# Software for Machine Learning

Mechatronics seminar series, University of Newcastle.

Johan Dahlin

uni@johandahlin.com

## Who am I?

2011: MSc Engineering Physics (stats and financial math).

2011: BSc Economics (finance and econometrics).

2016: PhD Automatic Control (Bayesian comp. inference).

2016 Sectra: medical images with deep learning.

2017 LiU: Active learning for searching disaster areas.

2017-2018 UoN: Bayesian system identification.

# The machine learning hype in popular media [1/4]

BBC News Sport Weather More Search

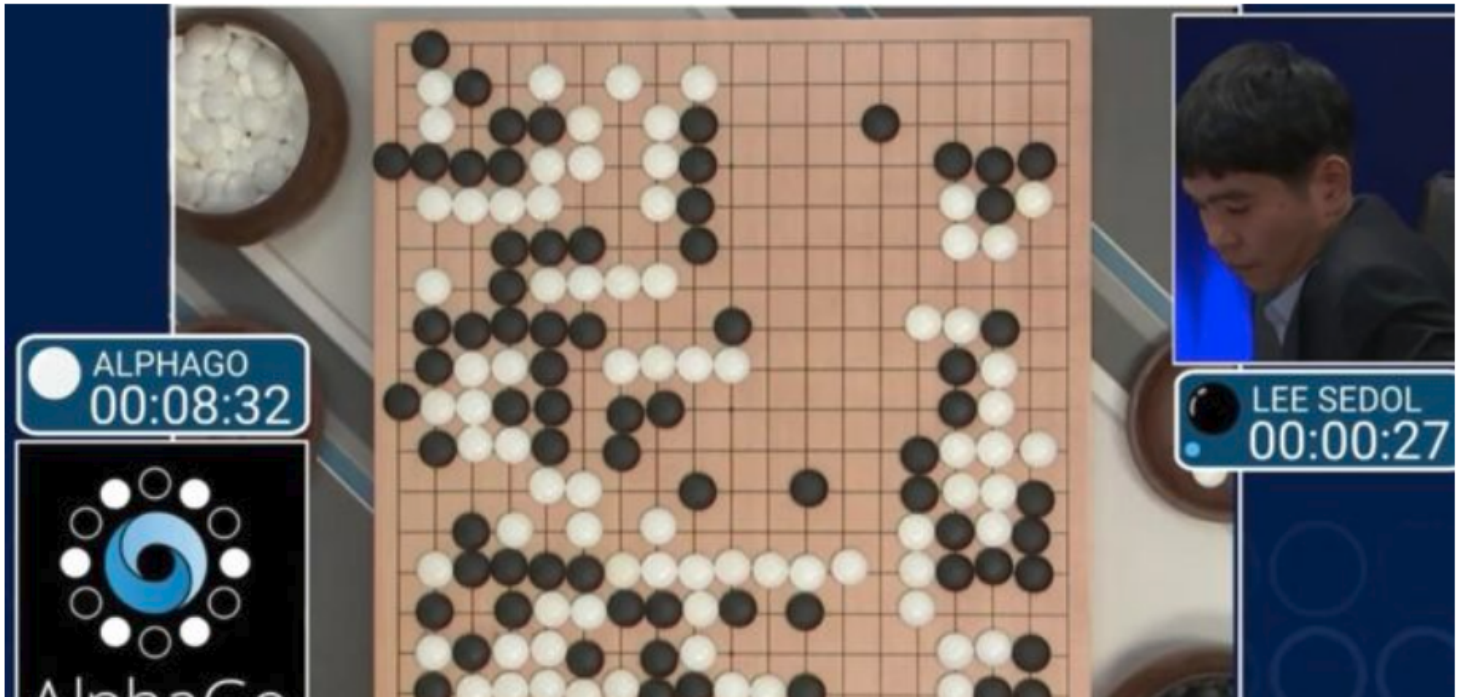
NEWS

Home UK World Business Politics Tech Science HealthMore

Technology

## Artificial intelligence: Google's AlphaGo beats Go master Lee Se-dol

12 March 2016 Share



# The machine learning hype in popular media [2/4]

MIT Technology Review

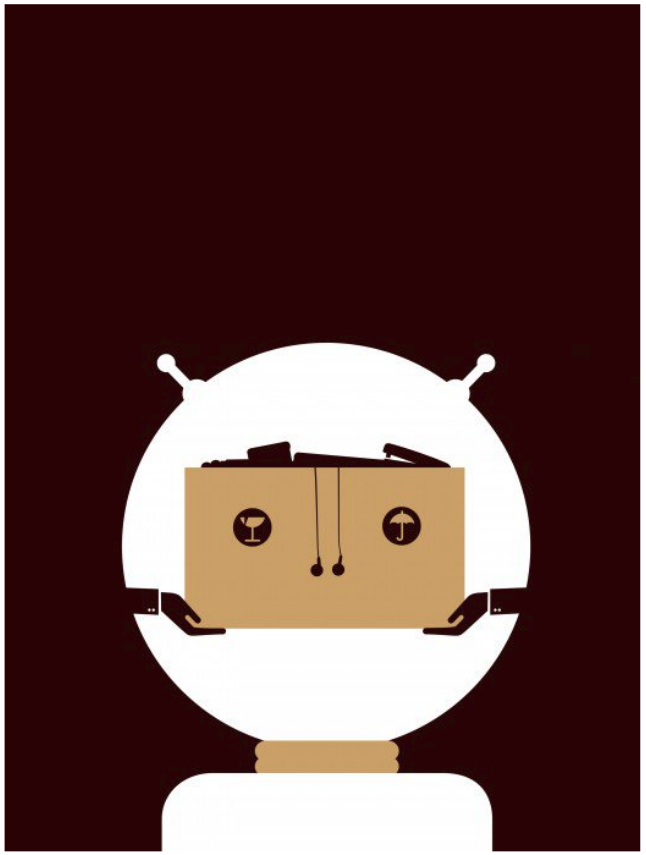
Log in / Create an account Search

Topics+ The Download Magazine Events More+

Subscribe

Business Impact

## How Technology Is Destroying Jobs



Automation is reducing the need for people in many jobs. Are we facing a future of stagnant income and worsening inequality?

by David Rotman June 12, 2013

# The machine learning hype in popular media [3/4]

IEEE SPECTRUM

Follow on:

Engineering Topics | Special Reports | Blogs | Multimedia | The Magazine | Professional Resou

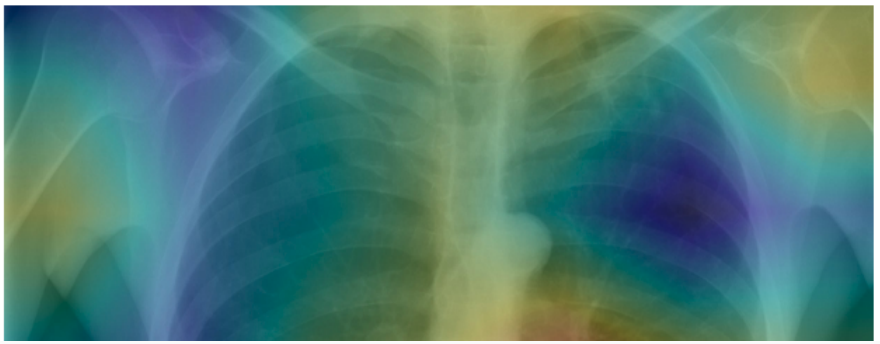
The Human OS | Biomedical | Diagnostics

17 Nov 2017 | 18:30 GMT

## Stanford Algorithm Can Diagnose Pneumonia Better Than Radiologists

It took Stanford AI researchers just a month to beat radiologists at the pneumonia game

By **Tekla S. Perry**



# The machine learning hype in academia [1/3]

REVIEW

doi:10.1038/nature14541

## Probabilistic machine learning and artificial intelligence

Zoubin Ghahramani[1]

How can a machine learn from experience? Probabilistic modelling provides a framework for understanding what learning is, and has therefore emerged as one of the principal theoretical and practical approaches for designing machines that learn from data acquired through experience. The probabilistic framework, which describes how to represent and manipulate uncertainty about models and predictions, has a central role in scientific data analysis, machine learning, robotics, cognitive science and artificial intelligence. This Review provides an introduction to this framework, and discusses some of the state-of-the-art advances in the field, namely, probabilistic programming, Bayesian optimization, data compression and automatic model discovery.

# The machine learning hype in academia [2/3]

LETTER

doi:10.1038/nature14236

## Human-level control through deep reinforcement learning

Volodymyr Mnih[1]*, Koray Kavukcuoglu[1]*, David Silver[1]*, Andrei A. Rusu[1], Joel Veness[1], Marc G. Bellemare[1], Alex Graves[1], Martin Riedmiller[1], Andreas K. Fidjeland[1], Georg Ostrovski[1], Stig Petersen[1], Charles Beattie[1], Amir Sadik[1], Ioannis Antonoglou[1], Helen King[1], Dharshan Kumaran[1], Daan Wierstra[1], Shane Legg[1] & Demis Hassabis[1]

**The theory of reinforcement learning provides a normative account[1], deeply rooted in psychological[2] and neuroscientific[3] perspectives on animal behaviour, of how agents may optimize their control of an environment. To use reinforcement learning successfully in situations approaching real-world complexity, however, agents are confronted**

agent is to select actions in a fashion that maximizes cumulative future reward. More formally, we use a deep convolutional neural network to approximate the optimal action-value function

$$Q^*(s,a) = \max_{\pi} \mathbb{E}\left[r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \ldots | s_t = s,\, a_t = a,\, \pi\right],$$

# The machine learning hype in academia [3/3]

ARTICLE

doi:10.1038/nature16961

## Mastering the game of Go with deep neural networks and tree search

David Silver[1]*, Aja Huang[1]*, Chris J. Maddison[1], Arthur Guez[1], Laurent Sifre[1], George van den Driessche[1], Julian Schrittwieser[1], Ioannis Antonoglou[1], Veda Panneershelvam[1], Marc Lanctot[1], Sander Dieleman[1], Dominik Grewe[1], John Nham[2], Nal Kalchbrenner[1], Ilya Sutskever[2], Timothy Lillicrap[1], Madeleine Leach[1], Koray Kavukcuoglu[1], Thore Graepel[1] & Demis Hassabis[1]

**The game of Go has long been viewed as the most challenging of classic games for artificial intelligence owing to its enormous search space and the difficulty of evaluating board positions and moves. Here we introduce a new approach to computer Go that uses 'value networks' to evaluate board positions and 'policy networks' to select moves. These deep neural networks are trained by a novel combination of supervised learning from human expert games, and reinforcement learning from games of self-play. Without any lookahead search, the neural networks play Go at the level of state-of-the-art Monte Carlo tree search programs that simulate thousands of random games of self-play. We also introduce a new search algorithm that combines Monte Carlo simulation with value and policy networks. Using this search algorithm, our program AlphaGo achieved a 99.8% winning rate against other Go programs, and defeated the human European Go champion by 5 games to 0. This is the first time that a computer program has defeated a human professional player in the full-sized game of Go, a feat previously thought to be at least a decade away.**

# Some trends in programming languages

**Worldwide**, Jun 2018 compared to a year ago:

| Rank | Change | Language | Share | Trend |
|---|---|---|---|---|
| 1 | ↑ | Python | 23.04 % | +5.2 % |
| 2 | ↓ | Java | 22.45 % | -0.6 % |
| 3 | ↑↑ | Javascript | 8.6 % | +0.3 % |
| 4 | ↓ | PHP | 8.21 % | -1.6 % |
| 5 | ↓ | C# | 8.01 % | -0.4 % |
| 6 | | C/C++ | 6.15 % | -1.1 % |
| 7 | ↑ | R | 4.14 % | +0.1 % |
| 8 | ↓ | Objective-C | 3.46 % | -1.0 % |
| 9 | | Swift | 2.75 % | -0.8 % |
| 10 | | Matlab | 2.15 % | -0.4 % |

## What are we going to do?

- Discuss some modern software tools for machine learning.
- Write code for deep learning.

## Why are we doing this?

- Give a small introduction to machine learning.
- Give you the tools to apply these methods in your own work.

## How will we do this?

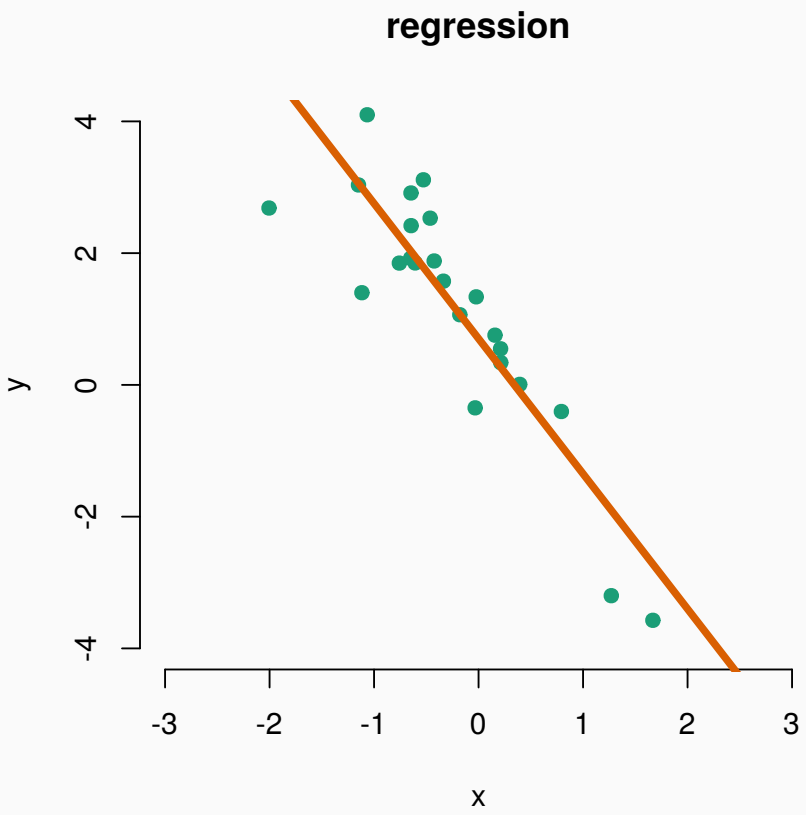
- Discuss the three machine learning paradigms.
- Give an overview of Python, R and Git/Docker.

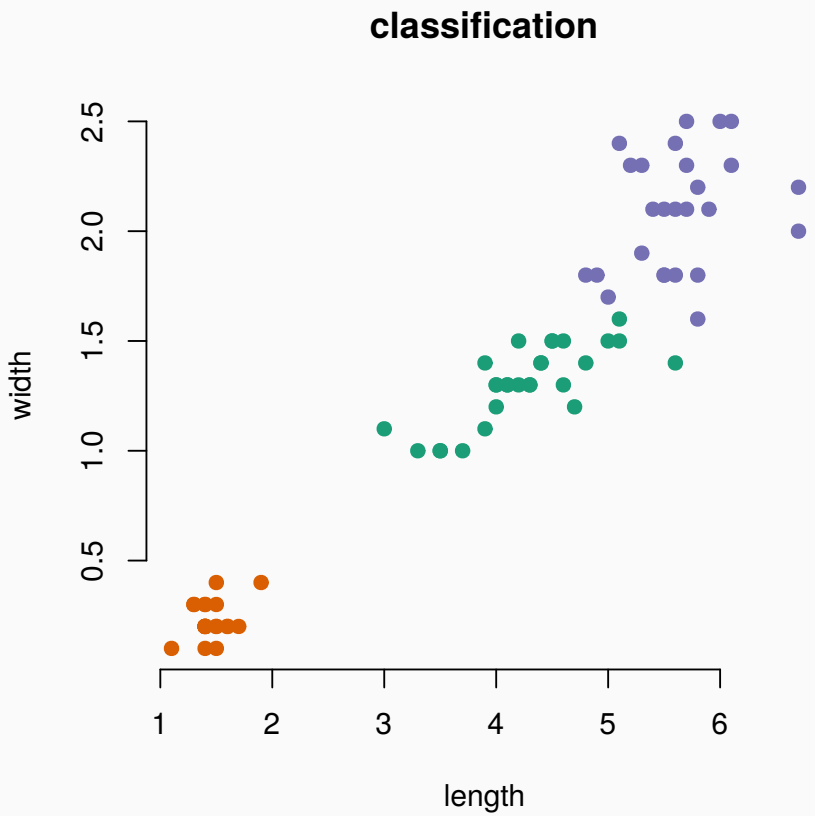
# A short introduction to machine learning

# Three major types of methods

- Supervised learning - labeled data set (y, x).
- Unsupervised learning - unlabeled data set (y).
- Reinforcement learning - environments (s, a, r).

# Supervised learning



# Supervised learning methods

- Linear regression / classification.
- Support vector machines with kernels.
- Ensemble methods using decision trees. Boosting.
- Neural networks.
- Gaussian processes.

# Python

# What is Python?

- Simple programming language to learn.
- Open source and platform independent.
- Very large community.
- Huge number of free libraries available.

`https://www.anaconda.com/download/`

# Packages

Numpy: linear algebra.

Scipy: sparse matrices, optimisation and statistics.

Matplotlib: plotting.

Pandas: data handling.

Scikit-learn: machine learning.

Keras + Tensorflow: deep learning.

pystan: probabilistic programming for Bayesian inference.

# How to run Python code

- Interactive mode using e.g., Spyder or VS Code.
- As a standalone program from the console.
- Writing Jupyter notebooks. `jupyter notebook`

# Python syntax

```
import numpy as np
import matplotlib.pylab as plt

x = np.arange(start=0.0, stop=10.0, step=0.1)
n = len(x)
K = np.zeros((n, n))

for i in range(n):
    for j in range(n):
        K[i, j] = np.exp(-0.5 * (x[i] - x[j])**2)

y = np.random.multivariate_normal(np.zeros(n), K)

plt.plot(x, y)
plt.show()
```

A machine learning use case

# Deep learning for images

# Deep learning and neural networks

Deep learning is a family of supervised machine learning algorithms, which is a type of neural network.

`http://playground.tensorflow.org`

# Tensorflow and Keras


- A framework for deep neural networks made by Google.
- Runs on CPUs, GPUs and in the cloud (TPUs).
- Low-level and efficient library for linear algebra.


- Implements most optimizers and network topologies.
- Weights for pre-trained networks can be downloaded!

# Demostration

Using Python with `keras` and `tensorflow`.

`https://keras.io/`

`https://www.tensorflow.org/`

# Other software tools

# R

- Open source.
- Popular with statisticians and data scientists.
- Efficient work flows for data analysis.
- Similar syntax to Python and MATLAB.
- A lot of packages but a bit fragmented.

`https://www.r-project.org/`

`https://www.rstudio.com/`

# Git(Hub) and Docker

- Version control and collaboration.
- Repository for code and data when running in the cloud.
- Distributing data and software.

```
https://www.github.com/
https://www.docker.com/
```

# Cloud computing

- VMs running Docker containers.
- Data storage.
- Running back-end servers with APIs.
- Clusters of CPUs, GPUs and TPUs.

- Pay as you go.

`https://cloud.google.com/`

`https://aws.amazon.com/`

## What did we do?

- Discussed some modern software tools for machine learning.
- Wrote code for deep learning.

## Why did we do this?

- Give a small introduction to machine learning.
- Give you the tools to apply these methods in your own work.

## What are you going to do now?

- Go home and Google some keywords from this seminar.
- Start learning Python/R, scikit-learn and cloud computing.

# Thank you for listening

Comments, suggestions and/or questions?

Johan Dahlin
uni@johandahlin.com
research.johandahlin.com