A4

Part I: 20 points each

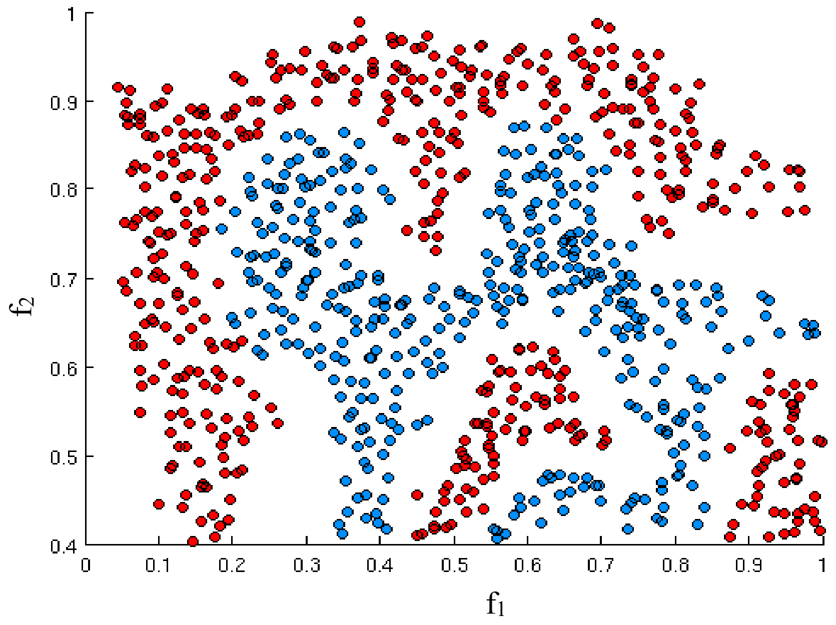
1. Consider the following data taken from a cognitive experiment on learning (Pazzani 1991 *J. Exp. Psych: Learning, Memory, and Cognition, 416-422)* to predict whether or not someone could inflate a balloon. There are small and large yellow and purple balloons that have either been dipped in water or stretched. A child or adult then attempted to inflate the balloon after being given instructions on how to blow up a balloon. The table below shows the results of the experiment:

	Color	Size	Preparation	Age	Success
0	YELLOW	SMALL	STRETCH	ADULT	Yes
1	YELLOW	SMALL	STRETCH	CHILD	Yes
2	YELLOW	SMALL	DIP	ADULT	Yes
3	YELLOW	SMALL	DIP	CHILD	Yes
4	YELLOW	SMALL	STRETCH	ADULT	Yes
5	YELLOW	SMALL	STRETCH	CHILD	Yes
6	YELLOW	SMALL	DIP	ADULT	Yes
7	YELLOW	SMALL	DIP	CHILD	Yes
8	YELLOW	LARGE	STRETCH	ADULT	No
9	YELLOW	LARGE	STRETCH	CHILD	No
10	YELLOW	LARGE	DIP	ADULT	No
11	YELLOW	LARGE	DIP	CHILD	No
12	PURPLE	SMALL	STRETCH	ADULT	No
13	PURPLE	SMALL	STRETCH	CHILD	No
14	PURPLE	SMALL	DIP	ADULT	No
15	PURPLE	SMALL	DIP	CHILD	No
16	PURPLE	LARGE	STRETCH	ADULT	No
17	PURPLE	LARGE	STRETCH	CHILD	No
18	PURPLE	LARGE	DIP	ADULT	No
19	PURPLE	LARGE	DIP	CHILD	No

   Compute the information gain (show your work) for each attribute (question) and determine which question should be asked first in a decision tree.

2. Assume a 0-1 loss rule and a test set of examples: $e_1, e_2, \ldots, e_N$ with known labels $l_1, l_2, \ldots, l_N$. Assume that we have learned a function h* such that $h^*(e_i) = \hat{l}_i$ where $\hat{l}_i$ is the predicted label. Write pseudocode for function estimate loss (examples, labels, h*) that determines the empirical loss (also known as the empirical risk) of a test set of N examples.

3. Given the two-feature data set below where two classes are distinguished by color,

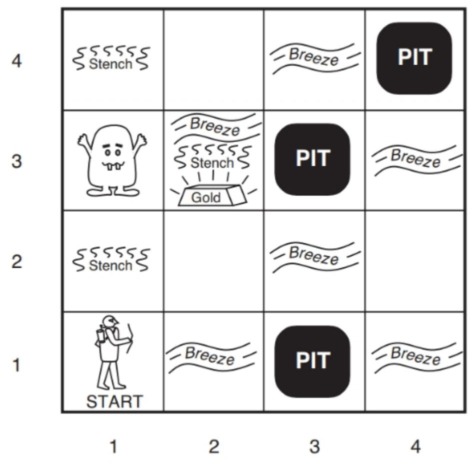


would a single neuron neural net be more likely to have high bias or high variance? Justify your answer.

4. A world model consists of three variables, X, Y, and Z, which each have domains of True and False.

   a. How many worlds are possible?
   b. Suppose our knowledge base consists of X = False, Z = True. Considering all worlds in this model, is $\neg X \wedge Y$ entailed?

5. Given the wumpus world of Figure 7.2 from your book where position (x, y) denotes the cave in column x, row y (e.g., wumpus is at 1,3):



   a. What are the percepts for an agent at 2,3?

   b. Write a set of logical sentences (see section 7.4.3) that describe the predicates that can be inferred by percept breeze at 3, 2 and a knowledge base consisting of the rules of Wumpus world. To keep it relatively simple, do not include knowledge from any other positions that the agent may have picked up while moving to 3,2 that could infer more details.

6. In addition to what we know about the starting conditions of a wumpus world, assume that a stench has been observed at locations (2,1) and (1,2). Write the knowledge base in conjunctive normal form and show whether or not the question $W_{2,2}$ (there is a wumpus at location 2,2) is entailed by the knowledge base using the resolution rule.

Part II – Dolphin identification (140 points)

The code package for this assignment on Canvas contains support functions for developing a neural network using Tensorflow. Instructions for setting up Tensorflow are included in the code package. Data (29 MB) are provided to distinguish acoustically between two species of dolphins (Figure 1): Pacific white-sided dolphins and Risso's dolphins.

Figure 1 – Pacific white-sided dolphin (Lagenorhynchus obliquidens, left) and Risso's dolphin (Grampus griseus, right). Photos: Marine Biaoacoustics Research Collaborative

Dolphins produce echolocation clicks for several purposes including navigation and finding food. The clicks are highly influenced by the animals' biology; the melon (forehead) acts as an acoustic lens to focus sound energy. We showed that these two species could be distinguished by analyzing their echolocation clicks (Soldevilla *et al.*, 2008). The data provided to you contain representations of echolocation clicks recorded at seven locations off the coast of Southern California. The data in this assignment are from a study that showed how instrumentation and recording location could influence performance (Roch *et al.*, 2015). You will be replicating a small subset of the experiments from this study. If you are curious about the cepstral features used in this study, see Roch *et al.* (2011).

The focusing mechanism of the melon makes recordings of echolocation clicks highly sensitive to the angle between the animal's longitudinal axis and the location of the recording device (Figure 2). As a consequence, while we can learn species identity from training data that consists of individual clicks, during classification we usually need to make decisions based on a group of clicks in order to have accurate classification.

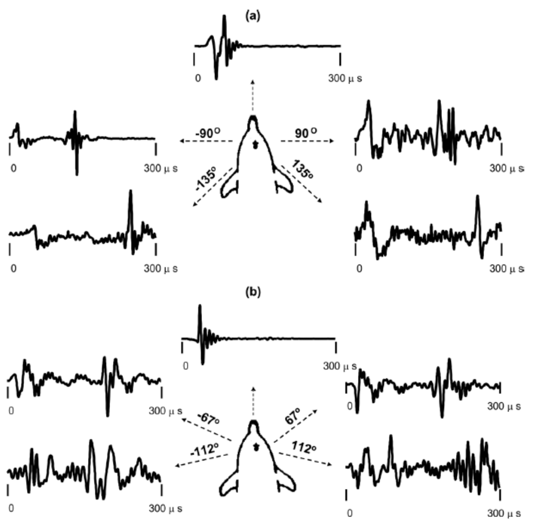


Figure 2 – Echolocation clicks measured at different angles (Fig 2, Au et al., 2012)

In this assignment, you will create a neural network to classify 100 clicks at a time to species. Echolocation clicks have been extracted and processed with a noise compensation routine as described in Roch *et al.* (2015). Files are organized into a directory containing the Risso's dolphins clicks (Gg, the first letter of each word in the Latin species name), and another one for the Pacific white-sided dolphin clicks (Lo). The Pacific white-sided dolphin directory is further subdivided, but that is not relevant to this assignment.

There are a number of functions that are provided to you that will be helpful in this effort. Module lib.file_utilities contains two functions that you will need to call. The first is get_files. Given a directory, it will recursively explore the directory for files with a specific extension which can be overridden. It also supports a stop_after keyword which lets you specify a maximum number of files to return. During the development cycle of your software, this feature is worth using as it will enable you to load small amounts of data quickly. You may not produce a good model when you use this, but it is nearly always a good idea to get everything working first with a small data set before training on something that will take a little while to complete.

The second function is called parse_files. Given a list of filenames, parse_files extracts information about when and where the data were recorded, the start time of the data, the species that were recorded, and a numpy 2D tensor (matrix) of features where each row represents an echolocation click. Parse files returns a list of named tuples that can be accessed with an attribute name instead of an index (e.g., .features, see function documentation for details on attributes).

Data should be split into training and test data separately for each species. The criterion for splitting is that all data from a given day must be entirely in the training data or entirely in the test data. Module lib.partition contains a function that can be used to create a dictionary whose keys are days and values are a list of all of the recording sessions that started on that day. Splitting data from the same animal group in the same behavioral state with generally the same direction to the recorder makes the task easier than it should be, which is why we do not want two sessions that are close in time to be partially in the training data and partially in the test.

Neural networks should have a relatively balanced number of examples for each category. A simple way to ensure somewhat balanced data between the species with the above criterion is to load and parse the files for each species separately. You can also use class weighting dictionary that is addressed at the end of this assignment. Splitting the data for each species will result in two dictionaries, one for Risso's dolphins and one for Pacific white-sided dolphins. The keys of each of these are days and you can use a function to partition the dictionary keys (convert them to a list first) into training days and test days. Scikit learn has a function train_test_split that can be used to split the keys into training keys and test keys. By default, this uses a 70% training, 30% test split. Gather all of the list elements associated with the date keys for both species into lists for training and testing.

The next step is to prepare data to train your model. You will need to construct example and label tensors for all the days in the training data. The list items you previously constructed will have attributes for features and label that contain the echolocation click features and the Gg/Lo label code. Numpy's array manipulation routines are excellent for this, and you might want to consider looking at concatenate, hstack, or vstack for this purpose.

Train a model with these data. With the splitting criterion of any given day not being split across the train/test boundary, this task should be relatively easy for a neural network and your choice of network is not critical to having good performance. You must use some form of regularization in your network, such as an L2 regularizer. Regularization of weights in keras is done by setting the kernel_regularizer keyword argument in Dense nodes to a regularization object. As these data are relatively easy to classify, and the data set is relatively large, you should not need more than a few epochs of training.

A good starting point might be a four-layer feed forward network with 100 nodes each and an L2 regularizer with a value of 0.01 (you can probably improve this by increasing the network capacity). Don't be surprised if you do not see wonderful classification performance on the training data; predicting species from one highly-variable echolocation click is a very difficult problem.

Predicting species from more than one echolocation clicks yields much better results. During test, you will process each feature set that is in the test data. After you have your predictions on each click, group the probabilities into groups of 100 predications and compute their joint likelihood (under the assumption that they are independent from one another). In theory, this is accomplished by multiplication, but with likelihoods this can lead to underflow. Take the log of the probabilities and sum them instead. The maximum log likelihood indicates the class. If the last group in a file has less than 100 clicks, discard these predictions.

As you predict, keep track of correct and incorrect predictions. Create a confusion matrix, a matrix where the rows represent the actual classes (Gg, Lo) and the columns represent what these were actually classified as. If Gg is encoded as 0 and Lo as 1, misclassifying a group of 100 Gg clicks as Lo would add one to confusion[0][1]. Output your confusion matrix either graphically (submit the file) or to the console. Be sure that your rows and columns are labeled. If your system is performing well, most of the counts should be on the diagonals. In addition, you must compute and print your overall error rate.

Error rate depend significantly on the random split of training and test data, but on average it should be achievable to have an error rate under 3%. Neural networks are non-deterministic due to their random starting points. Small differences in data handling and shuffling of training data also contribute to making exactly reproducible results difficult.

As a consequence, there is no automated grader for functionality in this assignment. Points for functionality will be assigned as follows:

- Valiant Effort/Right Track – You made a good faith effort but were unable to produce a network that classified appropriately and did not produce the appropriate metrics. If you were a good portion of the way to having a successful program, right track will be awarded.
- Mostly Right – You must produce a network that properly partitions and uses the data and produces an error rate under 25% and produce a confusion a matrix.
- Good – The above must hold, your error rate must be under 10%, and your confusion matrix must be a plot that is well labeled with rows and column functionality clearly indicated, and use a weighted loss function. You can compensate to some extent for class imbalance by providing a dictionary whose keys are class numbers 0:N-1 and whose values are a weight which will scale the loss present for each class. The dictionary is passed to the fit function with the class_weight parameter. If our training data had a 10 examples of class 0, 5 of class 1, and 5 of class 2, we would use weights={0:1, 1:2, 2:2}which would double the loss of the classes with fewer examples.
- Excellent – All of the above and write a new function that partitions data by site (location) instead of day. Show the error rates for the original experiment plus the new one.

References

Au, W. W. L., Branstetter, B., Moore, P. W., and Finneran, J. J. (2012). "Dolphin biosonar signals measured at extreme off-axis angles: Insights to sound propagation in the head," J Acoust Soc Am 132(2). 1199-1206.

Roch, M. A., Klinck, H., Baumann-Pickering, S., Mellinger, D. K., Qui, S., Soldevilla, M. S., and Hildebrand, J. A. (2011). "Classification of echolocation clicks from odontocetes in the Southern California Bight," J Acoust Soc Am 129(1). 467-475.

Roch, M. A., Stinner-Sloan, J., Baumann-Pickering, S., and Wiggins, S. M. (2015). "Compensating for the effects of site and equipment variation on delphinid species identification from their echolocation clicks," J Acoust Soc Am 137(1). 22-29.

Soldevilla, M. S., Henderson, E. E., Campbell, G. S., Wiggins, S. M., Hildebrand, J. A., and Roch, M. A. (2008). "Classification of Risso's and Pacific white-sided dolphins using spectral properties of echolocation clicks," J Acoust Soc Am 124(1). 609-624.