Congratulations! You have finished the coding part of this assignment! Now we can run some experiments to gain insight into the behaviors and advantages of unsupervised representation learning (when compared with supervised learning under fair conditions). The settings for running each experiment will be given to you. Here is a list of the arguments you will change:

rotnet_arch (str): architecture of the RotNet model (explained in Args class)
prednet_arch (str): architecture of the PredNet model (explained in Args class)
rotnet_ckpt_dir (str): RotNet checkpoints in this run will be stored in this directory
prednet_ckpt_dir (str): PredNet checkpoints in this run will be stored in this directory
rotnet_epochs (int): You will train RotNet for this number of epochs
prednet_epochs (int): You will train PredNet for this number of epochs
no_grad (bool): If you set this to True, gradients will not flow through RotNet when training PredNet
verbose (bool): If you set this to True, model and params info will be printed

Once we finish training the RotNet (unsupervised), we can extract the output from the last convolutional layer (before feeding it into the fully connected layer) as the feature representation to be used for the image classification task (achieved by PredNet).

```
--rotnet_arch rotnet3_feat3 --prednet_arch prednet0 --rotnet_ckpt_dir
./ckpts/rot3_feat3_pred0_unsup/rotnet --prednet_ckpt_dir
./ckpts/rot3_feat3_pred0_unsup/prednet
```

We can also use supervised learning for the image classification task. Train a PredNet with the same network structure and compare the learning curve (validation accuracy) with the unsupervised representation learning method. Which method gives better accuracy? Which method trains faster?

```
--rotnet_arch rotnet3_feat3 --prednet_arch prednet0 --rotnet_ckpt_dir
./ckpts/rot3_feat3_pred0_sup/rotnet --prednet_ckpt_dir ./ckpts/rot3_feat3_pred0_sup/prednet
--rotnet_epochs 0
```

Instead of using the output from the last convolutional layer as the feature representation, we can also use the output from the second-last convolutional layer or the third-last convolutional layer, etc. Run the following experiments and compare the results with that in the first question. Which feature representation leads to better performance of PredNet?

```
--rotnet_arch rotnet3_feat2 --prednet_arch prednet0 --rotnet_ckpt_dir
./ckpts/rot3_feat2_pred0_unsup/rotnet --prednet_ckpt_dir
./ckpts/rot3_feat2_pred0_unsup/prednet
```

```
--rotnet_arch rotnet3_feat1 --prednet_arch prednet0 --rotnet_ckpt_dir
./ckpts/rot3_feat1_pred0_unsup/rotnet --prednet_ckpt_dir
./ckpts/rot3_feat1_pred0_unsup/prednet
```

Notice that in the above experiments, we always allow gradients to flow through RotNet when training PredNet, which means we are tuning the model parameters of RotNet together with those of Prednet. Use the “no_grad” flag to disallow gradients to flow through RotNet when we train PredNet. Do you expect the performance of PredNet to be better or worse? First reason about it, then run the following experiment and compare with what you got in question 1.

```
--rotnet_arch rotnet3_feat3 --prednet_arch prednet0 --rotnet_ckpt_dir
./ckpts/rot3_feat3_pred0_no_grad/rotnet --prednet_ckpt_dir
./ckpts/rot3_feat3_pred0_no_grad/prednet --no_grad
```

Repeat question 2, but disallow the gradient to flow through RotNet.

```
--rotnet_arch rotnet3_feat2 --prednet_arch prednet0 --rotnet_ckpt_dir
./ckpts/rot3_feat2_pred0_no_grad/rotnet --prednet_ckpt_dir
./ckpts/rot3_feat2_pred0_no_grad/prednet --no_grad
```

```
--rotnet_arch rotnet3_feat1 --prednet_arch prednet0 --rotnet_ckpt_dir
./ckpts/rot3_feat1_pred0_no_grad/rotnet --prednet_ckpt_dir
./ckpts/rot3_feat1_pred0_no_grad/prednet --no_grad
```

Until now, our PredNet only has a single fully connected layer for classifying the images. Add three convolutional layers followed by batch normalization and relu activation before applying the original fully connected layer.