Real world systems need to re-train when requirements change

Use (sequential) **continual learning**

For **safety** we prefer preserving already validated solutions over learning new solutions (**stability over plasticity**).
- Generic regularized training objective:
  \[ \mathcal{L}_i(\theta) + \lambda G_{1:i-1}(\theta, \theta^{*}_{1:i-1}) \]
  - loss for the new task
  - stability / plasticity parameter
  - loss to preserve the configuration that solves the old tasks

- Popular methods (eg. EWC) use an L2 objective:
  \[ \mathcal{L}_i(\theta) + \lambda \sum_j F_{jj} (\theta_j - \theta^{*}_{j,1:i-1})^2 \]

- An alternative is to use an L1 objective:
  \[ \mathcal{L}_i(\theta) + \lambda \sum_j C_{jm}^n |\theta_j - \theta^{*}_{j,1:i-1}| \]
L1 vs L2 Objective

L1 objective often leads to solutions with stronger preservation.

2-task, Sim-EMNIST

2-task, Split-MNIST
Other Contributions

- We **quantify** amount of forgetting in continual learning for classification
  - we thus propose a per-parameter importance metric for the L1 regularization loss
  - and propose methods that:
    - never forget beyond a certain specified amount (finer control)
    - freeze weights in a more principled manner to achieve higher stability (Fisher freezing)
- We conduct **experiments** on grayscale and real world datasets

Please visit our poster for more details:

**Simple Continual Learning Strategies for Safer Classifiers**