MIT 6.148
Smart Web apps using Machine Learning
Hello!
I am Carlos Aguayo

◎ ~13 years at Appian
◎ Director, Software Development
◎ Master's student

https://www.linkedin.com/in/carlosaguayo/
What is Machine Learning?
Machine learning is the subfield of computer science that "gives computers the ability to learn without being explicitly programmed" - Arthur Samuel, 1959
A computer program is said to **learn** from **experience** $E$ with respect to some class of **tasks** $T$ and **performance** measure $P$, if its performance at tasks in $T$, as measured by $P$, improves with experience $E$. - Tom Mitchell
Let's start with a demo!
Gender Recognition by Voice and Speech Analysis

Given an audio, tell if the voice in the audio is male or female.
Acoustic Properties Measured

**meanfreq** mean frequency (in kHz)
**sd** standard deviation of frequency
**median** median frequency (in kHz)
**Q25** first quantile (in kHz)
**Q75** third quantile (in kHz)
**IQR** interquantile range (in kHz)
**skew** skewness
**kurt** kurtosis
**sp.ent** spectral entropy
**sfm** spectral flatness
**mode** mode frequency

**centroid** frequency centroid
**peakf** peak frequency
**meanfun** average of fundamental frequency
**minfun** minimum fundamental frequency
**maxfun** maximum fundamental frequency
**meandom** average of dominant frequency
**mindom** minimum of dominant frequency
**maxdom** maximum of dominant frequency
**dfrange** range of dominant frequency
**modindx** modulation index
How?
How?
How?
What did we do?

3,168 voice samples
What did we do?

3,168 voice samples

Machine Learning Algorithm
What did we do?

3,168 voice samples

Machine Learning Algorithm

$f(x)$
What did we do?

3,168 voice samples → Machine Learning Algorithm → $f(x)$

$f(x)$ → Male → Female
How?
How?
How?

Given an $X$ and $Y$, is this point pink or blue?
How?

Given an X and Y, is this point pink or blue?
How?

Blue!
How?
How?
How?
How?
K-Nearest Neighbors

One of the simplest, yet effective, machine learning algorithms.
How?
Support Vector Machine

Hyperplane that represents the largest separation between classes
How?
Decision Trees

Another simple, and effective, supervised learning algorithm.
1. mode
2. minfun
3. maxdom
4. Q25
5. median
6. meanfun
7. skew
Human vs. Machine

Up to 3 dimensions!

High dimensional space!
Supervised Learning
Supervised learning is the machine learning task of inferring a function from labeled training data.

The training data consist of a set of training examples. Each example is a pair consisting of an input object and an output value. A supervised learning algorithm analyzes the training data and produces an inferred function, which can be used for mapping new examples.
Supervised learning is the machine learning task of inferring a function from labeled training data. The training data consist of a set of training examples. Each example is a pair consisting of an input object and an output value. A supervised learning algorithm analyzes the training data and produces an inferred function, which can be used for mapping new examples.
Supervised learning is the machine learning task of inferring a function from labeled training data. The training data consist of a set of training examples. Each example is a pair consisting of an input object and an output value. A supervised learning algorithm analyzes the training data and produces an inferred function, which can be used for mapping new examples.
Supervised learning is the machine learning task of inferring a function from labeled training data. The training data consist of a set of training examples. Each example is a pair consisting of an input object and an output value.

A supervised learning algorithm analyzes the training data and produces an inferred function, which can be used for mapping new examples.
Neural Networks
Will you go out to the party tonight?
Will you go out to the party tonight?

Can I wake up late tomorrow?
Can I wake up late tomorrow?

Will the person that I like be there?

Will you go out to the party tonight?
Will you go out to the party tonight?

- Can I wake up late tomorrow?
- Will the person that I like be there?
- **Will my friends be there?**
Will you go out to the party tonight?

- Can I wake up late tomorrow?
- Will the person that I like be there?
- Will my friends be there?
- **Do I have any other plans tonight?**
- Have I gone to that party before?
It’s just a weighted decision. If the output is equal or larger than 10, I’ll be there!
Supervised Learning

Yes
Crush?

No
Friend?

No
Late?

No
No Plans?

No
New?

Sum

It's just a weighted decision. If the output is equal or larger than 10, I'll be there!

= 10
Yes! I'll be at the party!
It's just a weighted decision. If the output is equal or larger than 10, I'll be there!

No, raincheck.
Supervised Learning

Crush? (No: 10)

Friend? (Yes: 7)

Late? (Yes: 5)

No Plans? (No: 5)

New? (No: 3)

It's just a weighted decision. If the output is equal or larger than 10, I'll be there!

7 + 5 = 12

Yes! I'll be at the party!
Supervised Learning

- Crush? 10
- Friend? 7
- Late? 5
- No Plans? 5
- New? 3

Sum

**Neuron scheme**

- Dendrites
- Synapses
- Axon

*In* → *Out*
Supervised Learning

- Input layer
- Hidden layers
- Output layer
Supervised Learning

input layer

Crush
Friend
Late
Plan
New
Other

hidden layers

output layer

Sum
Crush
Friend
Late
Plan
New
Other
Supervised Learning

input layer

hidden layers

output layer
Deep Learning
Supervised Learning

Convolutional Neural Networks (CNNs)
High Level Summary
High Level Summary

Labeled Data
You get a set of samples, each of them with an answer.
High Level Summary

**Labeled Data**
You get a set of samples, each of them with an answer.

**Model**
Learn a model that can successfully predict the seen and unseen samples.
High Level Summary

**Labeled Data**
You get a set of samples, each of them with an answer.

**Model**
Learn a model that can successfully predict the seen and unseen samples.

**Predict**
A number, face, voice, price of a house, stock, etc.
High Level Summary

**Labeled Data**
You get a set of samples, each of them with an answer.

**Model**
Learn a model that can successfully predict the seen and unseen samples.

**Predict**
A number, face, voice, price of a house, stock, etc.
The Future ...
Deep Blue (1996)

The system derived its playing strength mainly from brute force computing power. Chess knowledge was fine tuned by grandmasters. Studied thousands of games.
Deep Blue (1996)

The system derived its playing strength mainly from brute force computing power. Chess knowledge was fine tuned by grandmasters.

Studied thousands of games.
Another way to think of it is to compare Go to chess, which in the '90s was hard enough to imagine AI mastering before IBM came along. After the first two moves of a Chess game, there are 400 possible next moves. In Go, there are close to 130,000.

"The search space in Go is vast... a number greater than there are atoms in the universe," Google wrote in a January blog post about the game.
0 days

AlphaGo Zero has no prior knowledge of the game and only the basic rules as an input.
3 days

AlphaGo Zero surpasses the abilities of AlphaGo Lee, the version that beat world champion Lee Sedol in 4 out of 5 games in 2016.
21 days

AlphaGo Zero reaches the level of AlphaGo Master, the version that defeated 60 top professionals online and world champion Ke Jie in 3 out of 3 games in 2017.
40 days

AlphaGo Zero surpasses all other versions of AlphaGo and, arguably, becomes the best Go player in the world. It does this entirely from self-play, with no human intervention and using no historical data.
40 days

AlphaGo Zero surpasses all other versions of AlphaGo and, arguably, becomes the best Go player in the world. It does this entirely from self-play, with no human intervention and using no historical data.
Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm

David Silver,1* Thomas Hubert,1* Julian Schrittwieser,1*
Ioannis Antonoglou,1 Matthew Lai,1 Arthur Guez,1 Marc Lanctot,1
Laurent Sifre,1 Dharshan Kumaran,1 Thore Graepel,1
Timothy Lillicrap,1 Karen Simonyan,1 Demis Hassabis1

1DeepMind, 6 Pancras Square, London N1C 4AG.
*These authors contributed equally to this work.

Abstract

The game of chess is the most widely-studied domain in the history of artificial intelligence. The strongest programs are based on a combination of sophisticated search techniques, domain-specific adaptations, and handcrafted evaluation functions that have been refined by human experts over several decades. In contrast, the AlphaGo Zero program recently achieved superhuman performance in the game of Go, by tabula rasa reinforcement learning from games of self-play. In this paper, we generalise this approach into a single AlphaZero algorithm that can achieve, tabula rasa, superhuman performance in many challenging domains. Starting from random play, and given no domain knowledge except the game rules, AlphaZero achieved within 24 hours a superhuman level of play in the games of chess and shogi (Japanese chess) as well as Go, and convincingly defeated a world-champion program in each case.

The study of computer chess is as old as computer science itself. Babbage, Turing, Shannon, and von Neumann devised hardware, algorithms and theory to analyse and play the game.
Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm

David Silver,1* Thomas Hubert,1* Julian Schrittwieser,1*
Ioannis Antonoglou,1 Matthew Lai,1 Arthur Guez,1 Marc Lanctot,1
Laurent Sifre,1 Dharshan Kumaran,1 Thore Graepel,1
Timothy Lillicrap,1 Karen Simonyan,1 Demis Hassabis1

1DeepMind, 6 Pancras Square, London N1C 4AG.
*These authors contributed equally to this work.

Abstract

The game of chess is the most widely-studied domain in the history of artificial intelligence. The strongest programs are based on a combination of sophisticated search techniques, domain-specific adaptations, and handcrafted evaluation functions that have been refined by human experts over several decades. In contrast, the AlphaGo Zero program recently achieved superhuman performance in the game of Go, by tabula rasa reinforcement learning from games of self-play. In this paper, we generalise this approach into a single AlphaZero algorithm that can achieve, tabula rasa, superhuman performance in many challenging domains. Starting from random play, and given no domain knowledge except the game rules, AlphaZero achieved within 24 hours a superhuman level of play in the games of chess and shogi (Japanese chess) as well as Go, and convincingly defeated a world-champion program in each case.

The study of computer chess is as old as computer science itself. Babbage, Turing, Shannon, and von Neumann devised hardware, algorithms and theory to analyse and play the game.
Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm

David Silver, 1* Thomas Hubert, 1* Julian Schrittwieser, 1*
Ioannis Antonoglou, 1 Matthew Lai, 1 Arthur Guez, 1 Marc Lanctot, 1
Laurent Sifre, 1 Dharshan Kumaran, 1 Thore Graepel, 1
Timothy Lillicrap, 1 Karen Simonyan, 1 Demis Hassabis 1

1DeepMind, 6 Pancras Square, London N1C 4AG.

Starting from random play, and given no domain knowledge except the game rules, AlphaZero achieved within 24 hours a superhuman level of play in the games of chess and shogi (Japanese chess) as well as Go, and convincingly defeated a world-champion program in each case.

The study of computer chess is as old as computer science itself. Babbage, Turing, Shannon, and von Neumann devised hardware, algorithms and theory to analyse and play the game.
How?
How?
How?
How?
Reinforcement Learning
Elements of Reinforcement Learning

**States**
The agent is in a given state at all times.
## Elements of Reinforcement Learning

<table>
<thead>
<tr>
<th>States</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>The agent is in a given state at all times.</td>
<td>The agent can perform a finite number of actions that will take it into a new state.</td>
</tr>
</tbody>
</table>
# Elements of Reinforcement Learning

<table>
<thead>
<tr>
<th><strong>States</strong></th>
<th>The agent is in a given state at all times.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Actions</strong></td>
<td>The agent can perform a finite number of actions that will take it into a new state.</td>
</tr>
<tr>
<td><strong>Rewards</strong></td>
<td>The agent is awarded a reward for each state that it is in. Typically an integer number.</td>
</tr>
</tbody>
</table>
Elements of Reinforcement Learning

**States**
The agent is in a given state at all times.

**Actions**
The agent can perform a finite number of actions that will take it into a new state.

**Rewards**
The agent is awarded a reward for each state that it is in. Typically an integer number.

![Diagram showing the transition from Hungry to Not Hungry after eating.](image-url)
Elements of Reinforcement Learning

**States**
The agent is in a given state at all times.

**Actions**
The agent can perform a finite number of actions that will take it into a new state.

**Rewards**
The agent is awarded a reward for each state that it is in. Typically an integer number.
Elements of Reinforcement Learning

**Objective**

The agent goal is to maximize the reward.
Elements of Reinforcement Learning

**Objective**
The agent goal is to maximize the reward.

**Policy**
A policy states the action to take at each possible state.
## Elements of Reinforcement Learning

<table>
<thead>
<tr>
<th>Objective</th>
<th>Policy</th>
<th>Optimal Policy</th>
</tr>
</thead>
<tbody>
<tr>
<td>The agent goal is to maximize the reward.</td>
<td>A policy states the action to take at each possible state.</td>
<td>Maximizes the long time expected reward</td>
</tr>
</tbody>
</table>
**World** - 3 by 3 grid

**Actions** - Up, Down, Left, Right

**Rewards** - All states have a -1 with the exception of top right

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>-1</td>
<td>-1</td>
<td>+100</td>
</tr>
<tr>
<td>-1</td>
<td>-1</td>
<td>-1</td>
</tr>
<tr>
<td>-1</td>
<td>-1</td>
<td>-1</td>
</tr>
</tbody>
</table>
**World** - 3 by 3 grid

**Actions** - Up, Down, Left, Right

**Rewards** - All states have a -1 with the exception of top right

What action should we take if we are in this state?

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>-1</td>
<td>-1</td>
<td>+100</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-1</td>
<td>-1</td>
<td>-1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-1</td>
<td>-1</td>
<td>-1</td>
</tr>
</tbody>
</table>
Can we teach a Taxi to pick up a passenger and drive to destination?
**World** - 5 by 5 grid, 4 designated locations

**Actions** - Up, Down, Left, Right, Pickup, Dropoff

**Rewards** - All states have a -1 with the exception of being at the destination and dropping passenger which has +20
**World** - 5 by 5 grid, 4 designated locations

**Actions** - Up, Down, Left, Right, Pickup, Dropoff

**Rewards** - All states have a -1 with the exception of being at the destination and dropping passenger which has +20

Taxi is on this square with the passenger, dropoff location is "G". Which action should it take?
**World** - 5 by 5 grid, 4 designated locations

**Actions** - Up, Down, Left, Right, Pickup, Dropoff

**Rewards** - All states have a -1 with the exception of being at the destination and dropping passenger which has +20

Taxi is on this square with the passenger, dropoff location is "G". Which action should it take? dropoff
World - 5 by 5 grid, 4 designated locations
Actions - Up, Down, Left, Right, Pickup, Dropoff
Rewards - All states have a -1 with the exception of being at the destination and dropping passenger which has +20

Taxi is on this square with the passenger, dropoff location is "G". Which action should it take?
**World** - 5 by 5 grid, 4 designated locations

**Actions** - Up, Down, Left, Right, Pickup, Dropoff

**Rewards** - All states have a -1 with the exception of being at the destination and dropping passenger which has +20

---

Taxi is on this square with the passenger, dropoff location is "G". Which action should it take?

right
Reinforcement Learning

**World** - 5 by 5 grid, 4 designated locations

**Actions** - Up, Down, Left, Right, Pickup, Dropoff

**Rewards** - All states have a -1 with the exception of being at the destination and dropping passenger which has +20

How many states can we possibly have?
Reinforcement Learning

World - 5 by 5 grid, 4 designated locations

Actions - Up, Down, Left, Right, Pickup, Dropoff

Rewards - All states have a -1 with the exception of being at the destination and dropping passenger which has +20

How many states can we have?

5x5 grid = 25
Passenger can be at either of 4 locations or on board = 5
Destination = 4

25 * 5 * 4 = 500 states
Reinforcement Learning

**World** - 5 by 5 grid, 4 designated locations

**Actions** - Up, Down, Left, Right, Pickup, Dropoff

**Rewards** - All states have a -1 with the exception of being at the destination and dropping passenger which has +20

What if we create a table and learn what action to take at each state?
World - 5 by 5 grid, 4 designated locations

Actions - Up, Down, Left, Right, Pickup, Dropoff

Rewards - All states have a -1 with the exception of being at the destination and dropping passenger which has +20

What if we create a table and learn what action to take at each state?
What if the state space is really big (continuous)?
What if the state space is really big (continuous)?
Balance a pole

Keep a pole standing for as long as possible
Land in the moon!

Fire the spaceship engines to land in the moon!
MIT 6.148
Smart Web apps using Machine Learning
Sentiment & Text Analysis

Extract Information about Text and understand Sentiment

CLOUD NATURAL LANGUAGE
Derive insights from unstructured text using Google machine learning

VIEW DOCUMENTATION
VIEW CONSOLE
Image classification

Detect object within image

- Classification
- Classification + Localization
- Object Detection
- Instance Segmentation

CAT
CAT
CAT, DOG, DUCK
CAT, DOG, DUCK

Single object
Multiple objects
Thank you!

Questions?

Carlos Aguayo

aguayo@appian.com

https://www.linkedin.com/in/carlosaguayo/