04.30.25

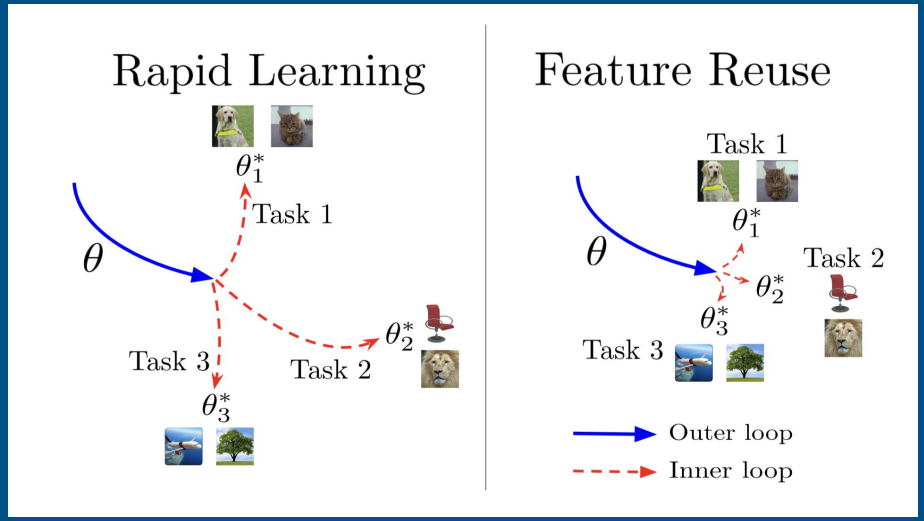
Model Agnostic Meta Learning for Neural Signal Prediction ("NeuraNIL")

Yishu Li
Jania Vandevoorde
Carlos Ramos

CSCI 1470: Deep Learning

Meta Learning

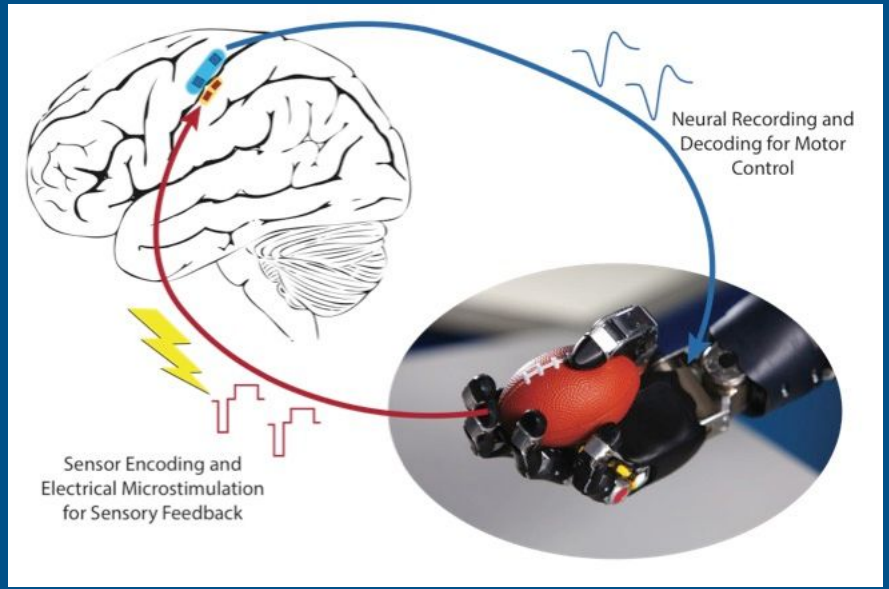
- AKA learning to learn
- Training models that can adapt to other tasks with limited training examples by using experience from previous tasks it's learned



[1] Raghu et al. (2020)

Intracortical brain-computer interfaces (iBCIs)

- Device that records neural signals from the brain's cortex
- Enable people with advanced conditions to communicate by understanding their movement intentions from their neural signals



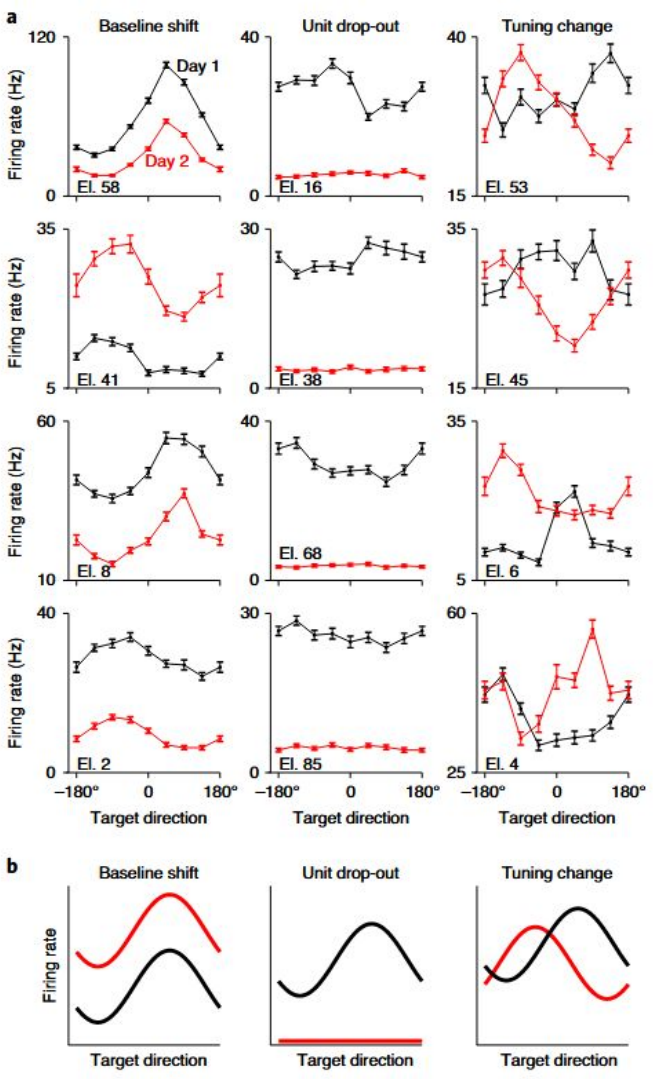
[2] https://www.rnel.pitt.edu/research/neuroprosthetics/intracortical-brain-computer-interfaces

Motivation

- Modern iBCI systems require frequent signal decoder re-calibration to maintain accuracy due to the non-stationary of neural signal

→ Can we train a model to help with this?

→ Each day is a new task!



[3] Degenhart, A. D. et al. Stabilization of a brain–computer interface via the alignment of low-dimensional spaces of neural activity. Nat Biomed Eng 4, 672–685 (2020).

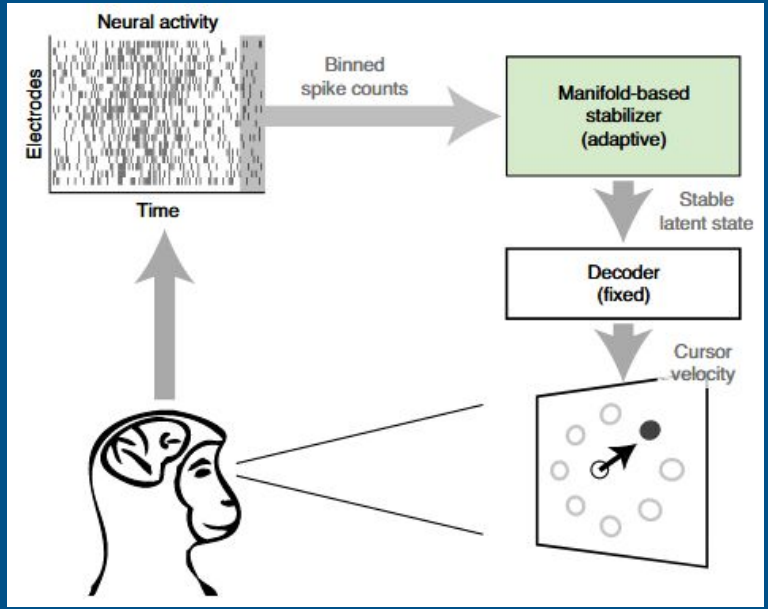
Relevant Related Works

- Finn et. al (2017): Model Agnostic Meta-Learning (MAML) learns a parameter initialization via an outer loop and then adapts all parameters in inner loops
- Raghu et al. (2019): Almost No Inner Loop (ANIL) simplifies MAML by updating only the final classifier layer in the inner loop which is faster with equal performance

	MAML	ANIL
Outer loop training	$\theta_i \leftarrow \theta_i - \beta \nabla_{\theta_i} \sum_{\tau_i \sim p(\tau)} L_{\tau_i}(f_{\theta_i'})$	$\theta_{i+1} = \theta_i - \alpha \nabla_{\theta_i} \sum_{\tau_i \sim p(\tau)} L_T(w_i'^T \phi_{\theta_i}(x), y)$ $w_{i+1} = w_{i+1} - \nabla_{w_i} \sum_{\tau_i \sim p(\tau)} L_T(w_i'^T \phi_{\theta_i}(x), y)$
Inner loop Adaptation	$\theta_i' = \theta_i - \alpha \nabla_{\theta_i} L_{\tau_i}(f_{\theta_i})$	$w_i' = w_i - \beta \nabla_{w_i} L_T(w_i'^T \phi_{\theta_i}(x), y)$
Model	Any Model	Learner + Task-specific heads

[1] Raghu et al. (2020) & [4] Finn et al. (2017)

- There are linear decoders like the Kalman filter that have been used for neural decoding
- Models trained on signals from previous days may suffer from performance degradation
- A proposed solution is to use manifold-based decoders that map neural data to a lower dimensional latent space.



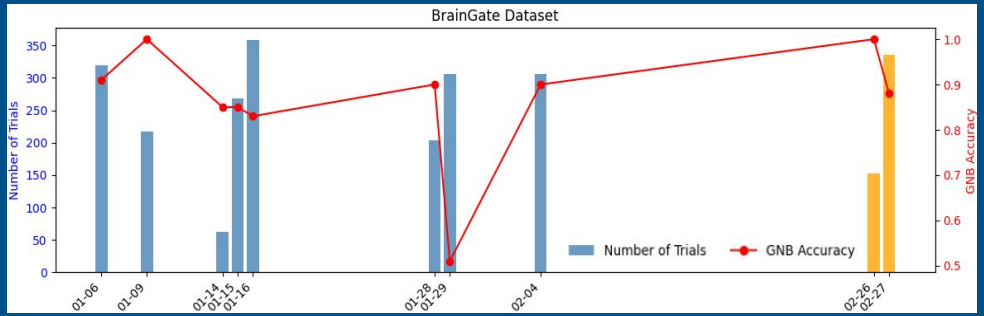
[3] Degenhart, A. D. et al. Stabilization of a brain–computer interface via the alignment of low-dimensional spaces of neural activity. Nat Biomed Eng 4, 672–685 (2020).

The Data

(1) BRAINGATE2 CLINICAL TRIAL DATA

Clinical trial Data (ongoing study): BrainGate2 participant with advanced ALS trying to communicate by imagining a set of gestures. The raw signal was recorded as 30000 Hz voltage data

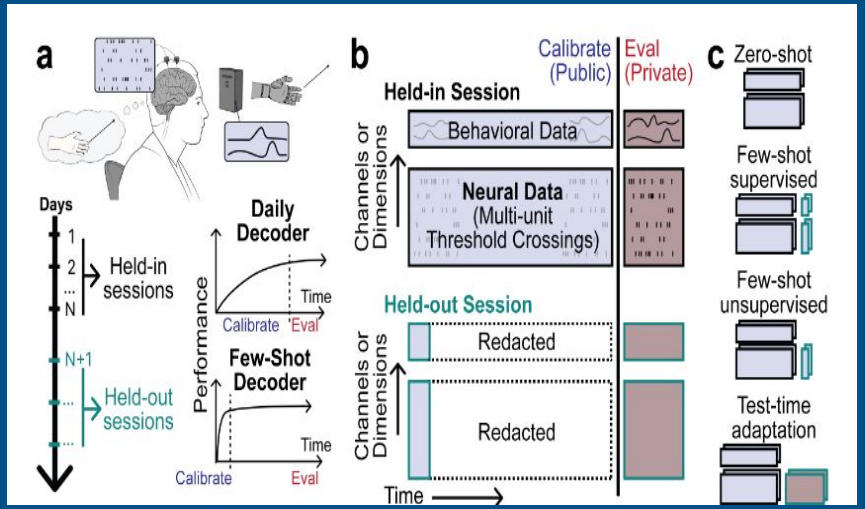
We preprocessed by re-referencing, filtering, threshold crossing, and power calculation will be calculated to get binned spiking rate and spiking power features.

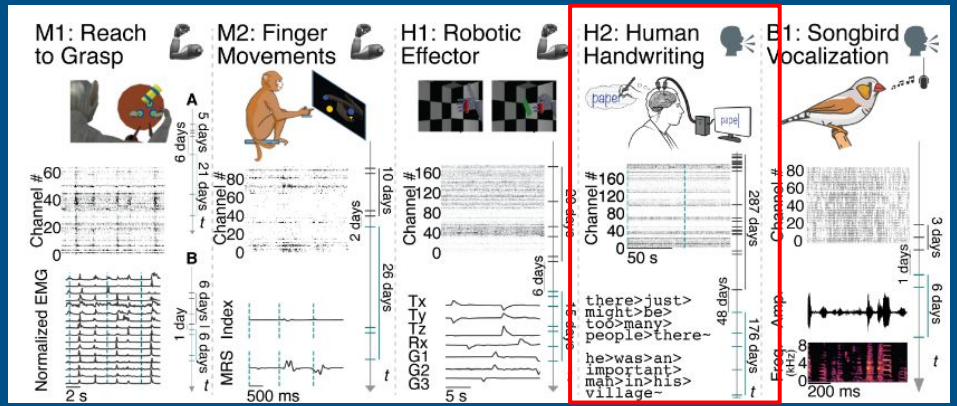


The Data

(2) FALCON H2 DATASET

- A iBCI user imagine writing letters
- 11 days, 26 classes





[5] Karpowicz et al. (2024)

Methodology

- We want to implement both a base model (standard DL model) and our NeuraNIL model (based on ANIL) to compare their accuracy
- We implement an outer loop (learner) which is a model that takes in the raw neural features and maps neural data into a lower dimensional latent space, and a inner loop that classifies the data in the latent space

The Base Models

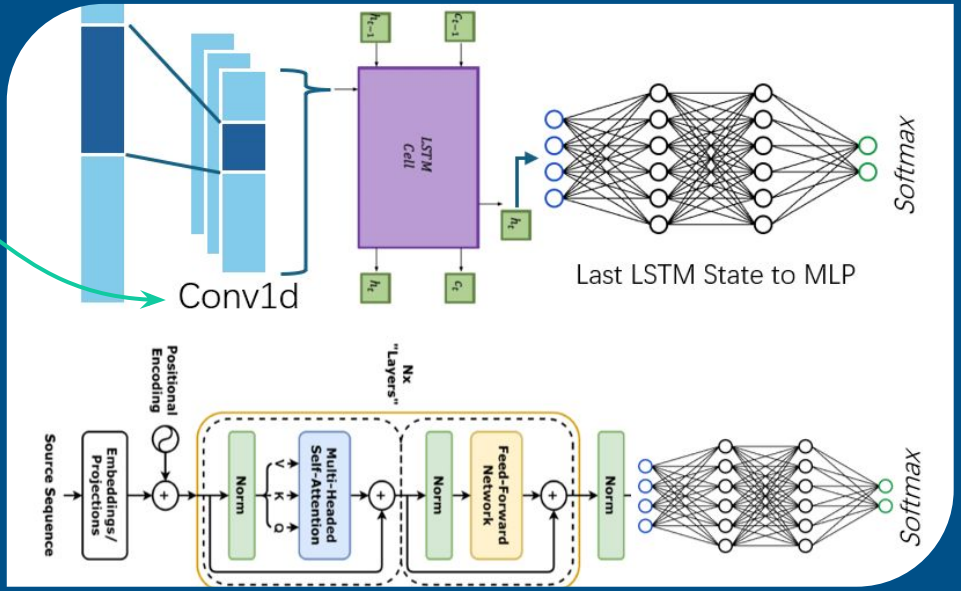
- A standard neural network that leverages n*ormalization* using `BatchNorm1d` on our time series neural signal data
- Since we're working in a model agnostic setting, we can select any kind of gradient descent model.

  We chose Gaussian Naive Bayes (GNB) as baseline, LDA for visualization, MLP, CRNN, and transformer encoder.

The Base Models

CRNN and Transformer Encoder

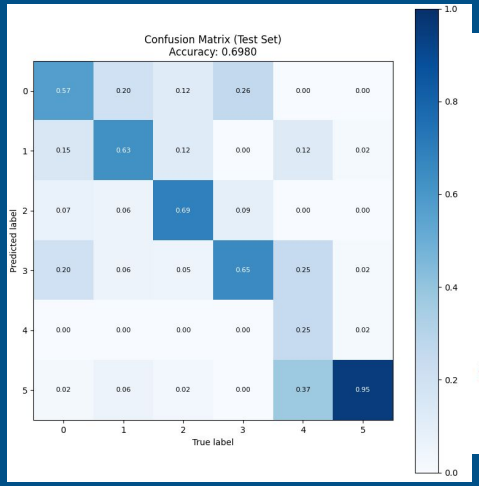
Down-sampling
Feature Extraction

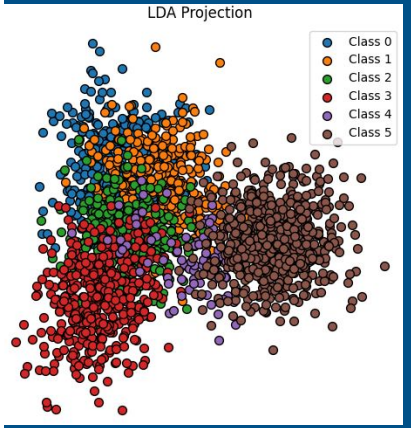


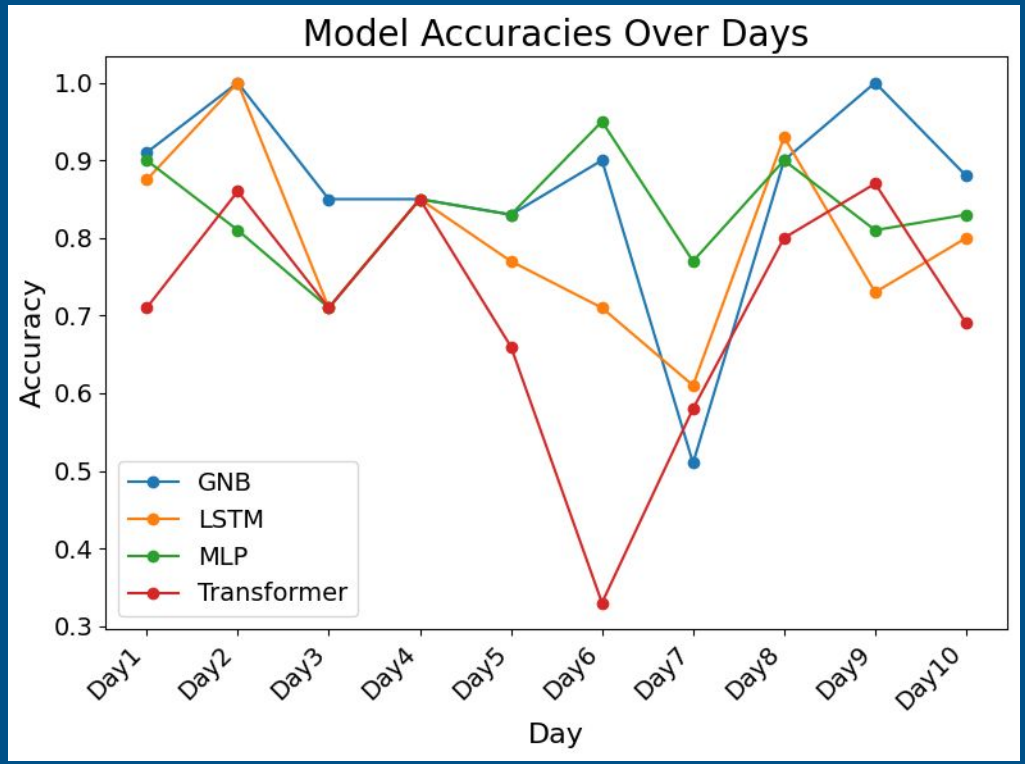
We only need the encoder model as feature extraction layer because we don't need the next word prediction

The Base Models

Decoder performance on BrainGate data

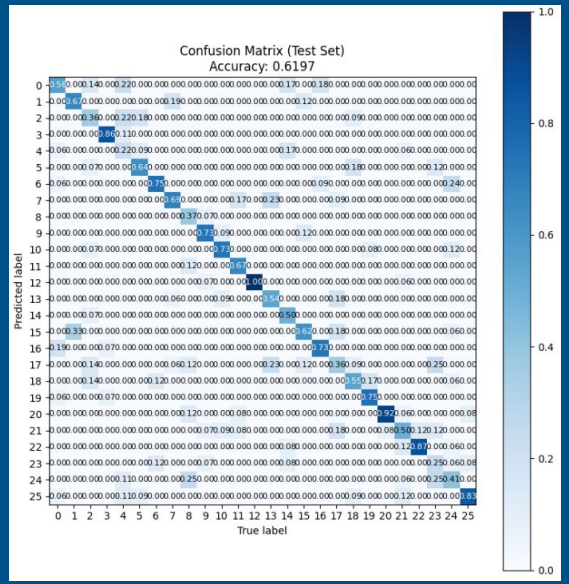


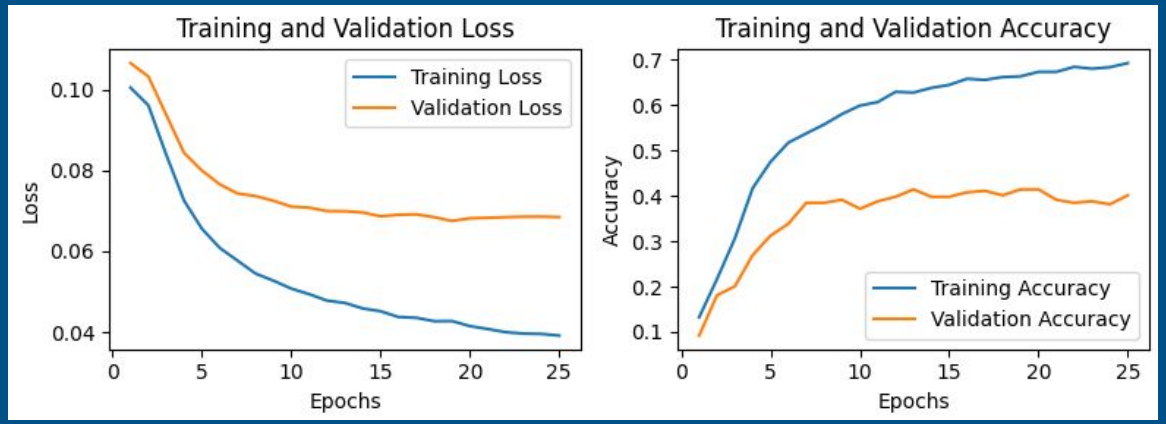




The Base Models

FALCON dataset and model overfitting (Normalizing may help sometimes)



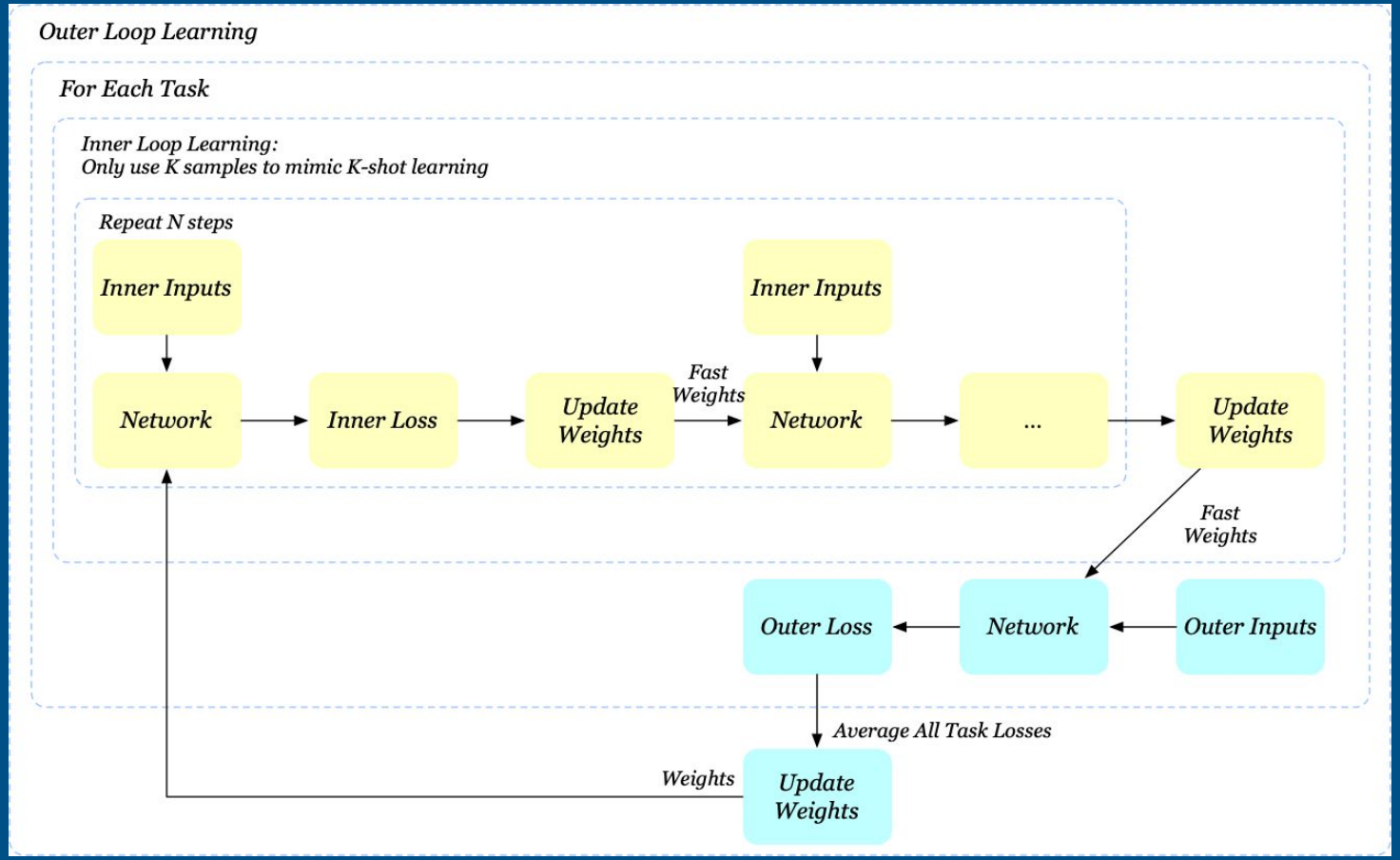


Our NeuraNIL Model

NeuraNIL can be broken down into two parts:

→ *Outer loop*: also called the *learner*. Updates the meta-initialization of the NN parameters to enable fast adaptation. Takes in the raw neural features and maps it to a lower dimensional latent space to avoid non-stationary. Will be an LSTM or transformer due to the data's sequential nature.

→ *Inner loop*: also called the *classifier*. Performs task-specific adaptation over a few labeled examples. Takes the latent space neural data and makes a prediction. Typically an MLP but any other model trained on gradient descent will work.



[6] https://pub.towardsai.net/how-to-train-maml-model-agnostic-meta-learning-90aa093f8e46

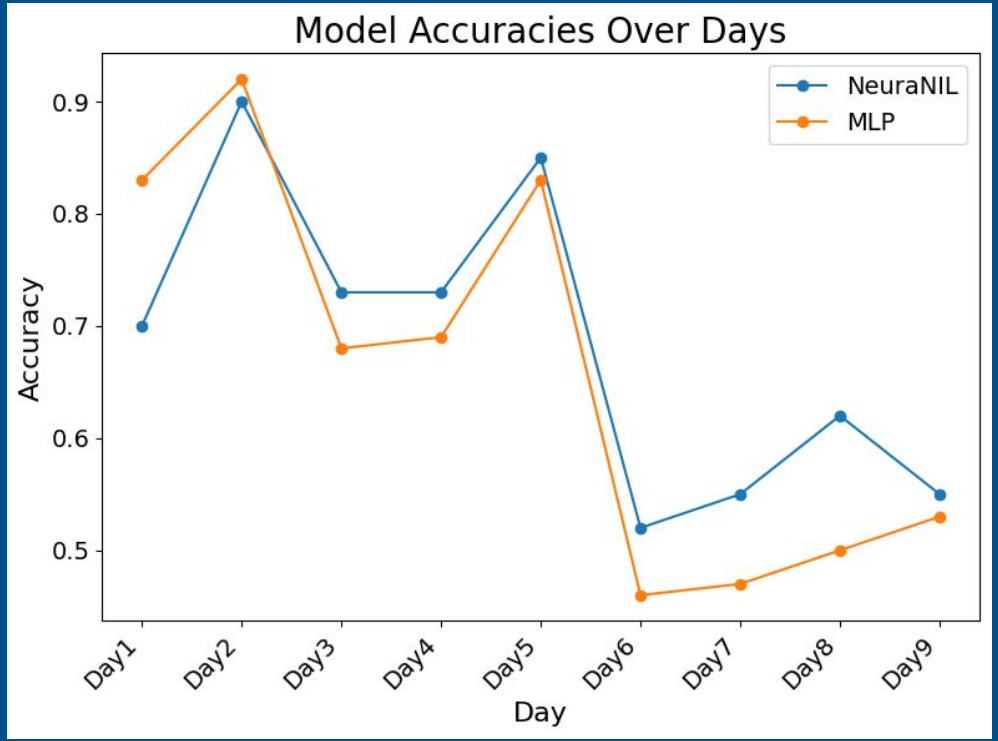
Algorithm 1 NeuraNIL training

Input: Training set divided to n episodes $\mathcal{D} = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$. Each (x_i, y_i) is an episode that can be divided into support set (x_{Si}, y_{Si}) and query set (x_{Qi}, y_{Qi})

Initialize learner parameter θ_0 and classifier w_0
outer loop → **while not done do**
  Sample *batch_size* episodes x_i, y_i from $\mathcal{D}$ as B_j
  $\mathcal{L}_j = 0$
inner loop → **for each** *episode* **do**
    $x_{Si}, y_{Si}, x_{Qi}, y_{Qi} \leftarrow x_i, y_i$
    Learner forward pass $x'_{Si} = \phi_{\theta_j}(x_{Si})$
    Classifier coefficient $w'_i = w_i - \beta \nabla_{w_i} L_T(w_i'^T \phi_{\theta_i}(x_{Si}), y_{Si})'$
    $\hat{y}_{Qi} = w_i'^T(\phi_{\theta_j}(x_{Qi}))$
    $\mathcal{L}_j += CE(y_{Qi}, \hat{y}_{Qi})$
  end for
  $\theta_{i+1} = \theta_i - \alpha \nabla_{\theta_i} \sum_{\tau_i \sim p(\tau)} L_T(w_i'^T \phi_{\theta_i}(x), y)$
  $w_{i+1} = w_{i+1} - \nabla_{w_i} \sum_{\tau_i \sim p(\tau)} L_T(w_i'^T \phi_{\theta_i}(x), y)$
end while

Next-day prediction

- BrainGate dataset
- We use $0 \sim (n\text{-}1)$ day's data to predict n day's data
- NeuraNIL performs slightly better than MLP

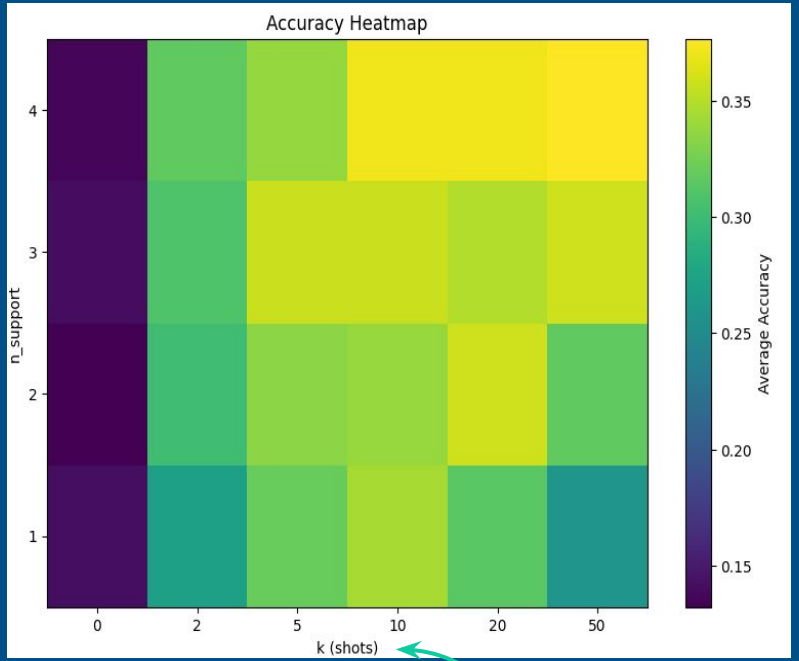


Our NeuraNIL Model

K–N sweep
- K: Inner loop epochs
- N: Support set

The trade-off: higher K means more generalized learner but slower adaptation to new tasks!

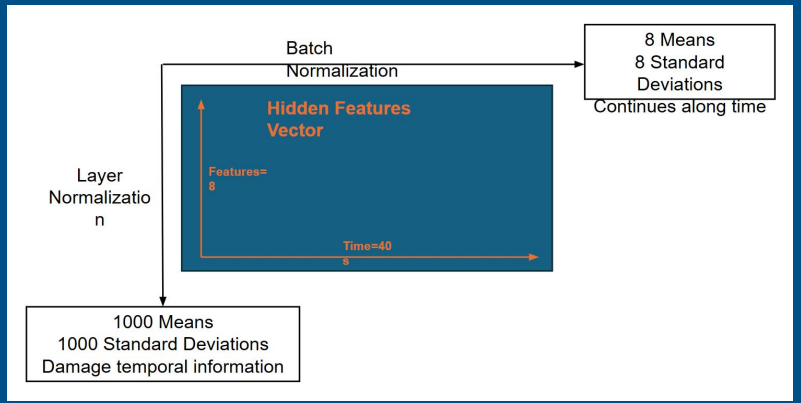
number of samples in the support set



number of inner loop iterations

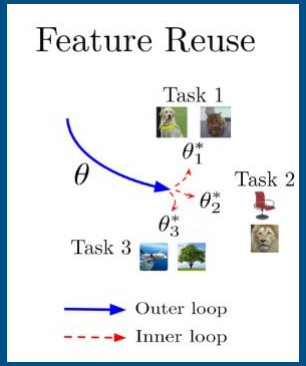
Discussion.....

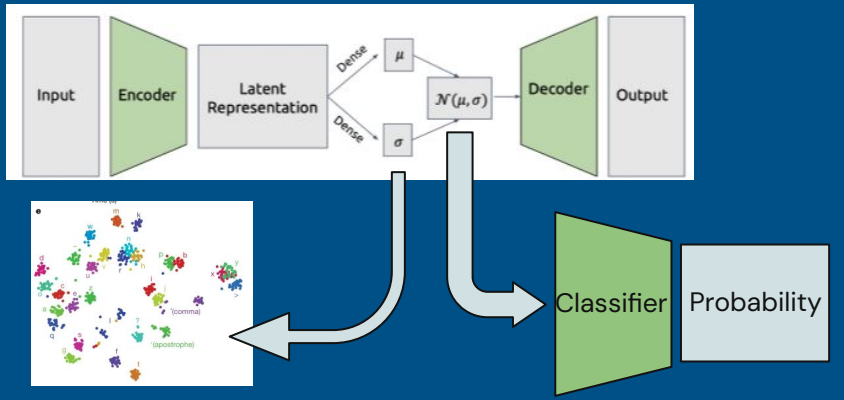
1. Given the nature of adapting to new tasks, NeuraNIL should be able to fast-adapt to new classes.
2. The DL models overfits easily, and preprocessing, especially normalization seems to help a lot. When the task is simple, traditional ML methods seems to perform better. → Think twice before you decide to use DL!
3. Batch or Layer norm for temporal data? Layer norm may damage the temporal information!



Discussion.....

4. What if auto-encoder?





5. Interpretability and Ethics for clinical trials
   We need to be really careful with patients!

References

[1] Raghu et al. (2020), *Rapid Learning or Feature Reuse? Towards Understanding the Effectiveness of MAML*. https://arxiv.org/abs/1909.09157.

[2] https://www.rnel.pitt.edu/research/neuroprosthetics/intracortical-brain-computer-interfaces

[3] Degenhart, A. D. et al. Stabilization of a brain–computer interface via the alignment of low-dimensional spaces of neural activity. Nat Biomed Eng 4, 672–685 (2020).

[4] Finn et al. (2017), *Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks*. https://arxiv.org/abs/1703.03400.

[5] Karpowicz et al. (2024), *Few-shot Algorithms for Consistent Neural Decoding (FALCON) Benchmark*. ttps://www.biorxiv.org/content/10.1101/2024.09.15.613126v1.

[6] https://pub.towardsai.net/how-to-train-maml-model-agnostic-meta-learning-90aa093f8e46.