Ralph Mora

Assignment 6.1

ECE 495

First, we represent our data with a = 0.2, This is easily linearly separable. but when a = 1.5, we cannot linearly classify each class.

A screenshot of a computer

Description automatically generated A screenshot of a computer

Description automatically generated

Figure 1 – Data with a = 0.2 Figure 2 – Data with a = 1.5



To classify Figure 2, we must transform it from a set of 2-dimensional data points, into a multidimensional set of vectors. Then we can use the kernel trick to linearly solve our non-linear datapoints. To represent our data as multi-dimensional vectors, we will use a 3rd order Volterra Expansion.

Order 0 = 1

1st Order = x[n], x[n-1]

2nd Order = x2[n], x2[n-1], x[n]x[n-1]

3rd Order = x3[n], x3[n-1], x2[n]x[n-1], x[n]x2[n-1]

Then we put all these elements together and that gives us a vector in a 10-dimensional Hilbert space. Then we program a linear estimator using the function = wT(xn), where (xn) represents our 10-dimensional vectors, and is the predicted y value. wT is linear in the function so we can find the mean squared error and minimize it.

w =( *T )*-1

Finally, we plot the points wT\*(xn) = 0, and we can now see the boundaries that non-linearly classify all the points.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Although this method did a great job classifying each group. Multiplying a 10-element vector takes considerably more time to process.

P = = 10

P = = 30

So, if we increase the input space to 4 dimensions and use a third order Volterra expansion we need 30 elements now, and for 5 dimensions we will need 56 elements. This is known as the curse of dimensionality and because of this there are computational limitations to this method.