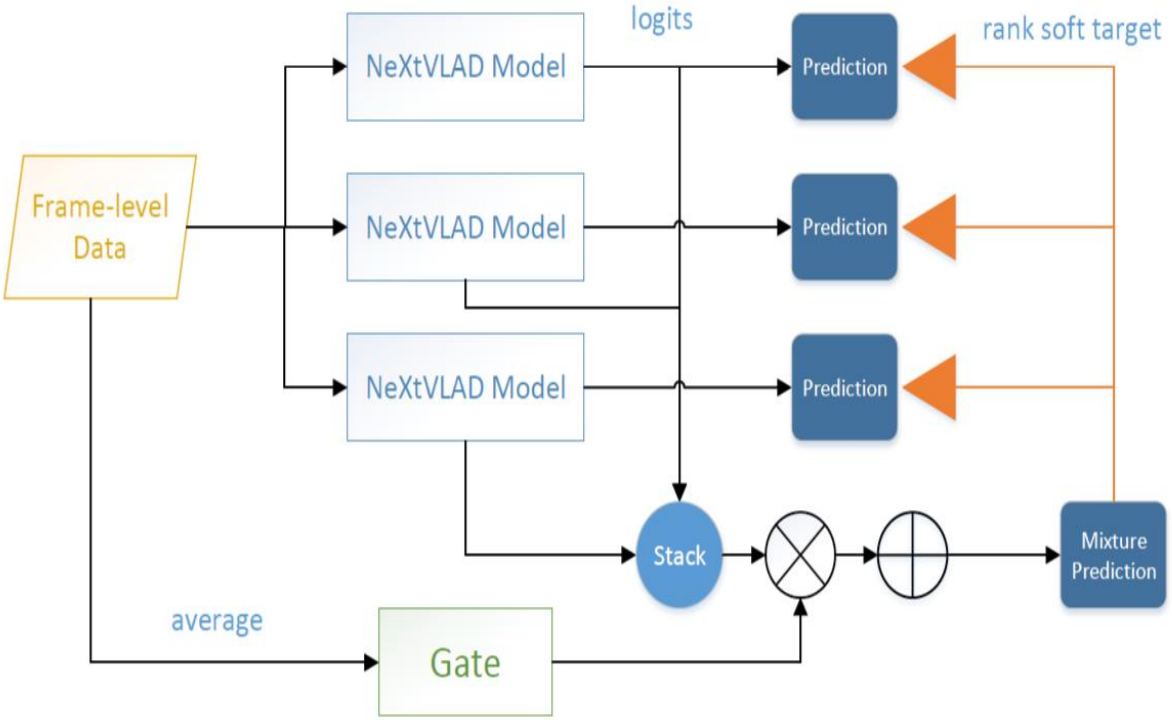
NeXtVLAD: An Efficient Neural Network to Aggregate Frame-level Features for Large-scale Video Classification

Rongcheng Lin, Jing Xiao, Jianping Fan

University of North Carolina at Charlotte

UNC CHARLOTTE

Final solution overview:



Individual loss

distill loss

$$\mathcal{L} = \sum_{m=1}^{3} \mathcal{L}_{bce}^{m} + \mathcal{L}_{bce}^{e} + T^2 * \sum_{m=1}^{3} \mathcal{L}_{kl}^{m,e}$$

mixture loss

[X. Lan 2018]

UNC CHARLOTTE

Table 2. The GAP scores of submissions during the competition. All the other parameters used are (0.5drop, 112K, 2048H). The final submissions are tagged with *

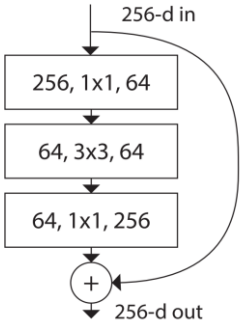
Model	Parameter	Private GAP	Public GAP
single NeXtVLAD(460k steps)	79M	0.87846	0.87910
3 NeXtVLAD (3T, 250k steps)	237M	0.88583	0.88657
3 NeXtVLAD (3T, 346k steps)	237M	0.88681	0.88749
3 NeXtVLAD* (3T, 460k steps)	237M	0.88722	0.88794
3 NeXtVLAD* (3T, 647k steps)	237M	0.88721	0.88792

- 79M parameters ≈ 316M storage using Float32
- 2 nvidia gtx 1080 TI
- 400+ examples/sec with SSD
- About 2 days to reach the optimal results

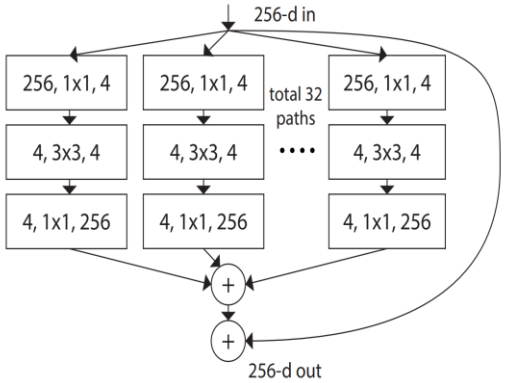
UNC CHARLOTTE

Motivations: feature groups for aggregation?

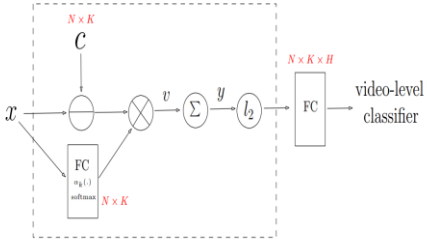
ResNet



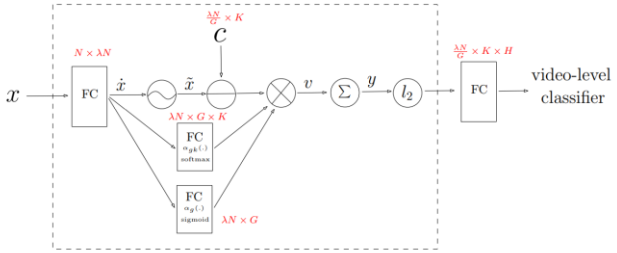
ResNeXt



NetVLAD



NeXtVLAD



UNC CHARLOTTE

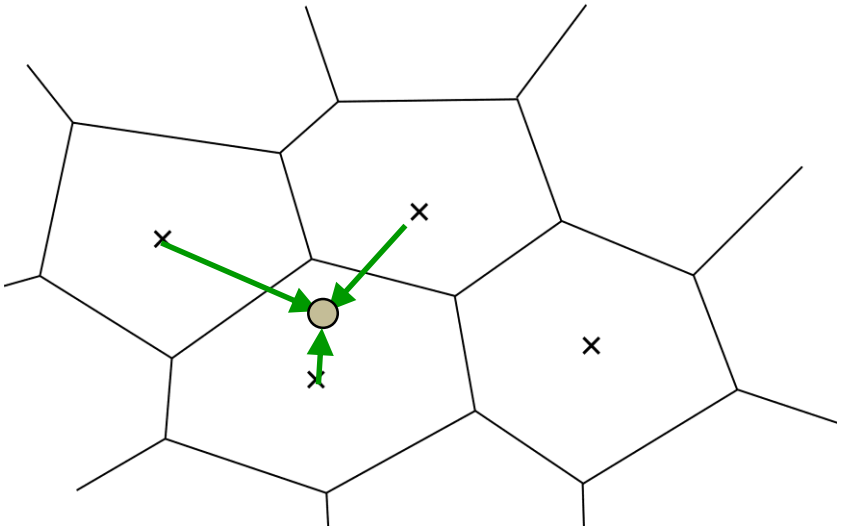
NetVLAD: a learnable pooling approach

Soft assignment of frame feature i to cluster k

$$y_{jk} = \sum_i \alpha_k(x_i)(x_{ij} - c_{kj})$$

Residual vector

Sum over all M frames in the video



$$\alpha_k(x_i) = \frac{e^{w_k^T x_i + b_k}}{\sum_{s=1}^{K} e^{w_s^T x_i + s_t}}$$

[R. Arandjelović, 2016]

UNC CHARLOTTE

NetVLAD: a learnable pooling approach

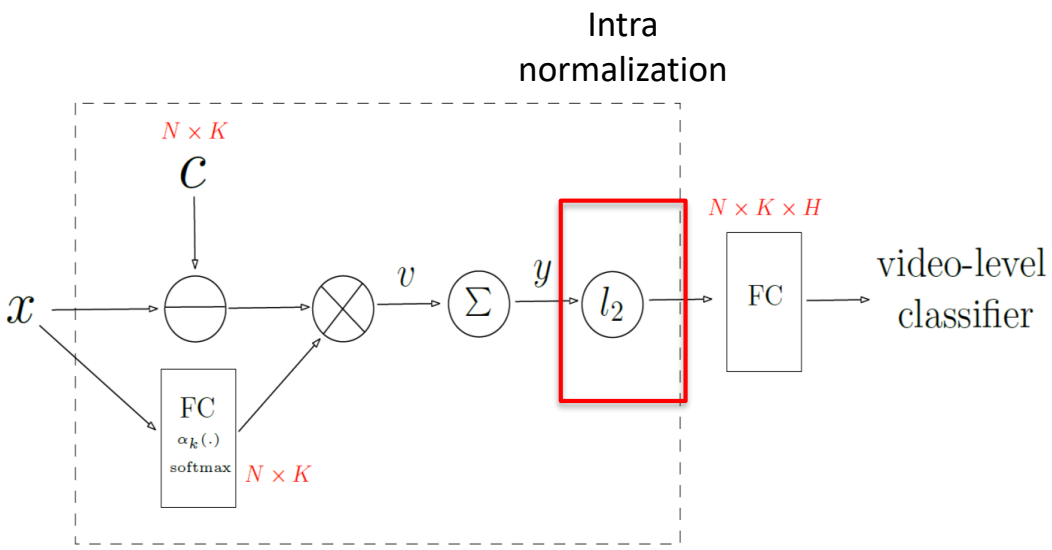


Fig. 1. Schema of NetVLAD model for video classification. Formulas in red denote the number of parameters (ignoring biases or batch normalization). FC means fully-connected layer.

- Parameters Number: $N \times K \times (H + 2)$
- N(input dimension), K(cluster number), H(hidden size)

e.g. N = 1024, K = 256
H = 2048 will result in
537 millions parameters

UNC CHARLOTTE

NeXtVLAD: a mixture of NetVLAD over group features

Attention function over groups

$$y_{jk} = \sum_{g} a_g(x_i) \sum_{i} \alpha_{gk}(x_i)(x_{ij}^{g} - c_{kj})$$

Sum over all groups

Group level NetVLAD aggregation

$$\alpha_g(x_i) = \sigma(w_g^T x_i + b_g)$$

UNC CHARLOTTE

NeXtVLAD: a mixture of NetVLAD on group features

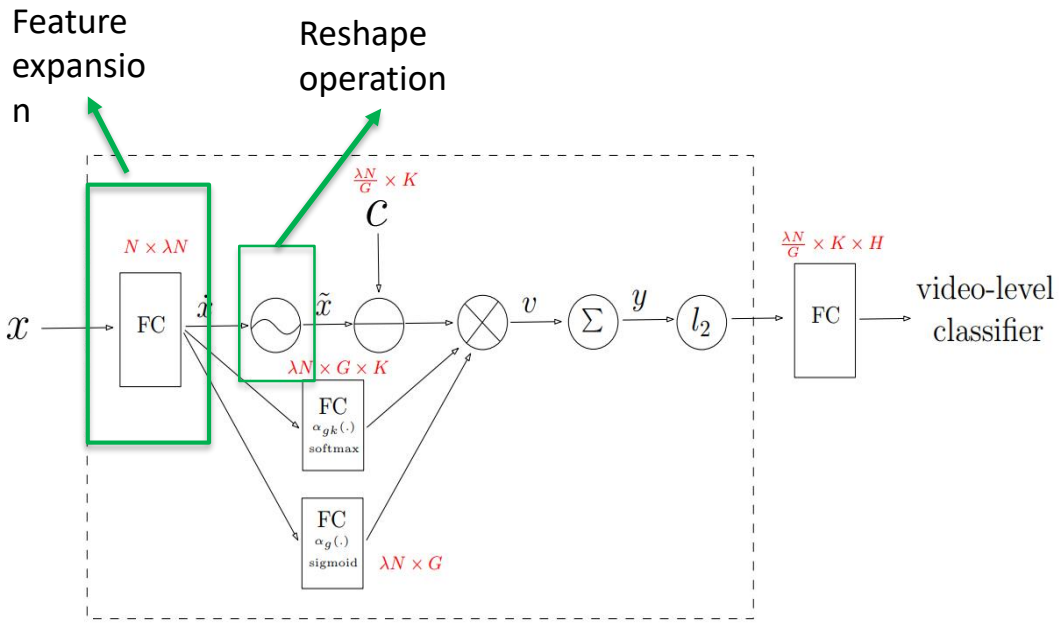


Fig. 2. Schema of our NeXtVLAD network for video classification. Formulas in red denote the number of parameters (ignoring biases or batch normalization). FC represents a fully-connected layer. The wave operation means a reshape transformation.

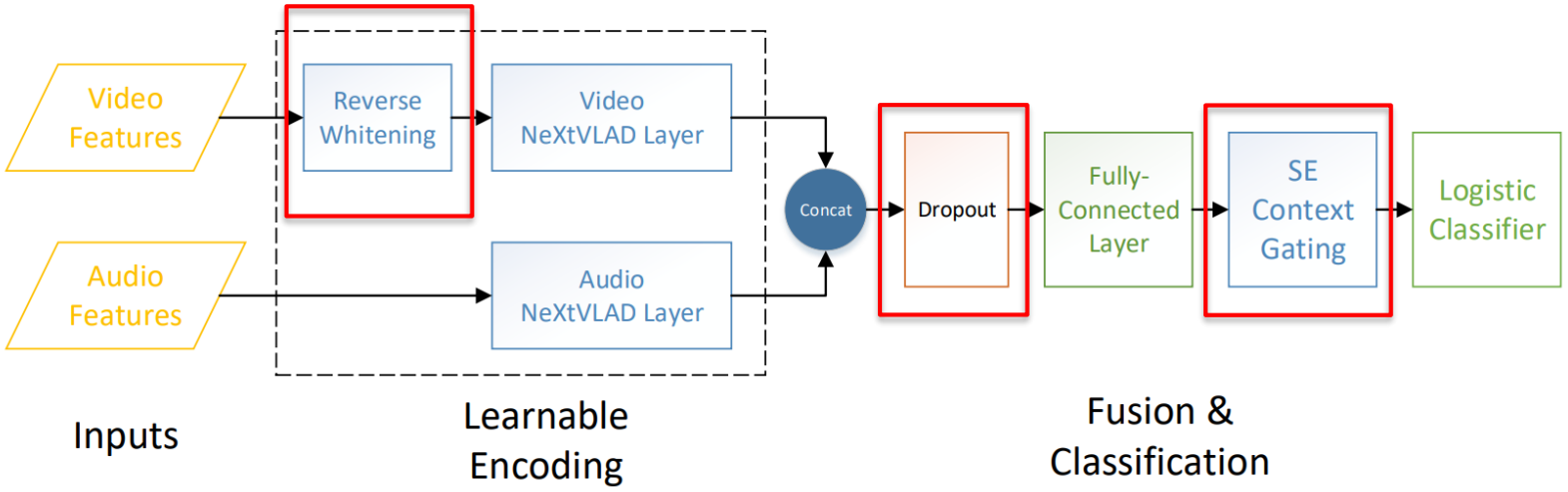
- Parameters Number:

$$\lambda N \times (N + G + K \times (G + \frac{H+1}{G}))$$

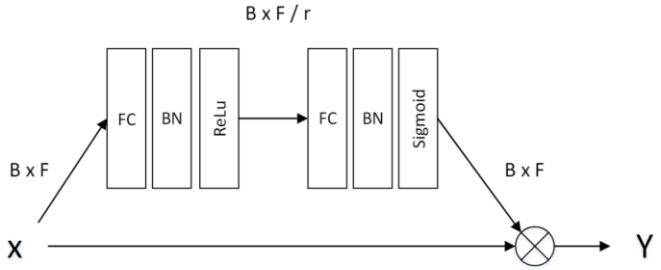
- N(input dimension), G(group number), K(cluster number), H (hidden size), λ(expansion factor)
- About $\frac{G}{\lambda}$ times smaller than NetVLAD with same input and output dimension
- e. g. N=1024, G=8, K=256, H=2048, λ = 2 will results in 140 million parameters

UNC CHARLOTTE

Overview of NeXtVLAD model



- Reverse whitening: $\hat{x}_{ij} = x_{ij} * \sqrt{e_j}$
- SE Context Gating:



[A. Miech 2017]

UNC CHARLOTTE

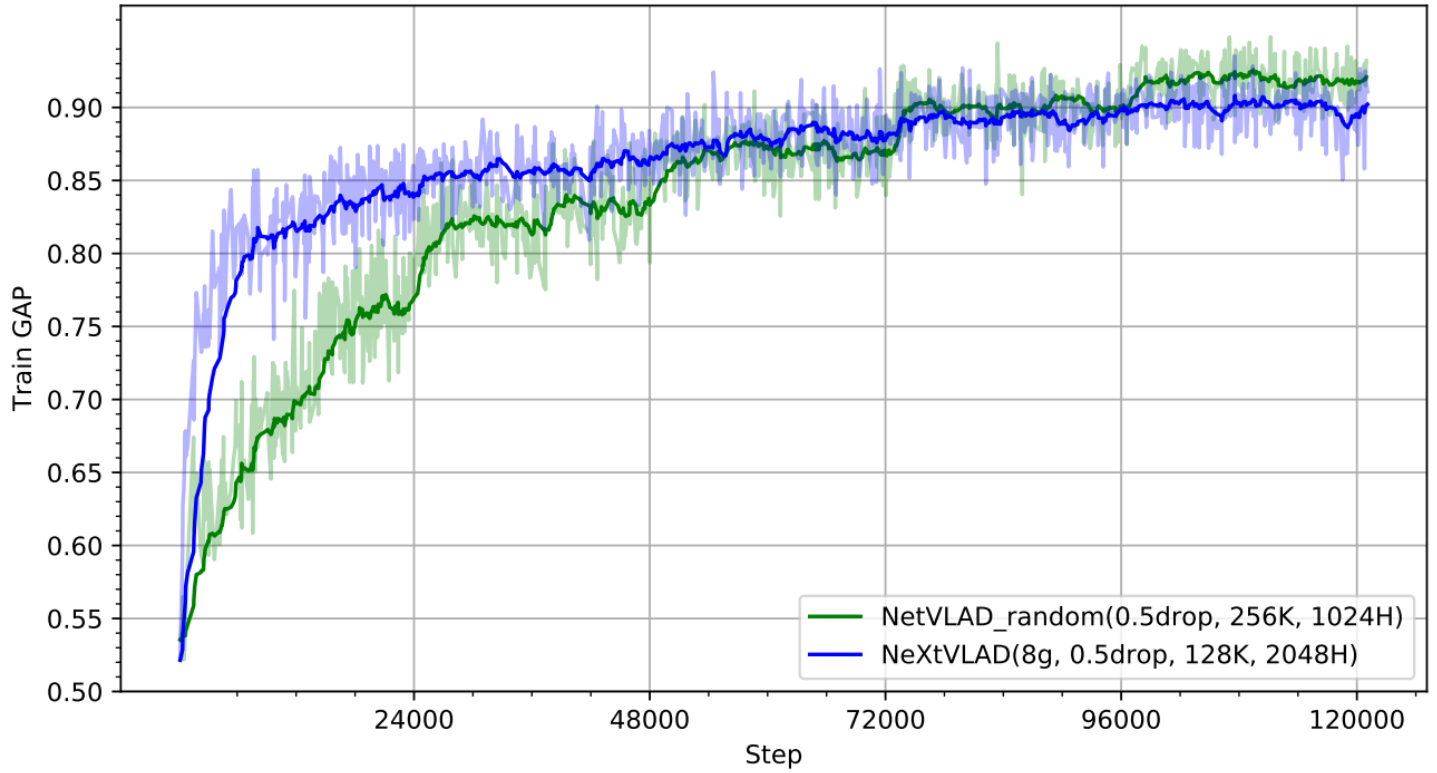


Fig. 6. Training GAP on Youtube-8M dataset. The ticks of x axis are near the end of each epoch.

UNC CHARLOTTE

Single model performance comparison

Table 1. Performance (on local validation partition) comparison for single aggregation models. The parameters inside parenthesis represents (group number G, dropout ratio, cluster number K, hidden size H)

Model	Parameter	GAP
NetVLAD (-, 0.5drop, 128K, 2048H)	297M	0.8474
NetVLAD_random (-, 0.5drop, 256k, 1024H)	274M	0.8507
NetVLAD_small (-, 0.5drop, 128K, 2048H)	88M	0.8582
NeXtVLAD (32G, 0.2drop, 128K, 2048H)	55M	0.8681
NeXtVLAD (16G, 0.2drop, 128K, 2048H)	58M	0.8685
NeXtVLAD (16G, 0.5drop, 128K, 2048H)	58M	0.8697
NeXtVLAD (8G, 0.5drop, 128K, 2048H)	89M	0.8723

UNC CHARLOTTE

Questions & More Details - - >
https://www.kaggle.com/c/youtube8m-2018/discussion/63223
Code -- >
https://github.com/linrongc/youtube-8m

UNC CHARLOTTE