Notes on *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*

Jonathan Chen

May 27, 2025

Contents

1 Machine Learning Landscape

1.1 What is ML?

- **ML**: science of programming computers so they can learn from data
- **training set**: examples the system uses to learn
  - each example = a sample / training instance
- **model**: system that learns and makes predictions
- **performance measures**: used to evaluate the model
- situations to use ML
  - if rules are too complicated to write completely
  - rules could be changing constantly
  - you want to use ML and learn from how they learned
- **data mining**: digging into large amounts of data and finding patterns

1.2 Types of ML

- training supervision
  - supervised
    - training set contains labels
    - divided into classification and regression tasks based on the type of labels
      - **classification**: categorical labels
      - **regression**: numerical labels
  - unsupervised
    - unlabeled data
    - **clustering** algorithms: cluster data into bins
    - **visualization** algorithms: outputs 2D or 3D representations of data that can be plotted
    - **dimensionality reduction**: simplify data without losing too much information
    - **feature extraction**: transforming raw data into meaningful features; often a good idea to do dimensionality reduction before feeding it to a supervised model
    - **anomaly detection**: check for rare samples
    - **novelty detection**: check for completely unseen data before
    - **association rule learning**: dig into large amounts of data and discover relationships between attributes

  - semi-supervised
    - partially labeled data
    - most are combinations of unsupervised and supervised algorithms
    - example: use unsupervised for grouping, label those data based on grouping, then put into supervised model
  - self-supervised
    - turn fully unlabeled data to fully labeled data
    - example: use masked images as inputs and full images as outputs
    - usually not the final goal, example: want to train network to repair images, then transfer learn and classify
  - reinforcement learning
    - agent observes
    - agent selects action using policy
    - agent chooses action
    - agent gets rewarded / punished
    - agent updates policy as a result
    - iterates until optimal policy is found

- batch vs. online learning
  - batch learning
    - trained using all available data
    - offline and requires lots of time / computing resources
    - **offline learning**: train and then deploy after it is done learning – doesn't learn after deployment
    - **model rot**: performance decays over time as the world evolves, so should regularly retrain
    - impossible to use if:
      - amount of data is huge
      - need to be autonomous
      - limited resources
      - continuous stream of data
  - online learning
    - training a model incrementally with data fed one instance at a time in small **minibatches**
    - fast, on-the-fly learning that adapts to rapidly changing environments
    - good for limited resources
    - **learning rate**
      - high learning rate = fast adaptation but risks forgetting old data

    - low learning rate = slower learning but more stable and less prone to noise or outliers
* bad data can quickly degrade performance so live systems must be monitoried closely through anomaly detection, disabling learning, reverting to previous state of the model

References

[1] Aurélien Géron, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*, 3rd Edition, O'Reilly Media, 2022.