Project 2: Machine Learning for Science Translational Neural Engineering Lab

Dupont-Roc Maud, Grosjean Barbara, Ingster Abigaïl
CS-433 Machine learning, EPFL

***Abstract*—The aim of this project is to decode an electroencephalogram (EEG) recorded in subjects who are presented with a visual stimulus on a screen: a five-by-five pixel image made of black and gray squares. To this end, we used machine learning models. The first method involved implementing a pixel-wise approach, i.e. training a Support Vector Machine (SVM) model on each pixel for a binary classification task, thereafter grouping each model to form the whole stimulus. The second approach consisted of using a U-shaped fully convolutional neural network (UNet) in order to capture the spatial dependence between each recording channel (electrode). Although our primary objective of overfitting the data was achieved, modifying the training algorithm was essential to improve the generalisability of the model, which was ultimately not satisfactory enough.**

I. Introduction

One approach to restore vision in cases of acquired blindness consists of using electrical stimulation (ES) to evoke visual sensations in the form of phosphenes. Neural correlates of elicited perceptions could be used for automatic and real-time ES parameter optimization. In the scope of our project, feedback cortical signals from electroencephalography (EEG) were used as neural correlates. Machine learning methods offer a solution for creating closed-loop systems that can instantly modify stimulation settings according to ongoing neural activity. Such models are trained on existing data of brain activity to predict the effect of electrical stimulation. Yet, feeding recorded signals into conventional ML models requires deriving meaningful features from these inputs, which typically involves working with broader and more basic stimuli (e.g. left vs. right). In turn, this reduced the usability of existing data, generated by presenting 60 different complex stimuli to healthy subjects. In order to enlarge the range of predictable stimuli, we intend to explore a pixel-wise approach as a first try, using several SVM models performing a simple binary classification task (no stimulus: black vs. stimulus: gray), although losing the spatial correlation of the output. SVM uses the maximization of the margin as a learning objective. SVM is a linear model and therefore assumes the data to be linearly separable, unless embedding it into a peculiar (and nonlinear) feature space using the kernel trick. On another hand, the use of deep learning techniques for EEG decoding is until now limited by the scarcity of available data, which is in our case partly solved by the usability of *all* stimulus classes, not only simple ones. Accordingly, our aim is to investigate the use of a U-shaped fully convolutional (deep) neural network to perform a 1-vs-59 stimuli single-trial decoding task. A typical UNet, as introduced in [1] is composed of a contracting path responsible for extracting and encoding the high-level features of the input (EEG epoch), followed by an expansive path (decoder) responsible for refining the features found in the encoder to reconstruct the spatio-temporal information and increase the resolution. Such a refinement is notably allowed by the skip connections bridging the information extracted from the encoder together with reconstructed information from the decoder. The architecture we used is depicted in Fig. 1. UNets have initially been introduced in the field of medical image segmentation. Yet, EEG signals reveal spatiotemporal relationships that can be considered analogous to the spatial dependencies exhibited by images.

II. Models and Methods

A. Data Set and Data Processing

The data originates from a study involving 10 subjects positioned behind a 75-inch screen where a specific pattern of gray pixels is shown for 0.75 seconds while recording an electroencephalogram from 128 electrodes. Each subject was presented with a random display of the 60 different stimuli 50 times.

This study focuses on the detection of sensory phenomena in EEG. As these measures have a great inter-subject variability, the model is built for a single subject. For the following, the data from the subject with the best recording quality was used, according to the lab. The original data set was presented as follows: EEG signals were constituted of recording epochs for one stimulus display. Channels with unrealistic values, most likely caused by measurement failure, were deleted and all the artifacts were removed. The lab performed baseline subtraction and applied a low-pass filter at 70Hz.

The data was split into test and train sets with a test ratio of 0.2. A subset of the train set was used as a validation set for the optuna hyperparameters tuning with a validation-ratio of 0.3. As the data set is small due to the challenges involved in obtaining EEG recordings, a relatively low ratio of the data is used for the testing, and the validation set is included in the training set. The signals were cropped into 0.75 seconds epochs, to match the stimulus display. As the signals provided were sampled at 256 Hz, the EEG recordings were thus converted to 192 time steps x 128 channel matrices. They were standardized across each channel, by subtracting the mean over the channel and dividing it by the standard deviation. The labels were converted to 5x5 tensor array of [0, 1] labels. The data set was balanced in terms of stimuli (each of the 60 stimuli presented 50 times).

B. Exploring SVM and UNet Models

1) Loss Function: The idea behind the pixel-wise SVM model is to follow a naive approach in which a SVM classifier is built for each pixel, incorporating its own parameters and hyperparameters. The 25 individual SVM classifiers are then concatenated into a larger model. We used the L_2-regularized hinge loss (multi-class margin loss), as is customary for SVM models and where the regularization parameter is a tunable hyperparameter. For training the UNet, we used a customized version of the traditional cross-entropy loss, which we called the weighted focal loss and is defined for each batch i as:

$$l_i = -\alpha_i (1 - p_i)^{\gamma} log(p_i), \tag{1}$$

where p_i represents the softmax-activated input. Firstly, α_i enables the compensation of class imbalance along each batch (fewer gray than black pixels: $N_g < N_b$) and is simply defined as:

$$\alpha_i = \begin{cases} 1 - f_{min} & \text{if } y = 1 \\ f_{min} & \text{if } y = 0 \end{cases}, \tag{2}$$

where $f_{min} = \frac{N_g}{N_g + N_b}$. Secondly, the modulation factor $(1 - p_i)^{\gamma}$ aims at improving the model's performance on difficult examples, i.e. on examples for which the model lacks confidence (small p_i). The focusing parameter γ is a hyperparameter to be tuned.

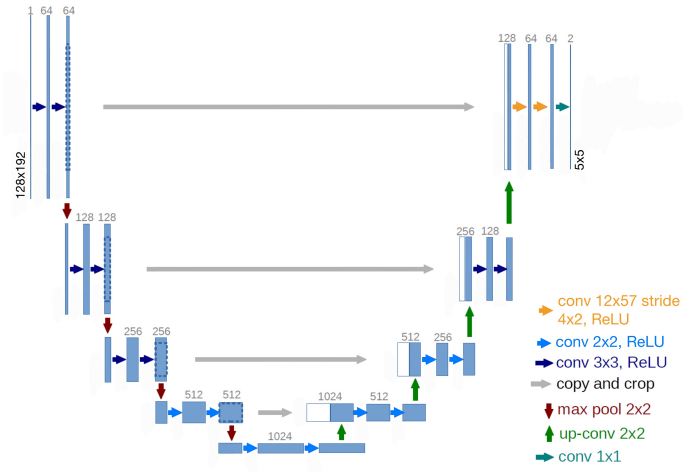

Fig. 1. UNet's architecture. *copy and crop* refers to skip connections, while *up-conv* refers to transpose convolutions. Figure edited from [1].

2) Training Algorithm: As for most machine learning models, we performed the training through gradient descent steps. The choice of the optimization algorithm is crucial to guarantee an efficient training process. This implies considering an appropriate optimizer as well as a learning rate scheduler. For the SVM model, Adam was chosen over classical gradient descent methods such as stochastic gradient descent (SGD), for its enhanced convergence speed and numerical stability, avoiding exploding gradients. It was also preferred over AdamW, as the loss being used (2) already incorporates a regularization term. For both the UNet and SVM, we selected two potentially interesting learning rate schedulers: stepLR and CosineAnnealingLR, which we tuned afterwards with optuna. On another hand, given that improper parameter initialization can lead to vanishing or exploding gradients, we operated a variance-preserving initialization of weights (He initialization, for ReLU networks) to control the layerwise variance of activations.

An important challenge arose from the UNet's constrained output shape: a 5-by-5 pixel image. We overcame this constraint by employing diophantine equations to determine suitable kernel sizes and strides. For the sake of simplicity, we decided to only adapt the last layer of the UNet, at the expense of compromising its symmetry, as depicted in Fig. 1.

We finally introduced dropout layers before each downsampling layer in the encoder and upsampling layer in the decoder, arbitrarily selecting a dropout probability of 0.5, which is commonly used as a starting value. Indeed, introducing such stochastic behavior in the training process is thought to improve the generalization error.

C. Hyperparameter Tuning

TABLE I
UNET'S HYPERPARAMETERS TUNED WITH OPTUNA, F1 SCORE AS OBJECTIVE FUNCTION

Optimizer	AdamW
Weight Decay λ	0.00026
Learning Rate	0.00004
β_1	0.952
β_2	0 .999
Loss	Weighted focal loss
Focusing parameter γ	0.188
Scheduler	CosineAnnealingLR
η_{min}	0.00000026

TABLE II
PIXEL $n°3$ SVM'S HYPERPARAMETERS TUNED WITH OPTUNA, BALANCED ACCURACY AS OBJECTIVE FUNCTION

Optimizer	AdamW
Weight Decay λ	0.0000281
Learning Rate	0.00004
β_1	0.9828
β_2	0.9120
Loss	MultiMarginLoss
Regularisation term λ	0.0000169
Scheduler type	StepLR
γ	0.0001676
step size	9.55679

All the hyperparameters mentioned above were tuned with optuna [2], a model hyperparametrization framework, automating and optimizing hyperparameter search, going beyond basic grid search. Such tuning was performed on a separate validation set, which was then merged back and shuffled with the training set. Indeed, even though this might have added some overfitting of the validation set (to a very small extent), the relatively few data points we have compared to the number of trainable parameters led us to keep as many samples as possible for the training. For both SVM and UNet, considering

that less than 10 hyperparameters were to be tuned and that we chose relatively broad search intervals for each, we judged our search space as moderately sized and therefore ran 20 trials of 10 epochs each. In the case of the UNet, due to using a small number of training epochs on limited data to simulate the model's training phase, we considered the hard accuracy as an overly stringent criterion that would systematically cause early trial pruning, and thus defined the objective function to be maximized as the F1 score. For the SVM model, we used a balanced version of the accuracy which accounts for both the model's sensitivity and specificity, although it could have been interchanged with the F1 score. The hyperparameters yielded with optuna validation for the UNet are summarized in Table I. The pixel-wise SVM hyperparameters are tuned separately. An example of the hyparameters obtained for the SVM model is summarized in II.

D. Use of EPFL clusters

The training and validation phases were performed on SCITAS clusters. Indeed, the computer resources given by our local CPU and the free GPU on Google Colab were far from sufficient to train and validate our models, especially the UNet (1 hour of training for 2 epochs on Google Colab). This part has been a key to achieve the training of our model, but it was also an important time-consuming issue as the usage was not trivial.

III. Results and Discussion

Our data set being unbalanced towards black pixels, a key aspect was to select appropriate metrics to monitor during the training and testing phases. We used the F1 score as well as the balanced accuracy implemented with the scikit-learn library [3].

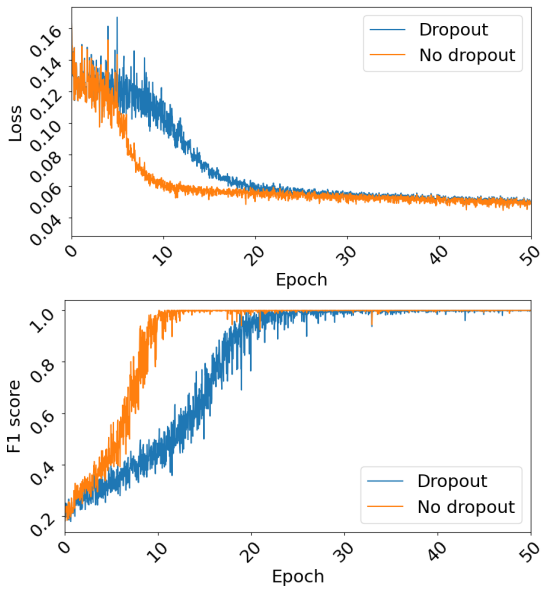

Fig. 2. Training set overfitting. Weighted focal loss (top) and F1 score (bottom) throughout the training of the UNet.

Since the model is overparametrized, after training the UNet with 200 epochs, we yielded a perfect overfit of the training set (Fig. 2) and voluntarily restarted a training session, terminating it after 18 epochs with dropout layers and 8 epochs without. We opted for this revised training epoch count based on the achieved F1 score during the complete training, choosing a moderate value to mitigate further overfitting. The F1 score slightly improved after this correction. With hindsight, we realized that a more efficient and standardized approach would involve monitoring test metrics at each training epoch in order to visualize when the model starts to overfit and to stop the training accordingly. We also faced overfitting with the SVM model, with accuracies reaching 100% (see Fig. 3) for most of the pixels.

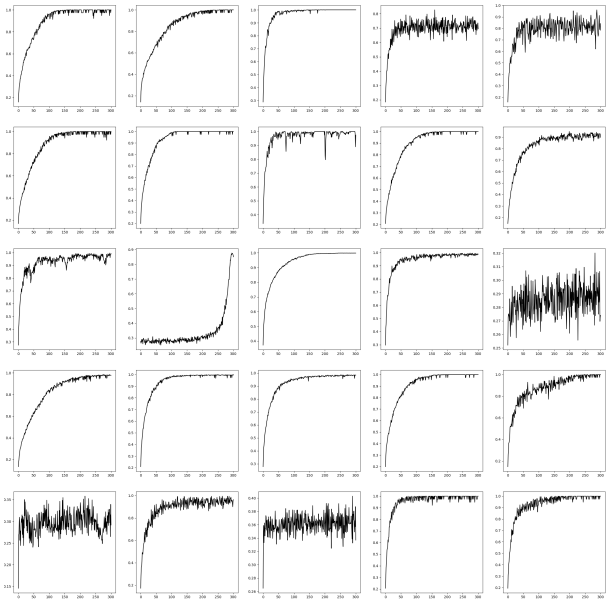

Fig. 3. Pixel-wise balanced accuracy throughout the training of the SVM model, indicating overfitting in most of the cases.

Because the initial predicted stimulus might not match an actual stimulus, we've substituted it with the existing stimulus that shows the highest linear correlation to our initial prediction. For the UNet predictions, the best-predicted stimuli tend to be punctual (one square), while the worst-predicted stimuli initially cover a larger part of the display screen Fig. 4. Therefore, our CNN model seems partially capable of recognizing local features in the input data, but hardly captures more global patterns. The neurons comprising the network likely possess small receptive fields. Yet, we did not incorporate this matching strategy during the training phase, so the F1 score depicted in Fig. 2 is computed from the raw prediction. This did not affect the training, as the function under optimization is the loss.

Such a concept of receptive field cannot be applied to the SVM model, which by definition loses the spatial correlation between the input and the output.

The outcomes of the testing phase are presented in Table III in terms of hard accuracy and F1 score, after matching

TABLE III
TESTING OUTCOMES FOR THE SVM AND UNET (ABORTED TRAINING) MODELS

	SVM	UNet	Dropout UNet
Mean hard accuracy	0.0484	0.0251	0.0301
Mean F1 score	0.1942	0.2140	0.2156

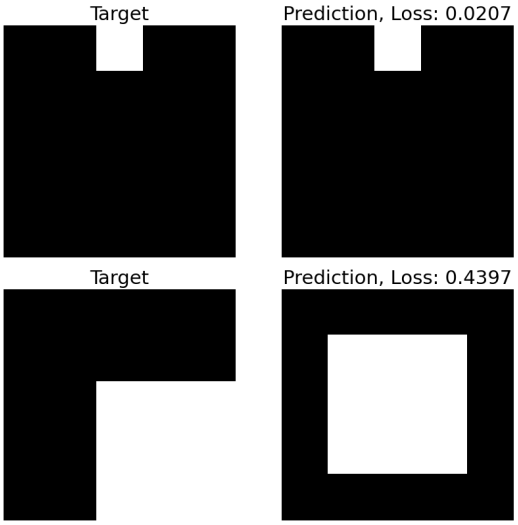

Fig. 4. UNet's test predictions under dropout, depicting both the lowest and highest losses after matching the UNet's prediction with an existing stimulus on the basis of linear correlation.

the raw prediction to an existing stimulus. The mean accuracy is a hard measure of similarity between prediction and target stimuli (either 0 or 1), whereas the F1 score operates a pixel-wise comparison which, while relaxing the criterion of perfect similarity, considers class imbalance to fairly evaluate the quality of the prediction. Although the mean accuracy lies above chance for both the UNet and SVM models, i.e. 1/60, such low values unequivocally signify overfitting. One can also notice that the pixel-wise balanced accuracy depicted in Fig. 5 remains in the range of the chance level for all pixels. Employing early stopping of the training phase and tuning the previously defined set of hyperparameters proves insufficient in addressing this issue for both models. The UNet model could improve by restoring its characteristic symmetry, achieved through recalculating new kernel and stride sizes and fine-tuning its depth (adjusting the number of layers in the contracting and expansive paths).

As anticipated, yet with a modest effect, introducing dropout layers positively affected the test predictions.

IV. ETHICS

The recorded EEG signals are confidential and highly sensitive. Although the data are recorded from healthy subjects, the diagnosis of a pathological condition cannot be excluded. The subjects were informed and signed a consent form, and the study was performed under the approval of an ethical committee. We cared to keep the data private by exclusively working with local copies. Inter-subject variability is a crucial aspect to thoroughly consider if one intends to apply the developed models on a different population, and even on a different subject. Indeed, the studied sample (10 subjects) is homogeneous regarding some features, such as age and education level which could correlate with specific biological conditions.

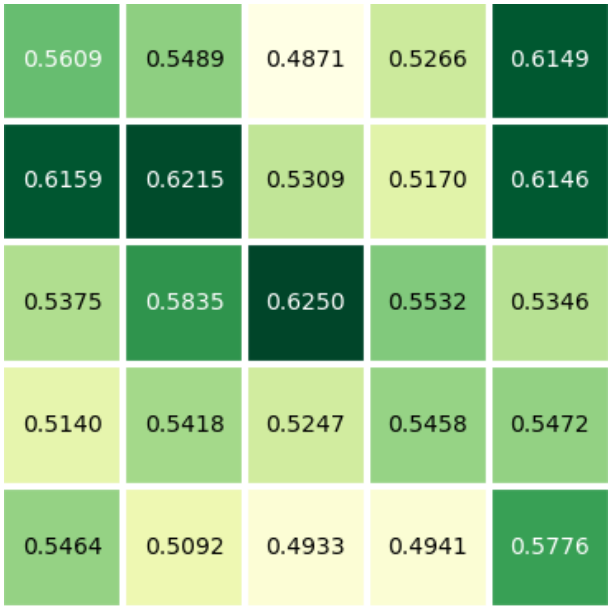

Fig. 5. Pixel-wise testing balanced accuracy of the SVM model.

V. CONCLUSION

In conclusion, although the UNet's architecture offers a promising framework to decode spatio-temporally correlated EEG signals, our model strongly lacks robustness in predicting the presented visual stimulus. In particular, it can barely compete with the pixel-wise SVM model baseline. Data augmentation as well as UNet's architecture fine-tuning present a potential solution to mitigate this issue. An interesting perspective to examine is the similarity between our model's architecture with brain structures involved in the visual processing pathway.

References

1. Ronneberger, O., P.Fischer & Brox, T. *U-Net: Convolutional Networks for Biomedical Image Segmentation* in *Medical Image Computing and Computer-Assisted Intervention (MICCAI)* **9351.** (available on arXiv:1505.04597 [cs.CV]) (Springer, 2015), 234–241. http://lmb.informatik.uni-freiburg.de/Publications/2015/RFB15a.
2. Akiba, T., Sano, S., Yanase, T., Ohta, T. & Koyama, M. *Optuna: A Next-generation Hyperparameter Optimization Framework* in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2019).
3. Pedregosa, F. *et al.* Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* **12,** 2825–2830 (2011).
4. Romeni, S., Toni, L., Artoni, F. & Micera, S. Decoding EEG correlates of visual stimuli compatible with electrical stimulation (not published).
5. Harris, C. R. *et al.* Array programming with NumPy. *Nature* **585,** 357–362. https://doi.org/10.1038/s41586-020-2649-2 (Sept. 2020).
6. Paszke, A. *et al.* in *Advances in Neural Information Processing Systems 32* 8024–8035 (Curran Associates, Inc., 2019). http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.
7. Gramfort, A. *et al.* MEG and EEG Data Analysis with MNE-Python. *Frontiers in Neuroscience* **7,** 1–13 (2013).