Machine Learning – 6/12/2018

Continual Learning

Lorenzo Brigato

Lab Ro.Co.Co - DIAG

SAPIENZA
UNIVERSITÀ DI ROMA

Supervised Machine Learning

- Current Machine Learning paradigm
  - Fixed labeled data set: $\{\boldsymbol{x}_i, \boldsymbol{y}_i{}^*\}$, $\quad i = 1, \ldots P$
  - Samples x_i may come from different domains (e.g. images, words, stock prices …)
  - Divide the samples in **Training** and **Test** set

Goal

Find a function $\boldsymbol{f}$ represented by the weights $\boldsymbol{w}$ tuned on the **Training** set:

$$\underset{\text{sample}}{\boxed{\boldsymbol{x}_i}} \longrightarrow \underset{\text{function}}{\boxed{\boldsymbol{f}(\cdot\,,\boldsymbol{w})}} \longrightarrow \underset{\text{predicted label}}{\boxed{\boldsymbol{y}_i}} = \underset{\text{target label}}{\boxed{\boldsymbol{y}_i{}^*}}$$

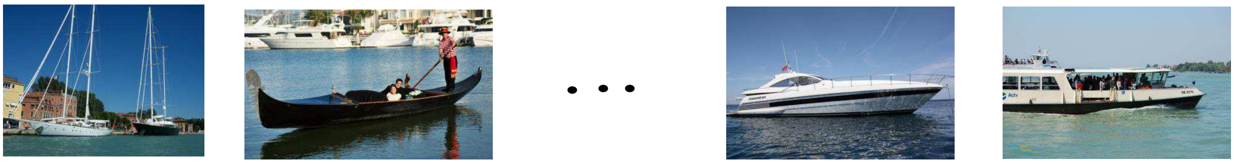
- **Very successful!!!**

Facebook, Google, Twitter →  → High perfomance and generalization capability

Supervised Machine Learning

- Successful even with our limited resources (MarDCT dataset run on a laptop)

Training phase



$$\underbrace{\qquad\cdots\qquad}_{\boxed{f_1(\cdot, w_1)}}$$

Testing phase

$\rightarrow \boxed{f_1(\cdot, w_1)} \rightarrow$

- Speed boat
- Sail boat
- Gondola
- Ferry boat

Supervised Machine Learning

- What happens if we do not have the entire dataset available?
  - Expand an existing model (e.g. add a new class)

Training phase

Update the model with the new class **without** using the **entire dataset**

$\rightarrow \boxed{f_1(\cdot, w_1)} = \boxed{f_2(\cdot, w_2)}$

Testing phase

$\} \rightarrow \boxed{f_2(\cdot, w_2)} \rightarrow$

- Raft
- Speed boat
- Sail boat
- Gondola
- Ferry boat

Supervised Machine Learning

- Stress the idea with a continous data set building
  - Repeat the training on a new class for K times

Training phase

$\rightarrow \boxed{f_2(\cdot, w_2)} = \boxed{f_3(\cdot, w_3)} \cdots$

$\rightarrow \boxed{f_{K-1}(\cdot, w_{K-1})} = \boxed{f_K(\cdot, w_K)}$

Testing phase

$\} \longrightarrow \boxed{f_K(\cdot, w_K)} \longrightarrow$

Supervised Machine Learning

- Assume $K = 4$, the Jet Sky is the last learned class
- You should expect the following result:

$\} \rightarrow \boxed{f_K(\cdot, w_K)} \rightarrow$

- Cruise ship
- Raft
- Speed boat
- Jet sky
- Gondola
- Ferry boat
- Sail boat

Chatastropchic forgetting

- Why do we witness a decrease of perfomance?
  - Every time you retrain the model you change the weight distribution
  - New weights $\longrightarrow$ Chatastropchic forgetting of older functions

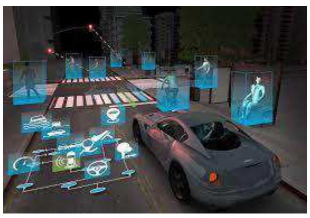
$$\boxed{f_1(x_1, w_1)} \neq \boxed{f_2(x_1, w_2)}$$

- **Short-term pratical Problem:**
  - Every class addition $\longrightarrow$ training with the **entire** dataset
    - It is not always available (mobile application with memory constraints)
    - It is computationally inefficient (linearly increase with the number of tasks)
- **Long-term Problem:**
  - Impossibility to build truly intelligent ML models

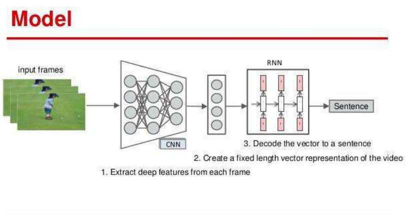
State of the art

- Hot Research topic
  - Continual Learning is gaining more and more attention in the scientific community
  - Diverse Projects have recently started:

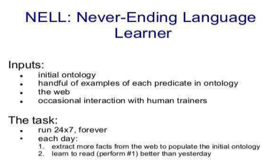
Lifelong Learning Machine by DARPA



Lifelong Learning of Visual Scene Understanding at Institute of Science, Austria



Never-Ending Language Learner at Carnegie Mellon University


NELL: Never-Ending Language Learner

Inputs:
- initial ontology
- handful of examples of each predicate in ontology
- the web
- occasional interaction with human trainers

The task:
- run 24x7, forever
- each day:
  1. extract more facts from the web to populate the initial ontology
  2. learn to read (perform #1) better than yesterday

Spatio-Temporal Representation and Activities for Cognitive Control in Long-Term Scenarios

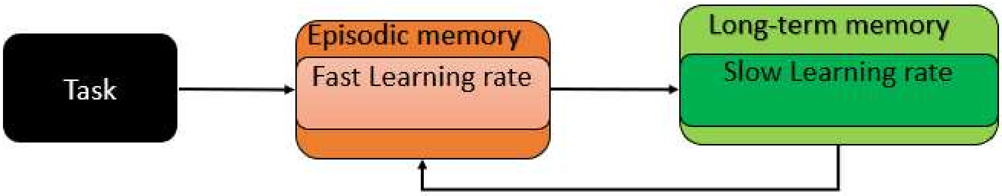
STRANDS
Spatio-Temporal Representation and Activities for Cognitive Control in Long-Term Scenarios

State of the art

- Neuroscientific background to tackle the problem
  - Biologically inspired theories in the 90's (survey by French et. al 1997)
  - Two distinct but interacting functional areas (Hyppocampus and Neocortex)
  - Dynamical separation of internal representations during learning

Complementary Learning Systems Theory

- Protection of older memories from the interference of novel tasks



State of the art

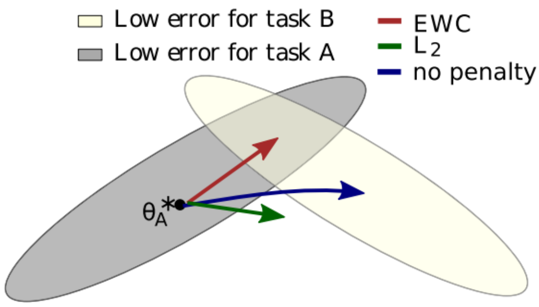
- ML approaches to mitigate catastrophic forgetting:
  - **Regularized**
    - Penalizing changes of network parameters
    - **Relatively easy to implement**
    - **Performance trade-off among tasks**
    - E.g. Elastic weight consolidation - (Kirkpatricka et. al 2017)
  - **Dynamic Architectures**
    - **Partially solve the problem**
    - **Scalability issues**
    - E.g. Progressive networks - (Rusu et. al 2016), Dynamically expandable networks - (Yoon et. al 2018)
  - **Dual-memory systems**
    - Inspired by CLS theory to different extents
    - **Most performing arcthitectures**
    - **Highest overall complexity**
    - E.g. Deep Generative Replay - (Shin et. al 2017)

Regularized Approach

- **Elastic weight consolidation** (Kirkpatricka et. al 2017)
  - Preservation of relevant parameters ($\theta = w$)
  - Old task is represented by θ_A and the new task by θ_B
  - Minimize the following loss function:

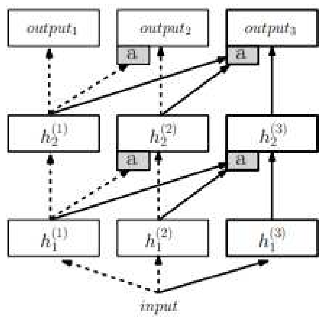
$$\mathcal{L}(\theta) = \mathcal{L}_B(\theta) + \sum_i \frac{\lambda}{2} F_i (\theta_i - \theta^*_{A,i})^2$$

  - With F_i coming from the Fisher information matrix (equivalent to the second derivative of the loss near a minimum)

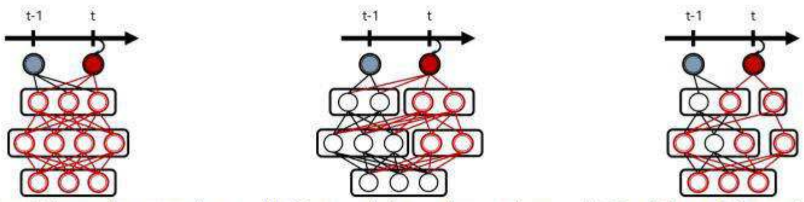


Dynamic Architectures

- **Progressive networks** (Rusu et. al 2016)
- Add parallel layers for successive tasks fixing previous weights
  - Mitigates catastrophic forgetting and enables features transfer
- Tested in Reinforcement Learning domain (only computer games)



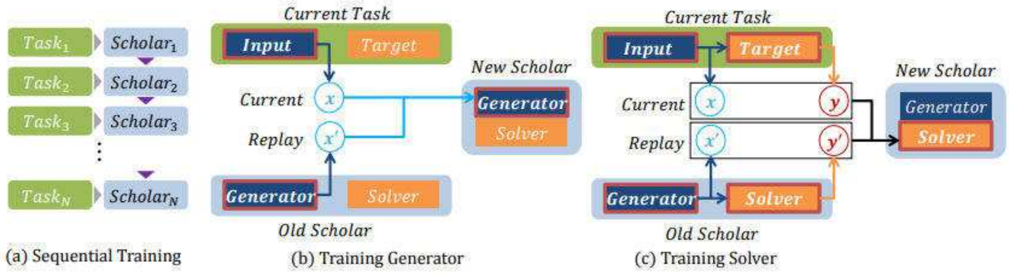
- **Dynamically expandable networks** (Yoon et. al 2018)
- Fuse the two previous approaches:
  - Retrain the parameters which are «less» relevant
  - Add new resources when the network capacity is reached



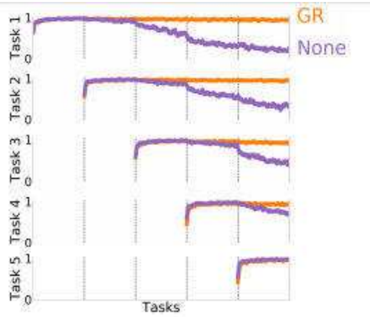
(a) Retraining w/o expansion (b) No-retraining w/ expansion (c) Partial retraining w/ expansion

Dual-memory systems

- **Deep Generative Replay** (Shin et. al 2017)
  - Generator model used to replay past experience to a solver
  - Accuracy highly depends on how well the generator reproduces old samples



- Sequential Training
  - Very high performance if you reply old samples (orange curve)
  - Accuracy drop for each task if nothing is recalled (purple curve)

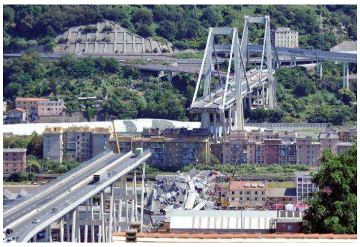


Continual News Filtering

- **Problem modeling**
  - Train a classifier able to recognize if a text corpus is relevant for the police
  - $\{\boldsymbol{x}_i, \boldsymbol{y}_i{}^*\}$ with $\boldsymbol{x}_i$ that is a vector of words and $\boldsymbol{y}_i{}^*$ is the class (+1 relevant, -1 not relevant)

- Very unpredictable and not constant behaviour:
  - Some news might be very critical for few weeks (easy to predict)
  - But what if they appear again after months? (Morandi bridge investigation)
  - The model has probably forgot about them!!

- Difficulty for linguistic correlations (both are «Morandi»):



Continual News Filtering

- **Use a Continual Learning approach**
  - Example: Recall past samples using Deep Generative Replay
  - Retrain the network with batches of news every some time. In total K steps:
    - K can be very large since news are arriving every day!