

# Understanding Siamese networks

SERI LEE\*

Seoul National University  
sally20921@snu.ac.kr

January 9, 2021

## Abstract

*Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.*

## I. INTRODUCTION

A Siamese network [?], as the name suggests, is an architecture with two parallel layers. In this architecture, instead of model learning to classify its inputs using classification loss functions, the model learns to differentiate between two given inputs. It compares two inputs based on a similarity metric and checks whether they are the same. Similar to any deep learning architecture, a Siamese network also has two phases—a training and a testing phase. But, for a one-shot learning approach (as we won't have a lot of data points), we will be training the model architecture on one dataset and testing it on a different dataset. To put this in simpler terms, we learn image embeddings using a supervised metric-based approach with Siamese neural networks, and then reuse that network's features for one-shot learning without fine-tuning or retraining.

Extraction of good features for machine learning algorithms plays a crucial role in determining the efficiency of a model. In various scenarios, it is proven to be either computationally expensive or difficult when we have

limited data available.

Our objective is to train a network to understand whether two images or sounds are the same. The key idea might sound similar to transfer learning, but it is a little different. Siamese networks learn these features using the contrastive loss function. Secondly, a Siamese network approach is only valid for similar domains, as it also needs to take care of domain adaptation, that is, it needs to try to ensure that our training and testing datasets are close in terms of the domain.

## II. ARCHITECTURE

A Siamese network consists of two identical neural networks that share similar parameters, each head taking one input data point. In the middle layer, we extract similar kinds of features, as weights and biases are the same. The last layer of these networks are fed to a contrastive loss function layer, which calculates the similarity between the two inputs.

One question you might have is why do Siamese networks' layers share parameters? If we are already putting the effort into changing the loss function, won't it help us to train the

---

\*A thank you or further information

layers separately? There are two major reasons why we are not training layers separately.

- For every layer, we have thousands of parameters being added. Therefore, similar to how we do in a convolutional neural network approach where we share parameters, we can optimize the network faster.
- Sharing weights guarantees that two similar images won't be mapped to different locations in the feature embeddings space.

Feature embeddings are the projection of features to some higher dimensional space, also known as feature embeddings space, depending upon the task we want to achieve.

The distance function decides if the output vectors are close enough to be similar. The Neural Network transforms the input into a properties vector.

### III. PREPROCESSING

For training a Siamese network, we need to apply a special kind of preprocessing to the dataset. While preprocessing the dataset, we have to carefully create pairs of data points as follows: (1) Pairs of similar images (2) Pairs of dissimilar images. We also need to create labels accordingly for similar data points ( $y = 1$ ) and dissimilar data points ( $y = 0$ ); then, each pair is fed to the Siamese architecture. At the end of the layer, the Siamese network uses a differentiating form of the loss function to learn the differentiating features across layers. Commonly, we use just two types of function for Siamese networks-the contrastive loss function, and the triplet loss function.

### IV. CONTRASTIVE LOSS FUNCTION

The whole idea of using Siamese architecture is not to classify between classes but to learn to discriminate between inputs. So, it requires a differentiating form of the loss function known as the contrastive loss function.

Though the contrastive loss function is a good method for learning discriminative features, in other modified versions of the Siamese

architecture, the contrastive loss function isn't able to learn decision boundaries very clearly. In this case, we can use a new loss function known as triplet loss, which helps the architecture to get better results.

### V. TRIPLET LOSS FUNCTION

The triplet loss function is an alternative to the contrastive loss function. It has convergence advantages over contrastive loss functions. To learn about the triplet loss function, first, we need to define data points in pairs as follows:

- Anchor : The main data point
- Positive : A data point similar to Anchor
- Negative : A different data point than Anchor

The triplet loss function converges better than the contrastive loss function because it considers three examples at a time, maintaining the distance between the Positive and Negative points, thereby learning decision boundaries more accurately, whereas the contrastive loss function only considers pairwise examples at a time, so in a sense, it is more greedy, which affects decision boundaries.