

UNet-HCRF Integration for EEG Pattern Recognition

Xiaohang Ma¹, Shiyong Xiao², Xiaohui Yin²

¹Department of Mathematics, University of Connecticut

²Department of Statistics, University of Connecticut

Storrs, CT

December 9, 2024

Overview

Introduction

Methods

Experiment

Results and Conclusion

Overview

Introduction

Methods

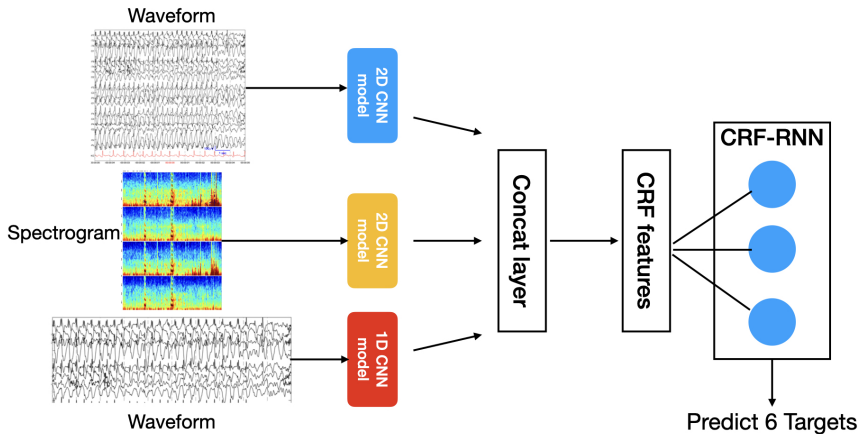
Experiment

Results and Conclusion

Introduction

- Electroencephalography (EEG) pattern recognition tasks are crucial for healthcare, where accurate and timely data classification can significantly impact patient outcomes.
- Deep learning has recently emerged as a powerful tool for sequence labeling.
- Conditional random fields (CRFs) have been employed for modeling label dependencies, demonstrating promising results.

Framework



Overview

Introduction

Methods

Experiment

Results and Conclusion

CNN-HCRF Model Structure

- Given the feature of the input data \mathbf{I} learned by CNN, i.e. $\mathbf{x} = \mathbf{F}(\mathbf{I})$, its corresponding hidden part states \mathbf{h} , and the class label y , a hidden conditional random field (HCRF) has the exponential form:

$$\mathbb{P}(y, \mathbf{h} | \mathbf{x}; \theta) = \frac{\exp(\Phi(y, \mathbf{h}, \mathbf{x}; \theta))}{\sum_{y' \in Y} \sum_{\mathbf{h} \in H^N} \exp(\Phi(y', \mathbf{h}, \mathbf{x}; \theta))},$$

where θ is the model parameter, H^N denotes the set of all possible hidden part states of N hidden parts, and $\Phi(y, \mathbf{h}, \mathbf{x}; \theta)$ refers to potential function depending on the feature \mathbf{x} .

- The probability of class label y for the given feature \mathbf{x} is:

$$\mathbb{P}(y | \mathbf{x}; \theta) = \sum_{\mathbf{h} \in H^N} \mathbb{P}(y, \mathbf{h} | \mathbf{x}; \theta) = \frac{\sum_{\mathbf{h} \in H^N} \exp(\Phi(y, \mathbf{h}, \mathbf{x}; \theta))}{\sum_{y' \in Y} \sum_{\mathbf{h} \in H^N} \exp(\Phi(y', \mathbf{h}, \mathbf{x}; \theta))}.$$

CNN-HCRF Model Structure (cont.)

The jointly probability distribution of the HCRF model is:

$$\begin{aligned}\Phi(y, \mathbf{h}, \mathbf{x}; \theta) = & \underbrace{\sum_{j \in \nu} \phi(h_j, x_j; \omega) + \sum_{i \neq j} \psi(h_i, h_j, x_i, x_j; \eta)}_{\text{Measures log-likelihood } \log \mathbb{P}(\mathbf{h}|\mathbf{x}; \theta)} \\ & + \underbrace{\sum_{j \in \nu} \varphi(y, h_j, x_j; \delta) + \vartheta(y, x_0; \varpi)}_{\text{Measures log-likelihood } \log \mathbb{P}(y|\mathbf{h}, \mathbf{x}; \theta)},\end{aligned}$$

where:

- Unary potential $\phi(h_j, x_j; \omega)$: Measures the likelihood of the local feature x_j is assigned as the hidden part state h_j , with parameter ω is learned by end-to-end CNN-UNet structure.

CNN-HCRF Model Structure (cont..)

- Binary potential $\psi(h_i, h_j, x_i, x_j; \eta)$: Balances the hidden label compatibility of the neighboring pixels p on the feature \mathbf{x} :

$$\psi(h_i, h_j, x_i, x_j; \eta) = \mu(h_i, h_j) \left[\omega_1 \exp \left(-\frac{|p_i - p_j|^2}{2\eta_\alpha^2} - \frac{|x_i - x_j|^2}{2\eta_\beta^2} \right) + \omega_2 \exp \left(-\frac{|p_i - p_j|^2}{2\eta_\gamma^2} \right) \right],$$

where $\mu(h_i, h_j)$ is a label compatibility function, ω_1 and ω_2 are linear combination weights of different kernels, and the parameters η_α , η_β and η_γ control the influence of the corresponding feature spaces.

CNN-HCRF Model Structure (cont...)

- Unary potential $\varphi(y, h_j; \delta)$: Measures the compatibility between global class label y and the hidden local state h_j :

$$\varphi(y, h_j; \delta) = \sum_{a \in Y} \sum_{b \in H} \delta_{a,b} \cdot \mathbb{1}(y = a) \cdot \mathbb{1}(h_j = b),$$

where $\mathbb{1}(\cdot)$ is a indicator function, and $\delta_{a,b}$ denotes the likelihood of class label $y = a$ containing a joint with hidden state $h_j = b$, learned during training.

- Global potential $\vartheta(y, x_0; \varpi)$: Measures the likelihood of the global feature x_0 is assigned as the action label y , with parameter ϖ learned during training process.

Variational Inference of HCRF

- The mean-field varepsilon family approximation of $\log P(\mathbf{h} \mid y, \mathbf{x}, \theta)$ is expressed as $Q(h) = \prod_{i=1}^N q_i(h_i)$.
- The evidence lower bound (ELBO) under the mean-field family is:

$$\begin{aligned} \text{ELBO}(Q) &= \mathbb{E}_{Q(\mathbf{h})} \log \mathbb{P}(y, \mathbf{h} \mid \mathbf{x}; \theta) - \mathbb{E}_{Q(\mathbf{h})} \log q(\mathbf{h}) \\ &= \sum_{i=1}^N \mathbb{E}_{q_i(h_i)} [\phi(h_j, x_j; \omega) + \varphi(y, h_j, x_j; \delta)] \\ &\quad + \sum_{i=1}^N \sum_{j=1}^N \sum_{l, l'} \mathbb{1}_{\{i \neq j\}} Q(h_i = l) Q(h_j = l') \mu(l, l') \\ &\quad \left[\omega_1 \exp \left(-\frac{|p_i - p_j|^2}{2\eta_\alpha^2} - \frac{|\mathbf{F}_i(\mathbf{l}) - \mathbf{F}_j(\mathbf{l}')|^2}{2\eta_\beta^2} \right) + \omega_2 \exp \left(-\frac{|p_i - p_j|^2}{2\eta_\gamma^2} \right) \right] \\ &\quad - \sum_{i=1}^N \mathbb{E}_{q_i} \log q_i(h_i) + \vartheta(y, \mathbf{x}; \varpi). \end{aligned}$$

Variational Inference of HCRF (cont.)

- By the coordinate ascent variational inference (CAVI) algorithm, fixing $q_j(h_j), \forall j \neq i$, the optimal $q_i(h_i)$ that maximizes ELBO($q_i(h_i)$) is:

$$Q(h_i) \propto \exp\{\mathbb{E}_{Q(\mathbf{h}_{-i})} \log \mathbb{P}(y, h_i, \mathbf{h}_{-i} \mid \mathbf{x}; \theta)\}.$$

- The explicit form of $q_i(l)$ is derived as follows:

$$q_i(l) = \frac{1}{Z_i} \exp \left\{ \phi(h_j, x_j; \omega) + \varphi(y, h_j, x_j; \delta) + \sum_{j \neq i} \sum_{l'} q(l') \mu(l, l') \right. \\ \left. \left[\omega_1 \exp \left(-\frac{|p_i - p_j|^2}{2\eta_\alpha^2} - \frac{|\mathbf{F}_i(\mathbf{l}) - \mathbf{F}_j(\mathbf{l})|^2}{2\eta_\beta^2} \right) + \omega_2 \exp \left(-\frac{|p_i - p_j|^2}{2\eta_\gamma^2} \right) \right] \right\},$$

where Z_i is the normalization constant.

Mean Field Inference in HCRF

Algorithm 1: Mean-field inference in fully-connected CRF

Input: $Q_i(h_i) \leftarrow U_i(h_i)$, $i = 1, 2, \dots, N$ // Initialize $Q(h)$

1 **while** not reach max iteration number **do**

2 $\hat{Q}_i^{(m)}(h_i) \leftarrow \sum_{j \neq i} k^{(m)}(f_i, f_j) Q_j(h_j)$ for all m // Message
 Passing;

3 $\tilde{Q}_i^{(m)}(h_i) \leftarrow \sum_{l \in \mathcal{L}} \mu^{(m)}(h_i, l) \sum_m \omega^{(m)} \hat{Q}_i^{(m)}(l)$ // Compatibility
 Transform;

4 $Q_i(h_i) \leftarrow \exp\{\phi(h_j, x_j; \omega) + \varphi(y, h_j, x_j; \delta) - \tilde{Q}_i(x_i)\}$ // Local
 Update;

5 Normalize $Q_i(h_i)$;

6 **end**

Overview

Introduction

Methods

Experiment

Results and Conclusion

Harmful Brain Activity Classification Dataset

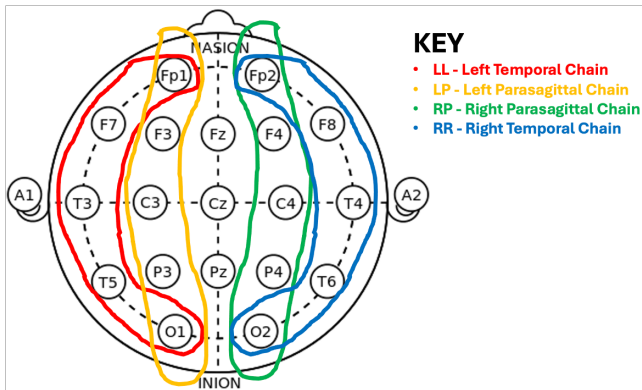
- The dataset contains 50-second long EEG samples covering a 10-minute window as well as spectrograms matching each sample.
- The EEG samples were labeled by expert annotators to determine which of the six features each spectrogram belonged to.
- The longer 10-minute window allows for the model to pick up on further insights regarding the nature of the brain activity.

Data Preparation

- Extract 10-minute long spectrograms centered at the midpoint time, where the expert ratings (Global EEGs) were provided.
- Convert the 50-second long EEG waveforms into spectrograms.

Make Spectrogram from EEG

- Bipolar Montage combines multiple EEG electrode signals by calculating the difference between adjacent electrodes, yielding signals representing specific brain regions.



Make Spectrogram from EEG (cont.)

- A single time series signal is required to generate a spectrogram. To create four spectrograms from the dataset's 19 EEG time series signals, we employ Bipolar Montage to combine them into four representative signals.

$$\text{LL Spec} = \frac{1}{4} [\text{spec}(Fp1 - F7) + \text{spec}(F7 - T3) + \text{spec}(T3 - T5) + \text{spec}(T5 - O1)],$$

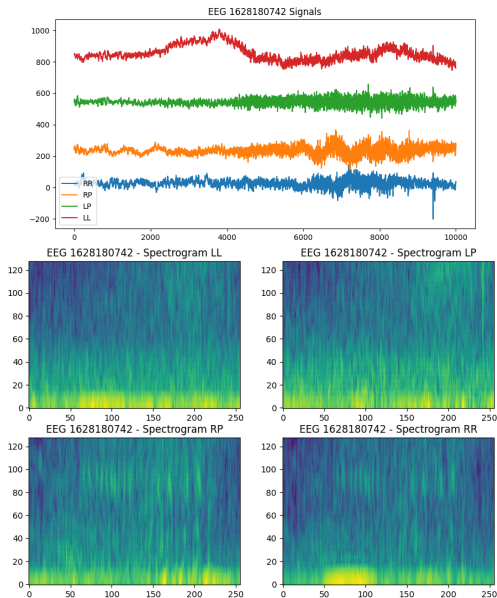
$$\text{LP Spec} = \frac{1}{4} [\text{spec}(Fp1 - F3) + \text{spec}(F3 - C3) + \text{spec}(C3 - P3) + \text{spec}(P3 - O1)],$$

$$\text{RP Spec} = \frac{1}{4} [\text{spec}(Fp2 - F4) + \text{spec}(F4 - C4) + \text{spec}(C4 - P4) + \text{spec}(P4 - O2)],$$

$$\text{RR Spec} = \frac{1}{4} [\text{spec}(Fp2 - F8) + \text{spec}(F8 - T4) + \text{spec}(T4 - T6) + \text{spec}(T6 - O2)],$$

where $\text{spec}(\cdot)$ is a custom function implemented using the `pywt` and `librosa` packages for signal transformation.

Make Spectrogram from EEG (cont..)



Training Schedule

The loss function was designed as Kullback-Leibler (KL) divergence.

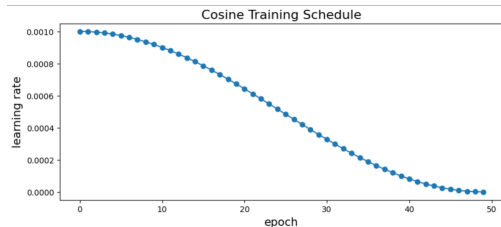


Figure 1: Pre-train UNet learning rate schedule.

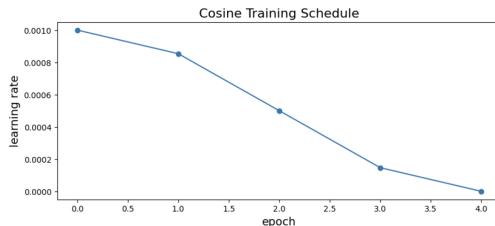


Figure 2: Fine tuning HCRF learning rate schedule.

Overview

Introduction

Methods

Experiment

Results and Conclusion

Results

We evaluate our model by 5 fold cross-validation.

- Our model achieves a **0.4992** KL-divergence.
- The baseline model from Kaggle using EfficientNet obtains a 0.59 cross-validation KL-divergence.

However, there still are many teams on Kaggle have achieved even lower KL-divergence scores (0.2-0.3) by leveraging the power of large models like Vision Transformers (ViT).

Conclusion

- We build a Hidden Conditional Random Fields for EEG based Spectrograms image classification.
- The latent CRF layer is desgined to capture the latent segmentation information of the output of Neural Networks.
- A variational mean-field inference scheme is used to address the posterior distribution.
- We integrate the Unet and HiddenCRF into an end-to-end training structure. The Neural Networks feature learning and variational inference can be accelerated jointly by GPUs.