50.035 Computer Vision

Term Paper

Ashlyn Goh 1002840

Topic Chosen: **Topic A**

(i) What are the differences between this work (Self-supervised GAN) and the original unconditional GAN as discussed in 50.035?

1. Self-supervised vs unsupervised learning

- The original unconditional GAN uses unsupervised learning where the input training data comes with no labels. The goal of the GAN is to model the probability distribution of the training data, and be able to generate new examples of what it has learnt.
- In this work, self-supervision is added to the discriminator by applying a rotation-based loss. The angle of rotation thus becomes the pseudo-label, and the self-supervised task is to predict the angle of rotation (0, 90, 180 or 270 degrees). This encourages the discriminator to understand the global structure in the input training data and learn important feature representations that are not forgotten during training. Hence, the discriminator is trained to learn more stable and improved representations to counter catastrophic forgetting.

2. Collaborative Adversarial Training

- Similar to the original GAN, the generator and discriminator in Self-supervised GAN are adversarial with regards to the binary classification (true/fake) loss, competing with each other to improve themselves.
- However, in this work, the generator and discriminator are also collaborative with regards to the rotation task (which is not seen in the original GAN). The goal of the discriminator on rotated real images is to detect the rotation angle, while the generator aims to generate images matching the observed data, in which the rotation angle is easy to detect by the discriminator. This makes it harder for the discriminator to distinguish them from the real images, helping the discriminator to learn better representations, thus bringing its performance level close to the state-of-the-art conditional GANs (which require labelled data).

(ii) In Section 2, the authors discussed discriminator forgetting. Explain what is discriminator forgetting. Why does it happen in training a GAN?

Discriminator forgetting is the problem where the discriminator loses the ability to remember synthesised samples from previous generator samples. At iteration t during training, the discriminator in GAN classifies samples as coming from the true data distribution, P_{data} or the distribution induced by the generator mapping, $P^{(t)}_G$. However, the distribution of the generated fake images changes ($P^{(t)}_G$) as the generator's parameters are updated over time. The knowledge required to separate $P^{(t)}_G$ from P_{data} is different from that for the pair $\{P^{(t-1)}_G, P_{data}\}$. This thus implies a non-stationary online learning problem for the discriminator, making it difficult for the discriminator to learn useful features to identify real images from fake ones.

One source of training instability is also that the discriminator is not incentivised to maintain useful representations provided that the current representation is useful to classify between real and fake images. Thus, the discriminator will "forget" the data representation it has learnt to distinguish between the real images and a previous iteration of generated images, if the representation is not useful for the current task.

(iii) What are the results in Figure 3? What is the motivation of this experiment? There are repetitive dips in Figure 3(a) and 3(b). What causes these dips? Why are some dips more significant than the others?

Figure 3 shows the classification accuracy when a classifier is trained sequentially on 1-vs-all classification tasks on each of the 10 classes in the CIFAR10 dataset. The classifier is trained for 1000 iterations on each task before switching to the next. At 10, 000 iterations, the training cycle repeats from the first task.

The results in Figure 3(a) shows that the image classification accuracy suffers from periodic drastic declines that match with the task switches, suggesting that the classifier suffers from forgetting, despite the tasks being similar. However, when the self-supervised rotation loss is added to the same classifier (in Figure 3(b)), the dips are not as drastic, and the performance continually improves. On the second cycle through the tasks from 10,000 iterations onwards, performance improved as

compared to when the discriminator meets the same task the first time, demonstrating that self-supervision helps to mitigate the forgetting problem.

The motivation of this experiment is to illustrate catastrophic forgetting in classification problems, which can be extended to discriminator forgetting. This problem can be alleviated by adding a self-supervised loss.

The repetitive dips happen when the task switches, and thus they are caused by the model not retaining generalisable representations in the non-stationary environment. The representations learnt in the previous task may not be relevant anymore, leading to a drop in performance.

Some dips are more significant than the others because the last learnt representation is not very useful for the current task. This may suggest that the previous class is significantly different from the current class (i.e. tasks are very different), thus the network did not learn a representation that could be generalised and transferred across to this current task. This leads to a significant drop in performance as represented by the notable dips.

(iv) What are the purposes to perform rotation degree classification in Figure 1? How is Figure 1 related to the equations in Section 3?

The purpose of performing rotation degree classification is so that the discriminator can learn stable representations. The intuition is that for the discriminator to effectively perform the rotation classification, it must have learnt to recognise and detect classes of image and their semantic parts in the images. The model must necessarily learn to localise salient objects in the image, recognise their object type and its current orientation. Since the weights are shared between the original binary classification task and rotation task, this auxiliary task helps the discriminator learn better, countering the discriminator forgetting problem.

From Figure 1, we can see that both the images produced by the generator and the real images are rotated by 0, 90, 180, and 270 degrees. These rotated images are then used for the rotation task. The rotation-based loss for the generator thus involves penalising the generator for not producing fake images whose rotation angle is easy to detect by the discriminator. In other words, as seen in the loss functions in Section 3, the generator's objective is to minimise the cross-entropy loss for rotation prediction for the generated images. Similarly, the discriminator's rotation error is penalised based on whether it could predict the rotation of real images correctly. Only real images are used for the discriminator's loss to discourage unhealthy convergence from the generator, encouraging the generator to generate images whose rotation is easy to predict because they actually share features with real images that are used for rotation classification. That is, the generator should not be generating images that are rotation-detectable but in reality not matching the observed data and sharing features with the real images.

At the same time, only the upright images are used for the binary classification (real/fake) in the discriminator. The binary classification loss is used in the loss function of both the generator and discriminator, L_G and L_D, by including V(G, D) which follows the original value function for GAN training.

(v) In Section 3, in the equation for L_G, why P_{data} is not used in the rotation-based loss?

This is because the generator is unable to directly impact the rotation-based loss of the training image dataset. If P_{data} was used, the loss will be constant through the iterations because the generator cannot affect the loss value. The distribution, P_G is thus used instead of P_{data} , where the loss function of the generator includes whether the discriminator is successful in prediction the rotation in the generated (fake) images. Thus, the generator wants to generate images in which rotation angle is easy to detect by the discriminator, which makes it harder for the discriminator to distinguish them from the real images, ultimately helping the discriminator to learn better.