Accelerating Spark-ML with Redis modules

Dvir Volk, Shay Nativ

redislabs

SPARK SUMMIT
EUROPE 2016

Hello World

redis

Open source. The leading in-memory database

redislabs

The open source home and commercial provider of Redis - cloud and on-premise

dvirvolk

Senior System Architect at Redis Labs. Redis user and contributor for ~6 years

shaynativ

Senior Software Developer at Redis Labs

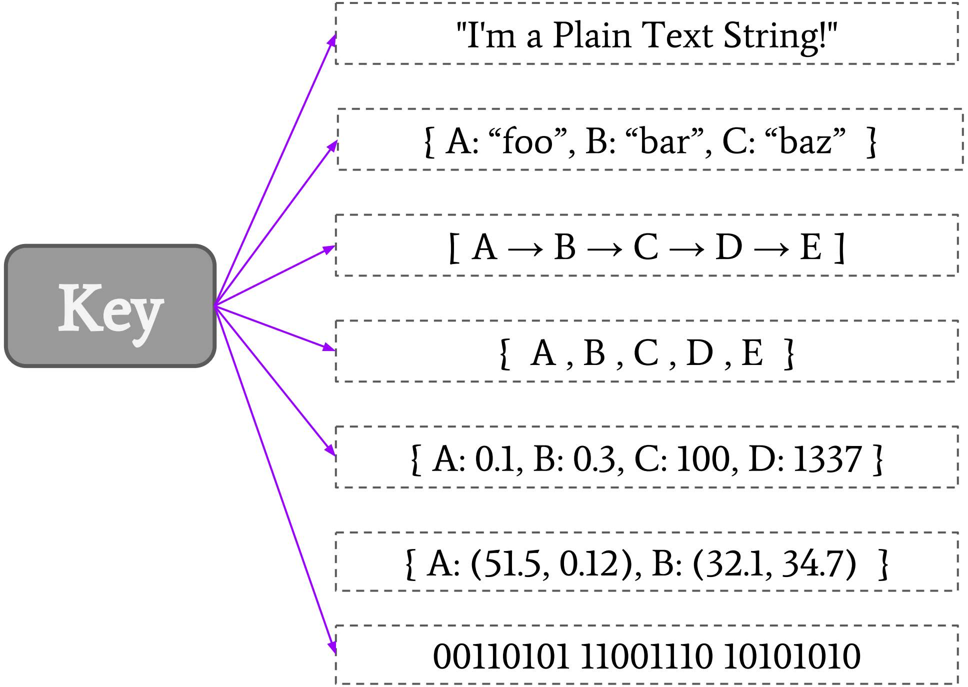
A Brief Overview of Redis

- Started in 2009 by Salvatore Sanfilippo
- Mostly a one man show
- Most popular KV store
- Notable Users:
    - Twitter, Netflix, Uber, Groupon, Twitch
    - Many, many more...

A Brief Overview of Redis

- **Key => Data Structure** server
- In memory disk backed
- Optional cluster mode
- Embedded Lua scripting
- Single Threaded!
- Key features: Fast, Flexible, Simple

A Lego For Your Database



Key

"I'm a Plain Text String!"	Strings/Blobs/Bitmaps
{ A: “foo”, B: “bar”, C: “baz” }	Hash Tables (objects!)
[A → B → C → D → E]	Linked Lists
{ A , B , C , D , E }	Sets
{ A: 0.1, B: 0.3, C: 100, D: 1337 }	Sorted Sets
{ A: (51.5, 0.12), B: (32.1, 34.7) }	Geo Sets
00110101 11001110 10101010	HyperLogLog

Redis In Practice

- "Front End Database"
- Real Time Counters
- Ad Serving
- Message Queues
- Geo Database
- Time Series
- Cache
- Session State
- Etc

Redis + Spark

- Spark-Redis connector
- Redis RDD
- SparkSQL integration
- Redis as a data source
- Redis as the final output

Full Text Search?

Secondary Index?

SQL?

Machine Learning?

But Can Redis Do X?

AutoComplete?

Graph?

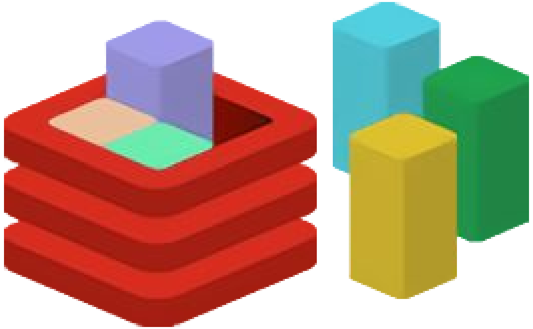
Time Series?

So You Want a New Feature?

- Try a Lua script
- Convince @antirez
- Fork Redis
- Build Your Own Database!

Enter Redis Modules

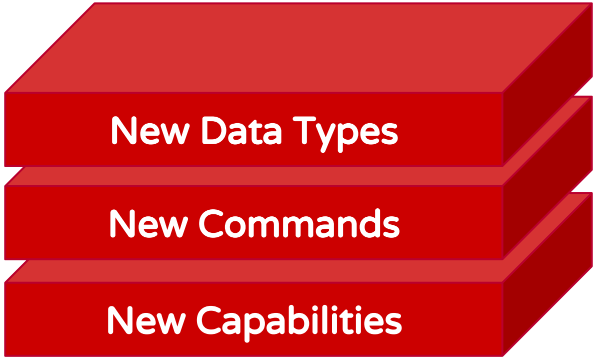
- In development since March 2016
- Redis 4.0 RC out soon
- Several modules already exist
- Key paradigm shift for Redis



Modules In Action

What Modules Actually Are

- Dynamic libraries loaded to redis
- Written in C/C++
- Use a C ABI/API isolating redis internals
- Near Zero latency access to data



New Data Types

New Commands

New Capabilities

Obligatory Module Example

LEFTPAD Example

```
127.0.0.1:6379> MODULE LOAD "./example.so"
OK
127.0.0.1:6379> COMMAND INFO EXAMPLE.LEFTPAD
1) 1) "example.leftpad"
...
127.0.0.1:6379> EXAMPLE.LEFTPAD "foo" 8
     foo
127.0.0.1:6379> EXAMPLE.LEFTPAD "foo" 8 "_"
_____foo
```

Real Module: RediSearch

- From-Scratch search index over redis
- Uses Strings for holding compressed index data
- Includes stemming, exact phrase match, etc.
- Fast Fuzzy Auto-complete
- Up to X5 faster than Elastic / Solr

```
> FT.SEARCH “lcd tv” FILTER price 100 +inf
> FT.SUGGET “lcd” FUZZY
```

Real Module: Indexing

- Support for secondary indexes for redis
- Supports indexing HASH keys with their properties
- Optional raw indexes as data types
- SQL-like syntax for querying indexes

```
> IDX.CREATE users_name_age TYPE HASH  SCHEMA name STRING age INT32
> IDX.INTO users_name_age  HMSET user1 name "alice" age 30
> IDX.FROM users_name_age  WHERE "name LIKE 'ali%' AND age < 31" HGETALL $
```

Real Module: JSON

- Stores JSON objects into redis
- Allows retrieval of part of a document
- Allows atomic manipulation of document elements

```
> JSON.SET foo ‘{“name”: {“first”: “bob”, “last”:”doe”},
  “age”: 32}`
> JSON.GET foo name.first
> JSON.SET foo age 33
```

Spark ML + Redis modules

Redis + Spark So Far

- ML is not addressed specifically
- Used for pre-computed results
- We felt that we can take it further

Addressing The ML Pain

- The missing piece of ML: **Serving your model**
  - Not standardized
  - Vendor-lock with cloud platforms
  - Reliable services are hard to do
  - If only we had a “database” for this!
  - Well, maybe we do?

Why Modules for ML?

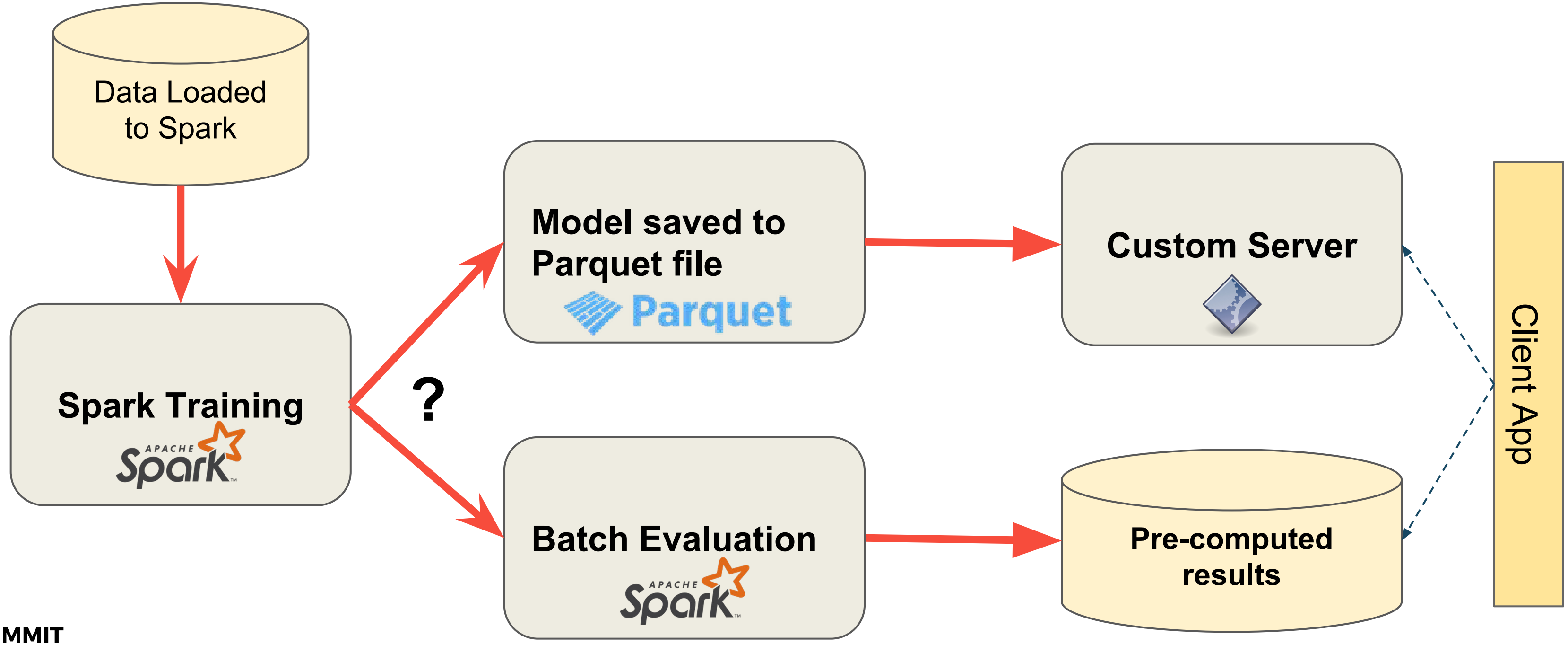
With modules we can:

- Define data structures for models
- Store training output as “hot model”
- Perform evaluation directly in Redis
- Easily integrate existing C/C++ libs

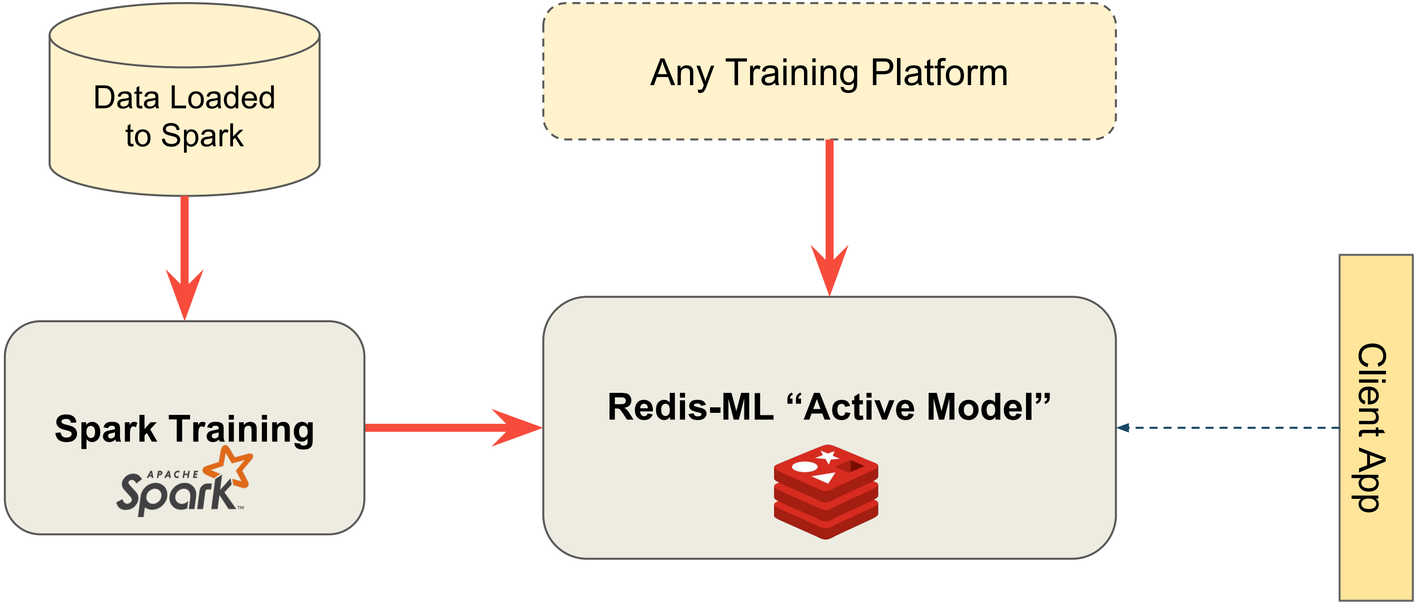
Spark + Modules = AWESOME

- Train ML model on Spark
- Save model to Redis and get:
  - High availability
  - Clustering
  - Persistence
  - Performance
  - Client libraries

Spark-ML End-to-End Flow



Adding Redis Into The Mix



Redis-ML Module

- Tree Ensembles
- Linear Regression
- Logistic Regression
- Matrix + Vector Operations
- More to come...

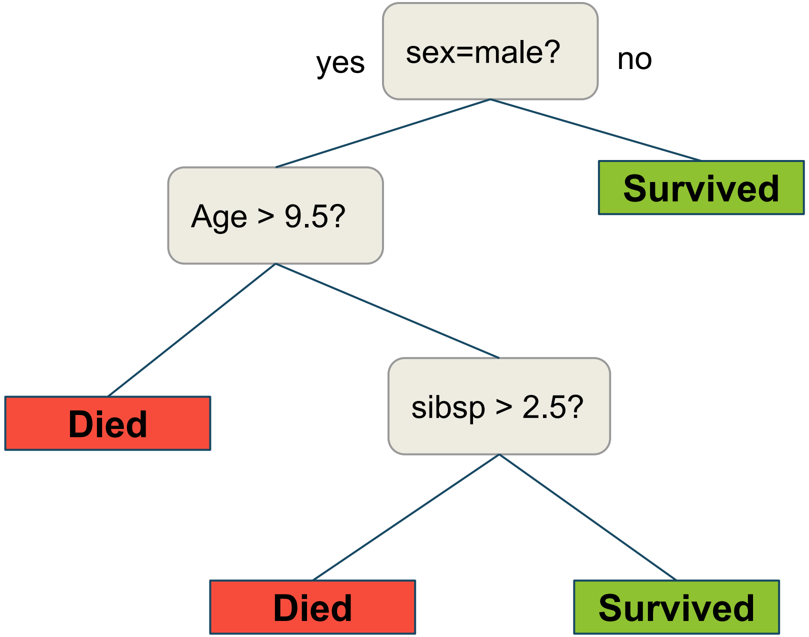
Example: Random Forest

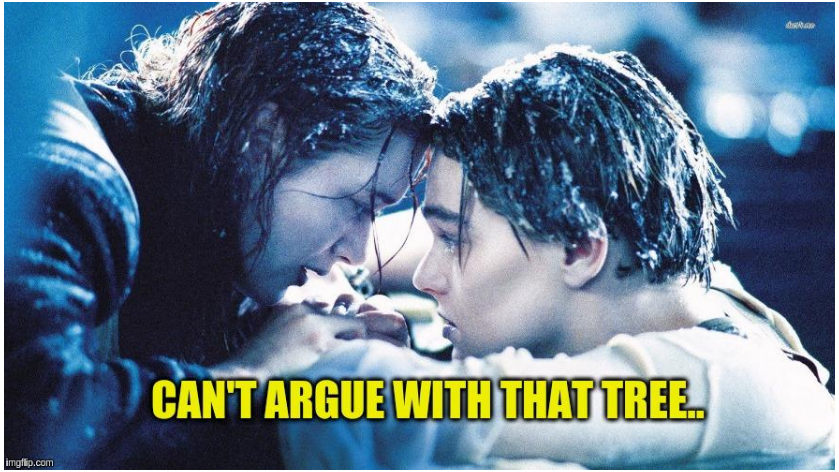
Forest Data Type

- A collection of decision trees
- Supports classification & regression
- Splitter Node can be
  - Categorical (e.g. day == “Sunday”)
  - Numerical (e.g. age < 43)

Decision Tree Example

The famous Titanic survival predictor





*sibsp = siblings + spouses

Forest Data Type API

Add nodes to a tree in a forest:

```
ML.FOREST.ADD <forestId> <treeId> <path>
    [ [NUMERIC|CATEGORIC] <splitterAttr> <splitterVal> ] |
    [LEAF] <predVal>
```

Perform classification/regression of a feature vector:

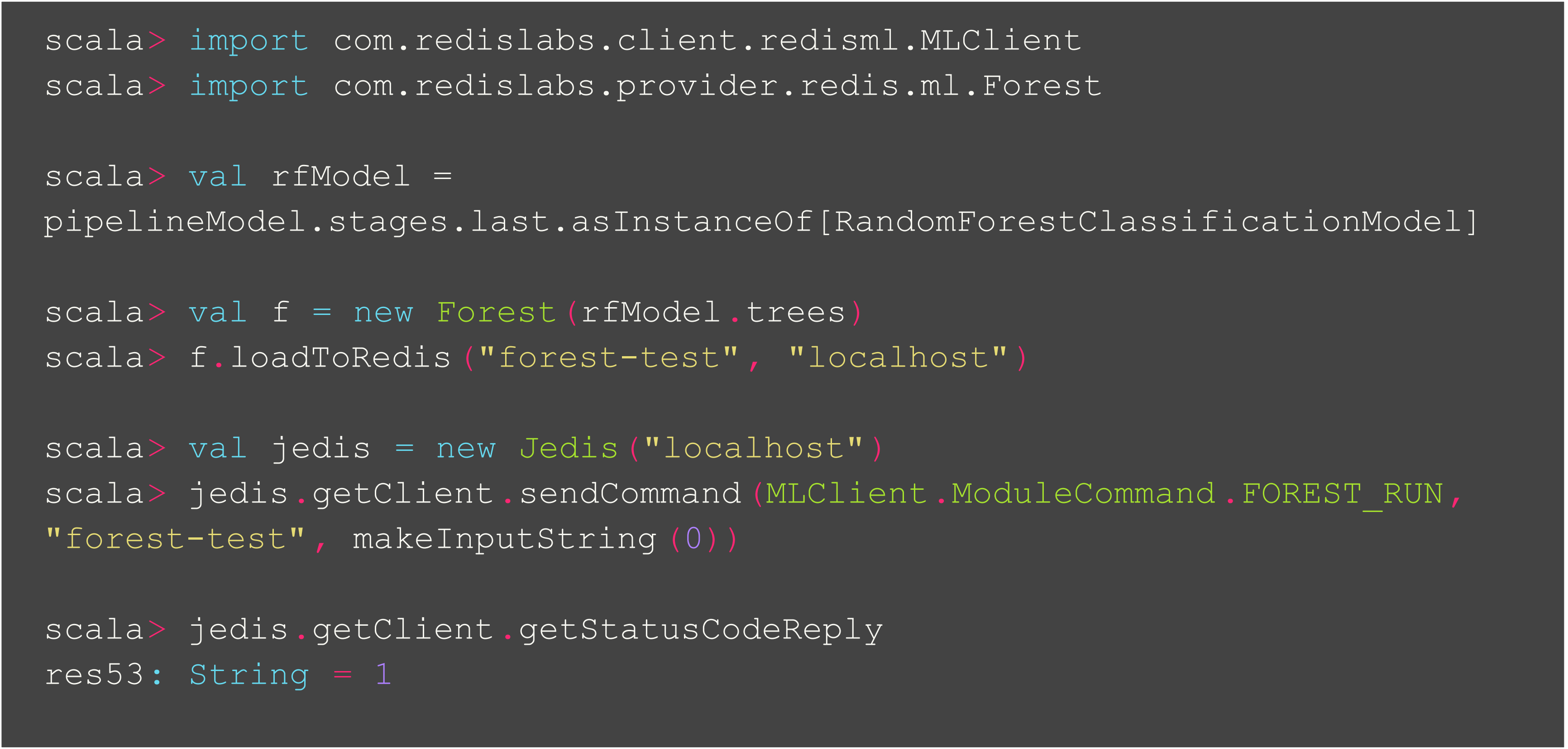
```
ML.FOREST.RUN <forestId> <features>
    [CLASSIFICATION|REGRESSION]
```

*feature vector is in libSVM format k:v k:v ...

Forest Data Type Example

```
> MODULE LOAD "./redis-ml.so"
OK
> ML.FOREST.ADD myforest 0   . CATEGORIC sex “male” .L
  LEAF 1 .R LEAF 0
OK
> ML.FOREST.RUN myforest sex:male
"1"
> ML.FOREST.RUN myforest sex:yes_please
"0"
```

Using Redis-ML With Spark



```
scala> import com.redislabs.client.redisml.MLClient
scala> import com.redislabs.provider.redis.ml.Forest

scala> val rfModel =
pipelineModel.stages.last.asInstanceOf[RandomForestClassificationModel]

scala> val f = new Forest(rfModel.trees)
scala> f.loadToRedis("forest-test", "localhost")

scala> val jedis = new Jedis("localhost")
scala> jedis.getClient.sendCommand(MLClient.ModuleCommand.FOREST_RUN,
"forest-test", makeInputString(0))

scala> jedis.getClient.getStatusCodeReply
res53: String = 1
```

Benchmarking Redis-ML

- Forest size: 15000 trees
- Data: $(SPARK_HOME)/data/mllib/sample_libsvm_data.txt

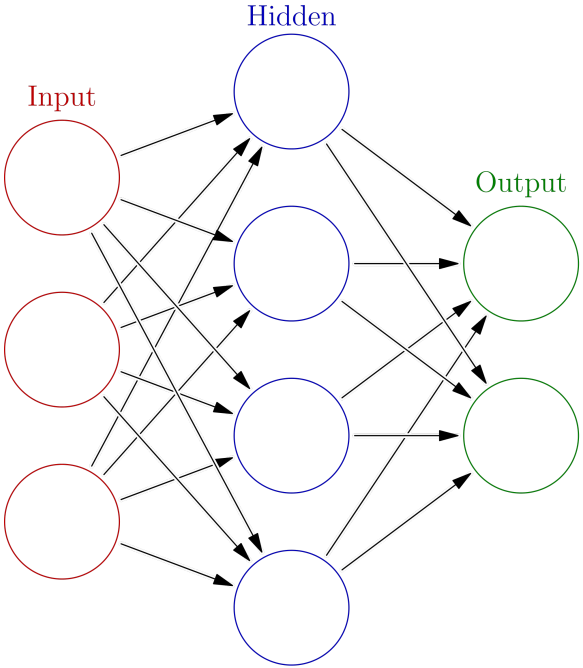
-	**Spark + Parquet**	**Spark + Redis ML**
Model Preparation + Save	3785ms	292ms
Model Load	2769ms	0ms (model is on memory)
Classification (AVG)	13ms	1ms

Going Forward - More Features

- Implement more Spark-ML model types
  - SVM
  - Naive Bayes Classifier
  - Neural Networks
- Integration with Redis' native types

PS: Neural Redis

- Developed by Salvatore
- Training is done inside redis
- Online continuous training process
- Builds Fully Connected NNs



More Resources

Redis-ML:
https://github.com/RedisLabsModules/redis-ml

Spark-Redis-ML:
https://github.com/RedisLabs/spark-redis-ml

Neural-Redis:
https://github.com/antirez/neural-redis

KTHXBAI!
quickmeme.com