CS229 Fall 2017, Problem Set #3: Deep Learning & Supervised Learning

Armand Sumo – `armandsumo@gmail.com`

June 11, 2021

Collaborators:

By turning in this assignment, I agree by the Stanford honor code and declare that all of this is my own work.

(The code used to generate the graphs can be found in the file *assignment-3.py*.)

1. A Simple Neural Network

(a) Assuming we use a learning rate of α, the gradient descent update to $w_{1,2}^{[1]}$ takes the form:

$$w_{1,2}^{[1]} := w_{1,2}^{[1]} - \alpha \frac{\partial l}{\partial w_{1,2}^{[1]}}$$

Where l is the average squared loss:

$$l = \frac{1}{m} \sum_{i=1}^{m} (o^{(i)} - y^{(i)})$$

We will begin by looking at the stochastic gradient descent update rule and drop the superscript $(\cdot)^{(i)}$. Next, we apply the chain rule to the term $\frac{\partial l}{\partial w_{1,2}^{[1]}}$

$$\frac{\partial l}{\partial w_{1,2}^{[1]}} = \frac{\partial l}{\partial o} \frac{\partial o}{\partial h_2} \frac{\partial h_2}{\partial w_{1,2}^{[1]}} \tag{1}$$

The structure of the neural network imposes:

$$\begin{aligned} h_2 &= w_{1,2}^{[1]} x_1 + w_{2,2}^{[1]} x_2 + w_{0,2}^{[1]} \\ o &= \sigma(w_1^{[2]} \sigma(h_1) w_2^{[2]} \sigma(h_2) + w_3^{[2]} \sigma(h_3) + w_0^{[2]}) \end{aligned}$$

From these equations and the definition of the the squared loss l for a single example, we derive the gradients:

$$\frac{\partial l}{\partial o} = 2(o-y) \tag{2}$$

$$\frac{\partial h_2}{\partial w^{[1]}_{1,2}} = x_1 \tag{3}$$

$$\frac{\partial o}{\partial h_2} = \frac{\partial o}{\partial \sigma(h_2)}\sigma'(h_2) = w^{[2]}_2 \sigma'(\sum_{j=1}^{3} w^{[2]}_j \sigma(h_j) + w^{[2]}_0)\sigma'(h_2) \tag{4}$$

Plugging (2),(3) and (4) into (1) and property of the sigmoid function $\sigma'(x) = \sigma(z)(1-\sigma(z))$,we obtain:

$$\frac{\partial l}{\partial w^{[1]}_{1,2}} = \Big[x_1 w^{[2]}_2 \sigma'(\sum_{j=1}^{3} w^{[2]}_j \sigma(h_j) + w^{[2]}_0)\Big(1-\sigma(\sum_{j=1}^{3} w^{[2]}_j \sigma(h_j) + w^{[2]}_0)\Big)\times$$
$$\sigma(\sum_{k=1}^{2} w^{[1]}_{k,2} + w^{[1]}_{0,2})\Big(1-\sigma(\sum_{k=1}^{2} w^{[1]}_{k,2} + w^{[1]}_{0,2})\Big)2(o-y)\Big]$$

Hence, the gradient update to $w^{[1]}_{1,2}$ is

$$w^{[1]}_{1,2} := w^{[1]}_{1,2} - \alpha\frac{1}{m}\sum_{i=1}^{m}\Big[x^{(i)}_1 w^{[2]}_2 \sigma'(\sum_{j=1}^{3} w^{[2]}_j \sigma(h_j) + w^{[2]}_0)\Big(1-\sigma(\sum_{j=1}^{3} w^{[2]}_j \sigma(h_j) + w^{[2]}_0)\Big)\times$$
$$\sigma(\sum_{k=1}^{2} w^{[1]}_{k,2} + w^{[1]}_{0,2})\Big(1-\sigma(\sum_{k=1}^{2} w^{[1]}_{k,2} + w^{[1]}_{0,2})\Big)2(o^{(i)}-y^{(i)})\Big]$$

(b) The classification of our data set x into 2 classes can be understood as the separation of the $\mathbb{R}\times\mathbb{R}$ plane into two sets S and its complement with respect to the entire plane $\mathbb{R}\times\mathbb{R}-S$. Where S is the set containing positive examples. For our neural network to have 100% accuracy, it must for any ordered pair $x^{(i)} = (x_1, x_2)$ produce the following output:

$$neuralnet(x^{(i)})\begin{cases}1 & (x_1,x_2)\in S\\ 0 & (x_1,x_2)\notin S\end{cases}$$

To find the structure of S, we must observe the dataset. The negative examples(labeled (0)) are on the inside of a triangle. The set of positive examples S can be formulated as: $S = S_1 \cup S_2 \cup S_3$ where:

$$S_1 = \{(x_1,x_2)\in\mathbb{R}^2 \mid x_2 \leq \frac{1}{4}, x_1\in\mathbb{R}\}$$
$$S_2 = \{(x_1,x_2)\in\mathbb{R}^2 \mid x_1 \leq \frac{1}{4}, x_2\in\mathbb{R}\}$$
$$S_3 = \{(x_1,x_2)\in\mathbb{R}^2 \mid x_1 + x_2 \leq \frac{1}{4}\}$$

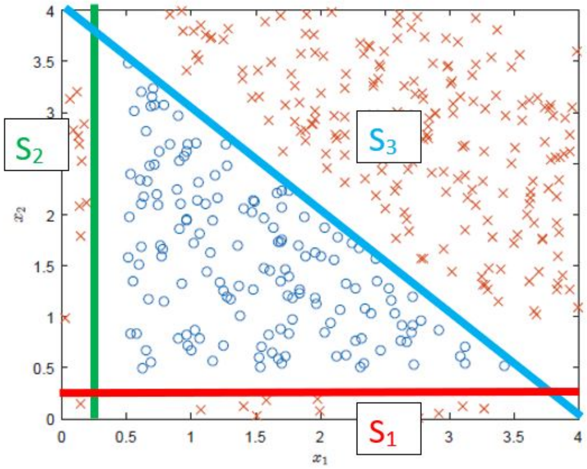


Figure 1: Training data and decision boundary learned by neural network

Figure 1. shows the subsets S_1 and S_2 and S_3 and their respective decision boundaries in red, green and blue. The 3 neurons inside the hidden layer are defined as following:

$$
\begin{aligned}
h_1 &= w_{0,1}^{[1]} + w_{1,1}^{[1]} x_1 + w_{2,1}^{[1]} x_2 \\
h_2 &= w_{0,2}^{[1]} + w_{1,2}^{[1]} x_1 + w_{2,2}^{[1]} x_2 \\
h_3 &= w_{0,3}^{[1]} + w_{1,3}^{[1]} x_1 + w_{2,3}^{[1]} x_2
\end{aligned}
$$

We can therefore use h_1 and h_2 and h_3 to encode the boundaries of respectively S_1 and S_2 and S_3:

$$
\begin{cases} x_2 - \frac{1}{4} = 0 \\ x_1 - \frac{1}{4} = 0 \\ x_1 + x_2 = 0 \end{cases} \quad \text{with} \begin{cases} w_{0,1}^{[1]} = -\frac{1}{4},\ w_{1,1}^{[1]} = 0,\ w_{2,1}^{[1]} = 1 \\ w_{0,1}^{[1]} = -\frac{1}{4},\ w_{1,1}^{[1]} = 1,\ w_{2,1}^{[1]} = 0 \\ w_{0,1}^{[1]} = -4,\ w_{1,1}^{[1]} = 1,\ w_{2,1}^{[1]} = 1 \end{cases}
$$

Using the the step function

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

as our activation function, we can tell the neural network to distinguish between a positive and negative input:

$$
\begin{cases} f(h_1) = 1 & h_1 = x_2 - \frac{1}{4} \leq 0 \\ f(h_2) = 1 & h_2 = x_1 - \frac{1}{4} \leq 0 \\ f(h_3) = 1 & h_3 = x_1 + x_2 - 4 \leq 0 \end{cases}
$$

Our neural network now distinguishes between the interior and exterior of S_1 and S_2 and S_3. The output neuron o is defined as:

$$o = w_0^{[2]} + w_1^{[2]} f(h_1) + w_2^{[2]} f(h_2) + w_3^{[2]} f(h_3)$$

We now apply the step function to the output o:

$$\begin{cases} f(o) = 1 & w_0^{[2]} + w_1^{[2]} f(h_1) + w_2^{[2]} f(h_2) + w_3^{[2]} f(h_3) \geq 0 \\ f(o) = 0 & w_0^{[2]} + w_1^{[2]} f(h_1) + w_2^{[2]} f(h_2) + w_3^{[2]} f(h_3) < 0 \end{cases}$$

We want our neural network to output 1 when $(x_1, x_2) \in S$ and 0 when $(x_1, x_2) \in \mathbb{R} \times \mathbb{R} - S$.
$\mathbb{R} \times \mathbb{R} - S$ can be rewritten as $\mathbb{R} \times \mathbb{R} - S_1 \cup S_2 \cup S_3 = \overline{S_1} \cap \overline{S_2} \cap \overline{S_3}$.
An example must be classified as negative when it belongs to the intersection of the complements of S_1 and S_2 and S_3. We can now translate this into our neural network:

$$\begin{cases} f(o) = 1 & \Leftrightarrow f(h_1) = 0 \vee f(h_2) = 0 \vee f(h_3) = 0 \\ f(o) = 0 & \Leftrightarrow f(h_1) = 1 \wedge f(h_2) = 1 \wedge f(h_3) = 1 \end{cases}$$

By setting the hidden layer's weights to:

$$w_0^{[2]} = 2,\ w_1^{[2]} = -1,\ w_2^{[2]} = -1,\ w_3^{[2]} = -1$$

we get the desired behavior. As an example, if:

$$x_2 - 0.25 = h_1 \geq 0,\ x_1 - 0.25 = h_2 \geq 0,\ x_1 + x_2 - 4 = h_3 \geq 0$$

The data point (x_1, x_2) is inside the "triangle" defined $\mathbb{R} \times \mathbb{R} - S = \overline{S_1} \cap \overline{S_2} \cap \overline{S_3}$ and should be classified as (0). We have

$$f(h_1) = 1,\ f(h_2) = 1,\ f(h_3) = 1$$

hence

$$o = w_0^{[2]} + w_1^{[2]} f(h_1) + w_2^{[2]} f(h_2) + w_3^{[2]} f(h_3) = 2 - 1 \times 1 - 1 \times 1 - 1 \times 1 = -1 < 0$$

$f(o) = 0$ as desired. If our data point is outside of the triangle, it means that at least one of the $f(h_j)$'s is equal to 0. Then $o \geq 0$ and $f(o) = 1$ as desired.
To conclude, for our neural network to achieve 100% accuracy on the provided dataset, it must have the following weights:

$$w_{0,1}^{[1]} = -\frac{1}{4},\ w_{1,1}^{[1]} = 0,\ w_{2,1}^{[1]} = 1$$

$$w_{0,1}^{[1]} = -\frac{1}{4},\ w_{1,1}^{[1]} = 1,\ w_{2,1}^{[1]} = 0$$

$$w_{0,1}^{[1]} = -4,\ w_{1,1}^{[1]} = 1,\ w_{2,1}^{[1]} = 1$$

$$w_0^{[2]} = 2,\ w_1^{[2]} = -1,\ w_2^{[2]} = -1,\ w_3^{[2]} = -1$$

(c) The loss is made zero if and only if our predictions on all examples are correct(a sum of positive terms is equal to zero if and only if all the terms are zero). If we let the activation functions for h_1, h_2 h_3 be linear functions $f(x) = x$ and the activation function for o be the same as before, we get the following expression:

$$\begin{aligned} o = \underbrace{(w_0^{[2]} + w_1^{[2]}w_{0,1}^{[1]} + w_2^{[2]}w_{0,2}^{[1]} + w_3^{[2]}w_{0,3}^{[1]})}_{a_0} + \\ \underbrace{(w_1^{[2]}w_{1,1}^{[1]} + w_2^{[2]}w_{1,2}^{[1]} + w_3^{[2]}w_{1,3}^{[1]})}_{a_1} x_1 + \\ \underbrace{(w_1^{[2]}w_{2,1}^{[1]} + w_2^{[2]}w_{2,2}^{[1]} + w_3^{[2]}w_{2,3}^{[1]})}_{a_2} x_2 \end{aligned}$$

The neural network can be therefore be reduced to:

$$\text{neuralnet}((x_1, x_2)) = \begin{cases} 1 & a_0 + a_1x_1 + a_2x_2 \geq 0 \\ 0 & a_0 + a_1x_1 + a_2x_2 < 0 \end{cases}$$

For this neural network to predict all classes correctly, it must be capable of representing the 3 decision boundaries S_1, S_2 and S_3 described in (b). For any example (x_1, x_2) at the boundary ($o = 0$), the following system of equations must have a solution:

$$\begin{cases} a_1x_1 + a_2x_2 & = -a_0 \\ 0 + x_2 & = \frac{1}{4} \\ x_1 + 0 & = \frac{1}{4} \\ x_1 + x_2 & = 4 \end{cases}$$

This is not the case because $x_1 + x_2$ cannot simultaneously be equal to 1/2 and 4. Hence, there are is no sets of weights a_0, a_1 and a_2 such that the neural network can learn the decision boundaries necessary to classify all examples correctly and obtain a loss of 0.

2. EM for MAP estimation

(a) The MAP estimate of the parameters θ found by maximizing:

$$\left(\prod_{i=1}^{m} p(x^{(i)} \mid \theta)\right) p(\theta)$$

is the same as the one found by maximizing

$$\sum_{i=1}^{m} \log p(x^{(i)} \mid \theta) + \log p(\theta) \tag{5}$$

We rewrite the objective in terms of the joint densities $p(x^{(i)}, z^{(i)} \mid \theta)$ and $p(\theta, z)$

$$\sum_{i=1}^{m} \log \sum_{z^{(i)}} p(x^{(i)}, z^{(i)} \mid \theta) + \log \sum_{z} p(\theta, z) \tag{6}$$

Our strategy is to repeatedly construct a lower bound on our objective(E-step) and maximize that lower bound(M-step).
Let Q_i be a distribution over the possible values of $z^{(i)}$ and R over the possible values of z such that $\sum_{z^{(i)}} Q_i(z^{(i)}) = 1$ and $\sum_z R(z) = 1$. We then have:

$$\begin{aligned} \sum_{i=1}^{m} \log p(x^{(i)} \mid \theta) = \sum_{i=1}^{m} \log \sum_{z^{(i)}} p(x^{(i)}, z^{(i)} \mid \theta) &= \sum_{i=1}^{m} \log \sum_{z^{(i)}} Q_i(z^{(i)}) \frac{p(x^{(i)}, z^{(i)} \mid \theta)}{Q_i(z^{(i)})} && (7) \\ &\geq \sum_{i=1}^{m} \sum_{z^{(i)}} \log \frac{p(x^{(i)}, z^{(i)} \mid \theta)}{Q_i(z^{(i)})} && (8) \end{aligned}$$

The last step of the derivation used Jensen's inequality. Specifically, $f(x) = \log x$ is a concave function and

$$Q_i(z^{(i)}) \left[\frac{p(x^{(i)}, z^{(i)} \mid \theta)}{Q_i(z^{(i)})} \right]$$

is the expected value of the quantity $[p(x^{(i)}, z^{(i)} \mid \theta)/Q_i(z^{(i)})]$ with respect to $z^{(i)}$ drawn according to the distribution given by Q_i. By Jensen's inequality, we have:

$$f\left(E_{z^{(i)} \sim Q_i} \left[\frac{p(x^{(i)}, z^{(i)} \mid \theta)}{Q_i(z^{(i)})} \right] \right) \geq E_{z^{(i)} \sim Q_i} \left[f\left(\frac{p(x^{(i)}, z^{(i)} \mid \theta)}{Q_i(z^{(i)})} \right) \right]$$

Similarly,

$$\log p(\theta) = \log \sum_{z} p(\theta, z) = \log \sum_{z} R(z) \frac{p(\theta, z)}{R(z)} \geq \sum_{z} \log \frac{p(\theta, z)}{R(z)} \tag{9}$$

For **any** distribution Q_i and R, formulas (8) and (9) give respectively a lower bound for $\sum_{i=1}^{m} \log p(x^{(i)} \mid \theta)$ and for $\log p(\theta)$.
There are many possible choices of Q's and R's. A good guess it to choose the one that makes the bound tight for our particular value of θ, for this to be the case, we need the step involving Jensen's inequality in our derivation to hold for equality. We'll start with inequality (8). For it to be an equality, it is sufficient the expected value be taken over a "constant" valued random variable, I.e we require that:

$$\left[\frac{p(x^{(i)}, z^{(i)} \mid \theta)}{Q_i(z^{(i)})} \right] = c$$

For some constant c that does not depend on z. This can be accomplished by choosing

$$Q_i(z^{(i)}) \propto p(x^{(i)}, z^{(i)} \mid \theta)$$

Since we know that $\sum_z Q(z) = 1$(because it is a distribution), this further tells us that:

$$\begin{aligned} Q_i(z^{(i)}) &= \frac{p(x^{(i)}, z^{(i)} \mid \theta)}{\sum_{z^{(i)}} p(x^{(i)}, z^{(i)} \mid \theta)} \\ &= \frac{p(x^{(i)}, z^{(i)} \mid \theta)}{p(x^{(i)} \mid \theta)} \\ &= p(z^{(i)} \mid (x^{(i)}, \theta)) \end{aligned}$$

Similarly, from equation (9), we choose:

$$\begin{aligned} R(z) &= \frac{p(\theta, z)}{\sum_z p(\theta, z)} \\ &= \frac{p(\theta, z)}{p(\theta)} \\ &= p(z \mid \theta) \end{aligned}$$

For the sake convenience, for each example i, we denote the expression found in equation (8):

$$\text{ELBO}(x^{(i)}, Q_i, \theta) = \sum_{z^{(i)}} \log \frac{p(x^{(i)}, z^{(i)} \mid \theta)}{Q_i(z^{(i)})}$$

and from the expression found in equation (9):

$$\text{ELBO}(R, \theta) = \sum_z \log \frac{p(\theta, z)}{R(z)}$$

If we assume that $\sum_{i=1}^m \log p(x^{(i)} \mid \theta)$ and $\log p(\theta)$ are both concave in θ, then the M-step is tractable since it only requires maximizing a linear combination of these quantities.
We can therefore write a lower bound $\text{ELBO}(x^{(i)}, Q_i, R, \theta)$ such that:

$$\forall Q_i, R, \theta, x^{(i)} \quad \sum_{i=1}^m \log p(x^{(i)} \mid \theta) + \log p(\theta) \geq \sum_{i=1}^m \text{ELBO}(x^{(i)}, Q_i, R, \theta)$$

The EM Algorithm can therefore be formulated as:

Repeat until convergence{

(E-Step) For each , set:

$$\begin{aligned} Q_i(z^{(i)}) &= p(z^{(i)} \mid (x^{(i)}, \theta) \\ R(z^{(i)}) &= p(z^{(i)} \mid \theta) \end{aligned}$$

(M-Step) Set:

$$\begin{aligned}\theta &:= \underset{\theta}{\operatorname{argmax}}\, \mathrm{ELBO}(x^{(i)}, Q_i, R, \theta) \\ &:= \underset{\theta}{\operatorname{argmax}} \sum_{i=1}^{m} \sum_{z^{(i)}} \log \frac{p(x^{(i)}, z^{(i)} \mid \theta)}{Q_i(z^{(i)})} + \sum_{z^{(i)}} \log \frac{p(\theta, z^{(i)})}{R(z^{(i)})}\end{aligned}$$

}

To know if this EM algorithm converges, let $\theta^{(t)}$ and $\theta(t+1)$ be two successive iterations of EM, we will now prove that

$$\sum_{i=1}^{m} \log p(x^{(i)} \mid \theta^{(t)}) + \log p(\theta^{(t)}) \leq \sum_{i=1}^{m} \log p(x^{(i)} \mid \theta^{(t+1)}) + \log p(\theta^{(t+1)})$$

Which shows EM always monotonically improves our objective function.

$$\begin{aligned}\sum_{i=1}^{m} \log p(x^{(i)} \mid \theta^{(t+1)}) + \log p(\theta^{(t+1)}) &\geq \mathrm{ELBO}(x^{(i)}, Q_i, R, \theta^{(t+1)}) && (10) \\ &\geq \mathrm{ELBO}(x^{(i)}, Q_i, R, \theta^{(t)}) && (11) \\ &= \sum_{i=1}^{m} \log p(x^{(i)} \mid \theta^{(t)}) + \log p(\theta^{(t)}) && (12)\end{aligned}$$

Where (10) holds by definition of the lower bound on the objective function and (11) holds because we choose $\theta_{(t+1)}$ such as to maximize the lower bound.

Our M-step is tractable and our EM algorithm converges.

3. EM application

We let $x^{(pr)}$ denote the score given to paper $p \in \{1, \cdots, P\}$ by reviewer $r \in \{1, \cdots, R\}$

First, we decompose the distribution of the observed variable $x^{(pr)}$ as the sum of latent variables $y^{(pr)}$ and $z^{(pr)}$, with the addition of a Gaussian noise $\epsilon^{(pr)}$ with fixed, known parameters:

$$x^{(pr)} = y^{(pr)} + z^{(pr)} + \epsilon^{(pr)}$$

where

$$\begin{aligned}y^{(pr)} &\sim \mathcal{N}(\mu_p, \sigma_p^2) \\ z^{(pr)} &\sim \mathcal{N}(\nu_r, \tau_r^2) \\ \epsilon^{(pr)} &\sim \mathcal{N}(0, \sigma^2)\end{aligned}$$

The variables $y^{(pr)}$,$z^{(pr)}$ and $\epsilon^{(pr)}$ are independent.
We will estimate the parameters $\mu_p, \sigma_p^2, \nu_r, \tau_r^2$ by maximizing the marginal likelihood of the observed data $x^{(pr)}, p \in \{1, \cdots, P\},\ r \in \{1, \cdots, R\}$. This problem cannot be solved in closed form so we will use the EM algorithm.

(a) In this part, we'll derive the E-step.

(i) The joint distribution $p(y^{(pr)}, z^{(pr)}, x^{(pr)})$ has the form of a multivariate Gaussian density :

$$\begin{bmatrix} y^{(pr)} \\ z^{(pr)} \\ x^{(pr)} \end{bmatrix} \sim \mathcal{N}(\mu_{yzx}^{(pr)}, \Sigma_{yzx}^{(pr)})$$

Where $\mu_{xyz}^{(pr)}$ is its mean vector and $\Sigma_{yzx}^{(pr)}$ its covariance matrix. To find the mean vector we first note that

$$\mathbb{E}[x^{(pr)}] = \mathbb{E}[y^{(pr)} + z^{(pr)} + \epsilon^{(pr)}] = \mu_p + \nu_r$$

hence

$$\mu_{yzx}^{(pr)} = \begin{bmatrix} \mu_p \\ \nu_r \\ \mu_p + \nu_r \end{bmatrix}$$

We write down the covariance matrix as:

$$\Sigma_{yzx}^{(pr)} = \begin{bmatrix} \sigma_{yy} & \sigma_{yz} & \sigma_{yx} \\ \sigma_{zy} & \sigma_{zz} & \sigma_{zx} \\ \sigma_{xz} & \sigma_{xy} & \sigma_{xx} \end{bmatrix}_{(pr)}$$

First we compute the terms on the diagonal

$$\begin{aligned} \sigma_{xx} &= \mathbb{E}[(x - \mathbb{E}[x])^2] \\ &= \mathbb{E}[x^2] - \mathbb{E}[x]^2 = \sigma^2 \end{aligned}$$

similarly,

$$\begin{aligned} \sigma_{yy} &= \sigma_p^2 \\ \sigma_{zz} &= \tau_r^2 \end{aligned}$$

Next we compute the terms above the diagonal

$$\begin{aligned}\sigma_{yx} &= \mathbb{E}[(x - \mathbb{E}[x])(y - \mathbb{E}[y])] \\ &= \mathbb{E}[xy - x\mathbb{E}[y] - y\mathbb{E}[x] + \mathbb{E}[x]\mathbb{E}[y]] \\ &= \mathbb{E}[xy] - \mu_p(\mu_p + \nu_r)\end{aligned}$$

since y,z and ϵ are independent and $\mathbb{E}[\epsilon] = 0$

$$\begin{aligned}\sigma_{xy} &= \mathbb{E}[y^2] + \mathbb{E}[y]\mathbb{E}[z] \\ &= \sigma_p^2 + \mu_p^2 + \mu_p\nu_r\end{aligned}$$

Similarly,

$$\sigma_{zx} = \tau_r^2 + \nu_r^2 + \nu_r\mu_p$$

Finally, because y and z are independent

$$\sigma_{yz} = 0$$

Because the covariance matrix is symmetric:

$$\Sigma_{yzx}^{(pr)} = \begin{bmatrix} \sigma_p^2 & 0 & \sigma_p^2 + \mu_p^2 + \mu_p\nu_r \\ 0 & \tau_r^2 & \tau_r^2 + \nu_r^2 + \nu_r\mu_p \\ \sigma_p^2 + \mu_p^2 + \mu_p\nu_r & \tau_r^2 + \nu_r^2 + \nu_r\mu_p & \sigma^2 \end{bmatrix} \tag{13}$$

We therefore have:

$$\begin{bmatrix} y^{(pr)} \\ z^{(pr)} \\ x^{(pr)} \end{bmatrix} \sim \mathcal{N}(\begin{bmatrix} \mu_p \\ \nu_r \\ \mu_p + \nu_r \end{bmatrix}, \Sigma_{yzx}^{(pr)}) \tag{14}$$

(ii) In the E-step, we derive a *lower bound* for the function we wish to maximize which is the log-likelihood of the parameters our latent variables $y^{(pr)}$ and $z^{(pr)}$:

$$\sum_p \sum_r \log p(y^{(pr)}, z^{(pr)}, x^{(pr)}; \mu_p, \sigma_p^2, \nu_r, \tau_r^2) \tag{15}$$

We then introduce a distribution Q_{pr} that allows us to express the lower bound as the expected value of some random vector:

$$\sum_p \sum_r \log p(y^{(pr)}, z^{(pr)}, x^{(pr)}; \mu_p, \sigma_p^2, \nu_r, \tau_r^2)$$
$$= \sum_p \sum_r \log \int_{(y^{(r)}, z^{(r)})} Q_{pr}(y^{(pr)}, z^{(pr)}) \frac{p(y^{(pr)}, z^{(pr)}, x^{(pr)}; \mu_p, \sigma_p^2, \nu_r, \tau_r^2)}{Q_{pr}(y^{(pr)}, z^{(pr)})} d(y^{(pr)}, z^{(pr)})$$
$$= \sum_p \sum_r \log \mathbb{E}_{(y^{(pr)}, z^{(pr)}) \sim Q_{pr}} \left[\frac{p(y^{(pr)}, z^{(pr)}, x^{(pr)}; \mu_p, \sigma_p^2, \nu_r, \tau_r^2)}{Q_{pr}(y^{(pr)}, z^{(pr)})} \right]$$

Here, the "$(y^{(pr)}, z^{(pr)}) \sim Q_{pr}$" subscript denotes that the expected value is with respect to $\begin{bmatrix} y^{(pr)} \\ z^{(pr)} \end{bmatrix}$ drawn from Q_{pr}. Using Jensen's inequality for the concave function $f(x) = \log x$ and the distribution $\frac{p(y^{(pr)}, z^{(pr)}, x^{(pr)}; \mu_p, \sigma_p^2, \nu_r, \tau_r^2)}{Q_{pr}(y^{(pr)}, z^{(pr)})}$, we derive the desired lower bound:

$$\sum_p \sum_r \log \mathbb{E}_{(y^{(pr)}, z^{(pr)}) \sim Q_{pr}} \left[\frac{p(y^{(pr)}, z^{(pr)}, x^{(pr)}; \mu_p, \sigma_p^2, \nu_r, \tau_r^2)}{Q_{pr}(y^{(pr)}, z^{(pr)})} \right]$$
$$\geq \sum_p \sum_r \mathbb{E}_{(y^{(pr)}, z^{(pr)}) \sim Q_{pr}} \log \left[\frac{p(y^{(pr)}, z^{(pr)}, x^{(pr)}; \mu_p, \sigma_p^2, \nu_r, \tau_r^2)}{Q_{pr}(y^{(pr)}, z^{(pr)})} \right]$$

Next, We choose a value of $\begin{bmatrix} y^{(pr)} \\ z^{(pr)} \end{bmatrix}$ that makes that lower bound tight i.e. that makes the case of equality for Jensen's inequality hold. Using similar steps to the ones employed in section (2), we derive an expression for Q_{pr}:

$$Q_{pr}(y^{(pr)}, z^{(pr)}) = p(y^{(pr)}, z^{(pr)} \mid x^{(pr)})$$

We now develop this expression using the rules for conditioning on subsets of jointly Gaussian random variables. We find that

$$y^{(pr)}, z^{(pr)} \mid x^{(pr)}, \mu_p, \sigma_p, \nu_r, \tau_r \sim \mathcal{N}(\mu_{y^{(pr)}, z^{(pr)} | x^{(pr)}}, \Sigma_{y^{(pr)}, z^{(pr)} | x^{(pr)}})$$

where

$$\mu_{y^{(pr)}, z^{(pr)} | x^{(pr)}} = [\mu_{(y,z)} + \Sigma_{((x,y),z)} (\Sigma_x)^{-1} (x - \mu_x)]_{(pr)} \tag{16}$$

and

$$\Sigma_{y^{(pr)},z^{(pr)}|x^{(pr)}} = [\Sigma_{(y,z)} - \Sigma_{((x,y),z)}(\Sigma_x)^{-1}\Sigma_{(z,(x,y))}]_{(pr)} \tag{17}$$

Here,

$\mu_{(y,z)}$ is the mean term of the marginal distribution of (y,z)

$\Sigma_{(y,z)}$ is the covariance term of the marginal distribution of (y,z)

$\Sigma_{((x,y),z)}$ is the covariance term of the joint distribution of $((y,z),x))$.

$\Sigma_{(z,(x,y))} = \Sigma^T_{((x,y),z)}$ is the covariance term of the joint distribution of $(z,(x,y))$

Σ_x is the covariance term of the marginal distribution of x

x is the marginal random term x

μ_x is the mean term of the marginal distribution of x

The expression of $Q_{pr}(y^{(pr)}, z^{(pr)})$ is:

$$Q_{pr}(y^{(pr)}, z^{(pr)}) = \frac{1}{(2\pi)^2|\Sigma_{y^{(pr)},z^{(pr)}|x^{(pr)}}|^{\frac{1}{2}}}$$

$$\times \exp\left(-\frac{1}{2}\left[\begin{bmatrix} y^{(pr)} \\ z^{(pr)} \end{bmatrix} - \mu_{y^{(pr)},z^{(pr)}|x^{(pr)}}\right]^T \Sigma^{-1}_{y^{(pr)},z^{(pr)}|x^{(pr)}} \left[\begin{bmatrix} y^{(pr)} \\ z^{(pr)} \end{bmatrix} - \mu_{y^{(pr)},z^{(pr)}|x^{(pr)}}\right]\right) \tag{18}$$

(b) Let's now work out the M-step update for the parameters $\{\mu_p, \sigma_p^2, \nu_r, \tau_r^2\}$. We want to find the values for these parameters that maximize the lower bound we just found:

$$\sum_p \sum_r \mathbb{E}_{(y^{(pr)},z^{(pr)})\sim Q_{pr}} \log\left[\frac{p(y^{(pr)}, z^{(pr)}, x^{(pr)}; \mu_p, \sigma_p^2, \nu_r, \tau_r^2)}{Q_{pr}(y^{(pr)}, z^{(pr)})}\right] \tag{19}$$

Here the subscript "$(y^{(pr)}, z^{(pr)}) \sim Q_{pr}$" indicates that the expectation is with respect to $(y^{(pr)}, z^{(pr)})$ drawn from Q_{pr}. We will now omit this subscript when there is no risk of ambiguity. For convenience, we also call the expression (20) the **evidence lower bound** and denote it by:

$$\begin{aligned}
&\sum_p \sum_r \mathrm{ELBO}(x^{(pr)}, Q_{pr}, \mu_p, \sigma_p^2, \nu_r, \tau_r^2) \\
&= \sum_p \sum_r \mathbb{E} \log \left[\frac{p(y^{(pr)}, z^{(pr)}, x^{(pr)}; \mu_p, \sigma_p^2, \nu_r, \tau_r^2)}{Q_{pr}(y^{(pr)}, z^{(pr)})} \right] \\
&= \sum_p \sum_r \mathbb{E} \Bigg[\log p(x^{(pr)} | y^{(pr)}, z^{(pr)}; \mu_p, \sigma_p^2, \nu_r, \tau_r^2) \\
&\qquad + \log p(y^{(pr)}, z^{(pr)}) - \log Q(y^{(pr)}, z^{(pr)})) \Bigg] \qquad (20)
\end{aligned}$$

Dropping the terms that do not depend on the parameters [1], we find that we need to maximize:

$$\begin{aligned}
&\sum_p \sum_r \mathbb{E}[\log p(x^{(pr)} | y^{(pr)}, z^{(pr)})] \\
&= \sum_p \sum_r \mathbb{E} \Bigg[\log \frac{1}{\sqrt{2\pi\sigma^2}} \exp \Bigg(-\frac{1}{2\sigma^2}(x^{(pr)} - y^{(pr)} - z^{(pr)})^2 \Bigg) \Bigg] \\
&= \sum_p \sum_r \mathbb{E} \Bigg[-\frac{1}{2}\log(2\pi\sigma^2) - \frac{1}{2}(x^{(pr)} - y^{(pr)} - z^{(pr)})^2 \Bigg]
\end{aligned}$$

From here, we can compute its partial derivative with respect to each of the parameters. We will start by computing a general version of these partial derivatives for any of the parameter $\{\mu_p, \sigma_p^2, \nu_r, \tau_r^2\}$ that will will denote “param”.

$$\begin{aligned}
&= \frac{\partial}{\partial \mathrm{param}} \sum_p \sum_r \mathbb{E} \Bigg[\log p(x^{(pr)} | y^{(pr)}, z^{(pr)}) \Bigg] \\
&= \sum_p \sum_r \frac{\partial}{\partial \mathrm{param}} \mathbb{E} \Bigg[-\frac{1}{2\sigma^2}(x^{(pr)} - y^{(pr)} - z^{(pr)})^2 \Bigg]
\end{aligned}$$

[1] see Lecture Notes 9 on Factor Analysis, page 7. I have not well understood this step. I have tried to compute the parameter updates without it and found that the only sensible way to perform the updates would be through an iterative method such as gradient descent. Normally, the function Q(y,z) is a function of the parameters param: $Q(param, param^{(t)} = E_{y,z|x,param^{(t)}}[\log L(param, y, z, x)]$

Now comes the question of whether or the partial derivative operator can be swapped with the expected value operator. If this were the case, the result would be zero which would imply that $p(y, z)$ and $Q(y, z)$ actually depended on the parameters. This would lead, as presented in [1] to very difficult parameter updates. We will therefore try to use the reparametrization trick [2] We write:

$$[y^{(pr)}, z^{(pr)}]^T = \mu_{y,z|x}^{(pr)} + \Sigma_{y,z|x}^{(pr)} \odot Z_{pr} \qquad \text{(where } Z_{pr} \sim \mathcal{N}(0, I_2))$$

Hence:

$$\begin{aligned} &\frac{\partial}{\partial \text{param}} \sum_p \sum_r \mathbb{E}_{(y^{(pr)}, z(pr)) \sim Q_{pr}} \left[\log p(x^{(pr)} | y^{(pr)}, z^{(pr)}) \right] \\ &= \frac{\partial}{\partial \text{param}} \sum_p \sum_r \mathbb{E}_{Z^{(pr)} \sim \mathcal{N}(0, I_2)} \left[\log p(x^{(pr)} \mid \mu_{y,z|x}^{(pr)} + \Sigma_{y,z|x}^{(pr)} \odot Z^{(pr)}) \right] \\ &= \sum_p \sum_r \mathbb{E}_{Z^{(pr)} \sim \mathcal{N}(0, I_2)} \left[\frac{\partial}{\partial \text{param}} \log p(x^{(pr)} \mid \mu_{y,z|x}^{(pr)} + \Sigma_{y,z|x}^{(pr)} \odot Z^{(pr)}) \right] \end{aligned} \tag{21}$$

No matter the approach I employ, the terms $\mu_{y,z|x}$ and $\Sigma_{y,z|x}$ will always make it hard to directly solve for the parameters. This leads me to conclude that an algorithm such as gradient ascent must be used to optimize the evidence lower bound over each parameter. The final algorithm would be:
For each parameter "param" of $\{\mu_p, \sigma_p^2, \nu_r, \tau_r^2\}$
Repeat until convergence{

(E-Step) For each, set:

[2] see Lecture Notes 8, The EM Algorithm, page 13

$$Q_{pr}(y^{(pr)}, z^{(pr)}) = p(y^{(pr)}, z^{(pr)} \mid x^{(pr)})$$
$$= \frac{1}{(2\pi)^2 |\Sigma_{y^{(pr)}, z^{(pr)}|x^{(pr)}}\ |^{\frac{1}{2}}}$$
$$\times \exp\left(-\frac{1}{2} \left[\begin{bmatrix} y^{(pr)} \\ z^{(pr)} \end{bmatrix} - \mu_{y^{(pr)}, z^{(pr)}|x^{(pr)}} \right]^T \Sigma^{-1}_{y^{(pr)}, z^{(pr)}|x^{(pr)}} \left[\begin{bmatrix} y^{(pr)} \\ z^{(pr)} \end{bmatrix} - \mu_{y^{(pr)}, z^{(pr)}|x^{(pr)}} \right] \right)$$

(M-Step) Repeat until convergence{

$$\text{param} = \text{param} + \alpha \sum_p \sum_r \mathbb{E}_{Z^{(pr)} \sim \mathcal{N}(0, I_2)} \left[\frac{\partial}{\partial \text{param}} \log\, p(x^{(pr)} \mid \mu^{(pr)}_{y,z|x} + \Sigma^{(pr)}_{y,z|x} \odot Z^{(pr)}) \right] \}$$

4. KL divergence and Maximum Likelihood

(a)

$$KL(P||Q) = \sum_x P(x) \log P(x)/Q(x)$$
$$= \mathbb{E}[-\log Q(x)/P(x)] \geq -\log \underbrace{\mathbb{E}[Q(x)/P(x)]}_{\in(0,1)} \geq 0 \tag{22}$$

We used Jensen's inequality for the convex function $f(x) = \log(x)$, the distribution $Q(x)/P(x)$.
Moreover, because $f(x) = -\log(x)$ is strictly convex, $\mathbb{E}[f(X)] = f(\mathbb{E}[X])$ implies $X = \mathbb{E}[X]$ with probability 1. Hence:

$$P = Q \Rightarrow \sum_x P(x) \log P(x)/Q(x) = 0 \Leftrightarrow KL(P||Q) = 0$$
$$KL(P||Q) = 0 \Leftrightarrow \mathbb{E}[-\log Q(x)/P(x)] = -\log \mathbb{E}[Q(x)/P(x)] = 0 \Rightarrow Q = P$$

Hence,

$$KL(P||Q) = 0 \Leftrightarrow Q = P \tag{23}$$

(b)

$$\begin{aligned}
KL(P(X,Y)||Q(X,Y)) &= \sum_x \sum_y P(x,y)\text{log }P(x,y)/Q(x,y) \\
&= \sum_x \sum_y P(x \mid y)P(y)\text{log }\frac{P(y \mid x)P(x)}{Q(y \mid x)Q(x)} \\
&= \sum_y P(y)\Bigg(\sum_x P(x \mid y)\text{log }\frac{P(y \mid x)}{Q(y \mid x)}\Bigg) \\
&+ \sum_x \underbrace{\sum_y P(x,y)}_{P(x)} \log \frac{P(x)}{Q(x)} \\
&= KL(P(Y \mid X), Q(Y \mid X)) + KL(P(X)||Q(X)) \qquad (24)
\end{aligned}$$

(c) Here, we use the fact that the mean of the empirical distribution is an unbiased estimator of the mean of the population distribution:

$$E_m[X] = \frac{1}{m}\big(\sum_{i=1}^{m} x_i\big) \qquad (25)$$

$$\begin{aligned}
\operatorname*{argmin}_\theta KL(\hat{P}||P_\theta) &= \operatorname*{argmin}_\theta \sum_x \hat{P}(x)\log\frac{\hat{P}(x)}{P_\theta(x)} \\
&= \operatorname*{argmin}_\theta \mathbb{E}\Bigg[\log\frac{\hat{P}(x)}{P_\theta(x)}\Bigg] \\
&= \operatorname*{argmax}_\theta -\frac{1}{m}\sum_{i=1}^{m}\log\frac{\hat{P}(x^{(i)})}{P_\theta(x^{(i)})} \qquad \text{(here, we used (25))} \\
&= \operatorname*{argmax}_\theta \frac{1}{m}\sum_{i=1}^{m}\log P_\theta(x^{(i)}) - \frac{1}{m}\sum_{i=1}^{m}\log\hat{P}(x^{(i)}) \\
&= \operatorname*{argmax}_\theta \frac{1}{m}\sum_{i=1}^{m}\log P_\theta(x^{(i)})
\end{aligned}$$

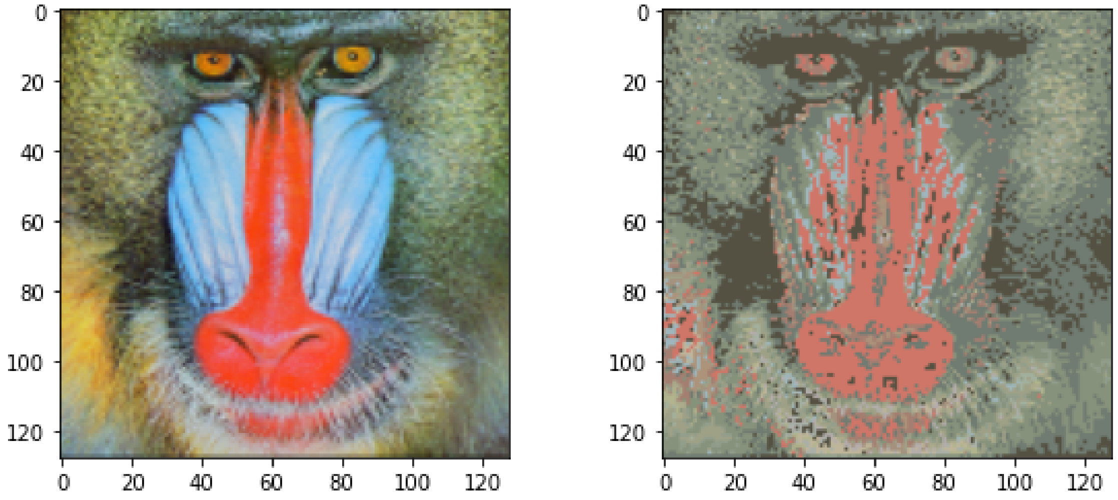
This proves that finding the maximum likelihood estimate for the parameter θ is equivalent to finding P_θ with minimal KL divergence from $\hat{P}$.

5. K-means for compression

If we represent the image with these 16 reduced colors, we have compressed the image approximately by a factor

$$\frac{16}{256} = 6.25\%$$

.



(a) Unompressed image

(b) Compressed image

Figure 2: Uncompressed and compressed image with k-means algorithm. It might be interesting to allocate more clusters near the rgb values most present in the image.