Exam for ML for Physicists, PHYS-467, academic year 2023/24

Name/Sciper:

Instructions:

- Duration of the exam: 3 hours, 17. 01. 2024 from 15h15 to 18h15. Rooms CE14, CE16.
- Material allowed: 2 pages (i.e. one sheet recto-verso or two one-sided sheets) of personal notes. Pen and paper.
- Problems can be solved in any order.
- Write your full name on **each** additional sheet of paper you hand in.
- Total number of points is 90.

1 Match tools to tasks. [5 points]

Match one of the tools (a)-(e) to each task 1.-5. in the list below (1 point for each correct match):

Tools:

(a) Supervised regression, prediction

(b) Supervised regression, estimation

(c) Supervised classification

(d) Unsupervised clustering

(e) Unsupervised generative models

Tasks:

1. You have data about gene expression in tumours for a database of patients. You want to know whether there are several distinct types of tumours.
2. In a physics experiment, you measure the dependence of one quantity z on another r. You have a physical theory predicting how this dependence $z(r, \theta)$ should look like that is parametrized by some physically meaningful parameters of unknown value. You are interested in the value of these parameters.
3. In your e-mail inbox, you manually split the incoming mail into e-mails from your friends, work e-mail and advertisements. You would like to automatize this process.
4. You are a collector, and you want to estimate the price of a rare collectable item, such as a vintage movie poster, based on various features like the poster's dimensions, historical significance, the condition of the item, and the presence of signatures. You have a dataset of similar collectables and the prices for which they were sold on eBay.
5. You want to create a unique, artistic fashion design using a dataset of existing fashion designs.

Solution:

1, (d)
2, (b)
3, (c)
4, (a)
5, (e)

2 Maximum likelihood for Laplace distribution [8 points]

Consider a real random variable $x \in \mathbb{R}$ taken from the Laplace distribution

$$\rho(x) = \frac{1}{2b} e^{-\frac{|x-\mu|}{b}} \tag{1}$$

where $\mu, b \in \mathbb{R}$, $b > 0$.

1. (2 points) What is the mean and variance of this distribution?

   Solution: Mean μ, variance $2b^2$.

2. (3 points) Consider that one observed n independent samples from this distribution x_i, $i = 1, \ldots, n$. Use the maximum likelihood method to write the estimator of the constant μ.

   Solution: $\hat{\mu} = \arg\min_\mu \sum_i |x_i - \mu|$ which corresponds to the median of the samples. Add derivation.

3. (3 points) Consider that one observed n independent samples from this distribution x_i, $i = 1, \ldots, n$. Use the maximum likelihood method to write the estimator of the constant b.

   Solution: Similar to the decay constant,

   $$\hat{b} = \frac{1}{n} \sum_{i=1}^{n} |x_i - \hat{\mu}|$$

3 Weighted least squares [10 points]

Consider a regression problem on a dataset with input data $X \in \mathbb{R}^{n\times d}$. The output labels were obtained from noisy physical measurements. For each sample $\mu = 1, \ldots, n$ you are given the label in terms of its mean $\overline{y}_\mu \in \mathbb{R}$ and its standard deviation $\sigma_\mu \in \mathbb{R}$ as

$$\overline{y}_\mu \pm \sigma_\mu \tag{2}$$

As in the class, we will assume a probabilistic model where, for each sample μ independently, the output label was obtained as

$$y_\mu = \sum_{i=1}^{d} X_{\mu i} w_i^* + \xi_\mu \tag{3}$$

for some unknown ground truth vector of parameters $w^* \in \mathbb{R}^d$, and unknown Gaussian-distributed noise ξ_μ.

1. (1 point) Given the above dataset $\{X_\mu, \overline{y}_\mu, \sigma_\mu\}_{\mu=1}^n$ and assumptions, what is a sensible choice for the mean and the variance of the noise ξ_μ?

   Solution: The measurement noise ξ_μ is Gaussian with zero mean and variance σ_μ^2.

2. (2 points) Write the likelihood for the probabilistic model above.

   Solution: The true labels are $y_\mu \sim \mathcal{N}(\overline{y}_\mu, \sigma_\mu^2)$ independently for each sample. The likelihood is then

   $$P(\overline{y}, \sigma_\mu | X, w) = \prod_{\mu=1}^{n} \frac{1}{\sqrt{2\pi}\sigma_\mu} e^{-\frac{(\overline{y}_\mu - \sum_{i=1}^d X_{\mu i} w_i)^2}{2\sigma_\mu^2}} \tag{4}$$

   .

3. (2 points) Write a loss function $L(w)$ that, when minimized, yields the maximum likelihood estimator for the probabilistic model above.

   Solution: Maximization of the likelihood corresponds to minimization of $-\log$

   $$L(w) = \frac{1}{n} \sum_{\mu=1}^{n} \frac{(\overline{y}_\mu - \sum_{i=1}^d X_{\mu i} w_i)^2}{\sigma_\mu^2} \tag{5}$$

   The overall normalization of the loss can differ, but all are to be considered correct.

4. (1 point) Consider that a few samples in the dataset have particularly large standard deviation σ_μ. Justify why the loss you wrote is advantageous compared to the standard square loss.

Solution: The samples are weighted by $1/\sigma_\mu^2$; thus, those with large errors have a very small weight in the loss function.

5. (4 points) Consider a loss function of the form

$$L(w) = \frac{1}{n}\sum_{\mu=1}^{n}\left[a_\mu(\overline{y}_\mu - \sum_{i=1}^{d} X_{\mu i}w_i)^2\right] \tag{6}$$

where a_μ is a given sample-dependent quantity. Write the minimizer $\hat{w} = \text{argmin}_w L(w)$ in a matrix notation analogously as we had $\hat{w}^{\text{OLS}} = (X^TX)^{-1}X^Ty$ for the ordinary least-square loss.

Solution: One needs to compute the derivatives and set them to zero to obtain

$$\hat{w} = (X^TAX)^{-1}X^TAy \tag{7}$$

where A is a $n \times n$ diagonal matrix with a_μ on the diagonal.

4 (Stochastic) Gradient descent for regression and classification [7 points]

1. (1 point) Consider a linear classification problem, where we are given a dataset $(X_\mu, y_\mu)_{\mu=1}^n$ with $X_\mu \in \mathbb{R}^d$, $y_\mu \in \{\pm 1\}$. Consider the following loss function

$$L(w) = \sum_{\mu=1}^{n} \delta_{z_\mu(w), y_\mu}$$

where we defined $z_\mu(w) = \text{sign}(\sum_{i=1}^d X_{\mu i} w_i)$, $\delta_{a,b}$ is the Kronecker delta, and $w \in \mathbb{R}^d$ is a parameter vector. Describe briefly the main difficulty with minimizing this loss.

Solution: Answer: The error is not differentiable.

2. (1 point) What is an example of a loss that solves this issue for classification problems? Write down its expression.

Solution: Answer: E.g. Logistic loss.

3. (2 points) Gradient Descent (GD) can be used to minimize the loss function from point 2. Explain one issue with GD when the number of samples is very large. Define stochastic gradient descent (SGD) and explain how it solves this issue.

Solution: The shortcoming of GD is that it is slow and expensive as it requires computing the gradient over all samples. Advantage of SGD: each iteration is faster; it does not require computing the gradient over all the samples

4. (1 point) Consider now the linear regression problem with $(X_\mu, y_\mu)_{\mu=1}^n$ with $X_\mu \in \mathbb{R}^d$, $y_\mu \in \mathbb{R}$. We aim to solve it by minimizing the usual square loss. Consider a training set of n samples such that $d > n$. Is the minimizer of the square loss unique? Briefly justify.

Solution: No. There is a subspace of dimension at least $d - n$ of solutions.

5. (2 points) Consider gradient descent applied on the square loss from point 4, initialized at $w^{t=0} = \vec{0}$. To what solution does the gradient descent converge? The derivation is not needed. Hint: Recall that this property of GD is called *implicit regularization*.

Solution: GD converges to the least square norm estimator, corresponding to the solution found by infinitesimal square regularization.

5 Singular Value Decomposition (6 points)

Given a matrix $X \in \mathbb{R}^{n \times m}$, recall that its (squared) Frobenius norm is defined as $||X||_F^2 = \text{tr}(X^T X)$.

1. (2 points) Prove that
$$\|X\|_F^2 = \sum_{i=1}^{\min\{n,m\}} \sigma_i^2 \tag{8}$$
where σ_i indicates the i-th singular value of $\mathbf{X}$ (ordered by size).
(Hint: *Remember that* $\text{tr}(ABCD) = \text{tr}(BCDA) = \text{tr}(CDAB) = \text{tr}(DABC)$)

Solution: $\text{tr}\left(X^\top X\right) = \text{tr}\left(V\mathbf{\Sigma}^\top U^\top U\mathbf{\Sigma}V^\top\right) = \text{tr}\left(V\mathbf{\Sigma}^\top\mathbf{\Sigma}V^\top\right) = \text{tr}\left(\mathbf{\Sigma}^\top\mathbf{\Sigma}V^\top V\right) = \text{tr}\left(\mathbf{\Sigma}^\top\mathbf{\Sigma}\right)$ (1 points if they start the derivation using SVD. 0 otherwise)

2. (2 points) How can we obtain the best (in the Frobenius norm sense) rank-k approximation X_k of X? (Only state the result; proof is not required.)

Solution: The Young-Eckart theorem states that the best rank-k approximation of X is given by $X_{\mu i}^{(k)} = \sum_{\alpha=1}^{k} U_{\mu\alpha}\sigma_\alpha V_{i\alpha}$. (2 points if the truncated SVD is correctly written, 0 otherwise.)

3. (2 points) Show that the squared error of the best low-rank approximation (as obtained in point 2) can be expressed as
$$\|X - X_k\|_F^2 = \sum_{i=k+1}^{\min\{n,m\}} \sigma_i^2 \tag{9}$$

Solution: Follows immediately from point 1 and $X - X_k = U\,\text{diag}\left(0, \ldots, 0, \sigma_{k+1}, \ldots, \sigma_{\min\{n,m\}}\right) V^\top$ (1.5 points is the correct intuition is provided. -0.5 points for every inconsistency in arguments.)

6 MCMC sampling a ball [9 points]

Consider the d-dimensional ball defined as $B_d = \{x \in \mathbb{R}^d \text{ such that } ||x|| \leq 1\}$. You aim to sample uniformly points inside the ball. We define the following algorithms:

(a) Direct sampling: generate points uniformly in $[-1, 1]^d$, and reject all samples that fall outside B_d.

(b) MCMC 1: Random walk inside the ball. Consider every step after a certain time as a sample. Every time a step makes you exit from the ball, do not accept the step but still keep the point where you are as a sample and augment the counter of time steps.

(c) MCMC 2: Random walk inside the ball. Consider every step after a certain time as a sample. Every time a step makes you exit from the ball, do not accept the step and re-draw a new step (without augmenting the counter of time steps) until it stays in the ball.

1. (2 points) State the main condition we covered in class that ensures the target probability distribution is stationary for a given Markov chain.

   Solution: The detailed balance. For a uniform distribution, it implied that transition probabilities are the same in both directions.

2. (3 points) Which of the above algorithms sample the ball uniformly at random? Provide a brief reason for each.

   Solution: (a) works by definition, (b) works, the detailed balance is satisfied, (c) violates detailed balance.

3. (2 points) Compare, in words, the efficiency of the direct sampling and MCMC approach in the context of moderate dimensionality, say $d = 2$, of the ball.

   Solution: Comparable. The rejection rate of direct sampling is less than 1/2, for MCMC it depends on the stepsize. Direct sampling may even be more efficient.

4. (2 points) Compare, in words, the efficiency of the direct sampling and MCMC approach in the context of high dimensionality d of the ball.

   Solution: If the dimension is large, then the direct sampling will perform poorly. The volume of the disk becomes exponentially small with respect to the volume of the cube as d increases, meaning that the overwhelming majority of points sampled while running direct sampling will be rejected. Thus, MCMC is to be preferred in large dimensions.

7 Kernel feature maps [9 points]

Consider the following kernel in d dimensions $K(x,y) = (1 + x^\top y)^2 = (1 + \sum_{i=1}^d x_i y_i)^2$ for $x, y \in \mathbb{R}^d$.

1. (3 points) Write a feature map associated to the kernel K.

Solution: Let us expand the square as

$$K(x,y) = 1 + 2\sum_{i=1}^d x_i y_i + \sum_{i=1}^d x_i^2 y_i^2 + 2\sum_{i<j}^d x_i y_i x_j y_j = \phi(x)^\top \phi(y) \tag{10}$$

where $\phi(x) = \left(1, \sqrt{2}x_1, \sqrt{2}x_2, ..., \sqrt{2}x_d, x_1^2, \ldots, x_d^2, \sqrt{2}x_1x_2, \sqrt{2}x_1x_3, ...\right)$. The first component is 1, the following d components correspond to the components of x, and the following d components correspond to x_i^2 and finally all the products of two components $x_i x_j$ for $1 \leq i < j \leq d$. //Be careful when grading; the correct answer is not unique.

2. (1 point) What is the dimension of the feature space you proposed?

Solution: The feature space is the length of $\phi(x)$ as a vector. By the above solution, it is $1 + 2d + d(d-2)/2$.

3. (3 points) If you are given sufficiently many samples, can you learn up to zero test error the function $f(x) = (\sum_{i=1}^d x_i^2)^2$ using kernel regression with K? *Briefly* justify your answer.

Solution: Kernel regression corresponds to linear regression in feature space. The feature map only involves up to quadratic terms in the components of x, while $f(x)$ involves quartic terms like $x_1^2 x_3^2$. It can thus not be learnt perfectly.

4. (1 points) Same question for $f(x) = x_1x_2 + x_4$? *Briefly* justify your answer.

Solution: This function is quadratic, so yes.

5. (1 points) Same question for $f(x) = \cos(x_3)$? *Briefly* justify your answer.

Solution: As argued above, only quadratic polynomials can be learned perfectly. Cosine is not a quadratic polynomial, so it cannot be learnt up to zero error.

8 Deep Learning - CNNs [5 points]

Consider a dataset of black and white images of size 13×13 and consider a convolutional neural network architecture composed by the following layers: (i) one convolutional layer with 3 channels, filter sizes 4×4 and stride 1 (no padding is applied); (ii) one max pooling layer with filter size 2×2; (iii) one fully connected layer linking the flattened output from the previous layer to 4 output neurons.

- (3 points) Write the dimensions of the internal representations after each hidden layer, including the output one.

  Solution: After convolutional layer: $10 \times 10 \times 3$; after max pooling layer: $5 \times 5 \times 3$; after fully-connected layer: 4 (1 point for each layer, the answer with max pooling with stride 1 is considered correct.)

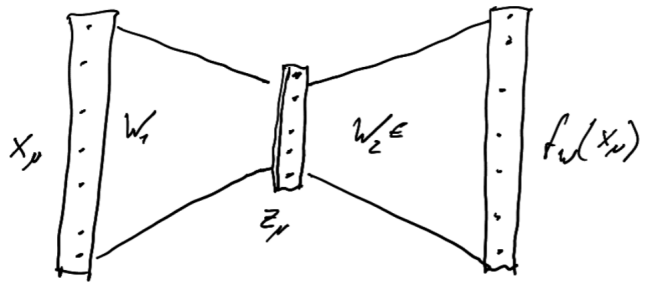
- (2 points) Write the total number of learnable weights in the considered neural network (excluding biases).

  Solution: Convolutional layer: $4 \times 4 \times 3 = 48$; max pooling layer: 0; after fully-connected layer: $5 \times 5 \times 3 \times 4 = 300$. 348 parameters in total. (1 point if they get at least the params of the conv layer. 0 otherwise)

9 Autoencoder [9 points]

1. (3 points) Consider a dataset $X \in \mathbb{R}^{n \times d}$. Draw a schematic of a fully connected auto-encoder with one hidden layer of width p. Add symbols for the weights and activations. Write the function that the drawing represents for a generic nonlinearity ϕ (you may omit biases). State the dimensionality of each variable.

Solution:



The network function is $f_w(X_\mu) = W_2\phi(W_1 X_\mu) \in \mathbb{R}^d$. $W_2 \in \mathbb{R}^{d \times p}$, $W_1 \in \mathbb{R}^{p \times d}$, $X_\mu \in \mathbb{R}^d$

2. (2 points) Write the square loss for the training of the auto-encoder as a function of the trainable parameters.

Solution:

$$\mathscr{L}(W_1, W_2) = \frac{1}{n} \sum_{\mu=1}^{n} (X_\mu - f_w(X_\mu))^2$$

3. (4 points) Assume the auto-encoder has been repeatedly trained on a large structured dataset for varying sizes p of the hidden layer. How do you expect the training and test losses to behave as a function of p? Hint: For the test loss, compare the classical expectation for its dependence on p (based on the notion of overfitting) with the common property of modern neural networks, referred to as overparametrization.

Solution: Training loss goes down with p, the more parameters, the easier the optimization. Classically, test loss should be U-shaped, but in deep learning, it may keep decreasing with p because of overparametrization, still leading to good generalization.

10 Maximum entropy distribution, mean [5 points]

(5 points) Consider a non-negative random variable $x \in \mathbb{R}_+$. We observe its mean $\mu \geq 0$ from data. Derive the probability distribution $P(x)$ that maximizes the Shannon entropy of $P(x)$ given the constraint that the mean of the distribution is equal to the observed one.

Solution: To solve the functional optimization problem, we introduce the following Lagrangian

$$L(P(\cdot), \lambda_\mu, \lambda_\Delta) = -\int P(x) \log P(x) dx - \lambda_\mu \left(\mu - \int xP(x)dx\right) - \lambda_n \left(1 - \int P(x)dx\right)$$

we now take the functional derivative of L w.r.t. $P(\cdot)$. This gives

$$\frac{\delta L}{\delta P(x)} = -\log P(x) - 1 + \lambda_\mu x + \lambda_n.$$

Setting this derivative to zero gives $P(x) = \exp\left[\lambda_n - 1 + \lambda_\mu x\right]$. The multipliers λ_n, λ_μ will take the values that satisfy the constraints respectively on normalization and mean. So that at the end

$$P(x) = \frac{1}{\mu} e^{-x/\mu}$$

11 Course questions: [17 points]

1. (1 point) Assume you have 5 classes with labels $y_i \in \{0, 1, 2, 3, 4\}$. Write one-hot encoded versions of labels $y_1 = 3$ and $y_2 = 0$.

   Solution: $y_1 = (0, 0, 0, 1, 0)$
   $y_2 = (1, 0, 0, 0, 0)$

2. (5 points) Classify the following quantities as parameters (P) or hyperparameters (H) of a learning algorithm.
   - The key, query and value matrices in attention layers.
   - The number of clusters in k-means clustering.
   - The minibatch size in stochastic gradient descent.
   - The coordinates of the centroid in Gaussian mixture clustering.
   - The discretization step in diffusion-based generative models.

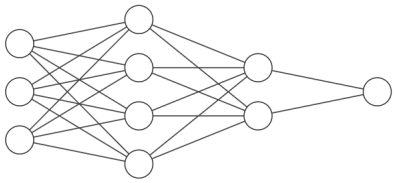
   Solution:
   - The key, query and value matrices in attention layers. (P)
   - The number of clusters in k-means clustering. (H)
   - The minibatch size in stochastic gradient descent. (H)
   - The coordinates of the centroid in Gaussian mixture clustering. (P)
   - The discretization step in diffusion-based generative models. (H)

3. (1 point) Derive Bayes theorem from the definition of conditional probability.

   Solution: By definition $p(a, b) = p(a|b)p(b) = p(b|a)p(a)$. Thus $p(a|b) = \frac{p(b|a)p(a)}{p(b)}$.

4. (3 points) Write the function represented by a feed-forward fully connected neural network with two hidden layers. Input size d, size of the first hidden layer p_1, and the 2nd p_2, the output is scalar. Use the ReLU activation function in the first hidden layer and tanh in the second. Use a bias term in all the layers. Write the dimensions of each of the quantities you use. Draw a schema of the same network for $d = 3$, $p_1 = 4$, $p_2 = 2$.

Solution: The function is $f(x) = W_3 \tanh(W_2 \operatorname{relu}(W_1 x + b_1) + b_2) + b_3$ with $x \in \mathbb{R}^d$, $W_1 \in \mathbb{R}^{p_1 \times d}$, $W_2 \in \mathbb{R}^{p_2 \times p_1}$, $b_1 \in \mathbb{R}^{p_1}$, $b_2 \in \mathbb{R}^{p_2}$, $W_3 \in \mathbb{R}^{1 \times p_2}$. The network looks like this:



5. (1 point) Consider that you use gradient descent to train a neural network on a given training data. At convergence, the training error is very close to 0. Does this mean that you obtained a good neural network that will be useful in the task you are interested in? Justify briefly.

Solution: No, test error can still be large.

6. (2 points) Consider that you are working with a dataset for predicting house prices. You notice that the dataset contains some outliers due to exceptionally high-value properties. Describe the concept of robust linear regression and how it helps in dealing with outliers in the dataset. In particular, write a loss function you would use to mitigate the influence of the outliers.

Solution: Samples far away from the prediction should be penalized less than in the square loss. To reduce the sensitivity to outliers it is possible to use the *mean absolute error loss*, which is given by

$$\ell(y_\mu, X_\mu \cdot w) = \gamma |y_\mu - X_\mu \cdot w|. \tag{11}$$

7. (3 points) Consider a Markov chain on a state space A defined by the transition probability $p(a \to b)$ of going from state $a \in A$ to state $b \in A$ at each time step. Consider a probability distribution $\pi(a)$ on the state space, and suppose that it satisfies the detailed balance condition. Write this condition and use it to prove that $\pi(a)$ is a stationary distribution for the Markov chain, i.e.,

$$\sum_a \pi(a) p(a \to b) = \pi(b)\,. \tag{12}$$

Solution:

$$\pi(a)p(a \to b) = \pi(b)p(b \to a)\, \forall a, b \in X\,. \tag{13}$$

To prove that π is stationary, we have

$$\sum_a \pi(a)p(a \to b) = \sum_a \pi(b)p(b \to a) = \pi(b) \sum_a p(b \to a) = \pi(b) \tag{14}$$

where we used first the detailed balance condition, and then that the transition probability is normalised.

8. (1 point) For an input $X_\mu \in \mathbb{R}^d$, consider the deep linear network with output $f(X_\mu) = W_L W_{L-1} \cdots W_2 W_1 X_\mu$, where W_i are weight matrices of appropriate dimension. We train this neural network to predict a scalar label y_μ. Does increasing the depth L and the width of the hidden layers allow us to model more and more complex relations between X_μ and y_μ? Justify briefly.

Solution: No, as the previous model is always linear in X_μ independently of L. Activations need to be non-polynomial for universal approximation theorem to hold.