Assignment 3

av

Lars-Erik Bakkland Moi and Oskar Markussen

i

IKT450
Deep Neural Networks

Universitetet i Agder

Grimstad, August 2023

Contents

1 Task

In this task you are suppose to develop a deep feed forward neural network using a high level library, e.g. PyTorch. You should test different network structures.

- Download the ECG database https://www.cs.cmu.edu/ bobski/data/ data.html
- Predict the 2 classes: Normal and abnormal.
- You need fix the input size to the data. One option is to include only the first 75 measurements and either remove any recordings below 75 or pad with 0s or averages.
- Choose the network architecture with care.
- Train and validate all algorithms.

2 Deep Feed Forward

We utilized the tensorflow and keras libraries to develop our multilayer perceptron for ECG data classification. For building our neural network layers, we employed the Keras Sequential model, which offers a convenient linear stack of layers.

In our preprocessing step, we loaded ECG data from two directories: one for abnormal ECGs and another for normal ECGs. After loading, we combined the data from both classes and subsequently divided it into training and validation datasets using an 80/20 split, ensuring reproducibility through a random seed during the split.

Our neural network architecture is as follows:

- An input layer designed to accommodate a feature vector of size 150.
- The first hidden layer consists of 64 neurons and uses the ReLU activation function to introduce non-linearity.
- The second hidden layer comprises 32 neurons, also applying the ReLU activation function.
- The network culminates in an output layer with a single neuron, employing the sigmoid activation function to map the outputs between 0 and 1 where 1 is mapped to the abnormal ECG.

For the model's compilation, we selected the Adam optimizer with a learning rate of 0.001 and experimenting with the learning rate of 0.0001 giving us the provided result in 1 and 2 . We used binary cross-entropy as our loss function. We trained the model for 100 epochs, as we can see from the 1 it does converge on around the 30 epochs, but from the **??** we can see that the training and validation accuracy is increasingly higher after 30 epochs and stops progressing around 80 epochs.

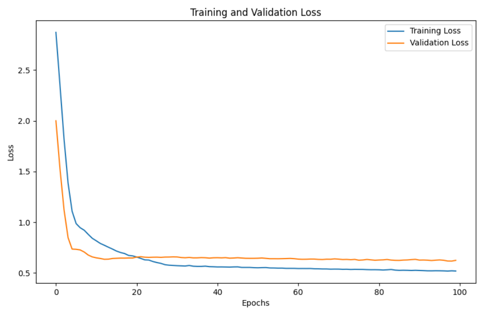

Figure 1: Training Loss & Validation Loss

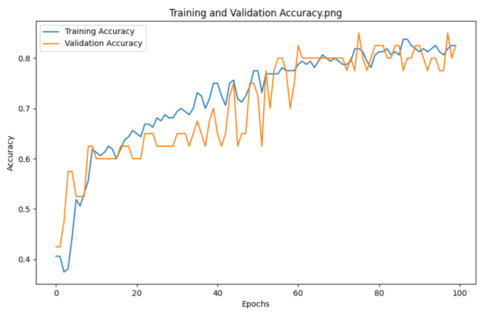

Figure 2: Training Accuracy & Validation Accuracy

3 Conclusion

In conclusion we where able to create the deep forward network using the different libraries such as TensorFlow, our results provided us with an end result of around 80% accuracy giving us a reasonably good model for further use, but would need a bigger data set to further train our model.