**BA865: Individual Homework Assignment**

Submit this file, edited to include your answers to conceptual questions, along with your Collab notebook (i.e., an .ipynb file) via Blackboard. Leave rubric blank (this is for us to use in grading).

| **Criteria** | **Comments** | **Grade** |
| --- | --- | --- |
| **Comments / Formatting (3)**   * Clean code, intuitive variable naming conventions, code-re-use, etc. * Comments to clarify what purpose your code blocks serve. * Clean notebook formatting, with subsections, e.g., #\*\*TEXT\*\* headers, sub-headers. |  |  |
| **Code & Conceptual Implementation (7)**   * Code executes as is without any errors. * Data imported and pre-processed. * Some assessment of relevant descriptive statistics. * Implements a NN architecture (employing appropriate layers, activations, etc.) * Fits the model using an appropriate loss function. * Implements out of sample model evaluation. * Make suggestions about things to explore going forward that may improve performance. |  |  |
| **Conceptual Questions (5)**   * Responses to conceptual questions (detailed below) are clear, coherent, and correct. |  |  |
| **Total** | | /15 |

**Context & Instructions**

You are a data scientist working for Boston Blue Bikes. As part of an operational improvement initiative, you have been asked to develop a predictive model that can accurately forecast, at the outset of a particular bike rental, how long the customer’s rental will last (i.e., the rental / trip duration). Your predictive model will be used to inform logistics within the bike network, e.g., the reallocation of bikes across / between rental stations, to ensure bike availability. Your manager has provided you with a sample of historical data on bike rentals, including some customer details, geographic information, information on rental timing, and the rental duration. However, your manager was careful to point out that you are welcome to merge in additional predictors / features from external public sources if it helps improve your model.

**Conceptual Questions**

**Draw a visual representation of the neural network that you just implemented (i.e., like the diagrams I have shown you in class, depicting the input layer, hidden layers, and the output layer). State how many weights your network includes.**



**What loss function did you use with your neural network? Why did you choose this loss function? What activation function did you use in the output layer? Why did you choose this activation function?**

Considering the problem of predicting the usage duration of shared bikes, it's likely that the Mean Squared Error (MSE) loss function would be a good choice. MSE is often used for regression problems because it penalizes large errors more than small ones, encouraging the model to focus on reducing those large errors.

As for the activation function in the output layer, I used the Exponential Linear Unit (ELU) activation function. The choice of ELU activation function might have been made based on its performance during model experimentation. ELU can help mitigate the vanishing gradient problem during training, as it has a smooth, non-linear curve that is differentiable everywhere. This smooth curve can help the model learn more effectively, especially in the presence of noisy data.

**Choose any other activation function (one you did not use) and explain why that activation function would have been a bad choice in this setting.**

Choosing the 'softmax' activation function would be a bad choice in this setting. The 'softmax' activation function is commonly used in multi-class classification problems, as it converts a vector of real numbers into a probability distribution over multiple classes. It ensures that the sum of probabilities for all the classes is equal to 1.

However, in this case, we are dealing with a regression problem where we aim to predict a continuous value (bike usage duration) rather than assigning instances to discrete classes. Using 'softmax' in this scenario would be inappropriate, as it would force the output to represent probabilities rather than continuous values.

As for my model improvement process, I started with the 'elu' activation function, which worked best among 'relu', 'elu', 'tanh', 'leakyrelu', and 'sigmoid'. By iteratively adding dropout and layers, I compared their performance and fine-tuned the model architecture to achieve better results. This systematic approach allowed me to identify the optimal model configuration for this specific problem.

**Imagine your prediction task has changed, and your objective is not to predict trip duration, but is instead to predict trip destination (i.e., the station at which the customer is most likely to return the bike). What activation function and loss function would you use for that alternative prediction task?**

If the objective changes to predicting the trip destination, this becomes a classification problem, as we are trying to assign the trip to one of several possible destination classes. In this case, we should make the following changes:

**Activation function**: I would use the 'softmax' activation function. This function will convert the output values into a probability distribution over the possible destination classes. The model will then predict the class with the highest probability.

**Loss function**: Since this is now a multi-class classification problem, I should use the categorical cross-entropy loss function. This loss function is suitable for multi-class classification tasks and measures the difference between the predicted probability distribution and the true distribution (e.g., one-hot). The goal is to minimize this difference, making the model's predictions closer to the true labels. If my target variable is in the form of integer labels instead of one-hot encoded labels, I can use sparse categorical cross-entropy loss function.

**Did you consider using any external datasets to expand your predictor set? If so, which ones? If not, why not? Did you use all the provided predictors as inputs to your neural network? If not, why not?**

In the given context, I did not consider using any external datasets to expand the predictor set due to the limitations in the starttime format of the dataset, which is in Minute:Second:Millisecond. Merging this dataset with external datasets could be challenging. However, if it were possible to obtain more precise timestamp information, I would consider incorporating external datasets like weather data (e.g., weather conditions such as snowing, sunny, etc.).

Regarding predictor selection, I began with every single predictor, and gradually added other predictors to assess their impact on the model's performance. Ultimately, I chose the following predictors for my neural network model: 'start station latitude', 'start station longitude', 'end station latitude', 'end station longitude', 'birth year', and two one-hot encoded features ('usertype\_onehot' and 'gender\_onehot'). These predictors were found to be suitable for the model and contributed to its performance.

**Peers Support**

Indicate here what other students you spoke to when you were working on this project, to brainstorm solutions or work through problems:

None

**ChatGPT (Bing AI, or Bard, etc.)**

Document your use of any AI large-language models (LLMs) here that you used when working on the assignment. Provide the exact prompts you provided to the LLM as input. You can also provide screenshots.

Prompt: How to set learning rate for my model?

Output: