NLP Pre-Transformers

PD Dr. Juan J. Durillo

Working with text

- Working with Neural Networks requires inputs in numerical representation
  - Character based representation (e.g., ascii code)
  - Word based encoding (each word a different representation)
    - Dictionary with all possible words; representation is based on the position on this dictionary

```
from tensorflow.keras.preprocessing.text import Tokenizer
sentences = [‘Messi is the best player in the world’, ‘Barcelona
is the best team in the world’]
tokenizer = Tokenizer(num_words=100)
tokenizer.fit_on_texts(sentences)
print(tokenizer.word_index)
{'the': 1, 'is': 2, 'best': 3, 'in': 4, 'world': 5, 'messi': 6,
'player': 7, 'barcelona': 8, 'team': 9}
```

Working with text

- Alternative representation: One Hot Encoding
  - Vector of the dictionary length, with all components to 0 except 1
- Assuming the following dictionary

```
{'the': 1, 'is': 2, 'best': 3, 'in': 4, 'world': 5, 'messi':
6, 'player': 7, 'barcelona': 8, 'team': 9}
```

  - The word Messi would be represented by the vector

    [0 0 0 0 0 1 0 0 0]

  - The word player by

    [0 0 0 0 0 0 1 0 0]

  - The word the by

    [1 0 0 0 0 0 0 0 0]

Text to Sequences

- A sequence (i.e., a sentence) is simply a list of (ordered) tokens
- Previous idea could be used for representing sentences

```
{'the': 1, 'is': 2, 'best': 3, 'in': 4, 'world': 5,
'messi': 6, 'player': 7, 'barcelona': 8, 'team': 9}
```

- The sentence ‘Messi is the best player in the world’ can be represented as the array

  [6, 2, 1, 3, 7, 4, 1, 5]

- And the sentence ‘Barcelona is the best team in the world’ can be represented as the array

  [8, 2, 1, 3, 9, 4, 1, 5]

tokenizer.texts_to_sequences([str])

Text to Sequences

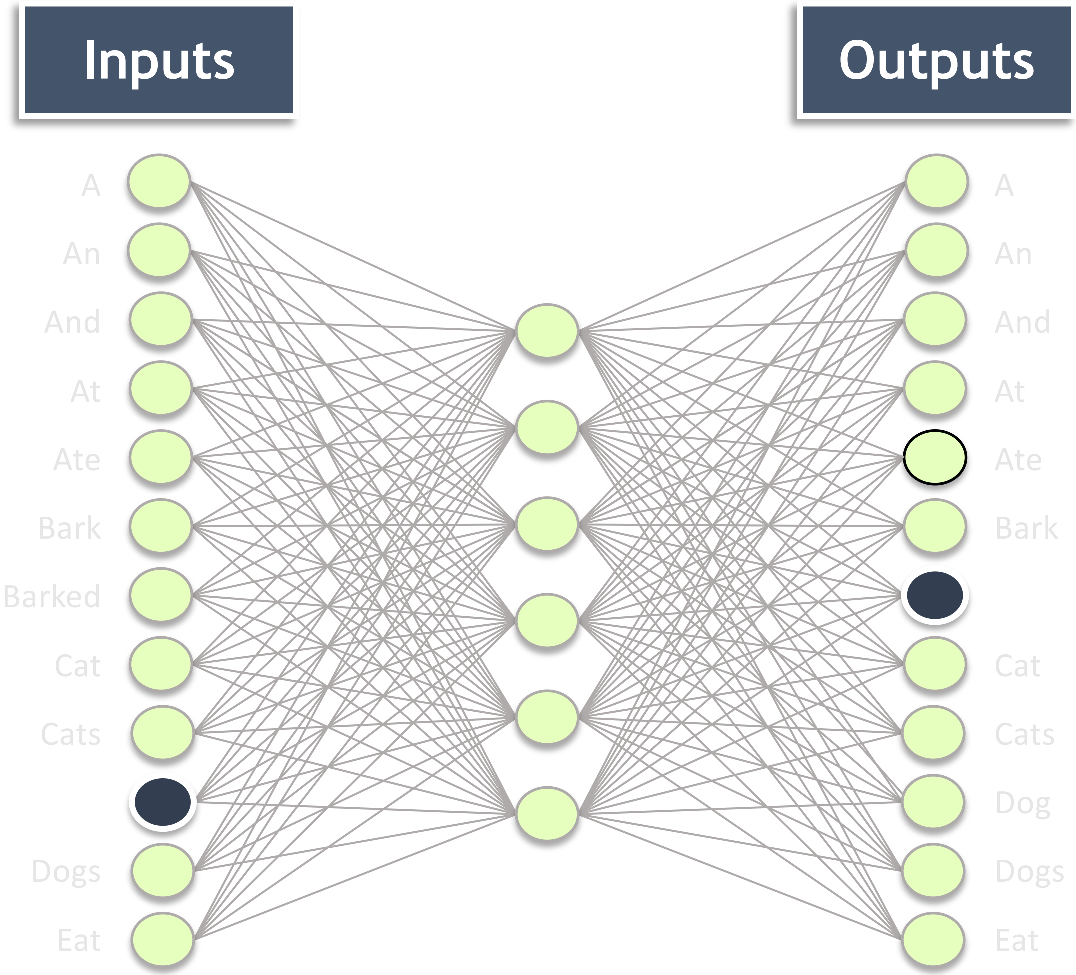
- Alternatively, if the One Hot Encoding has been chosen, given the dictionary

```
{'the': 1, 'is': 2, 'best': 3, 'in': 4, 'world': 5, 'messi': 6,
'player': 7, 'barcelona': 8, 'team': 9}
```

- The sentence ‘Messi is the best player in the world’ can be represented as the matrix

```
[[0 0 0 0 0 1 0 0 0]
 [0 1 0 0 0 0 0 0 0]
 [1 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 1 0 0]
 [0 0 0 1 0 0 0 0 0]
 [1 0 0 0 0 0 0 0 0]
 [0 0 0 0 1 0 0 0 0]]
```

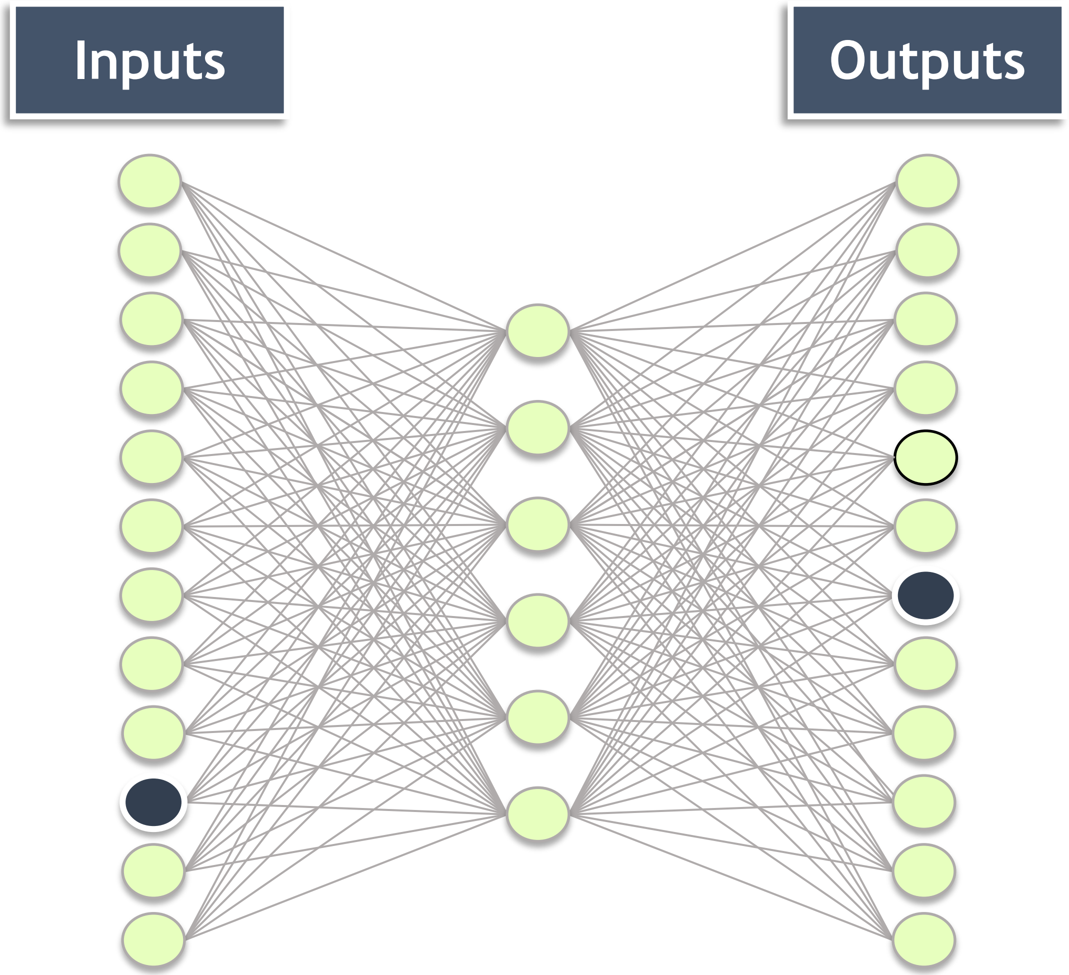
From Words to Numbers



Dictionary

1. A
2. An
3. And
4. At
5. Ate
6. Bark
7. Barked
8. Cat
9. Cats
10. Dog
11. Dogs
12. Eat

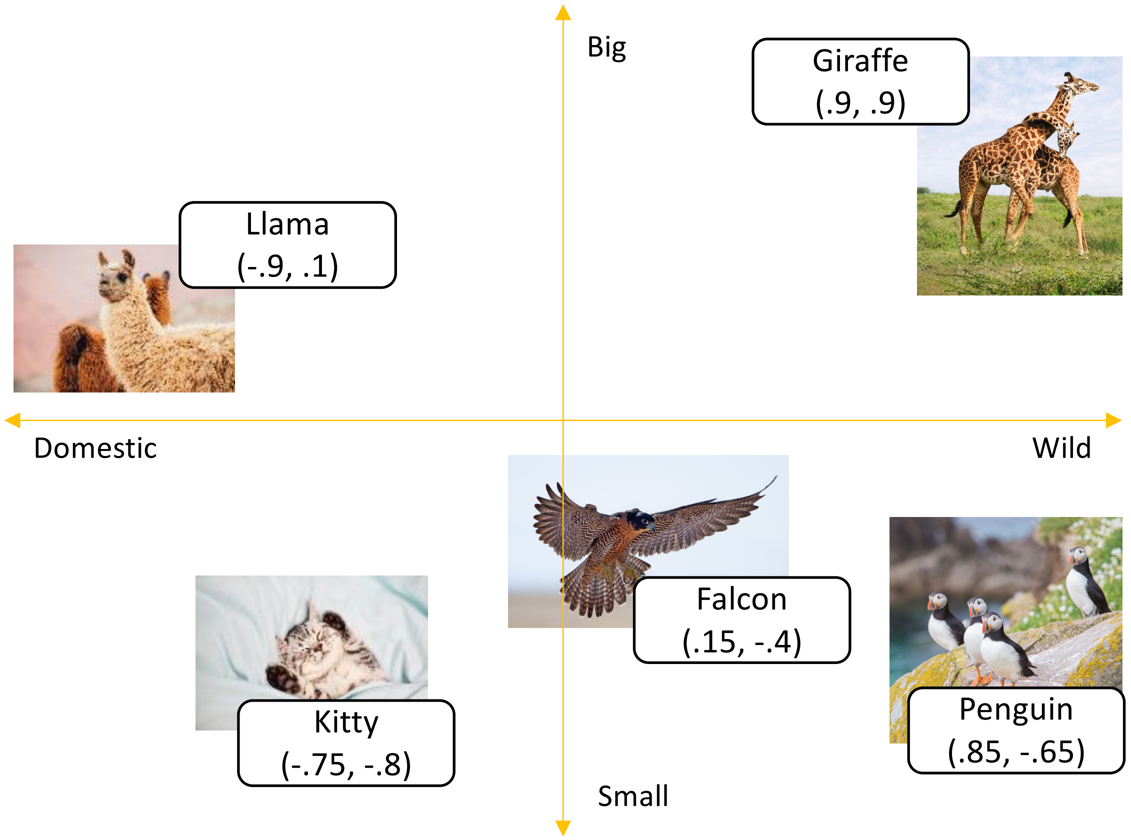
From Words to Numbers



Dictionary

1. A
2. An
3. And
4. At
5. Ate
6. Bark
7. Barked
8. Cat
9. Cats
10. Dog
11. Dogs
12. Eat

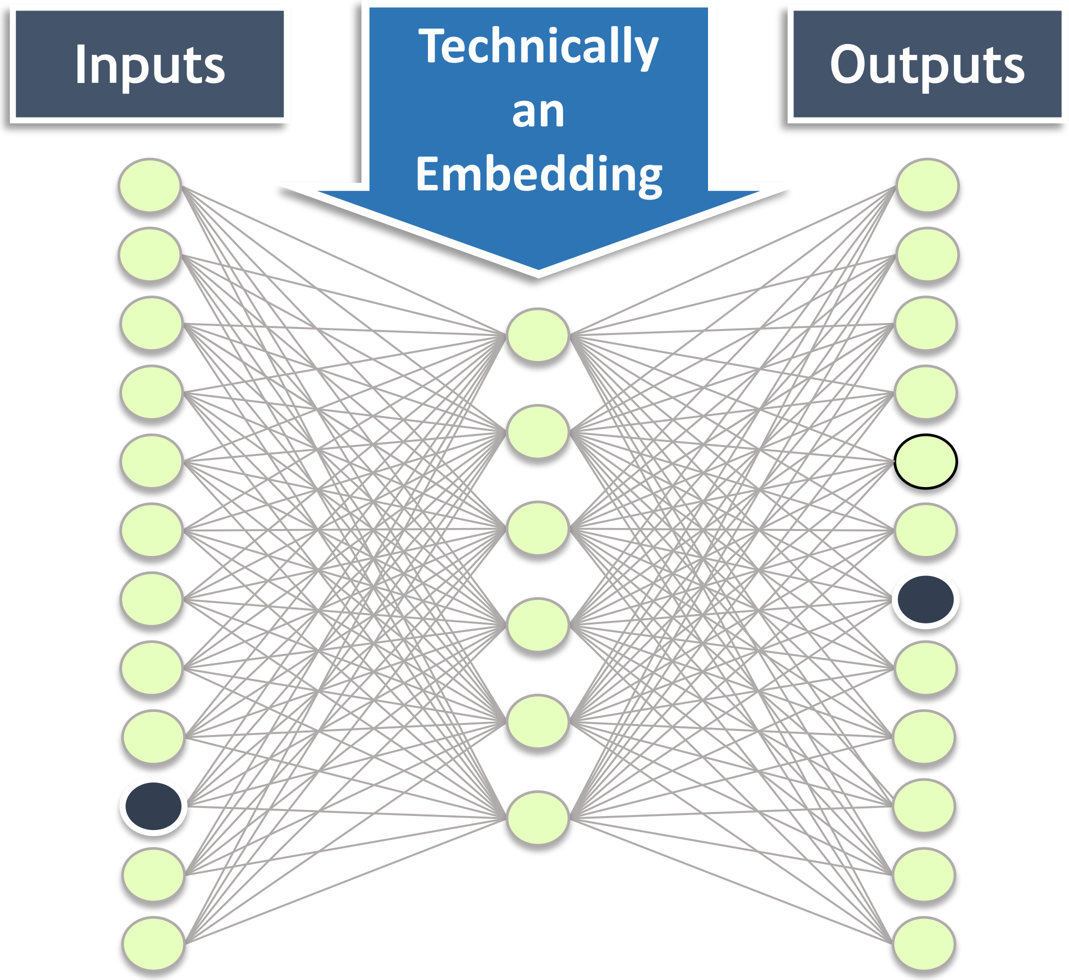
From Words to Numbers



Bigger Dictionary

1. A
2. An
3. And
4. At
5. Ate
6. Bark
7. Barked
8. Cat
9. Cats
10. Dog
11. Dogs
12. Eat
13. Eaten
14. A
15. An
16. And
17. At
18. Ate
19. Bark
20. Barked
21. Cat
22. Cats
23. Dog
24. Dogs
25. Eat
26. Eaten
27. A
28. An
29. And
30. At
31. Ate
32. Bark
33. Barked
34. Cat
35. Cats
36. Dog
37. Dogs
38. Eat
39. Eaten
40. A
41. An
42. And
43. At
44. Ate
45. Bark
46. Barked
47. Cat
48. Cats
49. Dog
50. Dogs
51. Eat
52. Eaten
53. A
54. An
55. And
56. At
57. Ate
58. Bark
59. Barked
60. Cat
61. Cats
62. Dog
63. Dogs
64. Eat
65. Eaten
66. A
67. An
68. And
69. At
70. Ate
71. Bark
72. Barked
73. Cat
74. Cats
75. Dog
76. Dogs
77. Eat
78. Eaten
79. …
80. …
81. …
82. …

From Words to Numbers



Dictionary

1. A
2. An
3. And
4. At
5. Ate
6. Bark
7. Barked
8. Cat
9. Cats
10. Dog
11. Dogs
12. Eat

Recurrent Neural Networks

Learning From Text

- If you read the partial sentence:
  - Today there is an amazing blue …
- What do you think of next?

Learning From Text

- If you read the partial sentence:
  - Today there is an amazing blue …
- What do you think of next?
  - Today there is an amazing blue sky.

Learning from Text

- If you read the partial sentence:
  - She was born in Munich, therefore at school the primary language was ….
- In contrast to the previous example, the word that influences what we need to predict now is not the previous was, but was way beyond in the text
  - Do RNN still help in this case?

Recurrent Neural Networks

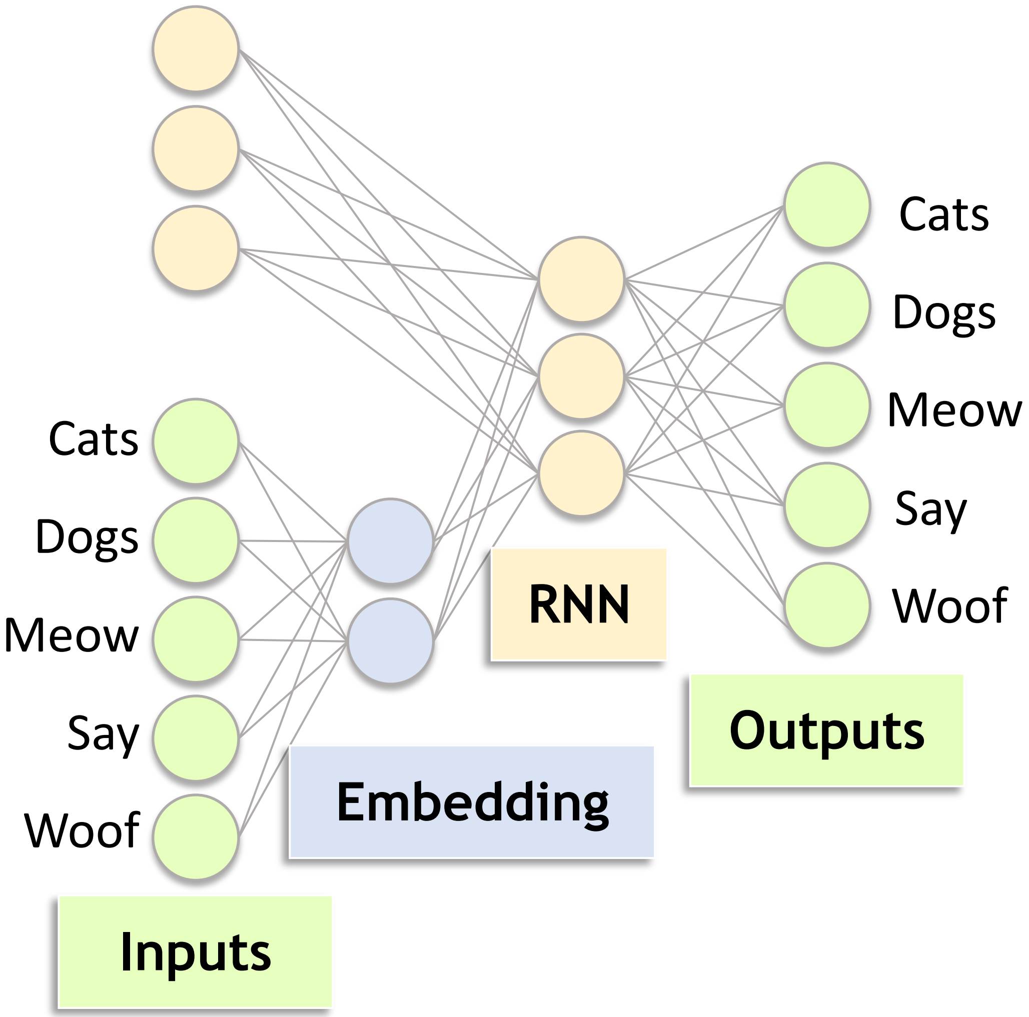
“Cats say ___.”

“Dogs say ___.”

Dictionary

1. Cats
2. Dogs
3. Meow
4. Say
5. Woof

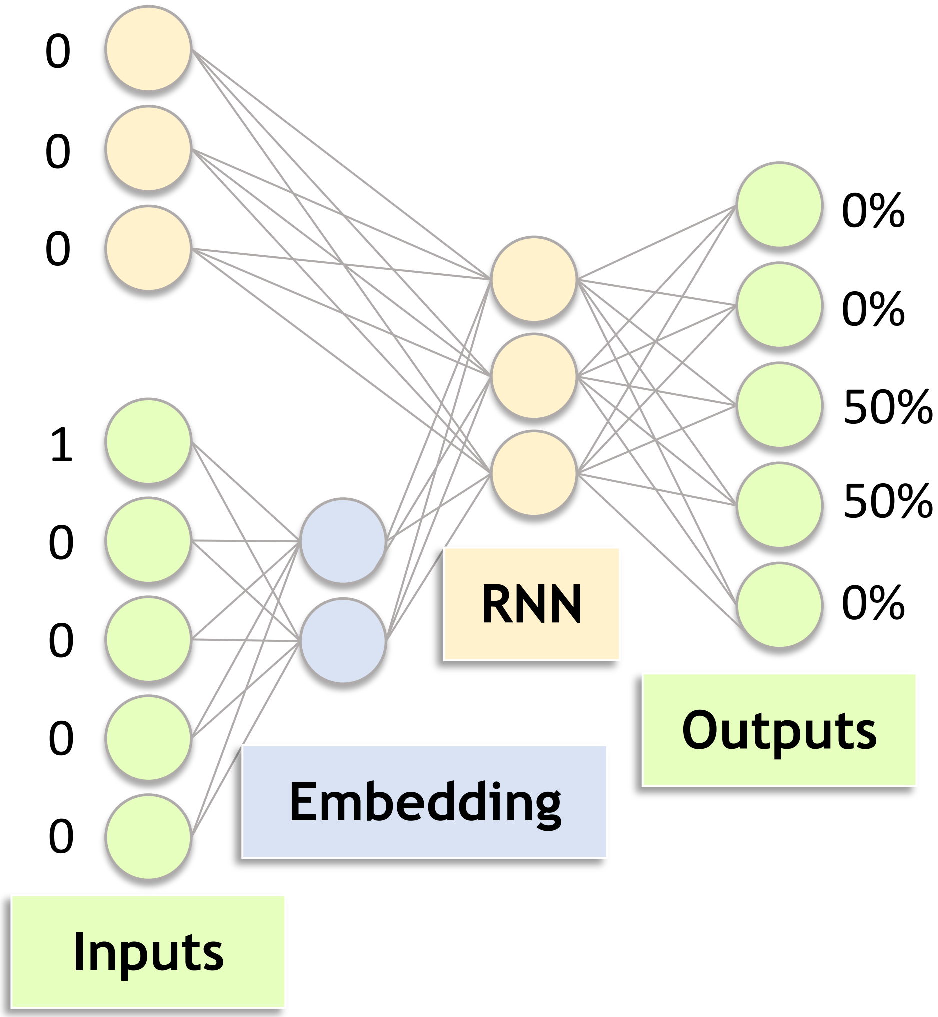
Recurrent Neural Networks



Dictionary

1. Cats
2. Dogs
3. Meow
4. Say
5. Woof

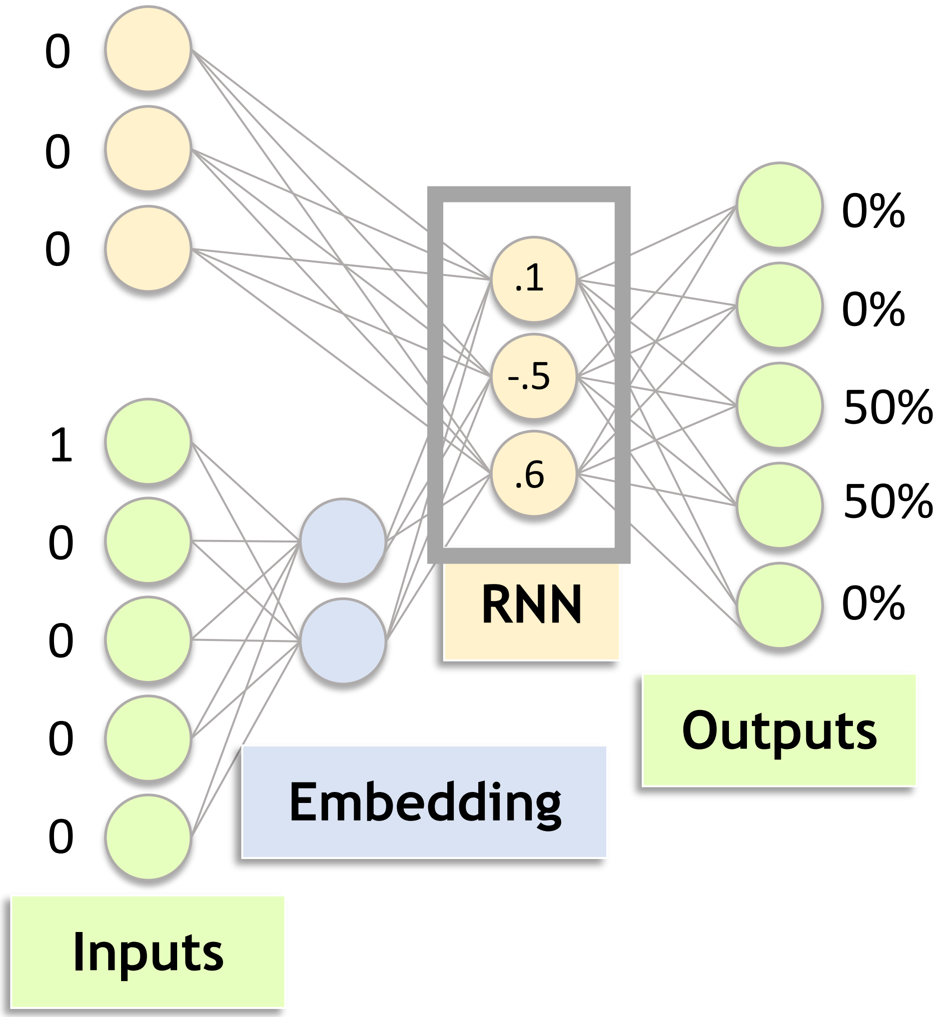
Recurrent Neural Networks



Dictionary

1. Cats
2. Dogs
3. Meow
4. Say
5. Woof

Recurrent Neural Networks



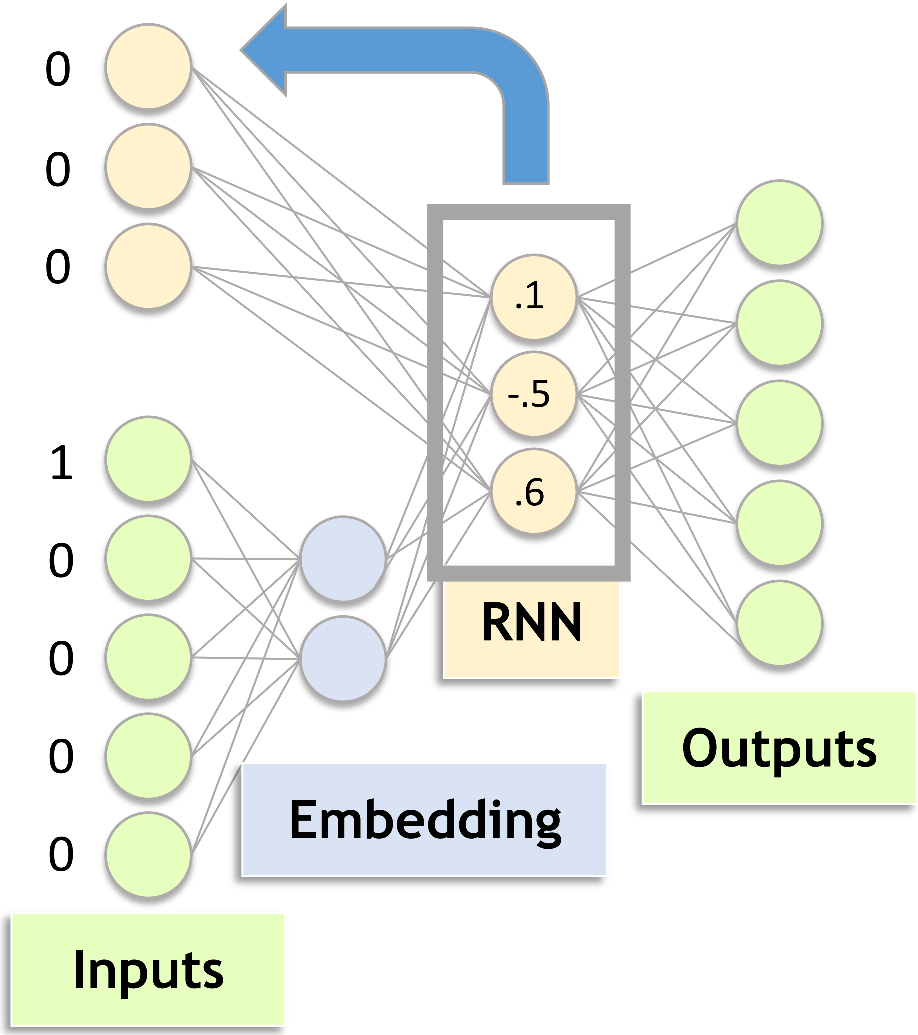
“Cats say ___.”

“Dogs say ___.”

Dictionary

1. Cats
2. Dogs
3. Meow
4. Say
5. Woof

Recurrent Neural Networks



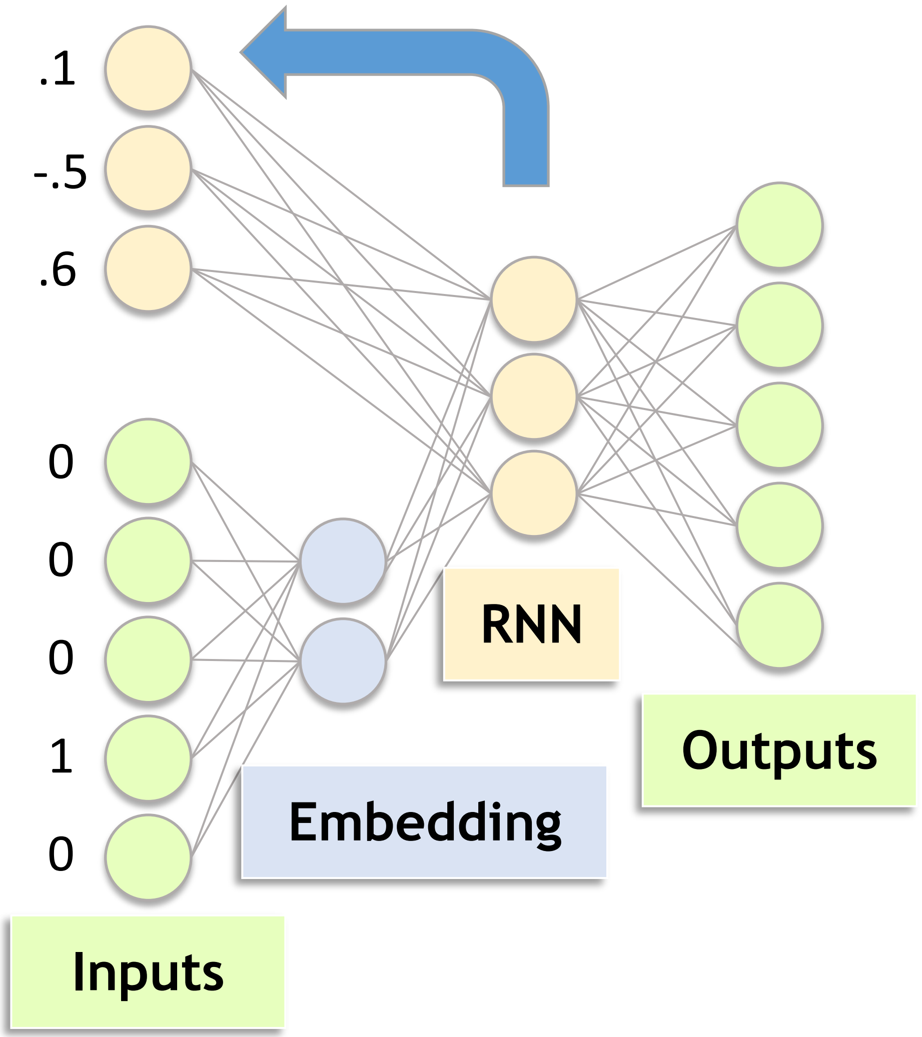
"Cats say ___."

"Dogs say ___."

Dictionary

1. Cats
2. Dogs
3. Meow
4. Say
5. Woof

Recurrent Neural Networks



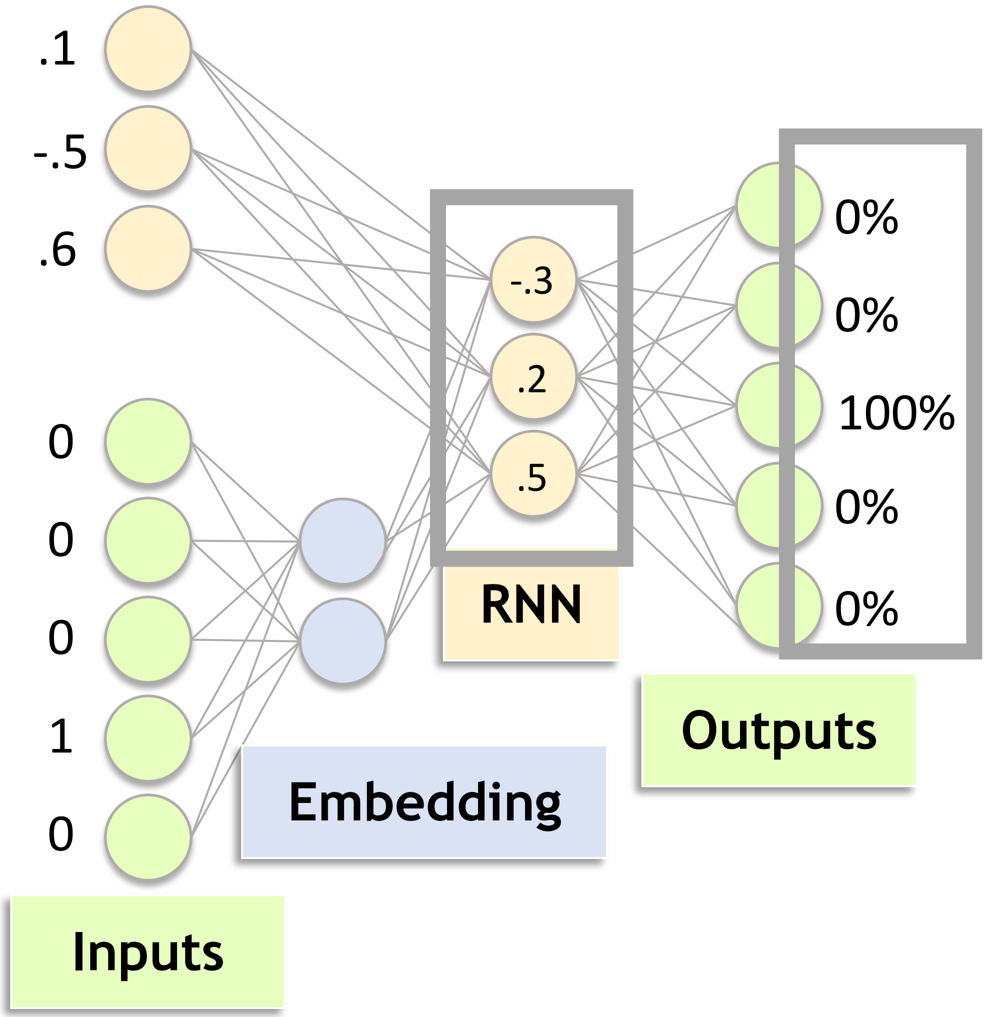
“Cats say ___.”

“Dogs say ___.”

Dictionary

1. Cats
2. Dogs
3. Meow
4. Say
5. Woof

Recurrent Neural Networks



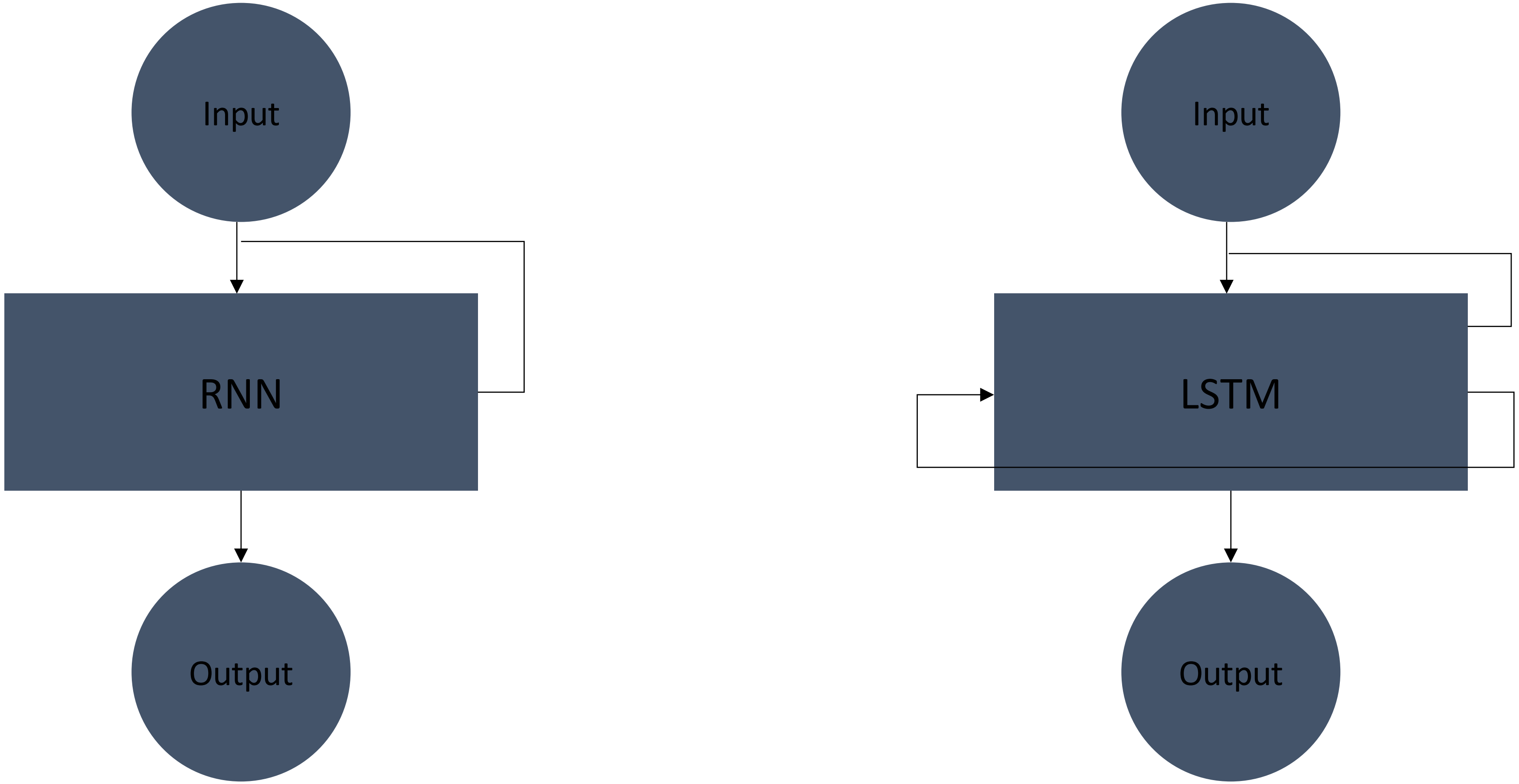
"Cats say ___."

"Dogs say ___."

Dictionary

1. Cats
2. Dogs
3. Meow
4. Say
5. Woof

Recurrent Neural Networks


Input
RNN
Output
Input
LSTM
Output