Keras for Tabular Data

May 2020
Agenda

• Quick overview of the models
• How to improve the performance of your Keras models
• Focus a lot on the hyperparameters for Keras models
Overview

- Automated preprocessing
- Flexible architecture
- Automated hyperparameter selection/tuning
- Interpretable
- One click deployment

Automation means you quickly get a very good deep learning model
Preprocessing & Architecture

- **MultiModal Learning**
- Handle Numeric, Categorical, Text, and Images
- Variety of preprocessing for numeric, categorical, text, and images... Explore the blueprints!

- Feedforward
- Self-Normalizing Neural Networks
- Neural Architecture Search
Blueprints

- Different Preprocessing
- Different Architectures
- Different Training Schedules
- All Tunable!

Easy starting point . . . try them all
Improving your Keras Model

• Run All the Keras models
• Feature selection
• Tune
  • Hidden Activation Functions
  • Output Activation Functions (for regression, linear vs. exponential)
  • Loss Functions (RMSE, RMSLE, Poisson, Gamma, Tweedie, etc.)
  • Learning Rates - we support using a schedule or constant
  • Network architecture - experiment with depth and width
  • Regularization

Remember AutoPilot models are tuned to run quickly
Impute missing values and normalize
Fold and holdout
Establish a benchmark

Build a simple network
Determine the output activation and loss functions
Learn the hidden activation

Start with 100 steps per epoch
Use the LR Range Test and 1cycle LR policy
Try a few epochs before many epochs

Benchmark your model
Examine and track the distribution of predictions
Tune your model, reassess, repeat
Formulaic Tuning with a focus on fit

Regularize
When Overfitting

- Increase learning rate
- Decrease batch size
- Run fewer epochs
- Reduce network capacity
- Increase dropout

Specialize
When Underfitting

- Decrease learning rate
- Increase batch size
- Run more epochs
- Increase network capacity
- Reduce dropout
Train your Model Longer

• It’s not as simple as increasing your epochs, it’s important to balance
  • Network capacity
  • Batch size
  • Learning Rate

• Use the “adaptive training schedule” to automate choosing the optimal number of epochs
Training Schedules

- Vary the learning rate according to 1cycle\(^1\) to allow for larger learning rates, discourage getting stuck in a local minima.
- DataRobot has conducted many experiments to determine appropriate learning rate schedule
- DataRobot has also developed a new adaptive training schedule

1. From "A disciplined approach to neural network hyper-parameters. Part 1 -- learning rate, batch size, momentum, and weight decay". Leslie Smith
Choose the output activation and loss function according to the target distribution

RMSE

RMSLE, Gamma, or Poisson

Tweedie

If the target grows exponentially, use an exponential activation function
Some Final Thoughts

• Try them - especially for text

• Great for ensembling
  • They help add diversity!

• For tabular data:

  *Deep Learning is not always the best model*
Questions & Answers
● **Engage, learn, and accelerate** your AI/ML journey
● **Connect** with peers to find solutions to AI challenges
● **Explore** helpful content to take your AI to the next level
● **Build** your brand as an AI expert & thought leader
● **Join** your peers today at [community.datarobot.com](http://community.datarobot.com)

**Questions:** [aisuccess-webinars@datarobot.com](mailto:aisuccess-webinars@datarobot.com)