Interpretable, highly accurate brain decoding of subtly distinct brain states from fMRI using intrinsic functional networks & LSTM

By Hongming Li and Yong Fan From UPenn
PRESENTATION BY FLYN SEQUEIRA

Long Short Term Memory

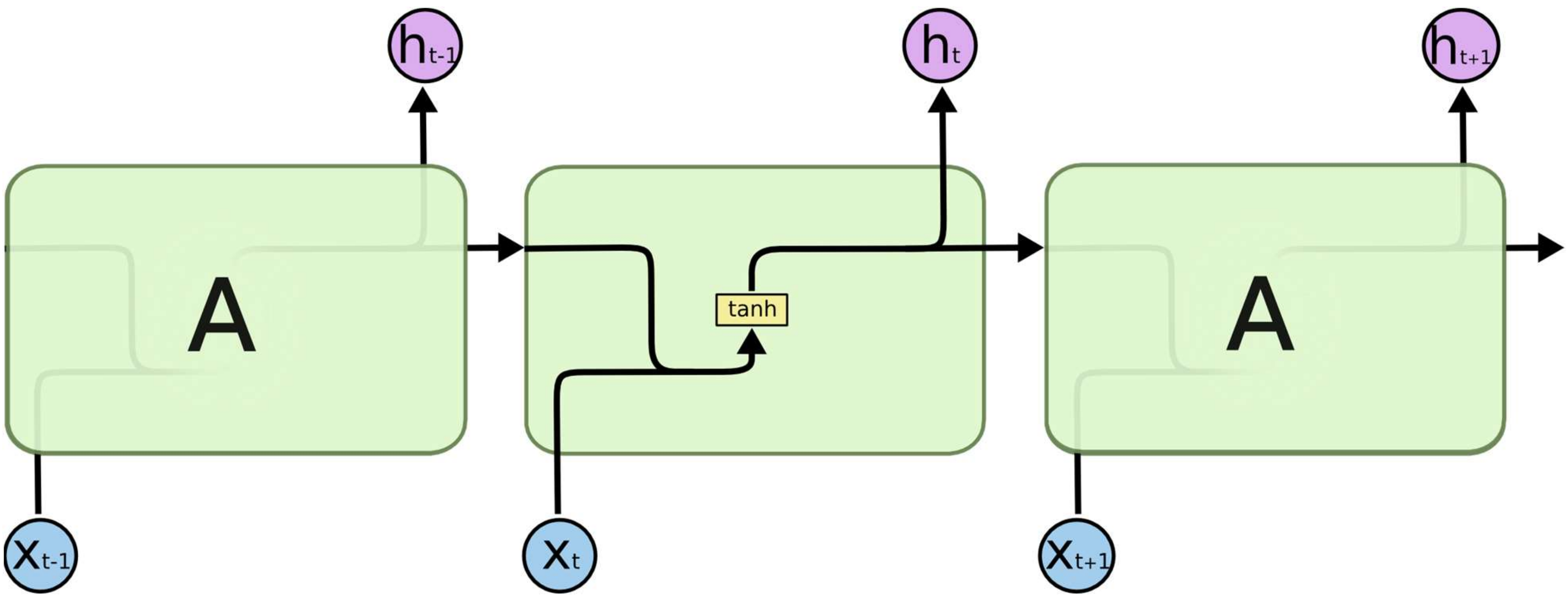
Or LSTM is a special kind of RNN BUT can learn long-term dependencies

Sepp Hochreiter(left) & Jurgen Schmidhuber (Right)

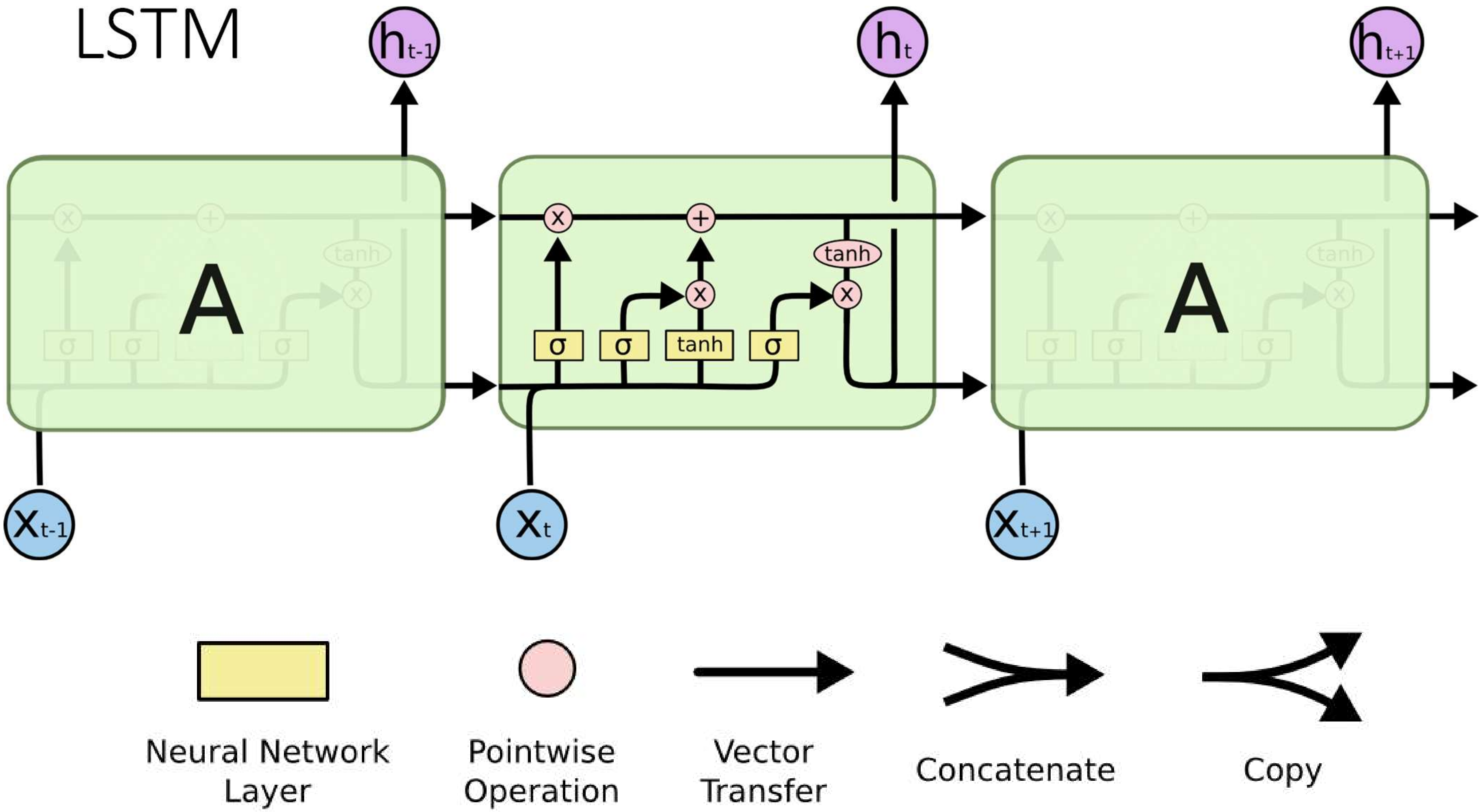
From Germany and Switzerland respectively

Check out the TEdX talk by Jurgen Schmidhuber with this QR Code

RNN



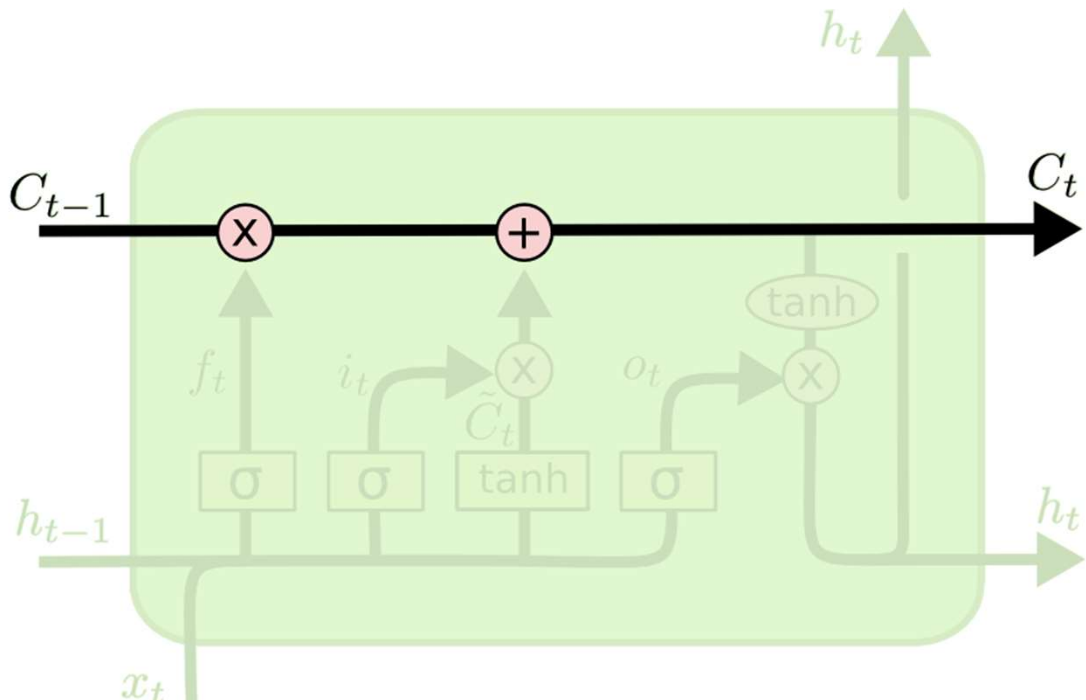
LSTM

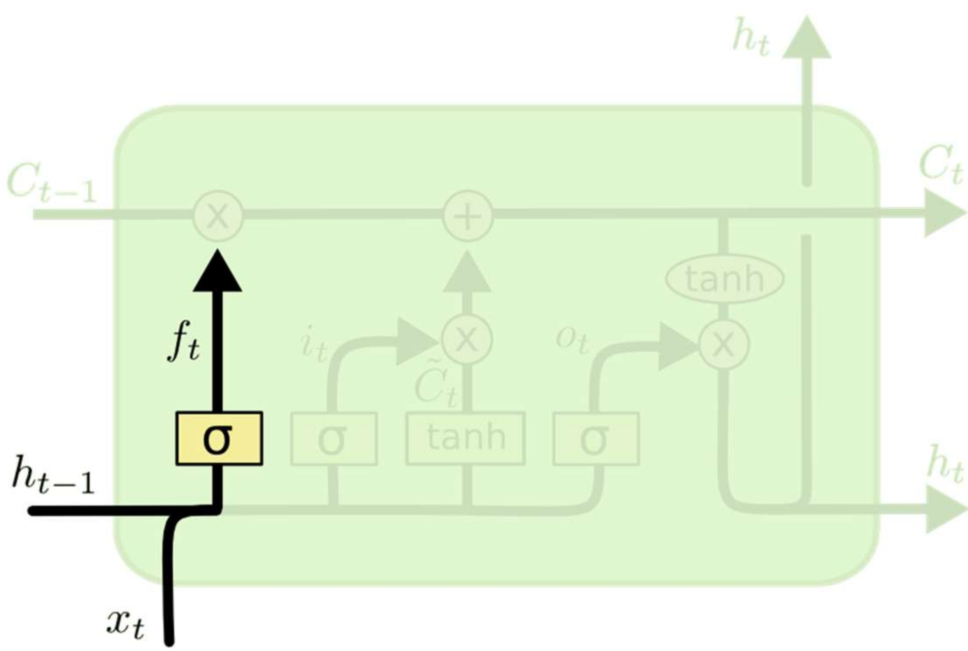


h_{t-1}
h_t
h_{t+1}
A
A
σ
σ
tanh
σ
tanh
X_{t-1}
X_t
X_{t+1}
Neural Network Layer
Pointwise Operation
Vector Transfer
Concatenate
Copy

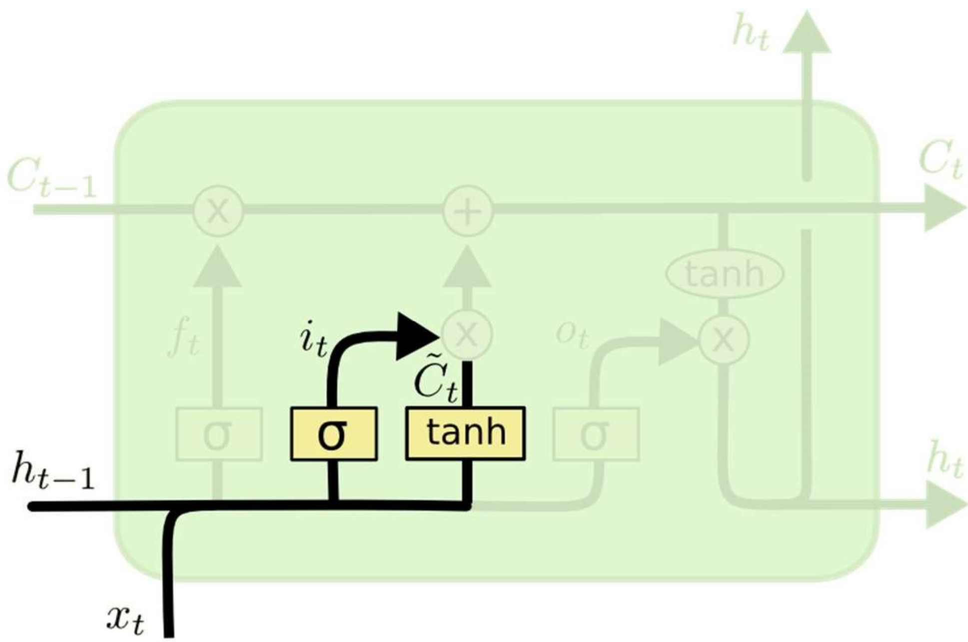
There is this Conveyor Belt

- Minor Interaction
- Information Flows easily with minimal changes



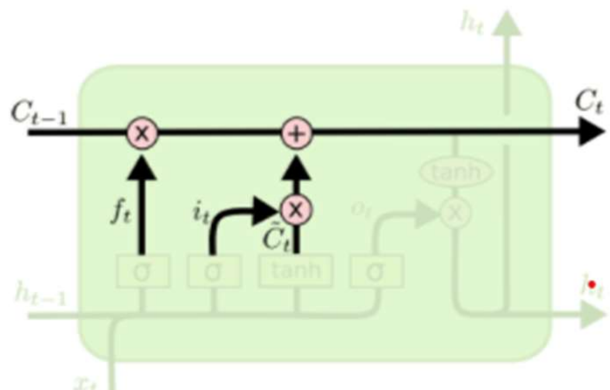


$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \;+\; b_f\right)$$

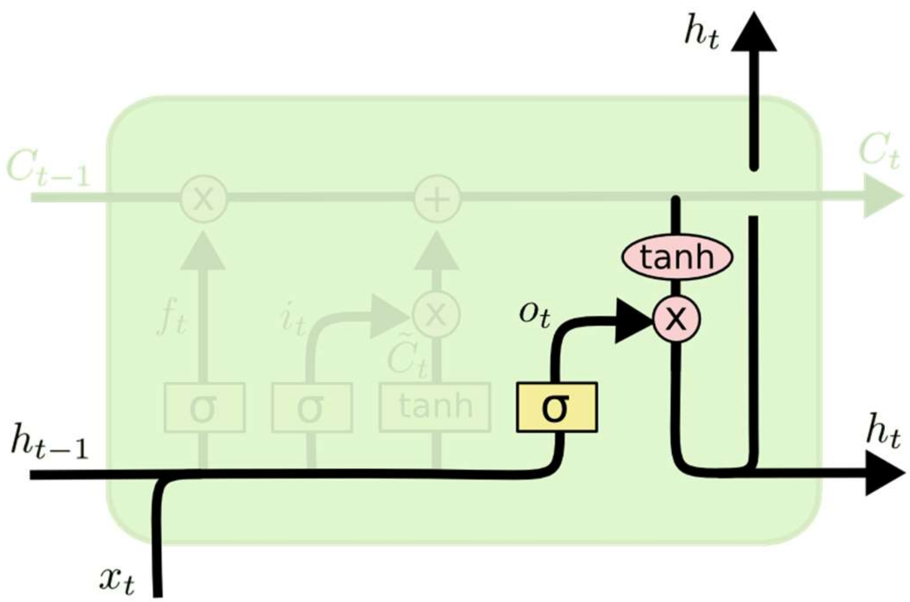


$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \ + \ b_i\right)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \ + \ b_C)$$



$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$



$$o_t = \sigma\left(W_o\ [\,h_{t-1}, x_t]\ +\ b_o\right)$$

$$h_t = o_t * \tanh\left(C_t\right)$$

WHY LSTM RNN for fMRI data

- Earlier brain decoding models analyzed functional signatures at
  - **Individual Time Points**
  - **Fixed length Time Window**
- E.g. SVM & Logistic Regression used the functional signatures at an **individual timepoint**
- Sequential fMRI data inherently has temporal dependencies. Utilizing it improves brain decoding performance
- **Fixed Length Time window** improved the performance over individual Time Point, **BUT**

Variable length Time Window is BETTER

Because ….
Brain states may change at unpredictable intervals
THEREFORE Long Short Term Memory RNN.

Advantage of LSTM RNN

- Better performance of Sequential Modelling
- Doesn’t require Fixed number of timesteps
- So far given better performance for Intra Subject EEG and ECoG brain decoding models
- CAN be adopted for Classification and Modelling Functional Dynamics of fMRI data
- Issue – Never applied for large Cohorts

This Paper’s GOAL

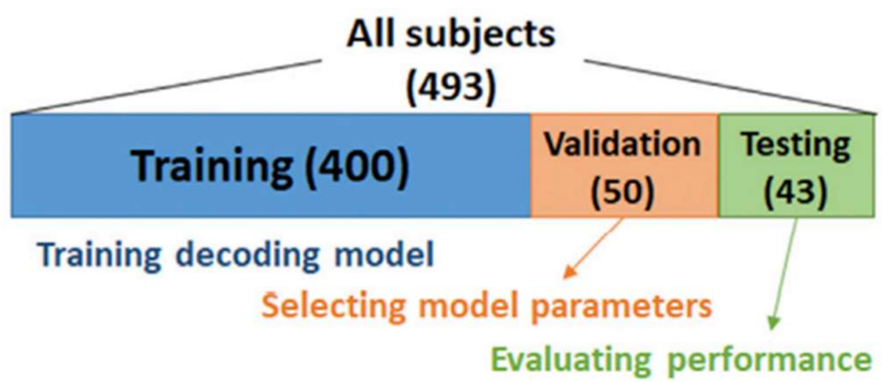
- Decode brain state from fMRI data.
  MORE SPECIFICALLY…
- Learn Mapping between functional signatures and brain states, with LSTM
  MORE SPECIFICALLY…
- Predicting brain states based on **working memory** fMRI data from HCP dataset
- Also showed promising results for decoding Motor and Social Cognition Tasks

Framework

LSTM is train based on Functional Signatures using **Functional brain decomposition technique**

Imaging Dataset

- 493 Subjects from HCP
- **Task fMRI data** were used to build and evaluate proposed decoding framework
- **Corresponding Resting state fMRI** data were used to obtain subject specific intrinsic functional networks (FNs)
- Focus of this study
  - WORKING MEMORY
  - MOTOR TASKS
  - SOCIAL COGNITION TASKS



Dataset Split

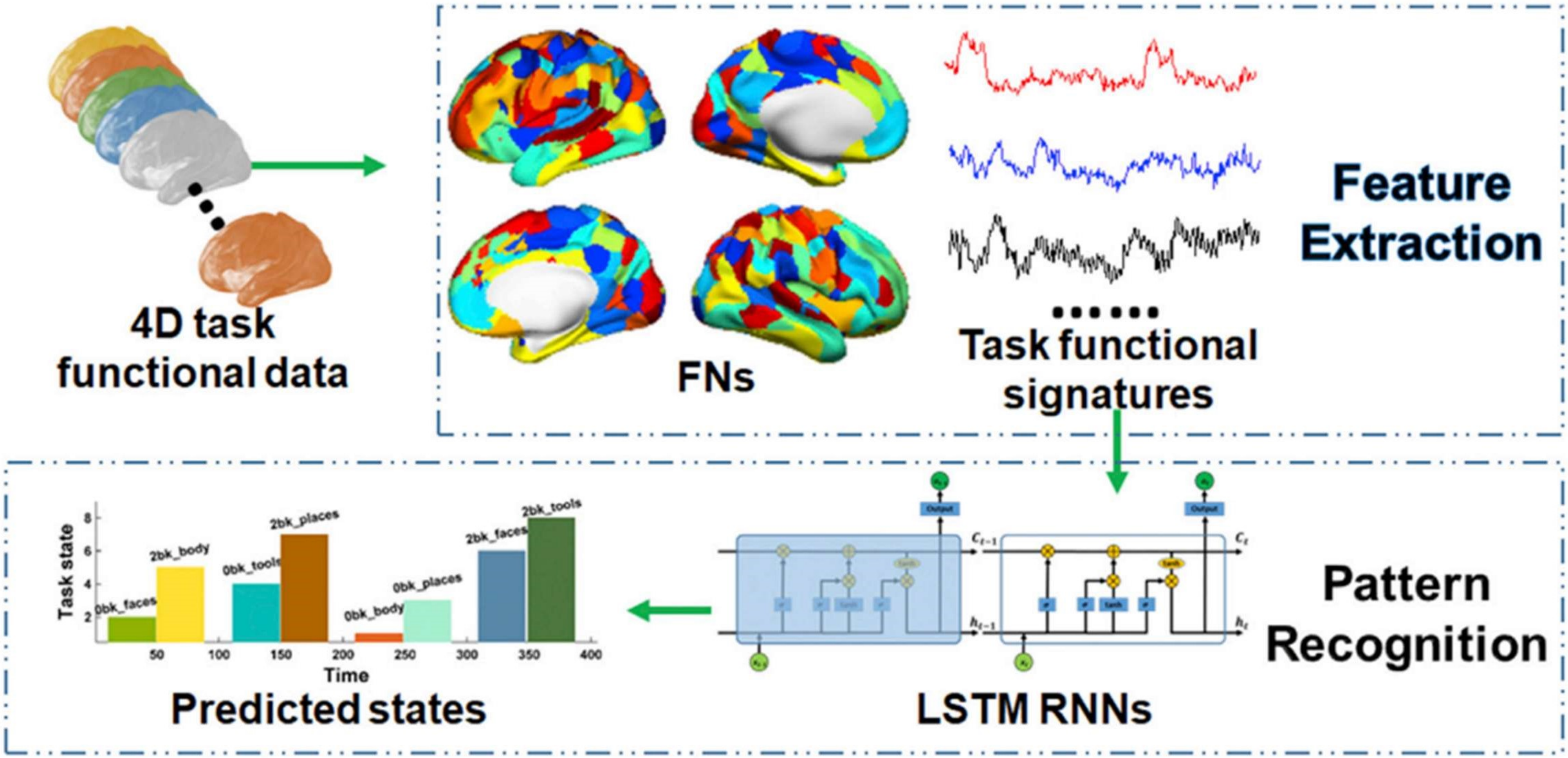
fMRI data characteristics included in this study.

fMRI data information					
	# of time points	TR (s)	# of events	Duration of event block (s)	Task events
Working memory	405	0.72	8	27.5	2-back and 0-back task blocks of tools, places, faces, and body
Motor	284	0.72	5	12	Left foot, left hand, right foot, right hand, tongue
Social cognition	274	0.72	2	16.6	Mental, random
Resting-state	1200	0.72	N.A.	N.A.	N.A.

- TR – Time Rate in Seconds for each Time Point
- **Events –** Events that stimulate the working Memory
- Duration of each event blocks
- ***n*-Back Task Measure:** paradigm to assess Working Memory

Feature Extraction

- We use Functional networks to extract functional signatures of brain state decoding
- 3D fMRI data could represent by a low-dimensional feature vector
  - Applicable to different brain decoding tasks
  - Provide better interpretability
- Instead of ROI at group level -> collaborative sparse brain decomposition model



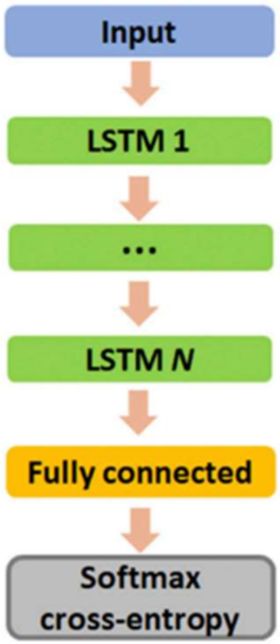
4D task functional data
FNs
Task functional signatures
Feature Extraction
Task state
0bk_faces
2bk_body
0bk_tools
2bk_places
0bk_body
0bk_places
2bk_faces
2bk_tools
Time
Predicted states
LSTM RNNs
Pattern Recognition

Features

- n subjects
- Resting state fMRI scan $D^i \in R^{T \times S}, i = 1, 2, \ldots, n,$
  for S voxels and T timepoints
- From this we get K Functional Networks $V^i \in R_+^{K \times S}$
- And corresponding K Functional Time Courses $U^i \in R^{T \times K}$
  (Using Collaborative Sparse Decomposition Model)
- More here…

Pattern Recognition: LSTM

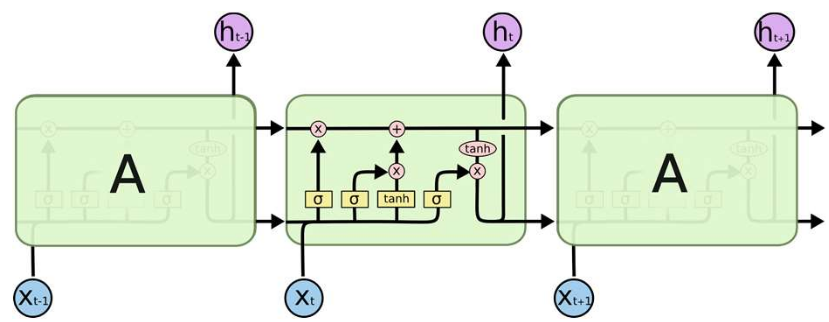
- Functional signatures F^i for n subjects where i = 1…n
  We use LSTM (The original model discussed above) to predict brain state at each timepoint. Two things are considered
  - Functional Profile
  - Dependence on previous time points
- The layers include
  - 2 Hidden LSTM Layers – To encode functional information with dependency
  - 1 Fully connected Layer – To map between learned feature representation and brain states



Functional representation encoded in each LSTM Layer

- Output of forget gate f_t^l :
- Input Gate i_t^l
- Hidden gate C_t^l
- Cell state h_t^l
- And input feature vector of l-th LSTM layer (l = 1,2) x_t^l

$$
\begin{aligned}
f_t^l &= \sigma\left(W_f^l \cdot [h_{t-1}^l, x_t^l] + b_f^l\right),\\
i_t^l &= \sigma(W_i^l \cdot [h_{t-1}^l, x_t^l] + b_i^l),\\
\tilde{C}_t^l &= tanh(W_C^l \cdot [h_{t-1}^l, x_t^l] + b_c^l),\\
C_t^l &= f_t^l * C_{t-1}^l + i_t^l * \tilde{C}_t^l,\\
o_t^l &= \sigma(W_o^l \cdot [h_{t-1}^l, x_t^l] + b_o^l),\\
h_t^l &= o_t^l * tanh(C_t^l),
\end{aligned}
$$



Still an Equation Slide …

- **Input features of first LSTM** network are **Functional Network signature**
- **Input of second LSTM layer** is the **hidden State Vector from the first LSTM network.**
- Fully Connected layer has S output – Corresponing to the number of Brain states
- Softmax converts these values to a probability of which state it might be

$$s_t = softmax\left(W_s \cdot h_t^2 + b_s\right),$$

- h_t^2 is the hidden state output of the second LSTM layer which encodes input functional signature at t-th time point

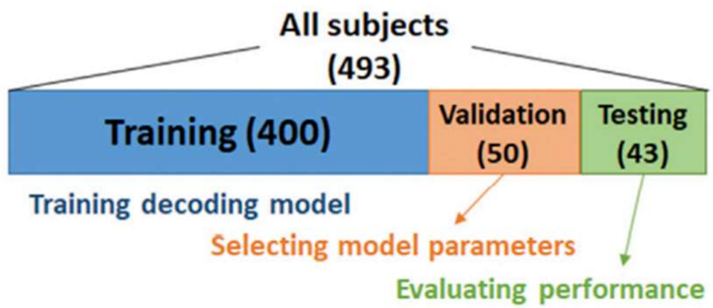
Optimizer

- **Softmax Crossentropy** – It is the standard optimizer used to classify with softmax activation in the last layer
- The softmax crossentropy is taken between real and predicted brain state and used as objective function to optimize the LSTM RNN Model.

Preprocessing Steps

1. Collaborative Sparse Brain Decomposition Model to resting fMRI
2. Number of FNs were determined based on resting-state fMRI data using MELODIC's LAPS criteria – Decomposition using cortical surface area and NO white matter, ventricle, etc.
3. Each subjects fMRI data had 3 matrices
   - 405X90 – Working Memory
   - 284X90 – Motor Tasks
   - 274X90 – Social Cognition Tasks
4. Magnitudes normalized using z-score and Input into LSTM RNN

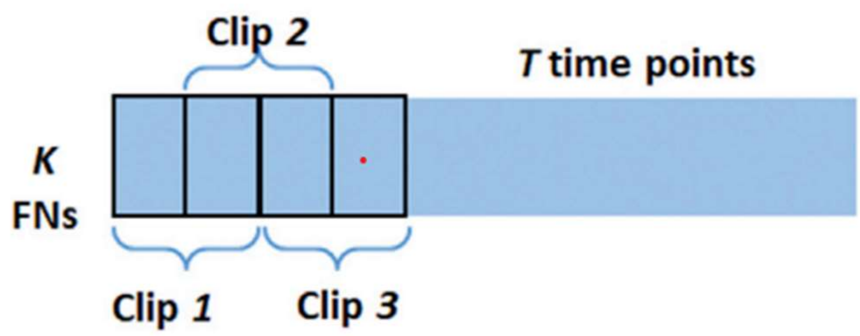
Validation



- 5 fold Cross Validation
- ISSUE: delay in Blood Oxygen Level Dependent (BOLD) response => Response not synchronized with presentation of Stimuli
- Solution: Brain State adjusted to BOLD delay (6 seconds or 8 Time Points. Remember 0.75 seconds is 1 timepoint)

Training

- 20 Time points overlapped in 40 Clips
- PURPOSE
  - Data Augmentation
  - To avoid same output all the time based on ground truth
  - Why? Ground truth is the same for most subjects.



Library Used - Tensorfow

- Optimizer – ADAM
- Learning Rate – 0.001 with Decay 0.1 at every 50k steps
- Training Steps 200,000
- Batch size – 32

Parameters Tested on

- Number of Hidden Layers – {1,2,3}
- Number of nodes in Hidden Layers - {32, 64, 128, 256, 512, 1024}
- The optimal value among these are selected based on their performance with validation dataset
- Parameter selection was done on Working Memory Task fMRI data only.
- Selected parameter used for all other experiments without any more optimization.
- WHY? Avoid overfitting towards better results.

Results compared with

Random Forest

Investigating what the LSTM encoder did

- Investigate association between 2nd hidden LSTM cell and brain state.
- How? Comparing Matrix Representation of
  - 128 hidden cells of LSTM layer – (Rows)
  - Time Point – Column
- The hidden cell rows were sorted on absolute weight of corresponding weights of trained prediction model.
- WHY? All the high impact LSTM cells would be on top, and easier to investigate.
- Then the maximum correlated LSTM cells can be identified for each task.

Performance Evaluation

- Removed Temporal dependency
- Decoding performance at each time point (LR)
- Subjects in-scanner performance and involvement level (RL)

Does it work without temporal dependency?

- Randomly jumble data along temporal dimensions.
- No temporal dependency was obtained by LSTM
- Therefore Prediction Performance decreased

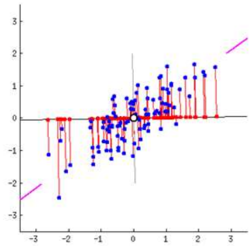
Sensitivity of Functional Networks (FN)

- Principal Component Analysis (PCA) based Sensitivity Analysis
- For 43 test subjects – 90 FN's signatures were set to 0, one by one.
- For each set to 0, we obtained a change in decoding accuracy.
- Therefore change in decoding accuracy can be represented as a matrix of 90 X 43]
- This is passed through PCA to obtain Principal components (PC)
- This encodes main direction of prediction changes w.r.t. functional signatures.
- Sensitivity analysis was carried out on working memory, motor and social cognition tasks.

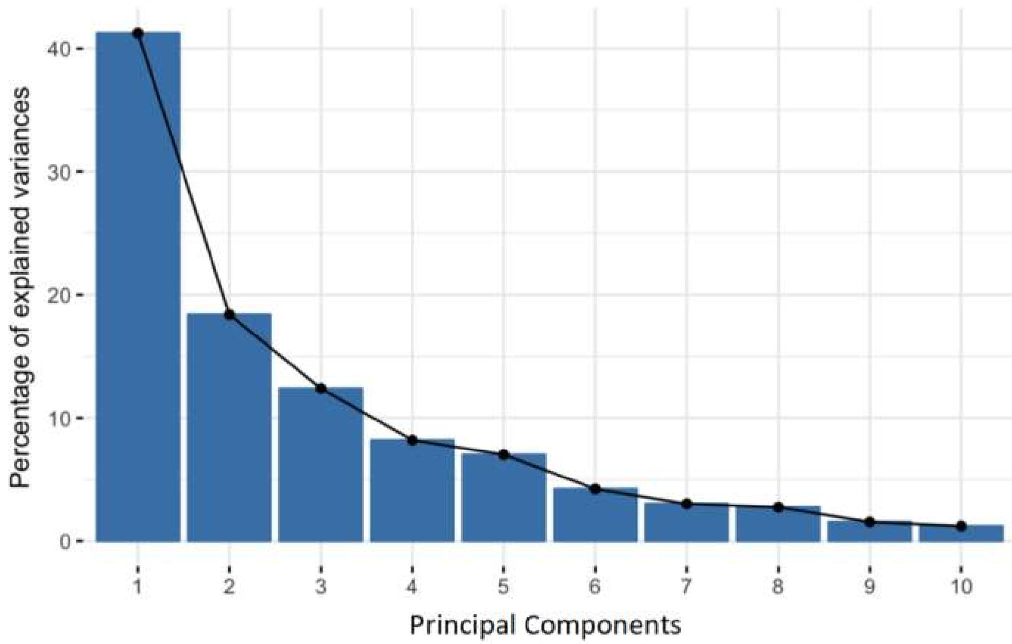
Principal Component Analysis

- Purpose – Dimensionality Reduction
- Reduces dimensionality of dataset with large set of variables to smaller set of variables that has most of the information.
- Principal components are new variables that are constructed as linear combinations or mixtures of the initial variables.

PCA



- These combinations are done in such a way that the new variables (i.e., principal components) are uncorrelated and most of the information within the initial variables is squeezed or compressed into the first components.
- So, the idea is 10-dimensional data gives you 10 principal components, but PCA tries to put maximum possible information in the first component, then maximum remaining information in the second and so on, until having something like shown in the scree plot below.

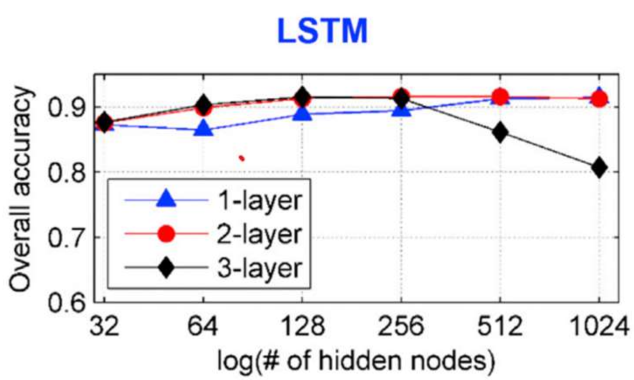

Percentage of explained variances
0
10
20
30
40
1
2
3
4
5
6
7
8
9
10
Principal Components

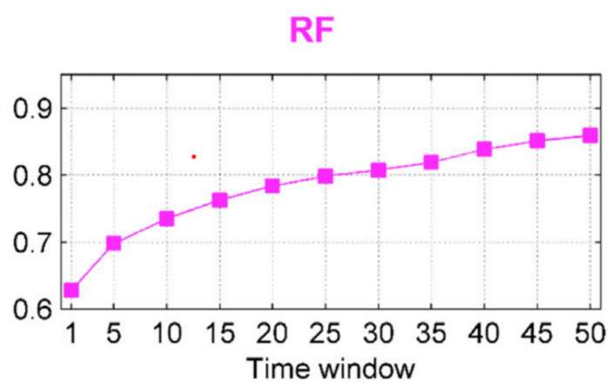
Problem

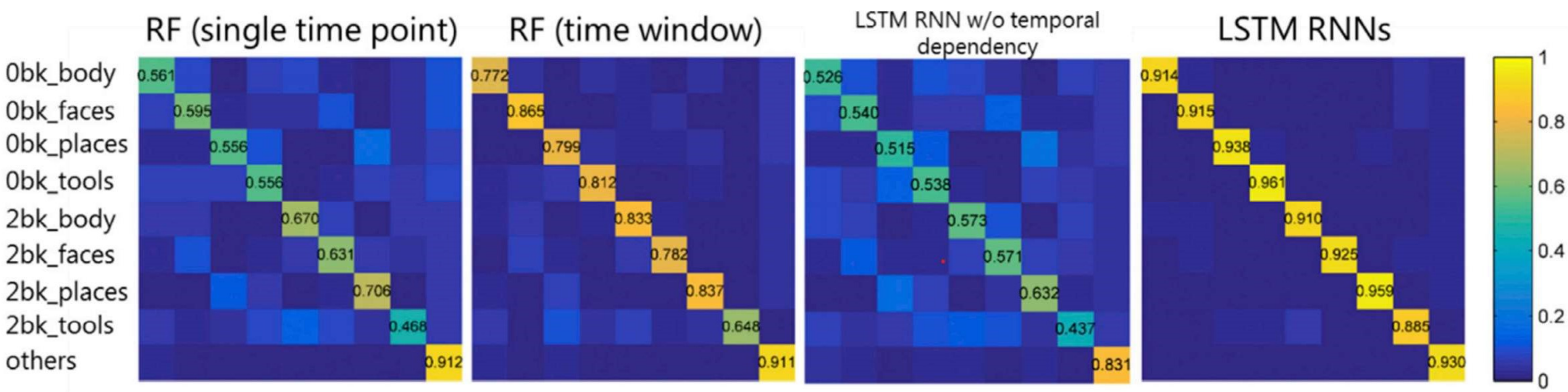
- An important thing to realize here is that, the principal components are less interpretable and don’t have any real meaning since they are constructed as linear combinations of the initial variables.

Results

- Accuracy was optimal with 2 hidden layer
- 3 hidden layers created overfitting and reduced accuracy.
- RF had fixed time window. Longer time-window gave better performance – Optimal time-window – 40 time points.

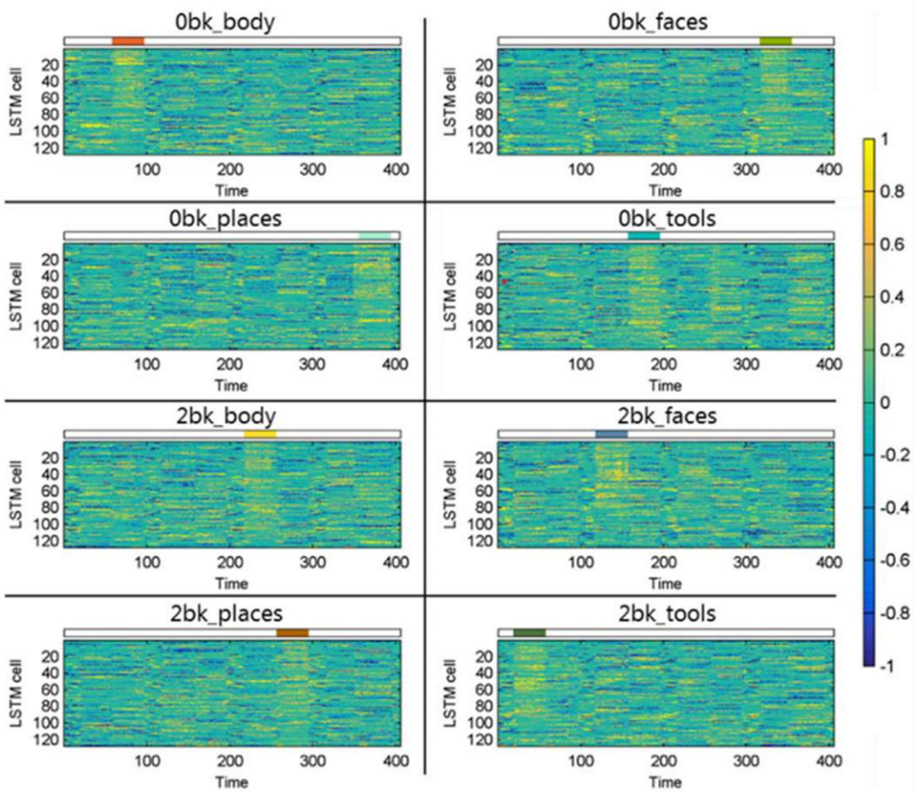


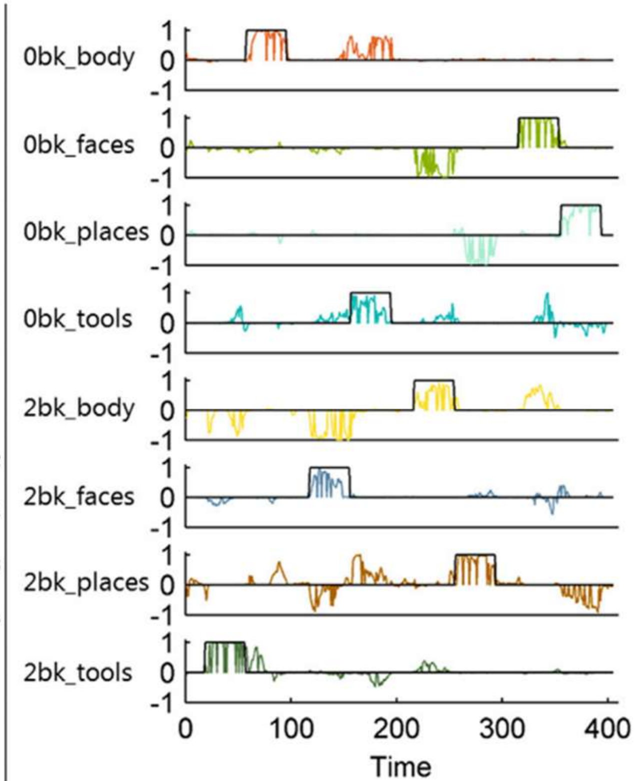




RF (single time point)
RF (time window)
LSTM RNN w/o temporal dependency
LSTM RNNs
0bk_body
0bk_faces
0bk_places
0bk_tools
2bk_body
2bk_faces
2bk_places
2bk_tools
others
0.561
0.595
0.556
0.556
0.670
0.631
0.706
0.468
0.912
0.772
0.865
0.799
0.812
0.833
0.782
0.837
0.648
0.911
0.526
0.540
0.515
0.538
0.573
0.571
0.632
0.437
0.831
0.914
0.915
0.938
0.961
0.910
0.925
0.959
0.885
0.930
1
0.8
0.6
0.4
0.2
0

Next is a diagram of activities of hidden cells in the second LSTM layer of one randomly selected testing subject on the working memory task fMRI dataset.

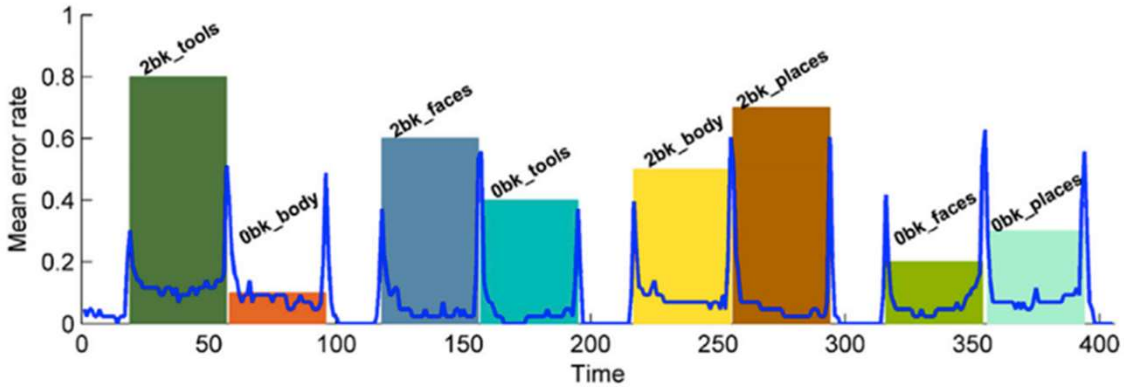




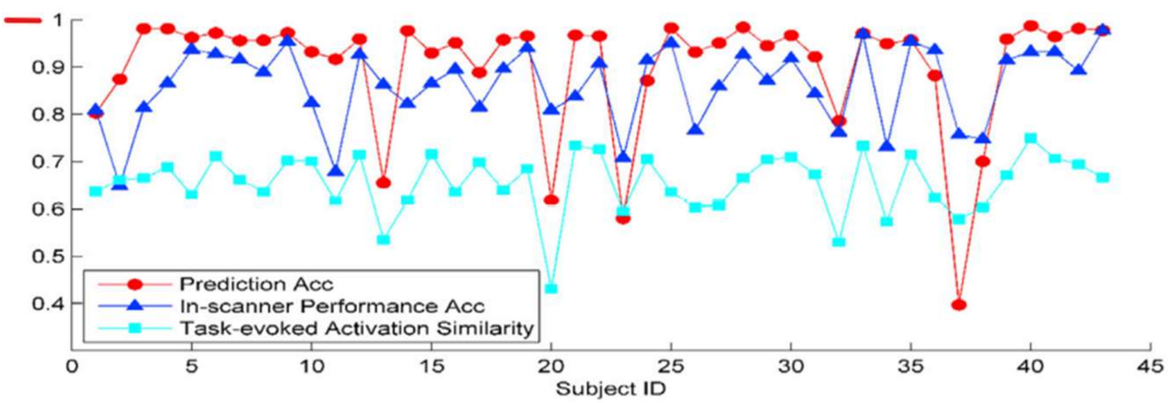
Activities of all 128 hidden cells, the cells in each subplot are shown in descending order according to absolute values of their prediction weights in the prediction model for each task event, the colored block above each subplot indicates the onset interval of each task event, and the colorbar indicates activities (values) of each cell on different time points;

Hidden cells with activities (values) maximally correlated with onset of each task event, the black line indicates the onset of each task event, and the colored curve is the activity of the LSTM hidden cell. The activity (value) of the hidden cell is scaled using its maximal absolute value for the visualization.

Performance analysis of the LSTM RNNs based brain decoding model on working memory task fMRI data.



Most of the misclassified time points are located at inter-state regions



The prediction accuracy is significantly correlated with the subjects' in-scanner performance accuracy and task-evoked activation similarity between training data and testing.

Conclusion

- LSTM better distinguished subtly distinct brain states based on task fMRI data than those built using RF
- Why - LSTM based brain decoding model extract temporal dependencies and learn the mapping between functional signatures and their corresponding brain states better
- It was capable of decoding brain states across different categories of tasks with improved performance.
- Computing functional connectivity patterns requires to specify a time window with a certain length to reliably estimate the functional connectivity.
- This is not a trivial task to determine an optimal time window
- Although the proposed brain decoding method has achieved better performance than RF, further efforts are needed in following aspects.
    - The brain decoding models were evaluated using block designed. It can be evolved for event designed task fMRI to investigate how the temporal duration of brain states impact the decoding performance.
    - FNs at a single spatial scale were used to compute functional signatures for the brain decoding.

Thank you

Any more questions?