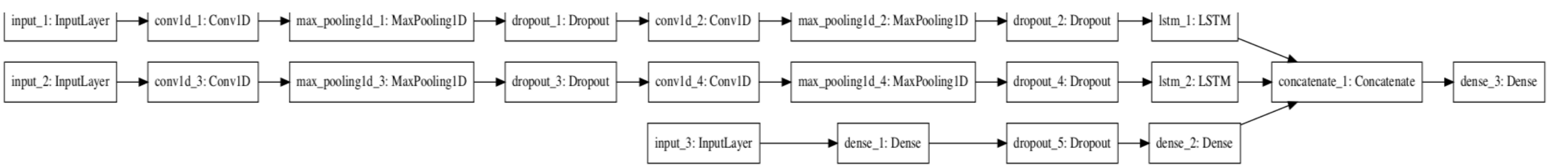
Deep Learning Assignment: Classification of Sleep Signals

Group 41 || Members: Arati Sharma (SNR: 2047448) Aayushi Pandey (SNR: 2038981) || Codalab id: u848144

Pre-processing and feature engineering: We used features from both the raw EEG signals and the spectrograms for this task. We extracted features from the raw signals using Discrete Wavelet Transform (DWT). DWT overcomes the limitations of Fourier transform to analyse non-stationary signals and has been widely employed to analyze EEG signals ([1,2,3,4,5]). While using DWT, it is extremely important to identify the appropriate wavelet type and decomposition level depending on the problem at hand [2]. We selected a DWT decomposition of 6 levels using Daubechies of order 2 (db2) wavelet on raw signals after experimenting with different combinations and examining validation accuracies. Entropy, percentiles (5,25,50,75,95), mean, variance, root mean square values, zero-crossings and mean-crossings were computed for each sub-band [6]. A total of 168 features were extracted from raw signals (84 from each channel). These features were standard scaled. No preprocessing was done on spectrogram data. The underrepresented classes from data were upsampled to deal with class imbalance. The train-validation split was made using stratification to ensure identical class distribution in the splits.

Figure 1. Schematic of the model



Learning algorithms, parameters, and hyperparameter tuning: The deep learning model for this task uses MLP, CNNs and LSTMs. We specifically picked LSTM for modelling the time series properties of the spectrograms because it solves the problem of vanishing gradients by using gated cells which allow the network state to be updated, reset, or to propagate without any modification [7,8]. Also, LSTM has shown great success in processing signals like EEG [9, 10] and ECG [11]. One component of our model consists of two convolution 1d layers (size 30), each taking as input spectrogram from one channel (Fpz/pz). The convolution layer is followed by maxpooling layer (size 4), dropout function (probability 0.2) and a LSTM layer (size 30). Another component is a multi-layer perceptron (MLP) with 2 hidden layers (size 32 and 16) and a dropout function (probability 0.2) taking DWT features as input. The output from the two LSTMs and MLP are concatenated, and then passed through a softmax layer to do the classification. A schematic of the model is shown in Figure 1. We tried different configurations of learning rate, optimisers, layer sizes, epochs and activations. The model was finally trained using relu activation in all layers, Adam optimiser with a learning rate 0.001 and 30 epochs. In another variation, we trained the same model on 25 epochs without upsampling. In the combined + 1 layer variation, a hidden layer of 16 size is added before the softmax layer. The results are shown in Table 1. We started with two separate models (MLP without the dropout layer and Conv/LSTM), and only later decided to combine them. Results obtained from separate models are also reported in Table 1.

Table 1. Performance

Model	Optimiser	Epochs	Upsample	Train accuracy	Val. accuracy	Test Accuracy
Combined	Adam (lr =0.001)	30	Yes	0.8404	0.8236	0.6836
Combined	Adam (lr =0.001)	25	No	0.8292	0.8202	0.6739
MLP only	Adam (lr = 0.0004)	30	No	0.8329	0.8084	0.6653
Combined (+1 layer)	Adam (lr =0.001)	30	Yes	0.8446	0.8349	0.6602
MLP only	Adam (lr = 0.0004)	20	Yes	0.8272	0.8000	0.6420
Conv+LSTM only	Adam (lr =0.001)	70	Yes	0.7871	0.7700	0.5895

Discussion of results: The model Conv+LSTM only had the lowest performance (test acc 59%). Among the separate models, the MLP model with no upsampling had the highest performance (test acc 66%). The combined model with upsampling had the highest performance of all (test acc 68.36%, above baseline). We find that statistical properties of the DWT features improved classification rate, which is consistent with findings in the literature [12]. Upsampling had a negative impact on the performance of MLP only model (test acc 66% vs. 64%) but it had a positive impact on the combined model (test acc 67% vs. 68%). The differences in train and validation accuracies are very low for all of the models (range:0.9%, 2.7%). Therefore, we do not think the model is overfitting on the training set [13]. However, the differences in the test and train accuracies are very high (range: 14%, 18%). This could suggest that the test set has more examples from the difficult classes which our model did not learn to classify very well. Particularly our model doesn't classify very concretely between classes (1,5) and (3,4) as is evident from the confusion matrix (see Appendix).

References

[1] Oropesa, E., Cycon, H. L., & Jobert, M. (1999). Sleep stage classification using wavelet transform and neural network. *International computer science institute*.

[2] Şen, B., Peker, M., Çavuşoğlu, A., & Çelebi, F. V. (2014). A comparative study on classification of sleep stage based on EEG signals using feature selection and classification algorithms. *Journal of medical systems*, 38(3), 18.

[3] Ebrahimi, F., Mikaeili, M., Estrada, E., & Nazeran, H. (2008, August). Automatic sleep stage classification based on EEG signals by using neural networks and wavelet packet coefficients. In *2008 30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society* (pp. 1151-1154). IEEE.

[4] Subasi, A., Kiymik, M. K., Akin, M., & Erogul, O. (2005). Automatic recognition of vigilance state by using a wavelet-based artificial neural network. *Neural Computing & Applications*, 14(1), 45-55.

[5] HAŞİLOĞLU, A. (2001). Rotation-Invariant texture analysis and classification by artificial neural networks and wavelet transform. *Turkish Journal of Engineering and Environmental Sciences*, 25(5), 405-413.

[6] Ahmet Taspinar. (2018, December 21). A guide for using the wavelet transform in machine learning [Blog post]. Retrieved from http://ataspinar.com/2018/12/21/a-guide-for-using-the-wavelet-transform-in-machine-learning/

[7] Alhagry, S., Fahmy, A. A., & El-Khoribi, R. A. (2017). Emotion recognition based on EEG using LSTM recurrent neural network. *Emotion*, 8(10), 355-358.

[8] Won Park, H., Busche, J., Schuller, B., Breazeal, C., & Picard, R. W. (2019). Personalized Estimation of Engagement From Videos Using Active Learning With Deep Reinforcement Learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (pp. 0-0).

[9] Nagabushanam, P., George, S. T., & Radha, S. (2019). EEG signal classification using LSTM and improved neural network algorithms. *Soft Computing*, 1-23.

[10] Wang, P., Jiang, A., Liu, X., Shang, J., & Zhang, L. (2018). LSTM-based EEG classification in motor imagery tasks. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, *26*(11), 2086-2095.

[11] Gao, J., Zhang, H., Lu, P., & Wang, Z. (2019). An Effective LSTM Recurrent Network to Detect Arrhythmia on Imbalanced ECG Dataset. *Journal of healthcare engineering*, *2019*.

[12] Vatankhah, M., Akbarzadeh-T, M. R., & Moghimi, A. (2010, July). An intelligent system for diagnosing sleep stages using wavelet coefficients. In *The 2010 International Joint Conference on Neural Networks (IJCNN)* (pp. 1-5). IEEE.

[13] Chollet, F. (2018). Deep Learning with Python. United States of America: Manning Publications Company.

Appendix

Confusion matrix, classification metrics and graphs for the final (combined model)

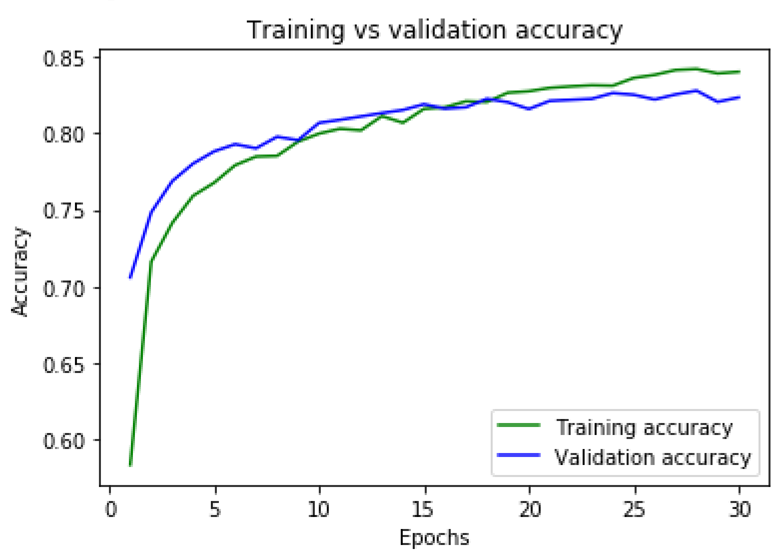
Confusion matrix for training data (True on y axis, predicted on x axis)

	0	**1**	**2**	**3**	**4**	**5**	**Total**	**Accuracy**
0	2064	109	60	16	0	1	2250	0.9173
1	169	1791	33	15	0	242	2250	0.796
2	37	69	2041	94	1	8	2250	0.9071
3	4	9	45	2026	164	2	2250	0.9004
4	0	0	0	397	1852	1	2250	0.8231
5	5	185	0	2	0	2058	2250	0.9147

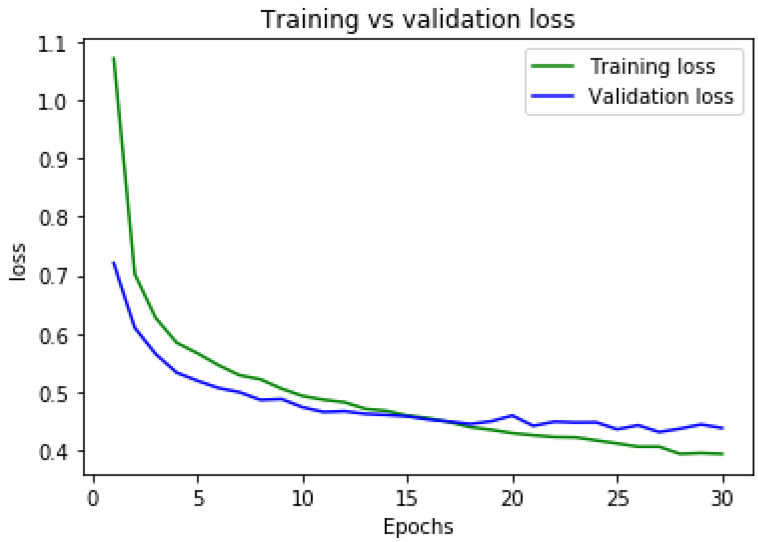
Confusion matrix for validation data (True on y axis, predicted on x axis)

	0	**1**	**2**	**3**	**4**	**5**	**Total**	**Accuracy**
0	641	64	35	9	0	1	750	0.8546
1	60	537	24	9	0	120	750	0.716
2	16	25	661	46	1	1	750	0.8813
3	5	7	25	632	80	1	750	0.8427
4	0	0	2	153	595	0	750	0.7933
5	9	99	2	0	0	640	750	0.8533

Graph training/validation accuracy vs no. of epochs



Graph training/validation loss vs no. of epochs



Classification report

	Training Data			Validation data		
Class	**Precision**	**Recall**	**F1-score**	**Precision**	**Recall**	**F1-score**
0	0.91	0.92	0.91	0.88	0.85	0.87
1	0.83	0.80	0.81	0.73	0.72	0.72
2	0.94	0.91	0.92	0.88	0.88	0.88
3	0.79	0.90	0.84	0.74	0.84	0.79
4	0.92	0.82	0.87	0.88	0.79	0.83
5	0.89	0.91	0.90	0.84	0.85	0.85