0.1 Usage

Overview: The presented software consists of 3 components:

- A set of functions for parsing the mat-files to python, extracting the desired data from the structure, pre-processing
- A set of handcrafted neural network models, a common base class for them and the implementation of a Neural Architecture search for Regression
- A set of functions to visualize the results of the handcrafted networks and the architecture search.

For additional visualizations of hand-crafted models, navigate in a shell to the path where the weights are saved (should contain the folder 'tensorboard-logs') and execute the command " tensorboard –logdir="tensorboard-logs" –port=6006" and access the web interface with your browser at localhost:6006

Configuration The paths of the data folder and the one to store model weights and metrics to, the electrodes and the frequencies to use, training rates and duration can be configured in the file config.py.

Adding a new model requires the user to inherit from the AbstractNet class and then implement the build function and the __init__ function with the provided arguments:

```
def __init__(self, model_out_dir, frequency, electrodes):
    """
    Args:
        model_out_dir: directory to save model to
        frequency: the frequency to be used (shape[1])
        electrodes: the electrodes to be used (shape[0])
    """
    super(DeepDenseNet, self).__init__('WhatsTheNameOfYourModel?', model_out_dir, frequency, electrodes)
```

The following is a minimal example. For the complete documentation for creating a model see https://keras.io/models/model/

```
def build(self):
    """
    constructs a model using 1 Dense layer
    Returns:
        the constructed model
    """
    a = Input(self.input_shape)
    x = Flatten()(a)
    b = Dense(1, activation='relu', kernel_initializer='glorot_normal', bias_initializer='zeros', name='1fc2')(x)
    self.model = Model(a, b)
    super().build()
```

Changing the data format requires the user to extend preprocessing according to the .mat struct of the data. It should however be similar compared to the included preprocessor if data is in the frequency domain, e.g. fourier transformed.

0.2 Methodology

The basic workflow was derived from [Tzallas et al., 2007]

0.2.1 Pre-Processing

The provided data contains the coherence spectrum of the fourier transformed recorded EEG and audio stimulus. The provided labels are comprehension scores between $[0, 1]$ in steps of 0.2, where each step means an understood word.
Below is the covariance matrix of the coherence spectrum and the comprehension scores.

```
Mean Covariance between the Electrode-Audio Coherence spectrum at frequency 5 Hz and the Comprehension Scores
[[1.00992995 0.00153135]
 [0.00153135 0.13221692]]


Mean Covariance between the Electrode-Audio Coherence spectrum at frequency 10 Hz and the Comprehension Scores
[[1.00728738 0.00286027]
 [0.00286027 0.13221692]]
```

Thus no pre-processing in terms of signal processing was possible.
Data was extracted from the provided .mat-files, filtered to only use user-defined which were determined by cluster analysis of Dr. Strauß or all electrodes and only a certain frequence or all frequences. The width of the frequency window was the provided value in Hz ± 1, which left 5 channels, if a frequency was specified. The values in the coherence spectrum were then standardised (z-scores).

0.2.2 Classification: Neural Architecture Searching for Regression

Depending on the user input the shape of the input into the neural networks varied between $(n, 64, 101)$, $(n, 64, 5)$, $(n, 6, 101)$, $(n, 6, 5)$, thus different network architectures are required to accept different inputs [Abadi et al., 2015]. This requires the programmer to train at least one model per input shape and in some cases to adjust the architecture when the results differ [Glorot and Bengio, 2010]. Architecture search can be a quite time consuming process when done by hand and involves the tuning not only of the architecture but also hyperparameters like the learning rate. Thus for each input shape every model would have to be trained a couple of times, which may consum up to several hours per model per input shape [Bengio, 2012].

A promising approach to automate architecture search and hyper-parameter tuning is the Neural Architecture Search [Jin et al., 2018]. It uses bayesian statistics over the metrics of the composed networks to infer an optimal model configuration.

0.2.3 Results

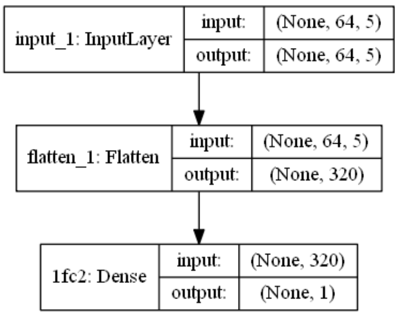
MSE	SmallDense	WideDense	DeepDense	MediumConv1D	NAS
5 Hz	0.1578	0.153	0.1401	0.1390	0.126088
10 Hz	0.16263	0.1542	0.14183	0.1400	0.14444911

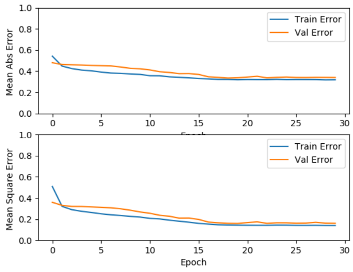
0.3 Appendix: Logs and Plots

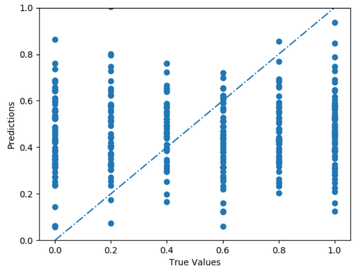
Due to an error of mine (passed 2 Metrics: MAE and MSE, the latter was also the loss function), the MAE is shown as MSE wrongly!

5 Hz component

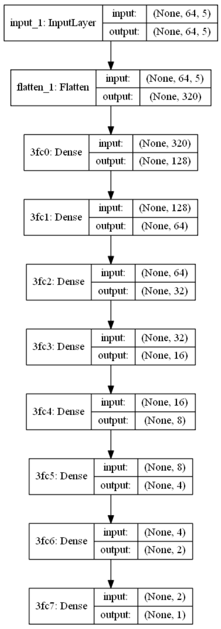
```
==========================Start of Log==========================
Trained model SmallDenseNet-1.json
Sunday March 31,2019 10:33PM
Dataset dir: C:\Users\Fabi\ownCloud\workspace\uni\7\neuroling\neuroling_project\data\v1
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         (None, 64, 5)             0
_________________________________________________________________
flatten_1 (Flatten)          (None, 320)               0
_________________________________________________________________
1fc2 (Dense)                 (None, 1)                 321
=================================================================
Total params: 321
Trainable params: 321
Non-trainable params: 0
_________________________________________________________________
__________Parameters:______________
Batch size     : 128
Epochs         : 10000
Learning rate  : 0.02
__________Metrics__________________
Loss          : 0.15788422232073857
MSE       : 0.3429352627324231
==========================End of Log==========================
```

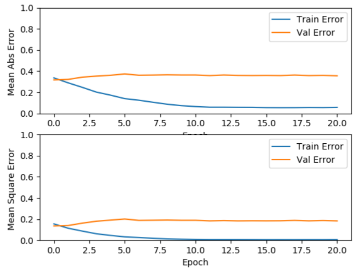


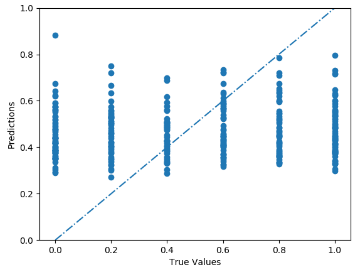




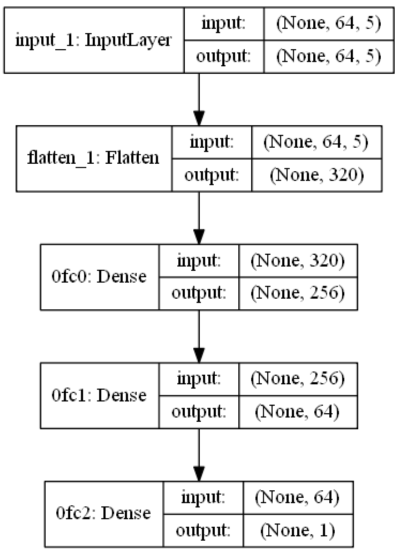

```
==========================Start of Log==========================
Trained model DeepDenseNet-1.json
Sunday March 31,2019 10:34PM
Dataset dir: C:\Users\Fabi\ownCloud\workspace\uni\7\neuroling\neuroling_project\data\v1
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         (None, 64, 5)             0
_________________________________________________________________
flatten_1 (Flatten)          (None, 320)               0
_________________________________________________________________
3fc0 (Dense)                 (None, 128)               41088
_________________________________________________________________
3fc1 (Dense)                 (None, 64)                8256
_________________________________________________________________
3fc2 (Dense)                 (None, 32)                2080
_________________________________________________________________
3fc3 (Dense)                 (None, 16)                528
_________________________________________________________________
3fc4 (Dense)                 (None, 8)                 136
_________________________________________________________________
3fc5 (Dense)                 (None, 4)                 36
_________________________________________________________________
3fc6 (Dense)                 (None, 2)                 10
_________________________________________________________________
3fc7 (Dense)                 (None, 1)                 3
=================================================================
Total params: 52,137
Trainable params: 52,137
Non-trainable params: 0
_________________________________________________________________
__________Parameters:______________
Batch size     : 128
Epochs         : 10000
Learning rate  : 0.02
__________Metrics__________________
Loss          : 0.14017007822935293
MSE       : 0.32465702725972745
==========================End of Log==========================
```

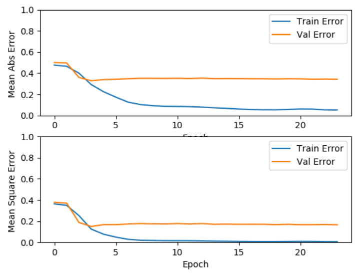


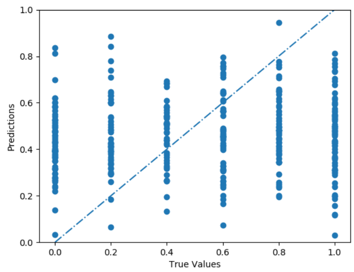





```
=====================Start of Log=====================
Trained model WideDenseNet-1.json
Sunday March 31,2019 10:34PM
Dataset dir: C:\Users\Fabi\ownCloud\workspace\uni\7\neuroling\neuroling_project\data\v1
-----------------------------------------------------------------
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         (None, 64, 5)             0
-----------------------------------------------------------------
flatten_1 (Flatten)          (None, 320)               0
-----------------------------------------------------------------
0fc0 (Dense)                 (None, 256)               82176
-----------------------------------------------------------------
0fc1 (Dense)                 (None, 64)                16448
-----------------------------------------------------------------
0fc2 (Dense)                 (None, 1)                 65
=================================================================
Total params: 98,689
Trainable params: 98,689
Non-trainable params: 0
-----------------------------------------------------------------
__________Parameters:______________
Batch size    : 128
Epochs        : 10000
Learning rate : 0.02
__________Metrics__________________
Loss          : 0.15357901974220498
MSE      : 0.33593259657049457
=====================End of Log=====================
=================================================================
```

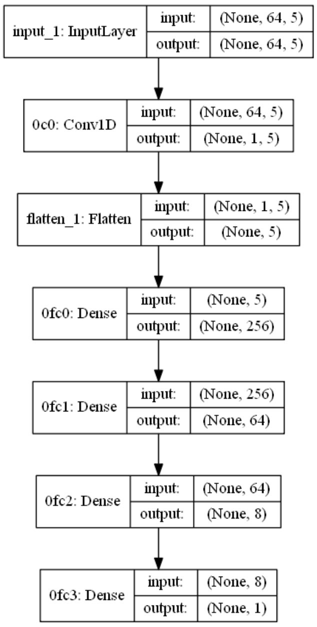


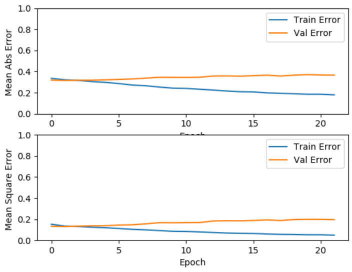


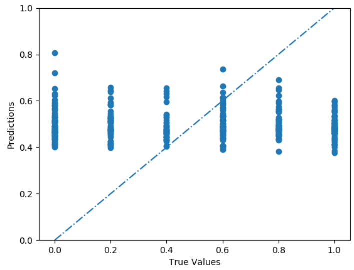


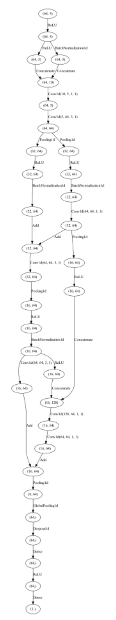

```
=====================Start of Log=====================
Trained model MediumConv1DNet-1.json
Sunday March 31,2019 10:34PM
Dataset dir: C:\Users\Fabi\ownCloud\workspace\uni\7\neuroling\neuroling_project\data\v1
-----------------------------------------------------------------
Layer (type)                 Output Shape              Param #
=================================================================
```

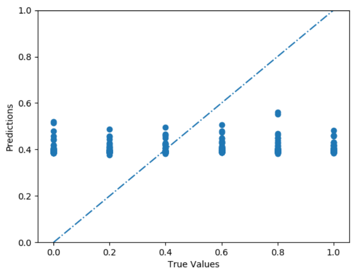
```
input_1 (InputLayer)          (None, 64, 5)            0
_________________________________________________________________
0c0 (Conv1D)                  (None, 1, 5)             1605
_________________________________________________________________
flatten_1 (Flatten)           (None, 5)                0
_________________________________________________________________
0fc0 (Dense)                  (None, 256)              1536
_________________________________________________________________
0fc1 (Dense)                  (None, 64)               16448
_________________________________________________________________
0fc2 (Dense)                  (None, 8)                520
_________________________________________________________________
0fc3 (Dense)                  (None, 1)                9
=================================================================
Total params: 20,118
Trainable params: 20,118
Non-trainable params: 0
_________________________________________________________________
__________Parameters:______________
Batch size     : 128
Epochs        : 10000
Learning rate : 0.02
___________Metrics__________________
Loss           : 0.13900083992522577
MSE       : 0.3265306901380506
====================================End of Log====================================
==================================================================================
__________________________________________________________________________________
```







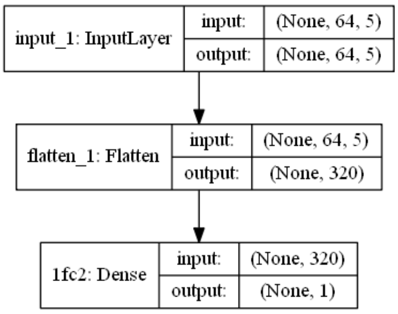


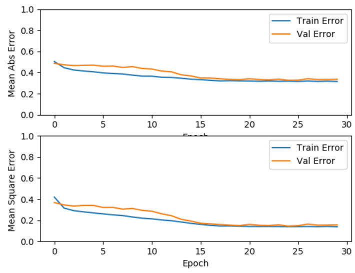


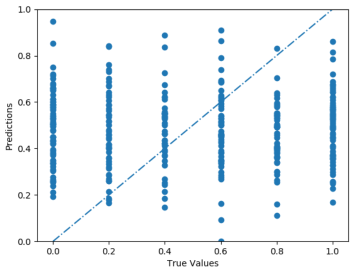
10 Hz Component

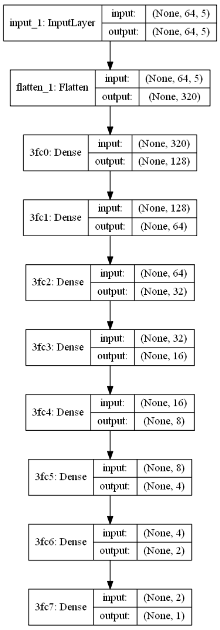
```
====================================Start of Log====================================
Trained model SmallDenseNet-2.json
Sunday March 31,2019 10:29PM
```

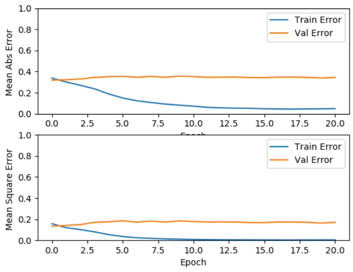
```
Dataset dir: C:\Users\Fabi\ownCloud\workspace\uni\7\neuroling\neuroling_project\data\v1
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         (None, 64, 5)             0
_________________________________________________________________
flatten_1 (Flatten)          (None, 320)               0
_________________________________________________________________
1fc2 (Dense)                 (None, 1)                 321
=================================================================
Total params: 321
Trainable params: 321
Non-trainable params: 0
_________________________________________________________________
__________Parameters:_____________
Batch size      : 128
Epochs         : 10000
Learning rate : 0.02
__________Metrics_________________
Loss            : 0.1626390130016845
MSE        : 0.34865260744370474
==========================================End of Log==========================================
==============================================================================================
______________________________________________________________________________________________
```

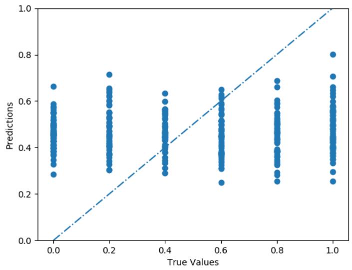






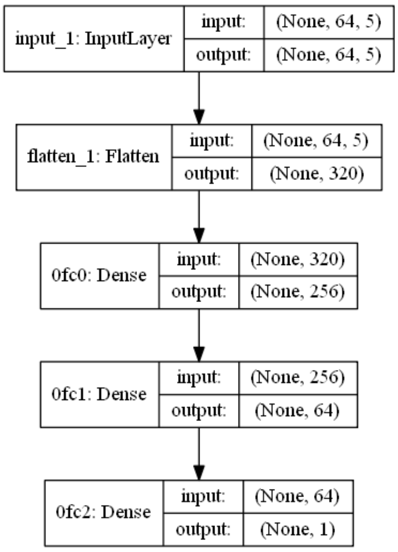


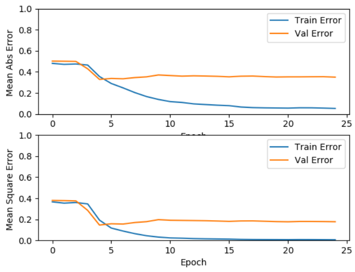


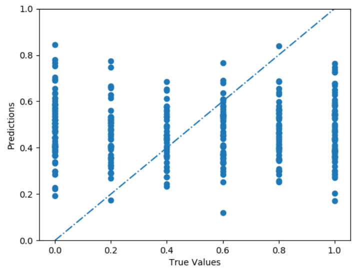



```
=========================================Start of Log=========================================
Trained model WideDenseNet-2.json
Sunday March 31,2019 10:30PM
Dataset dir: C:\Users\Fabi\ownCloud\workspace\uni\7\neuroling\neuroling_project\data\v1
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         (None, 64, 5)             0
_________________________________________________________________
flatten_1 (Flatten)          (None, 320)               0
_________________________________________________________________
0fc0 (Dense)                 (None, 256)               82176
_________________________________________________________________
0fc1 (Dense)                 (None, 64)                16448
_________________________________________________________________
0fc2 (Dense)                 (None, 1)                 65
=================================================================
Total params: 98,689
Trainable params: 98,689
Non-trainable params: 0
_________________________________________________________________
__________Parameters:_____________
Batch size      : 128
Epochs         : 10000
```

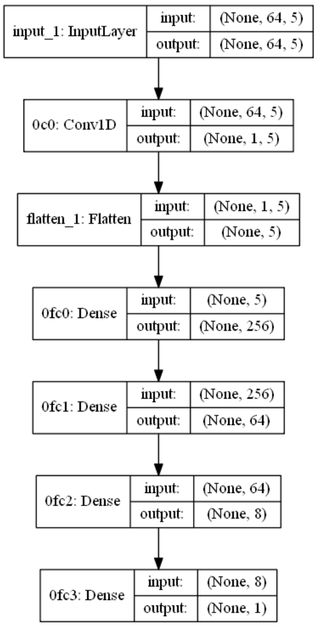
```
Learning rate : 0.02
___________Metrics___________________
Loss            : 0.154226843073878
MSE        : 0.3378447390705175
===========================================End of Log===========================================
================================================================================================
________________________________________________________________________________________________
```

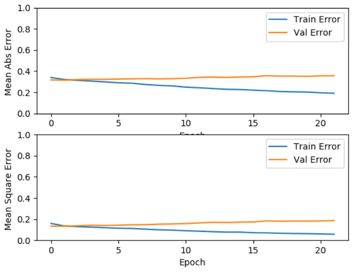


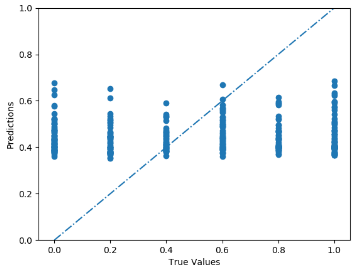


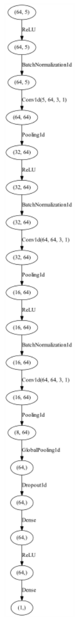


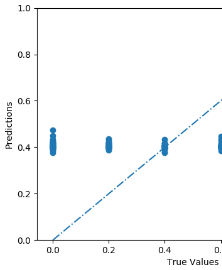

```
===========================================Start of Log=========================================
Trained model MediumConv1DNet-2.json
Sunday March 31,2019 10:30PM
Dataset dir: C:\Users\Fabi\ownCloud\workspace\uni\7\neuroling\neuroling_project\data\v1
-----------------------------------------------------------------
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         (None, 64, 5)             0
-----------------------------------------------------------------
0c0 (Conv1D)                 (None, 1, 5)              1605
-----------------------------------------------------------------
flatten_1 (Flatten)          (None, 5)                 0
-----------------------------------------------------------------
0fc0 (Dense)                 (None, 256)               1536
-----------------------------------------------------------------
0fc1 (Dense)                 (None, 64)                16448
-----------------------------------------------------------------
0fc2 (Dense)                 (None, 8)                 520
-----------------------------------------------------------------
0fc3 (Dense)                 (None, 1)                 9
=================================================================
Total params: 20,118
Trainable params: 20,118
Non-trainable params: 0
-----------------------------------------------------------------
__________Parameters:______________
Batch size      : 128
Epochs          : 10000
Learning rate : 0.02
___________Metrics___________________
Loss            : 0.14003431685053544
MSE        : 0.32515802600480226
===========================================End of Log===========================================
================================================================================================
________________________________________________________________________________________________
```


input_1: InputLayer
input: (None, 64, 5)
output: (None, 64, 5)
0c0: Conv1D
input: (None, 64, 5)
output: (None, 1, 5)
flatten_1: Flatten
input: (None, 1, 5)
output: (None, 5)
0fc0: Dense
input: (None, 5)
output: (None, 256)
0fc1: Dense
input: (None, 256)
output: (None, 64)
0fc2: Dense
input: (None, 64)
output: (None, 8)
0fc3: Dense
input: (None, 8)
output: (None, 1)


Mean Abs Error
Train Error
Val Error
Mean Square Error
Epoch


Predictions
True Values




Predictions
True Values

Bibliography

[Abadi et al., 2015] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

[Bengio, 2012] Bengio, Y. (2012). Practical recommendations for gradient-based training of deep architectures. *Neural Networks: Tricks of the Trade*, page 437–478.

[Glorot and Bengio, 2010] Glorot, X. and Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In Teh, Y. W. and Titterington, M., editors, *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 249–256, Chia Laguna Resort, Sardinia, Italy. PMLR.

[Jin et al., 2018] Jin, H., Song, Q., and Hu, X. (2018). Efficient neural architecture search with network morphism. *CoRR*, abs/1806.10282.

[Tzallas et al., 2007] Tzallas, A. T., Tsipouras, M. G., and Fotiadis, D. I. (2007). Automatic seizure detection based on time-frequency analysis and artificial neural networks. *Computational Intelligence and Neuroscience*, 2007:313 – 333.