Homework

Homework
Machine Learning & Neural Networks
(a) Adam Optimizer
(b) Dropout
Neural Transition-Based Dependency Parsing
(a)
(b)
(f)

Machine Learning & Neural Networks

(a) Adam Optimizer

1

$$\begin{aligned} m &\leftarrow \beta_1 m + (1-\beta_1)\nabla_\theta J_{minibatch}(\theta) \\ \theta &\leftarrow \theta - \alpha m \end{aligned}$$

- 由于超参数 β_1 一般被设为0.9，此时对于移动平均的梯度值 m 而言，主要受到的是之前梯度的移动平均值的影响，而本次计算得到的梯度将会被缩放为原来的 $1-\beta_1$ 倍，即时本次计算得到的梯度很大（梯度爆炸），这一影响也会被减轻，从而阻止更新发生大的变化。

- 通过减小梯度的变化程度，使得每次的梯度更新更加稳定，从而使模型学习更加稳定，收敛速度更快，并且这也减慢了对于较大梯度值的参数的更新速度，保证其更新的稳定性。

2

$$\begin{aligned} m &\leftarrow \beta_1 m + (1-\beta_1)\nabla_\theta J_{minibatch}(\theta) \\ v &\leftarrow \beta_2 v + (1-\beta_2)(\nabla_\theta J_{minibatch}(\theta) \odot \nabla_\theta J_{minibatch}(\theta)) \\ \theta &\leftarrow \theta - \alpha \odot m / \sqrt{v} \end{aligned}$$

- 移动平均梯度最小的模型参数将得到较大的更新。
- 一方面，将梯度较小的参数的更新变大，帮助其走出局部最优点（鞍点）；另一方面，将梯度较大的参数的更新变小，使其更新更加稳定。结合以上两个方面，使学习更加快速的同时也更加稳定。

(b) Dropout

$$\begin{aligned} h_{drop} &= \gamma d \circ h \\ E_{p_{drop}}[h_{drop}]_i &= h_i \quad for\ all\ i \in \{1, \dots, D_h\} \end{aligned}$$

1

$$\gamma = 1/(1 - p_{drop})$$

prove:

$$\begin{aligned}\sum_i [h_{drop}]_i = \sum_i h_i = E[h] \\ = \gamma \sum_i (1 - p_{drop}) h_i\end{aligned}$$

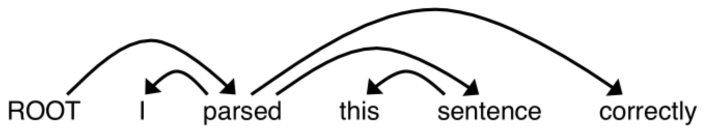
2

如果我们在评估期间应用 dropout ，那么评估结果将会具有随机性，并不能体现模型的真实性能，违背了正则化的初衷。通过在评估期间禁用 dropout，从而观察模型的性能与正则化的效果，保证模型的参数得到正确的更新。

Neural Transition-Based Dependency Parsing

(a)

“I parsed this sentence correctly”



STACK	BUFFER	NEW DEPENDENCY	TRANSITION
[ROOT]	[I, parsed, this, sentence, correctly]		Initial Configuartion
[ROOT, I]	[parsed, this, sentence, correctly]		SHIFT
[ROOT, I, parsed]	[this, sentence, correctly]		SHIFT
[ROOT, parsed]	[this, sentence, correctly]	parsed → I	LEFT_ARC
[ROOT, parsed, this]	[sentence, correctly]		SHIFT
[ROOT, parsed, this, sentence]	[correctly]		ShIFT
[ROOT, parsed, sentence]	[correctly]	sentence → this	LEFT_ARC
[ROOT, parsed]	[correctly]	parsed → sentence	RIGHT_ARC
[ROOT, parsed, correctly]	[]		SHIFT
[ROOT, parsed]	[]	parsed → correctly	RIGHT_ARC

STACK	BUFFER	NEW DEPENDENCY	TRANSITION
[ROOT]	[]	ROOT → parsed	RIGHT_ARC

(b)

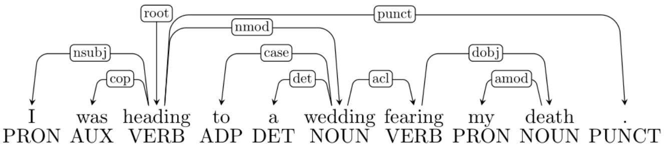
A sentence containing n words will be parsed in how many steps (in terms of n)? Briefly explain why

n steps SHIFT and n steps ARC(LEFT and RIGHT)

$n + n = 2n$

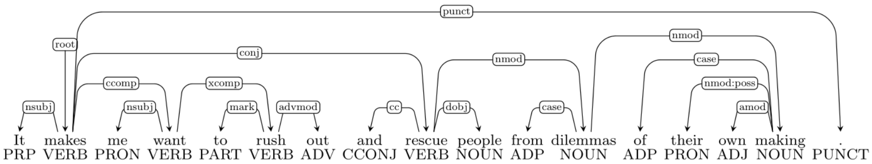
(f)

i.



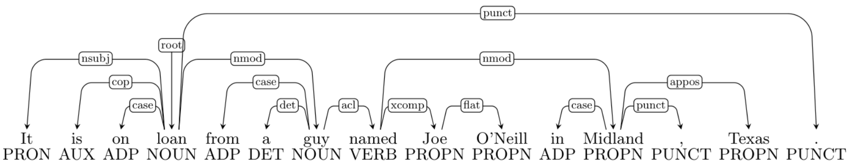
- Error type: Verb Phrase Attachment Error
- Incorrect dependency: wedding → fearing
- Correct dependency: heading → fearing

ii.



- Error type: Coordination Attachment Error
- Incorrect dependency: making → rescue
- Correct dependency: rush → rescue

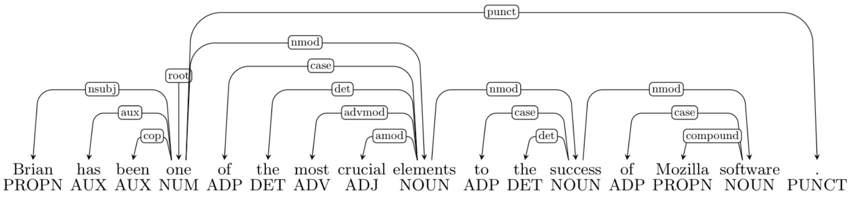
iii.



- Error type: Prepositional Phrase Attachment Error
- Incorrect dependency: named → Midland

- Correct dependency: named → Joe

iv.



- Error type: Modifier Attachment Error
- Incorrect dependency: element → most
- Correct dependency: crucial → most