AI & Radio Astronomy

By: Thomas Merle, Tanya Ahn, Sungbin Park, Rémi Doguet and Marleen Jonker.

The team

Marleen JONKER, The Netherlands

Sungbin PARK, Republic of Korea

Thomas MERLE, France

Rémy DOGUET, France

Tanya AHN, Republic of Korea

Table of Contents

Our goal

1. To figure out whether a star has at least one Exoplanet in orbit by using ML.
2. Build a neural network to find pattern of exoplanet.
3. Use of the Dataset from kaggle

Personal Learning Objective

1. Tanya AHN: Learn more about the specific algorithms used in astronomy fields

2. Marleen JONKER: Learn more about Tensorflow and how to apply to machine learning.

3. Sungbin PARK: Learn how to install Tensorflow, ML librarys, Applying ML algorithm.

4. Remi DOGUET: Learn more about Neural Network and Machine Learning

5. Thomas MERLE: communication in an international environment, learn the basics of a new subject (machine learning)

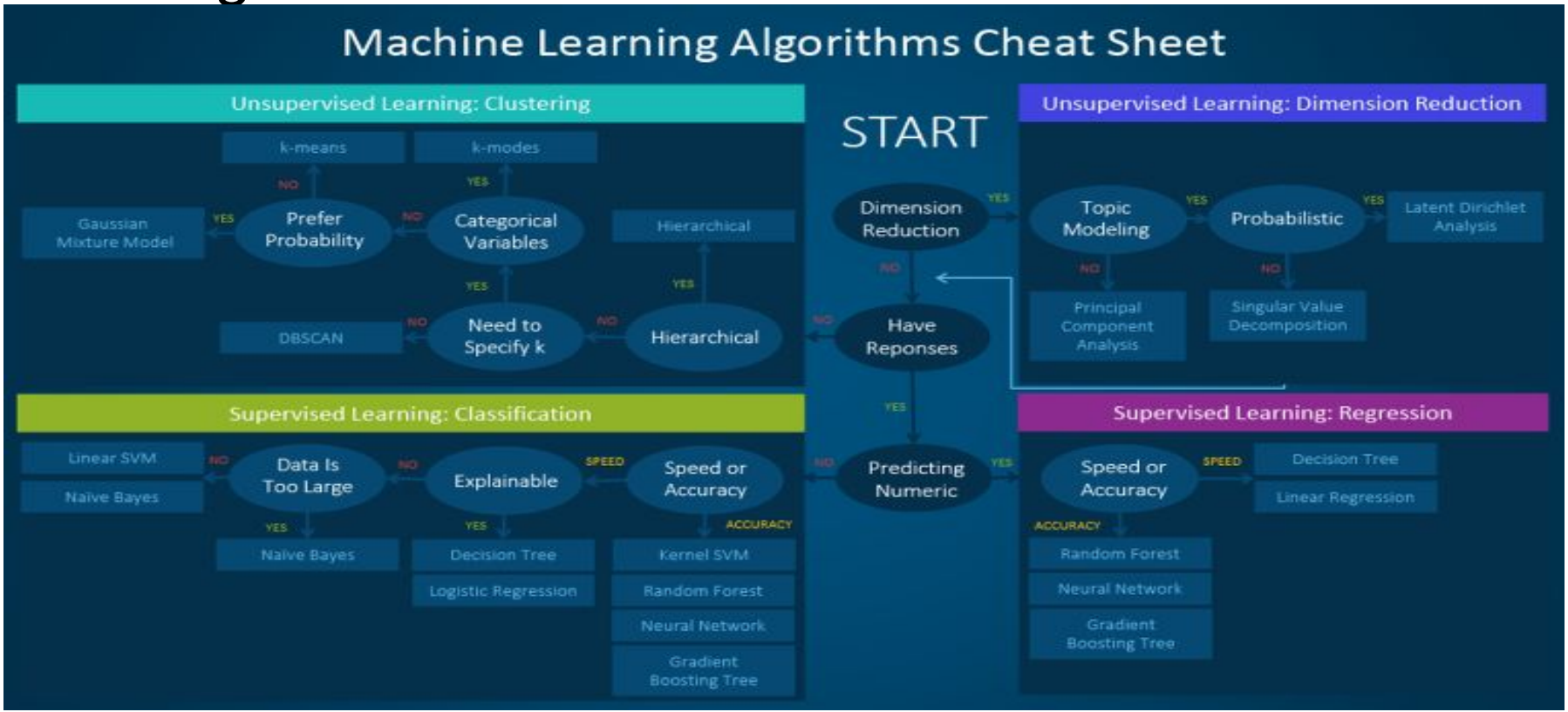
How we did it

1. Research - 1st Week
2. Plan - 1st Week ~ 2nd Week
3. Found free ML course to learn basic algorithm, Tensorflow and Keras. - 2nd Week
4. Using - 2nd Week

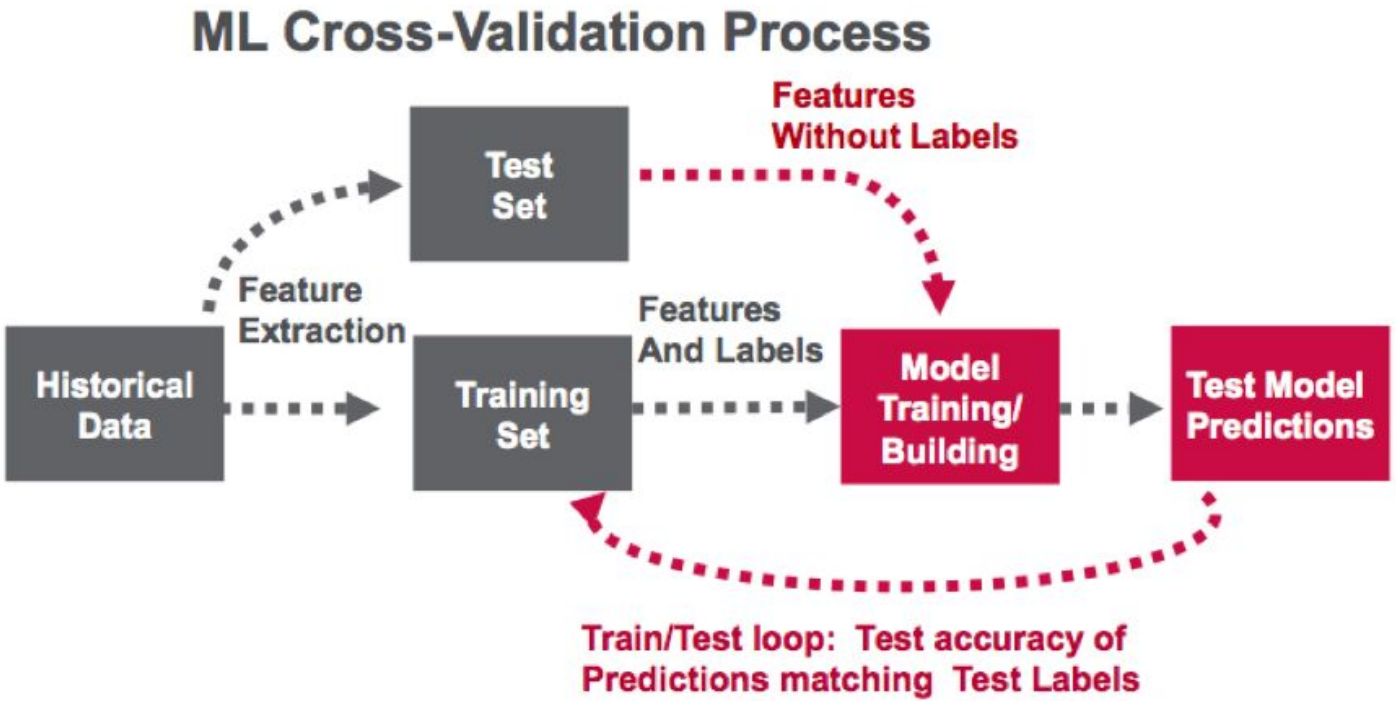
What we did

1. Learned ML by Demo, Example.
2. Built our Neural network using reLU, sigmoid.
3. Wrote a document about our work.
4. Distributed task each person.
5. Used Github to communicate efficiently.

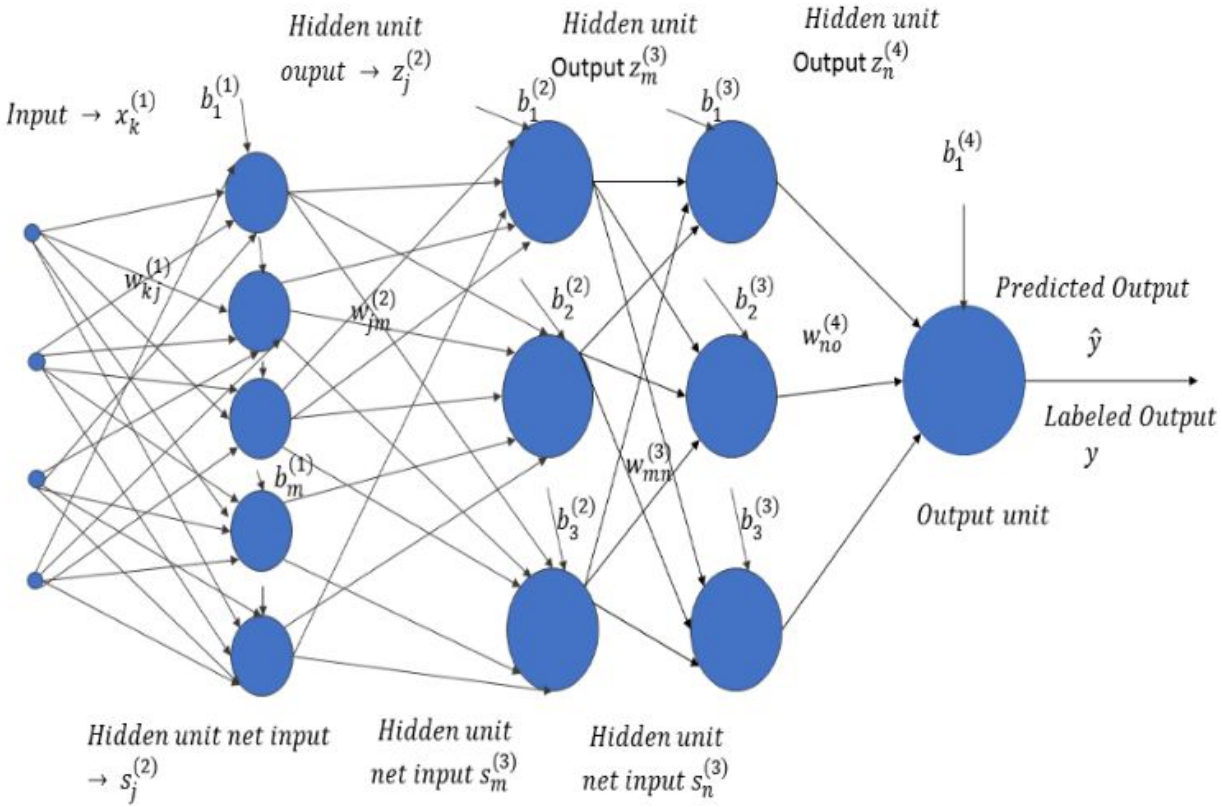
Building Neural network



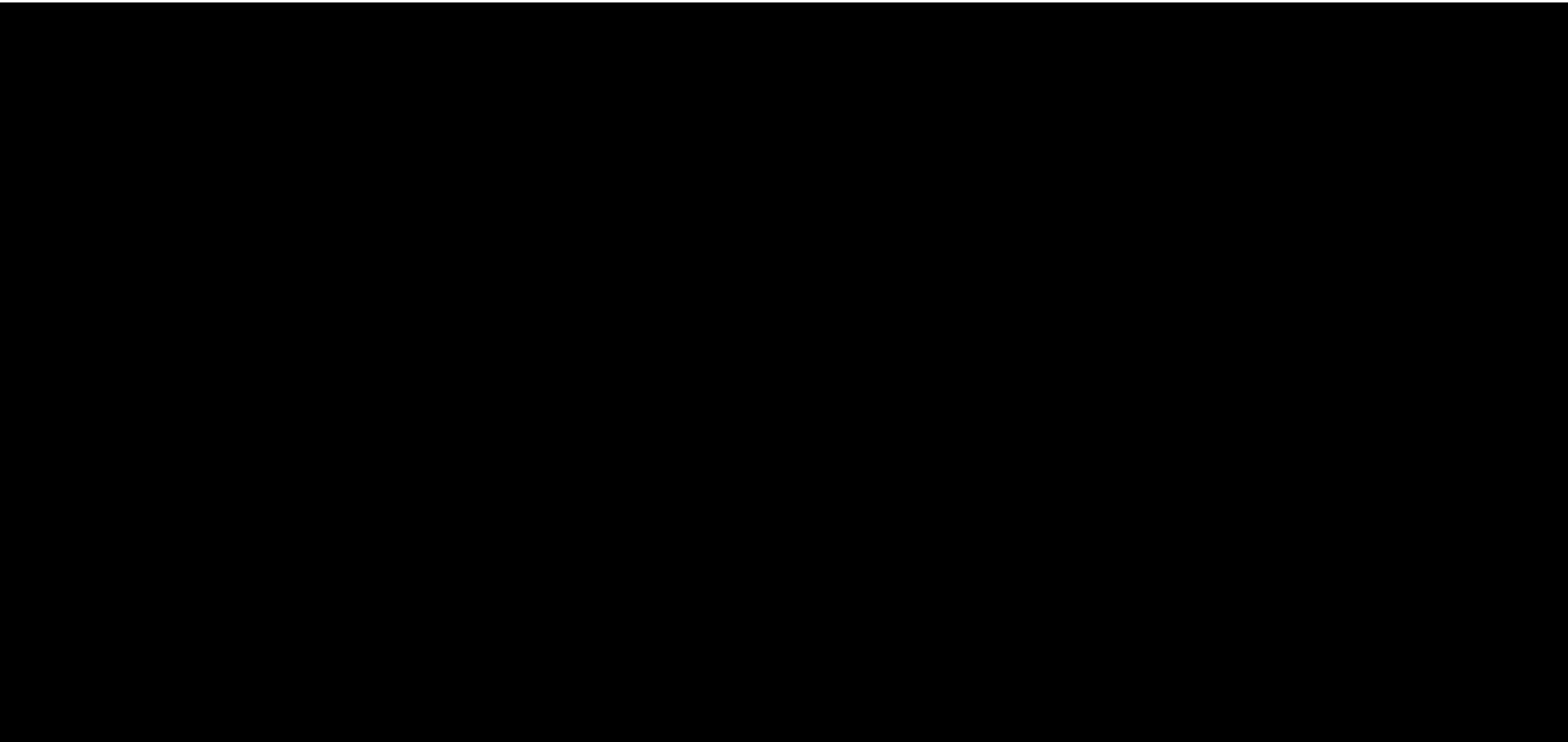
Machine Learning Process



How Neural Network works



Backpropagation



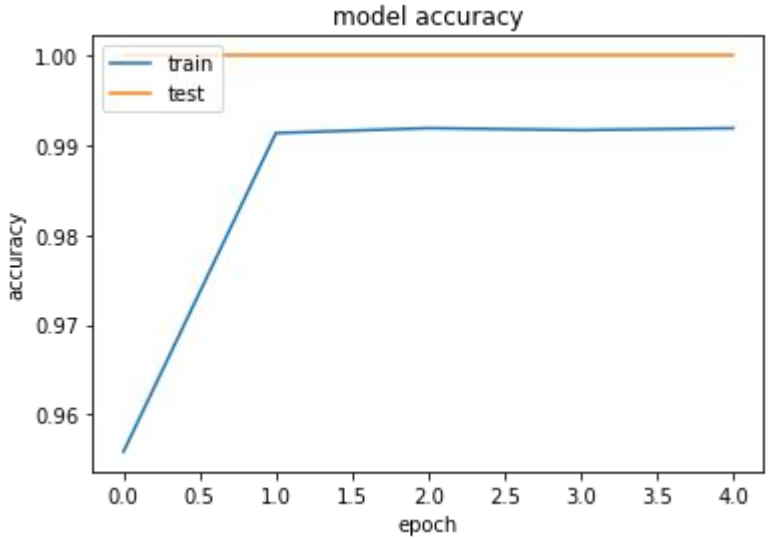
Tensorflow

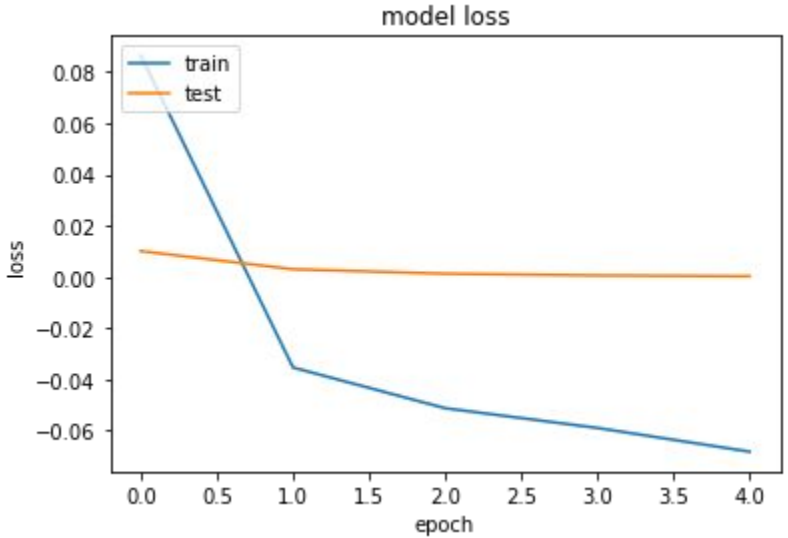
- Software library
- User friendly
- Machine learning

Keras

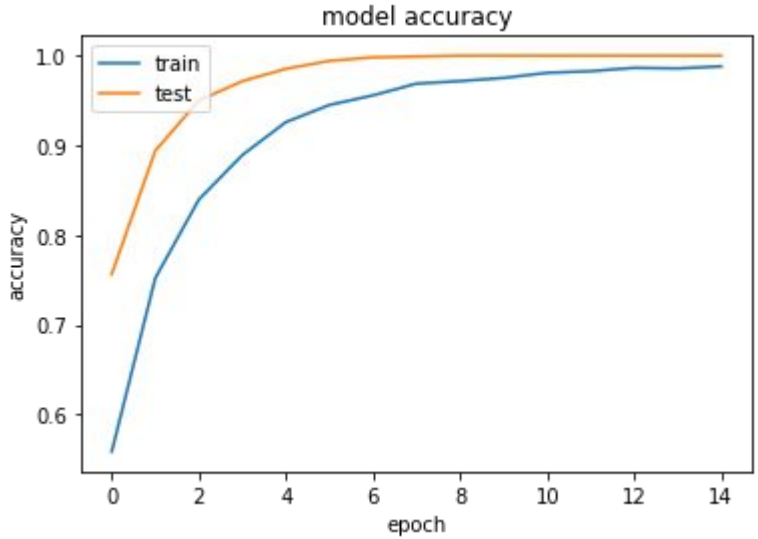
- Library upon Tensorflow
- Written in Python
- Higher levels

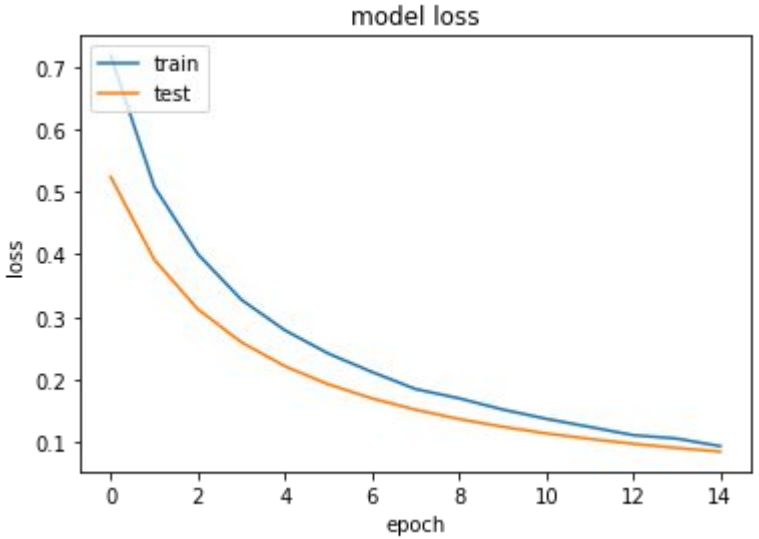
Results


model accuracy
train
test
accuracy
epoch
1.00
0.99
0.98
0.97
0.96
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0

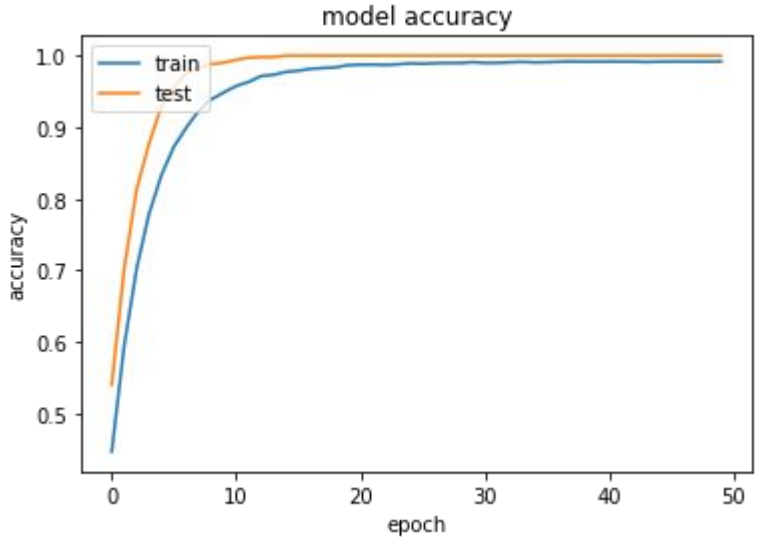

model loss
train
test
loss
epoch
0.08
0.06
0.04
0.02
0.00
−0.02
−0.04
−0.06
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0

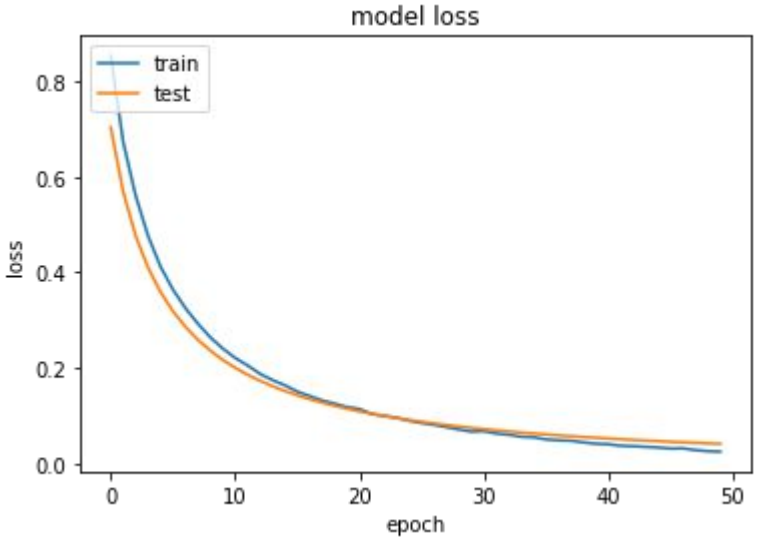
Attempts to improve the model





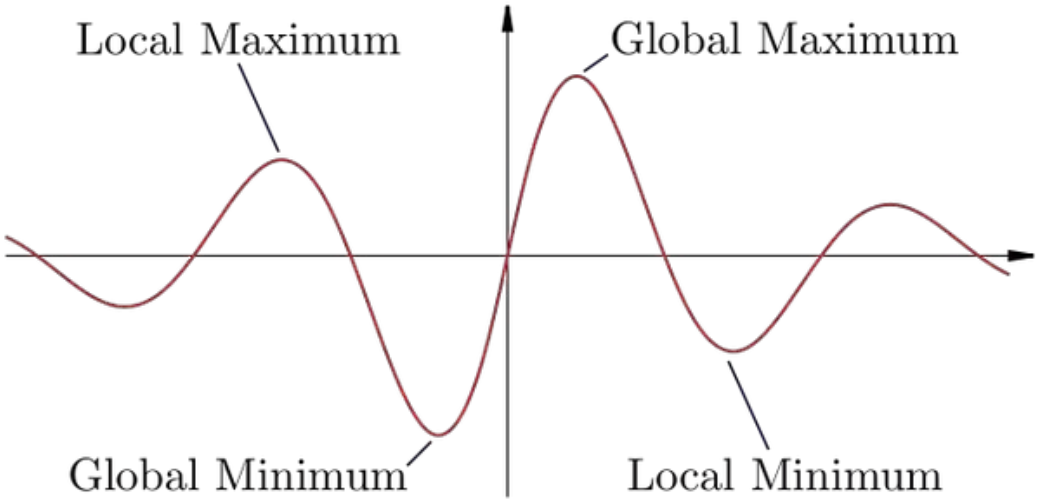
Attempts to improve the model (slight overfitting)





How to improve

```
accuracy: 0.99122804
```



What we've tried to improve

1. Add an initializer to initialize every weight to a random number before training instead of 0 as default value.

```
tf.keras.initializers.RandomUniform(minval=-500, maxval=500, seed=None)
```

2. Playing with the optimizer by lowering the learning rate.

```
optimizer=(tf.keras.optimizers.Adagrad(lr=0.02, epsilon=None, decay=0.0)
```

3. Make more epochs with larger batches.

```
model.fit(data_train, label_train, validation_split=0.1, batch_size=100, epochs=5)
```

What we've tried to improve

Make a deeper network with more layers. and changing the activation function.

```
tf.keras.layers.Dense(512, activation=tf.nn.relu),

tf.keras.layers.Dense(64, activation=tf.nn.sigmoid),

tf.keras.layers.Dense(2, activation=tf.nn.sigmoid)
```

Add dropout layers, it prevent for overfitting while allowing for better training.

```
tf.keras.layers.Dropout(0.5)
```

Get faster learning

Performance gain thanks to GPU acceleration (NVIDIA / CUDA)

30s => 3s

i5-8250U vs 950m (640 CUDA core)

Failures

1. Operating Intel movidius on Ubuntu virtual machine.
2. Hard time to make the neural network working because of a bad loss function.
3. Hard time understanding the assignment.
4. Dataset’s too easy to work with, lack of challenge.

Devices

NCS2 - Intel Neural Compute Stick 2

- 16 VPU (visual processing units)
- specifically for image and video processing / leverage CNN
- Possible to process IoT data where it is generated

Collaboration & Communication

1. Communication went well although each students has different culture.
2. Being late some time.
3. Difficulty in Communication due to imperfect English.
4. We were equal, everyone participated evenly.
5. Everyone was willing to help one another.

Conclusion

1. Completed our goal
2. Great time
3. Had some failures.

END

Questions?

References

1. House Pricing Demo:
   https://www.freecodecamp.org/news/how-to-build-your-first-neural-network-to-predict-house-prices-with-keras-f8db83049159/
2. Neural Network tutorial with Keras
   https://www.freecodecamp.org/news/how-to-build-your-first-neural-network-to-predict-house-prices-with-keras-f8db83049159/
3. NCS2
   https://software.intel.com/en-us/articles/get-started-with-neural-compute-stick