POLITECNICO DI MILANO

Artificial Neural Networks & Deep Learning

Homework2

Team:

Persepolis

Students:

Ali Noshad - 101108

Kiarash Rezaei - 991495

Ashkan Ramezani - 101150

ACADEMIC YEAR 2022-2023

1. Data Analysis and Visualization: At first, we standardized the dataset to put all the values on the same scale. Then we tried to perform different analyses on our dataset. The main ones were the following: class distribution and feature correlation. Regarding the first analysis, we realized that the dataset is unbalanced. For example observations belonging to the 10th class was more than 700, however class 0 had only 32 samples. In terms of features, we realized that the last two features are more correlated (with correlation coefficient of 0.7) than the others. To cope with the unbalance characteristic of the dataset, we proceed with the two following approaches: 1) Data Augmentation 2) Class weight. We tried to increase the size of our dataset using the "Tsaug" library. We used various augmentation techniques such as cropping, reversing, time warp and quantization with different parameters. Then to evaluate the augmented data, we again computed the feature correlation of our new dataset to see whether the augmented samples caused any considerable difference in the correlation of features or they are aligned with the original dataset. After performing a number of experiments and using different augmentation techniques, we have found that the augmented data makes dramatic changes in the correlation between features and basically changes the distribution of our dataset. Furthermore, as another evaluation we recognize that accuracies were getting worse in the training phase, by feeding our models with augmented dataset. The same has happened with class weight methods and the obtained accuracies were not satisfactory. As a result, we continued with the standardized dataset.
2. Data Splitting: During experiments, in the first phase, for selecting appropriate models and hyperparameter tuning, we divided the dataset into 3 parts, 20% of data were considered as the test set, and the rest of data were divided into 80% and 20% for training and validation, respectively. The training data used for training the model. The hyperparameter tuning procedure was done using the validation data, and finally the best obtained results were considered for testing on the local test set. However, after selecting the best model and tuning the hyperparameters, in the second phase, we divided the whole data into two parts: training and validation in which we considered 90% images as the training and the rest (10%) as the validation set. Because in this way we were training on more samples and the model can generalize better on the test set of the challenge. For credibility of our experiments and accurate tuning of the models, in the first phase of our experiments we defined a seed for the random function which allows reproducibility of the results. This is how we could fully observe the effect of hyperparameters on the models and compare them after we tuned them.
3. Feature Selection: The feature selection procedures can discard the redundant information and select the prominent subset of features, however, selection of the best subset with respect to the number of feature can be exponential in time, so one of the best approach that now being used in many deep learning task for such optimization problems is the meta-heuristic algorithms. In this work, we implemented from scratch one of the most known algorithms named Genetic algorithms as a binary feature selection approach. This approach creates an 2D population (we considered 10x6 as the population size, the dimension is selected with respect to the number of features we are trying to select), then we initialized the population with random values in [0-1] for each dimension (features), then for performing a binary selection of the dimension we passed the values to a threshold and the values below a certain value were considered as zero, and values above that threshold were considered as 1. Then fit the data based on the selected features and we measured the loss on the validation, and saved the best results for performing the specific operations of the genetic algorithm such as mutation and crossover. Finally we discovered that by removing the feature zero we obtained slightly better results on the validation and test set, however it was not much effective during the submission.
Note: we also studied feature selection based on the correlation between them but as anticipated the corresponding results were not adequate. Because the highest correlation coefficient value (which belonged to the last two features) was not that high(0.7).
4. Window selection: As mentioned by the professor as a hint, we took into account that the current window may not be optimal, so we considered experimenting with the models with different window sizes. First we divided the provided window 36 into equal 9 pieces and obtained more samples. Then we increased the window

size by 6, 4, 3 and 2 pieces, respectively. Furthermore, during the experiments we also tried the window selection by only considering some parts of the provided window, and we discovered that the approximately first half of the data in the provided window was highly important for prediction in comparison to the second half. However, using the first half only, slightly decreased the performance on the test and validation sets and also during the submission. Moreover, based on the performed experiments we made an assumption that the best subset selection in this procedure might not be in order, so we tried to use the aforementioned feature selection approach (Genetic algorithm) for selecting the optimal subset of time legs in the provided window (36 was considered as the dimension), but also this approach was not much effective during the submissions. Finally, we proceed with the provided window.

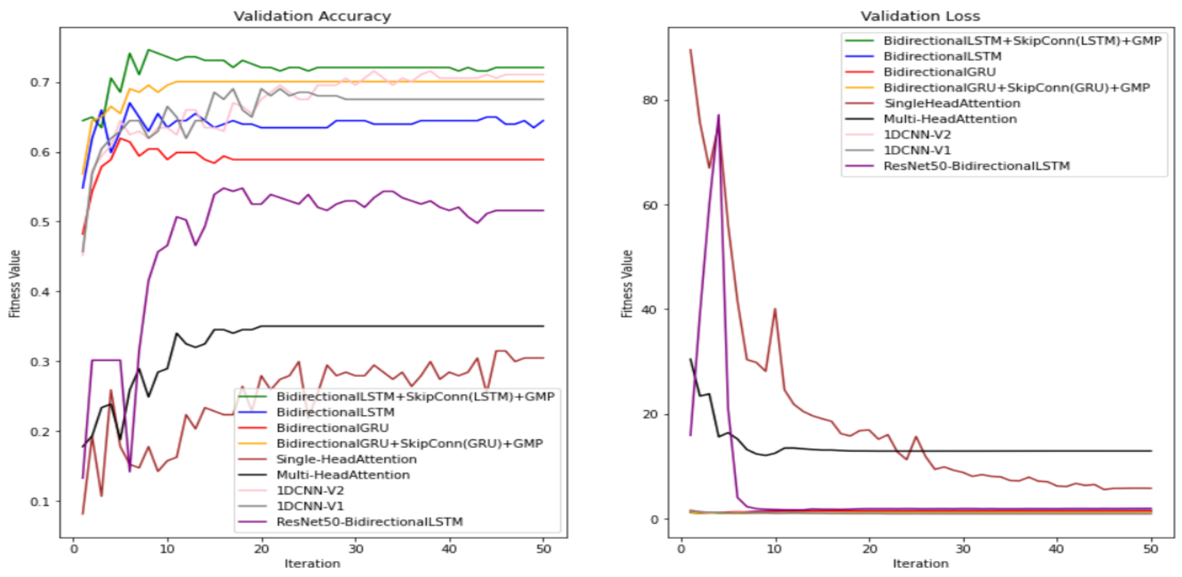
5. LSTM model designing: First we started off by using the implementation provided in the exercise sessions by the professor. Using the basic model we obtain an accuracy around ~64% on the submission. Then we managed to enhance the model's performance by adding skip connections in the architecture. Then we also considered using bidirectional architectures, and combined it with simple LSTM architecture with skip connections. Furthermore, we tried different combinations and architectures such GRU, 1DCNN and LSTM architecture, during the experiments we tried different configurations and layers such as global max pooling (GMP), global average pooling (GAP), dropout, dense, and different activation functions to further optimize our model. The best obtained results were obtained using LSTM and GRU with bidirectional layers and skip connections followed by GMP and dropout layers. Moreover, different architecture such as single and multi head attentions were experimented, but we did not attain the desired performance.

6. 1DCNN model designing: Parallel to LSTM model designing, we managed to design a couple of 1d Conv Net models. We started with a basic model that we learned to build in the exercise sessions and we could obtain an accuracy of ~64% on submission. After carrying out a number of experiments with different architectures, we came up with developing the model by increasing the number of neurons and adding one more layer in addition to using Sigmoid as the activation function of the second and third layer and a ReLu activation function for the later layer. It resulted in a 4% increment in submission accuracy. At last, after hyper parameter tuning and using skip connections, we could obtain an accuracy of ~69% on submission.

7. ResNet model: For further exploration on models, we decided to take our chance on transfer learning approaches especially, ResNet models. Since we could improve the performance of our models using skip connections, we decided to choose Resnet which comprises a number of skip connections. However, there was a problem of dimensional compatibility with our dataset, since ResNet architecture was designed specifically for image processing and trained on the ImageNet dataset. We managed to address it by using an additional LSTM, dense and reshape layers before our ResNet50 model. Moreover, since ResNet is pretrained based on the ImageNet dataset ,we decided to train it from the scratch by putting trainable parameters for all layers equal to False. After performing a couple of experiments and trying different types of layers such 1DCNN, LSTM, GRU, and bidirectional, we could not obtain an accuracy more than ~62% so we did not consider this model anymore.

Figure.1 demonstrates the performance of the experimented models on the validation dataset during training.

All of the described models were trained and tested with batch size 16, training epoch 50, default learning rate (0.001) using the learning rate reduction technique with patience 3 and the reduction factor 0.3 monitoring on the validation loss.



8. Ensemble Model: As the last step we proceeded with the ensemble model approach for obtaining better performance as well as generalizing our final model (since we struggled a lot with overfitting while training our models). Thus we chose our top models (Table 1) to build an ensemble model. For the prediction we used the most voted technique, in which the most voted class is considered as the final prediction of the ensemble model. All the models retrained several times using different distributions of data on up to 90% of it. To be more specific we used a 1DCNN (trained on 3 different data distributions), 1DCNN+SkipConnections (trained on 5 different data distributions), LSTM (trained on 9 different data distributions) and one BiLSTM. We managed to obtain an approximate accuracy of ~74% in the final phase competition. Table.1 summarizes the best obtained models and results during experiments.

Model	Local Accuracy	Submitted Accuracy
BidLSTM_LSTMSkipConn_GMP	~0.72	0.6916
BidGRU_GRUSkipConn_GMP	~0.70	0.6916
BidGRU_GRUSkipConn_GMP-ExcludingFeature0	~0.72	0.6807
1DCNN_v1	~0.71	0.6844
1DCNN_v2(withSkipConnections)	~0.71	0.6965