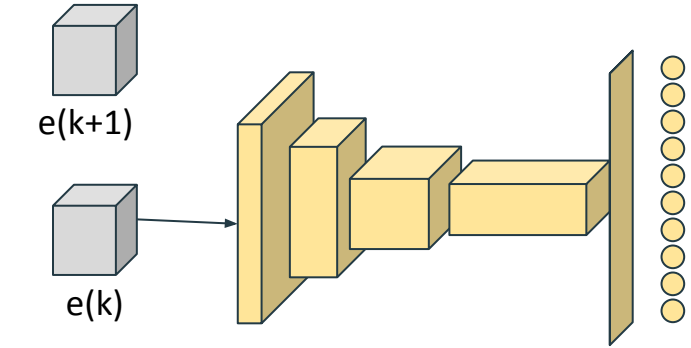
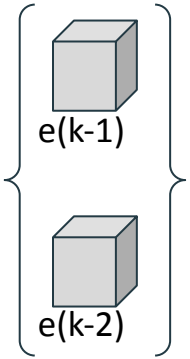
Negative examples for continual learning

Gabriele Graffieti

Biolab reading group - March 12, 2021

Continual Learning





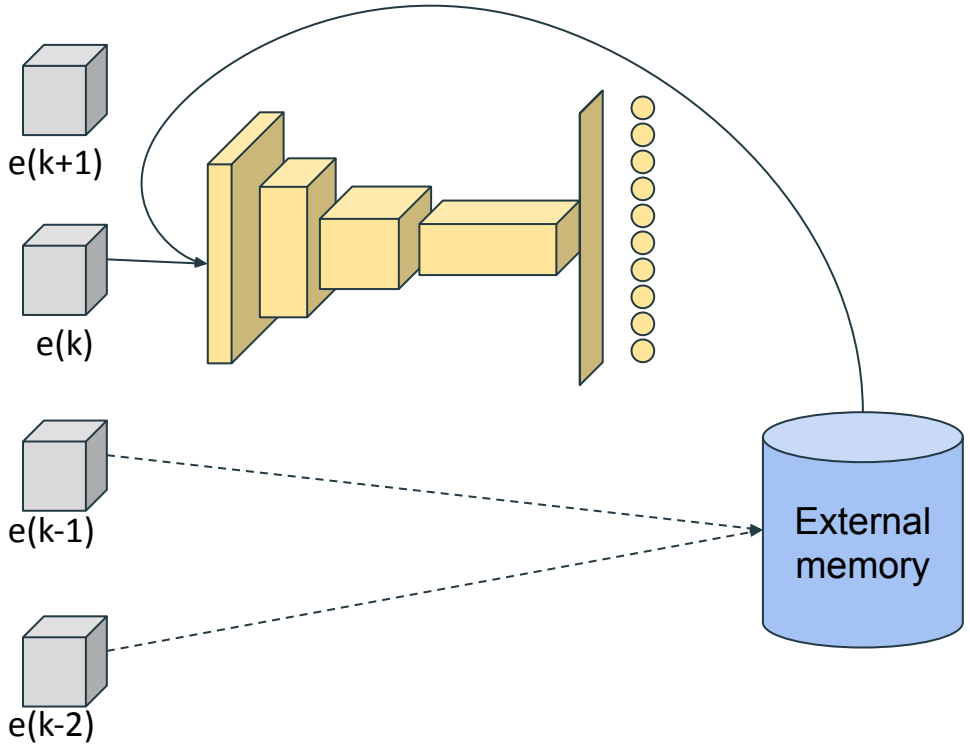
Past data (not accessible anymore)

Problems:

- Model trained **only** on the current experience!
- SGD is a greedy algorithm, it does not take into account past experience.
- Only goal of SGD is to optimize the parameters of the network on the data currently available

-> Catastrophic forgetting!

Replay



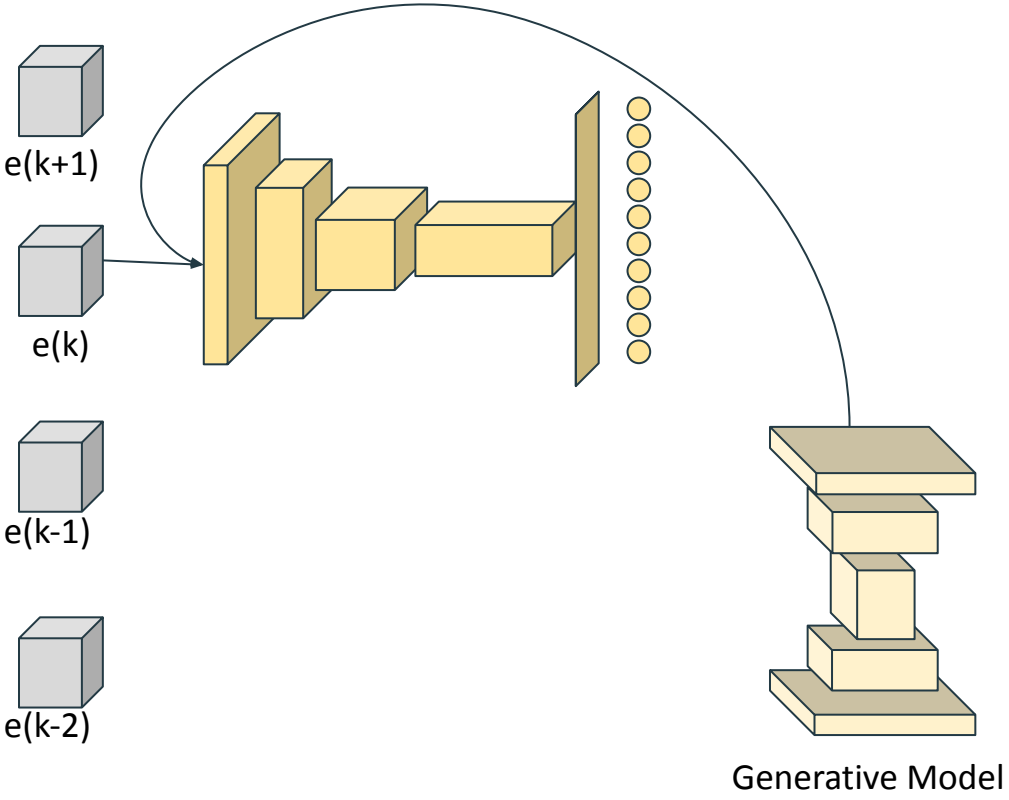
Pro:

- Catastrophic forgetting highly reduced.
- Simple and easy to implement strategy.
- Memory is cheap and abundant.

Cons:

- Memory is not infinite (the stream of experience can be).
- What about privacy and private data?
- Not biologically plausible.

Generative replay



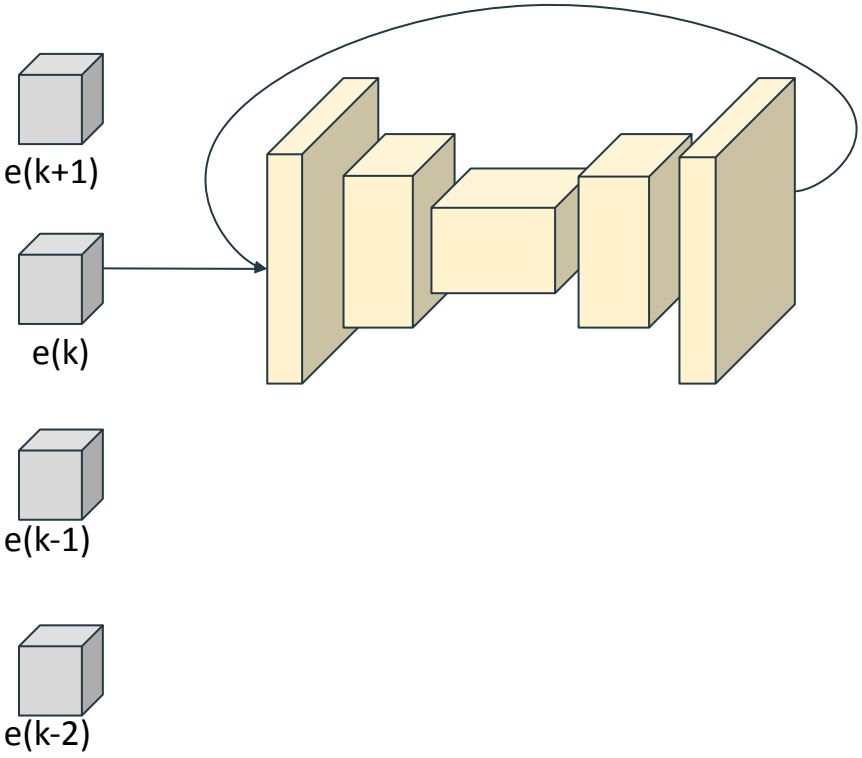
Pro:

- No replay memory is needed.
- More biologically plausible.
- Can also generate unseen or new data that is totally plausible.

Cons:

- How to train the generative model??
  - The problem is now on the continual training of the generator instead of the classificator.

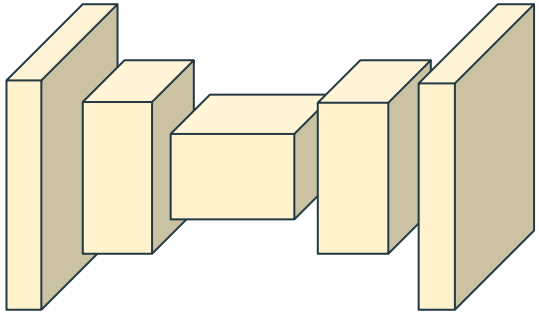
Generative replay



- We can use replay also for the generator?
- In this case is a sort of auto-replay.
- When trained on experience k, the generator generates data for all the past experience and it is trained on the union of generated and current data.

-> Photocopy problem!

The photocopy problem (generation loss)



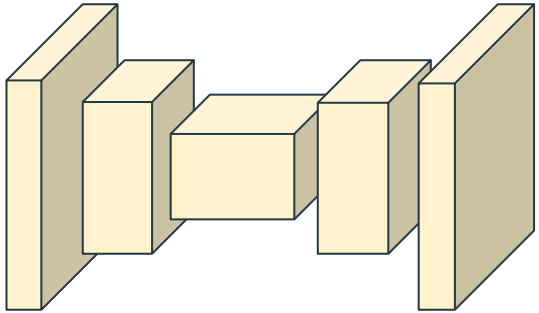
Original image

Step 1

Generated image

The photocopy problem (generation loss)

Generated image after step 1

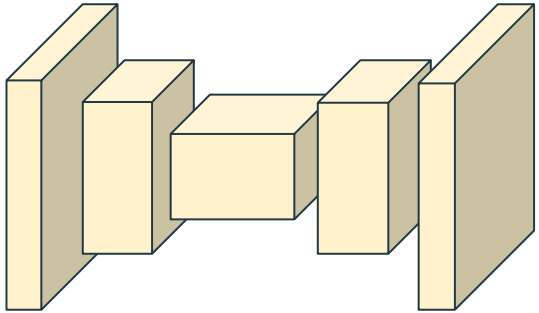


Step 2

Generated image

The photocopy problem (generation loss)

Generated image after step 2

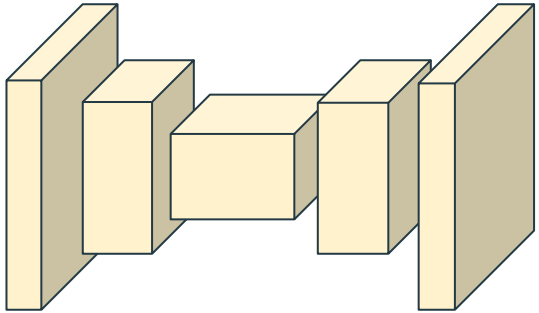


Step 3

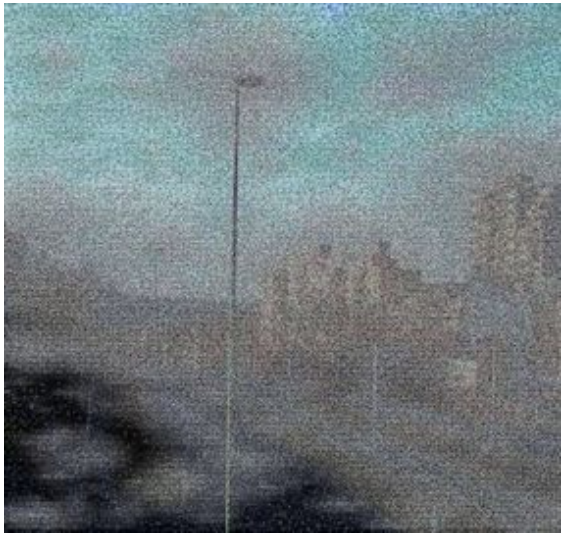
Generated image

The photocopy problem (generation loss)

Generated image after step 3



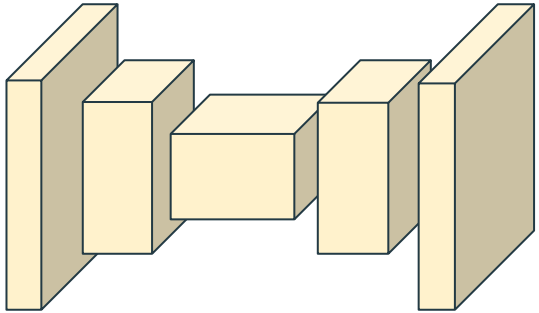
Step 4



Generated image

The photocopy problem (generation loss)

Generated image after step 4



Step 5

Generated image

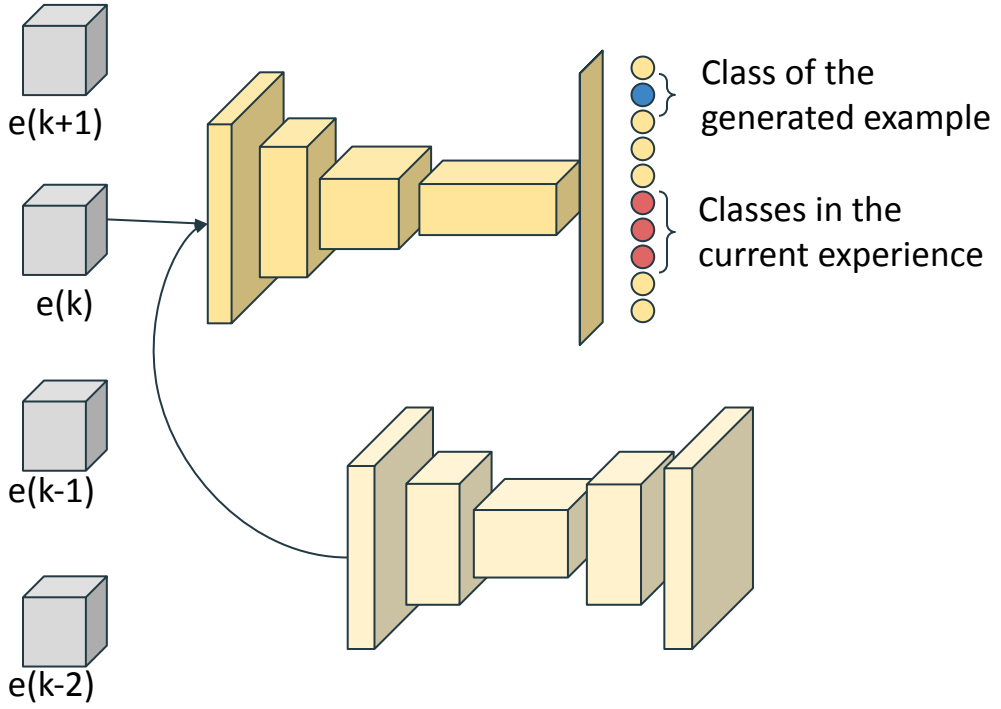
The photocopy problem (generation loss)

After only 5 experiences, we train our classifier with images like this one.
This data is totally different from the original and the test data, so it’s no surprise that the classifier performs poorly.

But why?

We use this data as “positive” example, so basically we are saying to the network: “this is an image of class c, if you misclassify it you will be penalized”. But this image is not an image of class c! In fact it does not belong to any class!

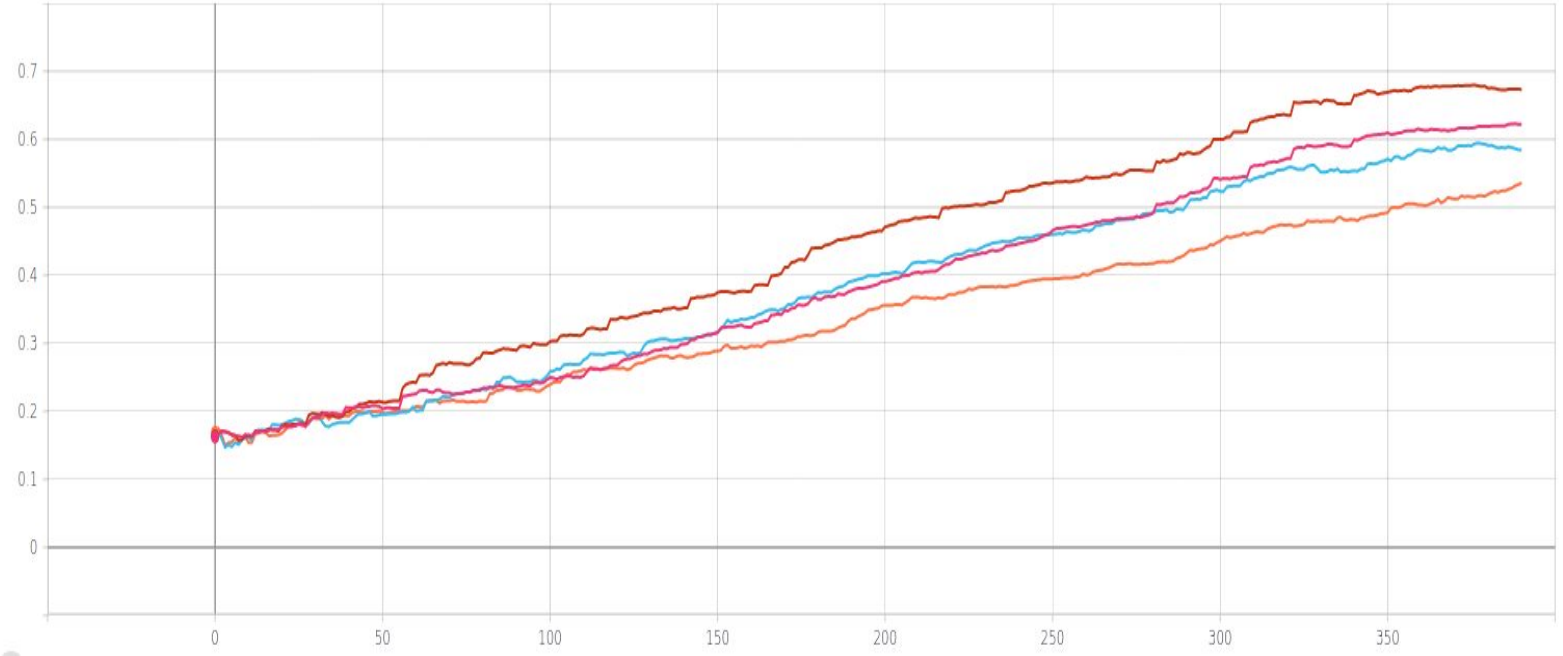
Negative examples



Idea:

- Use the generated example only as negative examples for the classes in the current experience.
  - Avoid classification shortcuts especially with few (or only one) classes in the experience.
- Do not update the weight related to the classes of the generated patterns.
  - We cannot increase the performance on a class using generated data.

Negative examples



Problems and questions

- Using negative examples is effective, but it is a “working trick”. The real problem is still here and we need to fix to gain substantial advancement in the field.
- From preliminary tests, negative examples seems to work mainly where the experience contains few classes, since the effect of “out-of-distribution” examples is amplified.
- Why use generated data? Random data work as well, why we need a generator? Could we use data from different but similar dataset as negative examples in the same way we use random data?
- Continual training of generative models is still an open issue, not only related to replay or classification problems.
- Are negative examples biologically plausible? Are few classes experience biologically plausible? In the real world there are examples of few classes learning?

Thank you!

Questions?