Package ‘attention’

November 10, 2023

Title Self-Attention Algorithm

Version 0.4.0

Description Self-Attention algorithm helper functions and demonstration vignettes of increasing depth on how to construct the Self-Attention algorithm, this is based on Vaswani et al. (2017) <doi:10.48550/arXiv.1706.03762>, Dan Jurafsky and James H. Martin (2022, ISBN:978-0131873216) <https://web.stanford.edu/~jurafsky/slp3/> ``Speech and Language Processing (3rd ed.)'' and Alex Graves (2020) <https://www.youtube.com/watch?v=AIiwuClvH6k> ``Attention and Memory in Deep Learning''.

License GPL (>= 3)

Encoding UTF-8

RoxygenNote 7.2.3

Suggests covr, knitr, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

NeedsCompilation no

Author Bastiaan Quast [aut, cre] (<https://orcid.org/0000-0002-2951-3577>)

Maintainer Bastiaan Quast <bquast@gmail.com>

Repository CRAN

Date/Publication 2023-11-10 03:10:02 UTC

R topics documented:

attention 2
ComputeWeights 2
RowMax 3
SoftMax 4

Index **6**

`attention` *Attnention mechanism*

Description

Attnention mechanism

Usage

```
attention(Q, K, V, mask = NULL)
```

Arguments

`Q`	queries
`K`	keys
`V`	values
`mask`	optional mask

Value

attention values

`ComputeWeights` *SoftMax sigmoid function*

Description

SoftMax sigmoid function

Usage

```
ComputeWeights(scores)
```

Arguments

`scores`	input value (numeric)

Value

output value (numeric)

Examples

```
# Set up a scores matrix
scores <- matrix(c( 6,  4, 10,  5,
                    4,  6, 10,  6,
                   10, 10, 20, 11,
                    3,  1,  4,  2),
                 nrow  = 4,
                 ncol  = 4,
                 byrow = TRUE)

# Compute the weights based on the scores matrix
ComputeWeights(scores)

# this outputs
#            [,1]       [,2]      [,3]       [,4]
# [1,] 0.10679806 0.03928881 0.7891368 0.06477630
# [2,] 0.03770440 0.10249120 0.7573132 0.10249120
# [3,] 0.00657627 0.00657627 0.9760050 0.01084244
# [4,] 0.27600434 0.10153632 0.4550542 0.16740510
```

RowMax *Maximum of Matrix Rows*

Description

Maximum of Matrix Rows

Usage

```
RowMax(x)
```

Arguments

x	input value (numeric)

Value

output value (numeric)

Examples

```
# generate a matrix of integers (also works for floats)
set.seed(0)
M = matrix(floor(runif(9, min=0, max=3)),
           nrow=3,
           ncol=3)
print(M)
```

```
# this outputs
#      [,1] [,2] [,3]
# [1,]    2    1    2
# [2,]    0    2    2
# [3,]    1    0    1

# apply RowMax() to the matrix M, reformat output as matrix again
# to keep the maxs on their corresponding rows
RowMax(M)

# this outputs
#      [,1]
# [1,]    2
# [2,]    2
# [3,]    1
```

`SoftMax` *SoftMax sigmoid function*

Description

SoftMax sigmoid function

Usage

```
SoftMax(x)
```

Arguments

`x`	input value (numeric)

Value

output value (numeric)

Examples

```
# create a vector of integers (also works for non-integers)
set.seed(0)
V = c(floor(runif(9, min=-3, max=3)))
print(V)

# this outputs
# [1]  2 -2 -1  0  2 -2  2  2  0

# apply the SoftMax() function to V
sV <- SoftMax(V)
print(sV)
```

```
# this outputs
# [1] 0.229511038 0.004203641 0.011426682 0.031060941
# 0.229511038 0.004203641 0.229511038 0.229511038 0.031060941
```

Index

attention, 2

ComputeWeights, 2

RowMax, 3

SoftMax, 4