Day 11, 07th Jan 2024

NLP Live Class – by mentor Dr Ajay Dhamija

Ph D from IIT Delhi & Harvard University

Recap – RNN → Backpropagation

Coding → Tensorflow, RNN (basic + our project)

Topic – Code, RNN types & details, Bi directional RNN

V.V. imp
interview

Tokenizer
pad sequence
Embedding → embedded layers in RNN

d = dimension
Embedding layer.

SES → RNN
↳ Sequential
Keras
Tensorflow

layer → RNN
layer → Dense
model.

classification

EVE classification
softmax
softmax f/c
IMDB

Sigmoid → Binary Classification
activation f/c

RNN → 0 →

Tensor → model fit

model predict

1 - sigmoid

$z = f(y) = f(\sum x_i w_i + w_0)$
$= f(0.4)$
$= 0 \rightarrow$ class → -ve Review
$z = f(y) = f(0.54) = 1 \rightarrow$ +ve Review

→ RNN →

↳ Embedding → RNN

vector → Representation

RNN advantage → sequence of data
Sequence i/p

RNN → Disadvantage → Vanishing Gradient Problem, Exploding gradient problem

V.V. imp
interview

Backpropagation

↳ weight updation

$L = y - \hat{y}$

$w_n = w_n - \eta \frac{\partial L}{\partial w_n}$

input layer
hidden layer
output layer

chain Rule

$w_3^n = w_3^o - \eta \frac{\partial L}{\partial w_3^o}$

see product

$\frac{\partial L}{\partial w_3^o} = \frac{\partial L}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial o_3} \frac{\partial o_3}{\partial w_3^o}$

$\frac{\partial o_3}{\partial w_3^o} = 0.02$

$w_3^n = w_3^o - \eta \frac{\partial L}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial o_3} \frac{\partial o_3}{\partial o_2} \frac{\partial o_2}{\partial w_3^o}$

$\frac{\partial o_2}{\partial w_3^o} = 0.02$

$f: x^2 \rightarrow x^2$

$\frac{df}{dx} = 2x \rightarrow 3x$

output → 0.01

$0.65 \times 0.06 \times 0.02$

Smaller no. × smaller

$0.026 \times 0.065 = 0.001695$

→ chain rule of differentiation

$w_3^n = w_3^o - \eta \left[\text{small} \times \text{small} \times \text{small} \times \ldots \right]$
≈ 0

$w_3^n \approx w_3^o$ → old weight & new weight are similar

No weight update

Gradient Descent

loss will not reduce
model will not learn

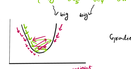
'Vanishing gradient problem'
V.V. imp

$y = f(\sum x_i w_i)$

$w_3^n = w_3^o - \eta \frac{\partial L}{\partial w_3^o}$

$w_3^n = w_3^o - \eta \frac{\partial L}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial o_3} \frac{\partial o_3}{\partial o_2} \frac{\partial o_2}{\partial w_3^o}$

Sigmoid



$\frac{\partial \hat{y}}{\partial o_3}$, 0 sigmoid

0.25

Tanh

Exploding Gradient Problem

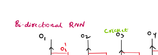
$w_3^n = w_3^o - \eta \frac{\partial L}{\partial w_3^o}$

see product

$\frac{\partial L}{\partial w_3^o} = \frac{\partial L}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial o_3} \frac{\partial o_3}{\partial o_2} \frac{\partial o_2}{\partial w_3^o}$

$w_3^n = w_3^o - \eta \frac{\partial L}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial o_3} \frac{\partial o_3}{\partial o_2} \frac{\partial o_2}{\partial w_3^o}$

big weight



Gradient descent

global minima → minimum loss

oscillating around global minima without converging

RNN → Disadvantage → ① VGP,
② EGP
③ forgets of input sequence is long
④ future sequence ✗

IMDB movie rating dataset
↳ 218 words

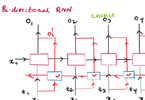
predictive dependent on future seq

example → I love to Eat. Cricket is a good game, played between two team. Each team has 11 player.

Eat → wrong prediction
cricket → right prediction

Past Sequence & Future Sequence → RNN
↳ Bi directional RNN

Bi directional RNN



I love cricket good game

Output → RNN or Bi RNN ??

generally, accuracy of BiRNN is improved

Dataset, problem statement → RNN > BiRNN || BiRNN > RNN

BiRNN → Disadvantage → complexity
Computation cost ↑
Computation resource
model Training ↑
Processing time ↑

BiRNN → advantage → ① Process both past & future sequence
② ability to handle sequence dependence
③ resistant to noise
④ less dependent on irrelevant info

Applications of BiRNN

① Sentiment analysis
② Text classification
③ Named Entity Recognition
④ Part of speech Tagging
⑤ Language Translation / machine Translation