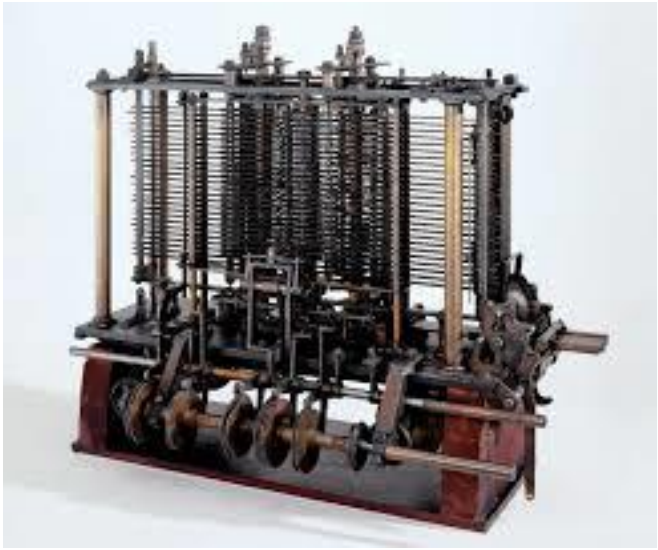
Deep Learning with Structured Data

Hands-on Workshop

June 17, 2020

Mark Ryan

Introduction



- Hands-on follow on to Apr. 29 session
- Agenda:
  - Review environment & set up steps
  - Run training experiments for MNIST
  - Run training experiments for streetcar delay prediction model
  - Complete exercises that allow you to make your own updates to the model training and re-run training
  - Wrap up
- You'll learn how to:
  - Clone a repo into Drive and access its code in Colab
  - Train a simple deep learning model and assess the trained model's performance
  - Take advantage of the specialized hardware that is available in Colab

Image: brittanica.com

Review Environment

Google Drive:

- Target for repo clone
- Location of files used for training

Google Colaboratory (Colab):

- Environment for running notebooks
- Offers deep learning specific hw (GPUs and TPUs)

Repo directory structure:

```
├──data
|       2014_2019_df_cleaned_remove_bad_values_may16_2020.pkl
|       routedirection.csv
|
├──models
|
├──notebooks
|       custom_classes.py
|       exercises_streetcar_model_training.ipynb
|       keras_functional_api_mnist.ipynb
|       keras_sequential_api_mnist.ipynb
|       streetcar_model_training.ipynb
|       streetcar_model_training_config.yml
|
└──pipelines
```

Review Set Up Steps

1. Access Drive using Google ID
2. Create new folder in Drive and use Colab to clone repo into it https://github.com/ryanmark1867/dl_structured_data_hands_on
3. Open notebook from the repo in Colab
4. Mount Drive in notebook
5. Make /notebooks directory the current directory in the notebook

NOTE: Steps 3-5 can be reused for new Colab sessions or to work with new notebooks

Hands-on Gameplan

1. Run Keras sampler training notebooks:
   - `keras_sequential_api_mnist.ipynb`
   - `keras_functional_api_mnist.ipynb`
2. Run basic streetcar delay training notebook `streetcar_model_training.ipynb`
   - Start with config file `streetcar_model_training_config.yml` as-is to get a basic run
   - Update the runtime for the notebook to use a GPU
   - Re-run experiment 1
- Work through exercises in notebook `exercises_streetcar_model_training`

Image: tvseriesfinale.com

Experiment Summary

Experiment	Epochs	Early stop enabled?	Weight for "1" (delay) values	Early stop controls	
				monitor	mode
0	1	no	1.0	NA	NA
1	10	no	1.0	NA	NA
2	50	no	1.0	NA	NA
3	50	no	No delay / delay	NA	NA
4	50	yes	No delay / delay	Validation loss	min
5	50	yes	No delay / delay	Validation accuracy	max
9	20	yes	No delay / delay	Validation accuracy	max

- Control which experiment is run with the `current_experiment` parameter in the config file `streetcar_model_training_config.yml`

```
test_parms:
      testproportion: 0.2 # proportion of data reserved for test set
      trainproportion: 0.8 # proportion of non-test data dedicated to training (vs. validation)
      current_experiment: 0
      repeatable_run: False # switch to control whether runs are repeated identically
      get_test_train_acc: False # switch to control whether block to get test and train accuracy is after training
```

Monitoring Model Training Runs

- Output of `fit` command (feedback on what's happening for each epoch):

```
1889663/1889663 [==============================] - 15s 8us/sample -
loss: 0.8849 - accuracy: 0.7141 - accuracy_1: 0.7709 - val_loss:
0.5020 - val_accuracy: 0.7208 - val_accuracy_1: 0.7208
```

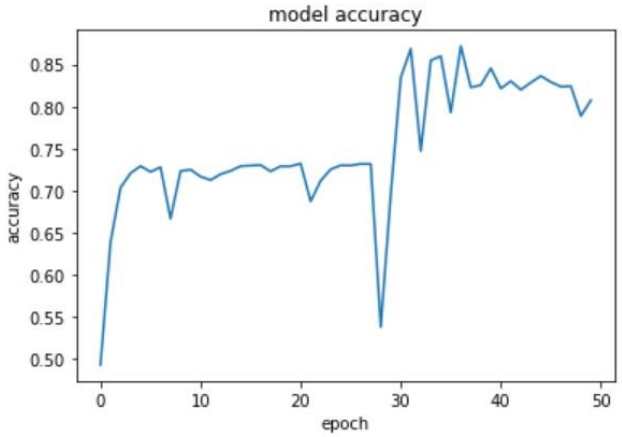
- Output of `evaluate` (controlled by `get_test_train_acc` parameter):

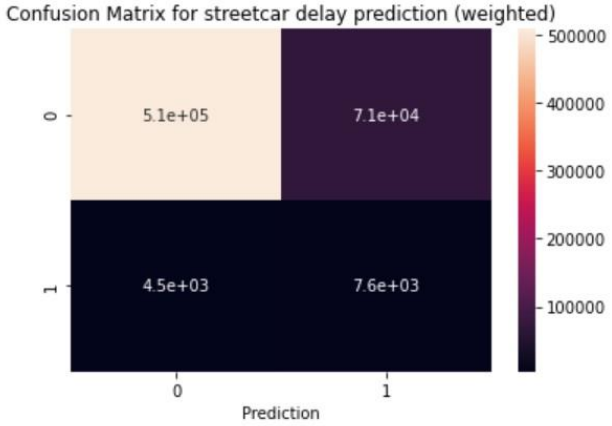
```
metrics names  ['loss', 'accuracy', 'accuracy_1']

Train: 0.873, Test: 0.873
```

- Output of `%%time` command:

```
Wall time: 12min 12s
```

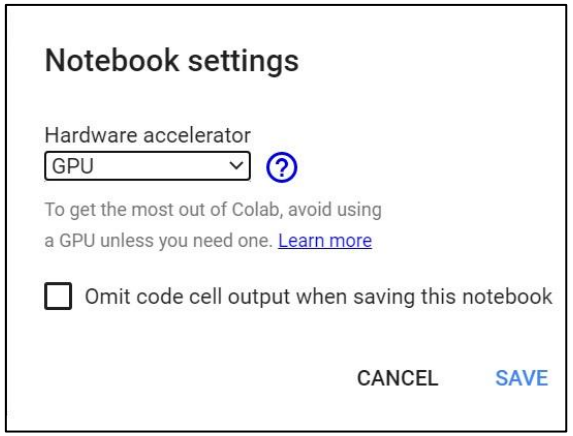
- Results by epoch chart and confusion matrix:





Monitoring Model Training Runs

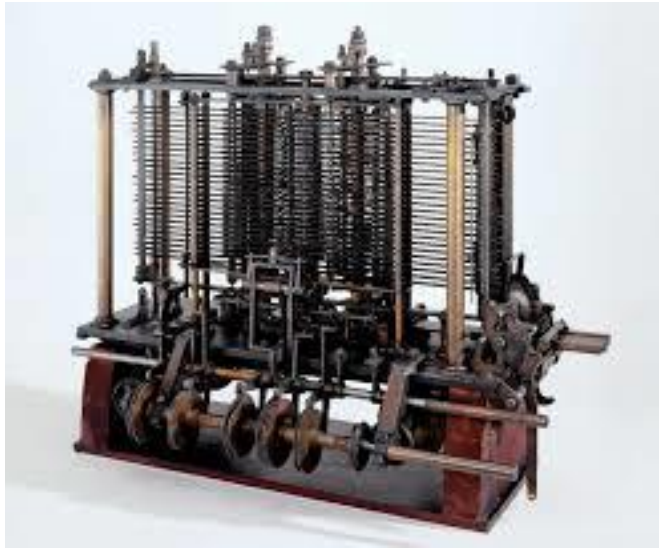
- In Colab, you can request GPU/TPU by Runtime -> Change runtime type:



- Once you have done that, run `!nvidia-smi` in a new cell to get details:

```
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 440.82       Driver Version: 418.67       CUDA Version: 10.1     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|===============================+======================+======================|
|   0  Tesla P100-PCIE...  Off  | 00000000:00:04.0 Off |                    0 |
| N/A   44C    P0    27W / 250W |      0MiB / 16280MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+
```

Wrap up



- Let Dragos know if you found this session useful and would like to have a hands-on session on model deployment
- Check out the book: Deep Learning with Structured Data
- Check out the full repo: https://github.com/ryanmark1867/deep_learning_for_structured_data
- Connect with me:
  - LinkedIn: https://www.linkedin.com/in/mark-ryan-31826743/
  - Medium: https://medium.com/@markryan_69718

Image: brittanica.com