浙江大学爱丁堡大学联合学院
ZJU-UoE Institute

Lecture 9 - Machine learning in image analysis

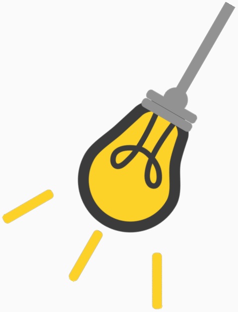
Nicola Romanò - nicola.romano@ed.ac.uk

Plan for the next lectures

- Lectures 9-11 - Traditional ML approaches in image analysis
- Lectures 12-14 - Convolutional neural networks (CNN)
- Lectures 15-17 - Practical aspects of using CNNs.

Learning objectives

- Describe use cases for machine learning in image analysis
- Explain the difference between supervised and unsupervised allgorithms
- Discuss the bias-variance tradeoff and methods to reduce overfitting

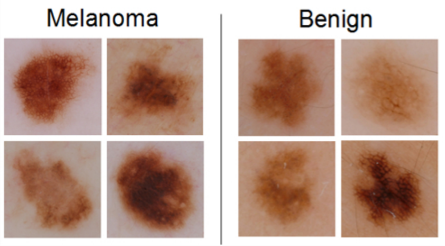


Introduction

How can machine learning help?

Some example tasks that can be solved through ML

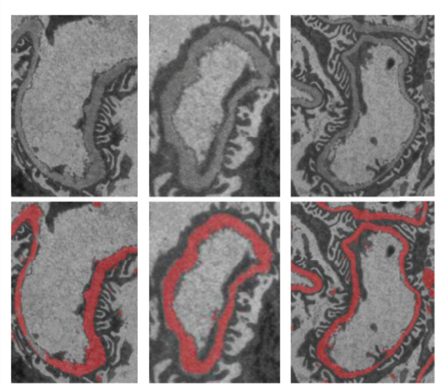
- Classification of images



ISIC melanoma classification competition. Many different solutions, including neural networks, support vector machines, deep learning...

Some example tasks that can be solved through ML

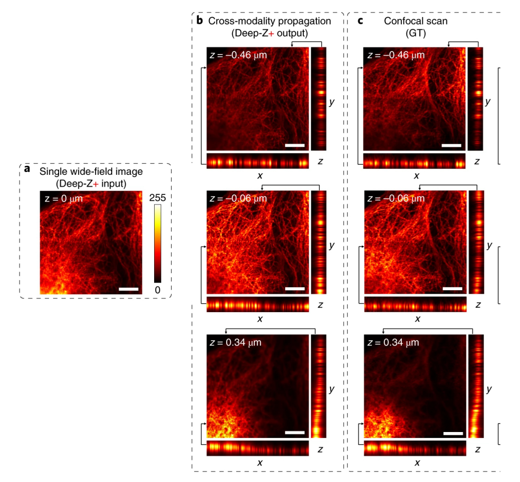
- Classification of images
- Classification of pixels (segmentation)



Cao et al. 2019, Classification of glomerular basament membrane using Random Forests.

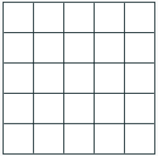
Some example tasks that can be solved through ML

- Classification of images
- Classification of pixels (segmentation)
- "Prediction" of images

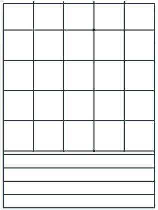


Wu et al., 2019 - Three-dimensional virtual refocusing of fluorescence microscopy images using deep learning

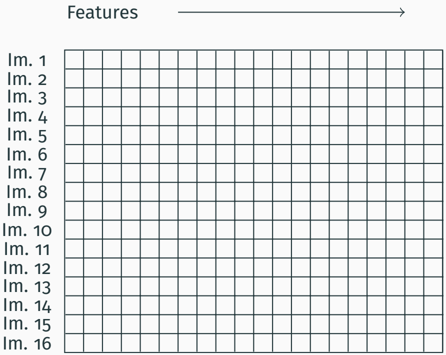
The general process



Image



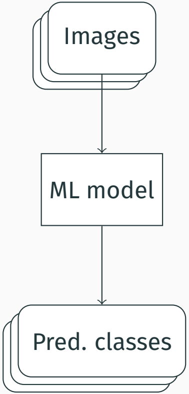
Features
(Intensity, Edges, texture, ...)



Unwrap
(and feed to model!)

Supervised vs unsupervised ML

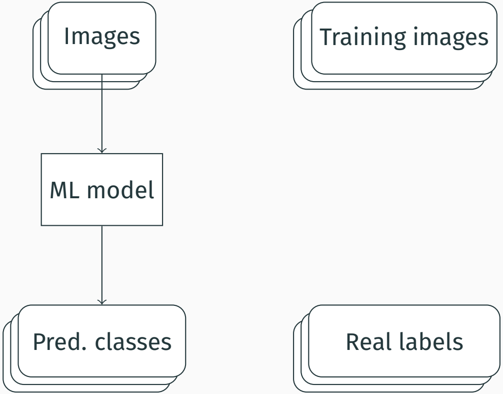
Unsupervised



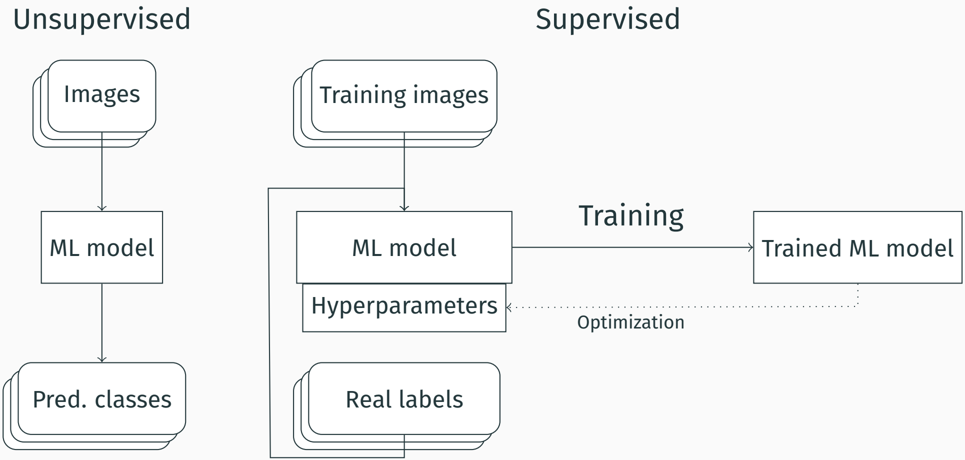
Supervised vs unsupervised ML

Unsupervised

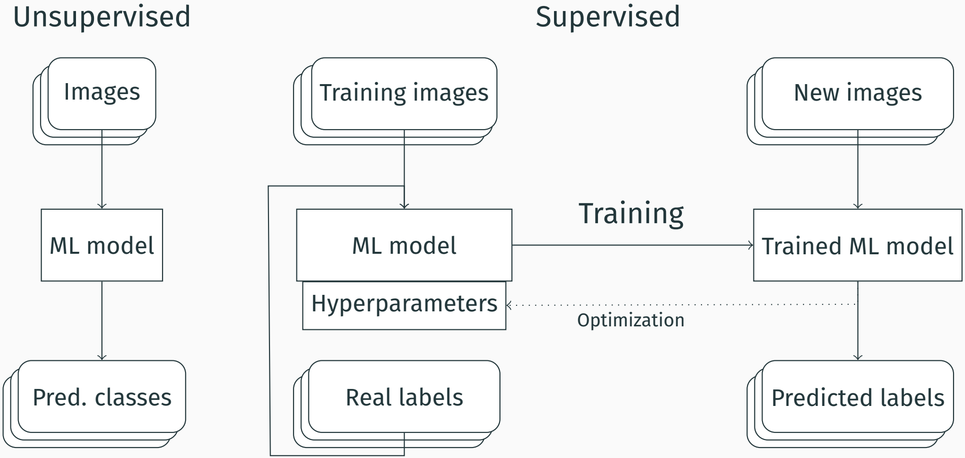
Supervised



Supervised vs unsupervised ML



Supervised vs unsupervised ML



Unsupervised methods

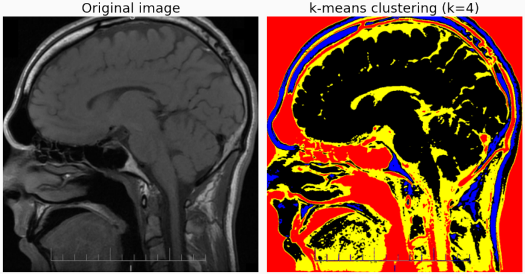
Unsupervised learning

Examples of unsupervised learning include clustering methods (e.g. k-means) often combined with dimensionality reduction (PCA, UMAP).

Unsupervised learning

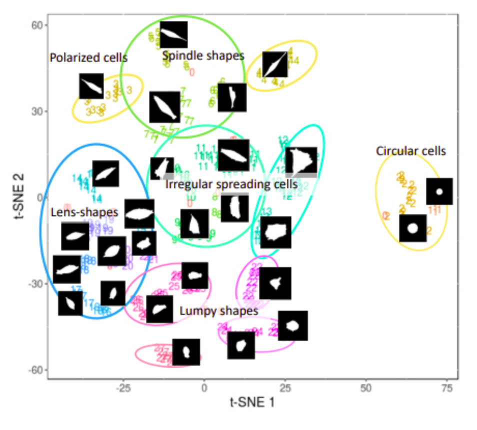
Examples of unsupervised learning include clustering methods (e.g. k-means) often combined with dimensionality reduction (PCA, UMAP).

k-means for segmentation (see Lecture 7)



Examples of unsupervised learning include clustering methods (e.g. k-means) often combined with dimensionality reduction (PCA, UMAP).

t-SNE clustering of images



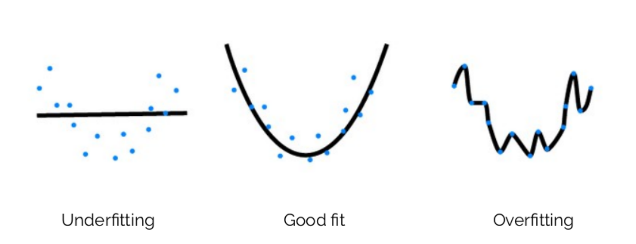
Bhaskar et al, 2019

Dimensionality reduction methods map $Y = f(x_1, x_2, ..., x_n)$ to $Y = f(DR_1, ..., DR_m)$ with $m \leq n$.
They include linear transformations, such as PCA (principal component analysis), and nonlinear transformations, such as t-SNE (t-distributed stochastic neighbor embedding) or UMAP (uniform manifold approximation).

Supervised methods

The bias-variance tradeoff

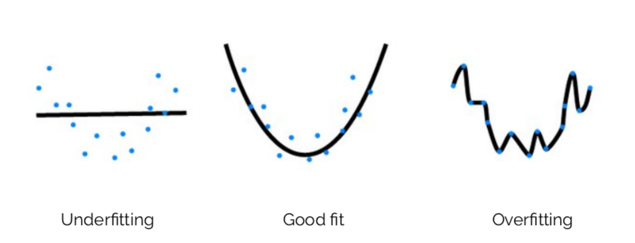
We want to train our model to perform some task. However, just like any statistical model, we don't want to **overfit**.



In ML, we often describe this in terms of **bias** and **variance** errors.

The bias-variance tradeoff

We want to train our model to perform some task. However, just like any statistical model, we don't want to **overfit**.



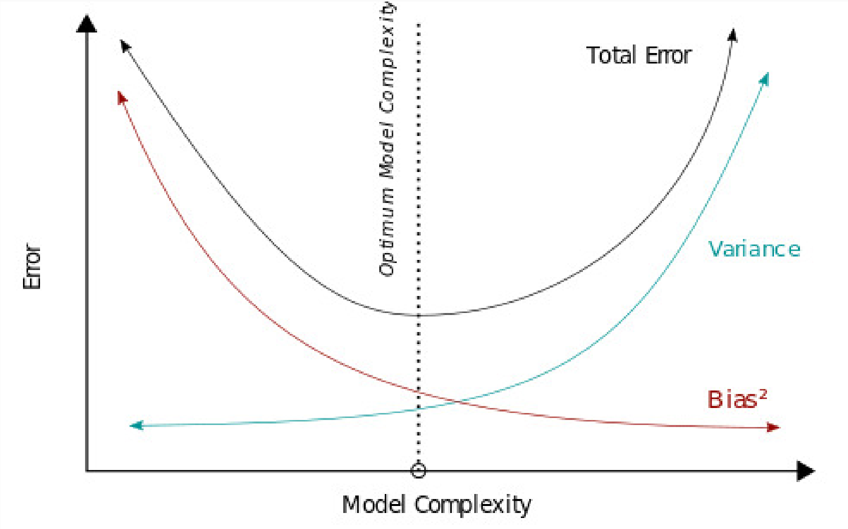
In ML, we often describe this in terms of **bias** and **variance** errors.

- **Bias** derives from erroneous assumptions in the learning algorithm. High bias can cause an algorithm to miss the relevant relations between features and target outputs (underfitting).
- **Variance** derives from sensitivity to small fluctuations in the training set. High variance may result from an algorithm modeling the random noise in the training data (overfitting).

(Adapted from Wikipedia)

The bias-variance tradeoff

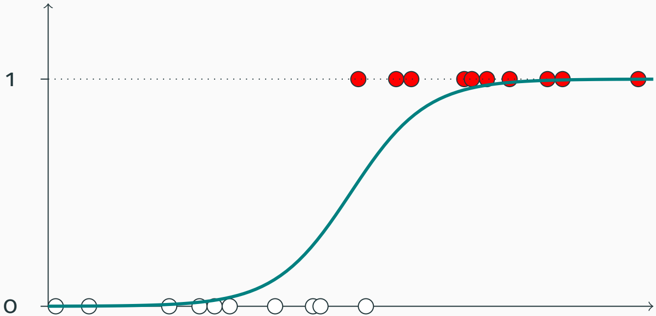
We want to train our model to perform some task. However, just like any statistical model, we don't want to **overfit**.



Many different supervised learning algorithms have been used for image analysis.

Examples of commonly used algorithms include:

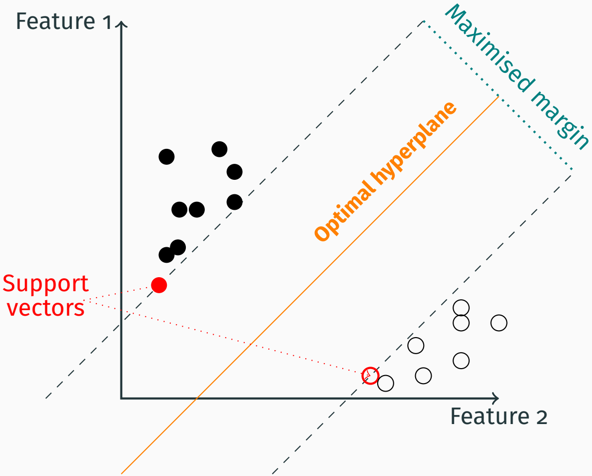
- Logistic regression
- Support vector machines (SVM)
- Random forests (RF)
- Neural networks
- Convolutional neural networks



Logistic regression is a simple supervised learning algorithm that is used to predict the class of a given data point.

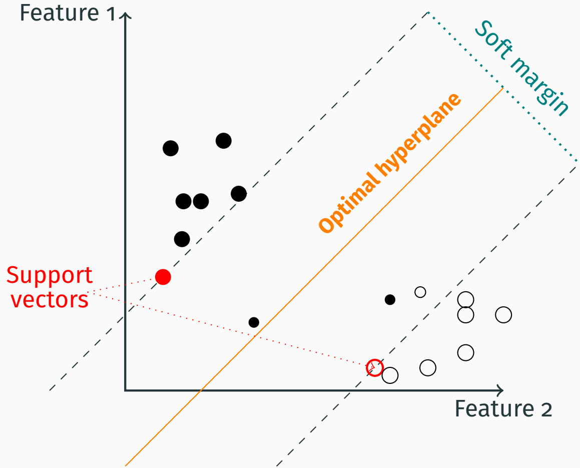
It is mostly used to predict binary outcomes but can be extended to multi-class classification (multinomial logistic regression).

Supervised learning algorithms - support vector machines



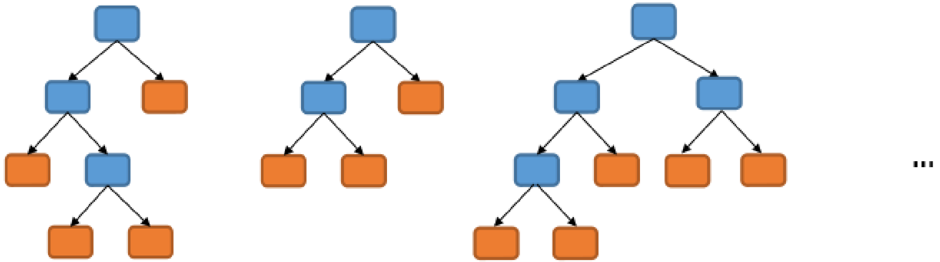
A support vector machine (SVM) uses a linear decision boundary to classify data points. It determines the optimal hyperplane that separates the data points into two classes.

Supervised learning algorithms - support vector machines



A support vector machine (SVM) uses a linear decision boundary to classify data points. It determines the optimal hyperplane that separates the data points into two classes.

Random forest

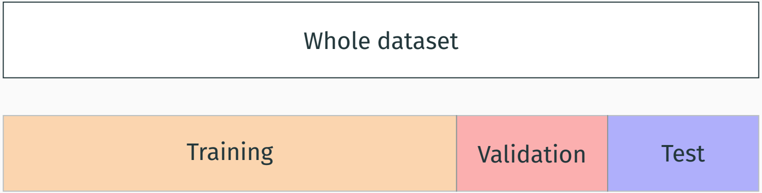


Random forest is an ensemble method for classification and regression.

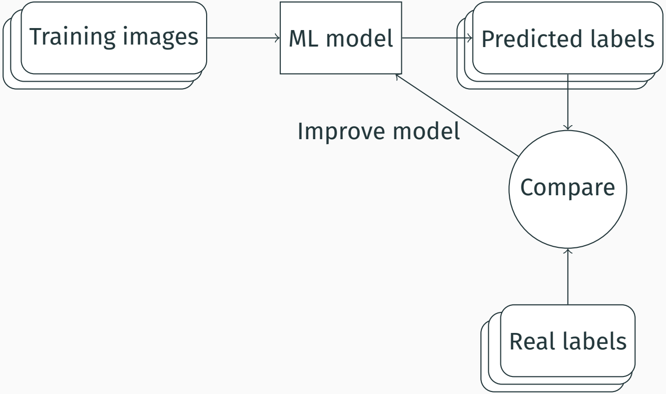
It classifies samples using many binary trees, fitted on various sub-samples of the dataset. A majority votes from these trees decides the outcome. This improves prediction accuracy and controls over-fitting.

In order to avoid overfitting we can split our dataset in three parts:

- **Training set** - used to train the model
- **Validation set** - used to estimate model performance during training or while tuning the model hyperparameters. Especially important for neural network.
- **Test set** - used to test the trained model



The training process



We will explore this more in details in the upcoming lectures!

Model evaluation

Model evaluation

Model evaluation is a crucial step in the machine learning pipeline.

It is used to estimate the performance of the model on unseen data (i.e. the test set).

Depending on the task, we can use different metrics to evaluate the model performance.

Example - classification metrics

When performing a classification task, we can use the following metrics to evaluate the model performance:

- **Accuracy** - the fraction of correct predictions
- **Precision** - the fraction of true positives among all positive predictions
- **Recall** - the fraction of true positives among all actual positives
- **F1 score** - the harmonic mean of precision and recall = $2 \cdot \frac{precision * recall}{precision + recall}$

When performing a classification task, we can use the following metrics to evaluate the model performance:

- **Accuracy** - the fraction of correct predictions
- **Precision** - the fraction of true positives among all positive predictions
- **Recall** - the fraction of true positives among all actual positives
- **F1 score** - the harmonic mean of precision and recall = $2 \cdot \frac{precision * recall}{precision + recall}$

These can also be visualized using a **confusion matrix**:

Predicted $\rightarrow$ **Actual** $\downarrow$	**Positive**	**Negative**
Positive	True positive	False negative
Negative	False positive	True negative

How do I code this?

Lecture 11 will be a coding session, showing you examples of using machine learning for image analysis.