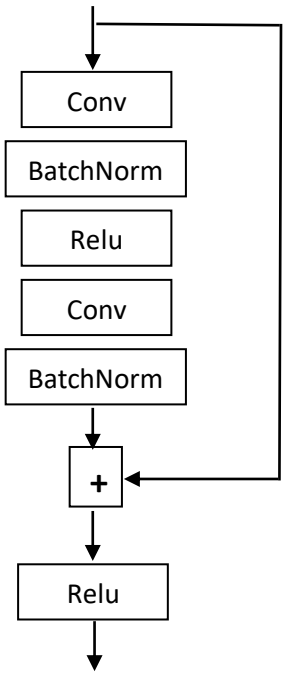
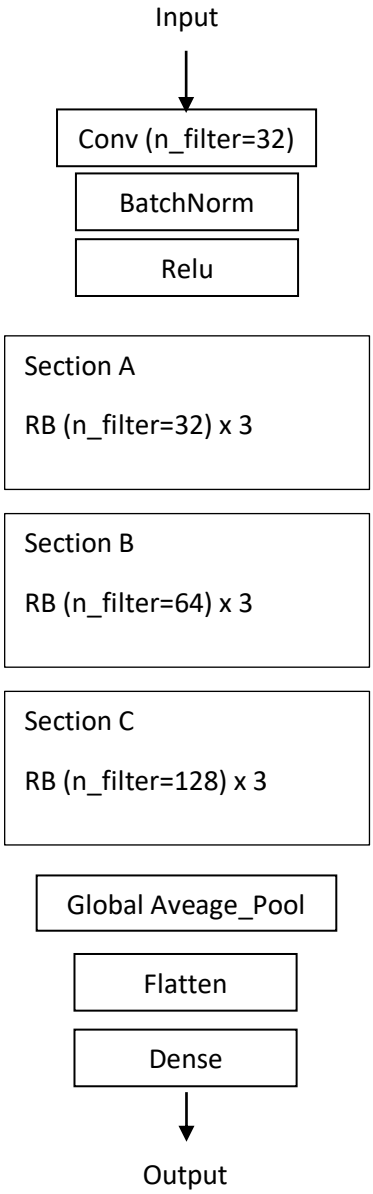
CSC 4343 Homework 1

Complete the code in the **ResModel.py** file to implement the residual network specified in the "ResNet Structure" diagram. Sections A, B, and C consist of multiple residual blocks (RBs) as shown in the "residual block" diagram. All convolutional layers use kernel size = 3, stride = 1 and padding = 'same', except the following: Section B, RB0, conv0, and Section C, RB0, conv 0. These two conv layers use stride = 2.

Residual Block (RB):



ResNet Structure



Note that RB0 of sections B and C changes the feature map size and the number of channels. You cannot simply add the result tensor with the skip tensor as they have different shapes. You can use a (1x1) conv layer (no BN or activation) to change the shape of the skip tensor to match the shape of the result tensor.

The overall structure is already implemented in the **ResModel** class in the **ResModel.py** file. You should add the following to the **ResModel.py** file but should not change any other part of the class ResModel:

- Implement **ResSect** class
- Implement **load_trained_model** member function of class ResModel
- Write additional code to train your model on the cifar10 dataset. (Make sure we can import ResModel class from the file **without running your training code**). Save the trained model after training.

Homework Submission:

Upload the modified **ResModel.py** Python file **with your code** to moodle. (Do not submit a .ipynb file.) We will test your model on the cifar10 dataset.

In the function **load_trained_model**, your code should not train the model from scratch. Rather, the code should download a saved trained model and load it from the downloaded file.

Saved Model:

You should not upload the file of the trained model to moodle. Instead, you should share it in your google drive (or other online storage which can be *shared by link*) and have code in **load_trained_model** to download it. Before submission, make sure people other than yourself can run the code and download the model file when calling the function.