

## Assignment 2

Deadline: Friday, 29th May (23:59)

15th May, 2020

### Question 2: Triplet networks & one-shot learning (10pt)

In practice 4b.4, we train a Siamese network for one-shot learning task on the Omniglot dataset. In this assignment, we will work on the same data set with the same task but extend it to triplet networks, we will also compare our model performance under different triplet selection method. The assignment contains the following 4 tasks.

#### Task 1.1: Build the triplet network (3pt)

You can find the formula of the triplet loss function in our lecture note. When training our model, make sure the network achieves a smaller loss than the margin and the network does not collapse all representations to zero vectors.

HINT: If you experience problems to achieve this goal, it might be helpful to tinker the learning rate.

#### Task 2.2: Define triplet loss (2pt)

You can find the formula of the triplet loss function in our lecture note. When training our model, make sure the network achieves a smaller loss than the margin and the network does not collapse all representations to zero vectors.

HINT: If you experience problems to achieve this goal, it might be helpful to tinker the learning rate, you can also play with the margin value to get better performance.

#### Task 2.3: Select triplets for training (3pt)

We have two different options for the triplet selection method, and we will compare the model performance under these two selection methods after building our model.

- a) Random triplets selection, including the following steps:
  - Pick one random class for anchor.
  - Pick two different random picture for this class, as the anchor and positive images.
  - Pick another class for Negative, different from anchor class.
  - Pick one random picture from the negative class.
- b) Hard triplets selection. For easy implement, for a picked anchor, positive pair, we will choose the hardest negative to form a hard triplet, that means, after picking an anchor, positive image, we will choose the negative image which is nearest from anchor image from a negative class, ie: " $-d(a,n)$ " can get the maximum value. The whole process including the following steps:
  - Pick one random class for anchor.
  - Pick two different random picture for this class, as the anchor and positive images.

- Pick another class for Negative, different from anchor class.
- Pick one hardest picture from the negative class.

HINT: when picking the hardest negative, you may need the `model.predict` to get the embedding of images, then calculate the distances

### Task 2.4: One-shot learning with different selection method (2pt)

With different triplets selecting method (random and hard), we will train our model and evaluate the model by one-shot learning accuracy.

- You need to explicitly state the accuracy under different triplets selecting method
- When evaluating model with `test_one_shot` function, set the number (k) of evaluation one-shot tasks to be 250, then calculate the average accuracy

HINT: After training our model with random selection method, before train model under hard triplets selection, you should re-build our model (re-run the cell in Task 2.1) to initialize our model and prevent re-use the trained model of random selection.

HINT: you can re-use some code from practice 4b.4.