
Real-Time Machine Learning with Wekinator

IDE1

March 2019

Becky Stewart

r.stewart@imperial.ac.uk

Aims for Today

- Think about real-time contexts for machine learning.
- Learn how to use Wekinator for real-time machine learning.
- Introduction to using OSC to pass messages between applications.

The Format for Today

3 x

15-30 minutes of lecture to introduce a new concept and demo how to work with it.

15-30 minutes to work through exercises in small groups or individually.

Plus I'll be sticking around until 5pm to answer any questions.

Please ask many questions during the lectures and when working through the exercises.

What is Wekinator?

Wekinator

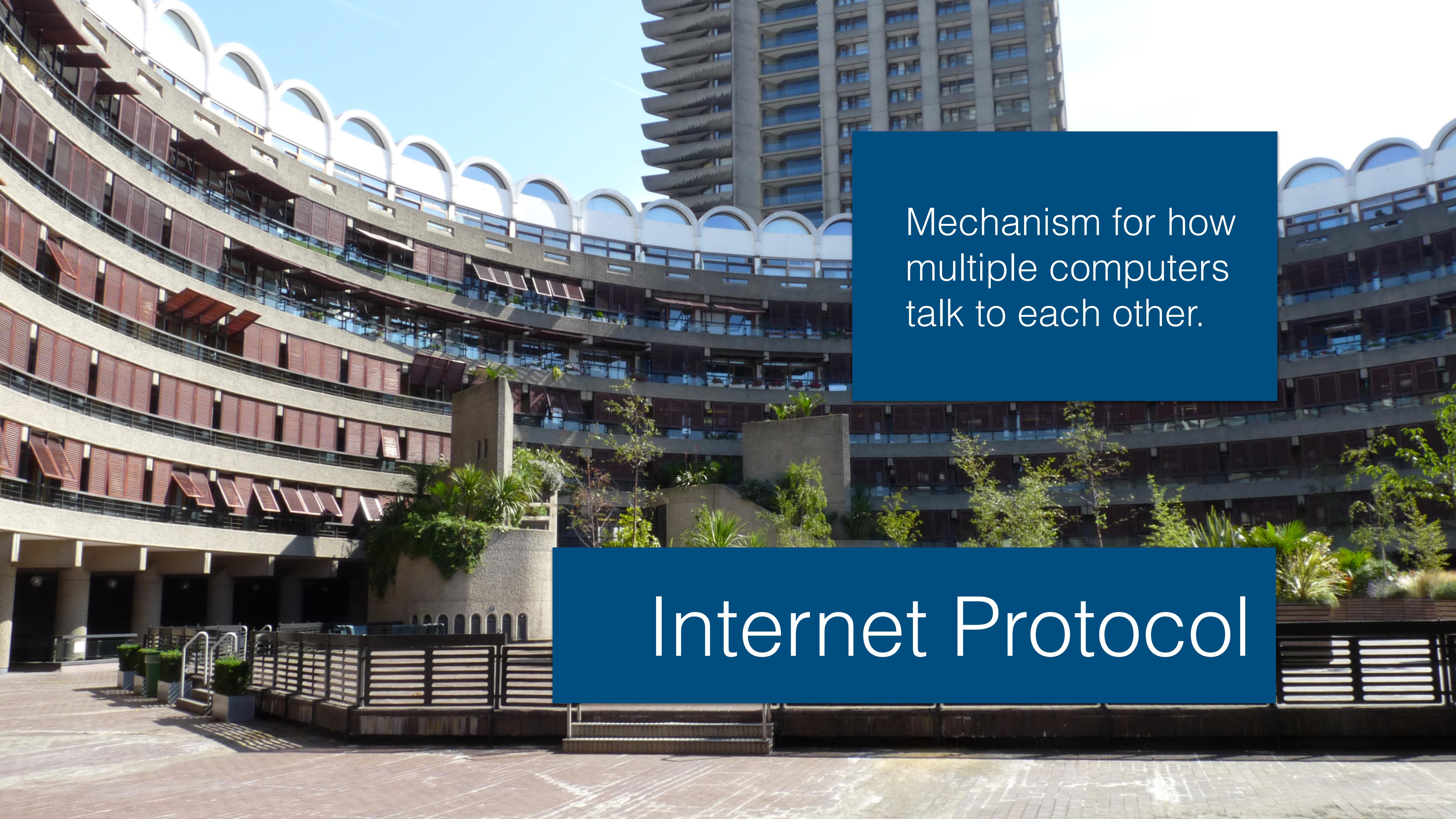


- Interface to easily use the machine learning library Weka.
- Send it OSC messages to train a model.
- Run a model while sending a stream of OSC messages, and it will output OSC messages according to your model.
- Either output continuous data or a classification label.

OSC

OSC Messages

- Open Sound Control - more details at <http://opensoundcontrol.org/introduction-osc>
- Designed by Adrian Freed and Matt Wright at CNMAT (Berkeley)
- Does not need to be related to sound or audio at all.
- Can be seen as next generation of MIDI.
- Protocol or format for how to send messages between applications running on different computers or on the same computer.
- Uses much of the same underlying tech that makes the internet work.



Internet Protocol

Mechanism for how
multiple computers
talk to each other.

The background image shows a modern apartment complex with several buildings featuring multiple levels and prominent arched balconies. The architecture is a mix of concrete and glass, with some greenery visible on the upper levels. The sky is clear and blue.

Each computer has an address,
like an apartment building.

Here it's the IP address.

A solid blue rectangular box is positioned in the lower right quadrant of the slide. It contains the text "Internet Protocol" in a large, white, sans-serif font.

Internet Protocol



Within the computer there are ports where messages can be sent.

Ports are like apartments.

Internet Protocol

Microsoft Windows [Version 6.2.9200]

(c) 2012 Microsoft Corporation. All rights reserved.

C:\Windows\system32> Telnet 127.0.0.1

There's no place like
127.0.0.1

Microsoft Windows [Version 6.2.9200]

(c) 2012 Microsoft Corporation. All rights reserved.

C:\Windows\system32> Telnet 127.0.0.1

There's no

127.0.0.1

The computer gives itself
the port 127.0.0.1 or
'localhost' if you want to
send and receive internal
messages.

Internet Protocol



TCP and UDP are two different methods in which messages can be packaged up and sent.

TCP (Transport Control Protocol) is slower than UDP, but has more guarantees. Like a signed-for package.

UDP (User Datagram Protocol) is faster, but you can't guarantee your message will arrive.

Sending Messages

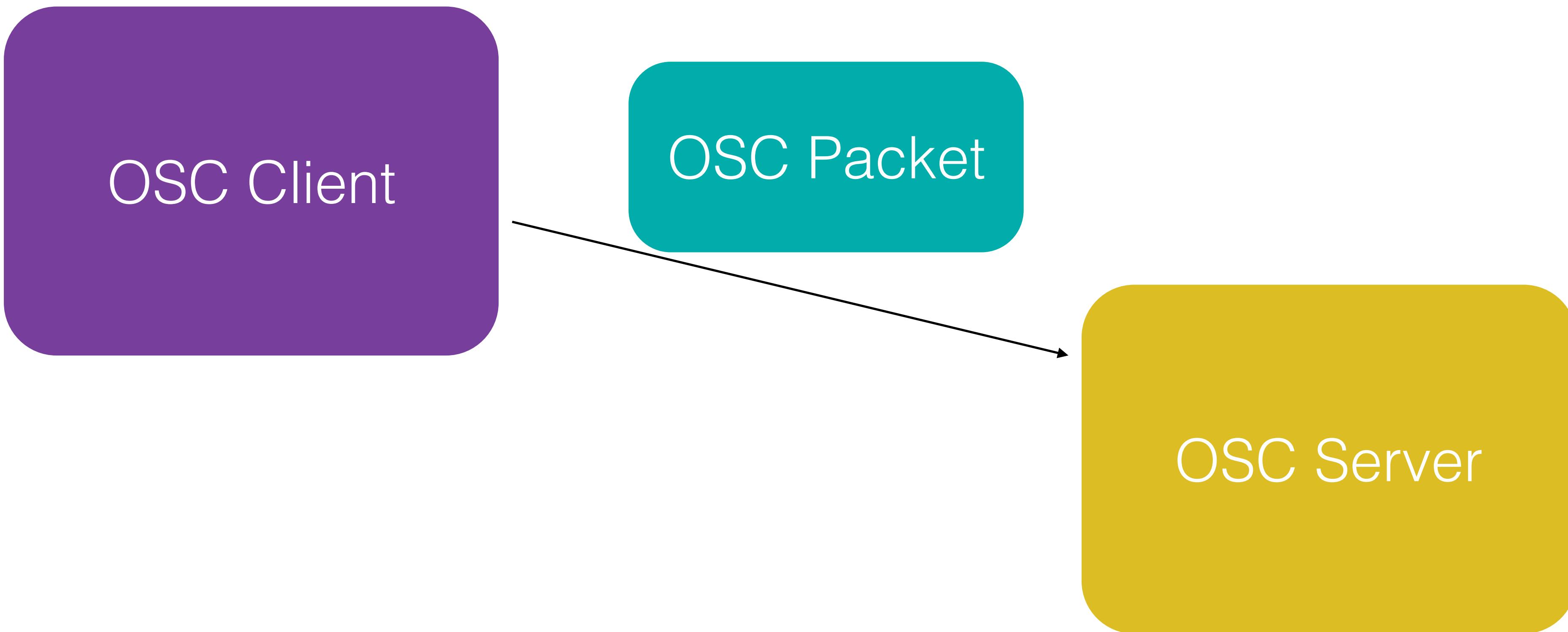


Open Sound Control are
messages packaged and
sent via UDP or TCP.

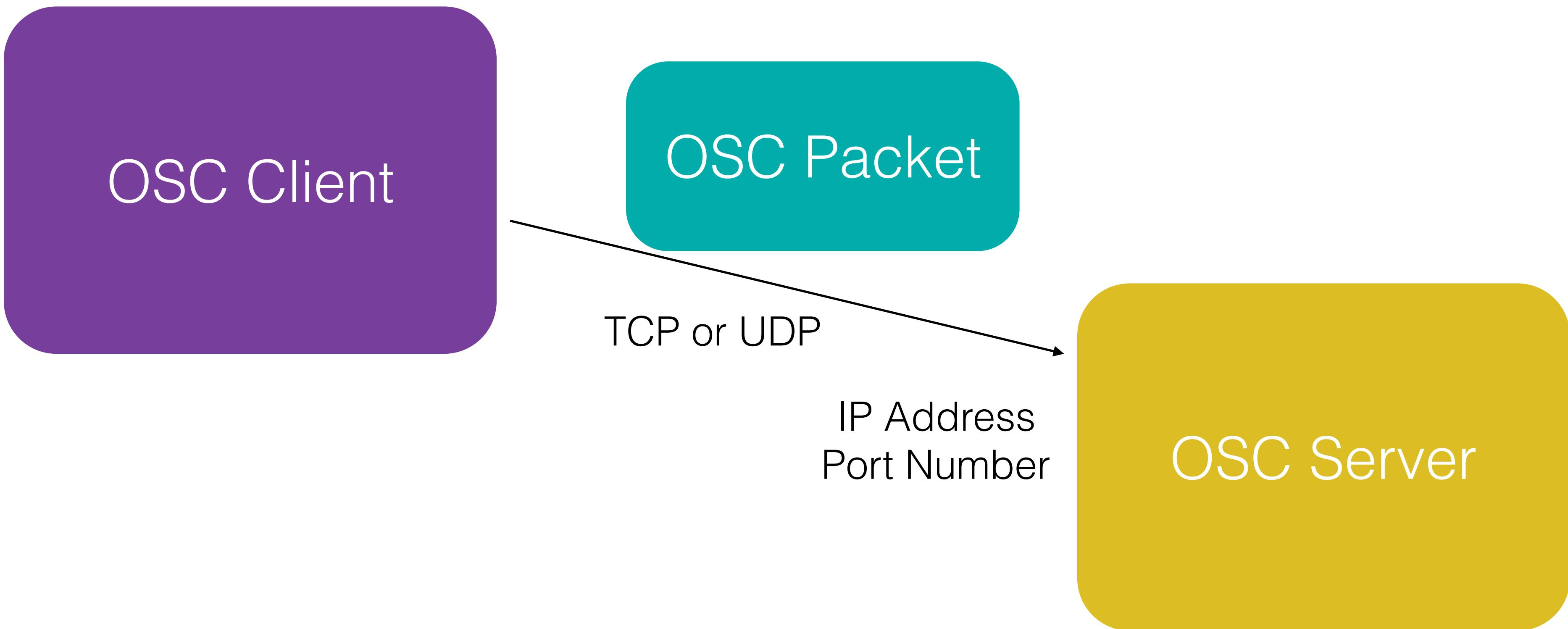
UDP more commonly used.



OSC



OSC



OSC

OSC Packet

OSC Message

Address Pattern

- starts with /
- e.g. /sensor/l dr/

Arguments

- the data being sent

Type Tag

- i - int
- s - string
- f - float
- b - blob

OSC

Message Example

/mycategory/myname

iif

0

1

3.14

Path

Typetag: 2 ints and 1 float

Data

OSC

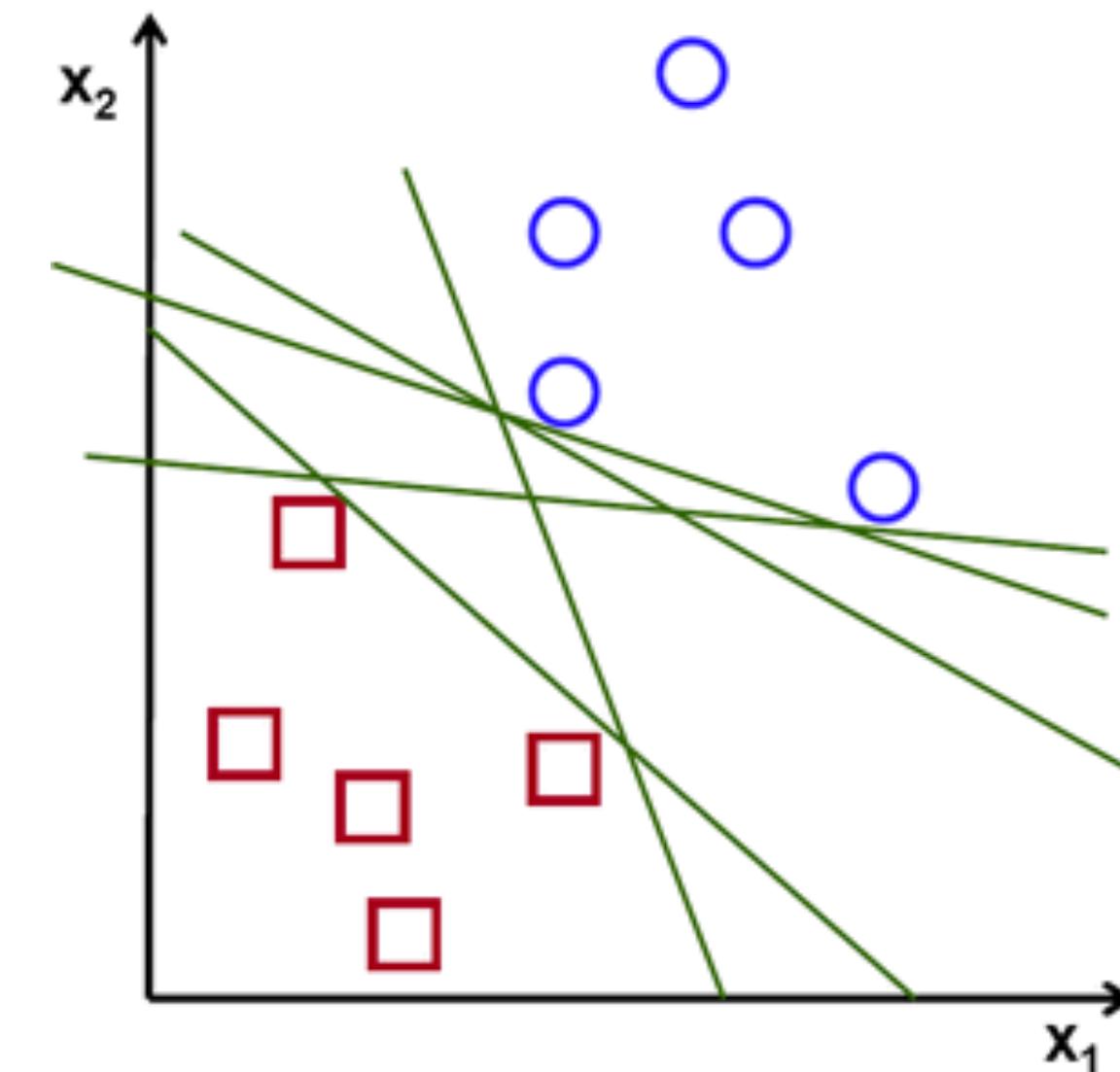
OSC Task

1. Download Processing from <http://processing.org>
2. Open Processing and install the OSC library (we will do this together).
3. Open the Processing sketches oscP5A.pde and oscP5B.pde.
4. A sends a message to itself when the mouse is clicked in the application window. B will print any message received. Change the code of each so that the message from A is printed in the B application. (Hint, think about the addresses.)

Classifier Output

Classifiers

- Useful to think about a case with two variables with a single labelled output.
- Trying to find the line that best separates the different labels.



Classifier Task

1. Create a Wekinator mapping that reads in the mouse X and Y position from the Processing sketch `send_mouse_position.pde` and classifies 5 different areas in the window.

2. Train the Wekinator model.

3. Run the model and show the determined class in the Processing sketch `receive_int.pde`.

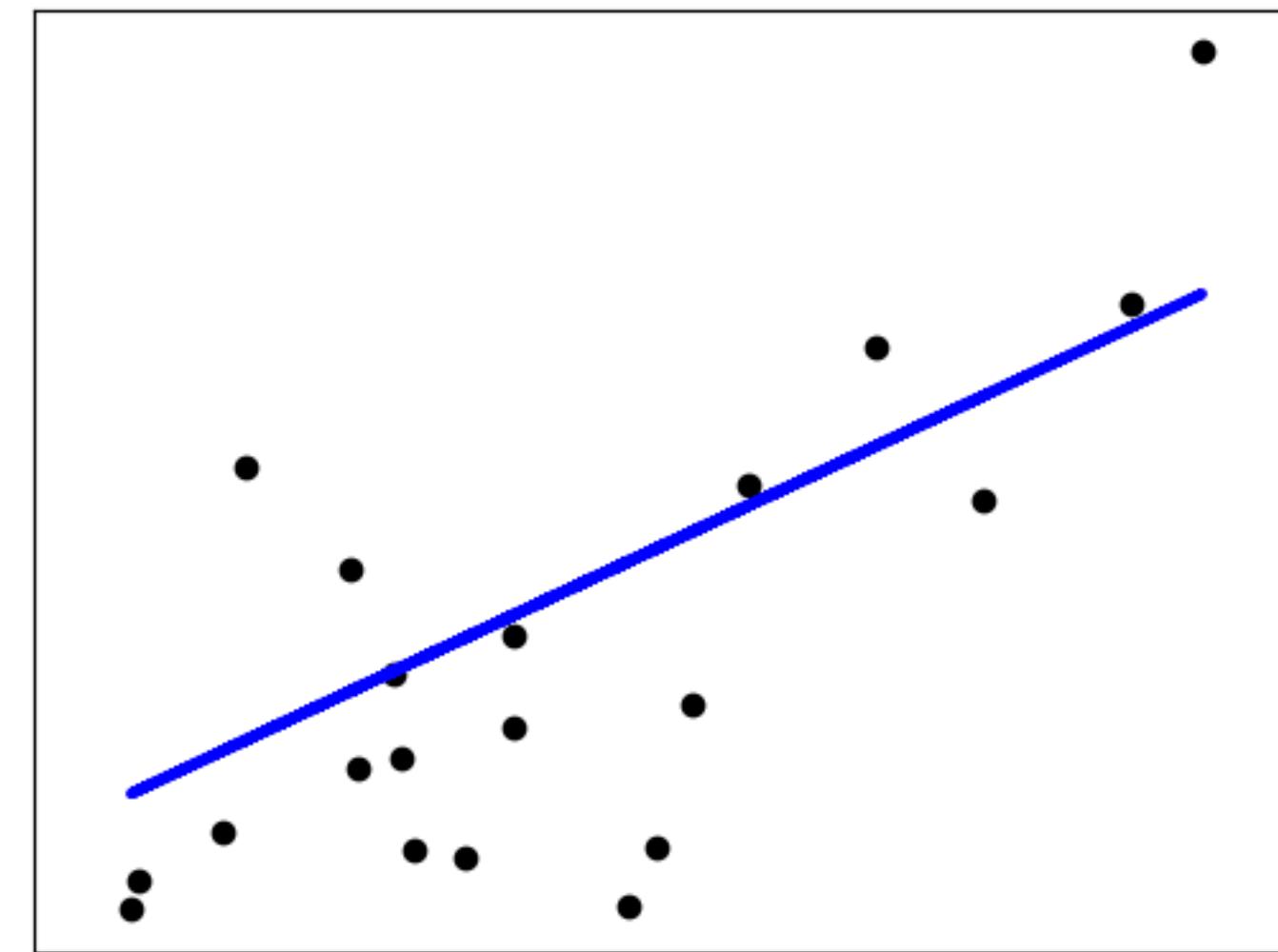
Further Considerations

- What is the input data?
- How would mouse position vs mouse speed vs mouse acceleration change the model?
- Haven't discussed how to evaluate if your model is doing a good job.

Continuous Output

Linear Regression

- Assuming our data can be described by a straight line.
- Looking only at one input variable and one output.
- Often lines are described by the equation $y = mx + b$



Linear Regression

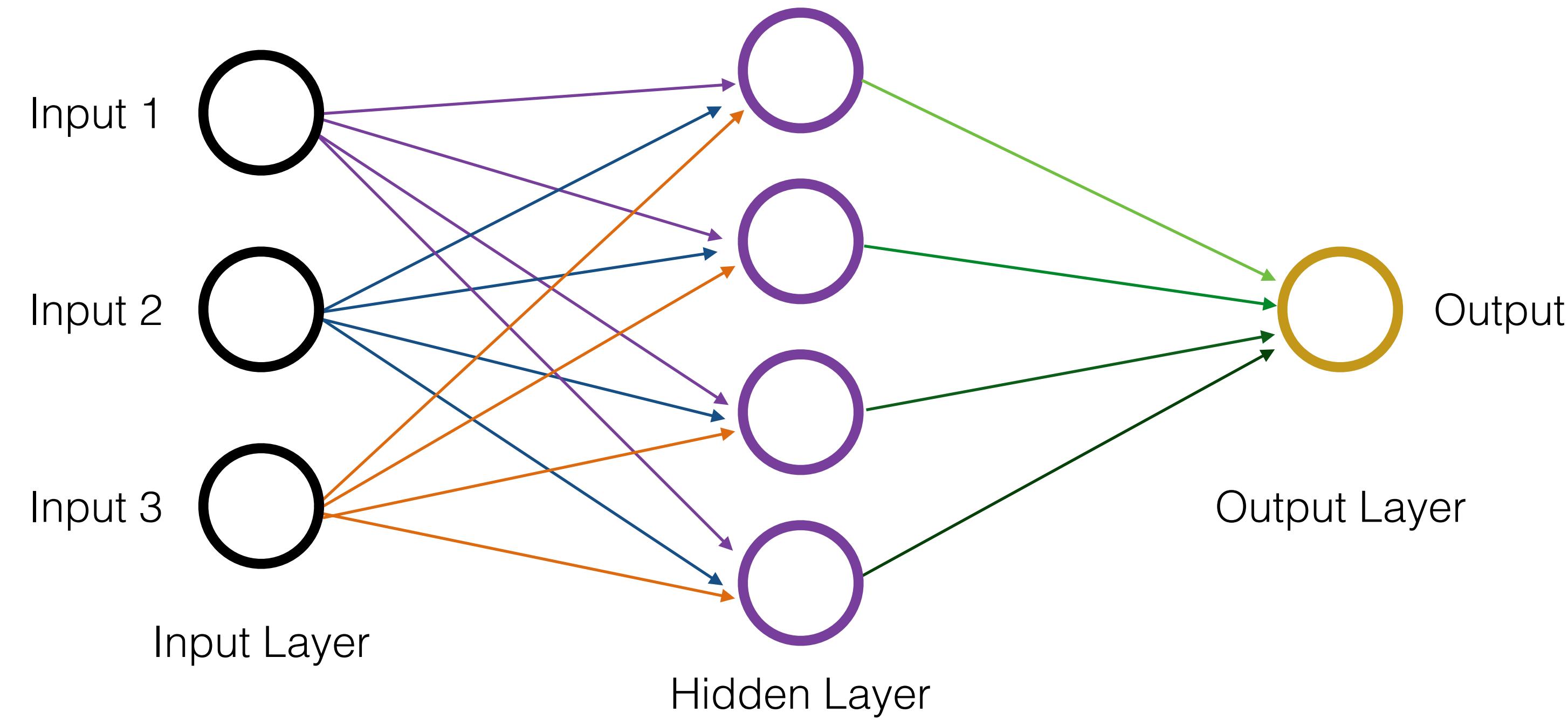
What we can do with linear regression:

- Plot already existing data
 - Fit a line using a chosen method
 - Use the line to predict the output of new data
 - Calculate the error between line and the data
- Train a model
- Run a model
- Evaluate a model

(Artificial) Neural Networks

Another approach for looking at existing continuous data and using it to predict future data (**supervised learning**).

Especially useful for large numbers of variables (**dimensions**).



Continuous Output Task

1. Explore the available example code at <http://www.wekinator.org/examples/>
2. Choose a continuous input (can just use previous mouse example) and continuous output.
3. Create a Wekinator mapping between the two.

ML as Artistic Tool

Machine learning can be a powerful and expressive tool for creating artwork. These examples are active research projects, certainly beyond the scope of Wekinator!

- Memo Akten - Learning to See: <https://vimeo.com/260612034>
- Bob Sturm - The Machine Folk Sessions: <https://themachinefolksession.org/tune/351>

Additional Reading

Anatomy of an AI System by Kate Crawford and Vladan Joler <https://anatomyof.ai/>

Radical Technologies by Adam Greenfield <https://www.versobooks.com/books/2742-radical-technologies>

New Dark Age by James Bridle <https://www.versobooks.com/books/2698-new-dark-age>

People's Councils for Ethical Machine Learning by Dan McQuillan http://research.gold.ac.uk/23040/1/people_councils_for_ethical_machine_learning.danmcquillan.author-accepted.pdf

Data Science as Machinic Neoplatonism by Dan McQuillan <https://link.springer.com/article/10.1007%2Fs13347-017-0273-3>

Algorithms of Oppression by Safiya Umoja Noble <https://nyupress.org/books/9781479837243/>