A Time Delay Neural Network with Shared Weight Self-Attention for Small-Footprint Keyword Spotting[3]

Tran Nguyen Duc Tho, Pham Hoang

26, September 2020

Tóm tắt nội dung

Keyword spotting thường được dùng trên các thiết bị có phần cứng hạn chế với dung lượng bộ nhớ thấp. Tuy nhiên, các model từ trước tới nay vẫn dùng đến vài trăm ngàn tham số để đạt được hiệu suất tốt. Trong bài báo này tác giả muốn giới thiệu small-footprint keyword spotting model sử dụng Time Delay Neural Network với shared weight self-attention. Dataset Google Speech Command v1[2] được dùng để đánh giá models. Số lượng tham số của model là 12K bằng 1/20 so với model hiện đại nhất ResNet model(239k). Và model mà tác giả giới thiệu có error rate bằng 4.19%.

1 Introduction

Keyword spotting(KWS) là một loại công nghệ trong nhận dạng tiếng nói, người dùng có thể điều khiển các thiết bị thông minh bằng giọng nói chẳng nhạn như điện thoại, tablets và các thiết bị smart home. Bởi vì, nó thường chạy trên các thiết bị có hiệu năng phần cứng hạn chế đòi hỏi số lượng tham số của model phải ít và hiệu năng tính toán hiệu quả.

Có 2 hướng tiếp cận chính với KWS là dùng Large Vocalbulary Continuous speech recognition (LVCSR) và keyword/filter Hidden Markov models(HMMs). Tuy nhiên, cả hai hướng tiếp cận này đều tốn một dung lượng bộ nhớ khá lớn và đòi hỏi khả năng tính toán nhiều. Do vậy không phù hợp với các mobile devices.

Deep KWS xem bài toán Keyword Spotting như là bài toán classifier trong đó mỗi từ khóa là 1 class. Và thêm 1 class filter để biểu diễn các từ không phải từ khóa. Cách tiếp cận này có hiệu suất cao, độ trễ thấp và tốn ít chi phí tính toán phù hợp khi chạy trên các thiết bị di động. Tuy nhiên các model hiện số lượng tham số vẫn tương đối nhiều, với model hiện đại nhất là ResNet based KWS có 200k tham số.

The attention mechanism đã đạt kết quả tốt trong nhiều bài báo. Tuy nhiên, các recurrent structures lại khó để thực hiện tính toán song song, dẫn đến tốc độ tính toán chậm. Self-attention mechanism với hiệu suất tốt đã được ứng dụng trong machine translation và ASR.

Trong bài báo này, tác giả muốn giới thiệu phương pháp dùng TDNN với shared weight self-attention (SWSA)- một biến thể của self-attention. TDNN có chức năng bắt các local features của chuỗi và self-attention module captures global features. Với self-attention, tác giả sửa dụng SWSA module để giảm tham số . Self-Attention gốc sử dụng 3 bộ tham số để tìm features trong các không gian khác nhau. Thay vì dùng 3 ma trận tham số, SWSA module chỉ dùng 1 ma trận tham số (shared weight matrix) vì thế số lượng tham số được giảm đi. Các thực nghiệm tiến hành trên dữ liệu Speech Command V1 dataset[2]. So sánh với model tốt nhất là Resnet, model của có

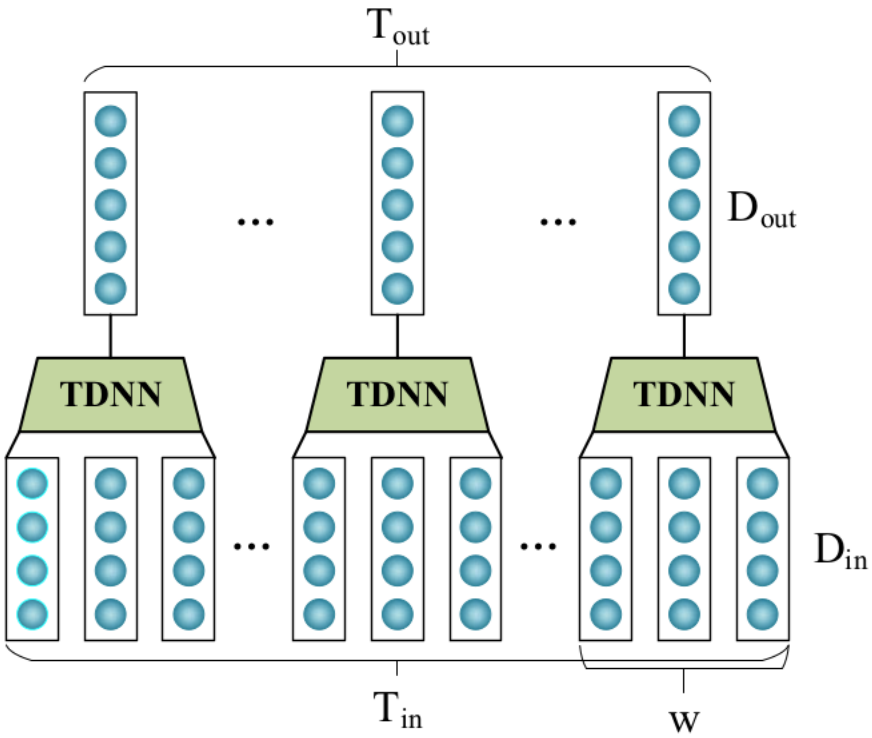
tác giả có classification error rate gần bằng nhau (4.19% vs 4.12%) và số lượng tham số nhỏ hơn rất nhiều (12k vs 239k).

2 Proposed Method

2.1 Time delay neural network

TDNN là 1 kiến trúc mạng cổ điển dùng nhiều trong speech recognition có nhiều biến thể và cách hiểu khác nhau về TDNN. Sau đây xin trình bày cách TDNN mà tác giả thực hiện. Một TDNN layer có thể xem là một DNN dịch chuyển theo thời gian. Tại mỗi bước, w features liền kề được nối và nhập vào một TDNN, và output của TDNN là 1 vector. Sau đó, TDNN dịch chuyển với stride là k. Thông thường, k được đặt bằng 1 có thể hiểu là TDNN thường hoặc là TDNN unsubsampling.

TDNN based subsampling, được biểu diễn trong hình 1, có chức năng là để giảm độ dài chuỗi. Và giá trị k được đặt lớn hơn 1 và nhỏ hơn w + 1



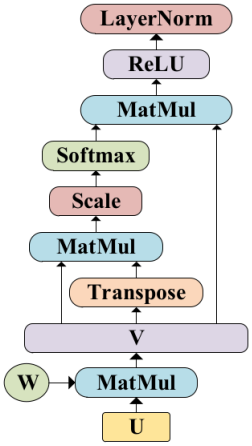
Hình 1: TDNN based subsampling. Tin và Din là length và dimensionality của input. Tout và Dout là length và dimensionality của output. Sau subsampling, chiều dài của chuỗi được giảm [(Tin-w+1)/k]. w là chiều dài của TDNN window.

2.1.1 Hiểu TDNN subsampling theo cách khác

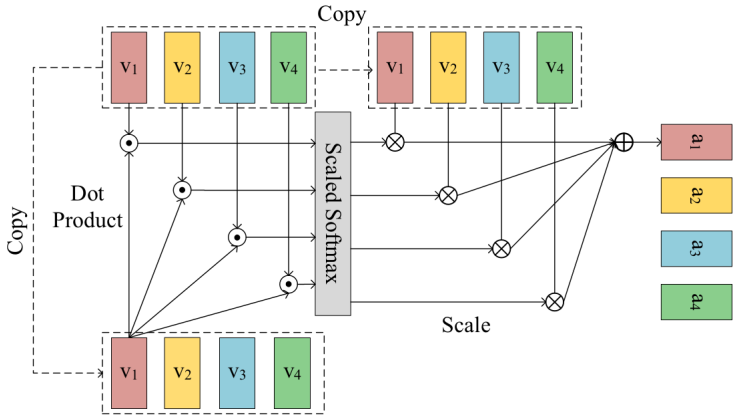
Ngoài ra, có thể hiểu TDNN subsampling theo cách khác. TDNN layer có 2 hyperparameter là stride k, length window w. Thay vì dùng w hyperparameter, người ta dùng input_context

hypterparameter. ví dụ trường hợp TDNN unsubsampling, input context là [-2, 2] có nghĩa là với mỗi step lấy các feature từ frame-2 đến frame + 2. Với TDNN-subsampling, input_context là {-2, 2} tức là chỉ có frame - 2 và frame + 2 được cho vào TDNN trong mỗi step vì người ta cho rằng có sự trùng lặp không cần thiết các features trong các step kề nhau. Do đó, giảm được lượng tham số đáng kể so với TDNN thông thường. Tham khảo [1] để biết thêm về TDNN.

2.2 Shared weight self-attention



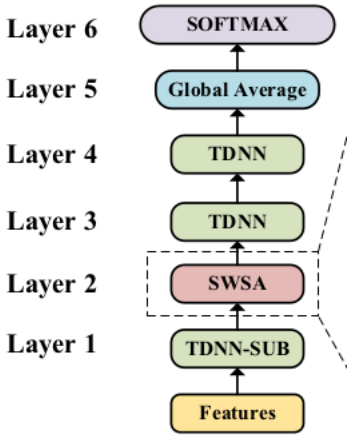
Hình 2: Shared Weights Self-Attention với U là input có chiều là (Tu, Du), W là weight có chiều là (Du, D).



Hình 3: vi là vector thứ i của kết quả phép nhân W * U, ai là vector thứ i của output attention.

2.3 Architecture

Input: Ma trận TxD,với T là độ dài của chuỗi đặc trưng và D là chiều của mỗi đặc trưng. Hình 4 biểu diễn kiến trúc của model với 6 layers. Layer 1 là TDNN based subsampling TDNN-SUB trên Hình 4. Layer 2 là SWSA layer. Layer 3 và layer 4 là TDNNs. Layers 5 là Global Mean Pooling. Và layer cuối là Softmax layer.



Hình 4: Kiến trúc model

3 Experiments

3.1 Datasets

Sử dụng English dataset Google Speech Command V1. Dataset bao gồm 64752 bản ghi của 30 từ. Mỗi bản ghi có thời lượng là 1s, và chỉ duy nhất 1 từ. 10 words được dùng làm keyword bao gồm "down", "go", "left", "no", "off", "on", "right", "stop", "up", "yes". 20 filters là các từ còn lại"bed", "bird", "cat", "dog", "happy", "house", "marvin", "sheila", "tree", "wow", "zero", "one", "two", "three", "four", "five", "six", "seven", "eight", "nine". Tất cả các filters đều được gán nhãn là __unknown__. Họ chia 6835 mẫu cho test set, 6798 cho validation set, 51088 cho trainining set. Có 11 classes: 10 class ứng với mỗi keyword, và class __unknown__. (Theo em việc chia dữ liệu như tác giả không hợp lý class __known__ có số mẫu gấp 17 lần so với mỗi class keyword còn lại dẫn đến model sau khi train chỉ nhận dạng tốt các từ __known__.)

3.2 Experimental setup

Model được đề xuất được đặt tên là tdnn-swsa, cấu hình model được nếu ra ở Table 2. Tác giả dùng 40 chiều MFCC với window stride là 10ms và window size là 25 ms. Các trọng số được khởi tạo Xavier initialization, thuật toán optimizer được sử dụng là Adam, với learning rate khởi tạo ban đầu là 0.001 và batch_size = 32. Sau mỗi epoch, model được đánh giá trên validation set. Nếu average cross entropy không cải thiện 10% thì chia đôi learning rate. Tổng số epoch traning là 13. Model được save theo tiêu chí là best accuracy trên validation set.

Tác giả dùng classification error rate để đo hiệu quả của model:

$$error = 1 - \frac{\sum_1^N I(y\_hat, y)}{N}$$

I(.,.) bằng 1 nếu y_hat bằng y, bằng 0 nếu ngược lại với y_hat là predicted label, y là truth label. N là tổng số mẫu.

Mỗi lần thực hiện tính error rate tác giả thực hiện 5 lần, và họ với khoảng tin cậy là 95%. Và khoảng tin cậy được tính bằng:

$$1.96\frac{\sigma}{5}$$

với σ là độ lệch chuẩn

Layer	w	k	d	l	#Para.	#Mult.
INPUT	-	-	40	99	-	-
TDNN-SUB	3	3	32	33	3840	126720
SWSA	-	-	32	33	1056	72768
TDNN	3	1	32	33	3072	101376
TDNN	3	1	32	33	3072	101376
Global Pooling	-	-	32	-	-	-
SOFTMAX	-	-	-	-	352	352
Total					11755	402592

Bảng 1: model tdnn-swsa. w và k là window length và stride của mỗi TDNN. Chiều của mỗi output maxtrix là l x d, l là length of sequence, d là dimensionality. #Para là số lượng tham số của mỗi layer. #Mult là số phép nhân trong khi nhân các ma trận.

3.3 Experimental results

3.3.1 The effectiveness of shared weight self-attention

Model	Error Rate	#Para.
tdnn	5.62 ± 0.341	12k
tdnn-blstm	5.79 ± 0.189	20k
swsa	9.81 ± 0.203	8k
tdnn-sa	4.24 ± 0.149	16k
tdnn-swsa	$\mathbf{4.19} \pm 0.191$	12k

Bảng 2: The effectiveness of SWSA

3.3.2 The impact of location of SWSA

Model	Error Rate
tdnn-swsa-13	4.32 ± 0.255
tdnn-swsa-14	4.74 ± 0.259
tdnn-swsa	$\mathbf{4.19} \pm 0.191$

Bảng 3: The impact of location of SWSA. tdnn-swsa-13 có nghĩa SWSA layer được đặt ở layer 3, tdnn-swsa-14 có nghĩa là SWSA layer được đặt ở layer 4.

3.3.3 A comparison with other models

Từ bảng 4, ta có thể thấy tdnn-swsa đạt 4.19% error rate chỉ với 12k parameters bằng 1/20 so với res15. So sánh với res8-narrow and cnn-trad-fpool, tdnn-swsa có độ chính xác hơn hẳn. So với stack-tdnn dùng transfer learning phức tạp, tdnn-swsa có error rate nhỏ hơn.

Model	Error Rate	#Para.
ResNet15	4.2	238k
stack-tdnn	5.7	251k
cnn-trad-fpool3	7.82 ± 0.373	878k
res15	4.12 ± 0.232	239k
res8-narrow	10.69 ± 0.867	20k
tdnn-swsa	4.19 ± 0.191	**12**k

Bảng 4: A comparison with other models

Tài liệu

[1] Sanjeev Khudanpur Vijayaditya Peddinti Daniel Povey. "A time delay neural network architecture for efficient modeling of long temporal contexts". In: *Proc. Interspeech 2015* (2015), pp. 3214–3218.

[2] Pete Warden. "Speech Commands: A Dataset for Limited-Vocabulary Speech Recognition". In: *arXiv preprint arXiv:1804.03209* (2018).

[3] Jiangyan Yi Ye Bai. "A Time Delay Neural Network with Shared Weight Self-Attention for Small-Footprint Keyword Spotting". In: *Interspeech 2019* (2019).