Recurrent Neural Network

Danish Dewani
Electrical and
Computer Engineering
North Carolina State University
Raleigh, NC
Email: dadewani@ncsu.edu

Priya Diwakar
Electrical and
Computer Engineering
North Carolina State University
Raleigh, NC
Email: psdiwaka@ncsu.edu

Kalyan Ghosh
Electrical and
Computer Engineering
North Carolina State University
Raleigh, NC
Email: kghosh@ncsu.edu

***Abstract*—Design Long Short Term Memory (LSTM) cells for counting digits in a sequence and perform Language modeling using Recurrent Neural Network (RNN) with LSTM units.**

***Index Terms*—Long Short Term Memory, LSTM, Recurrent Neural Network, RNN, count, digits, language modeling**

I. Introduction

This document is the documentation for the Project5 as a part of the course ECE 542 for Fall 2018.

II. Counting digits in a sequence

We manually choose the parameters for the LSTM cells to count the digit '0' in a given sequence under two different criteria.

1) Task 2: Count number of 0 after receiving the first 2 in a given sequence.

2) Task 3: Count number of 0 after receiving the 2 in the sequence, but erase the counts after receiving 3, then continue to count from 0 after receiving another 2.

A. Structure of LSTM

At the g node of Input gate we get the output as described in (1)

$$g^{(t)} = tanh(W_{xg}^{T}x^{(t)} + W_{hg}^{T}h^{(t-1)} + b_g) \tag{1}$$

At the i node of Input gate we get the output as described in (2)

$$i^{(t)} = \sigma(W_{xi}^{T}x^{(t)} + W_{hi}^{T}h^{(t-1)} + b_i) \tag{2}$$

At the Forget gate we get the output as described in (3)

$$f^{(t)} = \sigma(W_{xf}^{T}x^{(t)} + W_{hf}^{T}h^{(t-1)} + b_f) \tag{3}$$

At the Output gate we get the output as described in (4)

$$o^{(t)} = \sigma(W_{ox}^{T}x^{(t)} + W_{oh}^{T}h^{(t-1)} + b_o) \tag{4}$$

Then we have internal state $c^{(t)}$ is as described in (5)

$$c^{(t)} = f^{(t)} \otimes c^{(t-1)} + i^{(t)} \otimes g^{(t)} \tag{5}$$

Finally the output of the LSTM cell $y^{(t)}$ is as described in (6)

$$y^{(t)} = h^{(t)} = o^{(t)} \otimes c^{(t)} \tag{6}$$

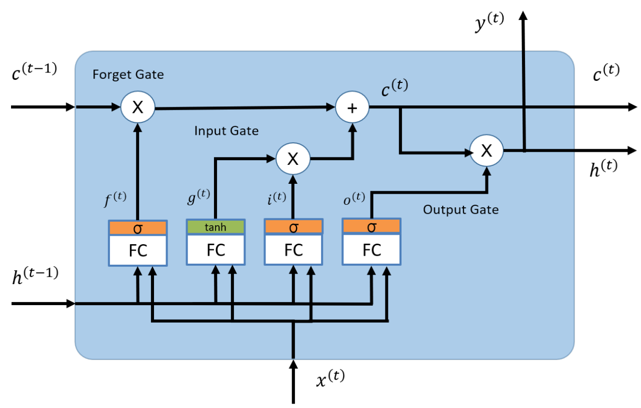

Fig. 1. LSTM Structure

B. Parameters for Task 2 and Task 3

1) Task 2: The parameters required to perform this task are:

- For this task, we use a 2 dimensional internal state for higher capacity.
- We find the optimal set of parameters by manipulating the weights and biases of the tanh and sigmoid Fully Connected Layers of the Input Gate.
- We use the indim=10 and the outdim=2
- Initially the weight matrix of the g input for the input value ('wgx') is set to ['wgx']=np.zeros((indim,outdim)). Then the weights for the 0 digit and 2 digit are initialized to ['wgx'][0]=[0.,100.] & ['wgx'][2]=[100.,0.].
- The weight matrix of the g input for the previous state ('wgh') is set to ['wgh'] = np.zeros((outdim, outdim)) and the bias term ('bg') is set to ['bg'] = np.zeros((1, outdim)).
- The weight matrix of the i input of the input gate for the input value ('wix') is set to ['wix'] $=-100$*np.ones((indim,outdim)) and then assign the value for the digit 2 as ['wix'][2]=[100.,-100.].
- The weight matrix of the i input for the previous state ('wih') is set to ['wih'] = 200*np.ones((outdim, outdim)) and the bias term ('bi') is set to ['bi'] = np.zeros((1, outdim)).
- Here we keep the Forget Gate always ON by keeping

the bias term ['bf'] = 100. * np.ones((1, outdim)). The weight matrix for the input value ('wfx') is set to ['wfx'] = np.zeros((indim, outdim)). The weight matrix for the previous state ('wfh') is set to ['wfh'] = np.zeros((outdim, outdim)).
- Here we also keep the Output Gate always ON by keeping the bias term ['bo'] = 100. * np.ones((1, outdim)). The weight matrix for the input value ('wox') is set to ['wox'] = np.zeros((indim, outdim)). The weight matrix for the previous state ('woh') is set to ['woh'] = np.zeros((outdim, outdim)).

2) Task 3: The parameters required to perform this task are:

- For this task, we use a 2 dimensional internal state for higher capacity.
- We find the optimal set of parameters by manipulating the weights and biases of the tanh and sigmoid Fully Connected Layers of the Input Gate.
- We use the indim=10 and the outdim=2
- Initially the weight matrix of the g input for the input value ('wgx') is set to ['wgx']=np.zeros((indim,outdim)). Then the weights for the 0 digit and 2 digit are initialized to ['wgx'][0]=[0.,100.] & ['wgx'][2]=[100.,0.].
- The weight matrix of the g input for the previous state ('wgh') is set to ['wgh'] = np.zeros((outdim, outdim)) and the bias term ('bg') is set to ['bg'] = np.zeros((1, outdim)).
- The weight matrix of the i input of the input gate for the input value ('wix') is set to ['wix']=−100*np.ones((indim,outdim)) and then assign the value for the digit 2 as ['wix'][2]=[100.,-100.].
- The weight matrix of the i input for the previous state ('wih') is set to ['wih'] = 200*np.ones((outdim, outdim)) and the bias term ('bi') is set to ['bi'] = np.zeros((1, outdim)).
- Here we use the Forget Gate to reset the memory of the LSTM cell each time we encounter a 3 in the sequence. The weight matrix of the forget gate for all inputs except 3 is set as ['wfx'] = 100* np.ones((indim, outdim)) and for the digit 3 is set as ['wfx'][3] = [-100. , -100.] so that the gate closes when it sees a 3 in the sequence.
- The weight matrix of the forget gate for the previous state ('wfh') is set to ['wfh'] = np.zeros((outdim, outdim)) and the bias term ('bf') is set to ['bf'] = np.zeros((1, outdim)).
- Here we also keep the Output Gate always ON by keeping the bias term ['bo'] = 100. * np.ones((1, outdim)). The weight matrix for the input value ('wox') is set to ['wox'] = np.zeros((indim, outdim)). The weight matrix for the previous state ('woh') is set to ['woh'] = np.zeros((outdim, outdim)).

C. Plot values of input node, internal state, input gate, forget gate and output gate as functions of time step.

1) Plots for Task 2 : Plotting timing diagrams for Task 2 :

Note: Time is plotted in the X axis and the states are plotted in the Y axis. Each new arrival of input sequence index is considered as a new discrete time interval.

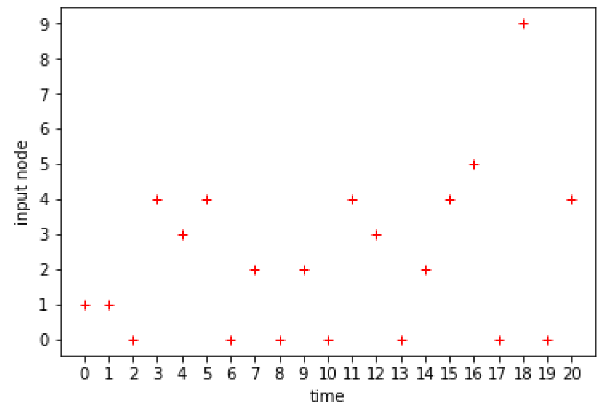

Fig. 2. Input vs Time - Task2

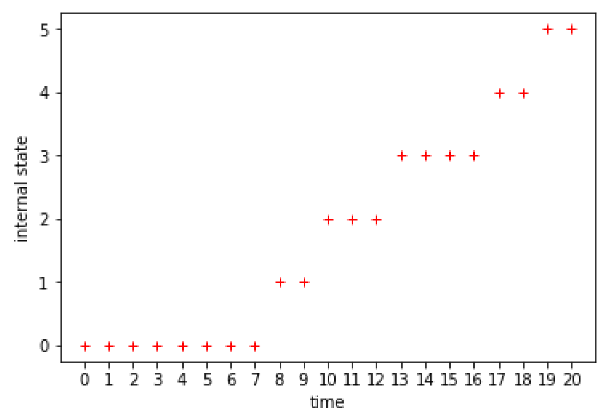

Fig. 3. Internal State vs Time - Task2

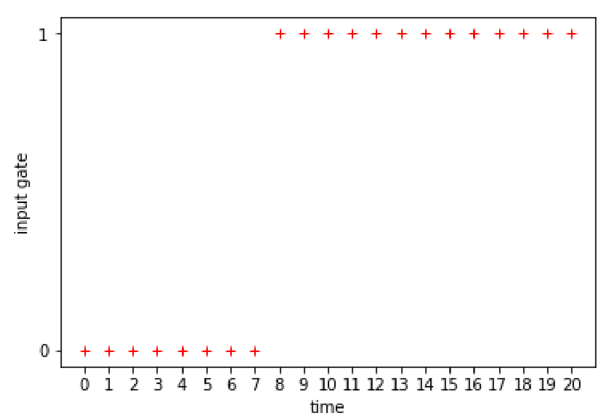

Fig. 4. Input Gate vs Time - Task2

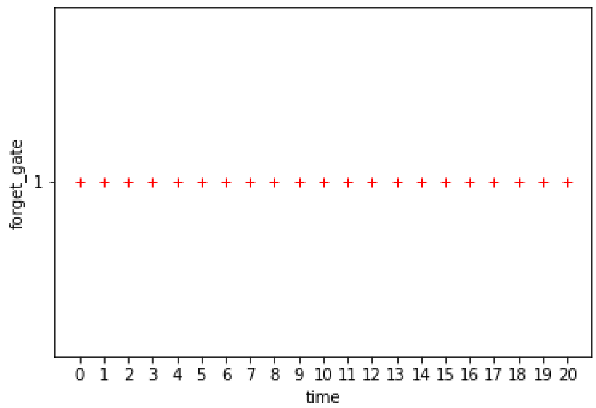

Fig. 5. Forget Gate vs Time - Task2

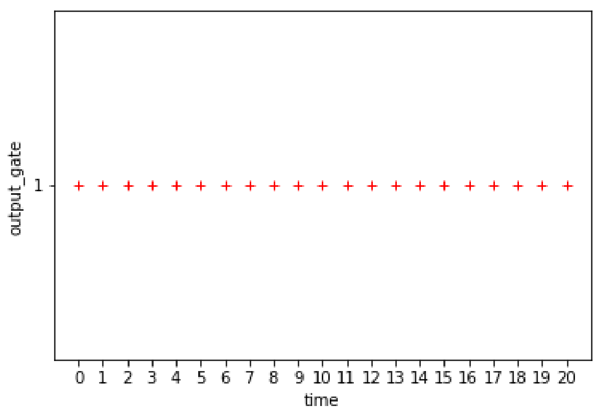

Fig. 6. Output Gate vs Time - Task2

2) Plots for Task 3: Plotting timing diagrams for Task 3:

Note: Time is plotted in the X axis and the states are plotted in the Y axis. Each new arrival of input sequence index is considered as a new discrete time interval.

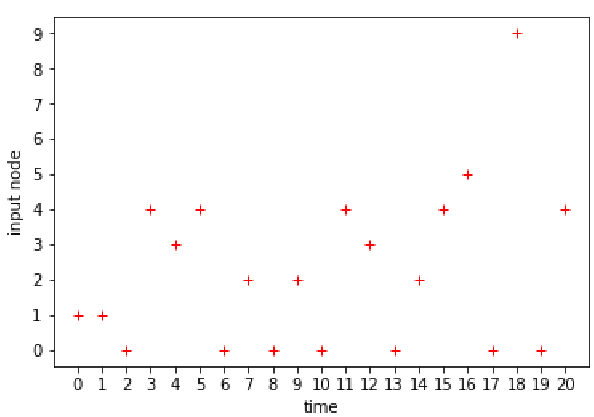

Fig. 7. Input vs Time - Task3

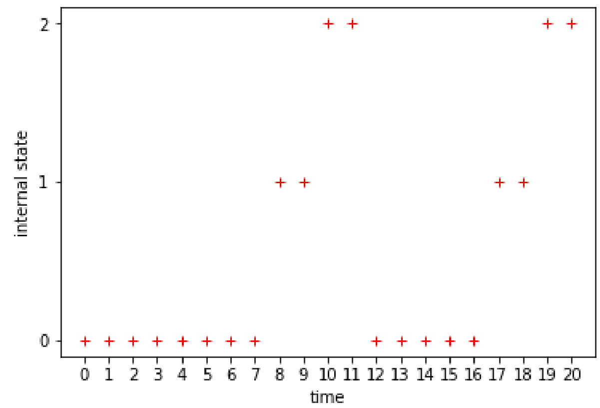

Fig. 8. Internal State vs Time - Task3

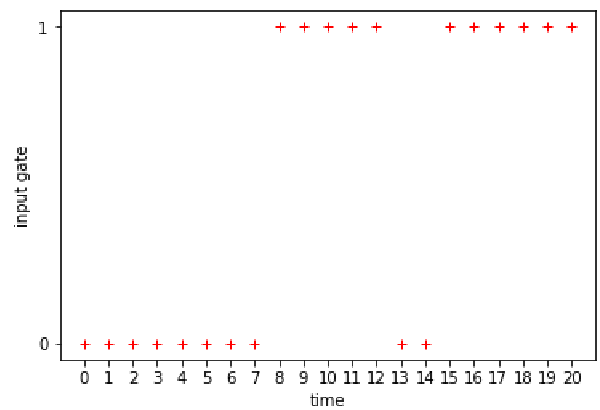

Fig. 9. Input Gate vs Time - Task3

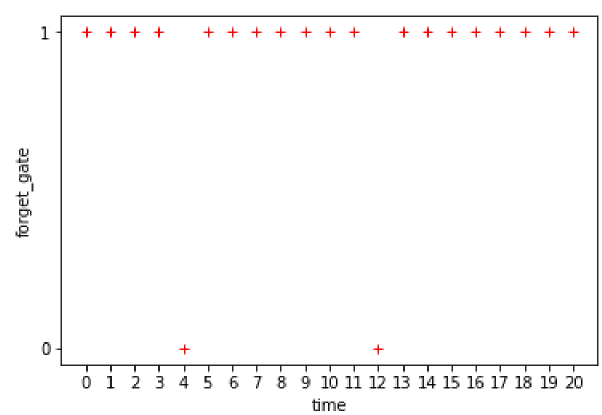

Fig. 10. Forget Gate vs Time - Task3.

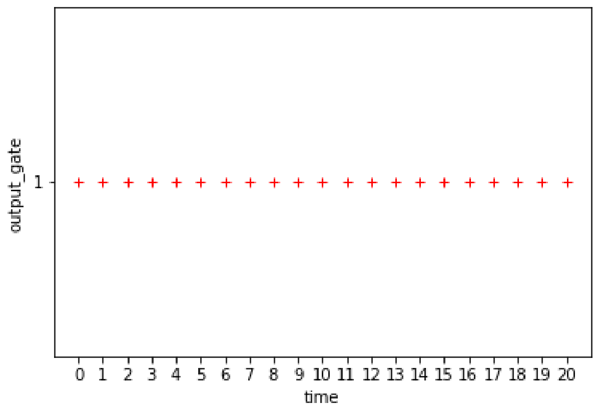

Fig. 11. Output Gate vs Time - Task3

III. Language Modeling using RNN with LSTM

We design a RNN with LSTM units for language modeling and implement the network using Keras with TensorFlow backend. We train two models, the Word-level model and the Character-level model. Using the two trained models we generate text.

A. Network Structure

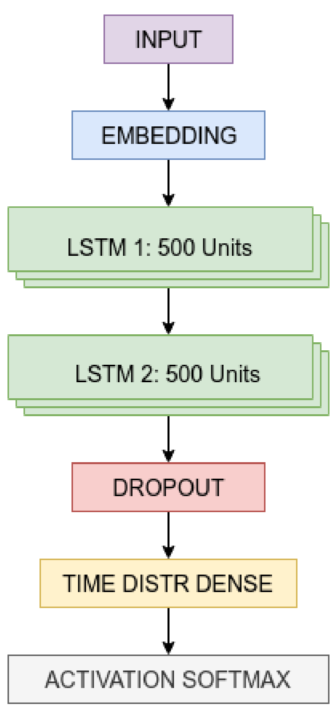

Fig. 12. RNN Structure

The network structure is as shown in Figure 12. The input shape of the text data is: (batch size, number of time steps, hidden size). For each batch sample and each word in the number of time steps, there is a 500 length embedding word vector to represent the input word. These embedding vectors will be learned as part of the overall model learning. The input data is then fed into two stacked layers of LSTM cells (of hidden layer size 500). The output from the LSTM cells is (batch size, number of time steps, hidden size). The output data is then passed to a Keras layer called TimeDistributed. The output layer has a softmax activation applied to it.

B. Hyperparameters

The hyperparameters chosen for this network are:

- Batch Size is 20
- Number of time steps is 30
- Hidden Layer Size is 500
- Adam Optimizer with learning rate 0.001, β_1 0.9 and β_2 0.999
- Dropout with rate 0.5
- Number of epochs is 30

C. Perplexity

1) Word Level Model: The plot for perplexity for training and validation sets for the Word Level model is shown in Figure 13

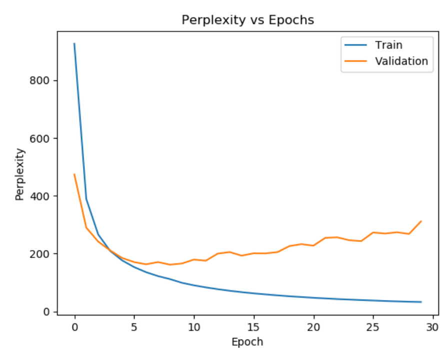

Fig. 13. Perplexity Word

2) Character Level Model: The plot for perplexity for training and validation sets for the Character Level model is shown in Figure 14

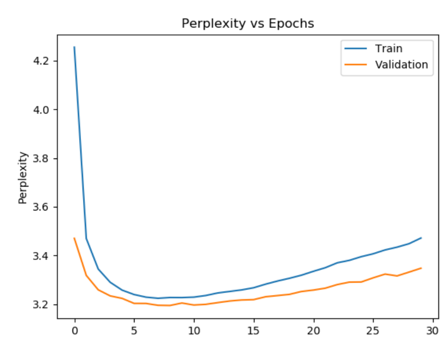

Fig. 14. Perplexity Character

D. Text generated by Word Level Model

Test data: Actual: futures the N stock specialist firms on the big board floor the buyers and sellers of last resort who were criticized after the N crash once again could n't handle the selling pressure big investment banks refused to step up to the plate to support the beleaguered floor traders by buying big blocks of stock traders say heavy selling of standard poor 's 500-stock index futures in chicago beat stocks downward seven big board stocks ual amr bankamerica walt disney capital cities\/abc philip morris and pacific telesis group stopped trading and never resumed the has already begun the equity market was once again the specialists were not able to handle the imbalances on the floor of the new york stock exchange said christopher senior vice president at securities corp james chairman of specialists henderson

Predicted: the contracts the nasdaq N market program will the other board 's of company said of week is had by the crash the has be resist the proposal board bankers to on the to be a of the the bonds of stock and said the trading in stocks poor 's index lost fell the 's the off in in the trading currencies issues closed shares and fell rose and bell said on the nasdaq been been to the fed market of quoted up the the the have yet to cope the the stock traders the u.s. york stock exchange mr. chairman vice president of shearson international inc the and of

E. Text generated by Character Level Model

Training data:

Actual: n hydro-quebec ipo kia memotec mlx nahb punts rake regatta rubens sim snack-food ssangyong swapo wachter pierre N years old will join the board as a nonexecutive director nov. N mr. is chairman of n.v. the dutch publishing group rudolph N years old and former chairman of consolidated gold fields plc was named a nonexecutive director of this british industrial conglomerate a form of asbestos once used to make kent cigarette filters has caused a high percentage of cancer deaths among a group of workers exposed to it more than N years ago researchers reported the asbestos fiber is unusually once it enters the with even brief exposures to it causing symptoms that show up decades later researchers said inc. the unit of new york-based corp. that makes kent cigarettes stopped using in its cigarette filters in N although preliminary findings were reported more than a year ago the latest results appear in today 's new england jou

Predicted: n aa .ewcuiterosnantenl<arbri ttaa aet ocrrt aete oepure ntellrteaanpooarksiord aaiida cair rahshe r trccro tunk>a Nears afd aitl buintthe canrd on a sete etive oesector oete N tr. <unk>sn toairman of tunk>aej. ahe cetch orrlicheng troup telewlh aunk>a Nears afd and tor er coairman of tontulidated teld tonld ara ahs a'med a <ete cutive oesector of thes taotish an ustryal aomtreseral tnsirmeof t estor af e tned to take teet <otarynte aone rs aav bolsed t sagheprrientage of tolaer aeclh anong t sooup of thrk rs axperud th tnsiare than N Nears ano aesearchers aeported the cnsastic aoner tunk>an tndsual y aunk>af e tn wxdere toe cunk>ahth txen teitf txperure ah tnsionseng tospt r ahet toow tp tecldes taser testarch rs aaid tunk>an . ahe c.it of tew york based ounk>aorp. ahet taner aeeniaotarente taac ed tneng tunk¿ an tts cunk>aotatent aone rs an t t lhough trosiminary sona ng aire aecorted tore than N sear ego aoe catest teselts anprar tn th ay ts sew ynviand toh

Test data:

Actual: new york stock exchange did n't fall apart friday as the dow jones industrial average plunged N points most of it in the final hour it barely managed to stay this side of chaos<eos>some circuit breakers installed after the october N crash failed their first test traders say unable to cool the selling panic in both stocks and futures <eos>the N stock specialist firms on the big board floor the buyers and sellers of last resort who were criticized after the N crash once again could n't handle the selling pressure<eos >big investment banks refused to step up to the plate to support the beleaguered floor traders by buying big blocks of stock traders say<eos >heavy selling of standard poor 's 500-stock index futures in chicago <unk>beat stocks downward<eos>seven big board stocks ual amr bankamerica walt disney capital cities\/abc philip morris and pacific telesis group stopped trading and never resumed<eos>the <unk>has already begun<eos>the equity market was <unk><eos>once again the specialists were not able to handle t

Predicted: cew york stock exchange ced n't bocl tnprtmtooday 'n ahe celnjones andustrial average aranged N Noints aart of ttsws the siran aaus tnswenr y tanagem ao teat ahes ytme of toais tome aotcurt oeoakers an tinlad t ter the cnt.ber N Neis aolled toa r cirst thrt ahadeds aai tndble to tomp ahe ctcl ng arrac an tanh ceack and totures the c Ntack aaocial stsaonm af the cag board aoowr aoe casers and taclir af tott yeseltstie ware aoemicaae t ter the c Neash af e tneinstomnd b't bavd esthe ctcl ng trosi re tul bntestoent tenk aeplned to teap tp th the crane th tepport the caliat e toorr thaders ae tesing aal boocks af teocksmradi s aaiste di oacling tf taotdard a coor 's s00-stack endex arnures an toaeago tunk>aect aaacks ae n atd toper tog board aaacks anl cne senk m iaa ahslesescey 'olital aotics /anoharalip iareis and <rrific ahlepts aooup aaac ed thading and tewer aeseled the cunk¿ aas b leady aecan the cxuipy oarket aas aunk>tnee tnainstha ctocial zt aire a't t oe to bavd e t

References

[1] Lipton, Zachary C., John Berkowitz, and Charles Elkan. "A critical review of recurrent neural networks for sequence learning." arXiv preprint arXiv:1506.00019 (2015).

[2] Karpathy, Andrej, Justin Johnson, and Li Fei-Fei. "Visualizing and understanding recurrent networks." arXiv preprint arXiv:1506.02078 (2015).

[3] Colah's Blog Understanding LSTM Networks http://colah.github.io/posts/2015-08-Understanding-LSTMs

[4] https://cs224d.stanford.edu/lectures/CS224d-Lecture8.pdf