Machine Learning

Summer Semester 2019, Homework 1

Prof. Dr. J. Peters, H. Abdulsamad, S. Stark, D.Koert

TECHNISCHE UNIVERSITÄT DARMSTADT

Total points: 90 + 10 bonus
Due date: Sunday, 26 May 2019 (before midnight)
Hand in a PDF over Moodle **and** a printed version to the postbox at (S2/02 | E315)

Name, Surname, ID Number

Problem 1.1 Linear Algebra Refresher [20 Points]

a) **Matrix Properties [5 Points]**

A colleague of yours suggests matrix addition and multiplication are similar to scalars, thus commutative, distributive and associative properties can be applied. Prove if matrix addition and multiplication are commutative and associative analytically or give counterexamples. Is matrix multiplication distributive with respect to matrix addition? Again, prove it analytically or give a counterexample. Considering three matrices A, B, C of size $n \times n$.

To answer the questions examples calculated by following matrices will be used:

$$A = \begin{bmatrix} 2 & -1 \\ 0 & 1 \end{bmatrix} \quad B = \begin{bmatrix} 1 & 0 \\ 4 & 3 \end{bmatrix} \quad C = \begin{bmatrix} 2 & 2 \\ 2 & 2. \end{bmatrix}$$

The commutative property for matrix addition states: $A + B = B + A$.

$$A + B = \begin{pmatrix} 3 & -1 \\ 4 & 4 \end{pmatrix} \tag{5}$$

$$B + A = \begin{pmatrix} 3 & -1 \\ 4 & 4 \end{pmatrix} = A + B \tag{6}$$

The commutative property for matrix multiplacation states: $A \cdot B = B \cdot A$

$$A \cdot B = \begin{bmatrix} -2 & -3 \\ 4 & 3 \end{bmatrix} \tag{7}$$

$$B \cdot A = \begin{bmatrix} 2 & -1 \\ 8 & -1 \end{bmatrix} = A \cdot B \tag{8}$$

Thus $A \cdot B \neq B \cdot A$

The distributiv property for matrices states: $A \cdot B + A \cdot C = A(B + C)$

$$\begin{bmatrix} -2 & -3 \\ 04 & 3 \end{bmatrix} + \begin{bmatrix} 2 & 2 \\ 2 & 2 \end{bmatrix} = \begin{bmatrix} 2 & -1 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 3 & 2 \\ 6 & 5 \end{bmatrix}$$

$$\begin{bmatrix} 0 & -1 \\ 6 & 5 \end{bmatrix} = \begin{bmatrix} 0 & -1 \\ 6 & 5 \end{bmatrix}$$

b) **Matrix Inversion [6 Points]**

Given the following matrix

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 1 & 2 & 4 \\ 1 & 4 & 5 \end{pmatrix}$$

analytically compute its inverse A^{-1} and illustrate the steps.
If we change the matrix in

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 1 & 2 & 4 \\ 1 & 2 & 5 \end{pmatrix}$$

is it still invertible? Why?

First the determinant of the matrix is calculated using the Rule of Sarrus

$$det\,(a) = 10 + 8 + 12 - 16 - 10 - 6 = -2 \neq 0 \tag{22}$$

After that we can calculate the Hauptminors of the matrix

$$det\left(A_{1,1}\right) = -6 \tag{23}$$

$$det\left(A_{1,2}\right) = -1 \tag{24}$$

$$det\left(A_{1,3}\right) = 2 \tag{25}$$

$$det\left(A_{2,1}\right) = -2 \tag{26}$$

$$det\left(A_{2,2}\right) = 2 \tag{27}$$

$$det\left(A_{2,3}\right) = 2 \tag{28}$$

$$det\left(A_{3,1}\right) = 2 \tag{29}$$

$$det\left(A_{3,2}\right) = 1 \tag{30}$$

$$det\left(A_{3,3}\right) = 0 \tag{31}$$

Using the Rule of Cramer

$$x_i = \frac{det\left(A_i\right)}{det\,(A)} \forall i \tag{32}$$

We calculate:

$$A^{-1} = \begin{bmatrix} 3 & -1 & -1 \\ 1/2 & -1 & 1/2 \\ -1 & 1 & 0 \end{bmatrix} \tag{33}$$

If we now change the matrix, than it's not invertible anymore, because

$$det\,(A) = 10 + 8 + 6 - 8 - 10 - 6 = 24 - 24 = 0 \tag{34}$$

c) **Matrix Pseudoinverse [3 Points]**

Write the definition of the right and left Moore-Penrose pseudoinverse of a generic matrix $A \in \mathbb{R}^{n \times m}$.
Given $A \in \mathbb{R}^{2 \times 3}$, which one does exist? Write down the equation for computing it, specifying the dimensionality of the matrices in the intermediate steps.

Definition of the left Pseudoinverse:

$$J^* J = \left(J^T J\right)^- 1 J^T \cdot J = I_m \tag{39}$$

Definition of the right Pseudoinverse:

$$J J^* = J \cdot J^T \left(J J^T\right)^- 1 \cdot = I_m \tag{40}$$

Given: $A \in \mathfrak{R}^{2x3}$ with $m > n$ which implies full row rank we use the right Pseudoinverse

$$A^* = \underbrace{\underbrace{A^T}_{3x2} \cdot \underbrace{\left(J J^T\right)^{-1}}_{2x2}}_{3x2} \tag{41}$$

$$\implies A^* \in \mathfrak{R}^{3x2} \tag{42}$$

d) **Eigenvectors & Eigenvalues [6 Points]**

What are eigenvectors and eigenvalues of a matrix A? Briefly explain why they are important in Machine Learning.

In general for Eigenvektors the following equation is true:

$$A \cdot v = \lambda \cdot v \tag{48}$$

Example for calculating the Eigenvalues. We start be calculating the characistic Polynom of the matrix.

$$det\,(A - E \cdot \lambda = 0) \tag{49}$$

$$det \begin{bmatrix} 1-\lambda & 2 & 3 \\ 1 & 2-\lambda 4 & \\ 1 & 4 & 5-\lambda \end{bmatrix} = -\lambda^3 + 8\lambda^2 + 4\lambda - 2 \tag{50}$$

This equation can now be solved to get the Eigenvalues. If the Eigenvalues are now placed back into the Matrix it is possible to calculate the corresponding Eigenvectors.

Eigenvectors and values are Vectors and Scalars for which

$$W \cdot v = \lambda v \tag{51}$$

holds true.

They're defined individually for each Transformation Matrix W. If we now want to change the length of an Eigenvector we don't need to multiply it with W, instead we just multiply it with a corresponding Eigenvalue. Since this is a scalar instead of a Matrix, it saves computational power fo this calculation. Actually any transformation W applied to an vector can be seen as a lineare combination of eigenvectors.

$$\boldsymbol{u} = W v = c_1 \lambda_1 \cdot v_1 + ... + c_n \lambda_n \cdot v_n \tag{52}$$

This again is a huge time saver.

Eigenvalues also tell us about the numerical stability of a Matrix transformation. To achieve a vanishing matrix Eigenvalues of $\leq$ 1are needed. If the Eigenvalues are bigger than one, than the matrix will explode. Ideally the Eigenvalues are all equal to one. The Markov Matrix is one of these matrices and thus used in machine learning.

This knowledge is also used in the orthogonal weight initialization method for Neural Networks.

Name, Surname, ID Number

Problem 1.2 Statistics Refresher [25 Points]

a) **Expectation and Variance [8 Points]**

Let Ω be a finite set and $P : \Omega \to \mathbb{R}$ a probability measure that (by definition) satisfies $P(\omega) \geq 0$ for all $\omega \in \Omega$ and $\sum_{\omega \in \Omega} P(\omega) = 1$. Let $f : \Omega \to \mathbb{R}$ be an arbitrary function on Ω.

1) Write the definition of expectation and variance of f and discuss if they are linear operators.

2) You are given a set of three dices $\{A, B, C\}$. The following table describes the outcome of six rollouts for these dices, where each column shows the outcome of the respective dice. (Note: assume the dices are standard six-sided dices with values between 1-6)

A	4	4	2	4	1	1
B	3	6	3	3	4	3
C	5	5	2	1	1	1

Estimate the expectation and the variance for each dice using unbiased estimators. (Show your computations).

3) According to the data, which of them is the "most rigged"? Why?

b) **It is a Cold World [7 Points]**

Consider the following three statements:
a) A person with a cold has backpain 24% of the time.
b) 5% of the world population has a cold.
c) 12% of those who do not have a cold, still have backpain.

1) Identify random variables from the statements above and define a unique symbol for each of them.
2) Define the domain of each random variable.
3) Represent the three statements above with your random variables.
4) If you suffer from backpain, what are the chances that you suffer from a cold? (Show all the intermediate steps.)

c) **Journey to THX1138 [10 Points]**

After the success of the Rosetta mission, ESA decided to send a spaceship to rendezvous with the comet THX1138. This spacecraft consists of four independent subsystems A, B, C, D. Each subsystem has a probability of failing during the journey equal to $1/3$.
1) What is the probability of the spacecraft S to be in working condition (i.e., all subsystems are operational at the same time) at the rendezvous?
2) Given that the spacecraft S is not operating properly, compute analytically the probability that **only** subsystem A has failed.
3) Instead of computing the probability analytically, do a simple simulation experiment and compare the result to the previous solution. Include a snippet of your code.
4) An improved spacecraft version has been designed. The new spacecraft fails if the critical subsystem A fails, or any two subsystems of the remaining B, C, D fail. What is the probability that **only** subsystem A has failed, given that the spacecraft S is failing?

Problem 1.3 Optimization and Information Theory [45 Points + 10 Bonus]

a) **Entropy [5 Points]**

You work for a telecommunication company that uses a system to transmit four different symbols S_1, S_2, S_3, S_4 through time. In the current system, each symbol has a probability to occur according to the following table

	S_1	S_2	S_3	S_4
p_i	0.05	0.61	0.27	0.07

Compute the entropy of the system and write the minimum number of bits requires for transmission.

b) **Constrained Optimization [25 Points]**

After an upgrade of the system, your boss asks you to change the probabilities of transmission in order to maximize the entropy. However, the new system has the following constraint

$$4 = \sum_{i=1}^{4} 2 p_i i.$$

1) Formulate it as a constrained optimization problem. Do you need to include additional constrains beside the one above?
2) Write down the Lagrangian of the problem. Use one Lagrangian multiplier per constraint.
3) Compute the partial derivatives of the Lagrangian above for each multiplier and the objective variable. Is it easy to solve it analytically?
4) Formulate the dual function of this constrained optimization problem. Solve it analytically.
5) Name one technique for numerically solve these problems and briefly describe it.

c) **Numerical Optimization [10 Points]**

Rosenbrock's function (to be minimized) is defined as

$$f(\boldsymbol{x}) = \sum_{i=1}^{n-1} \left[100 \left(x_{i+1} - x_i^2 \right)^2 + \left(x_i - 1 \right)^2 \right].$$

Write in Python a simple gradient descent algorithm and simulate it for 10,000 steps on Rosenbrock's function with $n = 20$. Attach a snippet of your algorithm, discuss the effects of the learning rate and attach a plot of your learning curve with your best learning rate.

Choosing the right learning rate was tricky, because choosing it to high or to low would result in exploding gradients. From all tested cases, a learning rate between 0.001 and 0.0001 worked best.

```
for k in range(max_iteration):
    prev_x = cur_x
    cur_x = prev_x - learning_rate * rosen_der(prev_x)
    error = abs(cur_x - prev_x)
    k = k + 1
    history[k, :] = np.linalg.norm(error)
```

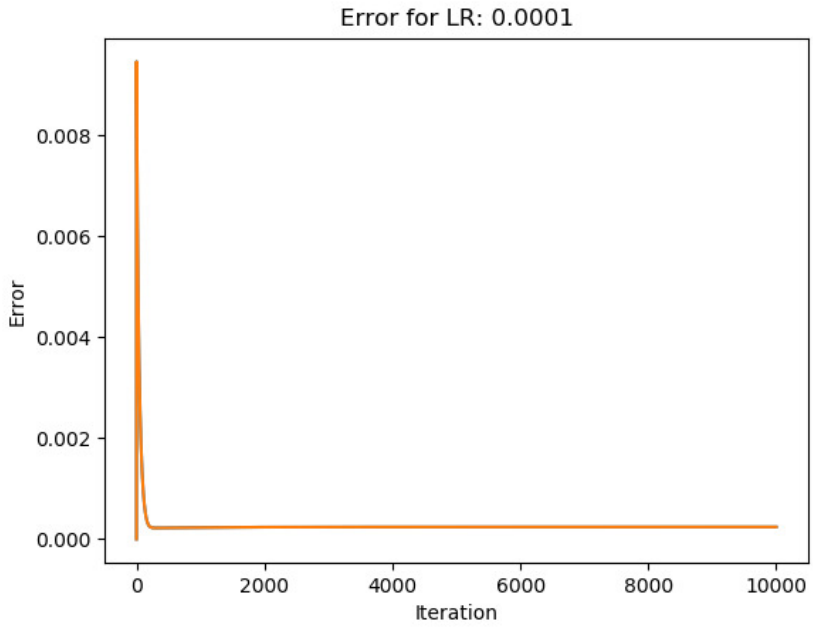

Figure 1: Best Learning Rate was 0.0001

d) **Gradient Descent Variants [5 Points]**

Throughout this class we have seen that gradient descent is one of the most used optimization techniques in Machine Learning. This question asks you to deepen the topic by conducting some research by yourself.

1) There are several variants of gradient descent, namely *batch, stochastic* and *mini-batch*. Each variant differs in how much data we use to compute the gradient of the objective function. Discuss the differences among them, pointing out pros and cons of each one.

2) Many gradient descent optimization algorithms use the so-called *momentum* to improve convergence. What is it? Is it always useful?

Batch gradient descent *in machine Learning calculates the gradient using the whole dataset. For this the computer needs to have the whole dataset in memory, which is not always doable. Because of that, its also very time consuming. It converts against the global Minimum of konvex functions and local Minimums of non konvex functions.*

Stochastic gradient descent *calculates the gradient for a single traingingexample. Because of this stochastic gradient descent learns a lot faster than batch gradient descent, however its variance is very high, resulting in highly fluctuating costfunction.*

Mini-batch gradient descent *can be seen as an combination between stocahstic gradient descent and batch gradient descent, using mini-batches containing n training examples. This results in reduced variance and more stable conversion rate. When mini-batch gradient descent is calculated on gpus, it is possible to reduce the workload by using matrix operations.*

Momentum *is often used to help the gradient descent algorithm to find the global Minimum.*

Without Momentum there is a high possibility, that GD starts oscillating around a local minimum. In order to reach the global Minimum, this local one needs to be "jumped over". This can be interpeted as a physical impulsive giving a downhill rolling sphere an impuls of energy, to help it overcome obstacles. This "impuls" is usually calculated by looking at the last gradient and adding a part of it to the current gradient. This means, that if the last update to our parameters was big, we guess that the next one will be big, too.

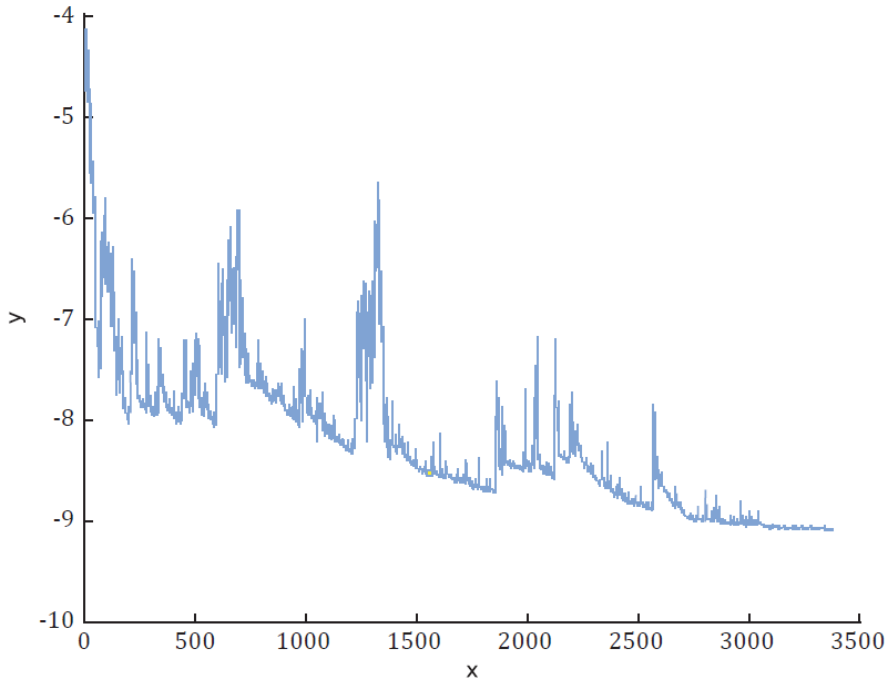

Figure 2: Example for fluctuating cost function for a Neural Network when using SGD [Source: Me]

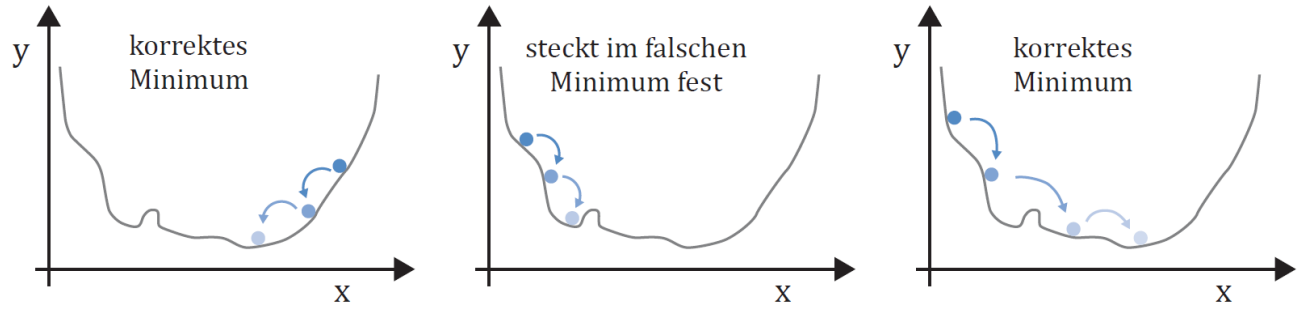

Figure 3: Physically interpretation of Momentum [Source: Me]

e) **Natural Gradient [10 Bonus Points]**

Let $\theta \in \mathbb{R}^n$ be a parameter vector and $J : \mathbb{R}^n \to \mathbb{R}$ a cost function. The negative gradient $-\nabla J(\theta)$ is sometimes called the *steepest descent direction*. But is it really? To be able to claim that it is *the* steepest descent direction, we should compare it to other descent directions and pinpoint what is so unique about the negative gradient direction.

Covariant gradient. A fair way to compare descent directions is to make a small step of fixed length, say ε, in every direction $\Delta\theta$ and check which direction leads to the greatest decrease in $J(\theta)$. Since we assume that the step size is small, we can evaluate the decrease in $J(\theta)$ using its first-order Taylor approximation

$$J(\theta + \Delta\theta) - J(\theta) \approx \nabla J(\theta)^T \Delta\theta.$$

To make precise what we mean by *small* step size, we need to introduce a norm (or a distance) in the space of parameters θ. A good choice, that among other advantages captures the intuition that some parameters may influence the objective function more than others, is the generic quadratic norm

$$\|\Delta\theta\|^2 = \frac{1}{2}\Delta\theta^T F(\theta)\,\Delta\theta$$

with a positive-definite matrix $F(\theta)$; note that in general F may depend on θ.

1) Find the direction $\Delta\theta$ that yields the largest decrease in the linear approximation of $J(\theta)$ for a fixed step size ε. Does this direction coincide with $-\nabla J(\theta)$? The direction that you found is known as the negative covariant gradient direction.

Natural gradient. In statistical models, parameter vector θ often contains parameters of a probability density function $p(x;\theta)$ (for example, mean and covariance of a Gaussian density); thus, the cost function J depends on θ indirectly through $p(x;\theta)$. This two-level structure gives a strong hint to what matrix F to pick for measuring the distance in the parameter space in the most *natural* way. Namely, one can carry over the notion of 'distance' between probability distributions $p(x;\theta+\Delta\theta)$ and $p(x;\theta)$ (which is known from information theory to be well captured by the Kullback-Leibler divergence) to the distance between the corresponding parameter vectors $\theta+\Delta\theta$ and θ.

2) Obtain the quadratic Taylor approximation of the KL divergence from $p(x;\theta)$ to $p(x;\theta+\Delta\theta)$ in the form

$$KL\left(p(x;\theta+\Delta\theta)\,||\,p(x;\theta)\right) \approx \frac{1}{2}\Delta\theta^T F(\theta)\,\Delta\theta.$$

Covariant gradient with the matrix $F(\theta)$ that you found is known as the natural gradient.