Assignment-1

Pattern and Speech Recognition

November 10, 2016

1 Linear Algebra

1 Let M be a real symmetric matrix. Show that $\frac{x^T M x}{||x||_2^2}$ is upper and lower bounded by λ_{max} and λ_{min} (where λ_{max} and λ_{min} are the largest and smallest eigenvalues of M)(*2 points*)

2 Let M be a real symmetric matrix. Using the properties of eigendecompsition, present a method to compute M^{100} elegently (*2 points*)

3

$$\begin{bmatrix} 2 & -1 & -1 \\ -1 & 3 & -1 \\ -1 & -1 & x \end{bmatrix}$$

(i) Find a value for x such that the above matrix is positive definite. (*1 point*)
(ii) Find a value for x such that the above matrix is rank-2 (*1 point*)

4

$$\begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -5 & 3 & y \end{bmatrix}$$

(i) Find a value for y such that the vector $(1, 1, 1)$ is in the null space of above matrix. (*1 point*)
(ii) Find a value for y such that sum of its eigenvalues is 0. (*1 point*)

5 Find a non-trivial upper and lower bound for $||x||_1$ in terms of $||x||_\infty$ (*2 points*)

2 Probability Theory

1 In an experiment of tossing a fair die, let $A = \{2, 4, 6\}$ and $B = \{1, 2, 3, 4\}$ be two events. Check whether A and B are independent. If $C = \{1, 3, 5\}$, check for independency of A and C. (*2 points*)

2 Assume 30% of computer owners use Macintosh, 50% use Windows and rest use Linux. Suppose that 65% of Mac users have succumbed to a computer virus, 82% of the Windows users get the virus and 50% of the Linux users get the virus. We select a person at random and learn that his/her system is infected with the virus. What is the probability that he/she is a Windows user? (*2 points*)

3

$$f(x) = \begin{cases} 0 & \text{if } x < 0 \\ \frac{1}{1+x} & \text{otherwise} \end{cases}$$

$$g(x) = \begin{cases} 0 & \text{if } x < 0 \\ \frac{1}{(1+x)^2} & \text{otherwise} \end{cases}$$

Check whether f and g are valid probability density functions(PDF). If it is a valid distribution, compute its mean and report your observation. (*3 points*)

4 Let X be a continuous random variable uniformly distributed between (0,1). Let $0 < a < b < 1$

$$Y = \begin{cases} 1 & \text{if } 0 < x < b \\ 0 & \text{otherwise} \end{cases}$$

and

$$Z = \begin{cases} 1 & \text{if } a < x < 1 \\ 0 & \text{otherwise} \end{cases}$$

Are Y and Z independent? Why/Why not? (*2 points*)

5 Let X and Y be two independent random variables. Show that $E[XY] = E[X]E[Y]$. Also compute the covariance between X and Y. (*2 points*)

3 Multivariable Calculus

1 Let $f(x, y) = x^3 - 3xy^2$. Find all the extermal points for the function f and report if they are local/global maxima/minima or saddle point. Also, report the maximum and minimum value f can take. (*3 points*)

Universität des Saarlandes
FR 4.7 Sprach- und Signalverarbeitung

Prof. Dietrich Klakow
Pattern and Speech Recognition

Exercise Sheet 2

PCA and Numerical Computation

Deadline: 22.11.2016, 23:56

PCA

Exercise 2.1 *(2 + 2 + 2 = 6 points)*

In this exercise we will exploit the PCA in order to compress a 2-dimensional dataset into a 1-dimensional set. Do not forget to normalize your data such that the mean becomes 0, when taking the eigenvalue approach as described in the lecture for part a) and b).

a) Consider the following dataset consisting of 4 2-dimensional vectors:

$$\mathbf{x}^{(1)} = (1,1)^T,\ \mathbf{x}^{(2)} = (2,2)^T,\ \mathbf{x}^{(3)} = (3,1)^T,\ \mathbf{x}^{(4)} = (4,1)^T.$$

Compress this dataset to a 1-dimensional set using the PCA i.e. derive the encoder function $f(\mathbf{x}) = \mathbf{D}^T \cdot \mathbf{x}$ as defined in the lecture. Then apply f to the dataset in order to compress it.

b) Now consider the set:

$$\mathbf{x}^{(1)} = (-1,1)^T,\ \mathbf{x}^{(2)} = (-2,2)^T,\ \mathbf{x}^{(3)} = (-1,3)^T,\ \mathbf{x}^{(4)} = (-1,4)^T.$$

As in part a) compress this set by deriving the encoder function f and apply it to the set.

c) For both the parts a) and b) sketch the corresponding datasets in a separate figure. Also include the reconstructed vectors into the corresponding figures. Explain the values of the reconstructed vectors.

Numerical Computation

Exercise 2.2 *(2 + 2 = 4 points)*

In the lecture you encountered the function

$$\text{softmax}(\mathbf{x})_i = \frac{\exp(x_i)}{\sum\limits_{j=1}^{n} \exp(x_j)}$$

and learned about numerical issues that might arise when computing this function on a computer which can only compute this up to a certain precision.

a) Name the proposed approach to circumvent the mentioned over- and underflows and show why this approach prevents the stated problems.
Hint: Consider cases where the entries of $\mathbf{x}$ *are in the same range of some extreme.*

b) The proposed approach takes care of some over- and underflows. However using this scheme still over- or underflows can occur. State where we might still might encounter problems and give an example which makes softmax useless even when using the proposed scheme. I.e. your computation using softmax would lead to an undefined result ($-\infty$ or ∞).

Exercise 2.3 *(2 + 2 + 1 = 5 points)*

Consider the following function

$$f(x, y) = 20 \cdot x^2 + \frac{1}{4} \cdot y^2.$$

a) Implement a python script `gradient_descent.py` which tries to find the minimum of f using the method of gradient descent as discussed in the lecture. As a starting point use $\mathbf{x} = (-2, 4)^T$ and as learning rate $\epsilon = 0.04$.

b) In which way does your implemented script find the minimum of f? Draw a sketch or plot of f (e.g. as a contour plot) and indicate which intermediate points $\mathbf{x}$ are computed to find the minimum. Is this path optimal? Justify your answer and explain why the path looks like that.

c) What happens if you change the learning rate to $\epsilon = 0.1$? Briefly describe and explain your observations.

Exercise 2.4 *(3 + 1 = 4 points)*

Consider the function f from exercise 3.

a) Implement a python script `newton.py` that finds the minimum of f by incorporating the Hessian matrix and using newtons method as discussed in the lecture.

b) Compare the path of finding the minimum to the one from exercise 3.

Exercise 2.5 *(1 points)*

Design a meaningful stopping criterion for stopping the iterations from exercise 3 or 4.

Submission instructions

The following instructions are mandatory. If you are not following them, tutors can decide to not correct your exercise.

Submission architecture

You have to generate a **single ZIP file** respecting the following architecture:

```
tutorial2_<matriculation_nb1>_<matriculation_nb2>_<matriculation_nb3>
|
+--- source
|      |
|      +------- file 1
|      +------- file 2
|      +------- ...
+--- rapport.pdf
+--- README.txt
```

where

- **source** contains the source code of your project,
- **rapport.pdf** is the report where you present your solution with **the explanations** and the plots,
- **README** which contains group member informations (name, matriculation numbers and emails) and a **clear** explanation about how to compile and run your source code

The ZIP filename has to be :

`tutorial2_<matriculation_nb1>_<matriculation_nb2>_<matriculation_nb3>.zip`

Some hints

We advice you to follow the following guidelines in order to avoid problems :

- Avoid building complex systems. The exercises are simple enough.
- Do not include any executables in your submission, as this will cause the e-mail server to reject it.

Grading

Send your assignment to the tutor who is responsible of your group:

- Merlin Köhler s9mnkoeh@stud.uni-saarland.de
- Yelluru Gopal Goutam goutamyg@lsv.uni-saarland.de
- Ahmad Taie ataie@lsv.uni-saarland.de

The email subject should start with `[PSR TUTORIAL 2]`

Universität des Saarlandes
FR 4.7 Sprach- und Signalverarbeitung

Prof. Dietrich Klakow
Pattern and Speech Recognition

Exercise Sheet 3

Machine Learning Basics

Deadline: 29.11.2016, 23:59

Linear Regression

Exercise 3.1 *(17 points)*

In this exercise we will use Linear Regression to estimate a model for our data.

a) Download the dataset Auto MPG from here:

`https://archive.ics.uci.edu/ml/datasets/Auto+MPG`

Load the first and third columns (mpg and displacement) which represent the Miles per galon and Engine displacement in cubic inches of various car models. They will act as our $\mathbf{X}$ and $\mathbf{Y}$ respectively for this exercise. Use the first **50** point pairs as a training set, and the next **50** will be the test set. *(1 point)*

b) Plot the training data points. What is your estimation for the degree of the polynomial that would approximate this data? *(1 point)*

c) We will fit a first order model to this training set. State the equation for such a model and the gradient descent update rule. *(2 points)*

d) Implement gradient descent to find the parameters of the model. Choose a suitable learning rate and initialization. Plot the value of the cost function per epoch. *(4 points)*

e) Using the parameters that your algorithm converged to, plot the model over the data. What is final value of the cost function? *(1 point)*

f) Now we will fit a second order model. Again, state the equations for the model and the gradient descent. *(1 point)*

g) Find the parameters of that model using gradient descent. Plot your model over the data and also plot the value of the cost function per epoch. What is the final value of the cost function in this case? Is it expected? *(3 points)*

h) Use numpy's polyfit in Python, or a similar function, to fit a 9th degree polynomial. We will use that polynomial to evaluate the performance of our model (You can use numpy's polyval). *(1 point)*

i) Use the 3 models you estimated (first, second, and 9th order polynomials) to evaluate the prediction values $\mathbf{Y}$ for the test set. Calculate the mean square error (MSE as defined in the lecture) for each case. What do you observe? *(3 points)*

Regularization

Exercise 3.2 *(3 points)*

In the lecture (slide 15) We saw the equation.

$$\mathrm{J}(\mathbf{w}) = \mathrm{MSE}_{\mathrm{train}} + \lambda\, \mathrm{w}^{\top}\, \mathrm{w}$$

- Drive a closed form expression for the regularization term given **w** from slide 9 in the lecture.

Submission instructions

The following instructions are mandatory. If you are not following them, tutors can decide to not correct your exercise.

Submission architecture

You have to generate a **single ZIP file** respecting the following architecture:

```
tutorial2_<matriculation_nb1>_<matriculation_nb2>_<matriculation_nb3>
|
+--- source
|      |
|      +------- file 1
|      +------- file 2
|      +------- ...
+--- report.pdf
+--- README.txt
```

where

- **source** contains the source code of your project,
- **report.pdf** is the report where you present your solution with **the explanations** and the plots.
- **README** which contains group member informations (name, matriculation numbers and emails) and a **clear** explanation about how to compile and run your source code

The ZIP filename has to be :

`tutorial2_<matriculation_nb1>_<matriculation_nb2>_<matriculation_nb3>.zip`

Some hints

We advice you to follow the following guidelines in order to avoid problems :

- Avoid building complex systems. The exercises are simple enough.
- Do not include any executables in your submission, as this will cause the e-mail server to reject it.

Grading

Send your assignment to the tutor who is responsible of your group:

- Merlin Köhler s9mnkoeh@stud.uni-saarland.de
- Yelluru Gopal Goutam goutamyg@lsv.uni-saarland.de
- Ahmad Taie s8ahtaie@stud.uni-saarland.de

The email subject should start with `[PSR TUTORIAL 3]`

Universität des Saarlandes
FR 4.7 Sprach- und Signalverarbeitung

Prof. Dietrich Klakow
Pattern and Speech Recognition

Exercise Sheet 4

Machine Learning Basics

Deadline: 6.12.2016, 23:59

k-Means Clustering

Exercise 4.1

In this exercise, you will implement k-means clustering algorithm.

a) Download the data from course website. Load *data_kmeans.txt* and plot the 2 dimensional datapoints. (1 point)

b) Implement k-means algorithm as follows: (6 points)

Let X = $\{x_1, x_2 ... x_n\}$ be the set of datapoints, and C = $\{c_1 ... c_k\}$ be the cluster centers (initialized randomly). Implement the steps described in slides and iterate till cluster centers don't change OR the objective J doesn't change, where

$$J = \sum_{i=1}^{k} \sum_{x \in c_i} ||x - c_i||^2$$

(Refer: https://en.wikipedia.org/wiki/K-means_clustering for more details)

c) Plot the clustering results for k=2. Use different colors to represent the clusters (1 point)

d) What is the smallest k for which J attains the value zero? (Present a theoritical arguement, don't run your code for different values of k)(1 point)

Maximum Likelihood Estimation

Exercise 4.2

A football team scores 2,3,0,2,1 and 5 goals in six matches played. Assuming these samples are drawn from a Poission distribution, find the maximum likelihood estimate for the parameter λ.(Match outcomes are independent of each other) What is the probability that the team will score 2 goals in the next match? (3 points)

Composite functions

Exercise 4.3

Compute the first and second order partial derivatives for the following function $f(x, y) = \log(\sin(xy))$ (3 points)

Classification

Exercise 4.4

In this task, you will use logistic regression to classify Iris plants as into two categories: Setosa and Virginica based on the sepal length and petal width of flowers. (5 points)

Download the Iris dataset from course website. (filename: *iris.data*)

- Delete the last 50 rows, i.e. data corresponding to Versicolor class.
- Use column-1 (sepal length) and column-4 (petal width) from the file as features. The last column contains the ground truth data.
- Modify the code *logistic_regression.py* provided on the course website and use it for the classification task.
- Plot the classification result and decision boundary.

Submission instructions

The following instructions are mandatory. If you are not following them, tutors can decide to not correct your exercise.

Submission architecture

You have to generate a **single ZIP file** respecting the following architecture:

```
tutorial2_<matriculation_nb1>_<matriculation_nb2>_<matriculation_nb3>
|
+--- source
|      |
|      +------- file 1
|      +------- file 2
|      +------- ...
+--- report.pdf
+--- README.txt
```

where

- **source** contains the source code of your project,

- **report.pdf** is the report where you present your solution with **the explanations** and the plots.
- **README** which contains group member informations (name, matriculation numbers and emails) and a **clear** explanation about how to compile and run your source code

The ZIP filename has to be :

`tutorial2_<matriculation_nb1>_<matriculation_nb2>_<matriculation_nb3>.zip`

Some hints

We advice you to follow the following guidelines in order to avoid problems :

- Avoid building complex systems. The exercises are simple enough.
- Do not include any executables in your submission, as this will cause the e-mail server to reject it.

Grading

Send your assignment to the tutor who is responsible of your group:

- Merlin Köhler s9mnkoeh@stud.uni-saarland.de
- Goutam Y G goutamyg@lsv.uni-saarland.de
- Ahmad Taie s8ahtaie@stud.uni-saarland.de

The email subject should start with `[PSR TUTORIAL 4]`

Universität des Saarlandes
FR 4.7 Sprach- und Signalverarbeitung

Prof. Dietrich Klakow
Pattern and Speech Recognition

Exercise Sheet 5

Neural Networks

Deadline: 15.12.2016, 23:56

Exercise 5.1 *(2+1 = 3 points)*

In the lecture you have encountered the following activation functions:

- $\mathrm{ReLU}(z) = \max\{0, z\}$
- $\mathrm{g}(z) = \frac{1}{1+\mathrm{e}^{-z}}$

a) Compute the derivative of each of the functions.

b) What do you notice for $\mathrm{g}'(z)$? Which influence might this have for backpropagation?

Exercise 5.2 *(5 + 12 = 17 points)*

In this exercise we want to implement a simple Neural Network. For the functions `predict` and `train` you are not allowed to use an existing implementation.

You can assume that we only deal with fully connected networks as indicated below in figure 1. Furthermore, for simplicity, you can assume that we have only one hidden layer in the network as indicated in figure 1. Therefore you do not have to implement nodes explicitly, i.e. it suffices if you describe the layers by a matrix $\mathbf{W_h}$ and $\mathbf{W_o}$ containing the weights of nodes from the hidden layer and the output layer respectively. Furthermore, you will have a vector $\mathbf{b_h}$ containing the biases of the hidden nodes and $\mathbf{b_o}$, containing the biases of the output layer nodes.

Your task now is to implement a neural network in a script `nn.py`, based on the description above. The implementation should provide the following functionalities:

a) `predict(x)`:
The function `predict` should compute the output of your network. This reflects a Forward Propagation through the network. The output of `predict` is the computed vector $\hat{\mathbf{y}}$ for the input $\mathbf{x}$. As an activation function use the sigmoid function

$$\sigma(x) = \frac{1}{1 + \mathrm{e}^{-x}}.$$

b) `train(X, epochs, `ϵ`)`:
The function `train` determines the parameters of the network using the training set `X` consisting of input vectors $\mathbf{x}$ and corresponding output vectors $\mathbf{y}$. To do so, proceed as follows:

For each sample $\mathbf{x}$ in your training data you should

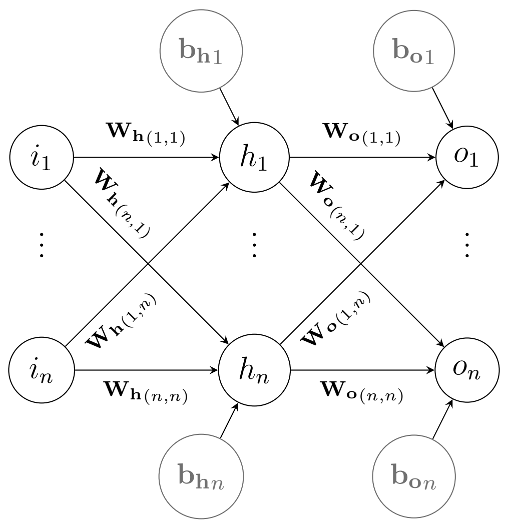

$\mathbf{b}_{\mathbf{h}1}$
$\mathbf{b}_{\mathbf{o}1}$
$\mathbf{W}_{\mathbf{h}(1,1)}$
$\mathbf{W}_{\mathbf{o}(1,1)}$
i_1
h_1
o_1
$\mathbf{W}_{\mathbf{h}(n,1)}$
$\mathbf{W}_{\mathbf{o}(n,1)}$
$\mathbf{W}_{\mathbf{h}(1,n)}$
$\mathbf{W}_{\mathbf{o}(1,n)}$
i_n
h_n
o_n
$\mathbf{W}_{\mathbf{h}(n,n)}$
$\mathbf{W}_{\mathbf{o}(n,n)}$
$\mathbf{b}_{\mathbf{h}n}$
$\mathbf{b}_{\mathbf{o}n}$

Figure 1: Exemplary Neural Network.

- compute the value $\hat{\mathbf{y}}$ predicted by the network.
- adjust your parameters based on the error using backpropagation. Here you try to minimize the obtained error

$$\text{MSE} = \frac{1}{2}\sum_{i=1}^{n} (\hat{\mathbf{y}}_i - \mathbf{y}_i)^2 .$$

  with respect to each of the individual weights in $\mathbf{W_h}$ and $\mathbf{W_o}$, where $\mathbf{y}$ is the true value corresponding to $\mathbf{x}$. I.e. you compute

$$\frac{\partial}{\partial \mathbf{W}_{\mathbf{h}(i,j)}} \text{MSE} \text{ and } \frac{\partial}{\partial \mathbf{W}_{\mathbf{o}(i,j)}} \text{MSE}$$

  for every (i, j), to obtain the slope of MSE for each (i, j). Using this information, update each of the weights using the gradient descent method with learning rate ϵ. So your updated weight $\mathbf{W}_{\mathbf{m}(i,j)}$, where $\mathbf{m} \in \{h, o\}$ becomes

$$\mathbf{W}_{\mathbf{m}(i,j)} = \mathbf{W}_{\mathbf{m}(i,j)} - \epsilon \cdot \frac{\partial}{\partial \mathbf{W}_{\mathbf{m}(i,j)}} \text{MSE} .$$

Similarly you have to update the biases of your model.

After you have adjusted the parameters for each sample in $\mathbf{X}$ you have completed a so called epoch.

As you have no parameters available during the first epoch initialize those randomly.

After each epoch you should compute the error your network produces using the adjusted weights on the training data again using the MSE (now for all samples in your training data).

In total your algorithm should run `epochs` epochs.

c) Provide a plot of the cost for each of the epochs. I.e. the x-axis denotes the epoch and the y-axis denotes the error.

As training data use the `mnist` dataset. The python script provided on the webpage shows an implementation of training a neural network to classify the data provided by the dataset. You can use this script and modify it for your purposes, i.e. you have to implement the optimization part on your own. For any functionality you need that is not related to the backpropagation implementation (e.g. importing the dataset, etc.) feel free to use any library or the given script.

Submission instructions

The following instructions are mandatory. If you are not following them, tutors can decide to not correct your exercise.

Submission architecture

You have to generate a **single ZIP file** respecting the following architecture:

```
tutorial5_<matriculation1>_<matriculation2>_<matriculation3>
|
+--- source
|       |
|       +------- file 1
|       +------- file 2
|       +------- ...
+--- rapport.pdf
+--- README.txt
```

where

- **source** contains the source code of your project,
- **rapport.pdf** is the report where you present your solution with **the explanations** and the plots,
- **README** which contains group member informations (name, matriculation numbers and emails) and a **clear** explanation about how to compile and run your source code

The ZIP filename has to be :

```
tutorial5_<matriculation1>_<matriculation2>_<matriculation3>.zip
```

Some hints

We advice you to follow the following guidelines in order to avoid problems :

- Avoid building complex systems. The exercises are simple enough.
- Do not include any executables in your submission, as this will cause the e-mail server to reject it.

Grading

Send your assignment to the tutor who is responsible of your group:

- Merlin Köhler s9mnkoeh@stud.uni-saarland.de
- Yelluru Gopal Goutam goutamyg@lsv.uni-saarland.de

- Ahmad Taie ataie@lsv.uni-saarland.de

If you are assigned to different tutorials send your assignment to the tutor to whom most of you are assigned to.
The email subject should start with `[PSR TUTORIAL 5]`

Universität des Saarlandes
FR 4.7 Sprach- und Signalverarbeitung

Prof. Dietrich Klakow
Pattern and Speech Recognition

Exercise Sheet 6

Regularization for Deep Learning

Deadline: 03.01.2017, 23:59

Exercise 6.1 *(2 points)*

For each of the following statements state if it's true or false and justify your answer

a) Regularization is any modification we make to a learning algorithm that is intended to reduce its generalization error as well as its training error.

b) The model parameters α can be learned using linear regression for the model: $y = \log(x^{\alpha_1} e^{\alpha_2})$

Exercise 6.2 *(2 points)*

In what way is Dropout similar to Bagging and how do they defer? What about Boosting and Bagging? (Your answer should not exceed 4 sentences)

Exercise 6.3 *(5 points)*

Consider a quadratic error function of the form

$$E = E_0 + \frac{1}{2}(w - w^*)^T H (w - w^*)$$

where w^* represents the minimum, and the Hessian matrix H is positive definite and constant. Suppose the initial weight vector $w^{(0)}$ is chosen to be at the origin and is updated using simple gradient descent

$$w^\tau = w^{\tau-1} - \epsilon \nabla E$$

where τ denotes the step number, and ϵ is the learning rate (which is assumed to be small). After τ steps, the components of the weight vector parallel to the eigenvectors of H can be written

$$w_j^\tau = \{1 - (1 - \epsilon\lambda_j)^\tau\} w_j^*$$

where $w_j = w^T v_j$ and v_j and λ_j are the eigenvectors and eigenvalues, respectively, of H.

a) Show that as $\tau \to \infty$, this gives $w_\tau \to w^*$ as expected, provided $|1 - \epsilon\lambda_j| < 1$.

b) Now suppose that training is halted after a finite number of τ steps. Show that the components of the weight vector parallel to the eigenvectors of the Hessian satisfy

$$w_j^\tau \simeq w_j^* \text{ when } \lambda_j \gg (\epsilon\tau)^{-1}$$

$$|w_j^\tau| \ll |w_j^*| \text{ when } \lambda_j \ll (\epsilon\tau)^{-1}$$

c) Show that $(\epsilon\tau)^{-1}$ is analogous to the regularization parameter α from slide 6 in L_2 Parameter Regularization.

Exercise 6.4 *(11 points)*

a) Download the dataset from the course website. Plot the data, where the first and second columns are the 2d features (u, v), and the third column is the class. Give each class a different representation. *(1 points)*

b) We want to fit a regularized logistic regression model to the data. Recall the hypothesis function is

$$h_\theta(x) = \frac{1}{1 + e^{-\theta^T x}}$$

Start by coding a function that builds the feature vector x from all monomials of u and v up to the sixth power. *(1 points)*

c) The cost function for the logistic regression with L_2 regularization is

$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}[y^{(i)}\log(h_\theta(x^{(i)})) + (1 - y^{(i)})\log(1 - h_\theta(x^{(i)}))] + \frac{\lambda}{2m}\sum_{j=1}^{n}\theta_j^2$$

Where λ is the regularization parameter and y is the class. Derive the gradient and the hessian for this cost function. (You don't have to write the full vector/matrix) *(2 points)*

d) Implement Newton's method using the equations you derived in the previous question. *(4 points)*

e) Plot the decision boundary (Evaluate $\theta^T x$ over a grid of points (u, v) and plot $\theta^T x = 0$) for λ with values 0, 1, and 10. Comment on the plots. *(3 points)*

Christmas *Bonus*

Exercise 6.5 *(1 points)*

Santa wants to use a Single-layer perceptron to decide who has been naughty and who has been nice, Which of the following can he do with it?

a) A person is nice if they helped an old person or gave food to a beggar.

b) A person is naughty only if they pick their nose and they don't finish their food.

c) A person is nice if they like Santa or the Grinch, but not both.

Submission instructions

The following instructions are mandatory. If you are not following them, tutors can decide to not correct your exercise.

Submission architecture

You have to generate a **single ZIP file** respecting the following architecture:

```
tutorial2_<matriculation_nb1>_<matriculation_nb2>_<matriculation_nb3>
|
+--- source
|       |
|       +------- file 1
|       +------- file 2
|       +------- ...
+--- report.pdf
+--- README.txt
```

where

- **source** contains the source code of your project,
- **report.pdf** is the report where you present your solution with **the explanations** and the plots.
- **README** which contains group member informations (name, matriculation numbers and emails) and a **clear** explanation about how to compile and run your source code

The ZIP filename has to be :

`tutorial2_<matriculation_nb1>_<matriculation_nb2>_<matriculation_nb3>.zip`

Some hints

We advice you to follow the following guidelines in order to avoid problems :

- Avoid building complex systems. The exercises are simple enough.
- Do not include any executables in your submission, as this will cause the e-mail server to reject it.

Grading

Send your assignment to the tutor who is responsible of your group:

- Merlin Köhler s9mnkoeh@stud.uni-saarland.de
- Yelluru Gopal Goutam goutamyg@lsv.uni-saarland.de
- Ahmad Taie ataie@lsv.uni-saarland.de

The email subject should start with `[PSR TUTORIAL 6]`

Universität des Saarlandes
FR 4.7 Sprach- und Signalverarbeitung

Prof. Dietrich Klakow
Pattern and Speech Recognition

Exercise Sheet 7

Optimization for Training Deep Models

Deadline: 10.01.2017, 23:59

First Order Methods on MNIST

Exercise 7.1

In this exercise, you will use different first order optimization methods on MNIST dataset.

a) Download the code *multilayer_perceptron.py* from course website.

b) Run the code using following optimization schemes for 5 epochs: (i) gradient descent, (ii) gradient descent with momentum with momentum parameter = 0.5, (iii) AdaGrad with initial accumulator value = 0.1, (iv) RMSProp with decay = 0.9, mometum = 0. Submit the accuracy on test data for the four cases. *(4 pts)*

c) Why does gradient descent perform better than AdaGrad if the magnitude of gradients is large? Explain using the update rule equations for both the methods. *(1 pt)*

Visualization of Optimization Algorithms

Exercise 7.2

This task will help you visualize the behavior of different optmization techniques on the three dimensional error surface $f(x, y) = 3x^2 - y^2$

a) Implement gradient descent(gd) algorithm (or reuse the code from Assignment-2) with learning rate = 0.01 and $(x, y) = (5, -1)$ as the starting point. Run the code for 30 iterations and store the intermediate points. *(2 pts)*

b) Implement gd with momentum with $\alpha = 0.7$ while keeping other parameters same as above. Run the code for 30 iteratons and store the intermediate points. *(2 pts)*

c) Plot the 3D/contour plot of the function with paths of gd and gd with momentum using different colors. *(2 pts)*
For plotting in matlab, functions like *scatter3* can be used for 3D plot and *contour* for contour plot. In python, there are is a list of functions available, refer the link: http://matplotlib.org/mpl_toolkits/mplot3d/tutorial.html

d) Evaluate the function value at the termination points (after 30 iterations) for both the methods and produce the values. Which algorithm has made better progress in minimizing $f(x, y)$? *(1 pt)*

e) Implement Newton's method (or reuse the code from Assignment-2), run it for 5 iterations with same starting point and report your observations. Verify your observations by analytically computing Newton steps. This should explain why plain vanilla Newton's method is not suited for training deep neural networks. *(3 pts)*

Exercise 7.3

To understand the effectiveness of AdaGrad, consider $f(x, y) = 0.001x^2 - 0.001y^2$ which is a very flat function.

a) With learning rate = 0.1, and $(x, y) = (3, -1)$ as the starting point, run the *gd* algorithm for 300 iterations. *(1 pt)*

b) Implement the AdaGrad algorithm as described in the slide 21 of chapter-7 and run it with the same settings as above. Let $\delta = 10^{-9}$. *(3 pts)*

c) Plot the 3D/contour plot of the function with paths of *gd* and AdaGrad using different colors. *(1 pts)*

Submission instructions

The following instructions are mandatory. If you are not following them, tutors can decide to not correct your exercise.

Submission architecture

You have to generate a **single ZIP file** respecting the following architecture:

```
tutorial2_<matriculation_nb1>_<matriculation_nb2>_<matriculation_nb3>
|
+--- source
|      |
|      +------- file 1
|      +------- file 2
|      +------- ...
+--- report.pdf
+--- README.txt
```

where

- **source** contains the source code of your project,
- **report.pdf** is the report where you present your solution with **the explanations** and the plots.

- **README** which contains group member informations (name, matriculation numbers and emails) and a **clear** explanation about how to compile and run your source code

The ZIP filename has to be :

```
tutorial7_<matriculation_nb1>_<matriculation_nb2>_<matriculation_nb3>.zip
```

Some hints

We advice you to follow the following guidelines in order to avoid problems :

- Avoid building complex systems. The exercises are simple enough.
- Do not include any executables in your submission, as this will cause the e-mail server to reject it.

Grading

Send your assignment to the tutor who is responsible of your group:

- Merlin Köhler s9mnkoeh@stud.uni-saarland.de
- Goutam Y G goutamyg@lsv.uni-saarland.de
- Ahmad Taie s8ahtaie@stud.uni-saarland.de

The email subject should start with `[PSR TUTORIAL 7]`

Exercise Sheet 8

Convolutional Neural Networks

Deadline: 17.01.2017, 23:56

Convolution

Exercise 8.1 *(1+1+2+1+2=7 points)*

Consider the following 1-D signal

$$f = \boxed{3}\boxed{1}\boxed{8}\boxed{6}\boxed{3}\boxed{9}\boxed{5}\boxed{1}.$$

Furthermore consider the kernel

$$k = \boxed{\tfrac{1}{2}}\boxed{\tfrac{1}{2}}.$$

a) Compute the discrete convolution of the signal f with kernel k, i.e $s_1 = f * k$.

b) Compute the discrete convolution of s_1 and the kernel k, i.e. $s_2 = s_1 * k$.

c) Provide a plot of f, s_1 and s_2. Briefly describe the effects of convolving f with the kernel k. What would be the outcome of convolving f with k for n times when $n \to \infty$?

d) Compute a new kernel k' as $k' = k * k$. Now compute $s_3 = f * k'$.

e) Compare s_2 and s_3. What do you notice? Which mathematical concept seems to be fulfilled here by convolution? Prove that this concept is fulfilled by convolution for arbitrary signals.

Convolution and Convolutional Neural Networks

In this section we want to get a feeling for the similarities between a standard 2-D convolution as it is used e.g. for image processing and its corresponding implementation as a Convolutional Neural Network.

Exercise 8.2 *(4+1+1+1=7 points)*

The goal of this exercise is to implement a standard 2-D convolution as defined in lecture 8, slide 5. Therefore download the password protected `zip`-archive `ex08.zip` from the course website. The password is `psr_ws_1617`. The archive contains some images used later. To import images as a `numpy`-matrix in `python` you can use the `misc` module from `scipy` (`from scipy import misc`). With `im = misc.imread(<path>)` you obtain the matrix `im` as a matrix containing the color values of the image `<path>`. You can display a gray value image `im` using `matplotlib` (`import matplotlib.pyplot as plt`) with the function call `plt.imshow(im, cmap=plt.cm.gray)`.

a) Supplement the python script `convolution.py` which performs a 2-D convolution with a given 2-D signal $f(x, y)$ (e.g. a grey value image) and a kernel $k(x, y)$. You can assume that the kernels will have a shape like

$$k(x,y) = \begin{array}{|c|c|c|c|c|}\hline \dots & \dots & \dots & \dots & \dots \\ \hline \dots & (-1,-1) & (0,-1) & (1,-1) & \dots \\ \hline \dots & (-1,0) & (0,0) & (1,0) & \dots \\ \hline \dots & (-1,1) & (0,1) & (1,1) & \dots \\ \hline \dots & \dots & \dots & \dots & \dots \\ \hline \end{array},$$

i.e. they their center weight corresponds to the index $(0, 0)$.

As indicated on slide 6 the resulting image will have smaller dimensions than the original one, depending on the kernel size. To overcome this issue assume that the original image is padded with as many zeros as needed to preserve the original dimensions.

As the result of the convolution may not be in the range $[0; 255]$ anymore apply the function `min-max-rescale()` to the result, before displaying or saving it.

b) Apply the kernel

$$k_1(x,y) = \frac{1}{9} \cdot \begin{array}{|c|c|c|}\hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array}$$

to the image `clock_noise.png`. Describe the effect of the kernel and an application where such a kernel can be used for.

c) Apply the kernel

$$k_2(x,y) = \begin{array}{|c|c|c|}\hline 0 & 0 & 0 \\ \hline 1 & -2 & 1 \\ \hline 0 & 0 & 0 \\ \hline \end{array}$$

to the image `clock.png`. As before, describe the effect of this kernel and possible applications. Which meaning do different gray values have?

d) In part a) we padded the image with zeros in order to obtain an image with the same size as the original one. Why is this not always optimal? Give an example where this leads to unwanted results.

Hint: Think of the kernel k_2 from c). Which mathematical concept does this kernel implement. Which effect does the zero padding have for border pixels? What would be a more suitable padding in this case?

Exercise 8.3 *(3 points)*

The goal of this exercise is to build a Convolutional Neural Network using `Tensorflow`.

As our dataset we again want to use the `MNIST` dataset. The images have a size of 28×28. As they come as a vector in the training data you have to reshape them first to a 28×28 matrix.

The weights you are using should be initialized randomly. The biases should be initialized with a value of 0.1.

Now build your network as follows:

- The first hidden layer applies convolution (`tf.nn.conv2d`) with a 5×5 mask to the input nodes. For each of the patches you should produce 32 outputs. In order to retain the size of the layer perform zero padding. As an activation function for the output use the `RELU` function.

  In order to reduce the number of resulting nodes perform pooling by averaging with a 2×2 mask afterwards.

- The second hidden layer should be a fully connected layer, which takes the outputs of the previous layer as inputs. As an activation function for the output again use the `RELU` function. Try different sizes for this layer and comment on the effect to the accuracy.

- The output layer is again a fully connected layer and should yield a vector $\mathbf{x}$ of length 10. Each entry x_i should contain the probability of the given sample belonging to the number $i \in \{0, ..., 9\}$.

As a cost function use the cross entropy (`softmax_cross_entropy_with_logits(...)`) which you should normalize (`reduce_mean(...)`). As an optimization algorithm you can use the `AdamOptimizer` with a learning rate of 10^{-4}.

Train the model on the training data (You do not have to consider the full training set if it takes too long). Provide a plot of the evolving prediction accuracy.

Then evaluate the model on the test set. Report the accuracy the model achieved.

Gabor Filters

Exercise 8.4 *(3 points)*

In this exercise you should make yourself familiar with the Gabor filter.

The image `hide.png` shows a person which is painted such that it is hard to recognize in front of the background. Your goal is to indicate the presence of the person using Gabor filters.

Therefore proceed as follows:

- Implement a python script `gabor.py` which computes the convolution of an image and a Gabor filter. You are allowed to use existing implementations for that. I.e. you can modify this http://scikit-image.org/docs/dev/auto_examples/plot_gabor.html script for your purposes.

- In order to outline the person you might have to compute several filtered versions of the image and average them.

- Report the different parameters you used and provide a plot of the corresponding filters. Also briefly state why you have chosen these parameters.

- Make sure to also include the result in your report.

Submission instructions

The following instructions are mandatory. If you are not following them, tutors can decide to not correct your exercise.

Submission architecture

You have to generate a **single ZIP file** respecting the following architecture:

```
tutorial8_<matriculation1>_<matriculation2>_<matriculation3>
|
+--- source
|      |
|      +------- file 1
|      +------- file 2
|      +------- ...
+--- rapport.pdf
+--- README.txt
```

where

- **source** contains the source code of your project,
- **rapport.pdf** is the report where you present your solution with **the explanations** and the plots,
- **README** which contains group member informations (name, matriculation numbers and emails) and a **clear** explanation about how to compile and run your source code

The ZIP filename has to be :

```
tutorial8_<matriculation1>_<matriculation2>_<matriculation3>.zip
```

Some hints

We advice you to follow the following guidelines in order to avoid problems :

- Avoid building complex systems. The exercises are simple enough.
- Do not include any executables in your submission, as this will cause the e-mail server to reject it.

Grading

Send your assignment to the tutor who is responsible of your group:

- Merlin Köhler s9mnkoeh@stud.uni-saarland.de
- Yelluru Gopal Goutam goutamyg@lsv.uni-saarland.de
- Ahmad Taie ataie@lsv.uni-saarland.de

If you are assigned to different tutorials send your assignment to the tutor to whom most of you are assigned to.
The email subject should start with `[PSR TUTORIAL 8]`

Universität des Saarlandes
FR 4.7 Sprach- und Signalverarbeitung

Prof. Dietrich Klakow
Pattern and Speech Recognition

Exercise Sheet 9

Convolutional Neural Networks

Deadline: 24.01.2017, 23:59

Gabor filters and Convolution

Exercise 9.1 *(2 points)*

Consider the signal

$$g(x) = e^{-(x-1)^2} \cos 3x$$

And the filter

$$e^{-x^2} \cos kx$$

Calculate the continuous convolution of the signal with the Gabor filter analytically. For which parameters of the Gabor filter k and at which displacement do you get the maximum output?

Exercise 9.2 *(2 points)*

The fourier transform of a signal $f(x)$ is given as

$$F(v) = \int_{-\infty}^{\infty} f(x) e^{-2\pi i x v} dx$$

For two signals f and g. Show that the Fourier transform of the convolution is equal to the product of the Fourier transforms. This is also known as the Convolution theorem. How does this help us in Convolution?

Exercise 9.3 *(2 points)*

Use the convolution theorem from **Exercise 9.2** and apply it to **Exercise 9.1** by transforming both signal and filter. What do you observe in frequency space when multiplying signal and filter?

Max and Average Pooling

Exercise 9.4 *(2 + 1 + 1 + 2 = 6 points)*

In this excercise you will implement a simple max pooling function. And see how it can detect a feature (brightest pixel) in a certain region. We will be using the same image **clock.png** from the previous assignment. Follow the same procedure from **Excercise 8.2** to load the image.

a) Implement a function **maxPool** that takes two parameters: **data** and **split**.
Your function should take a square 2^nx2^n matrix as **data** and return a square matrix of size **split**x**split** where each entry is the result of a max pool in that patch.
For e.g. for our image of size 256x256, and a **split** of value 2 , we have an output 2x2 matrix, where each entry will correspond to the result of a max pool on each quadrant of the image, with no overlaps.

b) Apply your function on the image with **split** as 8,4,2, then 1.
What is the maximum value? Where can you find it in each of the splits? Provide the position(s) of that maximum value in each output matrix >1x1 as a tuple (i,j).

c) Change your function to an average pooling function. Name it **meanPool** and use it on **clock.png** with **split** as 128. Save the output image as **clockMean.png** and submit it with your code. Do the same thing with **maxPool** and **split** as 128. Save the image as **clockMax.png**. What do you observe?

d) To do Backpropagation in the pooling layer we need to get the derivative of the pooling function. Derive the derivatives for the max pooling and mean pooling w.r.t. the input. Explain how the error would be backpropagated for each case, also known as the Upsampling operation.

Training a Convolutional neural network

Exercise 9.5 *(2 points)*

Now that you are familiar with the parts that build up a Convolutional Neural network, derive the backpropagation equations for the weights of the Convolutional layer. Provide a pseudocode for the implementation.

In practice, CNN are rarely trained from scratch. Explain how Transfer Learning can help us train CNNs faster.

Revision Questions

Exercise 9.6 *(2 + 2 + 2 = 6 points)*

a) Explain how you would go about implementing a ConvNet to be used in the CIFAR-10 task. Show the layers you would use and ellaborate on the function of each layer, its connections, and dimensions. You answer should not exceed 10-12 sentences.

b) Present the Backpropagation algorithm for Multi-Layer-Perceptrons and explain its steps. Assume you have a squared error function as the cost function and the logistic function as the activation function, derive the equation(s) for the derivative of the error w.r.t the weights in terms of the output of the next layer and the target output.

c) Given a fully connected network with 3 neurons in the input layer, 5 neurons in a hidden layer and 2 neurons in the output layer. If the activation function is linear, show that this network can be simplified into a one-layer network. Draw both networks.

The following instructions are mandatory. If you are not following them, tutors can decide to not correct your exercise.

Submission architecture

You have to generate a **single ZIP file** respecting the following architecture:

```
tutorial2_<matriculation_nb1>_<matriculation_nb2>_<matriculation_nb3>
|
+--- source
|       |
|       +------- file 1
|       +------- file 2
|       +------- ...
+--- report.pdf
+--- README.txt
```

where

- **source** contains the source code of your project,
- **report.pdf** is the report where you present your solution with **the explanations** and the plots.
- **README** which contains group member informations (name, matriculation numbers and emails) and a **clear** explanation about how to compile and run your source code

The ZIP filename has to be :

`tutorial2_<matriculation_nb1>_<matriculation_nb2>_<matriculation_nb3>.zip`

Some hints

We advice you to follow the following guidelines in order to avoid problems :

- Avoid building complex systems. The exercises are simple enough.
- Do not include any executables in your submission, as this will cause the e-mail server to reject it.

Grading

Send your assignment to the tutor who is responsible of your group:

- Merlin Köhler s9mnkoeh@stud.uni-saarland.de
- Yelluru Gopal Goutam goutamyg@lsv.uni-saarland.de
- Ahmad Taie ataie@lsv.uni-saarland.de

The email subject should start with `[PSR TUTORIAL 9]`

Universität des Saarlandes
FR 4.7 Sprach- und Signalverarbeitung

Prof. Dietrich Klakow
Pattern and Speech Recognition

Exercise Sheet 10

Sequence Modelling: Recurrent Neural Networks

Deadline: 31.01.2017, 23:59

Power Method

Exercise 10.1

In this exercise, you will implement Power Method to find the eigenvector of a matrix corresponding to the largest eigenvalue by magnitude. It helps to understand the concept of exploding/vanishing gradients while training RNNs. (Chapter 9, slide 31)

a) Consider the matrix,

$$M = \begin{bmatrix} -2 & -2 & 3 \\ -10 & -1 & 6 \\ 10 & -2 & -9 \end{bmatrix}$$

Find its eigenvalues and the eigenvector v corresponding to the largest eigenvalue by magnitude analytically. ($\|v\|_2$=1) *(2 pts)*

b) Start with a random vector u to initiate the computations of Power Method, repeat until $|\langle u, v\rangle| \approx 1$. Resize the vector u after each iteration s.t. $\|u\|_2$=1. *(2 pts)*

c) Store the value of $|\langle u, v\rangle|$ for each iteration and plot its value against number of iterations. *(1 pt)*

Sequence Prediction using RNN

Exercise 10.2

In this task, you will train a RNN with a sequence of nucleotide bases of a given DNA snippet. The sequence contains 4 elements/nucleotide bases: adenine(A), cytonsine(C), guanine(G), thymine(T).

a) Download the file *data.txt* from the course webpage. Write a function to read the file and store the whole sequence as a string. *(2 pts)*

b) Write a function to extract a random chunk of length k from the sequence. These random chunks are used to train the RNN. *(2 pts)*

c) Write a function to encode each element of a given chunk as a *one-hot vector*. Add a vector of zeros in the beginning, which acts as *Start of Sequence(SoS)* symbol. For a given chunk of length k, the function should return a numpy array of shape $(k+1, 4)$ *(2 pts)*

d) Implement a simple RNN with network architecture defined in slide 11 of chapter 9 using Tensorflow. Let $h \in R^4$. Use softmax cross-entropy to compute the loss and $ADAM$ optimizer for backpropagation with learning rate $= 10^{-4}$.*(5 pts)*

e) To train the network, extract a random chunk x of length $k = 5$ from the sequence, encode it and feed into the network. If $x^{(i)}$ is the i^{th} element of the chunk, use the element at $x^{(i+1)}$ as $y^{(i)}$. (since the task is sequence prediction) *(3 pts)*

f) Repeat it for 50000 iterations, compute the average loss for every 100 iterations and plot the result. *(1 pts)*

Bonus

Exercise 10.3

In this task, you will measure the perdiction performance of the RNN model using *perplexity*. For details about perplexity, refer: `https://en.wikipedia.org/wiki/Perplexity`

a) Compute *perplexity* $= 2^{-\frac{1}{N}\sum_{i=1}^{N} log_2 P(x^{(i)}|x^{(i-1)},h^{(i-1)})}$ for the whole sequence of training data, where $P(x^{(i)}|x^{(i-1)},h^{(i-1)})$ is the output of the softmax layer for the element $x^{(i)}$. *(4 pts)*

b) To interpret the value of perplexity, consider the following cases: If the model predcits each element in the sequence correctly with $p(x^{(i)}|x^{(i-1)}, h^{(i-1)}) = 1$, then value of perplexity is 1. Other extreme case is when the predictions by the model are random, all outcomes are equiprobable and value of perplexity will be 4.

Based on the perplexity value you have obtained, what can you say about the performance of the model? *(1 pt)*

Submission instructions

The following instructions are mandatory. If you are not following them, tutors can decide to not correct your exercise.

Submission architecture

You have to generate a **single ZIP file** respecting the following architecture:

```
tutorial10_<matriculation_nb1>_<matriculation_nb2>_<matriculation_nb3>
|
+--- source
|      |
|      +------- file 1
|      +------- file 2
|      +------- ...
+--- report.pdf
+--- README.txt
```

where

- **source** contains the source code of your project,
- **report.pdf** is the report where you present your solution with **the explanations** and the plots.
- **README** which contains group member informations (name, matriculation numbers and emails) and a **clear** explanation about how to compile and run your source code

The ZIP filename has to be :

`tutorial10_<matriculation_nb1>_<matriculation_nb2>_<matriculation_nb3>.zip`

Some hints

We advice you to follow the following guidelines in order to avoid problems :

- Avoid building complex systems. The exercises are simple enough.
- Do not include any executables in your submission, as this will cause the e-mail server to reject it.

Grading

Send your assignment to the tutor who is responsible of your group:

- Merlin Köhler s9mnkoeh@stud.uni-saarland.de
- Goutam Y G goutamyg@lsv.uni-saarland.de
- Ahmad Taie s8ahtaie@stud.uni-saarland.de

The email subject should start with `[PSR TUTORIAL 10]`

Universität des Saarlandes
FR 4.7 Sprach- und Signalverarbeitung

Prof. Dietrich Klakow
Pattern and Speech Recognition

Exercise Sheet 11

Hidden Markov Models

Deadline: 06.02.2017, 10:00

Exercise 11.1 *(5+5+5=15 points)*

Consider a HMM with 3 states (K = 3) and 2 output symbols 1,2, with the transition matrix:

$$A = \begin{pmatrix} 0.5 & 0 & 0 \\ 0.3 & 0.6 & 0 \\ 0.2 & 0.4 & 1 \end{pmatrix}$$

where $A_{i,j} = P(\pi_{t+1} = i|\pi_t = j)$. The emission matrix is (first row corresponds to 1, second row to 2):

$$B = \begin{pmatrix} 0.7 & 0.4 & 0.8 \\ 0.3 & 0.6 & 0.2 \end{pmatrix}$$

where $B_{i,j} = P(x_t = i|\pi_t = j)$. The initial probabilities are 0.9, 0.1, 0 for the three states respectively. Given that the observed sequence is $x_1 = 1, x_2 = 2, x_3 = 1$:

a) Compute $P(x_1, x_2, x_3)$.

b) Compute $P(\pi_1|x_1, x_2, x_3)$ for all states.

c) Find the most likely hidden state sequence.

Perform all the computations by hand, i.e. there is no implementation required here.

Exercise 11.2 *(3+2=5 points)*

Consider an HMM representation of a coin tossing experiment. Assume a 3-state model (corresponding to 3 different coins) with the same structure as in exercise 1. The emission matrix is given by

$$B = \begin{pmatrix} 0.5 & 0.75 & 0.25 \\ 0.5 & 0.25 & 0.75 \end{pmatrix}$$

where all the first row corresponds to `H` and the second row to `T`. All state transition have probabilities equal to $\frac{1}{3}$. Assume initial state probabilities of $\frac{1}{3}$.

a) You observe the sequence: `O=H H H H T H T T T T`. What is the most likely state sequence and its probability (likelihood)?

b) What is the probability that the observation sequence came entirely from state 1?

Submission instructions

The following instructions are mandatory. If you are not following them, tutors can decide to not correct your exercise.

Submission architecture

You have to generate a **single ZIP file** respecting the following architecture:

```
tutorial11_<matriculation1>_<matriculation2>_<matriculation3>
|
+--- source
|      |
|      +------- file 1
|      +------- file 2
|      +------- ...
+--- rapport.pdf
+--- README.txt
```

where

- **source** contains the source code of your project,
- **rapport.pdf** is the report where you present your solution with **the explanations** and the plots,
- **README** which contains group member informations (name, matriculation numbers and emails) and a **clear** explanation about how to compile and run your source code

The ZIP filename has to be :

`tutorial11_<matriculation1>_<matriculation2>_<matriculation3>.zip`

Some hints

We advice you to follow the following guidelines in order to avoid problems :

- Avoid building complex systems. The exercises are simple enough.
- Do not include any executables in your submission, as this will cause the e-mail server to reject it.

Grading

Send your assignment to the tutor who is responsible of your group:

- Merlin Köhler s9mnkoeh@stud.uni-saarland.de
- Yelluru Gopal Goutam goutamyg@lsv.uni-saarland.de
- Ahmad Taie ataie@lsv.uni-saarland.de

If you are assigned to different tutorials send your assignment to the tutor to whom most of you are assigned to.
The email subject should start with `[PSR TUTORIAL 11]`

Name:
Matriculation number:
E-Mail:

Lecture "Pattern and Speech Recognition"

Prof. Dr. D. Klakow

Exam

Tuesday, February 16th, 2016
14:00-16:00
Günter-Hotz-Hörsaal, Geb. E2 2

All questions are weighted equally. You have to pick exaclty 13 questions. Please tick the 13 questions you want to be graded.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20

Note:

- Please read all of the questions. The questions are designed such that the answers should have limited overlap.
- Use a separate sheet for each question and mark the sheet with your name and matriculation number.
- Answer your questions in English (otherwise we will have problems in grading).
- Staple your answer in the order they are numbered. This question sheet goes as the cover sheet. Staplers will be provided.
- Answer only what you are asked for and try to be compact and precise.
- No books, notes, smartphones etc. are allowed during the exam.

Questions

Feature Extraction

1. Describe the chain used for the MFCC feature extraction and a brief verbal justification for each step in the chain.

2. Explain the process of windowing. Why would you use a Hamming window rather than a rectangular window? Justify.

Probability Theory

3. The joint distribution $P(X, Y)$ for two binary variables X and Y is given:

X \Y	0	1
0	$\frac{1}{3}$	$\frac{1}{3}$
1	0	$\frac{1}{3}$

   Compute the marginal probability distribution $P(X)$ and $P(Y)$ and the conditional probability distribution $P(X|Y)$.

4. Give the formula for the one dimensional normal distribution and derive the maximum likelihood estimate for the mean. If you don't remember the normalization just use a constant $1/Z$ and do the derivation using this constant.

Decision Theory

5. This table

Vote	Y	Y	N	Y	N	N	Y	N
Class label	D	D	D	D	R	R	R	R

   illustrates how some of the U. S. Representatives voted on the *Affordable Health Care for America Act.* Construct a loss function $\lambda(\alpha_i|\omega_j)$ (where α_i is the decision taken by the system and ω_i is the true class label) such that the conditional risk of predicting *republican* and *democrat* is equal, given that the vote was *yes.* Write down the definition of conditional risk and show that your loss function fulfills this property.

6. Give the formula for the Bayesian decision rule and explain the elements. Derive the Naive Bayes Classifier for a sequence of features $x_1 \cdots x_N$. What are the additional assumptions.

GMM

7. Prove that k–means clustering does not (necessarily) find a global optimum. Hint: Find a counterexample (e.g in one dimension with four training samples and two clusters) where k-means terminates at a local optimum. Recall that it tries to minimize the root-mean-square error (RMSE).

8. Explain the meaning of the covariance matrix Σ and compute Σ for the 2–dimensional data given in table below:

x_1	x_2
1	5
3	8
5	11

9. With the output of K-means algorithm explain how you would initialize a Gaussian Mixture Model.

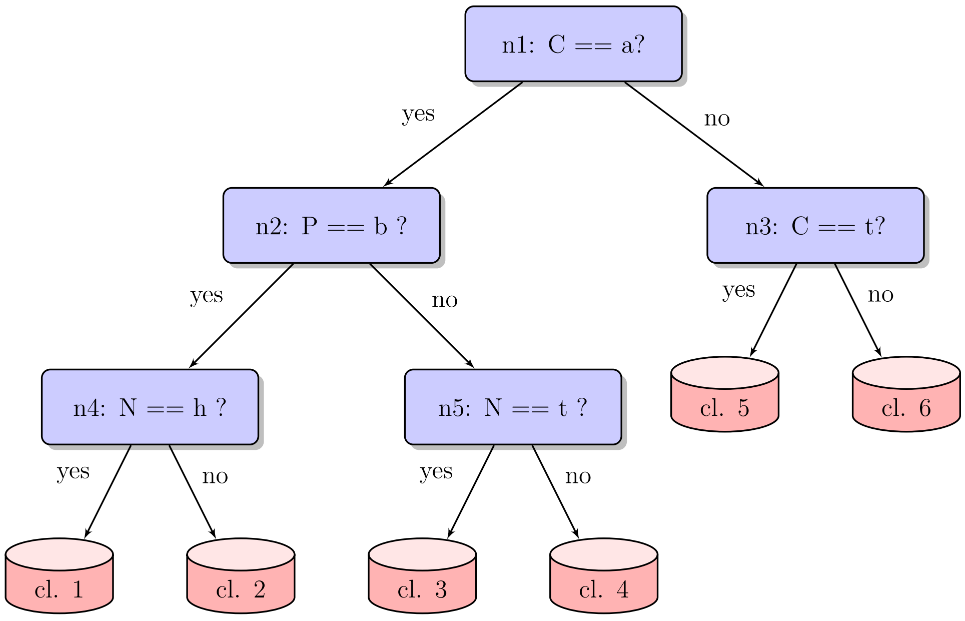
Speaker Recognition

10. Write down the decision rule for speaker recognition. Explain how you model a speaker and how this model can be trained.

Decision Trees

11. Indicate and justify all the input you need to **train** a decision tree. Which types of decision boundaries can be achieved for which type of questions.

12. Considering the following decision tree and the following set of data. For each element of the dataset, classify it using the given tree. Evaluate the final result with respect to the misclassification impurity for each node and compare it to the initial impurity of the root node.



The four data samples i1, i2, i3 and i4 are given by:

	P	C	N	Class
i1	c	a	c	1
i2	c	a	b	1
i3	j	a	d	2
i4	a	c	b	2

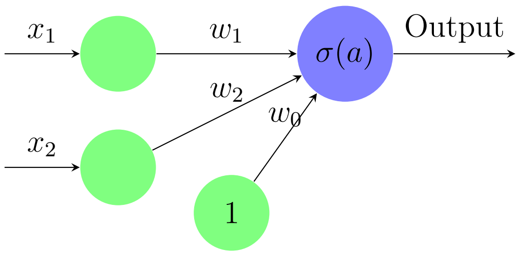
Neural Networks

13. Assume you want to classify the following four points which are real valued vectors in a two dimensional space (x_1, x_2):

x_1	0	0	1	1
x_2	0	1	0	1
Class	C_0	C_0	C_0	C_1

What is a simple neural network that can classify the above points? Fully specify all necessary parameters of a neural network that can classify these points.

14.



Imagine the following set-up. Initial parameters: $w_1 = 1$, $w_2 = -1$, $w_0 = 0.5$ For the input $x = (0, 1)$ the true label is $t = 1$.

$y(x, w) = \sigma(\sum_{i=1}^{D} w_i x_i + w_0)) \approx 0.38$

The activation function is the logistic sigmoid: $\sigma(x) = \frac{1}{1+e^{-x}}$.
The derivative of the sigmoid is: $\sigma(x)(1 - \sigma(x))$
The error function for a single datapoint is given by: $E(w, x) = \frac{1}{2}(y(x, w) - t)^2$

Update the weight w_1 using the backpropagation algorithm with learning rate $\eta = 1$.

15. What are advantages and disadvantages of neural networks compared to decision trees? Limit yourself to one page maximum including suitable diagrams or drawings.

HMM

16. Compute the transition matrix and the emission matrix using maximum likelihood for the observation $x = 1, 1, 3, 1, 2, 2, 1, 3, 2, 1$ and hidden state sequence $\pi = F, F, L, F, F, F, L, L, F, F$

17. Considering the following HMM of three states $Q = \{1, 2, 3\}$ and the following observation sequence $abaa$, apply the forward algorithm.

- Initialisation probabilities

k	1	2	3
a_{0k}	0.2	0.2	0.6

- Transition probabilities a_{ij}

i \j	1	2	3
1	0	0.2	0.8
2	0.5	0.5	0
3	0	0	1

- Emission probabilities

	1	2	3
a	0	0.5	1
b	1	0.5	0

18. Explain the viterbi training algorithm. Indicate (briefly !) the advantages/drawbacks compared to the Baum-Welch training, which is based on the EM algorithm. Justify the use of pseudo-counts.

CRF

19. Describe the advantages of CRFs over HMMs. Also compare the underlying independence assumptions using diagrams and explaining them.

20. CRFs can be used for named entity recognition. Please give four types of suitable feature functions.