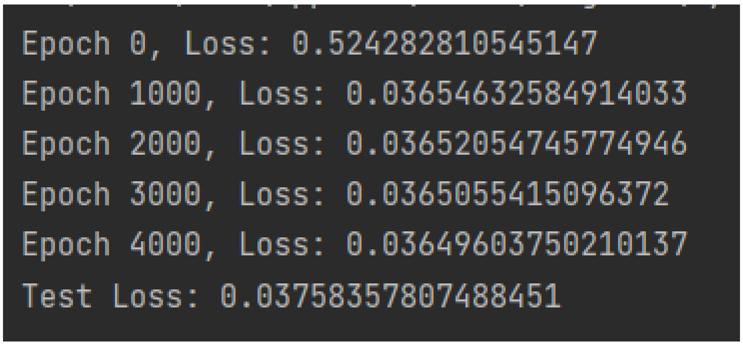
# Assignment 7 – Jakub Rysiak

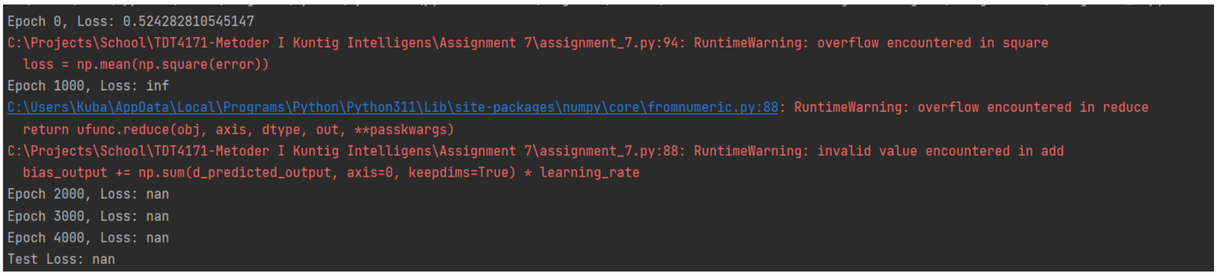
## Results

The training gave the following results:



```
Epoch 0, Loss: 0.524282810545147
Epoch 1000, Loss: 0.036546325849140 33
Epoch 2000, Loss: 0.03652054745774946
Epoch 3000, Loss: 0.0365055415096372
Epoch 4000, Loss: 0.036496037502101 37
Test Loss: 0.037583578074884 51
```

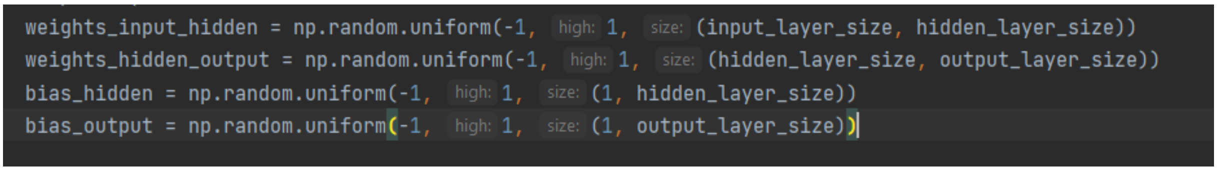
At first I used 5000 epochs and learning rate of 0.01, and it turned out that the learning rate was a bit high, since the parameters changed to much and produced an error too big to represent in python.



```
Epoch 0, Loss: 0.524282810545147
C:\Projects\School\TDT4171-Metoder I Kuntig Intelligens\Assignment 7\assignment_7.py:94: RuntimeWarning: overflow encountered in square
  loss = np.mean(np.square(error))
Epoch 1000, Loss: inf
C:\Users\Kuba\AppData\Local\Programs\Python\Python311\Lib\site-packages\numpy\core\fromnumeric.py:88: RuntimeWarning: overflow encountered in reduce
  return ufunc.reduce(obj, axis, dtype, out, **passkwargs)
C:\Projects\School\TDT4171-Metoder I Kuntig Intelligens\Assignment 7\assignment_7.py:88: RuntimeWarning: invalid value encountered in add
  bias_output += np.sum(d_predicted_output, axis=0, keepdims=True) * learning_rate
Epoch 2000, Loss: nan
Epoch 3000, Loss: nan
Epoch 4000, Loss: nan
Test Loss: nan
```

After I lowered it to 0.001 it worked fine, with the results I pasted at the top.

The parameters were initialized as such:



```
weights_input_hidden = np.random.uniform(-1, high: 1, size: (input_layer_size, hidden_layer_size))
weights_hidden_output = np.random.uniform(-1, high: 1, size: (hidden_layer_size, output_layer_size))
bias_hidden = np.random.uniform(-1, high: 1, size: (1, hidden_layer_size))
bias_output = np.random.uniform(-1, high: 1, size: (1, output_layer_size))
```

Random values between -1 and 1 to not produce too drastic numbers from the start.