KTH Computer Science and Communication

Exam in DD2421 Machine Learning
2021-03-19, kl 10.00 – 2021-03-20, kl 10.00

Aids allowed: *calculator*, *language dictionary*. This is also an open book exam.

To take this exam you must be registered to this specific exam as well as to the course. Then you have an access to the Canvas page, "Tentamen för DD2421/FDD3431 TEN1: 2021-03-19", https://kth.instructure.com/courses/29597. Read the instructions given there. For your submission go in "Assignments" on the top-left of the page where you also find "Code of conduct" – you can click submission buttons during the exam hours. Email is not accepted as a tool for submissions.

In order to pass this exam, your score x first needs to be 16 or more (out of 42, full point). In addition, given your points y from the Programming Challenge (out of 18, full point), the requirements on the total points, $p = x + y$, are preliminarily set for different grades as:

$$
\begin{aligned}
53 < p \leq 60 &\quad \rightarrow \quad A \\
46 < p \leq 53 &\quad \rightarrow \quad B \\
39 < p \leq 46 &\quad \rightarrow \quad C \\
32 < p \leq 39 &\quad \rightarrow \quad D \\
24 < p \leq 32 &\quad \rightarrow \quad E \quad \text{(A pass is guaranteed with the required points for 'E'.)} \\
0 \leq p \leq 24 &\quad \rightarrow \quad F
\end{aligned}
$$

This exam consists of sections **A**, **B**, and **C**, to which different zoom links are assigned in case of inquiries. **NB. Use different papers (answer sheets) for different sections.**

A Graded problems

Potential inquiries to be addressed to zoom link (A).

A-1 Terminology (5p)

For each term (a–e) in the left list, find the explanation from the right list which *best* describes how the term is used in machine learning.

a) Curse of dimensionality

b) Fisher's criterion

c) Bagging

d) RANSAC

e) Occams's razor

1) Problems in high computational cost

2) The bag-of-words model

3) An example of ensemble learning

4) A concept of accepting high model complexity

5) Robust method to fit a model to data with outliers

6) A principle to choose the simplest explanation

7) An approach to find useful dimension for classification

8) Issues in data sparsity in space

9) An example of unsupervised learning

10) Random strategy for amplitude compensation

Solution: a-8, b-7, c-3, d-5, e-6,

A-2 Nearest Neighbor, Classification (4p)

Suppose that we take a data set, divide it into training and test sets, and then try out two different classification procedures. We use *three-quarters* of the data for training, and the remaining *one-quarter* for testing. First we use *Logistic Regression* and get an error rate of 10% on the training data. We also get the average error rate (weighted average over both test and training data sets) of 12%. Next we use 1-nearest neighbor and get an average error rate (weighted average over both test and training data sets) of 6%.

a) What was the error rate with 1-nearest neighbor on the training set? (1p)

b) What was the error rate with 1-nearest neighbor on the test set? (1p)

c) What was the error rate with Logistic Regression on the test set? (1p)

d) Based on these results, indicate the method which we should prefer to use for classification of new observations, with a simple reasoning. (1p)

Solution:

a) 0%. Training error for 1-NN is always zero.
b) 24%. Given the answer in **a**), the testing error is 24%.
c) 18%.
d) Logistic Regression, because it achieves lower error rate on the test data (18% < 24%).

A-3 Entropy and Decision Trees/Forests (5p)

a) Indicate a correct one as the basic strategy for selecting a question (attribute) at each node in decision trees.

i. To minimize the expected reduction of the entropy.
ii. To minimize the expected reduction of gini impurity.
iii. To maximize the expected reduction of the entropy.

Simply indicate your choice. (1p)

b) Draw a decision tree of *depth two* that is consistent with the training data in the table. $a_1, ..., a_4$ are attributes of the training data. (It is not required to use information gain to solve the task.) (2p)

a_1	a_2	a_3	a_4	$class$
0	0	0	0	-
0	0	1	1	+
0	1	0	1	+
0	1	1	0	-
1	0	0	0	+
1	0	1	1	+
1	1	1	0	-
1	1	0	1	-

c) Suppose we have trained a Decision Forest using five bootstrapped samples from a data set containing three classes, {Red, Yellow, Green}. We then applied the model to a specific test input, $\mathbf{x}$, and observed five estimates of class distributions from the five trees, respectively:

$\{P_1(\text{Red}|x), P_1(\text{Yellow}|x), P_1(\text{Green}|\mathbf{x})\}$ = {0.1, 0.5, 0.4}
$\{P_2(\text{Red}|x), P_2(\text{Yellow}|x), P_2(\text{Green}|\mathbf{x})\}$ = {0.2, 0.3, 0.5}
$\{P_3(\text{Red}|x), P_3(\text{Yellow}|x), P_3(\text{Green}|\mathbf{x})\}$ = {0.3, 0.6, 0.1}
$\{P_4(\text{Red}|x), P_4(\text{Yellow}|x), P_4(\text{Green}|\mathbf{x})\}$ = {0.2, 0.4, 0.4}
$\{P_4(\text{Red}|x), P_4(\text{Yellow}|x), P_4(\text{Green}|\mathbf{x})\}$ = {0.2, 0.3, 0.5}

Now, let us consider the average probability for combining these results into a single class prediction. What will the final class prediction be? *Motivate your answer by short phrases*. (2p)

Solution: **a**)-iii. **c**)-Yellow.

A-4 Regression with regularization (4p)

For a set of N training samples $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\}$, each consisting of input vector $\mathbf{x}$ and output y, suppose we estimate the regression coefficients $\mathbf{w}^\top (\in \mathbf{R}^d) = \{w_1, \ldots, w_d\}$ in a linear regression model by minimizing

$$\sum_{n=1}^{N} (y_n - \mathbf{w}^\top \mathbf{x}_n)^2 + \lambda \sum_{i=1}^{d} w_i^2$$

for a particular value of λ. Now, let us imagine we consider a model with an extremely large λ and then models with *gradually smaller* values of λ (towards 0).

For parts **a**) and **b**), indicate which of i. through v. is correct. Briefly justify your answer for each part.

a) The variance of the model will:
i. Remain constant.
ii. Steadily decrease.
iii. Steadily increase.
iv. Decrease initially, and then eventually start increasing in a U shape.
v. Increase initially, and then eventually start decreasing in an inverted U shape. (2p)

b) Repeat **a**) for test RSS. (2p)

Solution: **a**)-iii, **b**)-iv

A-5 PCA, Subspace Methods (4p)

Given a set of feature vectors which all belong to a specific class C (i.e. with an identical class label), we performed PCA on them and generated an orthonormal basis $\{\mathbf{u}_1, ..., \mathbf{u}_p\}$ which spans a p-dimensional subsapce, $\mathcal{L}$, as the outcome. Provide an answer to each question below.

a) What criterion are considered as useful for generating a basis in PCA?
i. Minimum variance criterion.
ii. Minimum squared distance criterion.
iii. Maximum variance criterion.
iv. Maximum squared distance criterion.
Simply choose what is valid. (2p)

b) Briefly explain what information can be referred for choosing an effective dimensionality of $\mathcal{L}$. (1p)

c) Now, we consider to solve a K-class classification problem with the Subspace Method and assume that a subspace $\mathcal{L}^{(j)}$ $(j = 1, ..., K)$ has been computed with training data for each class, respectively. Given a new input vector $\mathbf{x}$ whose class is unknown, we computed its projectiton length on each subspace as $S^{(j)}$ $(j = 1, ..., K)$, respectively. For a few classes among those with labels $\{l,m,n\}$, we had the following observations:
$S^{(l)}$ was the minimum of all $S^{(j)}$'s,
$S^{(m)}$ was the maximum of all $S^{(j)}$'s, and
$S^{(n)}$ was the closest to the average of all $S^{(j)}$'s.

Which class should $\mathbf{x}$ belong to? Simply choose a class label. (1p)

Solution:

a) ii, iii

b) The eigenvalues of the covariance (or autocorrelation) matrix

c) m

B Graded problems

Potential inquiries to be addressed to zoom link (B).

B-1 Warming up With Bayes' (3p)

Before I took a COVID test, the doctor said 99% of the people in the area have COVID, and 90% of them are testing positive. A few days later the doctor called and said my test was positive, and that the probability I have COVID given this positive test is p% — I can't remember because I was in shock. Find the minimum value of p such that I can compute the probability I got a positive test but don't have COVID, and then compute the maximum probability I don't have COVID given my positive test. Show your work.

Solution: Let C mean presence of COVID, $\neg C$ mean absence of COVID, and $+$ a positive test. From the wording of the problem we know $P[C] = 0.99$, $P[+|C] = 0.9$. The doctor claims $P[C|+] = p$. I want to find $P[+|\neg C]$. By Bayes' Theorem:

$$P[+|\neg C] = \frac{P[\neg C|+]P[+]}{P[\neg C]} = \frac{(1 - P[C|+])P[+]}{1 - P[C]} = \frac{(1-p)P[+]}{1 - P[C]} \tag{1}$$

since $P[\neg C|+] = 1 - P[C|+]$, and $P[\neg C] = 1 - P[C]$ as there are only two possibilities. We also know

$$P[+|C] = \frac{P[C|+]}{P[C]} = \frac{p}{P[C]} \tag{2}$$

and so solving for $P[+]$

$$P[+] = \frac{P[+|C]P[C]}{P[C|+]} = \frac{P[+|C]P[C]}{p} \tag{3}$$

Substituting that into the above produces

$$P[+|\neg C] = \frac{(1 - P[C|+])}{1 - P[C]} \frac{P[+|C]P[C]}{P[C|+]} = \frac{P[C]}{1 - P[C]} \frac{(1 - P[C|+])}{P[C|+]} P[+|C] \tag{4}$$

$$= \frac{P[C]}{1 - P[C]} \frac{(1-p)}{p} P[+|C]. \tag{5}$$

For the left hand side to be a probability, we need

$$\frac{P[C]}{1 - P[C]} \frac{(1-p)}{p} P[+|C] \leq 1 \tag{6}$$

$$\frac{(1-p)}{p} \leq \frac{1 - P[C]}{P[+|C]P[C]}. \tag{7}$$

Substituting our values from above and solving for p produces

$$p \geq \frac{891}{901} \approx 0.988. \tag{8}$$

Finally,

$$P[\neg C|+] = 1 - P[C|+] = 1 - p \leq 1 - \frac{891}{901} = \frac{10}{901} = 0.011. \tag{9}$$

B-2 Maximum likelihood estimation (3p)

Consider the data in Table 1. Assume all observations are independent. For each doggo, fit the number of its poopoos with a Poisson distribution using maximum likelihood estimation of the parameters, and compute those parameters. Show your work.

walk	poopoos	doggo
1	2	Shoogee Nulnul
2	3	Shoogee Nulnul
3	1	Shoogee Nulnul
4	2	Shoogee Nulnul
5	0	Shoogee Nulnul
1	2	Max the Tax
2	3	Max the Tax
3	4	Max the Tax
4	3	Max the Tax
5	4	Max the Tax

Table 1. The number of poopoos emitted by two doggos on five walks.

Solution: Define the random variable X mapping the number of poopoos to the natural numbers. Since X is distributed Poisson with parameter λ_k

$$P_X(X = x|\lambda_k) = \frac{\lambda_k^x e^{-\lambda_k}}{x!}\mu(x) \tag{10}$$

where $\mu(x) = 1, x \geq 0$ and zero otherwise. Since the number of poopoos by doggo k on N_k walks are independent, the log likelihood of the data D_k is

$$\log P(D_k|\lambda_k) = \sum_{n=1}^{N_k} \log P_X(X = x_n|\lambda_k) = \sum_{n=1}^{N_k} \log \frac{\lambda_k^{x_n} e^{-\lambda_k}}{x_n!} \tag{11}$$

$$= \sum_{n=1}^{N_k} (x_n \log \lambda_k - \lambda_k - \log x_n!) \tag{12}$$

We want to maximize this, so taking the derivate and setting to zero:

$$\frac{d}{d\lambda_k} \log P(D_k|\lambda_k) = \sum_{n=1}^{N_k} \frac{d}{d\lambda_k}(x_n \log \lambda_k - \lambda_k - \log x_n!) \Rightarrow -\lambda_k N_k + \sum_{n=1}^{N_k} x_n = 0 \tag{13}$$

And thus

$$\lambda_k = \frac{1}{N_k} \sum_{n=1}^{N_k} x_n. \tag{14}$$

For Shoogee Nulnul ($k = 0$) $\lambda_0 = 1.6$, and for Max the Tax ($k = 1$) $\lambda_1 = 3.2$.

B-3 Maximum a posteriori estimation (3p)

Consider the data in Table 1. Assume all observations are independent. For each doggo, fit its number of poopoos with a Poisson distribution using maximum a posteriori estimation of the parameters, and compute those parameters. Assume the prior distribution of the parameters $f_\Theta(\theta_k)$ is exponential with parameter $\gamma = 3$. Show your work.

Solution: Assuming the number of poopoos by doggo k on all walks are independent, the log posterior of the parameter λ_k is

$$\log f_\Lambda(\lambda_k|D_k) = \log \frac{P(D_k|\lambda_k)f_\Lambda(\lambda_k)}{P(D_k)} \tag{15}$$

$$= \log f_\Lambda(\lambda_k) + \log P(D_k|\lambda_k) - \log P(D_k) \tag{16}$$

$$= \log f_\Lambda(\lambda_k) + \sum_{n=1}^{N_k} \log P_X(X = x_n|\lambda_k) - \log P(D_k) \tag{17}$$

$$= \log \lambda_k e^{-\gamma\lambda_k} + \left[\sum_{n=1}^{N_k}(x_n \log \lambda_k - \lambda_k - \log x_n!)\right] - \log P(D_k) \tag{18}$$

$$= \log \lambda_k - \gamma\lambda_k + \left[-N_k\lambda_k + \sum_{n=1}^{N_k}(x_n \log \lambda_k - \log x_n!)\right] - \log P(D_k) \tag{19}$$

for $\lambda_k \geq 0$. We want to find the λ_k that maximizes this, so taking the derivate and setting to zero:

$$\frac{d}{d\lambda_k} \log f_\Lambda(\lambda_k|D_k) = \frac{1}{\lambda_k} - \gamma - N_k + \sum_{n=1}^{N_k} \frac{x_n}{\lambda_k} = 0 \tag{20}$$

$$\Rightarrow \lambda_k = \frac{1}{N_k + \gamma} \sum_{n=1}^{N} x_n. \tag{21}$$

And thus for Shoogee Nulnul $\lambda_0 = 1$, and for Max the Tax $\lambda_1 = 2$.

B-4 Bayes' predictive posterior (3p)

Consider the data in Table 1. Assume all observations are independent. For each doggo, find the Bayes' predictive posterior for the number of its poopoos. Model the number of poopoos of each dog with a Poisson distribution and assume the prior distribution of the parameters $f_\Theta(\theta_k)$ is exponential with parameter $\gamma = 3$. For convenience, assume $P(D_k) = 1$. Express the Bayes' predictive posterior in closed-form, i.e., not as an integral. Show your work.

Solution: We want to find

$$P_X(X = x|D_k) = \int_{\lambda_k=0}^{\infty} P_X(X = x|\lambda_k) f_\Lambda(\lambda_k|D_k) d\lambda_k. \tag{22}$$

The posterior distribution of the parameter is

$$f_\Lambda(\lambda_k|D_k) = \frac{1}{P(D_k)} \gamma e^{-\gamma\lambda_k} \prod_{n=1}^{N_k} \frac{\lambda_k^{x_n} e^{-\lambda_k}}{x_n!} = \gamma e^{-\gamma\lambda_k} \prod_{n=1}^{N_k} \frac{\lambda_k^{x_n} e^{-\lambda_k}}{x_n!} \tag{23}$$

The likelihood of a number of poopoos is

$$P_X(X = x|\lambda_k) = \frac{\lambda_k^x e^{-\lambda_k}}{x!} \mu(x) \tag{24}$$

The predictive posterior is thus

$$P_X(X = x|D_k) = \int_{\lambda_k=0}^{\infty} \frac{\lambda_k^x e^{-\lambda_k}}{x!} \gamma e^{-\gamma\lambda_k} \left[\prod_{n=1}^{N} \frac{\lambda_k^{x_n} e^{-\lambda_k}}{x_n!} \right] d\lambda_k \tag{25}$$

$$= \int_{\lambda_k=0}^{\infty} \frac{\lambda_k^x e^{-\lambda_k}}{x!} \gamma e^{-\gamma\lambda_k} e^{-N_k\lambda_k} \frac{\lambda_k^{\sum_n x_n}}{\prod_{n=1}^{N_k} x_n!} d\lambda_k \tag{26}$$

$$= \frac{1}{x! \prod_{n=1}^{N_k} x_n!} \int_{\lambda_k=0}^{\infty} \lambda_k^x e^{-\lambda_k} \gamma e^{-\gamma\lambda_k} e^{-N_k\lambda_k} \lambda_k^{\sum_n x_n} d\lambda_k \tag{27}$$

Define

$$g(x,k) = \frac{1}{x! \prod_{n=1}^{N_k} x_n!} \tag{28}$$

$$s_k = \sum x_n \tag{29}$$

$$\alpha_k = 1 + \gamma + N_k \tag{30}$$

Then

$$P_X(X = x|D_k) = g(x,k) \int_{\lambda_k=0}^{\infty} \lambda_k^x e^{-\lambda_k} \gamma e^{-\gamma\lambda_k} e^{-N_k\lambda_k} \lambda_k^{s_k} d\lambda_k \tag{31}$$

$$= g(x,k) \int_{\lambda_k=0}^{\infty} \lambda_k^{x+s_k} e^{-\lambda_k(1+\gamma+N_k)} d\lambda_k \tag{32}$$

$$= g(x,k) \int_{\lambda_k=0}^{\infty} \lambda_k^{x+s_k} e^{-\alpha_k\lambda_k} d\lambda_k \tag{33}$$

From a table of integrals we see

$$\int_0^{\infty} \lambda^{z-1} e^{-\lambda} d\lambda = \Gamma(z) \tag{34}$$

where $\Gamma(z)$ is the gamma function. So, define $\lambda'_k = \alpha_k\lambda_k$, and so $d\lambda'_k = \alpha_k d\lambda_k$, and $z - 1 = x + s_k$. Substituting this into the above

$$P_X(X = x|k, D_k) = g(x,k) \int_{\lambda'_k=0}^{\infty} \left(\frac{\lambda'_k}{\alpha_k} \right)^{x+s_k} e^{-\lambda'_k} \frac{1}{\alpha_k} d\lambda'_k \tag{35}$$

$$= \frac{g(x,k)}{\alpha_k^{x+s_k+1}} \int_{\lambda'_k=0}^{\infty} \lambda_k'^{z-1} e^{-\lambda'_k} d\lambda'_k \tag{36}$$

$$= \frac{g(x,k)}{\alpha_k^{x+s_k+1}} \Gamma(x + s_k + 1) \tag{37}$$

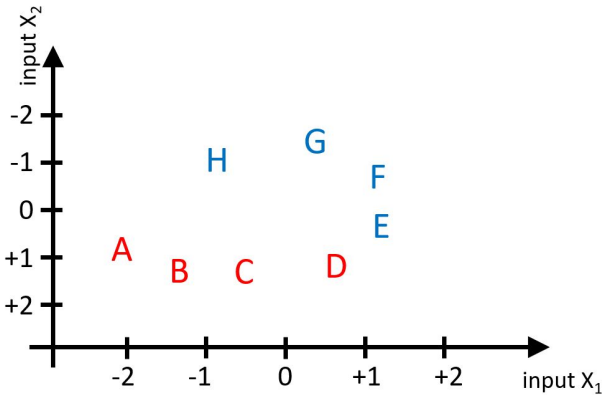
where we have defined $z - 1 = x + s_k \Rightarrow z = x + c_k + 1$.

C Graded problems

Potential inquiries to be addressed to zoom link (C).

C-1 Support Vector Classification (4p)

The following diagram shows a small data set consisting of four RED samples (A, B, C, D) and four BLUE samples (E, F, G, H). This data set can be linearly separated.

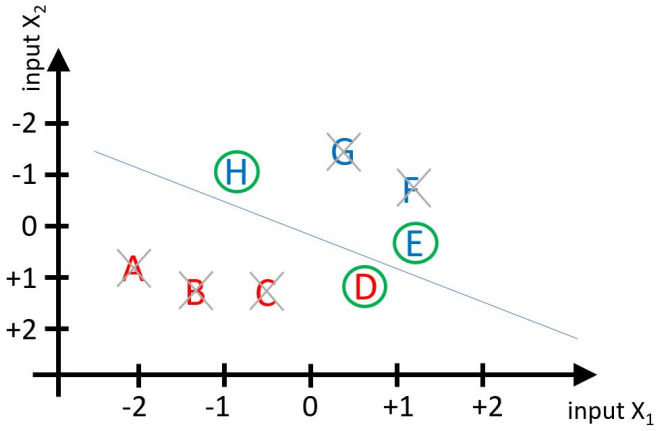


a) We use a linear support vector machine (SVM) without kernel function to correctly separate the BLUE and the RED class. Which of the data points (A-H) can be removed for training without changing the resulting SVM decision boundary? No explanation needed; name the point(s). (2p)

b) Assume someone suggests using a non-linear kernel for the SVM classification of the above data set (A-H). Give one argument in favor and one argument against using non-linear SVM classification for such a data set. **USE KEYWORDS!** (2p)

Solution:

a) The blue line shows the linear decision boundary between the two classes.



The data points in green circles are support vectors for the linear decision boundary. If any of those data points change, the decision boundary changes. In contrast, we can remove all other data points: **A, B, C, F, and G**.

b) at least one of each; max 1 point for (+) and 1 point for (-)

+) The decision boundary margin might get wider with a non-linear kernel.

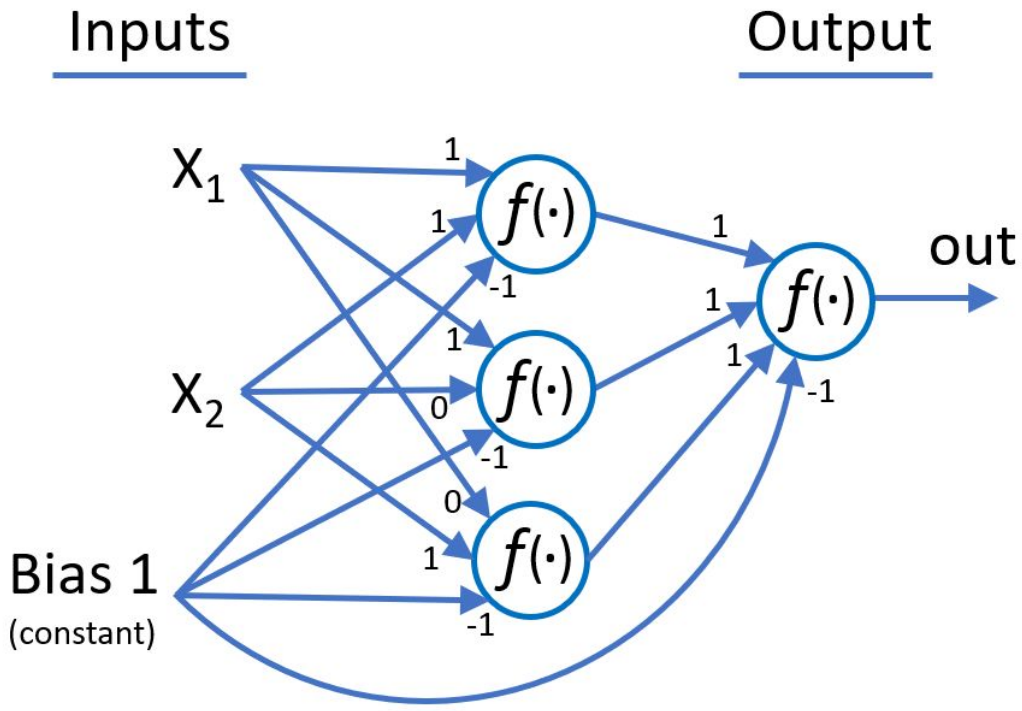
+) The same learning approach is likely to work for additional (possibly more complex) data.

-) More computing resources required.

-) The algorithm is more difficult to implement.

C-2 Neuronal Networks (4p)

The following diagram shows a simple neuronal network with step activation functions in all neurons.



$f(\cdot)$ = step neuronal activation function

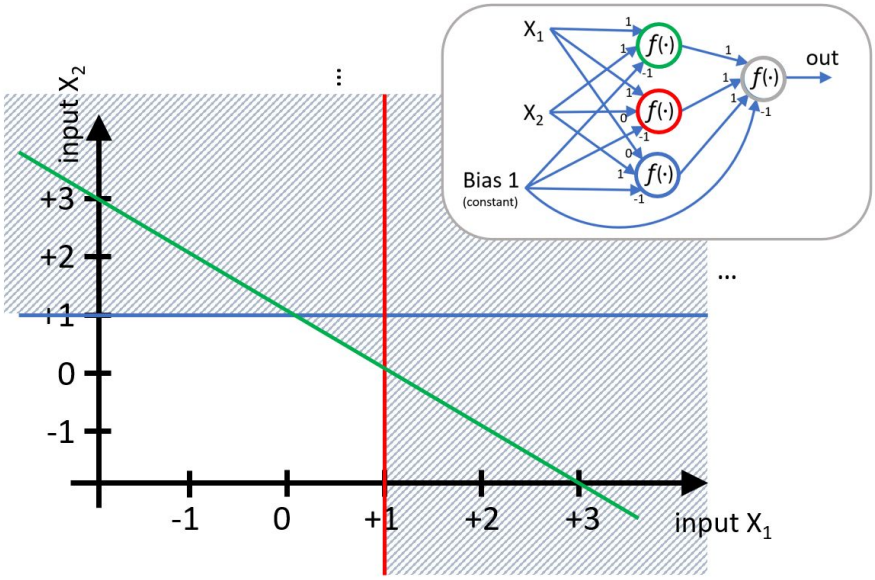
$$\text{step}\left(\sum_i (inp_i * wi)\right) = \begin{cases} 1 \text{ if } (\cdot) \geq 0 \\ 0 \text{ otherwise} \end{cases}$$

a) Draw an input space diagram (two dimensional plot of X1 and X2) and show for which area of the input space the network produces a positive output. (2p)

b) Can this network be implemented in a single neuron with linear activation function (yes/no)? Explain **in KEYWORDS**. (1p)

c) Assume all multiplicative weights in the network double their value. What happens with the output? (1p)

Solution:



a) Diagram shown above. Each separation line (colored) is given by one of the three neurons in the first layer. The output neuron is active if at least one of those neurons is active (OR function). Hence, the shaded region corresponds to the area of the input space (x1,x2), where the network generates a positive output.

b) The output function of this network is highly non-linear (a step with a piece-wise linear decision boundary in input space); hence a single linear neuron **CANNOT** implement this function.

c) **No change at the output**, as all input (including bias) for all neurons doubles. Therefore, all total input that was below zero is still below zero; all total input that was above zero is still above zero. All neurons show the exact same output signals.