



Machine Learning Exposed

*James Weaver
Pivotal*



@JavaFXpert

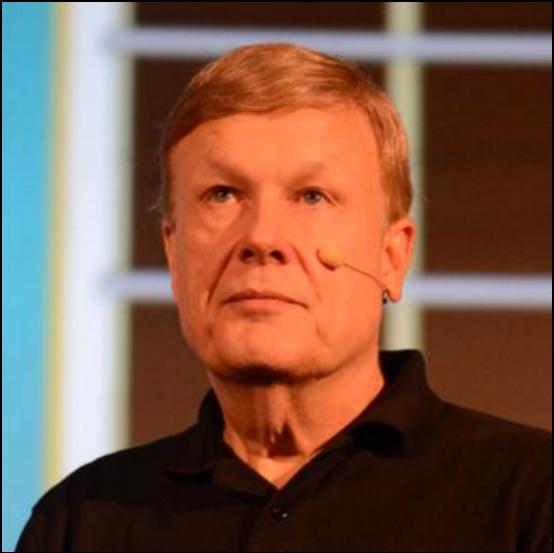


**Click 'Rate Session'
to rate session
and ask questions.**

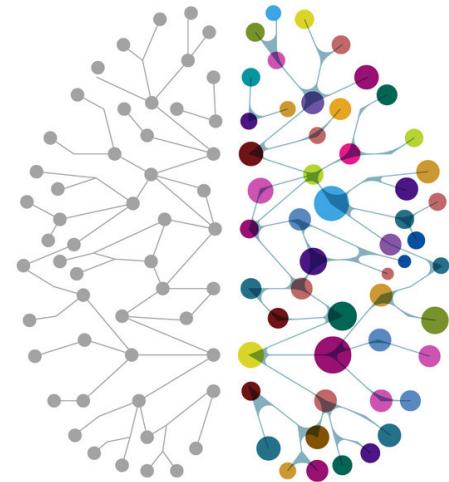
Machine Learning

EXPOSED!

The Fundamentals



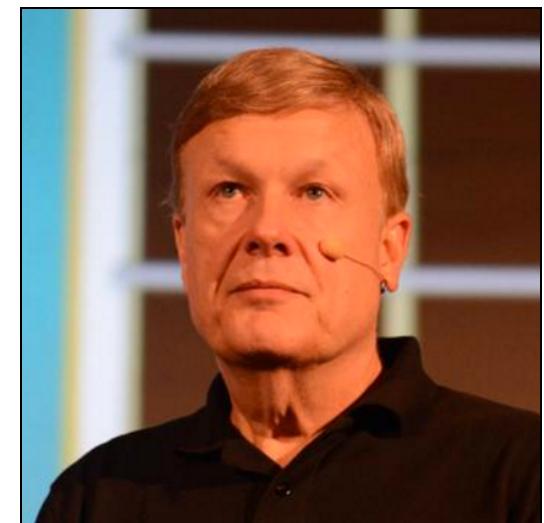
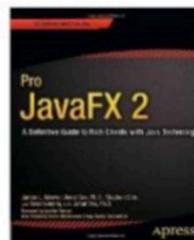
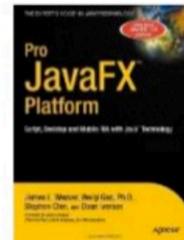
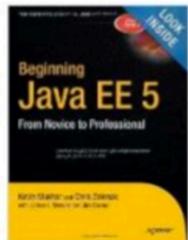
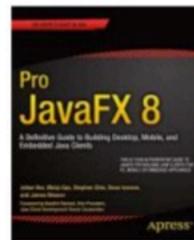
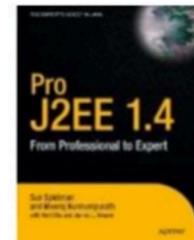
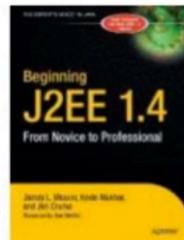
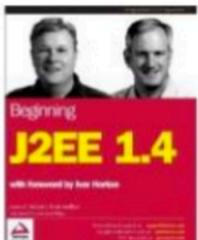
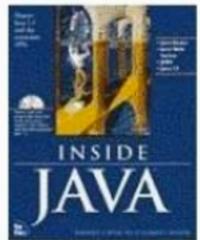
James L. Weaver
Developer Advocate
PivotalTM
jweaver@pivotal.io
JavaFXpert.com
[@JavaFXpert](https://twitter.com/JavaFXpert)



About Presenter James Weaver

Developer Advocate & International Speaker for Pivotal

Author of several Java/JavaFX/RaspPi books



Java Champion, JavaOne Rockstar, plays well with others, etc :-)

NullPointers band rehearsing in Chicago for tonight's party



Chicago: May 30, 31

Spring Days 2017

\$200.00

2 Days, 10 sessions

2 Spring team talks

Live Open Space discussions



SpringOnePlatform.io

SpringOne Platform

REGISTER NOW.

December 4–7, 2017
Moscone West, San Francisco

Pivotal





From [introductory video](#) in Machine Learning course (Stanford University & Coursera) taught by Andrew Ng.



Machine learning is the science of getting computers to learn, without being explicitly programmed.

A woman with long brown hair and glasses, wearing a blue denim shirt over a red and white plaid shirt, is sitting in the driver's seat of a car. She is looking down at her smartphone in her right hand. In her left hand, she holds a brown paper coffee cup with a white lid. The background shows the interior of the car and a blurred green landscape outside, suggesting motion. The lighting is warm and focused on her face and hands.

Self-driving cars

@JavaFXpert

Generating image descriptions



"man in black shirt is playing guitar."



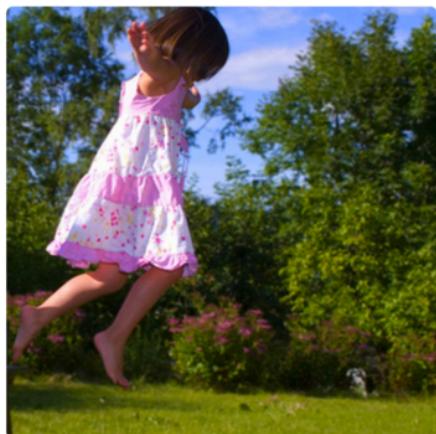
"construction worker in orange safety vest is working on road."



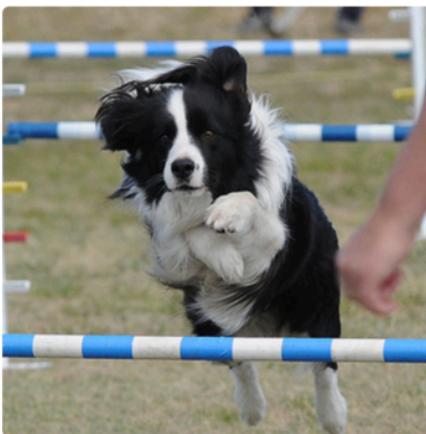
"two young girls are playing with lego toy."



"boy is doing backflip on wakeboard."



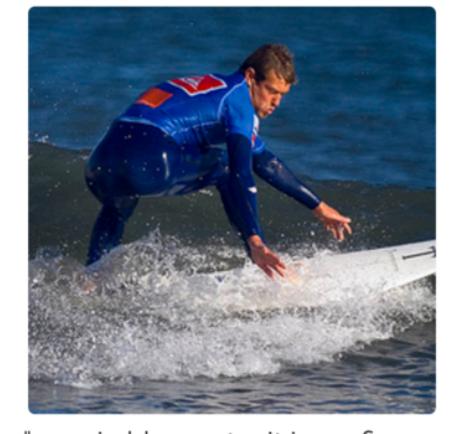
"girl in pink dress is jumping in air."



"black and white dog jumps over bar."



"young girl in pink shirt is swinging on swing."



"man in blue wetsuit is surfing on wave."

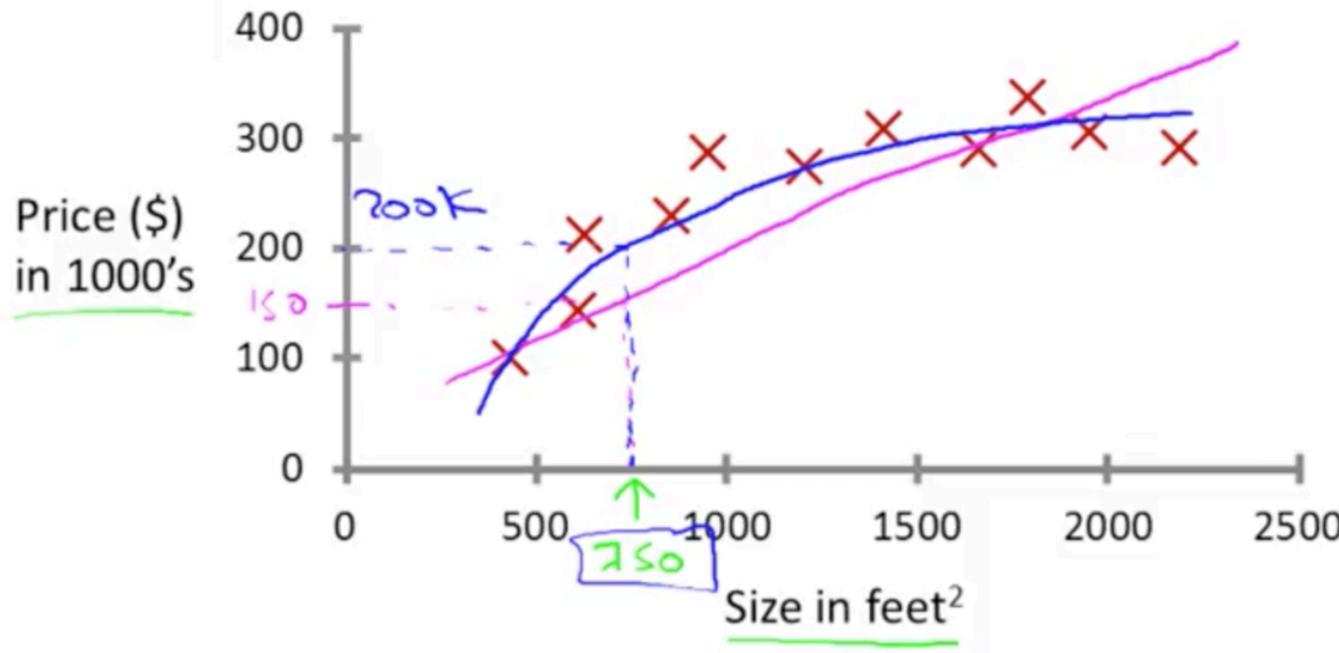
Supervised Learning



Supervised learning regression problem

(from Andrew Ng's Machine Learning course)

Housing price prediction.



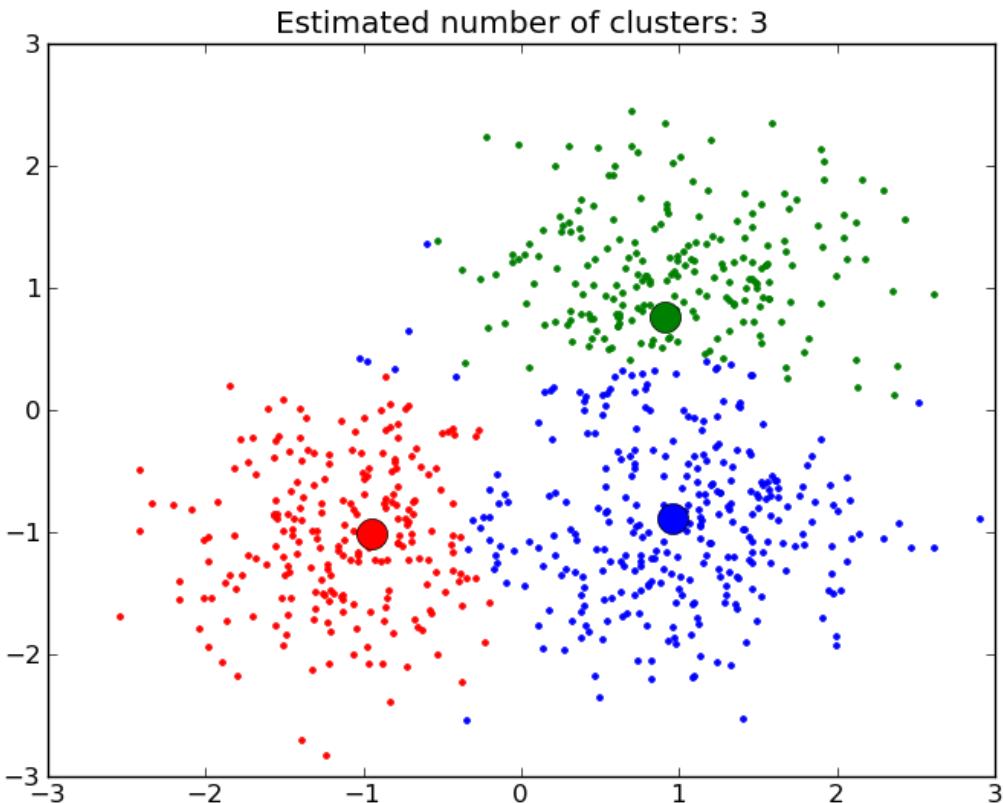
Supervised Learning
‘right answers’ given

Regression: Predict continuous
valued output (price)

Unsupervised Learning



Unsupervised learning finds structure in unlabeled data

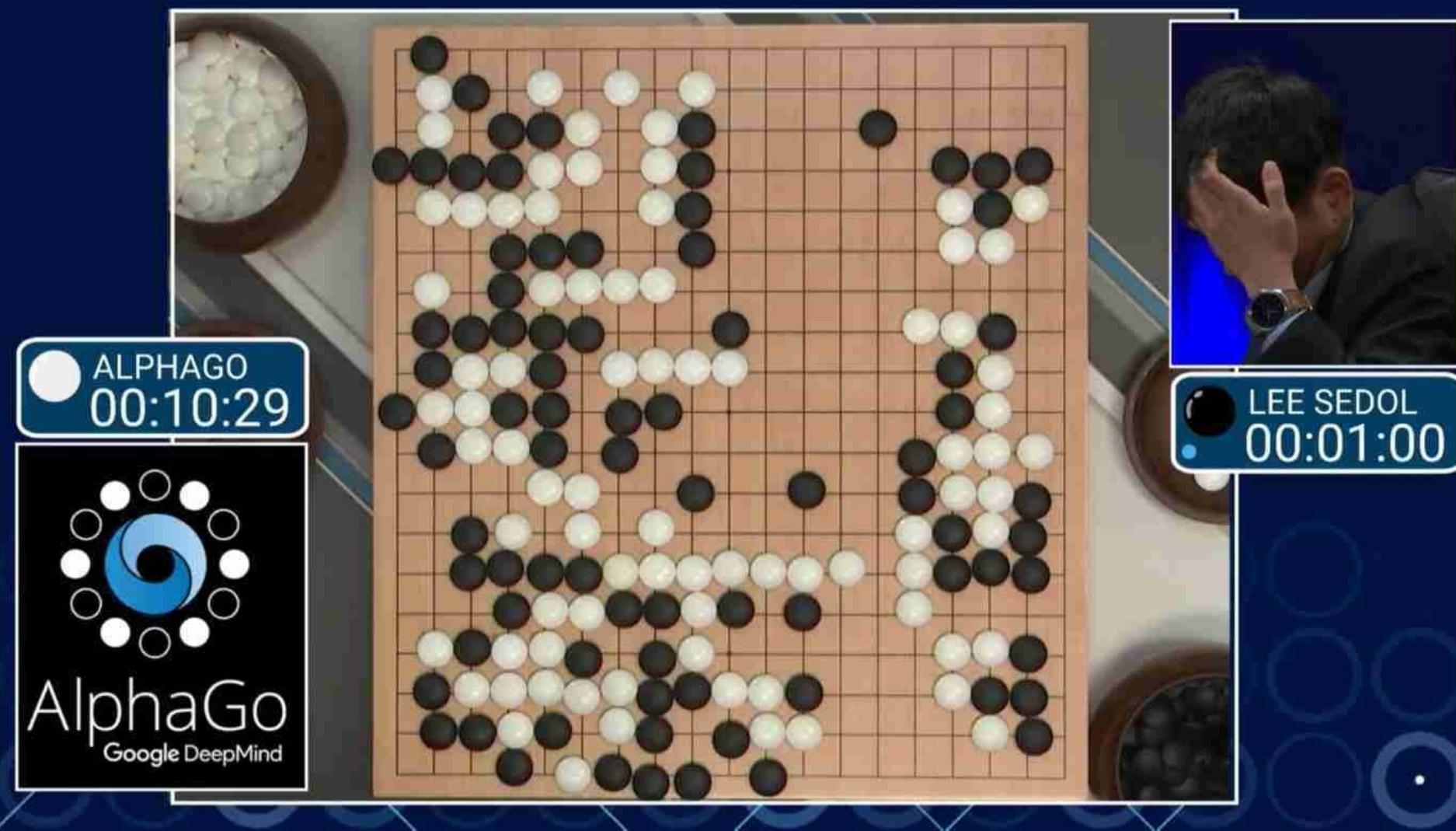


(e.g. market segment discovery,
and social network analysis)

Reinforcement Learning



AlphaGo is a recent *reinforcement learning* success story



Source: <https://gogameguru.com/i/2016/03/AlphaGo-Lee-Sedol-game-3-game-over.jpg>

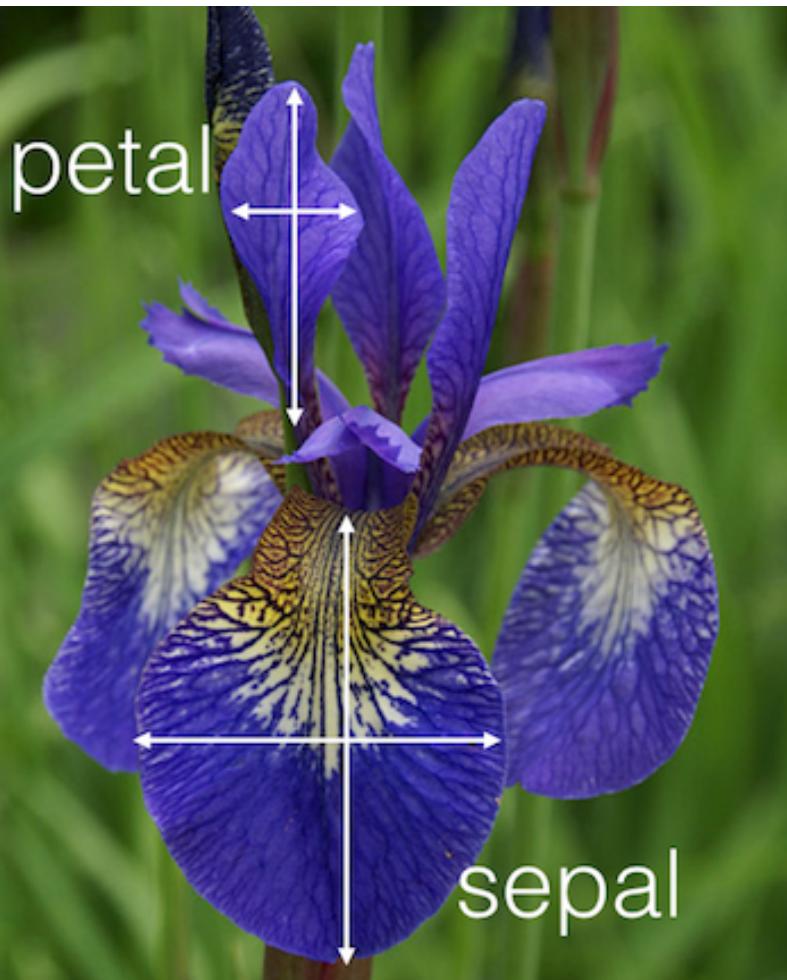
@JavaFXpert

Supervised Learning

(Let's dive in now)



Supervised learning classification problem (using the [Iris flower data set](#))

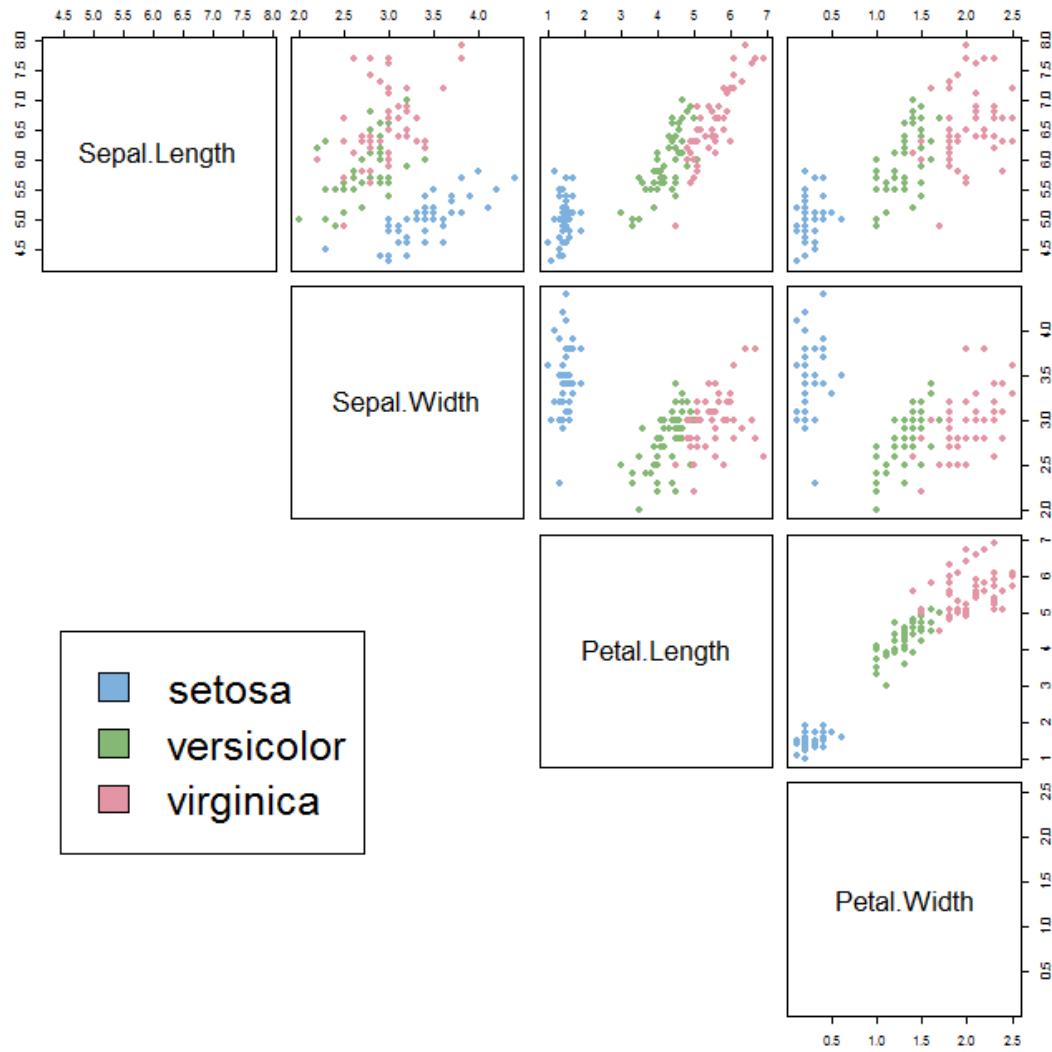


Training / test data

Features

Labels

Sepal length	Sepal width	Petal length	Petal width	Species
5.1	3.5	1.4	0.2	Iris setosa
4.9	3.0	1.4	0.2	Iris setosa
7.0	3.2	4.7	1.4	Iris versicolor
6.4	3.2	4.5	1.5	Iris versicolor
6.3	3.3	6.0	2.5	Iris virginica
5.8	3.3	6.0	2.5	Iris virginica

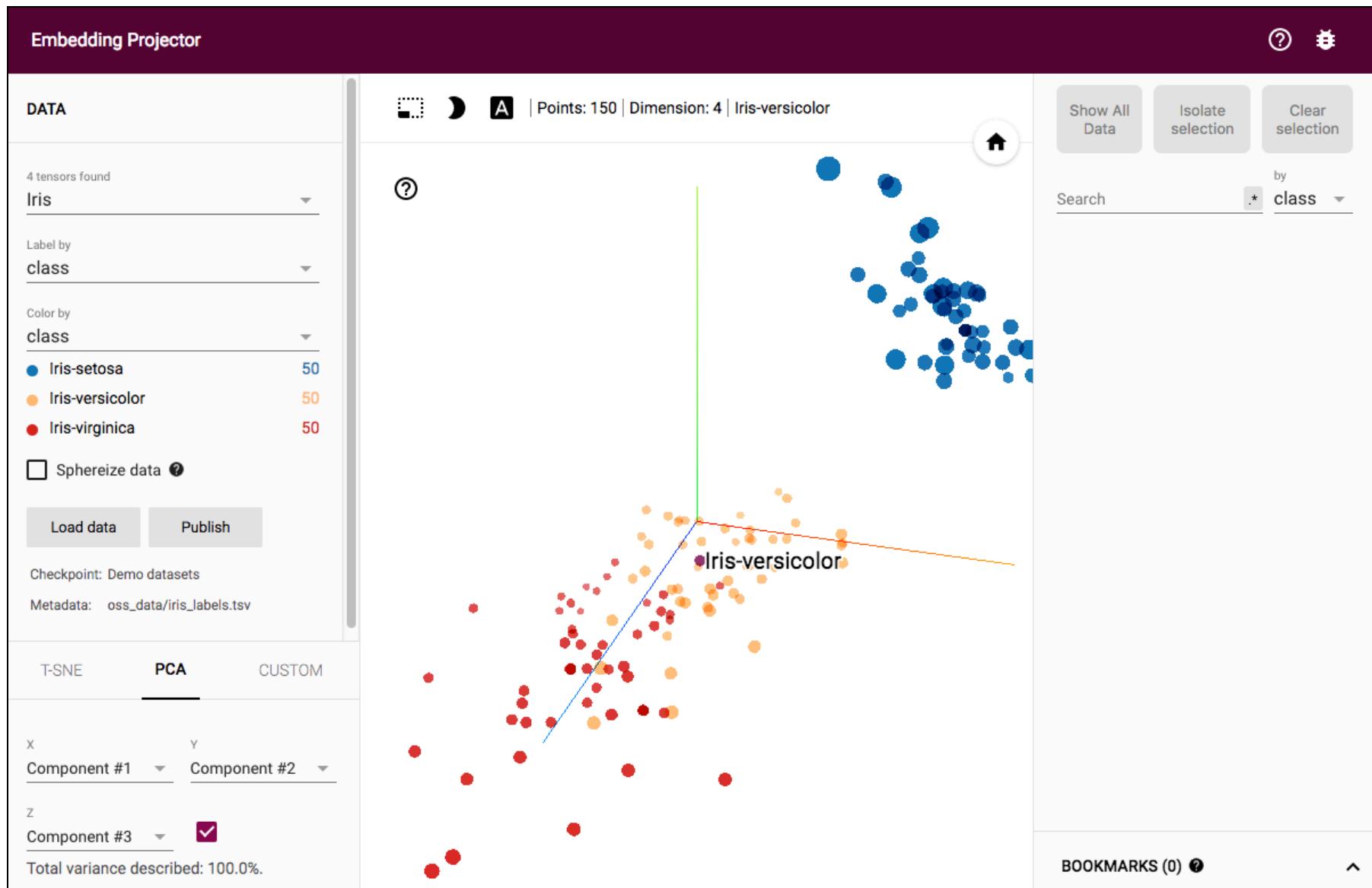


Iris data classified in
four dimensions:
“decision boundaries”

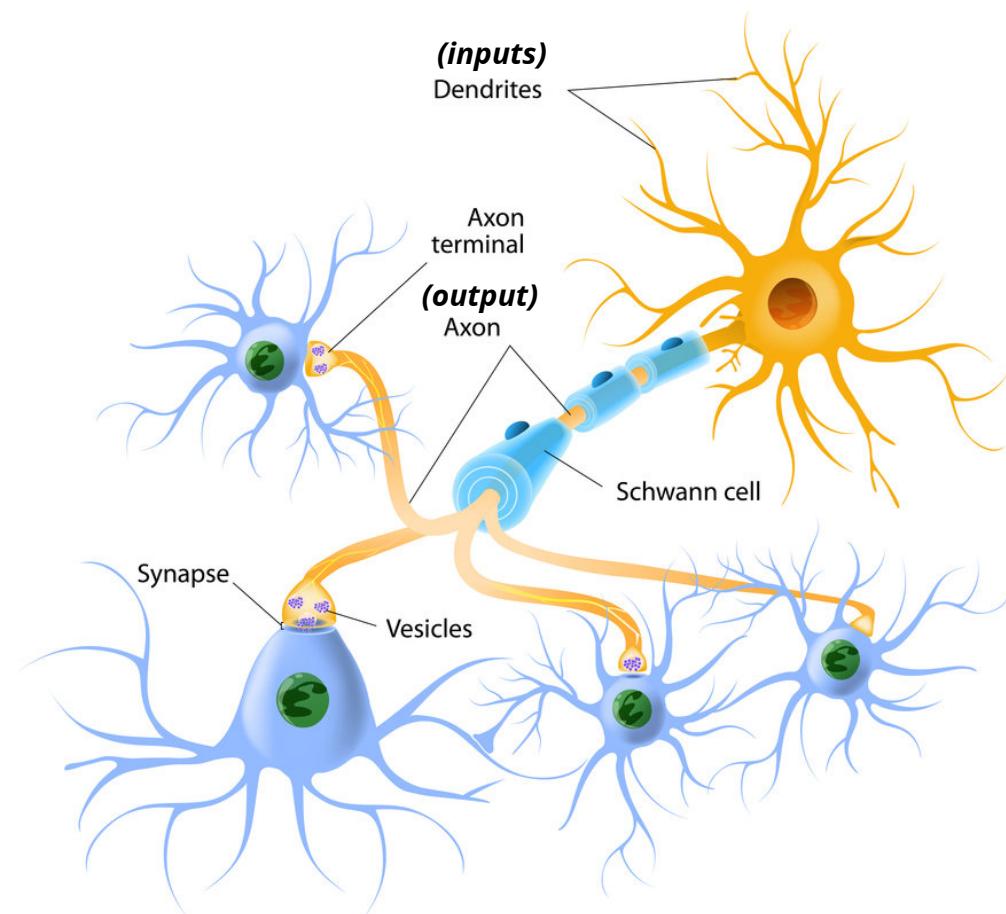
Training / test data

Features				Labels
Sepal length	Sepal width	Petal length	Petal width	Species
5.1	3.5	1.4	0.2	Iris setosa
4.9	3.0	1.4	0.2	Iris setosa
7.0	3.2	4.7	1.4	Iris versicolor
6.4	3.2	4.5	1.5	Iris versicolor
6.3	3.3	6.0	2.5	Iris virginica
5.8	3.3	6.0	2.5	Iris virginica

Visualizing Iris dataset with TensorFlow tool

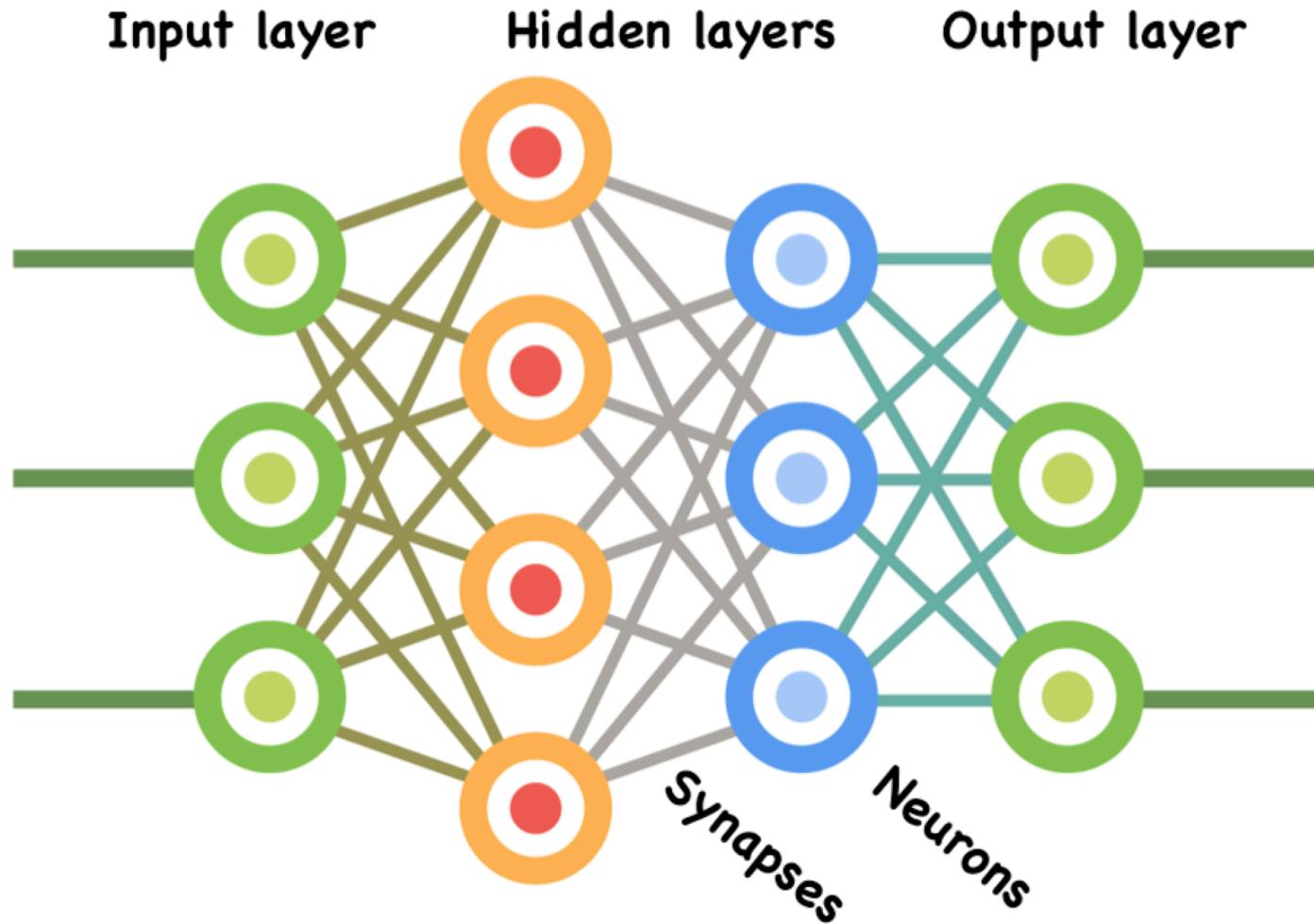


Modeling the brain works well with machine learning (ya think?)



Anatomy of an Artificial Neural Network

(aka *Deep Belief Network* when multiple hidden layers)

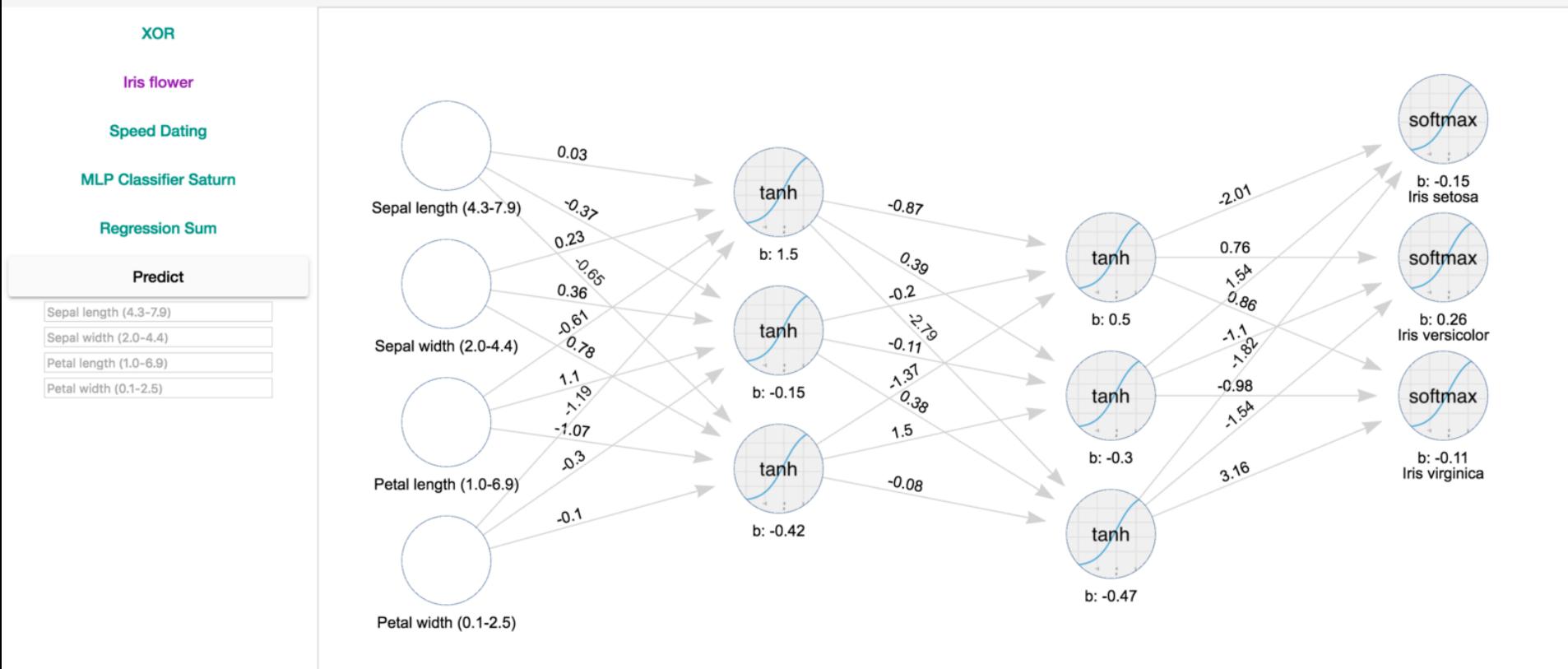


Neural net visualization app (uses Spring and DL4J)

github.com/JavaFXpert/visual-neural-net-server

github.com/JavaFXpert/ng2-spring-websocket-client

Exposed! Neural Network



Entering feature values for prediction (classification)

Iris flower

Speed Dating

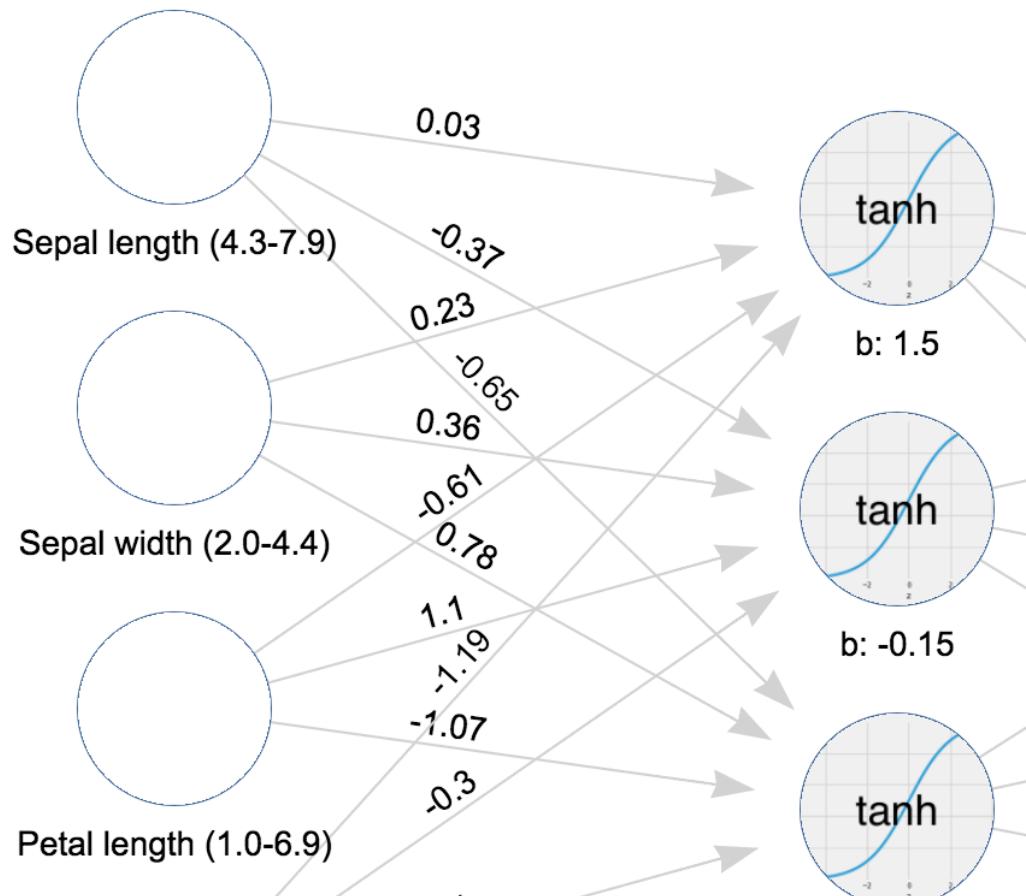
MLP Classifier Saturn

Regression Sum

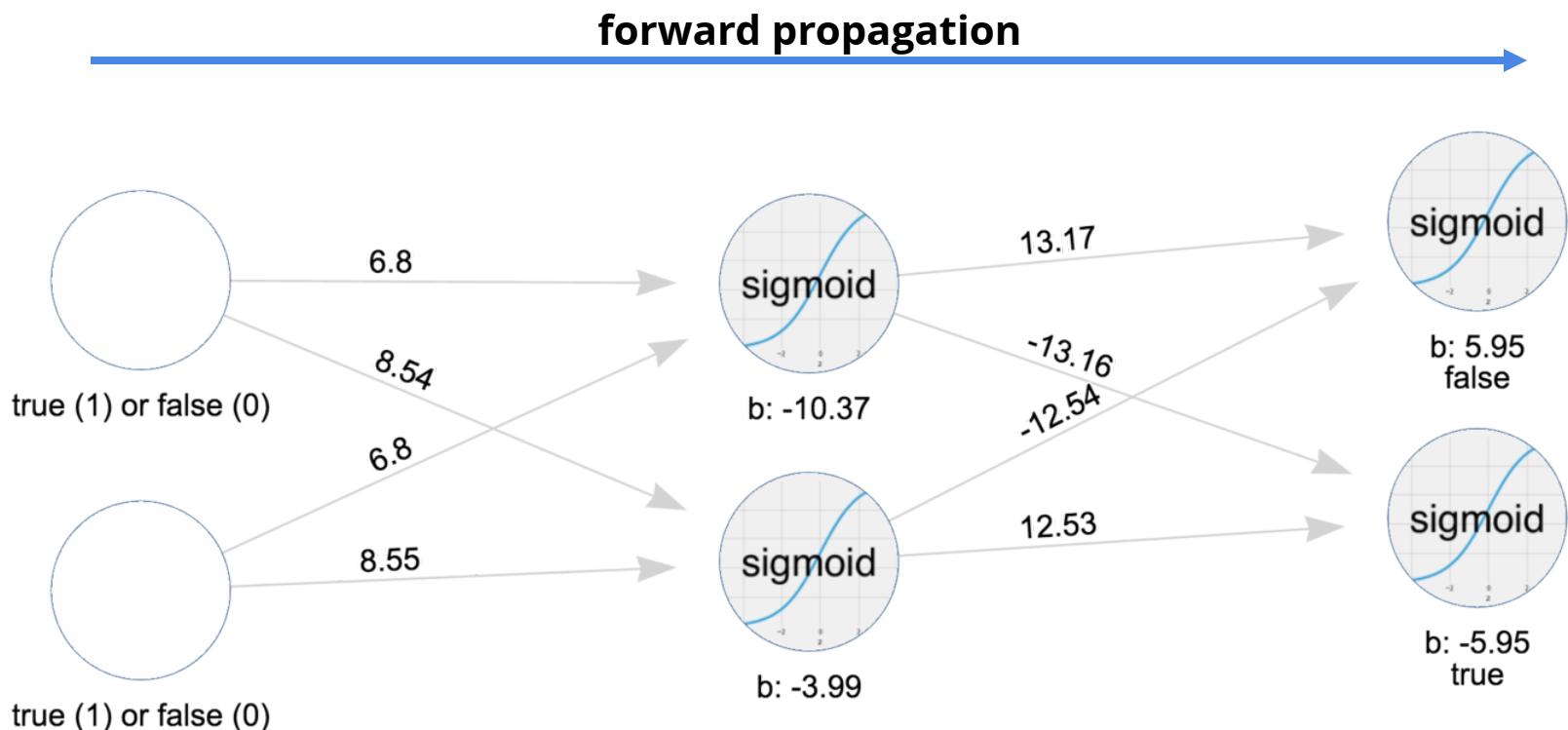
Predict

6.4
3.2
4.5
1.5

A diagram illustrating the input process for a MLP Classifier. On the left, there is a user interface with four input fields containing the values 6.4, 3.2, 4.5, and 1.5. Blue arrows point from each of these fields to a central 'Predict' button. Above the 'Predict' button, the text 'Regression Sum' is displayed with a blue arrow pointing towards it. To the right of the 'Predict' button, the text 'Iris flower', 'Speed Dating', and 'MLP Classifier Saturn' are listed vertically.



Simple neural network trained for XOR logic



Feedforward calculations with XOR example

For each layer:

Multiply inputs by weights:

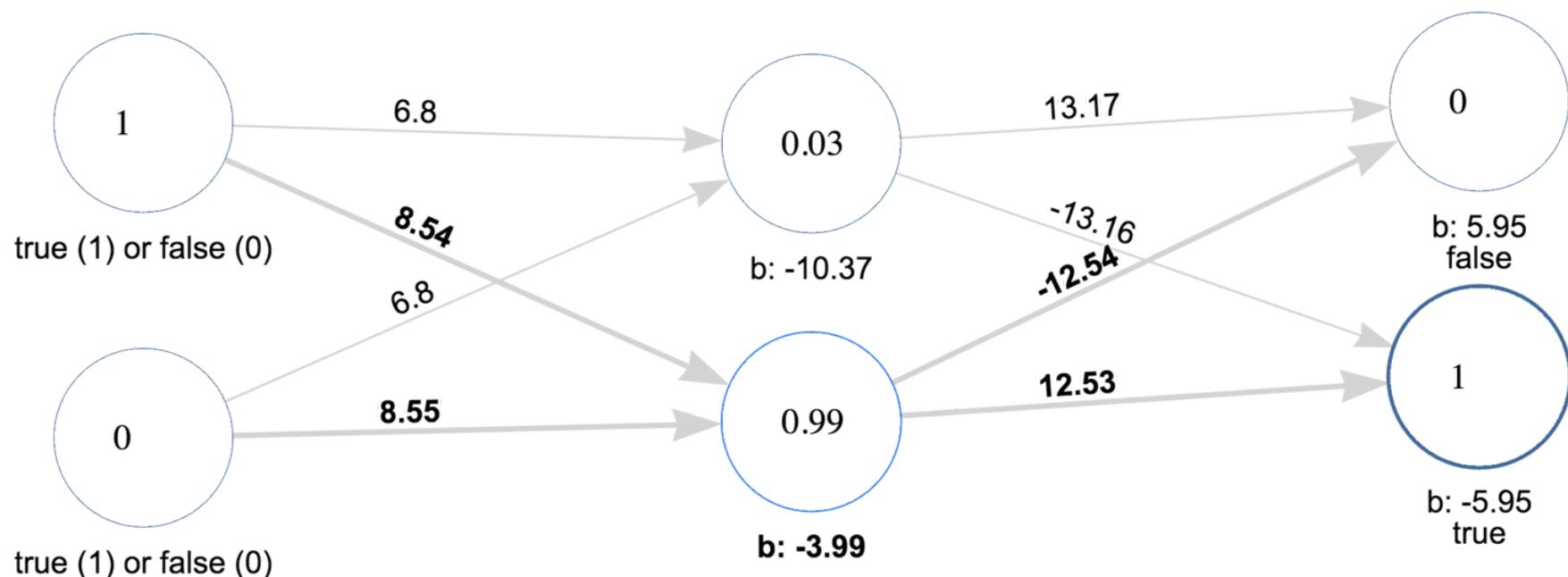
$$(1 \times 8.54) + (0 \times 8.55) = 8.54$$

Add bias:

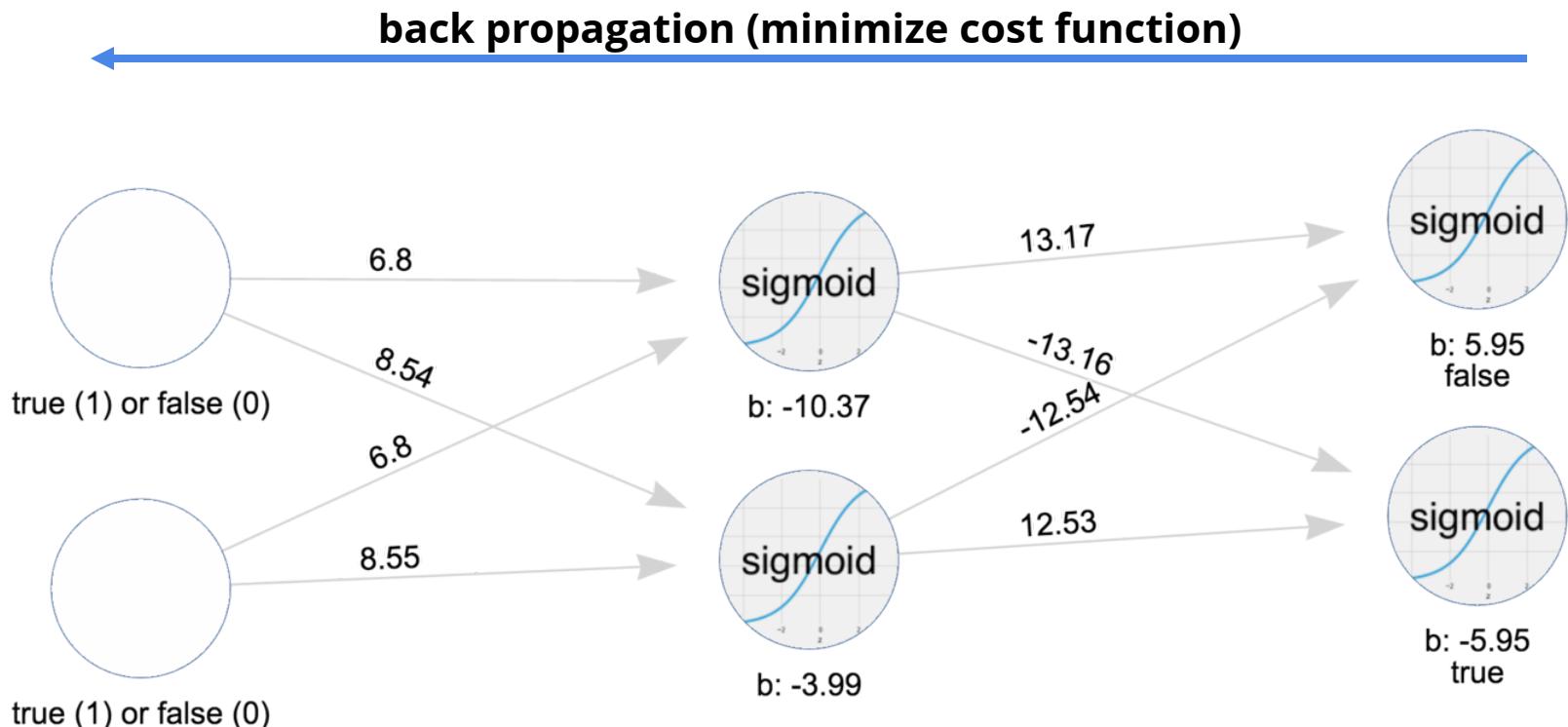
$$8.54 + (-3.99) = 4.55$$

Use sigmoid activation function:

$$1 / (1 + e^{-4.55}) = 0.99$$



Simple neural network trained for XOR logic



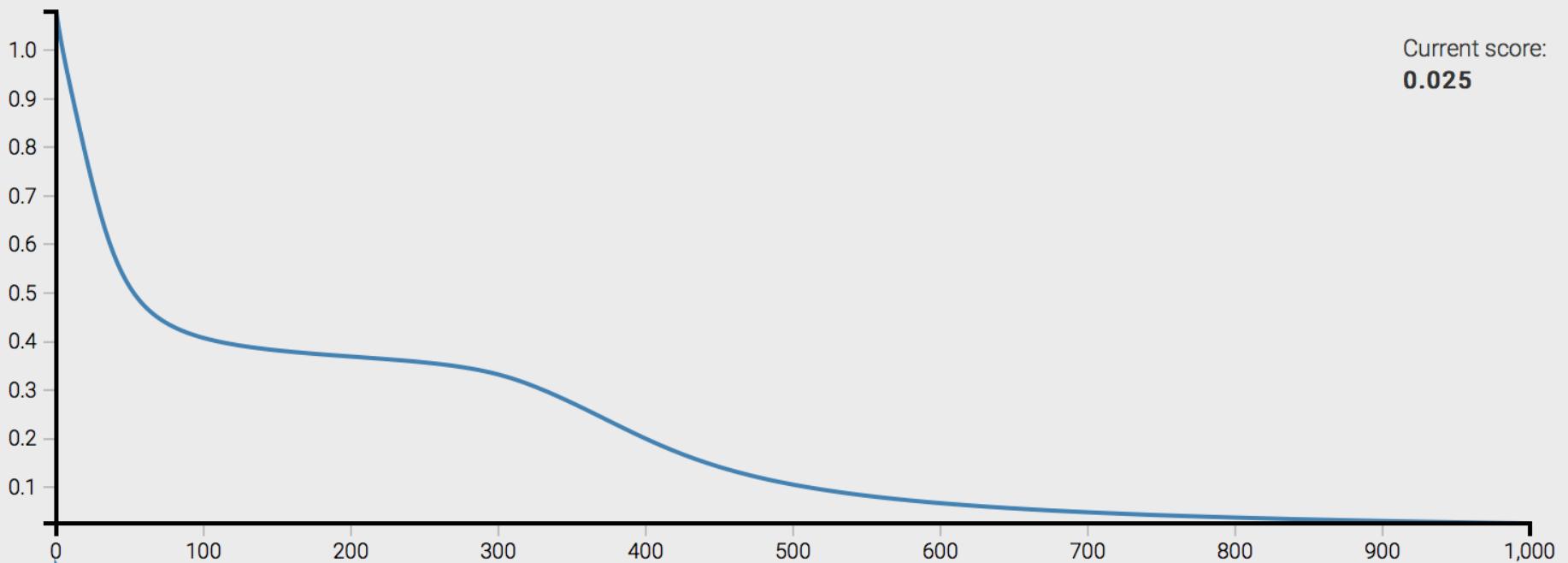
Back propagation

(Uses *gradient descent* to iteratively minimize the cost function)



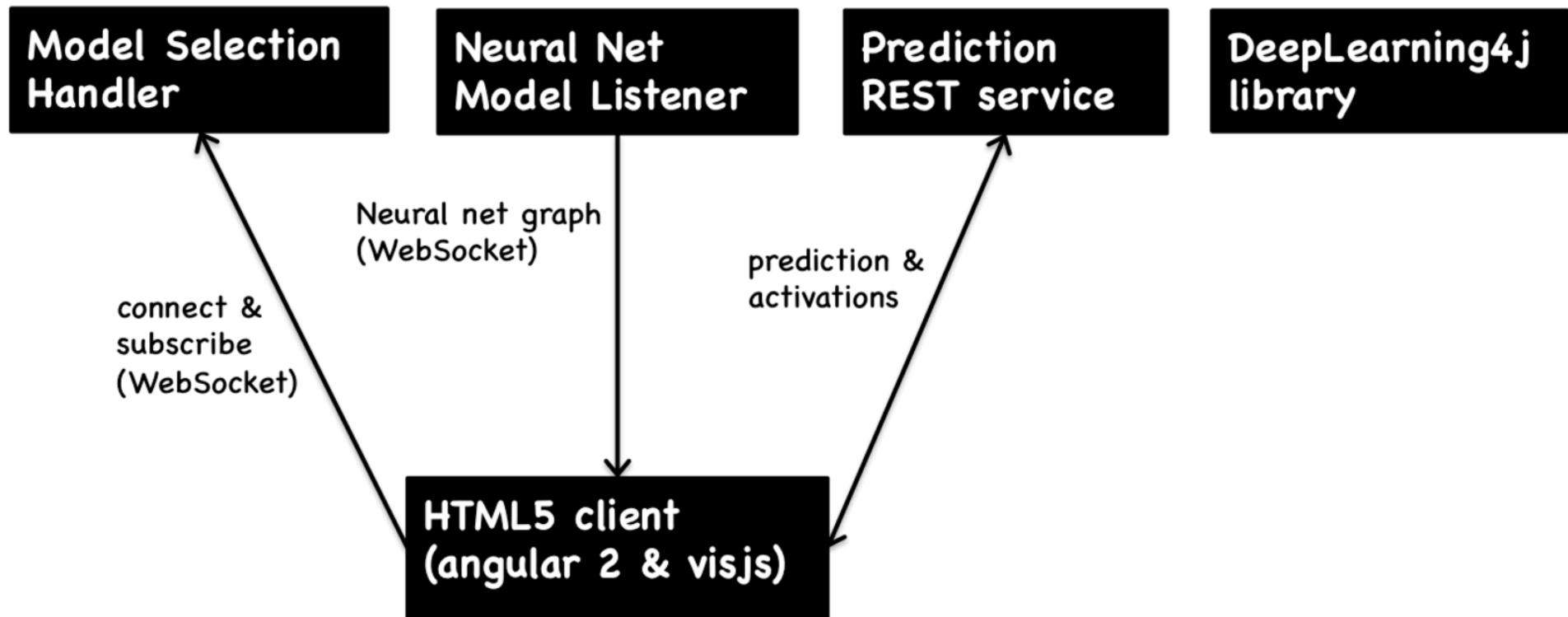
DeepLearning4j UI

Score vs. Iteration #

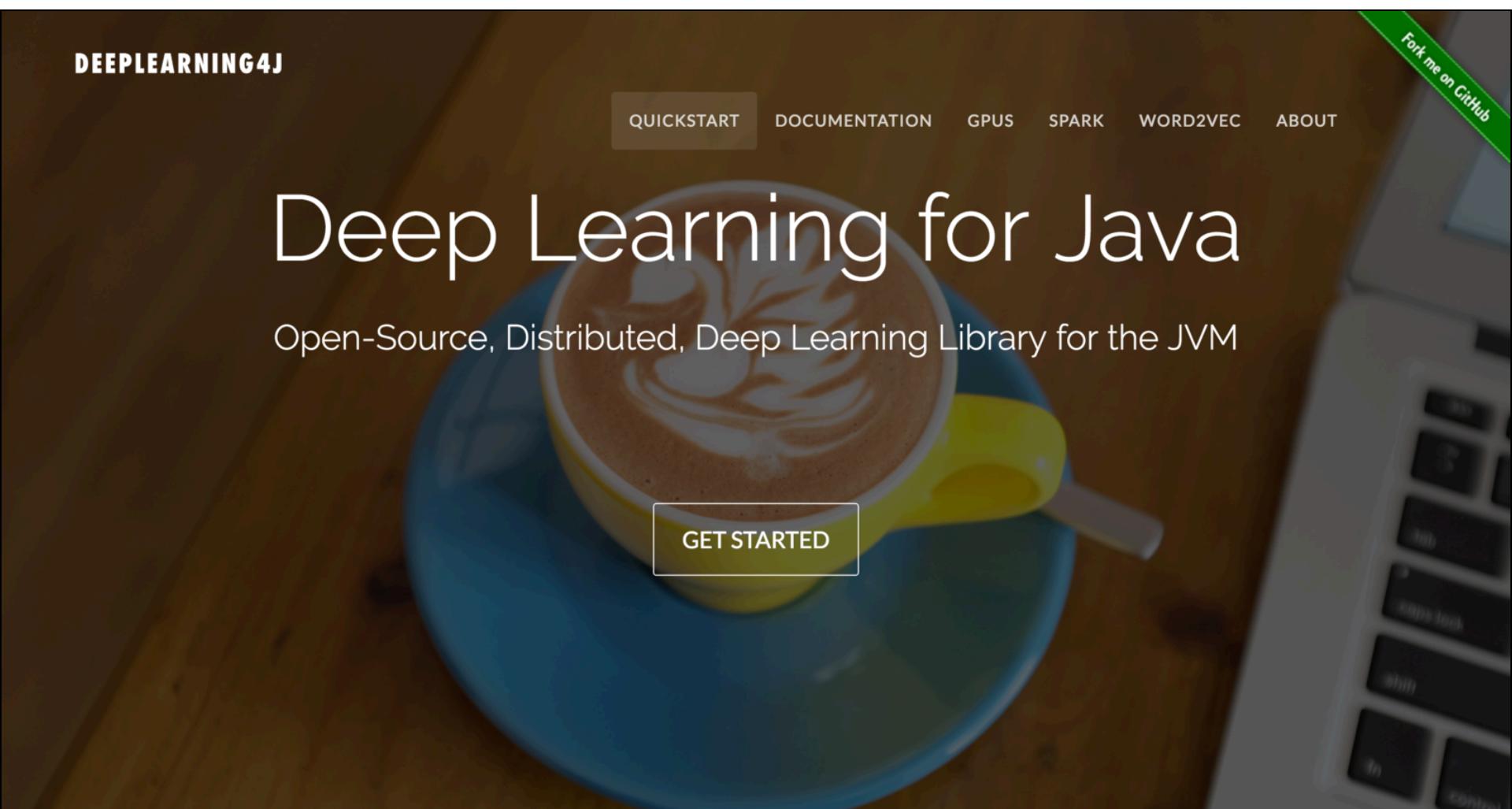


Visual Neural Network application architecture

Spring makes REST services and WebSockets easy as π



The app leverages machine learning libraries found at deeplearning4j.org



To quickly create a Spring project, visit
start.spring.io

SPRING INITIALIZR bootstrap your application now

Generate a with Spring Boot

Project Metadata

Artifact coordinates

Group

Artifact

Dependencies

Add Spring Boot Starters and dependencies to your application

Search for dependencies

Selected Dependencies

Generate Project ⌘ + ↵

Don't know what to look for? Want more options? [Switch to the full version.](#)

Is Optimizing your Neural Network a Dark Art ?

Excellent article by [Preetham V V](#) on neural networks and choosing hyperparameters



Preetham V V [Follow](#)

#AI & #MachineLearning enthusiast. Author: Java Web Services / Internet Security & Firewalls. VP, B...

Aug 10 · 14 min read

Is Optimizing your Neural Network a Dark Art?

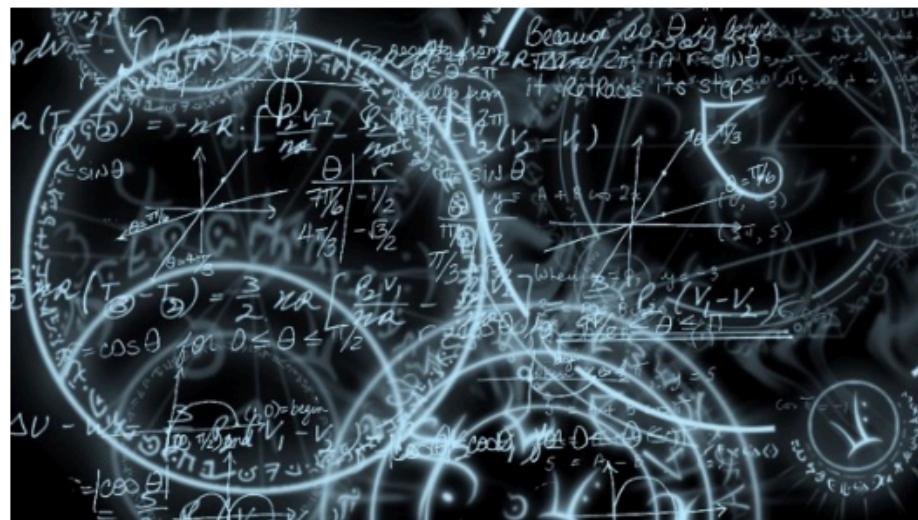
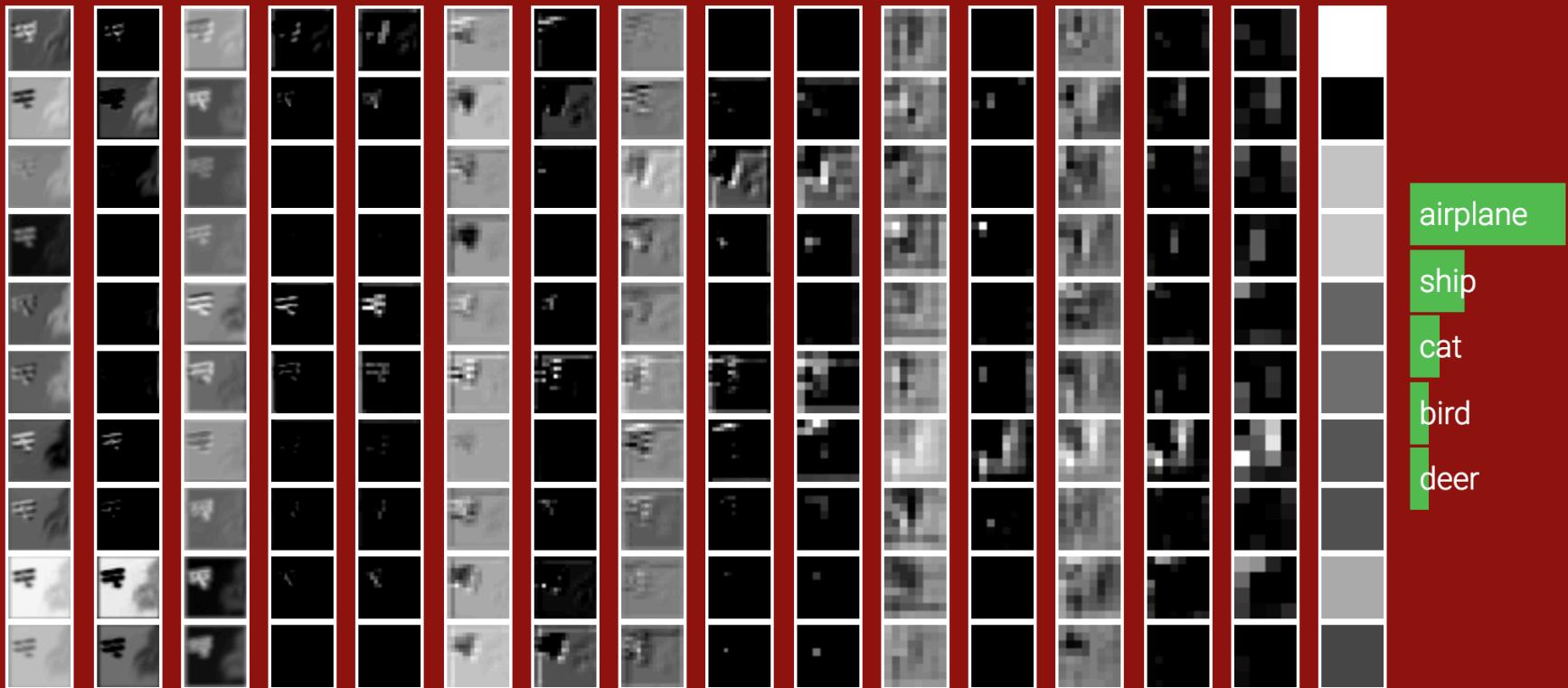


photo credit

Just like in real life, different people learn at different pace, using different techniques and understand or retain qualitatively different aspects of what they have learnt. Artificial Neural Networks or ANNs are no different. ANNs share the same pitfalls of the brain, while it tries to replicate the strengths of the brain. In other words, they learn

Various Neural Networks

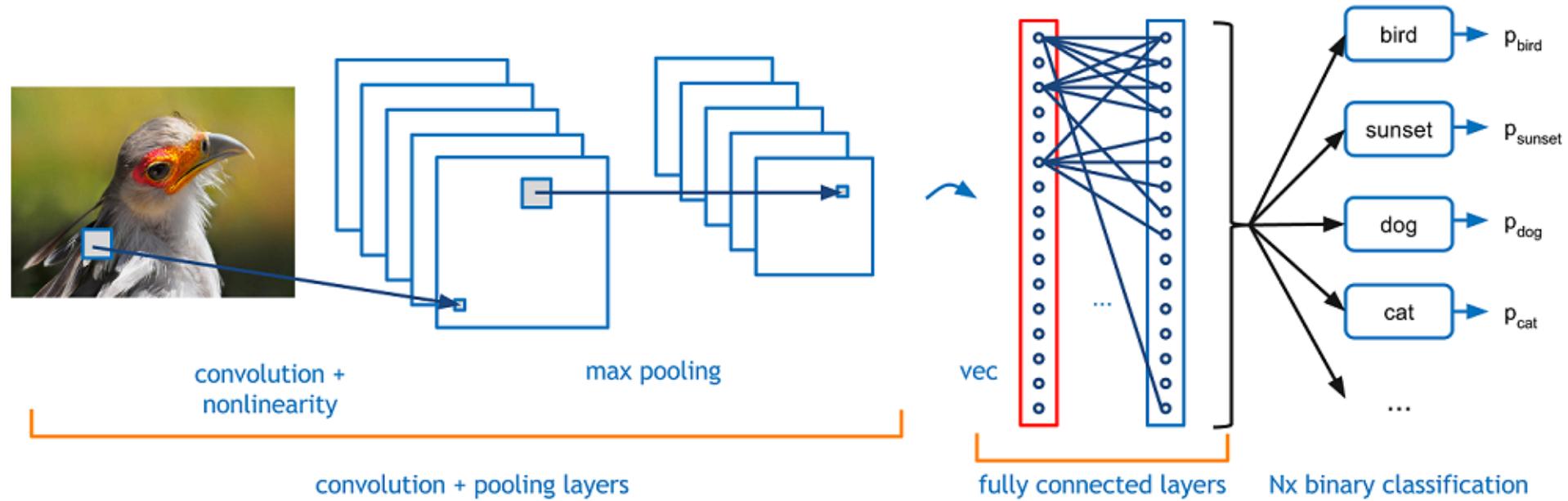
Convolutional Neural Network for recognizing images



cs231n.stanford.edu/

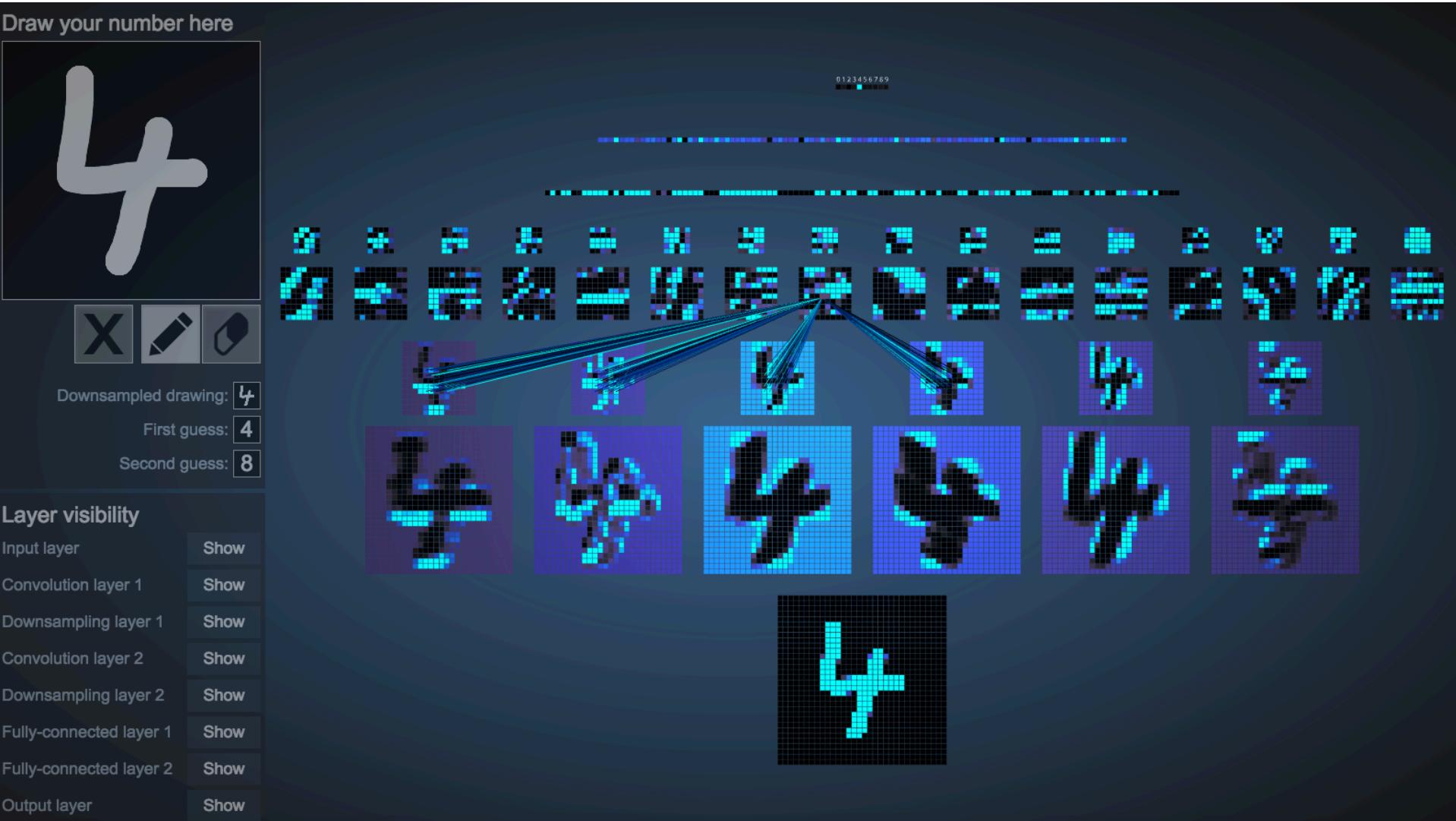
@JavaFXpert

Convolutional neural network architecture



adeshpande3.github.io/A-Beginner's-Guide-To-Understanding-Convolutional-Neural-Networks/
[by Adit Deshpande]

Peeking into a convolutional neural network



<http://scs.ryerson.ca/~aharley/vis/> [by Adam Harley]

@JavaFXpert

Time series prediction with neural networks

What is happening? What is most likely to happen next?

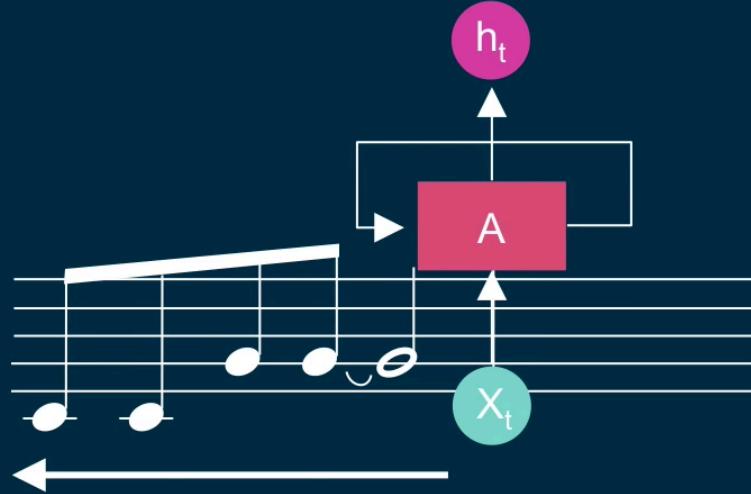


This is a job for a Recurrent Neural Network

What is happening? What is most likely to happen next?

Music composition with an RNN

Prediction



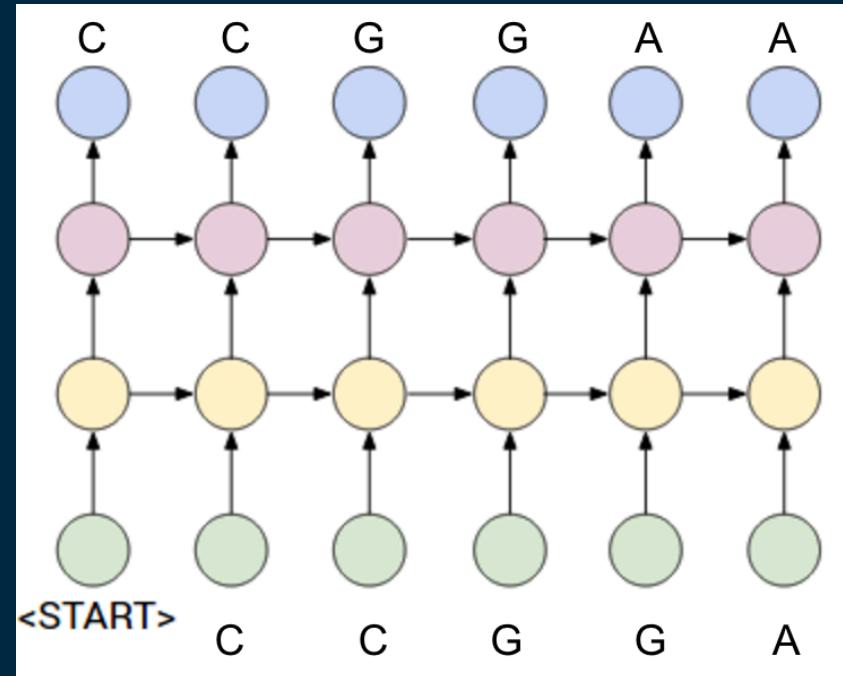
From: Music and Art Generation using Machine Learning | Curtis Hawthorne

Predicting the most likely next note

Note-based Recurrent Neural Network

Extract melodies from songs

Train the network to predict the next note



From: Music and Art Generation using Machine Learning | Curtis Hawthorne

Playing a duet with neural networks

A.I. Duet

A piano that responds to you.

 PLAY

 How it works

This is an
A.I.
Experiment

Made with
some friends from



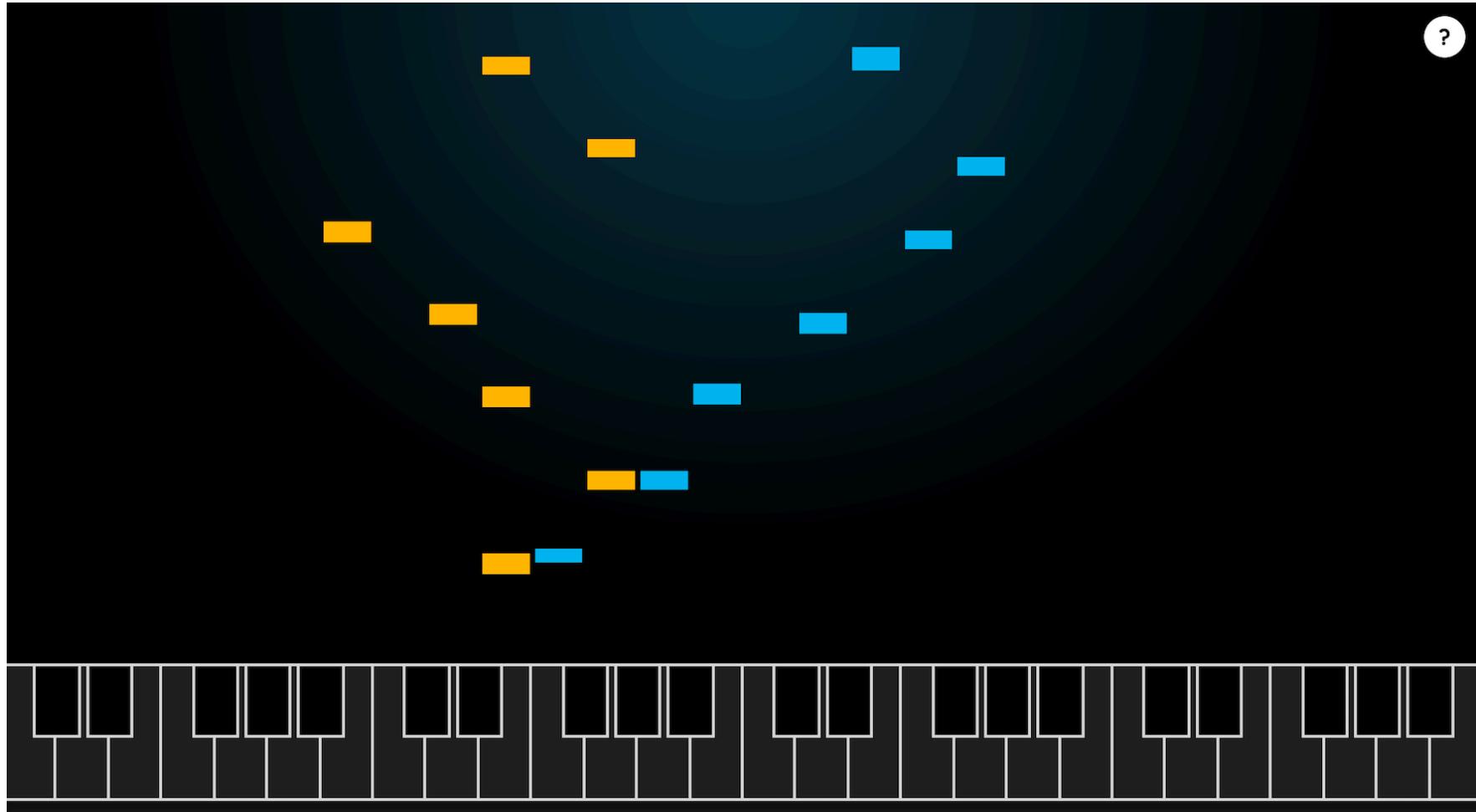

Built using


[Privacy & Terms](#)

<https://aiexperiments.withgoogle.com/ai-duet/view/>

@JavaFXpert

Playing a duet with neural networks



<https://aiexperiments.withgoogle.com/ai-duet/view/>

Unsupervised Learning

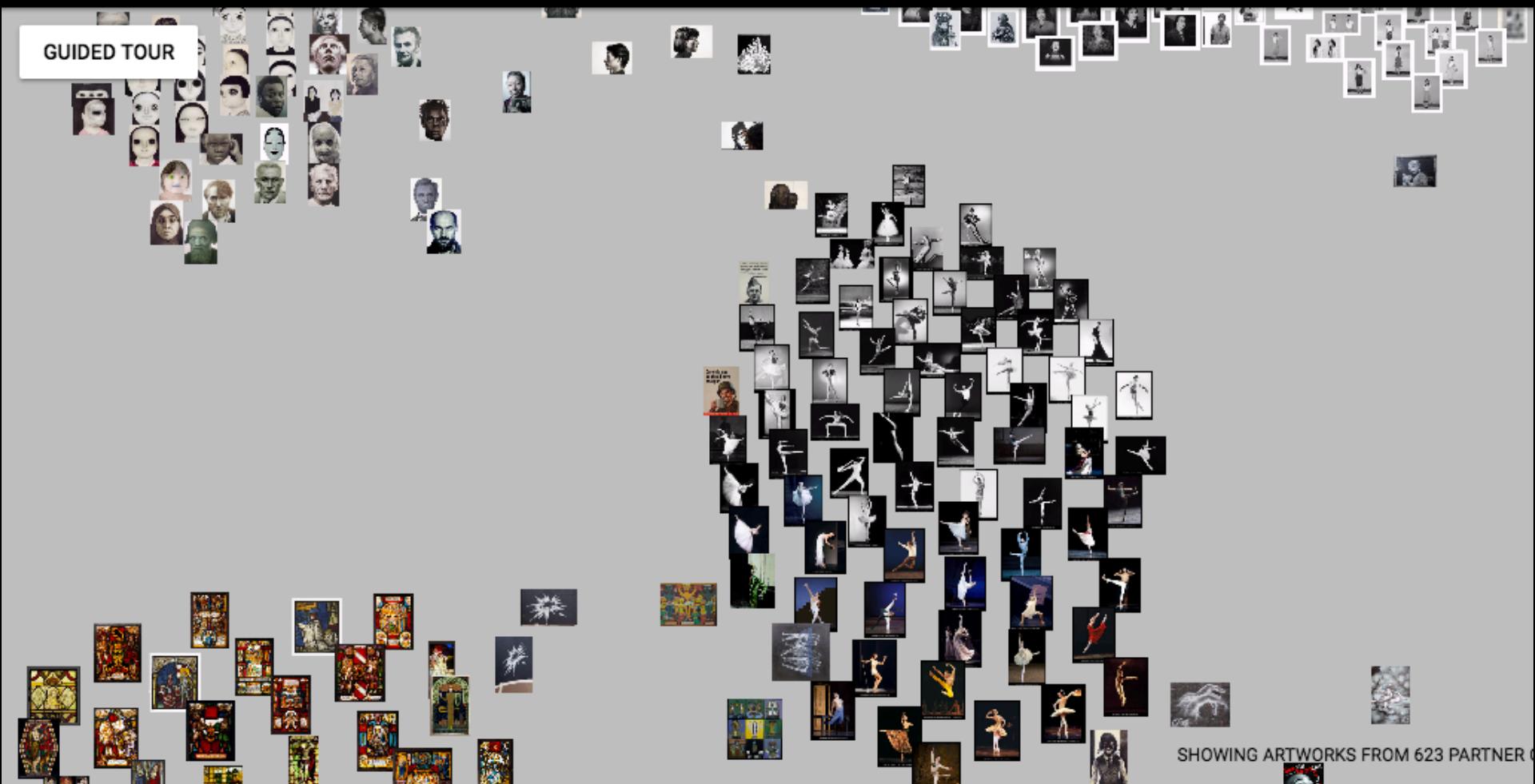
(Let's dive in now)



@KatharineCodes @JavaFXpert

Using unsupervised learning to map artworks

Google Arts & Culture | Experiments | t-SNE Map

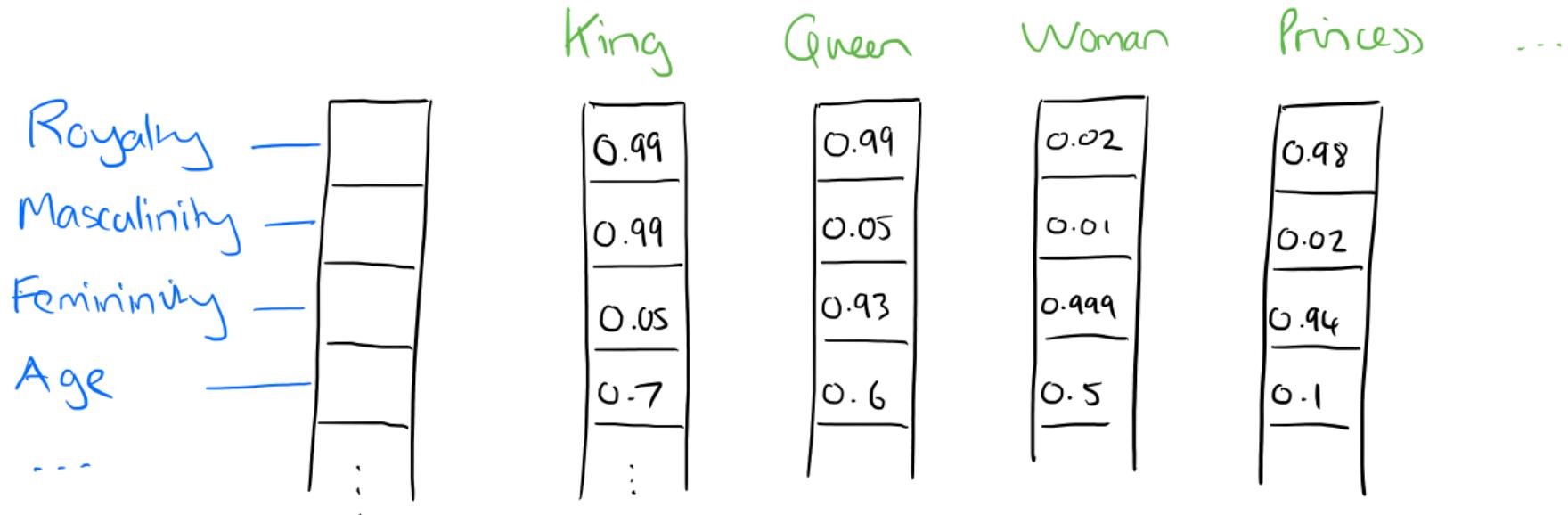


artsexperiments.withgoogle.com/tsnemap

@JavaFXpert

Using unsupervised learning to map words

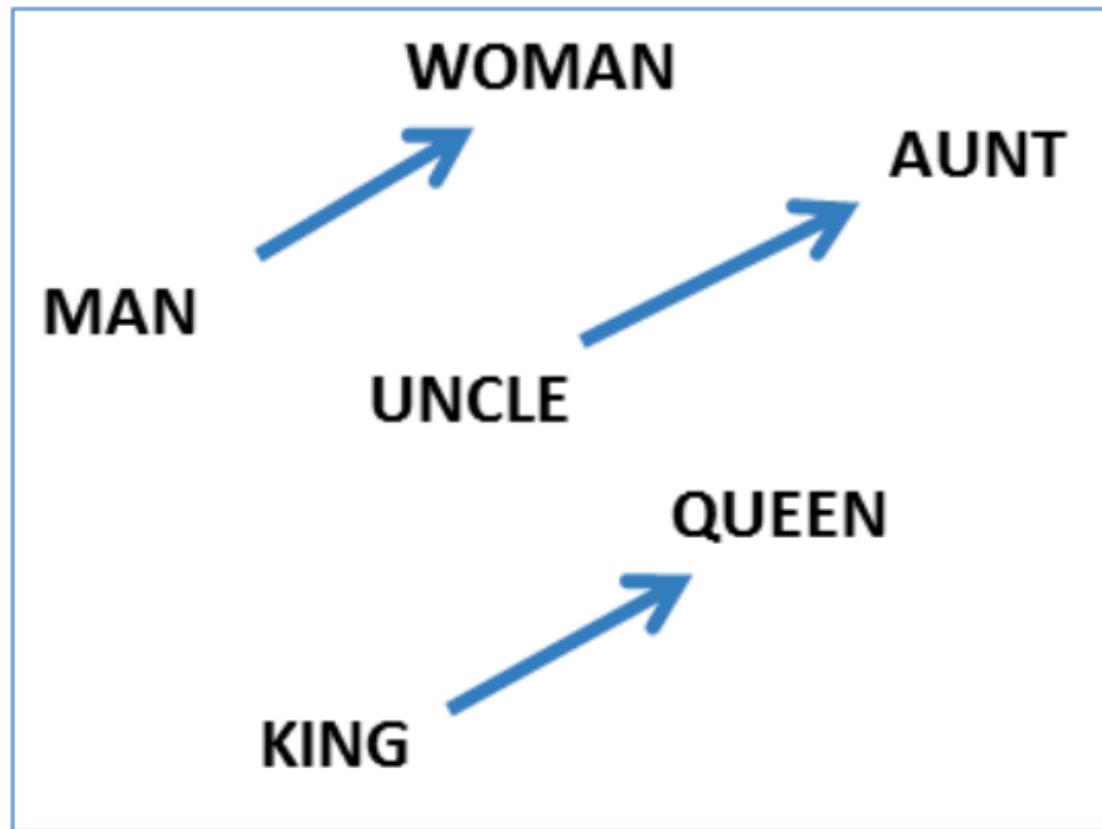
word2vec vector representations of words



The amazing power of word vectors - Adrian Colyer

Using unsupervised learning to map words

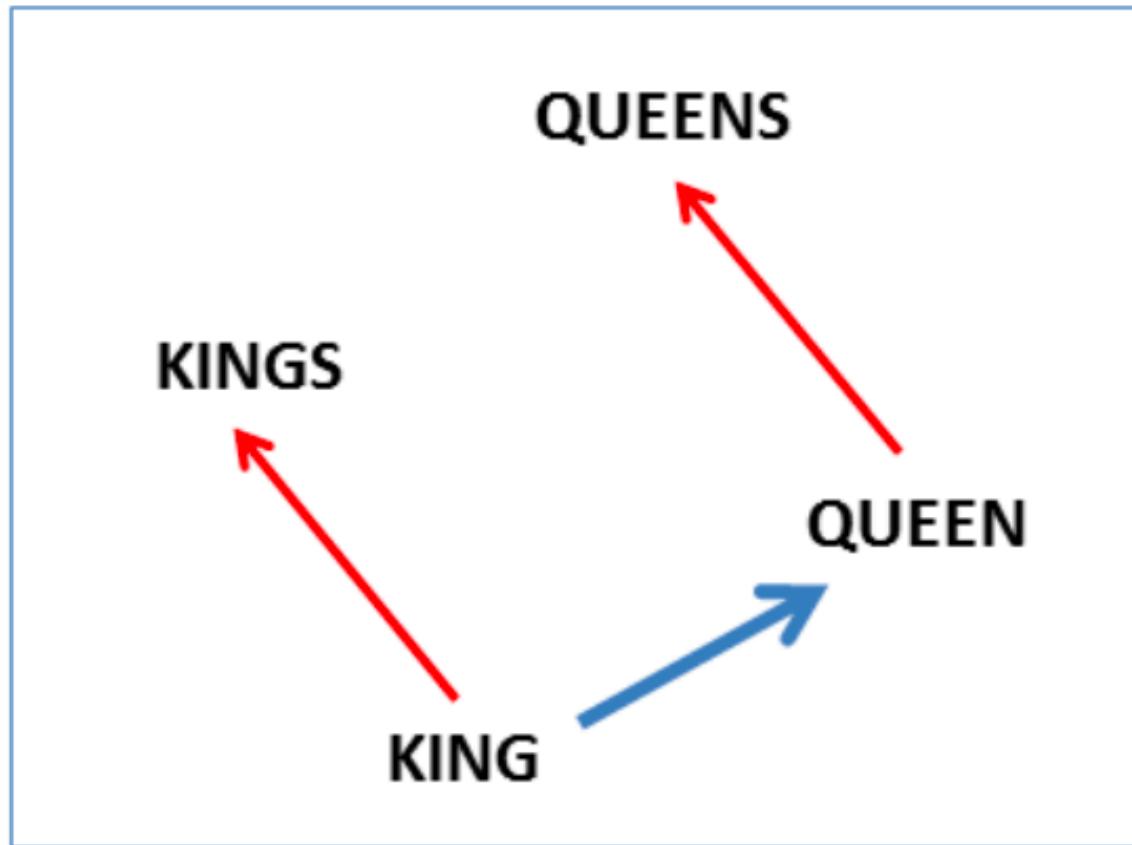
word2vec vector offsets for gender relationships



The amazing power of word vectors - Adrian Colyer

Using unsupervised learning to map words

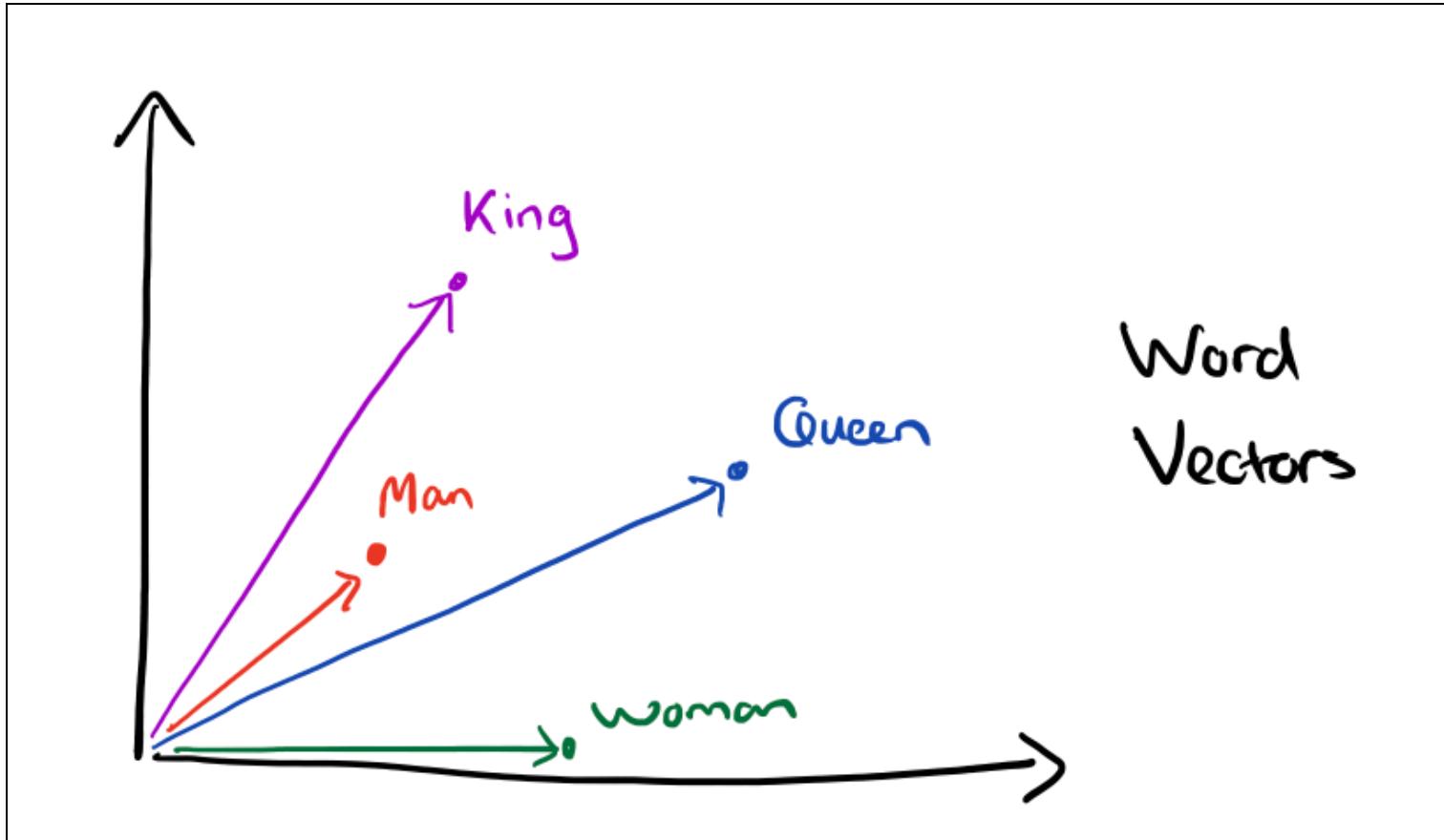
word2vec vector offsets for plural relationships



The amazing power of word vectors - Adrian Colyer

Using unsupervised learning to map words

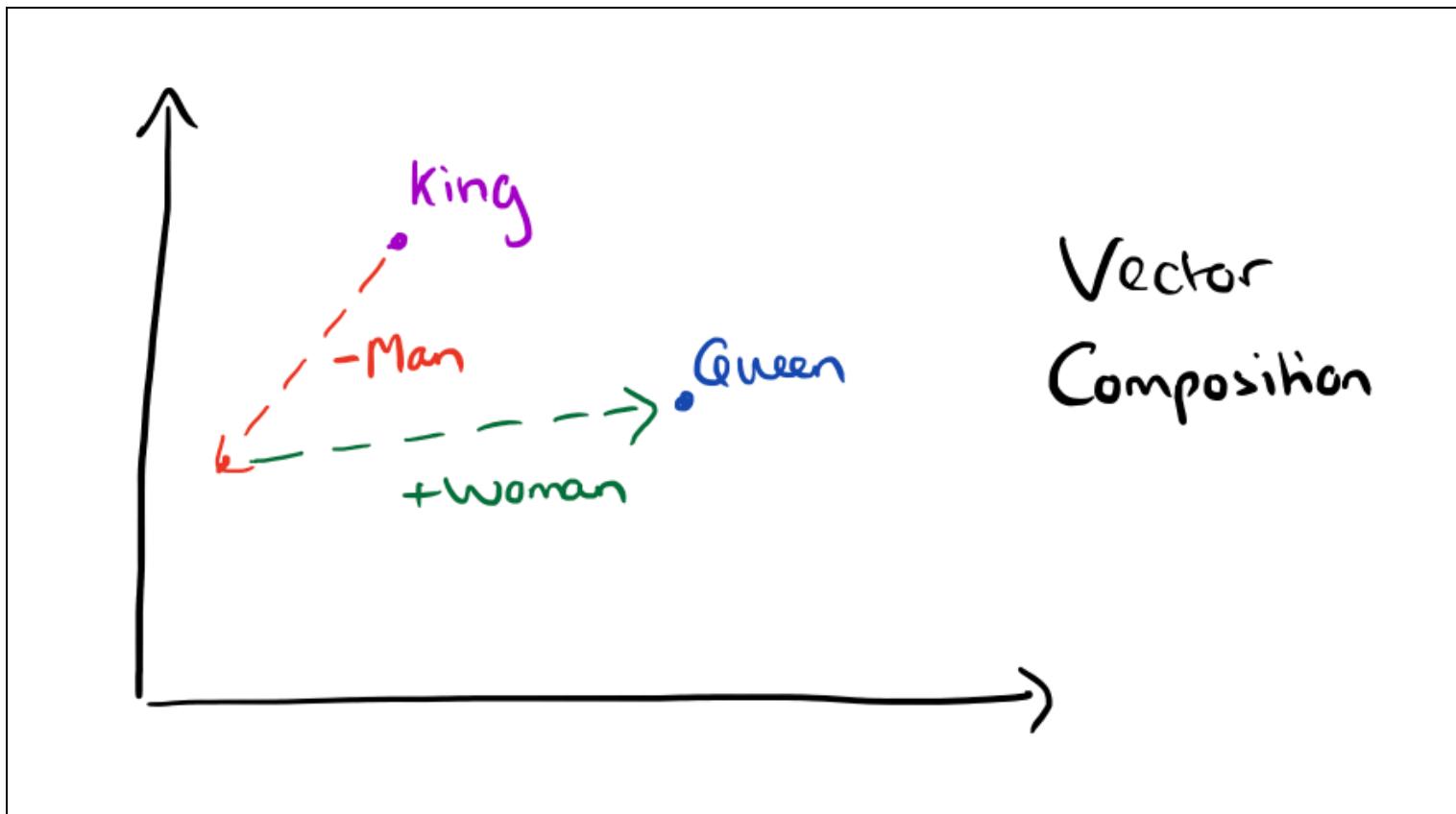
word2vec vector arithmetic



The amazing power of word vectors - Adrian Colyer

word2vec vector arithmetic

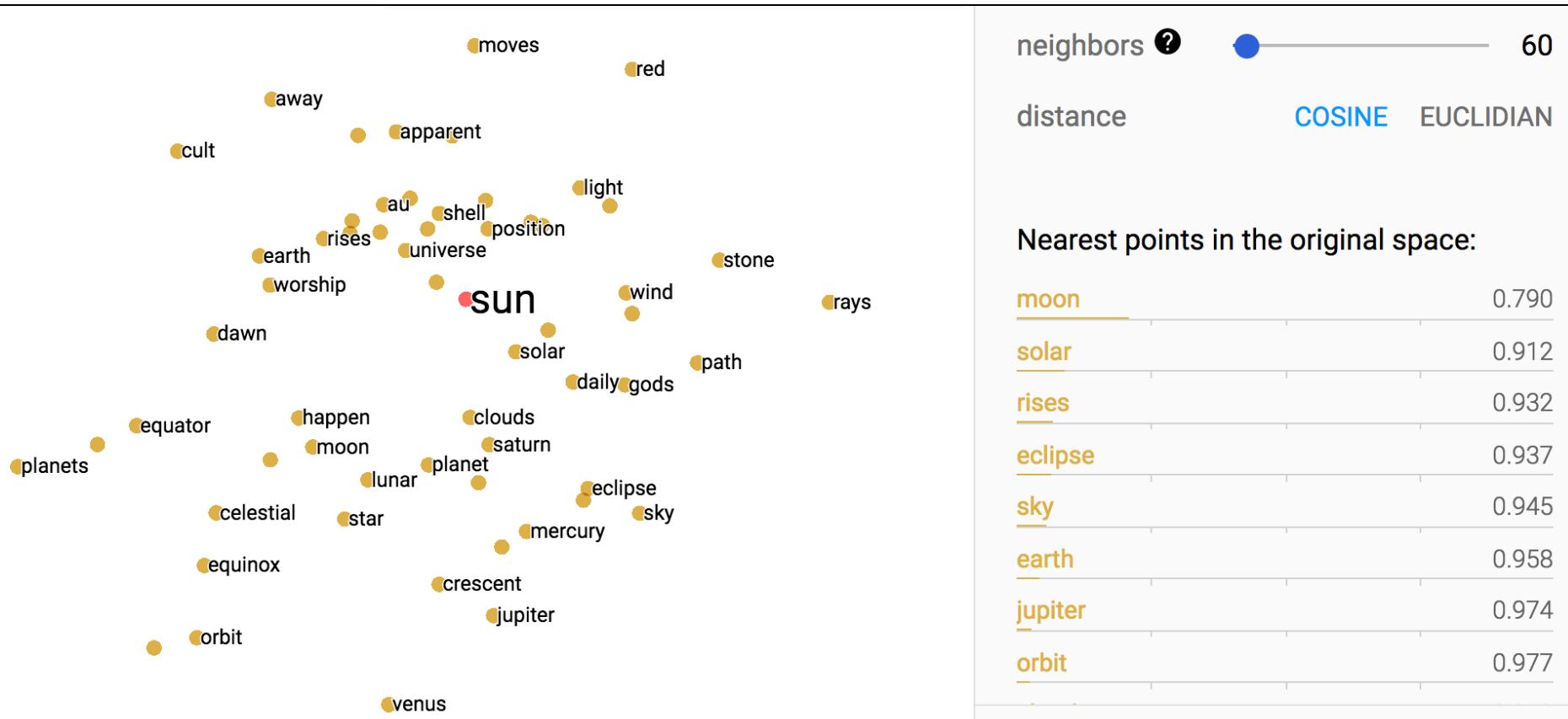
$$\text{King} - \text{Man} + \text{Woman} = \text{Queen}$$



The amazing power of word vectors - Adrian Colyer

Visualizing word2vec words & points

using Tensorflow Embedding Projector



Reinforcement Learning (Let's dive in now)

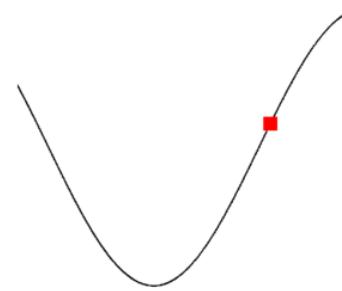
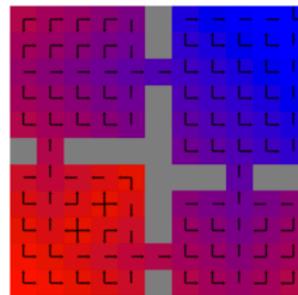
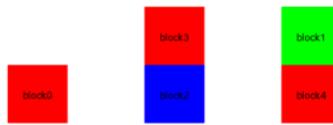


Using BURLAP for Reinforcement Learning

burlap.cs.brown.edu

BURLAP

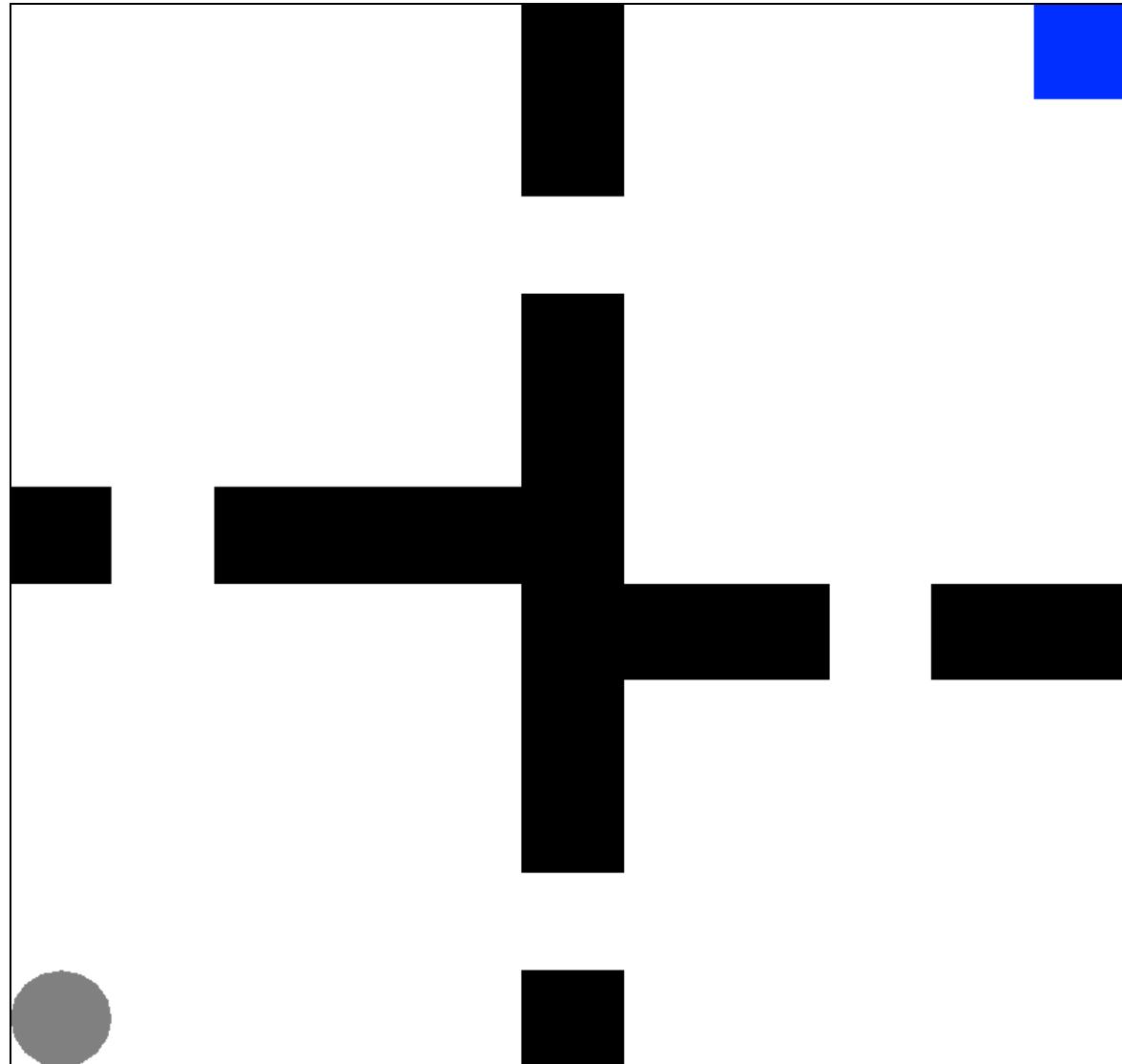
[Home](#) | [Updates](#) | [Information](#) | [F.A.Q.](#) | [Tutorials](#) | [Java Doc](#)



About

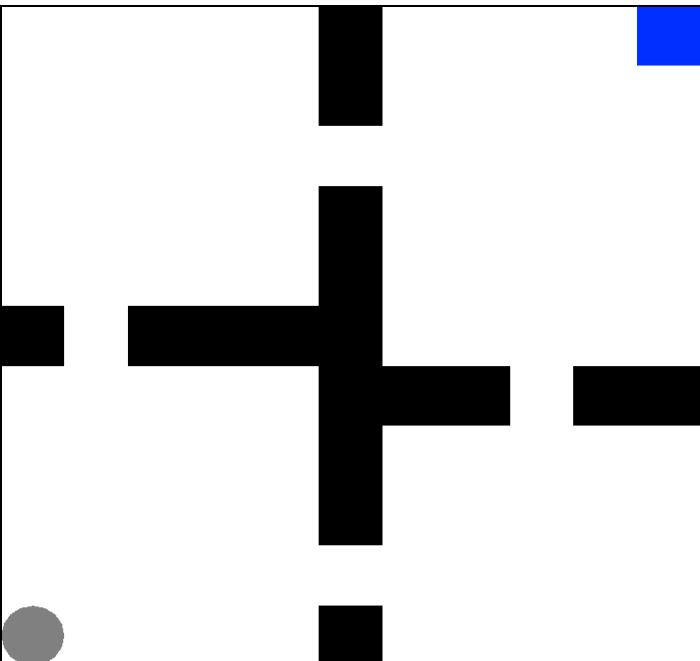
The Brown-UMBC Reinforcement Learning and Planning (**BURLAP**) java code library is for the use and development of single or multi-agent planning and learning algorithms and domains to accompany them. BURLAP uses a highly flexible system for defining states and actions of nearly any kind of form, supporting discrete, continuous, and relational domains. Planning and learning algorithms range from classic forward search planning to value function-based stochastic planning and learning algorithms. Also included is a set of analysis tools such as a common framework for the visualization of domains and agent performance in various domains.

Learning to Navigate a Grid World with Q-Learning



Rules of this Grid World

Goal of an episode is to maximize total reward

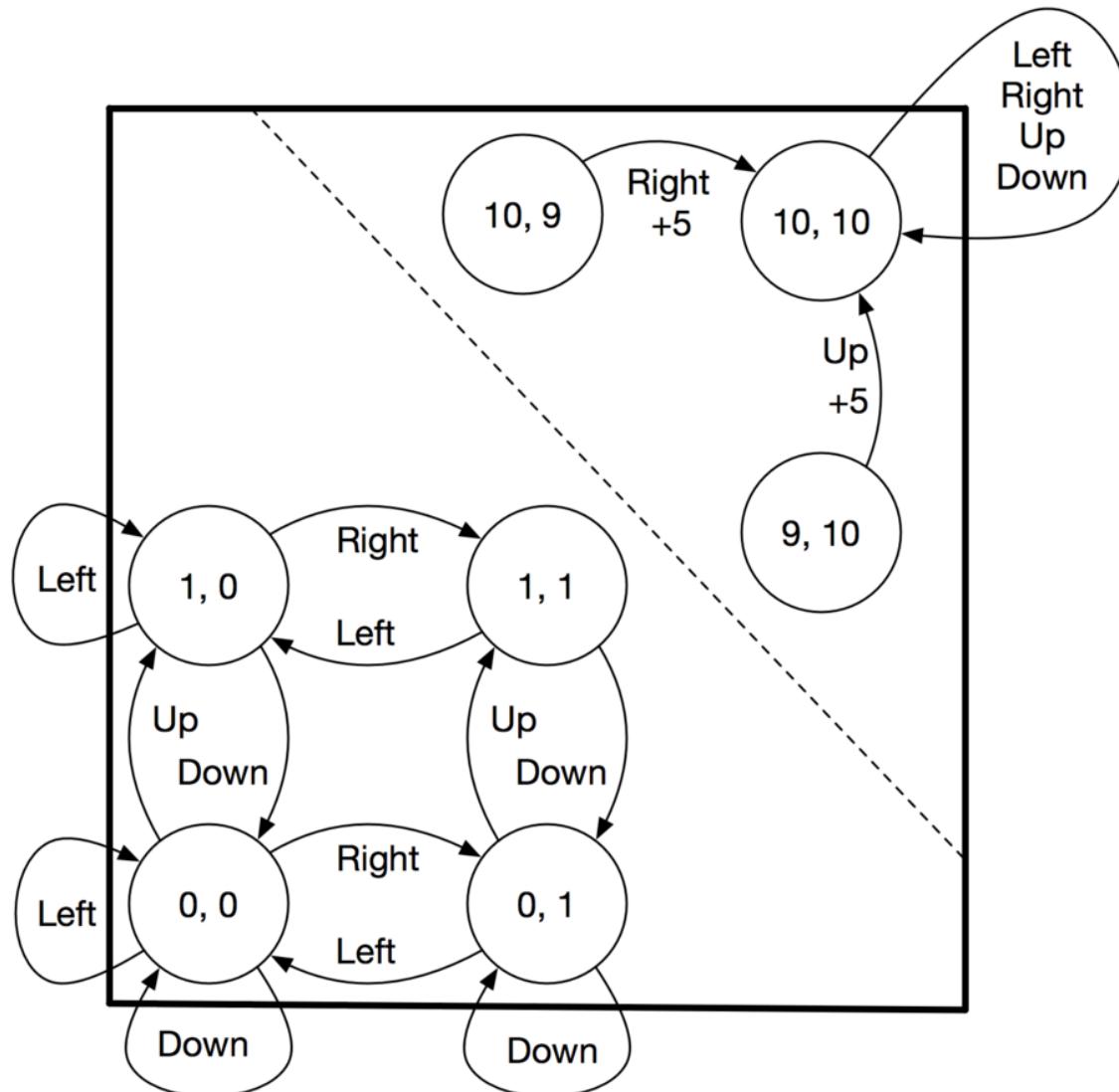


- Agent may move left, right, up, or down (*actions*)
- Reward is 0 for each move
- Reward is 5 for reaching top right corner (*terminal state*)
- Agent can't move into a wall or off-grid
- Agent doesn't have a model of the grid world. It must discover as it interacts.

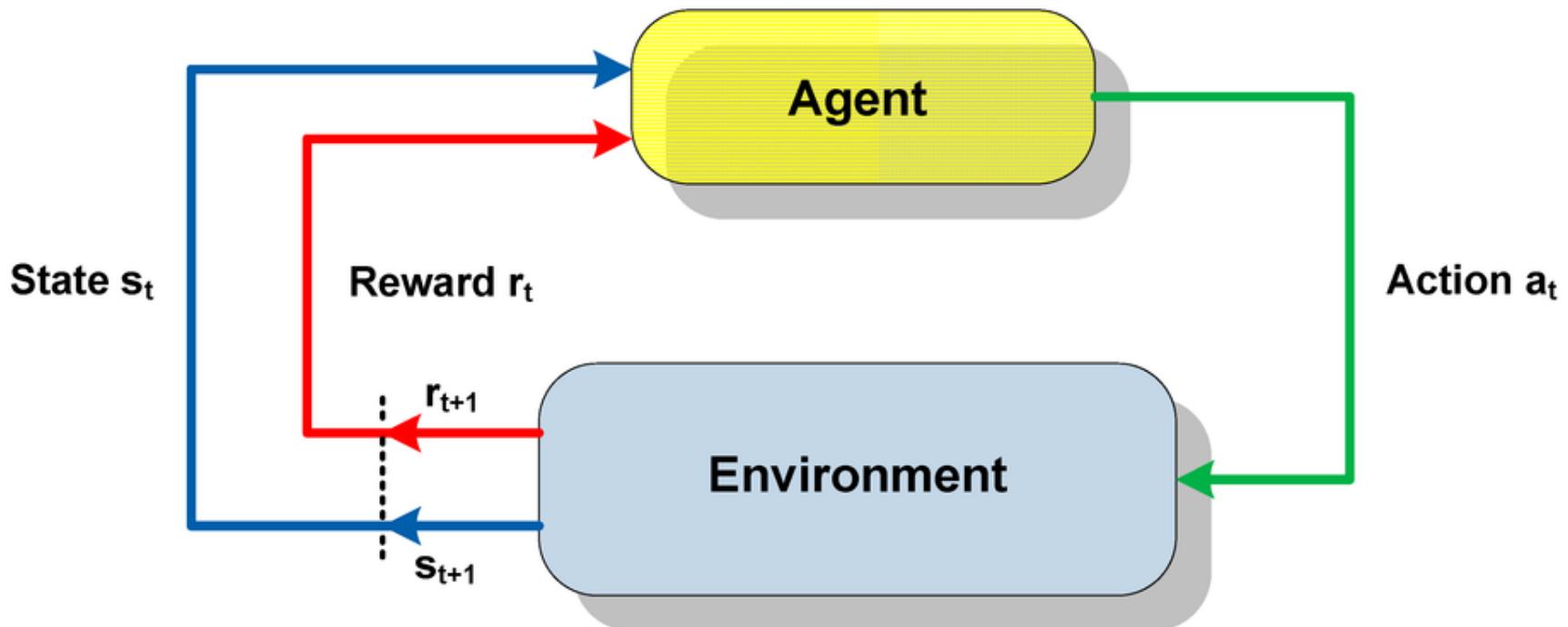
Challenge: Given that there is only one state that gives a reward, how can the agent work out what actions will get it to the reward?
(AKA the *credit assignment problem*)

This Grid World's MDP (Markov Decision Process)

In this example, all actions are deterministic

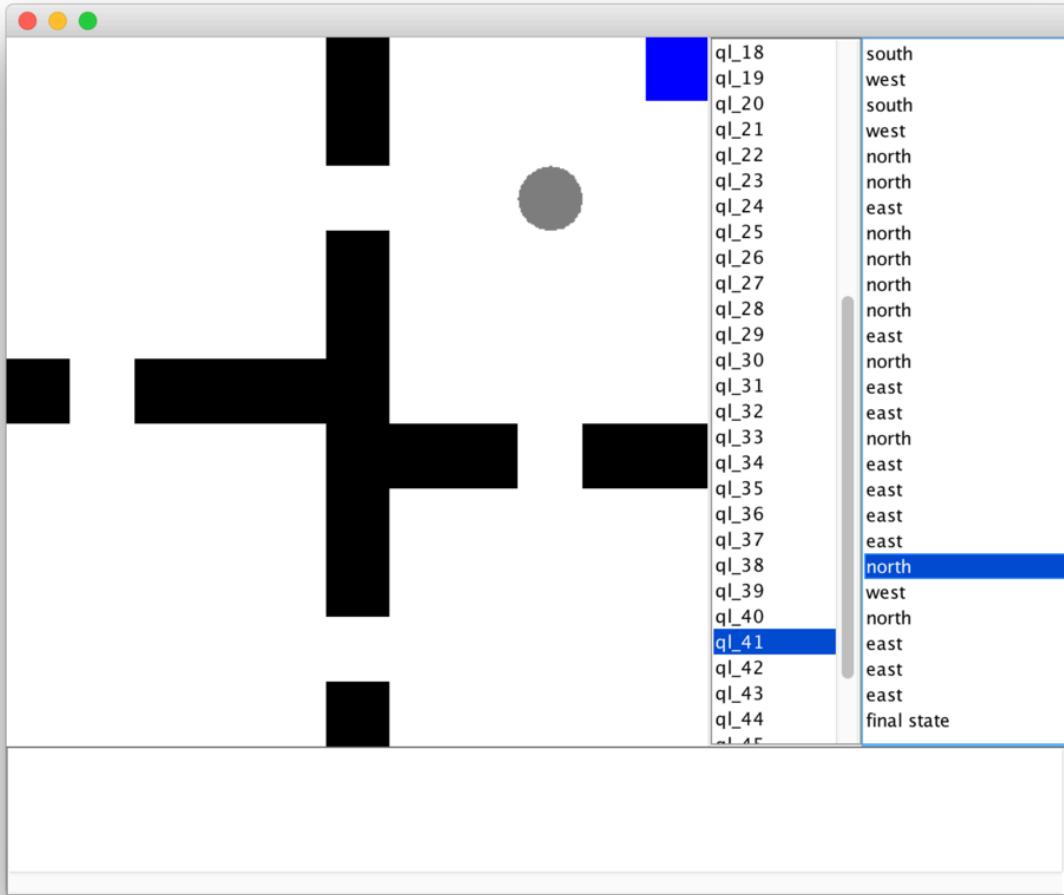


Agent learns optimal policy from interactions with the environment (s, a, r, s')



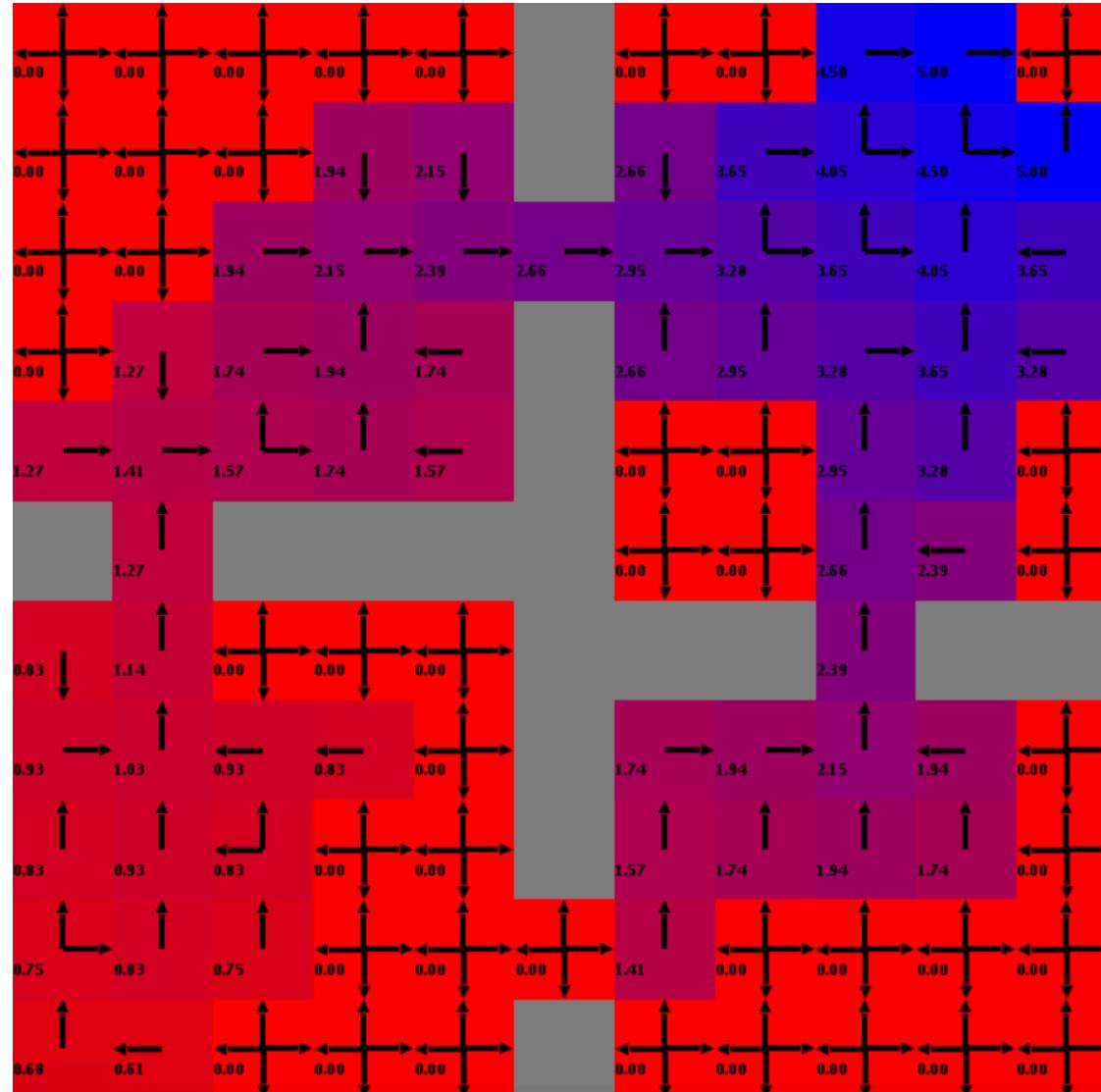
Source: http://www.mdpi.com/sensors/sensors-15-06668/article_deploy/html/images/sensors-15-06668-g002-1024.png

Visualizing training episodes



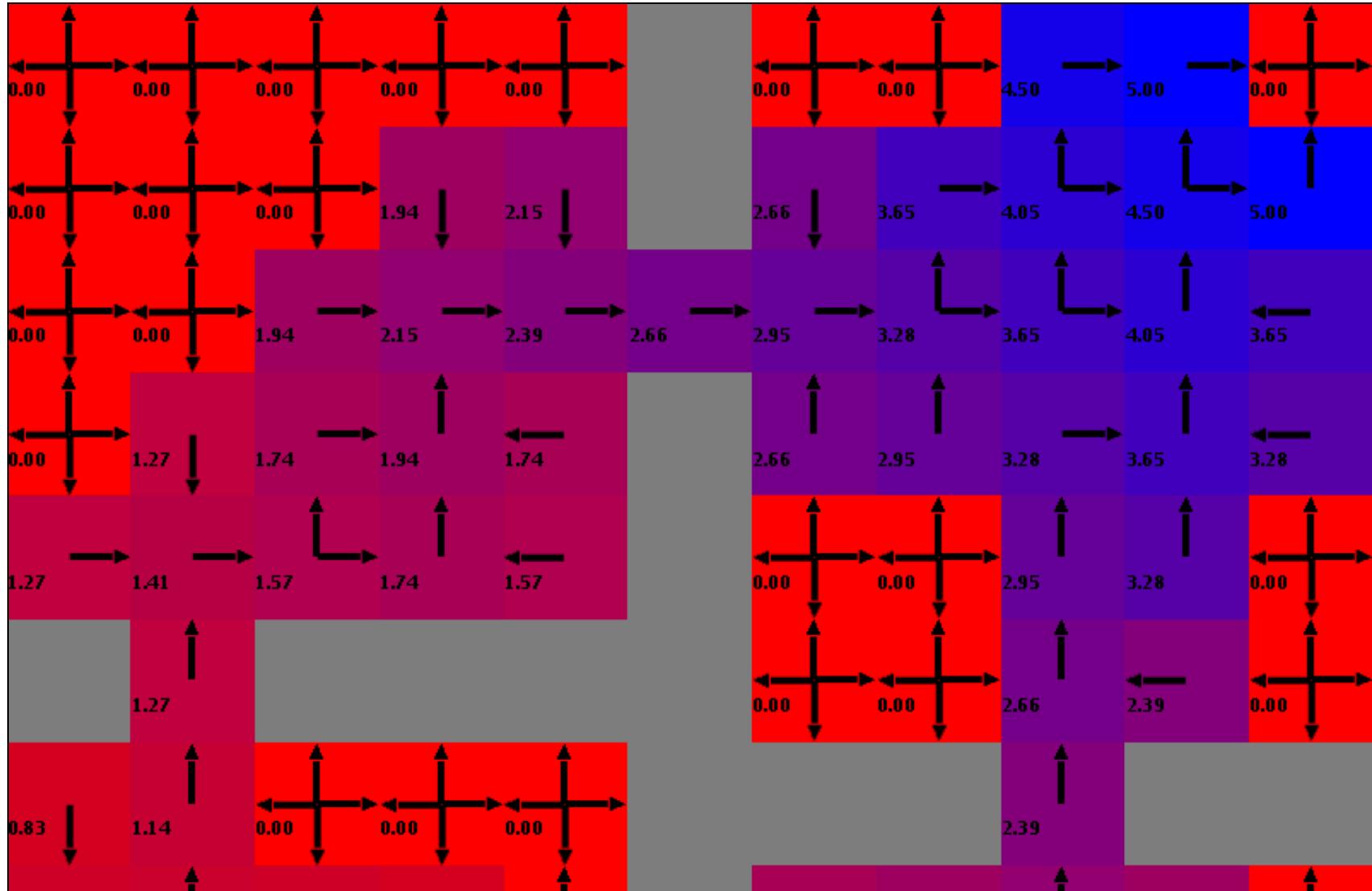
From BasicBehavior example in
https://github.com/jmacglashan/burlap_examples

Expected future discounted rewards, and polices



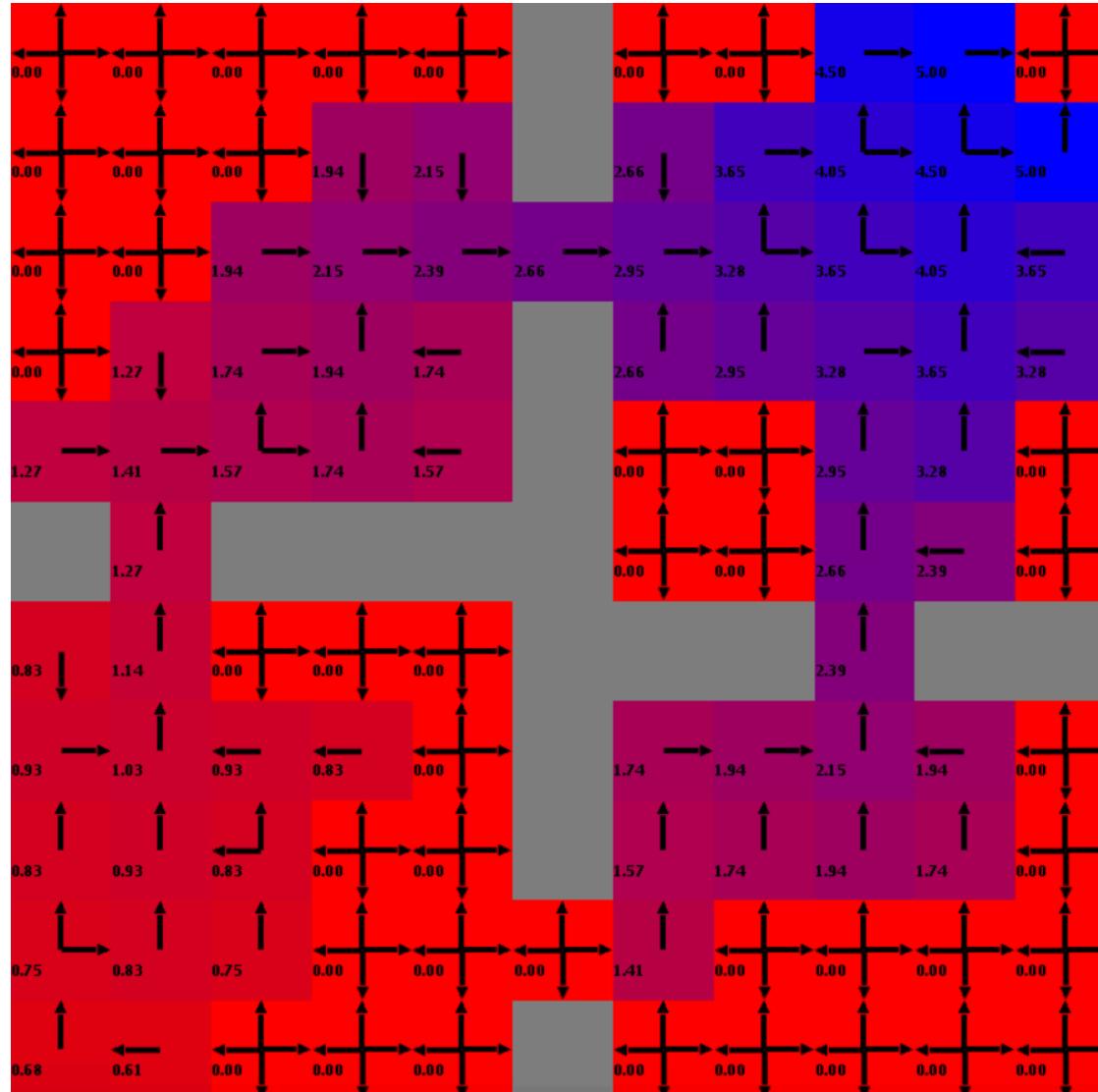
This example used discount factor 0.9

Low discount factors cause agent to prefer immediate rewards



Exploration vs. Exploitation

How often should the agent try new paths vs. greedily taking known paths?



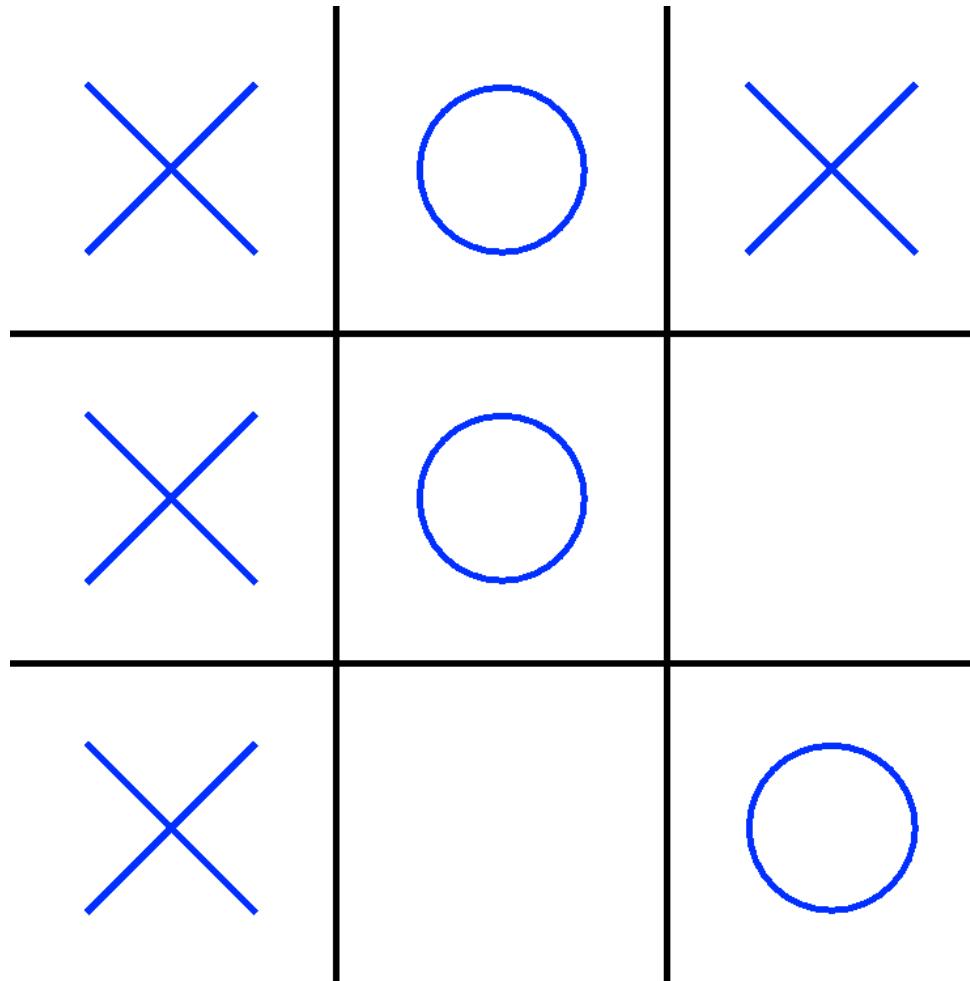
Q-Learning approach to reinforcement learning

		Actions			
		Left	Right	Up	Down
States	...				
	2, 7	2.65	4.05	0.00	3.20
	2, 8	3.65	4.50	4.50	3.65
	2, 9	4.05	5.00	5.00	4.05
	2, 10	4.50	4.50	5.00	3.65
	...				

Q-Learning table of expected values (cumulative discounted rewards) as a result of taking an action from a state and following an optimal policy. Here's an explanation of how calculations in a Q-Learning table are performed.

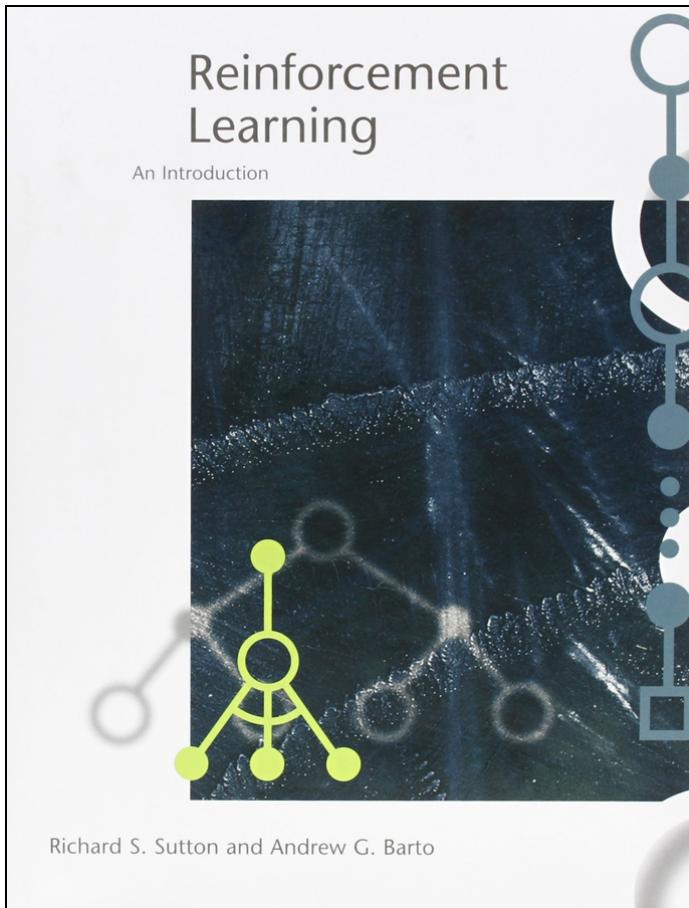
Tic-Tac-Toe with Reinforcement Learning

Learning to win from experience rather than by being trained



Inspired by the Tic-Tac-Toe Example section...

...of Reinforcement Learning: An Introduction



Reinforcement Learning: An Introduction

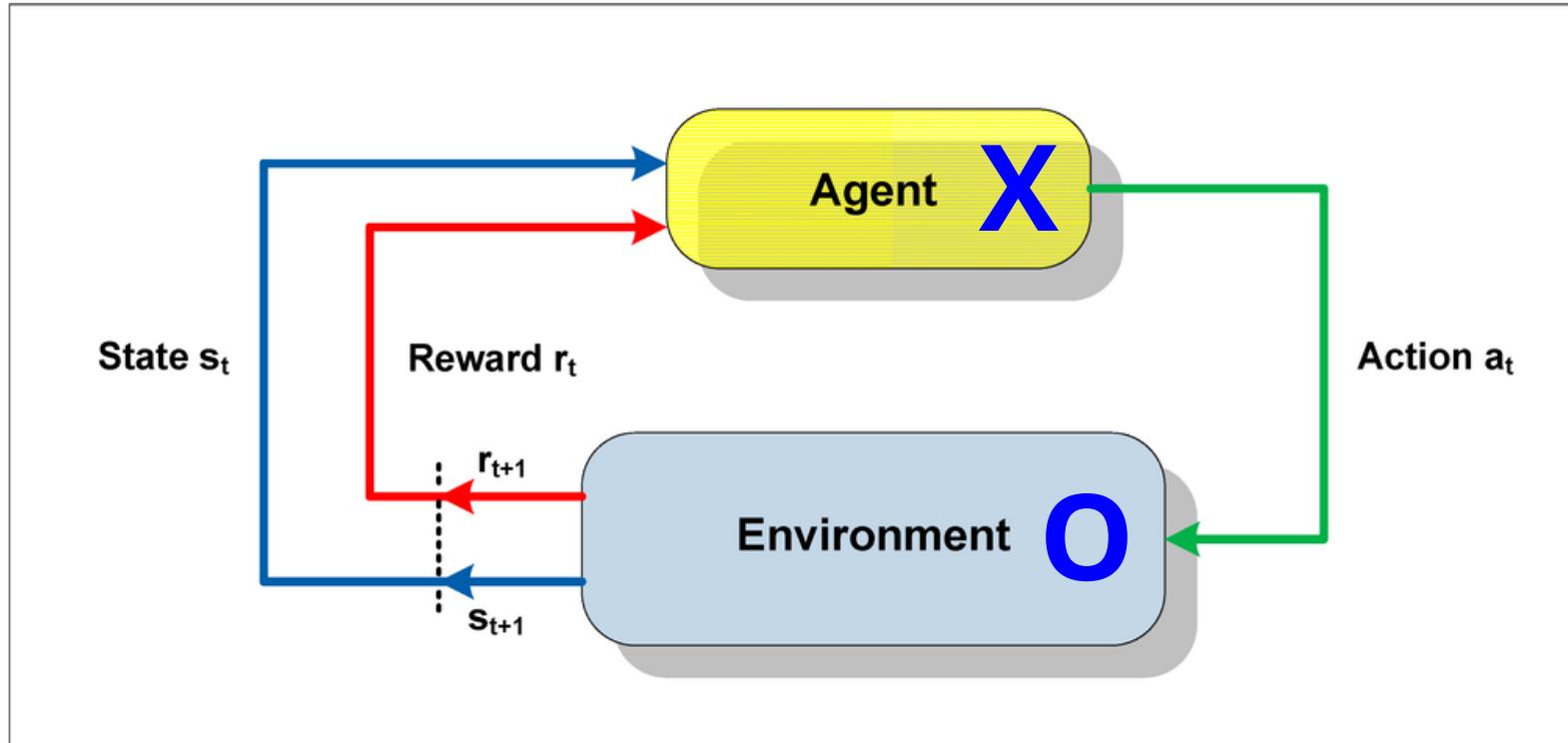
Second edition, in progress

****Draft****

Richard S. Sutton and Andrew G. Barto
© 2014, 2015, 2016

Tic-Tac-Toe Learning Agent and Environment

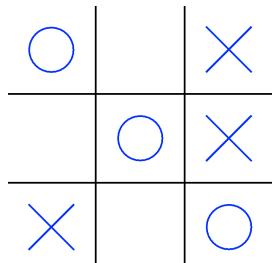
Our learning agent is the "X" player, receiving +5 for winning, -5 for losing, and -1 for each turn



The "O" player is part of the Environment. State and reward updates that it gives the Agent consider the "O" play.

Tic-Tac-Toe state is the game board and status

Unoccupied cell represented with an I in the **States** column



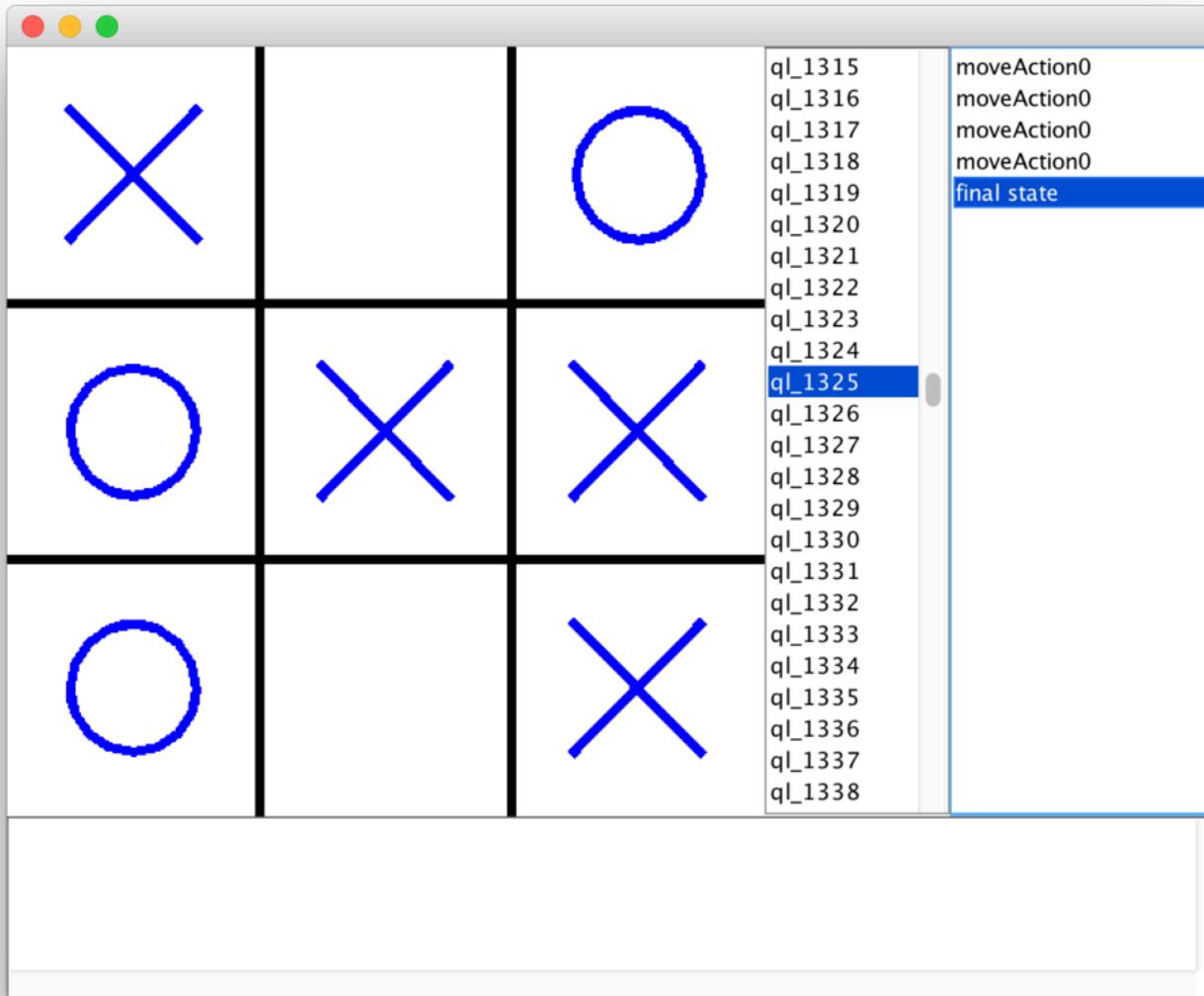
Actions (Possible cells to play)

States	0	1	2	3	4	5	6	7	8
OIXIOXXIO, O won	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
IIIIIIIOIX, in prog	1.24	1.54	2.13	3.14	2.23	3.32	N/A	1.45	N/A
IIOIIXOIX, in prog	2.34	1.23	N/A	0.12	2.45	N/A	N/A	2.64	N/A
IIOOXXOIX, in prog	+4.0	-6.0	N/A	N/A	N/A	N/A	N/A	-6.0	N/A
XIOIIXOIX, X won	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
...	<i>Q-Learning table of expected values (cumulative discounted rewards) as a result of taking an action from a state and following an optimal policy</i>								

Q-Learning table of expected values (cumulative discounted rewards) as a result of taking an action from a state and following an optimal policy

Tic-Tac-Toe with Reinforcement Learning

<https://github.com/JavaFXpert/tic-tac-toe-rl>



Through the Eyes of a Self-Driving Tesla



Click 'Rate Session'

Rate **10** sessions to get the supercool GOTO reward

Summary of links

Andrew Ng video:

<https://www.coursera.org/learn/machine-learning/lecture/zcAuT/welcome-to-machine-learning>

Iris flower dataset:

https://en.wikipedia.org/wiki/Iris_flower_data_set

Visual neural net server:

<http://github.com/JavaFXpert/visual-neural-net-server>

Visual neural net client:

<http://github.com/JavaFXpert/ng2-spring-websocket-client>

Deep Learning for Java: <http://deeplearning4j.org>

Spring initializr: <http://start.spring.io>

A.I Duet application: <http://aiexperiments.withgoogle.com/ai-duet/view/>

Self driving car video: <https://vimeo.com/192179727>

Machine Learning

EXPOSED!

The Fundamentals



James L. Weaver
Developer Advocate
Pivotal[™]
jweaver@pivotal.io
JavaFXpert.com
[@JavaFXpert](https://twitter.com/JavaFXpert)

