**📘 README Summary of Concept (Baseline MLP vs DynamicGate-MLP)**

**1. Compare concepts**

| **entry** | **Baseline MLP** | **DynamicGate-MLP** |
| --- | --- | --- |
| What you'll learn | Weight , Bias | Weight , Bias **+ Gate Logit** |
| Values that are updated | Change through learning | + Gate Log (→ Gait Probe) |
| Gate Parameters | none |  |
| Forward |  |  |
| Backward |  | + (Learning with STE) |
| Structural changes | Dense | Connections are turned off or on during the learning process (rarefiing) |
| Interpretability | Simple Weight Size  base | You can see directly which connections survived with the Gate matrix (0/1) |

**2. Commonality**

* Both **learn Weight and Bias and create a classifier.**
* Both can be trained on datasets such as MNIST using CrossEntropy Loss.
* The forward/backward learning structure is based on the same PyTorch MLP.

**3. Differences**

* **Baseline MLP**: All connections are maintained until the end. → Weight, only Bias changes.
* **DynamicGate-MLP**:
  + Gate Logit → Sigmoid → Gate Prob () → Threshold → Gate Hard (0/1)
  + Some connections are turned off as it learns(0) → **the network is effectively diluted**
  + In Forward, the connection that is turned off is completely excluded from the operation.
  + In Backward, gradient is also delivered to Gate Logit via STE.

**4. Why is it needed?**

* **Efficient inference**: Although training is heavy, it can be pruned with Gate Hard in the inference phase→ reducing the amount of computation/model size.
* **Controllability**: Adjust the intensity of rarefaction through hyperparameters (β, τ) → adjust the trade-off between speed and accuracy.
* **Interpretability**: Directly interpret which connections are important with a gate matrix (Prob/Hard).
* **Fusion of Dropout/Pruning**: Dynamic like Dropout, Removing Permanent Connections Like Pruning.

**5. Structure Comparison Diagram**

Insert the picture below into the README to intuitively show the difference.

**Baseline MLP vs DynamicGate-MLP**

graph TD

A[inputx (784)] --> B[fc1: 784→256 W1,b1]

B --> C[ReLU]

C --> D[fc2: 256→10 W2,b2]

D --> E[output y]

A2[inputx x (784)] --> B2[fc1: 784→256 W1,b1]

B2 --> G[Gate matrix G1 (learn)]

G --> C2[ReLU]

C2 --> D2[fc2: 256→10 W2,b2]

D2 --> H[Gate matrix G2 (learn)]

H --> E2[output y]

classDef base fill=#CDE,stroke=#333,stroke-width=1px;

classDef gate fill=#FDD,stroke=#933,stroke-width=1px;

class B,D base;

class B2,D2 base;

class G,H gate;

* **Left**: Baseline MLP — Learn only Weight and Bias
* **Right**: DynamicGate-MLP — Weight, Bias learning + Gate matrix (Prob/Hard) learning

**6. One-line summary**

While Baseline MLP simply **learns W, b,**   
DynamicGate-MLP **learns W, b+Gate together to**   
create a sparse structure where "only the necessary connections survive"**, ensuring inference efficiency and interpretability.**