

Julia for Scientific Computing

Antonio Ulloa, PhD

High Performance Computing Services

antonio.ulloa@nih.gov

April 7 2021



Download these slides from:
<https://hpc.nih.gov/training/handouts/julia4sc.pdf>

Outline

- Goals
 - Motivation
 - History of Julia
 - Julia vs others
 - Scientific Julia
 - Julia IDEs
 - Example: ML example using Flux
 - Conclusion

Outline

- Goals
- Motivation
- History of Julia
- Julia vs others
- Scientific Julia
- Julia IDEs
- Example: ML example using Flux
- Conclusion

Outline

- Goals
- Motivation
- History of Julia
- Julia vs others
- Scientific Julia
- Julia IDEs
- Example: ML example using Flux
- Conclusion

Outline

- Goals
- Motivation
- History of Julia
- Julia vs others
- Scientific Julia
- Julia IDEs
- Example: ML example using Flux
- Conclusion

Outline

- Goals
- Motivation
- History of Julia
- Julia vs others
- Scientific Julia
- Julia IDEs
- Example: ML example using Flux
- Conclusion

Outline

- Goals
- Motivation
- History of Julia
- Julia vs others
- Scientific Julia
- Julia IDEs
- Example: ML example using Flux
- Conclusion

Outline

- Goals
- Motivation
- History of Julia
- Julia vs others
- Scientific Julia
- Julia IDEs
- Example: ML example using Flux
- Conclusion

Outline

- Goals
- Motivation
- History of Julia
- Julia vs others
- Scientific Julia
- Julia IDEs
- Example: ML example using Flux
- Conclusion

Goals

At the end of the class, you should:

Goals

At the end of the class, you should:

1. have a general idea of the pros and cons of using Julia, and

Goals

At the end of the class, you should:

1. have a general idea of the pros and cons of using Julia, and
2. know how to run Julia scripts interactively on Biowulf.

Motivation

- Need for a language that can run fast but have a high-level syntax

Motivation

- Need for a language that can run fast but have a high-level syntax
- Desirable to be familiar with Julia given Julia usage growing in science

Motivation

- Need for a language that can run fast but have a high-level syntax
- Desirable to be familiar with Julia given Julia usage growing in science
- Need for a language that can make use of other language's packages

Motivation

- Need for a language that can run fast but have a high-level syntax
- Desirable to be familiar with Julia given Julia usage growing in science
- Need for a language that can make use of other language's packages
- Useful to be familiar with other scientific computation languages

History of Julia

- 2009: Four collaborators developed it from scratch with the idea of having a language that had high-level coding and fast performance.

History of Julia

- 2009: Four collaborators developed it from scratch with the idea of having a language that had high-level coding and fast performance.
- 2012: Julia released to developers

History of Julia

- 2009: Four collaborators developed it from scratch with the idea of having a language that had high-level coding and fast performance.
- 2012: Julia released to developers
- 2014: JuliaCon academic conference starts

History of Julia

- 2009: Four collaborators developed it from scratch with the idea of having a language that had high-level coding and fast performance.
- 2012: Julia released to developers
- 2014: JuliaCon academic conference starts
- 2018: Julia 1.0 released

History of Julia

- 2009: Four collaborators developed it from scratch with the idea of having a language that had high-level coding and fast performance.
- 2012: Julia released to developers
- 2014: JuliaCon academic conference starts
- 2018: Julia 1.0 released
- 2020: Julia for VS Code extension 1.0 released

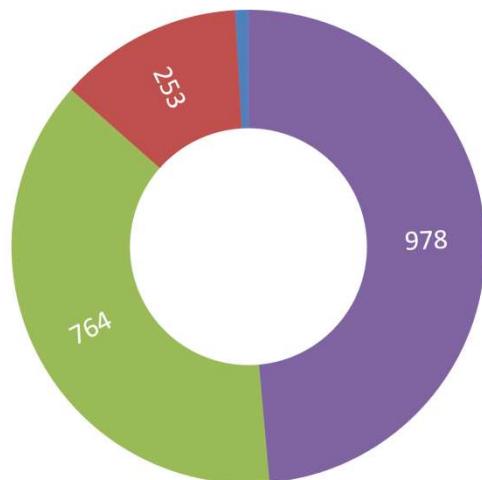
History of Julia

- 2009: Four collaborators developed it from scratch with the idea of having a language that had high-level coding and fast performance.
- 2012: Julia released to developers
- 2014: JuliaCon academic conference starts
- 2018: Julia 1.0 released
- 2020: Julia for VS Code extension 1.0 released
- 2020: Julia 1.5 released

Julia vs Matlab/Python/R

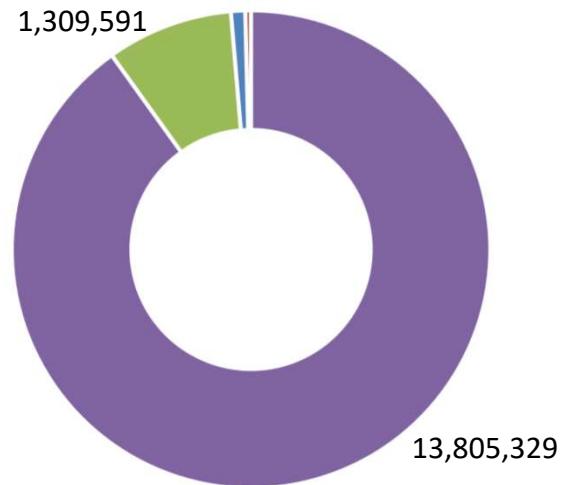
Unique users

■ Julia ■ Matlab ■ Python ■ R



Module loads

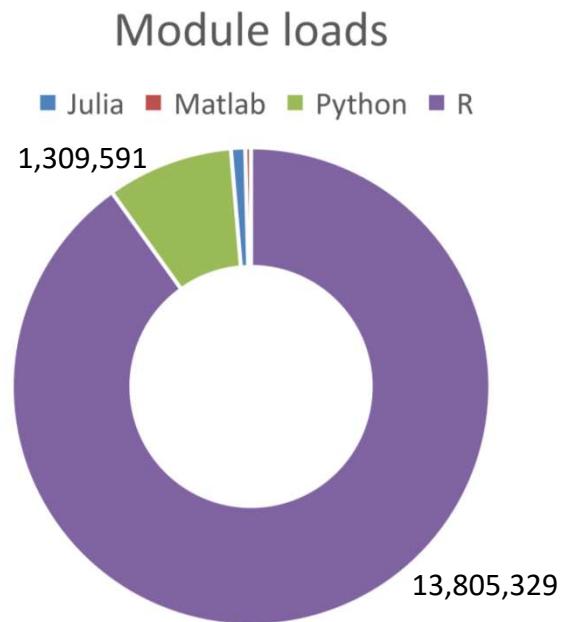
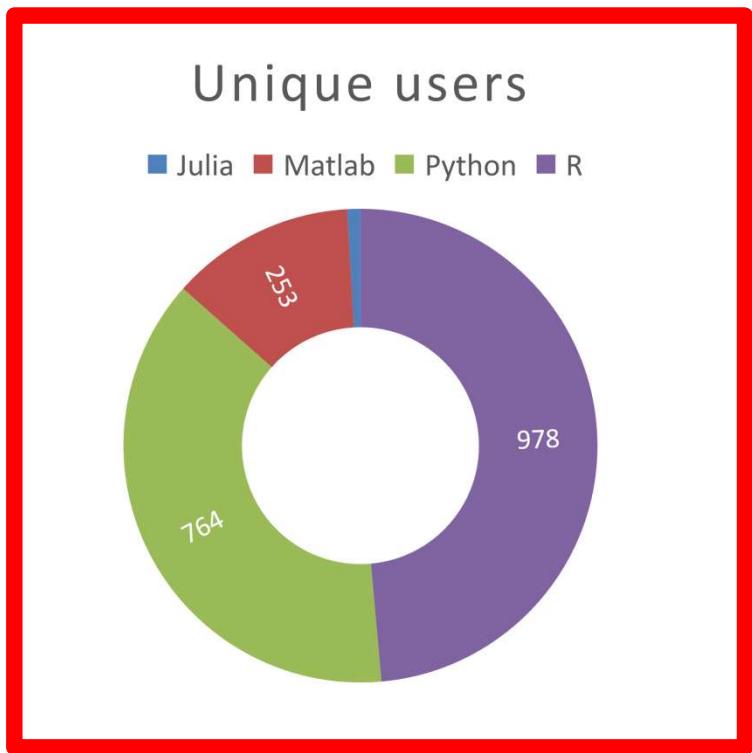
■ Julia ■ Matlab ■ Python ■ R



*Biowulf usage for the year ended on 16 March 2021

Source: Internal NIH-HPC database

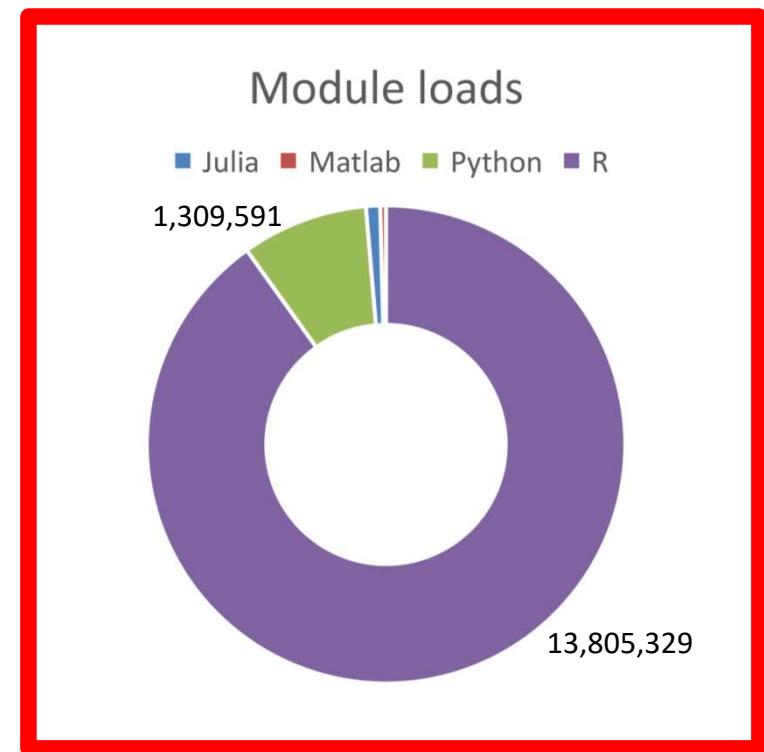
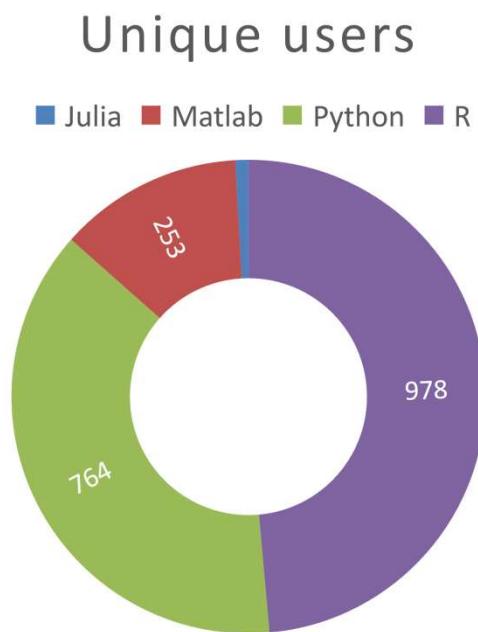
Julia vs Matlab/Python/R



*Biowulf usage for the year ended on 16 March 2021

Source: Internal NIH-HPC database

Julia vs Matlab/Python/R



*Biowulf usage for the year ended on 16 March 2021

Source: Internal NIH-HPC database

Julia vs Matlab/Python/R

Worldwide, Mar 2021 compared to a year ago:

Rank	Change	Language	Share	Trend
1		Python	30.17 %	-0.2 %
2		Java	17.18 %	-1.2 %
3		JavaScript	8.21 %	+0.2 %
4		C#	6.76 %	-0.6 %
5	↑	C/C++	6.71 %	+0.8 %
6	↓	PHP	6.13 %	+0.0 %
7		R	3.81 %	+0.0 %
8		Objective-C	3.56 %	+1.1 %
9		Swift	1.82 %	-0.4 %
10	↑	Matlab	1.8 %	-0.0 %
11	↑	Kotlin	1.76 %	+0.2 %
12	↓↓	TypeScript	1.74 %	-0.1 %
13	↑	Go	1.34 %	+0.0 %
14	↓	VBA	1.22 %	-0.1 %
15		Ruby	1.13 %	-0.1 %
16	↑↑	Rust	1.13 %	+0.5 %
17	↑↑↑↑↑↑	Ada	0.68 %	+0.4 %
18	↓	Visual Basic	0.67 %	-0.3 %
19	↓↓↓	Scala	0.66 %	-0.4 %
20	↑↑↑↑↑	Lua	0.55 %	+0.2 %
21	↓	Dart	0.53 %	+0.1 %
22	↓↓↓	Abap	0.44 %	-0.1 %
23	↓↓	Perl	0.44 %	-0.0 %
24	↓↓	Groovy	0.4 %	-0.0 %
25		Cobol	0.36 %	+0.1 %
26	↑	Julia	0.35 %	+0.1 %

*Index of language popularity based on frequency of tutorial searches on google.

Reproduced from: PYPL Index, <https://pypl.github.io/PYPL.html>

Julia vs Matlab/Python/R

Worldwide, Mar 2021 compared to a year ago:				
1	Python	30.17 %	-0.2 %	
2	Java	11.18 %	+0.2 %	
3	JavaScript	8.21 %	+0.2 %	
4	C#	6.76 %	-0.6 %	
5	↑ C/C++	6.71 %	+0.8 %	
6	↓ PHP	6.13 %	+0.0 %	
7	R	3.81 %	+0.0 %	
8	Objective-C	3.56 %	+1.1 %	
9	Swift	1.82 %	-0.4 %	
10	↑ Matlab	1.8 %	-0.0 %	
11	↑ Kotlin	1.76 %	+0.2 %	
12	↓↓ TypeScript	1.74 %	-0.1 %	
13	↑ Go	1.34 %	+0.0 %	
14	↓ VBA	1.22 %	-0.1 %	
15	Ruby	1.13 %	-0.1 %	
16	↑↑ Rust	1.13 %	+0.5 %	
17	↑↑↑↑↑ Ada	0.68 %	+0.4 %	
18	↓ Visual Basic	0.67 %	-0.3 %	
19	↓↓↓ Scala	0.66 %	-0.4 %	
20	↑↑↑↑↑ Lua	0.55 %	+0.2 %	
21	↓ Dart	0.53 %	+0.1 %	
22	↓↓↓ Abap	0.44 %	-0.1 %	
23	↓↓ Perl	0.44 %	-0.0 %	
24	↓↓ Groovy	0.4 %	-0.0 %	
25	Cobol	0.36 %	+0.1 %	
26	↑ Julia	0.35 %	+0.1 %	

*Index of language popularity based on frequency of tutorial searches on google.

Reproduced from: PYPL Index, <https://pypl.github.io/PYPL.html>

Julia vs Matlab/Python/R

Worldwide, Mar 2021 compared to a year ago:				
Rank	Change	Language	Share	Trend
1		Python	30.17 %	-0.2 %
2		Java	17.18 %	-1.2 %
3		JavaScript	8.21 %	+0.2 %
4		C#	6.76 %	-0.6 %
5	↑	C/C++	6.71 %	+0.8 %
7		R	3.81 %	+0.0 %
8		Objective-C	3.55 %	+1.1 %
9		Swift	1.82 %	-0.4 %
10	↑	Matlab	1.8 %	-0.0 %
11	↑	Kotlin	1.76 %	+0.2 %
12	↓↓	TypeScript	1.74 %	-0.1 %
13	↑	Go	1.34 %	+0.0 %
14	↓	VBA	1.22 %	-0.1 %
15		Ruby	1.13 %	-0.1 %
16	↑↑	Rust	1.13 %	+0.5 %
17	↑↑↑↑↑↑↑	Ada	0.68 %	+0.4 %
18	↓	Visual Basic	0.67 %	-0.3 %
19	↓↓↓	Scala	0.66 %	-0.4 %
20	↑↑↑↑↑	Lua	0.55 %	+0.2 %
21	↓	Dart	0.53 %	+0.1 %
22	↓↓↓	Abap	0.44 %	-0.1 %
23	↓↓	Perl	0.44 %	-0.0 %
24	↓↓	Groovy	0.4 %	-0.0 %
25		Cobol	0.36 %	+0.1 %
26	↑	Julia	0.35 %	+0.1 %

*Index of language popularity based on frequency of tutorial searches on google.

Reproduced from: PYPL Index, <https://pypl.github.io/PYPL.html>

Julia vs Matlab/Python/R

Worldwide, Mar 2021 compared to a year ago:				
Rank	Change	Language	Share	Trend
1		Python	30.17 %	-0.2 %
2		Java	17.18 %	-1.2 %
3		JavaScript	8.21 %	+0.2 %
4		C#	6.76 %	-0.6 %
5	↑	C/C++	6.71 %	+0.8 %
6	↓	PHP	6.13 %	+0.0 %
7		R	3.81 %	+0.0 %
8		Objective-C	3.56 %	+1.1 %
10	↑	Matlab	1.8 %	-0.0 %
11		Julia	1.6 %	+0.1 %
12	↓↓	TypeScript	1.74 %	-0.1 %
13	↑	Go	1.34 %	+0.0 %
14	↓	VBA	1.22 %	-0.1 %
15		Ruby	1.13 %	-0.1 %
16	↑↑	Rust	1.13 %	+0.5 %
17	↑↑↑↑↑↑↑	Ada	0.68 %	+0.4 %
18	↓	Visual Basic	0.67 %	-0.3 %
19	↓↓↓	Scala	0.66 %	-0.4 %
20	↑↑↑↑↑	Lua	0.55 %	+0.2 %
21	↓	Dart	0.53 %	+0.1 %
22	↓↓↓	Abap	0.44 %	-0.1 %
23	↓↓	Perl	0.44 %	-0.0 %
24	↓↓	Groovy	0.4 %	-0.0 %
25		Cobol	0.36 %	+0.1 %
26	↑	Julia	0.35 %	+0.1 %

*Index of language popularity based on frequency of tutorial searches on google.

Reproduced from: PYPL Index, <https://pypl.github.io/PYPL.html>

Julia vs Matlab/Python/R

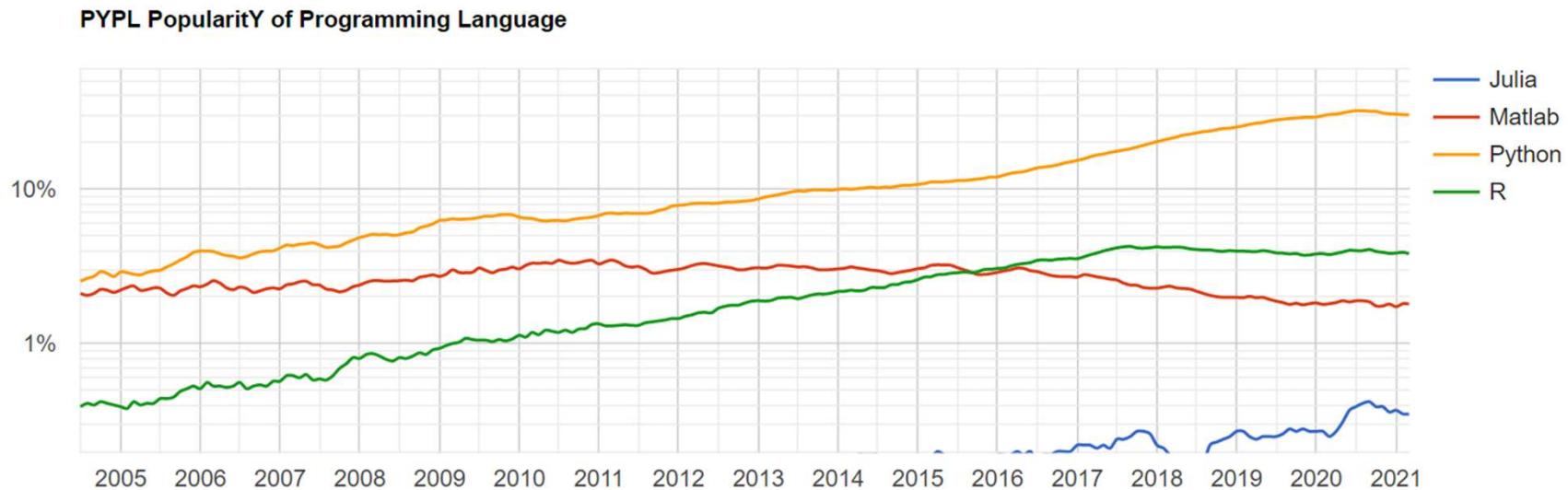
Worldwide, Mar 2021 compared to a year ago:

Rank	Change	Language	Share	Trend
1		Python	30.17 %	-0.2 %
2		Java	17.18 %	-1.2 %
3		JavaScript	8.21 %	+0.2 %
4		C#	6.76 %	-0.6 %
5	↑	C/C++	6.71 %	+0.8 %
6	↓	PHP	6.13 %	+0.0 %
7		R	3.81 %	+0.0 %
8		Objective-C	3.56 %	+1.1 %
9		Swift	1.82 %	-0.4 %
10	↑	Matlab	1.8 %	-0.0 %
11	↑	Kotlin	1.76 %	+0.2 %
12	↓↓	TypeScript	1.74 %	-0.1 %
13	↑	Go	1.34 %	+0.0 %
14	↓	VBA	1.22 %	-0.1 %
15		Ruby	1.13 %	-0.1 %
16	↑↑	Rust	1.13 %	+0.5 %
17	↑↑↑↑↑↑	Ada	0.68 %	+0.4 %
18	↓	Visual Basic	0.67 %	-0.3 %
19	↓↓↓	Scala	0.66 %	-0.4 %
20	↑↑↑↑	Lua	0.55 %	+0.2 %
21	↓	Dart	0.53 %	+0.1 %
22	↓↓↓	Abap	0.44 %	-0.1 %
23	↓↓	Perl	0.44 %	-0.0 %
24	↓↓	Groovy	0.4 %	-0.0 %
26	↑	Julia	0.35 %	+0.1 %

*Index of language popularity based on frequency of tutorial searches on google.

Reproduced from: PYPL Index, <https://pypl.github.io/PYPL.html>

Julia vs Matlab/Python/R



Reproduced from: PYPL Index, <https://pypl.github.io/PYPL.html>

Julia vs MATLAB/Python/R

	Total Cumulative as of Jan 1, 2020	Total Cumulative as of Jan 1, 2021	Change
Number of Julia Downloads (JuliaLang + Docker + JuliaPro)	12,950,630	24,205,141	+87%
Number of Julia Packages	2,787	4,809	+73%
GitHub stars (Julia language repo + registered packages)	99,830	161,774	+62%
YouTube views (Julia language channel)	1,562,223	3,320,915	+113%
Published citations of Julia: A Fast Dynamic Language for Technical Computing (2012) + Julia: A Fresh Approach to Numerical Computing (2017)	1,680	2,531	+51%
Discourse posts	137,399	211,888	+54%
TIOBE Index Rank	#47	#23	+24

Reproduced from: *Julia Computing*,
<https://juliacomputing.com/blog/2021/01/newsletter-january/>

Julia vs MATLAB/Python/R

	Total Cumulative as of Jan 1, 2020	Total Cumulative as of Jan 1, 2021	Change
Number of Julia Downloads (JuliaLang + Docker + JuliaPro)	12,950,630	24,205,141	+87%
Number of Julia Packages	2,787	4,809	+73%
GitHub stars (Julia language repo + registered packages)	99,830	161,774	+62%
YouTube views (Julia language channel)	1,562,223	3,320,915	+113%
Published citations of Julia: A Fast Dynamic Language for Technical Computing (2012) + Julia: A Fresh Approach to Numerical Computing (2017)	1,680	2,531	+51%
Discourse posts	137,399	211,888	+54%
TIOBE Index Rank	#47	#23	+24

Reproduced from: *Julia Computing*,
<https://juliacomputing.com/blog/2021/01/newsletter-january/>

Julia vs MATLAB/Python/R

	Total Cumulative as of Jan 1, 2020	Total Cumulative as of Jan 1, 2021	Change
Number of Julia Downloads (JuliaLang + Docker + JuliaPro)	12,950,630	24,205,141	+87%
Number of Julia Packages	2,787	4,809	+73%
GitHub stars (Julia language repo + registered packages)	99,830	161,774	+62%
YouTube views (Julia language channel)	1,562,223	3,320,915	+113%
Published citations of Julia: A Fast Dynamic Language for Technical Computing (2012) + Julia: A Fresh Approach to Numerical Computing (2017)	1,680	2,531	+51%
Discourse posts	137,399	211,888	+54%
TIOBE Index Rank	#47	#23	+24

Reproduced from: Julia Computing,
<https://juliacomputing.com/blog/2021/01/newsletter-january/>

Julia vs MATLAB/Python/R

	Total Cumulative as of Jan 1, 2020	Total Cumulative as of Jan 1, 2021	Change
Number of Julia Downloads (JuliaLang + Docker + JuliaPro)	12,950,630	24,205,141	+87%
Number of Julia Packages	2,787	4,809	+73%
GitHub stars (Julia language repo + registered packages)	99,830	161,774	+62%
YouTube views (Julia language channel)	1,562,223	3,320,915	+113%
Published citations of Julia: A Fast Dynamic Language for Technical Computing (2012) + Julia: A Fresh Approach to Numerical Computing (2017)	1,680	2,531	+51%
Discourse posts	137,399	211,888	+54%
TIOBE Index Rank	#47	#23	+24

Reproduced from: *Julia Computing*,
<https://juliacomputing.com/blog/2021/01/newsletter-january/>

Julia vs MATLAB/Python/R

	Total Cumulative as of Jan 1, 2020	Total Cumulative as of Jan 1, 2021	Change
Number of Julia Downloads (JuliaLang + Docker + JuliaPro)	12,950,630	24,205,141	+87%
Number of Julia Packages	2,787	4,809	+73%
GitHub stars (Julia language repo + registered packages)	99,830	161,774	+62%
YouTube views (Julia language channel)	1,562,223	3,320,915	+113%
Published citations of Julia: A Fast Dynamic Language for Technical Computing (2012) + Julia: A Fresh Approach to Numerical Computing (2017)	1,680	2,531	+51%
Discourse posts	137,399	211,888	+54%
TIOBE Index Rank	#47	#23	+24

Reproduced from: Julia Computing,
<https://juliacomputing.com/blog/2021/01/newsletter-january/>

Julia vs MATLAB/Python/R

	Total Cumulative as of Jan 1, 2020	Total Cumulative as of Jan 1, 2021	Change
Number of Julia Downloads (JuliaLang + Docker + JuliaPro)	12,950,630	24,205,141	+87%
Number of Julia Packages	2,787	4,809	+73%
GitHub stars (Julia language repo + registered packages)	99,830	161,774	+62%
YouTube views (Julia language channel)	1,562,223	3,320,915	+113%
Published citations of Julia: A Fast Dynamic Language for Technical Computing (2012) + Julia: A Fresh Approach to Numerical Computing (2017)	1,680	2,531	+51%
Discourse posts	137 399	211 888	+54%
TIOBE Index Rank	#47	#23	+24

Reproduced from: Julia Computing,
<https://juliacomputing.com/blog/2021/01/newsletter-january/>

Julia vs MATLAB/Python/R

	Julia	Matlab	Python	R
First release	2009	1984	1991	1995
No. packages	4,809	64+454	292,782	17,319
Current version	1.5.4	2021a	3.9.2	4.0.4
Package repos	Github registry	Mathworks	Pypi, ...	CRAN, ...
IDEs	Juno, VSCode	MATLAB IDE	Spyder, vscode, Pycharm	Rstudio
Comp. process	Compiled	Interpreted	Interpreted	Interpreted
Code structure	Free style	Free style	Indented	Free style
Typing	Dynamic	Dynamic	Dynamic	Dynamic
ML/DL	Flux, Knet	ML/DL toolboxes	Tensorflow, Keras, Pytorch	Keras

Sources: julialang.org, mathworks.com, python.org, r-project.org,
https://www.openriskmanual.org/wiki/Overview_of_the_Julia-Python-R_Universe, <https://hyperpolyglot.org/numerical-analysis>, juliapackages.com

Julia vs MATLAB/Python/R

	Julia	Matlab	Pvthon	R
First release	2009	1984	1991	1995
No. packages	4,809	64+454	292,782	17,319
Current version	1.5.4	2021a	3.9.2	4.0.4
Package repos	Github registry	Mathworks	Pypi, ...	CRAN, ...
IDEs	Juno, VSCode	MATLAB IDE	Spyder, vscode, Pycharm	Rstudio
Comp. process	Compiled	Interpreted	Interpreted	Interpreted
Code structure	Free style	Free style	Indented	Free style
Typing	Dynamic	Dynamic	Dynamic	Dynamic
ML/DL	Flux, Knet	ML/DL toolboxes	Tensorflow, Keras, Pytorch	Keras

Sources: julialang.org, mathworks.com, python.org, r-project.org,
https://www.openriskmanual.org/wiki/Overview_of_the_Julia-Python-R_Universe, <https://hyperpolyglot.org/numerical-analysis>, juliapackages.com

Julia vs MATLAB/Python/R

	Julia	Matlab	Python	R
First release	2009	1984	1991	1995
No. packages	4,809	64+454	292,782	17,319
Current version	1.5.4	2021a	3.9.2	4.0.4
Package repos	Github registry	Mathworks	Pypi, ...	CRAN, ...
IDEs	Juno, VSCode	MATLAB IDE	Spyder, vscode, Pycharm	Rstudio
Comp. process	Compiled	Interpreted	Interpreted	Interpreted
Code structure	Free style	Free style	Indented	Free style
Typing	Dynamic	Dynamic	Dynamic	Dynamic
ML/DL	Flux, Knet	ML/DL toolboxes	Tensorflow, Keras, Pytorch	Keras

Sources: julialang.org, mathworks.com, python.org, r-project.org,
https://www.openriskmanual.org/wiki/Overview_of_the_Julia-Python-R_Universe, <https://hyperpolyglot.org/numerical-analysis>, juliapackages.com

Julia vs MATLAB/Python/R

	Julia	Matlab	Python	R
First release	2009	1984	1991	1995
No. packages	4.809	64+454	292.782	17.319
Current version	1.5.4	2021a	3.9.2	4.0.4
Package repos	Github registry	Mathworks	Pypi, ...	CRAN, ...
IDEs	Juno, VSCode	MATLAB IDE	Spyder, vscode, Pycharm	Rstudio
Comp. process	Compiled	Interpreted	Interpreted	Interpreted
Code structure	Free style	Free style	Indented	Free style
Typing	Dynamic	Dynamic	Dynamic	Dynamic
ML/DL	Flux, Knet	ML/DL toolboxes	Tensorflow, Keras, Pytorch	Keras

Sources: julialang.org, mathworks.com, python.org, r-project.org,
https://www.openriskmanual.org/wiki/Overview_of_the_Julia-Python-R_Universe, <https://hyperpolyglot.org/numerical-analysis>, juliapackages.com

Julia vs MATLAB/Python/R

	Julia	Matlab	Python	R
First release	2009	1984	1991	1995
No. packages	4,809	64+454	292,782	17,319
Current version	1.5.4	2021a	3.9.2	4.0.4
Package repos	Github registry	Mathworks	Pypi, ...	CRAN, ...
IDEs	Juno, VSCode	MATLAB IDE	Spyder, vscode, Pycharm	Rstudio
Comp. process	Compiled	Interpreted	Interpreted	Interpreted
Code structure	Free style	Free style	Indented	Free style
Typing	Dynamic	Dynamic	Dynamic	Dynamic
ML/DL	Flux, Knet	ML/DL toolboxes	Tensorflow, Keras, Pytorch	Keras

Sources: julialang.org, mathworks.com, python.org, r-project.org,
https://www.openriskmanual.org/wiki/Overview_of_the_Julia-Python-R_Universe, <https://hyperpolyglot.org/numerical-analysis>, juliapackages.com

Julia vs MATLAB/Python/R

	Julia	Matlab	Python	R
First release	2009	1984	1991	1995
No. packages	4,809	64+454	292,782	17,319
Current version	1.5.4	2021a	3.9.2	4.0.4
Package repos	Github registry	Mathworks	Pypi, ...	CRAN, ...
IDEs	Juno, VSCode	MATLAB IDE	Spyder, vscode, Pycharm	Rstudio
Comp. process	Compiled	Interpreted	Interpreted	Interpreted
Code structure	Free style	Free style	Indented	Free style
Typing	Dynamic	Dynamic	Dynamic	Dynamic
ML/DL	Flux, Knet	ML/DL toolboxes	Tensorflow, Keras, Pytorch	Keras

Sources: julialang.org, mathworks.com, python.org, r-project.org,
https://www.openriskmanual.org/wiki/Overview_of_the_Julia-Python-R_Universe, <https://hyperpolyglot.org/numerical-analysis>, juliapackages.com

Julia vs MATLAB/Python/R

	Julia	Matlab	Python	R
First release	2009	1984	1991	1995
No. packages	4,809	64+454	292,782	17,319
Current version	1.5.4	2021a	3.9.2	4.0.4
Package repos	Github registry	Mathworks	Pypi, ...	CRAN, ...
IDEs	Juno, VSCode	MATLAB IDE	Spyder, vscode, Pycharm	Rstudio
Comp. process	Compiled	Interpreted	Interpreted	Interpreted
Code structure	Free style	Free style	Indented	Free style
Typing	Dynamic	Dynamic	Dynamic	Dynamic
ML/DL	Flux, Knet	ML/DL toolboxes	Tensorflow, Keras, Pytorch	Keras

Sources: julialang.org, mathworks.com, python.org, r-project.org,
https://www.openriskmanual.org/wiki/Overview_of_the_Julia-Python-R_Universe, <https://hyperpolyglot.org/numerical-analysis>, juliapackages.com

Julia vs MATLAB/Python/R

	Julia	Matlab	Python	R
First release	2009	1984	1991	1995
No. packages	4,809	64+454	292,782	17,319
Current version	1.5.4	2021a	3.9.2	4.0.4
Package repos	Github registry	Mathworks	Pypi, ...	CRAN, ...
IDEs	Juno, VSCode	MATLAB IDE	Spyder, vscode, Pycharm	Rstudio
Comp. process	Compiled	Interpreted	Interpreted	Interpreted
Code structure	Free style	Free style	Indented	Free style
Typing	Dynamic	Dynamic	Dynamic	Dynamic
ML/DL	Flux, Knet	ML/DL toolboxes	Tensorflow, Keras, Pytorch	Keras

Sources: julialang.org, mathworks.com, python.org, r-project.org,
https://www.openriskmanual.org/wiki/Overview_of_the_Julia-Python-R_Universe, <https://hyperpolyglot.org/numerical-analysis>, juliapackages.com

Julia vs MATLAB/Python/R

	Julia	Matlab	Python	R
First release	2009	1984	1991	1995
No. packages	4,809	64+454	292,782	17,319
Current version	1.5.4	2021a	3.9.2	4.0.4
Package repos	Github registry	Mathworks	Pypi, ...	CRAN, ...
IDEs	Juno, VSCode	MATLAB IDE	Spyder, vscode, Pycharm	Rstudio
Comp. process	Compiled	Interpreted	Interpreted	Interpreted
Code structure	Free style	Free style	Indented	Free style
Typing	Dynamic	Dynamic	Dynamic	Dynamic
ML/DL	Flux, Knet	ML/DL toolboxes	Tensorflow, Keras, Pytorch	Keras

Sources: julialang.org, mathworks.com, python.org, r-project.org,
https://www.openriskmanual.org/wiki/Overview_of_the_Julia-Python-R_Universe, <https://hyperpolyglot.org/numerical-analysis>, juliapackages.com

Julia vs MATLAB/Python/R

	Julia	Matlab	Python	R
First release	2009	1984	1991	1995
No. packages	4,809	64+454	292,782	17,319
Current version	1.5.4	2021a	3.9.2	4.0.4
Package repos	Github registry	Mathworks	Pypi, ...	CRAN, ...
IDEs	Juno, VSCode	MATLAB IDE	Spyder, vscode, Pycharm	Rstudio
Comp. process	Compiled	Interpreted	Interpreted	Interpreted
Code structure	Free style	Free style	Indented	Free style
Typing	Dynamic	Dynamic	Dynamic	Dynamic
ML/DL	Flux, Knet	ML/DL toolboxes	Tensorflow, Keras, Pytorch	Keras

Sources: julialang.org, mathworks.com, python.org, r-project.org,
https://www.openriskmanual.org/wiki/Overview_of_the_Julia-Python-R_Universe, <https://hyperpolyglot.org/numerical-analysis>, juliapackages.com

Julia vs MATLAB/Python/R

	Julia	Matlab	Python	R
Notebooks	Jupyter	Live Editor	Jupyter	Jupyter, R Markdown
Visualization	Plots.jl	Built-in	Matplotlib, seaborn, bokeh, plotly	ggplot
Package install	Built-in pkg manager	Installer	pip, conda, ...	Built-in install.packages , ...
Package load	using <pkg>, import <pkg>	No need to load	import <pkg>	library(pkg)
Array indexing	A[i, j]	A(i, j)	A[i, j]	A[i, j]
Assignment	b=5	b=5	b=5	B<-5

Sources: julialang.org, mathworks.com, python.org, r-project.org,
https://www.openriskmanual.org/wiki/Overview_of_the_Julia-Python-R_Universe, <https://hyperpolyglot.org/numerical-analysis>, juliapackages.com

Julia vs MATLAB/Python/R

	Julia	Matlab	Python	R
Notebooks	Jupyter	Live Editor	Jupyter	Jupyter, R Markdown
Visualization	Plots.jl	Built-in	Matplotlib, seaborn, bokeh, plotly	ggplot
Package install	Built-in pkg manager	Installer	pip, conda, ...	Built-in install.packages , ...
Package load	using <pkg>, import <pkg>	No need to load	import <pkg>	library(pkg)
Array indexing	A[i, j]	A(i, j)	A[i, j]	A[i, j]
Assignment	b=5	b=5	b=5	B<-5

Sources: julialang.org, mathworks.com, python.org, r-project.org,
https://www.openriskmanual.org/wiki/Overview_of_the_Julia-Python-R_Universe, <https://hyperpolyglot.org/numerical-analysis>, juliapackages.com

Julia vs MATLAB/Python/R

	Julia	Matlab	Python	R
Notebooks	Jupyter	Live Editor	Jupyter	Jupyter, R Markdown
Visualization	Plots.jl	Built-in	Matplotlib, seaborn. bokeh, plotly	ggplot
Package install	Built-in pkg manager	Installer	pip, conda, ...	Built-in install.packages , ...
Package load	using <pkg>, import <pkg>	No need to load	import <pkg>	library(pkg)
Array indexing	A[i, j]	A(i, j)	A[i, j]	A[i, j]
Assignment	b=5	b=5	b=5	B<-5

Sources: julialang.org, mathworks.com, python.org, r-project.org,
https://www.openriskmanual.org/wiki/Overview_of_the_Julia-Python-R_Universe, <https://hyperpolyglot.org/numerical-analysis>, juliapackages.com

Julia vs MATLAB/Python/R

	Julia	Matlab	Python	R
Notebooks	Jupyter	Live Editor	Jupyter	Jupyter, R Markdown
Visualization	Plots.jl	Built-in	Matplotlib, seaborn, bokeh, plotly	ggplot
Package install	Built-in pkg manager	Installer	pip, conda, ...	Built-in install.packages
				, ...
Package load	using <pkg>, import <pkg>	No need to load	import <pkg>	library(pkg)
Array indexing	A[i, j]	A(i, j)	A[i, j]	A[i, j]
Assignment	b=5	b=5	b=5	B<-5

Sources: julialang.org, mathworks.com, python.org, r-project.org,
https://www.openriskmanual.org/wiki/Overview_of_the_Julia-Python-R_Universe, <https://hyperpolyglot.org/numerical-analysis>, juliapackages.com

Julia vs MATLAB/Python/R

	Julia	Matlab	Python	R
Notebooks	Jupyter	Live Editor	Jupyter	Jupyter, R Markdown
Visualization	Plots.jl	Built-in	Matplotlib, seaborn, bokeh, plotly	ggplot
Package install	Built-in pkg manager	Installer	pip, conda, ...	Built-in install.packages , ...
Package load	using <pkg>, import <pkg>	No need to load	import <pkg>	library(pkg)
Array indexing	A[i, j]	A(i, j)	A[i, j]	A[i, j]
Assignment	b=5	b=5	b=5	B<-5

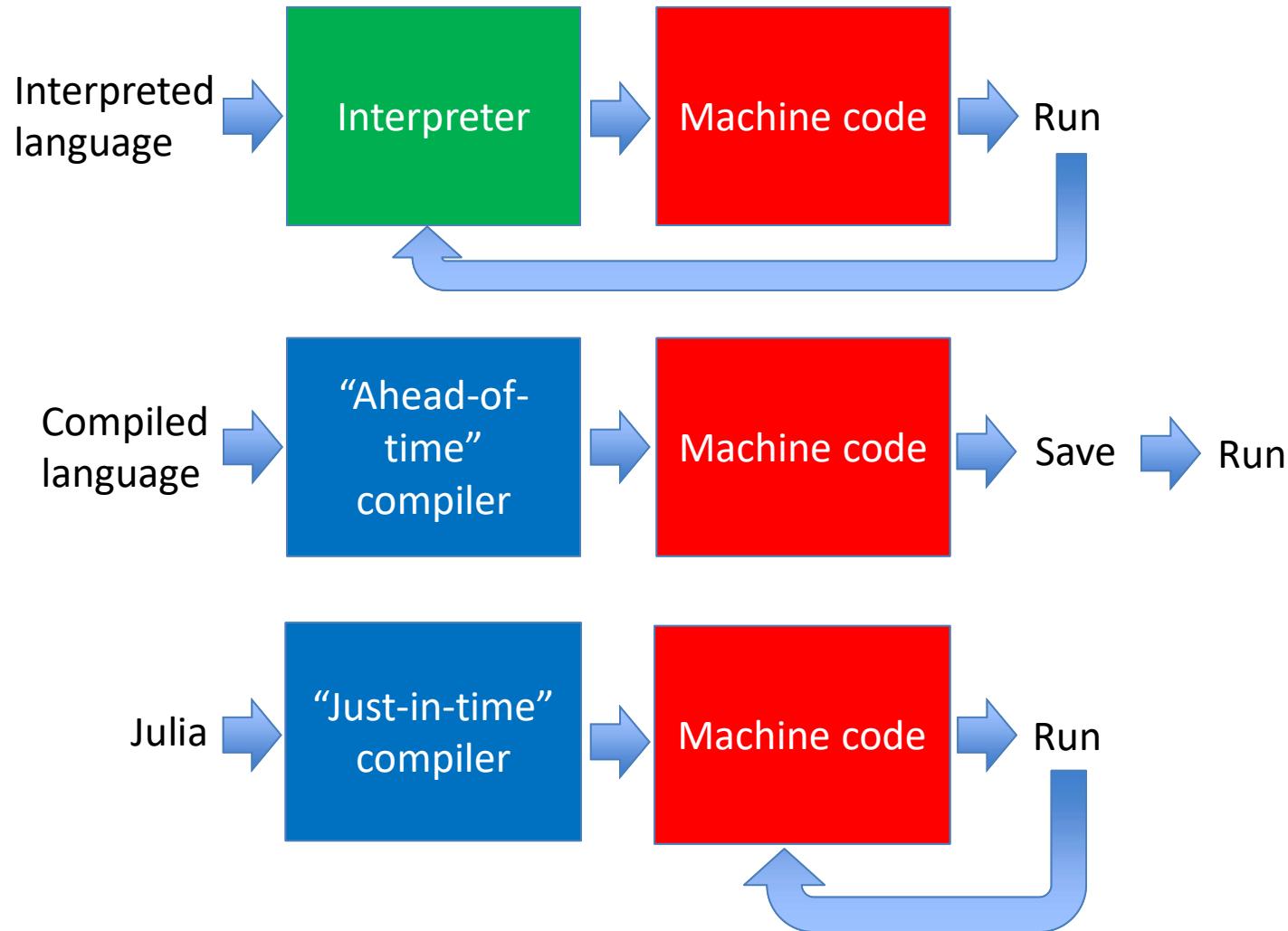
Sources: julialang.org, mathworks.com, python.org, r-project.org,
https://www.openriskmanual.org/wiki/Overview_of_the_Julia-Python-R_Universe, <https://hyperpolyglot.org/numerical-analysis>, juliapackages.com

Julia vs MATLAB/Python/R

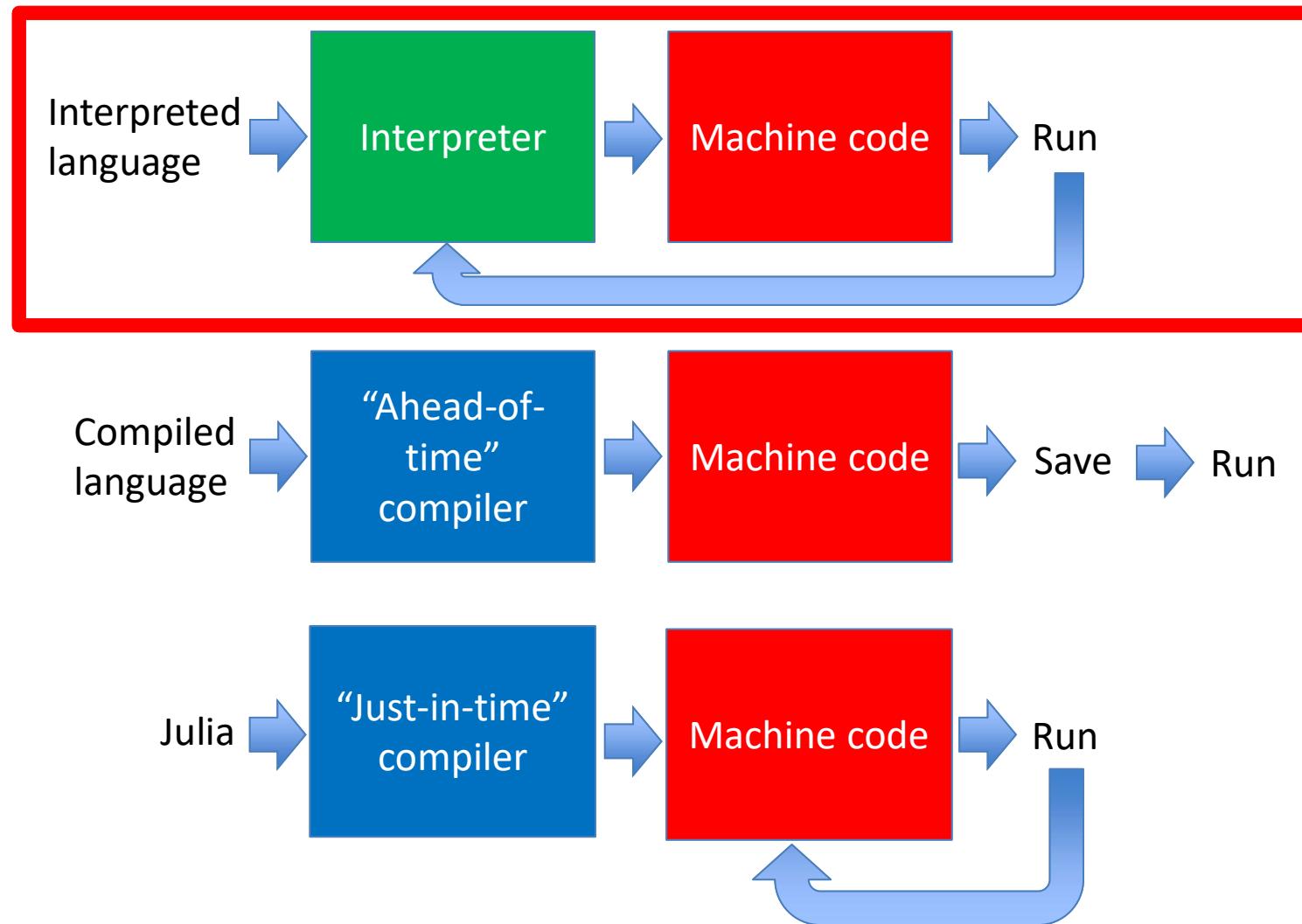
	Julia	Matlab	Python	R
Conditional evaluation	if <condition> action1() else action2() end	if <condition> action1() else action2() end	if <condition>: action1() else: action2()	if <condition> { action1() } else { action2() }
Repeated evaluation	while <cond> action() end	while <cond> action() end	while <cond>: action()	while <cond> { action() }
Repeated evaluation	for <count> action() end	For <count> action() end	for <count> action()	for <count>: action() }

Sources: julialang.org, mathworks.com, python.org, r-project.org,
https://www.openriskmanual.org/wiki/Overview_of_the_Julia-Python-R_Universe, <https://hyperpolyglot.org/numerical-analysis>, juliapackages.com

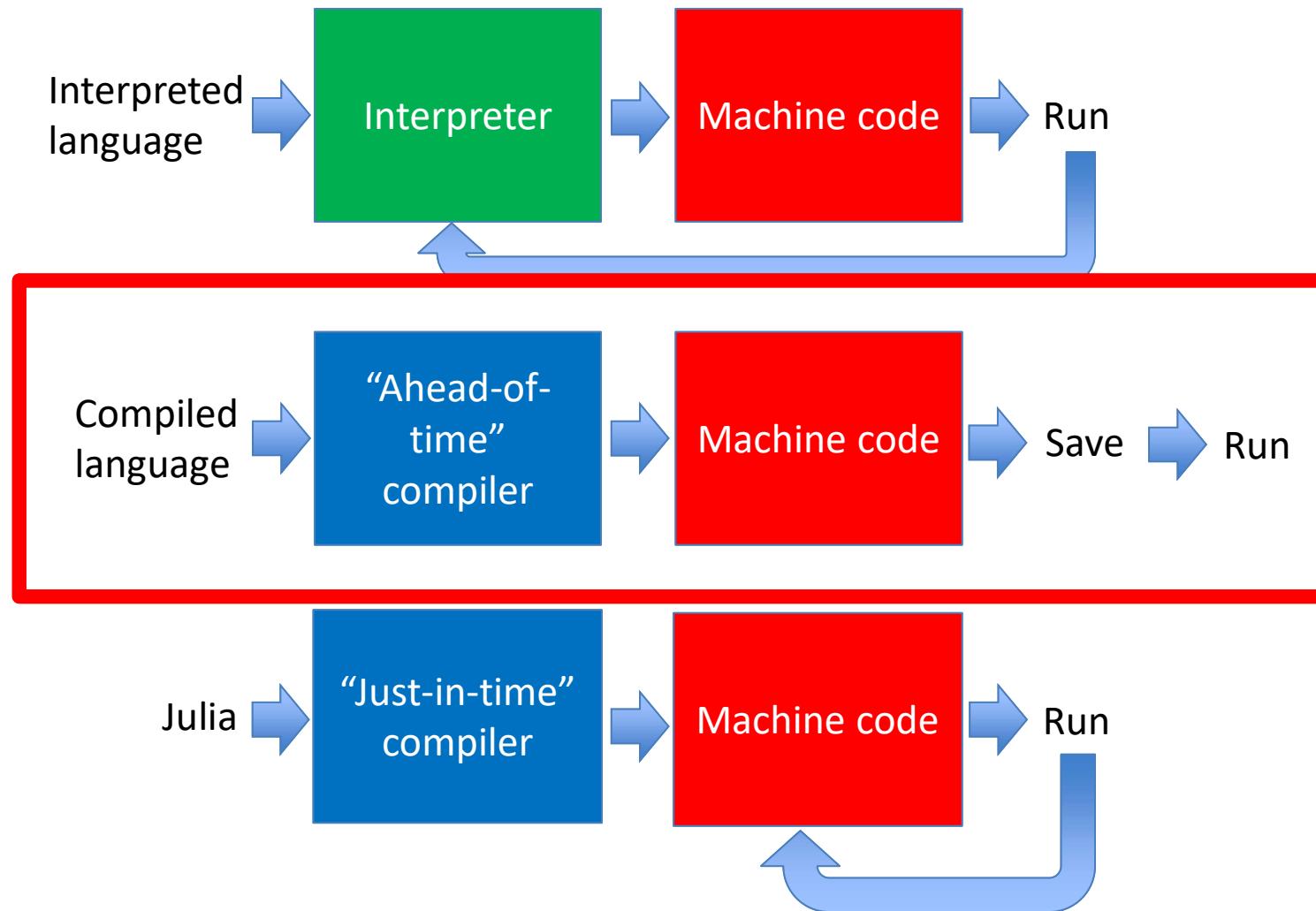
Julia vs others



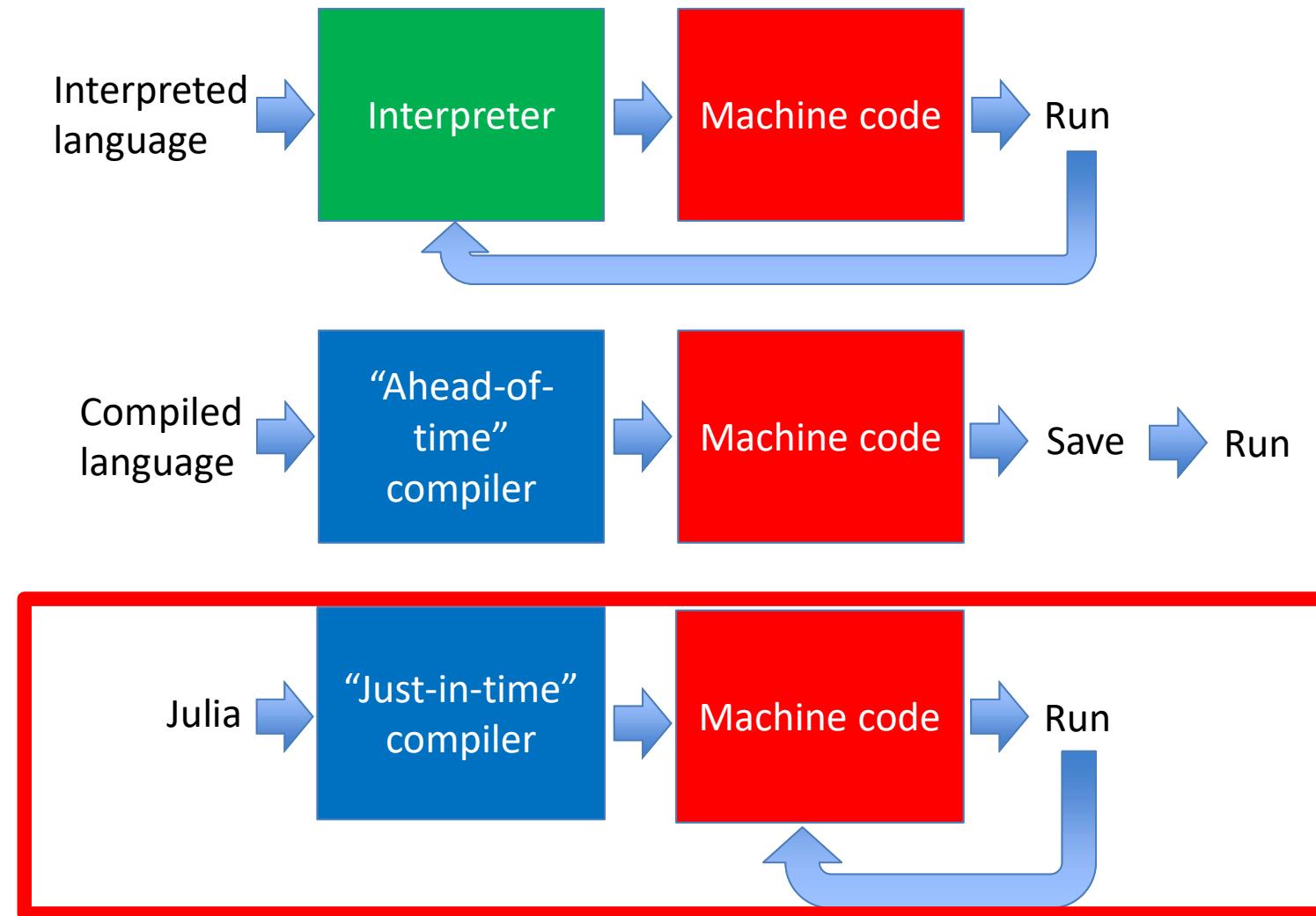
Julia vs others



Julia vs others



Julia vs others



Scientific Julia

Packages:

Scientific Julia

Packages:

- Differential equations: DifferentialEquations.jl

Scientific Julia

Packages:

- Differential equations: DifferentialEquations.jl
- Optimization: JuMP.jl

Scientific Julia

Packages:

- Differential equations: DifferentialEquations.jl
- Optimization: JuMP.jl
- Visualization: Plots.jl, PyPlot.jl, Gadfly.jl

Scientific Julia

Packages:

- Differential equations: DifferentialEquations.jl
- Optimization: JuMP.jl
- Visualization: Plots.jl, PyPlot.jl, Gadfly.jl
- Machine Learning / Deep Learning: Flux.jl, Knet.jl, Mocha.jl

Scientific Julia

Packages:

- Differential equations: DifferentialEquations.jl
- Optimization: JuMP.jl
- Visualization: Plots.jl, PyPlot.jl, Gadfly.jl
- Machine Learning / Deep Learning: Flux.jl, Knet.jl, Mocha.jl
- Jupyter: IJulia.jl

Scientific Julia

Packages:

- Differential equations: DifferentialEquations.jl
- Optimization: JuMP.jl
- Visualization: Plots.jl, PyPlot.jl, Gadfly.jl
- Machine Learning / Deep Learning: Flux.jl, Knet.jl, Mocha.jl
- Jupyter: IJulia.jl
- Tabular data: DataFrames.jl

Scientific Julia

Pros of using Julia:

Scientific Julia

Pros of using Julia:

- Easy syntax

Scientific Julia

Pros of using Julia:

- Easy syntax
- Compiled (faster than interpreted)

Scientific Julia

Pros of using Julia:

- Easy syntax
- Compiled (faster than interpreted)
- Dynamically typed

Scientific Julia

Pros of using Julia:

- Easy syntax
- Compiled (faster than interpreted)
- Dynamically typed
- Interoperable with other languages

Scientific Julia

Cons of using Julia:

Scientific Julia

Cons of using Julia:

- Number of mature packages is small

Scientific Julia

Cons of using Julia:

- Number of mature packages is small
- Relatively small scientific community: Most colleagues are unlikely to use Julia

Scientific Julia

Cons of using Julia:

- Number of mature packages is small
- Relatively small scientific community: Most colleagues are unlikely to use Julia
- Relatively limited support

Julia IDEs

Julia IDEs

- VSCode * (julia-vscode.org/)

Julia IDEs

- VSCode * (julia-vscode.org/)
- Jupyter (jupyter.org)

Julia IDEs

- VSCode * (julia-vscode.org/)
- Jupyter (jupyter.org)
- Juno (junolab.org)

Julia IDEs

- VSCode * (julia-vscode.org/)
- Jupyter (jupyter.org)
- Juno (junolab.org)

*For details on how to VSCode remote into Biowulf's compute nodes see <https://hpc.nih.gov/apps/vscode.html>

Example: ML using Flux

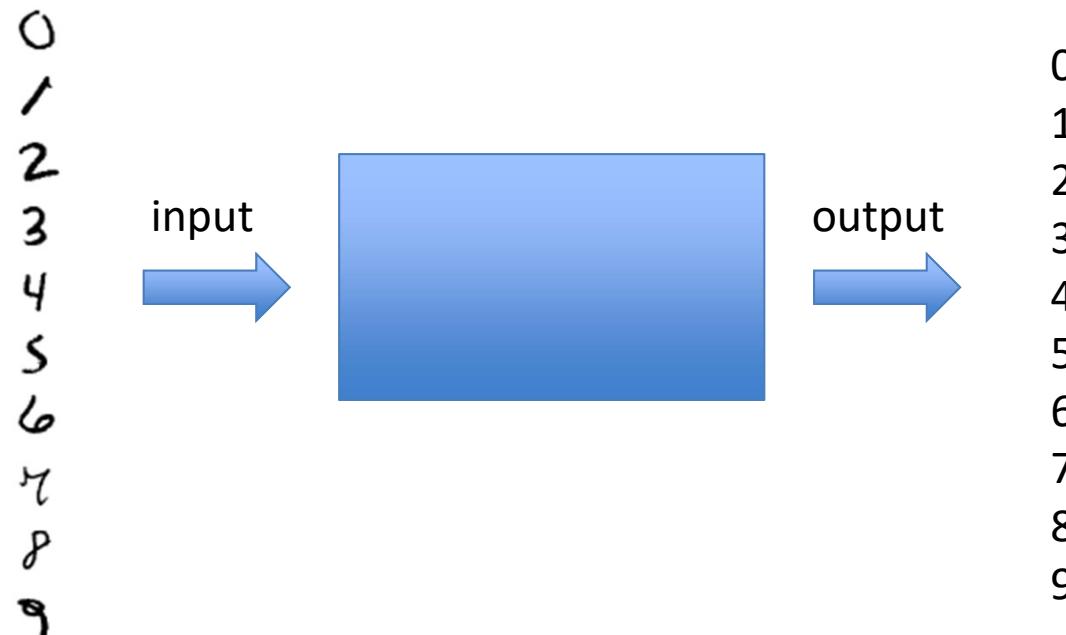
Example: ML using Flux

- Problem: recognition of hand-written digits.



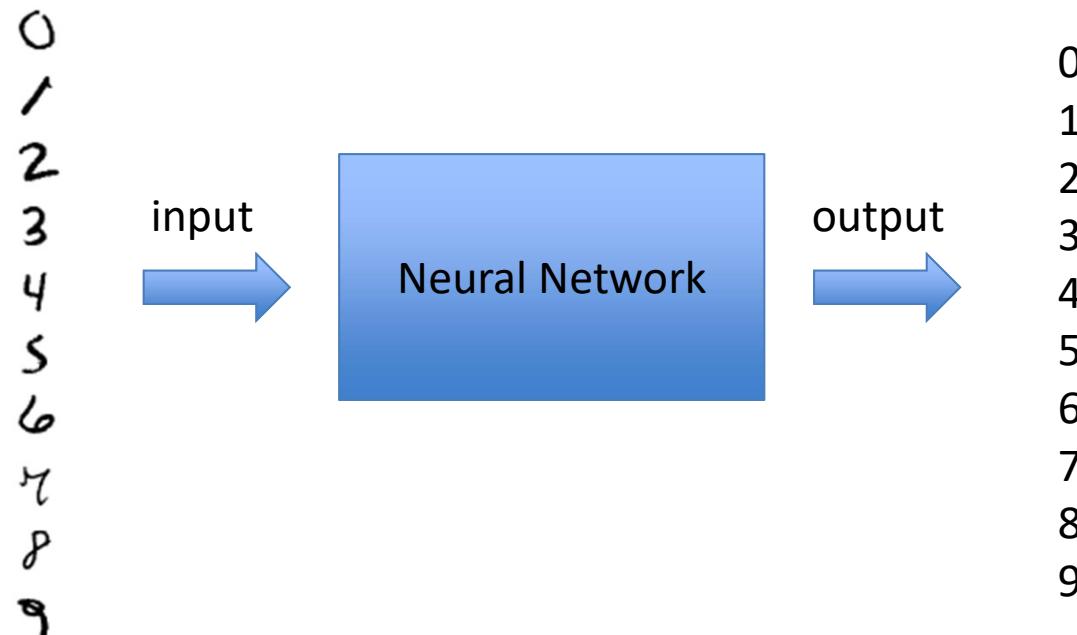
Example: ML using Flux

- Problem: recognition of hand-written digits.



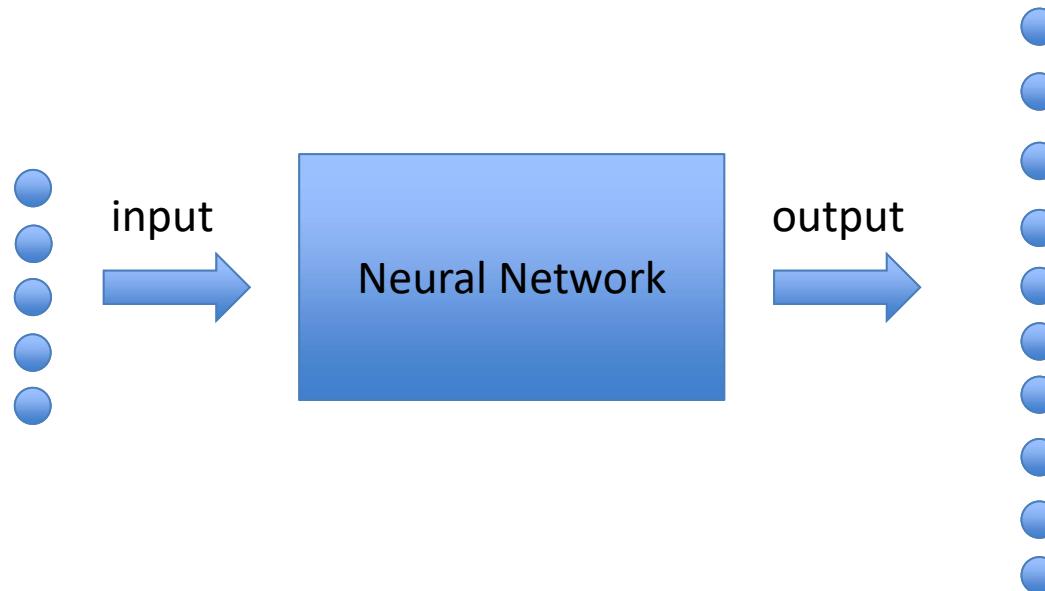
Example: ML using Flux

- Problem: recognition of hand-written digits.



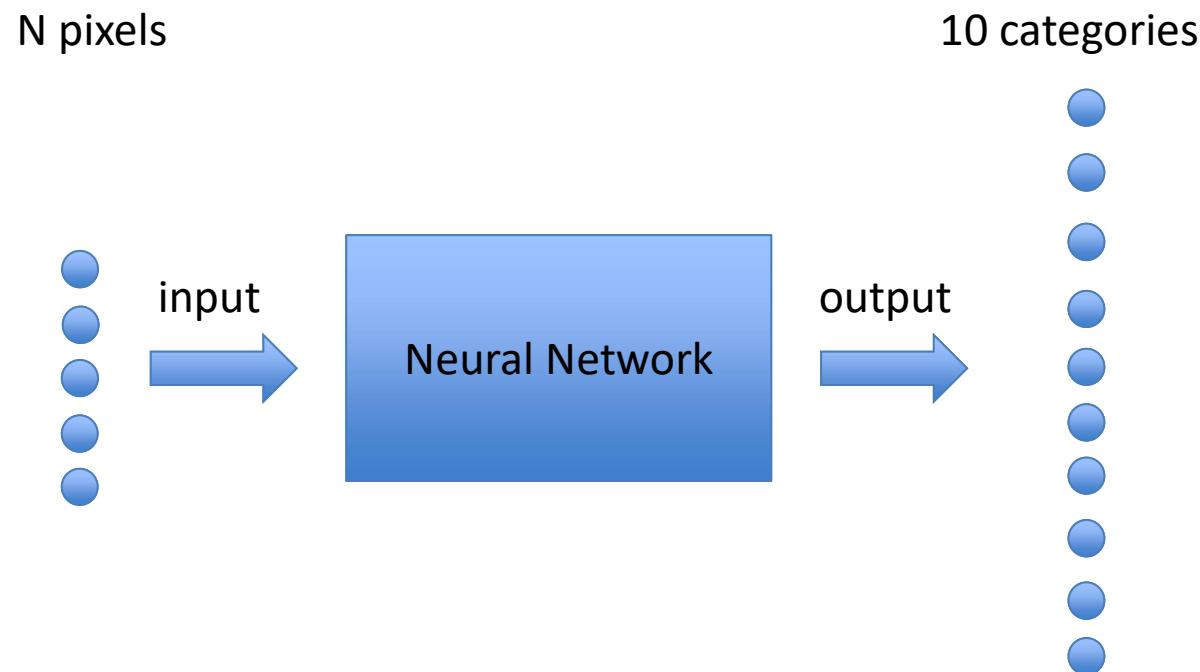
Example: ML using Flux

- Problem: recognition of hand-written digits.



Example: ML using Flux

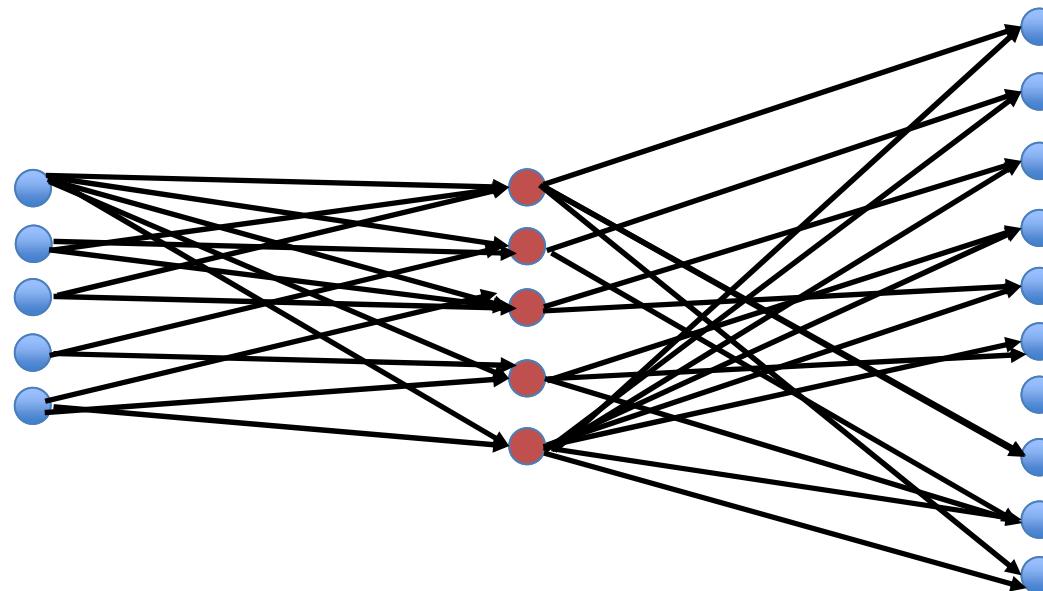
- Problem: recognition of hand-written digits.



Example: ML using Flux

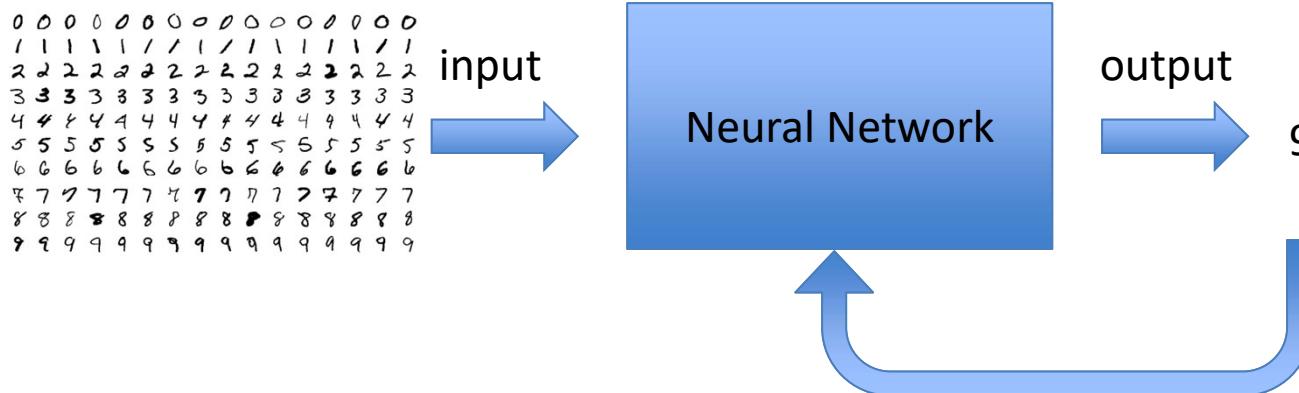
- Problem: recognition of hand-written digits.

N pixels 10 categories



Example: ML using Flux

- Goal: run a Julia script to train a classifier using a set of 60,000 images of handwritten digits (MNIST).



Example: ML using Flux

- Goal: run a Julia script to train a classifier using a set of 60,000 images of handwritten digits (MNIST).



Example: ML using Flux

- 1) Connect to biowulf
- 2) Start interactive session
- 3) Load the julia module
- 4) Move into local disk
- 4) Run Julia
- 5) Start package manager
- 6) Install packages
- 7) Check status of packages
- 8) Exit Julia
- 9) Clone Flux ML example
- 10) Move into example dir
- 11) Inspect script w nano
- 12) Exit nano
- 12) Run ML example

```
ssh username@biowulf.nih.gov
sinteractive --mem=10g -c14 --gres=lscratch:2,gpu:p100:1
module load julia
cd /lscratch/${SLURM_JOB_ID}
julia
]
add CUDA Flux MLDatasets Statistics
status
<Ctrl-D>
git clone https://github.com/FluxML/model-zoo.git
cd model-zoo/vision/mlp_mnist/
nano mlp_mnist.jl
<Ctrl-X>
julia mlp_mnist.jl
```

Example: ML using Flux

1) Connect to biowulf

```
ssh username@biowulf.nih.gov
```

2) Start interactive session

```
sinteractive --mem=10g -c14 --gres=lscratch:2,gpu:p100:1
```

3) Load the julia module

```
module load julia
```

4) Move into local disk

```
cd /lscratch/${SLURM_JOB_ID}
```

4) Run Julia

```
julia
```

```
]
```

5) Start package manager

```
add CUDA Flux MLDatasets Statistics
```

6) Install packages

```
status
```

7) Check status of packages

```
<Ctrl-D>
```

8) Exit Julia

```
git clone https://github.com/FluxML/model-zoo.git
```

9) Clone Flux ML example

```
cd model-zoo/vision/mlp_mnist/
```

10) Move into example dir

```
nano mlp_mnist.jl
```

11) Inspect script w nano

```
<Ctrl-X>
```

12) Exit nano

```
julia mlp_mnist.jl
```

Example: ML using Flux

- 1) Connect to biowulf
- 2) Start interactive session
- 3) Load the julia module
- 4) Move into local disk
- 5) Run Julia
- 6) Start package manager
- 7) Install packages
- 8) Check status of packages
- 9) Exit Julia
- 10) Clone Flux ML example
- 11) Move into example dir
- 12) Inspect script w nano
- 13) Exit nano
- 14) Run ML example

```
ssh username@biowulf.nih.gov
sinteractive --mem=10g -c14 --gres=lscratch:2,gpu:p100:1
module load julia
cd /lscratch/${SLURM_JOB_ID}
julia
]
add CUDA Flux MLDatasets Statistics
status
<Ctrl-D>
git clone https://github.com/FluxML/model-zoo.git
cd model-zoo/vision/mlp_mnist/
nano mlp_mnist.jl
<Ctrl-X>
julia mlp_mnist.jl
```

Example: ML using Flux

- 1) Connect to biowulf
- 2) Start interactive session
- 3) Load the julia module
- 4) Move into local disk
- 5) Run Julia
- 6) Start package manager
- 7) Install packages
- 8) Check status of packages
- 9) Exit Julia
- 10) Clone Flux ML example
- 11) Move into example dir
- 12) Inspect script w nano
- 13) Exit nano
- 14) Run ML example

```
ssh username@biowulf.nih.gov
sinteractive -mem=10g -c14 -gres=lsscratch:2,gpu:p100:1
module load julia
cd /lscratch/$SLURM_JOB_ID
julia
]
add CUDA Flux MLDatasets Statistics
status
<Ctrl-D>
git clone https://github.com/FluxML/model-zoo.git
cd model-zoo/vision/mlp_mnist/
nano mlp_mnist.jl
<Ctrl-X>
julia mlp_mnist.jl
```

Example: ML using Flux

1) Connect to biowulf	ssh username@biowulf.nih.gov
2) Start interactive session	sinteractive --mem=10g -c14 --gres=lscratch:2,gpu:p100:1
3) Load the julia module	module load julia
4) Move into local disk	cd /lscratch/\${SLURM_JOB_ID}
4) Run Julia	julia
5) Start package manager]
6) Install packages	add CUDA Flux MLDatasets Statistics
7) Check status of packages	status
8) Exit Julia	<Ctrl-D>
9) Clone Flux ML example	git clone https://github.com/FluxML/model-zoo.git
10) Move into example dir	cd model-zoo/vision/mlp_mnist/
11) Inspect script w nano	nano mlp_mnist.jl
12) Exit nano	<Ctrl-X>
12) Run ML example	julia mlp_mnist.jl

Example: ML using Flux

- 1) Connect to biowulf
- 2) Start interactive session
- 3) Load the julia module
- 4) Move into local disk
- 4) Run Julia
- 5) Start package manager
- 6) Install packages
- 7) Check status of packages
- 8) Exit Julia
- 9) Clone Flux ML example
- 10) Move into example dir
- 11) Inspect script w nano
- 12) Exit nano
- 12) Run ML example

```
ssh username@biowulf.nih.gov
sinteractive --mem=10g -c14 --gres=lscratch:2,gpu:p100:1
module load julia
cd /lscratch/$SLURM_JOB_ID
julia
add CUDA Flux MLDatasets Statistics
status
<Ctrl-D>
git clone https://github.com/FluxML/model-zoo.git
cd model-zoo/vision/mlp_mnist/
nano mlp_mnist.jl
<Ctrl-X>
julia mlp_mnist.jl
```

Example: ML using Flux

- 1) Connect to biowulf
- 2) Start interactive session
- 3) Load the julia module
- 4) Move into local disk
- 4) Run Julia
- 5) Start package manager
- 6) Install packages
- 7) Check status of packages
- 8) Exit Julia
- 9) Clone Flux ML example
- 10) Move into example dir
- 11) Inspect script w nano
- 12) Exit nano
- 12) Run ML example

```
ssh username@biowulf.nih.gov
sinteractive --mem=10g -c14 --gres=lscratch:2,gpu:p100:1
module load julia
cd /lscratch/${SLURM_JOB_ID}
julia
]
add CUDA Flux MLDatasets Statistics
status
<Ctrl-D>
git clone https://github.com/FluxML/model-zoo.git
cd model-zoo/vision/mlp_mnist/
nano mlp_mnist.jl
<Ctrl-X>
julia mlp_mnist.jl
```

Example: ML using Flux

- 1) Connect to biowulf
- 2) Start interactive session
- 3) Load the julia module
- 4) Move into local disk
- 5) Start package manager
- 6) Install packages
- 7) Check status of packages
- 8) Exit Julia
- 9) Clone Flux ML example
- 10) Move into example dir
- 11) Inspect script w nano
- 12) Exit nano
- 12) Run ML example

```
ssh username@biowulf.nih.gov
sinteractive --mem=10g -c14 --gres=lscratch:2,gpu:p100:1
module load julia
cd /lscratch/${SLURM_JOB_ID}
julia
1
add CUDA Flux MLDatasets Statistics
status
<Ctrl-D>
git clone https://github.com/FluxML/model-zoo.git
cd model-zoo/vision/mlp_mnist/
nano mlp_mnist.jl
<Ctrl-X>
julia mlp_mnist.jl
```

Example: ML using Flux

- 1) Connect to biowulf
- 2) Start interactive session
- 3) Load the julia module
- 4) Move into local disk
- 5) Run Julia
- 6) Install packages
- 7) Check status of packages
- 8) Exit Julia
- 9) Clone Flux ML example
- 10) Move into example dir
- 11) Inspect script w nano
- 12) Exit nano
- 12) Run ML example

```
ssh username@biowulf.nih.gov
sinteractive --mem=10g -c14 --gres=lscratch:2,gpu:p100:1
module load julia
cd /lscratch/${SLURM_JOB_ID}
julia
]
add CUDA Flux ML Datasets Statistics
status
<Ctrl-D>
git clone https://github.com/FluxML/model-zoo.git
cd model-zoo/vision/mlp_mnist/
nano mlp_mnist.jl
<Ctrl-X>
julia mlp_mnist.jl
```

Example: ML using Flux

- 1) Connect to biowulf
- 2) Start interactive session
- 3) Load the julia module
- 4) Move into local disk
- 5) Run Julia
- 6) Start package manager
- 7) Install packages
- 8) Check status of packages
- 9) Exit Julia
- 10) Clone Flux ML example
- 11) Move into example dir
- 12) Inspect script w nano
- 13) Exit nano
- 14) Run ML example

```
ssh username@biowulf.nih.gov
sinteractive --mem=10g -c14 --gres=lscratch:2,gpu:p100:1
module load julia
cd /lscratch/${SLURM_JOB_ID}
julia
]
add CUDA Flux MLDatasets Statistics
status
```

<Ctrl-D>

```
git clone https://github.com/fluxML/model-zoo.git
cd model-zoo/vision/mlp_mnist/
nano mlp_mnist.jl
<Ctrl-X>
julia mlp_mnist.jl
```

Example: ML using Flux

- 1) Connect to biowulf
- 2) Start interactive session
- 3) Load the julia module
- 4) Move into local disk
- 4) Run Julia
- 5) Start package manager
- 6) Install packages
- 7) Check status of packages
- 8) Exit Julia
- 9) Clone Flux ML example
- 10) Move into example dir
- 11) Inspect script w nano
- 12) Exit nano
- 12) Run ML example

```
ssh username@biowulf.nih.gov
sinteractive --mem=10g -c14 --gres=lscratch:2,gpu:p100:1
module load julia
cd /lscratch/${SLURM_JOB_ID}
julia
]
add CUDA Flux MLDatasets Statistics
status
<Ctrl-D>
git clone https://github.com/FluxML/model-zoo.git
cd model-zoo/vision/mlp_mnist/
nano mlp_mnist.jl
<Ctrl-X>
julia mlp_mnist.jl
```

Example: ML using Flux

- 1) Connect to biowulf
- 2) Start interactive session
- 3) Load the julia module
- 4) Move into local disk
- 4) Run Julia
- 5) Start package manager
- 6) Install packages
- 7) Check status of packages
- 8) Exit Julia
- 9) Clone Flux ML example
- 10) Move into example dir
- 11) inspect script w nano
- 12) Exit nano
- 12) Run ML example

```
ssh username@biowulf.nih.gov
sinteractive --mem=10g -c14 --gres=lscratch:2,gpu:p100:1
module load julia
cd /lscratch/${SLURM_JOB_ID}
julia
]
add CUDA Flux MLDatasets Statistics
status
<Ctrl-D>
git clone https://github.com/FluxML/model-zoo.git
cd model-zoo/vision/mlp_mnist/
nano mlp_mnist.jl
<Ctrl-X>
julia mlp_mnist.jl
```

Example: ML using Flux

- 1) Connect to biowulf
- 2) Start interactive session
- 3) Load the julia module
- 4) Move into local disk
- 4) Run Julia
- 5) Start package manager
- 6) Install packages
- 7) Check status of packages
- 8) Exit Julia
- 9) Clone Flux ML example
- 10) Move into example dir
- 11) Inspect script w nano
- 12) EXIT nano
- 12) Run ML example

```
ssh username@biowulf.nih.gov
sinteractive --mem=10g -c14 --gres=lscratch:2,gpu:p100:1
module load julia
cd /lscratch/${SLURM_JOB_ID}
julia
]
add CUDA Flux MLDatasets Statistics
status
<Ctrl-D>
git clone https://github.com/FluxML/model-zoo.git
cd model-zoo/vision/mlp_mnist/
nano mlp_mnist.jl
<Ctrl-X>
julia mlp_mnist.jl
```

Example: ML using Flux

- 1) Connect to biowulf
- 2) Start interactive session
- 3) Load the julia module
- 4) Move into local disk
- 4) Run Julia
- 5) Start package manager
- 6) Install packages
- 7) Check status of packages
- 8) Exit Julia
- 9) Clone Flux ML example
- 10) Move into example dir
- 11) Inspect script w nano
- 12) Exit nano
- 12) Run ML example

```
ssh username@biowulf.nih.gov
sinteractive --mem=10g -c14 --gres=lscratch:2,gpu:p100:1
module load julia
cd /lscratch/${SLURM_JOB_ID}
julia
]
add CUDA Flux MLDatasets Statistics
status
<Ctrl-D>
git clone https://github.com/FluxML/model-zoo.git
cd model-zoo/vision/mlp_mnist/
nano mlp_mnist.jl
<Ctrl-X>
julia mlp_mnist.jl
```

Example: ML using Flux

- 1) Connect to biowulf
- 2) Start interactive session
- 3) Load the julia module
- 4) Move into local disk
- 4) Run Julia
- 5) Start package manager
- 6) Install packages
- 7) Check status of packages
- 8) Exit Julia
- 9) Clone Flux ML example
- 10) Move into example dir
- 11) Inspect script w nano
- 12) Exit nano
- 12) Run ML example

```
ssh username@biowulf.nih.gov
sinteractive --mem=10g -c14 --gres=lscratch:2,gpu:p100:1
module load julia
cd /lscratch/${SLURM_JOB_ID}
julia
]
add CUDA Flux MLDatasets Statistics
status
<Ctrl-D>
git clone https://github.com/FluxML/model-zoo.git
cd model-zoo/vision/mlp_mnist/
nano mlp_mnist.jl
<Ctrl-X>
julia mlp_mnist.jl
```

Conclusion

Hopefully you:

Conclusion

Hopefully you:

1. have a general idea of the pros and cons of using Julia, and

Conclusion

Hopefully you:

1. have a general idea of the pros and cons of using Julia, and
2. know how to run Julia scripts interactively on Biowulf.

Questions? Comments?

staff@hpc.nih.gov