Literature Notes: Continual Learning

Julia Boehlke

August 27, 2021

1 Requirements of CL Literature

- Avoid forgetting (*)
- Fixed memory and compute
- Enable forward transfer
- Enable backward transfer (*)
- Do not store examples

2 Survey Papers / Meta Papers

GDumb: A Simple Approachthat Questions Our Progressin Continual Learning [13]

- GDumb = Greedy Smapler and Dumb Learner (class balanced fixed memory buffer, retrained from scratch using samples in buffer)
- Simplifying Assumptions in CL

    1. Disjoint Task Formulation: at a particular duration in time datastream will provide samples specific to one task. Sometimes this assumption also entails that there is only *one* specific time, where data for a specific task is streamed. This means there is no backward transfer.
    2. Task-Incremental(TI_CL) : along with the disjoint task assumption, the task information (or id) is also passed during training and inference (multi-head). In Class-incremental continual learning (CI-CL) no such task information is given.

3. Online CL: restricting the learner to use each sampel only once to update parameters (unless stored in buffer). In offline CL there is unrestricted access to entire current dataset for training multiple epochs.

- Online CL preferable in situations with fast spitting data stream.
- Found GDumb outperforms most methods by large margin
- Table 1 gives a great overview/categorization of methods and assumptions
- non of the reviewd papers seem to match our assumptions/requirements exactly: not dijoint, class-incremental, offline.

CVPR 2020 Continual Learning in Computer Vision Competition: Approaches, Results, Current Challenges and Future Directions [8]

- CVPR Continula Learning challenge on CORe50 dataset including three different tasks: New Instances (8 batches of all classes, i.e., focus on backward transfer); Multi-Task New Classes (multi-head); New Instances and Classes (batches containing examples of single class may contain previously seen or new classes, i.e., disjoint setting focused on improving on seed classes with single-head classification)
- evaluate on a weighted sum of scores on accuracy, Disk usage, RAM, time
- baseline include naive fine-tuning, rehersal with growing memory (20 images of each batch stored), and ARI* with latent replay [11] (described below)
- winning team uses replay method for NIC and devided network outputs by prior probability for each class to handle class imbalance (Buda et al. 2018)
- top-4 solutions employ rehersal-based technique
- on NI challenge, UT_LG Team rehersal training with batch instead of mini-batch level (for every epoch one memory batch and current new batch is concatenated) and introduce review step (with lower learning rate) before testing, where only memory data is used.
- code available for all submissions

3 Papers

3.1 Rehersal Based Mehthods

Rehersal Methods allow at least some data to be stored and used to *reherse* previously learned knowledge. This is also known as Expereience Replay (ER). When no storage of data is possible, rehersal is often perforemed using generated images ([15]), where previousy learned knowledge is stored indirectly.

Online Continual Learning with Maximally Interfered Retrieval (MIR) [1]

- CI-CL, online, disjoint, rehersal based approach.
- Sample criterion for controlled selection (from rehersal) of samples from buffer where predictions will be most negatively impacted by forseen parameter update. Their Research question: what samples should be replayed from the previous history when new samples are received
- most negatively impacted = loss changes most, when updating on new data (estimated for subset of buffer data).
- also applicable to generative replay approaches
- in a reltivel small number of total classes/task case (MNIST SPLIT) their approach (ER+MIR) is significantly better (87,6%) than random sampling ER (82.1%). In other scenarios, their approach outperforms with a smaller margin.
- (I dont really understand, why they restrict themselves to a disjoint setting, thos should work in non-disjoint situation.)

Gradient based sample selection for online continuallearning [2]

- CI-CL, online, non-disjoint expand GEM approach to situation where task boundries are not available
- formulate replay buffer population problem as constrained minimization of the solid angle. Use a surrogate objective, which maximizes diversity of samples using the parameter gradients of samples instead of feature representations

- indirectly adress the issue of class imbalance
- reevaluate replay buffer once a so called *recent* buffer is full
- also propose cheap alternative greedy sample selection for large buffers (removes overhead of gradient computation for all samples solving constrained optimization). Idea: compute score based on max. coisne simitlarity of current sample gradients with randomly selected subset of buffer gradients. When new sample arrives, compute its score and randomly select candidate for replecement (probability of normalized scores) and compare scores to decide. Replace constrained optimization when buffer is large with soft regularization equivalent to rehersal.
- Experiments performed on low resolution datasets such as MNIST and CIFAR10
- compared with random or clustering-based, buffer population methods and reservoir population methods, their approached show merit, especially the greedy approach using rehersal instead of constrained optimization.
- code availble

Random Sampling with a Reservoir

- 1985 algorithm designed to uniformly sample from a stream of data where the total number of elements the stream will entail is unknown.
- This algorithm could be used to continuously update a fixed size buffer with samples from a stream while ensuring, that at the end, when the stream is done, that every sample has a probability of 1/(totel stream) of being in the buffer.

More Is Better: An Analysis of InstanceQuantity/Quality Trade-off in Rehearsal-basedContinual Learning [12]

- evaluated for class-incremental setting, CI-CL, disjoint
- state that rehersal based methods are 'emerging as the most effective methodology to tackle CL' and refer to [5] for theoretical justification (optimal CL would require perfect memory)

- investigate several dimensionality reductions (deep encoders, variational autoencoders, random projections). They compare their methods to GDumb, Greedy sampler and Dumb learner, which does not use any clever selection strategy for buffer or training approach.
- evaluated on final accuracy with several datasets(MNIST, CIFAR, ImageNet, Core50). Given a fixed memory size different numbers of samples can be stored when using different parameters for reduction. (peak performance achieved when storing 8x8 pixel images to fill memory)
- only performed experiemtns for disjoint setting, i.e., where datastream shows one task once during training.
- code available

Latent Replay for Real-Time Continual Learning [11]

- store representations from some intermediate layer in the network instead of images in inputspace to reduce memory requirement. To keep representations valid, they propose slowed-down learning for the layers below the latent replay layer.
- 'a robot should be able to incrementally improve its object recognition capabilities while being exposed to newinstances of both known and completely new classes (de-noted as NIC setting - New Instances and Classes)'
- this paper aims at imporving overall accuracy for the non-rehersal based methods such as AR1 and CWR [10] (described below)

3.2 Kowledge Distillation

This category is based on the disitllation loss. Basically, the output of old samples of the model becomes the new desired output when new data is available for updating. Especially in a multi-task/ multi-head scenario, the logits on heads for previously seen data shoul not change much when a new head is learned. The most famous, original introduction of distillation loss in continual learning was made by [6], which does not enable any backward transfer of knowledge and required task knowledge at inference.

iCaRL: Incremental Classifier and Representation Learning [14]

- CI-CL, offline, disjoint and assumes that samples from each task (a batch of classes) are only present at one point in time of the data stream.
- assumes, there is a fixed size memory available to store examples from previous classes
- use nearest-mean-of-examples classifier (using representatiosn) for inference. At training time, the sample memory buffer and model parameters are updated. When samples for a new class is available, a new training batch is constructed from the new and stored data. The output of the current network for all stored images of previous classes are stored since they are needed for the distillation loss. The model is updated with the cross-entropy loss for samples from the new class while the model is encouraged to reproduce the previously stored outputs (disitllation loss) for the old samples.
- when new classes are introduced and weights are added to the network, some sampels in buffer are dropped to make room for samples from new class. The set of examples for each class is selected based on the current class mean of the feature vectors.
- Evalutaed using CIFAR100 and ImageNet datastes showing impressive results compared to previous methods for the disjoint task formulation
- (Whil I think the idea of using the distillation loss for previously stored samples could be applicable in a non-disjoint task set formulation. The distillation loss is designed to preserve previosly infered knowledge in a model and allow forward transfer. In our situation backward transfer is one of the most important requirements, which the distillation loss is not designed for. I dont think it would be wise in our scenario to penelize model outputs changing for previously seen data since that might be necessary to improve the classification boundries.)

3.3 Regularization Approaches

The basic idea behind regularization based approaches is to penelize a model for changing *too much* with newly seen and finding a sensible trade-off between plasticity and stability of the network over time. Most influential in

this category is the Elasitc Weight Consolidation Approach proposed by [4]. Each parameters importance for classification of previous task is estimated using the Fisher Information (related to curvature of loss function). Updates to *important* parameters are penelized proportionaly in the loss function when new tasks are learned. This approach is designed for task-incremental learning and does not allow backward transfer of knowledge.

Riemannian Walk for Incremental Learning:Understanding Forgetting and Intransigence [3]

- CI-CL, offline, disjoint
- RWalk is a generalization of EWC to CI setting. They use KL-divergence based regularization over conditional likelihood p(y—x) and a parameter importance score based on the sensitivity of the loss over the movement on the Riemannian manifold (induced by Fischer information) to mitigate catastrophic forgetting. By accumaulating parameter importance over the entire training trajectory, their approach allows class incremental learning.
- define task-wise measures for Forgetting(: diff between maximum knowledge and current knowledge) and Intransigence, the inability of a netowrk to learn new tasks (:diff btw model trained on entire data and incrementally learned model trainied up to specific task).
- they show that for small number of samples their approach has much greater impact than when large datasets are availbale
- suggest entropy-based sampling for creating the buffer dataset of old examples. Samples where the output of the neural network has a larger cross-entropy are more likely picked.
- (while this approach allows for single-head classification, it still heavily relies on the disjoint dataset assumption. The basic idea is still that specific parameters have more importance for specific tasks and updating them when training new tasks should be avoided/reduced. For our application goals, this regularized loss could be used for a brief duration of the training when a new class is introduced. The first task would be defined as all previously known classes and the second task would consist of one new class only. This could be used in a strategy to focus learning of the new class while mitigating forgetting of the previous classes)

Gradient Episodic Memory for Continual Learning (GEM) [9]

- TI-CL, online, rehersal (+regularization) based approach
- They introduce metrics for evaluting backward and forward transfer
- No assumptions on the number of tasks are made.
- Use Memory buffer to constraint updates when training new tasks
- Constraint: gradient direction of past task (estimated with memory) has positive dot product with gradient from batch (of new task).
- Disadvantage: slow optimization with constraint and TASK INCEMENTAL

Continuous Learning in Single-Incremental-Task Scenarios [10]

- CI-CL, disjoint, introduce CWR and AR1 for NC (new class) learning, where each batch can contains new classes, but argue this could be adapted for NIC (new instance or new class) learning.
- main idea: for the final layer have one set of consolidated weights used for inference and tempory weights reset to 0 for each batch used to updated the subset of weichts in the consolidated weights matrix relevant to the class seen in the current batch (CWR)
- while CWR uses fixed represenations extracted from a model, AR1 allows end-to-end CL by allowing model used for extracting to be trained simultaneously using regularized loss in a controlled manner. They use Synaptic Inteligence (a variant of EWC [16])
- [7] expanded on this approach for the NIC task by updating weights for a class already seen using a weighted sum of past and current weights for the consolidation step
- the results of this approach was further imporved on using laten replay method [11]
- [7] also provides a benchmark protocol for Core50 dataset on github for a NIC task

3.4 Parameter Isolation

Generally, this approach to continual learning is again originaly designed for task-incremental learning. The main idea is to generate binary masks for parameters for each task indicating the their importance for specific tasks. Susequently learned tasks are learned only using the leftover parmeters in a network. This approach generally relies on task-incremental learning.

Conditional Channel Gated Networks for Task-Aware Continual Learning

- CI-CL, offline, disjoint (assumes stream produced samples for one task for a duration in time, but during inference, no task information is provided.)
- Original parameter isolation methods are not designed for class-incremental learning. This paper tries to generalize the formulation to class-incremental learning scenarios using some rehersal.
- main idea: jointly predict task and class label.
- use gating module for each convolutional layer which decides which kernel in the layer should be applied (binary decision) based on the input feature. The gating module consists of a very shallow neural network tained with a sparsity objective such that the smallest possible number of kernels are applied. after the training of a task, the most important parameters are frozen, i.e., their gradients are zeroed out during updates for subsequent task learning.
- (I dont see the advantage of parameter isoltion methods for class incremental learning. This approach practically splits the network into subsets for each task.)

References

[1] R. Aljundi, L. Caccia, E. Belilovsky, M. Caccia, M. Lin, L. Charlin, and T. Tuytelaars. Online continual learning with maximally interfered retrieval. *ArXiv*, abs/1908.04742, 2019.

[2] R. Aljundi, M. Lin, B. Goujaud, and Y. Bengio. Gradient based sample selection for online continual learning. In *NeurIPS*, 2019.

[3] A. Chaudhry, P. K. Dokania, T. Ajanthan, and P. H. Torr. Riemannian walk for incremental learning: Understanding forgetting and intransigence. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 532–547, 2018.

[4] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.

[5] J. Knoblauch, H. Husain, and T. Diethe. Optimal continual learning has perfect memory and is np-hard. In *ICML*, 2020.

[6] Z. Li and D. Hoiem. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947, 2017.

[7] V. Lomonaco, D. Maltoni, and L. Pellegrini. Rehearsal-free continual learning over small non-i.i.d. batches. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 989–998, 2020.

[8] V. Lomonaco, L. Pellegrini, P. Rodríguez, M. Caccia, Q. She, Y. Chen, Q. Jodelet, R. Wang, Z. Mai, D. Vázquez, G. I. Parisi, N. Churamani, M. Pickett, I. H. Laradji, and D. Maltoni. Cvpr 2020 continual learning in computer vision competition: Approaches, results, current challenges and future directions. *ArXiv*, abs/2009.09929, 2020.

[9] D. Lopez-Paz and M. Ranzato. Gradient episodic memory for continual learning. volume 30, pages 6467–6476, 2017.

[10] D. Maltoni and V. Lomonaco. Continuous learning in single-incremental-task scenarios. *Neural networks : the official journal of the International Neural Network Society*, 116:56–73, 2019.

[11] L. Pellegrini, G. Graffieti, V. Lomonaco, and D. Maltoni. Latent replay for real-time continual learning. *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 10203–10209, 2020.

[12] F. Pelosin and A. Torsello. More is better: An analysis of instance quantity/quality trade-off in rehearsal-based continual learning. *ArXiv*, abs/2105.14106, 2021.

[13] A. Prabhu, P. H. S. Torr, and P. Dokania. Gdumb: A simple approach that questions our progress in continual learning. In *ECCV*, 2020.

[14] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 2001–2010, 2017.

[15] H. Shin, J. K. Lee, J. Kim, and J. Kim. Continual learning with deep generative replay. *arXiv preprint arXiv:1705.08690*, 2017.

[16] F. Zenke, B. Poole, and S. Ganguli. Continual learning through synaptic intelligence. In *International Conference on Machine Learning*, pages 3987–3995. PMLR, 2017.