

---

---

# MLDS 2017 Spring

## HW1 - Language Model

— [mldsntu2017@gmail.com](mailto:mldsntu2017@gmail.com) —

---

---

# Outline

- Introduction - The Microsoft Sentence Completion Challenge
- How to solve this task
- Output layer issue and the solutions
- How to use NCE in tensorflow
- How to reach the baseline
- Submission

# The Microsoft Research Sentence Completion Challenge

- The task is to complete the sentence with multiple choices given the contextual information
- Each sentence contains a underline indicating the missing word in the real-world literature
- Testing data source: Sherlock Holmes novels by Sir Arthur Conan Doyle
- Accuracy as evaluation metric

# The Microsoft Research Sentence Completion Challenge

example of training data:

There was no religious ceremony connected with marriage among us, while on the other hand the relation between man and woman was regarded as in itself mysterious and holy. It appears that where marriage is solemnized by the church and blessed by the priest, it may at the same time be surrounded with customs and ideas of a frivolous, superficial, and even prurient character. We believed that two who love should be united in secret, before the public acknowledgment of their union, and should taste their apotheosis alone with nature. The betrothal might or might not be discussed and approved by the parents, but in either case it was customary for the young pair to disappear into the wilderness, there to pass some days or weeks in perfect seclusion and dual solitude, afterward returning to the village as man and wife. An exchange of presents and entertainments between the two families usually followed, but the nuptial blessing was given by the High Priest of God, the most reverend and holy Nature.

# The Microsoft Research Sentence Completion Challenge

example of testing data:

```
id,question,a,b,c,d,e
1,I have it from the same source that you are both an
  orphan and a bachelor and are _____ alone in London.
,crying,instantaneously,residing,matched,walking
2,"It was furnished partly as a sitting and partly as
  a bedroom , with flowers arranged _____ in every noo
k and corner.",daintily,privately,inadvertently,miser
ably,comfortably
```

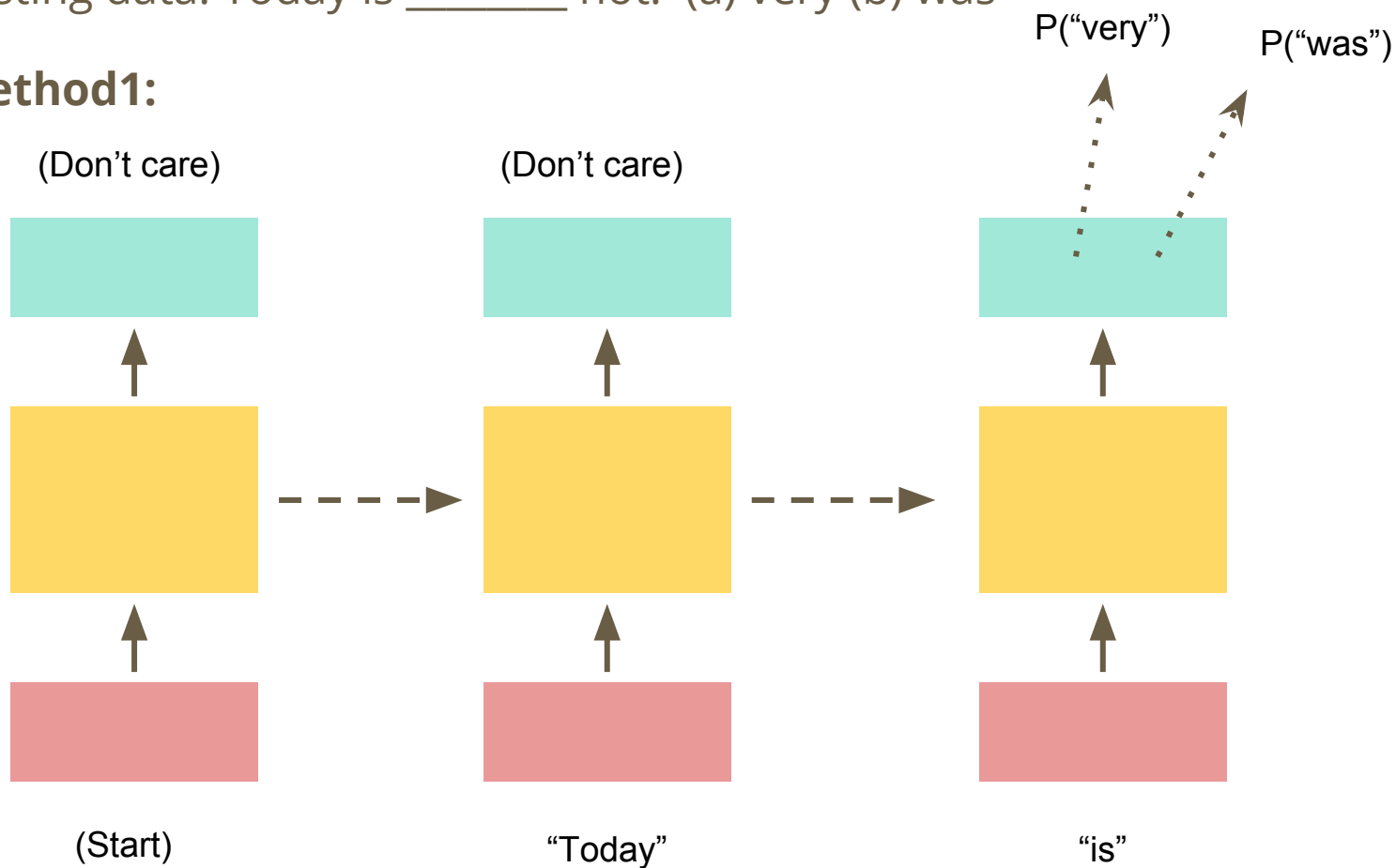
output format:

```
id,answer
1,a
2,a
3,a
4,a
```

# How To Solve This Task

Testing data: Today is \_\_\_\_\_ hot. (a) very (b) was

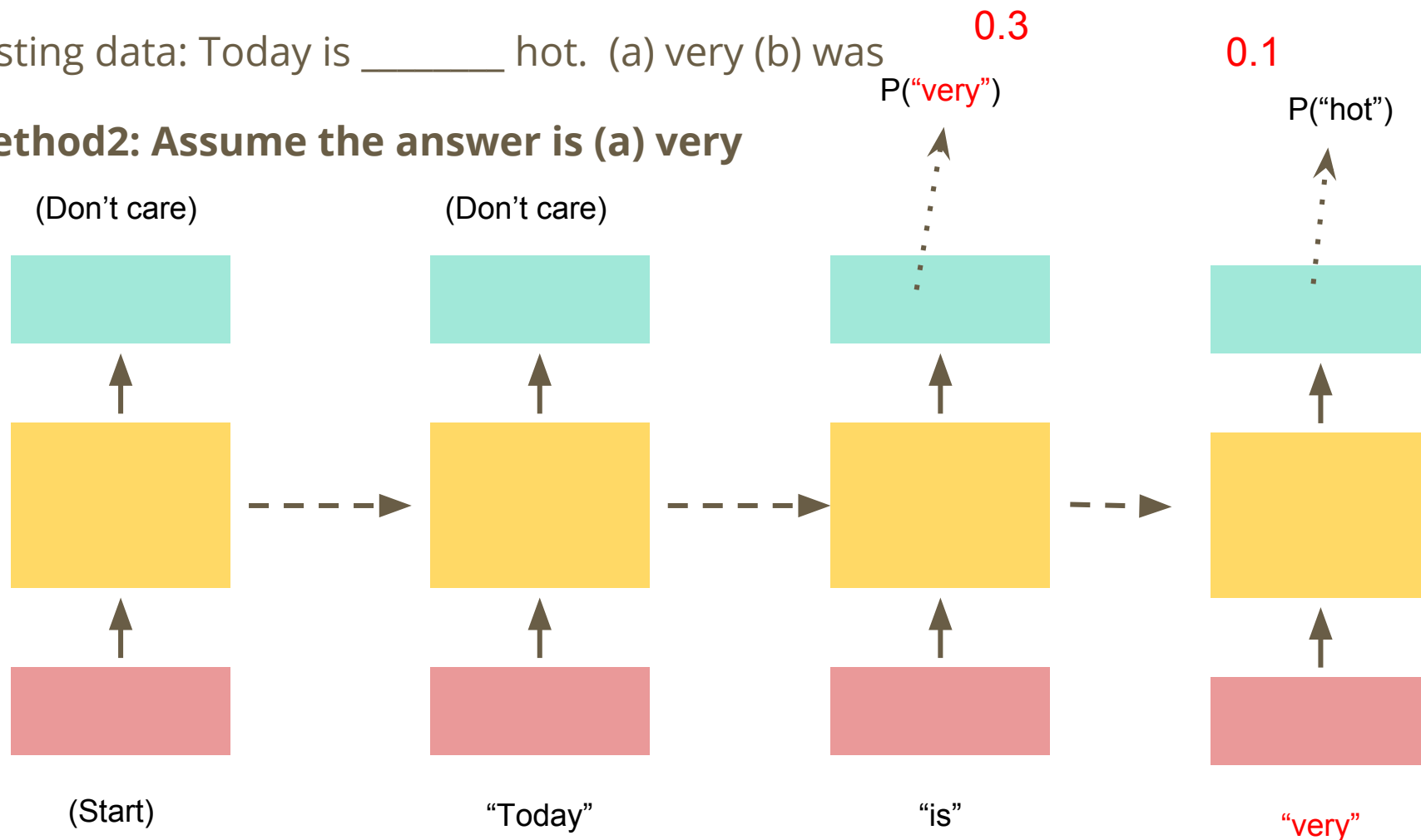
**method1:**



# How To Solve This Task

Testing data: Today is \_\_\_\_\_ hot. (a) very (b) was

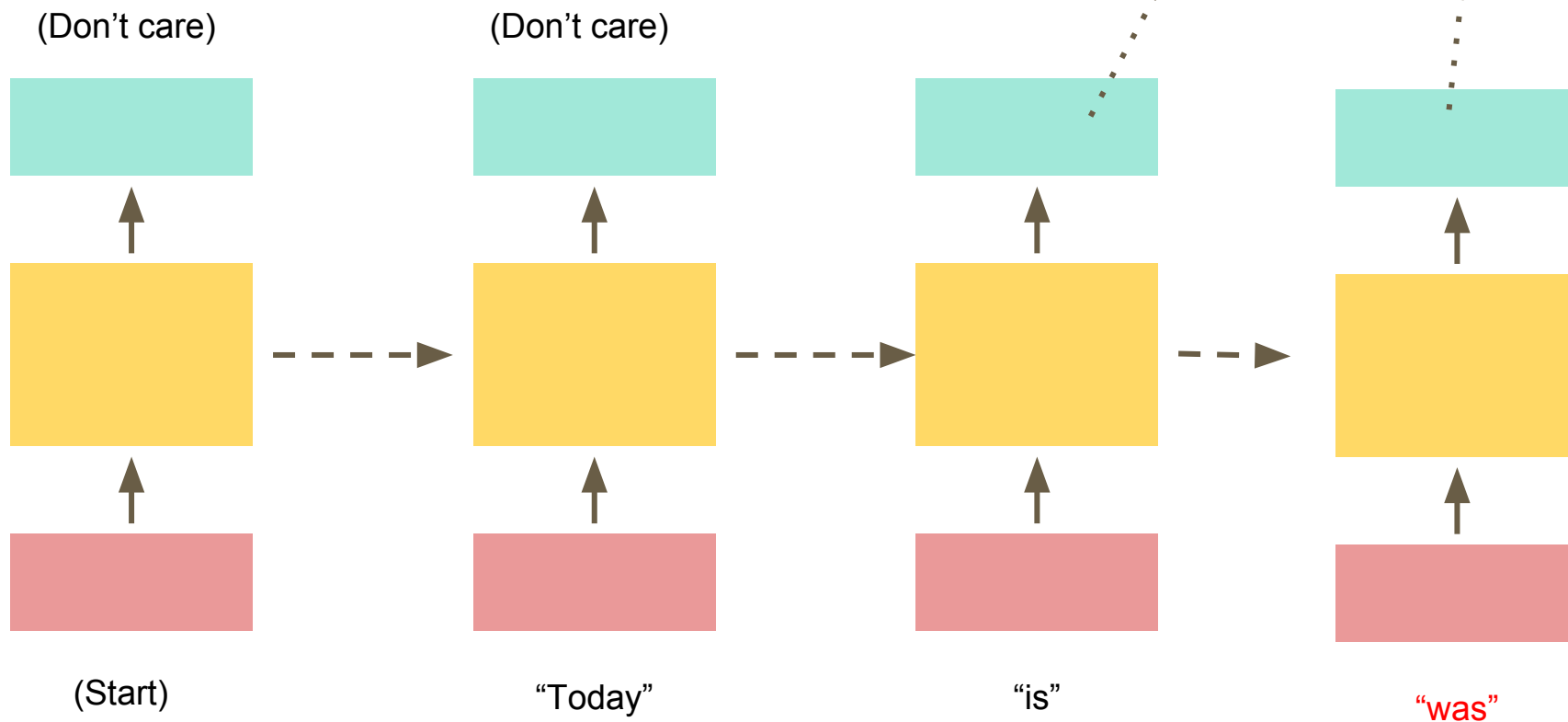
**method2: Assume the answer is (a) very**



# How To Solve This Task

Testing data: Today is \_\_\_\_\_ hot. (a) very (b) was

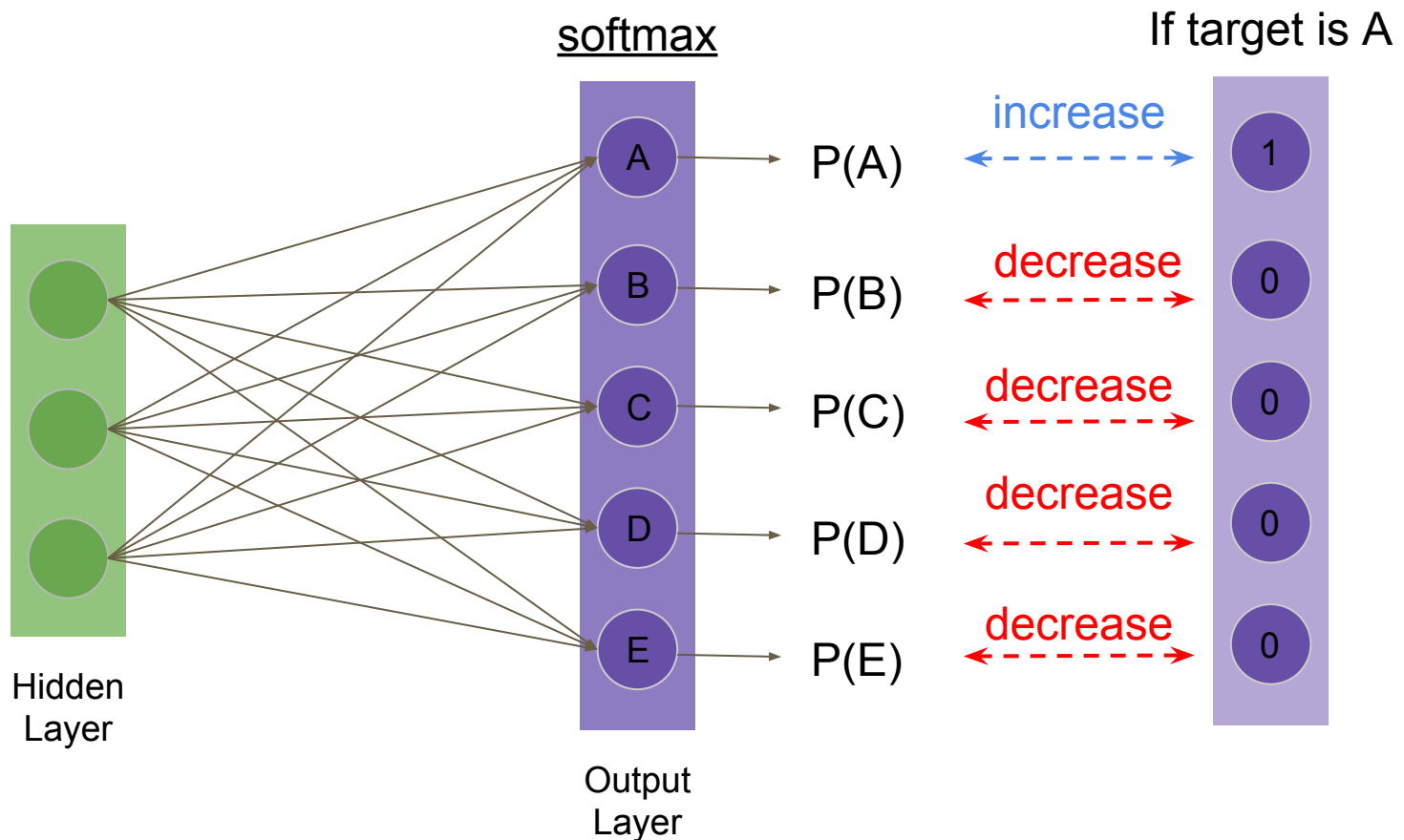
**method2: Assume the answer is (b) was**





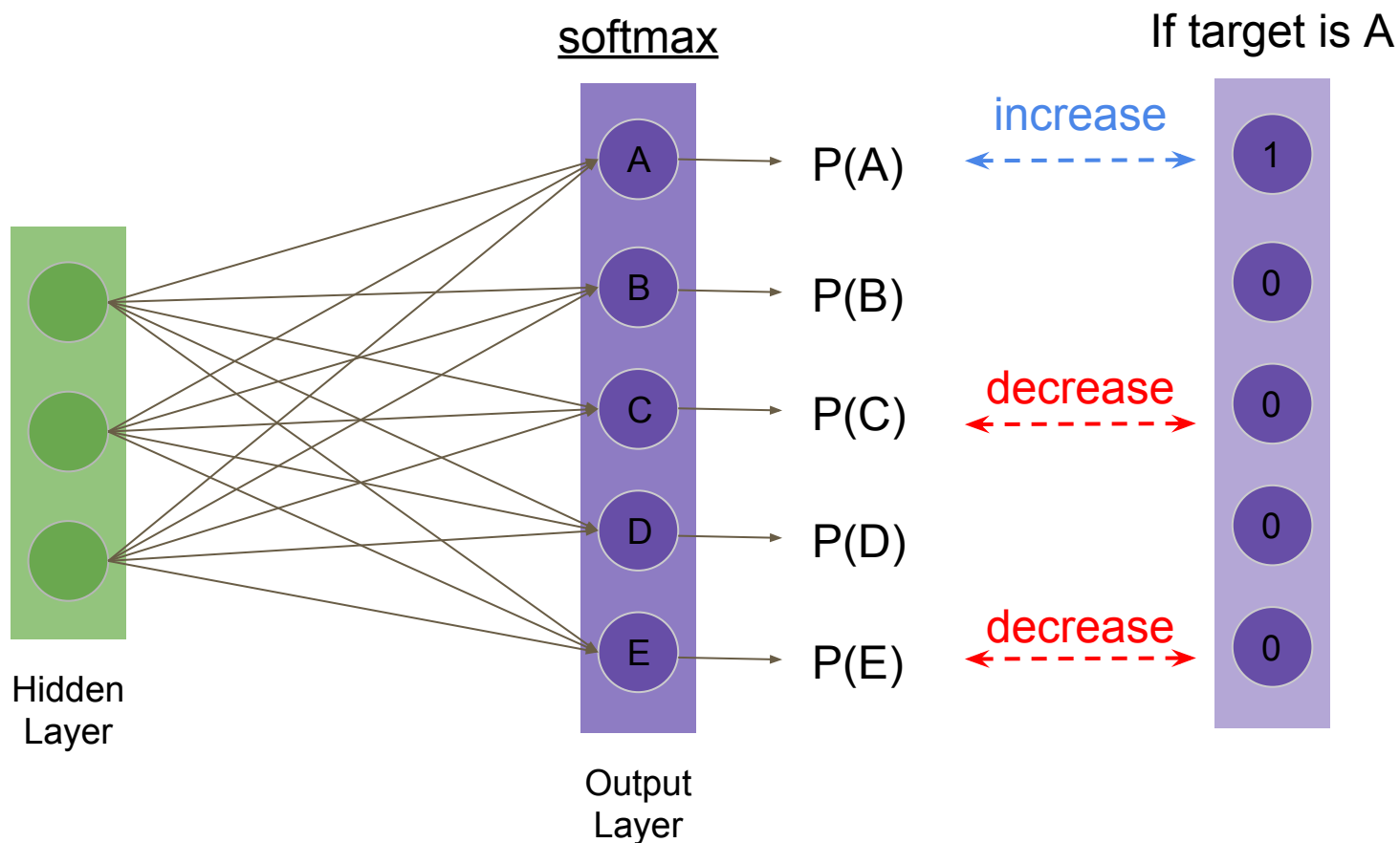
# Output Layer Issue

- output layer dimension = |vocabulary| ----> super high computing cost



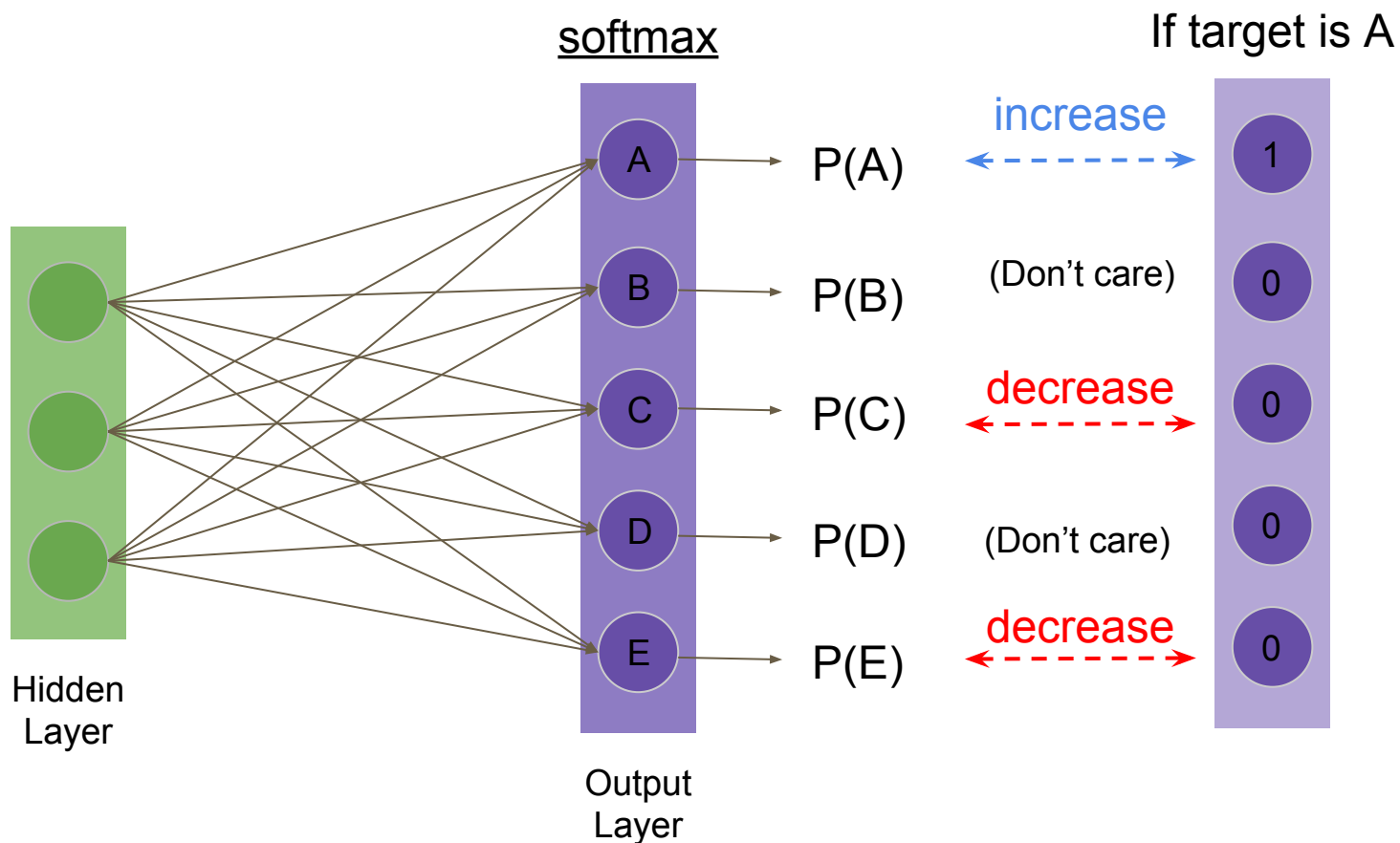
# Sampling Method - Noise Contrastive Estimation

- Noise Contrastive Estimation (NCE)
- Randomly sample some words to suppress the probability



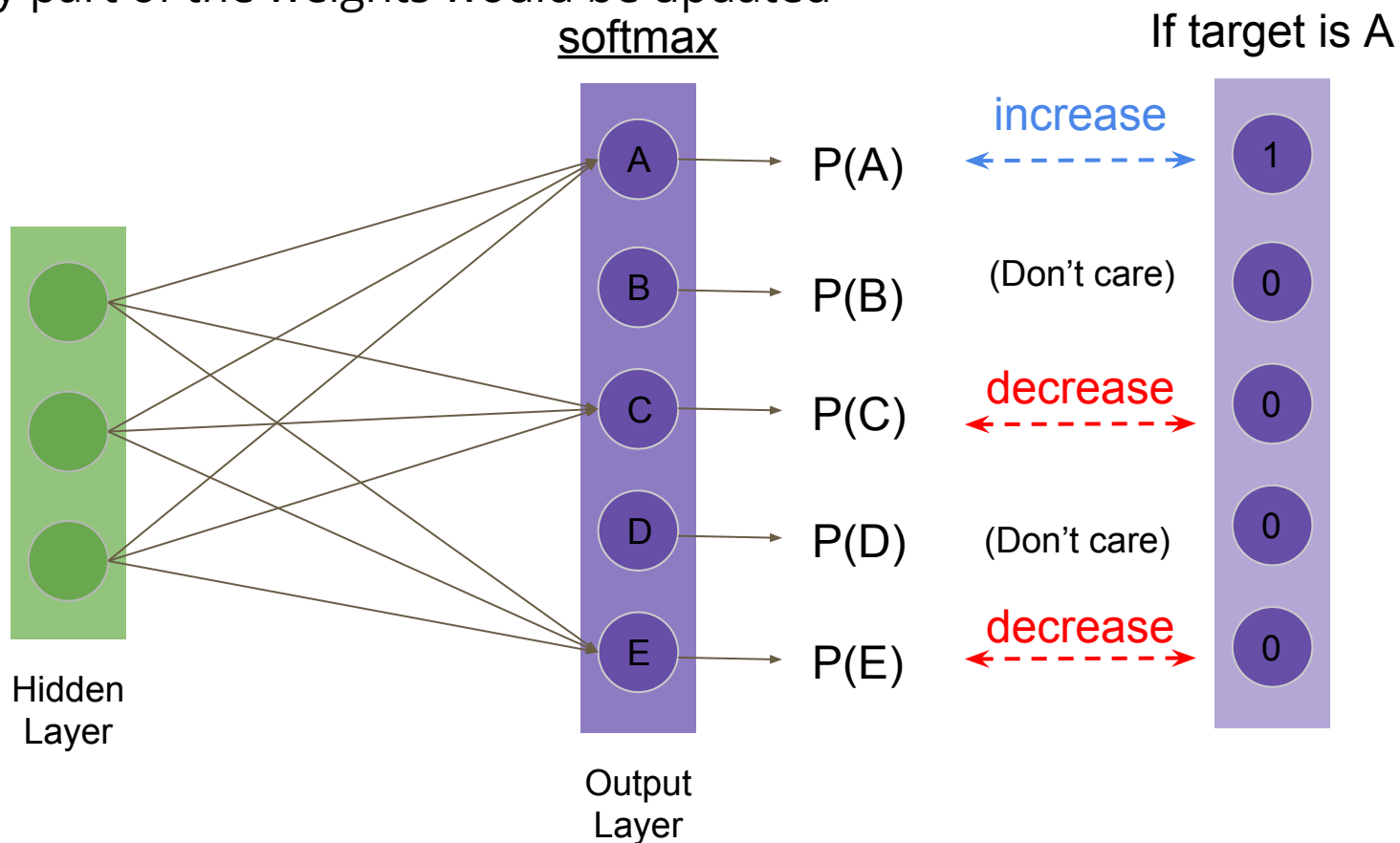
# Sampling Method - Noise Contrastive Estimation

- Noise Contrastive Estimation (NCE)
- Randomly sample some words to suppress the probability



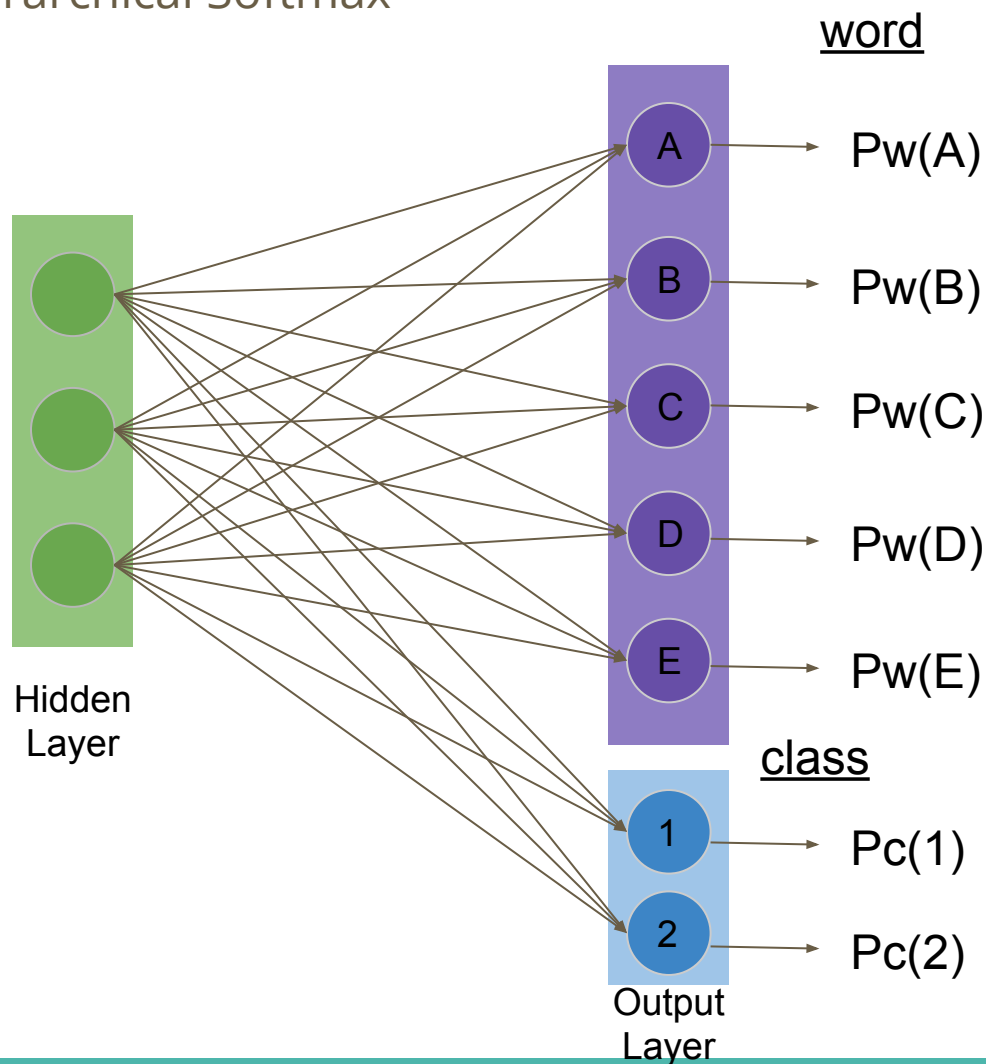
# Sampling Method - Noise Contrastive Estimation

- Noise Contrastive Estimation (NCE)
- Randomly sample some words to suppress the probability
- Only part of the weights would be updated



# Softmax-based Method - Hierarchical Softmax

- Hierarchical Softmax



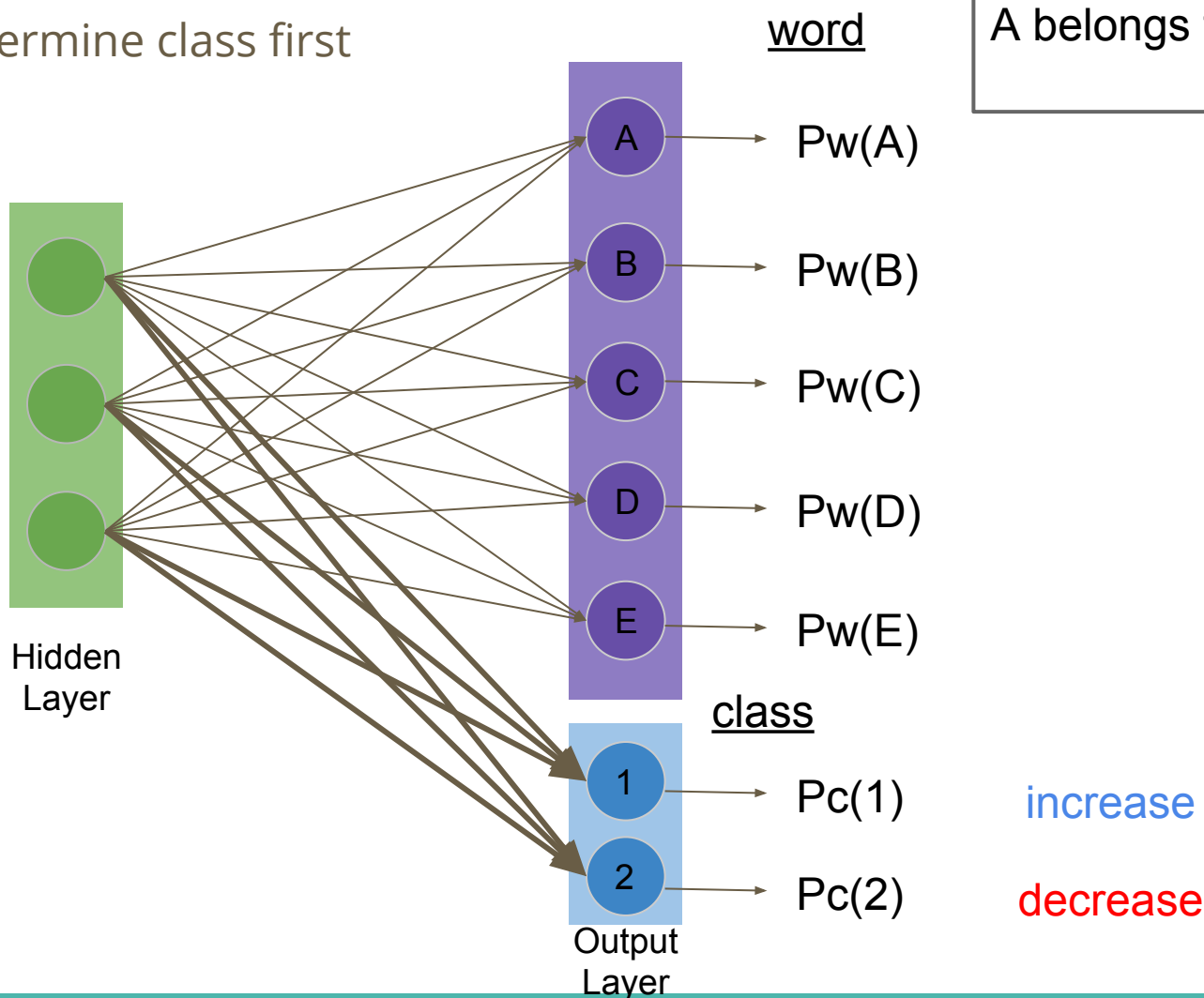
Word	Class
A	1
B	1
C	1
D	2
E	2

$$P(A) = P_w(A) \times P_c(1)$$

$$P(D) = P_w(D) \times P_c(2)$$

# Softmax-based Method - Hierarchical Softmax

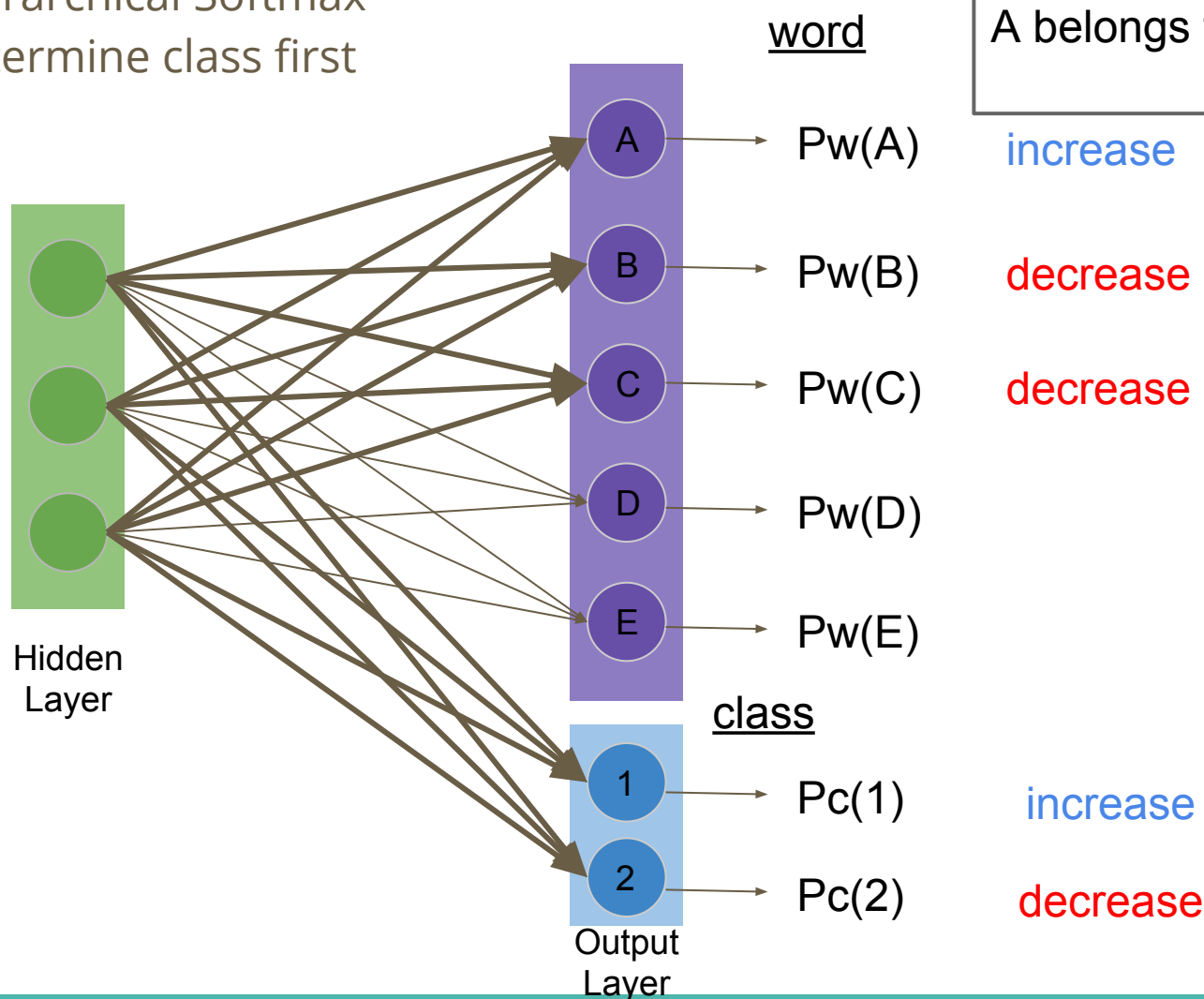
- Hierarchical Softmax
- determine class first



If target is A,  
A belongs to class 1

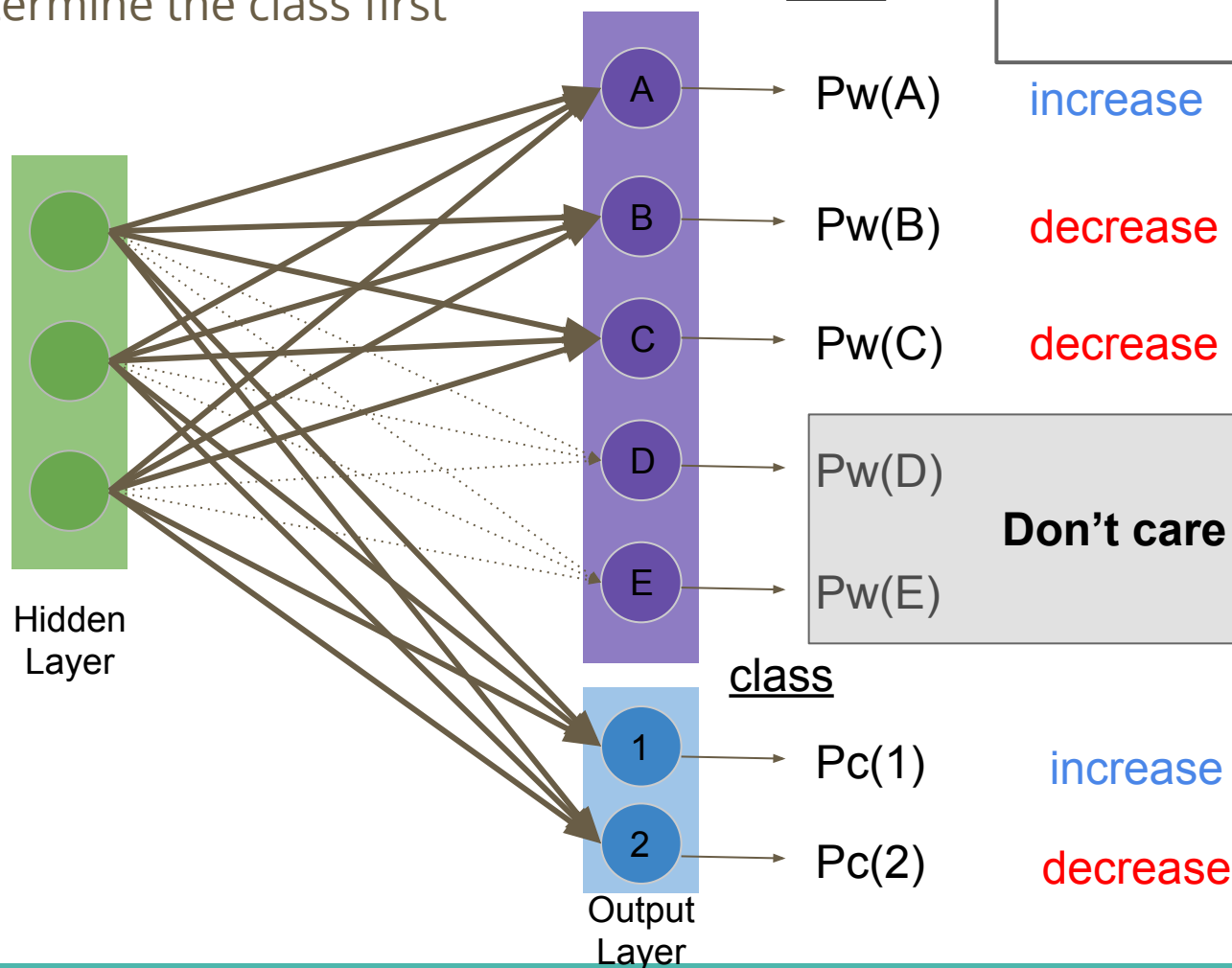
# Softmax-based Method - Hierarchical Softmax

- Hierarchical Softmax
- determine class first



# Softmax-based Method - Hierarchical Softmax

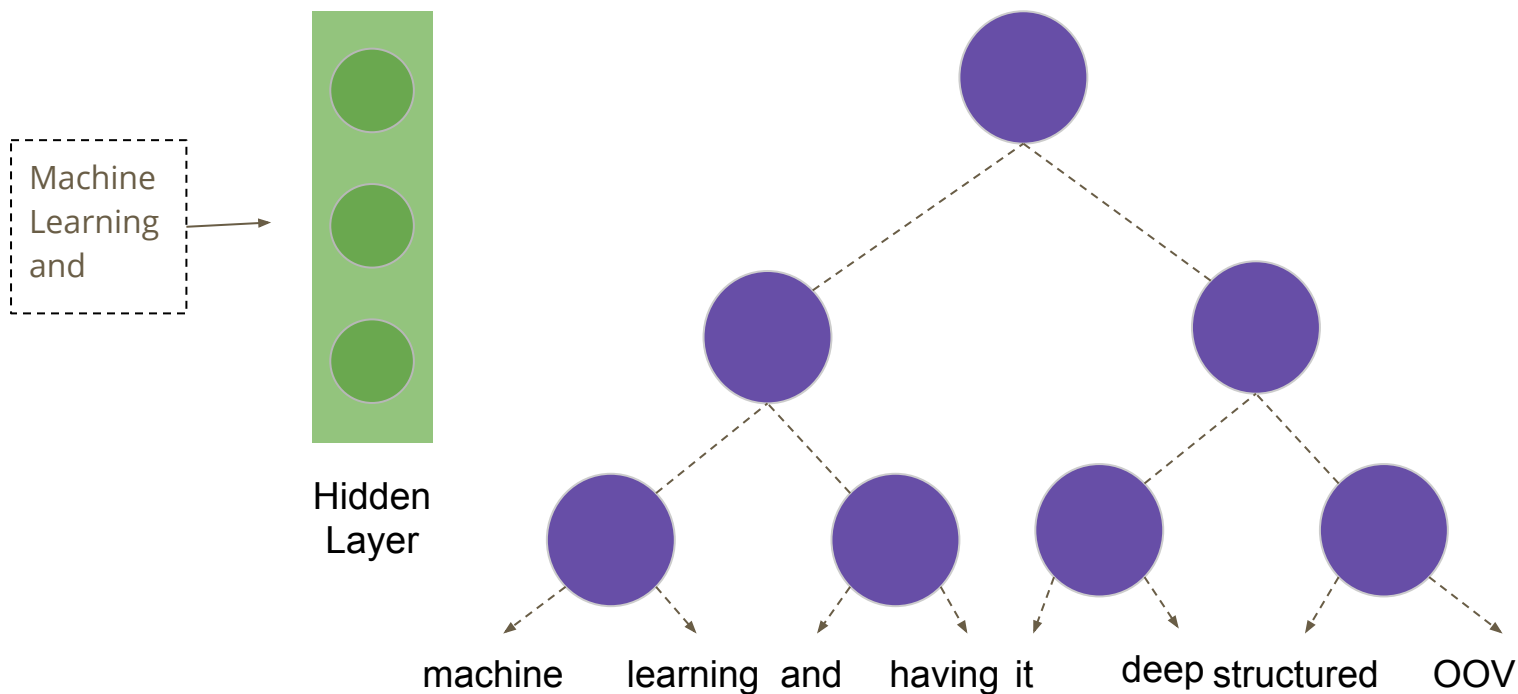
- Hierarchical Softmax
- determine the class first





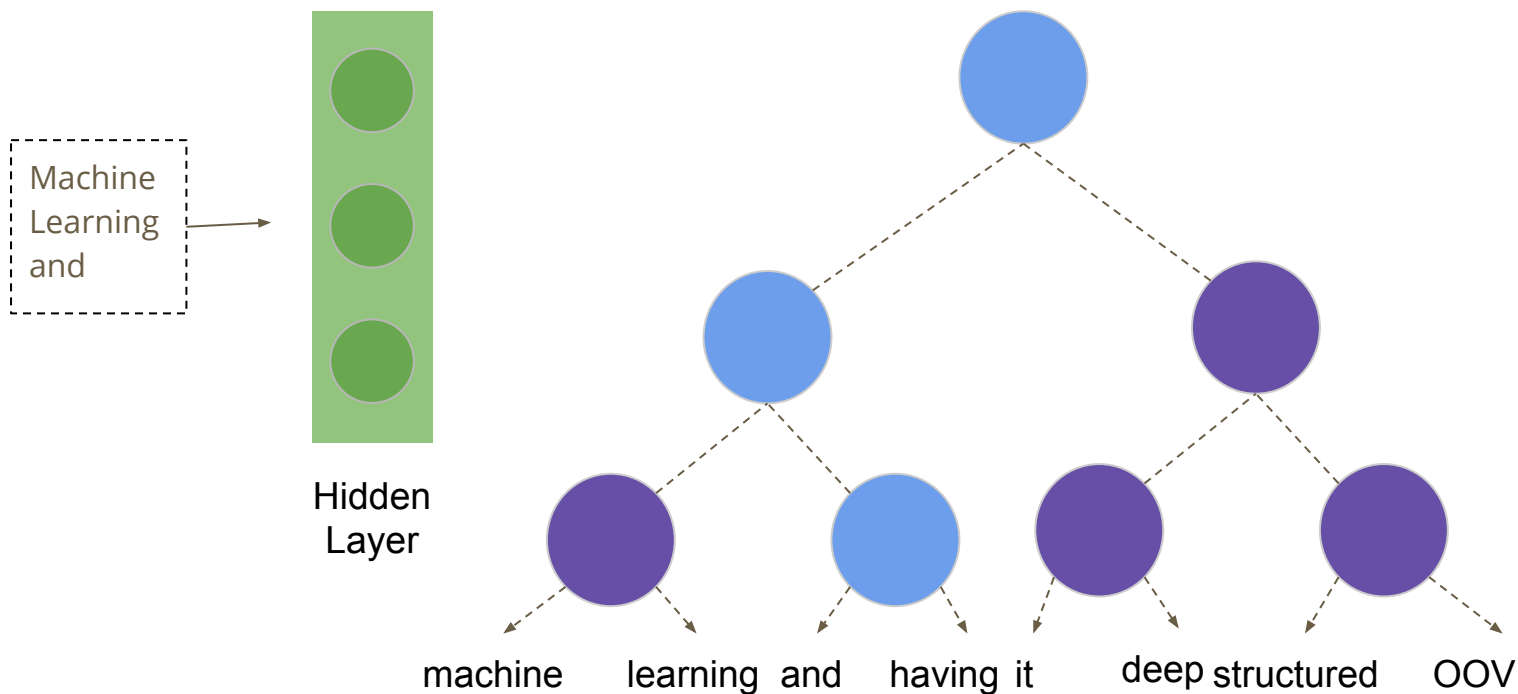
# Softmax-based Method - Hierarchical Softmax

- balanced binary tree
- if  $|\text{vocabulary}| = 10000$ , cost from 10000 to 13.3



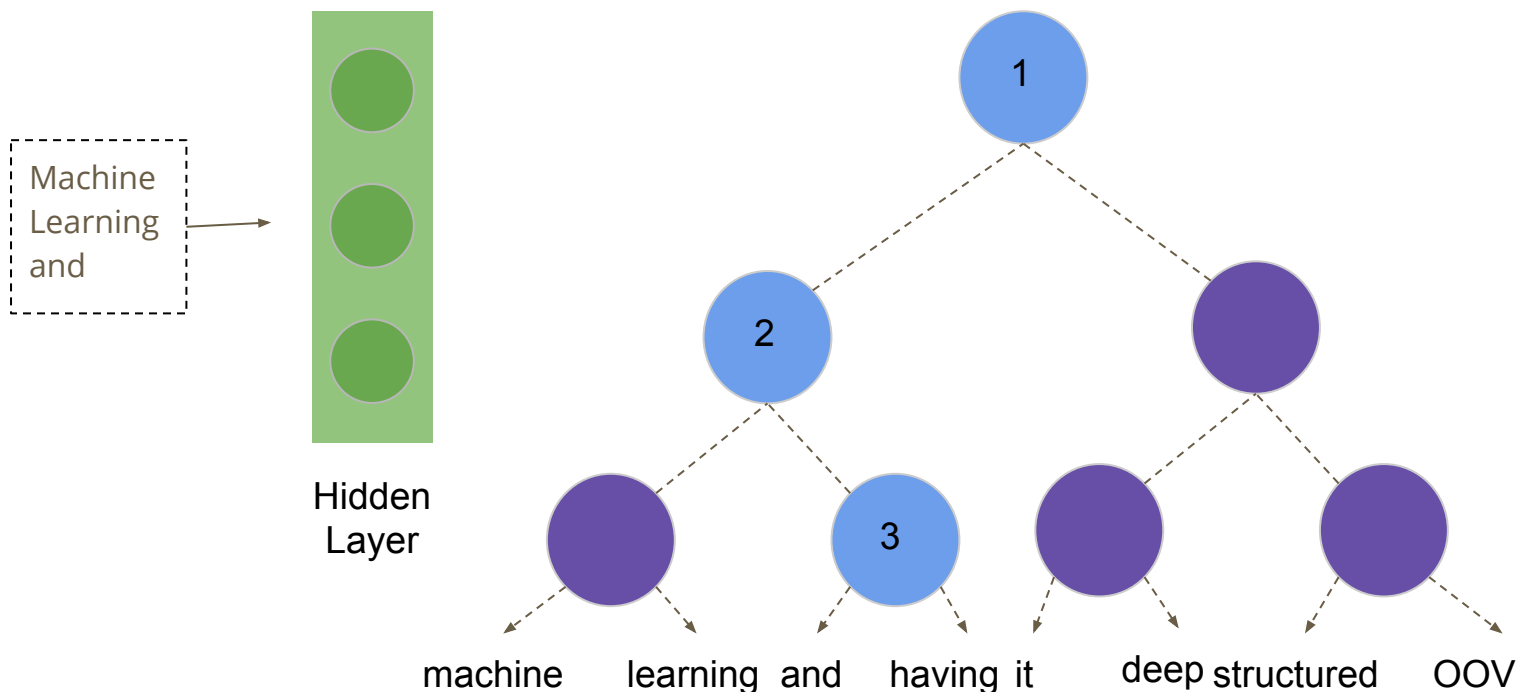
# Softmax-based Method - Hierarchical Softmax

- balanced binary tree
- if  $|\text{vocabulary}| = 10000$ , cost from 10000 to 13.3



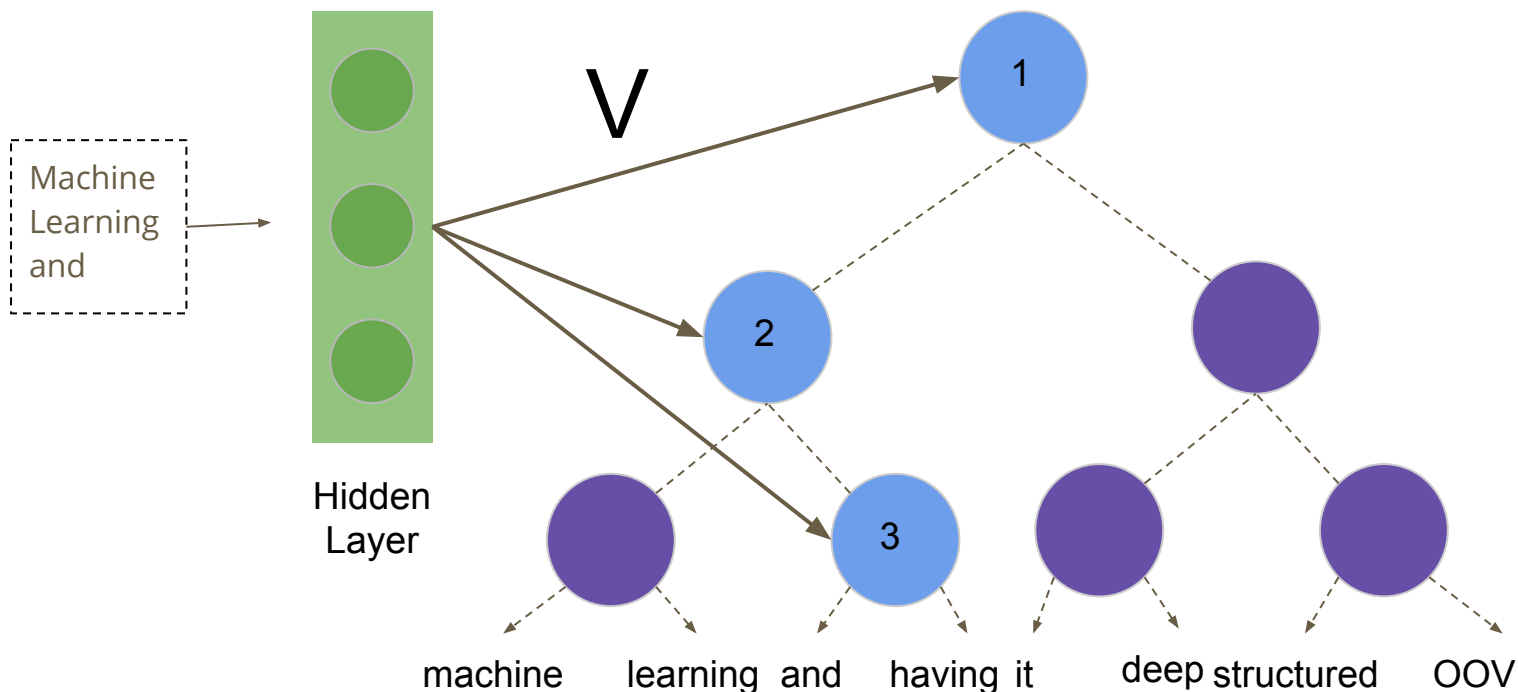
# Softmax-based Method - Hierarchical Softmax

- $$p(\text{"having"} \mid \text{context}) = (1 - p(\text{branch right at node 1} \mid \text{context})) \times p(\text{branch right at node 2} \mid \text{context}) \times p(\text{branch right at node 3} \mid \text{context})$$



# Softmax-based Method - Hierarchical Softmax

- $$p(\text{"having"} \mid \text{context}) = (1 - \text{sigmoid}(b_1 + v_1 \cdot h(x))) \times \text{sigmoid}(b_2 + v_2 \cdot h(x)) \times \text{sigmoid}(b_3 + v_3 \cdot h(x))$$

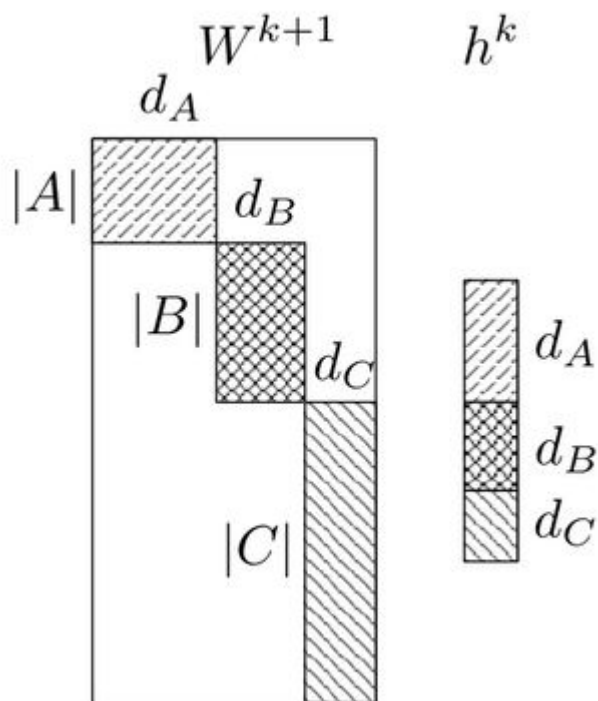


# Softmax-based Method - Hierarchical Softmax

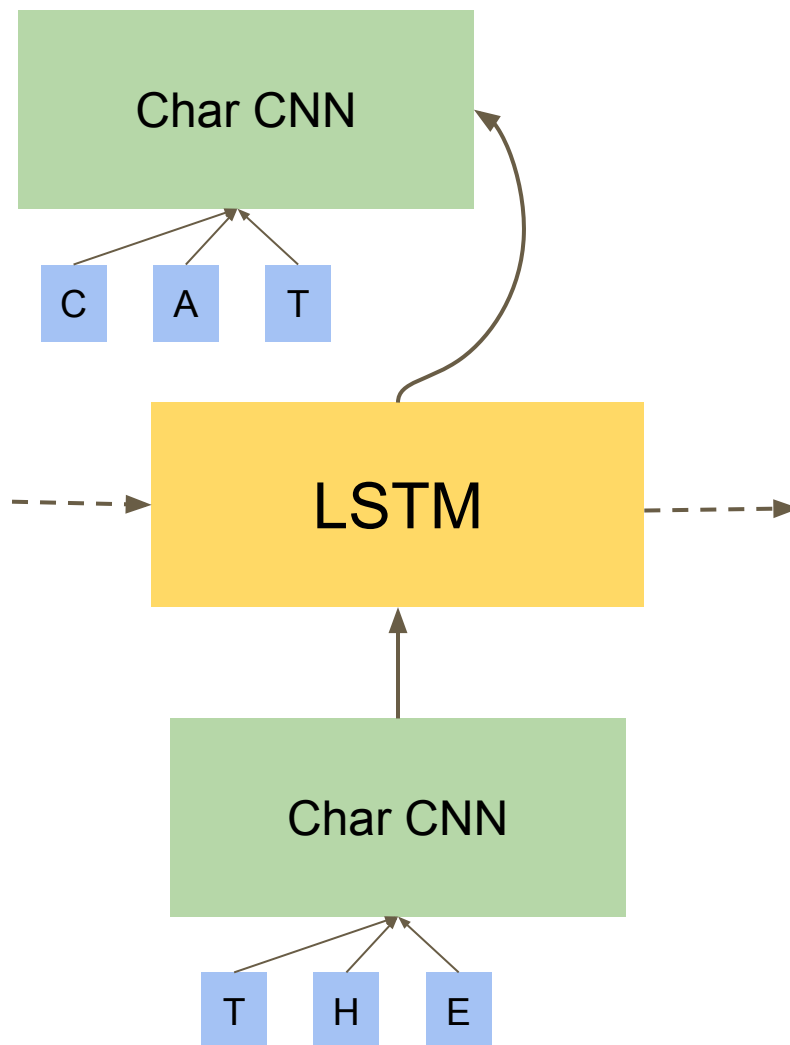
- How to define the word hierarchy?
  - randomly generated tree
  - existing linguistic resources, ex: WordNet
  - hierarchical clustering

# Softmax-based Method - Differentiated Softmax

- As many words will only require comparatively few parameters, the complexity of computing the softmax is reduced, which speeds up training. In contrast to H-Softmax, this speed-up persists during testing.



# Softmax-based Method - CNN-softmax



# Output Layer Issue - Note

- some techniques can only apply in training
- choose a technique regarding your computing resource
- out of memory / too slow



# How to use NCE in tensorflow

```
tf.nn.nce_loss(weights, biases, labels, inputs, num_sampled, num_classes,  
num_true=1, sampled_values=None, remove_accidental_hits=False,  
partition_strategy='mod', name='nce_loss')
```

Returns:

A **batch\_size** 1-D tensor of per-example NCE losses.

Be careful to the shape of weights and labels

# How to reach the baseline?

- Method 1
  - BasicLSTM with hidden state dimension 256
  - Standard softmax
  - AdamOptimizer
  - 12000 words
  - 2 epochs
  - Training time = 76 mins, by using TITAN X

Accuracy = 0.3317

# Submission on Kaggle

- [[Kaggle link](#)] will be published a few days later
- Data: <https://goo.gl/9jIYDd>
- Deadline: **3/25(Sat.) 23:59:59 (UTC+8)**
- Number of submissions per day : 5  
Refresh at 8:00:00 (UTC+8) every day
- You need to choose two submissions before deadline

```
id,answer
1,a
2,a
3,a
4,a
5,a
6,a
7,a
8,a
9,a
10,a
11,a
12,a
13,a
14,a
15,a
16,a
17,a
18,a
19,a
```

# Submission on Github

- Only **Python** with **Tensorflow r1.0** (TAs will run your code in Tensorflow-only environment).
- Deadline: **3/26(Sun.) 23:59:59 (UTC+8)**
- **MLDS2017/hw1** should contain all the things you use.  
Ex. **run.sh**, model, **report.pdf**, pre-trained word embedding, etc.

If some files are too big, upload to your cloud and write a script to download them. Remember to call the download script in run.sh.

- Usage:  
bash run.sh [path to testing data] [output path]

# Submission on Github

- You must create two branches **master** and **best**.
- **master** stores the model by **using only RNN** (external data is not allowed except for pre-trained word embeddings)
- **best** stores your best model. If your best model is the same as your baseline model, just copy all files in master to this branch.
- The format is mentioned in the previous slide.
- You have to specify the best performance you achieved by using only RNN in the experiment part of your report.

# Grading Policy (20%)

- Report (10%)
- Code (6%)
  - You **have to** use RNN and your model should output in **10 minutes**
- Baseline (2%+2%)
- Bonus
  - First place (10%)
  - Second place (5%)
  - Third place (3%)

You have to present how you beat other teams to get your bonus at 3/31.

# What report should cover?

- Environment (1%)
  - Ex. OS, CPU, GPU, Memory, libraries you used and version, etc.
- Model description (3%)
- How do you improve your performance (3%)
- Experiment settings and results (2%)
  - Ex. Epochs, training time, hyperparameters, etc.
- Team division (1%)
- No more than **3 pages**
- Please written in Chinese (unless you don't know how to type Chinese)

## Other Policy

- Incompatible format will not be graded.
- Late policy: 30% off per day late afterwards.  
[Delay form will be announced afterwards]
- You must register Kaggle using **NTU account**.
- Remember to fill the [group form](#), and add MLDS2017TA as github collaborator.
- No plagiarism is allowed.
- Use the given data only, except for pre-trained word embedding (you should specify the source in your report).



# For Tensorflow tutorial

- Install python3 shell on your laptop [[demo](#)]
- Install [Tensorflow r1.0](#)
- Download [dataset](#)
- Install [CUDA](#)