Using Machine Learning to Understand the Topology of Knots

Joseph M. Cameron - 2117625

December 15, 2017

1 Status report

1.1 Proposal

1.1.1 Motivation

Knots have played an extremely important role throughout human history. In ancient times, the use and development of knots contributed to advances in agriculture, transport and construction. Thus, knots have played a pivotal role in advancing human civilisation into the modern era. In fact today, knots are still useful in agriculture, science and recreation among many other fields. With such a wide array of knots still in use, it is important to be able to classify and fully understand these individual knots and their properties. For example, knots are commonly used in mountain climbing, where the life of a climber may depend on the properties of a knot that is tied in climbing rope. It would be extremely useful to automatically identify/classify the knot as a means of checking the correct knot has been tied. If the correct knot has not been tied, or it has been tied at an incorrect tension, it may also be useful to show how to fix these errors. With the application of machine learning, the three-dimensional topology of knots may be better understood. Furthermore, a neural network could begin to understand and recover the topology of a knot in order to provide extra insight.

1.1.2 Aims

This project will develop deep learning software that can classify ten individual knots, and independently generate manipulations of those ten knots. The classification will be accomplished through the use of a convolutional neural network (CNN), and the recovery and generation of knot topology will be accomplished through the use of an auto-encoder and generative adversarial network (GAN) setup. Furthermore, the project will include software that can automate the creation of datasets that can be used to train each individual neural network. Deep learning visualisation techniques, such as the use of the t-SNE algorithm, will be used to display representations learned by the neural networks and evaluate the effectiveness of the deep learning tasks. There will also be significant analysis into what techniques, neural network models and training data help or hinder the accuracy and effectiveness of the deep learning tasks.

1.2 Progress

- Language and Deep Learning Framework chosen: The project will be implemented in Python 2.7 using Keras, with a Tensorflow back-end, as a deep learning framework.
- Background research conducted on deep learning techniques, convolutional neural networks, generative adversarial networks and dataset creation/management.
- Creation of controlled and wild (externally sourced) datasets containing images of all ten knots complete.
- Fully completed and tested a script that can copy all images from an existing dataset and then resize all images to a specific size. Used to alter datasets and make them appropriate as inputs to the neural networks.
- Dataset augmentation complete. This combats overfitting.
- Developed a convolutional neural network capable of accurately classifying the topology of five different knots.
- Developed an initial version of a convolutional neural network capable of accurately classifying the topology of ten different knots and the tension at which they have been tied.
- Developed an initial version of a python script that utilises the Blender API to create three dimensional representations of knots and save images of these representations from many different points of view under variable conditions.
- Developed an initial version of software utilising the t-SNE algorithm to visualise the representations of knots learned by neural networks.

1.3 Problems and risks

1.3.1 Problems

- Tricky to observe what features the neural networks consider when learning knot representations.
- Many knots to consider for classification. Much time was taken to carefully pick a diverse selection of knots to classify.
- Many variables to control in the curation of datasets. Much time was taken to carefully consider what variables should be analysed.

1.3.2 Risks

- Tricky to observe what features the neural networks consider when learning knot representations. **Mitigation:** will improve software that is responsible for visualisation and conduct further research on deep learning visualisation techniques such as t-SNE.
- Unclear whether the auto-encoder/GAN setup will accurately recover the three-dimensional topology of knots. **Mitigation:** will develop the auto-encoder/GAN setup and then evaluate it using t-SNE, with which I will be able to effectively analyse the deep learning behaviour regardless of the results.

1.4 Plan

Semester 2

- Week 1: Fully test the Blender python script and 10-Knot Classification CNN.
  – **Deliverable:** A python script that can create a new dataset containing images of animated knots made in Blender and a convolutional neural network that can classify ten different knots with visualisation of the representations learned.

- Week 2-4: Develop an auto-encoder and a generative adversarial network that can recover and manipulate knot topology.
  – **Deliverable:** An auto-encoder/GAN setup that can take a classification of knot type and tension (from the convolutional neural network) and return a new visual representation of that knot tied at a different tension.

- Week 5: Full analysis and evaluation of findings observed.
  – **Deliverable:** Detailed evaluation document detailing why each finding occurs and a detailed analysis of what factors influence the classification/recovery of knot topology.

- Week 6: Final implementation of software. Final code-refactoring.
  – **Deliverable:** Polished implementation of all software on the project repository.

- Week 6-10: Ongoing evaluation of the deep learning tasks.
  – **Deliverable:** Visual aids, such as graphs, to effectively communicate findings from the project.

- Week 6-10: Write the dissertation.
  – **Deliverable:** Multiple drafts of the dissertation submitted to the supervisor with plenty of time before the deadline.