due Monday, May 2 at 4pm.

To avoid any issues with github, I recommend to frequently upload the code to github, e.g. after solving each problem.

recap: Dimensionality Reduction & Clustering





Clustering

Clustering algorithms attempt to **group together similar objects** in a data set.

Clustering

Clustering algorithms attempt to **group together similar objects** in a data set.

This process allows us to

- put new objects into the resulting classes
- identify rare objects that do not fit any particular scheme.

Clustering is inherently an **unsupervised** as we do not know the classification of the objects.

Clustering

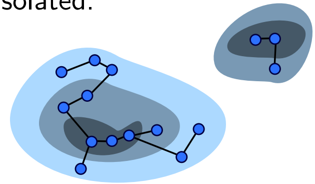
seen in lecture 10:

Percolation or 'Friends of Friends (FoF)' algorithm

1. Plot data points in a 2-dimensional diagram (or: calculate distances using a metric).

2. Find the closest pair, and call the merged object a *cluster*.

3. Repeat step 2 until some chosen threshold is reached. Some objects will lie in rich clusters, others have one companion, and others are isolated.



K-Means Clustering

K-Means Clustering seeks to minimize

$$\sum_{k=1}^{K} \sum_{i \in C_k} ||x_i - \mu_k||^2$$

where $\mu_k = \frac{1}{N_k} \sum_{i \in C_k} x_i$.

K-Means Clustering

K-Means Clustering seeks to minimize

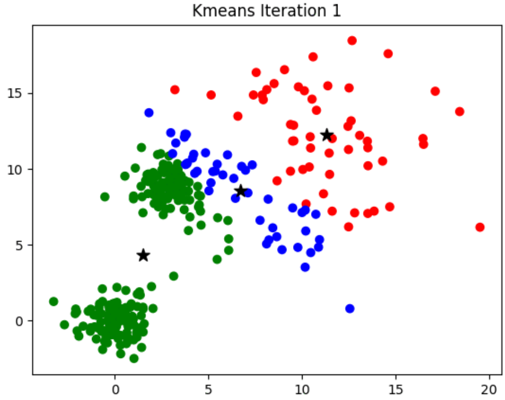
$$\sum_{k=1}^{K} \sum_{i \in C_k} ||x_i - \mu_k||^2$$

where $\mu_k = \frac{1}{N_k} \sum_{i \in C_k} x_i$.

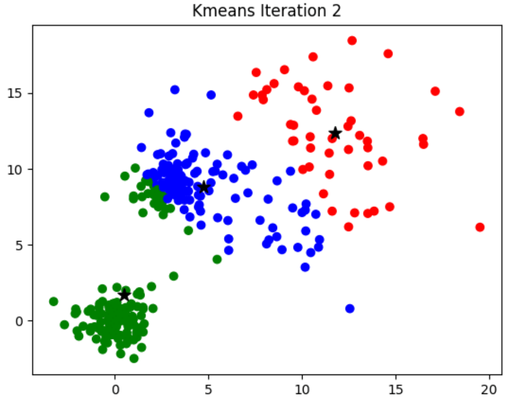
The steps are:

1. For a given value of K, initialize K number of centroids randomly and the data points are partitioned into K clusters
2. compute the distance between each of the input to the K centroids and re-assign it to the cluster with the least distance
3. after the re-assignment, update the centroid of each cluster by calculating the mean of the data points in the cluster
4. repeat steps 2 - 3 until there is no re-assignment required

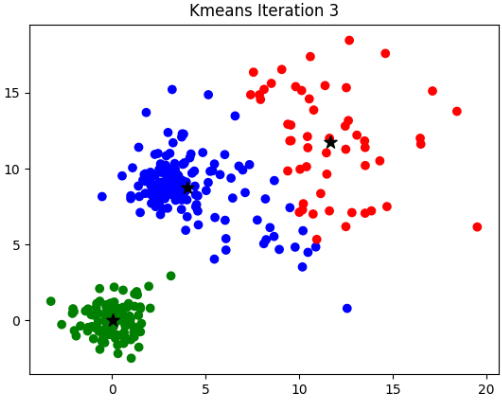
K-Means Clustering



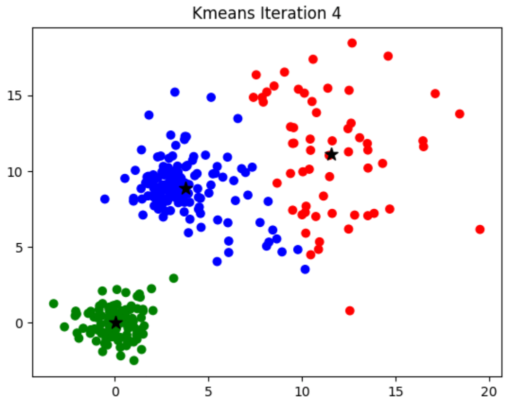
K-Means Clustering



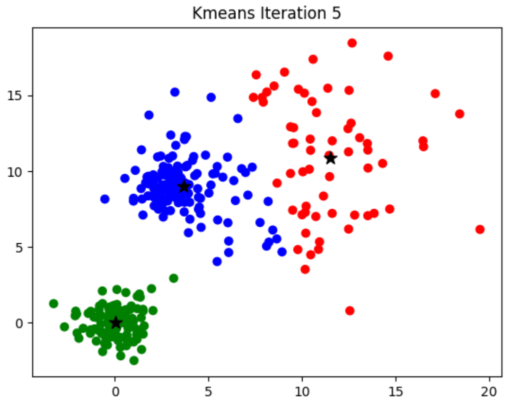
K-Means Clustering



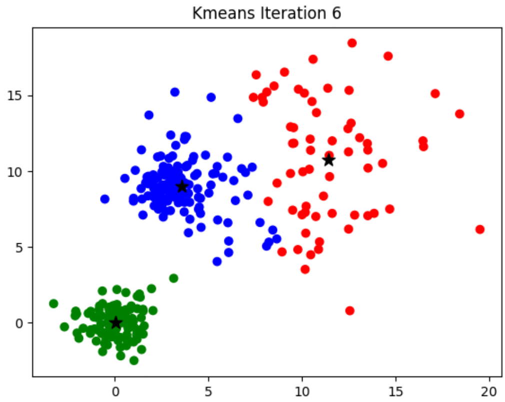
K-Means Clustering



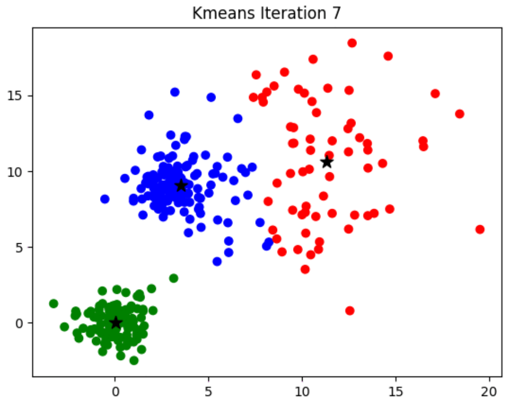
K-Means Clustering



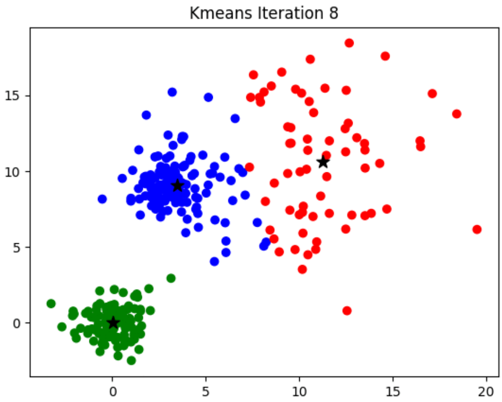
K-Means Clustering



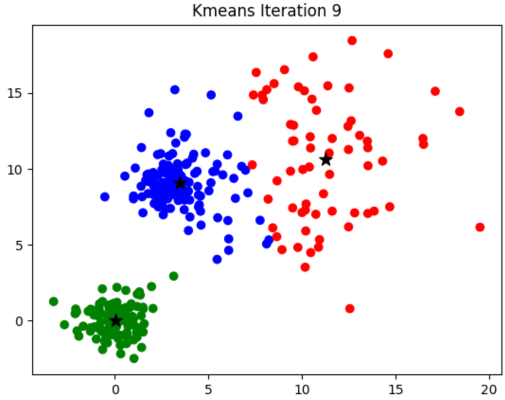
K-Means Clustering



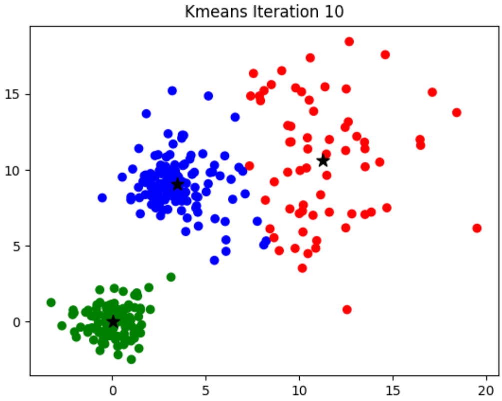
K-Means Clustering



K-Means Clustering



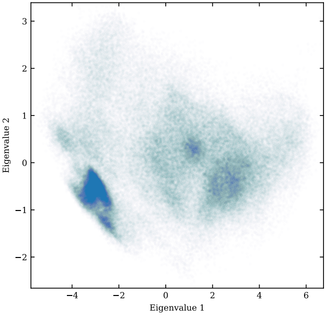
K-Means Clustering



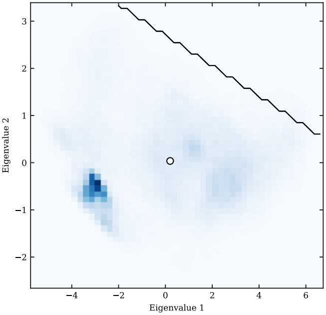
K-Means Clustering

multidimensional sample, first applying PCA and then a K-Means Clustering

sample after PCA



sample after PCA
and K-Means Clustering



Mean-shift Clustering

Mean-shift clustering works by finding the modes in a kernel density estimator (KDE) of the distribution. Clustering is achieved in the following way:

1. The KDE of the dataset is computed.
2. This allows the gradient of the distribution to be calculated. Easy to do since it's a bunch of overlapping Gaussians.
3. Each data point is shifted in the direction of increasing gradient, which drives the points toward the modes.
4. This process is iterated until all points have converged with clusters of other points at each of several distinct modes.
5. Each data point is then associated with a cluster of other points.

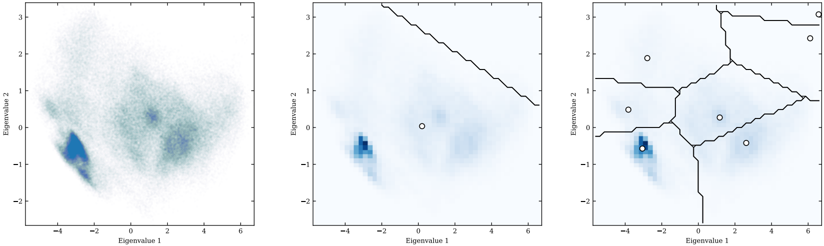
Mean-shift Clustering

multidimensional sample, first applying PCA and then a K-Means or Mean-Shift Clustering

sample after PCA

sample after PCA and K-Means Clustering

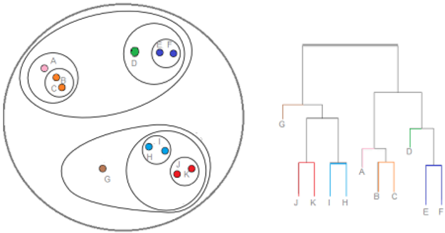
sample after PCA and Mean-Shift Clustering



Hierarchical clustering

Hierarchical clustering is a family of clustering algorithms that build **nested clusters**. This hierarchy of clusters is represented as a **tree (or dendrogram)**. The root of the tree is the unique cluster that gathers all the samples, the leaves being the clusters with only one sample.

The number of clusters aren't specified ahead of time. Instead, hierarchical clustering starts with clusters representing each data point and then the most similar clusters are joined iteratively.



Unsupervised Classification

methods like clustering, especially the more advanced, fall into the category of unsupervised classification

but generally, **Unsupervised Classification** means the more advanced techniques used to find substructure in data

Novelty and Outlier Detection

Many applications require being able to decide whether a new observation belongs to the same distribution as existing observations (it is an inlier), or should be considered as different (it is an outlier). Often, this ability is used to clean real data sets.

Novelty and Outlier Detection

Many applications require being able to decide whether a new observation belongs to the same distribution as existing observations (it is an inlier), or should be considered as different (it is an outlier). Often, this ability is used to clean real data sets.

outlier detection:
The training data contains outliers which are defined as observations that are far from the others. Outlier detection estimators thus try to fit the regions where the training data is the most concentrated, ignoring the deviant observations.

Novelty and Outlier Detection

Many applications require being able to decide whether a new observation belongs to the same distribution as existing observations (it is an inlier), or should be considered as different (it is an outlier). Often, this ability is used to clean real data sets.

outlier detection:
The training data contains outliers which are defined as observations that are far from the others. Outlier detection estimators thus try to fit the regions where the training data is the most concentrated, ignoring the deviant observations.

novelty detection:
The training data is not polluted by outliers and we are interested in detecting whether a new observation is an outlier. In this context an outlier is also called a novelty.

Novelty and Outlier Detection

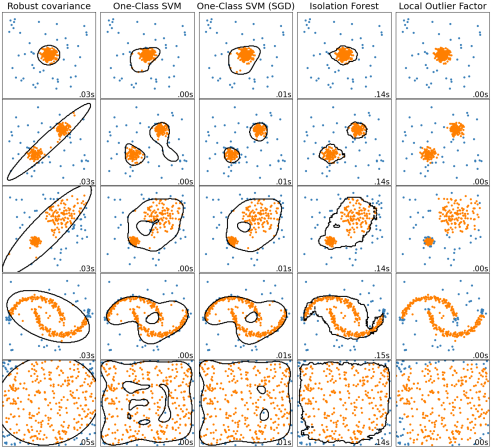
other terms used:

Outlier detection and novelty detection are both used for **anomaly detection**, where one is interested in detecting abnormal or unusual observations. In the context of outlier detection, the outliers/anomalies cannot form a dense cluster as available estimators assume that the outliers/anomalies are located in low density regions. On the contrary, in the context of novelty detection, novelties/anomalies can form a dense cluster as long as they are in a low density region of the training data, considered as normal in this context.

The `scikit-learn` library provides a set of unsupervised machine learning tools that can be used both for novelty or outlier detection.

Outlier Detection

A comparison of the outlier detection algorithms in scikit-learn.

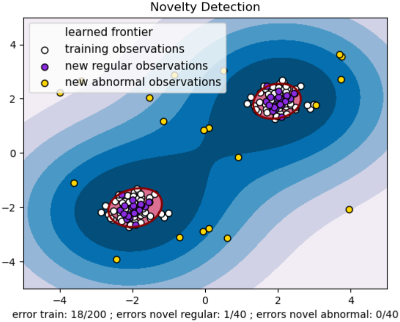


Local Outlier Factor (LOF) has no decision boundary (black) as it has no predict method to be applied on new data in outlier detection.

Novelty Detection

Consider a data set of n observations from the same distribution described by p features. Consider now that we **add one more observation** to that data set.

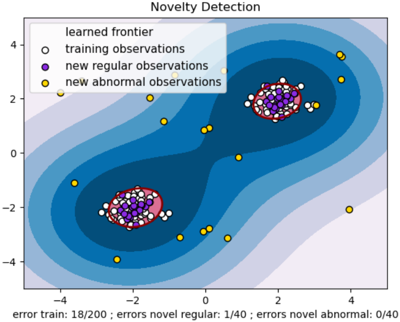
Is the new observation so different from the others that we can doubt it is regular? (i.e. does it come from the same distribution?) Or on the contrary, is it so similar to the other that we cannot distinguish it from the original observations?



Novelty Detection

Novelty detection works by learning a close frontier delimiting the contour of the initial observations distribution in p-dimensional space. New observations are considered as coming from the same population than the initial observations if they lay within that contour. Otherwise they are considered as novelty.

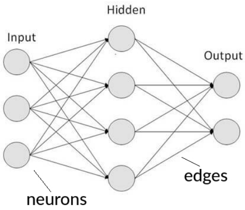
The **One-Class SVM** is implemented in the Support Vector Machines module in the scikit-learn `svm.OneClassSVM` object.



Neural network models

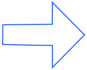
Artificial neural networks (ANNs), usually simply called neural networks (NNs), are information processing paradigms inspired by biological neural networks.

artificial neural network



applications:

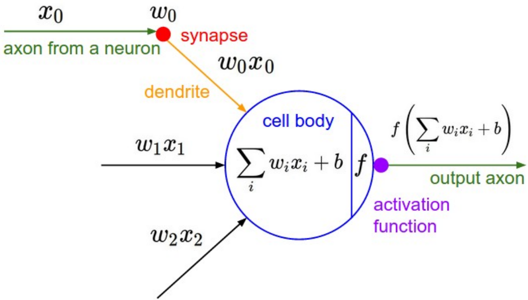
- pattern recognition
- predictive modeling
- robotics



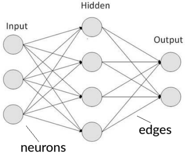
training, self-learning via experience to derive conclusions from complex data sets

Components of Neural Networks

neurons:

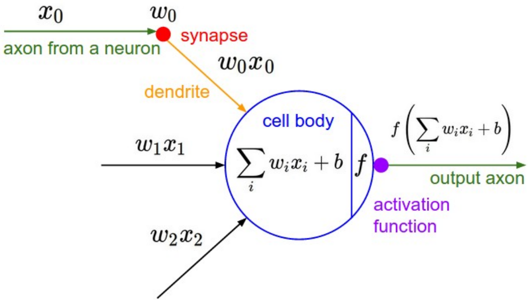


artificial neural network

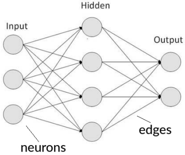


Components of Neural Networks

neurons:



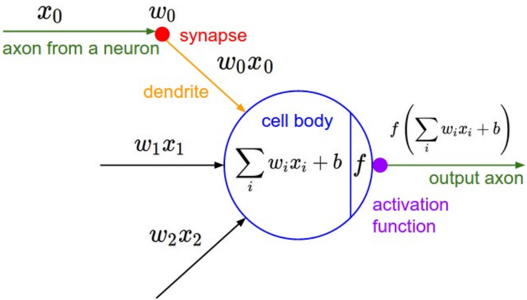
artificial neural network



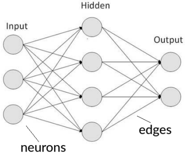
Each artificial neuron has 1 to *n* inputs and produces a single output which can be sent 1 to *m* other neurons.

Components of Neural Networks

neurons:



artificial neural network

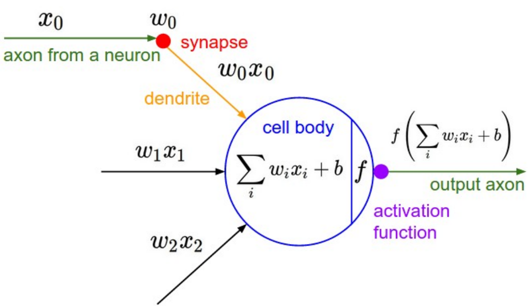


Neurons compute their **output as weighted sums** of their inputs x_i with weights w_i and a bias b:

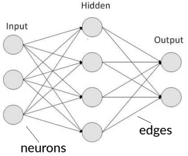
$$\sum_i w_i x_i \; + b$$

Components of Neural Networks

neurons:

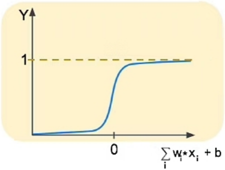


artificial neural network



Activation function f controls the amplitude of the output to be within $[0, 1]$:

$$f\left(\sum_i w_i x_i \;+ b\right)$$

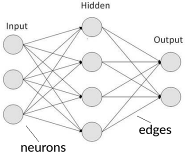


Components of Neural Networks

connections (*edges*) and layers:

Neurons are typically organized into multiple **layers**.

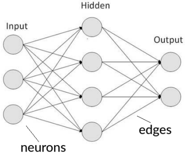
artificial neural network



Components of Neural Networks

connections (*edges*) and layers:

artificial neural network



Neurons are typically organized into multiple **layers**.

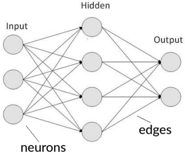
Neurons of one layer connect via **edges** only to neurons of the immediately preceding and immediately following layers.

The layer that receives external data is the **input layer**.

Components of Neural Networks

connections (*edges*) and layers:

artificial neural network



Neurons are typically organized into multiple **layers**.

Neurons of one layer connect via **edges** only to neurons of the immediately preceding and immediately following layers.

The layer that receives external data is the **input layer**.

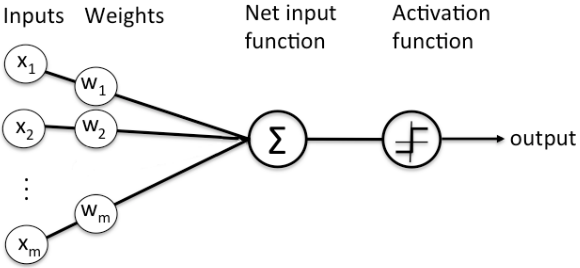
Neuron **inputs** can be external data or outputs of other neurons.

The outputs of the **final output neurons** of the neural net accomplish the task, such as recognizing an object in an image.

Perceptron

The simplest form:

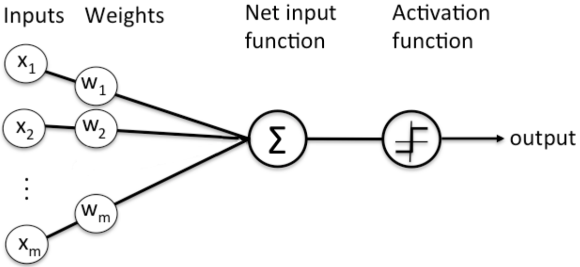
The perceptron is the oldest neural network, created by the Cornell University psychologist Frank Rosenblatt in 1957.



Perceptron

The simplest form:

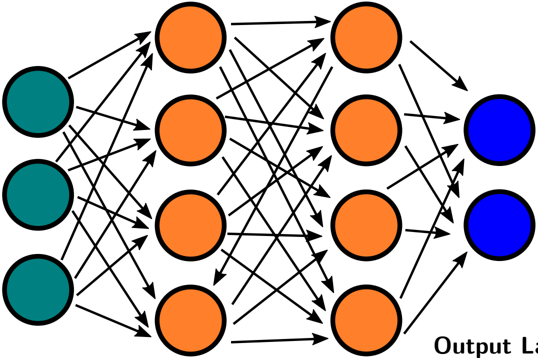
The perceptron is the oldest neural network, created by the Cornell University psychologist Frank Rosenblatt in 1957.



- a single neuron
- adjustable (trainable) weights
- an activation function
- input and output layer

Feedforward Neural Networks

A more complex form:



Input Layer
picks up signals and passes them to the next layer

Hidden Layers
calculations

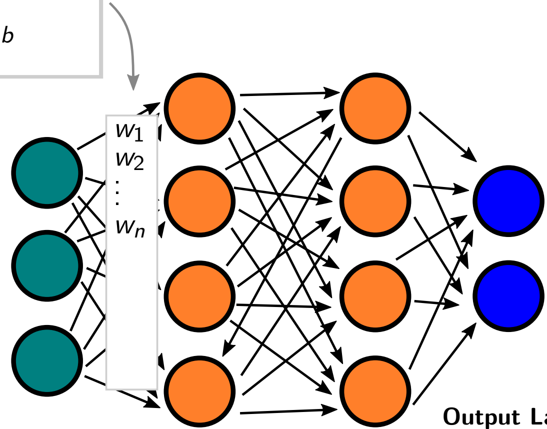
Output Layer
delivers the final result

Feedforward Neural Networks

A more complex form:

calculate weighted sum and add bias:

$$\sum_i w_i x_i + b$$



Input Layer
picks up signals and passes them to the next layer

Hidden Layers
calculations

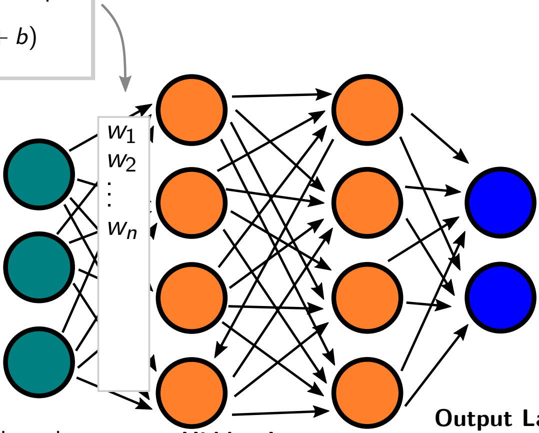
Output Layer
delivers the final result

Feedforward Neural Networks

A more complex form:

activation function f controls output amp.:

$$f(\sum_i w_i x_i \; + b)$$



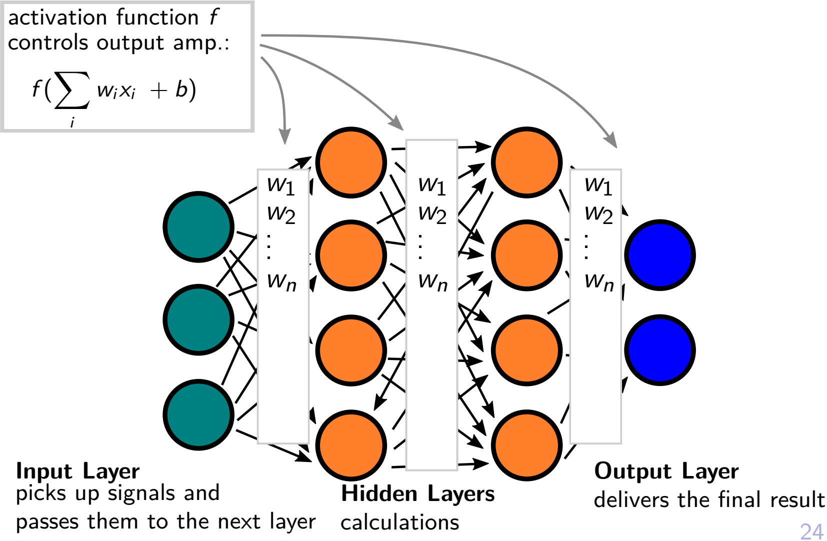
Input Layer
picks up signals and passes them to the next layer

Hidden Layers
calculations

Output Layer
delivers the final result

Feedforward Neural Networks

A more complex form:



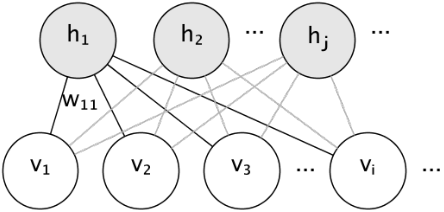
Unsupervised Neural Networks

Neural networks can be either supervised or unsupervised.

Restricted Boltzmann machines (RBM) are unsupervised nonlinear feature learners based on a probabilistic model. The features extracted by an RBM often give good results when fed into a linear classifier such as a linear SVM or a perceptron.

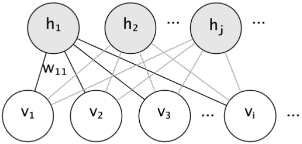
`scikit-learn` provides `BernoulliRBM`.

The graphical model of an RBM is a fully-connected bipartite graph.



Restricted Boltzmann machines

The graphical model of an RBM is a fully-connected bipartite graph.



The model is parameterized by the weights of the connections, as well as one intercept (bias) term for each visible and hidden unit.
The energy function measures the quality of a joint assignment:

$$E(\mathbf{v}, \mathbf{v}) = -\sum_j \sum_j w_{ij} v_i h_j - \sum_i b_i v_i - \sum_j c_j h_j$$

In this equation, **b** and **c** are the intercept vectors for the visible and hidden layers. The joint probability of the model is defined in terms of the energy:

$$P(\mathbf{v}, \mathbf{v}) = \frac{e^{-E(\mathbf{v},\mathbf{v})}}{Z}$$

The word restricted refers to the bipartite structure of the model, which prohibits direct interaction between hidden units, or between visible units.

Break & Questions

afterwards we continue with `lecture_12.ipynb` from the `github` repository