CSED490Y OptML
Midway Presentation

Team 1
Yechan Hwang/Sungbin Shin

Contents

01 Introduction

Understanding the Role of Training Regimes in Continual Learning (NeurIPS 2020)

Summary: This paper study the role of training regimes (learning rate, batch size, dropout, …) in continual learning, especially focusing on catastrophic forgetting.

Understanding the Role of Training Regimes in Continual Learning

---

Seyed Iman Mirzadeh
Washington State University, USA
seyediman.mirzadeh@wsu.edu

Mehrdad Farajtabar
DeepMind, USA
farajtabar@google.com

Razvan Pascanu
DeepMind, UK
razp@google.com

Hassan Ghasemzadeh
Washington State University, USA
hassan.ghasemzadeh@wsu.edu

Continual Learning

Goal: Even after learning Task 2, still maintaining good performance on the test dataset of task 1.

During training on each task, the data from previous ones are unavailable.

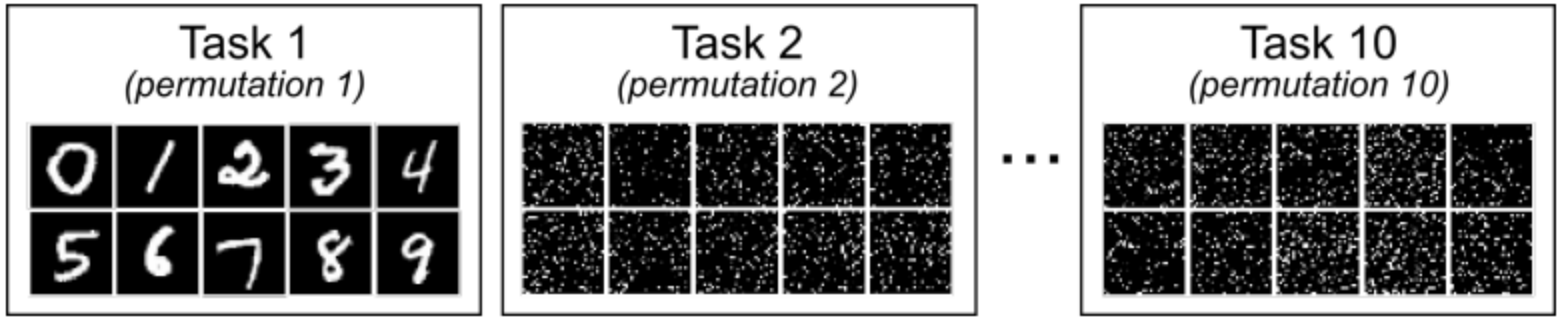
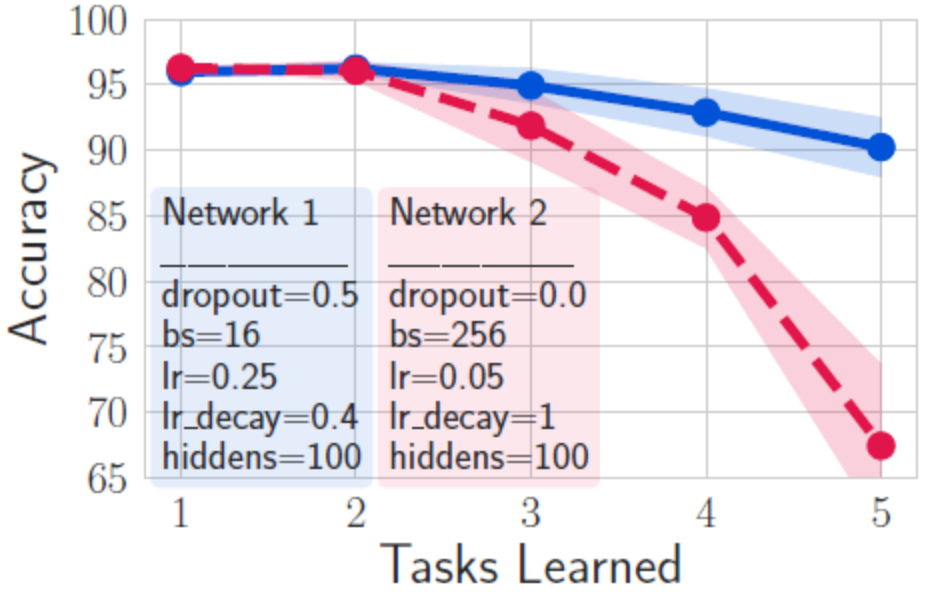


Figure 2: Schematic of permuted MNIST task protocol.

Three scenarios for continual learning (NeurIPS Workshop, 2018)

Catastrophic Forgetting

As the model learns newer tasks, the performance of the model on older ones degrades.



[Figure] For the same architecture and dataset (Rotation MNIST) and only changing the training regime, the forgetting is reduced significantly at the cost of a relatively small accuracy drop on the current task.

02 Hypothesis

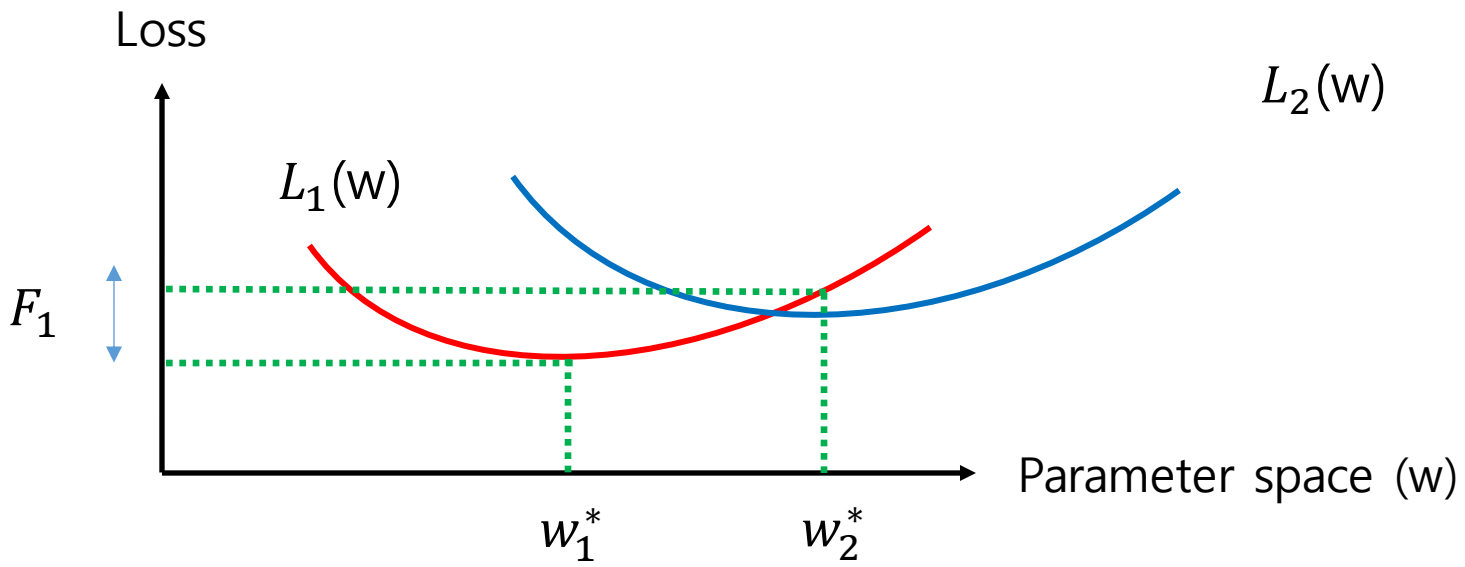
The amount of forgetting that a neural network exhibits from learning the tasks sequentially, correlates with the geometrical properties of the convergent points.
In particular, the wider these minima are, the less forgetting happens.

Total loss on the training set for task k

$$L_k(w) = \mathbb{E}[\ell_k(w; x, y)] \approx \frac{1}{|\mathcal{T}_k|} \sum_{(x,y)\in\mathcal{T}_k} , \ell_k(w; x, y)$$

Forgetting of the first task

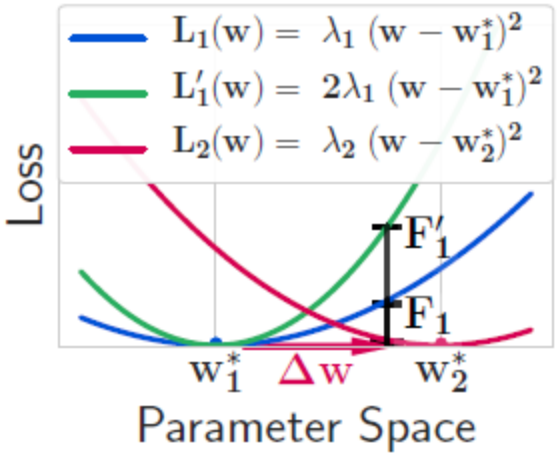
$$F_1 \triangleq L_1(w_2^*) - L_1(w_1^*).$$



02 Hypothesis

The amount of forgetting that a neural network exhibits from learning the tasks sequentially, correlates with the geometrical properties of the convergent points.
In particular, the wider these minima are, the less forgetting happens.

$$F_1 = L_1(w_2^*) - L_1(w_1^*) \approx \frac{1}{2}\Delta w^\top \nabla^2 L_1(w_1^*)\Delta w \leq \frac{1}{2}\lambda_1^{max} \|\Delta w\|^2$$



(a)

(a) - For a fixed Δw, the wider the curvature of the first task, the less the forgetting.

03 Training Regimes

Techniques that are known to affect the width of the minima(eigenvalues of Hessian) as well as the length of the path taken by learning($||\Delta w||$).

- A high learning rate or a small batch size limits the eigenvalues of Hessian of loss function.
  => increases the probability of converging to a wider minima.

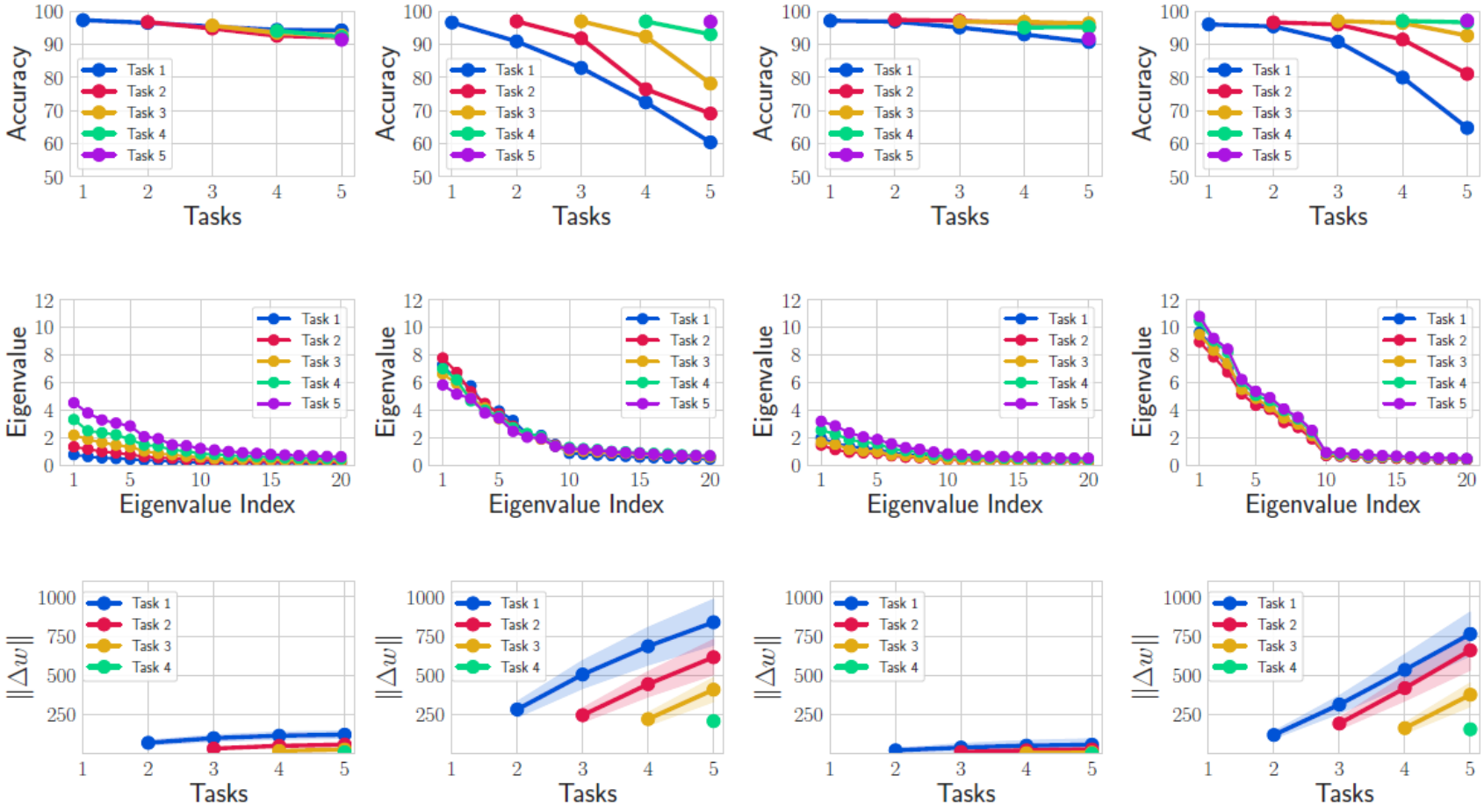
- A high learning rate contributes to the rate of change(i.e., Δw).
  => higher update to the neural network weights.($\Delta w \uparrow$)

- Start with a high initial learning rate for the first task to obtain a wide and stable minima. Then, for each subsequent task, slightly decrease the learning rate but also decrease the batch-size instead.

- Dropout encourages the wideness of the minima since it minimizes the second derivative of the loss.

04 Experiments and Results

Comparison of training regimes for MNIST datasets



(a) Permuted - Stable (b) Permuted - Plastic (c) Rotated - Stable (d) Rotated - Plastic

Stable network

- Dropout regularization
- Large initial learning rate
- Learning rate decay at the end of each task
- Small batch size

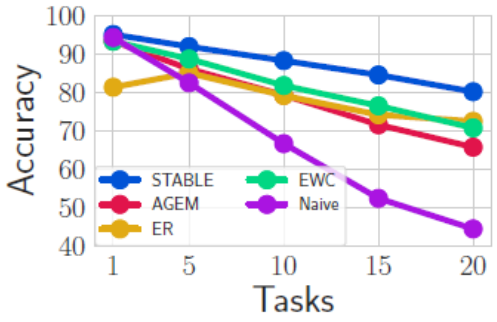
$$F_1 \leq \frac{1}{2}\lambda_1^{max} \|\Delta w\|^2$$

04 Experiments and Results

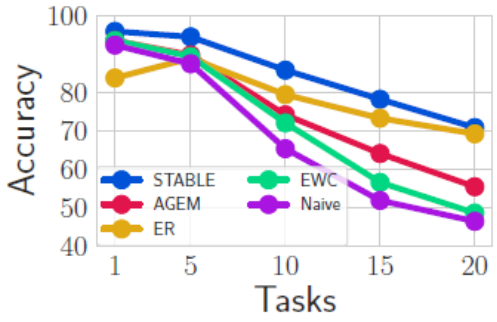
Comparison with several previously proposed methods

Table 2: Comparison of the average accuracy and forgetting of several methods on three datasets.

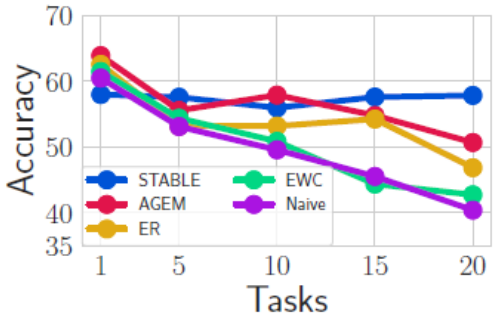
Method	Memoryless	Permuted MNIST		Rotated MNIST		Split CIFAR100	
		Accuracy	Forgetting	Accuracy	Forgetting	Accuracy	Forgetting
Naive SGD	✓	44.4 (±2.46)	0.53 (±0.03)	46.3 (±1.37)	0.52 (±0.01)	40.4 (±2.83)	0.31 (±0.02)
EWC	✓	70.7 (±1.74)	0.23 (±0.01)	48.5 (±1.24)	0.48 (±0.01)	42.7 (±1.89)	0.28 (±0.03)
A-GEM	✗	65.7 (±0.51)	0.29 (±0.01)	55.3 (±1.47)	0.42 (±0.01)	50.7 (±2.32)	0.19 (±0.04)
ER-Reservoir	✗	72.4 (±0.42)	0.16 (±0.01)	69.2 (±1.10)	0.21 (±0.01)	46.9 (±0.76)	0.21 (±0.03)
Stable SGD	✓	**80.1 (±0.51)**	**0.09 (±0.01)**	**70.8 (±0.78)**	**0.10 (±0.02)**	**59.9 (±1.81)**	**0.08 (±0.01)**
Multi-Task Learning	N/A	86.5 (±0.21)	0.0	87.3(±0.47)	0.0	64.8(±0.72)	0.0



(a) Permuted MNIST



(b) Rotated MNIST



(c) Split CIFAR-100

Figure 4: Evolution of the average accuracy during the continual learning experience with 20 tasks

05 Conclusion

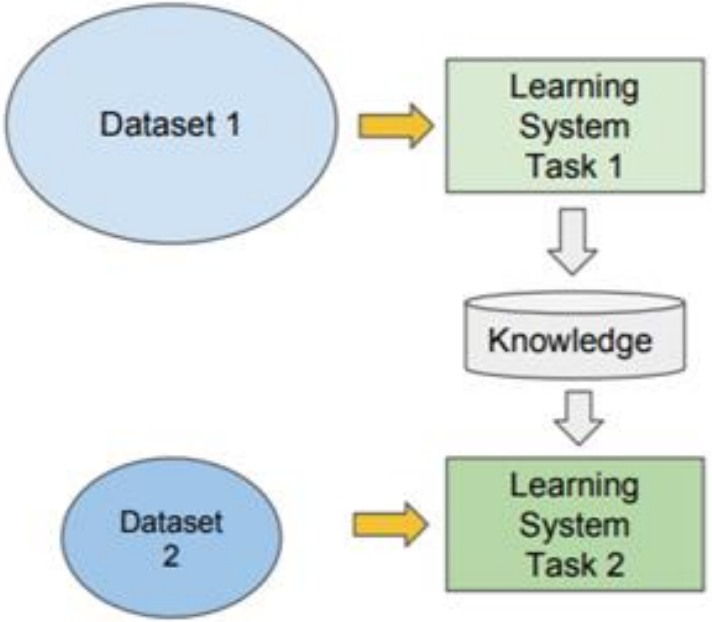
- They confirmed that in continual learning, the wider the minima are, the less forgetting happens.

- Empirically observed that simple techniques proved to be more effective than some of the recent approaches.

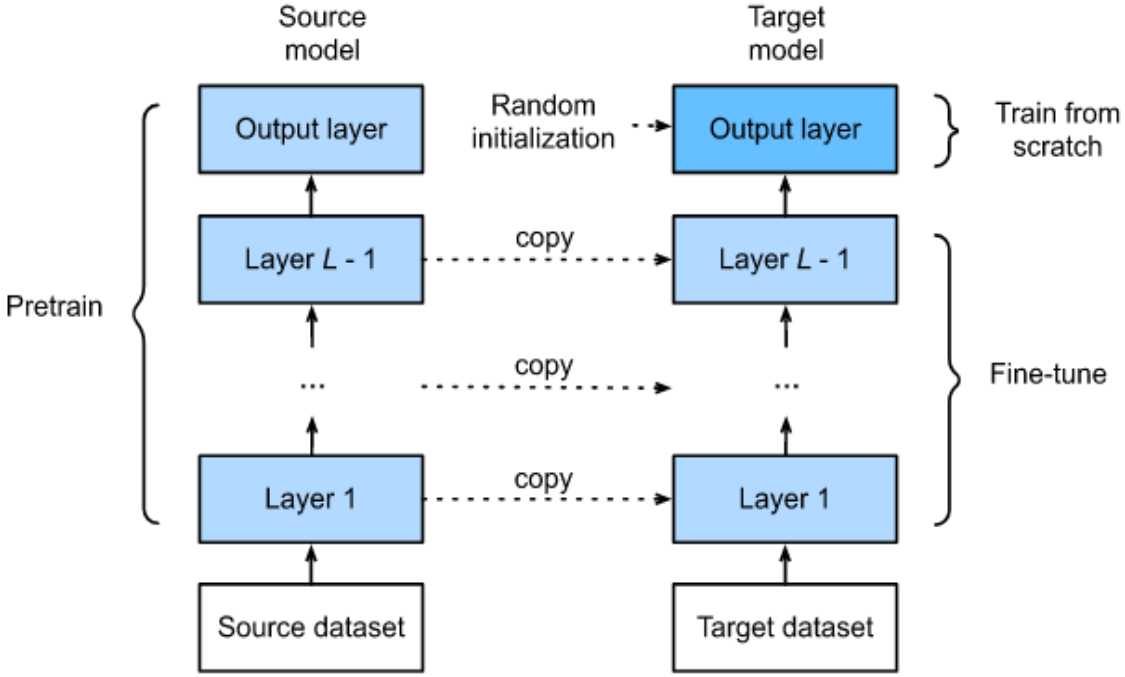
06 Relation to our project

Our Project: Effect of learning rate scheduling in transfer learning

Transfer learning



Pretrain & Fine-tune



Our Project: Effect of learning rate scheduling in transfer learning

How is the paper related to our project?

- We will do a similar experiment as the paper, but the task has been changed
- How important is the training regime in transfer learning?
- Experiment with different learning rate schedulers & hyperparameters
  - Learning rate schedulers: Constant learning rate, step weight decay, ...
  - Hyperparameters: Initial learning rate, weight decay factor, ...

Our Project: Effect of learning rate scheduling in transfer learning

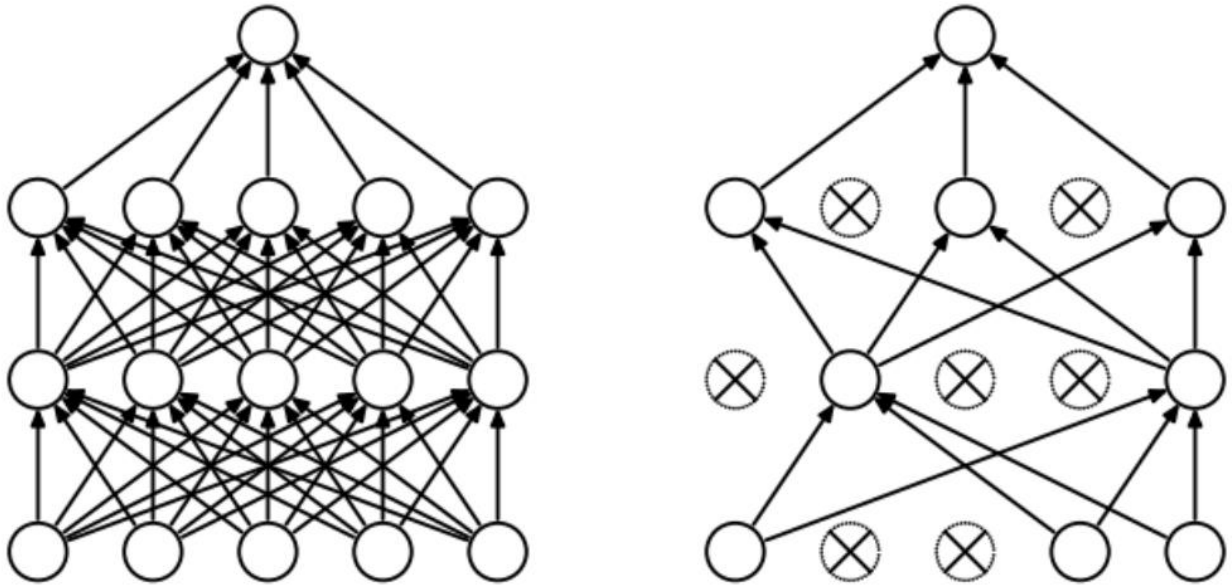
Difference between the paper and our project

- Only focus on the aspect of learning rate
- Focus on convergence speed rather than catastrophic forgetting

Thanks

Appendix

Dropout



(a) Standard Neural Net

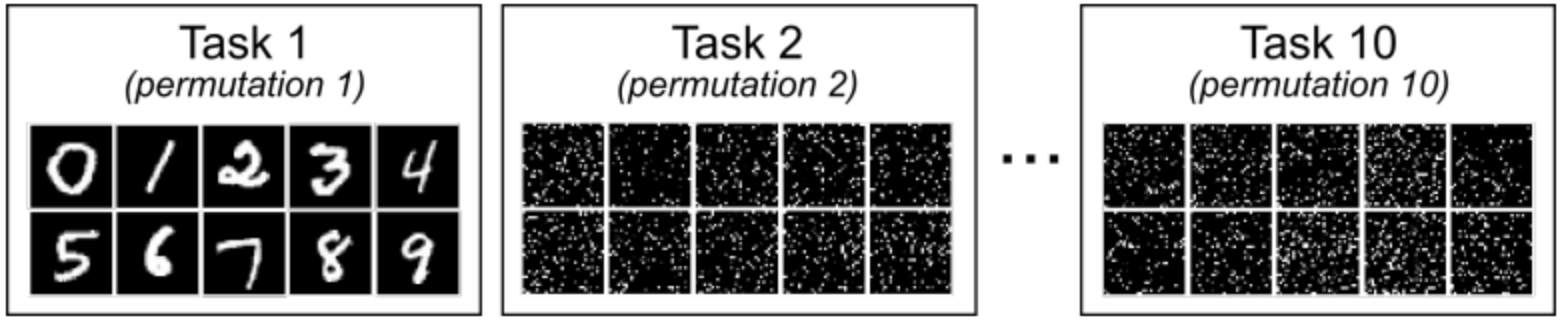
(b) After applying dropout.

Srivastava, Nitish, et al. "Dropout: a simple way to prevent neural networks from overfitting", JMLR 2014

Appendix

Datasets

Permuted MNIST



Rotated MNIST

Deep Domain Generalization via Conditional Invariant Adversarial Networks (ECCV, 2018)

Appendix

Datasets

Split CIFAR-100

A variant of the CIFAR-100 where each task contains the data from 5 random classes (without replacement) out of the total 100 classes.

Appendix

Evaluation metrics

(1) Average Accuracy: The average validation accuracy after the model has been trained sequentially up to task t, defined by:

$$\mathbf{A_t} = \frac{1}{t}\sum_{i=1}^{t} a_{t,i}$$

where, $a_{t,i}$ is the validation accuracy on dataset i when the model finished learning task t.

(2) Average Forgetting: The average forgetting after the model has been trained sequentially on all tasks. Forgetting is defined as the decrease in performance at each of the tasks between their peak accuracy and their accuracy after the continual learning experience has finished. For a continual learning dataset with T sequential tasks, it is defined by:

where, $a_{t,i}$ is the validation accuracy on dataset i when the model finished learning task t.

Appendix

Proof of $$F_1 = L_1(w_2^*) - L_1(w_1^*) \approx \frac{1}{2}\Delta w^\top \nabla^2 L_1(w_1^*)\Delta w \leq \frac{1}{2}\lambda_1^{max} \|\Delta w\|^2$$

Forgetting of the first task

$$F_1 \triangleq L_1(w_2^*) - L_1(w_1^*).$$

Using second-order Taylor expansion,

$$L_1(w_2^*) \approx L_1(w_1^*) + (w_2^* - w_1^*)^\top \nabla L_1(w_1^*) + \frac{1}{2}(w_2^* - w_1^*)^\top \nabla^2 L_1(w_1^*)(w_2^* - w_1^*)$$

$$\approx L_1(w_1^*) + \frac{1}{2}(w_2^* - w_1^*)^\top \nabla^2 L_1(w_1^*)(w_2^* - w_1^*),$$

$$F_1 = L_1(w_2^*) - L_1(w_1^*) \approx \frac{1}{2}\Delta w^\top \nabla^2 L_1(w_1^*)\Delta w \leq \frac{1}{2}\lambda_1^{max} \|\Delta w\|^2$$

where $\Delta w = w_2^* - w_1^*$

Appendix

Proof of $$F_1 = L_1(w_2^*) - L_1(w_1^*) \approx \frac{1}{2}\Delta w^\top \nabla^2 L_1(w_1^*)\Delta w \leq \frac{1}{2}\lambda_1^{max} \|\Delta w\|^2$$

Think of the eigendecomposition of the Hessian. Since Hessian is symmetric,

$$\begin{aligned}\nabla^2 L_1(w_1^*) &= Q\Lambda Q^{-1} \\ &= Q\Lambda Q^\intercal\end{aligned}$$

where Q is an orthgonal matrix whose columns are eigenvectors and Λ is an diagonal matrix with eigenvalues.

Then,

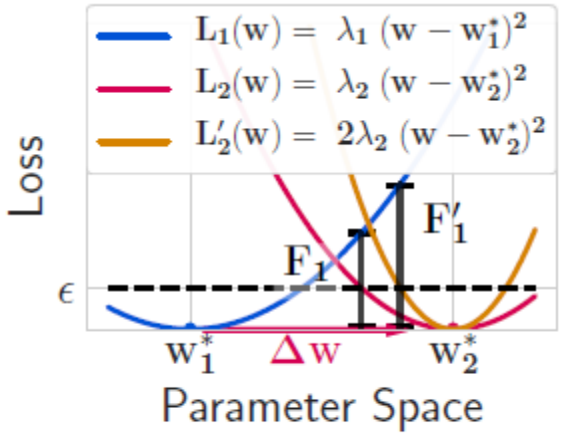
$$\begin{aligned}\frac{1}{2}\Delta w^\intercal \nabla^2 L_1(w_1^*)\Delta w &= \frac{1}{2}\Delta w^\intercal (Q\Lambda Q^{-1})\Delta w \\ &= \frac{1}{2}(Q^\intercal \Delta w)^\intercal \Lambda (Q^\intercal \Delta w) \\ &\leq \frac{1}{2}\lambda_1^{max}||Q^\intercal \Delta w||^2 \\ &= \frac{1}{2}\lambda_1^{max}||\Delta w||^2\end{aligned}$$

The last equality comes from the fact that $||Qx|| = ||x||$ if Q is an orthogonal matrix

Appendix

Curvature of the second task is also important!

$$F_1 = L_1(w_2^*) - L_1(w_1^*) \approx \frac{1}{2}\Delta w^\top \nabla^2 L_1(w_1^*)\Delta w \leq \frac{1}{2}\lambda_1^{max} \|\Delta w\|^2$$



(b)

(b) - The wider the curvature of the second task, the smaller $||\Delta w||$.