Engineering Tripos Part IIB FOURTH YEAR

Module 4F10: DEEP LEARNING AND STRUCTURED DATA

Solutions to Examples Paper 1

1. Average risk in choosing class ω_i is

$$\begin{aligned} R(\omega_i|\boldsymbol{x}) &= \sum_{j=1}^{c} \lambda(\omega_i|\omega_j) P(\omega_j|\boldsymbol{x}) \\ &= 0.P(\omega_i|\boldsymbol{x}) + \sum_{j=1, j\neq i}^{c} \lambda_s P(\omega_j|\boldsymbol{x}) \end{aligned}$$

where $\lambda(\omega_i|\omega_j)$ is used to mean the cost of choosing class ω_i where the true class is ω_j.

Hence

$$R(\omega_i|\boldsymbol{x}) = \lambda_s \left(1 - P(\omega_i|\boldsymbol{x})\right)$$

Associate $\boldsymbol{x}$ with class ω_i if highest posterior class probability and the average risk is less than the cost of rejection

$$\begin{aligned} \lambda_s \left(1 - P(\omega_i|\boldsymbol{x})\right) &\leq \lambda_r \\ P(\omega_i|\boldsymbol{x}) &\geq 1 - \lambda_r/\lambda_s \end{aligned}$$

If the ratio λ_r/λ_s is close to 1 then the reject region will tend to zero. If λ_r/λ_s is close to zero then nearly all examples will be rejected.

2. (a) A point $\boldsymbol{x}$ will lie on the decision boundary when the posterior of the two classes are 0.5. This will occur when

$$\exp(-(\mathbf{a}^{\mathsf{T}}\boldsymbol{x} + b)) = 1$$

Thus

$$\mathbf{a}^{\mathsf{T}}\boldsymbol{x} + b = 0$$

This is a linear decision boundary.

(b) The parameters can be trained to maximimse the likelihood of the correct label

$$\begin{aligned} \hat{\mathbf{a}}, \hat{b} &= \arg\max_{\mathbf{a},b} \left\{ \sum_{i=1}^{N} \log(P(y_i|\boldsymbol{x}_i, \mathbf{a}, b)) \right\} \\ &= \arg\max_{\mathbf{a},b} \left\{ \sum_{i:y_i=\omega_1} \log(\phi(\mathbf{a}^{\mathsf{T}}\boldsymbol{x}_i + b)) + \sum_{i:y_i=\omega_2} \log((1 - \phi(\mathbf{a}^{\mathsf{T}}\boldsymbol{x}_i + b))) \right\} \end{aligned}$$

There is no simple closed-form solution to find the parameters, so gradient descent based approaches are required (standard logistic regression training).

(c) The parameters can be trained to maximimse the likelihood of the correct label

$$\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} \left\{ \sum_{i:y_i=\omega_1} \log(\mathcal{N}(\boldsymbol{x}_i; \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)) + \sum_{i:y_i=\omega_2} \log(\mathcal{N}(\boldsymbol{x}_i; \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)) \right\}$$

There are simple closed-form solutions to this problem, maximum likelihood estimates for example

$$\hat{\boldsymbol{\mu}}_1 = \frac{\sum_{i:y_i=\omega_1} \boldsymbol{x}_i}{\sum_{i:y_i=\omega_1} 1}$$

To ensure that the resulting decision boundary is linear the covariance matrices must be constrained to be the same

$$\boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2 = \boldsymbol{\Sigma}$$

[Note you should be able to derive the estimate for this, there is still a closed form solution.]

3. (a) The factorization is

$$\begin{aligned} p(x_1, \ldots, x_5, y_1, \ldots, y_5) &= p(x_1)p(y_1|x_1)p(x_2|x_1)p(y_2|x_2)p(x_3|x_2)p(y_3|x_3) \\ &\quad p(x_4|x_3)p(y_4|x_4)p(x_5|x_4)p(y_5|x_5) \end{aligned}$$

(b) In a Bayesian network, any random variable (node in the graph) is conditionally independent of its non-descendants given its parents. In our case, we have the following set of conditional independences:

$$\begin{aligned} y_1 \perp x_2, x_3, x_4, x_5, y_2, y_3, y_4, y_5 | x_1 \\ y_2 \perp x_1, x_3, x_4, x_5, y_1, y_3, y_4, y_5 | x_2 \\ x_3 \perp x_1, y_1, y_2 | x_2 \\ y_3 \perp x_1, x_2, x_4, x_5, y_1, y_2, y_4, y_5 | x_3 \\ x_4 \perp x_1, x_2, y_1, y_2, y_3 | x_3 \\ y_4 \perp x_1, x_2, x_4, y_1, y_2, y_3, y_5 | x_4 \\ x_5 \perp x_1, x_2, x_3, y_1, y_2, y_3, y_4 | x_4 \\ y_5 \perp x_1, x_2, x_3, x_4, y_1, y_2, y_3, y_4 | x_5 \end{aligned}$$

(c) The conditional distribution $p(x_1, \ldots, x_5|y_1, \ldots, y_5)$ is obtained by Bayes rule from the ratio of the joint distribution $p(x_1, \ldots, x_5, y_1, \ldots, y_5)$ and the marginal distribution $p(y_1, \ldots, y_5)$. To compute the marginal $p(y_1, \ldots, y_5)$, we have to evaluate $p(x_1, \ldots, x_5, y_1, \ldots, y_5)$ on each possible value of $x_1, \ldots, x_5$ and sum the resulting probabilities. Therefore, since each of $x_1, \ldots, x_5$ can take ten different values, the computation of $p(x_1, \ldots, x_5|y_1, \ldots, y_5)$ in this case has cost 10^5.

(d) By making use of the factorization, we can iterate over the variables and sum the product of only those factors that depend on the current variable. In particular, we can compute

$$\begin{aligned} p(y_1,\ldots,y_5) &= \left(\sum_{x_5}\left(\sum_{x_4}\left(\sum_{x_3}\left(\sum_{x_2}\left(\sum_{x_1} p(x_1)p(y_1|x_1)p(x_2|x_1)\right)\right.\right.\right. \\ &\quad p(y_2|x_2)p(x_3|x_2)) \\ &\quad p(y_3|x_3)p(x_4|x_3)) \\ &\quad \left.\left.\left. p(y_4|x_4)p(x_5|x_4)\right)\, p(y_5|x_5)\right) \end{aligned}$$

Evaluating each of these sums involves, first, computing an auxiliary probability table and then summing over all the values of the current variable. For example, for the sum $\sum_{x_1} p(x_1)p(y_1|x_1)p(x_2|x_1)$ we first generate a probability table with entries equal to $p(y_1, x_1, x_2) = p(x_1)p(y_1|x_1)p(x_2|x_1)$, with one entry for each value of x_1 and x_2. This has cost 10^2. After this, we sum over x_1 and obain an updated probability table with entries equal to $p(y_1, x_2)$, with one entry for each possible value of x_2. The process repeates until the last sum for x_5, which involves generating a probability table with entries equal to $p(y_1, y_2, y_3, y_4, y_5, x_5)$ and summing over x_5, with cost 10. The total cost is then $4 \times 10^2 + 10$.

4. (a) The terms in the auxiliary function are

- $P(q_t = \mathtt{s}_j|\boldsymbol{x}_1,\ldots,\boldsymbol{x}_T;\boldsymbol{\theta})$ is the state posterior given the current model parameters are observation sequence
- $P(\boldsymbol{x}_t|\mathtt{s}_j,\hat{\boldsymbol{\theta}})$ is the probability of the observation given the "new" model parameters and state $\mathtt{s}_j$.

(b) From lecture notes

$$\begin{aligned} \alpha_j(t) &= \log\left(p(\boldsymbol{x}_1,\ldots,\boldsymbol{x}_t, q_t = \mathtt{s}_j)\right) \\ \beta_j(t) &= \log\left(p(\boldsymbol{x}_{t+1},\ldots,\boldsymbol{x}_T|q_t = \mathtt{s}_j)\right) \end{aligned}$$

Adding this expressions

$$\alpha_j(t) + \beta_j(t) = \log\left(p(\boldsymbol{x}_1,\ldots,\boldsymbol{x}_T, q_t = \mathtt{s}_j)\right)$$

This can be used to obtain the "normalisation" term

$$Z = \log\left(\sum_{j=2}^{N-1}\exp(\alpha_j(t) + \beta_j(t))\right) = \log\left(p(\boldsymbol{x}_1,\ldots,\boldsymbol{x}_T)\right)$$

The expression required can then be written as

$$P(q_t = \mathtt{s}_j|\boldsymbol{x}_1,\ldots,\boldsymbol{x}_T;\boldsymbol{\theta}) = \exp\left(\alpha_j(t) + \beta_j(t) - Z\right) = \gamma_j(t)$$

(c) The auxiliary function as

$$\mathcal{Q}(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}) = \sum_{t=1}^{T} \sum_{j=2}^{N-2} \gamma_j(t) \sum_{k=1}^{d} \left[x_{tk} \log(\hat{\lambda}_{jk}) + (1 - x_{tk}) \log(1 - \hat{\lambda}_{jk}) \right]$$

Differentiate this with respect to λ_{qr} give

$$\frac{\partial \mathcal{Q}(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}})}{\partial \hat{\lambda}_{qr}} = \sum_{t=1}^{T} \gamma_q(t) \left[\frac{x_{tr}}{\hat{\lambda}_{qr}} - \frac{(1 - x_{tr})}{(1 - \hat{\lambda}_{qr})} \right]$$

Equating to zero gives

$$(1 - \hat{\lambda}_{qr}) \sum_{t=1}^{T} \gamma_q(t) x_{ir} = \hat{\lambda}_{qr} \sum_{t=1}^{T} \gamma_q(t)(1 - x_{ir})$$

Rearranging yields the answer.

5. (a) The complete proof takes the form

$$p(\boldsymbol{x}) = \int p(\boldsymbol{x}|\boldsymbol{z}) p(\boldsymbol{z}) d\boldsymbol{z}$$

and solve. However as everything is Gaussian just need the first and second moments. Thus

$$\mathcal{E}\left\{\boldsymbol{x}\right\} = \boldsymbol{C}\mathcal{E}\left\{\boldsymbol{z}\right\} + \mathcal{E}\left\{\boldsymbol{v}\right\} = \boldsymbol{0}$$

and

$$\begin{aligned} \mathcal{E}\left\{\boldsymbol{x}\boldsymbol{x}^{\mathsf{T}}\right\} &= \boldsymbol{C}\mathcal{E}\left\{\boldsymbol{z}\boldsymbol{z}^{\mathsf{T}}\right\}\boldsymbol{C}^{\mathsf{T}} + \mathcal{E}\left\{\boldsymbol{v}\boldsymbol{v}^{\mathsf{T}}\right\} \\ &= \boldsymbol{C}\boldsymbol{C}^{\mathsf{T}} + \boldsymbol{\Sigma}_{\mathtt{diag}} \end{aligned}$$

Similarly

$$\mathcal{E}\left\{\boldsymbol{z}\right\} = \boldsymbol{0}; \quad \mathcal{E}\left\{\boldsymbol{x}\boldsymbol{z}^{\mathsf{T}}\right\} = \mathcal{E}\left\{(\boldsymbol{C}\boldsymbol{z} + \boldsymbol{v})\boldsymbol{z}^{\mathsf{T}}\right\} = \boldsymbol{C}\mathcal{E}\left\{\boldsymbol{z}\boldsymbol{z}^{\mathsf{T}}\right\} = \boldsymbol{C}$$

(b) Using the standard equality

$$\boldsymbol{\beta} = \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1} = \boldsymbol{C}^{\mathsf{T}}\left(\boldsymbol{C}\boldsymbol{C}^{\mathsf{T}} + \boldsymbol{\Sigma}_{\mathtt{diag}}\right)^{-1}$$

$$p(\boldsymbol{z}|\boldsymbol{x}; \boldsymbol{C}, \boldsymbol{\Sigma}_{\mathtt{diag}}) = \mathcal{N}\left(\boldsymbol{z}; \boldsymbol{\beta}\boldsymbol{x}, \boldsymbol{I} - \boldsymbol{\beta}\boldsymbol{C}\right)$$

(c) Substituting these expressions into the auxiliary function

$$\mathcal{Q}(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}) = \sum_{i=1}^{N} \int \mathcal{N}\left(\boldsymbol{z}; \boldsymbol{\beta}\boldsymbol{x}_i, \boldsymbol{I} - \boldsymbol{\beta}\boldsymbol{C}\right) \log\left(\mathcal{N}(\boldsymbol{x}_i; \hat{\boldsymbol{C}}\boldsymbol{z}, \hat{\boldsymbol{\Sigma}}_{\mathtt{diag}})\right) d\boldsymbol{z}$$

Expanding out the term in the log

$$\log\left(p(\boldsymbol{x}_i|\boldsymbol{z};\hat{\boldsymbol{\theta}})\right) = K_1 - \frac{1}{2}\log(|\hat{\boldsymbol{\Sigma}}_{\texttt{diag}}|) - \frac{1}{2}(\boldsymbol{x}_i - \hat{\boldsymbol{C}}\boldsymbol{z})^{\mathsf{T}}\hat{\boldsymbol{\Sigma}}_{\texttt{diag}}^{-1}(\boldsymbol{x}_i - \hat{\boldsymbol{C}}\boldsymbol{z})$$

Using the fact that $\hat{\boldsymbol{\Sigma}}_{\texttt{diag}}$ is diagonal yields for dimension k

$$-\frac{1}{2}\log(\hat{\sigma}_k^2) - \frac{1}{2\hat{\sigma}_k^2}\left(x_{ik}^2 - 2x_{ik}\hat{\boldsymbol{c}}_k\boldsymbol{z} + \hat{\boldsymbol{c}}_k\boldsymbol{z}\boldsymbol{z}^{\mathsf{T}}\hat{\boldsymbol{c}}_k^{\mathsf{T}}\right)$$

where $\hat{\boldsymbol{c}}_k$ is the k^{th} row of $\hat{\boldsymbol{C}}$. Replacing the values of $\boldsymbol{z}$ by their expected values

$$-\frac{1}{2}\log(\hat{\sigma}_k^2) - \frac{1}{2\hat{\sigma}_k^2}\left(x_{ik}^2 - 2\hat{\boldsymbol{c}}_k\boldsymbol{\beta}\boldsymbol{x}_i x_{ik} + \hat{\boldsymbol{c}}_k\left(\boldsymbol{\beta}\boldsymbol{x}_i\boldsymbol{x}_i^{\mathsf{T}}\boldsymbol{\beta}^{\mathsf{T}} + \boldsymbol{I} - \boldsymbol{\beta}\boldsymbol{C}\right)\hat{\boldsymbol{c}}_k^{\mathsf{T}}\right)$$

This can now be solved for $\hat{\boldsymbol{c}}_k$. Differentiating and equating to zero yields

$$\begin{aligned}
\hat{\boldsymbol{c}}_k^{\mathsf{T}} &= \left(\sum_{i=1}^{N}\mathcal{E}\{\boldsymbol{z}\boldsymbol{z}^{\mathsf{T}}|\boldsymbol{x}_i;\boldsymbol{\theta}\}\right)^{-1}\sum_{i=1}^{N}\mathcal{E}\{\boldsymbol{z}|\boldsymbol{x}_i;\boldsymbol{\theta}\}x_{ik} \\
&= \left(\sum_{i=1}^{N}\boldsymbol{\beta}\boldsymbol{x}_i\boldsymbol{x}_i^{\mathsf{T}}\boldsymbol{\beta}^{\mathsf{T}} + \boldsymbol{I} - \boldsymbol{\beta}\boldsymbol{C}\right)^{-1}\sum_{i=1}^{N}\boldsymbol{\beta}\boldsymbol{x}_i x_{ik}
\end{aligned}$$

Can now solve for $\hat{\boldsymbol{\Sigma}}_{\texttt{diag}}$

$$\begin{aligned}
\hat{\sigma}_k^2 &= \frac{1}{N}\left(\sum_{i=1}^{N}x_{ik}^2 - 2\hat{\boldsymbol{c}}_k\mathcal{E}\{\boldsymbol{z}|\boldsymbol{x}_i;\boldsymbol{\theta}\} + \hat{\boldsymbol{c}}_k\mathcal{E}\{\boldsymbol{z}\boldsymbol{z}^{\mathsf{T}}|\boldsymbol{x}_i;\boldsymbol{\theta}\}\hat{\boldsymbol{c}}_k^{\mathsf{T}}\right) \\
&= \frac{1}{N}\left(\sum_{i=1}^{N}x_{ik}^2 - 2\hat{\boldsymbol{c}}_k\boldsymbol{\beta}\boldsymbol{x}_i x_{ik} + \hat{\boldsymbol{c}}_k\left(\boldsymbol{\beta}\boldsymbol{x}_i\boldsymbol{x}_i^{\mathsf{T}}\boldsymbol{\beta}^{\mathsf{T}} + \boldsymbol{I} - \boldsymbol{\beta}\boldsymbol{C}\right)\hat{\boldsymbol{c}}_k^{\mathsf{T}}\right)
\end{aligned}$$

(d) For a full covariance matrix model the number of model parameters is $d(d+1)/2$. If this is modelled with a p-dimenensional latent variable FA model the number of parameters is $d\times(p+1)$. By controlling p it is possible to control the number of parameters, whilst still modelling correlations in the data.

If p is set to 2 or 3 it is possible to use FA for visualisation. Given an observation $\boldsymbol{x}$, it is possible to get the Gaussian distribution of the latent variable $p(\boldsymbol{z}|\boldsymbol{x};\boldsymbol{\theta})$. The mean of this distribution can be used as a low-dimensional projection of the point $\boldsymbol{x}$.

6. Total number of weights in the system is

- input to hidden layer: $(d+1)M$
- the $L-1$ hidden to hidden: $(L-1)M(M+1)$
- hidden to output $(M+1)K$

The number of hidden layers determines the decision boundaries that can be produced (see lecture notes), the activation function determines the nature of the output - binary (step), sum to one (soft max), continuous (linear) etc. The number of hidden units should be large enough to model the problem, but small enough so that *generalisation* is not an issue.

7.

$$\phi(z) = \frac{1}{1+\exp(-z)}, \quad \begin{aligned} \frac{\partial \phi(z)}{\partial z} &= \frac{\exp(-z)}{(1+\exp(-z))^2} \\ &= \phi(z)(1-\phi(z)) \end{aligned}$$

The activation function affects the form of the error back propagation algorithm. The derivation given in lectures assumes a sigmoid, however the output layer can be more complex if a sum squared error is used with a softmax function (not not if used with a cross-entropy measure) since in this case the partial derivative for a particular weight in the output layer depends on all output values due to the normalisation in the softmax.

8. Need to compute the first and second moments. First moment given by

$$\int_{-\infty}^{\infty} \phi(x)p(x)dx = \alpha \int_{-\infty}^{0} xp(x)dx + \int_{0}^{\infty} xp(x)dx = \int_{0}^{\infty} xp(x)dx - \alpha \int_{0}^{\infty} xp(x)dx$$

It is possible to show that when $p(x) = \mathcal{N}(x; 0, \sigma^2)$

$$\int_{0}^{\infty} xp(x)dx = \frac{\sigma}{2}\sqrt{\frac{2}{\pi}}$$

Hence

$$\int_{-\infty}^{\infty} \phi(x)p(x)dx = (1-\alpha)\frac{\sigma}{2}\sqrt{\frac{2}{\pi}} = (1-\alpha)\sigma\sqrt{\frac{1}{2\pi}}$$

and the second moment

$$\int_{-\infty}^{\infty} (\phi(x))^2 p(x)dx = \int_{-\infty}^{0} \alpha^2 x^2 p(x)dx + \int_{0}^{\infty} x^2 p(x)dx = (1+\alpha^2)\sigma^2/2$$

So the total variance on the output is

$$\hat{\sigma}^2 = (1+\alpha^2)\sigma^2/2 - (1-\alpha)^2\frac{\sigma^2}{2\pi}$$

The simplest approach is to ensure that the output variance matches the input variances for the initialisation (as discussed in lectures). This function has an added complexity as the mean is non-zero. If the network is deep this could result in a large offset for some layers. This could be addressed by considering an offset on the bias term initialisation, but is usually ignored.

9. (a) The Hessian may be used to obtain the Newton direction. This requires computing $\mathbf{H}^{-1}\mathbf{g}$. (see lecture notes for more details).

   (b)

$$\frac{\partial E}{\partial w_{ij}} = \sum_{p=1}^{n}(y(x_p) - t(x_p))\frac{\partial y(x_p)}{\partial w_{ij}}$$

   and

$$\frac{\partial^2 E}{\partial w_{ij}\partial w_{lk}} = \sum_{p=1}^{n}\frac{\partial y(x_p)}{\partial w_{lk}}\frac{\partial y(x_p)}{\partial w_{ij}} + \sum_{p=1}^{n}(y(x_p) - t(x_p))\frac{\partial^2 y(x_p)}{\partial w_{ij}\partial w_{lk}}$$

   For the conditions described the network will train so that

$$y(x_p) = t(x_p)$$

   In this condition the second term is zero.

   (c) From the conditions given

$$\mathbf{H}_{N+1} = \mathbf{H}_N + \mathbf{g}^{(N+1)}(\mathbf{g}^{(N+1)})^\mathsf{T}$$

   Consider the inverse

$$\begin{aligned}
\mathbf{H}_{N+1}^{-1} &= \left(\mathbf{H}_N + \mathbf{g}^{(N+1)}(\mathbf{g}^{(N+1)})^\mathsf{T}\right)^{-1} \\
&= \mathbf{H}_N^{-1} - \mathbf{H}_N^{-1}\mathbf{g}^{(N+1)}\left(1 + \mathbf{g}^{(N+1)\mathsf{T}}\mathbf{H}_N^{-1}(\mathbf{g}^{(N+1)})\right)^{-1}(\mathbf{g}^{(N+1)})^\mathsf{T}\mathbf{H}_N^{-1} \\
&= \mathbf{H}_N^{-1} - \frac{\mathbf{H}_N^{-1}\mathbf{g}^{(N+1)}(\mathbf{g}^{(N+1)})^\mathsf{T}\mathbf{H}_N^{-1}}{1 + \mathbf{g}^{(N+1)\mathsf{T}}\mathbf{H}_N^{-1}(\mathbf{g}^{(N+1)})}
\end{aligned}$$

   The calculation of the inverse can be computationally expensive for large numbers of weights (naive implementation $\mathcal{O}(W^3)$). This scheme directly calculates the inverse. An initial value is needed for this scheme ($\mathbf{H}_0$). The simplest approach is to use a diagonal matrix with very small values on the leading diagonal (easy to invert and will not distort the final results).

M.J.F. Gales & J. M. Hernandez-Lobato
Oct 2017