CMPUT466/566 Final Exam
Fall 2020

Problem 1 [10 marks]

Consider a deep neural network with L layers of d-dimensions. Is it going to be more overfitting or underfitting if we modify the neural network as follows?

1. [3 marks] Increase the number of layers, i.e., increase L.
2. [3 marks] Increase the dimension, i.e., increase d.
3. [3 marks] Add larger ℓ_2-penalty for the weights of the neural networks.
4. [1 marks] You'll get another 1 mark If all the above answers are correct.

Write one word ("overfitting" or "underfitting") for 1--3 sub-questions. Number your answers.

1.overfitting

2. overfitting

3. underfitting

CMPUT466/566 Final Exam
Fall 2020

Problem 2 [10 marks]

Suppose we have a model about $P(\mathcal{D}|w)$, where $\mathcal{D}$ is the training data and w is the parameter. Present the general training objectives of

1. [3 marks] Maximum likelihood estimation (MLE), and
2. [3 marks] Max a posteriori (MAP) estimation.

Present

3. [4 marks] the predictive density of target t given input $\boldsymbol{x}_*$ in Bayesian learning, i.e., $p(t|\boldsymbol{x}_*)$.

Note, here we talk about a general model $P(\mathcal{D}|w)$, rather than a specific linear/logistic regression model. If your answer involves maximizing or minimizing, be clear about the optimization variable.

Hint: For an integral $\int f(x)dx$, you can write int f(x) dx, or integral f(x) dx. Be clear about your integration variable.

1. w* = argmax_w p(D|w)

2. w* = argmax_w p(D|w)p(w) where p(w) is the prior

3. Predictive density

p(t|x*) = int p(t|w, x*)p(w|D) dw

where p(w|D) is propto p(D|w)p(w)

Mark deduction (1 point) if you say p(w|D) = p(D|w)p(w)

CMPUT466/566 Final Exam
Fall 2020

Problem 3 [15 marks]

Consider a K-way classification problem $t \in \{1, \cdots K\}$ based on features x. Present the general MLE training objective for

1. [4 marks] Generative learning, and
2. [4 marks] Discriminative learning.

For inference of a classification model (either discriminative or generative), present

3. [4 marks] the max a posteriori (MAP) inference criterion, and
4. [3 marks] what is the benefit of doing MAP inference?

Note: no proof is needed.

1. argmax_w prod p(t^(m) | x^(m))

2. argmax_w prod p(t^(m), x^(m)) = argmax_w prod p(t^(m)) p(x^(m) | t^(m))

Note: Of course, if you wrote max log-likelihood, it's also correct. If you present cross-entropy loss for 2, it's okay. But if you present a specific model, it may result in mark deduction.

3. t* = argmax_t p(t|x*)

4. MAP inference <=> minimizing expected error

Mark deduction: If you thought MAP inference is MAP estimation, then it's wrong.

CMPUT466/566 Final Exam
Fall 2020

Problem 4 [10 marks]

For a linear classifier with decision boundary $\boldsymbol{w}^\top \boldsymbol{x} + b = 0$, the original training objective of a hard support vector machine (SVM) is

$$\underset{\boldsymbol{w},b}{\text{maximize}} \quad \min_{m=1\ldots M} \quad t^{(m)}(\boldsymbol{w}^\top \boldsymbol{x}^{(m)} + b)/\|\boldsymbol{w}\|$$

for target $t^{(m)} \in \{+1, -1\}$.

1. [2 marks] Give a natural language description of the intuition of the above objective.
2. [5 marks] Transform the above non-convex formulation into a convex form. State your assumptions explicitly.
3. [3 marks] Present the soft-SVM objective.

Note: Solving the SVM optimization problem is not required.

1. Find the decision boundary that maximizes the margin between the data points and the boundary.

2. minimize_{w,b} 1 / 2 * ||w||^2

subject to t(m) (w.T x(m) + b) >=1

The assumption is the minimal value of t(m) (w'x(m) + b) is 1, which is then equivalently relaxed to t(m) (w'x(m) + b) >= 1

3. minimize_{w,b,xsi} 1 / 2 * ||w||^2 + C * sum xsi(m), where sum is for m=1..M subject to t(m)(w' x(m) + b) >= 1 - xsi(m), xsi(m)>=0, for m = 1..M

CMPUT466/566 Final Exam
Fall 2020

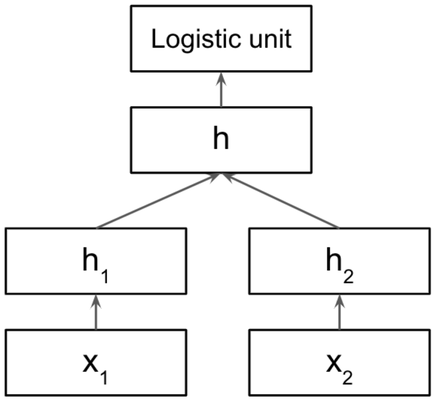
Problem 5 [10 marks]

Consider a "matching" neural network that takes two data samples x_1, x_2 as input and predicts if the two samples are similar or not (shown below).

We are particularly interested in how $\boldsymbol{h}$ is computed from $\boldsymbol{h}_1$ and $\boldsymbol{h}_2$. Suppose $\boldsymbol{h}, \boldsymbol{h}_1, \boldsymbol{h}_2 \in \mathbb{R}^d$ are all d -dimensional vectors, and $\boldsymbol{h}$ is computed by $\boldsymbol{h} = W_1\boldsymbol{h}_1 + W_2\boldsymbol{h}_2 + \boldsymbol{h}_1 \circ \boldsymbol{h}_2 + \boldsymbol{b}$, where $W_1, W_2, \boldsymbol{b}$ are parameters. The circle $\circ$ is pointwise multiplication, i.e., for vectors $\boldsymbol{z} = \boldsymbol{x} \circ \boldsymbol{y}$, we have $z_i = x_i y_i$.

1. [3 marks] Give the dimensions of $W_1, W_2, \boldsymbol{b}$, i.e., how many rows and columns?
2. [7 marks] Derive the recursion step of backpropagation from $\boldsymbol{h}$ to $\boldsymbol{h}_1$ and $\boldsymbol{h}_2$. That is to say, suppose $\frac{\partial J}{\partial h_j}$ is known for $j = 1, \cdots, d$, what is $\frac{\partial J}{\partial h_{1,i}}$ and $\frac{\partial J}{\partial h_{2,i}}$?

Here, h_j is the jth element of the $\boldsymbol{h}$ vector, and $h_{1,i}$ is the i element of the $\boldsymbol{h}_1$ vector.
Note: we ask for a point-wise representation. No matrix calculus is needed.
Typing suggestions: Use the notations like h1[i], W1[i,j], dJ_dh1[i], and dJ_dh[j].



1. W1, W2: d x d matrix, b: d-dimensional vector.

2. dJ_dh1[i] = sum_j dJ_dh[j] dh[j]_dh1[i] = [sum_j dJ_dh[j] * W1[j,i]] + dJ_dh[i] * h2[i]

Likewise, dJ_dh2[i] = sum_j dJ_dh[j] * W2[j,i] + dJ_dh[i] * h1[i]

Mark deduction (2 pts): dealing with matrix/vector as if it's a scalar calculus is wrong dJ_dh1[i] = sum_j dJ_dh * dh_dh1[i]
At least, you need an inner-product here.

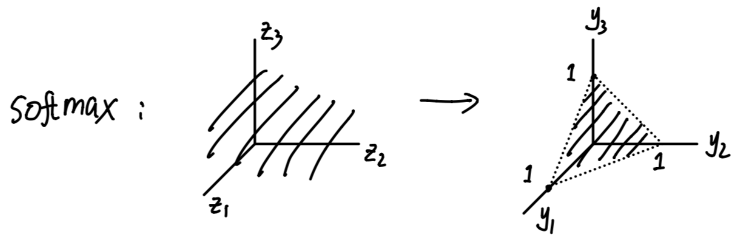
Mark deduction: Treating W as a scalar results in mark deduction. First, a scalar is usually represented by a lower case letter. Second, nobody multiplies a layer with a scalar parameter in neural networks. So it's unambiguous.

CMPUT466/566 Final Exam
Fall 2020

Problem 6 [10 marks]

Consider a K-way classification problem $t \in \{1, \cdots, K\}$ based on features $\boldsymbol{x}$.

1. [3 marks] Assume our model is $\boldsymbol{y} = \mathrm{softmax}(W\boldsymbol{x} + \boldsymbol{b})$. Write out the softmax function. That is to say, how do you compute y_i, which is an element of the $\boldsymbol{y}$ vector?
2. [3 marks] Suppose the groundtruth target category is represented as a one-hot vector $\boldsymbol{t} = (t_1, \cdots, t_K)$. Write out the cross-entropy loss of a data sample. The superscript (m) can be omitted.
3. [4 marks] We know softmax is actually a mapping from a vector $\boldsymbol{z} \in \mathbb{R}^K$ to a probability distribution $\boldsymbol{y} \in \Delta_K$, where Δ_K is called a simplex. The below figure shows an example in the 3-D case. The range/image of softmax is the simplex excluding the boundary, i.e., for any point $\boldsymbol{y}$ in the dashed triangle, there exists a $\boldsymbol{z} \in \mathbb{R}^K$ such that $\boldsymbol{y} = \mathrm{softmax}(\boldsymbol{z})$.



Now suppose in implementation, we had a double-softmax bug, i.e., we mistakenly implement $\boldsymbol{y} = \mathrm{softmax}(\mathrm{softmax}(\boldsymbol{z}))$. What will happen? Is the problem more severe when K is large or small? And why?

1. y_i = exp(w_i' x + b_i) / sum_j (w_j' x + b_j), where w_i, as a column vector, is the ith row of W matrix.

2. The sample loss is J = - sum t_k log y_k, where the sum is for k=1...K

3. The predicted probability will be over-smoothed, i.e., the model cannot learn a sharp distribution at the corner/edge of the simplex. The problem will be more severe if K is large, which will give you a very bad performance. If K is small, it's still a bug, but it's okay.

CMPUT466/566 Final Exam
Fall 2020

Problem 7 [10 marks]

Suppose the training data Dtrain has Ntrain samples. The validation set Dval has Nval samples. And the test set Dtest has Ntest samples. All samples are iid from data distribution. A hold-out validation algorithm is as follows:

```
For different hyperparameters:
    Train the model on Dtrain
    Validate the trained model on Dval
Pick the best model that yields the lowest error on Dval
Report the performance on Dtest
```

(1) [5 marks] Prove that, for any fixed hyperparameter, the error (i.e., the measure of success) on Dval is an unbiased estimate of the expected error on Dtest. Here, the error on a dataset means the average of per-sample errors.

(2) [5 marks] For the hyperparameter selected by the validation algorithm, is it still guaranteed that the validation error is an unbiased estimate for the expected test error under the same hyperparameter. If yes, prove it. If not, give one counterexample.

Suggested notations:

- Use Dtrain, Dval, and Dtest to denote the datasets (no subscript formatting is need)
- Use Err to denote the error function

1. The expected test error

E[1/Ntest * sum Err(x*, t*, h)]

= 1/ Ntest sum E[Err(x*, t*, h)]

= E[Err(x*, t*)]   -----(*)

where the expectation E is for each (x*, t*) sampled from data distribution and sum is over (x*, t*) in Dval.

ValErr = 1/Nval sum Err(x*, t*, h), for sum over x* in Dval.

By unbiasedness, we hope E[ValError] = expected test error. This indeed holds by manipulating the expectation and averaging as (*)

2. It may be biased. Counterexample:

Suppose all hyperparameters give an error rate uniform from U[0.4, 0.6]. E[TestErr] is 0.5

But if the number of hyperparameters we have tries goes to infinity, E[MinValErr] = 0.4

Problem 8 [15 marks]

CMPUT466/566 Final Exam
Fall 2020

Let $\boldsymbol{p} = (p_1, \cdots, p_K)$ be the parameters of a K-way categorical distribution, i.e., the probability of being category k is p_k. The entropy of a probability is defined as

$$H[\boldsymbol{p}] = -\sum_{k=1}^{K} p_k \log p_k$$

1. [5 marks] Show that $-H[\boldsymbol{p}]$ is a convex function of $\boldsymbol{p}$.
2. [10 marks] Use the Lagrange multiplier method to compute the maximum-entropy probability, i.e., seek $\boldsymbol{p}_*$ such that it maximizes $H[\boldsymbol{p}]$. Give the closed-form solution with key derivation steps.

Hints:

1. Calculus identities: $(fg)' = f'g + fg'$, $(\log x)' = 1/x$
2. Duality

For a convex optimization

$$\begin{aligned} \underset{\boldsymbol{x}}{\text{maximize}} \quad & f_0(\boldsymbol{x}) \\ \text{subject to} \quad & f_i(\boldsymbol{x}) \le 0 \quad \text{for } i = 1, \cdots, m \\ & h_i(\boldsymbol{x}) = 0 \quad \text{for } i = 1, \cdots, n \end{aligned}$$

where f_i is a convex function and h_i is an affine function.
The Lagrangian is defined to be

$$L(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\nu}) = f_0(\boldsymbol{x}) + \sum_{i=1}^{m} \lambda_i f_i(\boldsymbol{x}) + \sum_{i=1}^{n} \nu_i h_i(\boldsymbol{x})$$

For a convex optimization problem, the sufficient and necessary conditions for the optimality are

1) $f_i(\boldsymbol{x}) \le 0$ for $i = 1, \cdots, m$
2) $h_i(\boldsymbol{x}) = 0$ for $i = 1, \cdots, n$
3) $\lambda_i \ge 0$
4) $\lambda_i f_i(\boldsymbol{x}) = 0$ for $i = 1, \cdots, m$
5) $\nabla_{\boldsymbol{x}} L(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\nu}) = 0$

Typing suggestion: You can use a, b to represent Lagrange multipliers $\boldsymbol{\lambda}$ and $\boldsymbol{\nu}$.

1. First, the domain of H is a simplex, which is convex. Then,

d(-H)_d(p_k) = log p_k + 1

d2(-H)_d(p_k)^2 = 1/p_k

d2(-H)_dpk_dpj = 0 for j != k

Hence, the Hessian is PSD, concluding that -H is convex in p.

Mark deduction (1pts): Not computing d2(-H)_dpk_dpj = 0 for j != k is wrong. Element-wise convex does not imply joint convex.

CMPUT466/566 Final Exam
Fall 2020

2 The objective is

minimize_p sum_k p_k log p_k

subject to p1 + p2 +...+pK -1 = 0

(p_k >=0 can be ignored temporarily and will be satisfied anyway)

The Lagrangian: L = sum_k p_k log p_k + a(p1 + ... p_K - 1) = 0

Using the gradient vanishing property, we'll have dL_dpk = 1 + log pi_k + a = 0

i.e., pk = exp(-1-a)

We also know that

p1 + p2 + ... p_K = 1

then, exp(-1-a) = 1/K

Thus pk = 1/K for k = 1..K

Problem 9 [10 marks]

A Poisson distribution is a "natural" distribution for counting the number of events. For discrete $x = 0, 1, 2, \cdots$, x follows a Poisson distribution if $P(x) = \frac{\lambda^x e^{-\lambda}}{x!}$ for some parameter λ. We know the expectation of x from a Poisson distribution is just λ, i.e., $\mathbb{E}[x] = \lambda$. For typing purposes, you may use r to represent λ in your solution.

1) [5 marks] Show that the Poisson distribution is a member of the exponential family. Provide derivation steps and show what $h(x), \boldsymbol{\eta}, \boldsymbol{T}(x)$, and $A(\boldsymbol{\eta})$ are. You may represent $A(\boldsymbol{\eta})$ in λ.
2) [5 marks] Design a generalized linear model with canonical response for Poisson regression. Write out the model assumption, i.e., how to predict the expected count y based on features $\boldsymbol{x}$, where the target follows a Poisson distribution controlled by $\boldsymbol{x}$. Present the stochastic gradient descent algorithm and give the derivative. Results in the hints can be used without proof.

Hints:

1. An exponential family on x has the form $p(x; \boldsymbol{\eta}) = h(x) \exp\{\boldsymbol{\eta}^\top \boldsymbol{T}(x) - A(\boldsymbol{\eta})\}$. In the lecture, we learned
   a. Moment generating property: $\frac{\partial}{\partial \boldsymbol{\eta}} A(\boldsymbol{\eta}) = \mathbb{E}[\boldsymbol{T}(x)]$, $\frac{\partial^2}{\partial \boldsymbol{\eta} \partial \boldsymbol{\eta}^\top} A(\boldsymbol{\eta}) = \mathrm{Var}[\boldsymbol{T}(x)]$

   b. Convexity: The domain of η is a convex set, and $A(\boldsymbol{\eta})$ is a convex function in η.
   c. MLE is equivalent to moment matching.

2. A generalized linear model (GLIM) is

$$\begin{tikzcd}
\mathbf{w} \arrow[dr] & & & & \\
& \xi \arrow[r, shift left, "f\ (response)"] & \mu \arrow[l, shift left, "f^{-1}\ (link)"] \arrow[r, "\psi"] & \eta \arrow[r, "EXP"] & t \\
\mathbf{x} \arrow[ur] & & & &
\end{tikzcd}$$

where canonical response means that $f = \psi^{-1}$. We learned that $\frac{\partial J^{(m)}}{\partial \boldsymbol{w}} = (y^{(m)} - t^{(m)})\boldsymbol{x}^{(m)}$ for GLIM with canonical response, where $J^{(m)}$ is the negative log-likelihood of a particular data sample.

1. P(x) = r^x exp(-r) / x! = 1/x! * exp(log r^x) - r) = 1/x! * exp(x * log r - r)

h(x) = 1/x!

eta = log r

A(eta) = r

T(x) = x

2. Using the canonical response function, we assume eta = w'x, thus, k = exp(eta) = exp(w'x) We assume t | x ~ Poisson(k) = Poisson(exp(w'x))

The Poisson regression output is y = E[t|x] = exp(w'x)

We can use SGD to train the model.

The SGD algorithm is

```
for epochs = 1, 2, 3, ... until satisfied
        for sample m = 1...M
                w <- w- learning-rate * dJ^(m)_dw
```

where dJ_dw = (y - t) x for the data sample m, superscript omitted for J, y, t, x.