

Introduction to Machine Learning

What is learning?

“Learning is any process by which a system improves performance from experience.” - Herbert Simon

What is machine learning?

The hype answer:

“Machine learning is the next Internet” - Tony Tether,
Director, DARPA

“Machine learning is the hot new thing” - John Hennessy,
President, Stanford

“Machine learning is today’s discontinuity” - Jerry Yang,
CEO, Yahoo

“Machine learning is the new electricity” - Andrew Ng, Chief
Scientist Baidu

“Software is eating the world.” - Marc Andreessen, co-
Author of Mosaic, founder of Andreessen Horowitz (VC
firm).

“AI is eating software.” - Jensen Huang, CEO Nvidia

What is machine learning?

The hype is real.



From slate.com

What is machine learning?

Definition by Tom Mitchell (1998):

Machine Learning is the study of algorithms that

- ▶ improve their performance P
- ▶ at some task T
- ▶ with experience E .

A well-defined learning task is given by $\langle P, T, E \rangle$.

Examples of Machine Learning Tasks ²

Improve on task **T**, with performance **P**, given experience **E**

T: Playing checkers

- ▶ **P**: Percentage of games won against an arbitrary opponent
- ▶ **E**: Playing practice games against itself

T: Recognizing hand-written words

- ▶ **P**: Percentage of words correctly classified
- ▶ **E**: Database of human-labeled images of handwritten words

T: Driving on four-lane highways using vision sensors

- ▶ **P**: Average distance traveled before a human-judged error
- ▶ **E**: A sequence of images and steering commands recorded while observing a human driver.

T: Categorize email messages as spam or legitimate.

- ▶ **P**: Percentage of email messages correctly classified.
- ▶ **E**: Database of emails, some with human-given labels

²Slide credit: Ray Mooney

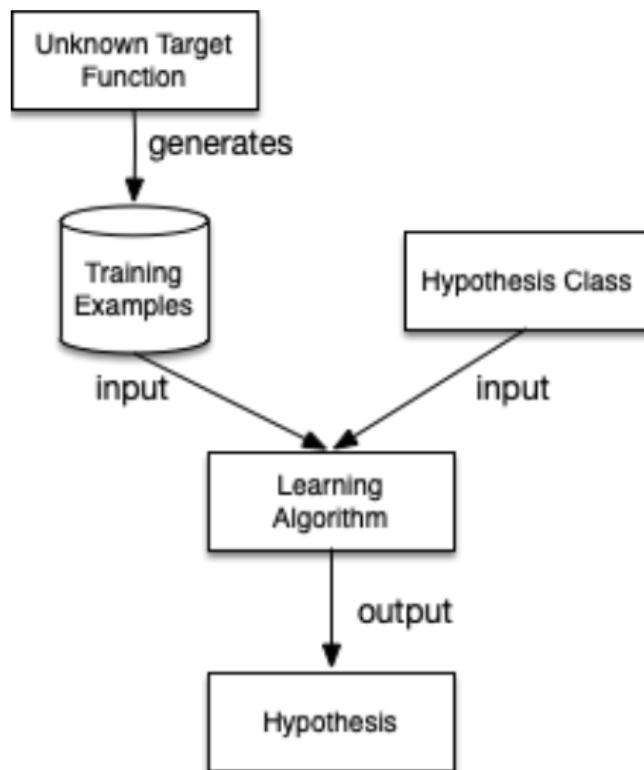
Elements of Machine Learning Tasks

Every machine learning task includes

- ▶ data from which to learn,
- ▶ a model that takes input data and produces an “answer”,
- ▶ a loss function that quantifies the badness of our model, and
- ▶ an algorithm that adjusts the model’s parameters to minimize the loss

Our model, or hypothesis, comes from a model/hypothesis class. Once the parameters are learned, we have an instance of the hypothesis class tuned to our particular machine learning problem.

Machine Learning Architecture



Traditional Programming vs Machine Learning

Traditional Programming



Machine Learning



When to use Machine Learning

ML is used when ³:

- ▶ Human expertise does not exist (navigating on Mars)
- ▶ Humans can't explain their expertise (speech recognition, vision)
- ▶ Models must be customized (personalized medicine)
- ▶ Models are based on huge amounts of data (genomics)

³Ethem Alpaydin, Introduction to Machine Learning

Kinds of Machine Learning Tasks

- ▶ Classification: identify the correct label for an instance
 - ▶ Is this a picture of a dog?
 - ▶ Which radio emitted the signal we received?
 - ▶ Will this customer respond to this advertisement?
- ▶ Clustering: identify the groups into which instances fall
 - ▶ What are the discernible groups of ... customers, cars, colors in an images
- ▶ Agent behavior
 - ▶ Given the state, which action should the agent take to maximize its goal attainment?

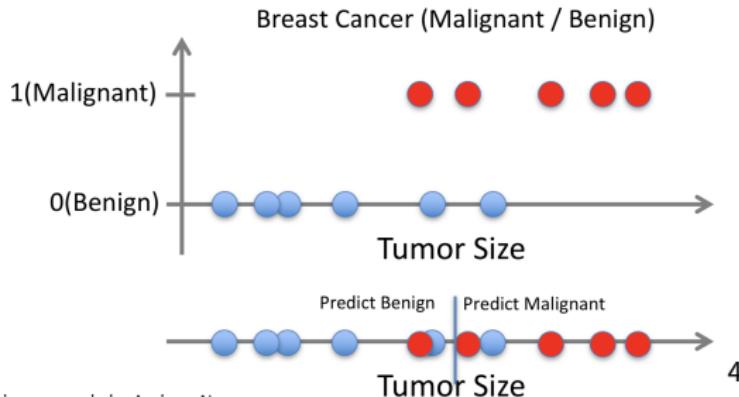
Categories of Machine Learning Algorithms

- ▶ Supervised
 - ▶ Learn from a training set of labeled data – the supervisor
 - ▶ Generalize to unseen instances
- ▶ Unsupervised
 - ▶ Learn from a set of unlabeled data
 - ▶ Place an unseen instance into appropriate group
 - ▶ Infer rules describing the groups
- ▶ Reinforcement learning
 - ▶ Learn from a history of trial-and-error exploration
 - ▶ Output is a /policy/ – a mapping from states to actions (or probability distributions over actions)

Classification using supervised learning methods makes up the lion's share of machine learning.

Supervised Learning Example: Breast Cancer Identification

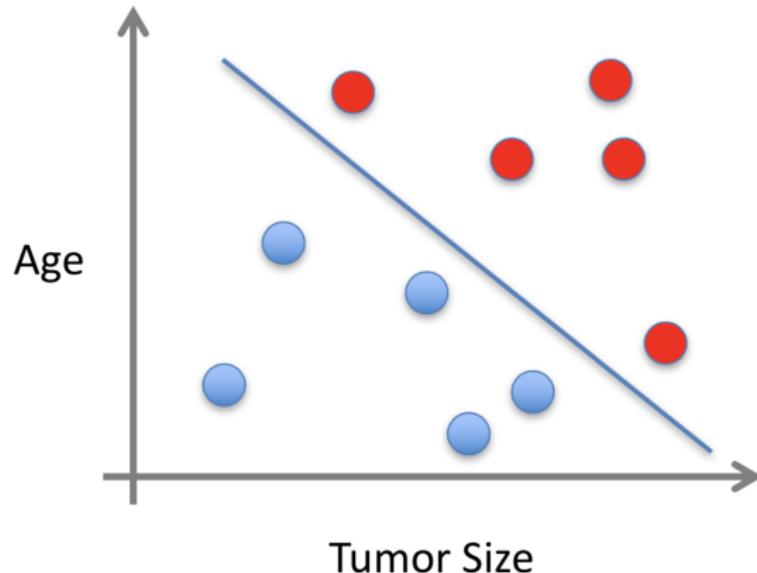
- Given $(\vec{x}_1, y_1), (\vec{x}_2, y_2), \dots, (\vec{x}_n, y_n)$
- Learn a function $h(\vec{x})$ to approximate $f(\vec{x})$ – the *target* function – to predict y given \vec{x} – if y is categorical == classification – if y is real-valued == regression



⁴From Andrew Ng

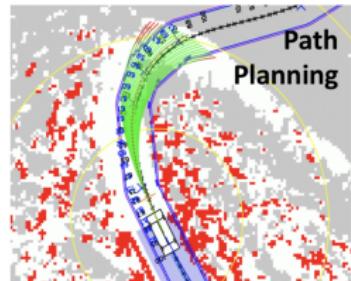
Supervised Learning Example: Breast Cancer Identification

We can often obtain a better model with more features.



But as we'll learn, we don't get this extra power for free.

State of the Art in Machine Learning – Autonomous Cars



Images and movies taken from Sebastian Thrun's multimedia website.

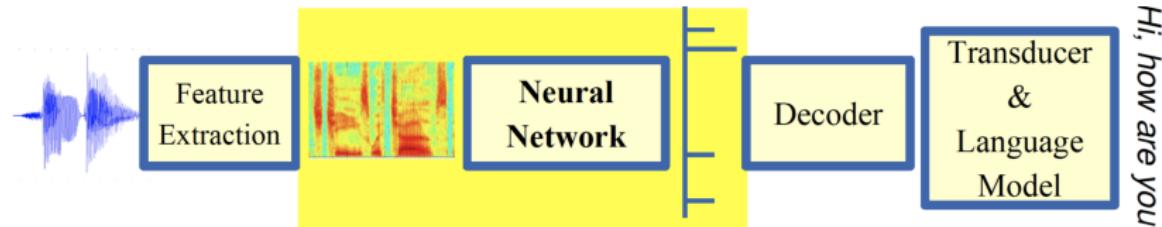
16

State of the Art in Machine Learning – Scene Labeling

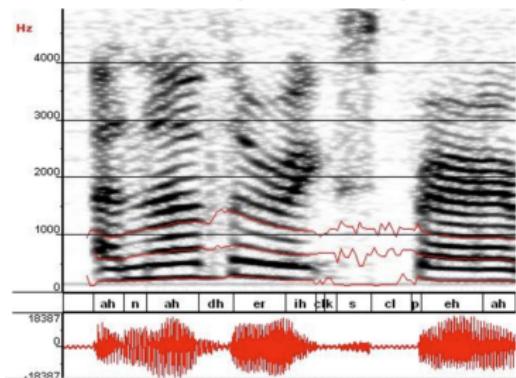


⁵Farabet et al. ICML 2012, PAMI 2013

State of the Art in Machine Learning – Speech Recognition



ML used to predict of phoneme states from the sound spectrogram



Deep learning has state-of-the-art results

# Hidden Layers	1	2	4	8	10	12
Word Error Rate %	16.0	12.8	11.4	10.9	11.0	11.1

Baseline GMM performance = 15.4%

[Zeiler et al. "On rectified linear units for speech recognition" ICASSP 2013]

A Practical Example: Iris Classification

It's a rite of passage to apply supervised learning to the Iris data set. The canonical source for the Iris data set is the [UCI Machine Learning Repository](#). Download [iris.data](#).



Iris Data

The data set contains 150 instances of Iris flowers with

- ▶ 4 features:
 - ▶ sepal_length
 - ▶ sepal_width
 - ▶ petal_length
 - ▶ petal_width

and

- ▶ 3 classes:
 - ▶ Iris-setosa
 - ▶ Iris-versicolour
 - ▶ Iris-virginica

Scikit-learn Recipe

1. Set up feature matrix and target array
2. Choose (import) model class
3. Set model parameters via arguments to model constructor
4. Fit model to data
5. Apply model to new data

Let's apply this recipe to a data set.

Scikit-learn Data Representation

The basic supervised learning setup in Scikit-learn is:

- ▶ Feature Matrix
 - ▶ Rows are instances
 - ▶ Columns are features
- ▶ Target array
 - ▶ An array of `len(rows)` containing the training labels for each instance

We can easily obtain these with a Pandas DataFrame.

Step 1: Iris feature matrix and target array

From the description on the [Iris Data Set page](#) we know that the Iris instances have four features – (sepal_length, sepal_width, petal_length, petal_width) – and three classes – (Iris-setosa, Iris-versicolour, Iris-virginica). We can read these into a DataFrame with

```
1 import pandas as pd
2 import sklearn
3 iris = pd.read_csv("iris.data", names=["sepal_length",
4                                         "sepal_width",
5                                         "petal_length",
6                                         "petal_width",
7                                         "species"])
```

Note: this DataFrame is in [tidy format](#).

Step 1.1: Scikit-learn Input Data

For Scikit-learn we need a feature matrix `x` and target array `y`:

```
1 X_iris = iris.drop("species", axis=1)  
2 y_iris = iris["species"]
```

We can check that the number of samples in the feature matrix equals the number of labels in the target array with

```
1 X_iris.shape[0] == y_iris.shape[0] # True
```

There are 150 samples and 150 target labels.

Step 2: Choose a model

No hypothesis class (aka model class, aka algorithm, aka estimator) is best for all data⁶. You must choose your model class based on the data. Things to consider:

- ▶ What's the dimensionality of your data?
- ▶ Are your features linearly separable?
- ▶ Are your features numeric or categorical?

Scikit-learn calls models *estimators*.

A good first choice is a linear model class, so we'll use a support vector machine (SVM).

⁶Wolpert and Macready, *No Free Lunch Theorems for Optimization*

Step 3: Set model parameters

```
1 from sklearn import svm  
2 model = svm.SVC(kernel="linear")
```

Most parameters are optional, with reasonable default values.
Because we know the Iris data set is so well-suited to linear classifiers
we choose a `linear` kernel (default is `rbf` – radial basis function)

Step 4: Fit model to data

We want to separate our data into non-overlapping training and test subsets. Since the data in our data set are arranged in a neat order, we should randomize the samples and split in a way that represents each class equally in the training and test sets. Scikit-learn provides a library function to do this:

```
1 from sklearn.model_selection import train_test_split
2 X_iris_train, X_iris_test, y_iris_train, y_iris_test = \
3     train_test_split(X_iris,
4                       y_iris,
5                       random_state=1)
```

Now we can train our classifier on the training data (fit the model to the training data).

```
1 model.fit(X_iris_train, y_iris_train)
```

Step 5: Apply model to new data

To apply the trained model to new (unseen) data, pass an array of instances to `predict`:

```
1 y_iris_model = model.predict(X_iris_test)
```

We can test the generalization error (how well the classifier performs on unseen data) using the built-in accuracy score:

```
1 from sklearn.metrics import accuracy_score
2 accuracy_score(y_iris_test, y_iris_model)
3 1.0
```

As you can see, a linear SVM classifier works perfectly on the Iris data. Try out different classifiers to see how well they perform.

A Scikit-learn estimator (model/hypothesis) is an object that has `fit` (train) and `predict` (test) methods.

Closing Words

“A breakthrough in machine learning is worth 10 Microsofts.”
– Bill Gates