This work

- **Formal language** to encode search spaces over architectures
  - Decouples implementation of search space and search algorithm
  - Search algorithms interact with search spaces through a well-defined interface
- **Modular and programmable framework** for architecture search over general domains
  - Use architecture search for your use cases
  - Implement your search spaces and search algorithms
  - Mix and match implementations of search spaces and search algorithms without writing combinations from scratch
  - Code and documentation at github.com/negrinho/deep_architect.
  - Contribute your search spaces, search algorithms, benchmarks, and more.
  - Easy wrapping of Pytorch, Tensorflow, and Keras layers.

**Highlights**

- **Deep Architect**

Motivation

- Hyperparameter optimization not focused on architecture search
- No existing general architecture search systems
  - Ad-hoc encodings for search spaces
  - Intertwined search space and search algorithms
  - Task-specific, e.g., image classification.
- Programmable frameworks (e.g., deep learning) had transformative impact on machine learning

Search space example

**Description**

- Input convolutional layer
- Optional dropout w/ rate 0.25 or 0.5
- Two parallel convolutional chains:
  - One chain length 1, 2, or 4
  - Other chain double the first
- Chain outputs concatenated
- Each convolution has 64 or 128 filters (chosen separately)

25008 possible architectures

**Transitions**

a) Search space encoded by code

```python
def search_space():
    h_n = D[1, 2, 4]
    d_n, _ = D[optional_hyperparameter(lambda: lambda: lambda)]
    c_inputs, c_outputs = conv2D((64, 128))
    r_inputs, r_outputs = sin_repeat((64, dh), (64, dh))
    c_inputs, c_outputs = sin_repeat((64, dh), (64, dh))
    c_inputs, c_outputs = conv2D((64, dh))
    c_outputs = c_outputs[0]
```

b) Value assigned to IH-1

- Triggers value assignment to DH-1
- Triggers substitution for Repeats (1 and 2)

c) Value assigned to IH-2

- Triggers substitution for Optional-1

d) Assignments to IH-1 (64), IH-2 (128), IH-4 (128), IH-5 (128), and IH-7 (0.5)
- Can be mapped to implementation

**Language**

**Constructs**

- Independent hyperparameters
  - Value picked from set (e.g., IH-1)
- Dependent hyperparameters
  - Value computed as function of other hypo (e.g., DH-2)
- Basic modules
  - Deep learning operation (Conv2D-1)
- Substitution modules
  - Lazy transformations (e.g., Repeat-1) to computation graph through substitutions (replace and reroute)
- Auxiliary functions
  - Helps compose search spaces into larger search spaces (e.g., rnn_cell)

**Mechanics**

- Search algorithms interface with search spaces by assigning values to independent hyperparameters
- After all hyperparameters have values assigned, architecture mapped automatically to implementation

**Experiments**

Mix and match search spaces and search algorithms without implementing each combination from scratch

One search algorithm (random), many search spaces

One search space (genetic), many search algorithms