Continual Learning with Deep Generative Replay

NIPS 2017

Hanul Shin
Massachusetts Institute of Technology
SK T-Brain
skyshin@mit.edu

Jung Kwon Lee*, Jaehong Kim*, Jiwon Kim
SK T-Brain
{jklee,xhark,jk}@sktbrain.com

2023.03.10 내부미팅

이소영

Contents

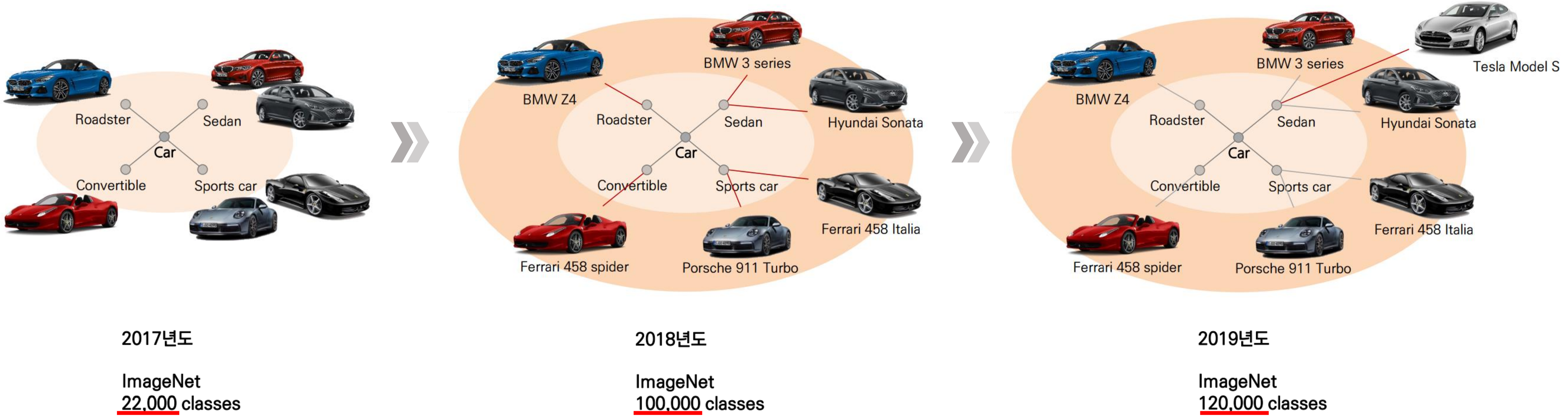
Abstract

- Solving multiple task (assuming sequential) → catastrophic forgetting
  - Replaying all previous data : large memory, access limit (in real world)

- **Deep Generative Replay** : Cooperative dual model architecture
  - Deep generative model (“**generator**”) + task solving model (“**solver**”)
  - Hippocampus: generative nature, short-term memory system in primate brain
  - Training data = sampled previous tasks data +(interleave) new task data
  - Test : several sequential learning settings involving image classification tasks

Introduction & Related Works – continual learning

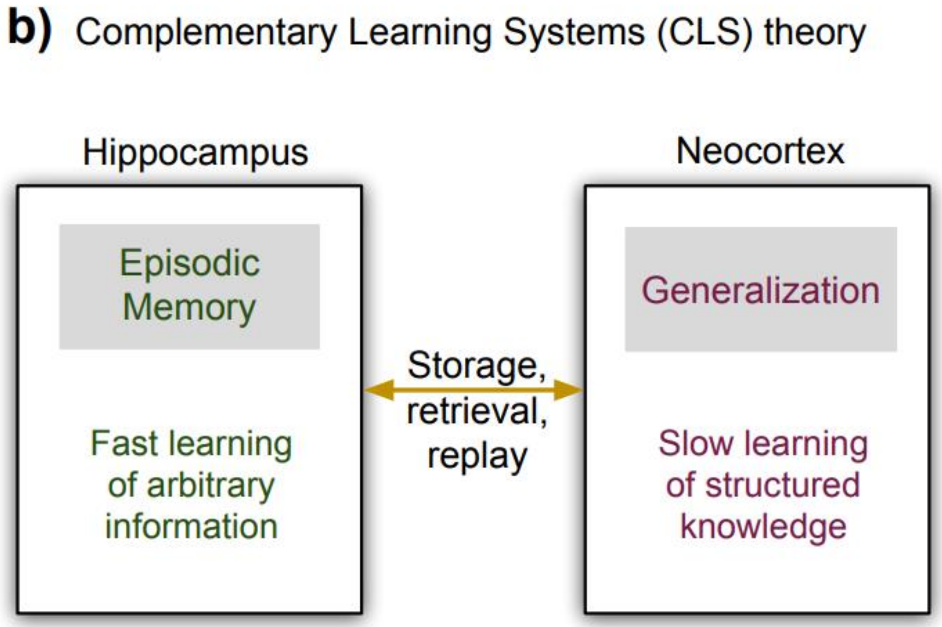
- Supervised learning : 데이터에 대한 정답이 주어진 상태에서 학습
  - Challenges : Incomplete, Growing Datasets (데이터는 시간의 흐름에 따라 끊임 없이 성장한다)
  - 연구의 방향이나 시장의 수요에 따라 데이터/클래스가 세분화
  - 증감된 데이터/클래스에 따라 새로운 task 부여



https://www.slideshare.net/NaverEngineering/lifelong-learning-for-dynamically-expandable-networks

Introduction & Related Works – continual learning

- Neurophysiological aspects (humans and large primate brain)
  - Complementary Learning Systems (CLS) & Plasticity and stability dilemma
    - 배경 지식은 일반화하여 장기기억으로 저장되어 있음
    - 새롭게 지식을 배울 때 배경지식을 활용하고, 새로운 정보(단기기억) 중 중요도를 잘 판별해서 배경지식(장기기억)으로 저장
    - 새롭게 무언가 배웠다고 해서 이전에 배운 것들을 잘 못하지 않음



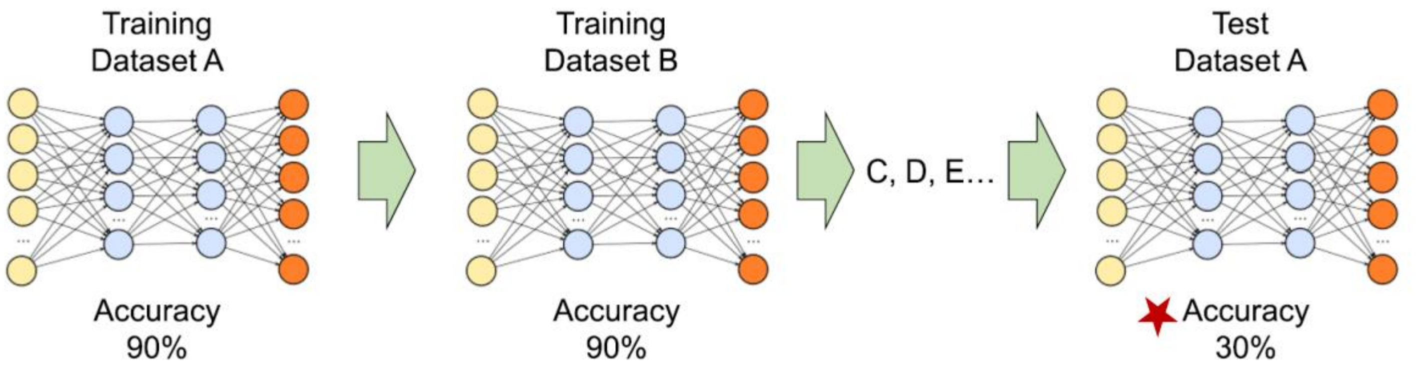
Parisi et al. (2019) Continual lifelong learning with neural networks: A review.

Introduction & Related Works – continual learning

- **Continual learning** scenario
  - 여러 task를 하나의 모델에 **순차적**으로 학습하여 최종적으로 **모든 task의 수행**이 가능한 모델을 학습

Continual Learning ≈ Lifelong Learning ≈ Incremental Learning ≈ Online Multi-task Learning

- Challenges : **catastrophic forgetting** (McCloskey and Cohen, 1989)
  - model's performance on **previously learned tasks abruptly degrades** when trained for a new task
  - 새로운 task에 대해 학습하게 되면 신경망모델이 이전에 배운 task에 대해서는 네트워크가 까먹는 현상



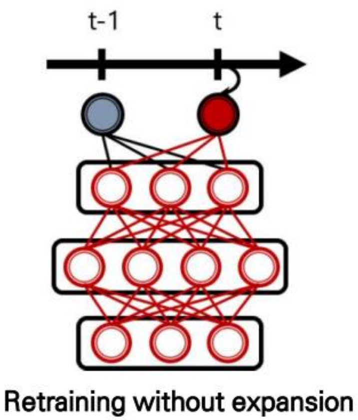
Introduction & Related Works

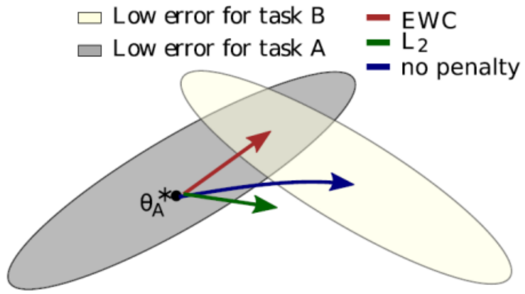
- Continual learning 대표적인 접근 방법

1. Regularization approach : 현재 task update 시, 이전 task에서 중요했던 파라미터는 조금만 변경되도록 regularization term 추가
2. Dynamic Architecture : 새로운 task를 수용하기 위해 네트워크 구조를 동적으로 변경
3. **Memory Replay : 이전 task들의 data를 저장, 현재 task update에 사용. 생물학적인 기억 메커니즘을 모방하자는 아이디어**

Regularization approach 예시

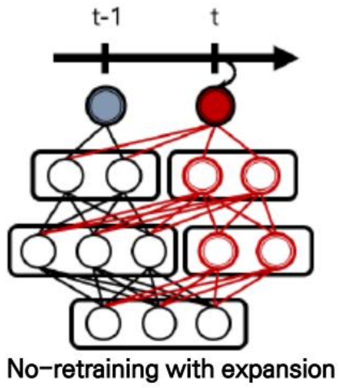
EWC (Elastic Weight Consoliation, Kirkpatrick et al., 2017)





Dynamic Architecture 예시

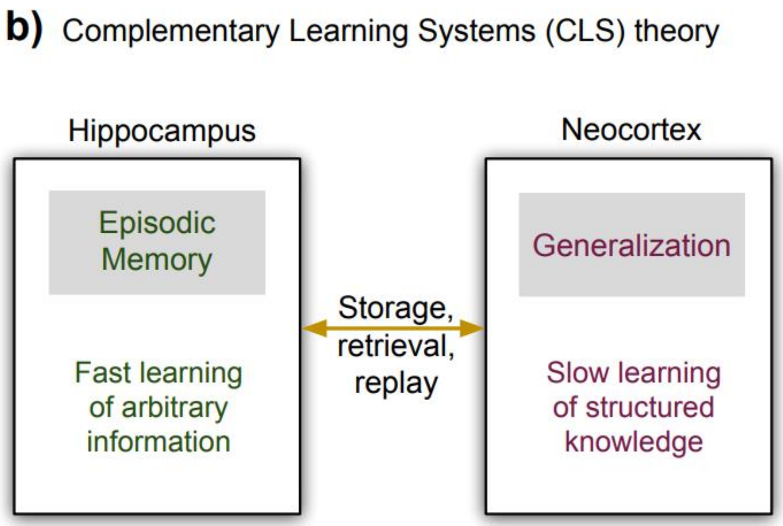
Progressive networks (Rusu et al., 2016)



Introduction & Related Works

- In paper …
- Hippocampus : more than a simple experience replay buffer
  - Reactivation of the memory traces yields rather flexible outcomes
  - better paralleled with a **generative model** than a replay buffer

  → deep **generative replay** framework



Parisi et al. (2019) Continual lifelong learning with neural networks: A review.

Introduction & Related Works

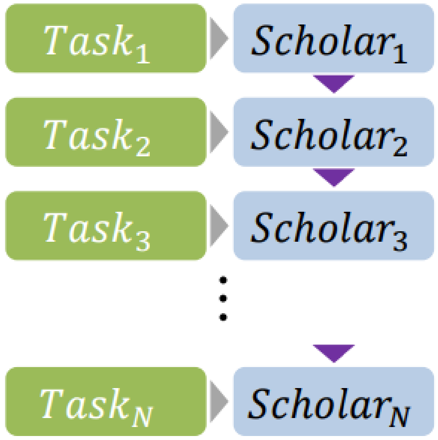
- **“Scholar”** model : **generator-solver pair**
    - Without referring to past data
    - Learn the new task without forgetting its own knowledge
        - Retains previously acquired knowledge by the concurrent replay of generated pseudo-data
        - Update the generator and solver networks : generated data & new data
    - Any deep generative model as a generator

Generative Replay - Terminology

- Sequence of task : $\boldsymbol{T} = (T_1, T_2, \cdots, T_N)$
- **Definition 1**
  - task T_i는 데이터 분포 D_i에 대해 모델을 최적화하는 것
  - D_i 의 학습용 데이터는 $(\boldsymbol{x}_i, \boldsymbol{y}_i)$로 주어진다.
- Scholar : generator-solver pair
- **Definition 2**
  - Scholar $H = < G, S >$ tuple
  - generator G : generative model that produces real-like samples
  - solver S : task solving model parameterized by θ
- Solver의 objective function $\mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})\sim D}[L(S(\boldsymbol{x};\theta),\boldsymbol{y})]$

Generative Replay

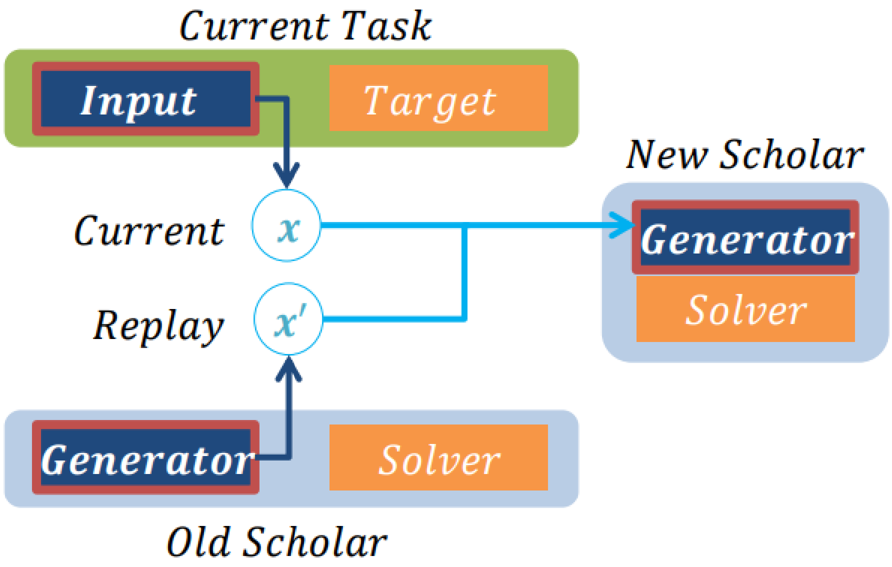
- Training a sequence of scholar models : $(H_i)_{i=1}^{N}$
  - H_n $(n > 1)$ learns T_n & knowledge of H_{n-1}



(a) Sequential Training

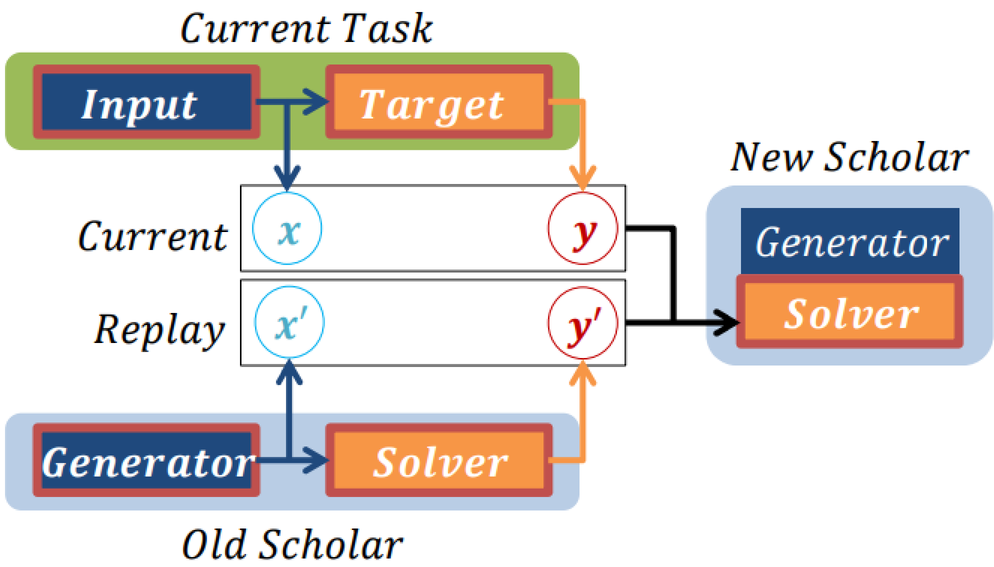
Generative Replay

- New generator input : x & x'
  - x : current task input
  - x' : old generator replayed inputs



(b) Training Generator

- New solver training : using (x, y) & (x', y')
  - y' : old solver's output to x'



(c) Training Solver

- x and x' mixed at a ratio that depends on the desired importance of a new task compared to the older tasks.

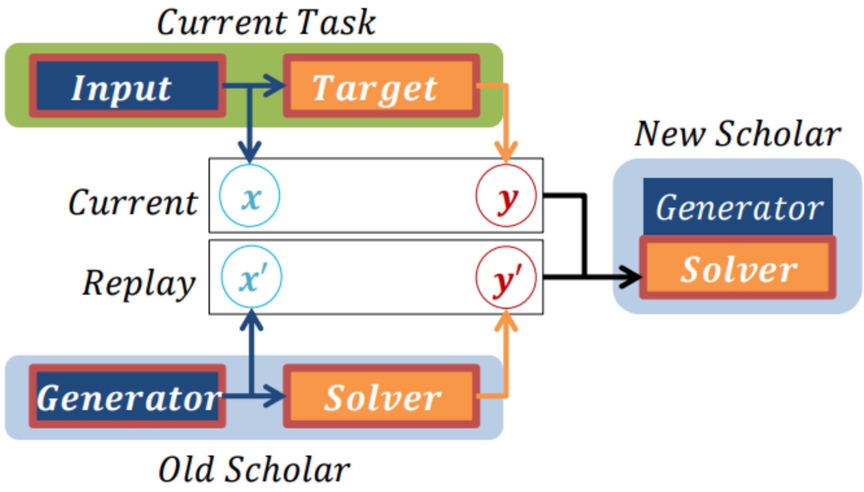
Generative Replay

$$\mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})\sim D}[L(S(\boldsymbol{x};\theta),\boldsymbol{y})]$$

$$L_{train}(\theta_i) = \underbrace{r\mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})\sim D_i}[L(S(\boldsymbol{x};\theta_i),\boldsymbol{y})]}_{\text{Current Task}} + (1-r)\underbrace{\mathbb{E}_{\boldsymbol{x}'\sim G_{i-1}}[L(S(\boldsymbol{x}';\theta_i),S(\boldsymbol{x}';\theta_{i-1}))]}_{\text{Old Scholar}}$$

$$L_{test}(\theta_i) = r\mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})\sim D_i}[L(S(\boldsymbol{x};\theta_i),\boldsymbol{y})] + (1-r)\,\mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})\sim D_{past}}[L(S(\boldsymbol{x};\theta_i),\boldsymbol{y})]$$ (D_{past} : 과거 데이터의 누적 분포)

- 첫 번째 solver는 replayed 데이터가 없기 때문에 i = 1 에서 두번째 항은 무시



(c) Training Solver

Experiments

- Generator : WGAN-GP
- Figure notation
    - GR : **Generative Replay**. (our model) assuming a situation when the **generator is perfect**
    - ER : Exact Replay. (upper bound) **replayed actual past data** paired with the predicted targets from the old solver
    - Noise : generated samples do not resemble the real distribution at all
        - **Replaying random gaussian noises** paired with recorded responses
    - None : baseline of naively trained solver network

Experiments 1

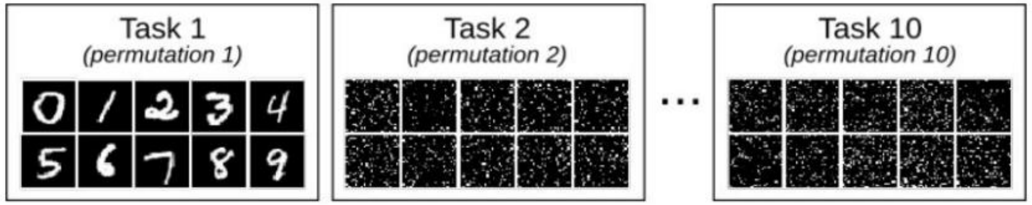
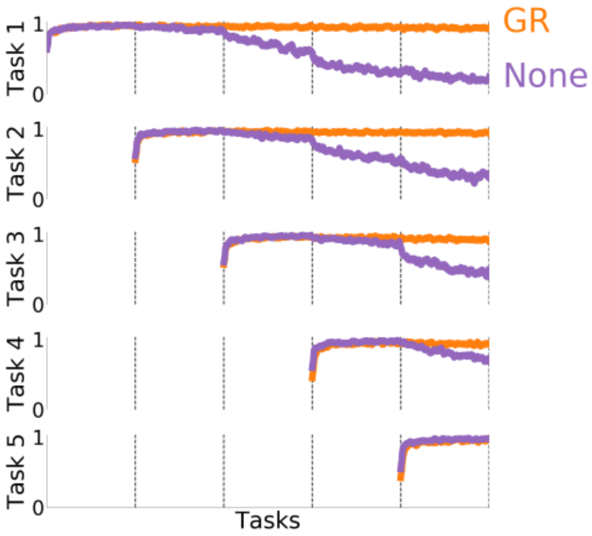


Figure 2: Schematic of the permuted MNIST task protocol.

Gido van de Ven and Andreas S. Tolias.
Three scenarios for continual learning. arXiv:1904.07734, 2019.

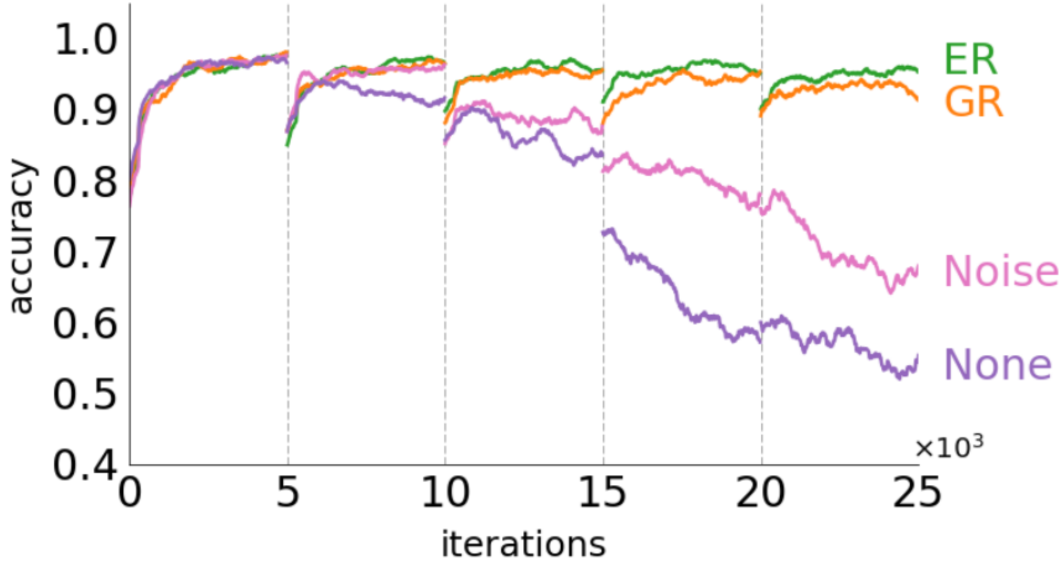
- Learning **independent tasks** : MNIST pixel permutation task
  - Solver : classify pixel permuted inputs into the original classes
  - Tasks are technically independent from each other, being a good measure of memory retention strength of a network

Test performances on each task during sequential training.



GR maintained the former task performances

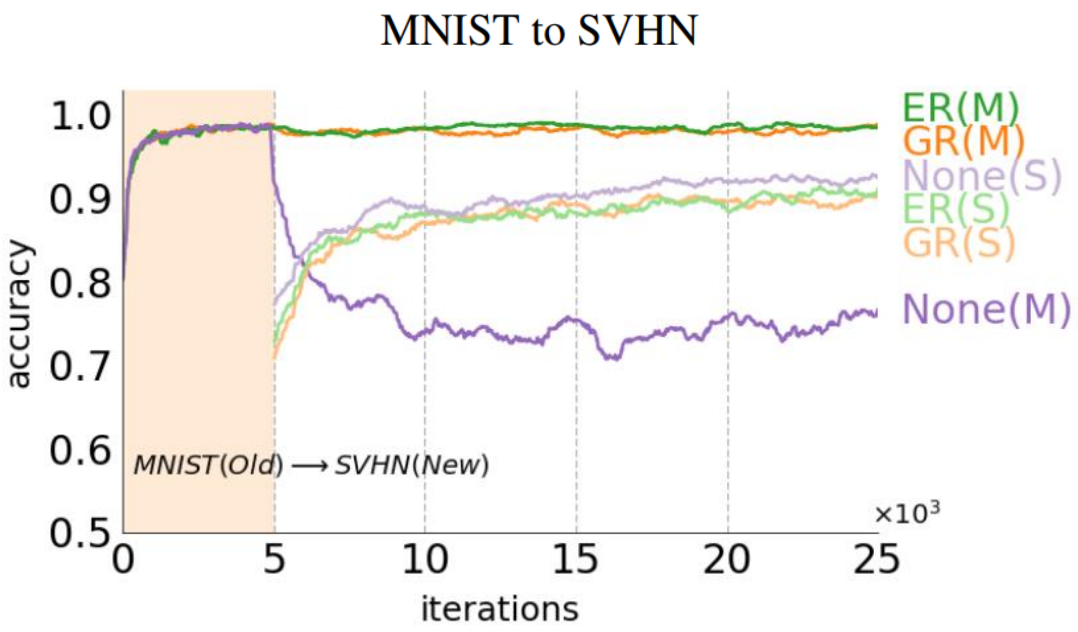
Average accuracy measured on cumulative tasks

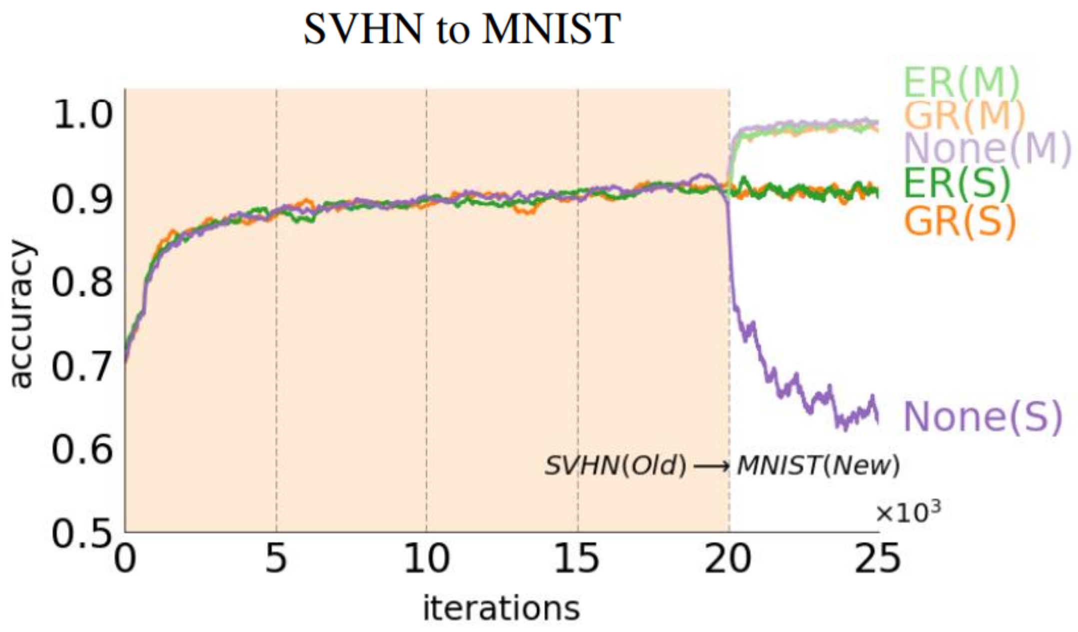


GR : almost full performance
None : catastrophic forgetting

Experiments 2

- Learning **new domains**
  - Tested if the model can incorporate the knowledge of a new domain with generative replay





GR : maintained its performance on the first task while accomplishing the second one

None slightly better performance on new task : network was solely optimized to solve the new task

Experiments 2

- Learning **new domains**

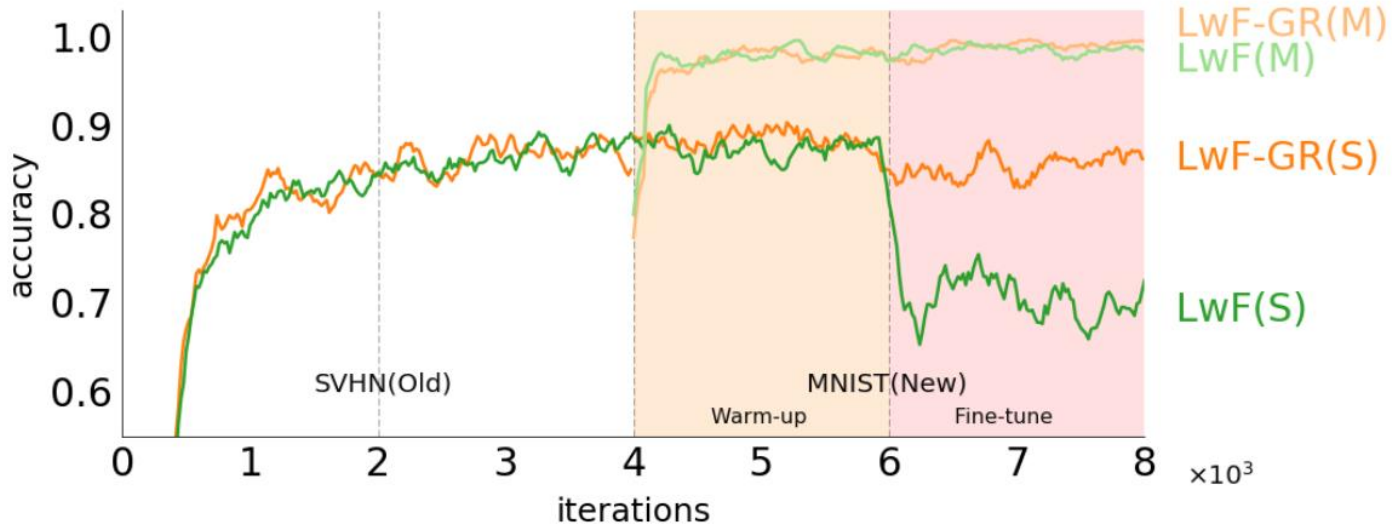
Samples from trained generator in MNIST to SVHN experiment

1000 iterations | 2000 iterations | 5000 iterations | 10000 iterations | 20000 iterations

The samples are diverted into ones that mimic either SVHN or MNIST input images

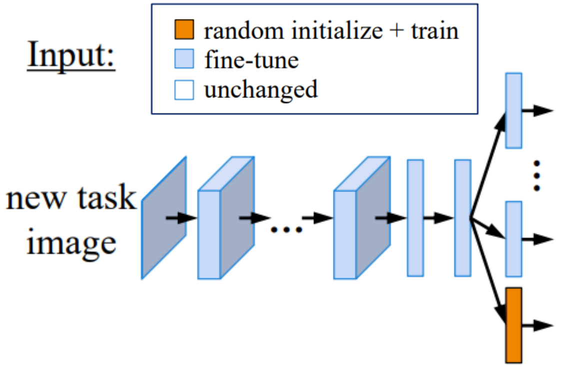
Experiments 2’

- Learning **new domains**



Generative replay is compatible with other continual learning models

LwF (Learning without Forgetting, Zhizhong Li et al., 2017)



Target:

model (a)’s response for old tasks

$$\mathcal{L}_{old}(\mathbf{y}_o, \hat{\mathbf{y}}_o) = -H(\mathbf{y}'_o, \hat{\mathbf{y}}'_o) = -\sum_{i=1}^{l} y'^{(i)}_o \log \hat{y}'^{(i)}_o$$

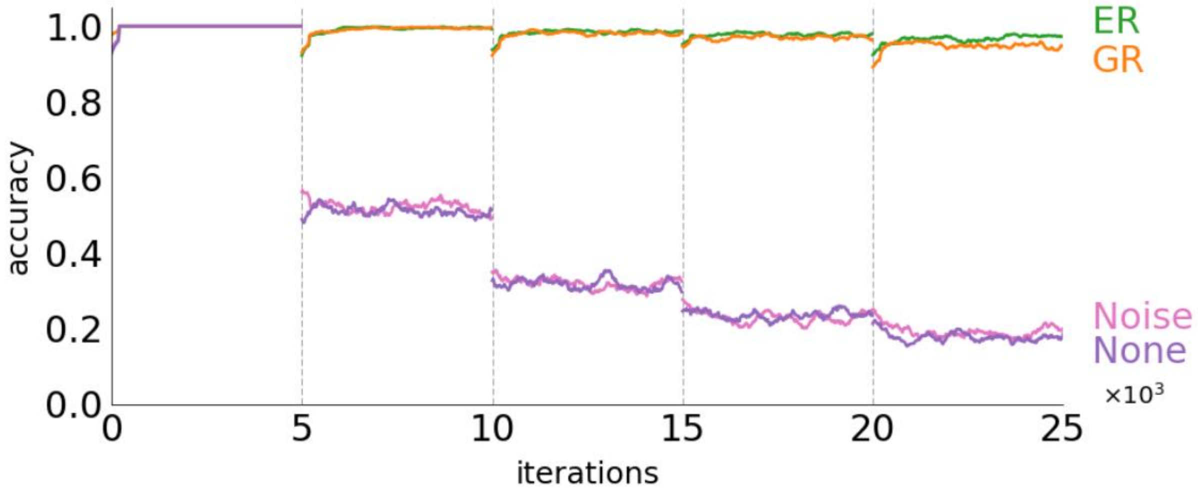
new task ground truth

$$\mathcal{L}_{new}(\mathbf{y}_n, \hat{\mathbf{y}}_n) = -\mathbf{y}_n \cdot \log \hat{\mathbf{y}}_n$$

Experiments 3

- Learning **new classes**

Sequentially trained on 5 tasks
each task : MNIST images belong to 2 out of 10 labels

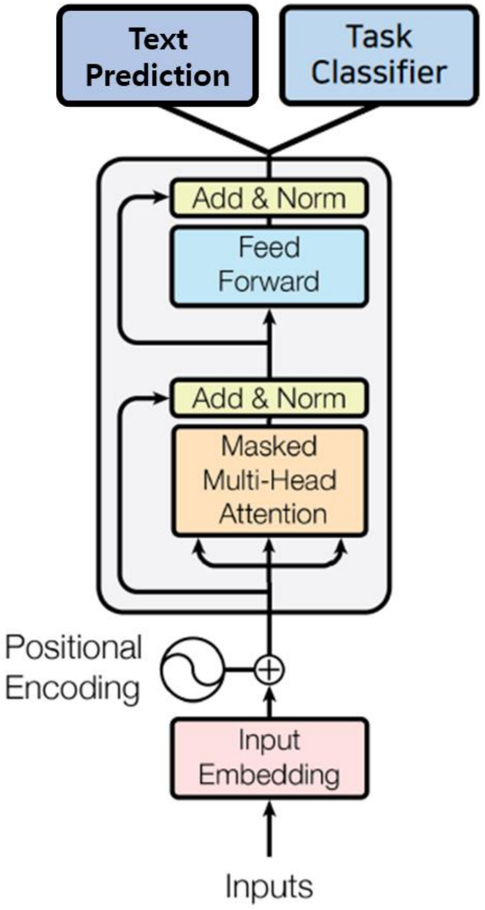


Discussion

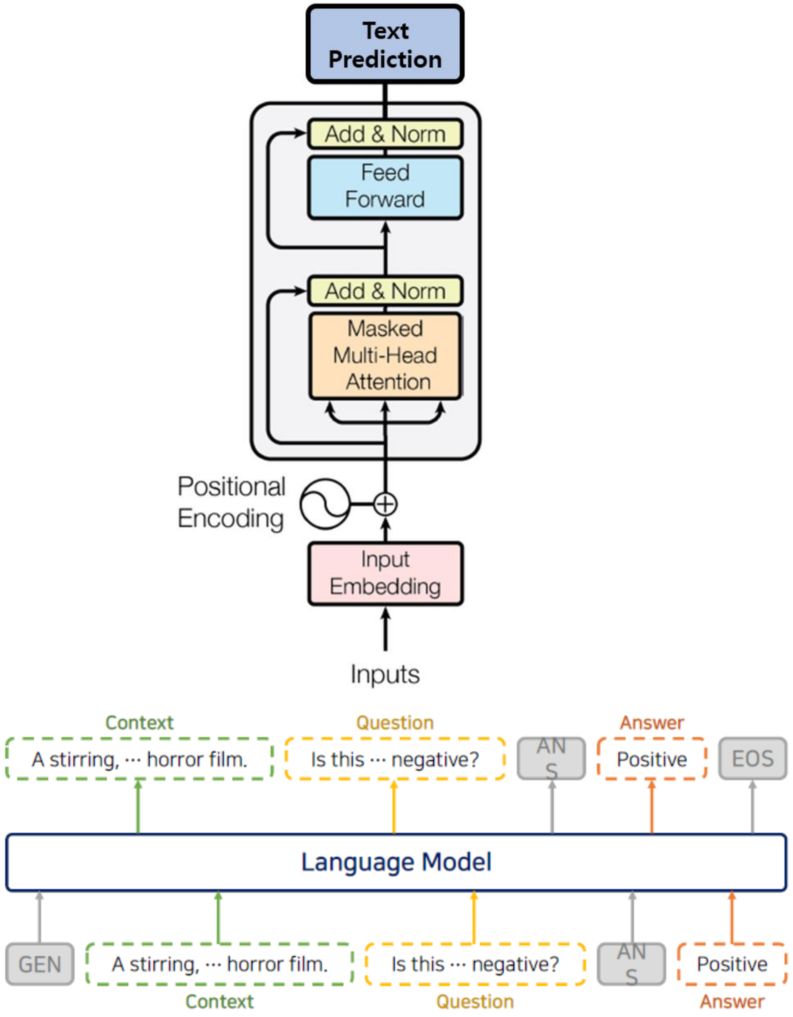
- Deep generative replay framework
  - Scholar model : generator and solver
- Benefit
  - Maintains the former knowledge : input-target pairs produced from the saved networks (not old data)
  - Ease of balancing the former and new task performances
  - Flexible knowledge transfer
- Defect
  - Heavily depends on the quality of the generator

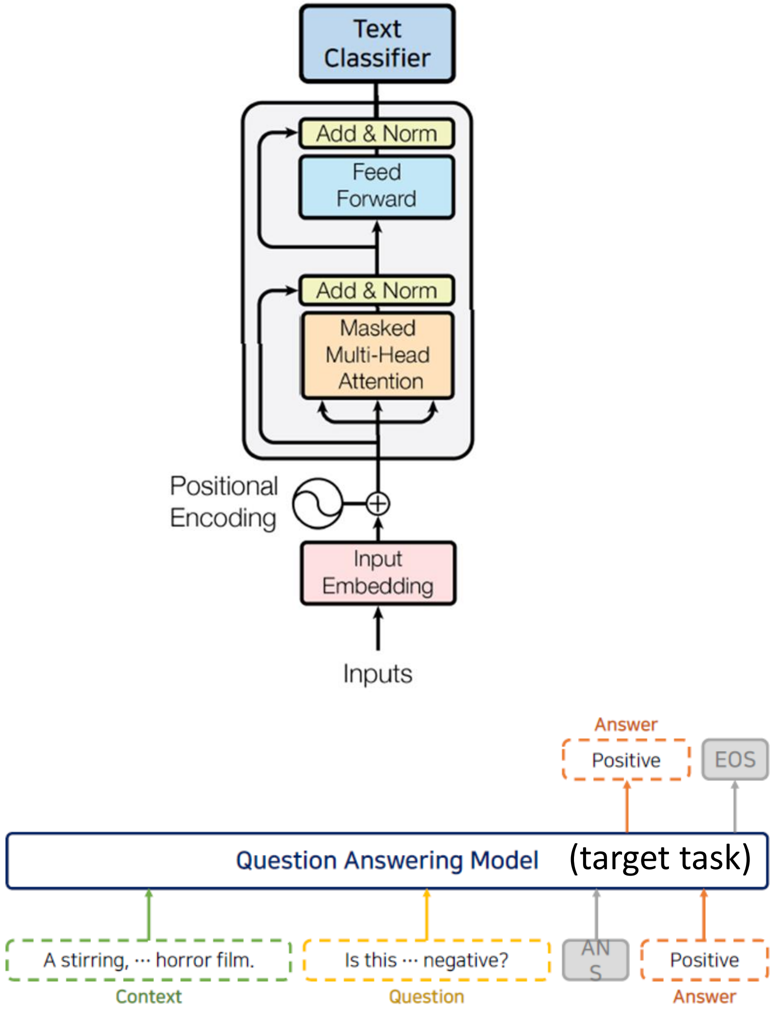
Moreover : language

- LAMOL (Lifelong Language Learning with Effective Generative Replay)



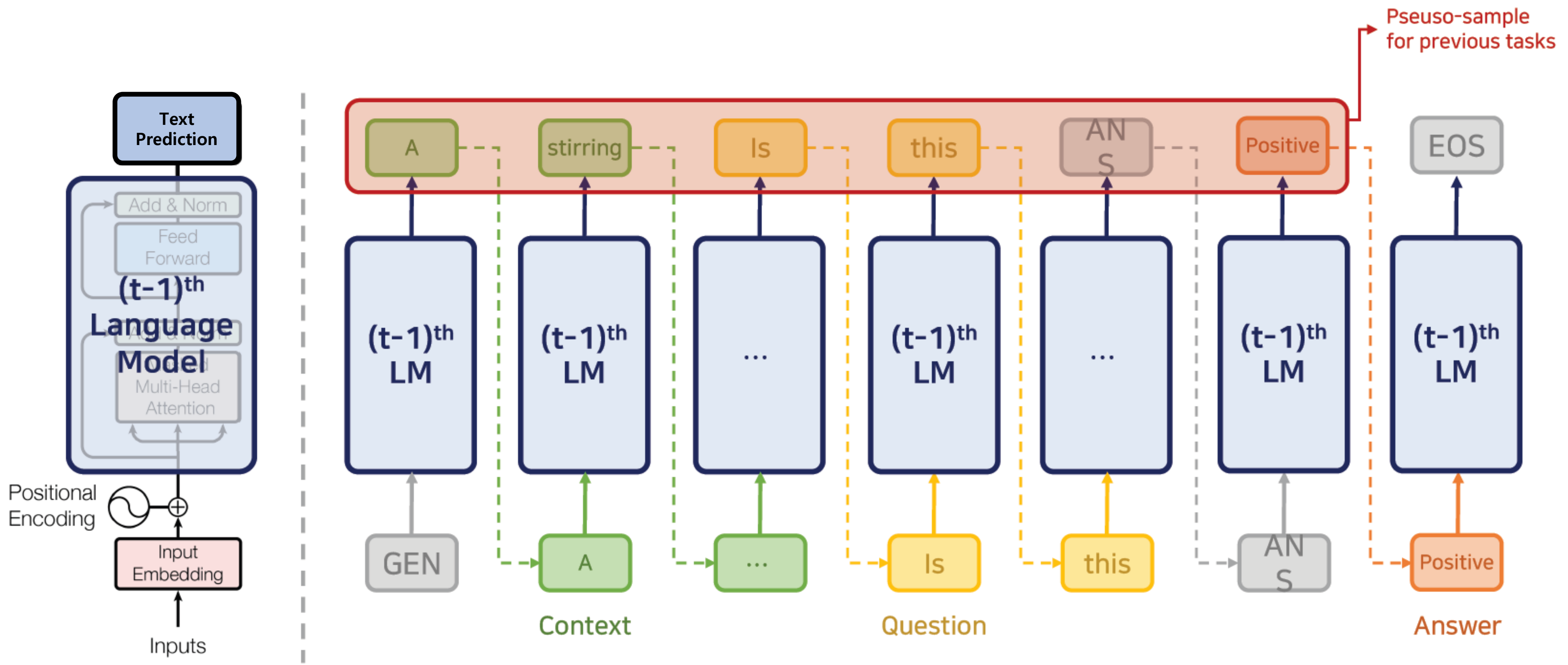
LAMOL





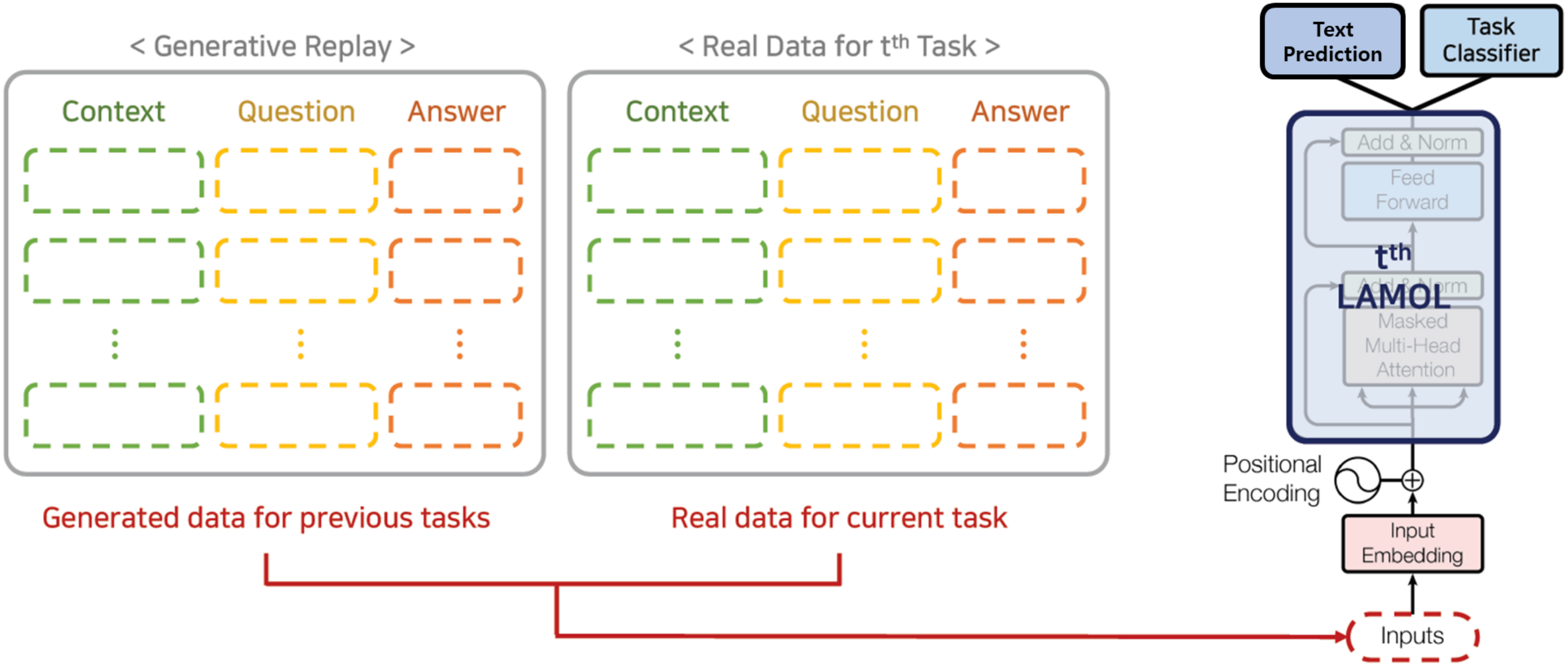
Moreover : language

- LAMOL
- Text Prediction model = 일종의 generative model
  - 새로운 task를 학습하기 전 이전 tasks의 데이터 분포를 따르는 샘플 생성



Moreover : language

- LAMOL



END