EECS 182 Deep Neural Networks
Fall 2022 Anant Sahai

Discussion 11

1. Catastrophic Forgetting

The neural networks are vulnerable to the distributional shift. Many questions in AI/ML are related to the distributional shift: out-of-distribution (OoD), domain adaptation/generalization, meta-learning, and so on. In this discussion, we study one of those problems, catastrophic forgetting, alternatively called catastrophic interference. The catastrophic forgetting is the tendency of neural networks to lose information about previously learned tasks when learning the new one. This is also referred to as stability-plasticity dilemma[1]. One potential drawback of a model that is too stable is that it will not be able to consume new information from the future training data. Conversely, a model with sufficient plasticity may suffer from large weight changes and forgetting of previously learned representations.

Neural network models are trained on the assumption of i.i.d, meaning that data points are sampled from a mutually independent and identical distribution. However, this assumption does not apply to real-world applications, such as sequential data stream training settings, which can lead to catastrophic forgetting.

Let's dissect the underlying mechanisms of catastrophic forgetting. Three factors cause catastrophic forgetting mainly: parameter shift, activation shift, and inter-task confusion[2].

(a) **Parameter shift** - when the gradient descent step updates the neural networks, the parameters drift to the region where the previous tasks' error is high in the parameter space.

(b) **Activation shift** - Activation shift is the direct ramification of parameter shift. However, focusing on activation can relax catastrophic forgetting as long as activations change minimally when the neural networks are trained with the new task.

(c) **Inter-task confusion** - Since all the tasks are not jointly trained, the outputs and intermediate activations lead to inter-task misclassification.

A cartoon depiction of catastrophic forgetting is in Figure 1. Without any implicit or explicit methods that prevent the model from catastrophic forgetting, the model would forget the useful features to discriminate samples in task 1 when it learns the new task. The right image in Figure 1 is the ideal case in which the model can capture the discriminative features for both previous and new tasks.

(a) What happens in parameter space

Figure 2 is the schematic parameter space of the given model and tasks. The orange oval is the simplified low-loss region of task 0, and the blue oval is that of task 1. The model is trained on the task 0, and θ_0^* is the trained parameter of the model, which minimizes the loss for that task. We now train the model with task 1 data.

i. Mark the θ_1^* on the Figure 2, which is the trained parameter for task 1 if the model is too plastic.

ii. Mark the $\tilde{\theta_1}$ on the Figure 2, which is the trained parameter for task 1 if the model is too stable.

iii. What are the problems if the model is too plastic or stable?

[1] https://www.sciencedirect.com/science/article/pii/S1364661399012942

[2] https://arxiv.org/pdf/2010.15277.pdf

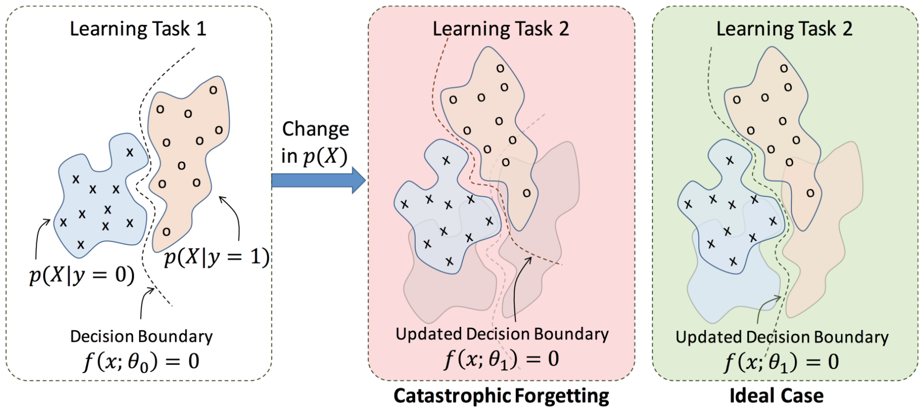

Figure 1: The binary classification model $f(x;\theta_0)$ is initially trained with task 1. If the task shifts from 1 to 2, the model may suffer catastrophic forgetting. https://arxiv.org/pdf/1903.06070.pdf

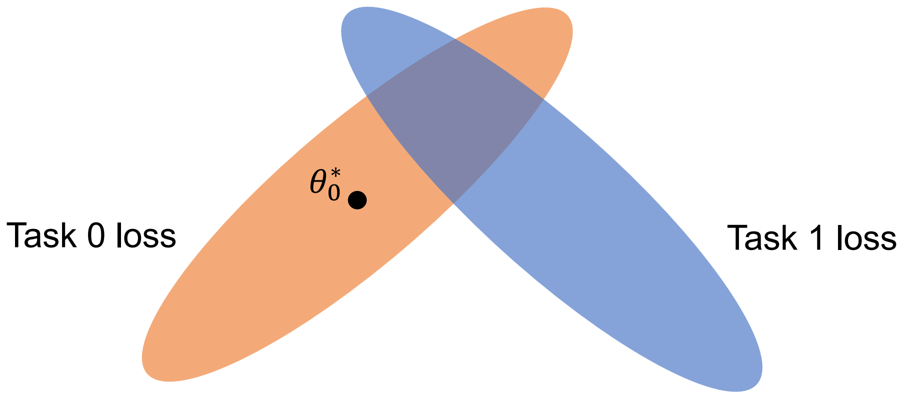

Figure 2: The schematic parameter space.

(b) Dealing with catastrophic forgetting

We consider three traditional approaches for learning the new tasks: feature extraction, fine-tuning, and joint training.

i. feature extraction - The model trained with the previous tasks is frozen. The output of this model with the new task input is used for to train the new classifier for the new task.

ii. fine-tuning - The model trained with the previous tasks is trained with the new task. To prevent the large shift, the learning rate is typically low.

iii. joint training - The model is trained with both previous task data and new task data

Let's study pros and cons of those methods. Fill in the table below.

Category	Feature Extraction	Fine tuning	Joint training
New task performance			Good
Old task performance	Good		Good
Data storage requirment		Low	
Storing old task data	No	No	

(c) Continual Learning (Optional)

As shown in the previous problem, fine-tuning, feature extraction and joint training are not suitable for handling catastrophic forgetting. Then, what should we do? The answer is continual learning.

Continual learning (or incremental learning) is the subfield of AI/ML in which a single neural network model learns a series of tasks sequentially.

In this part, we assume that during training the task is clearly separated and the current task's data is only available. Then, we can categorize three scenarios of continual learning problems.[3]

i. Task continual learning: Solve tasks so far, task-ID provided
ii. Domain continual learning: Solve tasks so far, task-ID not provided
iii. Class continual learning: Solve tasks so far and infer task-ID

In the first scenario, models are informed about which task they need to perform. This is the easiest scenario since task identity is always provided. In the second scenario, task identity is not available at test time. However, models only need to solve the task at hand and they are not required to infer which task it is. The third scenario is where models must be able to both solve each task seen so far and infer which task they are presented with. It includes the common real-world problem of incrementally learning new classes of objects.

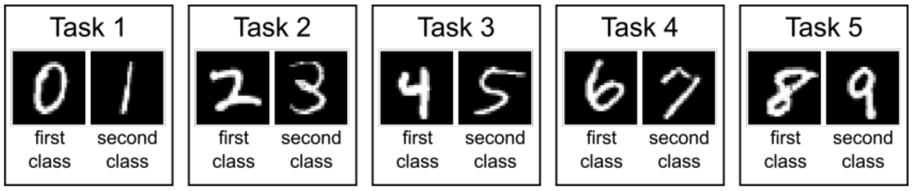

Figure 3: Continual learning task protocol. The dataset is MNIST.

To understand the difference between the three continual learning scenarios, we will perform continual learning task protocols for all scenarios. The simplest task protocol is to classify MNIST in 3. Reading the description of scenarios in the right column and fill in the blank.

Scenarios	Description
	With the task id given, is it the 1st or 2nd class? E.g. if the given task is 1, is the test image 0 or 1?
	Without task id, is it the 1st or 2nd class? I.e. predict the test image is in {0, 2, 4, 6, 8} or {1, 3, 5, 7, 9}
	Without task id, predict the class that the test image belongs to. i.e. choice the label of image from 0 to 9

[3] https://arxiv.org/pdf/1904.07734.pdf