LAB REPORT: SW02

TSM_DELEARN

Andrin Bürli[*], Nursinem Dere[†], Fabian Gröger[‡]
Hochschule Luzern

Monday 8th March, 2021

1 Introduction

In the second week of Deep Learning (TSM_DeLearn) course, learning is formulated as optimization problem, to solve this optimization problem gradient descent method, data preparation(normalization, training-test sets), cost functions(MSE, CE) and entropy are introduced with examples. The solutions of exercises are included in the next section of this report.

2 Exercise 1: Numpy in a Nutshell

Show the red, green and blue components of the image in 3 different by using sub-plots.

```
from matplotlib import pyplot as plt
import imageio
import numpy as np
img1 = imageio.imread('sponge-bob.jpg')
figure, plots = plt.subplots(ncols=3, nrows=1)
for i, subplot in zip(range(3), plots):
temp = np.zeros(img1.shape, dtype='uint8')
temp[:,:,i] = img1[:,:,i]
subplot.imshow(temp)
subplot.set_axis_off()
plt.show()
```

Figure 1: Exercise 1.1

[*]andrin.buerli@hslu.ch
[†]nursinem.dere@stud.hslu.ch
[‡]fabian.groeger@hslu.ch

Perform the following distorsion of the image and display it:
- vertical flip
- rotation 90 degrees
- horizontal flip

```
import imageio
import matplotlib.pyplot as plt
import numpy as np
img = imageio.imread('sponge-bob.jpg')
img1 = img[::-1] #vertical flip
img2 = np.rot90(img) #90degree turn
img3 = img[:, ::-1, :] #horizontal flip
f, ax = plt.subplots(nrows=1,ncols=3)
plt.sca(ax[0]);
plt.imshow(img1); plt.title('Vertical flip')
plt.sca(ax[1]);
plt.imshow(img2); plt.title('90-degree turn')
plt.sca(ax[2]);
plt.imshow(img3); plt.title('Horizontal flip')
plt.show()
```

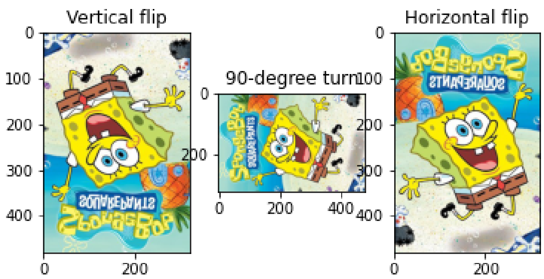

Figure 2: Exercise 1.2

3 Exercise 2: Sigmoid Function

1. Compute the derivative of the sigmoid function $\sigma(z) = \frac{1}{1+e^{-z}}$
   $(1 + e^{-z})\sigma = 1$
   $\Rightarrow -e^{-z}\sigma + (1 + e^{-z})\frac{d\sigma}{dz} = 0$
   $\Rightarrow \frac{d\sigma}{dz} = \sigma.\frac{e^{-z}}{(1+e^{-z})} = \sigma.\frac{(1+e^{-z})-1}{(1+e^{-z})}$
   $\sigma.\left[1 - \frac{1}{(1+e^{-z})}\right] = \sigma(z).(1 - \sigma(z))$

2. Show that the derivative fullfills the equation $\sigma'(z) = \sigma(z) \cdot (1 - \sigma(z))$
   Let $\sigma = \frac{1}{1+e^-z} \Rightarrow \sigma = \frac{e^z}{1+e^z}$
   By the quotient rule, $\frac{d}{dx}\left(\frac{f(x)}{g(x)}\right) = \frac{\frac{d}{dx}f(x)g(x) - f(x)\frac{d}{dx}g(x)}{g^2(x)}$
   we set $f(z) = e^z$ and $g(z) = (1 + e^z)$

By using $f(z)$ and $g(z)$ on the rule, we get $\frac{d\sigma}{dx} = \frac{(e^z)(1+e^z)-e^z e^z}{(1+e^z)^2} = \frac{e^z}{1+e^z} \frac{(1+e^z)-e^z}{1+e^z}$
$= \frac{e^z}{1+e^z}(1 - \frac{e^z}{1+e^z})$
And finally, $\sigma'(z) = \sigma(z)(1 - \sigma(z))$

3. Compute the first and second derivative of $\zeta(z) = -log(\sigma(-z))$
   $\zeta'(z) = \frac{d\zeta(z)}{dz} = \dfrac{\mathrm{e}^z}{\mathrm{e}^z + 1}$
   $\zeta''(z) = \frac{d\zeta'(z)}{dz} = \dfrac{\mathrm{e}^z}{(\mathrm{e}^z + 1)^2}$
   Compute the asymptotes for $z \Rightarrow \mp\infty$
   $\lim_{z \to +\infty} \zeta(z) = -log(\frac{1}{1+e^{\infty}}) = -log(\frac{1}{\infty}) = -log(0)...not finished...$

4. Plot the sigmoid function and its derivative by using matplotlib.

```
import matplotlib.pyplot as plt
z = np.arange(-5., 5., 0.2)
s= 1/(1+np.exp(-z))
plt.plot(z, s, 'r--', label = 'sigmoid')
plt.plot(z, s*(1-s) , 'b--', label = 'derivative')
plt.legend(loc="upper left")
```

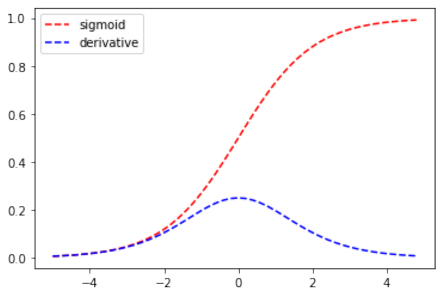

Figure 3: Plot of sigmoid function and its derivative

5. Show that the function $c_1(x) = (\sigma(x) - 1)^2$ is non-convex
   Explain in which situations (initial settings) optimising $c_1(x)$ with gradient descent may become difficult. For the explanation create a plot.

6. Compute the first and second derivative of the function w.r.t. $\omega \in \mathbb{R}$ and for given $y \in \{0, 1\}$
   $c_2(x) = -(ylog(\sigma(w \cdot x)) + (1 - y)log(1 - \sigma(w \cdot x)))$

4 Exercise 3: Gradient Descent for Perceptron

Solutions are listed in *MNIST_Binary_batchgd_stud.ipynb* file.

5 Exercise 4: Review Questions

1. *Explain why normalisation is beneficial.*
Different features do not have similar ranges of values and hence gradients may end up taking a long time and can oscillate back and forth and take a long time before it can finally find its way to the global/local minimum. To overcome the model learning problem, normalize the data and make sure that the different features take on similar ranges of values so that gradient descents can converge more quickly.

2. *In what sense are optimisation techniques important for machine learning problems?*
Machine learning optimization is the process of adjusting the hyperparameters in order to minimize the cost function by using one of the optimization techniques. It is important to minimize the cost function because it describes the discrepancy between the true value of the estimated parameter and what the model has predicted.

3. *Describe what problems gradient descent can be applied to. In what problems will gradient descent lead to a unique solution ? Describe what can go wrong in more general problems and why?*
In machine learning, we use gradient descent to update the parameters to get better results. If the given function is convex, then the solution is unique. If the given function is not convex, then the algorithm may stuck on local minimum other than finding the global value.

4. *Why is learning with MSE cost considered less suitable for classification problems ?*
The MSE function is non-convex for binary classification. If a binary classification model is trained with MSE Cost function, it is not guaranteed to minimize the Cost function. This is because MSE function expects real-valued inputs in range($-\infty$, $+\infty$), while binary classification models output probabilities in range (0,1) through the sigmoid/logistic function.

5. *What may happen if the learning rate is chosen too large ?*
If the learning rate is too large, it can jump over the minima we are trying to reach, we overshoot. This can lead to osculations around the minimum or in some cases to outright divergence.

6. *Why is it possible that the test error starts increasing after some epochs of training?*
Yes, test error may start increasing with the number of epochs of training since it may lead to overfitting.

7. *Is it becoming more or less difficult to reach small values of the cost function if we have more training data?*
If we have large amount of training data, it will be more difficult for the optimization algorithm to convergence to the small values of the cost function.