STA4026S Analytics – Neural Networks.

Assignment 2 2025 (30 Marks)

Collider Data

You are a Data Scientist working at a particle collider. The collider operates by sending proton beams in opposite directions through a tunnel loop and observing the post-impact scattering of sun-atomic particles after beams collide at a detector location. Your team is tasked with classifying rarely observed sub-atomic particles (given experimental codenames) as they scatter after impact. You are given data from a cross-section (a slice of the collision path) where scientists have been able to classify, 'by hand', 360 such particles. You are tasked with building a neural network which automates this classification process based on these data.

Variable	Description
X1	First coordinate in cross-section.
X2	Second coordinate in cross-section.
X3	Detector type. Encoded as 1 for Type A and 0 for type B.
Y1	Response: 1 if code-α particle, 0 otherwise.
Y2	Response: 1 if code-β particle, 0 otherwise.
Y3	Response: 1 if code-ρ particle, 0 otherwise.

```
> dat = read.table('Collider Data 2025.txt', h = TRUE,stringsAsFactors =TRUE)
> dim(dat)
```

```
[1] 360   5
```

```
> head(dat) # First five observations are all of class 1.
```

```
           X1          X2 X3 Y1 Y2 Y3
1   3.2509506 -0.2478462  0  1  0  0
2   0.1130716 -2.6077716  1  1  0  0
3   2.0345077 -2.1380898  0  1  0  0
4 -1.6595216 -1.7533739  1  1  0  0
5   3.1474216  0.4139403  1  1  0  0
```

For purposes of modelling the present dataset, we consider a special kind of neural network that's totally not just made up for purposes of this assignment called an **Auto-Feature**-netork or AFnetwork for short. The premise for this network is that it automatically augments (adds to) the feature space with transformed inputs (this is typically called feature engineering). One augmented input node is added for each feature. These are calculated as the result of a fully connected hidden layer in the usual fashion. Figure 1 gives the directed graph for this species of network.

Use the questions that follow to conduct the analysis.

Input X → (augmented layer) → Hidden Layer 1 → Hidden Layer 2 → Output Layer (with softmax)

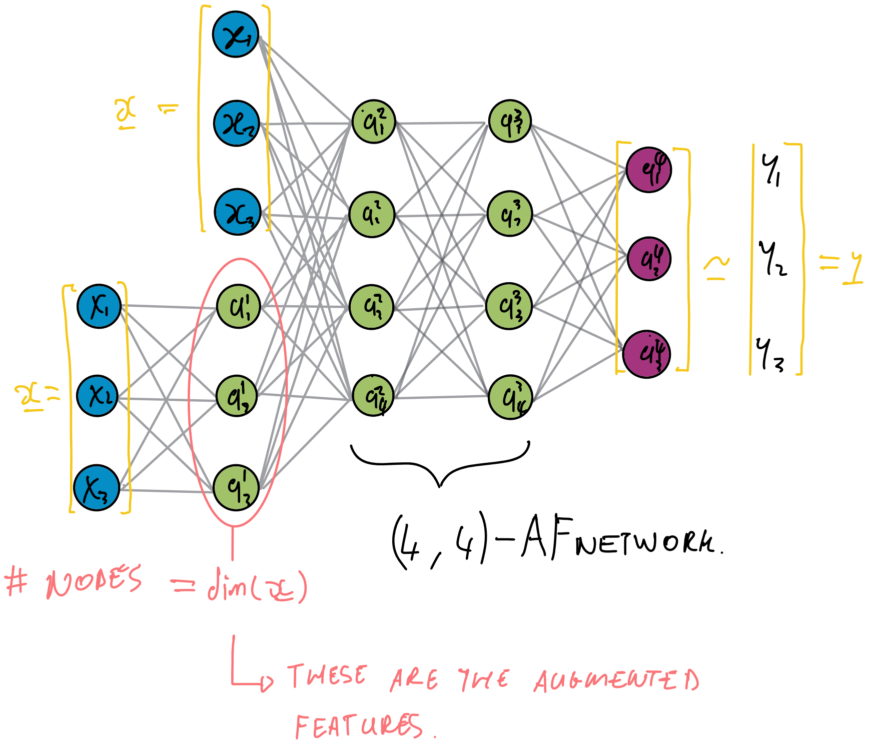

Figure 1: Directed graph of a (4,4)='Auto-Feature'-network. Note that I've left out the biases in the graph to make the structure legible but each node does have a bias term.

Note: I've added leading indicators to each question telling you what should be reflected in your compiled write-up. To that end, here's what they mean explicitly:

- Render code: This means, I want to see your R code typeset at that point in your document. So make sure the code-chunk options are set such that the code renders.
- Render plots: This means, I want to see a plot rendered in the write up.
- In-text response: This means I want a paragraph-style response where you type out your response in one or a few sentences.
- Combinations of the above: If you see R code + plots, that means render the R code and the plot, and no typed response is required. If you see R code, that means just render the R code and no other response is required.

Problem Set

1. (a) **Render code + plot**: Plot the coordinates in the feature space and colour-code according to the response. Comment on whether non-linear machinery such as a neural network is an appropriate model class for the present problem. (3)

   (b) **Render code**: Write an R-function which evaluates the soft-max activation function in matrix form. The scalar form of the soft-max activation function is given by: (2)

$$\sigma(z_j) = \frac{\exp(z_j)}{\sum_{i=1}^{3} \exp(z_i)},$$

   for a three-class problem. Hints:

   - In matrix-form the resulting calculation will be of dimensions $3 \times N$.
   - There are at least two ways in which this can be achieved.
   - `apply()`
   - `colSums()`

   (c) The cross-entropy error function for a polytomous response $\mathbf{y}_i^T = [y_{i1}, y_{i2}, y_{i3}]$ is given by the following expression: (2)

$$\begin{aligned}
\text{Obj} &= -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{3} y_{ij}\log(\hat{y}_{ij}) \\
&= \frac{1}{N}\sum_{i=1}^{N}(y_{i1} \times -\log(\hat{y}_{i1}) + y_{i2} \times -\log(\hat{y}_{i2}) + y_{i3} \times -\log(\hat{y}_{i3})) \\
&= \frac{1}{N}\sum_{i=1}^{N} C_i
\end{aligned}$$

   where $\mathbf{y}^T \in \{[1, 0, 0], [0, 1, 0], [0, 0, 1]\}$.
   Render typed response: Write C_i as an if-statement/function with case patterns and explain why it may be advantageous for numerical purposes (for purposes of evaluating the objective function in a machine) to evaluate only the terms corresponding to $y_{i.} = 1$.

   (d) **Render code**: Write an R-function, `g(Yhat,Y)`, that evaluates the aforementioned objective. Full marks only for evaluation which excludes the possibility of evaluating any $0 \times \log(\hat{y}_{i.})$ whilst not using any if-statement or looping through an if-statement. (2)

   (e) **Render typed response**: Write down an expression for calculating the number of parameters in an (m,m)-AFnetwork with $\dim(\mathbf{x}) = p$ and $\dim(\mathbf{y}) = q$. (2)

   (f) **Render code**: Write an R-function that evaluates a forward pass of an (m,m)-AFnetwork. Use tanh activations on all hidden nodes (including the augmentation layer. This function must also include evaluation of an appropriate objective function and regularisation mechanism for the present dataset (let ν denote the regularization parameter and use the L2-norm). (5)

   Failover: If you cannot figure out (b) and (c) you may use a standard (m, m)-network to conduct the analysis but you will be forgoing some marks for that.

In any case, see if you can get the analysis to work for the standard case first. That might be useful for saving time.

(g) **Render code + plot + in text response**: Conduct a validation analysis for $m = 4$ using standard R optimization routines in order to fit the models. Note: no scaling is required for the input variables here. Use `set.seed(2025)` before splitting the data into 80-20 training vs. validation. Plot the validation error vs. ν and use this figure to motivate your choice of regularization level. Indicate the level of regularisation chosen and report the value. `exp(seq(-6,2,length = n_val))` may be useful but experiment until you get the appropriate curve. (4)

(h) **Code + plot**: Use the regularised model to construct response curves over the input variables. Use these plots to determine if there is any difference in the predicted response from the respective detector types? (5)

(i) **In-text response**: Are there any practical advantages you can envisage for using an AFnetwork over standard feedforward neural networks? Clearly motivate your response. (2)

(j) Three marks for the quality of your write-up. These are (some of) the things I'll be looking for: (3)

- Are your figures 1-1 aspect ratio?
- Are your figure captions full sentences and punctuated?
- Are your equations punctuated and properly typeset? Equations should read as sentences and thus should be punctuated.

Notes, Instructions, Etc.

Submission instructions:
For submission, I'd like you to give me

- A pdf containing your write-up. The write-up has an 8-page limit (excluding appendix). Use the naming convention `STDNUM001_STDNUM002_Analytics_2025_A2.pdf` for your file. Note the underscores.

- Your R code. Use the naming convention `STDNUM001_STDNUM002_Analytics_2025_A2.R` for your code file. Your R code should NOT contain any of the following:

  ```
  install.packages()
  rm()
  setwd()
  ```

  I want to be able to run your code on my computer without having to manually edit your code, installing libraries or calling to external files.

ANY deviation from the above conventions WILL be penalised. Doing so wastes my time and I've grown quite weary of such simple instructions being ignored at my expense.