

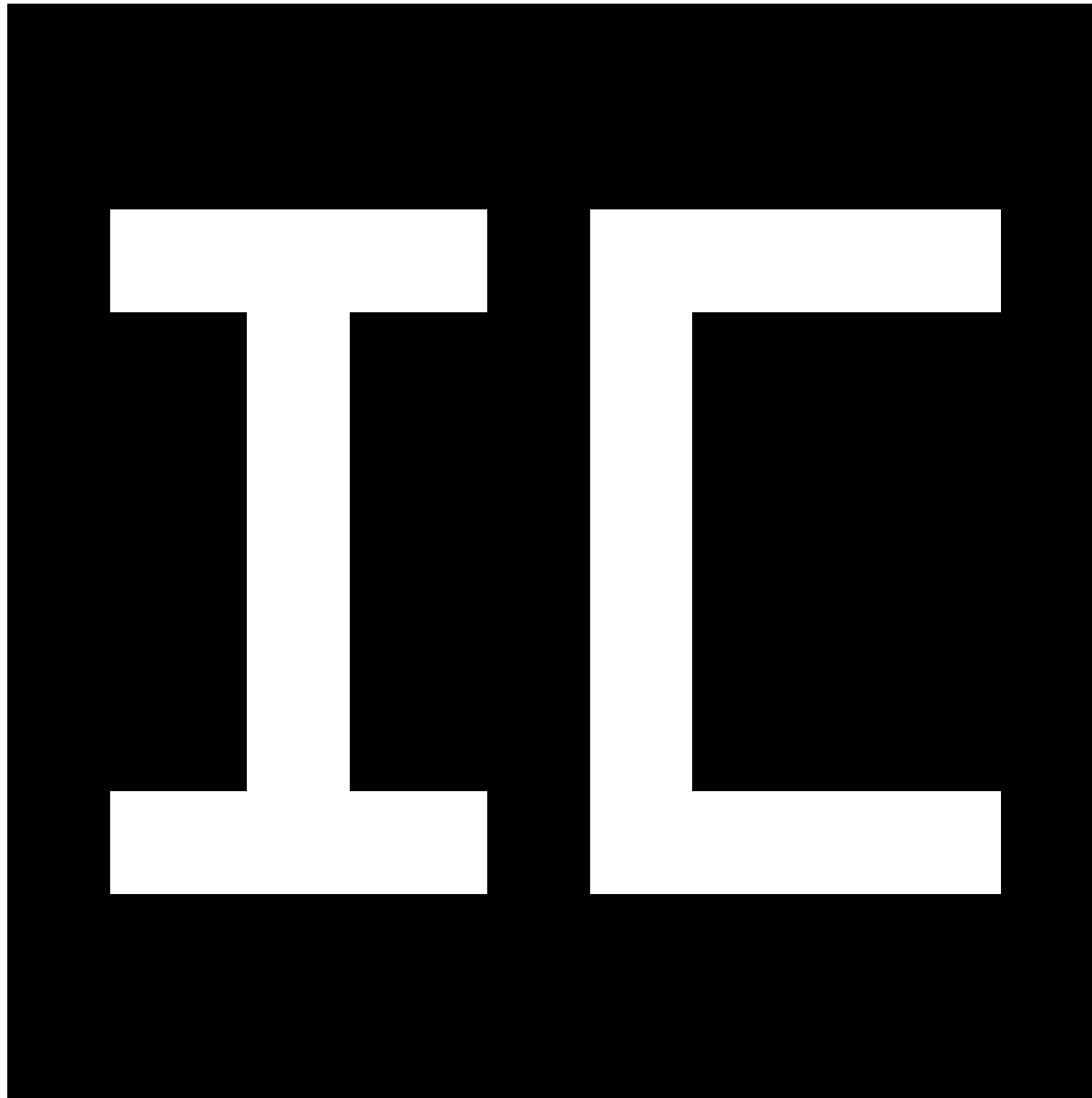
# Aprendizado de Máquina e Deep Learning

## Classificação de imagens

Prof. Dr. Thiago Meirelles Ventura

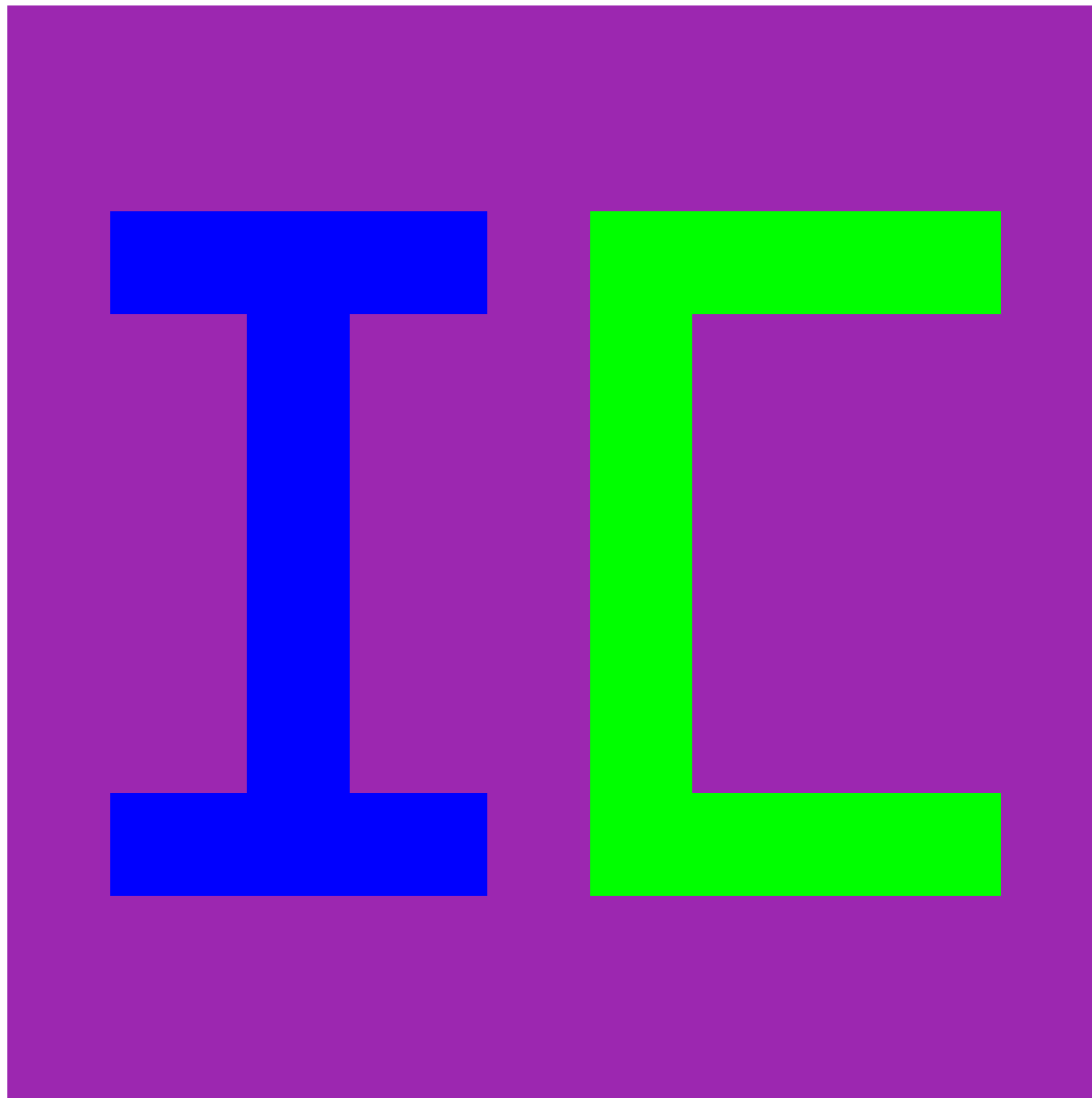
# Discussão

- Já se sabe estimar valores com uma rede neural artificial
- E se for necessário realizar uma tarefa de classificação?
- E se os dados forem imagens?



# Imagem vista pelo computador





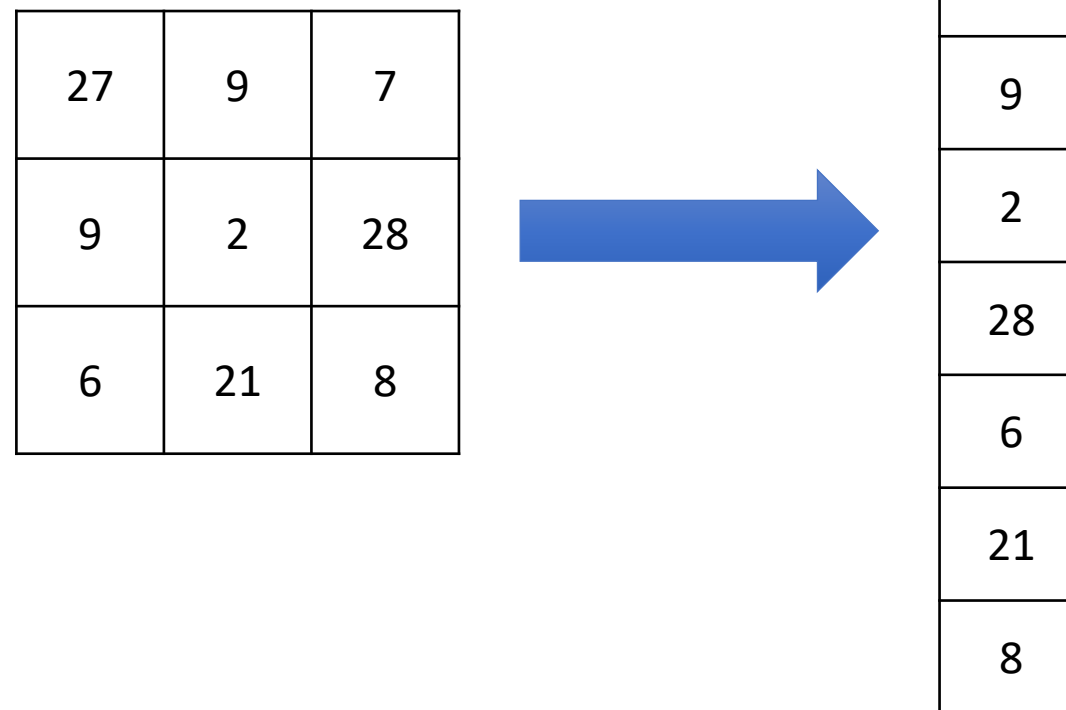
# Imagem colorida vista pelo computador





# Flattening

- Processo para converter vetores N dimensionais em um único vetor 1D



# Flattening

- Camada Flatten no Keras

```
model = keras.Sequential()  
model.add( keras.layers.Flatten() )
```

# Classificação no Keras

- Camadas

```
model.add( keras.layers.Dense(128,  
                                activation=keras.activations.relu ) )
```

```
model.add( keras.layers.Dense( 10,  
                                activation=keras.activations.softmax ) )
```

# Classificação no Keras

- Camadas – Softmax
  - Gera probabilidades como resultados

$$S(y_i) = \frac{e^{y_i}}{\sum_j e^{y_j}}$$

# Classificação no Keras

- Camadas – Softmax

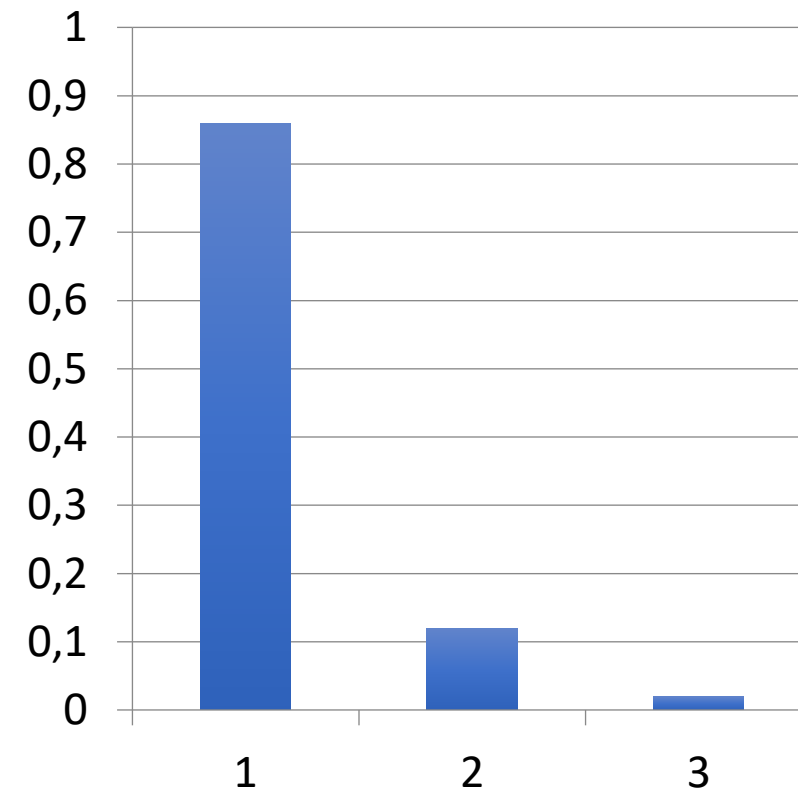
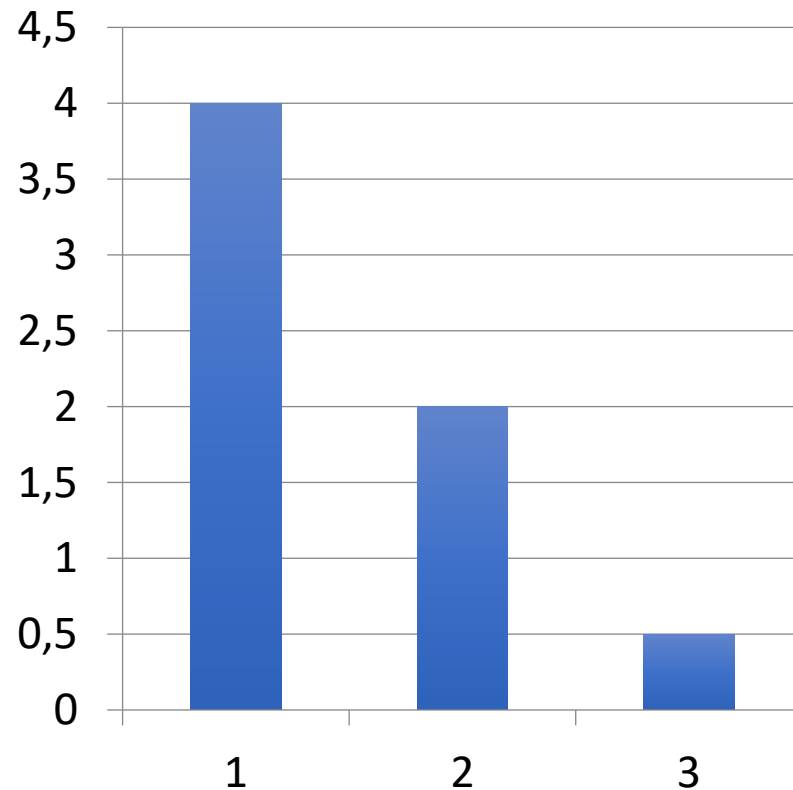
- Gera probabilidades como resultados

$$S(y_i) = \frac{e^{y_i}}{\sum_j e^{y_i}}$$

- Imaginando as saídas [4, 2, 0.5]
  - Qual o resultado após a aplicação do Softmax?
    - Após Softmax: [0.86, 0.12, 0.02]

# Classificação no Keras

- Camadas – Softmax



# Configuração do modelo

```
model.compile(  
    optimizer = keras.optimizers.Adam(),  
    loss = keras.losses.sparse_categorical_crossentropy,  
    metrics = ['accuracy']  
)
```

# Treinamento e classificação

```
hist = model.fit(  
    x_train,  
    y_train,  
    validation_split=0.2,  
    epochs=50)
```

```
classifications = model.predict(x_test)
```



# Datasets

- O Keras facilita a obtenção de algumas bases para testes
  - CIFAR10
  - CIFAR100
  - IMDB Movie reviews
  - Reuters newswire topics
  - MNIST
  - Fashion-MNIST
  - Boston housing price

# Datasets

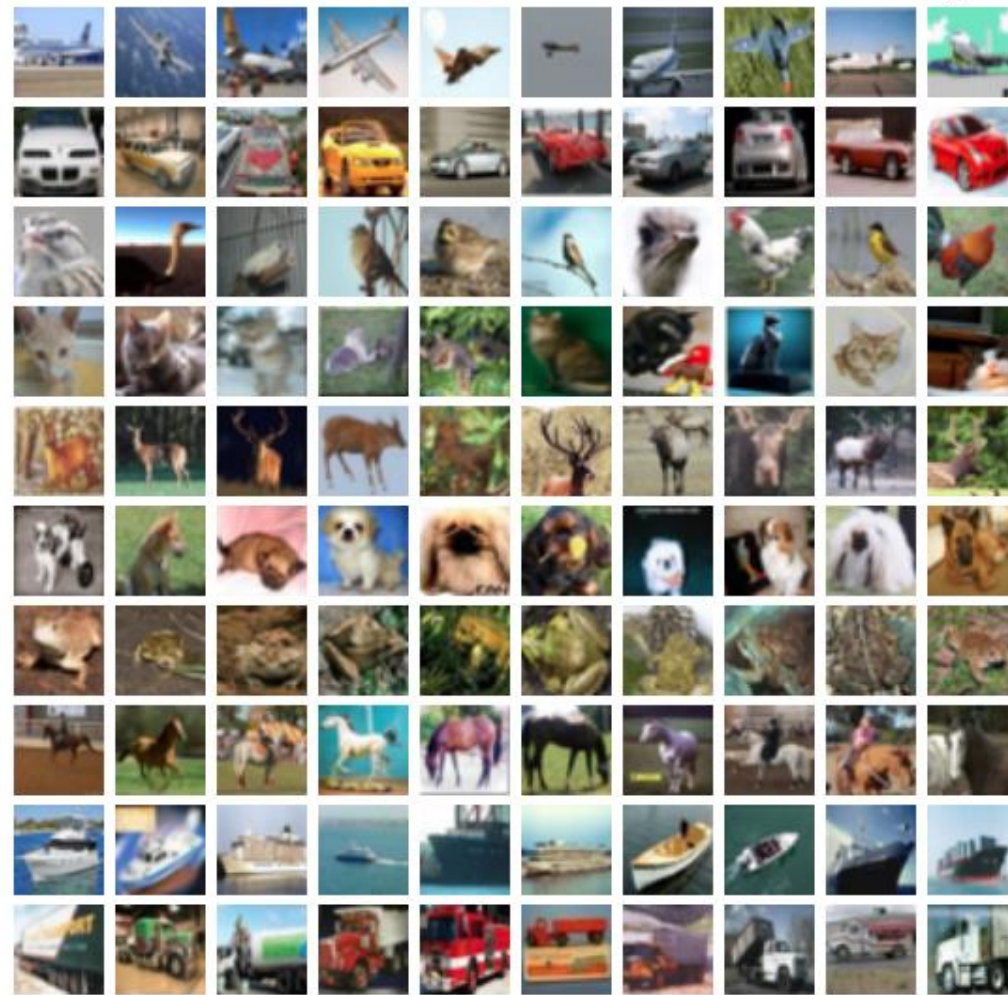
- O Keras facilita a obtenção de algumas bases para testes

```
dataset = keras.datasets.mnist
```

```
(x_train, y_train), (x_test, y_test) = dataset.load_data()
```

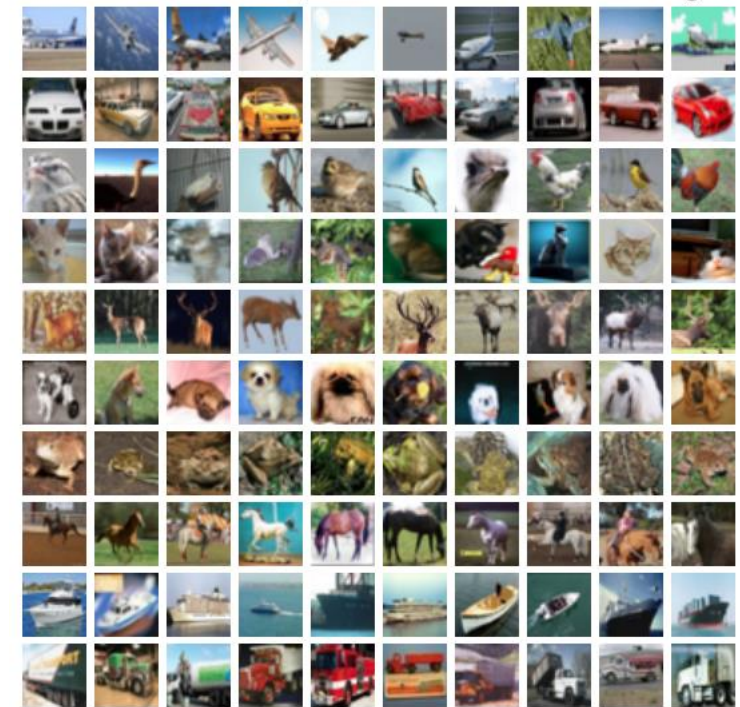
# Datasets

- CIFAR-10



# Datasets

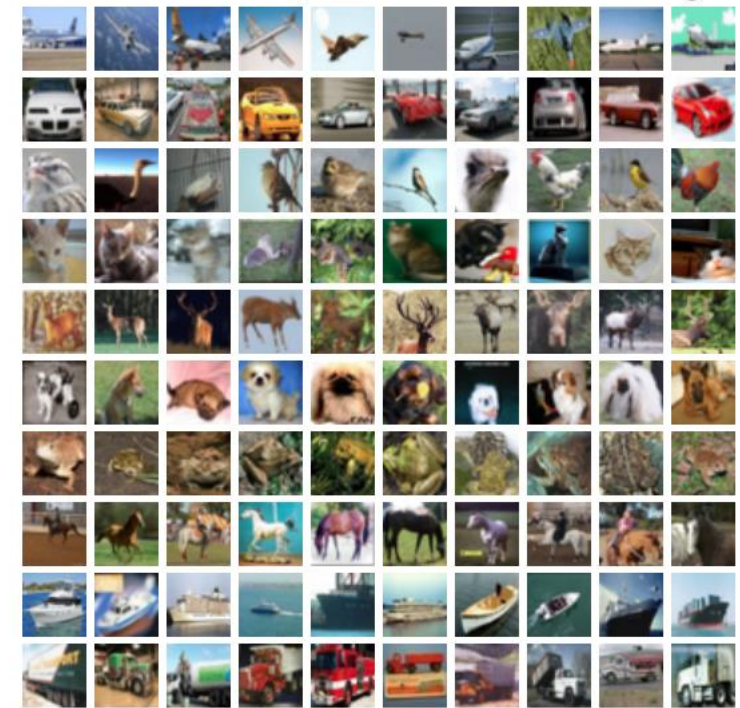
- CIFAR-10
  - 32x32 pixels
  - Em cores
  - 60.000 imagens
    - 50.000 para treinamento
    - 10.000 para teste



# Datasets

- CIFAR-10

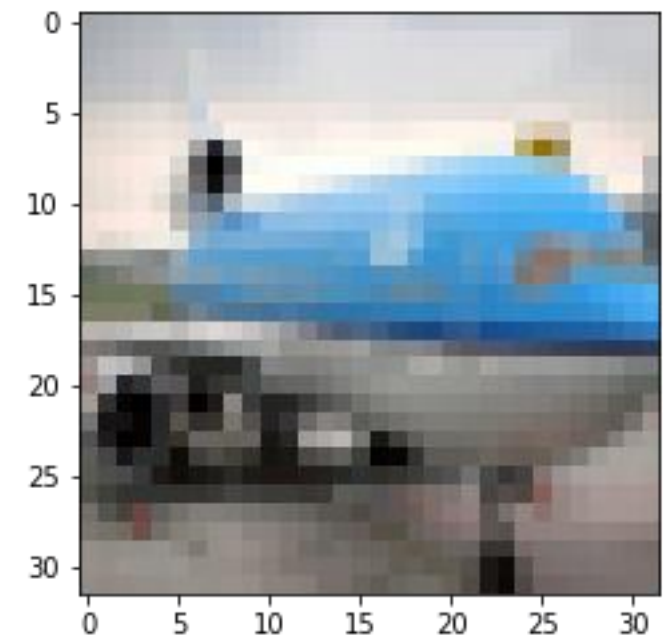
- 10 classes
- Classes: Airplane, Automobile, Bird, Cat, Deer, Dog, Frog, Horse, Ship, Truck
- 6.000 imagens por classe



# Pré-visualização da imagem

- O matplotlib consegue mostrar imagens

```
import matplotlib.pyplot as plt  
plt.imshow(x_train[700])
```



# Demonstração

- Classificar as imagens do CIFAR-10
  - Carregar os dados de treinamento e de teste do CIFAR-10
  - Construir um modelo para classificar imagens
  - Testar o desempenho da rede criada
  - Mostrar algumas imagens que a rede está classificando de maneira incorreta



# Exercício 8

- Faça a classificação da base de dados

MNIST

- Qual o desempenho?
- Por que foi tão diferente do desempenho do CIFAR-10?

