**Project Report 2**

**Robust Split Federated Learning for U-shaped Medical Image Networks**

**Group Member and Contribution:**

**Group No: 8**

|  |  |  |  |
| --- | --- | --- | --- |
| **Team member Name** | **Team member Email IDs** | **R# IDs** | **Contribution** |
| Venkata Ramana Sathivada | venkata.sathivada@ttu.edu | R11850665 | * Splitting of the data and Models |
| Hemanth Reddy Nagireddy | henagire@ttu.edu | R11804804 |
| Madhu Kiran Jalla | mjalla@ttu.edu | R11848145 | * Forward propagation of data in the divided model and the shared data to store the info of output tensors |
| Sri Ram Koppaku | ksriram@ttu.edu | R11842335 |
| Charishma Chokkakula | cchokkak@ttu.edu | R11869144 | * Backward propagation from the model based on loss/cost function and propagating it from last model to previous |
| Manisha Ratna | mratna@ttu.edu | R11847768 |
| Manoj Metuku | mmetuku@ttu.edu | R11847308 | * Communication strategy between the overall model for start and end of run |
| Mani Madhuri Gokaraju | mgokaraj@ttu.edu | R11848127 |

**Problem Statement:**

Creating a U-Shaped Medical Image segmentation that uses both Federated and Split learning to increase the model privacy with decentralized machine learning and reducing the computation cost on the agents. The problem also is defined to tackle the drift between local and global model and their parameters that is caused due to data Homogeneity that is mostly encountered in Medical data ML cases.

**Problem Solution Status:**

The problem solution for the above problem can be divided into 3 parts:

1. Creating an FCT (Fully Convolution Transformer) to use as a Medical Image segmentation model without using split and federating learning.

We used CNN algorithm and Transformer to build this, in which CNN is used to learn effective image representations and combines them with the ability of transformers to effectively capture long-term dependencies in its inputs.

A CNN is a kind of network architecture for deep learning algorithms and is specifically used for image recognition and tasks that involve the processing of pixel data.

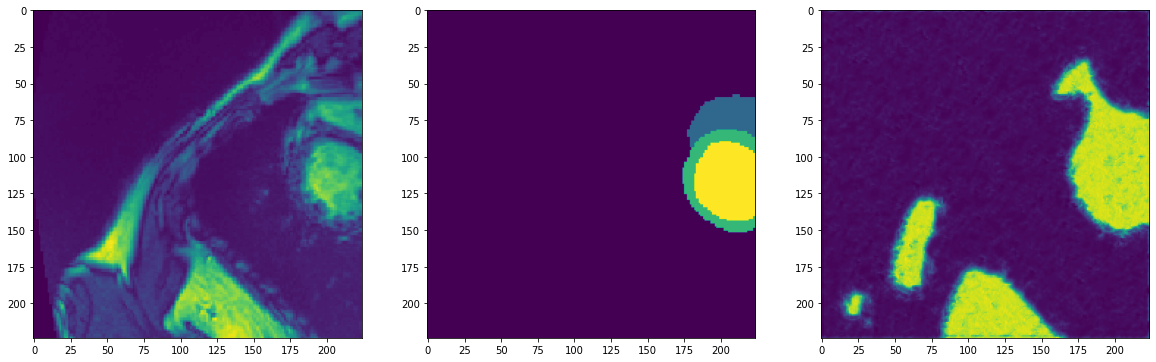
A Transformer is a type of Neural Network architecture that’s based on the concept of self-attention, where each position in the input sequence attends to all positions in the sequence to compute a weighted sum of the values at each position.

2. Dividing the model into 3 parts as `head`, `body` and `tail` where the data, the `head` (initial layers) and ‘tail’ (end layer) would be at agent(s) and the main model `body` with large layers would be at the computational server. This step incorporates federated and split learning.

3. Implementation of Dynamic Weight Correction Strategy (DWCS) to stabilize the training process and avoid model drift.

**Experiment Results:**

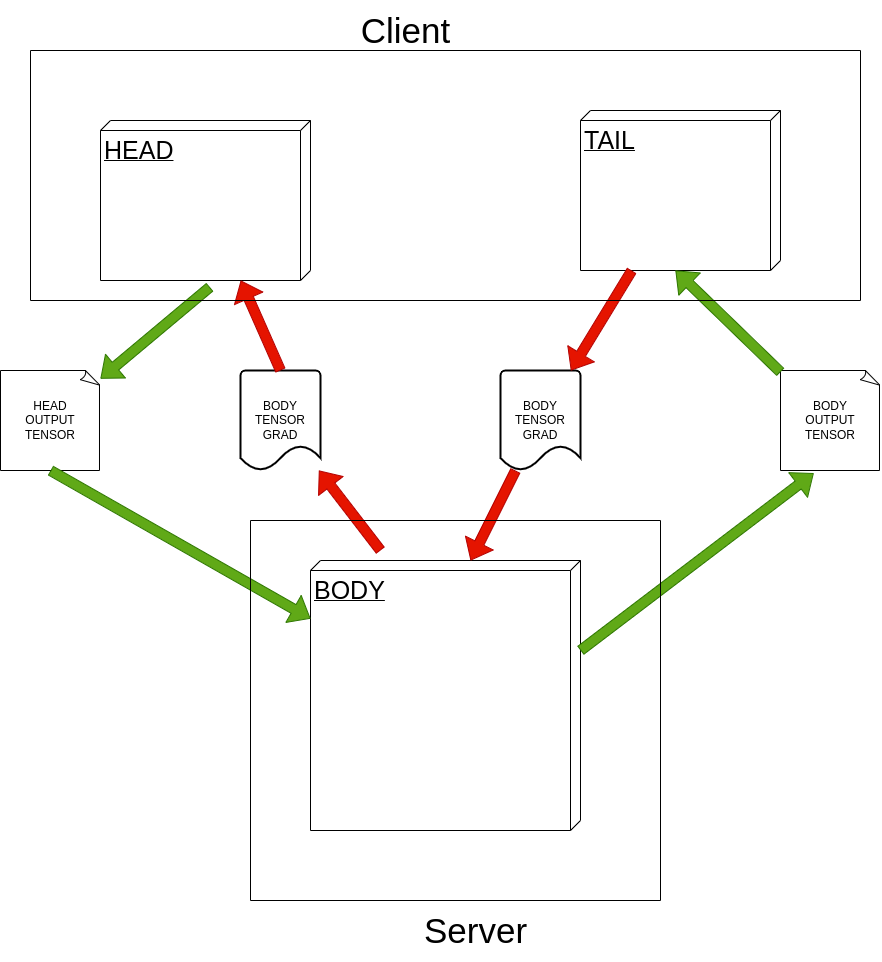
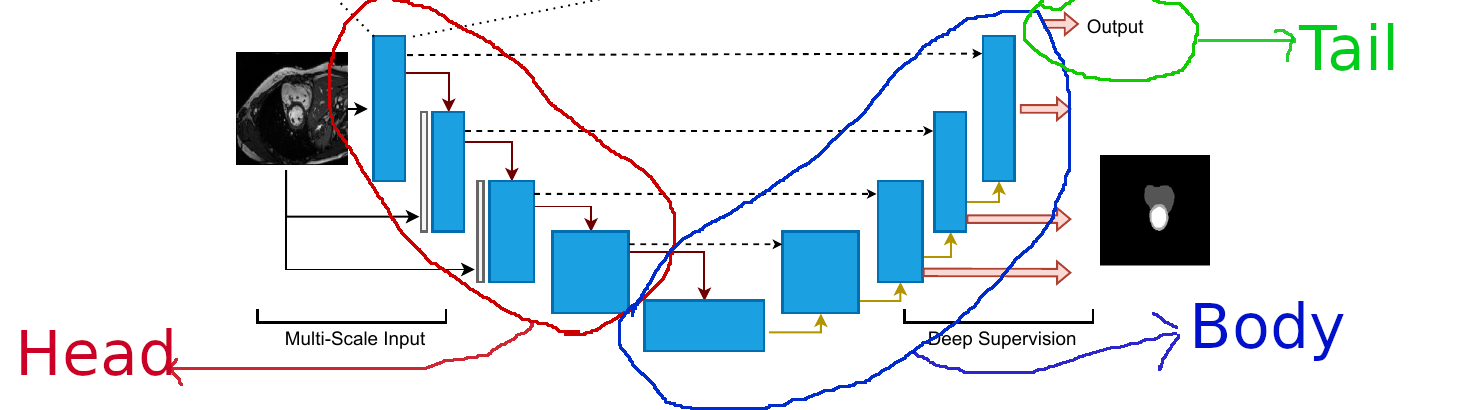
1. After the first centralized FCT has been created and trained on the data we were able to achieve the required result after around 200+ epochs on the training data. Dice score for the same was achieved to be around 0.90 to 0.91 which is nearly similar to the results obtained by the original paper.



2. After the same is accomplished we try dividing the models into 3 parts,

the head, the body and the tail and try saving the output of each model into a tensor for batch which is used by the next layer for forward propagation during for information exchange. Similarly the gradients are updated to another document that is picked up by the previous model and the parameters and updated.

Worflow of the overall model break up is mentioned shown below

The steps followed are as follows

1. The overall data for training is split in to 2 sets consisting of overall 80% of data selected for Client 1 and Client 2
2. For the Client 1 we created a Head Model which consists on the FCT encoder layers where input image data will be passed on to each layer in pyramid format
3. For the Overall Computational body the Body Model is created that consist of a single encoder without the image data input and rest of the similar decoder layer of FCT in same number as the Head Model
4. For the Client 1 the Tail model would be a sequential convolution, activation and sigmoid layer that would just give the final out to be compared to the original mask data vs predicted.
5. The input data goes into the Head layer and after forward pass within the model saves the output to a “head\_forward\_pass.hdf5” file and train data in “train\_values.hdf5” file
6. The body would read the “head\_forward\_pass.hdf5” file for input and after running the forward propagation would output the tensors which would be saved to “body\_forward\_pass.hdf5” file
7. The tail would read the “body\_forward\_pass.hdf5” file and calculate the prediction of segmentation which would be then compared to “train\_values.hdf5”
8. The loss calculated by tail above can be used to compute gradient and the tail grad can be saved in the “tail\_back\_prop.hdf5” file
9. The gradient of the tail model can be read by the Body Model for the file “tail\_back\_prop.hdf5” and same can be used to calculate its own gradient using back propagation method and its body gradient can be saved into “body\_back\_prop.hdf5” file
10. The gradient of body in “body\_back\_prop.hdf5” file can be read by Head Model and it gradient can be calculated by using back propagation method.
11. This signifies one epoch is completed and a complete forward pass and a complete back propagation has been carried out, and all the model parameters(weights and biases) would have been adjusted.
12. We can carry the same steps (from a.) all over again for a next set of iteration/epoch and we would be iterating it over around 200-300+ epochs.

**References:**

[1] We are taking the code from the GitHub Link:

<https://github.com/thanos-db/fullyconvolutionaltransformer>.

This link is the code implemented on the paper – “The Fully Convolutional Transformer for Medical Image Segmentation”. This code is being used, implemented and improved further to include federated and split learning. The approach to use this is taken from this paper i.e. – “**Robust Split Federated Learning for U-shaped Medical Image Networks**”.

[2] Paper: Robust Split Federated Learning for U-shaped Medical Image Networks.

The code reference is being used from this paper: The Fully Convolutional Transformer for Medical Image Segmentation.