

COMP5046
Natural Language Processing

Lecture 3: Word Classification and Machine Learning

Dr. Caren Han
Semester 1, 2021
School of Computer Science,
University of Sydney

1

0 LECTURE PLAN

Lecture 3: Word Classification and Machine Learning

1. Previous Lecture: Word Embedding Review
2. Word Embedding Evaluation
3. Deep Neural Network for Natural Language Processing
 1. Perceptron and Neural Network (NN)
 2. Multilayer Perceptron
 3. Applications
4. Next Week Preview
See how the Deep Learning can be used for NLP
 - Text Classification, etc.

2

1 Previous Lecture Review

Word2Vec Models

CBOW

Predict center word from (bag of) context words

Skip-gram

Predict context words given center word

3

1 Previous Lecture Review

Word2Vec with Continuous Bag of Words (CBOW)

Predict center word from (bag of) context words

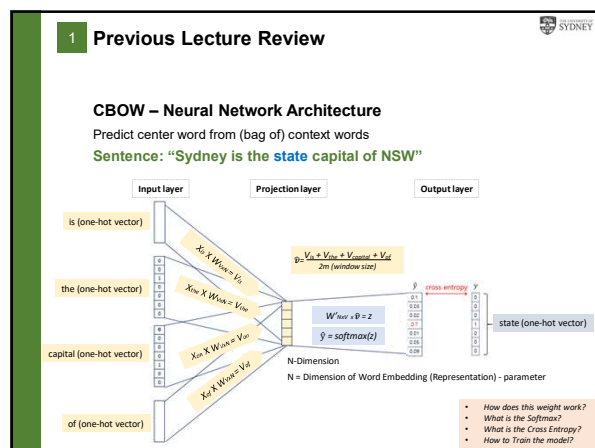
Sentence: "Sydney is the state capital of NSW"

Using window slicing, develop the training data

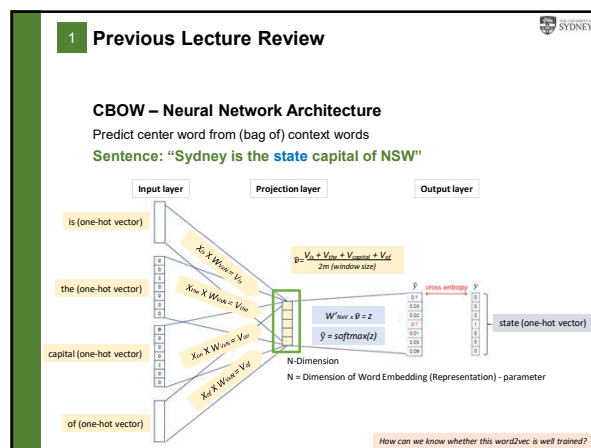
Center word	Context ("outside") word
[1,0,0,0,0,0,0]	[0,1,0,0,0,0,0], [0,0,1,0,0,0,0]
[0,1,0,0,0,0,0]	[1,0,0,0,0,0,0], [0,0,1,0,0,0,0]
[0,0,1,0,0,0,0]	[1,0,0,0,0,0,0], [0,1,0,0,0,0,0], [0,0,0,1,0,0,0]
[0,0,0,1,0,0,0]	[0,1,0,0,0,0,0], [0,0,1,0,0,0,0], [0,0,0,0,1,0,0], [0,0,0,0,0,1,0]
[0,0,0,0,1,0,0]	[0,0,1,0,0,0,0], [0,0,0,1,0,0,0], [0,0,0,0,0,1,0]
[0,0,0,0,0,1,0]	[0,0,0,1,0,0,0], [0,0,0,0,1,0,0], [0,0,0,0,0,0,1]
[0,0,0,0,0,0,1]	[0,0,0,0,1,0,0], [0,0,0,0,0,1,0]

Center word
Context ("outside") word

4



5



6

0 LECTURE PLAN

Lecture 3: Word Classification and Machine Learning

1. Previous Lecture: Word Embedding Review
2. **Word Embedding Evaluation**
3. Deep Neural Network for Natural Language Processing
 1. Perceptron and Neural Network (NN)
 2. Multilayer Perceptron
 3. Applications
4. Next Week Preview
See how the Deep Learning can be used for NLP
- Text Classification, etc.

7

2 Word Embedding Evaluation

How to evaluate word vectors?

Type	How to work / Benefit
Intrinsic	Evaluation on a specific/intermediate subtask <ul style="list-style-type: none"> Fast to compute Helps to understand that system Not clear if really helpful unless correlation to real task is established
Extrinsic	Evaluation on a real task <ul style="list-style-type: none"> Can take a long time to compute accuracy Unclear if the subsystem is the problem or its interaction or other subsystems

Male-Female: king, queen, male, female

Verb tense: wait, waited, waiting, waitings

Country-Capital: Berlin, Paris, London, Tokyo, Sydney, Moscow, Beijing, New York, Washington, Ottawa, Wellington, Nairobi, Lima, Lagos, Nairobi, Lima, Lagos

8

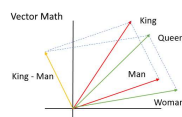
2 Word Embedding Evaluation

Intrinsic word vector evaluation

Word Vector Analogies

$a \leftrightarrow b :: c \leftrightarrow ???$
 $man \leftrightarrow women :: king \leftrightarrow ???$

- Evaluate word vectors by how well their cosine distance after addition captures intuitive semantic and syntactic analogy questions



9

2 Word Embedding Evaluation

Intrinsic word vector evaluation

Word Vector Analogies

King - Man + Woman = ?

No	Training Dataset	Type	Result
1	TED Script	word2vec CBOW	President
2		word2vec Skip-gram	Luther
3		fastText CBOW	Kidding
4		fastText Skip-gram	Jarring
5	Google News	word2vec CBOW	queen
6		word2vec Skip-gram	queen

10

2 Word Embedding Evaluation

Intrinsic word vector evaluation

Evaluation Result Comparison

The Semantic-Syntactic word relationship tests for understanding of a wide variety of relationships as shown below.

Using 640-dimensional word vectors, a skip-gram trained model achieved 55% semantic accuracy and 59% syntactic accuracy.

Table 3: Comparison of architectures using models trained on the same data, with 640-dimensional word vectors. The accuracies are reported on our Semantic-Syntactic Word Relationship test set, and on the syntactic relationship test set of [20].

Model Architecture	Semantic-Syntactic Word Relationship test set		MSR Word Relatedness Test Set [20]
	Semantic Accuracy [%]	Syntactic Accuracy [%]	
RNNLM	9	36	35
NNLM	23	53	47
CBOW	24	64	61
Skip-gram	55	59	56

(Original Word2vec Paper - Mikolov et al. 2013)

11

2 Word Embedding Evaluation

Intrinsic word vector evaluation

Evaluation Result Comparison

The Semantic-Syntactic word relationship tests for understanding of a wide variety of relationships as shown below.

Table 2: Results on the word analogy task, given as percent accuracy. Underlined scores are best within groups of similarly-sized models; bold scores are best overall. HPCA vectors are publicly available¹; (v)LBL results are from (Mnih et al., 2013); skip-gram (SG) and CBOW results are from (Mikolov et al., 2013a,b); we trained SG² and CBOW³ using the word2vec tool³. See text for details and a description of the SVD models.

Model	Dim	Size	Sem.	Syn.	Rec.
vLBL	100	1.5B	55.9	50.1	53.2
HPCA	100	1.6B	4.2	16.4	10.8
GloVe	100	1.6B	<u>67.5</u>	<u>54.3</u>	<u>60.3</u>
SG	300	1B	61	61	61
CBOW	300	1.6B	16.1	52.6	36.1
vLBL	300	1.5B	54.2	<u>64.8</u>	60.0
vLBL	300	1.5B	65.2	63.0	64.0
GloVe	300	1.6B	<u>80.8</u>	61.5	70.3
SVD	300	6B	6.3	8.1	7.3
SVD-S	300	6B	36.7	46.6	42.1
SVD-L	300	6B	56.6	63.0	60.1
CBOW ³	300	6B	63.6	62.4	65.7
SG ²	300	6B	73.0	66.0	69.1
GloVe	300	6B	77.4	67.0	71.7
CBOW	1000	6B	57.3	68.9	63.7
SG	1000	6B	66.1	65.1	65.6
SVD-L	300	42B	38.4	58.2	49.2
GloVe	300	42B	<u>81.9</u>	<u>69.3</u>	<u>75.8</u>

(Original Glove Paper - Pennington et al. 2014)

12

2 Word Embedding Evaluation

Intrinsic word vector evaluation

Evaluation Result Comparison

The Semantic-Syntactic word relationship tests for understanding of a wide variety of relationships as shown below.

Window-Size (m) and Vector Dimension (N)

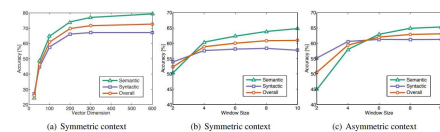


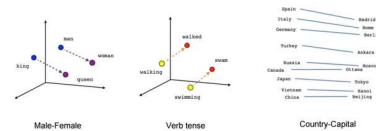
Figure 2: Accuracy on the analogy task as function of vector size and window size/type. All models are trained on the 6 billion token corpus. In (a), the window size is 10. In (b) and (c), the vector size is 100. (Original GloVe Paper - Pennington et al. 2014)

13

2 Word Embedding Evaluation

How to evaluate word vectors?

Type	How to work / Benefit
Intrinsic	Evaluation on a specific/intermediate subtask <ul style="list-style-type: none"> Fast to compute Helps to understand that system Not clear if really helpful unless correlation to real task is established
Extrinsic	Evaluation on a real task <ul style="list-style-type: none"> Can take a long time to compute accuracy Unclear if the subsystem is the problem or its interaction or other subsystems



14

0 LECTURE PLAN

Lecture 3: Word Classification and Machine Learning

1. Previous Lecture: Word Embedding Review
2. Word Embedding Evaluation
3. Deep Neural Network for Natural Language Processing
 1. Perceptron and Neural Network (NN)
 2. Multilayer Perceptron
 3. Applications
4. Next Week Preview

See how the Deep Learning can be used for NLP

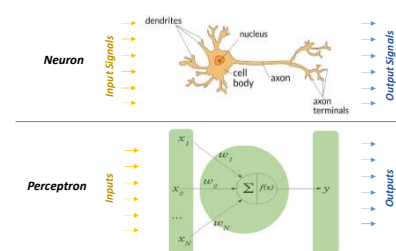
 - Text Classification, etc.

15

3 Deep Learning for NLP

Deep Learning with Neural Network


Neuron and Perceptron



16

3

Deep Learning for NLP

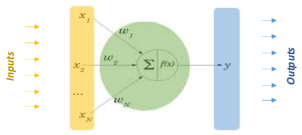


Deep Learning with Neural Network

Inputs and Outputs (Labels) for Natural Language Processing

x_i	Inputs	Features
y_i	Outputs (labels)	What we try to predict/classify


Perceptron



17

3

Deep Learning for NLP

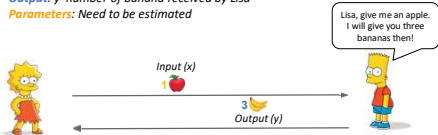


Deep Learning with Neural Network

Input: x =number of apple given by Lisa

Output: y =number of banana received by Lisa


Parameters: Need to be estimated



18

3

Deep Learning for NLP

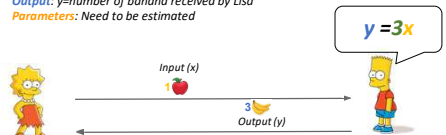


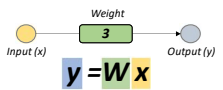
Deep Learning with Neural Network - Model

Input: x =number of apple given by Lisa

Output: y =number of banana received by Lisa

Parameters: Need to be estimated







19

3

Deep Learning for NLP





20

3 Deep Learning for NLP

Deep Learning with Neural Network - Model

Input: x =number of apple given by Lisa
 Output: y =number of banana received by Lisa
 Parameters: Need to be estimated

21

3 Deep Learning for NLP

Deep Learning with Neural Network - Parameter

Input: x =number of apple given by Lisa
 Output: y =number of banana received by Lisa
 Parameters: Need to be estimated

$y = Wx$
 What is W then?

22

3 Deep Learning for NLP

Deep Learning with Neural Network - Parameter

Input: x =number of apple given by Lisa
 Output: y =number of banana received by Lisa
 Parameters: Need to be estimated

$y = Wx$
 What is W then?

23

3 Deep Learning for NLP

Deep Learning with Neural Network - Parameter

Input: x =number of apple given by Lisa
 Output: y =number of banana received by Lisa
 Parameters: Need to be estimated

x	y
1	0
5	16
6	20

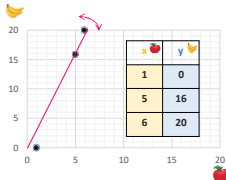
$y = Wx$
 What is W then?

24

3 Deep Learning for NLP

Deep Learning with Neural Network - Parameter

Input: x =number of apple given by Lisa
 Output: y =number of banana received by Lisa
 Parameters: Need to be estimated



x	y
1	0
5	16
6	20

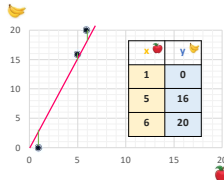
$y = Wx$
 What is W then?

25

3 Deep Learning for NLP

Deep Learning with Neural Network - Parameter

Input: x =number of apple given by Lisa
 Output: y =number of banana received by Lisa
 Parameters: Need to be estimated



x	y
1	0
5	16
6	20

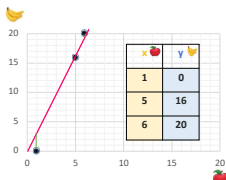
$y = Wx$
 What if W is 3?
 $3 = 3 \times 1$
 $15 = 3 \times 5$
 $20 = 3 \times 6$

26

3 Deep Learning for NLP

Deep Learning with Neural Network - Parameter

Input: x =number of apple given by Lisa
 Output: y =number of banana received by Lisa
 Parameters: Need to be estimated



x	y
1	0
5	16
6	20

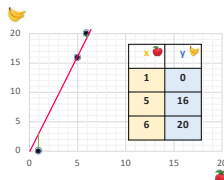
$y = Wx$
 What if W is 3.2?
 $3.2 = 3.2 \times 1$
 $16 = 3.2 \times 5$
 $19.2 = 3.2 \times 6$

27

3 Deep Learning for NLP

Deep Learning with Neural Network - Parameter

Input: x =number of apple given by Lisa
 Output: y =number of banana received by Lisa
 Parameters: Need to be estimated



x	y
1	0
5	16
6	20

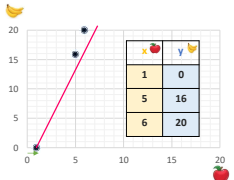
$y = Wx + b$
 Weight is not enough...

28

3 Deep Learning for NLP

Deep Learning with Neural Network - Parameter

Input: x =number of apple given by Lisa
Output: y =number of banana received by Lisa
Parameters: Need to be estimated



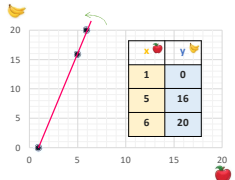
weight bias
 $y = Wx + b$
How can we find the parameters, w and b ?

29

3 Deep Learning for NLP

Deep Learning with Neural Network - Parameter

Input: x =number of apple given by Lisa
Output: y =number of banana received by Lisa
Parameters: Need to be estimated



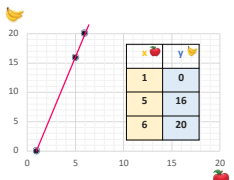
weight bias
 $y = Wx + b$
How can we find the parameters, w and b ?

30

3 Deep Learning for NLP

Deep Learning with Neural Network - Parameter

Input: x =number of apple given by Lisa
Output: y =number of banana received by Lisa
Parameters: Need to be estimated



weight bias
Model $y = Wx + b$
How can we find the parameters, w and b ?

31

3 Deep Learning for NLP

Deep Learning with Neural Network - Cost

Actual Data

	weight	bias
$y = ? x + ?$		
1	0	
5	16	
6	20	

Model Ex#1

	weight	bias
$\hat{y} = 1 x + 0$		
predicted	1	0
actual	5	16
	6	20

Model Ex#2

	weight	bias
$\hat{y} = 2 x + 2$		
predicted	4	0
actual	5	16
	6	20

Input (x) → Weight (?) → Bias (?) → Output (y)

Which one is closer?

32

3 Deep Learning for NLP

Deep Learning with Neural Network – Cost (loss)

Actual Data

x	y
1	0
5	16
6	20

Model Ex#1

x	\hat{y}	actual	cost
1	1	0	$(1-0)^2$
5	5	16	
6	6	20	

Model Ex#2

x	\hat{y}	actual	cost
1	4	0	
5	12	16	
6	14	20	

Let's calculate the cost(loss)!

Mean Squared Error (MSE)

$$C(w,b) = \sum_{n \in [0,1,2]} (y_n - \hat{y}_n)^2$$

Input (x) → Weight → Bias → Output (y)

33

3 Deep Learning for NLP

WAIT! Loss Function? Cost Calculation?

Input (x) → Weight → Bias → Loss Function → Predicted (\hat{y}) → Output (y)

1) Mean Squared Error (MSE): measures the average of the squares of the errors

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

2) Cross Entropy: calculating the difference between two probability distributions

$$L_{\text{cross-entropy}}(\hat{y}, y) = - \sum_i y_i \log(\hat{y}_i)$$

cross entropy

\hat{y}	y
0.1	0
0.03	0
0.02	0
0.7	1
0.01	0
0.05	0
0.08	0

34

3 Deep Learning for NLP

Deep Learning with Neural Network - Cost (loss)

Actual Data

x	y
1	0
5	16
6	20

Model Ex#1

x	\hat{y}	actual	cost
1	1	0	1
5	5	16	121
6	6	20	196

Model Ex#2

x	\hat{y}	actual	cost
1	4	0	16
5	12	16	16
6	14	20	36

$C(1,0) = 318$

$C(2,2) = 68$

Let's calculate the cost!

$$C(w,b) = \sum_{n \in [0,1,2]} (y_n - \hat{y}_n)^2$$

Input (x) → Weight → Bias → Output (y)

35

3 Deep Learning for NLP

Deep Learning with Neural Network - Cost (loss)

Actual Data

x	y
1	0
5	16
6	20

Model Ex#1

x	\hat{y}	actual	cost
1	1	0	1
5	5	16	121
6	6	20	196

Model Ex#2

x	\hat{y}	actual	cost
1	4	0	16
5	12	16	16
6	14	20	36

$C(1,0) = 318$

$C(2,2) = 68$

Let's calculate the costs and get the lowest one!

$\arg \min_{w,b \in [-\infty, \infty]} C(w,b)$

Input (x) → Weight → Bias → Output (y)

36

3 Deep Learning for NLP

Deep Learning with Neural Network - Optimizer

Actual Data

x	y
1	0
5	16
6	20

weight bias

$y = ? x + ?$

Input (x) Weight Bias Output (y)

weight bias

$y = wx + b$

The lowest!
 $w, b = 4, -4$
 $C(w, b) = 0$

Let's calculate the costs and get the lowest one!

$\arg \min C(w, b)$
 $w, b \in [-\infty, \infty]$

37

3 Deep Learning for NLP

Deep Learning with Neural Network - Optimizer

Backpropagation (weight update)

x	y
1	0
5	16
6	20

weight bias

$y = 4x - 4$

Input (x) Weight Bias Predicted (y) Output (y)

weight bias

$y = wx + b$

The lowest!
 $w, b = 4, -4$
 $C(w, b) = 0$

Backpropagation (weight update)

Loss Function

$\arg \min C(w, b)$
 $w, b \in [-\infty, \infty]$

38

3 Deep Learning for NLP

Deep Learning with Neural Network - Optimizer

Backpropagation (weight update)

x	y
1	0
5	16
6	20

weight bias

$y = 4x - 4$

Input (x) Weight Bias Predicted (y) Output (y)

weight bias

$y = wx + b$

The lowest!
 $w, b = 4, -4$
 $C(w, b) = 0$

Oh, Wait...
 Do we need to calculate all cost for all options of W and B?

Expensive to compute (hours or days)

$\arg \min C(w, b)$
 $w, b \in [-\infty, \infty]$

39

3 Deep Learning for NLP

Finding the Optimal weight and bias – Gradient Descent

Cost (Error)

weight

Gradient = slope

There are different types of Gradient descent optimization algorithms:
 Batch Gradient Descent, Stochastic Gradient Descent, Momentum, Adam, etc.

40

3 Deep Learning for NLP

Choose the optimal Learning Rate!

Learning Rate: a hyperparameter that controls how much to change the model in response to the estimated error each time the model weights are updated.

new_weight = existing_weight - learning_rate * gradient
 new_weight = existing_weight - learning_rate * (current_output - desired output) * gradient(current output) * existing_input

41

3 Deep Learning for NLP

Finding the Optimal weight and bias – Gradient Descent

There are different types of Gradient descent optimization algorithms:
 Batch Gradient Descent, Stochastic Gradient Descent, Momentum, Adam, etc.

42

3 Deep Learning for NLP

Stochastic Gradient Descent

The cost would be very expensive if we calculate it for all windows in the corpus!
 You would wait a very long time before making a single update!

The Solution can be used different Gradient Descent Method.
 The most common – “Stochastic Gradient Descent (SGD)”

Vanilla (Batch) gradient descent performs redundant computations for large datasets, as it recomputes gradients for similar examples before each parameter update. SGD does away with this redundancy by performing one update at a time. It is therefore usually much faster and can also be used to learn online.

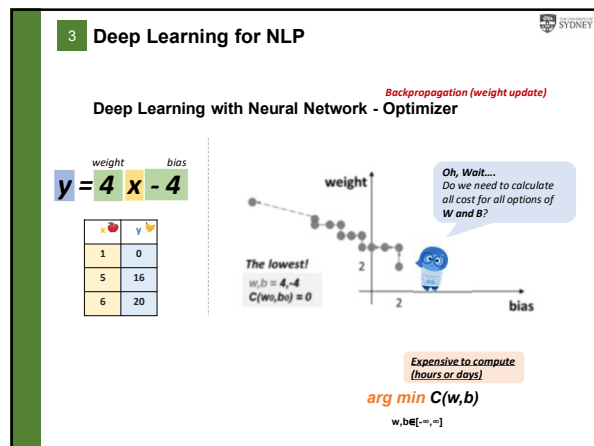
43

3 Deep Learning for NLP

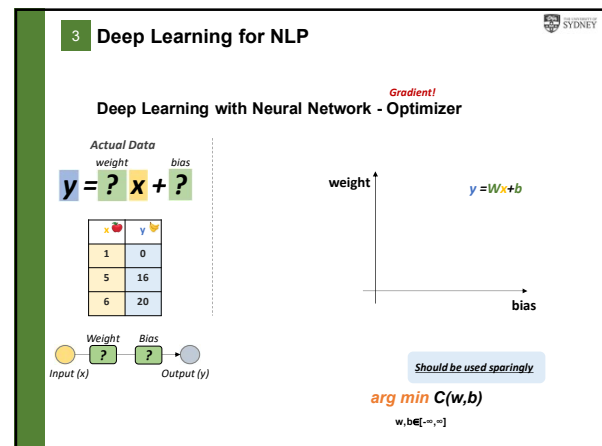
Finding the Optimal weight and bias – Gradient Descent

There are different types of Gradient descent optimization algorithms:
 Batch Gradient Descent, Stochastic Gradient Descent, Momentum, Adam, etc.

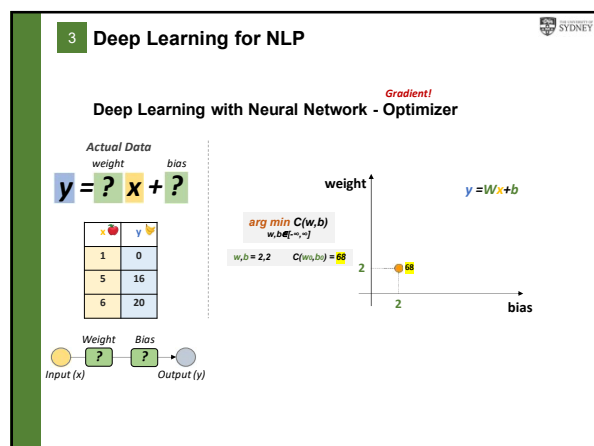
44



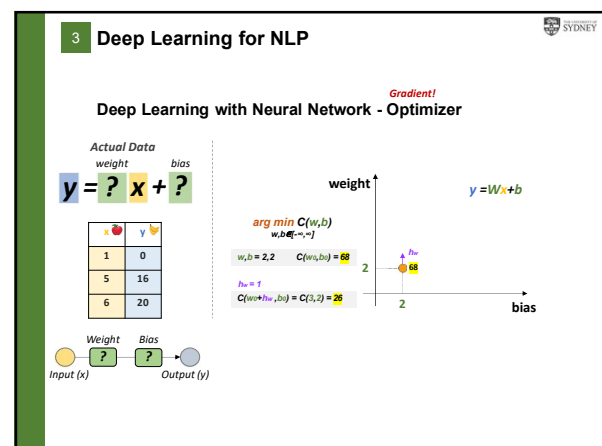
45



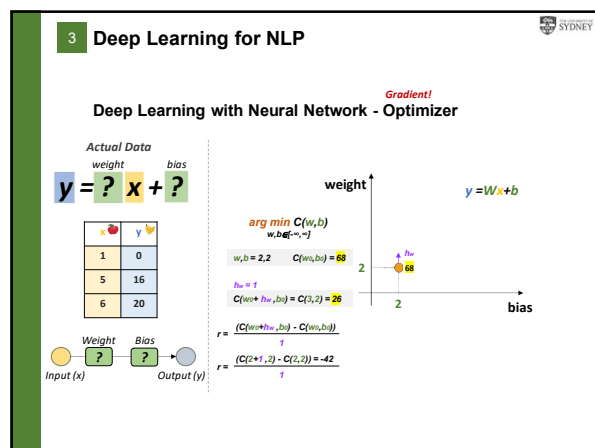
46



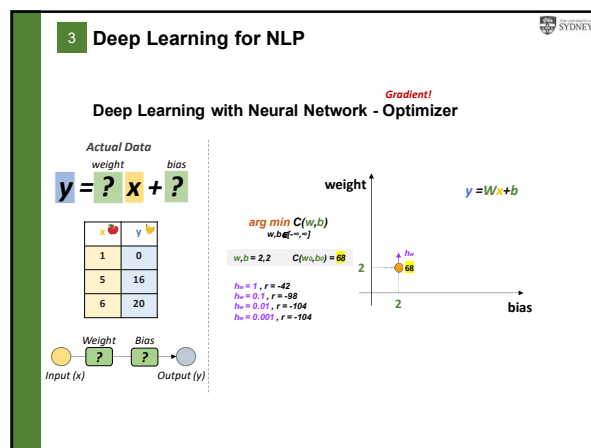
47



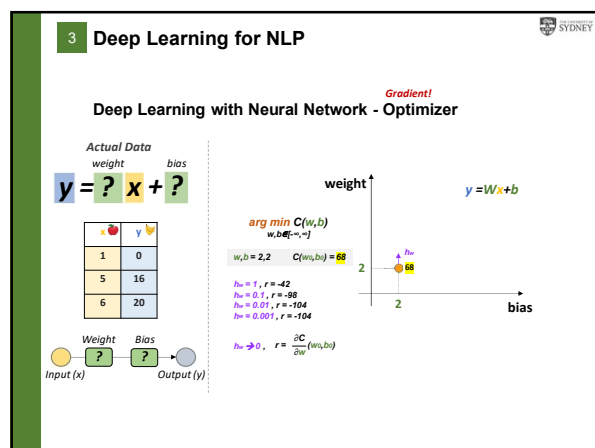
48



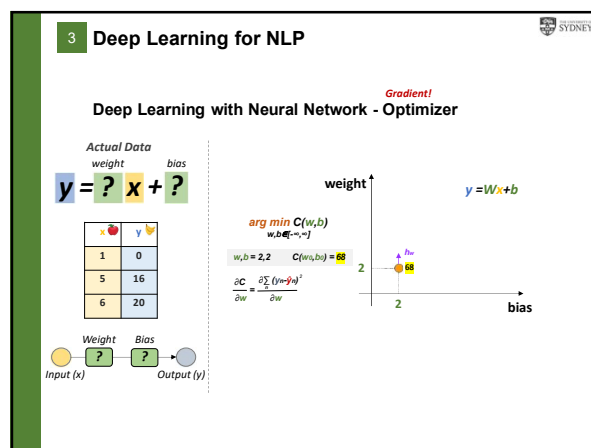
49



50



51



52

3 Deep Learning for NLP

Deep Learning with Neural Network - Optimizer Gradient!

Actual Data

weight	bias
x	y
1	0
5	16
6	20

$y = ? x + ?$

Weight Bias

Input (x) Output (y)

weight

bias

$y = Wx + b$

$\arg \min_{w,b \in \mathbb{R}^n} C(w,b)$

$w, b = 2, 2 \quad C(w,b) = 68$

$\frac{\partial C}{\partial w} = \frac{\partial}{\partial w} \sum_{i=1}^n (y_i - \hat{y}_i)^2 = \sum_{i=1}^n 2(y_i - \hat{y}_i) \cdot x_i$

$\frac{\partial C}{\partial w} = 2$

$h_w \rightarrow 0, r = \frac{\partial C}{\partial w}(w,b) = 104$

53

3 Deep Learning for NLP

Deep Learning with Neural Network - Optimizer Gradient!

Actual Data

weight	bias
x	y
1	0
5	16
6	20

$y = ? x + ?$

Weight Bias

Input (x) Output (y)

weight

bias

$y = Wx + b$

$\arg \min_{w,b \in \mathbb{R}^n} C(w,b)$

$w, b = 2, 2 \quad C(w,b) = 68$

$\frac{\partial C}{\partial w} = \frac{\partial}{\partial w} \sum_{i=1}^n (y_i - \hat{y}_i)^2 = \sum_{i=1}^n 2(y_i - \hat{y}_i) \cdot x_i$

$\frac{\partial C}{\partial w} = 2$

$h_w \rightarrow 0, r = \frac{\partial C}{\partial w}(w,b) = 104$

54

3 Deep Learning for NLP

Deep Learning with Neural Network - Optimizer Gradient!

Actual Data

weight	bias
x	y
1	0
5	16
6	20

$y = ? x + ?$

Weight Bias

Input (x) Output (y)

weight

bias

$y = 2x + 2$

$\arg \min_{w,b \in \mathbb{R}^n} C(w,b)$

$w, b = 2, 2 \quad C(w,b) = 68$

$\frac{\partial C}{\partial w} = \frac{\partial}{\partial w} \sum_{i=1}^n (y_i - \hat{y}_i)^2 = \sum_{i=1}^n 2(y_i - \hat{y}_i) \cdot x_i$

$\frac{\partial C}{\partial w} = 2$

$h_w \rightarrow 0, r = \frac{\partial C}{\partial w}(w,b) = 104$

predicted	actual	$(y - \hat{y})$	$2(y - \hat{y}) \cdot x$
1	4	0	-4
5	12	16	40
6	14	20	72

55

3 Deep Learning for NLP

Deep Learning with Neural Network - Optimizer Gradient!

Actual Data

weight	bias
x	y
1	0
5	16
6	20

$y = ? x + ?$

Weight Bias

Input (x) Output (y)

weight

bias

$y = Wx + b$

$\arg \min_{w,b \in \mathbb{R}^n} C(w,b)$

$w, b = 2, 2 \quad C(w,b) = 68$

$\frac{\partial C}{\partial w} = \frac{\partial}{\partial w} \sum_{i=1}^n (y_i - \hat{y}_i)^2 = \sum_{i=1}^n 2(y_i - \hat{y}_i) \cdot x_i$

$\frac{\partial C}{\partial w} = 2$

$h_w \rightarrow 0, r = \frac{\partial C}{\partial w}(w,b) = 104$

56

3 Deep Learning for NLP

Deep Learning with Neural Network - Optimizer

Gradient!

Actual Data

weight	bias
$y = ? x + ?$	
1	0
5	16
6	20

Weight Bias
Input (x) Output (y)

Model

$y = 2x + 2$

Cost

$C(w,b) = \sum (y_i - \hat{y}_i)^2$

Optimizer

$\arg \min C(w,b)$

$w, b \in \mathbb{R}^n$

$w, b = 2, 2 \quad C(w,b) = 68$

$\frac{\partial C}{\partial w} = 104$

$\frac{\partial C}{\partial b} = 12$

predicted	actual	$(y - \hat{y})$	$2(y - \hat{y})$
1	4	0	-4
5	12	16	8
6	14	20	12

57

3 Deep Learning for NLP

Deep Learning with Neural Network - Optimizer

Gradient!

Actual Data

weight	bias
$y = ? x + ?$	
1	0
5	16
6	20

Weight Bias
Input (x) Output (y)

Model

$y = 2x + 2$

Cost

$C(w,b) = \sum (y_i - \hat{y}_i)^2$

Optimizer

$\arg \min C(w,b)$

$w, b \in \mathbb{R}^n$

$w, b = 2, 2 \quad C(w,b) = 68$

$\frac{\partial C}{\partial w} = 104$

$\frac{\partial C}{\partial b} = 12$

$y = Wx + b$

weight

bias

Graphs showing cost function and its gradient.

58

3 Deep Learning for NLP

Deep Learning with Neural Network

Data

weight	bias
$y = ? x + ?$	
1	0
5	16
6	20

Model

$y = 4x - 4$

Cost

$C(w,b) = \sum (y_i - \hat{y}_i)^2$

Optimizer

$\arg \min C(w,b)$

$w, b \in \mathbb{R}^n$

$w, b = 4, -4$

System

59

3 Deep Learning for NLP

Image showing a person sitting at a desk, possibly a student or researcher, in a classroom or office setting.

60

3 Deep Learning for NLP

Deep Learning with Neural Network

Input: x =number of apple given by Lisa
 Output: y =number of banana received by Lisa
 Parameters: Need to be estimated

$y = 4x - 4$

61

3 Deep Learning for NLP

Deep Learning with Neural Network

$y = w_1x_1 + w_2x_2 + w_3x_3 + w_4x_4 + \dots + w_nx_n + b$

Data

Millions of Parameters
 Millions of Samples

62

3 Deep Learning for NLP

Deep Learning with Neural Network

$y = w_1x_1 + w_2x_2 + w_3x_3 + w_4x_4 + \dots + w_nx_n + b$

Data

Millions of Parameters
 Millions of Samples

63

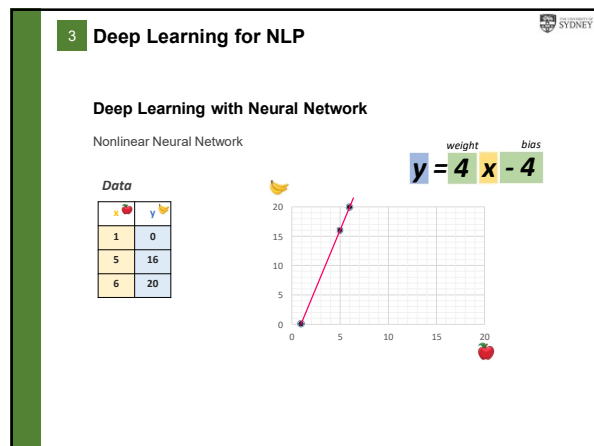
3 Deep Learning for NLP

Deep Learning with Neural Network

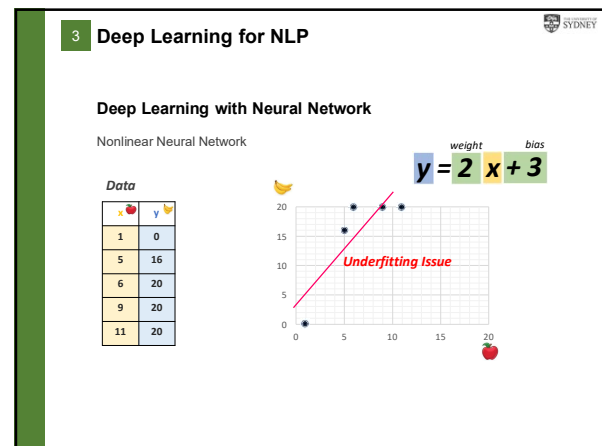
Input: x =number of apple given by Lisa
 Output: y =number of banana received by Lisa
 Parameters: Need to be estimated

There is a limit of bananas I can give you

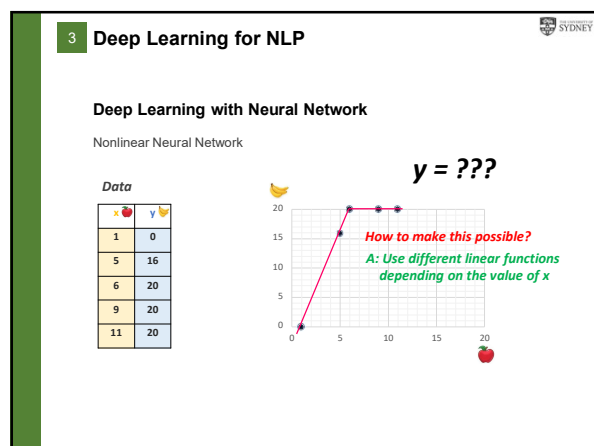
64



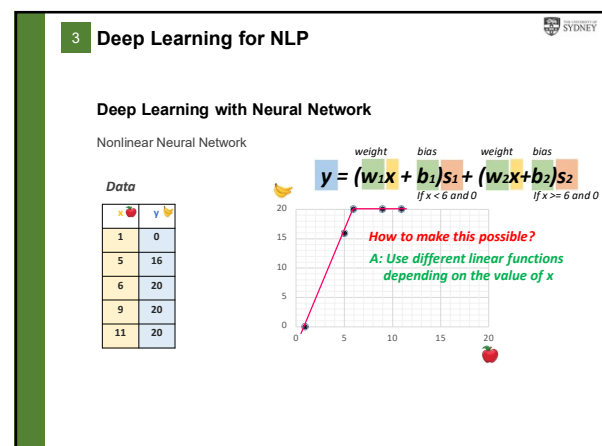
65



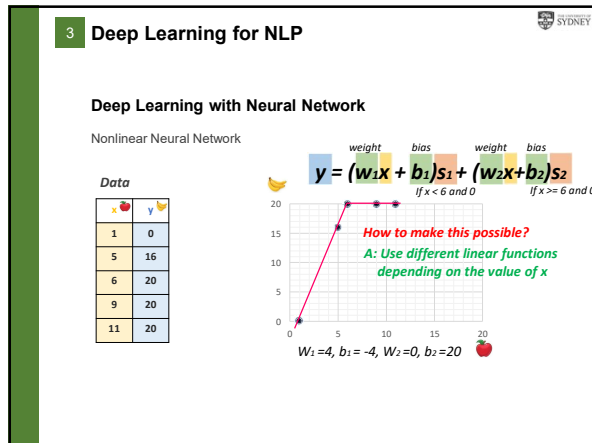
66



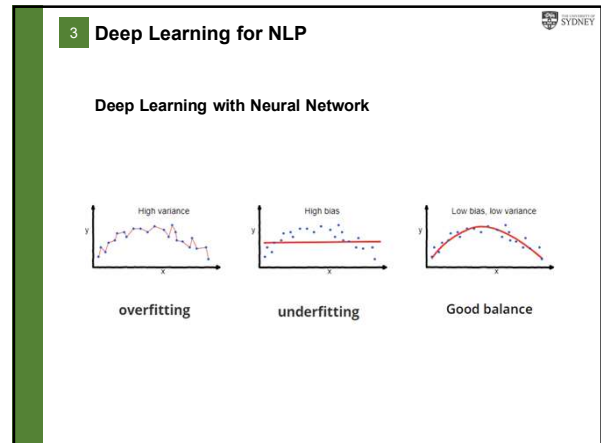
67



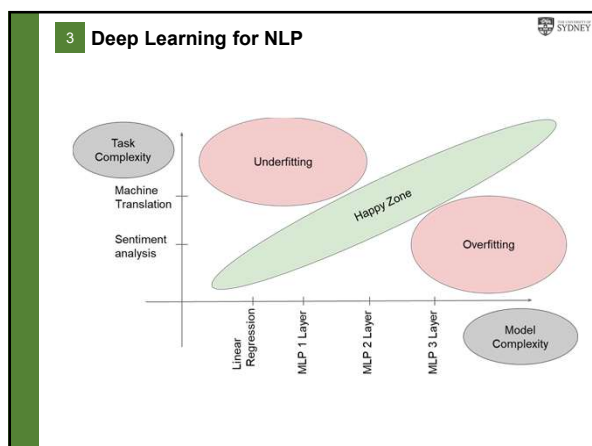
68



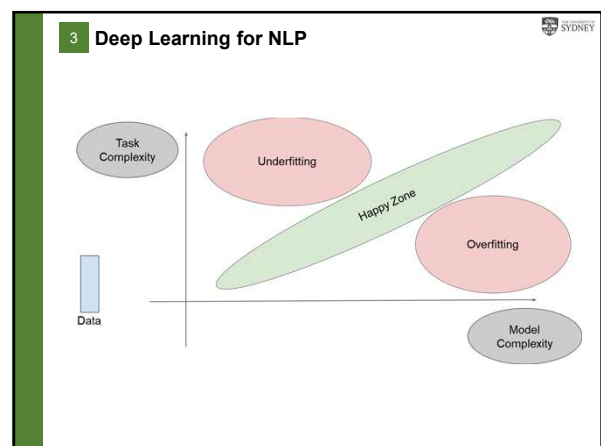
69



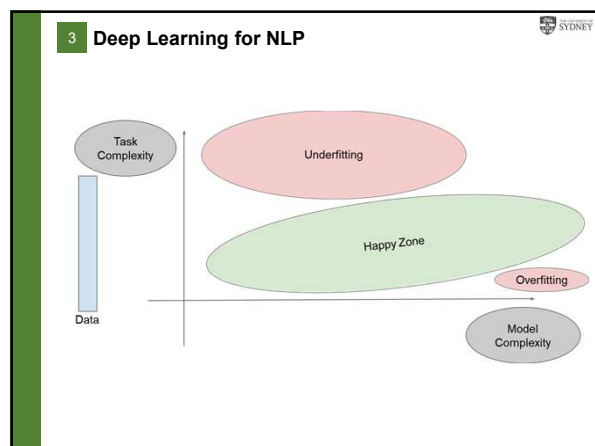
70



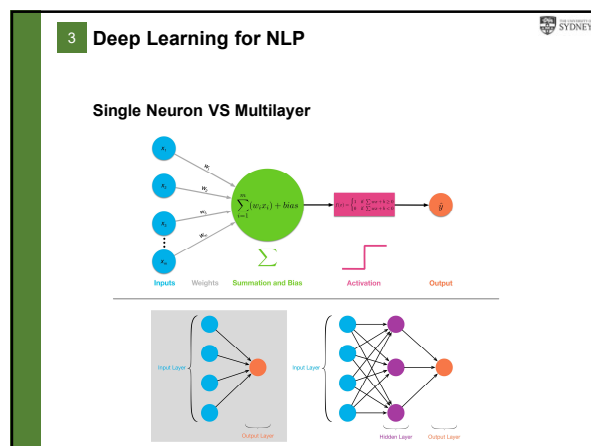
71



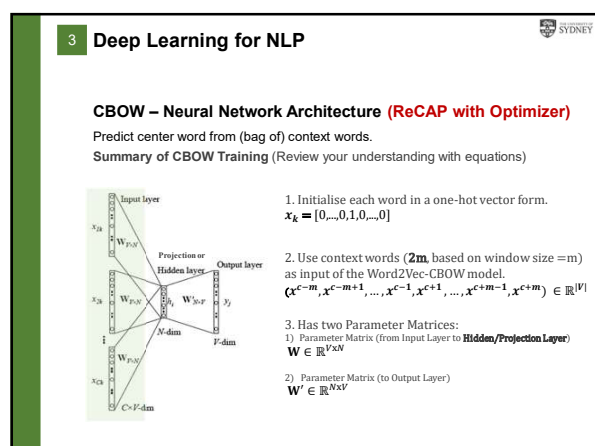
72



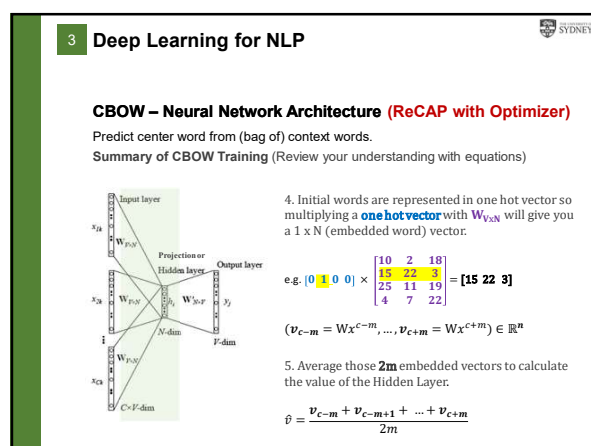
73



74



75



76

3 Deep Learning for NLP

CBOW – Neural Network Architecture (ReCAP with Optimizer)
 Predict center word from (bag of) context words.
 Summary of CBOW Training (Review your understanding with equations)

6. Calculate the score value for the output layer. The higher score is produced when words are closer.
 $z = W' \times \hat{p} \in \mathbb{R}^{|V|}$

7. Calculate the probability using softmax
 $\hat{y} = \text{softmax}(z) \in \mathbb{R}^{|V|}$

8. Train the parameter matrix using **objective function**.

$$H(\hat{y}, y) = - \sum_{j=1}^{|V|} y_j \log(\hat{y}_j)$$

* Focus on minimising the value

We use an one-hot vector (one 1, the rest 0) so it will be calculated in only one.
 $H(\hat{y}, y) = -y_j \log(\hat{y}_j)$

77

3 Deep Learning for NLP

CBOW – Neural Network Architecture (ReCAP with Optimizer)
 Predict center word from (bag of) context words
 Sentence: "Sydney is the state capital of NSW"

6. Calculate the score value for the output layer. The higher score is produced when words are closer.
 $z = W' \times \hat{p} \in \mathbb{R}^{|V|}$

7. Calculate the probability using softmax
 $\hat{y} = \text{softmax}(z) \in \mathbb{R}^{|V|}$

8. Train the parameter matrix using **objective function**.

$$H(\hat{y}, y) = - \sum_{j=1}^{|V|} y_j \log(\hat{y}_j)$$

* Focus on minimising the value

We use an one-hot vector (one 1, the rest 0) so it will be calculated in only one.
 $H(\hat{y}, y) = -y_j \log(\hat{y}_j)$

78

3 Deep Learning for NLP

CBOW – Neural Network Architecture (ReCAP with Optimizer)
 Predict center word from (bag of) context words.
 Summary of CBOW Training (Review your understanding with equations)

6. Calculate the score value for the output layer. The higher score is produced when words are closer.
 $z = W' \times \hat{p} \in \mathbb{R}^{|V|}$

7. Calculate the probability using softmax
 $\hat{y} = \text{softmax}(z) \in \mathbb{R}^{|V|}$

The softmax is an operator that will be used frequently. It transforms a vector into a vector whose i-th component is:

$$\frac{e^{y_i}}{\sum_{j=1}^{|V|} e^{y_j}}$$

- Exponentiate to make positive
- Dividing by $\sum_{j=1}^{|V|} e^{y_j}$ normalizes the vector ($\sum_{j=1}^{|V|} \hat{y}_j = 1$) to give probability

79

3 Deep Learning for NLP

CBOW – Neural Network Architecture (ReCAP with Optimizer)
 Predict center word from (bag of) context words.
 Summary of CBOW Training (Review your understanding with equations)

6. Calculate the score value for the output layer. The higher score is produced when words are closer.
 $z = W' \times \hat{p} \in \mathbb{R}^{|V|}$

7. Calculate the probability using softmax
 $\hat{y} = \text{softmax}(z) \in \mathbb{R}^{|V|}$

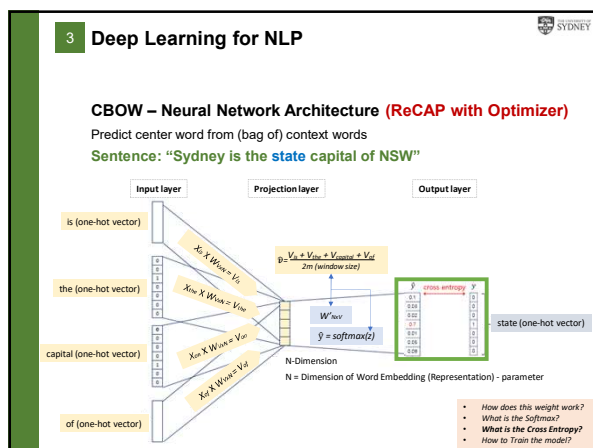
8. Train the parameter matrix using **objective function**.

$$H(\hat{y}, y) = - \sum_{j=1}^{|V|} y_j \log(\hat{y}_j)$$

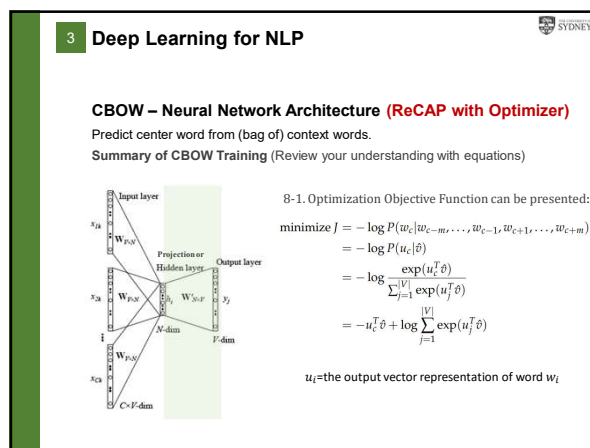
* Focus on minimising the value

We use an one-hot vector (one 1, the rest 0) so it will be calculated in only one.
 $H(\hat{y}, y) = -y_j \log(\hat{y}_j)$

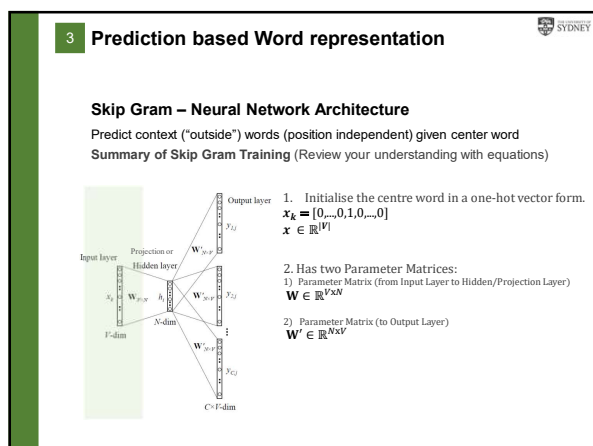
80



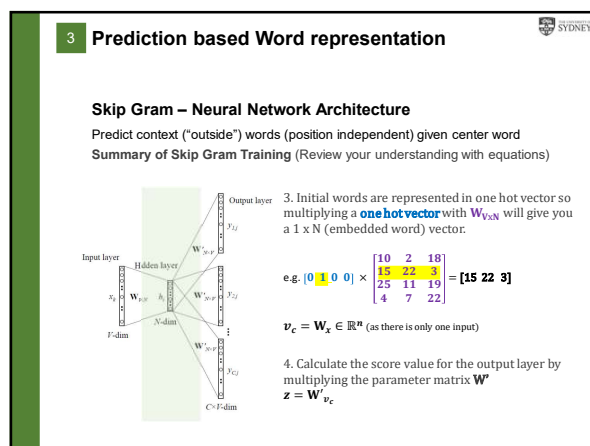
81



82



83

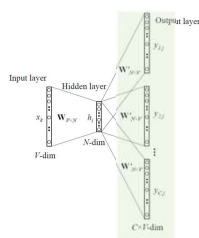


84

3 Prediction based Word representation

Skip Gram – Neural Network Architecture

Predict context ("outside") words (position independent) given center word
Summary of Skip Gram Training (Review your understanding with equations)



5. Calculate the probability using softmax
 $\hat{y} = \text{softmax}(\mathbf{z})$

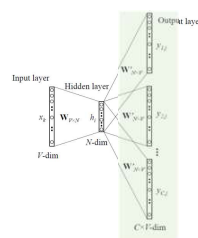
6. Calculate 2m probabilities as we need to predict 2m context words.
 $\hat{y}_{c-m}, \dots, \hat{y}_{c-1}, \hat{y}_{c+1}, \dots, \hat{y}_{c+m}$
and compare with the ground truth (one-hot vector)
 $y_{c-m}, \dots, y_{c-1}, y_{c+1}, \dots, y_{c+m}$

85

3 Prediction based Word representation

Skip Gram – Neural Network Architecture

Predict context ("outside") words (position independent) given center word
Summary of Skip Gram Training (Review your understanding with equations)



8. As in CBOW, use an objective function for us to evaluate the model. A key difference here is that we invoke a Naïve Bayes assumption to break out the probabilities. It is a strong naïve conditional independence assumption. Given the centre word, all output words are completely independent.

$$\begin{aligned} \text{minimize } J &= -\log P(w_{c-m}, \dots, w_{c-1}, w_{c+1}, \dots, w_{c+m} | w_c) \\ &= -\log \prod_{j=0, j \neq m}^{2m} P(w_{c-m+j} | w_c) \\ &= -\log \prod_{j=0, j \neq m}^{2m} \frac{\exp(u_{c-m+j}^T v_c)}{\sum_{k=1}^{|V|} \exp(u_k^T v_c)} \\ &= -\sum_{j=0, j \neq m}^{2m} u_{c-m+j}^T v_c + 2m \log \sum_{k=1}^{|V|} \exp(u_k^T v_c) \end{aligned}$$

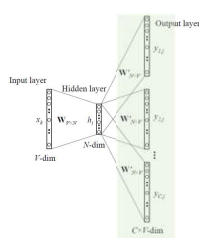
u_i = the output vector representation of word w_i

86

3 Prediction based Word representation

Skip Gram – Neural Network Architecture

Predict context ("outside") words (position independent) given center word
Summary of Skip Gram Training (Review your understanding with equations)



8-1. With this objective function, we can compute the gradients with respect to the unknown parameters and at each iteration update them via Stochastic Gradient Descent

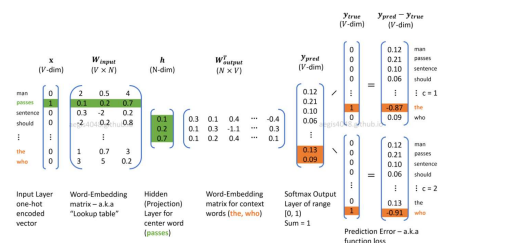
$$\begin{aligned} J &= -\sum_{j=0, j \neq m}^{2m} \log P(u_{c-m+j} | v_c) \\ &= -\sum_{j=0, j \neq m}^{2m} H(\hat{y}_j, y_{c-m+j}) \end{aligned}$$

87

4 Deep Learning for NLP

Word2Vec-SkipGram Overview

With a simple diagram



<https://github.com/kyegomez/optimizing-computational-efficiency-of-skip-gram-with-negative-sampling>

88

4 Deep Learning for NLP

Key Parameter (2) for Training methods: Negative Samples (From lecture 2)

The number of negative samples is another factor of the training process.

Negative samples to our dataset – samples of words that are not neighbors

Negative sample: 2

Input word	Output word	Target
eat	mango	1
eat	exam	0
eat	tobacco	0

*1= Appeared, 0=Not Appeared

Negative sample: 5

Input word	Output word	Target
eat	mango	1
eat	exam	0
eat	tobacco	0
eat	pool	0
eat	supervisor	0

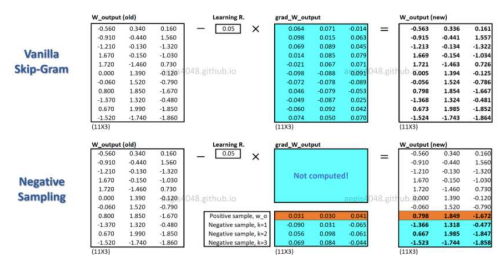
The original paper prescribes 5-20 as being a good number of negative samples. It also states that 2-5 seems to be enough when you have a large enough dataset.

89

4 Deep Learning for NLP

Word2Vec-SkipGram Overview – negative sampling

With a simple diagram



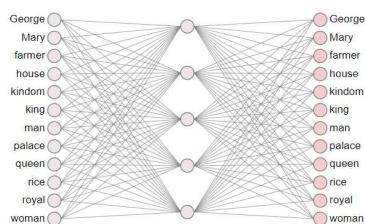
<https://github.com/448/optimizing-computational-efficiency-of-skip-gram-with-negative-sampling>

90

4 Deep Learning for NLP

Application

Application #1: Embedding Pretraining



<http://roaman.github.io/word/>

91

0 LECTURE PLAN

Lecture 3: Word Classification and Machine Learning


1. Previous Lecture: Word Embedding Review
2. Word Embedding Evaluation
3. Deep Neural Network for Natural Language Processing
 1. Perceptron and Neural Network (NN)
 2. Multilayer Perceptron
 3. Applications

4. Next Week Preview

See how the Deep Learning can be used for NLP

- Text Classification, etc.

92

SYDNEY

Reference

Reference for this lecture

- Deng, L., & Liu, Y. (Eds.), (2018). Deep Learning in Natural Language Processing. Springer.
- Rao, D., & McMehan, B. (2019). Natural Language Processing with PyTorch: Build Intelligent Language Applications Using Deep Learning. "O'Reilly Media, Inc."
- Manning, C. D., Manning, C. D., & Schütze, H. (1999). Foundations of statistical natural language processing. MIT press.
- Blunsom, P 2017, Deep Natural Language Processing, lecture notes, Oxford University
- Manning, C 2017, Natural Language Processing with Deep Learning, lecture notes, Stanford University