

RNNの概要は前の図　で示した通り、

はｔ時刻の入力

はｔ時刻の中間層の出力

は誤差

は入力層から中間層の重み

は中間層同士の重み

は中間層から出力層への重み

時刻ｔの中間層の計算をする際に入力層からの入力とt-1時刻の中間層の出力に重み付けを行った結果を加算したものを中間層への入力として扱う。この際にポイントが2つある。

一つ目は、一般的なRNNでは時刻のｔ＝0のときには時刻t-1の中間層が存在しないため、過去の時間からの入力はすべての値が0のものを時刻t-1の中間層からの入力として設定する。

ある時刻tにおける中間層からの出力を、時刻tにおける入力

t时刻の誤差関数 定义为交叉熵损失

在这里，是时刻t上正确的值， 为预测出来的值。使用一整个序列作为一次训练，每个时刻的误差累计构成总的误差。

だから、時刻tにおける誤差をとすると、合計誤差は次のように表すことができる。

これから、すべての重み、、、についてそれぞれの勾配を計算する

まずは、連鎖律を用いると、重みに関する誤差関数の偏微分は以下のようになる。

次は、についての勾配を計算すると以下のようになる。

はｔ時刻の中間層の出力を表すため、式（１）によると、はの関数であり、に関する計算はとに依存する。またt-1のときに、であり、 に関する計算はとに依存する。だからを展開する必要がある。

を展開すると

そして、を展開すると、

式（８）を式（7）に代入すると

これにより、式(6)のという部分が展開された。

式（9）を式（6）に代入すると

最後に、重みに関する誤差関数の偏微分を計算する。

の計算と同様に、連鎖律を用いると、重みに関する誤差関数の偏微分は以下のようになる

を展開すると、

-

RNNの勾配に関する問題点

先ほど計算した勾配の特性上、普通のRNNは長時間のデータを意識した学習をすることができないという問題が起こる。

重みに関する誤差関数の偏微分との計算に関しては、時刻tだけではなくて、時刻kからtまで、すべての時刻を一緒に計算する必要があるから、また、総乗の部分が入っているため、時刻ｊと時刻ｋの距離が大きく離れた場合は勾配爆発と勾配消失の問題が発生する。

なぜなら、

の計算結果は

tanh関数の導関数は｛｝

图 tan 函数和tan函数的导数

よって：

勾配爆発が発生するの原因は

>4のとき、 、もしJとKの間の距離が大きすぎると、連乗も多すぎて，勾配爆発が発生し，無限に近づいてしまう。

勾配消失が発生する原因は

<4のとき、であるため、距离过大，会导致连乘项过多，产生梯度消失，趋近于0

勾配消失が起こると、ある程度先の時刻以降の誤差は全く考慮しないで重みを更新するという現象が起こっている。実際には4－５時刻程先の誤差を考慮した勾配を計算するとほぼ0になる。

この勾配消失の問題に対する対策として、簡単な方法としては活性化関数としてReLUを用いることが挙げられる。

しかし、ReLUは正の領域での勾配が常に１になるので勾配消失が起こりにくくなる。しかし、マイナスの値を全て0にしてしまうので長期時刻順伝播させる際に値がほとんど0になってしまう可能性がある。

RNNもう1つの問題点として、入力層―中間層、中間層―中間層、中間層―出力層で常に共通の重みを用いるために、重要な入力を通すために重みを大きくするように学習が進んでしまうと逆に時系列上にある不必要な情報も大きく通すようになっていまい、

そのため、RNNの重みは学習の際に常に矛盾したアップデートをされていることにより、学習がなかなか進まない。

これらの問題を、時間展開される中間層の構造を変化させることにより回避したのがLong　Short　Term　Memory（LSTM）である。

改良したRNN―LSTM

LSTM的定义和结构

LSTMはHochreiter&Schmidhuber在1997年提出的，目前在NLP领域已经取得了很好的应用效果，远远优于其他算法。

LSTMは勾配爆発問題と勾配消失問題をCEC（Constant　Error　Carousel）とゲートという概念を導入するにより解決することができる。

従来のRNNの中間層出力は前の時刻の出力と今の入力の重み付き線形和に活性化関数を作用されるだけの形だったため、時系列が長くなればなるほど活性化関数にネストされて、勾配消失が起こる。

CECは過去の情報を保存するためのユニットである。例えば、前の時刻ｔ－１の出力　と今の入力　の重み付き線形和をそのまま通すではなく、前の時刻ｔ－１のCECの値を加算して出力される。

ゲートに関しては、ゲートの導入することによってLSTMでは各時刻ごとに入出力を制御することが可能になる。

入力ゲート

忘却ゲート

出力ゲート

Z则是将结果通过tanh函数转换成-1到1之间的值

