Question 1

Prepare a pre-trained resnet18 model :

Step 1: Load the pre-trained model `resnet18` Set the parameter `pretrained` to true:

In [9]:

```
# Step 1: Load the pre-trained model resnet18
import torchvision.models as models

model = models.resnet18(pretrained = True)
```

```
Downloading: "https://download.pytorch.org/models/resnet18-5c106cde.pth"
to /home/dsxuser/.cache/torch/checkpoints/resnet18-5c106cde.pth

HBox(children=(IntProgress(value=0, max=46827520), HTML(value='')))
```

Step 2: Set the attribute `requires_grad` to `False` . As a result, the parameters will not be affected by training.

In [11]:

```
# Step 2: Set the parameter cannot be trained for the pre-trained model
for param in model.parameters():
    param.requires_grad = False
```

`resnet18` is used to classify 1000 different objects; as a result, the last layer has 1000 outputs. The 512 inputs come from the fact that the previously hidden layer has 512 outputs.

Step 3: Replace the output layer `model.fc` of the neural network with a `nn.Linear` object, to classify 2 different classes. For the parameters `in_features` remember the last hidden layer has 512 neurons.

In [12]:

```
d_hidden = 512
d_out = 2
model.fc = nn.Linear(d_hidden, d_out)
```

Print out the model in order to show whether you get the correct answer.
(Your peer reviewer is going to mark based on what you print here.)

In [13]:

```
print(model)
```

```
ResNet(
  (conv1): Conv2d(3, 64, kernel_size=(7, 7), stride=(2, 2), padding=(3,
3), bias=False)
  (bn1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_runn
ing_stats=True)
  (relu): ReLU(inplace=True)
  (maxpool): MaxPool2d(kernel_size=3, stride=2, padding=1, dilation=1, ce
il_mode=False)
  (layer1): Sequential(
    (0): BasicBlock(
      (conv1): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=
(1, 1), bias=False)
      (bn1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_
running_stats=True)
      (relu): ReLU(inplace=True)
      (conv2): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=
(1, 1), bias=False)
      (bn2): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_
running_stats=True)
```