Prof. Dr. L. Paditz

Uebung am 11.02.2019, 1.DS:

============================

```
parallels@parallels-Parallels-Virtual-Platform:~$ python3

Python 3.6.7 (default, Oct 22 2018, 11:32:17)

[GCC 8.2.0] on linux

Type ”help”, ”copyright”, ”credits” or ”license” for

more information.

>>> import tensorflow as tf

>>> a = tf.constant(23.)

>>> print(a)

Tensor(”Const:0”, shape=(), dtype=float32)

>>> b = tf.constant([1,2,3])

>>> print(b)

Tensor(”Const_1:0”, shape=(3,), dtype=int32)

>>> c = tf.constant([[1,2,3],[4,5,6]])

>>> print(c)
```

```
Tensor("Const_2:0", shape=(2, 3), dtype=int32)

>>> d =

tf.constant([[[1,2,3],[4,5,6]],[[1,1,1],[0,0,0

>>> print(d)

Tensor("Const_3:0", shape=(3, 2, 3), dtype=int32)

>>> e =

tf.constant([[[[1],[2],[3]],[[1],[2],[3]]]])

>>> print(e)

Tensor("Const_4:0", shape=(1, 2, 3, 1),

dtype=int32)

>>>
```

Uebung am 12.02.2019, 1.DS:

==============================

Define loss(x)=$2x^2-3x+5$

done

2D−Grafik Y1:… Y2:…

loss(x)

$2\cdot x^2-3\cdot x+5$

Define $\mathrm{grad}(x)=\frac{d}{dx}(\mathrm{loss}(x))$

done

grad(x)

$4\cdot x-3$

solve(grad(x)=0,x)

$\left\{x=\frac{3}{4}\right\}$

$\frac{d^2}{dx^2}(\mathrm{loss}(x))>0$

$4>0$

fMin(loss(x),x)

$\left\{\mathrm{MinValue}=\frac{31}{8},x=\frac{3}{4}\right\}$

approx(ans)

{MinValue=3.875,x=0.75}

Sei loss(x) eine Verlustfunktion,

ges. ist der minimale Verlust

Falls $x>\frac{3}{4}$ ist grad(x)>0 und der min. Verlust liegt in neg. Richtung, d.h. in **Richtung des neg. Gradienten.**

Falls $x<\frac{3}{4}$ ist grad(x)<0 und der min. Verlust liegt in pos. Richtung, d.h. ebenfalls in **Richtung des neg. Gradienten.**

Die Richtung des neg. Gradienten ist stets die "steilste" Abstiegsrichtung hin zum Min.!

tensorflow-Skript:

```
python3
import tensorflow as tf
x_var = tf.Variable(0., name='x_result')
# Schrittzähler
step_var = tf.Variable(0, trainable=False)
# Verlust
loss = 2 * tf.multiply(x_var, x_var) -
tf.multiply(3.0, x_var) + 5.0
# Variablenwert finden, der Verlust minimiert
learn_rate = tf.constant(0.1)
num_epochs = 200
optimizer =
tf.train.GradientDescentOptimizer(learn_rate).minimize(▶
global_step=step_var)
# Variable initialisieren
init = tf.global_variables_initializer()
# Session starten
with tf.Session() as sess:
    sess.run(init)
    sess.run([step_var, x_var, learn_rate, loss])
    for epoch in range(num_epochs):
        sess.run(optimizer)
        if epoch % 10 == 0:
            sess.run([step_var, x_var, loss])
```

Rechnerprotokoll:

```
parallels@parallels-Parallels-Virtual-Platform:~$ python3
Python 3.6.7 (default, Oct 22 2018, 11:32:17)
```

```
[GCC 8.2.0] on linux
Type "help", "copyright", "credits" or "license" for
more information.
>>> import tensorflow as tf
>>> x_var = tf.Variable(0., name='x_result')
>>> # Schrittzähler
... step_var = tf.Variable(0, trainable=False)
>>> # Verlust
... loss = 2 * tf.multiply(x_var, x_var) -
tf.multiply(3.0, x_var) + 5.0
>>> # Variablenwert finden, der Verlust minimiert
... learn_rate = tf.constant(1.)
>>> num_epochs = 50
>>> optimizer =
tf.train.GradientDescentOptimizer(learn_rate).minimize(
global_step=step_var)
>>> # Variable initialisieren
... init = tf.global_variables_initializer()
>>> # Session starten
... with tf.Session() as sess:
...     sess.run(init)
...     sess.run([step_var, x_var, learn_rate, loss])
...     for epoch in range(num_epochs):
...         sess.run(optimizer)
...         if epoch % 5 == 0:
...             sess.run([step_var, x_var, loss])
...
[0, 0.0, 1.0, 5.0]
[1, 3.0, 14.0]
```

```
[6, -546.0, 597875.0]
[11, 132861.0, 35303694000.0]
[16, -32285040.0, 2084647700000000.0]
[21, 7845265400.0, 1.2309638e+20]
[26, -1906399500000.0, 7.268718e+24]
[31, 463255100000000.0, 4.292106e+29]
[36, -1.1257098e+17, 2.534445e+34]
[41, 2.7354745e+19, inf]
[46, -6.647203e+21, inf]
>>>
```

Verfahren divergiert! learn_rate verkleinern!

learnig_rate=0.1:

```
[0, 0.0, 0.1, 5.0]
[1, 0.3, 4.2799997]
[6, 0.715008, 3.877449]
[11, 0.74727905, 3.8750148]
[16, 0.7497884, 3.875]
[21, 0.74998355, 3.875]
[26, 0.7499987, 3.875]
[31, 0.7499999, 3.875]
[36, 0.74999994, 3.875]
[41, 0.74999994, 3.875]
[46, 0.74999994, 3.875]
=============================
```

learnig_rate=0.01:

```
[0, 0.0, 0.01, 5.0]
```

```
[1, 0.03, 4.9118]
[11, 0.2713205, 4.333268]
[21, 0.43175822, 4.0775557]
[31, 0.5384224, 3.96453]
[41, 0.6093363, 3.9145727]
[51, 0.6564822, 3.892491]
[61, 0.68782634, 3.8827312]
[71, 0.7086649, 3.878417]
[81, 0.7225191, 3.8765106]
[91, 0.73172975, 3.8756676]
[101, 0.73785335, 3.8752952]
[111, 0.7419244, 3.8751302]
[121, 0.7446311, 3.8750577]
[131, 0.7464306, 3.8750255]
[141, 0.74762696, 3.8750114]
[151, 0.74842244, 3.8750048]
[161, 0.74895126, 3.8750021]
[171, 0.74930286, 3.875001]
[181, 0.7495365, 3.8750005]
[191, 0.74969184, 3.8750005]
```

3D-Grafik Paraboloid