A Standardised Benchmarking of Denoising Auto-Encoders in fNIRS Systems

Konstantinos Agusti Demiris
First Year Undergraduate Student
Physiological Computing and Intelligence Lab, UCL
UCL Undergraduate Research Program
under the supervision of
Professor Youngjun Cho
and guidance of
Zikun Quan

18 July, 2024

Abstract

We investigate the application of Denoising Auto-Encoders in the removal of confounding physiological noise from fNIRS signals, which to the best of our understanding has not yet been covered in any previous study. Among five different common implementations from literature, and two newly proposed variations, we achieved significant increases in signal quality metrics including signal-to-noise ratio and contrast-to-noise ratio, and a 93% classification rate on experimental data from unseen participants, with unknown activation periods and no additional sensors.

We show the potential of further development and usage of these techniques in supporting the development of lighter and less intrusive fNIRS systems and provide an open-source toolkit for signal denoising, data generation and performance benchmarking, both as a proof of concept and as a foundation for further studies.

keywords: Denoising Auto-Encoder, functional Near-Infrared Spectroscopy, Data Processing, Deep Learning, Brain Computer Interface, Physiological Computing

1 Introduction:

In recent years, there has been a rise in deep learning based approaches to classification and processing in fNIRS systems [1], due to the unique benefits that deep learning has over traditional methods of analysis that are widely used [1, 2]. They achieve similar, or superior performance in classification and processing, even with smaller datasets [1], and minimise the amount of expert knowledge required to use the systems [2].

1.1 Functional Near Infrared Spectroscopy

Functional Near Infrared Spectroscopy is a well-established non-invasive optical imaging technique that uses near infrared light to measure regional changes in cerebral blood-oxygenation levels [1, 7, 8]. Commonly used in clinical settings [14] as well as research, one prominent trend in recent years has been its increasing usage in the assessment and characterisation of task-related cortical function [7]. When a patient is presented with a stimulus, we can estimate changes in local brain activity through a phenomenon known as neurovascular coupling [14], in which cerebral metabolism increases in response to neuronal activity. This results in increased cerebral blood flow and so a detectable change in optical absorption, on which we can use the modified Beer-Lambert Law to estimate oxygenated and deoxygenated haemoglobin concentrations [8]. The particular signal representing these changes is known as the haemodynamic response function.

However, single event-related analysis can be difficult in practice due to contamination by both systemic physiological signals and general signal quality issues that contribute to a low signal-to-noise ratio (SNR) in fNIRS measurements [8], such as the haemodynamics of the peripheral circulatory system in the scalp, low scalp connectivity due to sensor displacement with motion, or changes in other factors affecting Cerebral Blood Flow [1, 2, 7]. These are numerous and difficult to simultaneously measure or quantify, including but not limited to cardiac output, blood pressure (BP), inter-cranial pressure, respiration rate and so forth. [7, 13]

1.2 Auto-Encoder Architecture

Auto-Encoders are a form of unsupervised neural network architecture that encode (compress) input data in a set of latent space features (the code) through a dimensionality reduction process, then reconstructs the input from those features. Auto-Encoders are considered to be "Self-Supervised", as they are trained by minimising their reconstruction error against their own unlabelled input, based on a loss function. Denoising Auto-Encoders are a specialised form of Auto-Encoders that intentionally corrupt the input data using noise before passing it through the model, then minimise the reconstruction error against the uncorrupted representation. They've seen extensive usage in Anomaly detection and Image Denoising, including in practical medical imaging for enhancing the signal quality of MRI images.

Auto-encoders describe a wide category of models, as the particular hidden layers used in both the encoder and decoder depend on the specific implementation of the Auto-Encoder. Some commonalities among this broad classification is that the encoding process is a form of lossy compression, with the amount of loss being dependent on the complexity of the signal and the number of features in the code (the size of the bottleneck), causing underfitting. On the other hand, an excessively large bottleneck size can cause the model to learn the identity function of the data, resulting in overfitting and poor generalisation. Similarly, a model with too many layers can also overfit. That is to say that the performance of an auto-encoder is very variable depending on choices about its structure.

1.3 Current Preprocessing Methods

The application of preprocessing techniques to fNIRS data has been shown to improve classification accuracy and reduce the rate of false positives or negatives [1]. Denoising techniques generally seek to remedy the signal quality issues arising from systemic physiological signals and motion artefacts [2]. The most common techniques used in the removal of physiological noise are frequency filters, wavelet filters, individual component analysis (ICA) and general linear models (GLMs) [3, 6]. As of currently, there has not yet been a decided standard technique from these, with each method having its respective situational advantages and drawbacks [1, 6].

For instance, the most used technique of bandpass filtering, a form of frequency filtering where the energy of all signals outside of the HRF's typical frequency band are suppressed, is the simplest to implement and effective. However, it cannot remove confounding physiological noise that overlaps in frequency with the HRF such as Mayer waves. Wavelet denoising performs similarly, while ICA assumes a statistical independence between non-gaussian components, and can often overcorrect and remove target frequencies as well. When additional physiological sensors, such as accelerometers or BP monitors, or short separation channels are available, GLMs generally outperform other methods [5], with many variations, many of which address the short comings of the previous three mentioned methods. However, their performance is heavily dependent on the amount of additional physiological information available and also assumes that the HRF belongs to a family of pre-defined HRF functions, an assumption which doesn't necessarily hold in studies on significant fields of investigation, such neonatal or infant brain activity, or of those with cerebrovascular diseases [1, 6].

2 Method

2.1 Data Acquisition and Generation

The dataset used was taken from an open access dataset, containing task-free haemodynamic activity in 4-month old infants during sleep. There are 104 sets of raw measurements in the dataset, with no preprocessing having been performed on the data. Further details are described in the related paper [4].

As the ground-truth in our study, we use simulated clean HRF data, generated by the difference of two gamma functions [1, 13]. We obtain the parameters for this by fitting a linear regression model to the canonical HRF [13], and in order to prevent the model from overfitting, we vary the parameters for its generation slightly, as well as varying their magnitude. For each set of generated clean HRF data, we also included a blank reading, to allow the model to learn to de-noise samples without prior knowledge of activation period. We additionally applied horizontal shifts in activation period to promote generalisation.

In terms of the resting state data used in training our models, we used three different sets of data: One set of completely simulated data [13], a set of real experimental resting state data , and a set of augmented experimental data. The simulated data was generated using an adapted method based on that of J. Gemignani et Al. [15], by simulating each of the majorly contributing individual physiological signals in resting state fNIRS data according to data taken from literature, over random baseline white noise that was temporally correlated using a order 30 Autoregressive model [15]. The physiological signals we considered were cardiac activity, respiration, Mayer waves, and very low frequency oscillations [6, 15]. Motion artefacts were generated using spikes based on the exponential distribution. We augmented the real dataset by generating similar synthetic data samples using an Auto-Regressive model of order 6 (appendix 1.3). The Auto-Regressive model was fit to random sub-sets of recordings from several different patients, and used to predict multiple "tapes" of measurements, which were then sampled and mixed with the real data at a ratio of 1:1.

The data was segregated into a 10 to 1 split, in which for each 10 samples used in training and validation, 1 was removed and stored for testing and evaluation. We additionally adopted a leave-one-out approach in order to be able to test the generalisation of the models, leaving out the measurements from a participant for testing.

2.2 Data Processing

To ensure the validity of our resting state data generation and experimental results, we discard all short-separation channels (which we took as any channel shorter than 10mm [16]) which can only detect extra-cerebral tissue haemodynamics and so cannot contain an HRF. Additionally, we truncated all recordings to the same length of 10 minutes. With the remaining data, we converted the raw data to optical density measurements, then applied Beer-Lambert's Law to obtain a set of haemoglobin measurements.

We used two separate representations of the data in training the model. The first is as a set of time series data, which we applied standardisation to on a per-sample basis. Secondly, we used a set of time frequency representations of the signals, obtained using a Discrete Wavelet Transform with a 4th order Symlet wavelet as the mother wavelet. We standardised the wavelet transformed data on a per-sample basis, standardising each frequency band independently, which we found to be an improvement over the approach of standardising data

then converting it to the time frequency domain, and then additionally applied a set of frequency band weights that prioritised bands that coincided with known HRF frequencies [15].

2.3 Models

In order to fairly assess the potential of the Auto-Encoder architecture, we considered a wide range of different architectures for both the encoder and decoder modules of our models, based on on particular relevant characteristics identified from literature. We tested 7 different models in total, and as such we will discuss the general structure of each model, but the particular implementation will not be described. It can instead be found in the accompanying code. When applicable, we used a LeakyReLU activation function and employed dropout layers with probability varying between 0.1 and 0.4.

The two basic architectures that we used were the linear neural network for its simplicity and the convolutional neural network for its ability to capture spatial relationships and local patterns within the data. We employed a simple Linear model with 5 linear layers in both its encoder and decoder as a baseline with which to compare the other models against. For the convolution neural network approach, we considered two models with different structures. The first (basic) used 3 1D convolutional layers in the encoder and 4 in the decoder, with 64 channels in the feature space. The second (benchmark) follows the structure laid out in Yuanyuan G. et al's similar study [2], with 4 encoder layers and 5 decoder layers and 32 channels.

We also considered the combination of Auto-Encoders and Recurrent Neural Networks (RNNs), which have been commonly used in time series forecasting and denoising for their strong ability to capture temporal relationships [17]. Particularly, we used 'Long Short Term Memory' (LSTM) networks. We considered two approaches based on this architecture, the first using an LSTM encoder but a Linear decoder, and the second using both an LSTM based encoder and decoder. For the first, we employed 2 LSTM layers and a linear fully connected layer in the encoder and 4 linear layers in the decoder, while in the second we used 4 LSTM layers in the encoder and 4 LSTM layers with a linear fully connected layer in the decoder.

The two final models we considered were firstly an ensemble Auto-Encoder, that used convolution and LSTM based Auto-Encoders to extract both temporal and spatial features and reconstruct two different representations of the signal, then used a stacking approach, using a linear layer to combine their predictions. We used the basic CNN and LSTM-Linear models for these respectively (see appendix 1.7). Secondly, we consider a 2D convolutional model that uses an image of the wavelet transformed representation of the time series. We used 4 2D Convolutional layers in the encoder, and 6 in the decoder, and chose to have 32 intermediary channels.

In summary, we consider a Linear model, two 1D Convolutional models with different structures, an LSTM-encoder Linear-decoder model, an LSTM encoder-decoder model, a stacked-ensemble model with convolutional and LSTM models, and a 2D Convolutional model that uses an image of wavelet coefficients as input.

2.4 Training Process

We trained each model over 100 epochs, with an initial learning rate of 1e-3 and a learning rate scheduler than halves it every 25 epochs. For each of the models, we trained three separate versions, each version being trained on a different dataset as described in the data generation section.

Dataset	SNR	CNR	No. Samples	Sample Length
Synthetic Training	-53.00	0.0467	10000	234
Experimental Training	14.099	2.4714	2000	234
Augmented Training	10.514	2.0835	4000	234
Experimental Testing	28.81	0.2433	1000	234

Table 1: Initial Conditions

2.5 Evaluatory Metrics

We evaluated the performance of our model against a set of unseen data from a participant excluded from the training data set. We did so to evaluate its ability to generalise, and its usability in a practical settings.

The measures that we used to evaluate signal quality before and after, and as such model performance, were signal-to-noise ratio (SNR), in decibels, and contrast-to-noise ratio (CNR), also in decibels. Additionally, we assessed prediction and classification accuracy by comparing our predictions to the ground-truth data using the root mean squared error (RMSE), mean absolute error (MAE) and classification accuracy metrics. [For more detail on individual metrics see appendix]

3 Results

Model	SNR	CNR	MAE	RMSE	Pos. CA	Neg. CA
Basic Linear	28.81	13.45	0.5399	0.6949	0.996	0.126
Basic CNN	24.94	0.610	0.6377	0.8339	1.0	0.018
Bench. CNN	28.85	0.059	0.5445	0.7516	1.0	0.058
LSTM-Linear	37.72	2.474	0.4576	0.5609	0.996	0.286
Deep LSTM	62.93	43.75	0.3273	0.4211	0.99	0.488
Stacked AE	57.18	20.67	0.3587	0.4630	0.96	0.46

Table 2: Training Using Synthetic Data

Model	SNR	CNR	MAE	RMSE	Pos. CA	Neg. CA
Basic Linear	60.65	3.052	0.1493	0.1947	0.928	0.928
Basic CNN	51.28	2.121	0.2097	0.2867	0.9744	0.718
Bench. CNN	54.30	0.097	0.1809	0.2460	0.97	0.784
LSTM-Linear	54.14	7.103	0.1959	0.2459	0.95	0.754
Deep LSTM	75.52	12.71	0.1402	0.1833	0.932	0.874
Stacked AE	66.73	4.63	0.1407	0.1828	0.934	0.926

Table 3: Training Using Real Experimental Data

Model	SNR	CNR	MAE	RMSE	Pos. CA	Neg. CA
Basic Linear	64.46	3.552	0.5399	0.6949	0.936	0.918
Basic CNN	49.53	3.634	0.2056	0.2904	0.974	0.718
Bench. CNN	57.69	2.605	0.1620	0.2290	0.98	0.644
LSTM-Linear	55.52	9.56	0.1853	0.2351	0.96	0.792
Deep LSTM	78.44	11.29	0.1249	0.1623	0.954	0.864
Stacked AE	67.47	5.231	0.1323	0.1711	0.934	0.918
Wavelet CNN	58.37	3.872	0.2312	0.2074	0.98	0.83

Table 4: Training Using Augmented Experimental Data

We found that under equal and constrained conditions, the best performing model varied depending on target metric. The greatest signal-to-noise reduction was that of a deep-LSTM auto-encoder trained on AR-enhanced experimental data, achieving a 49.63 dB improvement, from a base 28.81dB to a post-processing 78.44. The greatest contrast to noise enhancement also was achieved by the Deep-LSTM with a 43.50 dB CNR improvement, but instead when trained with synthetic data alone. In terms of classification accuracy, the stacked Auto-encoder out-performed the Deep-LSTM, reaching a 93% classification accuracy when trained with un-augmented experimental data, and still achieving a 37.92 dB SNR increase. In our initial findings, the models with the worst performance were those in the class of models that used CNN based architectures, both due to their significantly lower accuracy in classifying null signals.

We achieved high improvement in all signal quality metrics using all datasets, with the highest performance generally observed in models trained using augmented experimental data, but we still observed good performance from LSTM models trained entirely on synthetic data.

4 Discussion

These results provide a reasonable proof of concept for the use of Auto-Encoders in the physiological denoising of fNIRS signals. The primary advantages offered by these over current methods is the limited number of assumptions made about the structure, distribution and composition of the input data, and consequently a wider set of potential use cases. Our method provides a superior denoising performance compared to currently used assumption free methods, as well as providing

Our method achieves a high degree of improvement in signal quality metrics without a requirement for short-separation channel and additional sensors. Additionally, due to its reasonable degree of classification

accuracy even as a proof of concept, when combined with a sliding window approach, it can be used in real time applications. With the rising development of real-time fNIRS systems for practical usage, our method shows potential for increasing signal quality in lighter and smaller sensor arrays. [see appendix]

On the other hand, in its current state, there are still many limitations in our methods, and while the majority are addressable with further improvements, some are intrinsic to the method. Using an auto-encoder to preprocess data is more computationally expensive than employing a bandpass filtering approach, and still requires prior training. Another flaw in our method (as a proof of concept) is that our tests and training was predicated on a set of conditions that are unrealistic in a true practical setting, limiting their application in wide-scale usage. These assumptions, such as the synthetic HRFs being from a variation of canonical HRF [see appendix], are not necessary for the method but were made for convenience of testing and to ensure our findings were generally valid.

We hope that this study can serve as a foundation for further study of deep-learning based fNIRS denoising techniques, and as a short guide for others aiming to apply these techniques in a practical setting. All of the code is available at the link below, both for the reproduction of the results of this study and for general use, with additional functionality not described in this study available.

https://github.com/KostasDemiris/Auto-fNIRS-toolkit

5 References

[1] Eastmond C, Subedi A, De S, Intes X. Deep learning in fNIRS: a review. Neurophotonics. 2022 Oct;9(4):041411. doi: 10.1117/1.NPh.9.4.041411. Epub 2022 Jul 20. PMID: 35874933; PMCID: PMC9301871.

[2] Gao Y, Chao H, Cavuoto L, Yan P, Kruger U, Norfleet JE, Makled BA, Schwaitzberg S, De S, Intes X. Deep learning-based motion artifact removal in functional near-infrared spectroscopy. Neurophotonics. 2022 Oct;9(4):041406. doi: 10.1117/1.NPh.9.4.041406. Epub 2022 Apr 23. PMID: 35475257; PMCID: PMC9034734.

[3] Zhang F, Cheong D, Khan AF, Chen Y, Ding L, Yuan H. Correcting physiological noise in whole-head functional near-infrared spectroscopy. J Neurosci Methods. 2021 Aug 1;360:109262. doi: 10.1016/j.jneumeth.2021.109262. Epub 2021 Jun 17. PMID: 34146592.

[4] Blanco B, Molnar M, Carreiras M, Caballero-Gaudes C. Open access dataset of task-free hemodynamic activity in 4-month-old infants during sleep using fNIRS. Sci Data. 2022 Mar 25;9(1):102. doi: 10.1038/s41597-022-01210-y. PMID: 35338168; PMCID: PMC8956728.

[5] Pinti P, Scholkmann F, Hamilton A, Burgess P, Tachtsidis I. Current Status and Issues Regarding Preprocessing of fNIRS Neuroimaging Data: An Investigation of Diverse Signal Filtering Methods Within a General Linear Model Framework. Front Hum Neurosci. 2019 Jan 11;12:505. doi: 10.3389/fnhum.2018.00505. PMID: 30687038; PMCID: PMC6336925.

[6] von Lhmann A, Ortega-Martinez A, Boas DA, Ycel MA. Using the General Linear Model to Improve Performance in fNIRS Single Trial Analysis and Classification: A Perspective. Front Hum Neurosci. 2020 Feb 18;14:30. doi: 10.3389/fnhum.2020.00030. PMID: 32132909; PMCID: PMC7040364.

[7] von Lhmann A, Li X, Mller KR, Boas DA, Ycel MA. Improved physiological noise regression in fNIRS: A multimodal extension of the General Linear Model using temporally embedded Canonical Correlation Analysis. Neuroimage. 2020 Mar;208:116472. doi: 10.1016/j.neuroimage.2019.116472. Epub 2019 Dec 20. PMID: 31870944; PMCID: PMC7703677.

[8] Chen WL, Wagner J, Heugel N, Sugar J, Lee YW, Conant L, Malloy M, Heffernan J, Quirk B, Zinos A, Beardsley SA, Prost R, Whelan HT. Functional Near-Infrared Spectroscopy and Its Clinical Application in the Field of Neuroscience: Advances and Future Directions. Front Neurosci. 2020 Jul 9;14:724. doi: 10.3389/fnins.2020.00724. PMID: 32742257; PMCID: PMC7364176.

[13] J. Gemignani and J. Gervain, "A practical guide for synthetic fNIRS data generation" 2021 43rd Annual International Conference of the IEEE Engineering in Medicine Biology Society (EMBC), Mexico, 2021, pp. 828-831, doi: 10.1109/EMBC46164.2021.9631014.

[14] Barker JW, Aarabi A, Huppert TJ. Autoregressive model based algorithm for correcting motion and serially correlated errors in fNIRS. Biomed Opt Express. 2013 Jul 17;4(8):1366-79. doi: 10.1364/BOE.4.001366. PMID: 24009999; PMCID: PMC3756568.

[15] Yoo SH, Huang G, Hong KS. Physiological Noise Filtering in Functional Near-Infrared Spectroscopy Signals Using Wavelet Transform and Long-Short Term Memory Networks. Bioengineering (Basel). 2023 Jun 4;10(6):685. doi: 10.3390/bioengineering10060685. PMID: 37370616; PMCID: PMC10295283.

[16] Gagnon L, Cooper RJ, Ycel MA, Perdue KL, Greve DN, Boas DA. Short separation channel location impacts the performance of short channel regression in NIRS. Neuroimage. 2012 Feb 1;59(3):2518-28. doi: 10.1016/j.neuroimage.2011.08.095. Epub 2011 Sep 8. PMID: 21945793; PMCID: PMC3254723.

[17] Dudek, G., Smyl, S., Peka, P. (2023). Recurrent Neural Networks for Forecasting Time Series with Multiple Seasonality: A Comparative Study. In: Valenzuela, O., Rojas, F., Herrera, L.J., Pomares, H., Rojas, I. (eds) Theory and Applications of Time Series Analysis. ITISE 2022. Contributions to Statistics. Springer, Cham. https://doi.org/10.1007/978-3-031-40209-$8_1$2

6 Appendix

6.1 Choice of Order in Autoregressive Data Generation

We chose an order of 6 as it was the last statistically significant value according to the results of a partial Auto-Correlation Function (pACF) test on the experimental data, with an arbitrary significance threshold of 0.2. The pACF measures the correlation between a value and its k'th time lagged term, after adjusting for the intermediary lagged terms, and we want to take the larges possible order such that the model includes all relevant lagged terms that have a direct effect on the model's prediction of the subsequent value. The 4'th order also had a significant result.

6.2 Reasoning for choice of Sample Size

Although it was possible to use more samples in training the models, our reasoning was in constraining the size of the training set was that it more closely simulates the conditions of a real fNIRS experiment with more limited readings. In addition, due to very limited computing power, using very large datasets resulted in excessively long training times.

6.3 Potential Reason for higher Stacked Auto-encoder accuracy

This is likely due to the deep LSTM's large number of layers and the limited size of the dataset. It is likely that it somewhat overfitted on the data and did not generalise as well to unseen data. Further increasing the auto-encoders depth beyond 4 layers in the encoder and decoder respectively decreased the performance of the Deep LSTM. The results could potentially change for this if increased regularisation was applied, and it was trained on a larger dataset.

6.4 Stacked Auto-Encoder design

In my initial design for the stacked auto-encoder, I added other models for consideration, such as the Linear AE, but found that this decreased performance, either because overfitting started to occur, or because the other models performed worse and so introduced poor quality features. I also investigated the usage of a stacked approach for each of the individual encoders and decoder, so using multiple models to encode the data, aggregating them in a code, then applying multiple decoder models to them and using a stacked approach to aggregate their predictions. Similarly, I found that this reduced performance, but I did not investigate in too much depth and as such this remains a potential direction of improvement.

6.5 More detail on selected Haemo-Dynamic Responce Function

As described in the report, we generated Synthetic HRFs using the difference of two gamma functions followed by some additional variations, but this choice was just to show that the results hold for the standard HRF. We can ignore this assumption by using larger datasets that include non-canonical HRFs, multiple overlayed HRFs and any random signals.

6.6 More detail on evaluatory metrics

In order to calculate signal-to-noise ratio, we used the formula:

$$SNR = 10 \log_{10} \left(\frac{P_{signal}}{P_{noise}} \right)$$

where P_{signal} is the power of the signal, and P_{noise} is the power of the noise.

We calculated contrast-to-noise ratio using the formula:

$$CNR = \frac{\mu_{target} - \mu_{noise}}{\sigma_{noise}}$$

Where μ_{target} is the target HRF's mean, μ_{noise} is the mean of the noise, and σ_{target} is the standard deviation of the noise.

We used these two metrics as they are commonly used in signal quality analysis in fNIRS and fMRI. The other three (RMSE, MAE and CA) are standard statistical metrics. The particular implementation of classification accuracy I used was flawed for the sake of convenience, as to determine whether a prediction was correct, I thresholded the prediction then compared it to the input signal which I had tagged as either null or actual. This was done for the sake of time as it was not reasonable to visually inspect every test set, and as such the exact values are likely not very precise.