**Inf2B Coursework 1 Report**

**Task 1 – Anuran-Call analysis and classification**

**1.2) Findings from the correlation matrix**  
Include graphs

**1.3) b) Graph of cumulative variance**  
Include graphs

**1.3) c) Plotting of data on 2D-PCA plane**  
Graph must clarify different classes.

**1.4) Accuracy for CovKind = 1,2,3**Accuracy = correct classification rate.

**1.5) A graph of classification accuracy vs epsilon value**  
Include graphs

**Task 2 – Neural networks**

A close up of a map

Description automatically generated**2.3) Structure of the neural network**

**How weights were determined:**  
Weights are created in order to classify data correctly using the activation function (hNeuron). This classification is dependent on the coordinate of a point being within the boundaries of polygon A or not. Thereby we must calculate the functions of these boundaries and find how they can be converted to appropriate weights.

Given we are calculating weights we must ensure to account for every variable:

The sign of Y is negative after isolating all terms to one side of the equation, and the default coefficient of Y has a constant magnitude of 1 so we only need to calculate the gradient and y-intercept.

So firstly, I created a function *task2\_calcGrads(x)* which calculates the constants for all boundary functions of a given polygon x. It takes an input of a polygon coordinate matrix and produces an output of a boundary function matrix (where each row represents a boundary function and each column represents the associated gradient and y-intercept respectively). Implementing this function promotes much higher accuracy for boundary calculations than mere hardcoding.

Now, given we have the respective boundary function constants we must identify which boundaries are the maxima and minima of this polygon. This is very relevant as it denotes which side of the boundaries represents the polygon. I identified these maxima and minima by capitalizing on the order of polygon vertex input, in which the maximum Y vertex is always first and the other vertices follow in a clockwise/anti-clockwise fashion. In order to represent these maxima and minima boundaries

rearrange them in a way that isolates a 0 (puts all terms on one side of the function). This is useful as it allows us

Initial weights are calculated to represent all boundaries of polygon A.

2.

Diagrams repr. The structure of the neural network.  
Explanation on how I determined the weights.

**2.10) Findings from the correlation matrix**  
Investigation: difference in decision boundary calculations for hNeuron (step function) and sNeuron (sigmoid function).