INFO 7375 - Neural Networks & AI

Quiz - 7

Submitted By:

Abdul Haseeb Khan
NUID: 002844724
khan.abdulh@northeastern.edu

Describe the triple loss function and explain why is it needed.

The **triplet loss** is a loss function used to train neural networks to learn useful embeddings (representations) where similar items are close together in the embedding space while dissimilar items are far apart.

The triplet loss operates on triplets of examples:
- **Anchor (a)**: A reference sample
- **Positive (p)**: A sample similar/same class as the anchor
- **Negative (n)**: A sample dissimilar/different class from the anchor

The loss function is formulated as:
$L = \max(0, ||f(a) - f(p)||^2 - ||f(a) - f(n)||^2 + \alpha)$

Where:
- f(x) is the embedding function (neural network output)
- $||\cdot||$ denotes the distance metric (typically Euclidean)
- α is the margin (a hyperparameter that enforces minimum separation)

1. Direct Optimization of Similarity Unlike classification losses (softmax/cross-entropy) that learn to distinguish between fixed classes, triplet loss directly optimizes the relative distances between examples. This makes it ideal for:
- Face verification/recognition
- Person re-identification
- Image retrieval systems
- One-shot/few-shot learning scenarios

2. Handles Large/Open Class Sets Traditional classification requires retraining when new classes are added. With triplet loss, you can:
- Add new identities without retraining
- Work with thousands of classes (e.g., faces)
- Handle classes with very few examples

3. Creates Meaningful Embedding Space The learned representations are geometrically meaningful:
- Distance directly corresponds to similarity
- Can use simple k-NN for classification
- Enables clustering and visualization

4. Better Generalization By learning relative relationships rather than absolute class boundaries, the model generalizes better to:
- Unseen classes
- Different domains
- Varying conditions (lighting, pose, etc.)

Key Implementation Considerations

Hard Triplet Mining: Not all triplets are equally informative. Strategies include:
- Hard positive: furthest positive from anchor
- Hard negative: closest negative to anchor
- Semi-hard negative: negative within the margin but farther than positive

Batch Construction: Careful sampling is crucial:

- Need multiple examples per class in each batch
- Balance between easy and hard triplets
- Online triplet generation during training

This loss function has been fundamental in applications like FaceNet for face recognition and has inspired variations like quadruplet loss and n-pair loss for even better performance in metric learning tasks.