14/09/2021

Beam Search Algorithm **[ALGORITHM]:**

```
from math import log
from numpy import array
from numpy import argmax


# beam search
def beam_search_decoder(data, k):
	sequences = [[list(), 0.0]]
	# walk over each step in sequence
	for row in data:
		all_candidates = list()
		# expand each current candidate
		for i in range(len(sequences)):
			seq, score = sequences[i]
			for j in range(len(row)):
				candidate = [seq + [j], score - log(row[j])]
				all_candidates.append(candidate)
		# order all candidates by score
		ordered = sorted(all_candidates, key=lambda tup:tup[1])
		# select k best
		sequences = ordered[:k]
	return sequences
```

```
# define a sequence of 10 words over a vocab of 5 words
data = [[0.1, 0.2, 0.3, 0.4, 0.5],
                [0.5, 0.4, 0.3, 0.2, 0.1],
                [0.1, 0.2, 0.3, 0.4, 0.5],
                [0.5, 0.4, 0.3, 0.2, 0.1],
                [0.1, 0.2, 0.3, 0.4, 0.5],
                [0.5, 0.4, 0.3, 0.2, 0.1],
                [0.1, 0.2, 0.3, 0.4, 0.5],
                [0.5, 0.4, 0.3, 0.2, 0.1],
                [0.1, 0.2, 0.3, 0.4, 0.5],
                [0.5, 0.4, 0.3, 0.2, 0.1]]
data = array(data)
# decode sequence
result = beam_search_decoder(data, 3)
# print result
for seq in result:
        print(seq)
```

Output:

```
[[4, 0, 4, 0, 4, 0, 4, 0, 4, 0], 6.931471805599453]
[[4, 0, 4, 0, 4, 0, 4, 0, 4, 1], 7.154615356913663]
[[4, 0, 4, 0, 4, 0, 4, 0, 3, 0], 7.154615356913663]
```