Deep Learning Project Report: Classification on EEG Data

Yahya Essa
yezzeldin@g.ucla.edu
UID: 704-515-916

Gaurav Kumar Agarwal
gauravagrarwal@ucla.edu
UID: 904-558-368

Abstract

We use the "default" dataset [1] given for this course to do the classification task based on EEG data. We use three different architectures: the first architecture is a convolution neural network based architecture while the second and the third architectures are based on recurrent neural networks. These three models are then tested on 50 *samples from each subject (which was kept separate from the training and validation data). We evaluate these models for two cases: (1) where we use only* 22 *electrode and (2) we use all* 25 *electrodes (i.e., use the signals from EOG as well). We make interesting observations on the performance when training across different subjects vs. a single subject. We also explore how the performance of a model trained for a subject performs when tested against a different subject.*

1. Introduction

Machine learning and Deep Learning have recently been quite promising in learning relationship between data and labels. Although major results are still in the field of computer vision (MNIST, CIFAR, IMAGENET etc.), other fields such as Human Computer Interactions have potential to leverage these advances in Deep Learning. For this project, we apply tools from deep learning, particularly convolution and recurrent neural networks and see if we can classify actions based on EEG data.

2. Results

2.1. Data Preprocessing

EEG signals for each subject were recored for 288 trials, and each trial has EEG signals of dimension (22, 1000) along 22 EEG channels. We also have data along three EOG channels of dimension (3,1000). We first remove trials where there are some occurrences of NaN. We ends up removing 34 trials due to this.

Next, from each subject we take 50 random samples and put them as test data. From the remaining data, for each subject we take $\sim 20\%$ samples as validation data and rest of the samples as training data.

With this we will have 1687 training samples, 421 validation samples and 450 test samples. If we use only 22 electrodes our training data is of shape $(1687, 22, 1000)$ and if we rather use all 25 electrodes, then our data has size $(1687, 25, 1000)$.

2.2. CNN Based Architecture

Detailed architecture is shown in Fig. 4a. Our loss function is softmax loss function added with regularization of all the kernal weight with a scale of 0.1. We train this network with Adam optimizer in batch sizes of 50 for 10000 steps using Adam Optimizer. Our accuracies are shown in Fig. 5.

2.3. RNN based architecture

Detailed architecture is shown in Fig. 4b. We use two LSTM units. Our loss function is softmax loss function. We train this network with Adam optimizer in batch sizes of 50 for 1000 steps using Adam Optimizer. Again the accuracies for each subject are shown in Fig 6.

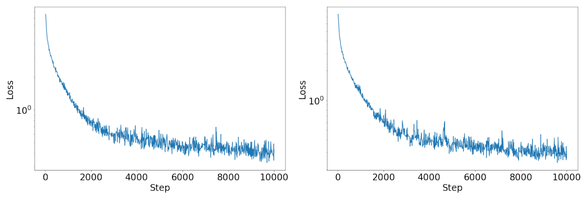
2.4. Architecture based on Bidirectional RNN

Detailed architecture is shown in Fig. 4c. Here we use a bidirectional RNN unit, having LSTM units in both forward and backward layer. Our loss function is softmax loss function. We train this network with Adam optimizer in batch sizes of 50 for 1000 steps using Adam Optimizer. Accuracies are shown in Fig. 7.

3. Discussions

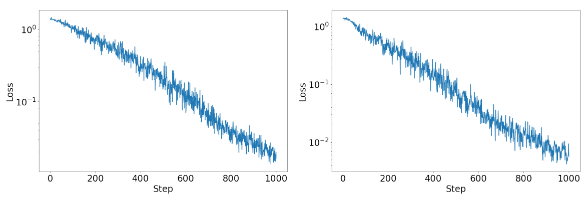
3.1. Selection of Hyper-parameters

- **Number of iterations:** We trained our CNN based architecture for larger iterations (10000) compare to number of iterations we used to two RNN based architectures (1000). This was mostly due to the fact that RNNs took a much longer time to train compared (50x per batch) compared to CNNs. We do, however, expect RNNS to improve as it drops exponentially as we see from the Loss functions for both 22 electrodes case and for 25 electrodes case in Fig. 1, 2 and 3 respectively.



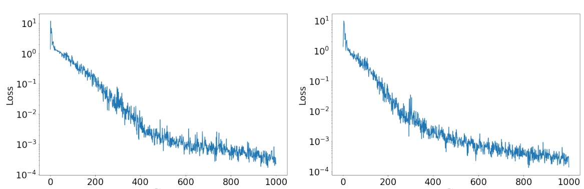
(a) 22 Electrodes (b) 25 Electrodes

Figure 1: Loss functions when we use CNN based architecture of Fig. 5.



(a) 22 Electrodes (b) 25 Electrodes

Figure 2: Loss functions when we use CNN based architecture of Fig. 6.



(a) 22 Electrodes (b) 25 Electrodes

Figure 3: Loss functions when we use CNN based architecture of Fig. 4c.

- **Number of units:** Increasing the number of units per cell in RNN and BiRNN have shown to increase the performance of the networks on the validation set (for all 9 subjects). Beyond 128 units, the improvement is very marginal but the training time is extremely high. Due to memory limitations on the GPU and to be perform sufficient number of training cycles, we opted to use 128 units per cell for both RNN and BiRNN.

- **Filter Size in CNN:** EEG measurement measures the level of neural activity in the brain as measured by the electrodes. As shown in Fig.3 in [1], the electrodes are ordered in a row-like fashion with every row being of different length. Therefore, although there is spatial correlation between electrode measurements, if one is not careful with applying a correlation filter, we can view, for example, electrodes 13 and 14 as if they are spatially correlated. We have experimented with the performance on the validation data set when we set the correlation window between the electrodes to be (5,7, 11 and ALL) in index and to size (21,41 and 61) temporally. We have found that setting the filter to size ALL spatially gives the best performance (with other sizes dropping the performance to the range of 30%), while increasing the size correlation window in time causes a marginal decrease in performance on the validation set. Therefore, for the convolution filter in Fig. 4a, we take the filter size spanning across all channels whether these are 22 or 25.

  Intuitively, the current design allows us to view the electrodes through a set of masks (depending on the number of filters we use). Thus, the CNN tries to learn which masks are more suitable for the classification task. Furthermore, since our output is of size $(1 \times 980 \times 128)$, this reduces the complexity (number of trainable parameters) of the consequent fully-connected layers.

3.2. What if we do not use data from other subject?

Recall that aggregated trials from all subjects in our training models. Here we restrict our training data only to a subject and see how well we perform when tested on the same subject. In the interest of not repeating same thing again, we do this only for one subject that got the maximum average accuracy (across models) while training with samples from all subjects.

We selected subject 5. Following table summarizes the test and validation accuracies when we use all three networks for both 22 and 25 electrodes.

NN	Channels	Validation Accuracy	Test Accuracy
CNN	22	0.326	0.32
CNN	25	0.80	0.72
RNN	22	0.56	0.5
RNN	25	0.847	0.84
BiRNN	22	0.739	0.76
BiRNN	25	0.87	0.88

Table 1: Accuracies for Subject 5after training only with Subject 5training data.

By comparing these to the accuracies in the Methods document, we see that more or less the same accuracy in RNN networks but a worse performance (compared to training on all subjects) in CNN. We attribute this to the possibility of the network over-fitting the training data in the CNN case when trained over only one subject. In other words, the CNN network complexity is actually higher than needed for training one a single subject and although the training accuracy reaches 100%, the validation and test accuracies suffer drastically. Training the data on different subjects actually performs a regularization effect on the training process allowing the classifier to generalize better as it needs to account of the different characteristics that each subject brings to the training data set.

3.3. Future Direction: How does a classifier generalize to un-seen subjects ?

In all the classification tasks described earlier, the classifier has been exposed to training EEG signals from the subject on which it is to be tested. Although the trials to be tested on are hidden, some individual characteristics could indeed be sampled by the training set. To explore how our current classifiers perform, we test the performance of the trained CNN, RNN and BiRNN with 22 channels for subject 5on the remaining subjects. In other words, we would like to see how does training perform cross-individually. For brevity, we only report the results for the test accuracies in the following table below.

Subj	CNN Accuracy	RNN Accuracy	BiRNN Accuracy
1	0.24	0.22	0.18
2	0.28	0.34	0.38
3	0.34	0.32	0.42
4	0.4	0.34	0.40
6	0.28	0.38	0.52
7	0.32	0.38	0.56
8	0.18	0.22	0.50
9	0.32	0.4	0.36

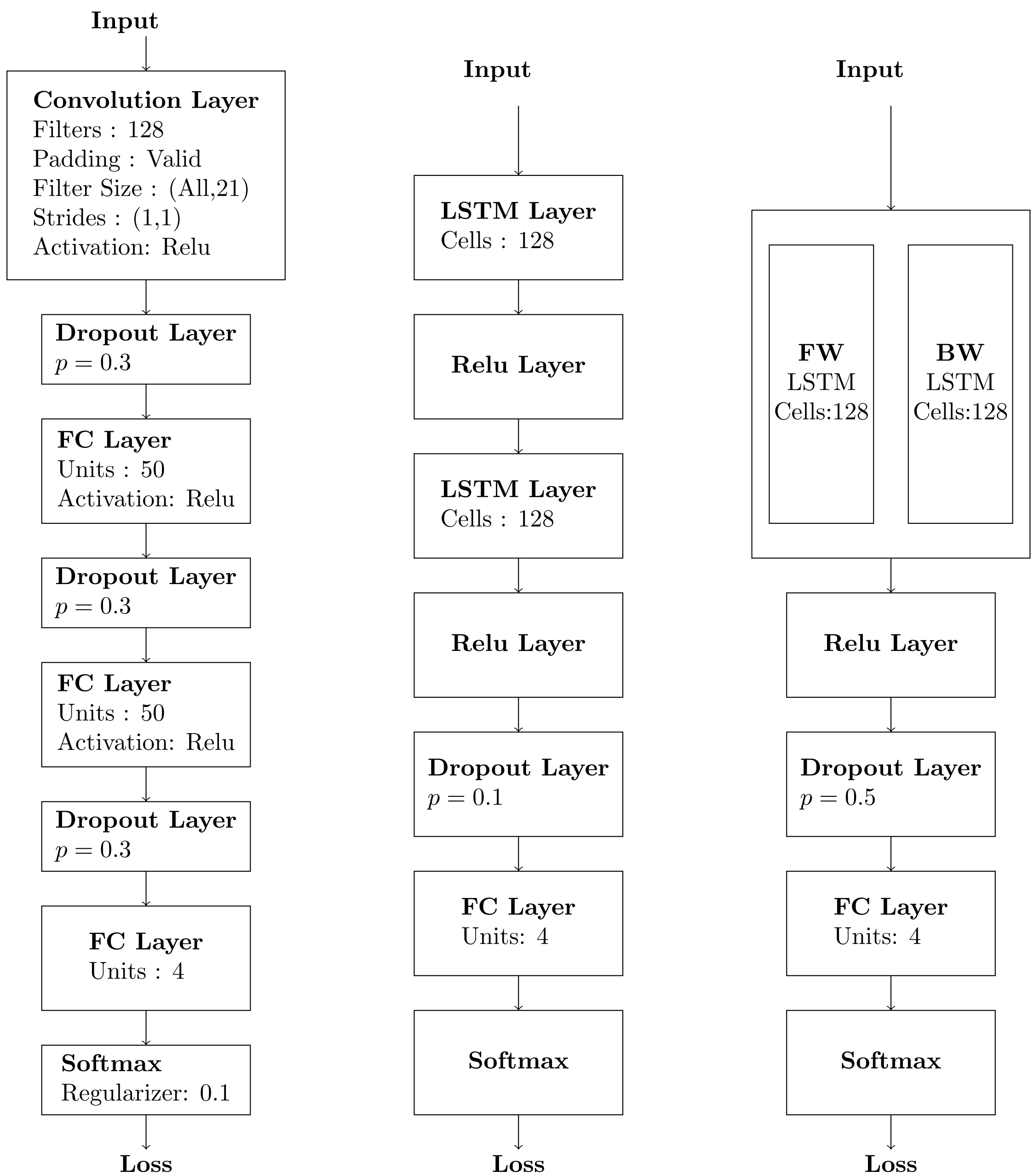
Table 2: Test Accuracies for other subjects on models trained for subject 5.

As seen from the table, the performance suffers when a model trained for a user is transfered to another. In some cases, the performance even dropped below chance level. It would of interest to understand whether training of warm-initialized models from other users can be fitted to another user with a few training iterations. The intuition is that such a model already understands the structure of an EEG signal but misses the tunings needed due to the individual characteristics of each subject. We pose this question as an interesting point of further investigation in the future.

References

[1] C. Brunner, R. Leeb, G. Muller-Putz, A. Schlogl, and G. Pfurtscheller. Bci competition 2008 - graz data set a, 2008. bbci.de.

Classifier Models



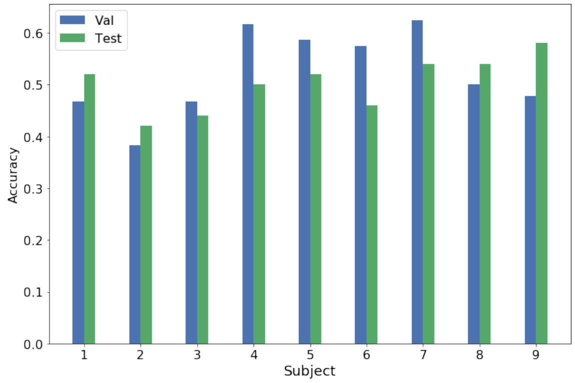
(a) CNN Based Architecture

(b) RNN based architecture

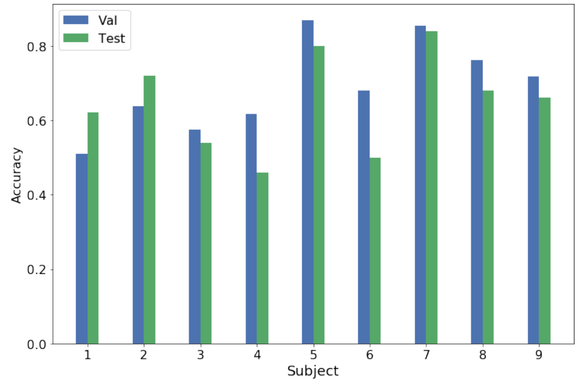
(c) Bidirectional RNN based architecture

Figure 4: Various Architectures: Adam Optimizer was used to optimize in all three architectures.

Performance of different Classifiers

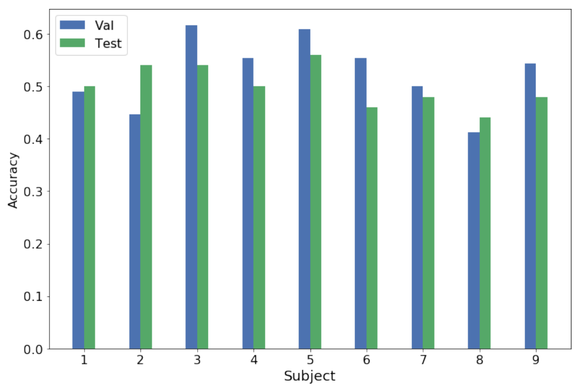


(a) 22 electrodes [Avg:52.26%(Val) 50.22%(Test)]

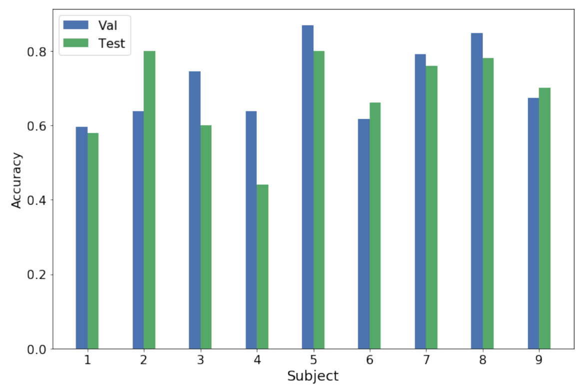


(b) 25 electrodes [Avg:69.12%(Val) 64.67%(Test)]

Figure 5: Accuracy on validation and test dataset for all 9 subjects for CNN based classifier.

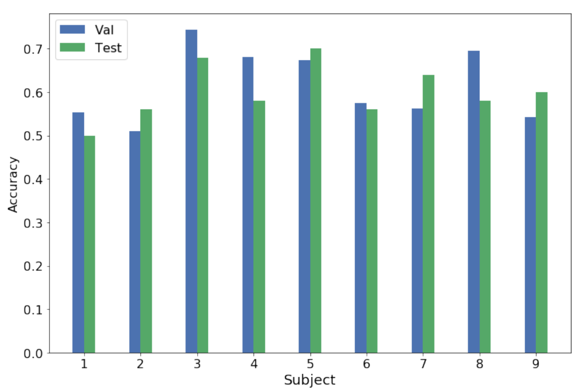


(a) 22 electrodes [Avg:52.49%(Val) 50%(Test)]

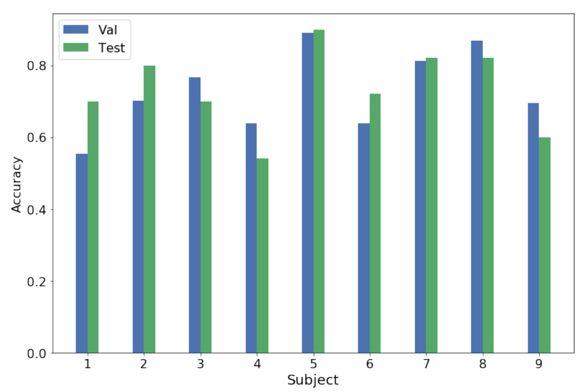


(b) 25 electrodes [Avg:71.26%(Val) 68%(Test)]

Figure 6: Accuracy on validation and test dataset for all 9 subjects for RNN based classifier.



(a) 22 electrodes [Avg:61.52%(Val) 60%(Test)]



(b) 25 electrodes [Avg:72.92%(Val) 73.33%(Test)]

Figure 7: Accuracy on validation and test dataset for all 9 subjects for Bidirectional RNN based classifier.