## Results from NoNoise

* Results from trying to predict class of question asked
* Bursts of packets were collected and then the following features were extracted from these bursts:
  + Minimum packet length, max, median and mean length
  + For each IP address in any burst: whether or not this burst included that IP
  + For each protocol in any burst: whether or not this burst included that protocol
* 5 classes were considered:
* Time:
  + “Alexa what’s the time”
  + “Aelxa what is the time?”
  + Etc
* Weather
  + “Alexa what’s the weather like in…”
  + “Alexa, what’s the weather in…”
  + For 10 different city names
* Joke
  + “Alexa tell me a joke”
* Song author
  + “Alexa who sings…”
  + “Alexa who composed…”
  + For 4 different song/piece titles
* Conversions:
  + “Alexa, what is 35F in Celsius”
  + “Alexa, how many gallons in a pint”
  + “Alexa, what is 3ft in cm”
  + “Alexa, how many pounds are there in 3 kilos”
* Each of these was recorded and played back so that each question was asked at least 6 times
* Note: the predominant class was weather as more examples of this were recorded than anything else; but it is not too predominant
* Then basic neural nets were used to try to predict the class
* I recorded the train and test accuracies after a reasonable number of epochs, with a random 1/3 of the data set aside as test
* I also recorded the confusion matrix on the testing data each time
* With 10 hidden nodes and only mean packet length and median length as features:
* With only time and weather:
  + Epoch = 1000, train accuracy = 94.62%, test accuracy = 93.48%
  + [[ 16. 2.]
  + [ 1. 27.]]
* Time, weather and joke:
  + Epoch = 500, train accuracy = 80.00%, test accuracy = 74.55%
  + [[ 16. 4. 1.]
  + [ 0. 24. 5.]
  + [ 0. 4. 1.]]
* Time, weather, joke and sings:
  + Epoch = 500, train accuracy = 67.97%, test accuracy = 71.88%
  + [[ 14. 1. 1. 8.]
  + [ 0. 32. 6. 1.]
  + [ 0. 1. 0. 0.]
  + [ 0. 0. 0. 0.]]
* All five:
  + Epoch = 500, train accuracy = 59.31%, test accuracy = 62.50%
  + [[ 13. 3. 0. 11. 5.]
  + [ 0. 30. 5. 0. 3.]
  + [ 0. 0. 2. 0. 0.]
  + [ 0. 0. 0. 0. 0.]
  + [ 0. 0. 0. 0. 0.]]
* With 10 hidden nodes and only mean packet length and median length as features, with 25% dropout:
  + Epoch = 500, train accuracy = 59.31%, test accuracy = 62.50%
  + [[ 13. 3. 0. 11. 5.]
  + [ 0. 30. 5. 0. 3.]
  + [ 0. 0. 2. 0. 0.]
  + [ 0. 0. 0. 0. 0.]
  + [ 0. 0. 0. 0. 0.]]
* With 10 hidden nodes, all features and no dropout:
* Only time and weather:
  + Epoch = 500, train accuracy = 96.77%, test accuracy = 93.48%
  + [[ 16. 2.]
  + [ 1. 27.]]
* All five:
  + Epoch = 500, train accuracy = 66.21%, test accuracy = 58.33%
  + [[ 12. 4. 0. 10. 4.]
  + [ 1. 27. 5. 0. 2.]
  + [ 0. 2. 2. 0. 0.]
  + [ 0. 0. 0. 1. 2.]
  + [ 0. 0. 0. 0. 0.]]
* Hence, when all 5 categories are considered the train accuracy is marginally better than with only packet length statistics, but the test accuracy has dropped
* The telling one is the following:
* With 10 hidden nodes, all features except average length and median length and no dropout:
* On two categories
  + Epoch = 500, train accuracy = 69.89%, test accuracy = 65.22%
  + [[ 1. 0.]
  + [ 16. 29.]]
* On five:
  + Epoch = 500, train accuracy = 47.59%, test accuracy = 40.28%
  + [[ 0. 3. 0. 1. 0.]
  + [ 13. 28. 7. 9. 6.]
  + [ 0. 0. 0. 0. 0.]
  + [ 0. 0. 0. 1. 2.]
  + [ 0. 2. 0. 0. 0.]]
* As we can see, without the mean and median length of packets the accuracy is significantly worse and from the confusion matrices we can see that the nets simply predict the most prevalent burst category in the data set (which is the weather)
* This leads me to provisionally conclude that the Echo simply sends all the recorded command to its servers, and then sends back all the voice that Alexa will speak back
* Further research into this is necessary to see if one can, by studying simply the statistics of the lengths of packets in a burst, accurately predict the length of spoken command and response by both the person using the device and Alexa’s response