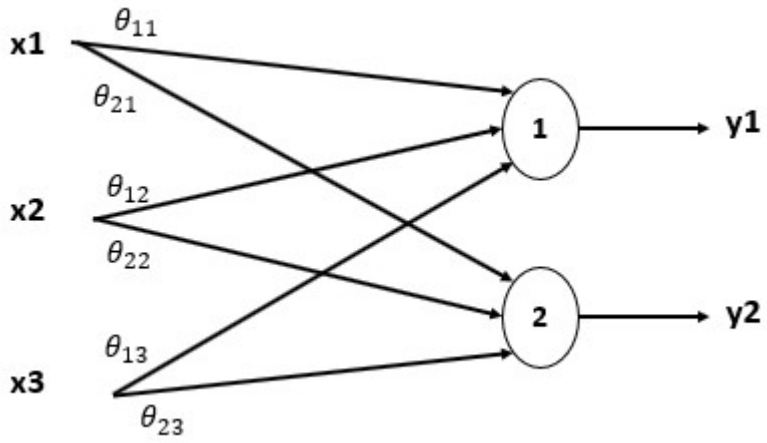
APPENDIX A

Proof for Novelty

The novelty of the proposed model (with neuronal level constraint) can be proved by comparing with models of synapse level constraints (EWC) and whole network level constraints (OGD) in terms of the structure and the performance on simple networks.

As explained in the proof, when the change in the parameters is orthogonal to the gradients of the earlier trained task, then the earlier task information is retained. Now we impose this to a synapse level, and the whole network level to see if this retaining of earlier task information still holds. For proving the novelty, we compare these 3 models in two ways.

1. In terms of parameter updation constraint equations.
2. Simulation studies.



SubFig 1 A Simple 2 neuron network

Consider a simple network with two neurons, which gets three inputs (SubFig 1). Consider the inputs X, the input parameters θ. From equation 6, we denote $\Delta\theta$ as g, and $(\theta - \theta^*)$ as d for demonstration purpose. Here g is the gradient values estimated from the first task data at the end of training, and d as the change in the parameter values while training on the new task. The formulation for the three models is shown below. The loss functions (constraint) are represented as $L_{synapse}$ for EWC, $L_{network}$ for orthogonal gradient approach, and L_{neuron} for the proposed model. Although $L_{network}$ isn't explicitly used in orthogonal gradient descent method, they both approach implies the same constraint.

$$L_{neuron} = (g_{11}d_{11} + g_{12}d_{12} + g_{13}d_{13})^2 + (g_{21}d_{21} + g_{22}d_{22} + g_{23}d_{23})^2$$

$$L_{synapse} = g_{11}^2d_{11}^2 + g_{12}^2d_{12}^2 + g_{13}^2d_{13}^2 + g_{21}^2d_{21}^2 + g_{22}^2d_{22}^2 + g_{23}^2d_{23}^2$$

$$L_{network} = (g_{11}d_{11} + g_{12}d_{12} + g_{13}d_{13} + g_{21}d_{21} + g_{22}d_{22} + g_{23}d_{23})^2$$

Here, The subscripts represent the neuron index, and the input index respectively.

By comparing the equations, we can see that the three approaches differ in polynomial terms. Among the three equations, with respect to the polynomial terms, $L_{network}$

is the superset of the other two equations. In comparison, L_{neuron} is a superset of $L_{synapse}$ and a subset of $L_{network}$. $L_{synapse}$ is a subset for both $L_{network}$ and L_{neuron}.

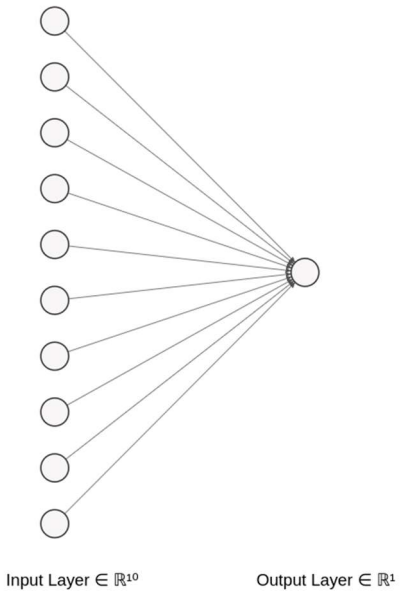
$L_{network}$ gives the constraint at the whole network level, which doesn't hold the locality principle. $L_{neuron,}$ since it gives the constraint at the neuronal level, preserves the locality principle. $L_{synapse}$ by default follows locality principle. Though $L_{synapse}$ and L_{neuron} look more biologically plausible, the $L_{synapse}$ gives the constraint at the synapse level, which means when the constraint parameter is big, it doesn't allow the updation of the related parameter. But L_{neuron} allows the updation even if the magnitude is higher(as long as the orthogonality is maintained, the parameters can be updated to any magnitude).

From the above equations, we can observe that they will be equal only if one of the vectors (either g or d) is 0. In all the other conditions, it is well known that they are not same.

Simulation studies:

The abovementioned three approaches are compared using simulations on two simple logistic regression networks for proving novelty (One network with a single neuron, and the other with two neurons).

Logistic Regression (Single Output)



SubFig 2 Simple logistic regression with 10 inputs

Consider a logistic regression problem with 10 inputs and a single output (subfig 2). For demonstration purpose, a single data point is considered for each task.

Two tasks are trained one after the other sequentially (250 epochs each). subfig 3 and 4 shows the loss comparision between the synapse level constraint and the neuronal level constraint while training the tasks. The loss for the two approaches is estimated at the end of training the second task. Here at the end, with the synapse level constraint (EWC), the average loss comes to .0053 (task1: 0.0037, task2: 0.0068). While with the neuronal level constraint (the proposed model) the average loss becomes 0.00019 (task1: 7.20e-08, task2: 0.00038).

Here the neuronal level and the network level constraints are the same since we are dealing with a single neuron.

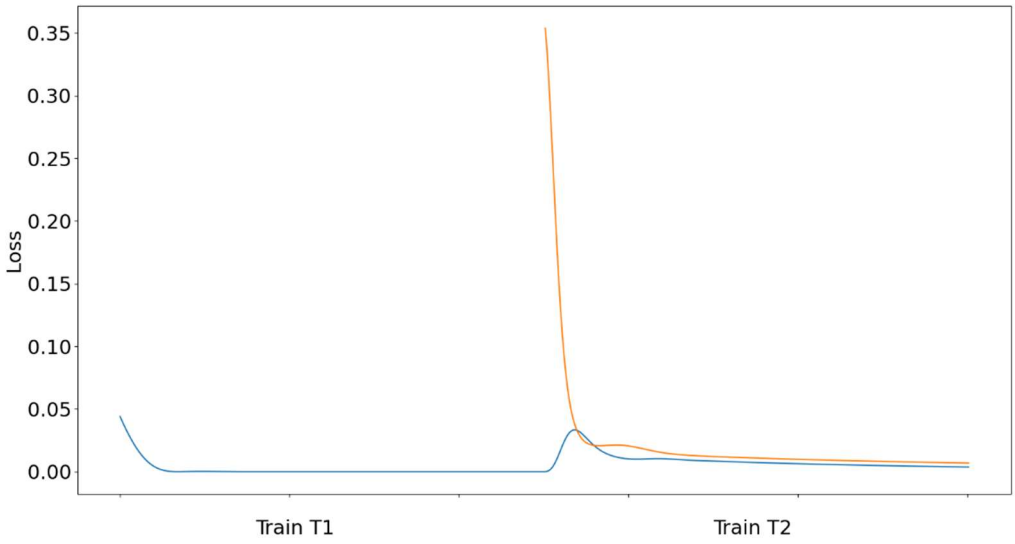


Figure 3 Logistic Regression loss with EWC

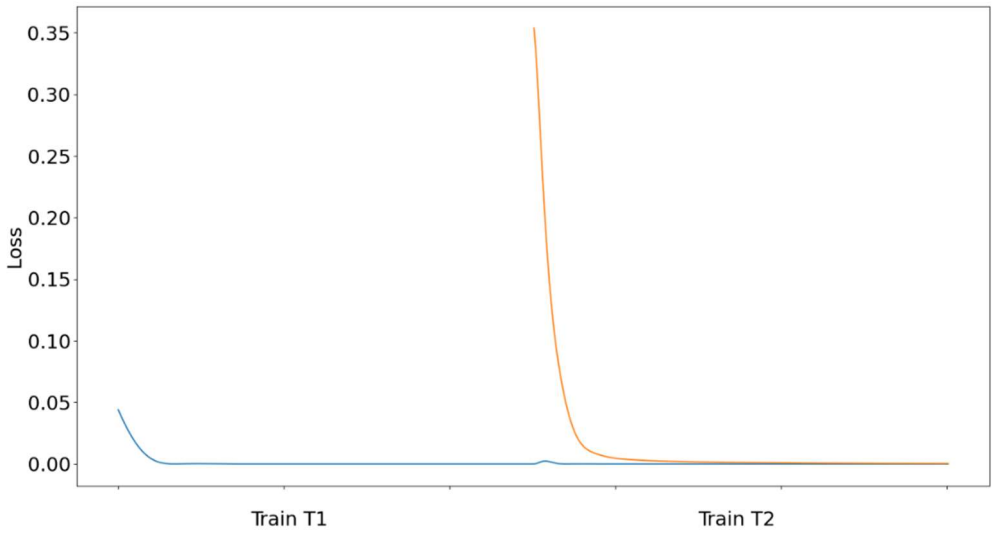


Figure 4 Logistic Regression loss with the proposed model

Logistic Regression (Two Outputs)

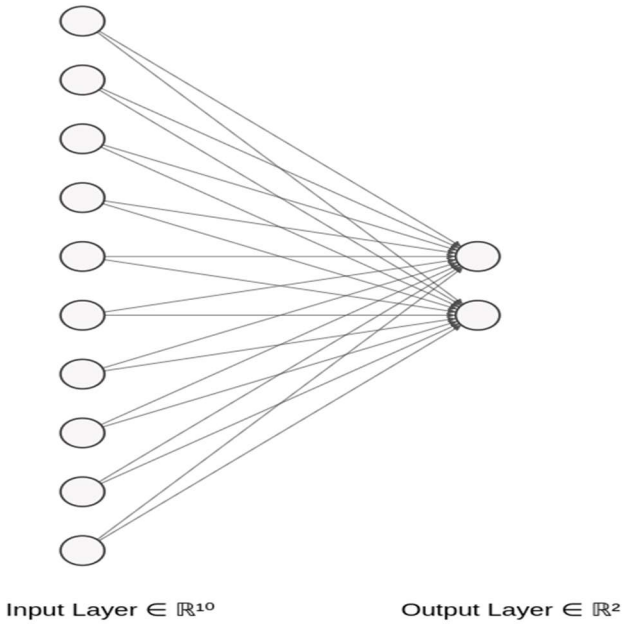
Consider a logistic regression problem with 10 inputs and two outputs (subfig 5).

Two tasks are trained one after the other sequentially. subfig 6,7 and 8 shows the loss comparision between the synapse level constraint (EWC), the neuronal level constraint (proposed model) and the network level constraint (orthogonal gradient descent) while training the tasks. The average loss for all the three approaches is estimated at the end of training the second task.

Here at the end of training, the synapse level constraint yields the average loss of 0.0004 (task1: 0.00081 , task2: 1.0074e-05). the network level orthogonality constraint yields

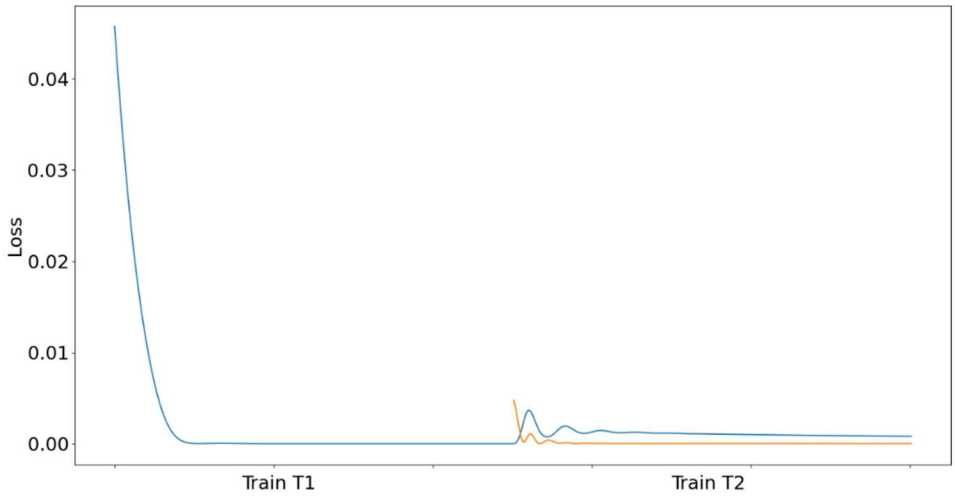
0.00055 (task1: 0.0011, task2: 1.776e-14). the neuronal level orthogonality constraint produces 1.7e-13 (task1: 3.255e-13, task2: 1.998e-14).

From the above two approaches, it can be observed that the proposed model is unique in terms of the formulation and the objective of the constraint.

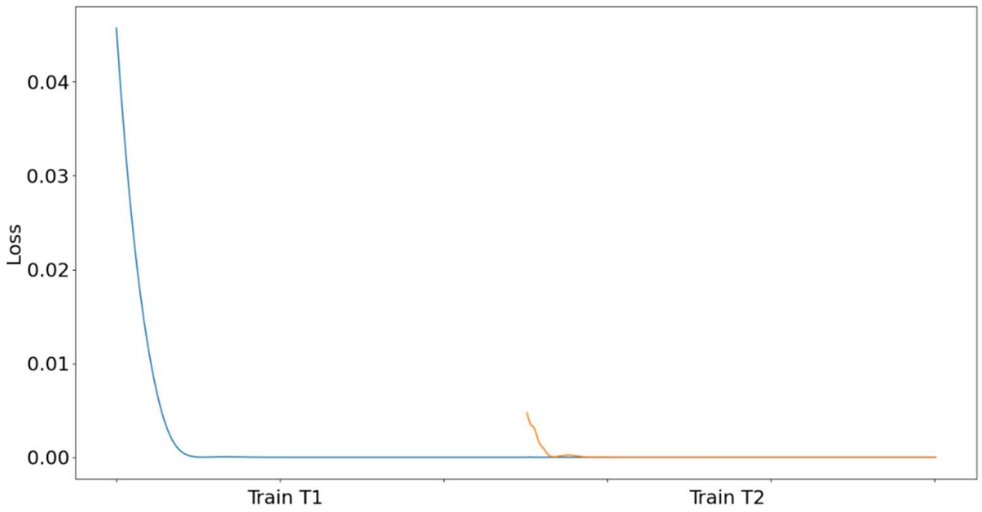


.

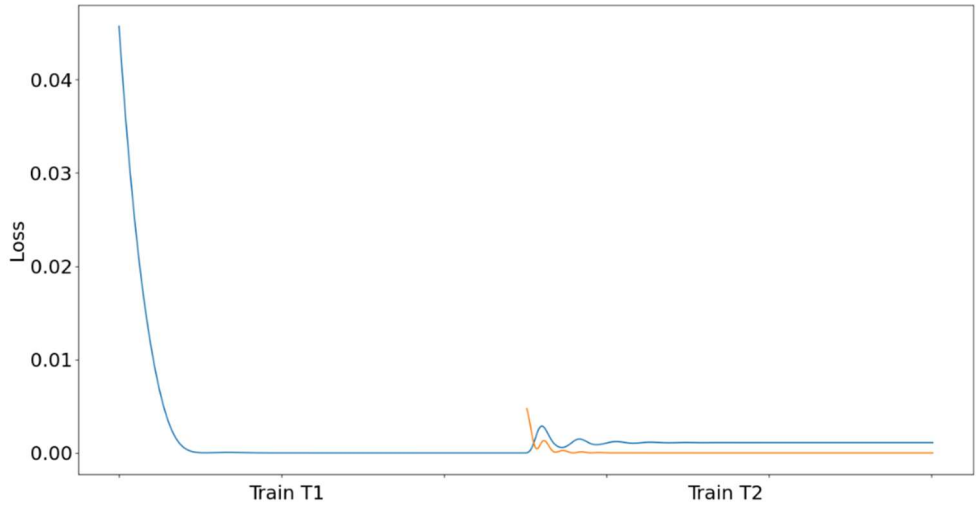
SubFig 5 Logistic regression with 2 neuron



SubFig 6 Logistic Regression loss with synapse level constraint (EWC)



SubFig 7 Logistic Regression loss with neuronal level constraint (proposed model)



SubFig 8 Logistic Regression loss with network level constraint (Orthogonal Gradient Descent)