

Democratizing AI with sparklyr

@javierluraschi
RStudio PBC - OSS



RStudio AI

RStudio is a Public Benefit Corporation supporting R. The RStudio AI team is focused on making AI accessible to the Data Science community.



RStudio Server Pro
Take control of your R and Python code



RStudio Connect
Connect data scientists with decision makers



RStudio Package Manager
Control and distribute packages



The R Consortium is a collaboration between the R Foundation, RStudio, Microsoft, TIBCO, Google, Oracle, HP



RStudio and Shiny are affiliated projects of the Foundation for Open Access Statistics. FOAS works to



RStudio is proud to be a corporate sponsor of NumFOCUS, whose mission is to promote open practices



RStudio PBC is a sponsor of the Linux Foundation, which supports the creation of sustainable open source



@rstudio AI Blog

Home Gallery About Contributing

April 28, 2020

Towards privacy: Encrypted deep learning with Syft and Keras

Deep learning need not be incompatible with privacy protection. Federated learning enables on-device, distributed model training; encryption keeps model and gradient updates private; differential privacy prevents the training data from leaking. As of today, private and secure deep learning is an emerging technology. In this post, we introduce Syft, an open-source framework that integrates with PyTorch as well as TensorFlow. In an example use case, we obtain private predictions from a Keras model.



April 21, 2020

sparklyr 1.2: Foreach, Spark 3.0 and Databricks Connect

A new sparklyr release is now available. This sparklyr 1.2 release features new functionalities such as support for Databricks Connect, a Spark backend for the 'foreach' package, inter-op improvements for working with Spark 3.0 preview, as well as a number of bug fixes and improvements addressing user-visible pain points.



Credit: rstudio.com, blogs.rstudio.com/ai

SUBSCRIBE
Enjoy this blog? Get notified of new posts by email:

Please check this box if you accept the RStudio privacy policy:
 Submit
Posts also available on [Bloggers](#)

CATEGORIES
Audio Processing (2)
Bayesian Modeling (4)
Cloud (3)
Concepts (9)
Data Management (1)
Distributed Computing (1)
Explainability (2)
Image Recognition & Image Processing (11)
Meta (3)
Natural Language Processing (8)
Packages/Releases (14)
Privacy & Security (3)
Probabilistic ML/DL (11)

Artificial Intelligence

Artificial Intelligence is a rapid-growing field with significant transformational impact to many industries and civilization at large.

ImageNet Classification with Deep Convolutional Neural Networks

Ales Krizhevsky
University of Toronto
kriz@cs.utoronto.ca

Ilya Sutskever
University of Toronto
ilya@cs.utoronto.ca

Geoffrey E. Hinton
University of Toronto
hinton@cs.utoronto.ca

Abstract

We trained a large, deep convolutional neural network to classify the 1.2 million high-resolution images in the ImageNet LSVRC-2010 contest into the 1000 different classes. On the test data, we achieved top-1 and top-5 error rates of 37.5% and 11.7%, which is very close to the best entries in the competition. The neural network, which has 60 million parameters and 650,000 neurons, consists of eight convolutional layers, some of which are followed by max-pooling layers, and three fully-connected layers with a final 1000-way softmax. To train faster, we used non-spatially-separable layers and a more efficient GPU implementation of the convolution operation. To reduce overfitting in the fully-connected layers we used a recently-developed regularization method called “dropout” that turned out to be very effective. We also tested a variant of this model in the ILSVRC-2012 competition and achieved a winning top-5 test error rate of 15.3%, compared to 26.2% achieved by the second-best entry.

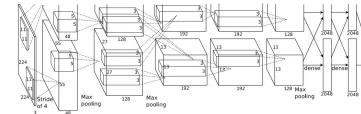


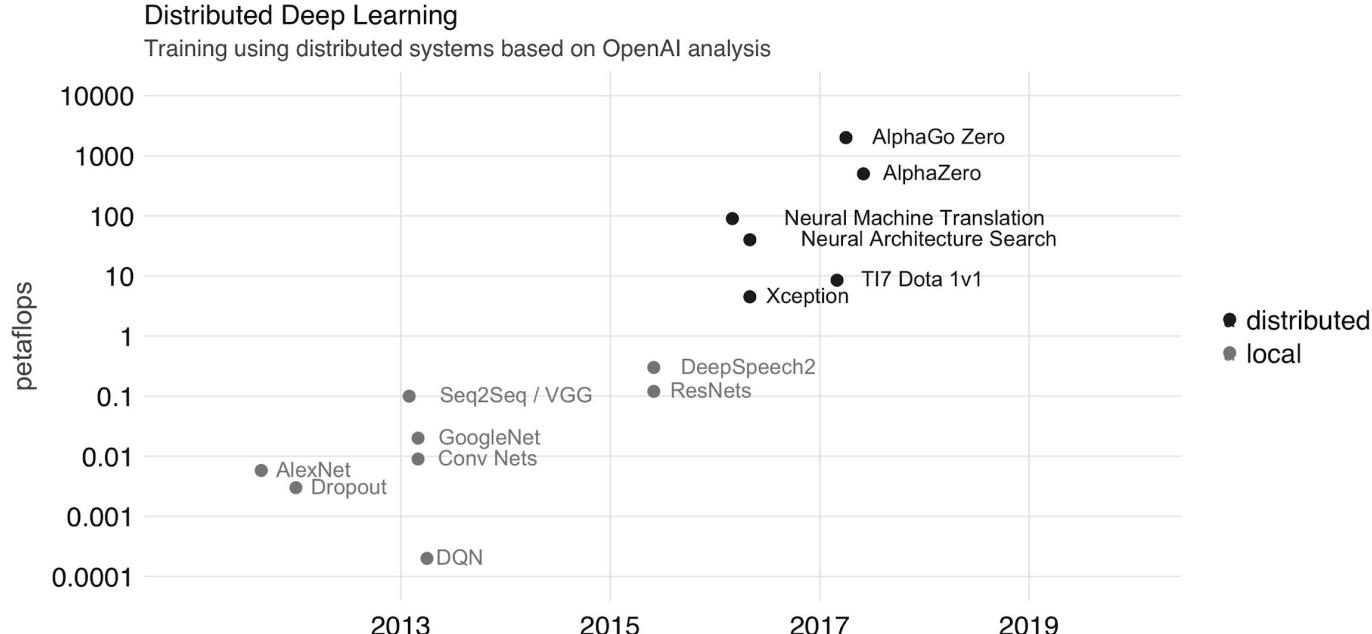
Figure 2: An illustration of the architecture of our CNN, explicitly showing the delineation of responsibilities between the two GPUs. One CPU runs the layer-parts at the top of the figure while the other runs the layer-parts at the bottom. The GPUs communicate only at certain layers. The network’s input is 150,538-dimensional, and the number of neurons in the network’s remaining layers is given by 253,440–186,624–64,596–64,596–43,254–4096–4096–1000.



Modern AI starts with the deep learning breakthrough from AlexNet and more recently with achievements like DeepMind's AlphaGo.

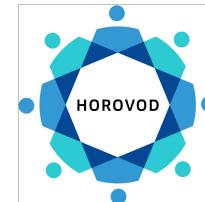
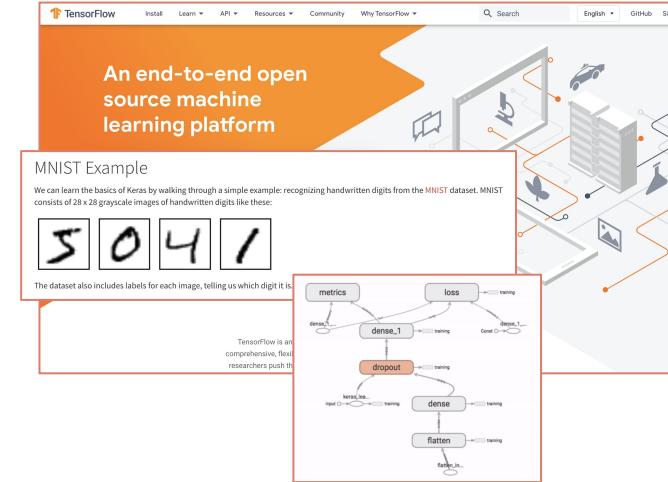
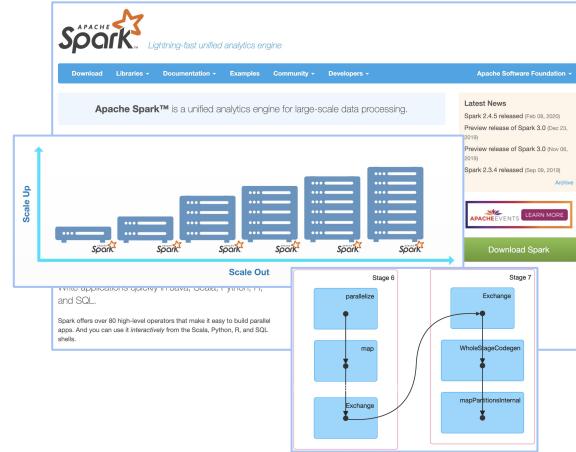
Artificial Intelligence

Deep Learning is a key discipline of modern AI, but Reinforcement Learning, statistical analysis and distributed computing are as relevant today.



Frameworks

Apache Spark, TensorFlow, PyTorch, MLflow, Horovod, and so on, are arguably required frameworks to build AI

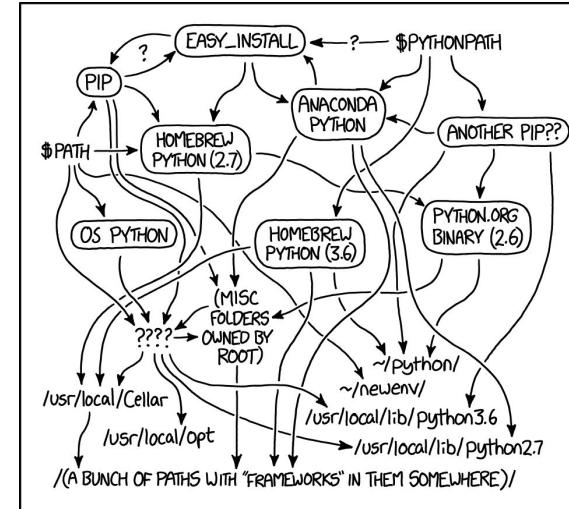


PyTorch **mlflow**

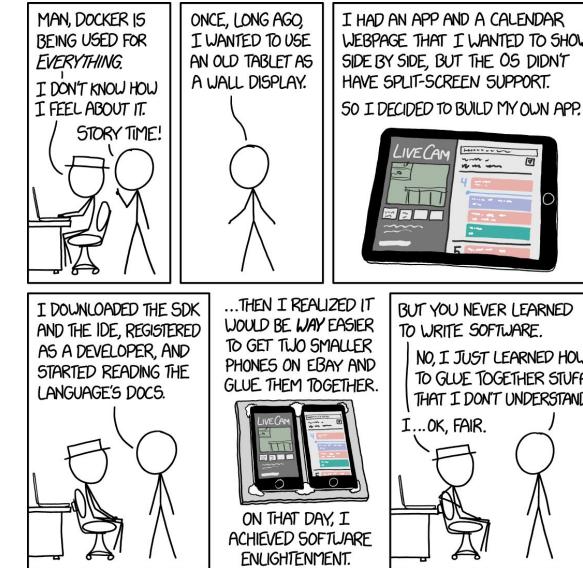


Today

AI is still hard, requires highly-qualified teams to set up the infrastructure and successfully apply AI techniques.



MY PYTHON ENVIRONMENT HAS BECOME SO DEGRADED
THAT MY LAPTOP HAS BEEN DECLARED A SUPERFUND SITE.



Installing and using say: TensorFlow, Spark and MLflow, is still hard.

Credit: xkcd.com



R

R is a language and environment for statistical computing and graphics, with emphasis on making Data Science accessible to everyone and growing!



[Home]

Download

CRAN

R Project

About R

Logo

Contributors

What's New?

Reporting Bugs

Conferences

Search

Get Involved: Mailing Lists

Developer Pages

R Blog

R Foundation

Foundation

Board

Members

Donors

Donate

Help With R

Getting Help

Documentation

Manuals

FAQs

The R Journal

Books

What is R?

Introduction to R

R is a language and environment for statistical computing and graphics. It is a [GNU project](#) which is similar to the S language and environment which was developed at Bell Laboratories (formerly AT&T, now Lucent Technologies) by John Chambers and colleagues. R can be considered as a different implementation of S. There are some important differences, but much code written for S runs unaltered under R.

R provides a wide variety of statistical (linear and nonlinear modelling, classical statistical tests, time-series analysis, classification, clustering, ...) and graphical techniques, and is highly extensible. The S language is often the vehicle of choice for research in statistical methodology, and R provides an Open Source route to participation in that activity.

One of R's strengths is the ease with which well-designed publication-quality plots can be produced, including mathematical symbols and formulae where needed. Great care has been taken over the defaults for the minor design choices in graphics, but the user retains full control.

R is available as Free Software under the terms of the [Free Software Foundation's GNU General Public License](#) in source code form. It compiles and runs on a wide variety of UNIX platforms and similar systems (including FreeBSD and Linux), Windows and MacOs.

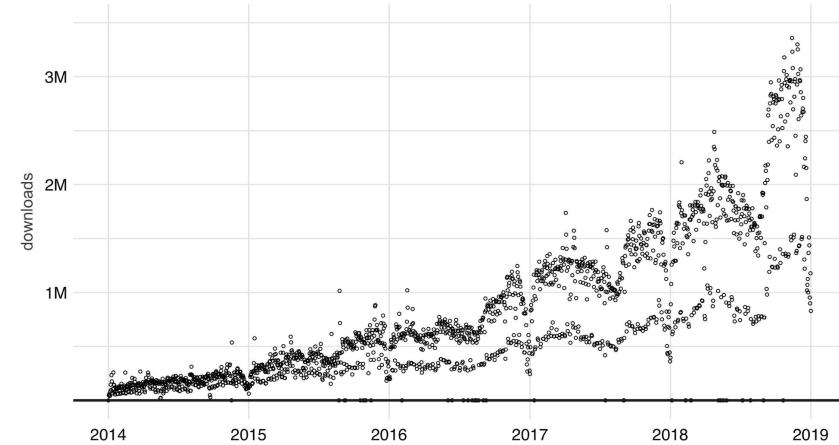
The R environment

R is an integrated suite of software facilities for data manipulation, calculation and graphical display. It includes

- an effective data handling and storage facility,
- a suite of operators for calculations on arrays, in particular matrices,
- a large, coherent, integrated collection of intermediate tools for data analysis,
- graphical facilities for data analysis and display either on-screen or on hardcopy, and
- a well-developed, simple and effective programming language which includes conditionals, loops, user-defined recursive functions and input and output facilities.

The term "environment" is intended to characterize it as a fully planned and coherent system, rather than an incremental accretion of very specific and inflexible tools, as is frequently the case with other data analysis software.

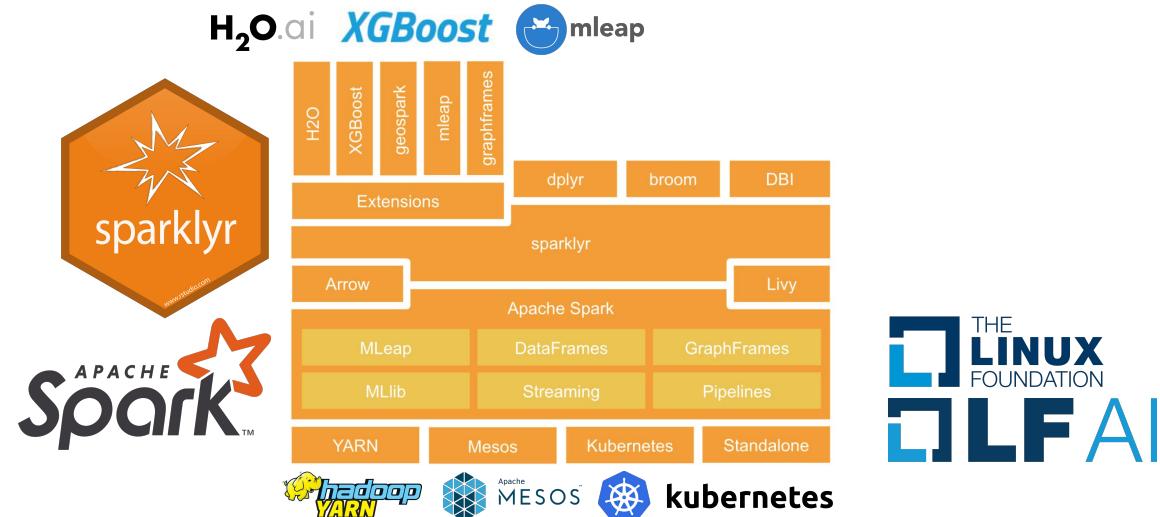
CRAN Packages
Total daily downloads over time



Credit: r-project.org, therinspark.com

Sparklyr

Sparklyr is an open-source and modern interface to scale data science and machine learning workflows using Apache Spark™ and R.



Sparklyr joins the Linux Foundation under LF AI in 2020.



Credit: sparklyr.ai, lfai.foundation

Ease of use

Sparklyr focuses on ease of use. You can install sparklyr and dependencies like Spark with one line of code, even in Windows:

```
# Install Spark  
install.packages("sparklyr")  
sparklyr::spark_install()
```

You can also then connect to local or remote Spark clusters with a line of code:

```
# Connect to Spark  
sc ← sparklyr::spark_connect(master = "local")
```

Comprehensive

```
spark_install()
sc <- spark_connect(master = "local")

cars <- spark_read_csv(sc, "cars", "input/")

summarize(cars, n = n())
dbGetQuery(sc, "SELECT count(*) FROM cars")

ml_linear_regression(cars, mpg ~ wt + cyl)

ml_pipeline(sc) %>%
  ft_r_formula(mpg ~ wt + cyl) %>%
  ml_linear_regression()

spark_context(sc) %>% invoke("version")
spark_apply(cars, nrow)

stream_read_csv(sc, "input/") %>%
  filter(mpg > 30) %>%
  stream_write_json("output/")

# Install local Spark
# Connect to Spark cluster

# Read data in Spark

# Count records with dplyr
# Count records with DBI

# Perform linear regression

# Define Spark pipeline
# Add formula transformation
# Add model to pipeline

# Extend sparklyr with Scala
# Extend sparklyr with R

# Define Spark stream
# Add dplyr transformation
# Start processing stream
```

Rich in Functionality

But you can also use sparklyr to train complex models using frameworks like TensorFlow and Spark.



```
library(sparklyr)
sc <- spark_connect(master = "local|yarn|etc")

# partition dataset
sdf_len(sc, 3, repartition = 3) %>%
  spark_apply(function(df, barrier) {
    library(tensorflow)
    library(keras)

    # define configuration from barrier
    Sys.setenv(TF_CONFIG = "")

    # define strategy and model
    strategy <- MultiWorkerMirroredStrategy()
    with (strategy$scope(), {
      model <- keras_model_sequential() # %>% ...
      model %>% compile()
    })

    # fit and retrieve model
    model %>% fit()
  }, barrier = TRUE) %>% collect()
```



Sponsors and Users

Sparklyr is supported by all major cloud providers, is sponsored by Databricks, Qubole and RStudio, serves many users and is hosted within LF AI.



Credit: sparklyr.ai

Ecosystem

Sparklyr makes use of Apache Spark, MLlib, MLeap, Apache Livy and Apache Arrow from R, then pipelines can be used to export to Python or Java.



Local development is encouraged before running analysis in Spark clusters.

Contributing

We have dozens of [major features](#) requested and hundreds of [issues](#), connect with us at github.com/sparklyr!

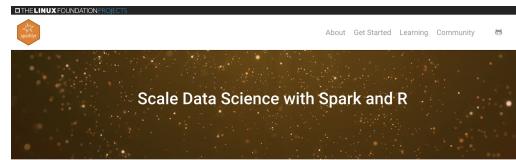
The screenshot shows the GitHub repository page for `sparklyr / sparklyr`. The top navigation bar includes links for Search or jump to..., Pull requests, Issues, Marketplace, and Explore. The repository header shows 82 Watchers, 739 Stars, and 273 Forks. Below the header, there are tabs for Code, Issues (477), Pull requests (3), Actions, Projects (1), Wiki, Security, Insights, and Settings. The main content area displays a list of recent commits:

Author	Commit Message	Time Ago
yl790	committed 90c9666 2 days ago	3 days ago
	replace spark master with 3.0.0 in CI workflow (#2575)	
	R handle custom scala_version correctly in all places (#2577)	2 days ago
	ci Add (arrow) to Suggests (#2544)	16 days ago
	docs/reference re-generate docs/references/*.html and revise .gitignore (#2410)	2 months ago
	inst support spark 2.4 built with scala 2.12 (#2570)	3 days ago
	java avoid floating point error in Date type serialization (#2563)	11 days ago
	man-roxygen allow passing of argument to underlying pipeline stages for ml_Ida	16 months ago
	man support spark 2.4 built with scala 2.12 (#2570)	3 days ago
	tests support spark 2.4 built with scala 2.12 (#2570)	3 days ago
	tools/readme attach expected output of the Databricks example in README.Rmd (#...	2 months ago
	.Rbuildignore rename ci script to .ci.R (#2481)	2 months ago
	.codecov.yml add .codecov.yml file to allow customizing	3 years ago
	.gitignore Revise gitignore (#2411)	2 months ago
	CODE_OF_CONDUCT.md Updated email contact	6 months ago

On the right side of the page, there are sections for About (R interface for Apache Spark), sparklyr.ai (rstudio, apache-spark, machine-learning, dplyr, sparklyr, remote-clusters, lily, ide, distributed, spark), Readme, Apache-2.0 License, Latest release (CRAN v1.2.0, 2 days ago), and Packages (No packages published, Publish your first package).

Resources

Thank you! Please learn more at sparklyr.ai, Mastering Spark with R, or the RStudio AI blog.

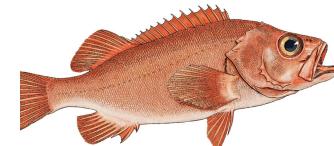


sparklyr.ai

O'REILLY®

Mastering Spark with R

The Complete Guide to Large-Scale Analysis and Modeling



Javier Luraschi,
Kevin Kuo & Edgar Ruiz
Foreword by Matei Zaharia



RStudio AI Blog

April 28, 2020

Towards privacy: Encrypted deep learning with Syft and Keras

Deep learning need not be incompatible with privacy protection. Federated learning enables on-device, end-to-end privacy protection for training models and gradient updates private differential privacy protects the training data from leaking. As of today, private and secure deep learning is an emerging technology. In this post, we introduce Syft, an open-source framework that integrates with PyTorch as well as TensorFlow. In an example use case, we obtain private predictions from a Keras model.

April 21, 2020

sparklyr 1.2: Foreach, Spark 3.0 and Databricks Connect

A new sparklyr release is now available. This sparklyr 1.2 release introduces several new features such as support for relational structures in databases such as support for Databricks Connect, a Spark connector for the sparklyr package, inter-op improvements for working with Spark 3.0 preview, as well as a number of bug fixes and improvements addressing user-visible pain points.

blogs.rstudio.com/ai