COMP 4446 / 5046
Natural Language Processing

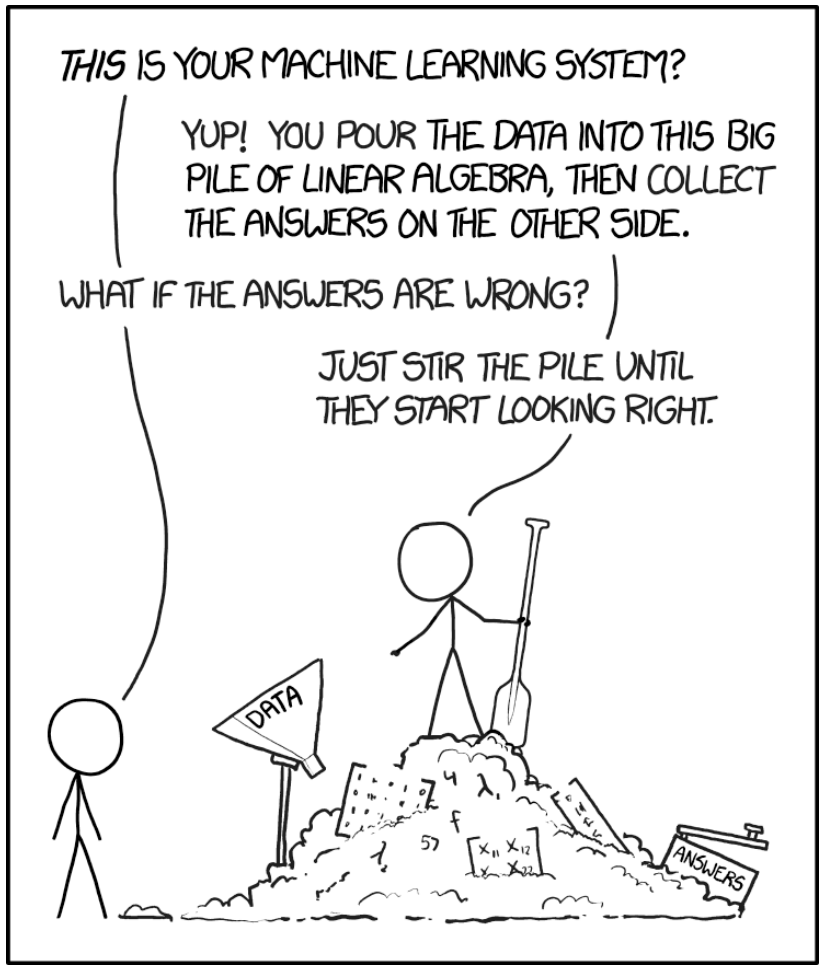
Lecture 3: Word Classification and Machine Learning

Dr. Jonathan Kay Kummerfeld

Semester 1, 2023
School of Computer Science,
University of Sydney

Questions: www.menti.com 2455 0800

Machine Learning



[The pile gets soaked with data and starts to get mushy over time, so it's technically recurrent.]

Source: https://xkcd.com/1838/

0 Admin

- Thursday labs online
- Extra materials
  - From lab demonstrators in Ed, in the Resources section
  - From me on Canvas, link in the Lecture Materials section
- Assignment 1 deadline to be pushed back 2 days.
  Working on official change, will update the website once confirmed
- Student who enrolled after their lab last week may submit Lab 2 a week late without penalty. Please email me to confirm you plan to do so
- Lab 2 marks delayed (will release after all are marked)

Questions: www.menti.com 2455 0800

0 Admin

Communication

- Ed for content questions
- Email for admin questions
- Not checking Zoom chat. Ask and vote on questions using menti

Questions: www.menti.com 2455 0800

0 LECTURE PLAN

THE UNIVERSITY OF SYDNEY

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

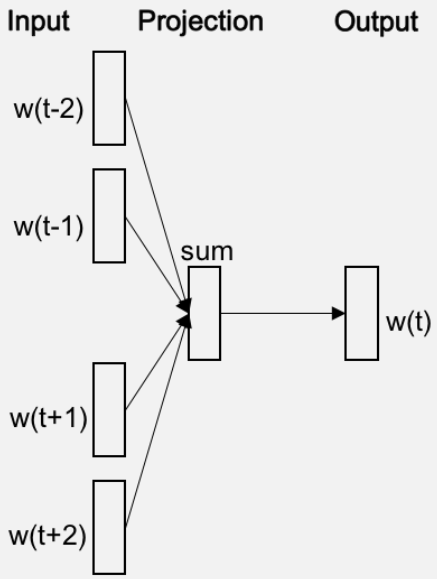
    See how the Deep Learning can be used for NLP
    - Text Classification, etc.

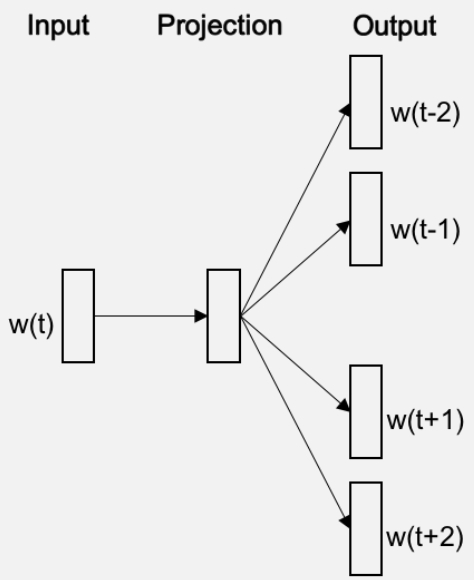
Questions: www.menti.com 2455 0800

1 Previous Lecture Review

Word2Vec Models

CBOW	*Skip-gram*
Predict center word from (bag of) context words	*Predict context words given center word*





Questions: www.menti.com 2455 0800

1 Previous Lecture Review

Word2Vec with Continuous Bag of Words (CBOW)

Predict centre word from (bag of) context words

Sentence: "Sydney is the state capital of NSW"

Using window slicing, form the training data

Sydney	is	the	state	capital	of	NSW
Sydney	**is**	the	state	capital	of	NSW
Sydney	is	**the**	state	capital	of	NSW
Sydney	is	the	**state**	capital	of	NSW
Sydney	is	the	state	**capital**	of	NSW
Sydney	is	the	state	capital	**of**	NSW
Sydney	is	the	state	capital	of	**NSW**

Center word

Context ("outside") word

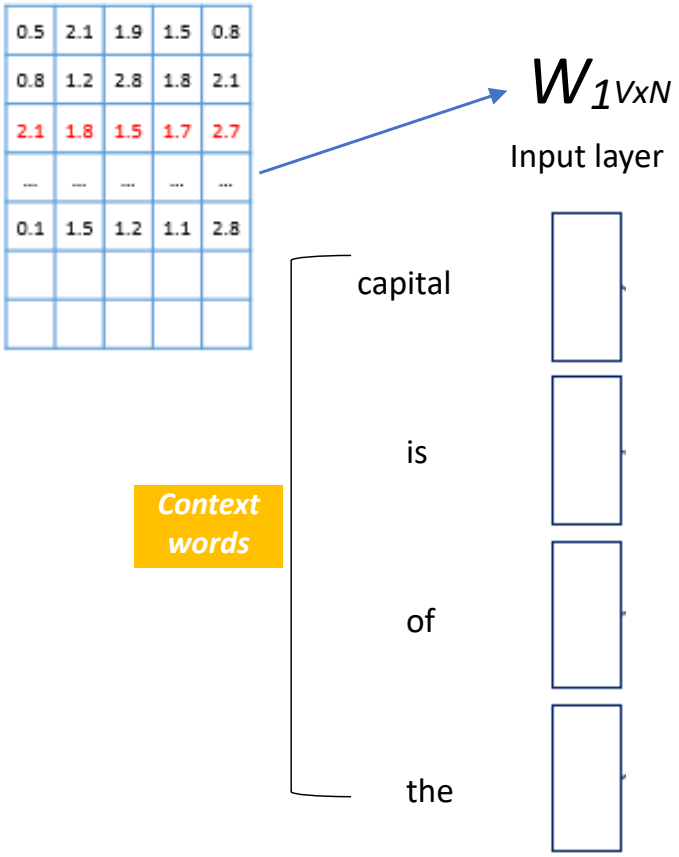
Questions: www.menti.com 2455 0800

1 Previous Lecture Review

CBOW – Neural Network Architecture

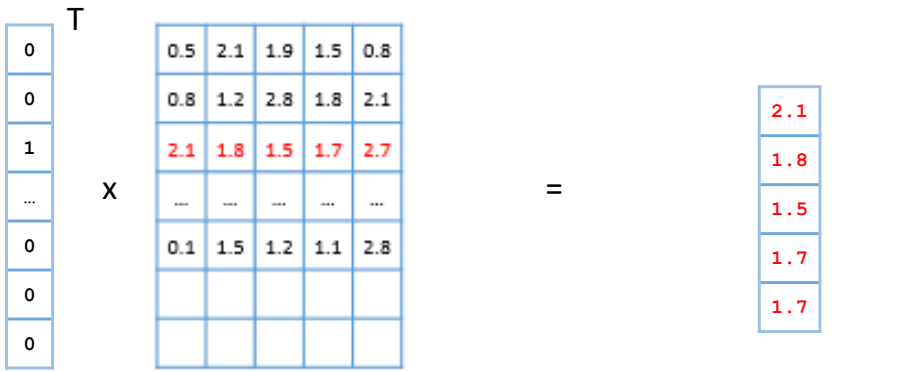
Predict centre word from (bag of) context words

Sentence: "Sydney is the state capital of NSW"



One hot or not? - Equivalent in practise

Option (1)
Take a one-hot vector and multiply it by the W_1 matrix



Option (2)
Look up a vector in the W_1 matrix

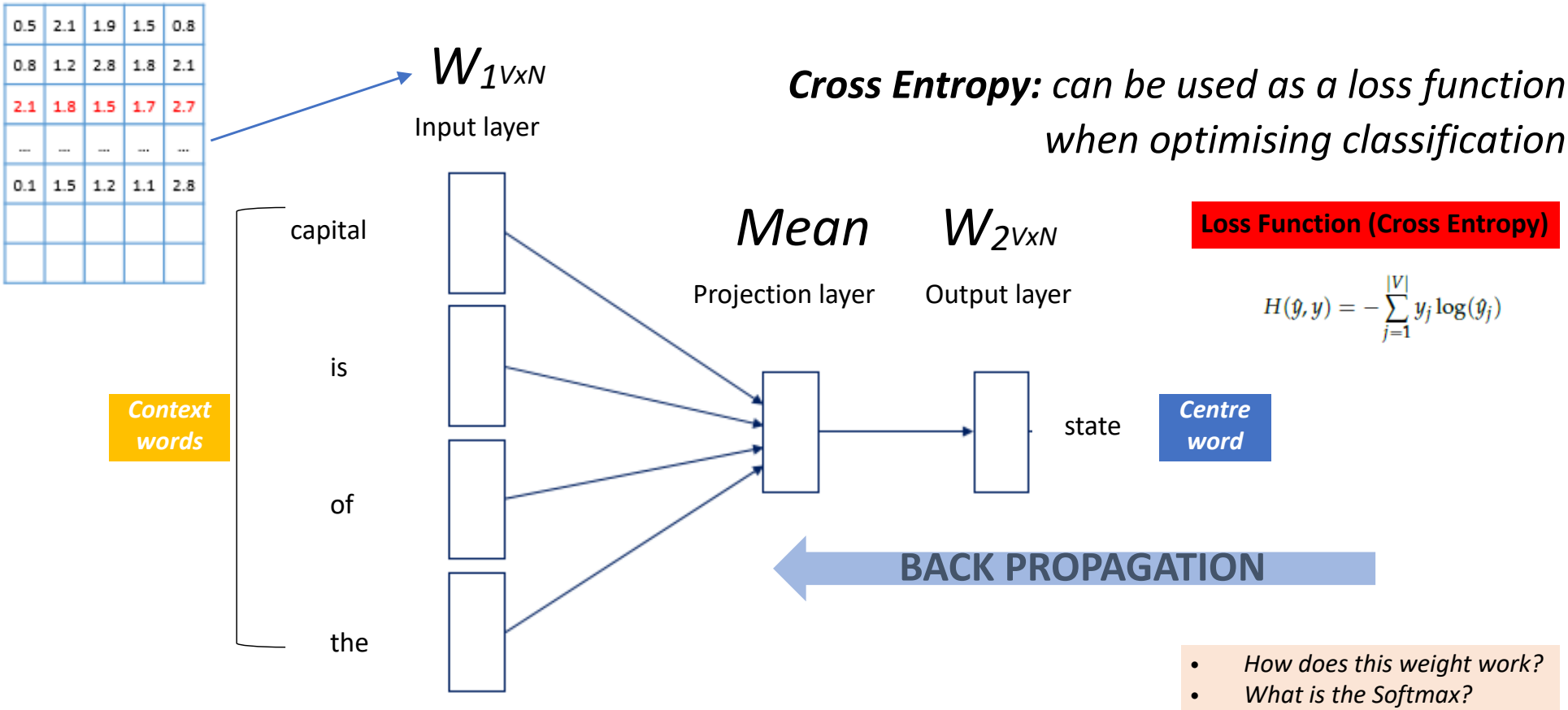
Questions: www.menti.com 2455 0800

1 Previous Lecture Review

CBOW – Neural Network Architecture

Predict centre word from (bag of) context words

Sentence: "Sydney is the state capital of NSW"



Cross Entropy: *can be used as a loss function when optimising classification*

- *How does this weight work?*
- *What is the Softmax?*
- *What is the Cross Entropy?*
- *How to Train the model?*

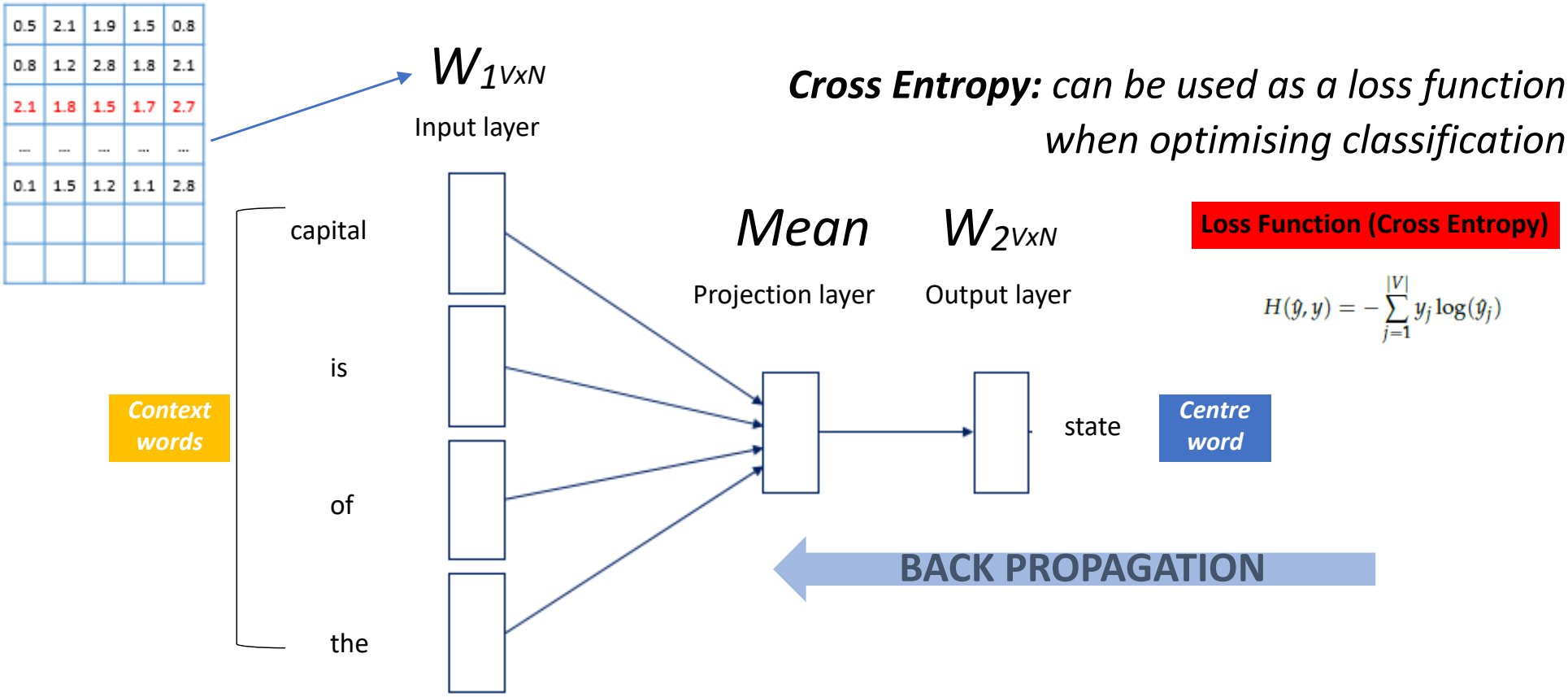
Questions: www.menti.com 2455 0800

1 Previous Lecture Review

CBOW – Neural Network Architecture

Predict centre word from (bag of) context words

Sentence: "Sydney is the state capital of NSW"



Cross Entropy: *can be used as a loss function when optimising classification*

Questions: www.menti.com 2455 0800

How can we know whether this word2vec is well trained?

0 LECTURE PLAN

THE UNIVERSITY OF SYDNEY

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

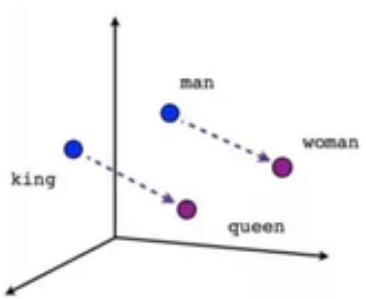
    See how the Deep Learning can be used for NLP
    - Text Classification, etc.

Questions: www.menti.com 2455 0800

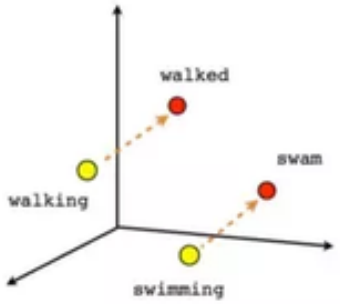
2 Word Embedding Evaluation

How to evaluate word vectors?

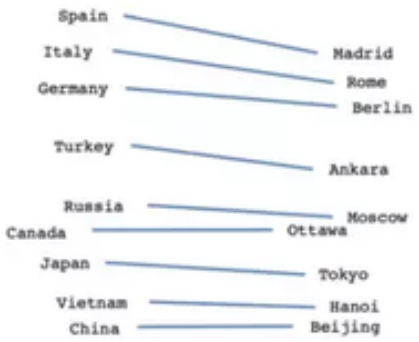
Type	How to work / Benefit
Intrinsic	Evaluation that only uses the word embeddings
	• Fast to compute • Helps to understand that system • Useful? Depends on whether they correlate with a real task
Extrinsic	Evaluation as part of a larger model or system
	• Can take a long time to compute accuracy • Unclear if the embeddings are the problem or their interaction with other subsystems



Male-Female



Verb tense



Country-Capital

Questions: www.menti.com 2455 0800

2 Word Embedding Evaluation

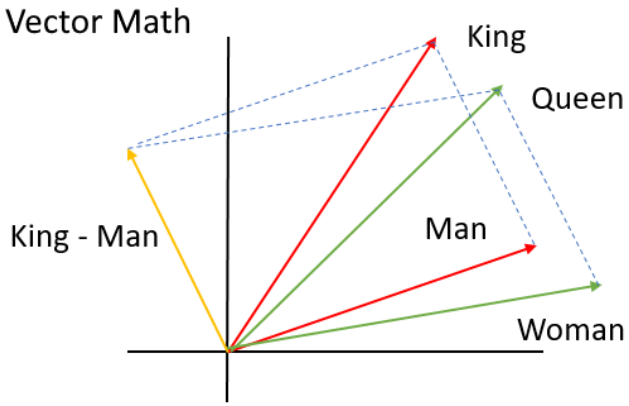
Intrinsic word vector evaluation

Word Vector Analogies

$a \longleftrightarrow b \quad :: \quad c \longleftrightarrow ???$

$man \longleftrightarrow women :: king \longleftrightarrow ???$

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



Questions: www.menti.com 2455 0800

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

Questions: www.menti.com 2455 0800

2 Word Embedding Evaluation

Intrinsic word vector evaluation

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)

Questions: www.menti.com 2455 0800

2 Word Embedding Evaluation

Intrinsic word vector evaluation

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

Model	Dim.	Size	Sem.	Syn.	Tot.
ivLBL	100	1.5B	55.9	50.1	53.2
HPCA	100	1.6B	4.2	16.4	10.8
GloVe	100	1.6B	67.5	54.3	60.3
SG	300	1B	61	61	61
CBOW	300	1.6B	16.1	52.6	36.1
vLBL	300	1.5B	54.2	64.8	60.0
ivLBL	300	1.5B	65.2	63.0	64.0
GloVe	300	1.6B	80.8	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†	300	6B	63.6	67.4	65.7
SG†	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	**81.9**	**69.3**	**75.0**

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 available2 ; (i)vLBL results are from (Mnih et al., 2013); skip-gram (SG) and CBOW results are from (Mikolov et al., 2013a,b); we trained SG† and CBOW† using the word2vec tool3 . See text for details and a description of the SVD models.

(Original Glove Paper - Pennington et al.2014)

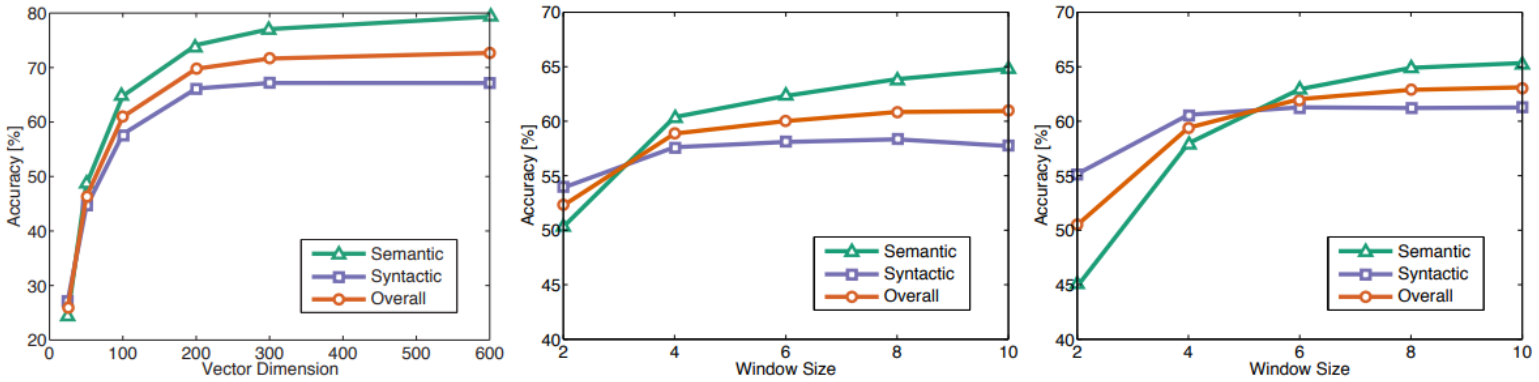
Questions: www.menti.com 2455 0800

Word Embedding Evaluation

Intrinsic word vector evaluation

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

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



(a) Symmetric context (b) Symmetric context (c) Asymmetric context

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.

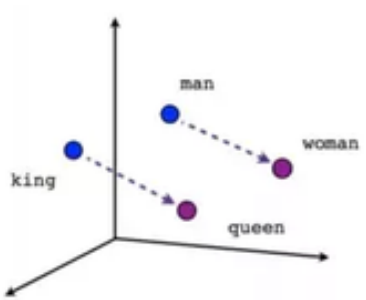
(Original Glove Paper - Pennington et al.2014)

Questions: www.menti.com 2455 0800

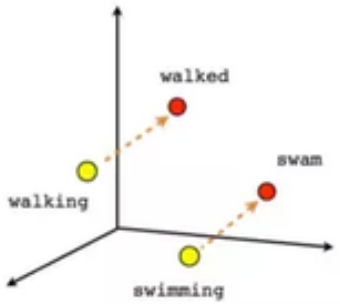
2 Word Embedding Evaluation

How to evaluate word vectors?

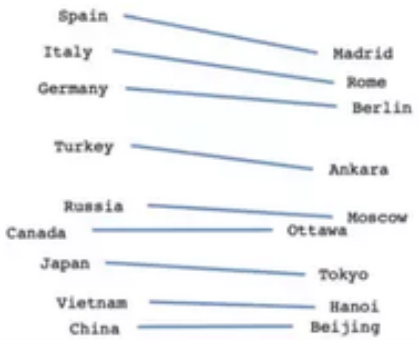
Type	How to work / Benefit
Intrinsic	Evaluation that only uses the word embeddings
	• Fast to compute • Helps to understand that system • Useful? Depends on whether they correlate with a real task
Extrinsic	**Evaluation as part of a larger model or system**
	• **Can take a long time to compute accuracy** • **Unclear if the embeddings are the problem or their interaction with other subsystems**



Male-Female



Verb tense



Country-Capital

Questions: www.menti.com 2455 0800

0 LECTURE PLAN

THE UNIVERSITY OF SYDNEY

Lecture 3: Word Classification and Machine Learning

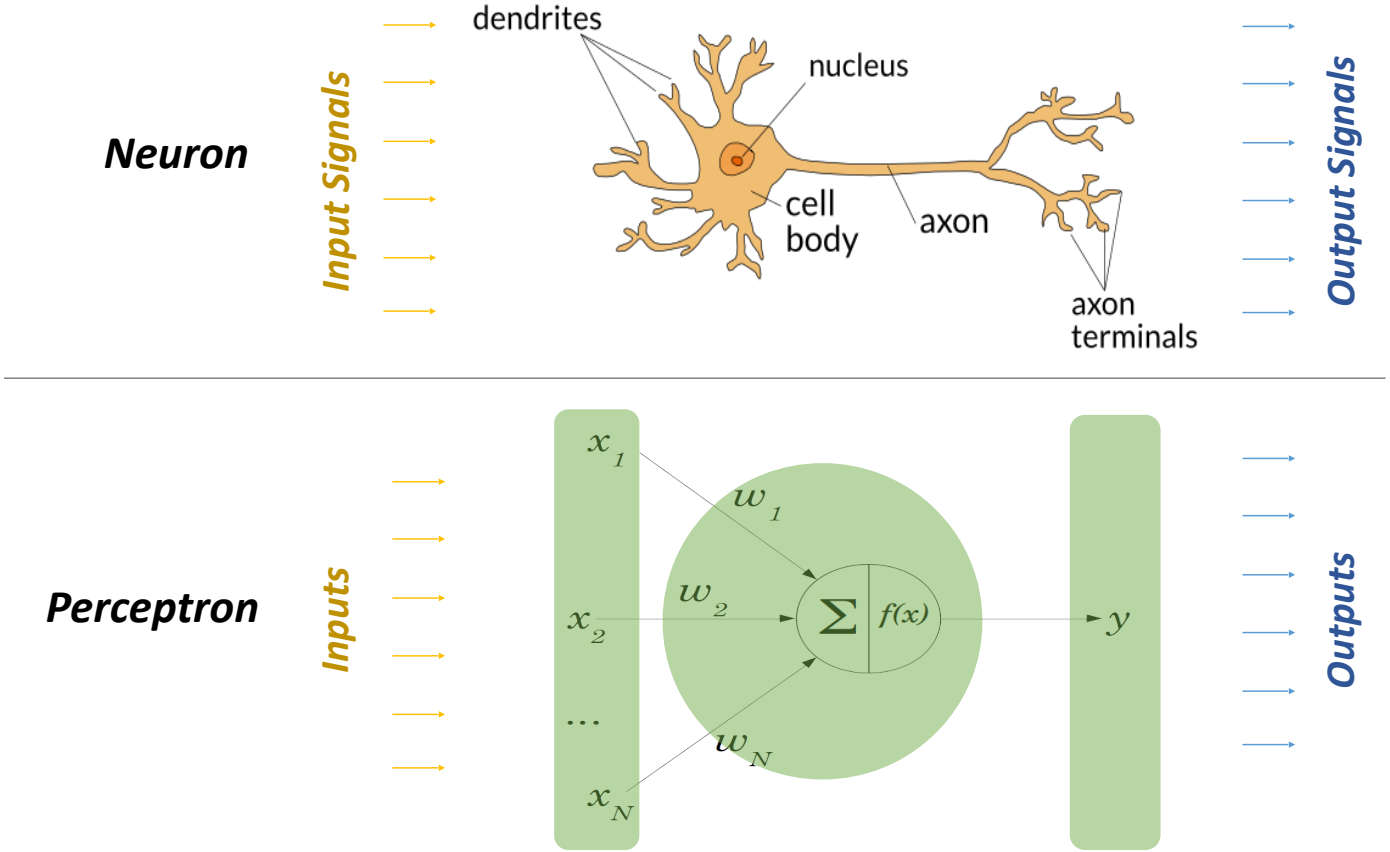
1. Previous Lecture: Word Embedding Review
2. Word Embedding Evaluation
3. **Deep Neural Network for Natural Language Processing**
    1. Perceptron and Neural Network (NN)
    2. Multilayer Perceptron
    3. Applications
4. Next Week Preview

    See how the Deep Learning can be used for NLP
    - Text Classification, etc.

Questions: www.menti.com 2455 0800

Deep Learning with Neural Network

Neuron and Perceptron



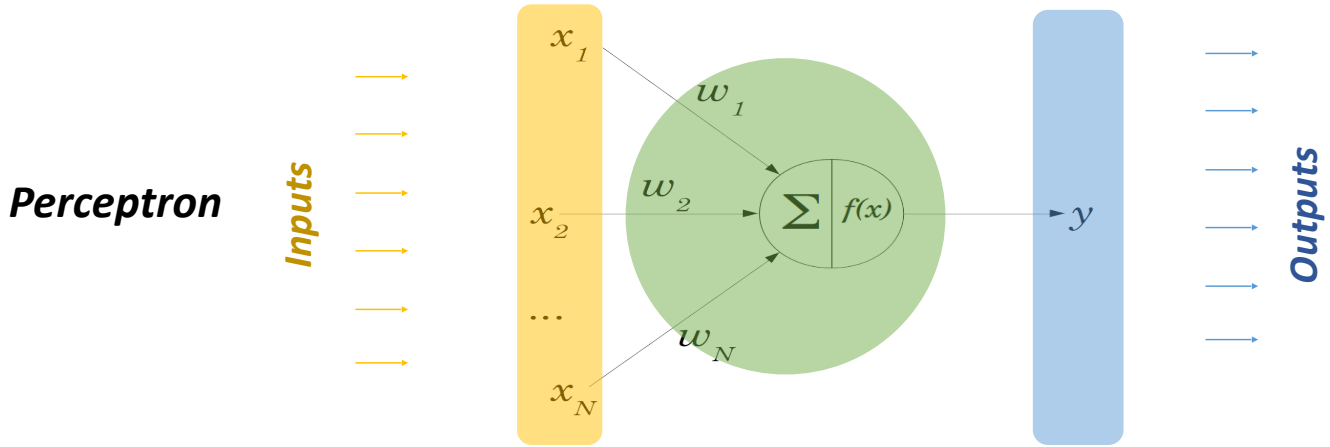
Questions: www.menti.com 2455 0800

THE UNIVERSITY OF SYDNEY

Deep Learning with Neural Network

Inputs and Outputs (Labels) for Natural Language Processing

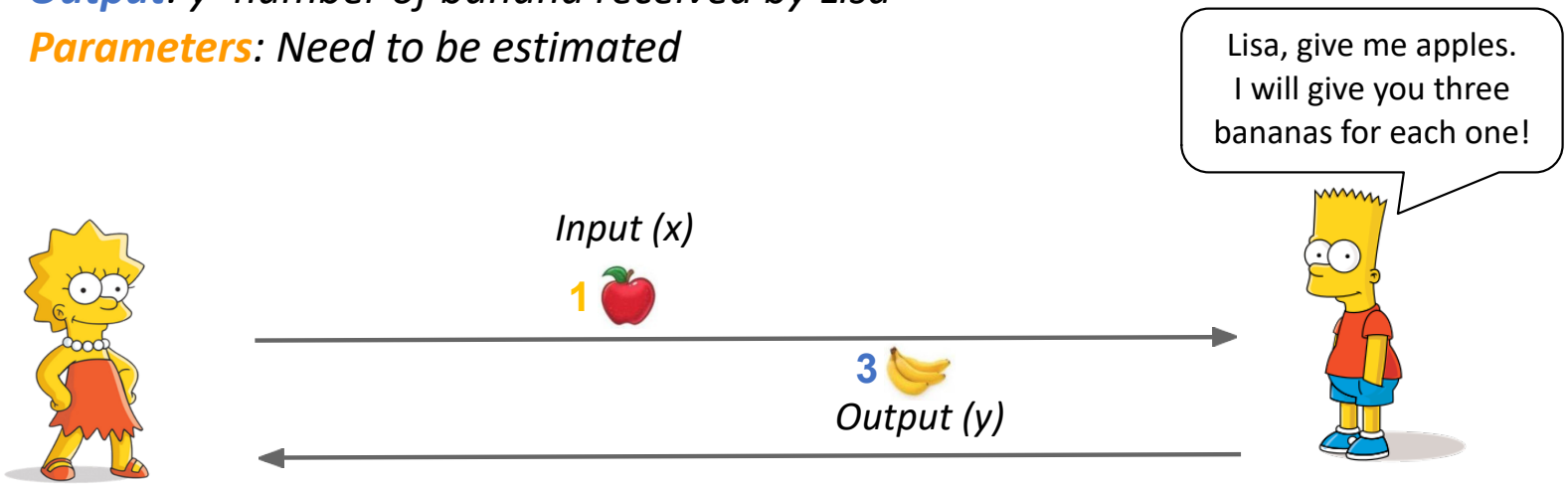
x_i	Inputs	**Features** words (indices or vectors!), context windows, sentences, documents, etc.
y_i	Outputs (labels)	**What we are trying to predict/classify** • E.g. word meaning, sentiment, name entity



Questions: www.menti.com 2455 0800

Deep Learning with Neural Network

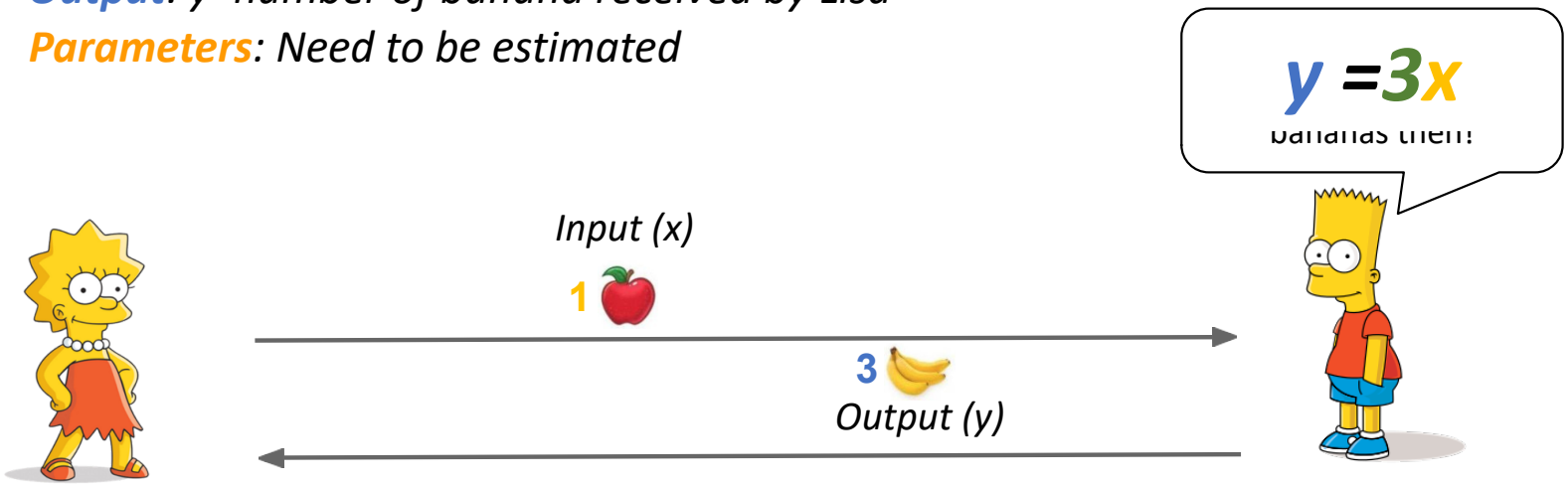
Input: *x=number of apple given by Lisa*
Output: *y=number of banana received by Lisa*
Parameters: *Need to be estimated*

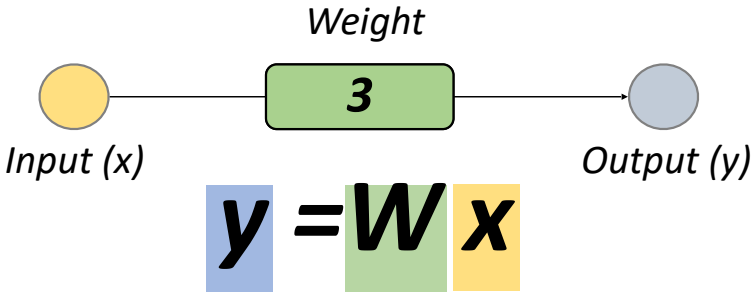


Questions: www.menti.com 2455 0800

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





Questions: www.menti.com 2455 0800

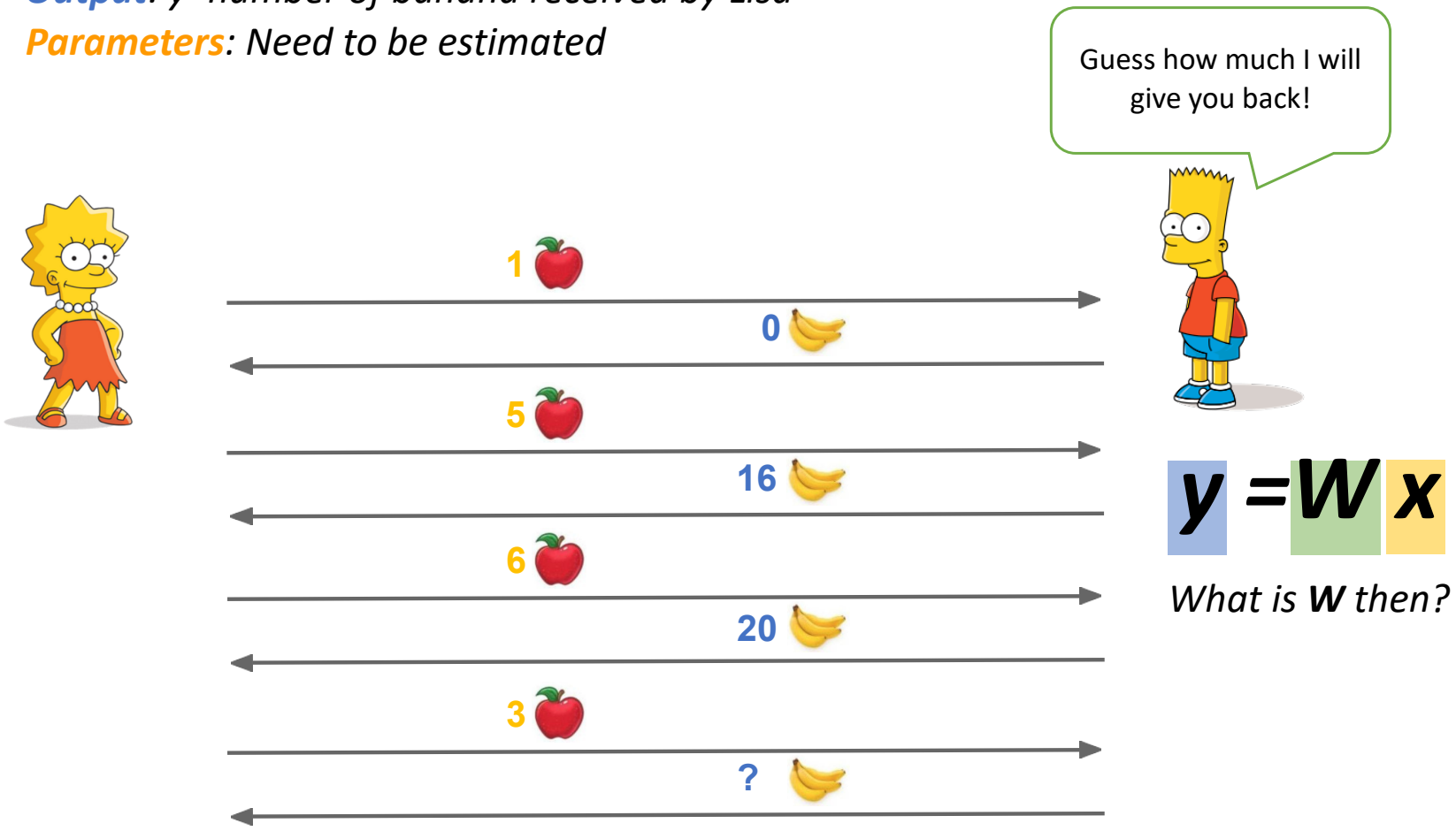
Deep Learning with Neural Network - Model

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

Questions: www.menti.com 2455 0800

Deep Learning with Neural Network - Parameter

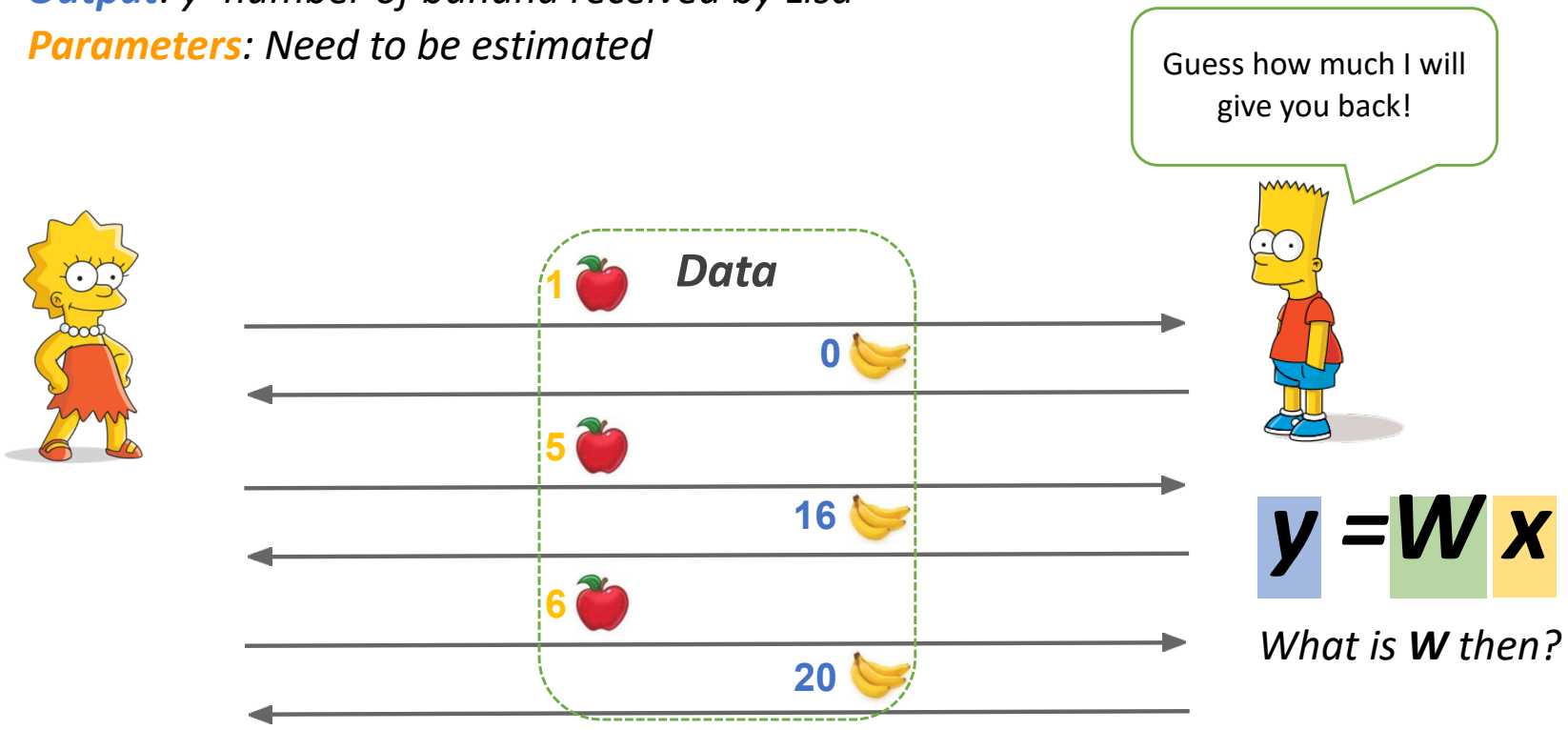
***Input**: x=number of apple given by Lisa*
***Output**: y=number of banana received by Lisa*
***Parameters**: Need to be estimated*



Questions: www.menti.com 2455 0800

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*



Questions: www.menti.com 2455 0800

THE UNIVERSITY OF SYDNEY

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*

Data

x 🍎	y 🍌
1	0
5	16
6	20

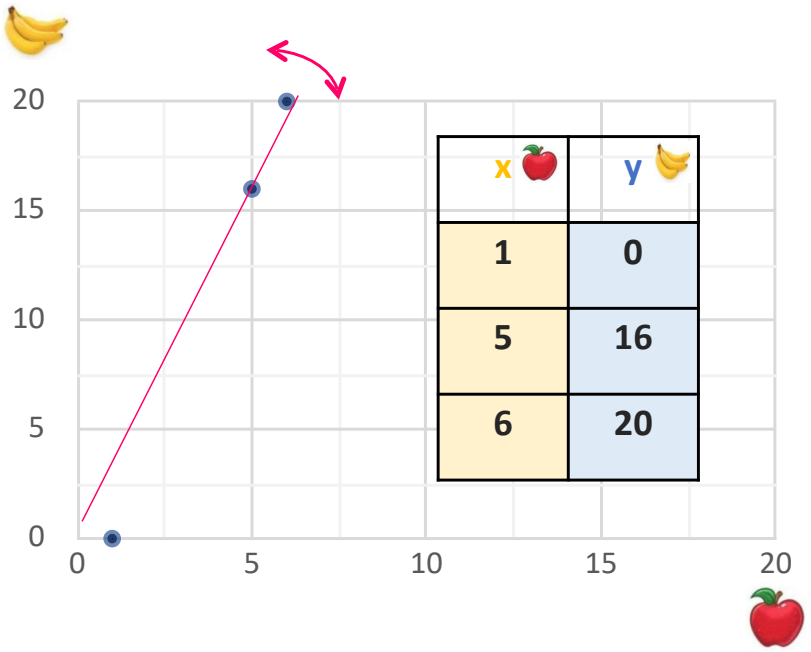
$$y = Wx$$

*What is **W** then?*

Questions: www.menti.com 2455 0800

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

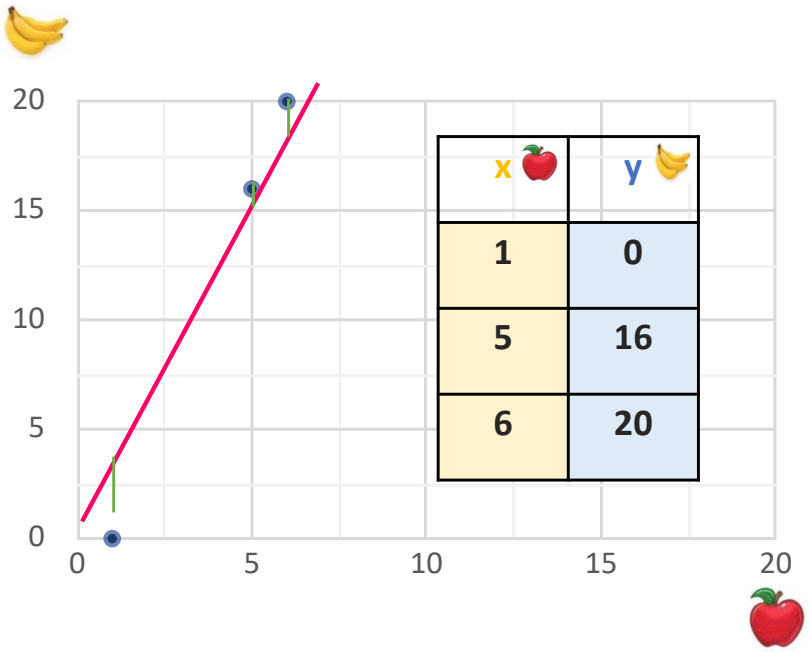
$$y = Wx$$

*What is **W** then?*

Questions: www.menti.com 2455 0800

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

$$y = Wx$$

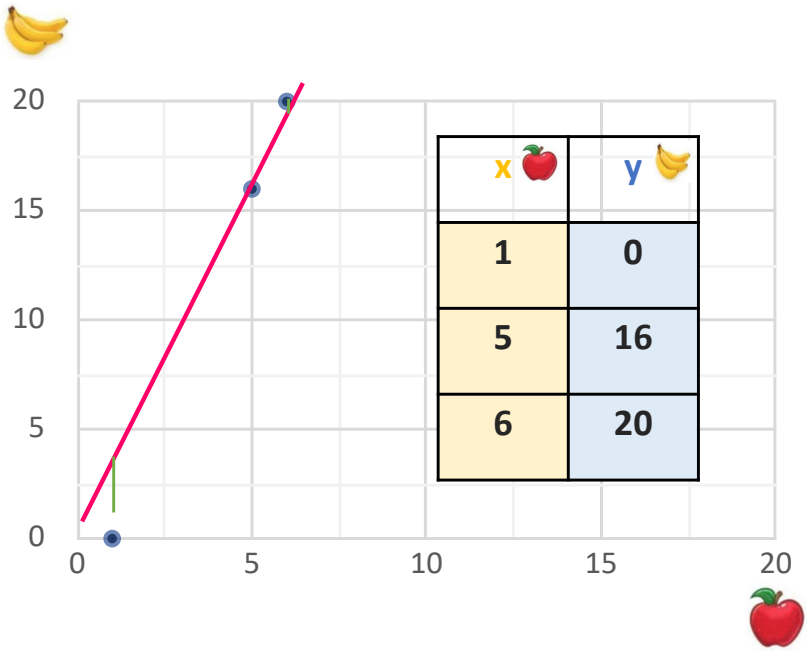
*What if W is **3**?*

3 = 3 x 1
15 = 3 x 5
20 = 3 x 6

Questions: www.menti.com 2455 0800

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

$$y = Wx$$

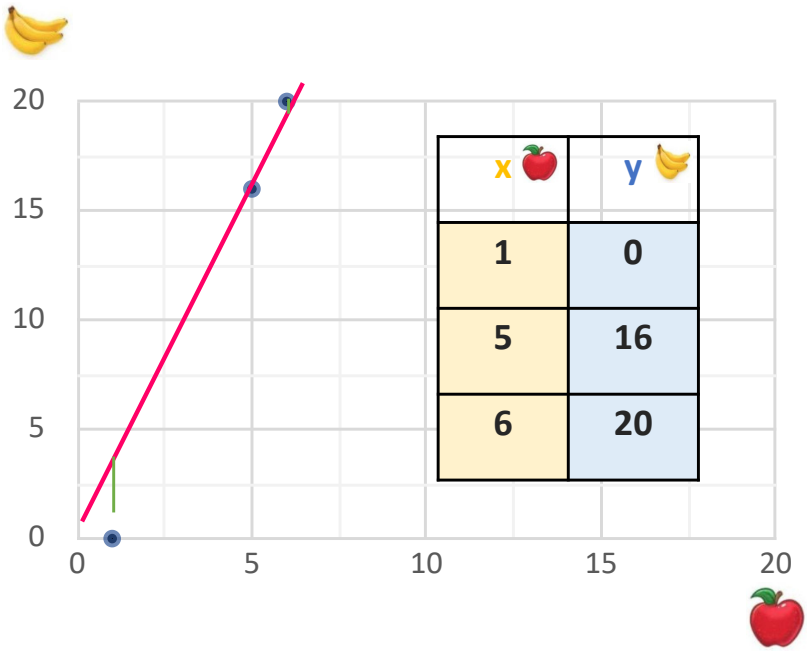
*What if W is **3.2**?*

3.2 = 3.2 x 1
16 = 3.2 x 5
19.2 = 3.2 x 6

Questions: www.menti.com 2455 0800

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 *bias*

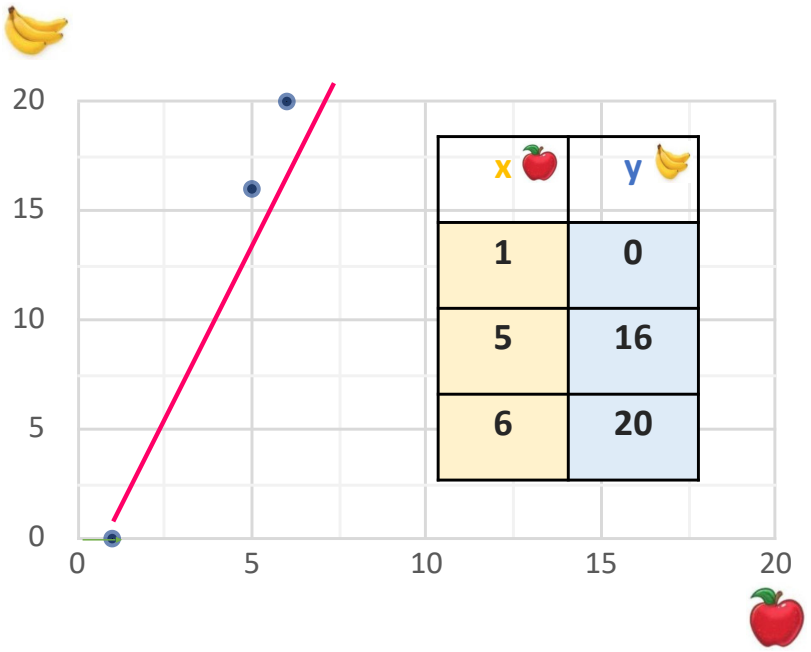
Weight is not enough…

Questions: www.menti.com 2455 0800

THE UNIVERSITY OF SYDNEY

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

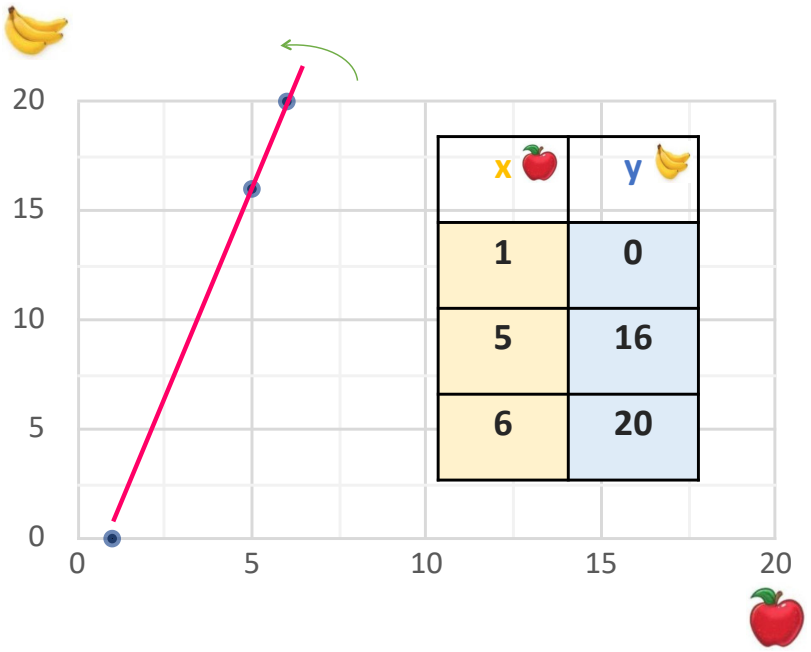
$$y = \underbrace{W}_{\text{weight}} x + \underbrace{b}_{\text{bias}}$$

*How can we find the parameters, **w** and **b**?*

Questions: www.menti.com 2455 0800

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

$$y = Wx + b$$

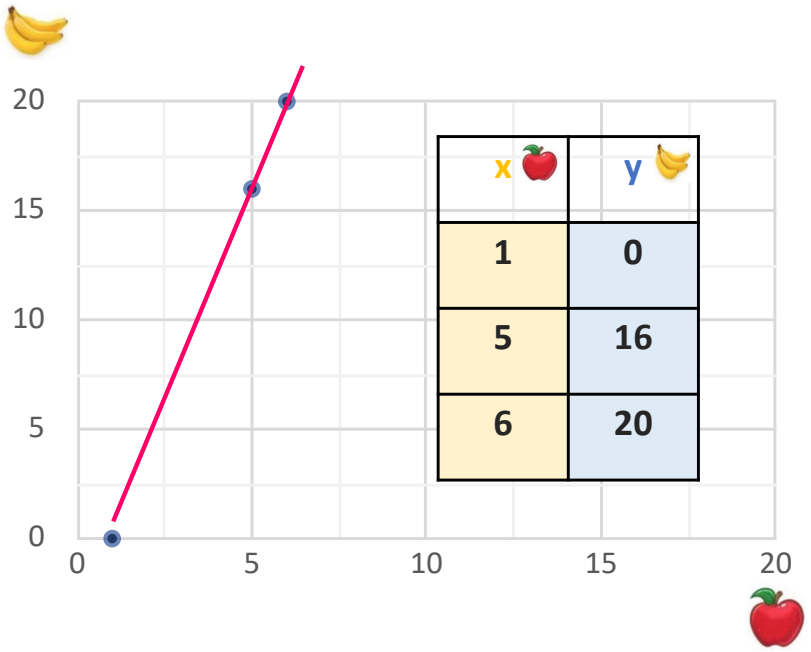
weight *bias*

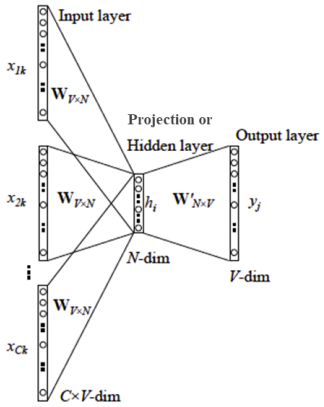
*How can we find the parameters, **w** and **b**?*

Questions: www.menti.com 2455 0800

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





Model $y = Wx + b$ (*weight*, *bias*)

*How can we find the parameters, **w** and **b**?*

Questions: www.menti.com 2455 0800

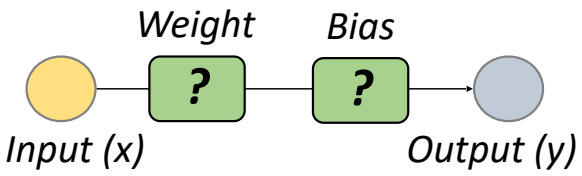
Deep Learning with Neural Network - Cost

Actual Data

weight *bias*

$y = ? \; x + ?$

x 🍎	y 🍌
1	0
5	16
6	20



Model Ex#1

weight *bias*

$\hat{y} = 1 \; x + 0$

x 🍎	ŷ 🍌	y 🍌
	predicted	*actual*
1	1	0
5	5	16
6	6	20

Model Ex#2

weight *bias*

$\hat{y} = 2 \; x + 2$

x 🍎	ŷ 🍌	y 🍌
	predicted	*actual*
1	4	0
5	12	16
6	14	20

🤔 ***Which one is closer?***

Questions: www.menti.com 2455 0800

Deep Learning for NLP

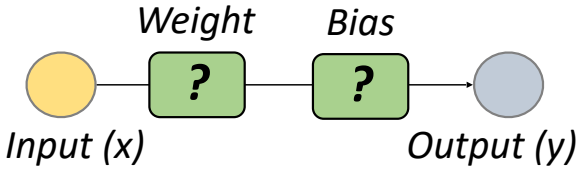
Deep Learning with Neural Network – Cost (loss)

Actual Data

weight *bias*

$$y = ? \; x + ?$$

x 🍎	y 🍌
1	0
5	16
6	20



Model Ex#1

weight *bias*

$$\hat{y} = 1 \; x + 0$$

x 🍎	ŷ 🍌	y 🍌	cost
1	1	0	
5	5	16	
6	6	20	

predicted *actual*

Model Ex#2

weight *bias*

$$\hat{y} = 2 \; x + 2$$

x 🍎	ŷ 🍌	y 🍌	cost
1	4	0	
5	12	16	
6	14	20	

predicted *actual*

😎 ***Let's calculate the cost(loss)!***

Mean Squared Error (MSE)

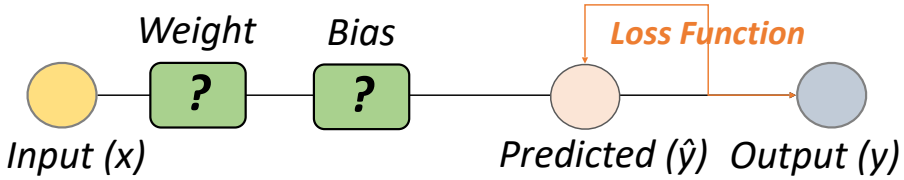
$$C(w,b) = \sum (y_n - \hat{y}_n)^2$$

$n \in \{0,1,2\}$

Questions: www.menti.com 2455 0800

Deep Learning for NLP

WAIT! Loss Function? Cost Calculation?

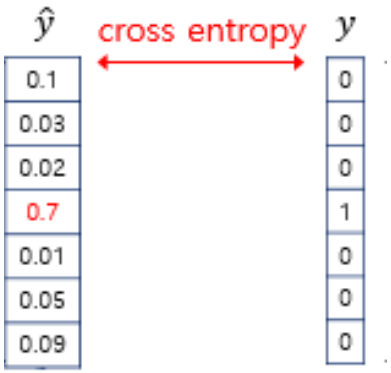


1) Mean Squared Error (MSE): measures the average of the squares of the errors

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

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

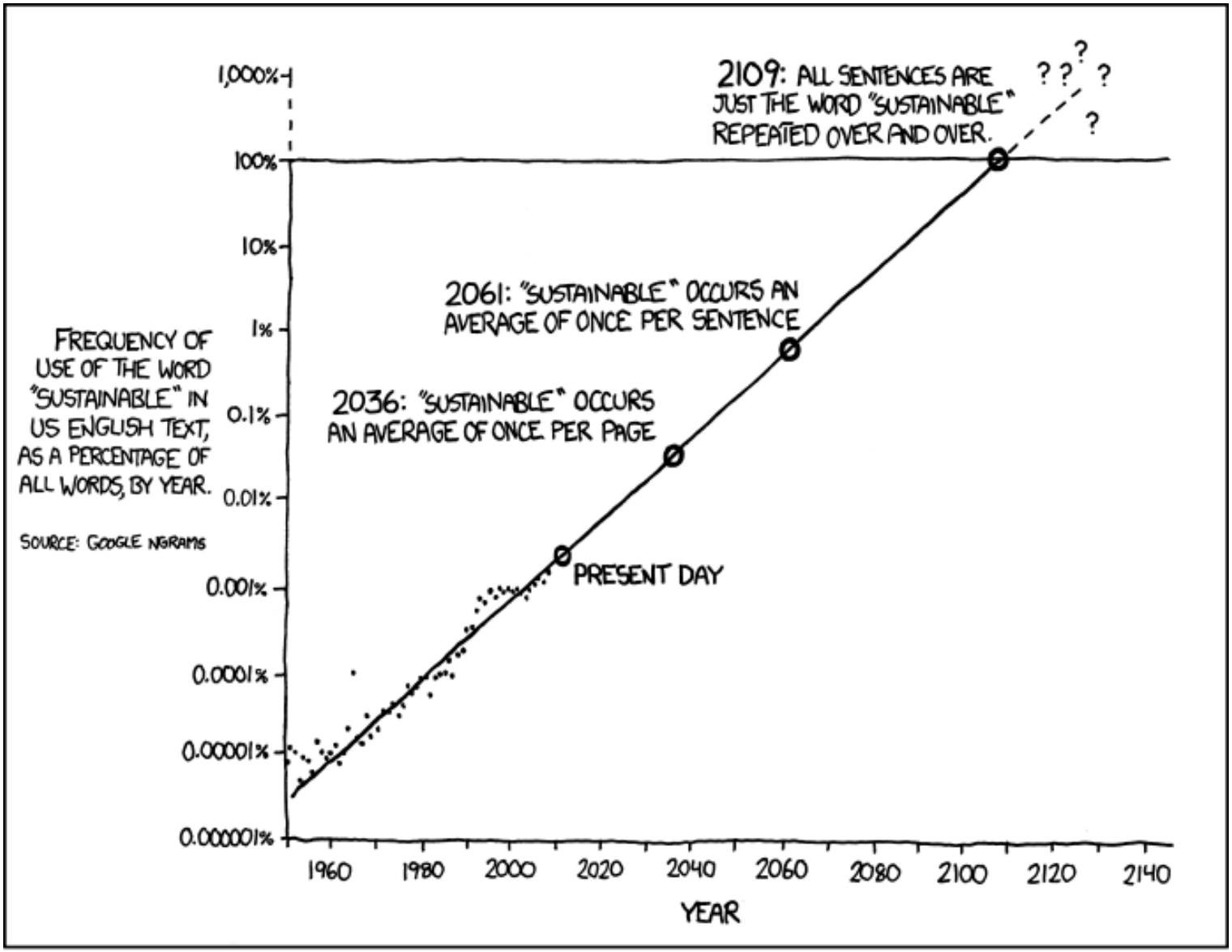
$$L_{\text{cross-entropy}}(\hat{\mathbf{y}}, \mathbf{y}) = -\sum_i y_i \log(\hat{y}_i)$$



Questions: www.menti.com 2455 0800

Break

Sustainable



[Though 100 years is longer than a lot of our resources.]

stions: www.menti.com 2455 0800

Source: https://xkcd.com/1007/

THE UNIVERSITY OF SYDNEY

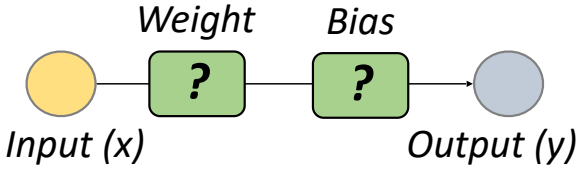
Deep Learning with Neural Network - Cost (loss)

Actual Data

weight *bias*

$y = ? \; x + ?$

x 🍎	y 🍌
1	0
5	16
6	20



Model Ex#1

weight *bias*

$\hat{y} = 1 \; x + 0$

x 🍎	ŷ 🍌	y 🍌	cost
1	1	0	1
5	5	16	121
6	6	20	196

C(1,0) = 318

Model Ex#2

weight *bias*

$\hat{y} = 2 \; x + 2$

x 🍎	ŷ 🍌	y 🍌	cost
1	4	0	16
5	12	16	16
6	14	20	36

C(2,2) = 68

😎 ***Let's calculate the cost!***

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

$n \in \{0,1,2\}$

Questions: www.menti.com 2455 0800

THE UNIVERSITY OF SYDNEY

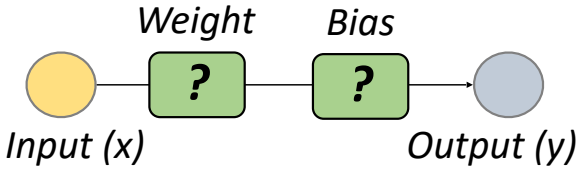
Deep Learning with Neural Network - Cost (loss)

Actual Data

weight bias

$y = ? \; x + ?$

x 🍎	y 🍌
1	0
5	16
6	20



Model Ex#1

weight bias

$\hat{y} = 1 \; x + 0$

predicted actual

x 🍎	ŷ 🍐	y 🍌	$(y-\hat{y})^2$
1	1	0	1
5	5	16	121
6	6	20	196

C(1,0) = 318

Model Ex#2

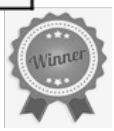
weight bias

$\hat{y} = 2 \; x + 2$

predicted actual

x 🍎	ŷ 🍐	y 🍌	$(y-\hat{y})^2$
1	4	0	16
5	12	16	16
6	14	20	36

C(2,2) = 68



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

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

$$w,b \in [-\infty, \infty]$$

Questions: www.menti.com 2455 0800

THE UNIVERSITY OF SYDNEY

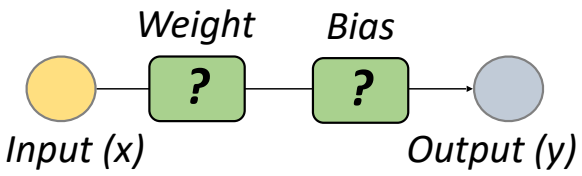
Deep Learning with Neural Network - Optimizer

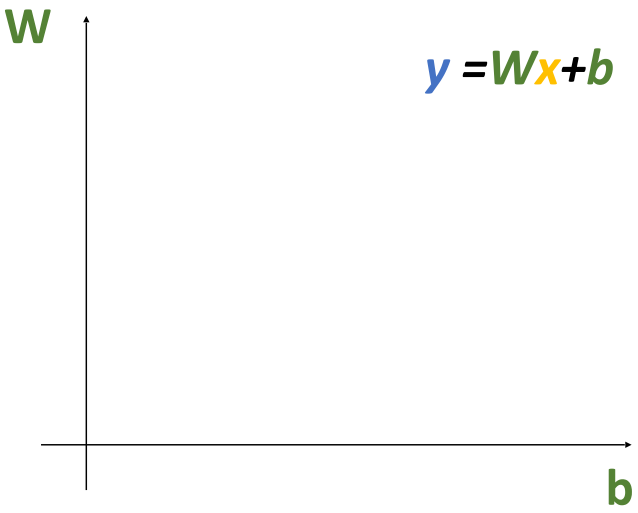
Actual Data

weight *bias*

$y = ? \; x + ?$

x 🍎	y 🍌
1	0
5	16
6	20





$y = Wx + b$

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

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

$$w,b \in [-\infty, \infty]$$

Questions: www.menti.com 2455 0800

THE UNIVERSITY OF SYDNEY

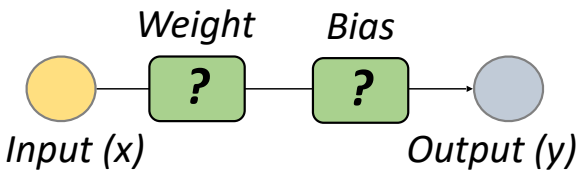
Deep Learning with Neural Network - Optimizer

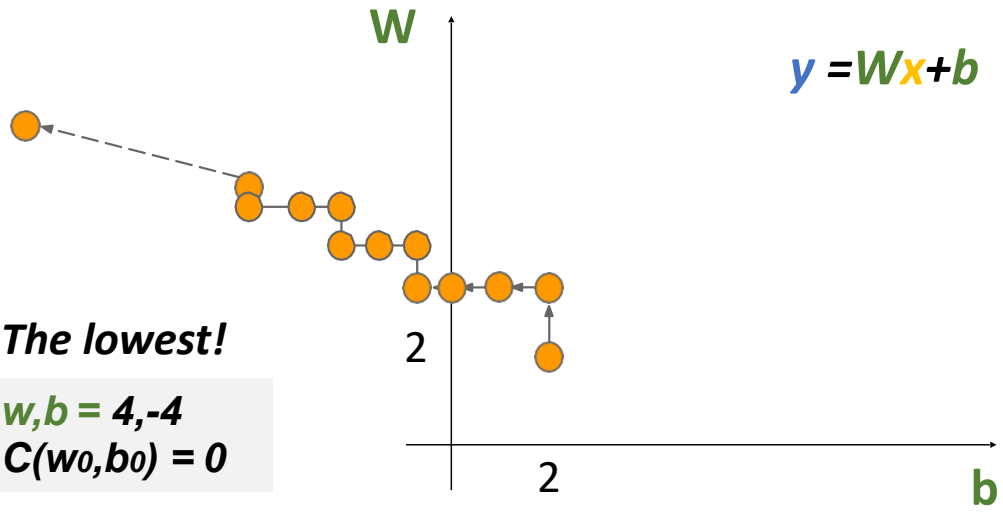
Actual Data

weight *bias*

$y = ? \; x + ?$

x 🍎	y 🍌
1	0
5	16
6	20





$y = Wx + b$

The lowest!

$w,b = 4,-4$

$C(w_0,b_0) = 0$

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

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

$$w,b \in [-\infty,\infty]$$

Questions: www.menti.com 2455 0800

THE UNIVERSITY OF SYDNEY

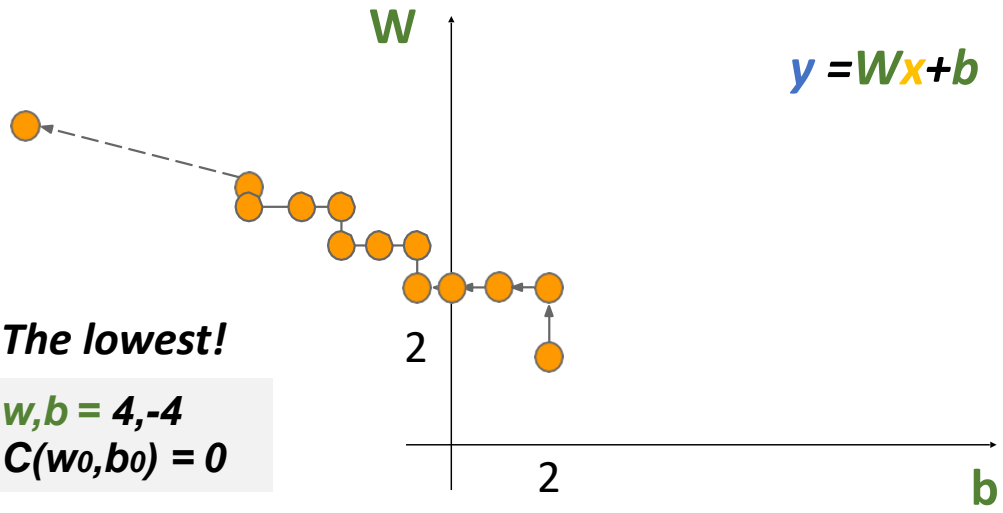
Backpropagation (weight update)

Deep Learning with Neural Network - Optimizer

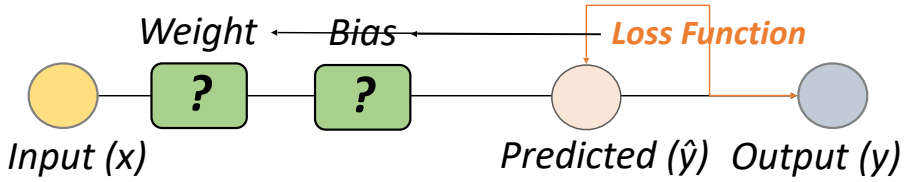
weight *bias*

$$y = 4x - 4$$

x 🍎	y 🍌
1	0
5	16
6	20



Backpropagation (weight update)



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

$w,b \in [-\infty, \infty]$

Questions: www.menti.com 2455 0800

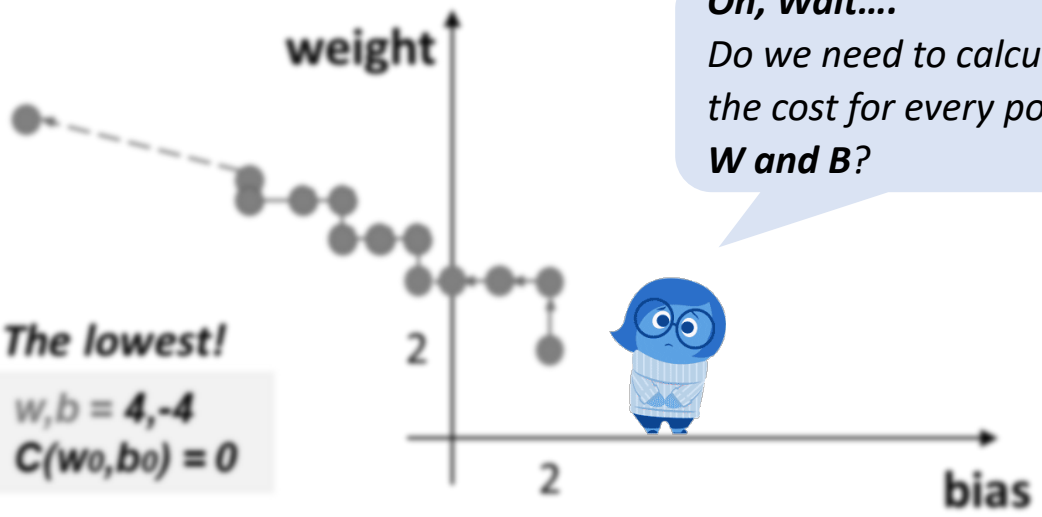
Backpropagation (weight update)

Deep Learning with Neural Network - Optimizer

weight *bias*

$$y = 4x - 4$$

x 🍎	y 🍌
1	0
5	16
6	20



Expensive to compute (hours or days)

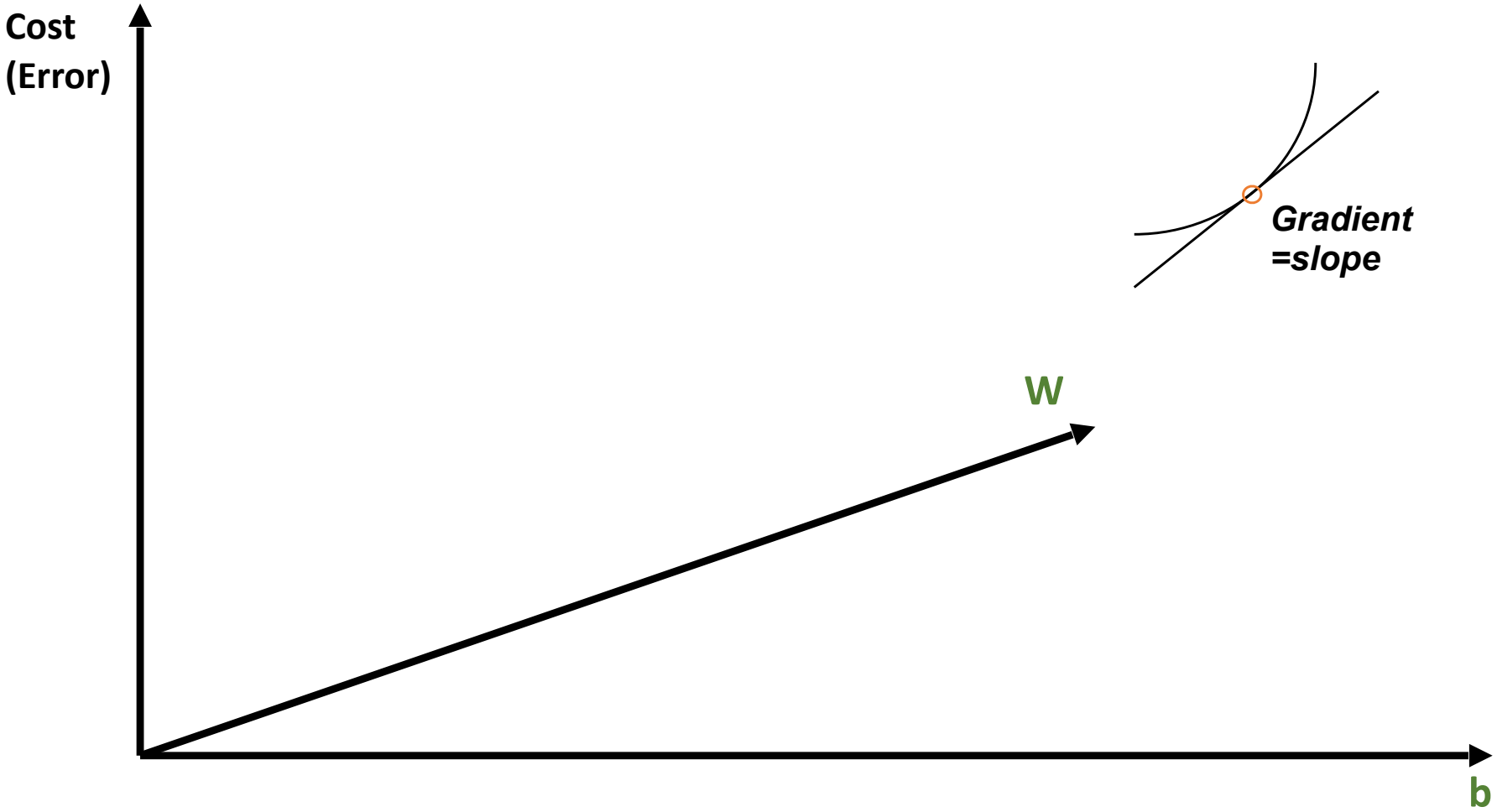
$$\arg\min C(w,b)$$

$$w,b \in [-\infty, \infty]$$

Questions: www.menti.com 2455 0800

THE UNIVERSITY OF SYDNEY

Finding the Optimal weight and bias – Gradient Descent



There are different types of Gradient descent optimisation algorithms:

Batch Gradient Descent, Stochastic Gradient Descent, Momentum, Adam, etc.

Questions: www.menti.com 2455 0800

Deep Learning for NLP

Finding the Optimal weight and bias – Gradient Descent

There are different types of Gradient descent optimisation algorithms:

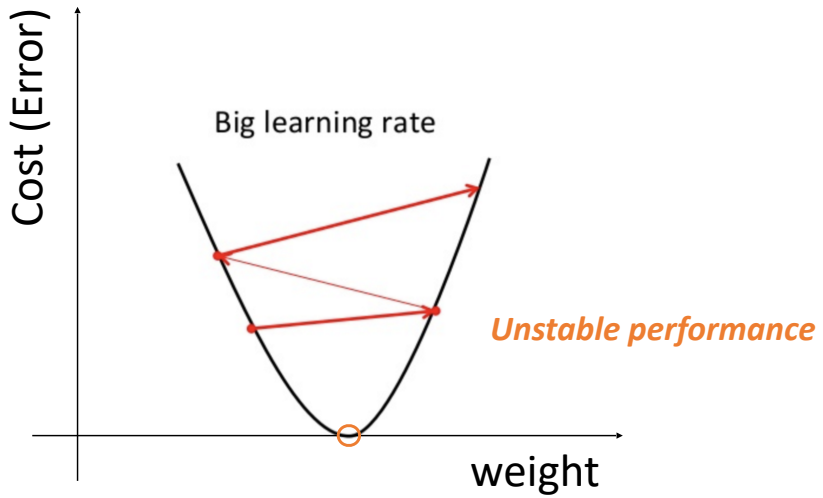
Batch Gradient Descent, Stochastic Gradient Descent, Momentum, Adam, etc.

Questions: www.menti.com 2455 0800

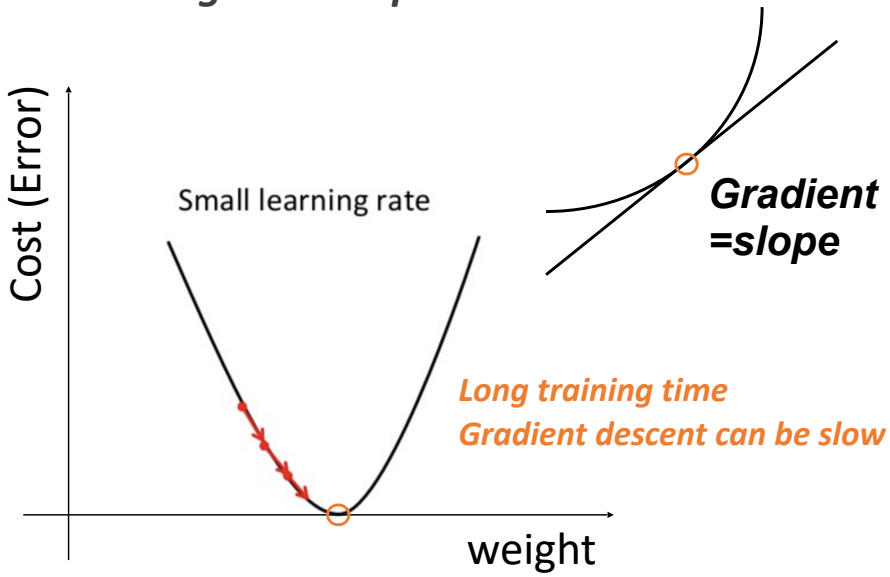
Deep Learning for NLP

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.*



Unstable performance



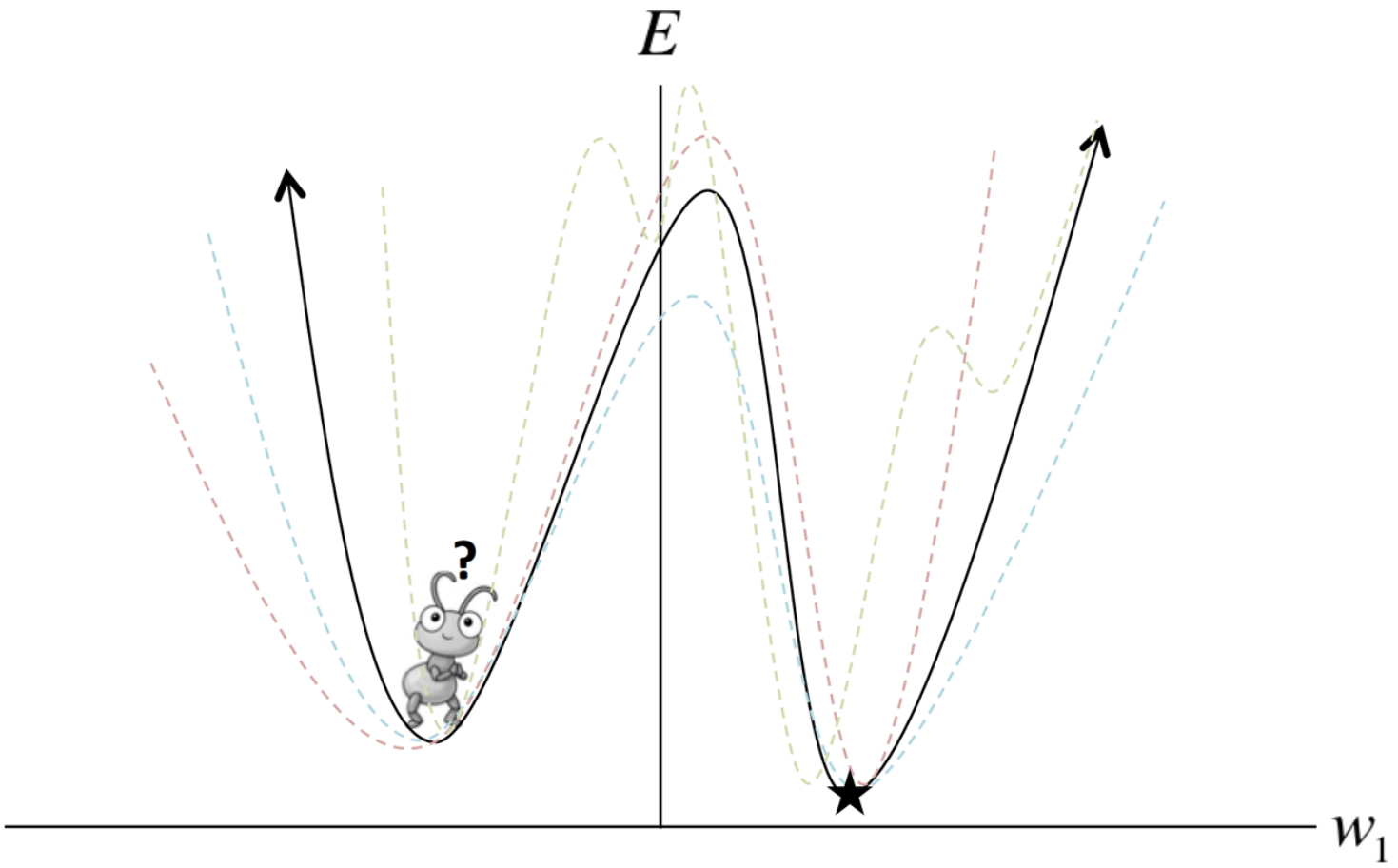
Long training time
Gradient descent can be slow

*new_weight = existing_weight — learning_rate * gradient*

Questions: www.menti.com 2455 0800

THE UNIVERSITY OF SYDNEY

Finding the Optimal weight and bias – Gradient Descent



There are different types of Gradient descent optimisation algorithms:

Batch Gradient Descent, Stochastic Gradient Descent, Momentum, Adam, etc.

Questions: www.menti.com 2455 0800

THE UNIVERSITY OF SYDNEY

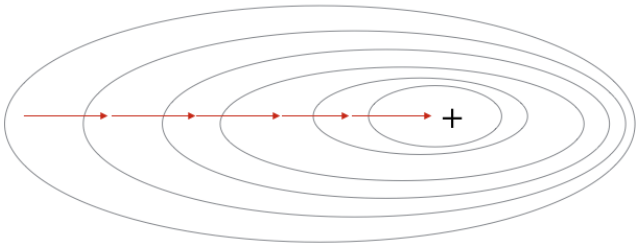
Stochastic Gradient Descent

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

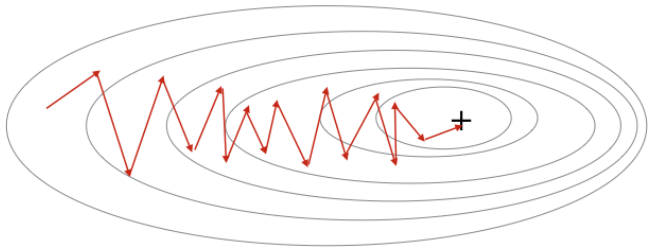
One solution is a different Gradient Descent Method.
The most common – "Stochastic Gradient Descent (SGD)"

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.

Gradient Descent



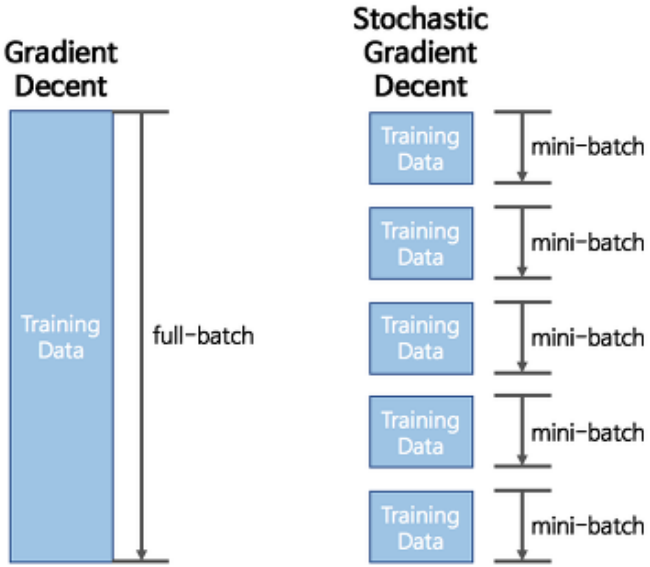
Stochastic Gradient Descent

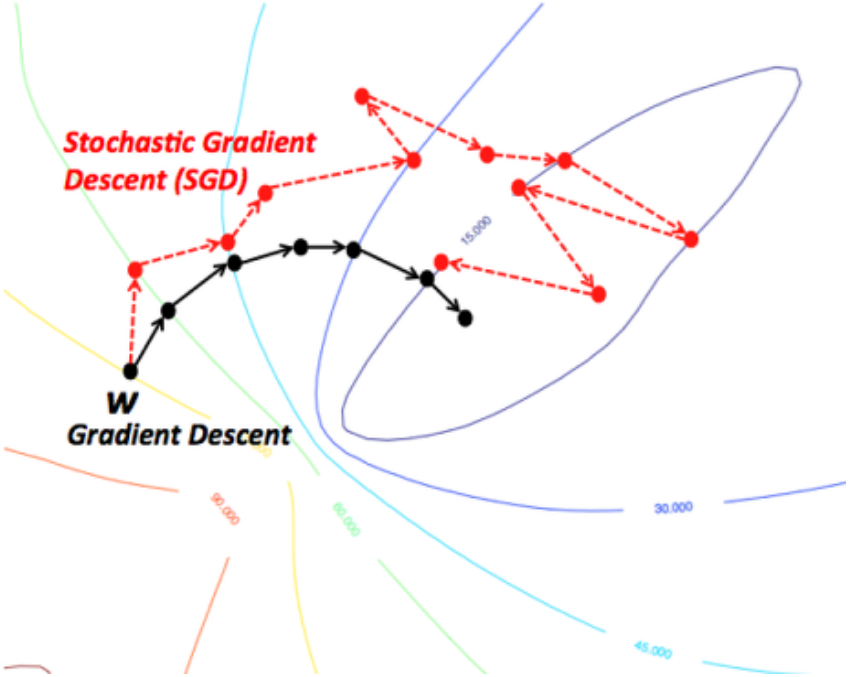


Questions: www.menti.com 2455 0800

THE UNIVERSITY OF SYDNEY

Stochastic Gradient Descent





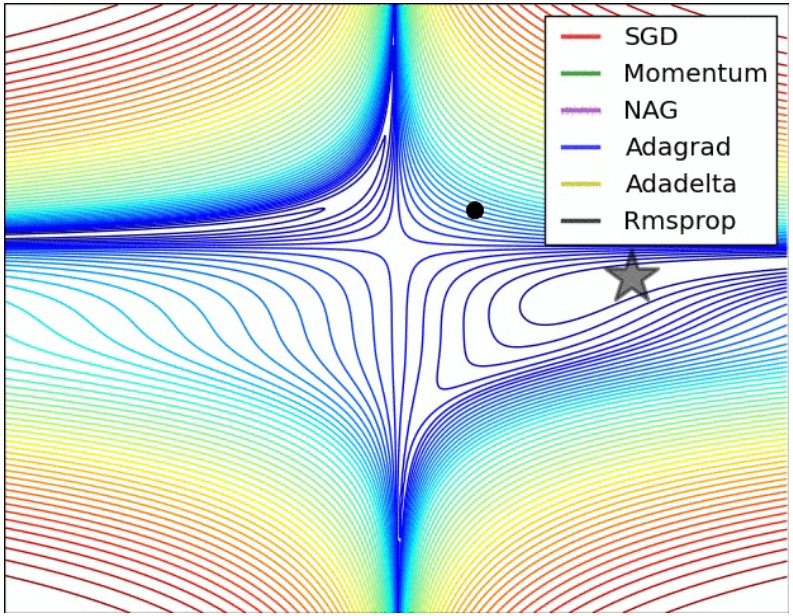
Gradient Descent
- *Calculate with all of the training data*
- *Move in the optimal direction, one step at a time*
- *Accurate but very slow*
- *e.g. 6 steps * 1 hour = 6 hours*

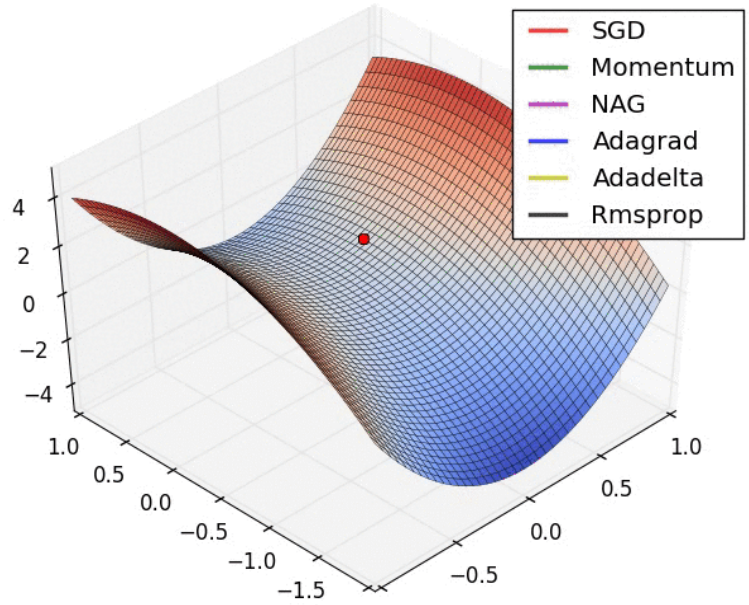
Stochastic Gradient Descent
- *Calculate with a mini-batch of data*
- *Move in a good direction, many small steps*
- *Can bounce around, but finish faster and/or at a better point*
- *e.g. 10 steps * 12 mins = 2 hours*

Questions: www.menti.com 2455 0800

THE UNIVERSITY OF SYDNEY

Finding the Optimal weight and bias – Gradient Descent





There are different types of Gradient descent optimisation algorithms:

Batch Gradient Descent, Stochastic Gradient Descent, Momentum, Adam, etc.

Questions: www.menti.com 2455 0800

THE UNIVERSITY OF SYDNEY

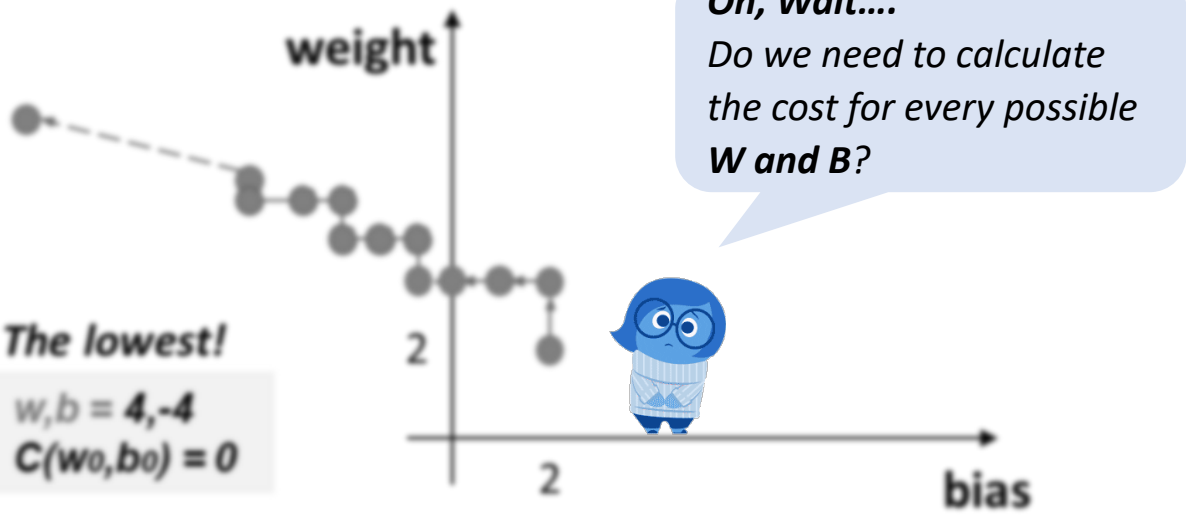
Backpropagation (weight update)

Deep Learning with Neural Network - Optimizer

weight *bias*

$$y = 4x - 4$$

x 🍎	y 🍌
1	0
5	16
6	20



Expensive to compute (hours or days)

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

$$w,b \in [-\infty, \infty]$$

Questions: www.menti.com 2455 0800

THE UNIVERSITY OF SYDNEY

Gradient!

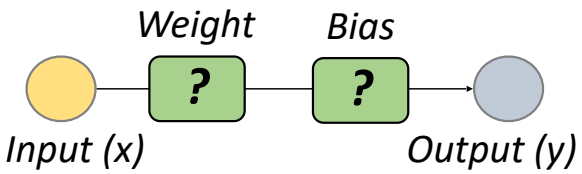
Deep Learning with Neural Network - Optimizer

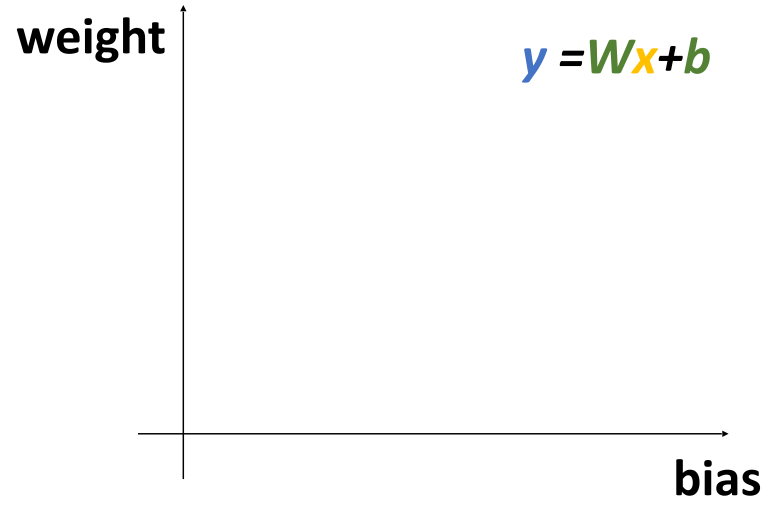
Actual Data

weight *bias*

$y = ? \; x + ?$

x	y
1	0
5	16
6	20





Should be used sparingly

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

$$w,b \in [-\infty,\infty]$$

Questions: www.menti.com 2455 0800

THE UNIVERSITY OF SYDNEY

Gradient!

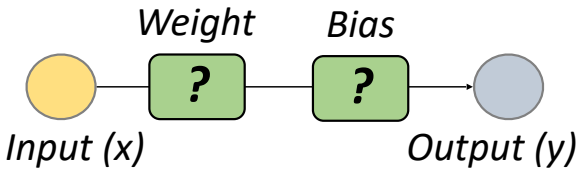
Deep Learning with Neural Network - Optimizer

Actual Data

weight *bias*

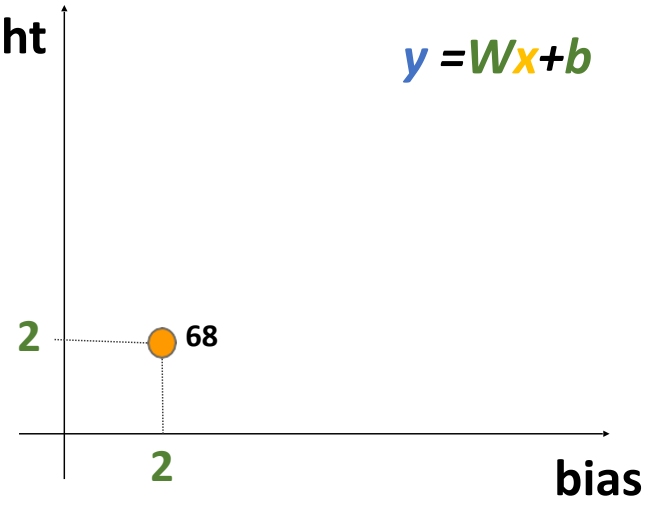
$y = ? \; x + ?$

x	y
1	0
5	16
6	20



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

$w,b = 2,2 \qquad C(w_0,b_0) = 68$



Questions: www.menti.com 2455 0800

THE UNIVERSITY OF SYDNEY

Gradient!

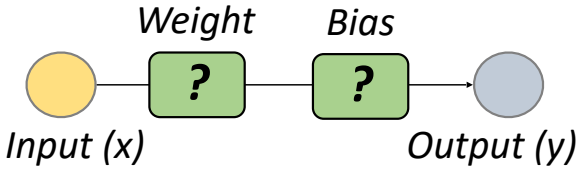
Deep Learning with Neural Network - Optimizer

Actual Data

weight *bias*

$y = ? \; x + ?$

x	y
1	0
5	16
6	20

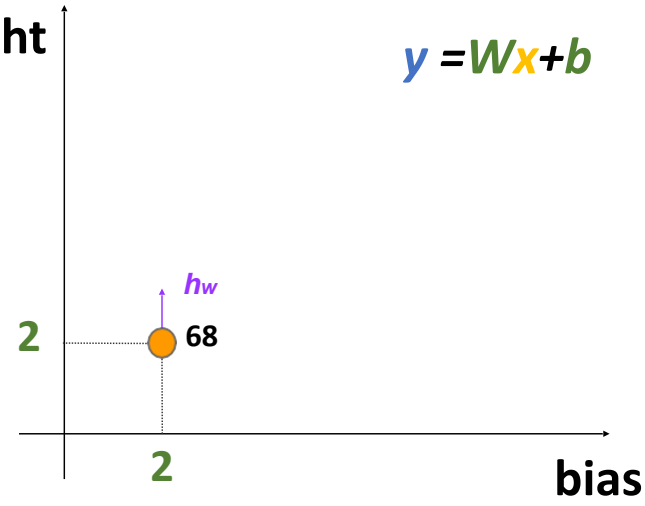


$\arg\min_{w,b \in [-\infty,\infty]} C(w,b)$

$w,b = 2,2 \qquad C(w_0,b_0) = 68$

$h_w = 1$

$C(w_0+h_w, b_0) = C(3,2) = 26$



Questions: www.menti.com 2455 0800

THE UNIVERSITY OF SYDNEY

Gradient!

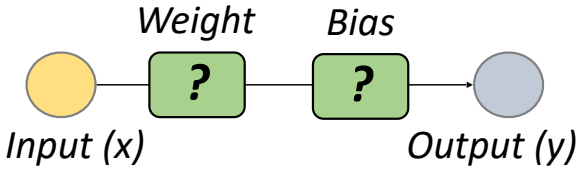
Deep Learning with Neural Network - Optimizer

Actual Data

weight *bias*

$y = ? \; x + ?$

x	y
1	0
5	16
6	20



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

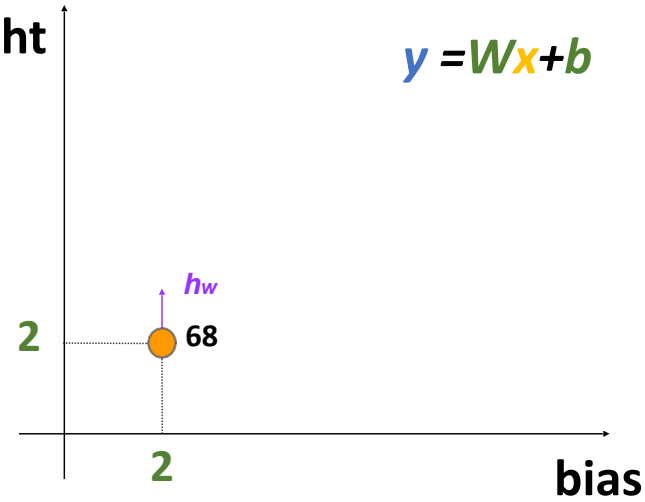
$w,b = 2,2 \qquad C(w_0,b_0) = 68$

$h_w = 1$

$C(w_0 + h_w, b_0) = C(3,2) = 26$

$$r = \frac{(C(w_0+h_w, b_0) - C(w_0,b_0))}{1}$$

$$r = \frac{(C(2+1, 2) - C(2,2)) = -42}{1}$$



Questions: www.menti.com 2455 0800

THE UNIVERSITY OF SYDNEY

Gradient!

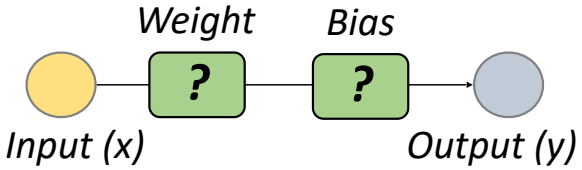
Deep Learning with Neural Network - Optimizer

Actual Data

weight *bias*

$y = ? \; x + ?$

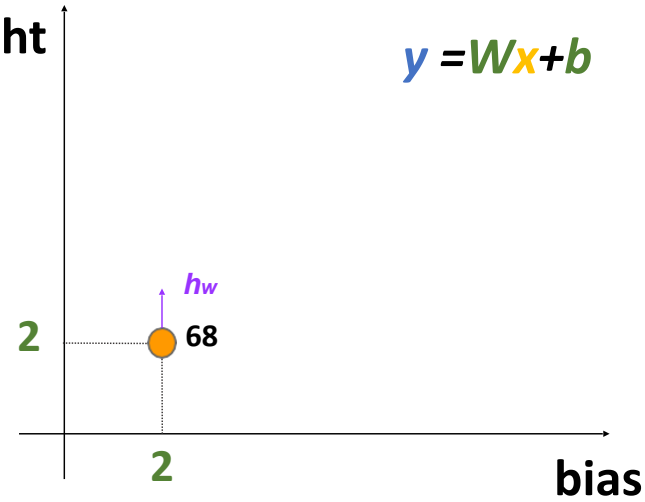
x	y
1	0
5	16
6	20



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

$w,b = 2,2 \qquad C(w_0,b_0) = 68$

$h_w = 1$, $r = -42$
$h_w = 0.1$, $r = -98$
$h_w = 0.01$, $r = -104$
$h_w = 0.001$, $r = -104$



Questions: www.menti.com 2455 0800

THE UNIVERSITY OF SYDNEY

Gradient!

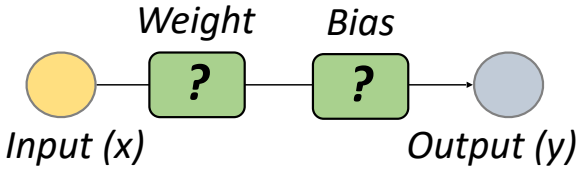
Deep Learning with Neural Network - Optimizer

Actual Data

weight *bias*

$y = ? \; x + ?$

x	y
1	0
5	16
6	20

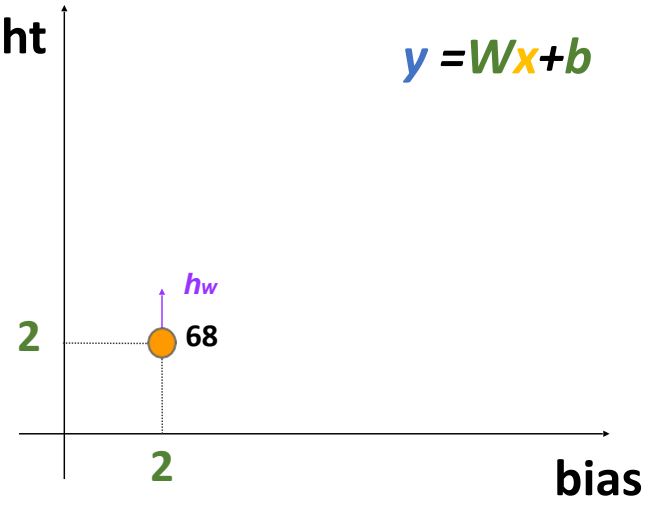


$\arg\min_{w,b \in [-\infty,\infty]} C(w,b)$

$w,b = 2,2 \qquad C(w_0,b_0) = 68$

$h_w = 1$, $r = -42$
$h_w = 0.1$, $r = -98$
$h_w = 0.01$, $r = -104$
$h_w = 0.001$, $r = -104$

$h_w \rightarrow 0$, $r = \frac{\partial C}{\partial w}(w_0,b_0)$



Questions: www.menti.com 2455 0800

THE UNIVERSITY OF SYDNEY

Deep Learning with Neural Network - Optimizer

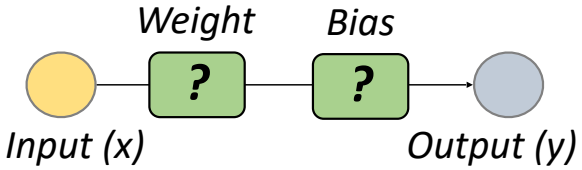
Gradient!

Actual Data

weight *bias*

$y = ? \; x + ?$

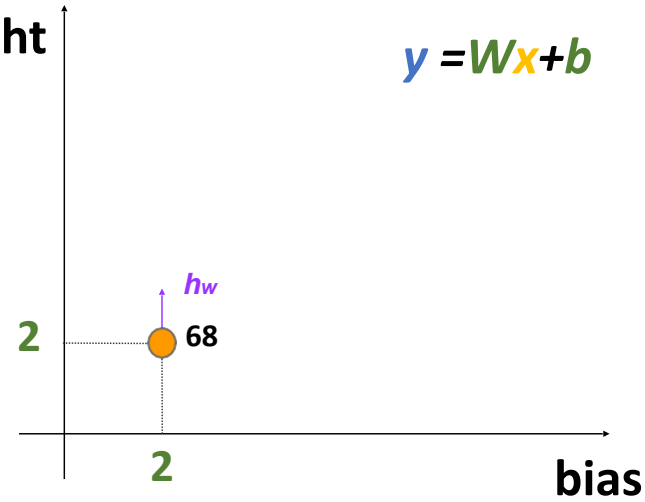
x	y
1	0
5	16
6	20



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

$w,b = 2,2 \qquad C(w_0,b_0) = 68$

$$\frac{\partial C}{\partial w} = \frac{\partial \sum_n (y_n - \hat{y}_n)^2}{\partial w}$$



Questions: www.menti.com 2455 0800

THE UNIVERSITY OF SYDNEY

Deep Learning with Neural Network - Optimizer

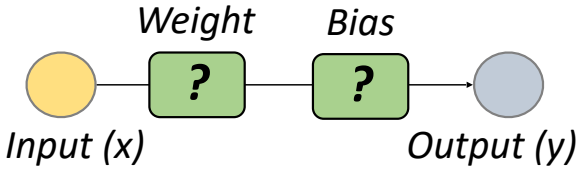
Gradient!

Actual Data

weight *bias*

$y = ? \; x + ?$

x	y
1	0
5	16
6	20

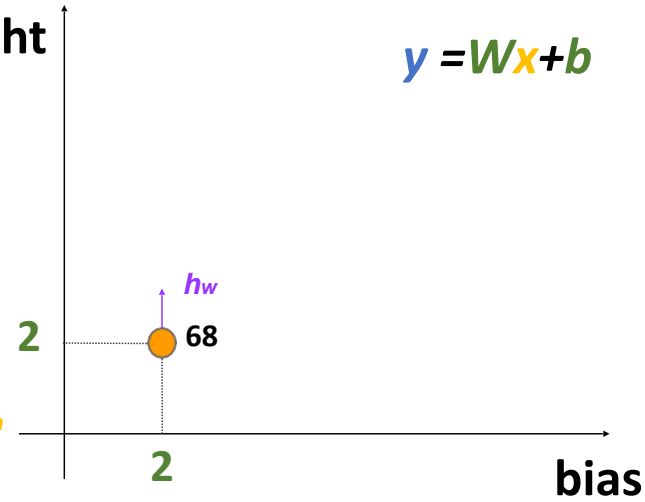


$\arg\min_{w,b \in [-\infty,\infty]} C(w,b)$

$w,b = 2,2 \qquad C(w_0,b_0) = 68$

$$\frac{\partial C}{\partial w} = \frac{\partial \sum_n (y_n - \hat{y}_n)^2}{\partial w} = \sum_n 2(y_n - \hat{y}_n)x_n$$

$h_w \rightarrow 0 \; , r = \frac{\partial C}{\partial w}(w_0,b_0)$



Questions: www.menti.com 2455 0800

THE UNIVERSITY OF SYDNEY

Deep Learning with Neural Network - Optimizer

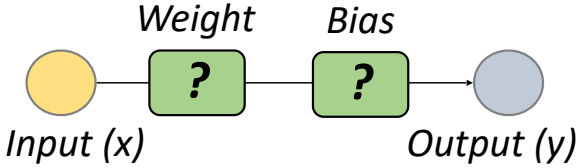
Gradient!

Actual Data

weight *bias*

$y = ? \; x + ?$

x	y
1	0
5	16
6	20



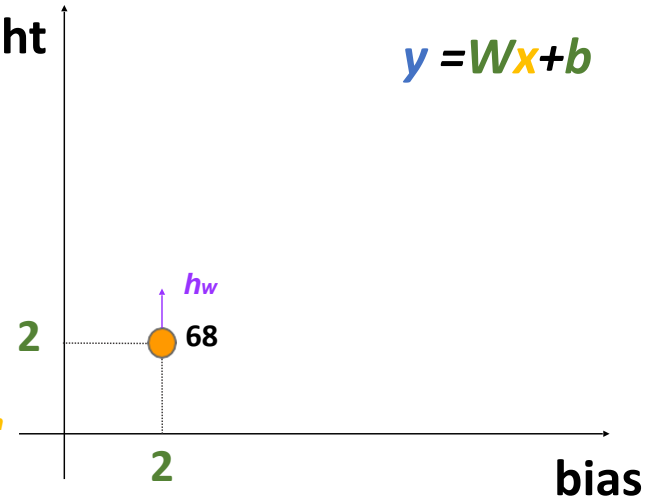
$\arg\min_{w,b \in [-\infty,\infty]} C(w,b)$

$w,b = 2,2 \qquad C(w_0,b_0) = 68$

$$\frac{\partial C}{\partial w} = \frac{\partial \sum_n (y_n - \hat{y}_n)^2}{\partial w} = \sum_n 2(y_n - \hat{y}_n)x_n$$

Chain rule

$$\frac{\partial C}{\partial w} = \frac{\partial C}{\partial y} \; x \; \frac{\partial y}{\partial w}$$



Questions: www.menti.com 2455 0800

THE UNIVERSITY OF SYDNEY

Deep Learning with Neural Network - Optimizer

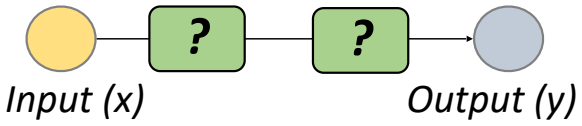
Gradient!

Actual Data

weight *bias*

$y = ? \; x + ?$

x 🍎	y 🍌
1	0
5	16
6	20



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

$w,b = 2,2 \qquad C(w_0,b_0) = 68$

$$\frac{\partial C}{\partial w} = \frac{\partial \sum_n (y_n - \hat{y}_n)^2}{\partial w} = \sum_n 2(y_n - \hat{y}_n)x_n$$

$h_w \rightarrow \quad 0 \; r = \frac{\partial C}{\partial w}(w_0,b_0) = -104$

weight *bias*

$y = 2 \; x + 2$

	predicted	*actual*		
x 🍎	ŷ 🍌	y 🍌		2x
1	4	0	-4	-8
5	12	16	4	40
6	14	20	6	72

Questions: www.menti.com 2455 0800

THE UNIVERSITY OF SYDNEY

Deep Learning with Neural Network - Optimizer

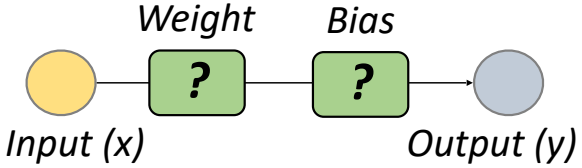
Gradient!

Actual Data

weight *bias*

$y = ? \; x + ?$

x	y
1	0
5	16
6	20

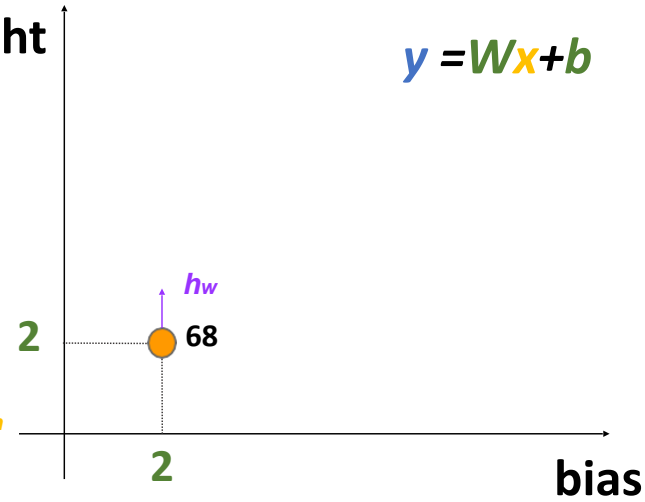


$$\arg\min_{w,b \in [-\infty,\infty]} C(w,b)$$

$w,b = 2,2 \qquad C(w_0,b_0) = 68$

$$\frac{\partial C}{\partial w} = \frac{\partial \sum_n (y_n - \hat{y}_n)^2}{\partial w} = \sum_n 2(y_n - \hat{y}_n)x_n$$

$$\frac{\partial C}{\partial b} = \frac{\partial \sum_n (y_n - \hat{y}_n)^2}{\partial b} = \sum_n 2(y_n - \hat{y}_n)$$



Questions: www.menti.com 2455 0800

THE UNIVERSITY OF SYDNEY

Gradient!

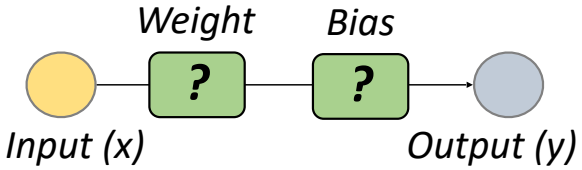
Deep Learning with Neural Network - Optimizer

Actual Data

weight *bias*

$y = ? \; x + ?$

x 🍎	y 🍌
1	0
5	16
6	20



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

$w,b = 2,2 \qquad C(w_0,b_0) = 68$

$h_w \rightarrow 0 \; r = \frac{\partial C}{\partial w}(w_0,b_0) = -104$

$h_b \rightarrow 0 \; r = \frac{\partial C}{\partial b}(w_0,b_0) = -12$

weight *bias*

$y = 2 \; x + 2$

	predicted	*actual*		
x 🍎	ŷ 🍌	y 🍌		2
1	4	0	-4	-8
5	12	16	4	8
6	14	20	6	12

Questions: www.menti.com 2455 0800

Deep Learning for NLP

Deep Learning with Neural Network - Optimizer

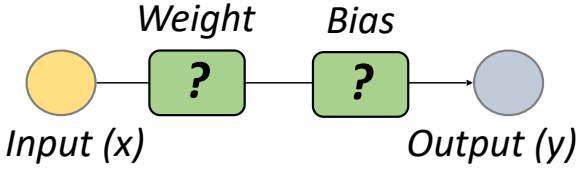
Gradient!

Actual Data

weight *bias*

$y = ? \; x + ?$

x 🍎	y 🍌
1	0
5	16
6	20

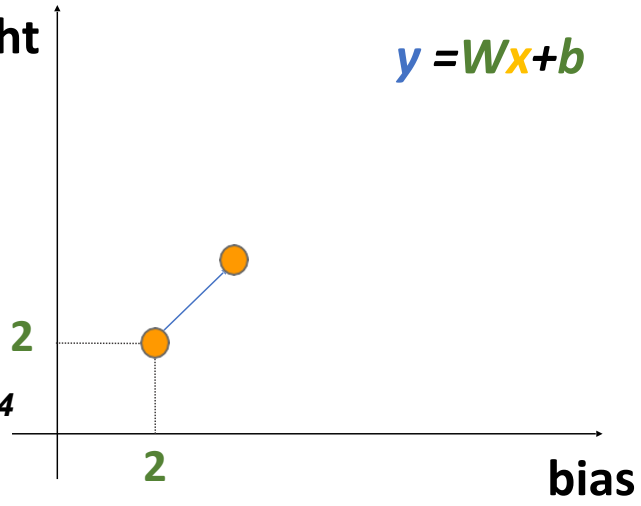


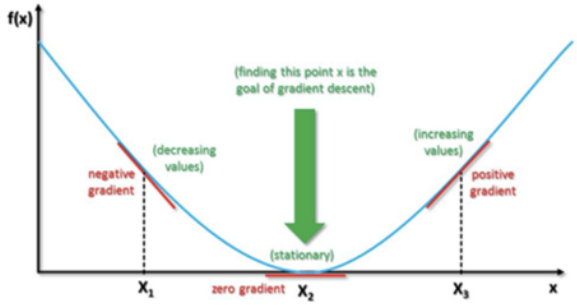
$\arg\min_{w,b \in [-\infty,\infty]} C(w,b)$

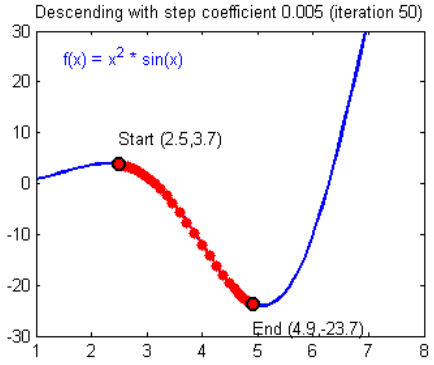
$w,b = 2,2 \qquad C(w_0,b_0) = 68$

$h_w \rightarrow 0 \; r = \frac{\partial C}{\partial w}(w_0,b_0) = -104$

$h_b \rightarrow 0 \; r = \frac{\partial C}{\partial b}(w_0,b_0) = -12$







Questions: www.menti.com 2455 0800

Deep Learning for NLP

Deep Learning with Neural Network

Data

x 🍎	y 🍌
1	0
5	16
6	20

Model

weight *bias*

$y = ? \; x + ?$

Cost

$C(w,b) = \sum(y_n - \hat{y}_n)$

$n \in \{0,1,2\}$

Optimizer

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

$w,b \in [-\infty,\infty]$

System

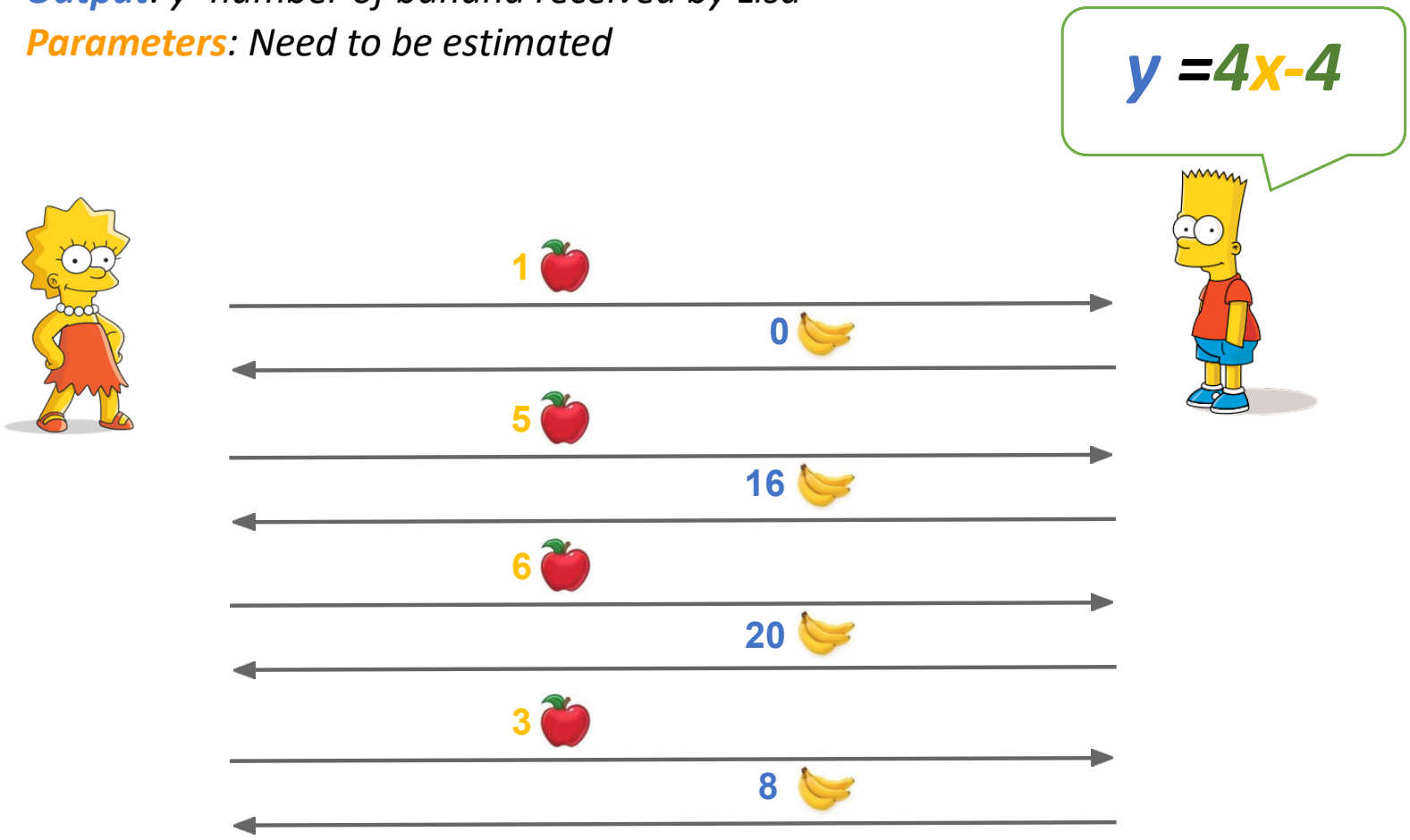
weight *bias*

$y = 4 \; x - 4$

Questions: www.menti.com 2455 0800

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



Questions: www.menti.com 2455 0800

Deep Learning for NLP

Deep Learning with Neural Network

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

pixel(1,1) *pixel(1,2)*

*Millions of **Parameters***

*Millions of **Samples***

Questions: www.menti.com 2455 0800

3 Deep Learning for NLP

Deep Learning with Neural Network

$$y = \underbrace{w_1 x_1}_{\text{Vector1}} + \underbrace{w_2 x_2}_{\text{Vector2}} + w_3 x_3 + w_4 x_4 + \ldots + w_n x_n + b$$

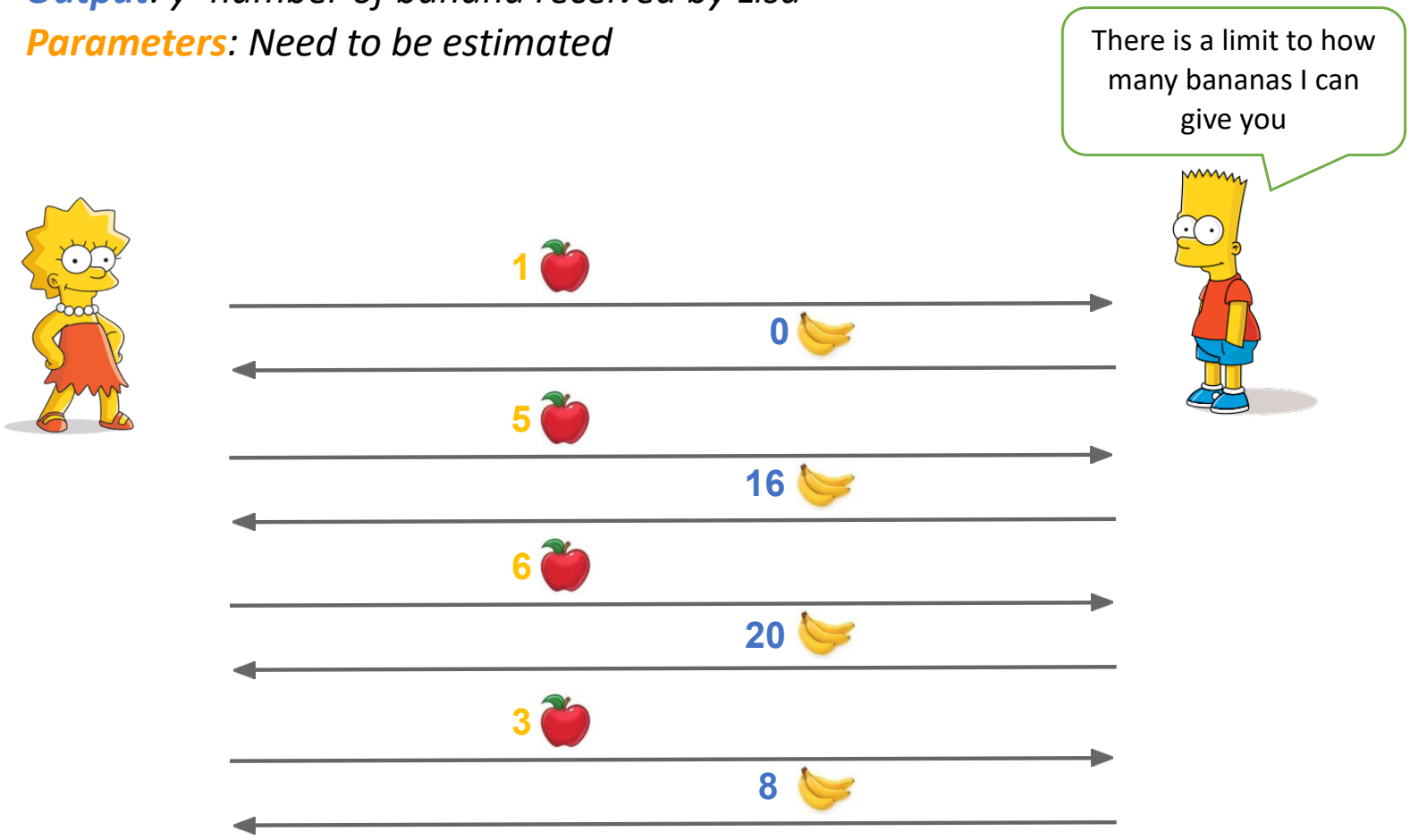
*Millions of **Parameters***

*Millions of **Samples***

Questions: www.menti.com 2455 0800

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



Questions: www.menti.com 2455 0800

Deep Learning with Neural Network

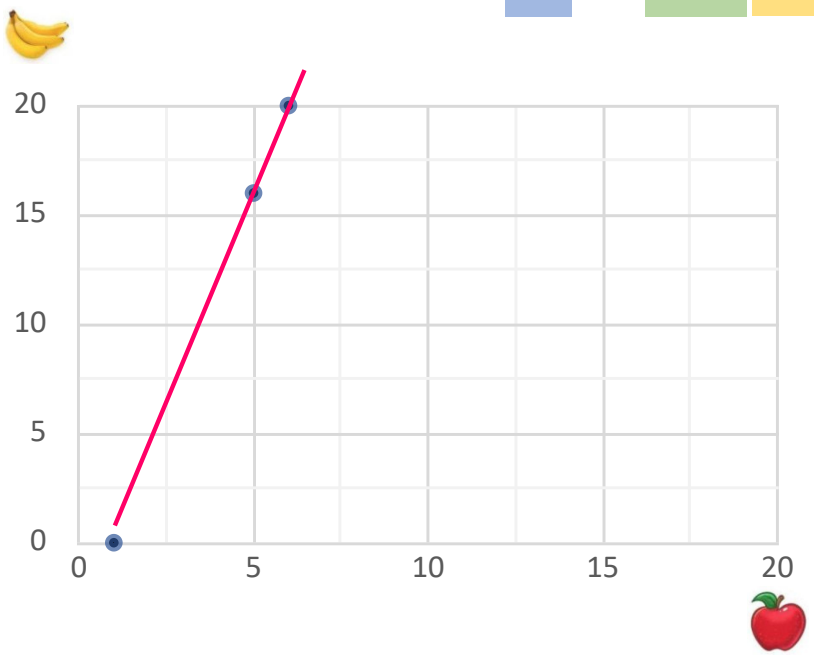
Nonlinear Neural Network

weight *bias*

$$y = 4x - 4$$

Data

x 🍎	y 🍌
1	0
5	16
6	20



Questions: www.menti.com 2455 0800

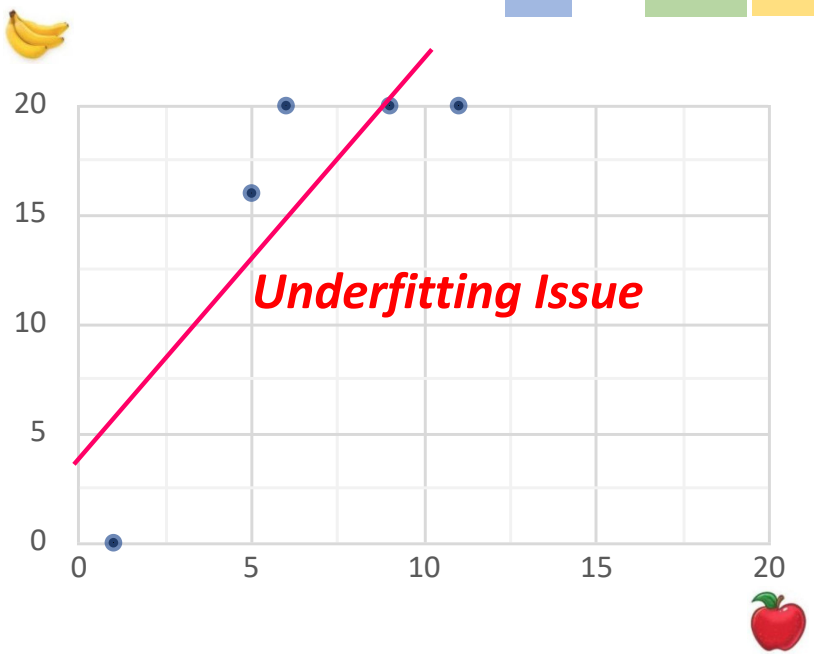
Deep Learning with Neural Network

Nonlinear Neural Network

$$y = \underbrace{2}_{\text{weight}} x + \underbrace{3}_{\text{bias}}$$

Data

x 🍎	y 🍌
1	0
5	16
6	20
9	20
11	20



Questions: www.menti.com 2455 0800

THE UNIVERSITY OF SYDNEY

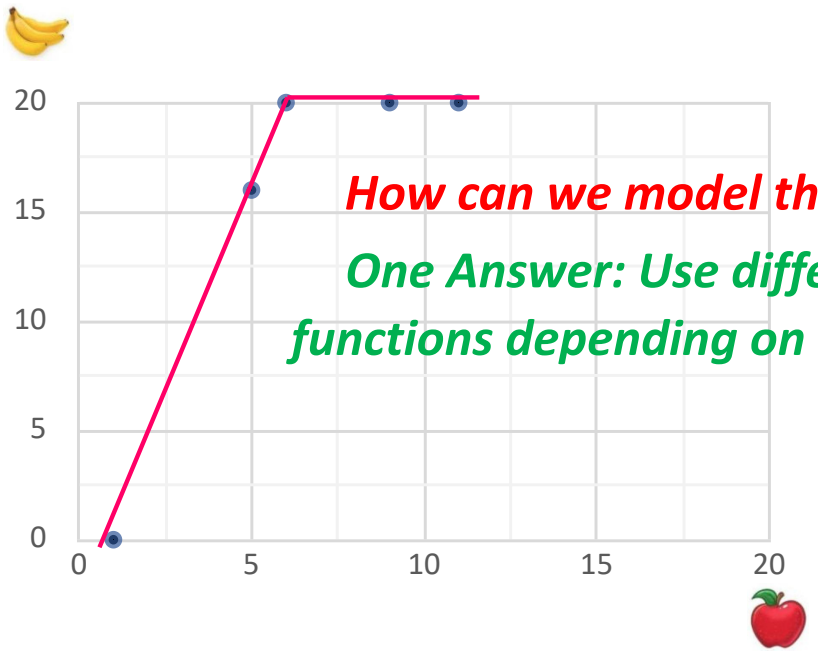
Deep Learning with Neural Network

Nonlinear Neural Network

y = ???

Data

x 🍎	y 🍌
1	0
5	16
6	20
9	20
11	20



How can we model this?

One Answer: Use different linear functions depending on the value of x

Questions: www.menti.com 2455 0800

THE UNIVERSITY OF SYDNEY

Deep Learning with Neural Network

Nonlinear Neural Network

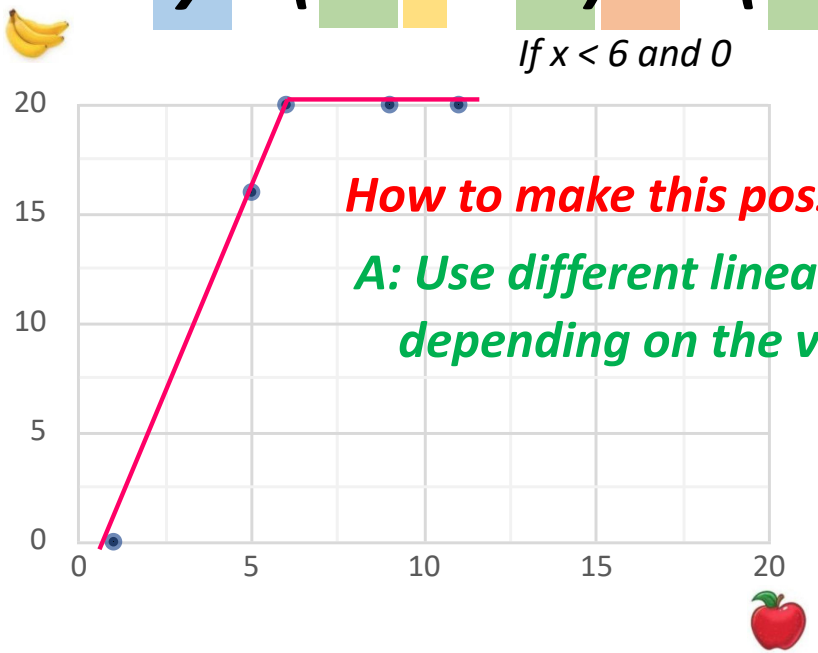
Data

x 🍎	y 🍌
1	0
5	16
6	20
9	20
11	20

weight *bias* *weight* *bias*

$$y = (w_1 x + b_1)s_1 + (w_2 x + b_2)s_2$$

If x < 6 and 0 *If x >= 6 and 0*



How to make this possible?

A: Use different linear functions depending on the value of x

Questions: www.menti.com 2455 0800

Deep Learning with Neural Network

Nonlinear Neural Network

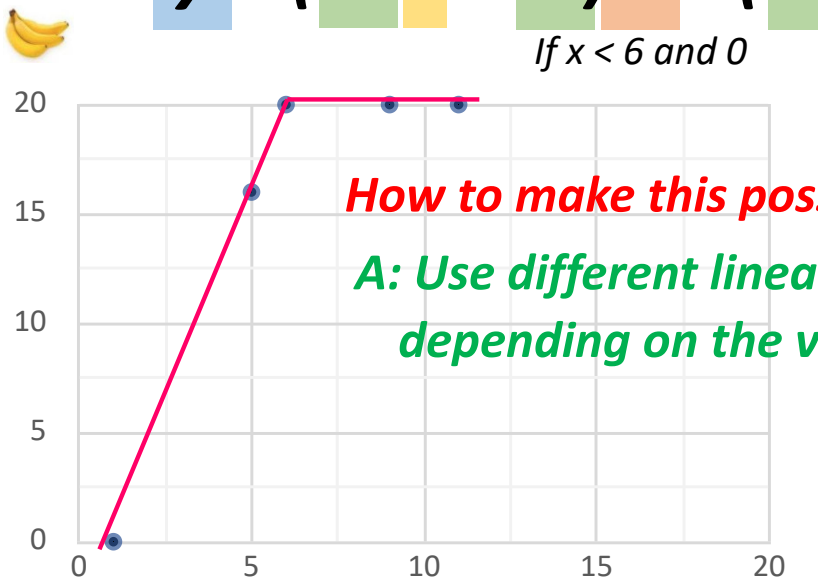
Data

x 🍎	y 🍌
1	0
5	16
6	20
9	20
11	20

weight *bias* *weight* *bias*

$$y = (w_1 x + b_1)s_1 + (w_2 x + b_2)s_2$$

If x < 6 and 0 *If x >= 6 and 0*



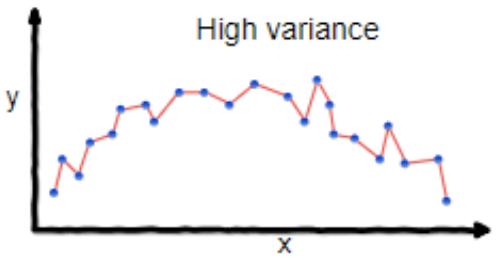
How to make this possible?

A: Use different linear functions depending on the value of x

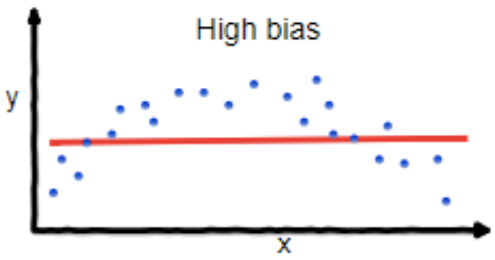
$W_1 = 4,\ b_1 = -4,\ W_2 = 0,\ b_2 = 20$ 🍎

Questions: www.menti.com 2455 0800

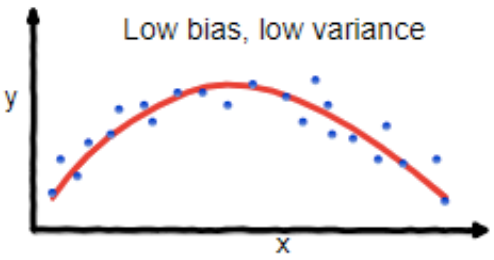
Deep Learning with Neural Network



overfitting



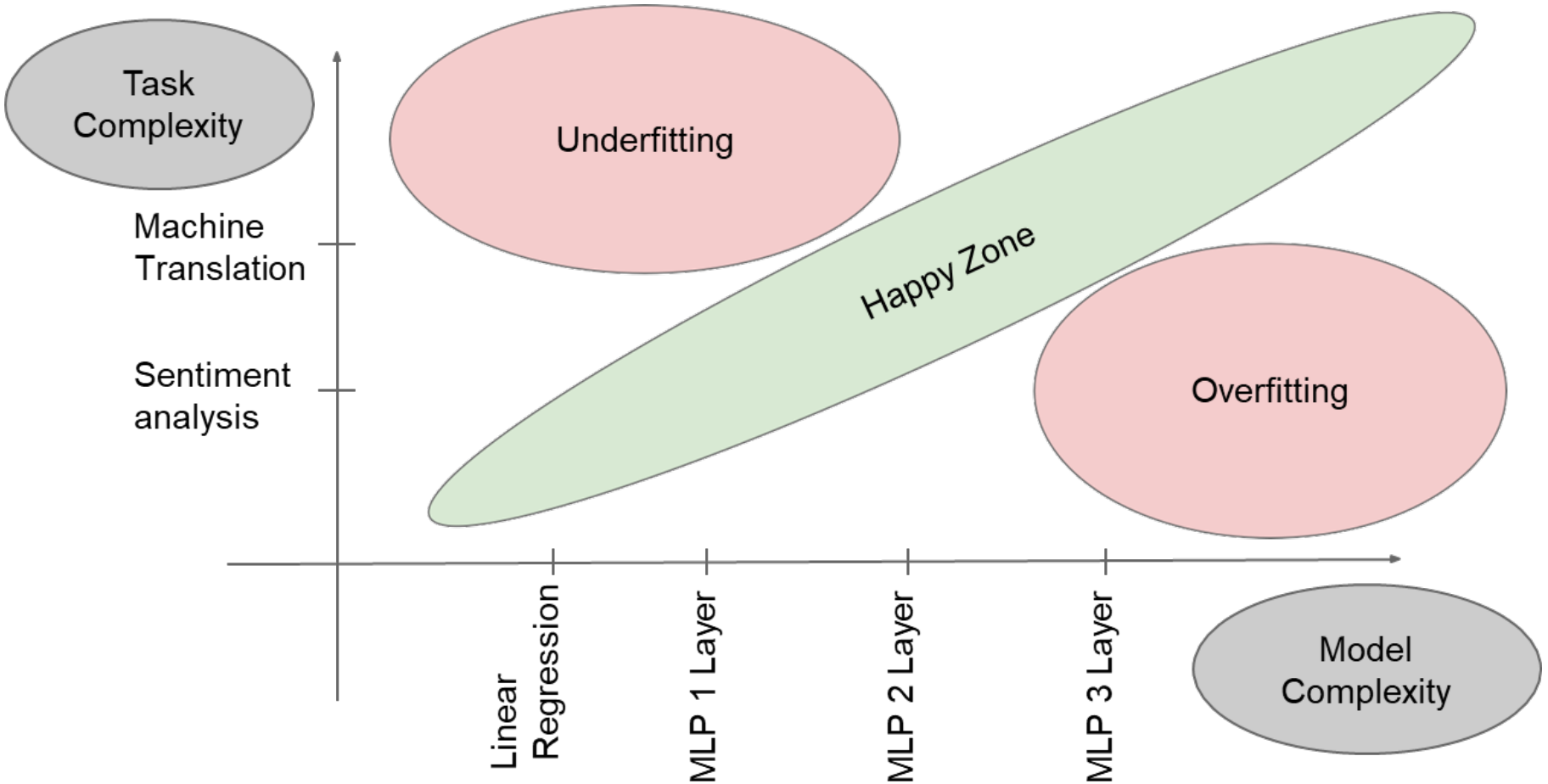
underfitting



Good balance

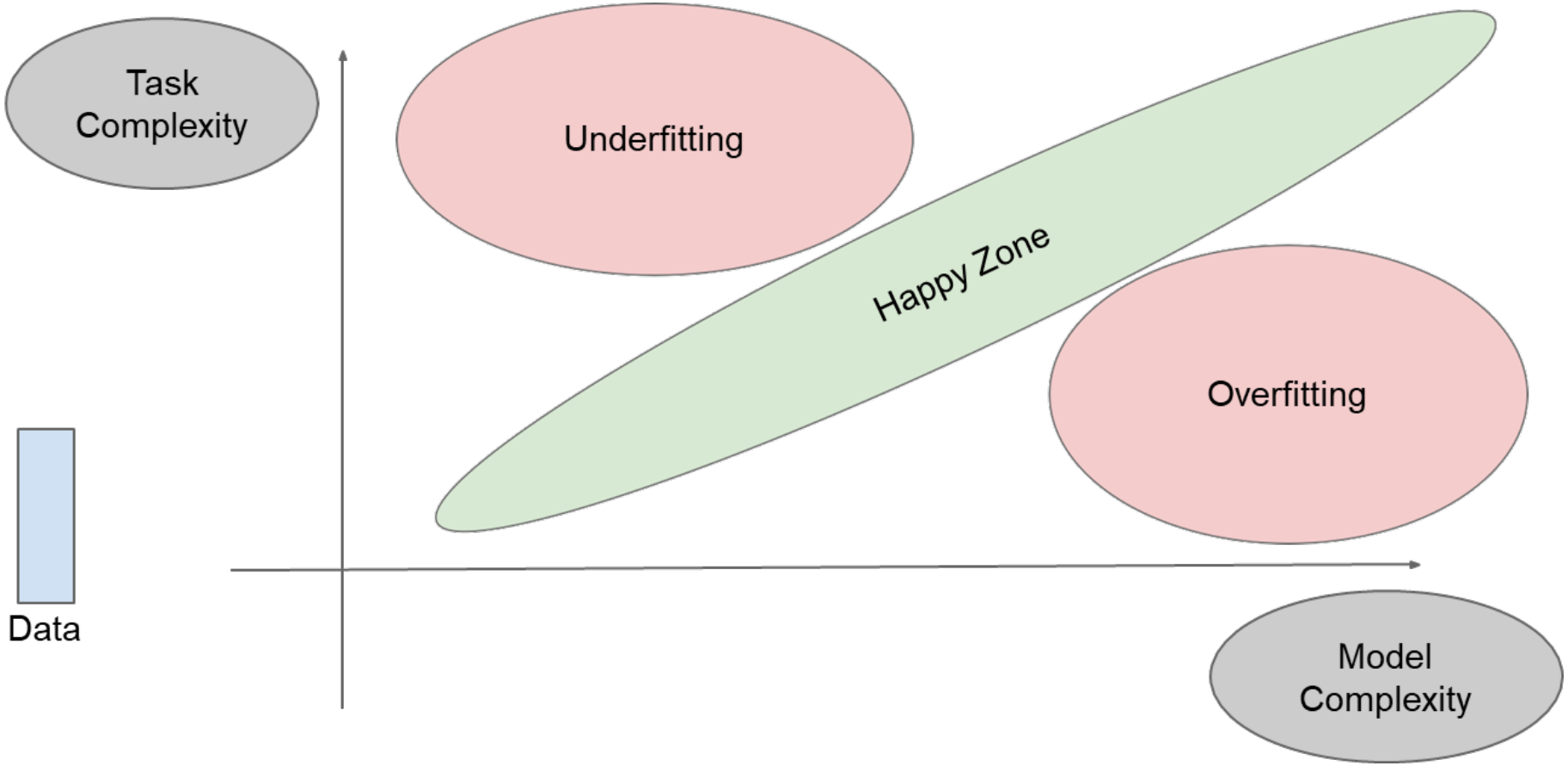
Questions: www.menti.com 2455 0800

3 Deep Learning for NLP



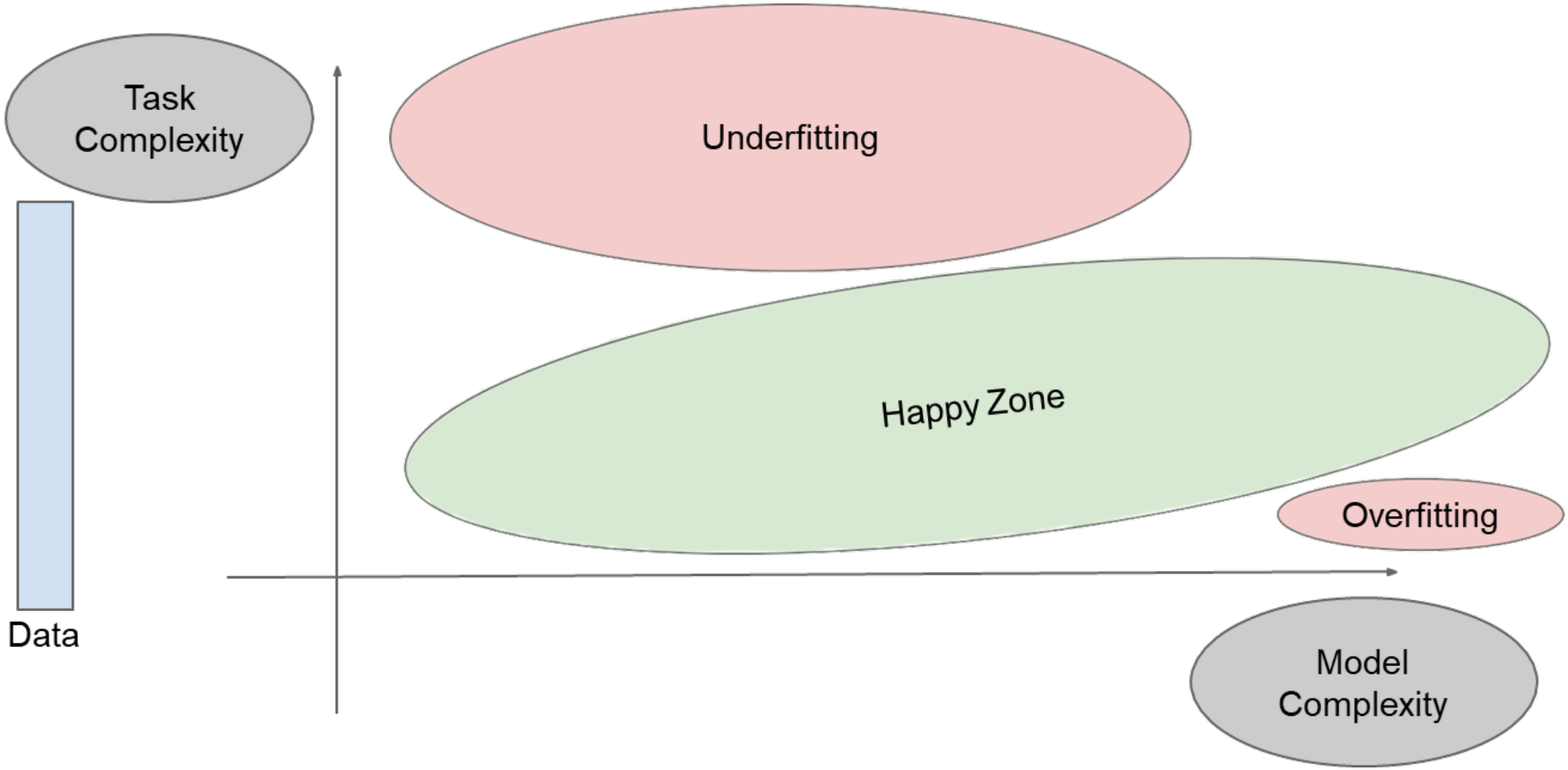
Questions: www.menti.com 2455 0800

3 Deep Learning for NLP



Questions: www.menti.com 2455 0800

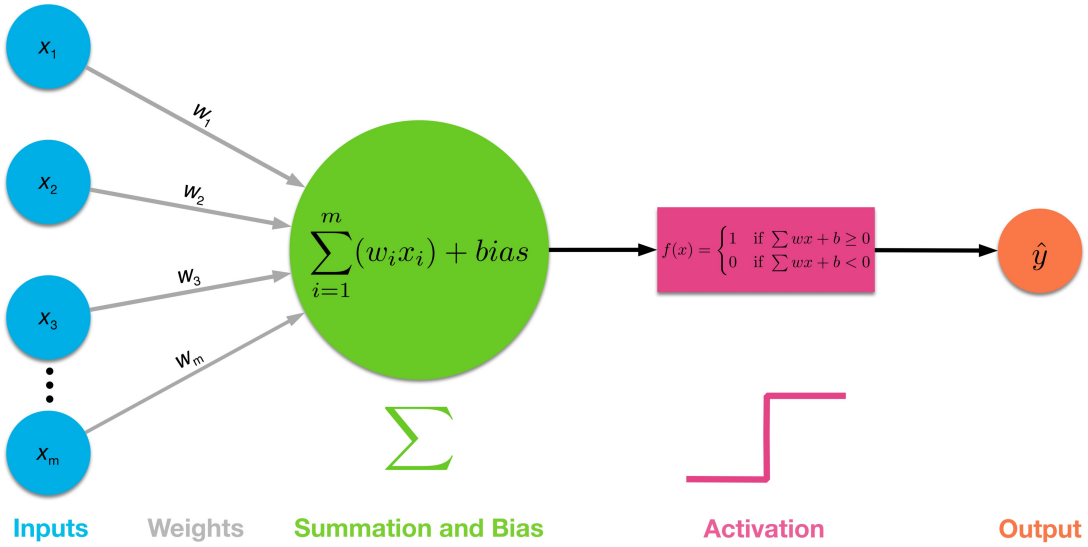
3 Deep Learning for NLP

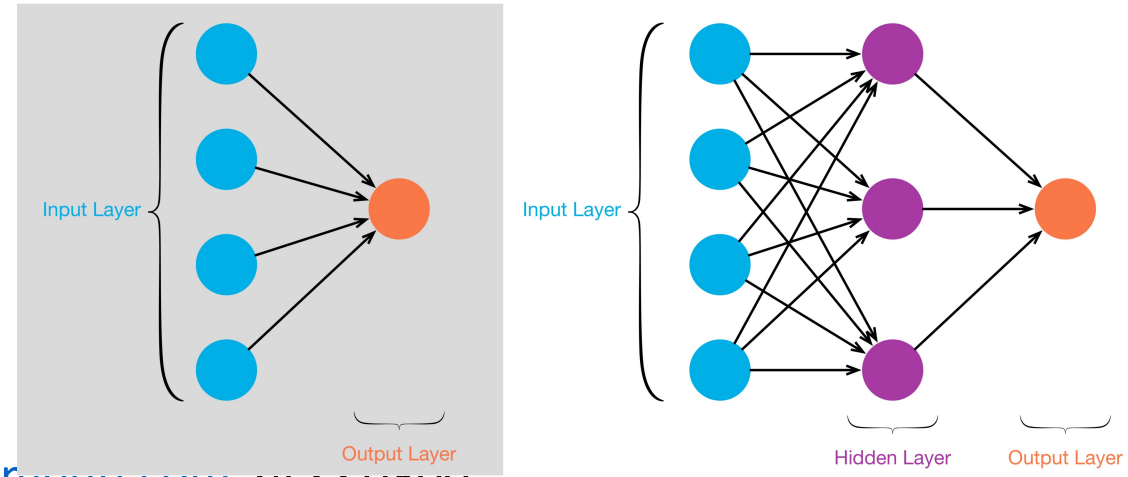


Questions: www.menti.com 2455 0800

Deep Learning for NLP

Single Neuron VS Multilayer



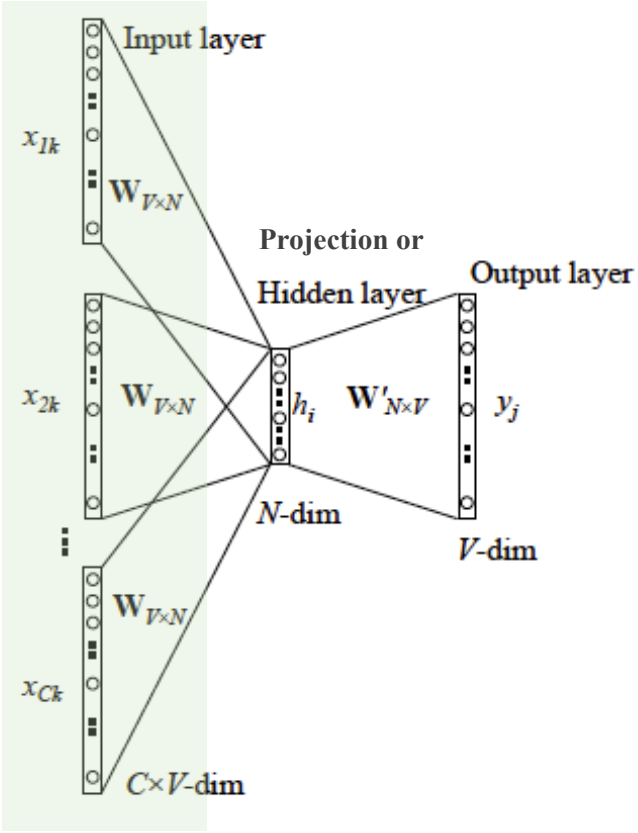


Questions: www.menti.com 2433 0800

CBOW – Neural Network Architecture (ReCAP with Optimizer)

Predict centre word from (bag of) context words.

Summary of CBOW Training (Review your understanding with equations)



Uses context words (2m, based on window size =m) as input of the Word2Vec-CBOW model.

$(\boldsymbol{x}^{c-m}, \boldsymbol{x}^{c-m+1}, \ldots, \boldsymbol{x}^{c-1}, \boldsymbol{x}^{c+1}, \ldots, \boldsymbol{x}^{c+m-1}, \boldsymbol{x}^{c+m}) \in \mathbb{R}^{|V|}$

Has two Parameter Matrices:

1) Parameter Matrix (from Input Layer to Hidden/Projection Layer)

$\mathbf{W} \in \mathbb{R}^{V\mathrm{x}N}$

2) Parameter Matrix (to Output Layer)

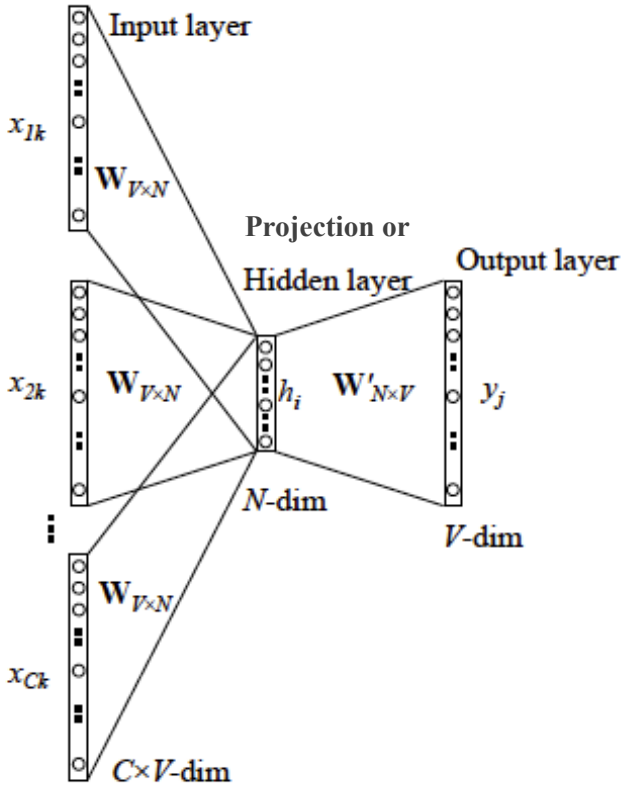
$\mathbf{W}' \in \mathbb{R}^{N\mathrm{x}V}$

Questions: www.menti.com 2455 0800

CBOW – Neural Network Architecture (ReCAP with Optimizer)

Predict centre word from (bag of) context words.

Summary of CBOW Training (Review your understanding with equations)



Initial words are looked up in $\mathbf{W}_{\mathbf{VxN}}$ to get a 1 x N (embedded word) vector.

$$(\boldsymbol{v}_{c-m} = \mathrm{W}x^{c-m}, \ldots, \boldsymbol{v}_{c+m} = \mathrm{W}x^{c+m}) \in \mathbb{R}^{\boldsymbol{n}}$$

Average those 2m embedded vectors to calculate a single vector.

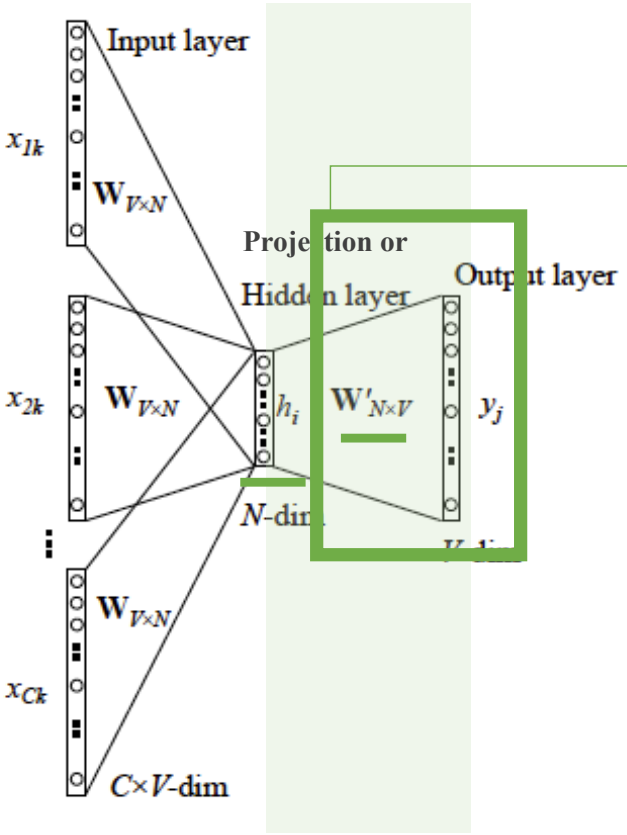
$$\hat{v} = \frac{\boldsymbol{v}_{c-m} + \boldsymbol{v}_{c-m+1} + \ \ldots + \boldsymbol{v}_{c+m}}{2m}$$

Questions: www.menti.com 2455 0800

CBOW – Neural Network Architecture (ReCAP with Optimizer)

Predict centre word from (bag of) context words.

Summary of CBOW Training (Review your understanding with equations)



Calculate the similarity between the mean vector and each of the vectors in W2

$$z = \mathbf{W}'\hat{v} \in \mathbb{R}^{|V|}$$

Convert this to a probability using Softmax

$$\hat{y} = softmax\,(z) \in \mathbb{R}^{|V|}$$

Train the parameter matrix using objective function.

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

* We are minimising the value

Only one term is non-zero, the one for the true word:

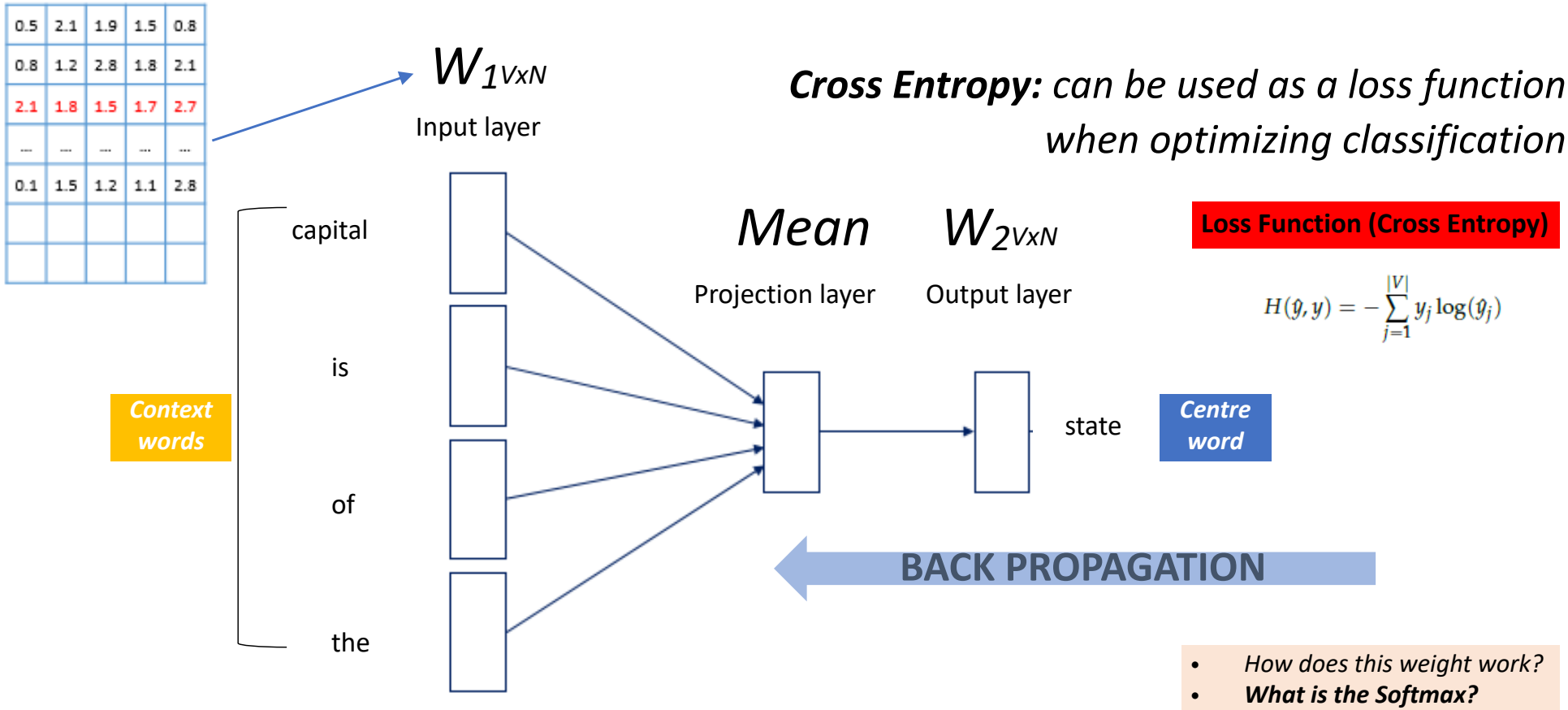
$$\mathrm{H}\left(\hat{y},\ y\right) = -y_j\, log(\hat{y}_j)$$

Questions: www.menti.com 2455 0800

CBOW – Neural Network Architecture (ReCAP with Optimizer)

Predict centre word from (bag of) context words

Sentence: "Sydney is the state capital of NSW"



Cross Entropy: *can be used as a loss function when optimizing classification*

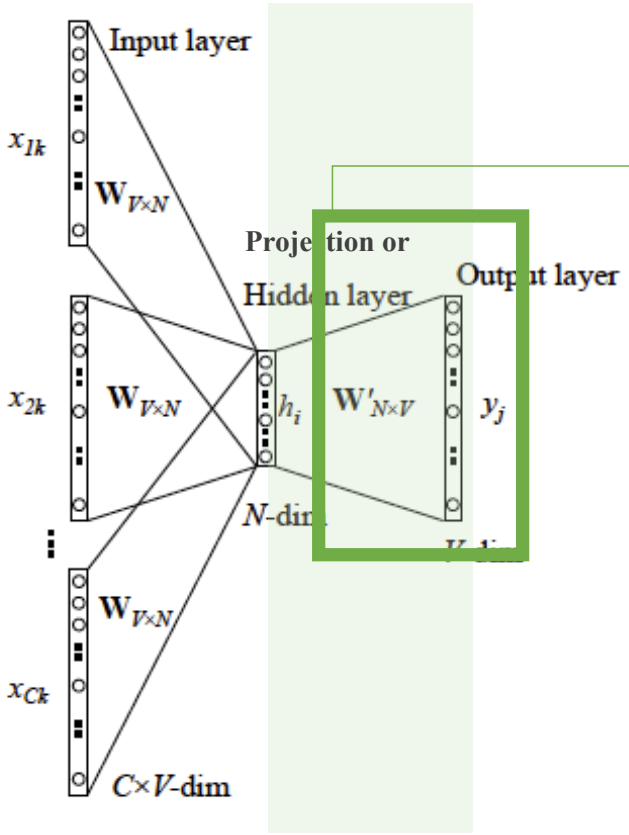
- *How does this weight work?*
- ***What is the Softmax?***
- *What is the Cross Entropy?*
- *How to Train the model?*

Questions: www.menti.com 2455 0800

CBOW – Neural Network Architecture (ReCAP with Optimizer)

Predict centre word from (bag of) context words.

Summary of CBOW Training (Review your understanding with equations)



Calculate the similarity between the mean vector and each of the vectors in W2

$$\boldsymbol{z} = \mathbf{W}'\hat{v} \in \mathbb{R}^{|V|}$$

Convert this to a probability using Softmax

$$\hat{y} = softmax\,(\boldsymbol{z}) \in \mathbb{R}^{|V|}$$

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

$$\frac{e^{\hat{y}_i}}{\sum_{j=1}^{|V|} e^{\hat{y}_j}}$$

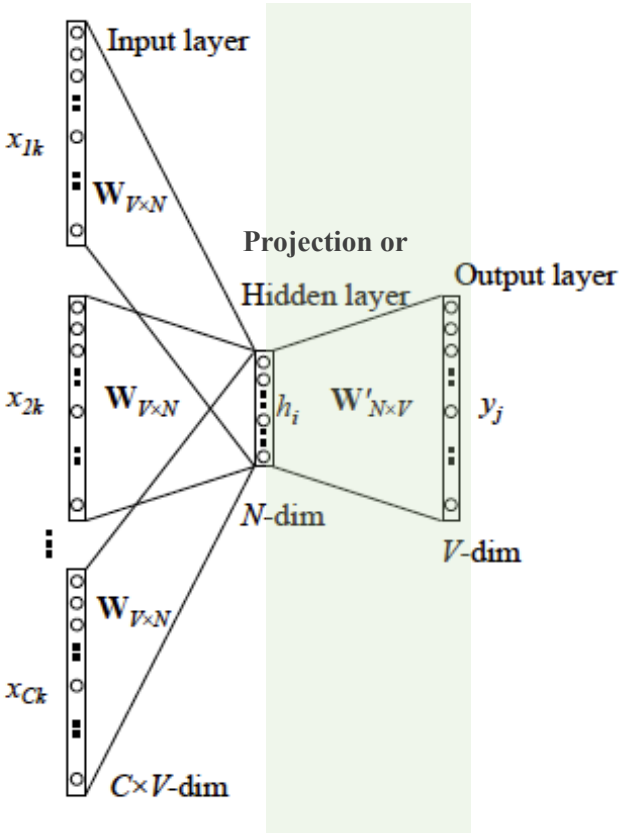
- Exponentiate to make positive
- Dividing by $\sum_{j=1}^{|V|} e^{\hat{y}_j}$ normalizes the vector ($\sum_{j=1}^{n} \hat{y}_j = 1$) to give probability

Questions: www.menti.com 2455 0800

CBOW – Neural Network Architecture (ReCAP with Optimizer)

Predict centre word from (bag of) context words.

Summary of CBOW Training (Review your understanding with equations)



Calculate the similarity between the mean vector and each of the vectors in W2

$$\boldsymbol{z} = \mathbf{W}'\hat{v} \in \mathbb{R}^{|V|}$$

Convert this to a probability using Softmax

$$\hat{y} = softmax\,(\boldsymbol{z}) \in \mathbb{R}^{|V|}$$

Train the parameter matrix using objective function.

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

Cross Entropy

* We are minimising the value

Only one term is non-zero, the one for the true word:

$$\mathrm{H}\left(\hat{y},\ y\right) = -y_j\, log(\hat{y}_j)$$

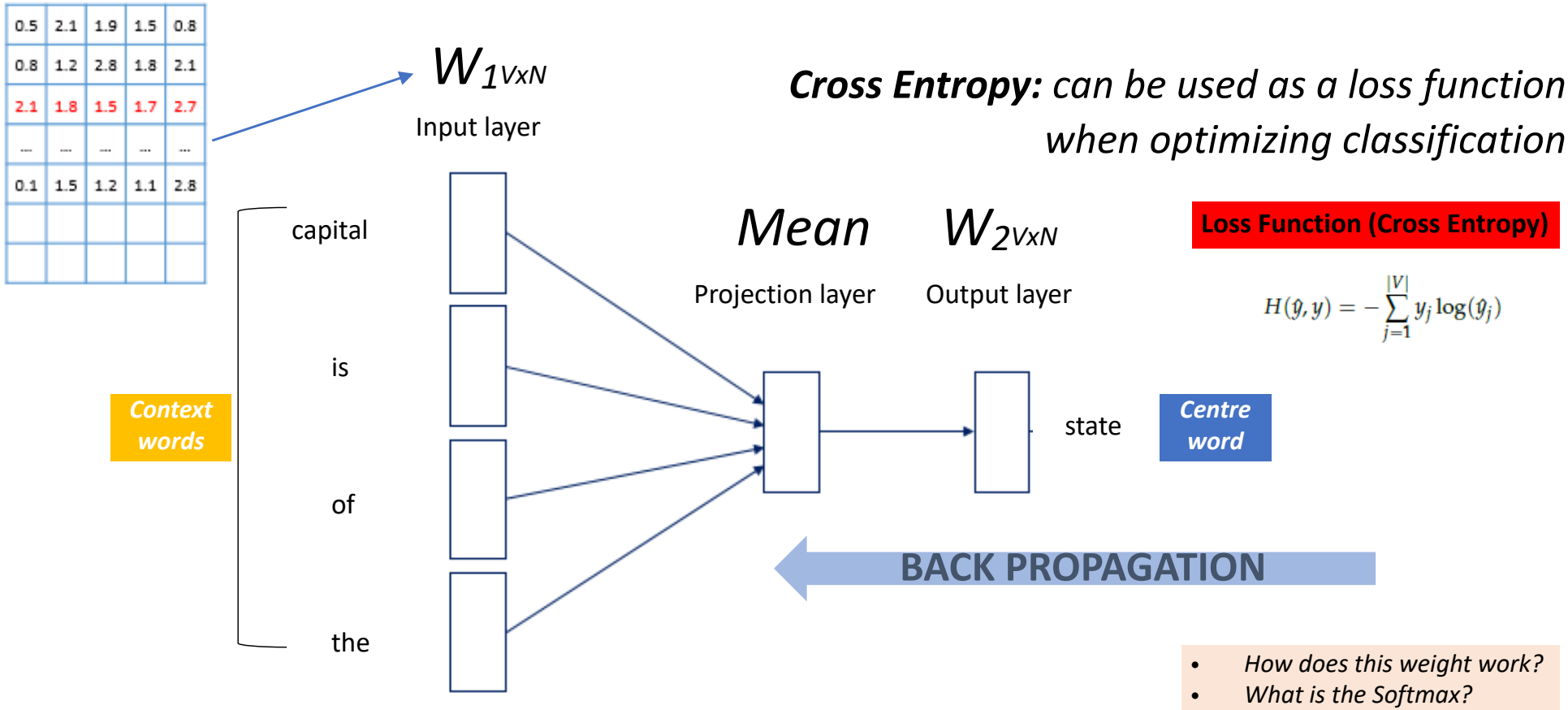
Questions: www.menti.com 2455 0800

THE UNIVERSITY OF SYDNEY

CBOW – Neural Network Architecture (ReCAP with Optimizer)

Predict centre word from (bag of) context words

Sentence: "Sydney is the state capital of NSW"



Cross Entropy: *can be used as a loss function when optimizing classification*

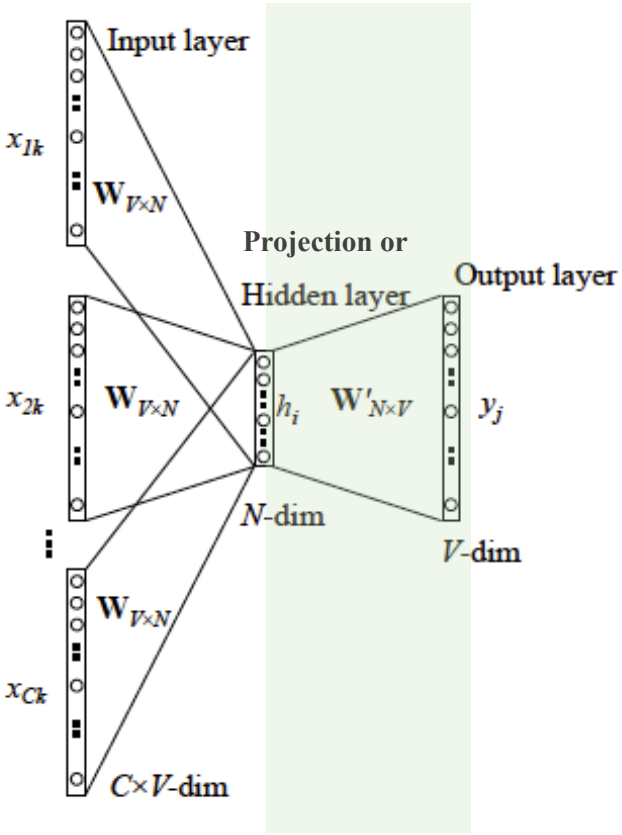
- *How does this weight work?*
- *What is the Softmax?*
- ***What is the Cross Entropy?***
- *How to Train the model?*

Questions: www.menti.com 2455 0800

CBOW – Neural Network Architecture (ReCAP with Optimizer)

Predict centre word from (bag of) context words.

Summary of CBOW Training (Review your understanding with equations)



Where does the optimisation function on the last slide come from?

$$\begin{aligned}\text{minimize } J &= -\log P(w_c|w_{c-m},\ldots,w_{c-1},w_{c+1},\ldots,w_{c+m}) \\ &= -\log P(u_c|\hat{v}) \\ &= -\log \frac{\exp(u_c^T\hat{v})}{\sum_{j=1}^{|V|}\exp(u_j^T\hat{v})} \\ &= -u_c^T\hat{v} + \log\sum_{j=1}^{|V|}\exp(u_j^T\hat{v})\end{aligned}$$

u_i=the output vector representation of word w_i

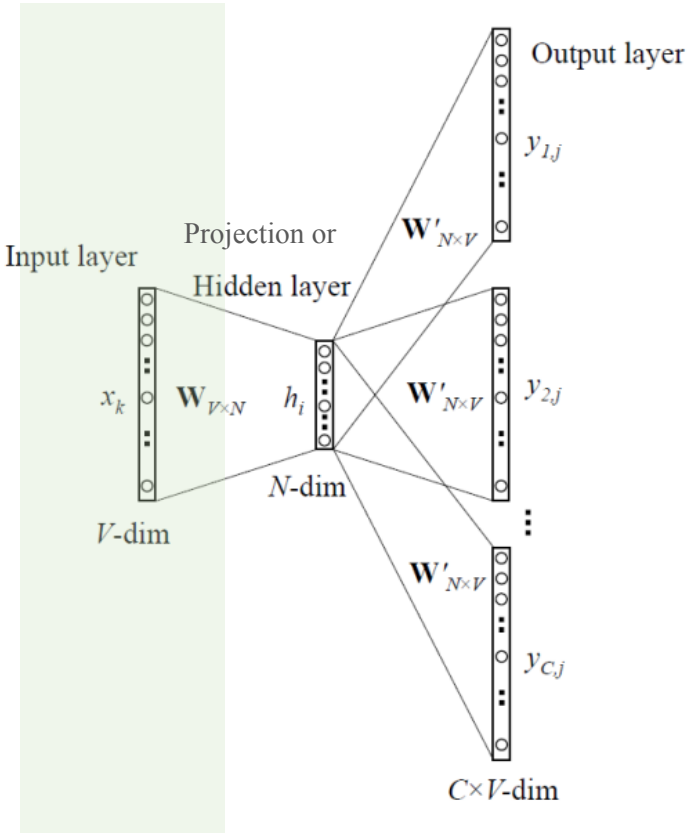
Questions: www.menti.com 2455 0800

3 Prediction based Word representation

Skip Gram – Neural Network Architecture

Predict context ("outside") words (position independent) given centre word

Summary of Skip Gram Training (Review your understanding with equations)



Has two Parameter Matrices:

1) Parameter Matrix (Input)

$$\mathbf{W} \in \mathbb{R}^{V\text{x}N}$$

2) Parameter Matrix (Output)

$$\mathbf{W}' \in \mathbb{R}^{N\text{x}V}$$

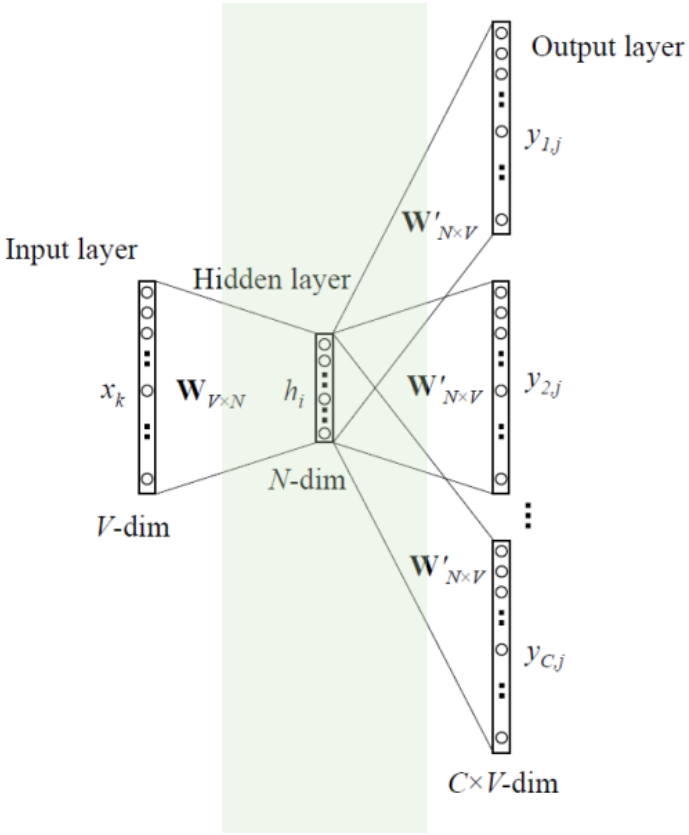
Questions: www.menti.com 2455 0800

3 Prediction based Word representation

Skip Gram – Neural Network Architecture

Predict context ("outside") words (position independent) given centre word

Summary of Skip Gram Training (Review your understanding with equations)



Initial words are looked up in $\mathbf{W_{VxN}}$ to get a 1 x N (embedded word) vector.

$\boldsymbol{v_c} = \mathbf{W}_{\boldsymbol{x}} \in \mathbb{R}^{\boldsymbol{n}}$ (as there is only one input)

Calculate the score value for the output layer by multiplying by the parameter matrix W'

$\boldsymbol{z} = \mathbf{W}'\boldsymbol{v_c}$

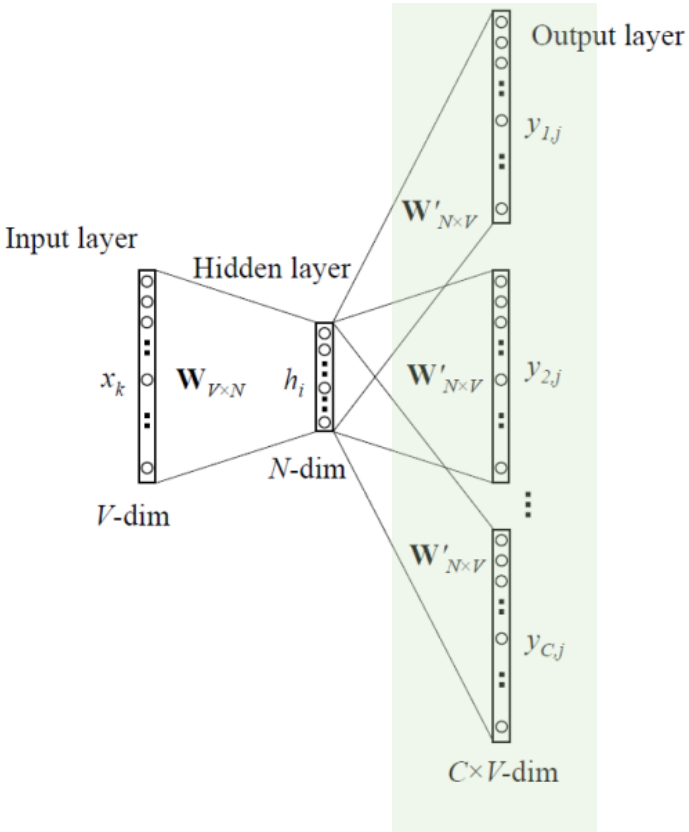
Questions: www.menti.com 2455 0800

Prediction based Word representation

Skip Gram – Neural Network Architecture

Predict context ("outside") words (position independent) given centre word

Summary of Skip Gram Training (Review your understanding with equations)



Calculate the probability using softmax

$$\hat{y} = softmax\,(\boldsymbol{z})$$

Calculate 2m probabilities as we need to predict 2m context words.

$$\hat{y}_{c-m},\ \ldots, \hat{y}_{c-1}, \hat{y}_{c+1},\ \ldots, \hat{y}_{c+m}$$

and compare with the ground truth

$$y^{(c-m)}, \ldots, y^{(c-1)},\ y^{(c+1)}, \ldots, y^{(c+m)}$$

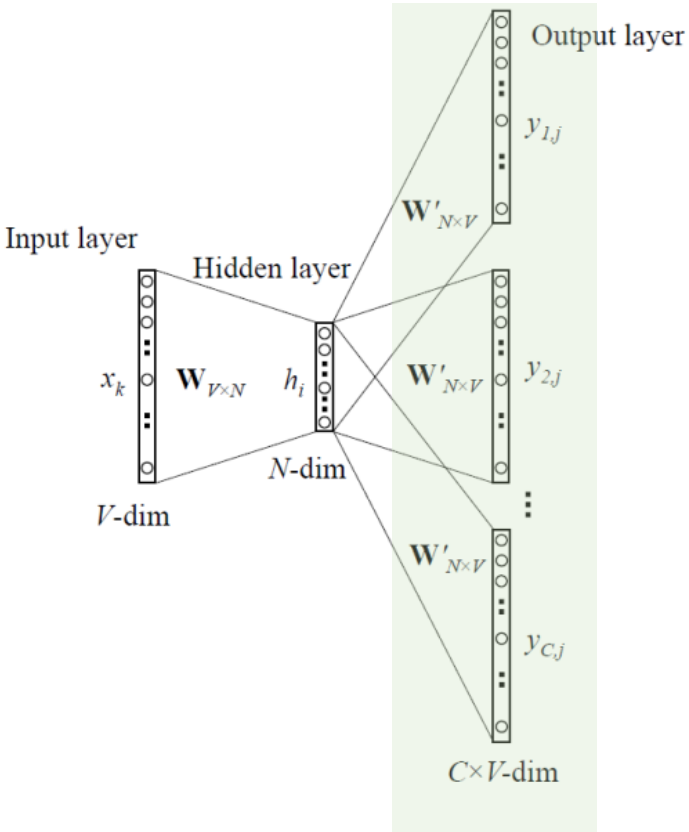
Questions: www.menti.com 2455 0800

Prediction based Word representation

Skip Gram – Neural Network Architecture

Predict context ("outside") words (position independent) given centre word

Summary of Skip Gram Training (Review your understanding with equations)



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.

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

Questions: www.menti.com 2455 0800

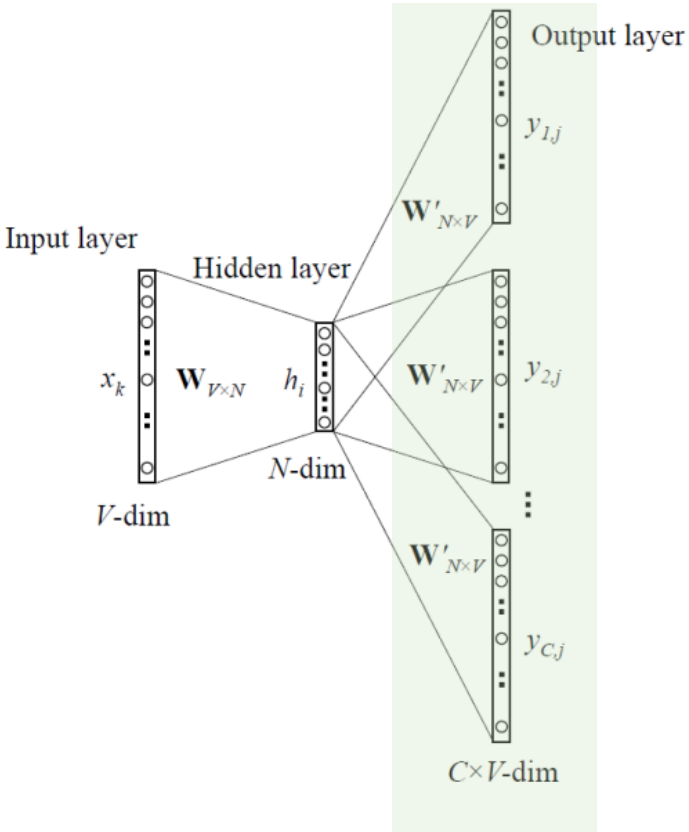
u_i=the output vector representation of word w_i

Prediction based Word representation

Skip Gram – Neural Network Architecture

Predict context ("outside") words (position independent) given centre word

Summary of Skip Gram Training (Review your understanding with equations)



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

$$J = -\sum_{j=0, j\neq m}^{2m} \log P(u_{c-m+j}|v_c)$$

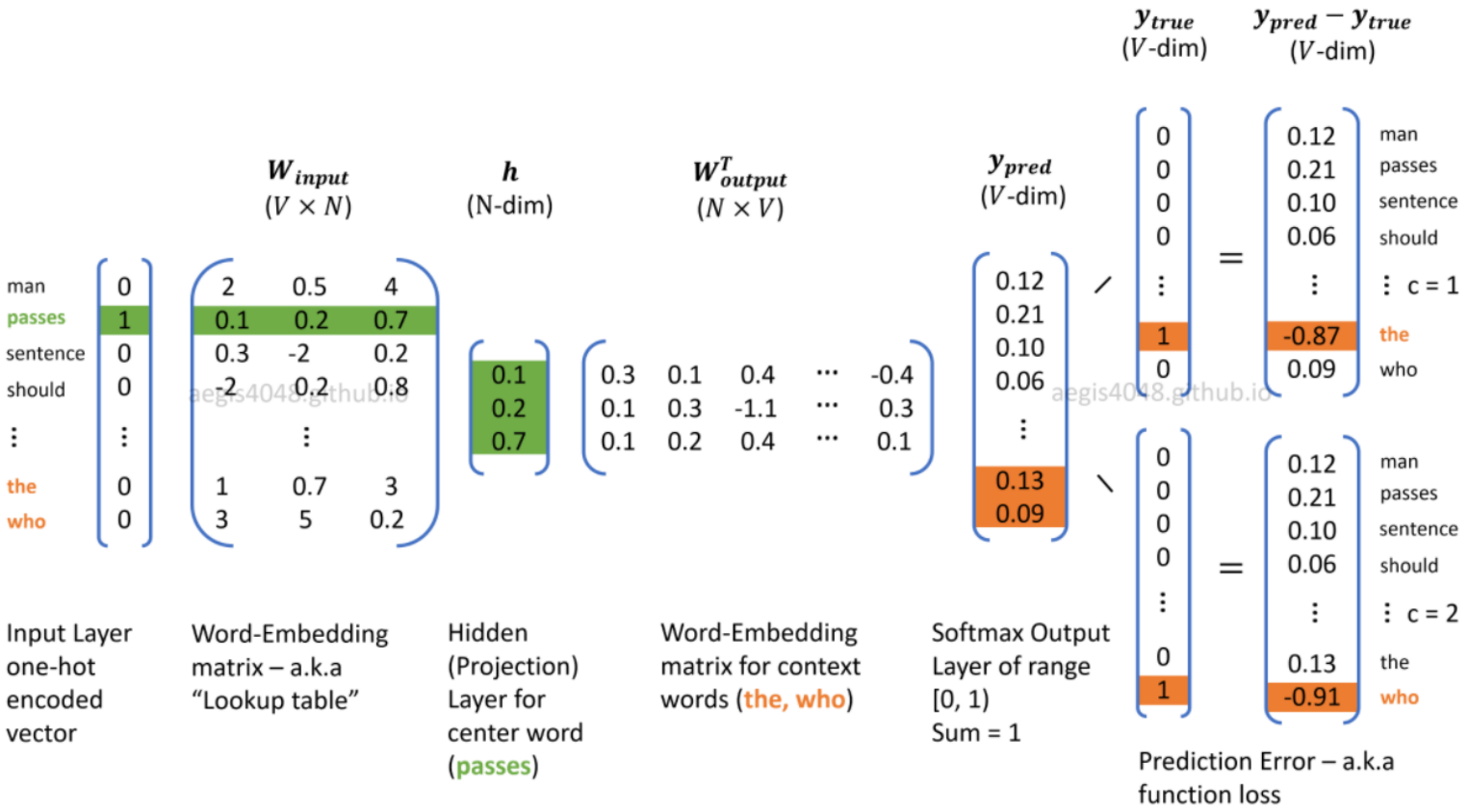
$$= \sum_{j=0, j\neq m}^{2m} H(\hat{y}, y_{c-m+j})$$

Questions: www.menti.com 2455 0800

Deep Learning for NLP

Word2Vec-SkipGram Overview

With a simple diagram



Questions: www.menti.com 2455 0800 *https://aegis4048.github.io/optimize_computational_efficiency_of_skip-gram_with_negative_sampling*

THE UNIVERSITY OF SYDNEY

Key Parameter (2) for Training methods: Negative Samples

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

Negative samples in our dataset – samples of words that are not neighbours

Negative sample: 2

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

1 = Appeared

0 = Did Not Appear

Negative sample: 5

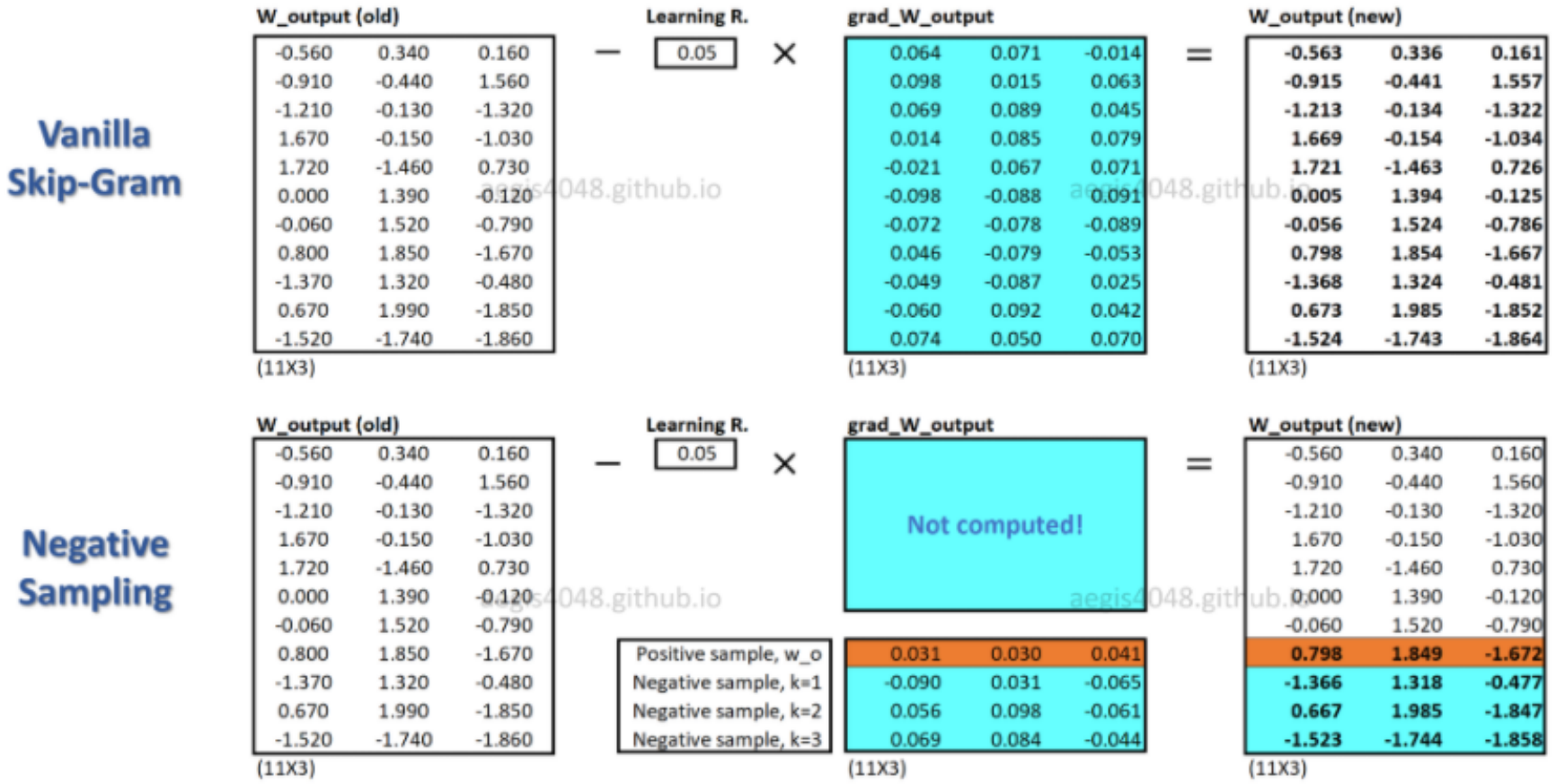
Input word	*Output word*	*Target*
eat	mango	1
eat	exam	0
eat	tobacco	0
eat	pool	0
eat	supervisor	0
eat	building	0

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**.

Questions: www.menti.com 2455 0800

Word2Vec-SkipGram Overview – negative sampling

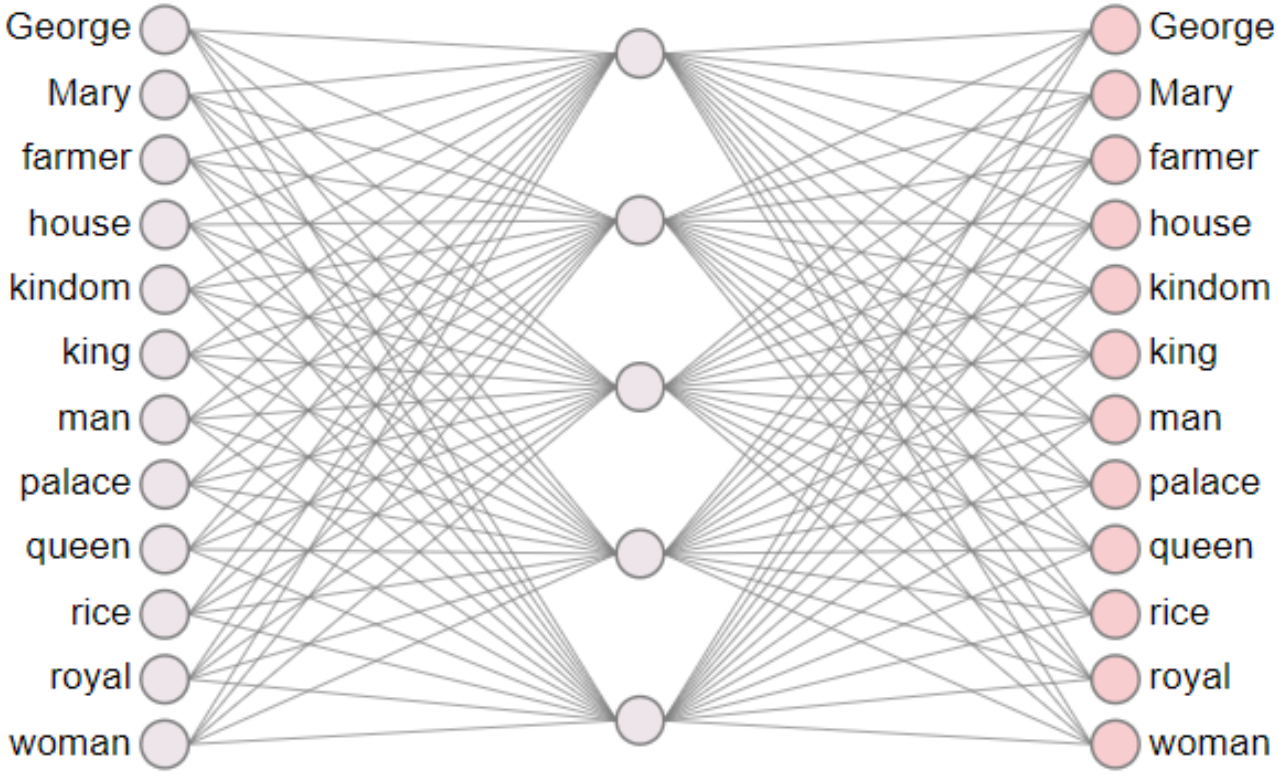
With a simple diagram



Questions: www.menti.com 2455 0800 *https://aegis4048.github.io/optimize_computational_efficiency_of_skip-gram_with_negative_sampling*

Application

Application #1: Embedding Pretraining



Questions: www.menti.com 2455 0800

http://ronxin.github.io/wevi/

0 LECTURE PLAN

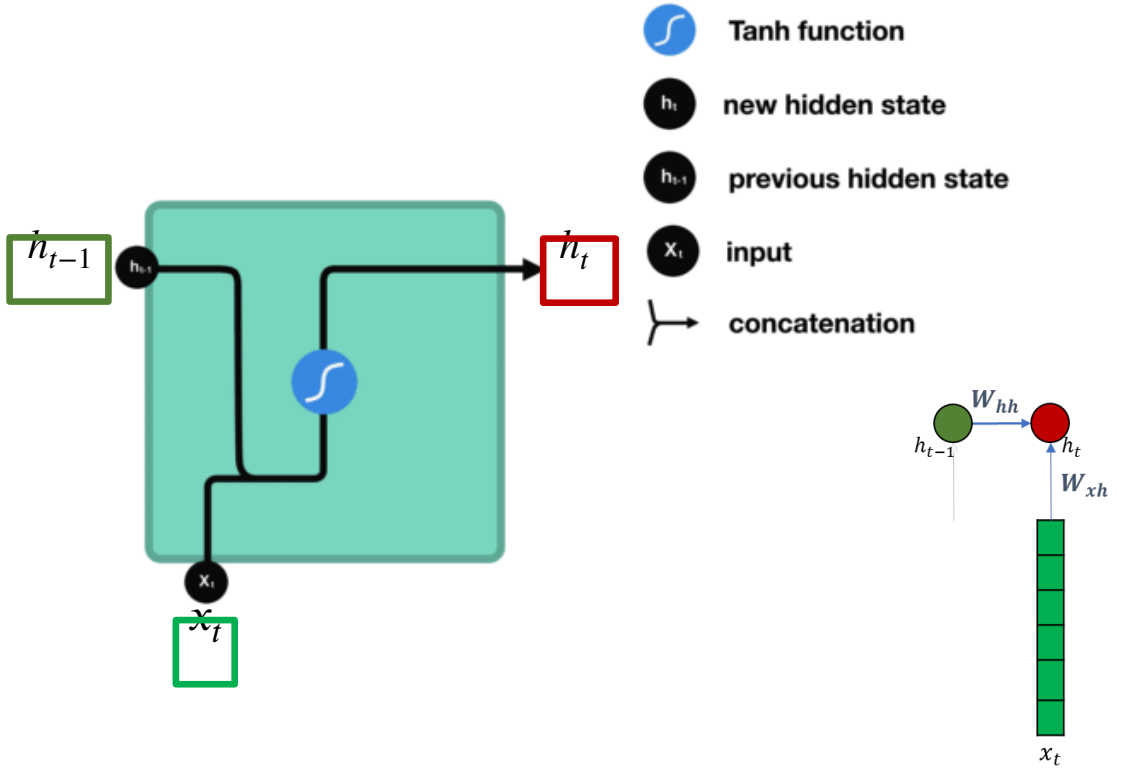
Lecture 3: Word Classification and Machine Learning

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.

Questions: www.menti.com 2455 0800

4 NEXT WEEK PREVIEW



$$h_t = \tanh(W_{hh}h_{t-1} + W_{xh}x_t + b_h)$$

New hidden state — *A function with parameters W* — *Previous state* — *input*

Questions: www.menti.com 2455 0800

Reference

References for this lecture

- Deng, L., & Liu, Y. (Eds.). (2018). Deep Learning in Natural Language Processing. Springer.
- Rao, D., & McMahan, 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

Questions: www.menti.com 2455 0800