Seizure prediction using convolutional neural networks

Iryna Korshunova

email: irene.korshunova@gmail.com

1 Introduction

This document explains a solution for Kaggle Seizure Prediction Challenge, which finished in the 12th place. The goal of the competition was to classify 10 minute intracranial EEG (iEEG) data clips into "Preictal" for pre-seizure data or "Interictal" for non-seizure data segments. To tackle this problem I used convolutional neural networks (convnets) because of the following reasons:

1. EEG signal is nonstationary, so one needs to consider shorter time windows to extract meaningful features. This in turn requires a method for combining information from different blocks to get a prediction for the whole 10 minutes clip. Convnets with convolution through time seemed to be a good approach to deal with this. Moreover, using smaller windows increases the number of features, but because of shared weights number of convnet's parameters remain small relatively to a standard neural network architecture.

2. In the previous competition on seizure detection the winning solution effectively exploited FFT features and correlations between EEG channels, so presumably if one does convolution across all channels on FFT data, convnet can learn correlations in frequency domain automatically, but for now I have no idea what my networks learned.

2 Features

The first preprocessing step was to filter iEEG data to 0.1-180 Hz. Then I used a similar approach as described in [1]: each 10 minutes clip was partitioned into non-overlapping 1(2) minute frames, which were Fourier transformed and resulting amplitude spectrum was divided into 6 frequency bands: delta (0.1–4 Hz), theta (4–8 Hz), alpha (8–12 Hz), beta (12–30 Hz), lowgamma (30–70 Hz), and highgamma (70–180 Hz). Within each band I took a log10 of geometrical mean of the amplitude spectrum over band frequencies. So one 10 minutes data clip with N channels transforms into $N \times 6 \times 10$ image.

I tried two normalization schemes (for each channel separately):

1. Use 6×10 distinct features, so we account for the position in time and frequency

2. Consider only 6 distinct features, for instance, values in delta frequency band from different time frames are of the same feature. So from one example 6×10 we have 10 examples 6×1 each.

Some models from the resulting ensemble were trained with an additional feature: standard deviation of the signal in a particular time frame, so the input would be $N \times 7 \times 10$.

3 Network architecture

The architecture of the model, which appeared to be the best on a public leaderboard is shown on Fig. 1. It receives an input after the 1st normalization scheme, and its first layer performs convolution in a time dimension over all channels and all frequency bins, so the shape of its filters is (64×1). Second convolutional layer is fully-connected with a hidden layer. Rectified linear units are used in all layers, and last 2 layers have dropout of 0.2 and 0.5. On public/private leaderboard this model scored 0.81448/0.76256.

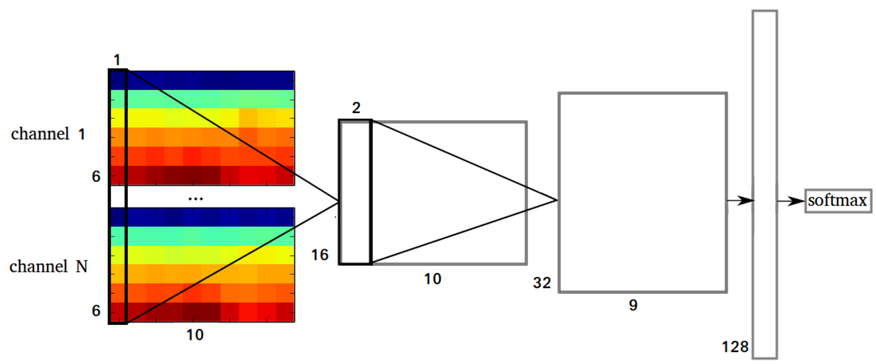

Figure 1: Example of a convnet

Intuitively the location of some feature in time should not be relevant, therefore I tried global temporal pooling with mean, max, min, variance, geometric mean and L2 norm calculated across time axis within each feature map in the 2nd convolutional layer. With global pooling layer previous model would look like on Fig. 2. From my experiments models with global pooling worked best with 2nd normalization scheme, tanh activation in the hidden layer, stride in the 2nd layer and when using standard deviation of the signal as an additional feature.

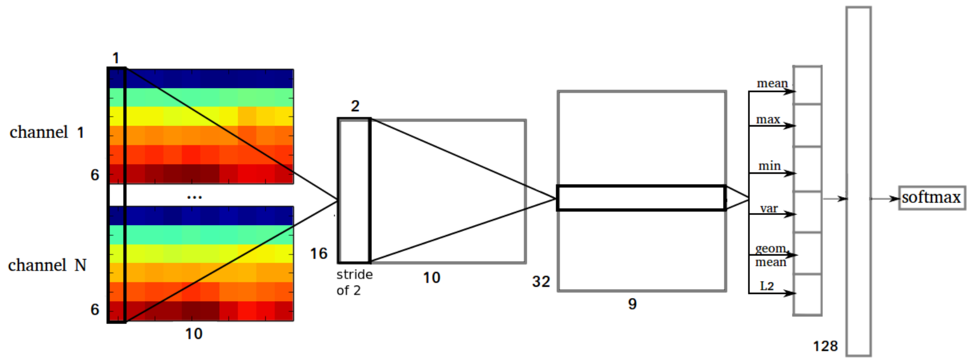

Figure 2: Example of a convnet with global temporal pooling

Public LB scores were misleading by scoring no-global-pooling models higher than models with global pooling layer. It appeared to be completely opposite situation on a private leaderboard. However, public LB was the only source of validation as I used stratified split without keeping the sequences intact, thus getting very optimistic results. I tried to keep the sequences, but for some train-validation splits models were not training long enough (when using early-stopping). Later I started to train my models for a fixed number of updates and using data augmentation by overlapping clips from the same sequence on some number of time frames, e.g. if clip consists of 10 time frames, overlap of 9 frames yields approximately 9 times bigger dataset.

To calibrate the predictions between subjects, I used min-max scaler on test probabilities, which used to improve the score on ∼0.015.

4 Model averaging

The submission, which finished the 12th place and public/private score of 0.81292/0.78513 was an average prediction of models with global pooling layer and different combinations of parameters: number of kernels in convolutional layers, number of hidden units, amount of dropout, normalization scheme, number of time frames, overlap between clips, use of test data during normalizing the training data etc.

References

[1] Stead SM Brinkmann B Vasoli V Crepeau D Vite CH Sturges B Ruedebusch V Mavoori J Leyde K Sheffield WD Litt B Worrell GA Howbert JJ, Patterson EE. Forecasting seizures in dogs with naturally occurring epilepsy, 2014.