Technology Review:
MACHINE LEARNING TOOLKITS

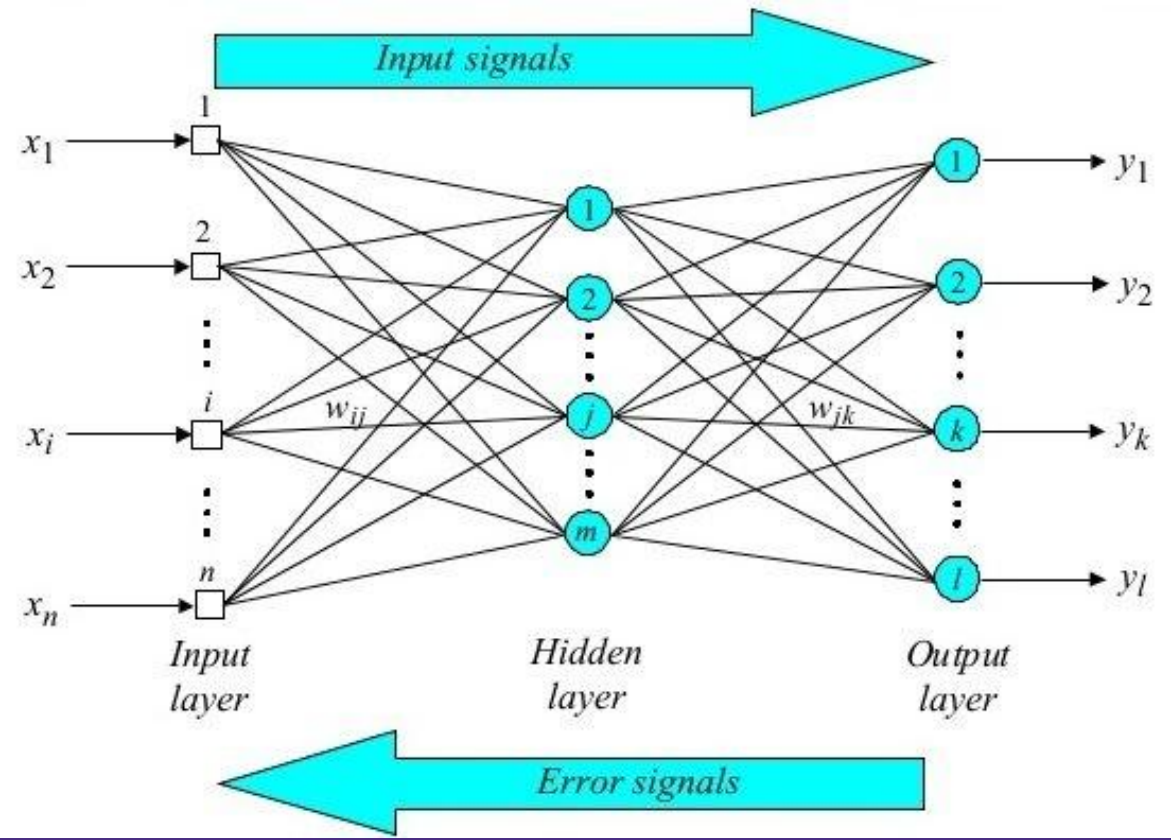
Ryan Carlin, Andrew Copsey, Corey Johnson and Michelle Katz

UNIVERSITY of WASHINGTON

PROJECT BACKGROUND

- Project goal: use crystallography to predict electronic properties (Density of State)
- Data form Materials Project Database
- Will use neural network to model

NEURAL NETWORKS



TOOL: Scikit Learn

How it works:

> **A python-specific machine learning tool built using other python packages: numpy, scipy + matplotlib**

Appeal:

> **Simple to use, good for beginners(us)**
> **Actively being developed, renowned**
> **Lots of great documentation/tutorials**
> **Has neural network tools**
> **Research and startup friendly BSD License**

Drawbacks:

> **Might be too simple for our purposes(doesn't support deep learning as well as other toolkits such as TensorFlow)**

TOOL: PyBrain

How it works:

- Python machine learning library using SciPy toolkit, utilizing “network” and “connection” objects with specific training algorithms

Appeal:

- Well documented with tutorials
- User friendly and easy to learn quickly
- Neural networking platform with functions for (un)supervised learning, black box optimization, etc.

Drawbacks

- Dated with few updates
  - no commits since December 2017
  - GitHub issue ratio of 120 open/43 closed
- Doesn’t appear to be managed or updated

TOOL: KERAS

How it works:

- A high-level neural networks API, that is written in Python and runs on top of Tensorflow, CNTK, and/or Theano

Appeal:

- Simplicity and suitability
- Modularity and extensibility
- Documentation
- Active development and popularity
  - 5,039 commits
  - 6,639 issues closed with 2,215 open
  - 46 releases w/ most recent in Oct 2018

Drawbacks:

- Made specifically for deep learning, not the best for standard machine learning operations

CONCLUSION

- Scikit Learn vs Keras
- Next Steps
  - Wrangling data to be appropriate for model