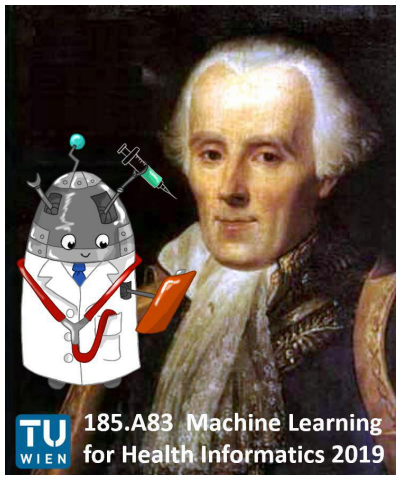
Intro to Neural Networks
Machine Learning for Health Informatics

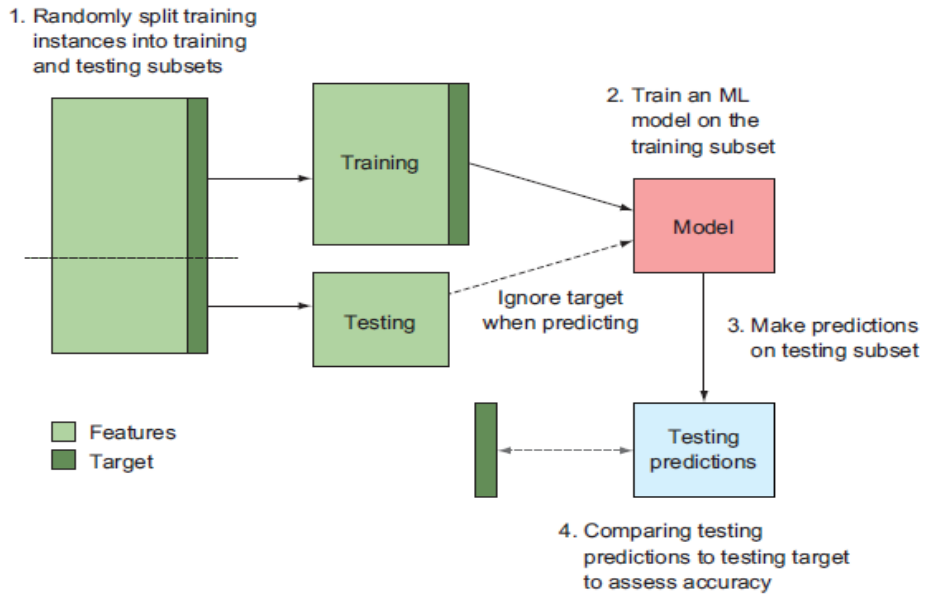
Anna Saranti

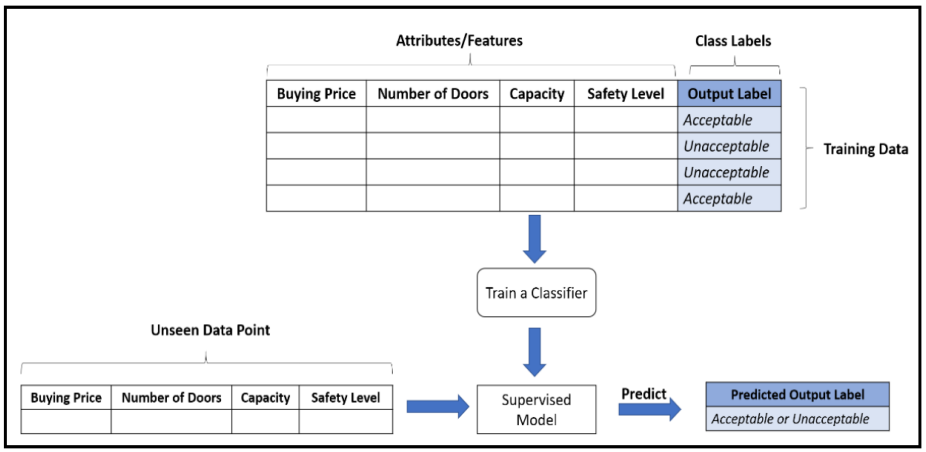
Holzinger Group hci-kdd.org

01.04.2019

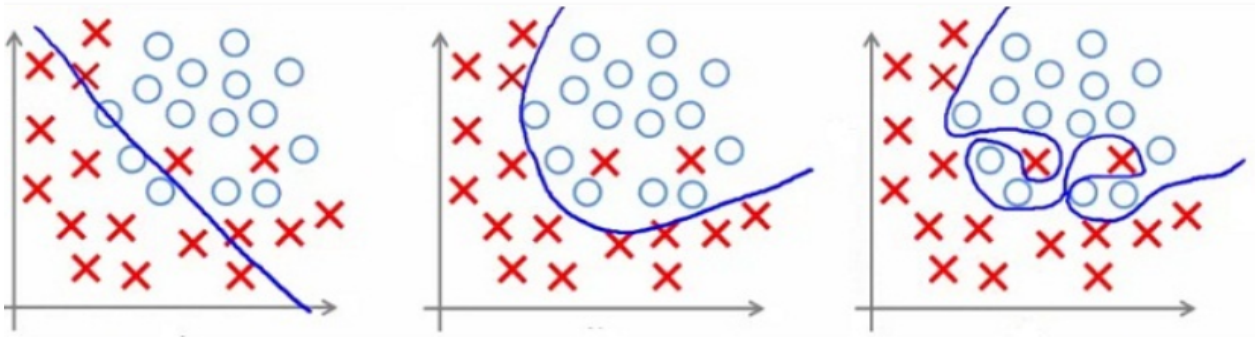


Dataset splitting and overfitting (1/5)





Dataset splitting and overfitting (2/5)



Under-fitting

(too simple to explain the variance)

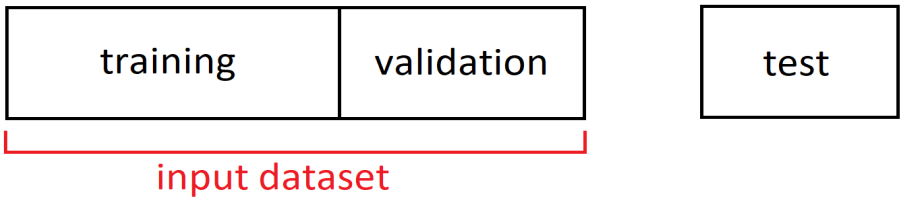
Appropriate-fitting

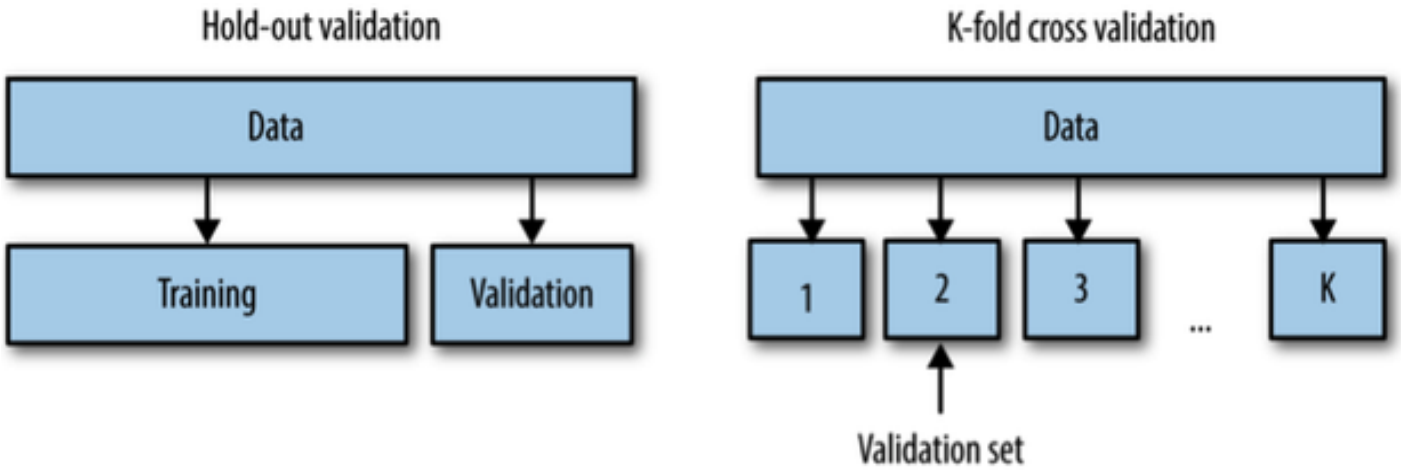
Over-fitting

(forcefitting -- too good to be true)

Dataset splitting and overfitting (3/5)

- Pretend that part of the data is not seen yet
- The model makes a statement about this set - does this hold?



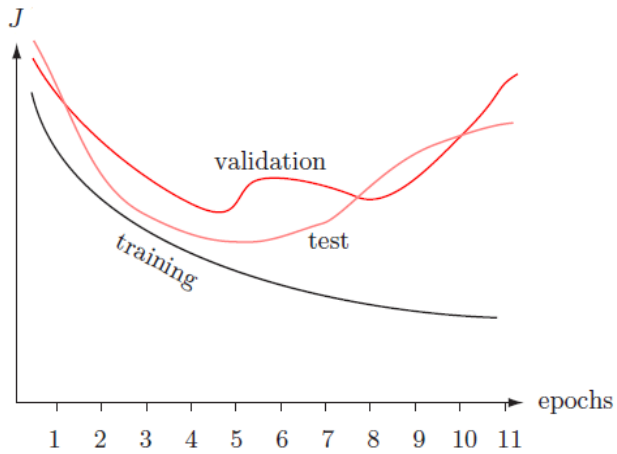


Dataset splitting and overfitting (4/5)

- `https://scikit-learn.org/stable/`
- `sklearn.model_selection.train_test_split`
- `sklearn.model_selection.StratifiedShuffleSplit`, `sklearn.model_selection.StratifiedKFold`
- Training and test set should have roughly the same percentage of data in each class
- Beware that shuffling may spread a characteristic that exists in one part of the data in all the folds!

Dataset splitting and overfitting (5/5)

- Overfitting: Model memorizes and does not generalize to unseen data (test set)



Classification Metrics

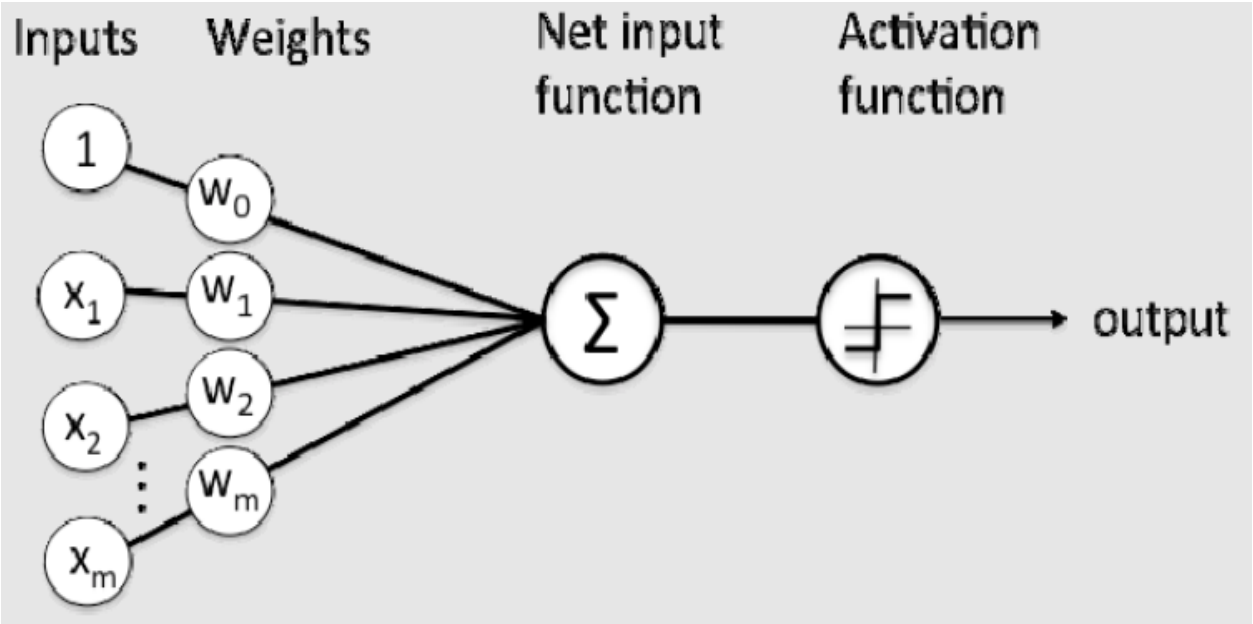
- **Don't use accuracy!** The ratio of correct predictions) is not enough when the dataset is imbalanced. If a data set contains 90% of samples belonging to a single class, the classifier that by default (no programmed logic) only chooses this class, has 90% accuracy. The classifier is not informative - did not learn anything.
- Confusion matrix - compare the predicted class with the actual class:

		Predicted	
		Negative	Positive
Actual	Negative	TN	FP
	Positive	FN	TP

- Mutual information - measures how many bits the classifier's output conveys about the predicted target.
  Helps comparing multi-class classifiers, not directly possible with confusion matrices 2×2, 4×4.

Neural Network for Classification (1/5)

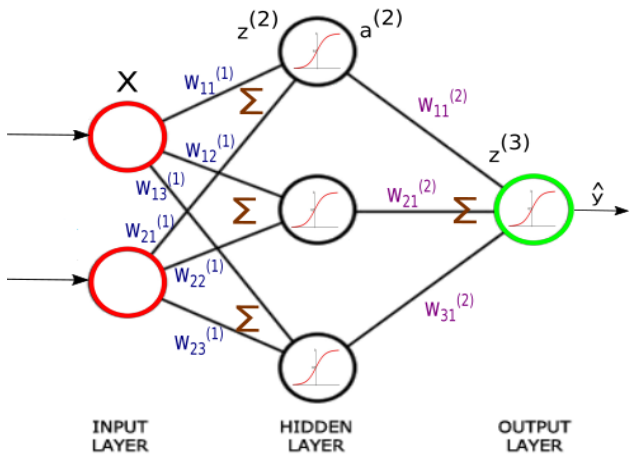
One artificial neuron



- Prediction: $\hat{y} = \text{activation}(\sum_i x_i w_i + b) = f(x)$
- Error: $J = y - \hat{y}$

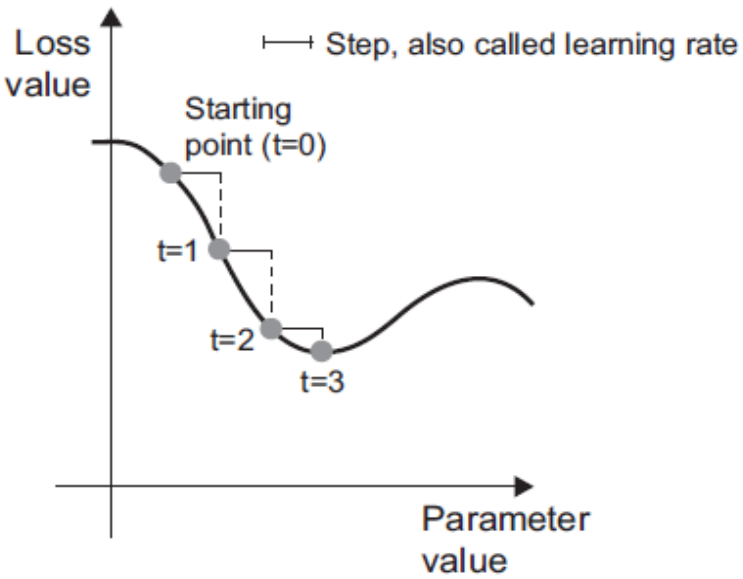
Neural Network for Classification (2/5)

- Learn a non-linear function of the input that approximates the output/target variable
- Architecture is not known, it must be learned as the parameters
- The input must be numeric and has to be scaled

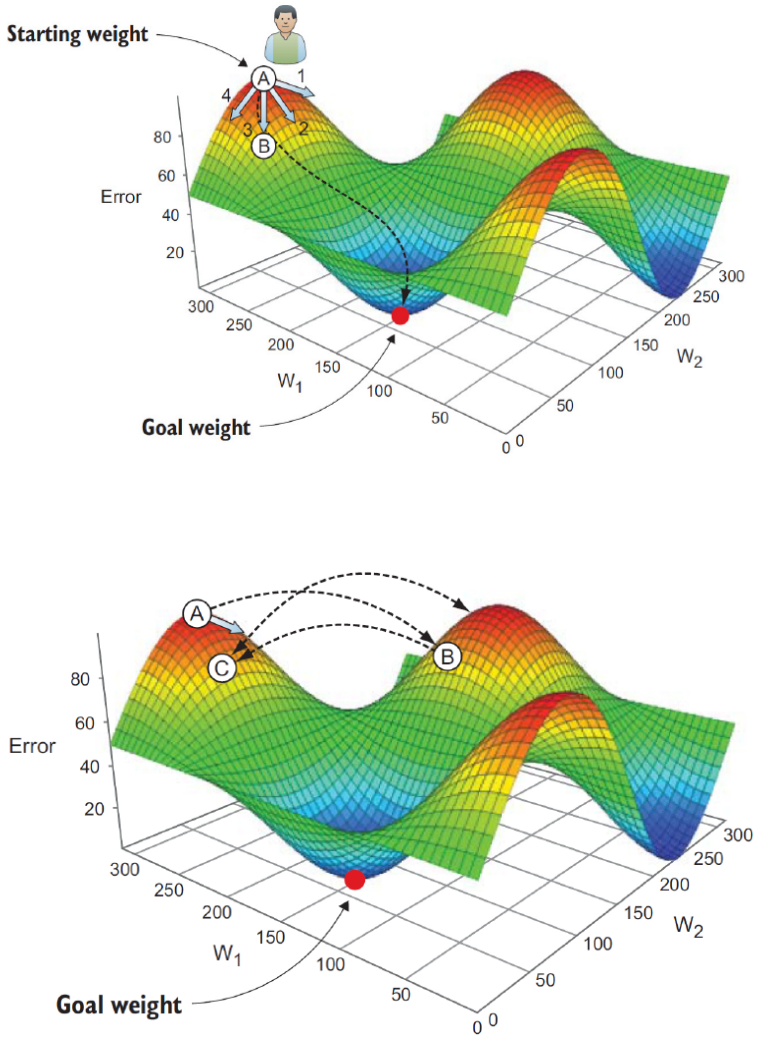


Neural Network for Classification (3/5)

- Various non-linear activation functions (sigmoid, tanh, ReLU)
- Different methods to adapt step-wise the parameters to minimize the error - Bigger steps in the beginning, smaller in the end.
- Different architectures depending on the problem (CNN, RNN)
- Adjust a little the parameters/weights a little according to the error (difference) of the computed with the actual (target) (Back-propagation)



Neural Network for Classification (4/5)



Neural Network for Classification (5/5)

- Repeat until the parameters do not change a lot
- Don't learn too much - don't overfit.
- Check the output of the hidden layers and the values of the weights to discover some patterns

References

- Duda, Richard O., Peter E. Hart, and David G. Stork (2012)
  Pattern classification [Chapter 6]
  John Wiley & Sons
- Christopher M. Bishop (2006)
  Pattern Recognition and Machine Learning [Chapter 5]
  Springer
- Aurélien Géron (2017)
  Hands-On Machine Learning with Scikit-Learn and TensorFlow
  O'Reilly
- David J.C. MacKay (2003)
  Information Theory, Inference, and Learning Algorithms
  Cambridge University Press