Table of Contents

[A) General motivation for starting to a DNN project: 2](#_Toc130919643)

[1. Recurrent Neural Networks (RNNs): 2](#_Toc130919644)

[2. Convolutional Neural Networks (CNNs) for Time Series: 2](#_Toc130919645)

[3. Temporal Convolutional Networks (TCNs): 2](#_Toc130919646)

[4. Encoder-Decoder Architectures: 2](#_Toc130919647)

[5. Attention Mechanisms (Transformers): 2](#_Toc130919648)

[B) What we had done and what we might try based on this: 3](#_Toc130919649)

[a) EEG data analysis: 3](#_Toc130919650)

[1. Convolutional Neural Networks (CNNs) for Time Series: 3](#_Toc130919651)

[2. Recurrent Neural Networks (RNNs) with LSTM or GRU layers: 4](#_Toc130919652)

[3. Hybrid CNN-RNN architectures: Combining 4](#_Toc130919653)

[b) Behavioral data analysis: 4](#_Toc130919654)

[1. Feedforward Neural Networks (FNNs): 4](#_Toc130919655)

[2. Recurrent Neural Network (RNN) with LSTM or GRU layers 4](#_Toc130919656)

[C) Advantageous of having multiple models for our project: 5](#_Toc130919657)

[D) Useful sources: 6](#_Toc130919658)

[a) Papers: 6](#_Toc130919659)

[b) Codes: 7](#_Toc130919660)

# **General motivation for starting to a DNN project:**

We have been using psychophysics and traditional machine learning to model the behavioural and EEG data. However, these methods are not always optimal to analyse and capture complex data and patterns. In that sense, DNN models might be more powerful in analysing more complex human data.

**Keeping in mind that we have a time series data (EEG), and serial dependence is a dependency to the past events, I would recommend considering the following deep learning approaches:**

1. **Recurrent Neural Networks (RNNs):** RNNs are specifically designed to handle time series data by maintaining an internal state that can represent information from previous time steps. Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) are two popular types of RNNs that can efficiently capture long-range temporal dependencies, making them well-suited for modeling serial dependence in our data.
2. **Convolutional Neural Networks (CNNs) for Time Series:** Although traditionally used for image processing, CNNs can also be applied to time series data by using 1D convolutional layers. These layers can learn to detect local patterns and dependencies in the data, and when combined with pooling layers, they can capture information at multiple timescales.
3. **Temporal Convolutional Networks (TCNs):** TCNs are an extension of CNNs for time series data that employ causal convolutions to ensure the model only uses information from the past. They are designed to capture long-range dependencies and have shown strong performance on a variety of sequence-to-sequence tasks.
4. **Encoder-Decoder Architectures:** These models consist of two main components, an encoder that processes the input time series data and a decoder that generates the output based on the encoded information. This architecture can be combined with RNNs, LSTMs, or GRUs to model the serial dependence in our data effectively.
5. **Attention Mechanisms (Transformers):** Attention mechanisms allow models to selectively focus on different parts of the input sequence when generating output. Integrating attention mechanisms into RNNs or encoder-decoder architectures can help our model better capture the influence of past events on the current perception and decision.

Additionally, we might consider applying different preprocessing or down sampling our data (based on the research question), such as filtering, normalization, or channel removal, to ensure optimal performance of our model. Once we have selected a model, we can train it using our behavioral and EEG data, fine-tune the model's hyperparameters, and validate its performance using cross-validation. We can then use the trained model to analyse our data, make predictions, and hopefully draw conclusions about the nature of serial dependence.

# **What we had done and what we might try based on this:**

In one of our EEG projects, we conducted an experiment where the participants reported the orientation of two different “spatiotemporally overlapping” features (RDK and Gabor) of an object in a sequence of trials. Participants were randomly pre-cued to report the orientation of one feature (either the dot motion or the Gabor tilt) in each trial. Behavioural data showed that participants' decisions about the currently targeted features are biased towards the orientations of the previously targeted features (reported 1 trial ago) independently of the feature type (similar effects from Gabor to RDK and from RDK to Gabor). Additionally, there was no bias to the non-reported feature in the past. This showed us that the attention, decision making and task relevance are crucial in serial dependence since it only occurred for the attended task relevant features that participants made decisions about in the past, and not for the non-target features, although they were simultaneously presented at the very same location.

During the experiment, we recorded the brain signals via EEG. The primary aim was to decode the orientations of the target features in each trial, and to see if the decoded orientations are shifted to the orientations of the previously reported features (i.e., serial dependence). Also, in case of an existence of a shift, we wanted to see WHEN that shift occurred. This is important to understand the origin of the serial dependence. A shift of the decoded orientations obtained by the early feedforward brain signals (right after the presentation of a stimulus) might be more evidence to the perceptual origin of serial dependence. On the other hand, a shift occurring in the late decoded orientations, at the time period where the high level information processing mechanisms (such as memory, decision making, predictions) are involved, might suggest more a post-perceptual origin of serial dependence (visit the talk on a very similar research here: https://www.youtube.com/watch?v=L4LdpnAIxkQ&list=PLd11O9rJl7-luO7m9nWa4ospjSxVHGCVi&index=2&t=10s).

Unfortunately, with the traditional methods so far, we could not decode the stimulus features in the EEG data. Therefore, more complex DNN models might be better alternative to achieve this initial plan of that project. NOTE that the poor decoding performances that we previously obtained might be (and very likely) due to the complexity of the stimulus and experiment design. A simpler design including sequential presentation of each feature (as we do now in the current experiment) instead of simultaneous presentation of two overlapping would potentially lead to better decoding performances.

**Based on what we have so far, I would recommend the following deep learning architectures for decoding the stimulus features from EEG data and modeling the behavioral data:**

## **EEG data analysis:**

**1. Convolutional Neural Networks (CNNs) for Time Series:** CNNs with 1D convolutional layers can capture local patterns and dependencies in the time series EEG data. We can experiment with different layer depths and kernel sizes to optimize the model's ability to decode the orientation of target features.

**2. Recurrent Neural Networks (RNNs) with LSTM or GRU layers:** These models are designed to handle time series data, and they can capture long-range temporal dependencies in the EEG signals. We can train an RNN to predict the orientation of the target feature based on the EEG data, and then examine the decoded orientations to determine if they are biased towards previously reported features.

**3. Hybrid CNN-RNN architectures: Combining** CNN and RNN layers in a single architecture can enable our model to capture both local patterns in the EEG data (using CNN layers) and long-range dependencies (using RNN layers). This hybrid approach may help improve the decoding performance.

The hybrid architecture here seems to me more promising considering our data and the research question. Nonetheless, we can develop the models one after the other to ease the work, optimizing the parameters for each and testing their performance step by step.

## **Behavioral data analysis:**

1. **Feedforward Neural Networks (FNNs):** FNNs are a simple and sometimes an effective choice alone for modeling behavioral data in our field. We can experiment with different numbers of hidden layers and neurons to find the optimal network architecture for our dataset.
2. **Recurrent Neural Network (RNN) with LSTM or GRU layers:** Again, these models can capture long-range temporal dependencies and are suitable for modeling serial dependence if it involves recurrent processing. We can train an RNN with LSTM or GRU layers to predict the behavioral responses based on the past trials' information.

Though, we should keep in mind that deep learning models might require a large amount of data to achieve optimal performance. If our dataset is relatively small (which might be), we might consider using **data augmentation techniques** or **transfer learning** through pre-trained models in the literature.

# **Advantageous of having multiple models for our project:**

Based on our research to date and the findings from the literature, serial dependence seems to require a recurrent information processing. Although, there exist studies claiming that serial dependence is a low level bias and originates perceptual via early feedforward signals, there are many other studies showing the involvement of and modulation by the post-perceptual mechanisms that feeds back the perception.

Considering these, we can build a simple FNN and another more complex RNN model, and use Bayesian model comparison techniques to quantitatively compare the performance of these models in explaining the behavioral data.

If the recurrent model outperforms the feedforward model, this might suggest that serial dependence is likely to involve post-perceptual mechanisms and recurrent processing. On the other hand, if the feedforward model performs as well as the recurrent model, this might indicate that serial dependence can be explained by simpler, low-level processes without the need for recurrent mechanisms.

After comparing the performance of the feedforward and recurrent models on the behavioral data, we can focus on the model that better explains the data and adapt it to work with our EEG data. For example, if the recurrent model is found to be more effective, we can integrate the RNN with LSTM or GRU layers into a model that takes both behavioral and EEG data as inputs to predict the target variable. In this case, we would likely benefit from the use of convolutional layers to process the spatial information in the EEG signals before feeding them into the RNN.

# **Useful sources:**

Unfortunately, there is not an extensive literature on using deep learning for serial dependence, particularly with EEG or fMRI data. However, we can refer to some studies that have used deep learning in the context of Neuroscience.

## **Papers:**

1. Schirrmeister, R. T., Springenberg, J. T., Fiederer, L. D. J., Glasstetter, M., Eggensperger, K., Tangermann, M., & Ball, T. (2017). Deep learning with convolutional neural networks for EEG decoding and visualization. Human Brain Mapping, 38(11), 5391-5420.

<https://onlinelibrary.wiley.com/doi/full/10.1002/hbm.23730>

**•** *This paper demonstrates the use of convolutional neural networks for EEG decoding.*

1. Bashivan, P., Rish, I., Yeasin, M., & Codella, N. (2015). Learning representations from EEG with deep recurrent-convolutional neural networks. arXiv preprint arXiv:1511.06448.

<https://arxiv.org/pdf/1511.06448.pdf>

**•** *This paper combines convolutional and recurrent layers for EEG analysis.*

1. Quiroga, M. D. M., Morris, A. P., & Krekelberg, B. (2019). Short-term attractive tilt aftereffects predicted by a recurrent network model of primary visual cortex. Frontiers in Systems Neuroscience, 13, 67.

<https://www.frontiersin.org/articles/10.3389/fnsys.2019.00067/full>

**•** *In this paper, the model successfully predicts the attractive tilt aftereffects. While the model is a type of neural network, it is not a generic deep neural network (DNN). Instead, it is a custom, biologically inspired recurrent network model specifically designed to study the primary visual cortex's processing and adaptation mechanisms.*

1. Walker, E. Y., Cotton, R. J., Ma, W. J., & Tolias, A. S. (2020). A neural basis of probabilistic computation in visual cortex. Nature Neuroscience, 23(1), 122-129.

<https://www.nature.com/articles/s41593-019-0554-5>

**•** *In this paper, a DNN model was used to decode the uncertainty in the form of likelihood functions from the V1 population response.*

## **Codes:**

**For the coding part, I highly recommend Neuromatch Academy resources:**

1. Main webpage: <https://academy.neuromatch.io/>
2. **Computational Neuroscience course** from previous years: <https://compneuro.neuromatch.io/tutorials/intro.html>
3. **Deep learning course** from previous years: <https://deeplearning.neuromatch.io/tutorials/intro.html>

Building a DNN model is a long process requiring too much hyperparameter search and constant tunning. Therefore, I suggest that we start with an existing model to accelerate the whole process. Neuromatch provides general codes and preprocessed data that we can highly benefit to start with a new project.

**Here are the additional sources from GitHub:**

1. Deep Learning with PyTorch - A 60-minute blitz to get started with PyTorch: <https://pytorch.org/tutorials/beginner/deep_learning_60min_blitz.html>
2. Deep Learning for Time Series with PyTorch - A GitHub repository with examples for time series forecasting using PyTorch: <https://github.com/hyliush/deep-time-series>

These resources should provide a good starting point for our project. The best way is to start with simple models and gradually adjust their complexity based on the performance. I prefer using **Python** and **PyTorch** on **Google Colab**. Having a **GitHub** account is a must. For EEG data processing and analyse we can use **MNE,** Python package for neurophysiological data processing, and we can refer to the tutorials here <https://mne.tools/stable/index.html>.