CNN

March 25, 2023

[1]:

```
import matplotlib.pyplot as plt
from tensorflow.keras.models import Sequential, Model
from tensorflow.keras.layers import Dense, Conv2D, Flatten, MaxPooling2D, Activation, Input, GlobalAveragePooling2D
from tensorflow.keras.datasets import cifar10
from tensorflow.keras.preprocessing import image
import warnings
warnings.filterwarnings("always")
%matplotlib inline
```

[2]:

```
(x_train, y_train), _ = cifar10.load_data()
```

[3]:

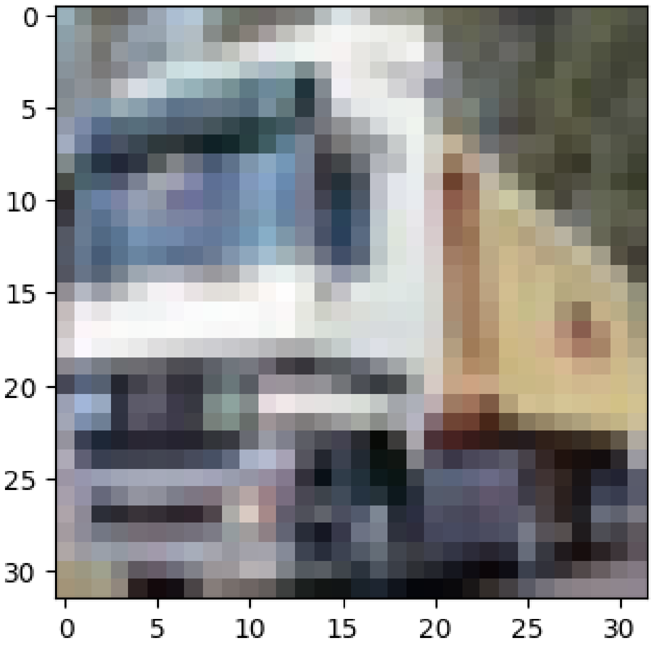
```
x_train[0].shape
```

[3]:

```
(32, 32, 3)
```

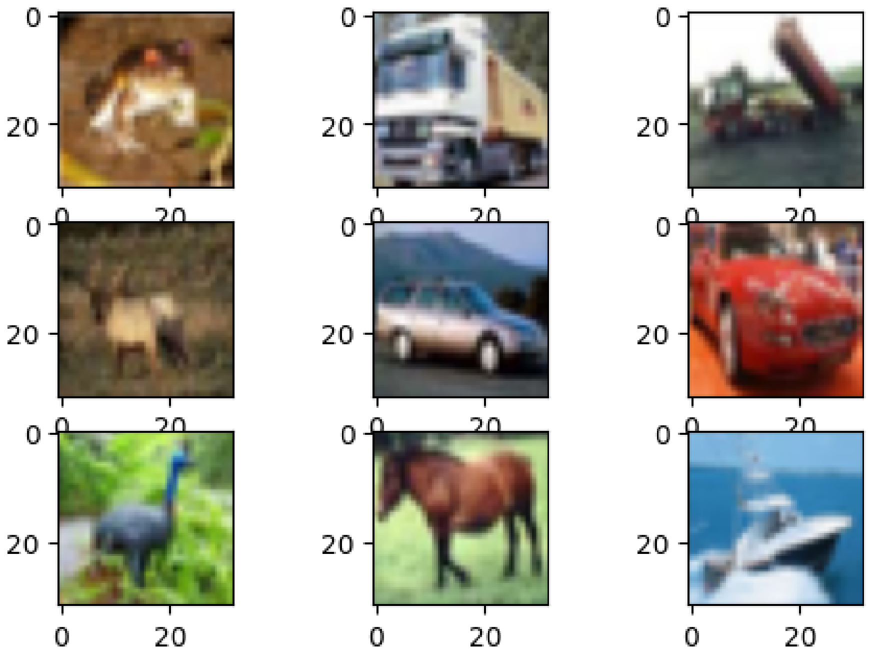
[4]:

```
plt.figure(figsize=(6,4))
plt.imshow(x_train[1])
plt.show()
```



[5]:
```
fig = plt.figure(figsize=(6, 4))
for i in range(9):
    fig.add_subplot(3, 3, i+1)
    plt.imshow(x_train[i])

plt.show()
```



[6]:
```python
model   = Sequential()
model.add(Conv2D(input_shape = (32, 32, 3),
                 filters =10,
                 kernel_size = (3, 3),
                 #strides = (2, 2),
                 padding = "valid"
                ))

model.output_shape
```

[6]:
```
(None, 30, 30, 10)
```

[7]:
```python
model   = Sequential()
model.add(Conv2D(input_shape = (32, 32, 3),
                 filters =10,
                 kernel_size = (3, 3),
                 strides = (1, 1),
                 padding = "same"
                ))

model.output_shape
```

[7]:
```
(None, 32, 32, 10)
```

[8]:

```
model  = Sequential()
model.add(Conv2D(input_shape = (32, 32, 3),
                 filters =10,
                 kernel_size = (3, 3),
                 strides = (1, 1),
                 padding = "same"
                ))

model.output_shape
```

[8]:

```
(None, 32, 32, 10)
```

[9]:

```
model  = Sequential()
model.add(Conv2D(input_shape = (32, 32, 3),
                 filters =10,
                 kernel_size = (3, 3),
                 strides = (1, 1),
                 padding = "same"
                ))
model.add(MaxPooling2D(pool_size  = (2, 2)))
model.output_shape
```

[9]:

```
(None, 16, 16, 10)
```

[10]:

```
model  = Sequential()
model.add(Conv2D(input_shape = (32, 32, 3),
                 filters =10,
                 kernel_size = (3, 3),
                 strides = (1, 1),
                 padding = "valid"
                ))
#model.add(MaxPooling2D(pool_size  = (2, 2)))
model.output_shape
```

[10]:

```
(None, 30, 30, 10)
```

[11]:

```
model  = Sequential()
model.add(Conv2D(input_shape = (32, 32, 3),
                 filters =10,
                 kernel_size = (3, 3),
                 strides = (1, 1),
                 padding = "valid"
                ))
model.add(MaxPooling2D(pool_size  = (2, 2)))
model.output_shape
```

[11]:

```
(None, 15, 15, 10)
```

[12]:

```
model  = Sequential()
model.add(Conv2D(input_shape = (32, 32, 3),
                 filters =10,
                 kernel_size = (3, 3),
                 strides = (1, 1),
                 padding = "valid"
                ))
model.add(MaxPooling2D(pool_size  = (2, 2), padding = "valid"))
model.output_shape
```

[12]:

```
(None, 15, 15, 10)
```

[13]:

```
model  = Sequential()
model.add(Conv2D(input_shape = (32, 32, 3),
                 filters =10,
                 kernel_size = (3, 3),
                 strides = (1, 1),
                 padding = "valid"
                ))
model.add(MaxPooling2D(pool_size  = (2, 2), padding = "same"))
model.output_shape
```

[13]:

```
(None, 15, 15, 10)
```

[14]:

```
model  = Sequential()
model.add(Conv2D(input_shape = (32, 32, 3),
                 filters =10,
                 kernel_size = (3, 3),
                 strides = (1, 1),
                 padding = "valid"
                ))
model.add(MaxPooling2D(pool_size  = (2, 2), padding = "valid", strides  =(1,␣
 ↪1)))
model.output_shape
```

[14]:

```
(None, 29, 29, 10)
```

[15]:

```
model  = Sequential()
model.add(Conv2D(input_shape = (32, 32, 3),
                 filters =10,
                 kernel_size = (3, 3),
                 strides = (1, 1),
                 padding = "valid"
                ))
model.add(MaxPooling2D(pool_size  = (2, 2), padding = "valid", strides  =(2,␣
 ↪2)))
model.output_shape
```

[15]:
```
(None, 15, 15, 10)
```

[16]:
```
model  = Sequential()
model.add(Conv2D(input_shape = (32, 32, 3),
                 filters =10,
                 kernel_size = (3, 3),
                 strides = (1, 1),
                 padding = "valid"
                ))
model.add(MaxPooling2D(pool_size  = (2, 2), padding = "same", strides  =(1, 1)))
model.output_shape
```

[16]:
```
(None, 30, 30, 10)
```

[17]:
```
model  = Sequential()
model.add(Conv2D(input_shape = (32, 32, 3),
                 filters =10,
                 kernel_size = (3, 3),
                 strides = (1, 1),
                 padding = "same"
                ))
model.add(MaxPooling2D(pool_size  = (2, 2), padding = "same", strides  =(1, 1)))
model.output_shape
```

[17]:
```
(None, 32, 32, 10)
```

[18]:
```
model  = Sequential()
model.add(Conv2D(input_shape = (32, 32, 3),
                 filters =10,
                 kernel_size = (3, 3),
                 strides = (1, 1),
                 padding = "valid"
                ))
model.add(GlobalAveragePooling2D())
model.output_shape
```

[18]:
```
(None, 10)
```

[19]:
```
model  = Sequential()
model.add(Conv2D(input_shape = (32, 32, 3),
                 filters =10,
                 kernel_size = (3, 3),
                 strides = (1, 1),
                 padding = "valid"
                ))
print(model.output_shape)
model.add(Flatten())
print(model.output_shape)
```

```
(None, 30, 30, 10)
(None, 9000)
```

[20]:

```python
model  = Sequential()
model.add(Conv2D(input_shape = (32, 32, 3),
                 filters =10,
                 kernel_size = (3, 3),
                 strides = (1, 1),
                 padding = "valid"
                ))
print(model.output_shape)
model.add(Flatten())
print(model.output_shape)
model.add(Dense(2))
print(model.output_shape)
```

```
(None, 30, 30, 10)
(None, 9000)
(None, 2)
```