|  |  |
| --- | --- |
| **Assignment Case** |  |
| COMP7117  Artificial Neural Network |
| **Computer Science** | **E193-COMP7117-CX01-02** |
| ***Valid on*** *Even Semester Year 2018/2019* | **Revision 00** |

## Soal

*Case*

**Hawaii Air Quality**

**Carbon Dioxide levels** have been **increasing** everywhere all around the world, including in Hawaii. The **Department of Environment** in **Hawaii** are concerned whether **carbon dioxide levels** will continue to **increase** and thus **effecting** the **air quality** in **Hawaii**. Therefore, you are hired to **create a system** to **predict the carbon dioxide levels** based on the previous carbon dioxide levels in Hawaii. The system will be built using **recurrent neural network** technique, implemented in **Python** with the help of **Numpy** and **Tensorflow** library.

1. **Dataset Description**

The dataset consists of **384 time-series data** taken from the **history of carbon dioxide levels** in **Hawaii** from **January 1987** until **December 2018**. The data is used to **determine the carbon dioxide levels** on the following month.

1. **Architecture**

The architecture for the neural network is as follows:

|  |  |
| --- | --- |
| **Architecture** | **Notes** |
| Input Layer | 1 neuron |
| Context unit (hidden) | 5 neurons |
| Output Layer | 1 neuron |
| Unroll count (timestep) | 3 |

The activation function used for all layers is **sigmoid function**.

1. **Training**

The training is done with **gradient descent** **optimization** formula for **1,000 epochs** and **use 70% of the dataset**. The **training method** is **mini-batch gradient descent** with **4 data per batch**. 3 minibatch, 4 data : 1 input 12x training

The training procedures are as follows:

1. **Initialization**

The initialization step needs to be **run once** before starting the training iteration:

1. The **data in the dataset need to be normalized** before being passed to the neural network. Below is the **formula** of **normalization**:

kata aslabnya gausah pake yg bawah, pake normalisasi biasa aja (MinMaxScaler)

|  |
| --- |
|  |

1. **Initialize** the **weights and biases randomly**
2. **Iteration**

For **1,000 epochs**, repeatthe following:

1. **Calculate the error** by comparing the **output** of the **neural** **network** to the target in the dataset **using mean squared error** (**MSE**)
2. **Update** the **weights and biases** using **gradient descent optimization**
3. **For every 250 epochs**, **print** the **current error** and **iteration number** to the console
4. **For every 200 epochs**, **save** the **variables** to **checkpoints**

The formula required for the training process can be found in the Appendix section.

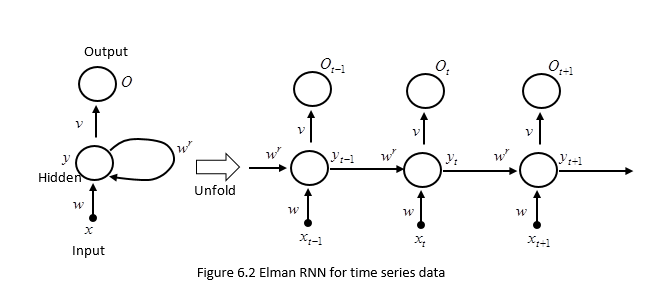
1. **Evaluation**

The neural network will be **evaluated** using **30% of the dataset** using the **model** that **previously** has been **saved**. The result will be the **predicted** **occupancy** **rates** **of** **customers** on the **following** **month**.

1. **Appendix**

**Recurrent Neural Network**

These are the formulas for a neural network trained with recurrent neural network.



**Forward Pass**

The forward pass formulas are:

* First layer (Input layer to hidden layer)
  + Before feedback loop
  + After feedback loop
* Second layer (hidden layer to output layer)

**Backward Pass**

The backward pass formulas are:

* Error calculation (Mean square error)
* Weight update
* Bias update

\* In TensorFlow, use an optimizer instead of updating the weights and biases manually.

**Reference**

* The dataset is retrieved from Census at School website with a few changes (https://new.censusatschool.org.nz/wp-content/uploads/2013/02/Rachel-Passmore-TS-data-sets-2.zip)