

Practical Machine Learning



Practical Machine Learning

Lecture: Introduction to Machine Learning

Ted Scully

Course Breakdown and Assessment

- Email: ted.scully@cit.ie
- Weekly Schedule
 - Two hour lecture
 - Practical lab exercises each week (commencing Weds September 18th)
- Module is 100% Continuous Assessment.
 - Project 1 - Develop a machine learning model for a real-world problem and perform a comprehensive analysis. (50%).
 - Project 2 - Perform a comparative analysis of a range of machine learning classification algorithms applied to a dataset from an application domain. (50%).

Content

- Pre-processing
 - Application of pre-processing techniques such as outlier detection, feature selection, imputation of missing data, encoding, normalization, etc.
- Evaluation and Model Selection
 - Best practice evaluation techniques such as precision, recall, confusion matrices and ROC curves. Debugging algorithms using validation and learning curves. Cross fold validation. Model selection using hyper parameter optimization.
- Classification Algorithms
 - Classification algorithms such as decision trees, ensemble technique (bagging and boosting, gradient-boosting), support vector machines, instance-based algorithms, naïve bayes, Bayesian networks, etc.
- Unsupervised Algorithms
 - Overview of unsupervised learning techniques. Example applications of clustering techniques. Introduction to algorithms such as k-means, k-median, dbscan and hierarchical clustering techniques. Optimization and distortion cost function. Random initialization and methods of selecting number of clusters. Silhouette plots.

Resources

We will be using Python 3 (preferably 3.7) as our programming language in this module.

NumPy and Pandas

- [DataCamp NumPy Tutorial](#) – Accessible and easy to understand tutorial to get started with NumPy
- [NumPy Tutorial](#) – Short overview of NumPy and basic Python data structures. It also covers SciPy (which you don't need) and basic Matplotlib (which you will be covering later in the programme as part of visualization).
- [DataCamp Pandas Tutorial](#) – Short and easy to understand tutorial on using Pandas

Resources

- **Websites**

- [Machine Learning Stanford](#) – Andrew Ng
- [Machine Learning Class \(Washington\)](#) - Pedro Domingos
- [Udacity Machine Learning](#) - Sebastian Thrun
- [UCI Data Repository](#)
- [Kaggle](#)

Resources

- **Books**

- [Hands-On Machine Learning with Scikit-Learn and TensorFlow](#) – Aurelien Geron (** New hard back version due out on September 30th)
- [Python Machine Learning](#) 2nd Edition - (Sebastian Raschka)
- [Fundamentals of Machine Learning for Predictive Data Analytics](#) – (John Kelleher, Brian MacNamee , Aoife D'Arcy)

Software Options (1. Anaconda)

- Anaconda is an open-source distribution of Python.
- It comes with a range of essential packages such as NumPy, Pandas, Scikit-Learn and Matplotlib.
- Spyder IDE or Jupyter Notebook.
- Version control is managed by Conda.
- Download [Anaconda](#) (Python 3.7 version)



Windows



macOS



Linux

Anaconda 2019.07 for Windows Installer

Python 3.7 version

Download

64-Bit Graphical Installer (486 MB)

32-Bit Graphical Installer (418 MB)

Python 2.7 version

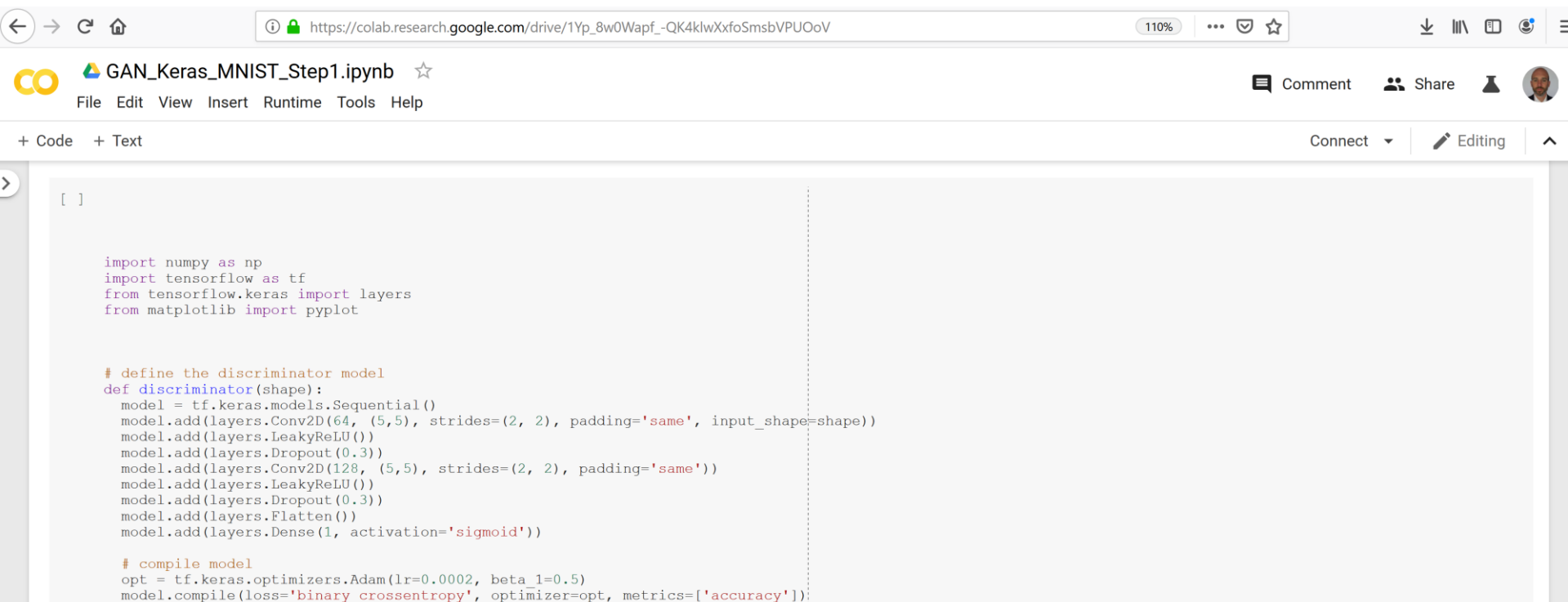
Download

64-Bit Graphical Installer (427 MB)

32-Bit Graphical Installer (361 MB)

Software Options (2. Colab)

- [Google Colab](#) is a free Jupyter notebook environment that requires no setup and runs entirely in the cloud.
- Again comes with essential packages such as Scikit-Learn, NumPy, etc all pre-installed.
- Comes with the option of a free GPU or TPU
- **Drawback**. When you connect to a VM runtime, you have a maximum of 12 hours on the VM. You can easily connect to another VM after the 12 hours expires but you will lose access to an data you had in the previous VM.



The screenshot shows a web browser window with the Google Colab interface. The address bar shows the URL https://colab.research.google.com/drive/1Yp_8w0Wapf_-QK4klwXxfoSmsbVPUOoV. The notebook title is "GAN_Keras_MNIST_Step1.ipynb". The interface includes a menu bar with "File", "Edit", "View", "Insert", "Runtime", "Tools", and "Help". There are also buttons for "Comment", "Share", and a user profile icon. The notebook content shows a Jupyter cell with the following Python code:

```
[ ]

import numpy as np
import tensorflow as tf
from tensorflow.keras import layers
from matplotlib import pyplot

# define the discriminator model
def discriminator(shape):
    model = tf.keras.models.Sequential()
    model.add(layers.Conv2D(64, (5,5), strides=(2, 2), padding='same', input_shape=shape))
    model.add(layers.LeakyReLU())
    model.add(layers.Dropout(0.3))
    model.add(layers.Conv2D(128, (5,5), strides=(2, 2), padding='same'))
    model.add(layers.LeakyReLU())
    model.add(layers.Dropout(0.3))
    model.add(layers.Flatten())
    model.add(layers.Dense(1, activation='sigmoid'))

# compile model
opt = tf.keras.optimizers.Adam(lr=0.0002, beta_1=0.5)
model.compile(loss='binary_crossentropy', optimizer=opt, metrics=['accuracy'])
```


Software Options (2. Colab)

- To use Colab you will need a Google (GMail) account.
- Another aspect of Colab is that you can **mount files from your Google Drive**. This allows you to easily access data from the Colab VM instance.
- I have included a short guide to getting started with Google Colab in the Week 1 unit on Canvas.
 - Describes how to create a Colab Notebook from your Google Drive.
 - Mount a data file
 - Open the data file and perform some basic pre-processing on the data file.
- Please note that Option 1 (downloading and installing Anaconda locally) is preferable.

Machine Learning

1. Machine Learning is an important sub-discipline of AI.
 2. One goal of AI is building programs to perform tasks, which **humans are currently better at**. Machine learning is an avenue that has had success doing exactly that.
- How do you program a computer to:
 - Recognize faces?
 - Identify objects in images
 - Interpret hand written text
 - Interpreting spoken language?
 -

Machine Learning

How do we define Machine Learning?

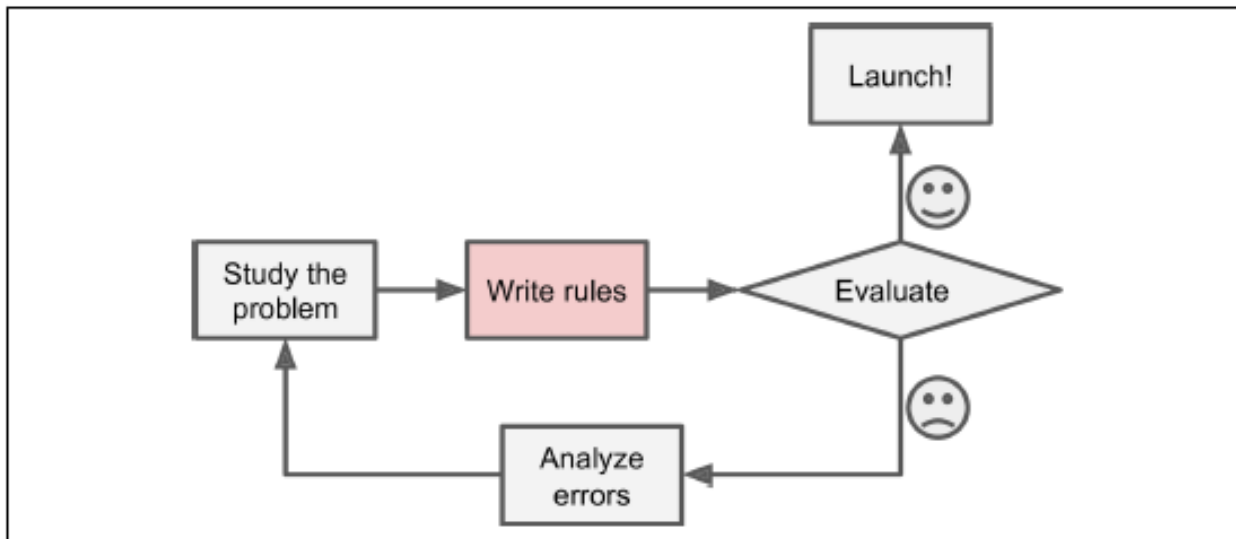
A computer program that will learn from experience E with respect to some task T and some performance measure P , if its performance on T , as measured by P , improves with experience E .

—Tom Mitchell, 1997

Machine learning (ML) provides a means by which programs can infer new knowledge from observational data.

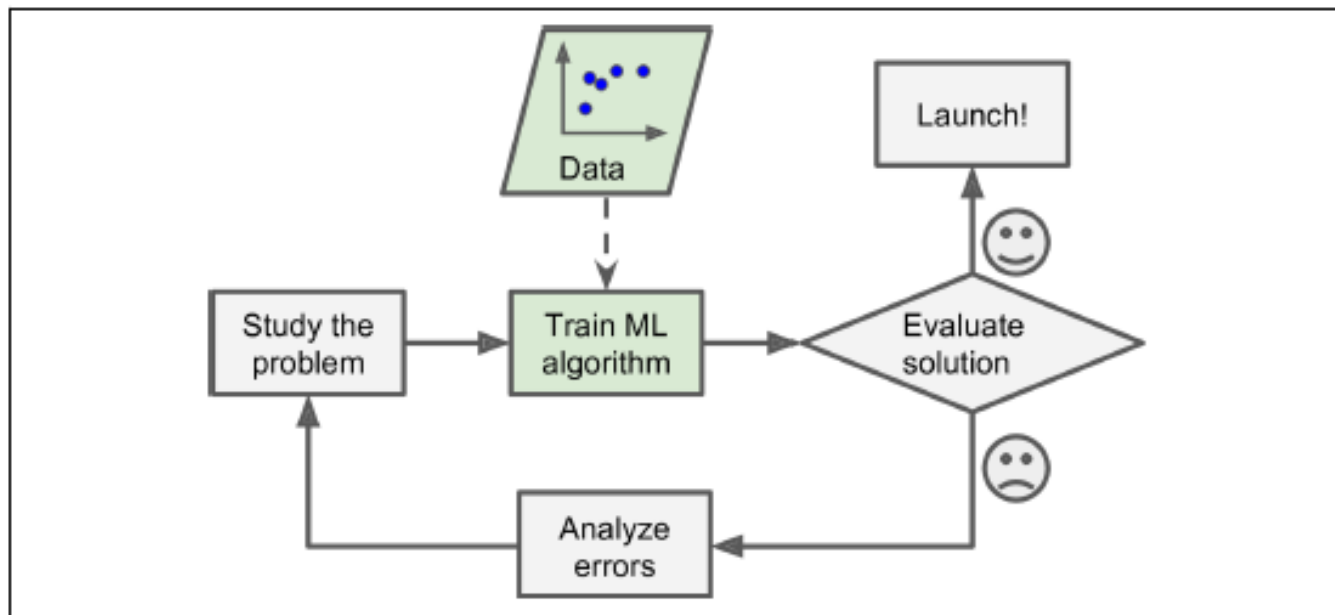
Why Use Machine Learning?

- Let's take a basic problem such as building a **spam filter**.
- We could attempt to build a spam filter using traditional programming techniques.
- First you would look at what spam typically looks like and observe that certain words tend to occur quite frequently in spam.
- You would write a detection algorithm for each of the patterns that you noticed, and your program would flag emails as spam if a number of these patterns are detected.



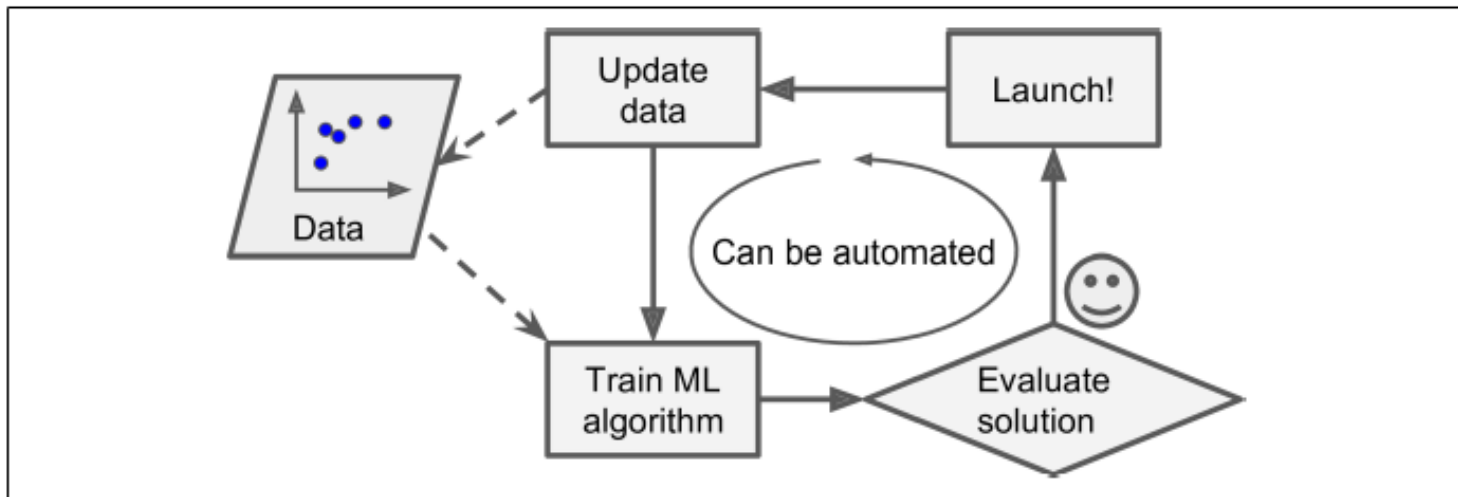
Why Use Machine Learning?

- In contrast, a spam filter based on Machine Learning techniques automatically learns a model (by looking at the words and phrases that are good predictors of spam).
- The program is much shorter, easier to maintain, and most likely more accurate.



Why Use Machine Learning?

- Building such a system using machine learning also means that we can easily **update our model**.
- It is often necessary to retrain models periodically. This is particularly important in scenarios where there is drift in the data over time.



Machine Learning Applications



Spam

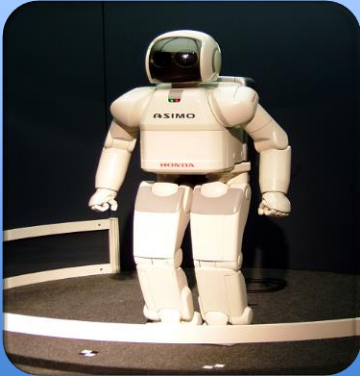
- Host of machine learning algorithms that will learn to classify emails as spam.



Natural Language Processing

- Speech recognition
- Language Translation

Machine Learning Applications



Robotics

- Often machine learning has been studied in conjunction with robotics
- Recognition of objects , navigation of spaces, etc.



Commercial/Finance

- Applications include trading agents that interact with the bond, stock or commodity markets
- Sentiment Analysis
- Forecasting and Prediction

Machine Learning Applications



Navigation

- Research in self-driving cars goes back to early 1990's.
- From Alvin and Stanley (212 km course, 2005) to Google's Self Driving Car.



Recommender Systems

- Netflix, Amazon, Google all use recommender systems
- Collaborative and Content Filtering

Machine Learning Applications



Games

- Machine Learning has proved successful in its application to gaming from Arthur Samuel's checkers program to IBMs Watson and Alpha Go.

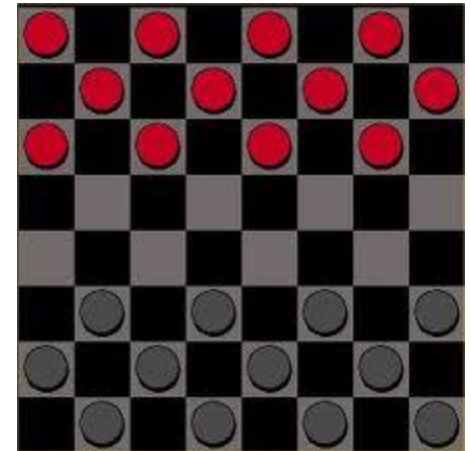


Medicine

- Medical applications can provide decision support systems for assisting in the diagnosis of patients or identification of particular illnesses.

ML in Games

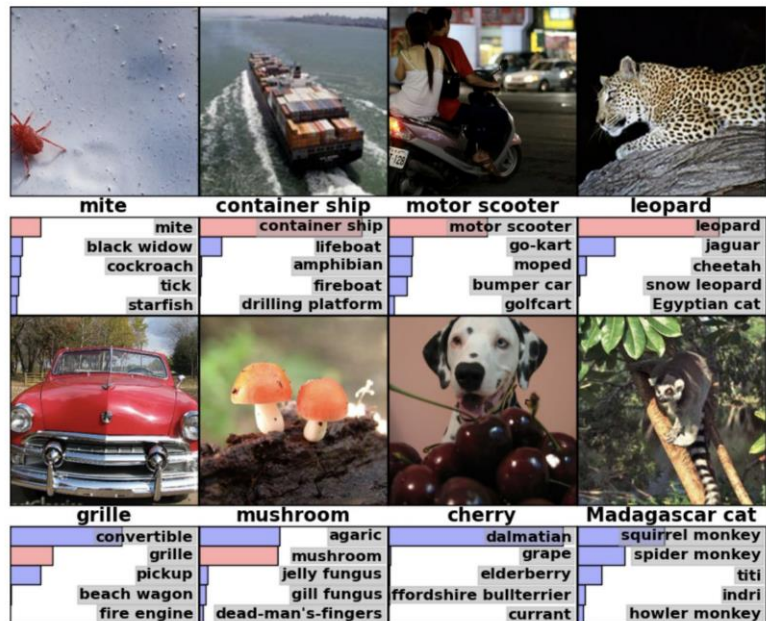
- In the 1960s Arthur Samuel developed a checker player that learned from experience. This work was one of the earliest and most influential examples of machine learning
- Had the program play thousands of games against itself and over time it began to recognise patterns that lead to wins and patterns that lead to losses
- Could play draughts better than Samuel



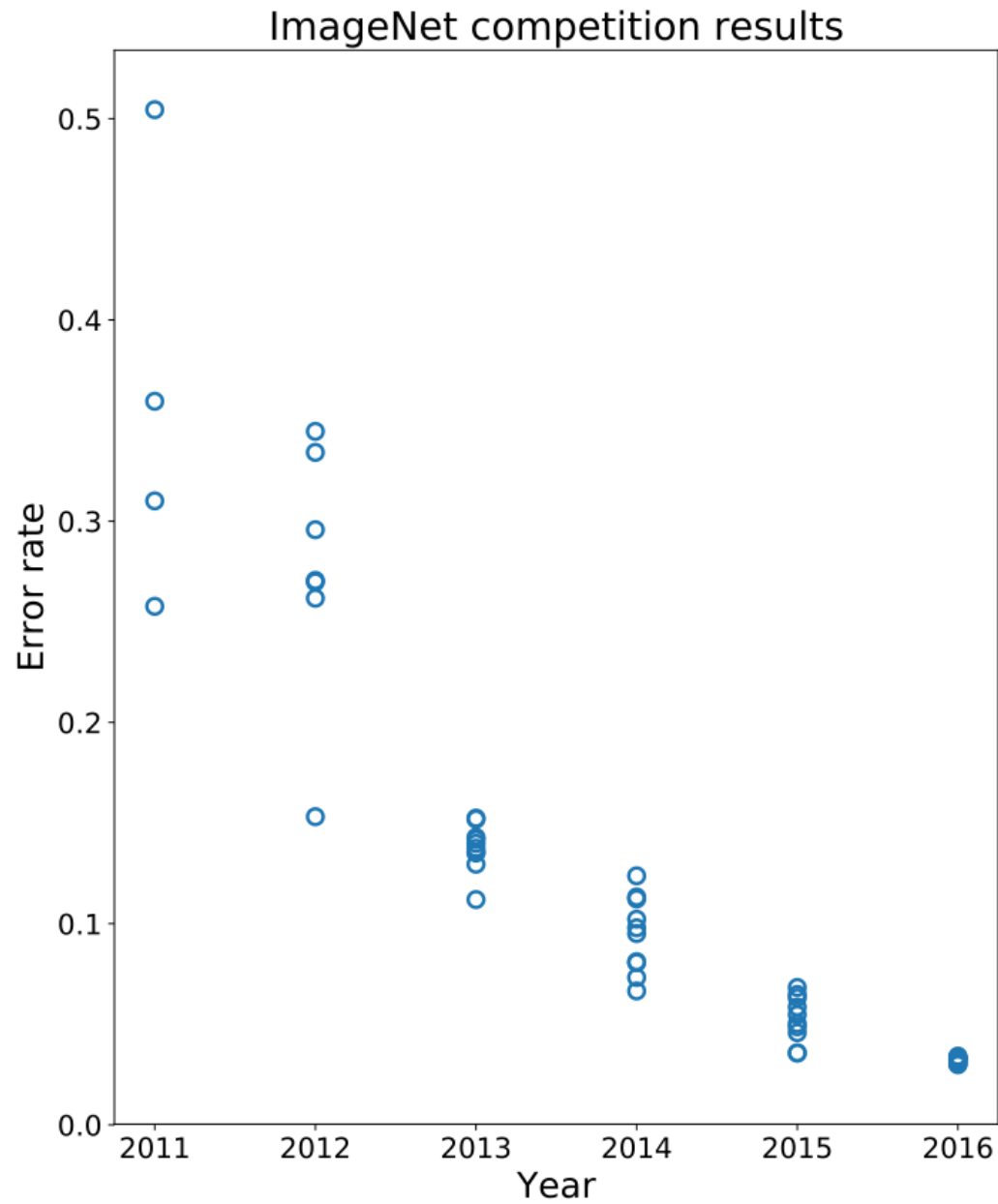
ImageNet Challenge

IMAGENET

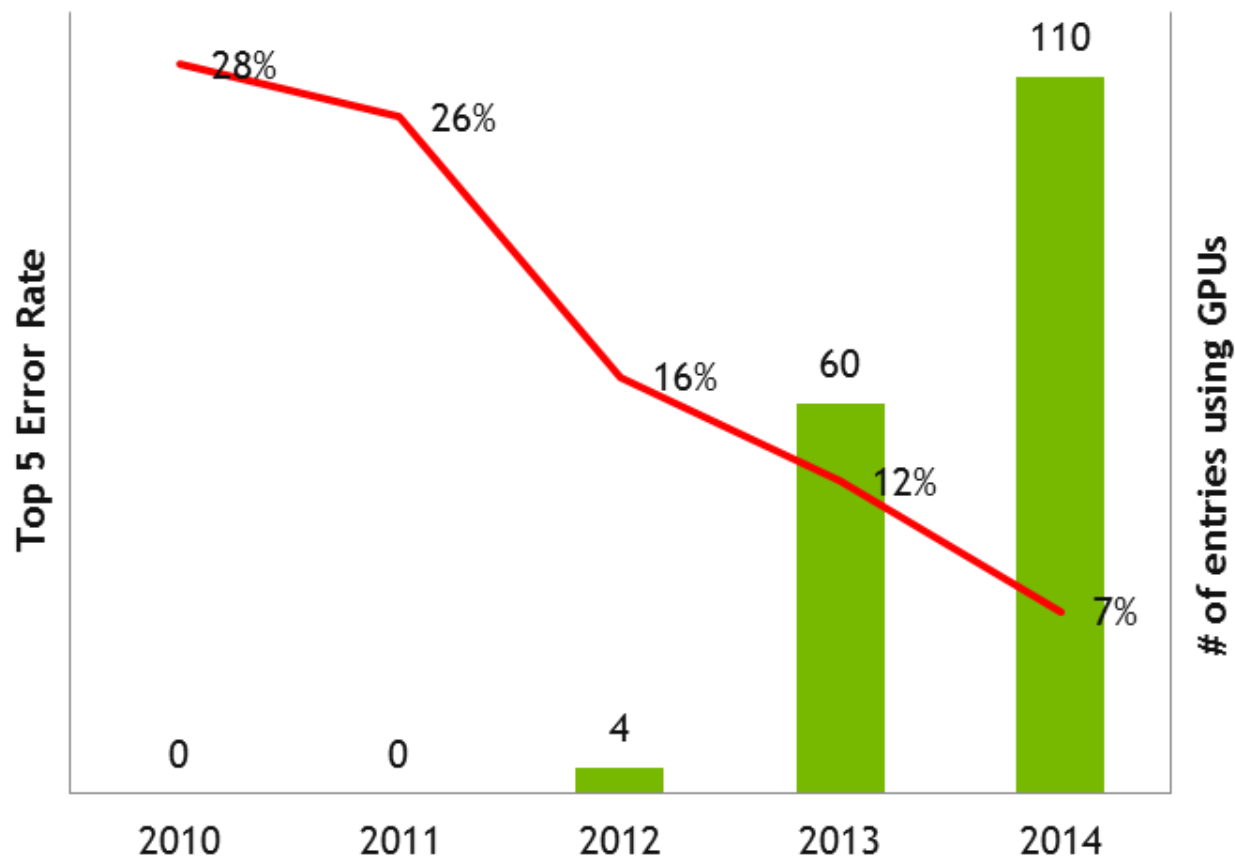
- 1,000 object classes (categories).
- Images:
 - 1.2 M train
 - 100k test.



- Since 2010, the ImageNet project runs an annual software contest, the ImageNet Large Scale Visual Recognition Challenge (ILSVRC), where software programs compete to correctly classify and detect objects and scenes.
- In 2012 a submission called **AlexNet** achieved a **top-5 error of 16%**, more than 10.8 percentage points ahead of the runner up.
- As of 2018 it has been cited over 25,000 times.



IMAGENET



ImageNet Challenge

- GoogLeNet (also called Inception V1) won the ImageNet competition in 2014.
- ResNet won the ILSVRC 2015 competition with an incredible 3.6% error rate (human performance is 5-10%).
- In 2017, 29 of 38 competing teams got less than 5% wrong.

