CS 536 : Final Project - Data Completion and Interpolation

Haoyang Zhang, Han Wu, Shengjie Li

May 11, 2019

1 Introduction, group members and division of workload

In this group project, we implemented an autocoder for interpolating missing features from features we have and achieved .

Name NetID	Workload
Han Wu hw436	Fine-tuned the parameters of our model. Did some experiments for the evaluation of our model. Wrote part of the report.
Haoyang Zhang hz333	Analyzed and wrote scripts to clean the data. Wrote scripts to restore human-friendly data from the output of our model. Wrote part of the report.
Shengjie Li sl1560	Implemented the basics of neural networks including back-propagation and several loss functions and activation functions. Wrote part of the report.

2 Prerequisites

1. How to represent or process the data. Data features may contain a number of diverse data types (real values, integer values, categorical or binary values, ordered categorical values, open/natural language responses). How can you represent these for processing and prediction?

2. How to model the problem of interpolation. What are the inputs, what are the outputs? An important if subtle question to consider here - what does it mean to predictor or interpolate a missing feature?

3. Model selection. What kind of model or models do you want to consider?

    - A transfer learning approach:
    - Autoencoders with RNN:

4. Quantifying loss or error. How can you quantify how good a model is, how to measure its loss/error? This is important not only in terms of evaluating your model, but in terms of training as well - how can you refine and improve your model without a way of comparing them?

    Because we are using an autoencoder, we are measuring the reconstruction errors. For the reason that there are many data types, we are treating the data over different loss functions. For real values and integer values, we are using mean squared error $L(\theta) = \frac{1}{m}\sum_{i=1}^{m}(y^i - p^i)^2$(m denotes the number of data points) and root-mean-squared error

5. Training. What kind of training algorithm can you apply to your model(s)? What design choices do you have to make here?
6. Feature selection. It is frequently useful in learning problems to focus on specific features and exclude others, to try to eliminate spurious features and focus on what matters. How can that be applied here?
7. Validation. How can you prevent or avoid over-fitting? Can you apply the usual training/testing/validation paradigm to this problem? How do you choose the training or testing data? Note that a record won't need to be complete to still be useful, potentially, in interpolation. Can cross-validation be applied here? This can be especially important when the data set is not overwhelmingly large and data must be used carefully.
8. Evaluation. How good is your final model? How can you evaluate this? What are the limits and strengths of your model - how many features does a new record actually need to be able to interpolate well?

3 Requirements

4 Bonus