Deep Learning with Structured Data
Mark Ryan

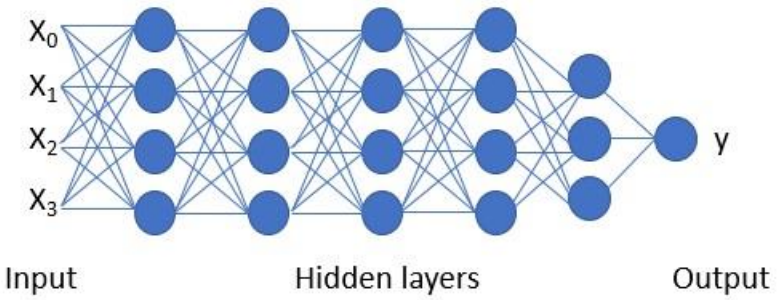
1. Tell us about yourself.

- I have a broad range of academic and professional experience that is related to the topic and relevant to the target audience:
  - Bachelor of Mathematics in Computer Science from the University of Waterloo; Masters of Science in Computer Science from the University of Toronto
  - 20 years leading teams delivering IBM's premier relational database product: Db2
  - Conference presentations on machine learning:
    - Machine learning bootcamp at CASCON conference (Oct 2018):
    - Machine learning with DSX and Db2 at IDUG conference (April 2018):
  - Extensive hands-on experience applying deep learning to practical business problems involving structured data (described in Medium blog posts)
  - Extensive experience as a consumer of machine learning / deep learning educational material, including:
    - Deep Learning Specialization by Andrew Ng
    - Fast.ai by Jeremy Howard
- Unique characteristics and experiences that make me stand out as an author:
  - Extensive experience technical writing (e.g. Getting Started book for IBM VisualAge for Java, numerous blog posts on Medium and developerWorks)
  - Professional experience that spans the worlds of relational database and deep learning
  - Proven ability to connect with stakeholders and audiences
  - Empathy for the reader.
    - I anticipate a significant portion of the audience for this book will be coming from a background of traditional relational database.
    - I also come from a business-oriented relational database background, and because of this I believe that my experience will be both relevant and credible to this audience.

2. Tell us about the book.

- What is the technology or idea that you're writing about?
  - Applying deep learning techniques to problems that involve structured data (such as the data found in relational databases)
- Why is it important now?
  - Over the last six years deep learning has had a transformational

impact on fields like image recognition and natural language processing. For example, two years ago Google replaced its phrase-based machine translation infrastructure with a deep learning system
    - Deep learning is on the inflection point of being adopted widely by businesses beyond the current hotbeds of machine learning (Google, Facebook, Amazon, Uber, etc).
    - The appetite for material on deep learning will increase as it penetrates more industries
    - Current material on deep learning focuses on applications with unstructured data (images, audio, text).
    - In most businesses, people work primarily with structured data. These people want to see examples of deep learning that are relevant to them.
    - The views and feedback I've received on my blogs about deep learning with structured data reinforce this point: there's now a significant audience for details on using deep learning with structured data.
- *In a couple sentences, tell us roughly how it works or what makes it different from its alternatives? Feel free to use pictures.*
    - Deep learning (DL) is a machine learning approach that exploits a multi-layered neural network architecture to train a model on input data
    - The neural network has a series of layers starting with the input layer, followed by a number of hidden layers, and culminating with an output layer



    - In each of these layers, the output of the previous layer goes through a series of operations (multiplication by a matrix of weights, addition of a bias, and application of a non-linear activation function) to produce the input for the next layer. The final output layer generates the prediction of the model based on the input.
    - Deep learning works by iteratively applying **back propagation** to update the weights in each layer. Back propagation uses the partial derivative of the error function (the delta between the actual outcome and the predicted outcome) to update the weights in the neural network in such a way that with repeated applications, the

error function is minimized (and the accuracy of the model is maximized)
    - Once the training is complete (that is, backpropagation has been applied repeatedly to update the weights in the model to achieve the desired accuracy with the training data), the model can then be used to make predictions new data that the model has never seen.
    - Unlike other machine learning approaches, deep learning can take advantage of a high degree of automation and can define a model that accurately identifies extremely complex relationships in the training data.
    - Deep learning has achieved best-of-class performance across a wide range of applications, including image recognition and audio-to-text
    - There are two significant alternatives to deep learning with structured data:
        - Non-DL machine learning: e.g. linear regression, logistic regression, random forest
        - Traditional Business Intelligence (BI)
    - Deep learning provides the following benefits over non-DL machine learning approaches:
        - Performance improves as more data is available for training the model
        - Less demand for feature tuning (time-consuming and error-prone tinkering with the input data in order to get adequate performance from the model)
        - Superior overall performance, given an adequate volume of training data, sufficient compute resources, and parameter tuning
    - Deep learning provides the following benefits over traditional BI techniques:
        - Greater flexibility to adapt to data changes, including schema changes (e.g. new columns) and overall workload changes (e.g. spike in sales in a given jurisdiction)
        - Less code to maintain
        - Potential to automate detection of new benefits.
            - Relational databases document their own structure in catalog tables.
            - It is possible to use these catalog tables to iteratively crawl tables in a database and apply deep learning to detect unanticipated patterns in the data
- *What type of book are you planning to write?*
    - Is your book a tutorial or a reference?
        - Tutorial
    - Does this book fall into a Manning series such as In Action, In Practice, Month of Lunches, or Grokking?
        - Deep learning

- Are there any unique characteristics of the proposed book, such as a distinctive visual style, questions and exercises, supplementary online materials like video, etc?
    - I plan to use a complete, end-to-end code example (subset of it here) with open source data to illustrate the concepts and practice of applying deep learning to structured data. See next section for a description of the example topic.
    - Because the data and code will be openly available, the reader will be able to follow along and develop their own variations on the code as they are guided through an exploration of the data and the deep learning code that consumes the data

3. *Please give us 5 or 6 representative tasks in the domain of your book.*

To set the scene, I'll summarize here the example problem that will be used throughout the book to illustrate how to do deep learning with structured data.

- The TTC (Toronto's transit system) runs a large network of on-street light rail vehicles – streetcars – as an essential part of its overall public transit system.
- This streetcar system is the largest network of on-street light rail transit in North America. It is also the only streetcar network in North America that survived the post-war trend to replace streetcars with buses.
- Along with their many advantages (including lower construction/maintenance cost than subway and lower greenhouse gas emissions / driver cost than buses), streetcars have one big disadvantage: when they are delayed or disabled they can generate gridlock because other vehicles can't easily get around them
- The city of Toronto publishes a detailed dataset that describes all streetcar delays since 2014
- By applying deep learning to this dataset, we can generate useful predictions (e.g. which routes / vehicles / neighbourhoods are prone to delays) to help prevent the gridlock caused by marooned streetcars. For example, we can predict the likelihood of whether a streetcar on a given route on a given day of the week at a given time of day will encounter a delay of over 5 minutes.
- This problem is a good example for the book because the input dataset is:
    - Open, which means the reader will have full access to the dataset
    - Broadly accessible, which means the reader doesn't need background in any particular industry to understand the data
    - Messy, which means it has the gaps, errors, and anomalies that are common in real-world datasets. Readers will be able to extrapolate their experience dealing with this messy dataset when they need to prepare other real-world datasets for machine learning
- This table shows a snippet of the input dataset:

	Report Date	Route	Time	Day	Location	Incident	Min Delay	Min Gap	Direction	Vehicle
0	2014-01-02	505	06:31:00	Thursday	Dundas and Roncesvalles	Late Leaving Garage	4.0	8.0	E/B	4018.0
1	2014-01-02	504	12:43:00	Thursday	King and Shaw	Utilized Off Route	20.0	22.0	E/B	4128.0
2	2014-01-02	501	14:01:00	Thursday	Kingston road and Bingham	Held By	13.0	19.0	W/B	4016.0
3	2014-01-02	504	14:22:00	Thursday	King St. and Roncesvalles Ave.	Investigation	7.0	11.0	W/B	4175.0
4	2014-01-02	504	16:42:00	Thursday	King and Bathurst	Utilized Off Route	3.0	6.0	E/B	4080.0

Here is a description of the columns of this table:

- **Report Date**: date of the incident
- **Route**: route number
- **Time**: time of day of the incident
- **Day**: day of week of the incident
- **Location**: cross street or landmark where the incident took place
- **Incident**: free-form text description of the incident
- **Min Delay**: # of minutes of delay caused by the incident
- **Min Gap**: # of minutes of gap between streetcars caused by incident
- **Direction**: travel direction of the streetcar when the incident occurred
- **Vehicle**: ID code of the streetcar involved in the incident

Following are six representative coding tasks that a reader needs to follow to create a model using deep learning with structured data:

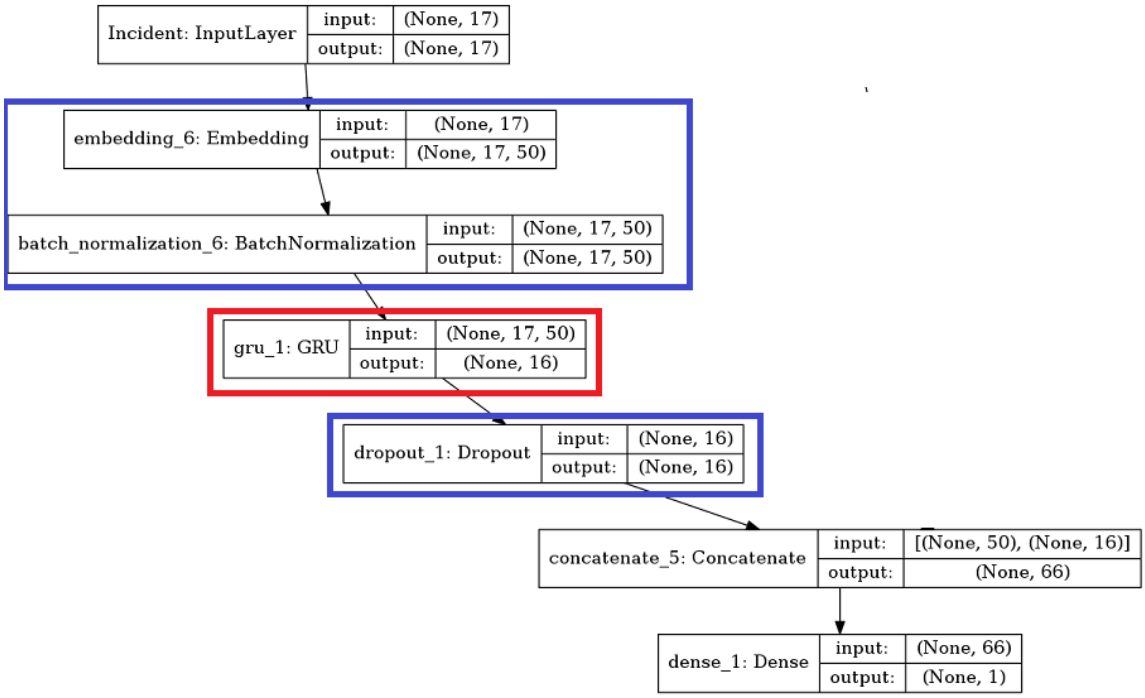
1. **Classify input columns** into one of the following categories:

Category	Example columns	Processing required
Continuous	Min Delay Min Gap	Replace missing values with zeros
Categorical	Route Day Vehicle	• Replace missing values with “missing” token • Replace values with integer IDs • Include embedding layers
Text	Incident	• Replace missing values with “missing” token • Remove punctuation and stop words • Replace words with integer IDs

- Correct categorization of the input data is critical. All the

processing that follows (e.g. how missing values are dealt with, how the input to the Keras model is built, and ultimately how the deep learning model is built) depends on these categorizations.

- To see how the categorization impacts the final model structure, consider this snippet of the deep learning model that shows the layers for the **Incident** column:
    - model layers that are automatically generated for text and categorical columns are highlighted in blue
    - model layers that are automatically generated for text columns are highlighted in red:



2. **General data cleanup**:

- "Wild" data (such as the dataset for the sample problem in this book) has inconsistencies and problems that need to be corrected before the data can be used to train an ML algorithm:
    - **Missing data**: the ML algorithm won't work if there are Python "NaN" values in the input. Your code needs to replace missing values with appropriate placeholders (such as a "missing" token for categorical columns, or zeros for continuous columns)
    - **Values that mismatch the overall type of the column**: the preprocessing of values won't work if, for example, there are text values in a column that you have categorized as continuous. You need to use Python casting to ensure that for each column in the input dataframe all the values match the expected type.

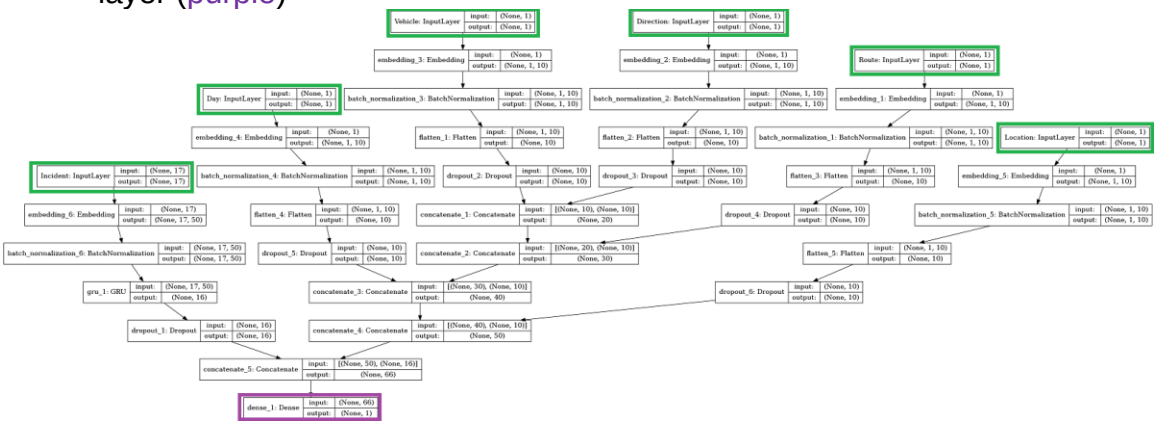
- **Data entry inconsistencies**: if part of your dataset is input manually in free-form fields, you can end up with multiple distinct tokens for the same categorical value. If you don't fix these inconsistencies the performance of your ML model will be undermined. For example, the following three **Location** values indicate a single physical location. Each of the three has a high number of incidents. If you don't correct these Location values by making them all a single value, your ML algorithm's performance will suffer worse performance because it will treat these three as distinct locations:

Location value	**Frequency**
"roncy yard"	936
"roncesvalles yard"	898
"ronc. carhouse."	425

3. **Prepare categorical columns** – for columns with categorical values, invoke Python APIs to replace string values with tokens
    - ML algorithms operate on numeric input, so you need to replace the string values for categorical values with IDs that uniquely identify each value
4. **Prepare text columns** – for columns with multi-word text values, invoke Python APIs to perform basic text preparation (e.g. convert to all lower case, remove stop words and punctuation) and replace strings with tokens:
    - Just like categorical columns, string values in text columns need to be replaced with integer IDs that ML algorithms can consume
    - In addition, text columns include tokens that can reduce the performance of the ML model if you don't remove them. Such tokens include:
        - Punctuation
        - Articles: "the", "an", "a"
        - Conjunctions: "and", "or"
5. **Define model input data structure**: use the continuous, categorical, and text columns that you have prepared to create the input structure for the Keras deep learning model
    - The code that builds the Keras model expects the input in a specific format
    - You need to run code that takes the output of the cleaning and categorization steps and creates the Python dictionary data structure that the Keras model expects
6. **Define the deep learning model**: use the input data structure you defined in the previous task to generate the embedding, RNN, and other layers for your model.
    - With the input to the Keras model you defined in the previous step, you need to run the code that defines the layers of the model for each category of input and then concatenates these layers

together to create the model

- The following plot shows the result – each input (green) goes through a set of layers depending on its category and the layers are concatenated together to ultimately provide input to the final output layer (purple)



4. The minimally-qualified reader (MQR)

The minimally-qualified reader will have the following skills prior to reading this book:

- Beginner-Intermediate skill in Python
- Beginner-Intermediate skill in Jupyter Notebooks
- Beginner-Intermediate skill in non-DL machine learning
- Beginner skill with at least one of the common Cloud environments (e.g. AWS, Azure, IBM Cloud)

What is the primary job role of your readers?

- The minimally-qualified reader will be an IT professional (coder, DBA, data scientist) with at least 2 years of experience
- This book would also be appropriate for:
  - Business stakeholders who are familiar with coding and basic relational database and want to understand the potential of applying deep learning to structured data
  - Students who are studying deep learning and are interested in its potential applications in business

What do you expect your MQR to already know before they start reading?

Prior to starting to read this book, I expect that the MQR will already know the following:

- Beginner-Intermediate skill in Python, including
  - Type handling

- Basic constructs (e.g. lists, dictionaries)
- Control structures (e.g. if, while) and operators
- Numpy

- Beginner skill in Jupyter Notebooks
  - Code vs. markup cells
  - Executing and debugging code cells
- Beginner skill in non-DL machine learning
  - Familiarity with basic algorithms: e.g. Logistic regression, linear regression, random forest
  - Rough grasp of mathematical concepts (linear regression, basic calculus) sufficient to understand back propagation, the "secret sauce" of DL: define a function to capture the error (delta between the actual outcome for a given set of incomes and the outcome predicted by the model) and use the derivative of that function to adjust the weights in the model to minimize the error
- Beginner Cloud skill, such as setting up a basic service on one of the mainline Cloud providers: AWS, Azure, IBM Cloud

5. Q&A

Here are some common questions the come up about deep learning and deep learning with structured data:

- How is deep learning different from other forms of machine learning?
- What are the advantages and disadvantages of deep learning compared to non-DL machine learning approaches?
- Why use deep learning with structured data? Aren't there simpler / cheaper ways to get the same result?
- Deep learning needs tons of data: how can you get enough structured data for a given problem to be adequate to train a deep learning model?
- Where can I find an open structured data sources to learn with?
- What kinds of problems with structured data lend themselves to a deep learning approach?

6. Tell us about the competition and the ecosystem.

- *What other books are available on this topic?*
  - There are a few foundational texts on the overall topic of deep learning (e.g. Deep Learning, Introduction to Deep Learning, Deep Learning with Python)
  - In addition there are books (e.g. Deep Learning: Natural Language Processing in Python) and courses (e.g. Natural Language Processing with Deep Learning in Python) that cover key concepts for deep learning with structured data, like embeddings
  - However, I am not aware of any other books that focus on deep

learning with structured data specifically

- *How does the proposed book compare to them?*
    - Unlike the books described above, my proposed book is specifically focused on deep learning with structured data, and covers an end-to-end practical example (including open data & code) of applying deep learning to structured data
    - Unlike many deep learning books, my proposed book is aimed at a professional, rather than an academic, audience. While it will introduce theoretical topics, the focus of the book is practical application using an accessible stack (Python developed in Notebooks with Keras as the deep learning framework)
    - Because of this combination of down-to-earth applications and easy-to-use technology, my proposed book will appeal to a broad and largely unserved audience.
- *What resources would you currently recommend to someone wanting to learn this subject?*
    - Sections of the Fast.ai course by Jeremy Howard and sections of the deeplearning.ai specialization from Andrew Ng
- *What are the most important web sites and companies associated with this topic?*
    - I am not aware of any web sites specifically dedicated to deep learning with structured data. This area is promising but as-yet unexplored.
    - This article indicates that the majority of commercial applications of deep learning are on unstructured (e.g. image) data
    - This article covers some related ideas about applying ML to yield structured data from unstructured data
- *Where do others interested in this topic gather online?*
    - Medium – posts on this topic include Using Deep Learning for Structured Data with Entity Embeddings and Structured Deep Learning
    - The forum for Jeremy Howard's fast.ai course has some interesting discussions on the topic, including this thread

7. Book size and illustrations

Please estimate:

- The approximate number of published pages to within a 50-page range
    - 300 pages
- The approximate number of diagrams and other graphics
    - 30
- The approximate number of code listings
    - 40

8. Contact information

Formal Name: Mark B.P. Ryan
Name: Mark Ryan

Mailing Address:

1398 Rennie Street
Oshawa, Ontario
Canada
L1K 0H1

Preferred email: ryanmark@rogers.com
Preferred phone: 416 434 9173
Skype: live:ryanmark_3

Website/blogs/Twitter, etc:

blogs: https://medium.com/@markryan_69718
LinkedIn: https://ca.linkedin.com/in/mark-ryan-31826743
Twitter: @MarkRyanMkm

***9*. Schedule**

- *Most authors require 2-4 weeks to write each chapter. Please estimate your writing schedule*

  Assuming a green light before the end of January 2019:

  Chapter 1:
  - End of Feb 2019

  1/3 manuscript:
  - End of April 2019

  2/3 manuscript:
  - End of June 2019

  3/3 manuscript:
  - End of July 2019

- *Are there any critical deadlines for the completion of this book?*
  - No

10. Table of Contents

Table of Contents
Deep Learning with Structured Data

3.5 Summary

4 Preparing Structured Data for Deep Learning Part 2: Cleaning and Transforming the Data

In this chapter the reader will learn how to clean up real-world data for deep learning by eliminating errors / missing data and by converting all the data into numbers that the deep learning algorithm can handle.

4.1 The big picture: machine learning algorithms run on numbers, so the input data needs to be massaged so all the input values are numbers

4.2 Dealing with missing values: how to quickly assess where the missing values are, options for replacing missing values, what can happen if you don't deal with missing values

4.3 Dealing with inconsistent values: how to identify and deal with duplicate categories and other anomalies

4.4 Dealing with type problems: the pros and cons of Python's approach to types, common errors that you encounter when values don't match your type assumptions, how to correct type problems

4.5 Augmenting your data with derived values

4.6 Transforming categorical and text columns: replacing strings with integer identifiers, basic text processing (common casing, stop word elimination)

4.7 Summary

5 The Deep Learning Stack: a Deeper Look

In this chapter the reader will take a deeper look at the open deep learning stack that used in the book, from the key Python APIs to the deep learning model architecture.

5.1 The big picture: an easy-to-use end-to-end environment for deep learning with structured data

5.2 The data manipulation object: more on Pandas dataframes, how to do common SQL operations (selects, joins) with Pandas

5.3 The DL library: Keras vs. TensorFlow, Keras sequential model vs. model class with sequential APIs

5.4 Embeddings: learned vectors for the relationship between categorical or text values, how embeddings allow you to get unsupervised learning benefits while you are solving a supervised learning problem

5.5 The model architecture: use the data categories to automatically define the layers of the Keras model, how to apply the same code to different input datasets

5.6 Summary

6 Preparing and Building the Model

In this chapter the reader will learn how to prepare a Pandas dataframe to be fed into a Keras model and how to use the data categorizations to automatically build the layers of the model.

6.1 The big picture: creating a Keras model automatically by building up layers defined by the structure of the input dataset

6.3 Transforming the dataframe into the format expected by the Keras model

6.2 Layers created for continuous columns

6.3 Layers created for categorical and text columns

6.4 How to understand the structure of the model: model.summary() and plot_model()

6.5 Model options: output layer activation function and optimization function, learning rate

6.6 Summary

7 Training the Model

In this chapter the reader will learn how to train a Keras deep learning model, how to measure the performance of the model, and what to do when the model is not performing as expected.

7.1 The big picture: running the model on the training data to set weights that will allow the model to accurately predict outcomes on new data

7.2 Training, Validation, and Test datasets: what each slice of data is for, recommended ratios for the validation and test datasets

7.3 Hyperparameters: learning rate, batch size, number of epochs, regularization parameters to control overfitting

7.4 Training iterations: getting a successful first iteration, measuring training progress, training problems (overfitting, memory exhaustion) and how to address them

7.5 Saving your trained model

7.6 Summary

8 Deploying and Maintaining the Model

In this chapter the reader will learn how to deploy and maintain a Keras deep learning model in production

8.1 The big picture: the model lifecycle - deploy, update, deploy

8.2 Deployment options

8.3 Model maintenance: maintaining a model once it has been deployed, retraining cycles, how to measure whether a model is still working well as the live data changes

8.4 Summary

9 Recommended Next Steps

In this chapter the reader will get advice on how to select a deep learning with structured data project of their own. The reader will also be introduced to more resources to learn about deep learning

9.1 The big picture: you've tackled your first project - now what?

9.2 How to select a new project on deep learning with structured data: structured datasets that are amenable to deep learning, further potential for automation

9.3 Sources for additional learning on deep learning: deeplearning.ai, fast.ai, key blogs and Youtube resources

9.4 Summary