ANALYSIS OF WAVEFORMS IN THE SATELLITE ALTIMETRY BY USING NEURAL NETWORKS

By Dennis Mattes
dennis-mattes@gmx.net
What is retracking and why is it important?

Receiving height information by using radar signals

\[ \Delta R = \frac{1}{2} \cdot \Delta x \cdot c \cdot \tau \]

- \( \Delta R \) = Retracked range
- \( \Delta x \) = offset
- \( c \) = speed of light in vacuum
- \( \tau \) = Puls duration

Therefore we need special techniques to define the correct position of the water peak

\[ H = A - (R + \Delta R) \]
Motivation

The satellite track covers water and land areas.

The first question which occurs is: is the measurement over water or land?

Neural networks can:
- Learn characteristic pattern
- Detect the correct waveform
- Do the retracking

![Diagram of satellite track over water and land with measurement graphs showing differences between water and land measurements.]
How neural networks work

Introduction

• First works about neural networks are published in the 1950s
• With the resources of big companies (e.g. Google, Baidu, Huawei, ...) they are now on a level that they can be used in our daily life
• Even human like interactions are now possible (e.g. Sophia from Hanson Robotics)

As a basic for this work and also for the presentation the book Make your own neural network from Tariq Rashid was used

Sophia at the Finastra University (Singapore)

Source: https://twitter.com/realsophiarobot
How neural networks work
The basic idea

A biological neuron as an idea:

Source: https://askabiologist.asu.edu/sites/default/files/resources/plosable/Brain_Speed/connected-neurons.jpg

Basic idea of a simple artificial neuron:

$x = a + b + c$

Input a
Input b
Input c

Output y

Does every input should create an output?
How neural networks work
How to activate a neuron – Part 1

Actual situation:

Input a → $x = a + b + c$ → Output y

Input b

Input c

Every input creates an output

Improvement:

Input a → $x = a + b + c$ → $y(x)$ → Output y

Input b

Input c

Smooth increase between 0 and 1

Source: https://commons.wikimedia.org/w/index.php?curid=4310325
To avoid an abrupt jump we will use the sigmoid function:

\[ y = \frac{1}{1 + e^{-x}} \]

With this:
- Small signals are suppressed
- Strong signals are increased
- Range is between 0 and 1
Now we know how one artificial neuron works

The next step is to combine them

The human brain is also organized in different layers of neurons to propagate the signals to their destination

Three layer neural network:

- **Input layer**
- **Hidden layer**
- **Output layer**

**Connections**

**Neurons**

**Input**

**Output**
The question now is, how can a neural network learn?

The answer is in the **connections** between the layer!

Each connection has a special weight which will be multiplied with the transmitted value.

Example for node $N_{2,1}$: $x = i_{1,1} \cdot w_{1,1} + i_{2,1} \cdot w_{2,1}$

With these weights it is possible:
- To strength a connection which provides useful informations
- To suppress a connection which provides less useful informations
How neural networks work
Backpropagation

- The neural network adjusts the **weights** of the connections during the training phase.
- The weight adjustment depends on the **error** during the learning process:

**Step 1:**

- Estimated output $y_{est}$

**Step 2:**

- The learning error $e$ is then calculated by:
  
  $$ e = y_{true} - y_{est} $$

**Step 3:**

- Now the error is distributed to the connections depending on their actual weight:
  
  $$ e_{1,1} = \frac{w_{1,1}}{w_{1,1} + w_{2,1}} \cdot e $$
Now we learned how the neural network works and how they can learn. But the last question is, what is the output?

The sigmoid function has values between 0 and 1 in this range is the output.

The output layer has the size of the number of the searched labels. We will receive a value at the position of the label.

Source: https://commons.wikimedia.org/w/index.php?curid=4310325

Output of our neural network:
- Water = Label 1
- Land = Label 2

The tested waveform is water with 99.66%.
Overview of the developed algorithm

1. Analysing the study area
2. Data processing
   - **First network**: decision if it is water or land
     - **Input**: Normalized waveforms with labels
   - **Second network**: tracking the peak in the water waveform
     - **Input**: Water waveforms with labels
     - **Input**: Detected peak position
   - **Input**: Detected peak position
3. Calculate the water height
In this study we analyse the waveforms of the Cupari river, which is located in Brazil.

Analysing the study area

Data processing

First network: decision if it is water or land

Second network: tracking the peak in the water waveform

Calculate the water height
Processing the data

- Because the sigmoid function is in a range between 0 and 1, the input data also has to be in that range → Waveforms have to be normalized

- That the neural network can learn the characteristics, the datasets have to be labelled:

  **Input for the first neural network**

  - Water = Label 1
  - Land = Label 2

Source: https://commons.wikimedia.org/w/index.php?curid=4310325
Analysing the first results

The first neural network

- The network will label all waveforms in water and land
- The advantage is, that above the water area are very clear peaks

<table>
<thead>
<tr>
<th>Label</th>
<th>Detection rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Water</td>
<td>100 %</td>
</tr>
<tr>
<td>Land</td>
<td>20.07 %</td>
</tr>
</tbody>
</table>

- The land area shows very different pattern which are difficult to detect for the network
Analysing the first results

The second neural network

- The useful water waveforms are now detected
- To find the correct peak, we need more informations than only the label

How can we select the assumed water peak?

The output gives us the probability for each label.
Analysing the first results
The second neural network – Methodology 1

1. Create a window with the size of 23 bins which defines the input for the network
2. Save the label and probability from the output
3. Move the window 2 bins and repeat it

1. First network: decision if it is water or land
2. Second network: tracking the peak in the water waveform
3. Calculate the water height

⇒ This data is now the input for the next calculation

We count how often each peak is tracked!
Analysing the study area

Data processing

First network: decision if it is water or land

Second network: tracking the peak in the water waveform

Calculate the water height

The tracked peaks are now the input for the water height calculations!

\[ \bar{P}_1 = 0.9995 \quad \bar{P}_2 = 0.9923 \]

\[ n_1 = 6 \quad n_2 = 3 \]

\[ \Rightarrow \bar{P}_1 \cdot n_1 = 5.997 \]

\[ \Rightarrow \bar{P}_2 \cdot n_2 = 2.9769 \]
Analysing the first results
Calculation of the water heights

• Comparison of the calculated water level with the water level, measured by in situ stations

\[ \text{water level} = A - (R + \Delta R) + \text{corr} \]

- **Analysing the study area**
- **Data processing**
  - First network: decision if it is water or land
  - Second network: tracking the peak in the water waveform
- **Calculate the water height**
Analysing the first results
Calculation of the water heights

Now we compare the water level with water levels, generated by the MLE4 retracker:

- As it can be seen, there is a delay in the peak maximum compared to the in situ data.

Beside this problem, the main variations of the in situ water level can be reconstructed.
Analysing the first results
Calculation of the water heights

• At least, the residuals can be calculated to determine the standard deviation and the mean value from it:

\[ res = waterLevel_{\text{InSitu}} - waterLevel_{\text{Retracked}} \]

<table>
<thead>
<tr>
<th></th>
<th>Water level with 2. NN [m]</th>
<th>Water level with MLE4 [m]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard deviation of the residuals</td>
<td>1.2872</td>
<td>1.1886</td>
</tr>
<tr>
<td>Mean value of the residuals</td>
<td>0.9549</td>
<td>1.0420</td>
</tr>
</tbody>
</table>

• Reagrding this statistics the results are comparable with each other
Analysing the first results

Summary

- There are good results by using a neural network for the classification of waveforms (first neural network).
- We have still several problems by using a neural network for the retracking but already good results (second neural network).

⇒ Neural networks show a big potential for further studies in this area.
Future work

Until now, it is not possible to handle noisy datasets, where we have multi-peaks close to each other:

Which peak is from the water signal?

Solution is regarding the time aspect:

Recurrent neural networks can predict the position of a signal by regarding the previous signals.
Thank you very much for your attention!