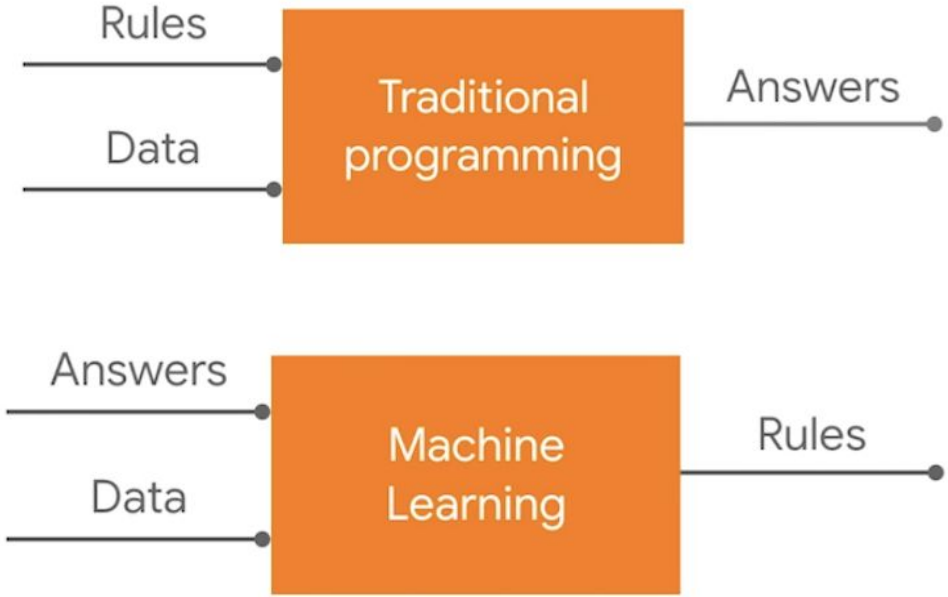
HOCHSCHULE DER MEDIEN

Deep Learning

Introduction

Prof. Dr. Jan Kirenz
HdM Stuttgart

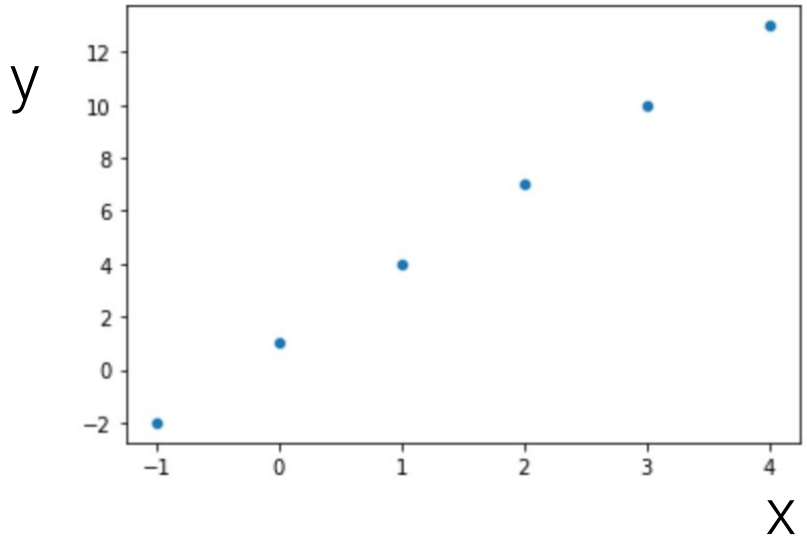
Source: Moroney (2019)



Source: Moroney (2019)

Simple regression example

Variable	Value 1	Value 2	Value 3	Value 4	Value 5	Value 6
x	-1	0	1	2	3	4
y	-2	1	4	7	10	13



Source: Moroney (2019)

import libraries

```
import numpy as np
import tensorflow as tf
from tensorflow import keras
```

```
# Data
x = np.array([-1.0, 0.0, 1.0, 2.0, 3.0, 4.0], dtype=float)
y = np.array([-2.0, 1.0, 4.0, 7.0, 10.0, 13.0], dtype=float)

# Model definition
model = tf.keras.Sequential([keras.layers.Dense(units=1, input_shape=[1])])
model.compile(optimizer='sgd', loss='mean_squared_error')

# Model fitting
model.fit(x, y, epochs=50)

# Model prediction
print(model.predict([10.0]))
```

Source: Moroney (2019)

create data

```
import numpy as np
import tensorflow as tf
from tensorflow import keras

# Data
x = np.array([-1.0, 0.0, 1.0, 2.0, 3.0, 4.0], dtype=float)
y = np.array([-2.0, 1.0, 4.0, 7.0, 10.0, 13.0], dtype=float)

# Model definition
model = tf.keras.Sequential([keras.layers.Dense(units=1, input_shape=[1])])
model.compile(optimizer='sgd', loss='mean_squared_error')

# Model fitting
model.fit(x, y, epochs=50)

# Model prediction
print(model.predict([10.0]))
```

Source: Moroney (2019)

define model architecture

```
import numpy as np
import tensorflow as tf
from tensorflow import keras

# Data
x = np.array([-1.0, 0.0, 1.0, 2.0, 3.0, 4.0], dtype=float)
y = np.array([-2.0, 1.0, 4.0, 7.0, 10.0, 13.0], dtype=float)

# Model definition
model = tf.keras.Sequential([keras.layers.Dense(units=1, input_shape=[1])])
model.compile(optimizer='sgd', loss='mean_squared_error')

# Model fitting
model.fit(x, y, epochs=50)

# Model prediction
print(model.predict([10.0]))
```

Source: Moroney (2019)

we use a single layer

```
import numpy as np
import tensorflow as tf
from tensorflow import keras

# Data
x = np.array([-1.0, 0.0, 1.0, 2.0, 3.0, 4.0], dtype=float)
y = np.array([-2.0, 1.0, 4.0, 7.0, 10.0, 13.0], dtype=float)

# Model definition
model = tf.keras.Sequential([keras.layers.Dense(units=1, input_shape=[1])])
model.compile(optimizer='sgd', loss='mean_squared_error')

# Model fitting
model.fit(x, y, epochs=50)

# Model prediction
print(model.predict([10.0]))
```

Source: Moroney (2019)

with one neuron

```
import numpy as np
import tensorflow as tf
from tensorflow import keras

# Data
x = np.array([-1.0, 0.0, 1.0, 2.0, 3.0, 4.0], dtype=float)
y = np.array([-2.0, 1.0, 4.0, 7.0, 10.0, 13.0], dtype=float)

# Model definition
model = tf.keras.Sequential([keras.layers.Dense(units=1, input_shape=[1])])
model.compile(optimizer='sgd', loss='mean_squared_error')

# Model fitting
model.fit(x, y, epochs=50)

# Model prediction
print(model.predict([10.0]))
```

Source: Moroney (2019)

and only one input x

```
import numpy as np
import tensorflow as tf
from tensorflow import keras

# Data
x = np.array([-1.0, 0.0, 1.0, 2.0, 3.0, 4.0], dtype=float)
y = np.array([-2.0, 1.0, 4.0, 7.0, 10.0, 13.0], dtype=float)

# Model definition
model = tf.keras.Sequential([keras.layers.Dense(units=1, input_shape=[1])])
model.compile(optimizer='sgd', loss='mean_squared_error')

# Model fitting
model.fit(x, y, epochs=50)

# Model prediction
print(model.predict([10.0]))
```

Source: Moroney (2019)

we compile the model

```
import numpy as np
import tensorflow as tf
from tensorflow import keras

# Data
x = np.array([-1.0, 0.0, 1.0, 2.0, 3.0, 4.0], dtype=float)
y = np.array([-2.0, 1.0, 4.0, 7.0, 10.0, 13.0], dtype=float)

# Model definition
model = tf.keras.Sequential([keras.layers.Dense(units=1, input_shape=[1])])
model.compile(optimizer='sgd', loss='mean_squared_error')

# Model fitting
model.fit(x, y, epochs=50)

# Model prediction
print(model.predict([10.0]))
```

Source: Moroney (2019)

generate a guess for y

sgd = stochastic gradient descent

```
import numpy as np
import tensorflow as tf
from tensorflow import keras

# Data
x = np.array([-1.0, 0.0, 1.0, 2.0, 3.0, 4.0], dtype=float)
y = np.array([-2.0, 1.0, 4.0, 7.0, 10.0, 13.0], dtype=float)

# Model definition
model = tf.keras.Sequential([keras.layers.Dense(units=1, input_shape=[1])])
model.compile(optimizer='sgd', loss='mean_squared_error')

# Model fitting
model.fit(x, y, epochs=50)

# Model prediction
print(model.predict([10.0]))
```

Source: Moroney (2019)

and calculate the error

```
import numpy as np
import tensorflow as tf
from tensorflow import keras

# Data
x = np.array([-1.0, 0.0, 1.0, 2.0, 3.0, 4.0], dtype=float)
y = np.array([-2.0, 1.0, 4.0, 7.0, 10.0, 13.0], dtype=float)

# Model definition
model = tf.keras.Sequential([keras.layers.Dense(units=1, input_shape=[1])])
model.compile(optimizer='sgd', loss='mean_squared_error')

# Model fitting
model.fit(x, y, epochs=50)

# Model prediction
print(model.predict([10.0]))
```

Source: Moroney (2019)

```
import numpy as np
import tensorflow as tf
from tensorflow import keras

# Data
x = np.array([-1.0, 0.0, 1.0, 2.0, 3.0, 4.0], dtype=float)
y = np.array([-2.0, 1.0, 4.0, 7.0, 10.0, 13.0], dtype=float)

# Model definition
model = tf.keras.Sequential([keras.layers.Dense(units=1, input_shape=[1])])
model.compile(optimizer='sgd', loss='mean_squared_error')

# Model fitting
model.fit(x, y, epochs=50)

# Model prediction
print(model.predict([10.0]))
```

we do this 500 times

Source: Moroney (2019)

predict y for x=10

```
import numpy as np
import tensorflow as tf
from tensorflow import keras

# Data
x = np.array([-1.0, 0.0, 1.0, 2.0, 3.0, 4.0], dtype=float)
y = np.array([-2.0, 1.0, 4.0, 7.0, 10.0, 13.0], dtype=float)

# Model definition
model = tf.keras.Sequential([keras.layers.Dense(units=1, input_shape=[1])])
model.compile(optimizer='sgd', loss='mean_squared_error')

# Model fitting
model.fit(x, y, epochs=50)

# Model prediction
print(model.predict([10.0]))
```

Source: Moroney (2019)

Resources

The slides are based on the excellent video tutorial “Intro to Machine Learning (ML Zero to Hero - Part 1)” by Lawrence Moroney.