Test Exam Answers

November 2017

1 Which of the following can cross-validation not help us obtain?

(a) A more accurate estimate of the final ("true") performance of a model
(b) Values for so-called "hyper-parameters" of a model, such as the number of clusters in a clustering problem, or the amount of regularisation in a prediction problem
(c) For a given model, a better estimation of the parameter values
(d) An quantification of the uncertainty over the estimated true performance of the model
(e) None of the above

Answer: C
Reason: Parameters are elements, like weights. These are optimized using batch gradient decent (or similar). KCV optimizes how the model is learned, aka the hyper-paramenters. Such as learning rate, number of neurons in a hidden layer etc.

2 Suppose that we are training a linear classifier using the perceptron learning rule and that the current linear classifier is given by the line 2 - x1 + 2x2 = 0. The next feature point in our training set is given by x = (3, 6). Assume that this feature point is misclassified, what will be the new value for the weight w if one applies a learning rate of 0.4?

(a) (1.0, 3.0, 6.0)
(b) (1.6, −2.2, −0.4)
(c) (2.4, 0.2, 4.4)
(d) (2, −2.2, −0.4)

Answer: B
For LINEAR:

$$w^t = w^{t-1} + \lambda x_n t$$

For linear perception if $y > 0$ is classified as $+1$, for $y < -1$ is classified as -1 Filling in the variables gives us $x_1 = 3$, $x_2 = 6$, $w_0 = 2$, $w_1 = -1$ and $w_2 = 2$. Filling this in formula given results in 11, hence it is classified as $+1$. The question states that it is classified incorrectly. hence t=-1. Just fill in the new weight formula.

3 Once again consider the situation of the previous question. In addition to the perceptron learning rule with learning rate 0.4 one also applies regularization of the form $|w1| + |w2|$ (L1 regularization) with parameter 0.2. What will now be the new value of w?

(a) (1.6, −2.0, −0.8)
(b) (1.6, −2.0, −0.2)
(c) (1.4, −2.0, −0.2)
(d) None of the above

Answer: B

The bars are not regarded as absolutes! According to the answer sheet. However, all other sources say they are absolutes.
Make sure to not update w_0 since it is not included in regularization.

$$w^t = w^{t-1} + \lambda x_n t + \eta(|w_1| + |w_2|)$$

Just filling in the variables gives us the answer.

4 Consider the neural network (NN) for which the input is 2 dimensional and that there are 3 neurons in the hidden layer and that there is 1 output neuron. The activation for all hidden neurons and the output neuron is the sigmoid function σ. The weights of the NN are as follows. Hidden layer: $w^{(1)}_{1,0} = -5$, $w^{(1)}_{1,1} = 1$, $w^{(1)}_{1,2} = 2$ $w^{(1)}_{2,0} = -3$, $w^{(1)}_{2,1} = 1$, $w^{(1)}_{2,2} = 1$ $w^{(1)}_{3,0} = 1$, $w^{(1)}_{3,1} = 1$, $w^{(1)}_{3,2} = -2$ Output layer: $w^{(2)}_{1,0} = 2$, $w^{(2)}_{1,1} = -1$, $w^{(2)}_{1,2} = 1$, $w^{(2)}_{1,3} = 1$ What will be the output for the NN on the input (x1, x2) = (1, 1)? Select the value which is closets to your answer.

(a) 0.5
(b) 0.1

(c) 0.6
(d) 0.9

Answer: D

Sum activiations with the σ. $\sigma(a) = \frac{1}{1+e^{-a}}$, with $a = w^T x$. Calculated the output of the hidden neurons and used those outputs as inputs for the calculating of the output of the output neuron.

$H_1 : \sigma(-5x_0 + 1x_1 + 2x_2) = \sigma(-5(1) + 1(1) + 2(1)) = \sigma(-2) = 0.119$ Repeat this for H_2 and H_3. The output of $H_2 = 0.269$, the output of $H_3 = 0.5$. $Outputneuron : \sigma(2(1) + -1(0.119) + 1(0.269) + 1(0.5)) = \sigma(2.65) = 0.93$

5 Once again consider the NN of the previous question. Assume that for a given input the output is 2/3 and the target output is 1. Moreover assume that one applies stochastic gradient descent and the error function is given by:

$$\frac{1}{2}(y-t)^2$$

What will be the δ for the output neuron?

(a) 2/27
(b) 1/3
(c) -2/9
(d) -2/27

Answer: D

2/3 is the output, hence the output is $y = \sigma(a) = \frac{2}{3}$. δ is given as the derivative of the error function. Using the chain rule we derive that the derivative is $(y-t)y' = (\sigma(a) - t)\sigma(a)(1 - \sigma(a))$, with $\sigma(a) = 2/3$, this gives us $-2/27$

6 Once again consider the above NN. Assume that the error δ of the output neuron is 0.2 and that the output of the hidden neuron 1 is 0.4, the output of the hidden neuron 2 is 0.6 and the output of hidden neuron 3 is 0.1. What will be the delta ($\delta_2^{(1)}$) of the hidden neuron 2?

(a) 0.048
(b) -0.048
(c) 0.2
(d) 0.002

Answer A

$$\delta_2^{(1)} = h'(a_2) * w_{1,2} * \delta_o uptut$$

δ is given, the weight is from the earlier NN, which is 1. Since it is given that $\sigma(a_2) = 0.6$ and $h'(a_2) = \sigma(a_2)(1 - \sigma(a_2)) = 0.6(1 - 0.6)$, we use this to fill it in the above formula: $\delta_2^{(1)} = 0.6 * 0.4 * 1 * 0.2 = 0.048$

7 Once again consider the same situation as in the question above (the error δ of the output neuron is 0.2 and that the output of the hidden neuron 1 is 0.4, the output of the hidden neuron 2 is 0.6 and the output of hidden neuron 3 is 0.1), but now we assume that the NN shares the following weights:

$$w_{1,1}^{(1)} = w_{3,2}^{(1)}$$

, meaning these two variables are identical. What will be the adaptation dw to the weight $w_{1,1}^{(1)}$ if we apply a learning rate of 1? Assume that the input $(x1, x2) = (1, 1)$.

(a) -0.048
(b) 0.018
(c) 0.03
(d) None of the above

Answer: C
First calculate the two deltas: $\delta_1^{(1)}$ and $\delta_3^{(1)}$.

$$dw = -\eta * \sum_n x_n * \delta_n$$

We calculate $\delta_1^{(1)}$ and $\delta_3^{(1)}$ in the same way as we calculated $\delta_2^{(1)}$ in Q6:
$\delta_j^{(1)} = outputH_j * (1 - outputH_j) * w_{output,hidden} * output$
$\delta_1^{(1)} = 0.4 * 0.6 * -1 * 0.2 = -0.048$
$\delta_3^{(1)} = 0.1 * 0.9 * 1 * 0.2 = 0.018$
When filling in these numbers in the formula for dw we get: $-0.048 * 1 + 0.018 * 1 = 0.03$

8 For marketing purposes a retailer wants to distinguish between costumers younger than 35 (class Y) and customers older than 35 (class O). The following table summarizes the data set in the data base of the retailer in an abstract form. The relevant attributes, determined by domain knowledge, are for convenience denoted by A with values a1, a2 and a3, B with values b1 and b2, C with values c1 and c2 and D with values d1 and d2. What is the posterior probability P(Y —(a1, b2, c1, d2)) if the retailer assumes a multinomial distribution for the likelihoods? Choose the alternative which is closest to your answer.

(a) 2/3
(b) 2/15
(c) 3/75
(d) 6/10
Answer: A

$$p(y|a_1, b_2, c_1, d_2) = \frac{p(a_1, b_2, c_1, d_2|y) * p(y)}{p(x)}$$

and just count:
$p(C|x) = \frac{(8/60)(60/100)}{((8/60)(60/100)+(4/40)(40/100)} = 2/3$

9 Consider the above dataset. What is the entropy of this dataset with respect to the class labels Y and O?

(a) 0.92
(b) 0.08
(c) 0.33
(d) 0.67
Answer: A acoording to template, 0.97 according to me
Entropy $= -\sum p(x_i) * log_2(p(x_i))$, fill in the probabilities. Use the table given (the table gives the $-p(x_i) * log_2(p(x_i))$
Entropy $= -0.6 * log_2(0.6) - (0.4) * log_2(0.4)$

10 Consider the above dataset. What will be the gain for attribute (feature) C if one uses the classification error as heuristic?

(a) 0.00
(b) 0.33
(c) 0.26

(d) 0.07
Answer: D
No clue. I did: Divide in c1 and c2. Count instances in both classes (Y and O). $Ec_1 = 1 - max(\frac{Y}{Y+O}, \frac{O}{Y+O})$ where only instance split by that feature are counted. The same is done for Ec2. Then the result is $E_{original}-$percentage in C1$*Ec_1-$percentage in C2$*Ec_2$.

$E_overall = 1 - max[60/100, 40/100] = 40/100$
$E_C1 = 1 - max[50/64, 14/64] = 14/64$
$E_C2 = 1 - max[10/36, 26/36] = 10/36$
$Gain = 40/100 - (64/100 * 14/64 + 36/100 * 10/36) = 0.16$
This results in 0.16, what obviously not correct is.

11 Consider the following part of the decision tree, with two leaf nodes and one parent node which splits on attribute A. The notation P:x N:y means that the node has x positive examples and y negative examples.

(a) $\hat{n1} = 30$ and $\hat{p2} = 25$
(b) $\hat{n1} = 20$ and $\hat{p2} = 10$
(c) $\hat{n1} = 30$ and $\hat{p2} = 50$
(d) $\hat{n1} = 20$ and $\hat{p2} = 40$
answer: D
$\hat{n1} = n * \frac{n1+p1}{n+p}$, change the varibles accordingly and just fill it in.

12 What is the accuracy of this classifier?

(a) 140/466
(b) 110/148+140/158+130/160
(c) 380/466
(d) 110/145+140/166+130/155
Answer: C
Sum the diagonals and divide by total aka $\frac{correct}{all}$

13 Once again consider the confusion matrix of the previous question. What is the recall for class C2?

(a) 140/158
(b) 140/166
(c) 140/26
(d) 140/28
Answer: B recall $= \frac{tp}{allintrueC2}$

14 Which of the following terms describes the ratio between correctly classified instances in class C and all instances classified as class C?

(a) Accuracy
(b) Precision
(c) Recall
(d) Entropy
Answer: B
Just take a look at the formulae and read carefully. It could be helpful to draw in a confusion matrix what they mean exactly.

15 Consider a two class classification problem for which we apply a probabilistic approach. The loss matrix for this classification problem is given by: Assume that we apply a classification rule of the form: if $P(C2|x) > \Theta$ then x is classified as C2. What is the optimal value for Θ given the loss matrix above?

(a) 0.33
(b) 0.67
(c) 0.50
(d) 0.20
Answer: A
Since loss matrix is 0,2,1,0. The answer is 1/3. You actually prefer the classify something as FN, rather than FP. Because it worse if you do not do so. I think it is 1/(2+1) for the calculation. I am not sure about the maths here, but just follow logic.

16 16. Consider the following visualization of a two dimensional dataset.

(a) $(1,1)^T$
(b) $(-1,1)^T$
(c) $(1,0)^T$
(d) $(1,0)^T$
Answer: Not given, also one of the answers should be $(0,1)^T$
The axis with the greatest variance I think. What does the transpose exactly do? Does it change the direction of the lines?

17 Consider a 5 dimensional dataset on which one applies PCA. The covariance matrix corresponding to the PCA's has the the following elements on the diagonal 12.0, 8.2, 4.0, 1.1, 0.1. How much variance is explained by the first two PCAs?

(a) 12.0%
(b) 20.2%
(c) 52.8%
(d) 74.7%
Answer: D
I thought $\frac{12.0+8.2}{12.0+8.2+4.0+1.1+0.1}$, this gives 79.5% however. So not really a clue how to do it.

18 Support Vector Machines find a "sparse solution" to the classification problem: they express the discriminant as a function of a subset of the training examples. This means that:

(a) Only a subset of the training data needs to be stored and compared to at testing, making actual classifications faster.
(b) Training is very fast since not all training datapoints are considered
(c) Training is slow since we need to find the best set of training datapoints that
together result in the largest margin, but the model is less likely to overfit because it is kept as simple as possible, but speed at test time is unaffected
(d) Both training and testing are faster because fewer datapoints are used
Answer: C
All training points are considered with support vector machines in order the create the decision boundary. Hence it is not a, we compare to the decision boundary; not the data points. B is not true, we actually take a look at all data points, similarly C.

19 Let D be the set of training data points and Θ the model parameters. Which of the following quantities will Maximum a Posteriori (MAP) learning maximise?

(a) $p(D, \Theta)$
(b) $p(D|\Theta)$
(c) $p(\Theta|D)$
(d) $p(D)$
(e) $p(\Theta)$
Answer: C

It optimizes the parameters given the datapoints, you can hardly optimize the datapoints nor just the parameters.

20 You try to apply some classification technique to a dataset, and you observe that the error on the training set is low, but the error on the test set is high. Why is that?

(a) Your method finds a solution that is too simple compared to the actual optimal solution to the problem
(b) Your method finds a solution that is too complicated compared to the actual optimal solution to the problem
(c) Your method gets stuck in a bad local optimum
(d) Any of the above could be the reason
Answer: B
classic example of overfitting.

21 You try to apply some classification technique to a dataset, and you get a bad performance on the training set, but a good performance on the test set. What is happening here?

(a) You are overfitting your training set
(b) Your technique is not well suited for this problem
(c) Your training set is way too small
(d) Your test set is way too small
Answer: D
performance on test is generally worse, otherwise you have not enough test data.

22 Batch gradient descent (BGD) optimises a function on a training dataset, by following the gradient as computed on the complete training set. As a consequence:

(a) BGD is guaranteed to find the optimal solution for the function
(b) BGD can get stuck in local optima, while stochastic gradient descent (SGD) can avoid these more easily because not every step is guaranteed to improve the solution.
(c) BGD is faster than SGD, because it actually computes the correct gradient and therefore requires fewer iterations to converge
(d) Iterations of SGD are faster than iterations of BGD, so that SGD is guaranteed to converge faster
Answer: B
The steps make sure that you step ”back” sometimes.

23 Maximise $1 - x_1^2 - 2x_2^2$, subject to the constraint that $x_1 + x_2 = 1$. The solution is:

(a) $x1 = \frac{1}{3}$, $x2 = \frac{2}{3}$
(b) $x1 = -\frac{1}{3}$, $x2 = \frac{2}{3}$
(c) $x1 = \frac{1}{3}$, $x2 = -\frac{2}{3}$
(d) $x1 = -\frac{1}{3}$, $x2 = -\frac{2}{3}$
Answer: A
Although I have x_1 and x_2 switched around.

$$f(x_1, x_2) = 1 - x_1^2 - 2x_2^2$$

$$g(x_1, x_2) = x_1 + x_2 - 1 = 0$$

$$L(x_1, x_2, \lambda) = f(x_1, x_2) + \lambda * g(x_1, x_2)$$

$$= 1 - x_1^2 - 2x_2^2 + \lambda(x_1 + x_2 - 1)$$

Taking the derivatives over all three variables and equal them to 0:

$$-2x_1 + \lambda = 0$$

$$-4x_2 + \lambda = 0$$

$$x_1 + x_2 - 1 = 0$$

rewrite into:

$$x_1 = \lambda/2$$

$$x_2 = \lambda/4$$

fill in into $x_1 + x_2 - 1 = 0$ Solve for λ Solve for x_1 and x_2

24 Mixture of Density Networks (MDN) are a way of dealing with non-Gaussian noise on the target values of a neural network. Which of the following statements is correct:

(a) MDN do not predict the target value for a given input
(b) MDN predict a distribution over outputs for a given output in the form of a Gaussian Mixture Model
(c) MDN are not suitable for "inverse problems"
(d) The number of mixture elements of an MDN must always be two.
Answer: B
I never heard of MDNs

25 When using k-Nearest-Neighbours:

(a) The actual value of k does not matter much, as long as it is odd
(b) Increasing the value of k reduces the risk of overfitting, but may lead to oversmoothing
(c) It is impossible to have region in the space where the only strategy is to take a random guess, as long as k is odd
(d) All of the above are correct
answer: B

26 Which of the following statements is true? The "dual representation" of a Support Vector Machine (SVM)

(a) Requires us to define a "kernel function", but allows us to create non-linear discriminants
(b) Expresses the weights of the SVM in terms of the training datapoints
(c) Allows us to use a Lagrangian to indicate which training points are "support vectors", and is therefore much faster to compute than the "primal representation"
(d) Allows us to find a sparse solution for the problem
(e) Projects the training data to a high-dimensional "kernel space"
answer: B
Would not have guessed this myself

27 The "regularisation" of a model reduces the risk of overfitting by:

(a) Reducing the number of free parameters, making the model simpler
(b) Adding a penalty models with many weights, favouring simpler models
(c) Adding a penalty to large weight values, because large weights tend to result in overfitting
(d) Constraining different weights to have the same value, to make the model more "regular"
Answer: C
large weights are an indication of overfitting, hence penalizing large weights reduces overfitting.

28 Which of the following statements is true? Generative and discriminative models are probabilistic models, where:

(a) Generative models model the joint probability of data and target, which allow them to compute the probability of the target even if information is missing

(b) Generative models generate the correct target values by generating new data points
(c) Discriminative models do not model the probability of the targets, but only the probability of the input data
(d) Discriminative models only model the value the targets given the input data, not their distribution
Answer: A
A generative algorithm models how the data was generated in order to categorize a signal. It asks the question: based on my generation assumptions, which category is most likely to generate this signal?

A discriminative algorithm does not care about how the data was generated, it simply categorizes a given signal.