LSTM

|  |  |
| --- | --- |
| C:\Users\jakob\AppData\Local\Microsoft\Windows\INetCache\Content.Word\LSTM_Cell.png  http://colah.github.io/posts/2015-08-Understanding-LSTMs/img/LSTM2-notation.png | Spomin celice  Izhod iz celice  Spomin prejsnje celice omrezja  Izhod iz prejsnje celice omrezja  Vhod v celico  Izhod iz vrat za pozabljanje  Izhod iz filtra za adicijska vrata  Izhod iz filtra za izhodna vrata  Kandidati za adicijska vrata |

**Forward Propagation**

Forward propagation je metoda, ki prejme , in vrne ter . gre preko vec vrat, predenj se vrne kot in

se najprej filtrira preko vrat za pozabljanje

|  |  |
| --- | --- |
|  |  |



se doda vektor iz adicijskih vrat in to je

|  |  |
| --- | --- |
|  |  |



Iz , lahko dobimo , tako da najprej damo v tanh funkcijo in nato se filtrira preko izhodnih vrat.

|  |  |
| --- | --- |
|  |  |



**Backpropagation**

Backpropagation je učenje omrežja. Ko celica vrne izhod, lahko pogledamo, koliko je zgrešil. To naredimo z »error« funkcijo. V mojem primeru »mean squared error«:



Ce hočemo predvidevati naslednjo črko v besedi »test«. Slovar: { 0 : 't', 1 : 'e', 2 : 's'}.

Torej najprej imamo vhod 't' in zeleni izhod 'e':

Vhod: , zeleni izhod: . Torej ce nam celica vrne , to pomeni da je error:

Torej mi hočemo to število cim bolj zmanjšati. To lahko naredimo z »gradient descent«. To pomeni da dobimo odvod od »error funkcije« glede na en parameter ( sinapso ) in zmanjšamo to sinapso za toliko. To ponovimo za vse sinapse.

Odvod glede na htn

Odvod glede na Ctn

+

Zanji element, je samo, ce celica ni zadnja v verigi

Odvod glede na S0

Odvod glede na S1

Odvod glede na S2

Odvod glede na T0

Odvod glede na ht-1

Odvod glede na Ct-1

T0 sem ponekod napisal kot S3 zaradi lažjega dela z tabelami

Odvod glede na htn



Odvod glede na Ctn



Odvod glede na S0



Odvod glede na S1



Odvod glede na S2



Odvod glede na T0



Odvod glede na ht-1

Odvod glede na Ct-1

