Semi-supervised Learning:

Some algorithms that can deal with partially labeled data, usually a lot of unlabeled data and a little bit of labeled data. This is called semisupervised learning. See fig.

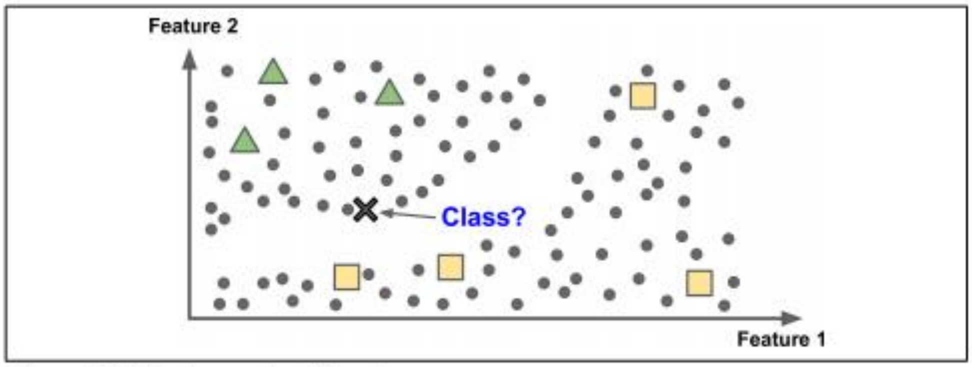

Figure 1-11. Semisupervised learning

Most semisupervised learning algorithms are the combination of supervised and unsupervised learning algorithms. For example, *deep belief networks* (DBNs) are based on unsupervised components called *restricted boltzmann machines* (RBMs) stacked on the top of one another. RBMs are trained in an unsupervised manner, and then the whole system is fine-tuned using supervised learning techniques.

Batch and Online Learning:

Another criterion used to classify ML systems is whether or not the system can learn incrementally from a stream of incoming data.

- **Batch Learning:**
    - In batch learning, the system is incapable of learning incrementally, it must be trained for the available data.
    - It is type of offline learning
    - First the model is trained and then it is launched into production and runs without learning anymore, it just applies what it has learned.
    - For new data the system will be trained again from the scratch containing the new and old data, then stop the old system and replace it with the new one.
    - By updating the system for new data will be hard because it will consume a lot of time and requires computing resources. The updating will be done only after 24 hours or weekly basis.
    - If there is huge amount of training data and the model is automated to train every data, it will cost a lot and with huge amount of data it is impossible to use batch learning.

In the case of huge amount of data and automated model we use Online learning.

- **<u>Online learning:</u>**
  - The system is trained incrementally by feeding it data instances sequentially, either individually or by small groups called mini-batches.

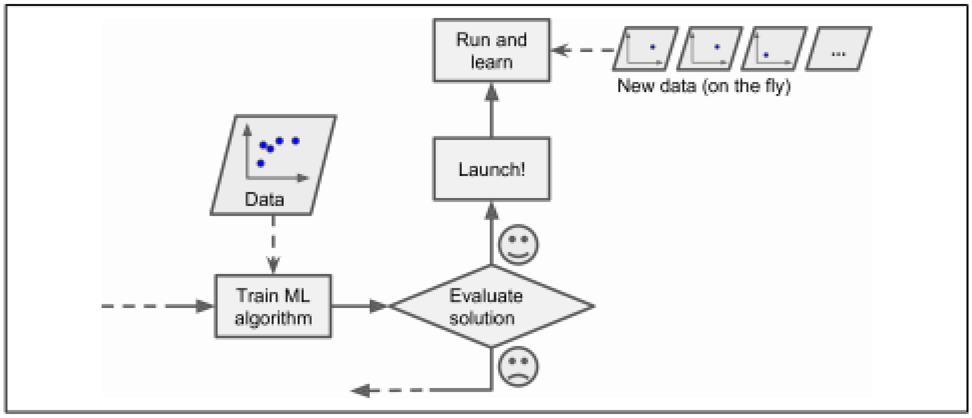

Figure 1-13. Online learning

  - Each learning step is fast and cheap, so the system can learn about the new data on the fly, as it arrives.
  - Online learning is great for the systems that receive the data as a continues and need to adapt to change rapidly.
  - It is good option for someone having limited computing resources. In this learning once the system learns about the new data then that data can be discarded (unless not needed for replay). This can save a huge amount of memory.
  - It is also helpful for very huge datasets that cannot fit in one machine's main memory (this is called out-of-core learning).

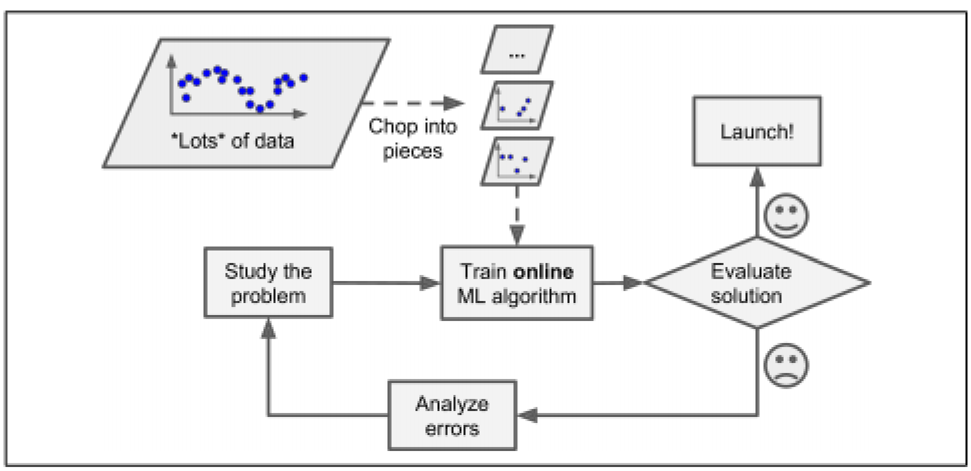

Figure 1-14. Using online learning to handle huge datasets

  - One important parameter of the online learning is that how fast it should adapt to changing data: this is called learning rate. If the learning rate is set high, it will quickly forget about the old data by focusing only on new data. Conversely, if the learning rate is very slow, the system will have more inertia , that it will learn more slowly but it will less sensitive to noise in the new data or outliers.

- Big challenge for online learning is that if bad data is fed to the system, its performance will decline. So the system will monitored very closely.

Main Challenges of Machine Learning:

We can two problems if we want to train an algorithm using some data, that are "bad algorithm" and "bad data". Let's start with examples of bad data.

- **Insufficient Quantity of Training data.** (quiet large amount of data is required for a ML model to perform good enough)
- **Non-representative Training data.** (containing missing values)
- **Poor Quality Data** (data full of errors, outliers and noise)
- **Irrelevant Features.** (Requires *feature engineering*)

Examples of bad algorithms:

- **Overfitting the training data:** (Considering a population similar as an individual which is overgeneralizing)
  Constraining a model to make it simpler and reduce the risk of overfitting is called regularization. The amount of regularization to apply during learning can be controlled by a *hyper-parameter.* A hyper-parameter is a parameter of a learning algorithm. It must be set prior to training and remains constant during training. If the hyper-parameter is too large then model will be flate (a slope close to zero), the learning algorithm will not tend to overfit the training data, but it will be less likely to find a good solution.
  Degree of freedom: In machine learning we may refer degree of freedom as number of parameter. Consider example of Linear regression having two parameters (theta1 and theta2). This gives the learning algorithm two degree of freedom to adapt the model to the training data. Degree of freedom also define the complexity level of the algorithm.

  Number of parameters estimated from data = Degree of freedom

  Degree of freedom= complexity of the model

- **Underfitting the training data:**
  It is the opposite of the overfitting. Model is too simple to learn the underlying structure of the data. Data will be complex for the model, so it will make inaccurate predictions even on the training examples.

  The main options to fix this problem are:
  1. Select a more powerful model, with more parameters.

2. Select best features to the learning algorithm.
3. Reducing constraints on the model (e.g., reducing regularization hyper-parameter)

- **Testing and Validating**
- **Hyperparameter Tuning and Model Selection**
- **Data Mismatch**