CSCI 252: Neural Networks
Prof. Levy

Architecture #7: Biologically Realistic Networks Part II: The Semantic Pointer Architecture (Eliasmith 2012)

Semantic Pointer

- Semantics: Meaning, as distinguished from form (syntax)
- Classic example: passive vs. active = different form, same meaning:

  The cat chased the rat.

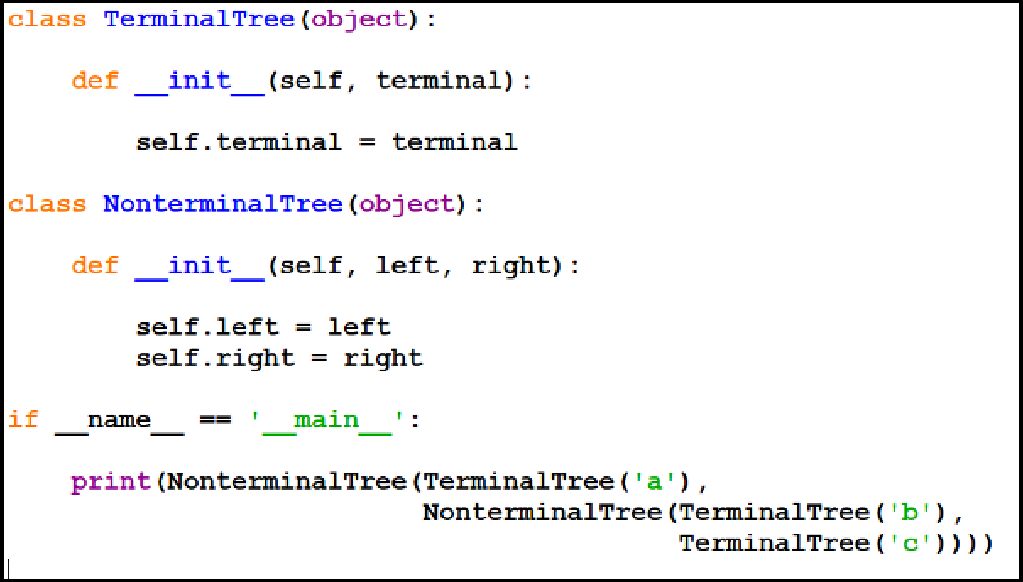
  The rat was chased by the cat.
- As we saw in our PDP “multiple simultaneous constraints” discussion, the distinction isn’t that clear:

  The cow kicked the bucket.

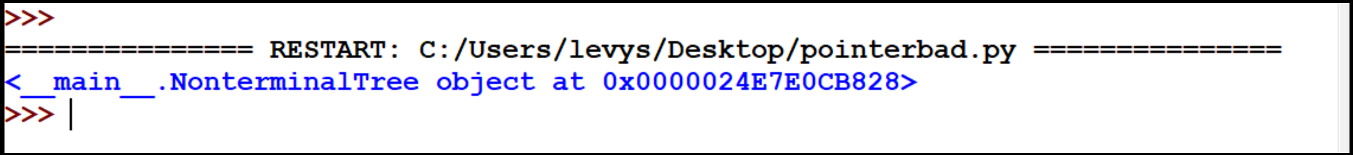
  ** The bucket was kicked by the cow.*

Semantic Pointer

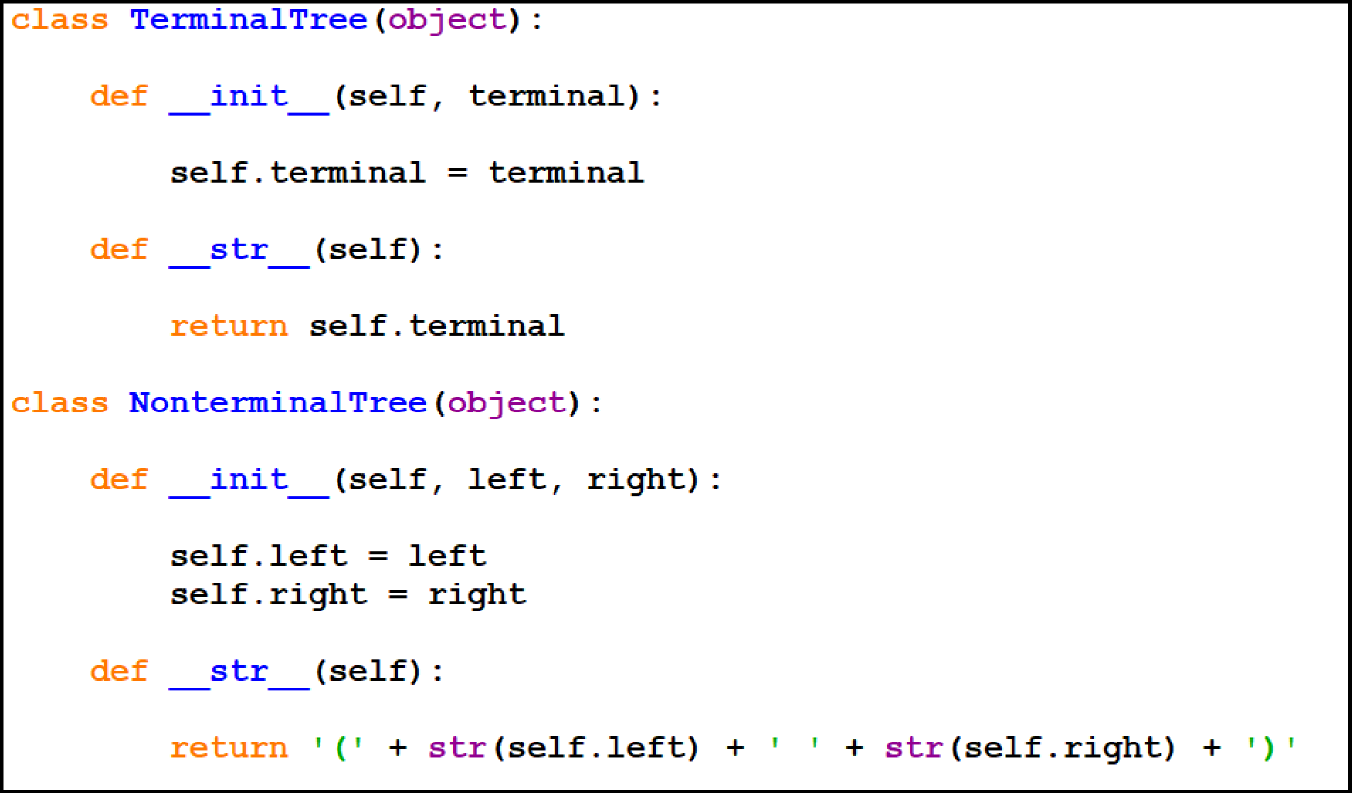
- Pointer: In computer science, a pointer (a.k.a. reference) is the address (memory location) of a data structure (e.g., Python object);
- We'll see this address if we forget to write a `__str__` method for our Python class:



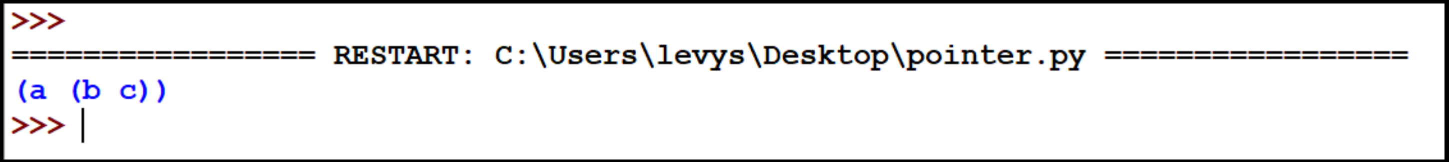
```
class TerminalTree(object):

    def __init__(self, terminal):

        self.terminal = terminal

class NonterminalTree(object):

    def __init__(self, left, right):

        self.left = left
        self.right = right

if __name__ == '__main__':

    print(NonterminalTree(TerminalTree('a'),
                          NonterminalTree(TerminalTree('b'),
                                          TerminalTree('c'))))
```



```
>>>
=============== RESTART: C:/Users/levys/Desktop/pointerbad.py ===============
<__main__.NonterminalTree object at 0x0000024E7E0CB828>
>>> |
```



```
class TerminalTree(object):

    def __init__(self, terminal):

        self.terminal = terminal

    def __str__(self):

        return self.terminal

class NonterminalTree(object):

    def __init__(self, left, right):

        self.left = left
        self.right = right

    def __str__(self):

        return '(' + str(self.left) + ' ' + str(self.right) + ')'
```



```
>>>
================= RESTART: C:\Users\levys\Desktop\pointer.py =================
(a (b c))
>>> |
```

Because a pointer is also called a reference, the old-fashioned term for “unpacking” the contents like this is called dereferencing.

Dereferencing Semantic Pointers: Convolution Net Example

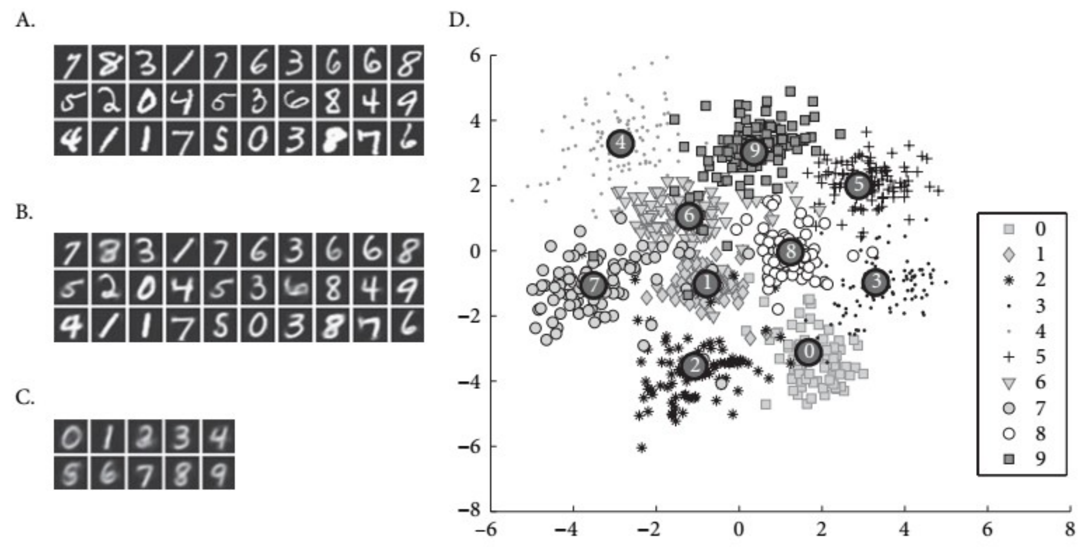


FIGURE 3.7 Dereferencing semantic pointers. **A**. The original input of several randomly chosen images. These are compressed into a 50D state space representation at the highest level of the model. **B**. The resulting images from dereferencing the 50D semantic pointers. The 50D representations generated by the network based on the input are clamped at the highest level, and the rest of the network is used to generate a guess as to what input would produce that semantic pointer. **C**. The dereferenced average semantic pointer for each category. These can be thought of as a prototype of the category. **D**. The compressed 50D representations are plotted in two dimensions using dimensionality reduction. Large dark circles indicate the mean value of the semantic pointers for each category. These can be thought of as the middle of the dimples on the conceptual golf ball (*see* Fig. 3.1).

Eliasmith (2012) p. 96

Semantics: Deep vs. Shallow

- As the Python example shows, a pointer should provide access to the full structure, including hierarchical relationships (trees), role/filler, etc.: this is the "deep semantics"
- "Shallow semantics" is simple word association (like LSA): extremely useful (web search, automatic essay grading, word disambiguation), but not the whole picture (*cat chased rat* vs. *rat chased cat*)
- With VSA, we already know how to build structures, however, we still have a problem ....

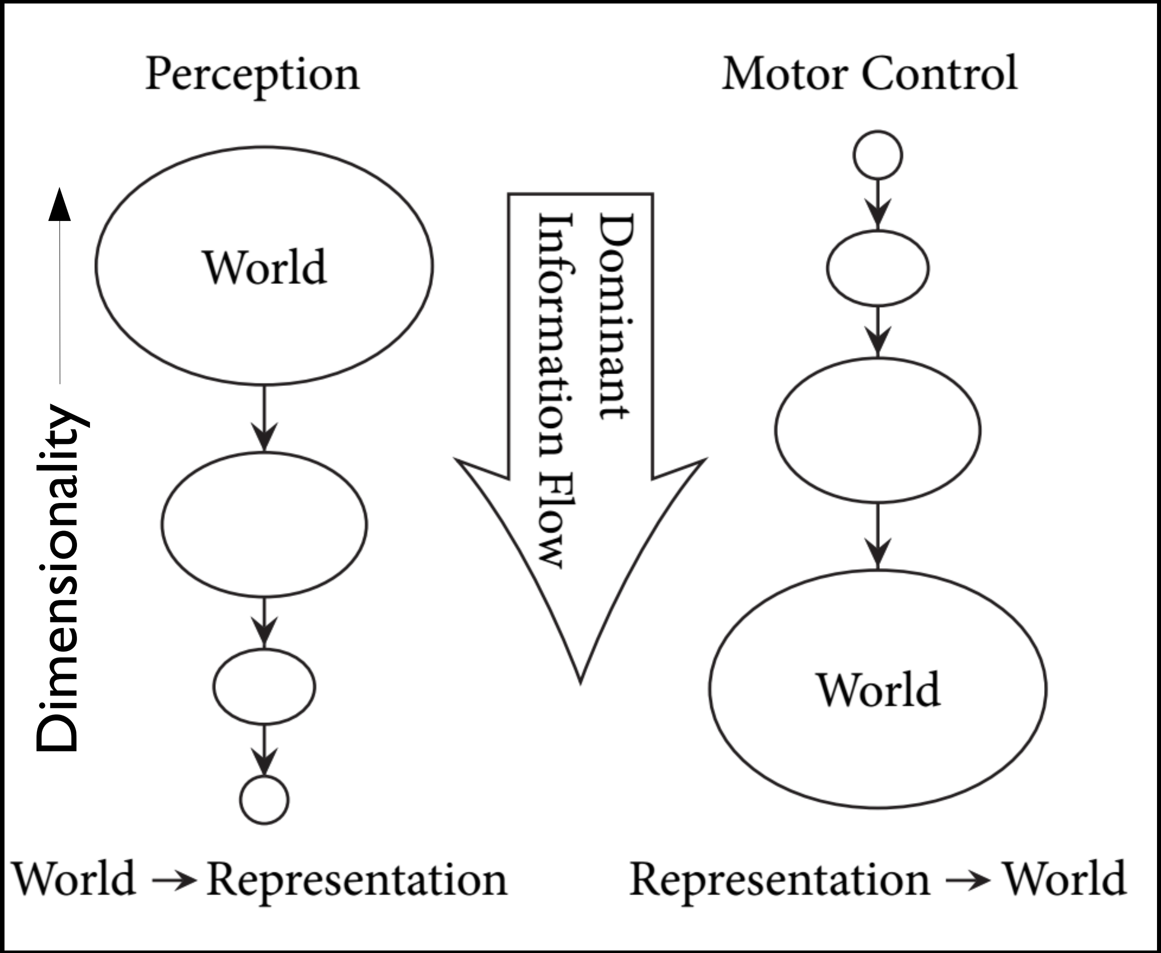
Symbol Grounding

- In models like MAP and LSA, the symbols bear no relationship to the real (physical) world
  - MAP: Random +1,-1 values
  - LSA: “Chicken-and-egg” problem: meaning of a word is its relationship to other words, whose meaning is their relationship to other words, etc.
- Symbol Grounding Problem (Harnad 1999): Without a connection to real-world objects and events, symbols are like a circuit that isn’t connected to ground, so current can’t flow.

Perceptual Symbol Systems (Barsalou 1999)

During perceptual experience, association areas in the brain capture bottom-up patterns of activation in sensory-motor areas. Later, in a top-down manner, association areas partially reactivate sensory-motor areas to implement perceptual symbols ....Through the use of selective attention, schematic representations of perceptual components are extracted from experience and stored in memory (e.g., individual memories of green, purr, hot). As memories of the same component become organized around a common frame, they implement a simulator that produces limitless simulations of the component (e.g., simulations of purr).

Grounding Semantic Pointers in Perception and Action



Adapted from Eliasmith (2012) p. 111

nengo_spa allows us to build models like this at a high level of abstraction ….