What is Machine Learning

Field of study that gives computer the ability to learn without being explicitly programmed.

— Arthur Samuel, 1959

Why use Machine Learning

- Problems for which existing solutions require a lot of hand tuning or long list of rules: One ML algorithm can often simplify code and perform better.
- Complex problems for which there is no good solution at all using traditional approaches: The best ML techniques can find a solution.
- Fluctuating environments: A ML system can adapt to new data.
- Getting insights about complex problems and large amounts of data.

Types of Machine Learning Systems

- We categorize them according to three identities:
    - Trained with human supervision (How much): Supervised, unsupervised, semi-supervised and reinforcement learning.
    - Incrementally or on the fly: Online vs Batch learning
    - Comparing data points or detecting patterns and build a model for predictions: Instance-based vs model-based learning
- The categories are not distinct from each other, they can be combined. ex. Online, model-based, supervised system.

Supervised/Unsupervised Learning

Supervised learning

- The training data includes the desired solutions, called labels.
- Classification and Regression are two typical tasks.
- Some regression algorithms can be used for classification and vice versa.
- Some important supervised algorithms are:
    - kNN
    - Linear Regression
    - Logistic Regression
    - SVMs
    - Decision Trees and Random Forests
    - Neural Networks

Unsupervised Learning

- The training data is unlabeled
- Some important unsupervised algorithms are:
    - Clustering: Try to detect similar groups.
        - k-Means
        - Hierarchical Cluster Analysis (HCA)
        - Expectation Maximization
    - Visualization: Understand how the data is organized and perhaps identify some unsuspected patterns. and dimensionality reduction: Simplify the data without losing too much information.
        - Principal Component Analysis (PCA)
        - Kernel PCA
        - Locally-Linear Embedding (LLE)
        - t-distributed Stochastic Neighbor Embedding (t-SNE)

        - Association rule Learning: Dig into large data and discover the interesting relation between attributes.
            - Apriori
            - Eclat
    - Another important unsupervised task is anomaly detection. Helps to determine outliers and unusualities in data.

Semisupervised Learning

- Semisupervised algorithms can deal with partially labeled data. ex. Google Photos

Reinforcement Learning

- The learning system, called an agent, can observe the environment, select and perform actions, and get reward in return. It must learn the best strategy called policy.
- ex. Deepmind - AlphaGo

Batch and Online Learning

- Whether or not the system can learn incrementally from a stream of incoming data.

Batch learning

- The system is incapable of learning incrementally; it must be trained using all the data available.
- Generally takes a lot of time and computing resources.
- First trained and then launched.
- All the data is needed to retrain the system.
- May not be able to use if the data is really huge.

Online learning

- You train the system incrementally by feeding it data instances sequentially, either individually or by small

groups called mini-batches. Each step is fast and cheap, so the system can learn on the fly.

- Great system for continuous flow.
- You can get rid of the data after using it.
- One important parameter of this system is how fast they should adapt to changing data: this is called learning rate. If it is high, the system will adapt better but will forget quickly. If it is low, will learn slower, but less sensitive to noise in the new data.
- With new data, your system's performance may decrease.

Instance-Based vs Model-Based Learning

Instance-Based learning

- The system learns examples by heart and then generalizes to new cases using similarity measure.

Model-Based learning

- Another way to generalize from a set of examples is to build a model of these examples, then use that to make predictions. This is called mode based learning.
- How can you know which values will make your model perform best? To answer this question, you need to specify a performance measure. You can either define a utility function (fitness function) that measures how good your model is, or you can define a cost function that measures how bad it is.
- Typical ML project:
    - You study the data
    - You select a model
    - You train in on the training data
    - Finally, you apply the model to make predictions on new cases (inference)

Main Challenges of Machine Learning

Two things that can go wrong are Bad Data, Bad Algorithm…

Insufficient Quantity of Training Data

- It takes a lot of data for most Machine Learning algorithms to work properly.
- If you don't have enough data, none of the other things that you can do can save your system…

Nonrepresentative Training Data

- In order to generalize well, it is crucial that your training data be representative of the new cases you want to generalize to.
- Sampling bias: Systematic error due to a non-random sample of a population
- ex. US Presidental Election 1936, the Literary Digest's biased poll

Poor Quality Data

- If your training data is full of errors, outliers, and noise, it will make it harder for system to detect underlying patterns, so your system is less likely to perform well.

Irrelevant Features

- A critical part of the Machine Learning project is coming up with a good set of features to train on. This process called feature engineering involves:
    - Feature selection: Selecting the most useful features.
    - Feature extraction: Combining existing features to produce a more useful one.
    - Creating new features by gathering new data.

Overfitting the Training data

- The model performs well on the training data, but it does not generalize well.
- Overfitting happens when the model is too complex relative to the amount and noisiness of the training data. The possible solutions are:

    - To simplify the model by selecting one with fewer parameters, by reducing the number of attributes in the training data or by constructing the model.
    - To gather more training data
    - To reduce the noise in the training data
- Constraining a model to make it simpler and reduce the risk of overfitting is called regularization.
- You want to find the balance between fitting the data perfectly and keeping the model simple enough to ensure that it will generalize well.
- The amount of regularization to apply during learning can be controlled by a hyperparameter. A hyperparameter is a parameter of a learning algorithm (not of the model).
- Tuning hyperparameters is an important part of building a Machine Learning system.

Underfitting the Training data

- Opposite of overfitting the data.
- It occurs when your model is too simple to learn the underlying structure of the data.
- Main options to fix this problem are:
    - Selecting more powerful model, with more parameters.
    - Feeding better features to the learning algorithm.
    - Reducing the constraints on the model.

Testing and Validating

- The only way to know how well a model will generalize to new cases is to actually try it out on new cases.
- Split the data: Training set and test set
- The error rate on new cases is called the generalization error (or out of sample error), and by evaluating your model on the test set, you get an estimation of this error. This value tells you how well your model performs on instances that it has never seen before.

- If the training error is low and generalization error is high that means that your model is overfitting.
- It is common to use 80% of the data for training and hold out 20% for testing.
- To solve the issue of overfitting hyperparameters for the test data, you can divide the data to 3, adding validation set to existing ones.
- Train multiple models with various hyperparameters using the training set, you select the model and hyperparameters that perform best on the validation set, and when you are happy with your model you run a single final test against the test set to get an estimate of the generalization error.
- To avoid 'wasting' too much training data in validation sets, use cross-validation: The training set is split into complementary subsets, and each model is trained against a different combination of these subsets, and each model is trained against a different combination of these subsets and validated against the remaining parts. Next final model is trained using these hyperparameters on the full training set.
- ----------

Exercises

1. How would you define machine learning?

Machine Learning is about building systems that can learn from data. Learning means getting better at some task, given some performance measure.

2. Can you name four types of applications where it shines?

Machine Learning is great for complex problems for which we have no algorithmic solution, to replace long lists of hand-tuned rules, to build systems that adapt to fluctuating environments, and finally to help humans learn (e.g., data mining).

3. What is a labeled training set?

A labeled training set is a training set that contains the desired solution (a.k.a. a label) for each instance.

4. What are the two most common supervised tasks?

The two most common supervised tasks are regression and classification.

5. Can you name four common unsupervised tasks?

Common unsupervised tasks include clustering, visualization, dimensionality reduction, and association rule learning.

6. What type of algorithm would you use to allow a robot to walk invarious unknown terrains?

Reinforcement Learning is likely to perform best if we want a robot to learn to walk in various unknown terrains, since this is typically the type of problem that Reinforcement Learning tackles. It might be possible to express the problem as a supervised or semi-supervised learning problem, but it would be less natural.

7. What type of algorithm would you use to segment your customers into multiple groups?

If you don't know how to define the groups, then you can use a clustering algorithm (unsupervised learning) to segment your customers into clusters of similar customers. However, if you know what groups you would like to have, then you can feed many examples of each group to a classification algorithm (supervised learning), and it will classify all your customers into these groups.

8. Would you frame the problem of spam detection as a supervised learning problem or an unsupervised learning problem?

Spam detection is a typical supervised learning problem: the algorithm is fed many emails along with their labels (spam or not spam).

9. What is an online learning system?

An online learning system can learn incrementally, as opposed to a batch learning system. This makes it capable of adapting rapidly to both changing data and autonomous systems, and of training on very large quantities of data.

10. What is out-of-core learning?

Out-of-core algorithms can handle vast quantities of data that cannot fit in a computer's main memory. An out-of-core learning algorithm chops the data into mini-batches and uses online learning techniques to learn from these mini-batches.

11. What type of algorithm relies on a similarity measure to makepredictions?

An instance-based learning system learns the training data by heart; then, when given a new instance, it uses a similarity measure to find the most similar learned instances and uses them to make predictions.

12. What is the difference between a model parameter and a model hyperparameter?

A model has one or more model parameters that determine what it will predict given a new instance (e.g., the slope of a linear model). A learning algorithm tries to find optimal values for these parameters such that the model generalizes well to new instances. A hyperparameter is a parameter of the learning algorithm itself, not of the model (e.g., the amount of regularization to apply).

13. What do model-based algorithms search for? What is the most common strategy they use to succeed? How do they make predictions?

Model-based learning algorithms search for an optimal value for the model parameters such that the model will generalize well to new instances. We usually train such systems by minimizing a cost function that measures how bad the system is at making predictions on the training data, plus a penalty for model complexity if the model is regularized. To make predictions, we feed the new instance's features into the model's prediction function, using the parameter values found by the learning algorithm.

14. Can you name four of the main challenges in machine learning?

Some of the main challenges in Machine Learning are the lack of data, poor data quality, nonrepresentative data, uninformative features, excessively simple models that underfit the training data, and excessively complex models that overfit the data.

15. If your model performs great on the training data but generalizes poorly to new instances, what is happening? Can you name three possible solutions?

If a model performs great on the training data but generalizes poorly to new instances, the model is likely overfitting the training data (or we got extremely lucky on the training data). Possible solutions to overfitting are getting more data, simplifying the model (selecting a simpler algorithm, reducing the number of parameters or features used, or regularizing the model), or reducing the noise in the training data.

16. What is a test set, and why would you want to use it?

A test set is used to estimate the generalization error that a model will make on new instances, before the model is launched in production.

17. What is the purpose of a validation set?

A validation set is used to compare models. It makes it possible to select the best model and tune the hyperparameters.

18. What is the train-dev set, when do you need it, and how do you use it?

The train-dev set is used when there is a risk of mismatch between the training data and the data used in the validation and test datasets (which should always be as close as possible to the data used once the model is in production). The train-dev set is a

part of the training set that's held out (the model is not trained on it). The model is trained on the rest of the training set, and evaluated on both the train-dev set and the validation set. If the model performs well on the training set but not on the train-dev set, then the model is likely overfitting the training set. If it performs well on both the training set and the train-dev set, but not on the validation set, then there is probably a significant data mismatch between the training data and the validation + test data, and you should try to improve the training data to make it look more like the validation + test data.

19. What can go wrong if you tune hyperparameters using the test set?

If you tune hyperparameters using the test set, you risk overfitting the test set, and the generalization error you measure will be optimistic (you may launch a model that performs worse than you expect).