ExtraCreditMP

May 9, 2022

[116]:

```
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.datasets import cifar10
from tensorflow.keras.layers import Conv2D
from tensorflow.keras.layers import BatchNormalization
from tensorflow.keras.layers import Activation
from tensorflow.keras.layers import Flatten
from tensorflow.keras.layers import Dropout
from tensorflow.keras.layers import Dense
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import MaxPooling2D
from tensorflow.keras.losses import *
import torch.nn
import torchvision.models as models
import tensorflow as tf
import matplotlib.pyplot as plt
from random import *
import numpy as np
```

[100]:

```
(x_vals, y_vals), (x_test, y_test) = cifar10.load_data()
height = 32
width = 32
channels = 3
nb = 10
clamp = 255
labelsCIFAR = ['airplane', 'automobile', 'bird', 'cat', 'deer', 'dog', 'frog',␣
 ↪'horse', 'ship', 'truck']
x_vals = x_vals / clamp
x_test = x_test / clamp
x_vals = x_vals.reshape((-1, height, width, channels))
x_test = x_test.reshape((-1, height, width, channels))
y_vals = tf.keras.utils.to_categorical(y_vals, nb)
y_test = tf.keras.utils.to_categorical(y_test, nb)
print(x_vals, x_test, y_vals, y_test)
```

```
[[[[0.23137255 0.24313725 0.24705882]
   [0.16862745 0.18039216 0.17647059]
   [0.19607843 0.18823529 0.16862745]
   …
```

```
  [0.61960784 0.51764706 0.42352941]
  [0.59607843 0.49019608 0.4       ]
  [0.58039216 0.48627451 0.40392157]]

 [[0.0627451  0.07843137 0.07843137]
  [0.         0.         0.        ]
  [0.07058824 0.03137255 0.        ]
  …
  [0.48235294 0.34509804 0.21568627]
  [0.46666667 0.3254902  0.19607843]
  [0.47843137 0.34117647 0.22352941]]

 [[0.09803922 0.09411765 0.08235294]
  [0.0627451  0.02745098 0.        ]
  [0.19215686 0.10588235 0.03137255]
  …
  [0.4627451  0.32941176 0.19607843]
  [0.47058824 0.32941176 0.19607843]
  [0.42745098 0.28627451 0.16470588]]

 …

 [[0.81568627 0.66666667 0.37647059]
  [0.78823529 0.6        0.13333333]
  [0.77647059 0.63137255 0.10196078]
  …
  [0.62745098 0.52156863 0.2745098 ]
  [0.21960784 0.12156863 0.02745098]
  [0.20784314 0.13333333 0.07843137]]

 [[0.70588235 0.54509804 0.37647059]
  [0.67843137 0.48235294 0.16470588]
  [0.72941176 0.56470588 0.11764706]
  …
  [0.72156863 0.58039216 0.36862745]
  [0.38039216 0.24313725 0.13333333]
  [0.3254902  0.20784314 0.13333333]]

 [[0.69411765 0.56470588 0.45490196]
  [0.65882353 0.50588235 0.36862745]
  [0.70196078 0.55686275 0.34117647]
  …
  [0.84705882 0.72156863 0.54901961]
  [0.59215686 0.4627451  0.32941176]
  [0.48235294 0.36078431 0.28235294]]]


[[[0.60392157 0.69411765 0.73333333]
```

```
 [0.49411765 0.5372549  0.53333333]
 [0.41176471 0.40784314 0.37254902]
 …
 [0.35686275 0.37254902 0.27843137]
 [0.34117647 0.35294118 0.27843137]
 [0.30980392 0.31764706 0.2745098 ]]

[[0.54901961 0.62745098 0.6627451 ]
 [0.56862745 0.6        0.60392157]
 [0.49019608 0.49019608 0.4627451 ]
 …
 [0.37647059 0.38823529 0.30588235]
 [0.30196078 0.31372549 0.24313725]
 [0.27843137 0.28627451 0.23921569]]

[[0.54901961 0.60784314 0.64313725]
 [0.54509804 0.57254902 0.58431373]
 [0.45098039 0.45098039 0.43921569]
 …
 [0.30980392 0.32156863 0.25098039]
 [0.26666667 0.2745098  0.21568627]
 [0.2627451  0.27058824 0.21568627]]

…

[[0.68627451 0.65490196 0.65098039]
 [0.61176471 0.60392157 0.62745098]
 [0.60392157 0.62745098 0.66666667]
 …
 [0.16470588 0.13333333 0.14117647]
 [0.23921569 0.20784314 0.22352941]
 [0.36470588 0.3254902  0.35686275]]

[[0.64705882 0.60392157 0.50196078]
 [0.61176471 0.59607843 0.50980392]
 [0.62352941 0.63137255 0.55686275]
 …
 [0.40392157 0.36470588 0.37647059]
 [0.48235294 0.44705882 0.47058824]
 [0.51372549 0.4745098  0.51372549]]

[[0.63921569 0.58039216 0.47058824]
 [0.61960784 0.58039216 0.47843137]
 [0.63921569 0.61176471 0.52156863]
 …
 [0.56078431 0.52156863 0.54509804]
 [0.56078431 0.5254902  0.55686275]
 [0.56078431 0.52156863 0.56470588]]]
```

```
[[[1.         1.         1.        ]
  [0.99215686 0.99215686 0.99215686]
  [0.99215686 0.99215686 0.99215686]
  …
  [0.99215686 0.99215686 0.99215686]
  [0.99215686 0.99215686 0.99215686]
  [0.99215686 0.99215686 0.99215686]]

 [[1.         1.         1.        ]
  [1.         1.         1.        ]
  [1.         1.         1.        ]
  …
  [1.         1.         1.        ]
  [1.         1.         1.        ]
  [1.         1.         1.        ]]

 [[1.         1.         1.        ]
  [0.99607843 0.99607843 0.99607843]
  [0.99607843 0.99607843 0.99607843]
  …
  [0.99607843 0.99607843 0.99607843]
  [0.99607843 0.99607843 0.99607843]
  [0.99607843 0.99607843 0.99607843]]

 …

 [[0.44313725 0.47058824 0.43921569]
  [0.43529412 0.4627451  0.43529412]
  [0.41176471 0.43921569 0.41568627]
  …
  [0.28235294 0.31764706 0.31372549]
  [0.28235294 0.31372549 0.30980392]
  [0.28235294 0.31372549 0.30980392]]

 [[0.43529412 0.4627451  0.43137255]
  [0.40784314 0.43529412 0.40784314]
  [0.38823529 0.41568627 0.38431373]
  …
  [0.26666667 0.29411765 0.28627451]
  [0.2745098  0.29803922 0.29411765]
  [0.30588235 0.32941176 0.32156863]]

 [[0.41568627 0.44313725 0.41176471]
  [0.38823529 0.41568627 0.38431373]
  [0.37254902 0.4        0.36862745]
  …
```

```
   [0.30588235 0.33333333 0.3254902 ]
   [0.30980392 0.33333333 0.3254902 ]
   [0.31372549 0.3372549  0.32941176]]]


…


[[[0.1372549  0.69803922 0.92156863]
   [0.15686275 0.69019608 0.9372549 ]
   [0.16470588 0.69019608 0.94509804]
   …
   [0.38823529 0.69411765 0.85882353]
   [0.30980392 0.57647059 0.77254902]
   [0.34901961 0.58039216 0.74117647]]

 [[0.22352941 0.71372549 0.91764706]
   [0.17254902 0.72156863 0.98039216]
   [0.19607843 0.71764706 0.94117647]
   …
   [0.61176471 0.71372549 0.78431373]
   [0.55294118 0.69411765 0.80784314]
   [0.45490196 0.58431373 0.68627451]]

 [[0.38431373 0.77254902 0.92941176]
   [0.25098039 0.74117647 0.98823529]
   [0.27058824 0.75294118 0.96078431]
   …
   [0.7372549  0.76470588 0.80784314]
   [0.46666667 0.52941176 0.57647059]
   [0.23921569 0.30980392 0.35294118]]

 …

 [[0.28627451 0.30980392 0.30196078]
   [0.20784314 0.24705882 0.26666667]
   [0.21176471 0.26666667 0.31372549]
   …
   [0.06666667 0.15686275 0.25098039]
   [0.08235294 0.14117647 0.2       ]
   [0.12941176 0.18823529 0.19215686]]

 [[0.23921569 0.26666667 0.29411765]
   [0.21568627 0.2745098  0.3372549 ]
   [0.22352941 0.30980392 0.40392157]
   …
   [0.09411765 0.18823529 0.28235294]
   [0.06666667 0.1372549  0.20784314]
```

```
  [0.02745098 0.09019608 0.1254902 ]]

 [[0.17254902 0.21960784 0.28627451]
  [0.18039216 0.25882353 0.34509804]
  [0.19215686 0.30196078 0.41176471]
  …
  [0.10588235 0.20392157 0.30196078]
  [0.08235294 0.16862745 0.25882353]
  [0.04705882 0.12156863 0.19607843]]]


[[[0.74117647 0.82745098 0.94117647]
  [0.72941176 0.81568627 0.9254902 ]
  [0.7254902  0.81176471 0.92156863]
  …
  [0.68627451 0.76470588 0.87843137]
  [0.6745098  0.76078431 0.87058824]
  [0.6627451  0.76078431 0.8627451 ]]

 [[0.76078431 0.82352941 0.9372549 ]
  [0.74901961 0.81176471 0.9254902 ]
  [0.74509804 0.80784314 0.92156863]
  …
  [0.67843137 0.75294118 0.8627451 ]
  [0.67058824 0.74901961 0.85490196]
  [0.65490196 0.74509804 0.84705882]]

 [[0.81568627 0.85882353 0.95686275]
  [0.80392157 0.84705882 0.94117647]
  [0.8        0.84313725 0.9372549 ]
  …
  [0.68627451 0.74901961 0.85098039]
  [0.6745098  0.74509804 0.84705882]
  [0.6627451  0.74901961 0.84313725]]

 …

 [[0.81176471 0.78039216 0.70980392]
  [0.79607843 0.76470588 0.68627451]
  [0.79607843 0.76862745 0.67843137]
  …
  [0.52941176 0.51764706 0.49803922]
  [0.63529412 0.61960784 0.58823529]
  [0.65882353 0.63921569 0.59215686]]

 [[0.77647059 0.74509804 0.66666667]
  [0.74117647 0.70980392 0.62352941]
  [0.70588235 0.6745098  0.57647059]
  …
  [0.69803922 0.67058824 0.62745098]
  [0.68627451 0.6627451  0.61176471]
  [0.68627451 0.6627451  0.60392157]]

 [[0.77647059 0.74117647 0.67843137]
  [0.74117647 0.70980392 0.63529412]
  [0.69803922 0.66666667 0.58431373]
  …
  [0.76470588 0.72156863 0.6627451 ]
  [0.76862745 0.74117647 0.67058824]
  [0.76470588 0.74509804 0.67058824]]]


[[[0.89803922 0.89803922 0.9372549 ]
  [0.9254902  0.92941176 0.96862745]
  [0.91764706 0.9254902  0.96862745]
  …
  [0.85098039 0.85882353 0.91372549]
  [0.86666667 0.8745098  0.91764706]
  [0.87058824 0.8745098  0.91372549]]

 [[0.87058824 0.86666667 0.89803922]
  [0.9372549  0.9372549  0.97647059]
  [0.91372549 0.91764706 0.96470588]
  …
  [0.8745098  0.8745098  0.9254902 ]
  [0.89019608 0.89411765 0.93333333]
  [0.82352941 0.82745098 0.8627451 ]]

 [[0.83529412 0.80784314 0.82745098]
  [0.91764706 0.90980392 0.9372549 ]
  [0.90588235 0.91372549 0.95686275]
  …
  [0.8627451  0.8627451  0.90980392]
  [0.8627451  0.85882353 0.90980392]
  [0.79215686 0.79607843 0.84313725]]

 …

 [[0.58823529 0.56078431 0.52941176]
  [0.54901961 0.52941176 0.49803922]
  [0.51764706 0.49803922 0.47058824]
  …
  [0.87843137 0.87058824 0.85490196]
  [0.90196078 0.89411765 0.88235294]
  [0.94509804 0.94509804 0.93333333]]
```

```
[[0.5372549  0.51764706 0.49411765]
 [0.50980392 0.49803922 0.47058824]
 [0.49019608 0.4745098  0.45098039]
 …
 [0.70980392 0.70588235 0.69803922]
 [0.79215686 0.78823529 0.77647059]
 [0.83137255 0.82745098 0.81176471]]

[[0.47843137 0.46666667 0.44705882]
 [0.4627451  0.45490196 0.43137255]
 [0.47058824 0.45490196 0.43529412]
 …
 [0.70196078 0.69411765 0.67843137]
 [0.64313725 0.64313725 0.63529412]
 [0.63921569 0.63921569 0.63137255]]]] [[[[0.61960784 0.43921569 0.19215686]
 [0.62352941 0.43529412 0.18431373]
 [0.64705882 0.45490196 0.2       ]
 …
 [0.5372549  0.37254902 0.14117647]
 [0.49411765 0.35686275 0.14117647]
 [0.45490196 0.33333333 0.12941176]]

[[0.59607843 0.43921569 0.2       ]
 [0.59215686 0.43137255 0.15686275]
 [0.62352941 0.44705882 0.17647059]
 …
 [0.53333333 0.37254902 0.12156863]
 [0.49019608 0.35686275 0.1254902 ]
 [0.46666667 0.34509804 0.13333333]]

[[0.59215686 0.43137255 0.18431373]
 [0.59215686 0.42745098 0.12941176]
 [0.61960784 0.43529412 0.14117647]
 …
 [0.54509804 0.38431373 0.13333333]
 [0.50980392 0.37254902 0.13333333]
 [0.47058824 0.34901961 0.12941176]]


…

[[0.26666667 0.48627451 0.69411765]
 [0.16470588 0.39215686 0.58039216]
 [0.12156863 0.34509804 0.5372549 ]
 …
 [0.14901961 0.38039216 0.57254902]
 [0.05098039 0.25098039 0.42352941]
 [0.15686275 0.33333333 0.49803922]]
 [[0.23921569 0.45490196 0.65882353]
  [0.19215686 0.4        0.58039216]
  [0.1372549  0.33333333 0.51764706]
  …
  [0.10196078 0.32156863 0.50980392]
  [0.11372549 0.32156863 0.49411765]
  [0.07843137 0.25098039 0.41960784]]

 [[0.21176471 0.41960784 0.62745098]
  [0.21960784 0.41176471 0.58431373]
  [0.17647059 0.34901961 0.51764706]
  …
  [0.09411765 0.30196078 0.48627451]
  [0.13333333 0.32941176 0.50588235]
  [0.08235294 0.2627451  0.43137255]]]


[[[0.92156863 0.92156863 0.92156863]
  [0.90588235 0.90588235 0.90588235]
  [0.90980392 0.90980392 0.90980392]
  …
  [0.91372549 0.91372549 0.91372549]
  [0.91372549 0.91372549 0.91372549]
  [0.90980392 0.90980392 0.90980392]]

 [[0.93333333 0.93333333 0.93333333]
  [0.92156863 0.92156863 0.92156863]
  [0.92156863 0.92156863 0.92156863]
  …
  [0.9254902  0.9254902  0.9254902 ]
  [0.9254902  0.9254902  0.9254902 ]
  [0.92156863 0.92156863 0.92156863]]

 [[0.92941176 0.92941176 0.92941176]
  [0.91764706 0.91764706 0.91764706]
  [0.91764706 0.91764706 0.91764706]
  …
  [0.92156863 0.92156863 0.92156863]
  [0.92156863 0.92156863 0.92156863]
  [0.91764706 0.91764706 0.91764706]]

 …

 [[0.34117647 0.38823529 0.34901961]
  [0.16862745 0.2        0.14509804]
  [0.0745098  0.09019608 0.04313725]
  …
  [0.6627451  0.72156863 0.70196078]
  [0.71372549 0.77254902 0.75686275]
  [0.7372549  0.79215686 0.78823529]]

 [[0.32156863 0.37647059 0.32156863]
  [0.18039216 0.22352941 0.14117647]
  [0.14117647 0.17254902 0.08627451]
  …
  [0.68235294 0.74117647 0.71764706]
  [0.7254902  0.78431373 0.76862745]
  [0.73333333 0.79215686 0.78431373]]

 [[0.33333333 0.39607843 0.3254902 ]
  [0.24313725 0.29411765 0.18823529]
  [0.22745098 0.2627451  0.14901961]
  …
  [0.65882353 0.71764706 0.69803922]
  [0.70588235 0.76470588 0.74901961]
  [0.72941176 0.78431373 0.78039216]]]


[[[0.61960784 0.74509804 0.87058824]
  [0.61960784 0.73333333 0.85490196]
  [0.54509804 0.65098039 0.76078431]
  …
  [0.89411765 0.90588235 0.91764706]
  [0.92941176 0.9372549  0.95294118]
  [0.93333333 0.94509804 0.96470588]]

 [[0.66666667 0.78431373 0.89803922]
  [0.6745098  0.78039216 0.88627451]
  [0.59215686 0.69019608 0.78823529]
  …
  [0.90980392 0.90980392 0.9254902 ]
  [0.96470588 0.96470588 0.98039216]
  [0.96470588 0.96862745 0.98431373]]

 [[0.68235294 0.78823529 0.88235294]
  [0.69019608 0.78431373 0.87058824]
  [0.61568627 0.70196078 0.78039216]
  …
  [0.90196078 0.89803922 0.90980392]
  [0.98039216 0.97647059 0.98431373]
  [0.96078431 0.95686275 0.96862745]]

 …

 [[0.12156863 0.15686275 0.17647059]
  [0.11764706 0.15294118 0.17254902]
  [0.10196078 0.1372549  0.15686275]
  …
  [0.14509804 0.15686275 0.18039216]
  [0.03529412 0.05098039 0.05490196]
  [0.01568627 0.02745098 0.01960784]]

 [[0.09019608 0.13333333 0.15294118]
  [0.10588235 0.14901961 0.16862745]
  [0.09803922 0.14117647 0.16078431]
  …
  [0.0745098  0.07843137 0.09411765]
  [0.01568627 0.02352941 0.01176471]
  [0.01960784 0.02745098 0.01176471]]

 [[0.10980392 0.16078431 0.18431373]
  [0.11764706 0.16862745 0.19607843]
  [0.1254902  0.17647059 0.20392157]
  …
  [0.01960784 0.02352941 0.03137255]
  [0.01568627 0.01960784 0.01176471]
  [0.02745098 0.03137255 0.02745098]]]


…


[[[0.07843137 0.05882353 0.04705882]
  [0.0745098  0.05490196 0.04313725]
  [0.05882353 0.05490196 0.04313725]
  …
  [0.03921569 0.03529412 0.02745098]
  [0.04705882 0.04313725 0.03529412]
  [0.05098039 0.04705882 0.03921569]]

 [[0.08235294 0.0627451  0.05098039]
  [0.07843137 0.0627451  0.05098039]
  [0.07058824 0.06666667 0.04705882]
  …
  [0.03921569 0.03529412 0.02745098]
  [0.03921569 0.03529412 0.02745098]
  [0.04705882 0.04313725 0.03529412]]

 [[0.08235294 0.0627451  0.05098039]
  [0.08235294 0.06666667 0.04705882]
  [0.07843137 0.07058824 0.04313725]
  …
  [0.04705882 0.04313725 0.03529412]
  [0.04705882 0.04313725 0.03529412]
  [0.05098039 0.04705882 0.03921569]]

 …

 [[0.12941176 0.09803922 0.05098039]
  [0.13333333 0.10196078 0.05882353]
  [0.13333333 0.10196078 0.05882353]
  …
  [0.10980392 0.09803922 0.20392157]
  [0.11372549 0.09803922 0.22745098]
  [0.09019608 0.07843137 0.16470588]]

 [[0.12941176 0.09803922 0.05490196]
  [0.13333333 0.10196078 0.05882353]
  [0.13333333 0.10196078 0.05882353]
  …
  [0.10588235 0.09411765 0.20392157]
  [0.10588235 0.09411765 0.21960784]
  [0.09803922 0.08627451 0.18431373]]

 [[0.12156863 0.09019608 0.04705882]
  [0.1254902  0.09411765 0.05098039]
  [0.12941176 0.09803922 0.05490196]
  …
  [0.09411765 0.09019608 0.19607843]
  [0.10196078 0.09019608 0.20784314]
  [0.09803922 0.07843137 0.18431373]]]


[[[0.09803922 0.15686275 0.04705882]
  [0.05882353 0.14117647 0.01176471]
  [0.09019608 0.16078431 0.07058824]
  …
  [0.23921569 0.32156863 0.30588235]
  [0.36078431 0.44313725 0.43921569]
  [0.29411765 0.34901961 0.36078431]]

 [[0.04705882 0.09803922 0.02352941]
  [0.07843137 0.14509804 0.02745098]
  [0.09411765 0.14117647 0.05882353]
  …
  [0.45098039 0.5254902  0.54117647]
  [0.58431373 0.65882353 0.69411765]
  [0.40784314 0.45882353 0.51372549]]

 [[0.04705882 0.09803922 0.04313725]
  [0.05882353 0.11372549 0.02352941]
  [0.13333333 0.15686275 0.09411765]
  …
  [0.60392157 0.6745098  0.71372549]
  [0.61568627 0.68627451 0.75294118]
  [0.45490196 0.50588235 0.59215686]]

 …

 [[0.39215686 0.50588235 0.31764706]
  [0.40392157 0.51764706 0.32941176]
  [0.40784314 0.5254902  0.3372549 ]
  …
  [0.38039216 0.50196078 0.32941176]
  [0.38431373 0.49411765 0.32941176]
  [0.35686275 0.4745098  0.30980392]]

 [[0.40392157 0.51764706 0.3254902 ]
  [0.40784314 0.51372549 0.3254902 ]
  [0.41960784 0.52941176 0.34117647]
  …
  [0.39607843 0.51764706 0.34117647]
  [0.38823529 0.49803922 0.32941176]
  [0.36078431 0.4745098  0.30980392]]

 [[0.37254902 0.49411765 0.30588235]
  [0.37254902 0.48235294 0.29803922]
  [0.39607843 0.50196078 0.31764706]
  …
  [0.36470588 0.48627451 0.31372549]
  [0.37254902 0.48235294 0.31764706]
  [0.36078431 0.47058824 0.31372549]]]


[[[0.28627451 0.30588235 0.29411765]
  [0.38431373 0.40392157 0.44313725]
  [0.38823529 0.41568627 0.44705882]
  …
  [0.52941176 0.58823529 0.59607843]
  [0.52941176 0.58431373 0.60392157]
  [0.79607843 0.84313725 0.8745098 ]]

 [[0.27058824 0.28627451 0.2745098 ]
  [0.32941176 0.34901961 0.38039216]
  [0.26666667 0.29411765 0.31764706]
  …
  [0.33333333 0.37254902 0.34901961]
  [0.27843137 0.32156863 0.31372549]
  [0.47058824 0.52156863 0.52941176]]
 [[0.27058824 0.28627451 0.2745098 ]
  [0.35294118 0.37254902 0.39215686]
  [0.24313725 0.27843137 0.29019608]
  …
  [0.29019608 0.31764706 0.2745098 ]
  [0.20784314 0.24313725 0.21176471]
  [0.24313725 0.29019608 0.27058824]]

 …

 [[0.48235294 0.50196078 0.37647059]
  [0.51764706 0.51764706 0.4       ]
  [0.50588235 0.50196078 0.39215686]
  …
  [0.42352941 0.41960784 0.34509804]
  [0.24313725 0.23529412 0.21568627]
  [0.10588235 0.10588235 0.10980392]]

 [[0.45098039 0.4745098  0.35686275]
  [0.48235294 0.48627451 0.37254902]
  [0.50588235 0.49411765 0.38823529]
  …
  [0.45098039 0.45490196 0.36862745]
  [0.25882353 0.25490196 0.23137255]
  [0.10588235 0.10588235 0.10588235]]

 [[0.45490196 0.47058824 0.35294118]
  [0.4745098  0.47843137 0.36862745]
  [0.50588235 0.50196078 0.39607843]
  …
  [0.45490196 0.45098039 0.36862745]
  [0.26666667 0.25490196 0.22745098]
  [0.10588235 0.10196078 0.10196078]]]] [[0. 0. 0. … 0. 0. 0.]
[0. 0. 0. … 0. 0. 1.]
[0. 0. 0. … 0. 0. 1.]
…
[0. 0. 0. … 0. 0. 1.]
[0. 1. 0. … 0. 0. 0.]
[0. 1. 0. … 0. 0. 0.]] [[0. 0. 0. … 0. 0. 0.]
[0. 0. 0. … 0. 1. 0.]
[0. 0. 0. … 0. 1. 0.]
…
[0. 0. 0. … 0. 0. 0.]
[0. 1. 0. … 0. 0. 0.]
[0. 0. 0. … 1. 0. 0.]]
```

[120]:

```
dimensionVal = -1
x = Sequential()
x.add(Conv2D(128, (3, 3), strides = (2, 2), padding = "same", input_shape =␣
 ↪(height, width, channels)))
x.add(Activation("relu"))
x.add(Dropout(0.1))

x.add(Conv2D(64, (3, 3), strides = (2, 2), padding = "same", input_shape =␣
 ↪(height, width, channels)))
x.add(Activation("relu"))
x.add(Dropout(0.1))

x.add(Conv2D(64, (3, 3), strides = (2, 2), padding = "same", input_shape =␣
 ↪(height, width, channels)))
x.add(Activation("relu"))
x.add(Dropout(0.1))
x.add(MaxPooling2D(pool_size = (2, 2)))

x.add(Conv2D(64, (3, 3), strides = (2, 2), padding = "same", input_shape =␣
 ↪(height, width, channels)))
x.add(Activation("relu"))
x.add(Dropout(0.1))
x.add(MaxPooling2D(pool_size = (2, 2), strides = (2, 2), padding = "same"))

x.add(Conv2D(64, (3, 3), strides = (2, 2), padding = "same", input_shape =␣
 ↪(height, width, channels)))
x.add(Activation("relu"))
x.add(Dropout(0.1))
x.add(MaxPooling2D(pool_size = (2, 2), strides = (2, 2), padding = "same"))

x.add(Dropout(0.1))
x.add(Flatten())

x.add(Dense(10, activation = 'softmax'))
x.compile(optimizer = 'adam', loss = tf.keras.losses.BinaryCrossentropy(),␣
 ↪metrics = ['accuracy'])
```

[121]:

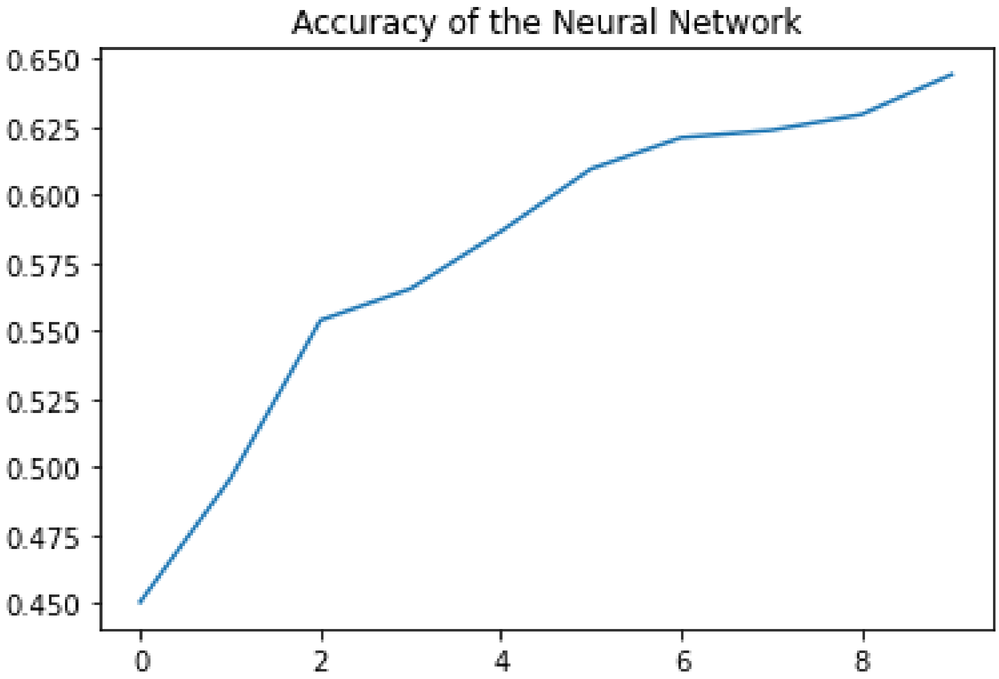
```
history = x.fit(x_vals, y_vals, batch_size = 16, epochs = 10, validation_data =␣
 ↪(x_test, y_test))
```

```
Epoch 1/10
3125/3125 [==============================] - 21s 6ms/step - loss: 0.2803 -
accuracy: 0.3004 - val_loss: 0.2311 - val_accuracy: 0.4505
Epoch 2/10
3125/3125 [==============================] - 20s 6ms/step - loss: 0.2323 -
accuracy: 0.4496 - val_loss: 0.2138 - val_accuracy: 0.4957
Epoch 3/10
```

```
3125/3125 [==============================] - 19s 6ms/step - loss: 0.2128 -
accuracy: 0.5082 - val_loss: 0.1953 - val_accuracy: 0.5539
Epoch 4/10
3125/3125 [==============================] - 19s 6ms/step - loss: 0.2015 -
accuracy: 0.5395 - val_loss: 0.1914 - val_accuracy: 0.5655
Epoch 5/10
3125/3125 [==============================] - 19s 6ms/step - loss: 0.1928 -
accuracy: 0.5642 - val_loss: 0.1813 - val_accuracy: 0.5865
Epoch 6/10
3125/3125 [==============================] - 19s 6ms/step - loss: 0.1854 -
accuracy: 0.5829 - val_loss: 0.1750 - val_accuracy: 0.6095
Epoch 7/10
3125/3125 [==============================] - 19s 6ms/step - loss: 0.1800 -
accuracy: 0.6001 - val_loss: 0.1709 - val_accuracy: 0.6210
Epoch 8/10
3125/3125 [==============================] - 19s 6ms/step - loss: 0.1758 -
accuracy: 0.6114 - val_loss: 0.1670 - val_accuracy: 0.6237
Epoch 9/10
3125/3125 [==============================] - 19s 6ms/step - loss: 0.1715 -
accuracy: 0.6243 - val_loss: 0.1669 - val_accuracy: 0.6295
Epoch 10/10
3125/3125 [==============================] - 19s 6ms/step - loss: 0.1677 -
accuracy: 0.6330 - val_loss: 0.1612 - val_accuracy: 0.6441
```

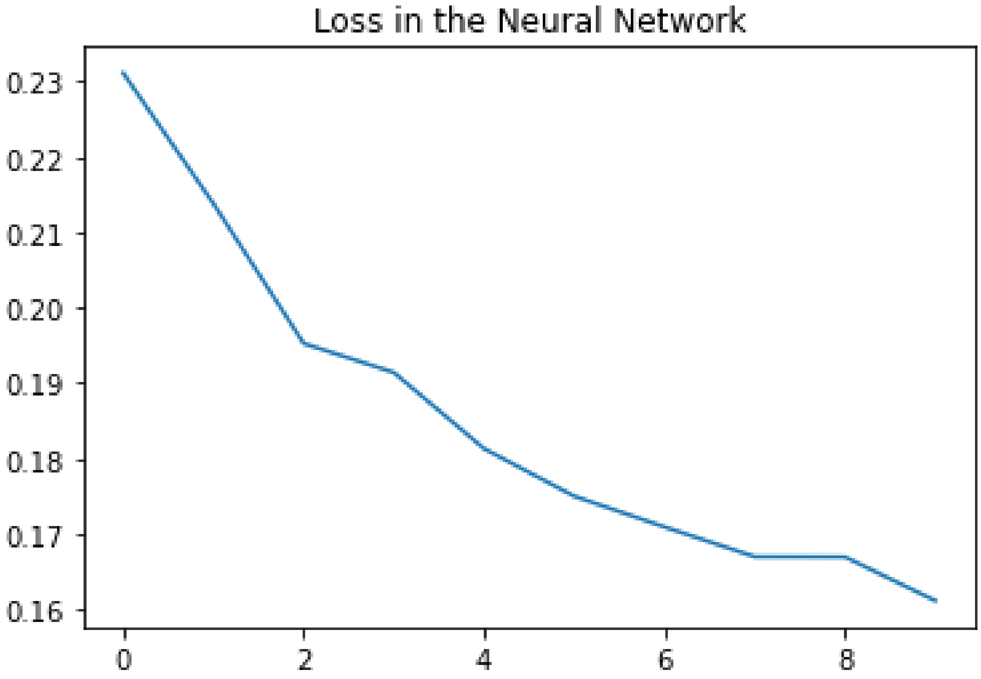
```
[122]: plt.plot(history.history['val_accuracy'], label = 'accuracy')
       plt.title("Accuracy of the Neural Network")
```

```
[122]: Text(0.5, 1.0, 'Accuracy of the Neural Network')
```



```
[123]: plt.plot(history.history['val_loss'], label = 'loss')
       plt.title("Loss in the Neural Network")
```

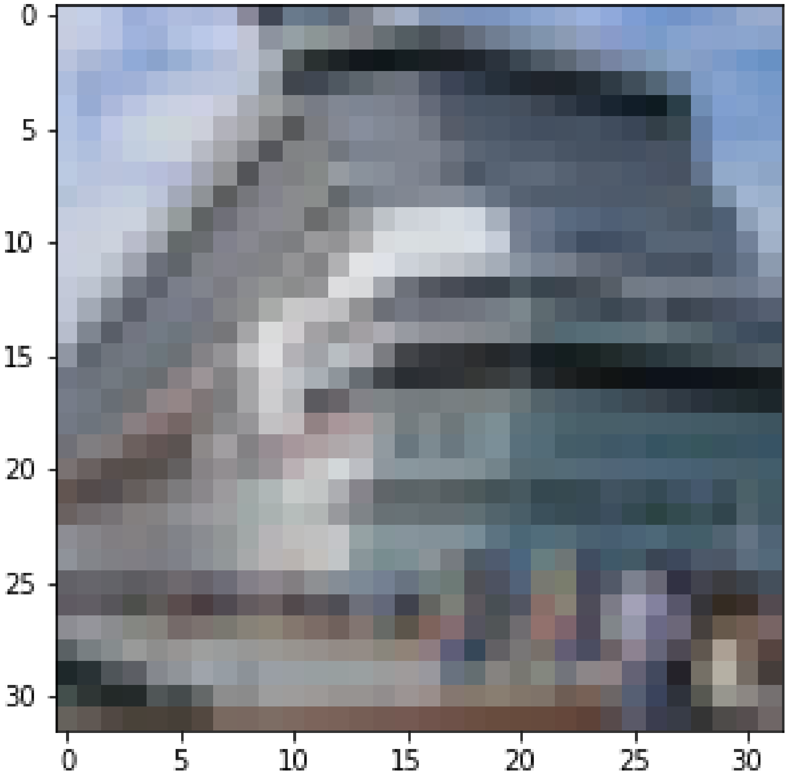
```
[123]: Text(0.5, 1.0, 'Loss in the Neural Network')
```



[133]:

```
randImage = randint(0, 49999)
image = x_vals[randImage].reshape((1, height, width, channels))
label = y_vals[randImage]
print(f"Prediction from CNN: {labelsCIFAR[np.where(label == 1)[0][0]]}")
plt.figure(figsize = (5, 5))
plt.imshow(image.reshape((height, width, channels)))
plt.show()
```

```
Prediction from CNN: ship
```



[134]:

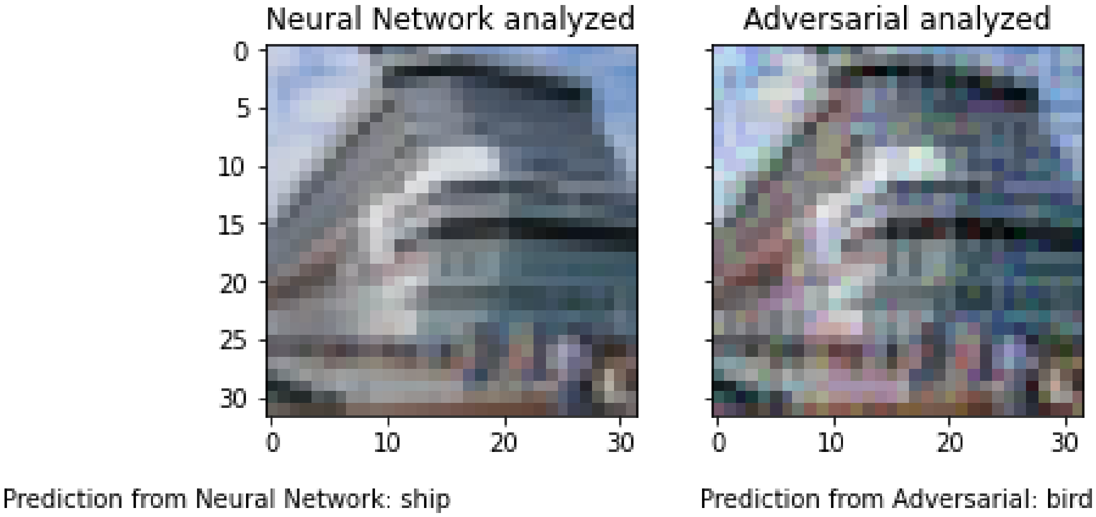
```
def adversary_create(image, label):
    image = tf.cast(image, tf.float32)
    with tf.GradientTape() as t:
        t.watch(image)
        prediction = x(image)
        ls = MSE(label, prediction)
    grad = t.gradient(ls, image)
    signedGrad = tf.sign(grad)
    return signedGrad
```

[135]:

```
noise = adversary_create(image, label).numpy()
ad_picture = image + (noise * 0.05)
```

[136]:

```
randImage = randint(0, 49999)
picture, (nnPic, adPic) = plt.subplots(1, 2, sharey = True)
nnPic.imshow(image.reshape(height, width, channels))
nnPic.set_title("Neural Network analyzed")
nnPic.text(0.5, -0.25, f"Prediction from Neural Network: {labelsCIFAR[x.
 ↪predict(image).argmax()]}", ha = 'right', transform = nnPic.transAxes)
adPic.imshow(ad_picture.reshape(height, width, channels))
adPic.set_title("Adversarial analyzed")
adPic.text(0.5, -0.25, f"Prediction from Adversarial: {labelsCIFAR[x.
 ↪predict(ad_picture).argmax()]}", ha = 'center', transform = adPic.transAxes)
plt.show()
```



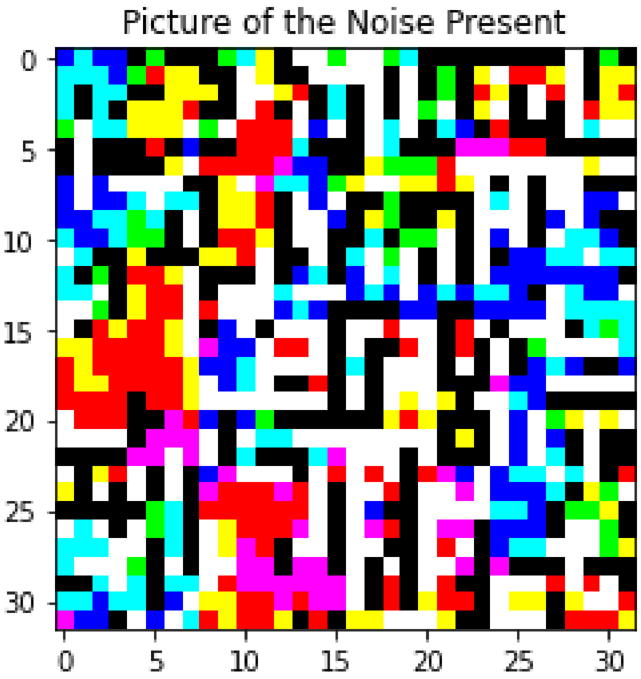
[153]:

```
plt.imshow(noise[0])
plt.title("Picture of the Noise Present")
```

```
Clipping input data to the valid range for imshow with RGB data ([0..1] for
floats or [0..255] for integers).
```

[153]:

```
Text(0.5, 1.0, 'Picture of the Noise Present')
```



[154]:
```
def adversary_generator(batch_size):
    while True:
        arrIm = []
        arrL = []
        for element in range(batch_size):
            rand = randint(0, 49999)
            label = y_vals[rand]
            image = x_vals[rand].reshape((1, height, width, channels))
            arrIm.append(ad_picture)
            arrL.append(label)
        arrIm = np.asarray(arrIm).reshape((batch_size, height, width, channels))
        arrL = np.asarray(arrL)
        yield arrIm, arrL
```

[155]:
```
x_ad, y_ad = next(adversary_generator(5000))
accuracy = x.evaluate(x_ad, y_ad)
print(f"Accuracy of Network after FGSM attack: {accuracy[1]}")
```

```
157/157 [==============================] - 1s 5ms/step - loss: 0.4191 -
accuracy: 0.0964
Accuracy of Network after FGSM attack: 0.09640000015497208
```

[ ]:
```

```