Profs. Martin Jaggi and Rüdiger Urbanke
Machine Learning – CS-433 - IN
Wednesday 15.01.2020
from 16h15 to 19h15 in STCC08328
Duration : 180 minutes

1

Heidi

SCIPER : **837495**

Do not turn the page before the start of the exam. This document is double-sided, has 16 pages. Do not unstaple.

- This is a closed book exam. No electronic devices of any kind.
- Place on your desk: your student ID, writing utensils, one double-sided A4 page cheat sheet (hand-written or 11pt min font size) if you have one; place all other personal items below your desk.
- You each have a different exam.
- Only answers in this booklet count. No extra loose answer sheets. Use loose sheets as scrap paper.
- For the **multiple choice** questions, we give :
  - $+2$ points if your answer is correct,
  - 0 points if you give no answer or more than one,
  - -0.5 points if your answer is incorrect.
- For the **true/false** questions, we give :
  - $+1$ points if your answer is correct,
  - 0 points if you give no answer or more than one,
  - -1 points if your answer is incorrect.
- Use a **black or dark blue ballpen** and clearly erase with **correction fluid** if necessary.
- If a question turns out to be wrong or ambiguous, we may decide to nullify it.

Respectez les consignes suivantes | Observe this guidelines | Beachten Sie bitte die unten stehenden Richtlinien

choisir une réponse | select an answer Antwort auswählen	ne PAS choisir une réponse | NOT select an answer NICHT Antwort auswählen	Corriger une réponse | Correct an answer Antwort korrigieren
☒ ☑ ■	□	□ □

ce qu'il ne faut **PAS** faire | what should **NOT** be done | was man **NICHT** tun sollte

▨ ☒ ○ ⊡ ▨ □

First part: multiple choice questions

For each question, mark the box corresponding to the correct answer. Each question has **exactly one** correct answer.

Shrinking Confidence

Assume that you get a confidence interval of size δ for some problem given N iid samples.

Question 1 Expressed as a function of N, how many iid samples do you need to get a confidence interval of size $\delta/3$?

- □ $3N$
- □ $N/3$
- □ N^3
- ■ $9N$
- □ $\sqrt{3N}$
- □ e^{3N}

Solution: The size of the confidence interval decays as $1/\sqrt{N}$. We need therefore 9 times as many samples, i.e., $9N$ samples.

Family Expectations

For $\theta > 0$, consider the following probability distribution defined for $y \geq 0$,

$$p(y) = e^{(-y^3)\theta - A(\theta)}, \quad A(\theta) = -\frac{1}{3}\log(\theta) + c,$$

where c is an appropriate constant.

Question 2 What is $\mathbb{E}_{Y\sim p}[Y^3]$, expressed as a function of θ?

- □ θ^3
- □ y
- □ $A(\theta)$
- □ c
- □ 1
- ■ $\frac{1}{3\theta}$

Solution: We know that $A'(\theta) = \mathbb{E}_{Y\sim p}[\phi(Y)]$. For our case $\phi(\theta) = -y^3$. Hence $\frac{1}{3\theta} = -A'(\theta) = \mathbb{E}_{Y\sim p}[-\phi(Y)] = \mathbb{E}_{Y\sim p}[Y^3]$. Therefore the correct answer is $\frac{1}{3\theta}$.

SVMs versus Logistic Regression

Consider a classification problem using either SVMs or logistic regression and separable data. For logistic regression we use a small regularization term (penality on weights) in order to make the optimum well-defined. Consider a point that is correctly classified and distant from the decision boundary. Assume that we move this point slightly.

Question 3 What will happen to the decision boundary?

□ Small change for SVMs and small change for logistic regression.

□ No change for SVMs and large change for logistic regression.

□ No change for SVMs and no change for logistic regression.

■ No change for SVMs and a small change for logistic regression.

□ Large change for SVMs and large change for logistic regression.

□ Large change for SVMs and no change for logistic regression.

□ Small change for SVMs and no change for logistic regression.

□ Small change for SVMs and large change for logistic regression.

□ Large change for SVMs and small change for logistic regression.

Solution: The hinge loss used by SVMs gives zero weight to such a point while the log-loss used by logistic regression gives a small but non-zero weight to these points. Therefore no change for SVMS and a small change for logistic regression.

KNN Classifier

You are in D-dimensional space and use a KNN classifier with $k = 1$. You are given N samples and by running experiments you see that for most random inputs $\mathbf{x}$ you find a nearest sample at distance roughly δ.

Question 4 You would like to decrease this distance to $\delta/2$. How many samples will you likely need? Give an educated guess.

■ $2^D N$

□ N^D

□ $2D$

□ $\log(D)N$

□ N^2

□ D^2

□ $2N$

□ DN

Solution: We assume that the data is indeed distributed in $\mathbb{R}^D$ and not in a lower-dimensional subspace. Imagine balls of radius δ around the samples. We know that these balls more or less just cover the space since by assumption most instances are covered by such a ball but not by balls that are much smaller. If we want to cover the same space with balls of radius $\delta/2$ then we likely will need of the order $2^D N$ samples, i.e., 2^D as many samples. This is true since the volume of a ball of radius r scales like r^D.

Tricky Question

You are given samples $\mathcal{S} = \{\mathbf{x}_n\}_{n=1}^N$, where each sample has two components, i.e., $\mathbf{x} = (x_1, x_2)$. You compute from this the corresponding kernel matrix $\mathbf{K}$ with entries $\mathbf{K}_{i,j} = \mathbf{x}_i^T \mathbf{x}_j$.

Assume now that you transform the feature vector to $\tilde{\mathbf{x}} = (x_1^2, x_2^2, \sqrt{2}x_1x_2)$ and compute from this new feature vector the corresponding kernel matrix $\tilde{\mathbf{K}}$ with entries $\tilde{\mathbf{K}}_{i,j} = \tilde{\mathbf{x}}_i^T \tilde{\mathbf{x}}_j$.

Question 5 What function does this transform correspond to? I.e., what function $f(\cdot)$ can you pick so that $\tilde{\mathbf{K}} = f(\mathbf{K})$, where the function $f(\cdot)$ is applied component-wise?

□ $f(z) = e^z$

■ $f(z) = z^2$

□ $f(z) = 1$

□ $f(z) = z^3$

□ $f(z) = \log(z)$

□ $f(z) = z$

□ $f(z) = \sqrt{2}z$

Solution: The correct answer is $f(z) = z^2$ since $f(\mathbf{x}_i^T \mathbf{x}_j) = f((\mathbf{x}_i)_1(\mathbf{x}_j)_1 + (\mathbf{x}_i)_2(\mathbf{x}_j)_2)^2 = \tilde{\mathbf{x}}_i^T \tilde{\mathbf{x}}_j$.

Houston we have an Overflow Problem

Assume that you want to implement the sigmoid function: $\sigma(x) = e^x/(e^x+1)$. You know that your computer can handle numbers with very small absolute value but might overflow when dealing with numbers that have a very large absolute value. Let $f_1(x) = e^x/(e^x+1)$ and $f_2(x) = 1/(e^{-x}+1)$.

Question 6 Which of the following implementations is best?

□ $f_1(x)$

□ $f_2(x)$

□ $f_1(x)$ if $x > 0$ and $f_2(x)$ otherwise

■ $f_2(x)$ if $x > 0$ and $f_1(x)$ otherwise

Solution: If we choose $f_2(x)$ if $x > 0$ and $f_1(x)$ otherwise then we avoid overflows.

K-means Clustering

Consider K-means clustering in D-dimensional real space and assume that K is known. We have N samples. We have seen in class that this corresponds to solving the following optimization problem:

$$\begin{aligned} \min_{\mathbf{z},\boldsymbol{\mu}} \ \mathcal{L}(\mathbf{z},\boldsymbol{\mu}) &= \sum_{n=1}^{N}\sum_{k=1}^{K} z_{nk}\|\mathbf{x}_n - \boldsymbol{\mu}_k\|_2^2 \\ \text{where } \mathbf{z}_n &= [z_{n1}, z_{n2}, \ldots, z_{nK}]^\top \\ \mathbf{z} &= [\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_N]^\top \\ \boldsymbol{\mu} &= [\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \ldots, \boldsymbol{\mu}_K]^\top \end{aligned}$$

Question 7 What extra conditions do we need to render this formulation correct?

□ s.t. $\boldsymbol{\mu}_k \in \mathbb{R}^N$, $z_{nk} \in [0,1]$, $\sum_{n=1}^{N} z_{nk} = 1$

■ s.t. $\boldsymbol{\mu}_k \in \mathbb{R}^D$, $z_{nk} \in \{0,1\}$, $\sum_{k=1}^{K} z_{nk} = 1$

□ s.t. $\boldsymbol{\mu}_k \in \mathbb{R}^N$, $z_{nk} \in \{0,1\}$, $\sum_{n=1}^{N} z_{nk} = 0$

□ s.t. $\boldsymbol{\mu}_k \in \mathbb{R}^D$, $z_{nk} \in \{-1,1\}$, $\sum_{k=1}^{K} z_{nk} = 0$

□ s.t. $z_{nk} \in \mathbb{R}^D$, $\boldsymbol{\mu}_k \in \{-1,1\}$, $\sum_{k=1}^{K} z_{nk} = 1$

□ s.t. $z_{nk} \in \mathbb{R}^D$, $\boldsymbol{\mu}_k \in \{0,1\}$, $\sum_{k=1}^{K} z_{nk} = 1$

□ s.t. $z_{nk} \in \mathbb{R}^D$, $\boldsymbol{\mu}_k \in \{0,1\}$, $\sum_{k=1}^{K} \boldsymbol{\mu} = 1$

□ s.t. $z_{nk} \in \mathbb{R}^K$, $\boldsymbol{\mu}_k \in \{-1,1\}$, $\sum_{k=1}^{K} z_{nk} = 0$

□ s.t. $\boldsymbol{\mu}_k \in \mathbb{R}^K$, $z_{nk} \in [0,1]$, $\sum_{n=1}^{N} z_{nk} = 0$

Solution: The correct conditions are: s.t. $\boldsymbol{\mu}_k \in \mathbb{R}^D$, $z_{nk} \in \{0,1\}$, $\sum_{k=1}^{K} z_{nk} = 1$. In more detail, the optimization is over the K centers $\boldsymbol{\mu}_k$, each being a vector in $\mathbb{R}^D$ and the “membership” functions z_{nk}; the latter are elements of $\{0,1\}$ and for each n exactly one of the K has to be 1 and the rest 0.

Finding Adversarial Examples

Consider a binary classification problem with classifier $f(\mathbf{x})$ given by

$$f(\mathbf{x}) = \begin{cases} 1, & g(\mathbf{x}) \geq 0, \\ -1, & g(\mathbf{x}) < 0, \end{cases}$$

and $\mathbf{x} \in \mathbb{R}^6$. Consider a specific pair $(\mathbf{x}, y = 1)$ and assume that $g(\mathbf{x}) = 8$. In particular this means that this point is classified correctly by f. Assume further that we have computed the gradient of g at $\mathbf{x}$ to be $\nabla_{\mathbf{x}} g(\mathbf{x}) = (+1, -2, +3, -4, +5, -6)$. You are allowed to make one step in order to (hopefully) find an adversarial example. In the following four questions, assume $\epsilon = 1$.

Question 8 Which offset δ with $\|\delta\|_1 \leq 1$ yields the smallest value for $g(\mathbf{x} + \delta)$, assuming that g is (locally) linear?

☒ $(0, 0, 0, 0, 0, 1)$

□ $(+1, -1, +1, -1, +1, -1)$

□ $(+1, -2, +3, -4, +5, -6)$

□ $(+1, +1, +1, +1, +1, +1)$

□ $(-1, +2, -3, +4, -5, +6)$

□ $-(0, 0, 0, 0, 0, 1)$

□ $(-1, +1, -1, +1, -1, +1)$

□ $(-1, -1, -1, -1, -1, -1)$

Solution: We want to find an offset δ so that $\|\delta\|_1 \leq 1$ and so that $g(\mathbf{x} + \delta)$ is as small as possible (and in particular < 0). We know that to first order

$$g(\mathbf{x} + \delta) = g(\mathbf{x}) + \delta^T \nabla_{\mathbf{x}} g(\mathbf{x}).$$

Hence, we want to find a vector δ of ℓ_1 norm 1 that minimizes $\delta^T \nabla_{\mathbf{x}} g(\mathbf{x})$. The solution is $\delta = (0, 0, 0, 0, 0, 1)$. I.e., we pick 1 in the component where $\nabla_{\mathbf{x}} g(\mathbf{x})$ has the entry of maximal modulus and we pick the sign so that the inner product is negative.

Question 9 What is the value of $g(\mathbf{x} + \delta)$ for this ℓ_1-optimal choice assuming that g is (locally) linear?

□ $+13$

□ -4

□ -5

□ -7

☒ 2

□ 4

□ -13

□ -2

□ $+7$

□ 0

Solution: According to the first order approximation $g(\mathbf{x} + \delta) = 8 - 6 = 2 > 0$.

Question 10 Which offset δ with $\|\delta\|_\infty \leq 1$ yields the smallest value for $g(\mathbf{x}+\delta)$, assuming that g is (locally) linear?

□ $(+1, -2, +3, -4, +5, -6)$
□ $-(0, 0, 0, 0, 0, 1)$
□ $(0, 0, 0, 0, 0, 1)$
□ $(-1, -1, -1, -1, -1, -1)$
□ $(+1, +1, +1, +1, +1, +1)$
■ $(-1, +1, -1, +1, -1, +1)$
□ $(+1, -1, +1, -1, +1, -1)$
□ $(-1, +2, -3, +4, -5, +6)$

Solution: we want to find a vector δ of ℓ_∞ norm 1 that minimizes $\delta^T \nabla_{\mathbf{x}} g(\mathbf{x})$. The solution is $\delta = (-1, 1, -1, 1, -1, 1)$. I.e., we walk in each direction by 1 and choose the sign so that the function decreases.

Question 11 What is the value of $g(\mathbf{x}+\delta)$ for this ℓ_∞-optimal choice assuming that g is (locally) linear?

□ -5
□ -2
□ -7
□ $+7$
□ 4
□ 0
□ $+13$
□ 2
□ -4
■ -13

Solution: According to the first order approximation $g(\hat{\mathbf{x}}) = 8 - 21 = -13 < 0$.

Second part: true/false questions

For each question, mark the box (without erasing) TRUE if the statement is **always true** and the box FALSE if it is **not always true** (i.e., it is sometimes false).

Question 12 (Bayes Nets) We are given a Bayes net involving the variables $X_1, \cdots, X_n$. We determine, using our standard rules, that $X_1 \perp X_2 \mid X_3$.
Assume now that you delete some edges in the original Bayes net. For the modified Bayes net, is it *always* true that $X_1 \perp X_2 \mid X_3$?

■ TRUE □ FALSE

Solution: True. Our rules require that every path from X_1 to X_2 is blocked by X_3. By deleting edges we have potentially fewer paths but any path that survives will still be blocked.

Question 13 (Nearest Neighbor) The training error of the 1-nearest neighbor classifier is zero.

■ TRUE □ FALSE

Solution: True. For each element in the training set we will output it's own label.

Question 14 (Backpropagation) Training via the backpropagation algorithm always learns a globally optimal neural network if there is only one hidden layer and we run an infinite number of iterations and decrease the step size appropriately over time.

□ TRUE ■ FALSE

Solution: False. Backpropagation computes the gradient and hence this is a greedy first-order algorithm. The problem is in general non-convex. Hence in general there is no reason that a globally optimal solution is reached.

Question 15 (Infinite Data) Assume that your training data $\mathcal{S} = \{(\mathbf{x}_n, y_n)\}$ is iid and comes from a fixed distribution $\mathcal{D}$ that is unknown but is known to have bounded support. Assume that your family of models contains a finite number of elements and that you choose the best such element according to the training data. You then evaluate the risk for this chosen model. Call this the training risk. As $|\mathcal{S}|$ tends to infinity, this training risk converges to the true (according to the distribution $\mathcal{D}$) risk of the best model in this family.

■ TRUE □ FALSE

Solution: True. As the number of samples increases, for every given model the empirical risk converges to the true risk. By assumption the family only has a finite number of elements. Hence this convergence is uniform. Choosing the model hence according to the empirical risk is equivalent to choosing according to the true risk. Hence, the training risk will converge to the true risk of the best model in this finite family of models.

Question 16 (Robustness) The l_1 loss is less sensitive to outliers than l_2.

■ TRUE □ FALSE

Solution: True.

Question 17 (Convex I) Unions of convex sets are convex.

□ TRUE ■ FALSE

Solution: False. Look at a line and take two non-intersecting closed intervals. Each of them is convex but the union is not.

Question 18 (Convex II) Intersections of convex sets are convex.

■ TRUE □ FALSE

Solution: True. Look at two points that are in the intersection of all sets. Then for each set these two points are in there and, since all sets are assumed to be convex, all points on the connecting line are in each set. Hence these points are in the intersection of all such sets.

Question 19 (Convex III) Let $f, g : \mathbb{R} \to \mathbb{R}$ be two convex functions. Then $h = f \circ g$ is always convex.

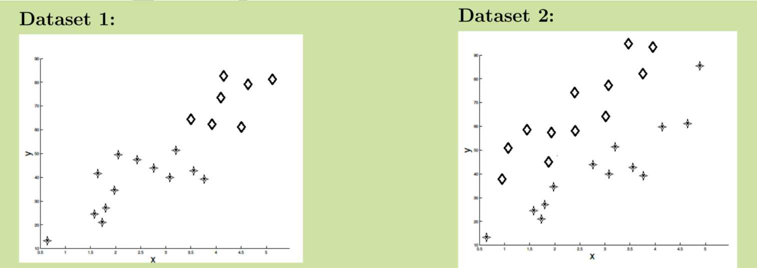
□ TRUE ■ FALSE

Solution: False. The second derivative is $f''(g)(g')^2 + f'(g)g''$. If $f' < 0$ and very large then we can make the whole expression negative.

Question 20 (Maximum Likelihood) Assume that $X \in \{0, 1\}$ and that $p(X = 0) = \frac{1}{3}$. Assume further that $Y = X + Z$ where Z is a zero-mean Gaussian noise of variance 1. We observe Y and are asked to guess X. The maximum likelihood estimator $\hat{X}(Y) = \text{argmax}_{x \in \{0,1\}} p(Y = y|X = x)$ minimizes the probability of error.

□ TRUE ■ FALSE

Solution: False. The estimator that minimizes the probability of error is the *maximum a posteriori* estimator which is given by $\hat{X}(Y) = \max_{x \in \{0,1\}} p(X = x)p(Y = y|X = x)$.



Question 21 (Due to Matt Gormley) (PCA) Consider the two datasets given in the figure. Assume that you first project the points into the first principal component and then you use a threshold function to classify the data. This approach works better for Dataset 2 than for Dataset 1.

□ TRUE ■ FALSE

Solution: False. As you can see from the figures in both datasets the first principal component is along the fourty-five degree direction. But only for the first dataset is this a good discriminator.

Question 22 (Stochastic Gradient Descent) One iteration of standard SGD for SVM, logistic regression and ridge regression costs roughly $\mathcal{O}(D)$, where D is the dimension of a data point.

■ TRUE □ FALSE

Solution: True.

Question 23 (Stochastic Gradient Descent, cont) SGD in typical machine learning problems requires fewer parameter updates to converge than full gradient descent.

□ TRUE ■ FALSE

Solution: False. To the contrary we typically require more parameter updates but each iteration is considerably cheaper.

Question 24 (SGD & Matrix Factorization) For optimizing a matrix factorization problem in the recommender systems setting, as the number of observed entries increases, the computational cost of full gradient steps increases, while the cost of an SGD step remains the same.

■ TRUE □ FALSE

Solution: True.

Question 25 (Alternating Least Squares & Matrix Factorization) For optimizing a matrix factorization problem in the recommender systems setting, as the number of observed entries increases but all K, N, D are kept constant, the computational cost of the matrix inversion in Alternating Least-Squares increases.

□ TRUE ■ FALSE

Solution: False. The matrix to be inverted is always of size $K \times K$.

Question 26 (Text Representation Learning, GloVe) Learning GloVe word vectors is identical to approximating the observed entries of the word/context co-occurence counts by $\mathbf{W}\mathbf{Z}^\top$, in the least square sense, if the f_{dn} weights are set to 1 for all observed entries.

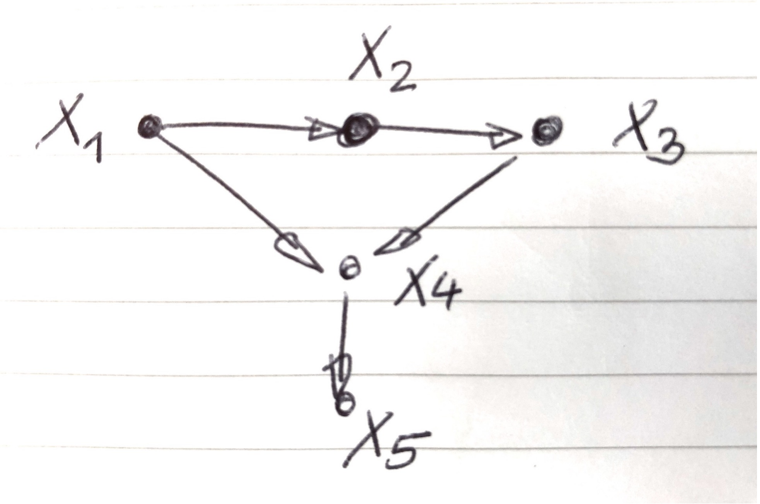
□ TRUE ■ FALSE

Solution: False. It's the log of the co-occurence counts, sorry.

Question 27 (Text Representation Learning, word2vec) An SGD step for learning GloVe word vectors is computationally equally expensive to an SGD step in word2vec, however only GloVe requires memory the size of the co-occurence matrix.

■ TRUE □ FALSE

Solution: True.



Third part, open questions

Answer in the space provided! Your answer must be justified with all steps. Do not cross any checkboxes, they are reserved for correction.

Independence

Consider the following joint distribution that has the factorization

$$p(x_1, x_2, x_3, x_4, x_5) = p(x_1)p(x_2|x_1)p(x_3|x_2)p(x_4|x_1, x_3)p(x_5|x_4).$$

Question 28: (*4 points.*) Determine whether the following statement is correct.

$$X_1 \perp X_3 \mid X_2, X_5$$

Show your reasoning.

□0 □1 □2 □3 ■4

Solution: The figure shows the Bayes net corresponding to this factorization. Note that the path from X_1 to X_3 via X_4 is not blocked since it is head to head and we are conditioning on X_5, and X_5 is a child of X_4. The statements is therefore in general not true.

Grading Notes: We gave one point if you correctly drew the Bayes net and full points if you answered correctly and explicitly wrote down the correct reasoning. We accepted as correct if you wrote as answer that the statement is false.

Common mistakes: Most people answered this question correctly. Some people answered that the statement is correct since at least one path is blocked.

Question 29: (*3 points.*) We say that a data point y follows a Poisson distribution with parameter θ if the probability of the observation y, $y \in \mathbb{N}$, is given by

$$p(y \mid \theta) = \frac{\theta^y e^{-\theta}}{y!}.$$

Assume that you are given the samples $\mathcal{S} = \{y_1, \cdots, y_N\}$.

(a) Write down the log-likelihood, call it $\mathcal{L}$, of these samples as a function of θ assuming that the samples are iid and follow a Poisson distribution with parameter θ.

(b) What is the parameter θ that maximizes this log-likelihood expressed as a function of the samples?

(c) Interpret the result.

□ 0 □ 1 □ 2 ■ 3

Solution:

(a) The log-likelihood is

$$\begin{aligned}\mathcal{L} &= (\sum_{n=1}^{N}(y_n \log(\theta) - \theta - \log y_n!) \\ &= \log(\theta)\sum_{n=1}^{N} y_n - N\theta - \log(\prod_{n=1}^{N} y_i!).\end{aligned}$$

(b) Taking the derivative with respect to θ, setting the result to 0 and solving for θ we get

$$\theta = \frac{1}{N}\sum_{n=1}^{N} y_n.$$

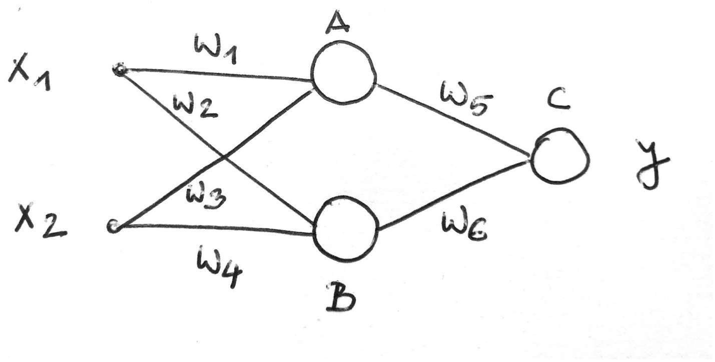
(c) The parameter θ represents the mean of the Poisson distribution and the optimum choice of θ is to set this mean to the empirical mean of the samples.

Grading Notes: We gave one point for correctly answering each of the three points.
Common Mistakes:

(a) Most people wrote down the log-likelihood correctly. Some people forgot the index and the sum.

(b) Several people forgot the $1/N$ factor in front of the sum. If you realized that there was a mistake, said so, but perhaps could not find it, you got the point. But if you did not realize you got 0 points. The reason for this is that the parameter θ is the mean of the distribution and so it makes no sense to set it to the unnormalized quantity.

(c) For the last part you only got a point if you mentioned that the θ is the mean of the Poisson distribution (and so that it is natural that you set this mean to the empirical mean) or argued that the Poisson distribution is an element of the exponential family and that θ is a sufficient statistics.



Question 30: (*4 points.*)[Question due to Eric Xing and Tom Mitchell] Consider the neural network shown in the figure. This network has one input layer, one hidden layer and one output layer. Further, it has two input features, denote them by x_1 and x_2, and one output, call it y.

Assume that we have two types of general activiation functions.

(a) $S : S(a) = \text{sign}(\frac{1}{1+e^{-a}} - \frac{1}{2})$,

(b) $L : L(a) = \gamma a$, for some scalar γ

where in both cases $a = \sum_i w_i x_i$ denotes the sum over the incoming connections to the current neuron.

(a) Assume that we assign the activation function L to all three nodes A, B, and C. What model does this network express? Explicitly express the function computed by the network in terms of $w_1, ...w_6$.

(b) Assume that we assign the activation function L to nodes A and B and the activation function S to node C. What model does this network express? Does it remind you of something you have seen in class? Explicitly express the function computed by the network in terms of $w_1, ..., w_6$.

Note that we are only interested in the resulting input-output relationship and there is no loss function involved.

□0 □1 □2 □3 ■4

Solution:

(a) The overall function corresponds to linear regression $y = \alpha_1 x_1 + \alpha_2 x_2$ with $\alpha_1 = c^2(w_1 w_5 + w_2 w_6)$ and $\alpha_2 = c^2(w_3 w_5 + w_4 w_6)$.

(b) The overall function corresponds to logistic regression regression $y = \text{argmax}_{y \in \{0,1\}} p(Y = y \mid x)$ with $p(Y = 1 \mid x) = \frac{e^{\alpha_1 x_1 + \alpha_2 x_2}}{1+e^{\alpha_1 x_1 + \alpha_2 x_2}}$ with $\alpha_1 = c(w_1 w_5 + w_2 w_6)$ and $\alpha_2 = c(w_3 w_5 + w_4 w_6)$.

Grading Notes:

a 1 point - expressing the function computed by the network correctly; 1 point - recognizing that the network expresses: linear regression/ linear model,

b 1 point - expressing the function computed by the network correctly; 1 point - recognizing that the network expresses logistic regression

Common mistakes and exceptions:

(a) Errors in the expressions constituted "small mistakes". For instance having c instead of c^2 in the expression for point a constituted one small mistake,

(b) One Small mistake was acceptable but two small mistakes resulted in decreasing the number of points given by 1,

(c) Not expressing the function explicitly resulted in one small mistake,

(d) In point b a common mistake was to say that the model expresses: classification task, binary classification, linear classifier, linear binary classifier, ... We decided not to give points for these types of answers as we were really looking for “logistic regression”.

(e) In point b in rare cases a point was also given if someone explained that what is expressed by the network can be understood as a probability of a particular class. The reason is that the essence of “LOGISTIC regression” is exactly regressing (estimating a value) of a probability of events.