Connor McCurley

EEE 6814 **Homework 2** Fall 2019

Experiment 1 - Single Hidden Layer MLP

Baseline Architecture

To provide a baseline model, I designed a single hidden layer MLP and implemented it in pytorch (code included). The model architecture consisted of an input layer which took the EEG features, a hidden layer with a variable number of units (described more in the following), each of which employed ReLU activation functions and six output neurons which fed into a soft-max activation (Figure 1).

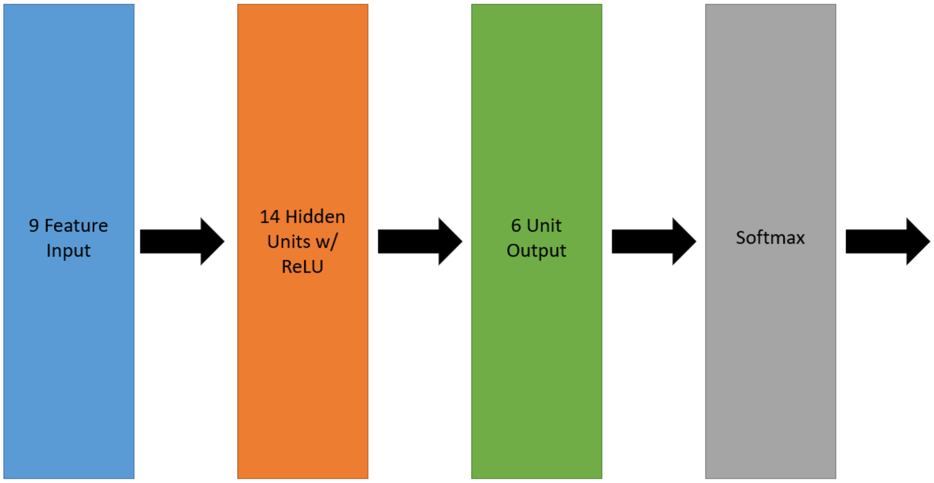


Figure 1: Architecture of the baseline MLP. The network defines a single hidden layer MLP with 14 units in the hidden layer and 6 neurons in the output layer. ReLU activations were applied at the hidden units, while a softmax was used at the output to provide class score probabilities.

Training

Since the data labels were categorical, I employed a cross-entropy loss function. I split the data from patient 1 into training and validation folds to train the MLP. A variety of parameters were explored in training (described in section 2). Below I show the results of training a single hidden layer MLP with 14 units in hidden layer. A learning rate of $\eta = 0.1$ was applied with stochastic gradient descent to update the network parameters. Ten percent of the training data was randomly chosen for validation. The model was randomly initialized 10 times. Training was ended when validation loss began to increase. The model with the best validation performance after training was chosen as the final model. After training, the data from patient 2 was used in test. Learning curves for the “best” model and a confusion matrix for the test performance are shown in Figures 2 and 3, respectively.

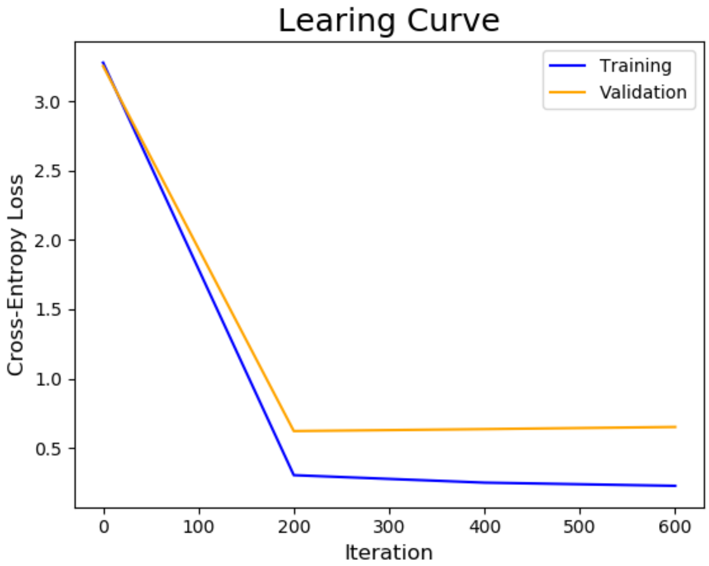


Figure 2: Learning curves for the "best" instance of the model described in Figure 1. At 200 epochs the cross-entropy loss on the training dataset is approximately 0.2, while validation loss is around 0.65. After 200 batch epochs, the model begins to overfit the training data.

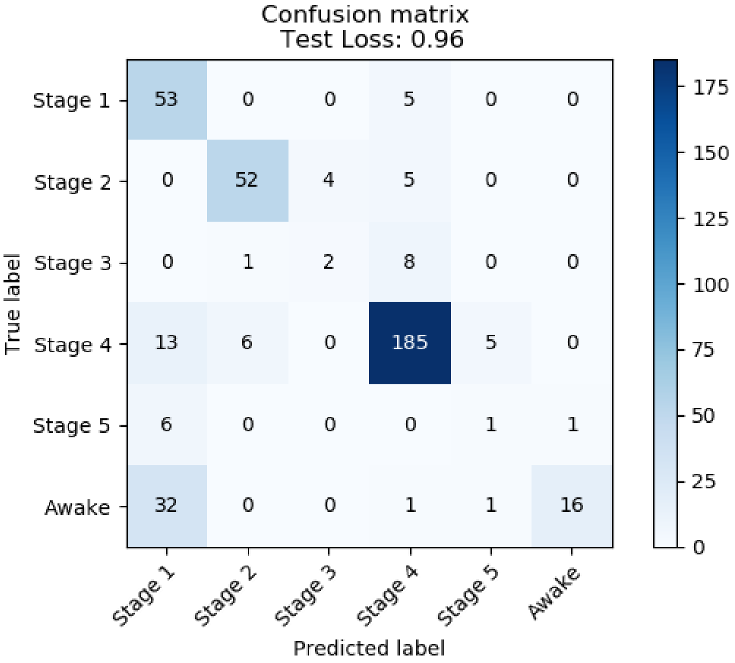


Figure 3: Confusion matrix for testing on patient 2. It can be observed from the figure that sleep stages 2 and 4, as well as the awake state, were generally classified with low error. Stage 1 was often predicted as stage 4 or awake and stage 5 was sometimes mis-judged as state 4. The cross entropy loss on this test set was calculated as 0.96.

Baseline Results

It can be observed from the figures that the network begins to overfit at approximately 200 batch training epochs. This is determined by the fact that training error continues to decrease with additional updating, while performance loss on the sequestered validation data increases. At the end of training for this particular model, the cross-entropy loss for the training data was approximately 0.2, while validation error was around 0.65. Testing this model on patient 2's data provided the results shown in Figure 3. It can be observed from the figure that sleep stages 2 and 4, as well as the awake state, were generally classified with low error. Stage 1 was often predicted as stage 4 or awake and stage 5 was sometimes mis-judged as state 4. The cross entropy loss on this test set was calculated as 0.96.

Experiment 2 - Parameter Variation

Next, experiments were conducted to test the effects of hyper-parameter variations. Only one parameter was changed in each test. Training was conducted in the same fashion as the baseline model for each parameter set. The models were randomly initialized 10 times and trained until the validation loss began to increase. The training and validation sets were constructed from patient 1's data and the test set was formed from patient 2's data, exclusively.

Note: I did not test stopping criteria in this experimentation, as it seemed unfruitful to purposely overtrain the model. (Although I acknowledge the possibility that my validation fold was not truly representative of the test data.)

Learning Rate

The effects of varying learning rate are summarized in Tables 1 and 2. The model exhibiting the best validation performance was used to generate the test loss for single hidden layer MLPs with 14 and 40 units in the hidden layers, respectively. It can be observed that moderate learning rates consistently provided the "best" performance.

Table 1: Test loss for a single hidden layer MLP with 14 units in the hidden layer. Cross-entropy loss on a sequestered test set is shown as well as the number of epochs for varying learning rates.

-	$\eta = 0.0001$	$\boldsymbol{\eta = 0.001}$	$\eta = 0.01$	$\eta = 0.1$	$\eta = 1$	$\eta = 10$
Epochs	600	**200**	200	200	200	200
Test Loss	2.99	**1.14**	1.22	1.22	2.71	3.46

Table 2: Test loss for a single hidden layer MLP with 40 units in the hidden layer. Cross-entropy loss on a sequestered test set is shown as well as the number of epochs for varying learning rates.

-	$\eta = 0.0001$	$\eta = 0.001$	$\boldsymbol{\eta = 0.01}$	$\eta = 0.1$	$\eta = 1$	$\eta = 10$
Epochs	600	600	**200**	200	200	200
Test Loss	1.78	1.49	**1.04**	1.45	2.26	3.86

Number of Layers

Next, the effects of layer size were tested. The number of hidden layers was varied from 1 to 6. The number of units in each hidden layer was fixed as 10, and the learning rate was set to $\eta = 0.01$. The test loss and number of training epochs for each model are shown in Table 3. It can be observed that the required number of epochs for parameter convergence generally increased as the number of hidden layers increased. Additionally, the networks greatly over-fit as they got deeper.

Table 3: Test loss and number of training epochs as a function of the number of hidden layers. Each hidden layer contained 10 neurons. The learning rate was fixed to $\eta = 0.01$.

# Hidden Layers	1	**2**	3	4	5	6
Epochs	200	**1200**	1000	1500	1700	2500
Test Loss	1.09	**1.03**	1.27	1.14	1.46	2.76

Number of Units in Each Layer

Finally, given that that a 2 hidden layer MLP obtained the best performance in the previous experimentation, this architecture was chosen to test the effects of the number of units. The learning rate was fixed again at $\eta = 0.01$. The number of units in the first and second hidden layer were varied. Table 4 shows the number of training epochs required for convergence as well as the cross-entropy loss on the test set. From the experimentation, it was determined that a smaller number of units in the first hidden layer and a larger number (in comparison) in the second hidden layer provided the best performance on an un-seen test set. Although this result may not be general.

Table 4: Test loss and number of training epochs as a function of the number of units in a 2 hidden layer MLP. The learning rate was fixed to $\eta = 0.01$.

# Units 1st HL	10	10	**5**	40
# Units 2nd HL	10	5	**10**	40
Epochs	2000	4000	**1000**	800
Test Loss	1.16	1.13	**0.94**	1.38

Experiment 3 - Generalization

Finally, given that the data from each patient could be drastically different, it makes sense that data from both patients should be incorporated in training. In attempt to improve generalization, K-fold cross validation incorporating data from **both patients** was implemented. A 2 hidden layer MLP was implemented with 5 units in the first hidden layer and 10 in the second hidden layer. The learning rate was, again, fixed at $\eta = 0.01$. To perform this cross-validation, the data was first split into train/ test sets. 5-fold cross-validation was then applied to the training data, where a model was trained 10 times on 4 folds and validated on the 5th. The most accurate model was selected from the cross-validation and applied to the combined test set. Since training was relatively fast, the process was repeated 5 times. Learning curves and test confusion matrix for one k-fold run is shown in Figures 4 and 5. The mean test accuracy for the 5 trials was found as: 0.61 ± 0.08. This confirms the hypothesis that augmenting the training set with data from both patients did, indeed, improve generalization.

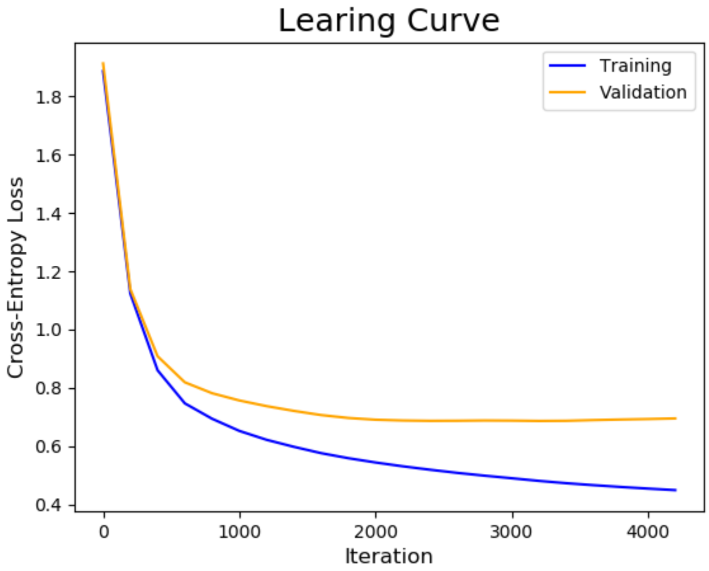
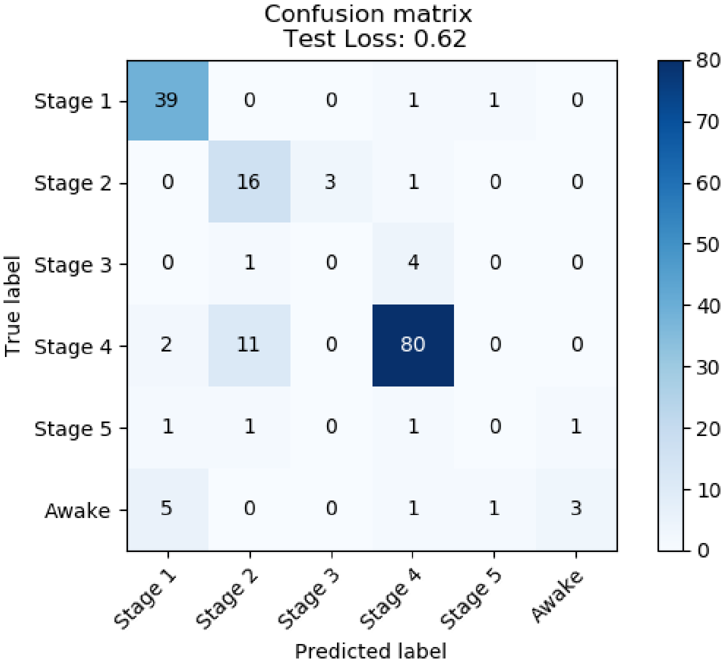


Figure 4: Learning curves for the "best model" chosen during 5-fold cross-validation. Validation loss begins to increase after approximately 3500 training epochs.



	Stage 1	Stage 2	Stage 3	Stage 4	Stage 5	Awake
Stage 1	39	0	0	1	1	0
Stage 2	0	16	3	1	0	0
Stage 3	0	1	0	4	0	0
Stage 4	2	11	0	80	0	0
Stage 5	1	1	0	1	0	1
Awake	5	0	0	1	1	3

Figure 5: Confusion matrix for the "best model" chosen during 5-fold cross-validation. The total cross-entropy loss evaluated to 0.62 for the combined test set.

References

[1] Principe, Jose C., Euliano, Niel R., Lefebvre, W. Curt. ”Chapter III- Multilayer Perceptrons,” in Neural and Adaptive Systems: Fundamentals Through Simulation, 1997