1. RNN-based approach

The data was provided as sequences of frames requiring classification. This made Recurrent Neural Networks (RNN) perfect for the task. A bidirectional Long Short-Term Memory (bi-LSTM) layer was used as the RNN layer, in order to extract the features from the data, preserving temporal relations. The features were subsequently fed into a linear layer with one output per class in the dataset, representing the score for that particular class.

The dataset featured a few interesting challenges. Its rather small size would give most neural networks a tough time learning meaningful properties while avoiding overfitting to the exact input. To combat the aforementioned issue, we designed our LSTM to be relatively small in size, only including one hidden layer of 128 neurons. Additionally, the provided dataset was heavily imbalanced; a weighted cross-entropy loss criterion, whose weights reflected this imbalance, was used in the training loop. The possibility of using focal loss [1] was investigated, but no noticeable improvement during training was observed.

The coordinates of the data were centered around (0, 0, 0) and normalized to lay within the range [-1, 1], keeping the aspect ratio intact. Xavier initialization [2] was used to initialize the trainable parameters of the network and the Adam optimizer with a learning rate of 0.001 was used in the training process. The data was not fed in batches into the network. We experimented using batches and padding the samples to include the same number of frames but, probably due to the vastly different number of frames between each sample, the results were significantly worse.

The training took place for just under 2 hours on our NVIDIA RTX™ 2060 SUPER GPU with 8GB of video memory, over 3000 epochs (the dataset was kept loaded in RAM). Early results were promising, regularly managing higher than 50% accuracy on both the training and test datasets. The test dataset accuracy, specifically, was closely monitored throughout the training process. With no regularization means (other than the small network size), we had to ensure that the quick drop in training loss and increase in the training set accuracy was not a product of overfitting and that the accuracy of the test set remained close to that of the test set.

Despite getting good results almost immediately, the training proceeded until the loss remained below 0.01, we were posting higher than 90%-100% accuracy almost entirely and felt confident that the network had drawn the right and complete conclusions.

References

[1] Lin, TY, Goyal, P, Girshick, R, He, K, Dollár, P. Focal loss for dense object detection. In: Proceedings of the IEEE international conference on computer vision. 2017, p. 2980–2988.

[2] Glorot, X, Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. In: Proceedings of the thirteenth international conference on artificial intelligence and statistics. JMLR Workshop and Conference Proceedings; 2010, p. 249–256.