

Machine Learning, Spring 2020

DL Project One – Image Classifier

Python tutorial: <http://learnpython.org/>

TensorFlow tutorial: <https://www.tensorflow.org/tutorials/>

PyTorch tutorial: <https://pytorch.org/tutorials/>

Image classifier

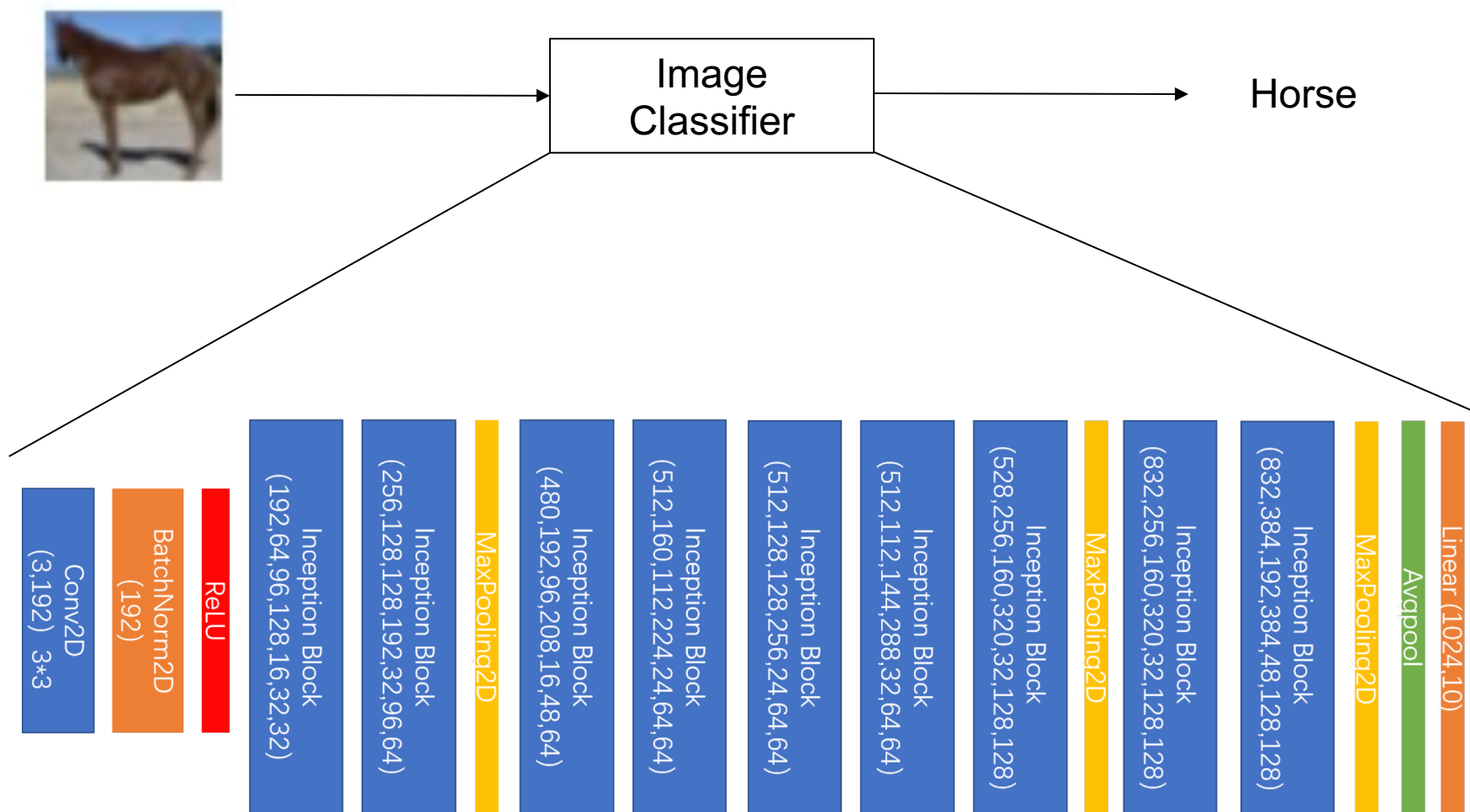
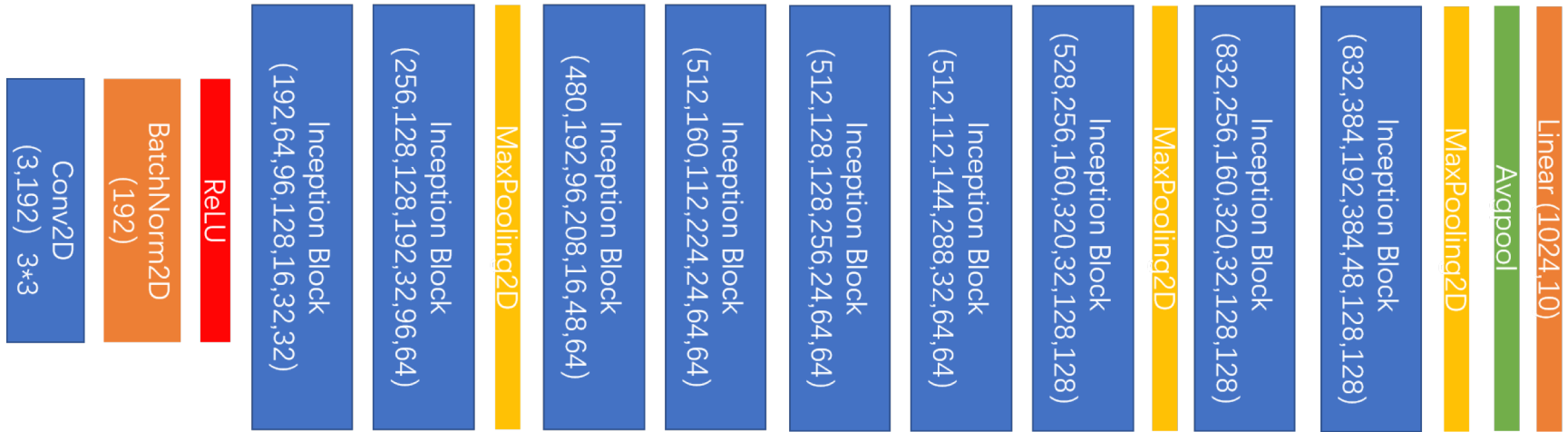
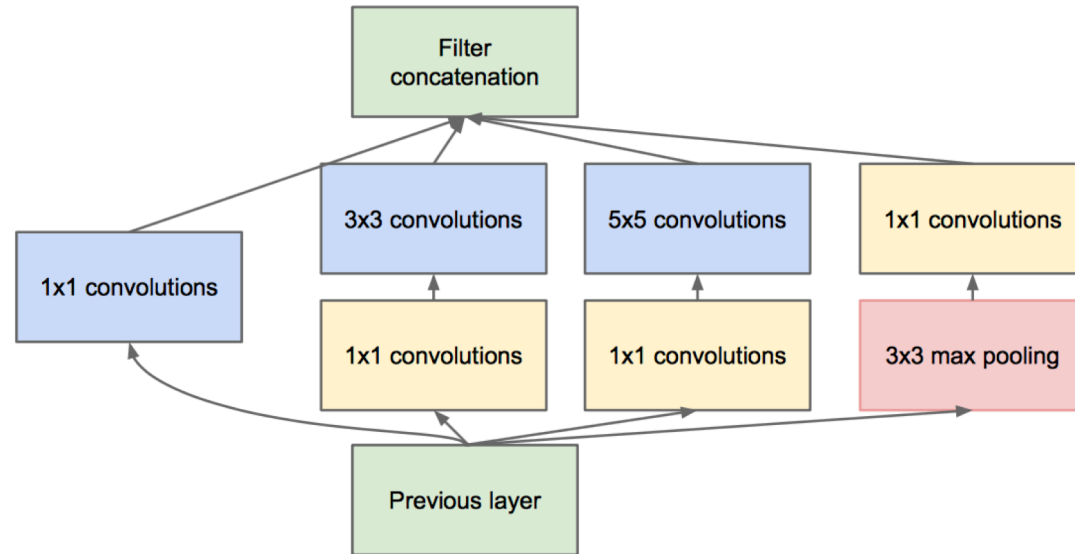


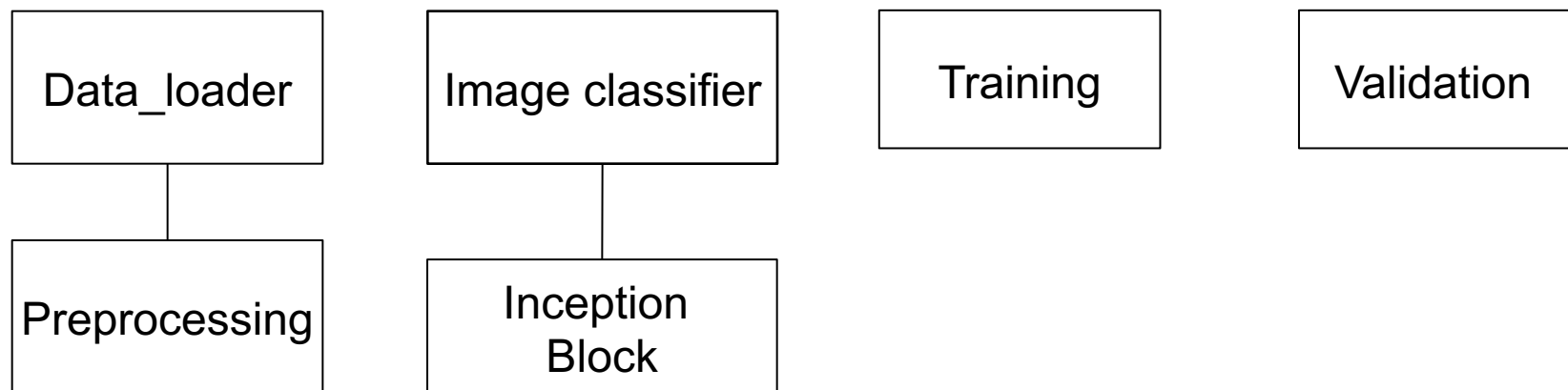
Image classifier



Structure for
each inception
block:

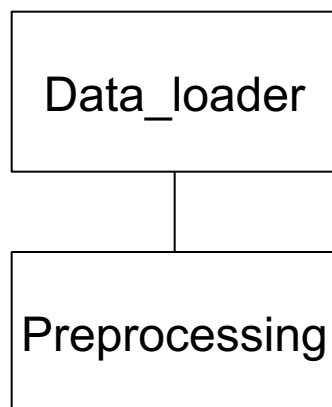


Main Modules for Image Classifier

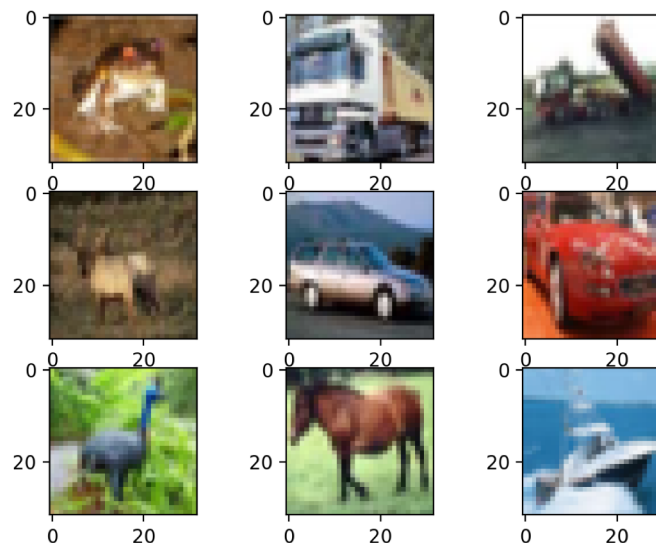


Main Modules for Image Classifier

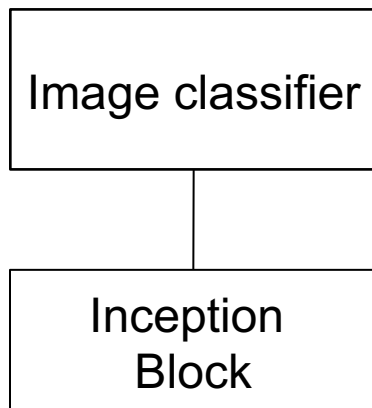
- Download CIFAR10 dataset using `torchvision.datasets.CIFAR10()`
- For each image (preprocessing):
 - Converts images into numbers
 - Random flip some training image using `RandomHorizontalFlip()`
- * Feel free to try other settings
- Obtain the training set and testing set using `torch.utils.data.DataLoader`



Example images
from CIFAR10 :



Main Modules for Image Classifier



```

In [ ]: class ImageClassifier(nn.Module):
        def __init__(self):

        def forward(self, x):
            #the network structure
            return output
  
```

```

In [ ]: class Inception(nn.Module):
        def __init__(self, in_planes, kernel_1_x, kernel_3_in, kernel_3_x, kernel_5_in, kernel_5_x,
            super(Inception, self).__init__()
            # 1x1 conv branch
            self.b1 = nn.Sequential(...)

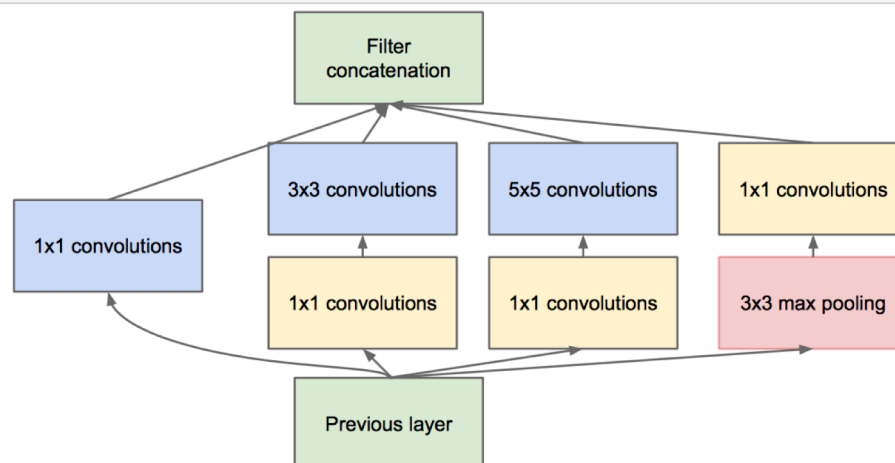
            # 1x1 conv -> 3x3 conv branch
            self.b2 = nn.Sequential(...)

            # 1x1 conv -> 5x5 conv branch
            self.b3 = nn.Sequential(...)

            # 3x3 pool -> 1x1 conv branch
            self.b4 = nn.Sequential(...)

        def forward(self, x):

            return torch.cat([...], 1)
  
```



Main Modules for Image Classifier

Training

- Define optimizer
- For each epoch
 - get the category prediction using *ImageClassifier*
 - Compute the Negative Log Likelihood loss and update the parameters

$$NLL(D, \mathbf{w}) = - \sum_{i=1}^N [(1 - y_i) \log(1 - \sigma(\mathbf{w}^T \mathbf{x}_i)) + y_i \log \sigma(\mathbf{w}^T \mathbf{x}_i)]$$

- save the model with updated parameters
- Plot the training losses

Validation

- Load the pretrained image classifier
- Feed the testing images into the classifier
- Get the category prediction
- Calculate the **accuracy**

Pipeline for Image Classifier

