人工智慧模型設計與應用 Lab5

NM6121030 余振揚

1. **Outline:**

	Loss	Accuracy
Teacher Model	0.46	87.53
Student Model	0.51	86.61
Response-based Distillation	0.73	89.11
Filter-based Distillation	0.50	86.78

2. **Train From Scratch:**
   - **Teacher Model (ResNet34):**
     - Total Params: 21,282,122
     - Epochs = 40
     - Learning Rate = 0.01
     - Batch Size = 128
   - **Student Model (ResNet18):**
     - Total Params: 11,173,963
     - Epochs = 40
     - Learning Rate = 0.01
     - Batch Size = 128
   - **Test Accuracy (Initial Performance):**
     - Teacher = 87.53
     - Student = 86.61

   其實使用 ResNet18 架構的 Student Model 在 CIFAR10 資料集上已經有很好的表現了，而後續我們要透過 Knowledge Distillation 的方法，讓 Student Model 能從 Teacher Model 學習到更多 feature，加以修正 weight。

3. **Response-based Distillation:**

   為了確保 accuracy 僅受 Distillation 影響，所以不改變 Hyper Parameters 的任何參數。

   - Alpha = 0.5
   - T = 5
   - Loss Function Design:

```
########################################################################
# Finish the loss function for response-based distillation. #
########################################################################
def loss_re(T, alpha, teacher_logits, student_logits, target):
    softmax_teacher = F.softmax(teacher_logits/T, dim=1)
    softmax_student = F.log_softmax(student_logits/T, dim=1)


    loss_soft_target = F.kl_div(softmax_student, softmax_teacher, reduction='batchmean')
    loss_hard_target = F.cross_entropy(student_logits, target)

    loss = alpha * T**2 * loss_soft_target + (1 - alpha) * loss_hard_target

    return loss
```

4. **Filter-based Distillation:**

   為了確保 accuracy 僅受 Distillation 影響，所以不改變 Hyper Parameters 的任何參數。

   - Alpha = 0.0001
   - Loss Function Design:

```
########################################################################
# Finish the loss function for feature-based distillation. #
########################################################################
def loss_fe(alpha, teacher_features, student_features, student_logits, target):

    # Euclidean distance
    loss_feature = 0.0
    for i in range(len(teacher_features)):
        loss_feature += F.mse_loss(teacher_features[i], student_features[i])

    loss_hard = F.cross_entropy(student_logits, target)

    loss = loss_hard + alpha * loss_feature/128


    return loss
```

5. **Feedback and Problem Encounter:**

   當初在設計 filter-based distillation 的 loss function 時，僅考慮了 feature layer 之間的歐氏距離，但沒有傳入 hard_target 的參數，導致最後的 loss 僅參考 Techer 的 feature，最後 train 出來的 model accuracy 只有 8%，最後是問了組員才發現自己的錯誤。

6. **Reference:**

   - https://pytorch.org/tutorials/beginner/knowledge_distillation_tutorial.html