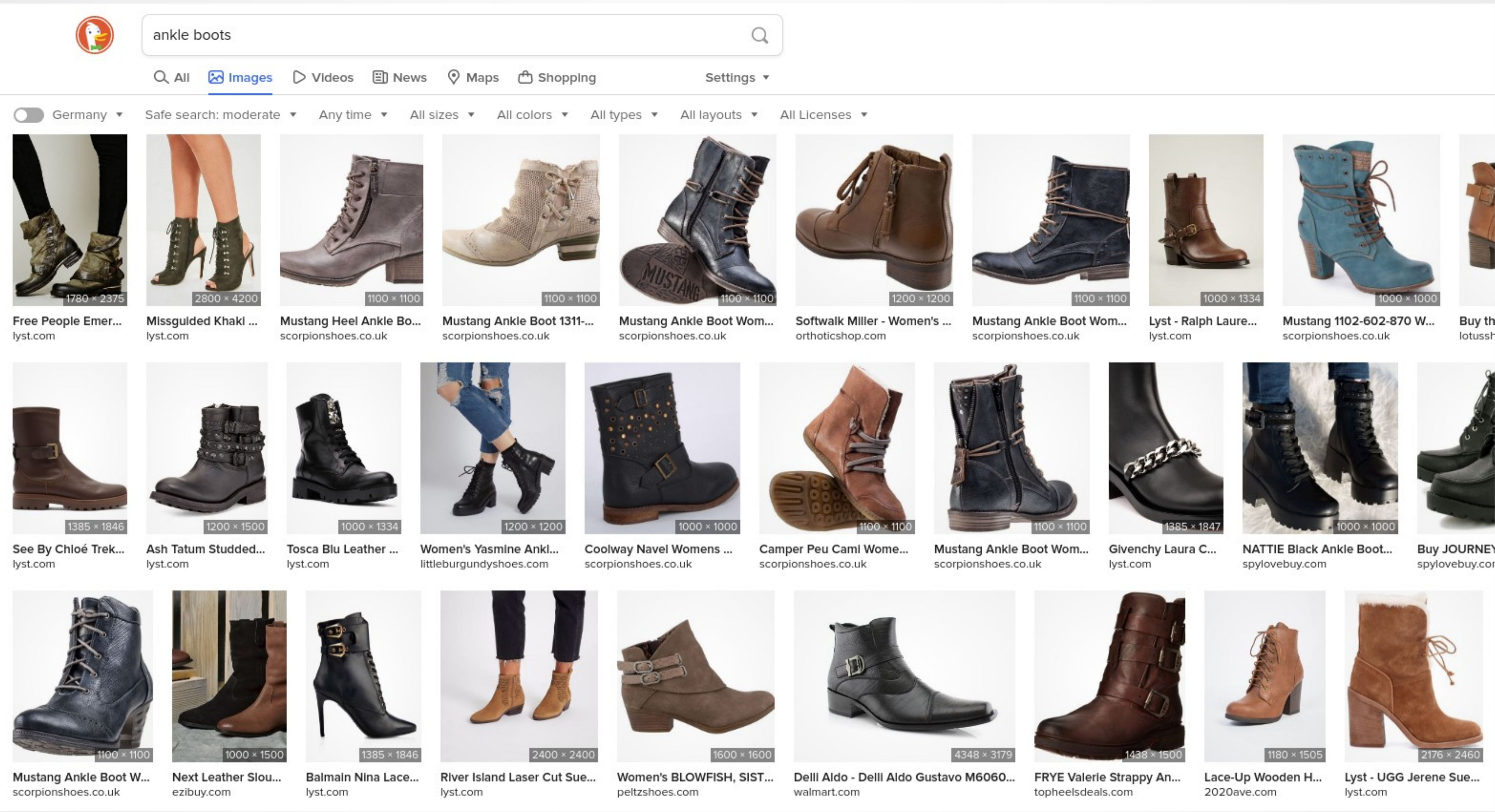
Machine Learning

Jan Grosser, 2021/10/22

Inhalt

- Was ist Machine Learning?
  - Was sind kuenstliche neuronale Netze?
- ML Frameworks/APIs
- ML “Hello world” (primitive Datentypen)
- ML mit komplexen Daten (Fashion MNIST)
- Convolution/Pooling
- ML in Embedded Systems

Was ist Machine Learning?

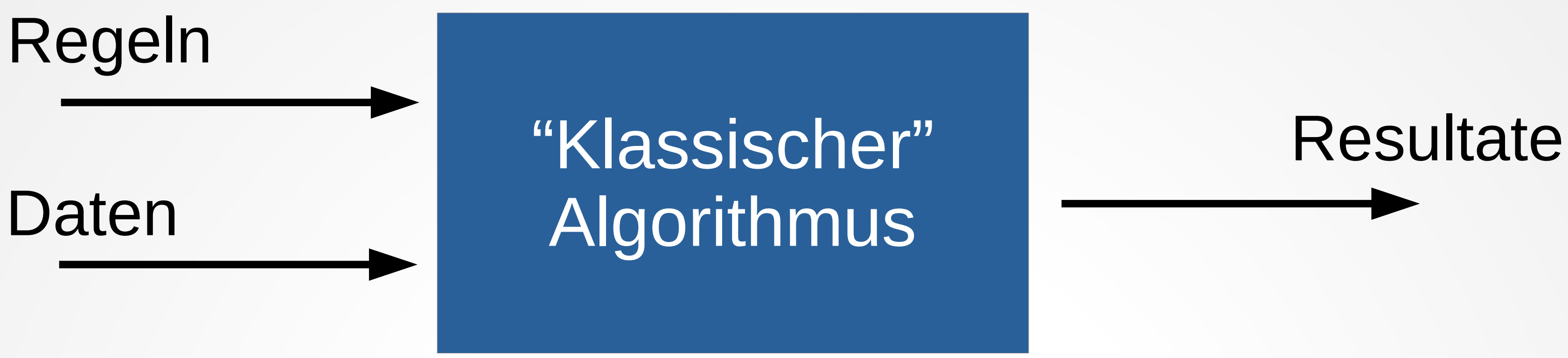


Was ist Machine Learning?

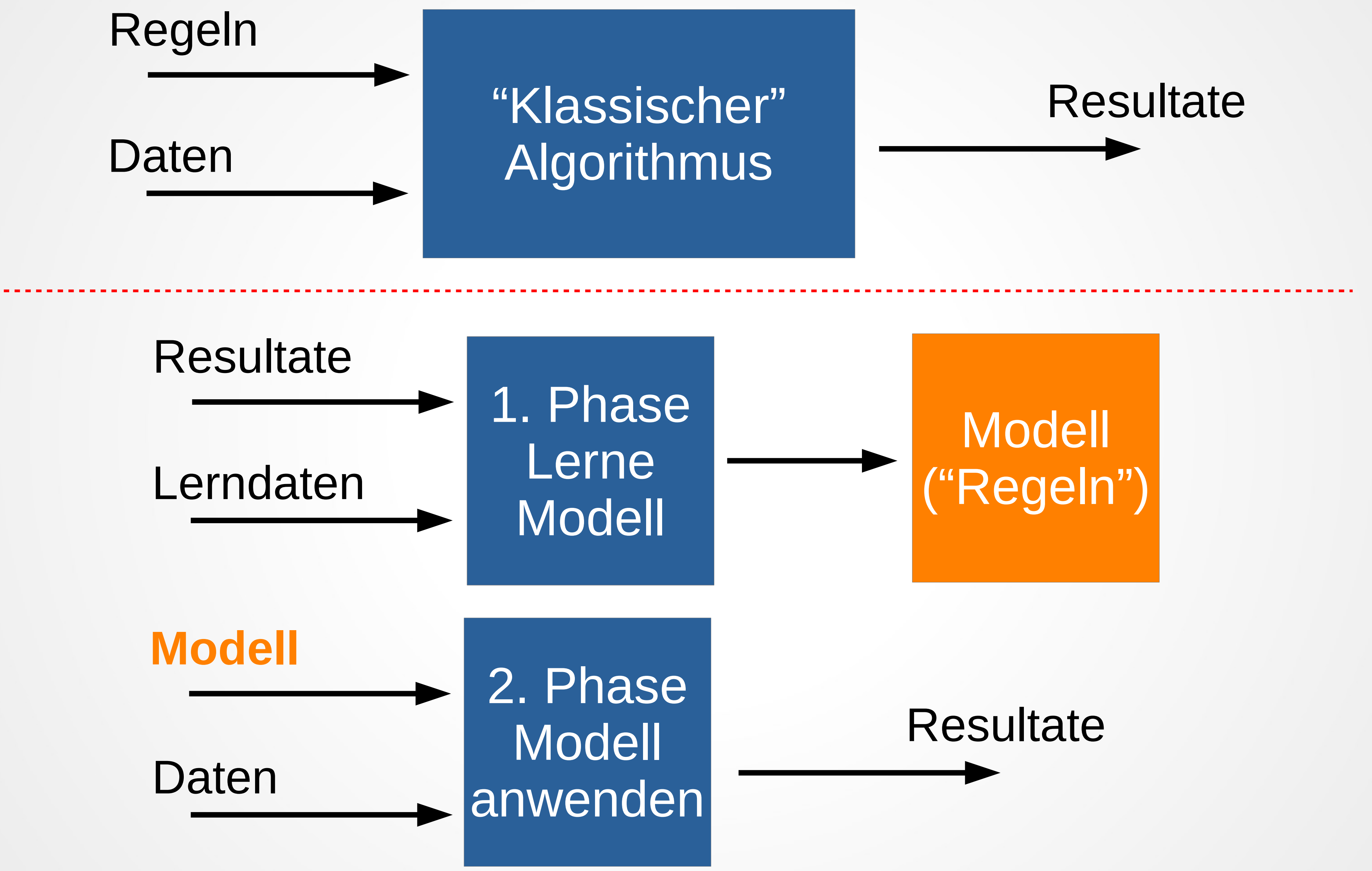
ML Algorithmen

- lernen auf Basis von **Trainingsdaten**
- koennen Charakteristika auch in komplexen Datentypen erkennen
- treffen Vorhersagen fuer Inputs ausserhalb der Trainingsdaten
- koennen sich **kuenstlicher neuronaler Netze** bedienen

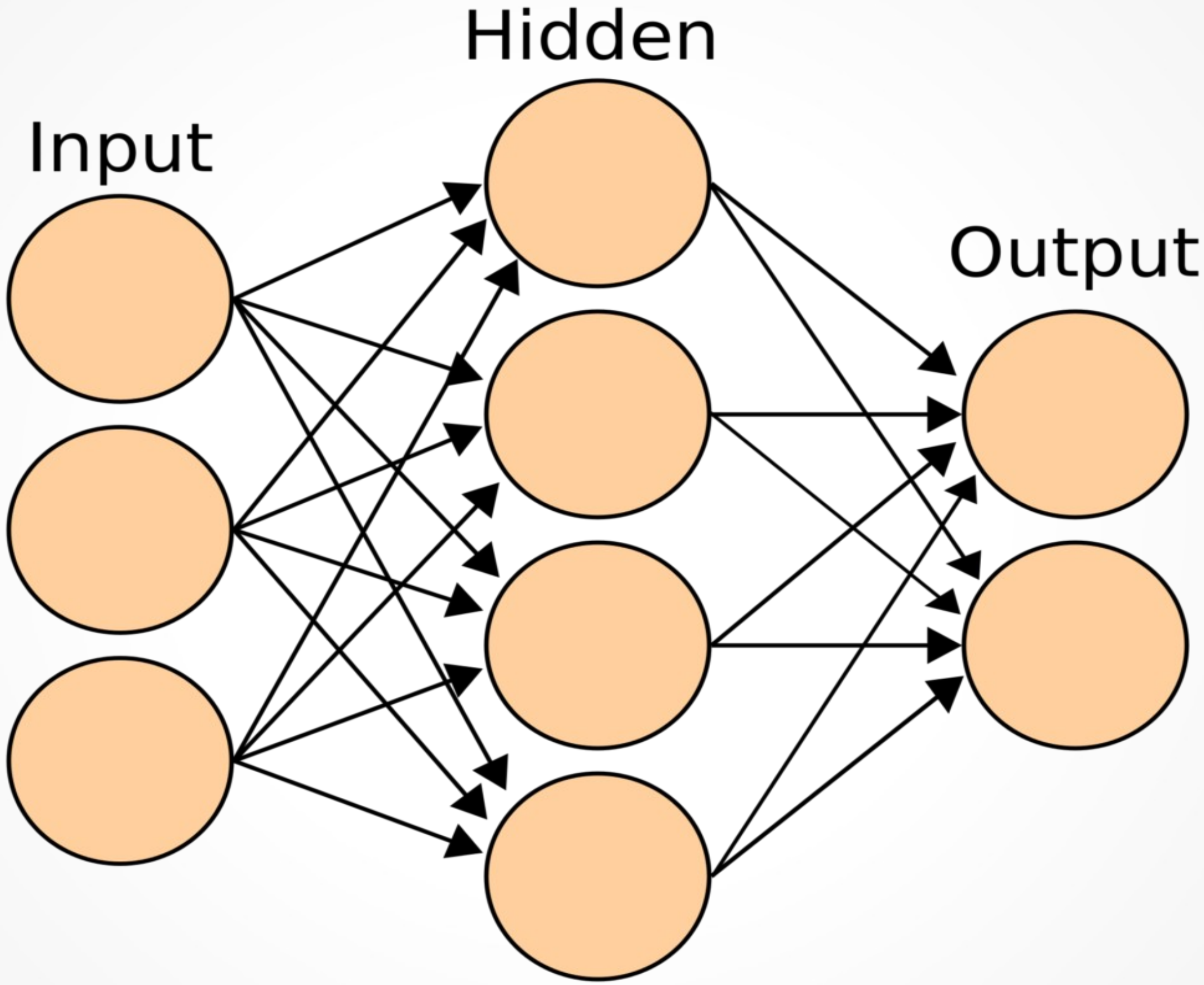
Was ist Machine Learning?



Was ist Machine Learning?



Was sind kuenstliche neuronale Netze?



Was ist Machine Learning?

Grundlegende Konzepte

1) Supervised Learning
2) Unsupervised Learning
3) Reinforcement Learning

Was ist Machine Learning?

Grundlegende Konzepte

1) **Supervised Learning**
2) Unsupervised Learning
3) Reinforcement Learning

ML Frameworks/APIs

Name	Developer	License
TensorFlow	Google	Apache 2.0
PyTorch	Facebook	BSD
Keras (API)	François Chollet (currently working for Google)	MIT License
Microsoft Cognitive Toolkit (CNTK)	Microsoft	MIT License

ML “Hello World”

x = [-1.0, 0.0, 1.0, 2.0,   3.0,   4.0]

y = [-2.0, 1.0, 4.0, 7.0, 10.0, 13.0]

ML “Hello World”

x = [-1.0, 0.0, 1.0, 2.0, 3.0, 4.0]

y = [-2.0, 1.0, 4.0, 7.0, 10.0, 13.0]

y = 3 * x + 1

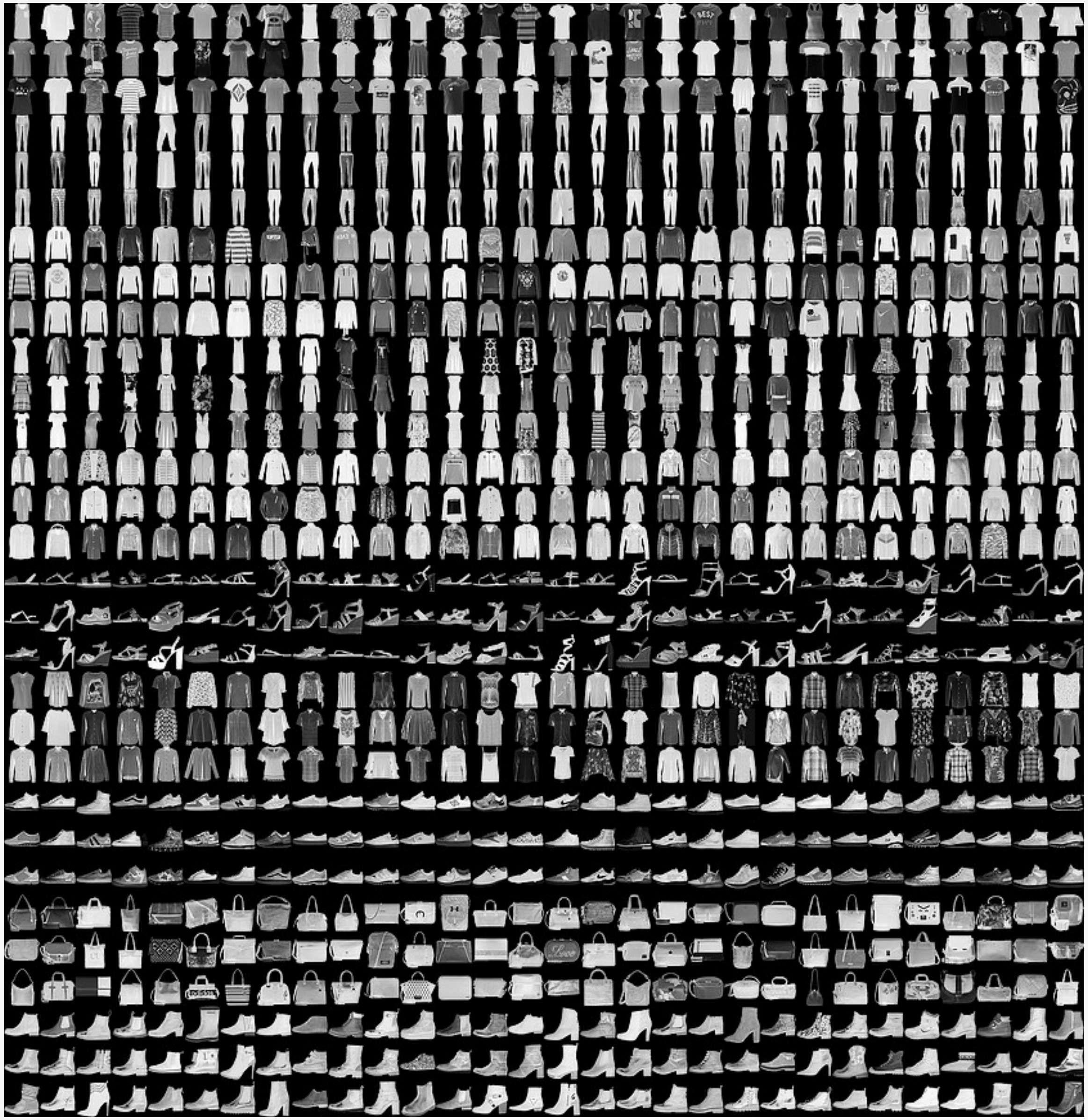
ML “Hello World”

```
import tensorflow as tf
import numpy as np
model = tf.keras.Sequential([tf.keras.layers.Dense(units=1,
    input_shape=[1])])
model.compile(optimizer='sgd', loss='mean_squared_error')
xs = np.array([-1.0, 0.0, 1.0, 2.0, 3.0, 4.0], dtype=float)
ys = np.array([-2.0, 1.0, 4.0, 7.0, 10.0, 13.0], dtype=float)
n_epochs = 100
model.fit(xs, ys, epochs=n_epochs)
x_test = 10
print(model.predict([x_test]))
```

ML “Hello World”

- Jupyter Notebook

ML mit komplexen Daten (Fashion MNIST)



https://github.com/zalandoresearch/fashion-mnist

ML mit komplexen Daten (Fashion MNIST)

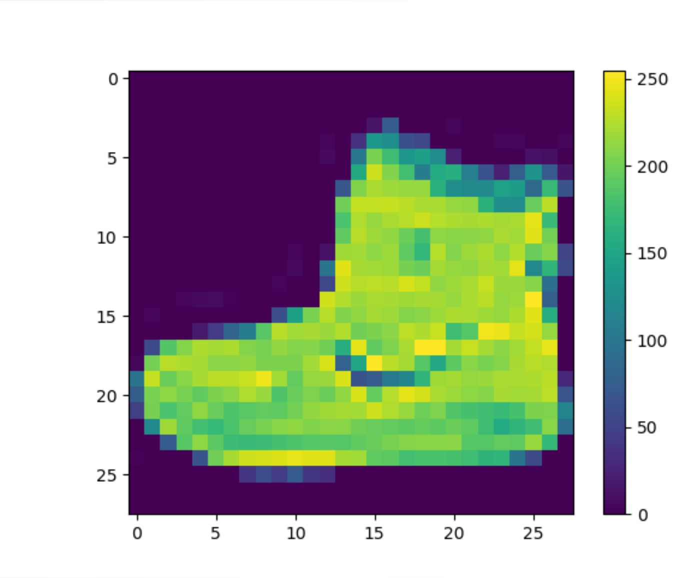
- 70.000 von Modeartikeln
- Jedes Bild 28 x 28 Pixel, 255 Graustufen
- Alle Bilder sind 10 Kategorien zugeordnet

  'T-shirt/top', 'Trouser', 'Pullover', 'Dress', 'Coat', 'Sandal', 'Shirt', 'Sneaker', 'Bag', 'Ankle boot'
- Trainingssatz: 60.000 Bilder
- Testsatz: 10.000 Bilder
- Jedes Bild zeigt nur ein Objekt
- Objekte in Bildern zentriert

ML mit komplexen Daten (Fashion MNIST)

- Jupyter Notebook

Convolution/Pooling



Fashion MNIST

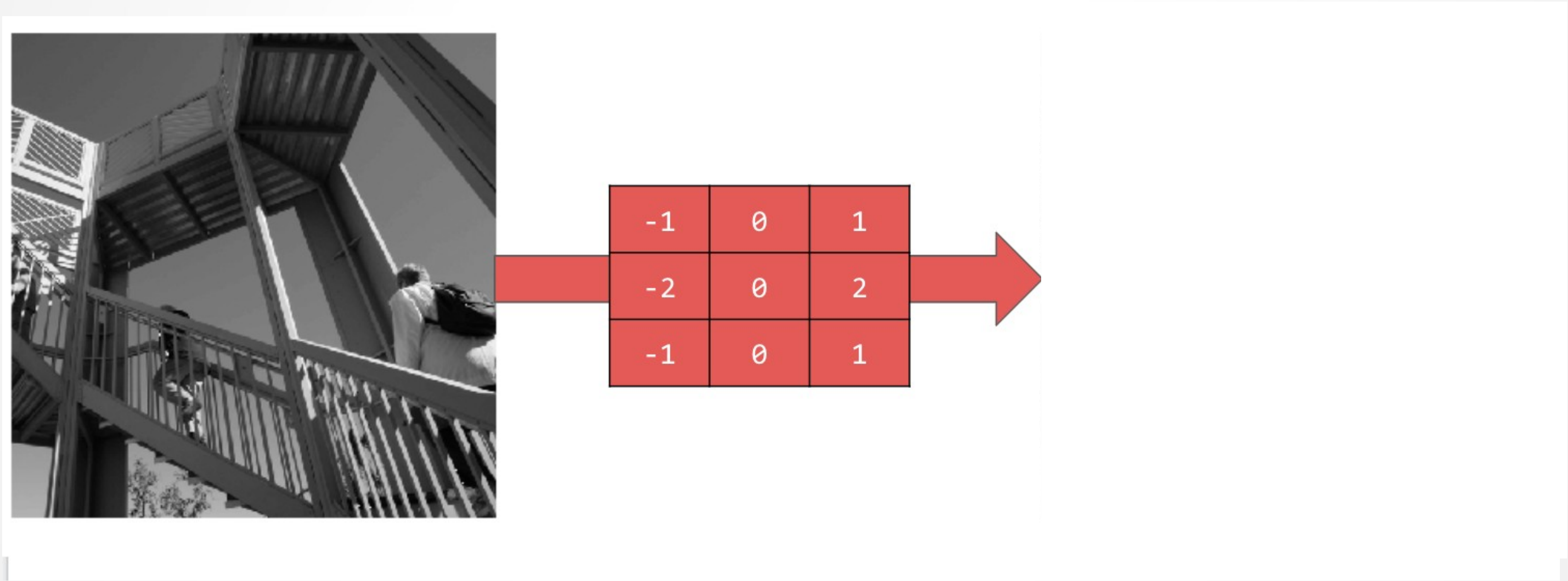
“Real World”

Convolution/Pooling

Convolution: Anwendung eines Filters zur Hervorhebung charakteristischer Merkmale

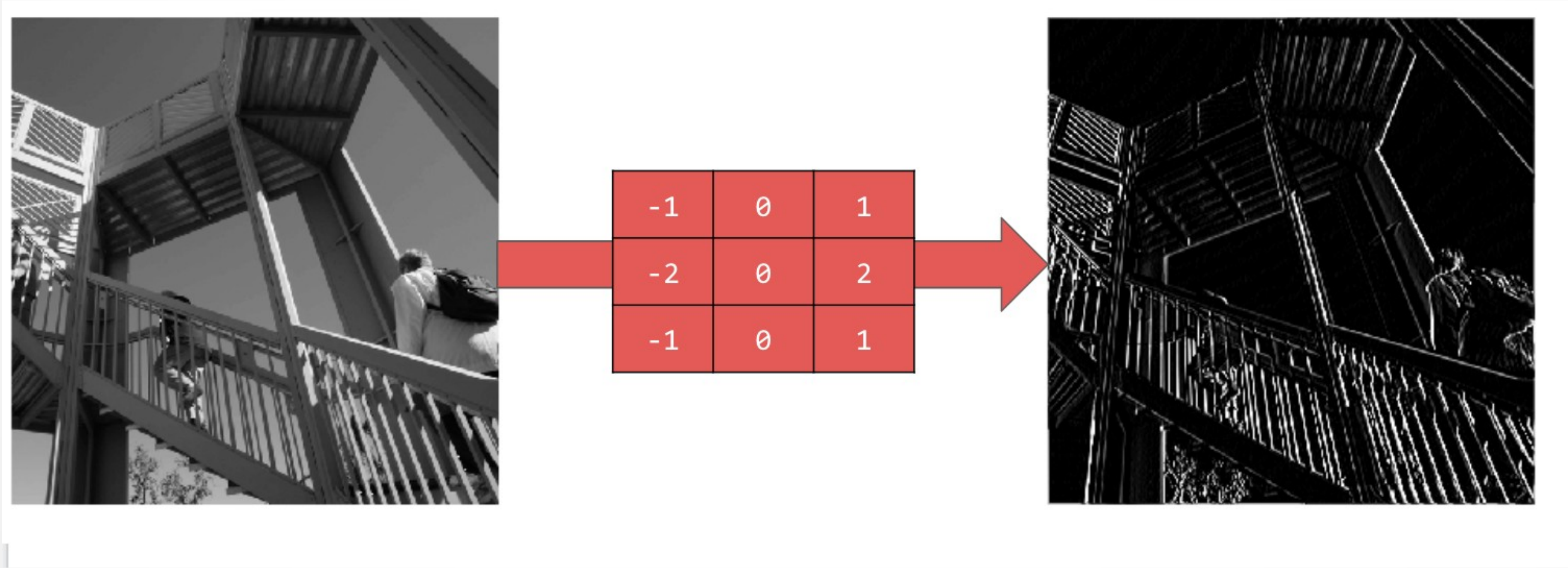
Convolution/Pooling

Convolution: Anwendung eines Filters zur Hervorhebung charakteristischer Merkmale



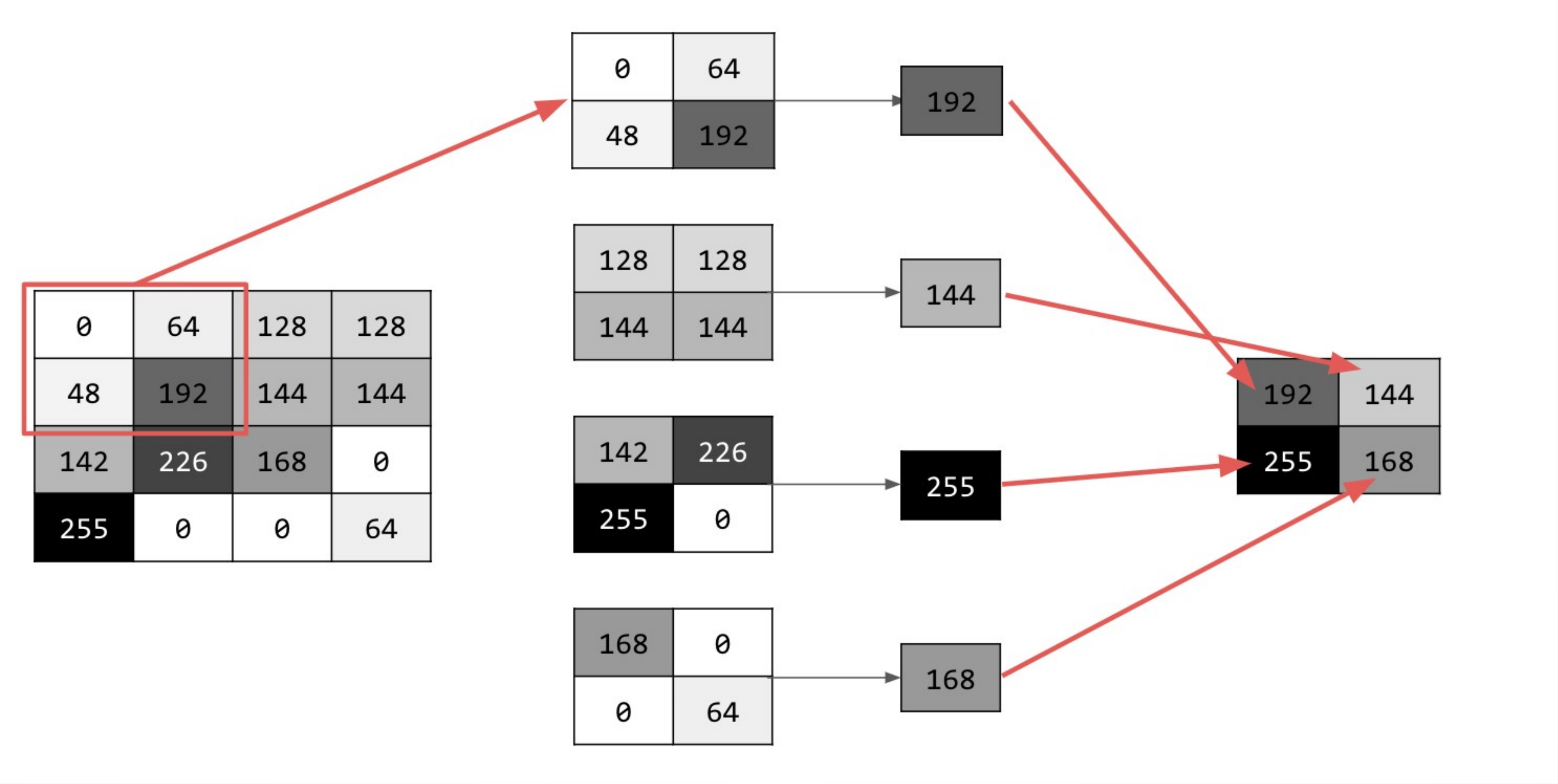
Convolution/Pooling

Convolution: Anwendung eines Filters zur Hervorhebung charakteristischer Merkmale



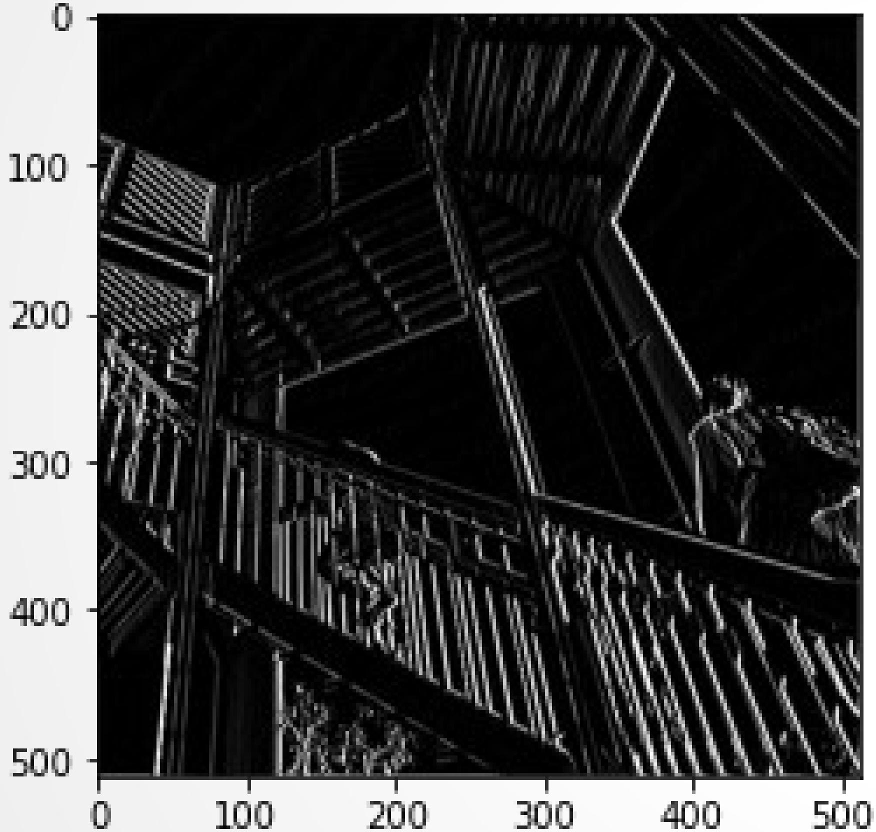
Convolution/Pooling

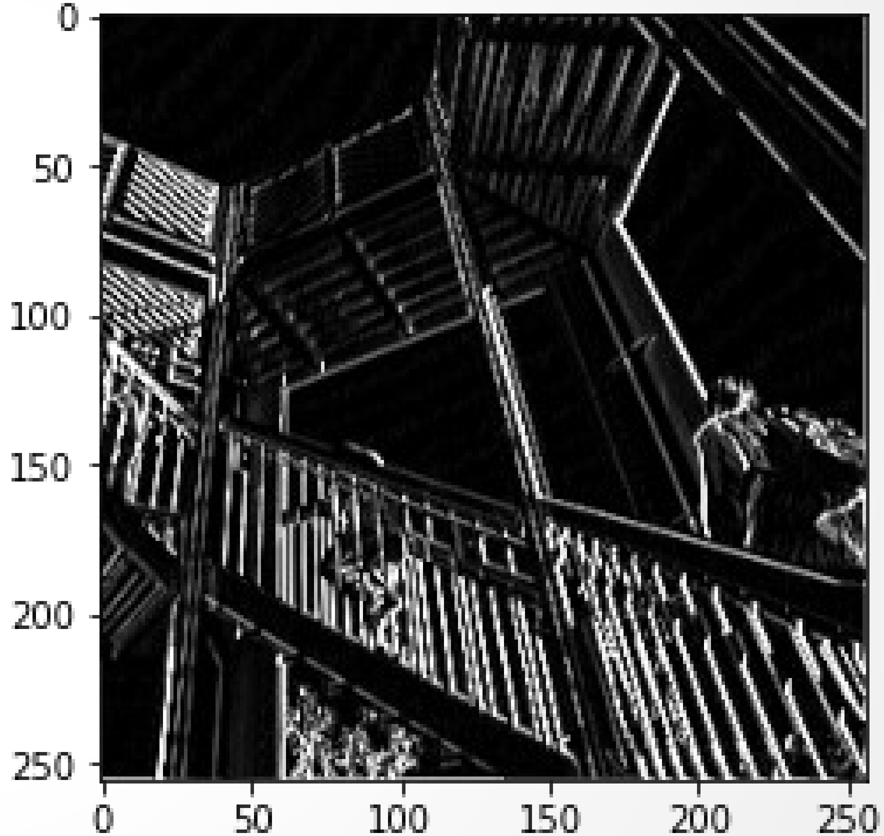
Pooling: Reduktion der Daten bei Beibehaltung der relevanten Information



Convolution/Pooling

Beispiel: MaxPooling2D(2,2)





Convolution/Pooling

- Jupyter Notebook

Convolution/Pooling

Bsp ohne Convolution/Pooling:

```
model = tf.keras.Sequential([
    tf.keras.layers.Flatten(input_shape=(28, 28)),
    tf.keras.layers.Dense(128, activation='relu'),
    tf.keras.layers.Dense(10)
])
```

Korrespondiert mit
den 10 Kategorien!

Convolution/Pooling

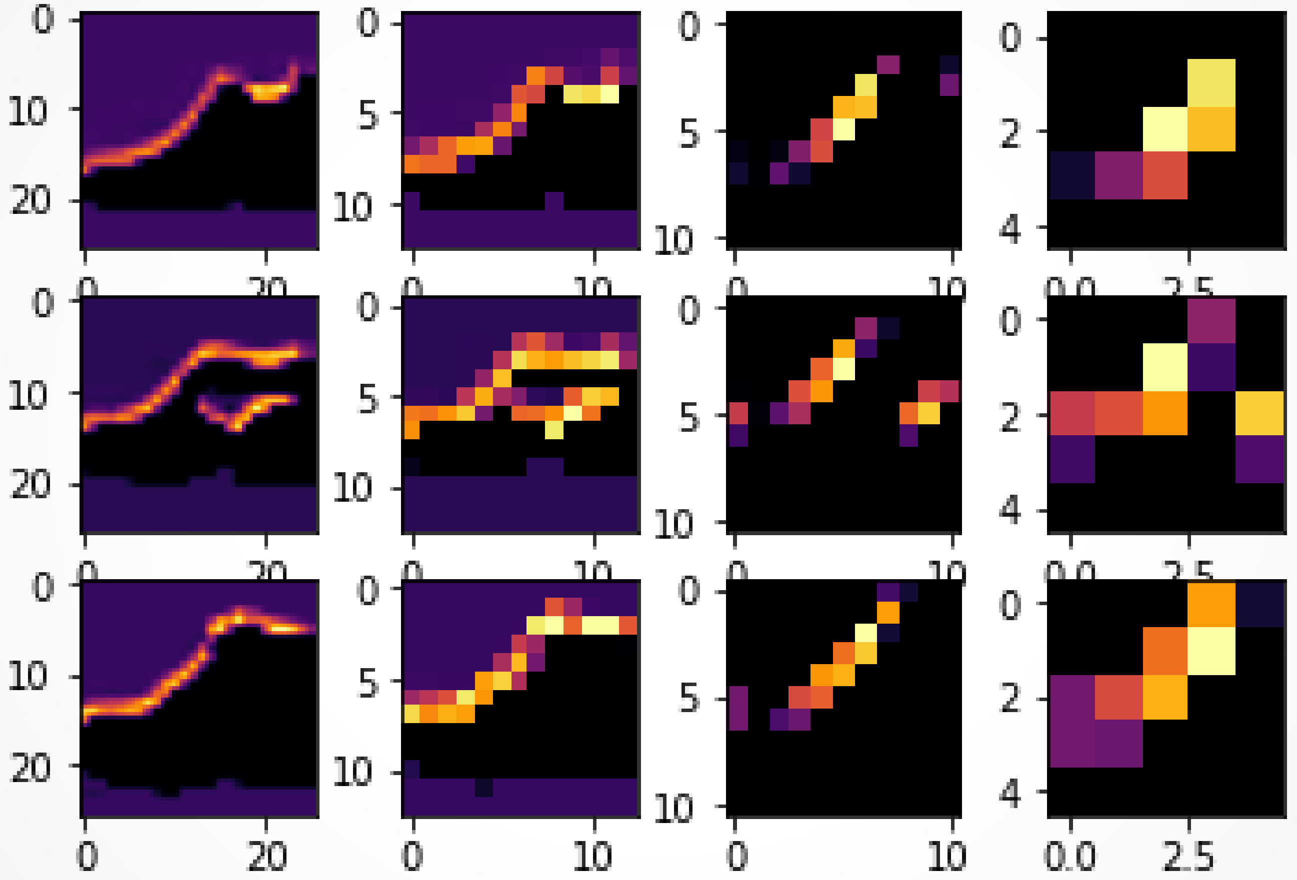
Bsp MIT Convolution/Pooling:

```
training_images=training_images.reshape(60000, 28, 28, 1)
model2 = tf.keras.models.Sequential([
  tf.keras.layers.Conv2D(64, (3,3), activation='relu', input_shape=(28, 28, 1)),
  tf.keras.layers.MaxPooling2D(2, 2),
  tf.keras.layers.Conv2D(64, (3,3), activation='relu'),
  tf.keras.layers.MaxPooling2D(2, 2),
  tf.keras.layers.Flatten(),
  tf.keras.layers.Dense(128, activation='relu'),
  tf.keras.layers.Dense(10, activation='softmax')
])
```

Convolution/Pooling

- Jupyter Notebook

Convolution/Pooling

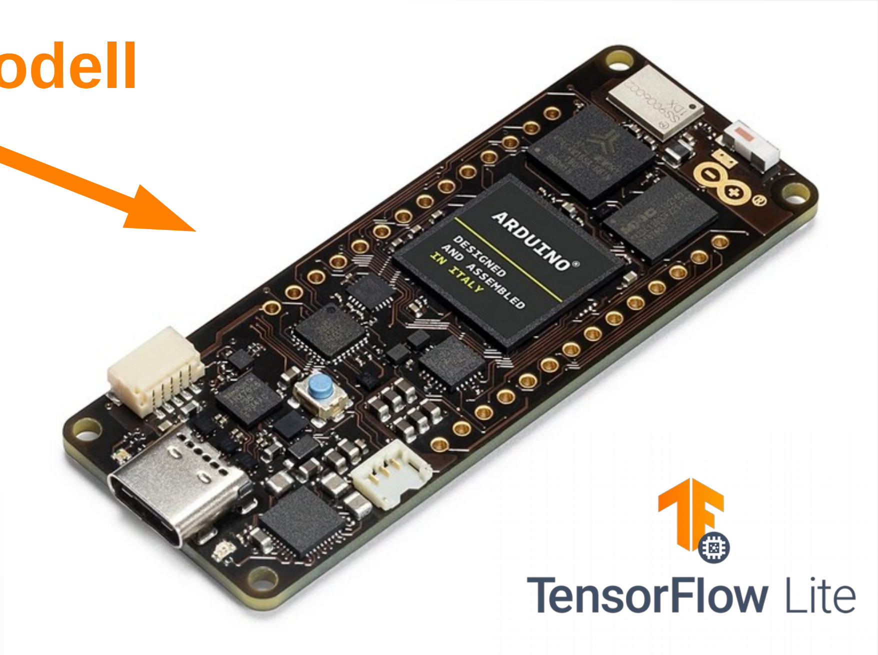


ML in Embedded Systems

1) Lernen

Modell

2) Anwenden



Zum Weitermachen ...

- YouTube: Intro to Machine Learning (ML Zero to Hero)
- https://github.com/lmoroney/mlday-tokyo
- https://www.tensorflow.org/tutorials

- https://github.com/rzbrk/ml-demo