Valgfag: Introduktion til anvendt Machine Learning i webapplikationer

AI with:

TensorFlow.js

Machine learning in the browser
Day 2

Agenda (Part 1 + 2)

01 Tensorflow core concepts →

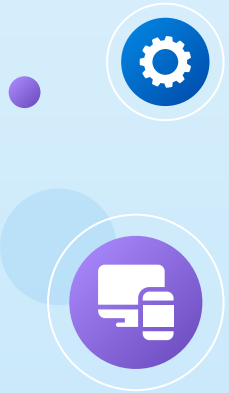
02 Garbage collection →

03 Lab 3.1 →

04 Data preparations →

05 Lab 4.1 →

06 Starting HomeWork (in groups)



Core Concepts

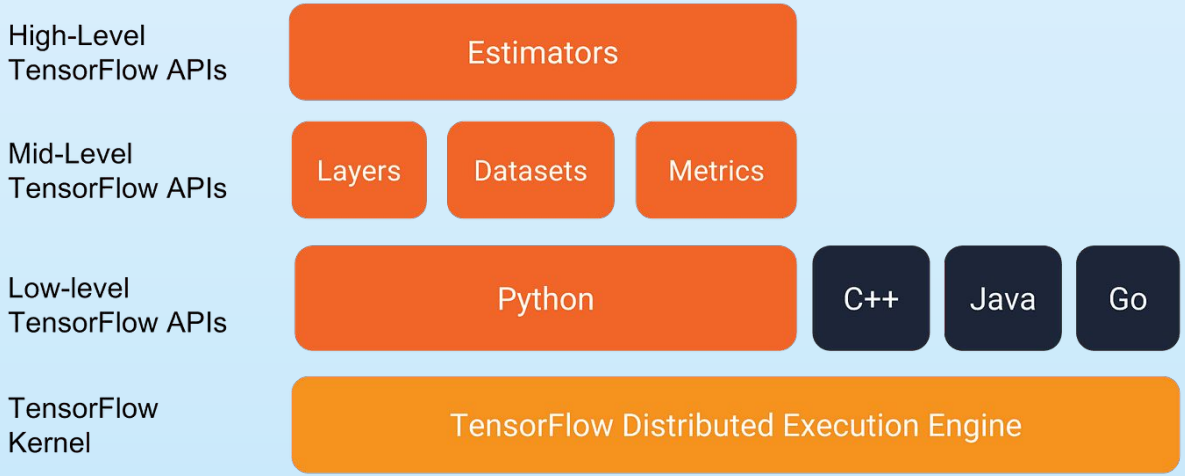
TensorflowJS APIs

- Low level: Operations OPS
- High level: Layers

TensorFlow.js

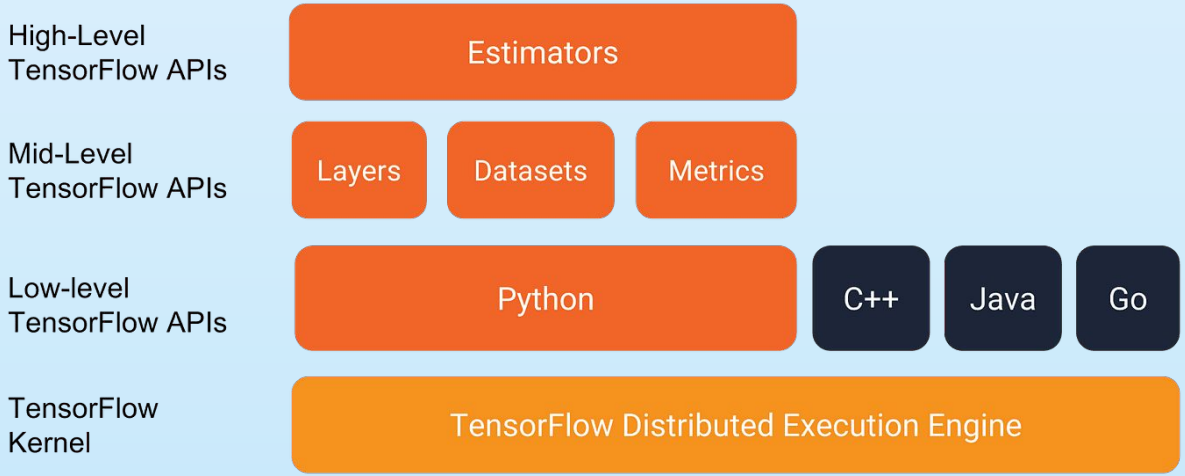
Low level part of Tensorflow API's

- The TensorflowJS api are similar the Python API
- We will use mathematical operations to manipulate data.
- Build learning models (the hard way)



High Level part og TensorFlow.js APIs

- Defining complex models easily
- Can be compared to Keras API in Python
- Used for Artificial Neural Networks

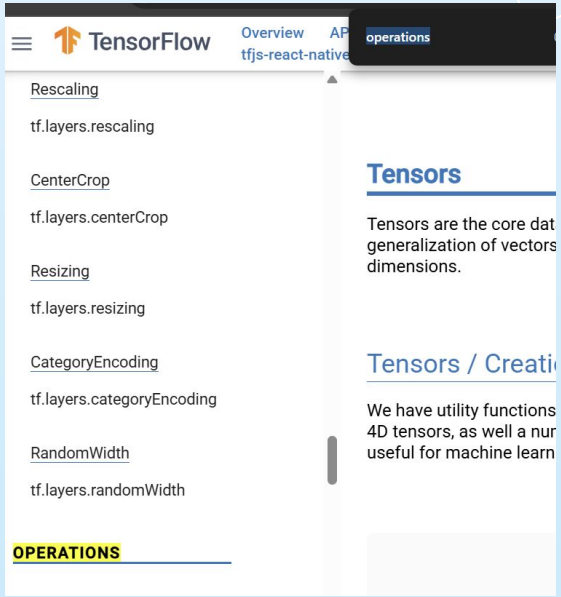


Demo

Go to: https://js.tensorflow.org/api/latest/

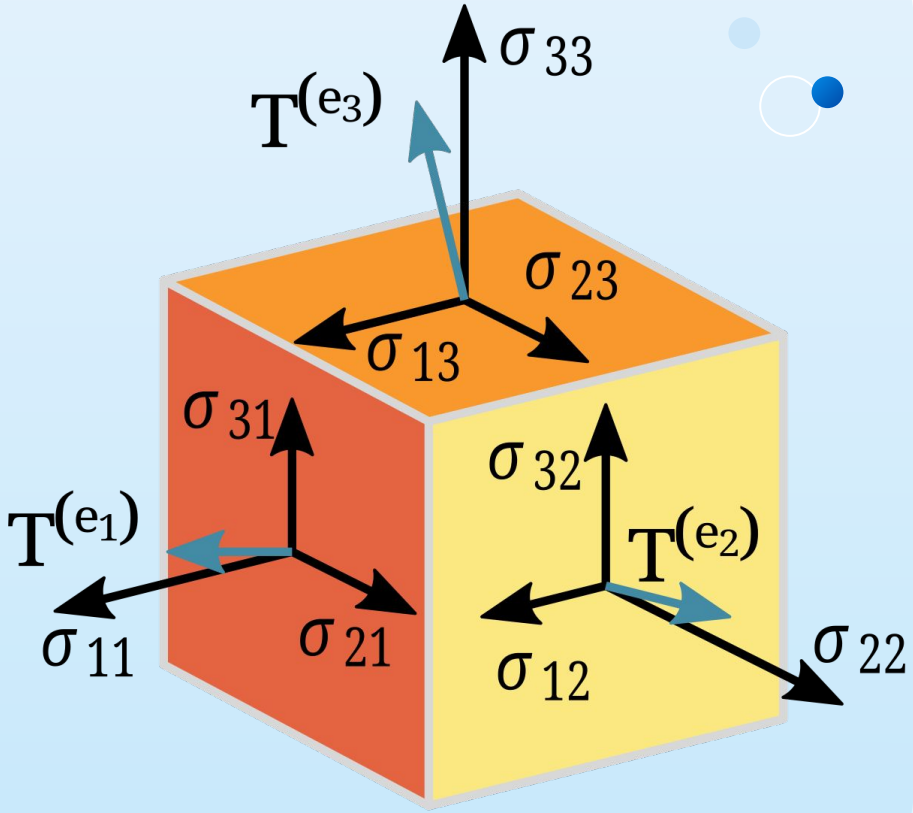
User browser search (ctrl/cmd +f) and search for: operations
Find the functions you can call under operations.

User browser search (ctrl/cmd +f) and search for: Layers
Find the functions you can call under Layers in the left side



What is a Tensor

A tensor is a mathematical, multidimensional array of numbers that generalizes scalars, vectors, and matrices to higher dimensions, acting as a container for data in \(N\)-dimensions.

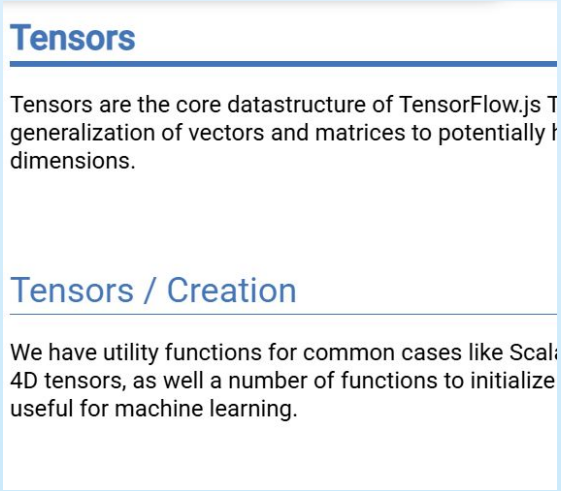


What we need to know as developer

We do NOT need to understand all the math!!!

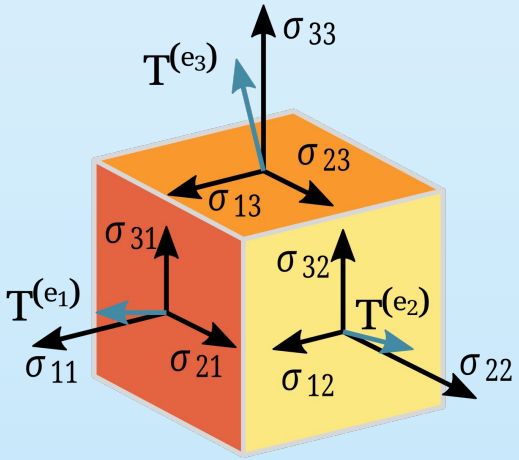
We need to understand:
What a tensor is in practical terms and how we can create a Tensor

How can we create a tensor:
https://js.tensorflow.org/api/latest/#Tensors



Defining a Tensor

- A list of values
- A data type int, float, string
- A number of dimensions
- A shape defines how the values are distributed through the system

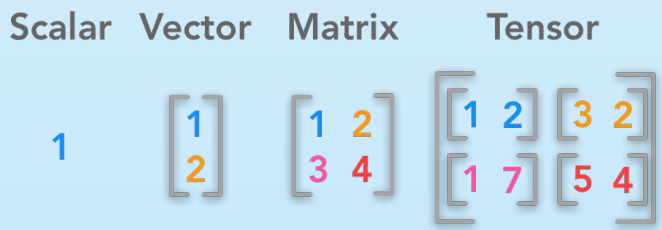


Scalar Tensor

- It holds a single value
- 0 dimensions aka rank 0

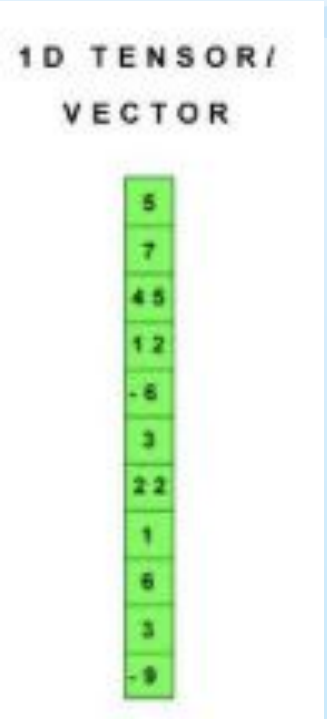
tf.scalar (32) = tf.tensor([32],[])

It is simpler and safer to use scaler so there is no misunderstanding about the rank we want here.



1D Tensor

- A list of values and positions
- tf.tensorId([4,7,45,....]) (recommended)
- tf.tensor([1,2,3],[3]) is 1d (Do not use this version)



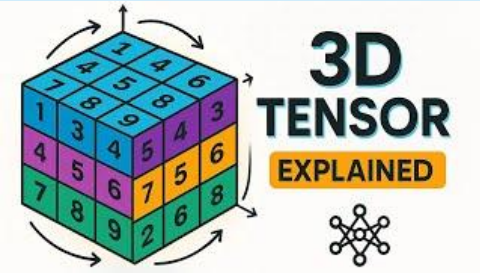
https://mylearningsinaiml.wordpress.com/tensorflow/tensors/

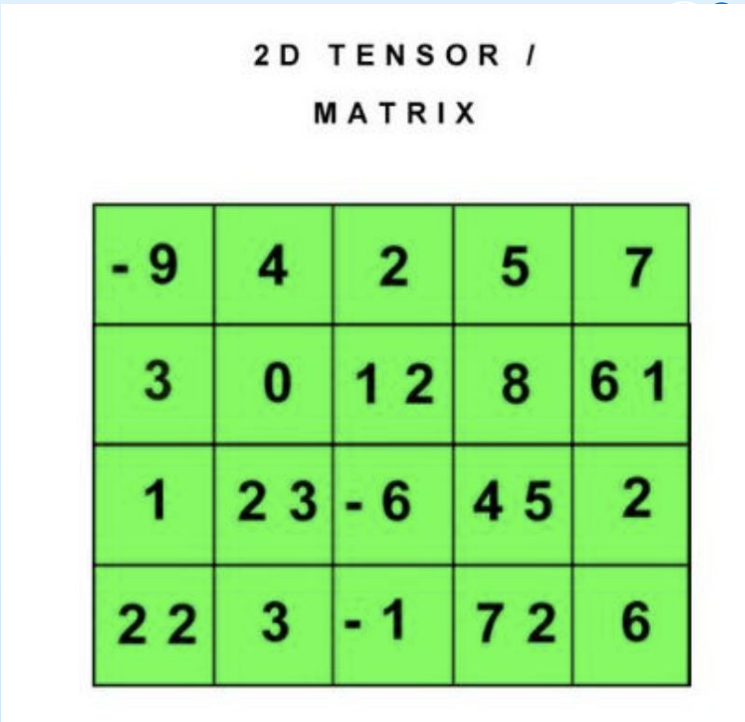
2D Tensor

- Array of arrays
- tf.tensor2d([[-9,4,2,5,7],[3,0,12,8,61],.....])

3d is a cube:

We will be working with 1-3 dimensions.



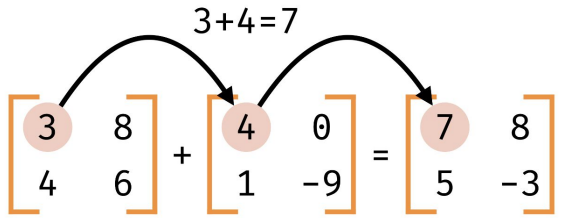


Tensor math operations

- Tf.add (a,b) simple
- Try to find some of the more adv. Operations

Why use math operations

- Prepare data for processing (normalisation)
- Build low level models with operations (to low level)
- Extend Tensorflow.js write optimisation



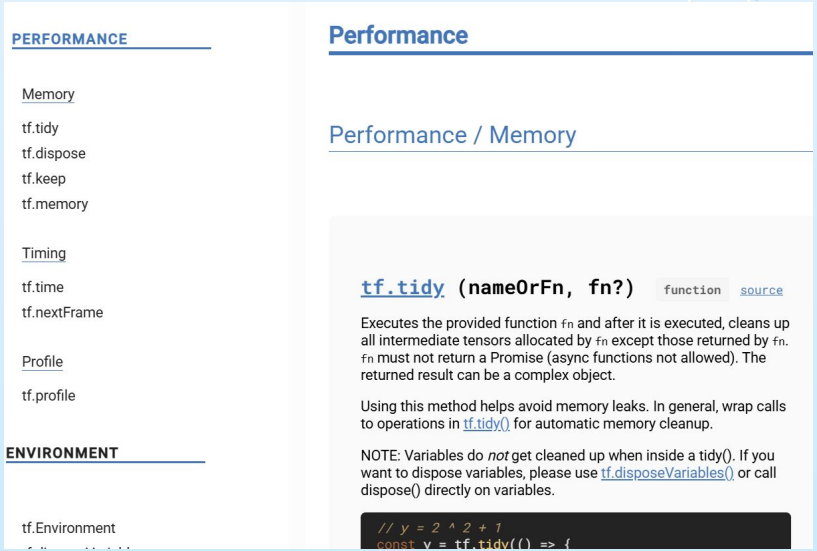
Memory management

In TensorFlow.

Tensors are stored in WebGL
Operations happen in WebGL

WebGL has no garbage collecting

Note that underlying GL object will be automatically marked for deletion when the JS object is destroyed, but can we wait for it?

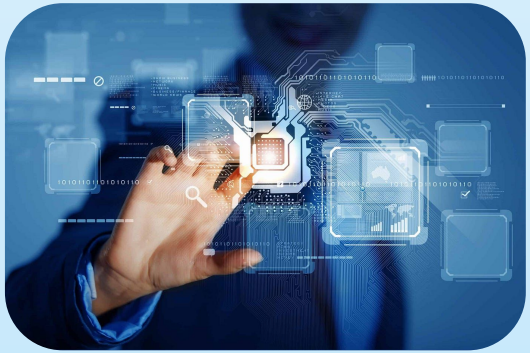


Memory management

Tf.memory : returns info about memory usage
Tf.dispose : clean memory from container

If we use tf.tidy we can better control when we want to clean all that are ready for cleaning
Tf.keep is used for keeping a bit for later use.

DEMO



What have we learned from part 1

- Getting to know TensorFlow.js APIs
- A tensor has values, dimensions, shape, data type
- TensorFlow.js provides math operations for tensors
- Optimised depending on backend, eg. WebGL
- Memory usage must be managed

Lab 3.1

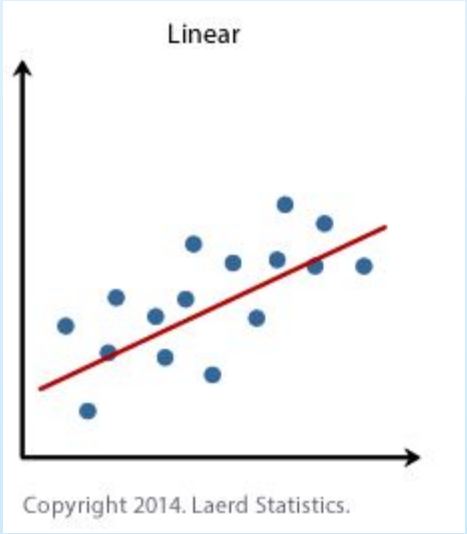
3.1 Tensor Math and Memory Management

Part 2:
Data preparations

- Import CSV
- Visualise data
- Features and Labels
- Normalising data
- Split tensor into sub tensors

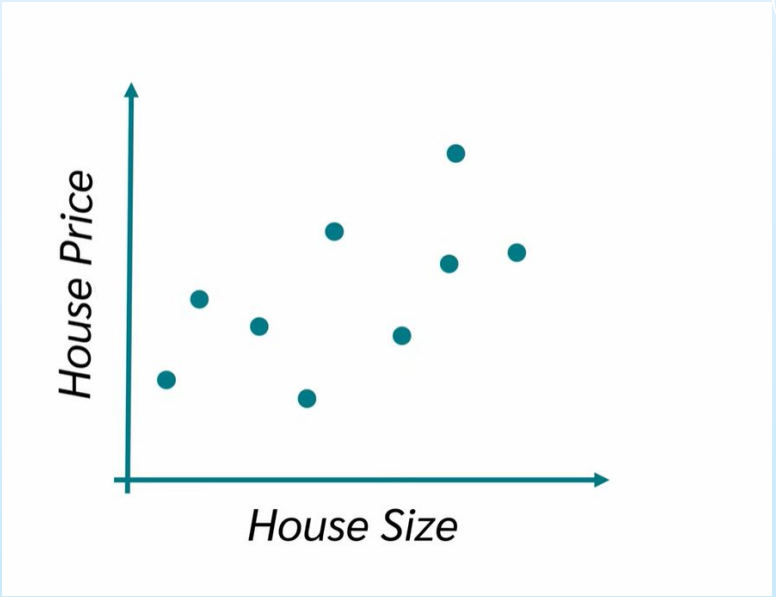
Linear regression

- Simple machine learning problem
- Plot a line to represent a correlation
- Make predictions for new data points



Linear regression

- Prepare the data
- Create a model
- Train the model
- Test the model
- Make predictions



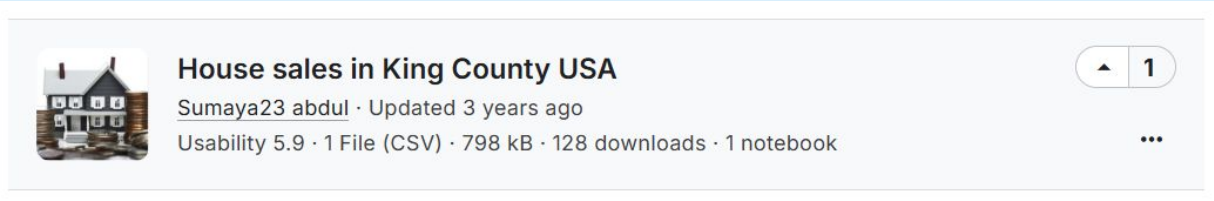
https://www.youtube.com/watch?v=T5AoqxQFkzY

Preparing the data

1. What is the problem we want to solve?

Go to https://www.kaggle.com/datasets

Find this



What data can we use to predict the sales price of a house? Keep it simple
If you can't find the data then look in itsLearning.

Explore the data

Open the csv and find the data we talked about before
Are there any problems with the data?

How much data are there? Is it enough?

Lets import using tf.data.csv – find the dokumentation to start with.

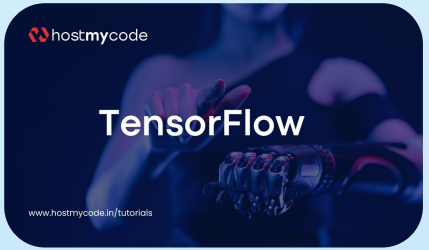
Hosting the csv

Setting up the web server is dependant on your OS.

Find out :) get advice from fellow student, use ai chat, or google a video on it, find more info on itsLearning…

When everyone in the group are ready let me know.

```
<script type=”text/javascript”>
        Const houseSalesDataset = tf.data.csv(“--URL to file–”)
</script>
```



Inspect

In the documentation, inspect the following:

Tf.data.csv

csvConfig

columnNames

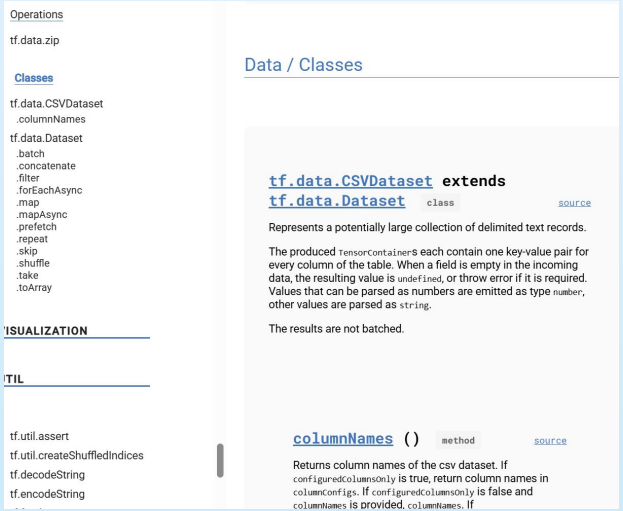
tf.data.CSVDataset (is returned)

Verify that you can read data

```
<script type="text/javascript">
        Const houseSalesDataset = tf.data.csv("--URL to file–")
        log.console(houseSalesDataset.take(10).toArray())

</script>
```

Why is this not working? What is promise??



Visualising data

Inspect tfjs-vis.
https://github.com/tensorflow/tfjs/tree/master/tfjs-vis

- tfvis.render.scatterplot

Visualization

tfjs-vis is a companion library for TensorFlow.js that provides in-browser visualization capabilities for training and understanding models. API docs for tfjs-vis are available here

Preparing features and labels

- Create a single array of x-values. (features) and place it in a 2d-tensor
- Create a single array of y-values (Label) and place it in a 2d-tensor
- Print to see result.

```
const featurevalues = await points.map(p => p.x).toArray();
const featureTensor = tf.tensor2d(featurevalues, [featurevalues.length, 1]);


const labelvalues = await points.map(p => p.y).toArray();
const labelTensor = tf.tensor2d(labelvalues, [labelvalues.length, 1]);

featureTensor.print();
labelTensor.print();
```

Label (output)

Feature (input)

Lab 4.1

4.1 Prepare the Data

Thank you

See you next time.

You find next time in timeedit.net search for course name or you initials to find your personal schema.