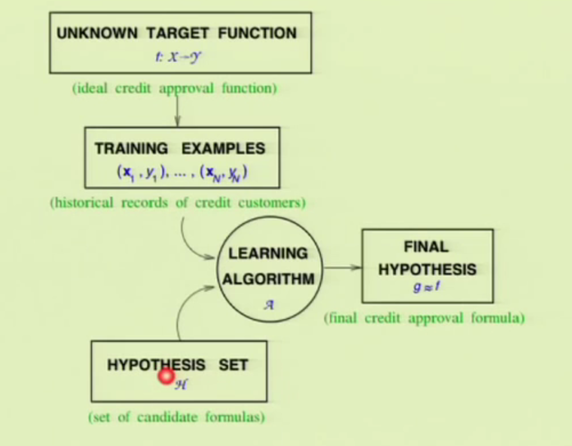
Hypothesis in machine learning:

In a machine learning problem where the input is denoted by **x** and the output is *y*  
  
In order to do machine learning, there should exist a relationship (pattern) between the input and output values. Lets say that this the function  
*y*=*f*(**x**), this known as the **target function.**  
  
However, *f*(.) is unknown function to us.  so machine learning algorithms try to guess a ``hypothesis'' function *h*(**x**) that approximates the unknown *f*(.), the set of all possible hypotheses is known as the Hypothesis set *H*(.), the goal is the learning process is to find the final hypothesis that best approximates the unknown target function.  
  
Different machine learning models have different hypothesis sets, For example the 2d- perceptron has the hypothesis set  
*H*(**x**)={*sign*(*w*1∗*x*1+*w*2∗*x*2+*w*0)∀*w*0,*w*1,*w*2}



Terminology:

Essentially, the terms "classifier" and "model" are synonymous in certain contexts; however, sometimes people refer to "classifier" as the learning algorithm that learns the model from the training data. To makes things more tractable, let's define some of the key terminology:

* *Training sample:* A training sample is a data point *x* in an available training set that we use for tackling a predictive modeling task. For example, if we are interested in classifying emails, one email in our dataset would be one training sample. Sometimes, people also use the synonymous terms *training instance* or *training example*.
* *Target function:* In predictive modeling, we are typically interested in modeling a particular process; we want to learn or approximate a particular function that, for example, let's us distinguish spam from non-spam email. The *target function* *f(x) = y* is the true function *f* that we want to model.
* *Hypothesis:* A hypothesis is a certain function that we believe (or hope) is similar to the true function, the *target function* that we want to model. In context of email spam classification, it would be the *rule* we came up with that allows us to separate spam from non-spam emails.
* *Model:* In machine learning field, the terms *hypothesis* and *model* are often used interchangeably. In other sciences, they can have different meanings, i.e., the hypothesis would be the "educated guess" by the scientist, and the *model* would be the manifestation of this *guess* that can be used to test the hypothesis.
* *Learning algorithm:* Again, our goal is to find or approximate the *target function*, and the learning algorithm is a set of instructions that tries to *model* the target function using our training dataset. A learning algorithm comes with a *hypothesis space*, the set of possible hypotheses it can come up with in order to model the unknown target function by formulating the *final hypothesis*
* *Classifier:* A classifier is a special case of a *hypothesis* (nowadays, often learned by a machine learning algorithm). A *classifier* is a *hypothesis* or *discrete-valued function* that is used to assign (categorical) class labels to particular data points. In the email classification example, this classifier could be a hypothesis for labeling emails as spam or non-spam. However, a *hypothesis* must not necessarily be synonymous to a *classifier*. In a different application, our *hypothesis* could be a function for mapping study time and educational backgrounds of students to their future SAT scores.

So, we can say that a *classifier* is a special case of a *hypothesis* or *model*: a classifier is a function that assigns a class label to a data point.

Notes from paper:

. Cost effective to generalize from example, rather than manual programming if we have lots of example.

. Purpose of this article is to communicate knowledge not found in text books, which should be known by practitioners. Twelve pitfalls are being described in this article.

. While choosing a machine learning approach for an identified problem, we consider three components for our selection of one approach among many other available in literature.

. Three components are representation, evaluation, and optimization.

. The way we represent our data to computer and hence the set of classifiers (hypothesis space) plays an important step in choosing one ml approach.

. Evaluation function (scoring function) is needed to pull out bad classifiers from the bad ones.

. If evaluation functions have equivalent scores for several classifiers, one with higher efficiency of learning is chosen and this is where we look for optimization technique for learners.

. It is easy to get hope of false confidence by getting higher accuracy on a model with smaller data set, unless a small data set is kept separate and unseen as test set.

. For parameter tuning and testing, cross-validation is the best thing to do.

. We optimize function we don’t know using training error to reduce test error, so many a times, local optimum(greedy) might produce better result than global optimum.

. Every learner must embodies some knowledge about problem statement, otherwise it is difficult to beat random guessing.

. Machine learning uses the power of induction, putting little knowledge to take out large knowledge (generalization). Hence, representation of this knowledge helps in our choice in a approach

. It is good to get 75% accuracy on both training and test set, not 100% on training and 50% on test set. The inconsistency is a result of over fitting.

. Over fitting has generalization error which can be decomposed into bias and variance. While bias means learning wrong thing consistently, variance is learning randomness irrespective of real signals.

. Strong false assumptions can be better than weak true ones, because a learner with the latter needs more data to avoid over fitting.

. Over fitting can be avoided by cross-validation, but popular one is regularization term to evaluation function that penalize classifier with more structure.

. In case of scares data availability, statistical significance test such as chi-squared can be done before adding new structure.

. It is easy to avoid over fitting(variance) by under fitting (bias) which is wrong thing to do as well. Hence, simultaneously avoiding both require learning a perfect classifier.

. Curse of dimensionality can outweigh the more number of feature collection for an example problem.

. The main role of theoretical guarantee in machine learning is not as a criterion for practical decisions, but as a source of understanding and driving force for algorithm design.

. If learner A is better than learner B given infinite data, B is often better than A given finite data.

. Constructing features about of raw data is the key for getting learners learn easy and fast. Most useful learners are those that incorporate knowledge.

. Sometimes, features that look irrelevant in isolation may be relevant in combination. So, there is no replacement of smart feature engineering.

. Machine learning researchers are mostly concerned with designing a better learning algorithm, but quick success path most of the time lies in collecting even more data for a dumb learning algorithm.

. It pays to try the simplest learners first because complex learners have more knobs to tune in order to get good result.

. There are two kinds of learner, parametric (fixed) that can only take advantage of so much data. Example linear representation.

. Another one is non-parametric. Example is decision tree. They usually end up learning many more parameters. In principal, they can learn any function given sufficient data, but they may not.

. Thus, fixed one that make the most of the data and computing resources available often pay off in the end.

. Integrating learners (model ensembles) is now standard. Bagging, boosting and stacking are those methods of ensemble.

. In bagging, we generate random variations of training set by resampling, learn classifier on each, and combine the results by voting.

. In boosting, training examples have weights, and these are varied so that each new classifier focuses on the examples that previous ones tended to get wrong.

. In stacking, the outputs of individual classifier become the inputs of a high level learner that figures out how best to combine them.

. BMA average the individual predictions of all classifiers in hypothesis space, weighted by how well the classifiers explain the training data and how much we believe in them a priori.

. The generalization error of a boosted ensemble continues to improve by adding classifiers even after the training error has reached zero.

. Contrary to intuition, there is no necessary connection between the number of parameters of a model and its tendency to over fit.

. A learner with a larger hypothesis space that tries fewer hypotheses from it is less likely to over fit than one that tries more hypotheses from a smaller space.

. Simpler hypotheses should be preferred because simplicity is a virtue in its own right, not because of a hypothetical connection with accuracy.

. Given finite data, time and memory, standard learners can learn only a tiny subset of all possible functions, and these subsets are different for learners with different representation.

. Even a learner can be represented easily, more concerning point is whether it can be learned. And it pays to try different learners and possibly combine them.