CS6700: Reinforcement Learning Project
evaluation worksheet

19/01/2019

Problem 1

This problem aims to test your ability to understand and analyse literature. Write a brief summary and detailed critique on the paper, **Pixel RNN**(https://arxiv.org/abs/1601.067595)

Problem 2

This problem attempts to gauge your ability to write code comfortably in tensorlow:

Consider a matrix $\mathcal{A}$ as follows :

$$\mathcal{A} = \begin{bmatrix} x_{11} & x_{12} & x_{13} & . & . & . & . & x_{1n} \\ x_{21} & x_{22} & x_{23} & . & . & . & . & x_{2n} \\ x_{31} & x_{32} & x_{33} & . & . & . & . & x_{3n} \\ . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . \\ x_{n1} & x_{n2} & x_{n3} & . & . & . & . & x_{nn} \end{bmatrix}$$

The matrix must be an input placeholder whose size(n) is not known beforehand. Design a tensorflow function that computes the below enlisted operations on the matrix. All operations must be in tensorflow (not using numpy manipulations). The following operations must be fit into a single function that returns the variable *finalVal*. You can setup separate code to test this function by calling

sess.run(finalVal, feed_dict={"matrix":matrixA})

The operations are as follows:

- Transpose the elements in the bottom-right triangle of $\mathcal{A}$
- Take the maximum value along the columns of $\mathcal{A}$ to get a vector $\vec{m}$ (i.e. for each column, pick a value that is the maximum among all rows)

- Consider $\vec{m}$ to be of the form $[m^1, m^2, \cdots, m^n]$. Create a new matrix $\mathcal{B}$ such that:

$$\mathcal{B} = \begin{bmatrix} \longleftarrow\text{softmax(m[1:n])}\longrightarrow & & & & \\ \longleftarrow\text{softmax(m[1:n-1])}\longrightarrow & 0 & & & \\ \longleftarrow\text{softmax(m[1:n-2])}\rightarrow & 0 & 0 & & \\ \cdots\cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots\cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots\cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots\cdots & \cdots & \cdots & \cdots & \cdots \\ \text{softmax(m[1:1])} & 0 & \cdots\cdots & 0 & 0 & 0 \end{bmatrix} \tag{1}$$

  For example, the following vector $\vec{m} = [1.0, 2.0, 3.0]$, produces the matrix

$$\mathcal{B} = \begin{bmatrix} 0.09003057 & 0.24472847 & 0.66524096 \\ 0.26894142 & 0.73105858 & 0.0 \\ 1.0 & 0.0 & 0.0 \end{bmatrix}$$

- Sum along the rows of $\mathcal{B}$ to obtain vector $\vec{v}_1$ (along row for martix B would be $(0.90030 + 0.244 + 0.665)$)
- Sum along the columns of $\mathcal{B}$ to get another vector $\vec{v}_2$
- Concatenate the two vectors and take a softmax of this vector: $\vec{v} = \text{softmax}(\text{concat}(\vec{v}_1, \vec{v}_2))$
- Get the index number in vector $\vec{v}$ with maximum value
- Perform the following conditional computation
  - If the index number is greater than $\frac{n}{3}$ store

$$\text{finalVal} = ||v_1 - \vec{v}_2||_2$$

  - Otherwise, store:

$$\text{finalVal} = ||v_1 + \vec{v}_2||_2$$

- Return the variable *finalVal*

Figure on the next page presents a template which can also be accessed at:
https://gist.github.com/rahul13ramesh/6732040a398bf3730f2ff3b2a977ea31

```python
#!/usr/bin/env python3
import numpy as np
import tensorflow as tf


def customOps(n):
    """
    Define custom tensorflow operations
    """
    # Define placeholder
    # perform tf operations listed in list
    # Even if condition, matrix manipulaitons should be in Tensorflow
    pass


if __name__ == '__main__':
    mat = np.asarray([[0, 1],
                      [1, 0]])
    n = mat.shape[0]

    finalVal = customOps(n)

    init = tf.global_variables_initializer()
    sess = tf.Session()
    sess.run(init)
    outVal = sess.run(finalVal, feed_dict={"matA": mat})
    print(outVal)
    sess.close()

```