Technical University of Denmark

02450 - Introduction to Machine Learning and Data Mining

E19

Project 2

Student:	**Student id**
Rasmus Tuxen	s173910
Callum Blair	s193096

November 12, 2019

Contents

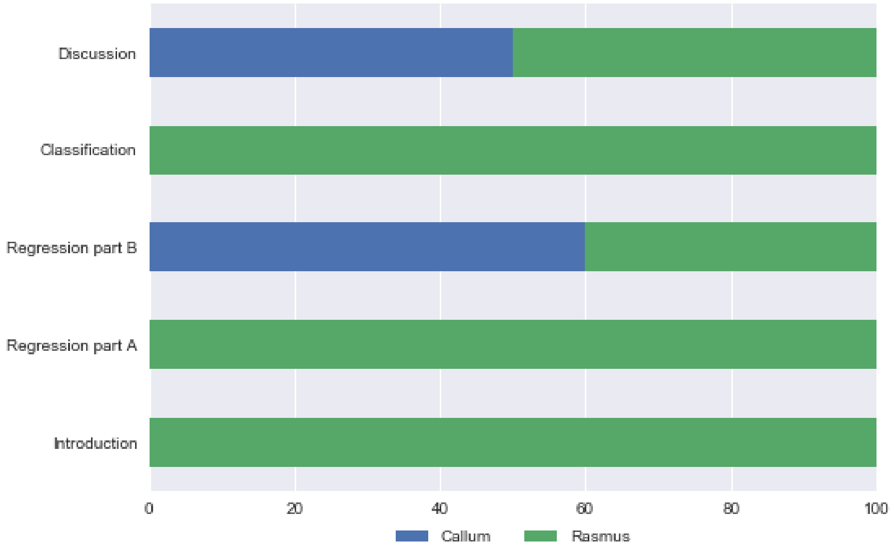

Figure 1: Split of responsibility per section

1 Introduction

This report is made for the course 02450: Introduction to Machine Learning and Data mining at DTU (Technical University of Denmark). The objective of this assignment is to analyse data using supervised learning methods using regression and classification. The data set selected is the Ecoli Data Set. The data set is retrieved from the UCI Machine Learning Repository. The title of the data set is *"Protein Localization Sites"*. It contains 336 instances and 9 attributes. A complete list of all the attributes featured in the data set can be seen in table 1.

	Variable:	**Description**	Type of attribute
1	**Sequence Name**	Accession number for the SWISS-PROT database	Discrete/Nominal
2	**mcg**	McGeoch's method for signal sequence recognition	Continous/Interval
3	**gvh**	von Heijne's method for signal sequence recognition	Continous/Interval
4	**lip**	von Heijne's Signal Peptidase II consensus sequence score	Discrete/Nominal
5	**chg**	Presence of charge on N-terminus of predicted lipoproteins	Discrete/Nominal
6	**aac**	score of discriminant analysis of the amino acid content of outer membrane and periplasmic proteins	Continous/Interval
7	**alm1**	score of the ALOM membrane spanning region prediction program	Continous/Interval
8	**alm2**	score of ALOM program after excluding putative cleavable signal regions from the sequence	Continous/Interval
9	**type**	The class is the localization site	Discrete/Nominal

Table 1: Table of attributes in Ecoli data set

2 Regression

The data set used in this report is one most suited for classification. However it is also possible to use the data set for a regression problem by selecting a less obvious feature to attempt to predict. For the following regression problem the continuous attribute **alm2** is selected as the value the models are going to predict.
In order to use the multi class attribute, **type**, for the regression problem, one-of-K coding is applied. Since regularization will be used for the following section about linear regression, a feature transformation to the data matrix, $\mathbf{X}$, so that the mean of each column is 0 and had a standard deviation of 1.
The estimated generalization error for the regression problem is the squared loss per observation:

$$E = \frac{1}{N^{test}} \sum_{i=1}^{N^{test}} (y_i - \tilde{y}_i)^2 \tag{1}$$

A Linear Regression

A linear regression model is trained on the data set with 1 level cross-validation with $K = 10$ folds. A regularization parameter λ is introduced and from each cross-validation fold the tested range of λ forr this model is a logarithmically spaced vector in the interval $[10^{-2}\ 10^{7}]$. The weights for each trained model is calculated using [1]:

$$\mathbf{w} = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{y} \tag{2}$$

The estimated generalization error as a function of λ can be seen in figure 2. In this particular example the regularization parameter which achives the lowest genearlization error is $\lambda = 1.048$, however it would appear that the optimal generalization error estimates are for small lambda from the look of the graph. The true optimal regularization parameter would then be $\lambda = 0$, which is the same as normal linear regression. Running the linear regression model multiple times with different random states confirms this suspicion as the optimal λ almost always is the lowest possible no matter the range tested.

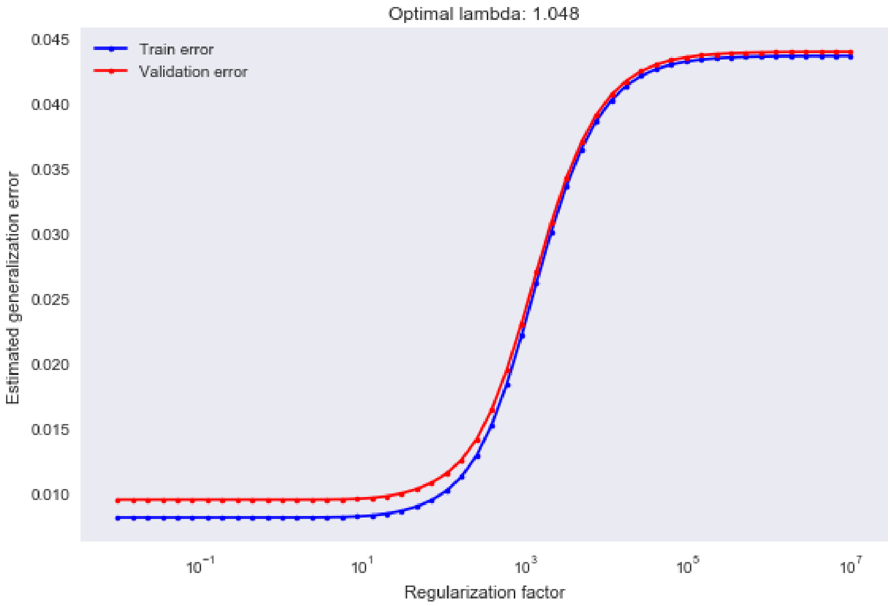

Figure 2: Estimated generalization error as a function of λ for regression problem with $K = 10$ cross-validation folds

The weights for the optimal regularization parameter $\lambda = 1.048$ can be seen in table 2. The parameters from 'cp' and down correspond to the one-of-K encoded parameters. From the weights it seems that 'alm1' has the biggest influence on the predicted value. This could however be explained by the fact that 'alm1' and 'alm2' has a somewhat strong correlation (see last report). Meanwhile the attributes 'aac', 'cp' and 'om' seem to have the next largest effect on determining the predicted value of 'alm2' in the linear regression model. Some other values like 'mcg' and 'imL' seem to be almost irrelevant for this model.

[1] Lecture_8.pdf - slide 11

Parameter	weight
Offset	0.5
mcg	0.004
gvh	-0.01
lip	0.012
chg	-0.012
aac	0.031
alm1	0.158
cp	0.026
im	0.018
imL	0.002
imS	-0.003
imU	0.017
om	-0.049
omL	-0.047
pp	-0.023

Table 2: Weights for attributes in the linear regression model

B Evaluation of 3 regression models

Three different regression methods were applied to the data set; Linear regression, Artificial Neural Network (ANN) and a baseline model. The models were evaluated using 2 level cross-validation with $K_1 = K_2 = 5$ folds where the inner fold screened for for the optimal complexity controlling parameters for each model while the outer cross validation level estimated the generalization error. The number of folds was chosen due to time constraints.

The complexity controlling parameter for the Linear Regression model is the regularization value λ. The tested range of λ for this model is a logarithmically spaced vector from $[10^{-2}\ \ 10^{7}]$.
The ANN regression model used number of hidden nodes as complexity controlling parameter $h = 1, 2, 3, 4$ which based on trial runs seemed to be a sufficient range. As increasing the number of nodes above 10 lead to little improvement in accuracy in trials with fewer folds.
The baseline model computes the mean of the training data and predicts everything in the test data as having that value. This means that the baseline model only made use of the outer cross-validation level since it didn't need to optimize any controlling parameters.
An example of how to optimal complexity controlling parameter was chosen for each model can be seen in figure 3, where the errors for the last inner loop is shown.

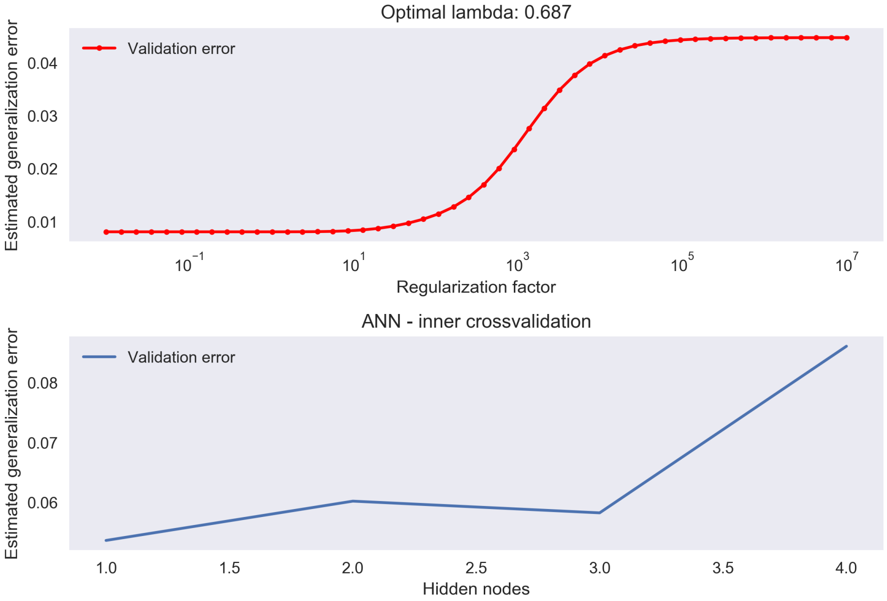

Figure 3: Regression error rate for complexity controlling parameters (top: Linear Regression bottom: ANN) from inner cross-validation level

For each of the outer fold the generalization error was computed for each model as well as the optimal complexity parameter. The result of the 2 level cross-validation can be seen in figure 3.

Outer fold	ANN		Linear regression		Baseline
i	h_i^*	E_i^{test}	λ_i	E_i^{test}	E_i^{test}
1	0.01	0.039	1	0.006	0.039
2	0.68664885	0.049	1	0.007	0.049
3	0.68664885	0.04	1	0.008	0.039
4	0.68664885	0.045	3	0.014	0.046
5	0.68664885	0.044	1	0.012	0.045

Table 3: Example Data: Two-level cross-validation table for three models for regression containing optimal complexity parameter and generalization error for each model

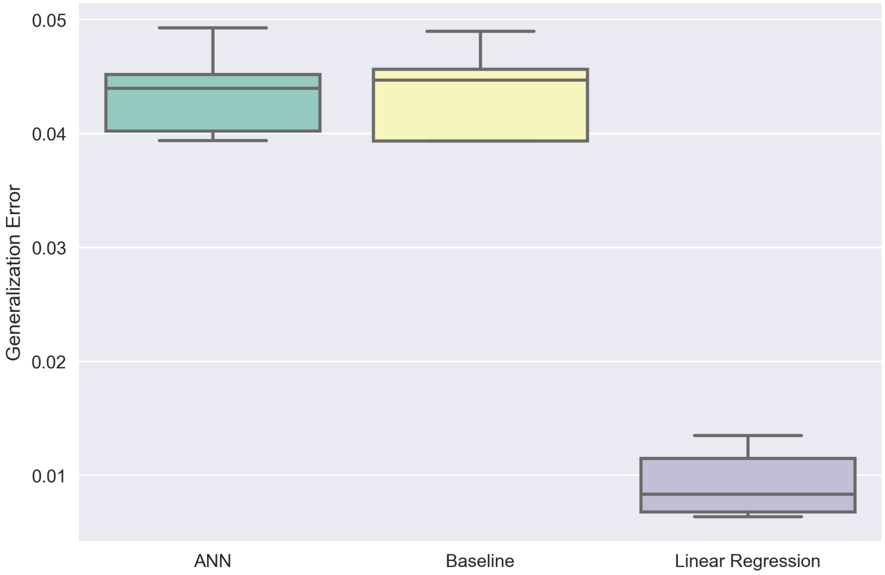

Figure 4: Boxplot of generalization errors from outer level of cross-validation for regression models

The table seen above can quickly allow for the evaluation of different models by both their complexity and error. For example, some models may be slightly more accurate but at a much higher cost of complexity.
In order to statistically compare the performance of the models a paired t-test was applied using $\alpha = 0.05$. **Setup I** was used for this evaluation.
From this evaluation we observe that the linear regression seems to be the best model for the regression problem (see figure 4), and the two p-values for the the paired t-test are both very low indicating strong evidence against the models being similar. Meanwhile from the confidence interval the baseline and ANN performs the same, but due to the p-value there is some evidence agains this hypothesis.

Model A	Model B	Confidence Interval	p - value
Baseline	Linear Regression	[0.03 0.04]	$2.00 \cdot 10^{-30}$
Baseline	ANN	$[-0.\ 0]$	0.4989
Linear Regression	ANN	$[-0.04 \ \ -0.03]$	$1.28 \cdot 10^{-30}$

Table 4: Pairwise statistical evaluation of the three regression models using **Setup I**

It should also be mentioned that the linear model for part a and part b recieved similar results.

3 Classification

In this section a classification problem for the data set is addressed. The obvious feature for a classification problem to be predicted is the **type** feature which contains 8 different protein locations in the observed cells. It is thus a multi class classification problem.

Three different classification methods were applied to the data set; Logistic regression, K-nearest-neighbor (KNN) and a baseline model.
The models were evaluated using 2 level cross-validation with $K_1 = K_2 = 10$ folds where the inner fold screened for for the optimal complexity controlling parameters for each model while the outer cross validation level estimated the generalization error, which for classification problem is the miss classification rate:

$$E = \frac{[Number\ of\ missclassified\ observations]}{N^{test}} \tag{3}$$

The Logistic regression model used is extended to be an 'multinomial regression' model to account for the multiclass classification problem. The difference from a regular logistic regression model is that the softmax function $f_c(o) = \frac{exp(o_c)}{\sum_{c'} exp(o'_c)}$ [2] is used to predict the multiple classes. The complexity controlling parameter for this model is the regularization value λ. The tested range of λ for this model is a logarithmically spaced vector from $[10^{-3}\ 10^{1}]$. This choice was based on trial runs.
The K-nearest-neighbor classification model uses complexity controlling parameter $k = 1, 2, ..., 40$ which based on trial runs seemed to be a sufficient range.
The baseline model computes the largest class on the training data and predicts everything in the test data as belonging to that class. This means that the baseline model only made use of the outer cross-validation level since it didn't need to optimize any controlling parameters.

An example of how the optimal complexity controlling parameter for each model in each iteration of the inner cross-validation level is selected can be seen on figure 5. In this graph the classification error rate for each complexity controlling parameter for each model is plotted. The parameter with best performance is selected to be trained in the outer level of the cross-validation. In this particular example the optimal k-nearest neighbor is $k = 7$ and the optimal regularization parameter is $\lambda \approx 1.29$. It should also be mentioned that all errors are from the test set and not the training error.

[2] wiki : softmax

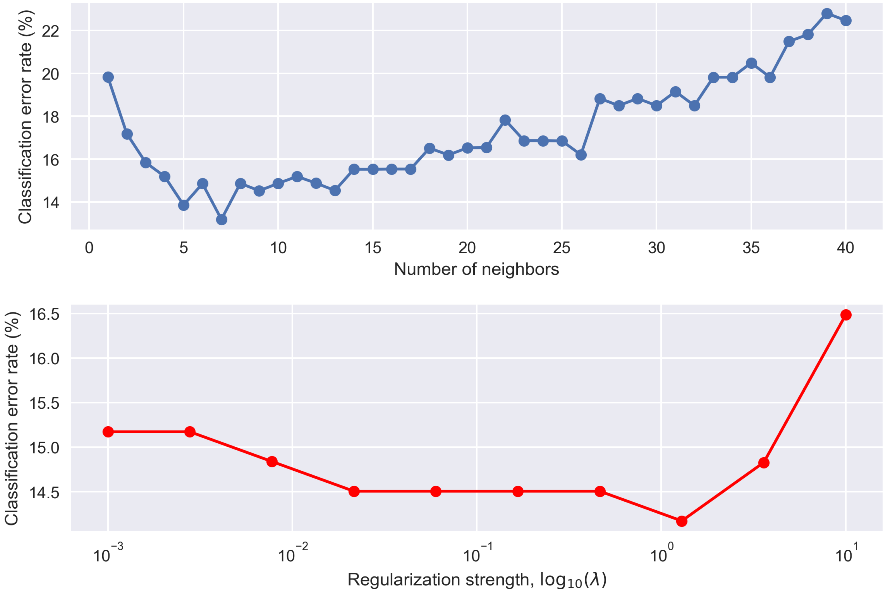

Figure 5: Classification error rate for complexity controlling parameters (top: KNN bottom: Logistical Regression) from inner cross-validation level

For each of the outer folds the generalization error was computed for each model as well as the optimal conplexity parameter for that model. The results for the 2 level cross-validation can be seen in table 5. From this table it is clear that the baseline has a poor miss-classfication rate which makes a lot of sense given that is a multi class classification problem. Meanwhile the logistical regression and k-nearest-neighbor models perform better with an average miss-classfication rate of ≈ 13% and ≈ 13% respectivly. This is more clear from figure 6, which shows the boxplot of all the gerneralization errors from table 5. The optimal nearest neighboor is 7, and the optimal regularization parameter seems to be in 1.29.

Outer fold	K-nearest-neighbor		Logistic regression		baseline
i	k_i^*	E_i^{test}	λ_i	E_i^{test}	E_i^{test}
1	7	0.09	1.29	0.09	0.5
2	7	0.09	1.29	0.09	0.56
3	7	0.09	1.29	0.09	0.62
4	7	0.18	1.29	0.18	0.65
5	7	0.15	1.29	0.15	0.56
6	7	0.15	1.29	0.15	0.53
7	7	0.12	1.29	0.12	0.73
8	7	0.12	1.29	0.15	0.52
9	7	0.06	1.29	0.03	0.55
10	7	0.27	1.29	0.3	0.55

Table 5: Two-level cross-validation table for three models for classification containing optimal complexity parameter and generalization error for each model

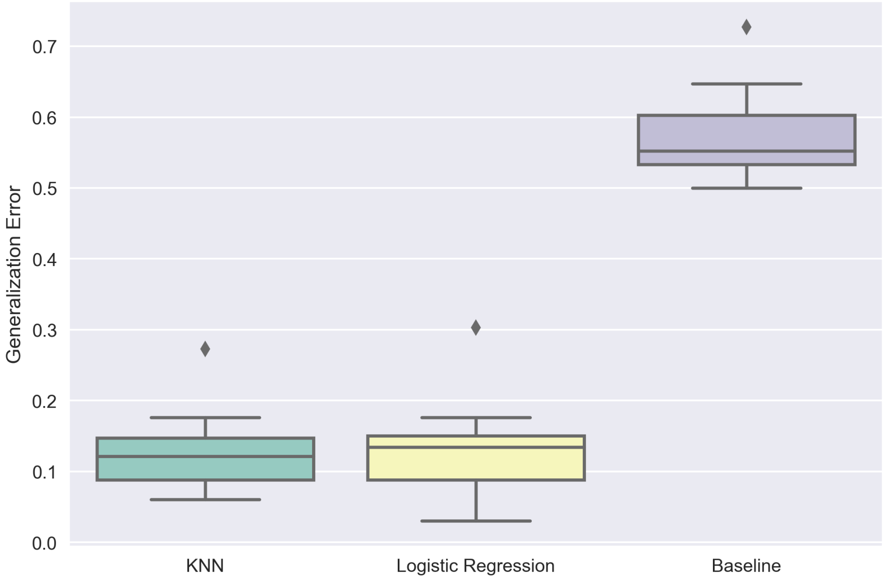

Figure 6: Boxplot of generalization erros from outer level of cross-validation for KNN, Logistic Regression and the baseline model

In order to statistically compare the performance of the the models, a paired t-test is applied using $\alpha = 0.05$. We have selected to use **Setup I** for statistical tests of performance considering the specific training set used. The result of the three pairwise tests can be seen in figure 6. Using McNemar's test to compare the accuracy difference $\theta = \theta_A - \theta_B$ a negative θ indicates that the performance of model B is better than model A.
For the two first pairwise tests the confidence interval doesn't contain 0 and the p-value is very low which is very strong statistical evidence that the logistical regression and k-nearest-neighbor models are better than the baseline model.
For the last paired test the confidence interval barely contains 0 which is weak evidence towards logistical regression having a higher accuracy than k-nearest-neighbor. But the p-value is relatively high, indicating there is little or no evidence against the null hypothesis and that the result is likely due to chance. This means that it can't be concluded that one model is better than the other from a statistical standpoint.

Model A	Model B	$\tilde{\theta}$	Confidence Interval	p - value
Baseline	KNN	-1.89	$[-0.48 \;\; -0.4]$	$2.06 \cdot 10^{-42}$
Baseline	Logistical Regression	-1.88	$[-0.48 \;\; -0.4]$	$5.33 \cdot 10^{-41}$
KNN	Logistical Regression	-0.99	$[-0.01 \;\; -0.02]$	1.00

Table 6: Pairwise statistical evaluation of the three classification models using **Setup I** (McNemera's test

A multi-class logistic regression modes is trained using a suitable value of the regularization parameter $\lambda = 1.29$. The model has computed a matrix of weight for each attribute for each class in the training phase. The number of attributes are 7 and classes are 8 and thus the weight matrix has dimensions $(8, 7)$. The model makes a prediction by multiplying each observation in the test data and then find the index with the highest probability using the *softmax* function.

$$\theta = softmax[\mathbf{x}^T\mathbf{w}_1 \ ... \ \mathbf{x}^T\mathbf{w}_K] \tag{4}$$

As an example the weights for the class 'cp' are $w = [-1.21 \ \ -1.08 \ \ -0.25 \ \ -0.02 \ \ -0.34 \ \ -2.16 \ \ 0.2]$, which means that the attributes with the greatest effect on this this model are 'mcg', 'gvh' and 'alm1'.

4 Discussion

Previously, this data set has been used mainly for classification. Namely, predicting the binding site of proteins. Therefore, the suitability of this data set for conducting regression tasks is questionable. As the tables have not been generated for the comparison, insight here has been taken from debug information outputted to the terminal.
Comparison of the performance of the models for the regression problem shows that the artificial neural network was not considerably more effective than the baseline. Indeed, the predictions of the ANN seemed to be strongly clustered around the mean with little variance. Meaning that the models gave similar predictions in practice.
The Linear Regression model performed the best of the regression models with an average generalization error or $E_{gen} = 0.0093$.
The fact that the baseline performs strongly and the ANN poorly may be an indication that the feature chosen for the regression problem cannot be well deduced from the other features. Or at least, that the feature was a poor choice for performing regression.
Furthermore, as the neural networks tended to predict around the mean in most cases, the optimum number of neurons was largely independent between outer folds and varied widely. Numbers of neurons up to 15 were trailed in the hidden layer, with no improvement. From

the classification problem the best models were KNN and Logistical Regression models. In the 2 level cross-validation the average mis-classification rate for the outer layer of the KNN model was $E_{KNN} \approx 13\%$ and for the Logistical regression was $E_{LOG} \approx 13\%$. From a statistical standpoint it cannot be concluded which of these were best because of a high t-value in the t-test.
Both of these models did however handsomely beat the baseline model that performed poorly. This was however expected since this was a multi-class classification problem with no clear class imbalance.

The data set has previously been analysed with machine learning models as a classification problem4. In this study the models feed-forward neural network, binary decision tree, naive Bayesian classifier and KNN was used and also as ensemble-based methods. The best performing ensemble method used achieved a classification rate of 91.7% and the average accuracy for single models was 86% with KNN as the best performing model.
Comparatively the accuracy for the KNN and Logistic Regression models in this report are $\approx 87\%$ which seem to fit well with the results achieved by the previous paper.

References

[1] Chen, Yetian : Predicting the Cellular Localization Sites of Proteins Using Decision Tree and Neural Networks
pdfs.semanticscholar.org/749e/51315f9a5fce4331994ffcab6a887aa1d5b6.pdf

[2] Aristoklis D. Anastasiadis, George D. Magoulas, and Xiaohui Liu : Classification of Protein Localisation Patterns via Supervised Neural Network Learning
http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.2.4228&rep=rep1&type=pdf

[3] wikipedia/Softmax_function
https://en.wikipedia.org/wiki/Softmax_function