DL Assignment #1

Vikrant Dewangan
Robotics Research Center
IIIT-Hyderabad

June 5, 2020

Problem 1: MLE MAP

Problem 1.a

Show that the MLE and the MAP distribution is the same in case of uniform prior.
In the special case of uniform prior distribution, we have equal weights to all possible value to

$$\begin{aligned}\theta_{MAP} = & \\ &= arg\ max \sum_i log\ p\big(x|\theta\big) + log\ p\big(\theta\big) \\ &= arg\ max \sum_i log\ p\big(x|\theta\big) + const \\ &= arg\ max \sum_i log\ p\big(x|\theta\big) \\ &= \theta_{ML}\end{aligned}$$

Thus ML and MAP probabilities are same.

Problem 1.b

Find the parameters of a Gaussian distribution using MLE. Is the mean μ and sigma σ estimated using MLE unbiased?
We have

$$f(x|\mu,\theta) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Assuming I.I.D, we take

$$\theta = \mu, \sigma^2$$

$$f(x_1, x_2, x_3 \dots x_n|\mu, \theta) = \prod_{i=1}^{i=n} f(x_i|\theta)$$
$$= \frac{1}{(\sigma\sqrt{2\pi})^n} e^{\sum_{i=1}^{i=n}(-\frac{(x_i-\mu)^2}{2\sigma^2})}$$

We take log ,

$$\hat{\theta}_{ML} = \arg\max\ log\big(f(x_1, x_2, x_3 \dots x_n|\mu, \theta)\big)$$
$$= \arg\max\ -\frac{n}{2} log\ \big(2\pi\sigma^2\big) - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(x_i - \mu)^2$$

To check whether they are biased or not, we would have to find out $\hat{\mu}$ and $\hat{\sigma}^2$.

$$\frac{\partial L}{\partial \mu} = \frac{1}{\sigma^2}\sum_{i=1}^{n}(x_i - \mu)$$
$$= 0$$

Thus, we get

$$E[\mu] = \hat{\theta_\mu}$$
$$= \frac{1}{N}\sum_{i=1}^{n} x_i$$

Since this is the predicted value of μ over N distributions and its bias is 0, mean is unbiased. And for σ^2, we get

$$\frac{\partial L}{\partial \sigma^2} = -\frac{N}{2\sigma^2} + \frac{1}{2\sigma^4}\sum_{i=1}^{n}(x_i - \mu)^2$$
$$= 0$$

We get,

$$\sigma^2 = \sum_{i=1}^{n}\frac{(x_i - \mu)^2}{N}$$

Bias of variance over N samples distribution would be -

$$Bias(\sigma^2) = -\frac{\sigma^2}{N}$$

Since the bias is not 0, we get that it is biased.

Problem 1.c

Find the parameter μ of a Gaussian distribution with known standard deviation σ using MAP estimation. Assume the prior to be a gaussian distribution with mean 0 and standard deviation γ. Determine the optimal value of μ by setting the derivative to 0.
For MAP, we have

$$f(\theta) = \frac{1}{2\sqrt{\pi\gamma^2}} e^{(\frac{-1}{2\gamma^2})(\mu)^2}$$

We would thus have to maximize -

$$\sum_{i=1}^{i=n} (\frac{\mu}{\gamma^2} + \frac{(x_i - \mu)^2}{\sigma^2})$$

which turns out to be, by setting derivative equal to 0,

$$\hat{\mu} = \frac{\gamma^2 \sum_{i=1}^{N} x_i}{\gamma^2 N + \sigma^2}$$

Problem 2: Gradient Descent

Problem 2.a

Write the Taylor series expansion of the cost function J in terms of its derivatives (eg. Jacobian and Hessian).
Taylor series is Jacobian can be written as

$$f(x + \delta x) = f(x) + g^T \delta x + \frac{1}{2} \delta x^T H \delta x + \dots$$

Where f is the cost function, g is the Jacobian and H is the Hessian. It is to be noted that all terms after degree 3 can be ignored.

Problem 2.b

Assuming strong convexity, show that the gradient descent will always decrease the function value at each step. [Use the equation in part (a) for the same]
By definition, if f is a strongly convex function, then there exists constants m and M s.t

$$mI \leq H(x) \leq MI$$

Using the above on the taylor series expansion, we get (we take k = $\frac{1}{M}$ here.)-

$$\begin{aligned} f(x_k - ng(x_k)) &\leq f(x_k) + g^T(-ng(x_k)) + \frac{M}{2}||-kg(x_k)||^2 \\ &= f(x_k) - \frac{1}{2M}{||g(x_k)||_2}^2 \end{aligned}$$

Let x^* be the value that it converges to instead. We have

$$||x^{k+1} - x^*||_2^{\ 2} = ||x^k - x^* - \frac{1}{2M}\nabla f(x^k)||_2^{\ 2}$$
$$\leq ||x^k - x^*||_2^{\ 2} - \frac{1}{M^2}||\nabla f(x^k)||_2^{\ 2}.$$

Thus, $||x^k - x^*||_2^{\ 2}$ is a decreasing sequence. Thus f(x^k) is a decreasing sequence.

Problem 2.c

Determine the optimal learning rate at each gradient descent iteration assuming a second order approximation of the cost function.
We have,

$$f(x_{k+1}) = f(x_k) + g^T\nabla x + \frac{1}{2}\delta x^T H \delta x$$
$$\delta x = -\alpha \nabla f(x_k)$$
$$f(x_{k+1}) = f(x_k) - \alpha g^T g + \frac{\alpha^2}{2}\delta g^T H g$$

Thus, we get

$$\alpha = \frac{||g||^2}{g^T H g}$$

Problem 2.d

Derive the Newton's update rule for optimization of cost functions. It involves the second order term of the taylor expansion
We have that

$$f(x + \delta x) = f(x) + g^T \delta x + \frac{1}{2}\delta x^T H \delta x$$

Now assuming it is convex, and that the Jacobian is set to 0, we get -

$$g(x^n) + k.H(x^n) = 0$$
$$k = -\frac{g(x^n)}{H(x^n)}$$

Thus, we have

$$x^{n+1} = x^n - \frac{g(x^n)}{H(x^n)}$$

Problem 2.e

Derive the vectorized update rules for the above update rule.
Thus, we have

$$x^{n+1} = x^n - \frac{g(x^n)}{H(x^n)}$$

Problem 3: Linear and Logistic Regression

Problem 3.a

It is often advised to do feature scaling before linear regression. Explain the intuition behind the same.
It basically helps to normalise the data within a particular range. The values can explode. The features with high magnitudes will weigh in a lot more in the distance calculations than features with low magnitudes.

Problem 3.b

Show that the other formulation of logistic regression cost function shown in class is convex
To show that it is convex, we shall show that it's second derivative is non-negative always.

$$\begin{aligned}
\frac{\partial \frac{\partial J(\theta)}{\partial \theta}}{\partial \theta} &= \frac{\partial \left(\frac{1}{m}\sum_{i=1}^{m}(h_\theta(x_i) - y_i){x_j}^i\right)}{\partial \theta} \\
&= \frac{1}{m}\sum_{i=1}^{m} {x_j}^i \sigma(\theta_j^T)(1 - \sigma(\theta_j^T))\frac{\partial \theta^T x}{\partial \theta} \\
&= \sigma(\theta_j^T)(1 - \sigma(\theta_j^T)){x_j}^{i2} \\
&\geq 0
\end{aligned}$$

This the function is always convex.

Problem 4: Vector and Tensor Derivatives

Problem 4.a

Given $y = x^T Q x$, find $\frac{\partial y}{\partial x}$ and $\frac{\partial y}{\partial Q}$
Say, if x is 3 x 1 matrix and Q is a 3 x 3 matrix. We have

$$x^T Q x = \sum_{i=1}^{3} x_i (\sum_{j=1}^{3} q_{j1}x_1 + q_{j2}x_2 + q_{j3}x_3)$$

Thus, we get

$$\begin{aligned}\frac{\partial y}{\partial x} &= \frac{\partial x^T Q x}{\partial x} \\ &= (Qx)^T + x^T Q \\ &= x^T(Q + Q^T)\end{aligned}$$

and

$$\begin{aligned}\frac{\partial y}{\partial Q} &= \sum_{i=1}^{i=3}(x_{i\,i}^{x}) \\ &= x^T x\end{aligned}$$

Problem 4.b

Derive the closed form solution of linear regression with L2 Norm using the matrix derivatives method

$$\begin{aligned}J(\theta) &= \frac{1}{2n}(X\theta - y)^T(X\theta - y) + \theta^T\theta \\ \frac{\partial J(\theta)}{\partial \theta} &= 0 \\ (X^TX)\theta - X^Ty + 2\theta &= 0 \\ (X^TX + 2\lambda I)\theta &= X^Ty \\ \theta &= (X^TX + 2\lambda I)^{-1}X^Ty\end{aligned}$$

Problem 4.c

Derive the derivative of the softmax function with respect to its inputs.

$$\begin{aligned}\frac{\partial \frac{e^{W_j x}}{\sum e^{W_i x}}}{\partial W_i} &= \sum \frac{\partial \frac{e^{W_j x}}{\sum e^{W_i x}}}{\partial W_k x}\frac{\partial W_k x}{\partial W_i} \\ &= \sum \frac{\partial \frac{e^{W_j x}}{\sum e^{W_i x}}}{\partial W_k x}\delta_{ik}x \\ &= \sum \sigma(j)(\delta_{jk} - \sigma(k))\delta_{ik}x \\ &= \sigma(j)(\delta_{ij} - \sigma(i))x\end{aligned}$$

Problem 5: Neural Networks and Backpropagation

Problem 5.a

Neural networks can be used to represent simple logic gates such as binary AND OR. Define a single layer neural network (its weights and activation function) to

represent the above two functions (AND and OR).
First, we will define an activation function which will give 1 when Wx+b ¿ 0 and 0 when Wx+b $\leq$ 0. This can be satisfied by sigmoid.

- AND gate
  Say having 2 inputs, W_1 and W_2, we have a function $W_1x_1 + W_2x_2 + b$, where b = -1. Having σ as sigmoid(σ) function.
  - when $x_1 = 0$, and $x_2 = 0$, we have $z = -1$, thus $a = 0$.
  - when $x_1 = 1$, and $x_2 = 0$, we have $z = W_1 - 1$, thus $a = 0$ iff $W_1 = 1$.
  - when $x_1 = 0$, and $x_2 = 1$, we have $z = W_2 - 1$, thus $a = 0$ iff $W_2 = 1$.
  - when $x_1 = 1$, and $x_2 = 1$, we have $z = W_1 + W_2 - 1$, thus $a = 1$.

  Thus we get $a = \sigma(x_1 + x_2 - 1)$.

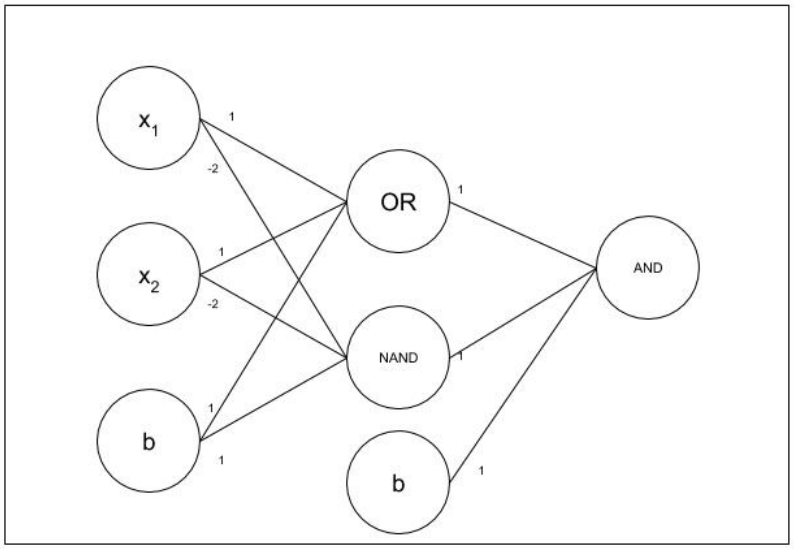
- OR gate
  Say having 2 inputs, W_1 and W_2, we have a function $W_1x_1 + W_2x_2 + b$, where b = -1. Having σ as sigmoid(σ) function.
  - when $x_1 = 0$, and $x_2 = 0$, we have $z = -1$, thus $a = 0$.
  - when $x_1 = 1$, and $x_2 = 0$, we have $z = W_1 - 1$, thus $a = 1$ iff $W_1 = 2$.
  - when $x_1 = 0$, and $x_2 = 1$, we have $z = W_2 - 1$, thus $a = 0$ iff $W_2 = 2$.
  - when $x_1 = 1$, and $x_2 = 1$, we have $z = W_1 + W_2 - 1$, thus $a = 1$.

  Thus we get $a = \sigma(2x_1 + 2x_2 - 1)$.

Problem 5.b

A single layer neural network cannot represent certain complex gates such as XOR and XNOR. We need a two layer neural network with non-linearities to define such functions. Define a 2 layer neural network with non linear activation functions like Signum/ReLU that can represent the XOR and XNOR gates.
We shall define XOR gate.

$$x_1 \text{ XOR } x_2 = (x_1 \text{ OR } x_2) \text{ AND } (x_1 \text{ NAND } x_2)$$



Here, the activation function used is sigmoid function.

Problem 5.c

Find the derivative of f(x;W) with respect to x and w.

$$\begin{aligned} f(x, W) &= \sum_{i=1}^{n} (Wx)_i{}^2 \\ &= (Wx)((Wx)^T) \\ &= (Wxx^TW^T) \end{aligned}$$

Problem 5.d

$$\begin{aligned} z_1 &= XW_1 \\ h_1 &= ReLU(z_1) \\ \hat{y} &= h_1W_2 \\ L &= ||\hat{y}||^2 \end{aligned}$$

We have,

$$\begin{aligned}\frac{\partial L}{\partial W_1} &= \frac{\partial L}{\partial y}\frac{\partial y}{\partial h_1}\frac{\partial h_1}{\partial z_1}\frac{\partial z_1}{\partial W_1} \\ &= 2\hat{y}W_2X^T(if z_1 \geq 0), 0otherwise. \\ \frac{\partial L}{\partial W_2} &= \frac{\partial L}{\partial y}\frac{\partial y}{\partial W_2} \\ &= 2\hat{y}{h_1}^T.\end{aligned}$$

Problem 5.e

We have

$$\begin{aligned}z &= w_0 + w_1x + w_2x^2 \\ y &= 1 + e^z \\ L &= \frac{1}{2}(logy - logt)^2 \\ \frac{\partial L}{\partial W_2} &= \frac{\partial L}{\partial y}\frac{\partial y}{\partial z}\frac{\partial z}{\partial W_2} \\ \frac{\partial L}{\partial y} &= \frac{1}{y}(logy - logt) \\ \frac{\partial y}{\partial z} &= e^z \\ \frac{\partial z}{\partial W_2} &= x^2\end{aligned}$$

Thus, we get

$$\frac{\partial L}{\partial W_2} = \frac{1}{y}(logy - logt)e^z x^2$$

Problem 6: Convolutional Neural Networks

Problem 6.a

If the previous layer has size JxK, and a filter of sizeMxN is applied with strides and zero-padding of width P, what will be the size of the resulting convolutional layer?

We have

$$\begin{aligned}W &= \left[\frac{J + 2P - M}{S} + 1\right] \\ H &= \left[\frac{K + 2P - N}{S} + 1\right]\end{aligned}$$

Problem 6.b

If max pooling with filter size F and strides is applied to a layer of size JxK, what will be the size of the resulting (downsampled) layer
We have

$$W = \left[\frac{J-F}{S}\right]$$
$$H = \left[\frac{K-F}{S}\right]$$

Problem 6.c

Can fully connected layers be represented using convolution layers itself? If yes, show how?
Yes we shall accomplish this in 2 steps -

1. choosing a convolutional kernel that has the same size as the input feature map
2. using 1x1 convolutions with multiple channels.

Thus, if size of input image is $w \times h$, we choose a kernel of size $w \times h$, and then choose $w \cdot h$ size 1×1 convolutions. In this way, we can fully connected layers be represented using convolution layers.