Adaptive Similarity Metric

Sulabh Shrestha
sshres2@gmu.edu
GMU
CS-782, Spring 2021

Metric Learning

- Map data into a representation or embedding space
- Embeddings in the space should have semantic meanings
  - Inputs that are similar should be close to each other
- Similarity depends on the task at hand
  - For images, it’s usually the objects belonging same class

Learning with Triplet Margin Loss

- Embedding model: f(.)
- For each individual triplet, $t_i = (x_i, y_i, z_i)$
- Get embeddings for each example in the triplet:
  - $a_i = f(x_i)$
  - $p_i = f(y_i)$
  - $n_i = f(z_i)$
- Triplet Loss [1]
  - $L_{margin}(t_i) = \max(D(a_i , p_i) - D(a_i , n_i) + m, 0)$
    - D is the distance function
    - m is the chosen margin ($m >= 0$)
  - Encourages anchor to be at least “m” distance farther away from negative than positive
- Margin is fixed for each triplet

[1] Weinberger, Kilian Q., et al. Distance Metric Learning for Large Margin Nearest Neighbor Classification. NIPS 2005

Classes vs Instances

- A class is a group of similar objects that belong to a single super-group
  - Examples: Bottles
- An instance is a specific object within the class
  - Example: Coca-cola glass bottle
- There can be multiple hierarchies
  - Related to Fine-Grained Classification which a different sub-field

Bottles [1]

[1] Ammirato, P., Poirson, P., Park, E., Kosecka, J., & Berg, A. C. (2017). A dataset for developing and benchmarking active vision. 2017 IEEE International Conference on Robotics and Automation (ICRA)

Similarity of Triplets

- All men and women are created equal
- But, all triplets are not
- Margin should be higher for bottom
- But, Margin is fixed
    - Same margin for each triplet

Anchor Positive Negative [1]

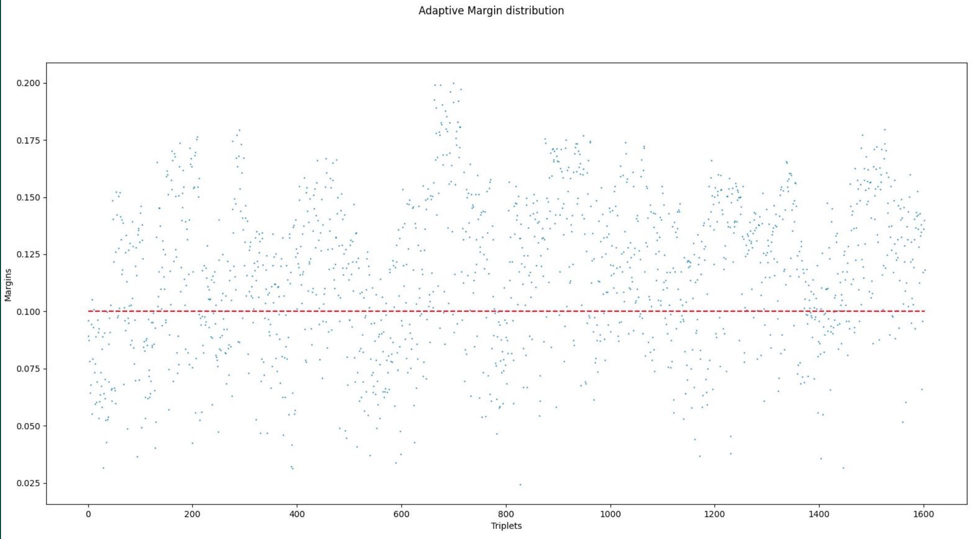
[1] Krause, J., Stark, M., Deng, J., & Fei-Fei, L. (2013). 3D Object Representations for Fine-Grained Categorization. 2013 IEEE International Conference on Computer Vision Workshops

Proposed Method

- Train a classifier, c(.) on the dataset
- Use the classifier (without softmax) as the margin supervision
  - Extract embeddings for the triplet, t_i
    - $ca_i = c(x_i)$
    - $cp_i = c(y_i)$
    - $cn_i = c(z_i)$
  - Calculate the adaptive margin based on the these:
    - $am_i = dist(ca_i, cni) - dist(ca_i, cp_i)$
- Loss function
  - $L_{adapt}(t_i) = max(D(a_i , p_i) - D(a_i , n_i) + am_i, 0)$
- Final loss function
  - Use both fixed margin as well as adaptive margin
  - $L = (1-w) L_{margin} + (w) L_{adapt}$
  - **w = Adaptive Loss weight**

Proposed Method (Nuances)

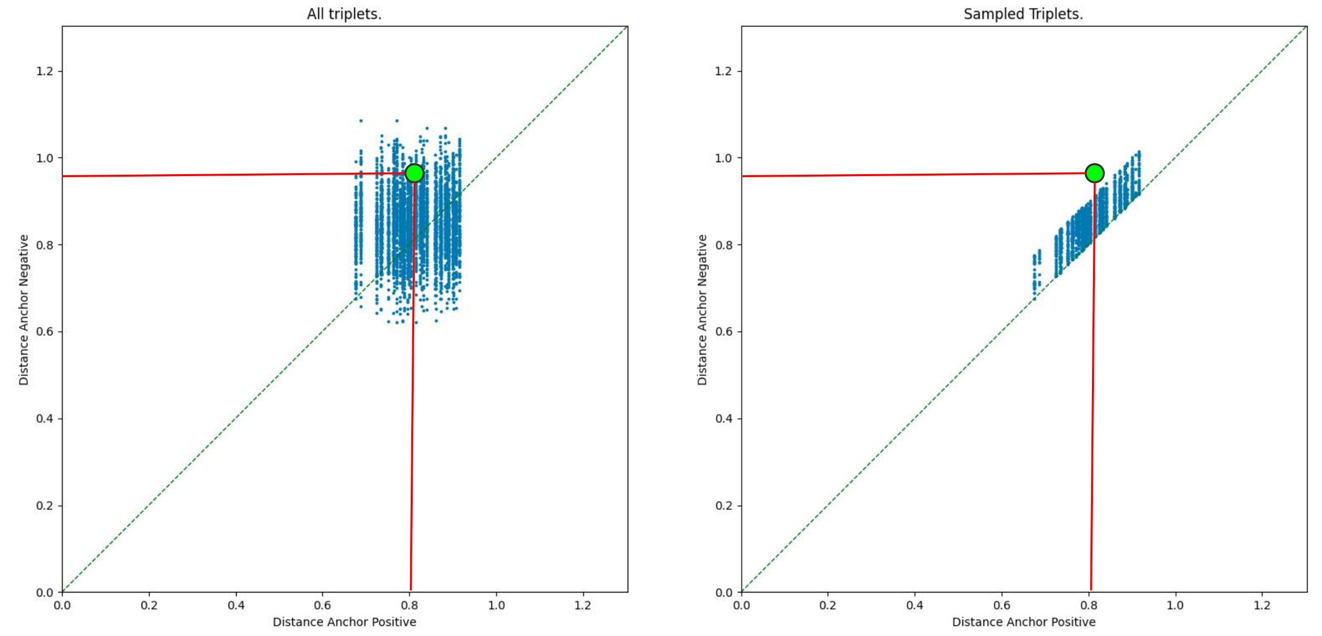
- Margins obtained from the classifiers have a wide range (>1)
  - Used directly, L_{margin} is very high compared to L_{adapt}
  - Normalized using the formula:
    - $am_i = [am_i + \min(am_j)] / \max(am_j) * \text{fixed margin} * 2$
  - Range of adaptive margin is now from 0 to 2 times fixed margin



Distribution of normalized adaptive margins when fixed margin=0.1

Proposed Method (Nuances)

- Sampling has to be done separately for two margins
- Example:
    - D_ap = 0.80   D_an = 0.97   D_ap - D_an = 0.17
    - fixed m   = 0.10
    - adapt m   = 0.20



Dataset

- CARS196 [1]
  - Different types of cars
  - Each class has specific Make, Model, Year of the car
  - ~8k Training images
  - ~8k Testing images
  - 196 classes
    - approximately evenly distributed
    - i.e. ~42 instances per class

GMC Terrain SUV 2012

Tesla Model S Sedan 2012

[1] Krause, J., Stark, M., Deng, J., & Fei-Fei, L. (2013). 3D Object Representations for Fine-Grained Categorization. 2013 IEEE International Conference on Computer Vision Workshops

Implementation Details

- Metric
  - Learning Rate : 0.0001
  - Epochs : 80
  - Fixed Margin : 0.1
  - Embedding dim : 128
  - Classes per batch : 64
  - Instances per class : 2
  - Validation size : 0.05
  - Weight Decay : 0.0005

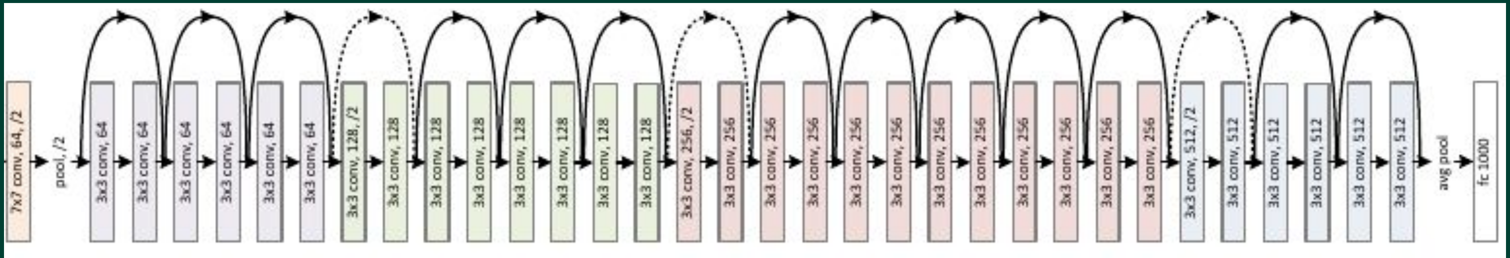
- Classification (2)
  - Network-1 : ResNet18 [1]
  - Epochs-1 : 150
  - Network-2 : ResNet50 [1]
  - Epochs-2 : 80
  - Learning Rate : 0.001
  - Epochs : 80
  - Batch Size : 32
  - Validation size : 0.05
  - Weight Decay : 1e-5

- Adaptive Metric
  - Learning Rate : 0.001
  - Adaptive Loss weight : 0.1:0.9:0.2
  - Classes per batch : 32 (Because 2 models used)

[1] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 770–778. https://doi.org/10.1109/CVPR.2016.90

Backbone Network

- ResNet-34 [1]
  - 34 implies the number of layers
- ResNet-18
  - 18 layers
- ResNet-50
  - 50 layers



ResNet-34 [1]

[1] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 770–778. https://doi.org/10.1109/CVPR.2016.90

Evaluation Criteria

- Recall at K
  - For each input, check if there is **at least 1 class** that exists in nearest K-neighbors
  - Average for all inputs
  - Recall at 1 is the most stringent criteria
- Recall@1 and Recall@5 used

Evaluation Results

- Classifier is pretty good already
- All base metric models gain boost from adaptive margin
- R50-R50 is the best overall

Best per type shown in different color

Model Type	Main model	Adaptive-Model	Recall@1	Recall@5
Classifier	ResNet18	---	0.652	0.843
Classifier	*ResNet50*	---	*0.715*	*0.882*
Metric	ResNet18	---	0.626	0.839
Metric	ResNet50	---	0.680	0.862
Adaptive Metric	ResNet18	ResNet18	0.685	0.857
Adaptive Metric	ResNet18	ResNet50	0.684	0.851
Adaptive Metric	ResNet50	ResNet50	**0.734**	**0.864**

Evaluation Results

- Both SCT and CosFace use different loss function than Triplet

Model Type	Main model	Adaptive-Model	Recall@1	Recall@5
SCT Loss [1]	ResNet18	---	0.732	**0.884** (@4)
CosFace [2][3]	*ResNet50*	---	**0.741**	**---**
Metric	ResNet18	---	0.626	0.839
Metric	ResNet50	---	0.680	0.862
Adaptive Metric	ResNet18	ResNet18	0.685	0.857
Adaptive Metric	ResNet18	ResNet50	0.684	0.851
Adaptive Metric	ResNet50	ResNet50	0.734	0.864

[1]Xuan, H., Stylianou, A., Liu, X., & Pless, R. (2020). Hard Negative Examples are Hard, but Useful. In A. Vedaldi, H. Bischof, T. Brox, & J.-M. Frahm (Eds.), Computer Vision – ECCV 2020
[2]Wang, H., Wang, Y., Zhou, Z., Ji, X., Gong, D., Zhou, J., Li, Z., & Liu, W. (2018). CosFace: Large Margin Cosine Loss for Deep Face Recognition. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition
[3]Musgrave, K., Belongie, S., & Lim, S.-N. (2020). A Metric Learning Reality Check. ArXiv:2003.08505 [Cs]. http://arxiv.org/abs/2003.08505

Ablation Studies (Weight)

- w = 0.5 is best except for R18-R50
- For R@5, w=0.9 is the best choice for **all**

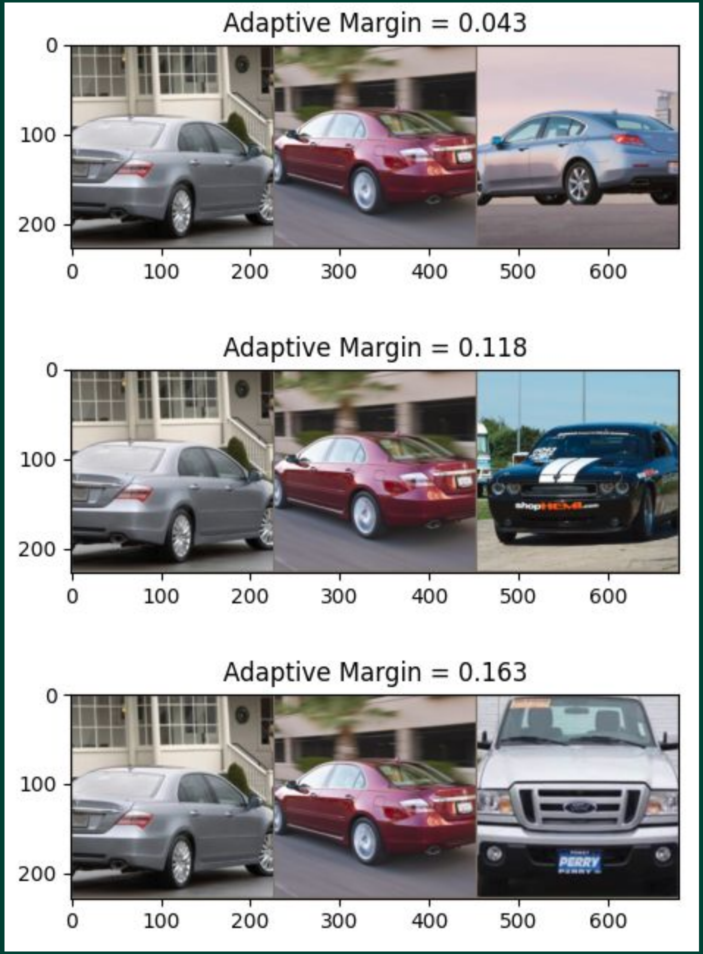
ResNet18-ResNet18		
Adaptive Loss Weight	R@1	R@5
0.1	0.663	0.839
0.3	0.682	0.844
0.5	**0.685**	0.857
0.7	0.679	0.857
0.9	0.677	**0.861**

ResNet18-ResNet50		
Adaptive Loss Weight	R@1	R@5
0.1	**0.690**	0.847
0.3	0.687	0.841
0.5	0.684	0.851
0.7	0.681	0.851
0.9	0.653	**0.852**

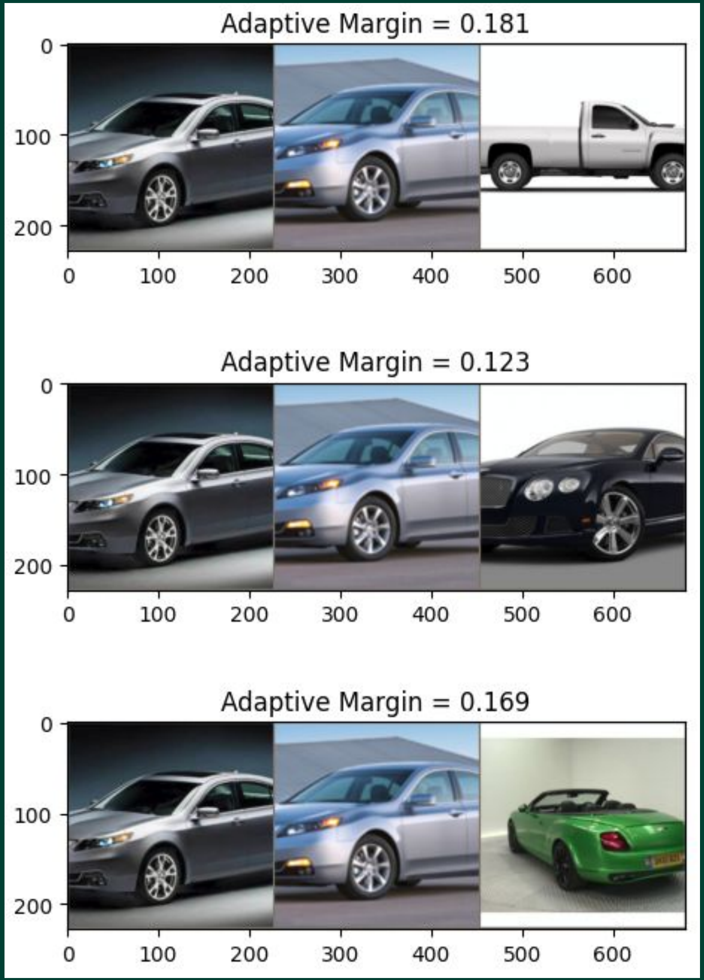
ResNet50-ResNet50		
Adaptive Loss Weight	R@1	R@5
0.1	0.723	0.857
0.3	---	---
0.5	**0.734**	0.864
0.7	---	---
0.9	0.689	**0.872**

Hyper-parameter searches

- For Metric models
  - Fixed LR performed very low (R@1~0.01)
  - Reduce LR with starting LR = 0.0001 was best
- For Adaptive metric models:
  - Reduce LR on validation saturation was used but not effective as fixed LR



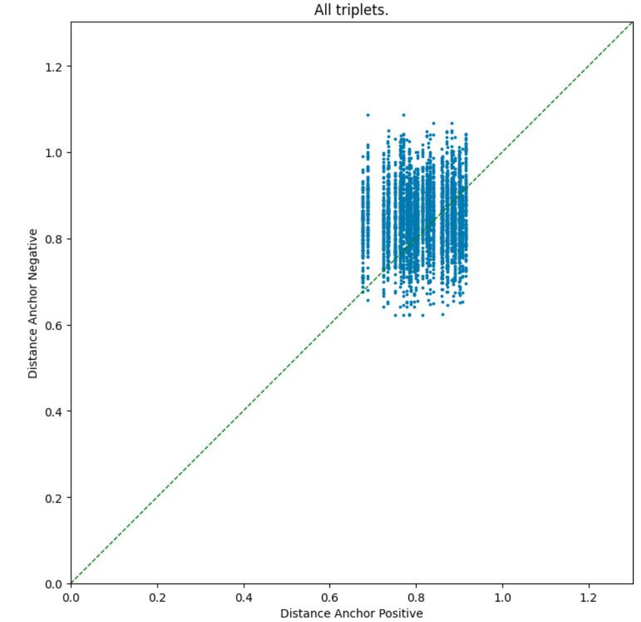
Anchor Positive Negative



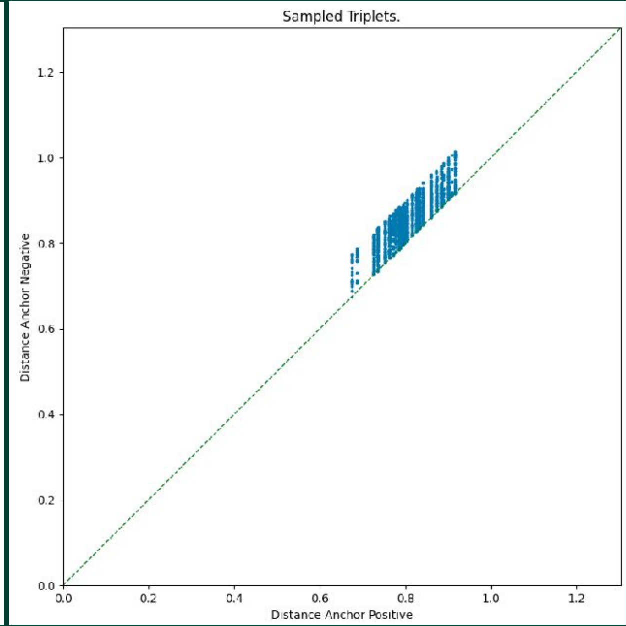
Anchor Positive Negative

Triplets Sampled

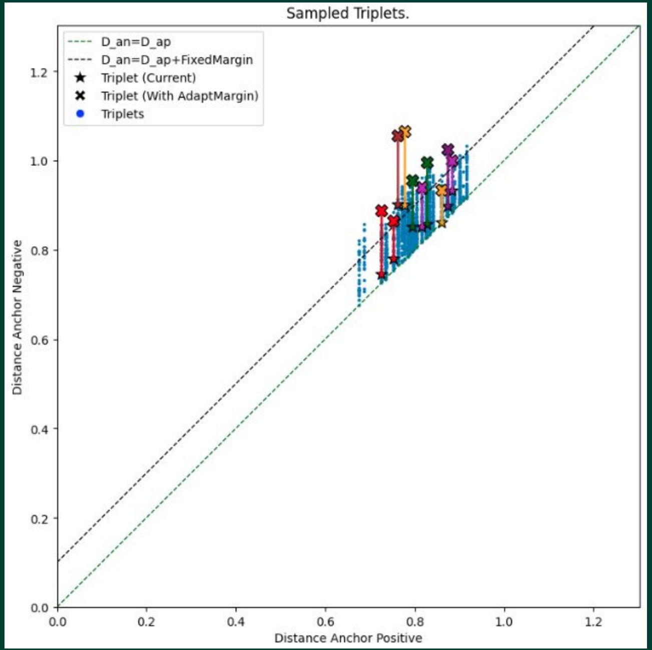
- Adaptive triplets may be different because different margin



All



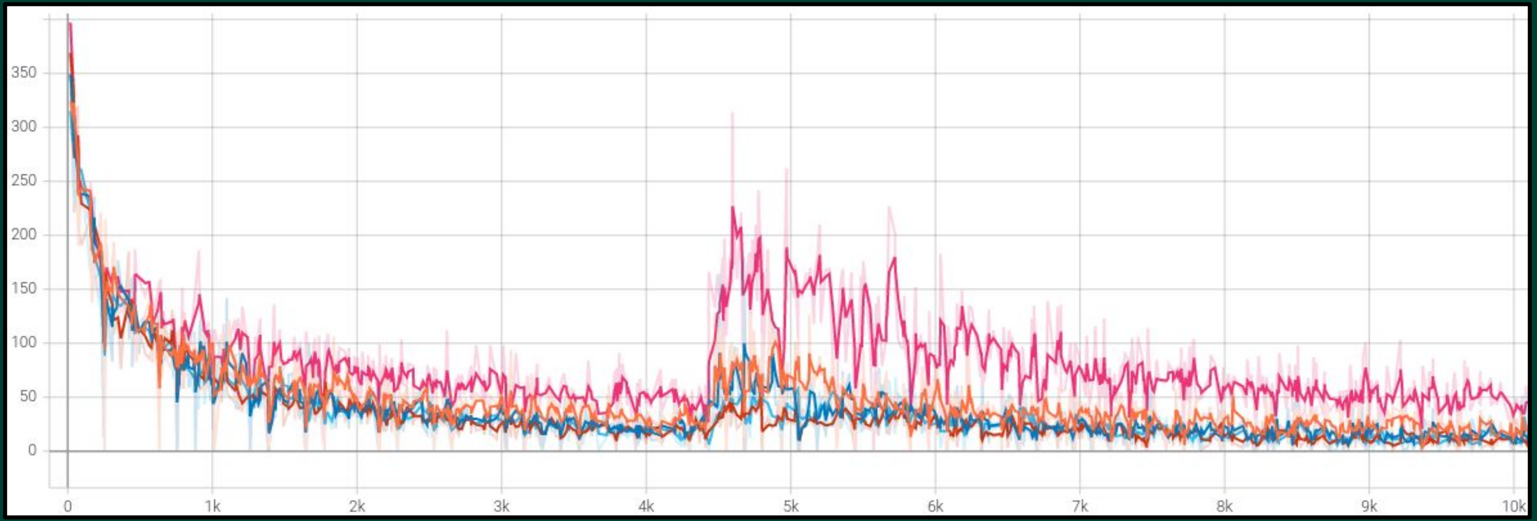
Fixed Margin Sampled

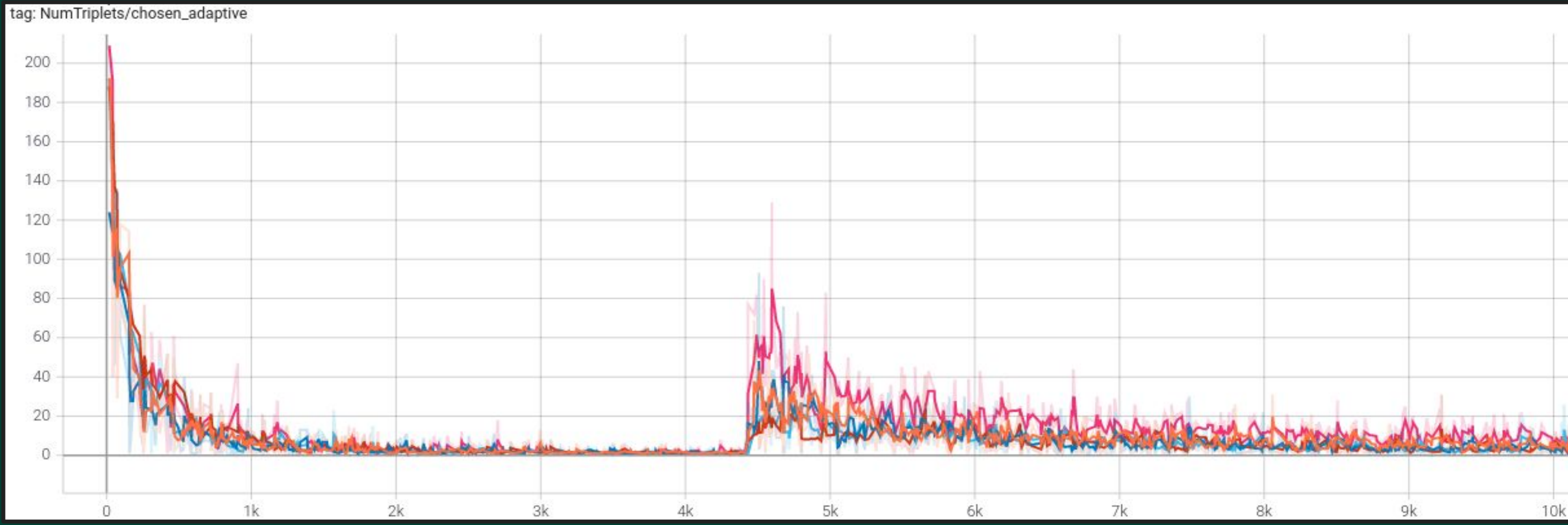


Adaptive Margin Sampled

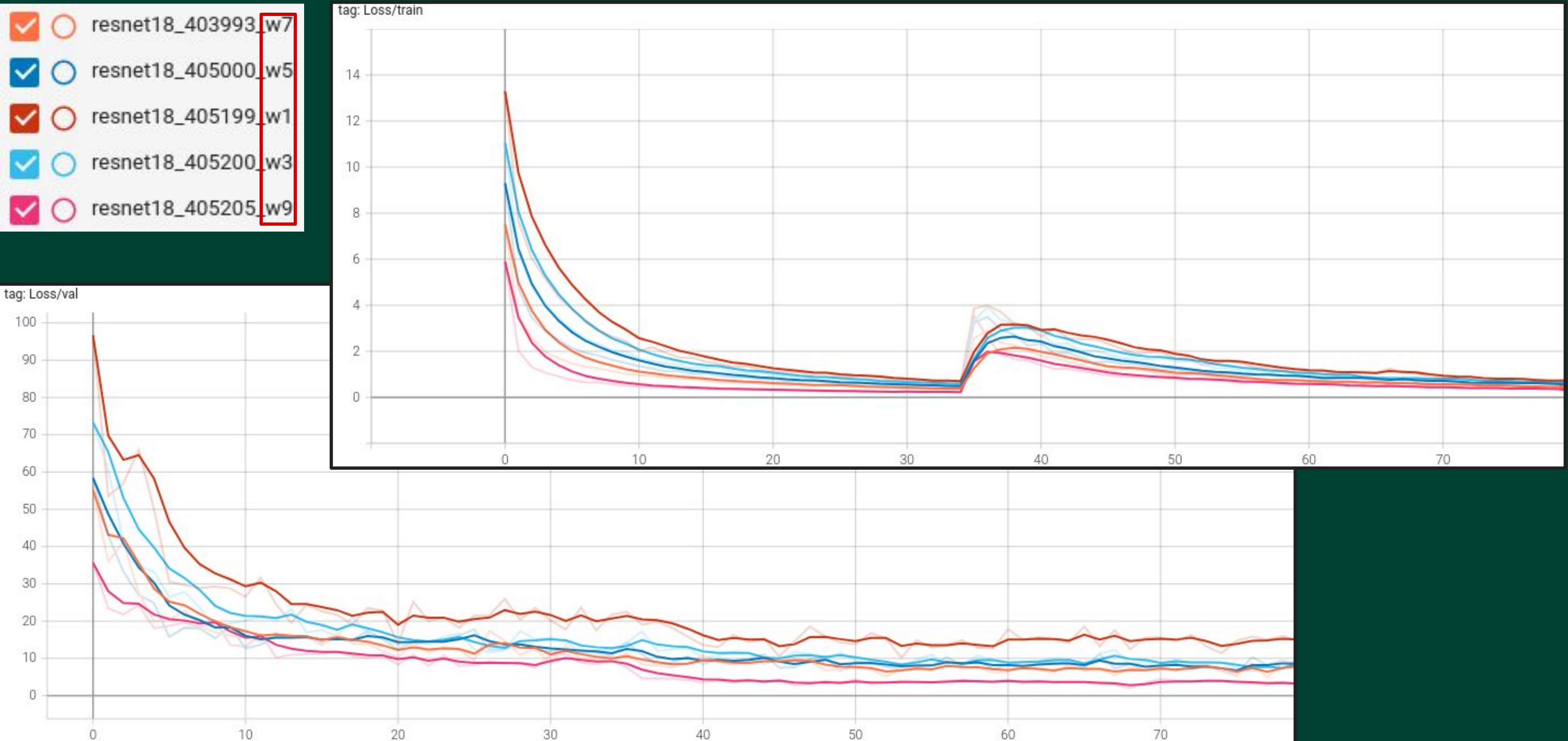
Triplets Sampled

- resnet18_403993_w7
- resnet18_405000_w5
- resnet18_405199_w1
- resnet18_405200_w3
- resnet18_405205_w9





Train and Validation loss



Future Work

- See how the adaptive margin works with other loss functions
- Check performance for scenario where classes are not present
  - Select and label a few classes
  - Train on these examples and use as supervision
- Classifier from related domain but different different dataset