ASSIGNMENT #2 – NEURAL NETWORKS

Context

This assignment is an opportunity to demonstrate your knowledge and practice solving problems about feedforward neural networks. This assignment contains an implementation component (i.e. you will be asked to write code to solve one or more problems) and an experiment component (i.e. you will be asked to experiment with your code and report results).

Logistics

Assignment due date: 2025-02-14

Assignments are to be submitted electronically through Brightspace. It is your responsibility to ensure that your assignment is submitted properly and that all the files for the assignment are included. Copying of assignments is NOT allowed. High-level discussion of assignment work with others is acceptable, but each individual or small group is expected to do the work themselves.

Programming language: Python 3

For all parts of the implementation, you may use the Python Standard Library (https://docs.python.org/3/library/) and the following packages (and any packages they depend on): Keras, NumPy, Pandas, scikit-learn, SciPy, TensorFlow. Unless explicitly indicated below or explicitly approved by the instructor, you may not use any additional packages.

You must implement your code yourself; do not copy-and-paste code from other sources. Please ensure your implementation follows the specifications provided; it may be tested on several different test cases for correctness. Please make sure your code is readable; it may also be manually assessed for correctness. You do not need to prove correctness of your implementation.

You must submit your implementation as a single file named “assignment2.py” with functions as described below (you may have other variables/functions/classes in your file). Attached is skeleton code indicating the format your implementation should take.

You must submit your report on experimental results as a single file named “assignment2.pdf”.

Implementation Component

Question 1 [10 marks]

Consider the neural network illustrated in Figure 1. It is designed for multi-class classification given an input vector. Using the Sequential API in Keras, write a function that creates a fully connected feedforward neural network with the same architecture. All neurons in hidden layers should use the ReLU activation function. All neurons in the output layer should use the softmax activation function. Compile the model using stochastic gradient descent as the optimizer and categorical cross-entropy loss.

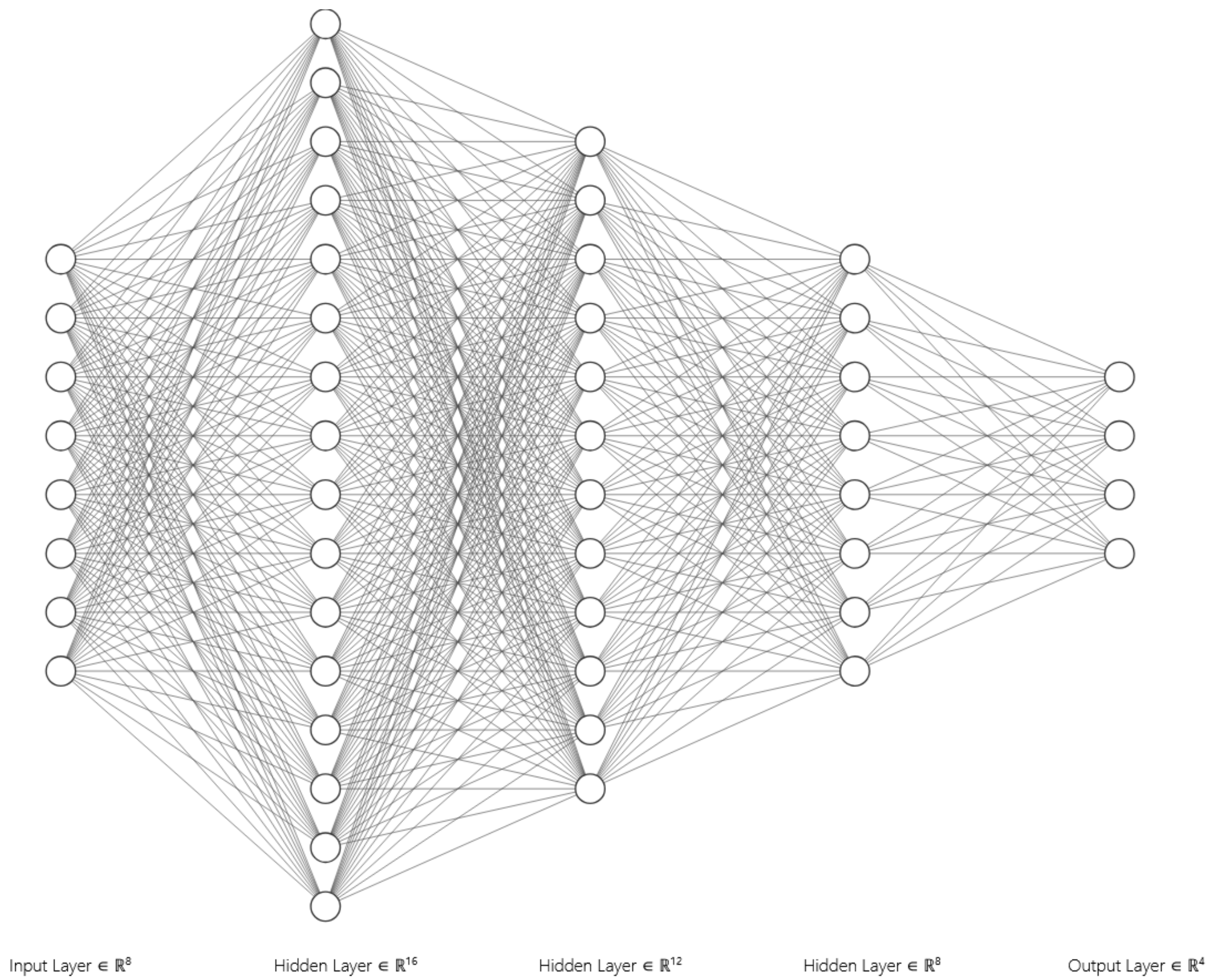

Figure 1. Feedforward neural network architecture. Figure generated using the following tool: http://alexlenail.me/NN-SVG/index.html.

The function must be named “sequential_model”.

The function should take no input arguments.

The function should return one value: (1) a Keras model object.

Question 2 [10 marks]

Consider the neural network illustrated in Figure 2. It is designed for multi-task learning given an input vector; it produces predictions for two binary classification tasks. Using the Functional API in Keras, write a function that creates a feedforward neural network with the same architecture. All neurons in hidden layers should use the ReLU activation function; the output layers should use the softmax activation function. Compile the model using stochastic gradient descent as the optimizer and loss computed as a sum of binary cross-entropy loss for each task.

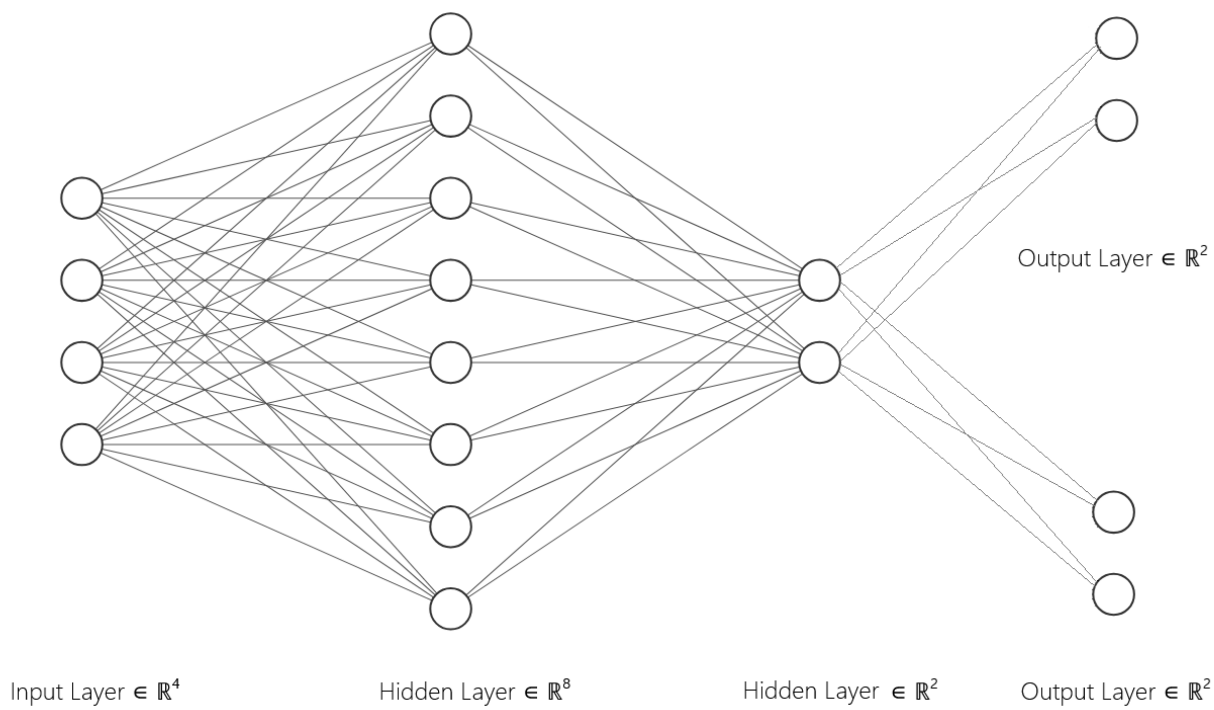

Figure 2. Feedforward neural network architecture. Figure generated using the following tool: http://alexlenail.me/NN-SVG/index.html.

The function must be named "functional_model".

The function should take no input arguments.

The function should return one value: (1) a Keras model object.

Question 3 [10 marks]

Consider developing a feedforward neural network for binary classification to predict recurrence of thyroid cancer following initial treatment using both demographic features (e.g., age, sex, etc.) and clinical features (e.g. thyroid function, subtype of cancer, etc.). A dataset for this task is provided in the attached "Thyroid_Diff.csv" file. The first sixteen columns of the CSV file refer to features. The last column of the CSV file refers to the target we wish to predict (i.e. whether the thyroid cancer recurred). Each row in the spreadsheet refers to an instance.

This dataset is publicly available and it comes from the UCI machine learning repository: https://archive.ics.uci.edu/dataset/915/differentiated+thyroid+cancer+recurrence. Full details on the dataset are provided at this link. Further details on the dataset are provided in the paper below.

Borzooei S, Briganti G, Golparian M, Lechien JR, Tarokhian A. Machine learning for risk stratification of thyroid cancer patients: a 15-year cohort study. European Archives of Oto-Rhino-Laryngology. 2024 Apr;281(4):2095-104.

Write a function that creates, trains, and evaluates a feedforward neural network to predict whether a patient had recurrence of thyroid cancer during the follow-up period from the provided features. If necessary, you may implement pre-processing and/or post-processing of the data within this function.

The function must be called "thyroid_cancer_recurrence_model".

The function should take one input argument: (1) the full file path to the "Thyroid_Diff.csv" dataset.

The function should return two values: (1) a Keras model object that is trained to predict whether a patient had recurrence of thyroid caner during the follow-up period and (2) the performance of the model on the validation set.

Experiment Component

Question 4 [10 marks]

Conduct a series of experiments on the neural network you implemented in the above Question 3 to predict whether a patient had recurrence of thyroid cancer during the follow-up period. For these experiments, hold out a test set.

a. Experiment on the number of neurons in each hidden layer. Plot the performance on the validation set as a function of the number of neurons in each hidden layer.
b. Experiment on the number of hidden layers in your network. Plot the performance on the validation set as a function of the number of hidden layers in your network.
c. Experiment on the number of epochs of training. Plot the performance on the validation set as a function of the number of epochs of training.

Using the held out test set, compute the following.

d. For the model with best performance on the validation set in the above experiments, report performance on the training set, validation set, and test set.