Stride and Pooling Performance for EEG Classification

Agustin Marinovic Sfeir
University of California, Los Angeles
Los Angeles, CA, 90024
amarinovic@ucla.edu

Abstract

ChronoNet is a deep neural network which uses inception convolutional layers and residual recurrent layers to classify electroencephalogram (EEG) time-series data. In this project the effectiveness of convolutional dimensionality reduction along the temporal axis for regularization. For this three methods are tested: using convolutions with stride of 2, using a max pool layer of size 2 or using an average pooling layer of size 2. Using variations of ChronoNet, the regularization power of max pooling is demonstrated as well as the speedup of using a stride of 2.

1. Introduction

ChronoNet [6], a deep neural network for EEG classification, uses inception modules [7] of 1D convolutions and residual gated recurrent units (GRU) [3] for EEG abnormality classification. In the work by Roy et al., the ChronoNet uses 3 convolutions of pool size 2, 4 and 8, with a stride of 2. Each convolution reduces the size of the temporal axis by half.

In this project three networks are tested on efficacy of EEG classification. The models (refer to Appendix A) are trained and tested on the dataset, provided by Brunner et al. [2], consists of EEG data using a 10-20 electrode placement of 9 patients. For each patient there are 288 trials of EEG measurements while moving a specific part of the body among: left hand, right hand, legs or tongue. Each data-point consists of 4 seconds of 250 Hz sampling rate (1000 time-steps).

Pooling has been used in convolutional neural networks (CNN) for dimensionality reduction and regularization [5]. By using a max pool layer of size 2, the dimensionality of the input is halved. The benefit of the max pool operation is that it adds temporal invariance. Temporal invariance is beneficial for ChronoNet as it adjusts to EEG oscillatory nature. In the default ChronoNet, dimensionality reduction is done via a stride of 2 during convolutions. In this project max pooling is compared to using a stride of 2 for generalization and temporal dimensionality reduction. Average pooling's performance is also compared to max pooling and striding.

2. Results

After running the three models over 30 epochs, the following information about runtime, loss and accuracies was obtained.

2.1 Runtime

The training time of each model can be seen in Table B1. The default ChronoNet runs much faster than both pool networks. The clear reason of this result is that with a stride of 1, the convolution layers perform double the operations. This is also reflected in the time per epoch, with the pooling models taking nearly 30% longer per epoch to execute.

2.2 Loss

For training, in all three models the loss decreased in a linear manner. It is clear that in all three models there was plenty of training left. For the default ChronoNet, on the last epochs the validation gradient started to saturate. Starting at epoch 21 there is not much decrease in loss. For the max pool model validation loss decreased constantly and fairly smoothly. On the last epochs it started to change more abruptly, however it still kept decreasing. For the average pool model, the validation loss decreased in a linear direction, but with sharper changes from epoch to epoch. Figures B1, B2 and B3 show the different loss descent.

2.3 Accuracy

The ChronoNet's validation accuracy increased with training accuracy. It is important to note however that, except at the beginning, validation accuracy is always less than training accuracy. Comparing Figure B4 and B5 show different results. In the case of the max pool model, validation accuracy is generally better than training accuracy. The average pool model's validation accuracy tended to be lower than training accuracy. Both Figure B3 and B6 have similar sharp turns. Sequential epochs tend to have opposite relative slopes.

At test time all models had similar results. As seen in Table B3, the three model's test accuracy were around 0.4. It implies then, that after 30 epochs all 3 models reached similar results, however, the ChronoNet trained 30% faster than the pooling networks. In all models validation and testing accuracy seemed to be on par with training accuracy. This can be attributed to the spatial dropout, the GRU dropout on input and recurrent matrix, and the regularization that stacked convolutions generate.

3. Discussion

The training and testing of the models on the EEG data brings interesting insights about the inception residual network that is ChronoNet. First of all, the fact that nearly 90% of the parameters are in the convolutional layers, hints towards most of the processing being convolutional rather than recurrent. The use of inception layers of varying sizes seems to capture the essence of oscillatory EEG data. A remarkable result from the stacked inception modules, is that minimal preprocessing was done. The only transformations applied were a high pass Butterworth filter and standardization along the temporal axis. The preprocessing of the data occurs implicitly on the convolutions.

All three models showed similar results. After 30 epochs all three had similar loss, training, validation and testing accuracy. A major insight from ChronoNet is that it trained faster, by an important margin, over both pooling models. The result is expected (as seen in Table B1) given that a convolution of stride 2 does half the operations a unit stride convolution does. The dimensionality of the layers is then reduced. It is also important to notice that the validation loss started saturating at the later epochs. Although not completely clear due to sharp changes, a general deviation can be appreciated in figure B1. This compares to both pooling models for which the validation loss remained similar to training loss.

It can be speculated that if the model is trained for more epochs (which wasn't possible due to time and budget constraints) the pooling models will saturate at a much higher accuracy than the ChronoNet. It also has to be taken into account the time differential. ChronoNet runs faster than the pooling models. If left running the same time both pooling models ran (two hours each for example), is using better regularizations ChronoNet could converge faster. There are three main approaches, that if used together can grant higher results. First, running for more epochs. Second is increasing convolutional drop rate and GRU drop rate. Third is more data augmentation. For this project windowing was done with a stride of 50 time-steps. Comparing to Ball [1] windowing with stride 1, the dataset used is nearly 50 times smaller. Memory and time constraints didn't allow for this. Even halving the stride to 25 double the samples but also double memory used and runtime per epoch.

One hypothesis of the performance of the max pool over the vanilla ChronoNet is the effect of dropout regularization. The use of a max pooling layer regularizes by dropping half the samples. Even though a stride of two achieves the same, the max pooling generates a richer distribution of values, for which half are dropped. This is added to the spatial dropout layer that follows the max pool layer.

In general max pool and average pool models had similar results. Comparing loss and accuracy plots for both, max pooling might be preferred due to the smoothness of the graph. The sharp subsequent changes of the average pool compared to the max pool. This effect can be hypothesized to the nature of the average pooling layer. If the two entries of the average pool have vastly different magnitude, a small number and a big number, the gradient will be either too big or too small, adding noise to the gradient.

The three main conclusions reached in this project are, average pooling does not perform as well, or smoothly, as a max pooling convolutional network. Using a stride of 2 is much faster than pooling. Using pooling layer regularizes more than striding.

References

[1] T. Ball et al. Deep learning with convolutional neural networks for EEG decoding and visualization. Human Brain Mapping, volume 38 issue 11, 2017.

[2] C. Brunner et al. BCI Competition 2008 - Graz Dataset A. BCI Competition, 2008.

[3] K. Cho et al. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. arXiv:1406.1078v3, 2014.

[4] D. P. Kingma and J. Ba. Adam: A Method for Stochastic Optimization. ICLR, 2014.

[5] M. Ranzato, K. Jarrett, K. Kavukcuoglu, Y. LeCun. What is the Best Multi-Stage Architecture for Object Recognition? ICCV, 2009.

[6] S. Roy, I. Kiral-Kornek, and S. Harrer. ChronoNet: A Deep Recurrent Neural Network for Abnormal EEG Identification. IJCAL, 2018.

[7] C. Szegedy et al. Going Deeper with Convolutions. CVPR, 2015.

Appendix A: Methods

1. Preprocessing:

For all networks, minimum preprocessing was done. This aims to test the effectiveness of the network on data without handpicked augmentations or cleaning. Three preprocessing techniques were applied: high-pass filtering, standardization and windowing.

First, all sample were high-pass filtered with a 4 Hz, 3rd order digital Butterworth filter, as done by Ball [1]. The 4 Hz was chosen as to remove delta waves and noise.

Second, all samples were standardized to zero mean and unit norm. The mean and variance are calculated from all training samples along the temporal axis. The mean and variance vector was then used to standardize both training and testing sets.

Third, windowing was used as means of data augmentation and temporal reduction. Since EEG are waves, there must be a repeating pattern in the data. Windowing was applied with a window size of 500 (2 seconds) and a stride of 50 (0.2 seconds). This greatly increased the dataset while regularizing the network on oscillatory start point.

2. ChronoNet architecture

The first model used is the default ChronoNet. The architecture consists of 3 inception modules of 1D convolutions. Each module has 32 channels. The inception module has kernel sizes of 2, 4 and 8, all with a stride of 2. The convolutional layers use same padding. The activation used for the convolutional layers is the ReLU. This was chosen due to its speed without impacting performance. The convolutional layers have 301,600 parameters, 88% of the total number of parameters. Each inception module is followed by a batch normalization layer and by a spatial dropout layer of value 0.6.

Following the inception modules are 4 residual GRU cells with 32 units each. The GRU cells are connected in a feed forward manner, such that the output of a GRU is the input of all subsequent GRUs, as seen in figure A1. The residual architecture allows for flexibility in model complexity. The low number of units makes training faster and reduces the number of parameters in the recurrent cells. The input activation of the GRU modules is a tanh. This was chosen as it saturates at large magnitudes and has a linear behavior near zero. The recurrent activation is a hard softmax. This activation is used due to its linearity and saturation at high values. One test run used a ReLU and the loss quickly became NaN. A hard softmax is used over a softmax as the extra computation doesn't improve performance significantly. Each GRU layer has a kernel and recurrent dropout of 0.31.

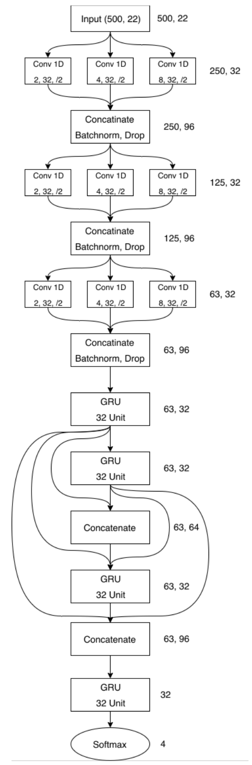

Figure A1: ChronoNet architecture with batch normalization and spatial dropout. The network has 3 inception modules and 4 GRUs connected in a residual way.

3. Max pool ChronoNet

The max pool ChronoNet is a variant architecture which differs in two aspects: uses a stride of 1 in convolutions and has a max pool layer before every batch normalization layer. Dropout is still used after every batch normalization layer to be consistent across models. All activation functions, drop rates and number of units remains constant.

4. Average pool ChronoNet

The average pool ChronoNet is equal to the max pool ChronoNet, except it uses average pooling instead of max pooling. As above, dropout, activations and number of units are maintained constant.

5. Training and testing

The network was trained with data from all 9 patients. The three networks were tested on 30 samples from each patient of the original dataset (which then were preprocessed), for a total of 270 samples before windowing. A validation set is used which consists of 20% of the training set (before windowing). All models are trained with a batch size of 64, learning rate of 0.0012 and GRU regularization of 0.0001. All models are trained using Adam [4] optimizer across 30 epochs.

Appendix B: Results

1. Training time per model:

Model	Total train time (minutes)	Time per epoch (seconds)
ChronoNet	54.68	108
Max pool	72.81	144
Average pool	72.23	143

Table B1: Train time and time per epoch for all three models.

1. Training and validation loss

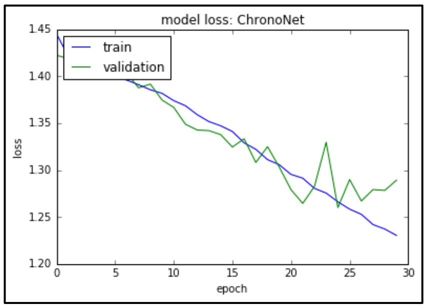

Figure B1: Training and validation loss for ChronoNet.

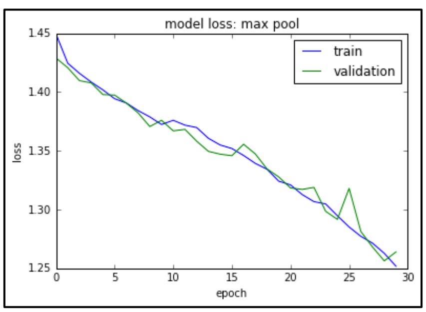

Figure B2: Training and validation loss for max pool network.

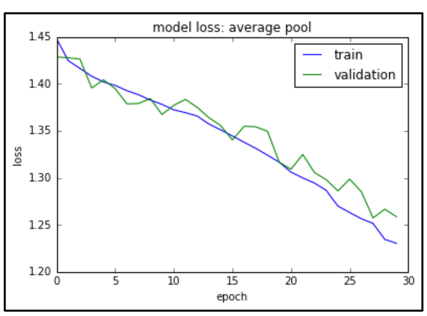

Figure B3: Training and validation loss for average pool network.

Model	Minimum training loss	Minimum validation loss
ChronoNet	1.220	1.263
Max pool	1.253	1.256
Average pool	1.230	1.257

Table B2: Minimum training and validation loss per model

2. Training, validation and testing accuracy

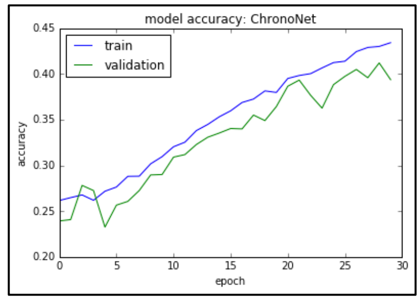

Figure B4: Training and validation accuracy for ChronoNet.

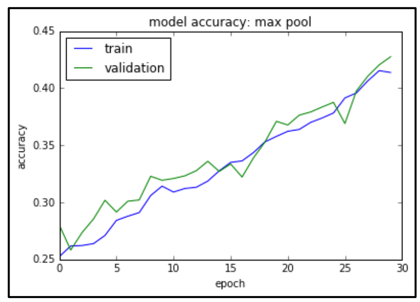

Figure B5: Training and validation accuracy for max pool network.

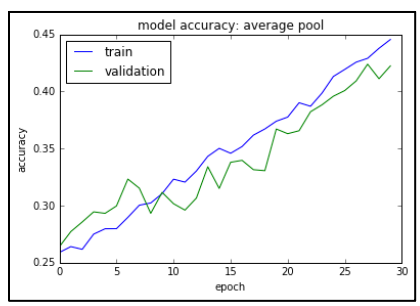

Figure B6: Training and validation accuracy for average pool network.

Model	Max train accuracy	Max validation accuracy	Testing Accuracy
ChronoNet	0.441	0.412	0.418
Max pool	0.415	0.427	0.397
Average pool	0.445	0.424	0.410

Table B3: Training, validation and testing acuracies per model.