FUNDAMENTALS OF MACHINE LEARNING
AA 2023-2024
Prova Finale (FACSIMILE)

18 Dicembre, 2023

> **Istruzioni**: Niente libri, niente appunti, niente dispositivi elettronici, e niente carta per appunti. Usare matita o penna di qualsiasi colore. Usare lo spazio fornito per le risposte.
> **Instructions**: No books, no notes, no electronic devices, and no scratch paper. Use pen or pencil. Use the space provided for your answers.
> *This exam has 5 questions, for a total of 100 points and 10 bonus points.*

Nome: ______________________________

Matricola: ______________________________

1. **Multiple Choice**: Select the correct answer from the list of choices.

   (a) [5 points] True or False: A K-nearest neighbor classifier is only able to learn linear discriminant functions. ◯ True √ **False**

   (b) [5 points] True or False: Projecting a dataset onto its first principal component maximizes the variance of the projected data. √ **True** ◯ False

   (c) [5 points] True or False: The K-means algorithm is guaranteed to find the best cluster centers for any dataset. ◯ True √ **False**

   (d) [5 points] True or False: A Parzen kernel density estimator uses only the nearest sample in the dataset to estimate the probability of an input sample $\mathbf{x}$. ◯ True √ **False**

   (e) [5 points] How many parameters will a Multilayer Perceptron (MLP) for binary classification with a single hidden layer of width 10 and an input dimensionality of 8 have?

   ◯ 80 √ **99** ◯ 88 ◯ None of the above

   (f) [5 points] What will the entries of the Gram matrix be for a linear kernel?

   ◯ $K[i,j] = (\mathbf{x}_i^T \mathbf{x}_j)^\gamma$
   ◯ $K[i,j] = \exp(-\gamma ||\mathbf{x}_i - \mathbf{x}_j||_2^2)$
   √ $K[i,j] = \mathbf{x}_i^T \mathbf{x}_j$
   ◯ None of the above

   (g) [5 points] Which of the following loss functions is called the negative log likelihood?

   ◯ $\mathcal{L}(\mathbf{y}, \hat{\mathbf{y}}) = -\sum_{c=1}^{C} (\ln y_c - \ln \hat{y}_c)^2$
   ◯ $\mathcal{L}(\mathbf{y}, \hat{\mathbf{y}}) = -\sum_{c=1}^{C} (y_c - \ln \hat{y}_c)^2$
   √ $\mathcal{L}(\mathbf{y}, \hat{\mathbf{y}}) = -\sum_{c=1}^{C} y_c \ln \hat{y}_c$
   ◯ $\mathcal{L}(\mathbf{y}, \hat{\mathbf{y}}) = -\sum_{c=1}^{C} \ln \hat{y}_c$

   (h) [5 points] How many iterations of gradient descent must we perform for an epoch of minibatch Stochastic Gradient Descent with a dataset of 1024 samples and a batch size of 16?

   ◯ 1024 ◯ 1 ◯ 32 √ **64**

Total Question 1: 40

2. **Multiple Answer**: Select **ALL** correct choices: there may be more than one correct choice, but there is always at least one correct choice.

(a) [5 points] What are the advantages of projecting data onto $K < D$ principal components?

- √ **We eliminate noise in the original representation.**
- ◯ Classes are guaranteed to be linearly separable.
- ◯ It is a nonlinear embedding that makes learning easy with simpler models.
- √ **Models trained on the reduced data are simpler.**

(b) [5 points] Which of the following are advantages of Ensemble Models (e.g. Committees)?

- √ **They reduce the variance of the resulting model.**
- ◯ They are much more efficient than the base model.
- √ **They can reduce the expected error of the final model.**
- ◯ The resulting model is nonlinear even if the base model is linear.

(c) [5 points] Which of the following are causes of the vanishing gradients when training neural networks?

- √ **Saturated inputs to activation functions with near-zero derivatives when saturated.**
- ◯ Badly scaled input values.
- √ **Very deep models.**
- ◯ Bad random initialization of the network parameters.

(d) [5 points] If we want to penalize classification errors less when training an SVM we should

- ◯ Increase the hyperparameter C.
- ◯ Use a radial basis kernel.
- √ **Decrease the hyperparameter C.**
- ◯ None of the above.

(e) [5 points] Which of the following are requirements for applying backpropagation to compute gradients in a deep network?

- ◯ The network must not be too deep.
- √ **The network must be a directed acyclic graph.**
- √ **All activation functions must be differentiable.**
- ◯ All activation functions must be continuous.

(f) [5 points] Which of the following are true of the Nadaraya-Watson estimator?

- ◯ It only requires some of the training data at test time.
- √ **It is a nonparametric method.**
- √ **It estimates a nonlinear function of the input.**
- ◯ It estimates a linear function of the input.

(g) [5 points] Which of the following models are nonparametric?

- ◯ The Multilayer Perceptron (MLP).
- ◯ Logistic regression.
- √ **The K-Nearest Neighbor Classifier**
- ◯ Decision Trees.

Total Question 2: 35

3. [10 points] Show that a Committee Ensemble model using N bootstrapped linear regression models is a linear regression (i.e. that can be expressed as $\mathbf{w}^T\mathbf{x} + b$ for some $\mathbf{w}$ and b).

Solution: A committee model with N bootstrapped linear regression models has this form:

$$f(\mathbf{x};\theta) = \frac{1}{N}\sum_{n=1}^{N} \mathbf{w}_n^T\mathbf{x} + b_n$$

for $\theta = (\mathbf{w}_n, b_b)_{n=0}^{N}$. But then by linearity and commutativity of inner products we have:

$$\begin{aligned}
f(\mathbf{x};\theta) &= \frac{1}{N}\sum_{n=1}^{N} \mathbf{w}_n^T\mathbf{x} + b_n \\
&= \frac{1}{N}\sum_{n=1}^{N} \mathbf{w}_n^T\mathbf{x} + \frac{1}{N}\sum_{n=1}^{N} b_n \text{ (by linearity)} \\
&= \frac{1}{N}\mathbf{x}^T\sum_{n=1}^{N} \mathbf{w}_n + \frac{1}{N}\sum_{n=1}^{N} b_n \text{ (by commutativity of inner product)} \\
&= \frac{1}{N}\hat{\mathbf{w}}^T\mathbf{x} + \hat{b}
\end{aligned}$$

For the new model parameters $\hat{\theta}$:

$$\hat{\mathbf{w}} = \frac{1}{N}\sum_{n=1}^{N} \mathbf{w}_n \text{ and } \hat{b} = \frac{1}{N}\sum_{n=1}^{N} b_n$$

$\square$

4. [15 points] Show that a Multilayer Perceptron with two hidden layers with activation function $\sigma(x) = x$ is only capable of learning linear functions.

Solution: An MLP with two hidden layers computes the function:

$$\begin{aligned} f(\mathbf{x}) &= W_{\text{out}}\sigma(W_2\sigma(W_1\mathbf{x} + \mathbf{b}_1) + \mathbf{b}_2) + \mathbf{b}_{\text{out}} \\ &= W_{\text{out}}(W_2(W_1\mathbf{x} + \mathbf{b}_1) + \mathbf{b}_2) + \mathbf{b}_{\text{out}} \text{ (since } \sigma \text{ is the identity function)} \\ &= (W_{\text{out}}W_2W_1)\mathbf{x} + [W_{\text{out}}W_2\mathbf{b}_1 + W_{\text{out}}\mathbf{b}_2 + \mathbf{b}_{\text{out}}], \end{aligned}$$

which is a linear (well, affine) function $f(\mathbf{x}) = W\mathbf{x} + \mathbf{b}$ for:

$$\begin{aligned} W &= W_{\text{out}}W_2W_1 \\ \mathbf{b} &= W_{\text{out}}W_2\mathbf{b}_1 + W_{\text{out}}\mathbf{b}_2 + \mathbf{b}_{\text{out}}. \end{aligned}$$

□

5. [10 points (bonus)] Design a Deep Convolutional Neural Network (with at least three convolutional layers and one or more pooling layers) to classify MNIST images (input size 28×28). Draw the network (or write pseudocode for its definition) and indicate how many parameters each layer has and the sizes of the intermediate feature maps.

Solution: I will write pseudocode in tabular form for the definition of each layer (with corresponding numbers of parameters and size of the activations:

Layer	**Type**	**Activation Size**	**# Parameters**
1	Input	$1 \times 28 \times 28$	0
2	Conv2D(32, 1, 3, 3)	$32 \times 26 \times 26$	320 $(32 * 3 * 3 + 32)$
3	ReLU	$32 \times 26 \times 26$	0
4	Conv2D(32, 32, 3, 3)	$32 \times 24 \times 24$	9248
5	ReLU	$32 \times 26 \times 26$	0
6	MaxPool(2, 2)	$32 \times 13 \times 13$	0
7	Conv2D(16, 32, 3, 3)	$16 \times 11 \times 11$	4624
8	ReLU	$16 \times 11 \times 11$	0
9	Conv2D(16, 16, 3, 3)	$16 \times 9 \times 9$	2320
10	ReLU	$16 \times 9 \times 9$	0
11	MaxPool(2, 2)	$16 \times 5 \times 5$	0
12	Flatten()	400	0
13	Linear(400, 128)	128	51328
14	ReLU	128	0
15	Linear(128, 64)	64	8256
16	ReLU	64	0
17	Linear(64, 10)	10	650