Report

1) Rotation training

- SGD with momentum was only able to achieve a maximum test accuracy of around 0.65, so I switched to the Adam optimizer for better convergence.
- Default hyperparameters were used: num_epochs=45, decay_epochs=15, init_lr=0.01.
- Max test accuracy = **78.60%**.

2) Fine-tuning late layers

- I used Adam with default hyperparameters: num_epochs=20, decay_epochs=10, init_lr=0.01
- Pre-trained model max test accuracy = **61.54%**, randomly initialized model max test accuracy = **45.37%**.

3) Fully supervised learning

- I used Adam with default hyperparameters: num_epochs=30, decay_epochs=10, init_lr=0.01
- Pre-trained model test accuracy = **83.94%**, randomly initialized model max test accuracy = **10.15%**. (So here I observe that the model is not able to learn from random weights using this setting of optimizers, learning rate, and epoch).
- I also trained the model for more epochs (150), and even in this case, the model gave a similar performance for random initialization. This shows that the current system of optimizer and scheduler is not enough to train the model with random initialization. Since the purpose of this experiment is to compare the *pre-trained model and the randomly initialized model,* I did not change the scheduler and optimizer.

4) With a more advanced architecture

The architecture used in this work is based on the EfficientNetV2-L model, a convolutional neural network known for its optimized parameter efficiency and strong performance on image classification tasks. I initialize the model without pre-trained weights to allow for training from scratch. This setup enables us to fully utilize the depth and compound scaling advantages of the EfficientNetV2-L backbone.

4.1) Rotation Prediction model

Hyperparameters I used: num_epochs=165, decay_epochs=135, init_lr=1e-3, Adam optimizer.

- Max test accuracy = **85.38%**.

4.2) Classification Model

Here, I use the pre-trained rotation model (the one that I trained in the previous case)
Hyperparameters I used: num_epochs=20, decay_epochs=10, init_lr=1e-3, Adam optimizer.

- Max test accuracy = **87.43%**.

For the train-from-scratch paradigm following is the score:

- Max test accuracy = **69.38%**

So here with observe that training efficientnetv2_l from scratch is relatively easy then training Resnet 18 from scratch.

5) Additional Implementations

I recreated the experiment comparing semi-supervised and supervised classification models using subsets of CIFAR-10. Subsets with [20, 100, 400, 1000, 5000] images per class were used.

Semi-supervised model (from 2.1):

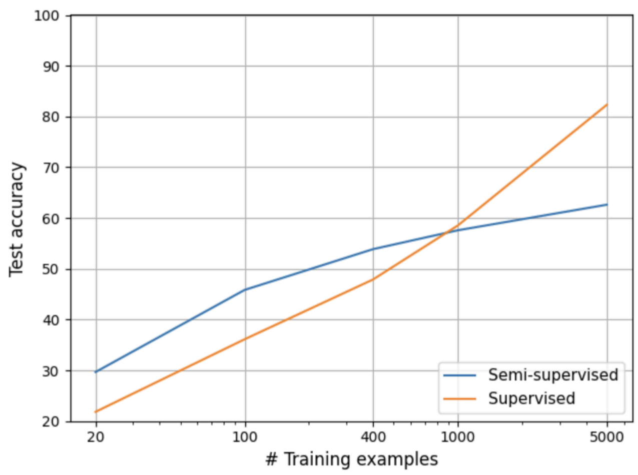
- ResNet18 was trained on **the rotation** prediction task using the entire CIFAR-10 dataset, which was fine-tuned on classification using a subset of labeled images. Only the last block (layer 4) and final linear layer (FC) were trainable; earlier layers were frozen.

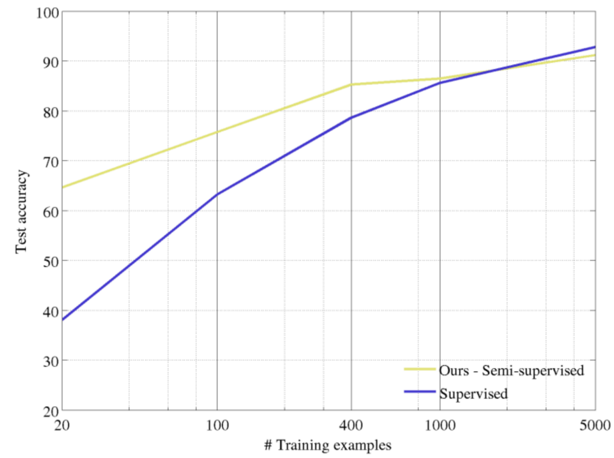
Supervised model (from 3.2):

- A new ResNet18 model was initialized with random weights trained on the same subsets as above, with all layers trainable.

Experimental Setup:
Optimizer: Adam
Loss: CrossEntropyLoss
Hyperparameters: num_epochs=30, decay_epochs=10, init_lr=0.01
Batch Size: 128
Evaluation: The entire CIFAR-10 test set

Results:





Recreated Figure 5b

Original Figure 5b

Discussion:
I was not able to achieve the exact accuracy values, probably because:

1) I used ResNet18, authors used NIN-based architecture
2) I had epochs = 30, authors had 100
3) I used Adam, and the authors used SGD with momentum

However, my graph still portrays a similar pattern discussed in the paper:

“We observe that our unsupervised trained model exceeds the supervised model in this semi-supervised setting when the number of examples per category drops below 1000.”
– *Gidaris et al. (2018), p. 9*

From the *Recerated Figure 5b*, we can see that the semi-supervised model performs better than the supervised model when the number of training examples per class is below 1000.