MLEARN	*Friday, January 17, 2025*
Professors: Eduard Garcia, David Remondo, Josep Vidal, Veronica Vilaplana Duration of exam: 2h 30min Solve the exam in these sheets, do not deliver additional ones	

Exercise 1

For the following statements about units 1, 2 and 3, please mark 'T' for true and 'F' for false. Note that each incorrect response decreases the exercise score by the same magnitude as a correct response increases it. The final grade for this exercise will not fall below zero.

1	Based on how training data is handled, there are only two types of Artificial Intelligence (AI): supervised learning and unsupervised learning.	**F**
2	A *confidence* interval provides a range of values within which the true parameter value we are estimating from a sample falls with a given probability P. The *significance* level is then 1-P.	**T**
3	A test designed to detect a given disease shows low sensitivity (recall) but high precision. Then it is prone to false negatives (FN) while producing a low number of false positives (FP).	**T**
4	Testing a binary classifier, we obtain the following: TP=10; FP=10; FN=20; TN=960. We can conclude that, although it shows a high accuracy and specificity, this is not a reliable model.	**T**
5	An input feature showing a uniform distribution will have a similar kurtosis to a normal distribution but a lower skewness.	**F**
6	We use Minimum Bayesian Risk (MBR) to base our classification of tumor diagnoses as benign (class 1) or malignant (class 2). Setting $\Pi_{12} > \Pi_{21}$ seems like a good idea.	**F**
7	*Feature engineering*, *feature extraction* and *feature selection* are all different terms for the same family of techniques, which aim to reduce the number of input features.	**F**
8	*Whitening* transforms a vector of random variables (e.g. input feature vectors) into a set of new variables whose covariance is the identity matrix, thus simplifying its analysis.	**T**
9	We can transform a discriminant $g_i(x)$ by means of a non-decreasing function (e.g. ln()) to simplify it without affecting its criterion or decisions.	**T**
10	When input features are uncorrelated across all classes in a classification problem and can be assumed to be Gaussian, simple linear boundaries are sufficient to minimize the misclassification error rate.	**F**
11	MAP-based decisions cannot be applied when input features have arbitrary covariance matrices $\boldsymbol{C}_i$ for the different classes.	**F**
12	A Maximum a Posteriori (MAP) model classifies fish as salmon (ω_1) or sea bass (ω_2) based on length. We caught a fish of length x=0.7m. If $f(0.7\|\omega_1) = f(0.7\|\omega_2)$, the model can ignore its length and just decide the fish that is more frequent in those waters (i.e. the largest prior).	**T**
13	When the receiver operating characteristic (ROC) curve is a diagonal line dividing the sensitivity vs. specificity space into two equal areas, the discriminability (D') between classes is maximal.	**F**
14	If input features follow a Gaussian distribution, Maximum Likelihood (ML) estimates of their mean and covariance are simply the mean and covariance of the training samples.	**T**
15	An estimator with high bias and low variance means that it is consistent and stable across different training data partitions, but it is also inaccurate.	**T**
16	Increasing model complexity can reduce bias, but it may also increase variance due to the risk of overfitting.	**T**
17	Increasing the number of input features always reduces misclassification errors, but also leads to more complex models, requiring more time and computational resources.	**F**
18	Both principal component analysis (PCA) and multiple discriminant analysis (MDA) aim to linearly transform input data to produce a new set of features that are uncorrelated.	**F**
19	MDA transformation on an input vector $\boldsymbol{x}$ provides a new feature vector $\boldsymbol{z}$, the dimension of which (d') cannot exceed c - 1, where c is the number of different classes in the dataset.	**T**
20	Shapiro-Wilk and Anderson-Darling give name to two feature selection mechanisms that can be used as a modern alternative to PCA or MDA.	**F**

Exercise 2

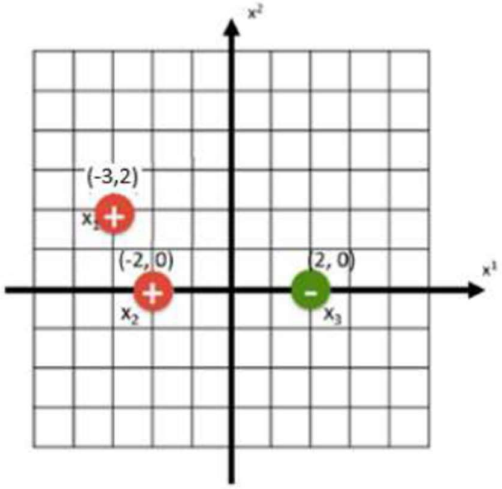
In the figures shown below we find two binary classification problems. The dots on the two-feature space correspond to two classes, as indicated by the plus (+) or minus (-) signs, respectively. The values of the coordinates of each dot are indicated on the figures. Each of the squares in the grids has a size of 1x1.

a) Consider that we use a hard margin linear SVM classifier (i.e. we use no slack variables) for the two problems. Circle the resulting support vectors for each of the problems. In each of the problems, what will be the distance between the margin boundaries?

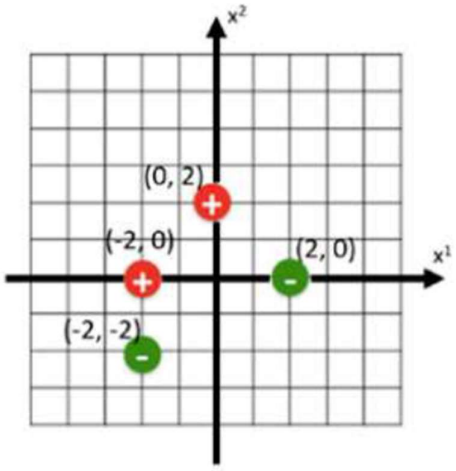
b) Now consider that a new labelled dot is provided on the left-hand side problem: the dot belongs to the minus (-) class and has coordinates (2, 4). Will the decision boundary of the hard margin linear classifier change? Will the margin boundaries change? What will be the number of support vectors after the addition of this dot?

c) On the right-hand side problem, if we resort to a soft margin linear SVM approach, how will the decision boundary evolve with decreasing values of the hyperparameter P? Motivate your answer.

d) The SVM method is based on an optimization procedure. Can the SVM be trapped in local minima? Explain your answer.



Problem A



Problem B

Solution:

a) In the left-hand side problem, the support vectors will be the dots on (-2,0) and (2,0) (indicated as x_2 and x_3 in the figure); the distance between the margin boundaries will be 4. In the right-hand side problem, the support vectors will be (-2,0), (-2,-2), and (2,0); the distance between the margin boundaries will be 2 cos(arctan(1/2))=1.789 (since the slope of the margins is 0.5, their angle with respect to the horizontal axis is 0.147π radians, and thus the distance between the two margins is the product of the distance between (-2,0) and (-2,-2), which is 2, by the cosine of that angle).

b) The decision boundary will not change; the margin boundaries will not change, either. The number of support vectors will be three after this addition.

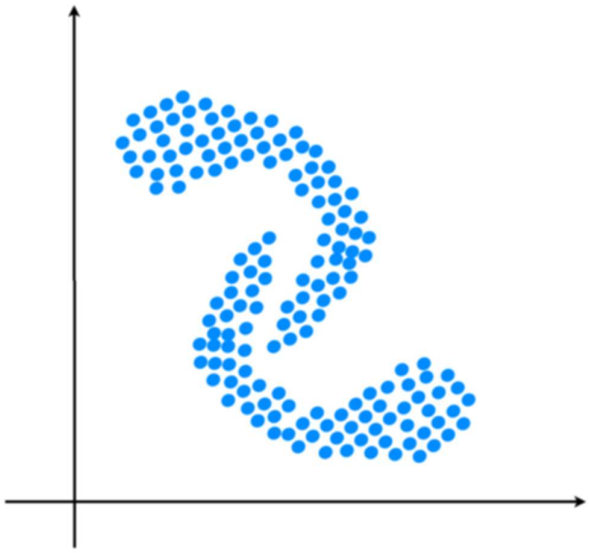
c) Decreasing values of P favour larger distances between margin boundaries despite the increased number of support vectors. If we start from the hard margin boundary case that we obtained in the first question, where P=∞, when we decrease the value of P, we will eventually come to a situation where all dots fall within the margin boundaries and will actually become support vectors; then, the decision boundary will come closer to dot (0,2) because this was the only dot that was not a support vector for the P=∞ case.

d) Local minima can appear in Gradient Descent based optimization methods applied to non-convex problems such as optimization of parameters in a Neural Networks. However, SVM is defined as a convex problem and hence the minimum is unique.

Exercise 3

We consider the two-dimensional set of unlabelled data in the figure shown below. Since data samples clearly seem to be grouped in two sets, for the next questions we assume that there are two classes.

a) How do you expect that the Expectation-Maximization (EM) algorithm with Gaussian mixture densities would perform? Explain your answer.
b) How do you think that the k-Means clustering technique will perform? Motivate your answer.



Solution:

a) Since the shape of each cluster is rather different from a Gaussian density, the algorithm will converge relatively slowly; we can expect that it will eventually yield two Gaussian densities centred close to the true centres of each cluster, with a rather circular shape (covariance matrix close to the identity); this will imply that in the resulting model, the samples of a cluster that are close to the centre of the other cluster will be classified as belonging to that other cluster.

b) K-Means is an iterative method that starts from a random choice of centroids for the different clusters (in this case, it is clear that we should choose k=2). Then, we re-assign the data samples to each cluster depending on its distance to the centroids using the Euclidean distance. From that, we re-calculate the centroids, and return to the previous step. In the problem at hand, we see that some data samples that lie in one cluster can be closer to the centre of the other cluster than many samples in the other cluster. This implies that the method will have important difficulties to converge, even after a large number of iterations.

Exercise 4

Neural networks

You want to solve a multi-class classification problem where samples belong to 3 classes (class 1, 2 and 3), and each sample is represented with 4 features, using a multilayer perceptron. The model has a single hidden layer with 3 neurons and ReLu activations, and the output layer has 3 neurons with softmax activation.

Consider the following input sample and parameter values:

input sample $x = [2.0, -1.0, 0.5, 1.0]$

1. Hidden layer

$$W^{(1)} = \begin{bmatrix} 0.2 & -0.3 & 0.5 & 0.1 \\ -0.4 & 0.7 & 0.2 & -0.1 \\ 0.6 & 0.6 & -0.5 & 0.0 \end{bmatrix} \qquad b^{(1)} = [0.1 \quad 0.0 \quad -0.2]$$

2. Output layer

$$W^{(2)} = \begin{bmatrix} 0.5 & 0.1 & -0.2 \\ -0.3 & 0.4 & 0.2 \\ 0.6 & -0.5 & 0.1 \end{bmatrix} \qquad b^{(2)} = [0.0 \quad 0.1 \quad 0.2]$$

a) Compute the forward pass and obtain the final scores for the input sample *x*. Which class does the MLP predict for this sample? (use argmax)

b) You decide to use the cross-entropy loss (logarithmic loss). Write the formula assuming that one-hot encoding is used to represent the ground truth. Compute the value of the loss for the input sample *x*, assuming that the ground truth class is 1.

c) During training, you can choose between batch gradient descent, mini-batch gradient descent, or stochastic gradient descent. Define each approach briefly. Give one advantage and one disadvantage of using mini-batch gradient descent compared to the other two.

d) You are considering basic SGD vs. Adam. State one main reason Adam might converge more quickly than plain SGD.

e) Would you consider reducing the learning rate over time, even when using Adam? Why or why not?

After training the MLP you observe that the model is overfitting, so you decide to add L2 regularization and Dropout.

f) Explain how each of these methods (L2 regularization and Dropout) helps reduce overfitting.

g) Discuss any drawbacks or trade-offs associated with each method (L2 regularization and Dropout), for example, how they might affect training speed or model capacity.

Solution

a)

$$z^{(1)} = W^{(1)}x + b^{(1)}$$
$$h = ReLU(z_1)$$
$$z^{(2)} = W^{(1)}h + b^{(2)}$$
$$\hat{y} = softmax(z^{(2)})$$
$$z^{(1)} = W^{(1)}x + b^{(1)} = [1.15 \quad -1.5 \quad 0.15]$$
$$h = ReLU(z_1) = [1.15 \quad 0 \quad 0.15]$$
$$z^{(2)} = W^{(1)}h + b^{(2)} = [0.545 \quad -0.215 \quad 0.905]$$
$$\hat{y} = softmax(z^{(2)}) = [0.344 \quad 0.162 \quad 0.494]$$

Predicted class = $argmax(\hat{y})$ -> class 3

b) $L(\hat{y}, y) = -\sum_{k=1}^{C} y_k \, log \; (\hat{y}_k) \; = -log(\hat{y}_p)$

were $y = (y_1, y_2, \dots, y_C)^T$ is the ground truth vector one-hot encoded, and $\hat{y} = (\hat{y}_1, \hat{y}_2, \dots, \hat{y}_C)^T$ is the predicted probability distribution over the C classes (C=3), and $\hat{y}_p$ is the softmax score predicted for the positive (ground truth) class.

For the given sample, L = -log(0.344)= 1.066

c) Batch Gradient Descent: Uses all training data for each parameter update (one update per epoch). Mini-Batch Gradient Descent: Uses small subsets to compute gradients and update parameters multiple times per epoch. Stochastic Gradient Descent: Updates parameters after every single sample.

Mini-Batch Advantage: Better computational efficiency on GPUs, and less noisy than pure stochastic updates. Disadvantage: Slightly more computational overhead than pure SGD per iteration; can still be noisier than batch GD.

d) Adam automatically adapts the learning rate per parameter based on moving averages of gradients and squared gradients. This often yields faster convergence in practice.

e) Even with Adam, it's common to reduce the learning rate over time (e.g., step decay or exponential decay). High initial learning rates speed up learning, but a lower learning rate later can help the model converge more precisely to a good minimum.

f) How Each Method Reduces Overfitting

L2 Regularization: it adds a penalty proportional to the sum of the squares of all weights. This forces the network to keep weights small, limiting the capacity to fit noise in the training data. Consequently, the model's complexity is reduced, helping it generalize better.

Dropout: Randomly "drops" (i.e., zeroes out) a fraction of neurons during training, preventing the network from relying too heavily on specific neurons or co-adaptations. This encourages the network to learn more robust, distributed representations, mitigating overfitting.

g) Drawbacks / Trade-Offs

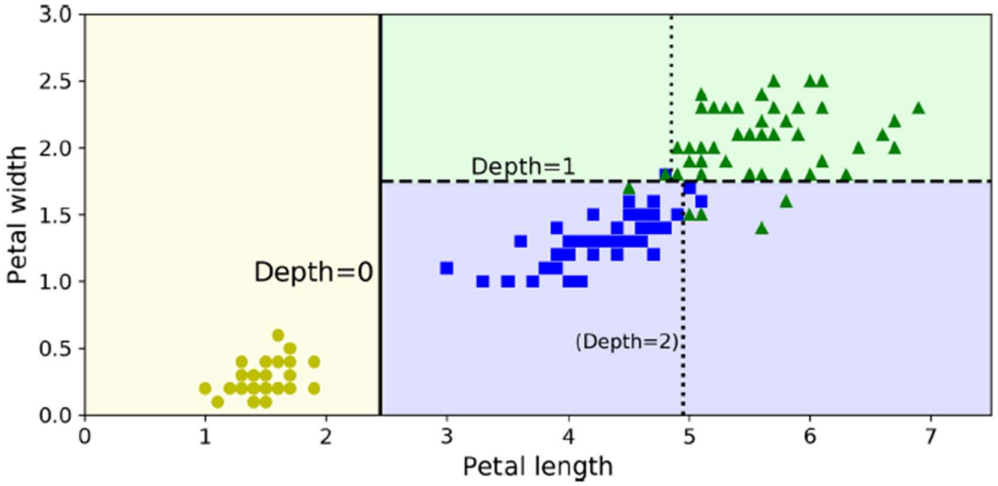
L2 Regularization: If the regularization coefficient is set too high, the model can underfit because weights become overly constrained. Training may take longer to converge because the model has to balance fitting the data with keeping weights small.

Dropout: Increases training stochasticity: with high dropout rates, the model may struggle to learn consistent feature mappings. Requires adjusting learning rates or other hyperparameters, because the effective network changes each mini-batch. Inference speed remains the same, but training often takes longer due to the noise introduced by dropping neurons.

Exercise 5

Decision trees

The following figure shows a plot of the database vectors and the decision boundaries of a three-class, two-features classification problem solved using a decision tree. The number of vectors of each class is Yellow: 22, Blue: 36, Green: 45.



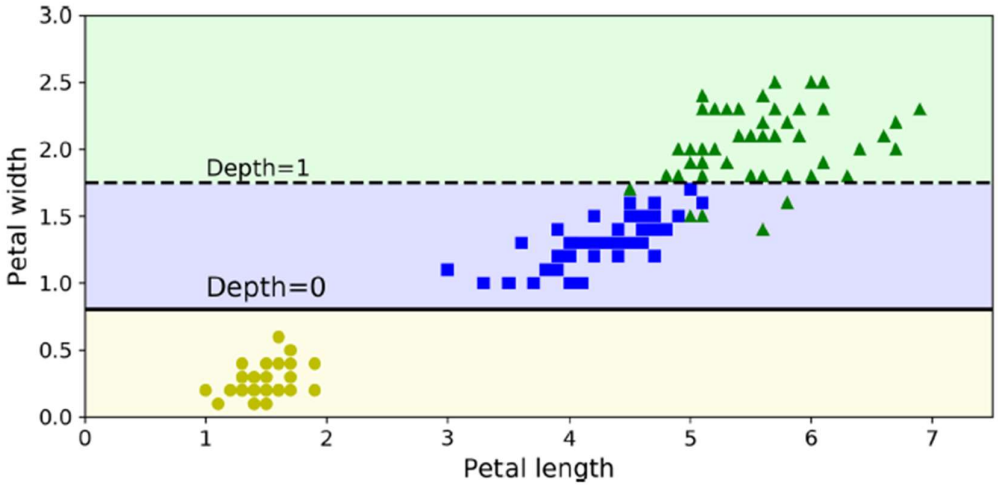
a) Draw the decision tree and clearly state the query and the number of vectors of each class at every node.

b) Check that the impurity of the root node is larger than the weighted sum of impurities of the two descendent nodes and explain why. You may use the Gini impurity: $i_G = \sum_i \Pr(\omega_i)\left(1-\Pr(\omega_i)\right) = 1 - \sum_i \Pr(\omega_i)^2$.

c) Do you think there is overfitting in the obtained tree? Explain why.

d) The vector **x** = (3.5, 0.75) is to be classified using this tree. Determine the posterior probabilities for each class, $\Pr(\omega_i|\mathbf{x})$, $\omega_i \in$ {yellow, blue, green}.

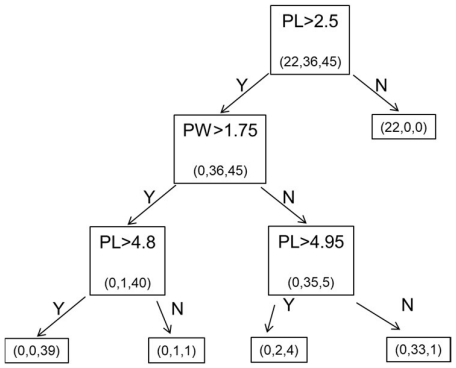
One of the vectors from the blue class has been removed from the training database, and the new decision boundaries obtained are these:



e) The large difference between the two trees is a consequence of the high variance of the tree models. Describe a method that can remove this unstability.

Solution:

a)



b) Impurity at the root: 0,641.
Impurity at the left descendent: 0,491.
Impurity at the right descendent: 0.

c) There is probably some overfitting because two leaves contain just a few vectors.

d) The vector is classified in the blue class.
Pr(Yellow|**x**) = 0, Pr(Blue|**x**) = 33/34, Pr(Green|**x**) = 1/34

e) The instability of decision trees comes from the high variance, which can be solved using boosting or random forests.

In Boosting, we randomly select subsets of the training data set allowing for replacement (i.e. a data sample can be selected in different subsets). Then we build a weak component classifier from each subset, and finally we build the ensemble classifier from the majority vote of the outputs of all the weak classifiers (or else from the combination of soft outputs). A different Decision Tree is built from each training data subset.

In Random Forests, we randomly eliminate features from the training process to build different trees. Then, the classifier results from the majority vote among all the different trees.